Whitepaper

Clearing & Settlement Modernization for Card Payments on AWS

Contents

1. Introduction

In recent years, significant advancements in payment modernization strategies have been witnessed across acquirers, issuers and card networks. These initiatives aim to enhance business growth and streamline payment acceptance for customers. Some of the key modernization initiatives include new payment orchestration flows supporting enterprise merchants, least-cost routing, integration of alternate payment methods, strong fraud prevention measures, automated chargeback dispute management, and enhanced loyalty programs.

According to the Federal Reserve's 2023 survey,[i] the majority of businesses prioritize faster payment options. Faster clearing and settlement cycles enable merchants to improve cash flow by receiving funds sooner, which facilitates quicker reinvestment in inventory and operations, reduces chargeback risks, and enhances overall customer experience. In a competitive market, expedited fund availability provides a strategic advantage.

Key players in the payment ecosystem, including card networks, issuing banks, acquiring banks, and processors, must identify opportunities to reduce costs and improve cash flows. Modernizing pricing engines and clearing and settlement platforms is crucial to introducing agile strategies for a quick turnaround time and leveraging technologies such as artificial intelligence (ML) and generative AI.

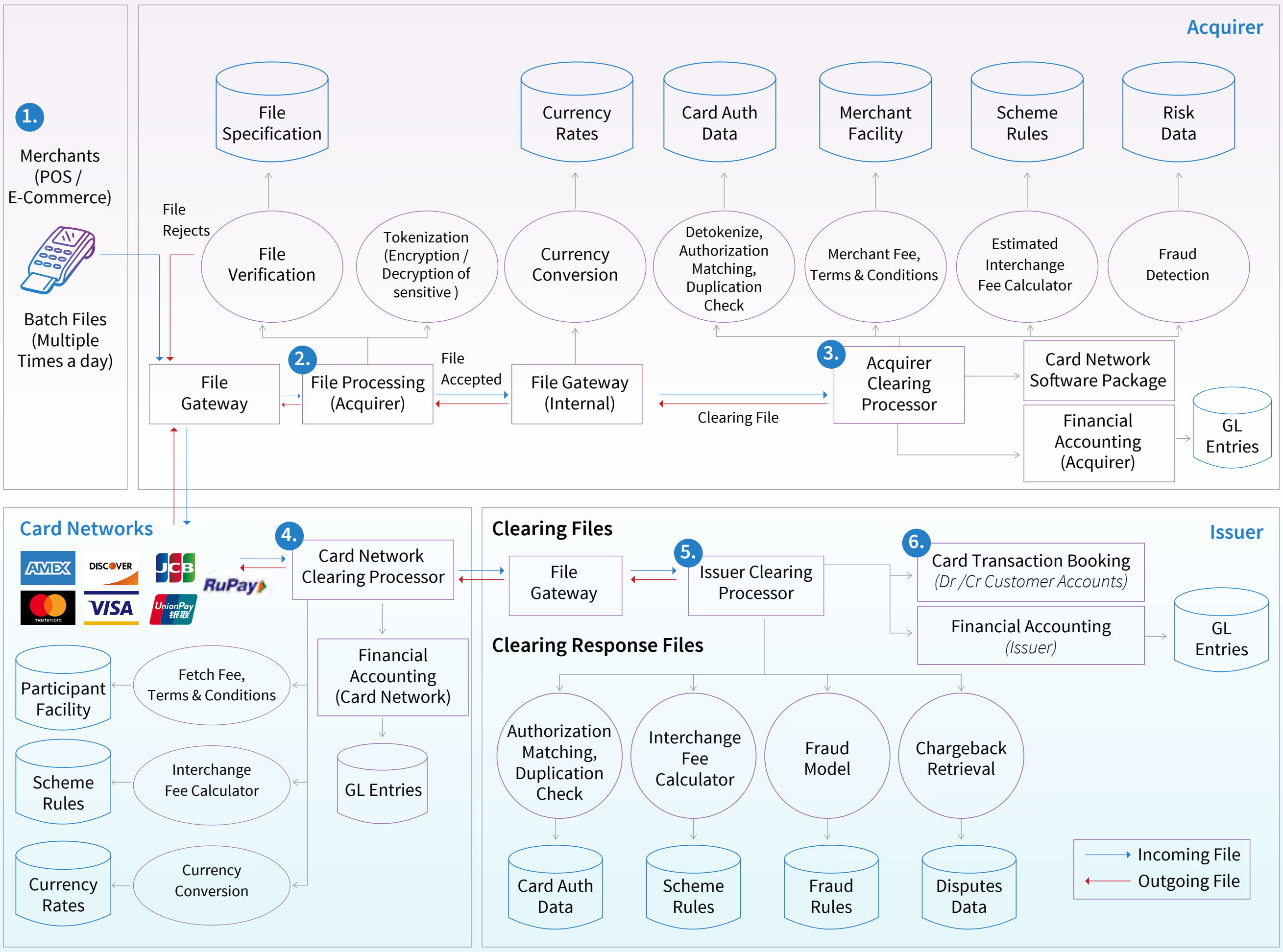
Non-traditional payment players like Stripe, Apple, Intuit, and others, are coming up with solutions which are transforming the way payments are conducted. For example, Stripe provides tools to facilitate money movement, store funds in financial accounts, and help businesses make more money.

