

# Event Market Platform

Fernando Rocha Urbano

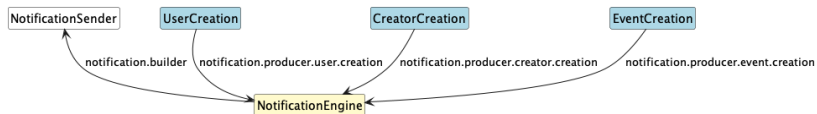
April 2024

# General Topic

- ▶ Event Ticket Purchase Platform
- ▶ Key components:
  - ▶ Event creation
  - ▶ Purchase app
  - ▶ Notification system
  - ▶ Payment validation
  - ▶ Event recommender engine

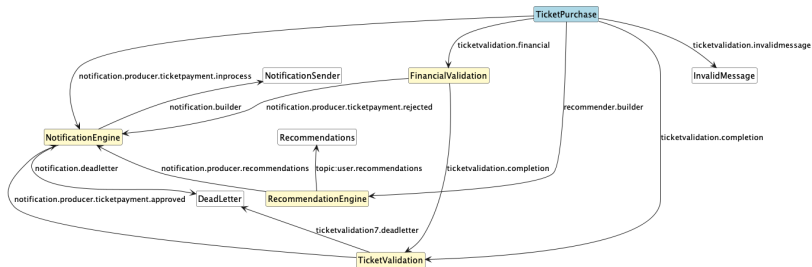
# Project Separation

## ► Creation Processes



# Project Separation

## ► Ticket Purchase Processes



# Applications

- ▶ Main Application with Database connection:
  - ▶ **Event Market:** event-market
- ▶ Creation Camel Applications:
  - ▶ **Event Creation:** event-market-event-creation
  - ▶ **User Creation:** event-market-user-creation
  - ▶ **Creator Creation:** event-market-event-creation
- ▶ Ticket Purchase Camel Applications:
  - ▶ **Ticket Purchase:** event-market-ticket-purchase
  - ▶ **Financial Validation:** event-market-financial-validation
  - ▶ **Ticket Validation:** event-market-ticket-validation
  - ▶ **Event Recommender Engine:**  
event-market-recommendation-engine
  - ▶ **Notification Engine:** event-market-notification-engine

# Iterator and Singleton Patterns

## ▶ Iterator Pattern

- ▶ Inside the **Ticket Purchase** app.
- ▶ Used in *TicketTypeIterator* class to iterate over ticket types.
- ▶ Helps users browse ticket options.

## ▶ Singleton Pattern

- ▶ Inside the **Financial Validation** app.
- ▶ Used in *FinancialValidator* class.
- ▶ Ensures a single instance for validating ticket requests.

# Strategy and Facade Patterns

## ► Strategy Pattern

- Inside the **Financial Validation** app.
- Used in *PaymentMethod* class.
- Supports various payment methods like *CheckingsAccountPayment* and *CardPayment*.

## ► Facade Pattern

- Inside the **Event Recommender Engine**.
- Used in *EventRecommender* class.
- Simplifies interactions with recommendation models.

# Template Method Pattern

- ▶ **Template Method Pattern**

- ▶ Inside the **Event Recommender Engine**.
- ▶ Used in *AbstractRecommenderModel* and its derived classes.
- ▶ Provides structure for creating recommendation models.



# Event Creation

- ▶ Creates *Event* instances with:
  - ▶ *TicketType* instances
  - ▶ *Location* instance
  - ▶ *Creator* instance

# User and Creator Creation

- ▶ Similar to Event Creation
- ▶ Communicates with **Notification Engine**

# Ticket Purchase

- ▶ Handles *TicketRequest* instances.
- ▶ Interacts with:
  - ▶ **Financial Validation**
  - ▶ **Ticket Validation**
  - ▶ **Event Recommender Engine**
  - ▶ **Notification Engine**

# Financial Validation

- ▶ Validates payment for *TicketRequest* instances.
- ▶ Uses *FinancialValidator* (Singleton) and *PaymentMethod* (Strategy).

# Ticket Validation

- ▶ Creates *Ticket* instances from approved *TicketRequest*.
- ▶ Updates *User* with new *Ticket*.

# Event Recommender Engine

- ▶ Recommends events to *User* based on purchases.
- ▶ Uses *EventRecommender* (Facade) and *AbstractRecommenderModel* (Template Method).

# Notification Engine

- ▶ Translates system information into notifications.
- ▶ Sends messages to users and creators.