Thinking Microservices

Erik Hoogendoorn

Microsoft Cloud Architect

Noordwijkerhout

https://www.erikhoogendoorn.com

erikh@delta-n.nl

@ehoog

Delta-N
Connecting the Cloud

Monolithic Application Model

Monolithic Application Model

- ✓ Straightforward to develop
- ✓ Simple to build
- ✓ Testable
- ✓ Easy to deploy
- ✓ Easy to scale

Monolithic Application Model

- ✗ High learning curve existing codebase
- ✗ Scaling teams is hard
- ✗ How to adopt new technologies
- ✗ Scales for the wrong reasons
- ✗ Huge databases
- ✗ Does not enforce loose coupling
- ✗ Deployment downtime

Monolithic Application Model

Delta-N
Connecting the Cloud

Direction in Software Development

- Agile
- Continuous Integration/Continuous Delivery
- Autonomous Teams
- Platform as a Service

Demands in Application Development

- Mobile + Web
- Resilient
- Scalable
- Fast time to market of features
- Always on, zero downtime

Delta-N
Connecting the Cloud

Microservices Architecture

...the microservice architectural style is an approach to developing a single application as a **suite of small services**, each running in its **own process** and **communicating with lightweight mechanisms**, often an HTTP resource API. These services are built around **business capabilities** and **independently deployable** by fully automated deployment machinery. There is a bare minimum of centralised management of these services, which may be written in **different programming languages** and use **different data storage technologies**. - *James Lewis and Martin Fowler**

*https://martinfowler.com/microservices/

Delta-N
Connecting the Cloud

Microservices Architecture

Microservice applications are composed of small, **independently versioned**, and **scalable** customer-focused services that **communicate** with each other **over standard protocols** with well-defined interfaces - *Microsoft Docs**

*https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview-microservices

Delta-N
Connecting the Cloud

Microservices Architecture
"Micro"

- How big or small? (No universal measure)
- Does one thing, does it well
  - Scoped by business capability
  - Identify sub-domains
  - Bounded Context (Domain-Driven Design)

