Orchestration and Decentralized Governance in Permissioned Blockchains: A Systematic Mapping of Operational Approaches

Matheus Lázaro Honório da Silva[1], Gislainy Velasco[1], Eliomar Araújo de Lima[1], Valdemar Vicente Graciano Neto[1], Sergio T. Carvalho[1]

[1] Instituto de Informática – Universidade Federal de Goiás (UFG)

{matheus.lazaro, gislainycrisostomo}@discente.ufg.br

{valdemarneto, eliomar.lima, sergiocarvalho}@ufg.br

Abstract. *Permissioned blockchains support interorganizational processes in regulated environments and require governance, identity and access management, infrastructure provisioning, and lifecycle automation. This study presents a systematic mapping of orchestration, provisioning, and decentralized governance in permissioned blockchains. The protocol follows PICOC and PRISMA and defines research questions, a three-block search string, selection criteria, and a quality assessment checklist. The study examines 158 records from seven sources and selects 23 primary studies. It classifies the evidence into operational categories and synthesizes the results by research question. The results show coverage of architectures,* stacks*, and operational procedures, but reveal gaps in governance, identity and access management, and end-to-end lifecycle automation. Hyperledger Fabric appears as the most frequently used platform, although the study does not use it as a search filter. The findings indicate that the main challenges lie in control plane design and in coordinating critical changes across organizations. The study derives design implications for open-source,* stack*-agnostic orchestration solutions.*

1. Introduction

Permissioned blockchains support interorganizational processes, particularly in regulated environments. In contrast to public blockchains, these systems require solutions for multi-organizational governance, identity and access management, infrastructure provisioning, and automation of operations such as organization onboarding and offboarding, channel configuration, and smart contract lifecycle management.

In practice, decentralization requires distributed transaction execution and coordinated, auditable operational procedures across organizations. Current platforms provide mechanisms for lifecycle and governance, but operational flows such as parameter sharing, approvals, execution windows, evidence recording, and *rollback* still rely on manual procedures and remain error-prone. This condition indicates a mismatch between the decentralization model and the requirements of operating blockchain consortia.

Existing studies address architectural aspects and platform configurations for permissioned blockchains [Sato et al. 2021, Sato et al. 2022, Tran et al. 2022, Mathwale 2023, Yu et al. 2024, Silva et al. 2025]. However, they do not provide a systematic view of how practice integrates orchestration, automation, and decentralized gov-

ernance. The literature also treats orchestration mechanisms and governance processes in a fragmented way, which constrains consortium operation.

This study presents a systematic mapping that aims to identify solutions and architectures for orchestration and operation of permissioned blockchains, characterize approaches to decentralized governance, identity and access management, and smart contract lifecycle, and synthesize research gaps to derive design principles for open-source, *stack*-agnostic orchestration solutions.

The paper is organized as follows. Section 2 describes the protocol (PICOC[1]), research questions, search string, sources, selection process, and quality assessment. Section 3 reports the PRISMA-based execution. Section 4 presents the results and discussion, including the synthesis by research question. Section 5 concludes the paper.

2. Mapping methodology

We conducted a systematic mapping focused on operational aspects of permissioned blockchains. The protocol was registered in Parsifal[2] and included the PICOC framework, research questions, the complete search string, selection criteria, and a quality checklist. The protocol adopts a platform-agnostic design and does not restrict the search to any specific stack. This decision ensures that the prevalence of particular platforms in the synthesis reflects the retrieved literature rather than an *a priori* filter. To support reproducibility and auditability, the study makes the protocol and execution artifacts available in a Zenodo repository DOI: https://doi.org/10.5281/zenodo.18616087, including source-specific search strings, exports, PRISMA counts, the quality matrix, and synthesis artifacts.

The study follows five objectives. It maps architectures, patterns, and orchestration and automation tools for permissioned blockchains. The analysis examines how solutions address multi-organizational governance, identity and access control, and smart contract lifecycle automation. It also compares provisioning and automation strategies and their implications for portability. In addition, the study synthesizes evidence on observability, resilience, security, and operational impact and identifies implementation gaps. Finally, it derives design implications for open-source, *stack*-agnostic orchestration solutions from recurring patterns and identified gaps.

PICOC. P (Population): permissioned blockchains such as Hyperledger Fabric, Corda, Quorum, and Besu, and their operational components, including nodes, channels, certificate authorities, policies, and *pipelines*. *I (Intervention):* orchestration and operational management solutions, including operators, controllers, management platforms, lifecycle automation, DevOps, and Infrastructure as Code (IaC), observability, and resilience. *C (Context):* multi-organizational consortia, production environments, and industrial and academic proofs of concept, including regulated sectors.

Research questions. We structured the protocol around three main questions and supporting subquestions:

RQ1 What solutions exist for orchestration and operational management of permis-

[1]Population, Intervention, Comparison, Outcome, and Context. In systematic mappings, the *Comparison* and *Outcome* elements are often less central because the objective is to map and characterize the literature. For this reason, these elements were not considered.

