



ENTERPRISE

Domain Driven Design for Realtime, Ubiquitous, Distributed Data

PRESENTED BY:

Laura Schornack

Maureen Penzenik

Introduction



Laura Schornack

Laura Schornack champions Domain Driven Design whenever possible. She holds a degree in computer science from the University of Illinois at Urbana-Champaign. She holds four patents. Laura presents frequently and recently presented at Kubecon. She is published through O'Reilly's Katakoda

Laura volunteers with “Girls who Code”, “Coder Dojo”, “Think Chicago”, and “City Scholars”.



Maureen Penzenik

Maureen Penzenik has been in Information Technology for over 25 years with experience in applying Data Warehousing and Business Intelligence methods and technologies. Maureen holds a B.S. degree in Business Administration and an MBA in Management Information Systems.

Agenda

Topics

- Summary of Problem
- Data Industry Past Attempts and Why They Were Flawed
- Proposed Solution
- What Is Domain Driven Design (DDD) and Enterprise DDD
- Leveraging Governance to Take the Leap from DDD to EDDD
- DDD And EDDD Example
- EDDD Example Applied in a Distributed Manner
- CQRS with Operational Systems
- Differences Between Data and Responsibility Segregation for Applications and for Data/Analytics Environments

Problem

Desired Outcome

- Cut the time of application delivery by not redoing all of the data work in every project.

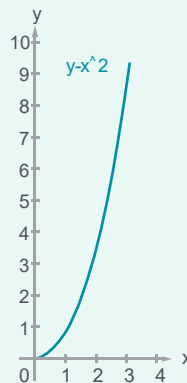
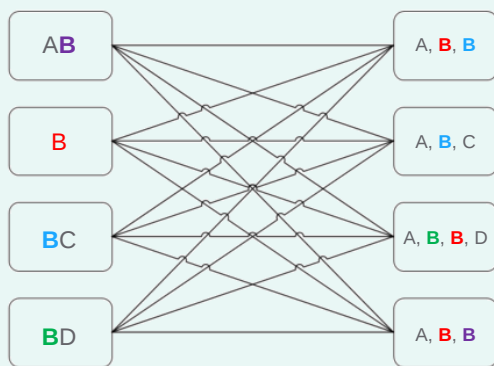
Problem

- Data cannot be sourced from and created in different applications, with the expectation that the boundaries are the same.
- Inconsistent sourcing and creation of duplicate datasets has Monetary, Opportunity, and Reputational costs:
 - Errors
 - Inconsistency
 - Exponential technical debt
 - Reactive approach to data problems
 - Audit issues: unknown authentic source and lack of identified data owners.
 - Data Owners are not included in decisions on how data is used, assuming the same data has consistent definitions.
 - Duplicate data being consumed that is not fit for purpose.
- Individual applications determining business definitions and boundaries for datasets – lack of uniformity.
- Governance in managing data sets.

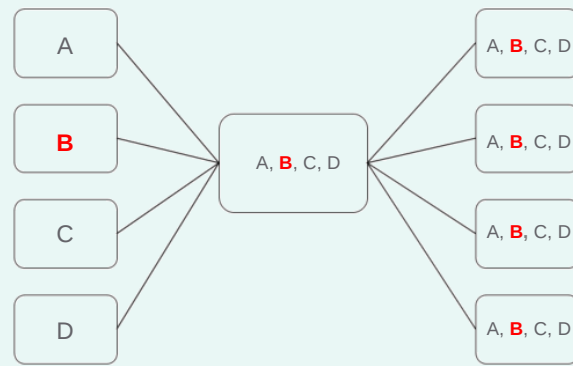
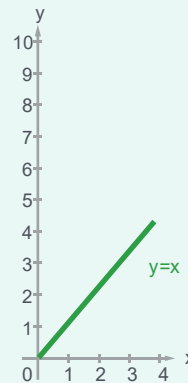
Problem

