Scaling Domain Driven Design

Greg Young IMIS

Presentation

Is

- Discussion of DDD being used in a large scale system
- Discussion of concepts that can be applied to other domains to solve common infrastructure issues

Is Not

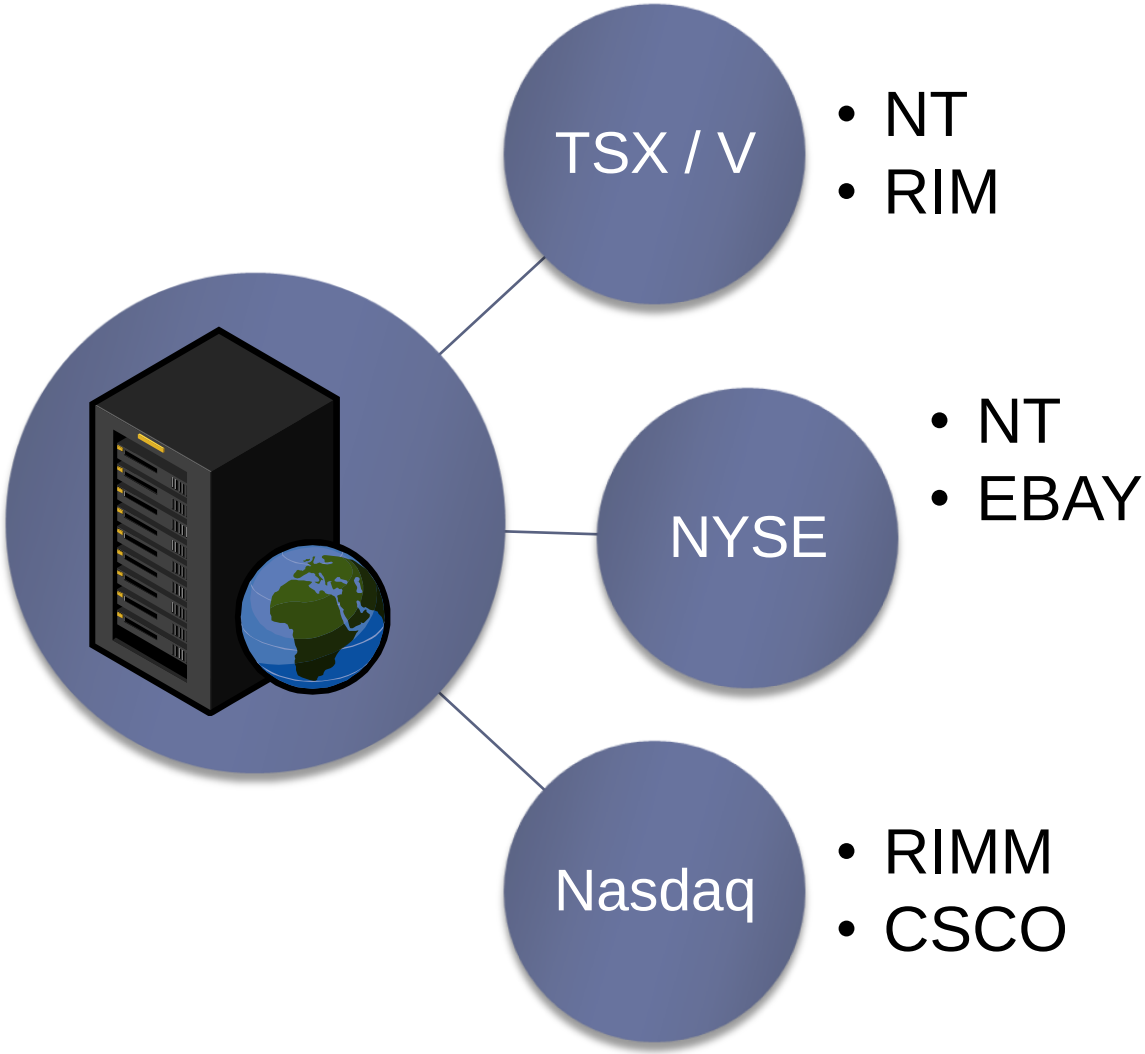
- A DDD Tutorial
- A Scalability Tutorial

