

Designing Smart-Contract-Enabled Liquidity Management for Wholesale Rupiah Digital: A Design Science Research Approach

Satria Rana Dityantomo ¹, Made Harta Dwijaksana ²

^{1,2} Faculty of Computer Science, University of Indonesia, Depok, Indonesia

Received:

October 14, 2025

Revised:

April 26, 2026

Accepted:

May 30, 2026

Published:

June 22, 2026

Corresponding Author:

Author Name*:

Satria Rana Dityantomo

Email*:

satria.rana@ui.ac.id

DOI:

10.63158/journalisi.v8i3.1609

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. This study proposes a smart-contract-enabled liquidity management reference architecture for Bank Indonesia's Wholesale Rupiah Digital (wRD), addressing a gap in Project Garuda where the Proof-of-Concept adopts a gross settlement model supplemented by decentralized queueing but lacks a liquidity-saving mechanism. Using Design Science Research (DSR), a Multivocal Literature Review (MLR) was conducted to synthesize eight academic and grey literature sources that include central bank reports, BIS publications, and industry white papers. Five key takeaways were extracted and mapped to architectural modules. The resulting architecture operates on a unified ledger and comprises five modules: a Policy & Criteria smart contract for participant eligibility and compliance, a centralized Queue smart contract with configurable release rules, a Liquidity-Saving Mechanism (LSM) for periodic multilateral netting, an Automated Liquidity Provisioning (ALP) module for just-in-time intraday repo against tokenized bonds when netting is insufficient, and a guarded Automated Market Maker (AMM) for cross-currency settlement. The architecture is illustrated through a payment state lifecycle with two scenario transaction walkthroughs and assessed through a criteria-based analytical evaluation that verifies design completeness against requirements. Empirical validation through prototyping, simulation, or expert review is left for future work. The main contribution is a literature-derived reference architecture that integrates global CBDC mechanisms adapted to Indonesia's context.

Keywords: Wholesale CBDC, Smart Contract Architecture, Liquidity-Saving Mechanism, Design Science Research, Multivocal Literature Review, Intraday Liquidity

1. INTRODUCTION

Central banks around the world are studying digital currencies to modernize payment infrastructure. A Central Bank Digital Currency (CBDC) gives the central bank a digital form of its own money that still carries finality, liquidity, and integrity [1]. One direction that has gained attention in recent years is programmable settlement through Distributed Ledger Technology (DLT), which allows operators to write smart contract rules directly into the ledger so that routine checks and transfers execute without manual steps and with fewer errors [2]. In practice, smart contracts have been used to handle money issuance [3], intraday collateral operations [4], payment queuing [5], and various other settlement tasks like cutting costs and reducing operational risk. Indonesia Wholesale CBDC (w-CBDC) named Project Garuda's Wholesale Rupiah Digital (wRD) in the latest Proof of Concept (PoC) report produced working prototypes where smart contracts manage issuance, redemption, and participant position across two permissioned DLT named R3 Corda and Hyperledger Besu [3]. The next phase of the Project Garuda roadmap calls for smart-contract-enabled liquidity management that adds liquidity saving mechanisms, priority queuing, and gridlock resolution [3], [6].

Although Project Garuda has made progress on w-CBDC, no dedicated architecture for liquidity management yet exists. The current PoC relies on gross settlement with decentralized queueing, a practical starting point but not a full proper liquidity saving mechanism proses. Improving this design requires smart-contract-enabled mechanisms for the key settlement and liquidity processes. The open question is how Bank Indonesia should determine, design, and configure such mechanisms, particularly liquidity-saving mechanisms, priority queuing, and gridlock resolution suited to Project Garuda's w-CBDC context. Getting these wrong carries real risk, where poorly specified or weakly governed smart contracts can introduce security vulnerabilities and financial losses [7], [8]. That risk makes designed architecture essential rather than an ad hoc approach.

Several foundational projects have addressed on liquidity management design in recent years. Projects such as Stella (ECB-BoJ) and Ubin (MAS) embed liquidity-saving mechanisms and queuing or gridlock controls within smart contracts, while Helvetia (SNB-BIS) and mBridge extend these to cross-border contexts [9], [10], [11], [12]. We can see most w-CBDC practical documentation research sits in the grey literature area

published in white papers and articles by BIS or the Central Bank rather than in peer-reviewed academic journals. This distribution of knowledge has two methodological implications that use for this study. First, a Multivocal Literature Review (MLR) is required to systematically capture both academic and grey literature sources that a traditional Systematic Literature Review (SLR) would fail to include [13]. Second, because the goal is not merely to synthesize existing knowledge but to produce a reference architecture, this study adopts Design Science Research (DSR) approach with the MLR serving as the knowledge-base-building phase.

Despite Project Garuda's PoC progress, no dedicated architecture for wRD liquidity management currently exists, and global design lessons from existing w-CBDC projects have not been systematically synthesized for the Indonesian context. To address this gap, this study proposes a reference architecture for Project Garuda's wRD with smart contract module specifications and governance considerations derived through a Multivocal Literature Review (MLR) embedded within a Design Science Research (DSR) framework. Two research questions guide the work: RQ1 – what broad approaches to smart-contract-enabled liquidity management have been explored globally, and what can be learned from them? RQ2 – how should these lessons inform the architecture of smart-contract-enabled liquidity management for Indonesia's wRD? The novelty lies in the traceable, evidence-based derivation of a wRD-specific liquidity management architecture grounded in a multivocal synthesis of global w-CBDC projects. The main contribution is a traceable, evidence-based reference architecture that integrates global w-CBDC design lessons into implementable smart contract modules adapted to Indonesia's institutional and regulatory setting [3], [6].

2. METHODS

This study uses Design Science Research (DSR) as its main methodology. DSR is oriented toward building and evaluating artifacts such as architectures, frameworks, models, or systems that address specific problems in the information systems field [14], [15]. Hevner et al. [15] describe DSR as operating in three interconnected cycles. The relevance cycle ties the work to an actual problem, the rigor cycle draws on what is already known in the literature, and the design cycle iterates between building and evaluating the artifact. Peffers et al. [14] translate this into six sequential activities, which are problem

identification, objective definition, design and development, demonstration, evaluation, and communication.

In this study, the artifact is a smart-contract-enabled reference architecture for liquidity management in Indonesia's Wholesale Rupiah Digital (wRD). The real-world problem is the absence of a comprehensive liquidity management architecture in Project Garuda's current PoC. The knowledge base is constructed through a Multivocal Literature Review (MLR) following Garousi et al. [13], which systematically synthesizes both peer-reviewed academic studies and grey literature. Grey literature is particularly important in the CBDC domain because much of the practical knowledge emerges from central bank reports, international financial institution publications (BIS, IMF, World Bank), technical white papers, and industry documentation rather than from peer-reviewed journals alone [16]. The design cycle translates the MLR findings into traceable design requirements, which are then mapped to architectural modules with specified operations, parameters, and governance. Table 1 maps each DSR activity to the corresponding section of this paper.

