

Original Article

# A Multi-Tier Orchestration Architecture for Regulated Cryptocurrency and Stablecoin Services within Large-Scale Payment Service Providers

**\*Anath Bandhu Chatterjee**  
Staff Software Engineer, PayPal Inc.

## Abstract:

Cryptocurrencies and regulated stablecoins will come with compliance, scalability, resilience, liquidity management, and interoperability, among other architectural challenges in the mainstream payment service provider (PSP) environment. Conventional payment systems are geared toward deterministic settlement and central control while the digital asset system is built using distributed network, partner, custody models, and changing regulatory frameworks. The purpose of this paper is to propose a multi-tier orchestration architecture that supports regulated cryptocurrency and stablecoin services in large-scale PSP environments, while keeping the vendors neutral. The architecture is based on 5 logical layers: client channels, experience and API services, transaction orchestration, digital asset core services and shared data infrastructure. An event-driven processing model allows to execute transactions on a scale, maintain regulatory traceability, and to integrate asynchronously with distributed services. It also features a standardized Outbound Connector Framework to connect custodians, market data vendors, compliance services, and tax platforms, alongside a Liquidity Pool Management architecture to enable treasury management, stablecoin conversion, settlement checks and constant reconciliation. It is important to note that the architectural analysis emphasizes regulatory compliance, operational resilience, flexibility, audibility and the ability to support the development of digital asset ecosystems. This model offers a good template for both PSPs and neobanks, and also for merchant platforms aiming to provide regulated digital asset services.

## Keywords:

Digital Asset Platform, Cryptocurrency Services, Stablecoin Infrastructure, Payment Service Providers, Multi-Tier Architecture.

## Article History:

**Received: 13.05.2026**

**Revised: 15.06.2026**

**Accepted: 23.06.2026**

**Published: 02.07.2026**

## 1. Introduction

Payment service providers (PSP) are faced with a new architecture challenge with the mainstreaming of cryptocurrencies and regulated stablecoins in primary payment systems [1]. However, digital asset services are now becoming an expectation, and not just a one-off addition, but a component of the traditional financial product that customers have come to expect [2][3]. This need offers opportunities for PSPs, but also significant technical and regulatory complexity [4][5]. While deterministic settlement, centralized ledger management, and tight audit rules are the key features of established payment platforms, blockchain-based systems are based on distributed networks, a synchronicity of settlement and the continuous nature of regulatory frameworks.

Digital asset ecosystems exhibit characteristics that differ fundamentally from conventional payment infrastructures [6]. Blockchain transaction finality is probabilistic rather than immediate, market prices can fluctuate rapidly, and custody functions are frequently delegated to regulated third-party providers. Further, compliance requirements such as anti-money laundering (AML),



sanctions screening, tax reporting and jurisdiction-specific regulatory compliance should be built into the transaction workflows [7]. There is a need for architectural solutions that meet the needs of the emerging digital asset services while maintaining the discipline and rigor necessary for regulated financial institutions [8].

This paper provides a vendor-neutral reference architecture for regulated services for cryptocurrencies and stablecoins in large-scale PSPs. The architecture will be designed to facilitate buy/sell/convert/custody/transfer activities and will be compliant, auditable, resilient and operationally scalable. The architecture doesn't just outline a particular implementation or technology stack; it offers a blueprint that's adaptable for various organizational, regulatory, and partner contexts. The major contributions of this paper are as follows:

- Layered Reference Architecture: A multi-layer service-oriented architecture that achieves separation of customer channels, experience and API services, transaction orchestration, digital asset processing, and supporting infrastructure to enhance modularity and maintainability.
- Event-Driven Transaction Model: An event-driven workflow that enables scalable transaction processing, regulatory traceability, auditability and failure-domain isolation of distributed services.
- Outbound Connector Framework: A common integration interface that hides interactions with custodians, market data services, compliance services, tax platforms and other external partners.
- Liquidity Pool Management Architecture: A specialized treasury-management structure to manage multi-pool liquidity operations, stablecoin conversion, settlement monitoring and continuous reconciliation in custodial environments.

### 1.1. Structure of the Paper

The paper is organized as follows. Section II reviews related work, while Section III presents the design principles and architectural requirements. Sections IV to VII describe the reference architecture, trade subsystem and failover mechanism, liquidity pool management architecture, and event-driven transaction model. Section VIII discusses deployment considerations and supporting technologies, Section IX provides architectural analysis, and Section X concludes the paper with future research directions.

## 2. Background and Related Work

This section will discuss the basics of the custodial wallet system, stablecoin architectures, blockchain-based AML analytics, and previous reference architectures for regulated digital asset systems.

