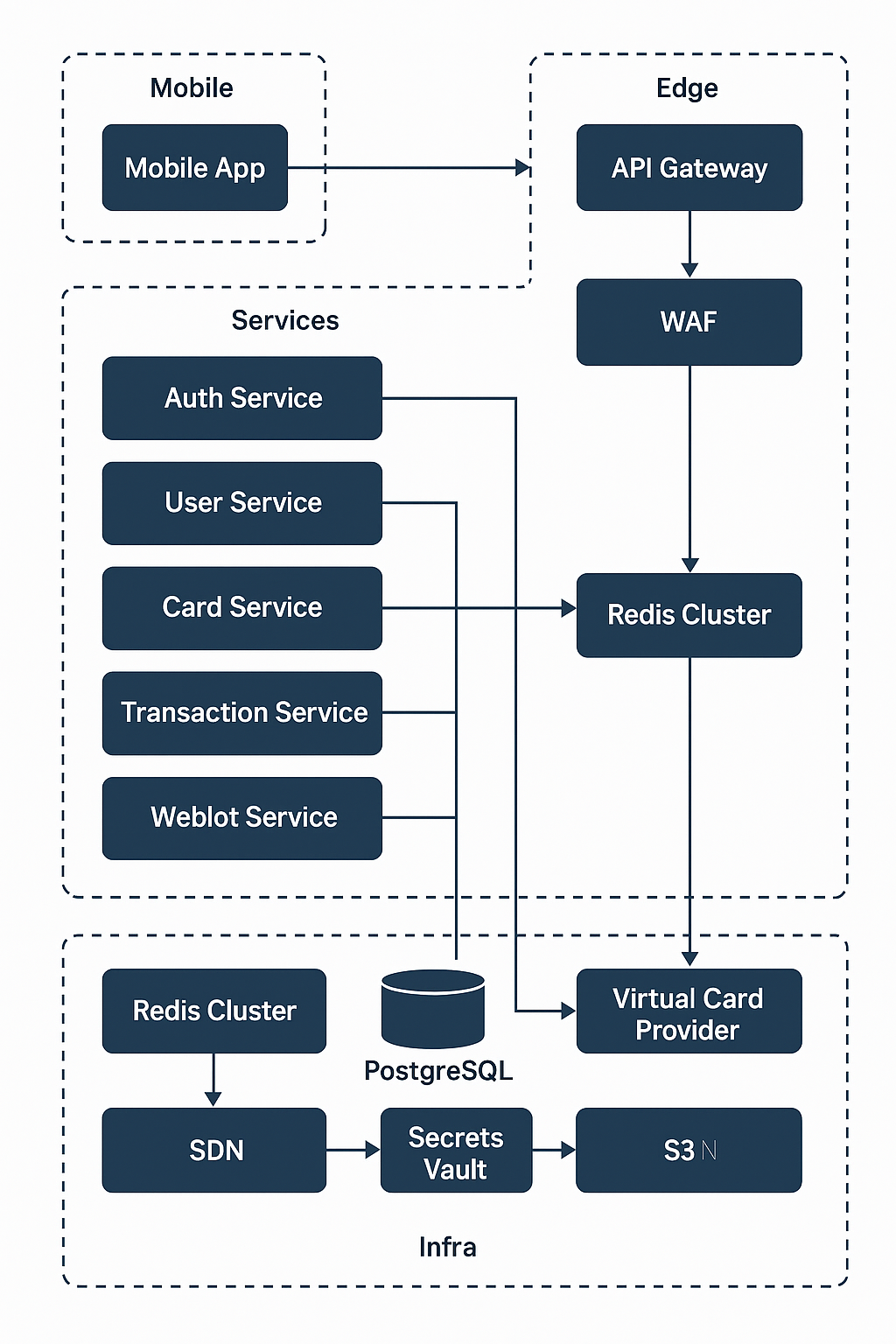
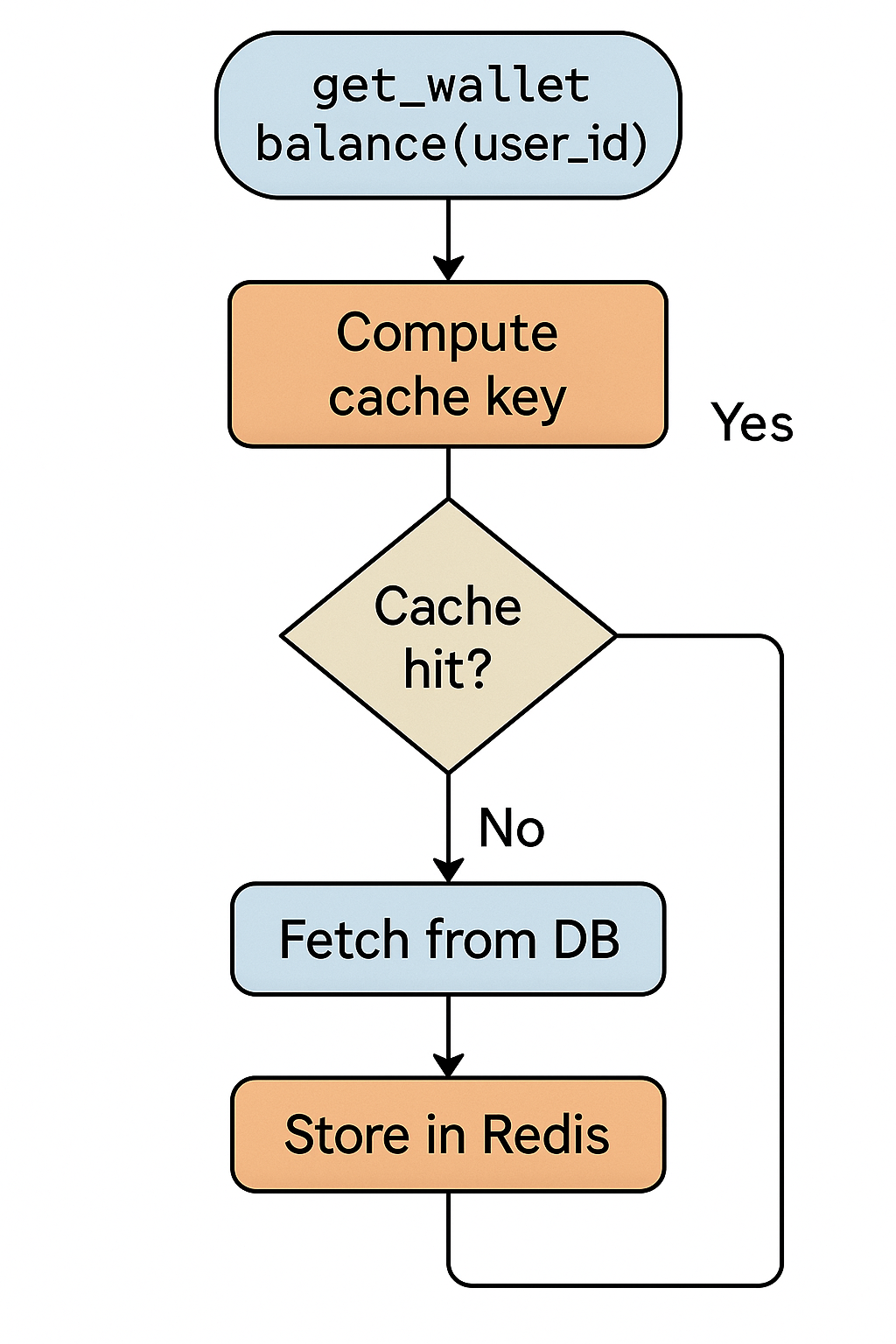
# Mobile Virtual Card Platform System Design Overview

