International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V74I10P102

Volume 74 Issue 10, 10-22, October 2026
© 2026 Seventh Sense Research Group®

Original Article

From Batch to Instant: A Reference Architecture and Process Model for Modernizing National-Scale Disbursement Platforms with Real-Time, Multi-Rail Payments

Chandrashekhar Kharche

Commercial Payment Systems & Financial, Leander, TX, USA.

Corresponding Author : Cpskharche@gmail.com

Received: 18 August 2026 Revised: 21 September 2026 Accepted: 09 October 2026 Published: 29 October 2026

Abstract *- For a long time, large-scale disbursement of funds relied on overnight processing and checks. However, the emergence of instant payment systems like the FedNow Service from the Federal Reserve and the RTP network from The Clearing House, as well as push-to-card and digital wallet payment systems, allows for the funds to be delivered to the recipients in seconds [2][3]. Furthermore, recipients of these funds have demonstrated a strong preference for instant delivery when it is an available option [7]. However, the platforms that carry national disbursement volume were built for batch processing and cannot be taken out of service for rebuilding. This paper is a conceptual study. The aim of this study is to propose a reference architecture and a process model to enable the migration of a national-scale disbursement platform from 24-hour batch processing to near-instant multi-rail payouts while ensuring the operations of the existing platform are not interrupted. The proposed architecture introduces a layer for disbursement orchestration and rail routing that selects the fastest, least-cost eligible rail for each payout while ensuring that any fraud, anti-money-laundering, and sanctions controls are performed on each payout before it is sent. If a payment fails, the system automatically routes the payment through an alternative rail after confirming the failure of the original payment to avoid duplicate payments. A phased, reversible migration is proposed for the existing platform to the event-driven system. The proposed architecture and migration plan are evaluated using a comparison with existing literature and a scenario-based walkthrough against the proposed requirements for the platform. As this is a conceptual study, no experiments were conducted to evaluate the proposed architecture, migration plan, and requirements. The benefits of the proposed platform have been reported alongside a classification of the evidence supporting these benefits as either published data, practitioner observations, or assumptions and estimates, with empirical measurement left for future work.*

Keywords *- Instant payments, Real-time payments, Disbursements, FedNow, RTP, Digital wallets, Push-to-card, Legacy modernization, Event-driven architecture, ISO 20022, Financial inclusion, Reference architecture.*

1. Introduction

Disbursing funds to a large population is a different task from collecting funds from the public. Whether it is a government agency making benefit payments to the public, a tax authority issuing tax refunds to citizens, or insurance companies settling the insurance claims of their policyholders, there are a large number of individuals to whom funds are to be disbursed.

In the past, the model for making these disbursements was to run a batch payment system at night, where the list of individuals to be paid was gathered during the day, and the payments were released overnight, with the funds reaching the recipients the next day or in the next few days. While this model is robust, it is also very slow, causing hardship for the recipients of these funds and increasing the workload for the agencies to reissue the checks that were lost or stolen from the recipients.

The way in which payments are made has undergone a transformation in recent years. In 2017, the RTP network was introduced in the United States, providing a means for payments to be made in real time between financial institutions [2][3]. In 2023, the FedNow Service was launched, which would serve as a second, government-operated real-time settlement network for the country [2][3][5]. Both of these networks would allow for the near-instantaneous settlement of individual credit payments between financial institutions [3][5]. Additionally, both networks allow for higher transaction limits to accommodate

 This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

large payments [3][5]. The government has begun to implement policies to take advantage of the benefits of real-time payments. For example, the U.S. Treasury has launched the Digital Payout Program to begin distributing disaster relief funds using the FedNow network [6]. Furthermore, Executive Order 14247 mandates that the government move its disbursement of funds away from paper checks and toward electronic payments [35]. Surveys have shown that approximately 72% of payment recipients prefer the use of instant payments over other forms of payment [7]. Finally, the success of other countries with real-time payments, such as Brazil and India, demonstrates the potential for these payment networks to scale [1][21].

The main problem is the disbursement platform, not the rails. The research problem that will be examined in this paper is the following: how can the disbursement platform that was built for overnight batch processing be altered while in continuous operation so that each payment can be made using the fastest eligible rail among the various instant and non-instant payment rails while ensuring that each payment is subjected to the appropriate risk and compliance controls before being released and that no payment is lost or made twice? The three components of this problem are as follows: What payment rail should be selected for each payment? How can the screening of payments be altered from being performed after the batch to inline to accommodate instant payments? What approach can be taken to ensure that no payments are lost or made twice in the case of failure of any payment?

Existing work addresses these parts separately. Studies of fast payment systems describe the design and adoption of the rails themselves [1][19][20][22]. The software engineering literature describes how monolithic and batch systems are decomposed and migrated in general [10][18][23][24]. Work on payment platform modernization concentrates on moving whole systems to cloud-native architectures [12][13]. As the structured comparison in Section 2.5 shows, none of the reviewed sources jointly treats per-payout routing across multiple rails, inline compliance, duplicate-safe fallback, and coexistence with a legacy batch pipeline for the disbursement use case. This combination is the research gap that the paper addresses.

This paper has four research objectives: first, to specify the requirements of a real-time, multi-rail disbursement platform; second, to specify the reference architecture and the disbursement process model of such a platform; third, to define a migration strategy from batch to real-time disbursement; and fourth, to assess the design of the platform and to state its expected benefits while separating evidence from assumptions.

The contributions correspond to these objectives. The first is a reference architecture organized around a disbursement orchestration and rail-routing layer (Section 5). The second is a seven-stage process model with a payout state model and rules for duplicate-safe fallback (Section 6). The third is a strangler-style migration strategy in which the legacy batch pipeline remains the safety net (Section 7). The fourth is an analytical evaluation together with an evidence-classified statement of expected benefits (Sections 8 and 9). The novelty lies in the integration of these elements for the disbursement domain and not in any individual pattern, each of which is established in prior work. The paper is a conceptual, reference architecture study. No prototype was built, and no experiment was run; every quantitative figure is a published benchmark, a practitioner-reported observation, an assumption, or a model estimate, and is labeled as such.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 describes the research methodology. Section 4 presents the problem formulation. Sections 5 to 7 describe the architecture, the process model, and the migration strategy. Section 8 presents the analytical evaluation of the design. Section 9 discusses the expected benefits. Section 10 discusses the findings. Section 11 presents the limitations of the research and the proposed future work. Finally, Section 12 concludes the paper.