### 2.1. Custodial Wallet Systems and Regulated Custody

These early exchanges introduced the custodial wallet offering, in which all the assets of users are held on the blockchain in a pool of hot wallets, and private user entitlements are tracked on an off-chain internal ledger. It allows for better operational efficiency and transaction volume, but can lead to counterparty, insolvency, and segregation risks. Zetzsche and Nikolakopoulou (2025) offer a comprehensive database for analyzing the Terms and Conditions (T&Cs) of the top crypto custody and wallet providers. The sample examined shows that there is a dichotomy in the crypto custody industry: some are willing to safeguard the rights and interests of their clients, and others offer T&Cs that are in conflict with any meaningful client protection. This article discusses the nature of held custody of crypto-assets and introduces the research sample. The article then offers results regarding custody practices, such as safekeeping, key storage, outsourcing, asset segregation, insolvency protection, client entitlement, reuse, liability cap and choice of law, courts and arbitration [9].

### 2.2. Stablecoin Issuance and Reserve Architectures

Stablecoins have evolved from settlement tools that are used on exchanges to widely used digital currencies representing fiat currency with short-term U.S. Treasury bonds, cash equivalents, and audited reserve frameworks. The development is a sign of an increasing convergence between blockchain settlement systems and the traditional finance system. Liao, Hadeed and Zeng (2023) present empirical evidence that shows a notable shift in the usage of stablecoins, a form of digital currency transacted over distributed ledgers, in recent years. At its inception, stablecoins were on-chain cash balances of crypto exchanges and traders, but over the last few years, they have come to look like a new type of money tokenized on public blockchains. As a result, the implementation of general-purpose payment stablecoins in real-time gross settlement (RTGS) could alleviate concentration risk and liquidity risk in the existing payment systems [10].

### 2.3. On-Chain Risk Analytics and AML Tooling

Blockchain analytics systems are vital to regulatory compliance as they enable address attribution, transaction tracing and sanctions screening and are compliant with the Bank Secrecy Act, FATF travel rule guidance and the EU Transfer of Funds Regulation. Chen et al. (2026) propose HeteroGCL, a framework that leverages heterogeneous graph contrastive learning for scalable and sustainable cryptocurrency AML. It proposes an approach that simulates cryptocurrency transactions as a multi-type graph and combines a multi-type graph attention network with a multi-type graph contrastive learning module. By leveraging unlabeled data through topology-aware and attribute-aware graph augmentations, HeteroGCL mitigates label scarcity while enabling scalable and label-efficient AML model training while reducing reliance on costly manual annotation. The results indicate that the proposed framework effectively captures complex money laundering patterns while supporting scalable deployment of AML systems and improving the economic and operational sustainability of blockchain AML infrastructures [11].

### 2.4. Prior Reference Architectures

There has been a continuous evolution of reference architectures for regulated blockchains and custody in recent years, each of which uses layered architectures to separate and compartmentalize identity, execution, and control. Staley (2026) defines composable primitives of identity, policy, payments and custody for orchestrated financial operations [12]. Similarly, Cruz et al. (2025) propose a modular wallet structure that combines custody solutions, layering, and blockchain connectivity to provide scalable and effective digital asset management [13]. Furthermore, Homoliak et al. (2021) to promote security in transaction systems, the Blockchain Security Reference Architecture presents a layered security from the network, consensus, and application perspectives [14]. Together, these architectures form the design principles that can be leveraged to create regulated, multi-tier crypto custody and PSP orchestration systems.

### 2.5. Research Gap

While the field has made great strides in custody and stablecoin design, AML analytics, and modular blockchain architecture, the current literature is split into individual system layers. Current custodial models are asset safekeeper-focused, studies of stablecoins focus on reserve models and liquidity, and AML frameworks are external compliance overlays. However, previous reference architectures only offer modular designs without any common orchestration between custody, settlement, compliance and payment processing in large-scale PSP environments. This means there is no real-time, dynamic orchestration system, driven by policies, that coordinates regulatory compliance, transaction execution, and liquidity management across a diverse blockchain and financial ecosystem.

## 3. Design Principles and Requirements

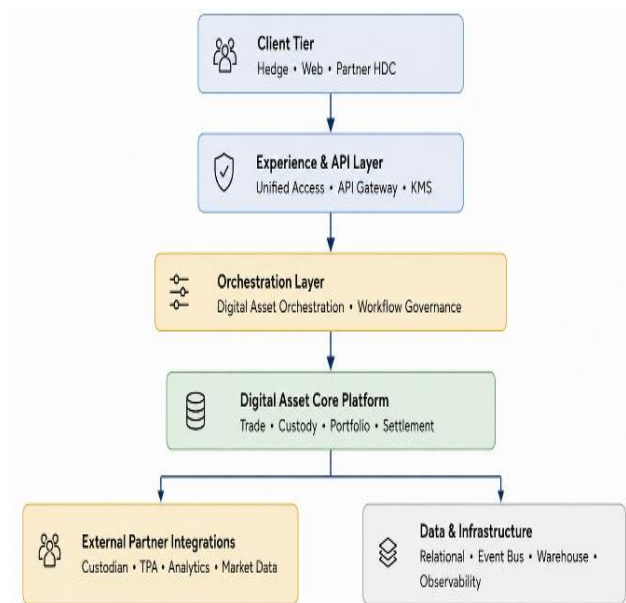
The architectural elements and principles are structured by seven design principles that are directly related to the above operational realities.