[2]Parsifal is a web tool for planning and conducting systematic reviews and mappings.

sioned blockchains?

RQ1a: How do these solutions implement provisioning and automation? *RQ1b:* Which governance, identity, and access mechanisms do these solutions support?

RQ2 Which patterns and architectural approaches support provisioning, decentralized governance, identity and access management, and smart contract lifecycle automation?

RQ2a: How do solutions automate *deploy*, *upgrade*, and *rollback*? *RQ2b:* How do APIs and *gateways* support discovery, routing, *caching*, and security?

RQ3 Which gaps and design implications emerge for open-source, *stack*-agnostic orchestration solutions?

RQ3a: Which requirements for observability, resilience, security, and compliance remain open? *RQ3b:* Which trade-offs arise from infrastructure and automation choices?

2.1. Keywords and search string

We refined keywords and synonyms iteratively and organized them into three blocks. The first block covers permissioned blockchains and related terms such as *consortium*, *enterprise*, *private blockchain*, and *permissioned DLT*. The second block covers orchestration and operations, including provisioning, automated *deployment*, *workflow* automation, and MANO[3]. The third block covers governance in multi-actor environments, including multi-organizational, multi-domain, and *multi-tenant* settings, as well as policies, *compliance*, separation of privileges, and controlled information sharing.

The final search string combines three blocks to capture studies that address governance, orchestration, and provisioning in permissioned blockchains without platform bias. The study covers publications from 2020 onward in Portuguese or English and searches the ACM Digital Library, IEEE Xplore, Engineering Village, ScienceDirect, Scopus, BDTD, and SBC OpenLib. It adapts the query syntax to each source while preserving the terms and logical structure. Figure 1 summarizes the three-block strategy. The complete string and its source-specific adaptations are available in the Zenodo repository DOI: https://doi.org/10.5281/zenodo.18616087.

Three-block search strategy

Block 1 - Permissioned blockchains: permissioned blockchain; consortium blockchain; enterprise blockchain; private blockchain; permissioned DLT. **AND**

Block 2 - Orchestration and operations: orchestration; operations; provisioning; deployment; workflow automation; management and orchestration (MANO); DevOps; Infrastructure as Code (IaC). **AND**

Block 3 - Governance and multi-organizational context: governance; multi-organization; multi-party; policy; compliance; identity and access management; privilege separation; controlled information sharing.

Figure 1. Three-block search strategy.

[3]MANO (*Management and Orchestration*) is a term from Networking, particularly in NFV-MANO. In this study, the term is used as a conceptual reference to orchestration and does not imply direct transfer of NFV-MANO models to permissioned blockchain consortia. In this context, the control plane involves multiple organizations, each with operational autonomy, and decisions depend on governance elements such as approvals, roles, evidence, and auditability.

2.2. Selection criteria

Inclusion criterion. The study includes primary studies, academic or industrial, that are peer-reviewed or indexed as theses or dissertations in BDTD, published from 2020 onward in Portuguese or English, with accessible full text, and that propose a concrete solution for management or operational orchestration of a permissioned blockchain, including network, consortium, channels, or infrastructure. As minimum evidence, the study must present an explicit architecture or operational *workflow*, or sufficient detail for replication, such as scripts, IaC, or configurations, even if only partially.

Exclusion criteria. The study excludes works without peer review, when applicable, that are not theses or dissertations or do not present evidence of implementation. It also excludes out-of-scope work focused solely on applications or contracts, and that does not address orchestration, operations, or governance. The study excludes public DLTs without a permissioned counterpart and purely theoretical or cryptographic studies without operational applicability. It further excludes duplicates or superseded versions, unavailable full texts, publications available only through the CAPES Journal Portal, publications before 2020, and publications in languages other than Portuguese or English. The study also excludes secondary studies such as *surveys*, SLRs, or mappings.

2.3. Quality assessment and synthesis

The study applies a quality assessment (QA) checklist consisting of 10 questions (QA1–QA10). The checklist covers operational focus, platform specification, architectural description, explicit *workflow* definition, reproducibility and artifacts, multi-organizational governance, identity and access management, *lifecycle* automation, reliability, security, compliance, and empirical evidence. Each question is rated as *Yes* (1.0), *Partial* (0.5), or *No* (0). The aggregated score supports synthesis prioritization.

