Institut für Kommunikationsnetze und Rechnersysteme (IKR)

Universität Stuttgart

Research Project

DLT-Based Credit and Compensation System for Open-Source Resources

Sheetal Kamthe

Dr. Jürgen Jähnert

16.10.2023

Table of Contents

List of Figures

List of Tables

1 Introduction

Data stands as the third most important strategic resource, after resources and energy. Currently, the growth of the big data industry is facing a challenge known as the "data island." In this environment, where everyone has data, but at the same time faces a lack of data, the right solution is to create a clear and effective data sharing and monetisation model. The main challenge presented in this research project is about the concept of securing ownership and defining terms of use within the open source sharing environment. In the context of open source sharing, it becomes important for resource providers to not only receive recognition for their contributions but also to have control over how their work is used. To solve this challenge, we propose the development of a governance model that defines clear terms and conditions for the use of shared resources and mechanism to establish value of these resources. The model should provide clear guidelines on who can access these resources, under what conditions, how content creators can enforce their rights and receive compensation.

The research follows a scenario based design [1] approach and is layed down as chapters, wherein the second chapter crafts a scenario and extracts requirements from this scenario. The third chapter provides a general overview of the technologies involved in the project. In the next Chapter we present the arguments in favor of the adoption of DLT and examine the compatibility of requirements with appropriate technological solutions. Chapter five presents criteria for evaluating our research scenario and makes a comparative evaluation of possible solutions. The next chapter delves into the practical aspects of implementing our research scenario. Chapter seven examines the generalizability and applicability of our research by considering similar requirements in different domains. The final chapter provides a general overview of the project, highlighting key findings and contributions.

Thus through this well-defined methodology we aim to provide valuable insights and find solutions to address the challenges mentioned at the beginning of Introduction.

2 Scenario-Based Design [1] Approach

2.1 Storytelling as a Design Tool

Storytelling and scenarios are the most important tools in the design process of a Software system. They help designers understand user experiences, needs, and pain points by embedding them into relevant stories. By creating stories and scenarios, designers can empathize with users, identify design needs, and create user-centered solutions, encouraging creativity and design usability.

2.2 Story or Scenario Summary

An educational software platform referred to as TechVerse in our research is a platform wherein learners take advantage of accessing the learning material uploaded by teachers. Betty, a character in our scenario, is the platform's lead developer and seeks to enhance its user experience and expand its offerings to meet the evolving needs of its users. To achieve this, the TechVerse development team decided to adopt a scenario-based design requirements engineering approach to identify, prioritize, and address user requirements effectively.

Meet our key characters and a day in their Life:

- **Professor Turner:** A dedicated and innovative university professor who believes in the power of collaboration and open education.
- **Archie:** An ambitious and curious student, passionate about contributing to the advancement of technology.
- **Liam:** An enthusiastic learner who is eager to explore trending topics.
- **Betty:** The platform's lead developer and a firm believer in the potential of DLT to revolutionize learning.

Chapter 1: A Day in the Life of Professor Turner (Content Producer Perspective): Professor Turner starts her day by creating a comprehensive lecture on Industrial Mechanics. Using the platform, she uploads a series of engaging video lectures, interactive exercises, and reading materials. She is assigned the role of a teacher on the platform and has access to uploading teaching material with answers to exercises to be viewed later and only by teachers. Prof Turner wishes to define rules on who can access the content and prevent unauthorized manipulation. Also, the Prof wants to grade/reward the contributions and also get credits/ownership for her content. If someone uses her content and gets paid for it, she wants to be notified about it since her motive is to make the content available for free only to the students.

Chapter 2: A Day in the Life of Archie (Content Contributor Perspective) Balanced Prosumer: Archie is excited to delve into Professor Turner's industrial mechanics lectures. He tries solving the exercises and finds the content insightful. He decides to contribute to it by adding additional real-world examples and relevant research but the role of the student assigned to Archie does not allow him to modify content. He approaches Prof Turner on his ideas of contribution. Prof Turner likes the idea since it will improve the content but the problem of authenticity has to be dealt with. On contributing to such open-source content, Archie will receive rewards to be used elsewhere.

Chapter 3: A Day in the Life of Liam (Unbalanced Prosumer): Liam is neither a student nor a teacher but is interested to learn on the trending topics and observing other talented contributors in creating content. Since Liam has no contributions, he has no credits and cannot access the premium content. Therefore Liam has to use a payment option on the platform to access the content.

Chapter 4: Role of Betty: Betty has to keep both learners and educators in mind. Betty is responsible for the performance of the platform, we could say this is a third perspective towards the platform. Providing current features for uploading and viewing the content, Betty also has ideas to gather insights into user engagement and platform performance.

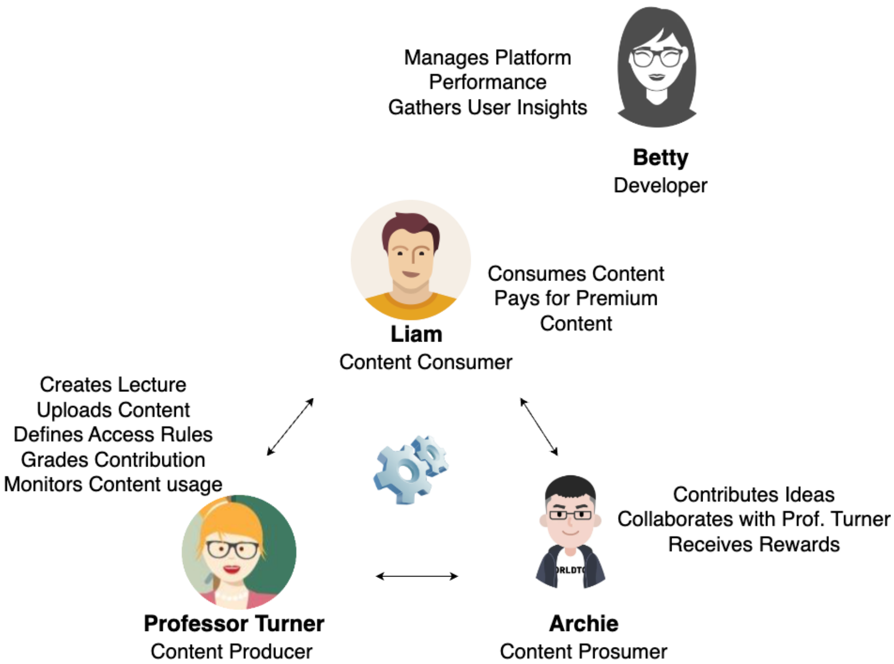

Figure 1: *Scenario Summary*

2.3 Requirements Elicitation from Scenario

In this Chapter, we analyze the specifics to identify and extract important requirements that will be directly included in the software system during its development. The requirements are further subdivided into Hardcoded and Configurable to understand the flexibility of the system.

- **Requirements to be Hardcoded During Development:**

  Requirements for User Roles and Permissions: In the current platform, Betty has administrative privileges to manage user accounts. There is a need for the platform to support multiple user roles in a decentralized manner. Therefore a mechanism is needed to identify the user and assign roles.

  Requirements for User Authentication: A robust and secure user authentication mechanism is required because Professor Turner wants to prevent her content (if made available globally) from unauthorized modifications. Therefore User authentication tokens shall be securely stored and managed using decentralized identity solutions.

  Requirements for Content Authentication Mechanisms: A decentralized authentication mechanism is needed to ensure the integrity and immutability of uploaded content.

  Requirements for Credit: The platform shall have a native token or credit system implemented using decentralized technology. Archie shall be rewarded with platform credits for contributing valuable content that gets approved by Prof. Turner and positively rated by other Users.

  Requirements for Transparent Content Ratings and Reviews: Users, including Archie and Liam, shall be able to rate and review the educational content available on the platform. Since Archie is a valuable contributor his rating may have more weightage(need a mechanism) than Liam.

  Requirements for Content Authorization Mechanisms: The amount of credit received for the content will be decided on the votes received by all the users authorized to access the content.

