Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Dipartimento di Informatica – Scienza e Ingegneria

Corso di Laurea Magistrale in Informatica per il Management

Design and Implementation of a Multi-level Architecture for Separation of Power in Legislative Process

Relatore:
Prof. Stefano Ferretti

Presentata da:
ARIANNA ARRUZZOLI

Correlatore:
Dr. Mirko Zichichi
Prof.ssa Monica Palmirani

I
a.a. 2024 - 2025

Abstract

The digitalization of governmental infrastructure presents a significant challenge in respecting the foundational principles of democracy, particularly within legislative contexts. The Italian legislative process serves as an exemplary case study for exploring how diverse technologies can be employed to uphold democratic values while mitigating procedural inefficiencies. This thesis proposes a multi-level architectural model designed to enhance the separation of powers within the legislative process. The proposed system is initially implemented using the IOTA blockchain, with governance mechanisms enforced through smart contracts written in the Move programming language. Legislative documents are structured and tracked using the Akoma Ntoso XML OASIS standard. Furthermore, access control is managed through Decentralised Self-Sovereign Identity (SSI), implemented via the IOTA Identity framework. A statistical performance analysis of the system has been conducted to assess its scalability and cost efficiency, with results indicating promising potential for real-world application.

Summary

Indice

Chapter 1

Introduction

The current formulation of the diverse forms of state in the Western world represents a natural historical outcome of the population's evolving needs. Contemporary democracies are the result of popular and political movements that enabled a transition from a monarchical form of state to a democratic one within which the State itself is subject to the laws that govern it.
Modern states are defined as "stable, centralised apparatuses that hold the monopoly on legitimate force within a given territory". In addition, they possess an organisational structure served by a professional bureaucracy.

The ideology that led to the definition and implementation of democracies has produced a fundamental concept embedded within these apparatuses: the rule of law [4].
The rule of law constitutes a legal concept founded upon several fundamental pillars, which enable the State to pursue its objectives in accordance with the forms and limits prescribed by the law through the enactment and application of legal norms [27].

Contemporary constitutional democracies are grounded on three founding principles:

1. the **institutional separation of powers**, in order to prevent the concentrarion of authority, whether is legislative, judiciary or government authority;

2. the **rapresentational inclusion** of diverse societal voices, ensuring an active role for minorities and oppositions;

3. the promotion of **civic engagement** alongside transparency in governmental functions.

The principle of separation of powers was developed with the aim of limiting political authority in order to safeguard individual liberty. Its initial theorisation is primarily attributed to Montesquieu, who, in his 1748 work *The Spirit of the Laws*, asserted that if the purpose of the State is to ensure political liberty, it is essential that political powers be three in number and that they remain distinct from one another [4].
The three defined powers are: the legislative power, the executive power, and the judicial power. Each power is identified by the function it performs; moreover, it is essential that each function be assigned to distinct authorities, as the concentration of multiple functions in the hands of a single entity would pave the way for arbitrariness.
The powers, although distinct and separate, should be able to exert reciprocal influence in such a way that each may act as a check on the excesses of the others.

In many countries in the European dimension, the legislative process is governed by different institutions with different roles [34], ex gratia Lords and Commons [6], Senate and Chamber Deputies [28] or EU Parliament and Council of EU [11].
Various actors enter the legislative process to contribute or monitor. Each of these actors should follow precise procedures and the rule of law while applying the separation of power in order to activate a mutual checking on the others.

Given the rapid technology evolution in the last century, the legislative processes are switching to a digital form. Government agents and their apparatus are under increasing pressure to quickly adapt to changing circumstances given a variety of technological, economic and societal reasons [43].

The design and implementation of legislative information systems require careful consideration to prevent any single branch from gaining excessive control over joint resources.
Since legislative documents form the foundation of the legal framework of any rule of law, their integrity, security and confidentiality are crucial to maintain transparency and effective governance, other than supporting participation of the citizen in the legislative process [34].

Current digital legislative systems typically operate within single institutions, failing to address the constitutional requirement for maintaining institutional autonomy, while enabling inter-institutional coordination [33]. Given the unavailability of adapted documents from different platforms and the absence of a unified document repository, significant inefficiencies are generated in legislative workflows.

This thesis designs and explores within the Italian constitutional framework a solution to ensure the respect of the founding principles of the rule of law, via blockchain technology. This technology is a decentralised digital ledger that securely stores records across a network of computers in a way that is transparent, immutable, and resistant to tampering [16]. Its characteristics can be adapted to legal documents in order to assure their accountability and immutability.

The architecture proposed consists of a multi-level blockchain system designed to protect the separation of powers in government whilst digitalising law-making processes.

The work will be depicted as follows: Chapter 2 will describe the state of the art and the technologies used in the development; then Chapter 3 will describe the Italian use case for legislative process management, and within Chapter 4 the design and its implementation will be described. Lastly, Chapter 5 will present the results of this implementation and its statistics in order to identify the strengths and weaknesses of the proposed solution.

Chapter 2

State of the art

This chapter provides an overview of the state of the art regarding the technologies employed in the development of the work presented in Chapter 4.
Section 2.1 presents the current solution adopted within the context of the italian legislative process previously introduced. The ensuing sections provide an detailed exploration of the fundamental concepts and technologies that constitute the foundation of the proposed solution outlined in Chapter 4.

2.1 xLeges

In Italy, the National Centre for Information Technology in Public Administration (CNIPA) was established by the Office of the Prime Minister pursuant to Article 176, paragraph 3, of Legislative Decree 196/2003 [42]. Its mandate is to foster the achievement of e-Government objectives and to guide the progressive development of the information society. These objectives contribute to the implementation of the principles and tools of information and communication technologies within public administrative functions and procedures, with the objective of upholding the fundamental principles of democracy.

In this context, the CNIPA has launched an initiative entitled *e-Leges*,

aimed at supporting technological advancement within public offices through the establishment of standards and the design of dedicated infrastructures. Among the various projects, xLeges is the one managing the exchange of electronic documents within the institutions involved in the legislative process.

xLeges was conceived and developed through a collaborative effort involving multiple institutions with the aim of supporting the collaborative processes underlying the approval, draughting, and publication of laws in the Official Journal [45].

The primary functionalities through which xLeges supports the legislative process include the exchange of documents, the management of any supplementary information exchanged, and the possibility to receive notifications upon the occurrence of relevant events.

The architecture of xLeges incorporates pioneering features for the time, including a pure peer-to-peer design. It employs Web Services technologies alongside the pervasive adoption of the XML format for information exchange. It constitutes the first instance in which the paradigms of peer-to-peer computing have been implemented in conjunction with Web Services technologies [45].

The primary objective of xLeges is to facilitate cooperation amongst organisations. This is accomplished through the adoption of standards that enable interoperability between heterogeneous sources. For this reason, the files conform to the XML format, as required for compliance with the Normein-Rete standards [14].
The files are organised in a hierarchical structure that gathers all documents necessary for the composition of the bill draft. These files are grouped within folders and exchanged via the xLeges system, accompanied by metadata that specifies the legislative phase, the structure of the normative act, and other relevant information. Furthermore, during information exchange, the transmitted data incorporates a workflow identifier comprising transmission-related elements, including timestamps and identification codes for both

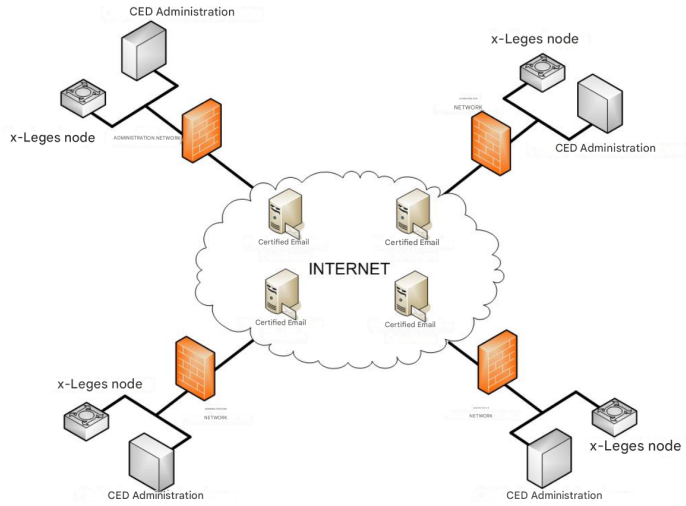

Figure 2.1: XLeges system architecture

sender and recipient.

Each institution functions as an autonomous node, operating as both a client and a server. Communication among these nodes is executed through Web Service technologies. Upon the completion of any step in the legislative process, a notification is sent to all peer nodes via Certified Electronic Mail (E-Mail). These event notifications pertain to specific actions, such as the transmission of a folder, a request for information, or a request to access a folder [46]. Each notification includes a query for the database of the originating node. To access the associated files, the recipient node must execute the query, thereby enabling an on-demand mechanism for information retrieval.

To ensure a high level of security, the xLeges system is designed to meet stringent requirements for integrity, confidentiality, and availability. These objectives are achieved through the use of Certified E-Mail messages exchanged via Web Service technologies, which are encrypted to guarantee data confidentiality. Security is reinforced through the participation of a trusted third party responsible for managing the authentication process. As a result,

robust security measures are implemented during both authentication and access control phases [45].

The xLeges solution enables the tracing of legislative folders iter and the recording of the chronological sequence of legislative steps. Its distinctive decentralised architecture has proven effective; however, it still presents certain risks related to information consistency. The system is decentralized, and so are the information concerning the legislative history, as they resides within individual databases and are retrieved only upon request.
Therefore, the solution proposed in this work seeks to implement an alternative architecture based on blockchain technology. The objective is to establish a decentralized system that also features a distributed yet uniform repository for storing the legislative process. This approach improves consistency and accountability while simultaneously facilitating the implementation of mechanisms for public accessibility, thereby ensuring that legislative data is more transparently available to citizens.

2.2 Smart Contract

Pursuant to Clause 1-bis of Article 20 of Legislative Decree 7 March 2005, n.82, known as the Digital Administration Code (CAD), a digital contract is deemed valid if it is executed using technical methods that ensure the security, integrity and immutability of the document [31]. Furthermore, it requires that the attribution of the document to its author be ensured in a clear and unequivocal manner. These characteristics are inherently present in the code contained within blockchains and are therefore referred to as smart contracts.

