

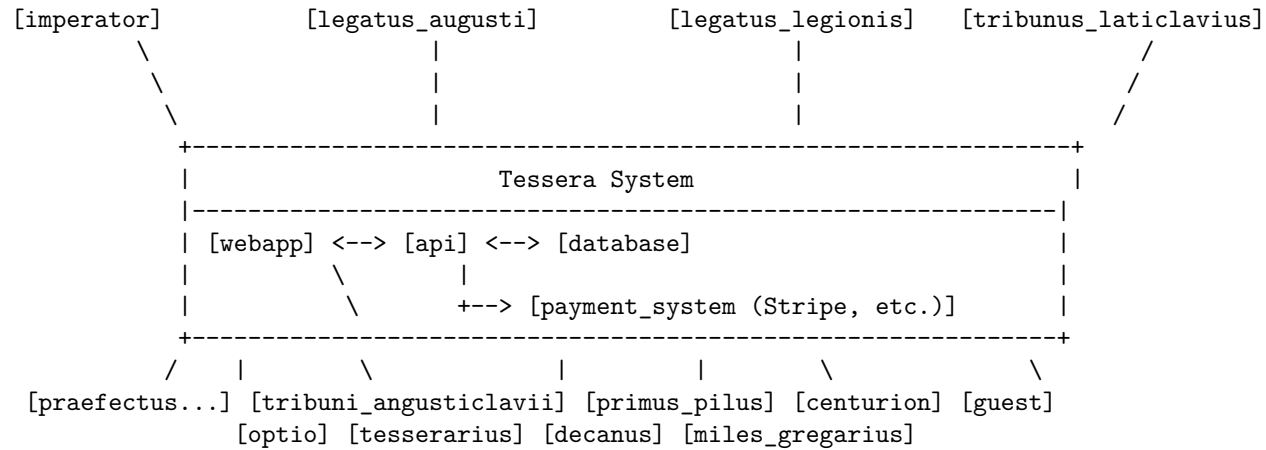
Tessera: C4 Model Documentation

System Context Diagram

Overview:

Tessera is a web application for managing and selling tickets to themed (antique) events, supporting different user roles and integrating with an external payment service.

Diagram:



Key Relationships:

- All user roles interact with Tessera (webapp).
 - Tessera (API) communicates with a MySQL database and an external payment system (e.g., Stripe).
-

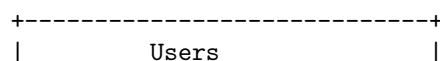
Container Diagram

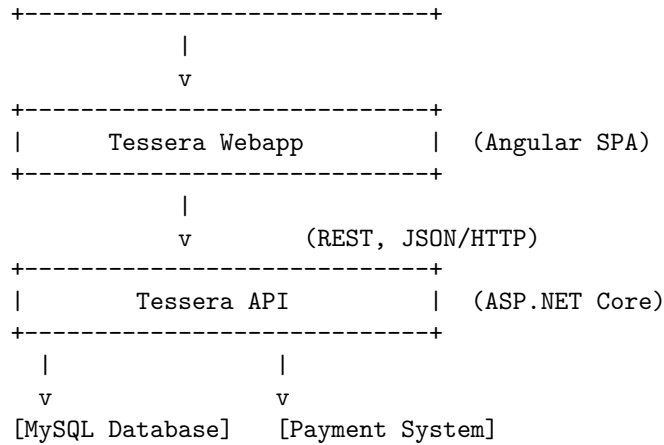
Overview:

Tessera is split into three main containers:

- **Webapp:** Angular SPA for event/ticket viewing and purchase.
- **API:** ASP.NET Core backend for event/ticket management.
- **Database:** MySQL for storing events, tickets, and users.
- **Payment System:** External payment provider.

Diagram:





Relations:

- Users interact with Webapp (different UI per role).
- Webapp calls API (REST).
- API accesses MySQL (SQL), and Payment System (HTTPS).

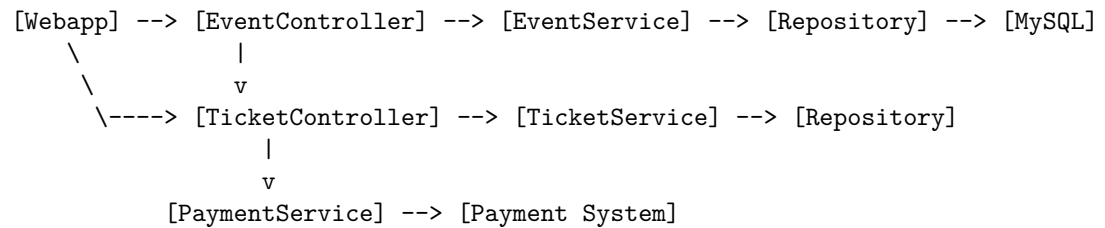
Component Diagram (API)

Overview:

Key backend components (all in ASP.NET Core):

- **EventController**: Handles event endpoints (GET /api/events)
- **TicketController**: Handles ticket endpoints (POST /api/tickets)
- **PaymentService**: Integrates with payment system
- **EventService/TicketService**: Business logic
- **Repository**: Database access

Diagram:



Component Diagram (Webapp)

Overview:

Key frontend components (all in Angular/TypeScript):

- **EventListComponent:** List events
- **TicketPurchaseComponent:** Ticket purchase form
- **AdminDashboardComponent:** Admin UI

Diagram:

```
[User] --> [EventListComponent] --\
      [TicketPurchaseComponent]----> (API)
      [AdminDashboardComponent]---/
```

Each component calls the relevant API endpoints.

Roles Table

Role	Description	Main Permissions/Interfaces Used
Imperator	/	Admin dashboard, all event/ticket actions
Legatus		Admin dashboard, system settings
Augusti		Admin/event management
Legatus		Admin/event management
Legionis		Admin/event management
Tribunus		Event organization interface
Laticlavius		Event moderation interface
Praefectus		Ticket/event moderation
Castorum		Ticket management
Tribuni		Ticket purchase/reservation
Angusticlavii		Ticket reservation
Primus Pilus	/	Ticket purchase
Centurion		Ticket purchase
Optio	/	Event listing (read-only)
Tesserarius		
Decanus		
Miles		
Gregarius		
Guest		

Key Relationships

- **Users → Webapp:** Interact through different interfaces based on role.
 - **Webapp → API:** REST API for all business logic.
 - **API → DB:** CRUD operations.
 - **API → Payment System:** Payment processing.
-

External Systems

- **Payment System (Stripe or similar):**
 - Used for payment processing.
 - Accessed by API during ticket purchase.
-

Technology Stack

- **Frontend:** Angular (TypeScript, HTML/CSS)
 - **Backend:** ASP.NET Core (C#)
 - **Database:** MySQL
 - **External Payment:** Stripe (or similar)
-

Summary

Tessera provides a robust and role-based system for managing themed events and tickets, with a clear separation of concerns (frontend, backend, database, external payments) and fine-grained access control.