Inconsistent Data Sourcing

Not Ideal



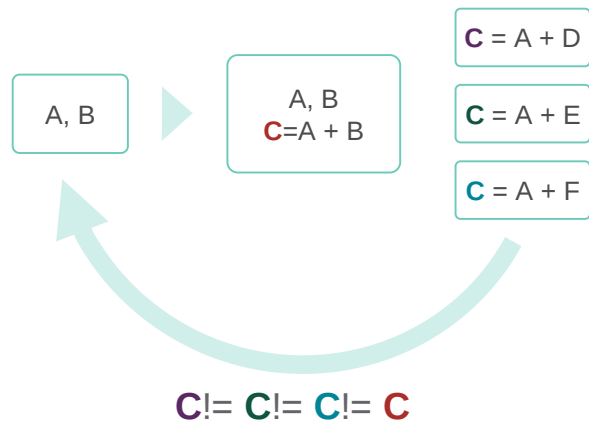
Ideal



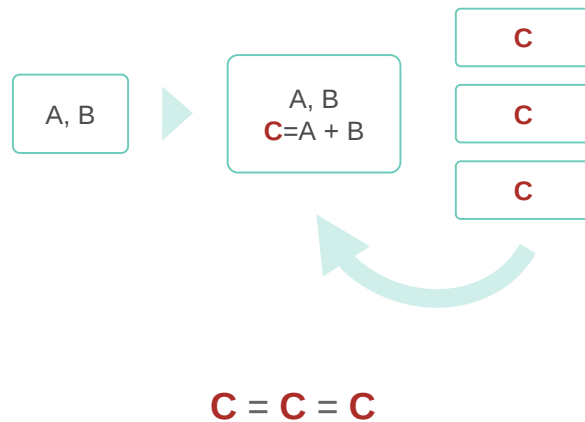
Problem

Inconsistent Data Derivation

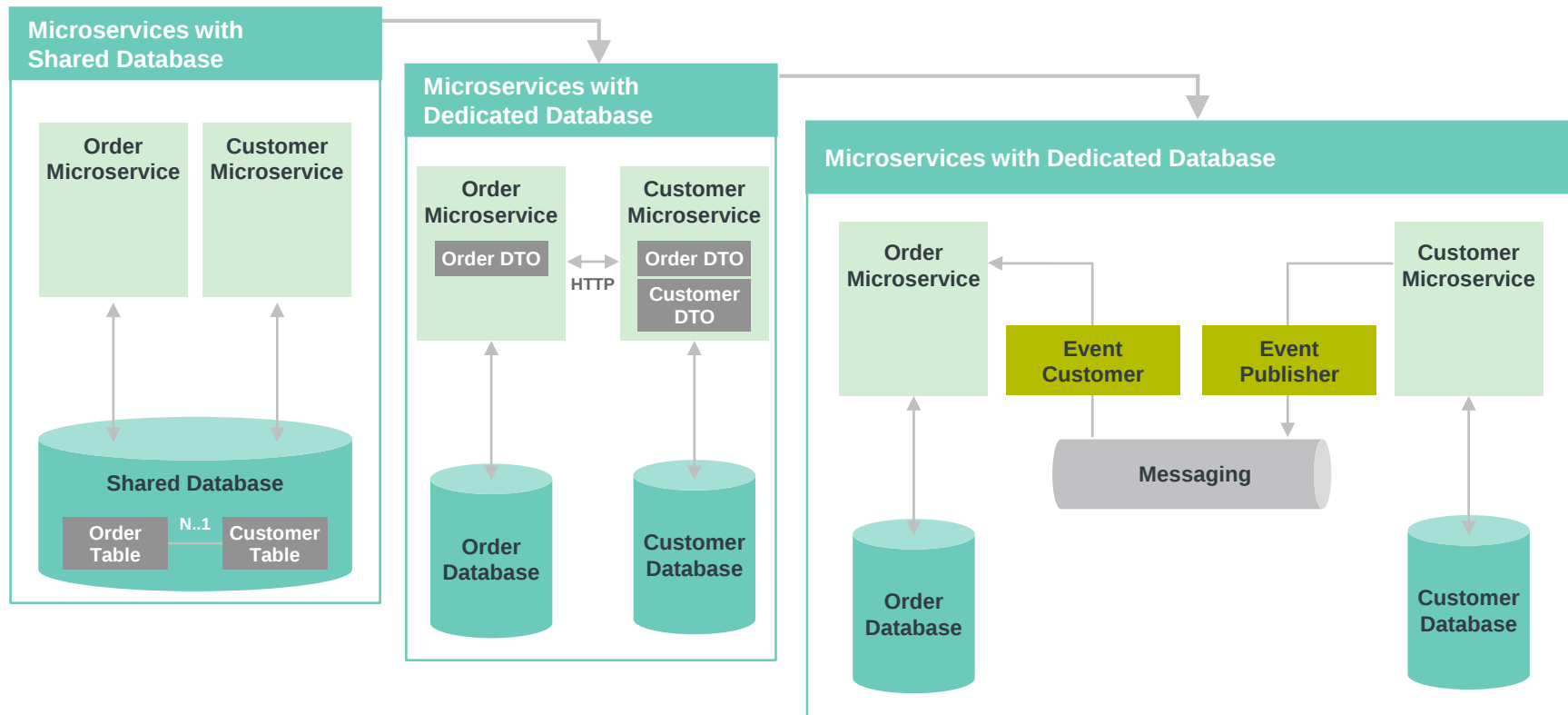
Not Ideal



