Explainability Requirement in Blockchain Smart Contracts: A Human-Centred Approach

By

Hanouf Al Ghanmi

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
July 2024

ABSTRACT

Blockchain smart contracts have emerged as a transformative technology, enabling the automation and execution of contractual agreements. These self-executing software programs leverage blockchain's distributed and immutable nature to eliminate the need for third-party intermediaries. However, this new paradigm of automation and authority introduces a complex environment with technical intricacies that users are expected to understand and trust. The irreversible nature of blockchain decisions exacerbates these issues, as any mistake or misuse cannot be rectified. Current smart contract designs often neglect human-centric approaches and the exploration of trustworthiness characteristics, such as explainability. Explainability, a renowned requirement in Explainable Artificial Intelligence (XAI) aimed at enhancing human understandability, transparency and trust, has yet to be thoroughly examined in the context of smart contracts. A noticeable gap exists in the literature concerning the early development of explainability requirements, including established methods and frameworks for addressing requirements analysis phases, design principles, evaluation of their necessity and trade-offs.

Therefore, this thesis aims to advance the field of blockchain smart contract systems by introducing explainability as a design concern, fundamentally prompting requirements engineers and designers to cater to this concern during the early development phases. Specifically, we provide guidelines for explainability requirements analysis, addressing what, why, when and to whom to explain. We propose design principles for integrating explainability into the early stages of development. To tailor explainability further, we propose a human-centred

framework for determining information requirements in smart contract explanations, utilising situational awareness theories to address the 'what to explain' aspect. Additionally, we present 'explainability purposes' as an integral resource in evaluating and designing explainability. Our approach includes a novel evaluation framework inspired by the metacognitive explanation-based theory of surprise, addressing the 'why to explain' aspect.

The proposed approaches have been evaluated through qualitative validations and expert feedback. We have illustrated the added value and constraints of explainability requirements in smart contracts by presenting case studies drawn from literature, industry scenarios and real-world projects. This study informs requirements engineers and designers regarding how to elicit, design and evaluate the need for explainability requirements, contributing to the advancement of the early development of smart contracts.

To Joanne and Faisal, my wonderful children.

Your boundless energy and love have fueled my determination to reach this milestone.

This work is dedicated to you, your dreams, and the limitless possibilities that lie ahead.

Always remember, you can achieve anything.

ACKNOWLEDGMENTS

I would like to extend my deepest gratitude to my supervisor, Dr. Rami Bahsoon, for his unwavering support and guidance throughout my PhD journey. His encouragement to explore and develop my own ideas has been instrumental in my growth as a researcher. Dr. Bahsoon's expertise and constructive feedback have been invaluable in shaping my work and leading to its publication in prestigious journals. Beyond his academic mentorship, his empathy and patience have given me the confidence and resilience to navigate this challenging path. Thank you, Dr. Bahsoon, for your outstanding mentorship and dedication.

I also extend my sincere thanks to my thesis group members, Dr. Leandro L. Minku and Dr. Kashif Rajpoot, for their valuable insights, support, and continuous feedback. Additionally, special thanks to Dr. Sabreen Ahmadjee for her invaluable guidance and friendship throughout my PhD journey—her support has been a true source of strength during challenging times. I am also grateful to researchers and experts Wendy Yanez, Hayatullahi Adeyemo, Mohammed Aljasser, and Saleh Alibrahim for their invaluable contributions.

Acknowledging the incredible support behind this PhD journey, I turn to the people who are the heart and soul of my journey—my family and friends. To my father Ahmed, whose endless support and encouragement have been my guiding star, and whose dedication and work ethic have always been my inspiration; to my children Joanne and Faisal, who have stood by me through all the ups and downs, providing endless joy and motivation; to my mother and siblings, whose unwavering love and constant support have been my foundation; and to my dearest friends, whose companionship has made this journey memorable. Thank you all for being my pillars of strength and for believing in me.

Contents

List of Figures

List of Tables

Key Term Definitions

Blockchain	A decentralised digital ledger that securely records transactions across multiple computers, creating an immutable chain of data blocks.
Transaction	A recorded action on the blockchain, typically involving the transfer of digital assets or information between accounts.
Smart Contract	Self-executing code on the blockchain that enforces the terms of an agreement automatically when conditions are met.
Agreement	An understanding or arrangement between parties that defines mutual rights and obligations. In a blockchain context, agreements often form the basis of automated processes within smart contracts.
Consensus	A process used in blockchain networks to achieve agreement on the validity of transactions among distributed participants.
Immutability	A property of blockchain data that ensures once records are added, they cannot be changed or removed, creating a permanent and secure transaction history.
Enforceability	Enforceability in the context of blockchain refers to the automatic execution of terms within a smart contract to eliminate the need for intermediaries.

Decentralisation	The distribution of control and decision-making across a network rather than relying on a central authority.
Governance	The structure of rules and decision-making processes used to manage a blockchain network or decentralised organisation.
User-Level Decision	A decision made within a smart contract that directly impacts an individual user or group of users.
System-Level Decision	A decision within a smart contract or blockchain network that affects the entire system or all participants.
DApp	Decentralised Application that runs on a blockchain network rather than a centralised server.
Ethereum	A decentralised blockchain platform that enables the creation and execution of smart contracts and decentralised applications (DApps).
Solidity	A high-level programming language specifically designed for writing smart contracts on the Ethereum blockchain.
EVM	The Ethereum Virtual Machine which is the runtime environment for executing smart contracts on the Ethereum blockchain.
Bytecode	A low-level, optimised code format that is generated by compiling high-level programming languages
Gas	A unit of measurement for computational work on the blockchain, particularly on Ethereum. Gas fees are paid in cryptocurrency to compensate for the resources required to execute transactions or run smart contracts.

Chapter One

Introduction

1.1 Overview

Blockchain technology has revolutionised the execution of agreements, introducing smart contracts as a cornerstone of this innovation. Smart contracts are a self-executing software programs that run applications on the blockchain according to predefined conditions [208, 307, 6]. These contracts leverage the blockchain's distributed and immutable nature to eliminate the need for third-party intermediaries [164, 333].

Various blockchain platforms have been developed to support the creation and execution of smart contract decentralised applications (DApps). Among these, the Ethereum blockchain stands out as the leading platform for smart contracts and DApps [89]. The Ethereum blockchain has popularised the term 'smart contracts' since its launch in late 2015 [206]. Many industries and sectors worldwide are exploring the potential benefits of smart contracts [308, 164, 334], which may exist in various use cases beyond cryptocurrencies such as finance, management, the Internet of Things, energy and healthcare.

Despite their potential, smart contracts face significant hurdles due to their inherent

complexity in development and user interaction [141, 155]. The shift to decentralised applications introduces challenges distinct from those experienced with centralised systems and often overwhelms users with technical intricacies. Blockchain's core features [136, 137, 333]—immutability, decentralisation and transparency—amplify this complexity, making smart contracts difficult for average users to understand and use confidently [207, 13]. Additionally, most research has focused on technical aspects, specially security [1, 326, 267, 269, 183], often neglecting the human-centred design necessary for fostering trust and broader adoption [335, 298].

In addition to complexity, blockchain and smart contracts introduce a new paradigm of automation and authority that is distinct from intelligent autonomous systems such as artificial intelligence (AI) [100, 234, 74, 18]. The absence of centralisation, while innovative, has led to considerable losses from malicious activities and scams [94]. In smart contract systems, immutability and enforceability mean that decisions and outcomes are irreversible, which underscores the importance of keeping humans in the loop. This decentralised environment necessitates the consideration of responsible and trustworthy characteristics such as explainability, similar to established AI practices that prioritise human-centric qualities and factors [242, 40, 37].

Therefore, this thesis aims to introduce explainability as a design concern to support the development of trustworthy systems that prioritise stakeholders' needs. The idea behind explainability is to make the operations and decisions of smart contracts transparent, accountable and understandable to stakeholders, thus increasing their confidence and trust in these systems [171, 39, 201, 72]. Explainability can help users grasp how and why certain outcomes are achieved by demystifying the complex processes underlying smart contracts. It can also encourage providers, owners and developers to build responsible systems [58]. Our objective is to assist engineers and designers by providing guidelines and approaches to incorporate explainability into the early stages of smart contract development.

To achieve this aim, this thesis first reports on a systematic literature review (SLR) that investigates and classifies common concerns and system qualities from the perspective of stakeholders during the development of and interaction with smart contracts. The review reveals that explainability is a neglected quality attribute and underscores the transparency and accountability limitations that impede the establishment of trustworthy smart contracts.

Second, this thesis systematises the current state of explainability, transparency, accountability and understandability across five system levels. This approach offers new insights into gaps, misconceptions and interrelations regarding these concepts. It contributes to developing an understanding of the role of explainability in smart contracts and provides detailed guidelines for explainability requirement analysis and design principles to aid engineers in approaching this new direction. An example case is instantiated to demonstrate their feasibility with qualitative validation.

Third, this thesis introduces a human-centred framework for identifying information requirements in smart contract explanations. Integrating principles from situation awareness (SA) and goal-directed task analysis (GDTA) [43, 86, 83, 311], it elicits requirements at three levels: perception, comprehension and projection. It also provides a detailed taxonomy of smart contract behavioural components and decision-making mechanisms, enabling tailored, contextual information elicitation. The framework evaluation involves expert consultations and a case study to exemplify its workings.

Fourth, this thesis introduces explainability purposes as an integral resource for evaluating and designing explainability in smart contract systems. It develops a novel evaluation framework adapted from the metacognitive explanation-based (MEB) theory of surprise [106] to systematically identify areas for improvement in information provision and explanation. This approach is evaluated through application to real-world lending projects and cost trade-off analysis.

Hence, this thesis informs requirements engineers and designers on how to elicit, design and evaluate the need for explainability requirements, contributing to the advancement of smart contract early development.

1.2 Problem Statement

Despite the potential of blockchain technology to revolutionise agreement execution with increased efficiency and cost reduction, the adoption of smart contract systems faces two significant challenges: complexity and trustworthiness.

Complexity: Smart contracts face significant hurdles that impede their widespread adoption, primarily due to their inherent complexity [141]. These contracts are embedded with technical details that can be daunting for the average user and, thus, make confident interaction difficult [207, 13]. The shift towards DApps adds another layer of complexity and requires users to navigate a new operational landscape. In contrast to centralised systems, smart contracts operate in an immutable and enforceable environment where decisions and outcomes cannot be reversed. If users lack understanding and explanations, this could potentially cause automation surprises due to underestimating or miscalculating the capabilities of automated systems [26, 167, 268, 124].

Although blockchain technology promises higher transparency through immutable transactions, the lack of standardised transparency complicates users' understanding of decentralised decision-making processes [13]. Transparency in smart contracts is often assumed to be guaranteed by blockchain's inherent transparency [51]. However, this assumption is misleading, as there is still opaque information beyond the technical aspects. Without standardisation, users struggle to understand what to expect from transaction records, which can seem meaningless and, thus less transparent than promised [59, 292]. Additionally, the

unexplored decision-making mechanisms add to the complexity of smart contracts and create a significant knowledge gap for researchers and users [303, 60, 234, 100].

Trustworthiness: Current research often emphasises technical aspects such as security, especially in light of incidents such as the reentrancy vulnerability and the Parity Wallet Bug, both of which resulted in substantial financial losses estimated in the hundreds of millions of dollars [1, 326, 267, 269, 183]. However, this narrow focus overlooks other critical aspects of trustworthiness, such as ethical considerations [12, 76], legal implications [99, 117, 211] and societal risks [176] including discrimination [189], inaccurate data and misuse [328, 164, 305]. This neglect of broader aspects of trustworthiness and user awareness have led to significant losses from malicious activities and scams. For example, the Federal Trade Commission reported over $1 billion in losses due to malicious activities and scams in 2021 alone [94].

While smart contracts aim to provide decentralisation, recent studies have indicated a shift towards centralisation among owners, developers and providers. This centralisation results in overcontrol, in which decision-making becomes concentrated in the hands of a few, leaving the process largely unexplored and opaque [114, 169, 5, 12, 237, 272]. Such a trend undermines decentralisation and highlights the need for greater transparency, accountability and responsible design. These challenges inhibit technical expression and successful business collaboration among parties, which depends on social mechanisms such as trust, honesty and information exchange [275]. These social mechanisms fall under the domain of explainability, which has gained massive attention in the field of Explainable Artificial Intelligence (XAI) [40, 72, 18, 2].

Explainability in AI aims to enhance responsible development and trustworthiness by making decisions transparent, accountable and understandable to diverse stakeholders [18, 40, 201]. However, these concepts are poorly defined in the domain of smart contracts and

public blockchains. Since blockchain technology is still in its infancy and new technologies often take around 20 years to reach full maturity [249, 274], introducing explainability as a design concern can significantly influence the development and evolution of these systems.

Tailoring explainability for smart contracts requires a deep investigation of their properties—immutability, enforceability and decentralisation—which present unique challenges in automation compared to AI. While AI, which utilises machine learning, excels at automating adaptive decision-making based on data patterns and probabilities [74, 126], smart contracts adhere to predetermined rules encoded in their logic and provide a deterministic framework for decision-making with irreversible outcomes. Understanding these distinctive characteristics is essential for developing explainable smart contracts suited to their unique operational paradigm.

Therefore, this thesis aims to address the complexity and trustworthiness challenges by investigating explainability requirement in the early development stages of smart contracts. It seeks to provide guidance for smart contract requirements engineers and designers in this new direction. Specifically, we aim to address the research questions presented in the next section.

1.3 Research Questions

In order to address the stated problems in Section 1.2 , this thesis examines the following research questions (RQ):

- **RQ1:** a) What are the commonly reported concerns regarding blockchain smart contracts from a human perspective, and how are these concerns currently being addressed? b) How can we identify quality attributes commonly associated with these

human-centred concerns?

- **RQ2:** a) What is the state of the art of explainability, transparency, accountability and understandability in blockchain smart contracts? b) How do these concepts align with standardised definitions? c) How can the interrelationships among these concepts guide the development of explainability in smart contract systems?

- **RQ3:** How can a human-centred design approach be utilised to identify the specific information requirements and content necessary for explaining smart contract decisions?

- **RQ4:** a) What primary explanation purposes can be integrated into the design of smart contracts? b) How can the MEB theory inform the creation of a systematic framework to assess the potential surprises in smart contracts when explanations are absent? c) What are the potential trade-offs regarding costs when integrating explanations into smart contracts?

1.4 Research Methodology

Design science research (DSR) is a rigorous and systematic methodology for creating and evaluating innovative artefacts to solve complex problems [135, 314, 19]. It employs iterative processes of design, development, demonstration and evaluation, ensuring artefacts are theoretically sound and practically effective. This methodology is instrumental in our pursuit of creating systematic human-centric analyses and solutions for explainable smart contracts. Therefore, this thesis adheres to the iterative approach outlined in the Design Science Research Methodology (DSRM) [163] to address the research questions outlined in Section 1.3.

- **Problem Identification and Motivation:** The first step in our research was to

explore human concerns and the effectiveness of current human-centred approaches. To this end, we conducted an SLR that examined vital stakeholders' challenges and the existing solutions. Our findings revealed a significant gap: the design of smart contracts often neglects human-centric and trustworthiness qualities, which are critical for addressing stakeholders' challenges regarding this emerging technology. Therefore, this thesis investigates explainability as a design concern with the aim of enhancing smart contracts' early development to better meet stakeholders' needs.

- **Define the Objectives for a Solution:** The main objective of this thesis is to prioritise explainability as a human-centric requirement to guide the development of trustworthy decentralised systems. We aim to equip software engineers and designers with guidelines and approaches to create explainable smart contract systems that empower users. We aim to raise awareness of the complexities and challenges associated with smart contracts, including the lack of transparent decision-making mechanisms which can lead to potential misuse and hinder adoption. This emphasis on explainability could inspire the development of trustworthy and responsible systems by considering early intervention in system requirements and design, enabling transparency, accountability and understandability.

- **Design and Development:** We leveraged several systematic approaches, including SLRs [162], thematic analysis [63] and knowledge systematisation [236, 92], to build a solid foundation for our methodologies. Given the novelty of explainability requirements in smart contracts, we integrated well-established theories from human factors and cognitive science. Specifically, we adopted a definition of SA [84] as a systematic approach to delineate information requirements into three levels, addressing users' needs for awareness, reasoning and projection. Additionally, we utilised the MEB theory of surprise [106], which provides empirical support for how explanations can link event outcomes with their settings, leading to surprise resolution. As a result,

this thesis establishes the foundational aspects of explainability requirements tailored specifically for smart contracts.

- **Demonstration:** To illustrate the practicality of our proposed approaches, we employed various cases and scenarios from the smart contract literature and industrial projects. These case studies include centralised decision-making functions, a flight insurance decision-making scenario and two real-world decentralised lending applications. These case studies were utilised to instantiate new requirements and design principles, demonstrate the relevance of our frameworks and assess the added value of explainability within the smart contract domain.

- **Evaluation and Reflection:** We evaluated our proposed frameworks and approaches using the evaluation techniques proposed for DSRM [302, 241, 282]. These measures included qualitative criteria such as completeness, usefulness, ease of use, orthogonality and benchmarking. Additionally, we sought expert feedback on draft frameworks through surveys and interviews. The applicability of the proposed approaches was demonstrated through case studies and actual implementations of smart contracts. We reflected on our hybrid evaluation techniques by applying the criteria established by Kitchenham et al., [166] to validate the basic, use and gain aspects of the artefacts.

1.5 Thesis Contributions

This thesis advances the blockchain smart contract systems field by introducing explainability as a design concern, prompting requirements engineers and system designers to address this concern early in the requirements and design phases of systems leveraging smart contracts. It contributes a novel human-centric framework and approaches to assist engineers in designing explainable smart contract systems that empower humans in the loop (i.e., stakeholders of

the contract). Specifically, this thesis makes the following contributions:

1. **A systematic review of existing literature on human concerns and considerations in blockchain smart contracts.** We conducted an SLR to identify common concerns from the stakeholders' perspective, revealing recurring themes related to development and interaction. We classify these themes into programming language complexity, legality, readability, ethics and social implications, usability, trust, governance and costs, affecting technical developers, non-technical experts and end-users. Additionally, we categorise the scattered solutions and interventions developed to address these concerns into new programming languages, code-comment methods, visualisation tools, natural language solutions, detection and assessment tools and development support methods. The SLR highlighted that current solutions mainly focus on creating new programming languages and external tools to address human concerns and overlook the design aspects of existing systems. To address this, we mapped frequently reported human-centric concerns to system quality attributes to provide a contextual understanding of the deficiencies in these systems. We observed that explainability and interpretability are overlooked qualities in smart contracts, while transparency and accountability receive limited exploration. Therefore, we advocate for incorporating explainability requirements in smart contracts to enhance human-centric design. As a result, we outline future research directions for smart contract explainability.

2. **A systematisation of transparency, understandability and accountability in smart contracts unveils the role of explainability.** We systematised knowledge about transparency, understandability and accountability into five levels: output, algorithm, external data, process and application. This knowledge was acquired through literature reviews and developer interviews. We provide a structured framework to understand the current application of these concepts in smart contracts that offers a detailed understanding of their gaps, consensus and interconnectedness. In addition,

we compared our findings with standardised definitions to reveal their alignments and discrepancies. We sought to provide valuable insights for blockchain and smart contract researchers by identifying areas requiring further investigation and motivate the need for standardised definitions, as not all definition attributes are aligned. As a result, we demonstrated that explainability can enable transparency, accountability and understandability in smart contracts by bridging low-level technical details with high-level considerations. Finally, we identified core explainability requirements analysis, utilising the fundamental questions of who, what, why, when and how. We also proposed design principles tailored to smart contracts, instantiated with an example case. Through this approach, we offer a detailed guidance for engineers to elicit and design explainability for smart contracts.

3. **A human-centric framework to determine information requirements for explainable smart contracts (XSC).** We developed a structured, human-centred framework for defining information requirements for the design of XSC systems. We addressed the lack of established methods for generating explanations within smart contract systems by adopting the SA definition with GDTA. We proposed three levels of XSC explanations: perception, comprehension and projection. These levels were tailored to determine explanatory information by considering smart contracts' behavioural properties and decision-making structures. We classified the behavioural properties into three main components: logic, data and human intervention, and categorised the decision-making mechanisms into governance structure, process location, degree of automation and behavioural pattern. These classifications can serve as a structured framework for requirements engineers, aiding them in determining informational requirements for smart contract decisions. These requirements, in turn, can inform the development of explanatory mechanisms through the three levels of XSC-tailored explanations, which are structured to align with users' needs for awareness,

reasoning and projection. We evaluated the framework through expert feedback, recognising its usefulness, ease of use and feasibility, and a case study has demonstrated its applicability.

4. **An evaluation of explanation needs in smart contracts through the lens of explainability purposes to reconcile surprises.** This study contributes to the field by proposing primary explainability purposes as integral resources in smart contract systems in two significant ways: evaluating the explanation needs and designing explainability requirements in new smart contracts. Drawing from contract law and XAI practices, we demonstrate how smart contract designers and requirements engineers can embed explanations to clarify, justify, ensure compliance and facilitate consent. Additionally, we introduce a novel assessment framework inspired by the MEB theory of surprise, which systematically identifies areas in smart contracts that require improvements in justification, clarification, compliance and consent. This framework evaluates the potential for surprises arising from insufficient or absent information (epistemic uncertainties). Utilising two real-world decentralised lending applications, we applied the MEB evaluation framework and explainability purposes to evaluate, define and implement new explainability requirements. We also provide a trade-off analysis of the costs associated with integrating explanations, offering insight into economic implications such as deployment and execution costs. These contributions establish a theoretical and practical foundation for enhancing explainability in smart contracts, which ultimately benefits designers and engineers in creating human-centered smart contract systems.

1.6 Thesis Roadmap

This section provides an overview of the thesis structure, which is illustrated in Figure 1.1.

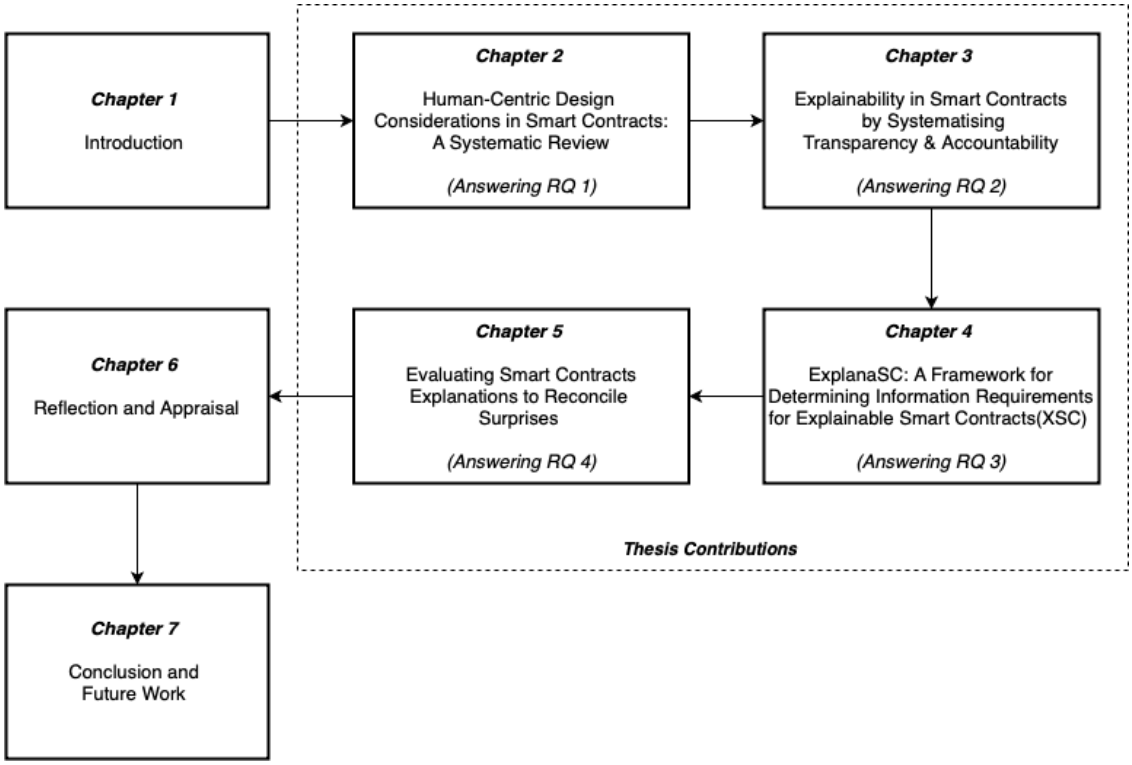


Figure 1.1: Thesis Roadmap: Chapters with Relevant Research Questions

Chapter 2 explores the current state of blockchain smart contracts by systematically reviewing the literature on human concerns and broadly categorising them into development and interaction. It provides insights into prevailing trends and solutions but reveals a neglect of the design aspects of smart contracts, including their quality attributes and constraints. This chapter identifies common and overlooked quality attributes, highlighting significant gaps in explainability and the limited attention paid to transparency and accountability. It concludes with a call for future research to address these neglected areas.

Chapter 3 systematises the existing knowledge on transparency, accountability and understandability in smart contracts into five system layers. It reveals gaps and misconceptions at each level and highlights alignments and discrepancies with standardised definitions. The chapter identifies the role of explainability in achieving transparent, accountable and understandable smart contract systems. Additionally, it provides in-depth guidelines for

explainability requirements analysis through fundamental questions (who, what, why, when and how) and proposes a holistic approach to design principles tailored to smart contracts.

Chapter 4 proposes a human-centric framework to determine information requirements for smart contract explanations (what to explain). This chapter integrates situational awareness theories to develop a framework for requirements engineers, enabling them to elicit information requirements based on three levels: perception, comprehension and projection. Additionally, it provides a detailed taxonomy of smart contract generic components that drive behaviour and a breakdown of decision-making mechanisms in these decentralised systems to facilitate tailored and contextual elicitation of information.

Chapter 5 introduces explainability purposes as integral resources for evaluating and designing smart contracts, specifically addressing the need for explanations (why to explain). It operationalises the MEB theory using surprises as a measure to evaluate the need for explanation in the context of smart contracts. Additionally, the chapter leverages explainability purposes to elicit, design and implement explanation requirements, addressing existing design gaps in clarification, justification, consent and compliance. This chapter also analyses the cost trade-offs associated with integrating explainability, revealing cost implications and suggesting mitigation techniques.

Chapter 6 evaluates the thesis by assessing the extent to which the research conducted in the previous chapters has addressed the research questions stated in Section 1.3. Additionally, it provides a thorough reflection on the evaluation methodologies applied to each contribution.

Chapter 7 concludes the thesis by summarising its main contributions and presenting an outlook for future research.

1.7 Publications Linked to This Thesis

- H. Al Ghanmi and R. Bahsoon, "ExplanaSC: A Framework for Determining Information Requirements for Explainable Blockchain Smart Contracts," in IEEE Transactions on Software Engineering, doi: 10.1109/TSE.2024.3408632. This publication is based on the work presented in Chapter 4 [4].

- H. Al Ghanmi, S. Ahmadjee and R. Bahsoon, "Evaluating Smart Contracts Explanations to Reconcile Surprises" ACM Transactions on Software Engineering and Methodology (TOSEM) - The continuous special section is on human-centric software. (under second review). This publication is based on the work presented in Chapter 5.

- H. Al Ghanmi, S. Ahmadjee and R. Bahsoon, "Explainability in Smart Contracts by Systematising Transparency, Accountability and Understandability", ACM Transactions on Software Engineering and Methodology (TOSEM) - The continuous special section is on human-centric software. (under review for publication). This publication is based on the work presented in Chapter 3.

- H. Al Ghanmi, S. Ahmadjee, H. Adeyemo and R. Bahsoon, "Human-Centric Design Considerations in Smart Contracts: A Systematic Review", ACM Computing Surveys (CSUR), (under review for publication). This publication is based on the work presented in Chapter 2.

Chapter Two

Human-Centric Design Considerations in Smart Contracts: A Systematic Review

2.1 Overview

Smart contracts are self-executing agreements encoded on blockchain platforms such as Ethereum [89]. They leverage the blockchain's distributed and immutable nature to eliminate the need for third-party intermediaries [6, 208, 164]. These contracts present unique challenges due to their immutability, automation and enforced execution. Such features demand high precision and security in the code, as deployment errors are permanent and can lead to significant financial losses [267, 269, 183]. Consequently, research on smart contracts primarily focuses on addressing these technical and security concerns [1, 308, 326].

Despite the technical focus, there is a growing recognition of the importance of human-centric approaches in smart contracts [70, 254, 112, 265, 113, 79, 214]. Stakeholders face various challenges, from development hurdles to complex user interactions, that hinder the broader adoption of this technology [141]. Therefore, there is a critical need for an in-depth exploration of the human-centric aspects of smart contracts.

This chapter aims to analyse smart contract concerns from a human-centred perspective. We conduct a systematic literature review (SLR) [162] to identify existing human concerns in smart contracts and examine the current state of solutions and interventions addressing these concerns. Additionally, we explore commonly discussed quality attributes related to human concerns and identify overlooked qualities that can aid in designing trustworthy systems.

We group the identified concerns into two main stages from the stakeholders' perspective: development and interaction. Development concerns fall into three categories: language, legality and ethical and social implications. Language emerged as the most frequently reported concern, with issues related to complexity, code readability and expressiveness affecting both technical and non-technical stakeholders involved in development. Interaction concerns include usability, governance, trust and cost, all of which impact end-users and are influenced by contract design decisions.

We categorise the current interventions addressing human concerns into analytical studies and human-centric approaches. The literature investigates these concerns through human-based surveys, empirical studies, conceptual and exploratory studies and comparative analyses. This group provides new insights into the current understanding of human concerns. The proposed human-centric approaches span various domains, including new languages and third-party tools. Therefore, we classify these solutions into new languages with their modelling approaches, code-comment methods, visualisation tools, natural language solutions, and detection and assessment tools. This mapping aims to clarify existing efforts, aiding researchers in identifying available resources and solutions.

Furthermore, we identify quality attributes derived from human concerns by mapping them to the ISO systems and software engineering standard [151], emphasising the need for greater attention in design areas such as understandability, transparency, accountability,

simplicity, learnability, usability and accessibility. We also examine trustworthiness characteristics from the National Institute of Standards and Technology (NIST) [220], highlighting explainability and interpretability as overlooked quality attributes.

As a result, we present several research directions to assist researchers in advancing human-centric smart contracts. Our main aim is to understand the current state, identify gaps and provide insights to drive further innovation in this domain. Specifically, the contributions of this chapter are as follows:

- A classification of concerns emerges from stakeholders' perspectives, delineated into two main stages: development and interaction. These perspectives reveal common themes: during development, concerns include language (complexity, code-readability and expressiveness), legality, ethical and social implications, while during interaction, the focus shifts to usability, human-readability, trust, governance and costs.
- A categorisation of scattered solutions comprises human-centred approaches and analytical studies aimed at integrating human considerations into smart contracts' development and interaction phases.
- An identification of commonly reported human-centric quality attributes and exploring new quality attributes that may have been overlooked in existing literature.
- A set of unexplored gaps and opportunities regarding human-centric design in smart contracts necessitates further investigation.

The remainder of this chapter is organised as follows: Section 2.2 provides a brief overview of blockchain smart contracts and their applications. Section 2.3 presents the systematic literature review methodology. Sections 2.4 and Section 2.5 offer the results and findings of the systematic review, addressing the research questions. Section 2.6 discusses

identified gaps, future directions and potential threats to validity. Finally, Section 2.7 compares our study with related work, followed by a summary in Section 2.8.

2.2 Background

This section provides an overview of the fundamental concepts of blockchain and smart contracts and explores their applications.

2.2.1 Overview of Blockchain

Blockchain technology is fundamentally a decentralised ledger that records transactions across multiple computers, ensuring that once transactions are registered, they cannot be altered [164, 284]. The term "blockchain" refers to a series of digital blocks connected by reference hashes. Each block references the previous block, creating a chain of blocks [332]. A transaction records the exchange of value or assets, and once accepted and added to a block, it cannot be updated or modified. Thus, blockchain technology is immutable to ensure the integrity of transactions [6]. The literature highlights three key characteristics of blockchain [136, 137, 333]: immutability, which ensures data cannot be changed or deleted; decentralisation, which operates on a distributed network making it resistant to tampering; and transparency, allowing all participants to view and verify transactions.

Blockchain technology gained prominence with the emergence of Bitcoin in 2009 [93]. Various blockchain platforms have been developed to support a wide range of use cases beyond cryptocurrency. Among these, Ethereum [89] stands out as the most popular platform for developing and executing smart contracts. Unlike Bitcoin, which primarily serves as a digital currency, Ethereum's architecture facilitates the deployment of a wide range of

DApps. Ethereum supports several high-level programming languages for smart contract development, including Solidity [280] and Vyper [291]. However, Solidity remains the most widely used and supported language due to its comprehensive tooling, extensive documentation and large developer community [230]. As a result, it is the preferred choice for building and deploying applications across different domains such as finance [308].

2.2.2 Overview of Smart Contracts

Smart contracts are agreements with predefined terms written directly into their code. These contracts automate the execution and enforcement of agreements, eliminating the need for intermediaries [208, 206]. Smart contracts have accelerated blockchain adoption by enabling computational tasks similar to object-oriented programming languages. While digital agreements and technology-driven rule enforcement are not new, smart contracts stand out due to their unique combination of asset control, automation, immutability, and enforceability.

A smart contract has three main elements: storage, balance and program code. The contract's state consists of its storage and balance. Once deployed on the blockchain, it is assigned a unique address, similar to a user account, which can receive and hold cryptocurrency. Therefore, the balance refers to the amount of cryptocurrency or digital tokens held by the contract at any given time [6]. The blockchain stores each contract's state and updates it every time it is invoked. Network users can invoke a smart contract by sending transactions to its address. Miners are responsible for creating smart contracts and recording every received transaction into a block through consensus [6, 238]. Based on the received transactions, smart contracts can read from or write to their storage and send or receive messages or funds, which alters their state in the blockchain network.

Applications of Smart Contracts

As discussed in the literature, smart contracts are widely used in various DApps, with commonly mentioned terms such as DAO (Decentralised Autonomous Organisation) and NFTs (Non-Fungible Tokens). For clarity, these terms will be consistently used throughout this chapter to refer to specific applications of smart contracts. DAOs are entities managed by smart contracts, where decision-making processes and governance are automated through code [309]. NFTs are distinct digital assets that cannot be divided or duplicated, making them ideal for representing ownership of digital art, collectables and other unique items [252]. In addition, smart contracts in cryptocurrencies facilitate automated trading strategies, manage processes within DeFi (Decentralised Finance) protocols, enable asset transfers and allow for assets tokenization [271].

2.3 Research Methodology

This study employed the SLR methodology to investigate our specific research area thoroughly. Following the well-established methodology by Kitchenham and Charters [162], we meticulously proceeded through several key stages: (i) identifying research questions, (ii) devising a search strategy, (iii) establishing inclusion and exclusion criteria, (iv) conducting a rigorous study selection process, (v) assessing the quality of selected studies, and (vi) extracting and analysing relevant data. This methodology enabled us to comprehensively assess the current state of the art and identify potential research gaps. Figure 2.1 illustrates our SLR research process.

Figure 2.1: Systematic Literature Review (SLR) Research Process

2.3.1 Research Questions

We aim to explore the common human concerns and considerations surrounding smart contracts. The following Research Questions (RQs) are formulated to guide this investigation:

- **RQ1:** What are the most commonly reported concerns regarding blockchain smart contracts from a human perspective? And how are these concerns currently being addressed?
- **RQ2:** How can we identify quality attributes commonly associated with these human-centred concerns?

2.3.2 Search Strategy

To gather relevant studies, we performed our literature search across several renowned databases and search engines, including IEEEXplore, ACM Digital Library, Web of Science and Scopus. These databases cover publications from publishers such as Elsevier and

Springer, ensuring a diverse and extensive collection of relevant literature. The search string keywords were developed based on insights from a scoping review, which involved evaluating potential keywords. During this review, we realised that focusing solely on terms related to human-centred was too restrictive. For example, when we searched for ("human-centric" AND "smart contracts") in IEEE Explore, we found fewer than ten results. To address this limitation, we expanded our search criteria to include additional relevant terms, particularly those related to legal, ethical and social concerns.

Furthermore, the literature contains various terms or combinations related to quality attributes (e.g., quality concern, non-functional requirements, concern, QAs, or NFR). Many studies also discuss specific quality attributes such as privacy, performance or transparency without explicitly using "quality" in the text. We experimented with different search strategies related to quality attributes but found that they retrieved studies unrelated to our research questions. Therefore, to avoid missing any relevant studies, we used the search string in a generic form and removed any terms related to concerns and qualities similar to [179]. As a result, the tailored search strategy specifically focused on blockchain smart contracts with consideration for humans in the loop as follows:

- *(Blockchain) AND ("smart contracts" or "smart contract") AND (human OR human-centred OR human-centric OR socio-technical OR ethical OR ethics OR legal OR compliance OR regulation OR social) AND (design OR requirement OR patterns OR specification)*

2.3.3 Study Selection

To ensure the relevance of the papers retrieved from the search, a screening process was developed. This process involved the identification and application of inclusion and exclusion

criteria to ensure objectivity in the outcomes, as follows:

Inclusion Criteria (I)

- **I1:** Papers are peer-reviewed journals, conferences, book chapters or workshops.
- **I2:** Papers include information about blockchain smart contracts, focusing on concerns arising from human perspectives and considerations.
- **I3:** Papers explicitly relate to the topics of research questions, focusing on human concerns and proposing or discussing solutions, methods, frameworks or approaches for addressing the identified concerns and qualities.

Exclusion Criteria (E)

- **E1:** Papers discuss blockchain with technologies such as IoT, cloud environments, mobile platforms and disciplines other than computer science.
- **E2:** Papers propose the utilisation of blockchain and smart contracts as solutions for specific use cases and applications.
- **E3:** Papers address concerns related to blockchain properties and have limited discussions on smart contracts.
- **E4:** Papers with restricted access or unavailable full text.
- **E5:** Papers written in languages other than English.

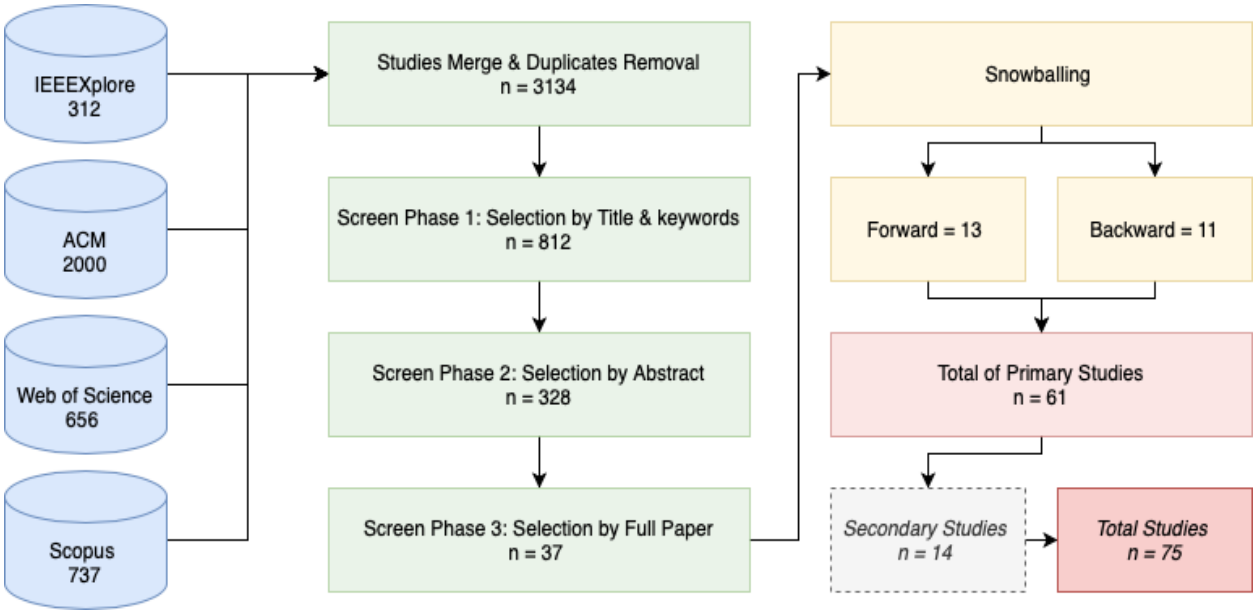


Figure 2.2: An Overview of Studies Selection Process

Screening Phases

The final selection of papers was determined through multiple rounds of filtering. Figure 2.2 illustrates the search and selection processes.

Initial Search: A total of 3705 relevant studies were retrieved from IEEE Explore, ACM Digital Library, Web of Science and Scopus using the search string defined in Section 2.3.2. The initial search on the ACM Digital Library yielded over 2,000 results. However, due to limitations in the library's retrieval process, only the first 2,000 studies were successfully retrieved.

Duplicates removal: After eliminating duplicate entries, the initial pool of studies was refined to 3134 unique results.

First Round: Studies were evaluated based on their titles and keywords to exclude irrelevant studies. This screening phase involved the application of criteria I1, E1 and E2, which led to the selection of 812 studies meeting our initial selection criteria. However, certain studies posed challenges in determining their relevance based solely on titles and keywords, prompting us to advance them to the following screening phase for further assessment.

Second Round: The authors independently reviewed the abstracts of every paper that passed from the previous phase. We applied criteria I2 and E3 to screen each paper resulting in 328 studies. Papers lacking clarity in their abstracts regarding a comprehensive discussion on smart contracts were advanced to the following screening phase.

Third Round: The authors reviewed the full text of the papers selected in the previous round to finalise the list. Criteria I3, E4 and E5 were applied during this stage, resulting in the final selection of 37 primary studies for in-depth analysis to address the research questions.

Snowballing: We implemented a snowballing strategy following Wohlin's approach [316], which involved both backward and forward snowballing. This strategy expanded our search by scrutinising the references of the initially selected papers (backward) and their citations (forward). Through multiple iterations of both forward and backward snowballing, and by applying inclusion and exclusion criteria, we identified additional studies that met our criteria. This process resulted in the inclusion of 24 more studies, bringing the total to 61 studies.

Secondary Studies: Given the emergence of human-centric aspects in smart technology, we included secondary studies for several reasons. Firstly, to explore areas that may have

been overlooked or under-discussed in primary studies. Secondly, to assess the classification and grouping of concepts within our research framework. Lastly, to comprehensively understand the evolving landscape by synthesising findings from primary and secondary sources. We selected 14 secondary studies during the screening phases that met our inclusion and exclusion criteria.

During our screening process, we used Cohen's Kappa (k) statistic to measure the level of agreement between reviewers [285], achieving a k value of 0.762, which indicates substantial agreement according to established benchmarks [285]. In cases of disagreement, reviewers discussed viewpoints to resolve conflicts. This process ensured that the selected studies met our selection criteria and maintained the integrity and objectivity of our final selection of the 75 studies.

2.3.4 Quality Assessment

To evaluate the quality and strength of evidence for each primary study, we conducted a quality assessment based on the framework established by Yang et al [324]. We adapted the commonly used criteria for quality assessment in software engineering to suit our needs. These criteria covered aspects of rationality, rigour and credibility, resulting in the selection of eight specific evaluation questions, as presented in Table 2.1.

Reviewers evaluated each primary study by answering questions using a matrix of values: (0, 0.5, or 1), representing "no," "to some extent," or "yes," respectively. The final quality score for each study was determined by summing these values. In cases of discrepancies between reviewers, discussions were initiated to reassess the studies to establish the final score.

Table 2.1: Studies Quality Assessment Criteria [324]

Characteristic	**QA Criteria**	**ID**
Rationality	Are the research aims/objectives clearly defined?	Q1
	Is the study context clearly outlined?	Q2
	Is the paper founded on research?	Q3
Rigour	Does the method adequately address the research aims?	Q4
	Is the data collection method adequately described?	Q5
	Is the data analysis sufficiently rigorous?	Q6
Credibility	Is there a clear description of findings?	Q7
	Do the researchers discuss any limitations and threats to the validity of their findings?	Q8

2.3.5 Data Extraction

Data extraction involves gathering the necessary data items to analyse the studies. Table 2.2 was specifically designed to rigorously extract the information required to address the research questions of this study. Data items D1 to D5 contain demographic information intended for quantitative analysis to demonstrate basic information about the studies. Conversely, items D6 to D8 are relevant to the research questions, requiring in-depth investigation through qualitative analysis. We utilised Nvivo [221], a software tool renowned for its text and word analysis capabilities, to identify the raw data needed to answer the research questions. Nvivo facilitates the coding process and provides easy reference for extracting information and classification, which can be shared among reviewers.

Table 2.2: SLR Data Extraction Form

ID	Item	Focus/Insight
D1	Study ID	Identification
D2	Study Title	Demographic
D3	Publication Year	Demographic
D4	Publication Type	Demographic
D5	Publisher	Demographic
D6	Concern Expressed	RQ1
D7	Proposed Solution or Discussion	RQ1
D8	Quality Attributes	RQ2

2.3.6 Data Synthesis

Our analysis of the primary studies data focuses on the qualitative information collected from the data extraction phase. To address RQ1, we started by defining the term 'concern' according to [151], where it is described as an aspect of a problem or consideration that is important or affects one or more stakeholders. We utilised thematic analysis [63] to navigate the data effectively within its context. The thematic analysis involves six distinct phases: familiarising with the data, generating initial codes, themes searching, reviewing themes, defining and naming themes and reporting. After adopting the coding techniques outlined in [260], we searched for themes once all data have been coded. During this phase, we analysed the relationships between codes and compiled relevant data within themes to identify key segments for addressing the first research question.

To address RQ2, we utilised two data synthesis processes. Firstly, we adopted the ISO/IEC/IEEE 24765 [151] standards as terminology guidelines to identify common qualities associated with concerns in smart contract development and interaction. This standard was chosen for its comprehensive collection of standardised terminologies contributed by

the International Organisation for Standardisation (ISO), the International Electrotechnical Commission (IEC), and IEEE Standards serving as a valuable reference for systems and software engineering standards. The compiled list of definitions established a foundation for identifying the qualities found in the literature. Studies may not explicitly mention quality attributes. For instance, studies referring to safety may use terms such as "safe" or "safer," and accessibility may be referred to as "accessible." We used Nvivo tool [221] to code segments and phrases related to these quality attributes. To be considered, a term must appear at least twice ($n \geq 2$) in the study, ensuring consistent recognition as a quality attribute. This approach prevents inflating the importance of less frequently mentioned terms, focusing on the commonly reported qualities. Subsequently, we mapped quality definitions with extracted terms.

The second synthesis aims to uncover quality attributes that may have been overlooked or received limited attention in the smart contracts literature. Recently, the concept of "trustworthiness" has gained attention in emerging technologies such as artificial intelligence (AI) and machine learning (ML). Trustworthy systems must meet stakeholders' expectations by demonstrating characteristics such as accountability, authenticity, availability, integrity, safety, transparency and usability [150], which align with our objectives. Therefore, adhering to the characteristics of trustworthy systems can systematically help identify overlooked quality attributes in smart contracts. While several guidelines and standards explore these characteristics [150, 149], we selected the trustworthiness characteristics from the NIST [220]. This choice was influenced by the widespread adoption of this framework in various systems literature and its recent publication. The framework provides a structured approach outlining seven main characteristics of trustworthiness, each with its quality attributes, allowing for clear identification.

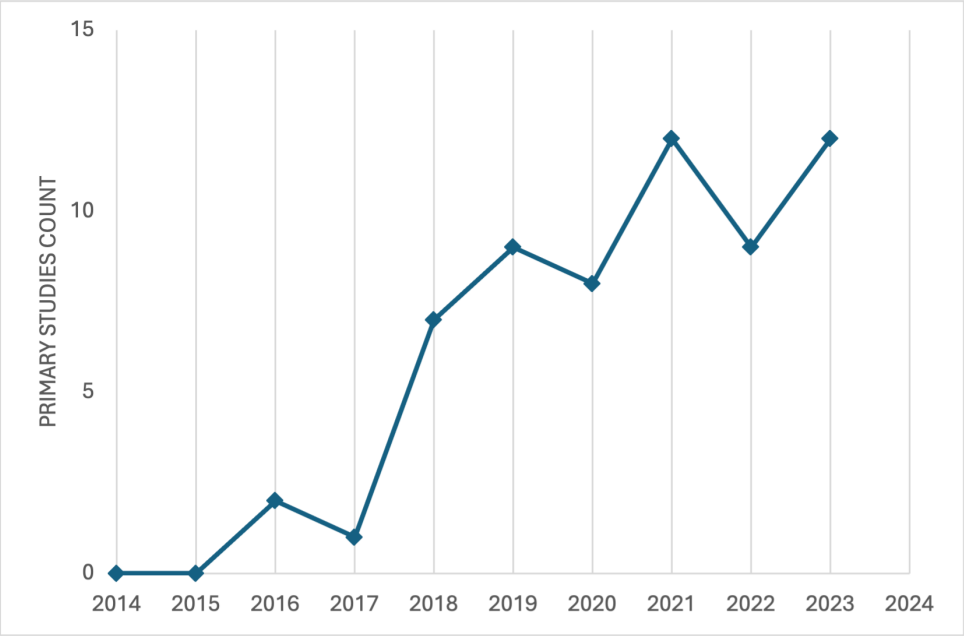


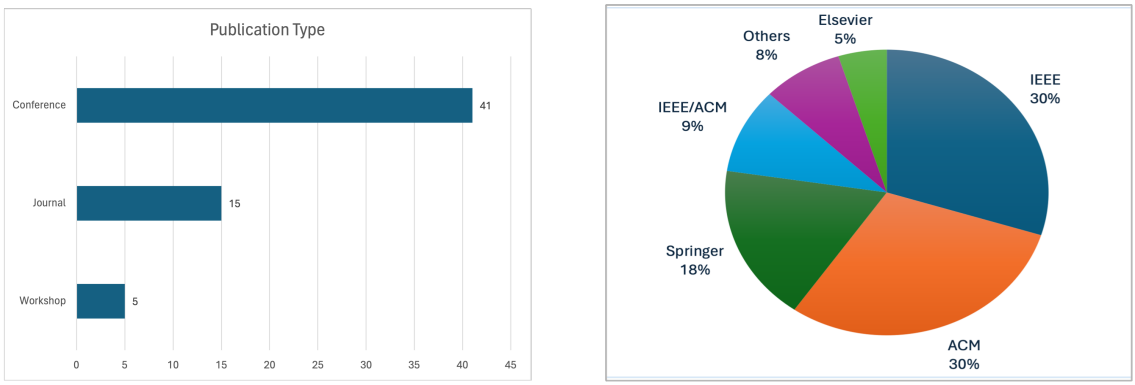
Figure 2.3: Distribution of Publication Years for Selected Primary Studies

2.4 Results

This section provides an overview of the descriptive metadata of primary studies, including demographics and quality scores.

2.4.1 Demography of Studies

The demographic details of the studies provide essential information about the selected papers, including publication year, study type and publishers. These details aid in statistically demonstrating paper information and presenting the quantitative analysis of the studies.



(a) Type

(b) Publisher

Figure 2.4: Distribution of Types and Publishers Among Selected Studies

Publication year

Figure 2.3 illustrates the distribution of publication years for the selected primary papers. Interest in smart contracts began to rise notably in 2016, following the introduction of Ethereum towards the end of 2015 [206]. The publications number gradually increased after 2016. Notably, there was a significant uptick in the number of studies in 2019, 2021 and 2023, accounting for 15%, 20% and 18% of the total studies, respectively. This increase is unsurprising, considering the growing interest in smart contracts and their capabilities, particularly in decentralised applications such as DeFi, NFTs and DAOs. The retrieval of the studies was conducted at the beginning of 2024. Since only one study from 2024 was selected, we did not include the number of studies for this year in the figure, as our collection does not adequately represent it.

Type and Publisher

Our study collection exclusively includes peer-reviewed articles. Figure 2.4 (a) offers insights into the distribution of publication types among the selected papers, highlighting conferences

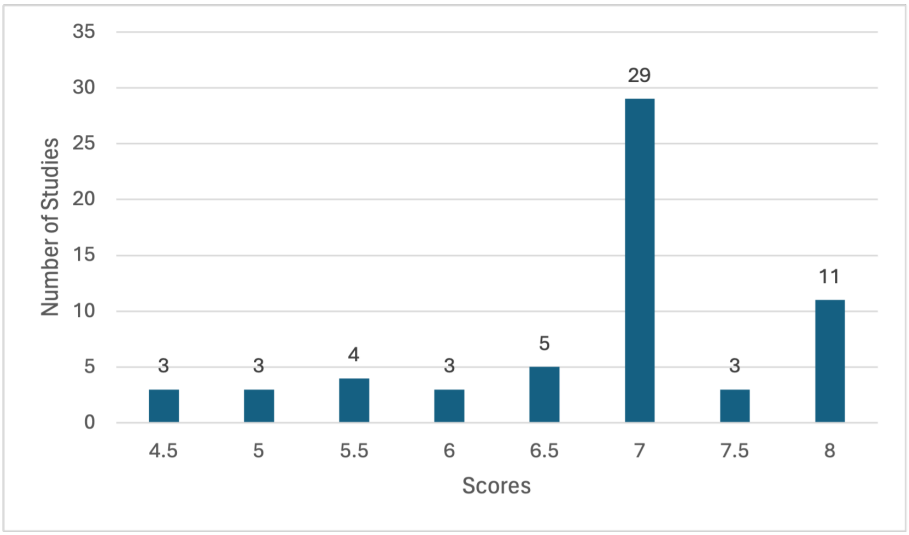


Figure 2.5: Distribution of Quality Score Among Selected Studies

(41 studies), journals (15 studies) and workshops (5 studies). Furthermore, Figure 2.4 (b) showcases the publishers of the selected studies. The majority of papers are attributed to IEEE (30%), ACM (30%) and Springer (18%). Other publishers include collaborations between IEEE/ACM (9%), Elsevier (5%) and a remaining 8% from different publishers.

2.4.2 Quality Assessment Scores

The quality scores of the primary studies were determined using the approach described in the methodology Section 2.3.4. Each study received a total score ranging from 4.5 to 8, with intervals of 0.5. These scores evaluate the quality of primary studies and are not used as grounds for excluding any study from consideration. Among the eight quality assessment criteria, Q8 received the lowest average score indicating limited or absent discussion on limitations and threats to validity for their proposed research. Most studies received scores between 7 and 8, as shown in Figure 2.5. Another criterion that affected several studies' scores is data collection (Q5), with limited discussion on collection methods. Appendix A provides each study's full results and corresponding scores.

2.5 Analysis of the Selected Studies

In this section, we provide a detailed analysis to answer the research questions. Section 2.5.1 and 2.5.2 discuss the findings related to answering RQ1, which explores the commonly reported concerns and the existing solutions. In Section 2.5.3, we address RQ2 by identifying both current and overlooked qualities for designing human-centric smart contracts. Figure 2.6 presents a summary of the findings and classifications. Detailed information on each study's concerns and solutions is provided in Appendix A.2.

2.5.1 Common Human Concerns in Smart Contract (RQ1)

The main concerns identified in the primary studies based on the iterative process employed [63] revolve around two key stages: (i) Development and (ii) Interaction, which are viewed from the perspectives of different stakeholders. The development stage impacts various stakeholders, including IT background developers, non-IT background developers and domain experts. Interacting with smart contracts mainly focuses on end-users. In the development category, concerns are classified into language, legality and social and ethical concerns. In the interaction category, concerns are classified into usability, human readability, governance, trust and cost. Table 2.3 and 2.4 present the main concerns along with their classification and corresponding studies.

Development Human Concerns

Language: The most concern reported within our primary collection revolves around language-related considerations and the coding process. Given that smart contracts embody contractual agreements, using code to represent these agreements poses challenges in interpretation. This complexity is further compounded by using imperative program-

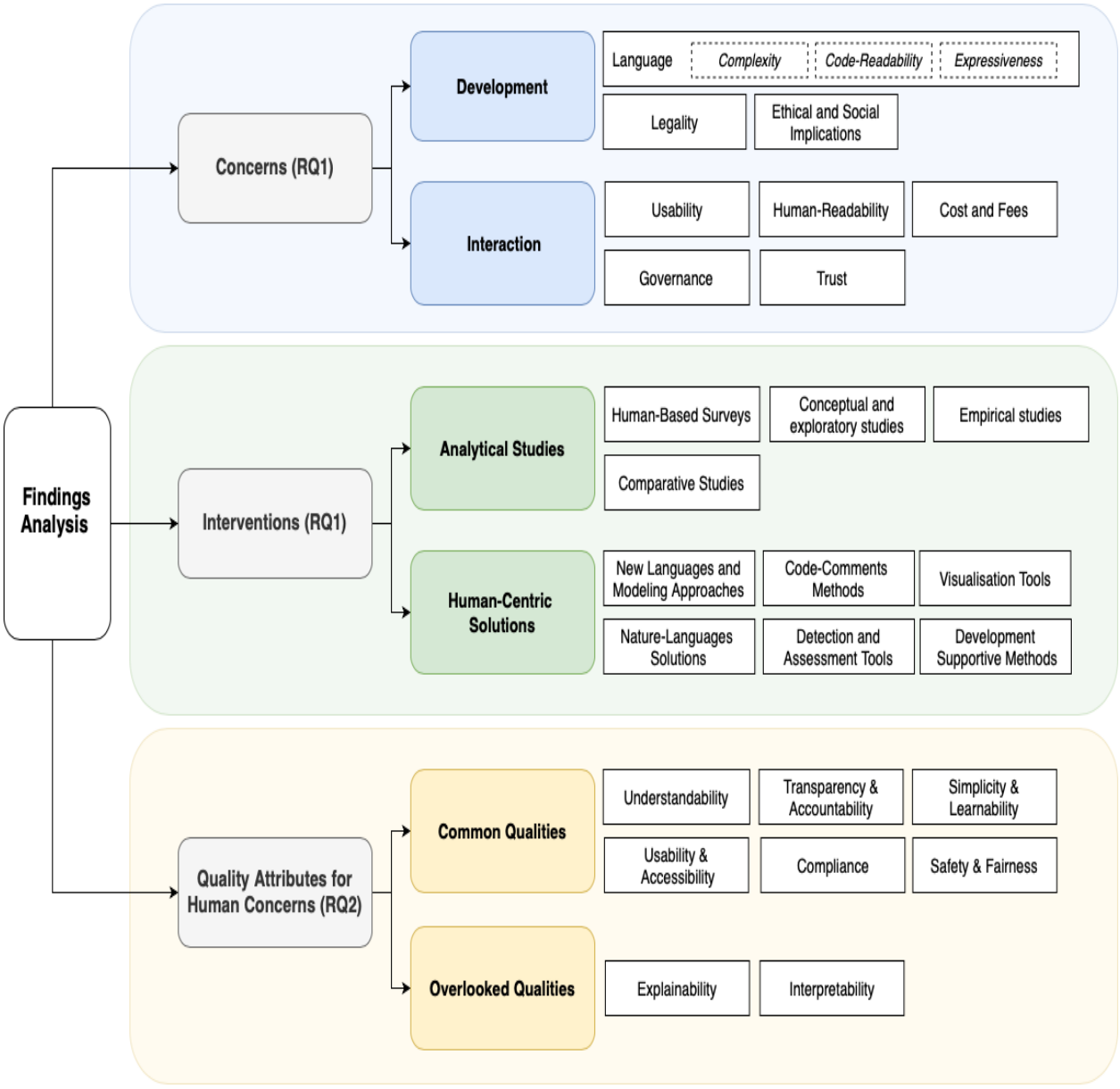


Figure 2.6: A Summary of SLR Findings and Classifications to Answer RQs

Table 2.3: Classification and Descriptions of the Main Human Concerns at the Development Stage

Concern Category	**Description of the Expressed Concerns**	**Relevant Studies**
Complexity	Concerns regarding the intricacy of a smart contract's code, especially when its structure or logic creates challenges in comprehension.	[120, 266, 296, 130, 55, 230, 245, 128, 127, 289, 62, 312, 287, 198, 109, 318, 243, 134, 293, 54]
Code-Readability	The lack of clarity and ease of understanding makes it difficult for developers to interpret, maintain, debug and reuse the code effectively.	[299, 32, 323, 329, 129]
Language Expressiveness	Challenges in ensuring that a smart contract's code can clearly and accurately convey its intended functionality, including the ability to express complex business logic and translate it into executable code.	[318, 108, 289, 48, 62, 76, 250, 5, 247, 293, 168, 121, 273, 295]
Legality	Concern over the adherence of a smart contract to relevant legal regulations and contractual obligations.	[73, 168, 200, 216, 12, 159, 295]
Ethical & Social Concerns	The lack of consideration of broader societal and ethical implications in the design and implementation of smart contracts, including issues related to fairness and accountability.	[79, 186, 303]

Table 2.4: Classification and Descriptions of the Main Human Concerns at the Interaction Stage

Concern Category	**Description of the Expressed Concerns**	**Relevant Studies**
Usability	Concerns regarding the limited usability of smart contracts, particularly in terms of interaction, navigation and comprehension.	[153, 111, 140, 227, 23, 157, 113, 214, 190, 156]
Human-Readability	Concern over the lack of accessible language and structure in smart contracts, making them difficult to understand for individuals with varying levels of technical expertise.	[250, 109, 243, 329]
Governance	Concerns related to the scope of governance and decision-making control mechanisms, including risks associated with centralisation.	[169, 152, 322, 114]
Cost & Fees	The challenges users face in comprehending and managing transaction expenses, particularly the complexities in understanding fee structures.	[153, 222, 113, 111, 96]
Trust	Concerns regarding users' confidence and trust in the reliability of a smart contract's operations, code and stakeholders.	[169, 60, 28, 113, 111, 114]

ming languages such as Solidity [280], which inherently limit the expression of legal clauses. Therefore, comprehending the code in terms of its embedded terms and policies becomes a significant hurdle. As a result, these language concerns critically impact developers and domain experts during the creation and development processes. We categorise them into key subcategories: Complexity, code-readability and expressiveness. Two secondary studies have performed SLRs to analyse the current landscape of smart contract languages, focusing on identifying the various languages and their characteristics [77, 300].

