Microservices

Matthew Renze

SOFTWARE CONSULTANT

@matthewrenze   www.matthewrenze.com

Overview

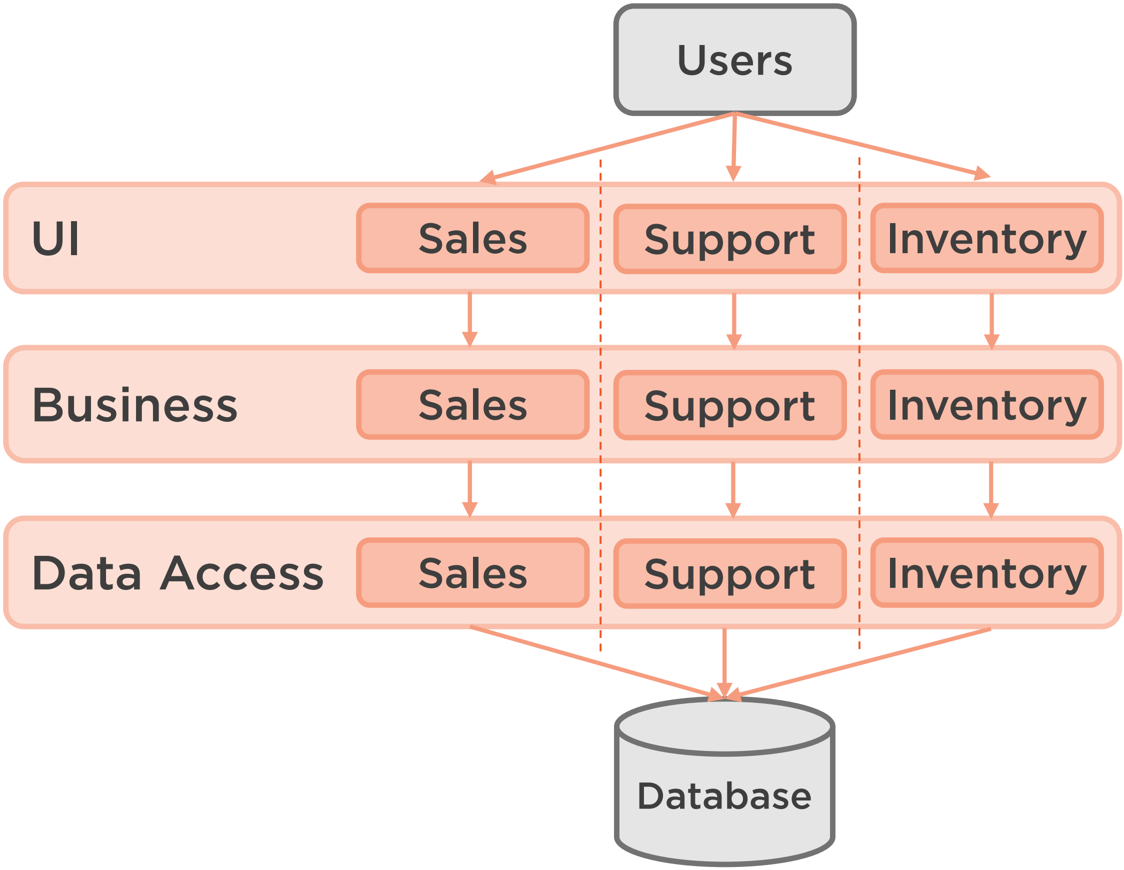
Bounded Context

Microservices

Pros and Cons

Demo

Components



Users

UI	Sales	Support	Inventory
Business	Sales	Support	Inventory
Data Access	Sales	Support	Inventory