The quality checklist (QA) consists of ten questions. QA1 verifies whether the study addresses the orchestration and operational management of a permissioned blockchain, including the network, consortium, channel, and infrastructure, rather than treating the blockchain solely as an auxiliary trust mechanism. QA2 verifies whether the study explicitly defines the permissioned platform or *stack*, such as Fabric, Corda, Besu, or Quorum, or provides sufficient detail about members and assumptions in the system description. QA3 examines whether the solution architecture is described, including components, interfaces, and responsibilities. QA4 verifies whether the operational procedure is explicit, including steps, *workflow*, automation, prerequisites, and inputs and outputs. QA5 evaluates whether the study provides an implementation or prototype, or sufficient detail for replication, such as scripts, IaC, or configuration artifacts. QA6 examines whether the study addresses multi-organizational governance with concrete elements, including consortium, channels, policies, *onboarding* and *offboarding*, roles, and decision processes. QA7 assesses whether the study implements identity and access control through concrete mechanisms, including certificate authorities, permission management, certificate lifecycle management, rotation or revocation, and integrations. QA8 verifies whether the study covers contract *lifecycle* automation, including *deploy*, *upgrade*, *rollback*, versioning, and policies. QA9 analyzes whether the study addresses operational reliability, including observability, resilience, security, compliance, disaster recovery, and auditability. QA10 verifies whether the study presents evidence and evaluation, including

metrics, experiments, case studies, comparisons, and a discussion of threats and limitations.

Unlike reviews that apply a rigid extraction form to all fields and studies, this mapping adopts a guided extraction approach. The process consolidates the contribution of each study through *narrative synthesis summaries* derived from full-text analysis and guided by the research questions (RQ1–RQ3) and the quality checklist (QA1–QA10). For each study, the analysis records a summary of the problem and contribution, the main operational elements including architecture, *workflow*, automation, artifacts, and evidence, and the QA score with identified gaps. This approach supports consistent comparison and categorization into operational categories, even without a fully uniform extraction structure across all fields.

The Zenodo repository DOI: https://doi.org/10.5281/zenodo.18616087, provides the complete protocol, source-specific search strings, result exports, PRISMA counts, the quality matrix with scores and justifications, and synthesis artifacts. These materials support auditability and replication.

3. Execution and selection (PRISMA)

The study follows the PRISMA 2020 phases, including identification, screening, eligibility, and inclusion. Parsifal supports protocol organization and decision recording. The search retrieves 158 records across seven sources. After duplicate removal (n=27), the process screens 131 records based on title and abstract. This stage excludes 108 records. The analysis then assesses 23 full texts and includes 23 primary studies in the qualitative synthesis.

Table 1. Selection stages according to PRISMA 2020 (absolute counts).

Stage	*Records (n)*	*Description / notes*
Identified	158	Parsifal import (7 sources)
ACM DL	64	-
BDTD	4	-
Engineering Village	16	-
IEEE Xplore	27	-
SBC OpenLib	0	-
ScienceDirect	29	-
Scopus	18	-
Duplicates removed	27	Automatic duplicate selection via Parsifal
Records screened (title and abstract)	131	Manual analysis
Records excluded	108	Out of scope, document type, or failure to meet the inclusion and exclusion criteria
Full texts assessed	23	-
Full texts excluded (eligibility)	0	Quality assessment
Studies included	23	Qualitative synthesis

Beyond the aggregated PRISMA counts, the distributions by search source and year reveal important characteristics of the corpus. At the identification stage, computing repositories concentrate most records. The ACM Digital Library contributes 64 of 158 records (40.5%), followed by ScienceDirect with 29 (18.4%), IEEE Xplore with 27 (17.1%), Scopus with 18 (11.4%), and Engineering Village with 16 (10.1%). BDTD contributes 4 records (2.5%), while SBC OpenLib returns no records.

Comparing retrieved records with the final inclusion shows a heterogeneous funnel. The 23 included studies concentrate in ACM DL (n=9), Scopus (n=5), and IEEE

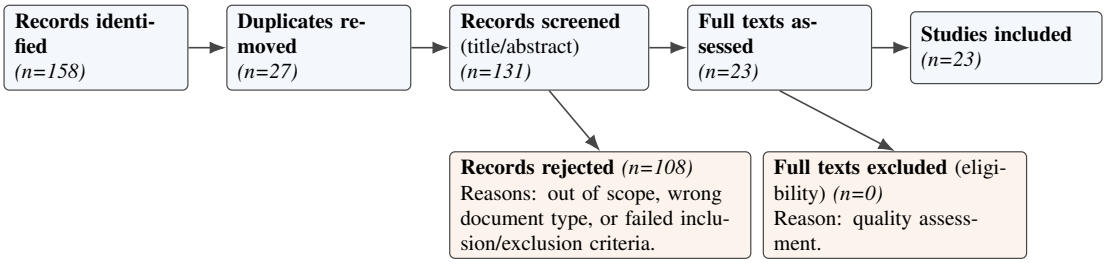

Figure 2. PRISMA flow diagram of the selection process.

Xplore (n=4), with Engineering Village contributing 3 studies and ScienceDirect and BDTD contributing 1 study each. These differences indicate variation in precision across sources. For example, Scopus reports an inclusion rate of approximately 28% (5/18), whereas ScienceDirect reports about 3% (1/29).