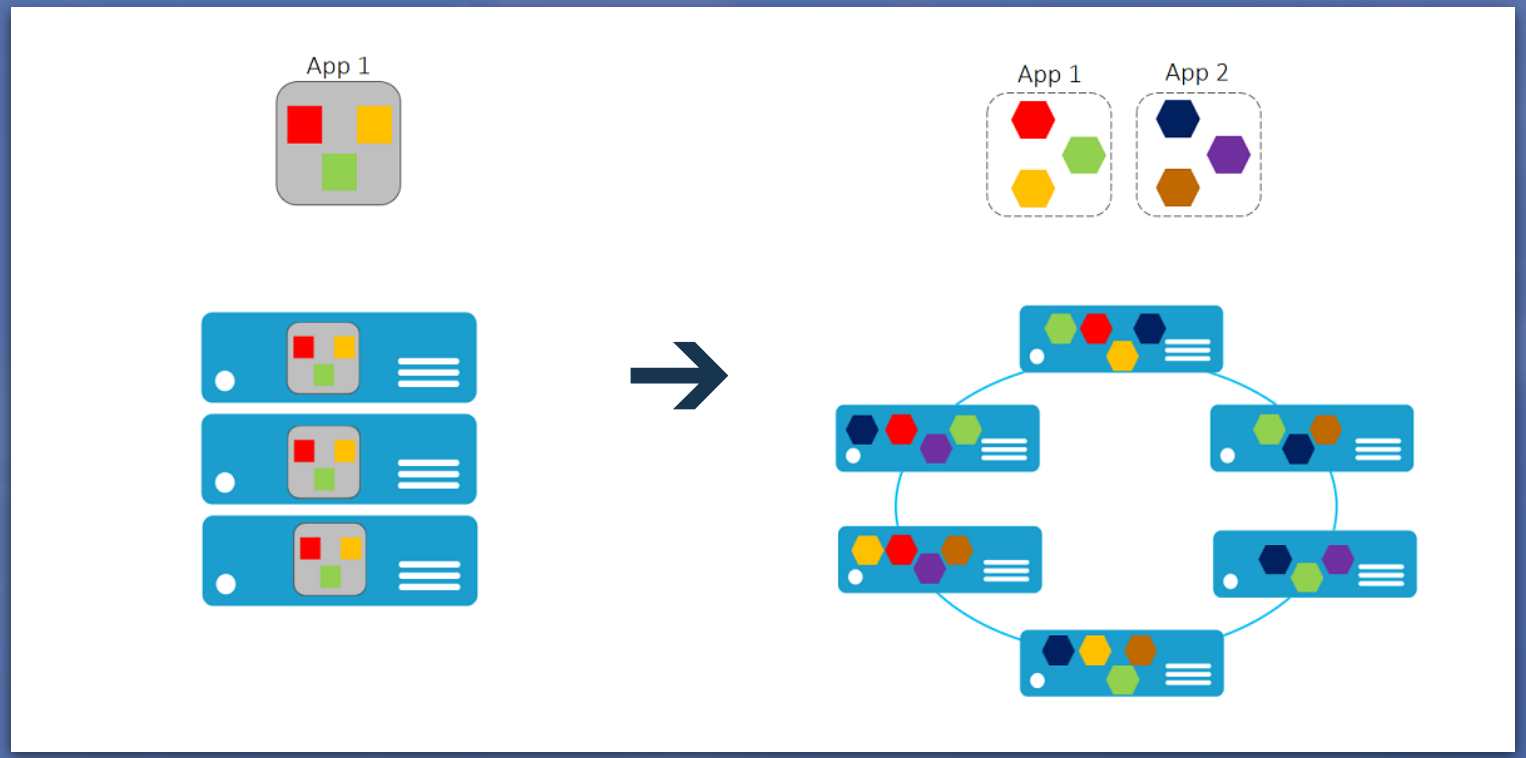
Microservices Architecture
"Service"

- Independently deployable component
- Interoperable with each other
- Versioning
- Monitored
- Independently scalable