Database

Problem Domain

Sales

Sales opportunity

Contact

Sales person

Product

Sales territory

Support

Support ticket

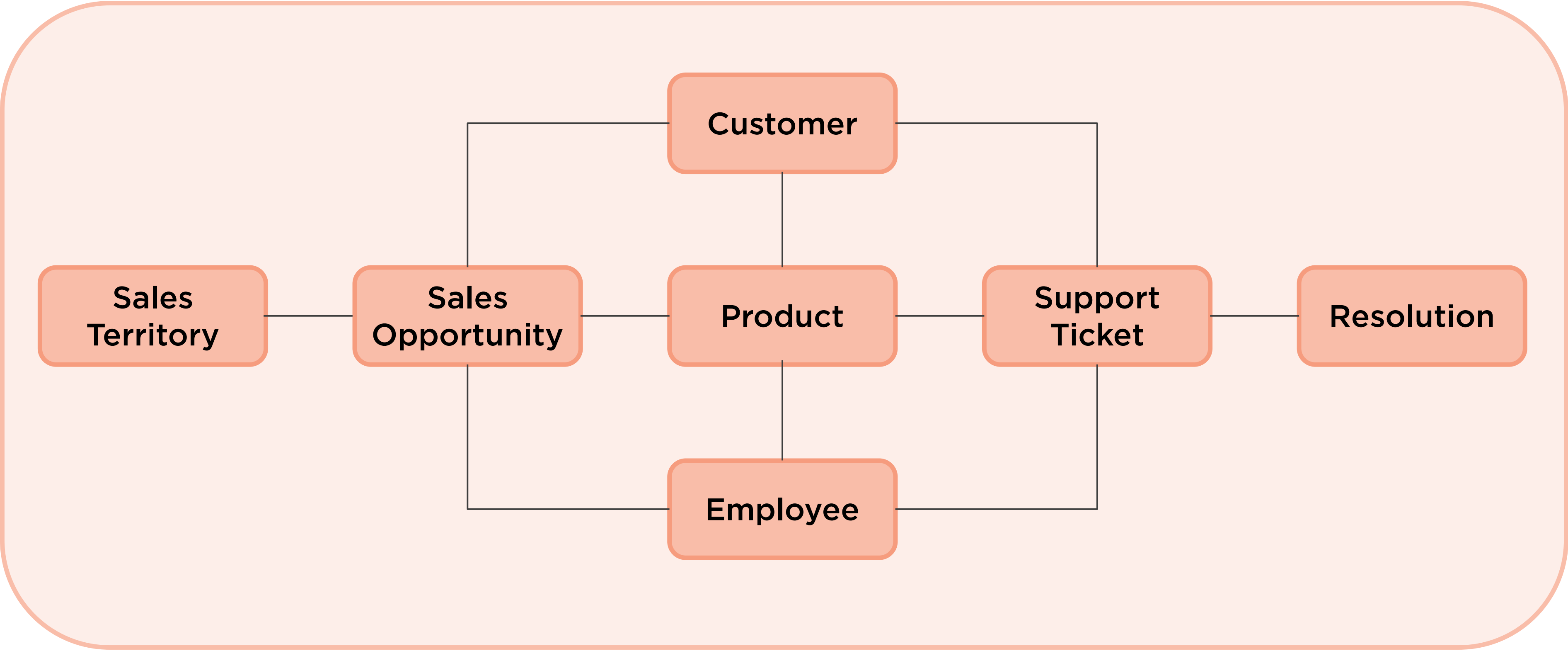
Customer

Support person