- **Requirements to be Configurable:**

  Requirements for Content Ownership:

  1. Access to the content created - global or restricted
  2. Copyright and Licensing

3. Respect for Usage Rights
4. Monetary benefits - for globally accessible content - implement measures to prevent unauthorized users from earning monetary benefits from the content

Requirements for Content Contribution and Modification:

1. Secure Timestamping and User Attribution - some form of points/credits to be received for contributions from users like Archie
2. Content Modification Approval - from the owner of the content (Requirement to be Hardcoded During Development)
3. Decentralized Dispute Resolution - For instances where content contributions by users are found to be inappropriate or inaccurate after approval, the platform shall implement a decentralized dispute resolution process, involving other Users, to address and rectify the issues. (Requirement to be Configurable)

Requirements for Payment Mechanism:

1. Access to Premium Content - if a user like Liam has no credits he shall have the option to purchase additional credits using cryptocurrencies or fiat through secure and transparent payment channels. (Requirement to be Hardcoded During Development)
2. Support for Payment NGOs/other non-profit organizations - users like Liam choose to pay to any one of them to gain access

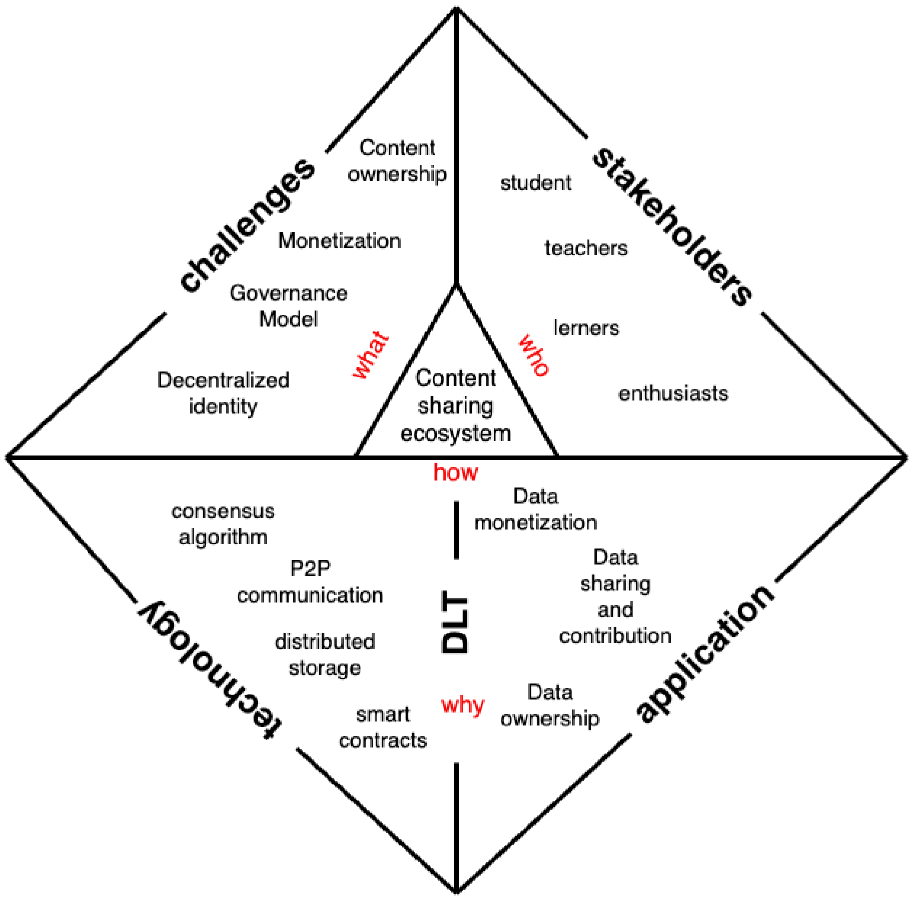

Figure 2: *A quick overview of scenario challenges*

3 Technology Overview

In this chapter, we will provide an introduction to DLT, a history of how it came into existence, what problems it solves, and an introduction to the major DLT technologies.

3.1 Introduction to DLT

In the dynamic landscape of today's digital world, the combination of technology and innovation has given rise to transformative solutions that redefine interactions, transactions, and data exchange. Distributed ledger technology (DLT), an innovative paradigm that traces its roots to millennia of development, has garnered much attention for its potential to reshape industries across the spectrum. Imagine an area where transactions receive validation not from a central authority but from a network of participants cooperating in a decentralized manner. Imagine a situation where the protection of data goes beyond just one strong defense and gets even stronger with the help of complicated cryptographic methods. Picture a landscape where transparency and accountability are intrinsic virtues rather than distant ideals. Our journey of discovery begins by exploring the fundamental concepts of DLT. We'll highlight the key principles that anchor this phenomenon, diving deep into cryptography, consensus mechanisms, and decentralization principles. However, our campaign does not end there. It will move beyond the limitations of traditional blockchains to explore alternative technologies like Tangle, each bringing a unique approach to decentralization and consensus. From protecting digital identities to redefining content ownership, enabling tamper-proof voting to ushering in new models of economic transactions, these technologies are the foundation of trust, security, and collaboration in the digital age.

Distributed ledger technology(DLT) is essentially a database that is shared amongst computers spread around the globe, creating a decentralized environment rather than a centralized one [2]. It enables multiple participants to maintain and update a shared digital record in a secure and transparent manner.

Many different types of distributed ledger technology exist, and ongoing innovation is likely to lead to improvements and the creation of new iterations within the existing landscape. DLT encompasses a variety of technologies and protocols, with blockchain being one of the most well-known implementations. It's vital to understand that blockchain tech improved upon DLTs because it was the first system to solve the double-spending problem [3].

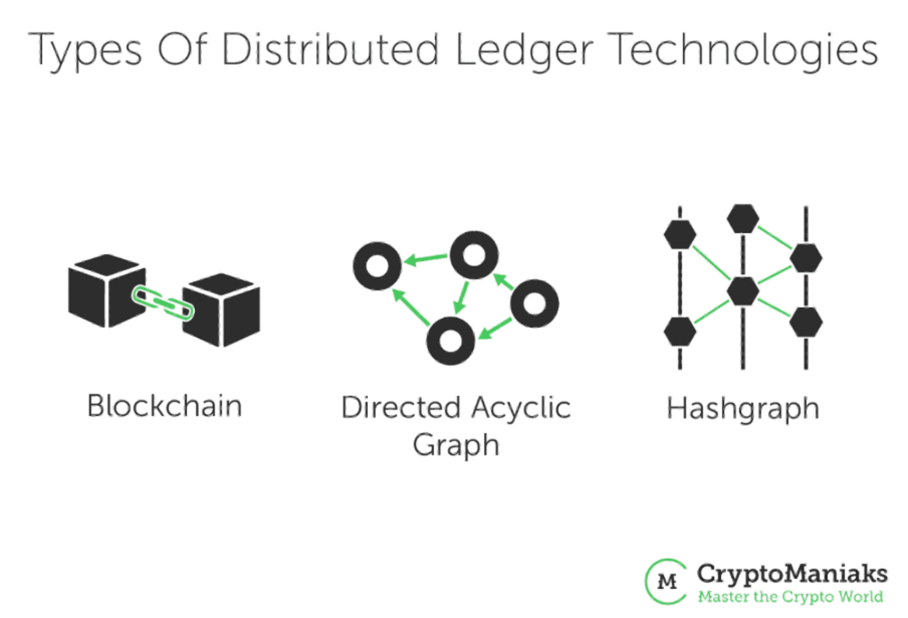

Figure 3: *Types of DLT*
[2]

Each of them has its advantages and disadvantages, but all offer the same solution – a fast, reliable, and secure layer for the transaction of value and data [4]. A distributed ledger emerges as a consensus formed among replicated, synchronized, and collaboratively accessible digital information that spans across a network of peers [5].

3.2 A Brief History[6]

1991: A cryptographically secured chain of blocks is introduced for the first time in the paper "How to Time-Stamp a Digital Document" by Stuart Haber and W Scott Stornetta.

1998: Computer scientist Nick Szabo works on ‘bit gold’, a decentralized digital currency.

2000: Stefan Konst contributes his theory of cryptographic secured chains and also some ideas of implementation.

2008: Operating under the pseudonym Satoshi Nakamoto, developers present a white paper outlining the foundational principles of a blockchain.

2009: Nakamoto brings the first blockchain to life(Bitcoin), serving as a public ledger for Bitcoin transactions and ingeniously solving the decentralized Double Spend Issue: the issue that digital information can be easily copied, and therefore previously required a centralized authority to indicate where funds were located.

2014: The Ethereum blockchain system introduces computer programs into the blocks known as smart contracts.

2021: Web3 is an idea for a new iteration of the World Wide Web which incorporates concepts such as decentralization, blockchain technologies, and token-based economics.

3.3 Blockchain

Blockchain is a contemporary form of distributed ledger technology (DLT) as opposed to traditional centralized databases. Unlike secure, closed systems that limit access to authorized operators, blockchain acts as a digital ledger distributed across a global network of computers. To begin, let’s grasp the meanings of "block" and "chain." "Block" signifies a collection of sequentially ordered digital records. These blocks of digital records are interconnected in a linear fashion, ensuring security through cryptographic hash functions for example SHA-256, all being stamped with timestamps. A block, essentially a bundle of data, undergoes mining processes to conform to the block structure and ensure its security. When creating a new block, it includes a hash of the preceding block, thus establishing a sequential chain from the initial block to the most recent one. This cycle continues as new blocks are added, thus sustaining the network [7] Fig.4. The process of adding a new block to a blockchain involves collecting and verifying data, reaching consensus through mining, structuring the block with data and a timestamp, linking it to the previous block through hashing, and confirming consensus across the network. This systematic process ensures the security, integrity, and continuity of the blockchain's data structure. Let us now look at some examples of successfully implemented Blockchains.