Microservices Architecture

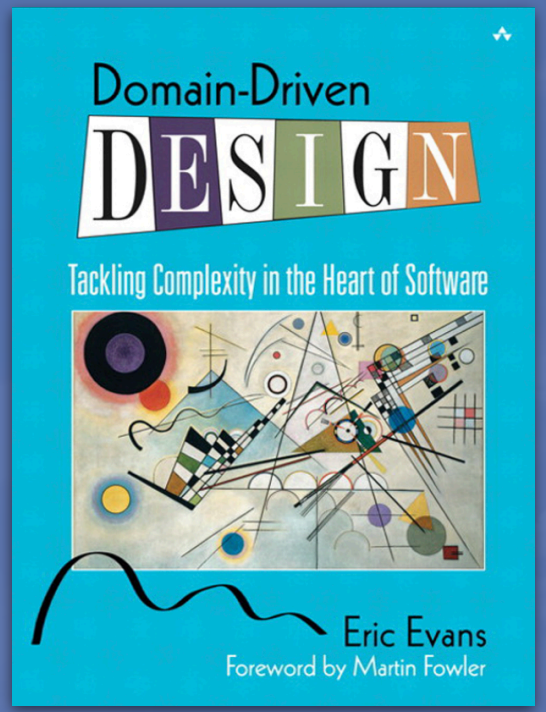
Delta-N
Connecting the Cloud

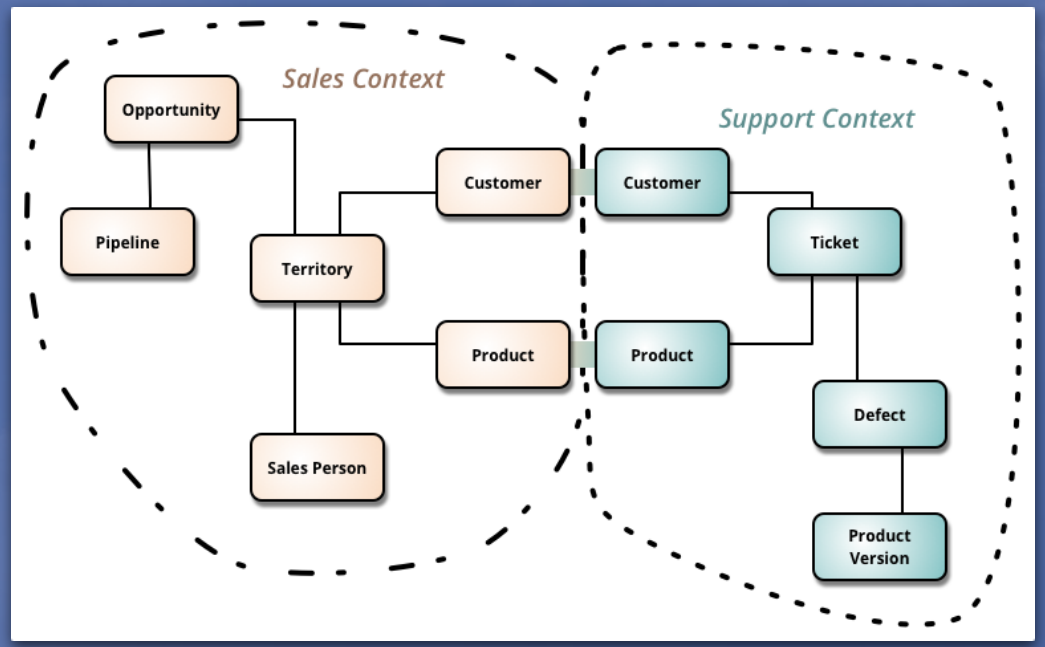