Ideal



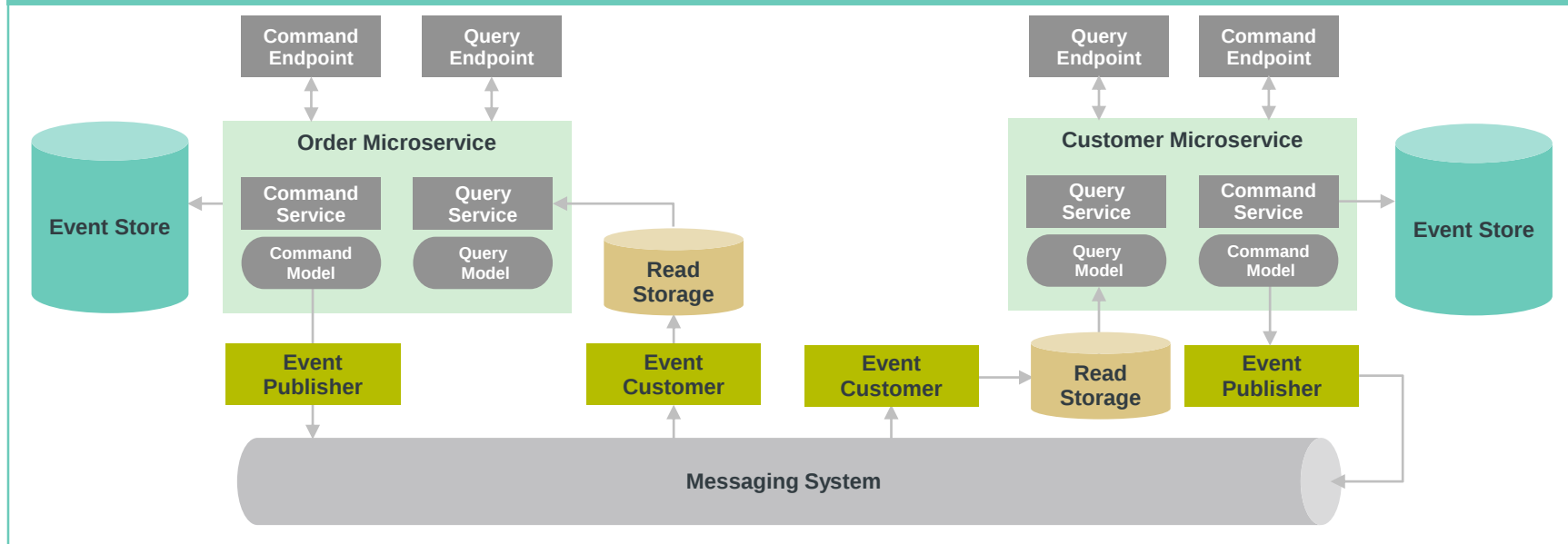
History of Data Segregation within Apps



CQRS for Operational Systems

Realtime, Ubiquitous, Distributed Data Architecture Achieved through DDD
and delivered through Kafka

