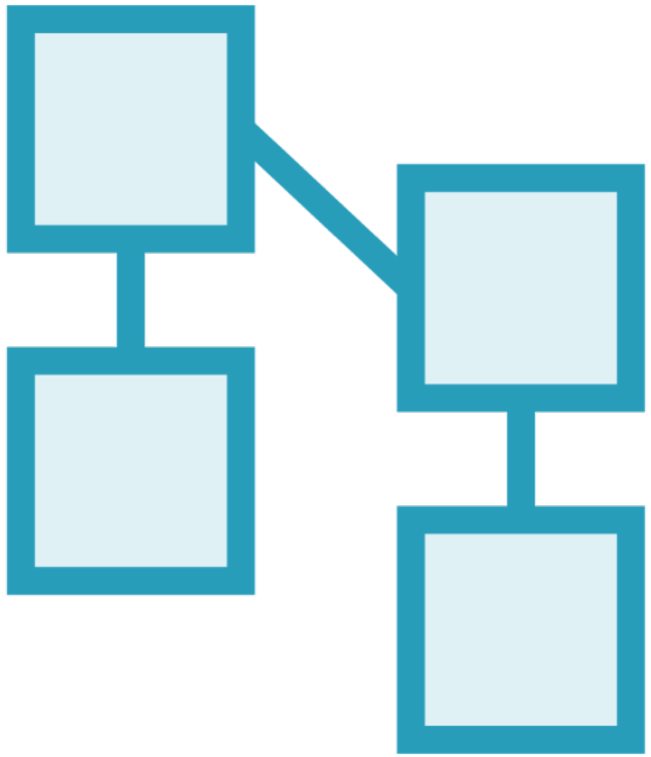
Implementing a Data Management Strategy for an ASP.NET Core Microservices Architecture

DATA MANAGEMENT ARCHITECTURE WITH MICROSERVICES

Neil Morrissey
SOLUTIONS ARCHITECT

@morrisseycode

www.neilmorrissey.net



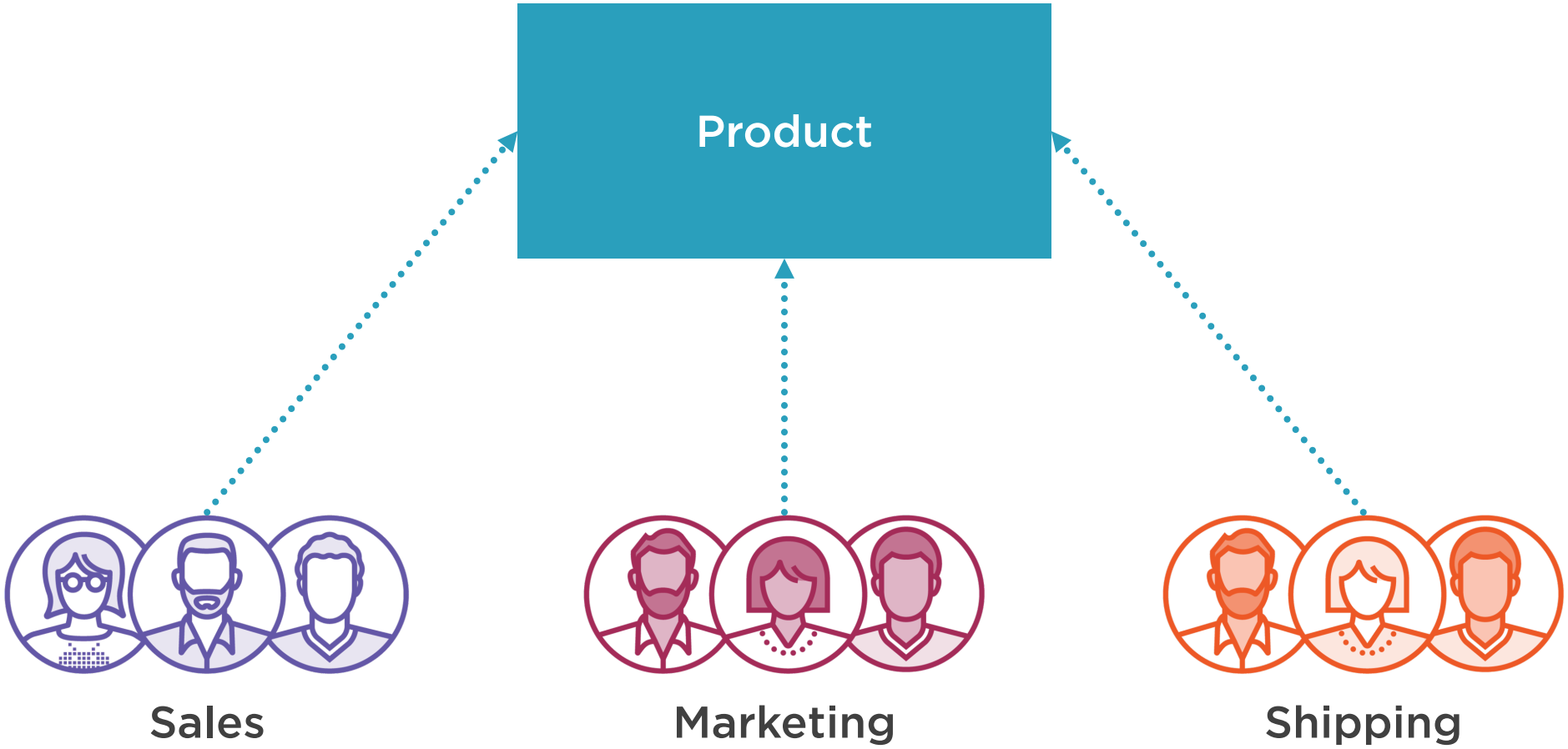
Domain Driven Design

Bounded Context Pattern

- Dividing domain into subdomains

Structure and language of software should match the business domain

Terms can mean different things for different groups


Product
Sales
Marketing
Shipping

Product

Sales

Product

Marketing

Product

Shipping

Product

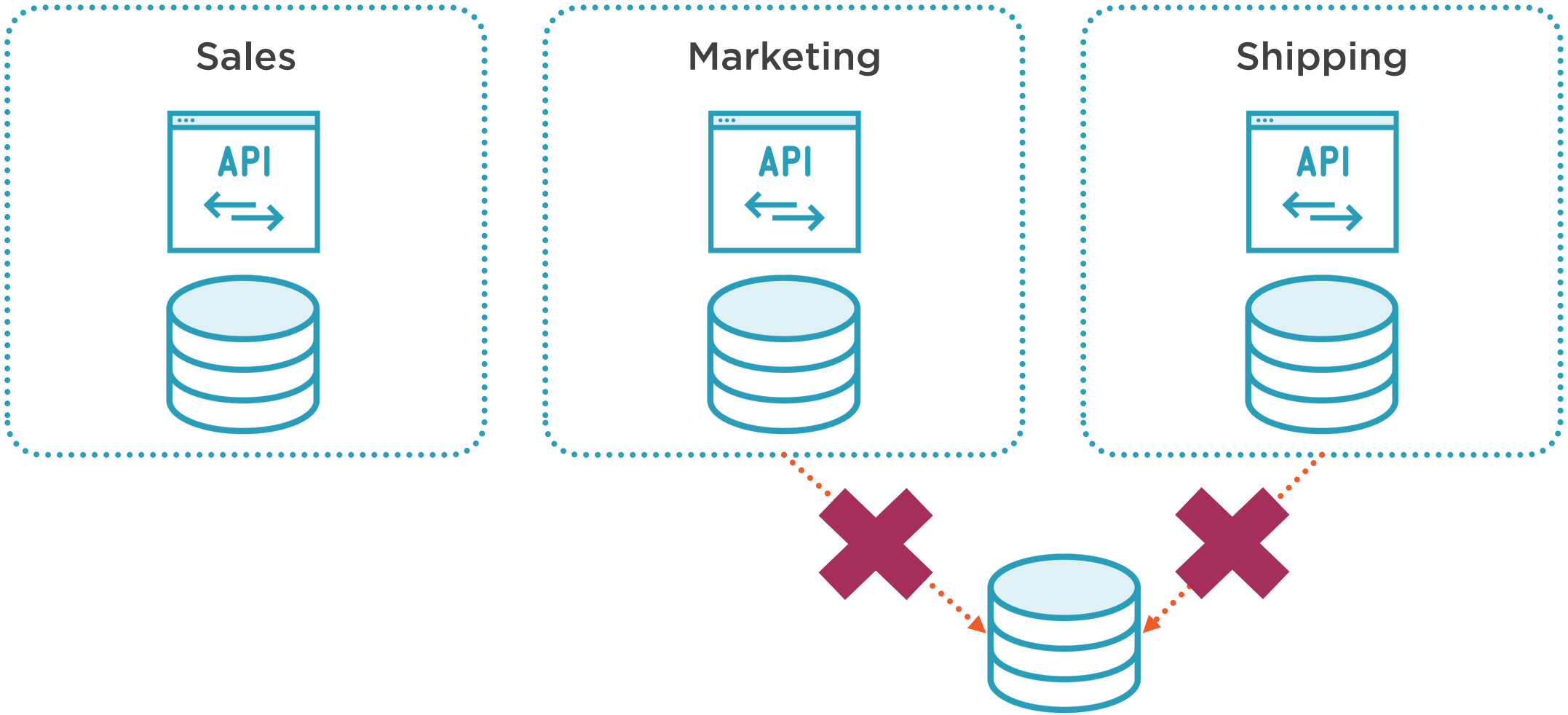
Sales

Product

Marketing

Product

Shipping


Sales
API
Marketing
API
Shipping
API

Opportunities and Challenges

Data Model

Data Storage

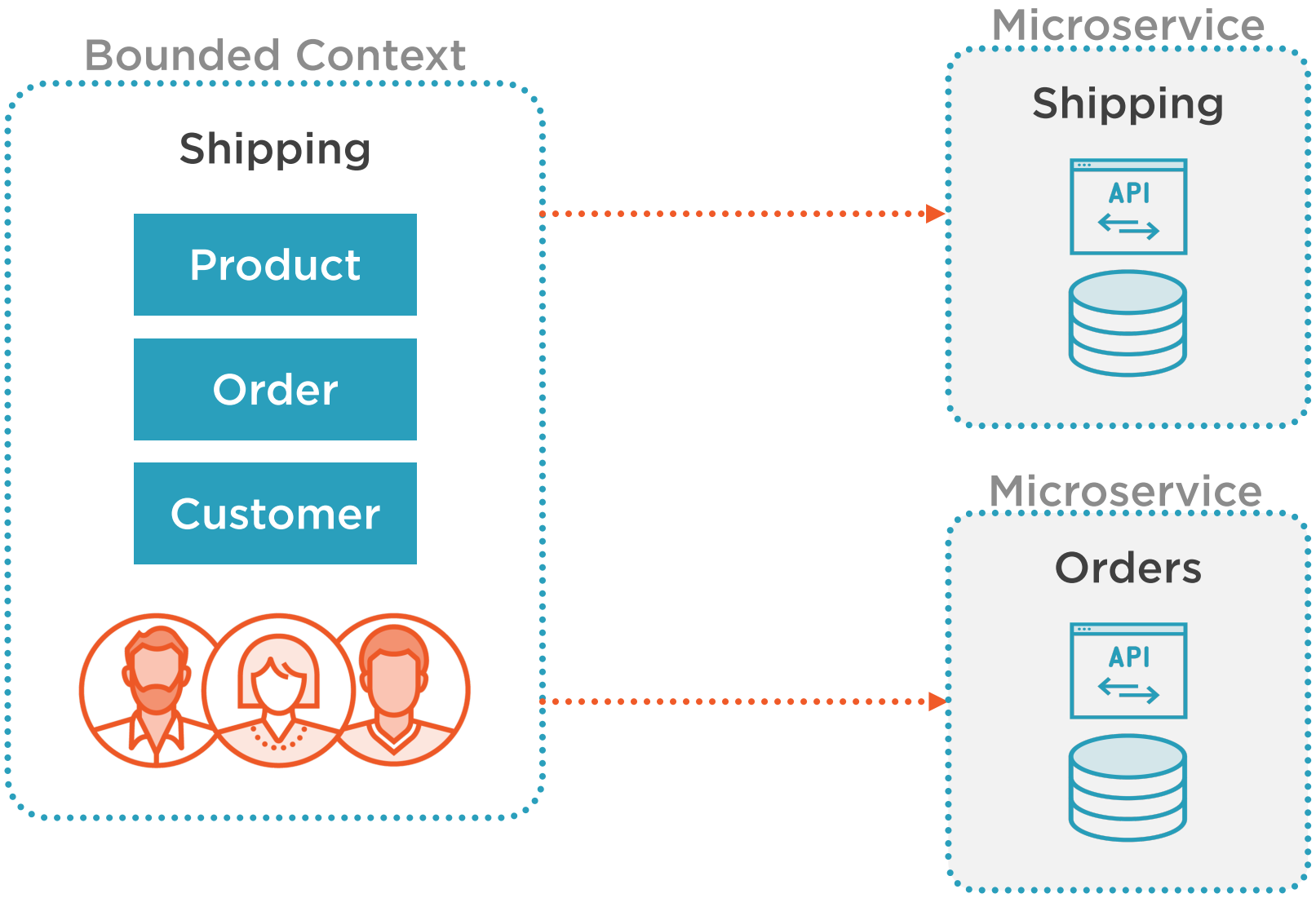
Consistency

Complex Queries

Module Overview

- Microservice design
- Polygot Persistence
- Eventual consistency
- Complex queries across microservices
- Sample project architecture
- Configuring the sample project
- Understanding the sample project code

