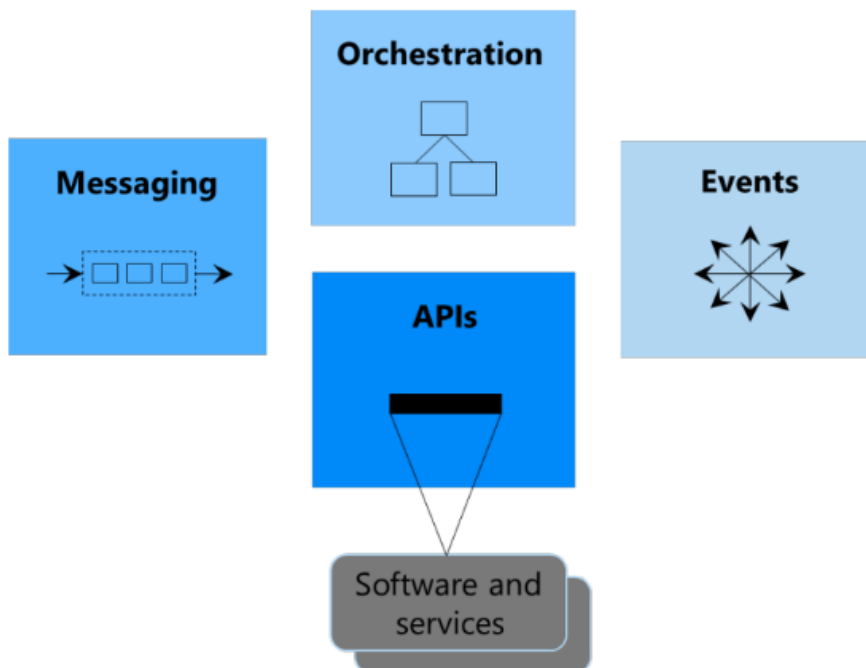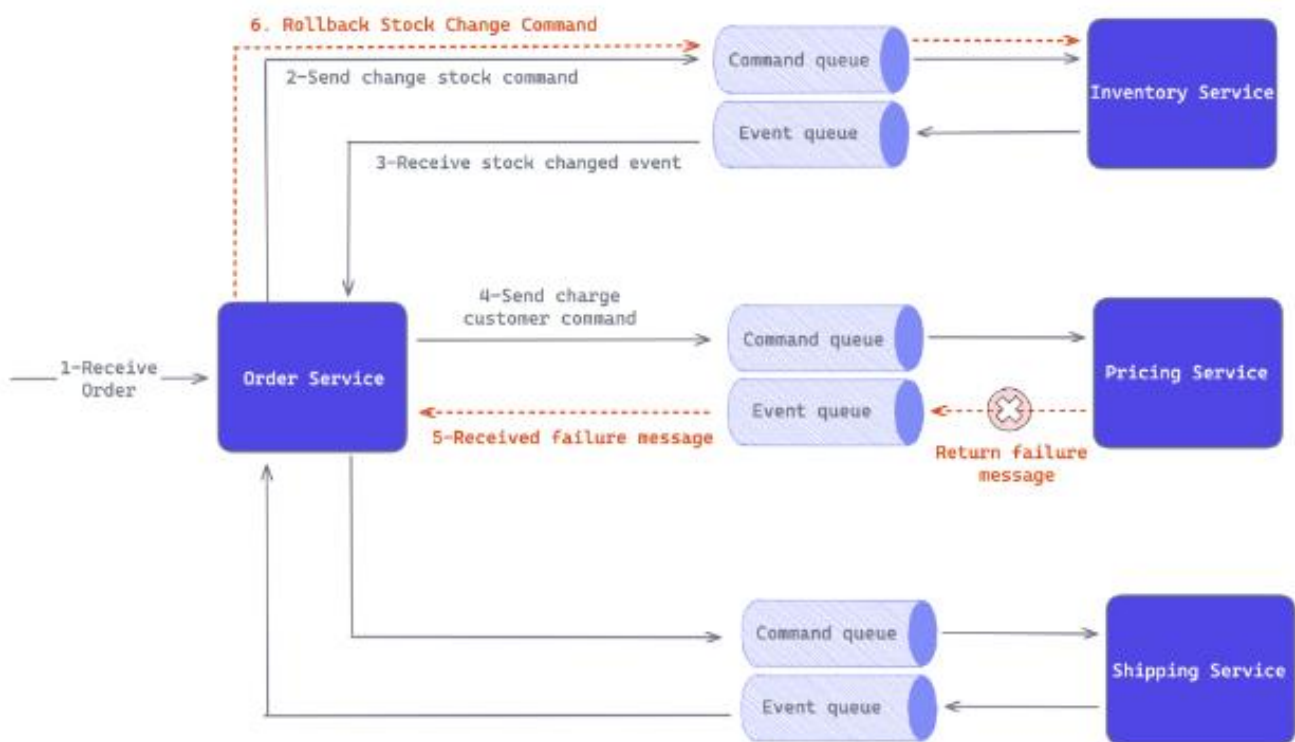