SOA & Microservices

Systems Integration

PBA Softwareudvikling/BSc Software Development

Tine Marbjerg

Fall 2017

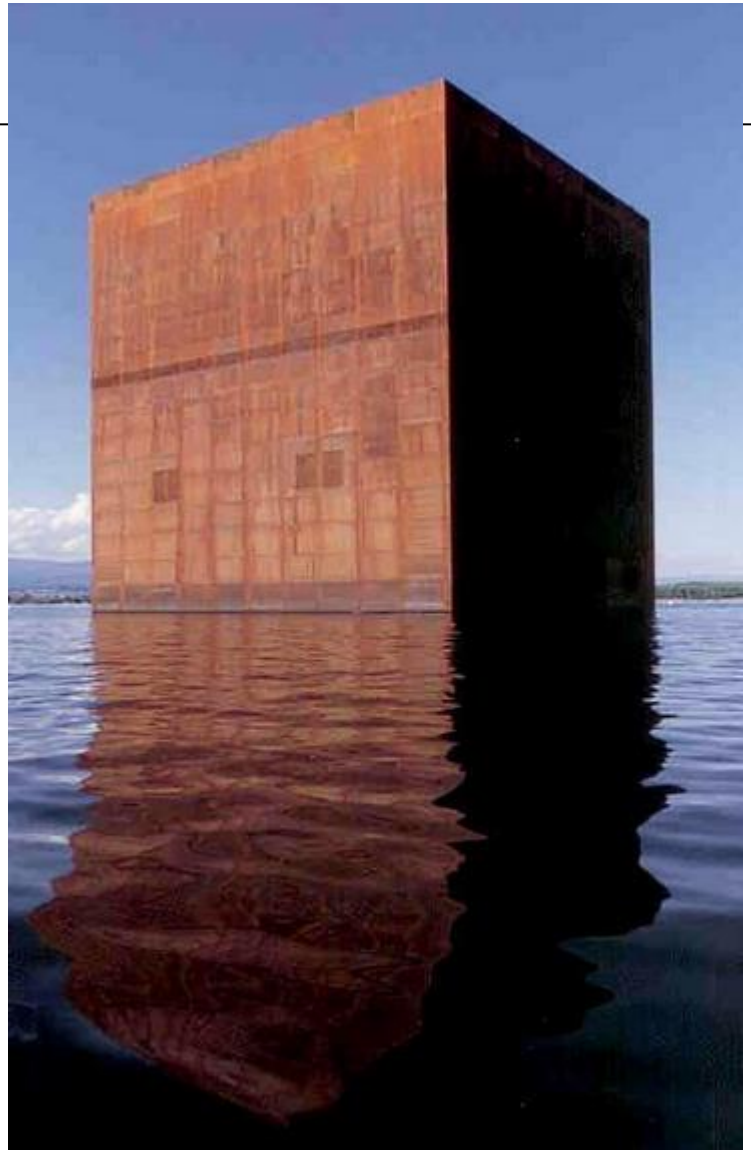
Today's Agenda

- Guest Lectures
  - Nordea Tore Green at 9.00
  - Process Factory Hans Peter Jensen at 10.30

- Definition of SOA
- Definition of microservices
- Service design
- ~~When/why use services?~~
- ~~IaaS, PaaS, SaaS~~

Before SOA

- Does it scale?
- Can we reuse parts?
- Is it maintainable?



Source: http://odino.org/on-monoliths-service-oriented-architectures-and-microservices/

Definition SOA

(SOA) is a design approach where

- multiple services **collaborate**
- to **provide** some set of **capabilities**

A service typically means

- a completely separate operating system process
- communication between these services occurs via calls across a network