- Compliance as a Deterministic Execution Invariant: Compliance is enforced; it's an invariant for all state-changing operations. Orchestration flow is performed for every transaction that includes KYC, AML, Sanctions and policy checks [15]. Compliance decisions are made at all times using a non-changeable event log that ensures complete auditing for regulatory compliance.
- Structural Heterogeneity of External Partners: Other components, such as custodians, pricing suppliers, analytics applications and tax systems have different API characteristics, reliability and latency. This heterogeneity is assumed and abstracted with a common Outbound Connector Framework, which provides uniform retry, idempotency, authentication, and audit logging.
- Service Continuity under Partner Failure: The system is designed to be independent of failures of external partners. A Stand-In execution mode allows for fallback to secondary providers for pricing and execution, and specifies risk limits and marks transactions for later reconciliation.
- End-to-End Auditability via Event Sourcing: Each transaction needs to be reconstructible end-to-end - from the original user interaction, all the way through any internal decision and external API call. An event-driven architecture, with each domain event written to an append-only event log, that can be replayed for regulatory reporting and disaster-recovery scenarios.
- Modularity Modular Decomposition for Regulatory Agility: Regulatory regimes change quickly and unevenly across jurisdictions. The architecture is decomposed into bounded contexts (trade, transfer, portfolio, tax, compliance, treasury) that can evolve independently. New assets, jurisdictions, or partners can be onboarded by extending the relevant bounded context without modifying the rest of the platform.

- **Tenant-Isolated Multi-Channel Design:** The platform must serve multiple consumer-facing brands and partner channels (Web3 partners, embedded SDK clients) without per-channel duplication of business logic. Orchestration and core services therefore expose tenant-aware APIs with logical isolation of identity, configuration, and accounting.
- **Unified Observability as a System Requirement:** Real-time monitoring of pool balances, partner health, transaction throughput, and reconciliation drift is required for both regulatory reporting and treasury management. The data and infrastructure layer provide these observability primitives as a shared service across all bounded contexts.

#### 4. Layered Reference Architecture

The multi-tier reference architecture for delivering regulated cryptocurrency and stablecoin services within a large-scale payment service provider. The architecture is designed to be logically layered into five layers to separate the customer interaction, transaction orchestration, digital asset processing and infrastructure. This separation of concerns allows independent scalability, regulatory isolation, and abstraction of partners and operational resilience, while keeping a common compliance and audit framework, as shown in Figure 1.



**Figure 1. Multi-Tier Orchestration Architecture for Regulated Digital Asset Services**

The architecture separates client channels, API services, transaction orchestration, digital asset processing, and shared infrastructure to support scalability, resilience, and regulatory compliance.

Architecture is built based on a layered decomposition model with the responsibilities of each layer being well defined and clear interfaces between adjacent layers. The orchestration layer provides coordination for business workflows, and the Bounded contexts in the digital asset core provide domain-specific processing. External dependencies are separated using a connector abstraction layer that enables the partner implementations to change independently with respect to internal services. This architecture reduces coupling between layers and facilitates regulatory, operational and technological changes.

##### 4.1. Client Tier

The client tier includes consumer-facing mobile and Web applications, under the client's primary and secondary brands, and a partner-facing embedded SDK, which lets third-party applications bring digital asset capabilities to the forefront of their own user experience. A separate merchant integration channel for external Web3 applications ensures compliance and orchestration controls are applied to the same degree as they are to native PSP experiences. All client surfaces are versioned and feature-flagged to help roll out new functionality incrementally to user cohorts and jurisdictions.

#### 4.2. Experience and API Layer

This layer hides cross-cutting issues of authentication, authorization, identity federation and request shaping. A Unified Login service is based on the OAuth 2.0 protocol, rotation of the tokens and the proof-of-possession bindings, while a Crypto SDK component provides a stable, language-independent API for client applications [16][17]. The API Gateway will drop the client connections, impose rate limits and quotas, validate the request against a schema and direct traffic to other orchestration services in the downstream. System for Cross-domain Identity Management (SCIM) protocols will help establish a consistency in customer status (KYC tier, sanctions flags, eligibility to buy digital asset products) throughout the fiat and digital surface.