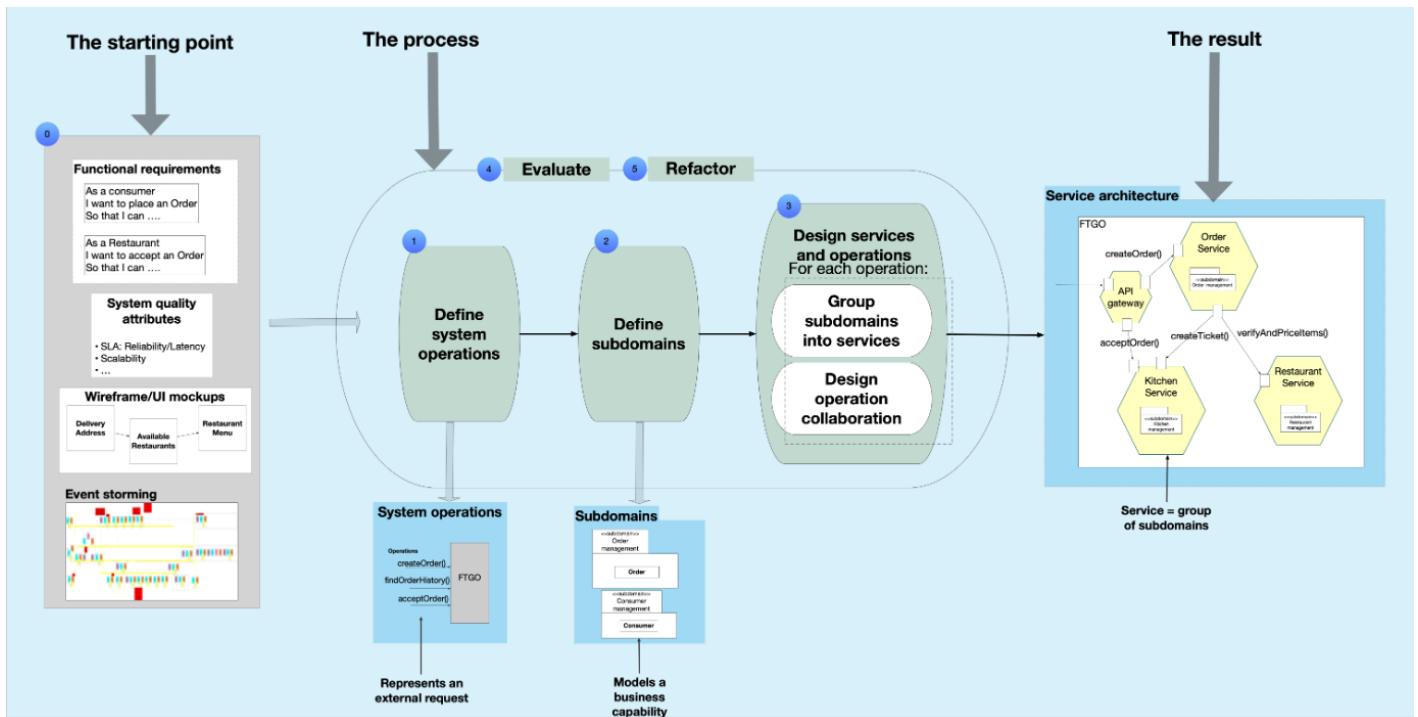Integration Architecture Guiding Principles