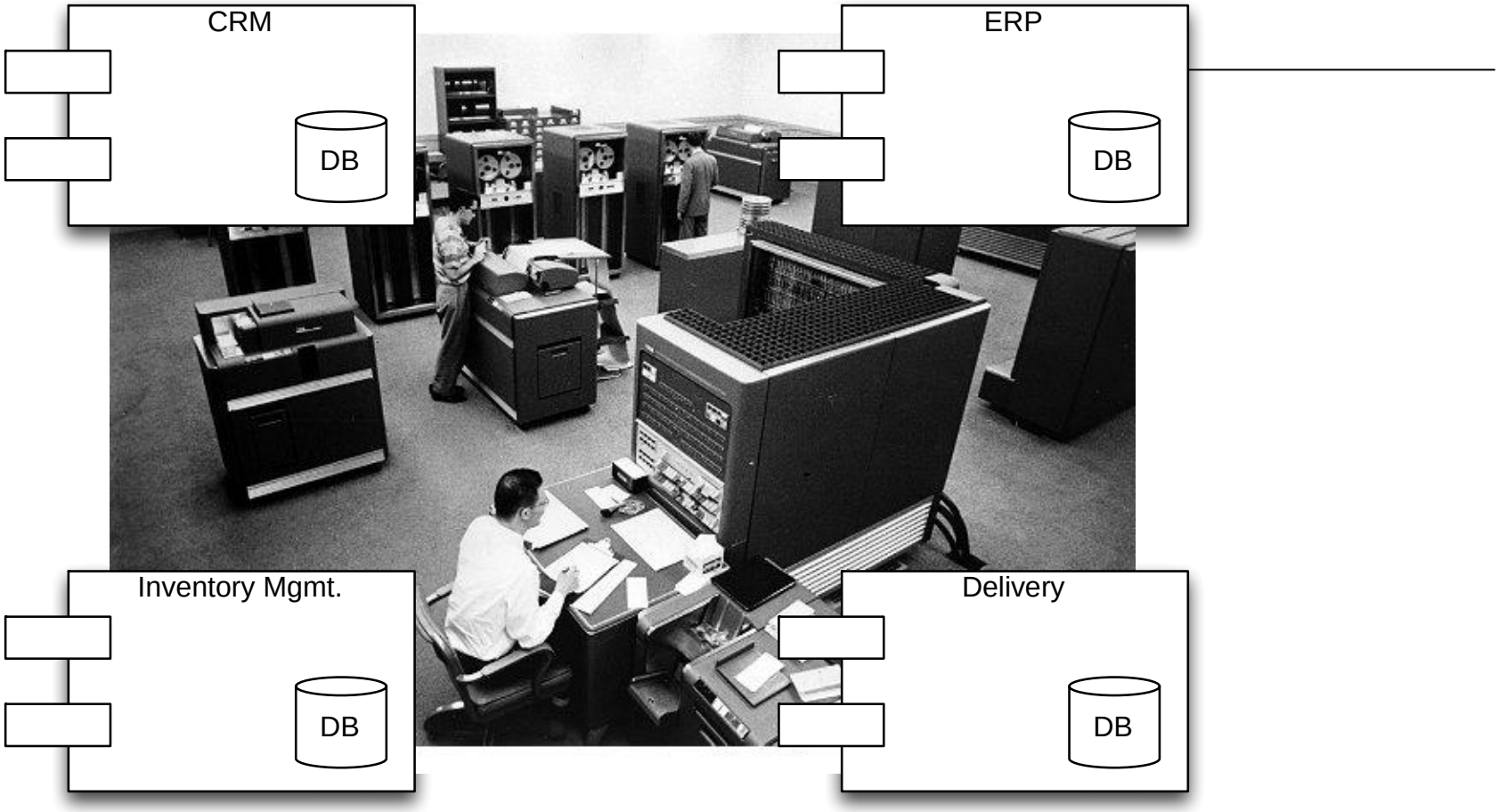
Sam Newman. “Building Microservices

What is it good for?

- Application integration
  - Intra-enterprise
  - Inter-enterprise (business-to-business)

- Application development & re-engineering

Need for integration of monoliths



- There is a need for systems to **cooperate,** e.g., to automate business processes supported by more than one application
- Main obstacle: apps developed **independently**, having different assumptions, data models, interfaces, platforms, etc.

EAI Example – Purchase Order Processing

Input: Purchase Order

- Validate customer ID and status
- Check customer credit
- Check inventory and package goods
- Start the delivery
- Prepare and send an invoice

Output: Delivery started, invoice sent.

EAI Example – Involved systems

1. Validate customer ID and status
   Customer Relation Management (CRM)

2. Check customer credit
   Enterprise Resource Planning (ERP)

3. Check inventory and package goods
   Inventory Management

4. Start the delivery
   Delivery System (outsourced)

5. Prepare and send an invoice
   Enterprise Resource Planning (ERP)

EAI Example – SOA says to ...

- publish relevant application functionality **as services**
- create **composite (integrating) application(s)** that call them

EAI Example – Some Involved Services

1. Validate customer ID and status
   - **Customer Relation Management (CRM)**
     - **GetCustomerDetails**
2. Check customer credit
   - **Enterprise Resource Planning (ERP)**
     - **CheckCustomerCredit**
3. Check inventory and package goods
   - **Inventory Management**
     - **PackageGoods**
4. Start the delivery
   - **Delivery System**
     - **StartDelivery**
5. Prepare and send an invoice
   - **Enterprise Resource Planning (ERP)**
     - **BillCustomer**

Software Services – Web Services

- *Web services* share the characteristics of more general services but:
  - expose their features over the Internet (or intranet) via standard (XML-based) languages & protocols,
  - are implemented via a self-describing interface based on open Internet standards.
- Web services can vary in function:
  - from **simple requests**, e.g., credit checking and authorization, pricing enquiries, inventory status checking, or a weather report
  - to **complete business applications** that *access & combine* info. from multiple sources, e.g., an insurance brokering system, an insurance liability computation, a package tracking system, etc.

