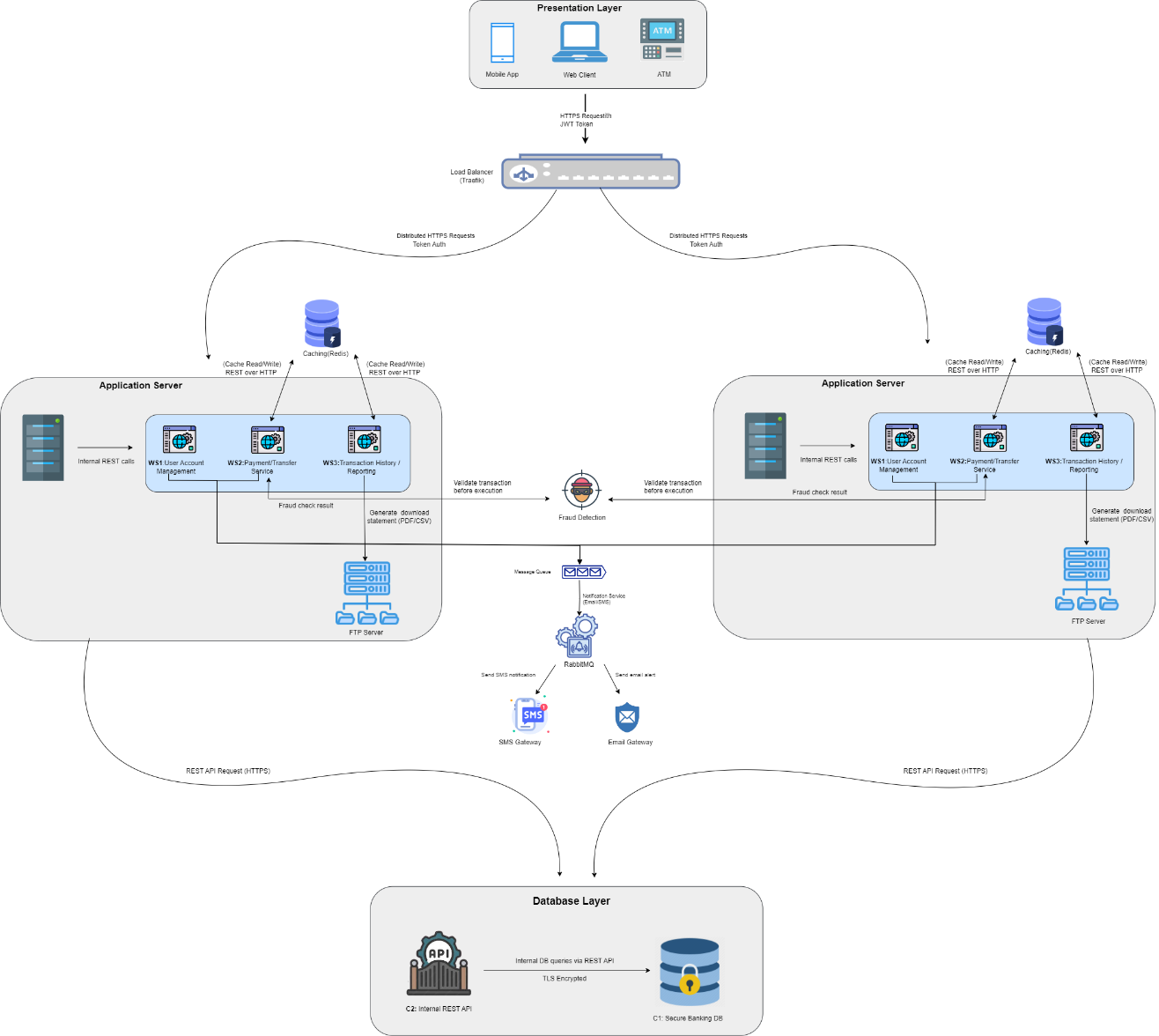
# Software Architecture (40353103-1)

## Assignment #2

Sdra Awameh 202210368

Deadline: 26/05/2025

### Task 1: Extend the Architecture

****

- Provide detailed descriptions of the new components and their interactions.

**Fraud Detection service:** Its an external API it checks every transaction before it completed the Payment service send transaction data to Fraud Detection: if the response is (safe) the transaction continues if (suspicious)the system blocks it or flags it.

**Notification service:** use to send alerts to users it uses email, SMS to notify users of transaction, get the event from the message queue then send the email or SMS gateways.

**Message Queue:** It connects service using events the Payment Service sends events to the queue. The Notification Service reads from the queue It keeps the system fast and decoupled

**Security layer:** communication is encrypted (using TLS) Users log in and get a JWT token. Sensitive data encrypted in the database

**Redundant**: if one service fails others continue to work this improve availability

### Task 2: Analyze Architectural Trade-offs

Evaluate the impact of the following architectural patterns on the updated system:  
- Microservices Architecture  
  
 Scalability: each service can scale alone no need to scale full system to save time and resourses  
 Availability: if one service fails other still work the system keeps running   
 Security: each service needs its own protection more tokens and access checks