Microservices with CQRS and Event Sourcing

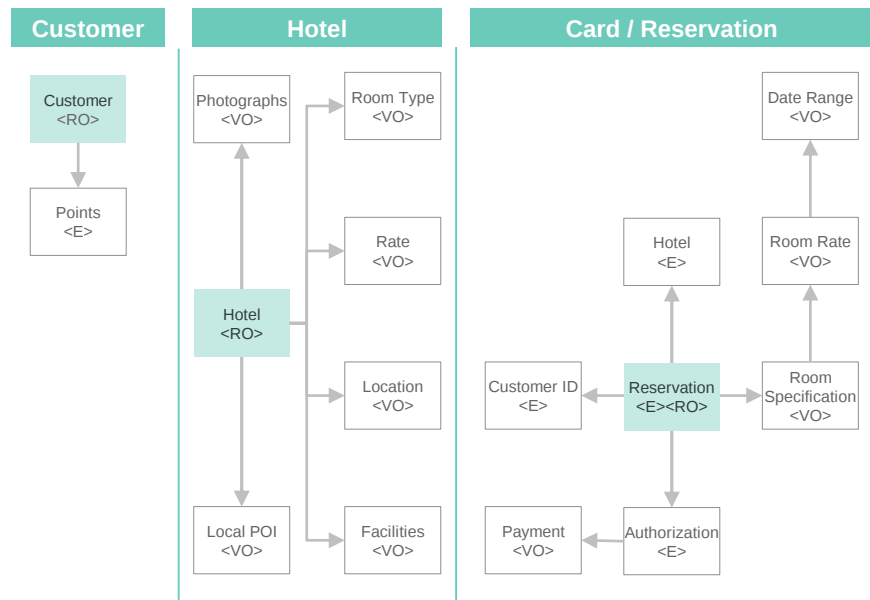


DDD for a Single Application – Generic Example

Use Case Table for a Hotel Booking System

ACTOR	USE CASE	INPUT	OUTPUT
Customer	Search Hotels	Location, Date, Range, Room Rate, Facilities	Hotels, Availability
	View Hotel Info	Hotel	Facilities, Room Specifications
	View Rates	Hotel, Room Specifications, Date Range	Room Rates
	Make a Reservation	Hotel, Room specification, Date Range, Customer, Customer Card Details	Reservation, Room availability
	Search for a Reservation	Reservation ID	Reservation
	Cancel a Reservation	Reservation ID	Confirmation Number
	View Reservation	Reservation ID	Reservation Details
	Edit Reservation	Reservation Confirmation Number	New Confirmation Number
	Create Rewards Account	Customer ID	Rewards Info
	Create Account	Customer Info	Customer Object ID
Hotel Manager	Manage Hotel Reservation	Reservation ID	Reservation
Card Vendor	Authorize Card	Customer Card Details	Card Authorization