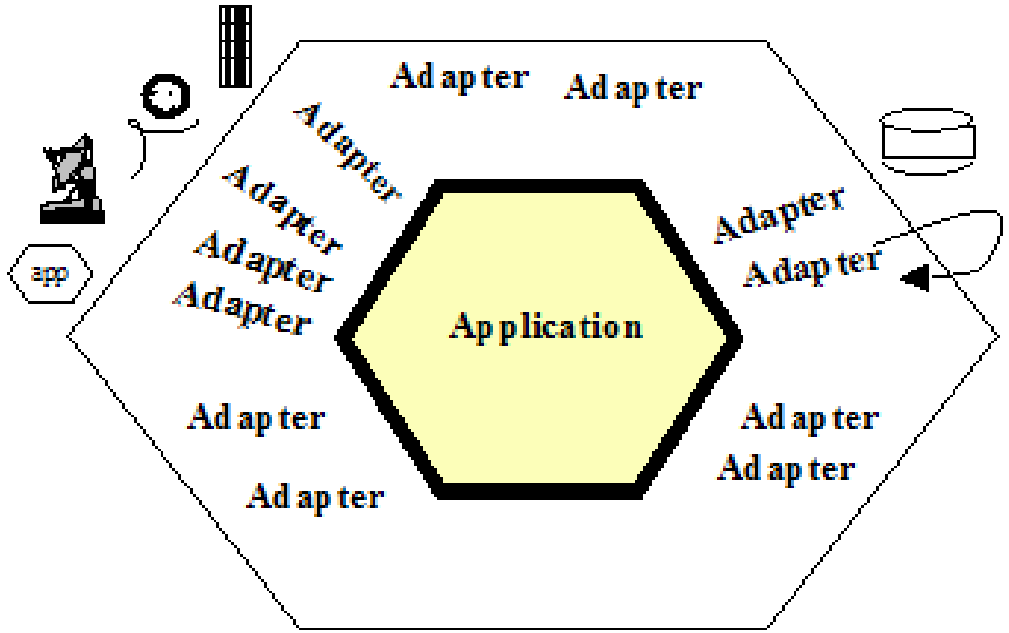
Software Services – Microservices

- Microservices are small, autonomous services that work together

Sam Newman. Building Microservices

Microservice: Small, and Focused on Doing One Thing Well

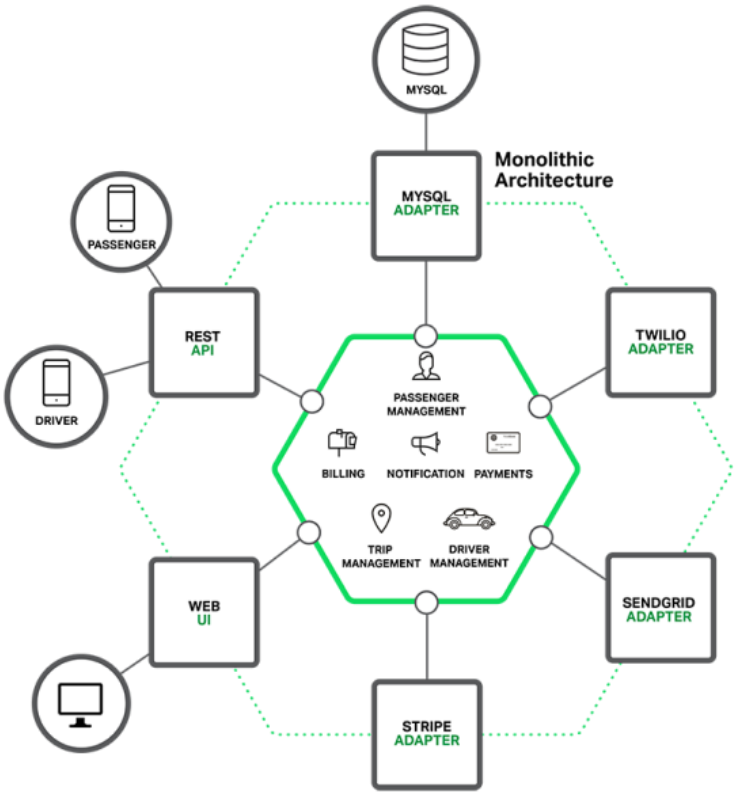
- Inspired by **Hexagonal architecture (**Alistair Cockburn) substitution of layered architecture where business logic could hide

