

Article

BRA-PS: A Blockchain Reference Architecture for Public Sector Citizen-Centric Applications

Sion Israel Sion , Kaiwen Zhang and Alain April 

Department of Software & IT, École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada;
kaiwen.zhang@etsmtl.ca (K.Z.); alain.april@etsmtl.ca (A.A.)

* Correspondence: sion-israel.sion.1@ens.etsmtl.ca

Abstract

Public sector organizations are under increasing pressure to modernize service delivery while preserving transparency, interoperability, accountability, and citizen trust. Blockchain technology offers relevant capabilities for these objectives, particularly through shared ledgers, cryptographic verification, and programmable rules. However, its adoption in public sector contexts remains constrained by the lack of architectural guidance tailored to inter-organizational services. This study proposes BRA-PS, a Blockchain Reference Architecture for Public Sector Citizen-Centric Applications, developed from a real-world digitalization project in Quebec, Canada. The architecture organizes components into six layers (presentation, business, communication, smart contract, blockchain, and data) with cross-cutting concerns addressing governance, access control, security, and monitoring. A key design principle is the public–private workflow separation, which enables inter-organizational collaboration while preserving each organization’s operational autonomy and data confidentiality. We validated the architecture through a case study involving a vehicle registration process between two public agencies, supported by a proof-of-concept implementation using Hyperledger Fabric. An Architecture Trade-off Analysis Method (ATAM) evaluation, conducted with a panel of five domain experts, identified six architectural risks, including InterPlanetary File System (IPFS) confidentiality exposure and smart contract inflexibility, six non-risks, six sensitivity points, and six trade-offs across three key quality attributes: autonomy, collaboration, and functional suitability. The results show that BRA-PS can support implementation decisions, clarify stakeholder responsibilities, and expose relevant architectural trade-offs. The recommendations derived from the evaluation provide practical guidance for the adoption of blockchain in citizen-centric public sector services.

Keywords: blockchain; reference architecture; public sector; interoperability; citizen-centric services; ATAM; architecture trade-off analysis method



Academic Editor: Manuel Mazzara

Received: 28 April 2026

Revised: 12 June 2026

Accepted: 6 July 2026

Published: 8 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Blockchain technology has significant potential in the public sector because it can support transparency, efficiency, and trust in government services [1]. Yet, as public organizations explore blockchain-based systems, they still lack clear architectural guidance for integrating this technology into operational services. This creates difficulties for stakeholders and architects who must translate blockchain capabilities into concrete, maintainable, and accountable systems. Traditionally, architects relied on accepted design concepts

to create high-quality designs with predictable results [2]. These concepts include reference architectures, architectural design patterns, deployment patterns, tactics, and certain externally developed components such as frameworks [3].

Reference architectures are particularly useful in this context because they provide a common framework for the design and implementation of a technology. They help stakeholders develop a shared vocabulary, make informed decisions, and align their efforts around a coherent structure. Without such guidance, blockchain adoption may lead to fragmented implementations and an underuse of the technology's potential. This issue became evident during our work on a blockchain-based proof of concept in the public sector.

In light of these considerations, this study proposes the following research question: What reference architecture can be employed to facilitate stakeholder alignment and guide blockchain integration in citizen-centric public sector services? To answer this question, the study pursues three research objectives: (RO1) to design a reference architecture that organizes blockchain-based public sector applications into coherent layers and cross-cutting concerns; (RO2) to validate the architecture through a real-world case study demonstrating interoperability between public agencies while preserving organizational autonomy and data privacy; and (RO3) to evaluate the architecture's trade-offs and quality attributes using the Architecture Trade-off Analysis Method (ATAM) method, producing actionable recommendations for practitioners. These objectives guide the structure of the paper: Section 3 addresses RO1, Section 4 addresses RO2, and Section 5 addresses RO3.

To address this problem, this paper proposes BRA-PS, a reference architecture for blockchain-based, citizen-centric applications in the public sector. Existing blockchain reference architectures, as shown in Table 1, address interoperability and security within business or industry-specific contexts but share three gaps when considered for public sector adoption: none are designed for citizen-centric service delivery, none address public sector governance and accountability requirements, and none employ structured architectural trade-off evaluation. BRA-PS addresses these gaps through three concrete design choices: a public-private workflow separation that enables inter-organizational collaboration while preserving each agency's operational autonomy; a centralized off-chain governance model suited to regulatory accountability requirements; and a citizen-facing Presentation Layer designed to make blockchain states transparent to non-technical users. To validate our approach, we applied the proposed architecture to a case study involving a vehicle registration process between two public agencies. This paper makes an architectural contribution: its primary goal is to propose and evaluate a reference architecture, not to report experimental performance results. Consistent with established software architecture research practice, we rely on Kruchten's 4+1 view model for design [4] and the ATAM method for evaluation [5], which is the recognized standard for assessing architectural decisions and trade-offs [2]. Empirical results from the proof-of-concept implementation, including transaction outputs and process validation, are reported in our companion paper [6].

This paper complements our previous work [6] by shifting the focus from implementation feasibility to architectural structure, design rationale, quality attributes, and trade-offs.

The remainder of this paper is organized as follows. Section 2 provides the background and discusses related work. In Section 3, the proposed reference architecture is introduced. Section 4 outlines the approach and details of the case study and system design. Section 5 presents the evaluation results. Finally, Section 6 discusses the findings and concludes the paper.

Table 1. Comparison of Blockchain Reference Architectures.

Architecture	Citizen-Centric	Public Sector	Governance Model	Privacy Mechanism	Interoperability	Evaluation Method
Viswanathan [7]	No	No	Business network governance	Member-based access	Cross-organization	Not specified
Liu [8]	No	No	Not specified	Channel-based	API-level	Not specified
Wang [9]	No	No	Supply chain governance	Role-based access	Microservice integration	Prototype
Gong [10]	Partial	No	Incentive-based	Participant-based	Task distribution	Simulation
Weber [11]	No	No	Compliance-driven	Cryptographic proofs	Cloud provider integration	Case study
Aviv [12]	No	No	Distributed	Channel-based	Hybrid cloud	Prototype
Gohar [13]	Partial	No	Patient-centric	Five-tier privacy	IoT/cloud integration	Case study
BRA-PS (ours)	Yes	Yes	Centralized, off-chain, coercive	Public-private workflow separation	Legacy system integration	ATAM + PoC

2. Background and Related Work

2.1. Reference Architecture

Reference architectures describe the logical structure of a family of applications within a specific domain. They provide a reference model aligned with one or more architectural patterns, typically supplemented by artifacts that facilitate their use [3]. The purpose of reference architecture is to standardize practices and align efforts with a common strategic vision, acting as a guide for creating coherent architectures across an organization [14].

Unlike architectural styles, which describe general types of components and relationships in a technology-agnostic way (e.g., “layered architecture” or “microservices”), reference architectures are domain-specific and often integrate multiple styles to suit particular needs. For instance, a layered architectural style organizes a system into presentation, business, and data layers, establishing a clear separation of responsibilities without prescribing specific components. However, a reference architecture for a web application may define these layers more concretely by specifying the components and roles within each layer, such as user interface, business, data access, and service agents. Additionally, reference architectures incorporate crosscutting concerns, such as security and communication, ensuring that these essential elements are considered throughout the design, even if they are not immediately visible in the component structure [3].

In this sense, a reference architecture acts as a standardized technical framework that aligns system design with strategic and organizational objectives. It promotes consistency across implementations and provides reusable principles for future systems.

2.2. Blockchain Technology

Blockchain is a distributed ledger technology designed to address issues of time, trust, and verification in multi-party environments. Its core features include immutability and a shared ledger, where transactional updates are verified through a consensus-based system, enabling digital interactions among multiple parties without the need for a central authority [15].

Smart contracts are fundamental to blockchain platforms, as they allow users to enforce rules in transaction processing. These self-executing protocols independently carry out the terms of an agreement, making it possible to automatically validate steps and encode conditions. On a blockchain platform, participants reach consensus on data formats within

smart contracts, along with the rules governing transactions, a process that involves precisely defining rules, exploring exceptions, and setting up frameworks for dispute resolution [16].

In the public sector, blockchain's benefits extend to reducing economic costs, time, and automation in information exchanges between government and public-private entities. Verified data on the blockchain are secure from alteration or falsification, supporting reliable public-sector operations through distributed ledgers and pre-programmed, streamlined contracts [17].

By establishing a shared chronological record of transactions and events, blockchain can help organizations coordinate activities across boundaries, improve process visibility, reduce disputes, and strengthen trust [18,19]. It can also replace some human-based verification procedures with cryptographic authorization and peer-to-peer automation, thereby streamlining workflows and contractual interactions [20].

Three terms are central to this paper and are defined here for consistency. A permissioned blockchain is a distributed ledger in which participation is restricted to known, authenticated entities, as opposed to public blockchains where anyone can join anonymously. A public-private workflow refers to the separation between shared inter-organizational processes visible to all blockchain participants (public) and internal organizational processes that remain confidential (private). A citizen-centric approach denotes a service design paradigm in which the needs, expectations, and experience of the citizen drive architectural and process decisions, rather than institutional or technical constraints [21].

2.3. Related Work

Blockchain reference architectures aim to provide a structured framework that ensures coherence, security, and interoperability within the systems. These architectures are designed to meet domain needs while addressing technical and governance challenges unique to blockchain-based systems [22].

The literature distinguishes between generic blockchain reference architectures and architectures tailored to specific sectors. Studies such as [7,8] proposed models intended for broad applications, whereas other studies have adapted architectures to address domain-specific needs.

Early frameworks such as Blockchain Solution Reference Architecture proposed by [7], focused on forming business networks, integrating members, managing transactions, and establishing governance structures. This foundational model enables organizations to incorporate blockchain into existing systems, with security and governance considerations in place. Similarly, ref. [8] introduced a layered architecture for segmenting blockchain functions into infrastructure, platform, Application Programming Interface (API), and user levels, thereby facilitating interoperability and modularity.