Bounded Contexts/Data Sets



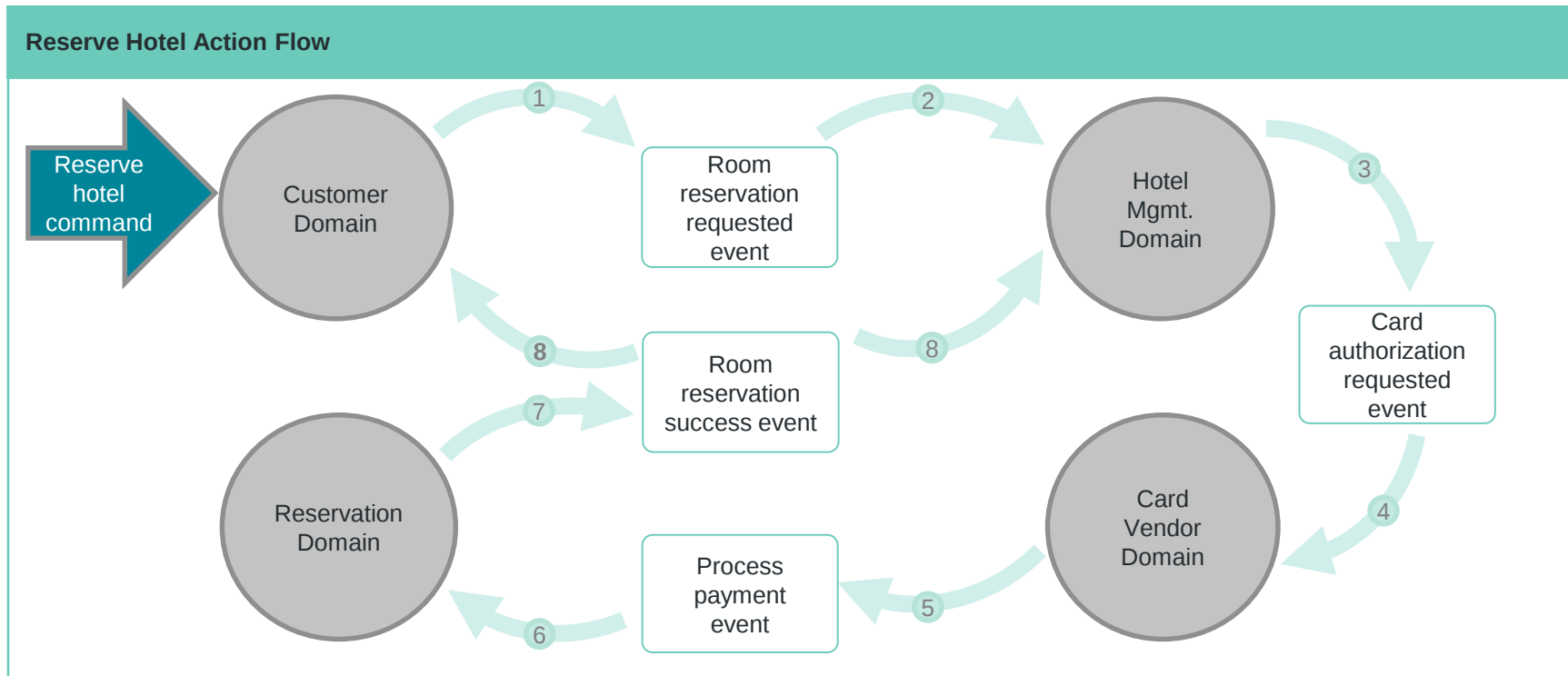
DDD merged with Event Storming

Use Case Merged with Event Storming Table

ACTOR	USE CASE	INPUT	OUTPUT	EVENT	COMMAND	QUERY
Customer	Create Customer	Customer Info	Customer	Customer Created	Create Customer	
	Edit Customer	Customer	Customer	Customer Updated	Update Customer	
	Make a Reservation	Hotel, Room specification, Date Range, Customer, Customer Card Details	Reservation, Room availability	Room Reservation Requested, Card Authorization Requested, Payment Processed, Payment Processing Failed, Reservation Succeeded, Reservation Failed	Reserve Hotel	
	Search for a Reservation	Reservation ID	Reservation	No event		Search Reservation
	Cancel a Reservation	Reservation ID	Confirmation Number	Room Cancelled	Cancel Room	
	View Reservation	Reservation ID	Reservation Details	No event		View Reservation
	Edit Reservation	Reservation Confirmation Number, Reservation	New Confirmation Number	Room Reservation Updated	Update Reservation	
	Create Rewards Account	Customer ID	Rewards Info	Rewards Account Created	Create Rewards Account	
	Send Email	Customer, email content	Status		Send Email	
Hotel Manager	Manage Hotel	Hotel Info	Hotel	Hotel Added	Add Hotel	
	Manage Hotel	Hotel	Hotel	Hotel Updated	Update Hotel	
	Manage Hotel	Hotel	Hotel	Hotel Deleted	Delete Hotel	
	Manage Hotel	Hotel ID, Picture	Status		Add Picture	
	Manage Hotel	Hotel ID, Picture ID	Status		Delete Picture	
	Search Hotels	Location, Date, Range, Room Rate. Requirements	Hotels, Availability	no event		Search Hotel
	View Hotel Info	Hotel	Facilities, Room Specifications	no event		View Hotel Information
	View Rates	Hotel, Room Specifications, Date Range	Room Rates	no event		View Hotel Rates
Card Vendor	Authorize Card	Customer Card Details	Card Authorization	Card Authorized Card Authorization Failed	Authorize Credit Card	
	Issue Refund	Customer Card Details	Card Authorization	Card Refund Authorized Card Refund Authorization Failed	Refund Credit Card	