Moving to Microservices



Delta-N
Connecting the Cloud

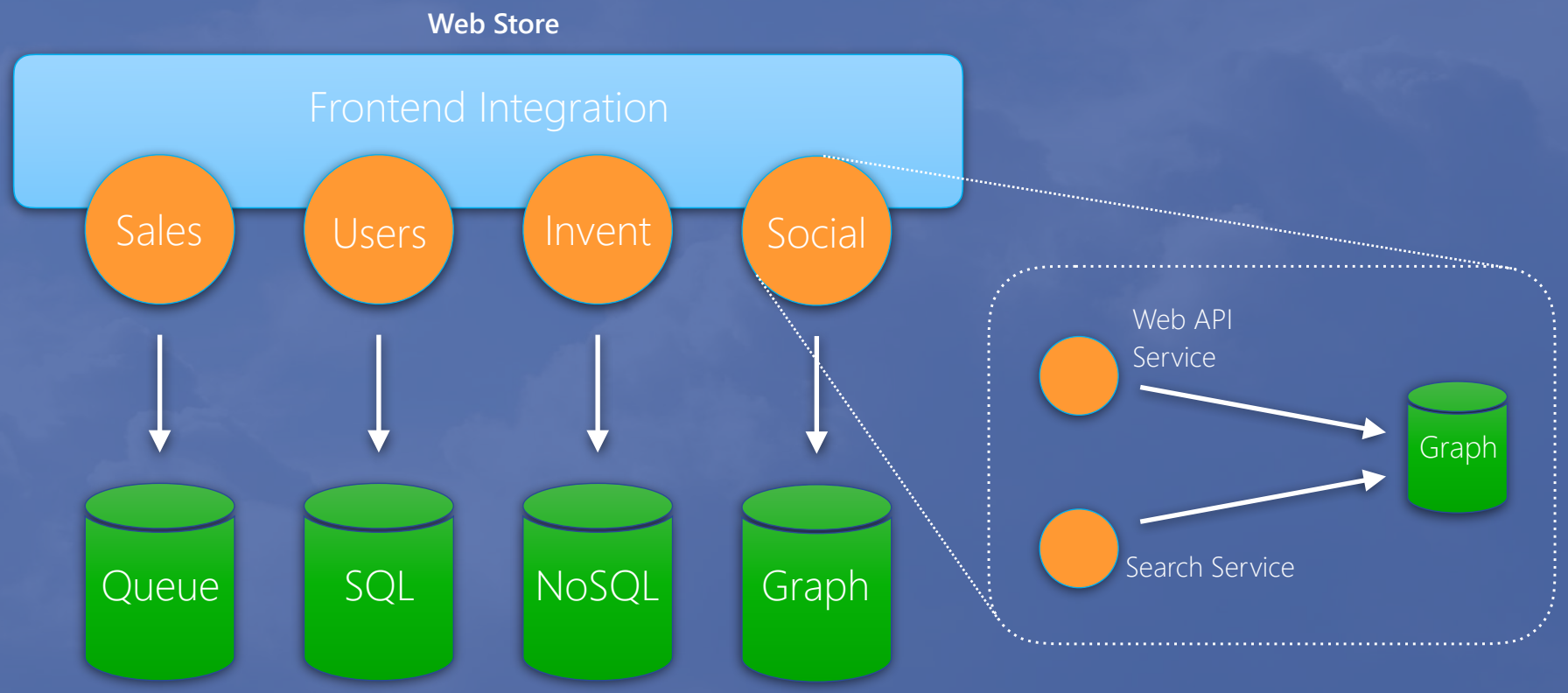
Moving to Microservices - Define sub-domains





