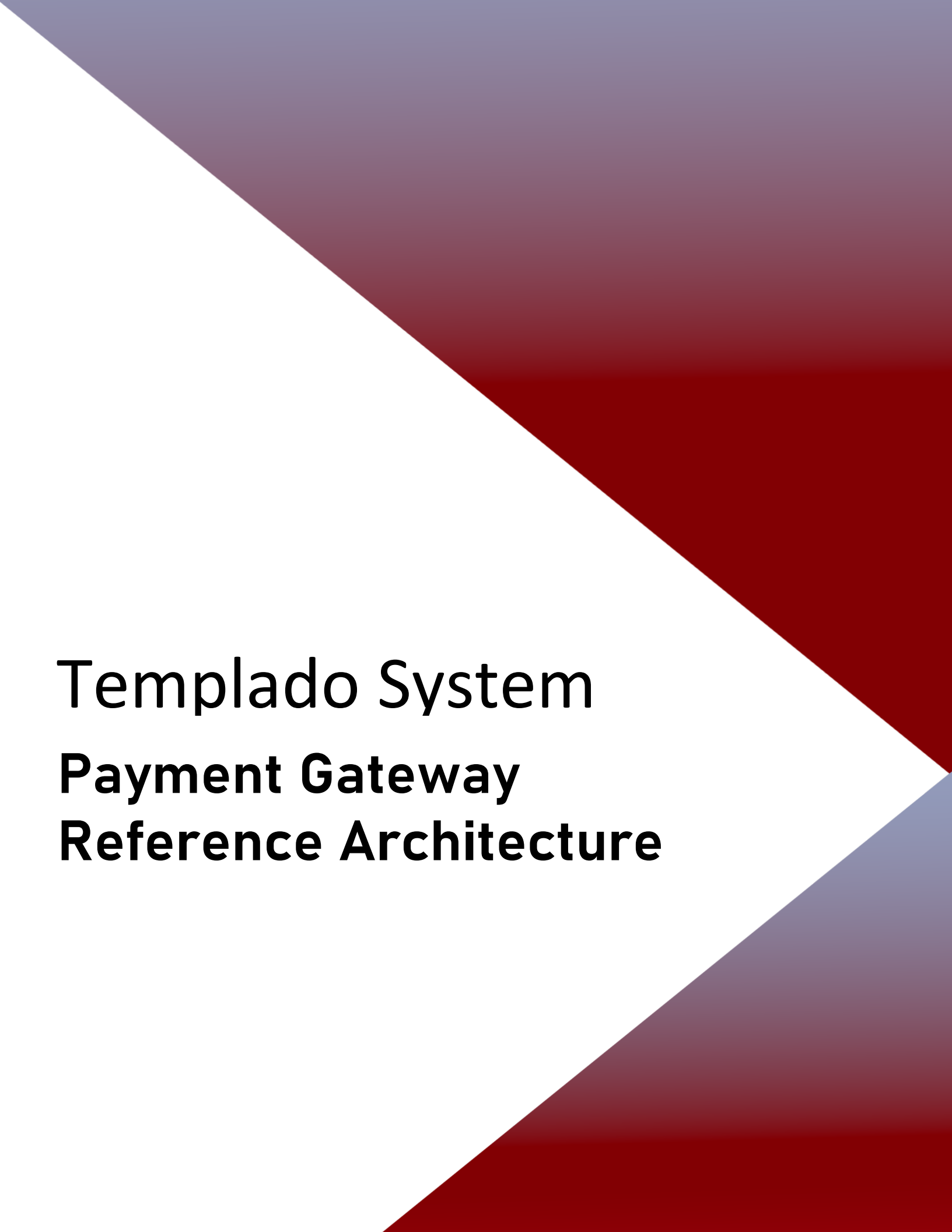




Templado System

Payment Gateway

Reference Architecture

Table of Contents

Introduction.....	2
<i>RISC-V.....</i>	<i>2</i>
<i>ARM.....</i>	<i>2</i>
<i>The Reference Architecture</i>	<i>3</i>
<i>Platform Component.....</i>	<i>4</i>
Web Gateway	4
Transaction Engine.....	4
API Parser	4
Fraud Prevention	5
Security.....	5
Database Engine.....	5
Recurring Billing	5
Error Handler	5
Acquiring Bank Communication.....	5
Merchant Portal.....	5
Payment Terminal.....	5
Admin Interface	6
<i>Merchant Models</i>	<i>6</i>
Direct Merchant	6
Aggregation Merchant.....	7
Hybrid Merchant	8
<i>High Risk Merchant.....</i>	<i>8</i>
Rolling Reserve (RR)	8
Chargeback	9
<i>Deployment.....</i>	<i>9</i>

Introduction

Payment gateways form the backbone of modern financial transactions, providing secure, reliable communication between customers, merchants, and financial institutions. As transaction volumes and security requirements continue to grow, the underlying compute architecture has become a critical design decision. This document describes how the Templado System payment gateway leverages RISC-V and ARM processors — two architectures that are reshaping server and edge computing — to deliver a secure, efficient, and scalable processing platform.

RISC-V