Table 1. DSR Activity Mapping

DSR Activity (Peppers et al. [14])	Implementation in This Study	Paper Section
Problem Identification & Motivation	Gap in Project Garuda's liquidity management architecture identified through analysis of PoC documentation [3], [6]	Introduction
Objectives of a Solution	Two research questions defined: RQ1 (global design lessons) and RQ2 (Indonesia-specific architecture)	Introduction
Design & Development – Knowledge Base	Multivocal Literature Review (MLR) following Garousi et al. [13] to synthesize academic and grey literature into key design takeaways	Method (Phase 1) and Results (RQ1)
Design & Development – Artifact Construction	Key takeaways mapped to architectural modules via traceability table; module specifications defined	Method (Phase 2) and Results (RQ2)
Demonstration	Payment state lifecycle defined; scenario walkthrough traces a sample transaction through the architecture	Results (RQ2)
Evaluation	Criteria-based assessment verifying that each design requirement is addressed by the architecture	Results (RQ2)
Communication	This paper	-

The methodology unfolds as a seven-stage operational workflow. Step 1 (literature gathering) runs Boolean searches across four academic databases (Scopus, IEEE Xplore, ScienceDirect, Taylor & Francis) and a domain-filtered Google search to produce an initial candidate pool, which is then expanded through forward and backward snowballing and narrowed by applying the inclusion/exclusion criteria (Table 4). Step 2 (synthesis) reads each retained source using the 3C+2S framework (Compare, Contrast, Criticize, Synthesize, Summarize) to extract recurring design patterns and mechanisms across the literature. Step 3 (requirement extraction) condenses the synthesis into a set of Key Takeaways that function as literature-derived design requirements for the architecture. Step 4 (traceability mapping) maps each Key Takeaway to one or more proposed architectural modules, ensuring that no module exists without literature support and no Key Takeaway is left unaddressed. Step 5 (module design) specifies, for each module, the core operations, key parameters, interaction logic, and state transitions at reference-architecture granularity. Step 6 (demonstration) traces sample payments through the modules using a payment state lifecycle and transaction scenario walkthroughs. Step 7 (evaluation) verifies through a criteria-based analytical assessment that every Key Takeaway is addressed by at least one module. Steps 1–3 build the knowledge base (Phase 1, MLR); Steps 4–5 construct the artifact (Phase 2); Steps 6–7 demonstrate and evaluate it (Phase 3).

2.1 Phase 1 - Knowledge Base: Multivocal Literature Review

The MLR follows guidelines from Garousi et al. [13], adapted from Kitchenham and Charters' SLR framework [17]. The MLR process comprises three main phases as shown in Table 2.

Table 2. Kitchenham phases of SLR guidelines [17]

Phase	Steps
Planning the review	• Identification of the need for a review
	• Commissioning a review
	• Specifying the research question(s)
	• Developing a review protocol
Conducting the review	• Evaluating the review protocol
	• Identification of research

Phase	Steps
	<ul style="list-style-type: none"> • Selection of primary studies • Study quality assessment • Data extraction and monitoring • Data synthesis
	<ul style="list-style-type: none"> • Specifying dissemination mechanisms
Reporting the review Support	<ul style="list-style-type: none"> • Formatting the main report • Evaluating the report

Planning. An MLR is needed here because evidence on w-CBDC liquidity management is scattered across academic journals and fast-moving policy and industry publications rather than residing in peer-reviewed papers alone. Before making goals and questions, MLR guidance recommends confirming that a review is truly important and clarifying usefulness to the intended audience already in the planning stage. After that need is confirmed, Research Questions (RQs) are formulated to define what the study sets out to answer. RQs should form the basis for the entire review and remain closely connected to the research goal [18]. The RQs for this study are already stated in Section 1. Planning also involves preparing a review protocol that lays out the procedural steps for conducting the review [19]. Search strings are generated using research keywords that build upon research questions and existing literature [20]. They are expressed as short, repeatable Boolean strings that combine the core concepts of w-CBDC, liquidity management, and smart contracts and are reused across White Literature databases and Grey Literature domains. Table 3 presents the protocol used in this research.

Table 3. Kitchenham MLR Protocol

Type	Database	Search Dates	Search String
White Literature (Peer-reviewed academic sources)	Scopus	11 August 2025 – 16 August	TITLE-ABS-KEY (("Wholesale Central Bank Digital Currency" OR "wholesale CBDC" OR "w-CBDC") AND "smart contract" AND ("liquidity management" OR "liquidity-saving mechanism" OR "queueing" OR "gridlock resolution"))
	IEEE Xplore	28 July 2025 – 1 August 2025	("Wholesale Central Bank Digital Currency" OR "wholesale CBDC" OR w-CBDC) AND "smart contract" AND ("liquidity

Type	Database	Search Dates	Search String
			management" OR "liquidity-saving mechanism" OR "queueing" OR "gridlock resolution")
	ScienceDirect	17 August 2025 – 24 August 2025	("Wholesale Central Bank Digital Currency" OR "wholesale CBDC" OR w-CBDC) AND "smart contract" AND ("liquidity management" OR "liquidity-saving mechanism" OR "queueing" OR "gridlock resolution")
	Taylor & Francis	24 August 2025 – 25 August 2025	("Wholesale Central Bank Digital Currency" OR "wholesale CBDC" OR w-CBDC) AND "smart contract" AND ("liquidity management" OR "liquidity-saving mechanism" OR "queueing" OR "gridlock resolution")
Grey Literature. (Industry and institutional reports, and practitioner documents (Domain: Government, Organization, Public financial reports, Fintech)	Google	25 August 2025 – 3 September 2025	site:bis.org OR site:imf.org OR site:worldbank.org OR site:ecb.europa.eu OR site:bankofengland.co.uk OR site:mas.gov.sg OR site:boj.or.jp OR site:bankofcanada.ca OR site:snb.ch OR site:hkma.gov.hk OR site:cbu.gov.ae OR site:sama.gov.sa OR site:rba.gov.au OR site:riksbank.se OR site:rbi.org.in OR site:banxico.org.mx OR site:bcg.gov.br OR site:resbank.co.za OR site:bankofkorea.or.kr OR site:hyperledger.org OR site:r3.com OR site:swift.com ("Wholesale Central Bank Digital Currency" OR "wholesale CBDC" OR w-CBDC) AND "smart contract" AND ("liquidity management" OR "liquidity-saving mechanism" OR "queueing" OR "gridlock resolution")

The literature selection is based on inclusion and exclusion criteria defined in Table 4 [17]. To determine this, the criteria from previous research will be used as the basis [21], [22].

Table 4. Inclusion and Exclusion Criteria

Inclusion	Exclusion
Literature written in English or Indonesia.	Literature not written in English or Indonesia.
Literature was published between 2019 and 2025.	Duplicate Literature was excluded.
Abstract related to smart contract and liquidity management	Literature that is not in the scope
Literature with title, date, and identified author	Literature that are not peer reviewed or not formal industry and institutional reports

Conducting. The conducting phase, as described by Garousi et al. [13]. Identification of research and selection of primary studies is initiated. Step one: an initial search was conducted using Boolean strings across curated databases and domain-filtered Google, with results already deduplicated and add into initial pool. Duplicates were removed by manual checking (comparing title, author(s), publication year). Step two: snowballing is conducted to expand coverage from seed items by doing forward and backward searching to add more relevant literature. Final Step: the enlarged pool was screened independently against the inclusion and exclusion criteria. Synthesis was conducted manually using narrative synthesis, where findings from the included studies were summarized and compared to produce an overall integrated interpretation and then the final pool was produced. After the final pool was produced, manual quality assessment was performed to evaluate the credibility of the studies. Table 5 outlines the details of this MLR approach for literature selection.