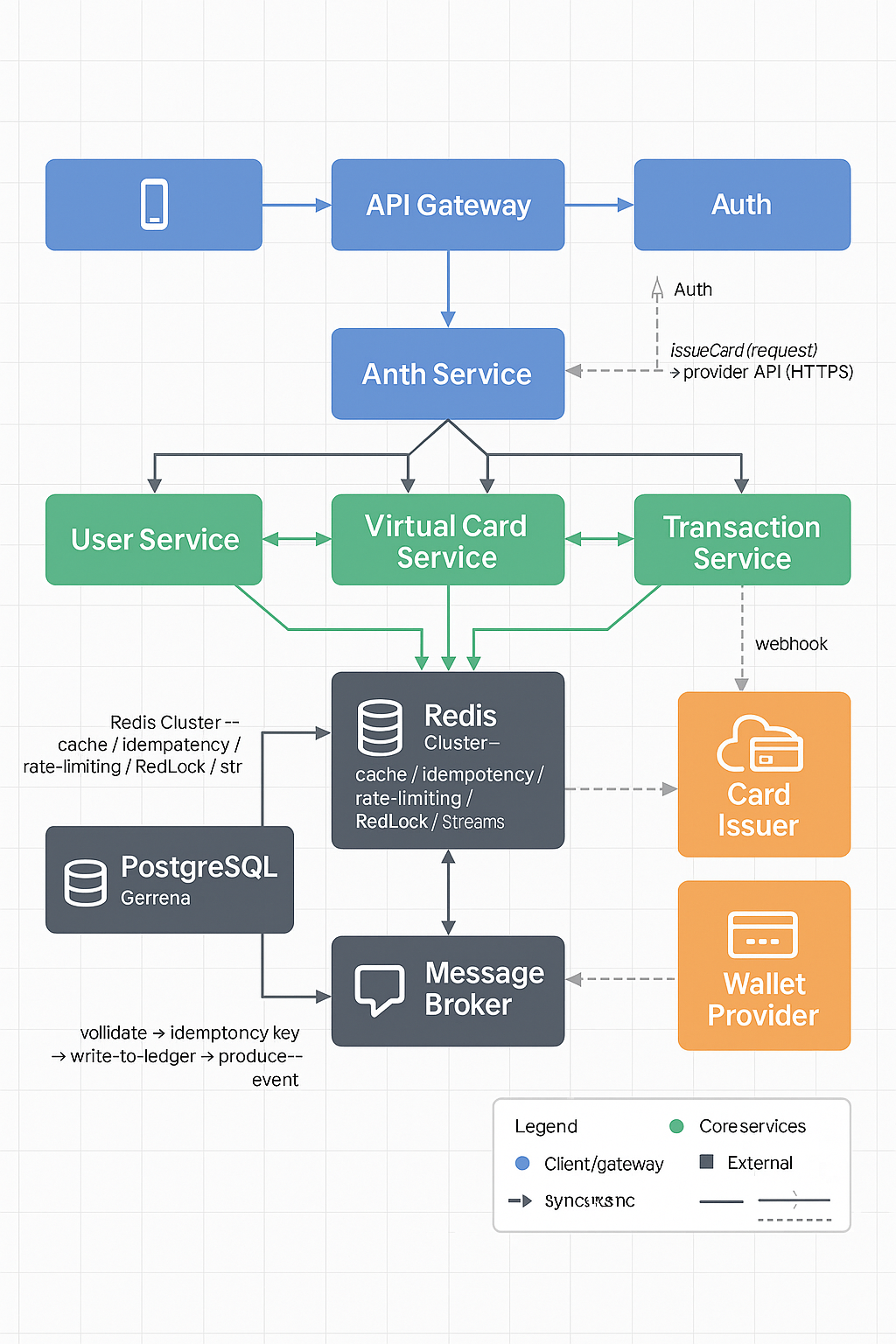
1. Architecture of the request response cycle



