A Requirements Analysis for a Decentralized Mathematics Prediction Market

Quentin Botha*, Laurent Bindschaedler†, Christoph Siebenbrunner*
*Research Institute for Cryptoeconomics, Vienna University of Economics and Business
{quentin.botha, christoph.siebenbrunner}@wu.ac.at
†Max Planck Institute for Software Systems
bindsch@mpi-sws.org

***Abstract*—Decentralized mathematics prediction markets promise new forms of collaboration and incentive alignment, but traditional requirements engineering methods fail to address the unique governance, incentive, and security challenges of such Web3 systems. This paper demonstrates how they can be addressed through a requirements-driven design of a decentralized prediction market for mathematical conjectures, and proposes concrete enhancements to existing frameworks. Our work delivers actionable guidelines for engineering secure, incentive-aligned decentralized platforms, and sets a new standard for early-stage RE in the Web3 era.**

***Index Terms*—Decentralized applications, Web3, requirements engineering, prediction markets, governance, incentive mechanisms, security.**

I. Introduction

The emergence of decentralized applications (DApps) powered by blockchain-based smart contracts has enabled new forms of collaboration, governance, and incentive alignment in open, trustless environments. A promising new direction for such 'Web3' applications is Decentralized Science (DeSci), where DApps are used to create incentive-aligned mechanisms for supporting scientific endeavors. An exciting domain for applying DApps in support of science is mathematics. That is because, in mathematics, the 'Oracle problem' of assessing the validity of a submitted contribution can be fully automated thanks to the availability of automated proof checkers.

We hence envision a DeSci-oriented DApp that incentivizes mathematical research by combining a proof checker with a prediction market, allowing other participants to bet on mathematical conjectures, thereby generating sustainable revenue streams that can be used to reward valid contributions. Designing such systems is challenging as it requires careful consideration of governance, incentives, and security in a distributed adversarial setting.

In traditional organizations, software development is often guided top-down, with structured teams and requirements engineering (RE) processes that translate organizational strategies into concrete actions. In contrast, DApps operate in a trustless, adversarial environment where governance is collective, incentives are encoded, and stakeholders are pseudonymous and distributed. These characteristics challenge many assumptions underlying traditional and well-defined RE processes for (open-source) software development.

Requirements-based design is critical in the Web3 context, given the immutability of deployed smart contracts and the adversarial nature of the environment in which they are situated. Missing or misunderstood requirements can lead to irreversible outcomes and systems that fail to serve their intended purpose. Therefore, we have recently seen a surge in research that re-examines RE frameworks, adapting them to better accommodate the unique properties of Web3 systems.

In this paper, we present a *requirements-driven approach* to designing a decentralized mathematics prediction market. Our work is novel in its explicit focus on the RE process for Web3 systems, with particular attention given to governance and incentive structures. We demonstrate that existing frameworks are too implementation-focused to address the early-phase security needs of such systems. To support our claims, we analyze leading RE frameworks in the context of decentralized application development. Overall, this paper makes the following key contributions:

- A systematic comparison of leading requirements engineering frameworks for Web3 systems, highlighting their strengths and limitations.
- A requirements-driven analysis and design of a concrete application inspired by a novel real-world use-case, a decentralized mathematics prediction market.
- An assessment of leading RE4Web3 frameworks based on this case study, which identifies critical gaps in current approaches, and a proposal for an augmented RE4Web3 framework to address these shortcomings.

The remainder of this paper is organized as follows: Section II presents relevant background on DApps and Web3 systems. Section III reviews the current literature on RE4Web3. Section IV presents the requirements analysis for our envisioned application based on a leading RE4Web3 framework as identified by our earlier literature review, as well as an analysis of critical elements these frameworks would have missed. In Section V, we propose a new RE4Web3 framework based on earlier work and extended by the gaps identified in our analysis. In Section VI, we discuss limitations of the present paper. Finally, Section VII summarizes our findings and presents some directions for future work.

II. Background on DApps and Web3 systems

A. Decentralized Applications

Decentralized applications (DApps) are software systems built on distributed ledger technology, primarily blockchains. They are characterized by their open-source nature, decentralized control, and trustless execution. Unlike traditional applications, DApps do not rely on centralized authorities. These applications are a key component of the emerging Web3 ecosystem, often governed by immutable and self-executing blockchain code, usually called smart contracts. A typical DApp architecture consists of a blockchain backend, which handles the application's logic and state transitions, and a user interface accessed via web or mobile clients. Well-known DApps include Uniswap (a decentralized exchange) [1], Compound (a lending platform) [2], and Augur (a decentralized prediction market) [3].

B. Prediction Markets

Prediction markets are platforms that forecast outcomes by allowing participants to place bets on the likelihood of future events. These markets can leverage blockchain technology to create open, censorship-resistant environments where incentivized forecasting and collaborative problem-solving can take place. Popular platforms such as Augur [3] and Polymarket [4] enable users to create markets, trade cryptographic tokens representing outcomes, and receive payouts based on event resolutions.

The process typically involves four stages:

1) **Market Creation**: Defining the event and possible outcomes.
2) **Trading**: Participants buying and selling shares.
3) **Resolution**: An oracle or consensus mechanism determines the outcome.
4) **Payout**: Distributing winnings to holders of correct outcome tokens.

Despite their attractive promise, decentralized prediction markets face several challenges, including ensuring sufficient liquidity for active trading, mitigating the risk of market manipulation, and maintaining reliable and tamper-resistant oracles for outcome resolution. These challenges are amplified in open systems where adversarial behavior and economic incentives can interact in complex ways.