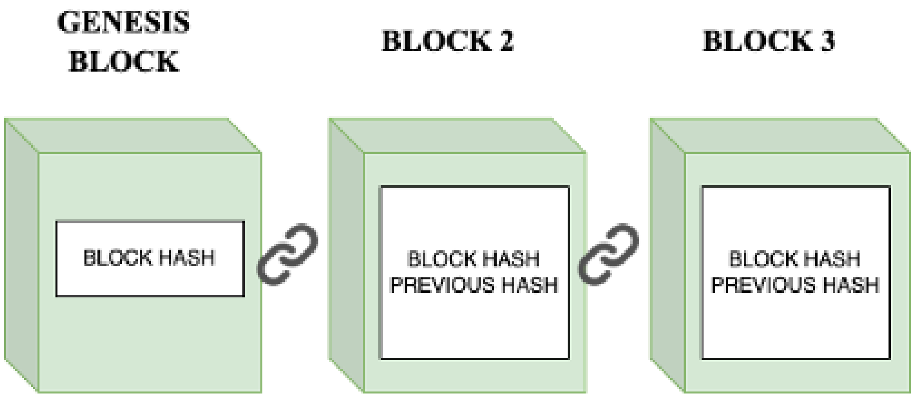

Figure 4: *Blockchain*

3.3.1 Bitcoin

Bitcoin operates on a Blockchain and is a decentralized digital cryptocurrency that is created by the mining process at a less predictable rate. The number of new bitcoins created each year will automatically halve over time until bitcoin issuance stops completely with a total of 21 million bitcoins in existence. Bitcoin splits into 100 million satoshi, meaning one satoshi equals 0.000000001 bitcoin or 100 million satoshi equals 1 bitcoin [8]. In the cryptocurrency world, dust refers to any little coin or token. It is generally so small that it is often forgotten or ignored. So, you can refer to an amount less than a few hundred satoshis as dust. These amounts are often smaller than the transaction fees required to transfer them! While Bitcoin has proven to be very successful in facilitating value transfers and acting as a decentralized currency, it was clear from the start that this approach could be extended to include a wider spectrum of information types on a common platform. In addition to its role in managing currency, blockchain technology enables the recording of various data forms – from basic messages and ownership details of assets, both physical and digital, to securities, voting results, and more.

3.3.2 Ethereum

While Bitcoin is a complex system that solves a simple problem — how to store and send value — Ethereum retools the Bitcoin design to enable the blockchain to tackle many more challenges. After Bitcoin was established as a currency between 2010 and 2013, enthusiasts began envisioning a decentralized organization and solving other problems using blockchain. While Bitcoin can only execute simple commands, such as "send," in theory, a similar system could run all kinds of complex programs. These programs are known as smart contracts. A system that hosts smart contracts is known as a smart contract platform. In the same way that Bitcoin is like a decentralized bank, a smart contract platform is like a decentralized app store. Similar to mobile phone app store platforms that host software apps developed by third parties, a smart contract platform can host applications (lines of code) created by third parties that are uploaded to the blockchain. Anyone can access and run these apps without asking permission. Unlike centralized app stores, developers do not need approval to upload their programs. Once deployed, apps cannot be forcibly removed or censored. Ethereum is the leading smart contract platform [9].

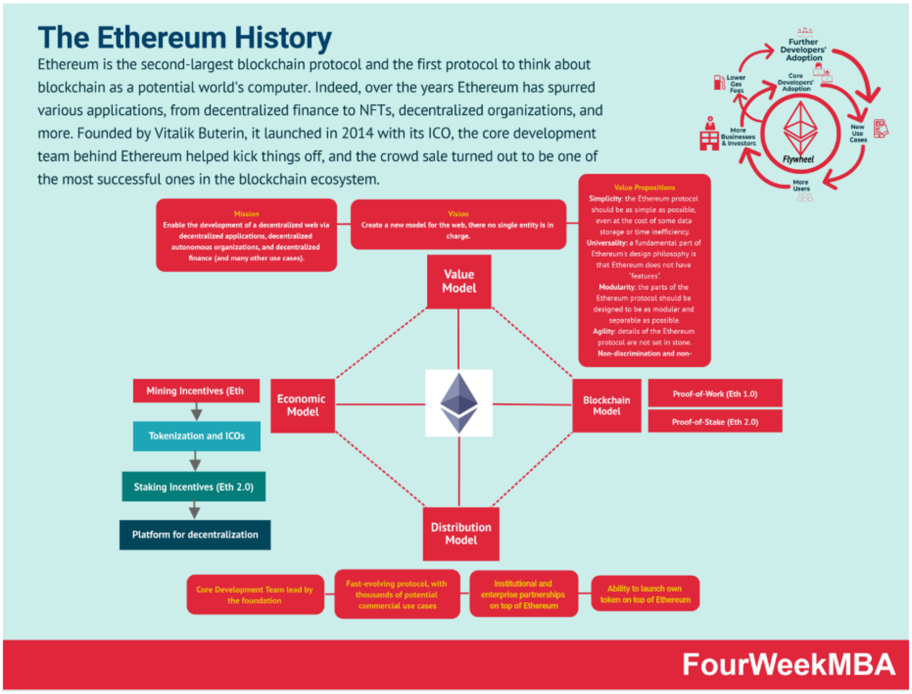

Figure 5: *History of Ethereum*
[10]

Ethereum distinguishes itself from earlier cryptocurrencies by aspiring to be a decentralized app store. It serves as a platform for executing smart contracts. At its core, Ethereum comprises four crucial layers:

1. Physical Network (First Layer): A distributed network of computers, or nodes, runs Ethereum software. Some nodes mine cryptocurrency, while others propagate the blockchain and transactions globally.
2. Ethereum Virtual Machine (Second Layer): This layer enables developers to upload smart contracts to the blockchain and allows users with external accounts to interact with these contracts. These contracts are essentially computer programs, stored redundantly across all nodes for resilience.
3. Smart Contracts (Third Layer): These are programs that receive and send instructions to other accounts or contracts. They can trigger actions like altering variables or transferring cryptocurrencies.
4. Decentralized Applications (Fourth Layer - Dapps): Dapps consist of collections of smart contracts that perform complex tasks, accessed through websites connecting users "Web 3" wallets to the Ethereum network.

An Ethereum transaction involves sending messages across the network. Users utilize wallet software to send messages from their addresses to other users or smart contracts. Transactions can be as straightforward as Ether transfers or more intricate, like running specific code or recording data. Users include a transaction fee denominated in Ether with their message. Nodes on the network listen for messages, which are then grouped into blocks by miners. Miners process and add these blocks to the blockchain, earning rewards and transaction fees in return. Transaction fees, known as "gas" fuel the network's operations and ensure security by deterring spam and infinite loops. Ethereum's architecture empowers advanced functionalities like running intricate smart contracts, setting it apart from simpler cryptocurrencies like Bitcoin.

"Given these dynamics, if Bitcoin can be viewed as digital gold for holding, Ether can be viewed as digital oil for burning [9]."

Currencies prioritize the exchange of value, smart contract platforms enable customized programming, and DApps provide functional applications. Each group's progress is shaped by its unique competitive strengths, technological advances, and network effects, which contribute to the ever-evolving blockchain ecosystem [11].

3.4 Tangle

Although the blockchain is a popular form of DLT, the costs associated with the use of mining processing power, as well as the limited transaction throughput, make many blockchain solutions unsuitable for some applications. DLTs using a Directed Acyclic Graph (DAG) topology, address these issues by storing transactions directly in a directory, which allows for a parallel chain of transactions, increasing throughput, and scalability. DLTs based on DAGs also eliminate mining, enabling instant transactions and lighter infrastructure due to reduced energy requirements. The IOTA Tangle, [12] a DAG-based DLT, is specifically targeting IoT networks and devices. The Tangle network(see Fig.6) consists of nodes that use P2P communication to broadcast messages and transactions. These nodes process and verify messages themselves, maintaining the state of the Tangle ledger. The DAG structure of the Tangle ledger provides increased scalability and transaction throughput. In Tangle, connectivity is shown as a series of connected nodes in a network-like arrangement. Each node is linked to two previous nodes, forming a pattern in which new tasks reinforce existing ones. As more nodes are added, the network design becomes more efficient and secure. The concept of "cumulative weight" is required, where computational effort is spent confirming that connections with higher cumulative weights are more reliable, generally consisting of older transactions, and therefore network security increases.