- **Complexity** refers to the intricacy and difficulty associated with the programming language and the resulting code. It considers factors such as syntax intricacies, limited features, lack of standardised development practices and the learning curve required to master the language [120, 266]. A complex language can impede the achievement of required goals and policies set by the system's stakeholders [296], leading to convoluted code structures and difficulties in comprehension and maintenance [130]. With limited features, developers often adapt their requirements to the rigid structures, resulting in error-prone code [55, 318, 289]. As a result, developers are faced with the daunting task of manually implementing and enforcing terms within the constraints of the existing languages, leading to increased complexity and potential risks [230]. In the primary studies, three perspectives are discussed regarding concerns related to language complexity: IT background developers, non-IT background developers, and collaborative developments.

  - *IT-background developers:* The complexity of generating or creating code can greatly affect how easily developers can work with and deploy smart contracts. The main concern revolves around the usability and complexity of programming languages and tools, which often demand significant time and expertise to navigate effectively [62, 54, 130, 55, 245, 128, 127, 289, 266, 120]. Balancing security and usability in smart contract development is challenging. While Solidity is the most

usable language for new developers, it is more prone to vulnerabilities. In contrast, Liquidity and Pact offer greater security but are less usable within the developer community [230]. In addition, complexity is further compounded when developers design smart contracts for different blockchain platforms [128, 127].

  - *Non-IT background developers:* The main concern is the complexity involved in smart contract development, which primarily caters to developers. These systems lack development usability for developers without programming experience, making it challenging for them to create smart contracts. Professionals in business areas and developers unfamiliar with smart contract technology face difficulties understanding and working with these contracts [62, 312, 287, 198, 120, 109, 318]. There is a recognised need for natural language to simplify the creation process, enabling non-programmers to participate in smart contract development.
  - *Collaborative Development:* The complexity of collaborative development with varying expertise and perspectives can lead to communication breakdowns and conflicts in drafting contract terms and conditions [243, 134, 293]. Semantic consistency poses a challenge, as contracting parties may interpret terms differently based on their contexts. Achieving common consent is difficult due to variations in natural language grammar, particularly in translations between languages, which may alter the original meanings of contract clauses. Furthermore, the need for multilingual understanding arises in the globalised business landscape, where companies operate in multiple markets and require contract interpretations across different languages. These challenges underscore the intricate nature of collaboratively developing smart contracts.

- **Code-Readability:** The literature highlights two key concerns of readability: code-readability and human-readability. Code-readability refers to how well developers can understand the code for maintenance and reuse, while human-readability concerns how

easily stakeholders can comprehend the code. In decentralised applications, smart contract code is visible to all users, enabling them to read, assess and understand it for interaction. Therefore, human-readability reflects an interaction aspect discussed in the next subsection. The challenge of ensuring code readability in smart contract development is particularly pronounced due to the extensive reuse of code [299, 32]. A Study indicates that 10% of security vulnerabilities are related to code reuse that lacks proper comments [323]. This challenge is compounded by the need to optimise code to reduce gas consumption. In this context, gas refers to the unit of measurement for computational work required to execute operations on the blockchain and determines the fees paid in cryptocurrency for deploying and running contracts. Since gas fees are directly tied to the complexity and length of the code, developers prioritise optimising smart contracts to minimise these costs. However, this often comes at the expense of readability. Enhancing readability can increase gas consumption, resulting in higher deployment and execution costs. On the other hand, poor readability can lead to errors, making it difficult for developers to maintain and reuse code effectively, thus increasing the risk of vulnerabilities [299, 32]. Furthermore, due to code reuse practices, comments often contain inconsistencies that can mislead developers and users, potentially introducing vulnerabilities to contracts [129]. The lack of effective comments in most smart contracts code is concerning in terms of code-readability [323, 329].

- **Expressiveness:** It highlights the language's ability to enable developers to articulate complex concepts or solutions concisely through its features and structures. The expressiveness concerns inherent in existing languages present significant challenges, particularly in accurately converting natural language contracts into machine-readable code while preserving validity and semantic fidelity [318, 108, 289, 48, 121, 250, 62]. This challenge is especially pronounced in the development of legal smart contracts, where a thorough understanding of legal contract terms is essential for their precise

translation into software requirements [76, 250, 5, 247, 293, 168, 121, 289]. The translation process involves mapping the content and structure of legal contracts to a formal representation understandable and executable by smart contract systems, demanding collaboration between engineers and domain experts to ensure that contract terms are expressed accurately into executable code. However, increasing the expressiveness of the code can compromise the safety of smart contracts [273]. Additionally, the study [295] argued that the focus should not be on the translation and expressiveness of programming languages but rather on the design of smart contracts.

Legality of Smart Contracts: One of the primary concerns highlighted in developing smart contracts is the absence of explicit legal regulations across many jurisdictions. This lack of clarity raises uncertainties regarding their legal validity and enforceability [73, 168, 200, 216, 12, 159]. Smart contracts, functioning as legally binding agreements, often encounter challenges in harmonising with established legal frameworks such as international law, securities law and general data protection regulations (GDPR) [73]. For example, smart contracts, inheriting immutability from blockchain technology, encounter challenges in adhering to the GDPR's "right to be forgotten" principle. This discrepancy leads to enforceability and legality issues, with potential implications for consumer protection and transactional clarity. Moreover, traditional contract law tools, designed for conventional settings, may not seamlessly adapt to the technological complexities and immutability of smart contracts such as termination, rescission, modification and reformation [200, 159]. Ongoing debates on smart contracts vary widely, with some disputing their classification as contracts altogether or perceiving them as a disruptive force in traditional contract law [12, 295].

Ethical and Social Implications: Few studies have shed light on ethical and social concerns during the development of blockchain and smart contracts technology. These concerns

include encoded biases, transparency, accountability in governance and decision mechanisms and the risks associated with commodifying social interactions and values [79, 186, 303]. The lack of public awareness and transparency in decision-making processes within smart contract systems leads to biases and may empower specific parties while excluding others. The opacity surrounding decision determinations may obscure the nuanced social interactions shaping contractual relationships.

Interaction Human Concerns

Usability: Users often face challenges when interacting with smart contracts and their transactions, particularly due to their complexity. Studies such as [153, 111, 140, 227, 23, 157, 113, 214] highlight users' deficiency in comprehending the underlying smart contract mechanisms, leading to uninformed decisions and exposure to risks and threats. These findings emphasise the critical need for clear and informative methods to aid users in understanding smart contract functionality. For example, the studies [113, 153] highlight challenges faced by first-time cryptocurrency users, particularly in terms of usability and user experience. Additionally, accessibility for users with disabilities poses significant barriers, requiring exploration into potential obstacles faced by individuals with disabilities when interacting with smart contracts, which comes with an contractual enforcement [190, 156].

Human-Readability: Blockchain and smart contracts technology introduce a novel mechanism where the code is visible to all users, allowing them to read and verify the terms embedded within the code. Therefore, the literature discusses human-readability for stakeholders who are not developers, especially in the context of legal or contractual agreements. Concerns related to human-readable contracts pertain to the comprehensibility of coding information and data for individuals with varying levels of expertise. Studies emphasise that the policies and terms encoded should be readable by humans, particularly if they are legal

documents [250]. It has been suggested that smart contracts should be as understandable as traditional contracts for stakeholders to grasp the written contractual agreement [109]. However, the existing programming language often lacks a straightforward mapping to natural language, impeding human understanding and reasoning [243, 329].

Governance: While it is often assumed that smart contracts operate in a decentralised manner akin to blockchain technology, this assumption does not accurately reflect the reality. Unlike the decentralised execution and approval of transactions inherent in blockchain, the governance of smart contracts often involves centralised mechanisms. In practice, the operation of smart contracts may be subject to centralised governance structures, including privileged accounts, third-party involvement and permission control mechanisms. The studies [169, 152, 322] collectively shed light on the centralised risks associated with smart contract governance, revealing a gap in academia concerning this area. The findings in [169, 114] specifically identifies the risk posed by access control mechanisms introducing privileges accounts into smart contracts. This risk materialises when privileged users access critical contract functionalities, potentially exposing vulnerabilities if their private keys are compromised. The access control can be a sensible measure for enhancing security in an open ecosystem; however, it may also undermine decentralisation. There is a need to balance authorisation and decentralisation within smart contract governance [169].

Cost and Fees: Fees emerge as a problematic area for users, often leading to incomplete or inaccurate understandings. The relationship between fees and transaction speed remains unclear, resulting in complexity and opacity for users [153, 222]. Users need to be aware of various fees, including deposit fees, exchange fees, withdrawal fees, merchant fees and network fees with their recipients. These fees are associated with different services, including wallets, exchange platforms, third-party services and miners incentives [113]. This

complexity, combined with ambiguous criteria for fee amounts and payment methods, poses challenges for users unfamiliar with blockchain technology [111] which can hinder user engagement and participation in blockchain-based applications [96]. Conversely, there is an argument for hiding the intricacies of fees, particularly the gas system in Ethereum, from end-users. Exposing users to the complexities of the gas triangle—composed of gas price, gas usage, and gas limit—can lead to confusion, inefficiencies and suboptimal user experiences [222].

Trust: There is a prevailing misunderstanding that blockchain systems function entirely without trust. While smart contract code operates automatically through the blockchain's consensus mechanism, without relying on a trusted intermediary, trust remains a significant concern. Trusting a smart contract entails placing confidence in its developers, owners and design decisions [169, 60, 28, 113]. The absence of established social contexts further complicates trust concerns surrounding smart contracts [111]. This lack of context, along with ongoing risks of centralisation and dependence on developers, increase trust concerns regarding smart contracts [114].

Secondary Studies

This subsection turns to secondary studies to validate our findings and find any overlooked insights that were not fully addressed in our primary research. The concerns expressed in secondary studies aligned closely with those discussed in our study. Studies provide a comprehensive overview of smart contracts [164, 308, 286, 254], legal aspects [78, 116, 117, 70, 239], smart contract languages [300, 77], interaction and social aspects [112, 265] and ethical considerations [288]. Furthermore, some secondary studies have expressed privacy concerns. However, we have excluded this concern from our classification because the issue primarily pertains to blockchain technology and its properties. Blockchain promotes transparency by

making transactions visible to all participants. This transparency poses challenges in safeguarding sensitive information within smart contract transactions. Therefore, while privacy is a critical human consideration in the broader context of blockchain technology, it falls outside the scope of our study, which focuses specifically on concerns related to smart contracts and their design and interaction.

2.5.2 Current Strategies and Solutions (RQ1)

This section presents current solutions and interventions aimed at addressing the discussed concerns. We exclude secondary studies from the solution discussion, as they solely review the current state of the art. Furthermore, we grouped the primary studies based on their contributions into analytical research and technical solutions for better discussion. The latter category focuses on presenting methods, tools, frameworks and approaches proposed as human-centric solutions. Figure 2.7 summarise the categorisation of the existing solutions in the primary studies.

Analytical Studies

This group involves the analysis of existing data, concepts, or phenomena within a specific domain, which captures the essence of examining and interpreting existing information to gain insights or draw conclusions [246]. These studies include human-based surveys, empirical investigations, conceptual and exploratory studies and comparative studies [118]. Given the infancy of this technology, these studies have provided insights and new concepts to be investigated further. Table 2.5 illustrates the focus areas of each analytical study in this subsection.

Table 2.5: An Overview of Analytical Studies and Their Focus Areas

Type	**Study**	**Description**
Empirical Human-Based Studies	[153]	Usability testing for DApp Application.
	[111]	Understanding of user-centred cryptocurrency threats.
	[28]	Social and legal acceptance of end users in blockchain smart contracts for energy markets.
	[303]	Evaluating design choices to support social collaborative economies.
	[60]	Understanding human trust in blockchain-based systems.
	[73]	Analysis to identify key barriers to adoption, the mismatch between legal requirements and IT capabilities.
	[214]	Using design tools and methods, along with research and public engagement, to explain new technology from an HCI perspective.
	[190]	Analysing accessibility in crypto through qualitative data from disabled individuals.
	[113]	Identifying challenges from UI to cryptocurrency-specific issues for the HCI community.
Empirical Studies	[322]	Detecting centralised security risks in existing decentralised ecosystems.
	[96]	The impact of fee prices on user activities on Ethereum.
	[222]	Supporting next-gen DApps that hide the gas triangle from users.
	[230]	Analysing programming practices for usability and security.
	[299]	Trade-off between code readability and gas Consumption.
Conceptual & Exploratory Studies	[169]	The dilemma between authorisation and centralisation.
	[79]	Bringing attention to fundamental conceptual and methodological challenges encountered by HCI researchers.
	[12]	Clarification of smart contracts in relation to the civil code.
Comparative Studies	[121]	Comparison of imperative and declarative languages from a legal and technical perspectives.
	[54]	Usability of Obsidian programming language compared to Solidity.

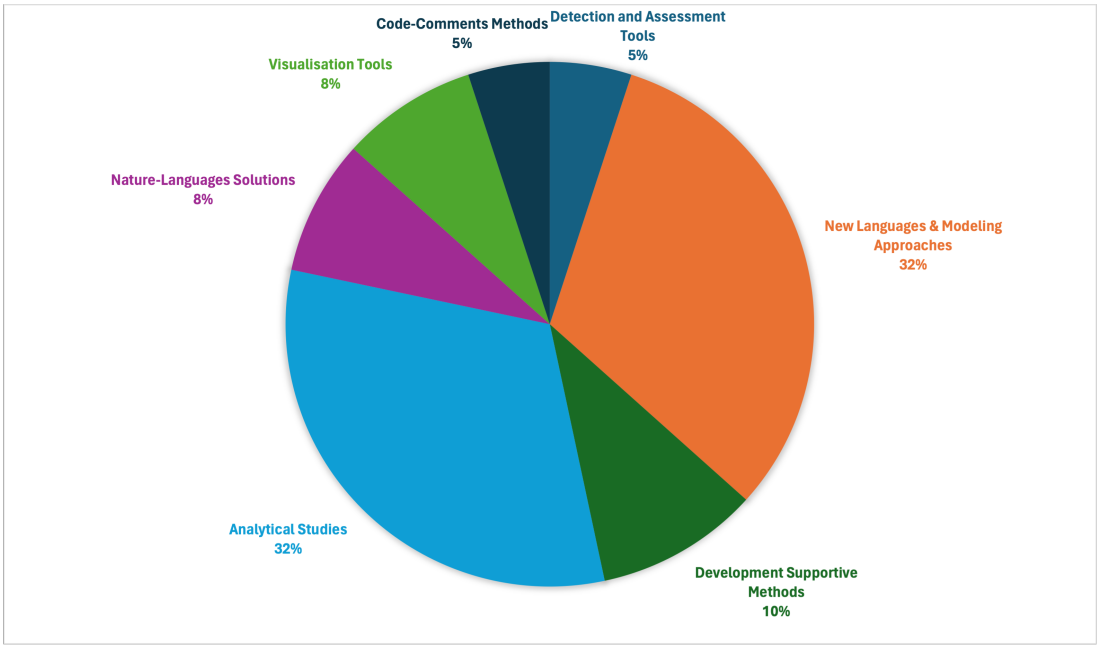


Figure 2.7: Categorisation of the Existing Solutions to Address Human Concerns

Human-Based Surveys: They have been conducted with users as subjects to test usability, accessibility and acceptance of decentralised applications [153, 28], understanding trust [60] and assess accessibility for disabled individuals [190]. Furthermore, experts have provided valuable insights into understanding user threats in cryptocurrency [111], evaluating societal implications of using blockchains [303] and identifying key barriers to adoption [73]. The studies [113, 214] have explained and highlighted the challenges for the Human-Computer Interaction (HCI) community to address.

Empirical Studies: These studies aim to understand specific phenomena. They investigate the impact of fees on user activity on Ethereum [96], examine the trade-offs between code readability and gas consumption [299] and explore methods to conceal gas transaction fees from end-users for the next generation of Dapps [222]. Another study provides comprehensive insights into identifying and detecting risks associated with centralised and privileged accounts [322]. Additionally, a separate study analyses the usability and security

aspects of programming languages used in developing smart contracts [230].

Conceptual and Exploratory: The study [79] examines the intricate landscape of blockchain in terms of trust, governance, decentralisation and the fundamental challenges of HCI. Additionally, research has explored emerging centralisation risks in the literature, particularly issues with privileged accounts and the complexities of authorisation and security [169]. Furthermore, the study [12] has provided valuable insights into the intersection of smart contracts and civil code, establishing a foundation for ensuring legal compliance in practical applications.

Comparative Studies: We encountered two comparative studies focusing on the languages used in smart contracts. The first study compares the differences and advantages between declarative and imperative languages [121]. It noted that while imperative languages are commonly used in practice, declarative languages offer better handling for legal and descriptive programming of smart contracts. In addition, the second study compares the Obsidian language (declarative) with Solidity (imperative) to provide further insights into their respective strengths and weaknesses [54].

Human-Centric Solutions

This group includes solutions and interventions such as tools, new languages, approaches, methods and frameworks. We classify and analyse these approaches based on their types, which may encompass development and interaction stages. Additionally, we specifically identify the target audiences of these solutions in our discussion.

New Programming Languages and Modelling Approaches: Addressing human considerations in smart contracts entails a focus on the languages utilised to code these contracts. This area of interest stems from the code embodying the business rules, agreements and policies, carrying obligations and enforceability upon execution. New languages, modelling approaches and their generation into executable contracts have emerged as highly reported solutions in our primary studies [23, 55, 250, 128, 127, 273, 296, 108, 318, 289, 245, 48, 134]. As a result, there is considerable interest in addressing the limitations of existing languages and how to encode policies and obligations, particularly concerning legal contracts [76, 168, 247]. A few studies have also proposed requirements to be considered when designing new languages for smart contracts [55, 5, 295, 168].

Code-Comments Methods: Several studies tackle concerns regarding the readability of smart contracts through the development of comment-generation frameworks. For instance, the CCGIR framework retrieves the most similar code from the repository and reuses its comments to generate comments for smart contracts [323]. The CCGRA approach leverages retrieval knowledge to produce high-quality comments for user-defined code [329]. Moreover, the study [129] introduces a tool for detecting inconsistencies between comments and code, aiming to minimise code misuse by users and developers, and ultimately reduce the risks of vulnerabilities.

Visualisation Tools: Some studies have proposed human-centric approaches to address issues of understandability and complexity in existing languages. Visualisation techniques have been created to serve users with non-IT backgrounds, whether for drafting smart contracts or interacting with them. For instance, studies such as [198, 287, 312] advocate for visual programming platforms to create smart contracts. Visual applications featuring user-friendly interfaces have been introduced to simplify interactions with smart contracts and

make them accessible to all, including socially vulnerable and underprivileged individuals [156]. A graph-based visualisation framework has also been developed for smart contracts, wallets and transaction data [157].

Nature-Languages Solutions: Researchers propose leveraging natural language techniques to enhance usability in creating and interacting with smart contracts. Studies such as [62, 109] aim to simplify smart contract creation using natural language input. Additionally, the SMARTDOC tool [140] assists users in understanding contract functions by generating natural language descriptions as user notices. The study [293] introduces AI-assisted frameworks using natural language processing techniques, which provide a universal representation of contracts for a common understanding of obligations. The Tx2TXT tool automatically generates security-centric textual descriptions directly from smart contract bytecode to facilitate user decision-making before executing contracts [227].

Detection and Assessment Tools: In the pursuit of enhancing human-centric aspects of smart contracts, various tools have been developed to address readability, fairness and ownership concerns. The FairCon framework automatically verifies the fairness properties of smart contracts [186]. The Ethpector is a technical solution to automatically extract privileged parties from smart contract bytecode [114]. Finally, the study [32] introduces a tool to assess code readability automatically.

Development Supportive Methods: Several approaches have been proposed to support developers and address shortcomings in smart contract development. These methods aim to streamline contract creation and provide flexibility for modification. For instance, the use of model-driven engineering approach to aid the design and creation of smart contracts [120] and a method for creating smart contracts akin to using a text editor [243]. The

Gifflar method [266] enables dynamic generation of smart contract code while a decentralised application remains operational. Additionally, The Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN) generates contract templates [130]. When addressing legal modification concerns, both [159] and [200] propose solutions—a system architecture for modification and a set of design standards, respectively. Lastly, to tackle privileged accounts and centralisation issues, the study by [152] offers a library for responsible ownership and management of ERC20 tokens.

2.5.3 Mapping Human Concerns with Quality Attributes (RQ2)

In this section, our objective is twofold. First, to identify the common qualities associated with human concerns by mapping them with the qualities outlined in the ISO/IEC/IEEE 24765 standard [151]. Second, to identify new qualities that may have been overlooked in the existing literature by adopting NIST [220] trustworthiness characteristics and qualities. The method and rationale for choosing these standards are described in Section 2.3.6.

Common Qualities

Table 2.6 presents the most commonly reported qualities and their mapped definitions from the ISO/IEC/IEEE 24765 standards [151]. We observed terms and characteristics in the context and mapped them with the most appropriate qualities defined in [151]. The second column of Table 2.6 provides representative examples of the terms used to describe concerns, which we then mapped with the terminologies in the first column. Additionally, the table includes a sample of primary studies where specific terms and characteristics of smart contracts were identified. The most commonly reported quality attributes with concerns across both phases of development and interaction include understandability, transparency and accountability, simplicity and learnability, usability and accessibility, safety and fairness.

Table 2.6: Mapping of Most Discussed Quality Attributes

Standard Mapping	**Representative Terms**	**Sample**
Understandability	"understanding SCs before executing", "deep technical understanding", "must understand the contents", "poorer understandability", "struggle to understand", "cannot understand functionality"	[23, 140, 243, 32, 293]
Transparency & Accountability	"does not speak to concrete implementations", "threat to accountability", "party's power", "trust in individuals/institutions", "destroying SCs", "control operation", "violate decentralization", "underlying intent", "parties without knowledge"	[79, 114, 227, 157, 322]
Simplicity & Learnability	"familiarity of the developers", "not easy to implement", "distinct terminologies", "steep learning curve"	[130, 54, 198, 120]
Usability & Accessibility	"how interfaces prevent adoption", "user interfaces suffer", "friendly to lawyers", "for the socially vulnerable", "accessibility violations", "multi language issue arises"	[243, 190, 113, 5, 153]
Compliance	"legally-binding DAO", "regulation and legislation uncertainty", "contracts require signatures", "conflict with SCs"	[76, 28, 295, 73, 12]
Safety	"protection", "stronger safety properties", "loss of money", "safety concern"	[273, 62, 289]
Fairness	"fair, secure, flexible", "biases will be encoded", "is unfair to certain participants"	[28, 79, 186]

The selected primary studies have also reported other quality attributes, such as flexibility, readability and efficiency.

Overlooked Qualities

Smart contracts, as a groundbreaking technology, necessitate thorough exploration as they are still in their infancy and lack standardisation and guidelines to address the challenges encountered by human in the loop. Current standards fail to adequately address the unique characteristics of immutability, automation, and enforcement inherent in smart contracts. Therefore, we extend our examination of smart contracts by exploring trustworthiness within systems that automate decision-making such as AI. Although smart contracts primarily adhere to more straightforward if/else statements for decision-making, their complexity arises during development and execution due to the immutability of blockchain, where decisions are final and irreversible, and the enforceability of outcomes in smart contracts may entail financial obligations.

Attributes of trustworthiness are deeply connected to behaviours in social and organisational settings. They are shaped by the decisions of those who develop these attributes and by interactions with individuals who offer insights and oversight to these systems [220]. Therefore, the concept of trustworthiness is akin to human considerations in designing and interacting with systems, which can help us explore potential qualities that can enhance the design and development of smart contracts.

The NIST standards [220] outline characteristics and principles for AI trustworthiness and they define quality attributes using well-known ISO standards. We leverage these characteristics to systematically identify qualities that may have been overlooked or not previously considered in the design of smart contracts. The NIST delineates seven key characteristics of trustworthiness: (1) Valid and Reliable, (2) Safe, (3) Secure and Resilient, (4)

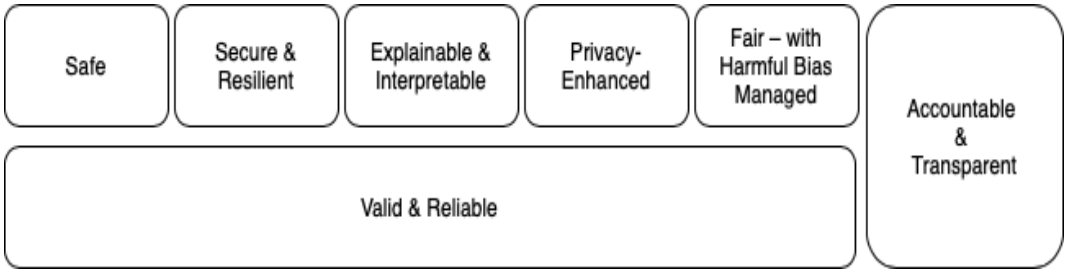


Figure 2.8: The AI Trustworthiness Characteristic by NIST [220]

Accountable and Transparent, (5) Explainable and Interpretable, (6) Privacy-Enhanced, and (7) Fair – with Harmful Bias Managed, as presented in Figure 2.8. The characteristics of "Valid and Reliable" serve as the foundation for the other trustworthiness attributes, while "Accountable and Transparent" are interconnected with all other characteristics, as illustrated in the vertical box. However, it is important to note that trade-offs typically occur. Not all characteristics apply equally in every setting; some may be more or less important depending on the situation [220].

Given the broader discussion of smart contract concerns, including technical and human aspects, principles such as validity, reliability, safety, security, privacy and fairness are already being discussed. However, characteristics that tend to be overlooked based on the seven principles are *explainability* and *interpretability*. Explainability involves representing the mechanisms underlying the operation of systems, while interpretability pertains to understanding the significance of system outputs within their intended functional contexts. Collectively, explainability and interpretability aid system operators and users in gaining deeper insights into the functionality and reliability of the system, including its outputs [220]. Additionally, smart contracts have had limited in-depth discussions on accountability and transparency, as emphasised in the human concerns Section 2.5.1. It is often assumed that smart contracts are transparent, similar to blockchain technology. However, this misconception has led users to trust risky smart contracts, resulting in financial losses [94]. The decision-making processes, policies, terms and privileged controls within smart contracts are

neither transparent nor explainable. This lack of transparency also impacts the accountability of these decisions.

Therefore, we recognise that the human-centred attributes of explainability and interpretability are frequently neglected in the domain of smart contracts. Additionally, the aspects of transparency and accountability have not been thoroughly explored in existing research. These attributes are essential for the design of human-centric smart contracts, as they address critical questions such as "how decisions are made," "who is responsible or involved in making those decisions," and "why specific actions are taken."

2.6 Discussion

This section summarises the most notable observations and highlights gaps and opportunities for human-centric smart contracts. Precisely, we carefully position our discussion on the lack of human consideration in designing smart contracts with their quality attributes. Additionally, we present the potential threats of validity and the methods employed to mitigate them.

2.6.1 Overview of Future Directions

Based on our findings, the literature on human considerations has primarily focused on developing new programming languages, code generation approaches and external tools while overlooking aspects related to the design of smart contracts. Several gaps can steer future directions as follows:

There is a gap in understanding smart contracts' current capabilities, characteristics and quality attributes. Recent research and development efforts have focused on establishing

new languages and tools for creating smart contracts, as presented in Section 2.5.2. According to [300], over a hundred new languages have been proposed for smart contracts. However, the proliferation of these new languages could complicate development efforts and present challenges for developers. Future work should focus on design aspects of smart contracts, which necessitates a thorough understanding of their properties and the establishment of new definitions and standards to support their unique nature.

Furthermore, legality and compliance concerns have driven many discussions. A significant aspect that has received attention in the literature is the conflict between smart contracts and regulations such as GDPR's "right to be forgotten" [117, 73, 239, 28]. However, the literature often overlooks Article 22 of the GDPR [131], which addresses automated algorithms. This article implies the right to receive *"meaningful information about the logic involved"*, sometimes referred to as the right to explanation [251, 145]. This article is significant in smart contracts, given that these decisions are automated with enforceable outcomes [101].

There is a noticeable absence of human-centred methodologies to identify and address smart contract requirements and design. Existing interventions primarily focus on external tools for visualisation and textual generation of current smart contracts [198, 287, 312, 156, 227, 140], without engaging in critical discussions regarding their design and requisite specifications. As smart contracts become increasingly complex, relying solely on external tools may prove insufficient and overwhelming for users. Therefore, there is an urgent need for design methodologies that prioritise human needs and comprehension when interacting with smart contracts. With this shift in perspective, we can explore new approaches to rethink the design of these systems, ensuring their decisions are understandable and trustworthy.

Limited attention is given to human qualities and attributes in smart contracts, such as understandability, transparency and accountability, all of which are integral for ensuring a

system's trustworthiness. Additionally, the gap in explainability and interpretability suggests that future research should explore approaches and frameworks to design trustworthy smart contracts with these qualities in mind. It is essential to understand how smart contracts make decisions, whether centralised or decentralised, who is responsible and what data drives the logic. This comprehensive understanding is critical to ensuring transparency, accountability, ultimately, explainability of the decisions.

There is a gap in reasoning approaches and implications within smart contract decisions. This lack of knowledge can lead to unexpected outcomes for users during the regular operation of decentralised applications. While the industry has made progress in explaining the behaviour of smart contracts, such as token swapping and auction behaviours [227], there is still a gap in providing clear explanations of the decision-making process. Future work should focus on developing methods to justify and clarify the logic behind smart contract actions, reducing bias and preventing privileged accounts from misleading users. Additionally, research should explore evaluation techniques to ensure these explanations are compelling and enhance trust in smart contract operations.

2.6.2 Gap Analysis

In our thesis investigation, we have chosen to address gaps and limitations stemming from the need for explainability to tackle some of the human concerns expressed in our findings. The key gaps that will be addressed in this thesis are as follows:

- **The oversight of human-centric system qualities in smart contracts limits understanding of the explainability role in smart contracts.** Our investigation has revealed that there has been limited focus on understanding transparency, accountability and understandability and their connection to explainability in smart

contracts. This missing knowledge pertains to the definition of transparency versus visibility, the assignment of responsibility and accountability versus traceability and the need for unbiased and non-discriminatory reasoning in decision-making. Therefore, as an initial step towards addressing this emerging need, we propose a knowledge framework to comprehend explainability in the context of smart contracts. This framework seeks to systematise knowledge of transparency, accountability and understandability across different system levels. We aim to understand the role of explainability in smart contracts and its relation to these concepts. Drawing from these findings, we provide explainability requirements analysis and design principles for smart contracts to guide designers and engineers toward explainable smart contracts. Chapter 3 provides a comprehensive exploration of unveiling smart contract explainability.

- **The lack of human-centric explanation requirements and design frameworks in smart contracts.** Our SLR findings uncovered that human-centric solutions primarily simplify development complexity and provide external tools to enhance interaction. As a result, a significant gap exists in frameworks that elicit information requirements, particularly in addressing the behavioural components and decision-making mechanisms in smart contracts. Therefore, we propose a structured, human-centred framework for defining information requirements to design eXplainable Smart Contract (XSC) systems. Combining principles from human factors, such as Situation Awareness (SA), with Goal-Directed Task Analysis (GDTA), a three-level framework has been developed to determine information and explanation requirements for XSC. Within this framework, a taxonomy has been provided to categorise existing decision mechanisms, serving as foundational elements for clarifying smart contract behaviours. As a result, the framework can assist requirements engineers to identify essential information necessary for rationalising each decision independently. The proposed framework is explained further in Chapter 4.

- **Lack of evaluation methods to assess the need for explanation in smart contracts to avoid potential surprises and epistemic uncertainties.** The field of smart contracts is rapidly evolving, yet there remains a significant gap in the development and implementation of evaluation methods to assess the necessity of explanations. This gap can result in misunderstandings or misinterpretations of smart contract behaviours and decisions, ultimately leading to automation surprises and epistemic uncertainties. To address these gaps, we introduce the concept of explainability purposes as integral resources for evaluating the explanation needs and designing explanations for smart contracts. Additionally, we develop a novel assessment framework inspired by the metacognitive explanation-based (MEB) theory of surprise to systematically evaluate the potential for surprises arising from epistemic uncertainties (lack of knowledge). These approaches can help designers and engineers evaluate explainability needs, design enhanced smart contracts with explainability and understand the cost implications of explanation. Chapter 5 provides additional details on the evaluation.

2.6.3 Threats to Validity

This section outlines the potential threats to validity identified in our study, guided by the insights provided in [10, 317].

Internal validity:

To mitigate the impact of irrelevant variables and potential biases in our study, we established a rigorous research protocol following guidelines by Kitchenham et al. [162]. Initially, we conducted a scoping review to formulate the search string by experimenting with a few databases. We then implemented a rigorous selection strategy with explicit inclusion and exclusion criteria. To enhance our coverage, we employed forward and backward snowballing

techniques [316], mitigating the risks of missed studies by automated searches. Additionally, reviewers independently conducted the paper selection process and resolved discrepancies through group discussions. To further minimise biases, we ensured all reviewers shared a common understanding and aligned the data extraction process with the research questions.

Construct validity:

A potential issue arises when the operational definitions or measurements of constructs (i.e., concepts or variables) in a study do not accurately represent the theoretical concepts intended to measure. Given the broad nature of concerns and human involvement, which could potentially introduce conflicting concepts during data extraction and synthesis, we carefully defined the terminology of concerns and perspectives related to human involvement. Specifically, we used well-established standards to define these terminologies and implemented a systematic approach to ensure consistency and clarity. Additionally, we considered secondary studies to cover insights not addressed in the primary studies and assess our identified concerns' coverage.

Conclusion validity:

A potential threat to conclusion validity is the possibility of incomplete coverage of smart contract concerns and classifications, as other classifications or themes may exist. To mitigate this risk, we employed thematic analysis [63], an iterative process that allowed us to refine our classifications as new concepts emerged. We also consulted secondary studies and standards to ensure any missing classifications or concerns were noticed. Multiple reviewers participated in this process to reach a common interpretation of the data. Nevertheless, our classification system remains adaptable and capable of evolving to accommodate new additions and changes over time.

Table 2.7: Related Work and Their Focus Areas

Study	**Focus Area**
[6]	Classifies technical challenges into six main categories: security, privacy, software engineering, application, performance and scalability.
[13]	Identifies main research streams, covering technical foundations, blockchain applications for IoT, standardisation, verification and security
[70]	Addresses automation and generation of smart contracts from a user perspective.
[112]	Cryptocurrency human challenges classified into six themes: trust, motivation, usability, user engagement, application-specific use cases and support tools.
[131]	Highlights key GDPR articles issues related to blockchain and smart contracts.
[164]	Categorises challenges into two primary categories: improvement and usage.
[286]	Discusses various technical and management challenges.
[298]	Focuses on challenges in software engineering aspects and platforms other than Ethereum.
[308]	Highlights general challenges of smart contracts across system layers.

2.7 Related Work

In this section, we review key systematic literature reviews that investigate different facets of smart contract technology. The identified SLRs fall into three main categories: technological developments, user interactions and scope-focused reviews. Our study, however, captures both aspects. Table 2.7 summarises these studies, highlighting their specific focus areas.

Technical Reviews: Alharby et al. [6] provided insights into the current research landscape, classifying studies into six categories: security, privacy, software engineering, application, performance and scalability. They have noted a lack of research on scalability, particularly in executing contracts in parallel to enhance throughput. Additionally, Wang et al. [308] presented a framework for smart contracts with six layers: infrastructures, contracts, operations, intelligence, manifestations and applications, highlighting security vulnerabilities. Khan et al. [164] categorised existing smart contract studies into two categories: smart

contract improvement and smart contract usage. The former addresses challenges such as functionality verification, performance optimisation, vulnerability mitigation and trustworthy data feeding. The latter focuses on domain-specific challenges through smart contract utilisation. Moreover, Taherdoost [286] have highlighted challenges such as the lack of solid data processing capacity, effective smart contract management and security vulnerabilities. Ante [13] conducted a bibliometric analysis of smart contracts research, identifying several main research streams: technical foundations, blockchain applications for IoT, standardisation, verification and security. It highlights emerging clusters, such as smart contracts and the law, indicating their interdisciplinary nature. Despite various findings, the study underscores the uncertainty surrounding smart contracts' potential.

These reviews primarily focus on the technical challenges associated with smart contracts. In contrast, our study takes a different approach by addressing challenges from a human perspective. We view smart contracts as social mechanisms, emphasising the importance of human understanding, trust and interaction in their design.

User-Centric Reviews: Dixit et al. [70] conducted a systematic literature review on smart contract automation models which focuses on technical features and legal significance. They highlight that existing approaches primarily cater to technical users, limiting accessibility for non-technical users and neglecting the social aspects. Our study extends this work by incorporating additional user perspectives, contributing to a broader understanding of smart contract usability. Additionally, the review by Fröhlich et al. [112] focuses on the interaction phase for a single use case, cryptocurrency, whereas our study provides a broader perspective. They identified six themes: trust, motivation, wallet usability, user engagement, application-specific use cases and support tools. Their review emphasises the importance of trust in decentralised systems and advocates for sociotechnical design perspectives. It aids in understanding blockchain interaction design and suggests future HCI research directions.

Scope-Focused Reviews: Vacca et al. [298] noted that existing literature reviews focus mainly on security and biomedical applications, rather than software engineering. Most research centers around Ethereum, with limited attention to Bitcoin and Hyperledger. This gap underscores the need to investigate software engineering issues in different blockchain platforms. The study highlights key challenges such as integrating blockchain with existing systems and evaluating associated costs and benefits. Another systematic review, conducted by Haque et al. [131], synthesised prior works on GDPR-compliant blockchains. They highlighted key GDPR articles, particularly issues with data deletion and modification. The study also explored role distribution among actors, emphasising GDPR compliance in IoT and blockchain-based industrial data contexts. However, it lacked discussion on Article 22, which pertains to providing meaningful information for automated processes and is linked to the informal right to explanation.

2.8 Summary

This chapter explored human concerns in blockchain smart contracts through a systematic literature review of 61 primary and 14 secondary studies. It identified issues in both development and interaction stages. Development concerns predominantly affect technical and non-technical stakeholders, including language complexity, legality and ethical implications. Interaction concerns, such as usability, governance, trust and cost, impact end-users.

The review highlighted important human quality attributes, including transparency, accountability and understandability, which often receive limited attention or are misunderstood. Notably, explainability and interpretability are rarely explored in smart contracts domain. The findings indicate a significant gap in addressing smart contracts' explanation requirements and qualities for designing trustworthy systems.

We argued that more attention should be given to trustworthy design elements. One direction is integrating explainability into the design of smart contracts to reshape future research in this field. As a result, this chapter presented several research directions: (i) Systematising knowledge on transparency, accountability and understandability to understand the role of explainability in the context of smart contracts. (ii) Developing a human-centric framework to determine information and explanation requirements for designing explainable smart contracts. (iii) Evaluating the need for explanations through the lens of explainability purposes to reconcile surprises and investigate cost trade-offs.

Chapter Three

Explainability in Smart Contracts by Systematising Transparency and Accountability

Chapter 2 has highlighted key quality attributes such as transparency, accountability and understandability that received limited attention in smart contracts. Our findings have revealed that explainability is rarely explored in this context. Therefore, this chapter aims to systematise existing knowledge on transparency, accountability and understandability. This structured understanding reveals gaps and areas of consensus. Building on this knowledge, we present a comparative analysis demonstrating the complementary relationship between explainability and the discussed concepts. We then provide a comprehensive analysis of explainability requirements tailored to smart contracts, derived from the foundational questions of who, what, why, when and how. This in-depth analysis serves as a foundation for the subsequent investigation of explainability requirements in the upcoming chapters.

3.1 Overview

The field of Explainable Artificial Intelligence (XAI) has emerged to address concerns in AI transparency, accountability and trust, aiming to make these systems more understandable to humans through explainability. The XAI literature often discusses explainability in terms that overlap with concepts such as interpretability, understandability, comprehensibility and transparency [18, 40, 201]. However, these terms are poorly defined in smart contracts and public blockchains' current standards, such as [146] due to their recent emergence.

To understand the role of explainability in smart contracts, it is essential to examine transparency, understandability and accountability. While these concepts are important in various systems, each with its own constraints, their application in smart contracts brings unique challenges due to the decentralisation and complexity. Blockchain technology is renowned for its transparency and accountability; however, the existing literature on smart contracts presents varied and sometimes conflicting perspectives on these concepts.

Several studies challenge the notion of transparency by highlighting that the intricate workings of blockchain and smart contracts are not understandable to different user groups [5, 214, 223, 292]. The term 'transparent' is inappropriate in the current state of smart contracts [13], as merely making the code visible is meaningless to regular users and does not guarantee its correctness, intentions, or intended functionalities [59, 207]. Similarly, accountability in smart contracts presents unique challenges despite blockchain's inherent ability to trace actions. The presence of designated privileged accounts with decision-making authority over contracts is often opaque to users, significantly undermining accountability [114, 261, 169]. Therefore, a systematic approach is needed to consolidate and clarify these viewpoints to understand the significance of explainability in smart contracts.

This chapter aims to systematise knowledge about smart contracts' transparency,

accountability and understandability by acquiring insights from literature and developers. We organise this knowledge into five levels: output, algorithm, external data, process and application. This structured approach provides a comprehensive understanding of each concept, revealing gaps, areas of consensus and interconnectedness. To extend this knowledge further, we compare the current state of each level with standardised definitions to assess their alignment and differences [151, 147, 148]. This comparison revealed that the attributes of these definitions did not fully align with the characteristics of smart contracts, underscoring the need for standardisation and tailored definitions within the blockchain domain [146]. For example, the visibility of output and algorithms did not equate to transparency, as these elements were incomprehensible to regular users, which is evident in the need for more understandability. Additionally, while accountability supports traceability, it is limited in allocating responsibility.

Therefore, we identify explainability in smart contracts as an enabler of transparency, accountability and understandability through a complementary relationship among these concepts. Although low-level aspects such as output and code are visible, they require design improvements to link them to high-level interpretations. This connection can be achieved through explainability in smart contracts, making them truly understandable, transparent and accountable.

To guide researchers and engineers in the early development of smart contracts that prioritise explainability, we outline the requirements analysis phase based on the foundational questions of who, what, why, when and how, as derived from existing explainability literature [40, 256, 283]. Additionally, we propose design principles that tailored to the unique characteristics of smart contracts which was inspired by the privacy-by-design approach [35, 144]. Specifically, the contributions of this chapter are as follows:

- A systematisation of smart contracts' transparency, understandability and account-

ability into five levels: output, algorithm, external data, process and application. This knowledge is acquired through literature reviews and developer consultations. We provide a structured framework to understand these concepts' current applications and gaps in smart contracts, offering researchers with structured knowledge requiring further investigation.

- A comparison of the current state of transparency, understandability and accountability in smart contracts with standardised definitions reveals their alignments and discrepancies across the five levels. This detailed comparison aims to identify how well smart contracts adhere to definition attributes and where improvements are needed at each level.

- A demonstration of explainability serves as a key enabler for transparency, accountability and understandability in the context of smart contracts. It facilitates a complementary relationship between these concepts, bridging low-level technical details with high-level considerations.

- An identification of explainability requirements through the elicitation of fundamental questions: who, what, why, when, and how. Additionally, we propose design principles tailored to smart contracts, instantiated with an example case. This approach provides detailed guidance for the early development of explainability.

In the remainder of this chapter, Section 3.2 provides essential background on smart contracts and explainability in XAI, serving as references for the aspects discussed throughout the chapter. Section 3.3 details our research approach, which comprises four stages. Section 3.4 presents the knowledge framework. Section 3.5 compares the current state with standardised definitions and their relationship to explainability. Section 3.6 outlines the explainability requirements and design principles for future smart contracts. Section 3.7 discusses our validation methods and threats to validity. Section 3.8 compares our work with

related work. Finally, Section 3.9 summarises this chapter.

3.2 Background

This section provides an overview of Ethereum smart contracts and explainability requirements in the field of XAI.

3.2.1 Ethereum Smart Contracts

The advent of Ethereum in late 2015 popularised the term “smart contracts", which are self-executing agreements with terms written into code [6, 284]. They automate the execution of agreements using logical flows such as if-else statements [333]. When a smart contract runs on blockchain nodes, it triggers transactions that result in status changes on the blockchain. These transactions are aggregated into blocks, and nodes must reach a consensus to add these new blocks to the chain [298]. This process makes transactions traceable, transparent and irreversible which replaces the need for a central authority, legal system, or external enforcement [308, 335, 208, 206].

Interest in smart contracts notably increased in 2017 [206], as evidenced by the studies retrieved for our analysis, which identify Ethereum as the leading platform of interest and its language, Solidity, for developing DApps. While other languages such as Vyper [291], Liquidity, and Pact [230] are available, most of the developer community and literature focus on Solidity due to its widespread adoption and integration within the Ethereum ecosystem. Consequently, this study centers on Ethereum and its predominant programming language, Solidity [280, 89]. The main building blocks of Solidity include elements similar to those found in high-level object-oriented programming language, such as functions, events, state

variables, errors and modifiers [263, 284, 95, 178, 184, 310]. Additionally, Solidity includes unique constructs designed explicitly for blockchain development, such as gas management, address data types and the capability to interact directly with the Ethereum Virtual Machine (EVM). These features enable smart contracts to perform automated tasks, manage data and enforce rules within decentralised applications.

Transactions are fundamental operations that change the state of the blockchain. In Solidity, an external account initiates a transaction to perform value transfers (sending Ether), deploy new contracts, or invoke functions. Each transaction consumes gas, a unit of computational effort required to process operations, which is paid by the transaction sender to prevent network abuse [141, 284]. On the other hand, bytecode is the low-level representation of a smart contract executed by EVM [11]. It is deployed to the blockchain when the contract is created and runs whenever its functions are called. Bytecode ensures consistent execution of smart contracts across all nodes in the Ethereum network.

Smart contracts have a variety of known DApps, as explained in Chapter 2, Section 2.2.2, such as Decentralised Autonomous Organisations (DAOs), Decentralised Finance (DeFi), cryptocurrencies, and Non-Fungible Tokens (NFTs). These references will be used throughout this study.

3.2.2 Explainability

In recent years, explainability has gained significant attention within the field of XAI. However, there is no standardised or universally accepted definition of explainability in academic or practical contexts [2]. The terms ‘explainability’ and ‘interpretability’ are often used interchangeably in some studies, while others distinguish between them [248]. ‘Explainable’ is more frequently used in the context of AI, whereas ‘interpretable’ is commonly associated

with machine learning (ML) [98, 2]. Definitions of explainability in AI primarily focus on the various ways AI systems can communicate their processes and decisions to human users.

Explainability was first introduced by [175] to describe the behaviour of AI-controlled entities. Initially, it was defined as a process or methodological procedure that enables users to trust and understand the outputs of machine learning algorithms [143, 98, 248]. This process involves demystifying the 'black box' nature of AI models and making them accessible. Over time, explainability has taken on a broader context, emphasising user-centric understandability. It moves beyond mere technical transparency to make AI systems comprehensible to a wider audience [278, 25, 148, 18, 39, 2]. Additionally, explainability is viewed as an interactive dialogue between AI systems and users, where the systems reveal the underlying reasons behind their decisions [209, 201, 212]. To provide further insight into this evolving trend, we quote several definitions from the XAI literature. The DARPA XAI program [126] defines explainability as the capability of AI systems to *"explain their rationale to a human user, characterise their strengths and weaknesses, and convey an understanding of how they will behave in the future."* Similarly, the ISO/IEC 22989 standard [148] defines it as the *"property of an AI system to express important factors influencing the AI system results in a way that humans can understand."* Also, the European Data Protection Supervisor (EDPS) defines explainability as delivering clear and coherent explanations for specific model predictions or decisions by providing justifications or reasons for a specific outcome that are understandable to humans [25].

Another perspective, as described by [18], asserts that the core of explainability lies in tailoring explanations to specific audiences as *"Given a certain audience, explainability refers to the details and reasons a model provides to make its functioning clear and easy to understand."* Moreover, the Fairness, Accountability and Transparency (FAT) organisation [65] defines explainability as the ability to explain both the decisions made by algorithms and the data driving those decisions to end-users in non-technical terms. FAT emphasises

that explainability helps achieve key AI principles such as fairness, accountability, and transparency.

The diversity of definitions highlights the multifaceted nature of explainability, which often focuses on initiatives, objectives and actions aimed at enhancing AI transparency, accountability, regulatory compliance, ethical decision-making and user trust [2, 16, 18]. Explainability is a fundamental aspect of responsible and trustworthy AI development and execution which ensures these technologies are beneficial and acceptable to society at large [7, 37].

3.3 Research Approach

In this chapter, our goal is to understand the role of explainability by systematising existing knowledge on transparency, accountability, and understandability in smart contracts. Therefore, we employed four main stages to achieve our goal: knowledge acquisition, knowledge systematisation, comparative analysis, and customising explainability for future smart contracts, as illustrated in Figure 3.1.

The Systematisation of Knowledge (SoK) approach, initially developed for security issues and vulnerabilities [236, 92], aims to gather and organise knowledge from existing works to provide a generalisation of knowledge. While SoK shares similarities with approaches such as SLR [162], systematic mapping [235] and taxonomies [217], their outcomes have notable differences. The SLR summarises research results to address specific questions by synthesising evidence, whereas systematic mapping provides an overview of a broad topic and maps research publications to identify trends, gaps and potential future directions. Taxonomies focus on categorising and organising concepts based on their characteristics and relationships to facilitate information retrieval. On the other hand, SoK provides a holistic overview

by systematising existing knowledge in an organised framework within a particular domain, generating new insights and offering a cohesive understanding of the field.

3.3.1 Knowledge Acquisition

This stage involved defining the sources and methods for gathering knowledge and synthesising the results. The primary knowledge sources were literature on smart contracts and consultations with developers.

Smart Contracts Literature Review

We adopted several steps from the SLR approach [165, 162] to reduce bias and ensure comprehensive coverage of existing studies. The objective was to systematically explore and synthesise the current knowledge surrounding key concepts—transparency, understandability and accountability—in the context of smart contracts. This process included defining a search strategy, selecting data sources, developing search strings, setting inclusion and exclusion criteria and creating data extraction templates.

We developed search strings to query the academic databases: IEEE Xplore, ACM Digital Library, SpringerLink and Web of Science (all databases), focusing on the terms: *(“smart contracts" AND “blockchain" AND (“Transparency" OR “Understandability" OR “Accountability"))*. To the best of our knowledge, explainability was not widely discussed in the smart contracts literature. However, we searched the literature using *(“smart contracts" AND “blockchain" AND (“Explainability" OR “Interpretability")* to gather any related insights that could be beneficial to our study.

We established clear selection criteria which include: **Relevance:** Studies must provide definition, context, practical applications or theoretical discussions of at least one of

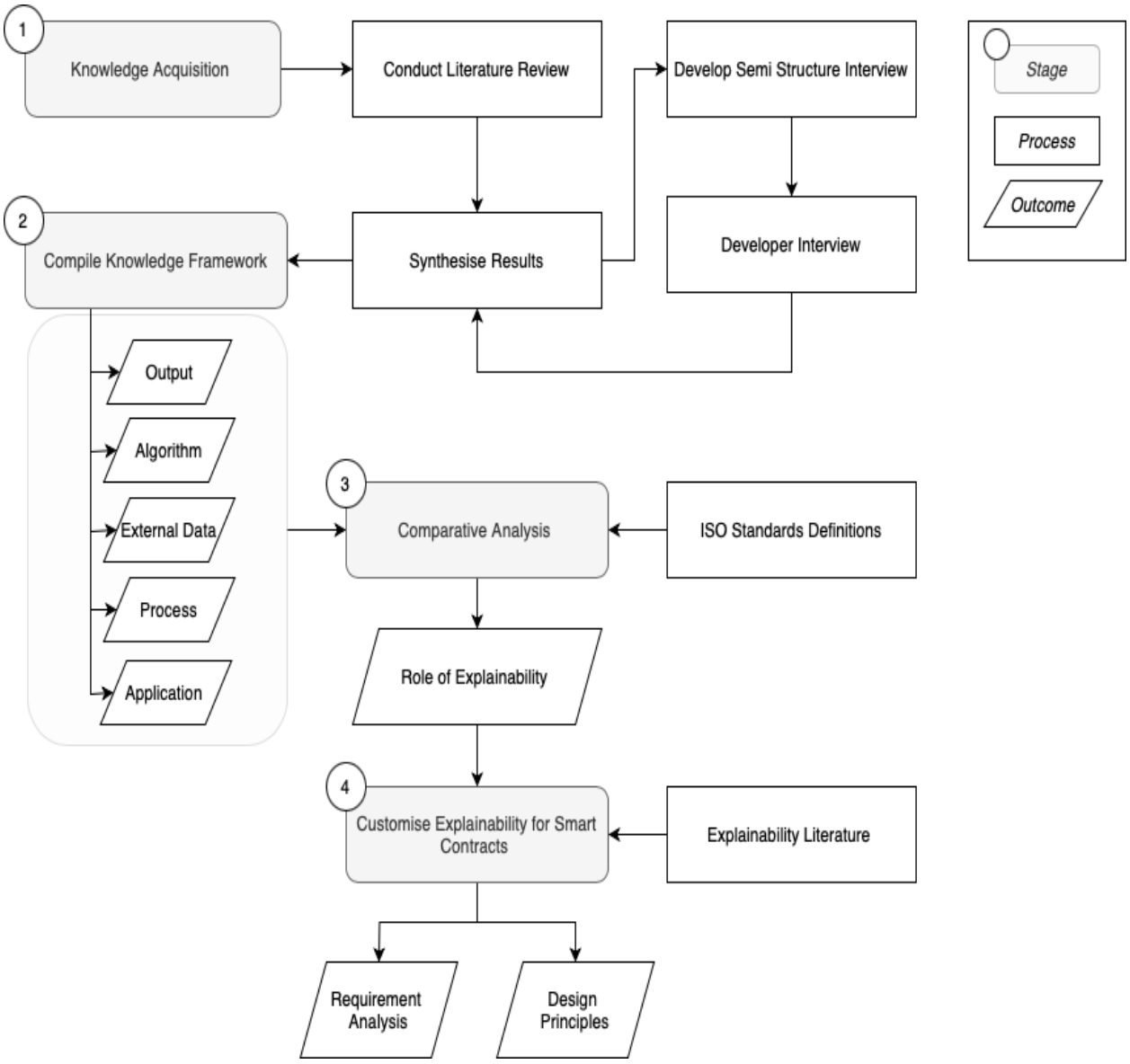


Figure 3.1: Chapter 3 Research Approach Process

the selected terms. Studies solely discussing these concepts within the context of blockchain, without smart contracts, were excluded. Also, studies that used smart contracts merely as solutions or use cases in other disciplines were excluded. **Study type:** Studies must be peer-reviewed journal articles or conference papers, excluding non-peer-reviewed studies and other types such as books and abstracts. **Discipline:** Studies must be from the fields of computer science and technology, excluding articles from other disciplines. **Language and Access:** Studies must be published in English and accessible in full text.

Our initial search retrieved a large number of studies due to the use of broad and generic search terms, particularly 'transparency,' which is a primary property of blockchain frequently mentioned in most blockchain studies. Second, these generic terms retrieved over 90% of the studies as use cases and books from various disciplines, particularly from IEEEXplore and SpringerLink. To manage this, we implemented a thorough multi-phase filtering process based on our selection criteria. First, we used libraries' automated filtering tools to include only peer-reviewed journals and conference papers, significantly reducing the number of studies. For instance, SpringerLink's initial results yielded over 6000 studies but the library filtering narrowed this down to 1332 accessible studies. As a result, the total number of studies from all libraries after this step was 5247. Next, we manually reviewed titles to exclude irrelevant studies, particularly use cases. This step further reduced the total to 342 studies given that over 90% of the retrieved studies were use cases. In the third phase, we applied all selection criteria, merged the results, and removed duplicates, resulting in 32 studies that directly addressed our criteria. To ensure thoroughness, we conducted a secondary search using snowballing techniques [316] and Google Scholar, adding 7 more studies. Our search and selection process resulted in 40 high-quality studies published between 2017 and 2024.

Finally, we developed a template for data gathering as presented in Table 3.1. This template primarily includes key information about each study, the terms and definitions

Table 3.1: Data Collection Template

Study ID	*[A unique identifier for each study]*			
Title	*[The title of the study]*			
Year	*[The publication year of the study]*			
Terms	**Concept**	**Context**	**Element**	**Supported/Issue**
[Term being addressed (e.g., Accountability - Transparency)]	*[A brief definition or explanation of the concept as used in the study]*	*[Context in which the term is used]*	*[Indicates which element the term is referring to]*	*Indicates whether the term is supported or presents an issue in the context.*

discussed, and the relationship of these discussions to smart contract elements such as code, transactions, interactions, or business objectives. It also notes whether these elements are supported in the context or present issues.

Developer Consultation

The second source of knowledge originated from semi-structured interview with experienced smart contract developers. These consultations aimed to (i) gain practical insights into the selected concepts and their real-world applications and (ii) mitigate the risk of overlooking important studies during our filtration process due to the large number of studies initially retrieved. The consultation process included: (i) developing semi-structured questions to explore transparency, accountability, and understandability from a practical, developer-focused perspective. After synthesising existing literature, as shown in Figure 3.1, we identified specific areas, recurring themes and points of consensus or contention related to these qualities. We then developed questions to capture developers' insights on these concepts, as they often present varied interpretations in the literature. Additionally, we sought developers' perspectives on the meaning of these qualities, how they relate to specific elements and layers within the system, and the current practices used to ensure these qualities are met. The questions

Table 3.2: Background Information of the Selected Developers

Develop ID	**Background**
ID01	A software developer and Chief Technology Officer with a background in cybersecurity, worked in both governmental and private sectors. He began his blockchain journey 2014 and co-founded a blockchain platforms company, establishing several projects. He has ten years of experience in blockchain and five years in Ethereum Solidity, and he teaches blockchain curriculum in a public academy.
ID02	An infrastructure developer with seven years of experience in smart contract development, starting in 2017. He founded a company specialising in blockchain infrastructure and smart contract services. His portfolio includes projects for public agencies, tokenisation, cryptocurrencies, and funding organisations. His expertise spans the entire blockchain ecosystem, from managing nodes to developing user interfaces.

are presented in Appendix B. (ii) Selecting developers based on their extensive experience in smart contract development, requiring a minimum of five years and involvement in multiple blockchain projects. To identify suitable candidates, we searched for developers through social media platforms, targeting individuals actively engaged in blockchain discussions and projects. This selection criterion ensures that the chosen developers have substantial expertise, as Ethereum was launched in late 2015, with initial developments and experiments starting in 2016 and 2017. Consequently, five years of experience indicates that these developers were part of the early wave of blockchain and DApp exploration and implementation. We reached out to developers meeting these criteria, and two developers agreed to participate in the interview, as shown in Table 3.2. The identification numbers (IDs) assigned to these developers will be used throughout this study to reference their statements.

3.3.2 Knowledge Systematisation

Initially, we synthesised data from the literature by comparing concepts, examining their context and identifying thematic consistencies or discrepancies [63]. The synthesis process was conducted by two researchers, each with five years of experience in smart contract research. Consultation transcripts were then analysed to integrate and refine the initial knowledge base with new findings to maintain the iterative approach as shown in Figure 3.1. The literature discussion is dispersed across different elements and layers of the system. For example, some discussions relate to specific low-level aspects such as logic, source code, transactions and external data feeds, while others address these concepts as high-level goals such as user interaction, the contracting process, stakeholder responsibilities, governance and legal considerations. We recorded each quality in relation to the elements referenced in its context, as shown in data collection Table 3.1. Using thematic analysis [63], we established five levels: output, algorithm, external data, process and application. These levels were derived from the available information and recurring themes in the literature on transparency, accountability and understandability, indicating where these qualities are referenced. We found that these levels correspond closely to the operational flow and layered structure of smart contract systems [308, 13]. Our literature synthesis and the corresponding levels are illustrated in Table 3.3.

- **Output-Level** focuses on the results and outcomes produced by the smart contract.
- **Algorithm-Level** focuses on the actual code, including its logic and implementation that drive the smart contract's functionality.
- **External Data Level** involves the external data fed into the smart contract, including their sources, how they are processed, and how this information is used within the contract.

- **Process-Level** addresses the development and workflows of the smart contract. It includes the steps and mechanisms that enable the smart contract to function and carry out tasks.

- **Application-Level** refers to the interfaces and interactions of end-users with smart contracts, including information about organisational objectives, the overall structure of the smart contract system and the information provided for these interactions.