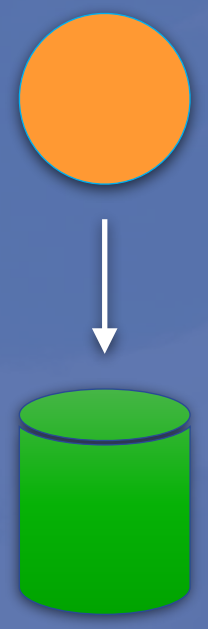
Delta-N
Connecting the Cloud

Moving to Microservices -Splitting things up



Delta-N
Connecting the Cloud

Moving to Microservices -Splitting things up



- Independently versioned, deployable & scalable
- No shared model code
- Well defined interfaces (REST, Messaging)
- Owns its own data store
- Belong to 1 bounded context, never more
- Technology of choice
- Owned by 1 team only (feature teams!)

Delta-N
Connecting the Cloud

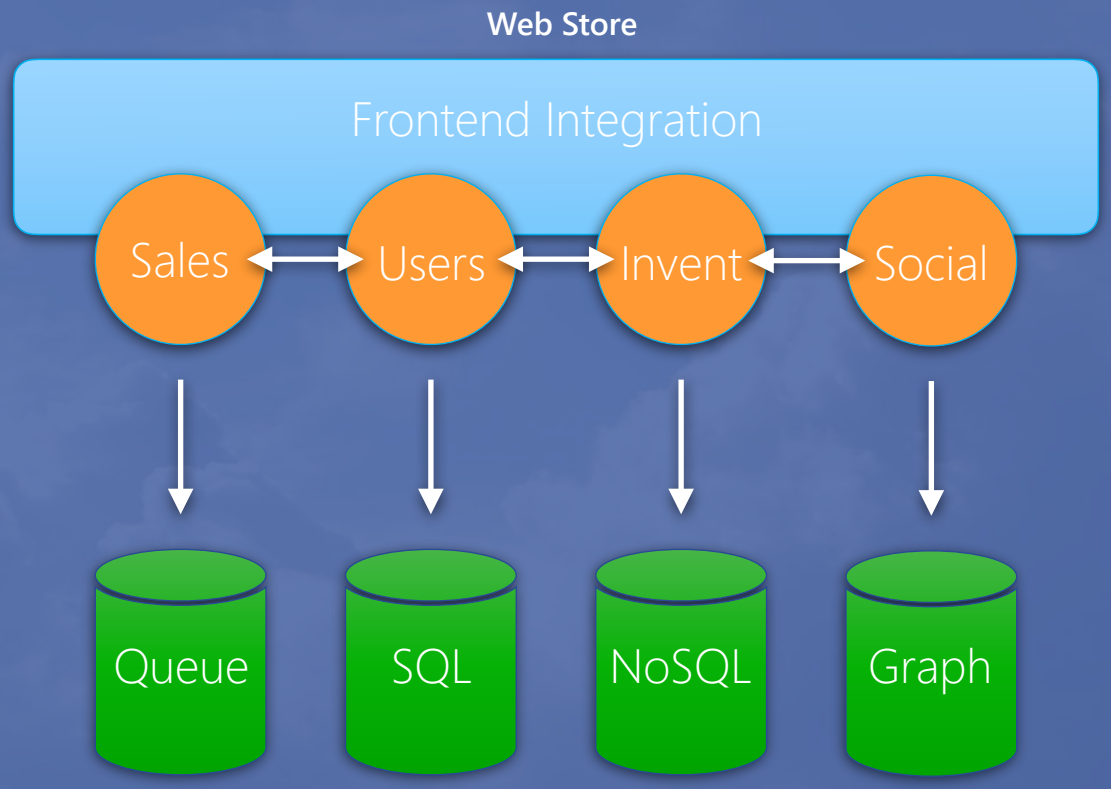
Moving to Microservices

Any organisation that designs a system will inevitably produce a design whose structure is a copy of the organisation's communication structure - ***Conway's Law****

*http://www.melconway.com/research/committees.html

Delta-N
Connecting the Cloud

Moving to Microservices -Splitting things up



Delta-N
Connecting the Cloud

Moving to Microservices - Event driven

- Services use event messages to notify others of changes in the system
- Service Bus, Event Grid
- Services use events to replicate data
- Services store replicated data in their own data store
- Extensibility!