1. How Radis Function Function in data catching



1. Application System Architecture



## 4. **Best Programming Languages**

### **Node.js (Typescript)**

**Why:**

* Great for building fast, event-driven, real-time APIs.
* Excellent integration with Redis and third-party APIs.
* Typescript adds type safety and scalability for large systems.
* Large ecosystem (e.g., Express.js, NestJS, Fastify).

**Use case in your project:**

* RESTful API layer for mobile app.
* Middleware for handling requests and authentication.
* Micro services for user, wallet, and transaction modules.

**Pros:**  
Fast I/O operations (ideal for real-time payments).  
 Excellent developer community and NPM ecosystem.  
 Works well with Redis and PostgreSQL.  
 Easy to deploy in Docker/Kubernetes.

**Cons:**  
Single-threaded nature might require clustering under high load.

### **Go (Golang)**

**Why:**

* Excellent concurrency and performance.
* Strong standard library for HTTP, JSON, and network operations.
* Compiles to fast, small binaries — ideal for production.

**Use case in your project:**

* Backend services that require speed and concurrency (e.g., transaction service).
* High-performance APIs with Redis caching and message queues.

**Pros:**  
High performance and low latency.  
Strong concurrency (goroutines).

Great memory management.  
Ideal for microservice architectures.

**Cons:**  
 Slightly steeper learning curve than Node.js.

### **Python (Django + FastAPI)**

**Why:**

* Rapid development and readable code.
* FastAPI supports async APIs, suitable for handling requests efficiently.

**Use case:**

* Authentication and admin dashboards.
* Background job orchestration (Celery + Redis).

**Pros:**  
Easy to build secure APIs quickly.  
Strong integration with PostgreSQL and Redis.  
 Good for analytics or internal admin systems.

**Cons:**  
Slightly slower under very high load compared to Go or Node.js.

* 1. Programming language and data base is best for the system

|  |  |  |
| --- | --- | --- |
| **Layer** | **Recommended Database** | **Why** |
| **Main Transaction Database** | **PostgreSQL** | ACID-compliant, highly secure, ideal for financial data. |
| **Cache Layer** | **Redis** | In-memory speed for sessions, tokens, temporary transaction data. |
| **Analytics/Logs** | **MongoDB / Elastic Search** | For flexible logging, reporting, and analytics. |