In particular, the term smart contract refers to a self-executing agreement written in code, stored and executed on a blockchain. It regulates the events inside transactions, controlling access to data, granting and revoking access. Upon generating a transaction, the smart contract automatically records the

corresponding operations on the blockchain, thereby establishing a transparent and immutable record that ensures accountability.

By definition, smart contracts ensure both transparency and flexibility. The terms and conditions encoded within them are accessible to all participating parties; each participant executes the same code, and every node in the network can independently verify the logic embedded in the contract, thereby guaranteeing transparency. Their implementation may be carried out using various programming languages, as long as they are Turing complete, which allows for a high degree of flexibility with diverse technologies. The primary programming languages used for developing smart contracts include Solidity, Rust, and Move. The language selection is typically determined by the specific blockchain platform with which the contract is intended to interact.

Smart contracts govern the internal logic of the blockchain and are employed to create digital assets and manage their lifecycle. A digital representation of an asset recorded on the blockchain is referred to as *token*.
Tokens are defined and regulated via smart contracts, which must conform to specific standard implementations in order to accurately represent particular types of assets. The principal standards originate from the Ethereum platform and are defined as Ethereum Requests for Comments (ERC), which outlines a set of guidelines for the development of smart contracts.
Among the most widely adopted standards are ERC-20 and ERC-721. The ERC-20 standard defines the technical specifications for fungible tokens, tokens that are mutually interchangeable and uniform in value. Meanwhile, the ERC-721 standard establishes the framework for non-fungible tokens (NFTs), which represent unique and non-interchangeable digital assets.

Smart contracts regulate tokens throughout their entire lifecycle, functioning as intermediaries between users and digital assets. They manage transactions, transfers, token creation, and deletion, recording every action within an immutable historical ledger.

Within the scope of this thesis, a series of smart contracts has been designed and implemented to enable the on-chain management of documents. These smart contracts formally define the rules, constraints, and operational requirements necessary to ensure the accurate governance and representation of legislative documents as digital assets.

2.3 IOTA

IOTA is a permissionless public blockchain ledger infrastructure that offers a different solution for the implementation of Distributed Ledger Technologies (DLTs). Although it shares certain characteristics with conventional blockchain systems, it diverges significantly in terms of architectural design, programming paradigm, and consensus mechanism.

Unlike traditional blockchains, which are typically structured as sequential chains of blocks, IOTA employs a fundamentally different data structure known as the Tangle. The Tangle is based on a Directed Acyclic Graph (DAG), wherein nodes are connected in a one-way and non-cyclic manner. This DAG architecture aims to enhance scalability by enabling parallel transaction validation, thereby avoiding the bottlenecks associated with linear blockchains [17]. Furthermore, this design allows multiple validators to submit transactions concurrently, optimising network throughput and reinforcing resistance to censorship.

The IOTA ledger adopts an object-based ledger, distinguishing between shared objects and owned objects. For transactions involving exclusively owned objects, IOTA enables immediate execution without requiring global consensus by leveraging a mechanism referred to as "*fast path consensus*" [24]. This approach minimises the potential of conflicts in common transaction types by allowing instant processing. Transactions involving shared objects, on the other hand, utilise a Delegated Proof-of-Stake (dPoS) consensus mechanism to ensure consistency and coordination across the distributed network.

The consensus protocol implemented in IOTA is based on the Mysticeti Protocol, a Byzantine Fault Tolerant (BFT)[1] consensus protocol that enables agreement among nodes in a distributed network, despite the presence of malicious or faulty actors [18]. The protocol maintains system reliability under the assumption that a majority of nodes behave honestly. Mysticeti is specifically optimised for low-latency and high-throughput performance, leveraging an uncertified Directed Acyclic Graph (DAG) structure. This design supports efficient and parallel transaction processing while preserving strong security guarantees [18].

The design principles of the IOTA protocol emphasise transparency, security, and ethical practices within digital environments, with the overarching goal of delivering high performance and fostering user empowerment.
In alignment with these principles, the IOTA ecosystem supports programmability across both protocol layers [17]. Within the blockchain context, Layer 1 refers to the data layer, which serves as the foundational ledger responsible for handling smart contracts and managing core data logic. In parallel, Layer 2 denotes the networking layer, which facilitates inter-node communication and enables seamless integration with off-chain applications.

Within the IOTA ecosystem, Layer 1 corresponds to the IOTA Mainnet, which is governed through smart contracts written in the Move language, an object-oriented, security-focused programming language adapted by IOTA to enable seamless interaction with its blockchain infrastructure. The IOTA Mainnet employs a UTXO-based ledger model, and while its functionality is intentionally constrained, it is primarily utilised for the creation of fungible and non-fungible tokens and data storage.

The Layer 1 architecture of the IOTA Mainnet incorporates a range of design features that contribute to its efficiency, security, and developer usability. One of its core strengths lies in scalable parallel processing, made possible by the use of independent objects that allow for concurrent execu-

[1] A Byzantine fault refers to inconsistent or conflicting observations of the same system state by different participants in a distributed network.

tion of transactions. This design enables high throughput while maintaining low transaction fees. The security of IOTA's Layer 1 is underpinned by the adoption of the Move programming language, which enforces a strict type system and resource-oriented design to mitigate common vulnerabilities. The unified token model simplifies token management, while compile-time checks enhance reliability by reducing runtime errors.

Layer 2 of the IOTA ecosystem is based on the Ethereum Virtual Machine (EVM) and is thus referred to as IOTA EVM. This layer periodically commits its state to the IOTA Mainnet, establishing interoperability between the two layers. Unlike Layer 1, which uses the UTXO model, IOTA EVM adopts an account-based ledger and serves as the execution environment for smart contracts written in the Solidity programming language, thereby enabling compatibility with Ethereum-based decentralized applications (dApps). The consensus mechanism employed in this layer is Proof of Authority (PoA), and it supports seamless interoperability of IOTA tokens with the Mainnet. The account-based model utilised in Layer 2 involves sequential execution, which may contribute to network congestion, higher transaction fees, and longer finality times. While the Ethereum Virtual Machine provides considerable flexibility and a well-established development environment, this architecture requires careful security audits to address potential vulnerabilities. Token management within this layer follows established ERC standards, which can introduce complexity in discovery and interoperability. Although the ecosystem benefits from extensive tooling, some features may influence the stability of the application.

The IOTA environment was utilised for the development and testing of the work presented in this thesis due to its seamless integration with the Move programming language and its support for rapid transaction testing.

2.4 IOTA Identities

An ever-growing volume of personal information is accessible online. In particular, ongoing interactions with digital networks generate persistent digital footprints, such as cookies and user preferences, which are subsequently analysed to derive inferences regarding individual characteristics. These insights are strategically employed to deliver targeted advertisements and to dynamically adjust pricing models in accordance with the estimated economic capacity of the consumer.

Global institutions are increasingly advancing initiatives aimed at restoring individuals' control over their personal data through the adoption of the Self-Sovereign Identity (SSI) paradigm. In the European context, this concept aligns with the principles established by the General Data Protection Regulation (GDPR) and serves as a valuable complement to the regulations introduced by Regulation 910/2014, referred to as eIDAS, which govern secure access to online services and establish the necessary requirements for proper and trustworthy digital transactions [13].

IOTA presents itself as a suitable solution for a universal identity implementation due to its scalable Distributed Ledger Technology (DLT) and proposed its own implementation defined IOTA Identities.
The IOTA Identity framework implements widely accepted standards and patterns for Decentralised Identity, providing a solution that bridges the gap between online identities and real-world personas [21].

IOTA Identities consist of Decentralised Identifiers (DIDs), which adhere to the W3C specification, and Verifiable Credentials. DIDs enable any entity to possess a unique identifier for which they can prove ownership. Through the development of the DID document, IOTA focuses on implementing mechanisms that ensure identity holders maintain control over the selective disclosure of their information. Verifiable Credentials, on the other hand, are digitally signed assertions issued by trusted third parties regarding a specific identity; they serve to verify identity claims and ensure that the published data is controlled and trustworthy. Combined, these components provide an

efficient and robust implementation for verifiability.

Furthermore, IOTA supports the creation of identities not only for individuals but also for organisations and objects. This capability is derived from its intrinsic design; in point of fact, IOTA Identities are implemented as Move shared objects. This approach facilitates interaction within smart contracts and enables full integration within the existing IOTA ecosystem [19].

Due to its nature as an object, it is possible to implement hierarchical structures that define who holds authority over the identity. This mechanism is realised through a controller capability, a tokenization system that grants the power to control an identity to the entity possessing the corresponding controller token. This intuitive approach enables the establishment of hierarchies within IOTA Identities, thereby facilitating management and adaptability across diverse scenarios.

Within the scope of the proposed work, IOTA identities are implemented to confer coherence with the overall system built using IOTA technologies. They introduce a simple integration of trusted identities within the system, thereby enabling the implementation of secure authorisation policies for the development of a correct model for access control and the separation of powers.

2.5 Move language

The Move language was first presented as a programming language in the Diem Blockchain[2], designed to encode ownership of digital assets and to define procedures for their management [10].

Move is a secure and verifiable programming language that maintains flexibility [30] using an object-oriented approach. It has been adapted and integrated into the IOTA protocol due to its robust security features and per-

[2]Formerly known as Libra, was a permissioned blockchain-based stablecoin payment system developed by Facebook [2].

formance within scalable environments. Move's properties enhance throughput, reduce network congestion, and lower transaction costs. As a result, Move provides a notable scalability advantage within the IOTA context over the Ethereum Virtual Machine's global shared state architecture, which is subject to scaling limitations [26].

The design of Move facilitates a natural and structured approach to managing state and behaviour associated with assets, users, and contracts, enabling intuitive modelling of complex data structures and interactions. The language emphasises the notion of digital asset ownership and integrates concepts analogous to physical properties, such as controlled scarcity and access control.

These features are implemented through first-class resources, which are custom-defined types that enforce the controlled scarcity principle by ensuring that a resource cannot be copied or implicitly discarded, and it may only be moved between program storage locations. Developers can safeguard access to critical operations using modules, which define resource types and procedures that govern the behaviour and rules for its declared resources.
Modules are key to flexibility, allowing code composition within the language. Although modules in Move do not possess a notion of self, the language employs static dispatch, enabling compile-time awareness of method calls, facilitating verification and increasing security [5].