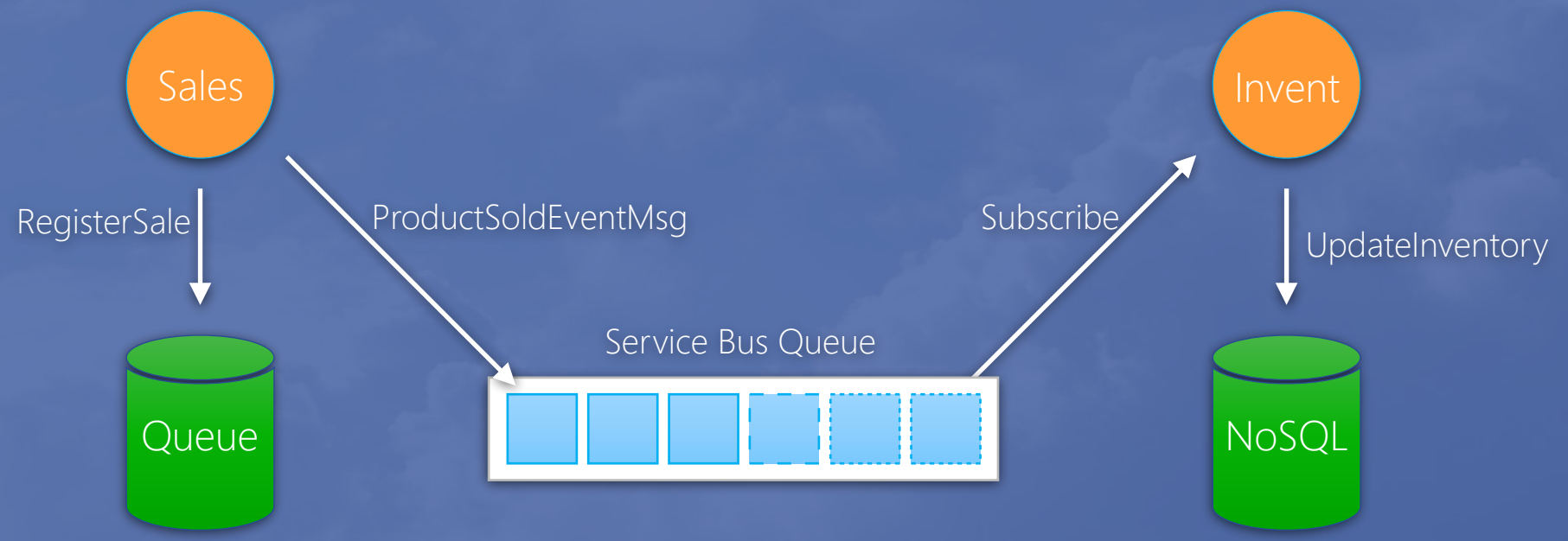
Moving to Microservices - Event driven

Challenges

- Data Consistency
- Decentralised Business Logic
- Fault reconciliation

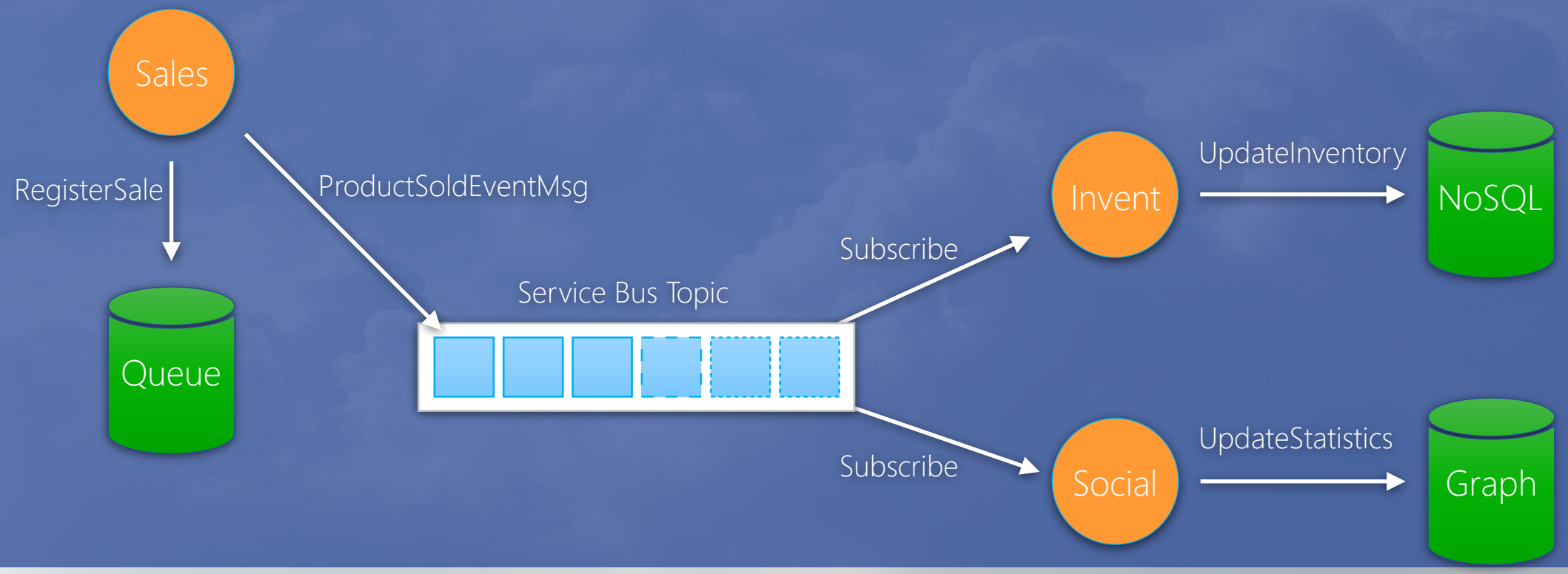
Delta-N
Connecting the Cloud

Moving to Microservices - Event driven



Delta-N
Connecting the Cloud

Moving to Microservices - Event driven



Delta-N
Connecting the Cloud