Prediction markets have also been proposed as mechanisms for incentivizing the solution of complex problems by aligning financial rewards with their successful resolution [5]. In such settings, the market not only serves as a forecasting tool but also as a platform for crowd-sourcing intellectual effort, with participants betting on the difficulty of resolving specific problems. This approach introduces additional design considerations and the careful alignment of incentives to ensure honest participation and the integrity of market outcomes.

C. Governance in Web3 Ecosystems

Governance in Web3 ecosystems refers to the processes by which decentralized protocols and applications make collective decisions, evolve rules, and manage upgrades or disputes. Governance typically uses smart contracts and token-based voting to automate proposal processes. Tokens often play a central role in governance, serving as voting rights, proposal submission collateral, or even as a means of funding protocol development. However, token-based governance introduces issues, such as the risk of plutocracy (where influence is concentrated among large holders), voter apathy, and the challenge of maintaining credible neutrality in protocol upgrades. Decentralized autonomous organizations (DAOs) exemplify the ambition of Web3 governance, aiming to encode organizational rules and decision-making processes directly into smart contracts, but must also deal with issues related to incentives, participation, and resilience against adversarial actions.

D. Security Issues in DApps

Security is a paramount concern for DApps, given the high-value assets and adversarial environments in which they operate. Smart contracts are a frequent target for exploitation due to their often immutable and transparent nature. Common vulnerabilities include reentrancy attacks [6]–[8], where malicious contracts repeatedly call into a vulnerable contract to drain funds; integer overflows and underflows [7], [8], which can lead to unintended state changes; and front-running, where attackers exploit the public mempool of transactions that users have submitted but not yet committed to the blockchain to preempt or manipulate transactions for profit [9]. The ERC20 token standard, widely used for fungible tokens, introduces its own set of permission issues, such as approval exploits and the risks associated with unlimited allowances or improper use of the `transferFrom` function [10]. These vulnerabilities have led to numerous high-profile exploits [7], [8] and underscore the need for rigorous security practices throughout the DApp development lifecycle.

Beyond technical vulnerabilities in smart contracts, DApps are also susceptible to economic attacks that exploit their incentive structures. Sybil attacks, where adversaries create multiple identities to gain disproportionate influence, can undermine governance or market integrity [11]. Market manipulation, including wash trading or coordinated attacks on oracles, can distort prices and outcomes, eroding trust in the platform. Rug pulls are episodes where users suddenly remove a large portion of funds, often leaving other users with trapped liquidity. A particularly salient class of economic attacks in blockchain systems is Maximal Extractable Value (MEV), where validators or miners reorder, insert, or censor transactions within a block to extract additional profits [12]. MEV can have profound implications for DApp security, especially in applications where transaction ordering determines the allocation of rewards. The exact nature of MEV usually depends on characteristics of the blockchain, such as the existence of a public mempool, as mentioned above.

Addressing security requires a comprehensive approach that includes protocol design, incentive engineering and a first assessment of security considerations. As DApps mature, an adaptive, multi-layered approach to security becomes essential to counter evolving Web3 threats.

III. Requirements Engineering for Web3 Systems

A. *Review of current RE4Web3 Frameworks*

The proliferation of DApps as interdependent infrastructure has introduced new challenges to the software development process. Namely, managing governance, ensuring security in an adversarial environment, and the absence of centralized control [13], [14]. These considerations invalidate many of the assumptions that underlie traditional requirements engineering, particularly with regard to stakeholders, trusted third parties, and Web3-specific concepts [15]. In response, recent efforts have sought to adapt existing, or proffer new, frameworks better suited to the development of DApps. We have conducted an extensive review of recent RE4Web3 frameworks, summarized in Table I.

Famideh et al. [15] conducted a systematic review of the literature on engineering blockchain-based software systems. They highlight the lack of standardized RE frameworks capable of addressing the unique characteristics of DApps. The authors point to the predominance of ad hoc adaptations and motivate the need for tailored RE frameworks to resolve the ”dire” problem of requirements analysis for Web3 systems.

Much research has explored the extension of traditional SE techniques, such as goal modeling, domain-specific modeling, and model-driven architectures, to the Web3 context. Chawla et al. [16] propose User Requirements Notation, a combination of Use Case Maps and Goal Requirements Language, to better visualize scenarios and draw explicit connections to associated requirements of the system. Porru et al. [13] similarly acknowledge the need to adapt modeling languages to the Web3 context.

Among model-driven approaches, [17] propose eAOM as a modeling language for blockchain applications. eAOM can help capture complex agent interactions but remains oriented toward high-level conceptualization and translation of requirements to working code. Similarly, MDAsmartCD [18] improves model-driven development frameworks for smart contracts, focusing on model-to-code consistency. However, the application of MDAsmartCD presumes formalized requirements, limiting its use for early-stage RE. Building upon this, MDAPW3 [19] addresses the broader DApp lifecycle, leveraging semi-automated code generation during development. However, MDAPW3 emphasizes the design and implementation phases over early-stage RE.

Traditional SE practices have historically been centered on code-centric and document-driven frameworks [20]. Marchesi et al. [21] introduce ABCDE, an agile framework promoting early goal elicitation and iterative development of Ethereum-based DApps. Although ABCDE advocates responsiveness and adaptability during development, it provides limited support to capture decentralized governance structures, incentives, and trust assumptions. Consequently, ABCDE only partially addresses early-stage RE. Applied together with the BOSE framework in [22], a clear and structured identification of actors and smart contract interactions was shown to improve system clarity.