Several studies have focused on specific domains of application. For example, ref. [9] developed a microservice-based architecture for supply chains to enhance product traceability and stakeholder verification. Ref. [10] focused on crowdsourcing and proposed an architecture that supports distributed task management and participant incentives using blockchain principles. Ref. [11] proposed a blockchain-based architecture for automated, transparent compliance in cloud applications. It uses smart contracts and cryptographic proofs to ensure secure, verifiable, and efficient compliance adjustments between cloud providers and organizations. In distributed information systems, ref. [12] designed a blockchain-native architecture that uses distributed ledger technology principles to improve availability, scalability, and security across hybrid cloud environments, with a framework built on Hyperledger Fabric (HLF) to support secure, interoperable operations. Additionally, ref. [13] introduced a patient-centric healthcare architecture that uses blockchain, cloud, and IoT to enhance semantic interoperability. This five-tier

framework protects electronic health records and allows healthcare providers to share data safely. These domain-specific architectures rely on core blockchain technologies such as consensus mechanisms, smart contract execution, and inter-chain interoperability to ensure robust and reliable systems.

From an architectural standpoint, these proposals reveal recurring limitations for the public sector context. The business-oriented models of [7,8] offer strong governance and modularity but treat participants as organizational actors, leaving no architectural provision for a non-technical end user as a first-class stakeholder. The domain-specific architectures [9,11,12] achieve interoperability within their verticals through microservices or cryptographic attestation, yet their privacy mechanisms are tied to organizational roles rather than to the public–private process distinction that inter-agency public services require. Healthcare and crowdsourcing models [10,13] come closest to a user-centered orientation, but their notion of the user is a patient or a worker within a single platform, not a citizen interacting across autonomous public agencies. Across all of these, evaluation is conducted through prototypes or simulations that demonstrate feasibility without exposing the architectural trade-offs that arise when autonomy, collaboration, and transparency must be balanced simultaneously.

Table 1 summarizes the comparison between BRA-PS and existing blockchain reference architectures across six dimensions relevant to public sector citizen-centric applications.

The six comparison dimensions were selected based on two complementary sources. First, the quality attribute framework of [2] and the reference architecture design principles of [3] identify interoperability, privacy, governance, and domain specificity as essential dimensions for assessing architectural fitness in domain-specific contexts; this motivates the inclusion of governance model, privacy mechanism, interoperability, and evaluation method as dimensions. Second, three gaps were identified inductively from the reviewed architectures: none are designed for citizen-centric service delivery, none address public sector governance and accountability requirements, and none employ structured architectural trade-off evaluation, motivating the citizen-centric and public sector dimensions. The “No” and “Partial” classifications in Table 1 are based on explicit statements in each source paper: for example, ref. [7] targets business network formation with no mention of citizen-facing interfaces; ref. [8] specifies governance only at the API level without addressing public sector accountability; and ref. [13] is classified as “Partial” for citizen-centricity because it prioritizes patient data management over citizen-driven process design [21].

As shown in Table 1, existing architectures address interoperability and security within their respective domains but share three common limitations when considered for public sector adoption. First, none are explicitly designed for citizen-centric service delivery, where the end user is a member of the public rather than a business stakeholder. Second, none address the specific governance and accountability requirements of public sector organizations, such as regulatory compliance, democratic oversight, and multi-agency coordination. Third, evaluation methods in existing work rely primarily on prototypes or simulations, without a structured architectural assessment of trade-offs.

BRA-PS addresses these gaps through a layered architecture specifically designed for public–private workflow separation, a centralized off-chain governance model suited to inter-agency coordination, and an ATAM-based evaluation that systematically identifies risks and trade-offs. The comparison in Table 1 positions BRA-PS with respect to gaps identified in the literature rather than asserting empirical superiority over the listed architectures. Establishing relative superiority would require subjecting each architecture to the same ATAM evaluation under an identical utility tree and deployment context, which is outside the scope of this study. The contribution of BRA-PS is therefore framed as

addressing requirements unmet by existing architectures in the public sector citizen-centric context, not as outperforming them on a common benchmark.

The citizen-centric orientation of this work is grounded in the public value tradition of public administration [23], which holds that public services should be designed to create value for citizens rather than merely to optimize internal administrative efficiency. In the digital government literature, this translates into citizen-centric and user-centered service design, where the citizen's needs and experience drive technological choices [21,24]. Recent work on the public value of e-government further argues that transparency and trust are themselves outcomes that digital infrastructures must actively produce [25].

3. Proposed Reference Architecture

3.1. Design Process and Rationale

The development of BRA-PS followed an iterative process grounded in three inputs: requirements elicited from a real-world blockchain digitalization project in the Quebec public sector [6], a systematic review of existing blockchain reference architectures [7–9], and stakeholder consultations conducted throughout the project. Initial requirements were collected through workshops with representatives from both public agencies involved in the case study, covering functional needs (process automation, document traceability, citizen access) and quality attribute priorities (autonomy, interoperability, security). These requirements were mapped to architectural decisions using the Attribute-Driven Design (ADD) method [3], which guided the decomposition of the system into layers and the identification of cross-cutting concerns.

Requirements elicitation was conducted in close collaboration with the partner public institution responsible for vehicle registration, through working sessions held over the course of the project [6]. Rather than relying on an informal list, requirements were derived through a traceable process: the deficiencies of the existing process were first documented as concrete areas of improvement, which were then translated into a set of ten design goals covering digitalization, automation, traceability, authentication, collaboration, security, citizen-centricity, and interoperability [6]. These design goals provided the link between stakeholder needs and architectural decisions, and the alignment between the participating organizations was maintained by validating each goal against the shared public workflow rather than any single organization's internal process. Architectural decisions were validated iteratively throughout development: each design goal was traced to the proof-of-concept implementation and confirmed through a scenario-based walkthrough of the vehicle registration process, in which every step was mapped to the design goals it satisfies [6]. This traceability from improvement areas to design goals, and from design goals to implemented and validated scenarios, ensured that the architecture remained grounded in the elicited requirements at each stage.

Alternative decompositions were considered and rejected: a flat microservices structure was dismissed due to governance complexity in a multi-organization public sector context, and a two-tier client-server model was dismissed as insufficient to isolate the public blockchain workflow from private organizational processes. The resulting six-layer structure and its cross-cutting concerns were iteratively refined through the case study implementation and subsequently validated through the ATAM evaluation described in Section 5. The citizen-centric orientation of BRA-PS manifests concretely in three architectural decisions: the Presentation Layer is explicitly designed to expose blockchain states in non-technical language accessible to citizens without blockchain expertise; the public-private workflow separation ensures that citizens interact only with public processes, protecting organizational data while maintaining process transparency; and the

digital wallet mechanism gives citizens direct ownership of their identity credentials, reducing dependence on institutional intermediaries.

In this study, citizen-centricity is treated as an architectural design principle, operationalized through the Presentation Layer, the public–private workflow separation, and the digital wallet mechanism. Empirical validation of the resulting user experience remains outside the scope of this architecture-centric evaluation and is identified as a necessary next step in Section 6.1.

The purpose of a reference architecture is to provide a structured foundation that connects business objectives with technical solutions and translates use cases into actionable designs [14,26]. A well-crafted architecture anticipates changes in technology and evolving requirements, minimizing business risks while providing the flexibility to adapt over time. This involves careful consideration of trade-offs between quality attributes, such as performance and security to meet the diverse needs of stakeholders [2].

BRA-PS provides a structured framework for blockchain-based applications that require both inter-organizational collaboration and the confidentiality of internal organizational processes. This architecture is divided into six distinct layers: presentation, business, communication, smart contract, blockchain, and data layers. Figure 1 illustrates the Blockchain Reference Architecture for Public Sector Citizen-Centric Applications (BRA-PS), which facilitates the integration of blockchain technology with both user-facing components and legacy systems.

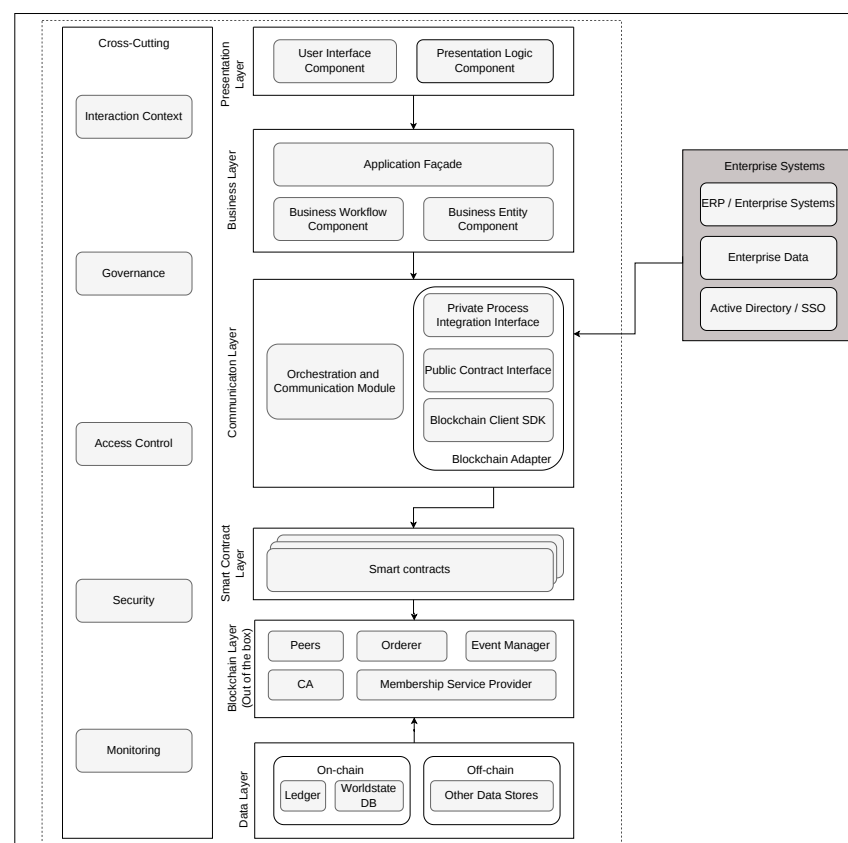


Figure 1. Blockchain Reference Architecture for Public Sector Citizen-Centric Applications (BRA-PS).

3.2. Presentation Layer

The Presentation Layer is the primary interface through which users interact with the system. Its role is to expose blockchain-related information in a way that improves transparency without requiring users to understand blockchain technical details. This layer consists of two main components.