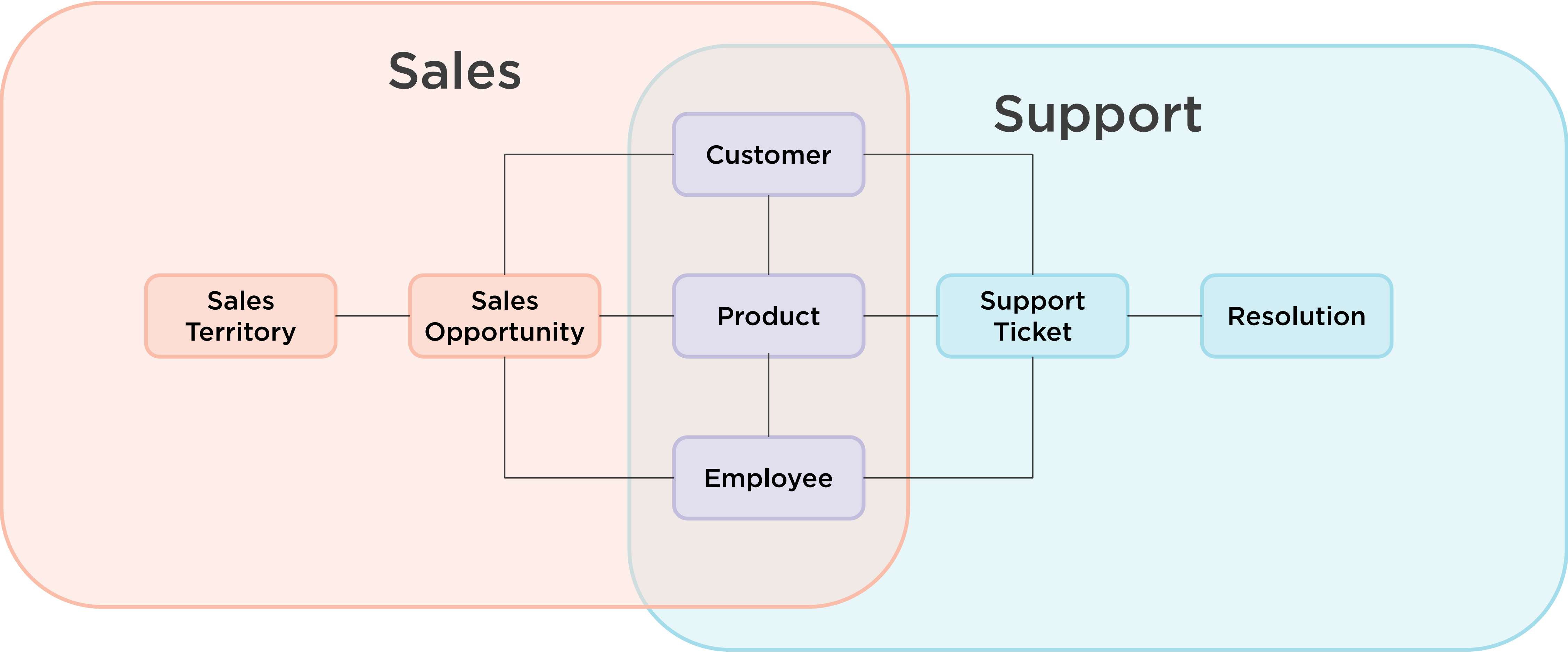
Product

Resolution

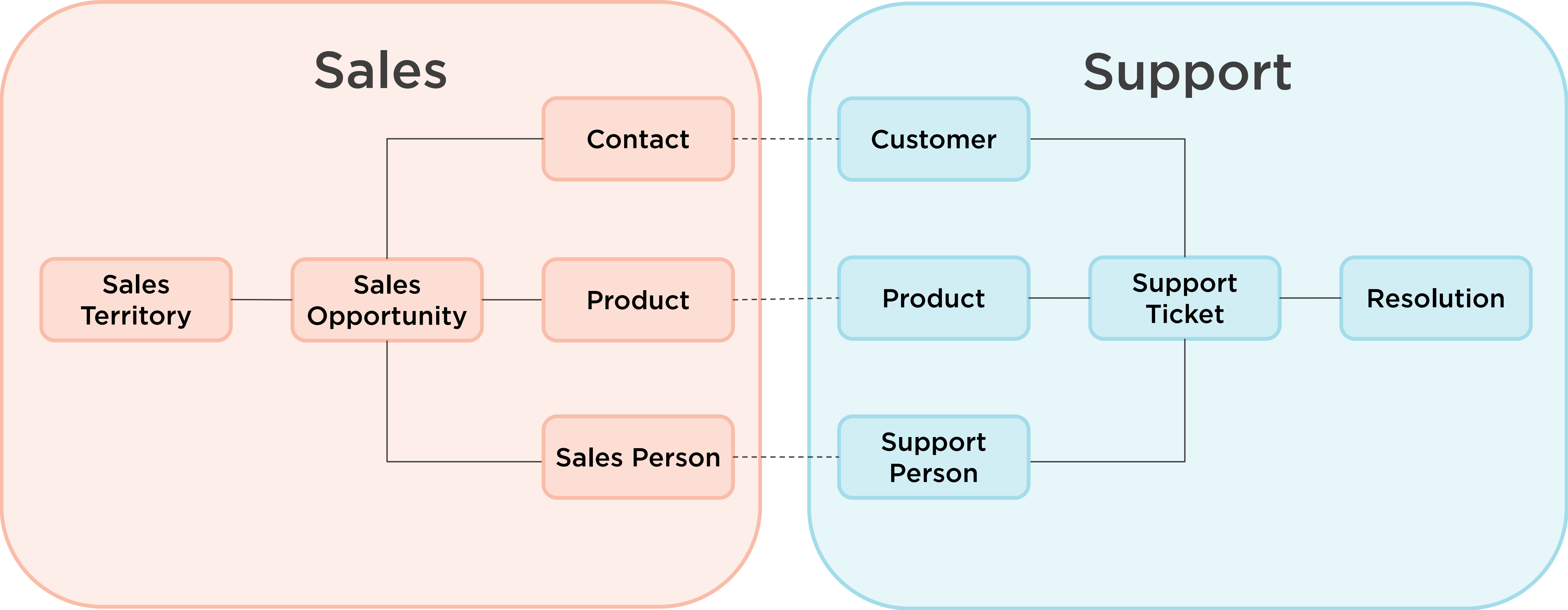
Single Domain Model



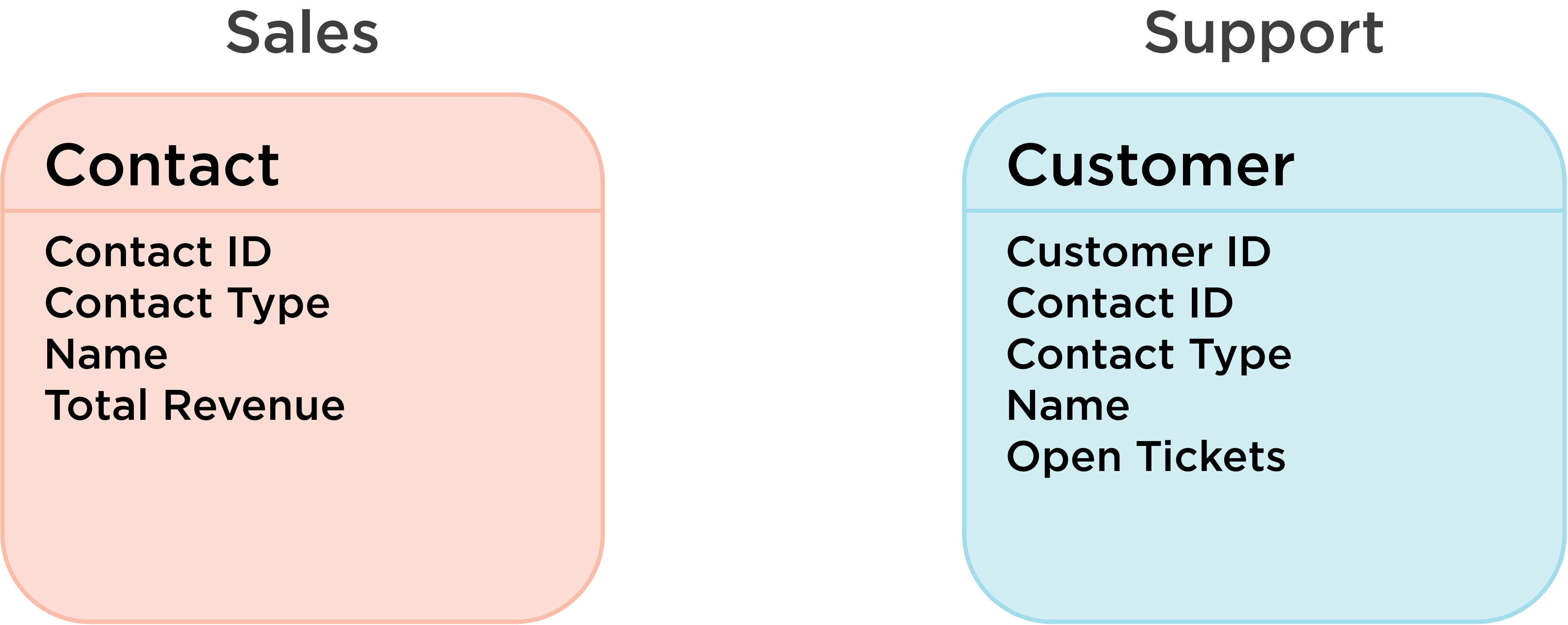
Overlapping Contexts