Among the reviewed frameworks, CASCADE [23] is distinct for its explicit targeting of early-phase DApp development. CASCADE integrates domain analysis, stakeholder modeling, and conceptual design into a structured RE framework. Furthermore, it places a strong emphasis on both technical and non-technical stakeholder engagement. The authors suggest a phased adoption of CASCADE for RE and ABCDE for subsequent development and management. This demonstrates the complementarity of conceptual and agile frameworks for DApp development. From the perspective of our intended case study, CASCADE is the best choice as a RE4Web3 framework due to its broad scope and explicit focus on the early-stage RE process.

TABLE I
Summary of Related Work

Article	**Methodology**	**Approach**	**RE Stage**
Famideh et al. (2022) [15]	Systematic Literature Review	Literature synthesis	N/A
Chawla et al. (2020) [16]	Goal-Oriented RE	User Requirements Notation for reasoning about business goals	Analysis
Porru et al. (2017) [13]	Exploratory study on 1184 GitHub repositories	Proposes new research directions	N/A
Waishiang et al. (2024) [17]	eAOM Framework	Extended agent-oriented modeling	Elicitation, Analysis, Development, Validation
Jurgelaitis et al. (2023) [18]	MDAsmartCD	UML-based modeling, automated model transformations	Elicitation, Specification, Validation, and Management
Samanipour et al. (2025) [19]	MDAPW3	BPMN requirements modeling, automated model transformations	Elicitation, Specification, Validation, and Management
Marchesi et al. (2020) [21]	ABCDE	UML-based modeling of contracts and interactions	Partial Elicitation and Analysis, Development, and Validation
Lallai et al. (2020) [22]	BOSE & ABCDE	DApp real-world case study	N/A
Bouraga (2025) [23]	CASCADE	Structured RE for DApps	Elicitation and Analysis

B. *Gaps in Current RE4Web3 Frameworks*

Although several frameworks claim to address early-stage RE4Web3, they do so from limited and often isolated perspectives. CASCADE contributes a structured approach to conceptual modeling, but its emphasis on domain and stakeholder modeling assumes a level of clarity often absent in nascent

DApp projects. More broadly, the reviewed frameworks either adopt top-down abstractions that bypass early socio-technical ambiguities or embed premature assumptions about stakeholder roles, trust, and incentives. By socio-technical ambiguities, we refer to the uncertainties that arise at the intersection of social and technical dimensions in DApp development—such as unclear stakeholder identities, undefined governance structures, and competing incentives—which traditional RE approaches are ill-equipped to manage. This leaves a methodological gap where clarity is needed: the collaborative formation of requirements in conditions of uncertainty, decentralization, and minimal a priori structure.

Through applying CASCADE to an envisioned DApp, we identify concrete limitations and propose extensions that better accommodate the distributed and strategic nature of Web3. In doing so, our contribution is twofold: first, we evaluate CASCADE in practice, and second, we advance contemporary methodological approaches to early-stage RE4Web3.

The most important finding from our analysis is that current RE4Web3 frameworks do not adequately consider security. In CASCADE, security considerations are only briefly mentioned as optional elements of the non-functional requirements and project analysis. ABCDE explicitly includes a security assessment phase and further refers to a separate paper by the same authors [24] with more extensive security checklists aimed specifically at Ethereum smart contract development. More recent contributions that focus on security in the design stage of Web3 development, without being full RE frameworks, include SecBPMN2BC [25] for designing secure business processes for blockchains; [26] and [27] suggest best practices for designing secure and efficient smart contracts; [28] propose best practices for smart contract testing, which can help with improving security. The problem we find that those contributions have in common with the approach suggested by ABCDE and its security extension paper [24] is that they focus on smart contract development. While secure smart contracts are certainly a crucial and necessary aspect of Web3 security, with smart contract vulnerabilities often being at the heart of attacks such as the DAO hack, we find that security for Web3 systems requires a more holistic perspective. Specifically, our analysis below shows that properties of the blockchain system as a whole, such as the possibility of MEV, can allow for attacks that can lead to losses for honest users of a DApp. The possibility of attacks can undermine the economic rationale of participating in the DApp and its ecosystem.

IV. Requirements analysis

This section systematically analyzes the requirements for the decentralized mathematics prediction market platform. We generally follow the CASCADE framework, which we identified as the most suitable RE4Web3 framework for our analysis, extended by a section to cover security considerations. The CASCADE framework consists of four main components: project, stakeholders, requirements, and governance, which we discuss below.

We focus on a minimal viable product (MVP) design to ground the discussion. Our MVP proposal excludes a native protocol token and simplifies the market structure to a single market per conjecture. Specifically, we only consider markets that posit a conjecture under a given set of axioms and that resolve whenever a proof in any direction (positive, negative, or undecidable under the set of axioms) is provided. This approach allows us to examine core requirements while minimizing complexity. Potential extensions going beyond the MVP design are discussed in the concluding section.