Table 5. Literature Selection Process

Database	Step one	Step two	Final Step
	Initial Search	Snowballing	Inclusion and Exclusion Criteria
IEEE	Included = 2	Included = 3	Included = 1
	Excluded = 0	Excluded = 0	Excluded = 2
Scopus	Included = 3	Included = 3	Included = 0
	Excluded = 0	Excluded = 0	Excluded = 3
ScienceDirect	Included = 7	Included = 8	Included = 1
	Excluded = 0	Excluded = 0	Excluded = 7
Taylor & Francis	Included = 0	Included = 0	Included = 0
	Excluded = 0	Excluded = 0	Excluded = 0
Google	Included = 318	Included = 318	Included = 6
	Excluded = 0	Excluded = 0	Excluded = 312
Total	Included = 330	Included = 332	Included = 8
	Excluded = 0	Excluded = 0	Excluded = 324

Following the study quality-assessment step of the MLR process (Table 2), the eight retained sources were assessed for credibility against the four criteria defined in the quality-assessment Table 6. Screening and assessment were conducted by the lead author. The peer-reviewed sources establish the theoretical and deterministic properties of smart-contract settlement, while the grey-literature primary reports supply implementation evidence from the institutions developing w-CBDCs. Below Table 6 list of criteria of the quality assessment:

Table 6. Quality Assessment Criteria

Criteria	Description
Authority	Issued by a central bank, the BIS, or a peer-reviewed venue
Objectivity	Balanced primary documentation, not promotional material
Methodology	States its approach or implementation basis so claims are traceable
Recency	Published recently enough to reflect the current state of the field

Reporting. The reporting phase as outlined by Kitchenham et al. [17] represents the final stage of the literature review process. In this phase, the data obtained from the selected studies from the selection proses are extracted, synthesized, and presented clearly and concisely [13]. This presentation provides a thorough examination of the review findings, along with a discussion of their implications. The selected literature is defined in Table 7.

Table 7. Chosen Literature

Code	Database	Literature Type	Title	year
WL1	IEEE	White Literature	A Compendium of Practices for Central Bank Digital Currencies for Multinational Financial Infrastructures	2020
WL2	ScienceDirect	White Literature	A Blueprint for Energy Systems in the Era of Central Bank Digital Currencies	2024
GL1	Google	Grey Literature	Bank for International Settlements, Annual Economic Report 2023	2023
GL2	Google	Grey Literature	Central Bank Digital Currency: Opportunities, Challenges and Design, Discussion Paper	2020

Code	Database	Literature Type	Title	year
GL3	Google	Grey Literature	Project Dunbar: International Settlements Using Multi-CBDCs	2022
GL4	Google	Grey Literature	Project Mariana Interim Report: Cross-Border Exchange of Wholesale CBDCs Using Automated Market-Makers	2023
GL5	Google	Grey Literature	Project Rialto: Improving Instant Cross-Border Payments Using Central Bank Money Settlement – Interim Report	2025
GL6	Google	Grey Literature	Inthanon Phase I: An application of Distributed Ledger Technology for a Decentralized Real-Time Gross Settlement system using wholesale central bank digital currency	2019

The pool of eight sources retained in Table 7 (WL1, WL2; GL1–GL6) emerged from applying the inclusion and exclusion criteria, not from a pre-set target. Each of the eight sources was assessed against the four quality criteria defined in table 6 and all eight satisfied every criteria. Its sufficiency for architecture derivation rests on two properties. First, the pool grounds queue management and liquidity management where the two architectural domains identified in Project Garuda's roadmap [3], [6] as the focus for the next phase of wRD development where each supported by at least one source. Supporting elements such as cross-currency settlement and policy enforcement are also represented. Second, the pool prioritizes peer-reviewed studies and primary project reports from the institutions implementing w-CBDCs, providing higher-trust design evidence than secondary commentary would. The energy-systems blueprint (WL2) is retained as cross-domain, peer-reviewed evidence that settlement can be governed by smart-contract-enforced rules, which reinforces the peer-reviewed basis of an otherwise grey literature heavy pool.

2.2 Phase 2 - Design & Development: Architecture Construction

The design and development phase translates the knowledge base from Phase 1 (MLR) into the paper's central artifact: a proposed reference framework for smart-contract-enabled w-CBDC liquidity management, instantiated as a five-module reference

architecture for Indonesia's wRD. Following Peffers et al. [14], the framework is constructed in three sequential steps that are requirement extraction, traceability mapping, and module specification with each step's output traceable back to specific sources in the literature pool, ensuring the framework is derived through traceable DSR logic.

Requirement Extraction. The eight sources identified through the MLR (Table 7) are analyzed and synthesized using the 3C+2S framework (Compare, Contrast, Criticize, Synthesize, Summarize) by Sugiyono [23]. The framework is operationalized as five reading lenses applied to each source, with the source treated as the unit of analysis. The Compare lens identifies recurring smart-contract liquidity-management mechanisms that appear in two or more sources, on the assumption that repetition across independent projects indicates a stable design pattern. The Contrast lens captures what is distinctive about each source – its scope, target architecture, mechanism focus, or institutional context to position each source relative to the others. The Criticize lens records methodological or substantive limitations of each source, including unresolved problems, missing references, scope gaps, or design weaknesses that constrain its applicability to the wRD context. The Synthesize lens extracts from each source the single design-relevant insight or mechanism that most directly informs the architecture (typically a specific liquidity-management mechanism, a smart-contract design principle, or an integration approach). The Summarize lens then consolidates the per-source insights into an integrative statement across the full pool. The output of this five-lens synthesis is reported in Table 8 (Results section). The recurring patterns identified through the Synthesize lens are condensed into Key Takeaways in Table 9 (Results section), which are then mapped to architectural modules via the traceability table in Table 10 (Results section).

Traceability Mapping. Each design key takeaway is mapped to one or more architectural modules establishing a link between the evidence from the MLR process with the design in the proposed architecture. This mapping ensures that no module is introduced without supporting evidence from the literature, and that no key takeaway is left unaddressed in the architecture. The traceability mapping is presented in Table 11 in the Results section (RQ2 discussion). This step addresses a core DSR requirement: that design decisions must be grounded in the existing knowledge base rather than derived from intuition alone [15].

Module Specification. For each architectural module identified through the traceability mapping, the study defines: (a) core operations: the functions the smart contract exposes and their input/output behavior, (b) key parameters: the configurable values that govern the module's behavior and their governance ownership, (c) interaction logic : how the module connects to adjacent modules in the architecture, and (d) state transitions: how a payment instruction's state changes as it passes through the module. The specification level is intended as a reference architecture sufficiently detailed to guide implementation but abstracted above platform-specific code. The module specifications are presented in the Results section (RQ2 discussion) and summarized in Table 11 (Results section).

2.3 Phase 3 - Demonstration and Evaluation

The demonstration and evaluation phase completes the DSR process by assessing whether the proposed reference framework addresses the problem identified in Phase 1 and satisfies the design requirements extracted from the MLR [14].

Demonstration. The architecture is demonstrated through two mechanisms. First, a payment state lifecycle is defined, specifying the complete set of states a payment instruction can occupy as it traverses the architecture and the transitions between them. This provides a formal description of the architecture's dynamic behavior. Second, a transaction scenario walkthrough traces a sample payment through the modules.

Evaluation. The architecture is evaluated using a criteria-based assessment. Each key design takeaway identified in the MLR (Table 10) serves as an evaluation criteria. The assessment verifies that every design requirement is addressed by at least one architectural module, and that no module exists without supporting evidence from the literature. This approach follows Hevner et al.'s [15] recommendation that DSR artifacts should be evaluated against the objectives of the solution. The study acknowledges that this evaluation constitutes an analytical assessment of design completeness; empirical validation through simulation, prototyping, or expert review falls outside the scope of this study and is recommended as future work.