Table 3.3: Literature Synthesis of Transparency, Accountability, and Understandability into Five Levels

No	*Study*	**Transparency**					**Understandability**					**Accountability**					*No*	*Study*	**Transparency**					**Understandability**					**Accountability**				
		Output	Algorithm	External Data	Process	Application	Output	Algorithm	External Data	Process	Application	Output	Algorithm	External Data	Process	Application			Output	Algorithm	External Data	Process	Application	Output	Algorithm	External Data	Process	Application	Output	Algorithm	External Data	Process	Application
1	[5]				✓										✓		21	[208]					✓					✓				✓	
2	[13]				✓												22	[207]					✓		✓							✓	
3	[42]						✓							✓			23	[214]	✓				✓					✓					
4	[50]											✓					24	[215]														✓	
5	[51]					✓	✓			✓	✓		✓		✓		25	[216]													✓		
6	[59]					✓				✓	✓	✓				✓	26	[223]				✓	✓				✓						
7	[12]	✓									✓				✓		27	[227]							✓			✓					
8	[78]	✓									✓	✓				✓	28	[231]			✓												
9	[88]														✓		29	[88]									✓						
10	[114]		✓	✓						✓					✓	✓	30	[237]	✓	✓			✓									✓	
11	[121]							✓									31	[243]							✓			✓					
12	[136]	✓											✓				32	[261]									✓						
13	[137]	✓	✓														33	[272]														✓	✓
14	[140]							✓									34	[276]							✓								
15	[141]	✓	✓														35	[292]	✓				✓				✓	✓					
16	[157]	✓					✓										36	[294]							✓		✓						
17	[161]		✓							✓						✓	37	[299]							✓								
18	[169]														✓		38	[305]										✓					
19	[178]																39	[323]							✓								
20	[182]			✓										✓																		

3.3.3 Comparative Analysis

To advance our understanding of transparency, accountability and understandability in the current landscape of smart contracts, we analysed how the current state aligns with established standards [119]. To perform this analysis: (i) We identified key attributes for each term based on established standards definitions. (ii) We classified these attributes to determine whether their meanings refer to the system level or organisational level. (iii) We identified three qualitative support measurements: Supported, Limited Support, and Not Supported, to map the current state of smart contracts with the attributes of the definitions. These steps and classifications provided a deep analysis of each level which highlight current alignments and areas needing improvement. The details of these steps and their outcomes are presented in Section 3.5 to enhance reporting and provide a clear structure for this chapter [324].

We initially considered blockchain and distributed ledger standards such as ISO 22739 [146] for this analysis. However, these standards did not fully cover the terms and definitions we were investigating. Therefore, we selected more suitable standards for our analysis. The chosen standards and the rationale behind their selection are detailed as follows: The ISO/IEC/IEEE 24765 [151] offers a comprehensive database of standardised terminologies for software engineering contributed by well-known organisations. As we are motivated by the development of responsible and trustworthy AI systems, we recognise the need to understand relevant terminologies to achieve these goals in smart contracts. Therefore, we selected ISO 26000 [147] guidance on social responsibility; ISO/IEC 22989 [148] AI concepts and terminology; and the European Data Protection Supervisor's XAI technical report [25].

3.3.4 Explainability for Smart Contracts

We performed a narrative literature review [123], focusing on the requirements elicitation phase in explainability literature. We synthesised results from high-quality studies with significant citations, which discuss various aspects of explainability requirements and their formation in different contexts. As a result, we developed a comprehensive set of explainability requirements tailored to the unique features of smart contracts. Additionally, we adopted principles from Privacy by Design [35, 144] to develop explainability design principles for smart contracts. This approach was chosen because it offers a holistic framework emphasising transparency, understandability and accountability in design principles similar to our objective. A similar approach has been proposed for IT system decisions [142] which advocates for proactive explainability rather than afterthought or add-on features. This work provides general explainability principles for software architecture, while our principles consider the unique characteristics of blockchain and smart contracts that are different from centralised systems. The results of the formation of explainability requirements and principles are illustrated in Section 3.6.

3.4 Body of Knowledge

This section presents a structured framework of knowledge on the current state of transparency, understandability and accountability in smart contracts. Each concept is discussed across the output, algorithm, external data, process and application levels. For each level, we begin by defining the concept based on literature and developers' insights, then discuss its different perspectives. Table 3.4 provides a summary of these findings. This framework can help researchers understand the current state and identify areas for further investigation. Finally, we present our exploration of explainability in the smart contracts literature.

3.4.1 Transparency

Transparency is a cornerstone of blockchain technology, yet its implementation in smart contracts reveals a complex layer of challenges that extends beyond the straightforward visibility of transactions. This section unpacks the multi-dimensional aspects of transparency in smart contracts, highlighting different levels that influence its effectiveness and perception.

Output-Level:

This level transparency refers to the visibility of transaction results and the data generated by executing smart contracts which are supported by public blockchains [141, 237, 78, 12, 136, 137] [ID01, ID02]. Although anyone can read these transactions, their complexity can limit transparency, as they may not be understandable to non-technical users and sometimes require intermediaries and tools to interpret them [214, 292, 157][ID01, ID02].

Algorithm-Level:

Transparency at the algorithm level refers to the visibility of smart contract code and the intention behind its logic. Numerous studies and developer insights emphasise transparency as the openness and visibility of smart contract code (open source), which anyone can verify through the public blockchains [237, 141, 137] [ID01, ID02]. The visibility of smart contract bytecode is inherently supported by public blockchain technology, such as Ethereum [ID01, ID02]. However, some challenge the notion of code visibility, pointing out the non-disclosure of code and lack of functions declaration [114, 161]. Simply making the code visible does not ensure transparency, as it does not guarantee correctness, intentions, or intended functionalities [207], [ID01, ID02]. Developer [ID01] confirms that code visibility on Ethereum is required for miners to execute smart contracts. Conversely, developer [ID02] states that

making the sources open for functionality is not compulsory but is a common practice in the Ethereum community to ensure trust and interaction with smart contracts. Without the source code, trust in the contract relies on decompiling bytecode, which is insufficient, as only about 40% of the bytecode can be reverted to the original source [ID01, ID02]. Therefore, decentralised communities make the code transparent for others to verify and trust.

External Data-Level:

External data level or oracle transparency refers to how the methods of data processing and external data sources are visible and communicated to users. Transparency at this level is often insufficient. Users are not clearly informed about data sources, aggregation processes and the values provided by oracles [182]. Oracles are off-chain processes (outside the blockchain) which bring transparency concerns because they are not as tamper-resistant or transparent as the blockchain itself [231, 114] [ID01].

Process-Level:

Transparency at the process level refers to how clearly the series of actions, decisions and operations involved in developing, executing and managing smart contracts are defined to different stakeholders. Using the term 'transparent' to describe smart contract processes may be inaccurate or misleading, indicating a need for ongoing examination of smart contract process transparency [13]. These processes can be complex and unclear, often not completely defined in advance. During development, the implementation processes and design decisions are also often opaque to consumers [5]. In interaction, while DApps simplify some tasks, the overall process of interacting with smart contracts—such as creating accounts, signing transactions and funding accounts—remains complex and unclear for the average user [223], [ID01]. Developers emphasise that making the code visible alone is not sufficient for the

transparency of the DApps process. The process is often complex, even for developers, whether it involves understanding the workflow of these contracts or the steps required to interact with them [ID01]. The code alone can be ambiguous for developers to understand the workflow, requiring additional explanations, comments and external documentation [ID02]. Therefore, the provision of explanations and documentation for workflows and functionalities heavily depends on the quality of the smart contract project at hand [ID01, ID02].

Application-Level:

Transparency at this level refers to organisational aspects involving the communication of key elements necessary for building stakeholder trust, including functionality, decision-making, governance, policies and conditions that mirror contract agreements. Transparency in smart contracts is often assumed to be guaranteed by the inherent transparency of the blockchain [51]. However, this assumption is inaccurate, as there is still opaque information beyond the technical aspects. For instance, in the context of NFTs, applications are immature in helping users understand their information. Even for firms and regulators, technical transparency alone is a superficial measure to claim compliance which necessitates further design investigation to determine the specific information each party needs [59]. Additionally, transparency in the governance and decision-making activities of DAOs remains an issue [237]. Developers [ID01, ID02] confirm that current practices do not put much effort into presenting contract information clearly in the user interface, resulting in numerous scams and concealed elements. This is why code visibility is needed to read and verify the code for trust [ID01, ID02].

This gap often leads to misplaced trust as regular users who cannot read code may not fully grasp the limitations and nuances of smart contract transparency [208]. Additionally, the lack of standardisation in smart contract documentation and interfaces makes it chal-

lenging for users to understand what to expect from transaction records. If these records remain exclusive and specific to blockchain outputs, the average user will find them meaningless and not as transparent as claimed [59, 292], [ID01, ID02]. The debate continues over the design of smart contract applications, specifically regarding what information should be transparent, to what level of detail and how it should be presented to ensure all stakeholders can understand them [51, 214, 223], [ID01, ID02].

3.4.2 Understandability

Understandability in smart contracts is essential for ensuring that all parties involved can effectively comprehend the contract's content, functionality, and implications. However, several challenges complicate this goal, ranging from technical complexity to issues with documentation and user interfaces. This section explores these challenges at the specified levels; however, based on our data collection, this term has not been discussed in relation to the data process level.

Output-Level:

Understandability at this level refers to the ease with which humans can comprehend output elements produced by blockchains and smart contracts such as bytecode and transactions. The technical nature of these elements makes them difficult for humans to understand [42] [ID01, ID02]. This is evident in the way transaction details are currently presented which are neither clearly articulated for novice users nor sufficiently informative for experienced users [51, 157]. There is a need to explore how to make these values more understandable and acceptable to society [51]. However, the current state of blockchain and smart contracts is designed for tech-savvy users rather than regular users [ID02].

Algorithm-Level:

Understandability at this level refers to the ease with which humans can comprehend the code of smart contracts and grasp the underlying logic. Despite code visibility, the complexity of smart contract code, especially in financial applications, makes it difficult for users to understand [227, 243, 276, 294] [ID01, ID02]. Although humans can read the source code, they often cannot grasp its meaning due to its specificity to a given programming language. This challenge indicates that readability does not automatically ensure understandability [207] [ID02]. Code comments are used to improve readability and understandability but their intended audience is unclear in literature and developer insights. Some claim comments are only for developers while others believe they are for both developers and users [323, 140] [ID01, ID02]. This misconception likely arises from the assumption that current smart contracts target only experienced users who can read and understand the code. Additionally, even developers face difficulties in understanding smart contracts due to their complexity [ID01]. Gas consumption optimisation further compounds this issue which can reduce code readability and understandability for developers [299] [ID01, ID02]. To address these issues, programming languages similar to natural language are being explored such as declarative languages [121, 243].

Process-Level:

Understandability at the process level refers to how easily users can comprehend smart contract workflows and navigate the steps required to interact with blockchain technology. Current practices still expose many low-level elements to users to promote contract visibility and execution traceability. [294, 51]. However, novice users often struggle to interact with blockchain and smart contracts due to the complexity of processes and steps such as creating wallets, managing ether accounts and using private and public keys to initiate transactions

[223, 292, 59] [ID01]. Understanding how a specific smart contract works, how to use it and its workflow can be daunting [161] [ID01, ID02]. For example, users often do not realise that smart contract applications can have centralised decision-making which grants power to specific parties known as privileged accounts [114] or employ an upgradability pattern to change rules [261, 88]. Although developer [ID01] states these elements are not intended for end-users to understand, he also stresses the importance of users checking these contracts for centralised risks or referring to auditing reports to uncover them, presenting conflicting concepts. However, developer [ID02] highlights recent advances in top DApps that consider users' understandability of the process through improved interfaces. Therefore, the learning curve associated with blockchain technology remains steep, making DApps prone to failure in terms of public adoption [59] [ID01].

Application-Level:

Understandability at the application level refers to how easily users can comprehend the information in the interfaces and documentation of smart contract applications including the contract's content, functionality and implications. However, these aspects are still overlooked where documents and interfaces fail to accurately reflect them [59, 51], [ID01, ID02]. Bridging the gap between low-level implementations and human understanding of high-level logic is required [227]. Nevertheless, nearly 90% of DApps fail to present important information, leading to misunderstandings and user misuse [ID01, ID02] such as granting ultimate approval for transactions without full comprehension [305]. These issues highlight the information shortage at this level, especially from a legal perspective. Smart contracts must be understandable to represent the parties' original intent and comply with laws. Misunderstandings about contract content can lead to disputes and potential dissolution if one party cannot comprehend the contract [12, 208]. Presenting smart contract code as the content of contractual agreements is not suitable for human understanding [243, 78]. Therefore,

this level needs more information and explanations to help users understand and ease the learning curve [214, 292].

3.4.3 Accountability

It is important to recognise that blockchain, as a standalone network, has its own accountability and decision-making rights [272]. Smart contracts and DApps also have distinct governance and accountability structures operating on the blockchain. For instance, the Ethereum blockchain has its own governance measures; however, it hosts numerous DApps owned by individuals and companies, each with varying levels of accountability for decisions within their contracts. Recognising these distinctions, we explore different perspectives on accountability, as follows:

Output-Level:

At this level, accountability refers to the traceability of actions, including the visibility of transactions and the ability to link actions to specific blockchain addresses. Public blockchains support this by allowing actions to be traced back to responsible parties through their account addresses in a verifiable way [59, 50, 78],[ID01, ID02].

Algorithm-Level:

Accountability at this level refers to the visibility and traceability of the code, its actions, and the alignment of the underlying logic with its intended purpose. This visibility opens the 'black box,' providing transparency in computational logic and transferring accountability to all members who see and interact with the code [136, 51]. However, the intricate logic and technical language can be difficult for non-technical users to comprehend, complicating

this accountability. Additionally, the ability to review or read the code is irrelevant when determining the accountability of the actions performed or agreed upon previously [207]. Developers and deployers (owners) of the contracts are accountable for the code's actions as expressed by [ID01, ID02]. It is generally assumed that developers are responsible for ensuring that the contract's meaning is clear and understandable to all parties and that all relevant information is available at the time of the contract [208] [ID02].

Data-Level:

Accountability at the data level refers to the mechanisms and developments used for oracles to feed contracts with data. A significant issue at this level is the lack of accountability for developing, selecting and processing data within oracles to support smart contract operations. This issue is compounded by a lack of transparency in their development practices which could potentially be misused, thus undermining the trust of DApp users [182, 216]. Another gap is data provenance, involving tracing data back to the original input values, which is particularly challenging in imperative languages such as Solidity [42].

Process-Level:

Accountability at this level refers to the responsibility and traceability of actions involved in the development, execution and management of smart contracts. It focuses on defining and managing roles, decisions and workflows throughout the lifecycle of the smart contract. Some studies suggest that the visibility of the code provides accountability for governance rules [51]. However, the existence of privileged accounts requires further exploration to understand their accountability and power dynamics, as it undermines their trustworthiness [114, 169, 272] [ID01]. These extra privileges must be disclosed and clearly communicated to users [ID02]. Developers highlight that top DApps have started to provide communication

about the workflow and governance decision-making process [ID01, ID02]. Additionally, a few studies highlight the absence of accountability in terms of clarity of roles and responsibilities, such as who has the authority to upgrade smart contract code [161, 261, 215, 237, 88]. Another aspect is the legal and regulatory uncertainties regarding who holds development accountability—lawyers or developers—especially for legal smart contracts [5, 12].

Application-level:

Accountability at this level in smart contracts refers to the clear communication and declaration of responsible parties and their decision rights. The literature inadequately addresses high-level accountability in smart contracts, leaving uncertainties about the declaration of who is responsible for the actions in a decentralised ecosystem. This gap highlights the complexity of enforcing accountability in environments where traditional hierarchical structures are absent [272]. Additionally, pseudonymity poses accountability challenges, as real-world identities behind blockchain addresses often remain unknown, particularly in Ethereum [161, 78]. For example, token system operators are barely accountable, often identified only by pseudonyms on social media [114]. The traceability alone does not guarantee accountability without considering other factors such as the necessary information for keeping firms accountable [59]. However, developers provide new insights into the current state of accountability at this level, noting that these practices were previously absent. Top DApps now clarify their accountability through better governance and decision-making mechanisms [ID01, ID02]. Additionally, the Ethereum Naming Service (ENS) is an emerging practice used to assign human-readable names to Ethereum accounts, making it easier to identify who owns which account [87].

3.4.4 Explainability

The term “explainability" is not frequently used in the context of smart contracts; however, the literature explores related concepts such as explanatory information and explanation mechanisms. This discussion is limited, as evidenced by the number of studies in this subsection. There are two primary user groups that need explanation: Experienced users often lack advanced transaction details such as key inputs, outputs, high fees and trends, while novice users require more comprehensive explanations to understand transactions by transforming complex data into more comprehensible forms [157]. The literature highlights concerns about the underutilisation of smart contract event logging features, resulting in a lack of explanatory information about smart contract actions and decisions [178]. Developer [ID02] emphasises event logs as the most useful construct for explanation. Additionally, using strings, despite their complexity on the EVM, is another way to provide explanations [ID01, ID02]. User notices and annotations, which help build explanations for functions, are often ignored in the Solidity community [140]. Moreover, there is a need to explain the core concepts of smart contracts in a way that supports non-experts by presenting what the program is supposed to do and what it will do, as well as explaining the terms, conditions and implications without requiring users to grasp all the implementation details [214, 207].

Table 3.4: A Summary of the Current State of Transparency, Understandability, and Accountability Across Each Level

Level	**Transperacny**	**Understandability**	**Accountability**
Output-Level	Visibility of transaction results and data generated by smart contracts on public blockchains. Complexity may limit transparency for non-technical users.	Ease of comprehending outputs like bytecode and transactions. Current presentations are not understandable to human.	Traceability of actions and visibility of transactions support linking actions to blockchain addresses.
Algorithm-Level	Visibility of smart contract code is supported by public blockchains but complexity can limit its transparency and fail to reveal the intentions or underlying logic.	Ease of comprehending smart contract code and logic. Complexity, makes it difficult for users and developers to understand.	Code visibility transfers logic accountability to those who see and interact with it but is hindered by complexity and lack of understandability in determining accountable actions.
Data-Level	Refers to how clearly data processing methods and sources are communicated. Often insufficient raise transparency concerns.	Not identified in this context.	Accountable mechanisms for supplying data is lacking for oracle development and data selection, impacting smart contract operations.
Process-level	Clarity of actions, decisions, and operations in smart contract development, execution, and management. Often complex and unclear, with opaque processes and intricate interactions.	Ease of understanding smart contract workflows and blockchain interactions. Users struggle with complex steps, compounded by misunderstanding of decisions workflow.	Responsibility of actions in development, execution and management are not well-defined.
Application-Level	Clear communication of organisational aspects to build stakeholder trust is not well-defined. There is a lack of standardised documentation and interfaces.	Ease of comprehending information in interfaces and documentation. Often lacks consideration, leading to misunderstandings.	Clear declaration of responsible parties and decision rights are lacking. Uncertainties in high-level accountability.

3.5 Comparative Analysis and Its Relation to Explainability

Our comparative analysis aims to evaluate the alignment and support of smart contract systems with standard definitions of transparency, accountability and understandability [119]. Table 3.5 provides a summary of these definitions which are derived from selected standards. The process of selecting these standards and their rationale are detailed in Section 3.3.3. We commenced this analysis by identifying key attributes for transparency, accountability and understandability. These attributes were derived by closely examining concepts' meanings and implications. We classified the attributes to determine whether their meanings refer to the system level or the organisational level. Table 3.6 illustrates this classification.

To measure the alignment of definitions attributes with the current state of transparency, accountability and understandability, we identified three qualitative measurements: Supported, Limited Support, and Not Supported.

- **Supported:** The attribute is supported in its current form.
- **Limited Support:** The attribute is partially supported but requires improvements to meet the necessary standards.
- **Not Supported:** The attribute is not supported in its current form, whether absent or ineffective, and it requires significant improvements or development to achieve support.

This assessment framework allowed us to systematically evaluate each definition attribute's alignment with smart contract capabilities and highlight areas where additional development is needed. Table 3.7 illustrates the results of this comparative analysis, showing the degree of alignment for each definition across five levels—Output, Algorithm, Data,

Table 3.5: Definitions Derived from ISO and EDPS Standards

Term	**Definition**	**Standards**
Transparency	*D1:"Openness about decisions and activities that affect society, the economy and the environment, and willingness to communicate these in a clear, accurate, timely, honest and complete manner"*	ISO 26000 [147]
	D2: "Organisation: Property of an organisation that appropriate activities and decisions are communicated to relevant stakeholders in a comprehensive, accessible and understandable manner", *D3: "System: Property of a system that appropriate information about the system is made available to relevant stakeholders."*	ISO/IEC 22989 [148]
	D4: "Refers to the ability for a specific model to be understood. In the strictest sense, a model is transparent if a person can contemplate the entire model at once."	EDPS [25]
Accountability	*D5: "Degree to which the actions of an entity can be traced uniquely to the entity"*	ISO/IEC/ IEEE 24765 [151]
	D6: "State of being answerable for decisions and activities to the organisation's governing bodies, legal authorities and, more broadly, its stakeholders"	ISO 26000 [147]
	D7: "Answerable for actions, decisions and performance", *D8:" Accountability relates to an allocated responsibility. The responsibility can be based on regulation or agreement or through assignment as part of delegation."* *D9: "Accountability involves a person or entity being accountable for something to another person or entity, through particular means and according to particular criteria."*	ISO/IEC 22989 [148]
	D10: "A transparent AI system enables accountability by allowing stakeholders to validate and audit its decision-making processes, detect biases or unfairness, and ensure that the system is operating in alignment with ethical standards and legal requirements."	EDPS [25]
Understandability	*D11: "Ease with which a system can be comprehended at both the system-organisational and detailed-statement levels, Understandability has to do with the system's coherence at a more general level than readability does."*	ISO/IEC/ IEEE 24765 [151]

Table 3.6: Key Attributes, Descriptions, and Mapping to Definitions

Concept	**Attribute**	**Level**	**Description**	**Reference**
Transparency	Visibility	System	How available and accessible information is.	D3
	Clarity	System	How clear and understandable the information within a system is presented	D3 - D4
		Organisation	How clear and understandable the communication of policies procedures, and decisions within an organisation is	D1 - D2
	Openness	Organisation	The quality of being open about decisions, actions and activities.	D1 - D2
	Proactive Communication	Organisation	Willingness to share information and dissemination of information to all relevant stakeholders	D1 - D2
Accountability	Traceability	System	The ability to trace actions, decisions and processes.	D5 - D9
	Responsibility Allocation	Organisation	Clearly defined responsibilities and roles.	D8 - D9
	Answerability	System	Provide explanations and justifications within its decisions.	D7
		Organisation	The requirement for explanations and justifications for organisational compliance and actions.	D6
	Auditing	System	The ability of the system to provide mechanisms for reviewing and verifying actions and data	D8 - D10
		Organisation	The process of reviewing and verifying organisational compliance and performance.	D8 - D10
Understandability	Ease of Understanding	Both levels	How easily stakeholders grasp the information provided.	D11
	Simplification	Both levels	Breaking down complex information into simpler, more digestible parts.	D11

Process and Application. Each cell in Table 3.7 is marked as Supported, Limited Support, or Not Supported, indicating the extent to which current smart contract systems meet definition attributes. For instance, accountability attributes such as Traceability and Auditing receive full support at the system level, reflecting strong alignment with current smart contract capabilities. However, attributes related to broader organisational concerns, such as Responsibility Allocation, often exhibit limited support, suggesting these areas need further development to fully align with standard definitions of accountability. We summarise the findings as follows:

Transparency in terms of *visibility* is supported at the output and algorithm levels but lacks *clarity* attributes. External data level *visibility* and *clarity* are not supported in their current forms. The process level lacks *openness, clarity* and *proactive communication.* The application level shows limited support for *openness, clarity* and *proactive communication* due to a lack of standardisation and scattered information based on the organisation's objectives. Therefore, smart contract systems generally only satisfy some of the attributes of transparency; they may not be transparent as they lack clarity and understandability at system and organisational levels based on definitions. While the system supports the visibility of low-level details, high-level aspects require better declaration, openness, and clarity highlighting the need for improvements.

Accountability at the system level is well-supported through output and algorithmic *traceability* and *auditing* but has limited support and lacks *answerability* in terms of providing explanations and justifications for their actions and decisions. The data level lacks all attributes of accountability. However, at the organisational level, there are opposite views and different perspectives. Auditing is an emerging concept in industrial tracks as a way to verify trustworthiness. Similarly, the top DApps' practices align with *answerability* but

it is not standardised as many projects still lack *auditability* and *answerability* at the organisational level, showing limited support in the table. *Responsibility allocation* is still not mature enough, showing only limited support in the current form of process and application levels. In general, low-level accountability is supported while answerability requires work and improvements. High-level accountability is supported to a certain degree but requires further clarity and enhancements for truly accountable DApps.

Understandability remains underdeveloped at both system and organisational levels due to the complexity of this technology, even for developers and experts. There is a lack of support for *ease of understanding* and *simplification*, indicating a significant area for future improvement to ensure stakeholders can easily grasp and digest complex information. Although some of top DApps have shown advancements in their interfaces and explanations, they still need to be standardised. As a result, the table indicates limited support at the organisational level for *ease of understanding*.

3.5.1 Explainability as a Complementary Concept

Our comparative analysis concluded that smart contracts support some aspects of transparency and accountability but lack understandability. Although transparency, accountability and understandability are often discussed alongside explainability, they do not define it precisely but rather contribute to its broader context as defined in Section 3.2.2. Explainability aims to make systems understandable to humans and leverage attributes of transparency and accountability to build trust. Explainability can act as an enabler to achieve these concepts through a complementary relationship. A complementary relationship exists when two or more elements enhance or complete each other. In such a relationship, each element brings unique strengths that address the weaknesses or gaps of the other for comprehensive

Table 3.7: The Comparative Analysis Results

Concept	Attribute	System Level			Organisational Level	
		Output	Algorithm	Data	Process	Application
Transparency	Visibility	●	●	○	N/A	N/A
	Clarity	○	○	○	○	◑
	Openness	N/A	N/A	N/A	○	◑
	Proactive Communication	N/A	N/A	N/A	○	◑
Accountability	Traceability	●	●	○	N/A	N/A
	Responsibility Allocation	N/A	N/A	N/A	◑	◑
	Answerability	◑	○	○	○	●
	Auditing	●	●	○	◑	◑
Understandability	Ease of Understanding	○	○	○	◑	◑
	Simplification	○	○	○	○	○

*Supported = ●, Limited Support = ◑, Not Supported = ○, and N/A is not applicable at this level

outcomes.

Explainability can connect low-level technical details with high-level conceptual clarity in smart contracts. Transparency provides visibility of code and transactions, while explainability ensures that this information is comprehensible. Accountability provides traceability, while explainability makes decision rights and responsibilities transparent and understandable. Explainability complements understandability by breaking down complex smart contract operations into simpler, more comprehensible explanations.

In summary, transparency, understandability and accountability are integral to explainability. These concepts are interconnected, with explainability serving as the overarching framework. This integration offers a new perspective for designing smart contract systems. Additionally, this perspective encourages the exploration of innovative methods to integrate explainability into existing smart contracts, given that their technical details are

already visible. Explainability can be added as an additional layer to complement understandability and transparency.

3.6 Explainability Early Development Phases

To effectively envision explainability for smart contracts, it is essential to consider early development phases. This section outlines the analysis of explainability requirements tailored to smart contracts' specific characteristics. Additionally, we propose design principles as a holistic approach for smart contract systems lifecycle.

3.6.1 Explainability Requirements Analysis

Given the diverse perspectives and lack of consensus on the definition of explainability, the literature has shifted focus towards addressing aspects that form explainability requirements to understand their significance and contributions. There is no one-size-fits-all explanation, which is why the literature focuses on forming explainability requirements in a generic form, as presented in Table 3.8. One common approach revolves around fundamental 'Ws' questions such as ***who*** the explanation is for, ***what*** needs to be explained, ***why*** the explanation is needed, and ***how*** to deliver it [40, 256, 2, 7, 39]. These questions help shape the requirements for designing explainability within a system [242]. The study by [283] expands the main questions to include archetype derivatives such as why not, what if, and what for, and more complex questions such as what effect and what reason. Another study [275] defines explanation requirements context by source, depth, and scope, covering the origin of information, the level of detail (attribute or model) and the scope (justification or teaching). We observe that the analysis of explanation requirements is influenced by two common factors: the intended audience of the explanation and the specific reasons why the user needs the

explanation [259, 18, 325].

We have deduced that explainability requirements analysis for smart contracts typically revolves around several fundamental questions: the intended recipients of the explanation ***(to whom to explain)***, the purpose of the explanation ***(why to explain)***, the elements requiring explanation ***(what to explain)***, and the methods of presenting these explanations ***(how to explain)***. Inspired by contracting processes, we have adopted the consideration of the timing of explanations ***(when to explain)***. These insights are instrumental in helping practitioners establish specific explainability requirements for smart contracts. Furthermore, employing this strategy can help prioritise the critical information that needs to be communicated to users. To provide further guidance, we elaborate on these fundamental questions in the context of smart contracts.

To Whom to Explain:

Understanding the diverse stakeholders involved in smart contracts is the foundation for defining explainability requirements. Stakeholders may come from diverse backgrounds, including novice users and experts. Each group has distinct roles and responsibilities, which drive their specific information needs. The details required to construct the explanations should be tailored to suit the target group. For example, legal professionals might require detailed contractual language and compliance information, whereas novice users might need straightforward explanations without technical jargon.

Identifying stakeholders in blockchain ecosystems has recently emerged as an essential component for defining blockchain governance and decision-making rights, as highlighted in [187, 14, 234]. Examples of main roles in blockchain include developers, administrators, gateways and participants [272]. However, in the context of smart contracts, knowledge about the various stakeholders is scattered [333, 154]. Examples of stakeholders include

Table 3.8: Synthesis of Explainability Requirements from Various Studies

Study	Title	Explainability Requirements	System
[2]	Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)	Ensuring the model is human-understandable by focusing on what is explained (model) and how it is explained (method), while considering the audience receiving explanation (explainee).	XAI
[7]	Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence	Clarifying decision-making processes, illustrating input-output connections, understanding reasons behind decisions, offering human-readable interpretations, and summarising the rationale for AI model decisions.	XAI
[18]	Explainable Artificial Intelligence(XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI	Involves providing details and reasons to make the model's functioning clear/easy to understand for a given audience.	AI & ML
[39]	Exploring Explainability: A Definition, a Model, and a Knowledge Catalogue	A system is explainable regarding an aspect X if an explainer provides information that enables the addressee to understand X within a specific context.	Not Specified
[40]	Explainability as a non-functional requirement: challenges and recommendations	Focus on addressing specific questions: "what," "why," and "how" that contribute to system transparency.	Not Specified
[242]	Asking 'Why' in AI: Explainability of intelligent systems – perspectives and challenges	Transparency through two mechanisms: responding to "HOW" questions about a conclusion and "WHY" in response to being asked a question by the system.	AI & ML
[256]	Explainability in Human–Agent Systems	Answering why, who, what, when, and how, which define the various aspects of the explanation.	Human-Agent
[259]	Cases for Explainable Software Systems: Characteristics and Examples	Addressing why users seek explanations and focus on tailoring these explanations to the specific needs of the users involved.	Not Specified
[275]	Defining explainable AI for requirements analysis	Explanation requirements are categorised by Source (origin of information), Depth (Attribute or Model explanations), and Scope (Justification/Teaching).	XAI
[283]	An objective metric for Explainable AI: How and why to estimate the degree of explainability	Addressing basic inquiries why, how, what, who, when, and where along with their variations "why not," "for what purpose," "what if," "how much," "what reason," "what cause," and "what effect."	XAI
[325]	What Does It Mean to Explain? A User-Centred Study on AI Explainability	Emphasises that the most important factor in explanation requirements is the human recipient, who asks questions and receives answers about AI models.	XAI

auditors, lawyers, domain experts, and end-users. From the developers' perspective [ID01, ID02], the primary actors are developers who create contracts, miners who execute them, and end-users who benefit from them. These roles can be more specific based on the application and use case [210, 159]. Therefore, it is important to understand the different types of stakeholders in the DApps ecosystem to tailor explanations accordingly [193].

What to Explain:

Any element or process in smart contract development and execution that raises concerns about transparency and human understanding qualifies for an explanation. These elements include decision-making processes, policies, services, data mechanisms, off-chain elements, trigger events, risk management and account privileges. Requirements engineers and designers should consider which elements users may question regarding the contracts' behaviour and actions. Disciplines such as human factors in systems' automation can help define these elements. For example, in the upcoming Chapter 4, we discuss the determination of the informational requirements for explanations in smart contracts using situational awareness theories, focusing on decision-making information [4]. However, other methodologies can also be used to identify elements that need explanation such as scenario-based design, user studies and experiments.

Why to Explain:

The purposes of explanations are often aligned with achieving high-level goals such as understandability, accountability, transparency, verification or trustworthiness. These goals are addressed by explaining specific aspects of smart contracts. For example, users rely on transparent information to make decisions about engaging with smart contracts and regulators need disclosures to verify that contracts comply with relevant laws and standards.

Research in XAI has defined many purposes for explanations, such as justification, information provision, validation, debugging, learning, and building trust [2, 18, 37, 132, 201, 248, 256]. Smart contracts might share similarities with AI regarding automated decision-making which makes some of these purposes functional for their explanations. However, the intricacies and complexities of smart contracts might require more customised purposes that match their contractual nature and enforcement aspects. Examples include facilitating consent, ensuring compliance, and clarifying processes, as will be discussed further in Chapter 5.

When to Explain:

We define four stages to help requirements engineers and designers determine which explanations should be delivered at each stage to provide more contextualised information: (1) Before execution, (2) During execution, (3) Post-execution, and (4) Regular updates/unexpected behaviour. Each stage requires a different explanation tailored to its context. For example, the explanation at the offering stage differs from that needed for decision justification, data clarification or governance decisions after unexpected behaviour. This approach can guide engineers to provide more contextualised explainability based on timing stages.

How to Explain

Explanations should be clear, concise and accessible, using appropriate formats. Various methods can be used to deliver explanations such as text, visualisations or images. We encourage Human-Computer Interaction (HCI) researchers and practitioners to investigate how smart contract information can be delivered in a user-friendly manner. Additionally, user interface design should cater to all users' needs, including those with disabilities. A few efforts have shown HCI interest in blockchain and smart contracts such as [115, 219,

214]. However, this area is beyond the scope of this thesis; therefore, we encourage HCI researchers to explore this direction to enhance explanation delivery.

3.6.2 Explainability Design Principles

Inspired by Privacy by Design [35, 144], we have customised these principles for blockchain smart contracts' explainability. This approach caters to smart contracts' unique properties and intricacies, providing a holistic view to ensure transparency, understandability and accountability throughout their lifecycle. This framework aims to motivate researchers to explore these principles in depth for future smart contracts.

Principle 1: Immutability Requires Proactive Measures

Since smart contracts cannot be altered once deployed, it is essential to take proactive rather than reactive measures. This principle emphasises the importance of anticipating and preventing issues related to explainability, transparency, accountability and understandability from the outset. We encourage further investigation through pre-deployment reviews and testing to ensure proactive measures are in place before deployment.

Principle 2: Explainability as the Default Setting

Explainability by default aims to deliver the maximum degree of explanations to enable stakeholders to understand all aspects without needing additional steps or third-party assistance. To operationalise explainability by default, developers need to consider the stages of when to explain. For example, before the execution phase, smart contract systems should maximise explainability by translating complex code and functions into clear, high-level explanations, making the actions, processes, authorities, risks and decisions understandable to

all stakeholders.

Principle 3: Explainability Embedded into Design

Currently, the design of smart contracts often neglects the human aspects and underutilises the features of existing languages. This principle necessitates rethinking the design of smart contract by embedding explainability in its code. The goal is to equip smart contracts with information that can facilitate explainability. The elicitation of explainability requirements, as explained in Section 3.6.1, offers a tailored framework that provides a meaningful analysis of the explanation needs of specific parties. This approach helps designers rethink the current design to embed all necessary elements for meaningful explanations.

Principle 4: Full Functionality – "Win-Win"

The full functionality principle seeks to achieve a balanced outcome where explainability and functionality coexist without compromise. This principle ensures that integrating explainability features into smart contracts does not reduce their overall effectiveness. It emphasises meeting all organisational and system objectives, including explainability, to provide comprehensive and beneficial outcomes for all stakeholders. The goal is to avoid the notion that explainability competes with other design goals. Instead, innovative solutions should be developed to meet explainability and other critical qualities. This principle presents an intriguing direction for exploring the impact of explainability on other smart contract qualities such as security, privacy and performance. Chapter 7 provides details for this future direction.

Principle 5: End-to-End Explainability – Full Lifecycle Clarity

End-to-End Explainability aims to embed explainability into smart contracts from the very beginning, extending comprehensively throughout the entire lifecycle of the contract. Although our research focuses on user perspectives and interaction, the literature on smart contracts highlights significant development issues due to their complexity. These concerns are relevant for technical developers, non-technical domain experts and collaborative development teams, as discussed in Chapter 2. All these groups contribute to smart contracts' design, implementation and deployment. Therefore, this principle emphasises the incorporation of explainability throughout the entire lifecycle of the contract to ensure that all processes and development decisions remain transparent and understandable. This approach enhances accountability and steers a new direction toward explainability at the development stage.

Principle 6: Keeping It Open Is Not Enough

Smart contracts provide visibility into low-level elements but often fall short in understandability and clarity. Keeping these elements open is a good start, but it does not capture the full potential of transparency. True transparency requires making both low-level and high-level components clear and understandable to foster accountability and trust.

Principle 7: Keep it Human-Centric

All the above principles aim to empower stakeholders to actively understand smart contracts and increase trust in their use. These principles highlight the importance of designers and developers prioritising the interests of individuals throughout the entire lifecycle of smart contracts, from development to user interaction. This thesis seeks to establish a new focus

on developing human-centric frameworks and approaches for smart contract design, reducing reliance on third parties and simplifying the user experience. Our work provides a foundation for explainability through requirements analysis and design principles, aiming to inspire more human-centric approaches.

3.6.3 A Case for Instantiating Explainability Requirements

This section presents an example case demonstrating the elicitation of explainability requirements and how certain design principles can be reflected in implementing a smart contract. We consider a function that represents authority decisions. The choice of this function stems from the frequent occurrence of this pattern in many dApps and has been expressed as privileged accounts in the literature [169, 284, 310, 95]. There are many scenarios for these functions such as adjusting loan collateral requirements, updating reward distribution rates, changing governance rules and ownership and draining contract balance in case of emergency. The pattern of this function involves creating a modifier to restrict the execution by the owner or another specified party. Here is an example of this function where the owner can change the interest rate.

```
modifier onlyOwner() {
    require(msg.sender == owner, "Only owner");
    _;
}
function updateInterestRate(uint newRate) public onlyOwner {
    interestRate = newRate;
}
```

In the requirements elicitation phase, engineers gather the necessary requirements for designing such a function. They should also consider eliciting explanation requirements by

addressing the fundamental questions, as illustrated in Section 3.6.1. For example, we have identified two explanation requirements:

- **R1:** Explanation is required for end-users (*who*), aimed at ensuring transparency and trust (*why*) by clarifying the owner's ability to change the interest rate, the decision's implications on users, and the expected range for these values (*what*). This explanation should be provided before users execute the contracts (*when*).
- **R2:** Explanation is required for end-users (*who*) to justify the decision (*why*) of changing the interest rate by the owner (*what*) after it has been updated (*when*).

For R1, the principle of **Immutability Requires Proactive Measures** necessitates the provision of explanation by connecting the low-level implementation of the smart contract to high-level considerations. This function should be explained to users within the context of the application functionalities. For example, the identification of the roles and responsibilities of the owner, ensuring that end-users are aware of who has the authority to make changes, under what conditions these changes can occur, and the allowable ranges for these changes. This clarification also aligns with the principle of **Explainability by Default**, which requires providing the maximum explanation to achieve the highest level of transparency. When users agree to execute the contract, they should understand the owner's authority and the potential implications of this function on their investments.

For R2, the principle of **Explainability Embedded into Design** implies reflecting this requirement in the design and implementation of smart contracts. Initially, the function is presented previously. We add a new condition for the interest rate changes, assuming that the user is already informed about the acceptable ranges before executing the contract from R1. Both requirements support the principle **Keeping It Open Is Not Enough** which implies that the visibility of code and transactions is not sufficient to ensure user understanding. The updated implementation reflecting the new design is as follows:

```
function updateInterestRate(uint newRate, string memory reason) public
    onlyOwner {
    require( newRate >= minInterestRate && newRate <=maxInterestRate,
        "Interest rate must be within the agreed range");
    uint oldRate = interestRate;
    interestRate = newRate;
    emit InterestRateUpdated(oldRate, newRate, msg.sender, block.
        timestamp, reason);
}
```

In the enhanced implementation, the `require` statement ensures that the new interest rate falls within the predefined range to prevent misuse. The emit statement logs the old and new interest rates, the owner's address who made the change, the timestamp and the reason for the change. When the `updateInterestRate` function is executed, the detailed information in the transaction can be displayed at the application layer in an understandable form to justify the owner's decision. We demonstrated one way of how explainability can be elicited and embedded into smart contract design to evaluate its feasibility and encourage further research and practical implementation of these principles in broader and more technical contexts.

3.7 Discussion

This section discusses our methods for validating our work and the potential threats to the validity of our approach. As our work has different stages, we have adopted different quality assessment techniques from validating SLR [324] and taxonomies [297] to ensure their reliability and usefulness as follows.

3.7.1 Validation

Methods Rigour:

This quality aspect aims to reflect on the validity of our research approach in terms of rigour and robustness [324]. In Section 3.3, we thoroughly explained the four stages of our methodology: knowledge acquisition, knowledge systematisation, comparative analysis and the exploration of explainability in smart contracts. These stages and their respective steps were designed to ensure the rigour of our study. The knowledge acquisition stage consists of predefined steps adopted from SLR, including string formation, source identification, selection criteria and a template for data gathering. Additionally, we detailed the filtering processes to ensure the robustness of the collected data. We also defined the consultation process for gathering knowledge from developers and provided the semi-structured questions. In the knowledge systematisation stage, we described our thematic and iterative process for organising the acquired knowledge into five levels of the system. This systematic approach ensures that the data is categorised and understood in a structured manner. The comparative analysis stage involved defining our method for comparing our findings with standard definitions and explaining the rationale behind choosing specific standards. We outlined the method of extracting attributes, their mapping and the support scale levels to provide a clear method for comparison. Finally, we thoroughly explained how we defined the explainability requirements and principles by synthesising existing explainability requirements from other disciplines and drawing inspiration from privacy-by-design principles to tailor them for smart contracts explainability.

Reliability of Sources:

This quality assessment aims to reflect on the reliability of the findings based on the knowledge sources selected to draw conclusions [324]. We defined strict inclusion criteria during the knowledge acquisition phase, focusing exclusively on peer-reviewed papers. To further validate our knowledge base, we consulted experienced developers to confirm our initial results and provide practical insights to enrich our study with industrial insights. Additionally, all studies selected for the subsequent methodological stages, including the exploration of explainability, were peer-reviewed, significantly cited and of high quality based on criteria from [324]. This meticulous selection process was designed to ensure that our findings and conclusions are both reliable and meaningful.

Orthogonality Demonstration:

We demonstrated the orthogonality [297] of transparency, accountability and understandability by organising these concepts into five distinct levels, as illustrated in Table 3.4. Initially, we started with low-level and high-level classifications. We continuously refined our categorisations through iterative content analysis to generalise similar concepts and elements [63]. However, we defined external data as a standalone level, even though it shares elements and concepts with the process level. Oracles serve as third parties that provide data to smart contracts, which means they operate independently from the contracts' design. Our synthesis includes some aspects of external data processes based on the selected sources. However, they require separate consideration due to their unique role in the ecosystem. The literature on oracles is extensive and warrants a separate study to fully understand their transparency and accountability, which is beyond the scope of this study.

Benchmarking:

It involves validating our findings by comparing them to similar concepts, classifications or studies [297]. In this study, we performed two comparisons. First, we conducted a comparative analysis of our knowledge base with standard definitions. This analysis revealed that some definitions do not satisfy the true meaning of transparency, accountability and understandability, providing deep insights into the claimed or appreciated use of these terms in the context of smart contracts. Second, we compared the objectives of our study with related works addressing similar concepts to highlight the similarities and differences. Table 3.9 presents this comparison, emphasising the distinct aspects of our contributions. Further explanation of this comparison is provided in the following section on related work 3.8.

Instantiation of Explainability Requirements and Principles:

We demonstrated the potential of the explainability requirements and principles through their instantiation using an example case [297]. This practical example showed one strategy of how these requirements and principles can transform requirement elicitation processes and improve the design of smart contracts to accommodate explainability. Moreover, this example underscores the practical impact of integrating explainability into the design phase; however, more research is required to fully realise the potential and implications of this requirement on the smart contract lifecycle.

3.7.2 Threats to Validity

This subsection outlines the potential threats to validity identified in our study, guided by the insights provided in [317, 258].

Internal Validity:

A potential threat to internal validity in our study is the bias in the knowledge acquisition phase, particularly in study selection. The use of generic search strings resulted in the retrieval of many studies, increasing the risk of missing relevant studies. To mitigate this threat, we followed SLR procedures by adopting established steps. We developed a detailed filtering process using library tools and well-defined selection criteria to reduce bias in study selection. Additionally, we incorporated insights from consultation with developers to provide an additional layer of knowledge and further reduce potential biases from the literature review alone.

External Validity:

A potential threat to external validity is the generalisation of knowledge systematisation and the customisation of explainability for smart contracts. Our investigation focused on Ethereum smart contracts due to their prominence in the literature and industry, meaning the systematisation results primarily reflect their state. However, the identified explainability requirements and principles concern early development stages and are generalised to be blockchain-agnostic.

Another potential threat to external validity is that consulting only two developers may not capture the full spectrum of developer perspectives. To ensure we gathered valid input, we selected developers with extensive experience and significant involvement in numerous blockchain projects. The insights we sought from these developers aimed to provide an overview of the current practices in various DApps and the working mechanisms of smart contracts. While our approach focused on generalisable knowledge within the current industrial context, future studies could expand this research to include a broader spectrum of

developer perspectives.

Construct validity:

A potential threat to construct validity in our study is the incomplete presentation of concepts in the systematisation process. We organised the knowledge into five levels: output, algorithm, external data, process and application, based on information gathered from various sources. This categorisation may not capture all relevant levels and categories. To mitigate this threat, we followed rigorous methods for selecting and validating sources and performed an iterative process to refine this categorisation. Despite these efforts, our systematisation does not guarantee completeness, as it is based on the available data and the scope of the study. However, it is designed to be adaptable to accommodate new knowledge variations as they emerge.

Conclusion Validity:

A potential threat to conclusion validity is the possibility of interpretation bias and inconsistent reporting and synthesis of results, which can lead to varying conclusions. To mitigate this threat, we developed a detailed data extraction template and involved two reviewers experienced in smart contracts research to synthesise the knowledge. We employed thematic analysis techniques for interpreting the results [63]. To further reduce interpretation bias, we discussed the initial data synthesis with experienced developers to ensure common agreement on the concepts.

3.8 Related Work

This section compares our study with existing research to evaluate its contribution to the field of smart contracts. To the best of our knowledge, explainability is a new concept in smart contracts and has not been precisely mentioned in the existing literature. However, we selected relevant studies that address similar concepts during the literature review stage and explore core aspects such as transparency, understandability and accountability. We aim to highlight the similarities and differences between our work and related studies. Table 3.9 highlights our contribution's different aspects.

Our study reaches conclusions similar to several existing studies regarding the transparency and accountability of smart contracts and blockchain technology. While these technologies are often claimed to be transparent and accountable at a basic level, they do not achieve these qualities at a more comprehensive level. For instance, the study by [76] emphasise the critical role of human and social aspects in implementing blockchain-based smart contracts. They stress the need for contractual semantics and business rules to be understood and agreed upon by all parties. Similarly, the study by [59] examines the transparency and accountability of current NFTs implementations. While NFTs are often praised for their transparency and accountability, they frequently fall short due to the lack of rigorous document standards. Additionally, the studies by [157] and [305] stress the transparency of blockchain transactions, but they also note that these transactions are not easily understandable to humans.

In terms of domains, the related studies focus narrowly on specific use cases such as NFTs [59], DAOs [76], and ERC20 tokens [305]. In contrast, our study provides a broader application scope by addressing the design of smart contracts on public blockchains without focusing on specific use cases.

The literature on smart contracts frequently centers on developing new languages, as seen in the work by [76]. With over 100 languages for smart contracts [300], this approach adds to the complexity and challenges faced by developers. We argue that the focus should shift towards improving the design of existing languages. Additionally, some studies propose using external and third-party intermediaries to interpret code and transactions to enhance their transparency. For example, [157] suggests a visualisation tool for this purpose, and the findings of [305] show that many interfaces fail to inform users adequately about the implementation and contract processes. Our study addresses this issue by considering high-level and low-level aspects to bridge the gap between implementation details and broader conceptual understanding. The existing literature tends to discuss these aspects separately; we propose explainability requirements to connect them. Our research aims to make smart contract systems more understandable and explainable to reduce the need for such intermediaries.

3.9 Summary

This chapter systematically examined the critical concepts of transparency, accountability, and understandability within smart contracts. Through a detailed literature review and consultations with experienced developers, we organised these concepts into five distinct levels: output, algorithm, external data, process and application. Our comparison with standardised definitions highlighted the misalignment between these definitions and the characteristics of smart contracts, emphasising the need for standardisation and tailored definitions within the blockchain domain. Therefore, explainability is a key requirement that can unlock transparency, accountability and understandability in smart contracts. It acts as a bridge, connecting the visible technical details to higher-level interpretations that resonate with human understanding. Ultimately, we aim to guide researchers and practitioners by providing

Table 3.9: Comparison of Our Study with Related Work

Study	Transparency	Understandability	Accountability	Explainability	Blockchain Type	Specific Domain	Contributions	Research Focus
[76]	✓	✓	✓		Public	DAOs	Development of a formal specification language (SLCML) for legally-binding DAO collaboration	Focuses on legal and business semantics
[59]	✓		✓		Public	NFTs	Examines NFT output information and their records in terms of transparency and accountability, concluding that document standards are necessary to be fair and transparent	Focuses on information provision from the perspectives of Users, Firms and Regulators
[157]		✓	✓		Public	Critical infrastructure	Proposes a user-centric visualisation framework for blockchain transactions to identify malicious events.	An external tool to visualise blockchain transactions, enabling users to have better communication and decision-making
[305]		✓		✓	Public	ERC20 Tokens	Systematic study of unlimited approval in transactions revealing security issues from interacting with DApps and wallets. The results show that few interfaces provide explanatory information for users to mitigate the risk of unlimited approval.	Investigates and analyses transaction process transparency for end-users and the understandability of explanations provided by interfaces for the contract process.
This Study	✓	✓	✓	✓	Public	General, applicable to all smart contracts	Explorations of the core concepts from both low and high level considerations. Introducing explainability design principles and requirements to design human-centric smart contracts.	Focuses on understanding the design and process of smart contracts to reveal their capabilities.

a foundation of explainability in early development through requirements analysis and design principles.

The explainability requirement analysis lays the groundwork for tailoring explainability for smart contracts by addressing who, what, why, when, and how to explain. In this chapter, we covered the main elements of each question. However, the unique characteristics of smart contracts necessitate a deeper investigation into "what to explain," especially their decision-making mechanisms, which will be further explored in Chapter 4. Similarly, "why to explain" also needs further examination. Smart contracts function as agreements with enforceable outcomes which may require specific goals not addressed in adaptive systems. Therefore, Chapter 5 will explore the specific purposes for smart contract explainability.

Chapter Four

ExplanaSC: A Framework for Determining Information Requirements for Explainable SC

Our findings in Chapter 2 indicated a noticeable absence of human-centred methodologies that identify and address smart contract requirements. Given the unique characteristics that influence smart contracts' behaviour and the unexplored aspects of their decision-making processes, further exploration is needed. Therefore, this chapter bridges these gaps by proposing a structured, human-centred framework for defining and eliciting information requirements. Chapter 3 presented the aspects of explainability requirements analysis for smart contracts based on the fundamental questions. This chapter extends this analysis by investigating the 'what to explain' aspect of smart contract behaviour and decision mechanisms. The proposed framework can aid requirements engineers in defining, eliciting and determining information requirements to design explainable smart contract (XSC) systems.

This chapter extends the published manuscript authored by the thesis writer, AL Ghanmi et al. [4], incorporating exact scripts alongside modifications tailored for this thesis.

4.1 Overview

Blockchain and smart contracts have introduced a new paradigm of automation and authority different from intelligent autonomous systems such as AI [208, 307]. They utilise automated decision algorithms, following predefined rules and inputs to execute specific instructions with unique features such as immutability and enforceability. The factors that distinguish smart contract decisions include the governance structure (centralised vs decentralised), the process location (on-chain vs off-chain), the degree of automation and whether the rules are fixed or dynamic. The need for transparency arises from these factors as they significantly impact the trustworthiness of smart contract decisions. However, the literature has overlooked exploring these mechanisms and their impact on smart contract outcomes.

Researchers have highlighted the scarcity of information regarding blockchain governance and decision-making processes, which leads to a limited understanding of how key decisions in blockchain systems are made [100], leaving smart contracts decisions largely unexplored and poorly understood [234]. This knowledge gap is confined to researchers and extends to the users who are part of the decision-making process. As a result, there is a significant lack of awareness and understanding of the decision-making process, raising the need for transparency and explainability to keep humans in the loop.

To tackle these challenges, this chapter aims to integrate explainability requirements in the early development of smart contracts and investigate the characteristics and components of smart contracts' behavioural and decision-making mechanisms, prioritising users' informational needs regarding these processes. Currently, there are no widely adopted or standardised frameworks for the explainability of smart contracts. Therefore, we adopt the principles of Situation Awareness (SA) and Goal-Directed Task Analysis (GDTA), extensively explored in human factors research and human-automation teams [43, 86, 84, 83]. These concepts determine the informational requirements for individuals' needs when oper-

ating in various scenarios, as explained further in Section 4.3.

There are several notable uses of SA in explainable systems, including [264, 43, 85]. These studies have commonly utilised the Endsley's definition of SA [84] to determine what information about agents should be shared with humans across three key levels: perception, comprehension and projection. The frameworks proposed by [43, 85] focus on agent attributes, providing specific information that humans need for their decision-making. On the other hand, the SAFE-AI framework introduced by [264] relates SA-derived information requirement to explainability and leverages XAI to meet these explanation requirements. This is consistent with our objective, which is to determine smart contract information requirements which align with users' needs for awareness, reasoning and projection.

Therefore, we adopt some of the fundamental guidelines proposed in [264], which are presented at SA's three levels definition: perception (input/output), comprehension (model information) and projection (changed inputs/effects of model changes/next agent actions). Our framework tailors [264] considering smart contracts behavioural properties and decentralised structures by introducing new models and constructs that incorporate factors such as logic, data, roles and responsibilities, autonomy, governance and decentralised decision-making mechanisms. Our ExplanaSC extension supports SA three levels to provide XSC explanations along perception (input/output), comprehension (system models) and projection (next actions and future behaviour).

Furthermore, we enrich these three levels by classifying the main components of smart contract behaviour that present system models: business logic, data and roles and responsibilities. Additionally, we categorise the scattered knowledge about decision-making mechanisms into autonomy, governance, process and behaviour. This classification can aid designers and requirements engineers in determining contextualising information for XSC explanations.

Our study is the first to craft an XSC framework and addresses the growing need for explainability as an emerging field. Specifically, the primary contributions of this chapter are as follows:

- It proposes a structured, human-centred framework for defining information requirements for the design of XSC systems to assist requirements engineers in determining the necessary information that should be supported by XSC systems.
- It classifies the main components of smart contracts that govern their behaviour into: business logic model, data model and roles & responsibility model. Additionally, it categorises decision-making aspects into autonomy, governance, process and behaviour, aiding designers and requirements engineers in contextualising information for XSC explanations.
- It addresses the lack of standardised methods for explainability within the context of smart contracts by leveraging SA levels proposed by Endsley [84] and employs GDTA techniques. The ExplanaSC framework promotes the generation of XSC explanations through three levels aligned with SA: XSC explanation for perception, XSC explanation for comprehension and XSC explanation for projection. These levels offer an effective means for engineers to design the information requirements to meet users' needs for awareness, reasoning and projection.

This chapter proceeds in the following sections: Section 2.2 presents the preliminaries of main concepts. The research approach is described in Section 4.3. Section 4.4 introduces the proposed framework, explaining its key components and elements. Section 4.5 outlines the evaluation strategies employed in this chapter. The threats to the validity are discussed in Section 4.6. Following this, Section 4.7 provides an overview of related work in the field. Finally, Section 4.8 summarises this chapter.

4.2 Background

This section provides background information on the decision-making process and hierarchy in smart contracts, which will be referenced throughout the chapter.

4.2.1 Decision-Making Process

Smart contracts operate by following basic “if/then" instructions, expressed in code on blockchains. When predetermined conditions are met and confirmed, a network of computers executes the corresponding activities [164]. Smart contract code can include as many conditions as necessary to ensure that the specified task is completed effectively. The technical aspects of the decision-making process can be summarised as follows:

Logic:

The underlying logic of a smart contract, implemented through code, defines the rules and conditions that dictate how the contract behaves and responds to various situations [208, 308]. These rules are typically encoded using programming languages designed explicitly for smart contracts [280]. This logic outlines the business terms and policies that serve as the foundation for the smart contract’s decision-making. It includes if-then statements, loops, calculations and other computational processes that guide the contract’s decisions. Smart contracts automate decisions through triggers, which are conditions that initiate execution without any need for manual intervention. These triggers can be time-based, event-based, input-based or conditional, depending on the specific requirements of the contract’s use case [308].

Data:

Smart contracts rely on data inputs to make decisions. These data inputs can come from various sources, including internal variables within the contract, external APIs, oracles or even user inputs [231]. The contract uses this data to evaluate predefined conditions and to determine appropriate actions or outcomes. Due to the deterministic environment of blockchains, direct access to real-world data is limited, resulting in a lack of reliable data feeds [231]. Oracles are developed to link smart contracts with real-world information outside the blockchain. Developers can build customised Oracles or use Oracle service providers such as Chainlink [36], Witnet [315] and Paralink [228]. Oracle services have their own data fetching and verifying processes to obtain data from different nodes such as voting or staking. The off-chain process typically involves data sources, data processing and computation that contribute to decision-making that is located outside the blockchain.

Human Intervention:

In some cases, smart contracts may incorporate human intervention as part of the decision-making process, which can be achieved through modifiers and specific functions [280]. Although these contracts are designed to be self-executing and autonomous, there may be instances in which human input is required or desired. For example, a contract may require an administrator to approve a specific transaction or provide additional information before executing a certain action [61]. Human intervention ensures critical decisions are not solely dependent on automated processes and allows flexibility and adaptability in complex scenarios. However, it may impose risks of overprivileged accounts, giving certain powers to a single entity and centralising the decision-making process. Many studies have introduced decentralised decision mechanisms to overcome single of point of failure problem [69, 213]. This mechanism employs the collaboration of stakeholders to make a decision through consensus

mechanisms. Thus, human intervention is an additional layer of decision-making input that influences the smart contracts behaviour.

4.2.2 Decision Hierarchy

Decision-making in smart contract systems comprises two distinct levels: the user level and the system level. This distinction allows for a more nuanced analysis of the decision-making processes involved.

User-Level Decision:

At the user level, decisions are made within the context of an end-user utilising the services provided by smart contracts. These decisions are governed by the contract's predefined rules, logic and conditions, and are executed autonomously to produce outcomes that directly impact the end-user. Decision-making at this level involves the contract executing its code to enforce specified rules and determine actions or results relevant to the individual user [208, 335]. For example, a contract might calculate rewards for an end-user based on predefined formulas or distribute assets according to established conditions. However, in some instances, user-level decision-making may involve additional input from other users or contract owners with special permissions. These privileged accounts, such as administrators or contract providers, can influence certain decisions by approving actions or modifying specific parameters, adding a layer of human oversight to control the automated process.

System-Level:

System-level decision-making pertains to the overall operation and governance of the smart contract system. These decisions affect all stakeholders in the system. Decisions regarding

smart contracts governance are system-level decisions [234]. This category includes determining the rules, procedures and mechanisms for proposing, approving and implementing changes to the system. In addition, it determines decisions related to system upgrades or modifications.

4.3 Research Approach

We use DSR approach [125] that focuses on creating meaningful artefacts to solve identified problems [135]. DSR produces various types of artefacts [199], including representational constructs, methods, models and instantiations. Representational constructs are often used to create frameworks of thought, allowing researchers to better understand and communicate their findings. They provide a means to organise and structure ideas, helping to make sense of complex topics.

Therefore, our research goal is to develop an artefact in a form of a framework for solving a relevant problem per the criteria described in [135]. The DSR methodology is a suitable approach to address the identified problem of *a lack of standardised methods for defining information requirements to explain smart contract decisions from a human-centred perspective*. The framework proposes explanations levels that define which information regarding smart contract decision-making processes should be supported by explanations. This approach aims to guide requirements engineers in determining information requirements to better design smart contract systems that take into account the human in the loop.

4.3.1 Framework Design

The proposed framework comprises three primary components: the compiled definition of the SA, the application of GDTA steps and the operational structure of the smart contracts to facilitate decision-making. The complete iteration of the proposed framework is presented in Figure 4.1, illustrating the integration of the three main components.

