Software Architecture for a Virtual Card Financial System

Enabling Secure and Efficient Digital Payments

Modern Financial Systems: The Virtual Card Advantage

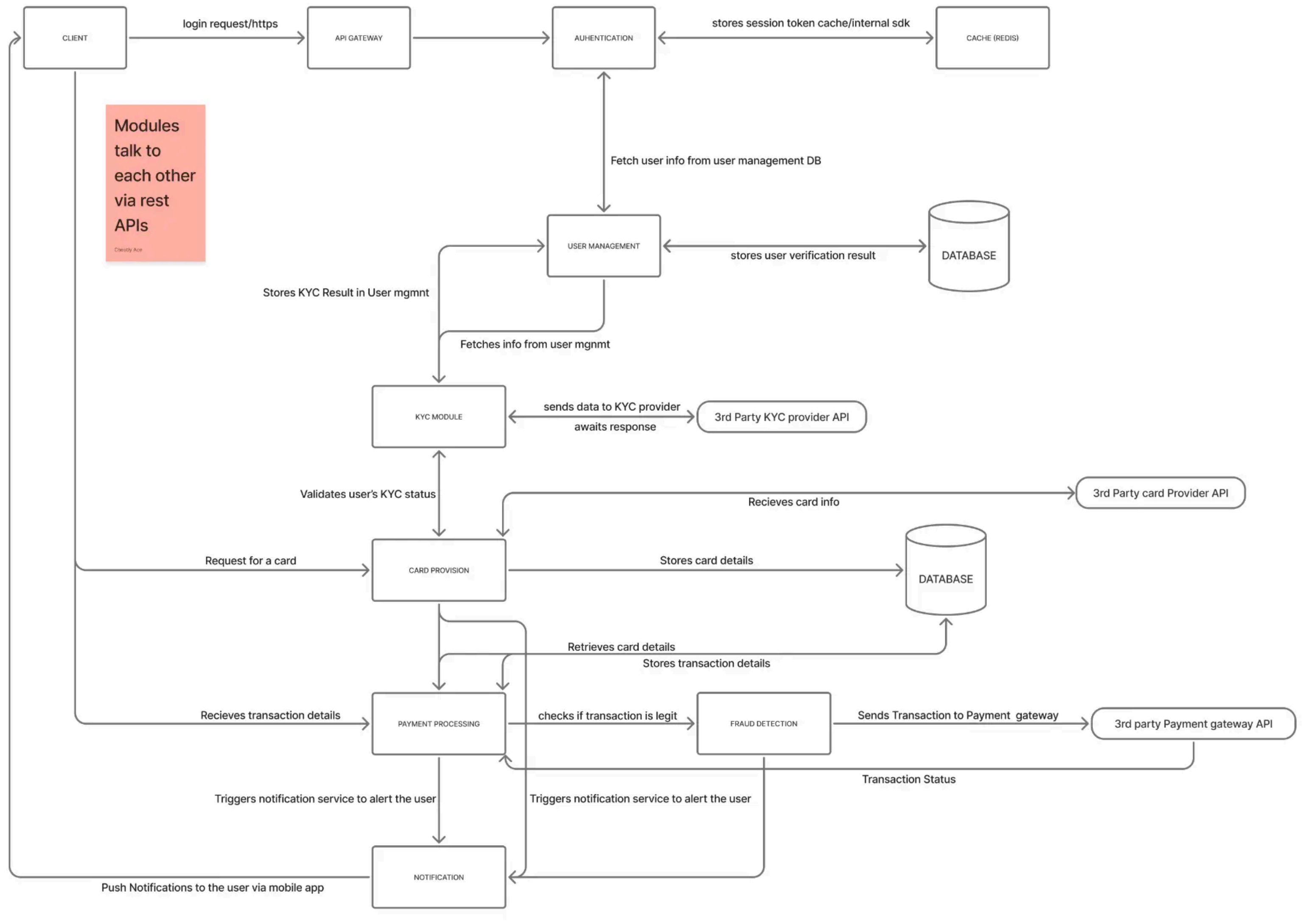
CORE FUNCTIONALITY

- **User Authentication** - Secure login and session management
- **Virtual Card Provisioning** - Third-party card issuance
- **KYC Verification** - Know Your Customer compliance
- **Fraud Detection** - Real-time transaction monitoring