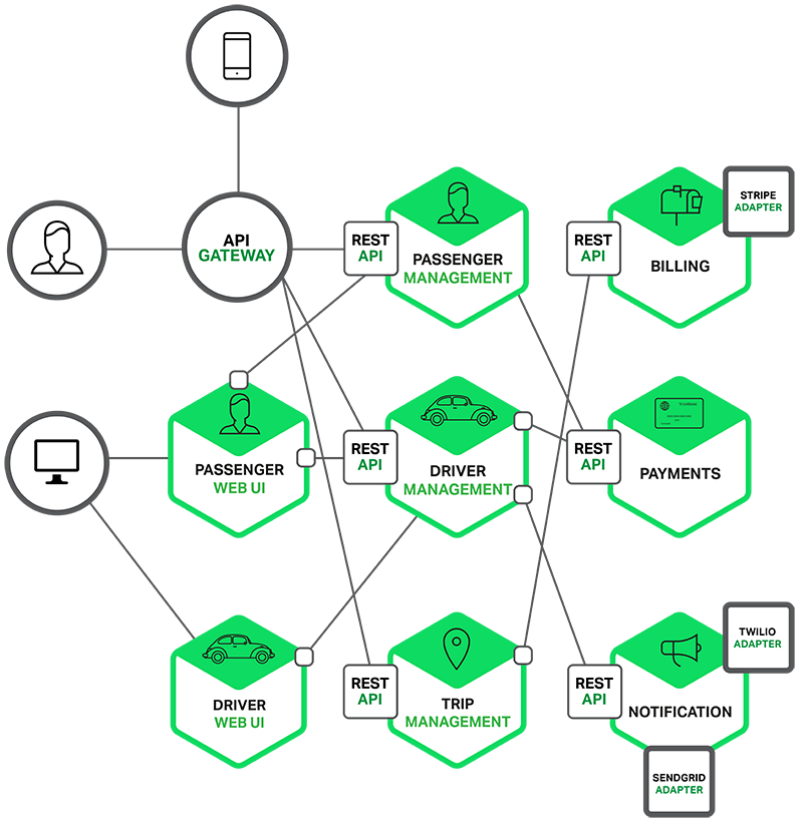


- Increased testability – automated testing of logic without UI
- Can be driven by another application

Monolithic vs. Micro Service Architecture

Example: Taxi system like Uber





Resource: https://www.version2.dk/artikel/traet-it-monolitten-proev-microservice-1070559

Microservice: Autonomous & scalable

A microservice is a **separate entity**.

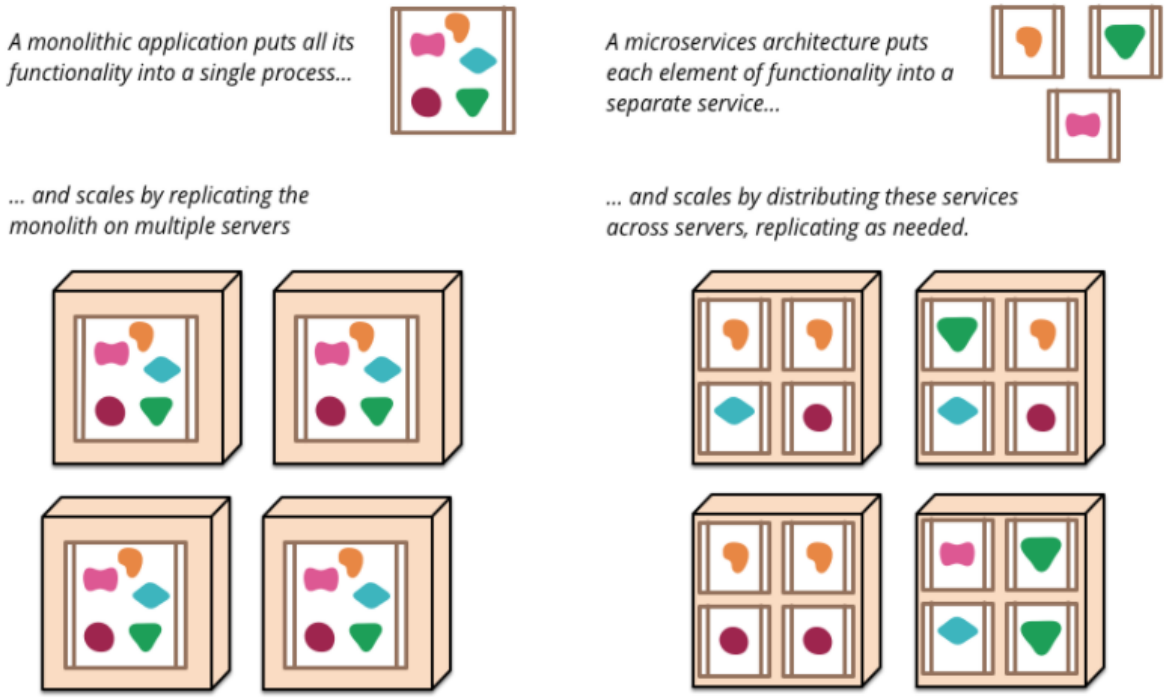


Figure 1: Monoliths and Microservices

Martin Fowler. Microservices

Microservices: other criteria & benefits

- It might be **deployed** as an **isolated** service on a platform as a service (PAAS), or it might be its own operating system process

- Services need to be able to **change** independently of each other, and be **deployed** by themselves without requiring consumers to change.

- "**Micro**" is aligned to **team structures**. If the codebase is too big to be managed by a small team, looking to break it down is very sensible.