3. RESULTS AND DISCUSSION

3.1 RQ1— What broad approaches to smart-contract-enabled liquidity management on w-CBDC research have been explored globally, and what can be learned from it?

Smart-contract-enabled liquidity management in w-CBDC Global research has expanded to improve the finality and programmability of transactions in digital money. Opare et al. [24] underscore permissioned DLT as the operational basis for w-CBDC by surveying real-world CBDC PoC and implementations. Although it does not focus on designing smart contract liquidity management, the document details multiple projects in which smart contracts encode settlement rules and liquidity-related controls (e.g., sufficient-liquidity checks, conditional holds, etc.), providing contextual evidence for the architecture discussed in this literature. Additionally, a use case exists where smart contracts automate energy/carbon market clearing and CBDC settlement, boosting transparency, cutting reconciliation costs, and enabling programmable real-time participation [25].

The BIS Annual Economic Report 2023 propose a concept of a unified ledger, a single programmable platform where tokenized money and assets live together and smart contracts handle transaction operations [26]. The report defines programmability as the capability to encode embed "if-this-then-that" condition that automates compliance, market design, and participant coordination. The Bank of England share a similar view where positioning smart contract programmability as a path toward "programmable money" which support atomic transactions. In an atomic transaction, CBDC transfers only finalize if both sides succeed. This will lower settlement risk for domestic and cross-currency payments alike [27]. Project Dunbar reinforces this point by showing that business rules like sufficient liquidity checks, conditional holds, and timed releases can be written into smart contracts on a shared ledger an automating settlement without extra intermediaries [28].

Project Mariana explores the use of programmable Automated Market Makers (AMM) specifically for wholesale CBDC. In this model, liquidity pools hold combination of wholesale CBDCs and algorithmically generate two-way prices. These allowing participants to swap one CBDC type for another CBDC as needed [29]. Unlike traditional order books that match individual buyers and sellers, AMM rely on smart contracts that

encode the pricing rule, fee structure, and pool-share accounting for liquidity providers [29]. Project Rialto takes a similar approach, The project structuring the AMM as one or more smart contracts that hold pooled liquidity and act as the sole counterparty in foreign exchange transactions [30]. However, Rialto also highlight some warning. Across the pilots, AMM-based FX in wholesale settlement showed vulnerability to price instability caused by exchange-rate volatility, and the mechanism can be exposed to manipulation when price determination is weak.

The Inthanon LionRock project report describes a Liquidity Saving Mechanism (LSM) concept that looks for netting opportunities, which are groups of queued payments that, if paid all at once, need less money than paying them one at a time [31]. A Bank of Thailand "LSM oracle" safely and privately coordinates this. It only asks banks for the minimum amount of information they need, such as their queued items, available cash, and the cash value of any bond tokens. It then makes a netting plan and tells only the banks involved what to do. The Corda notary still guarantees finality and stops double-spending. The project adds "deadlock" relief just in time through Automated Liquidity Provisioning (ALP) in case gridlock resolution does not find any netting solution. ALP is an automated intraday repo between the liquidity-short bank and the BOT that uses tokenized bonds as collateral to inject "just enough" cash tokens to clear the blockage [31]. These unblock queued payments without the need for manual treasury intervention.

To summarize all projects and literature that have been found, this study employs the 3C+2S framework by Sugiyono as a tool to organize insights systematically [23]. The purpose of this framework is to help researchers compare findings, highlight differences, identify limitations, integrate knowledge, and present results more clearly. The following table shows the review works using the 3C+2S approach.

Table 8. 3C+2S Framework

Literature	WL1	WL2	GL1	GL2	GL3	GL4	GL5	GL6
Compare	Across Literature, smart contracts are consistently used to achieve rule-based settlement and automate liquidity. Across pilot's projects, reported effects of implementation include near instant settlement and reduced manual intervention also integrating mechanism Automatic release of queued payments							
Contrast	Literature highlights high level w-CBDC	Literature applies smart	Literature is full report	Literatur e explains	introducin g multiple CBDCs for	Focus on cross-border infrastruktur	Literature targets retail cross-border	Focusing on exploring

Literature	WL1	WL2	GL1	GL2	GL3	GL4	GL5	GL6
	DLT concept that are already implemented across world CBDC project.	contracts to L2 energy/car bon markets liquidity	about future monetary system	about key guidance of CBDC impleme ntation	faster and cheaper internation al settlement .	e and Foreign exchange transaction	payments that use w-CBDC infrastructure of DLT for financial market	the potential of DLT for financial market
Criticize	Recommendation is to improve using framework that can be replicated by other people like MLR	Lacks a formal reference-gathering methodology.	The blueprint lacks sufficient references for diverse use cases.	Scoping-oriented rather than decision-ready.	Highlight in the literature where it ended with more questions than answers before it started.	Explicit scope, but lacks the documented dependencies needed for formal planning.	Literature describes the planned financial market infrastructure but offers too few visuals to make the architecture and flows clear.	Literature has no Reference s/Bibliography section that can be referenced and read by other reader
Synthesize	The literature identifies applicable records CBDC projects where smart contracts enforce settlement and liquidity controls.	Automates clearing and settlement, ensuring auditability and lowering reconciliati on costs.	Defines a unified ledger using smart contracts for integrated transactio n settlement	Enables atomic transacti ons (all-or-nothing settleme nt) to reduce settleme nt risk.	Embedded business rules automate settlement workflows without intermedia ries.	Introduce smart contract base AMM that have liquidity pool to automaticall y exchange tokenized assets	Automates intermediary-free settlement by embedding rules like liquidity checks and conditional holds.	Resolves queue gridlocks using LSM, ALP, and bond tokens.
Summarize	Overall, the literature converges that smart contracts can support liquidity, outlines the key checks required for reliable implementation, and documents the benefits of the implementation							

3.2. RQ2— How should these design lessons inform the architecture of smart-contract-enabled liquidity management for Indonesia's wRD?

As reported in Indonesia's w-CBDC PoC reports, Indonesia's initial wholesale Rupiah Digital architecture liquidity management design adopts a gross settlement model that is supplemented by decentralized queuing a pragmatic and lightweight approach to

liquidity management rather than a full liquidity saving mechanism [3]. Informed by the that, this paper sets out concrete updates to the liquidity management architecture to increase efficiency, resilience, and automation using smart contracts. From the discussion on RQ1, these are the key takeaways to be noted when using smart contracts to manage liquidity in a w-CBDC environment:

Table 9. Key Takeaways for Architecture Liquidity Management Recommendation

ID	Key Takeaways	Description
KT-1	Unified ledger integration	Smart contracts can be formed in unified ledgers where tokenized money and assets like w-CBDC token can live together. They use programmable logic to automate compliance and coordination between participants
KT-2	Programmability on Smart Contract for Risk Reduction	Programmability on smart contract makes atomic transactions possible. In these transactions, the exchange of assets between two parties is linked and only happens if both sides are successful. This is very important for lowering the risk of settlement in both domestic and cross-currency payments
KT-3	Embedded Business Rules in the Smart Contract	by embedding business rules like liquidity checks, conditional holds, and timed releases right into smart contracts it will automate settlement workflows and make them less dependent on intermediaries.
KT-4	AMM Smart Contract	Smart contracts can be designed as AMM that use algorithms to price and carry out on-demand swaps between different w-CBDCs, so there is no need for traditional order books.