- **User Interface Component:** This component is responsible for rendering all interactive elements of the application, such as buttons, forms, and visualizations that showcase blockchain states. It aims to make blockchain features, such as transaction confirmations and workflow stages, clearly visible and understandable to users. This layer was introduced in response to a stakeholder requirement that citizens be able to monitor the status of their requests without requiring technical knowledge of blockchain operations (SC5, SC6).
- **Presentation Logic Component:** This component manages the logic that processes and formats blockchain data for the user interface. By converting raw blockchain data (e.g., transaction status and workflow stages) into user-friendly displays, this component ensures that blockchain operations are effectively communicated to the users.

This layer addresses a recurring limitation of blockchain applications: although transactions may be transparent at the ledger level, their meaning is often not accessible to non-technical users.

3.3. Business Layer

The Business Layer contains the application's core business logic. It manages workflows and entities independently from the underlying blockchain infrastructure. This layer includes the following components:

- **Application Façade:** This component acts as a gateway for the application's business services, providing a simplified interface for accessing and interacting with the business logic. It abstracts the complexity of the backend processes, allowing other layers to interact simultaneously with the business services.
- **Business Workflow Component:** This component manages the execution of business-specific workflows, such as processing applications or approving transactions. It operates independently of the blockchain, but integrates with it to align private workflows with the public workflow exposed through the Communication Layer.
- **Business Entity Component:** This component is responsible for the management and persistence of business entities. It provides the necessary logic to manage entities, such as users, transactions, and documents, ensuring that they are handled according to business rules.

The Business Layer is therefore the functional core of the application: it defines and executes the processes that fulfill the application's business objectives. Separating business logic from blockchain logic was a deliberate decision to preserve each organization's ability to evolve its internal processes independently of the shared public workflow, directly supporting the autonomy quality attribute (SC1, SC2).

3.4. Communication Layer

The Communication Layer is central to the inter-organizational logic of BRA-PS. It implements a public–private workflow approach, which distinguishes between two categories of processes. Public processes define the shared inter-organizational steps that are visible to all blockchain participants and executed through smart contracts, in the vehicle registration case, these include certificate issuance events and registration status transitions. Private processes remain internal to each organization and are never exposed on the blockchain, in the vehicle registration case, these include Org2's internal tax calculation logic and Org1's internal document validation procedures. This separation, grounded in inter-organizational process theory [27], allows each organization to maintain full control over its internal operations while participating in a shared, transparent public workflow. The full empirical demonstration of this approach is reported in [6]. This layer includes:

- **Orchestration and Communication Module:** This module ensures synchronization between public and private processes, managing events and signals between the blockchain and each organization's internal systems. It coordinates interactions to maintain coherence across diverse organizational workflows.
- **Blockchain Adapter:** Serving as a core interface for managing blockchain interactions, the Blockchain Adapter is designed with flexibility to adapt and evolve as needs change. It encompasses the following components:
 - **Private Process Integration Interface:** Provides a secure interface for organizations to connect their internal processes with public workflows without disclosing sensitive data, allowing private workflows to align with the shared, public framework.
 - **Public Contract Interface:** Contains the public workflow logic that operates on the blockchain, representing the shared rules and states visible to all participants, thereby supporting transparency and common understanding among stakeholders.
 - **Blockchain Client Software Development Kit:** Provides essential tools and libraries for interacting with smart contracts on the blockchain, enabling the application to execute contract functions and effectively manage blockchain transactions.

A key consequence of this event-driven coordination concerns error handling when a private process cannot complete. In BRA-PS, the public smart contract state is advanced only after the corresponding private process has completed successfully: the Orchestration Module commits a public state transition in response to a confirmed private outcome, not in anticipation of it. As illustrated in Figure 2, when a private process fails before commit, no public state transition occurs and the citizen is notified to resubmit, so there is no inconsistent state to reverse. Because committed ledger state is immutable, the architecture does not rely on rollback; in the residual case where a failure is detected after a commit, the unfavorable outcome is recorded as an explicit forward transition (a compensating transaction moving the document to a rejection state). This design keeps each organization's internal failure handling private while ensuring that the shared public workflow remains consistent and auditable.

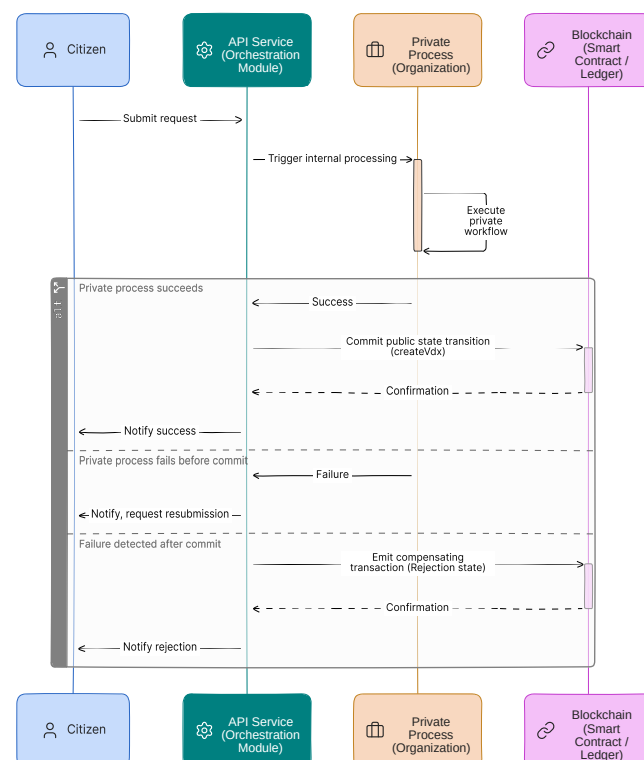


Figure 2. Error-handling sequence in the Communication Layer.

The communication layer ensures that organizations can engage in a unified workflow, exposing only the necessary public steps while retaining control over their private operations. This layer represents the most distinctive contribution of BRA-PS. The public–private separation it enforces was identified as the key architectural decision for enabling inter-organizational collaboration without requiring organizations to expose sensitive internal processes (SC3, SC4).

3.5. Smart Contract Layer

The Smart Contract Layer contains the smart contracts that automate the execution of shared business rules on the blockchain. These contracts facilitate transparency and immutability in the public workflow processes. They ensure that shared rules are consistently applied and cannot be tampered with once in place. This layer enforces the rules of the public workflow and improves the reliability and transparency of shared processes. Encapsulating shared rules in smart contracts rather than in each organization’s private system was chosen to ensure that all participants operate under identical, tamper-proof conditions, addressing the automation and compliance requirements (SC10).

3.6. Blockchain Layer

The Blockchain Layer contains the core network components required to build and maintain the blockchain infrastructure. These include network participants, known as peers, and a Certificate Authority (CA) that oversees network authentication. Both peers and the CA can be managed by network members or outsourced to third-party providers through a software-as-a-service arrangement. The use of a permissioned blockchain (HLF) rather than a public chain was driven by the confidentiality and governance requirements of the public sector context, where membership control and regulatory accountability are mandatory. Network members can participate through multiple channels, which enables segmented data sharing.

These requirements are specific to the public sector context and distinguish BRA-PS from business-oriented blockchain architectures: public agencies operate under legal mandates for transparency and accountability, cannot arbitrarily restrict citizen access to process status, and must comply with data protection regulations that preclude storing sensitive citizen data on a public chain.

3.7. Data Layer

The Data Layer manages business-related data through a hybrid combination of on-chain and off-chain storage. This includes the Ledger, which records and appends transaction details in a distributed format shared across network participants, and the World State Database (LevelDB or CouchDB), which keeps track of the most recent values of ledger entries for efficient access. Additionally, this layer integrates decentralized storage systems for securely storing larger files and documents such as images, analytics data, and other information that complements the blockchain solution. In the public sector context, the use of a public decentralized storage network such as InterPlanetary File System (IPFS) introduces confidentiality risks, as content stored on a public IPFS network is accessible to any participant who obtains the content identifier. BRA-PS addresses this risk through three complementary mechanisms: files are encrypted prior to upload using keys managed by the uploading organization, ensuring that content remains unreadable even if the content identifier is discovered; a private IPFS network, deployed within the organizational perimeter and interfaced with the HLF client, restricts content accessibility to authorized network participants only; and only IPFS content identifiers (hashes) are recorded on the blockchain ledger, ensuring that sensitive file content never resides on-chain. Together, these mechanisms allow BRA-PS to leverage the scalability and decentralization benefits

of IPFS while meeting the confidentiality requirements of public sector data governance. The hybrid on-chain/off-chain data strategy was chosen to balance transparency, keeping document states verifiable on the ledger, with performance and cost constraints, as storing large files directly on-chain is prohibitive in a high-volume public service context.

3.8. Cross-Cutting Concerns

In BRA-PS, cross-cutting concerns capture design issues that affect the architecture as a whole rather than a single layer. The main cross-cutting concerns include Interaction Context, Governance, Access Control, Security, and Monitoring.

3.8.1. Interaction Context

The interaction context, as presented in [7], enables the identification of participants and their assigned roles. The type of participation chosen by each organization also has a significant impact on architectural design. Organizations can join the blockchain network in various capacities: larger organizations may operate a peer node, whereas smaller organizations may access services through a presentation-layer user interface provided by another organization's peer node. The interaction context also determines access privileges, guiding which segments of an organization can interact directly with the blockchain network, and which segments will use intermediary interfaces. This flexibility in roles allows for a scalable and adaptable architecture that meets the diverse needs of participating entities.

3.8.2. Governance

Governance is essential for the success of blockchain-based systems, especially when several autonomous organizations share a common infrastructure [16]. For permissioned blockchain solutions, where participation is restricted to authenticated entities, a clearly defined, published, and accessible governance framework must be established to ensure coordination and compliance among the participants. Several important considerations are essential for building a strong governance structure:

- **Shared and Automated Processes:** Establishing agreements on shared and automated processes within the blockchain is essential. This includes defining the automated component, method, deployment location, and mechanisms for integration with all the network members. A consistent mechanism for member participation and endorsement must also be determined to ensure smooth collaboration.
- **Membership Management:** The governance framework must include guidelines for managing membership, encompassing processes for inviting new members, verifying their identities, and announcing new participants within the network. In addition, membership approval, denial, and revocation procedures should be outlined to maintain network integrity.
- **Network Operation and Fee Management:** The governance framework should address how new nodes are deployed, how they impact network functionality, and the structure of fees for network usage. Fee collection methods must be defined to ensure sustainable network operations.
- **Change Management:** Processes for managing changes to membership rules, updating smart contracts, and releasing new versions of the application must be outlined. This includes specifying how modifications are proposed, reviewed, and implemented to maintain system stability and alignment with organizational goals.
- **Dispute Resolution:** A clear framework for resolving disputes must be integrated into the governance model. This includes identifying the types of disputes that may

arise, their relationship with smart contracts, escalation procedures, and the parties responsible for dispute mediation and resolution.