A. *Project*

The CASCADE framework identifies multiple components within the Project category, such as the project team, roadmap, strategy, funding, and user guides. The most important aspect of our discussion is the Technology component. A decentralized mathematics prediction market requires more than a standard set of smart contracts typical for most Web3 projects. The most crucial technological component is an automated theorem verifier that can autonomously check proofs submitted by users. Many systems for automated theorem verification exist, such as Rocq [29], Isabelle/HOL [30], E [31], F* [32]. From our perspective, the most promising candidate is Lean4 [33], due to its popularity and ongoing projects to include large parts of known mathematics in its mathlib library, a prerequisite for proving theorems about more advanced areas of mathematics.

B. *Stakeholders*

Identifying system stakeholders is an essential first step to formalizing system requirements. Stakeholder identification must also account for both technical and social actors. In the language of the CASCADE framework, stakeholder modeling involves five categories: Users, Nodes, Blockchain, Oracle, and Devices. Our analysis identifies two principal stakeholder groups—Solvers and Traders—which align most directly with the Users category.

Solvers are expected to submit formal proofs for open conjectures. They are central to the system and provide the basis for market resolution upon submission of valid proof. Their incentives for doing so are financial. Their rewards are paid for by the revenue stream generated by the prediction market. Due to the potential for MEV, Solvers are therefore affected by security considerations. The role of Traders is to express beliefs about the potential resolution of conjectures by trading on the system. They serve a forecasting role akin to participants in classic prediction markets. Again, the incentive is financial. Traders are also sensitive to trust in the Automated Theorem Verifier—which serves as the system's Oracle—and the credible neutrality of the system.

The proposed system is Blockchain-agnostic in principle. As a reference, we consider deployment on Ethereum blockchain. Ethereum is the first smart-contract-enabled blockchain and currently the largest in terms of both market capitalization of its native token and stablecoin supply[1]. Within the CASCADE framework, this design decision concretely maps the Nodes stakeholder role to Ethereum Validators, who are deterministically granted monopoly power over transaction ordering under the Proof-of-Stake consensus protocol. This introduces new security and incentive considerations, particularly in relation to MEV and censorship.

The Devices stakeholder defined in CASCADE is not applicable in the context of the envisioned system, as there is no reliance on hardware sensors or devices external to the blockchain environment.

A contributing role is that of Market Creators, which may be filled by Solvers, Traders, or external stakeholders. In the MVP under consideration, this role is not financially incentivized, in order to avoid added complexity and potential unintended interactions in the incentive design. Further discussion is provided in Section IV-F.

C. Functional Requirements

This section specifies the core features and behaviors that the decentralized math prediction market must perform to enable market creation, proof submission, betting, and reward distribution.

- **FR-01** **Market Creation**: The system shall allow any user to create a new prediction market by submitting a conjecture and its corresponding axiom set.
- **FR-02** **Proof Submission**: The system shall allow users to submit formal proofs for conjectures in active markets.
- **FR-03** **Betting Mechanism**: The system shall allow users to place financial bets on the eventual resolution of a market, corresponding to the truth status of the conjecture.
- **FR-04** **Automated Reward Distribution**: The system shall automatically distribute the market's reward pool to the first user whose submitted proof is verified as valid under the declared axiom set and proof system.
- **FR-05** **Market Resolution**: The system shall automatically resolve a market when a valid proof is verified or when the market reaches its predefined expiration date.
- **FR-06** **Market Status Visibility**: The system shall provide real-time visibility into the status of all markets, including pending and resolved states.
- **FR-07** **Market Discovery**: The system shall provide a searchable and filterable interface that allows users to browse available markets.
- **FR-08** **Proof Validation**: The system shall reject proofs that fail verification under the specified formal system and shall prevent market settlement in such cases.
- **FR-09** **Market Settlement**: The system shall initiate settlement by distributing payouts to traders based on their positions, following successful market resolution.

D. Non-Functional Requirements

This section defines the qualitative characteristics and constraints that the platform should satisfy to ensure its sustainability, fairness, and effective operation.

- **NFR-01** **Economic Sustainability**: The protocol should not require external capital inflows from parties who do not expect to profit within the system. All reward pools should be sustainably funded through user-generated market activity.
- **NFR-02** **Permissionless Access**: The protocol should support permissionless participation in all major interactions, including market creation, proof submission, and trading, without requiring authentication or centralized approval.
- **NFR-03** **Efficiency in Reward Allocation**: The system should minimize fragmentation of solver effort and reward pools by discouraging redundant markets for logically similar conjectures, and should support mechanisms for discovering unresolved, high-value markets.
- **NFR-04** **Scalability**: The platform should support a growing number of users and markets without significant degradation in performance for core functionalities, including market visibility and discovery.
- **NFR-05** **Transparency**: All protocol operations, including market creation, resolution, and settlement, should be transparent and auditable by any participant through on-chain or verifiable off-chain records.
- **NFR-06** **Credible Neutrality**: The protocol should not favor any participant or group. In particular, ordering mechanisms for solver submissions should be credibly neutral to mitigate MEV-related manipulation of the "first valid proof" criterion.
- **NFR-07** **Immutability of Market Rules**: Rules governing active markets should be immutable, and no changes should be introduced that retroactively affect existing conjecture challenges or market participants.
- **NFR-08** **Verifier Auditability**: The automated theorem verifier should operate in a deterministic, reproducible, and publicly auditable manner to ensure trust in resolution outcomes.
- **NFR-09** **Timeliness of Resolution**: The protocol should guarantee that market resolution and settlement occur within a bounded time window following either the submission of a valid proof or market expiration.
- **NFR-10** **Data Availability**: All submitted conjectures, axioms, and proofs should be persistently accessible and verifiable via on-chain storage or a trusted, cryptographically anchored off-chain storage layer.