Move prioritises safety and efficiency. Its approach to memory management and resource control is inspired by Rust's ownership model.
The compiler enforces strict rules to prevent common programming errors, ensuring assets remain secure in user accounts and cannot be accessed without correct keys. The Move compiler catches many common errors like type mismatches and resource misuse before deployment, enhancing safety and correctness without relying solely on runtime checks. There are no re-entry issues, and smart contracts cannot touch assets owned by an account without knowing the account keys [5], providing confidentiality.

The Move programming language has been thoroughly studied and em-

ployed for the implementation of smart contracts at the foundation of the solution proposed in this thesis. Its inherent constraints, particularly those preventing resource discarding and enforcing strict control over scarcity, have proven highly beneficial during the development process. Moreover, the Move language's owned object model enabled the formulation of precise and efficient access control policies, enabling governance over digital assets.

2.6 Zero Trust Paradigm

With the rapid diffusion of the Internet of Things (IoT), cloud computing, and increasingly mobile workforces, the need for dynamic and resilient security solutions has become more pressing than ever. The solution to these evolving challenges has been the definition of a paradigm shift in security strategy that is the concept of *Zero Trust* [15].

Embracing the foundational maxim "never trust, always verify", the Zero Trust paradigm challenges traditional security implementations by requiring ongoing authentication, authorisation, and validation of all entities seeking access to critical systems and data [1]. The model emphasizes the necessity of continuous verification and entails the formulation of differentiated security policies tailored to the interactions of various roles. This represents a contemporary and adaptive approach to securing digital infrastructures in an environment marked by rapid evolution and pervasive security threats.

At the core of the Zero Trust paradigm lies the principle of *least privilege* [1], which stipulates that users must be granted only the minimal level of access required to carry out their designated roles and responsibilities. In alignment with this approach, the paradigm necessitates rigorous network segmentation, involving the creation of isolated segments, commonly referred to as micro-perimeters, within which users are permitted to operate. This fine-grained architectural division plays a crucial role in containing potential security breaches and significantly limiting the capacity of malicious actors to move laterally across the network.

The Zero Trust paradigm has been inserted within the smart contracts logic employed in the development of the solution proposed. The costraints introduced guarantee assertions regarding the respect and implementation of the Authentication, Authorisation, Accounting (AAA)[3] paradigm.

2.7 Akoma Ntoso Standard

The Akoma Ntoso naming convention define a precise way to identificate concepts and resources inside collections. Its aim is to present *meaningfulness*, of the description of the resource, *permanence*, providing stability, and *invariance* of the identifier [35].

The Akoma Ntoso standard defines a series of referenceable concepts that are employed throughout the whole lifecycle of legal documents. Specifically, the Akoma Ntoso standard defines four main concepts:

- *Work*: the abstract concept of the legal resource.
- *Expression*: different versions of the same Work, or the same version of the same Work expressed in different languages.
- *Manifestation*: any electronic or physical format of the Expression. It is characterized by the specific process that generates an electronic document in a specific format.
- *Item*: the physical copy of any Manifestation, even if digital.

Furthermore, the framework establishes conceptual definitions of *Components* and *Portions* for referencing destinations within documents, thereby supporting specific legal citation requirements.
The standard accommodates the existence of unpublished documents or circumstances wherein information are unavailable in advance. Therefore, it

[3]The Authentication, Authorisation, Accounting (AAA) is a security model to implement access control [36].

defines *Virtual Expressions*, a simple modification to the Expression structure to allow document identification.

The Akoma Ntoso standard is used to implement International Resource Identifiers (IRI) used as references to addressable resources. The IRI reference of the Work is the baseline for building the IRI reference of the Expression, which is baseline for the IRI reference of the Manifestation [35].

The IRI of the Item represent the unique reference to a specific resource saved in a precised location and time.

The structural design and field specifications defined within the Akoma Ntoso framework for the generation of identifiers are conforming to the XML schema employed in the draughting of legislative documents. Therefore, this approach ensures adherence to established standards for the formalisation and interoperability of document creation and exchange throughout the legislative process [14].

The Akoma Ntoso standard provides an innovative framework for generating structured identifiers for legal and legislative entities, thereby enhancing document navigability and interconnectivity. In the implementation proposed in this thesis, Akoma Ntoso is used as a reference model to structure document content in a manner that is both human-readable and machine-interpretable, allowing the identification of resources without disclosing their full content.

Chapter 3

Italian Use Case for Legislative Process Management

Article 55 of the Italian Constitution defines the Parliament as composed of the Chamber of Deputies and the Senate of the Republic [8].
It is defined as *perfect bicameralism*, since the two chambers are endowed with the same functionalities with only very slight structural differences within their composition [4].

Even though bicameralism defines the robustness of Italy's democratic structure, its consequences are also the main reasons for certain inefficiences. The primary reason lies in the time-consuming nature of the legislative process, due to the continuous exchange of texts between the two chambers prior to approval [4].
During the last legislatures, these limits were enhanced by the numerosity of decree laws pronounced to speed up the process [32].
These delays can be attributed to logistical complexities inherent in a decentralised system, where the coordination and exchange of documents across multiple independent entities employ more time than a centralised system might. Therefore, while translating into digitalised systems, it is necessary to implement a decentralised approach to be aligned with the constitutional principles, while asserting the impact on the timeliness and immutable ledger

capabilities throughout the whole process.

Decentralisation is fundamentally designed to uphold the integrity of the legislative process, ensuring that each governmental branch can operate independently without interference from centralised authority. Its aim is to insert by design the separation of powers and the respect of the rule of law. Its importance is explicitly reinforced by the Italian Constitutional Court's ruling no. 235/2015 [7], where is stressed the constitutional necessity to prevent any form of centralisation that could infringe upon the separation of powers [34].

Another important constitutional democratic foundation is the democratic participation, which is essential for the legitimacy and accountability of governance, ensuring the effectiveness of legislative systems' operations. It encompasses the mechanisms through which citizens engage in the political systems, influencing decision-making and hold public officials accountable. This principle is strongly connected with the concept of transparency, ensuring that citizens are informed and able to participate in public affairs.

In Italy, the transparency to allow citizen participation in the democratic process is enforced in law 241 of 1990 [44], which regulates administrative procedures and the right of citizen access to administrative documents. Recent amendments to the law have introduced requirements for the online publication of legal norms to facilitate broader public access. These updates align with the provisions of Legislative Decree No. 33 of 2013, commonly known as the Freedom of Information Act (FOIA), which affirms the right of individuals to access data and information held by public administrations, even in the absence of a legitimate interest [41].

3.1 The Italian Legislative Process

As touched upon in Chapter 1 and then described in Section 3, the Italian legislative system is designed with respect to the constitutional democracies' foundational principles.

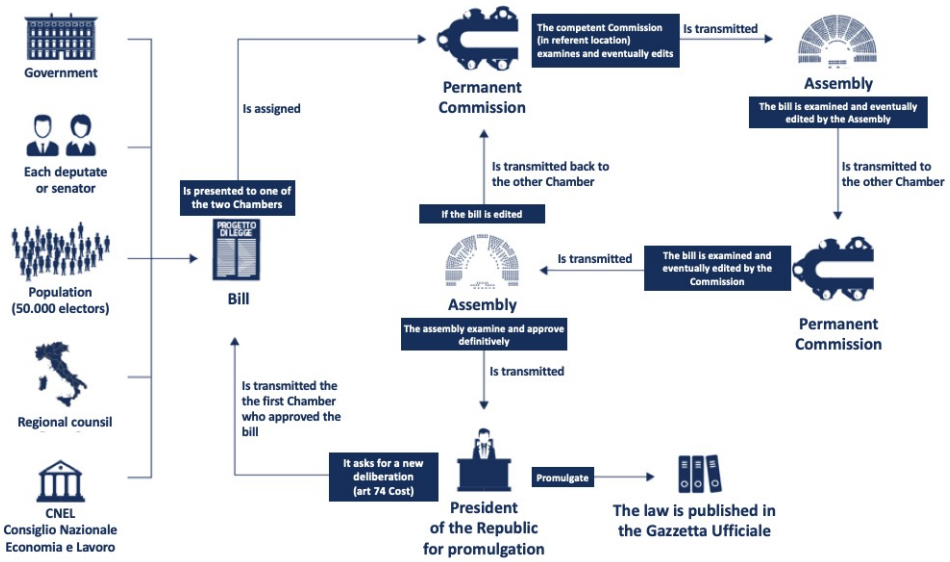

Figure 3.1: Depiction of the italian legislative process[1]

The whole legislative process is a coordinated series of acts aimed at a single final outcome, namely the enactment of a formal law.
The legislative process is organised and divided into three phases: the *legislative initiative*, which consists in the presentation of a bill to one of the chambers; the *legislative deliberation by the chambers*, during which the bill is discussed and examined; and finally, the *promulgation* of the law [4].

The legislative deliberation by the chambers entails the examination of a bill by the competent standing committee. The functions of the committee depend on the location in which it is called to examine the bill. Similarly, the role of the assembly also differs according to the location in which it is convened.

Regarding the different functions performed by the committee and the chamber, three main procedures are distinguished: the *ordinary procedure* with a reporting committee, the procedure with a *deliberative or legislative committee*, and the procedure with a *legislative drafting committee*[2].

[1]Credits for original image: `https://conoscere.camera.it/il-ruolo-della-camera/la-camera-esamina-le-leggi/il-percorso-di-una-legge`

[2]in italian they are defined as: procedimento ordinario per commissione referente, pro-

Once the work in one chamber is completed, the bill is transmitted to the other chamber. Here, the approval process begins anew, as the second chamber is free to choose the procedure to follow. As a result, the bill may pass several times from one chamber to the other until both chambers have approved an identical text. Only at that point is the approval phase considered complete.

The scenario terminates with the promulgation phase, confirmed by the President of the Republic. The President must ensure that the laws are in conformity with the Italian Constitution and holds a final decision-making power, which may be exercised through the referral of the law back to the chambers.

3.2 Description of the problem

In the field of legislation, the balancing of both efficiency and transparency is of the essence for modern legislatures in parliamentary democracies [43].

Significant challenges arise from the absence of a unified document repository capable of simultaneously ensuring the separation of powers as mandated by law. The lack of a centralised repository promotes redundant procedures, increasing risks of delays in legislative workflows and requiring manual reconciliation of documents across diverse platforms, risking the introduction of further human error [34].