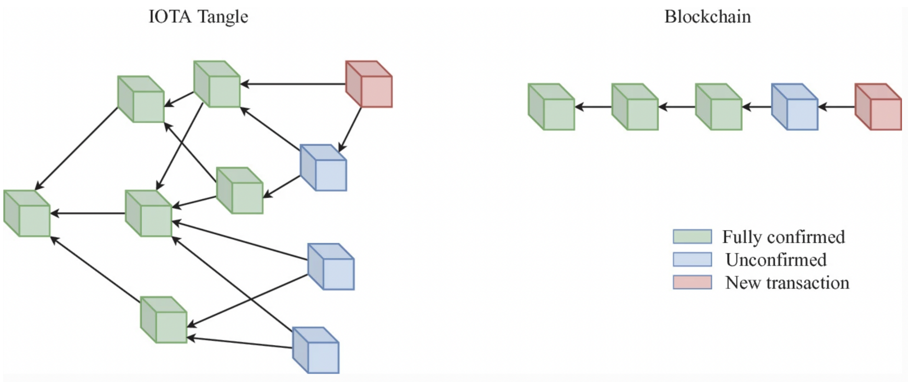

Figure 6: *Tangle*
[13]

IOTA's development network, IOTA 2.0, represents the future of the protocol. It greatly improves upon legacy protocols, supporting smart contracts and digital asset management. The protocol is fully decentralized, eliminating the need for a coordinator and improving scalability. The Ledger, Networking, and Consensus papers are the three cornerstones of IOTA 2.0 [14] [15] [16] [17].

Overall, DLT, especially in the form of IOTA 2.0, holds great promise for decentralized applications, providing the necessary features for a secure, scalable, and efficient implementation. It enables real-time data and value exchange in areas such as transportation, supply chain, healthcare, and energy. Overall, Tangle's new DAG system and associated features position itself as a promising alternative to traditional blockchains, solving many of the challenges of existing systems.

4 Analysis of Requirements and Technology Mapping

It is of the utmost importance to analyse if DLT is the highest level candidate as a solution to the problem at hand. This chapter addresses the question of why DLT should be used in our scenario and map technologies to the requirements extracted from the scenario.

4.1 Arguments for the Adoption of DLT

Out of the many considerations when determining whether a problem could be solved using DLT, following are the major three with respect to our scenario:

- Decentralization Requirement: In the current platform, Betty(persona) has complete access to the database and hence to the content , a trustless network is needed where multiple or any user(s) can participate without relying on the central authority.
- Immutable and Tamper Resistant Records: Content producers are willing to share their content as open source in order to make it available to a larger community and help in further development of society. But in a centralized ecosystem user is not aware on the content usage and some other entities incurring profit out of it.
- Smart Contract Functionality: DLT offers a feature for automating processes and agreements. A fair mechanism is needed to establish trust amongst users sharing data and making it available to the entire ecosystem as well as incurring benefits if any.

4.2 Requirements Technology Mapping

- Requirements for User Roles and Permissions:
  - **Technology:** Smart contract is an immutable program stored on DLT.

  - **Description:** The use of smart contracts to define and enforce the usage. Everyone's transactions can be encoded in a DLT, ensuring decentralized and tamper-resistant access control.

- Requirements for User Authentication:
  - **Technology:** Decentralized identification solution
  - **Description:** Implementation of a decentralized identity solution that allows users to securely authenticate without relying on a central authority. This can be done with universal interference or other DLT-based detection systems.

- Requirements for Content Authentication mechanism:
  - **Technology:** Store off chain (IPFS) is a distributed file storage network
  - **Description:** Content hashes or digital signatures stored in on chain to ensure the integrity and immutability of content. This can provide you with a transparent and inflexible way of verifying uploads. Example smart contract functions are: –storeContentHash –verifyContentHash

- Requirements for Content Authorization Mechanisms:
  - **Technology:** Smart contract is an immutable programs stored on DLT.
  - **Description:** Smart contract will define functions like –hasAccesstoContent –grantAccesstoContent –deductTokens –insufficientfundoptions

- Requirements for Content Ownership:
  - **Technology:** Smart contracts and licenses or NFT [1]
  - **Description:** Using smart contracts to define and execute copyright laws and licensing educational materials. This respects the rights of developers and ensures that users comply with defined user rights. Example smart contract functions are: –assignContentOwnership –hasRightsToContent –grantContentRights

- Requirements for Content Contributions and Modifications:
  - **Technology:** This would be a combination of Content Authentication + Content Authorization + Decentralized voting and consensus executed in sequence.

- Requirements for Payment Mechanism:
  - **Technology:** Cryptocurrency payments and smart contracts
  - **Description:** Users are allowed to purchase credits using cryptocurrencies(Cross chain asset transfers) using secure and transparent payment methods using DLT smart contracts.

- Requirements for Credit:
  - **Technology:** Native token[2] and smart contracts
  - **Description:** To use a native token or credit system using DLT-based tokens. Smart contracts can be used to generate the reward distribution process based on the approval of features and the number of users. Example smart contract functions are: –purchaseCredits –getUserCredits
- Requirements for Transparent Content Ratings and Reviews:
  - **Technology:** Token weightage and decentralized governance
  - **Description:** A token or reputation system used to assign weight to the number of users. Decentralized governance structures allow users to collectively decide on weighting algorithms, and ensure fairness and transparency.

5 Evaluation

Considering the wide scope of Distributed Ledger Technology (DLT) and its various applications, as described in Chapter 3 which outlines the key DLT technologies, it becomes a great challenge to comprehensively analyze the multitude of smart contract platforms available. Therefore, this chapter focuses on evaluation of several major platforms that offer key features based on criteria for the requirements extracted from our scenario in Chapter 4.2. Due to time constraints, we chose not to consider energy consumption as a primary measure of our evaluation. The assessment of energy consumption in the context of blockchain and smart contract platforms involves the analysis of many aspects, including the use of clean energy sources and the possible use of dirty energy, among other factors. Acknowledging the complexity and seriousness of this issue, we have passed it on to future research trials where it can receive the careful attention it deserves.

5.1 Evaluation Criteria for the Scenario

The criteria listed in this section represent only a fraction of the many criterias involved in selecting a DLT and Smart Contract Platform for a project. Due to the time constraint, I hereby restrict myself with the following criterias. These criteria provide a preliminary framework for our decision-making process.

- Enhanced Security and Scalability

- Robust Ecosystem and Ample Liquidity
- Dependable Consensus Mechanism
- Sustainable Transaction Throughput and Cost-Efficiency
- Comprehensive Development Toolkit and Smart Contract Languages
- Tokenization
- Transparent Leadership and Future Roadmap

5.2 Comparative Evaluation

We consider top 7 Smart contract platforms(Blockchain) and IOTA and evaluate each of them based on above 7 criteria's mentioned. The values for these criteria are extracted from a variety of sources, such as: The platform's website and documentation, Whitepapers and technical papers, Blog posts and articles, Social media posts, Community forums and chat rooms, Third party websites and tools.

We use a radar chart, given its ability to accurately display multidimensional data, to visually represent the performance of the top 7 smart contract platforms(Blockchain) as well as IOTA(DAG), considering the 7 criteria mentioned. By evaluating each platform based on these criteria summarized in Table 2, it becomes clear that choosing the platform that best suits our situation is a difficult task. Considering the blockchain trilemma[3], it is clear that smart contract platforms often make trade-offs between its three components. For example, some platforms prioritize achieving greater transaction throughput at the expense of compromising decentralization. In our case, decentralization is a non-tradeable option, leading us to exclude BNB Chain, Solana, and Avalanche from our consideration. Additionally, Cardano's use of Marlowe and Plutus [18] smart contract languages presents a steep learning curve for smart contract development, making it less desirable for our scenario. In addition, Tron is terminated due to concerns related to the reputation of its founder. As a result, we narrow our focus to Ethereum, Polygon, and IOTA. These platforms are thoroughly analyzed and represented on the radar chart to assess their performance across criteria. Finally, our choice is drawn to IOTA, mainly because of its feeless transactions, scalability, and its compatibility with the Ethereum Virtual Machine (EVM) and smart contracts.