Component 1: SA Definition Compilation

The initial component integrated into the framework revolves around the SA concept. Endsley [84] provides an extensive definition of SA, delineating it as the *"perception of elements in the environment within a volume of time and space (level 1), the comprehension of their meaning (level 2), and the projection of their status in the near future (level 3)."* Widely acknowledged and applied in various SA-related studies [313, 102], this definition has played a pivotal role in shaping models and frameworks for SA, particularly in the domain of human-automation teams [80, 43, 85, 264]. This definition is embraced because it comprehensively considers three levels, helping to create SA requirements that can be adapted to various scenarios [84, 262]. Adhering to this definition provides a systematic approach to constructing a conceptual framework.

The information requirements of SA influence the information that XSC systems should provide concerning smart contracts decisions. To align with the three levels of SA definition, the human in the loop needs to be aware of the decision and the corresponding action/outcome (perception - Level 1), understand why the system has taken this action and how it has made this decision (comprehension - Level 2) and be informed about the system's subsequent actions and future behaviour (projection - Level 3). Thus, the determination of this information mirrors the three levels of SA. The questions, as presented in Figure

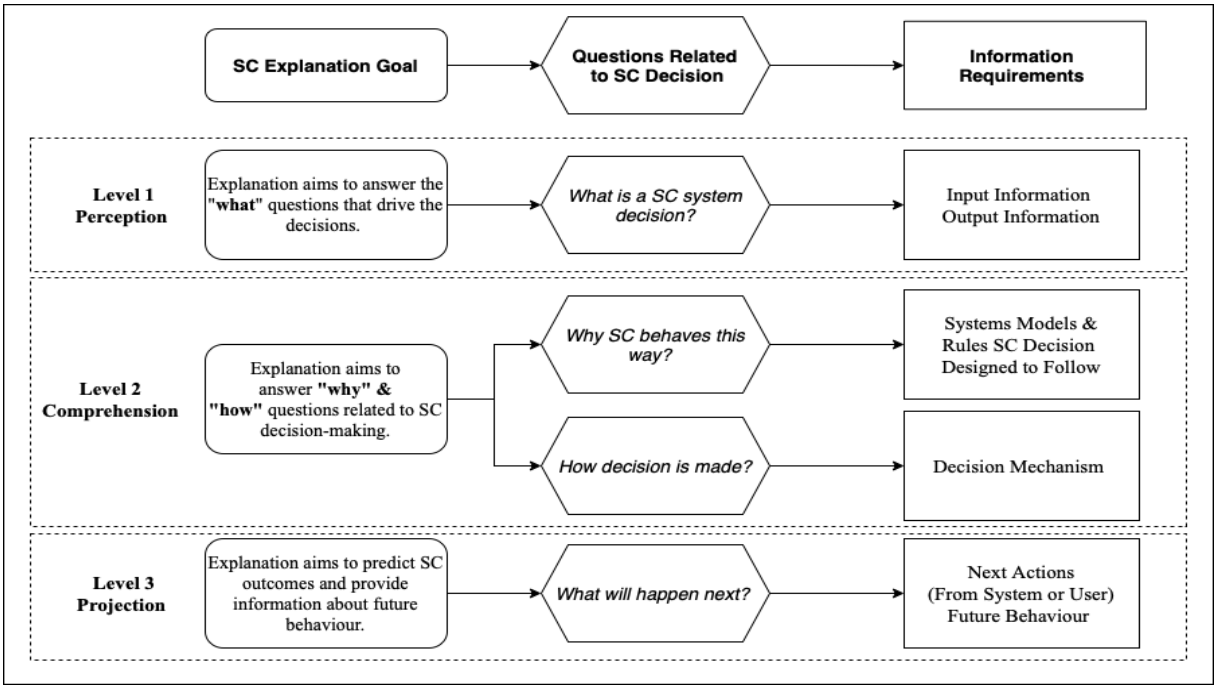


Figure 4.1: The Proposed Framework for Determining Information Requirements for XSC Explanations

4.1, facilitate the identification of a comprehensive list of information requirements that can be embedded into the smart contract design to be provided after the decision occurs. This framework specifically addresses the 'after decision/execution' phase for explanations, as discussed in Chapter 3 regarding 'when to explain'. This timing ensures that users can understand and reason the contract's actions and decisions.

Component 2: GDTA Analysis Integration

The second component integrated into the framework is GDTA, a form of cognitive task analysis using qualitative methodology to address SA requirements [311, 83]. It has been widely used to analyse individuals' SA needs and identify information requirements [83]. GDTA can help uncover SA requirements through three key steps: (1) identification of goals and subgoals, (2) identification of decisions for each goal or subgoal and (3) identification

of SA-level requirements for each decision [86, 83]. We use these steps in the framework construction to reveal informational requirements. We define a set of XSC goals and related questions regarding smart contract decisions as part of the ExplanaSC framework. These goals and questions address fundamental inquiries that users may raise during execution. The perceptual questions seek to understand the actions and input influencing smart contract decisions. The XSC perception explanation goal addresses the "what" questions that have driven the decision. Further inquiries related to comprehending the underlying rationale behind specific decisions and the decision-making process (comprehension). In this context, the XSC comprehension explanation seeks to clarify "why" and "how" questions regarding smart contract decisions. Lastly, the framework responds to inquiries regarding the system's anticipated future actions (projection). The XSC projection explanation aims to give a forward-looking view of the system's future behaviour or actions.

Component 3: Smart Contracts Decision Characteristics

The final component integrated into the framework involves understanding the main factors that influence the decision-making process of smart contracts, which requires understanding their operational structure. Smart contracts rely on logic, data and human intervention to make decisions as thoroughly explained in Section 4.2.1.

4.3.2 Framework Ex-Ante Evaluation

To assess the effectiveness of the artefact, we employ a combination of two evaluation methods to increase the credibility of the proposed framework as recommended by [302]: an ex-ante evaluation through expert feedback and an ex-post evaluation through a case study demonstrating the framework's applicability [302, 241]. The evaluation follows the techniques developed to assess DSR artefacts in [302, 241, 166]. In this subsection, we present the initial

evaluation results obtained from experts for the draft framework, followed by the refined framework in Section 4.4. The demonstration is provided in Section 4.5.

Experts Feedback

The first evaluation method is conducted through a series of expert feedback, which has been identified as an effective method to gather early feedback [19]. We design a survey to gather experts' insights and suggestions through multiple-choice questions and written format. We target participants with blockchain backgrounds and diverse disciplines. Thus, we can comprehensively understand the strengths and areas that need improvements in the proposed framework.

We defined a set of criteria to minimise bias and ensure a diverse and representative selection of experts for our evaluation. First, we targeted key categories of experts, including organisations and DApp owners, developers, researchers and professionals from other disciplines specialising in blockchain. We used various channels to identify these experts, such as LinkedIn, X (formerly Twitter), universities and scholar pages, and GitHub. Our second criterion was to verify the legitimacy of the experts by ensuring they had verified accounts listing their affiliations or demonstrating involvement in blockchain-related work. We excluded anonymous accounts with no background information. We distributed the survey to 50 experts whose profiles matched our criteria. However, a limitation of this approach is that we received responses from only 11 experts who were willing to collaborate with us, as shown in Table 4.1. Another limitation of this method is the potential for subjective interpretation of the framework based on each expert's background and expertise. To maintain the privacy of individuals, we have assigned each expert a unique identification number (ID) to reference the corresponding experts and their respective feedback.

To design the survey questions, we followed guidelines from [241, 19] to identify key

Table 4.1: List of Experts and Their Backgrounds

ID	**Specialty**	**Experiences**
ID-1	Researcher	A university assistant professor with four years of experience in blockchain smart contracts, specifically in oracles.
ID-2	Lawyer	A lawyer with a master's degree in information technology law was previously a lecturer interested in blockchain smart contracts from a legal perspective.
ID-3	Researcher	A university professor specialising in information systems and knowledge management, consultant at a research center with ten years' experience and a researcher with interests in IoT and blockchain.
ID-4	Engineers/ Developer	Software engineer with 8 years of experience in blockchain, cryptocurrencies and smart contracts. CEO and co-founder of a blockchain project.
ID-5	Researcher	IT consultant and researcher with 25 years' experience in many IT fields, including setting strategies for IT directions, researching blockchains and new technologies.
ID-6	Researcher	A university assistant professor with six years of experience in blockchain and Internet of Things.
ID-7	Engineers/ Developer	Senior blockchain developer
ID-8	Business organisation/ Developer/ Researcher	A university assistant professor, consulting and business analysis, with research interests in blockchain and IoT, specialising in automating regulatory and contractual processes and enforcement.
ID-9	Engineers/ Developer	A blockchain developer with extensive experience in DApps, DAOs, ERC20, NFT and token projects and the tokenisation of assets on Ethereum, Binance, Tron and Polygon.
ID-10	Engineers/ Developer	A blockchain developer with extensive experience working on over 30 blockchain and web3 projects, including NFTs, metaverses and decentralised financial systems
ID-11	Researcher	Fullstack developer with more than 4 years in blockchain and Ethereum

quality attributes for evaluating DSR artefacts. According to [19], evaluating a developed artefact requires defining relevant metrics and collecting appropriate data. Common evaluation criteria identified in [241] include usefulness, accuracy, performance, effectiveness, ease of use, robustness, scalability, and operational and technical feasibility. Based on these guidelines, we asked experts to evaluate the framework on attributes such as completeness, simplicity, understandability, operational feasibility, and usefulness [241, 19]. Criteria such as accuracy, performance and scalability are typically evaluated using system metrics.

The survey was designed as an online questionnaire to capture both quantitative and qualitative feedback. It combined yes/no questions, multiple-choice questions, and open-ended responses to gather diverse insights. The yes/no questions assessed specific attributes (e.g., completeness, simplicity) with an optional comment section for experts to provide additional context or clarification. Multiple-choice questions used a five-point scale to measure the perceived importance of explanations within a smart contract and to rate the framework's usability and applicability. As a final step, open-ended questions are given to allow experts to provide their viewpoints regarding additional information requirements, improvements, or any other relevant aspects of the framework. The survey questions and full experts' responses are presented in Appendix C

Explanation Importance Rate: There was a strong consensus among the experts about the significance of explanation requirements within smart contracts. Out of the 11 experts, 10 firmly believed in the importance of providing explanations. Conversely, a single expert believed that offering explanations was unimportant. The outcome of this question is presented in Table 4.2.

The results of the five qualitative evaluation characteristics: understandability, simplicity, usefulness, completeness and operational feasibility are presented in Table 4.3, and the experts' feedback is summarised as follows:

Table 4.2: Experts Rate for Explanations Importance

Importance Scale	**Experts Count**
Very Unimportant (1)	0
Unimportant (2)	1
Neutral (3)	0
Important (4)	1
Very Important (5)	9

Understandability and Simplicity: Most experts (8 out of 11) demonstrated a clear understanding of the framework's structure. Some experts expressed a preference for a demonstration example of the framework, which we have taken into consideration for inclusion in the subsequent section of this chapter. According to (ID-9), the framework has the potential to be user-friendly and well-structured, but its ease of use depends on factors such as smart contracts complexity and the level of technical expertise of users.

Usefulness: The majority of experts, except for one (ID-6), supported the framework's usefulness in facilitating the design of more transparent and understandable smart contracts. The expert (ID-6) expressed concerns about human decision-making and preferred a completely automated system. It is important to note that the framework does not advocate for or against human intervention. Instead, it highlights the importance of providing explanations, regardless of whether the decision is fully automated or involves some degree of human intervention.

Completeness: Based on the feedback, the framework received constructive suggestions to improve completeness. Five out of eleven experts found the framework needed to be more comprehensive, pointing out several aspects to consider such as security considerations, de-

Table 4.3: Summary of Experts Evaluation Results

Evaluation characteristic	Survey Answers	No of Responses	Prominent comments
Understandability	Yes No	8 3	ID-8:“Clear, but needs an example use case for usage illustration."
Simplicity	Yes No	8 3	ID-9:“The proposed framework is potentially easy to use, but it depends on a number of factors, including the complexity of the SC decision ,the domain in which it is used and the level of technical expertise of the users. The framework is structured and provides a clear roadmap for determining the information requirements for SC decisions. This makes it easy to follow and use."
Usefulness	Yes No	10 1	ID-9:“Yes, Once the framework has returned the SC output and the reason why it has returned that output, this information can be used to design SCs that are more understandable and transparent."ID-6:“No, I am not in favour of human intervention"
Completeness	Yes No	6 5	ID-5:“Integrity could be also considered as part of how the decision is made." ID-8:“It covers various important high-level factors. But it might also needs to consider security aspects, user-experience, nodes behaviour, public or private settings, the impact of the underlying infrastructure on the smart contract outcomes, upgradability, portaility, and so on."
Operational feasibility	Yes No	11 0	No comments were given.

cision integrity, legal requirements and unexpected events. Several of these insights shed light on the need for robust measures to ensure the integrity and security of smart contract decisions. Moreover, experts highlighted the importance of incorporating blockchain characteristics and elements directly related to smart contract decisions and behaviour.

Operational Feasibility: Regardless of the framework's operational feasibility, it is noteworthy that all experts unanimously supported its feasibility. This consensus demonstrates their collective belief that the framework can be applied in a practical environment.

Feedback on Potential Usage Scenarios of the Framework The presented results showcase the distribution of preferences for the framework application among the experts. The purpose of this inquiry was to provide experts with a clear understanding of the context in which the framework could be applied. Specifically, we asked whether they could utilise the framework in the following ways:

- **Case 1**: Starting point when designing a new dApp.
- **Case 2**: Testing information requirements in existing projects.
- **Case 3**: Checklist to ensure coverage of informational requirements.

Respondents were presented with these multiple-choice options, enabling them to express various perspectives on the potential applications of the framework. In examining the expert responses to queries regarding the application of our framework, it is essential to emphasise that the goal was not to seek unanimous agreement but to capture diverse perspectives among experts as shown in Figure 4.2.

The results revealed that 6 frequencies indicated a preference for Case 1, while 5 frequencies each expressed interest in Case 2 and Case 3. This distribution reflects the

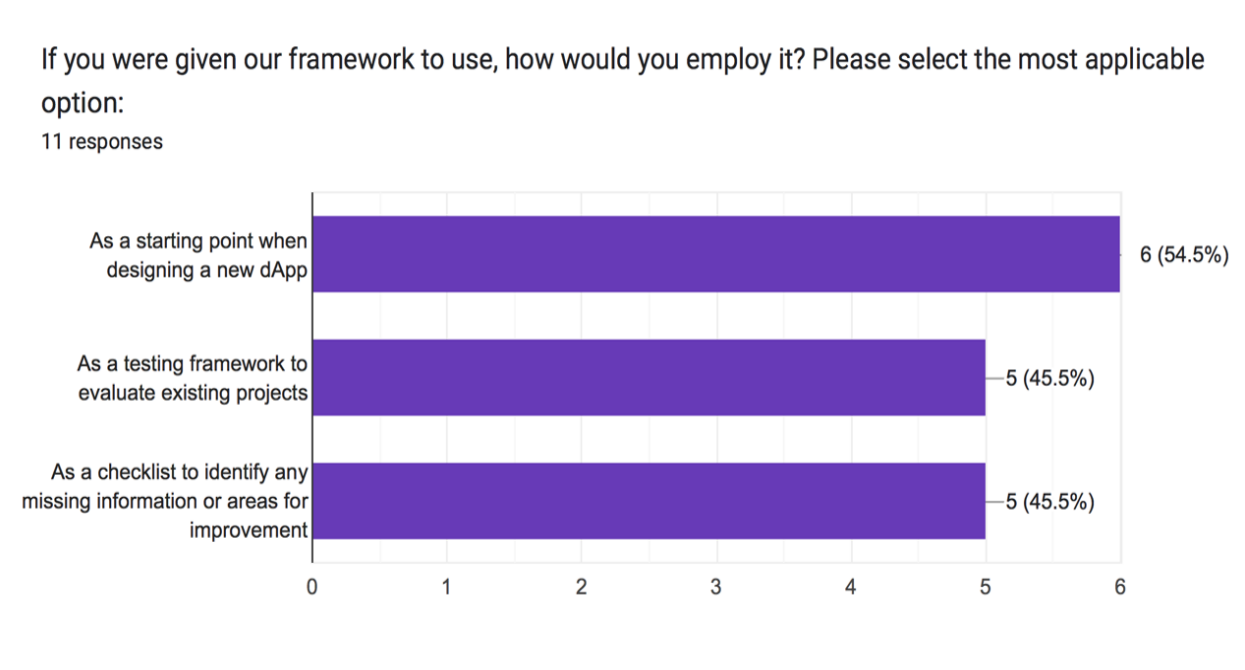


Figure 4.2: Distribution of Expert Preferences for Framework Applications

richness of expert opinions and contributes to a more nuanced understanding of the framework's applicability. Importantly, this diversity aligns with the framework's flexible nature, recognising that different contexts may warrant varying approaches.

Experts' feedback calling for further elaboration and refinement of the framework: Experts provided various insights regarding the information requirements that smart contract applications should support. The lawyer (ID-2) highlighted the importance of clarifying why users need to employ smart contracts and how courts handle these contracts-related matters. Other experts emphasised the need for explanations regarding smart contracts components in relation to blockchain technology. An intriguing perspective was offered by ID-4, who emphasised the significance of end users being able to comprehend how smart contracts function: *"It is important for the end user to view how the smart contract works and sometimes they read the code in order to trust the smart contract. It is their money after all. One good example, Uniswap smart contract it has a complex structure and it is quite hard to*

understand even for developers. Uniswap provide documentation that explains how the smart contract works in detail and they provided some visualisations to help understand it better. if the code isn't then the framework you provide could work for most people." Several experts expressed the need to consider various aspects, including security considerations, legal requirements, upgradability, integrity and user experience. The expert ID-9 supported the framework's elements, emphasising the significance of the smart contract input explanation and purpose in aiding users' comprehension of the decisions.

Additionally, experts provided valuable suggestions for enhancing the overall framework. Some of these suggestions included incorporating information about data storage and considering the legal implications of smart contracts. Experts also recommended illustrating the stages within the blockchain life cycle where the framework would be most applicable. The expert (ID-7) pointed out that the input information within the framework appeared overly detailed and could benefit from some adjustments. On the other hand, the expert (ID-10) highlighted the framework potential in promoting transparency during the minting and burning processes of ERC20 tokens to address concerns related to potential inflationary catastrophes.

Framework Refinements

In light of the valuable feedback provided by the experts, we have made several adjustments and clarifications to the framework, which have been incorporated into the framework as presented in the next Section 4.4. It is important to note that some suggestions received during the evaluation fall outside the scope of this study and require future investigations. For instance, exploring the impact of explainability requirements on non-functional aspects such as security, privacy, performance and costs would necessitate additional research. Recommendations concerning security explanations introduce a layer of complexity that demands

thorough examination. To facilitate this exploration, specific security aspects must be identified to enable non-technical users to assess the security and integrity of smart contracts. While our primary focus remains on decision-making, we invite security experts to provide insights into key elements for ensuring secure and trustworthy smart contracts, particularly for non-technical users. In the following, we summarise the refined framework details to capture the essence of the feedback received.

(i) In response to experts' suggestions regarding smart contract objectives, risks, regulations and unexpected events, we have refined some details of the framework models and stressed the adaptability of the framework to accommodate the suggested models. The emphasis is on providing explanations of the business model, ensuring transparent communication of the contract's objectives, intended outcomes, risk management and regulatory compliance. The updated framework incorporates the mentioned aspects within the business model, addressing the importance of explaining compliance and how smart contracts handle unexpected events and associated risks. Furthermore, the framework is designed to be adaptable to customised models, in which engineers can develop separate models for risk management and regulations to enhance the explanation models.

(ii) A number of aspects of the framework have been refined and enhanced based on experts' feedback. The first update was to make the input information aspect more versatile, allowing for a variety of forms to be used as well as integrating permission and access control aspects into the role and responsibilities model, emphasising clarification as recommended by experts.

(iii) In addressing the relationship between smart contracts and blockchain aspects, it is essential to note that our framework assumes implementation on existing public blockchains such as Ethereum [89] and Solana [279], where decisions are made within the

public blockchain's infrastructure. The decision-making processes related to blockchain are controlled by the governance of the public blockchain itself. Other studies, such as [234, 187, 139], explore how blockchain governance decisions are made. However, the decision-making in our framework relates to DApps developed for specific use cases such as insurance and trading, which have their own governance and mechanisms separate from blockchain governance. We believe that further research on the behaviour of smart contracts within blockchain nodes is necessary to enhance the explanation requirements. However, XSC explanations for projection can serve a vital role in keeping humans informed and engaged during unexpected events, including blockchain governance decisions, upgradability, hard forks, attacks and high-risk events.

4.4 The ExplanaSC Framework

As the adoption of smart contract systems continues to grow and people begin to interact with increasingly complex decentralised environments, the need to maintain adequate SA arises as a potential solution to preserving human engagement with these systems. Our framework aims to uncover information requirements that smart contracts should provide to their users that can be incorporated into system design to promote XSC.

Inspired by SA-oriented design [83], XSC design begins with a requirement analysis phase where information requirements necessary to comprehend smart contract decisions are identified. In order to effectively determine the information requirements for each smart contract decision, our proposed framework defines a set of questions to achieve the desired goals for XSC explanations. These goals call for an understanding of what the smart contract accomplishes explicitly in terms of input and output information, why it behaves in a particular manner by considering the logic it follows, how the decision-making process is

executed through the decision mechanism employed and what subsequent actions and future behaviours can be anticipated. We illustrate the framework application using user-level decisions and systems-level decisions in Figure 4.3 and Figure 4.4, respectively. These figures provide an overview of the key components and interactions involved in the process.

The framework provides a systematic approach to identifying the information requirements for a subset of key decisions in smart contract systems, focusing specifically on those that directly impact users. This information is provided to address users' needs for awareness, reasoning, and projection. Therefore, the framework prioritises a targeted selection of decisions most relevant to users, rather than attempting to include every possible decision, many of which may not require explanations. Given that smart contract environments incur fees for deployment and execution based on code complexity, this approach helps minimise costs by focusing on decisions that impact users and require explanations. The three levels of XSC explanations thus offer engineers an effective means of designing and delivering explanations that keep human users well-informed and reducing unnecessary computational expenses.

- **XSC Perception Explanation:** This level provides information about the decisions made by the smart contract system and the actions it has taken. It focuses on the input and output of a decision, ensuring that users are aware of the relevant data, parameters and resulting outcomes.
- **XSC Comprehension Explanation:** At this level, the framework offers explanations that help users understand the reasoning behind the system's decisions and how they were reached. It covers the logic, rules and mechanisms employed by smart contracts, providing clarity and transparency in the decision-making process.
- **XSC Projection Explanation:** This level of explanation enables users to understand the possible future behaviour of the smart contract system. It provides insights into

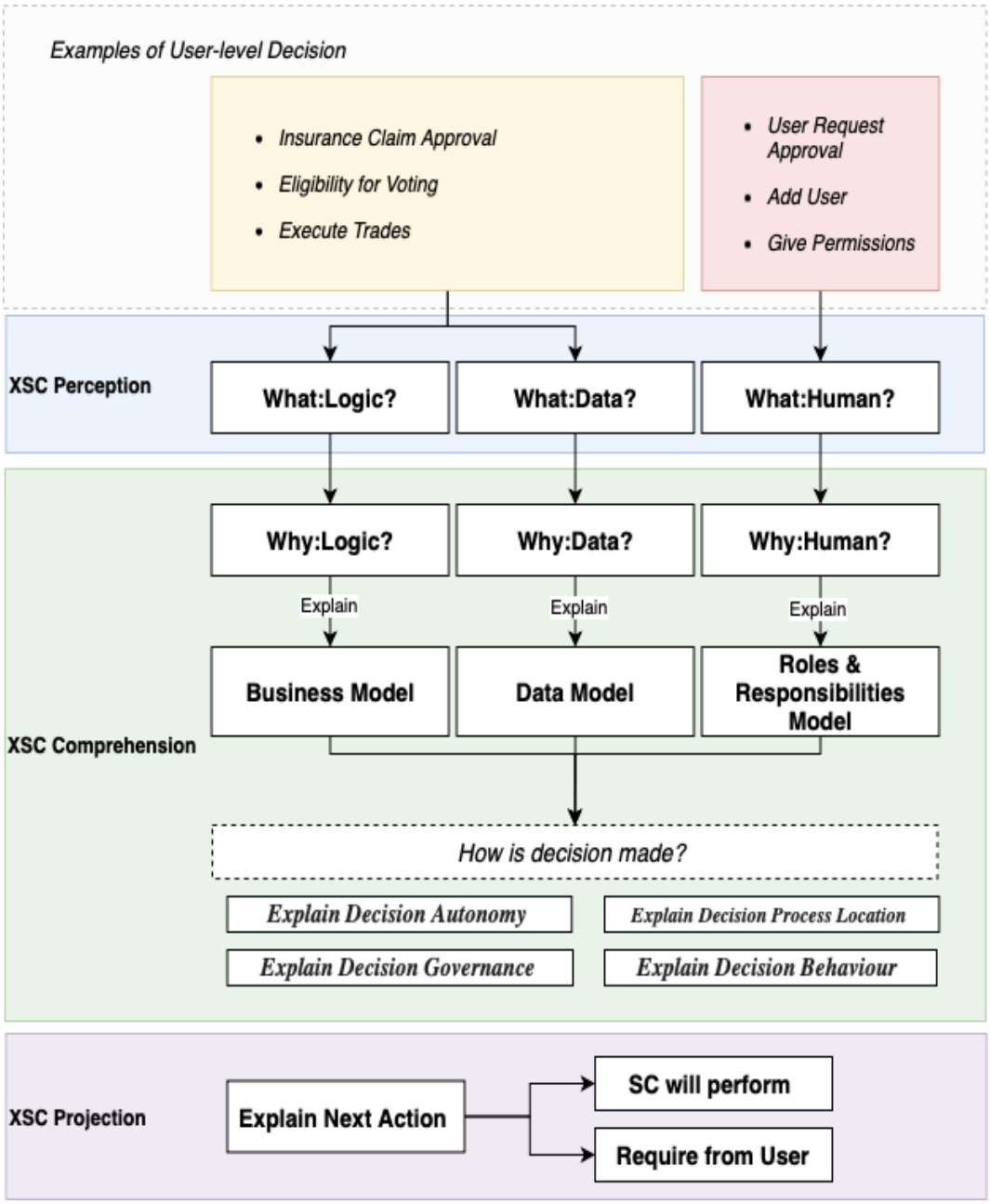


Figure 4.3: The ExplanaSC Application Summary of User-Level Decisions

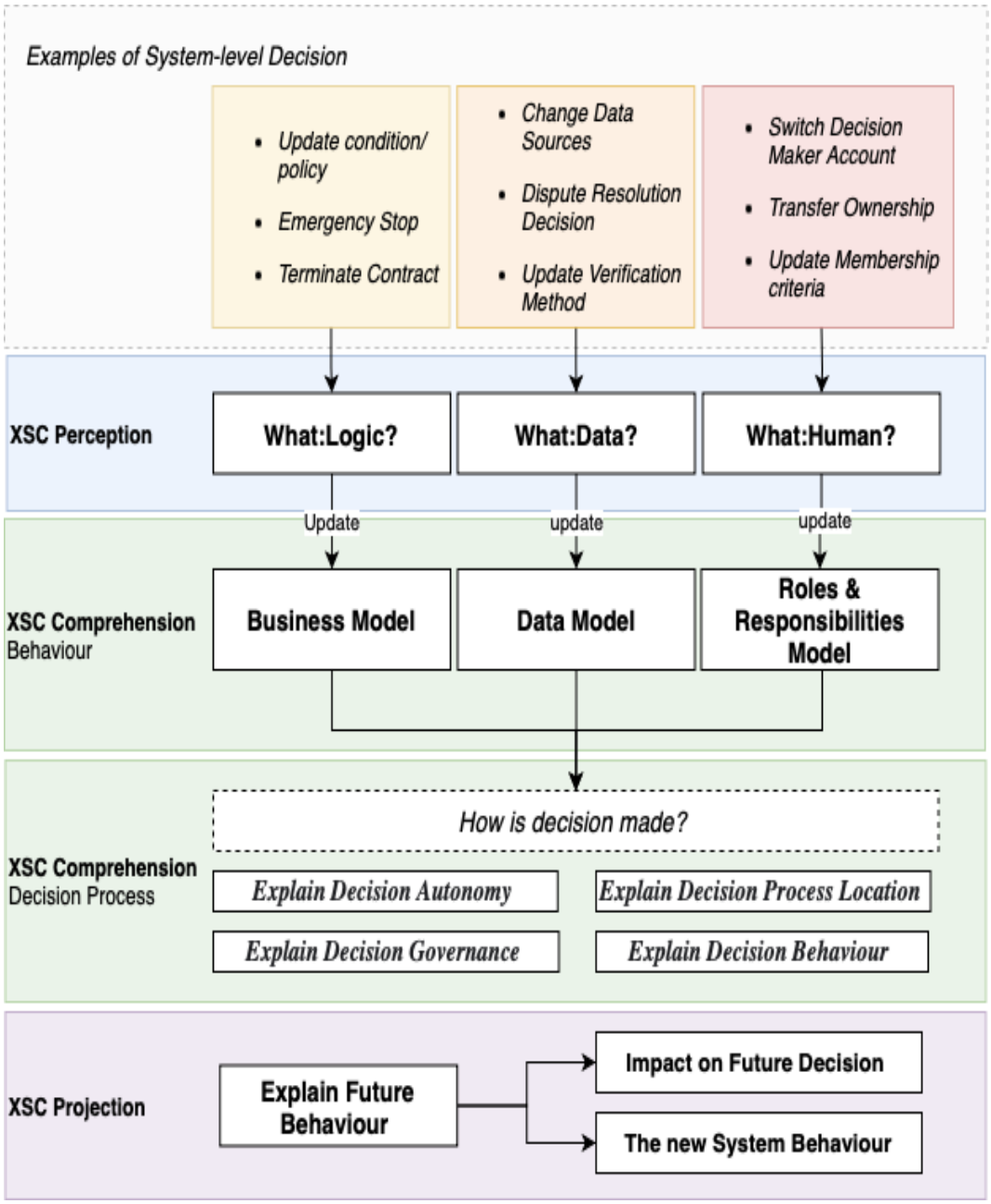


Figure 4.4: The ExplanaSC Application Summary of System-Level Decisions

next actions, consequences and any changes that may occur over time.

The ExplanaSC framework is designed to be adaptable across different smart contract projects, irrespective of their specific domains. The delivery mode of explanations, such as natural language or interfaces, is not our primary focus; instead, we focus on the informational content provided. In the following subsections, we offer a detailed description of each level of XSC explanations based on knowledge gathered from existing literature on smart contracts. As the technology is still emerging, we have consulted a broad range of sources, including white papers and grey literature, to gain insights into current research and industry practices related to smart contract decisions. By synthesising these perspectives [244], we aim to develop a comprehensive overview of the smart contract decision-making process and draw meaningful insights regarding XSC explanations. The final iteration of grouping concepts found in the literature is presented in Table 4.4.

4.4.1 XSC Explanation for Perception

The perception explanation (level 1) focuses on providing input and output information that determines decision outcomes in a manner similar to that of intelligent systems [181]. The input information includes data, parameters or conditions used in the decision-making process. For example, it may include user input, time-based events, digital signatures or specific input triggering contractual actions [280, 290, 208]. The smart contract can also receive input from external sources such as oracles or data feeds to incorporate real-world data into its logic [231, 213, 307, 36]. This level of explanation is critical to understanding the direct parameters that influence the system's decision-making process and ultimately lead to the final outcome. The framework proposed by [264] introduces input and output information for level 1 of explaining XAI behaviour, which is akin to the proposed level 1 explanation in the context of XSC. Moreover, input and output information plays an

Table 4.4: Final Iteration of the Grouped Knowledge

No	Source	Type of Study				Elements					Decision Knowledge								Level	
		Survey	Approach/Method	Use Case	White/Grey Liter	Input/ Output	Future Behaviour	Business/ Logic	Data/ Oracles	Roles	Fully-Automated	Semi-Automated	Centralised	Decentralised	On-Chain	Off-Chain	Static	Dynamic	User-level	System-level
1	[208]	✓				✓		✓			✓				✓				✓	
2	[307]	✓				✓					✓			✓	✓				✓	
3	[308]	✓				✓	✓	✓	✓		✓			✓	✓	✓	✓		✓	
4	[234]		✓							✓		✓	✓	✓	✓	✓				✓
5	[280]				✓	✓				✓	✓				✓		✓		✓	
6	[231]	✓				✓			✓				✓	✓						✓
7	[36]				✓	✓			✓							✓		✓		✓
8	[61]		✓							✓	✓	✓		✓	✓			✓	✓	✓
9	[69]		✓							✓		✓	✓		✓				✓	
10	[213]		✓			✓								✓						✓
11	[13]	✓				✓		✓	✓			✓		✓	✓	✓			✓	✓
12	[187]	✓						✓		✓				✓	✓	✓				✓
13	[38]		✓					✓		✓		✓	✓					✓	✓	✓
14	[203]				✓			✓						✓	✓					✓
15	[122]				✓			✓		✓			✓	✓				✓	✓	✓
16	[306]		✓						✓					✓		✓				✓
17	[89]				✓	✓			✓				✓	✓			✓	✓		✓
18	[110]				✓				✓	✓			✓	✓					✓	✓

Table continues on the next page.

No	Source	Type of Study				Elements					Decision Knowledge								Level	
		Survey	Approach/Method	Use Case	White/Grey Liter	Input/ Output	Future Behaviour	Business/ Logic	Data/ Oracles	Roles	Fully-Automated	Semi-Automated	Centralised	Decentralised	On-Chain	Off-Chain	Static	Dynamic	User-level	System-level
19	[29]		✓			✓			✓					✓		✓				✓
20	[3]		✓						✓					✓						✓
21	[321]		✓							✓				✓						✓
22	[218]			✓				✓				✓	✓						✓	
23	[68]			✓				✓		✓	✓				✓				✓	
24	[330]			✓		✓	✓				✓			✓	✓				✓	
25	[174]			✓		✓					✓			✓						✓
26	[240]				✓					✓	✓	✓		✓		✓				✓
27	[172]				✓					✓		✓	✓				✓			✓
28	[225]			✓				✓						✓						✓
29	[107]				✓					✓				✓	✓	✓		✓	✓	✓
30	[290]			✓		✓				✓		✓		✓		✓			✓	
31	[49]			✓				✓				✓		✓						✓
32	[177]		✓												✓	✓			✓	
33	[211]		✓					✓					✓	✓	✓	✓				✓
34	[226]				✓			✓				✓	✓	✓				✓		✓
35	[224]				✓					✓			✓					✓		✓
36	[139]	✓				✓	✓	✓				✓	✓	✓				✓		✓

important role in system-level decisions, including updates, protocol changes and policy adjustments [13]. Input information originates from various sources such as governing bodies, consensus mechanisms or predefined rules [139]. Similarly, output information at the system level reflects the outcomes and consequences of these decisions.

Numerous applications can provide level 1 XSC explanations since they only describe the system's inputs and outputs. However, this task for some applications can be challenging due to the inherently complex nature of smart contracts, which often involve numerous conditional statements, interactions among participants and execution on blockchains. One primary technical challenge lies in the need to preserve privacy while offering detailed explanations, especially when sensitive data is involved [308]. Engineers must ensure the technology can protect the confidentiality of certain information while still providing sufficient explanations for users.

4.4.2 XSC Explanation for Comprehension

Smart contract systems present a unique approach to automation and authority. The comprehension explanations (level 2) entail identifying specific causal information that individuals or groups need to comprehend regarding the system. In order to address this level, the framework explores two key questions concerning XSC explanations: "Why does smart contract behave this way?" and "How is the decision made?" In the behaviour subsection, we explore the underlying factors and models that influence the behaviour of smart contracts. In contrast, in the decision mechanisms subsection, we focus on the decision-making mechanisms employed within smart contract systems. The outline of these subsections and their contents are illustrated in Figure 4.5

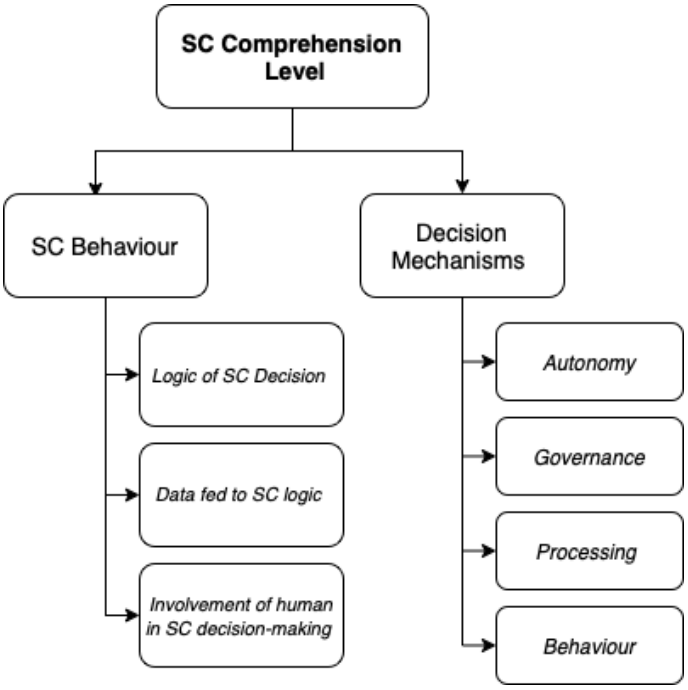


Figure 4.5: Smart Contract (SC) Elements in XSC for Comprehension

Smart Contract Behavioural Structure

The operational structure of smart contract decision can be categorised into three main components: business logic rules [208], input data influencing the outcome [231] and potential human involvement in decision-making [61]. These components can determine the behaviour of decision-making process. For example, requirements engineers need to identify information regarding the rules followed by the smart contract logic and data processing mechanisms that generate a single value for decision logic. In instances where human participation is involved, engineers must also determine the responsible parties and their roles in the decision-making process. Therefore, we employed the separation of concerns concept [66] and transformed these three components into main generic models: the business logic model, the data model and the roles and responsibilities model. These models guide engineers in determining the specific information requirements that govern smart contract behaviour and require explanation at level 2.

Our list of models is not exhaustive, as projects can vary in complexity and requirements. The aim of exploring these models is to understand smart contract behaviour in order to guide the provision of comprehensive explanations. While the business logic model, data model and roles and responsibilities model serve as the core pillars for explaining smart contracts behaviour, additional models may be necessary depending on the specific project and its decision-making elements. These additional models can address unique factors such as regulatory compliance, security model, risks management model or other project-specific models. Our three models serve as a starting point for understanding and explaining the behaviour of smart contract while allowing for the flexibility to incorporate additional models as needed.

Business Logic Model: The decision logic of smart contract is tailored to accommodate the requirements and processes of business specifications, including rules, conditions, contract scopes and objectives, applicable laws or regulations and dispute resolution procedures that govern interactions and transactions [13]. This logic serves as the foundation for the contract's decision-making capabilities, enabling it to autonomously execute transactions, validate conditions and enforce contractual obligations [308, 187, 38, 211]. Further, smart contracts often incorporate system-level conditions to handle unexpected behaviour, accidental transfers, updates or termination of contracts, which can only be executed by specified parties [203, 49, 226, 139]. However, such control raises ethical concerns, as some parties may have complete control over contract assets, which reduces users' confidence if these rights remain unexplained [122].

A business logic model can provide a structured representation of the policies and rules embedded in the smart contract system that guide the system's behaviour [218, 68]. Engineers can utilise this model to determine information regarding a specific decision logic outlined in our framework. This clarification helps provide targeted and contextual expla-

nations to users based on the specific rules and policies involved. Additionally, users can refer to the business logic model to gain insights into smart contract contractual agreements, overall regulations, policies and risks.

Data Model: When it comes to smart contracts, it is essential to be familiar with the data processes used to feed the contract decision-making to understand the outcomes produced by the system. The data model addresses questions related to the origin, selection, collection and verification of data that results in a single value [306, 36]. One of the key challenges presented in the literature is referred to as the "oracle problem", which pertains to the difficulties associated with importing data to the blockchain [231, 13]. This issue raises questions concerning the reliability, authenticity and correctness of external data sources, which are often provided by oracles that determine the outcomes of smart contracts. There can be uncertainties surrounding these data sources and mechanisms, particularly in projects where custom oracles are developed [89]. These custom oracles may operate as black boxes whose inner workings and data sources are not openly disclosed. While some DApps have demonstrated their ability to provide information about data sources and processes, as illustrated in [110], there is still uncertainty surrounding these mechanisms in many projects.

As part of the process of determining information requirements concerning data models, we adopt questions presented in Ethereum oracles documentation[1], given that Ethereum is the most popular blockchain platform supporting smart contracts development and execution [308]. These questions can help developers determine information requirements that can shape XSC explanations. Table 4.5 illustrates the questions and the possible information requirements with examples. These information requirements include identifying the sources or origins of the data, such as sensors, human input or APIs, and specifying calculations or algorithms used in processing and collecting mode [231, 306, 29, 3, 36]. This level of de-

[1]https://ethereum.org/en/developers/docs/oracles/

Table 4.5: Examples of Data Information Requirements Inspired by Ethereum Oracles Documentation

Question	**Information Requirement**	**Examples**
Which sources used to obtain the information requested by SC? Where is the data coming from?	Declaration & accessibility of data sources	External API links, Data Providers (e.g., Chainlink), human-generated data, sensors, on-chain data or oracle data
What is the process for extracting data values from data sources?	(1)Data retrieval, filtering, cleaning & transformation, (2) Data verification & validation	(1) Retrieval: establishing connections and protocols to access the data. Processing as removing irrelevant or inaccurate information and data conversion, (2) Authenticity proofs e.g., TLS & TEE Proofs Consensus-based validation of information e.g., Voting/staking or Schelling point mechanisms
How many oracle nodes can participate in retrieving the data?	Collection mode	Centralised, decentralised or semi-decentralised
Is there a way to manage discrepancies in oracle reports?	Dispute resolution procedures	Reputation-based oracle selection, voting or staking
How should submissions be filtered and aggregated into a single value?	Aggregated methods	Voting, mean or median

tail promotes trust and confidence in the system, particularly in decentralised environments where users need to evaluate systems and outcomes without relying on central authorities. The primary objective is to describe the key components of the data processes that directly influence specific decision outcomes. The explanations should be designed to be accessible and transparent, as excessive technical complexity may hinder user understanding.

Roles & Responsibilities Model: The decision-making processes in smart contract systems entail different involvement levels among the relevant parties [321]. Usually, roles are identified by the access control mechanism implemented in the logic, which defines the constraints and what each user can do [280, 172, 290, 224]. An example would be the administrative roles given to certain parties to register new users [61, 69, 107]. The coexistence of automated and human side-by-side decisions provides flexibility and adaptability; however, it also adds complexity to the reasoning process [240]. Many concerns have been expressed in the literature regarding who has the authority or power to make decisions and how stakeholders can be empowered in decision-making processes [234, 122, 187]. For example, there is a need to determine who is authorised to change administrative roles in the contract. This issue has led to the exploration of the centralised authority making critical system governance decisions. Many studies have recognised this problem and proposed solutions to decentralise the decision-making process [61, 38, 68]. These solutions are based on providing the same functionality but in a distributed manner, reducing the risk of a single entity having complete control.

However, our goal is not to distinguish between centralised or decentralised human decision-making processes but to be explicit about the roles and responsibilities of the various actors in the system. We aim to help in outlining information about parties involved in the system and clarify their responsibilities and permissions. By understanding the specific access permissions of stakeholders, we can determine the information requirements necessary

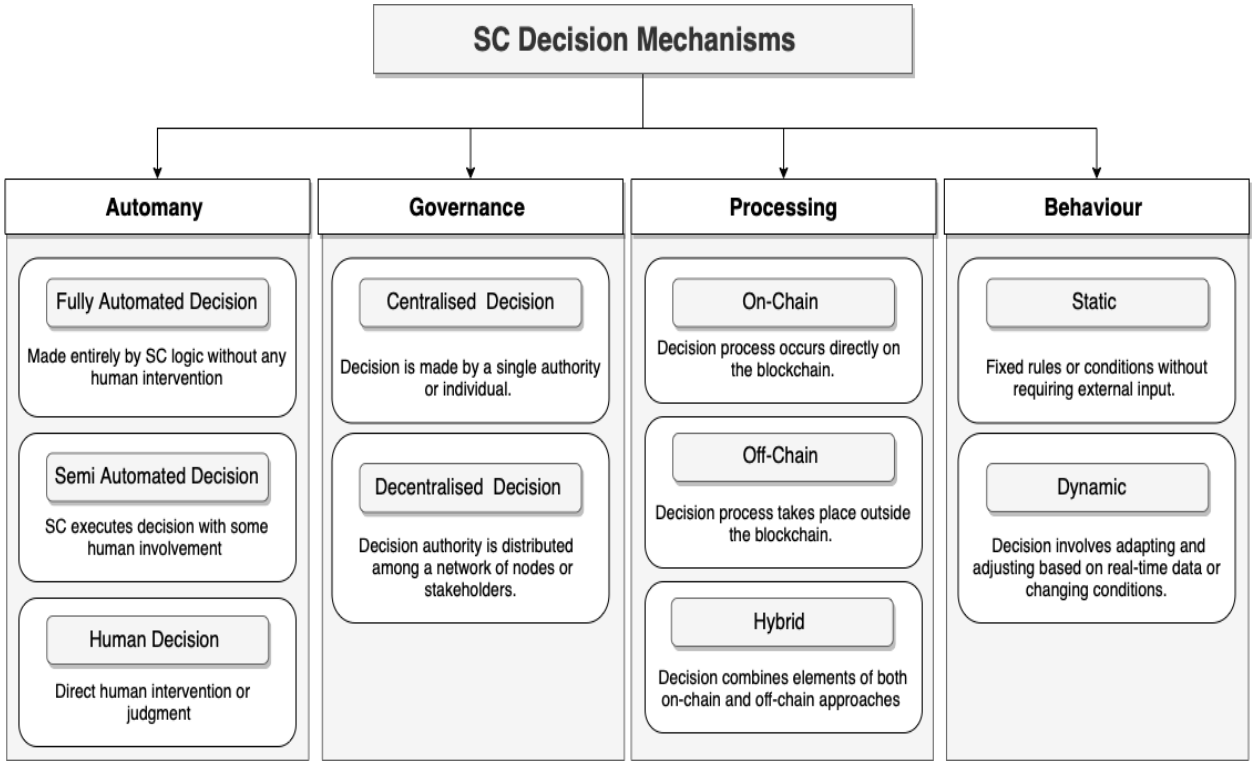


Figure 4.6: Groups of Decision Mechanisms in Smart Contract (SC) literature

for effective explanations. For example, the audit report in [110] expressed concerns about some centralised authorities. It recommended that users should be aware of and the roles and responsibilities of all parties should be clarified. Our framework focuses on providing information about the party that contributes to a particular smart contract decision and clarifying their authority. Additionally, the information requirements should include the membership or participation criteria if the decision is community-based.

Smart Contract Decision Mechanisms

In the previous section, we highlight the main elements that explain the behaviour of smart contract decisions. However, these decisions can involve different processing mechanisms. It is possible to combine centralised and decentralised decision-making, where some decisions are fully automated while others in the same system are semi-automated. This variety

of decision mechanisms can create a complex environment that makes it challenging for users to understand the system's decisions and outcomes. Throughout the literature, there is scattered knowledge about various mechanisms for decision-making in smart contracts. We use synthesis analysis [244] to identify shared characteristics among decision-making mechanisms and organise them into coherent groups. We established four main groups: autonomy, governance, processing and behaviour. Figure 4.6 summarises the findings.

Autonomy refers to the automation of decision-making. Fully automatic decisions refer to situations where the contract's execution is determined solely by its code and the data it receives [208, 307, 308]. Therefore, the outcome is solely determined by the logic programmed into it. Several use cases have been discussed regarding the use of smart contract systems for automated decision-making [330, 174]. By contrast, semi-automated decisions involve a level of human intervention or discretion during the execution process [172]. For instance, a decision may be designed to execute automatically based on certain conditions, but it requires human approval before it can be fully executed [218, 61, 68, 61, 69, 290]. Human decisions, on the other hand, are not automated within the smart contract itself. These decisions rely on direct human involvement and decision-making that can occur outside the project's scope. These decisions may include dispute resolution or protocol upgrades [240, 234, 49, 226, 139]. Understanding the degree of automation is important, as it determines the level of autonomy and reliance on human involvement in decision-making.

Governance refers to instances in which the ultimate decision-making power rests with a single authority, such as a governing body and a designated administrator, or power is shared among multiple parties. In centralised governance, decision-making authority rests with a single authority or entity [234]. This central authority has the ability to make and enforce decisions for entire systems such as modifying contracts, rules or data [218]. A

majority of smart contract systems appear to be controlled by a single centralised authority due to the operational implementation such as using modifiers in Solidity [280, 69, 172]. This implementation allows centralised control by assigning privileges to specific accounts to manage contract execution. The literature claims this approach violates the core principle of decentralisation in blockchain technology [61]. However, it has been considered a valid approach due to its simplicity, scalability and performance, which makes it suitable for a wide range of business domains and requirements.

Decentralised governance distributes decision-making authority among the participants or stakeholders involved through various consensus mechanisms such as voting mechanisms [3, 225, 107], off-chain processes [234, 240] and multi-signature contracts [290]. The goal is to empower participants to have a voice in shaping the rules, policies and updates to address the centralised single point of failure problem. Compared to centralised governance, decentralised decision-making offers several advantages. It reduces counter-party risks and promotes inclusiveness by involving communities in decision-making [49]. Community decision-making refers to situations where a group of individuals or a specific community has a vested interest in the smart contract's outcome and actively makes decisions related to it [203, 49] such as modifying policies or determining the role of administrators [61, 38, 139].

Processing: The processing location of decision-making in smart contracts can be categorised as on-chain or off-chain. On-chain processing refers to decision-making that takes place directly on the blockchain, with the decision logic encoded within the smart contract itself, thus ensuring the decision is transparent and cannot be compromised [208, 307, 203]. However, on-chain capabilities are limited by the blockchain's computing power and storage capacity, resulting in higher transaction fees [177]. Therefore, many systems have explored off-chain computation [36, 290], which offers greater flexibility, increased processing power and enhanced storage capacity. However, off-chain computation may introduce additional

complexity and potential risks, as it can be perceived as a “black box" where the correctness of the decisions is challenging to verify [29, 240, 308]. The choice between on-chain and off-chain decision-making depends on the specific requirements and limitations of the applications [234, 187, 107]. Some applications may prioritise fully decentralised and trustless decision-making processes, while others may prioritise speed and flexibility. Several studies have proposed hybrid approaches to leverage both mechanisms' strengths and mitigate their limitations [177, 211]. Understanding processing location can affect the overall user experience. For example, on-chain decisions provide transparency and immutability but can be costly. On the other hand, off-chain decisions offer cost-effectiveness but add additional layers of complexity and potential points of failure.

Behaviour: The behaviour of smart contract decisions can be classified as static or dynamic based on their structure and ability to adapt over time. Static decisions are predetermined and fixed, defined within the smart contract. Once the contract is deployed, these decisions remain unchanged and are not open to alteration or adjustment [308, 280, 172]. By contrast, dynamic decisions are designed to be flexible and adaptable. They incorporate mechanisms that enable updates, adjustments or refinements based on real-time data [61, 36, 38]. Currently, smart comtracts are designed with increased flexibility to accommodate various upgrades and changes such as policy updates, modifications to human roles or responses to high-risk events [89, 226, 224, 139]. Decisions made in this manner have the advantage of being able to adapt to changing conditions and respond to new information. However, dynamic decisions also introduce additional complexity to contract execution, as they can change over time, making it more challenging to understand the resulting outcomes and potentially raising unethical risks [122, 224]. For example, smart contract may have emergency functions, allow contract funds to be withdrawn by owners or suspend operations in response to high risk events [89, 107]. These dynamic decisions are processed based on real-time events impacting overall system operations.

4.4.3 XSC Explanation for Projection

Understanding smart contract behaviour relies on users' comprehension of level 1 and 2 information, including input, rules and decision-making mechanisms [308]. However, to ensure users have a forward-looking understanding of the system, it is also necessary to explain future behaviour [181]. This explanation includes information about the system's next actions, required user actions and anticipated behaviours in response to external factors such as evolving regulations or system upgrades [234, 110]. The projection explanations (level 3) are vital in offering insights into how the system is expected to adapt and behave in the face of changes. These explanations help users anticipate future actions and adapt accordingly. However, it is important to acknowledge the limitations of level 3 explanations. Predicting future behaviour and accounting for the impact of management decisions require ongoing research and development efforts to enhance predictability, adaptability and comprehensibility. The uncertainties arising from external factors and evolving circumstances may result in unexpected outcomes, making it challenging to explain future behaviour. Nonetheless, effective communication and feedback through explanation can enhance user awareness.

4.5 Framework Ex-Post Evaluation

This section presents the artefact ex-post evaluation process to assess its performance and effectiveness [199]. The evaluation follows the techniques developed to assess DSR [302, 241, 166]. In the following subsection, we describe the second strategy used to evaluate the proposed framework: demonstrating its applicability through a case study. The corresponding work of developing the framework, evaluation strategies and results, along with the implementation of smart contracts, has been uploaded to a public repository [2].

[2]https://github.com/halghanmi/ExplainableSC/tree/ExplanaSC-Framework

4.5.1 Framework Demonstration

The second approach, ex-post evaluation, focuses on demonstrating the utility of our framework. This evaluation method involves implementing the designed artefact in an actual context, allowing for real-world testing and assessment [53, 282]. However, applying our framework in a real-world context proved challenging, particularly due to the lack of collaboration from DApp providers. We reached out to several decentralised application providers to explore partnerships and opportunities for integration. However, despite our efforts, we received little to no response. Many DApps are startups or individually owned, operating with limited resources and focusing on specific use cases. This resource-constrained and individualised nature limited our ability to apply the framework directly, highlighting the challenges of securing cooperation in the current DApp ecosystem, where external collaborations or framework integrations often fall outside their immediate project scope. Due to these limitations and constraints, we demonstrate the framework practicality and effectiveness through a real-world scenario and example implementation. This approach allows us to showcase our framework's potential application and benefits in a simulated scenario that mirrors real-world challenges and requirements.

The demonstration activity provided a lighter version of the evaluation, focusing on showcasing the functionality and usage of the framework in solving a specific problem instance [233]. This activity aims to illustrate the practical application of the artefact. We assume the role of a framework user in an artificial setting, simulating real-world usage scenarios.

Smart contracts have limited real-world applications due to perceived novelty and developmental immaturity. However, one of the areas where smart contracts have been considered industrially is the insurance sector. Chainlink [36], a leading oracle service provider, has customised oracles to provide flight data for insurance DApps. Additionally, the flight

insurance case is best suited to illustrate all the concepts outlined in our framework. It covers the three central components of the framework: logically representing policy, integrating external data sources and involving humans in the decision-making process. This choice reflects the rich context it offers for demonstrating all the outlined framework constructs.

We introduce a flight insurance smart contract case that decides on users claim approval. Passengers can purchase insurance coverage to safeguard themselves against potential flight delays. In order to determine eligibility for a claim, the smart contract function stipulates a minimum delay time for flights, which is 120 minutes. Consequently, the claim is automatically rejected if the delay is less than 120 minutes. When the flight delay exceeds or equals 120 minutes, the claim amount is determined as 10% of the ticket price and is awarded to the claimant. To obtain real-time flight information, such as flight schedules, statuses and delays, we assume the utilisation of flight data APIs.

We divide the case into three distinct scenarios covering different decentralised decision mechanisms. In the first scenario, the claim approval process is entirely automated. When the delay meets the minimum duration requirement, the claim is automatically processed and the claimant receives the amount designated for the claim. In the second scenario, we introduce an additional layer of administration to the claim approval process. The claim cannot proceed until it is approved by the insurance company, which presents a semi-automated decision. The third scenario is a system-level decision where the policy minimum requirement can be updated. The decision is decentralised through a voting mechanism among decision makers in the company.

Based on our framework, we classify the decision mechanisms presented in the above scenario as follows: The delay policy is dynamic, as one of its parameters can be updated through scenario 3. The claim logic and system-level decisions are processed on-chain to ensure transparency. Flight data is obtained from an off-chain process with a centralised

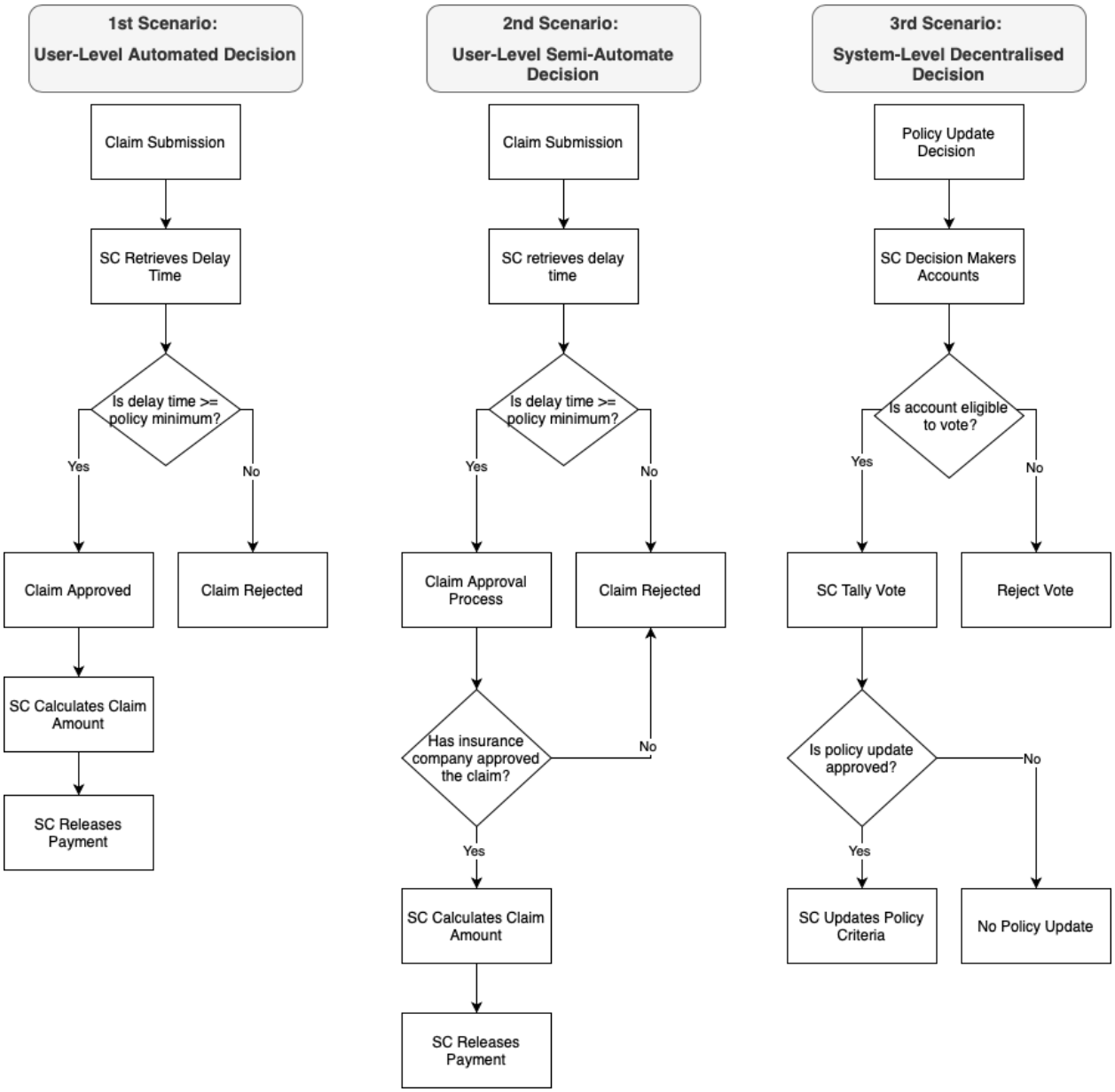


Figure 4.7: Flight Insurance Smart Contract (SC) Functional Requirements

mechanism, i.e., from one external source. The autonomous nature of the decisions varies: Scenario one presents a fully automated decision, while scenarios two and three present semi-automated decisions, i.e., with a degree of human involvement. Finally, scenario one and two governance is centralised, presented as a single party to approve and automate decisions. Scenario three involves a decentralised decision where a group of stakeholders vote for the decision.

The functional requirements presenting the flight insurance smart contracts before applying our framework are defined in Figure 4.7.

Using our framework, we can identify the breakdown of information requirements that can be translated into XSC explanations. However, it is imperative to note that developing smart contracts with lengthy explanation strings can increase transaction costs and contribute to blockchain bloat, which challenges the immutability and increases computational overhead for consensus mechanisms. Therefore, we utilise our framework to identify new information requirements, enhance existing data and implement and record this information on a blockchain. Using blockchain transaction information is critical for ensuring the credibility and trustworthiness of smart contract information. Subsequently, the recorded information can be transmitted to other parts of the smart contract system, such as the front-end, to generate comprehensive explanations without relying on lengthy strings within the smart contract code. Therefore, requirements engineers can define these information requirements using our framework, which will be implemented in smart contract systems for retrieval and explanation building. Here, the three levels of XSC explanations can offer a flexible framework for engineers to determine the design and provision of these explanations.