Integration architecture design - Azure Architecture Center | Microsoft Learn

The starting point

Functional requirements

As a consumer
I want to place an Order
So that I can ….

As a Restaurant
I want to accept an Order
So that I can ….

System quality attributes
- SLA: Reliability/Latency
- Scalability
- …

Wireframe/UI mockups

Delivery Address — Available Restaurants — Restaurant Menu

Event storming

The process

Evaluate   **Refactor**

1 Define system operations

2 Define subdomains

3 Design services and operations
For each operation:
- Group subdomains into services
- Design operation collaboration

System operations

Operations
createOrder()
findOrderHistory()
acceptOrder()
FTGO

Represents an external request

Subdomains

Order management
Order

Consumer management
Consumer

Models a business capability

The result

Service architecture

FTGO
- createOrder()
- API gateway
- Order Service (Order management)
- acceptOrder()
- createTicket()   verifyAndPriceItems()
- Kitchen Service (Kitchen management)
- Restaurant Service (Restaurant management)

Service = group of subdomains

---

6. Rollback Stock Change Command

2-Send change stock command

Command queue — Inventory Service

Event queue

3-Receive stock changed event

1-Receive Order → Order Service

4-Send charge customer command

Command queue — Pricing Service

Event queue

5-Received failure message

Return failure message
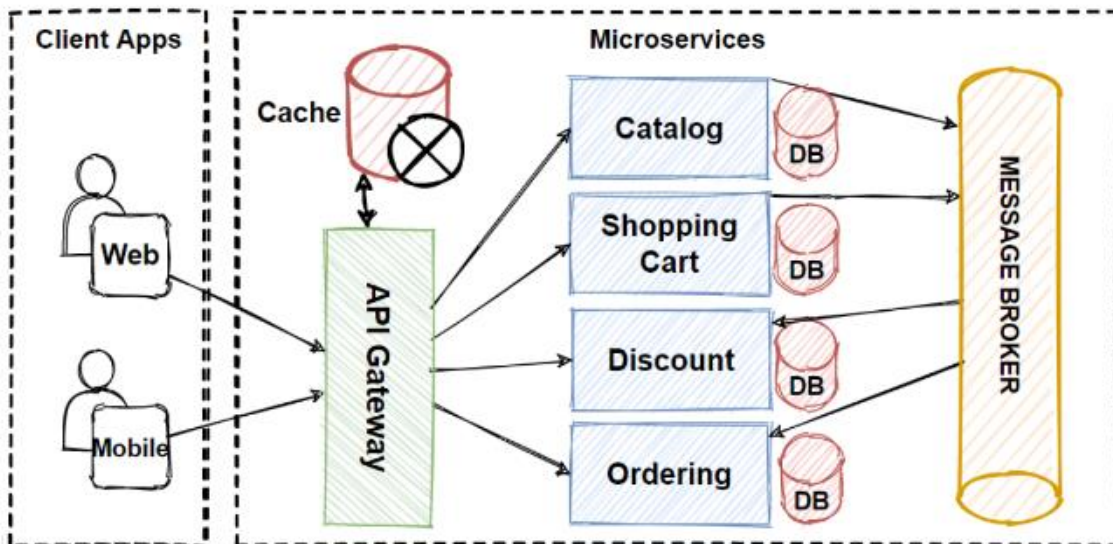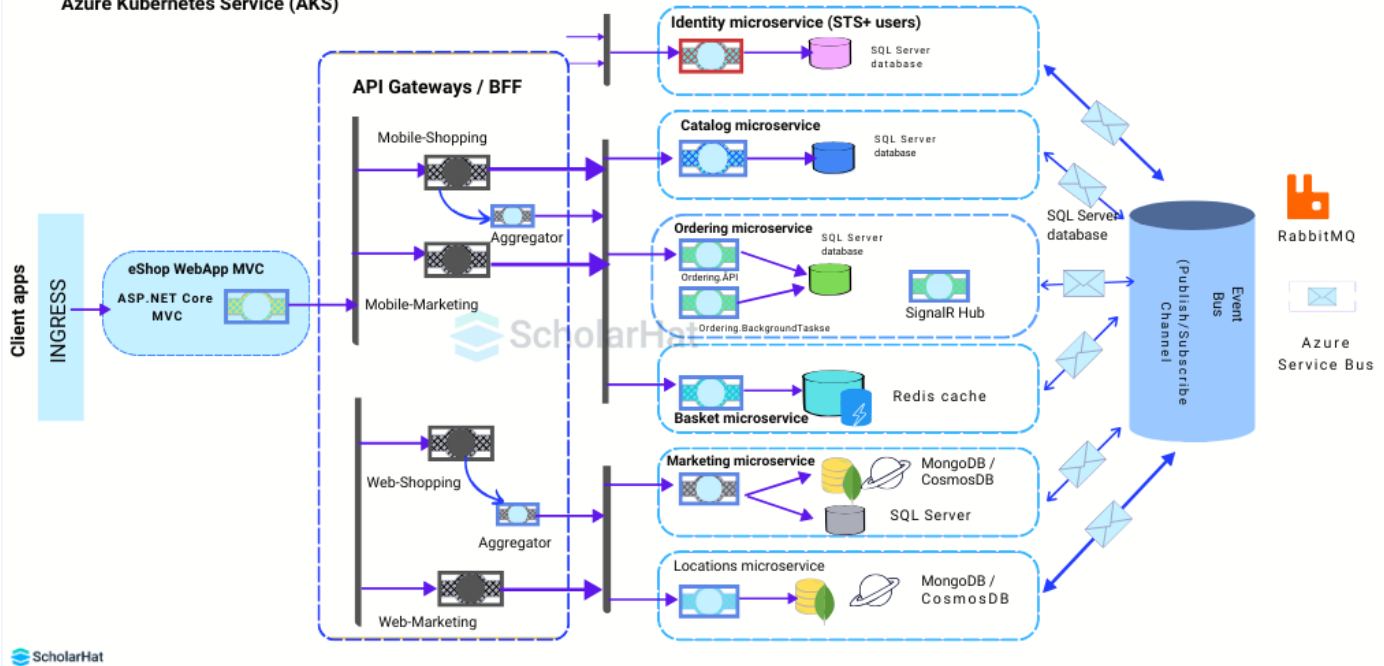
Command queue — Shipping Service