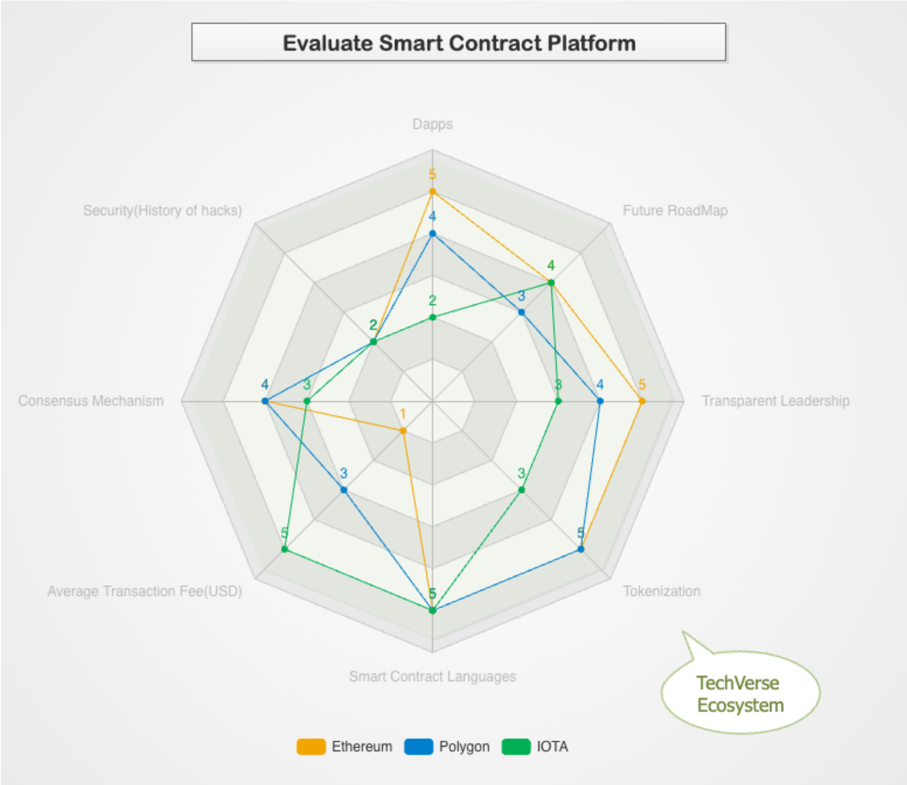

Figure 7: *Evaluation of Smart Contract Platform*

Ethereum and other blockchain platforms that offer enhanced security through gas fees are highly recommended for use cases like Real Estate, Notary Services, Financial Services etc. However, for implementing the specific scenario outlined in this paper, which involves executing transactions for each action, it may not be the most suitable choice for future implementations. This conclusion is supported by an example from the paper [19]. In this example Table 1, it becomes evident that the cost of a single transaction, averaging at 0.1 USD, is prohibitively high considering the number of transactions a single user performs as per our requirements.

Function	Transaction Cost (Gwei)[b]	Execution Cost (Gwei)[b]	Total Cost (Gwei)[b]	Total Cost (Ether)[a]	Total Cost (USD)[c]
Enrolling a Student	22680	49560	72240	0.00007224	0.037
Uploading a Credential	33560	189574	223134	0.000223134	0.114
Company retrieving a Credential	24472	134027	158499	0.000158499	0.081
Student sharing a Credential	35032	189699	224731	0.000224731	0.115
Set public key on blockchain	26264	2154	28418	0.000028418	0.015

Table 1: Cost of Important Functions

[a] Ether is the native cryptocurrency for Ethereum 1 ETH = $1,523.84 USD 16.10.2023 [b] Gwei is a small unit of Ether (ETH). 1 Gwei = 0.000000001 or 10^{-9} ETH [c] USD is official currency of the USA

Platform	Security (History of hacks)[a]	DApps[b]	Consensus Mechanism[c]	Avg. Transaction Fee (USD)[d]	Smart Contract Languages[e]	Tokenization	Transparent Leadership and Future Roadmap[f]
Ethereum	Yes[20]	4313	PoS	0.49[21]	Solidity, Vyper	Yes	5
BNB Chain	Yes[22]	4986	PoSA	0.0854[23]	Solidity, Vyper	Yes	2 (since centralized)
Cardano	No	42	PoS	0.16[24]	Plutus, Marlowe [18]	Not yet	4
Polygon	Yes[25]	1827	PoS	0.02[26]	Solidity	Yes	4
Solana	Yes[27]	192	PoH	0.0002[26]	Rust, C, C++, Python	Yes	2 (since centralized)
Avalanche	Yes[28]	508	Avalanche Consensus	0.0486[29]	Solidity	Yes	2 (since centralized)
Tron	Yes[30]	1371	DPoS	Not available	Solidity	Yes	1 (Founder's reputational concerns)
IOTA 2.0	Yes[31]	60	Mana	Feeless	Wasm, Solidity	On TestNet	3

Table 2: Rating 7 Smart Contract Platforms against 7 Criteria

[a] Enhanced Security and Scalability [b] Robust Ecosystem and Ample Liquidity [c] Dependable Consensus Mechanism
[d] Sustainable Transaction Throughput and Cost-Efficiency [e] Comprehensive Development Toolkit and Smart Contract Languages
[f] Transparent Leadership and Future Roadmap (comparative scores w.r.t to Ethereum)

6 Implementation Details and Design Considerations

6.1 Designing the TechVerse Ecosystem

Methodology: Proposed Architecture for a User-Centric Content Ecosystem

In a user-centric content Ecosystem, we use a multi-layered design approach to ensure an inclusive environment where participants can contribute, consume, develop, and earn from content. This architecture includes three distinct but interconnected components, which together form a comprehensive content sharing and monetization solution, called TechVerse in our research.

The TechVerse architecture operates at two primary levels, each addressing specific aspects of the ecosystem:

Level 1: Conceptual Foundation

At this initial level, we lay the theoretical groundwork, providing a conceptual structure for research and development within the user-centric content ecosystem. This level includes:

1. **Scenario Architecture**: A comprehensive framework that defines the principles and scenarios of underlying content sharing and collaboration within the ecosystem. It clarifies the roles, influences, and interactions of users, content creators, and consumers, with a strong focus on fostering an environment of collaboration and engagement.

Level 2: Pragmatic Implementation

Moving beyond the conceptual level, Level 2 serves as the practical blueprint for software development and system implementation. It encompasses:

3. **Technology Architecture**: A pragmatic technology design that meets the needs of the user content sharing ecosystem. The framework is based on the integration of a trusted DLT organization, which serves as the backbone of secure and transparent transactions among the ecosystem's stakeholders. It describes the technical infrastructure, protocols, and data management strategies necessary for maintaining the expected environment.

4. **Framework for Implementation**: A detailed framework that bridges the gap between theory and practice. It provides actionable guidelines and methodologies for developing, deploying, and maintaining a consortium DLT-based open content sharing ecosys-

tem. This framework extends to the creation of user-friendly interfaces, content curation mechanisms, and compensation structures, enabling a seamless and rewarding user experience.

Let us now look into each levels in detail.

6.2 Scenario Architecture

In general, the proposed business schema for the content sharing ecosystem can be formally described as a 9-tuple: Is, U, Br, Co, Tn, Bc, Sy, R, An. Each element of this tuple represents a key element within the ecosystem:

U: It refers to Users, who play a central role in the ecosystem. Users can be content creators, consumers, or contributors, collectively shaping the content structure.

Co: Signifies Content, the core of the ecosystem. Content consists of text, images, videos, or any digital assets contributed, shared, and consumed within the ecosystem.

Is: It refers to Information System, which includes various content sharing platforms and applications that users interact with.

Br: Represents Behaviors performed by Users. These behaviors include actions such as uploading content, reading content, modifying content etc.

Tn: Stands for Transactions, which are initiated by Users and recorded on the DLT. Transactions capture various activities within the ecosystem, such as content uploads, modifications, and payment transactions.

Dlt: Corresponds to DLT, indicating the underlying DLT infrastructure that ensures transparency, security, and immutability of ecosystem activities.

Sy: Refers to Strategy, encapsulating the rules, parameters, and governance mechanisms that govern the ecosystem. Strategies define how content is rewarded, content visibility, privacy, and other vital aspects.

Assets: Represents Rewards, which Users receive for their contributions and actions within the ecosystem. Rewards can take the form of tokens, digital assets, or any other agreed-upon compensation.

An: It describes Applications, especially DLT-enabled applications that improve ecosystem performance. These processes include content analysis, content monetization, identity solutions.

With these elements defined, the business schema of a content sharing ecosystem can be understood as a set of business processes that govern how users interact with content, others, and DLT-based infrastructure. This framework forms the basis for designing a data sharing ecosystem that prioritizes data sharing, user participation, and appropriate compensation.

To provide further explanation, we will outline two common business processes within the proposed schema.

Content Sharing Business Process 1: Content Upload and Reward

In this process, a User (U) acting as a content creator uploads content (Co) to an Information System (Is). The behavior (Br) involves content creation and submission (Tn) to the DLT (Dlt) for validation and recording. The Strategy (Sy) employed defines how the User is rewarded (R) for their contribution. This process promotes user-generated content and incentivizes participation.

Content Sharing Business Process 2: Content Reading,Modification and Compensation

Here, a User (U) acts as a content consumer and reads content (Co) from an Information System (Is). The behavior (Br) entails content consumption and interaction (Tn) recorded on the DLT (Dlt). The Strategy (Sy) dictates how the User is compensated (Assets) for their engagement, ensuring that content consumers are also valued contributors.

User-Centric Content Business Process 3: Paid Content Access

In this process, a User (U) acts as a content consumer interested in reading premium or paid content (Co) from an Information System (Is). The behavior (Br) involves the User initiating access to the paid content and indicating their intention to read it. However, since this content is not available for free, a Transaction (Tn) is generated to record the User's intent to access the content on the DLT (Dlt). The Strategy (Sy) dictates that the User must make a payment (Pay) to access the content. Once the payment is successfully processed, the User gains access to the paid content.