Bounded Contexts



Cross-boundary Entities



Sales

Contact

- Contact ID
- Contact Type
- Name
- Total Revenue

Support

Customer

- Customer ID
- Contact ID
- Contact Type
- Name
- Open Tickets

Microservices

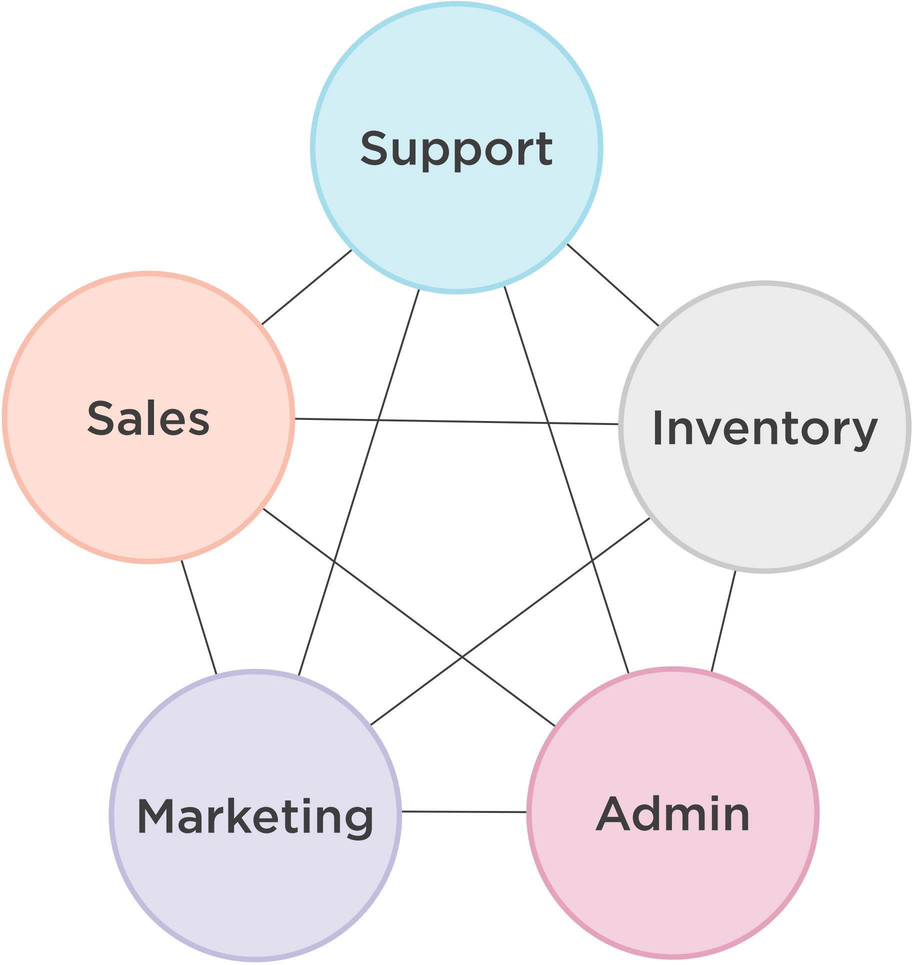
Subdivide monoliths

Clearly-defined interfaces

Small teams

Independent

Similar to SOA



Microservices

