Security and Access Control Strategy

Target Audience: Security Architects, Compliance Officers
Purpose: Defining the multi-layered security posture that protects user data, financial integrity, and operational continuity.

---

1. Authentication Model (Identity)

We adopt a **Stateless, Token-Based Authentication** model enabling secure interactions across Web, Mobile, and API clients.

- **Standard:** OAuth 2.0 / OpenID Connect compliant flows.
- **Mechanism:** JSON Web Tokens (JWT).
  - **Access Tokens:** Short-lived (15-60 mins). Signed with `EdDSA` or `RS256` (Asymmetric Keys). Used for API access.
  - **Refresh Tokens:** Long-lived (7 days). Securely stored (HttpOnly Cookies on Web, KeyChain on Mobile). Used to obtain new Access Tokens.
- **Zero Trust (Internal):** Services do not trust each other implicitly.
  - *Service-to-Service:* Internal requests must include a valid JWT or a mutual TLS (mTLS) certificate.
  - *Validation:* Every service validates the JWT signature individually against the public key exposed by the Auth Service.

---

2. Authorization Rules (Access)

We enforce **Role-Based Access Control (RBAC)** supplemented by **Resource-Based Scopes**.

2.1 The Principle of Least Privilege

Users are granted only the permissions essential for their specific function. A "Driver" cannot refund tickets; a "Ticket Agent" cannot edit Bus Schedules.

2.2 Scope Enforcement

Permissions are granular.

- `read:schedule` - Can view trips.
- `write:schedule` - Can create/edit trips.
- `write:refund` - Can authorize money return.

---

3. Role Definitions

Role	Scope / Description	Risk Level
Super Admin	**God Mode.** Can onboard new Companies, suspend accounts, view platform-wide revenue.	**Critical**
Company Admin	**Tenant Root.** Full control over *their specific company's* fleet, staff, and finances. Cannot see other companies.	High
Ops Supervisor	**Manager.** Can edit schedules, assign drivers, cancel trips. Cannot access top-level financial withdrawal.	Medium
Station Agent	**Sales.** Can searching trips, lock seats, issue tickets within their assigned station.	Low
Driver / Conductor	**Execution.** Can view manifest, scan QR codes, start/stop trips. Read-only access to sales data.	Low
Passenger	**Self-Service.** Can view *own* tickets and profile. No access to operational data.	Low

---

4. Multi-Tenancy & Data Isolation

The system is designed as a **Multi-Tenant SaaS**.

- **Logical Isolation:** Every database query includes a `WHERE company_id = X` clause automatically injected by the data access layer.
- **Token Binding:** The `company_id` is embedded in the JWT. A User from "Simba Coach" requesting data for "Kigali Bus" is rejected at the API Gateway level (403 Forbidden).

---

5. Sensitive Data Handling

5.1 Payment Information (PCI-DSS)

- **Card Data:** We **NEVER** touch or store raw credit card numbers.

- **Approach:** We use Tokenization. The Client sends card data directly to the Payment Provider (e.g., Stripe/PayPal) and receives a `payment_token`. We only store this token.

5.2 Passwords

- **Hashing:** Passwords are hashed using strong algorithms (Argon2id or bcrypt) with high work factors and per-user salts.
- **Policy:** Enforced complexity (Min 8 chars, mixed case, special chars).

5.3 Communication (Encryption in Transit)

- **TLS 1.2+:** All external and internal traffic uses HTTPS/TLS. Non-secure HTTP is rejected by the Gateway.

---

6. Audit and Traceability

To prevent fraud and ensure accountability, "who did what" is immutable.

- **The Audit Log:** A write-only, append-only log of critical actions.
- **Captured Data:** `Timestamp`, `Actor ID`, `Action` (e.g., `TICKET_REFUND`), `Target ID` (Ticket #123), `IP Address`, `User Agent`.
- **Retention:** Logs are retained for financial compliance (min 7 years for financial transactions).
- **Analysability:** Managers can query: *"Show me all Manual Refunds performed by Agent John in the last 24 hours."*

---

7. Operational Security

- **Rate Limiting:** Protects against Brute Force and DDoS. Configured at the Gateway (e.g., max 5 login attempts per minute).
- **Input Sanitization:** All API inputs are validated against strict schemas (Pydantic) to prevent SQL Injection and XSS attacks.