Mapping Bounded Contexts to Microservices



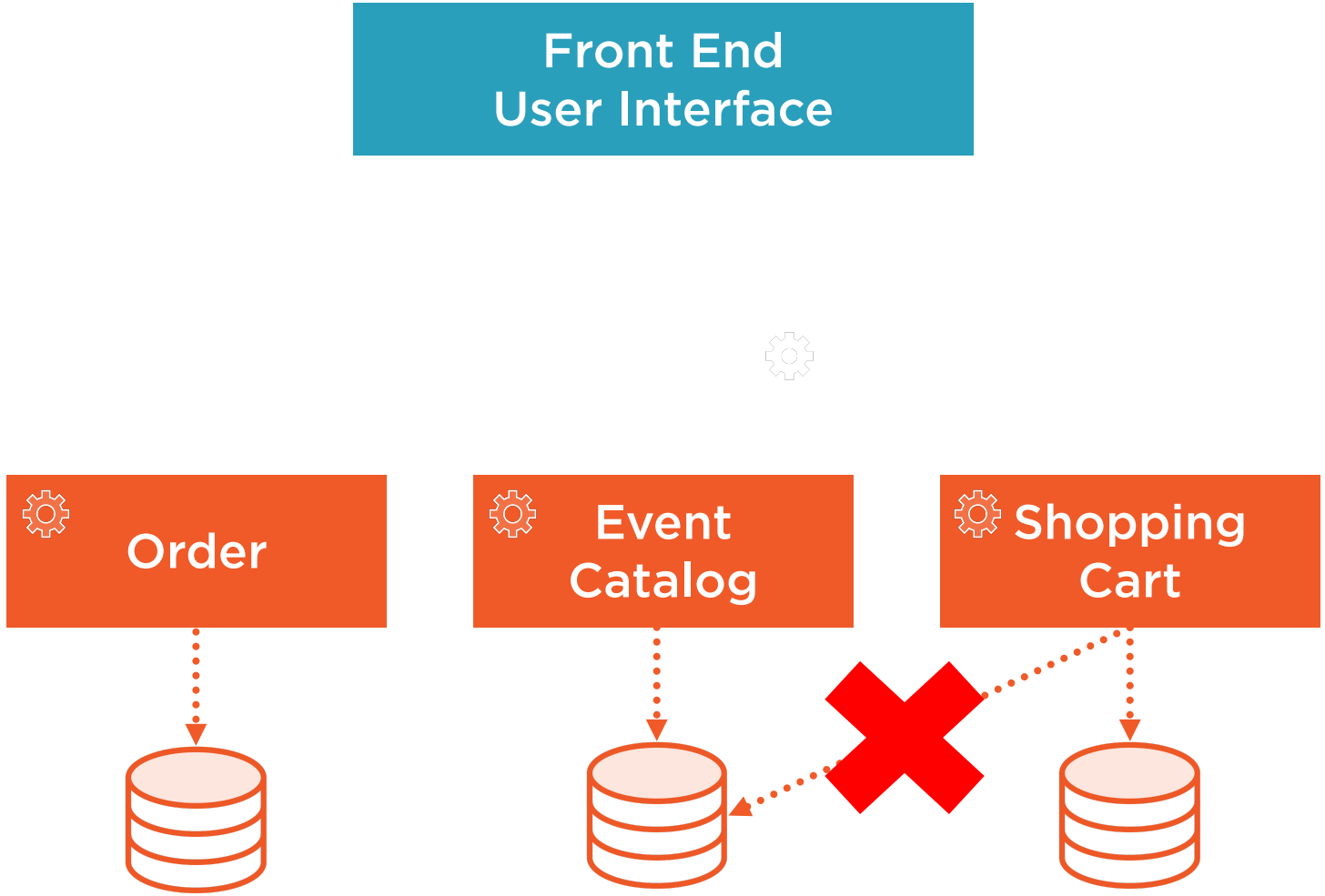
Microservice Design Considerations

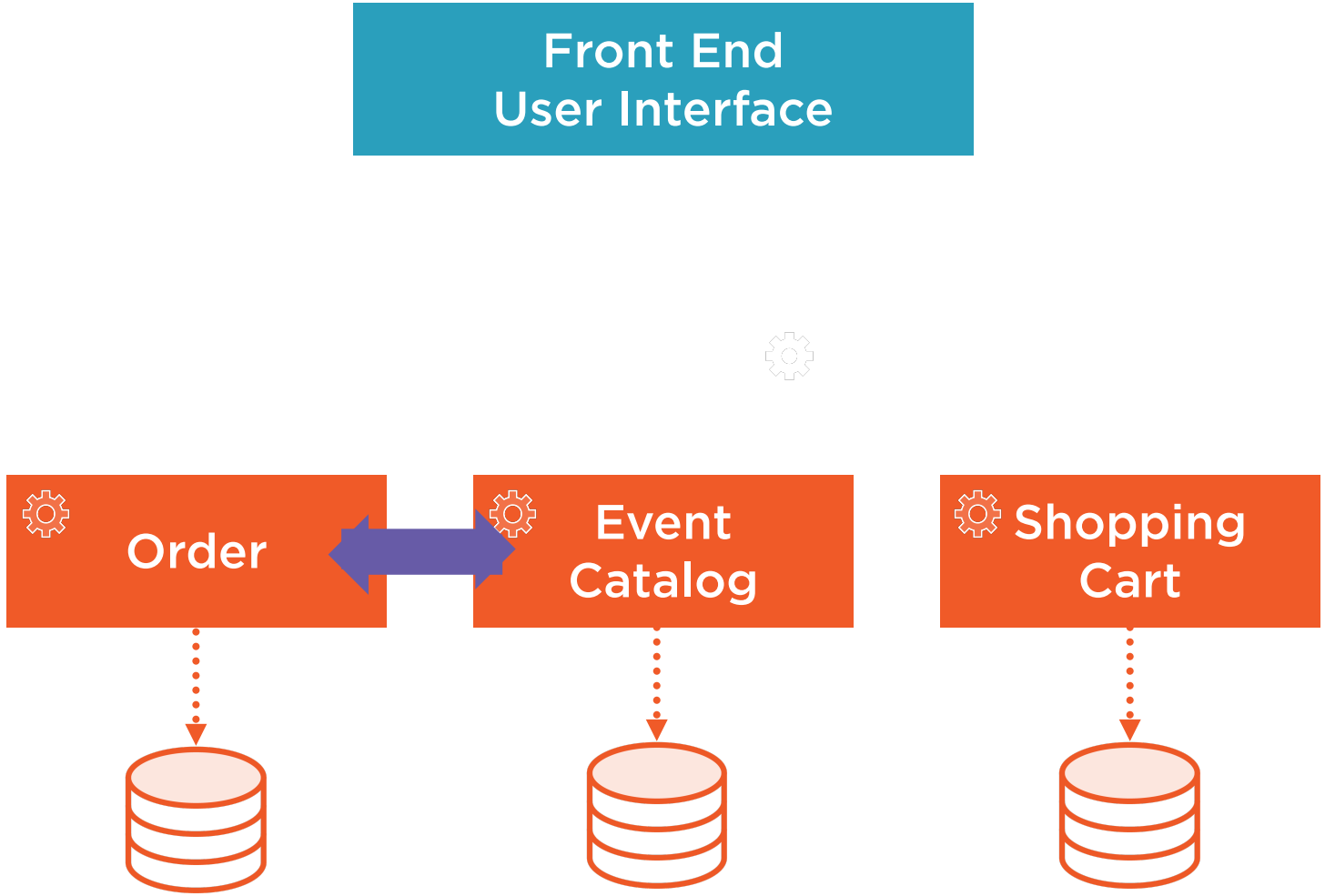
Independent Scalability

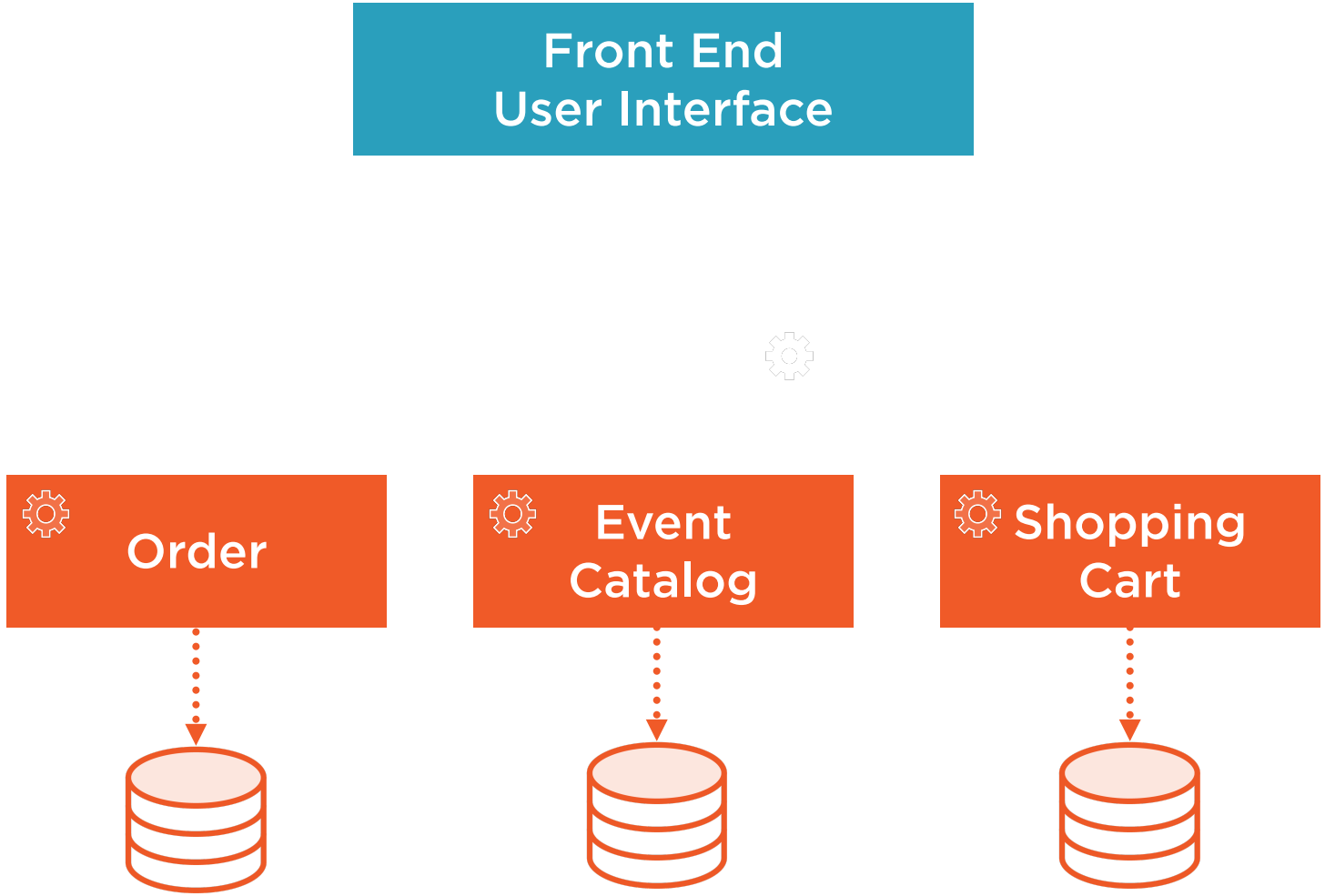
Release Cadence

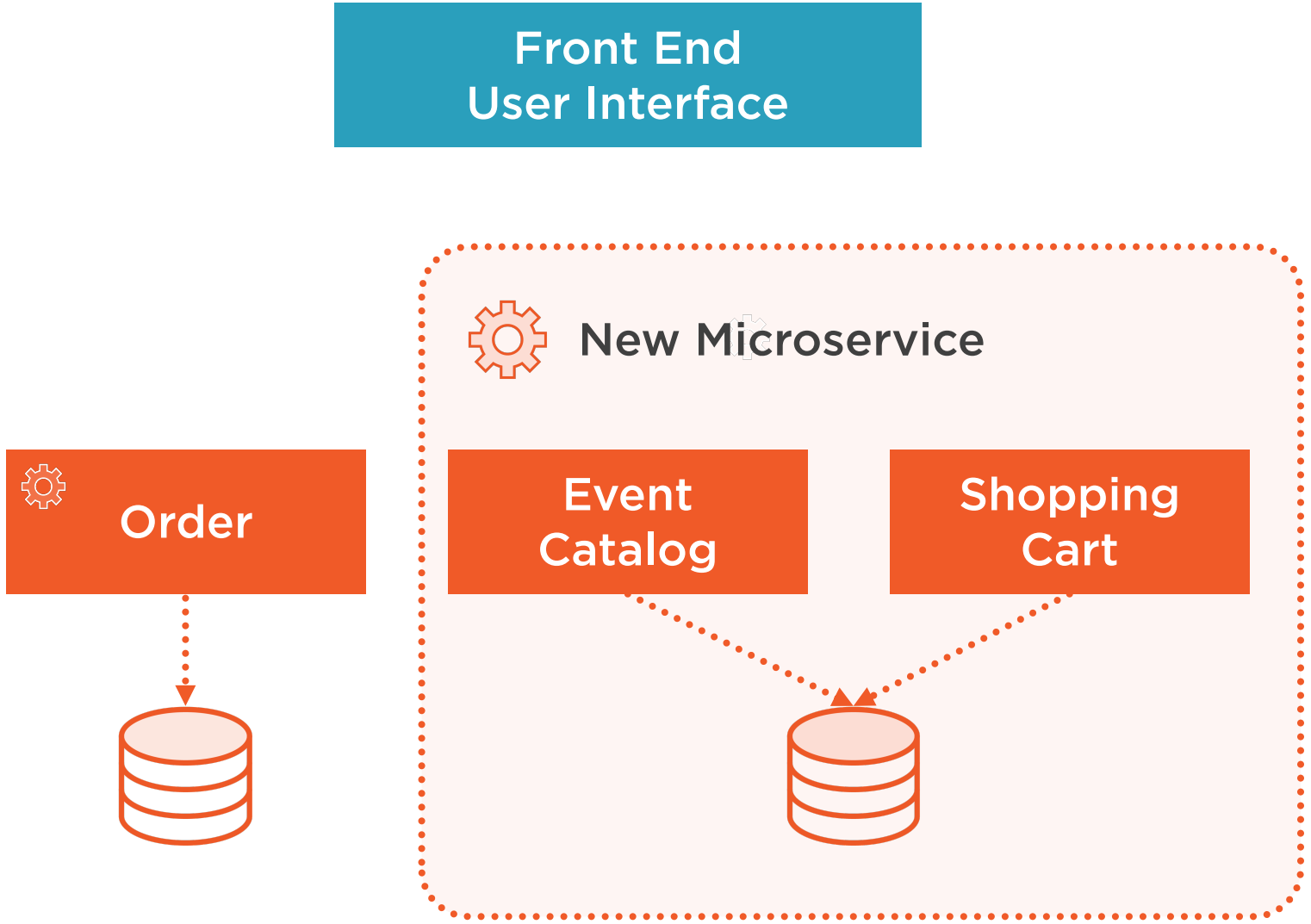
Implementation Technologies

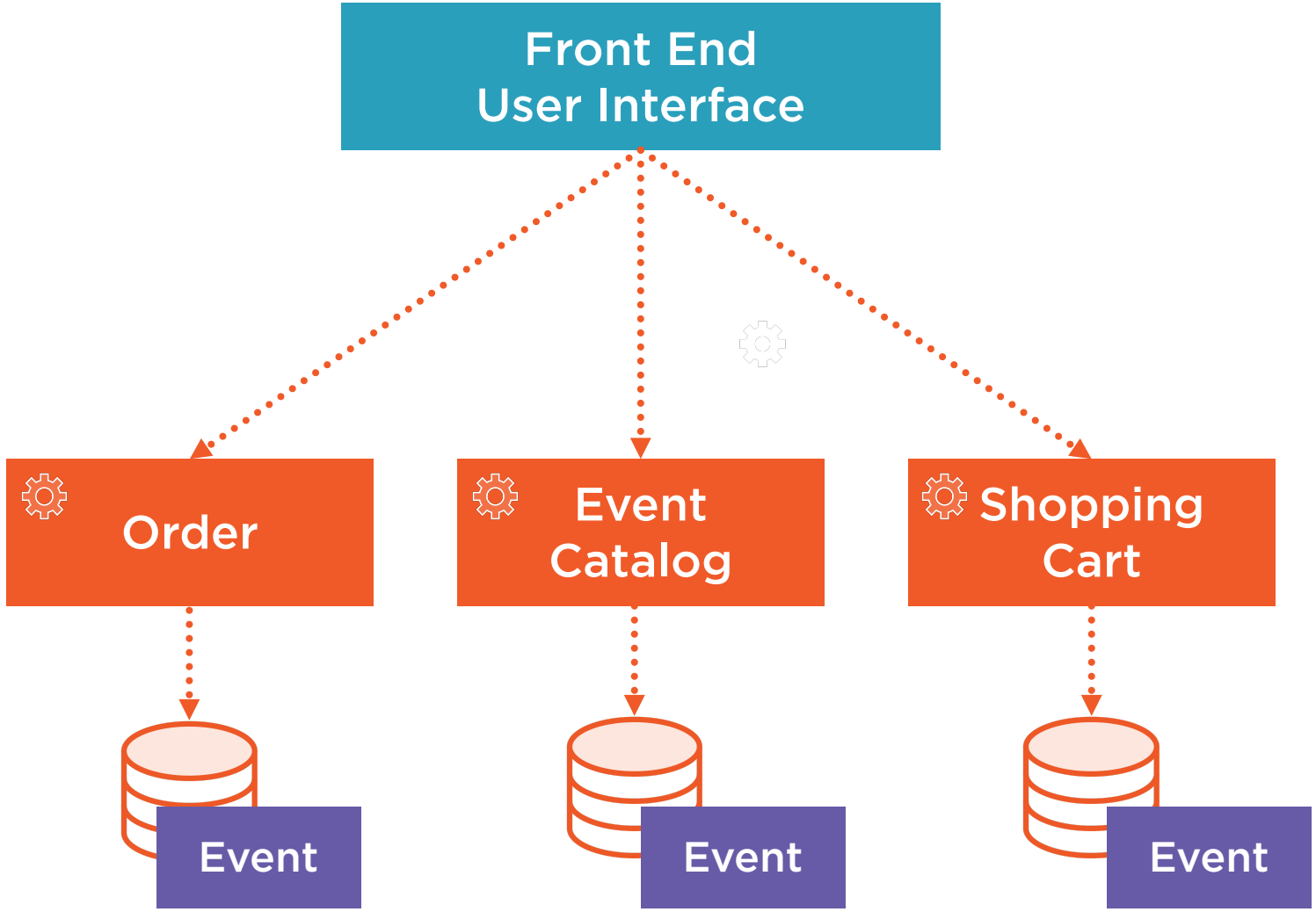
Data Ownership