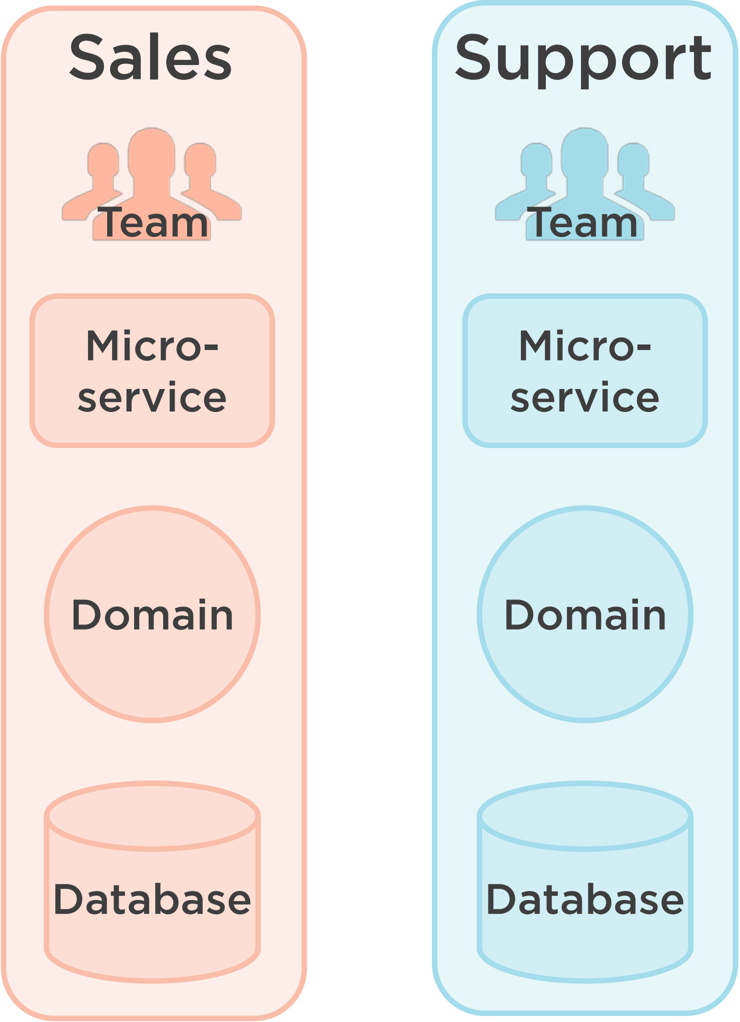
Bounded context

Cohesion/coupling

Single domain of knowledge

Consistent data model

Independence



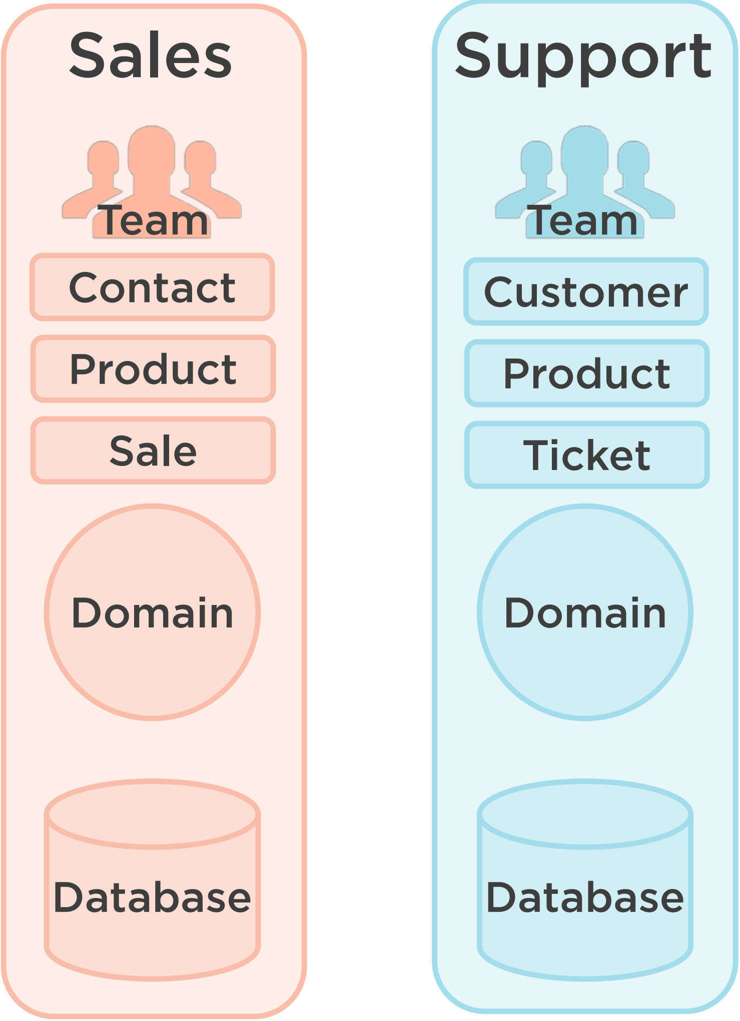
Microservices

How small?