ID	Key Takeaways	Description
KT-5	Gridlock Resolution and Deadlock Relief Using LSM and ALP	Gridlock resolution can be solved using LSM for netting across the queue transaction and an Automated Liquidity Provisioning (ALP) system for "just-in-time" deadlock relief to make sure the system stays strong.

Following the DSR design process described in the Methods section (Phase 2), each proposed module in the architecture is mapped to the key takeaways from the MLR that informed its design. This mapping ensures that every module is justified by evidence from the literature, and that every key takeaway is addressed by at least one module. Table 10 presents this traceability mapping.

Table 10. Traceability Mapping: Key Takeaways to Architecture Modules

Proposed Module	Module Type	Key Takeaway Addressed	Supporting Literature
Policy & Criteria	Smart contract	KT-1, KT-2,KT-3	GL2 [27], GL3 [28], WL1 [24], WL2 [25]
Centralised Queue	Smart contract	KT-1, KT-3	GL3 [28], GL6 [31], WL1 [24]
Liquidity-Saving Mechanism (LSM)	Automated system mechanism	KT-1, KT-2,KT-5	GL6 [31], WL1 [24]
Automated Liquidity Provisioning (ALP)	Automated system mechanism	KT-1, KT-2,KT-5	GL6 [31], WL1 [24]
Guarded AMM	Smart contract	KT-1, KT-4	GL4 [29], GL5 [30]

The proposed architecture is built on a unified ledger that functions as a programmable Financial Market Infrastructure (FMI). On this ledger, tokenized central bank money (w-CBDC) and other tokenized assets share the same data environment, execution layer, and governance structure. This follows the BIS vision of a future monetary system where

unified ledger platforms bring money and assets together to modernize existing processes and open new capabilities [26]. Drawing on the traceability mapping in Table 10, the proposed reference framework is instantiated as a five-module reference architecture, with each module derived through the DSR design cycle from the key takeaways. The sections below describe each module's core capabilities, key parameters, and governance assumptions.

Module 1: Policy & Criteria Smart Contract. This smart contract serves as the gateway of the architecture, encoding participant eligibility, prudential limits, and compliance rules on-ledger with a full audit trail consistent with CPMI-IOSCO PFMI standards [32]. Project Jura demonstrates how central banks can act as sole issuers of wholesale CBDC on a third-party platform while retaining control over participant access through dedicated subnetworks and notary nodes [33]. Central bank experiments such as Project Helvetia [9] and Denmark's National Bank [34] demonstrate the importance of placing policy and authorization logic close to the settlement layer, and peer-reviewed evidence confirms that smart contracts can enforce business rules deterministically for financial workflows [35]. Cross-domain peer-reviewed evidence further corroborates this pattern: Savelli et al. [25] show that smart-contract-enabled settlement in energy markets enforces eligibility and audit requirements with the same deterministic and traceable properties. Bank Indonesia, as the sole issuer and operator of wRD, retains exclusive authority over the participant registry and compliance parameters.

Module 2: Centralized Queue Smart Contract. When a payment meets all the policy checks but the payer does not have enough balance, the instruction is placed into the Centralized Queue contract. The contract keep track of each payment's priority, what is the timestamp, and policy details. It also manages action like enqueue, amend, cancel, and expire in predictable manner with everything logged on ledger for auditability. The way the queue is arranged in Project Garuda varies depending on the platform used. R3 Corda applies First In First Out (FIFO), while Hyperledger Besu applies First Available First Out (FAFO) [3]. The queue contract customizes its release policy to maintain the ordering dictated by the underlying ledger.

Module 3: Liquidity Saving Mechanism. When queued payments cannot be funded individually, the Liquidity-Saving Mechanism runs periodic cycles to identify multilateral

netting sets that are groups of queued payments that if settled together will require less total liquidity than settling them one by one. The approach follows the design documented in Project Inthanon Phase I, where the Bank of Thailand's LSM oracle identifies netting opportunities by evaluating queued items against available balances and bond token values [31]. When a valid netting set is found, settlement is executed atomically on ledger. The netting cycle interval and minimum set size are configurable parameters governed by Bank Indonesia.

Module 4: Automated Liquidity Provisioning. In situations where the LSM is unable to identify a feasible netting set, the ALP system is activated. ALP executes a rapid, seamless intraday repo transaction with Bank Indonesia, utilizing tokenized bonds as security. The system first verifies that the participating bank holds eligible bonds that have already been tokenized as bond tokens for repo purposes [31]. Once those conditions are satisfied, ALP injects just enough liquidity through the atomic repo to unblock the payment. The objective is to keep Bank Indonesia detached from routine transaction-level activities, allowing the system to independently manage liquidity shortages.

Module 5: Guarded AMM Smart Contract. This smart contract is activated only for cross-currency payments after domestic mechanisms have been exhausted. If the payment is in the same currency, it follows the normal path balance, queue, LSM, ALP without calling the AMM. For cross-currency cases, the system first checks whether the payer already holds the target w-CBDC; if so, settlement proceeds directly. If not, and only if guardrails are met (whitelisted participant, live bridge, acceptable rate and slippage, and sufficient pool depth), the AMM executes a one-step swap between the home currency w-CBDC and the target currency. Each guardrail addresses a specific risk. The slippage threshold blocks a swap when the exchange rate has moved too far, and the minimum pool depth blocks it when the liquidity pool is too thin to trade without distorting the price together these limit the price instability and manipulation that Rialto warns about. The whitelist limits who are allowed to do transaction, and the live-bridge check confirms the cross-network connection is working before the swap runs, which limits counterparty and operational risk. The AMM concept draws from Project Mariana [29] and Project Rialto [30], though Rialto warns that AMM-based foreign exchange can suffer price instability and manipulation when price discovery is inadequate [30]. The guardrail design is

intended to mitigate these risks, although threshold calibration remains a question for future testing.

The Table 11 below presents a comprehensive summary of the five modules, outlining the type of each module, their primary functions, essential parameters, and the governance responsibilities based on the architectural details mentioned earlier.

Table 11. Overview Propose Module

Module	Type	Core Capabilities	Key Parameters	Governance Owner
Policy & Criteria	Smart contract	Participant eligibility validation, prudential and position limit enforcement, compliance rule checking, full audit trail	Participant registry, position limits, supported currencies, compliance rules	Bank Indonesia
Centralised Queue	Smart contract	Priority-based queuing, deterministic enqueue, amend, cancel and expire actions, configurable release policy	Release policy mode (FIFO for R3 Corda / FAFO for Hyperledger Besu), priority schema, policy parameters	Bank Indonesia sets release policy; participants may amend own instructions within policy bounds
LSM	Automated system mechanism	Periodic netting cycle to identify multilateral netting sets, atomic netting settlement across queued payments	Netting cycle interval, minimum netting set size	Bank Indonesia
ALP	Automated system mechanism	Bond token collateral check, automated atomic intraday repo with Bank Indonesia, just-in-time liquidity injection	Eligible bond token types, repo conditions	Bank Indonesia
Guarded AMM	Smart contract	Guardrail verification, one-step cross-currency w-CBDC swap, liquidity pool management	FX participant whitelist, acceptable rate and slippage threshold, minimum pool depth, live bridge status	Bank Indonesia

Figure 1 illustrates the integration of the five modules together into a single architecture flow. It outlines the journey of a new payment instruction as it navigates through the Policy & Criteria contract, the Centralized Queue, the LSM, the ALP, and the secured AMM, traversing multiple decision-making points before arriving at the settlement.