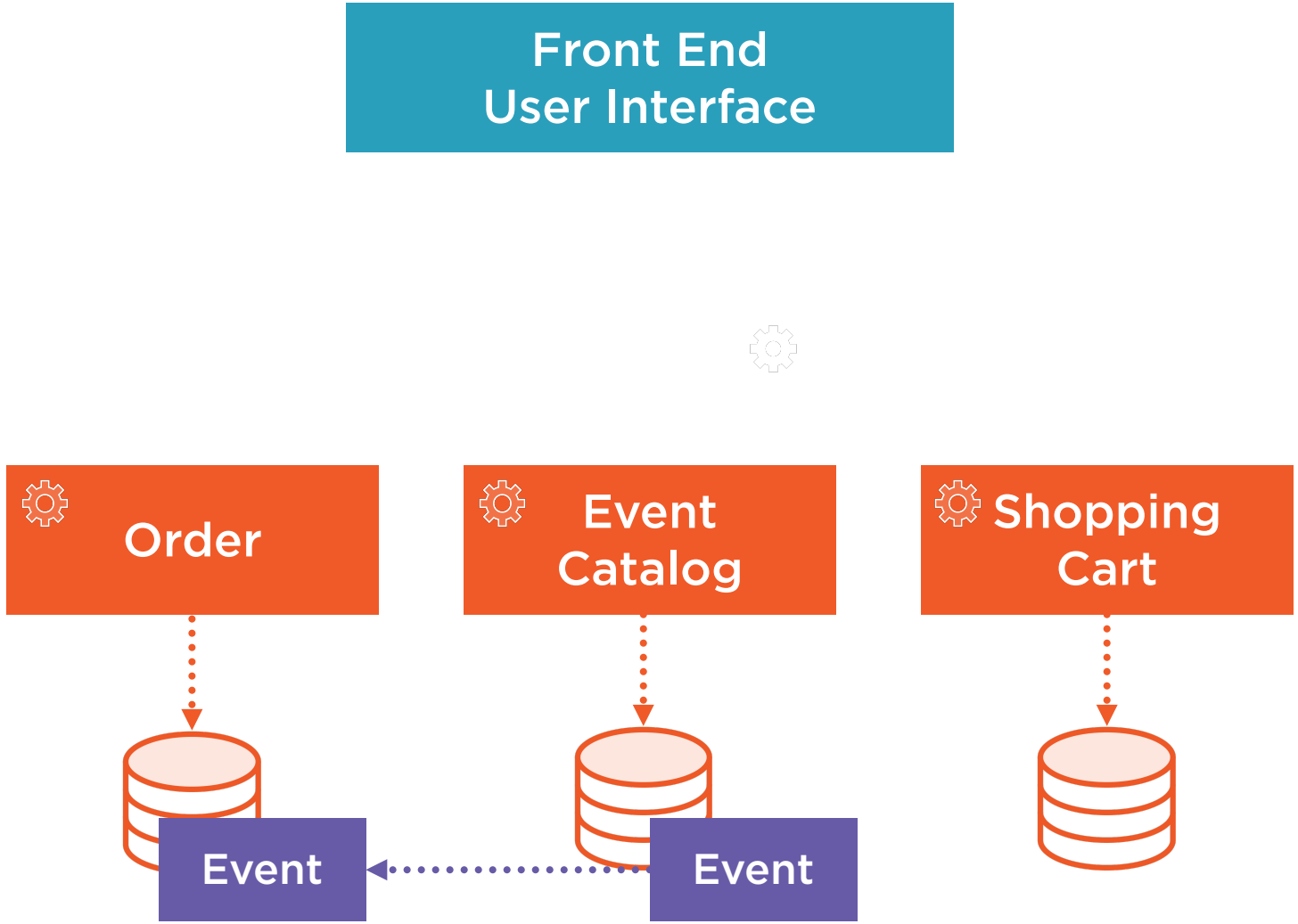

Front End
User Interface
Order
Event
Catalog
Shopping
Cart

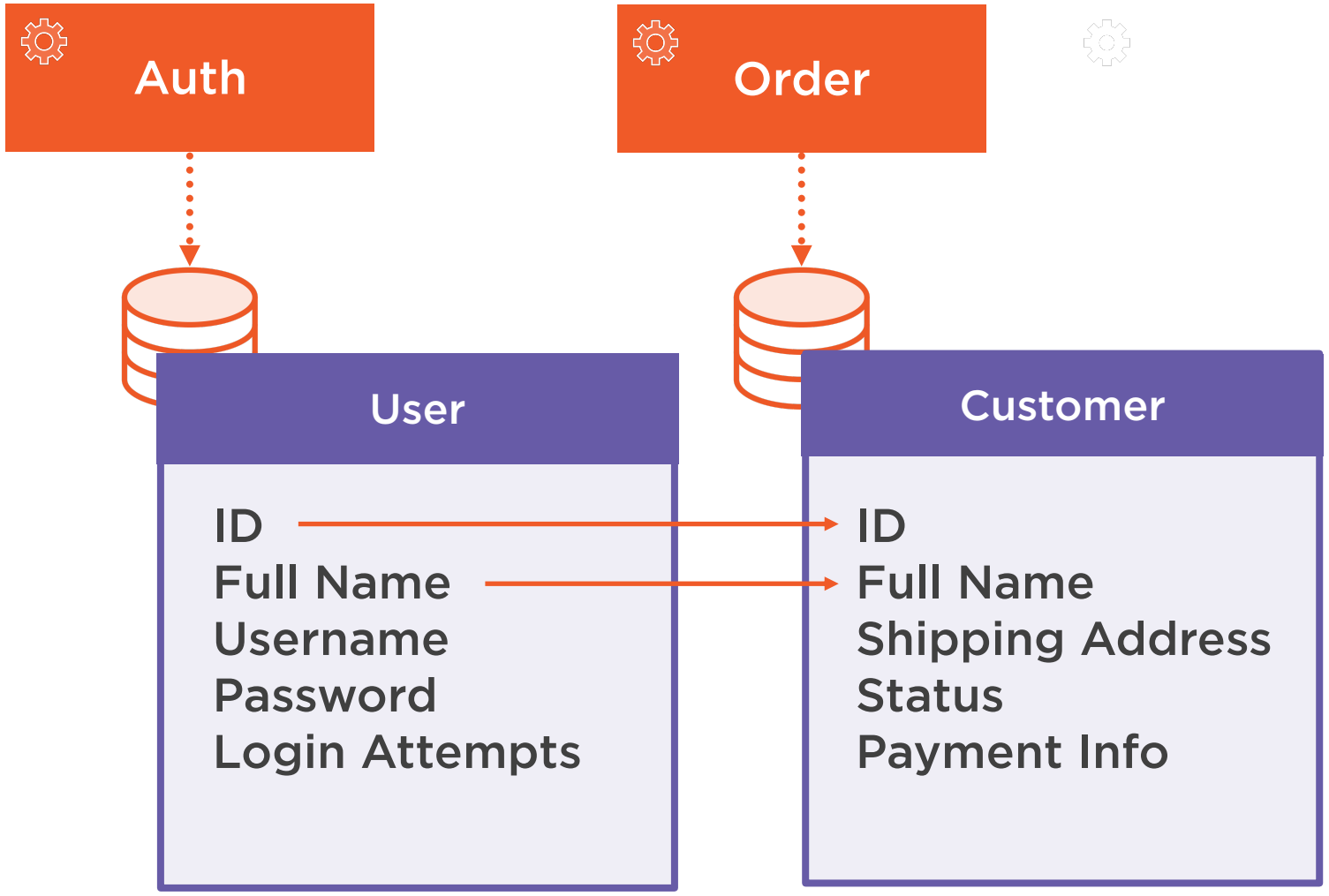

Front End
User Interface
Order
Event
Catalog
Shopping
Cart


Front End
User Interface
Order
Event
Catalog
Shopping
Cart


Front End
User Interface
New Microservice
Order
Event
Catalog
Shopping
Cart

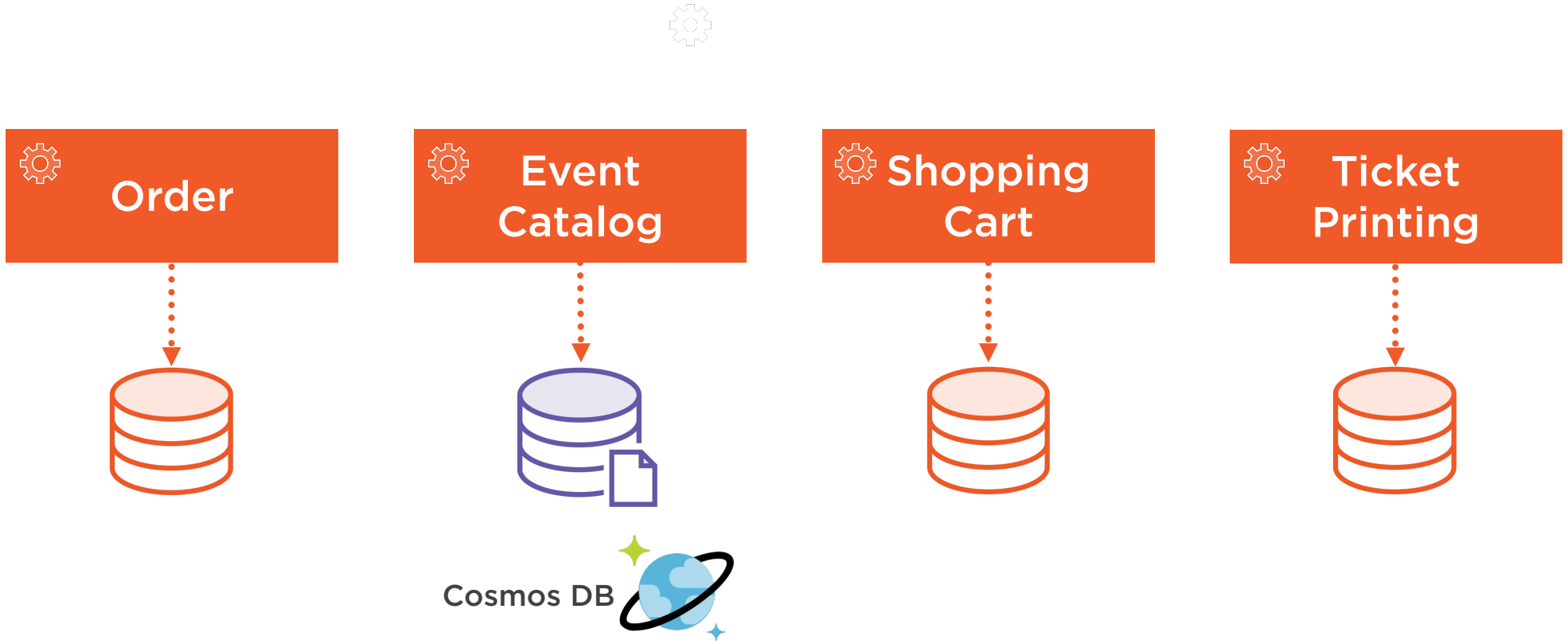