Where to divide?

Microservice per aggregate root

Database per bounded context



Microservices

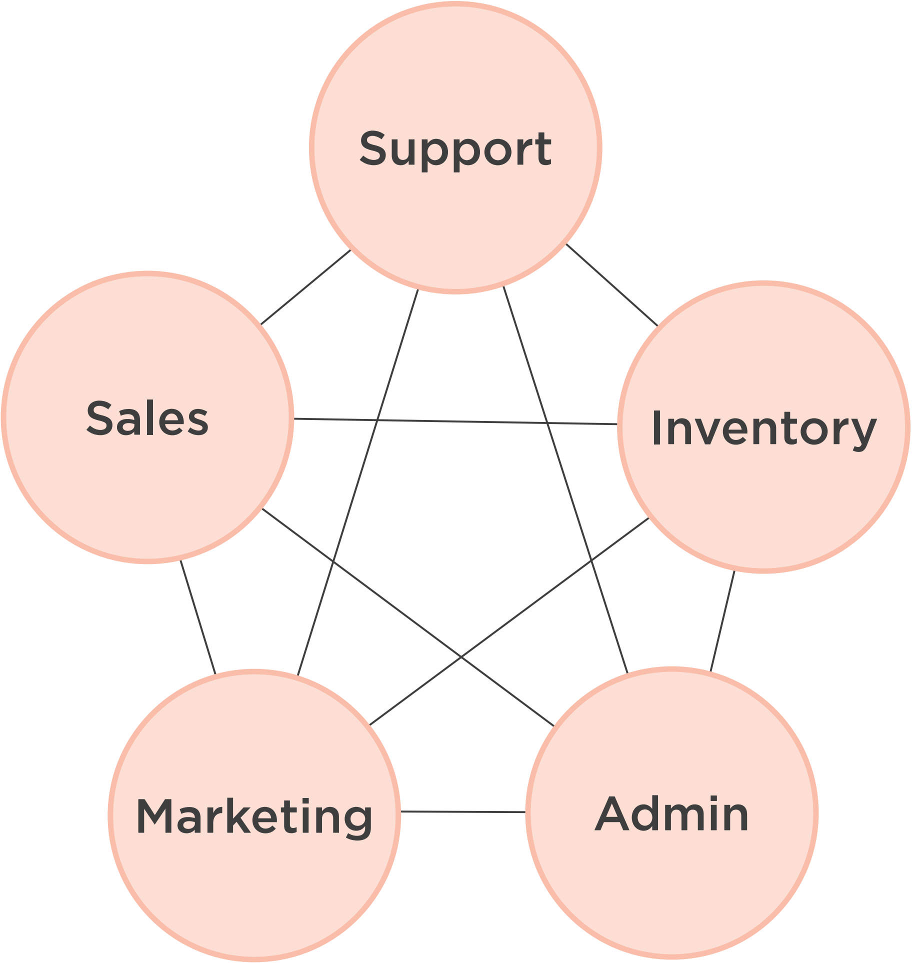
Why Use Microservices?