RISC-V is an open, royalty-free instruction set architecture (ISA) that has gained significant adoption in recent years due to its flexibility, extensibility, and suitability for customization. Unlike proprietary architectures, RISC-V places established vendors and emerging entrants on equal footing, which has accelerated innovation and cross-industry collaboration. Its open specification allows silicon designers to implement highly efficient, purpose-built processors for specific workloads — including payment processing.

Its flexibility, simplicity, and open governance have made RISC-V an increasingly popular choice in IC design. By adopting RISC-V processors, payment gateway providers can improve efficiency, security, and scalability across the transaction pipeline — from core transaction processing to fraud detection and prevention. This document describes how the Templado System platform applies these architectural advantages in production.

ARM

ARM is a reduced instruction set computing (RISC) architecture that has expanded from its origins in mobile devices into servers and data centers, where it now competes directly with mainstream x86 designs.

Price-to-performance is a primary factor driving ARM adoption in the data center. ARM processors deliver cloud computing performance comparable to x86 at a significantly lower total cost.

Organizations can reduce infrastructure costs across a broad spectrum of workloads without sacrificing performance, making ARM's price-to-performance ratio difficult for x86 architectures to match.

The Reference Architecture

Figure 1 illustrates the high-level architecture of the Templado System Payment Gateway.