2. Background and Related Work

2.1. The Rise of Instant and Real-Time Payment Rails

Instant payment rails transfer funds in a few seconds and are always in operation, unlike the batch payments executed through the Automated Clearing House (ACH) and wire systems that operate during business hours. The Committee on Payments and Market Infrastructures defines a fast payment as one that allows for the funds to be transferred from the payer to the payee in real time, or as close to real time as possible, 24 hours a day and seven days a week [19]. In the United States, the RTP network was launched in 2017 to facilitate real-time payments, followed by the launch of FedNow in 2023. Both networks have increased the transaction limits to accommodate larger payments [2][3][5].

Fast payment networks have been widely adopted worldwide. For instance, fast payments in Thailand, Brazil, and South Korea amounted to 35, 27, and 21 transactions per person per month, respectively, in 2022. The most successful fast payment networks are those in which the central bank owns the network and non-bank payment providers can participate in the payment system [1][22]. Similar conclusions can be drawn from the case study of Brazil's Pix system [21]. The World Bank has studied the development and adoption of fast payment networks in a variety of jurisdictions worldwide [20]. While these publications explain the development and adoption of fast payment networks, they do not explain how a platform can select from among the various available fast payment networks for making an individual payment. Due to the lack of interoperability between the two fast payment networks available in the U.S., it is the platform that must

select from among the available payment networks to make an individual payment [16].

2.2. Digital-Wallet and Push-to-Card Disbursement Channels

Instant account-to-account rails are not the only route to a fast payout. Push-to-card services deliver funds to a recipient's debit card through the card networks, and digital wallets allow funds to be credited to a wallet balance in near real time [17]. These channels matter for reach because they serve recipients whose bank does not yet participate in an instant rail, recipients with limited access to banking, and recipients who prefer a wallet.

The Global Findex evidence indicates that moving government-to-person payments to digital channels is associated with wider account ownership and greater financial resilience among recipients [14]. For a disbursement platform, cards and wallets therefore complement FedNow and RTP: together they form a set of rails from which the most suitable one can be selected for each recipient. They also bring distinct obligations, notably the Payment Card Industry Data Security Standard (PCI DSS) for any component that stores, processes, or transmits card data [9].

2.3. Legacy Batch Disbursement Systems and Their Modernization

The modernization of batch platforms is well studied in software engineering. Monolithic systems are commonly decomposed into independently deployable services [10][23], and an empirical investigation of such migrations reports that maintainability and scalability are the main motivations, while decoupling the monolith and splitting its data are the main difficulties [24]. Batch processing is replaced by stream processing, in which each event is handled as it arrives [11], and messaging patterns provide the integration between old and new components [25]. The strangler fig pattern replaces a legacy system incrementally, by routing a growing share of traffic to new components while the old system continues to run [18].

In the payments domain, published work describes the migration of legacy payment systems to cloud-native architectures through phased, controlled cutovers [12] and the decomposition of a global payment platform into microservices [13]. The ISO 20022 standard supplies the structured messages on which instant rails operate [8]. This body of work treats migration at the level of the whole system. It pays little attention to which rail should carry an individual payout or how a legacy batch file can be served by instant rails during the transition.

2.4. Reliability, Duplicate Prevention, and Inline Risk Controls

Two further strands of work bear on the problem. The first concerns reliability in distributed transaction processing. A payout crosses several services and an external rail, so it cannot be protected by a single atomic transaction; long-lived transactions are instead structured as sagas, that is, sequences of local steps with compensating actions [26][28]. Message delivery in such systems is at least once, so operations must be idempotent if a retry is not to repeat its effect [27], and stability patterns such as timeouts and circuit breakers are used to contain the failure of a remote dependency [29]. These principles are general. Their application to irrevocable payments, where a compensating action may not be available once funds have settled, requires specific treatment.

The second strand concerns risk and compliance. With instant settlement, it is not possible to stop a suspicious batch payment, so the screening must take place before the payment is dispatched. OFAC has published guidance on a risk-based approach for instant payment systems [34], and a survey of machine learning and deep learning models for detecting financial fraud and the challenges of deploying them is available [15]. Security controls such as NIST SP 800-53 [37] and PCI DSS [9] specify the controls that must be implemented for an organization, but not where they should be in the disbursement flow.

2.5. Comparison with Existing Work and Research Gap

Table 1 compares the literature on the subject to the current research in terms of the six aspects required by the research problem: the coverage of instant rails, the routing of payouts across multiple rails, the ability of the solution to coexist with and migrate from a legacy batch payment system, the risk and compliance controls that are applied inline prior to the disbursement of the payment, the handling of payment failures to prevent the duplication of payments, and the focus of the solutions on disbursement as the primary use case.

The aspects used for this comparison were drawn from the literature on instant payments, disbursement, and the modernization of legacy payment systems. Table 1, therefore, provides a comparison of the literature in relation to the current research, but does not provide an exhaustive systematic review of that literature.

While each of the aspects of the disbursement process is addressed by at least one of the sources reviewed for this paper, there is no source that addresses all of the aspects of the disbursement process. Therefore, the gap in the literature is that there is no source that describes a reference architecture and a process model that addresses the disbursement of funds to customers through FedNow, RTP, push-to-card, wallets, and ACH payment rails, while also ensuring that there are no duplicate payments in the event that one of the payment rails fails, and while the legacy batch payment system is also still in service. This gap in the literature is limited to the literature that was reviewed for this paper and to the disbursement use case.

Table 1. Comparison of the reviewed literature with the present work

Literature strand (sources)	Instant rails	Per-payout routing	Batch migration	Inline controls	Duplicate safety	Payout focus
Fast payment system design and adoption [1][19][20][21][22]	Yes	No	No	Partial	No	Partial
U.S. instant rails: operator and central bank publications [2][3][4][5][16]	Yes	No	No	No	No	Partial
Digital government-to-person payments, cards, and wallets [6][14][17]	Partial	No	No	No	No	Yes
Legacy and microservices migration [10][11][18][23][24][25]	No	No	Yes	No	Partial	No
Payment platform modernization [12][13]	Partial	No	Yes	Partial	No	No
Reliability in distributed transactions [26][27][28][29]	No	No	No	No	Yes	No
Risk, sanctions, and security controls [9][15][34][37]	Partial	No	No	Yes	No	No
This paper (conceptual; not experimentally validated)	Yes	Yes	Yes	Yes	Yes	Yes

Yes: the aspect is a focus of the sources. Partial: the aspect is mentioned but not developed. No: the aspect is No: the aspect is not addressed. Assessment by the author, based on the cited sources, based on the cited sources.