Sam Newman. "Building Microservices

Types of cloud computing

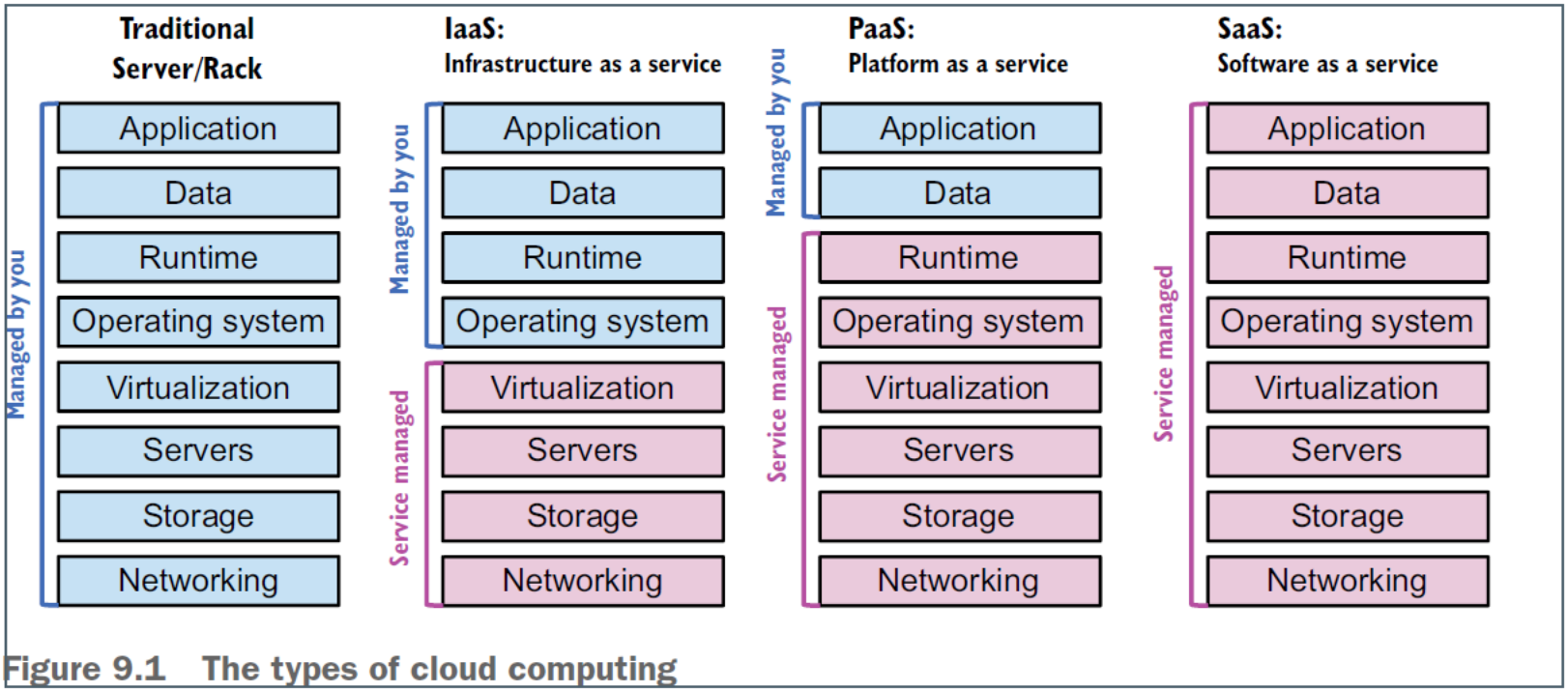
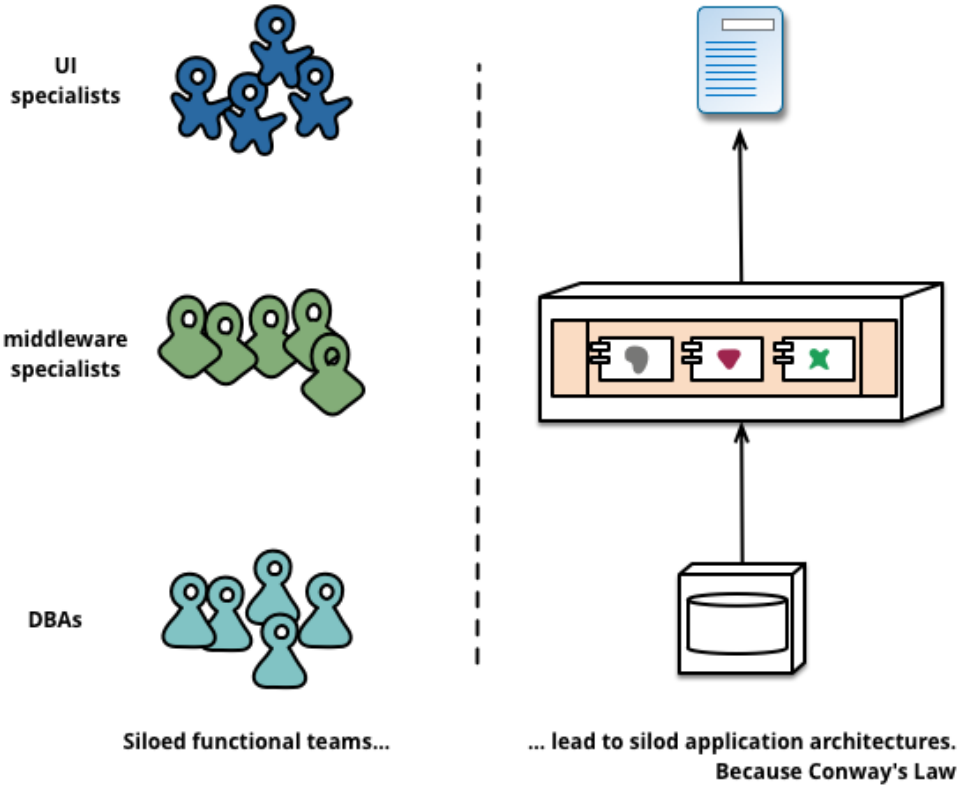


