Microservices Architecture

CSSE6400

Richard Thomas

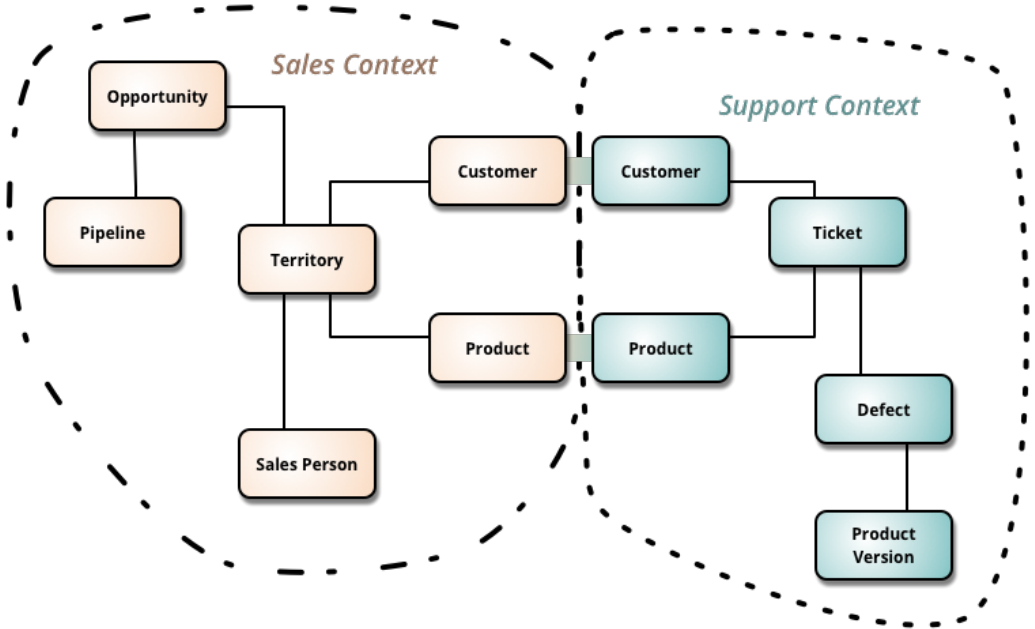
April 11, 2022

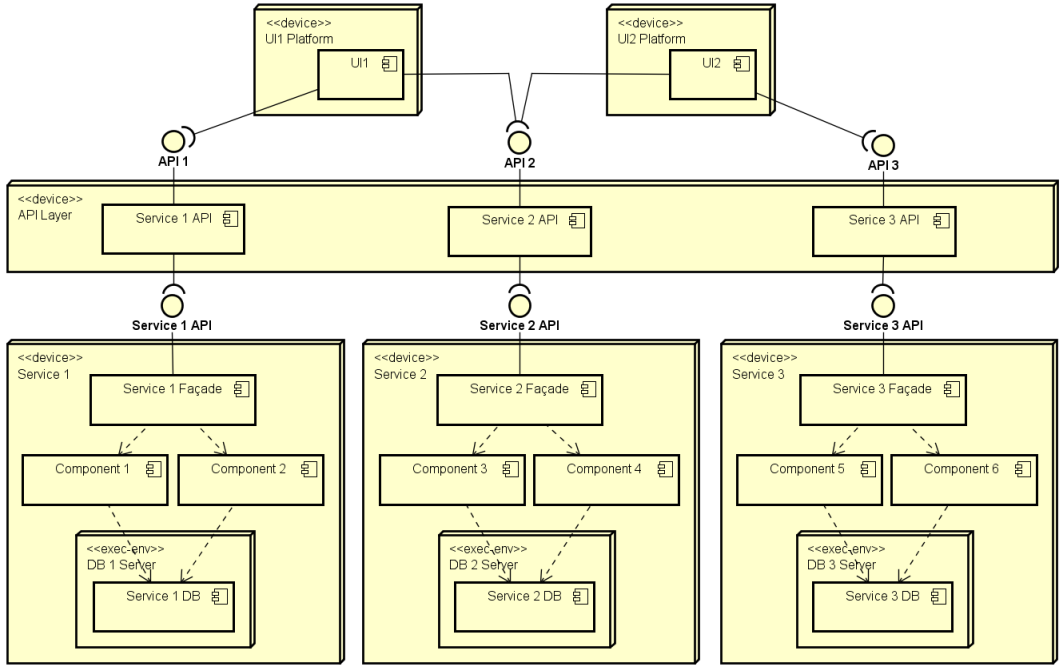
Microservices

Inspired by DDD

Definition 1. Bounded Context

Logical boundary of a domain where particular terms and rules apply consistently.