Furthermore, the fragmented nature of the current system hampers the tracing of the progression and chronological trajectory of legislative measures. The lack of traceability undermines transparency and results in significant challenges for the review and analytical examination of legislative evolution.

In times, there have been presented some solutions, e.g., Normattiva, which is based on the use of innovative information technologies aimed at

cedimento per commissione deliberante o legislativa, procedimento per commissione redigente

providing a reliable, free, and comprehensive service for information on Italian laws [37]. But it still remains only as a consultative platform, resolving merely the access to information challenge.

The lack of interoperability is still present, largely due to the existence of isolated data silos created by disconnected infrastructures. This fragmented configuration hinders effective communication between systems, leading to information gaps or significant data loss, thereby further complicating the legislative process.

With the introduction of the Interoperability Act of the EU [12], the request to assess the level of interoperability within the public administrations to reduce the burden and improve services, and also to improve democracy, increased the need to develop a solution.

For these reasons, this work presents a distributed solution aimed at implementing a unified repository that upholds the principle of separation of powers while ensuring precise version control of legislative documents. The goal is to design a decentralised architecture that integrates multiple blockchains to provide solutions to the previously identified issues, aiming to enhance the interoperability, security, and traceability of legislative processes. Guaranteeing immutability, transparency, and decentralisation by definition, the blockchain offers a secure and verifiable framework for managing legislative amendments and maintaining the integrity of institutional exchanges.

Chapter 4

Design and Implementation

4.1 Architecture

In order to ensure the separation of powers within a unified repository, it is essential to design an architecture composed of autonomous yet interconnected components. These modular elements should support the management of the various stages of the legislative process while remaining independent from one another, thereby preserving confidentiality and allowing for greater adaptability to existing systems. Therefore, this work proposes the implementation of a multi-leveled structure in which each level relies on a specific type of blockchain technology best suited to its intended functionalities. The design should facilitate seamless integration and conversion should the solution be applied in different contexts.

Figure 4.1 presents the proposed architecture in its entireness. Starting from the bottom, the first level consists of a Private Institutional Blockchain, where each module represents an independent legislative institution. These institutions manage their documents privately inside a private blockchain and, upon completing the legislative deliberation process, communicate and share information with other institutions via the central level.

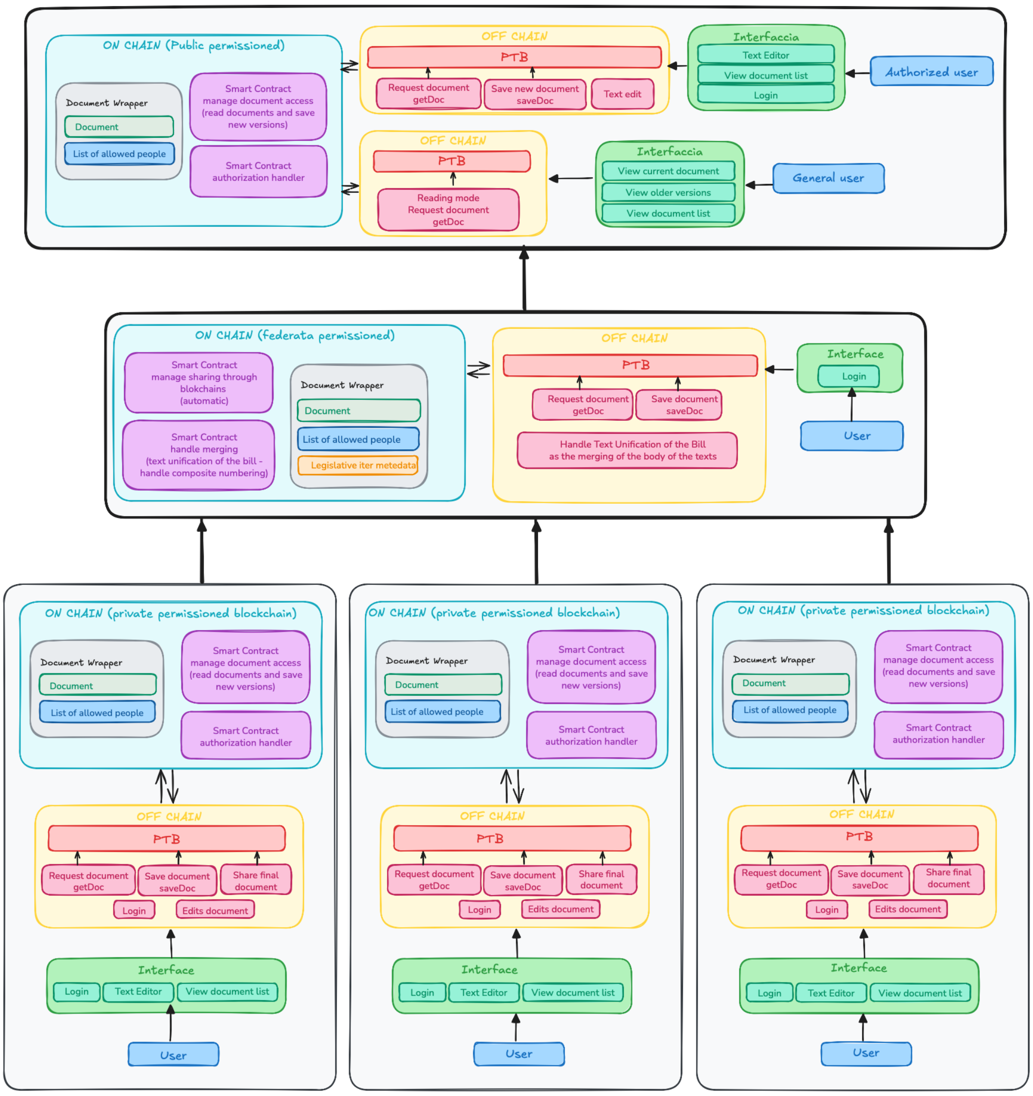

Figure 4.1: Complete architecture of the Multi-level Architecture for Legislative Process

The central level is both a public and private blockchain, acting as a coordinator among institutions and enabling the management of official unpromulgated legislative documents. It is also tasked with handling the communication with the uppermost level, ultimately representing the conclusion of the legislative process.

The top level, referred to as the Public Legislative Blockchain, manages the

publication of officially promulgated legal texts. It not only ensures the democratic principle of citizens' participation and engagement, but also provides a unified platform for institutions to easily update legal documents.

4.1.1 Level 1 - Private Institutional Blockchain

Each individual institution requires a private environment for the discussion and draughting of documents. Virtually, this environment must be implemented as an isolated software, accessible exclusively to members of the relevant committee. As previously described, it is essential for committees to have a private space where even controversial proposals can be debated before reaching a mature form. Therefore, the design includes a private institutional blockchain to safeguard the deliberative autonomy of the committees.

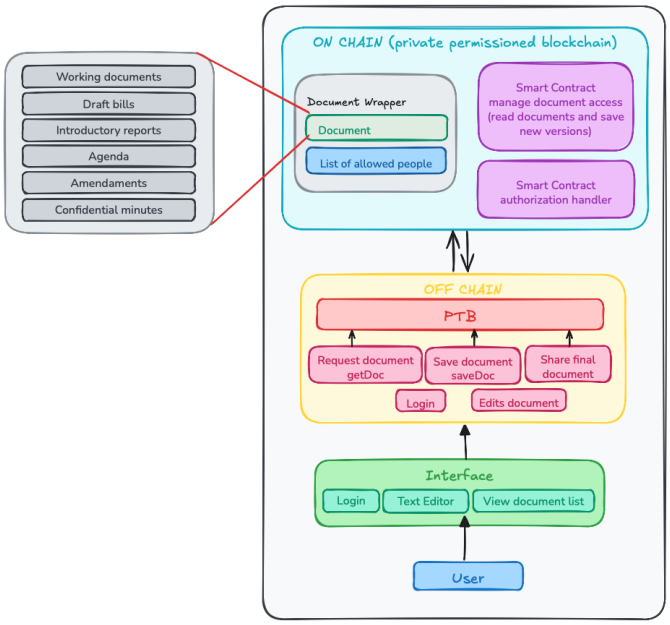

Figure 4.2: Composition of the Private Institutional Blockchain

The module defined as Level 1 (Figure 4.2) comprises in its structure three distinct components: an on-chain section, an off-chain section, and a user interface.

The on-chain component defines methods and conditions under which documents are stored and accessed. The implemented smart contracts regulate access based on the properties of identities described in Section 2.4, leveraging the programming logic of the Move language to implement a secure and reliable document access protocol.

The off-chain component constitutes the off-chain logic layer. It manages document modification operations and serves as an intermediary between the user's intentions expressed throughout the interface, and the logic by which these actions are executed by the system. This component is also entitled with the responsibility to elaborate the documents as the Akoma Ntoso standard requires. The off-chain component is the link between the interface and the on-chain logic, meaning it needs to implement access controls in order to respect the Zero Trust Authority pattern, identifying throughout the process the user and checking if its actions are within its role's authorisation limits.

Lastly, the user interface is of outmost importance in facilitating a seamless user experience while ensuring operational efficiency and functional relevance. It includes a text editor for the composition of documents and provides an efficient mechanism for defining document access policies. Furthermore, it assumes a critical role in overseeing and managing user sessions, thereby strengthening security measures and mitigating the risk of human errors that could potentially lead to systemic vulnerabilities.

Access to this level is restricted exclusively to individuals authorszed by the institution. The blockchain's governance model reflects this requirement through the implementation of a Proof of Authority (PoA) consensus mechanism, wherein validator nodes are solely managed by the institution itself. This ensures full institutional control over internal operations. Such a security-by-design approach is essential, given the sensitivity and importance of the documents processed at this stage of the legislative procedure, which include working papers, draft bills, introductory reports, agendas, amendments, and confidential minutes.

4.1.2 Level 2 - Inter-Institutional Coordination Blockchain

Once the draft bill receives approval from the committee, it is forwarded to the other Chamber for examination. Should the receiving Chamber introduce any amendments, the bill must be returned to the originating Chamber, thereby initiating the process commonly referred to as the "*navetta parlamentare*" (shuttle procedure). When the bill is ultimately approved by both Chambers, it is then submitted to the President of the Republic for final approval.