3. Research Methodology

The study follows a design science approach, in which an artifact is designed to address an identified problem and is then evaluated [30][31]. The artifacts are a reference architecture, a process model, and a migration strategy. The work proceeded in five steps. First, the problem was identified and motivated from the literature reviewed in Section 2 and from the author's professional experience with large-scale disbursement platforms. Second, the objectives of a solution were expressed as four requirements (Section 4). Third, the artifacts were designed by applying established architectural patterns to those requirements (Sections 5 to 7). Fourth, the design was evaluated analytically (Section 8). Fifth, the expected benefits were derived and classified by type of evidence (Section 9).

3.1. Evaluation Approach

Design science distinguishes several forms of evaluation, among them observational, experimental, analytical, and descriptive methods [30]. This study uses only analytical and descriptive evaluation: a structured comparison with existing work (Table 1), a scenario-based walkthrough in the manner of scenario-based architecture evaluation [32][33] (Table 2), and a reasoned derivation of the expected benefits (Section 9). No prototype, simulation, or production measurement was performed. The paper, therefore, does not claim to demonstrate that the architecture achieves a given level of performance. It claims that the design is consistent with the stated requirements, and it identifies the measurements that would be needed to test it.

3.2. Evidence Classification

Every numerical value in this paper is classified into one of six classes. A measured result is a value directly obtained from the measurements performed in this study; no value in this paper falls into this class. A published benchmark is a value obtained from a published, external source. A practitioner-reported observation is a value of which the author is aware from his experience with similar types of deployments; it is not a result of this study.

An assumption is a value used as an input for the purposes of this study. A model estimate is a value calculated based on the assumptions and benchmarks provided. Finally, an expected outcome is a value that is expected to be produced as a result of the design, but which depends upon factors outside of the design itself.

3.3. Development of the Figures

The author prepared Figures 1 and 2 for this paper based on the requirements described in Section 4. They are not adapted from any published source. These figures have been

examined against the requirements described in Section 8 through the scenario walkthrough presented there, but have not been validated by implementation.

4. Problem Formulation

This paper examines a disbursement platform that moves billions of dollars annually. The disbursement platform must move from overnight batch processing to near-instant, multi-rail payouts. Four requirements result from the research problem. R1, continuous operation, indicates that the disbursement platform cannot be taken out of service to perform updates, and every change must be small and reversible [12][18]. R2, multiple rails, indicates that no single rail reaches all recipients, requiring the disbursement platform to support several rails and provide an alternative should one rail be unavailable [16]. R3, real-time risk controls, indicates that instant payments are settled within seconds, meaning that fraud, anti-money-laundering, and sanctions checks must be performed before the payment is sent out [34]. R4, reach and inclusion, indicates that the platform must reach as many recipients as possible, including those with limited access to banking services [14]. The routing decision can be stated formally. Let R be the set of rails connected to the platform, and let p be a payout of amount a_p to a given payee. The set of eligible rails is

$$E(p) = \{r \in R : c_r(p) = 1, a_p \le L_r, u_r = 1\} \quad (1)$$

where $c_r(p) = 1$ if the payee holds a credential that can be reached on rail r, L_r is the transaction limit of the rail, and $u_r = 1$ if the rail is currently available. The selected rail minimizes a weighted combination of the expected time to funds t_r and the unit cost k_r:

$$r^*(p) = \arg\min_{r \in E(p)} (w_t t_r + w_k k_r) \quad (2)$$

The weights w_t and w_k are set by the policy of the disbursement program. If $E(p)$ contains no instant rail, the payout is sent by ACH, and the check path remains only for payees with no electronic credentials. Equations (1) and (2) specify the routing rule. They were not evaluated numerically in this study.

5. Reference Architecture of a Real-Time, Multi-Rail Disbursement Hub

Figure 1 shows the architecture as a set of layers around a central orchestration and rail-routing hub, which separates the originator's request from the rails that can fulfill it. The components are described in Sections 5.1 to 5.7.

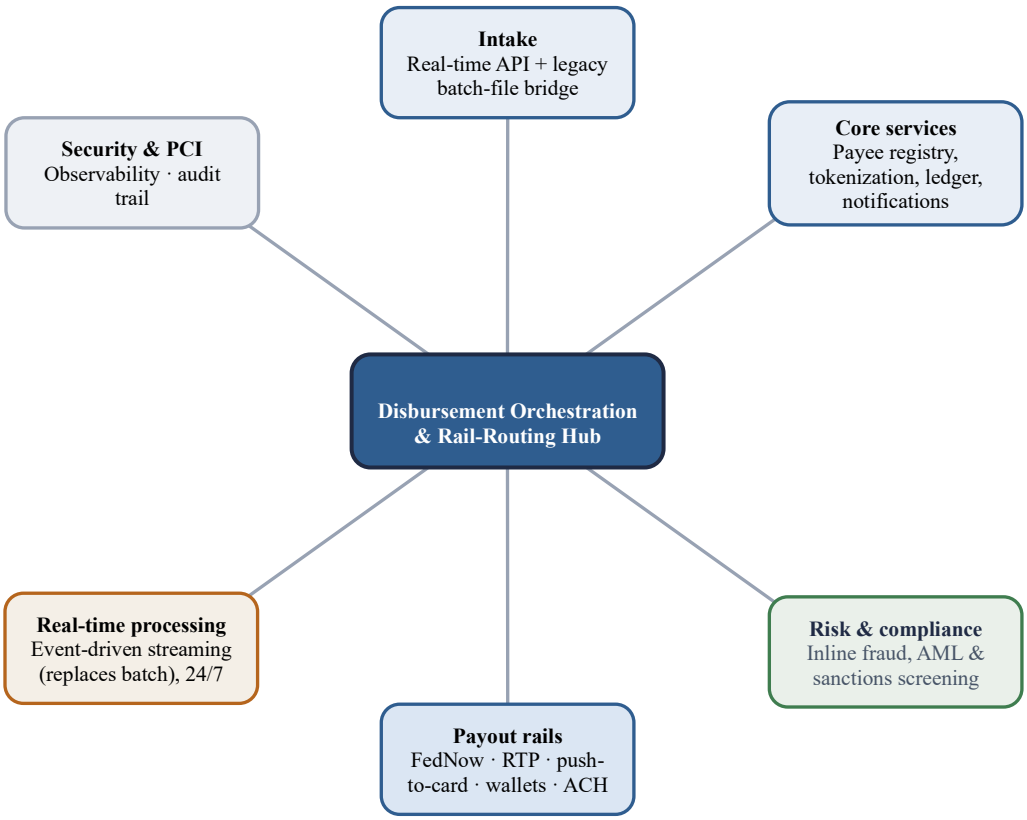


Fig. 1 Reference architecture for a real-time, multi-rail disbursement hub. Originators feed a central orchestration-and-routing core that selects the fastest, least-cost eligible rail per payout with status-confirmed fallback; risk and compliance run inline before dispatch, over a shared cross-cutting base