E. Governance

CASCADE considers two components of governance, Tokenomics and Consensus. In line with our MVP approach, we consider a design without a native token, so the main question regarding Tokenomics is what token to use as the numeraire

[1] Ethereum accounts for 50% of overall stablecoin supply according to https://defillama.com/stablecoins/ and Ether has the second-highest market capitalization according to https://coinmarketcap.com/ (both accessed May 27, 2025).

token for making bets and paying out rewards. A stablecoin token with a well-audited reserve seems to be the most suitable choice here. One important implication of the MVP design is that, in the absence of a protocol token, traditional governance through token-based consensus/voting protocols is not an option, at least with a decentralized design. This is reflected by the immutability requirement above. Adding a protocol token to enable governance and consensus-based evolution of the protocol is an important point for potentially expanding the protocol, but requires careful consideration of the interactions between token-based rewards and potential security concerns, as discussed below.

F. Security considerations

As mentioned before, CASCADE provides no guidance on how to include security considerations. ABCDE includes security considerations in three phases, design, coding, and testing, and proposes multiple proactive controls, the most important of which - according to the paper and for our analysis here - is to specify security requirements.

The most important security requirement for our proposed system is that it should be resilient to attempts to exploit its mechanics for gains. One facet of this resilience is already captured by **FR-04 Automated Reward Distribution**: the system should reward the first user to submit a correct proof for a conjecture. The notion of *first* can create vulnerabilities in the context of blockchains like Ethereum, whose design includes a mempool of submitted transactions that have not yet been added to the blockchain. Mempool transactions should be considered public knowledge, and they can be copied and added to the mempool by anyone. This creates a possible attack vector by inserting a copy of a valid first proof with a different recipient address than the rightful prover. Such an attack would constitute an instance of MEV, as discussed in Section II. The exact nature of how MEV can or cannot materialize depends on the blockchain design, and many blockchain protocols are actively working on eliminating this attack vector. But in the context of our proposed system, such MEV attacks could critically undermine the entire *raison d'être* of the system - if users cannot count on receiving a reward that should be rightfully theirs under the rules of the system protocol, they no longer have an incentive to participate in the ecosystem at all. This shows that MEV needs to be considered in the system design. A blockchain that allows MEV cannot be a sustainable home for the proposed system. Not all blockchains have MEV, but some important ones like Ethereum do. So, if Ethereum in its current form were chosen as the underlying blockchain, a different solution would have to be found. Such a solution could consist of using a different system, such as a Layer-2 blockchain or a centralized subsystem, commit-reveal schemes [34], private mempools [35], or potentially other solutions.

Another facet of resilience is that users should not be able to earn rewards by spuriously creating conjectures for which they already know a proof. Such behavior could amount to a denial-of-service attack, and would degrade user experience for other users in any case. Users who create spurious theorems should at least face the risk of some losses so that at least the expected value of such gains could be negative. In the MVP design, this should be straightforward to achieve by having some cost associated with the creation of conjectures, which could also just be in the form of gas fees that would have to be paid anyway on most blockchains. In designs that feature inflationary rewards, as would be typical for many protocol tokens, this requirement might require more careful calibration of payout functions to ensure exploit resilience.

Our analysis has identified two facets of security vulnerabilities that could lead to denial-of-service attacks or undermine the economic rationale for participating in the ecosystem. These vulnerabilities exist independently of the actual implementation of the protocol in smart contracts, which is the focus of existing RE4Web3 frameworks. One of these vulnerabilities is also specific to the Web3 context, and more specifically to the exact blockchain being targeted for deployment of the smart contracts. Together, these findings suggest that security should be considered early in the RE process, and that they should encompass a wider view than focusing on smart contracts, taking into account as well features of the blockchain such as MEV and how it may interact with the incentive mechanisms or other higher-level design aspects of a Web3 project.

V. Suggestion for an amended RE framework

Our analysis in the previous section showed that CASCADE is generally a suitable framework for an early-stage RE4Web3 framework, but it lacks security considerations. We have found that considering security at a later stage and focusing on smart contracts, as exemplified e.g. in the ABCDE framework, is not sufficient for avoiding potential vulnerabilities that could emerge based on the economic design of the system or specific features of the blockchain such as MEV. We hence propose an extended version of the CASCADE framework that takes these considerations into account.

We propose to extend CASCADE with security considerations in three dimensions, at the system protocol, smart contract, and at the blockchain level. By **system protocol** we mean aspects of the application that are concerned with the economic mechanism design and other considerations that sit above the technical implementation. Such considerations may include e.g. the potential for rug pulls; ways to rig token supply or reward mechanisms that go against the spirit of the protocol, as outlined for our example of the decentralized prediction market in the previous section; flash loans, i.e. uncollateralized on-chain loans that are repaid within the same block, may be another potential source of vulnerabilities at the protocol level. Security considerations at the **smart contract** level are well covered by the ABCDE framework [21] and its companion paper [24]. ABCDE foresees including security considerations in three phases, design, implementation, and testing. The suggestions by ABCDE for the design phase include careful consideration of managing authorizations, planning of what kinds of mutability patterns to use for smart contracts and