The temporal distribution shows an increase in publications up to a peak in 2023, followed by a variation in subsequent years. The decline in the most recent year may reflect a partial collection window. This pattern reinforces the need for multi-source searches and clear selection criteria to control false positives.

Figure 2 shows the PRISMA flow diagram and summarizes the corpus selection process. The search identifies 158 records across seven sources. After removing 27 duplicates, the study screens 131 records based on title and abstract. At this stage, it excludes 108 records that do not meet the inclusion and exclusion criteria, for example, because of scope mismatch or document type. The analysis then assesses 23 full texts. No additional exclusions occur at the eligibility stage (n=0), so the study includes the same 23 texts in the qualitative synthesis. The diagram makes the selection *funnel* and the distribution of losses across stages explicit, thereby supporting transparency, traceability, and reproducibility.

4. Results and discussion

4.1. Included studies and quality

Table 2 summarizes the 23 included studies and the quality checklist (QA1–QA10) according to the protocol. The mean score is 6.33 out of 10, and the median is 6.5, which indicates variation in operational maturity. The studies show coverage of the platform and *stack* specifications, with 19 out of 23 rated *Yes* in QA2. They also describe architecture and procedures, with 16 studies rated as *Yes* in QA3 and 15 in QA4. In contrast, multi-organizational governance, identity and access management, and *lifecycle* automation appear less frequently. QA6 has 3 studies rated as *Yes*, QA7 has 1, and QA8 shows 11 studies rated as *No*. Operational reliability is often only partially addressed, with 22 studies rated as *Partial* in QA9.

These results show that the literature describes solutions and architectures but fails to provide complete operational mechanisms for consortia. The gaps include roles, policies, identity management, and handling of critical changes. The studies also provide limited empirical evidence in long-term scenarios.

Table 2. Quality checklist applied to the included studies (QA1–QA10). Y=Yes, P=Partial, N=No.

Study	*QA1*	*QA2*	*QA3*	*QA4*	*QA5*	*QA6*	*QA7*	*QA8*	*QA9*	*QA10*	*Points*
1 [Sato et al. 2021]	Y	Y	Y	Y	Y	Y	P	Y	P	Y	*9.0*
2 [Sato et al. 2022]	Y	Y	Y	Y	Y	Y	P	Y	P	Y	*9.0*
3 [Hawashin et al. 2025]	P	Y	Y	Y	Y	P	Y	P	P	Y	*8.0*
4 [Hoiss et al. 2021]	Y	Y	P	Y	Y	Y	P	P	P	P	*7.5*
5 [Kassab et al. 2022]	Y	Y	Y	Y	Y	N	P	P	P	Y	*7.5*
6 [von Eitzen et al. 2026]	P	Y	Y	Y	Y	N	P	P	P	Y	*7.0*
7 [Marathe et al. 2023]	P	Y	Y	Y	Y	P	P	N	P	Y	*7.0*
8 [Putri et al. 2023]	Y	Y	Y	Y	Y	N	N	P	P	P	*6.5*
9 [Silva et al. 2025]	Y	Y	Y	P	P	P	P	P	P	P	*6.5*
10 [Rondanini et al. 2020]	N	Y	Y	Y	Y	N	P	P	P	Y	*6.5*
11 [Yu et al. 2024]	Y	Y	P	Y	Y	P	P	P	P	N	*6.5*
12 [Joseph et al. 2023]	P	Y	Y	P	P	P	P	N	Y	Y	*6.5*
13 [Tran et al. 2022]	Y	P	Y	Y	Y	P	N	N	P	Y	*6.5*
14 [Bruce 2022]	P	Y	Y	Y	P	P	P	N	P	Y	*6.5*
15 [Bandara et al. 2025]	P	Y	P	Y	P	P	P	N	P	Y	*6.0*
16 [Loghin et al. 2024]	P	Y	P	P	Y	P	P	N	P	Y	*6.0*
17 [Mathwale 2023]	Y	Y	P	P	Y	N	P	P	P	P	*6.0*
18 [Bandara et al. 2020]	P	N	Y	Y	P	P	P	P	P	P	*5.5*
19 [Belchior et al. 2024]	P	P	Y	Y	P	P	P	N	P	N	*5.0*
20 [Kashansky et al. 2022]	P	Y	Y	P	P	N	N	N	P	P	*4.5*
21 [Kashansky et al. 2021]	P	Y	P	N	P	P	P	N	P	P	*4.5*
22 [Sato et al. 2025]	Y	Y	P	P	N	P	N	N	P	N	*4.0*
23 [Belchior et al. 2023]	N	P	Y	P	N	P	N	N	P	P	*3.5*

4.2. Operational categories and findings

The analysis classifies the studies' contributions into non-mutually exclusive operational categories derived from the narrative synthesis summaries and refined iteratively. This approach enables the comparison of heterogeneous results, including methods, *frameworks*, platforms, and empirical studies, based on the operational problems they address. It avoids restricting the analysis to a platform or an application domain.