Moving to Microservices - Design for Failure

- Services can ***and will*** fail
- Build features that deal with failure
- Message idempotency
- Asynchronous over synchronous
- Stress testing
- Use design patterns*
  - Retry
  - Circuit breaker
  - Command Query Responsibility Segregation (CQRS)
  - Event Sourcing

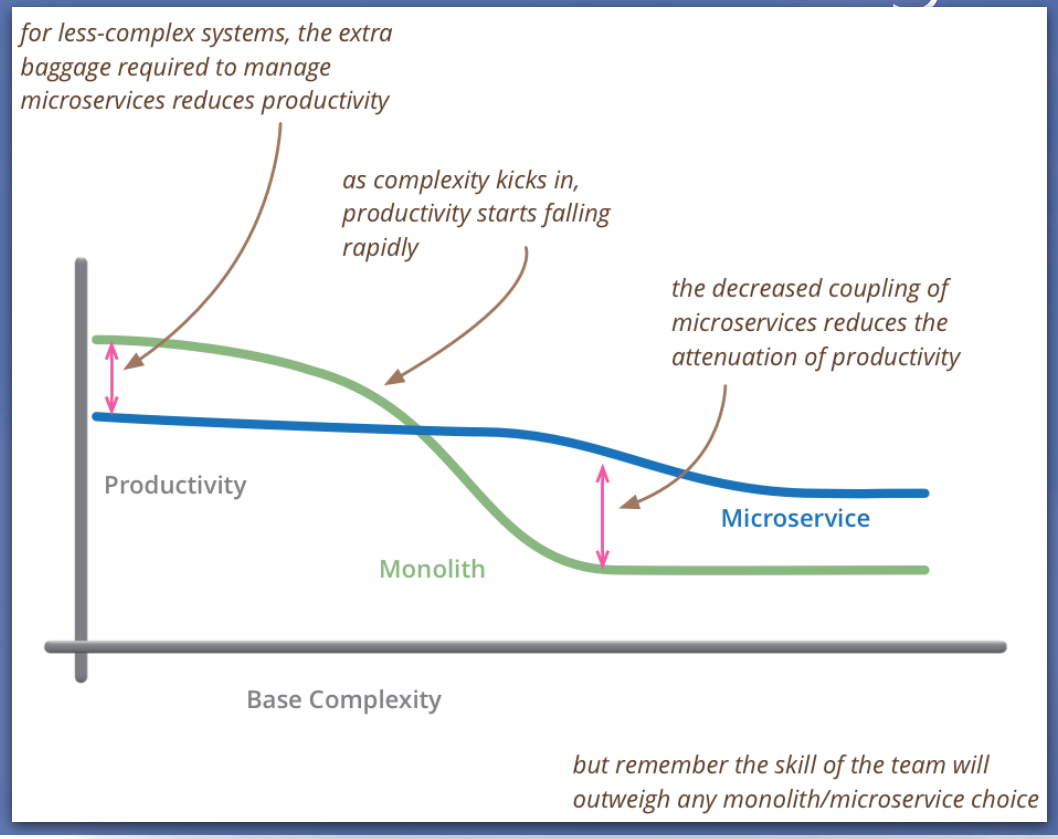
*https://docs.microsoft.com/en-us/azure/architecture/patterns/