Figure 9.1 The types of cloud computing

System design follows organization 1

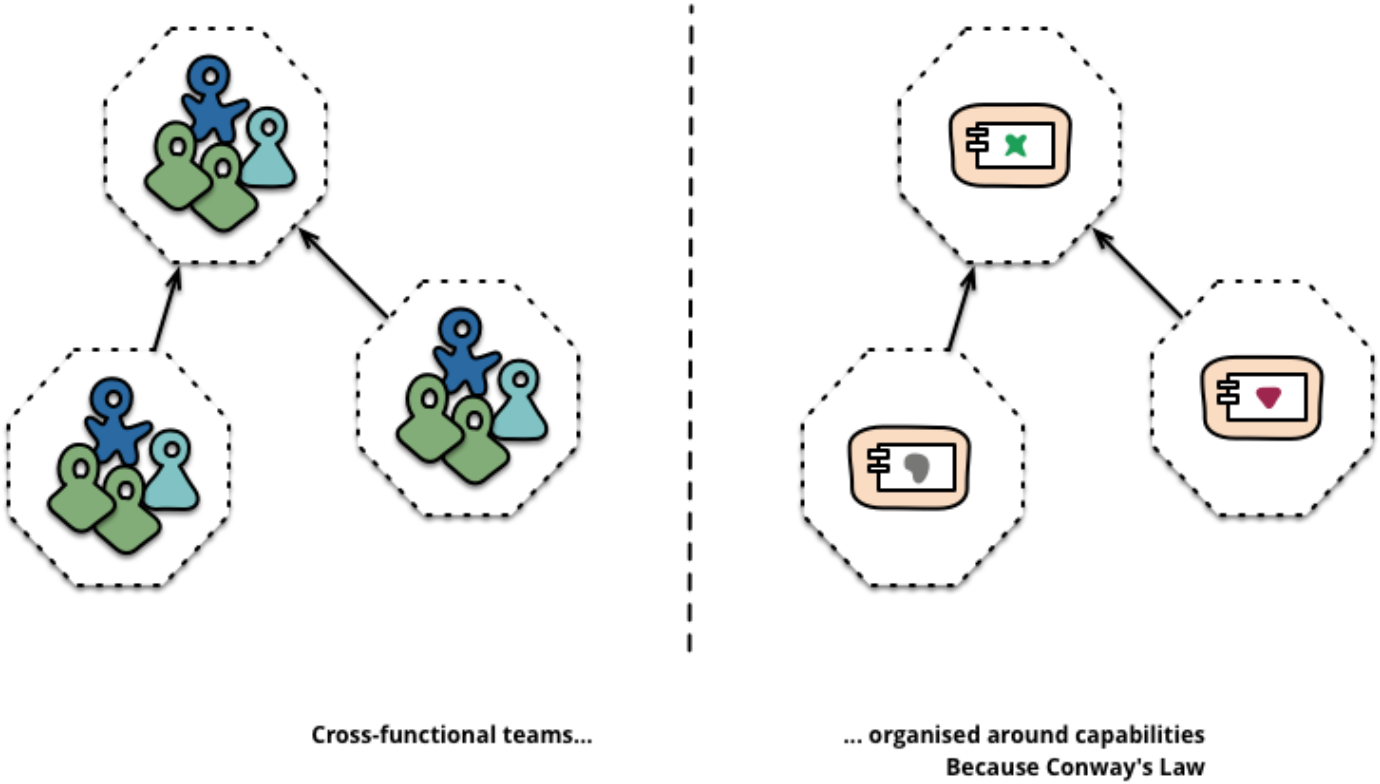
Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

-- Melvyn Conway, 1967



System design follows organization 2

Micro services are organized around **business capability** in **agile cross functional teams:**



SOA & Microservices

[microservices are] one form of SOA,
perhaps service orientation done right

https://martinfowler.com/articles/microservices.html

Software Services – Microservices

Microservices support:

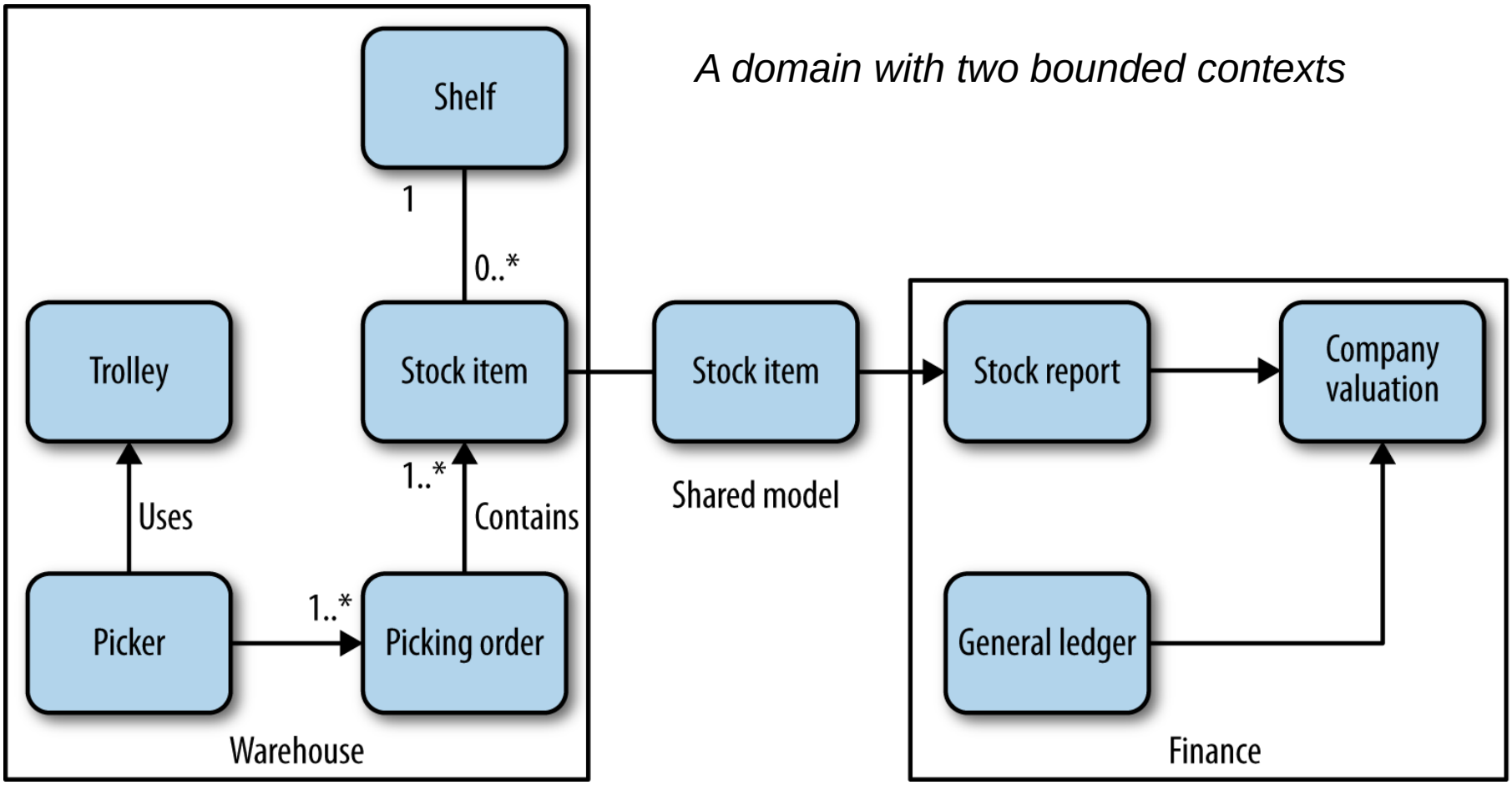
- Technology Heterogeneity
- Resilience
- Scaling
- Ease of Deployment
- Organizational Alignment
- Composability
- Optimizing for Replaceability