5.1. Intake Layer

The hub has two main ways to accept work: from new originators via an API, or through a bridge that takes in the existing disbursement files during the transition. The bridge splits each file into individual payout events, with each payout having a unique identifier based on the originator's own reference. This prevents the same file from creating two payouts for the same originator. The bridge is the main way the hub can work alongside the existing pipeline during the transition.

5.2. Disbursement Orchestration and Rail Routing

This layer is the heart of the architecture. For each payout, it determines the eligible rails and selects one of the eligible rails according to Equations (1) and (2). This layer also maintains the state of each payout, tracks the outcome of the payout attempts, and applies the payout failure rules described in Section 6.1 if the payout attempt does not succeed.

The originators interact only with this payout hub layer, which allows for the rails to be added, removed, or have their priority changed without the originators having to make any changes to their systems. This layer allows for the rails to be treated as a single disbursement service.

5.3. Core Services

The hub is supported by services that can be deployed and scaled independently [10]: a payee registry that holds enrolled accounts, cards, and wallet identifiers; a tokenization vault that isolates sensitive account and card data [9]; a ledger service that records every movement of funds; and a notification service that informs recipients of the status of their payout. Building these as separate services is what makes an incremental, reversible migration possible.

5.4. Real-Time Processing

The central change is the replacement of the nightly batch by an event-driven stream processor that handles each payout as it arrives, at any hour [11]. Payout events are written to a durable log before they are processed, so that work in progress survives the failure of a component and can be replayed.

Since delivery from such a log is at least once, every consumer is designed to be idempotent [27]. The expected effect is that processing delay falls from the length of the batch cycle to the time needed to process a single event.

5.5. Risk and Compliance

Since instant payouts are final, fraud, AML, and sanctions checks are run before the money is sent. These checks consist of four steps. Payee verification ensures that the destination account belongs to the enrolled payee and that their details have not changed recently. Sanctions screening checks the payee and beneficiary details against current sanctions lists [34]. Fraud scoring checks the transaction against the payee's transaction history and overall program transaction patterns [15]. Finally, there are limits on transaction amounts and the frequency of transactions for each payee, credential, and program.

Every check returns one of three outcomes - pass, hold, or reject - within the time allotted by the program. If a payout is held, it is removed from the payment queue and placed in a review queue. A payout will not be dispatched until it is released by an analyst.

If the check does not return within the allotted time, the payout will be held and not be released by default. This is because the cost of missing a control is greater than the cost of a delayed payout. Each decision is recorded in the audit trail along with the inputs that were used to make the decision and the version of the rule or model that was used to make it.

5.6. Integration and Payout Rails

The integration layer links the hub to the rails via an adapter for each of the rails: FedNow and RTP for account-to-account payments, push-to-card and digital wallets for card and wallet payments, and ACH as the fallback rail. Each of the instant rails exchanges ISO 20022 messages [8].

The adapter for each of the rails translates the payout event from the hub to the credit transfer message for the rails, includes the payout ID in the end-to-end reference of the message, and translates the status and return messages from the rails to the hub.

The adapters provide information about the availability and response time of each of the rails to the routing layer, which uses this information as the availability indicator in Equation (1). ACH serves as the safety net for the instant rails.

5.7. Cross-Cutting Concerns

Security, observability, and auditability features are incorporated into every layer of the system. All sensitive data is encrypted. Payment card data is stored in a tokenization vault to limit the scope of PCI DSS compliance [9].

User access to the system is based on user role and the principle of least privilege. Cryptographic keys are managed separately from sensitive data in accordance with established control catalogs [37].

All state changes in the system are recorded in an audit log that cannot be altered. For every payout, the system can determine when the payout was requested, the controls it passed, the rail used for the payout, and when the payout was confirmed.

Furthermore, the system provides end-to-end tracing for each payout to allow for operational and regulatory inspection of every payout made.

6. Disbursement Process Model

Figure 2 presents the life cycle of a payout as a seven-stage pipeline, designed for delivery within seconds at any hour.

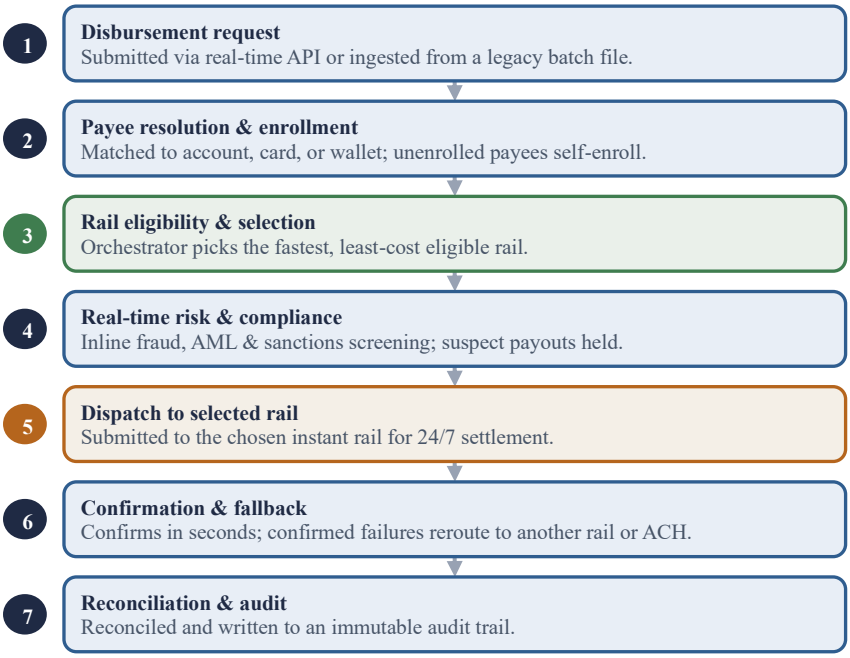


Fig. 2 The disbursement lifecycle from batch to near-instant. A request is resolved to a payee, routed to the fastest eligible rail, screened in-line, dispatched, and confirmed in seconds, with status-confirmed fallback to an alternate rail or ACH on failure, and full reconciliation and audit

The pipeline goes from (1) disbursement request (received via real-time API or the legacy batch file), through (2) payee resolution and enrollment, (3) rail eligibility and selection, (4) real-time risk and compliance screening, (5) dispatch to the selected rail, (6) confirmation and fallback, and (7) reconciliation and audit. Confirmation and failure information are fed back to the routing layer so that the best rail can be selected each time.