Tables 4.6 and 4.7 present the defined information requirements for the flight insurance case using our framework. In our public repository [3], we defined, designed and implemented

[3]https://github.com/halghanmi/ExplainableSC/tree/ExplanaSC-Framework

these information requirements alongside the functional requirements in a simulated smart contract project. This repository showcases the implementation of three scenarios before and after using our framework. We enriched the design with event logs that record all the information requirements and enhance the existing functions. This information can be retrieved through transactions to build explanations. To aid requirements engineers and designers with a clear demonstration of envisioned XSC explanations, Tables 4.6 and 4.7 present examples of XSC explanations utilising the defined information requirements. These examples illustrate how information requirements can be tailored to provide context-specific explanations for XSC.

Table 4.6: Information Requirements for Smart Contracts (SC) and XSC Explanation Examples: 1st & 2nd Scenarios

Level	**Information Requirements**	**SC System New Features**	**XSC Explanation Examples**
Perception	Input Data Information	The SC can return the specific input data	"**119** minutes and **50** seconds"
	Outcome Information	The SC can return the output e.g., claim amount	"Claim is rejected because the flight delay time is **119** minutes and **50** seconds"
Comprehension	Policy Information	The SC can return the policy	"The policy is a minimum **120-minute** delay; the system automatically approves the claim and initiates the payment process"
	External Data Sources	The SC can provide the sources of data	"Flight information is retrieved from **https://api.flightstats.com/...**"
	Claim Amount Calculation	The SC can return parameters used for calculation	Your claim amount is £**50**, as this represents **10%** of your ticket price of £**500**.
	Human Decision Justification	the SC can record justification from admin	The reason behind rejecting your claim is the **missing required documentation**.
Projection	Processing Time	The business model stated the processing time of admin decision	After you submit your claim, you can expect to receive the claim decision within 2 to 3 business days
	Next Action Information	The business model stated the next steps after the claim decision such as further verification or dispute resolution	"In some cases, additional verification may be required to process the claim fully. If necessary, our team will reach out to you to request any additional documents or information"

Information in bold is recorded in smart contracts and retrieved through transactions to build the example explanations.

Table 4.7: Information Requirements for Smart Contracts (SC) and XSC Explanation Examples: 3rd Scenarios

Level	**Information Requirements**	**SC System New Features**	**XSC Explanation Examples**
Perception	Current policy requirement	The SC can return the specific input data	**120** minutes.
	Proposed policy update		Increase the minimum requirement to **180** minutes.
	Voting options		In **favor** or **against** the policy update.
	Real-time vote count		**20** votes in **favor**, **8** votes **against**.
Comprehension	Rationale for update	the SC can record justification from stakeholders	The current minimum delay requirement **may not capture significant delays impacting passengers**.
	Potential impact	The business model stated the potential risks	It may result in fewer claims being approved due to the increased minimum requirement.
	Decision-making mechanism	The business model stated the governance of system-level decision-making	The decision is made by voting among the company decision makers.
Projection	Voting period	The SC can return the specific input data	**10** days.
	Outcome processing time	The SC is automated when conditions are met	Immediately after the end of the voting period
	Next steps	The SC can return the specific input data	If the policy update is approved, the minimum requirement will be updated to **180** minutes.
	Actions after decision outcome	The business model is updated with the new policy	Notify users with the updated details.

Information in bold is recorded in smart contracts and retrieved through transactions to build the example explanations.

4.6 Discussion

This framework has been demonstrated to be effective in identifying the information requirements for explaining smart contract decisions. We leverage the concepts of SA and GDTA to propose three levels of XSC explanations to capture the necessary information elements that meet users' informational needs. Moreover, the framework's adaptability makes it a versatile tool for addressing different use cases within the blockchain ecosystem.

In contrast to other studies, such as [234, 21, 187, 139], which primarily focus on understanding the governance of blockchain networks and the decision-making processes at the macro level, our framework is dedicated to exploring the decision mechanisms within DApps. Notably, many decisions are made in DApps that do not directly involve the blockchain itself. Instead, the blockchain serves as the underlying network that hosts these DApps. With a focus on the practical aspects of smart contract decision-making, we aim to contribute to the understanding of decentralised decision-making and its potential impact.

Although the framework has been applied to a typical usage scenario, its application to domain-specific industrial cases can invite requirements engineers to identify and further refine the XSC requirements for the specific domain, consequently validating its usability in context. However, the demonstration serves as an initial validation of the framework's potential. Our future work will look at the framework's application in different domains to better learn about its effectiveness and usefulness within and across domains, possibly through a field study.

4.6.1 Threats to Validity

A potential threat to the validity of integrating the three main components (SA, GDTA and smart contract decision operational structure) into the framework is the risk of oversimplification or overspecialisation that could overlook important nuances or variations in different smart contract systems or contexts. To address this threat, we adopted a comprehensive and iterative approach for the framework construction. We conducted an in-depth analysis of each component, considering its fundamental concepts, principles and techniques, following the solid theoretical establishment in the field.

Additionally, the framework was designed to be applicable across different smart contract systems while accommodating specific variations and requirements within each system. We sought to balance establishing a structured framework and allowing for customisations and contextualisations, recognising that flexibility and adaptability are essential. To achieve this, we employed the separation of concerns concept, which allows us to break down the operational mechanisms in smart contract systems and focus on the core elements relevant to decision-making that is applicable to any smart contract decisions. This approach ensures we capture the essential information requirements and design considerations while maintaining adaptability.

It is important to acknowledge that the limited number of studies and potential bias in the available literature may introduce threats to the internal validity of the framework. However, we have taken several steps to mitigate these potential threats and provide a comprehensive perspective. We utilised synthesis analysis to gather scattered knowledge about smart contract decision-making processes. By systematically grouping and categorising information from various sources, including use cases, blogs and white papers, we aimed to overcome the sample size limitations. This iterative process helped provide a more comprehensive understanding of the key elements involved in smart contracts decision-making.

Another potential threat to the validity of the evaluation process is the interaction of selection and treatment. This refers to the possibility that the selected group of experts may not be fully representative of the larger population that the framework aims to generalise to. In our case, the framework was evaluated by experts who possess extensive knowledge of smart contracts technology. To mitigate this threat, we took several measures. Firstly, we ensured that the group of experts selected for the evaluation represented diverse backgrounds, including experts from business organisations, lawyers, developers and researchers. This diverse representation helps capture different perspectives and insights. Additionally, during the survey process, we clearly indicated the applicability of the framework and its intended use. This way helped align the expectations of the experts and provided them with a clear understanding of the context in which the framework would be applied. As a result, we aimed to reduce any potential biases or misunderstandings that could arise from the interaction between the selection of experts and the treatment i.e., evaluation process.

4.7 Related Work

SA has been extensively explored in the human factors literature, particularly within human-automation teams navigating complex environments [84]. The literature offers various definitions and frameworks of SA [84, 22, 277]. The concept of SA refers to the ability of an individual to comprehend and perceive their environment, including cognitive and perceptual processes, social interactions and physical and environmental factors that affect human performance [82]. Empirical validation of SA has been conducted in diverse contexts, shedding light on its relevance to human factors issues such as workload, fatigue and decision-making [313, 81].

This concept relates to technology design in various industries, including automated

systems, which support human decision-making. SA has significantly influenced the design of automated and intelligent systems, offering valuable insights into determining the information to be presented to users. For example, in the study by [43], the SA-based-Agent Transparency (SAT) model directs the selection of information that should be conveyed about the system for human decisions. Another framework, as outlined in [264], employs the SA concept to define the information that XAI systems should share. SA goes beyond providing information; it provides relevant information tailored to the user's needs and current circumstances [229]. SA is, therefore, a valuable approach to determining the information users need, as we propose in this study. We employ an approach similar to that described in [43, 264, 44], focusing exclusively on smart contract systems.

The XAI field has explored several studies on informational requirements for achieving explainability. Recent publications, including [319, 52], introduce the concept of "explainability scenarios" as informational resources in XAI design, with [52] specifically focusing on fraud detection. Another study by [275] proposes a framework with three dimensions—Source, Depth and Scope—categorising explanation requirements based on origin, detail and coverage. Additionally, the study by [15] discusses different methods for defining information needs that are human-centred, including question banks and role-based requirements engineering. However, these methods and frameworks are frequently contextualised and tailored to AI characteristics. Given the nascent nature of explainability requirements in smart contracts, our decision to adopt the SA framework is rooted in its systematic structure and versatile applicability across diverse domains and systems.

In the blockchain and smart contracts domain, a comprehensive study conducted by [307] explores their applications, challenges and future trends. The research recommends exploring the integration of blockchain technology and AI for optimised social management and decision-making. It further suggests improving performance through accurate system descriptions, future outcome predictions and prescriptive recommendations. Our proposed

framework integrates some of these elements to enhance social systems management by offering detailed decision-making descriptions, insights into future performance and rationale explanations of the system's behaviour. Furthermore, additional studies have explored the parallel interaction between AI and blockchain, showcasing various use cases [31, 204, 232, 8, 301]. However, discussions on explainability have predominantly focused on the AI component of the technology.

Several papers have also examined the expansiveness of programming languages and the legal aspects of smart contracts, emphasising the need for improved human understanding and interaction through the use of specialised languages to implement these contracts that are more user-friendly [42, 77]. However, these efforts primarily focus on the terms and conditions of these contracts. In contrast, our work addresses smart contract decisions as layered systems, incorporating various aspects such as business models, legal considerations, human involvement and external factors that collectively contribute to the decision-making mechanisms.

Although explainability requirements in the context of blockchain smart contracts have not been precisely examined, our contribution represents the initial effort toward achieving explainable and understandable smart contracts.

4.8 Summary

Despite the growing interest in smart contracts for decision-making, existing research has predominantly focused on the technical aspects, often neglecting the role of human factors in designing such systems. This research gap has led to the development of a conceptual framework to address this issue and support requirement engineers in determining the information requirements necessary to explain smart contracts behaviour. Our framework recognises

the importance of integrating SA concept and the GDTA as core components to guide the specification of information requirements. Through that, it acknowledges the need for users to comprehend the decision-making processes, understand the underlying rationale behind decisions and grasp the implications of those decisions.

Furthermore, the conducted exploratory overview of the literature has provided valuable insights that guide engineers and designers in identifying the specific information requirements needed to achieve perception, comprehension and projection. We have proposed three fundamental models based on the concept of separation of concerns. These models are business logic, data and roles and responsibilities, which serve as key pillars for smart contract behaviour. One of our contributions involves grouping scattered knowledge about decision-making processes by organising them into autonomy, governance, processing and behaviour. The added value of our framework is that it considers each decision individually and determines the information requirements for rationalising that decision. The evaluation process helped in gathering valuable insights and feedback from experts which allowed us to understand the practicality, usefulness and potential limitations of the framework. Additionally, the framework demonstrated its utility through an example case of flight insurance, further validating its relevance and applicability.

Chapter Five

Evaluating Smart Contracts Explanations to Reconcile Surprises

In Chapter 4, we propose a human-centric framework to determine the information and explanation requirements for designing explainable smart contracts, which address the 'what to explain' aspect of the explainability requirements analysis presented in Chapter 3. In this chapter, we explore the 'why to explain' aspect of our analysis. Smart contracts function as agreements with enforceable outcomes, necessitating specific goals customised to their unique characteristics. Therefore, this chapter contributes to the broader comprehension of smart contract explainability requirements and lays a theoretical foundation for a generic evaluation method inspired by the metacognitive explanation-based (MEB) theory of surprise. Based on the theory, surprise can act as a mechanism directing attention to conflicting information in the environment, signaling the need for explanation to reconcile the discrepancy. Hence, we propose explainability purposes as valuable resources for designers and engineers to evaluate explanation needs, embed necessary explanations to reconcile surprises and understand cost implications.

5.1 Overview

The shift to DApps introduces a new operational paradigm that might challenge users accustomed to centralised systems. As users navigate the unfamiliar territory of smart contracts—known for their immutability and enforceability—there is a potential for *surprises* due to the irreversible nature of decisions. This risk is particularly pronounced when the provided information is inadequate for users to comprehend the actions executed by smart contracts [124]. The absence of information provision can result in epistemic uncertainty, also known as subjective uncertainty, stemming from a lack of knowledge or incomplete information about a system [26, 167]. This situation may lead to what is referred to as automation surprise [268], where individuals experience surprises due to underestimating or miscalculating the capabilities of automated systems.

This chapter is motivated by the metacognitive explanation-based (MEB) theory of surprise [106], which posits that surprise is fundamentally connected to explanations that help us make sense of the world and resolve the surprises we experience. This theory conceptualises surprise as fitting new information into existing mental frameworks, emphasising the role of explanations in connecting information regarding settings with event outcomes to resolve surprises. In our perspective, surprise acts as a mechanism directing attention to conflicting information in the environment, signaling the need for explanation to reconcile the discrepancy. The computation of the MEB theory requires an explanation that connects the setting and outcome of the scenario, providing a direct measure of alignment based on available contextual information.

Our approach is rooted in leveraging surprise as a key driver for introducing explanation purposes within smart contracts. In helping to envision explainability, we present the concept of 'explainability purposes' as integral resources for evaluating and designing explanations for smart contracts. Unlike a technology-centric approach that begins with

what smart contracts can explain, we advocate for a scenario–purpose perspective that explores the types of explanations users might require in using smart contracts systems. This subtle shift in perspective carries significant design implications. Specifically, it shifts the design focus towards potential usage scenarios and associated challenges, which serve as a foundation for generating potential technological advancements.

This chapter aims to set a foundation for explainable smart contracts by identifying purposes of explainability in smart contracts. Smart contract designers and requirements engineers can embed explanations along the following lines: Explain to clarify, explain to justify, explain to ensure compliance and explain to facilitate consent. We employed dual investigation to formulate the explainability purposes. First, we examine the essential characteristics of legally binding contracts to understand how explainability supports enforceability [158, 34, 97]. Second, we explore explainability goals within XAI to enrich smart contract design with established practices [18].

To assess the added value of these purposes, we develop a novel evaluation framework derived from MEB theory to assess setting and outcome information. We aim to measure potential surprises in scenarios where information related to justification, clarification, consent and compliance is inadequate or lacking. Using our approach, we evaluated two real-world lending DApps, showcased strategies for implementing explanations and conducted cost trade-off analyses. Specifically, the main contributions of this chapter are as follows:

- It introduces the concept of “explainability purposes" as integral resources for evaluating and designing explanations for smart contracts. Drawing inspiration from established practices in contract law and XAI, We posit that smart contract designers and requirements engineers can embed explanations to clarify, justify, ensure compliance and facilitate consent.

- It develops a novel assessment framework inspired by the MEB theory and its foundational principles. This framework establishes a theoretical basis for a generic assessment approach, systematically evaluating the potential for surprises arising from insufficient or absent information. The MEB frmaework is designed to be a valuable tool for software engineers and designers, aiding in evaluating the need for explanations in smart contract systems. It highlights areas that may require improvement by assessing setting and outcome information. Drawing on two cases, we exemplify the working of the framework and evaluate its applicability.

- It explores the potential trade-offs in terms of costs associated with integrating explanations into smart contract systems. We contribute a nuanced understanding of the economic implications considering deployment and execution costs. This insight provides valuable perspectives on the financial aspects of incorporating explanations in smart contracts.

This chapter proceeds in the following sections: Section 5.2 offers essential background information, Section 5.3 introduces the proposed explanation purposes and Section 5.4 outlines the methodology of the assessment approach inspired by MEB theory. In Section 5.5, we have applied the framework in real-world applications, assessing surprises, explanations and costs. Section 5.6 discusses the findings and the potential threats to the validity of the work. Section 5.7 discusses related work and Section 5.8 summarises this chapter.

5.2 Background

This section presents background information on smart contracts settings, outcomes and uncertainties, along with the theoretical foundation of the MEB theory of surprise. These concepts will be referenced throughout the chapter.

5.2.1 Setting Information

We refer to 'setting information' within the context of smart contract projects as the critical information provided to users to allow them to understand the contract. Setting information includes various aspects such as the project's objectives, purpose in a specific context, core functionality, decision-making mechanisms and terms and conditions. It also addresses legal compliance matters such as data protection, privacy and other legal considerations [76, 99, 12, 117] . In addition, it specifies consent mechanisms, which are essential for managing user consent in transactions involving sensitive data.

Setting information is typically presented to users through the project's front-end interface, acting as a comprehensive reference for all aspects of the contract. Users can easily access necessary information, review and agree to the specified terms and proceed to execute the contract using the provided details. While some projects rely on code comments to explain contract functionality, this method may not be user-friendly, especially for individuals with limited technical knowledge. This challenge becomes particularly pronounced in understanding complex contract terms and decision-making processes.

5.2.2 Outcome Information

The term 'outcome information' refers to transaction details that capture and record information about the execution of contracts. When a smart contract is triggered, specific aspects are typically documented in the transaction information. Transactions are initiated primarily by external interactions or state-altering operations within the contract, such as variable modifications, event emissions, fallback functions, self-destruct operations and internal payment transfers [46]. When a user or another smart contract triggers an external function call to modify the contract's state or initiate specific actions, it leads to a transaction [333].

Transaction information includes the smart contract address, specific function details, sender and receiver addresses and additional input data during a transaction's interaction. Examples of transaction information can be viewed through blockchain explorers such as Etherscan [91] and BscScan [30], showcasing actual transactions recorded in the blockchain. However, specific details may not be explicitly included in the transaction information, such as changes in the internal state of the smart contract. Instead, these changes are stored within the contract's internal storage and are not immediately visible in the transaction data. Similarly, function restrictions, such as using modifiers for pre-check conditions or limiting access to certain parties, are not apparent in the transaction details. In a DApp, multiple smart contracts serve distinct purposes, not all directly engaging in user or blockchain interactions. Some manage internal logic, data storage, or other non-transactional functions [11].

5.2.3 Smart Contracts Uncertainties

Uncertainties, in the context of smart contracts, can intricately link to aleatory and epistemic uncertainty, representing two distinct forms of uncertainty widely discussed across various disciplines, including statistics, engineering [167, 138] and artificial intelligence [9, 26].

Aleatory uncertainty, or stochastic uncertainty, measures the intrinsic variability or randomness of a system or process. It is characterised by unpredictable events or phenomena that are inherently uncertain. In the context of smart contracts, aleatory uncertainty could refer to unpredictable external factors affecting contract execution, such as market price fluctuations or unforeseen events in the real world.

On the other hand, epistemic or subjective uncertainty arises from a deficiency of knowledge or incomplete information about a system. This uncertainty results from a lack of comprehensive understanding and is potentially reducible with the acquisition of addi-

tional data or a deeper comprehension of the underlying processes. In the context of smart contracts, epistemic uncertainty might stem from incomplete or imperfect knowledge about the operation and behaviour of these contracts.

This chapter focuses on addressing epistemic uncertainties, mainly through the lens of providing explainability. By integrating explainability into smart contracts, we aim to better manage and reduce these uncertainties, offering users deeper insights into contract behaviours to enrich their understanding and expectations. However, it is important to recognise that explainability alone might not resolve all uncertainties. Therefore, we consider explainability as one aspect of a broader, human-centric design strategy. This strategy is designed to improve user experience and understanding across various dimensions, fostering more informed and confident interactions with the technology. In terms of aleatory uncertainty, which refers to inherent variability and unpredictability, we acknowledge its significance. However, we position it as a potential avenue for future work, as outline in Chapter 7, by exploring how explainability can be provided for unexpected events influenced by aleatory uncertainty to reduce surprises. A better understanding and mitigation of the impact of unpredictable external factors on smart contract execution can contribute to the broader goal of creating smart contract systems that are more reliable and transparent.

5.2.4 The Metacognitive Explanation-Based Theory (MEB)

The MEB theory proposes that surprise intensity is closely tied to the metacognitive effort of explaining an event [106]. According to this theory, events that are difficult to explain require more cognitive effort to assimilate into existing mental schemas, resulting in a greater sense of surprise. This connection highlights the role of cognitive processes in shaping our experiences of the unexpected, where the challenge of integrating new information influences the degree of surprise. This theory aligns with the Representation-Fit theory, which emphasises the

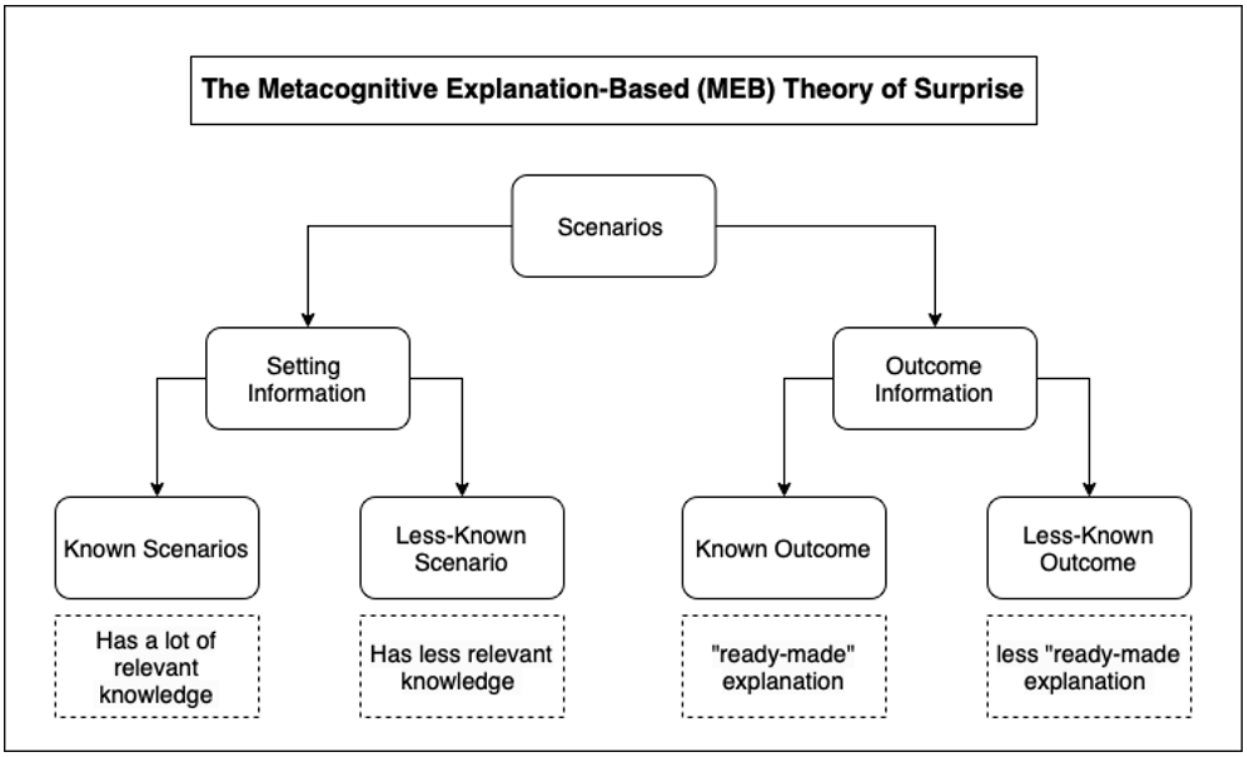


Figure 5.1: Scenarios Classifications Based on the MEB Theory

importance of how well an event fits with an individual's pre-existing mental models [195, 197]. Both theories suggest that the easier it is to integrate an event into these models, the less surprising the event will be. Empirical studies support this view, showing that providing clear and effective explanations can significantly lessen the surprise by facilitating the integration of unexpected events into our cognitive frameworks [195, 197, 196, 103, 105, 104]. These empirical studies demonstrate that explanations significantly mitigate the intensity of surprises and their findings are universally applicable, not confined to any specific system or domain. This focus on explanations aligns directly with our study's interest in how explanations help manage the experience of surprise, distinguishing MEB theory from models that center purely on probability or expectation-disconfirmation factors.

Other perspectives on surprise in cognitive science frequently investigate the dynamic between surprise, probability and expectation. These theories consider various aspects, such

as the low probability of an event, the contrast with more probable outcomes, the disconfirmation of expectations, or the divergence from established mental schemas. They offer insights into how surprise is processed and experienced in the human mind [194, 253, 191]. Many computational models of surprise are grounded in the probability and expectation-disconfirmation perspective. These models aim to replicate surprise in artificial systems by incorporating elements related to the probability of events and the extent to which those events deviate from expected outcomes. In essence, these models seek to capture the computational or quantifiable aspects of surprise within the framework of probability and expectation-disconfirmation [188, 24, 17, 192]. However, since our study emphasises the role of explanations in reducing surprise, MEB theory is more appropriate, as it directly addresses the explanatory process rather than focusing solely on an event's statistical rarity.

Our approach takes a new angle compared to existing surprise models by focusing on the role of explanations when faced with uncertainty. We focus on how explanations play a role when things are unclear, rather than just studying why something surprises us. This structure establishes connections between the end result and the circumstances leading up to it [173, 270]. Importantly, we contend that the resolution of surprise hinges on the construction of an explanation that effectively links the initial situation to the final outcome, ultimately leading to the resolution of the surprise. This emphasis on explanation construction as a means to manage surprise further supports our selection of MEB theory, as it provides a empirical foundation for understanding the importance of explanation in reducing the surprise. Leveraging these insights, our research suggests that crafting effective explanations can reduce surprises by adding clarity and context.

The MEB theory categorises scenarios into (i) setting information and (ii) outcome information. Each scenario commences with setting information, including key actors, pertinent contextual details and unfolding events [106]. When the setting information is either absent or not comprehensible to users, it can significantly contribute to the emergence of

surprising outcomes. The setting information forms the foundation upon which users build their expectations and mental models of how a given scenario should unfold. The theory classifies settings and outcomes into known and less-known scenarios, as seen in Figure 5.1. In addition, the theory presents four predictions that are (i) memory contents are critical in surprise, (ii) scenarios are cues, (iii) partial explanations will reduce surprise and (iv) task demands can affect surprise. Further exploration of the MEB theory and its application to our study is detailed in Section 5.4 and 5.5.

5.3 Explainability Purposes for Smart Contracts

Smart Contracts operate on fixed, immutable and enforceable rules, demanding a robust foundation of trust, particularly in decentralised contexts. Our strategy for formulating tailored explanation purposes for smart contracts involves a twofold investigation. In the first step, we evaluate the essential characteristics of legally binding contracts to determine their enforceability and align them with the distinct characteristics of smart contracts. As a second step, we conduct a comprehensive overview of XAI's existing practices as AI complements contract characteristics in automated decision-making processes.

We introduce the concept of 'explainability purposes' as vital components in developing explainable smart contracts. We outline four distinct explanation purposes — Explain to clarify, justify, ensure compliance and facilitate consent. These explanation purposes are fundamental in ensuring that smart contracts are comprehensible and legally compliant. We argue that these purposes are essential for addressing different scenarios where certain intelligent system behaviours may be incomprehensible, undesirable, or unexpected for users [124]. The defined purposes form the basis for benchmarking scenarios relevant to research on explainable smart contracts as follows:

Explain to Justify: This purpose is about justifying the decisions, actions and outcomes of smart contracts. By doing so, stakeholders can gain a deeper understanding of the underlying logic that drives the contract's behaviour, ensuring that every action is justified and supported by solid reasoning. This purpose involves providing explanations at various levels within the system, covering both high-level objectives and detailed explanations of specific elements in decision-making.

Explain to Clarify: Clarification is necessary in order to illuminate the complexities of smart contracts. Specifically, it aims to provide a clear understanding among stakeholders, especially when details concerning contract execution, including off-chain processes, role-based behaviour and risk management functions, are not immediately apparent. Therefore, this purpose addresses various aspects of smart contracts, enhancing user awareness of processes that are not transparent.

Explain for Compliance: The purpose of compliance is to explain how smart contracts adhere to established legal norms and regulations. This function ensures the legality of contract operations, providing users with a clear understanding of how the contract conforms to established legal requirements. Additionally, this explanation enables users to navigate the legal framework surrounding the contract, ensuring compliance with all applicable laws.

Explain for Consent (Offer & Acceptance): The consent process aims to provide users with transparent explanations of the smart contract's services and terms before its execution. This ensures that all parties understand and consent to the contract terms, following contract law's fundamental principles of offer and acceptance. It also secures the consent of the users when necessary, such as when using personal information.

It is essential to recognise that while these purposes are significant, they do not

represent an exhaustive list. Other purposes, such as managing, validating, evaluating or learning, could enhance smart contracts further. We acknowledge the existence of potential avenues for exploration in designing contracts that prioritise human comprehension and usability. However, in the context of this study, our emphasis has been on validating the proposed four purposes.

5.3.1 Development of Explanation Purposes

This section briefly discusses the development of the proposed explanation purposes. We established the aforementioned purposes by synthesising perspectives [244] from contract law principles and the field of XAI to craft purposes explicitly tailored for smart contracts. Table 5.1 and Table 5.3 present the summaries of the synthesis analysis.

Enforceable Traditional Contracts

Contract law serves as the foundation of legal systems, establishing the framework for the creation, interpretation and enforcement of contracts. Traditional contracts rely on key elements such as offer and acceptance, consideration, legality and capacity to validate and enforce their terms [97, 202]. As smart contracts emerge at the intersection of technology and law, it becomes imperative to align these innovative digital agreements with established contract principles [64] [176].

As we step into the domain of smart contracts, understanding the four key elements of a legally binding contract is fundamental. Table 5.1 summarises the alignment. First, offer and acceptance involve a proposal by one party and agreement by the other (consent), solidifying a binding contract [158]. Consideration, the second element, involves an exchange of value, signalling intent to establish a legally recognised relationship [34]. Legality dictates

Table 5.1: Alignment of Binding Contract Elements with Smart Contract (SC) Purposes

Binding Element	**Description**	**SC Purposes**
Offer & Acceptance	Mutual agreement between parties via offer and acceptance.	Consent
Consideration	Something of value (e.g., money, goods, services) exchanged between parties forming legal obligations.	SC holds value
Legality	Terms must not violate laws or public policy. Contracts with illegal purposes are unenforceable.	Compliance
Capacity	Parties must have legal capacity and sound mind to understand and agree to contract terms.	Clarification

that contracts must comply with the law, and contracts violating legal norms are void [97]. Capacity refers to individuals' mental competence to engage in a contract; parties must understand and fulfil their obligations. If a party is deemed incapable or presumed incapable of comprehending the agreement, they lack the requisite capacity to partake in a legally binding contract [76]. While these principles are universally applicable, specific EU directives and regulations can significantly impact certain contracts and industries [170, 57]. Building upon these elements, we have formulated specific explanation purposes: *Explain to facilitate consent*, *explain to ensure compliance* and *explain to clarify*. Smart contracts inherently satisfy the consideration element, holding value, services, or goods in exchange for fulfilling specified conditions.

To operationalise these explanation purposes: *Explain for compliance* ensures smart contracts adhere to established legal norms and regulations, aligning with the legality of binding contracts. *Explain for consent* provides detailed information on contract terms and

execution, allowing users to provide and withdraw their consent following the General Data Protection Regulation (GDPR) [251] and facilitating the offer and acceptance component for mutual agreement. The *Clarification* explanations address the need for users to comprehend coded contracts. This purpose aims to enhance user understanding by providing clear insights into the logic, terms and processes that govern smart contract operations. The purpose of clarification also shares similarities with explainability features in AI, focusing on understanding and gaining insights into decision-making processes.

Established Explainability Goals in XAI

The origins of explainability goals or purposes stem from the inherent need for transparency in AI [18]. To discover these purposes, we conducted a comprehensive literature review to investigate and gain insight into the prevalent purposes and goals within the field of XAI. Our review focuses specifically on surveys, reviews and SLRs that address the requirements of explainability and thoroughly discuss XAI's purposes.

To identify relevant studies, we searched popular databases such as IEEE Xplore, ACM Digital Library, ScienceDirect and Google Scholar. Our search strategy initially targeted studies covering the concept of explainability in AI using terms such as "XAI," "explainable AI," and "explainability." We then refined our search by incorporating specific terms such as purposes, goals, needs, drivers, or reasons for explainability. A clear inclusion criteria guided the selection process. We sought studies that explicitly stated the purposes or goals of explainability requirements in AI. Studies discussing methods and approaches for XAI were excluded to maintain our focus specifically on explainability requirements. Furthermore, we prioritised studies based on their quality and citation impact. We conducted a quality assessment following established guidelines for evaluating studies based on rationality, rigour, credibility and contribution [324]. Table 5.2 presents our final selection,

Table 5.2: Summary of Selected Studies on Explainability Purposes in AI

No	Study	Title	Year	Venue	Citation
1	[2]	Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence	2018	IEEE Access	4802
2	[18]	Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities & challenges toward responsible AI	2020	Information Fusion	6461
3	[37]	A Review of Trustworthy & Explainable Artificial Intelligence (XAI)	2023	IEEE Access	32
4	[67]	Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey	2022	Information Sciences	60
5	[75]	Explainable AI (XAI): Core Ideas, Techniques, & Solutions	2023	ACM Computing Surveys	173
6	[132]	Interpreting Black-Box Models: A Review On Explainable Artificial Intelligence	2024	Cognitive Computation	72
7	[205]	Explainable Artificial Intelligence: Objectives, Stakeholders, & Future Research Opportunities	2022	Information Systems Management	308
8	[201]	The Role of Explainability in Creating Trustworthy Artificial Intelligence for Health Care	2021	Journal of Biomedical Informatics	451
9	[248]	Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges, & Perspectives	2022	IEEE Transactions on Artificial Intelligence	104
10	[256]	Explainability in Human–Agent Systems	2019	Autonomous Agents & Multi-Agent Systems	270
11	[304]	Notions of explainability & evaluation approaches for explainable artificial intelligence	2021	Information Fusion	334

comprising 11 studies that offer a representative coverage of established concepts related to the purposes and goals of explainability within the literature on XAI.

Table 5.3 outlines the common goals of XAI as identified through our synthesis analysis, highlighting causality and knowledge discovery as the most common purposes. While knowledge discovery in AI enables systems to analyse and learn from data, enhancing decision-making capabilities autonomously, smart contracts are fundamentally different. They strictly execute predefined rules encoded in their scripts without learning or adaptation capabilities. Given these differences, we identify a shared trace in automated decision reasoning, leading us to propose two fundamental purposes: *Explain to justify* and *clarify*, which align with the deterministic nature of smart contracts.

The *explain to justify* purpose is particularly relevant in smart contracts when users seek assurance and reasoning behind smart contract outcomes. It provides clear justifications for the decisions made within the ecosystem. Further insights into justification explanations can be found in [27]. Additionally, studies such as [18, 205] extensively explore motivations for explainable AI models, highlighting 'Causality'/'Justification' as a fundamental purpose in the context of automated decision-making.

Finally, *explain to clarify* aligns with the objectives of providing users with clear and concise informative explanations regarding the contract's terms, conditions, processes and functionalities. The purpose of clarification resonates with principles of informativeness, ethical transparency and interactivity in AI, as it aids in ensuring that users fully understand the implications and operations of smart contracts. Similar to the findings in [259], which highlight clarity as a pivotal factor influencing user comprehension and trust in automated systems, a parallel principle can be applied to the domain of smart contracts. The need for clarification arises when system behaviours may be incomprehensible or unexpected, requiring explanations that instruct and convince users for effective human-system interaction

Table 5.3: An Overview of XAI Goals Across Selected Studies

XAI Goal/Purpose	**Studies**										
	[2]	[18]	[37]	[67]	[75]	[132]	[205]	[201]	[248]	[256]	[304]
Trustworthiness		✓	✓								
Causality	✓	✓		✓	✓	✓	✓		✓	✓	✓
Transferability		✓							✓		
Informativeness		✓		✓	✓		✓		✓	✓	
Confidence/Trust		✓	✓		✓				✓	✓	
Fairness/Ethics		✓	✓		✓	✓		✓		✓	
Accessibility		✓									
Interactivity	✓	✓	✓	✓				✓	✓		
Privacy Awareness		✓	✓						✓		
Knowledge Discovery	✓			✓	✓	✓	✓	✓		✓	✓
Validation	✓						✓	✓		✓	
Debugging					✓		✓	✓		✓	✓
Legality			✓			✓	✓	✓		✓	
Improvement	✓			✓		✓	✓	✓			✓

[124, 133].

5.3.2 Scenario-Based Design for Explainability Purposes

Scenario-based design approaches can be found in many fields such as user experience design, software development and systems engineering. Scenarios are detailed narratives or stories describing how users interact with a product, system, or service in specific situations [257]. The primary goal is to understand user behaviour, needs and goals in a context.

Scenario-based design differs from the solution-first approach. In solution-first design, a technical solution is proposed upfront and subsequent evaluations aim to understand the problem domain better. This technical solution is often seen in smart contract contexts, where specific technical solutions are introduced for real-world requirements. However, solution-first design has drawbacks, including prematurely committing to a solution, oversimplifying the problem and hesitating to change initial solutions [257, 33]. Scenarios help prevent premature commitments, avoid oversimplification and foster innovation that aligns with the complexities of the real world.

We advocate for a fundamental change in the approach to smart contract design, transitioning from a technology-centric or solution-oriented approach to embracing a scenario-based perspective. This shift can bring new perspectives to the design of explainable smart contracts. Rather than solely focusing on what a smart contract system can explain, prioritising scenarios encourages us to consider the kinds of explanations users might require during their interactions with smart contract systems.

As previously mentioned, our inspiration comes from the MEB theory of surprise [106]. The primary aim of explanation in this context is to alleviate surprises that users might encounter while interacting with smart contracts. According to the MEB theory, each

scenario starts with setting information, comprising key actors, relevant contextual details and unfolding events. The MEB scenario elements resonate with scenario-based design [257], where the four core elements are: (1) actors, (2) background information and assumptions about the environment, (3) goals or objectives of actors and (4) sequences of actions and events. To demonstrate the scenario-based design approach for explainability purposes, we provide examples that illustrate how these principles can transform the design of smart contracts into a more human-centred approach.

Scenario 1: Compliance with New Regulatory Requirements

- *Context:* Eve operates a DAO that manages funds. New regulatory requirements are introduced, impacting how DAOs should handle fund management. The smart contract governing Eve's DAO must adapt to these changes to remain compliant.
- *Compliance Purpose Design:* Assess the smart contract's responsiveness to changes in regulatory requirements. The smart contract's design should demonstrate the capability to uphold compliance through upgradable logic, ensuring adherence to updated regulations and its proficiency in providing clear explanations for user understanding.

Scenario 2: Consent for Terms Update in a Loan Smart Contract

- *Context:* Charlie has an active loan through a decentralised lending platform. The lending DApp introduces updated terms due to changes in regulations. Charlie is prompted to consent to the updated terms before continuing with the loan agreement.
- *Consent Purpose Design:* Evaluate the smart contract's handling of user consent for changes in contractual terms and assess the transparency of the consent process. The

smart contract is expected to facilitate the consent process, providing a straightforward mechanism for Charlie to express his consent or dissent.

Scenario 3: Justification for Access Control

- *Context:* Bob attempts to access a DApp that requires authentication based on specific criteria. The smart contract denies access and Bob requests justification for the restriction.
- *Justification Purpose Design:* Evaluate the effectiveness of the smart contract in providing clear justifications and communicating the underlying reasons to the user. The smart contract's design should carefully incorporate the rationale behind decisions, acknowledging the impact on its users.

Scenario 4: Clarification of Value Determination

- *Context:* Alice, an investor in a decentralised token ownership platform, seeks clarification on how the value of her tokens is determined. The valuation process involves dynamic elements influenced by market conditions, external data feeds and internal algorithms.
- *Clarification Purpose Design:* Assess the extent to which the smart contract elaborates on the factors influencing token valuation, including market conditions, algorithms and data sources. The smart contract should offer an explanation that is accessible and understandable for users with diverse levels of technical knowledge. It should facilitate real-time values on factors influencing token value, allowing Alice to stay informed about valuation changes.

5.4 The MEB Evaluation of Surprise

This section introduces a novel evaluation method for assessing explainability purposes, drawing from the MEB theory of surprise [106]. In our perspective, surprise acts as a mechanism directing attention to conflicting information in the environment, signalling the need for an explanation to reconcile the discrepancy. The computation of the MEB theory requires an explanation that connects the setting and outcome of the scenario, ensuring coherence and sense-making. This process evaluates the congruence between the setting and outcome information, providing a direct measure of alignment based on available contextual information.

In developing our evaluation method, we incorporate two predictions derived from the theory: *Scenarios are cues* and *partial explanations will reduce surprises*. The first prediction categorises the setting and outcome information into 'known' and 'less-known' as presented in Figure 5.1. A scenario that effectively activates a broad body of relevant knowledge tends to result in a lower level of surprise. Conversely, if the scenario fails to activate pertinent knowledge, it may lead to a higher surprise. Therefore, as per the theory, even if memory holds a reservoir of potentially relevant knowledge, inadequate triggering of this knowledge in the scenario can make comprehending the event challenging or unsuccessful, resulting in surprises.

The second prediction, *partial explanations will reduce surprises*, suggests that individuals can better comprehend the unfolding event by providing additional key information within the extended setting. It becomes evident that the provision of information plays a pivotal role in mitigating the level of surprise. To summarise the insights from the theory:

- The setting information establishes the context for what is occurring. It identifies the main actors, relevant background knowledge and unfolding events. If the setting

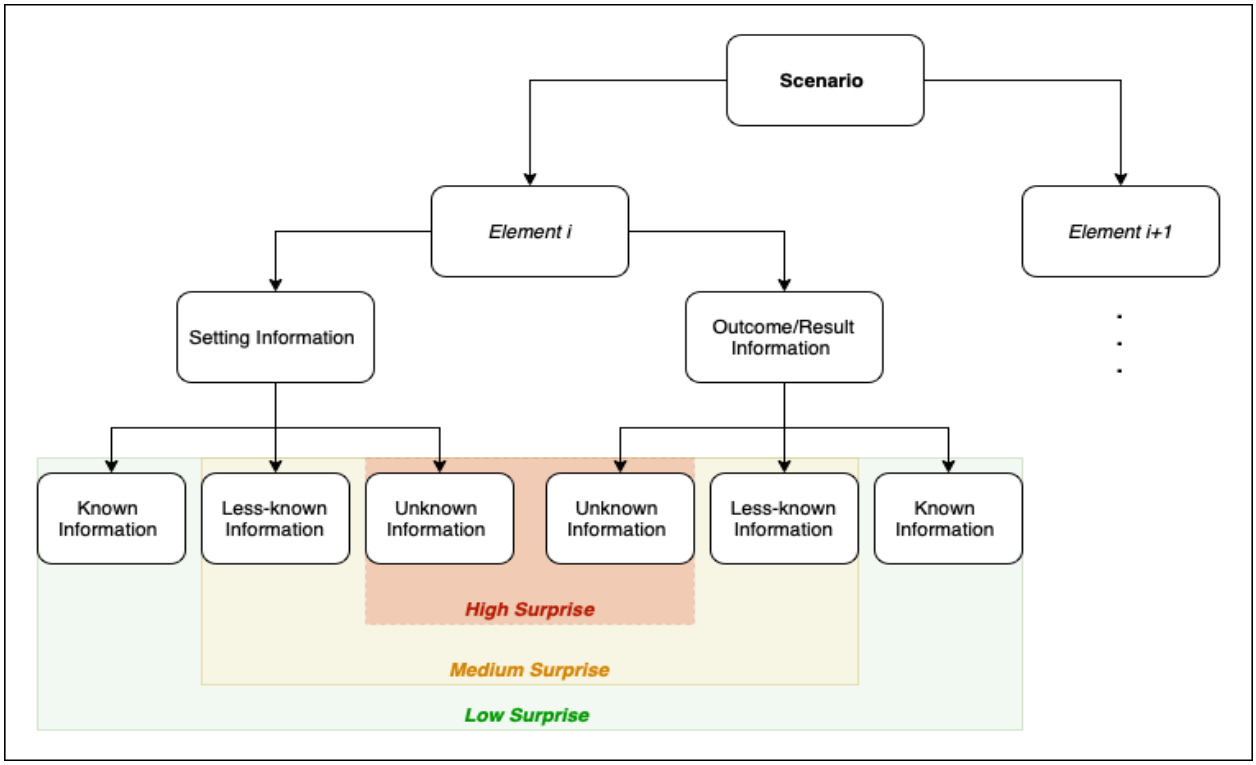


Figure 5.2: Breakdown of the Surprise Evaluation Method

information is less-known or unknown to the individual, constructing a coherent explanation for the surprising outcome becomes challenging. This increased cognitive effort to build an explanation contributes to the sense of surprise. The more difficult it is to bridge the gap between the unexplained outcome and the less-known setting, the higher the perceived surprise.

- An unexplained outcome refers to an event or outcome that is unexpected or not in line with the individual's prior knowledge or expectations. When faced with an unexplained outcome, individuals attempt to explain why it happened by drawing on their existing knowledge and trying to connect the outcome to the provided setting information.

These insights have guided the development of our evaluation method. In the MEB theory of surprise, the concepts of setting and outcome information play central roles [106].

Each scenario is explicitly segmented into (i) setting information and (ii) outcome information:

Setting Information is the information that establishes the context for what is occurring within a scenario. The setting information in the MEB theory is classified into 'known' and 'less-known' scenarios. The classification reflects the degree of available information and how well it cues relevant knowledge for explanation.

Outcome Information is the event or result that concludes or characterises the scenario. The outcome information is also classified into 'known' and 'less-known' categories. Known outcomes are those that align with the user's expectations or prior knowledge, leading to less surprise. In contrast, less-known outcomes are those that users do not expect, often resulting in high levels of surprise.

In our evaluation method, each element in a scenario is segmented into (i) setting information and (ii) outcome information. We then classify setting and outcome information into 'known,' 'less-known,' or 'unknown,' based on the degree of information availability, as illustrated in Figure 5.2. The distinction between 'less-known' and 'unknown' is defined by the level of available information: 'unknown' signifies a complete absence of information. In contrast, 'less-known' indicates the information is unclear or partially missing. To streamline the evaluation process, we apply the MEB theory's definition of surprise to categorise scenarios into known, less-known, or unknown. We qualitatively assess each scenario's potential degree of surprise using the matrix presented in Table 5.4.

This matrix makes it possible to assess the degree of surprise in different scenarios based on the known and less-known factors in both the setting and the outcome. This interpretation of the categorisation of the scenarios helps in understanding the likelihood and intensity of surprises that users may experience. Each scenario is analysed within this framework to determine its position on a spectrum from 'known' to 'unknown':

Table 5.4: Potential Degree of Surprise Qualitative Matrix

	Known Outcome	**Less-Known Outcome**	**Unknown Outcome**
Known Setting	Very Low Surprise	Low Surprise	Medium Surprise
Less-Known Setting	Low Surprise	Medium Surprise	High Surprise
Unknown Setting	Medium Surprise	High Surprise	Very High Surprise

- **Known Scenarios (Low Surprise):** Scenarios categorised as known are those with sufficient information about the setting and the outcomes. The surprise level in such cases is low because adequate resources and information are available to explain the actions and outcomes. The matrix presents these in green, indicating low surprise potential. **Example:** Consider a smart contract for rental agreements that clearly outlines rent amounts, due dates and penalties for late payments before execution, constituting the setting information. The outcome information includes details communicated after the contract is executed. For instance, if a payment is delayed, the contract communicates the specified penalty and explains the reasons for this enforcement action. In this case, the contract's implementation aligns with the setting information provided to all parties before execution. There are no hidden processes or missing details; the outcomes contain all the necessary information to explain its actions.

- **Less-Known Scenarios (Moderate Surprise):** Essential information may be partially available in these situations, leading to moderate surprise. Users have some visibility into the process or outcomes but lack complete information, creating a gap in understanding. Such scenarios often result in the need for more clarity and accessibility of information. These scenarios are marked in yellow on the matrix, such as the intersection of a less-known setting and a less-known outcome. **Example:** Consider

a smart contract for a variable interest rate loan that indicates rates may vary with market conditions. However, it lacks specifics in the setting information about the frequency or exact triggers for these adjustments. When a borrower encounters an unexpected rate increase, the contract fails to provide transparent outcome information or explain the reasons behind the adjustments. This scenario can lead to moderate surprise, as the borrower is aware that rates can change but is not informed about how and when these changes occur nor the underlying reasons for the adjustments. Here, the contract lacks an informative setting and its outcome lacks reasoning.

- **Unknown Scenario (High Surprise):** There is a high degree of surprise associated with scenarios with little or no relevant information regarding the setting or the outcome. **Example:** A smart contract utilising price oracles for real-time token pricing, users are informed that pricing is automated based on reliable data feeds. However, unbeknownst to users, the contract includes a hidden process that allows project owners to adjust prices manually. This manual intervention is neither disclosed in the setting information nor recorded on the blockchain (outcome). During periods of market fluctuation, owners may use this feature to alter prices, which could deviate significantly from actual market rates. Such undisclosed setting information and their unexplained outcomes lead to a high level of surprise, highlighting the urgent need for enhanced transparency to maintain user trust. The contract's setting information does not accurately reflect the implementation of the smart contracts.

The matrix in Table 5.4 provides a qualitative assessment of the potential degree of surprise. For this study, we convert these categories into quantitative assessments by assigning numerical values. The surprise value is assumed to lie between the intervals [0,1], where 0 indicates that no or minimal surprises may occur and 1 indicates that a very high level of surprises may occur. We assume that the potential degree of surprise can be quantified by adding the setting score to the outcome score. Table 5.5 presents a matrix of the average

values for the potential degree of surprises.

Table 5.5: Potential Degree of Surprise Quantitative Matrix

	Known Setting(0)	**Less-Known Setting(0.25)**	**Unknown Setting(0.5)**
Known Outcome(0)	0	0.25	0.5
Less-Known Outcome(0.25)	0.25	0.5	0.75
Unknown Outcome(0.5)	0.5	0.75	1

When the setting and outcome are known, the potential surprise value is zero, indicating minimal surprise potential. Conversely, when the setting is less-known (0.25) and the outcome is unknown (0.5), such as the intersection of the “Less-Known (Setting)" row with the “Unknown (Outcome)" column, the average surprise value is 0.75, indicating a higher potential of surprises. To illustrate the scoring process: In the unknown scenario example, where there is partial information about data feeds (setting score: 0.25) and the execution of a hidden process (outcome score: 0.5), the total score is 0.75, indicating a potential high level of surprise. The following section provides detailed steps for evaluating surprise potential.

5.4.1 The MEB Framework Steps

We develop a systematic and generic set of steps to evaluate the potential surprises arising from epistemic uncertainties within any scenario. The outcome of the evaluation is to classify scenarios into known, less-known and unknown, as explained in the Section 3.4. This structured approach allows for assessing multiple elements, providing a versatile evaluation framework applicable across various contexts.

1. **Develop Scenarios for Explanation Purposes** : The initial step involves crafting hypothetical scenarios to evaluate the provision of information and explanations during user interactions. Specifically, in the context of our study, we develop scenarios tailored to smart contracts to assess their explanatory capabilities, as presented in Section 5.3.2. These scenarios are designed to include various aspects of smart contract interactions such as decision-making processes and outcomes. This process includes identifying key actors, understanding the context, outlining expected outcomes and specifying sources of both setting and outcome information to evaluate explanations provision.

2. **Defining Evaluation Criteria:** We establish our evaluation criteria tailored explicitly to explanation purposes, including clarification, justification, consent and compliance. For example, within the context of *justification*, the absence of a clear rationale explanation yields a high impact on generating surprises, denoted by a score of 'unknown' (0.5). Conversely, when a clear and comprehensive rationale is provided, it registers a low level of impact and is assigned a 'known' score (0). This step is flexible and can be customised to define various criteria depending on the evaluation objectives.

3. **Define Scenario Elements and their Importance Weights:** In this step, we identify key elements within scenarios that require assessment for information provision within the system. Each scenario comprises multiple elements or components in a system that reflect the targeted interaction under assessment. These elements range from high-level components such as business terms, policies, or legal compliance requirements to more detailed aspects such as decision logic, data processes and access control conditions. Each element is assigned an importance weight on a scale from 0 to 1, indicating its relative impact on the potential for surprise. Elements with higher weights are deemed more critical; their absence or inadequacy in the system's information provision can significantly increase the likelihood of surprise. Conversely, elements with a weight of 0 are considered to have minimal impact on the overall

surprise potential.

The weight assigned to each element reflects its importance in conveying decision-related knowledge within the system. Certain elements carry a higher weight because they are essential for users to understand factors that directly impact their decisions. For example, specific conditions or criteria affecting user outcomes are given more weight, as missing information on these elements could lead to unexpected surprises for the user. In contrast, elements with a lower weight are less critical, as their presence or absence does not significantly alter the user's overall understanding of the decision.

4. **Calculate Potential Degree of Surprise (DoS)**: We evaluate each element by scoring its setting (S) and outcome (O) information according to the matrix criteria: known (0), less-known (0.25), or unknown (0.5). If the scenario comprises only one element, then the DoS is calculated by simply adding the scores of the setting and the outcome. The result of this addition is interpreted by the matrix in Table 5.5. However, the assigned importance weights are utilised if the scenario contains multiple elements. For each element, multiply the assigned weight by the sum of the corresponding setting and outcome scores. This calculation yields the potential degree of surprise (DoS) using the equation:

$$DoS_e = Weight_e \times (S_{score} + O_{score})$$

5. **Aggregate Element Scores:** We aggregate the Degree of Surprise (DoS) scores to determine the overall surprise potential for the entire scenario. This aggregation involves summing up the weighted surprise scores of all evaluated elements in a scenario.

$$Surprise_{Aggr} = \sum_{i=1}^{n} DoS_e$$

Where DoS_e represents the Degree of Surprise for each element and n is the total

number of elements in the scenario. This aggregation provides a quantitative measure that reflects the overall potential for surprise based on the combined impact of all elements involved.

6. **Normalise the Aggregate Score**: After aggregating the DoS scores for all elements in a scenario, it becomes essential to normalise this aggregate score to ensure that the final surprise value is interpretable within the defined surprise matrix, typically ranging from 0 to 1. This value can be achieved by dividing the aggregator score by the number of evaluated elements n.

$$Surprise_{norm} = \frac{Surprise_{Aggr}}{n}$$

7. **Interpretation of the Potential DoS**

   Once the Degree of Surprise (DoS) for each scenario has been normalised, it is important to interpret these scores to understand their potential surprise level. The normalised scores are categorised into distinct ranges, each representing a different level of potential surprise: a score from [0, 0.2] indicates a shallow potential for surprise; a score from [0.21, 0.4] indicates a low potential for surprise; a score from [0.41, 0.6] indicates a medium potential for surprise; a score from [0.61, 0.8] indicates a high potential for surprise; and a score from [0.81, 1] indicates a very high potential for surprises.

Given the subjective nature of assessing information understanding and explanation, we propose a peer coding process [20, 180], involving at least two evaluators to assess systems independently. This collaborative approach leverages diverse perspectives, mitigating individual subjectivity and enhancing the reliability of the assessment. When multiple evaluators are involved, measuring the level of agreement between them is essential. Cohen's Kappa, a well-established method, quantifies the extent of agreement between evaluators

[285]. A Cohen's Kappa rate exceeding 0.6 signals a strong consensus among evaluators, indicating an acceptable assessment. In instances of a low Kappa rate, a secondary review is initiated. This process involves a detailed discussion of the reasons for discrepancies, followed by a subsequent evaluation. The engagement of multiple evaluators and the iterative review process ensure a thorough and consistent interpretation of surprise potential in the assessment.

5.5 Application of the Evaluation Method and Explainability Purposes

This section unfolds in several stages. Initially, we evaluate the provision of explanations within two projects, considering our explanation purposes. Secondly, we formulate a strategy to implement explanation purposes in areas marked as high priority. Lastly, we conduct a cost analysis of integrating explanations into smart contracts.

We examine two lending DApps due to their substantial impact on users' financial status. Additionally, these applications incorporate fundamental elements necessary for their operations, including decision logic comprising conditions and rules, reliance on external data and dependencies for asset values and human involvement, where specific authorities have the privileges to set and modify interest rates. The use case also incorporates additional dimensions for consent and compliance, providing a robust foundation for application.

We utilise the Alchemy website [1], a comprehensive web3 development platform and DApps explorer that showcases over 1000 DApps across popular public blockchains such as Ethereum [89] and Solana [279]. We selected the final two from the top 10 lending DApps

[1] https://www.alchemy.com/best/decentralized-lending-dapps

listed based on their configuration and successful installation in a local environment. At this point, it is essential to clarify that our aim is not to make definitive judgments regarding the trustworthiness or quality of the chosen projects—instead, our focus centres on a thorough examination and comprehension of current industry practices. The decision to omit the projects' names underscores our commitment to providing an impartial analysis without implying any specific assessment of their overall merit or reliability.

To commence the evaluation (step 1), we define specific scenarios covering the three dimensions of decision-making: decision logic, external data and human involvement in both projects. These scenarios serve as the contexts for evaluating and implementing the explanation purposes.

- **Scenario 1 (Lending Decision):** Bob, a non-technical user of a decentralised lending platform, finds himself confused by the platform's borrowing decisions. Seeking clarity, he searches for information to comprehend the rationale behind the values used in the borrowing process. He is also interested in understanding how the platform aligns with compliance standards and ensures that the DApp has provisions for obtaining his consent in the lending process.
- **Scenario 2 (External Resources):** Emily, an experienced blockchain enthusiast, is actively engaging with a lending platform to borrow cryptocurrency. She is keen on ensuring the precision of her interest rate calculation. With her background in blockchain technology, she decided to examine information about the external data sources and input values used in the calculation to guarantee accuracy and transparency in the process.
- **Scenario 3 (Roles and Responsibilities):** Sarah, an active user of a decentralised lending platform, relies on smart contracts to manage her digital assets. While using the platform, she noticed that certain authorities or administrators have control over

critical functionalities. She wants to gain clarity on the roles and responsibilities of each authority and searches for available information on the website. Additionally, this discovery prompts Sarah to reflect on the confidentiality of her personal information.

After establishing scenarios, step 2 involves defining the evaluation criteria, which in our case include justification, clarification, compliance and consent. Step 3 includes defining scenario elements and their respective weights. We have defined a set of fixed elements for each scenario, as outlined in Figure 5.3. This approach ensures a systematic and unbiased assessment and guides the design of explanation purposes.

The elements in Figure 5.3 are identified by the scenarios given and supported by the standards of ACM responsible algorithmic systems [58], considering essential qualities such as understandability, transparency, accountability, interpretability and explainability. For example, transparency is defined as *"System developers are encouraged to clearly document the way in which specific datasets, variables and models were selected for development, training, validation and testing, as well as the specific measures that were used to guarantee data and output quality."* Elements such as external data sources, links and aggregation methods are defined accordingly. For accountability and responsibility, we define elements to understand the role of humans in smart contract decisions and operations, such as the stakeholders' roles and responsibilities involved in operating the smart contract and permission hierarchy. Furthermore, interpretability and explainability are highlighted: *"Managers of algorithmic systems are encouraged to produce information regarding both the procedures that the employed algorithms follow (interpretability) and the specific decisions that they make (explainability)."* We define elements in justification to help users understand decisions, justify changes in authorities and clarify values used in decisions, such as interest rates. Finally, understandability emphasises the software's ability to assist users in comprehending its suitability and policies of use [41]. Consequently, we define elements related to compliance,

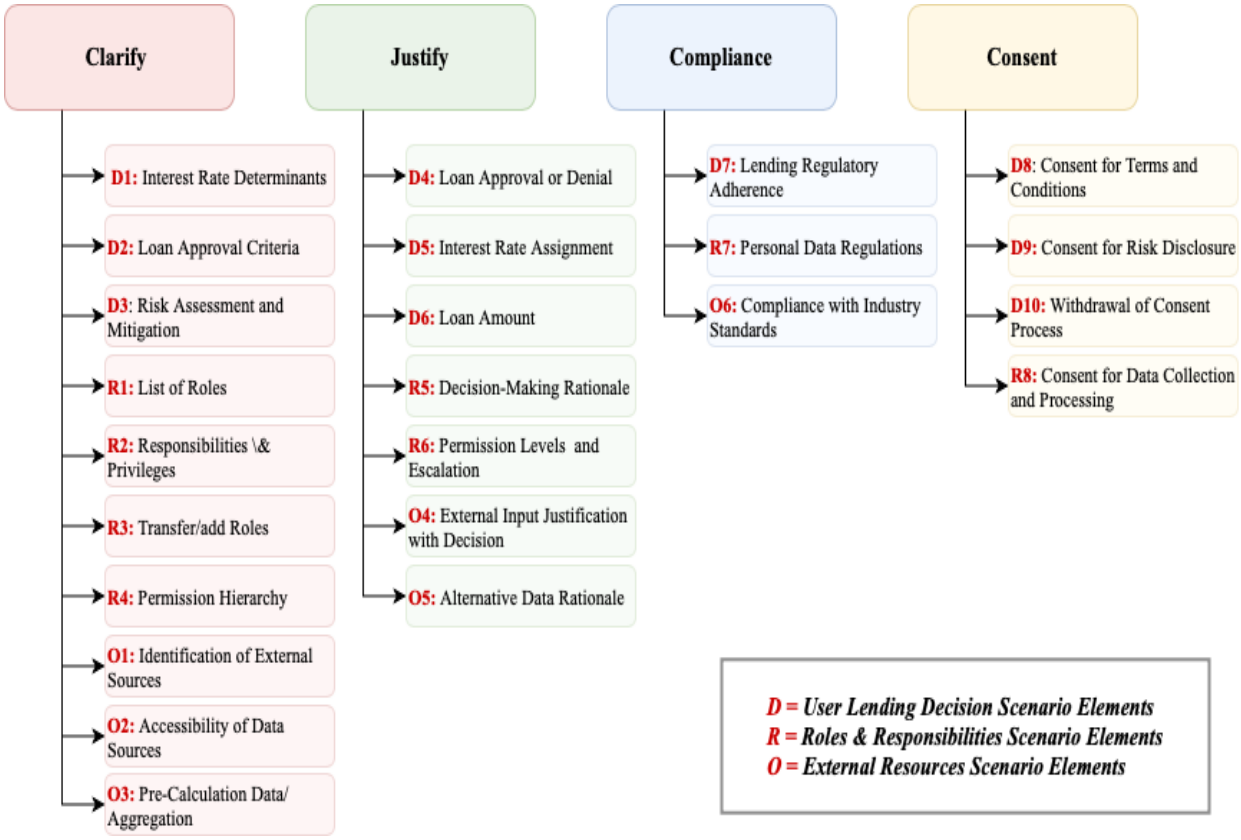


Figure 5.3: Fixed Elements for Explanation Evaluation and Implementation

consent and understanding of conditions and risks.