#### 4.3. Orchestration Layer

Orchestration layer is the nerve system of the platform. The Digital Asset Orchestration Service orchestrates all of the state-changing transactions for all downstream bounded contexts. It receives a validated API call, then calls the Pricing Service for fresh quote, then calls the Risk and Compliance services for KYC/AML/limit checks, then calls the Trade or Transfer subsystem for execution and finally calls the Tax and Portfolio subsystems for post-execution state updates. It keeps a state machine per transaction and is able to implement compensating actions in case of a transient failure from a downstream subsystem. Orchestration service is designed using a saga-style model of transactions, which allows for compensating transactions and controlled recovery from partial failures.

A Merchant Connector that supports the inbound traffic from Web3 partners outside the platform and Merchant payout, which converts partner-specific protocols to the platform's internal canonical event model. The architecture ensures that no transaction flows through the architecture without passing through the compliance checkpoints, whether it is entered through the front door or the back door of the Orchestration Service.

#### 4.4. Digital Asset Core Platform

The core platform consists of 4 bounded contexts. Trade Platform is the system of record for all buy, sell and convert operations and is integrated with the main custodial and liquidity partner through the Outbound Connector Framework. The Transfer Platform processes internal book-transfers between PSP wallets, external on-chain deposits and withdrawals, and will require address pre-screening against blockchain analytics services [18]. The Portfolio Tracker provides users with a real-time, single source of truth for their holdings, valuations and history, and feeds into activity views, downstream analytics and tax computation. The Subscription and Recurring Manager processes recurring and scheduled transactions, also competing price band and insufficient funds cases.

#### 4.5. External Partner Integrations

The regulated partners are outside the platform boundary, and rely on the PSP to provide the functions that the PSP does not provide. The custodian offers important management and settlement on-chain and prudentially regulated liquidity. The tax partner supplies the cost basis tracking along with the forms. The partner for analytics and AML offers on-chain screening and risk scoring. The Stand-In source is a source of the market-data partner who supplies redundant pricing. Each partner can be accessed through a dedicated connector module in the Outbound Connector Framework, which standardizes the authentication, retries with exponential backoff, idempotency token, handling of retries, and audit logging. The framework also features circuit-breaking and health-monitoring features to ward off cascading failures with external systems.

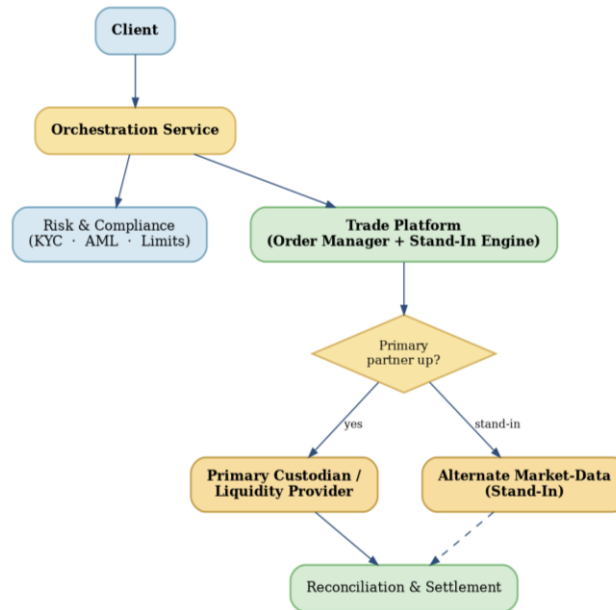
#### 4.6. Data and Infrastructure Layer

The data and infrastructure layer is where your relational stores of transactional state, reporting analytical warehouse, service-to-service asynchronous messaging (event bus), and a single observability stack (metrics, distributed traces, structured logs) for all the bounded contexts reside. Other resiliency and disaster recovery primitives exist at this layer, including multi-region replication, automated failover and periodic snapshot validation. Transaction replay, compliance investigations and reconciliation workflow depend on the single source of truth: the event store.

### 5. Trade Platform and Stand-In Execution Model

The trade execution architecture and Stand-In continuity mechanism is illustrated in Figure 2. The Orchestration Service processes the User's orders (Buy, sell, convert) and accesses the market's quotes, performs risk and compliance checks and submits the orders to the Trade Platform for execution through the primary custodian. Stand-In Engine monitors partner health at all times and, if they go over service thresholds, will attempt to run the second path using alternative market data. While the compliance control is active, transactions are treated as provisional commitments, based on the mirrored ledger balances, and are committed in case of any

outages. After the primary partner has recovered, the Reconciliation and Settlement subsystem reconciles provisional transactions, resolves discrepancies and updates final settlement records. Thus, transactions are accessible, yet additionally hearable and meet regulatory requirements.



**Figure 2. Trade Execution Architecture Incorporating the Stand-In Continuity Mechanism**

The Trade Platform supports both primary-partner execution and provisional execution during partner outages, with subsequent reconciliation and settlement once normal operations are restored.