Multi-organizational governance and workflows. The literature treats interorganizational operations as *workflows* with predefined parameters and coordinated execution. [Sato et al. 2021, Sato et al. 2022] model the operational *workflow* as a *smart contract* whose events are consumed by agents in each organization. This approach reduces coordination costs in tasks such as channel changes and *chaincode upgrades* and reports a 54% reduction in operational cost in an *onboarding* scenario with ten organizations. [Sato et al. 2025] extends this approach by introducing availability *guardrails* that aggregate the consortium state and constrain execution according to availability thresholds. The literature examines traceability and auditability of decisions and artifacts in regulatory contexts, including design control in the medical domain [Marathe et al. 2023]. It also describes interorganizational *workflows* with controlled information sharing in which access decisions depend on process state and task execution [Rondanini et al. 2020]. In the financial domain, studies propose consortium blockchain models for interbank operations that replace legacy integrations with distributed infrastructures that support auditing and monitoring through *smart contracts* [Bruce 2022].

These studies implement governance as coordinated operational processes. However, they provide limited support for integrated execution across organizations.

Operations and processes. The literature treats the structuring of operational work

as a requirement for transforming dispersed tasks such as scripts, commands, and configuration files into repeatable processes. A study proposes a taxonomy with seven categories and demonstrates its use in an integrated *onboarding* flow [Hoiss et al. 2021]. The same study examines IT management structures and the coupling between *on-chain* and *off-chain* elements required to operate a solution. These contributions support the standardization of operations in multi-actor environments. They address procedural and timing inconsistencies, which often lead to operational incidents.

Provisioning and lifecycle automation. The literature presents automation approaches that range from architecture-oriented *frameworks* to tools for deployment and management. A study automates blockchain deployment and evaluation from architectural descriptions by composing automation programs at *runtime* [Tran et al. 2022]. The case study reports automation of 65 blockchains across 12 architectures and generation of 295 evaluation datasets, with a planning and orchestration *overhead* of 95.5 ms. In the Hyperledger Fabric context, a study presents a solution for *deploy* and management in Kubernetes and automates tasks such as identity material generation, component configuration, and support for the *chaincode lifecycle* [Mathwale 2023]. Another study addresses automated *deploy* of *frameworks* with focus on interfaces and automation mechanisms [Yu et al. 2024]. A third study compares Docker- and Kubernetes-based architectures and incorporates mechanisms such as load balancing, exploration, and monitoring [Silva et al. 2025].

These studies show convergence toward *cloud-native* practices, including Infrastructure as Code, containers, and Kubernetes. These practices support automation and reduce deployment effort.

Security, privacy, and access control. The literature treats control-plane security and data protection as cross-cutting concerns, often implemented as layers on top of the platform. A study examines the assumption of trusted components in Fabric channel isolation and proposes separating privileges and per-channel component isolation using containers [Joseph et al. 2023]. This approach reduces the risk of data leakage in the event of node or container compromise while maintaining performance comparable to the *baseline*. A study introduces an application *firewall* for Ethereum platforms in enterprise environments that acts as an intermediary for dynamic access control and auditing, targeting privacy and compliance requirements such as GDPR [von Eitzen et al. 2026]. Another study addresses interorganizational *workflows* and shows that access policies must align with process states and execution evidence [Rondanini et al. 2020].

These studies show that access control and data protection depend on mechanisms that extend beyond the platform. However, they remain loosely integrated with orchestration and governance processes.

Resilience and observability. The literature treats resilience and observability as requirements that connect automation, telemetry, and fault tolerance in consortium operation. A study proposes a *cloud-native* suite for *benchmarking* and *chaos engineering* in Fabric to support experiment reproducibility and capacity planning [Kassab et al. 2022]. Another study investigates hybrid replication and monitoring mechanisms in Quorum and combines reactive replication with proactive monitoring to support fault tolerance during provisioning in cloud environments [Putri et al. 2023]. Studies present multiscale mon-

itoring architectures for logistics blockchains and introduce preliminary *workflow* models [Kashansky et al. 2022, Kashansky et al. 2021]. They highlight challenges related to data collection, publication, and signing, as well as dynamic task adaptation in distributed systems. Another study reports functional validation and large-scale performance evaluation in compliance-monitoring scenarios, including processing rates exceeding 72,000 *beacons* per second in a UAV context. It provides public artifacts to support replication [Hawashin et al. 2025].

These studies treat resilience and observability as operational concerns. However, they integrate them only weakly with orchestration and governance.