- Layered Architecture

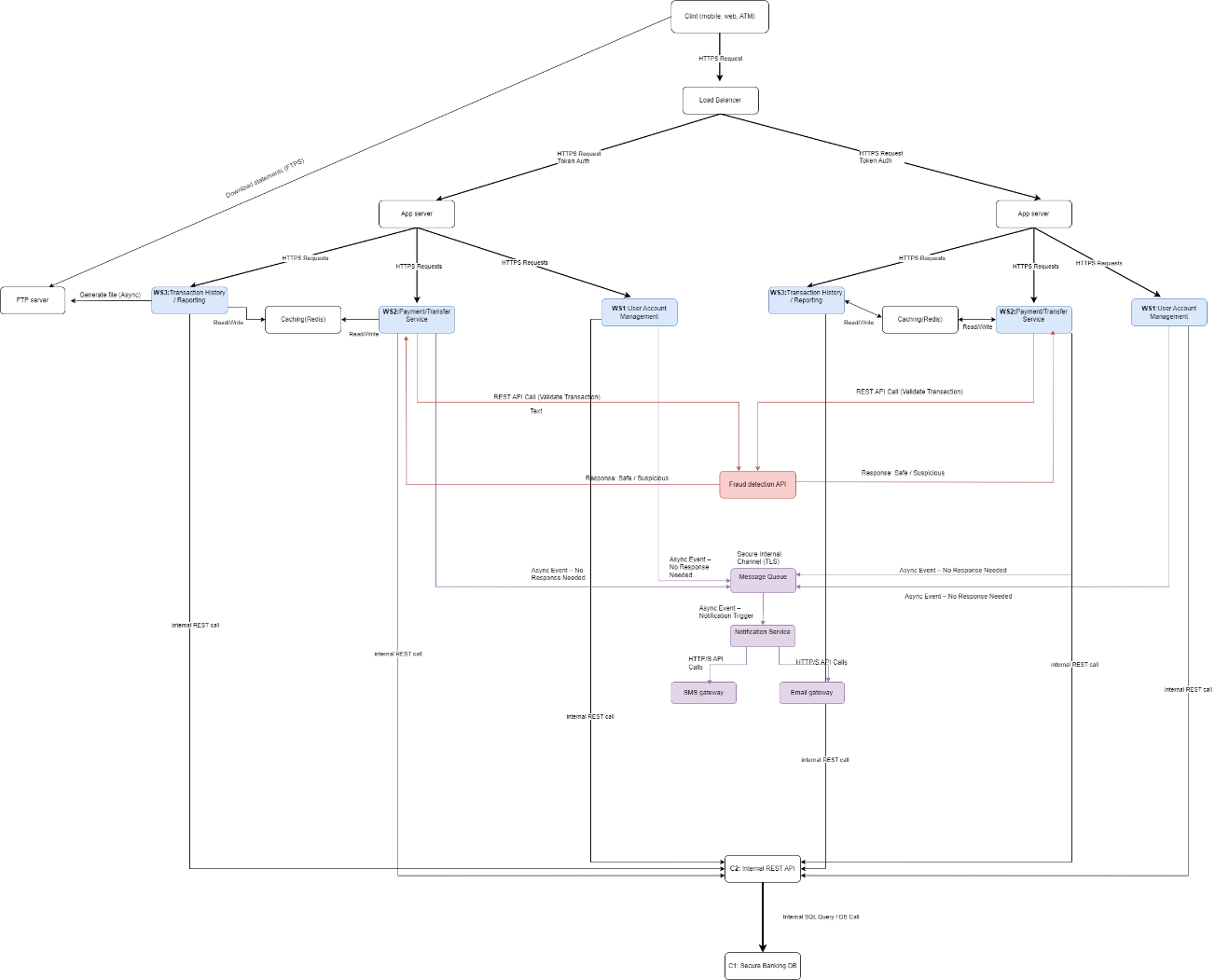
Scalability: must scale whole system together it use more resources   
 Availability: if one layer fails many things stop hard to keep system up   
 Security: one place to manage access easier to secure overall  **Provide a recommendation based on your analysis.**

I recommend microservice it work better with event based system

### Task 3: Quality Attribute Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Quality Att | Metric | Improvement | Trade-off: |
| Performance | Handel 1000 requests per second | Use caching for dashboard data | Use more caching data might not be real time put faster performance |
| Scalability | Must support 10x users | Add load balancer use microservice to scale parts | Hight cost |
| Reliability | Should be 99.9% uptime | Use redundant servers | More server cost more money |

### Task 4: Comprehensive Documentation

Document the updated architecture with the following:  


- Descriptions of each module, including their roles and interactions

|  |  |  |
| --- | --- | --- |
| Model | Role | Interaction |
| client | Allow users to access the system | Send requests to the system via HTTPS with JWT token |
| Load balancer | Distribute requests across multiple App servers | Forword client req to available App servers |
| App server | Main processor | Communicates with all WS,cache,DB |
| WS1 | Manages user profile and account | Work with App server and C1 |
| WS2 | Handel user transaction and payments | Send transaction data to fraud detection ,update DB,emits events to Queue |
| WS3 | Show past user transaction | Reads data from DB provide report |
| Fraud detection | External API to detect suspicious activity | Call by WS2 before processing payment |
| Message queue | Pass events between services | Receives events from WS2 then forward to NS |
| Notification service(NS) | Send alerts base on events | Call email /SMS gateway |
| Email /SMS gateway | External services for sending message | Called by NS to send alerts |

\*\*\*\*

End of Assignment 2 – Software Architecture.

Thanks for reading