The rule is that each request should be completed by the fastest eligible rail that is available, and that a rail failure should not result in the lost payout of that request or in the duplicate payout of that same request.

6.1. Payment Failure Handling and Duplicate-Payment Prevention

Setting up an automatic fallback to another rail must be designed with care. If a payout attempt fails to return a definite response, it is possible that the payout has already been successfully settled. In that case, attempting to send the same payout over a second rail will result in the payout being sent twice. Because instant payments are irrevocable, the only way to recover the duplicate payout is to request a return of funds from the recipient. Therefore, the model for payout attempts recognizes and treats these three outcomes of a payout attempt differently.

When a rail sends back a confirmed rejection prior to settlement, the routing layer automatically attempts to process the payout using the next eligible rail. A confirmed acceptance marks the payout as settled. Payouts that have an unknown outcome, such as when a timeout occurs or the response is lost, are never automatically rerouted. Instead, the payout is placed in a pending-investigation state, and the hub requests the status of the payout from the rail using the ISO 20022 standard [8]. The payout is only released for fallback if the rail confirms that the original attempt to process the payout was rejected or that it is no longer possible to settle the payout according to the rules of the rail. If it is not possible to confirm the status of the payout within the specified time period, the payout is escalated to the manual review team and is not automatically retried.

There are four mechanisms supporting this rule. Each payout includes a payout identifier, each attempt includes an attempt identifier, and the rail adapters are able to submit attempts idempotently [27]. The orchestration layer stores a state record for each payout. The states for a payout are created, screened, dispatched, pending investigation, settled, failed, rerouted, and returned. Only one attempt is permitted at one time, and a new attempt can only be created from a failed attempt. The state changes, and messages are committed together using the saga and transactional outbox patterns, ensuring that if any component fails, it is not possible for a payout to be dispatched without being recorded in the system [26][28]. The reconciliation stage compares the hub ledger with the settlement reports from each of the rails involved in the payout process. Any payout that is settled more than once

is identified and flagged for recovery from the rail. In addition, each rail adapter has a circuit breaker that stops dispatching payouts to a rail if it is failing. This limits the number of payouts that enter an unknown state [29].

These rules trade speed for safety in the uncertain case: a payout with an unknown outcome is delayed while its status is resolved. The choice is deliberate, since for a disbursement program, a short delay can be recovered, and a duplicate payment may not be.

7. Migration Strategy: From Batch to Real Time

Since everything is running around the clock (R1), paper is skipping the big-bang cutover. Instead, migration uses the strangler fig pattern [18] to break the migration into small, reversible steps. Right from the start, the legacy batch-file bridge (Section 5.1) lets the hub handle the disbursement files that are already in use. From there, the new capabilities can be enabled one program, region, or rail at a time, leaving the rest of the volume to be processed by the existing system.

Recipients are gradually moving to instant delivery. When a recipient adds a bank account, card, or wallet that is capable of receiving instant transfers, their payouts are moved to that instant transfer method, with ACH transfers remaining as the fallback for these payouts. To ensure a smooth transition from the legacy payment system to the new system, both systems can be run in parallel for a specified period to compare the payouts of each system. Once the transition is successful, the legacy system will be retired for that payment segment. By using this gradual approach, the replacement of the legacy system with the new system is less risky. Furthermore, each step of the process provides value to the recipient by increasing the amount of payments that are made instantly rather than overnight. The legacy system remains available for all payments until the last segment of payments has been migrated to the new system.

8. Analytical Evaluation

The design is evaluated analytically in two ways: by walking through operational scenarios to check that the architecture responds as each requirement demands, and by comparing the design with alternative approaches. The scenarios in Table 2 were chosen so that each requirement in Section 4 is exercised at least once, following the practice of scenario-based architecture evaluation [32][33]. The evaluation establishes that the design is consistent with its requirements. It does not measure how well a running system would perform.

Table 2. Scenario-based analytical evaluation of the reference architecture against requirements R1 to R4

Scenario	Stimulus	Designed architectural response	Req.	To be measured
S1. Legacy file intake	An originator submits an existing overnight batch file.	The batch-file bridge splits the file into payout events with persistent identifiers; eligible payouts are routed to instant rails, and the remainder to ACH.	R1	Throughput of the bridge at peak file size
S2. Instant rail unavailable	The selected instant rail stops responding before dispatch.	The circuit breaker marks the rail unavailable; routing selects the next eligible rail, or ACH.	R2	Time to detect the outage; share of volume diverted
S3. Unknown dispatch outcome	A dispatch times out, and no status is returned.	The payout enters pending investigation, and a status request is sent; fallback occurs only after a confirmed rejection, otherwise the payout goes to manual review.	R2, R3	Frequency and duration of unknown outcomes
S4. Screening alert	Sanctions or fraud screening returns a hold.	The payout is removed from the instant path and queued for review; nothing is dispatched until it is released.	R3	Screening latency; false positive rate
S5. Payee without an instant-capable account	The payee's bank does not participate in an instant rail.	Routing selects push-to-card or a wallet if one is enrolled; otherwise, ACH.	R4	Enrollment rate of cards and wallets
S6. Duplicate submission	The same file or API request is received twice.	The persistent payout identifier is recognized, and the second request is rejected as a duplicate.	R1, R3	Correctness of identifier derivation across originators
S7. Migration step reversed	A newly migrated program shows reconciliation breaks.	The program is routed back to the legacy pipeline through the bridge; other programs are unaffected.	R1	Time to reverse; reconciliation coverage

Source: analysis by the author. The table records the designed response and does not report observed behavior.