- **Network Regulation:** Defining the entity responsible for regulating the network is essential for ensuring accountability. The governance model should describe how decisions are made and the protocols for reaching consensus on regulatory matters within the network.

In permissioned blockchain systems, these governance components are important in establishing a transparent and coordinated environment. An early consensus on these aspects ensures that all parties are aligned in their decision-making processes, rule enforcement, and member responsibilities, thereby providing a stable foundation for blockchain-based collaboration.

One recommended governance model is a centralized, off-chain, and coercive governance approach.

- **Centralized:** Governance decisions are managed by a well-defined group of organizations, ensuring control and accountability.
- **Off-Chain:** Major decisions are made outside the blockchain, providing flexibility in decision-making and reducing on-chain overhead.
- **Coercive:** Decisions made by the governance group are implemented directly in the smart contracts, ensuring that governance rules are enforced across the network.

This choice aligns with recent analyses of public sector blockchain governance, which stress that governance must be aligned with policy objectives and institutional accountability rather than being treated as a purely technical matter [28].

3.8.3. Access Control

In this architecture, access control is based on two key mechanisms: authentication and privacy.

- **Authentication Mechanism:** In a blockchain network, identity is represented by digital certificates or cryptographic keys stored in a digital wallet. These credentials establish a participant's identity and authorize them to interact with the network.
- **Privacy Mechanism:** Privacy is ensured by restricting visibility of transactions to only those participants registered on a particular channel. This channel-based privacy is a key element for maintaining data confidentiality because it allows sensitive information to remain accessible only to authorized entities.

3.8.4. Security

Security is integral to the design and operation of blockchain applications. The architecture must ensure that data are protected both on- and off-chain. These include data encryption, secure key management, and secure storage. In addition, mechanisms for monitoring security events and detecting anomalies are necessary to ensure the integrity and reliability of the application.

The layers of BRA-PS expose distinct security surfaces, which we analyze using the STRIDE framework (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) [29]. Table 2 relates each STRIDE category to its manifestation in the architecture, the affected layers, and the corresponding mitigation provided by HLF's built-in mechanisms or by an architectural design decision. Spoofing is mitigated by X.509 identities issued through the CA, tampering by the immutable ledger and endorsement policy, repudiation by transaction signing and the complete transaction history, and information disclosure by channel-based privacy combined with off-chain encryption. Denial of service and elevation of privilege extend beyond Fabric's native guarantees and

are handled at the deployment level through API rate limiting, multi-host node distribution, access control lists, and role validation within chaincode. This model identifies the threats most relevant to BRA-PS and grounds the security analysis recommended above.

Table 2. Preliminary STRIDE Threat Model Mapped to BRA-PS Layers.

STRIDE Threat	Manifestation in BRA-PS	Affected Layer	HLF/Architectural Mitigation
Spoofing	Impersonation of a citizen or organization	Access Control; Communication	X.509 certificates issued by the CA; digital wallet identity
Tampering	Unauthorized modification of transaction or document state	Smart Contract; Data	Immutable ledger; endorsement policy; on-chain hashing of off-chain content
Repudiation	Participant denies having issued a transaction	Smart Contract; Access Control	Cryptographic signing of every transaction; complete transaction history
Information Disclosure	Exposure of confidential private-process data	Communication; Data	Channel-based privacy; private IPFS; file encryption prior to upload
Denial of Service	Flooding the API service or peer nodes	Communication; Blockchain	API rate limiting (deployment-level); multi-host node distribution
Elevation of Privilege	Gaining rights beyond an assigned role	Access Control; Smart Contract	Role validation in chaincode; access control lists; channel membership

3.8.5. Monitoring

Monitoring maintains the health, performance, and security of the blockchain network through continuous observation at three levels. At the infrastructure level, node-oriented metrics are tracked, including peer and orderer availability, transaction endorsement and commit latency, block propagation time, and per-node resource consumption (CPU, memory, and bandwidth); HLF exposes these through its operations service and Prometheus-compatible endpoints. At the ledger level, monitoring covers transaction throughput, the rate of endorsement-policy failures, and discrepancies between peers' world-state databases, which can indicate synchronization or consistency issues. At the application level, the API service and Communication Layer are monitored for request rates, error rates, and unexpected call patterns.

Anomaly detection builds on these metrics to identify unauthorized or abnormal behavior. Rule-based thresholds (for example, an abnormal rate of failed endorsements or repeated unauthorized access attempts) provide immediate alerting, while statistical baselining of transaction volume and latency can surface deviations that rule-based thresholds miss, such as gradual performance degradation or atypical access patterns from a given identity. Alerts are routed to the responsible network operators defined in the governance model, closing the loop between monitoring and the accountability structure of the network. Continuous monitoring is particularly important in the public sector context, where the integrity of citizen-facing processes and the early detection of misuse are prerequisites for maintaining public trust.

4. Approach

This study follows a Design Science Research (DSR) paradigm [30], which is appropriate for research whose primary outcome is the construction and evaluation of an artifact. In this case, the artifact is the BRA-PS reference architecture. The DSR cycle is instantiated through three activities: requirements elicitation from a real-world public sector digitalization project, artifact construction using the ADD method [3], and evaluation through a case study implemented with HLF followed by an architecture-centric assessment using ATAM. Figure 3 summarizes the sequence followed in this study. The case study applies the architecture to a real-world scenario, whereas the ATAM evaluation examines the archi-

tectural trade-offs and quality attributes that emerge from the proposed design. ATAM was selected over alternative architectural evaluation frameworks for the following reasons. The Software Architecture Analysis Method (SAAM) [5] is suited to evaluating individual quality attributes in isolation but does not support the simultaneous analysis of multiple competing quality attributes and their trade-offs, which is central to this study given the competing demands of autonomy, collaboration, and functional suitability. The Cost Benefit Analysis Method (CBAM) extends ATAM with economic reasoning but requires quantitative cost and benefit data that were not available at this stage of the project, as the focus is architectural rather than financial. ATAM was therefore the most appropriate choice because it is specifically designed to identify trade-offs between multiple quality attributes, involves stakeholders directly in the evaluation process, and produces actionable risk and sensitivity analyses suited to guiding implementation decisions in complex multi-organization systems [2,5].

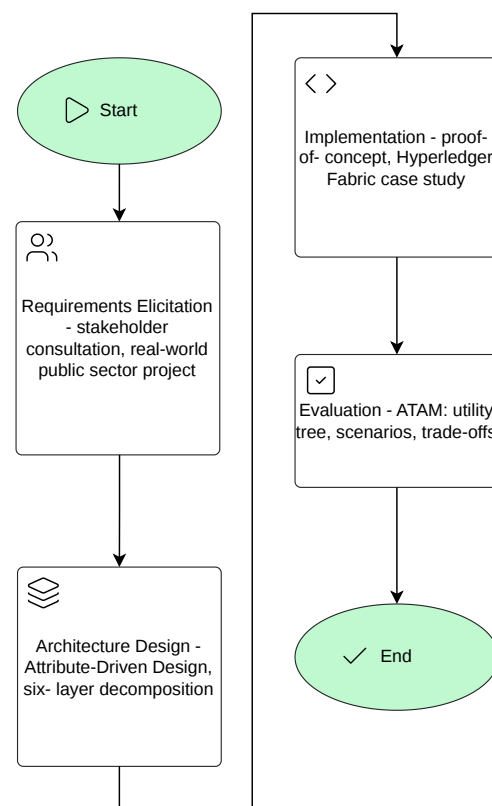


Figure 3. Research methodology: the Design Science Research cycle applied in this study.

4.1. Case Study

This case study demonstrates the architecture's application in the context of vehicle registration within the province of Quebec, Canada, involving two primary organizations: Org1 and Org2. They work together to streamline the vehicle registration process, while maintaining data integrity, transparency, and privacy. In the scenario analysed, Org1 is responsible for managing vehicle registration, whereas Org2 issues a tax certificate, which specifies the tax amount required by Org1 during registration.

The process can be summarized in three steps:

- Step 1: A citizen purchases a vehicle and initiates the registration process by applying for a certificate from Org2. Org2 uses its internal processes to determine the appropriate tax amount and issues a certificate.

- Step 2: The citizen submits the certificate along with other necessary documents to Org1. Org1 processes the application, collects the tax specified in the certificate, and finalizes registration.
- Step 3: Org1 issues a final registration document, VDX, to the citizen, completing the registration process and authorizing the citizen to legally operate the vehicle on the roads of Quebec.

4.2. System Design

To support this case study, the architecture of the proposed solution was presented following Kruchten’s “4+1” view model [4], which provides a comprehensive breakdown of the system. Each view was analyzed in detail and supported by Unified Modeling Language (UML) diagrams.

Table 3 explicitly maps each step of the vehicle registration process to the corresponding BRA-PS layers and components, demonstrating how the reference architecture concretely structures the case study implementation.

Table 3. Mapping of Vehicle Registration Process to BRA-PS Layers.

Step	Process Activity	BRA-PS Layer	Component
1	Citizen applies for tax certificate from Org2	Presentation	User Interface Component
1	Org2 calculates tax internally	Business	Business Workflow Component
1	Org2 issues certificate on blockchain	Smart Contract	VdxPaperContract
1	Certificate stored off-chain	Data	Decentralized Storage (IPFS)
2	Citizen submits documents to Org1	Presentation	User Interface Component
2	Org1 validates certificate via blockchain	Communication	Public Contract Interface
2	Org1 internal validation process	Business	Business Workflow Component
2	Org1 collects tax, updates blockchain state	Smart Contract	VdxPaperContract
3	Org1 issues VDX registration document	Communication	Private Process Integration Interface
3	Citizen receives and monitors status	Presentation	Presentation Logic Component

4.2.1. Logical View

This view elaborates the Smart Contract Layer of BRA-PS, detailing how VdxPaperContract implements the shared public workflow rules defined in the architecture. The logical view addresses the functionality of the system from an end-user perspective, focusing on smart contract operations. A smart contract plays a key role in managing data interactions on the blockchain using a unique identifier that combines a document number and the user’s digital signature. It enforces function execution based on operator validation, ensuring compliance with document status and user identity. Only authorized entities can interact with the contract, which records all transaction outcomes for traceability purposes.