Sam Newman. “Building Microservices

Good service design

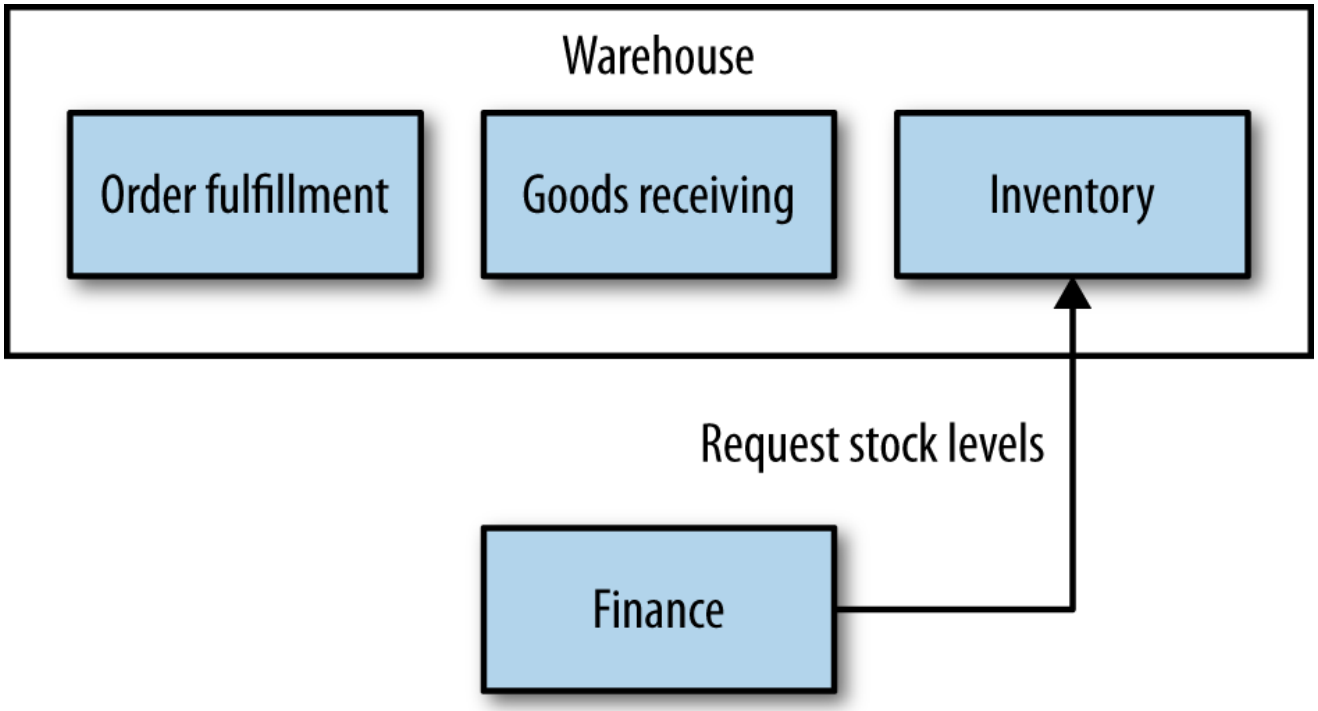
- Loose coupling
  - Easy deployment without needing to change other parts of the system
- High cohesion
  - If we want to change behaviour, we only want to do it in one place
- Bounded context
  - Explicit interface
  - Hide details

Shared model of finance & warehourse ex.



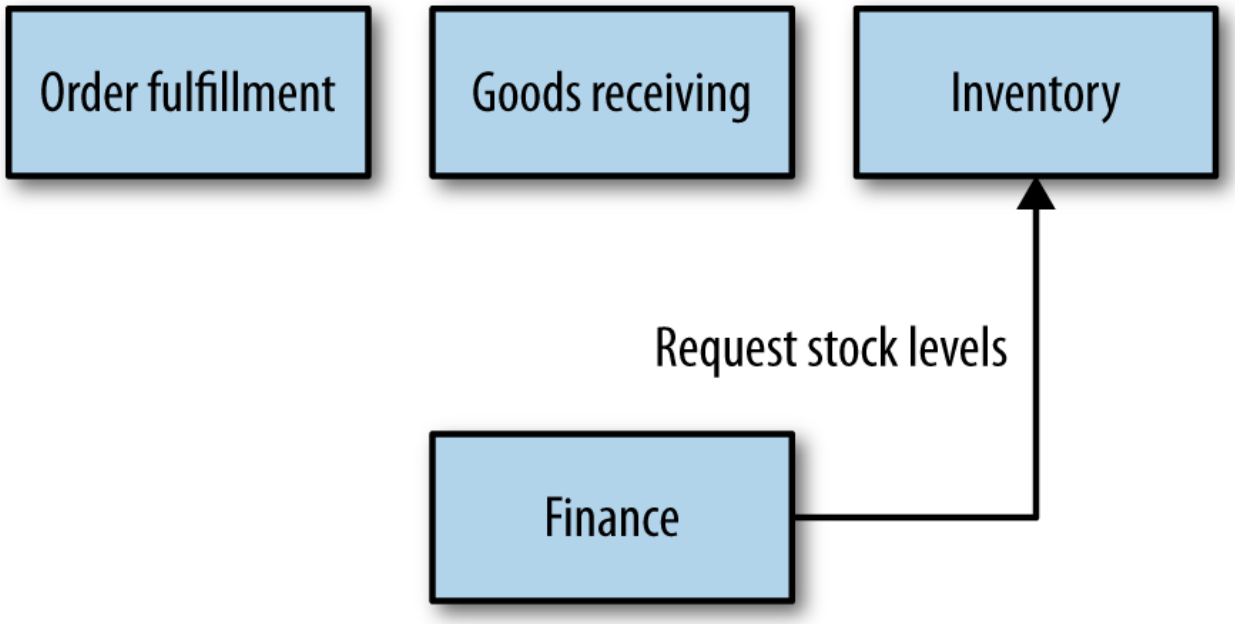
Modular boundaries are candidates for microservices 1

- Option 1: Decomposition into microservices with nested bounded contexts hidden inside the warehouse



Modular boundaries are candidates for microservices

- Option 2: The bounded contexts inside the warehouse being popped up into their own top-level contexts



Which option to choose?

?