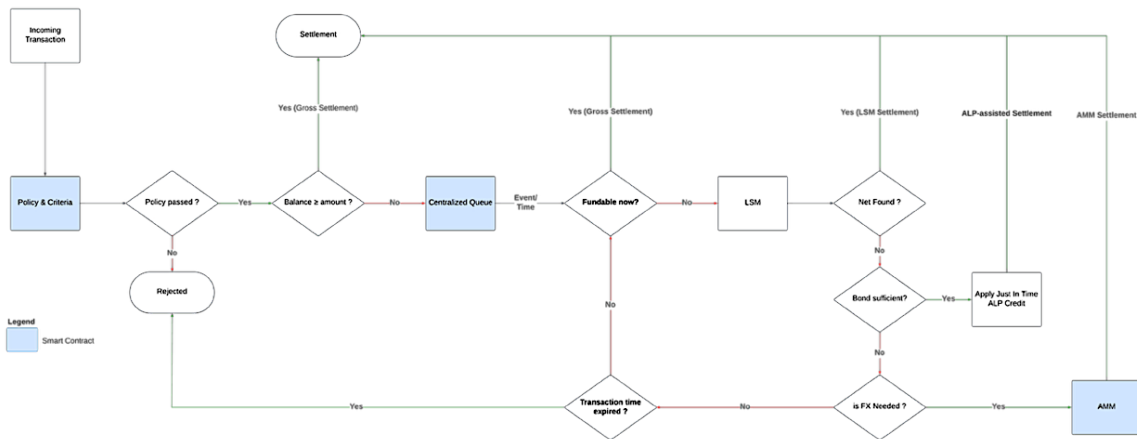


Figure 1. Propose Architecture Flow

To describe how the proposed architecture handles a payment from submission to final settlement, this study defines a payment state lifecycle as a complement to the architecture flow in Figure 1. While Figure 1 illustrates the processing logic and decision points across the five modules, Figure 2 presents the state diagram showing the discrete states a payment instruction can occupy and the conditions that trigger each transition.

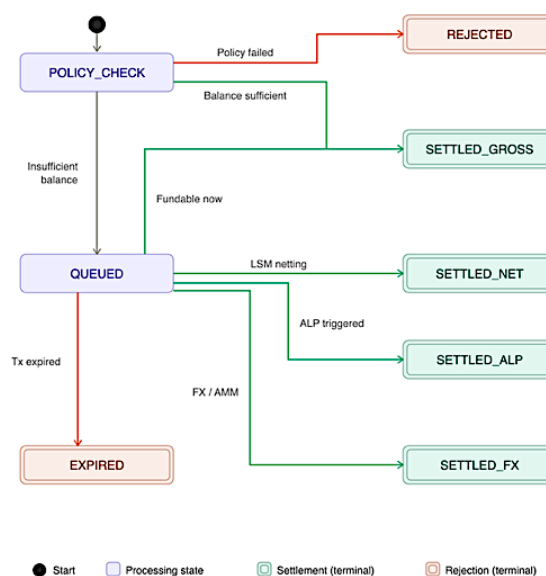


Figure 2. Payment State Lifecycle

The proposed states are derived directly from the modules, decision points, and terminals in Figure 1: POLICY_CHECK (instruction being validated by the Policy & Criteria contract), REJECTED (failed policy validation), SETTLED_GROSS (settled immediately when the payer holds sufficient balance, or when the payer becomes fundable from the queue), QUEUED (entered the Centralized Queue due to insufficient balance), SETTLED_NET (settled through LSM multilateral netting), SETTLED_ALP (settled after ALP injects just-in-time liquidity via intraday repo using bond tokens), SETTLED_FX (settled through the guarded AMM cross-currency swap), and EXPIRED (transaction time elapsed without settlement). The system attempts gross settlement first. Only when that is not possible does the payment enter the queue, where the LSM and ALP mechanisms attempt to resolve it. The AMM path is for cross-currency cases where domestic mechanisms alone are not sufficient.

3.3. Consideration

Due to most of the CBDC research still being in the research or pilot phase, it focuses more on sketches like blueprints and pilot experiments than finished regulatory. The reports for the major pilots always point out policy problems like non-normative status, unclear governance/upgrade rights, and legal basis for automated netting [9], [10], [28], [29], [31]. For a production rollout, the architecture should specify who is authorized to change parameters in the module (e.g., eligibility criteria, limits, queue release rules, and LSM/ALP thresholds), how smart contracts are upgraded, what emergency controls exist (pause mechanisms or circuit breakers), and how end-to-end auditability is achieved (immutable logs and monitoring) aligned with Bank Indonesia rules and Indonesian law.

3.4. Architecture Demonstration

To demonstrate how the proposed architecture operates in practice, two transaction scenarios are traced through the payment state lifecycle and the five modules.

Scenario 1 – Domestic Payment with LSM Netting.

Table 12. Initial w-CBDC Scenario 1

Participant	w-CBDC Balance
Bank A	200B w-CBDC
Bank B	300B w-CBDC
Bank C	250B w-CBDC

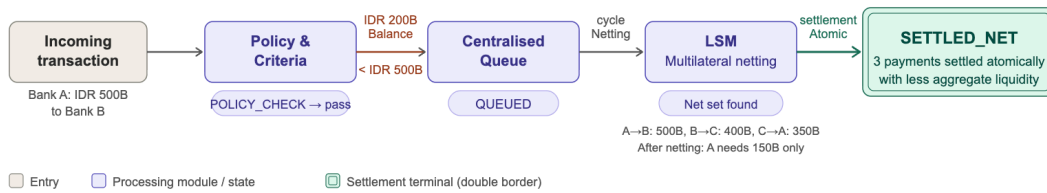


Figure 3. Scenario 1 LSM Netting

Table 12 shows the initial w-CBDC balances of the three participating banks. Bank A submits a payment of 500 billion w-CBDC to Bank B. The Policy & Criteria contract validates Bank A's eligibility and confirms the amount is within position limits (state: POLICY_CHECK → pass). Bank A's available balance is 200 billion w-CBDC which is insufficient for gross settlement, so the payment enters the Centralized Queue (state: QUEUED). During the next netting cycle, the LSM evaluates the queue and identifies that Bank B has a pending payment of 400 billion w-CBDC to Bank C, and Bank C has a pending payment of 350 billion w-CBDC to Bank A. As illustrated in Figure 3, the LSM calculates a multilateral netting set: after offsetting, Bank A needs to pay 150 billion w-CBDC net, Bank B ends with a 100 billion w-CBDC net inflow, and Bank C ends with a 50 billion w-CBDC net inflow. Since all participants hold sufficient balances to cover the netted amounts, the LSM executes the netting set atomically (state: SETTLED_NET). In gross terms, the three payments total 1,250 billion w-CBDC yet none can settle on its own, Bank A holds only 200 billion against the 500 billion it owes Bank B, and Banks B and C are likewise short of their gross obligations. Under gross settlement alone the three would deadlock in the queue, each waiting on liquidity that only another queued payment can release. The LSM clears all three in a single cycle by debiting only net positions, which is exactly the gridlock the module exists to resolve.

Scenario 2 – Cross-Currency Payment with ALP and AMM.