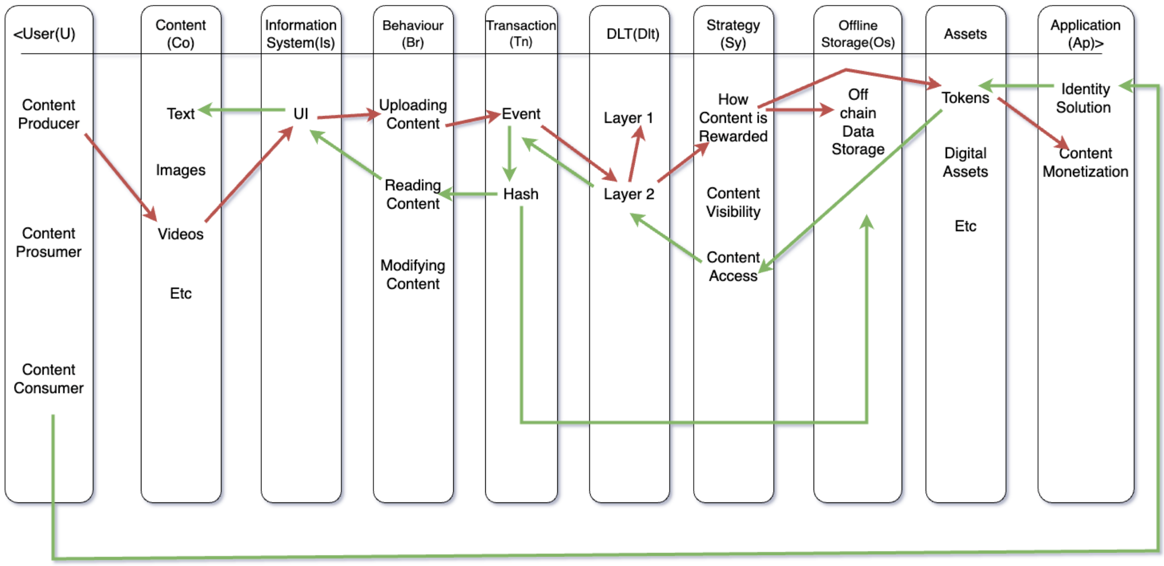

Figure 8: *Business Process Visualized*

6.3 Technology Architecture

Now, let's explore the process of constructing a decentralized application (DApp) structure. To illustrate, we'll examine the structure of a decentralized application built on the Ethereum platform. Afterward, we'll provide a concise step-by-step guide on how to create a DApp using IOTA 2.0.

Designing a decentralized ecosystem Components: DLT, smart contracts, user interfaces, storage solutions

A decentralized application (dApp) is a type of software application that operates on a distributed network, using blockchain technology for data processing and transaction processing. The dApp architecture consists of several main components:

1. Ethereum Blockchain: The Ethereum Blockchain acts as a network on which dApps run and store data. It securely creates and verifies smart contracts, which define the logic of an application.

2. Smart Contracts: Smart contracts are independent contracts whose code and data reside on the Ethereum blockchain. They describe state changes in the blockchain and are written in languages such as Solidity.

3. Ethereum Virtual Machine (EVM): EVM is a global virtual computer that handles

smart contract logic and status changes within the Ethereum network.

4. Frontend: The frontend is the user interface of the dApp, which enables users to interact with the application. It communicates with smart contracts to conduct transactions and generate information.

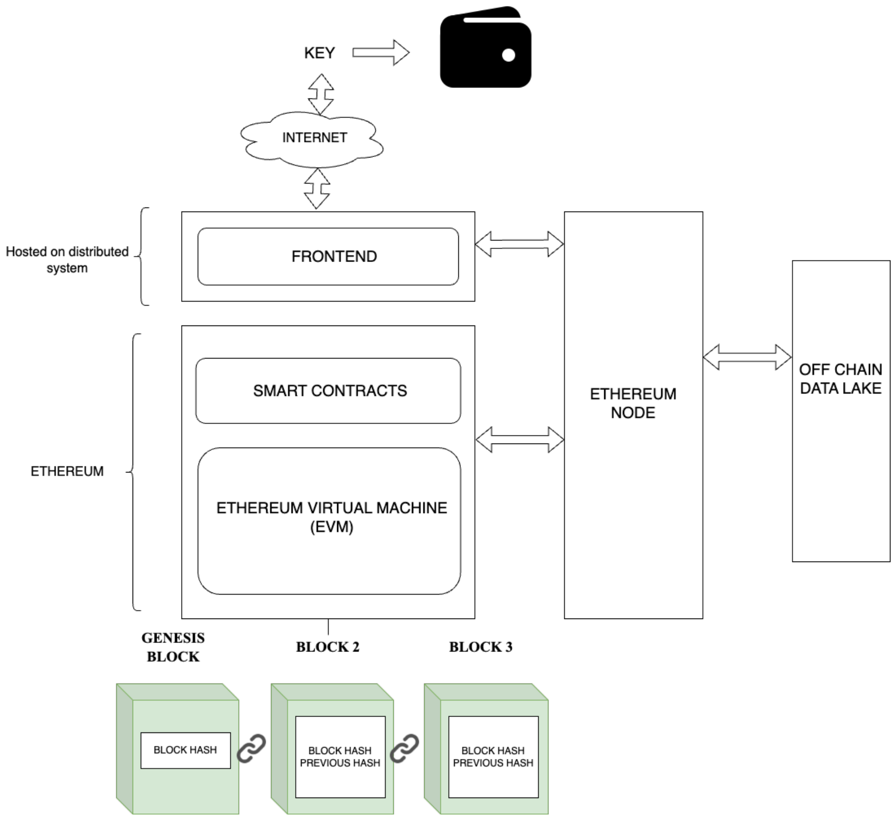

Figure 9: *Decentralized App Architecture*

To interact with the Ethereum blockchain, the front end communicates with nodes, which can be hosted on servers or accessed through third-party services. These nodes implement the JSON-RPC specification, which provides a uniform way of interacting with the blockchain. When a user needs to sign a transaction, Metamask, a browser plugin that acts as an Ethereum wallet, plays the role of the signer by securely managing the private keys. This framework ensures shared data processing and secure transaction processing in dApps.

However, Ethereum has some limitations, such as scalability and high gas fees.

Ethereum 2.0 will use sharding (parallel processing) as enhancement to improve on scalability factor. (Founder of Ethereum Vitalik Buterin, sharding is the solution for the scalability trilemma [32]) Making changes to an environment like Ethereum which holds billions of dollars and assets will naturally be a slow process.

There are many new networks out there, who are trying to overcome the shortcomings of Ethereum but most of them have given up decentralization and achieved more TPS. After learning how existing DApps are developed let us now learn how to integrate and develop a DApp using IOTA Tangle 3.4.

Integration of IOTA into the architecture The current production version of DApp can only be designed using IOTA Mainnet (Chrysalis protocol) [33] which does not support smart contracts. Also, there are some existing DApps which use Private Tangle [34].

A staging network i.e. Shimmer (Stardust protocol)[35] launched ShimmerEVM[4] which supports smart contracts and therefore we may explore further for our scenario if the Network fulfils all the requirements and later evaluate the same. With advancements in IOTA development the connection details would change, these recommendations are based on current development (Oct'2023).

Components: Shimmer(IOTA ecosystem's staging network), Layer 1(IOTA SDK, Cli Wallet, Identity framework), magic contracts, Layer 2 (Smart Contracts)

The three primary entities that are considered in the Layer 1 IOTA network are:

1. Clients
2. Nodes - run a node software which gives them read and write access to the IOTA network eg. Hornet
3. Tangle - network of Nodes

A visual representation of the components can be seen in Fig. 10

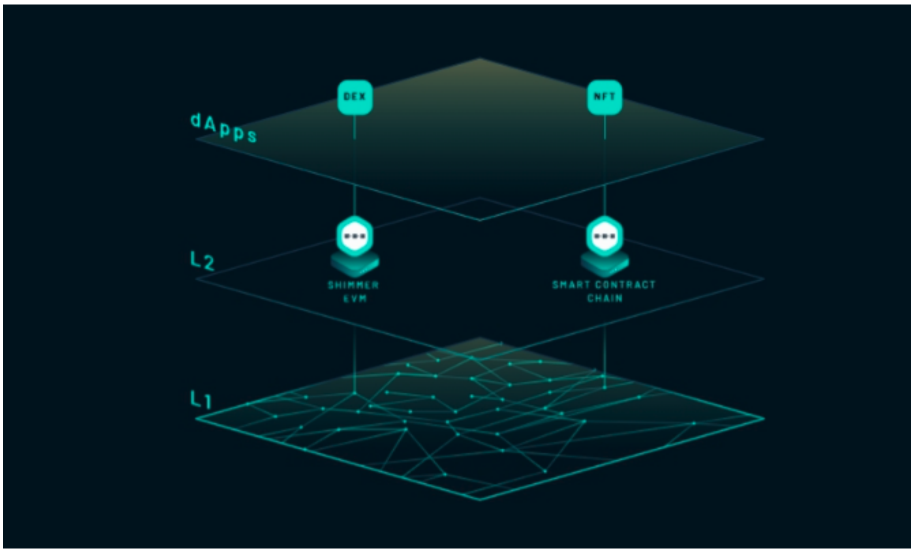

Figure 10: *Components included to develop a Tangle based ecosystem[36]*

The next section will introduce the key elements required to implement a Decentralized Application for our scenario.