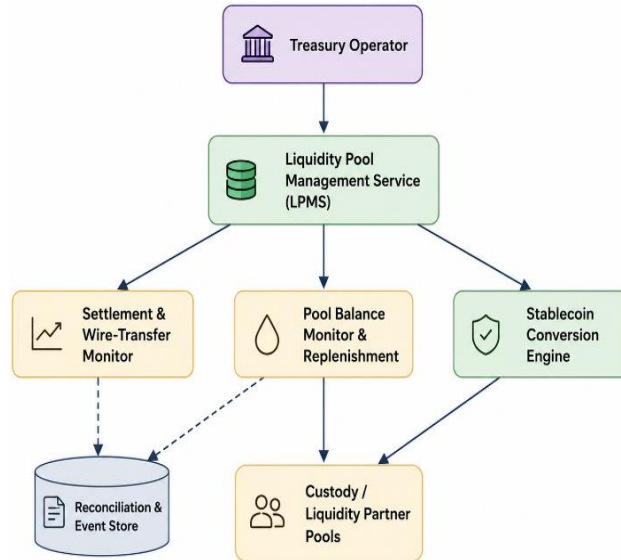
The proposed design is not a traditional failover design (the majority of them are basically about the infrastructure's availability), but rather an architectural approach of the business level that is designed for trade continuity. Partner outages are not uncommon in the regulated digital asset services industry because these services rely on third parties for their Custodial services and liquidity. The architecture thus provides a Stand-In execution model, ensuring compliance, auditability and reconciliation guarantee for transactions.

The Stand-In Engine is a major feature of the proposed Trades system, which will allow players to trade without being connected to the Primary Partner. The platform does not reject trades when the custodian is down, but keeps track of trade orders that were made for the trade to be done at the partner's recovery, and logs any trade order commitments that were done at the partner's recovery. This will facilitate user experience and maintain consistency of the reconciliation and regulatory controls.

The Stand-In mechanism mimics failover as a standard operational process, instead of an exception. All transactions in Stand-In mode are then reconciled in the Reconciliation and Settlement subsystem, which verifies the settlement results, ensures that there are no pricing errors, and enforces customer protection rules as applicable. This allows for ongoing trading availability and ensures settlement integrity and regulatory traceability.

## 6. Liquidity Pool Management Architecture

Large-scale digital asset platform requires a certain amount of liquidity in each pool, as well as regulatory separation and transparency in operation, and efficient settlement, etc. The Liquidity Pool Management Service (LPMS) is the Liquidity Pool treasury control plane that is responsible for monitoring pool balance, coordinating replenishment, handling stablecoin conversion and reconciling custodial records, as shown in Figure 3.



**Figure 3. Liquidity Pool Management subsystem**

The LPMS provides multi-pool treasury operations, automated USD-to-stablecoin conversion, settlement and wire-transfer monitoring, and reconciliation against custodial partner records.

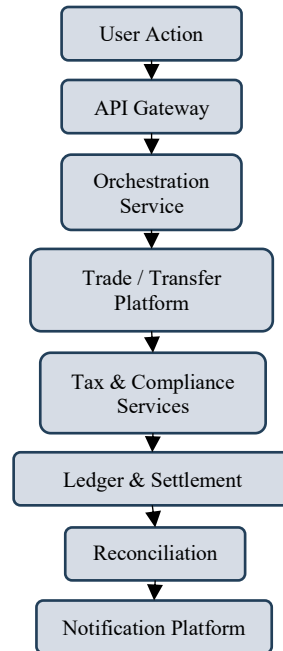
Under normal operation, the vast majority of pool management is automated, but LPMS provides a Treasury Watch Tool to authorized treasury operators to be inspected, configured and manually operated. A Master Pool Account Service keeps the "canonical" pool balances and their accounting attributes (who, what, where, when, and who can access). The Pool Balance Monitor will constantly poll the on-chain and trust account balance from the custodial partner, calculate the drift from the canonical book and initiate replenishment processes if a balance threshold is breached. A Stablecoin Conversion Engine enables automatic and manual conversion of the fiat reserves to a regulated stablecoin (and vice versa), based on the treasury rules, taking into consideration the volume of demand in the pool, forecasted activity of users and requirements for attesting the reserves [19].

The off-chain leg of adding funds to a pool is performed by a Settlement and Wire-Transfer Monitor, which monitors the status of a bank transfer between a PSP bank account and the custodial partner. When transfers are confirmed, the LPMS directs the custodian to mint, transfer, or convert the on-chain balance, and the resulting event is added to the Reconciliation and Event Store. Reconciliation occurs on an ongoing basis, not end-of-day: if there is any discrepancy between what they expect and what they see in the pool, an alert is sent through the Notification Platform, and an investigative workflow is kicked off within the case-management system in their treasury team.