Table 13. Initial w-CBDC Scenario 2

Participant	w-CBDC Balance	USD w-CBDC Balance	Tokenised bond collateral
Bank D	300B w-CBDC	0 (none held)	200B w-CBDC equiv. (eligible)
Bank E	-	Overseas counterpart	-

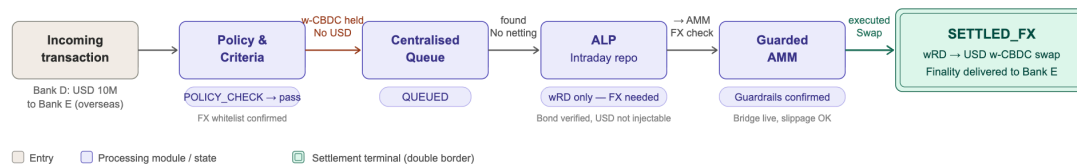


Figure 4. Scenario 2 Cross Currency

Table 13 shows the initial state for this scenario. Bank D submits a payment of 10 million USD equivalent to an overseas counterpart, Bank E. The Policy & Criteria contract validates Bank D's eligibility and confirms Bank D is whitelisted for cross-currency operations (state: POLICY_CHECK → pass). Bank D does not hold sufficient USD denominated w-CBDC, so the payment enters the queue (state: QUEUED). The LSM finds no netting opportunity for this instruction. The system triggers ALP where Bank D holds tokenized government bonds, and the ALP verifies collateral eligibility. However, the required USD liquidity cannot be sourced domestically through repo ALP only injects wRD, not foreign-currency w-CBDC. As traced in Figure 4, the system then checks whether the payment requires foreign currency and activates the AMM pathway. The guardrail checks in the AMM smart contract confirm that a live bridge exists, the current exchange rate is within the slippage threshold, and the pool has sufficient depth. Bank D's w-CBDC is bridged to the international network and swapped for USD w-CBDC through the AMM liquidity pool. The acquired USD w-CBDC is used to settle the payment to Bank E with finality (state: SETTLED_FX). In this scenario, the domestic settlement mechanisms (balance, queue, LSM, and ALP) are attempted first but cannot resolve the obligation, since the shortfall is in foreign currency rather than in domestic liquidity. Settlement is therefore completed by the Guarded AMM, which performs the cross-currency conversion by swapping Bank D's wRD into USD w-CBDC. This demonstrates the function of the AMM as the architecture's cross-currency settlement path, invoked only after the domestic mechanisms have been exhausted.

3.5. Architecture Evaluation

Following the DSR evaluation approach described in the Methods section (Phase 3), the proposed architecture is assessed against the five key design takeaways. The evaluation confirms that all five design criteria are addressed by the proposed architecture. Each key takeaway is traceable from the MLR findings (Table 9) through the traceability mapping (Table 10) to at least one architectural module (Table 11). These modules are then

integrated into the single settlement flow shown in Figure 1, so the chain runs unbroken from the literature-derived takeaways to the complete architecture, completing the DSR design cycle. This evaluation is analytical in scope. It verifies design completeness and traceability between literature evidence and architectural modules, not operational performance. The architecture has not yet been validated through prototype implementation, simulation, or expert review. Four questions remain open at this stage.

Governance rights are framed at the module level but not specified procedurally. The architecture names Bank Indonesia as the governance owner across all five modules, but it does not define the workflow for parameter changes. Values such as LSM netting-cycle frequency, ALP collateral haircuts, AMM slippage thresholds, and Centralized Queue release-policy mode are configurable. The process for proposing, reviewing, approving, and auditing changes to them is not defined.

Legal enforceability of automated netting, conditional settlement, and intraday repo under Indonesian law has not been confirmed. The architecture relies on automated operations that execute without manual intermediation, and whether these satisfy Indonesian payment-system law, repo regulations, and central-bank requirements for settlement finality requires a formal legal opinion before production deployment.

Upgrade control for the smart contracts has not been specified. The architecture does not yet define how deployed contracts will be versioned, how state will be migrated when contracts are replaced, what authorization the upgrade process requires, or how an emergency pause would be invoked if a contract behaves incorrectly.

Empirical performance under realistic transaction volumes cannot be assessed without simulation or prototype. The architecture's effectiveness depends on quantities that have not been measured, including how often netting sets of meaningful size form in the queue, how much aggregate liquidity is saved per netting cycle, how often ALP is triggered, and how often the AMM is invoked for cross-currency obligations. These require simulation against representative Indonesian payment-flow data or prototype deployment on Project Garuda's existing test platforms.

3.6. Discussion

The architecture decomposes into five modules because each handles a distinct stage in the payment lifecycle, and keeping the stages separate makes each module's logic and governance scope clear. Most of these splits follow obvious functional lines; the less obvious separation is between LSM and ALP. The LSM works across the queue, reducing aggregate liquidity demand by netting offsetting payments across many banks at once. The ALP works on individual banks, supplying intraday liquidity to a specific bank by exchanging its tokenized bonds for wRD.

The Guarded AMM is the only module that handles cross-currency conversion, and the only one that introduces external counterparty exposure since every swap touches a foreign-currency liquidity pool whose price discovery is outside Bank Indonesia's direct control. The queue, LSM, and ALP resolve a domestic liquidity shortfall, where a participant does not hold enough of the settlement currency and netting or an intraday repo can supply it. This reflects a distinction in the kind of problem each path solves. Project Rialto's warning about price instability and manipulation risk in AMM-based foreign exchange [30] motivates the guardrails on the module, which include the whitelist, slippage threshold, minimum pool depth, and live bridge check. The same concern shapes the decision to treat the AMM as a last resort. Domestic settlement through balance, queue, LSM, and ALP carries lower operational and price risk than a cross-currency swap, so the architecture exhausts those paths first and invokes the AMM only when none can resolve the obligation.

Project Garuda's current architecture uses gross settlement with decentralized queueing [3]. The proposed architecture keeps the gross-settlement default but adds capabilities that the PoC does not have. The LSM enables multilateral netting across queued payments, reducing the total liquidity required to clear the queue when offsetting flows exist (as illustrated in Scenario 1). Multilateral netting only works when all queued payments can be seen together, so the LSM can find sets that offset across different participants. A decentralized queue cannot support this, because each participant holds only its own payments. The architecture therefore uses a single central queue contract that holds the whole queue. Centralizing the queue means Bank Indonesia, as operator, has visibility and release control over all queued payments, instead of each participant controlling its own. The ALP introduces an automated intraday repo path against

tokenized bonds, removing the need for manual treasury intervention when gridlock occurs. The Guarded AMM extends settlement to cross-currency obligations, which is necessary for wRD to participate in the regional and cross-border w-CBDC payment landscape that other central banks are already piloting.

4. CONCLUSION

This paper adopted a Design Science Research (DSR) approach with a Multivocal Literature Review (MLR) to design a literature-derived reference architecture for smart-contract-enabled liquidity management in Indonesia's Wholesale Rupiah Digital (wRD), addressing a gap in Bank Indonesia's Project Garuda roadmap. The MLR synthesized eight academic and grey literature sources spanning global w-CBDC projects, from which five key design takeaways were extracted. These takeaways were mapped through a traceable design process into a five-module reference architecture comprising Policy & Criteria, a centralized Queue, the LSM, the ALP, and a guarded AMM, all operating on a unified ledger. The architecture is proposed at the reference-architecture level and has not yet been validated through prototype implementation, simulation, or expert review. The MLR scope is bounded to 2019–2025 across selected databases and search strings, so some relevant studies may have been missed. Grey literature also varies in quality, and the 3C+2S synthesis involves interpretive judgment in source selection. Future work should focus on empirical validation of the framework. Prototype implementation on Project Garuda's existing DLT platforms (R3 Corda and Hyperledger Besu) would test whether the modules can be encoded within each platform's constraints. Simulation of LSM netting efficiency and ALP utilization under realistic Indonesian payment-flow patterns would provide empirical grounding for the parameter choices. The governance, legal, and upgrade-control questions raised in Architecture Evaluation also remain open.