The Logical View, illustrated in Figure 4, shows the main components, including:

- **State:** Defines the unique lifecycle states of an entity.
- **StateList:** Serves as a virtual container for ledger states, minimizing transaction conflicts.
- **VdxPaper:** Represents a document in the application, extending the State class.
- **VdxPaperContract:** The primary smart contract that manages state transitions, including methods for querying history and retrieving data.

The HLF API facilitates smart contract development and provides transactional context, extending the base Contract and Context classes to integrate additional functionality, such as mapping application object IDs to composite keys in the state database.

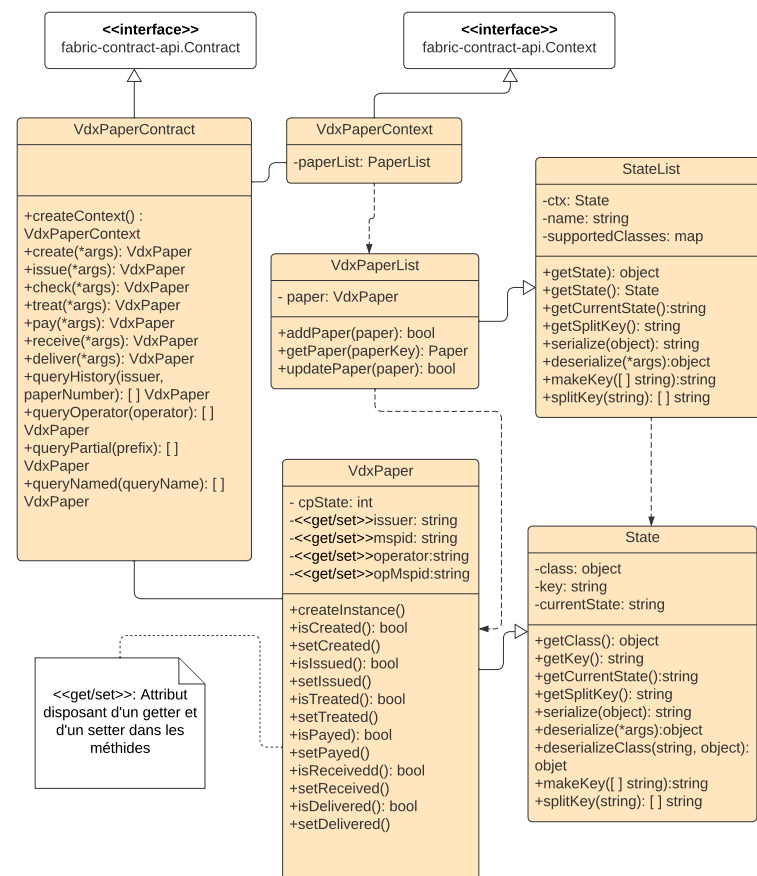


Figure 4. Logical view using the class diagram. The asterisk (*) denotes a UML multiplicity of “zero or more”.

4.2.2. Development View

This view maps the BRA-PS layers to concrete software components, showing how the Presentation, Business, Communication, and Blockchain layers are instantiated in the implementation. The development view provides a programmer’s perspective and describes the components and modules of the system. It outlines dependencies and organizes components according to their functional roles. Key components include:

- **User Interface (Web Browser):** Offers a user-friendly interface for interacting with the blockchain application.
- **API Service:** Connects the application to smart contracts, enabling citizen and organizational interactions with the blockchain.
- **Digital Wallet:** Manages digital identities on the blockchain and supports secure transactions for citizens.
- **Management System:** An organization-specific component that interfaces public blockchain processes with private workflows.
- **Blockchain (HLF):** Deploys smart contracts and stores data on public processes and document states.
- **Decentralized Storage System (IPFS):** Used for decentralized file management, storing files linked to public processes.

4.2.3. Process View

This view illustrates the runtime behavior of the public–private workflow separation central to BRA-PS, showing how citizen interactions traverse the Presentation and Communication layers while organizational processes remain within the Business layer. The

process view focuses on the dynamic behavior of the system during runtime, illustrating the interactions between actors and components for both citizens and organizations. Using activity diagrams, the workflow is as follows:

- **Citizen Workflow:** As presented in Figure 5, citizens log into the platform through a web interface and authenticate using a digital wallet. They store data on IPFS, with files encrypted prior to uploading. The IPFS metadata and file addresses are linked to blockchain requests to ensure secure and verifiable data management.
- **Organization Workflow:** As shown in Figure 6, organization users authenticate through an enterprise system. Internal management processes interface with blockchain processes via an API service, thereby allowing seamless transition between private and public workflows. Based on the public status of a document, the system triggers further actions within the organization's private system, keeping it synchronized with the blockchain.

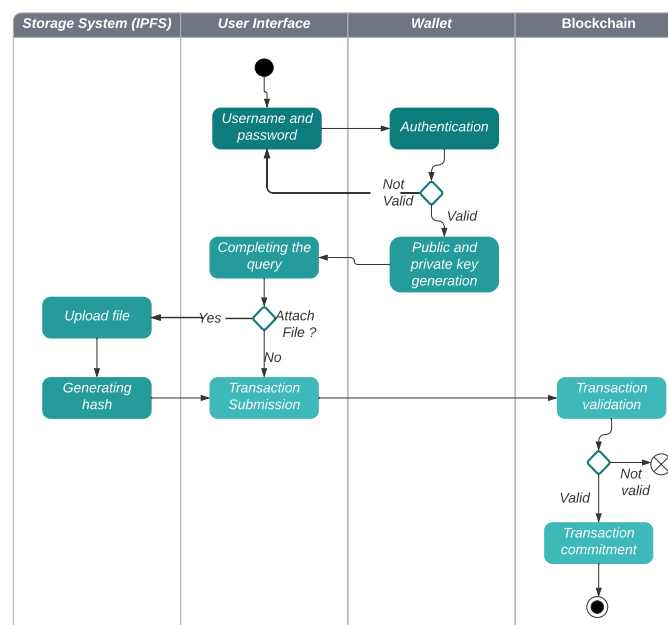


Figure 5. Activity diagram: Citizen view.

4.2.4. Physical View

This view describes how the BRA-PS Blockchain and Data layers are deployed across organizational boundaries, reflecting the architecture's decentralization principle. The physical view describes the system's deployment and infrastructure, emphasizing the connections between software and hardware components. As shown in Figure 7, the deployment setup included the following:

- **Client Device:** A laptop or workstation running a web browser to access the user interface (<https://github.com/besionisrael/BRAPS-web>) (accessed on 5 July 2026).
- **Application Server** (<https://github.com/besionisrael/BRAPS-FabricNetworkLab/tree/release-2.2/cross-process>) (accessed on 5 July 2026): Hosted on Node.js, incorporating HLF Software Development Kit (SDK) for blockchain communication, an IPFS component for decentralized storage, and an Express component for API services (<https://documenter.getpostman.com/view/3607507/2s8YsozEmas>) (accessed on 5 July 2026).
- **Blockchain Network** (<https://github.com/besionisrael/vde23-lab/tree/release-2.2/cop-network>) (accessed on 5 July 2026): Deployed in a series of Docker containers, each hosting elements like nodes, CAs, and the state database. Docker Swarm links

these containers, enabling multi-organization networks and scalable environments for public processes.

To ensure decentralization and transparency, the HLF network is deployed across multiple hosts (The full artefact configuration is available at: <https://github.com/besionisrael/BRAPS-HLFMultiHostVM>, accessed on 5 July 2026), with each organization operating its components on separate virtual machines. This setup enables each organization to manage its own resources independently while participating in the shared network.

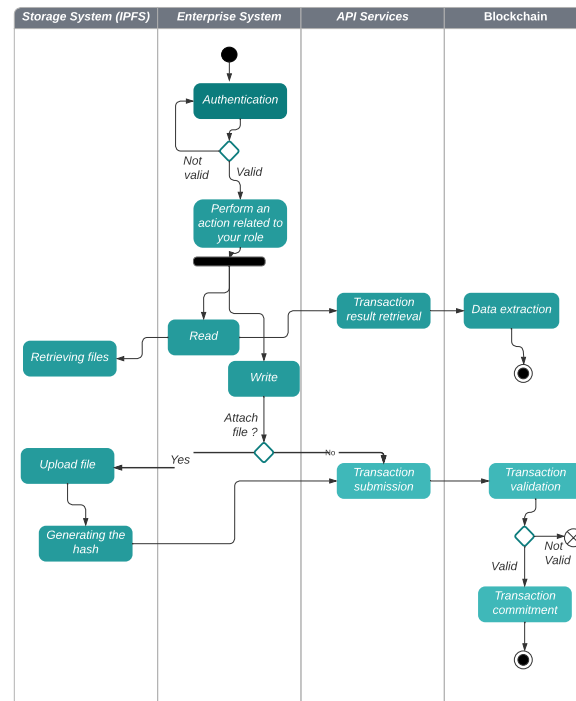


Figure 6. Activity diagram: Organization view.

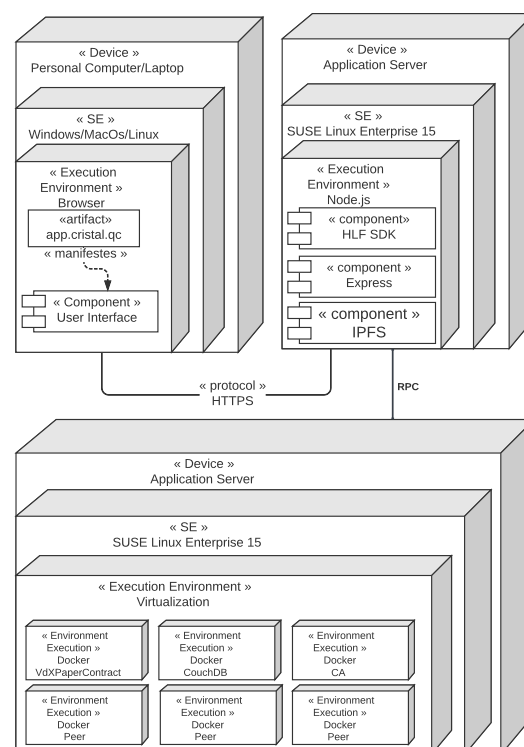


Figure 7. Physical view with deployment diagram.

