SAMFMS

Service Contracts

Team Firewall Five:

Team Member	Student number
Herman Engelbrecht	u22512374
Morné van Heerden	u21482153
Laird Glanville	u22541332
Stefan Jansen van Rensburg	u22550055
Nicolaas Johan Jansen van Rensburg	u22592732

Contents

1. Conventions

- Transport: REST over HTTP. JSON bodies (multipart/form-data for file uploads where stated).
- Auth: Bearer JWT in Authorization header. Role/permission enforcement per service.
- IDs: UUIDv4 strings (e.g., vehicle_id, driver_id, trip_id).
- Timestamps: ISO 8601 UTC (e.g., 2025-08-18T14:25:00Z).
- Numbers: Decimal for currency/quantities; integers for counts.
- Pagination (lists): page (default 1), page_size (default 25, max 200). Responses include items, total, page, page_size.
- Filtering (common): q (free text), status, start_time, end_time, sort (field[:asc|:desc]).
- Response envelope (success): { data: <object|array>, meta?: { page, page_size, total } }.
- Response envelope (error): { error: { code: string, message: string, details?: any } }.
- Error codes: 400 validation, 401 unauthenticated, 403 forbidden, 404 not found, 409 conflict, 422 semantic validation, 500 server error, 502/503/504 upstream/unavailable/timeout.
- Resilience: Circuit breakers, retries with backoff, request de-duplication (content TTL 60s; correlation TTL 300s).
- Messaging (AMQP): Exchange 'service_requests' → '<service>.requests'; responses on 'service_responses'. Messages JSON with correlation_id.

2. Core Gateway

Routes /management/*, /maintenance/*, /gps/*, /trips/* to respective services. Also exposes health and service registry.

Health

- GET /health/
  - Input: no input
  - Output: { data: { status: 'ok' } }.
- GET /health/ready
  - Input: none
  - Output: readiness booleans per dependency.
- GET /health/live
  - Input: none
  - Output: { data: { alive: true } }.
- GET /health/detailed
  - Input: none
  - Output: { data: { db, mq, discovery, circuits } }.
- GET /health/metrics
  - Input: optional ?limit
  - Output: basic counters/gauges.
- GET /services
  - Input: none
  - Output: { data: [{ name, base_path, version }] }.

Routing

- ANY /{service}/{path}

Auth Proxy

- All /auth/* paths are forwarded to Security.

3. Management Service

Base: /management/*

- Vehicle: { vehicle_id, plate, vin?, make, model, year?, status:'active'|'inactive'|'maintenance', mileage? }
- Assignment: { assignment_id, driver_id, vehicle_id, start_time, end_time?, start_mileage?, end_mileage? }
- Driver: { driver_id, name, email, phone?, license_no?, status }
- MileageRecord: { mileage_record_id, vehicle_id, odometer, reading_time }
- FuelRecord: { fuel_record_id, vehicle_id, liters, cost, currency, odometer?, time }
- Notification: { notification_id, type:'system'|'assignment'|'maintenance', message, recipients?, status:'unread'|'read'|'archived', created_at }

Vehicles (/vehicles)

- GET /vehicles
  - Input: paging, q, status, driver_id
  - Output: { items:[Vehicle], total,... }.
- POST /vehicles
  - Input: { plate, vin?, make, model, year?, status? }
  - Output: Vehicle.
- GET /vehicles/{vehicle_id}
  - Input: path id
  - Output: Vehicle.
- PUT /vehicles/{vehicle_id}
  - Input: partial Vehicle fields
  - Output: Vehicle.
- DELETE /vehicles/{vehicle_id}
  - Input: path id
  - Output: { data: { deleted: true } }.
- GET /vehicles/{vehicle_id}/assignments
  - Input: paging/date filters
  - Output: [Assignment].

- GET /vehicles/{vehicle_id}/usage
    - Input: start_time?, end_time?
    - Output: { mileage, trips, fuel }.