Event queue

Core Components of Microservices Architecture

Visit >

Microservices Architecture at eShop

Azure Kubernetes Service (AKS)

API Gateways / BFF

Client apps — INGRESS

eShop WebApp MVC
ASP.NET Core MVC

Mobile-Shopping

Aggregator

Mobile-Marketing

Web-Shopping

Aggregator

Web-Marketing

Identity microservice (STS+ users)
SQL Server database

Catalog microservice
SQL Server database

Ordering microservice
Ordering.API
Ordering.BackgroundTaskse
SQL Server database
SignalR Hub

Basket microservice
Redis cache

Marketing microservice
MongoDB / CosmosDB
SQL Server

Locations microservice
MongoDB / CosmosDB

SQL Server database

Event Bus (Publish/Subscribe Channel)

RabbitMQ

Azure Service Bus

ScholarHat

Client Apps

Microservices

Cache

Web

Mobile

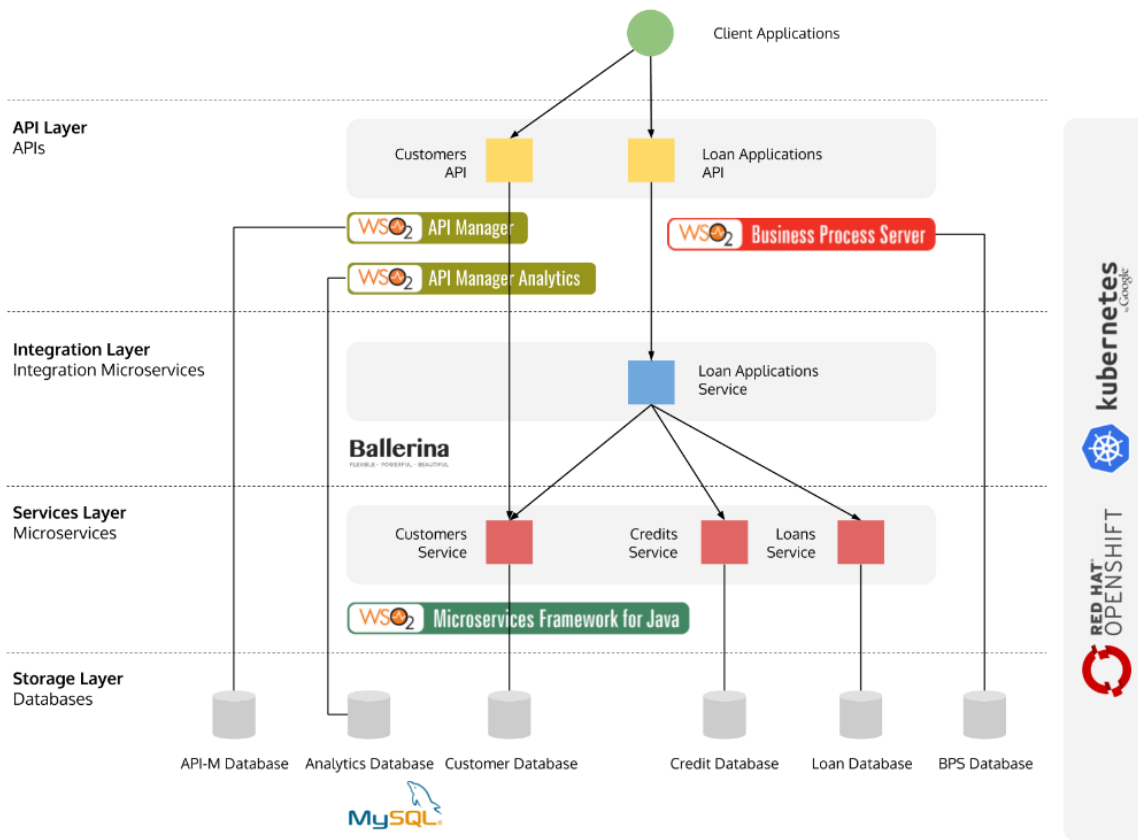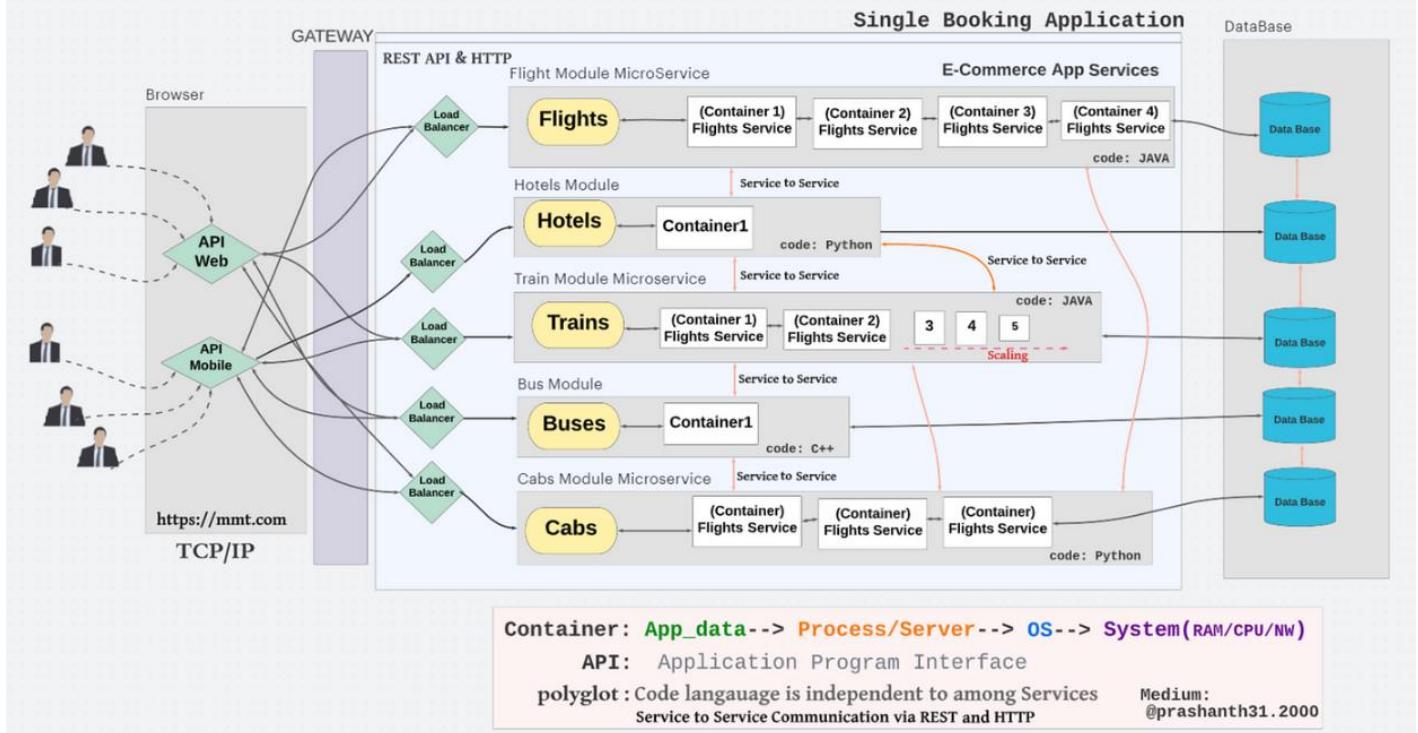API Gateway