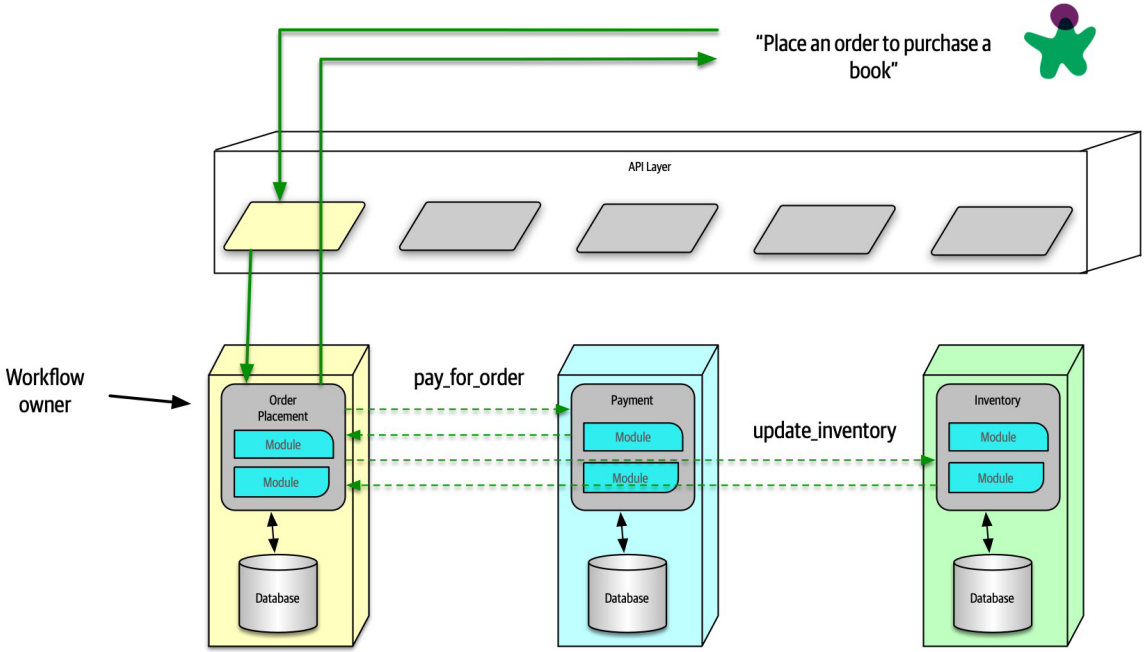

Sales Context
Support Context
Opportunity
Pipeline
Territory
Customer
Product
Sales Person
Customer
Ticket
Product
Defect
Product Version

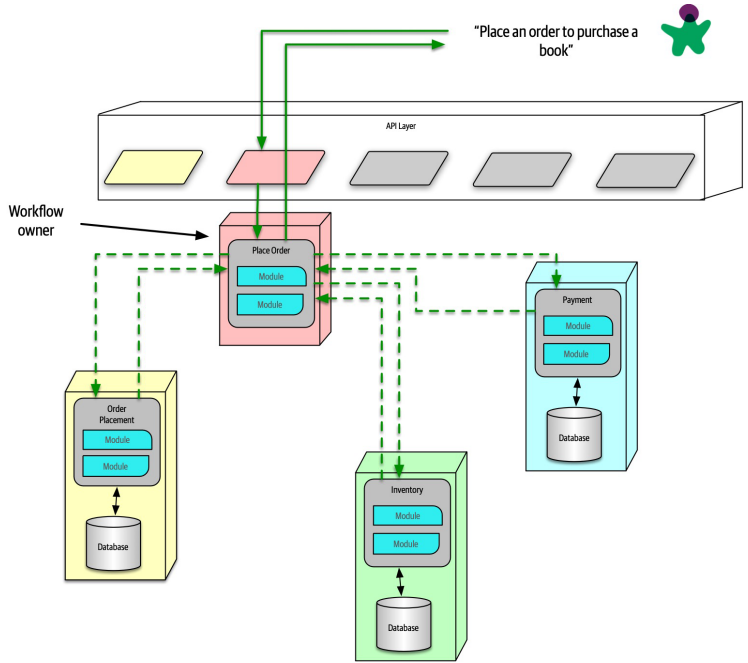

<<device>>
UI1 Platform
UI1
<<device>>
UI2 Platform
UI2
API 1
API 2
API 3
<<device>>
API Layer
Service 1 API
Service 2 API
Serice 3 API
Service 1 API
Service 2 API
Service 3 API
<<device>>
Service 1
Service 1 Façade
Component 1
Component 2
<<exec env>>
DB 1 Server
Service 1 DB
<<device>>
Service 2
Service 2 Façade
Component 3
Component 4
<<exec env>>
DB 2 Server
Service 2 DB
<<device>>
Service 3
Service 3 Façade
Component 5
Component 6
<<exec env>>
DB 3 Server
Service 3 DB