Drivers (/drivers)

- GET /drivers
    - Input: paging, q, status
    - Output: [Driver].
- POST /drivers
    - Input: { name, email, phone?, license_no?, status? }
    - Output: Driver.
- GET /drivers/{driver_id}
    - Input: none
    - Output: [Driver]
- PUT /drivers/{driver_id}
    - Input: partial Driver
    - Output: Driver.
- DELETE /drivers/{driver_id}
    - Input: None
    - Output: { data: { deactivated: true } }.
- POST /drivers/{driver_id}/assign-vehicle?vehicle_id=
    - Input: None
    - Output: Assignment.
- POST /drivers/{driver_id}/unassign-vehicle
    - Input: None
    - Output: { data: { unassigned: true } }.

Assignments (/assignments)

- POST /assignments
    - Input: { driver_id, vehicle_id, start_time?, start_mileage? }
    - Output: Assignment.
- DELETE /assignments/{assignment_id}?end_mileage=
    - Input: None
    - Output: { data: { closed: true } }.
- GET /assignments/active
    - Input: paging

    - Output: [Assignment].

Mileage (/mileage)

- POST /mileage/update
    - Input: { vehicle_id, odometer, reading_time? }
    - Output: { data: { updated: true }, vehicle_mileage }
- GET /mileage/records/{mileage_record_id}
    - Input: None
    - Output: MileageRecord

Fuel (/fuel)

- POST /fuel/records
    - Input: { vehicle_id, liters, cost, currency, odometer?, time? }
    - Output: FuelRecord.
- PUT /fuel/records/{fuel_record_id}
    - Input: partial FuelRecord
    - Output: FuelRecord.
- GET /fuel/records/{fuel_record_id}
    - Input: None
    - Output: FuelRecord.

Notifications (/notifications)

- POST /notifications
    - Input: { type, message, recipients?:[user_id]|'broadcast' }
    - Output: Notification.
- PUT /notifications/{notification_id}/read
    - Input: None
    - Output: { data: { read: true } }.
- GET /notifications/recipient/{recipient_id}
    - Input: status?
    - Output: [Notification].

Analytics (/analytics)

- GET /analytics/dashboard
    - Input: range?
    - Output: { kpis:{ utilization, active_drivers, mileage }, highlights:[...] }
- GET /analytics/driver-performance/{driver_id}
    - Input: range?
    - Output: { scores, incidents, mileage }.

4. Maintenance Service

Base: /maintenance/*

- MaintenanceRecord: { record_id, vehicle_id, category, service_date, cost?, currency?, next_due_date?, notes? }.
- License: { record_id, entity_type, entity_id, number, expiry_date, status:'active'|'expired'|'revoked' }.

Records (/records)

- POST /records
    - Input: { vehicle_id, category, description?, service_date, cost?, currency?, next_due_date? }
    - Output: MaintenanceRecord.
- GET /records
    - Input: paging, q, vehicle_id?, date range
    - Output: [MaintenanceRecord].
- PUT /records/{record_id}
    - Input: partial MaintenanceRecord
    - Output: MaintenanceRecord.

Licenses (/licenses)

- POST /licenses
    - Input: { entity_type:'vehicle'|'driver'|'company', entity_id, number, issued_date?, expiry_date, status? }
    - Output: License.

- GET /licenses/status/expiring?days=
    - Input: None
    - Output: [License].
- POST /licenses/{record_id}/renew
    - Input: { expiry_date, notes? }
    - Output: License.
    -

Notifications & Analytics

- GET /notifications/pending
    - Input: None
    - Output: [Notification].
- POST /notifications/process
    - Input: None
    - Output: { data: { processed: count } }.
- GET /analytics/summary/vehicle/{vehicle_id}
    - Input: range?
    - Output: { last_service, upcoming, costs }.

5. GPS Service

Base: /gps/*

- Location: { vehicle_id, lat, lon, time, speed?, heading? }.
- Place: { place_id, name, lat, lon, radius?, category?, tags?[] }.
- Geofence: { geofence_id, type, geometry, name, created_at }.

Locations (/locations)

- POST /locations/update
    - Input: { vehicle_id, lat, lon, time?, speed?, heading? }
    - Output: Location (current) and appended to history.
- GET /locations/{vehicle_id}
    - Input: path id
    - Output: Location.

- GET /locations?vehicle_ids=a,b,c
    - Input: query
    - Output: [Location].
- GET /locations/{vehicle_id}/history?start_time&end_time&limit
    - Input: None
    - Output: [Location].