Catalog — DB

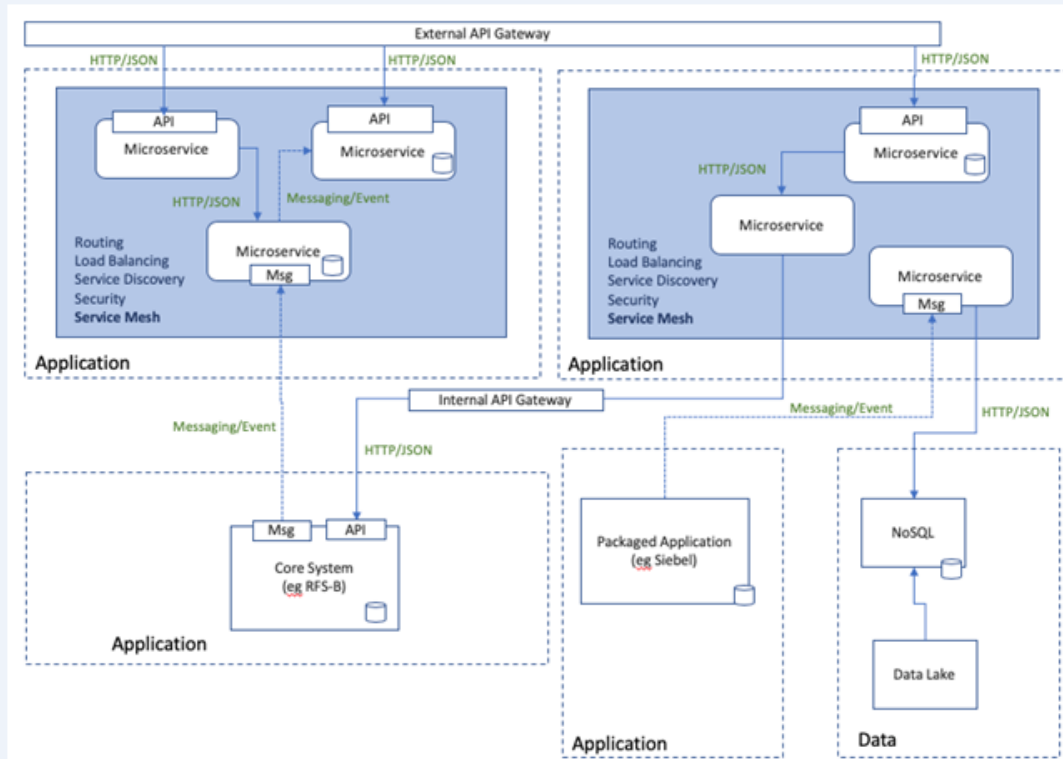Shopping Cart — DB

Discount — DB

Ordering — DB

MESSAGE BROKER

Microservices Distributed Caching | by Mehmet Ozkaya | Design Microservices Architecture with Patterns & Principles | Medium

Visit >

Microservices Architecture

Single Booking Application

GATEWAY

REST API & HTTP

E-Commerce App Services

Browser

DataBase

Flight Module MicroService

Flights → (Container 1) Flights Service → (Container 2) Flights Service → (Container 3) Flights Service → (Container 4) Flights Service → Data Base

code: JAVA

Hotels Module

Hotels — Container1

code: Python

Service to Service

Service to Service

→ Data Base

Train Module Microservice

Service to Service

Trains → (Container 1) Flights Service → (Container 2) Flights Service

code: JAVA

3 4 5

Scaling

→ Data Base

Bus Module

Service to Service

Buses — Container1

code: C++

→ Data Base

Cabs Module Microservice

Service to Service

Cabs → (Container) Flights Service → (Container) Flights Service → (Container) Flights Service

code: Python

→ Data Base

API Web

API Mobile

Load Balancer (×5)

https://mmt.com

TCP/IP

```
Container: App_data --> Process/Server --> OS --> System(RAM/CPU/NW)
     API:  Application Program Interface
polyglot : Code langauage is independent to among Services          Medium:
          Service to Service Communication via REST and HTTP      @prashanth31.2000
```

Client Applications

API Layer
APIs

Customers API

Loan Applications API

WS0₂ API Manager

WS0₂ API Manager Analytics

WS0₂ Business Process Server

Integration Layer
Integration Microservices

Loan Applications Service

Ballerina
FLEXIBLE · POWERFUL · BEAUTIFUL

Services Layer
Microservices

Customers Service

Credits Service

Loans Service

WS0₂ Microservices Framework for Java

Storage Layer
Databases

API-M Database   Analytics Database   Customer Database   Credit Database   Loan Database   BPS Database