This whitepaper provides guidance on how the clearing and settlement platform can be modernized by leveraging distributed technologies and Amazon Web Services (AWS) cloud infrastructure.

2. Clearing Functions

Credit card payments typically involve dual-message transactions processed in three primary stages: authorization, clearing, and settlement. Authorization verifies the availability of funds and transaction approval in real-time with the issuing bank. Clearing consolidates authorized transactions sent to an issuing bank for reconciliation, while settlement transfers funds to the merchant's account based on card scheme standards. Below is an outline of a functional flow of the clearing process:

(Federal Reserve Payments Insights Brief, 2023)



2.1 Merchant Batch File Initiation

Merchants submit batch files to acquirer multiple times daily for settlement, initiating the acquirer's clearing process.

2.2 Acquirer File Processing

- Incoming files from merchants may arrive in various formats. These files are transformed into a standard acquirer-specific file format for further internal processing. Files that do not conform to expected file specifications are rejected upfront by the acquirer.
- Any sensitive data, such as Personally Identifiable Information (PII) including card numbers, is tokenized before being sent for the clearing process. Similarly, once processing is completed, the file is decrypted before the outgoing clearing file is sent to the card network.

2.3 Acquirer Clearing Processing

- The transactions received from the merchant are matched against the acquirer authorization systems and merchant facility details. Data such as card number, transaction date, and amount are matched.
- Fee computation is done by the acquirer, which includes the interchange fee, assessment fees for card schemes, and acquirer fees/discounts based on products signed up for, subscribed to, or paid for by merchants.
- Transactions in foreign currency are converted using currency rates provided by the card association to the acquirer through daily feeds.
- The clearing file is created in a format supported by card associations.
- The acquirer also passes accounting entries in their general ledger system to track the amount claimed by merchant, fee amounts, disputes, chargeback amounts, etc.

2.4 Card Network Processing

- Upon receiving the clearing file from the acquirer, the card network validates the files to ensure the interchange fee, currency conversion, and amounts are correct. Finally, the network forwards the clearing file to the issuing bank.