Portability, cost, and performance. The literature shows that infrastructure decisions such as cloud versus edge, x86 versus ARM, and Docker versus Kubernetes affect *throughput*, latency, and cost, with implications for planning and capacity governance. A study compares Fabric and Quorum across hardware configurations, including ARM, and identifies distinct performance profiles across architectures [Loghin et al. 2024]. The results show that CPU and memory choices, as well as execution location, influence both performance and total cost. Platform-oriented and *benchmarking* studies extend this analysis by examining relationships among scalability, maintainability, and architectural or automation choices. Studies describe trade-offs among configuration speed, observability, and scalability [Silva et al. 2025, Kassab et al. 2022].

These studies show that infrastructure choices introduce trade-offs that affect system performance and operational cost. However, they do not provide guidance for decision-making in multi-organizational settings.

Interoperability, migration, and ecosystem design. The literature treats interoperability and ecosystem evolution as problems that depend on abstractions for state and decision models. A study introduces the concept of *views*, defined as state snapshots and partitions, and proposes a *view* generator that captures, builds, and merges views based on rules defined by *stakeholders* [Belchior et al. 2024]. This approach supports auditing and *cross-ledger* analysis. Another study organizes interoperability mechanisms and proposes decision models that guide the selection of infrastructure and functionality based on system requirements [Belchior et al. 2023]. Additional studies examine ecosystem evolution and show the need for explicit planning of platform changes, data fidelity, and data exposure in multi-party settings [Bandara et al. 2020, Bandara et al. 2025]. These studies also identify scenarios in which centralized services provide a more suitable alternative than shared *ledgers*.

These studies show that interoperability and migration depend on explicit models for state and decision processes. However, they provide limited support for integration with operational orchestration and governance.

4.3. Synthesis by research question

4.3.1. RQ1: What solutions (academic and industrial) exist for the orchestration and operational management of permissioned blockchains?

The evidence identifies three main classes of solutions. The first class includes decentralized operation methods that treat coordination and consistency as central problems within the consortium. These approaches model critical operations as *workflows*, such as chan-

nel changes and *chaincode lifecycle*, and distribute execution across organizations with consensus and audit mechanisms [Sato et al. 2021, Sato et al. 2022, Sato et al. 2025].

The second class includes automation *frameworks* and platforms that abstract blockchain provisioning, configuration, and evaluation. These solutions often adopt a *cloud-native* orientation, including IaC, containers, and Kubernetes, and support multichannel or multi-environment architectures [Tran et al. 2022, Mathwale 2023, Yu et al. 2024, Silva et al. 2025].

The third class focuses on operational validation and reliability in realistic scenarios. These studies emphasize reproducible *benchmarking*, *fault injection*, and monitoring architectures to support evaluation under controlled and production-like conditions [Kassab et al. 2022, Kashansky et al. 2022].

In terms of maturity (RQ1a/RQ1b), the studies range from concepts and prototypes to fully implemented systems with empirical evaluation. A workflow-based approach reports an estimated 54% reduction in operational cost in a typical consortium *onboarding* scenario. This result is associated with workflow formalization, including pre-sharing of parameters, coordination, and execution, as well as distribution of execution across organizations [Sato et al. 2021, Sato et al. 2022].

An architecture-driven automation approach demonstrates scalability by automating 65 blockchains across 12 architectures and generating 295 datasets, with a planning and orchestration *overhead* of 95.5 ms [Tran et al. 2022]. In the Fabric context, *frameworks* and platforms emphasize identity, configuration, and *chaincode lifecycle* modules. Monitoring and *compliance* studies report functional validation and large-scale performance, including processing rates above 72,000 *beacons* per second [Mathwale 2023, Hawashin et al. 2025].

Taken together, the evidence shows that the state of the art covers provisioning automation and architectural description. However, it provides limited support for multiparty governance and production-grade identity and access management.

4.3.2. RQ2: Which patterns and architectural approaches support provisioning, decentralized governance, identity and access management, and smart contract lifecycle automation?

The literature identifies recurrent architectural patterns that treat operations as first-class objects in the control plane. One pattern models changes as sets of parameters and verifiable steps, such as *change sets*, combined with approval and evidence mechanisms. This approach reduces divergences across organizations and supports auditing [Sato et al. 2021, Sato et al. 2022].

Another pattern separates the decision and coordination stage from the technical execution stage. The decision stage may be *on-chain* or immutably recorded, while execution occurs *off-chain* through agents and executors that consume events and operate on infrastructure and nodes under the control of each participant [Sato et al. 2022, Sato et al. 2025]. In *workflow* orchestration, *smart contracts* also act as coordination mechanisms and enable conditional authorizations for *off-chain* resource sharing [Rondanini et al. 2020]. Architecture-oriented *frameworks* and runtime-based *au-*

tomation further support *stack* heterogeneity and platform evolution without requiring a complete redesign of *pipelines* [Tran et al. 2022].