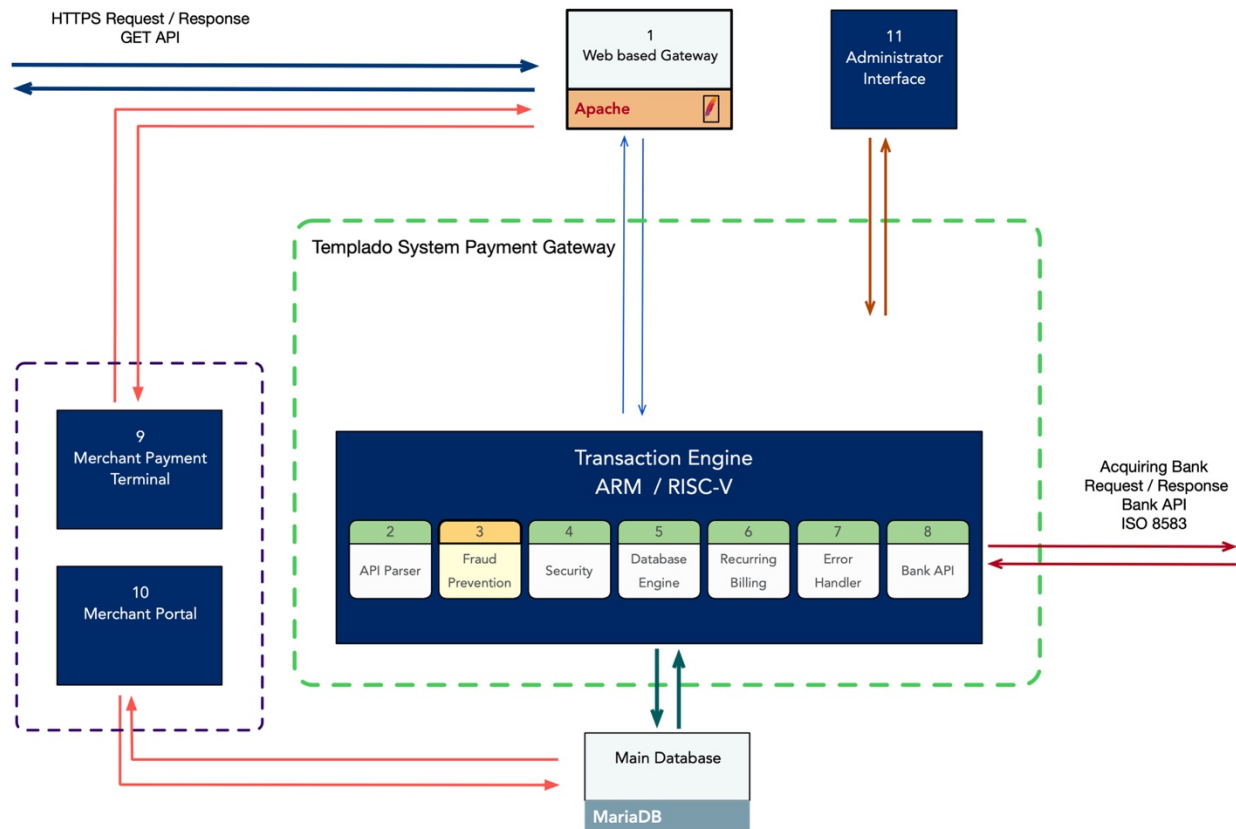


Figure 1

The platform was designed to be free of third-party licensing dependencies, such as Microsoft Windows Server or SQL Server, and is built entirely on open-source components. It runs on the Linux AArch64 platform (Ubuntu 22.04, Fedora ARM) with MariaDB as its primary database. Security is implemented with OpenSSL, which handles the cryptographic functions required for TLS/SSL termination, AES encryption of customer data, and key management.

Customers may substitute their own key management solution — for example, AWS Key Management Service (KMS) or a virtual HSM for cloud-based deployments, or a physical HSM, such as Thales or Futurex, for on-premises deployments.

Platform Component

This section describes the platform components where the core payment processing functions reside, including the web gateway, transaction engine, data storage layer, and merchant-facing modules.

Web Gateway

The web gateway is implemented on the Apache HTTP Server, which offers several advantages for a payment gateway deployment. As an open-source, freely licensed web server, Apache reduces infrastructure costs relative to commercial alternatives. It is cross-platform, running on Linux, Unix, Windows, and macOS, which allows the gateway to be deployed on a wide range of platforms depending on requirements. Apache natively supports TLS/SSL encryption to secure communication between clients and the server, and its modular architecture provides additional configuration options to mitigate common web server vulnerabilities. Apache HTTP Server runs natively on ARM-based systems, consistent with its cross-platform design.

Transaction Engine

The transaction engine is responsible for authenticating each transaction, verifying the merchant, performing fraud detection, selecting the appropriate acquiring bank, and selecting the network protocol or API used for bank authentication. It submits the request to the acquiring bank and, upon receiving a response, returns the result to the merchant through the merchant API. The transaction engine also performs all required database operations.

To fully exploit the ARM and RISC-V architectures, the transaction engine is implemented as a low-level program combining C/C++ with hand-optimized assembly. It is designed to scale across multi-core architectures, running on systems ranging from a 4-core ARM Cortex-A7x processor up to a 128-core Ampere Ultra CPU.

The transaction engine maintains a thread pool of 20 threads by default. On CPUs with more than 20 cores, thread utilization can be optimized in one of two ways:

Thread-per-core (1:1) — each available core is assigned a single thread from the pool, maximizing parallelism by letting every core execute independently. If the core count exceeds the thread count, the remaining cores stay idle.

Multi-threading — the engine spawns additional threads per core to fully utilize available resources; on a 40-core CPU, for example, it scales the pool to 40 threads accordingly. This scales to the full range of supported hardware, including the 32-, 64-, and 128-core Ampere Ultra ARM family and the 64-core SOPHGO SG2042 RISC-V processor, further increasing parallelism and throughput.