Places (/places)

- POST /places
    - Input: { name, lat, lon, radius?, category?, tags?[] }
    - Output: Place.
- GET /places
    - Input: paging, q, category?
    - Output: [Place].
- PUT /places/{place_id}
    - Input: partial Place
    - Output: Place.

Geofences (/geofences)

- POST /geofences
    - Input: { type:'circle'|'polygon'|'rectangle', geometry, name, description? }
    - Output: Geofence.
- GET /geofences
    - Input: paging, q
    - Output: [Geofence].

6. Trip Planning Service

Base: /trips/*

- Trip: { trip_id, status:'planned'|'in_progress'|'completed'|'cancelled', stops:[Stop], driver_id?, vehicle_id?, route?, analytics? }.
- Stop: { index, name?, lat, lon, eta?, etd?, window_start?, window_end?, arrived_at?, departed_at? }.
- ProgressSnapshot: { time, status, lat?, lon? }.

Trips (/trips)

- POST /trips
  - Input: { stops:[{ lat, lon, name?, window_start?, window_end? }], vehicle_id?, driver_id?, notes?, constraints? }
  - Output: Trip.
- GET /trips
  - Input: paging, q, status, date range
  - Output: [Trip].
- GET /trips/{trip_id}
  - Input: None
  - Output: Trip.
- PUT /trips/{trip_id}
  - Input: partial Trip
  - Output: Trip.
- POST /trips/{trip_id}/start
  - Input: { time? };
  - Output: Trip { status:'in_progress' }.
- POST /trips/{trip_id}/complete
  - Input: { time? }
  - Output: Trip { status:'completed', analytics } .
- POST /trips/{trip_id}/optimize-route
  - Input: { objective?'shortest'|'fastest', constraints? }
  - Output: Trip { optimized:true, route } .
- PUT /trips/{trip_id}/progress
  - Input: { time, location?, status? }
  - Output: ProgressSnapshot.

Drivers & Availability (/drivers)

- GET /drivers/availability?start_time&end_time&driver_ids[]
    - Input: None
    - Output: [{ driver_id, blocks:[{ start_time, end_time, status }] }].
- POST /drivers/{trip_id}/assign
    - Input: { driver_id, vehicle_id? }
    - Output: Trip with assignment.

Notifications & Analytics

- POST /notifications
    - Input: { type, message, recipients?:[user_id] }
    - Output: Notification.
- GET /analytics/trips/summary
    - Input: filters, range
    - Output: aggregate metrics (count, distance, duration, on-time%).

7. Security Service

Base: /auth/* and /admin/*

- User: { user_id, name, email, role, created_at }.
- Role: { name, permissions:[string] }.

Authentication (/auth)

- POST /auth/signup
    - Input: { name, email, password }
    - Output: { token, user }.
- POST /auth/login
    - Input: { email, password }
    - Output: { token, user }.
- POST /auth/logout
    - Input: none (header token)
    - Output: { data:{ logged_out:true } }.
- GET /auth/me
    - Input: none
    - Output: user profile.

- POST /auth/verify-token
    - Input: { token? } or header
    - Output: { valid, user }.
- GET /auth/roles
    - Input: None
    - Output: [Role].

Users & Admin (/users, /admin)

- GET /users
    - Input: paging, q, role?
    - Output: [User].
- PUT /users/{user_id}/permissions
    - Input: { role, permissions?[] }
    - Output: User.
- POST /admin/invite-user
    - Input: { email, role, expires_at? }
    - Output: { invitation_id, otp_masked }.
- POST /admin/verify-otp
    - Input: { invitation_id, otp }
    - Output: { verified:true }.
- POST /admin/activate-user/{user_id}
    - Input: None
    - Output: { data:{ activated:true } }.

8. Messaging Contracts

- Request message: { correlation_id, method, endpoint, user_context:{ user_id, roles[], tenant_id? }, data? }.
- Response message (success): { correlation_id, status:'success', data, timestamp }.
- Response message (error): { correlation_id, status:'error', error:{ code, message }, timestamp }.
- Timeouts (seconds): management 30–60; maintenance 30–60; gps 20–30; trips 45–60; security 30.