MySQL

kubernetes by Google

RED HAT OPENSHIFT

Top Diagram

External API Gateway

HTTP/JSON   HTTP/JSON   HTTP/JSON

Application

API
Microservice

API
Microservice

HTTP/JSON   Messaging/Event

Microservice
Msg

Routing
Load Balancing
Service Discovery
Security
Service Mesh

Application

API
Microservice

HTTP/JSON

Microservice

Microservice
Msg

Routing
Load Balancing
Service Discovery
Security
Service Mesh

Internal API Gateway

Messaging/Event   HTTP/JSON   Messaging/Event   HTTP/JSON

Msg   API

Core System
(eg RFS-B)

Application

Packaged Application
(eg Siebel)

Application

NoSQL

Data Lake

Data

DIGIT Diagram

DIGIT

Citizen Web Dashboard

WhatsApp

Employee Dashboard

| Service 1 Front End | Service 2 Front End | Other Front Ends | Chat Service | Service Specific Employee UI | Employee Inbox | Dashboard Front End |

API Gateway

Authenticate   Authorize

| Location Service | User Service | Report Service | Billing Service | Collection Registry | Payment Gateway | PDF Service | Dashboard Backend Service |

| Employee Registry | Role Service | **Any Service** | Write/Update | Queue Service | | | Workflow |

Read

| Encryption | Language Service | **Any Registry** | Persister | Indexer |

| Master Data Service | File Store | Relational DB | Searcher | SMS | Email | App Notification | Telemetry |

Analytical DB
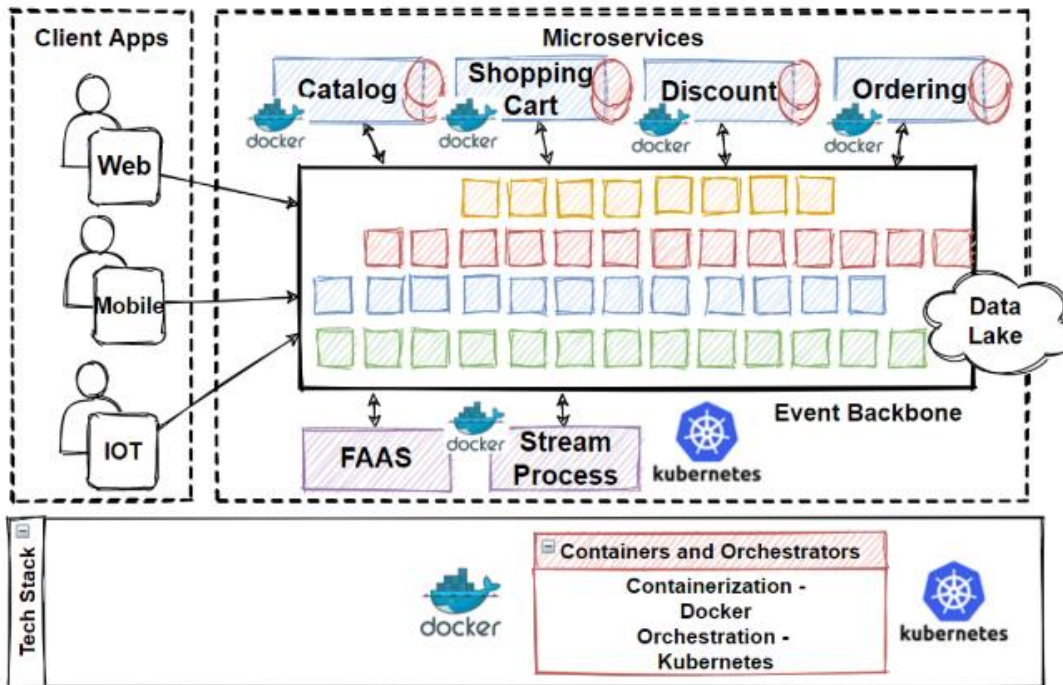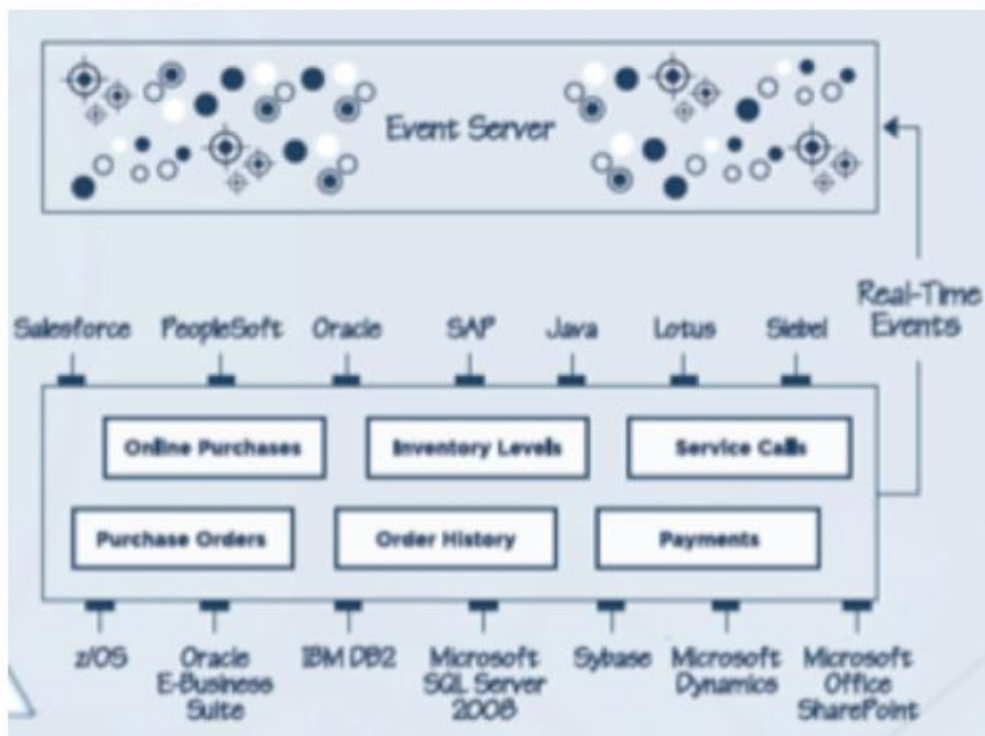
Existing Service   DIGIT Service   DIGIT Accelerator   Custom Development