ARCHITECTURE HIGHLIGHTS

- **Microservices Architecture** - Independent, scalable modules
- **REST API Communication** - Loose coupling between services
- **Push Notifications** - Real-time transaction alerts
- **Redis Caching** - High-performance data access

Holistic View: Virtual Card System Architecture


CLIENT
login request/https
API GATEWAY
AUHENTICATION
stores session token cache/internal sdk
CACHE (REDIS)
Modules talk to each other via rest APIs
Fetch user info from user management DB
USER MANAGEMENT
stores user verification result
DATABASE
Stores KYC Result in User mgmnt
Fetches info from user mgnmt
KYC MODULE
sends data to KYC provider
awaits response
3rd Party KYC provider API
Validates user's KYC status
Recieves card info
3rd Party card Provider API
Request for a card
CARD PROVISION
Stores card details
DATABASE
Retrieves card details
Stores transaction details
Recieves transaction details
PAYMENT PROCESSING
checks if transaction is legit
FRAUD DETECTION
Sends Transaction to Payment gateway
3rd party Payment gateway API
Transaction Status
Triggers notification service to alert the user
Triggers notification service to alert the user
NOTIFICATION
Push Notifications to the user via mobile app

Complete system showing client, API Gateway, core modules, databases, caching layer, and external integrations

Technology Stack: Building Blocks for Success

Frontend/Mobile

React Native

Cross-platform iOS/Android development with **shared codebase** and native performance

Flutter

High-performance alternative with **Material Design** and excellent UI consistency

Firebase

Push notifications, analytics, and **real-time messaging** for mobile apps

Backend/Infrastructure

Node.js + Express.js

Non-blocking I/O ideal for handling concurrent requests and microservices

Kong/AWS API Gateway

Request routing, **authentication**, rate limiting, and API management

Docker & Kubernetes

Containerization and orchestration for **scalable deployments**

Data/Caching

PostgreSQL

ACID compliance and data integrity essential for financial transactions

Redis

In-memory caching for **ultra-fast access** to sessions and user data

AWS/Google Cloud

Managed services, **global infrastructure**, and auto-scaling capabilities

Client Interaction and API Gateway Orchestration

Mobile Client

- **Primary Interface:** Native or cross-platform mobile app
- **Request Initiation:** Initiates login, card requests, and payment transactions
- **Notifications:** Receives push notifications for transaction alerts
- **Communication:** Connects to backend via API Gateway using HTTPS

API Gateway

- **Single Entry Point:** Routes all client requests to appropriate microservices
- **Request Validation:** Validates and transforms incoming requests
- **Rate Limiting:** Implements throttling to prevent abuse
- **API Management:** Handles versioning and load balancing

Secure User Access and Profile Management

Authentication Module

- Manages **user login** and session management for secure access
- Issues and validates **JWT tokens** for stateless authentication
- Integrates with **Redis cache** for fast token storage and retrieval
- Validates **user credentials** against the database
- Fetches **user information** from User Management module

User Management Module

- Stores and retrieves **user profile information** securely
- Maintains **user account details** in the database
- Provides **verification results** to other modules
- Manages **user permissions** and roles
- Fetches **user data** for KYC validation processes

Identity Verification and Virtual Card Issuance

KYC Module

IDENTITY VERIFICATION

- Validates user identity through **third-party KYC providers**
- Sends user data to **external KYC verification APIs**
- Awaits and processes **verification responses** in real-time
- Stores **KYC status** in User Management module
- Ensures **regulatory compliance** before card issuance

↓ Validates user eligibility ↓

Card Provision Module

VIRTUAL CARD ISSUANCE

- Handles **virtual card creation requests** from clients
- Validates **user KYC status** before provisioning
- Integrates with **third-party card provider APIs**
- Retrieves and stores **card information securely**
- Maintains **card lifecycle management**

↓ Issues virtual card ↓

Real-time Transactions and Proactive Fraud Prevention

Payment Processing Module

- **Transaction Processing** - Handles payment requests through third-party payment gateways
- **Card Data Retrieval** - Fetches card details securely from the database
- **Fraud Validation** - Sends transaction data to Fraud Detection Module for analysis
- **Transaction Recording** - Stores transaction details and status in database
- **Notification Trigger** - Initiates alerts to users via Notification Module