5. Evaluation

After presenting the system design, we evaluate the reference architecture through an architecture-centric assessment based on ATAM. This approach is appropriate for architectural contribution papers, where the goal is to assess the soundness of design decisions and their trade-offs rather than to measure runtime performance [2,5]. Selected quantitative indicators from the proof of concept are summarized in Section 5.1, while the complete benchmark is reported in [6]. This analysis assesses architectural trade-offs to identify and balance key quality attributes such as performance, security, modifiability, and scalability.

5.1. Proof-of-Concept Performance

The proof-of-concept implementation reported in [6] was benchmarked with Hyperledger Caliper under a linear transaction rate, distinguishing read (readVdx) and write (createVdx) operations on the ledger. Table 4 summarizes the resulting indicators.

Table 4. Key Performance Indicators from the Proof-of-Concept [6].

Indicator	Observed Value
Read latency (readVdx)	≈0.1 s average, stable across transaction volumes
Read throughput (readVdx)	≈100 transactions/s average
Write behavior (createVdx)	Latency and throughput inversely related; latency becomes significant beyond 5000 transactions
Node bandwidth (peers)	≈0.7 MB/s average
Orderer bandwidth	1.519 MB/s average

Read-dominant citizen operations, such as monitoring request status, sustain low latency across the tested transaction volumes, while write operations scale acceptably up to several thousand transactions before latency degrades. The Orderer node consumes the most bandwidth, consistent with its role in maintaining network consistency and ordering across the network. These figures indicate that the architecture is viable for the read-intensive interaction patterns characteristic of citizen-facing public services. Resource-level profiling, including per-node CPU and memory consumption, falls outside the scope of this benchmark and remains an avenue for dedicated performance study.

5.2. ATAM Evaluation

The ATAM evaluation examines how well the architecture addresses its intended goals within the constraints of the case study. It also identifies areas that require further refinement and involves stakeholders directly in the evaluation process. Table 5 outlines the composition of the evaluation team, and Table 6 provides a summary of the steps undertaken in the ATAM evaluation process.

Table 5. Evaluation Group Composition.

Role	Description	Citizen Proxy
1	Information Systems Solution Architect	No
2	IT Professional experienced in blockchain applications	No
3	Project Manager and R&D Coordinator	No
4	Associate Professor and Blockchain Technology Specialist	No
5	Non-IT Specialist	Yes

Table 6. Summary of ATAM Activities.

Activity	Description
ATAM Overview	The ATAM method was introduced to the coordinator group, covering its architectural approach, analysis framework, and scenario mapping process.
Design Objectives Overview	A presentation of the design objectives was given to create a shared understanding among participants.
Architecture Presentation	The system's architecture was demonstrated using Kruchten's 4+1 model, which includes logical, development, process, and physical views, along with scenario descriptions.
Identification of Architectural Techniques	The design objectives and approaches to meet these objectives were outlined, covering: Wallet (Citizen-centric focus, Security), Channel (Privacy), World State Database (History tracking), Smart Contract (Automation, History tracking), On-chain and off-chain data (Collaboration), Integration with legacy systems (Interoperability), Blockchain (Security), Public/private key encryption (Authentication, Security), Distributed ledger (Traceability, History tracking, Collaboration), REST API (Reliability, Security).
Creation of Quality Attribute Utility Tree	Based on participant questions, a utility tree for quality attributes was developed, emphasizing priority quality requirements for specific scenarios.
Architectural Approach Analysis	This step evaluated the architectural approaches by examining the prioritized scenarios to uncover risks, pain points, and trade-offs within the proposed architecture in relation to system interactions (An evaluation survey is accessible at: https://forms.gle/4R9teZLtVoFcrank7 , accessed on 5 July 2026).
Results Presentation	The evaluation results were shared with the participants at this stage.

Direct citizen participation in the ATAM evaluation was not feasible given the technical nature of the assessment process, which requires familiarity with architectural concepts and scenario-based reasoning. However, citizen perspectives were represented indirectly through two mechanisms. First, Participant 5, a non-IT specialist with no prior blockchain experience, served as a citizen proxy, ensuring that usability and accessibility concerns were raised during scenario elicitation. Second, the ATAM scenarios developed for functional suitability (SC5 through SC12) were derived directly from citizen-facing requirements identified during the real-world project, ensuring that the evaluation remained grounded in end-user needs. Future evaluations should consider involving citizen panels through structured co-design methods to provide more direct representation.

A further limitation concerns the empirical validation of the citizen-centric claim. While the citizen-centric orientation of BRA-PS is expressed through concrete architectural decisions, the present study does not include a usability study with citizens to confirm that the Presentation Layer achieves its intended accessibility goals. No think-aloud protocol, System Usability Scale assessment, or citizen survey was conducted. Establishing that non-technical users can effectively monitor and interpret blockchain-backed processes requires dedicated usability research, which constitutes an important direction for future work and would complement the architecture-centric evaluation reported here.

The utility tree identifies, prioritizes, and refines the most important quality attribute objectives, as illustrated in Figure 8. The tree's upper-level nodes represent the key quality attributes, which in this instance are autonomy, collaboration, and functional suitability.

The tree's leaves depict scenarios for achieving these respective quality attributes. Table 7 presents the specific scenarios developed to assess these attributes.

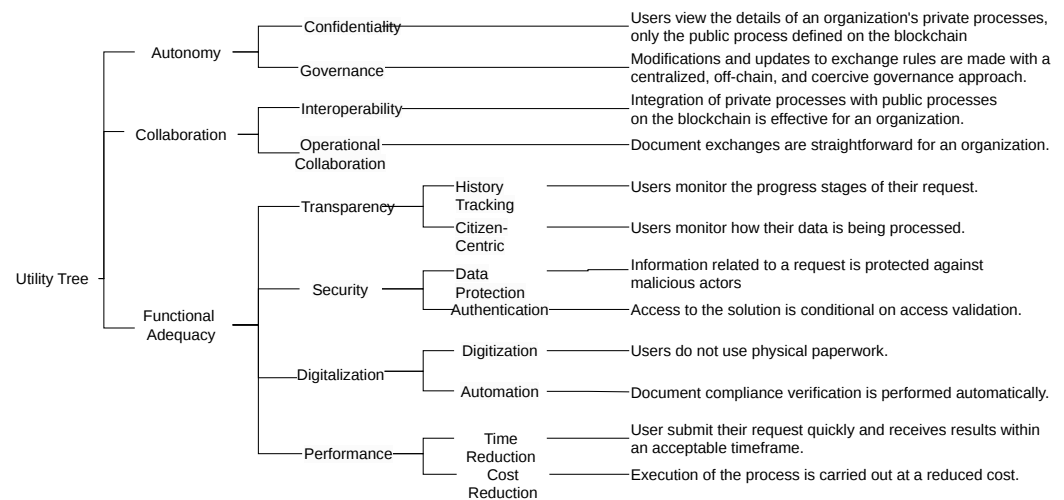


Figure 8. Utility Tree.

Scenario prioritization in this evaluation was performed through the construction of the utility tree itself, consistent with the ATAM method [5]: participants elaborated quality attribute objectives into concrete scenarios and, in doing so, surfaced those most important to the public sector context, which were retained as SC1–SC12 under the three priority attributes of autonomy, collaboration, and functional suitability. The prioritized scenarios were then analyzed individually to elicit the associated architectural approaches. Risks, non-risks, sensitivity points, and trade-offs were identified through qualitative categorization during this analysis rather than through numerical scoring, as the goal was to characterize the nature of each architectural decision relative to the quality attribute requirements. Categorization was reached by facilitator-led discussion: each candidate finding was classified only after the participants converged on its interpretation, and divergent assessments were resolved through the structured, scenario-by-scenario discussion described in Section 5.4.1.

Table 7. ATAM Attributes and Scenarios.

Attribute	Scenarios
Autonomy	SC1: Users view only public blockchain processes, not private organizational details. SC2: Exchange rule modifications are approved on-chain by an organization.
Collaboration	SC3: Private processes are effectively integrated with public blockchain processes for an organization. SC4: Document exchanges occur in a streamlined manner for an organization.
Functional Suitability	SC5: Users can monitor the progression of their requests. SC6: Users can view how their data is being handled. SC7: Request information is protected from malicious actors. SC8: Solution access requires access validation. SC9: Process is paperless. SC10: Compliance verification is automated. SC11: Users submit requests and receive responses in an acceptable time frame. SC12: Execution costs are minimized.

To make the relationship between the evaluation scenarios and the architecture explicit, Table 8 maps each scenario to the architectural components, layers, and cross-cutting concerns that address it. This traceability links the quality attribute objectives assessed in the ATAM to the concrete design decisions described in Section 3.

Table 8. Traceability Matrix: ATAM Scenarios to Architectural Elements.

Scenario	Architectural Element	Layer/Concern
SC1	Public–private workflow separation; Public Contract Interface	Communication Layer
SC2	Smart contract endorsement; governance model	Smart Contract Layer; Governance
SC3	Orchestration Module; Private Process Integration Interface	Communication Layer
SC4	Blockchain Adapter; Blockchain Client SDK	Communication Layer
SC5	User Interface Component; Presentation Logic Component	Presentation Layer
SC6	Presentation Logic Component; distributed ledger	Presentation Layer; Data Layer
SC7	File encryption; private IPFS; access control lists	Security; Access Control; Data Layer
SC8	Authentication mechanism (X.509, digital wallet)	Access Control
SC9	Digitalized document flow; off-chain storage	Data Layer
SC10	Smart contract automation (VdxPaperContract)	Smart Contract Layer
SC11	API Service; world state database (CouchDB)	Communication Layer; Data Layer
SC12	On-chain/off-chain data strategy; hashing	Data Layer

5.3. ATAM Results

The ATAM activities produced a set of risks, non-risks, sensitivity points, and trade-offs. Sensitivity and trade-off points refer to architectural decisions that significantly affect one or more quality attributes. In ATAM, a risk is characterized as an architectural choice that could potentially result in unfavorable outcomes based on the specified quality attribute requirements. Conversely, a non-risk is identified as an architectural decision that, following careful examination, is determined to be safe. Tables 9 and 10 show the results.