6.4 Framework for Implementation

Prerequisite terms before we start, we have to make sure we have the following:

Network: You should have access to a network like ShimmerEVM, where you can run and transact with the smart contract. **Wallet**: You will need a wallet to connect to the network, verify your identity, and pay transaction fees. You can choose any wallet you like; the document uses Metamask as an example. **Smart Contract**: Prepare a smart contract that you intend to use on the chain to manage network state records and related changes. **Client Library**: This document uses Ether js[5], but there are many other options like web3js as well.

As mentioned earlier and in Fig.10, a Layer 2 solution of Shimmer is best suitable for our scenario, therefore we can start as per the wiki page of IOTA [37].

Building DApps with Layer 2 Smart Contract platform

- Connect to the Public Test network and get test tokens from faucet to start developing [38].

- EVM compatible **Metamask** wallet can be used to interact with web applications [38].

- Remix IDE can be used to deploy smart contracts on the public test network. Or Hardhat command line tool can also be used [38].

- Users are free to use any frontend framework eg. Reactjs to develop the Web UI.

The major requirement of our scenario is to initialize native tokens for management of the content. Shimmer offers creating native tokens wherein we can specify the max supply once in a smart contract [39].

More information on creating a sample voting DApp can be found in A.1.

7 Generalization and Applicability

7.1 Case Studies of Similar Requirements

In today's digital world, people and organizations are constantly sharing a lot of information with tech giants like Google or any other platforms. However, in this data sharing system, something important is often missing - a fair and transparent governance model. Currently, data contributors do not receive any incentives to share their valuable information, while technology companies aggregate, analyze and monetize this information for their own benefit. This lack of a formal form of governance model raises concerns related to data privacy, ownership, quality, and compensation. In response to this issue, this research examines the importance of a regulatory framework aimed at establishing data privacy laws, identifying individuals, assessing the fundamental value of data, and providing appropriate compensation to data contributors. I created a scenario of the implementation of such a governance model, which sheds light on the ability of DLT to resolve existing data sharing discrepancies.

However, the situation discussed here is only one example of how this form of governance can be realized. In this section, we examine some scenarios that share a common framework while dealing with different domains and contexts. These situations serve as examples of flexibility and potential areas of application.

1. **Case 1: Toll Collect in Germany - Data Analytics for Economic Trends**

   Consider the German Tax Collection system, which collects data on the number of trucks traveling on highways. This information can be used for extensive data analysis. Economists can use it to predict German economic conditions. Like TechVerse(name of the ecosystem in this research) users, the Toll Collect system can define terms and conditions for monetizing data, enabling data-driven decision-making. In this case, the application can act as a shared data market, where individuals or companies define metadata for available data and monetize it. This covers business-to-business (B2B) and business-to-consumer (B2C) transaction, ensuring an element of trust that is often missing in centralized platforms.

2. **Case 2: Comprehensive Agricultural Data Repository**

   In the field of agriculture, the company integrates multiple data sources, including IoT sensors, weather forecasts, satellite imagery, and soil moisture data, to create a comprehensive agricultural data warehouse. Like Professor Turner's (persona in this research) content on TechVerse(name of the ecosystem in this research), this repository contains valuable information. The company can define data access and monetization rules, enabling supply chain optimization, crop yield monitoring, and research in the agricultural sector. Users, in this case, can access this agricultural data by agreeing to certain terms and conditions, similar to how students and teachers interact in TechVerse(name of the ecosystem in this research).

3. **Case 3: Catena-X, infrastructure of data sharing in the automotive industry**

   Catena-X has a vision to develop a platform where automotive industry stakeholders share data securely and transparently, while receiving incentives for their contributions. The goal is to create a strong ecosystem for data exchange within the industry [40].

In short, the scenario-based design approach used by TechVerse, as demonstrated by the experiences of Professor Turner, Archie Andrews, and Liam, can be applied to a variety of areas outside of education. It is a flexible system that enables the safe, scalable, and efficient implementation of platforms for different purposes.

8 Summary and Conclusion

In conclusion throughout the project, we followed a scenario-based design approach to derive meaningful results. The evaluation process played a key role in our selection of IOTA 2.0 as the technology of choice for creating an ecosystem that addresses the challenges of trust, ownership, and monetization. It is important to note that if we use different evaluation criteria, our decision may change. Although it is possible to realize such a scenario using Ethereum, IOTA 2.0 has provided the flexibility, scalability, and performance needed to achieve our goals, especially in the context of open-source data sharing. A major breakthrough of IOTA 2.0 is its support for Layer 2 smart contracts, which brings many benefits. Also, one of the key benefits is tokenization, which enables users to represent and exchange assets as tokens on the IOTA network. This paves the way for a wide range of applications, from representing physical assets digitally to enabling the creation of new cryptocurrencies.

With this choice, we are now close to proceeding with the implementation phase. The technology aims to support the creation of an environment that reflects our scenario. This ecosystem is designed to provide innovative solutions for achieving trust, content ownership, and monetization that traditional centralized platforms often struggle to provide. In short, our research project not only shed light on the problems but also provided a practical solution in the form of IOTA 2.0. As we enter the implementation phase, we expect to present a tangible solution that will change the conditions of open-source distribution, making it more fair and efficient.

References

[1] John M. Carroll. *Scenario-Based Design*. 2003, pp. 45–70.

[2] Cryptomaniaks. *DLT for dummies*. 2023. URL: https://cryptomaniaks.com/distributed-ledger-technology-for-dummies-currentyear.

[3] David Hamilton. *What is Distributed Ledger Technology (DLT)*. 6.06.2021. URL: https://www.securities.io/what-is-distributed-ledger-technology-dlt/.

[4] Ashutosh Dhar Dwivedi Gautam Srivastava Shalini Dhar and Jorge Crichigno. "Blockchain Education". In: (2019).

[5] Gautam Srivastava et al. "Blockchain Education". In: *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. 2019, pp. 1–5. DOI: 10.1109/CCECE.2019.8861828.

[6] ICAEW. *History of blockchain*. 2023. URL: https://www.icaew.com/technical/technology/blockchain-and-cryptoassets/blockchain-articles/what-is-blockchain/history.

[7] Mythili Boopathi Seetha.R. "Block Chain Technology And Its Applications– A Review". In: *Webology (ISSN: 1735-188X)*. Vol. 19. 3. 2022, pp. 2347–2360.

[8] Bitcoin. *Get started with Bitcoin*. 2023. URL: https://bitcoin.org/en/.

[9] Morgan Stanley. *Cryptocurrency 201 What Is Ethereum*. 31.01.2022. URL: https://advisor.morganstanley.com/daron.edwards/documents/field/d/da/daron-edwards/Cryptocurrency_201__What_is_Ethereum_.pdf.

[10] Gennaro Cuofano. *Ethereum: The History Of Ethereum*. June 30, 2023. URL: https://fourweekmba.com/history-of-ethereum//.

[11] Ethereum. *Welcome to Ethereum*. 5.10.2023. URL: https://ethereum.org/en/.

[12] Bibek Adhikari. "The Potential role of IOTA in DLT Technology". In: (2021).

[13] Sophie Chabridon Nathanael Denis and Maryline Laurent. *Bringing Privacy, Security and Performance to the Internet of Things Through Usage Control and Blockchains*. 644th ed. springer, 2022, p. 15. URL: https://link.springer.com/chapter/10.1007/978-3-030-99100-5_6#Fig1.

[14] IOTA. *IOTA 2.0 Solid Foundation*. Accessed on 16.10.2023. URL: https://blog.iota.org/solid-foundation-iota20/.

[15] Sebastian Muller et al. "Tangle 2.0 Leaderless Nakamoto Consensus on the Heaviest DAG". In: *IEEE Access* 10 (2022), pp. 105807–105842. DOI: 10.1109/access.2022.3211422. URL: https://doi.org/10.1109%2Faccess.2022.3211422.

[16] Sebastian Müller et al. *Reality-based UTXO Ledger*. 2023. arXiv: 2205.01345 [cs.DC].

[17] Andrew Cullen et al. "Access Control for Distributed Ledgers in the Internet of Things: A Networking Approach". In: *IEEE Internet of Things Journal* 9.3 (2022), pp. 2277–2292. DOI: 10.1109/JIOT.2021.3096129.

[18] Cardano. *Learn about Plutus*. Accessed on 16.10.2023. URL: https://docs.cardano.org/plutus/learn-about-plutus/.

[19] Raaj Mishra et al. "Privacy Protected Blockchain Based Architecture and Implementation for Sharing of Students' Credentials". In: *Information Processing and Management* 58 (Jan. 2021). DOI: 10.1016/j.ipm.2021.102512.