LPMS's key architectural strength is the clear separation of treasury management and the customer-facing transaction processing. The eligibility and liquidity policies are exposed on the orchestration layer instead of directly interacting with the pool balances, as is done with trading and transfer services. This segregation helps to mitigate risks from transaction workflows to the fluctuating state of the treasury, allowing treasury operations, reconciliation and liquidity controls to adapt and evolve separately. This enhances the operational resilience, treasury governance, auditability and scalability of architecture while also ensuring that there are no unhelpful dependencies in user-facing services.

## 7. Event-Driven Transaction Flow and Consistency Model

The platform uses end-to-end event-driven architecture to ensure coordination of transaction processing between distributed services. The user-to-notification lifecycle for a transaction that produces a structured domain event that is published to a common event bus is illustrated in Figure 4.



**Figure 4. End-To-End Transaction Flow**

Domain events generated at each processing stage are published to a shared event bus, supporting auditability, asynchronous processing, replay capability, and downstream service integration.

A user action—buy, sell, convert, send, receive—is recorded at the client tier and then ends up at the API Gateway as an authenticated request (stages 1-2). The Orchestration Service will check the request, order the required downstream operations and produce an orchestration-initiated event (stage 3). The Trade or Transfer Platform performs the operation, which is coupled to the custodial partner, the analytics partner for address screening (in the transfer case) and the pricing service as necessary (stage 4). Execution events generated in stage 4 are consumed by tax and compliance subsystems asynchronously, which execute in real time the calculation of gains/losses and AML monitoring (stage 5). The ledger and settlement subsystems update the user's mirrored wallet position and show the user fiat settlements in their PSP balance (stage 6). Reconciliation confirms that the datasets of the trade, custody, treasury and wallet are in balance (stage 7). Lastly, the Notification Platform sends a user-facing confirmation via the channel of their choice (stage 8).

There are three requirements to motivate the event-driven model. First, blockchain settlement, compliance checking and tax processing are run asynchronously, and there is no direct link between the blockchain and user-facing latency. Secondly, there are regulatory requirements for an unalterable and repeatable transaction activity record. Thirdly, a number of downstream consumers process the same transaction events independently, such as compliance, analytics, notification, and reporting services. A publish/subscribe architecture is a natural solution to these requirements and also provides loose coupling between services.

The consistency is similar to that of eventual consistency in bounded contexts, but the user's read-your-writes consistency. The Trade Platform's order ledger is a strongly consistent system of record for trades, while the Portfolio Tracker gets updated asynchronously, and can be stale for a few moments during peaks of activity. The orchestration layer ensures that the user's perspective on their most recent transaction is retrieved from a consistency-safe source, either directly from the Trade Platform or from a user's Portfolio Tracker read that is marked as 'caught up to the relevant transaction event'.

The event-driven model makes the main contribution in the separation of the execution of the transaction process from downstream operational and regulatory processes. The platform translates business activities into fixed domain events, allowing services to evolve independently, events to be scaled for consumption, business events to be replayed for recovery, and full auditability. Event-driven processing and bounded-context isolation ensure consistency suitable for regulated financial applications while ensuring flexibility for future product, compliance and partner integration.

## 8. Deployment Topology and Technology Considerations

Table I lists some technology categories that can be utilized in the proposed architecture. The design is technology-neutral and can be deployed on a range of enterprise platforms. The front-end channels are provided in a web and mobile application framework, and orchestration and core services are provided as independently deployable services. Infrastructure components have been selected at the enterprise level to meet organizational requirements and regulations for transactional persistence, analytical processing, distributed messaging, caching and observability.

**Table 1. Representative technology stack for a production deployment**

Layer	Representative Technology Category	Architectural Role
Client Layer	Reactive web frameworks, native mobile SDKs	User-facing access channels for digital asset services
Service Layer	Java-based microservices, Python analytical services	Business logic, orchestration, and workflow execution
Data Layer	Relational databases (e.g., Oracle, PostgreSQL)	Transactional system of record and persistent storage
Cache Layer	Distributed in-memory caching systems	Low-latency access to portfolio and pricing data
Analytics Layer	Cloud-native analytical warehouses (e.g., BigQuery)	Reporting, analytics, and regulatory data processing
Messaging Layer	Enterprise message brokers and cloud Pub/Sub services	Asynchronous event distribution and service decoupling
Observability Layer	Metrics, distributed tracing, structured logging	Operational monitoring, diagnostics, and audit visibility

The deployment architecture is built on a cloud-native multi-region deployment approach to achieve high availability, resilience and regulatory compliance [20]. Stateless services are containerized and horizontally scaled, and stateful services are replicated and then fail over according to their consistency level. Regional isolation boundaries aid in disaster objectives, while network segmentation helps isolate digital asset services from the overall PSP environment. This deployment model allows for independent scaling of orchestration, trading, transfer and portfolio workloads and provides operational isolation between key system areas.