Regarding identity and access management and security, the literature shows that effective governance depends on explicit control over identities, privileges, and policies in the control plane rather than solely on *ledger* guarantees. Mechanisms such as separation of privileges and domain isolation reduce operational risk and support confidentiality requirements in permissioned environments [Joseph et al. 2023]. In Ethereum-based architectures, intermediary components with policy *enforcement* and auditing capabilities address gaps in granular access control, especially in regulated scenarios [von Eitzen et al. 2026].

From the *lifecycle* perspective, studies report automation of network and *chaincode* tasks, including Kubernetes-based *deploy* and management *frameworks*. However, coverage remains fragmented, particularly for *rollback*, versioning, and multi-organizational policies [Sato et al. 2021, Sato et al. 2022, Mathwale 2023]. Finally, the evidence indicates that *lifecycle* automation and multi-party governance should be co-designed with observability and availability requirements to prevent service degradation during critical operations [Sato et al. 2025, Kassab et al. 2022].

4.3.3. RQ3: Which gaps and design principles emerge for an open-source, *stack*-agnostic orchestration solution?

The evidence shows persistent gaps in multi-organizational execution in real-world contexts. Prior coordination remains costly and poorly systematized, including negotiation of parameters, execution windows, and responsibilities. Traceability of decisions and evidence is often limited, and support for idempotent execution with safe recovery appears only in specific *workflow* proposals [Sato et al. 2021, Sato et al. 2022].

The quality checklist reinforces this assessment. Multi-organizational governance and identity and access management are rarely implemented as complete mechanisms, with QA6 and QA7 predominantly marked as *Partial* or *No*. *Lifecycle* automation also shows limited coverage, with a high incidence of *No* in QA8.

Integration between operation and reliability remains weak. Studies discuss availability *guardrails*, but do not consistently connect them to telemetry or to explicit operational criteria. In addition, few studies report longitudinal evaluation in production environments [Sato et al. 2025, Kassab et al. 2022]. Heterogeneity in artifact formats and the availability of implementation further limit comparison and reuse.

The synthesis defines a set of design principles for a *stack*-agnostic orchestration solution. First, the control plane should treat auditable change *workflows* as a core abstraction and support policy-driven multi-party approval and explicit evidence trails. Second, the solution should integrate identity and access management, as well as privilege segregation, into the operational *lifecycle*. Third, it should provide portable automation abstractions that reduce *lock-in* and support composition of *deploy*, configuration, and evaluation across different *stacks*. Fourth, it should connect operational decisions to telemetry, including observability, Service Level Objectives, and *guardrails*, to control risk during critical changes [Tran et al. 2022, Sato et al. 2025, Loghin et al. 2024, Bandara et al. 2020].

Finally, the solution should address interoperability and migration from the design stage. It should adopt methods and decision models that make explicit the *trade-offs* among privacy, cost, and data fidelity throughout ecosystem evolution [Belchior et al. 2023, Belchior et al. 2024, Bandara et al. 2025].

The evidence reveals persistent gaps in multi-organizational execution in real-world contexts. In particular, studies still do not systematize prior coordination, including negotiation of parameters, execution windows, and responsibilities, nor do they consistently support traceability of decisions and evidence or idempotent execution with safe recovery outside specific *workflow* proposals [Sato et al. 2021, Sato et al. 2022]. The quality checklist reinforces this diagnosis. Studies rarely implement multi-organizational governance and identity and access management as complete mechanisms, with QA6 and QA7 predominantly marked as *Partial* or *No*. They also often omit *lifecycle* automation, as indicated by the high incidence of *No* in QA8. In addition, studies discuss availability *guardrails* but do not consistently link them to telemetry and explicit operational criteria, and few studies report longitudinal evaluation in production environments [Sato et al. 2025, Kassab et al. 2022]. Heterogeneity in artifact formats and the availability of implementation further limit direct comparison and reuse.

4.4. Threats to validity

This study presents several threats to validity. First, publication bias and the unavailability of full texts may underrepresent industrial evidence. Second, the results depend on the search string and selected sources, which risks missing relevant terms or sources. Third, heterogeneity in the type of evidence, including concepts, methods, prototypes, and empirical studies, limits direct comparison across studies. To mitigate these threats, the protocol defined explicit criteria, applied a structured selection process, and documented all decisions and artifacts.

5. Conclusion

This systematic mapping consolidates a corpus of 23 primary studies (2020–2025) with direct contributions to orchestration and governance in permissioned blockchains. The quality checklist shows that the literature consistently describes platforms, *stacks*, architectures, and procedures. However, it reveals recurring weaknesses in complete mechanisms for multi-organizational governance, identity and access management, and *lifecycle* automation, particularly in *rollback*, versioning, and multi-organizational policies. These gaps highlight the distance between proposed solutions and the consortium's operations in production environments.