8.1. Comparison with Alternative Approaches

Three alternatives to the proposed design were considered. The first is to keep batch processing but shorten the cycle (e.g., use same-day ACH). This is the change that requires the least effort. However, since delivery would still be tied to processing windows and business days, it is not possible to achieve the goal of delivery in seconds at any time of the day. The second alternative is to integrate one instant rail into the existing platform.

This will allow for instant delivery to those that can be reached on that rail, but does not meet R2 and R4. Those that cannot be reached on that rail would not see a change in their delivery experience, nor is there an alternative if that rail is not available. The third alternative is to replace the platform with a new real-time system in one cutover. This would meet R2-R4, but would conflict with R1. A change this drastic cannot be easily reversed, which is something that is advised against in the migration literature [18][24].

The proposed design is different from the three in that it separates the routing decision from both the originator and the rails and keeps the legacy pipeline in service behind a bridge. The cost of this proposed design is the additional complexity required to build and operate an orchestration layer, a state model for every payout, and several rail adapters.

8.2. Threats to Validity

This evaluation has its limitations. The selection of scenarios was made by the author and may not be exhaustive of the failures that can occur in a production system. Table 1 reflects the author's interpretation of the sources cited. No measurements were made of any quantitative properties of the system, and the benefits described in Section 9 are based on assumptions that would need to be replaced by the data from the operator.

9. Expected Benefits and Evidence Classification

The expected benefits of real-time disbursement are expressed in latency, cost, and reach. Since no experiment was performed, none of the figures in this section is a measured result of this study. Table 3 lists each figure with its evidence class, as defined in Section 3.2, and with its source or derivation, so that published benchmarks, practitioner-reported observations, assumptions, and model estimates cannot be confused. Table 4 lists the assumptions on which the model estimates depend.

Table 3. Expected benefit levers of a real-time, multi-rail disbursement hub, by evidence class

Benefit lever	Metric	Figure	Evidence class	Source or derivation
MEASURED RESULTS (this study)				
None	No experimental or production measurement was performed	Not applicable	Measured result	Identified as future work (Section 11)
PRACTITIONER-REPORTED OBSERVATIONS (comparable deployment; not independently verified)				
National-scale disbursement platform	Annual volume processed	$11B+	Practitioner-reported	Author’s professional observation; not a result of this study
Wallet-rail integration	Disbursement latency	~24 hrs → near-instant	Practitioner-reported	Author’s professional observation; not instrumented
PUBLISHED BENCHMARKS				
Instant-rail settlement	Time to funds/availability	seconds, 24/7/365	Published benchmark	[2][3]
Recipient preference	Choose instant when offered	~72%	Published benchmark	[7]
Instant-rail transaction limit	Max per payment	Up to $10M	Published benchmark	[3][5]
Global fast-payment adoption	Transactions per person per month, 2022 (Thailand, Brazil, South Korea)	35 / 27 / 21	Published benchmark	[1]
Paper check issuance	Median cost per check issued	$2.01 to $4.00	Published benchmark	[36]
ACH payment	Median cost per transaction	$0.26 to $0.50	Published benchmark	[36]

MODEL ESTIMATES for this hub (hypothetical; computed from the assumptions in Table 4; not demonstrated)				
Event-driven core + instant rails	Payout latency reduction, instant-eligible payouts	est. >99% (24 hrs → under 1 minute)	Model estimate	Equation (3); assumption A1
Check displacement	Check reissue and exception handling	est. 40 to 60% lower	Model estimate	Proportional to check volume; assumption A2
Electronic vs. check issuance	Blended operating cost per payout	est. 30 to 50% lower	Model estimate	Equation (4); assumptions A3 to A5
EXPECTED OUTCOME (depends on recipient behavior)				
Wallet/card enrollment	Share of recipients paid instantly	est. 20 to 40 percentage points higher	Expected outcome	Assumption A6; to be confirmed with enrollment data

Model estimates and expected outcomes are hypothetical. They are not experimental results and require the operator's own instrumented measurement to confirm.

Table 4. Assumptions underlying the model estimates in Table 3

ID	Assumption	Value used	Basis
A1	End-to-end time for an instant-eligible payout after migration, including inline screening	Under 60 seconds, against a 24-hour batch cycle	Assumption, consistent with the settlement speed published for the instant rails [2][3]
A2	Share of paper checks displaced by electronic delivery	40 to 60%	Assumption (illustrative); reissue workload taken as proportional to check volume
A3	Share of payouts issued by check before migration	30%	Assumption (illustrative)
A4	Unit cost of a check, an ACH payment, and an instant payout	\$3.00, \$0.40, and \$0.50	Check and ACH values lie within published ranges [36]; the instant payout value is an assumption
A5	Share of check payouts that move to an instant rail	50 to 80%	Assumption (illustrative)
A6	Recipients who enroll an instant-capable account, card, or wallet during phased rollout	20 to 40% of recipients	Assumption, below the stated preference of about 72% [7]
A7	Annual rate used to value earlier availability of funds	5%	Assumption (illustrative)

All values are inputs chosen for illustration and should be replaced by the operator's own data.

9.1. Derivation of the Model Estimates

Latency. For a payout that is eligible for an instant rail, the relative reduction in time to funds is

$$\Delta T = \frac{T_{\text{batch}} - T_{\text{instant}}}{T_{\text{batch}}} \tag{3}$$

With T_{batch} = 24 hours (86,400 seconds) and T_{instant} below 60 seconds (A1), ΔT exceeds 99.9%. The estimate applies only to instant-eligible payouts; payouts that fall back to ACH keep the ACH timetable.

9.1.1. Cost

Let s be the share of payouts issued by check before migration, m the share of those checks that move to an instant rail, and k_{chk}, k_{ach}, and k_{inst} the unit costs of a check, an ACH payment, and an instant payout. With the remaining payouts sent by ACH, the relative reduction in blended cost per payout is

$$\Delta K = \frac{m\, s\, (k_{\text{chk}} - k_{\text{inst}})}{s\, k_{\text{chk}} + (1-s)\, k_{\text{ach}}} \tag{4}$$

With the values in Table 4 (s = 0.30, k_{chk} = \$3.00, k_{ach} = \$0.40, k_{inst} = \$0.50), the blended baseline cost is \$1.18 per payout, and ΔK is 32% for m = 0.5 and 51% for m = 0.8, which gives the range of approximately 30 to 50% in Table 3. The result is sensitive to s: a platform that issues few checks has correspondingly less to save.