2.5 Issuing Bank Processing

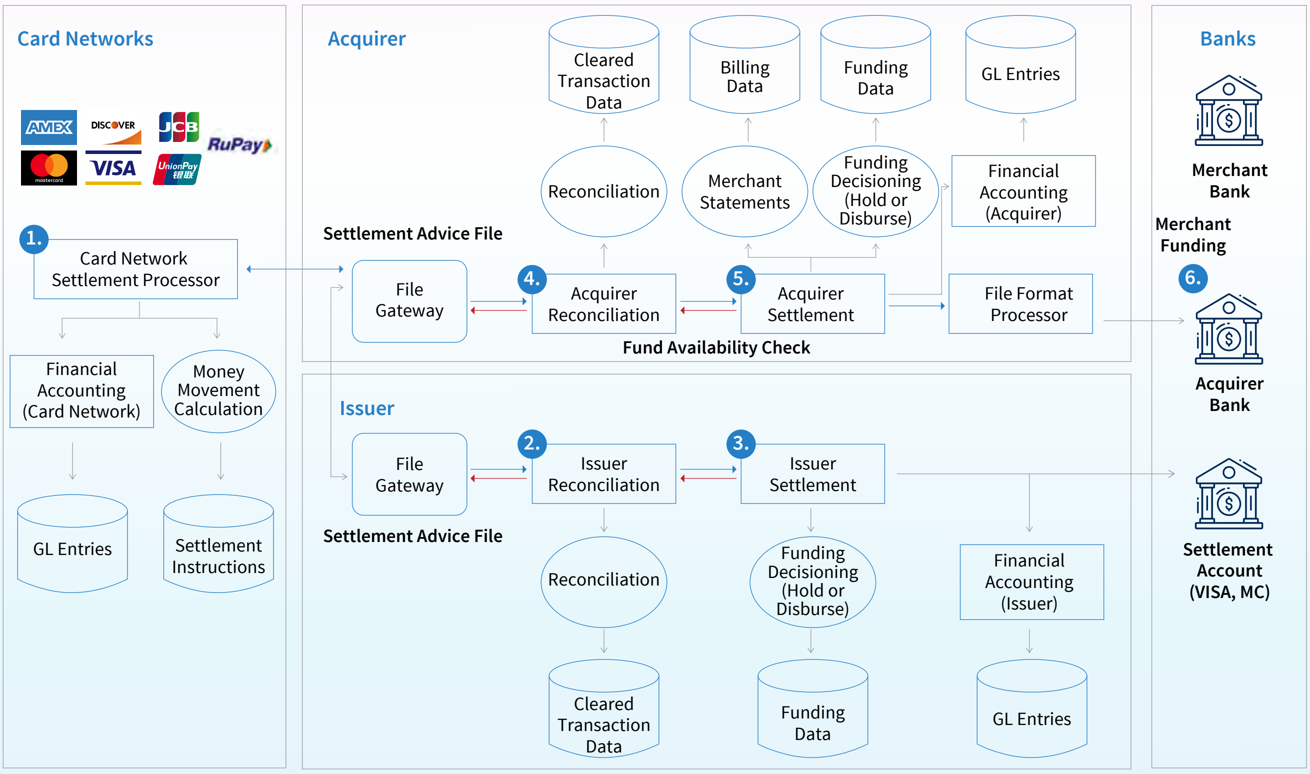
- The issuer processes the clearing file by verifying the transactions against their transaction authorization database. Fraud verification is done using fraud models, and the interchange fee amount is calculated to verify against the clearing data. The issuer also sends chargeback and retrieval data for disputed transactions to the association, which then goes to the acquirer.
- Once verified, the issuing bank also passes accounting entries in their general ledger system to track the amount claimed by the acquirer, fees to be received, disputes, chargeback amounts, etc.

2.6 Transaction Booking

- The issuing bank then books the transaction as a debit or credit to the cardholder accounts for collection.

3. Settlement Functions

Below is an outline of a functional flow of the settlement process:



3.1 Card Network Processing

- Upon receiving confirmation from the issuer, the card network initiates the settlement process, which involves calculating the money movement from the issuer to the acquirer. The card network then initiates the settlement advice file to the issuer.

3.2 Issuer Reconciliation

- Upon receipt of the settlement advice file, the issuer retrieves the clearing transaction data for reconciliation

3.2 Issuer Reconciliation

- Settlement is performed once reconciliation is successful, and the issuer passes entries to its general ledger system. Once this step is completed, the issuer remits the amount to the settlement account held by the card network.