Pros

Flatter cost curve

Cohesion/coupling

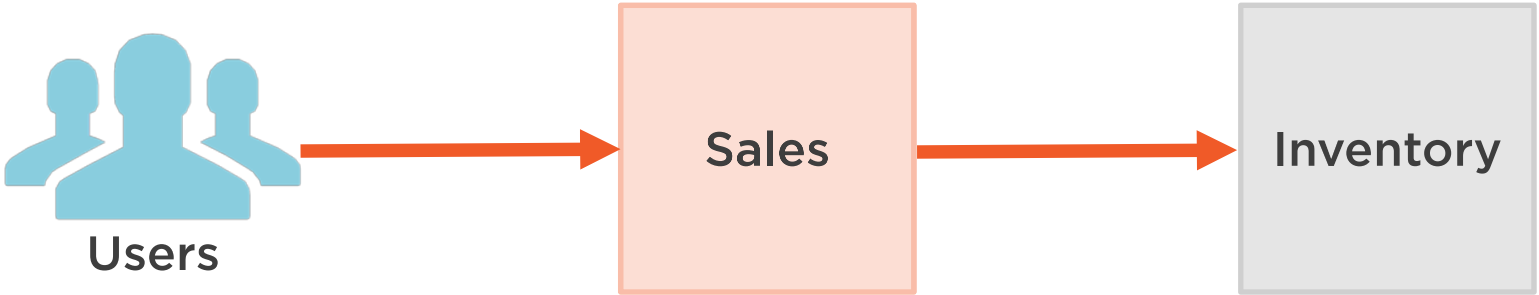
Independence



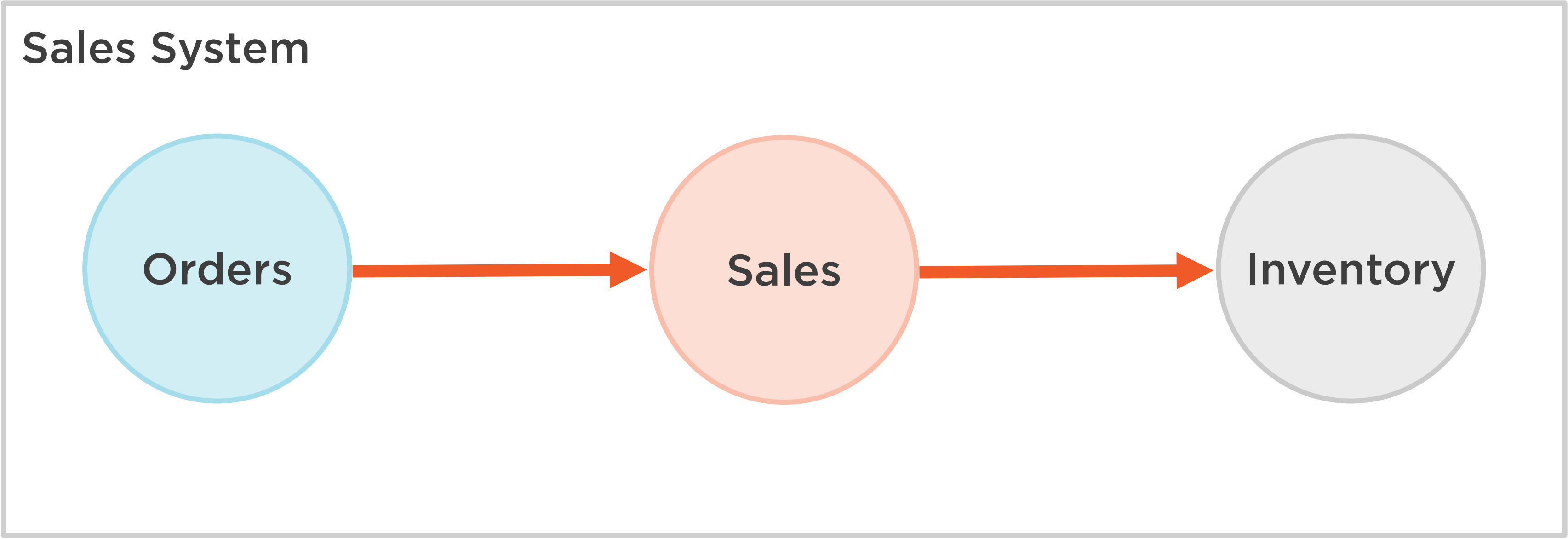
Why Use Microservices?

Pros	Cons
Flatter cost curve	Higher up-front cost
Cohesion/coupling	Conway’s Law
Independence	Distributed system costs

Microservice Demo



Microservice Demo



Setup 1

Setup 2

Show SalesController (API)

Show Create Sale Command

Show IInventoryService Interface

Show Inventory Service Class

Show IWebClient Interface

Show WebClient Class

Wrap-up

Summary

Bounded Context

Microservices

Pros and Cons

Demo