ACKNOWLEDGMENT

This research is funded by Directorate of Research Funding and Ecosystem, Universitas Indonesia under Hibah PUTI 2025 (Grant No. PKS-315/UN2.RST/HKP.05.00/2025). The authors acknowledge the use of Gemini for language editing, grammar enhancement, and structural suggestions during the drafting of this manuscript. The AI system was strictly utilized as an editorial aid and was not used to generate the literature searching,

literature synthesis, and the architectural suggestion for article literature review process. The authors retain full responsibility for the original ideas, literature interpretation, and the final content of this article.

REFERENCES

- [1] Bank for International Settlements, "CBDCs: an opportunity for the monetary system," Switzerland, Jun. 2021.
- [2] Committee on Payments and Market Infrastructures, "Distributed ledger technology in payment, clearing and settlement: an analytical framework," Switzerland, Feb. 2017. Accessed: Apr. 14, 2026.
- [3] S. Kuncoro *et al.*, "Proof of concept (PoC) report: Project Garuda wholesale Rupiah digital cash ledger," Jakarta, Dec. 2024.
- [4] Banque de France, "Experimenting settlement of French government bonds in central bank digital currency with blockchain technology," Paris, Oct. 2021.
- [5] S. Grybniak, O. Nashyvan, I. Mazurok, Y. Leonchyk, and A. Vorokhta, "Recurring and Deferred Transactions Based on Smart Contracts," *IET Blockchain*, vol. 5, no. 1, Jun. 2025, doi: 10.1049/blc2.70012.
- [6] Bank Indonesia, "Proyek Garuda: Wholesale Rupiah digital cash ledger," Jan. 2023.
- [7] N. Atzei, M. Bartoletti, and T. Cimoli, "A Survey of Attacks on Ethereum Smart Contracts (SoK)," in *Principles of Security and Trust*, Maffei Matteo and M. and Ryan, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 164–186.
- [8] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 49, no. 11, pp. 2266–2277, 2019, doi: 10.1109/TSMC.2019.2895123.
- [9] BIS Innovation Hub, "Project Helvetia Phase II: Settling tokenised assets in wholesale CBDC," Basel, Switzerland, Jan. 2022.
- [10] BIS Innovation Hub, "Project mBridge: Connecting economies through CBDC," Basel, Switzerland, Oct. 2022.
- [11] Monetary Authority of Singapore and Association of Banks in Singapore, "Project Ubin phase 2: Re-imagining interbank real-time gross settlement system using distributed ledger technologies," Singapore, Nov. 2017.
- [12] European Central Bank, "Payment systems: liquidity saving mechanisms in a distributed ledger environment," Frankfurt am Main, Germany, Sep. 2017.

- [13] V. Garousi, M. Felderer, and M. V Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Inf. Softw. Technol.*, vol. 106, pp. 101–121, 2019, doi: 10.1016/j.infsof.2018.09.006.
- [14] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, pp. 45–77, May 2007.
- [15] A. Hevner *et al.*, "Design Science in Information Systems Research," *Management Information Systems Quarterly*, vol. 28, p. 75, May 2004.
- [16] D. Giustini and D. Thompson, "Finding the Hard to Finds: Searching for Grey Literature," Vancouver, BC, Canada, May 2006.
- [17] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," vol. 2, May 2007.
- [18] Y. Xiao and M. Watson, "Guidance on Conducting a Systematic Literature Review," *J. Plan. Educ. Res.*, vol. 39, p. 0739456X1772397, May 2017, doi: 10.1177/0739456X17723971.
- [19] C. Okoli, "A Guide to Conducting a Standalone Systematic Literature Review," *Communications of the Association for Information Systems*, vol. 37, May 2015, doi: 10.17705/1CAIS.03743.
- [20] A. Garg, R. Kumar Kaliyar, and A. Goswami, "PDRSD-A systematic review on plan-driven SDLC models for software development," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2022, pp. 739–744. doi: 10.1109/ICACCS54159.2022.9785261.
- [21] M. Themistocleous, P. Rupino da Cunha, E. Tabakis, and M. Papadaki, "Towards cross-border CBDC interoperability: insights from a multivocal literature review," *Journal of Enterprise Information Management*, vol. 36, no. 5, pp. 1296–1318, Jul. 2023, doi: 10.1108/JEIM-11-2022-0411.
- [22] E. Carvalho Silva and M. Mira da Silva, "Central Bank Digital Currency: A Multivocal Literature Review," *Journal of Internet and Digital Economics*, vol. 5, no. 1, pp. 51–69, Mar. 2025, doi: 10.1108/JIDE-03-2024-0013.
- [23] Sugiyono, *Metode penelitian kuantitatif, kualitatif, dan R&D*, 19th ed., vol. 1. Bandung, Indonesia: Alfabeta, 2013.
- [24] E. A. Opare and K. Kim, "A Compendium of Practices for Central Bank Digital Currencies for Multinational Financial Infrastructures," *IEEE Access*, vol. 8, pp. 110810–110847, 2020, doi: 10.1109/ACCESS.2020.3001970.

- [25] I. Savelli, C. Hepburn, and T. Morstyn, "A blueprint for energy systems in the era of central bank digital currencies," *Technol. Forecast. Soc. Change*, vol. 207, p. 123637, 2024, doi: 10.1016/j.techfore.2024.123637.
- [26] Bank for International Settlements, "Annual economic report 2023," Basel, Switzerland, Jun. 2023.
- [27] Bank of England, "Central bank digital currency: Opportunities, challenges and design," London, UK, Mar. 2020.
- [28] Bank for International Settlements, "Project Dunbar: International settlements using multi-CBDCs," Basel, Switzerland, Mar. 2022.
- [29] Bank for International Settlements, "Project Mariana: Cross-border exchange of wholesale CBDCs using automated market-makers," Basel, Switzerland, Mar. 2023.
- [30] Bank for International Settlements, "Project Rialto: Improving instant cross-border payments using central bank money settlement," Feb. 2025.
- [31] Bank of Thailand, "Inthanon phase I: An application of distributed ledger technology for a decentralised real time gross settlement system using wholesale central bank digital currency," Bangkok, Thailand, May 2019.
- [32] Bank for International Settlements, "Principles for financial market infrastructures," Bank for International Settlements, Basel, Switzerland, Apr. 2012.
- [33] Banque de France, Bank for International Settlements, and Swiss National Bank, "Project Jura: Cross-border settlement using wholesale CBDC," Sep. 2021.
- [34] H. Hassan, J. Feveile Adolfsen, M. Rieland Jakobsen, R. B. Søndergaard, and S. Eddie Mogensen, "New technology and settlement in central bank money between banks," Copenhagen, Denmark, Jun. 2025.
- [35] T. Hewa, M. Ylianttila, and M. Liyanage, "Survey on blockchain based smart contracts: Applications, opportunities and challenges," *Journal of Network and Computer Applications*, vol. 177, p. 102857, 2021, doi: 10.1016/j.jnca.2020.102857.