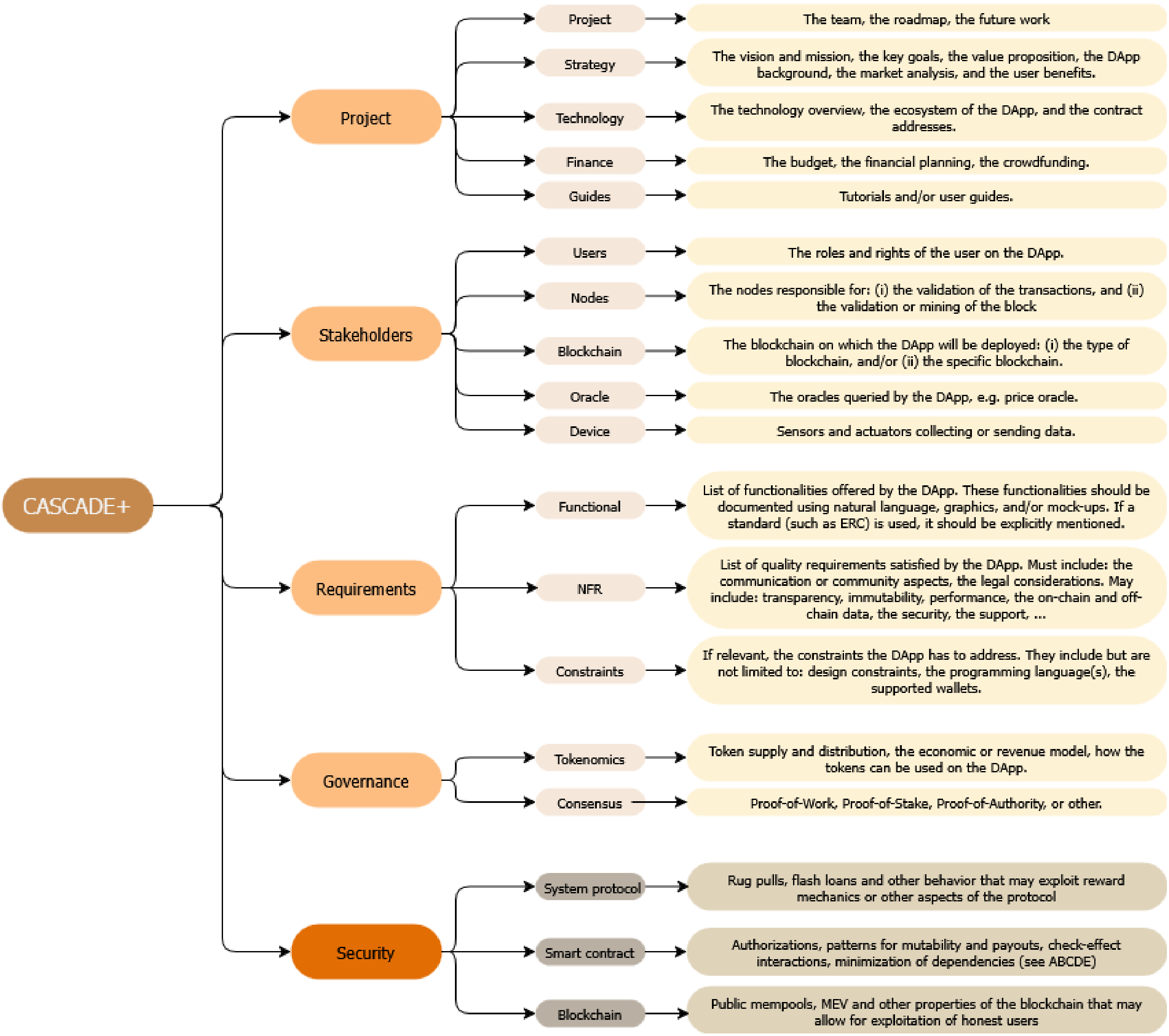
CASCADE+
Project
Project
The team, the roadmap, the future work
Strategy
The vision and mission, the key goals, the value proposition, the DApp background, the market analysis, and the user benefits.
Technology
The technology overview, the ecosystem of the DApp, and the contract addresses.
Finance
The budget, the financial planning, the crowdfunding.
Guides
Tutorials and/or user guides.
Stakeholders
Users
The roles and rights of the user on the DApp.
Nodes
The nodes responsible for: (i) the validation of the transactions, and (ii) the validation or mining of the block
Blockchain
The blockchain on which the DApp will be deployed: (i) the type of blockchain, and/or (ii) the specific blockchain.
Oracle
The oracles queried by the DApp, e.g. price oracle.
Device
Sensors and actuators collecting or sending data.
Requirements
Functional
List of functionalities offered by the DApp. These functionalities should be documented using natural language, graphics, and/or mock-ups. If a standard (such as ERC) is used, it should be explicitly mentioned.
NFR
List of quality requirements satisfied by the DApp. Must include: the communication or community aspects, the legal considerations. May include: transparency, immutability, performance, the on-chain and off-chain data, the security, the support, ...
Constraints
If relevant, the constraints the DApp has to address. They include but are not limited to: design constraints, the programming language(s), the supported wallets.
Governance
Tokenomics
Token supply and distribution, the economic or revenue model, how the tokens can be used on the DApp.
Consensus
Proof-of-Work, Proof-of-Stake, Proof-of-Authority, or other.
Security
System protocol
Rug pulls, flash loans and other behavior that may exploit reward mechanics or other aspects of the protocol
Smart contract
Authorizations, patterns for mutability and payouts, check-effect interactions, minimization of dependencies (see ABCDE)
Blockchain
Public mempools, MEV and other properties of the blockchain that may allow for exploitation of honest users