API Parser

The API parser module is responsible for parsing incoming API requests and extracting the fields required for downstream processing.

Fraud Prevention

The fraud prevention module detects and mitigates fraudulent activity across payment transactions. It applies a configurable rules engine to identify known fraud patterns and indicators, including velocity checks, IP blacklisting, address verification (AVS), and card verification value (CVV) validation.

Security

OpenSSL handles the platform's core security functions, including hashing, AES encryption and decryption, and TLS/SSL transport security.

Database Engine

The transaction engine communicates with the database through a C++ database library. Each processing thread maintains its own SQL connection, enabling database requests to execute in parallel.

Recurring Billing

The Templado System gateway supports recurring billing on a merchant-defined schedule, serving the needs of subscription-based merchants.

Error Handler

The error handler manages exception conditions across the system, including transaction timeouts, malformed API requests, failed merchant verification, and failed fraud checks.

Acquiring Bank Communication

Acquiring bank APIs are integrated directly into the transaction engine rather than through an abstraction layer. Because there is no universal standard for acquiring bank APIs — unlike the ISO 8583 message standard used across card networks — this direct integration was chosen to maximize processing speed. The trade-off is that changing acquiring partners requires updating and recompiling the integration within the engine, whereas some payment gateways add an abstraction layer specifically to avoid this. Templado System prioritizes simplicity and speed over that added flexibility.

Merchant Portal

The merchant portal is a web-based application that gives merchants a centralized interface for managing their payment processing activity — viewing and managing transaction data, tracking sales and revenue, and performing other day-to-day payment operations.

Payment Terminal

The payment terminal is a browser-based payment processing interface that allows merchants to accept payments without a physical point-of-sale (POS) device. Customers enter their payment information into a secure web form, which the gateway then processes directly.

Admin Interface

The admin interface is a web-based application used by the payment processor to manage platform-wide operations — monitoring and managing transactions, configuring payment settings, onboarding merchants, and performing other administrative tasks.

Merchant Models

Direct Merchant

In the Direct Merchant ID (DMID) model, a merchant holds a direct relationship with the payment processor and maintains its own merchant account. That account is linked to a unique DMID, which identifies the merchant for transaction processing and fund settlement.