9.1.2. Exception Handling

Check reissue and the handling of lost, stolen, or stale checks arise only for payouts made by check; under the assumption that this workload is proportional to check

volume, displacing 40 to 60% of checks (A2) reduces it by the same proportion.

9.1.3. Reach

The share of recipients paid instantly rises by the share who enroll an instant-capable credential (A6). This expected outcome depends on enrollment behavior and cannot be derived from the design.

9.2. A Worked Illustration of Dollar Impact

Two levers illustrate the financial effect on a platform of the scale reported in Table 3. Both calculations are hypothetical. The first is the value to recipients of receiving funds one day earlier. For an annual volume V and an annual rate i (A7), the value of one day is

$$B_{\text{time}} = \frac{V \times i}{365} \tag{5}$$

With V = \$11 billion and i = 5%, B_{time} is approximately \$1.5 million per year. The second lever is the cost of issuance. If N_{d} checks are displaced by instant payouts, the savings are

$$B_{\text{cost}} = N_{\text{d}} \left(k_{\text{chk}} - k_{\text{inst}}\right) \tag{6}$$

With the unit costs in Table 4, every million checks displaced saves approximately \$2.5 million; across the published range of check costs [36], the figure lies between about \$1.5 million and \$3.5 million. These illustrations indicate the order of magnitude of the benefit. They are not forecasts for any particular platform.

10. Discussion

The architecture implies a certain view. The hardest part of enabling instant disbursement is not the connection to a rail, but the coordination of several rails in real-time, while the existing system remains in service. The architecture has three consequences. First, since originators see only the hub, rails can be added or withdrawn and absorb their failures without any change on the originator's side. Second, the reliability of the system can be preserved with a status-confirmed fallback that ensures that the move to irrevocable instant payment does not weaken the guaranteed delivery that is required for disbursement, and that does not introduce the possibility of duplicate payments. Third, by pairing an event-driven core with a strangler-style migration, it is possible for the institution to obtain value in increments instead of having to stake the outcome of the migration on a single cutover.

Compared to existing literature, this work reports no experimental results and does not claim better measured performance than the methods discussed in the literature. Its claim is one of coverage of the problem. The fast payment literature [1][19][20][22] explains how rails are designed and adopted; this study adds the question of how an originator's platform chooses a rail for each payout. Current research on migration [10][18][23][24] and payment platform modernization [12][13] considers the system as a whole; the present study investigates the per-payout routing decision, the placement of inline controls within the system, and the bridge that allows legacy batch files to be served by instant rails. Research on the reliability of distributed systems [26][27][28][29] provides general principles on reliability; the present study adds the specific rule that irrevocable payouts with unknown outcomes are to be investigated before the payout is rerouted.

The reason the design encompasses these aspects is structural. By including the routing decision, the payout state, and the control point in one orchestration layer, there is one place where the eligibility, screening, and failure of each payout is determined. It remains to be seen whether this coverage translates into the latency, cost, and reach improvements discussed in Section 9.

The design is applicable to various disbursement domains, such as social benefits, tax refunds, insurance claims, and payroll. However, there are also costs to implementing this design. The orchestration layer is a critical component that must be highly available; the screening process adds time to each payout; and unknown outcomes are investigated before any rerouting, which delays some payouts to rule out duplicate payments.

11. Limitations and Future Work

This paper proposes a reference architecture and draws on the author's experience with large-scale disbursement platforms. However, it does not evaluate the proposed architecture for performance. The principal limitation is the lack of empirical validation. No prototype, simulation, or production dataset supports the model estimates. Furthermore, the practitioner-reported figures, namely the annual volume processed and the reduction in disbursement latency, have not been independently verified. The literature comparison is structured but not exhaustive. The scenario walkthrough was performed by the author only and was not reviewed by multiple stakeholders, as scenario-based evaluation methods recommend [32]. Finally, the cost model uses illustrative unit costs and volume shares, and the fees that financial institutions charge for instant payouts vary.

Future work follows in four directions. The first is empirical validation: a prototype of the orchestration layer tested against simulated rail behavior, followed by measurement in a production deployment of the time to funds, the screening latency, the rate of unknown outcomes, and the cost per payout. The second is the routing policy, where rail health and observed costs can be used to optimize the weights in Equation (2). The third is the real-time detection of fraud in irrevocable payouts, including the trade-off between screening time and detection quality [15]. The fourth is interoperability

between instant payment infrastructures, which would simplify routing if it were provided at the level of the rails [16].

12. Conclusion

Instant payment systems are now available to users, and the recipients of such payments prefer these systems to the alternative. However, the systems that are used to distribute social benefits, tax refunds, and insurance claims were developed for overnight processing of the payments and cannot be replaced. This paper has proposed an architecture for the modernization of these payment systems without disrupting the operation of the current systems. The proposed architecture introduces an orchestration and routing layer that presents the available payment rails to the system as a single service. Each payment is routed to the fastest eligible rail and screened before it is sent. If the first attempt fails, the payment is routed to an alternative rail only after the rail has confirmed that the original attempt was rejected, so that no payment is made twice; a payment whose outcome is unknown is investigated and is not retried automatically. The modernization plan proposes a phased approach to connecting the current batch payment system to the proposed event-driven platform. The conceptual architecture was evaluated according to the requirements and the existing literature on the subject. The benefits of the proposed system were estimated, such as reducing the time that payment recipients have to wait to receive the payments from one day to seconds for instant payments, reducing the cost of issuing the payments by eliminating the need for checks to be used to make payments, and increasing the number of ways in which payments can be made, such as by using payment cards and digital wallets. While these benefits were estimated, it has not yet been demonstrated in an experiment that the proposed architecture will yield these benefits. Therefore, the main task of future research on this topic will be to measure the benefits of the proposed system. In conclusion, it is possible for the current batch payment systems to be modernized to support real-time payments through a series of small, reversible, and useful steps.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Funding Statement

The author received no specific funding for the research or publication of this article.

Acknowledgments

The views and opinions expressed in this paper are solely those of the author and do not represent the views of the author's employer.

