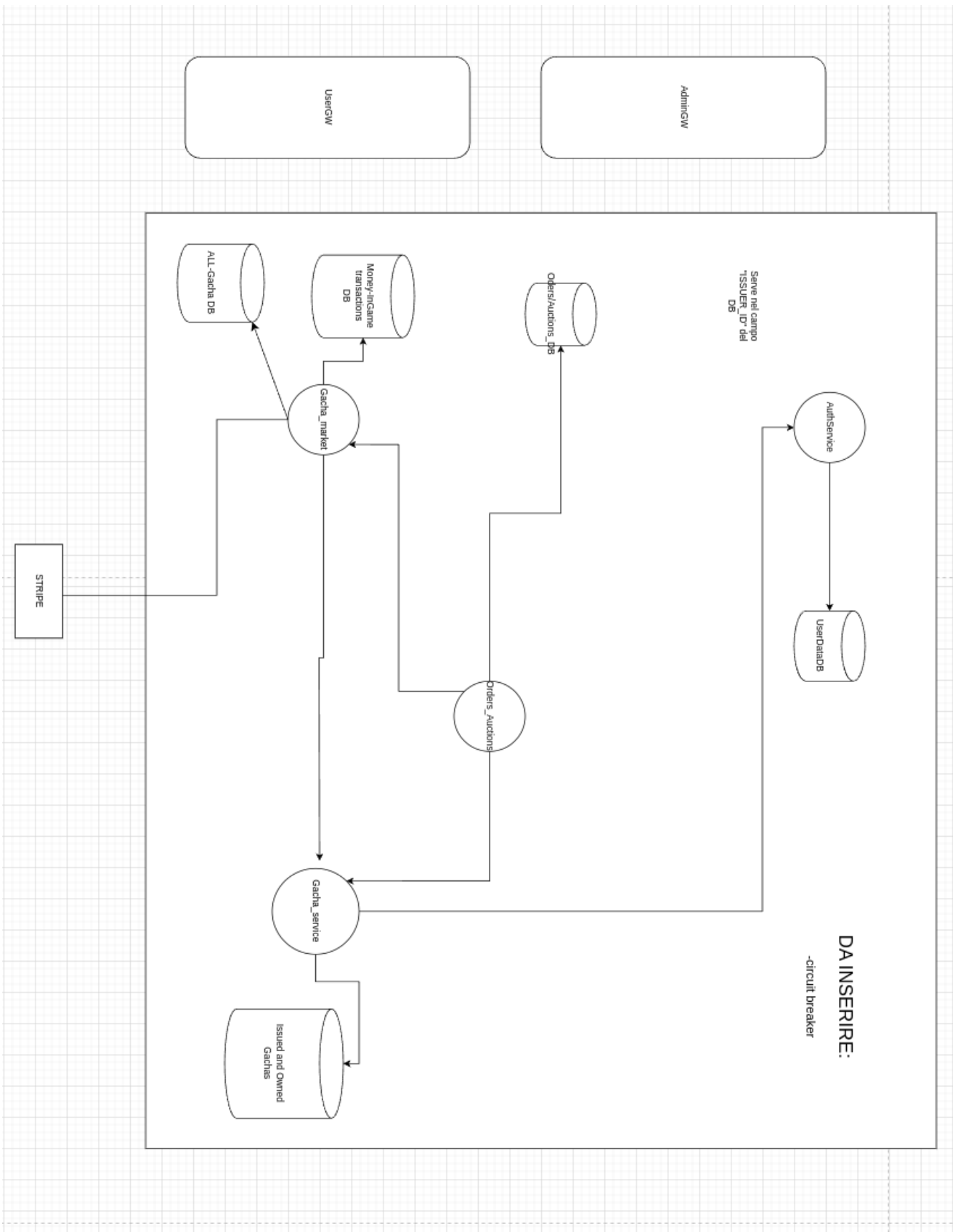


Architettura dei microservizi:



Descrizione dei microservizi:

Authentication Service

L'**Authentication Service** gestisce le operazioni di autenticazione e autorizzazione degli utenti. Le sue responsabilità includono la registrazione di nuovi utenti, il login e il controllo delle credenziali di autenticazione. Inoltre, il servizio fornisce operazioni per la gestione del wallet di ciascun utente, come l'aggiunta o la sottrazione di valuta virtuale.

Gacha Service

Il **Gacha Service** gestisce la collezione di "gachas" (oggetti virtuali) di ciascun giocatore. Questo servizio si occupa di memorizzare, aggiornare e recuperare i gachas che appartengono a un utente. Inoltre, fornisce la possibilità di trasferire il possesso dei gachas da un utente all'altro, funzionalità utile, ad esempio, per l'assegnazione degli oggetti a seguito delle aste.