Table 9. Risks and Non-Risks in the Proposed Architecture.

Risk	Non-Risk
IPFS is a public network, typically used with a public blockchain. Traffic and content are public, posing a risk to confidentiality, even though data can be encrypted.	Use of blockchain is a non-risk for autonomy, as it only executes public processes, separate from organizations' private processes.

Table 9. Cont.

Risk	Non-Risk
• Use of an API service poses a security risk if the service can execute unintended commands. • Integration points risk data transformation issues if data exchange between the blockchain and existing systems is not standardized. • A single global channel risks scalability issues if organizations outside the primary use case join the channel. • The approval policy, requiring all channel participants to endorse transactions, risks scalability and performance as more participants join. • Implementing decision logic in the smart contract risks inflexibility if frequent updates are required.	• Authentication with the X.509 certificate managed by HLF's default CA is a non-risk for authentication. • The automated operation of various architecture components is a non-risk for minimal human intervention. • Public-private separation in design enhances flexibility and poses no risk. • Lack of an access control list is not a risk if communication layer checks are performed. • HLF's proposed security mechanisms pose no risk if a security analysis relevant to the use case is conducted.

Table 10. Sensitivity and Trade-Off Points in the Proposed Architecture.

Sensitivity Point	Trade-Off
• A private blockchain ensures security, transparency, and collaboration, enabling a citizen-centered approach. • Off-chain storage enhances collaboration. • The API service connecting with the smart contract ensures organization autonomy and secure private-public system communication. • CouchDB as the state database impacts performance by supporting rich queries and enhancing security for compliance and data protection. • Multi-host deployment ensures decentralization and transparency. • HLF v2 allows governance mechanisms that enable organizations to agree on update procedures.	• Blockchain operations can limit organizational autonomy. • IPFS as an external network adds integration workload. • Direct citizen participation is a trade-off between transparency and performance. • Encapsulating decision logic within the smart contract code eases management but limits update flexibility. • Using a channel enforces data transaction confidentiality but requires additional collaboration effort. • Embedding decision logic in the smart contract reduces overall project costs but limits flexible governance.

5.4. Threats to Validity

Although the ATAM evaluation provides important insights into the proposed reference architecture, several factors may affect the validity of the results. We address these threats to validity in two categories: internal and external.

5.4.1. Internal Validity

Internal validity concerns whether the evaluation accurately reflects the assessed quality attributes. Two potential threats were identified and addressed. First, the composition of the evaluation team was carefully considered. The five participants were selected to represent distinct professional perspectives: a solution architect, a blockchain technology specialist, a project manager, an R&D coordinator, and a non-IT specialist. This

disciplinary diversity was intentional, as participants with different backgrounds are less likely to converge on identical assessments of architectural trade-offs. Second, the risk of groupthink was mitigated through the ATAM session structure itself. Scenarios were presented individually to each participant before group discussion, allowing independent judgments to be formed prior to any collective deliberation. The ATAM method's utility tree construction further structured the elicitation process by requiring participants to independently prioritize quality attributes before reconciling differences, a procedural safeguard that surfaces disagreement rather than suppressing it [5]. The facilitator, who was not directly involved in the architecture's design, ensured that dominant voices did not disproportionately influence the outcome. Despite these precautions, the small group size remains the principal limitation of the evaluation. Architecture evaluations of this kind benefit from broader stakeholder coverage, and the set of risks, non-risks, sensitivity points, and trade-offs reported here should be read as those surfaced by a focused expert panel rather than an exhaustive enumeration. We therefore recommend that future evaluations replicate the ATAM process with a larger and more diverse panel, including multiple citizen representatives engaged through structured co-design rather than a single proxy participant, to consolidate and extend the present findings.

5.4.2. External Validity

External validity concerns the extent to which the evaluation findings can be generalized to other contexts. This evaluation focused on a use case involving two public agencies and citizens interacting through a single communication channel. While this approach provides valuable insights, it may not fully capture the complexity and diversity of multi-channel environments in which citizens and agencies engage through various platforms and communication methods. Expanding future evaluations to include multi-channel use cases would offer a more comprehensive understanding of the adaptability and effectiveness of the architecture across different public sector contexts. Additionally, the evaluation does not look at cases where different public organizations each have their own blockchain but need to work together. This is likely to happen as governments start using decentralized identity solutions [31].

A further consideration concerns the durability of the findings. Blockchain platforms evolve rapidly, and platform-specific results, such as those tied to a particular HLF version, may age as the underlying technology changes. We mitigate this by framing BRA-PS at the level of architectural structure and design decisions, which are comparatively stable, rather than at the level of platform-specific configuration; the layered decomposition, the public-private workflow separation, and the cross-cutting concerns are independent of any single platform release. Nonetheless, the concrete mechanisms instantiating these decisions should be revisited as platforms mature, and periodic re-evaluation is advisable for long-lived deployments.

6. Discussion and Conclusions

6.1. Limitations

Despite the contributions of this study, several limitations should be acknowledged. Regarding internal validity, as discussed in Section 5.4.1, the five-participant panel limits the diversity of perspectives, and a larger, more diverse panel would strengthen the generalizability of the findings. Regarding external validity, the case study focused on a single use case involving two public agencies and citizens interacting through a single communication channel in the context of Quebec, Canada. This scope does not fully capture the complexity of multi-channel environments where citizens and agencies engage through various platforms, nor does it address scenarios involving multiple independent blockchains that must

interoperate, which is increasingly relevant as governments adopt decentralized identity solutions [31]. Regarding scope, this paper makes an architectural contribution and deliberately does not report runtime performance benchmarks; quantitative empirical results are reported in the companion paper [6]. Finally, citizen participation in the evaluation was indirect, mediated through a non-IT specialist proxy and citizen-derived scenarios, rather than through direct involvement of citizen panels. These limitations define the boundaries of the current study and motivate the future work directions discussed at the end of this section.

The evaluation outcomes align with and extend findings from prior blockchain adoption studies in public administration. The tension we observe between organizational autonomy and on-chain coordination echoes the coordination challenges repeatedly reported in inter-organizational public processes [27], while our finding that transparency must be deliberately engineered into the presentation layer supports the public value argument that trust is an outcome of design rather than an automatic property of the technology [25]. Practically, the architecture's explicit public-private separation offers public agencies a concrete mechanism to participate in shared services without surrendering control of internal processes, addressing a barrier frequently cited as an obstacle to blockchain adoption in the public sector.

6.2. Recommendations

Drawing from the ATAM results, a set of recommendations has been developed to improve the application of the proposed architecture in achieving the design objectives of this research. The identified risks serve as a basis for creating a risk mitigation plan. These recommendations target specific architectural components and can act as best practice guidelines for implementing the architecture.

- Perform a security analysis of HLF, outlining potential threats across different layers of the case study to assess how HLF's integrated functionalities can mitigate these risks.
- Implement a centralized authentication process to establish uniformity across all public services in case this architecture is replicated across various use cases.
- Design a "multi-channel" setup to enable multiple use cases involving organizations on the same network, facilitating private and confidential transactions.
- Deploy the HLF network across multiple hosts to ensure decentralization and transparency, with each organization deploying its own virtual machines to organize its components.
- Standardize exchange formats and establish inter-service standards between the blockchain and organizations' private management systems to extend implementation within services with similar structures.
- Externalize decision logic to an external rule engine rather than embedding it in the smart contract code, providing organizations with the flexibility to modify decision logic without changing the smart contract code.
- Utilize a decentralized storage solution with a private IPFS network interfaced with an HLF client for data exchange.
- Hash data before calling a smart contract and establish a mechanism to share source data.
- Implement access control lists to restrict data access to specific roles within the smart contract logic.

6.3. Practitioner Guide: Adopting BRA-PS

The following step-by-step guide is intended to support practitioners in applying BRA-PS to a new public sector blockchain initiative.

- (1) **Define the interaction context.** Using the Interaction Context cross-cutting concern, identify all participating organizations and their roles, determining which operate a peer node and which access the network through a presentation-layer interface hosted by another member. In HLF, these roles are formalized through Membership Service Provider (MSP) configuration. This step sets the governance perimeter and the access privileges that constrain every subsequent step.
- (2) **Establish the governance framework.** Apply the centralized, off-chain, coercive governance model recommended by BRA-PS: assign governance decisions to a defined group of organizations, conduct major decisions off-chain to reduce on-chain overhead, and encode agreed rules directly into smart contracts. In Fabric, these agreements are captured in channel configuration and endorsement policies. Cover membership, change management, dispute resolution, and network regulation before any technical implementation begins.
- (3) **Map public and private workflows.** Using the public–private separation enforced by the Communication Layer, classify each step of every inter-organizational process as public (executed on-chain through the Public Contract Interface) or private (handled internally and exposed only through the Private Process Integration Interface). Table 11 illustrates this classification for the vehicle registration case and serves as a reusable mapping worksheet. The result is the direct input to the smart contract design.
- (4) **Design the smart contracts.** From the public workflow map, define the state transitions enforced by the Smart Contract Layer, specifying for each transition the originating state, the triggering event, the resulting state, and the endorsing organization, following the VdxPaperContract state model. Smart contracts are implemented as chaincode (for example, in JavaScript or Go); where frequent updates to decision logic are anticipated, an external rule engine externalizes that logic to avoid chaincode redeployment.
- (5) **Configure the data strategy.** Apply the hybrid on-chain/off-chain strategy of the Data Layer: keep document states and transaction records on-chain in the world-state database (CouchDB, for rich queries), and store large or sensitive files off-chain. When using IPFS, deploy a private network within the organizational perimeter, encrypt all files prior to upload, and record only content identifiers (hashes) on the ledger.
- (6) **Implement the Communication Layer.** Develop the Blockchain Adapter and the Orchestration Module to synchronize public blockchain events with each organization's private management system, typically through the Fabric SDK and a REST API service (for example, Express on Node.js). Ensure the public state is advanced only after the corresponding private process completes, following the error-handling logic of Section 3.4, and standardize data exchange formats between the blockchain and legacy systems at this stage.
- (7) **Develop the Presentation Layer.** Build citizen-facing interfaces through the User Interface and Presentation Logic components, typically as a web client consuming the API service, exposing public blockchain states in accessible, non-technical language. Ensure citizens can monitor the progression of their requests and understand how their data is handled.
- (8) **Conduct an ATAM evaluation.** Before production deployment, convene a multi-disciplinary panel including at least one non-technical participant as a citizen proxy. Construct a utility tree over the priority quality attributes (autonomy, collaboration, functional suitability) and use scenario-based analysis to identify risks, sensitivity points, and trade-offs, benchmarking throughput and latency with a tool such as Hyperledger Caliper to inform the performance-related scenarios.
- (9) **Deploy and monitor.** Following the Blockchain Layer deployment model, distribute the HLF network across multiple hosts with Docker Swarm, each organization man-

aging its own infrastructure on separate virtual machines. Apply the Security and Monitoring cross-cutting concerns: implement access control lists, real-time activity logging, and anomaly detection through Fabric’s operations service and Prometheus-compatible endpoints to maintain network health and security post-deployment.