The architectural value in the deployment topology is the separation of stateless application services, stateful data stores and external partner integrations. This design is also highly scalable, allowing for failures to be isolated and built out regionally, and securing and complying with the security and compliance requirements of regulated digital asset platforms. Services are distributed and deployable across multiple regions, and event-driven, i.e., each component of a service has a high volume of transactions, a high degree of operation continuity and auditability.

## 9. Architectural Analysis and Discussion

The section is dedicated to highlighting key aspects of architecture, including: Scalability, Resilience, Auditability, Design trade-offs, and Comparative advantages in regulated digital asset service markets.

- **Scalability Considerations:** The modular decomposition and service independence result in the ability of scalability. Separating trading, transfer, portfolio, treasury and compliance into different bounded contexts allows each one to scale on the basis of its trading workload. The event-driven communication model minimizes inter-service dependencies and enables the processing of high volumes of transactions asynchronously. Furthermore, the multi-tenant orchestration framework allows multiple brands/partner channels to share the same infrastructure, yet remain logically isolated and operate efficiently.
- **Resilience and Service Continuity:** Resiliency can be supported by service isolation, communication in an asynchronous manner, and multi region deployment strategies. The Stand-In execution mechanism gives the flexibility of executing a transaction provisionally when the primary partner is disrupted, with settlement to be deferred. The bounded contexts are isolated, meaning that if one fails, it will not impact on the other, and if one partner recovers, consistency will be preserved by automatic reconciliation mechanisms between the provisional and finalized transactions states.

- **Auditability and Regulatory Alignment:** The auditability and regulatory traceability is built throughout the entire transaction life cycle. Compliance controls are embedded in orchestration workflows, such as KYC, AML, sanctions screening, transaction-limit validation and more to enable policy enforcement at every access point. Second, it's event-driven, which means it leaves an immutable trail of the events in the transactions themselves, enabling end-to-end reconstructing of the users' actions for regulatory reporting, compliance reviews, operational audits and investigation.

### 9.1. Discussion

The architecture depicts the integration of regulated digital asset services with the large payment service providers in balance. It offers a high degree of flexibility for scaling, resilience, auditability, and compliance and regulation traceability, and also supports the separation of customer interfaces, orchestration logic and digital asset operations. The advantages of such systems come at a cost, as does event management, event reconciliation, and schema change management for distributed systems. Despite the advantages of abstraction afforded by the Outbound Connector Framework, reliance on external custodians, market data suppliers and compliance partners is also a significant consideration in operations. The architecture is designed to decrease dependency between the fiat and digital asset services, making them more maintainable and flexible than monolithic integration types. Eliminates duplicate identity, compliance and customer-management processes on separate digital asset platforms. Accordingly, the architecture offers a viable middle-ground that can offer the requisite balance between operational efficiency, regulatory consistency, service continuity, and adaptability in a regulated financial environment.

## 10. Conclusion and Future Work

Payment systems need to be innovative, robust and compliant with regulations as cryptocurrencies and stablecoins increasingly invade the payments world. The architecture that addresses these requirements is a multi-tiered orchestration architecture, multi-layered service decomposition, event-driven transaction processing, standardized partner integration and dedicated liquidity management. This design pattern includes customer-facing channels, orchestration workflows, digital asset services & infrastructure components for scalability, auditability, resilience and regulatory compliance. The Outbound Connector Framework facilitates smooth connections to interactions with custodians, compliance providers, market data services and tax platforms, and the Liquidity Pool Management architecture strengthens treasury oversight, settlement monitoring and reconciliation. The components of the architecture are a viable and flexible design for the delivery of regulated digital asset services in large payment service providers and other financial services. Prototype implementations, industry case studies and scenario-based architectural assessment of the architecture could be the topics of further studies. Other studies could examine multi-custodian routing systems, programmable compliance policies, AI policy management, automated compliance with regulation shifts, etc. Other extensions include integration with new Web3 technologies, tokenizing financial assets, account abstraction and programmable settlement networks. The enhancements could impact interoperability, efficiency and regulatory readiness within these digital asset ecosystems and improve interoperability, efficiency and regulatory readiness of these digital asset ecosystems.