The exchange of documents between institutions is managed by the second level, which is built upon a federated permissioned blockchain. This type of blockchain, also referred to as a hybrid blockchain, combines characteristics of both public and private ledgers. Within this architecture, it enables the public sharing of documents among private institutional entities.
The governance model of the federated blockchain has an impact regarding the separation of power. The federated permissioned blockchain has consensus with validator nodes distributed proportionally among participating institutions, ensuring no single entity controls inter-institutional communications [33].

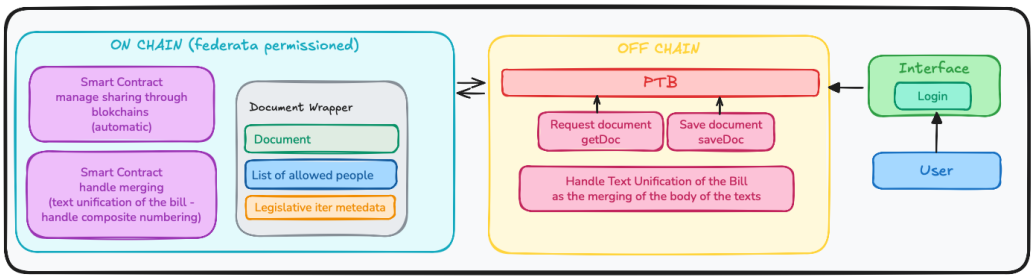

Figure 4.3: Composition of the Inter-Institutional Coordination Blockchain

As with the previous level, Level 2 (Figure 4.3) is structured around three core components: the on-chain section, the off-chain section, and the user interface.

The on-chain component defines the smart contracts responsible for managing the transmission of documents between private institutions. It is tasked with handling and reconciling the identifiers of the bills, as well as the metadata generated through these operations. Moreover, the smart contracts must implement robust access control mechanisms to ensure that only authorised individuals are permitted to access the documents.

The off-chain component, in conjunction with the user interface, must provide a text editor to enable users to review and supervise new versions of legislative texts resulting from previous edits or resulting from the merge of more than one bill. It is also responsible for monitoring user actions to ensure they remain within the boundaries of their authorised permissions. Additionally, the off-chain component handles the implementation of the authentication logic within the user interface, thereby ensuring secure and verified access to the system.

The user interface is essential for enabling user interaction with the system. It must ensure clear understanding of the current stage of the legislative iter and support the definition and management of the individuals authorised to interact with the documents.

The role of this level is crucial in its mediating position, as it formally replicates the legislative exchanges that occur within the Italian legislative iter. Its features enable the generation of immutable audit trails and versioning for formal inter-institutional exchanges, ensuring compliance with the principle of transparency while preserving institutional autonomy and control.

4.1.3 Level 3 - Public Legislative Blockchain

Once the bills have completed the legislative iter, they are published in accordance with the principle of publicity of normative acts. Therefore, they are recorded on the third level, referred to as the Public Legislative Blockchain.

The Public Legislative Blockchain has a public-permissioned architecture, using institutional validator nodes maintaining write authority while providing unrestricted public read access. This structure is organised in accordance with the principles of transparency and citizen participation.

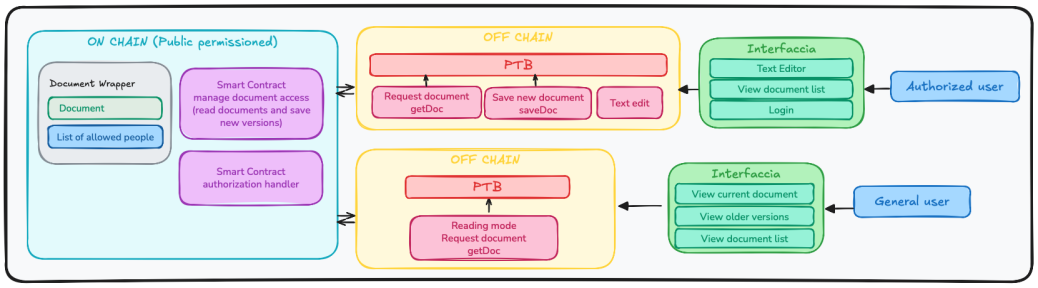

Figure 4.4: Composition of the Public Legislative Blockchain

It is specifically designed to manage the official and definitive versions of legislative documents. While it ensures public access for consultation, writing and modification rights are reserved exclusively for designated authorised identities.

The composition of Level 3 (Figure 4.4) is of three main components: on-chain section, off-chain section and user interface. However, the off-chain and user interface components differ according to the type of user accessing the platform.

The on-chain component is a unified repository of definitive versions of promulgated legislative acts, official legal texts and constitutionally mandated public documents. The smart contracts defined within this component manage document access and handle version control and updates.

The two types of user contemplated to interact with this level are General Users and Authorised Users.
For General Users, access to the platform is limited to viewing the publicly available information related to the most recent version of each document. Therefore, the interface and the off-chain components will implement a read-

ing policy, implementing a facade pattern[1] to show only public informations. Authorised Users are permitted to update the current version of the text, following a logic similar to the one applied in the first level. Consequently, both the user interface and the off-chain components must implement a robust and secure authentication mechanism to verify user identity. This ensures that operational boundaries are clearly defined, specifying which actions the user is authorised to perform and which documents may be edited.

This public blockchain embodies the constitutional principle of legislative transparency, serving as the digital equivalent of official government gazettes.

4.2 Implementation

Regarding the overall scope of the architecture presented in Section 4.1, the work carried out in this thesis focused on the implementation of smart contracts for access control and document storage, designed for the Private Institutional Blockchain.

Within the proposed use case, it is vital to define by-design implementations that can assure transparency while monitoring users' actions with respect to the Authentication, Authorisation, and Accounting (AAA) framework.

This section outlines the various solutions implemented to complete the on-chain component of the Private Institutional Blockchain, as well as the technology employed to test the performance of the developed system.

4.2.1 Document and DocumentWrapper

Documents constitute the core of the proposed architecture due to their central role throughout the entire legislative process. The *Document* entity is initially conceived as a legislative initiative, with the potential to evolve into a bill draft. However, given the intended use of blockchain technology,

[1]The Facade or Façade pattern is a structural pattern design that provides a simpler interface for a defined entity.

particularly for its traceability features, the Document entity has been abstracted to accommodate and encapsulate any type of document that can be described and defined through an Akoma Ntoso International Resource Identifier (IRI).

The *Document* entity stores the information regarding its own on-chain identifier (ID), the identifier of the document within the legislative process (usually defined as `S.XXX` or `C.XXX`, based on the chamber that is currently analysing the text), the document hash, which ensures traceability and immutability, the Akoma Ntoso IRI of the document, and, if applicable, the hash of the previous version of the document.

In order to enhance security, the concept of Wrapper [23], as defined within the Move IOTA framework, has been employed to prevent the Document from being directly exposed on the blockchain.

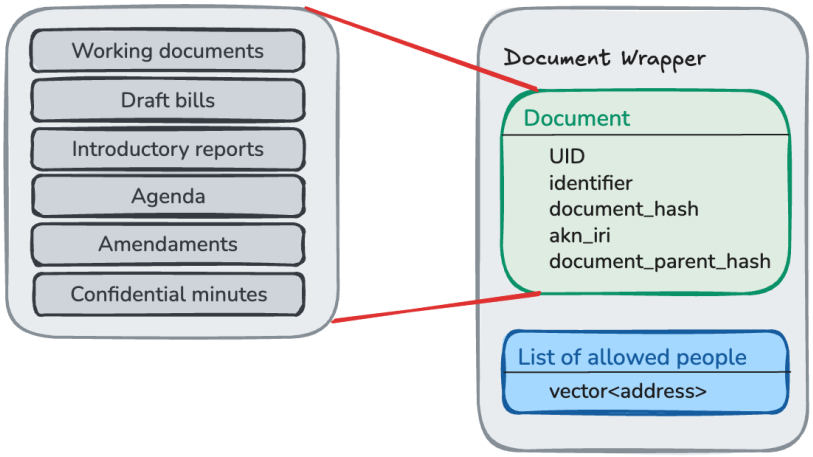

Figure 4.5: Document and Document Wrapper implementation

By definition, the Wrapper encapsulates the Document ID, such that the Document no longer exists independently on-chain [23]. Ownership of the Document is thus transferred to the Wrapper, reinforcing the overall security of the system. Since access to the information is intended to be restricted solely to authorised individuals, this structure enables the integration of both the document and the list of permitted users into a single entity.

Therefore, the *Document Wrapper* entity is introduced, enabling verification of the identity of the transaction sender and preventing any unauthorised access to non-public information. This is achieved by embedding multiple checks within each callable function, in accordance with the Zero Trust paradigm, thereby ensuring continuous validation of the sender's identity against the list of authorised entities. Furthermore, the Document's ownership is given to the Document Wrapper, ensuring that only the authorised people can operate on it via the smart contracts regulations.

Whenever an authorised user requests access to a document's information on-chain, the smart contract returns a structured interface representing the document. The *Visible Document* entity is designed to present information in a form suitable for human interpretation while retaining all critical metadata and technical details within the system's backend. This solution enables secure access to relevant data while preserving. The implementation of the Visible Document is designed with respect to the future development of both the Inter-Institutional Coordination Blockchain and the Public Legislative Blockchain. It already provides a potential solution for publicly displaying legislative documents, thereby upholding the principle of transparency and enabling citizens to exercise oversight over governmental actions.

4.2.2 Akoma Ntoso Ontology

The smart contracts implement a module containing the implementation of an interface which ables the composition of Akoma Ntoso standard International Resource Identifier (IRI), in order to create an identifier on-chain which can help humans and machines identify the context of the document without revealing its inner content.

The interface follows the composition required in the standard [35], implementing a Item composed by: Work, Expression and Manifestation. The composition of the Work and the Expression follows the requirements of the standard [35], consisting of their required and optional fields.

The Manifestation, which indicates the electronic or physical format of the Expression, has as value the Unique Identitificator (UID) of the Document before its incorporation within the Document Wrapper, which, as depicted in the previous section, causes the Document to be not accessible inside the blockchain due to IOTA security implementation.
This choice allows compliance with the definitions required by the standard while simultaneously ensuring a minimum level of information is described, preserving the confidentiality of the documents while enabling their identification.

The Akoma Ntoso ontology interface implements the Virtual Expression as well, while omitting the implementation of Components and Portions, considering that the whole document is saved on-chain and any retrieval request returns the complete document.

