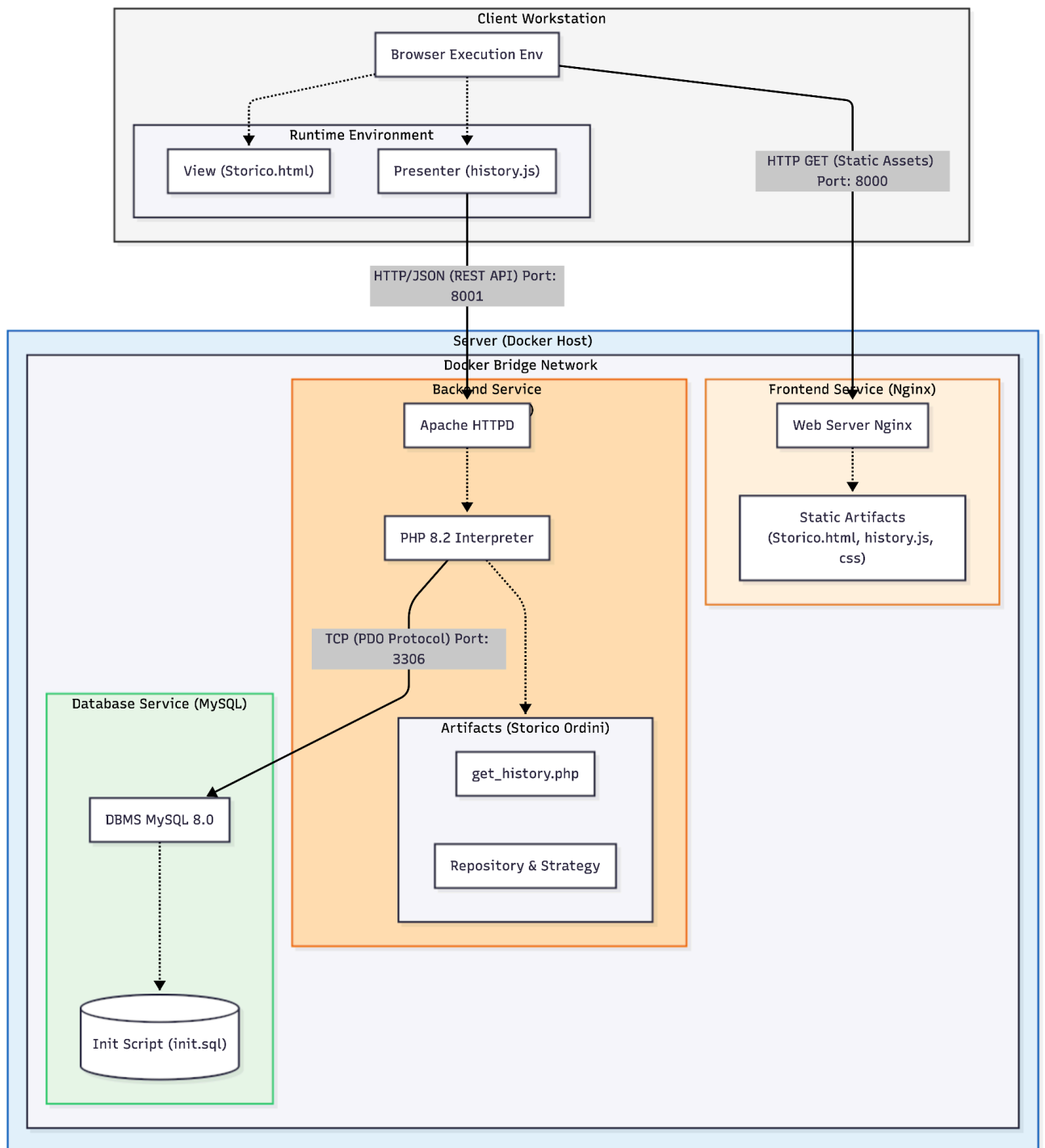


SVOLTO DA COLUCCI PASQUALE, MATR: 358141

## Diagramma di Deployment: Modulo Storico Ordini



## Descrizione Diagramma di Deployment: Modulo Storico Ordini

Il diagramma di deployment illustra l'architettura fisica del sistema, basata su containerizzazione **Docker** per garantire isolamento, portabilità e facilità di deploy.

## 1. Nodo Client (User Tier)

- Il punto di accesso è il browser dell'utente, che agisce da ambiente di esecuzione per il codice Frontend.
- **Artifacts:** Il browser scarica ed esegue i file statici (`Storico.html`, `global.css`) e lo script di logica client (`history.js`), che costituiscono il Presenter e la View del pattern MVP.

## 2. Nodo Server (Docker Host)

- L'infrastruttura di backend risiede su un host che esegue Docker Engine. L'applicazione è orchestrata tramite `docker-compose`, che definisce due servizi distinti connessi tramite una rete bridge interna.

## 3. Container Web (Application Tier)

- **Immagine:** Basata su `php:apache`.
- **Ruolo:** Gestisce le richieste HTTP in ingresso sulla porta pubblica (mappata nel `docker-compose.yml`, es. 8001).
- **Artifacts Backend:** Contiene la logica di business PHP (Controller, Factory, Strategy, Repository). Il server Apache inoltra le richieste `.php` all'interprete, che esegue l'algoritmo di recupero ordini.

## 4. Container Database (Data Tier)

- **Immagine:** `mysql:8.0`.
- **Ruolo:** Persistenza dei dati. Non è esposto direttamente su internet per sicurezza, ma è raggiungibile solo dal Container Web tramite la rete interna Docker sulla porta 3306.
- **Artifacts:** Contiene lo schema relazionale definito in `init.sql` e il volume persistente dei dati.

## 5. Protocolli di Comunicazione

- **Client → Web Server:** HTTPS/HTTP (JSON Payload).
- **Web Server → Database:** Protocollo nativo MySQL su TCP/IP (tramite driver PDO PHP).