Then, we identify sources for setting and outcome information for evaluators to ensure consisting assessment. The information of a project setting project include front-end interactions, documentation and websites. While some projects use code comments to explain smart contracts, we exclude this method due to its potential limitations in providing comprehensive user understanding. For outcome sources, we rely on transaction data, event logs and code implementation. We provide evaluators with a generic template, including scenarios, element definitions, their weights and evaluation matrices based on the MEB theory as outlined in Section 5.4.

Two researchers independently conduct the evaluation process, each with years of

expertise in research on smart contracts and blockchain technology. For this evaluation, we do not seek precise agreement on the reviewers' results. Instead, our focus is on achieving consensus regarding the degree of surprises based on the information provided, categorised as high, medium and low.

The level of agreement was measured using Cohen's Kappa [285], with an initial agreement rate of approximately 0.52, indicating moderate agreement. Notably, divergences occurred in two scenarios (2 and 3) due to differences in the setting information each researcher referenced. We investigated these differences, focusing on the sources each researcher used. For instance, Researcher 2 did not include the project's white paper, which outlined role explanations for Scenario 3, while Researcher 1 omitted developer documentation that clarified external data for Scenario 2. These omissions led to further discrepancies between reviewers, resulting in varied ratings such as [low, high]. After addressing these conflicts in source information, researchers conducted a second evaluation round.

Our method focuses on evaluating the presence of explanation and justification; thus, results such as [low, high] reflect the range of agreement or divergence in the perceived sufficiency of explanations, as discussed in the previous paragraph. Factors such as incomplete documentation, diverse data sources and the layered structure of DApps can contribute to [low, high] variations in evaluation results. Therefore, this analysis demonstrated that providing explanations within DApps is not straightforward. Various sources of information highlighted the layered nature of information provided to users, indicating a need for standardised settings and documentation in DApps.

To support transparency, the complete application, including the generic template, element definitions, weights, evaluation matrices and results, is available in a public repository[2]. Appendix D provides a snapshot of the evaluation process.

[2]https://github.com/halghanmi/ExplainableSC/tree/Explainability-Purposes-and-Surprises-Evaluation

Table 5.6: An Overview of Potential Surprises

Scenarios	**P1**		**P2**	
	Normalised DoS	Potential DoS	Normalised DoS	Potential DoS
Scenario 1 (User Decision)	[0.41, 0.46]	**Medium**	[0.33, 0.37]	**Low**
Scenario 2 (External Resources)	[0.65, 0.68]	**High**	[0.34,0.37]	**Low**
Scenario 3 (Roles and Responsibilities)	[0.56, 0.59]	**Medium**	[0.70, 0.73]	**High**

5.5.1 Evaluation Results

Overall, Project 1 (P1) displays a higher potential for surprises compared to Project 2 (P2), particularly in Scenarios 1 and 2. Conversely, P2 demonstrates a high potential for surprises in Scenario 3, while P1 exhibits a medium degree of surprise, as illustrated in Table 5.6. P2 exhibits lower potential levels of surprises, which is attributed to its comprehensive provision of setting information. It offers detailed explanations for numerous elements assessed. Moreover, the outcomes are more comprehensive than those of P1, owing to the recorded events providing additional insight into smart contracts' operations. This finding highlights the existing explanatory practices in P2 that are absent in P1. We summarise our findings as follows:

User Decision

The two projects demonstrate varying approaches to explaining decisions to users. P2 excels in providing clear and well-justified explanations for the decisions made by smart contracts, resulting in a low degree of surprises in terms of clarification and justification. P2 provides comprehensive setting information that specifies values, calculations and justifications. As a result, no significant surprises regarding user decisions were identified in P2. In contrast,

P1 lacks detailed settings and clarification for most elements related to user decisions. For instance, while it mentions that the admin sets the interest rate and liquidity sensitivity, the specific values are not provided. Additionally, the transaction did not provide the values used in the decision.

Roles and Responsibilities

Both projects exhibit a lack of clarity regarding the roles and responsibilities of stakeholders. P2 is notably affected by the absence of information about who controls the contracts and their respective responsibilities. Although the code implies the involvement of various parties in critical aspects of contract decisions, the setting lacks a clear list of these roles, along with their associated responsibilities and privileges. In this regard, P2 faces challenges in clarity compared to P1. While P1 setting information acknowledges different roles, it needs to be more consistent. The setting mentions the admin and operator roles without providing a list of responsibilities. Both projects allow owners/managers to set values in lending decisions, but no justifications are provided when these values change, directly impacting user decisions.

External Data and Dependencies

P1 lacks transparency in data feeds and dependencies. Despite mentioning the Chainlink oracle feed, the project's implementation uses on-chain oracle smart contracts without specifying their addresses. Additionally, P1 manually set token prices without justification as a backup method, which raises ethical concerns. In contrast, P2 identifies all external data and dependencies, resulting in fewer surprises. The outcome includes clarification through event logs. However, P2 lacks information on the project-customised backup oracles. Further explanation is needed to justify the backup oracle and clarify the exact token price used in calculations.

Compliance

Both projects, especially P2, need significant improvements in compliance. There is no indication of adherence to regulatory standards or relevant regulations and laws. P2 demonstrates compliance only in terms of data reliability and industry standards. While P1 mentions following some regulatory guidelines, specific information about the regulations referenced or how they are applied needs to be included. There is no indication of the specific data protection laws that both projects follow when handling personal data.

Consent

For personal data use consent, P1 explicitly states that by using the website and its services, users are giving their consent for the use of personal data. However, P2 lacks a clear procedure for obtaining consent from users for the use of personal information. Neither P1 nor P2 explain the process for obtaining consent for terms and conditions, risks, or consent withdrawal following the GDPR law [251].

One key observation derived from the assessment underscores the deficiency of information in the setting of both projects. The absence or inadequate setting information poses a significant challenge to establishing expectation models and building prior knowledge for users. While these projects often prioritise promoting their products, important explanations related to compliance and consent are absent, contributing to an overall perception of distrust. As these applications are still in their infancy, we emphasise the importance of providing users with the requisite information before consenting to the contract. To address these deficiencies and gaps in such projects, we advocate explicit explanation requirements for both setting and outcome. As presented in the next section, we focus on the design of explanations for smart contracts. However, we encourage researchers to contribute to

the development of solutions or standards that address setting information in decentralised applications.

5.5.2 Application of Explanation Purposes

In this section, we showcase how explanation purposes can serve as an integral resource for designing smart contracts. Expanding on our previous discussion about the limitations of setting information, we advocate for the scenario-based explanation purpose design approach. This approach is essential for capturing explanation requirements for the setting information such as (O1, O2 and O3) listed in Figure 5.3. We strongly recommend that designers and engineers adopt a proactive strategy of providing thorough details and explanations before users engage with contracts. This proactive measure aligns with the design principle “Immutability Requires Proactive Measures" proposed in Chapter 3, Section 3.6.2.

However, in this section, we shift our focus to the design of smart contracts, incorporating the explanation requirements envisioned by our purposes. We have leveraged the scenarios outlined at the beginning of Section 5.5—namely, Scenario 1 (Lending Decision), Scenario 2 (External Resources) and Scenario 3 (Roles and Responsibilities)—which have determined the requirements and elements presented in Figure 5.3. These elements helped define the specific implementation of explanations.

To incorporate explanation capabilities into the evaluated smart contracts, we must redesign existing practices and redeploy them as new projects. This step is essential due to the immutable nature of smart contracts. Additionally, there are various implementation strategies and the choice of strategy may vary based on the project's specific needs. Such flexibility showcases how our purposes can be adapted and tailored to meet the specific requirements of each smart contract application. Detailed applications are demonstrated

in the following cases and actual smart contracts implementation is provided in the public repository.

Case 1: Consent

We advocate integrating consent mechanisms within smart contract codes, which are critical for creating binding agreements. Recording the consent status on the blockchain can enable wider adoption in situations requiring explicit agreement. We distinguish two types of consent: personal information use and agreement to terms and conditions, with the latter important for high-risk decisions that necessitate explicit user consent before execution. Our chosen method for implementing consent requirements is detailed in the algorithm 1, where we introduce a boolean state variable to track the user's consent status. Additionally, we have defined a modifier that verifies the user's full consent status before executing high-risk functions. We also include a function that permits users to withdraw their consent anytime, aligning with GDPR principles [251].

Case 2: Compliance

Designing compliance mechanisms in smart contracts can be intricate due to the involvement of third parties in validating contract adherence. In our approach, we assume that compliance checks are overseen by auditors or third-party entities responsible for verifying the project's adherence to regulations. We underscore the importance of recording this information on the blockchain to facilitate contractual agreements, considering that some contracts require compliance to protect consumers. In our implementation, as shown in algorithm 2, we introduce a new state variable to track compliance status, which is only updated by the auditor. The contract providers or owners assign this role. For transparency, users can check the compliance status recorded on the blockchain. Furthermore, we designed a string

Algorithm 1 User Consent Smart Contract

Struct UserConsent:

personalInfoConsent (bool) - false

termsAndConditionsConsent (bool) - false

Mapping userConsents:

address → UserConsents

Modifier hasConsent:

require(user.personalInfoConsent == true, *error*)

require(userConsents.termsAndConditionsConsent == true, *error*)

function PROVIDEPERSONALINFOCONSENT

userConsents[msg.sender].personalInfoConsent ← true

function PROVIDETERMSANDCONDITIONSCONSENT

userConsents[msg.sender].termsAndConditionsConsent ← true

function WITHDRAWPERSONALINFOCONSENT

userConsents[msg.sender].personalInfoConsent ← false

function WITHDRAWTERMSANDCONDITIONSCONSENT

userConsents[msg.sender].termsAndConditionsConsent ← false

of explanations as an option for auditors when detailed explanations are necessary.

Algorithm 2 Compliance Smart Contract

State Variables:

bool isCompliant ← **false**

address auditor

Optional:

string personalInfoComplianceExplanation

string termsAndConditionsComplianceExplanation

Modifier onlyAuditor:

require(**msg.sender** == auditor, *error*)

function SETAUDITOR(**(address** _auditor**)** *onlyOwner*)

auditor ← _auditor

function SETCOMPLIANCE(**(bool** compliant**)** *onlyAuditor*)

require(isCompliant == **true**, *error*)

isCompliant ← compliant

function SETCOMPLIANCEEXPLANATIONS(**(string memory** personalInfo, **string memory** termsAndConditions**)**)

personalInfoComplianceExplanation ← personalInfo

termsAndConditionsComplianceExplanation ← termsAndConditions

Case 3: Improvement to Roles & Responsibilities

We observed discrepancies in the information provided by the evaluated projects concerning the roles and responsibilities. These discrepancies could lead to potential surprises, as users may not be aware of the various parties making decisions that impact them. To enhance the clarity of outcome information regarding roles and responsibilities, we introduced a new array

structure. This structure encapsulates role names, associated addresses and descriptions, aligning with the requirements in Figure 5.3. Users can access a list of roles within the project, along with their corresponding addresses and associated responsibilities, through the `getAllRoles()` function, as detailed in the algorithm 3.

Algorithm 3 Role Management Smart Contract

```
Struct RoleInfo:
    string name
    address roleAddress
    string description
State Variable:
    RoleInfo[] public roles
function createRole((string name, address _address, string _description))
        bool roleExists ← false
        for (uint i ← 0; i < roles.length; i++)
            if(roles[i].name == _name)) then
                roleExists ← true
                break
            endif
        endfor
        require(!roleExists, "Role already exists")
        roles.push(RoleInfo(_name, _address, _description))
function getRoles(() public view returns (RoleInfo[]))
        return roles
```

Moreover, we modified the existing design of evaluated smart contracts based on their specific implementations to enhance clarification and justification. For example, in project P1, regarding contract ownership (roles) changes, we integrated an event logging system to

record these changes. This log captures both the former and new owner addresses, along with justifications for the change to satisfy requirements R3 and R6, as specified in Figure 5.3. Additionally, we implemented an event to record owner executions, such as changes in interest rates, providing clarification and justifications to keep users informed (R5 and D1). In project P2, which already had programmed events, we adjusted parameters to include string justifications for owner decisions (R5 and R6), addressing role changes or supply adjustments that may impact the user's lending outcome.

Case 4: Improvement to External Resources

Initially, we modified the existing events related to lending decisions for both projects. We added variables to record the exact input retrieved from external sources, which justify the lending amounts to satisfy (O4). In project P1, we integrated new events to record changes in oracle addresses along with justifications (O1 and O5). Additionally, to address ethical concerns raised by reviewers regarding the manual price entry function in P1, we added an event to log manual price entries with justifications. For project P2, minimal adjustments were made. We expanded an existing event to justify oracles address changes (O5).

Case 5: Improvement to User Decision

As P2 demonstrated a low level of surprise potential in lending decision requirements listed in Figure 5.3, no additional implementation was deemed necessary. In the case of P1, the project already incorporates getter functions for all the variables used in the lending decision, which can facilitate the generation of explanations. We made minimal adjustments by adding a few parameters to existing events, with the goal of improving clarity on the values utilised in the lending decision-making process.

As demonstrated, explanation purposes can influence the design and enhancement of new and existing features. A variety of strategies can be employed to improve explainability, aligning with smart contract capabilities. One effective method involves prioritising the recording of key variables and values on the blockchain. This recorded information can then be integrated with front-end systems and Web3 applications to build a more comprehensible explanation for users.

5.5.3 Cost Analysis

Smart contracts on the Ethereum blockchain follow a transaction model based on gas, a unit measuring the computational work required for execution. Gas fees, paid in Ether (ETH), compensate miners or validators for their computational resources in processing and validating transactions. For instance, the deployment cost of a new contract can range from cents to thousands of US dollars, influenced by Ethereum prices ranging from $1,500 to $2,000 in 2023 [56]. This cost is calculated based on (i) the Ethereum token price, (ii) the compiled contract size (in bytes) and (iii) the current gas price on the Ethereum network. However, factors such as code complexity, tips, computational resource needs and network congestion can increase costs [89]. Additionally, fixed fees are associated with specific operations, such as 'CREATE' and 'TRANSACTION, whereas setting storage variables comes with distinct fees. The detailed breakdown of operational costs can be found in [320].

To examine the trade-offs between costs and explanations, we deployed and executed contracts relevant to the assessed scenarios, implementing a simplified approach by removing dependencies. Our emphasis was solely on constructs related to scenarios, creating a controlled environment for analysis. Gas amounts, representing transaction costs, were documented during the deployment and execution of specific functions linked to the presented scenarios before any modification. Following this, we implemented explanations and docu-

mented the subsequent deployment and execution fees of the same functions for comparison. This methodology allows us to analyse the additional computational expenses incurred by integrating explanations in smart contracts.

The computation cost involves gas used, gas price (measured in Gwei, a subunit of Ether) and the current Ether-to-USD exchange rate [71]. To calculate the cost, the formula based on Ethereum documentation [3] is

$$\textit{Total Cost (in ETH)} = \frac{\textit{Gas Units (Limit)} \times \textit{Gas Price}}{1{,}000{,}000{,}000}$$

- *Total Cost (in ETH)*: This represents the total cost of the transaction in ETH and it is calculated by dividing the product of gas limit and gas price by the conversion factor (1,000,000,000 Gwei = 1 ETH).
- *Gas Units (Limit)*: The maximum amount of gas units allocated for the transaction, representing computational resources. Gas units refers to the actual computational work consumed during the execution of a transaction or interaction with a smart contract.
- *Gas Price*: The price paid for each gas unit, measured in Gwei. Miners are more likely to prioritise transactions with higher gas prices when including them in blocks. The gas price influences the transaction's priority on the network.

The resulting cost in Ether is then converted to USD using the prevailing exchange rate. For our analysis, we adopted an average gas price of 39 Gwei and an Ether value of $1980, as of 16/11/2023, which was obtained from [56, 90]. The corresponding costs for each contract and function used in the evaluation have been recorded, as detailed in Table 5.7.

[3] https://ethereum.org/developers/docs/gas

Table 5.7: Overall Cost Calculation Before and After Explanations

	Before Explanation		**After Explanation**		**Changes**	
Element	**Gas**	**Cost in USD**	**Gas**	**Cost in USD**	**Difference**	**Percentage Increase**
P1-Contract1	1657342	$127.98	1826049	$141.01	$13.03	10%
P1-Contract2	4426651	$341.83	4795125	$370.28	$28.45	8%
P1-Contract3	4992326	$385.51	6962019	$537.61	$152.10	39%
P1-Function A	28510	$2.20	29406	$2.27	$0.07	3%
P1-Function B	28761	$2.22	30001	$2.32	$0.10	4%
P1-Function C	25858	$2.00	32046	$2.47	$0.48	24%
P2-Contract1	501512	$38.73	501512	$38.73	$0.00	0%
P2-Contract2	2373244	$183.26	4100839	$316.67	$133.40	73%
P2-Function A	47797	$3.69	49228	$3.80	$0.11	3%
P2-Function B	33372	$2.58	35970	$2.78	$0.20	8%
P2-Function C	30539	$2.36	31992	$2.47	$0.11	5%

It is important to recognise that the provided prices in Table 5.7 are approximate and do not precisely reflect the costs on the Ethereum mainnet. The exchange rate fluctuates daily and this experiment is conducted on local blockchains and testnets. Additionally, gas prices are subject to variations based on network conditions. Increased demand or congestion can increase gas prices, impacting the overall cost of deploying and interacting with smart contracts.

Cost Results Interpretation

Integrating explanations into smart contracts can lead to a noticeable increase in deployment costs, particularly in cases such as P1-contract3 and P2-contract2, where most explanation functions are implemented. This rise is due to using storage in smart contracts, especially

strings that consume significant storage. In smart contracts, there are three types of data storage: calldata, memory and storage [89]. Calldata and memory serve as temporary storage during contract execution and are cleared once the execution is finished. In contrast, storage involves the persistent storage of values on the blockchain and significantly impacts costs. Storing strings in state variables and emitting events for explanations can be costly. The business owners are primarily responsible for covering the deployment costs. In contrast, executing functions accompanied by explanations has experienced only a marginal increase. Users are required to pay a few extra cents when interacting with these functions, exemplified by P1-Function A and P2-Function A and C. However, in P1-function C, ethical concerns arise due to its allowance of manual token price manipulation. Therefore, we implemented justifications and events to record changes for transparency, leading to a significant increase in cost.

5.6 Discussion

While smart contracts hold significant promise, their design requires substantial refinement and innovation. To fully realise smart contracts potential and facilitate wider adoption in real-world applications, it is essential that we investigate current limitations and explore avenues for improvement, as attempted in this chapter.

We have developed an approach centred around explainability purposes, designed as integral resources for evaluating and designing blockchain-agnostic smart contracts. Although our demonstrations utilised Ethereum smart contracts due to their widespread use for deploying smart contracts, we employ a scenario-based design that can be adaptable and tailored to any blockchain platform that supports smart contracts. We aim to support engineers and designers in proactively eliciting requirements and design aspects that consider

explanation requirements for potential user interactions and challenges.

5.6.1 Key Purposes in Smart Contract Explanations

Our comprehensive synthesis establishes connections between contract law principles governing binding contracts and the goals of explainability in AI systems. This synthesis has unveiled primary purposes essential for improving smart contract design in terms of explanation. The identified purposes align with the inherent characteristics of smart contracts, emphasising the importance of justification, clarification, compliance and consent. Although our focus in this study centres on these four primary purposes, it is noteworthy that other goals exist within the broader landscape of explainability. Future research endeavours could explore additional dimensions such as learning, management, evaluation, or improvement to enhance further the understanding and implementation of explanations in smart contracts.

5.6.2 The MEB Evaluation Framework

We explored how the MEB theory informs a theoretical framework for evaluating surprise potential in smart contracts. This framework systematically assesses surprise potential across various scenarios and systems due to insufficient setting and outcome information.

A significant observation emerged during the evaluation process: The setting information of smart contract systems often lacks critical details, including terms and conditions, policy of use, legal compliance, consent information and associated risks. This information is valuable for establishing user expectations and forming the foundation for a contracting process where users fully understand and agree to all functionalities and associated risks before executing the contract. Users need the necessary background knowledge to build their expectations and knowledge models. Navigating this new paradigm highlights the need to

prioritise developing and standardising comprehensive setting information. Consequently, we encourage researchers and designers to investigate further into the development of standardised setting information for future research.

5.6.3 Cost Considerations

Our evaluation revealed an increase in the deployment costs of smart contracts with explanations. This rise is linked to the utilisation of storage in smart contracts, where the storage of strings in state variables and the emission of events for explanations result in substantial gas costs. Business owners who are responsible for deployment expenses must take these costs into account. However, there was only a marginal increase in the execution costs of functions with explanations. This slight rise, translating into a few extra cents for users, indicates that operational costs associated with explanations are relatively manageable.

This study emphasises the experimental aspects of implementing explanations in smart contracts. Effective optimisation strategies can be employed to mitigate the associated cost implications. One such optimisation approach involves prioritising the storage of critical explanation variables and functions that handle numerical values and booleans. These types generally incur lower costs compared to strings in the Ethereum virtual machine since strings involve more complex operations, resulting in higher gas costs. Additionally, leveraging established error mechanisms within smart contracts, where strings of errors are stored separately and referenced by numerical codes, offers a promising strategy to reduce gas costs significantly. These numerical codes can be integrated with web3, streamlining the retrieval process when specific codes are passed. Designing explainable smart contracts is not a one-size-fits-all solution. Instead, it necessitates a meticulous examination of diverse requirements and the thoughtful design of various aspects of the entire system.

5.6.4 Threats of Validity

One potential internal threat to validity is the formulation of explanation purposes. The risk lies in not having a comprehensive and complete list of explainability purposes for smart contracts. To mitigate this risk, we adopt a dual-perspective approach. The first step is to examine XAI objectives and goals and identify similar practices that can be applied to smart contracts which focus on transparency and understandability. Secondly, we investigated the traits of traditional contracts that make them enforceable and show similarities to smart contract characteristics to utilise them to formulate these purposes. Even though there may be additional explanation purposes, we have deliberately concentrated on the most pertinent ones within the scope of this study. This focus covers both the decision-making process and the contracting procedure for assessment and evaluation.

Furthermore, the concept of explainability purposes was meticulously designed for versatility, addressing the diversity of smart contract systems that necessitate human interaction. This adaptability serves as a foundation, enabling customisation and contextualisation while recognising the distinctive requirements and nuances inherent in smart contract systems.

A possible threat to our evaluation process arises from the novelty of the MEB approach to measuring surprises. While established studies in information systems and adaptive system research use surprise theory [188, 24, 17, 192], these primarily focus on measuring surprises based on the variance between expected and actual outcomes. In contrast, our methodology employs the MEB theory, asserting that explanations can effectively mitigate surprises arising from a deficiency of knowledge or incomplete information about a system (epistemic uncertainty). The MEB theory is well-established in cognitive science and has received empirical support from various studies [196, 106, 197, 103]. Several computational models have been developed across different disciplines [105, 104, 195]. Therefore, to ad-

dress this potential threat, our evaluation framework is founded on a theoretical base and in this study, we showcased its application to validate its feasibility and utility in real-world settings.

Additionally, given the subjectivity involved in evaluating explanations across various contexts, we have enhanced the MEB evaluation method to incorporate a process akin to code peer review. To effectively use our method, at least two evaluators are required to assess potential surprises and it is preferable to involve more evaluators. We also propose using an agreement measurement, similar to what we employed with Cohen's Kappa [285]. By doing this, we aim to mitigate potential biases that could be introduced by the evaluation method and increase the reliability of the results.

In evaluating potential threats to the validity of our process, a critical aspect to consider is the selection and treatment of study cases. This involves the risk of bias when choosing specific use cases and projects for evaluation. To address this concern, we conducted a thorough assessment of various use cases, emphasising those that include three essential elements: decision-making with substantial user impact, reliance on external data and the involvement of human authorities. Although some use cases, such as flight or weather insurance, shared similar attributes, they lacked real-world application, making them less suitable for evaluation. As a result, we opted for lending decentralised applications as they embodied all three pivotal elements, along with additional dimensions of consent and compliance, providing a robust foundation for application. Moreover, selecting two projects with the same use case facilitates valuable insights into the varying levels of potential surprises arising from information provision. This approach offers a meaningful comparison of established practices within different projects operating within the same use case.

5.7 Related Work

Smart Contracts: Recent studies have highlighted the increasing interest in blockchain smart contract technology, discussing its potential applications, challenges and future directions [164, 206, 1, 208, 308]. Some studies have focused on the legality of smart contracts by discussing their limitations in meeting the traditional legal requirements for contract formation. For instance, the lack of a universally accepted definition and their potential incompatibilities with existing legal frameworks, which raise significant challenges for their enforceability and regulatory acceptance [99, 117]. Ethical and social concerns are also prominent in discussions about smart contracts [12, 76, 176]. Automating contractual obligations can exclude necessary human judgment, and enforcement may lead to ethically questionable outcomes.

Therefore, the literature recognises the need to understand and address the limitations of trust. The study by [255] analyses trust in blockchain within the context of reputation systems, focusing on how different types of distributed ledger technologies impact trust. Similarly, the study by [5] emphasises that trust in smart contracts can be improved with the involvement of legal professionals. It proposes language requirements that are human-readable and user-friendly for both lawyers and programmers. These studies indicate a broader recognition within the literature of the importance of addressing trust, transparency and human understanding in smart contracts.

Surprise Theories: Theories of surprise in cognitive psychology fall into three categories: probability, expectations and sense-making. The probability theory examines surprising outcomes as events with low probabilities, utilising Bayesian theory to measure surprise by quantifying the change in an observer's beliefs through the divergence between prior and posterior distributions, and their computational models are exemplified in the studies [24, 17]. However, this approach requires calculated prior beliefs or expectations to calculate

surprise. Expectation-disconfirmation theory attempts to overcome this problem by suggesting that genuine surprise occurs when unexpected events conflict with expected ones, focusing on the subjective gap between what is expected and what occurs. This theory posits that surprise happens when an event deviates significantly from an expected schema, as demonstrated in studies [191, 192].

In contrast, sense-making theories such as MEB emphasise explaining and understanding surprising events, typically done retrospectively rather than predictively [160]. Several models have been proposed from this perspective of surprise akin to our study. For example, the study by [104] proposes the EAMoS model, based on the MEB theory, for analysing the explanation structure of surprising events. It constructs a directed graph of explanations from provided text descriptions, linking the setting to the outcome to predict the surprise rate of the outcome. Similarly, the study by [195] developed a computational model that takes short scenarios as input and outputs a surprise rating for the final sentence. This model consists of two stages: an integration stage, which creates a cohesive representation of the scenario using WordNet, and an analysis stage, which produces a surprise rating for a specific event based on the extent to which the prior representation supports that event.

5.8 Summary

This study explored the multifaceted landscape of smart contract explanations through their purposes, evaluation methodology and associated cost implications. Our investigation was driven by the overarching goal of reconciling surprises within smart contract interactions. We identified four primary purposes of explanations—justification, clarification, compliance and consent. These purposes were designed to be adaptable across diverse smart contract systems. We developed the MEB assessment framework to systematically evaluate surprise

potential in smart contracts, offering insights into the industrial practices of smart contract systems. We evaluated the effectiveness and applicability of our approach through two real-world DApps.

This evaluation highlighted the need for significant improvements in consent, compliance, justification and information enhancement within the setting. Additionally, we examined the cost implications of incorporating explanations. While we observed increased deployment costs, we highlighted that optimisation strategies, such as prioritising storage for critical variables and leveraging established error mechanisms which can effectively mitigate these costs. Therefore, this chapter contributes to the broader comprehension of smart contract explainability requirements as valuable resources for designers and engineers to evaluate explanation needs, embed necessary explanations and understand cost implications.

Chapter Six

Reflection and Appraisal

6.1 Overview

This chapter aims to revisit the research questions presented in Chapter 1 and assess how they have been addressed throughout the thesis. It also provides an overview of the evaluation process for each contribution made in the research.

6.2 Analysis of the Research Questions

This section examines the extent to which the previous chapters have addressed the four research questions.

RQ1: a) What are the most commonly reported concerns regarding smart contracts from a human perspective, and how are these concerns currently being addressed? b) How can we identify quality attributes commonly associated with these human-centred concerns?

In Chapter 2, we performed a systematic literature review to identify common con-

cerns from stakeholders' perspectives in the domain of smart contracts. Our findings indicate that human concerns related to smart contracts are primarily associated with two key stages: Development and interactions. We classified development concerns into three categories: Language, legality and ethical and social implications. The language concerns included complexity, code readability and expressiveness. These concerns affect technical developers, non-technical experts and collaborative development teams as they work together on the design, implementation and deployment of smart contracts. Interaction concerns, which impact end-users, centred on usability, human readability, governance, trust and costs. To deepen our understanding of the state of the art, we mapped frequently reported human-centric concerns to system quality attributes [151] to provide a contextual understanding of the deficiencies in these systems. Utilising the NIST standards for trustworthiness [220], we identified that explainability and interpretability are often overlooked in smart contracts while transparency and accountability have received limited attention in the literature.

Based on the SLR results, we observed a notable gap concerning the requirements and design aspects of human-centric smart contracts. Most research on human considerations has predominantly focused on developing new languages and external tools. This prevailing focus has led to the neglect of important trustworthiness qualities that consider the human in the loop. As a result, there was a critical gap in designing smart contracts with tailored human-centred quality attributes that support the unique nature of smart contracts.

In particular, explainability has not been recognised as a quality attribute within smart contracts. There is a lack of established methods and frameworks addressing explainability requirements, design and implementation. Moreover, evaluation methods and trade-offs associated with integrating explainability into smart contracts remain unexplored. To address these gaps, this thesis aimed to: (i) Systematise the existing knowledge of transparency, accountability and understandability to elucidate the role of explainability requirements in smart contracts. (ii) Develop a human-centric framework to determine information

and explanation requirements for designing explainable smart contracts. (iii) Evaluate the need for explanation through the lens of explainability purposes to reconcile surprises and investigate cost trade-offs.

RQ2: a) What is the state of the art of explainability, transparency, accountability and understandability in blockchain smart contracts? b) How do these concepts align with standardised definitions? c) How can the interrelationships among these concepts guide the development of explainability in smart contract systems?

In Chapter 3, we devised a systematic knowledge framework that classifies, defines and allows discussion of the current state of transparency, accountability and understandability of smart contracts. This framework categorises the acquired knowledge from developers' consultations and literature reviews into five distinct levels: (i) output, (ii) algorithm, (iii) external data, (iv) process and (v) application. Our findings revealed a complex array of challenges that unravel the multi-dimensional aspects and common misconceptions surrounding these concepts in smart contracts. This was compounded by the lack of standardised definitions specifically defining these qualities for blockchain and smart contracts, as evident in our comparison with general standardised definitions. The analysis revealed that, while smart contracts exhibit transparency and accountability in low-level aspects such as output and algorithm, they fall short in more complex dimensions such as process and application. Additionally, all levels demonstrated a pronounced deficiency in understandability.

These observations underscore a pivotal insight: There exists a complementary relationship between explainability and the triad of transparency, accountability and understandability. In smart contracts, explainability acts as an enabler that connects low-level technical details with high-level conceptual clarity. For example, transparency provides visibility of code and transactions, while explainability ensures that this information is com-

prehensible. Accountability provides traceability, while explainability makes decision rights and responsibilities transparent and understandable. Moreover, explainability complements understandability by breaking down complex smart contract operations into simpler, more comprehensible explanations.

Recognising the critical importance of explainability, we developed comprehensive guidelines to assist researchers and practitioners in this area. Our guidelines, specifically tailored to the unique characteristics of blockchain smart contracts, address two main stages of early development: (i) requirement analysis and (ii) design. The guidelines include the identification of explainability requirements through fundamental questions such as who, what, why, when and how Additionally, explainability design principles are proposed as a holistic approach encompassing the entire lifecycle of smart contracts. This approach emphasises the role of designers and developers in prioritising the interests of stakeholders throughout the development and interaction stages. To assess the feasibility and effectiveness of these guidelines in shifting smart contract design, we demonstrated one implementation strategy, prioritising explainability alongside transparency, accountability and understandability.

RQ3: How can a human-centred design approach be utilised to identify the specific information requirements and content necessary for explaining smart contract decisions?

In Chapter 4, we developed a structured human-centred framework to determine the information requirements necessary to design explainable smart contracts (XSC) systems. This framework addresses the elicitation and analysis of explainability requirements, focusing on the fundamental question of 'what to explain' in smart contracts. We integrated the SA definition and GDTA from human factors literature, proposing three levels of XSC explanations: For perception, comprehension and projection. These levels are tailored to determine explanatory information by considering the behavioural properties and

decision-making structures of smart contracts. We categorised behavioural properties into three main components that shape smart contract behaviour: Logic, data and human intervention. Additionally, we classified the decision-making mechanisms, according to their characteristics, into governance structure (centralised vs decentralised), process location (on-chain vs off-chain), degree of automation (fully vs semi) and behavioural pattern (fixed vs dynamic). These classifications serve as a structured framework for requirements engineers, aiding them in determining informational requirements for smart contract decisions. This elicitation of information requirements, in turn, informs the development of explanatory mechanisms through the three levels of XSC-tailored explanations, which are structured to align with the users' needs for awareness, reasoning and projection.

Our framework addresses a critical gap in the current landscape, where no standardised methods exist for determining explanations or information requirements in smart contract systems. Our framework has been recognised by smart contract experts for its usefulness, feasibility and ease of use. To enhance its clarity and understandability, we demonstrated the use of the framework through a practical scenario that highlighted its utility and applicability. The framework considers each decision individually, showcasing its versatility in addressing various use cases within the blockchain ecosystem, which further validates its relevance and effectiveness. This chapter was based on the work presented in [4].

RQ4: a) What primary explanation purposes can be integrated into the design of smart contracts? b) How can the MEB theory inform the creation of a systematic framework to assess the potential surprises in smart contracts when explanations are absent? c) What are the potential trade-offs regarding costs when integrating explanations into smart contracts?

We embarked on a comprehensive evaluation of explainability as an integral resource

for designing smart contracts through the lens of their purposes. This study aimed to address the fundamental question of 'why to explain' as part of our explainability requirements analysis. We posited that smart contract designers and requirements engineers can embed explanations to clarify, justify, ensure compliance and facilitate consent. These purposes are specifically tailored to the characteristics of smart contracts by combining insights from established AI explainability practices and elements of legally binding traditional contracts. This approach demonstrated effectiveness in two ways: First, by evaluating existing smart contracts in terms of potential surprises stemming from epistemic uncertainties (i.e., lack of knowledge and information) regarding justification, clarification, consent and compliance; and second, as a design approach helping to implement explainability in new smart contracts.

We developed a novel assessment framework that uses surprise as a guiding factor to systematically identify areas requiring improvement in terms of justification, clarification, compliance and consent. The evaluation method is based on the MEB theory, which conceptualises the resolution of surprise as a process of fitting new information into existing mental frameworks, emphasising the role of explanations. We created a generic computational model of this theory to systematically pinpoint areas that lack explanation and information provision, which can lead to potential surprises.

We demonstrated the utility and applicability of the explainability purposes as evaluation mechanisms by using the MEB evaluation method to systematically assess potential surprises in two real-world lending projects. The results showed that the most noteworthy instances of potential surprises originated from deficiencies in setting information, where users established their expectations and assessed eventual outcomes. Additionally, the outcomes of smart contracts revealed a lack of decision justification, clarity on the roles and responsibilities of privileged parties, decision mechanisms and critical information to facilitate consent and compliance.

Given these findings, our explainability purposes demonstrated their effectiveness and feasibility in designing explanations for smart contracts. We designed and implemented these requirements to address the shortcomings in existing projects to validate our approach. Additionally, we investigated the trade-offs between cost and explainability. Our evaluation revealed an increase in deployment costs linked to the complexity of Ethereum smart contracts, especially with the use of complex data types such as strings. However, there was only a marginal increase in the execution costs of functions with explanations, amounting to a few extra US cents per user. Furthermore, we highlighted that optimisation strategies, such as prioritising storage for critical variables and leveraging established error mechanisms, can effectively mitigate deployment and execution costs.

6.3 Reflection on the Research

In this thesis, we employed hybrid evaluation methods tailored for each chapter, incorporating DSRM evaluation techniques [302, 241, 282, 166]. We leveraged these methods by conducting expert surveys and consultations to assess draft frameworks and validating our work through qualitative measures such as usefulness, ease of use and benchmarking. Additionally, we demonstrated the applicability and feasibility of our frameworks through practical scenarios, evaluation of real-world projects and smart contract implementations, which highlight the frameworks' effectiveness in solving real-world problems.

6.3.1 Validation Criteria

We reflect on our hybrid evaluation techniques by applying the criteria established by Kitchenham et al. [166], which were originally utilised to validate design science methodology evaluation methods and tools (DESMET). This approach involves three levels of validation:

Basic, use and gain, as follows.

Basic Validation: This evaluation concerns the quality of the component documentation. For the reflection exercise, we selected the following subfeatures that were suitable for our context:

- **Documentation Completeness:** This quality may carry different interpretations because it involves various dimensions and subjective perspectives, making it nearly impossible to address every aspect completely. Therefore, we defined specific criteria focusing on reporting style, including scope and section coverage, depth and contextual information, practical examples and case studies and supplementary materials. To ensure comprehensive topic and section coverage, we followed rigorous reporting guidelines [258, 281] including defining clear research objectives, maintaining transparency in methodology, detailing data collection and analysis processes. To guarantee depth and contextual information, we analysed literature, industry reports, standards and experts' perspectives. Each contribution chapter includes practical examples and case studies, such as flight insurance, privileged account scenarios and lending decentralised applications. Given the word limit constraints of this thesis, we summarised some findings and implementations in the chapters. However, detailed supplementary materials are provided as appendices referenced in each relevant section. Additionally, we used GitHub repositories to share the full results of Chapters 4 and 5 including the case studies implementations.

- **Appropriateness for Audience:** This thesis was meticulously tailored to blockchain and smart contract researchers and requirements and software engineers. Our contribution statements specifically address this audience. We ensured the language and content aligned with their expertise and enabled common understanding by defining key concepts and glossaries in the background sections. In Chapters 3 and 4, inter-

views and surveys with blockchain and smart contract specialists provided consistent feedback, helping us improve our artefacts to better meet their needs. Additionally, we incorporated real scenarios and example cases throughout our chapters to bridge theory and practice, ensuring the documentation's relevance and usefulness. This practical approach helps the audience relate to the material and understand its contexts.

- **Organisation:** We rigorously structured our reporting for each contribution chapter (2, 3, 4 and 5) to include (i) an overview introducing the problem and contributions, (ii) fundamental concepts, (iii) a research approach explaining the methodology, (iv) results presenting novel frameworks and approaches, (v) evaluations, (vi) discussion and threats to validity, (vii) related work and, finally, (viii) a summary to conclude the chapter. For example, the research approaches in Chapters 3, 4 and 5 each followed structured reporting. Chapter 3 encompassed four main stages for knowledge acquisition, systematisation, comparison and customisation. Chapter 4 detailed the framework creation steps from SA and GDTA. Chapter 5 detailed the development stages for explainability purposes and the MEB evaluation method. Additionally, we used tables to organise information and show synthesis results in each contribution chapter, for example, Tables 3.3, 4.4, 5.3. This rigorous reporting method ensured consistency in presenting chapters, methodologies and results, thereby maintaining reader understandability and transparency throughout the thesis.

Use Validation: This evaluation concerns the quality of a component and its use. For the reflection exercise, we selected the following subfeatures that are suitable for our context:

- **Completeness:** This quality measures the extent to which the developed framework is self-contained and comprehensive. Although we cannot guarantee the absolute completeness of our work, we followed well-established research methodologies and provided

generalised and generic frameworks to accommodate variants. However, in Chapter 3, the completeness of our systematisation, classification and results were limited by the available data, despite our efforts in conducting the SLR and incorporating insights from developer consultations. Nevertheless, the explainability requirements and design principles in Chapter 3 are generic guidelines adaptable to different contexts and scenarios. In Chapter 4, we constructed the framework by integrating the three main components of SA, GDTA and smart contracts decision operational structure with predefined steps designed to accommodate variants and future advancements. To address potential incompleteness, we synthesised smart contract decision components, gathering knowledge from diverse sources such as literature, use cases, blogs and white papers. We employed the principle of separation of concerns to generalise components which are the fundamental building blocks in most smart contract decisions. Our framework is not exclusive to the proposed components as it is adaptable and customisable in order to address unique factors such as regulatory compliance, security models, or other project-specific models. In Chapter 5, we proposed four purposes for explainability, noting that this list may only cover some possibilities. We focused on the most relevant purposes within the thesis' scope, using established XAI practices and traits of traditional contracts. Although other purposes may exist, we introduced the concept of 'explainability purposes' as generalisable design resources, showing how these purposes can be utilised for evaluating and designing smart contracts. While we recognise the challenges in achieving absolute completeness, we ensured our approaches are robust, adaptable and capable of evolving to meet new challenges and requirements.

- **Ease of Implementation:** This quality measures the extent to which the intended audience can easily implement the developed framework. We ensured the explainability requirements analysis followed a generic template addressing the fundamental questions of who, what, why, when and how. This approach is appropriate for smart

contract practitioners with varying levels of expertise, and we provided examples to illustrate the analysis for ease of implementation. The framework in Chapter 4 extends these fundamental questions with predefined inquiries which experts confirmed its understandability and ease of use. However, experts also suggested demonstrating its application, which we addressed through practical flight insurance decentralised scenarios. The work presented in Chapter 5 may require a learning curve for some requirements engineers and designers due to the complexity of measuring subjective surprises. We defined detailed steps with simplified scenarios and examples to improve understandability. Additionally, we offered a complete application and implementation through real-world DApp to assist the target audience in following the steps, along with a generic template that can calculate surprises by simply inputting the results.

- **Application Demonstration:** We provided detailed demonstrations highlighting how our approaches can be applied to real-world scenarios. Although smart contracts have limited real-world applications due to their novelty and developmental immaturity, we selected relevant scenarios that have been discussed in the literature and industry. In Chapter 3, we instantiated the explainability requirements analysis and principles on privileged accounts functionality. In Chapter 4, we selected decentralised flight insurance, which covers policy representation, data integration and human decision-making processes, illustrating our framework's constructs. Additionally, this use case has been considered industrially by Chainlink, which has customised oracles to provide flight data for smart contracts DApps. In Chapter 5, we examined two lending DApps due to their substantial impact on users' financial status. These applications integrate essential elements such as decision logic, external data dependencies and human authority in setting and modifying interest rates. Additionally, they address dimensions of consent and compliance. These demonstrations collectively validate explainability's practical applicability and feasibility in addressing real-world use cases.

Gain Validation: This evaluation concerns benefits delivered by the component. For the reflection exercise, we selected the following subfeatures that are suitable for our context:

- **Usefulness:** First, our methods offer blockchain and smart contract researchers valuable new insights and knowledge, enabling them to pursue research opportunities and advancements in the field through (i) systematisation of the current state of some trustworthy qualities to pinpoint the lack of standardisation (Chapter 3); (ii) a taxonomy and detailed classification of decision-making processes in smart contracts, including behavioural components and decision mechanisms (Chapter 4); (iii) a theoretical evaluation framework designed to assess epistemic uncertainties, which is adaptable across various disciplines and systems (Chapter 5); and (iv) a cost analysis and suggestion of optimisation techniques that can advance the field. Second, our approaches and frameworks for explainability demonstrated their effectiveness and usefulness for requirements and software engineers by providing (i) guidelines to support early intervention in requirements analysis and design principles for diverse DApps (Chapter 3); (ii) a systematic framework to determine information requirements for smart contract decisions, which has been acknowledged by experts for its usefulness (Chapter4); (iii) the concept of 'explainability purposes' as a design and evaluation tool, showing approach usefulness in helping engineers evaluate, elicit, design and implement smart contracts through real-world applications (Chapter 5).

- **Support for decision-making:** Our objective was to inspire a paradigm shift in the perception and design of smart contract systems by addressing current limitations such as human interaction issues, misuse, scams and discrimination. We advocate for innovative thinking in transitioning from centralised to decentralised systems, emphasising the importance of developing responsible and trustworthy systems. The immutable and deterministic nature of decentralised applications, which enforce decisions and

outcomes without central authority oversight, necessitates careful and deliberate development. We propose integrating explainability as a fundamental design concern from the outset. Explainability can encourage decision makers (e.g., owners and developers) to make the development and interaction of smart contracts transparent, accountable and comprehensible, thereby meeting the needs of humans in the loop. We demonstrated the benefits of explainability designs through smart contract implementations and analysis of cost implications. This analysis helps decision makers prioritise initiatives and allocate resources efficiently.

- **Applicability:** This quality concerns the suitability and relevance of the components and artefacts developed in this thesis across various contexts and scenarios. Although we utilised Ethereum and Solidity smart contracts for real demonstration due to their prominence, our contributions focus on early development stages, emphasising applicability beyond specific technologies. For example, Chapter 4 presented an explainability requirement elicitation framework applicable to various decentralised application scenarios. We generalised system components and decision mechanisms to accommodate main characteristics while allowing customisation for specific use cases. Chapter 5 generalised the concept of explainability purposes. While we validated four specific purposes, it can be adapted to suit different objectives. The MEB framework, presented as a generic framework with specified steps, can be customised for other systems and disciplines. In Chapter 3, the explainability requirements and design principles were developed as a generic template and guidelines. However, our systematisation relies primarily on existing knowledge from the Ethereum blockchain and Solidity smart contracts, meaning the results are based on these specific findings.

6.3.2 Limitations of the Proposed Work

This section summarises the limitations and threats to validity of our research as discussed throughout the thesis.

Summary of Limitations

Decentralised applications and blockchain technology are still in early stages, often managed by small teams or individuals, which made collaboration with established organisations challenging. This limited availability and responsiveness within the blockchain community restricted opportunities to apply our frameworks directly in real-world settings. Despite extensive outreach, only a few developers, experts and researchers responded within the PhD's timeframe. To address this, we demonstrated the framework's feasibility and usefulness through real-world scenarios and industry-based applications in Chapters 4 and Chapter 5. Future research could expand this work through partnerships with blockchain organisations and industry collaborations.

Additionally, smart contract technology lacks well-established standards, best practices and a comprehensive understanding of its full potential. Much of the field remains experimental, with ongoing research needed to explore how smart contracts can address real-world challenges. This limitation required us to build foundational knowledge in each chapter due to fragmented information and a lack of consensus on smart contract capabilities. Our contributions, as shown in synthesis analysis outcomes in Tables 3.3, 4.4, 5.3, aim to advance this foundational knowledge and support future research.

Summary of Threats to Validity

Our study encountered several threats to validity which have been addressed through systematic mitigation techniques. Internal validity risks, such as selection bias, were managed by implementing SLRs with clear inclusion and exclusion criteria in Chapter 2 and Chapter 3. We also included diverse perspectives from developers and additional sources, including use cases, blogs, secondary studies and white papers. However, internal validity remained partly constrained by resource limitations and data availability. For the case studies in Chapters 4 and Chapter 5, selection bias was a concern; we mitigated this by establishing strict criteria and selecting cases with varied decision mechanisms to assess our framework's applicability.

External validity was impacted by our focus on Ethereum smart contracts, as the findings in Chapter 3 are largely shaped by this context due to Ethereum's prominence. However, the explainability requirements were designed to be blockchain-agnostic. Additionally, while our expert group may not fully represent the entire blockchain ecosystem, we selected individuals with diverse, proven experience in the field. To enhance generalisability, we developed a customisable framework in Chapter 4 and adaptable explainability purposes in Chapter 5, allowing application across various scenarios and platforms.

Construct validity was strengthened by defining key terms in Chapter 2, organising system layers into five levels in Chapter 3, and using a separation of concerns approach to structure decision-making elements in Chapter 4. Chapter 5 further refined explainability purposes specifically for smart contracts. While these approaches do not ensure completeness, they remain flexible to integrate new classifications and emerging knowledge.

Conclusion validity was supported through thematic analysis across the contribution chapters, alongside secondary studies, standards, grey literature and white papers. Multiple reviewers and developers contributed to achieving a shared interpretation of the data. Future

research with real-world implementation and broader developers input could further validate and extend the applicability of these findings across diverse blockchain ecosystems.

Chapter Seven

Conclusion Remarks and Future Work

This chapter outlines our contributions and suggests directions for future research.

7.1 Contributions

This thesis aims to provide human-centred approaches to guide requirements engineers and designers in creating explainable blockchain smart contracts concerning early development stages. Therefore, we have addressed the following:

- **A systematic review of literature on human-centric design concerns and considerations in blockchain smart contracts.** In Chapter 2, we conducted an SLR to identify stakeholders' common concerns, including programming languages, legality, ethics, usability, readability, trust, governance, and costs. This review revealed gaps in requirements and design interventions, highlighting frequently reported human-centric quality attributes and deficiencies in smart contract systems. Notably, explainability and interpretability are often overlooked, while transparency and accountability require deeper exploration. This review pinpointed gaps in human-centric qualities and

outlined future research directions in smart contract explainability.

- **A systematisation of transparency, understandability and accountability in smart contracts unveils the role of explainability.** In Chapter 3, we developed a framework categorising these attributes into five levels: output, algorithm, external data, process, and application. This framework, informed by literature and developer interviews, provides a structured view of these concepts in smart contracts. By comparing our findings to ISO standards, we identified both alignments and discrepancies, pinpointing improvement areas. We demonstrated that explainability bridges technical details with high-level considerations, enhancing transparency, accountability, and understandability. We also developed guidelines outlining explainability requirements, grounded in who, what, why, when and how. Additionally, we proposed design standards tailored to smart contracts, illustrated with an example case, providing detailed guidance to researchers and practitioners.

- **A human-centric framework to determine information requirements for explainable smart contracts (XSC).** In Chapter 4, we introduced a human-centered framework to define information requirements for XSC systems, utilising SA and GDTA methods. We proposed three levels of XSC explanations—perception, comprehension, and projection—to address users' needs for awareness, reasoning, and foresight. These levels cater to information requirements based on smart contracts' behaviour and decision-making structures. This framework, praised by experts for its feasibility and usefulness, assists requirements engineers in determining necessary explanations and supported by practical demonstrations of its application.

- **An evaluation of explanation needs in smart contracts through the lens of explainability purposes to reconcile surprises.** Chapter 5 introduces the concept of explainability purposes as key resources in smart contracts to (i) assess explanation needs and (ii) design explainability requirements. Drawing on contract law and XAI

practices, we show how designers and requirements engineers can use explanations to clarify, justify, ensure compliance, and facilitate consent. We developed an assessment framework based on MEB theory to evaluate potential surprises from insufficient or missing information. Applied to two real-world DApps, our approach identified potential surprises, established explainability requirements, and demonstrated the design and implementation of explanations. A trade-off analysis further examined the costs of integrating explanations. This work provides essential resources for designers and engineers to create human-centered smart contracts by evaluating explanation needs, embedding necessary explanations and understanding cost implications.

7.2 Future Work

Although our exploration of smart contract explanations has shed light on key aspects, numerous unexplored avenues await further research and exploration. Future directions for our work include the following:

7.2.1 Optimising Explanation Costs

As discussed in Chapter 5, incorporating explanations into smart contracts increases costs due to the additional computational resources required. The literature has investigated optimising smart contract code, often by reducing complexity at the expense of readability [299, 47]. However, our thesis presents a new direction in optimising explanations in smart contracts. We suggest new avenues for researchers by investigating established error mechanisms within smart contracts. This technique involves storing error messages as numerical codes rather than strings, which can incur lower costs in the Ethereum virtual machine.

Additionally, the concepts of global and local explanations in XAI present an intriguing area of investigation that can reduce costs. Global explanations provide the same information to all users while local explanations focus on specific transactions or user interactions. The system can avoid unnecessary duplication and associated costs by ensuring that each explanatory data is stored only once and referenced appropriately (global explanations). Local explanations, on the other hand, provide detailed, context-specific information for particular transactions or interactions. One approach for local explanations is designing smart contracts to return critical information, as shown in our practical demonstration in Chapter 4 [4]. However, our work focused on the early development phases. Future directions should concentrate on methods for developing and implementing contextual explanations, incorporating user studies to evaluate both the explanations and the methods of delivering them.

7.2.2 Aleatory Uncertainties

In Chapter 5, we investigated epistemic uncertainties in smart contracts. Exploring the integration of explanations to address aleatory uncertainty in smart contracts is a new direction worth exploring. Researchers should explore how explainability can be provided for unexpected events to reduce surprises. One study has proposed designing flexible smart contracts to handle unexpected situations by providing a list of actions for stakeholders to vote on in these scenarios [185]. However, it is essential to conduct research first to understand and analyse the context of unexpected events that influence smart contract execution.

Future work can focus on developing dynamic models and frameworks capable of identifying and interpreting aleatory uncertainties to generate suitable explanations for unexpected events, such as market price fluctuations or unforeseen real-world events. One promising area is utilising machine learning and AI techniques to predict by training models

on historical data and potential uncertainties. Then, explanations based on predictive analytics can be generated. Additionally, incorporating feedback mechanisms that allow users to interact with the system and provide input on the explanations is essential. This way can help refine the models and frameworks, making them more accurate and relevant. User feedback can also help identify gaps in the explanations and areas for further improvement.

7.2.3 Impact on Non-Functional Requirements and Lifecycle

In Chapter 3, we proposed the "Full Functionality— Win-Win" design principle, which ensures that integrating explainability features into smart contracts does not reduce their overall effectiveness. This principle opens new avenues for investigating the impact of explainability on other non-functional requirements (NFR) such as security, privacy, performance and cost. A study has explored the impact of explainability on NFRs in a system-agnostic manner [39]. However, most current research focuses on centralised systems.

Future work can explore the impact of NFRs on decentralised systems through several approaches, such as cost-benefit analysis, risk and return on investment analyses. Additionally, empirical studies and controlled experiments can be conducted to identify security vulnerabilities, privacy breaches and performance metrics when incorporating explainability. Future research directions can involve comparing various methods and techniques for embedding explanations, such as adding an additional system layer in smart contracts and evaluating their impact on non-functional requirements and trustworthiness characteristics. This analysis will help identify the most effective approaches.

Furthermore, in Chapter 3, we proposed several explainability design principles. One particularly intriguing principle that opens new streams of future research is "End-to-End Explainability—Full Lifecycle Clarity." This principle emphasises embedding explainability

throughout the smart contract's lifecycle, including the development stage. Our research efforts have primarily focused on user perspectives and interaction. However, the literature on smart contracts underscores significant development issues due to their complexity as explored in Chapter 2.

It is important that researchers investigate methods for embedding explainability into the development process of smart contracts by fostering collaboration between technical and non-technical stakeholders. Developing explainability approaches that bridge the communication gap between these groups can significantly enhance the design and implementation process. The literature mainly provides visual programming environments or domain-specific languages (DSLs). Explainability should be explored to improve clarity and reduce misunderstandings, minimising reliance on third parties. Maintaining explainability throughout the entire lifecycle of the smart contract can promote the design of responsible systems [58] that prioritise explainability, transparency and accountability.

References

[1] Manar Abdelhamid and Ghada Hassan. "Blockchain and Smart Contracts". In: *Proceedings of the 8th International Conference on Software and Information Engineering*. ICSIE '19. Cairo, Egypt: Association for Computing Machinery, 2019, pp. 91–95. ISBN: 9781450361057. DOI: 10.1145/3328833.3328857. URL: https://doi.org/10.1145/3328833.3328857.

[2] Amina Adadi and Mohammed Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052.

[3] John Adler et al. "Astraea: A Decentralized Blockchain Oracle". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2018, pp. 1145–1152. DOI: 10.1109/Cybermatics_2018.2018.00207.

[4] Hanouf Al Ghanmi and Rami Bahsoon. "ExplanaSC: A Framework for Determining Information Requirements for Explainable Blockchain Smart Contracts". In: *IEEE Transactions on Software Engineering* 01 (May 2024), pp. 1–21. ISSN: 1939-3520. DOI: 10.1109/TSE.2024.3408632.

[5] Firas Al Khalil et al. “Trust in Smart Contracts is a Process, As Well”. In: *Financial Cryptography and Data Security*. Ed. by Michael Brenner et al. Cham: Springer International Publishing, 2017, pp. 510–519. ISBN: 978-3-319-70278-0.

[6] Maher Alharby, Amjad Aldweesh, and Aad van Moorsel. “Blockchain-based Smart Contracts: A Systematic Mapping Study of Academic Research (2018)”. In: *2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB)*. 2018, pp. 1–6. DOI: 10.1109/ICCBB.2018.8756390.

[7] Sajid Ali et al. “Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence”. In: *Information Fusion* 99 (2023), p. 101805. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101805. URL: https://www.sciencedirect.com/science/article/pii/S1566253523001148.

[8] Sikandar Ali et al. “Metaverse in healthcare integrated with explainable AI and blockchain: enabling immersiveness, ensuring trust, and providing patient data security”. In: *Sensors* 23.2 (2023), p. 565.

[9] Cigdem Altintas et al. “Machine Learning Meets with Metal Organic Frameworks for Gas Storage and Separation”. In: *Journal of Chemical Information and Modeling* 61.5 (2021), pp. 2131–2146.

[10] Apostolos Ampatzoglou et al. “Identifying, categorizing and mitigating threats to validity in software engineering secondary studies”. In: *Information and Software Technology* 106 (2019), pp. 201–230. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2018.10.006. URL: https://www.sciencedirect.com/science/article/pii/S0950584918302106.

[11] Monika di Angelo and Gernot Salzer. “Characterizing Types of Smart Contracts in the Ethereum Landscape”. In: *Financial Cryptography and Data Security*. Ed. by Matthew Bernhard et al. Cham: Springer International Publishing, 2020, pp. 389–404. ISBN: 978-3-030-54455-3.

[12] Monika di Angelo, Alfred Soare, and Gernot Salzer. "Smart Contracts in View of the Civil Code". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC '19. Limassol, Cyprus: Association for Computing Machinery, 2019, pp. 392–399. ISBN: 9781450359337. DOI: 10.1145/3297280.3297321. URL: https://doi.org/10.1145/3297280.3297321.

[13] Lennart Ante. "Smart contracts on the blockchain – A bibliometric analysis and review". In: *Telematics and Informatics* 57 (2021), p. 101519. ISSN: 0736-5853. DOI: https://doi.org/10.1016/j.tele.2020.101519. URL: https://www.sciencedirect.com/science/article/pii/S0736585320301787.

[14] Bokolo Anthony Jnr. "A developed distributed ledger technology architectural layer framework for decentralized governance implementation in virtual enterprise". In: *Information Systems and e-Business Management* 21.3 (2023), pp. 437–470.

[15] Maria Aslam, Diana Segura-Velandia, and Yee Mey Goh. "A Conceptual Model Framework for XAI Requirement Elicitation of Application Domain System". In: *IEEE Access* 11 (2023), pp. 108080–108091. DOI: 10.1109/ACCESS.2023.3315605.

[16] Nagadivya Balasubramaniam et al. "Transparency and explainability of AI systems: From ethical guidelines to requirements". In: *Information and Software Technology* 159 (2023), p. 107197. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2023.107197. URL: https://www.sciencedirect.com/science/article/pii/S0950584923000514.

[17] Pierre Baldi and Laurent Itti. "Of Bits And Wows: A Bayesian Theory of Surprise With Applications to Attention". In: *Neural Networks* 23.5 (2010), pp. 649–666.

[18] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.12.012. URL: https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[19] Richard Baskerville, Jan Pries-Heje, and John Venable. "Soft Design Science Methodology". In: *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*. DESRIST '09. Philadelphia, Pennsylvania: Association for Computing Machinery, 2009. ISBN: 9781605584089. DOI: 10.1145/1555619.1555631. URL: https://doi.org/10.1145/1555619.1555631.

[20] Tobias Baum et al. "Factors Influencing Code Review Processes in Industry". In: *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. FSE 2016. Seattle, WA, USA: Association for Computing Machinery, 2016, pp. 85–96. ISBN: 9781450342186. DOI: 10.1145/2950290.2950323. URL: https://doi.org/10.1145/2950290.2950323.

[21] Roman Beck, Christoph Müller-Bloch, and John Leslie King. "Governance in the blockchain economy: A framework and research agenda". In: *Journal of the Association for Information Systems* 19.10 (2018), p. 1.

[22] Gregory Bedny and David Meister. "Theory of Activity and Situation Awareness". In: *International Journal of Cognitive Ergonomics* 3.1 (1999), pp. 63–72. DOI: 10.1207/s15327566ijce0301_5. eprint: https://doi.org/10.1207/s15327566ijce0301_5. URL: https://doi.org/10.1207/s15327566ijce0301_5.

[23] Wafa Ben Slama Souei et al. "Towards a Uniform Description Language for Smart Contract". In: *2021 IEEE 30th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. 2021, pp. 57–62. DOI: 10.1109/WETICE53228.2021.00022.