Fig. 1. Proposed framework for Web3 requirements engineering, based on CASCADE. The darker colored boxes correspond to the additions for security considerations.

for payouts. They also mention the importance of assumptions about transaction ordering. These considerations can be important for the viability of the project, as exemplified in our example of a decentralized prediction market for mathematics. The exact mechanics of how transaction ordering works depend on the **blockchain**, which is why we elevate this consideration to make it a point of its own. Features of the blockchain such as the existence of a mempool or private mempools, the possibility of tipping miners to prefer blocks can all have unforeseen impact on the working assumptions of a Web3 protocol, and addressing such vulnerabilities may require disruptive measures such as moving to a different blockchain or Layer-2 system. For this reason, we believe it is necessary to consider the interactions of blockchain characteristics with the planned Web3 protocol early on in the RE process.

Figure 1 shows our new extended framework, CASCADE+. It extends CASCADE [23] by adding security considerations at the system protocol, smart contract and blockchain level. These security considerations may interact with other considerations in the CASCADE framework, such as the blockchain part of the stakeholder component or the technology component. Our view is that security considerations should be conducted as a major step of its own, because requirements from a security perspective may be different than those from a stakeholder perspective. Treating both considerations as first class citizens allows for an iterated process to bring all these components in alignment.

VI. Limitations

While our RE analysis provides a systematic foundation for such a system, the proposed DApp has not been implemented. Instead, it serves as a proof-of-concept to demonstrate the applicability of existing RE4Web3 frameworks and to identify their limitations in practice. Future research could evaluate CASCADE+ in three key ways. First, the MVP could be developed and deployed to validate the requirements identified using the CASCADE+ framework. Second, CASCADE+ could be applied to other prospective DApps across different domains to assess the framework's generalizability beyond the proposed use case. Finally, CASCADE+ could be compared against the other RE4Web3 frameworks identified in Section III.

VII. Conclusion and Outlook

In this paper, we have conducted an extensive review of RE4Web3 frameworks. We have identified CASCADE [23] as the most fitting for an early-stage requirements-driven design analysis of a novel DApp, a decentralized prediction market for mathematical conjectures. By applying the CASCADE framework to a concrete project, we provide an evaluation of the theoretical framework in a practical context. This evaluation has shown that CASCADE is a useful framework for an early-stage RE process for Web3 projects that covers all relevant aspects in a structured manner - with one important exception, namely the lack of security considerations. We have applied a security analysis based on the recommendations of the ABCDE framework [21], which naturally extends CASCADE from the early-stage RE into the implementation phase. Our analysis has shown, however, that even ABCDE's security analysis is too narrowly focused on smart contract implementation aspects. We have identified potential vulnerabilities that need to be addressed in the design, stemming from features of the blockchain itself, namely MEV, or that emerge purely from the system protocol design. These vulnerabilities could support denial-of-service attacks or undermine the economic incentive model of the system protocol. Addressing these vulnerabilities may require choosing an appropriate blockchain or potentially complex technical solutions such as a Layer-2 chain added above the original blockchain. An analysis that focuses on the smart contract aspects of the system implementation would likely have missed these vulnerabilities.

Based on these findings, we have proposed a new framework for early-stage RE4Web3, including a holistic security analysis at the system protocol, smart contract, and blockchain level. Our framework, CASCADE+, builds on and extends CASCADE with these components. The security analysis at the smart contract level corresponds to that of the ADCBDE framework. We suggest extending the analysis to include system protocol-level vulnerabilities, such as flash loans and rug pulls, and blockchain-level vulnerabilities, such as MEV. A full description of the extended framework is left for future work.

Another aspect of future work revolves around the design of the decentralized mathematics prediction market itself. We have considered a 'minimum-viable product' implementation of such a system protocol for exposition purposes. Such an MVP design could be extended in multiple directions in future work. A critical limitation of the MVP design is the lack of a protocol native token. Without a native token, there is no clear mechanism for decentralized protocol governance, which is necessary for evolving the project in the future while maintaining its decentralized character. Introducing a token requires essential considerations, such as the emission schedule, potential uses and sinks for the token, and ensuring they do not create vulnerabilities, e.g., by incentivizing unwanted behavior. Another direction for extending the MVP system protocol is to enable multiple markets per conjecture, e.g., providing separate markets for the different possible outcomes of a conjecture (false, true, or undecidable under the given set of axioms). Multiple markets may introduce the potential for arbitrage opportunities, which should be carefully considered in an RE analysis.

References

[1] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson, "Uniswap v3 core," *Tech. rep., Uniswap, Tech. Rep.*, 2021.

[2] R. Leshner and G. Hayes, "Compound: The money market protocol," *White Paper*, vol. 89, 2019.

[3] J. Peterson, J. Krug, M. Zoltu, A. K. Williams, and S. Alexander, "Augur: a decentralized oracle and prediction market platform," *arXiv preprint arXiv:1501.01042*, 2015.

[4] K. Mattmuller, "Decentralized prediction markets," *Geo. L. Tech. Rev.*, vol. 8, p. 384, 2024.

[5] R. Hanson, "Could gambling save science? encouraging an honest consensus," 1995.

[6] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in *International conference on principles of security and trust*. Springer, 2017, pp. 164–186.

[7] H. Zhou, A. Milani Fard, and A. Makanju, "The state of ethereum smart contracts security: Vulnerabilities, countermeasures, and tool support," *Journal of Cybersecurity and Privacy*, vol. 2, no. 2, pp. 358–378, 2022.