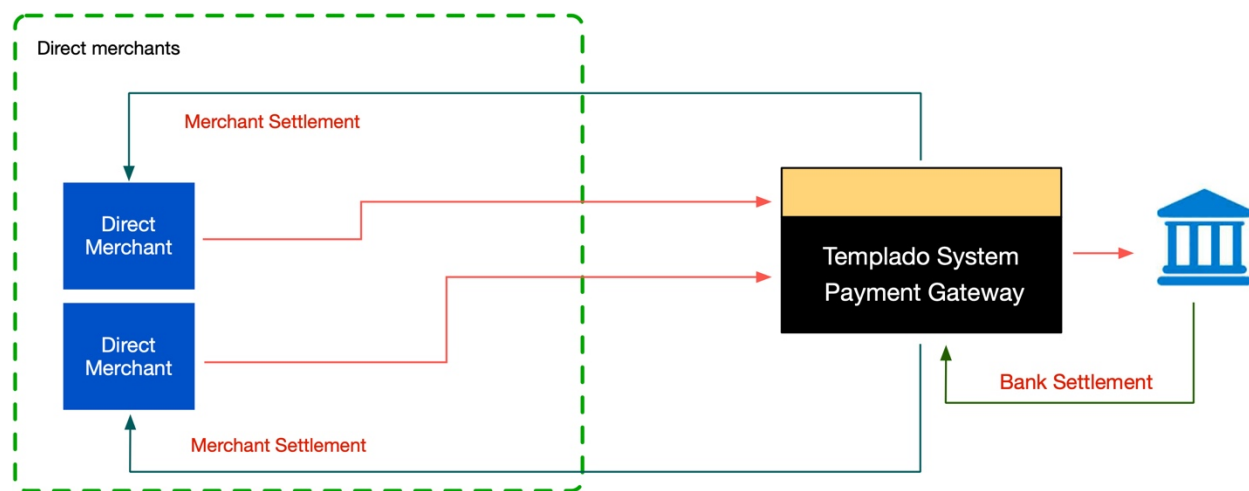


Figure 2. Direct merchant Model

Aggregation Merchant

In the merchant aggregation model, an aggregator acts as an intermediary between multiple sub-merchants and the payment processor. A Master Merchant ID (MMID) is assigned to the aggregator and shared across all sub-merchants operating under it, allowing the aggregator to manage payment processing on their behalf. When a customer pays a sub-merchant, funds are collected under the aggregator's MMID and subsequently disbursed to the appropriate sub-merchant account.

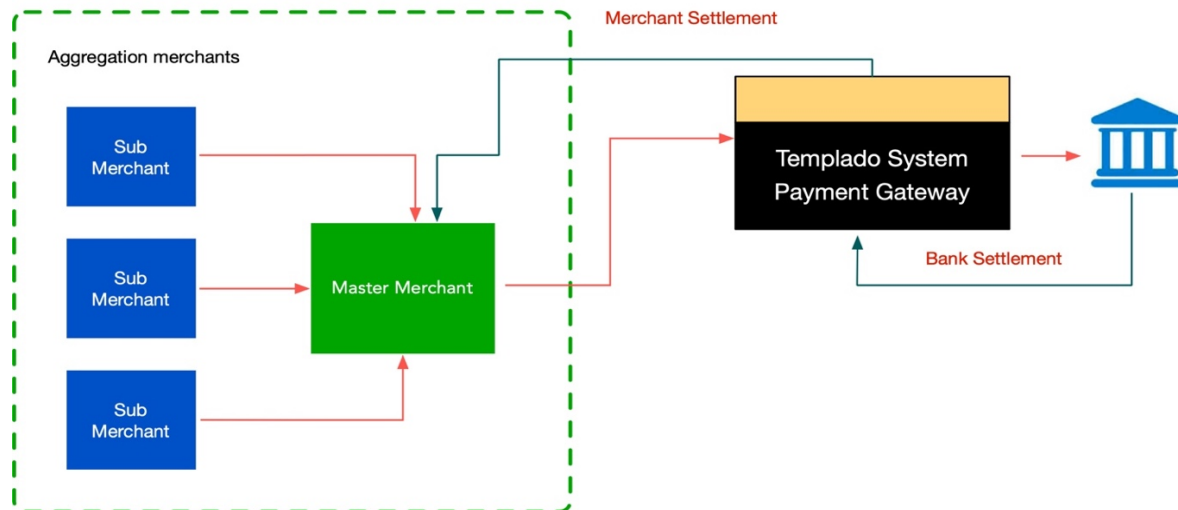


Figure 3. Aggregation Merchant Model

Hybrid Merchant

The hybrid model allows Templado System to combine direct merchants and aggregation merchants within a single platform deployment.