3.3 Issuer Settlement

- Settlement is performed once reconciliation is successful, and the issuer passes entries to its general ledger system. Once this step is completed, the issuer remits the amount to the settlement account held by the card network.

3.4 Acquirer Reconciliation

- Reconciliation is initiated once the acquirer receives the settlement advice from the card network. The acquirer performs reconciliation to verify the data against their clearing records.

3.5 Acquirer Settlement and Statements

- The acquirer creates a funding file with instructions of how much money needs to be paid to each merchant. In cases where a payment processor serves as a facilitator for an acquiring bank, funding files may need to be produced in multiple formats depending on the acquirer bank's requirements. The file processor converts funding files to the target format and sends them to the acquirer bank. The acquirer also passes entries to their general ledger system to track the amounts cleared and settled.
- The acquirer then generates merchant statements and sends feeds to other downstream systems

3.6 Merchant Funding

- The acquiring bank then remits the funds to merchant accounts according to the instructions from the acquirer and reports received from the card network.

4. Reasons to Modernize

According to a recent McKinsey article,[ii] there is $3 trillion worth of business value at stake for companies that successfully use cloud technology. Legacy platforms, like mainframe systems, still run many business-critical applications and offer some value. However, these systems also pose several challenges:

Redundant applications – In the past decade, there has been significant consolidation in the payments industry, resulting in issuers, acquirers, and card networks having duplicate IT platforms. Consolidating these redundant systems with new distributed technologies will be important to decompose components and adapt to rapidly changing requirements.

Speed to market – Legacy systems can be tightly coupled applications, which, when building new features or introducing new integrations, require longer cycle times.

Cost – There may be higher costs associated with third-party licensed products and mainframe systems, which may generate higher million instructions per second (MIPS) consumption costs.

Data insights – Moving away from outdated legacy databases to cloud transactional data lakes can help in implementing new technologies to explore data, leverage analytics, drive superior reporting, and process the data for AI/ML and generative AI use cases. These use cases include fraud prevention, personalized recommendations based on purchasing behavior, and credit decisioning.

Talent availability – Skills in mainframe programs like Common Business Oriented Language (COBOL) are quickly becoming less desirable and difficult to find as organizations accelerate the move to a cloud-based infrastructure.

Documentation – Lack of documentation has led to increased dependency on a small set of experts on such systems to triage and fix issues. The reliance on a few experts increases the risk if the expert leaves the company or is unavailable to address production failures.