[8] G. Wu, H. Wang, X. Lai, M. Wang, D. He, and S. Chan, "A comprehensive survey of smart contract security: State of the art and research directions," *Journal of Network and Computer Applications*, p. 103882, 2024.

[9] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges," *arXiv preprint arXiv:1904.05234*, 2019.

[10] M. Vladimirov and D. Khovratovich, "Erc20 api: an attack vector on approve/transferfrom methods," *TransferFrom methods*, 2018.

[11] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*. Springer, 2002, pp. 251–260.

[12] V. Gramlich, D. Jelito, and J. Sedlmeir, "Maximal extractable value: Current understanding, categorization, and open research questions," *Electronic Markets*, vol. 34, no. 1, p. 49, 2024.

[13] S. Porru, A. Pinna, M. Marchesi, and R. Tonelli, "Blockchain-oriented software engineering: Challenges and new directions," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 2017, pp. 169–171.

[14] C. Sillaber and B. Waltl, "Life cycle of smart contracts in blockchain ecosystems," *Datenschutz und Datensicherheit*, vol. 41, no. 8, pp. 497–500, 2017. [Online]. Available: https://doi.org/10.1007/s11623-017-0819-7

[15] M. Fahmideh, J. Grundy, A. Ahmad, J. Shen, J. Yan, D. Mougouei, P. Wang, A. Ghose, A. Gunawardana, U. Aickelin, and B. Abedin, "Engineering blockchain-based software systems: Foundations, survey, and future directions," *ACM Comput. Surv.*, vol. 55, no. 6, Dec 2022. [Online]. Available: https://doi.org/10.1145/3530813

[16] S. Chawla, “Goal oriented requirements engineering for blockchain based food supply chain,” *International Journal of Software Engineering and Computer Systems*, vol. 6, pp. 94–103, 08 2020.

[17] C. WaiShiang, M. T. LiBin, E. Phang, N. b. Jali, and M. A. bin Khairuddin, “eaom: Extended agent-oriented modeling as an alternative methodology for blockchain enabling application development,” *Journal of Software: Evolution and Process*, vol. 36, no. 5, p. e2610, 2024. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2610

[18] J. Mantas, L. Čeponienė, K. Butkus, R. Butkienė, and V. Drungilas, “Mda-based approach for blockchain smart contract development,” *Applied Sciences*, vol. 13, no. 1, p. 487, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/1/487

[19] A. Samanipour, O. Bushehrian, and G. Robles, “Mdapw3: Mda-based development of blockchain-enabled decentralized applications,” *Sci. Comput. Program.*, vol. 239, no. C, p. 28, Jan 2025. [Online]. Available: https://doi.org/10.1016/j.scico.2024.103185

[20] R. France and B. Rumpe, “Model-driven development of complex software: A research roadmap,” in *Future of Software Engineering (FOSE ’07)*, 2007, pp. 37–54.

[21] L. Marchesi, M. Marchesi, and R. Tonelli, “Abcde - agile block chain dapp engineering,” *Blockchain: Research and Applications*, vol. 1, no. 1, December 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2096720920300026

[22] G. Lallai, A. Pinna, M. Marchesi, and R. Tonelli, “Software engineering for dapp smart contracts managing workers contracts,” 03 2020.

[23] S. Bouraga, “Cascade - framework for the early-phase development of blockchain-based applications,” *Business Information Systems Engineering*, Jan. 2025. [Online]. Available: https://doi.org/10.1007/s12599-024-00912-4

[24] L. Marchesi, M. Marchesi, L. Pompianu, and R. Tonelli, “Security checklists for ethereum smart contract development: patterns and best practices,” *arXiv preprint arXiv:2008.04761*, 2020.

[25] J. Köpke, G. Meroni, and M. Salnitri, “Designing secure business processes for blockchains with secbpmn2bc,” *Future Generation Computer Systems*, vol. 141, pp. 382–398, 2023.

[26] M. Baldauf, E. Sonnleitner, and M. Kurz, “Exemplary ethereum development strategies regarding security and gas-saving,” *Electronics*, vol. 13, no. 1, p. 117, 2023.

[27] A. Eraig, R. A. M. Khalid, and L. Abdelgader, “Blockchain and cybersecurity: Addressing smart contract vulnerabilities in decentralized applications,” *Journal of International Crisis and Risk Communication Research*, vol. 7, no. S8, p. 1280, 2024.

[28] M. Barboni, A. Morichetta, and A. Polini, “Smart contract testing: challenges and opportunities,” in *Proceedings of the 5th International Workshop on Emerging Trends in Software Engineering for Blockchain*, 2022, pp. 21–24.

[29] “Rocq,” https://rocq-prover.org/, accessed: 27 May 2025.

[30] “Isabelle,” https://isabelle.in.tum.de/, accessed: 27 May 2025.

[31] “The e theorem prover,” https://wwwlehre.dhbw-stuttgart.de/ sschulz/E/E.html, accessed: 27 May 2025.

[32] “F* a proof-oriented programming language,” https://fstar-lang.org/, accessed: 27 May 2025.

[33] “Lean programming language and theorem prover,” https://lean-lang.org/, accessed: 27 May 2025.

[34] L. Lamport, “Constructing digital signatures from a one way function,” 1979.

[35] “Flashbots,” https://www.flashbots.net, accessed: 27 May 2025.