[24] Nelly Bencomo and Amel Belaggoun. "A World Full of Surprises: Bayesian Theory of Surprise to Quantify Degrees of Uncertainty". In: *Companion Proceedings of the 36th International Conference on Software Engineering*. 2014, pp. 460–463.

[25] Vítor Bernardo. *TechDispatch 2/2023 - Explainable Artificial Intelligence*. Tech. rep. Technology and Privacy Unit of the European Data Protection Supervisor (EDPS),

2023. URL: https://www.edps.europa.eu/data-protection/our-work/publications/techdispatch/2023-11-16-techdispatch-22023-explainable-artificial-intelligence_en.

[26] Umang Bhatt et al. "Uncertainty as a Form of Transparency: Measuring, Communicating, and Using Uncertainty". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '21. Virtual Event, USA: Association for Computing Machinery, 2021, pp. 401–413. ISBN: 9781450384735. DOI: 10.1145/3461702.3462571. URL: https://doi.org/10.1145/3461702.3462571.

[27] Or Biran and Courtenay Cotton. "Explanation and Justification in Machine Learning: A Survey". In: *IJCAI-17 workshop on explainable AI (XAI)*. Vol. 8. 1. 2017, pp. 8–13.

[28] Cruz E. Borges et al. "Blockchain application in P2P energy markets: social and legal aspects". In: *Connection Science* 34 (1 2022). ISSN: 13600494. DOI: 10.1080/09540091.2022.2047157.

[29] Hamda Al-Breiki et al. "Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges". In: *IEEE Access* 8 (2020), pp. 85675–85685. DOI: 10.1109/ACCESS.2020.2992698.

[30] BscScan. *BNB Smart Chain Explorer*. [Accessed 15-Nov-2023]. 2023. URL: https://bscscan.com.

[31] Davide Calvaresi et al. "Explainable Multi-Agent Systems Through Blockchain Technology". In: *Explainable, Transparent Autonomous Agents and Multi-Agent Systems*. Ed. by Davide Calvaresi et al. Cham: Springer International Publishing, 2019, pp. 41–58. ISBN: 978-3-030-30391-4.

[32] Gerardo Canfora et al. "iSCREAM: A suite for Smart Contract REAdability assessMent". In: *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 2021, pp. 579–583. DOI: 10.1109/ICSME52107.2021.00060.

[33] John M Carroll. "Becoming Social: Expanding Scenario-Based Approaches in Hci". In: *Behaviour & Information Technology* 15.4 (1996), pp. 266–275.

[34] John Cartwright. *Contract Law: An Introduction to the English Law of Contract for the Civil Lawyer*. Bloomsbury Publishing, 2023.

[35] Ann Cavoukian. *The 7 Foundational Principles Implementation and Mapping of Fair Information Practices*. Tech. rep. Information and Privacy Commissioner of Ontario, 2011. URL: https://iapp.org/media/pdf/resource_center/pbd_implement_7found_principles.pdf.

[36] Chainlink. *Chainlink-Connecting the world to blockchains*. [Accessed 26-May-2023]. 2023. URL: https://chain.link.

[37] Vinay Chamola et al. "A Review of Trustworthy and Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 11 (2023), pp. 78994–79015. DOI: 10.1109/ACCESS.2023.3294569.

[38] Arnab Chatterjee, Yash Pitroda, and Manojkumar Parmar. "Dynamic Role-Based Access Control for Decentralized Applications". In: *Blockchain – ICBC 2020*. Ed. by Zhixiong Chen et al. Cham: Springer International Publishing, 2020, pp. 185–197. ISBN: 978-3-030-59638-5.

[39] L. Chazette, W. Brunotte, and T. Speith. "Exploring Explainability: A Definition, a Model, and a Knowledge Catalogue". In: *2021 IEEE 29th International Requirements Engineering Conference (RE)*. Los Alamitos, CA, USA: IEEE Computer Society, Sept. 2021, pp. 197–208. DOI: 10.1109/RE51729.2021.00025. URL: https://doi.ieeecomputersociety.org/10.1109/RE51729.2021.00025.

[40] Larissa Chazette and Kurt Schneider. "Explainability as a non-functional requirement: challenges and recommendations". In: *Requirements Engineering* 25.4 (2020), pp. 493–514.

[41] Celia Chen et al. "Assessing Software Understandability in Systems By Leveraging Fuzzy Method and Linguistic Analysis". In: *Procedia Computer Science* 153 (2019), pp. 17–26.

[42] Haoxian Chen et al. "Declarative Smart Contracts". In: *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2022. <conf-loc>, <city>Singapore</city>, <country>Singapore</country>, </conf-loc>: Association for Computing Machinery, 2022, pp. 281–293. ISBN: 9781450394130. DOI: 10.1145/3540250.3549121. URL: https://doi.org/10.1145/3540250.3549121.

[43] Jessie Y Chen et al. "Situation awareness-based agent transparency". In: *US Army Research Laboratory* April (2014), pp. 1–29.

[44] Jessie Y. C. Chen et al. "Situation awareness-based agent transparency and human-autonomy teaming effectiveness". In: *Theoretical Issues in Ergonomics Science* 19.3 (2018), pp. 259–282. DOI: 10.1080/1463922X.2017.1315750. eprint: https://doi.org/10.1080/1463922X.2017.1315750. URL: https://doi.org/10.1080/1463922X.2017.1315750.

[45] Jiachi Chen et al. "Maintaining smart contracts on ethereum: Issues, techniques, and future challenges". In: *arXiv preprint arXiv:2007.00286* (2020).

[46] Jiachi Chen et al. "Why Do Smart Contracts Self-Destruct? Investigating the Self-destruct Function on Ethereum". In: *ACM Trans. Softw. Eng. Methodol.* 31.2 (Dec. 2021). ISSN: 1049-331X. DOI: 10.1145/3488245. URL: https://doi.org/10.1145/3488245.

[47] Ting Chen et al. "Under-optimized smart contracts devour your money". In: *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 2017, pp. 442–446. DOI: 10.1109/SANER.2017.7884650.

[48] Olivia Choudhury et al. "Auto-Generation of Smart Contracts from Domain-Specific Ontologies and Semantic Rules". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2018, pp. 963–970. DOI: 10.1109/Cybermatics_2018.2018.00183.

[49] Panayiotis Christodoulou and Klitos Christodoulou. "A Decentralized Voting Mechanism: Engaging ERC-20 token holders in decision-making". In: *2020 Seventh International Conference on Software Defined Systems (SDS)*. 2020, pp. 160–164. DOI: 10.1109/SDS49854.2020.9143877.

[50] Giovanni Ciatto, Stefano Mariani, and Andrea Omicini. "Blockchain for Trustworthy Coordination: A First Study with LINDA and Ethereum". In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. 2018, pp. 696–703. DOI: 10.1109/WI.2018.000-9.

[51] Nazli Cila et al. "The Blockchain and the Commons: Dilemmas in the Design of Local Platforms". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. , Honolulu, HI, USA, Association for Computing Machinery, 2020, pp. 1–14. ISBN: 9781450367080. DOI: 10.1145/3313831.3376660. URL: https://doi.org/10.1145/3313831.3376660.

[52] Douglas Cirqueira et al. "Scenario-Based Requirements Elicitation for User-Centric Explainable AI: A Case in Fraud Detection". In: *International cross-domain conference for machine learning and knowledge extraction*. Springer. 2020, pp. 321–341.

[53] Anne Cleven, Philipp Gubler, and Kai M. Hüner. "Design Alternatives for the Evaluation of Design Science Research Artifacts". In: *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*. DESRIST '09. Philadelphia, Pennsylvania: Association for Computing Ma-

chinery, 2009. ISBN: 9781605584089. DOI: 10.1145/1555619.1555645. URL: https://doi.org/10.1145/1555619.1555645.

[54] Michael Coblenz et al. "Can advanced type systems be usable? An empirical study of ownership, assets, and typestate in Obsidian". In: *Proc. ACM Program. Lang.* 4.OOPSLA (Nov. 2020). DOI: 10.1145/3428200. URL: https://doi.org/10.1145/3428200.

[55] Michael Coblenz et al. "Smarter Smart Contract Development Tools". In: *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*. 2019, pp. 48–51. DOI: 10.1109/WETSEB.2019.00013.

[56] CoinMarketCap. *Coin Market Cap*. [Accessed 16-11-2023]. 2023. URL: https://coinmarketcap.com/currencies/ethereum/.

[57] European Commission. *Consumer Contract Law*. [Accessed 11-Oct-2023]. 2023. URL: https://commission.europa.eu/law/law-topic/consumer-protection-law/consumer-contract-law_en.

[58] Association for Computing Machinery. *Statement on Principles for Responsible Algorithmic Systems*. 1701 Pennsylvania Ave NW, Suite 200 Washington: Association for Computing Machinery, 2022. URL: https://www.acm.org/articles/bulletins/2022/november/tpc-statement-responsible-algorithmic-systems.

[59] Kristin Cornelius. "Betraying Blockchain: Accountability, Transparency and Document Standards for Non-Fungible Tokens (NFTs)". In: *Information* 12.9 (2021). ISSN: 2078-2489. DOI: 10.3390/info12090358. URL: https://www.mdpi.com/2078-2489/12/9/358.

[60] Barnaby Craggs and Awais Rashid. "Trust Beyond Computation Alone: Human Aspects of Trust in Blockchain Technologies". In: *2019 IEEE/ACM 41st International*

Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS). 2019, pp. 21–30. DOI: 10.1109/ICSE-SEIS.2019.00011.

[61] Stefan Craß et al. "Collaborative Administration of Role-Based Access Control in Smart Contracts". In: *2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. 2022, pp. 87–94. DOI: 10.1109/BRAINS55737.2022.9909116.

[62] Lars Creutz, Jens Schneider, and Guido Dartmann. "Fides: Distributed Cyber-Physical Contracts". In: *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. 2021, pp. 51–60. DOI: 10.1109/TPSISA52974.2021.00006.

[63] Daniela S. Cruzes and Tore Dyba. "Recommended Steps for Thematic Synthesis in Software Engineering". In: *2011 International Symposium on Empirical Software Engineering and Measurement*. 2011, pp. 275–284. DOI: 10.1109/ESEM.2011.36.

[64] Gabriel R De En Goh. "Smart Contract Disputes and Public Policy in the ASEAN+ 6 Region". In: *Digital LJ* 3 (2022), p. 32.

[65] Nicholas Diakopoulos et al. *Principles for accountable algorithms and a social impact statement for algorithms*. URL: https://www.fatml.org/resources/principles-for-accountable-algorithms.

[66] Edsger W. Dijkstra. "On the Role of Scientific Thought". In: *Selected Writings on Computing: A personal Perspective*. New York, NY: Springer New York, 1982, pp. 60–66. ISBN: 978-1-4612-5695-3. DOI: 10.1007/978-1-4612-5695-3_12. URL: https://doi.org/10.1007/978-1-4612-5695-3_12.

[67] Weiping Ding et al. "Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey". In: *Information Sciences* 615 (2022), pp. 238–

292. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.10.013. URL: https://www.sciencedirect.com/science/article/pii/S002002552201132X.

[68] Yan Ding et al. "Blockchain-based Access Control Mechanism of Federated Data Sharing System". In: *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. 2020, pp. 277–284. DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00060.

[69] Yi Ding et al. "SC-RBAC: A Smart Contract based RBAC Model for DApps". In: *Human Centered Computing*. Ed. by Danijela Milošević, Yong Tang, and Qiaohong Zu. Cham: Springer International Publishing, 2019, pp. 75–85. ISBN: 978-3-030-37429-7.

[70] Abhishek Dixit et al. "Towards user-centered and legally relevant smart-contract development: A systematic literature review". In: *Journal of Industrial Information Integration* 26 (2022), p. 100314. ISSN: 2452-414X. DOI: https://doi.org/10.1016/j.jii.2021.100314. URL: https://www.sciencedirect.com/science/article/pii/S2452414X21001072.

[71] A. Donmez and A. Karaivanov. "Transaction Fee Economics in the Ethereum Blockchain". In: *Economic Inquiry* 60 (1 2021), pp. 265–292. DOI: 10.1111/ecin.13025.

[72] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. "Explainable Artificial Intelligence: A Survey". In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2018, pp. 0210–0215. DOI: 10.23919/MIPRO.2018.8400040.

[73] Daniel Drummer and Dirk Neumann. "Is code law? Current legal and technical adoption issues and remedies for blockchain-enabled smart contracts". In: *Journal of Information Technology* 35.4 (2020), pp. 337–360. DOI: 10.1177/0268396220924669.

eprint: https://doi.org/10.1177/0268396220924669. URL: https://doi.org/10.1177/0268396220924669.

[74] Yanqing Duan, John S. Edwards, and Yogesh K Dwivedi. "Artificial intelligence for decision making in the era of Big Data – evolution, challenges and research agenda". In: *International Journal of Information Management* 48 (2019), pp. 63–71. ISSN: 0268-4012. DOI: https://doi.org/10.1016/j.ijinfomgt.2019.01.021. URL: https://www.sciencedirect.com/science/article/pii/S0268401219300581.

[75] Rudresh Dwivedi et al. "Explainable AI (XAI): Core Ideas, Techniques, and Solutions". In: *ACM Comput. Surv.* 55.9 (Jan. 2023). ISSN: 0360-0300. DOI: 10.1145/3561048. URL: https://doi.org/10.1145/3561048.

[76] Vimal Dwivedi et al. "A Formal Specification Smart-Contract Language for Legally Binding Decentralized Autonomous Organizations". In: *IEEE Access* 9 (2021), pp. 76069–76082. DOI: 10.1109/ACCESS.2021.3081926.

[77] Vimal Dwivedi et al. "Legally Enforceable Smart-Contract Languages: A Systematic Literature Review". In: *ACM Comput. Surv.* 54.5 (June 2021). ISSN: 0360-0300. DOI: 10.1145/3453475. URL: https://doi.org/10.1145/3453475.

[78] Helen Eenmaa-Dimitrieva and Maria José Schmidt-Kessen. "Creating markets in no-trust environments: The law and economics of smart contracts". In: *Computer Law & Security Review* 35.1 (2019), pp. 69–88. ISSN: 0267-3649. DOI: https://doi.org/10.1016/j.clsr.2018.09.003. URL: https://www.sciencedirect.com/science/article/pii/S0267364918303558.

[79] Chris Elsden et al. "Making Sense of Blockchain Applications: A Typology for HCI". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. , Montreal QC, Canada, Association for Computing Machinery, 2018, pp. 1–14. ISBN: 9781450356206. DOI: 10.1145/3173574.3174032. URL: https://doi.org/10.1145/3173574.3174032.

[80] Mica R Endsley. "Situation awareness in future autonomous vehicles: Beware of the unexpected". In: *Congress of the International Ergonomics Association*. Springer. 2018, pp. 303–309.

[81] Mica R Endsley. "Situation awareness misconceptions and misunderstandings". In: *Journal of Cognitive Engineering and Decision Making* 9.1 (2015), pp. 4–32.

[82] Mica R Endsley. "Toward a theory of situation awareness in dynamic systems". In: *Human factors* 37.1 (1995), pp. 32–64.

[83] Mica R. Endsley. *Designing for Situation Awareness: An Approach to User-Centered Design, Second Edition*. 2nd. USA: CRC Press, Inc., 2011. ISBN: 1420063553.

[84] Mica R. Endsley. "Measurement of Situation Awareness in Dynamic Systems". In: *Human Factors* 37.1 (1995), pp. 65–84. DOI: 10.1518/001872095779049499. eprint: https://doi.org/10.1518/001872095779049499. URL: https://doi.org/10.1518/001872095779049499.

[85] Mica R. Endsley. "Supporting Human-AI Teams: Transparency, explainability, and situation awareness". In: *Computers in Human Behavior* 140 (2023), p. 107574. ISSN: 0747-5632. DOI: https://doi.org/10.1016/j.chb.2022.107574. URL: https://www.sciencedirect.com/science/article/pii/S0747563222003946.

[86] Mica R. Endsley and Mark D. Rodgers. "Situation Awareness Information Requirements Analysis for En Route Air Traffic Control". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 38.1 (1994), pp. 71–75. DOI: 10.1177/154193129403800113. eprint: https://doi.org/10.1177/154193129403800113. URL: https://doi.org/10.1177/154193129403800113.

[87] ENS. *Ethereum Name Service*. [Accessed 29-June-2024]. 2024. URL: https://ens.domains.

[88] Jörn Erbguth and Jean-Henry Morin. "Towards Governance and Dispute Resolution for DLT and Smart Contracts". In: *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. 2018, pp. 46–55. DOI: 10.1109/ICSESS.2018.8663721.

[89] Ethereum. *Ethereum Development Documentation*. [Accessed 15-May-2023]. 2023. URL: https://ethereum.org/en/developers/docs/.

[90] Etherscan. *Ethereum Average Gas Price Chart*. [Accessed 16-Nov-2023]. 2023. URL: https://etherscan.io/chart/gasprice.

[91] Etherscan. *The Ethereum Blockchain Explorer*. [Accessed 15-Nov-2023]. 2023. URL: https://etherscan.io.

[92] David Evans. *Systematizing systematization of knowledge*. URL: https://oaklandsok.github.io/.

[93] Andy Extance. "Bitcoin and beyond". In: *Nature* 526.7571 (2015), p. 21.

[94] Lesley Fair. *Reported crypto scam losses since 2021 top $1 billion, says FTC Data spotlight*. Aug. 2022. URL: https://www.ftc.gov/business-guidance/blog/2022/06/reported-crypto-scam-losses-2021-top-1-billion-says-ftc-data-spotlight.

[95] Yuzhou Fang et al. "Beyond "Protected" and "Private": An Empirical Security Analysis of Custom Function Modifiers in Smart Contracts". In: *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*. ISSTA 2023. Seattle, WA, USA, Association for Computing Machinery, 2023, pp. 1157–1168. ISBN: 9798400702211. DOI: 10.1145/3597926.3598125. URL: https://doi.org/10.1145/3597926.3598125.

[96] Youssef Faqir-Rhazoui et al. "Effect of the Gas Price Surges on User Activity in the DAOs of the Ethereum Blockchain". In: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI EA '21. , Yokohama,

Japan, Association for Computing Machinery, 2021. ISBN: 9781450380959. DOI: 10.1145/3411763.3451755. URL: https://doi.org/10.1145/3411763.3451755.

[97] E Allen Farnsworth et al. *Contracts: Cases and Materials*. Foundation Press, 2013.

[98] Alberto Fernandez et al. "Evolutionary Fuzzy Systems for Explainable Artificial Intelligence: Why, When, What for, and Where to?" In: *IEEE Computational Intelligence Magazine* 14.1 (2019), pp. 69–81. DOI: 10.1109/MCI.2018.2881645.

[99] Agata Ferreira. "Regulating Smart Contracts: Legal Revolution or Simply Evolution?" In: *Telecommunications Policy* 45.2 (2021), p. 102081. ISSN: 0308-5961. DOI: https://doi.org/10.1016/j.telpol.2020.102081. URL: https://www.sciencedirect.com/science/article/pii/S0308596120301713.

[100] Michèle Finck. *Blockchain regulation and governance in Europe*. Cambridge University Press, 2018.

[101] Michèle Finck. "Smart contracts as a form of solely automated processing under the GDPR". In: *International Data Privacy Law* 9.2 (May 2019), pp. 78–94. ISSN: 2044-3994. DOI: 10.1093/idpl/ipz004. eprint: https://academic.oup.com/idpl/article-pdf/9/2/78/29101514/ipz004.pdf. URL: https://doi.org/10.1093/idpl/ipz004.

[102] J. M. Flach. "Situation awareness". In: *Journal of Cognitive Engineering and Decision Making* 9 (1 2015), pp. 59–72. DOI: 10.1177/1555343414561087.

[103] Meadhbh I Foster and Mark T Keane. "The role of surprise in learning: Different surprising outcomes affect memorability differentially". In: *Topics in cognitive science* 11.1 (2019), pp. 75–87.

[104] Meadhbh I. Foster and Mark T. Keane. "Predicting Surprise Judgments from Explanation Graphs". In: Apr. 2015.

[105] Meadhbh I. Foster and Mark T. Keane. "Surprise! You've Got Some Explaining to Do". In: *ArXiv* abs/1308.2236 (2013). URL: https://api.semanticscholar.org/CorpusID:14226666.

[106] Meadhbh I. Foster and Mark T. Keane. "Why Some Surprises Are More Surprising Than Others: Surprise As A Metacognitive Sense of Explanatory Difficulty". In: *Cognitive Psychology* 81 (2015), pp. 74–116. ISSN: 0010-0285. DOI: https://doi.org/10.1016/j.cogpsych.2015.08.004. URL: https://www.sciencedirect.com/science/article/pii/S0010028515000626.

[107] Maker Foundation. *The Maker Protocol: MakerDAO's Multi-Collateral DAI (MCD) System*. [Accessed 26-May-2023]. URL: https://makerdao.com/en/whitepaper#notes.

[108] Christopher K. Frantz and Mariusz Nowostawski. "From Institutions to Code: Towards Automated Generation of Smart Contracts". In: *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. 2016, pp. 210–215. DOI: 10.1109/FAS-W.2016.53.

[109] Felix Franz et al. "Towards Human-readable Smart Contracts". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2019, pp. 38–42. DOI: 10.1109/BLOC.2019.8751309.

[110] Frax. *Frax Finance Stablecoin Protocol*. Tech. rep. 2022. URL: https://docs.frax.finance.

[111] Michael Froehlich, Philipp Hulm, and Florian Alt. "Under Pressure. A User-Centered Threat Model for Cryptocurrency Owners". In: *Proceedings of the 2021 4th International Conference on Blockchain Technology and Applications*. ICBTA '21. Xi'an, China: Association for Computing Machinery, 2022, pp. 39–50. ISBN: 9781450387460. DOI: 10.1145/3510487.3510494. URL: https://doi.org/10.1145/3510487.3510494.

[112] Michael Froehlich et al. “Blockchain and Cryptocurrency in Human Computer Interaction: A Systematic Literature Review and Research Agenda”. In: *Proceedings of the 2022 ACM Designing Interactive Systems Conference*. DIS ’22. , Virtual Event, Australia, Association for Computing Machinery, 2022, pp. 155–177. ISBN: 9781450393584. DOI: 10.1145/3532106.3533478. URL: https://doi.org/10.1145/3532106.3533478.

[113] Michael Froehlich et al. “Don’t Stop Me Now! Exploring Challenges Of First-Time Cryptocurrency Users”. In: *Proceedings of the 2021 ACM Designing Interactive Systems Conference*. DIS ’21. Virtual Event, USA: Association for Computing Machinery, 2021, pp. 138–148. ISBN: 9781450384766. DOI: 10.1145/3461778.3462071. URL: https://doi.org/10.1145/3461778.3462071.

[114] Michael Fröwis and Rainer Böhme. “Detecting Privileged Parties on Ethereum”. In: *Financial Cryptography and Data Security. FC 2023 International Workshops*. Ed. by Aleksander Essex et al. Cham: Springer Nature Switzerland, 2024, pp. 470–488. ISBN: 978-3-031-48806-1.

[115] Sanil S. Gandhi et al. “Usability Analysis for Blockchain-Based Applications”. In: *Intelligent Human Computer Interaction*. Ed. by Jong-Hoon Kim et al. Cham: Springer International Publishing, 2022, pp. 349–360. ISBN: 978-3-030-98404-5.

[116] Mark Giancaspro. “Is a ‘smart contract’ really a smart idea? Insights from a legal perspective”. In: *Computer Law & Security Review* 33.6 (2017), pp. 825–835. ISSN: 0267-3649. DOI: https://doi.org/10.1016/j.clsr.2017.05.007. URL: https://www.sciencedirect.com/science/article/pii/S026736491730167X.

[117] Jack Gilcrest and Arthur Carvalho. “Smart Contracts: Legal Considerations”. In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 3277–3281. DOI: 10.1109/BigData.2018.8622584.

[118] R.L. Glass, I. Vessey, and V. Ramesh. “Research in software engineering: an analysis of the literature”. In: *Information and Software Technology* 44.8 (2002), pp. 491–

506. ISSN: 0950-5849. DOI: https://doi.org/10.1016/S0950-5849(02)00049-6. URL: https://www.sciencedirect.com/science/article/pii/S0950584902000496.

[119] Robert L Glass. "A comparative analysis of the topic areas of computer science, software engineering, and information systems". In: *Journal of Systems and Software* 19.3 (1992), pp. 277–289. ISSN: 0164-1212. DOI: https://doi.org/10.1016/0164-1212(92)90056-P. URL: https://www.sciencedirect.com/science/article/pii/016412129290056P.

[120] Cristian Gómez et al. "Easing the use of smart contracts through model-based engineering". In: *2022 IEEE 24th Conference on Business Informatics (CBI)*. Vol. 02. 2022, pp. 126–132. DOI: 10.1109/CBI54897.2022.10058.

[121] Guido Governatori et al. "On legal contracts, imperative and declarative smart contracts, and blockchain systems". In: *Artificial Intelligence and Law* 26 (2018), pp. 377–409.

[122] Gratzki. *Decentralized Application updates and governance*. [Accessed 26-May-2023]. URL: https://medium.com/@gratzkis/decentralized-application-dapp-updates-and-governance-831f33d8368a.

[123] Bart N. Green, Claire D. Johnson, and Alan Adams. "Writing narrative literature reviews for peer-reviewed journals: secrets of the trade". In: *Journal of Chiropractic Medicine* 5.3 (2006), pp. 101–117. ISSN: 1556-3707. DOI: https://doi.org/10.1016/S0899-3467(07)60142-6. URL: https://www.sciencedirect.com/science/article/pii/S0899346707601426.

[124] Shirley Gregor and Izak Benbasat. "Explanations from Intelligent Systems: Theoretical Foundations and Implications for Practice". In: *MIS quarterly* (1999), pp. 497–530.

[125] Robert Wayne Gregory. "Design Science Research and the Grounded Theory Method: Characteristics, Differences, and Complementary Uses". In: *Theory-Guided Modeling and Empiricism in Information Systems Research*. Ed. by Armin Heinzl et al. Heidelberg: Physica-Verlag HD, 2011, pp. 111–127. ISBN: 978-3-7908-2781-1. DOI: 10.1007/978-3-7908-2781-1_6. URL: https://doi.org/10.1007/978-3-7908-2781-1_6.

[126] David Gunning. "DARPA's explainable artificial intelligence (XAI) program". In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. IUI '19. Marina del Ray, California: Association for Computing Machinery, 2019, p. ii. ISBN: 9781450362726. DOI: 10.1145/3301275.3308446. URL: https://doi.org/10.1145/3301275.3308446.

[127] Mohammad Hamdaqa, Lucas Alberto Pineda Met, and Ilham Qasse. "iContractML 2.0: A domain-specific language for modeling and deploying smart contracts onto multiple blockchain platforms". In: *Information and Software Technology* 144 (2022), p. 106762. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2021.106762. URL: https://www.sciencedirect.com/science/article/pii/S0950584921002081.

[128] Mohammad Hamdaqa, Lucas Alberto Pineda Metz, and Ilham Qasse. "IContractML: A domain-specific language for modeling and deploying smart contracts onto multiple blockchain platforms". In: *Proceedings of the 12th System Analysis and Modelling Conference*. 2020, pp. 34–43.

[129] Sicheng Hao et al. "SmartCoCo: Checking Comment-Code Inconsistency in Smart Contracts via Constraint Propagation and Binding". In: *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 2023, pp. 294–306. DOI: 10.1109/ASE56229.2023.00142.

[130] Zhihao Hao et al. "A novel method using LSTM-RNN to generate smart contracts code templates for improved usability". In: *Multimedia Tools and Applications* (2023), pp. 1–31.

[131] Akm Bahalul Haque et al. "GDPR Compliant Blockchains–A Systematic Literature Review". In: *IEEE Access* 9 (2021), pp. 50593–50606. DOI: 10.1109/ACCESS.2021.3069877.

[132] Vikas Hassija et al. "Interpreting black-box models: a review on explainable artificial intelligence". In: *Cognitive Computation* 16.1 (2024), pp. 45–74.

[133] Philip J. Hayes and D. Raj Reddy. "Steps Toward Graceful Interaction in Spoken and Written Man-Machine Communication". In: *International Journal of Man-Machine Studies* 19.3 (1983), pp. 231–284. ISSN: 0020-7373. DOI: https://doi.org/10.1016/S0020-7373(83)80049-2. URL: https://www.sciencedirect.com/science/article/pii/S0020737383800492.

[134] Xiao He et al. "SPESC: A Specification Language for Smart Contracts". In: *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. Vol. 01. 2018, pp. 132–137. DOI: 10.1109/COMPSAC.2018.00025.

[135] Alan Hevner and Samir Chatterjee. "Design Science Research in Information Systems". In: *Design Research in Information Systems: Theory and Practice*. Boston, MA: Springer US, 2010, pp. 9–22. ISBN: 978-1-4419-5653-8. DOI: 10.1007/978-1-4419-5653-8_2. URL: https://doi.org/10.1007/978-1-4419-5653-8_2.

[136] Tharaka Hewa, Mika Ylianttila, and Madhusanka Liyanage. "Survey on blockchain based smart contracts: Applications, opportunities and challenges". In: *Journal of Network and Computer Applications* 177 (2021), p. 102857. ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2020.102857. URL: https://www.sciencedirect.com/science/article/pii/S1084804520303234.

[137] Tharaka Mawanane Hewa et al. "Survey on Blockchain-Based Smart Contracts: Technical Aspects and Future Research". In: *IEEE Access* 9 (2021), pp. 87643–87662. DOI: 10.1109/ACCESS.2021.3068178.

[138] Eduard Hofer et al. "An Approximate Epistemic Uncertainty Analysis Approach in the Presence Of Epistemic And Aleatory Uncertainties". In: *Reliability Engineering & System Safety* 77.3 (2002), pp. 229–238. ISSN: 0951-8320. DOI: https://doi.org/10.1016/S0951-8320(02)00056-X. URL: https://www.sciencedirect.com/science/article/pii/S095183200200056X.

[139] Ying-Ying Hsieh, Jean-Philippe JP Vergne, and Sha Wang. "The internal and external governance of blockchain-based organizations: Evidence from cryptocurrencies". In: *Bitcoin and beyond*. Routledge, 2017, pp. 48–68.

[140] Xing Hu et al. "Automating User Notice Generation for Smart Contract Functions". In: *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 2021, pp. 5–17. DOI: 10.1109/ASE51524.2021.9678552.

[141] Junqin Huang et al. "Advancing Web 3.0: Making Smart Contracts Smarter on Blockchain". In: *Proceedings of the ACM on Web Conference 2024*. WWW '24. , Singapore, Singapore, Association for Computing Machinery, 2024, pp. 1549–1560. ISBN: 9798400701719. DOI: 10.1145/3589334.3645319. URL: https://doi.org/10.1145/3589334.3645319.

[142] Trung Dong Huynh et al. *A Methodology and Software Architecture to Support Explainability-by-Design*. Tech. rep. UK Engineering and Physical Sciences Research Council (EPSRC) for the PLEAD project, 2022. URL: https://api.semanticscholar.org/CorpusID:258947830.

[143] IBM. *What is explainable AI (XAI)?* Apr. 2024. URL: https://www.ibm.com/topics/explainable-ai.

[144] ICO. *Data protection by design and default*. Tech. rep. Information Commissioners Office, 2018. URL: https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/accountability-and-governance/guide-to-accountability-and-governance/accountability-and-governance/data-protection-by-design-and-default/.

[145] ICO. *Rights related to automated decision making including profiling*. Tech. rep. Information Commissioner's Office, 2024. URL: https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/individual-rights/individual-rights/rights-related-to-automated-decision-making-including-profiling/.

[146] ISO. *ISO 22739: Blockchain and distributed ledger technologies — Vocabulary*. Tech. rep. ISO - International Organization for Standardization, 2024. URL: https://www.iso.org/standard/82208.html.

[147] ISO. *ISO 26000: Guidance on social responsibility*. Tech. rep. ISO - International Organization for Standardization, 2010. URL: https://www.iso.org/standard/42546.html.

[148] ISO. *ISO/IEC 22989: Information technology — Artificial intelligence — Artificial intelligence concepts and terminology*. Tech. rep. ISO - International Organization for Standardization, 2022. URL: https://www.iso.org/standard/74296.html.

[149] ISO. *ISO/IEC 42001 Information technology- Artificial intelligence Management system*. Tech. rep. ISO - International Organization for Standardization, 2023. URL: https://www.iso.org/standard/81230.html.

[150] ISO. *ISO/IEC TS 5723-Trustworthiness — Vocabulary*. Tech. rep. ISO - International Organization for Standardization, 2023. URL: https://www.iso.org/standard/81608.html.

[151] ISO. *ISO/IEC/IEEE 24765: Systems and Software Engineering Vocabulary*. Tech. rep. ISO - International Organization for Standardization, 2017. URL: https://www.iso.org/standard/71952.html.

[152] Nikolay Ivanov, Hanqing Guo, and Qiben Yan. "Rectifying Administrated ERC20 Tokens". In: *Information and Communications Security*. Ed. by Debin Gao et al. Cham: Springer International Publishing, 2021, pp. 22–37. ISBN: 978-3-030-86890-1.

[153] Hyeji Jang, Sung H Han, and Ju Hwan Kim. "User perspectives on blockchain technology: User-centered evaluation and design strategies for dapps". In: *IEEE Access* 8 (2020), pp. 226213–226223.

[154] Kristof Jannes et al. "DEDACS: Decentralized and dynamic access control for smart contracts in a policy-based manner". In: *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. SAC '23. Tallinn, Estonia: Association for Computing Machinery, 2023, pp. 222–230. ISBN: 9781450395175. DOI: 10.1145/3555776.3577676. URL: https://doi.org/10.1145/3555776.3577676.

[155] Jameela Al-Jaroodi and Nader Mohamed. "Blockchain in Industries: A Survey". In: *IEEE Access* 7 (2019), pp. 36500–36515. DOI: 10.1109/ACCESS.2019.2903554.

[156] SoonHyeong Jeong and Byeongtae Ahn. "A study of application platform for smart contract visualization based blockchain". In: *The Journal of Supercomputing* 78.1 (2022), pp. 343–360.

[157] Samantha Tharani Jeyakumar, Ryan Ko, and Vallipuram Muthukkumarasamy. "A Framework for User-Centric Visualisation of Blockchain Transactions in Critical Infrastructure". In: *Proceedings of the 5th ACM International Symposium on Blockchain and Secure Critical Infrastructure*. BSCI '23. Melbourne, VIC, Australia: Association for Computing Machinery, 2023, pp. 44–52. ISBN: 9798400701986. DOI: 10.1145/3594556.3594624. URL: https://doi.org/10.1145/3594556.3594624.

[158] Jeff S Johnson and Ravipreet S Sohi. "Understanding and Resolving Major Contractual Breaches in Buyer–Seller Relationships: A Grounded Theory Approach". In: *Journal of the Academy of Marketing Science* 44 (2016), pp. 185–205.

[159] Eleanna Kafeza et al. "Legal smart contracts in Ethereum Block chain: Linking the dots". In: *2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW)*. 2020, pp. 18–25. DOI: 10.1109/ICDEW49219.2020.00-12.

[160] Daniel Kahneman and Dale T Miller. "Norm theory: Comparing reality to its alternatives." In: *Psychological review* 93.2 (1986), p. 136.

[161] Niclas Kannengießer et al. "Challenges and Common Solutions in Smart Contract Development". In: *IEEE Transactions on Software Engineering* 48.11 (2022), pp. 4291–4318. DOI: 10.1109/TSE.2021.3116808.

[162] Staffs Keele et al. *Guidelines for performing systematic literature reviews in software engineering*. Tech. rep. Technical report, ver. 2.3 EBSE technical report., 2007.

[163] Marcus A. Rothenberger Ken Peffers Tuure Tuunanen and Samir Chatterjee. "A Design Science Research Methodology for Information Systems Research". In: *Journal of Management Information Systems* 24.3 (2007), pp. 45–77. DOI: 10.2753/MIS0742-1222240302. eprint: https://doi.org/10.2753/MIS0742-1222240302. URL: https://doi.org/10.2753/MIS0742-1222240302.

[164] Shafaq Naheed Khan et al. "Blockchain smart contracts: Applications, challenges, and future trends". In: *Peer-to-peer Networking and Applications* 14 (2021), pp. 2901–2925.

[165] Barbara Kitchenham. "Procedures for performing systematic reviews". In: *Keele, UK, Keele University* 33.2004 (2004), pp. 1–26.

[166] Barbara Kitchenham, Stephen Linkman, and David Law. "DESMET: A method for evaluating software engineering methods and tools". In: *Keele University* (1996).

[167] Armen Der Kiureghian and Ove Ditlevsen. "Aleatory or Epistemic? Does It Matter?" In: *Structural Safety* 31.2 (2009). Risk Acceptance and Risk Communication, pp. 105–112. ISSN: 0167-4730. DOI: https://doi.org/10.1016/j.strusafe.2008.06.020. URL: https://www.sciencedirect.com/science/article/pii/S0167473008000556.

[168] Jan Ladleif and Mathias Weske. "A Unifying Model of Legal Smart Contracts". In: *Conceptual Modeling*. Ed. by Alberto H. F. Laender et al. Cham: Springer International Publishing, 2019, pp. 323–337. ISBN: 978-3-030-33223-5.

[169] Metin Lamby, Valentin Zieglmeier, and Christian Ziegler. "Trusting a Smart Contract Means Trusting Its Owners: Understanding Centralization Risk". In: *2023 5th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE. 2023, pp. 1–4.

[170] Ole Lando and Hugh Beale. *Principles of European Contract Law: Parts 1 and 2. Combined nd Revised*. English. Opstilling: 347.4 pri Løbe nr.: 001760. Netherlands: Kluwer Law International, 1999. ISBN: 9041113053.

[171] Markus Langer et al. "What Do We Want From Explainable Artificial Intelligence (XAI)? – A Stakeholder Perspective on XAI and A Conceptual Model Guiding Interdisciplinary XAI Research". In: *Artificial Intelligence* 296 (2021), p. 103473. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2021.103473. URL: https://www.sciencedirect.com/science/article/pii/S0004370221000242.

[172] Shane Larson. *Creating an Ownable Smart Contract in Solidity for Ethereum*. [Accessed 15-May-2023]. URL: https://grizzlypeaksoftware.com/articles?id=6NvaOcWdWhwGEKbI

[173] David B Leake. "Goal-Based Explanation Evaluation". In: *Cognitive Science* 15.4 (1991), pp. 509–545.

[174] Jiewu Leng et al. "ManuChain II: Blockchained Smart Contract System as the Digital Twin of Decentralized Autonomous Manufacturing Toward Resilience in Industry 5.0". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2023), pp. 1–14. DOI: 10.1109/TSMC.2023.3257172.

[175] Michael van Lent, William Fisher, and Michael Mancuso. "An explainable artificial intelligence system for small-unit tactical behavior". In: *Proceedings of the 16th Con-*

ference on Innovative Applications of Artifical Intelligence. IAAI'04. San Jose, California: AAAI Press, 2004, pp. 900–907. ISBN: 0262511835.

[176] Karen Levy. "Book-Smart, Not Street-Smart: Blockchain-Based Smart Contracts and The Social Workings of Law". In: *Engaging Science, Technology, and Society* 3 (Feb. 2017), p. 1. DOI: 10.17351/ests2017.107.

[177] Chao Li, Balaji Palanisamy, and Runhua Xu. "Scalable and Privacy-Preserving Design of On/Off-Chain Smart Contracts". In: *2019 IEEE 35th International Conference on Data Engineering Workshops (ICDEW)*. 2019, pp. 7–12. DOI: 10.1109/ICDEW.2019.00-43.

[178] Lantian Li et al. "Understanding Solidity Event Logging Practices in the Wild". In: *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2023. , San Francisco, CA, USA, Association for Computing Machinery, 2023, pp. 300–312. ISBN: 9798400703270. DOI: 10.1145/3611643.3616342. URL: https://doi.org/10.1145/3611643.3616342.

[179] Shanshan Li et al. "Understanding and addressing quality attributes of microservices architecture: A Systematic literature review". In: *Information and Software Technology* 131 (2021), p. 106449. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2020.106449. URL: https://www.sciencedirect.com/science/article/pii/S0950584920301993.

[180] Xiaosong Li. "Using Peer Review to Assess Coding Standards - A Case Study". In: *Proceedings. Frontiers in Education. 36th Annual Conference*. 2006, pp. 9–14. DOI: 10.1109/FIE.2006.322572.

[181] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. "Why and Why Not Explanations Improve the Intelligibility of Context-Aware Intelligent Systems". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI

'09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 2119–2128. ISBN: 9781605582467. DOI: 10.1145/1518701.1519023. URL: https://doi-org.bham-ezproxy.idm.oclc.org/10.1145/1518701.1519023.

[182] Bowen Liu, Pawel Szalachowski, and Jianying Zhou. "A First Look into DeFi Oracles". In: *2021 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*. 2021, pp. 39–48. DOI: 10.1109/DAPPS52256.2021.00010.

[183] Chao Liu et al. "ReGuard: finding reentrancy bugs in smart contracts". In: *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*. ICSE '18. Gothenburg, Sweden: Association for Computing Machinery, 2018, pp. 65–68. ISBN: 9781450356633. DOI: 10.1145/3183440.3183495. URL: https://doi.org/10.1145/3183440.3183495.

[184] Lu Liu et al. "Characterizing transaction-reverting statements in ethereum smart contracts". In: *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering*. ASE '21. Melbourne, Australia: IEEE Press, 2022, pp. 630–641. ISBN: 9781665403375. DOI: 10.1109/ASE51524.2021.9678597. URL: https://doi.org/10.1109/ASE51524.2021.9678597.

[185] Shuze Liu et al. "Strengthening Smart Contracts to Handle Unexpected Situations". In: *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*. 2019, pp. 182–187. DOI: 10.1109/DAPPCON.2019.00034.

[186] Ye Liu et al. "Towards automated verification of smart contract fairness". In: *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2020. Virtual Event, USA: Association for Computing Machinery, 2020, pp. 666–677. ISBN: 9781450370431. DOI: 10.1145/3368089.3409740. URL: https://doi.org/10.1145/3368089.3409740.

[187] Yue Liu et al. "A systematic literature review on blockchain governance". In: *Journal of Systems and Software* 197 (2023), p. 111576. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2022.111576. URL: https://www.sciencedirect.com/science/article/pii/S0164121222002527.

[188] Emiliano Lorini and Cristiano Castelfranchi. "The Unexpected Aspects of Surprise". In: *International Journal of Pattern Recognition and Artificial Intelligence* 20.06 (2006), pp. 817–833.

[189] Loi Luu et al. "Making Smart Contracts Smarter". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 254–269. ISBN: 9781450341394. DOI: 10.1145/2976749.2978309. URL: https://doi.org/10.1145/2976749.2978309.

[190] Nash Lyke, Benjamin M. Gorman, and Garreth W. Tigwell. "Exploring the Accessibility of Crypto Technologies". In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI EA '23. , Hamburg, Germany, Association for Computing Machinery, 2023. ISBN: 9781450394222. DOI: 10.1145/3544549.3585746. URL: https://doi.org/10.1145/3544549.3585746.

[191] L. Macedo and A. Cardoso. "A Contrast-Based Computational Model of Surprise and its Applications". In: *Topics in Cognitive Science* 11 (1 2017), pp. 88–102. DOI: 10.1111/tops.12310.

[192] Luís Macedo, Rainer Reisezein, and Amilcar Cardoso. "Modeling Forms of Surprise in Artificial Agents: Empirical and Theoretical Study of Surprise Functions". In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 26. 26. 2004.

[193] Daniel Macrinici, Cristian Cartofeanu, and Shang Gao. "Smart contract applications within blockchain technology: A systematic mapping study". In: *Telematics and Informatics* 35.8 (2018), pp. 2337–2354. ISSN: 0736-5853. DOI: https://doi.org/10.

1016/j.tele.2018.10.004. URL: https://www.sciencedirect.com/science/article/pii/S0736585318308013.

[194] P. Maguire et al. "Seeing Patterns in Randomness: A Computational Model of Surprise". In: *Topics in Cognitive Science* 11 (1 2018), pp. 103–118. DOI: 10.1111/tops.12345.

[195] Rebecca Maguire, Fintan Costello, and Mark Keane. "A Cognitive Model of Surprise Judgements". In: *Proceedings of the 28th annual conference of the cognitive science society* (2006), pp. 531–536.

[196] Rebecca Maguire and Mark Keane. "Surprise: Disconfirmed expectations or representation-fit?" In: *Proceedings of the 28th Annual Conference of the Cognitive Science Society* (2006).

[197] Rebecca Maguire, Phil Maguire, and Mark T Keane. "Making Sense of Surprise: An Investigation of the Factors Influencing Surprise Judgments". In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 37.1 (2011), p. 176.

[198] Dianhui Mao et al. "Visual and User-Defined Smart Contract Designing System Based on Automatic Coding". In: *IEEE Access* 7 (2019), pp. 73131–73143. DOI: 10.1109/ACCESS.2019.2920776.

[199] Salvatore T. March and Gerald F. Smith. "Design and natural science research on information technology". In: *Decision Support Systems* 15.4 (1995), pp. 251–266. ISSN: 0167-9236. DOI: https://doi.org/10.1016/0167-9236(94)00041-2. URL: https://www.sciencedirect.com/science/article/pii/0167923694000412.

[200] Bill Marino and Ari Juels. "Setting Standards for Altering and Undoing Smart Contracts". In: *Rule Technologies. Research, Tools, and Applications*. Ed. by Jose Julio Alferes et al. Cham: Springer International Publishing, 2016, pp. 151–166. ISBN: 978-3-319-42019-6.

[201] Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. "The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies". In: *Journal of Biomedical Informatics* 113 (2021), p. 103655. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2020.103655. URL: https://www.sciencedirect.com/science/article/pii/S1532046420302835.

[202] Paul A McDermott. *Contract law*. Bloomsbury Publishing, 2017.

[203] Medium. *A Complete Introduction To On-Chain Governance Pt.1*. [Accessed 15-May-2023]. URL: https://medium.com/nearweek/a-complete-introduction-to-on-chain-governance-pt-1-9cc787f5fdbe.

[204] Leo Mendiboure, Mohamed-Aymen Chalouf, and Francine Krief. *Dynamic identity and access management in the loT: Blockchain-based approach*. English. Cited by: 0. wiley, 2022, pp. 223–241. DOI: 10.1002/9781394156030.ch9.

[205] Christian Meske et al. "Explainable Artificial Intelligence: Objectives, Stakeholders, and Future Research Opportunities". In: *Information Systems Management* 39.1 (2022), pp. 53–63.

[206] William Metcalfe et al. "Ethereum, smart contracts, DApps". In: *Blockchain and Crypt Currency* 77 (2020), pp. 77–93.

[207] Eliza Mik. "Contracts in code?" In: *Law, Innovation and Technology* 13.2 (2021), pp. 478–509. DOI: 10.1080/17579961.2021.1977220. eprint: https://doi.org/10.1080/17579961.2021.1977220. URL: https://doi.org/10.1080/17579961.2021.1977220.

[208] Eliza Mik. "Smart contracts: terminology, technical limitations and real world complexity". In: *Law, Innovation and Technology* 9.2 (2017), pp. 269–300. DOI: 10.1080/17579961.2017.1378468. eprint: https://doi.org/10.1080/17579961.2017.1378468. URL: https://doi.org/10.1080/17579961.2017.1378468.

[209] Tim Miller. “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence* 267 (2019), pp. 1–38. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2018.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0004370218305988.

[210] Ken Miyachi and Tim K. Mackey. “hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design”. In: *Information Processing & Management* 58.3 (2021), p. 102535. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2021.102535. URL: https://www.sciencedirect.com/science/article/pii/S0306457321000431.

[211] Carlos Molina-Jimenez et al. “Implementation of Smart Contracts Using Hybrid Architectures with On and Off–Blockchain Components”. In: *2018 IEEE 8th International Symposium on Cloud and Service Computing (SC2)*. 2018, pp. 83–90. DOI: 10.1109/SC2.2018.00018.

[212] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. “Methods for interpreting and understanding deep neural networks”. In: *Digital Signal Processing* 73 (2018), pp. 1–15. ISSN: 1051-2004. DOI: https://doi.org/10.1016/j.dsp.2017.10.011. URL: https://www.sciencedirect.com/science/article/pii/S1051200417302385.

[213] Tucker Moore, Nathan Marshall, and Eric Burger. “Fortuna: A Novel Staked Voting System for Distributed Pari-Mutuel Gaming”. In: *2022 IEEE International Conference on Blockchain (Blockchain)*. 2022, pp. 244–249. DOI: 10.1109/Blockchain55522.2022.00041.

[214] Dave Murray-Rust et al. “Blockchain and Beyond: Understanding Blockchains Through Prototypes and Public Engagement”. In: 29.5 (2023). ISSN: 1073-0516. DOI: 10.1145/3503462. URL: https://doi.org/10.1145/3503462.

[215] Kelsie Nabben. “Is a "Decentralized Autonomous Organization" a Panopticon? Algorithmic governance as creating and mitigating vulnerabilities in DAOs”. In: *Pro-*

ceedings of the Interdisciplinary Workshop on (de) Centralization in the Internet. IWCI'21. Virtual Event, Germany: Association for Computing Machinery, 2021, pp. 18–25. ISBN: 9781450391382. DOI: 10.1145/3488663.3493791. URL: https://doi.org/10.1145/3488663.3493791.

[216] Minh Vu Nguyen et al. "Blockchain Oracles: Implications for Smart Contracts in Legal Reasoning and Addressing the Oracle Problem". In: *Proceedings of the 12th International Symposium on Information and Communication Technology*. SOICT '23. , Ho Chi Minh, Vietnam, Association for Computing Machinery, 2023, pp. 296–303. ISBN: 9798400708916. DOI: 10.1145/3628797.3628870. URL: https://doi.org/10.1145/3628797.3628870.

[217] Robert C Nickerson, Upkar Varshney, and Jan Muntermann. "A method for taxonomy development and its application in information systems". In: *European Journal of Information Systems* 22.3 (2013), pp. 336–359.

[218] Lina Nimer and Ashraf Tahat. "Implementation of a Peer-to-Peer Network Using Blockchain to Manage and Secure Electronic Medical Records". In: *2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. 2021, pp. 187–192. DOI: 10.1109/JEEIT53412.2021.9634102.

[219] Bettina Nissen et al. "GeoCoin: Supporting Ideation and Collaborative Design with Smart Contracts". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. , Montreal QC, Canada, Association for Computing Machinery, 2018, pp. 1–10. ISBN: 9781450356206. DOI: 10.1145/3173574.3173737. URL: https://doi.org/10.1145/3173574.3173737.

[220] NIST. *AI risks and trustworthiness*. Tech. rep. National Institute of Standards and Technology, 2023. URL: https://airc.nist.gov/AI_RMF_Knowledge_Base/AI_RMF/Foundational_Information/3-sec-characteristics.

[221] Nvivo. *Nvivo: A Software for Qualitative Data Analysis*. 2023. URL: https://help-nv.qsrinternational.com/20/win/Content/welcome.htm.

[222] Gustavo A. Oliva and Ahmed E. Hassan. "The gas triangle and its challenges to the development of blockchain-powered applications". In: *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ESEC/FSE 2021. Athens, Greece: Association for Computing Machinery, 2021, pp. 1463–1466. ISBN: 9781450385626. DOI: 10.1145/3468264.3473130. URL: https://doi.org/10.1145/3468264.3473130.

[223] Emanuel Onica and Marius Georgică. "Can Smart Contracts Become Smart? An Overview of Transaction Impact on Ethereum DApp Engineering". In: *Proceedings of the 4th International Workshop on Distributed Infrastructure for the Common Good*. DICG '23. , Bologna, Italy, Association for Computing Machinery, 2024, pp. 31–36. ISBN: 9798400704581. DOI: 10.1145/3631310.3633492. URL: https://doi.org/10.1145/3631310.3633492.

[224] OpenZepplin. *OpenZeppelin Documentation*. [Accessed 15-May-2023]. 2023. URL: https://docs.openzeppelin.com.

[225] Jonas Oppenlaender. "The Perception of Smart Contracts for Governance of the Metaverse". In: *Proceedings of the 25th International Academic Mindtrek Conference*. Academic Mindtrek '22. Tampere, Finland: Association for Computing Machinery, 2022, pp. 1–8. ISBN: 9781450399555. DOI: 10.1145/3569219.3569300. URL: https://doi-org.bham-ezproxy.idm.oclc.org/10.1145/3569219.3569300.

[226] *Pact: Solving Smart Contract Governance and Upgradeability*. [Accessed 15-May-2023]. URL: https://medium.com/kadena-io/pact-solving-smart-contract-governance-and-upgradeability-976aac3bbb31.

[227] Yu Pan et al. "Automated Generation of Security-Centric Descriptions for Smart Contract Bytecode". In: *Proceedings of the 32nd ACM SIGSOFT International Sym-*

posium on Software Testing and Analysis. ISSTA 2023. , Seattle, WA, USA, Association for Computing Machinery, 2023, pp. 1244–1256. ISBN: 9798400702211. DOI: 10.1145/3597926.3598132. URL: https://doi.org/10.1145/3597926.3598132.

[228] Paralink. *Paralink Network*. [Accessed 26-May-2023]. 2023. URL: https://paralink.network.

[229] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. "Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs". In: *Journal of cognitive engineering and decision making* 2.2 (2008), pp. 140–160.

[230] Reza M. Parizi, Amritraj, and Ali Dehghantanha. "Smart Contract Programming Languages on Blockchains: An Empirical Evaluation of Usability and Security". In: *Blockchain – ICBC 2018*. Ed. by Shiping Chen, Harry Wang, and Liang-Jie Zhang. Cham: Springer International Publishing, 2018, pp. 75–91. ISBN: 978-3-319-94478-4.

[231] Amirmohammad Pasdar, Young Choon Lee, and Zhongli Dong. "Connect API with Blockchain: A Survey on Blockchain Oracle Implementation". In: *ACM Comput. Surv.* 55.10 (Feb. 2023). ISSN: 0360-0300. DOI: 10.1145/3567582. URL: https://doi.org/10.1145/3567582.

[232] Dylan Paulin et al. "Histotrust: tracing AI behavior with secure hardware and blockchain technology". In: *Annals of Telecommunications* (2023), pp. 1–15.

[233] Ken Peffers et al. "A design science research methodology for information systems research". In: *Journal of management information systems* 24.3 (2007), pp. 45–77.

[234] Rowan van Pelt et al. "Defining Blockchain Governance: A Framework for Analysis and Comparison". In: *Information Systems Management* 38.1 (2021), pp. 21–41. DOI: 10.1080/10580530.2020.1720046. eprint: https://doi.org/10.1080/10580530.2020.1720046. URL: https://doi.org/10.1080/10580530.2020.1720046.

[235] Kai Petersen et al. “Systematic mapping studies in software engineering”. In: *12th international conference on evaluation and assessment in software engineering (EASE)*. BCS Learning & Development. 2008.

[236] David Evans PI. “NSF/IARPA/NSA Workshop on the Science of Security”. In: *the IEEE Symposium on Security and Privacy*. 2008.

[237] Pawel Pinio, Roman Batko, and Dagmara Lewicka. “Between Theory and Value Transactions: A Multifaceted Exploration of Relevance and Resilience of Decentralised Autonomous Organisations”. In: *Proceedings of the 2024 7th International Conference on Software Engineering and Information Management*. ICSIM ’24. , Suva, Fiji, Association for Computing Machinery, 2024, pp. 42–48. ISBN: 9798400709197. DOI: 10.1145/3647722.3647729. URL: https://doi.org/10.1145/3647722.3647729.

[238] Andrea Pinna et al. “A Massive Analysis of Ethereum Smart Contracts Empirical Study and Code Metrics”. In: *IEEE Access* 7 (2019), pp. 78194–78213. DOI: 10.1109/ACCESS.2019.2921936.

[239] Eugenia Politou et al. “Blockchain Mutability: Challenges and Proposed Solutions”. In: *IEEE Transactions on Emerging Topics in Computing* 9.4 (2021), pp. 1972–1986. DOI: 10.1109/TETC.2019.2949510.

[240] Education Portal. *Off-chain governance*. [Accessed 15-May-2023]. 2018. URL: https://education.district0x.io/general-topics/what-is-governance/off-chain-governance/.

[241] Nicolas Prat, Isabelle Comyn-Wattiau, and Jacky Akoka. “A Taxonomy of Evaluation Methods for Information Systems Artifacts”. In: *Journal of Management Information Systems* 32.3 (2015), pp. 229–267. DOI: 10.1080/07421222.2015.1099390. eprint: https://doi.org/10.1080/07421222.2015.1099390. URL: https://doi.org/10.1080/07421222.2015.1099390.

[242] Alun Preece. "Asking 'Why'in AI: Explainability of Intelligent Systems–Perspectives and Challenges". In: *Intelligent Systems in Accounting, Finance and Management* 25.2 (2018), pp. 63–72.

[243] Peng Qin et al. "Intelligible description language contract (IDLC)–A novel smart contract model". In: *Information Systems Frontiers* (2021), pp. 1–18.

[244] Abdullah Ramdhani, Muhammad Ali Ramdhani, and Abdusy Syakur Amin. "Writing a Literature Review Research Paper: A step-by-step approach". In: *International Journal of Basic and Applied Science* 3.1 (2014), pp. 47–56.

[245] Peiyun Ran et al. "Automatic Smart Contract Generation with Knowledge Extraction and Unified Modeling Language". In: *International Conference on Smart Computing and Communication*. Springer. 2022, pp. 461–474.

[246] Priya Ranganathan and Rakesh Aggarwal. "Study designs: Part 3-Analytical observational studies". In: *Perspectives in clinical research* 10.2 (2019), pp. 91–94.

[247] Aidin Rasti et al. "Symboleo2SC: from legal contract specifications to smart contracts". In: *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems*. MODELS '22. Montreal, Quebec, Canada: Association for Computing Machinery, 2022, pp. 300–310. ISBN: 9781450394666. DOI: 10.1145/3550355.3552407. URL: https://doi.org/10.1145/3550355.3552407.

[248] Atul Rawal et al. "Recent Advances in Trustworthy Explainable Artificial Intelligence: Status, Challenges, and Perspectives". In: *IEEE Transactions on Artificial Intelligence* 3.6 (2022), pp. 852–866. DOI: 10.1109/TAI.2021.3133846.

[249] Samuel T. Redwine and William E. Riddle. "Software Technology Maturation". In: *Proceedings of the 8th International Conference on Software Engineering*. ICSE '85. London, England: IEEE Computer Society Press, 1985, pp. 189–200. ISBN: 0818606207.

[250] Emanuel Regnath and Sebastian Steinhorst. "SmaCoNat: Smart Contracts in Natural Language". In: *2018 Forum on Specification and Design Languages (FDL)*. 2018, pp. 5–16. DOI: 10.1109/FDL.2018.8524068.

[251] European Union Regulation. *Data Protection Regulation (GDPR)*. 2022. URL: https://gdpr-info.eu/.

[252] Wajiha Rehman et al. "NFTs: Applications and Challenges". In: *2021 22nd International Arab Conference on Information Technology (ACIT)*. 2021, pp. 1–7. DOI: 10.1109/ACIT53391.2021.9677260.

[253] R. Reisenzein, G. Horstmann, and A. Schützwohl. "The Cognitive-Evolutionary Model of Surprise: A Review of the Evidence". In: *Topics in Cognitive Science* 11 (1 2017), pp. 50–74. DOI: 10.1111/tops.12292.

[254] Marten Risius and Kai Spohrer. "A blockchain research framework: What we (don't) know, where we go from here, and how we will get there". In: *Business and information systems engineering* 59 (2017), pp. 385–409.

[255] Bruno Rodrigues et al. "On trust, blockchain, and reputation systems". In: *Handbook on blockchain*. Springer, 2022, pp. 299–337.

[256] Avi Rosenfeld and Ariella Richardson. "Explainability in Human–Agent Systems". In: *Autonomous Agents and Multi-Agent Systems* 33 (2019), pp. 673–705.

[257] Mary Beth Rosson and John M Carroll. "Scenario Based Design". In: *Human-computer interaction. boca raton, FL* (2009), pp. 145–162.

[258] Per Runeson and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering". In: *Empirical software engineering* 14 (2009), pp. 131–164.

[259] Mersedeh Sadeghi, Verena Klös, and Andreas Vogelsang. "Cases for Explainable Software Systems: Characteristics and Examples". In: *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE. 2021, pp. 181–187.

[260] Johnny Saldaña. "Coding techniques for quantitative and mixed data". In: *The Routledge Reviewer's Guide to Mixed Methods Analysis* (2021), pp. 151–160.

[261] Mehdi Salehi, Jeremy Clark, and Mohammad Mannan. "Not so Immutable: Upgradeability of Smart Contracts on Ethereum". In: *Financial Cryptography and Data Security. FC 2022 International Workshops*. Ed. by Shin'ichiro Matsuo et al. Cham: Springer International Publishing, 2023, pp. 539–554. ISBN: 978-3-031-32415-4.

[262] Paul M. Salmon et al. "What really is going on? Review of situation awareness models for individuals and teams". In: *Theoretical Issues in Ergonomics Science* 9.4 (2008), pp. 297–323. DOI: 10.1080/14639220701561775. URL: https://doi.org/10.1080/14639220701561775.

[263] N. Sánchez-Gómez et al. "Blockchain Smart Contract Meta-modeling". In: *Journal of Web Engineering* 20.7 (2021), pp. 2059–2080. DOI: 10.13052/jwe1540-9589.2073.