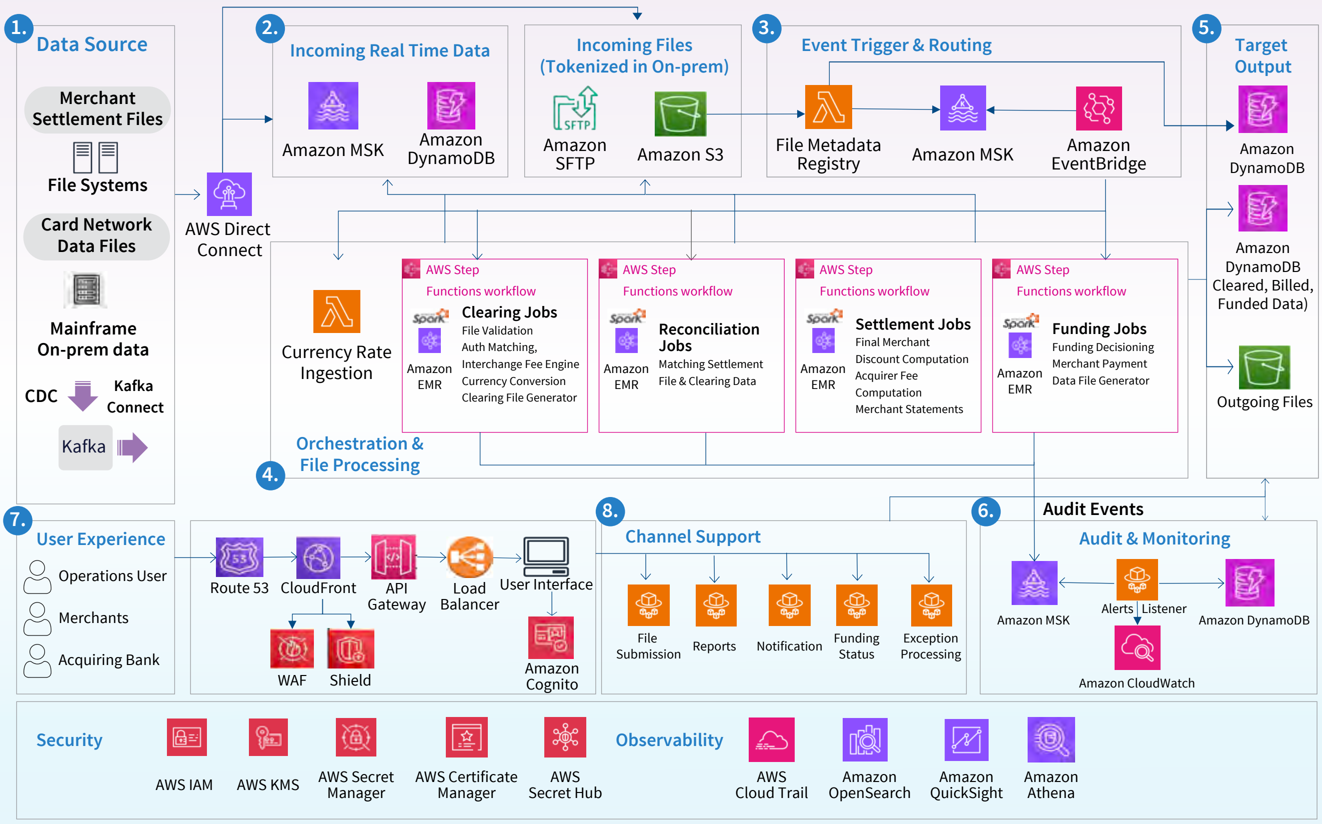
Continuous delivery and deployment – With rigid monolith architecture, it is difficult to deliver features for individual modules. Rebuilding features as microservices and serverless or containerized applications with CI/CD pipelines will enable more agility.

Modernizing clearing and settlement platforms on a distributed technology stack with AWS infrastructure can help organizations address several of these challenges:

- Cloud databases and large-scale distributed data processing can help organizations drive scalability and faster processing of batch jobs.
- AWS services such as Amazon MSK, Amazon EventBridge, Amazon DynamoDB, Amazon Aurora, Amazon S3, and Amazon EMR can be leveraged for a variety of use cases available with pay-per-use models, giving organizations control over costs. Cost optimization includes right-sizing compute workloads, the ability to scale down resources during non-business hours and applying lifecycle management policies to use the right storage classes.
- AWS customers can implement serverless computing with AWS Lambda or AWS Fargate for compute, Amazon Elastic Kubernetes Service (Amazon EKS) and Amazon Elastic Container Service (Amazon ECS) for containers, AWS Glue to perform data validation and quality checks, data extraction, transformation, and loading from multiple data sources, and Amazon Bedrock, a fully managed service that provides leading models to gain insights from large datasets.
- AWS offers data migration services to seamlessly and securely transfer data from on-premises data centers to AWS. Change data capture tools are available in the AWS Marketplace for real-time data replication. AWS managed streaming services can help implement event-based and data stream processing.

5. High-Level Reference Architecture

The high-level architecture diagrams below display the main components of an acquirer's clearing and settlement system and how it can communicate between different components when built on AWS.