Delta-N
Connecting the Cloud

Microservices in Microsoft Azure

- Web API App Service
- Azure (Durable) Functions
- Service Bus
- Event Grid
- CosmosDB
- API Management
- Service Fabric (Mesh)
- Containers

Choosing Microservices



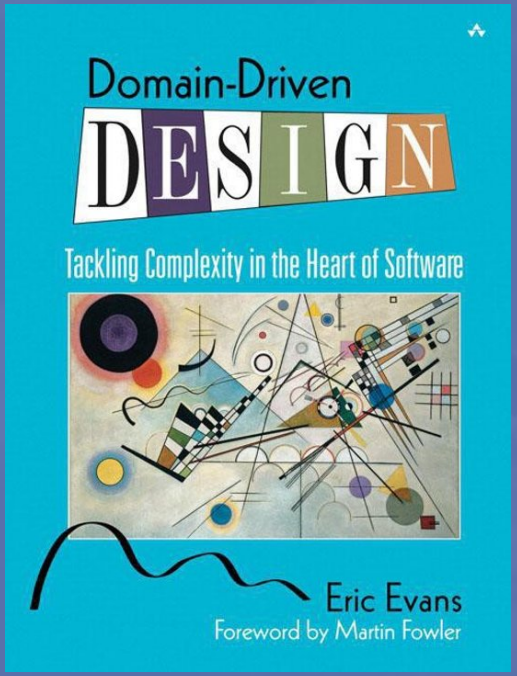
- Scalable
- Faster Time to Market
- Extensible
- Available
- Replaceable

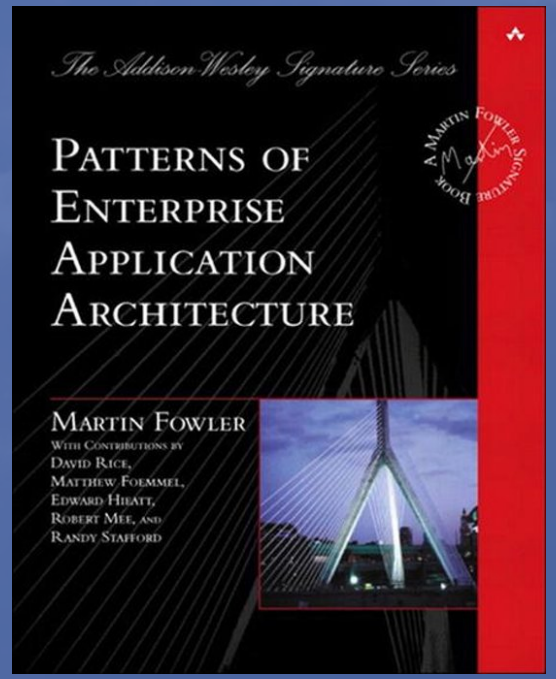
Delta-N
Connecting the Cloud

Take aways

- Start small, refactor and evolve
- Use Domain-Driven Design concepts
- Keep teams small and manageable
- Design to Fail
- Asynchronous over synchronous
- Use Azure PaaS to your advantage

Resources - Books





Delta-N
Connecting the Cloud

Resources - Web

- *Cloud Design Patterns*
  https://docs.microsoft.com/en-us/azure/architecture/patterns/
- *Understanding Microservices (SF Docs)*
  https://docs.microsoft.com/en-us/azure/service-fabric/service-fabric-overview-microservices

Resources - PluralSight

- Microservices: The Big Picture
- Microservices Architecture
- Microservices Architectural Design Patterns Playbook
- Containerizing a Software Application with Docker
- Domain-Driven Design Fundamentals

Questions?

Delta-N
Connecting the Cloud

THANK YOU!

Erik Hoogendoorn

Microsoft Cloud Architect

Noordwijkerhout

https://www.erikhoogendoorn.com

erikh@delta-n.nl

@ehoog

Delta-N
Connecting the Cloud