[20] Applicature. *History of Ethereum Security Vulnerabilities, Hacks, and Their Fixes*. Accessed on 12.10.2023. Sep 27, 2017. URL: https://applicature.com/blog/blockchain-technology/history-of-ethereum-security-vulnerabilities-hacks-and-their-fixes.

[21] YCharts. *Ethereum Average Transaction Fee (ETH)*. Accessed on 12.10.2023. 13.10.2023. URL: https://ycharts.com/indicators/ethereum_average_transaction_fee_eth.

[22] RAHUL NAMBIAMPURATH. *How Binance Got Hacked*. Accessed on 12.10.2023. October 07, 2022. URL: https://www.investopedia.com/binance-got-hacked-6748215.

[23] YCharts. *Binance Smart Chain Average Transaction Fee (BSCH)*. Accessed on 12.10.2023. 12.10.2023. URL: https://ycharts.com/indicators/binance_smart_chain_average_transaction_fee_es.

[24] Francisco Memoria. *Cardano Average ADA Transaction Fees Dropped After Vasil Along with On-Chain Activity*. Accessed on 12.10.2023. 7 Oct 2022. URL: https://www.cryptoglobe.com/latest/2022/10/cardano-average-ada-transaction-fees-dropped-after-vasil-along-with-on-chain-activity/.

[25] Hassan Maishera. *Polygon (MATIC) Reveals It Was Hacked Earlier*. Accessed on 13.10.2023. December 30, 2021. URL: https://finance.yahoo.com/news/polygon-matic-reveals-hacked-earlier-103532665.html.

[26] Tanner Philp. *Cardano Average ADA Transaction Fees Dropped After Vasil Along with On-Chain Activity.* Accessed on 12.10.2023. Jan 9, 2023. URL: https://www.coindesk.com/consensus-magazine/2023/01/06/solana-vs-polygon-a-developers-perspective/#:~:text=It%20is%20currently%20spec'd,whereas%20Solana%20is%20~%240.0002..

[27] Aleksandra Yudina. *2nd Solana Hacks Explained.* Accessed on 13.10.2023. FEBRUARY 1, 2023. URL: https://ackeeblockchain.com/blog/2022-solana-hacks-explained/.

[28] QuillAudits - Web3 Security. *2nd Avalanche Hack in a Row.* Accessed on 13.10.2023. Sep 25, 2021. URL: https://medium.com/quillhash/2nd-avalanche-hack-in-a-row-%EF%B8%8F-312d0dcfa14b.

[29] Snow Trace. *Trace.* Accessed on 12.10.2023. Jun-03-2023. URL: https://snowtrace.io/tx/0xaed8e5677da8bf35a3e6bddc5ef7f19795db548f527a03611ab0eaca4f3096c7on-a-developers-perspective/#:~:text=It%20is%20currently%20spec'd,whereas%20Solana%20is%20~%240.0002..

[30] Tomiwabold Olajide. *Crypto Hack: TRON's DeFi Ecosystem Reportedly Hit with 4,600 BNB Exploit.* Accessed on 13.10.2023. 10/11/2022. URL: https://u.today/crypto-hack-trons-just-ecosystem-reportedly-hit-with-4600-bnb-exploit.

[31] Paddy Baker. *IOTA Founder Personally Refunding Hack Losses to Safeguard Project's Remaining Reserves.* Accessed on 13.10.2023. Mar 12, 2020. URL: https://www.coindesk.com/business/2020/03/12/iota-founder-personally-refunding-hack-losses-to-safeguard-projects-remaining-reserves/.

[32] coinmarketcap. *Blockchain Trilemma.* Accessed on 03.11.2023. 3.11.2023. URL: https://coinmarketcap.com/academy/glossary/blockchain-trilemma.

[33] IOTA Chrysalis. *Path to Chrysalis.* Accessed on 16.10.2023. URL: https://wiki.iota.org/introduction/explanations/update/path_to_chrysalis.

[34] IOTA. *Private Tangle.* Accessed on 16.10.2023. URL: https://wiki.iota.org/introduction/how_tos/one_click_private_tangle/.

[35] IOTA Shimmer. *Shimmer.* Accessed on 16.10.2023. URL: https://wiki.iota.org/get-started/introduction/shimmer/introduction/.

[36] stardust. *Deep Dive into Stardust.* Accessed on 16.10.2023. URL: https://blog.shimmer.network/a-deep-dive-into-stardust/amp/.

[37] IOTA. *EVM/Solidity Based Smart Contracts*. Accessed on 16.10.2023. URL: https://wiki.iota.org/wasp-evm/introduction/.

[38] IOTA. *Public Testnet Quickstart Guide*. Accessed on 16.10.2023. URL: https://wiki.iota.org/wasp-evm/getting-started/quickstart/.

[39] IOTA. *ERC20 Example*. Accessed on 16.10.2023. URL: https://wiki.iota.org/wasp-evm/how-tos/ERC20/.

[40] BMW. *CATENA-X: INNOVATION THROUGH COOPERATION.* Accessed on 16.10.2023. 31.01.2023. URL: https://www.bmwgroup.com/en/news/general/2023/catenax.html.

[41] IOTA. *Create a Simple Voting dApp on ShimmerEVM*. Accessed on 16.10.2023. URL: https://wiki.iota.org/tutorials/shimmerevm-dapp-voting.

[42] Wikipedia. *Non Fungible Token*. Accessed on 16.10.2023. URL: https://en.wikipedia.org/wiki/Non-fungible_token.

A Appendix

A.1 Sample Voting DApp

This is a simple outline of creating a DApp on ShimmerEVM based on an online tutorial. [41] This DApp will allow users to cast votes, view results, and store all data securely on the DLT.

Let us start with creating a basic solidity smart contract.

1. The smart contract, called 'Voting.sol,' starts by defining an struct called 'Voter' to represent users.

2. It maps this 'Voter' struct to voter addresses on the DLT.

3. Another struct, 'Proposal' is created to nominate candidates for voting.

4. Adding Voting Functions: Implement a 'vote' function that registers the user's votes for a proposal. This process checks if the voter is eligible and has not voted before. To avoid double voting, this function marks the user as the voter and adds its weight to the selected proposal.

5. Calculating the Winning Proposal: Added 'winningProposal' function to calculate the winning proposal by considering the number of votes. Returns the index of the winning proposal.

6. Retrieving Winner Name: The 'winnerName' function is created to return the name of the winning entry for the human-readable result.

Deployment and User Interface: Once your smart contract is ready, you can deploy it using Remix, Hardhat etc. The UI interacts with the deployed contract by calling its functions and methods using etherjs,web3js.

This functionality can be easily extended to adapt to the TechVerse scenario and develop a fast and efficient ecosystem.

Notes

[1]A non-fungible token(NFT) is a unique digital identifier which is used to certify ownership and authenticity and that is recorded on a blockchain. It cannot be copied, substituted, or subdivided.[42]

[2]Native tokens are created natively on the DLT and only exist in the DLT environment. Payment tokens are a common analogue of native tokens. Another example of native tokens can be found in tokens produced through Initial Coin Offerings (ICOs). These tokens are generated directly in DLT and are not backed by off-chain assets or securities.

[3]The blockchain trilemma, a term introduced by Ethereum co-founder Vitalik Buterin, refers to a trade-off between the three key aspects of blockchain technology: security, scalability and decentralization.

[4]ShimmerEVM is a high-speed smart contract chain on top of Shimmer, which acts as a universal bridge for dynamic and feeless transfers between any type of smart contract chains.

[5]JavaScript library which enables frontend applications to interact with the smart contracts.

Platform	**Pros**	**Cons**
Ethereum	Established standards (e.g., ERC-20), Huge liquidity and market capitalization, Vibrant developer community	High gas fees, Slow transaction speed, Privacy concerns, Vulnerable to hacks
BNB Chain	Accelerator programs for projects, EVM compatibility, Privacy initiatives	Centralization, Limited dApps and DeFi projects, Moderate throughput
Cardano	Smooth fork transitions, Energy-friendly (PoS), Faster throughput	Upgrade delays, Limited popular dApps
Polygon	Security and scalability, Cost-effectiveness, Growing NFT and DeFi projects	Ethereum dependence, History of hacks, Moderate speed
Solana	Processing over 3,300 transactions per second, Cost-effective transactions, Energy-efficient (proof-of-history), Mainstream products like Saga project	Frequent network outages, Centralization of top validators
Polkadot	Interoperability among blockchains, Easier upgradeability, Supports custom blockchains using Substrate	Exclusion of Ethereum from bridging, Slow progress in development
Tron	low transaction fees, Fast transaction speed (2,000 TPS)	USDD depegs and instability, Few DeFi and NFT projects, Founder's reputational concerns
Avalanche	dApp suitability,bEVM compatibility, Security with active bug bounty programs	High validation requirements, No punishment for malicious validators

Table A.1: Comparison of Smart Contract Platforms