The findings indicate that the main challenge extends beyond the distributed execution of transactions to the design of the consortium control plane. Designing this control plane requires multi-party coordination of critical changes, management of identities and privileges, provisioning and *lifecycle* automation, and integration of reliability requirements such as observability, availability, and resilience into the change process. The classification by operational categories shows convergence toward *cloud-native* approaches and auditable *workflows*, while also exposing limitations in traceability and integration with telemetry.

This paper contributes by organizing the evidence by operational category and research question and by making explicit recurring patterns and design principles for

stack-agnostic orchestration solutions. The results support a research agenda that includes: expanding empirical evidence through comparative and reproducible studies in multi-organizational scenarios; operationalizing availability *guardrails* and change criteria based on Service Level Objectives; standardizing artifacts and interfaces to reduce friction across *stacks*; consolidating governance and identity and access management mechanisms within the control plane; and advancing interoperability and migration strategies that reconcile privacy, cost, and data fidelity. The protocol registration and documented decisions in Parsifal support replication and enable continuous evolution of this mapping.

Acknowledgments

The present work was carried out with support from the Brazilian National Telecommunications Agency (Anatel) and the Research Support Foundation of the State of Goiás (FAPEG).

References

Bandara, H. D., Staples, M., and Malik, S. (2025). Designing for shared ledgers in industry ecosystems.

Bandara, H. D., Xu, X., and Weber, I. (2020). Patterns for blockchain data migration.

Belchior, R., Riley, L., Hardjono, T., Vasconcelos, A., and Correia, M. (2023). Do you need a distributed ledger technology interoperability solution?

Belchior, R., Torres, L., Pfannschmidt, J., Vasconcelos, A., and Correia, M. (2024). Bungee: Dependable blockchain views for interoperability.

Bruce, R. A. (2022). Proposição de um modelo de blockchain para operações interbancárias. Master's thesis.

Hawashin, D., Madine, M., Nemer, M., Salah, K., Jayaraman, R., Damiani, E., and Yaqoob, I. (2025). Leveraging hyperledger fabric for enhanced compliance monitoring in uav operations.

Hoiss, T., Seidenfad, K., and Lechner, U. (2021). Blockchain service operations - a structured approach to operate a blockchain solution.

Joseph, A., Yadav, N., Ganapathy, V., Behl, D., and Jayachandran, P. (2023). Data protection in permissioned blockchains using privilege separation.

Kashansky, V., Prodan, R., Validi, A., Olaverri-Monreal, C., and Radchenko, G. (2022). Monitoring system architecture for the multi-scale blockchain-based logistic network.

Kashansky, V., Saurabh, N., Prodan, R., Validi, A., Olaverri-Monreal, C., Burian, R., Burian, G., Hirsch, D., Lv, Y., Wang, F.-Y., and Zuhge, H. (2021). The adapt project: Adaptive and autonomous data performance connectivity and decentralized transport network.

Kassab, A., Rivière, E., Rosinosky, G., Sadre, R., and Tran, V. H. (2022). C2b2: a cloud-native chaos benchmarking suite for the hyperledger fabric blockchain.

Loghin, D., Dinh, T. T. A., Gang, C., Teo, Y. M., and Ooi, B. C. (2024). Characterizing the performance and cost of blockchains on the cloud and at the edge.

Marathe, N., Chung, L., and Hill, T. (2023). Implementing cross-organizational fda medical device design controls using blockchain.

Mathwale, R. (2023). Ahfd: A framework for deployment and management of hyperledger fabric enterprise blockchain.

Putri, R. S. R., Bhawiyuga, A., Akbar, S. R., Shaffan, N. H., Amron, K., and Basuki, A. (2023). Implementation of fault-tolerance mechanism in quorum-based blockchain provisioning in cloud infrastructure using replication and monitoring protocols.

Rondanini, C., Carminati, B., Daidone, F., and Ferrari, E. (2020). Blockchain-based controlled information sharing in inter-organizational workflows.

Sato, T., Shimosawa, T., and Himura, Y. (2021). Opssc: Decentralized blockchain network operation workflow for hyperledger fabric.

Sato, T., Shimosawa, T., and Himura, Y. (2022). Operations smart contract to realize decentralized system operations workflow for consortium blockchain.

Sato, T., Shimosawa, T., and Yamai, N. (2025). Concept of haopssc: Toward decentralized operations for ensuring high availability in consortium blockchain-based systems.

Silva, P., Guimaraes, T., Duarte, R., and Filipe Santos, M. (2025). Scalable and sustainable blockchain: Architecting infrastructure and developing a platform for efficient management and exploration.

Tran, N. K., Babar, M. A., and Walters, A. (2022). A framework for automating deployment and evaluation of blockchain networks.

von Eitzen, C. D., Fernández-Iglesias, M. J., Anido-Rifón, L., and Mikic-Fonte, F. A. (2026). Blockchain beyond immutability: Application firewalls on ethereum-based platforms.

Yu, H., Wang, C., Wang, Z., and Xing, S. (2024). Automatic deployment of hyperledger fabric frameworks.