5.1 Data Source

- Files coming from merchants, card networks, and other systems from on-premises servers are tokenized before being transmitted to AWS. Additionally, there are requirements for sourcing other data from on-premises servers.

5.2 Incoming Batch File and Real Time Data Transmission

- The below section describes some of the AWS services that can help achieve an automated solution for handling incoming files with varying volumes in a secure manner and process them in real time with event driven design.

Batch file transmission – As part of the modernization journey, there are incoming and outgoing files between merchants, card networks, and acquirers. File transfers are required both ways, from on-premises servers and AWS. To avoid any indirect methods such as intermediate staging servers, AWS Transfer Family with the Secure Shell (SSH) File Transfer Protocol (SFTP) can be leveraged to upload mainframe files directly to Amazon S3. Incoming files are persisted on Amazon S3 with lifecycle policies implemented to archive processed files to lower-cost tiers.

Event notification – Once the files are uploaded, Amazon S3 event notification can invoke an AWS Lambda function to persist the file metadata entries to the Amazon DynamoDB database and post a message to an AWS Managed Kafka topic for further processing.

File validation – Incoming files from on-premises servers are expected to follow a predefined format and tokenize any sensitive data before transmitting the files to AWS.

Change data capture (CDC) – With the modernization of such critical functions, the best approach is through an incremental manner by moving one domain capability to AWS at a time. There can be dependency on other components that are in an on-premises monolith system. CDC tools such as Kafka connectors available in AWS Marketplace can be leveraged for such data-sharing requirements. Incoming data can then be ingested into an AWS database of choice based on the data model and latency requirements.

5.3 Event Trigger and Routing

Events-Driven Architectures (EDA), with a publish/subscribe pattern, make it possible to decouple processing based on reaction to events and give consuming applications the ability to scale their applications without any impacts.

Amazon Managed Streaming for Apache Kafka (MSK) is a fully managed streaming service that can be used as a message broker responsible for the durability of events and the availability of events for consumption until successfully processed.

Amazon EventBridge provides a serverless event bus that is built for event-driven architectures. It can listen to events emitted by AWS services and can be extended to process custom events. Rules can be created that look for events on a bus and route them to the correct destinations. Amazon EventBridge provides the capability to pick events from an Amazon MSK topic, and based on the rules configured, it can process these events in a variety of ways such as making a direct API call or invoking AWS Lambda services or schedulers to trigger batch jobs in a sequence. Additionally, there are options to trigger the batch job as a parallel process with a combination of AWS Step Functions. To handle failures and for debugging, Amazon EventBridge can replay events with its archive and replay feature.

5.4 Orchestration and File Processing

Modern batch file processing is a critical building block in clearing and settlement processing as each of the players in the payment ecosystem sends and receives batch files at predefined windows. To achieve this, a scheduler is needed to integrate with the batch process.

AWS Step Functions can be used for orchestration of the workflows by defining job flows. The job poller patterns within AWS Step Functions can orchestrate the start, as well as monitor and report, the success or failure of a job. A job template is created using the AWS Step Functions' job poller pattern with parameterized options to submit any batch script.

Apache Spark, when used for batch processing, provides the ability to perform distributed data processing for big data volumes. Apache Spark natively supports applications written in Scala, Python, and Java and offers APIs enabling reuse of code across various workloads.

Amazon EMR offers a suitable cloud environment for deploying Apache Spark. With Amazon EMR, customers get managed services, eliminating the need for oversight by development teams in providing nodes, configurations, tuning, etc. Daily currency rate files from card networks land in Amazon S3, triggering an event to AWS Lambda, which then registers file details, such as metadata, to Amazon DynamoDB. Subsequently, an event is posted to a Kafka topic for Amazon EventBridge to invoke another AWS Lambda function to ingest the currency exchange rate to the database for the clearing process to reference the same.

Multiple Apache Spark jobs can be built to execute the clearing process to perform file validation, transaction matching against the authorization systems data, interchange fee calculations as per the card network rules, and producing clearing files in the specific card network schemes format.