Front End
User Interface
Order
Event
Catalog
Shopping
Cart
Event
Event
Event

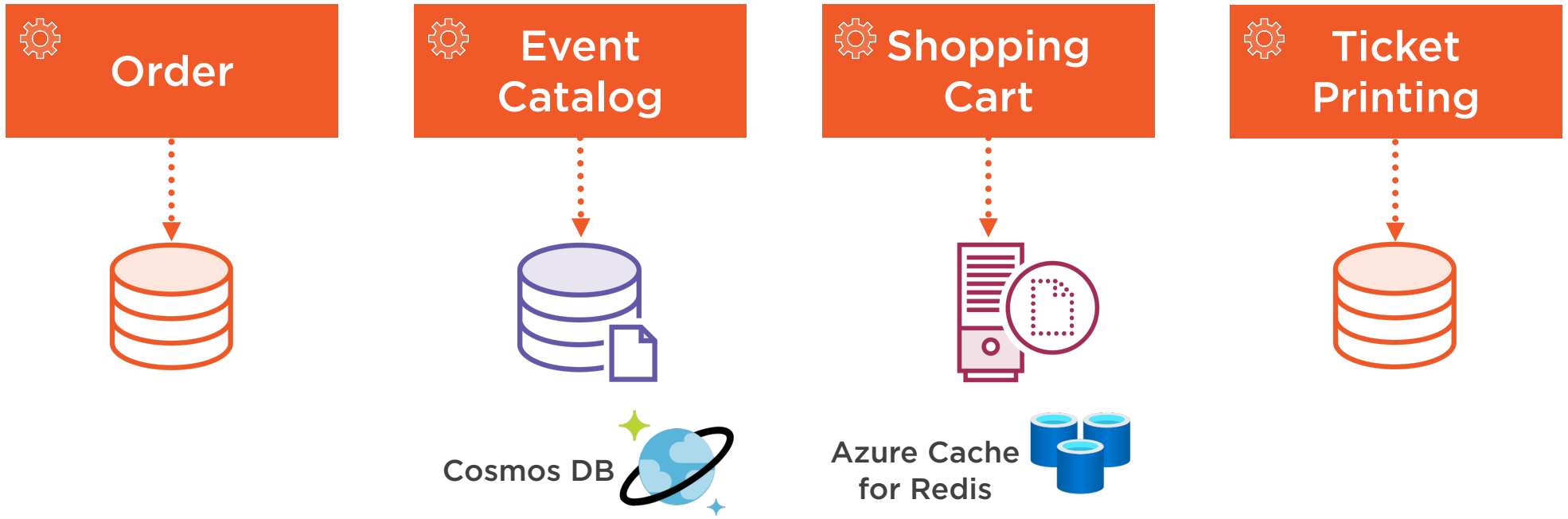

Front End
User Interface
Order
Event
Catalog
Shopping
Cart
Event
Event

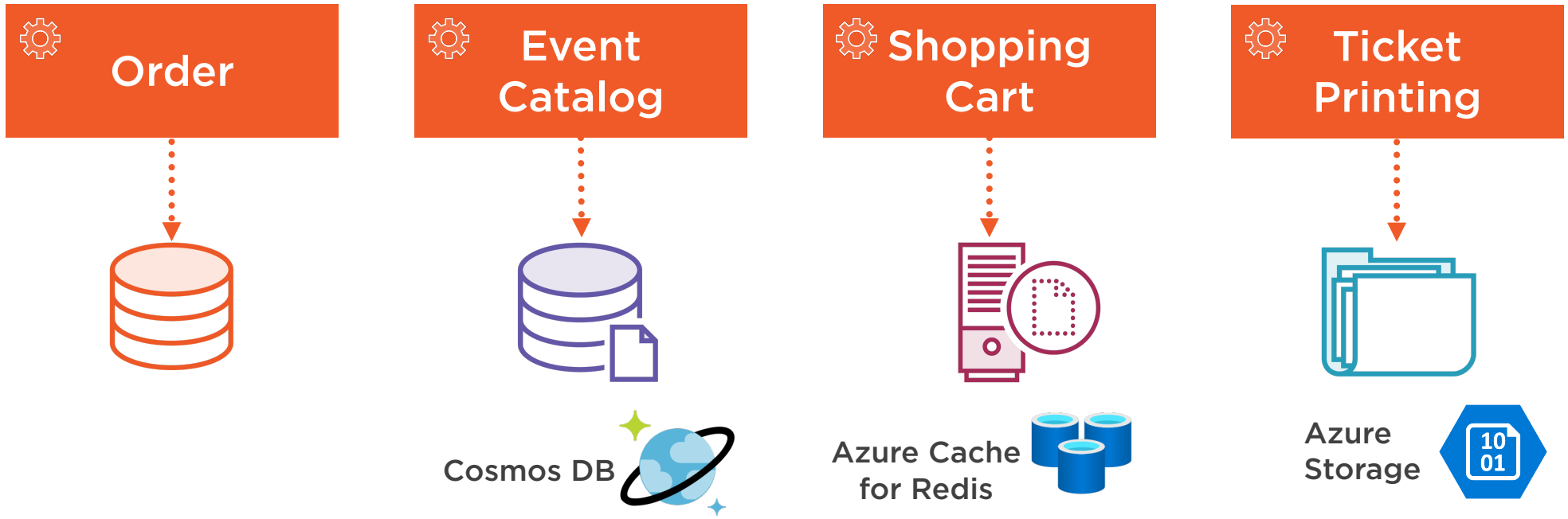

Auth
Order
User
ID
Full Name
Username
Password
Login Attempts
Customer
ID
Full Name
Shipping Address
Status
Payment Info