4.2.3 IOTA Identities Integration

To establish the identities of users interacting with the smart contracts, the IOTA Identities framework has been selected in order to maintain coherence within the system while leveraging a secure, reliable, and robust implementation.

Although IOTA Identities are defined as shared Move objects [19], their interaction with the Move language has been enhanced through the development of a simple interface that provides improved control over the definition and management of identities.

Specifically, the `intelligible_identity` package was developed to implement the correct anatomy of Decentralised Identifiers (DID), which is thereafter put to use in the creation of the identities that interact with the system.
The main function of the `did_move` module takes as input a complete DID address, then it performs controls on the address to assess conformity to the anatomy defined in the IOTA DID Method Specification v2.0[19]. If the

address is correct with respect to the specifications, then the DID entity is created.

```
module intelligible_identity::did_move{
...
    //DID stands for decentralised Identifier Document
    public struct DID has drop{
        //iota-specific-idstring = "did:iota:"
        iota_did: String,
        //iota-network = 8lowercase-hex
        iota_network: Option<u8>,
        //iota-tag = "0x"64lowercase-hex
        iota_tag: u32,
        metadata: String,
    }
...
}
```

Figure 4.6: DID document implementation

The creation of a DID is utilised to create a new IOTA Identity inside the `intelligible_identity` module. The DID address is passed to the previously described functions which assert its conformity before the creation of the entity. Thereafter, the DID entity is immediately used in the constructor of a new IOTA Identity, using the `iota_identity` library.

4.2.4 Exposed functions

The smart contracts implemented for the on-chain component of the Private Institutional Blockchain, expose three main functions enabling the Programmable Transaction Blocks (PTBs), developed on the off-chain component, to interact with the described entities.

The functions exposed implement the Create, Read, Update, Delete (CRUD)

```
module intelligible_identity::intelligible_identity{
...
    //=== Contructors ===
    public fun new(
        did_address: String,
        clock: &Clock,
        ctx: &mut TxContext): ID {
        let did_doc : DID= intelligible_identity::
                           did_move::
                           new_from_address(did_address);

        iota_identity::identity::new(
            option::some(std::bcs::to_bytes(&did_doc)),
            clock,
            ctx)
    }
...
}
```

Figure 4.7: Creation of a IOTA Identity given a DID address

paradigm, designed to align with the specific entities and technologies involved; for example, the Delete function is not implemented, as the immutable nature of blockchain technology inherently prevents the deletion of stored information. The removal of a document can be conceptually achieved by updating the document with specific data indicating its deletion, thereby preserving immutability while reflecting its deprecation within the system.

The creation of a new document requires the input of all information necessary for the definition of the Akoma Ntoso IRI, as well as the instantiation of the Document entity, as previously described. Once both the Item and Document entities have been defined, the Document Wrapper is created. This wrapper includes the specification of the list of authorised identities,

which from that point onwards are the sole entities authorised to access and modify the document.

The reading of a document is permitted only if the transaction sender is authorised to access it. Accordingly, access control mechanisms are enforced based on the sender's identity. If the sender is verified as an authorised user, the corresponding Visible Document entity of the requested document is returned.

Updating a document effectively corresponds to saving a new version of it. This new version requires the same set of information necessary to create a new document, with the addition of a reference to the previous Document Wrapper. In this way, versioning is preserved while maintaining a traceable and immutable history of all changes. Furthermore, the Akoma Ntoso IRI is renewed given the new Manifestation of the Document presented.

4.2.5 Programmable Transaction Blocks

Programmable Transaction Blocks (PTBs) are a feature in IOTA that allow the execution of multiple commands within a single IOTA transaction [25]. This capability enhances efficiency and reduces gas fees compared to processing individual transactions, as PTBs can perform up to 1,024 unique operations in one execution, which would typically require 1,024 individual executions on traditional blockchains. Programmable Transaction Blocks (PTBs) in IOTA are atomic operations, meaning that all the commands within a PTB are treated as a single, indivisible unit of work. The effects of all commands, including object modifications or transfers, are applied atomically at the end of the transaction.

PTBs have been used to test the core functionalities implemented by the smart contract, including the creation of a new document, updating the document's version and its fields, and requesting access to a document.

4.3 Code

In this section some code snippets are provided in order to further explain the implementation presented before.

4.3.1 Manage documents module

The package enabling interaction with documents is divided into two separate modules, responsible respectively for managing the Document and the Document Wrapper.
The `document` module encompasses the Document entity and the Visible Document entity, managing their creation process and regulating access to their information.
The Document entity is defined as having the `key` ability, which allows the struct to be considered an object inside the blockchain and define it as a unique asset. In the meantime, the `store` ability allows the Document asset to be stored inside the Document Wrapper.
The Visible Document possesses both the `copy` and the `drop` abilities, emphasising its role as a disposable representation rather than a persistent asset. This design facilitates the future implementation of functions without imposing constraints related to the use of the visible format. Furthermore, the absence of the `store` ability accentuates the independence and volatility of this object, as it cannot be enclosed within other objects.

The `bundle` module defines a limited set of straightforward functions to ensure compliance with the access control policies governing the documents contained within the Document Wrapper entity.
The Document Wrapper entity is crucial to create a unique reference for both the document and the authorised people. The `key` ability guarantees the asset's uniqueness within the blockchain, while the encapsulated Document entity enables control and monitoring of all actions performed on it.

```
module manage_documents::document{
...
    //== Structs ==
    //Document entity inside the blockchain
    public struct Document has key, store{
        id:UID,
        identifier: String, //may have value S.xxx or C.xxx
        hash: vector<u8>,
        akn_id: ItemIRI,
        parent: Option<vector<u8>>,
    }

    //Interface to return Document information
    public struct VisibleDoc has copy, drop{
        identifier: String,
        parent: vector<u8>,
        akn_id: String,
    }
...
}
```

Figure 4.8: Document and Visible Document structures and abilities

```
module manage_documents::bundle{
...
    //== Struct ==
    public struct DocumentWrapper has key{
        id: UID,
        doc: Document,
        accessList: vector<address>,
    }
...
}
```

Figure 4.9: Document Wrapper structure in the `bundle` module

4.3.2 Akoma Ntoso Ontology

Each Akoma Ntoso component is implemented inside its own struct, having as fields the pieces defined inside the standard [35].

The use of the Option type and its methods to represent and handle an optional value [22], allows the correct implementation and distinction between required pieces and optional pieces inside each component specification.

```
module akn_ontology::work{
...
    public struct WorkIRI has copy, store, drop{
        country_name: String,
        type_doc : String,
        subtype : Option<String>,
        author: Option<String>,
        creation_date: u64,
        number: Option<u16>,
        name: Option<String>
    }
...
}
```

Figure 4.10: Composition of Work struct

The required fields are expected by the constructor method of each struct, while the optional fields are to be set later in time via the setter functions implemented.

```
module akn_ontology::expression{
..
    public struct ExpressionIRI has copy, store, drop{
        human_lang_code: Option<String>,
        char: String,
        identifiers_list : vector<String>,
        content_spec_date: Option<u64>,
        authoriality: Option<String>,
        is_current_version : bool
    }
...
}
```

Figure 4.11: Composition of Expression struct

```
module akn_ontology::manifestation{
...
    public struct ManifestationIRI has copy, store, drop{
        authoring_info: Option<String>,
        manifestation_date: Option<u64>,
        annotation : Option<String>,
        dot: String,
        data_format: String
    }
...
}
```

Figure 4.12: Composition of Manifestation struct

4.3.3 Programmable Transaction Blocks

The code of the Programmable Transaction Blocks (PTBs) presented corresponds with the ones used to test the implementation of the system. Some fields have been kept generic due to the length of their actual values, which could negatively impact the readability of the code presented. All the snippets presented are written for the Command Line Interface (CLI), being

the most agile way to test, obtain and visualise results.

```
1 //PTB for the request to read a document
2 iota client ptb \
3 --move-call "0
    xefb57b389f64f773ae348315a05fb3f0090128fb0859d834dd1948a1775f636b
    ::manage_documents::request_doc" \
4 <DOCUMENT WRAPPER ADDRESS>
```

Figure 4.13: Composition of the Programmable Transaction Block to request the document

Both the PTB for creating a new document (Figure 4.14) and the PTB for saving a new document version (Figure 4.15) are relatively complex, due to the large number of parameters required to satisfy the constructors of the various entities involved. To ensure the correct typing of the input values, they are passed through functions that apply type casting, thereby delivering variables in the appropriate format as required by the target function and in compliance with its specifications.

```
1 //PTB for the creation of a new document
2 iota client ptb \
3 --make-move-vec "<address>" "<[ALLOWED PEOPLE ADDRESSES]>" \
4 --assign vec_identities \
5  --move-call "0x1::string::from_ascii" "\"<CONTENT OF THE
     DOCUMENT>\"" \
6  --assign my_string \
7  --move-call "0x1::string::into_bytes" my_string \
8  --assign bytes \
9  --move-call "0x1::hash::sha2_256" bytes \
10 --assign hash_doc \
11 --move-call "0x1::string::utf8" "\"C.123\"" \
12 --assign identifier \
13 --move-call "0x1::string::from_ascii" "\"it\"" \
14 --assign lang \
15 --move-call "0x1::string::from_ascii" "\"bill\"" \
16 --assign name \
17  --move-call "0x2::vec_map::empty" "<0x1::string::String, 0x1
     ::string::String>" \
18  --assign expr_vec_map \
19  --move-call "0
     xefb57b389f64f773ae348315a05fb3f0090128fb0859d834dd1948a1775f636b
     ::manage_documents::create_document" \
20  hash_doc vec_identities identifier lang name @0x6 true
     expr_vec_map
```

Figure 4.14: Composition of the Programmable Transaction Block to create a new document

```
1 //PTB for the saving of a new document version
2 iota client ptb \
3 --make-move-vec "<address>" "<[ALLOWED PEOPLE ADDRESSES]>" \
4 --assign vec_identities \
5  --move-call "0x1::string::from_ascii" "\"<NEW CONTENT OF THE
     DOCUMENT>\"" \
6  --assign my_string \
7  --move-call "0x1::string::into_bytes" my_string \
8  --assign bytes \
9  --move-call "0x1::hash::sha2_256" bytes \
10 --assign hash_doc \
11 --move-call "0x1::string::utf8" "\"C.124\"" \
12 --assign identifier \
13 --move-call "0x1::string::from_ascii" "\"it\"" \
14 --assign lang \
15 --move-call "0x1::string::from_ascii" "\"bill\"" \
16 --assign name \
17  --move-call "0x2::vec_map::empty" "<0x1::string::String, 0x1
    ::string::String>" \
18  --assign expr_vec_map \
19  --move-call "0
    xefb57b389f64f773ae348315a05fb3f0090128fb0859d834dd1948a1775f636b
    ::manage_documents:: save_new_version" \
20     <OLD DOCUMENT WRAPPER ADDRESS> vec_identities hash_doc
    identifier lang name @0x6 true expr_vec_map
```

Figure 4.15: Composition of the Programmable Transaction Block to save a new document version

Chapter 5

Results

To evaluate the scalability and performance of the system, an analysis of transaction costs was conducted.