Amazon Relational Database Service (RDS) supports multiple database engines such as MySQL, PostgreSQL, Oracle, Microsoft SQL Server, and Amazon Aurora, which can be chosen as targets to persist the processed data. The choice of the database stack will depend on the data model requirements, complexity involved, and other non-functional requirements.

Similarly, Amazon EventBridge can invoke reconciliation jobs, settlement jobs, and funding file creation jobs depending on the type of events received from each domain capability. Output files that are sent back to merchants as acknowledgments, clearing files, and chargeback files are stored in Amazon S3 and sent back to on-premises servers with tokenization applied to sensitive data elements.

5.5 Audit and Monitoring

Event driven architecture provides benefits such as improved performance, scalability, availability, resiliency, and ease of development. However, there are also audit and monitoring functions to consider:

- Release and deployment processes should be carefully designed and automated.
- Monitor the end-to-end workflows to create alerts when a signal's value crosses specified thresholds.
- Implement event ordering and exception handling to avoid loss of any event.
- While the clearing and settlement jobs process the files, they generate audit events for each transaction, which are sent to a Kafka topic. AWS Lambda polls these audit events, persists them to an event store in Amazon DynamoDB, and routes them to Amazon CloudWatch. Amazon CloudWatch Events is then configured to trigger actions for any environmental issues. Additionally, other backend processes can reconcile events from the event store and raise alerts if any events are missing.

5.6 Internal and External Users

The front-end application needs to meet the requirements of various stakeholders, including:

- The operations team needs to track system health, receive actionable alerts, and replay events to handle ad hoc files or exceptions.
- Merchants require real-time funding status updates, quick access to reports, and data on chargeback transactions.
- Users from acquirer banks rely on operational reports and access to chargeback retrievals.

All static content for the web applications, including single-page application frameworks, can be hosted on Amazon CloudFront. Users access these applications, download resources to their browsers, and connect to the backend using APIs. User authentication is managed through Amazon Cognito user pools or other federated user providers. Amazon API Gateway integrates seamlessly with Amazon Cognito, providing user authentication. Once authorized, clients receive a JSON Web Token (JWT) for making API calls.

5.7 Channel Support Layer

The business logic layer utilizes AWS Fargate, a serverless compute engine for containers fronted by Amazon API Gateway REST APIs. This architecture encompasses multiple integrated services, with each AWS Fargate service addressing distinct user application needs. These services reside behind Amazon API Gateway and are accessible via API URL paths. Each service operates under its own IAM role, ensuring appropriate access to data sources. While Amazon DynamoDB serves as the primary data storage, other purpose-built AWS databases or storage services can be utilized based on specific use cases and scenarios.

Defining business needs and requirements is essential for successful implementation. Non-functional requirements (NFR) play a crucial role in meeting business expectations by outlining how a system should operate, complementing functional requirements. The AWS Well-Architected Framework provides key concepts, design principles, and best practices for constructing secure, reliable, efficient, cost-effective, and sustainable workloads on AWS. The architecture diagram provided exemplifies a solution aligned with Well-Architected best practices. Adhering to these practices enhances the overall robustness and effectiveness of AWS deployments. Some of these practices are identified below:

Operational excellence – The reference architecture leverages fully managed services like Amazon EMR and Amazon MSK, reducing maintenance efforts for complex distributed clusters and infrastructure. Efficient monitoring is achieved by streaming operational and audit events using Amazon MSK and Amazon CloudWatch, with configurable actions for environment issues and system exceptions.

Security – Security measures include tokenizing sensitive data before transmission to Amazon S3, utilizing AWS, KMS, and IAM policies for encryption. User authentication is managed via Amazon Cognito, with JWTs enabling secure API calls. AWS Certificate Manager helps provision and manage certificates, ensuring secure access for users.