Polygot Persistence

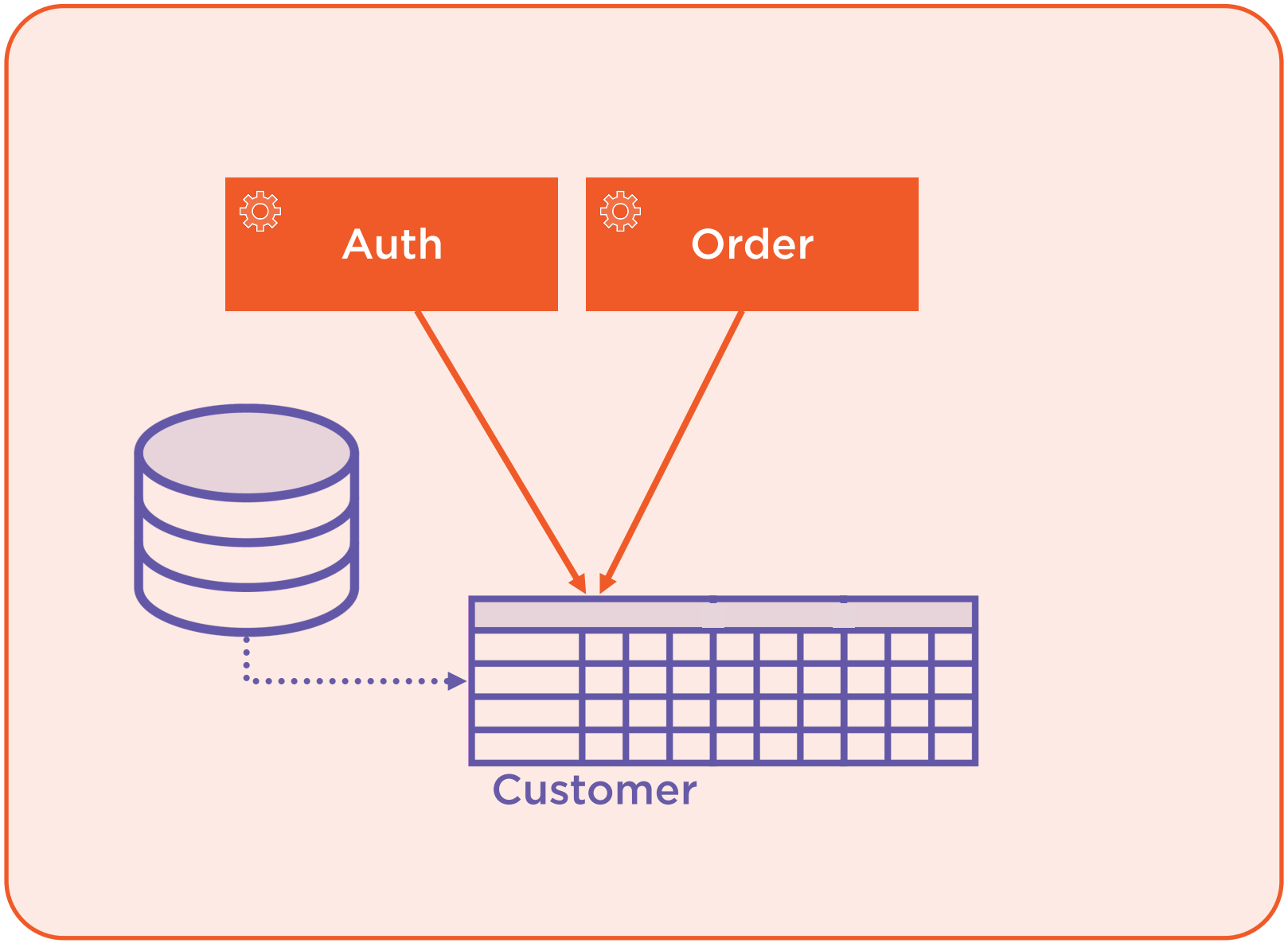

Order
Event Catalog
Shopping Cart
Ticket Printing

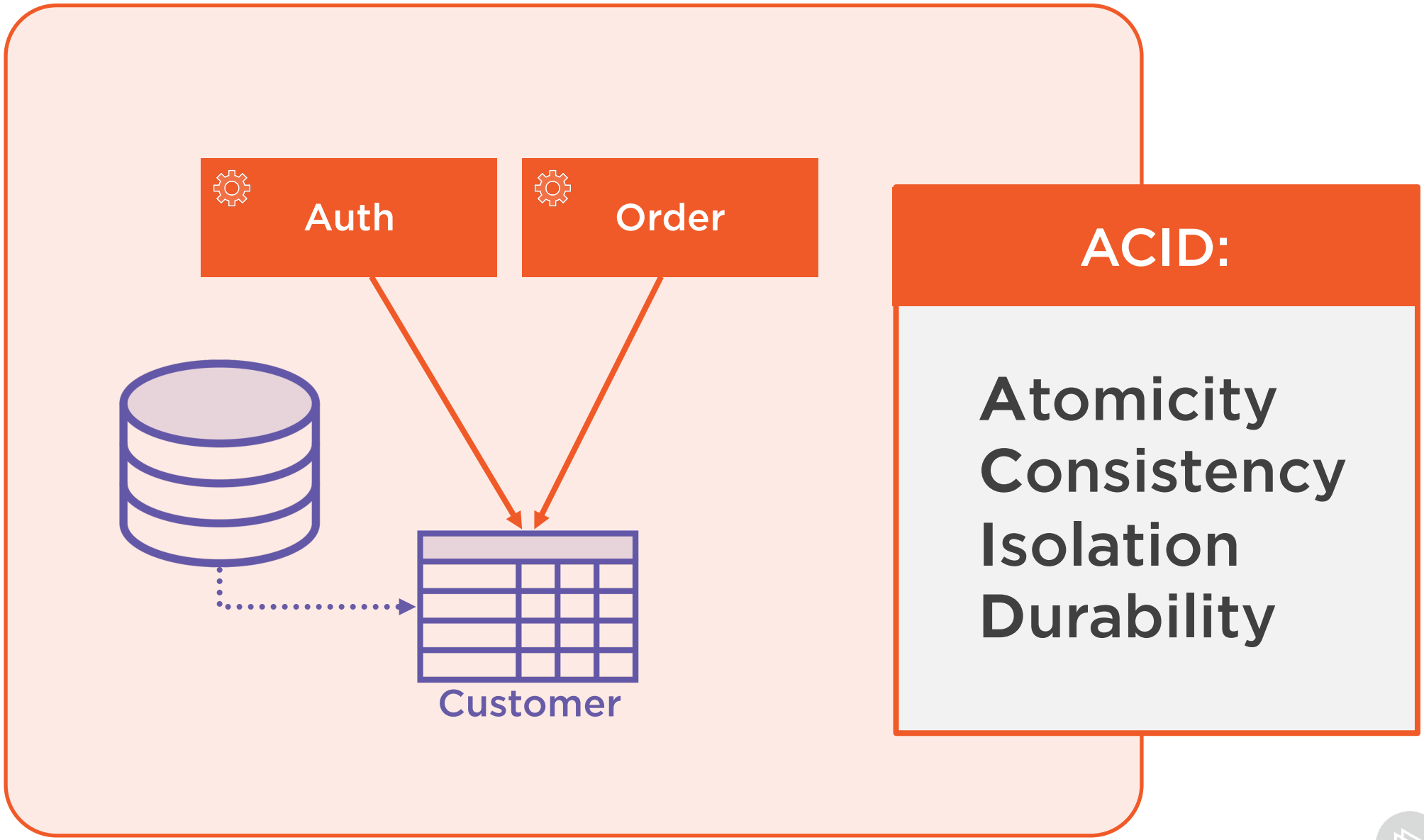

Order
Event Catalog
Shopping Cart
Ticket Printing
Cosmos DB

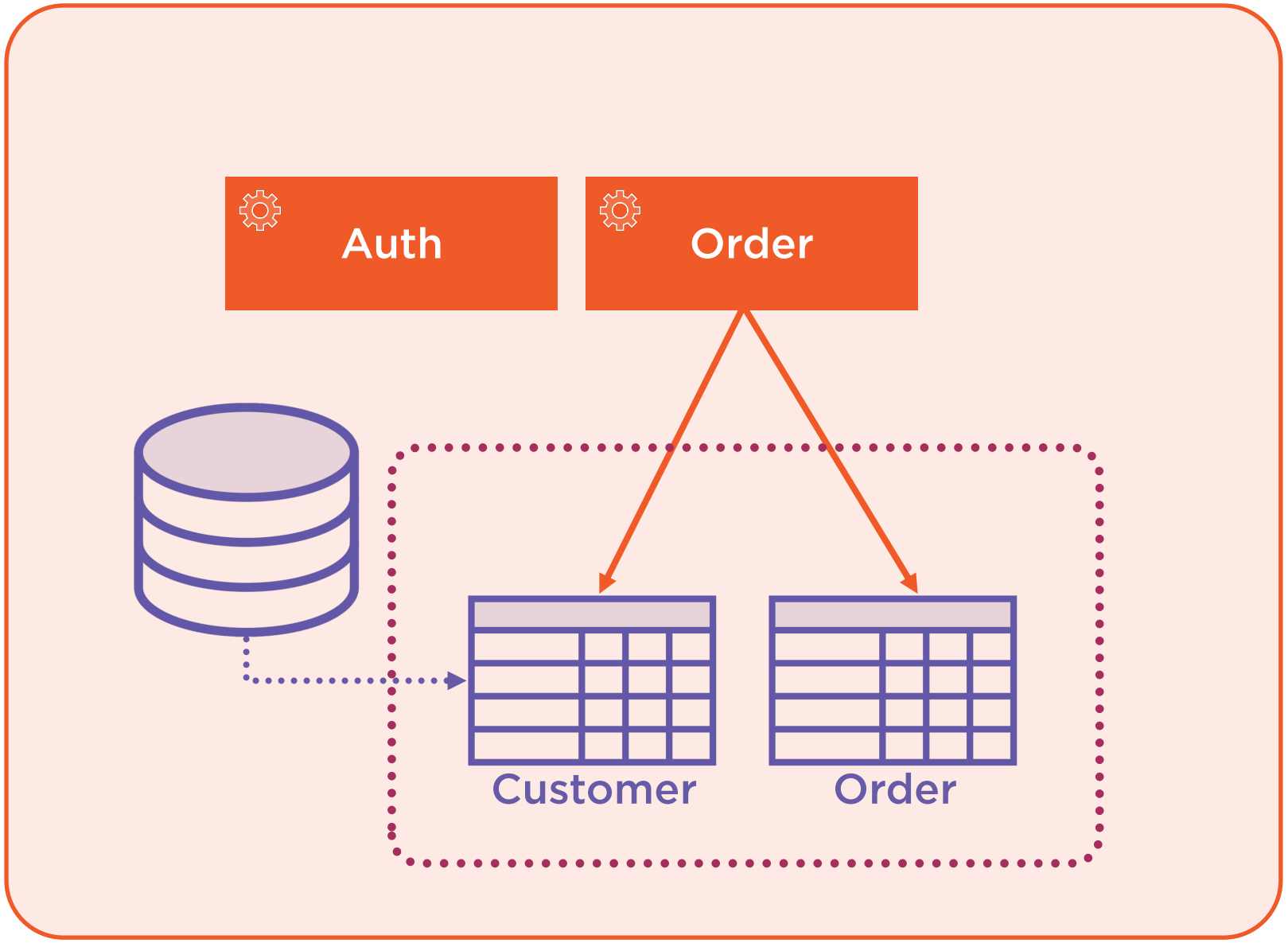

Order
Event Catalog
Shopping Cart
Ticket Printing
Cosmos DB
Azure Cache for Redis

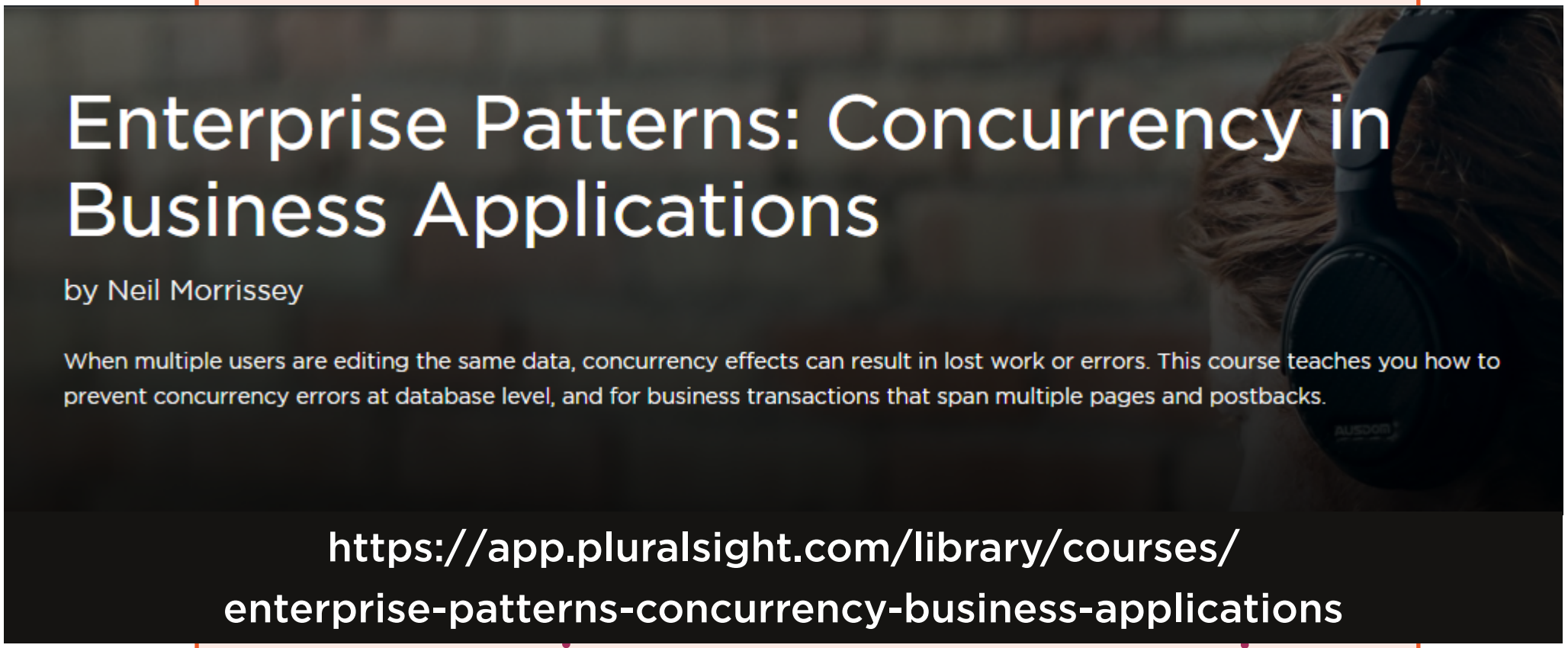

Order
Event Catalog
Shopping Cart
Ticket Printing
Cosmos DB
Azure Cache for Redis
Azure Storage
10
01