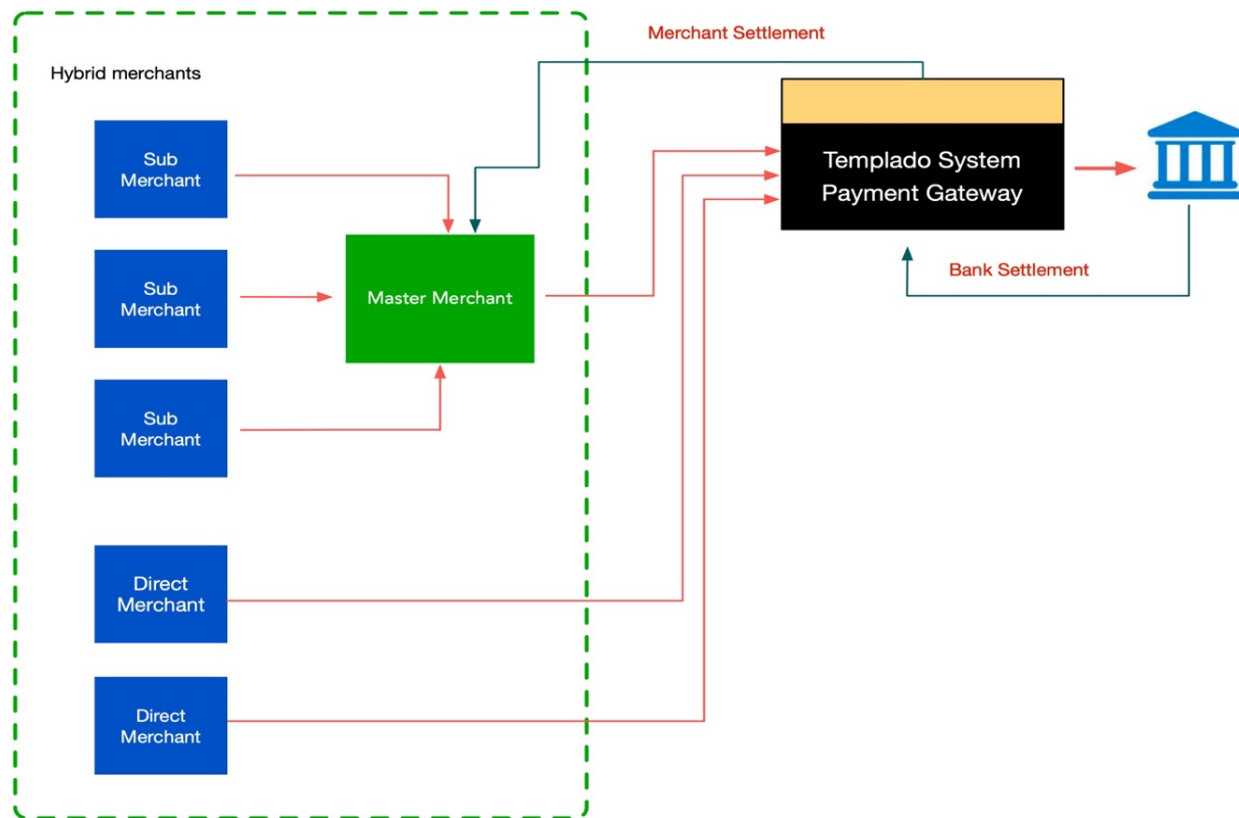


Figure 4. Hybrid Merchant Model

High Risk Merchant

The Templado System gateway includes additional features to support the requirements of high-risk merchants.

A high-risk merchant classification means that payment processors and card networks view the business as more likely to default on payments, incur elevated chargeback rates, or be exposed to fraud. High-risk businesses are typically newly established, have a history of credit issues, or operate in an industry considered volatile or controversial.

Rolling Reserve (RR)

A rolling reserve is a risk-mitigation mechanism used by acquiring banks to protect themselves and the merchant against potential losses from chargebacks. A percentage of each card transaction is withheld from the merchant's settlement and held by the bank against future chargeback risk. The Templado System gateway allows a rolling reserve to be configured per merchant, including the reserve period and the percentage withheld from settlement — typically 180 days and 10%, though the system administrator can adjust both parameters. The settlement module automatically calculates the reserve amount and its scheduled release.

Chargeback

The acquirer's exposure arises when a merchant becomes insolvent between the transaction date and the chargeback date, leaving it unable to reimburse the acquirer for the disputed amount. The Templado System gateway allows a chargeback fee and an acceptable chargeback percentage to be configured per merchant. The platform automatically calculates each merchant's chargeback ratio and flags accounts that exceed the configured threshold.

Deployment

The platform supports both on-premises and cloud deployment. As a cloud-native platform, it runs on the ARM-based compute instances offered by all major cloud providers: Amazon Web Services (AWS Graviton2 and Graviton3), Microsoft Azure (Ampere Altra), Google Cloud Platform (Tau T2A instances, built on Ampere Altra), and Alibaba Cloud (ECS g8y instances, built on the Yitian 710 processor).

On-premises deployments can run on any ARM- or RISC-V-based server.

The platform supports any Linux distribution built for the AArch64 (ARM64) or RISC-V architectures.