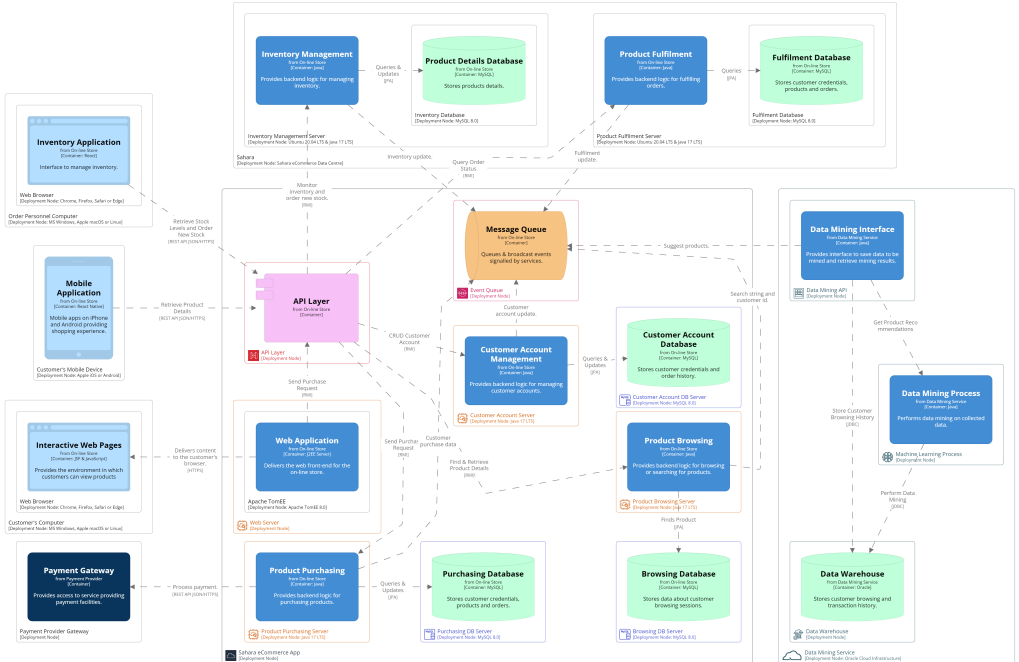
Choreography & Orchestration

Choreography Similar to event-driven *broker*

Orchestration Similar to event-driven *mediator*


"Place an order to purchase a book"
API Layer
Workflow owner
Order Placement
Module
Module
Database
pay_for_order
Payment
Module
Module
Database
update_inventory
Inventory
Module
Module
Database


"Place an order to purchase a book"
API Layer
Workflow owner
Place Order
Module
Module
Payment
Module
Module
Database
Order Placement
Module
Module
Database
Inventory
Module
Module
Database


Inventory Management
Provides backend logic for managing inventory.
Queries & Updates [JPA]
Product Details Database
Stores products details.
Inventory Database
Inventory Management Server
Product Fulfilment
Provides backend logic for fulfilling orders.
Queries [JPA]
Fulfilment Database
Stores customer credentials, products and orders.
Fulfilment Database
Product Fulfilment Server
Sahara
Inventory update.
Fulfilment update.
Inventory Application
Interface to manage inventory.
Web Browser
Order Personnel Computer
Monitor inventory and order new stock. [RMI]
Query Order Status [RMI]
Retrieve Stock Levels and Order New Stock [REST API (JSON/HTTPS)]
Message Queue
Queues & broadcast events signalled by services.
Event Queue
Suggest products.
Data Mining Interface
Provides interface to save data to be mined and retrieve mining results.
Data Mining API
Mobile Application
Mobile apps on iPhone and Android providing shopping experience.
Retrieve Product Details [REST API (JSON/HTTPS)]
API Layer
API Layer
Customer's Mobile Device
CRUD Customer Account [RMI]
Customer account update.
Search string and customer id.
Get Product Recommendations
Customer Account Management
Provides backend logic for managing customer accounts.
Queries & Updates [JPA]
Customer Account Database
Stores customer credentials and order history.
Customer Account DB Server
Customer Account Server
Data Mining Process
Performs data mining on collected data.
Machine Learning Process
Store Customer Browsing History [JDBC]
Send Purchase Request [RMI]
Web Application
Delivers the web front-end for the on-line store.
Apache TomEE
Web Server
Send Purchase Request [RMI]
Customer Purchase data
Find & Retrieve Product Details [RMI]
Product Browsing
Provides backend logic for browsing or searching for products.
Product Browsing Server
Perform Data Mining [JDBC]
Interactive Web Pages
Provides the environment in which customers can view products.
Delivers content to the customer's browser. [HTTPS]
Web Browser
Customer's Computer
Finds Product [JPA]
Payment Gateway
Provides access to service-providing payment facilities.
Process payment. [REST API (JSON/HTTPS)]
Payment Provider Gateway
Product Purchasing
Provides backend logic for purchasing products.
Queries & Updates [JPA]
Purchasing Database
Stores customer credentials, products and orders.
Product Purchasing Server
Purchasing DB Server
Browsing Database
Stores data about customer browsing sessions.
Browsing DB Server
Data Warehouse
Stores customer browsing and transaction history.
Data Warehouse
Sahara eCommerce App
Data Mining Service

Pros & Cons

- Modularity
- Extensibility
- Reliability
- Interoperability
- Scalability
- Security
- Deployability
- Testability
- Simplicity