## References

- [1] R. P. Sarode, D. G. Singh, Y. Watanobe, and S. Bhalla, "High-volume transaction processing in Bitcoin Lightning Network on blockchains," *Int. J. Comput. Sci. Eng.*, vol. 26, no. 4, pp. 445–458, 2023.
- [2] A. Nerella, P. Badri, and L. S. S., "Unified Finance: Balancing Mobile Wallets and Traditional Payment Methods," *J. Electr. Syst.*, vol. 17, no. 4, pp. 157–165, 2021.
- [3] D. Patel, "Explainable Risk Decision Systems Using Artificial Intelligence Models for Payment Fraud Identification with Mitigation," in *2026 14th International Symposium on Digital Forensics and Security (ISDFS)*, Boston, MA, USA: IEEE, 2026, pp. 01–06, April. doi: 10.1109/ISDFS69419.2026.11459006.
- [4] Y. Li, Y. Xiang, Q. Wang, T. H. Yuen, A. Deppeler, and J. Yu, "SoK: Stablecoins in Retail Payments," 2026.
- [5] D. W. Arner, R. Auer, and J. Frost, "Stablecoins: Risks, Potential and Regulation," *SSRN Electron. J.*, pp. 1–31, 2020, doi: 10.2139/ssrn.3979495.
- [6] A. B. Chatterjee, "Designing Zero-Downtime, Cloud-Native Transaction Processing Architectures for 24x7 Global Payment Networks," *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. 7, no. 1, pp. 137–145, Feb, 2026, doi: 10.63282/3050-9246.IJETCSIT-V7I1P120.
- [7] H. Wu *et al.*, "Blockchain for finance: A survey," *IET Blockchain*, vol. 4, no. 2, pp. 101–123, 2024.
- [8] H. N. Dholariya, "GVIF: A Governed Vector Intelligence Framework for AI-Driven Cloud Data Modernization in Regulated Financial Systems," *Int. J. Comput. Exp. Sci. Eng.*, vol. 12, no. 1, Jan. 2026, doi: 10.22399/ijcesen.4797.
- [9] D. Zetzsche and A. Nikolakopoulou, "Crypto Custody: An Empirical Assessment," *J. Financ. Regul.*, vol. 11, no. 1, pp. 73–97, 2025, doi: 10.1093/jfr/fja004.
- [10] G. Liao, T. Hadeed, and Z. Zeng, "Beyond Speculation: Payment Stablecoins for Real-time Gross Settlements," *SSRN Electron. J.*, 2023, doi: 10.2139/ssrn.4476859.

- [11] J. Chen, J. Liu, Y. Liang, and M. Zhou, "HeteroGCL: A Heterogeneous Graph Contrastive Learning Framework for Scalable and Sustainable Cryptocurrency AML," *Appl. Sci.*, vol. 16, no. 6, 2026, doi: 10.3390/app16062860.
- [12] I. Staley, "A Reference Architecture for AI Agents on Blockchain Infrastructure: Identity, Policy, Payments, and Custody as Composable Primitives," 2026. doi: 10.2139/ssrn.6650658.
- [13] E. M. Cruz, J. R. D. S. Júnior, Y. H. J. Souza, G. L. S. S. Jesus, and M. L. M. Peixoto, "ArchW3: An adaptive blockchain wallet architecture for Web3 applications," *Comput. Networks*, vol. 262, p. 111182, May 2025, doi: 10.1016/j.comnet.2025.111182.
- [14] I. Homoliak, S. Venugopalan, D. Reijsbergen, Q. Hum, R. Schumi, and P. Szalachowski, "The Security Reference Architecture for Blockchains: Toward a Standardized Model for Studying Vulnerabilities, Threats, and Defenses," *IEEE Commun. Surv. & Tutorials*, vol. 23, no. 1, pp. 341–390, 2021, doi: 10.1109/comst.2020.3033665.
- [15] S. Shivam, T. P. Patel, A. K. Padhy, C. Kulkarni, C. Medicherla, and V. Soni, "Anomaly Detection in Financial Payment Transactions Using Efficient Data-Driven Machine Learning Techniques," in *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*, 2026, pp. 1–6. doi: 10.1109/ICAIC67076.2026.11395758.
- [16] G. B. Gorton and J. Y. Zhang, "Taming wildcat stablecoins," *Univ. Chicago Law Rev.*, vol. 90, no. 3, pp. 909–972, 2023.
- [17] E. Ferry, J. O Raw, and K. Curran, "Security evaluation of the OAuth 2.0 framework," *Inf. Comput. Secur.*, vol. 23, no. 1, pp. 73–101, 2015.
- [18] P. Tasca and C. J. Tessone, "Taxonomy of Blockchain Technologies. Principles of Identification and Classification," *arXiv Prepr. arXiv1708.04872*, Apr. 2018.
- [19] M. Mita, K. Ito, S. Ohsawa, and H. Tanaka, "What is Stablecoin?: A Survey on Its Mechanism and Potential as Decentralized Payment Systems," 2020.
- [20] A. Warriar, "Managing Complexity with Multiple API Gateways," *J. Artif. Intell. Mach. Learn. Data Sci.*, vol. 3, no. 3, pp. 2907–2913, Sep. 2025, doi: 10.51219/JAIMLD/Arjun-warriar/606.