GachaAndGames System Documentation

Riccardo Fantasia, Leonardo Pantani, Christian Sabella

November 19, 2024

System Architecture Documentation

System Architecture

Architecture Overview

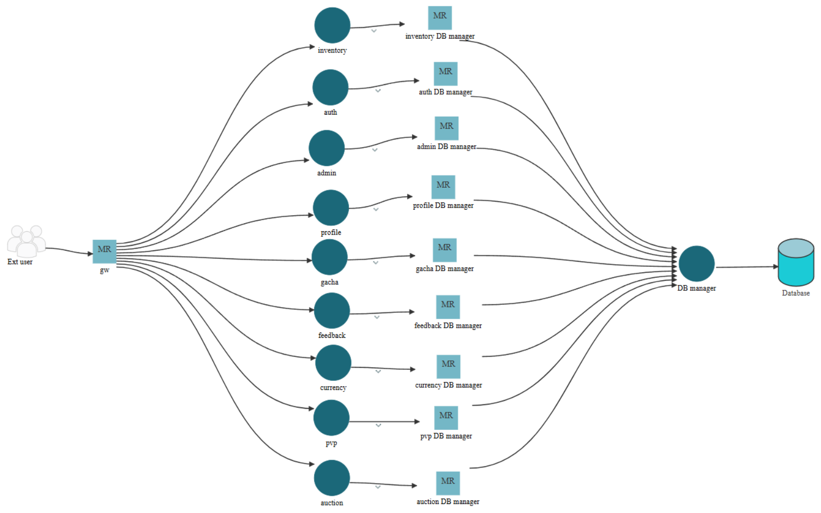

Figure 1: Microservice architecture diagram

Core Services

Auth Service

Service path: `services/auth`

- Manages user authentication through cookie-based sessions.
- Implements secure password hashing and verification.
- Communicates with db_manager for user data persistence.
- Uses circuit breakers for fault tolerance.

Profile Service

Service path: `services/profile`

- Manages user profiles with UUID-based identification.

- Handles profile updates (username, email) and deletions.
- Stores profile information like join date.
- Validates usernames to contain only letters, numbers, and underscores.
- Requires usernames to be at least 5 characters long.

Admin Service

Service path: `services/admin`

- Administrative dashboard for system management.
- Controls gacha pools and item configuration.
- Manages user permissions and roles.
- Views system logs and feedback.
- Can update/delete auctions and profiles.

Auctions Service

Service path: `services/auctions`

- Manages item auctions between users.
- Tracks auction status (active/closed).
- Handles bidding with minimum price validation.
- Records bid history and winners.
- Uses MySQL for auction persistence.

Currency Service

Service path: `services/currency`

- Handles in-game currency transactions.
- Manages currency bundles for purchases.
- Supports multiple currency types via codes (e.g., ”EUR”).
- Validates transaction amounts.
- Tracks user balances.

Inventory Service

Service path: `services/inventory`

- Manages user item collections.
- Tracks item ownership and transfers.
- Records item stats and attributes.
- Stores item acquisition dates.
- Maintains ownership history count.

Feedback Service

Service path: `services/feedback`

- Collects user feedback submissions.
- Validates feedback content.
- Routes feedback to administrators.
- Simple JSON payload with feedback string.
- Session-based user identification.

Gacha Service

Service path: `services/gacha`

- Implements gacha game mechanics.
- Manages item pools with configurable probabilities.
- Four rarity levels: common (50%), rare (30%), epic (15%), legendary (5%).
- Items have attributes rated A-E.
- Validates pool configurations.

PvP Service

Service path: `services/pvp`

- Manages player versus player battles.
- Handles matchmaking.
- Tracks battle results.
- Implemented as Flask service with MySQL backend.
- Uses session authentication.

DB Manager Service

Service path: `db_manager`

- Central database management service.
- Handles data persistence across services.
- Manages MySQL connections and queries.
- Implements retry logic for DB connections.
- Uses environment variables for configuration.

Infrastructure Components

API Gateway

Configuration path: `api_gateway/api_gateway.conf`

- Nginx-based reverse proxy.
- Implements least connections load balancing.
- Configures fail timeout and max fails.
- Routes requests to appropriate services.

Database

- MySQL Database for centralized storage.
- Used by all services through db_manager.
- Configured via environment variables.
- Stores user data, items, auctions, etc.