Table 11. Workflow Mapping Template (Vehicle Registration Instance).

Process Step	Class	BRA-PS Element
Tax calculation (Org2)	Private	Business Workflow Component
Certificate issuance	Public	Public Contract Interface; VdxPaperContract
Document validation (Org1)	Private	Private Process Integration Interface
Tax collection/state update	Public	VdxPaperContract
VDX issuance	Public	Public Contract Interface

6.4. Conclusions

This study presents a reference architecture for blockchain integration in public sector services, designed to enhance transparency and interoperability while maintaining organizational autonomy and data privacy. Through a layered architecture and cross-cutting concerns, we provide a reference architecture that addresses key requirements essential for citizen-centric public sector applications. Our case study on vehicle registration between two public agencies demonstrates the practical viability of the architecture, whereas an ATAM-based evaluation input highlights its strengths and areas for refinement.

The results show that the architecture can guide implementation decisions and support stakeholder alignment. Moving forward, this reference architecture can be adapted and extended to other public sector scenarios, assisting institutions in navigating blockchain adoption, fostering inter-organizational collaboration, and supporting citizen-centric solutions. Expanding future evaluations to include both multi-channel communication scenarios and multi-chain management would offer a more comprehensive understanding of the adaptability and effectiveness of the architecture across different public sector contexts.

Author Contributions: Conceptualization, S.I.S., K.Z. and A.A.; methodology, S.I.S.; software, S.I.S.; validation, K.Z. and A.A.; formal analysis, S.I.S.; investigation, S.I.S. and K.Z.; resources, K.Z. and A.A.; data curation, S.I.S.; writing—original draft preparation, S.I.S.; writing—review and editing, S.I.S., K.Z. and A.A.; visualization, S.I.S.; supervision, K.Z. and A.A.; project administration, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The source code and artefact configurations supporting the reported results are publicly available on GitHub via the repositories referenced in Section 4.2. No new datasets were generated during this study.

Acknowledgments: This article is a revised and expanded version of a paper entitled “Towards Citizen-Centric Services using Blockchain-Powered Digitalization of Public Sector Processes,” which was presented at the 2023 Fifth International Conference on Blockchain Computing and Applications (BCCA), Kuwait, Kuwait, October 2023. The authors used Paperpal (web version) to improve syntax and grammar.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Commission, E.; Centre, J.R.; Vaccari, L.; Pignatelli, F.; Allesie, D.; Sobolewski, M. *Blockchain for Digital Government—An Assessment of Pioneering Implementations in Public Services*; Publications Office of the European Union: Luxembourg, 2019. [CrossRef] [PubMed]
2. Bass, L.; Clements, P.; Kazman, R. *Software Architecture in Practice*, 4th ed.; Addison-Wesley Professional: Boston, MA, USA, 2021.
3. Cervantes, H.; Kazman, R. *Designing Software Architectures: A Practical Approach*; Addison-Wesley Professional: Boston, MA, USA, 2016.

4. Kruchten, P. The 4+1 View Model of architecture. *IEEE Softw.* **1995**, *12*, 42–50. [\[CrossRef\]](#)
5. Kazman, R.; Klein, M.; Clements, P. *ATAM: Method for Architecture Evaluation*; Carnegie Mellon University: Pittsburgh, PA, USA, 2000; Number CMU/SEI-2000-TR-004.
6. Sion, S.I.; Zhang, K.; April, A. Towards Citizen-Centric Services using Blockchain-Powered Digitalization of Public Sector Processes. In *2023 Fifth International Conference on Blockchain Computing and Applications (BCCA)*; IEEE: New York, NY, USA, 2023; pp. 361–368. [\[CrossRef\]](#)
7. Viswanathan, R.; Dasgupta, D.; Govindaswamy, S.R. Blockchain Solution Reference Architecture (BSRA). *IBM J. Res. Dev.* **2019**, *63*, 1:1–1:12. [\[CrossRef\]](#)
8. Liu, Y.; Lu, Q.; Yu, G.; Paik, H.Y.; Zhu, L. A Pattern-Oriented Reference Architecture for Governance-Driven Blockchain Systems. In *2023 IEEE 20th International Conference on Software Architecture (ICSA)*; IEEE: New York, NY, USA, 2023; pp. 23–34. [\[CrossRef\]](#)
9. Wang, Y.; Li, S.; Liu, H.; Zhang, H.; Pan, B. A Reference Architecture for Blockchain-based Traceability Systems Using Domain-Driven Design and Microservices. *arXiv* **2023**, arXiv:2302.06184. [\[CrossRef\]](#)
10. Gong, Y.; van Engelenburg, S.; Janssen, M. A Reference Architecture for Blockchain-Based Crowdsourcing Platforms. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 937–958. [\[CrossRef\]](#)
11. Weber, T.; Buchkremer, R. Blockchain-Based Reference Architecture for Automated, Transparent, and Notarized Attestation of Compliance Adaptations. *Appl. Sci.* **2022**, *12*, 4531. [\[CrossRef\]](#)
12. Aviv, I.; Barger, A.; Kofman, A.; Weisfeld, R. Reference Architecture for Blockchain-Native Distributed Information System. *IEEE Access* **2023**, *11*, 4838–4851. [\[CrossRef\]](#)
13. Gohar, A.N.; Abdelmawgoud, S.A.; Farhan, M.S. A Patient-Centric Healthcare Framework Reference Architecture for Better Semantic Interoperability Based on Blockchain, Cloud, and IoT. *IEEE Access* **2022**, *10*, 92137–92157. [\[CrossRef\]](#)
14. Josey, A. *TOGAF® Version 9.1-A Pocket Guide*; Van Haren: Amsterdam, The Netherlands, 2016.
15. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; Caro, A.D.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *EuroSys '18: Proceedings of the Thirteenth EuroSys Conference*; Association for Computing Machinery: New York, NY, USA, 2018. [\[CrossRef\]](#)
16. Gupta, M. *Blockchain for the Enterprise: The Definitive Guide for Enterprise Blockchain Adoption*; Manav Gupta: Toronto, ON, Canada, 2018; ISBN 978-1999387105. 2018.
17. Negara, E.S.; Hidyanto, A.N.; Andryani, R.; Erlansyah, D. A Survey Blockchain and Smart Contract Technology in Government Agencies. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1071*, 012026. [\[CrossRef\]](#)
18. Tapscott, D.; Tapscott, A. How Blockchain Will Change Organizations. *MIT Sloan Manag. Rev.* **2017**, *58*, 10–13.
19. Mendling, J.; Weber, I.; Aalst, W.V.D.; Brocke, J.V.; Cabanillas, C.; Daniel, F.; Debois, S.; Ciccio, C.D.; Dumas, M.; Dustdar, S.; et al. Blockchains for Business Process Management-Challenges and Opportunities. *ACM Trans. Manag. Inf. Syst.* **2018**, *9*, 1–16. [\[CrossRef\]](#)
20. Seebacher, S.; Maleshkova, M. A Model-driven Approach for the Description of Blockchain Business Networks. In *Proceedings of the Hawaii International Conference on System Sciences*, Hilton Waikoloa Village, HI, USA, 3–6 January 2018.
21. Eggers, W.D. Taking a Citizen-Centric Approach to State Government. *Deloitte CMO J.* **2018**. Available online: <https://deloitte.wsj.com/cmo/taking-a-citizen-centric-approach-to-state-government-1521000134> (accessed on 5 July 2026).
22. Bodkhe, U.; Tanwar, S.; Parekh, K.; Khanpara, P.; Tyagi, S.; Kumar, N.; Alazab, M. Blockchain for Industry 4.0: A Comprehensive Review. *IEEE Access* **2020**, *8*, 79764–79800. [\[CrossRef\]](#)
23. Moore, M.H. *Creating Public Value: Strategic Management in Government*; Harvard University Press: Cambridge, MA, USA, 1995.
24. Lindgren, I.; Jansson, G. Electronic services in the public sector: A conceptual framework. *Gov. Inf. Q.* **2013**, *30*, 163–172. [\[CrossRef\]](#)
25. Twizeyimana, J.D.; Andersson, A. The public value of E-Government—A literature review. *Gov. Inf. Q.* **2019**, *36*, 167–178. [\[CrossRef\]](#)
26. Valle, P.H.D.; Garcés, L.; Volpato, T.; Martínez-Fernández, S.; Nakagawa, E.Y. Towards suitable description of reference architectures. *PeerJ Comput. Sci.* **2021**, *7*, e392. [\[CrossRef\]](#)
27. Peristeras, V.; Tarabanis, K.; Goudos, S.K. Model-driven eGovernment interoperability: A review of the state of the art. *Comput. Stand. Interfaces* **2009**, *31*, 613–628. [\[CrossRef\]](#)
28. Tan, E.; Mahula, S.; Cromptvoets, J. Blockchain governance in the public sector: A conceptual framework for public management. *Gov. Inf. Q.* **2022**, *39*, 101625. [\[CrossRef\]](#)
29. Shostack, A. *Threat Modeling: Designing for Security*; Wiley: Hoboken, NJ, USA, 2014.
30. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75–105. [\[CrossRef\]](#)
31. Reece, M.; Mittal, S. Self-Sovereign Identity in a World of Authentication: Architecture and Domain Usecases. *arXiv* **2022**, arXiv:2209.11647. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.