* List of DIGIT services is not complete

Client Apps

Microservices

Catalog · Shopping Cart · Discount · Ordering

Web · Mobile · IOT

Data Lake

Event Backbone

FAAS · Stream Process · kubernetes

Tech Stack

Containers and Orchestrators
Containerization - Docker
Orchestration - Kubernetes

docker · kubernetes

Microservices Deployments with Containers and Orchestrators | by Mehmet Ozkaya | Design Microservices Architecture with Patterns &...

Visit >

Event Server

Salesforce · PeopleSoft · Oracle · SAP · Java · Lotus · Siebel

Real-Time Events

Online Purchases · Inventory Levels · Service Calls

Purchase Orders · Order History · Payments

z/OS · Oracle E-Business Suite · IBM DB2 · Microsoft SQL Server 2008 · Sybase · Microsoft Dynamics · Microsoft Office SharePoint

Diagram 1: Microservices Layered Architecture

External User/Device

APIs

- API level policy enforcement routing
- Versioning

Gateway Layer

μS gateway | μS gateway | μS gateway

μS Gateway | μS Gateway | μS Gateway | μS Gateway

- Data Format Transformation for external device
- Aggregation/Split/Normalizing
- Routing between App domains

Composite Layer

Load Balance / Service Discovery/Network Management

μS Composite | μS Composite | μS Composite | μS Composite

Containerized scalable capability

Event Bus

μS caching — JDG

- Composition of microservices
- Triggers events in domain
- Content Enrich
- Applying Integration patterns
- Caching

Base μService Layer

μS (BRMS) | μS (RHOAR) | μS (PHP) | μS (Python) | μS (Java) | μS (Fuse) | μS

- Simple microservice
- Divide upon function business capabilities

Legacy Systems

μS | μS — Anti-corruption Layer

- Hides legacy systems complexity
- SOAP/REST translation

Diagram 2: EDA Overview

Channel Apps
- Mobile Apps
- Client Apps
- Internal Apps
- Partner Apps
- Micro Frontends

MicroServices Chassis
- Service Discovery
- Circuit Breakers
- DevOps
- Monitoring
- Logging
- Distributed Tracing
- Authentication
- Authorization
- Audit

Application Integrations
- Enterprise Applications
- B2B Gateway
- IoT Devices

API Layer
- API Gateway
- Process API
- System API
- Data API

Event Backbone
- Event Broker
- Queues/Topics
- Event Streams
- Event Router(s)
- (De) Serializers
- Event Store

Data Layer
- Data-Grid / Cache
- Metrics Store
- Data Warehouse
- ODS

Services Layer
- Orchestration Services
- Microservice | Microservice | Microservice | Microservice
- Data & Analytics Services

Cloud Platform (Private, Hybrid, Public)

Architectural considerations for event-driven microservices-based systems - IBM Developer

Visit >