Reliability – Amazon EMR facilitates easy provisioning and auto-scaling of Apache Spark clusters, while Amazon EventBridge ensures reliable event delivery and scheduling. EventBridge’s archiving and replay features enhance reliability across multiple Availability Zones.

Performance Excellence – Apache Spark’s in-memory caching and optimized DAG execution engine provide efficient data operations at scale. Optimized Lambda functions further reduce latency and increase throughput.

Cost Optimization – Amazon EventBridge’s flexible pricing and Amazon MSK’s auto-scaling and message compression features contribute to cost-effective solutions.

Sustainability – Using managed services and dynamic scaling minimizes the environmental impact of backend services and compute resources.

4. Conclusion

The role of a clearing and settlement system in card payments is critical, involving the exchange of funds between multiple parties within the ecosystem. Given the limitations of existing legacy platforms, acquirers, issuers, and card networks are increasingly seeking to modernize these systems. Modernization can drive benefits such as improved cash flow for merchants, enhanced merchant acquisition for acquirers, increased flexibility in the pricing engine, and cost optimization. The AWS reference architecture and best practices provide financial institutions with a framework to design and implement a modern solution iteratively, accelerating business outcomes.

For more information about collaborating with LTMindtree, please contact your LTMindtree Account Manager. Additionally, the AWS Well-Architected tool enables organizations to review the state of their workloads and compare them to the latest AWS architectural best practices.

Disclaimer:

The reference architectures discussed in this document are intended for illustrative and informational purposes only. They are based on the information available at the time of publication and are not meant to serve as comprehensive enterprise solutions. For tailored architecture that addresses your specific organizational needs, please contact us.

5. Citations

i. Federal Reserve, Fed payments Improvement, 2023:
https://fedpaymentsimprovement.org/wp-content/uploads/051823-business-research-brief.pdf

ii. McKinsey article, Mark Gu, James Kaplan, November 30, 2023:
https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/getting-ahead-in-the-cloud

7. Contributors

Amrit Raheja is the Managing Principal Architect at LTIMindtree, he leads the Technology Architecture function for Cards & Payments client relationships. He collaborates with customer executives to build modernization roadmaps for migrating from legacy systems to cloud microservices. With over 24 years of payments industry experience, Amrit has previously served as an engineering leader at Fiserv and held leadership positions at Citi and other software solution providers, developing card issuing, merchant acquiring & payment platforms on AWS.

Mark Smith leads Amazon Web Services' (AWS) worldwide business and market development efforts for the payments market segment of the Financial Services industry. In this role, he is responsible for the development and execution of AWS's strategic initiatives across a wide range of payment companies, including merchant acquirers, payment processors, payment networks, card issuing banks, payment service providers and payment fraud detection companies. He works with these organizations to help them transform their existing businesses and bring new, innovative solutions to the market by leveraging AWS services.

Paul Chang is the AWS Head of Card Payment Networks in Financial Services with 22+ years of experience in data analytics and product management. He works with leading banks and financial institutions to help them leverage AWS tools and services to drive innovation and build new digital products. In addition, he helps companies leverage business insights to win market share and drive growth in volume and net revenue.

Sudhir Kalidindi is an AWS Principal Solutions Architect in Financial Services with 22+ years of experience in software architecture and the development of solutions involving business and critical workloads. He helps payments customers to innovate on the AWS Cloud by providing solutions using AWS products and services.

About LTIMindtree

LTIMindtree is a global technology consulting and digital solutions company that enables enterprises across industries to reimagine business models, accelerate innovation, and maximize growth by harnessing digital technologies. As a digital transformation partner to more than 700 clients, LTIMindtree brings extensive domain and technology expertise to help drive superior competitive differentiation, customer experiences, and business outcomes in a converging world. Powered by 81,000+ talented and entrepreneurial professionals across more than 30 countries, LTIMindtree — a Larsen & Toubro Group company — solves the most complex business challenges and delivers transformation at scale. For more information, please visit **https://www.ltimindtree.com/.**