The three primary functions designed to interact with the off-chain layer and the interface layer were tested using the Programmable Transaction Blocks (PTBs) presented in Section 4.3.3, to estimate the gas consumption[1] of each function.
The results have been very promising, as shown in Table 5.1.

Costs	**Create a new document [38]**	**Save a new document version [40]**	**Request document [39]**
Storage Cost	0,0058216	0,0092112	0,0058216
Computation Cost	0,001	0,001	0,001
Computation Cost Burned	0,001	0,001	0,001
Storage Rebate	-0,0009804	-0,0041192	-0,0058216
Non-refundable Storage Fee	0	0	0
Total Gas Fee	0,0068412	0,007092	0,002

Table 5.1: Transactions gas costs expressed in IOTA

[1]Gas refers to the computational effort required for executing operations on blockchain. In IOTA, gas is paid with the network's native currency IOTA. The cost of executing a transaction in IOTA units is defined as the transaction fee [20].

The situation reported in Table 5.1 refers to operations that involve a document associated with nine authorised identities (which represent the primary source of storage cost), a text of 129 words hashed using SHA-256, resulting in a 256-byte vector, and each component of the Akoma Ntoso IRI.

The entries considered inside the table are the following:

- *Storage Cost*: indicates the total cost required to write or update data inside the blockchain;
- *Computation Cost*: indicates the computational cost required to execute the code and the logic of the used smart contracts;
- *Computation Cost Burned*: refers to the total of computational cost burnt and never returned, paid as a prevention measure against spam;
- *Storage Rebate*: total cost returned to the user when the transaction is complete; therefore, it is indicated with a negative sign;
- *Non-refundable Storage Fee*: storage costs kept from the system to cover the permanent writing on the net.

The results show how the non-refundable computational costs are fixed at one-thousandth of an IOTA per operation, indicating that operations related to internal execution logic within smart contracts are inherently lightweight. The most expensive measured component is the Storage Cost, which represents the total cost of writing or updating data on the blockchain. Nevertheless, the gas required to complete the operations remains relatively low, even in scenarios involving document overwriting after saving a new version.

The operation of requesting and accessing the document in read-only mode is fully rebated except for the computational costs. Since this operation is expected to be performed frequently, its cost-effectiveness is highly advantageous within the development and application context.

The total costs per functionality are presented in Figure 5.1, calculated with respect to the market value of IOTA as of June 2025, denominated in

both dollars and euros. For both currencies, the overall cost per operation is exceedingly low, highlighting the economic efficiency of the implementation.

Figure 5.1: Single transactions costs expressed in USD and EUR given the current value of the IOTA crypto (June 2025)

In order to clearly demonstrate the low economic impact of this implementation, which is built using IOTA technologies, an estimation of the annual operational costs and their expected frequency has been carried out. Calculations were carried out using the euro market value of IOTA as in June 2025, with a comparative analysis conducted using the SUI token, converted to euros as well.

The estimated numbers for each operation are presented in Table 5.2. The analysis aims to account for metrics relevant to the Italian use case, including an assumed volume of operations based on the provisions presented in the article "*Normative production of XIX legislature*"[32].

Figure 5.2 illustrates the progression of total costs per operation type as the number of operations increases. This projection highlights how the use of IOTA enables cost containment even in the long term. Therefore, it is possible to affirm the system's scalability across various institutions, as the operational costs remain extremely low.

	Estimated number of annual operations	Annual costs IOTA	Annual Costs SUI
create new doc	50.000	€ 40,84	€ 123,97
request doc	150.000	€ 41,70	€ 126,58
save new version	150.000	€ 147,87	€ 341,11

Table 5.2: Estimated annual operation costs

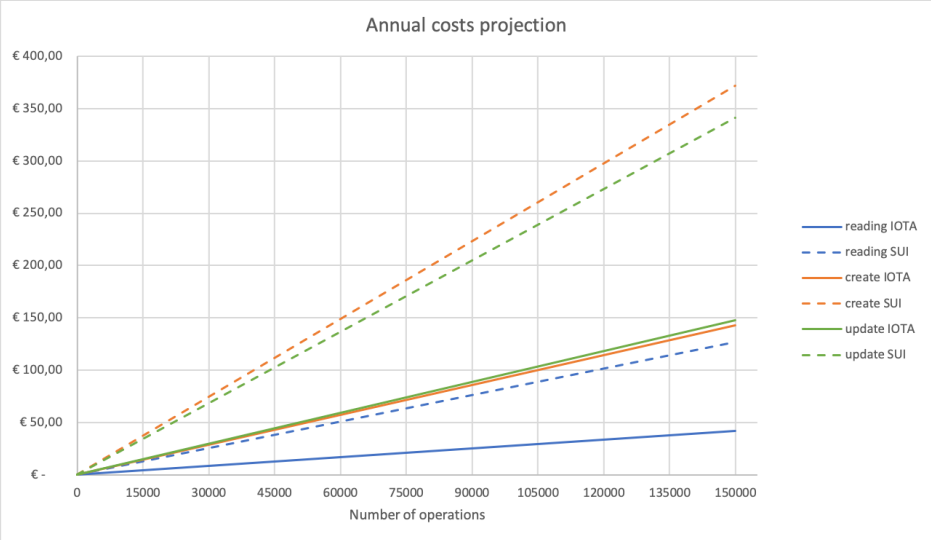

Figure 5.2: Annual costs projection of all operations, comparing IOTA and SUI costs

Comparative analyses involving other major blockchains, such as Ethereum, are considerably more complex due to fundamental differences in their technological implementations. While a comparison between SUI and IOTA can be conducted relatively straightforwardly, given their shared reliance on Directed Acyclic Graph (DAG) structures, a proper performance evaluation against Ethereum necessitates a more nuanced approach. Specifically, such an analysis must take into account the monolithic architecture of Ethereum, which is characterised by its global state model.
Furthermore, any study concerning scalability must carefully consider the baseline gas fees present in Ethereum, which fluctuate dynamically in re-

(a) Create a new document **(b)** Save a new version of the document **(c)** Request and reading document operation

Figure 5.3: Different operations costs in annual projection

sponse to market demand, as well as the varying response times of its validator nodes. Consequently, in order to establish a meaningful basis for performance comparison with Ethereum, it would be necessary to completely redevelop the implemented smart contracts in Solidity. Even then, the analysis must be conducted cautiously with particular attention to the contextual limitations and assumptions underlying the results.

Chapter 6

Conclusion

This thesis aimed to explore how blockchain technology can be used to support digitalization of legislative processes while respecting core democratic principles. With the Italian legislative system serving as a case study, the purpose of the investigation was to design and validate the practical applicability of the proposed architecture through a preliminary partial implementation.

The current state-of-the-art within this use case is represented by xLeges, a decentralised communication infrastructure that effectively provides secure and efficient interinstitutional exchange of legal documents. Although this system provides a foundation for digital legislative processes, its limitations generated evolving requirements for transparency, accountability, and data consistency. Consequently, this research proposes an architectural framework that incorporates blockchain technology to address these challenges.

The proposed solution is deployed within the IOTA ecosystem, utilising its technological infrastructure to provide an efficient and scalable system enhanced by the computational logic of the Move programming language. The integration of IOTA Identities and the Akoma Ntoso XML standard facilitated the development of an architectural component defined by smart contracts specifically designed for legislative document management within a Private Institutional context.

The implementation was subsequently subjected to testing procedures to validate its performance characteristics and assess its practical applicability. The empirical results demonstrate that the implemented solution exhibits superior efficiency and cost-effectiveness compared to alternative blockchain platforms. These findings enable inferences regarding the scalability potential of the system and confirm the validity of the proposed implementation.

The work presented in this thesis represents an initial step toward the comprehensive realisation of the proposed architecture, establishing the foundation for the development of each architectural level. Future research may build upon this groundwork by focusing on more advanced aspects of the implementation. In particular, additional efforts could also investigate the creation of a communication bridge to enable a fully operational system. Moreover, future studies might address the challenge of complete text storage, a component that lies beyond the scope of the present study.

In conclusion, the proposed solution demonstrates a solid foundation and appears to have selected the appropriate technologies. The proposed architecture offers a structured approach to addressing existing challenges in legislative system design through the application of blockchain technology, a powerful yet underexploited tool in the governance domain.

The advancement and broader adoption of secure digital systems can facilitate the interaction between institutions and citizenry, holding the potential to significantly enhance the efficiency and transparency of political processes. These systems serve not only to mitigate operational inefficiencies and reduce structural opacity within legislative processes, thereby facilitating the functions of political actors, but also to enhance communication with the citizenry. In turn, they foster greater public trust in institutional frameworks and contribute to the preservation of the fundamental principles of democratic governance in an increasingly digital and interconnected global context.