Auction Service

L'**Auction Service** gestisce tutte le operazioni legate alle aste dei gachas. Un utente può mettere un gacha all'asta e altri utenti possono fare delle puntate su tale asta. Il servizio gestisce tutte le dinamiche dell'asta, tra cui l'inserimento di nuove offerte, il rimborso delle puntate precedenti, e l'assegnazione finale del gacha al miglior offerente.

Market Service

Il **Market Service** gestisce le operazioni di acquisto e transazione della valuta virtuale all'interno del gioco. Fornisce agli utenti la possibilità di acquistare la

valuta virtuale, che può essere poi utilizzata, ad esempio, per partecipare alle aste o acquistare nuovi oggetti.

User API Gateway

L'**API Gateway** agisce come punto di ingresso centralizzato per tutte le richieste provenienti dai giocatori o dagli amministratori verso i vari microservizi. L'API Gateway indirizza le richieste ai rispettivi microservizi interni, fungendo da router e gestore delle richieste.

Operazioni effettuate da un utente quando vuole puntare su un asta:

Un giocatore (user ID = 21) desidera puntare su un'asta specifica (auction ID = 13) con un importo di 300 unità di valuta.

Il giocatore invia una richiesta POST all'API Gateway, all'endpoint `/auction_service/auctions/13/bid`, con `user_id = 21` e `importo = 300`.

L'API Gateway inoltra la richiesta al servizio `auction_service`, all'endpoint `/auction_service/auctions/13/bid`.

L'`auction_service` invia una richiesta GET all'`authentication_service` per ottenere le informazioni dell'utente, verificando che il giocatore (`user_id = 21`) abbia fondi sufficienti nel wallet (utilizzando `/authentication/players/21`).

L'`authentication_service` ritorna i dettagli del wallet del giocatore (200 OK).

L'`auction_service` verifica se l'importo offerto è superiore al `current_bid` e se l'utente ha fondi sufficienti. Se la puntata è valida, l'`auction_service` aggiorna l'asta (`current_bid` e `current_user_winner_id`) e chiama l'`authentication_service` per scalare l'importo dal wallet del giocatore (PATCH `/authentication/players/21/currency/update`).

L'`auction_service` ritorna una risposta (200 OK) all'API Gateway, indicando che la puntata è stata inserita con successo.

L'API Gateway restituisce la risposta al giocatore.

Flusso di Aggiunta di un Gacha alla Collezione di un Giocatore:

Un giocatore (user ID = 12) desidera aggiungere un nuovo gacha (gacha ID = 45) alla propria collezione.

Il giocatore invia una richiesta POST all'API Gateway, all'endpoint `/players/12/gachas`, con il corpo della richiesta contenente le informazioni del gacha (gacha ID = 45).

L'API Gateway inoltra la richiesta al Gacha Service, all'endpoint `/gacha_service/players/12/gachas`.

Il Gacha Service crea un nuovo oggetto `GachaCollection` utilizzando i dati forniti (user ID = 12, gacha ID = 45) e salva l'oggetto nel database `GachaCollection`.

Il Gacha Service ritorna una risposta di successo (201 Created) all'API Gateway.

L'API Gateway invia una risposta al giocatore, indicando che il gacha è stato aggiunto con successo alla collezione.

Flusso di Visualizzazione della Collezione Gacha del Giocatore

Un giocatore (user ID = 12) vuole visualizzare tutti i gacha della sua collezione.

Il giocatore invia una richiesta GET all'API Gateway all'endpoint `/players/12/gachas`.

L'API Gateway inoltra la richiesta al Gacha Service all'endpoint `/gacha_service/players/12/gachas`.

Il Gacha Service interroga il database `GachaCollection` per tutti i record con `user_id = 12`.

Il Gacha Service ritorna la lista dei gacha trovati all'API Gateway (200 OK).

L'API Gateway ritorna la risposta al giocatore, contenente l'elenco di tutti i gacha nella sua collezione.

Flusso di cancellazione di un utente

Un amministratore desidera cancellare un account giocatore (user ID = 12), cancellando anche tutta la collezione di gachas associati.

L'admin invia una richiesta DELETE all'API Gateway all'endpoint
/authentication/account?accountId=12.

L'API Gateway inoltra la richiesta all'authentication_service all'endpoint
/authentication/account?accountId=12.

L'authentication_service invia una richiesta DELETE al gacha_service per
cancellare la collezione di gachas associata all'utente (DELETE
/gacha_service/players/12/gachas).

Il gacha_service cancella tutti i record di GachaCollection per user_id = 12 e
restituisce una risposta (200 OK).

Dopo la conferma della cancellazione dei gachas, l'authentication_service
procede a cancellare l'utente dal database degli utenti (Users DB).

L'authentication_service ritorna una risposta (200 OK) all'API Gateway,
indicando che l'account è stato cancellato con successo.

L'API Gateway restituisce la risposta all'admin.