[264] Lindsay Sanneman and Julie A. Shah. "The Situation Awareness Framework for Explainable AI (SAFE-AI) and Human Factors Considerations for XAI Systems". In: *International Journal of Human–Computer Interaction* 38.18-20 (2022), pp. 1772–1788. DOI: 10.1080/10447318.2022.2081282. eprint: https://doi.org/10.1080/10447318.2022.2081282. URL: https://doi.org/10.1080/10447318.2022.2081282.

[265] Carlos Santana and Laura Albareda. "Blockchain and the emergence of Decentralized Autonomous Organizations (DAOs): An integrative model and research agenda". In: *Technological Forecasting and Social Change* 182 (2022), p. 121806. ISSN: 0040-1625. DOI: https://doi.org/10.1016/j.techfore.2022.121806. URL: https://www.sciencedirect.com/science/article/pii/S0040162522003304.

[266] Levy Santiago, Jauberth Weyll Abijaude, and Fabíola Greve. "Gifflar: a framework to generate smart contracts on the fly". In: *Proceedings of the 31st Annual International Conference on Computer Science and Software Engineering*. CASCON '21. Toronto, Canada: IBM Corp., 2021, pp. 214–219.

[267] Sapna and Deepak Prashar. "Analysis on Blockchain Vulnerabilities and Attacks on Wallet". In: *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. 2021, pp. 1515–1521. DOI: 10.1109/ICAC3N53548.2021.9725403.

[268] NB Sarter, David D Woods, and CE Billings. *Automation Surprises*. Vol. 2. New York Wiley, 1997.

[269] Sarwar Sayeed, Hector Marco-Gisbert, and Tom Caira. "Smart Contract: Attacks and Protections". In: *IEEE Access* 8 (2020), pp. 24416–24427. DOI: 10.1109/ACCESS.2020.2970495.

[270] Roger C Schank, Alex Kass, and Christopher K Riesbeck. *Inside Case-Based Explanation*. Psychology Press, 1994.

[271] Fabian Schär. "Decentralized finance: On blockchain-and smart contract-based financial markets". In: *FRB of St. Louis Review* (2021).

[272] Jan Schwiderowski, Asger Balle Pedersen, and Roman Beck. "Crypto Tokens and Token Systems". In: *Information Systems Frontiers* 26.1 (Mar. 2023), pp. 319–332. ISSN: 1387-3326. DOI: 10.1007/s10796-023-10382-w. URL: https://doi.org/10.1007/s10796-023-10382-w.

[273] Ilya Sergey et al. "Safer smart contract programming with Scilla". In: *Proc. ACM Program. Lang.* 3.OOPSLA (Oct. 2019). DOI: 10.1145/3360611. URL: https://doi.org/10.1145/3360611.

[274] Mary Shaw. "What Makes Good Research in Software Engineering?" In: *International Journal on Software Tools for Technology Transfer* 4 (2002), pp. 1–7. URL: https://api.semanticscholar.org/CorpusID:11398153.

[275] Raymond Sheh and Isaac Monteath. "Defining explainable AI for requirements analysis". In: *Artificial Intelligence within the German Informatics Society* 32 (2018), pp. 261–266.

[276] Nicolas Six et al. "A Blockchain-Based Pattern for Confidential and Pseudo-Anonymous Contract Enforcement". In: *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 2020, pp. 1965–1970. DOI: 10.1109/TrustCom50675.2020.00268.

[277] Kip Smith and Peter A Hancock. "Situation awareness is adaptive, externally directed consciousness". In: *Human factors* 37.1 (1995), pp. 137–148.

[278] The Royal Society. *Explainable AI: the basics Policy briefing*. Tech. rep. The Royal Society, 2019. URL: https://royalsociety.org/-/media/policy/projects/explainable-ai/AI-and-interpretability-policy-briefing.pdf.

[279] Solana. *Web3 Infrastructure for Everyone*. [Accessed 15-May-2023]. URL: https://solana.com.

[280] Solidity. *Solidity 0.8.23 documentation*. [Accessed 26-11-2023]. 2023. URL: https://docs.soliditylang.org/en/v0.8.23/?color=light.

[281] Selami Sönmez. "" 11 Steps" Process as a Research Method." In: *Universal Journal of Educational Research* 6.11 (2018), pp. 2597–2603.

[282] Christian Sonnenberg and Jan vom Brocke. "Evaluation Patterns for Design Science Research Artefacts". In: *Practical Aspects of Design Science*. Ed. by Markus Helfert and Brian Donnellan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 71–83. ISBN: 978-3-642-33681-2.

[283] Francesco Sovrano and Fabio Vitali. "An objective metric for Explainable AI: How and why to estimate the degree of explainability". In: *Knowledge-Based Systems* 278 (2023), p. 110866. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2023.110866. URL: https://www.sciencedirect.com/science/article/pii/S0950705123006160.

[284] Fabian Sparbrodt and Marisol García-Valls. "Digesting smart contracts in Ethereum blockchain networks". In: *2022 5th Conference on Cloud and Internet of Things (CIoT)*. 2022, pp. 60–66. DOI: 10.1109/CIoT53061.2022.9766685.

[285] Stephanie. *Cohen's Kappa Statistic*. [Accessed 15-Nov-2023]. 2014. URL: https://www.statisticshowto.com/cohens-kappa-statistic/.

[286] Hamed Taherdoost. "Smart Contracts in Blockchain Technology: A Critical Review". In: *Information* 14.2 (2023), p. 117.

[287] Sean Tan et al. "LATTE: Visual construction of smart contracts". In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2020, pp. 2713–2716.

[288] Yong Tang et al. "Blockchain Ethics Research: A Conceptual Model". In: *Proceedings of the 2019 on Computers and People Research Conference*. SIGMIS-CPR '19. Nashville, TN, USA: Association for Computing Machinery, 2019, pp. 43–49. ISBN: 9781450360883. DOI: 10.1145/3322385.3322397. URL: https://doi.org/10.1145/3322385.3322397.

[289] T. Tateishi et al. "Automatic smart contract generation using controlled natural language and template". In: *IBM Journal of Research and Development* 63.2/3 (2019), 6:1–6:12. DOI: 10.1147/JRD.2019.2900643.

[290] Erzhena Tcydenova et al. "Decentralized Access Control for Internet of Things Using Decentralized Identifiers and Multi-signature Smart Contracts". In: *2022 International*

Conference on Platform Technology and Service (PlatCon). 2022, pp. 66–70. DOI: 10.1109/PlatCon55845.2022.9932120.

[291] Vyper Team. *Vyper Documentation*. 2024. URL: https://docs.vyperlang.org/en/stable/.

[292] Miguel A. Teruel and Juan Trujillo. "Easing ĐApp Interaction for Non-Blockchain Users from a Conceptual Modelling Approach". In: *Applied Sciences* 10.12 (2020). ISSN: 2076-3417. DOI: 10.3390/app10124280. URL: https://www.mdpi.com/2076-3417/10/12/4280.

[293] Yu Tong et al. "Smart Contract Generation Assisted by AI-Based Word Segmentation". In: *Applied Sciences* 12.9 (2022), p. 4773.

[294] Lucian A. Trestioreanu et al. "Blockly2Hooks: Smart Contracts for Everyone with the XRP Ledger and Google Blockly". In: *2023 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*. 2023, pp. 145–150. DOI: 10.1109/DAPPS57946.2023.00027.

[295] Wei-Tek Tsai et al. "Invited Paper: Beagle: A New Framework for Smart Contracts Taking Account of Law". In: *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. 2019, pp. 134–13411. DOI: 10.1109/SOSE.2019.00028.

[296] Konstantinos Tsiounis and Kostas Kontogiannis. "Goal and Policy Based Code Generation and Deployment of Smart Contracts". In: *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. 2022, pp. 1227–1230. DOI: 10.1109/SANER53432.2022.00145.

[297] Muhammad Usman et al. "Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method". In: *Information and Software Technology* 85 (2017), pp. 43–59. ISSN: 0950-5849. DOI: https://doi.org/10.1016/

j.infsof.2017.01.006. URL: https://www.sciencedirect.com/science/article/pii/S0950584917300472.

[298] Anna Vacca et al. "A Systematic Literature Review of Blockchain and Smart Contract Development: Techniques, Tools, and Open Challenges". In: *Journal of Systems and Software* 174 (2021), p. 110891. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2020.110891. URL: https://www.sciencedirect.com/science/article/pii/S0164121220302818.

[299] Anna Vacca et al. "An Empirical Investigation on the Trade-off between Smart Contract Readability and Gas Consumption". In: *2022 IEEE/ACM 30th International Conference on Program Comprehension (ICPC)*. 2022, pp. 214–224. DOI: 10.1145/3524610.3529157.

[300] Ángel Jesús Varela-Vaca and Antonia M. Reina Quintero. "Smart Contract Languages: A Multivocal Mapping Study". In: *ACM Comput. Surv.* 54.1 (Jan. 2021). ISSN: 0360-0300. DOI: 10.1145/3423166. URL: https://doi.org/10.1145/3423166.

[301] Radha Madhu Seekar Vedula, Robin Singh Bhadoria, and Manish Dixit. *Integrating blockchain with AI*. English. IGI Global, 2020, pp. 1–25. DOI: 10.4018/978-1-7998-5876-8.ch001. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85129465780&doi=10.4018%2f978-1-7998-5876-8.ch001&partnerID=40&md5=f98d71e03df0a192e42a1a5bd94aab29.

[302] John Venable, Jan Pries-Heje, and Richard Baskerville. "A Comprehensive Framework for Evaluation in Design Science Research". In: *Design Science Research in Information Systems. Advances in Theory and Practice*. Ed. by Ken Peffers, Marcus Rothenberger, and Bill Kuechler. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 423–438. ISBN: 978-3-642-29863-9.

[303] Cristina Viano et al. "Civic Blockchain: Making blockchains accessible for social collaborative economies". In: *Journal of Responsible Technology* 15 (2023), p. 100066.

ISSN: 2666-6596. DOI: https://doi.org/10.1016/j.jrt.2023.100066. URL: https://www.sciencedirect.com/science/article/pii/S2666659623000094.

[304] Giulia Vilone and Luca Longo. "Notions of explainability and evaluation approaches for explainable artificial intelligence". In: *Information Fusion* 76 (2021), pp. 89–106. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2021.05.009. URL: https://www.sciencedirect.com/science/article/pii/S1566253521001093.

[305] Dabao Wang et al. "Penny Wise and Pound Foolish: Quantifying the Risk of Unlimited Approval of ERC20 Tokens on Ethereum". In: *Proceedings of the 25th International Symposium on Research in Attacks, Intrusions and Defenses*. RAID '22. Limassol, Cyprus: Association for Computing Machinery, 2022, pp. 99–114. ISBN: 9781450397049. DOI: 10.1145/3545948.3545963. URL: https://doi.org/10.1145/3545948.3545963.

[306] Shuai Wang et al. "A Novel Blockchain Oracle Implementation Scheme Based on Application Specific Knowledge Engines". In: *2019 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. 2019, pp. 258–262. DOI: 10.1109/SOLI48380.2019.8955107.

[307] Shuai Wang et al. "An Overview of Smart Contract: Architecture, Applications, and Future Trends". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 108–113. DOI: 10.1109/IVS.2018.8500488.

[308] Shuai Wang et al. "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.11 (2019), pp. 2266–2277. DOI: 10.1109/TSMC.2019.2895123.

[309] Shuai Wang et al. "Decentralized autonomous organizations: Concept, model, and applications". In: *IEEE Transactions on Computational Social Systems* 6.5 (2019), pp. 870–878.

[310] Ziyan Wang et al. "An Empirical Study of Solidity Language Features". In: *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. 2021, pp. 698–707. DOI: 10.1109/QRS-C55045.2021.00105.

[311] Paul Ward. "Cognitive Task Analysis". In: Jan. 2014, pp. 143–146.

[312] Tim Weingaertner et al. "Smart Contracts Using Blockly: Representing a Purchase Agreement Using a Graphical Programming Language". In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. 2018, pp. 55–64. DOI: 10.1109/CVCBT.2018.00012.

[313] Christopher D. Wickens. "Multiple Resources and Mental Workload". In: *Human Factors* 50.3 (2008). PMID: 18689052, pp. 449–455. DOI: 10.1518/001872008X288394. eprint: https://doi.org/10.1518/001872008X288394. URL: https://doi.org/10.1518/001872008X288394.

[314] Roel J. Wieringa. "Conceptual Frameworks". In: *Design Science Methodology for Information Systems and Software Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 73–91. ISBN: 978-3-662-43839-8. DOI: 10.1007/978-3-662-43839-8_8. URL: https://doi.org/10.1007/978-3-662-43839-8_8.

[315] Witnet. *Witnet-the decentralized oracle network*. [Accessed 26-May-2023]. 2023. URL: https://witnet.io.

[316] Claes Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. EASE '14. London, England, United Kingdom: Association for Computing Machinery, 2014. ISBN: 9781450324762. DOI: 10.1145/2601248.2601268. URL: https://doi.org/10.1145/2601248.2601268.

[317] Claes Wohlin et al. *Experimentation in software engineering*. Vol. 236. Springer, 2012.

[318] Maximilian Wöhrer and Uwe Zdun. "Domain Specific Language for Smart Contract Development". In: *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2020, pp. 1–9. DOI: 10.1109/ICBC48266.2020.9169399.

[319] Christine T Wolf. "Explainability scenarios: towards scenario-based XAI design". In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 2019, pp. 252–257.

[320] Gavin Wood et al. "Ethereum: A Secure Decentralised Generalised Transaction Ledger". In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[321] Long Xu and Yang Li. "Internet of Things Access Control System Based on Smart Contract". In: *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*. 2021, pp. 659–662. DOI: 10.1109/AIID51893.2021.9456510.

[322] Kailun Yan et al. "Bad Apples: Understanding the Centralized Security Risks in Decentralized Ecosystems". In: *Proceedings of the ACM Web Conference 2023*. WWW '23. , Austin, TX, USA, Association for Computing Machinery, 2023, pp. 2274–2283. ISBN: 9781450394161. DOI: 10.1145/3543507.3583393. URL: https://doi.org/10.1145/3543507.3583393.

[323] Guang Yang et al. "CCGIR: Information retrieval-based code comment generation method for smart contracts". In: *Knowledge-Based Systems* 237 (2022), p. 107858. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2021.107858. URL: https://www.sciencedirect.com/science/article/pii/S0950705121010406.

[324] Lanxin Yang et al. "Quality Assessment in Systematic Literature Reviews: A Software Engineering Perspective". In: *Information and Software Technology* 130 (2021), p. 106397. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2020.106397. URL: https://www.sciencedirect.com/science/article/pii/S0950584920301610.

[325] Lingxue Yang, Hongrun Wang, and Léa A. Deleris. "What Does It Mean to Explain? A User-Centered Study on AI Explainability". In: *Artificial Intelligence in HCI*. Ed. by Helmut Degen and Stavroula Ntoa. Cham: Springer International Publishing, 2021, pp. 107–121. ISBN: 978-3-030-77772-2.

[326] Semi Yulianto et al. "Security Risks and Best Practices for Blockchain and Smart Contracts: A Systematic Literature Review". In: *2023 International Conference on Information Management and Technology (ICIMTech)*. 2023, pp. 1–6. DOI: 10.1109/ICIMTech59029.2023.10278055.

[327] Vlad Zamfir. *Blockchain Governance [Video file]*. [Accessed 15-May-2023]. 2019. URL: https://www.youtube.com/watch?v=PKyk5DnmW50.

[328] Fan Zhang et al. "Town Crier: An Authenticated Data Feed for Smart Contracts". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 270–282. ISBN: 9781450341394. DOI: 10.1145/2976749.2978326. URL: https://doi.org/10.1145/2976749.2978326.

[329] Zhenhua Zhang et al. "CCGRA: Smart Contract Code Comment Generation with Retrieval-enhanced Approach." In: *SEKE*. 2023, pp. 212–217.

[330] Zibo Zhao, Kiyoshi Nakayama, and Ratnesh Sharma. "Decentralized Transactive Energy Auctions with Bandit Learning". In: *2019 IEEE PES Transactive Energy Systems Conference (TESC)*. 2019, pp. 1–5. DOI: 10.1109/TESC.2019.8843371.

[331] Gavin Zheng et al. "Basic Concepts". In: *Ethereum Smart Contract Development in Solidity*. Singapore: Springer Singapore, 2021, pp. 3–15. ISBN: 978-981-15-6218-1. DOI: 10.1007/978-981-15-6218-1_1. URL: https://doi.org/10.1007/978-981-15-6218-1_1.

[332] Zibin Zheng et al. "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends". In: *2017 IEEE International Congress on Big Data (BigData Congress)*. 2017, pp. 557–564. DOI: 10.1109/BigDataCongress.2017.85.

[333] Zibin Zheng et al. "An Overview on Smart Contracts: Challenges, Advances and Platforms". In: *Future Generation Computer Systems* 105 (2020), pp. 475–491. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2019.12.019. URL: https://www.sciencedirect.com/science/article/pii/S0167739X19316280.

[334] Zibin Zheng et al. "Blockchain challenges and opportunities: A survey". In: *International journal of web and grid services* 14.4 (2018), pp. 352–375.

[335] Weiqin Zou et al. "Smart Contract Development: Challenges and Opportunities". In: *IEEE Transactions on Software Engineering* 47.10 (2021), pp. 2084–2106. DOI: 10.1109/TSE.2019.2942301.

Appendix One

Primary Studies Quality Assessment and Summary - Chapter 2

Table A.1 presents the quality assessment results for the selected primary studies. Table A.2 summarizes their concerns and solutions, both were discussed in Chapter 2.

Table A.1: Quality Assessment Results

Study	Rationality			Rigor			Credibility		Total	Study	Rationality			Rigor			Credibility		Total
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8			Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	
[76]	1	1	1	1	1	1	1	0	7	[243]	1	1	1	1	1	1	1	0	7
[157]	1	1	1	1	1	1	1	0	7	[295]	1	1	0.5	1	0	0.5	1	0	5
[130]	1	1	1	1	1	1	1	0	7	[73]	1	1	1	1	1	1	1	0	7
[156]	1	1	1	1	0.5	0	0.5	0	5	[32]	1	1	1	1	1	1	1	0	7
[168]	1	1	1	1	0.5	0.5	1	0.5	6.5	[287]	1	1	1	1	0.5	0.5	0.5	0	5.5
[230]	1	1	1	1	1	1	1	1	8	[159]	1	1	1	1	1	1	1	0	7
[48]	1	0.5	1	1	0.5	0.5	1	0	5.5	[79]	1	1	1	1	1	1	1	0	7
[227]	1	1	1	1	1	1	1	0	7	[121]	1	1	1	1	1	1	1	0	7
[289]	1	1	0.5	1	1	1	1	0	6.5	[152]	1	1	1	1	1	1	1	0	7
[245]	1	1	0.5	1	0.5	0.5	1	0	5.5	[273]	1	1	1	1	1	1	1	0	7
[140]	1	1	1	1	1	1	1	1	8	[200]	1	1	1	1	1	1	1	0	7
[322]	1	1	1	1	1	1	1	0	7	[250]	1	1	1	1	1	1	1	0	7
[214]	1	1	1	1	1	1	1	0	7	[293]	1	1	1	1	1	1	1	0	7
[28]	1	1	1	1	1	1	1	1	8	[230]	1	1	1	1	1	1	1	1	8
[216]	1	1	1	1	0.5	0.5	1	0	6	[12]	1	1	1	1	0.5	1	1	0	6.5
[54]	1	0.5	0.5	1	1	1	1	1	7	[312]	1	1	1	1	0.5	1	1	0	6.5
[323]	1	1	1	1	1	1	1	1	8	[129]	1	1	1	1	1	1	1	1	8
[329]	1	0.5	0.5	1	1	1	1	1	7	[55]	1	1	1	0.5	0.5	0	0.5	0	4.5
[303]	1	1	1	1	1	1	1	0	7	[134]	1	1	1	1	1	1	0.5	0	6.5
[114]	1	1	1	1	1	1	1	1	8	[247]	1	1	1	1	1	1	1	1	8
[318]	1	1	1	1	1	1	1	0	7	[222]	1	1	1	1	1	1	1	0	7
[113]	1	1	1	1	1	1	1	0	7	[23]	1	1	1	1	1	0.5	0.5	0	6
[120]	1	1	1	1	0.5	0.5	1	1	7	[186]	1	1	1	1	1	1	1	1	8
[96]	1	1	1	1	1	1	1	0.5	7.5	[109]	1	1	1	1	0	0	1	0	5
[190]	1	1	1	1	1	1	1	1	8	[60]	1	1	1	1	1	1	1	0	7
[62]	0.5	0.5	1	1	1	1	1	0	6	[5]	1	1	1	1	0.5	0.5	0.5	0	5.5
[108]	1	1	1	1	1	1	1	0	7	[169]	1	1	1	0.5	0.5	0	0.5	0	4.5
[266]	1	1	1	1	0.5	1	1	1	7.5	[111]	1	1	1	1	1	1	1	0	7
[296]	1	1	1	1	0	0	0.5	0	4.5	[153]	1	1	1	1	1	1	1	0	7
[127]	1	1	1	1	1	1	1	1	8	[198]	1	1	1	1	1	1	1	0	7
[128]	1	1	1	1	1	1	1	0.5	7.5										

Table A.2: List of Primary Studies with Synthesis of the Results

No	Study	Concern	Interventions	Stakeholders	Area
1	[120]	Designing and creating SC has a steep learning curve, posing a challenge.	Using Model-Driven Engineering to aid design and development with Domain Specific Language	Developers (IT & non-IT)	Language
2	[266]	Complexity of smart contracts' construction	JSON-based module for contract modeling, and enables automatic code generation with TypeScript.	Developers (IT)	Language
3	[296]	Meeting stakeholders' goals and policies as complexity increases	Automates contract generation using goal models for tasks and policies.	Developers (IT)	Language
4	[130]	Complex coding process leads to unnormalized code, causing development and maintenance issues.	Uses Long Short-Term Memory Recurrent Neural Networks (LSTM-RNN) to generate contract templates and streamline coding	Developers (IT)	Language
5	[48]	Translating business rules into smart contracts is challenging due to frequent reuse across contracts	Domain-specific ontologies and semantic rules.	Developers (IT & non-IT)	Language
6	[55]	Languages should prioritize user needs, detect critical bugs at compile time, and be blockchain-agnostic	New Language "Obsidian" which aligns with these requirements	Developers (IT)	Language
7	[318]	Disconnect between contractual clauses and code hinders understanding and efficiency.	High-level domain-specific language auto-transformable to implementation	Developers (IT & non-IT) & Users	Language

No	Study	Concern	Interventions	Stakeholders	Area
8	[289]	Development is labor-intensive and error-prone, requiring business collaboration.	Human-readable documents using DSL4SC templates and parameter mapping	Developers (IT)	Language
9	[230]	Existing languages require a steep learning curve and lead to bugs.	Empirical evaluation of existing languages for usability and security.	Developers (IT & non-IT) & Users	Language
10	[62]	SC lacks social interaction capabilities and overlook non-digital currency applications.	Fides, a natural language contract creation framework.	Developers (IT & non-IT)	Language
11	[54]	Challenges in writing safe smart contracts.	Compares usability in Obsidian vs. Solidity.	Developers (IT)	Language
12	[245]	Lack of unified standards complicates development and automatic generation	Auto-generate SC using UML and knowledge extraction.	Developers (IT)	Language
13	[312]	Creation is inaccessible to non-technical experts.	Graphical programming language using modularized legal contracts.	Developers (IT & non-IT)	Language
14-15	[128, 127]	Development is challenging due to diverse platforms with unique terminologies and syntax	DSML language for deploying smart contracts on multiple blockchains	Developers (IT)	Language
16	[287]	Solidity requires programming skills, challenging non-programmers.	Visually creates Ethereum SC with a direct-manipulation interface.	Developers (IT & non-IT)	Language

No	Study	Concern	Interventions	Stakeholders	Area
17	[198]	Solidity's complexity makes it hard for average users to design SC.	Visual programming platform for users	Developers (non-IT) & Users	Language
18	[23]	Lack of code expressiveness hinders user understanding	MDA approach to enhance trust and clarity	Developers (IT) & Users	Language/ Usability
19	[243]	Contracts are not human-readable, hard to modify, and hinder collaborative drafting and mutual consent."	Intelligible Description Language Contract (IDLC) enables collaborative drafting like a text editor.	Developers (IT & non-IT) & Users	Language/ Human-Readability
20	[134]	Enabling domain experts with non-IT backgrounds to collaboratively understand, discuss, and specify the contract.	A specification language to facilitate collaborative design	Developers (IT & non-IT) & Users	Language
21	[293]	Semantic consistency and consent among diverse participants.	Automate converting contract clauses into MNL sentences, understandable by both smart contracts and humans.	Developers (IT & non-IT) & Users	Language
22	[299]	Readability vs gas consumption	Empirical study - Trade off analysis of code readability and gas consumption	Developers (IT)	Language
23	[32]	Lower readability makes smart contracts hard to understand and reuse	iSCREAM, a tool to help developers and researchers estimate code readability	Developers (IT)	Language
24	[323]	Limited budgets and experience result in missing or inaccurate code comments	Automatic code comment generation method	Developers (IT)	Language

No	Study	Concern	Interventions	Stakeholders	Area
25	[129]	Comment-code inconsistencies can mislead developers and introduce vulnerabilities	Detecting comment-code inconsistencies	Developers (IT)	Language
26	[329]	Lack of effective comments makes understanding SC challenging.	Uses retrieval knowledge to generate Solidity code comments.	Developers (IT & non-IT) & Users	Language/ Human-Readability
27	[108]	Limited mechanisms exist to make specification and interpretation accessible to a broader audience.	Model supports semi-automated translation of human-readable contracts	Developers (IT)	Language
28	[121]	Declarative vs. imperative languages	Comparison Analysis	Developers (IT)	Language
29	[250]	Existing language lacks clear mapping to natural language, hindering human understanding	Design concepts & domain-specific language (DSL)	Developers (IT)	Language/ Human-Readability
30	[76]	Developing legally-binding DAOs presents a complex challenge	SLCML: Markup language for legally-binding DAOs.	Developers (IT & non-IT)	Language
31	[5]	Trust concerns in development	Requirements for a lawyer-friendly, human and machine-readable contract authoring language	Developers (IT & non-IT)	Language
32	[247]	Knowledge gap between developers and legal experts	Symboleo: Formal specification language for legal contracts.	Developers (IT & non-IT)	Language
33	[273]	Trade-off between expressiveness and safety in language design.	Scilla: Intermediate-level language for safe smart contracts	Developers (IT)	Language

No	Study	Concern	Interventions	Stakeholders	Area
34	[295]	Challenges stem from design approach, not programming languages used.	Beagle framework integrates law into smart contracts	Developers (IT)	Language/ Legality
35	[168]	Developing and verifying legal smart contracts challenges	Unified model for language requirements.	Developers (IT)	Language/ Legality
36	[73]	Conflicts with existing laws, limitations at the individual contract level, and current technical design issues	Human-based survey - Identifying key barriers to adoption	Developers (IT & non-IT) & Users	Legality
37	[200]	Traditional legal tools fail in altering and updating SC	Standards for altering and undoing smart contracts.	Developers (IT)	Legality
38	[216]	Oracle trustworthiness and compliance in legal adjudication	A novel framework for secure and efficient oracle development.	Developers (IT)	Legality
39	[12]	Smart contracts lack clear legal regulation; applying current law is challenging	Clarification of SC in relation to the Civil Code	Developers (IT & non-IT) & Users	Legality
40	[159]	Updating SC and integrating blockchain in legal systems	System architecture with UI, application logic, and blockchain	Developers (IT)	Legality
41	[79]	The relationship between transaction accounting, immutable code trust, and leveraging distributed crowds and databases.	Examines the landscape of blockchain in terms of trust, governance, decentralization	Developers (IT & non-IT) & Users	Ethical and Social
42	[186]	Fairness issues arising from the logical design.	FairCon, a framework for verifying fairness properties	Developers (IT & non-IT) & Users	Ethical & Social

No	Study	Concern	Interventions	Stakeholders	Area
43	[303]	Ethical and design challenges in using blockchain for social impact.	Human-based survey - Evaluating design choices	Developers (IT & non-IT) & Users	Ethical & Social
44	[153]	Challenges users encounter when interacting with blockchain technology.	Human-based survey - Usability testing for dApp Application	Users	Usability/ Cost & Fees
45	[111]	lack of systematic understanding of user-centered cryptocurrency threats	Human-based survey - Insights for understanding user-centered threats	Users	Usability/ Cost & Fees/ Trust
46	[140]	Non-tech-savvy end-users cannot interpret the source code.	SMARTDOC: Automatically generates user notices for smart contract functions	Users	Usability
47	[227]	Users lack essential knowledge to avoid vulnerable and malicious contract code	Tx2TXT: Automatically generates security-centric textual descriptions	Users	Usability
48	[157]	Adversaries exploiting blockchain's pseudo-anonymity threaten accountability and attribution	A novel user-centric visualization framework for transactions	Users	Usability
49	[113]	Challenges faced by early users in the Cryptocurrency	Human-based survey- Identifying challenges for the HCI community.	Users	Usability/ Cost & Fees/ Trust
50	[190]	Accessibility for users with disabilities.	Human-based survey - Analyzing accessibility	Users	Usability
51	[156]	Accessibility and understandability concerns, especially for vulnerable individuals	Visualization platform for creating and verifying contract content.	Users	Usability

No	Study	Concern	Interventions	Stakeholders	Area
52	[214]	Developing public understanding of technologies	Human-based survey - Explaining blockchain from an HCI perspective.	Users	Usability
53	[109]	SCs are complex and not user-friendly for the average user.	SC generator using Ethereum's ERC standards with a configuration wizard	Users	Human-Readability
54	[222]	Complex gas triangle should not be exposed to end-users	Empirical evidence to support the claim.	Users	Cost & Fees
55	[96]	The impact of fee prices on user activities on Ethereum.	Empirical study - Insights and analysis of fees	Users	Cost & Fees
56	[169]	Centralization risk associated with smart contracts.	Insights on the implications of centralization risk	Users	Governance/ Trust
57	[152]	Centralization risk associated with smart contracts.	A library to ensure responsible ownership and management of ERC20 tokens	Users	Governance
58	[322]	Centralization risk associated with smart contracts.	Empirical study - Detecting centralized security risks	Users	Governance
59	[114]	Privileged parties at the application layer.	Ethpector tool for detecting privileged parties in binary smart contract code on Ethereum	Users	Governance/ Trust
60	[60]	Human trust in these systems is an issue	Human-based survey - Understand trust in blockchain systems	Users	Trust
61	[28]	Examines prosumers' concerns about smart contracts	Human-based survey - Social and legal acceptance.	Users	Trust

Appendix Two

Semi-Structure Interview Questions - Chapter 3

Figure B.1 presents the semi-structure interview questions for chapter 3. The responses from the developers' interviews have been uploaded to a public repository for better clarity and accessibility at https://github.com/halghanmi/ExplainableSC/tree/Systemisation-of-Knowledge

General Questions
• Do you consent to participate in this interview and agree to have your responses recorded and used for research purposes?
• How many years have you been working with smart contracts (Development Background)
Transparency
• Q1: How would you define transparency in the context of smart contracts?
• Q2: Is it compulsory for developers to make the source code available?
• Q3: If the code is disclosed, is that sufficient to ensure transparency for the DApp?
• Q4: Are the implementations of smart contracts reflected in high-level considerations such as the front-end? (For example, privileged accounts and risk management functions)
• Q5: How do you connect the implementation of a smart contract with high-level considerations? How do you explain to end-users?
• Q6: From your perspective, what information should be transparent for user understanding and trust?
• Q7: Do you agree with the statement: "Smart contract end-users can understand the functions of the contracts and transactions without any technical experience"?
Accountability
• Q8: How would you define accountability in the context of smart contracts?
• Q9: Who is accountable in smart contract applications? (Assignment of responsibility, making decisions or outcomes produced by the system)
• Q10: What mechanisms do you use to ensure that actions and decisions within a smart contract are accountable?
• Q11: Do you agree with the statement: " Accountability at the high/governance level is often unclear. There is often confusion about who makes decisions within the entities operating DApps."
Understandability
• Q12: From your perspective, are the current forms/presentation of smart contracts and transactions understandable to humans?
• Q13: Is commenting on smart contract code intended to enhance end-user understanding and provide explanations?
• Q14: Do you think users can grasp the whole process of interacting with Dapp or understand the underlying logic of smart contracts?
• Q15: Do you agree with the statement: "The design and presentation of a smart contract's content, data models, processes, authorities, and dependencies are often complex and poorly understood, especially by non-technical or beginner users"?
What low-level implementation can be utilizes for Information provision and explanation?

Figure B.1: Semi-Structure Interview Questions

Appendix Three

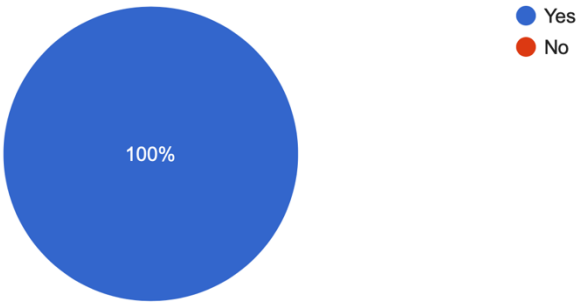
ExplanaSC Evaluation Survey and Results - Chapter 4

This appendix presents the survey questions and experts' responses to evaluate the ExplanaSC framework described in Chapter 4. Detailed information on the framework's iteration processes, evaluation, and the feasibility of implementing the scenarios demonstrated in Chapter 4 have been uploaded to the public repository at https://github.com/halghanmi/ExplainableSC/tree/ExplanaSC-Framework

Evaluating the Effectiveness of a Framework for Smart Contract Explanation Requirements

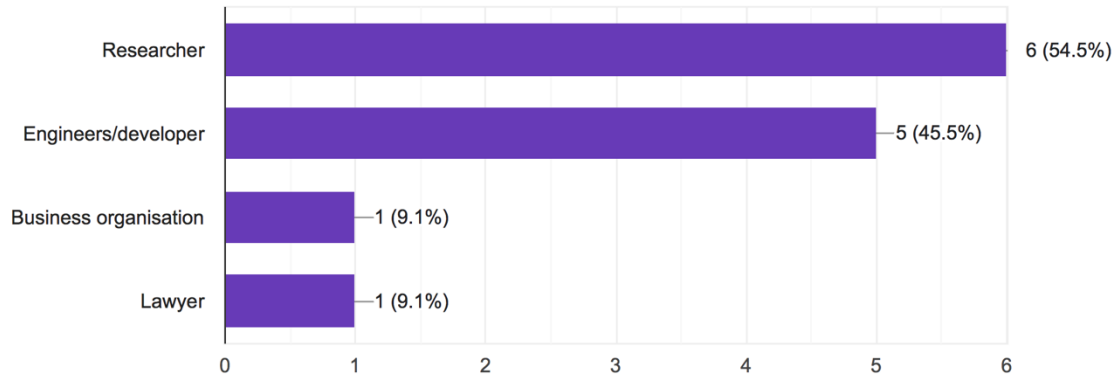
Would you be willing to provide your opinion on our proposed framework?

11 responses



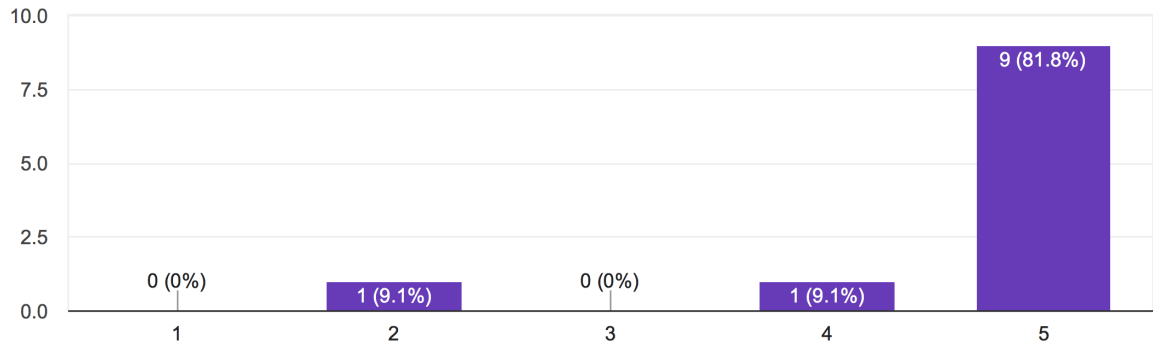
First, how would you describe your main role in the field of smart contracts?

11 responses



How important do you think it is to provide transparent and comprehensible explanations for SC decisions?

11 responses



Are the elements of the proposed framework clear and understandable?

ID	Response	Comments
ID-1	Yes	
ID-2	Yes	
ID-3	No	
ID-4	Yes	
ID-5	Yes	
ID-6	No	I believe an introduction would help in clearing any ambiguity
ID-7	No	
ID-8	Yes	Clear, but needs an example use case for usage illustration.
ID-9	Yes	
ID-10	Yes	
ID-11	Yes	

Is the proposed framework potentially easy to use?

ID	Response	Comments
ID-1	Yes	
ID-2	No	
ID-3	No	
ID-4	Yes	
ID-5	Yes	
ID-6	No	It is premature to decide by just looking to the illustration
ID-7	Yes	
ID-8	Yes	
ID-9	Yes	The proposed framework is potentially easy to use, but it depends on a number of factors, including the complexity of the SC decision ,the domain in which it is used and the level of technical expertise of the users. The framework is structured and provides a clear roadmap for determining the information requirements for SC decisions. This makes it easy to follow and use.
ID-10	Yes	
ID-11	Yes	

Is the proposed framework useful for supporting the design of human-centred SC?

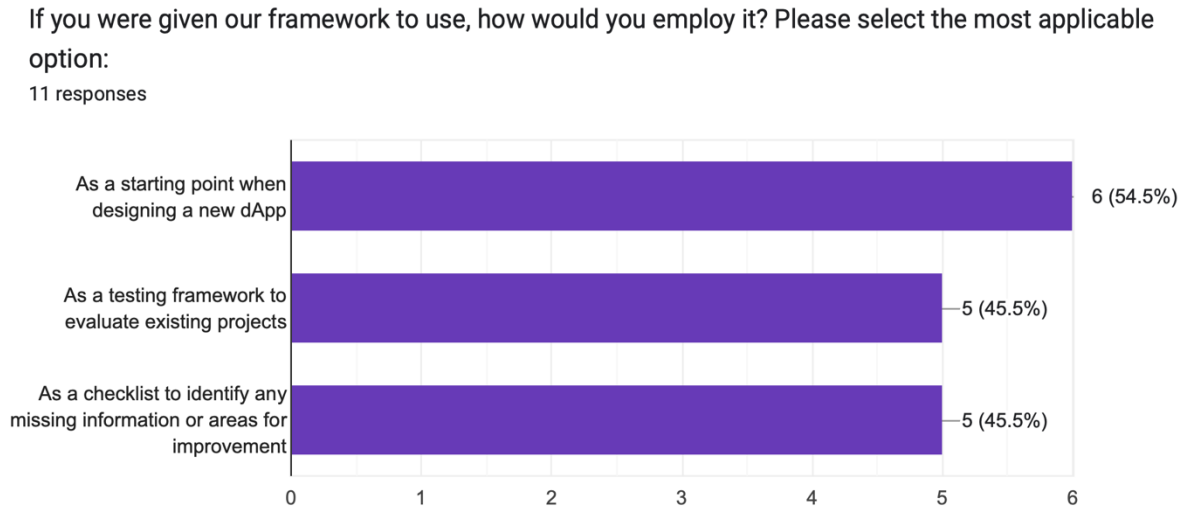
ID	Response	Comments
ID-1	Yes	
ID-2	Yes	
ID-3	Yes	
ID-4	Yes	
ID-5	Yes	
ID-6	No	I am not in favour of human intervention
ID-7	Yes	
ID-8	Yes	
ID-9	Yes	Yes, Once the framework has returned the SC output and the reason why it has returned that output, this information can be used to design SCs that are more understandable and transparent.
ID-10	Yes	
ID-11	Yes	

Are the elements of the proposed framework complete?

ID	Response	Comments
ID-1	Yes	
ID-2	No	
ID-3	Yes	
ID-4	Yes	
ID-5	No	Integrity could be also considered as part of how the decision is made.
ID-6	No	Can’t really judge at this stage
ID-7	Yes	
ID-8	No	It covers various important high-level factors. But it might also needs to consider security aspects, user-experience, nodes behaviour, public or private settings, the impact of the underlying infrastructure on the smart contract outcomes, upgradability, portaility, and so on.
ID-9	Yes	
ID-10	Yes	
ID-11	No	

Is the proposed framework feasible?

ID	Response	Comments
ID-1	Yes	
ID-2	Yes	
ID-3	Yes	
ID-4	Yes	
ID-5	Yes	
ID-6	Yes	
ID-7	Yes	
ID-8	Yes	
ID-9	Yes	
ID-10	Yes	
ID-11	Yes	



Based on your expertise, what specific information do you think is important for users to understand smart contract decisions?

ID-1	Users need to ba aware of the security controls put in place by the smart contract. This involves undrestanding the code for possible vulnerabilities. Users need to recogise the risks involved with using the smart contract by being aware of the security protocols. Users should be aware of any legal requirements, contractual terms, or potential risks associated with using the smart contract.
ID-2	The most importaing thing we need to clarify to users is why we need to use smart contracts it , as well as we need to think of how courts deal with smart contractsI'm
ID-3	components of the smart contract and what the relation with Blockchain

ID-4	It is important for the end user to view how the smart contract work and sometimes they read the code in order to trust the smart contract, it is their money after all. One good example, Uniswap smart contract it has a complex structure and it is quite hard to understand even for developers. Uniswap provide a documentation that explains how the smart contract works in details and they provided some visualizations to help understand it better. https://docs.uniswap.org/concepts/uniswap-protocol https://www.desmos.com/calculator/j8eppi5vvu and more in their docs
ID-5	Probably as part of how the decision is made, it will be great to consider the integrity of the smart contracts, especially if you look at upgradable smart contracts.
ID-6	Roles and responsibilities
ID-7	I wish there was some input on the unexpected event.
ID-8	It might also needs to consider security aspects, user-experience, nodes behaviour, public or private settings, the most importantly; the impact of the underlying infrastructure on the smart contract outcomes, upgradability, portaility, and so on.
ID-9	The input of the smart contract and its purpose or goal are indeed important factors that can help users understand the decisions made by the smart contract. These factors significantly influence the overall processing and functioning of the smart contract
ID-10	Mostrly ownability and permissions on the SC. For example who has permision to burn or mint tokens in a bridge execution.
ID-11	Design of human-centered SC

Please provide any feedback that could help us improve and refine our framework

ID-1	Smart contract users would find it helpful to know the location where the data will be stored and to have a clear understanding of all the legal implications and considerations associated with the smart contract. In genral, the framework is clear and understandable. It may be beneficial to give a catchy name to your framework
ID-2	No comments.
ID-3	Good luck to you
ID-4	for me personally, i could simulate any action with any smart contract using some developer toolings like https://tenderly.co in order to know how the smart contract behave. but i usually start with reading the smart contract code in https://etherscan.io if the code isn't then the framework you provide could work for most people. Great job!
ID-5	No comments.
ID-6	I believe illustrating/adding the point (at what level of the development lifecycle) where SC will take place in the BC application
ID-7	I think the input information is too specific. In my opinion, there is some input on the unexpected event like accident in our society.
ID-8	No comments.

ID-9	I believe that the usage and integration of the framework should be simple, clear, and easy to understand for users of all levels of expertise. This makes the framework more accessible to a wider range of users.
ID-10	Your framework would be applicable for transparency when minting and burning ERC20 tokens. Investidores always fear minting functions due inflationary catastrophe.
ID-11	You are doing good

Appendix Four

Evaluation Matrices & Results - Chapter 5

This appendix includes the generic template developed for evaluating potential surprises in the selected decentralized application. It also presents the metrics used during the evaluation, samples of two reviewers' evaluations and final results. The complete results of each reviewer's evaluation and the implementation of explainability requirements, considering the explanation purposes, have been uploaded to a public repository for better presentation at https://github.com/halghanmi/ExplainableSC/tree/Explainability-Purposes-and-Surprises-Evaluation.

The Generic Template for Evaluation

Scenario	Purpose	Element	Element Weight (W)	Setting Information Assessment	Setting Score (S)	Outcome Information Assessment	Outcome Score (O)	Potential DoS	Aggregate DoS	Normalized DoS	Required Improvement	Note
[Scenario 1]	*[Justify]*	*[Element 1]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*	*[Sum up the individual surprise degrees for all elements]*	*[Aggregate Surprise Degre/Number of Total Elements]*	*[High, Mediem. Low]*	*[Additional Comments]*
		[Element 2]	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
	[Clarify]	*[Element 1]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
		[Element 3]	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
	[Compliance]	*[Element 1]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
		[Element 3]	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
	[Consent]	*[Element 2]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
	[Learning]	*[Element 4]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
[Scenario 2]	*[Justify]*	*[Element 1]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*	*[Sum up the individual surprise degrees for all elements]*	*[Aggregate Surprise Degre/Number of Total Elements/*	*[high, Mediem. Low]*	*[Additional Comments]*
	[Clarify]	*[Element 1]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				
	[Consent]	*[Element 1]*	*[0,1]*	*[Evaluation Matrix]*	*[0,0.5]*	*[Evaluation Matrix]*	*[0,0.5]*	*W *(S+O)*				

Setting & Outcomes Evaluation Matrix

Justification		
Rationale/Justification	**Degree**	**Score**
Minimal or no rationale explanation provided	High	0.5
Some rationale is provided, but not for all aspects.	Medium	0.25
Clear and comprehensive rationale provided for all aspects	Low	0

Clarification		
Clarity of Element Aspects	**Degree**	**Score**
Element aspects are unclear, opaque or not specified.	High	0.5
Some aspects of elements may be unclear	Medium	0.25
Element aspects are clearly defined	Low	0

Consent (Offer and Acceptance)		
Transparency of Intent	**Degree**	**Score**
Intent behind obtaining consent is unclear nor not well-documented.	High	0.5
Some transparency in conveying consent process.	Medium	0.25
Clear & well-documented explanation of intent behind the obtaining consent.	Low	0

Compliance		
Regulatory Considerations	**Degree**	**Score**
Minimal or no mention of regulatory considerations.	High	0.5
Some regulatory considerations may be mentioned, but not comprehensive.	Medium	0.25
All relevant regulatory considerations are clearly mentioned and addressed.	Low	0

Interpretation of Potential Surprises Results

Potential Surprise	Values
Very Low	[0, 0.20]
Low	[0.21, 0.4]
Medium	[0.41, 0.6]
High	[0.61, 0.8]
Very High	[0.81, 1]

Sample of Reviewers Results

Reviewer 1 Results - Roles & Responsibilities Scenario

Project ID	Purpose of Explanation	Element to Evaluate	Element Weight (W)	Setting Information	Setting Score (S)	Outcome Information	Outcome Score (O)	Potential DoS	Aggregate DoS	Normalized DoS	Required Improvement	Note
P1	[Clarify]	List of Roles	0.8	Element aspects are clearly defined	0	Element aspects are unclear, opaque or not specified.	0.5	0.4	4.5	0.56	Medium	I reviewed the white paper and found a list of roles such as market manager, handler manager and operator. However, the code shows more roles such as owner, SI handler and there are two roles one for operator and one for operators.
		Responsibilities & Privileges	0.9	Some aspects of elements may be unclear	0.25	Element aspects are unclear, opaque or not specified.	0.5	0.675				Setting info has part of responsibilities list. Market manager and handler manager have clear responsibilities. Yet, there is a list of government responsibilities that will be implemented later as a decentralized process but it is a general list for all administrators and stakeholders, which has not specified who will be responsible for what. For now, it only mentions that the protocol administrators are responsible for all aspects of the protocol.
		Transfer/add Roles	0.6	Some aspects of elements may be unclear	0.25	Some aspects of elements may be unclear	0.25	0.3				The contract facilitates transferring and adding roles for stakeholders, but lacks guidance on how
		Permission Hierarchy	0.8	Element aspects are unclear, opaque or not specified.	0.5	Some aspects of elements may be unclear	0.25	0.6				There is no information available in setting regarding the accounts of the authorities or the specific privileges granted to different members of the project team
	[Justify]	Decision-Making Rationale	0.8	Minimal or no rationale explanation provided	0.5	Minimal or no rationale explanation provided	0.5	0.8				No justification has been provided for human decision-making
		Permission Levels and Escalation	0.6	Minimal or no rationale explanation provided	0.5	Minimal or no rationale explanation provided	0.5	0.6				No justification or explanation is provided for criteria to change roles. In the outcome, an emitted event shows information about the addresses of the new and old owners only.
	[Compliance]	User Data Regulations	0.9	Some regulatory considerations may be mentioned, but not comprehensive.	0.25	Minimal or no mention of regulatory considerations.	0.5	0.675				There is some mention of following regulatory guidelines, but there is no specific information provided about which regulations are being referred to, or how the project applies these regulations in both its settings and outcomes.
	[Consent]	Consent for Data Collection and Processing	0.9	Clear and well-documented explanation of the intent behind the obtaining consent.	0	Intent behind obtaining consent is unclear nor not well-documented.	0.5	0.45				The project clearly states that by using the website and its services, users are giving their consent for the use of personal data. However, there is no indication of the specific process that the project is following in handling this personal data.
P2	[Clarify]	List of Roles	0.8	Element aspects are unclear, opaque or not specified.	0.5	Some aspects of elements may be unclear	0.25	0.6	5.875	0.73	High	I was unable to find information about the roles in the setting, however, in the code I found roles such as owner, minter and controller.
		Responsibilities & Privileges	0.9	Element aspects are unclear, opaque or not specified.	0.5	Some aspects of elements may be unclear	0.25	0.675				No responsibilities were given. I can read some from the code but I assume it is difficult for regular users to understand.
		Transfer/add Roles	0.6	Element aspects are unclear, opaque or not specified.	0.5	Element aspects are unclear, opaque or not specified.	0.5	0.6				lacks information and guidance
		Permission Hierarchy	0.8	Element aspects are unclear, opaque or not specified.	0.5	Element aspects are unclear, opaque or not specified.	0.5	0.8				There is no information available in setting regarding the accounts of the authorities or the specific privileges granted to different members of the project team
	[Justify]	Decision-Making Rationale	0.8	Minimal or no rationale explanation provided	0.5	Minimal or no rationale explanation provided	0.5	0.8				No justification has been provided for human decision-making
		Permission Levels and Escalation	0.6	Minimal or no rationale explanation provided	0.5	Minimal or no rationale explanation provided	0.5	0.6				No justification or explanation is provided for criteria to change roles.
	[Compliance]	User Data Regulations	0.9	Minimal or no mention of regulatory considerations.	0.5	Minimal or no mention of regulatory considerations.	0.5	0.9				There is no mention of regulatory guidelines, or the pervicsn of terms and conditions.
	[Consent]	Consent for Data Collection and Processing	0.9	Intent behind the offer and acceptance is unclear.	0.5	Intent behind the offer and acceptance is unclear.	0.5	0.9				There is no clear procedure outlined for obtaining consent from users, especially in terms of consent given personal infromation.

Reviewer 2 Results - External Data Scenario

Project ID	Purpose of Explanation	Element to Evaluate	Element Weight (W)	Setting Information	Setting Score (S)	Outcome Information	Outcome Score (O)	Potential DoS	Aggregate DoS	Normalized DoS	Required Improvement	Note
P1	[Clarify]	Identification of External Sources	0.9	Element aspects are unclear, opaque or not specified.	0.5	Element aspects are unclear, opaque or not specified.	0.25	0.675	3.875	0.65	High	There is no confirmation on any of the data sources, the project used an opuge statement "integrate to external price feed oracles, such as Chainlink" so I did not understand if they using only chainlink or there is others.
		Accessibility of Data Sources	0.8	Element aspects are unclear, opaque or not specified.	0.5	Some aspects of elements may be unclear	0.25	0.6				Nothing in setting. As for the outcome, I found the sources in other files associated with the deployment, so it is not recorded in a smart contract or blockchain, i.e. it is not accessible or transparent.
		Pre-Calculation Data/ Aggregation	0.8	Element aspects are unclear, opaque or not specified.	0.25	Element aspects are unclear, opaque or not specified.	0.25	0.4				The setting mentioned using abstract oracles but no further clarification. Non-experts may not understand what abstract oracles mean. They typically refer to the use of decentralized oracles that obtain information from various off-chain or real-world sources.
	[Justify]	External Data Input Justification with Decision	0.9	Minimal or no rationale explanation provided	0.5	Minimal or no rationale explanation provided	0.5	0.9				Nothing in the setting. There has been no use of the value obtained from external sources to justify decisions, i.e., prices have not been utilized in events or recorded for easy access
		Alternative Data Consideration Rationale	0.7	Minimal or no rationale explanation provided	0.5	Minimal or no rationale explanation provided	0.5	0.7				Nothing in the setting. I found a function where the owner can set the token price, this is very concerning as it might be used for manipulation of the price.
	[Compliance]	Compliance with Industry Standards	0.8	Minimal or no mention of regulatory considerations.	0.5	Some regulatory considerations may be mentioned, but not comprehensive.	0.25	0.6				Nothing in the setting about compliance. The outcome required improvement in alternative data compliance and industry standards. It is not clear enough.
P2	[Clarify]	Identification of External Sources	0.9	Element aspects are clearly defined	0.25	Element aspects are clearly defined	0	0.225	2.05	0.34	Low	The project clearly identified external data sources and links. The outcome is also clear and has all the information. However, the project declares the use of their customized oracles and there is no access was provided and there is also a lack of information in the settings regarding the project's customized oracles such as the aggreagation method.
		Accessibility of Data Sources	0.8	Some aspects of elements may be unclear	0.25	Element aspects are clearly defined	0	0.2				
		Pre-Calculation Data/Aggregation	0.8	Some aspects of elements may be unclear	0.25	Element aspects are clearly defined	0	0.2				
	[Justify]	External Data Input Justification with Decision	0.9	Some rationale is provided, but not for all aspects.	0.25	Minimal or no rationale explanation provided	0.5	0.675				The project provides minimal justification in the setting information. However, the decision is not justified with the exact token price used for calculation.
		Alternative Data Consideration Rationale	0.7	Some rationale is provided, but not for all aspects.	0.25	Some rationale is provided, but not for all aspects.	0.25	0.35				The project specifies that if the price is at or below zero, it will invoke its own backup price oracle, presently managed by the project team. In cases where Chainlink does not provide prices, a service maintained by the team will post the price. However, additional explanation is needed regarding the justification for the backup price oracle.
	[Compliance]	Compliance with Industry Standards	0.8	Some regulatory considerations may be mentioned, but not comprehensive.	0.25	Some regulatory considerations may be mentioned, but not comprehensive.	0.25	0.4				The project has demonstrated compliance with data reliability, assurance, and industry standards. The primary area for improvement lies in providing more detailed information about the backup oracle compliance, both in the setting and outcome.

					Reviewer 1				Reviewer 2			
Project	Scenario	Purpose of Explanation	Element to Evaluate	W	S	O	DOS	NDOS	S	O	DOS	NDOS
P1	External Data/Oracles	*Clarify*	O1: Identification of External Sources	0.9	0.25	0.25	0.45	0.68	0.5	0.25	0.675	0.65
			O2: Accessibility of Data Sources	0.8	0.5	0.5	0.8		0.5	0.25	0.6	
			O3: Pre-Calculation Data/ Aggregation	0.8	0.5	0.25	0.6		0.25	0.25	0.4	
		Justify	O4: External Data Input Justification with Decision	0.9	0.5	0.5	0.9		0.5	0.5	0.9	
			O5: Alternative Data Consideration Rationale	0.7	0.5	0.5	0.7		0.5	0.5	0.7	
		Compliance	O6: Compliance with Industry Standards	0.8	0.5	0.25	0.6		0.5	0.25	0.6	
	Roles	*Clarify*	R1:List of Roles	0.8	0	0.5	0.4	0.56	0.25	0.25	0.4	0.59
			R2: Responsibilities & Privileges	0.9	0.25	0.5	0.675		0.25	0.5	0.675	
			R3: Transfer/add Roles	0.6	0.25	0.25	0.3		0.25	0.25	0.3	
			R4: Permission Hierarchy	0.8	0.5	0.25	0.6		0.5	0.5	0.8	
		Justify	R5: Decision-Making Rationale	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
			R6: Permission Levels and Escalation	0.6	0.5	0.5	0.6		0.5	0.5	0.6	
		Compliance	R7: User Data Regulations	0.9	0.25	0.5	0.675		0.25	0.25	0.45	
		Consent	R8: Consent for Data Collection and Processing	0.9	0	0.5	0.45		0.25	0.5	0.675	
	Decision	*Clarify*	D1: Interest Rate Determinants	0.8	0	0.5	0.4	0.41	0	0.25	0.2	0.46
			D2: Loan Approval Criteria	0.9	0	0.25	0.225		0	0.25	0.225	
			D3: Risk Assessment and Mitigation	0.8	0	0.25	0.2		0.25	0.25	0.4	
		Justify	D4: Loan Approval or Denial	0.9	0	0	0		0	0	0	
			D5: Interest Rate Assignment	0.9	0.25	0.25	0.45		0.25	0.5	0.675	
			D6: Loan Amount	0.8	0	0	0		0	0	0	
		Compliance	D7: Regulatory Adherence	0.9	0.25	0.25	0.45		0.25	0.5	0.675	
		Consent	D8: Consent for Terms and Conditions	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
			D9: Consent for Risk Disclosure	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
			D10: Withdrawal of Consent Process	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
P2	External Data/Oracles	*Clarify*	O1: Identification of External Sources	0.9	0	0	0	0.37	0.25	0	0.225	0.34
			O2: Accessibility of Data Sources	0.8	0	0	0		0.25	0	0.2	
			O3: Pre-Calculation Data/ Aggregation	0.8	0.25	0	0.2		0.25	0	0.2	
		Justify	O4: External Data Input Justification with Decision	0.9	0.5	0.5	0.9		0.25	0.5	0.675	
			O5: Alternative Data Consideration Rationale	0.7	0.5	0.5	0.7		0.25	0.25	0.35	
		Compliance	O6: Compliance with Industry Standards	0.8	0.5	0	0.4		0.25	0.25	0.4	
	Roles	*Clarify*	R1:List of Roles	0.8	0.5	0.25	0.6	0.73	0.5	0.25	0.6	0.72
			R2: Responsibilities & Privileges	0.9	0.5	0.25	0.675		0.25	0.5	0.675	
			R3: Transfer/add Roles	0.6	0.5	0.5	0.6		0.5	0.25	0.45	
			R4: Permission Hierarchy	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
		Justify	R5: Decision-Making Rationale	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
			R6: Permission Levels and Escalation	0.6	0.5	0.5	0.6		0.5	0.5	0.6	
		Compliance	R7: User Data Regulations	0.9	0.5	0.5	0.9		0.5	0.5	0.9	
		Consent	R8: Consent for Data Collection and Processing	0.9	0.5	0.5	0.9		0.5	0.5	0.9	
	Decision	*Clarify*	D1: Interest Rate Determinants	0.8	0	0	0	0.33	0	0	0	0.37
			D2: Loan Approval Criteria	0.9	0	0	0		0	0.25	0.225	
			D3: Risk Assessment and Mitigation	0.8	0	0	0		0	0.25	0.2	
		Justify	D4: Loan Approval or Denial	0.9	0	0	0		0	0	0	
			D5: Interest Rate Assignment	0.9	0	0	0		0	0	0	
			D6: Loan Amount	0.8	0	0	0		0	0	0	
		Compliance	D7: Regulatory Adherence	0.9	0.5	0.5	0.9		0.5	0.5	0.9	
		Consent	D8: Consent for Terms and Conditions	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
			D9: Consent for Risk Disclosure	0.8	0.5	0.5	0.8		0.5	0.5	0.8	
			D10: Withdrawal of Consent Process	0.8	0.5	0.5	0.8		0.5	0.5	0.8	

Appendix Five

Ethical Approval

We include the confirmation of ethical approval for conducting surveys.

UNIVERSITY OF BIRMINGHAM

Dear Rami Bahsoon and Hanouf al Ghanmi,

RE: A Survey for Evaluating a Framework for Explainable Smart Contract

Application for Ethical Review: ERN_1234-Jun2023

You project has been considered in line with the University of Birmingham's research ethics processes and on the basis of the information you have provided, it is understood that while your project does involve human participants, the project raises no substantial research ethics issues and therefore no further ethics review is required

Any adverse events occurring during the study should be promptly brought to the Committee's attention by the Principal Investigator and may necessitate further ethical review.

Please ensure that the relevant requirements within the University's Code of Practice for Research and the information and guidance provided on the University's ethics webpages (available at https://intranet.birmingham.ac.uk/finance/accounting/Research-Support-Group/Research-Ethics/Links-and-Resources.aspx) are adhered to.

Please be aware that whilst Health and Safety (H&S) issues may be considered during the ethical review process, you are still required to follow the University's guidance on H&S and to ensure that H&S risk assessments have been carried out as appropriate. For further information about this, please contact your School H&S representative or the University's H&S Unit at healthandsafety@contacts.bham.ac.uk.

Kind regards,

The Co-Chairs of the Science, Technology, Engineering and Mathematics Committee

E-mail: ethics-queries@contacts.bham.ac.uk