Bibliography

[1] Alevizos, Lampis ; Ta, Vinh T. ; Hashem Eiza, Max: Augmenting zero trust architecture to endpoints using blockchain: A state-of-the-art review. In: *SECURITY AND PRIVACY* 5 (2021), November, Nr. 1. – URL http://dx.doi.org/10.1002/spy2.191. – ISSN 2475-6725

[2] Alvarez, Edgar: Facebook's Calibra cryptocurrency wallet launches in 2020. In: *Engadget Blog* (2019), June. – URL https://www.engadget.com/2019-06-18-facebook-calibra-libra-cryptocurrency-digital-wallet.html. – visited 2025-07-02

[3] Alyas, T. ; Abbas, Q. ; Niazi, S. u. a.: Multi blockchain architecture for judicial case management using smart contracts. In: *Scientific Reports* 15 (2025), Nr. Article 8471. – URL https://doi.org/10.1038/s41598-025-92842-8

[4] Bin, Roberto ; Pitruzzella, Giovanni: *Diritto Costituzionale*. Bd. XXI. G. Giappichelli Editore, 2020. – 38–46 / 77–85 S

[5] Blackshear, Sam ; Cheng, Evan ; Dill, David L. ; Gao, Victor ; Maurer, Ben ; Nowacki, Todd ; Pott, Alistair ; Qadeer, Shaz ; Rain ; Russi, Dario ; Sezer, Stephane ; Zakian, Tim ; Zhou, Runtian: Move: A Language With Programmable Resources. (2019). – URL https://developers.diem.com/papers/diem-move-a-language-with-programmable-resources/2019-06-18.pdf

[6] Cabinet Office of the Government of the United King-

DOM: *Legislative process: taking a bill through Parliament.* – URL https://www.gov.uk/guidance/legislative-process-taking-a-bill-through-parliament. – visited 2025-06-23

[7] CORTE COSTITUZIONALE ITALIANA: *GIUDIZIO PER CONFLITTO DI ATTRIBUZIONE TRA ENTI.* – URL https://www.cortecostituzionale.it/actionSchedaPronuncia.do?anno=2015&numero=235. – visited 2025-06-24

[8] COSTITUZIONE DELLA REPUBBLICA ITALIANA: *Art. 55.* 1948. – URL https://www.senato.it/istituzione/la-costituzione/parte-ii/titolo-i/sezione-i/articolo-55. – visited 2025-06-24

[9] DE FILIPPI, Primavera ; MANNAN, Morshed ; REIJERS, Wessel: The alegality of blockchain technology. In: *Policy and Society* 41 (2022), 02, Nr. 3, S. 358–372. – URL https://doi.org/10.1093/polsoc/puac006. – ISSN 1449-4035

[10] DIEM ENGINEERING TEAM: *The language of money, Part one: Why build Move?* January 2021. – URL https://www.diem.com/en-us/blog/why-build-move/. – visited 2025-07-02

[11] EUROPEAN, Parliament: *Ordinary legislative procedure.* – URL https://www.europarl.europa.eu/infographic/legislative-procedure/index_en.html. – visited 2025-06-23

[12] EUROPEAN PARLIAMENT: *REGULATION (EU) 2024/903.* – URL https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=OJ:L_202400903. – visited 2025-06-26

[13] EUROPEAN PARLIAMENT: *Regulation (EU) No 910/2014.* – URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014R0910. – visited 2025-07-02

[14] Francesconi, Enrico: The "Norme in Rete" Project: Standards and Tools for Italian Legislation. In: *International Journal of Legal Information* 34 (2006), 01

[15] Ghasemshirazi, Saeid ; Shirvani, Ghazaleh ; Alipour, Mohammad: Zero Trust: Applications, Challenges, and Opportunities. (2023), 09. – URL 10.48550/arXiv.2309.03582

[16] Hayes, Adam: *Blockchain Facts*. – URL https://www.investopedia.com/terms/b/blockchain.asp. – visited 2025-06-23

[17] IOTA Stiftung: *About IOTA*. – URL https://docs.iota.org/about-iota. – visited 2025-06-30

[18] IOTA Stiftung: *Consensus on IOTA*. – URL https://docs.iota.org/about-iota/iota-architecture/consensus. – visited 2025-06-30

[19] IOTA Stiftung: *IOTA DID Method Specification v2.0*. – URL https://docs.iota.org/developer/iota-identity/references/iota-did-method-spec#anatomy-of-the-encoded-did-document. – visited 2025-06-30

[20] IOTA Stiftung: *IOTA Glossary*. – URL https://docs.iota.org/iota-glossary. – visited 2025-07-03

[21] IOTA Stiftung: *IOTA Identity Framework*. – URL https://docs.iota.org/developer/iota-identity/. – visited 2025-06-30

[22] IOTA Stiftung: *Move - Module std::option References*. – URL https://docs.iota.org/references/framework/std/option. – visited 2025-06-28

[23] IOTA Stiftung: *Move - Wrapped Objects References*. – URL https://docs.iota.org/developer/iota-101/objects/object-ownership/wrapped. – visited 2025-06-28

[24] IOTA Stiftung: *Transaction Life Cycle.* – URL https://docs.iota.org/about-iota/iota-architecture/transaction-lifecycle. – visited 2025-07-02

[25] IOTA Stiftung: *Use Programmable Transaction Blocks.* – URL https://docs.iota.org/developer/iota-101/transactions/ptb/programmable-transaction-blocks. – visited 2025-06-28

[26] IOTA Stiftung: *Why Move?.* – URL https://docs.iota.org/about-iota/why-move. – visited 2025-07-02

[27] Istituto dell'Enciclopedia Italiana: Stato di diritto. In: *Enciclopedia Italiana* (2025). – URL https://www.treccani.it/enciclopedia/stato-di-diritto_(Dizionario-di-Storia)/. – visited 2025-06-23

[28] Italian, Parliament: *The passage of Law through Parliament.* – URL https://en.camera.it/4?scheda_informazioni=15. – visited 2025-06-23

[29] Kumar, Rohit ; Agarwal, Harshit ; Tayal, Ananya ; Nagaraja, Hema: Courtsafe: Legal Records Storage & Management Using Blockchain. (2024). – URL https://doi.org/10.1145/3675888.3676140. ISBN 9798400709722

[30] *The Move Book.* – URL https://move-language.github.io/move/. – visited 2025-07-02

[31] *Decreto Legislativo 7 marzo 2005 n. 82 - Art. 20.* – URL https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.legislativo:2005-03-07;82. – visited 2025-06-30

[32] Openpolis: *La produzione normativa nella XIX legislatura.* – URL https://www.openpolis.it/la-produzione-normativa-nella-xix-legislatura/. – visited 2025-06-24

[33] Palmirani, Monica ; Papalia, Ludovico ; Bomprezzi, Chantal ; Arruzzoli, Arianna ; Zichichi, Mirko ; Ferretti, Stefano: *A Multi-Chain Approach to Transparent and Accountable Legislative Processes*. June 2025

[34] Palmirani, Monica ; Papalia, Ludovico ; Bomprezzi, Chantal ; Zichichi, Mirko ; Arruzzoli, Arianna ; Ferretti, Stefano: Multi-level Architecture for Separation of Powers in Legislative Process. In: *Proceedings of the 7th Distributed Ledger Technology Workshop (DLT 2025)*. Pizzo, Italy : CEUR-WS.org, June 2025 (CEUR Workshop Proceedings)

[35] Palmirani, Monica ; Sperberg, Roger ; Vergottini, Grant ; Vitali, Fabio: *Akoma Ntoso Version 1.0*. August 2018. – URL `http://docs.oasis-open.org/legaldocml/akn-core/v1.0/os/part1-vocabulary/akn-core-v1.0-os-part1-vocabulary.html`. – Latest version: `http://docs.oasis-open.org/legaldocml/akn-core/v1.0/akn-core-v1.0-part1-vocabulary.html`

[36] Panek, Crystal: *Understanding authentication, authorization, and accounting*. S. 33–109. In: *Security Fundamentals*, 11 2019

[37] Portale Normattiva, Versione 2.4.2: *Normattiva - Il progetto*. – URL `https://www.normattiva.it/staticPage/progettoObiettivo`. – visited 2025-06-24

[38] : *Create New Document Transaction*. – URL `https://iotascan.com/testnet/tx/BbsMMpJsKExvejNf9YiA1XoiTJmqGLhtnVWKeegd6MAF`

[39] : *Request Document Transaction*. – URL `https://iotascan.com/testnet/tx/5b5ktZn7F537HABwFHfUkHwo8kjbXbihhjg9vnwXQmMU`

[40] : *Save New Document Version Transaction*. – URL `https://iotascan.com/testnet/tx/3EC7HRKGMabKwezAYQTdqYzVN48rQvfLm6v2DrmhbKAv`

[41] Repubblica Italiana: *Decreto trasparenza (d. lgs. n.33/2013)*. – URL https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.legislativo:2013-03-14;33!vig=. – visited 2025-06-26

[42] Servizio Studi - Dipartimento istituzioni: L'informatizzazione delle pubbliche amministrazioni - Il quadro normativo. In: *Documentazione e ricerche* 31 (2008), 11. – visited 2025-07-02

[43] Voermans, Wim ; Napel, Hans-Martien ten ; and, Reijer P.: Combining efficiency and transparency in legislative processes. In: *The Theory and Practice of Legislation* 3 (2015), Nr. 3, S. 279–294. – URL https://doi.org/10.1080/20508840.2015.1133398

[44] X legislatura della Repubblica Italiana: *LEGGE 7 agosto 1990, n. 241*. – URL https://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:legge:1990-08-07;241. – visited 2025-06-24

[45] Santis, Luca ; Lupo, Caterina ; Marchetti, Carlo ; Mecella, Massimo: The x-Leges System: Peer-to-Peer for Legislative Document Exchange, 09 2006, S. 231–242

[46] Massimi, Fabio: *x-Leges: stato dell'arte e prospettive future*. 05 2015. – URL https://www.agid.gov.it/sites/default/files/repository_files/documentazione/20150525_x-leges_agid_f_massimi_v_1_0.pdf

[47] Zichichi, Mirko: *Decentralized systems for the protection and portability of personal data*, alma, Dissertation, April 2023. – URL https://amsdottorato.unibo.it/id/eprint/10662/

Ringraziamenti

Il principale ringraziamento va al Professor Ferretti, che con la sua energia mia coinvolta in questo interessantissimo lavoro. Grazie per i continui stimoli, per la fiducia riposta in me, e per tutte le opportunità, la ringrazio davvero dal profondo.

Un ringraziamento d'obbligo per Mirko, che mi ha seguita con pazienza nei miei ragionamenti più contorti con modo gentile e simpatico. Grazie per tutte le soluzioni offerte nei momenti in cui ero più incastrata.

Un ringraziamento speciale a papà, che con il suo "ma se invece ce la fai?" mi ha incoraggiata e spronata anche nei momenti più bui e incerti. Grazie per offrirmi sempre dei pacati consigli e per sostenermi. Ti voglio bene.

Alla mia famiglia, ai miei amici, a tutti coloro che ho avuto la fortuna di incontrare in questo percorso, grazie per avermi accompagnata e per aver reso questo viaggio più leggero.

Per aspera ad astra.