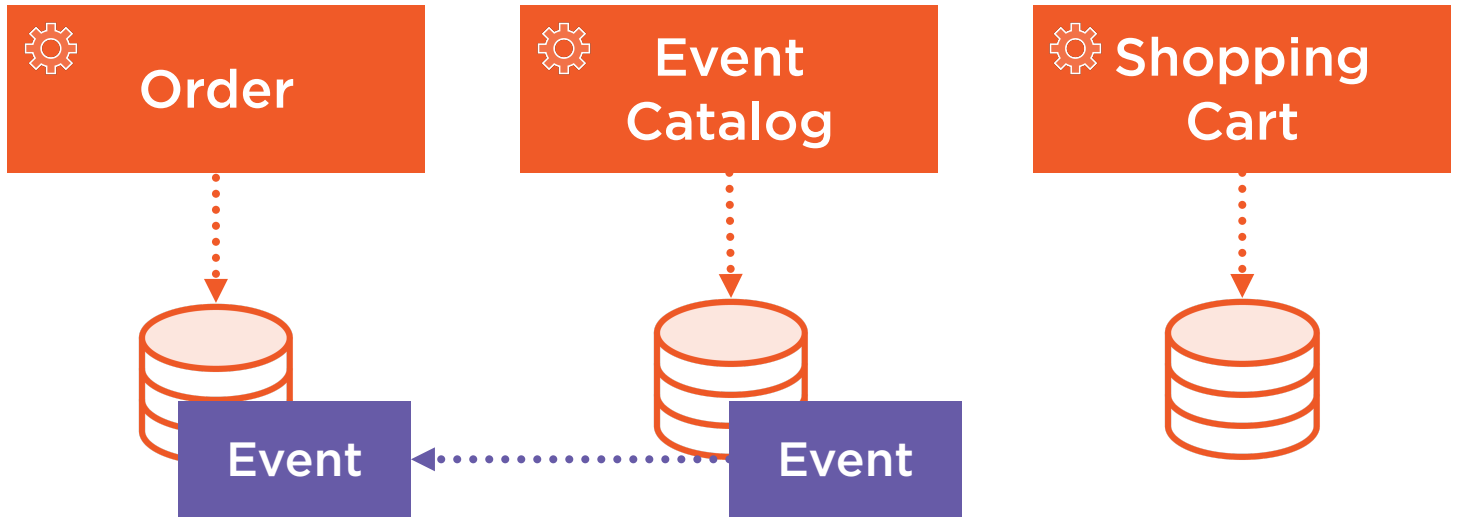
Eventual Consistency


Auth
Order
Customer


Auth
Order
Customer
ACID:
Atomicity
Consistency
Isolation
Durability


Auth
Order
Customer
Order


Enterprise Patterns: Concurrency in Business Applications
by Neil Morrissey
When multiple users are editing the same data, concurrency effects can result in lost work or errors. This course teaches you how to prevent concurrency errors at database level, and for business transactions that span multiple pages and postbacks.
https://app.pluralsight.com/library/courses/enterprise-patterns-concurrency-business-applications


Order
Event Catalog
Shopping Cart
Event
Event

CAP Theorem

Consistency

Every read gets updated data

Or an error is returned

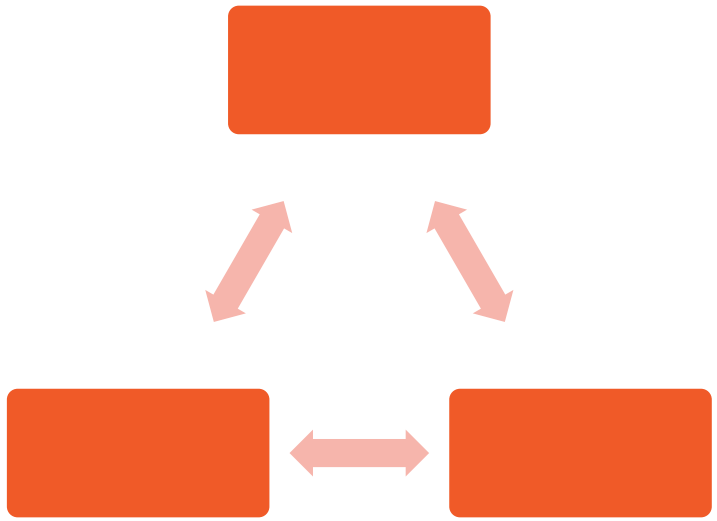
Availability

Every request gets a non-error response

Even if the data is not up to date

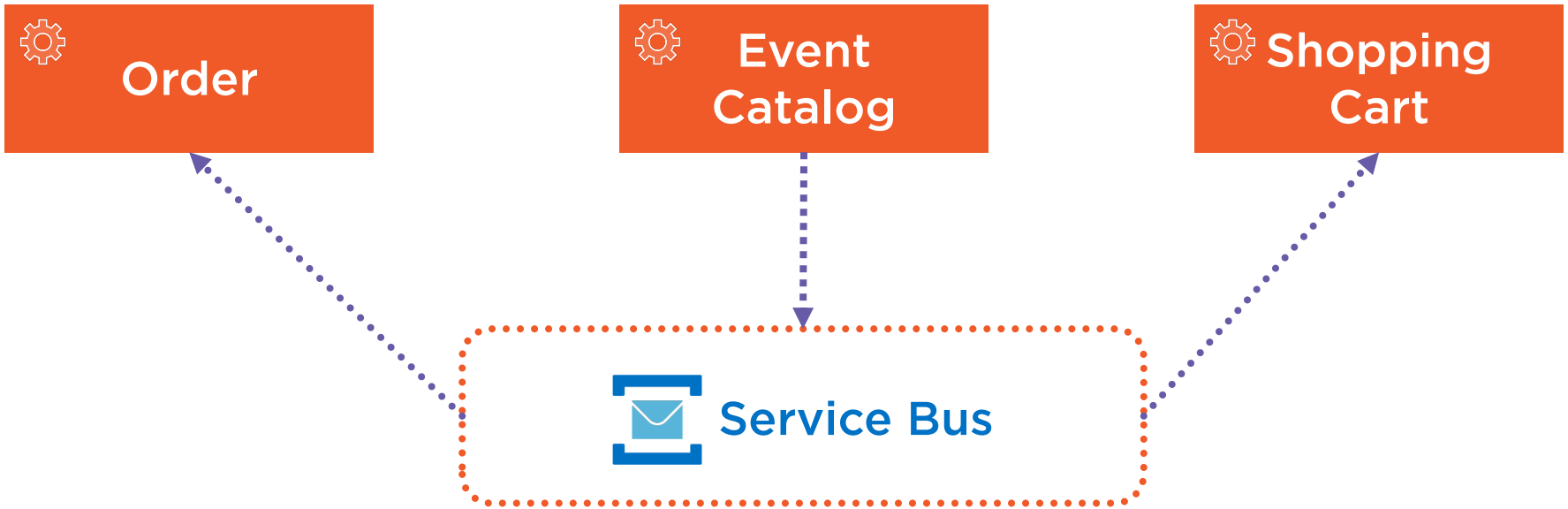
Partition Tolerance

System will continue to operate



Eventual Consistency

- Changes to data can ripple through microservices in their own time
- Microservices aren't blocked trying to update outside data
- Domain Name System is an example of eventual consistency
- Design system to accommodate eventual consistency
- Users may also need to be educated


Order
Event Catalog
Shopping Cart
Service Bus

Designing Queries with Distributed Data

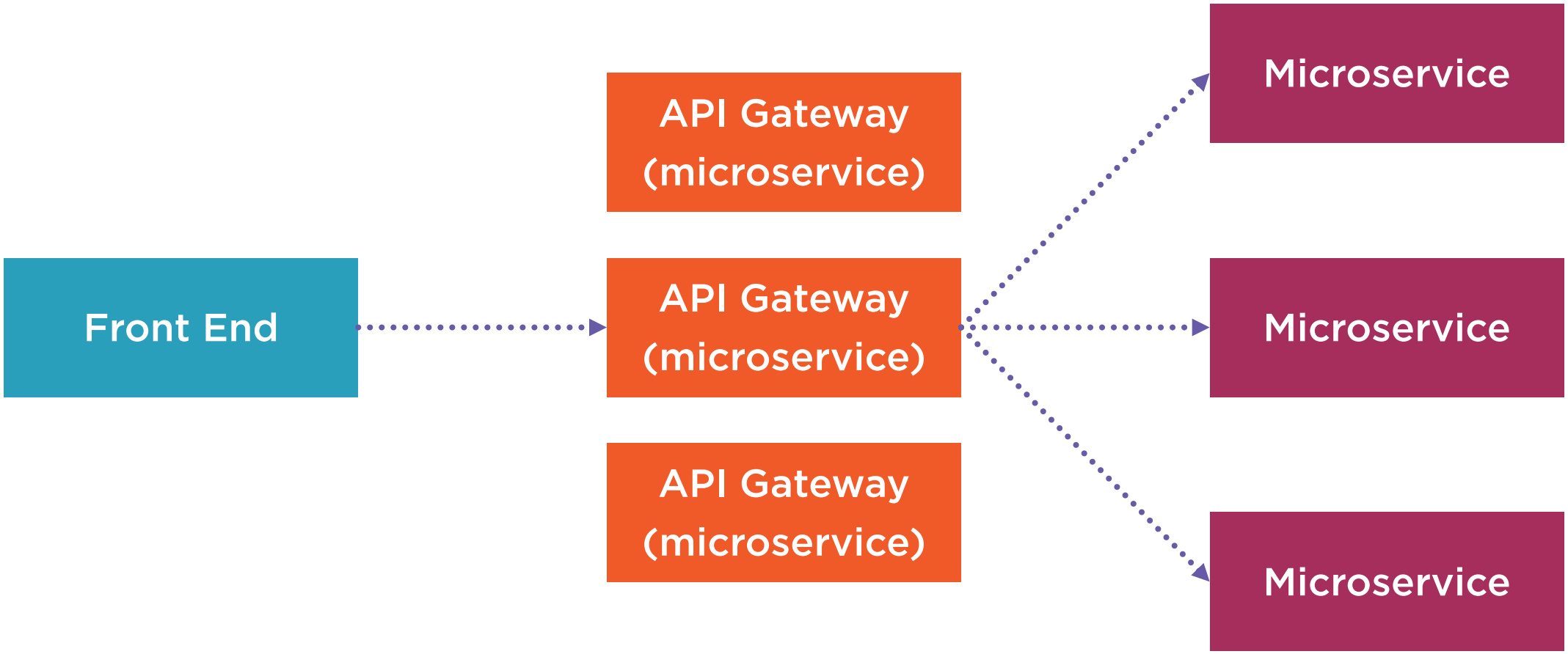
Strategies to Query Across Microservices