Fraud Detection Module

- **Pattern Analysis** - Analyzes transaction patterns for suspicious activity
- **Real-time Validation** - Checks transaction legitimacy in real-time
- **Risk Assessment** - Returns fraud assessment results immediately
- **Transaction Blocking** - May block or flag suspicious transactions
- **ML Integration** - Uses machine learning for anomaly detection

Instant Alerts and Performance Optimization

Notification Module

- **Push Notifications:** Delivers real-time alerts to mobile app users
- **Event Triggers:** Activated by Payment Processing and Fraud Detection modules
- **Transaction Alerts:** Notifies users of success, failure, and fraud alerts
- **Template Management:** Manages notification templates and delivery
- **Technology:** Firebase Cloud Messaging (FCM) for cross-platform support

Caching Layer (Redis)

- **Session Tokens:** Stores JWT tokens for fast authentication validation
- **User Data Caching:** Frequently accessed user profiles and card information
- **Load Reduction:** Reduces database load by 60-80% for read-heavy operations
- **TTL Management:** Automatic data expiration with time-to-live settings
- **In-Memory Performance:** Ultra-fast access for critical operations

Seamless Communication: REST APIs and Data Exchange

REST API COMMUNICATION

All modules communicate via **REST APIs** using HTTP/HTTPS protocols. This approach provides a standardized, language-agnostic method for inter-service communication.

KEY BENEFITS

- ✓ **Loose Coupling:** Modules operate independently
- ✓ **Standard Protocols:** HTTP/HTTPS universally supported
- ✓ **Technology Agnostic:** Any tech stack compatible
- ✓ **Easy Testing:** Independent endpoint testing

DATA EXCHANGE FORMAT

JSON (JavaScript Object Notation) is used for all data exchange. It provides lightweight, human-readable communication with native support in JavaScript/Node.js and easy parsing in mobile applications.

COMMUNICATION PATTERNS

Synchronous	Real-time REST calls for immediate responses (authentication, payments)
Asynchronous	Background processing for non-blocking operations (fraud detection)
Event-Driven	Notifications triggered by events from other modules

User Login Process: Request Phase

1 CLIENT → API GATEWAY
POST /api/auth/login with credentials

↓

2 API GATEWAY → AUTHENTICATION
Routes login request to Authentication Module

↓

3 AUTHENTICATION → REDIS CACHE
Checks for existing session token

↓

4 AUTHENTICATION → USER MANAGEMENT
GET /api/users/verify - Fetches user credentials

↓

5 USER MANAGEMENT → DATABASE
Queries user record to verify credentials

End-to-End Payment Transaction Flow

↓ Request Phase

1 **CLIENT**
Submits payment request with **cardId, amount, merchant details**

↓

2 **API GATEWAY**
Validates **JWT token** and routes to Payment Processing Module

↓

3 **PAYMENT PROCESSING**
Retrieves **card details** from database

↓

4 **FRAUD DETECTION**
Analyzes transaction for **suspicious activity**

↓ Processing Phase

5 **PAYMENT GATEWAY**
Sends **legitimate transaction** to third-party payment processor

↓

6 **DATABASE**
Stores **transaction details** and status

↓

7 **NOTIFICATION MODULE**
Triggers **push notification** to user's mobile app

↓

8 **API GATEWAY**
Returns **transaction result** to client

Robust, Scalable, and Secure Financial Architecture

Modularity

Each component can be developed, deployed, and scaled independently

Scalability

Horizontal scaling of individual services based on demand

Security

Multiple layers of protection for sensitive financial data

Performance

Caching and optimized data flows ensure fast response times

Maintainability

Clear separation of concerns and standard communication patterns

Reliability

99.9% uptime with distributed architecture and monitoring

Foundation for Success

This microservices architecture provides a **robust, scalable, and secure** foundation for a mobile-based virtual card financial system. The REST API communication pattern ensures flexibility and maintainability, while the chosen technology stack balances performance, developer productivity, and operational reliability—meeting the stringent security requirements of modern financial systems.