Eventing Diagram

Flow of Events



Enterprise Domain Driven Design

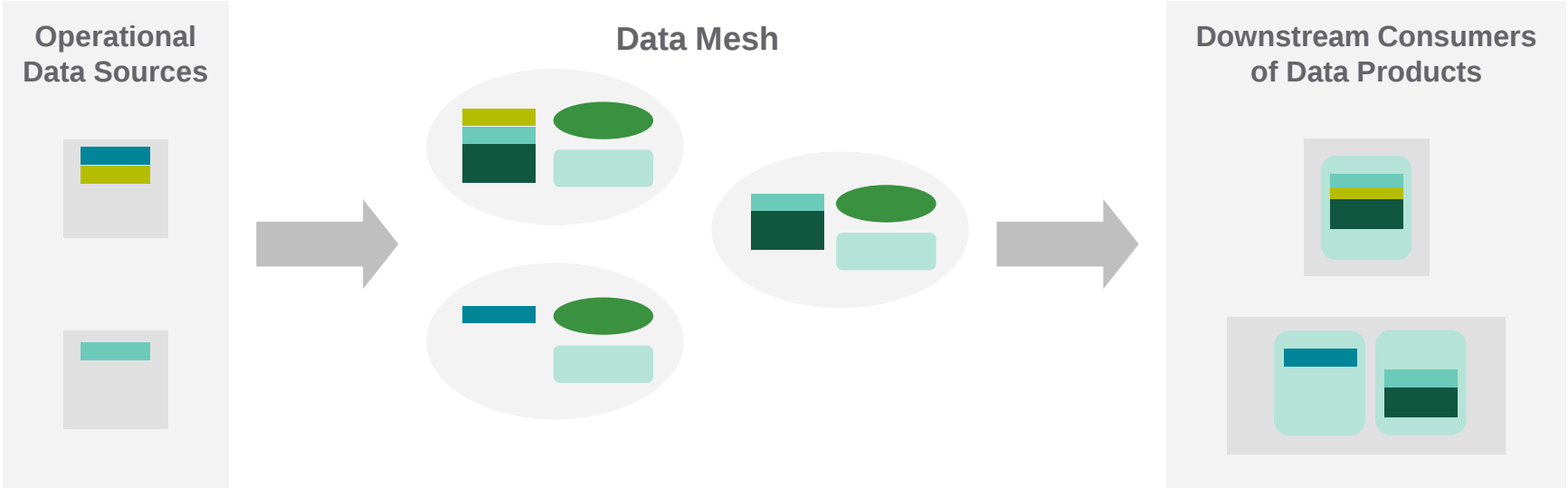
How to apply DDD to the Enterprise – Reuse of Datasets

Enterprise Domain Driven Design:

- Ubiquitous, Distributed Data across many systems working in concert
- Governance: Glossary that logs data set and metadata created by DDD
- Each bounded context should be listed along with its metadata:
 - Lineage
 - Business Data Owner
 - Data Steward
 - Technical Owner
 - Interoperability
 - Distributions
- Operational System needs
- Analytics systems needs

Operational vs Data/Analytics Ecosystems

Using Data Sets in your Analytics Ecosystem



Key

Data Set

Data Set

Data Set

Data Functionality

Compute

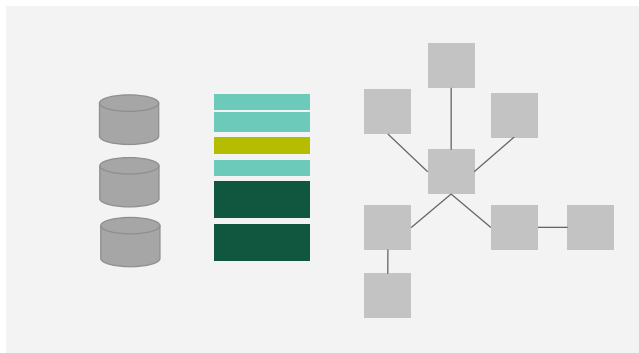
Delivery

App

Monoliths vs Data Mesh

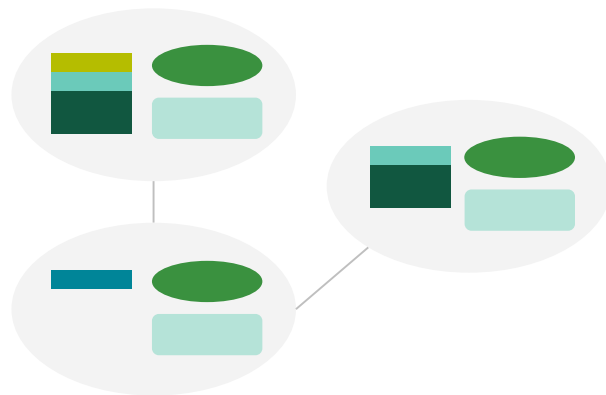
Monoliths

- Centralized data ownership within tech-focused, specialized team.
- Highly-coupled data pipelines; difficult to change.



Data Mesh*

- Domain-oriented data
- “Data as a product”
- Self-service data infrastructure
- Standards and governance



*Reference: Zhamak Dehghani “How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh”;
<https://martinfowler.com/articles/data-monolith-to-mesh.html>

Recap

- Advancing Domain Driven Design for one application into Enterprise Domain Driven Design.
- With Enterprise Domain Driven Design, we can merge with the data side of the house.
- Domain Driven Design for the data warehouse / analytics environment.
- Unify operational and analytical worlds around business-driven data domains.

Questions?



NORTHERN
TRUST