API Gateway

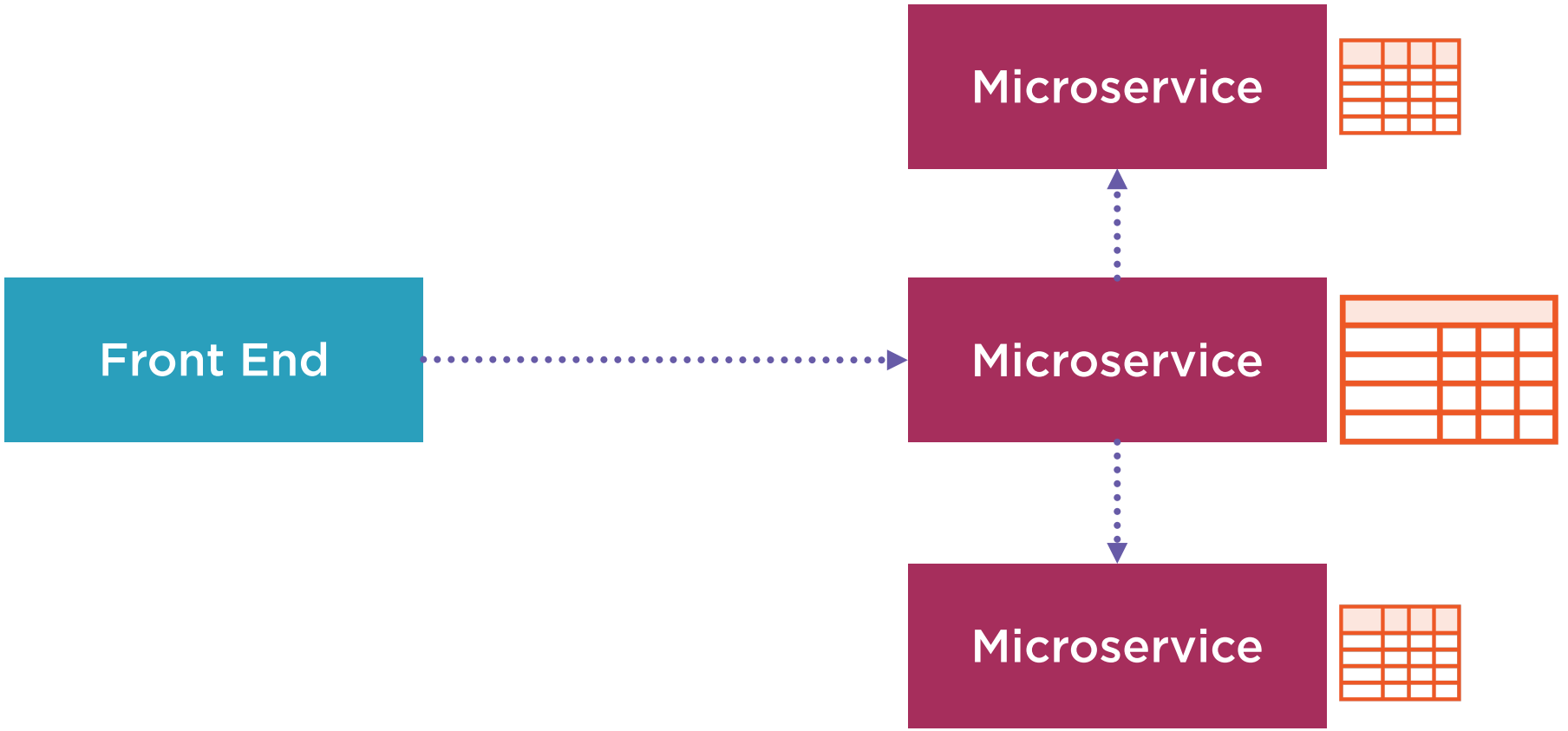
Materialized Views

Cold Data in Central Databases

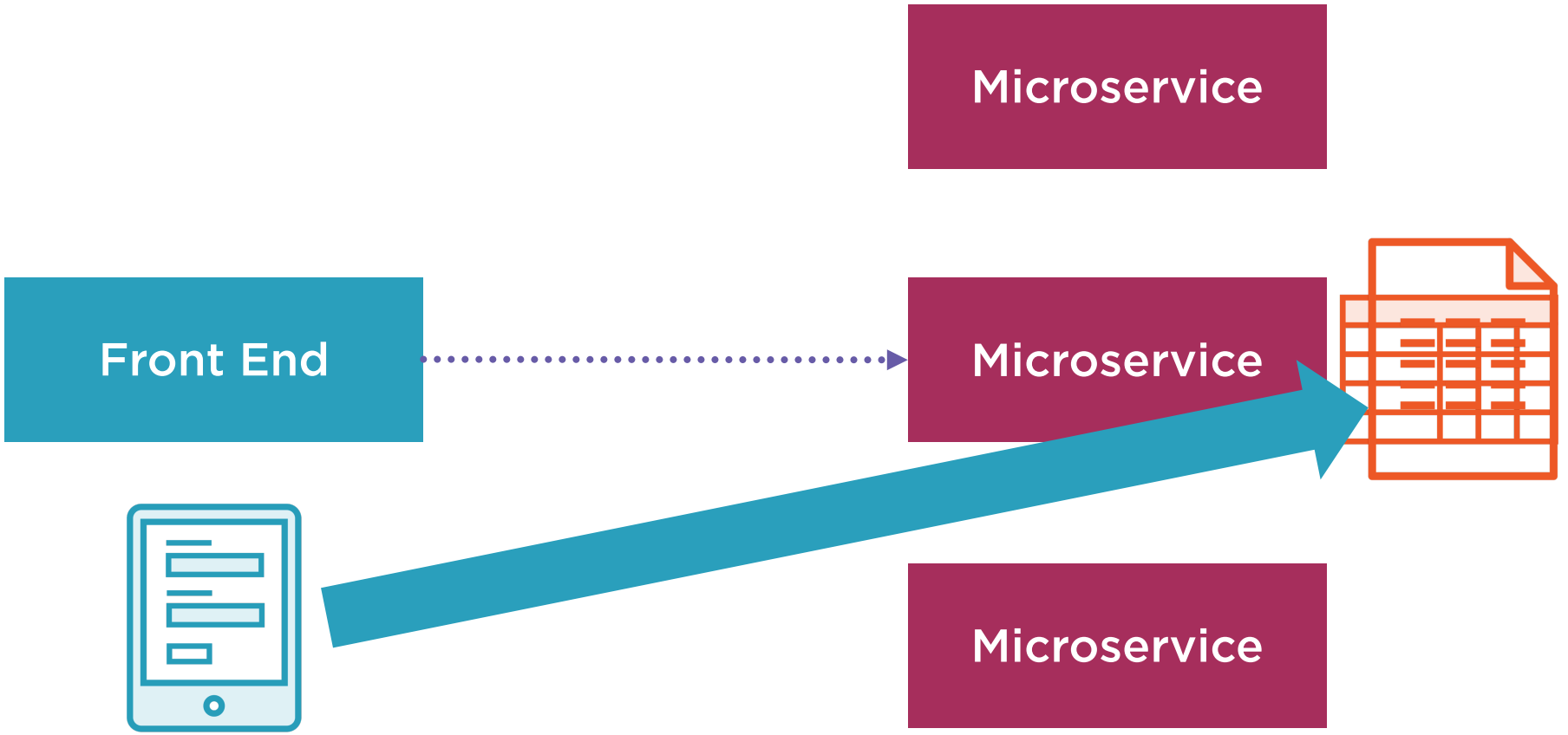
API Gateway


Front End
API Gateway (microservice)
API Gateway (microservice)
API Gateway (microservice)
Microservice
Microservice
Microservice