References

[1] Jon Frost et al., "Fast Payments: Design and Adoption," *BIS Quarterly Review*, 2024. [Google Scholar] [Publisher Link]

[2] The Federal Reserve, FedNow Service. [Online]. Available: https://www.frbservices.org/financial-services/fednow-service/

[3] The Clearing House, RTP (Real-Time Payments) Network. [Online]. Available: https://www.theclearinghouse.org/payment-systems/rtp

[4] Nick Stanescu, Infographic: Two years of the FedNow® Service, Federal Reserve, 2025. [Online]. Available: https://explore.fednow.org/explore-the-city?id=3&postId=90&postTitle=infographic:-two-years-of-the-fednow-service

[5] Zhu Wang, and Vinh Phan, "FedNow and The Development of U.S. Fast Payments," *Federal Reserve Bank of Richmond Economic Brief*, 2026. [Publisher Link]

[6] U.S. Department of the Treasury, Digital Payouts, 2025. [Online]. Available: https://fiscal.treasury.gov/payments-from-government/digital-pay

[7] PYMNTS Intelligence, 72% of Consumers Pay for Instant Income Access, 2026. [Online]. Available: https://www.pymnts.com/real-time-payments/2026/72percent-consumers-pay-for-instant-income-access/

[8] International Organization for Standardization, ISO 20022 - Universal Financial Industry Message Scheme, 2026. [Online]. Available: https://www.iso.org/standard/20022-1

[9] PCI Security Standards Council, Payment Card Industry Data Security Standard (PCI DSS). [Online]. Available: https://www.pcisecuritystandards.org/standards/pci-dss/

[10] Sam Newman, *Building Microservices*, 2nd ed., O'Reilly Media, 2021[Publisher Link]

[11] Martin Kleppmann, *Designing Data-Intensive Applications*, O'Reilly Media, 2017. [Google Scholar] [Publisher Link]

[12] Lokendra Singh Kushwah, "Enterprise Platform Modernization: Migrating Legacy Payment Systems to Cloud-Native Architectures," *European Journal of Computer Science and Information Technology (EJCSIT)*, vol. 13, no. 7, pp. 19-35, 2025. [CrossRef] [Publisher Link]

[13] "Global Payment Platform: Microservices at Scale," *World Journal of Advanced Research and Reviews*, vol. 26, no. 1, pp. 3107-3113, 2025. [CrossRef] [Publisher Link]

[14] Asli Demirgüç-Kunt et al., *The Global Findex Database 2021: Financial Inclusion, Digital Payments, and Resilience in the Age of Covid-19*, vol. 1, World Bank Group, 2022. [Google Scholar] [Publisher Link]

[15] Spiros Thivaios et al., "A Survey of Machine Learning and Deep Learning for Financial Fraud Detection: Architectures, Data Modalities, and Real-World Deployment Challenges," *Algorithms*, vol. 19, no. 5, pp. 1-39, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[16] Cheryl Winokur Munk, U.S. Real-time Payments are Set for a Boost. Hers's Why, American Bankers, 2025. [Online]. Available: https://www.americanbanker.com/payments/news/rtp-fednow-try-to-boost-real-time-pay-in-the-u-s

[17] Payment Trends in 2026: Innovation, Trust, & Growth, Mastercard, 2025. [Online]. Available: https://www.mastercard.com/in/en/business/payments/merchant-cloud/insights/payment-trends-in-2026.html.

[18] Martin Fowler, Strangler Fig, martinfowler.com, 2024. [Online]. Available: https://martinfowler.com/bliki/StranglerFigApplication.html

[19] BIS, Fast Payments: Enhancing the Speed and Availability of Retail Payments, BIS. [Online]. Available: https://www.bis.org/publications/fast-payments-enhancing-speed-and-availability-retail-payments

[20] Implementation Considerations for Fast Payment Systems, World Bank Group, 2021. [Online]. Available: https://documents1.worldbank.org/curated/en/515381630386274535/pdf/Implementation-Considerations-for-Fast-Payment-Systems.pdf

[21] Angelo Duarte et al., "Central Banks, the Monetary System and Public Payment Infrastructures: Lessons from Brazil's Pix," *SSRN*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Jose Aurazo et al., "The Design and Adoption of Fast Payments," *Journal of Payments Strategy and Systems*, vol. 18, no. 4, pp. 366-380, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Nicola Dragoni et al., *Microservices: Yesterday, Today, and Tomorrow*, Present and Ulterior Software Engineering, Springer, pp. 195-216, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[24] Davide Taibi, Valentina Lenarduzzi, and Claus Pahl, "Processes, Motivations, and Issues for Migrating to Microservices Architectures: An Empirical Investigation," *IEEE Cloud Computing*, vol. 4, no. 5, pp. 22-32, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[25] Gregor Hohpe, and Bobby Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*, Addison-Wesley, Boston, MA, United States, 2003. [Publisher Link]

[26] Hector Garcia-Molina, and Kenneth Salem, Sagas, pp. 249-259, 1987. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/38713.38742?trk=public_post_comment-text

[27] Pat Helland, "Idempotence is not a Medical Condition," *Communications of the ACM*, vol. 55, no. 5, pp. 56-65, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[28] Chris Richardson, *Microservices Patterns: With Examples in Java*, Manning Publications, 2018. [Google Scholar] [Publisher Link]

[29] Michael T. Nygard, *Release it! Design and Deploy Production-Ready Software*, 2nd ed., Pragmatic Bookshelf, 2018. [Google Scholar] [Publisher Link]

[30] Alan R. Hevner et al., "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75-106, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[31] Ken Peffers et al., "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45-77, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[32] Rick Kazman, Mark Klein, and Paul Clements, "*ATAM: Method for Architecture Evaluation*," Software Engineering Institute, Carnegie Mellon University, Technical Report CMU/SEI-2000-TR-004, 2000. [Google Scholar]

[33] Len Bass, Paul Clements, and Rick Kazman, *Software Architecture in Practice*, 4th ed., Addison-Wesley, 2021. [Google Scholar] [Publisher Link]

[34] U.S. Department of the Treasury, Office of Foreign Assets Control, Sanctions Compliance Guidance for Instant Payment Systems, 2022. [Online]. Available: https://ofac.treasury.gov/system/files/126/instant_payment_systems_compliance_guidance_brochure.pdf

[35] Modernizing Payments to and From America's Bank Account, Executive Order 14247, 2025. [Online]. Available: https://www.whitehouse.gov/presidential-actions/2025/03/modernizing-payments-to-and-from-americas-bank-account/

[36] Association for Financial Professionals, 2022 AFP Payments Cost Benchmarking Survey. [Online]. Available: https://www.financialprofessionals.org/training-resources/resources/survey-research-economic-data/Details/paymentscost

[37] Security and Privacy Controls for Information Systems and Organizations, NIST Special Publication 800-53, Revision 5, National Institute of Standards and Technology, 2020. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final