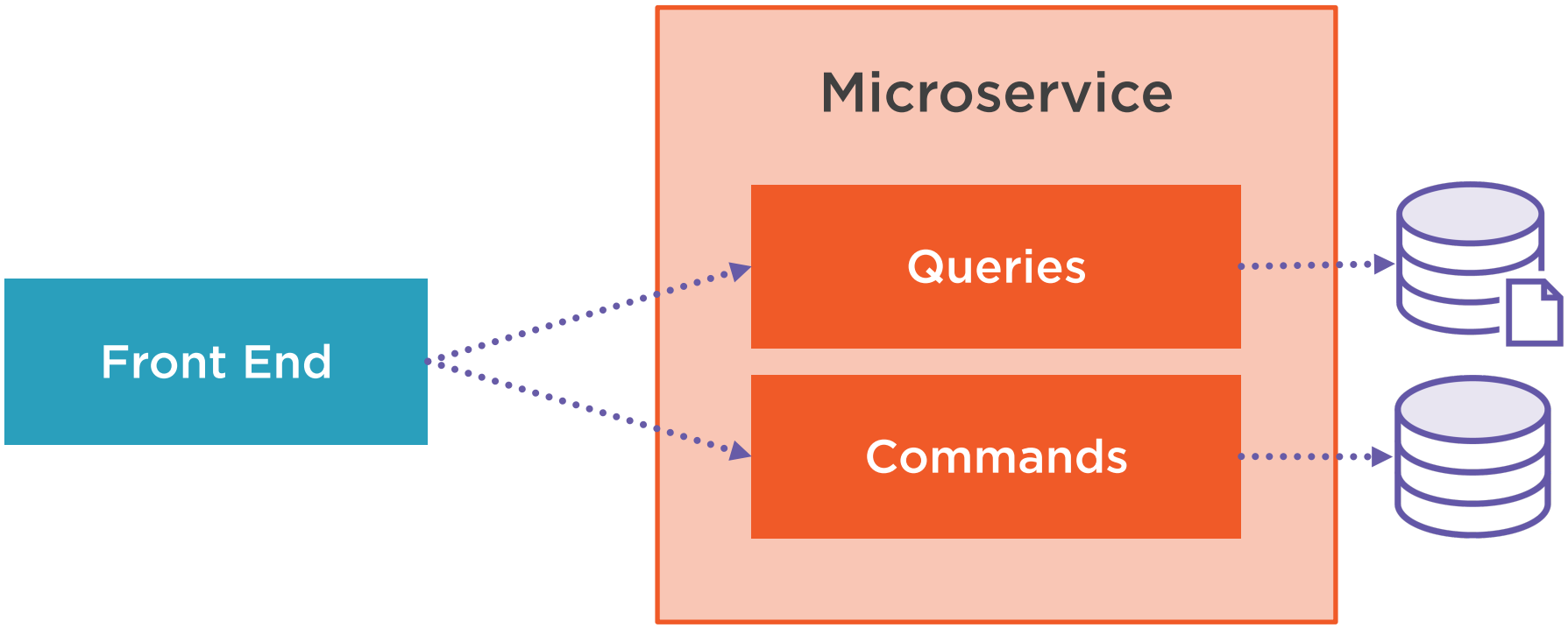
Materialized View



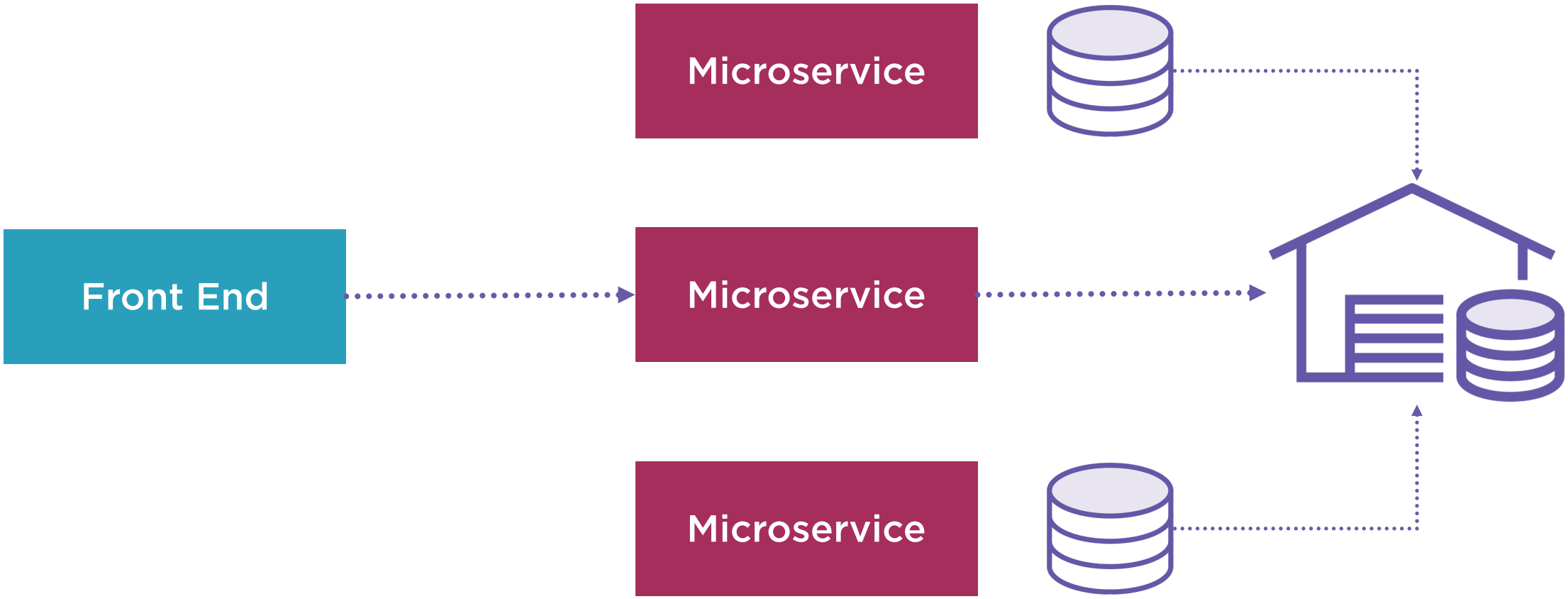
Materialized View



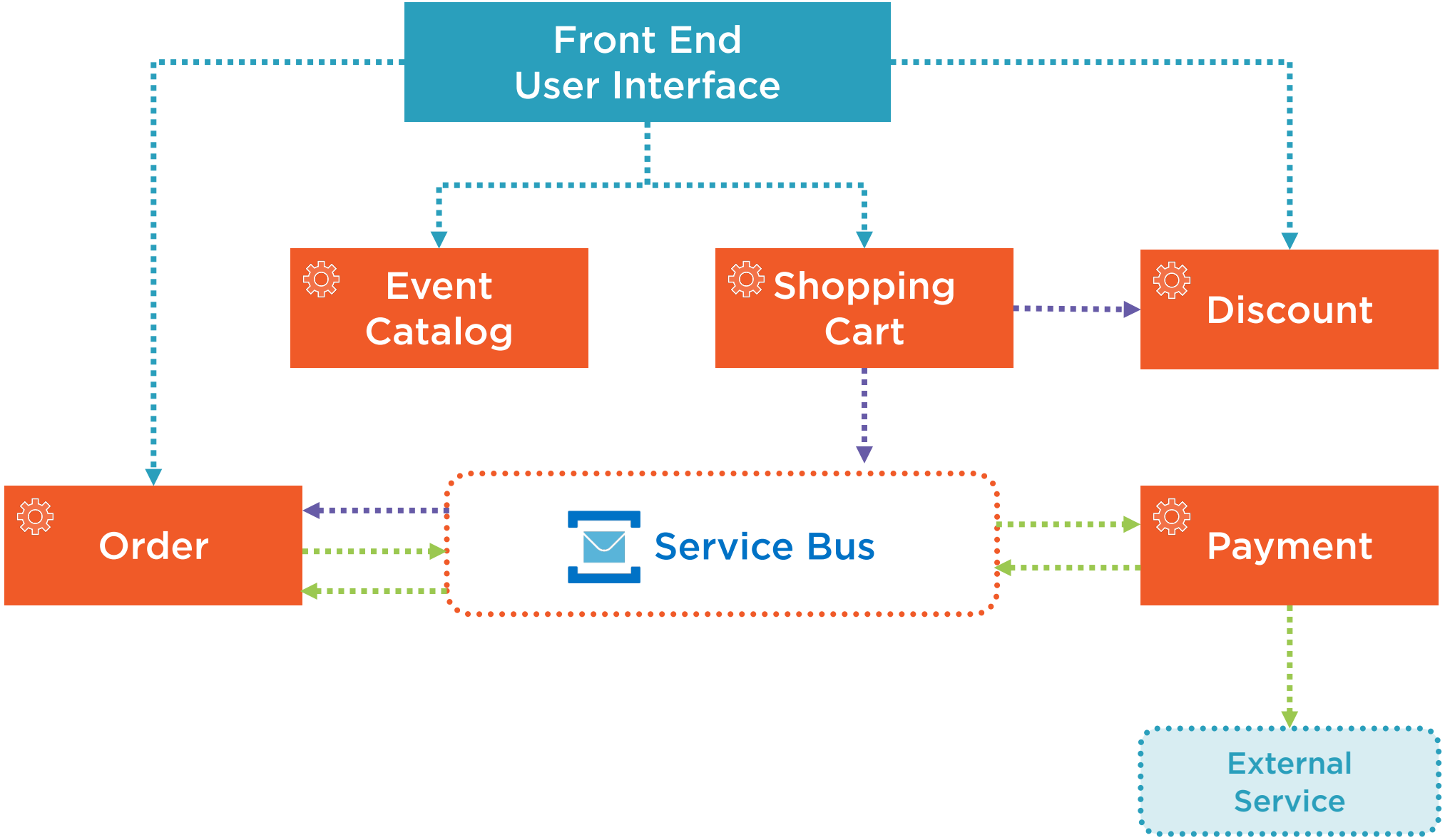
Materialized View (CQRS)

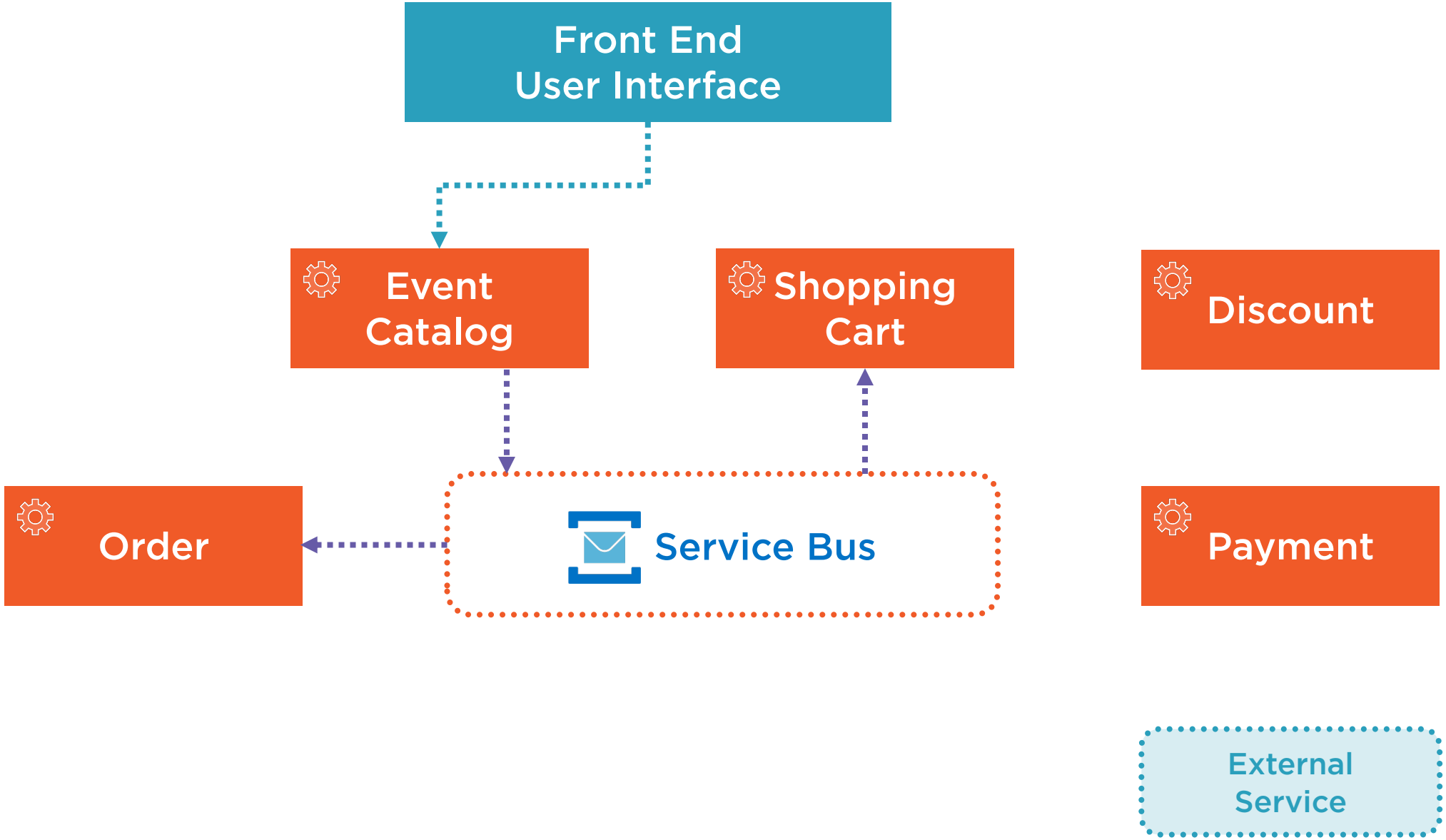


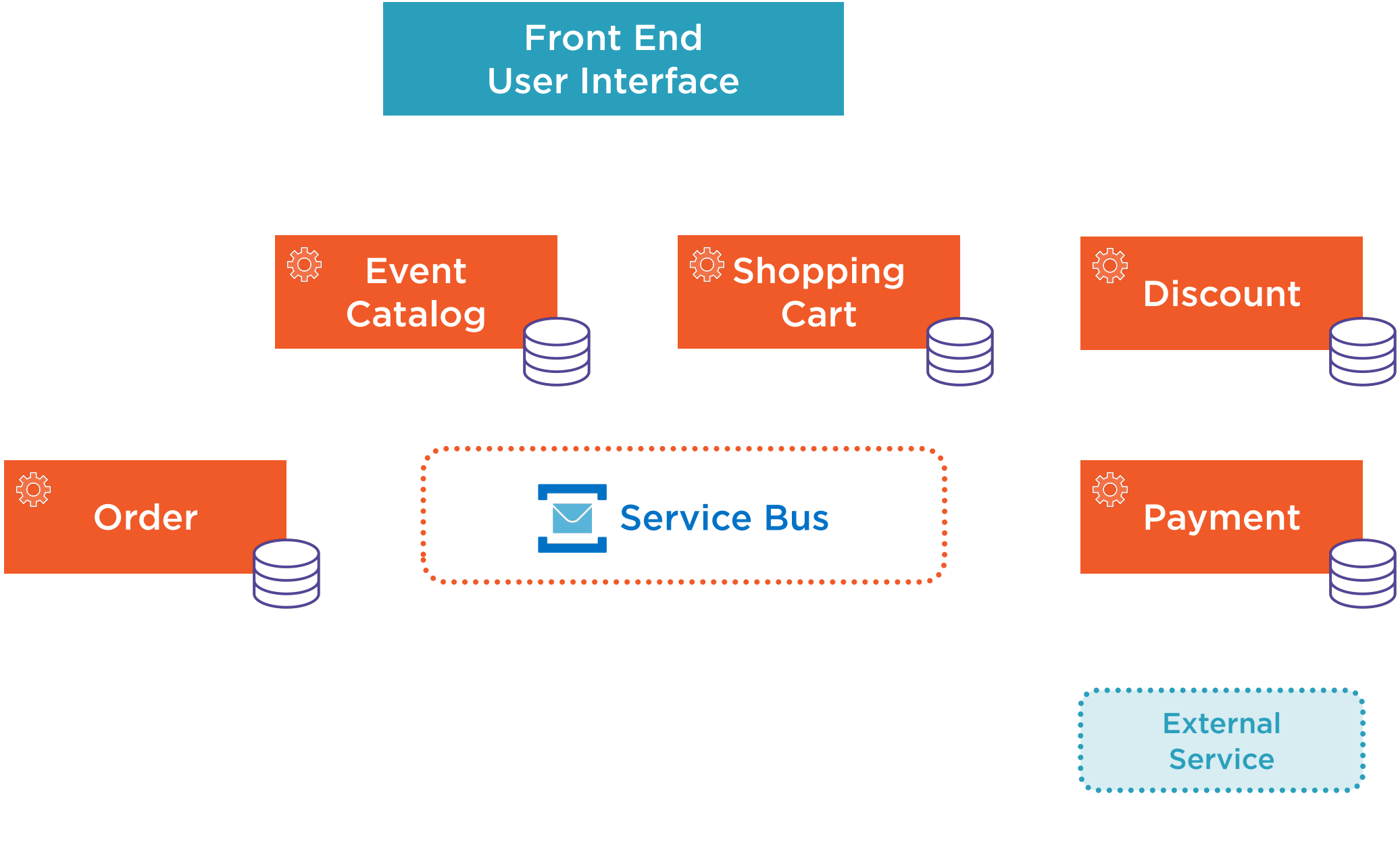
“Cold Data” in Central Databases



The Globoticket Sample Project Architecture


Front End
User Interface
Event
Catalog
Shopping
Cart
Discount
Order
Service Bus
Payment
External
Service


Front End
User Interface
Event
Catalog
Shopping
Cart
Discount
Order
Service Bus
Payment
External
Service


Front End
User Interface
Event
Catalog
Shopping
Cart
Discount
Order
Service Bus
Payment
External
Service

Front End
User Interface

Event Catalog

Cosmos DB

Shopping Cart

Azure Cache for Redis

Discount

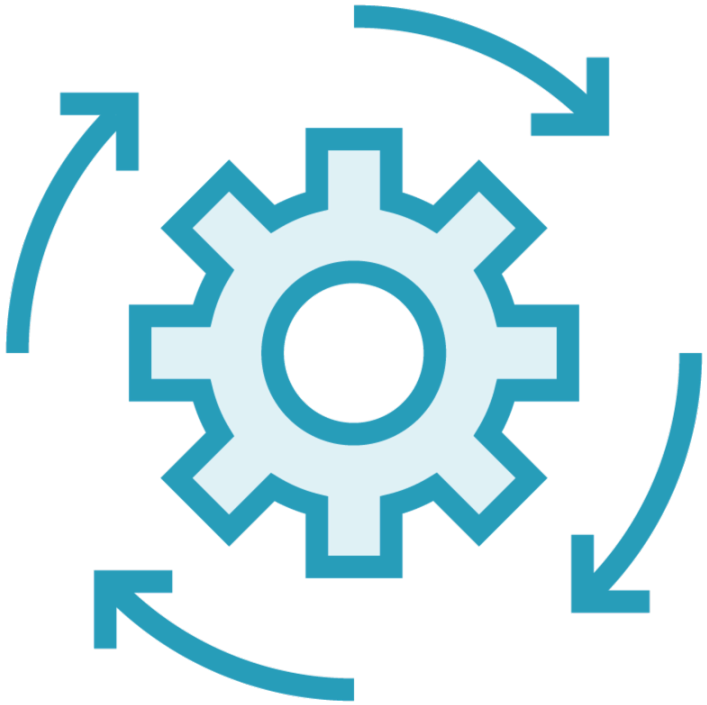
Order

Service Bus

Payment

External Service

Configuring the Sample Project



.NET Core 3.1 SDK

Entity Framework Core Tools

SQL Server Express

Visual Studio

Azure Service Bus instance

```
dotnet tool install --global dotnet-ef

dotnet tool update --global dotnet-ef

dotnet-ef
```

Entity Framework Core Tools

Command Prompt (Windows)

Module Summary

Data considerations in microservice design

Polygot persistence

Eventual consistency

Queries with distributed data

Sample project

- Demo
- Architecture
- Configuration
- Code walkthrough

Up Next:
Data Management within Microservices