Context

IMIS


Algorithmic Trading
Regression Analysis
Data Warehousing

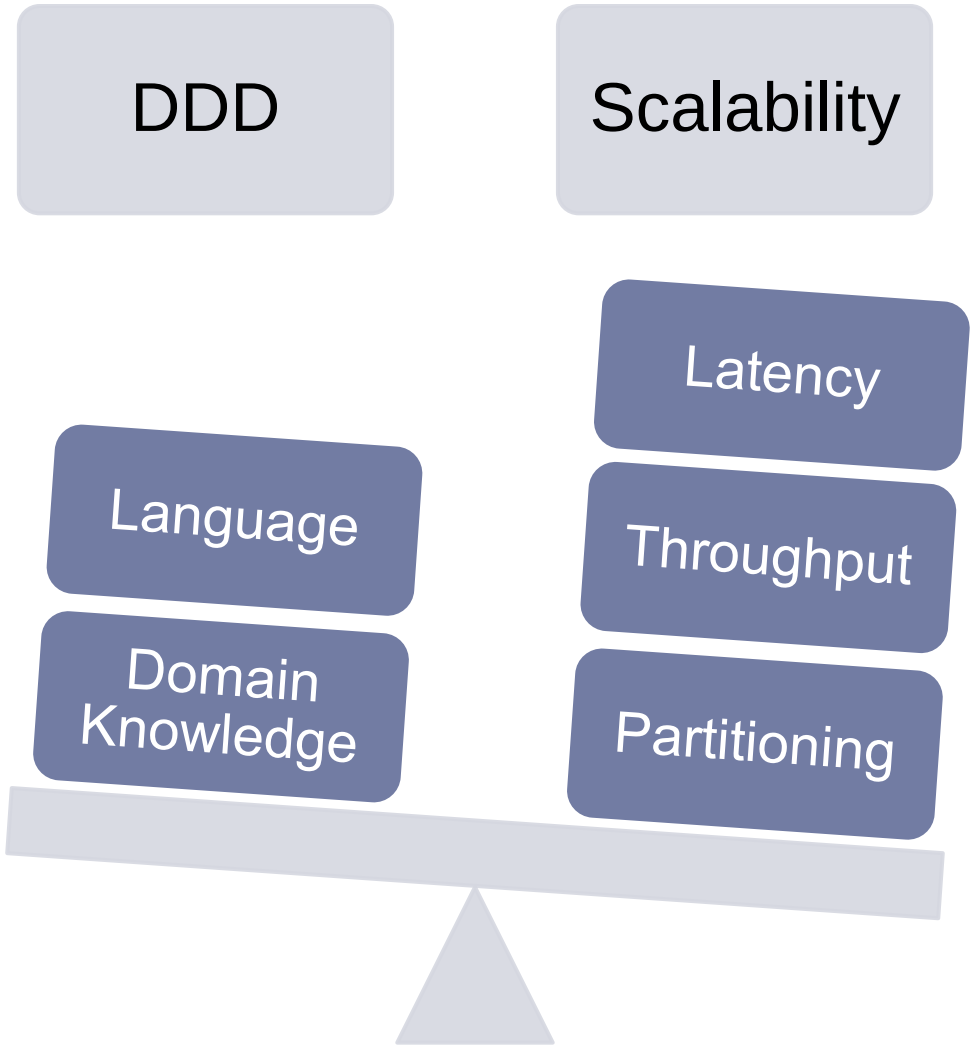
Incoming Data



Challenges

- Large Amounts of Complex Business Logic
- Large Amounts of Historical Data
- Large Amounts of Real-Time Data
- Nearly all Real-Time Data is Stateful
- Extremely Low Latency (< 100 ms)

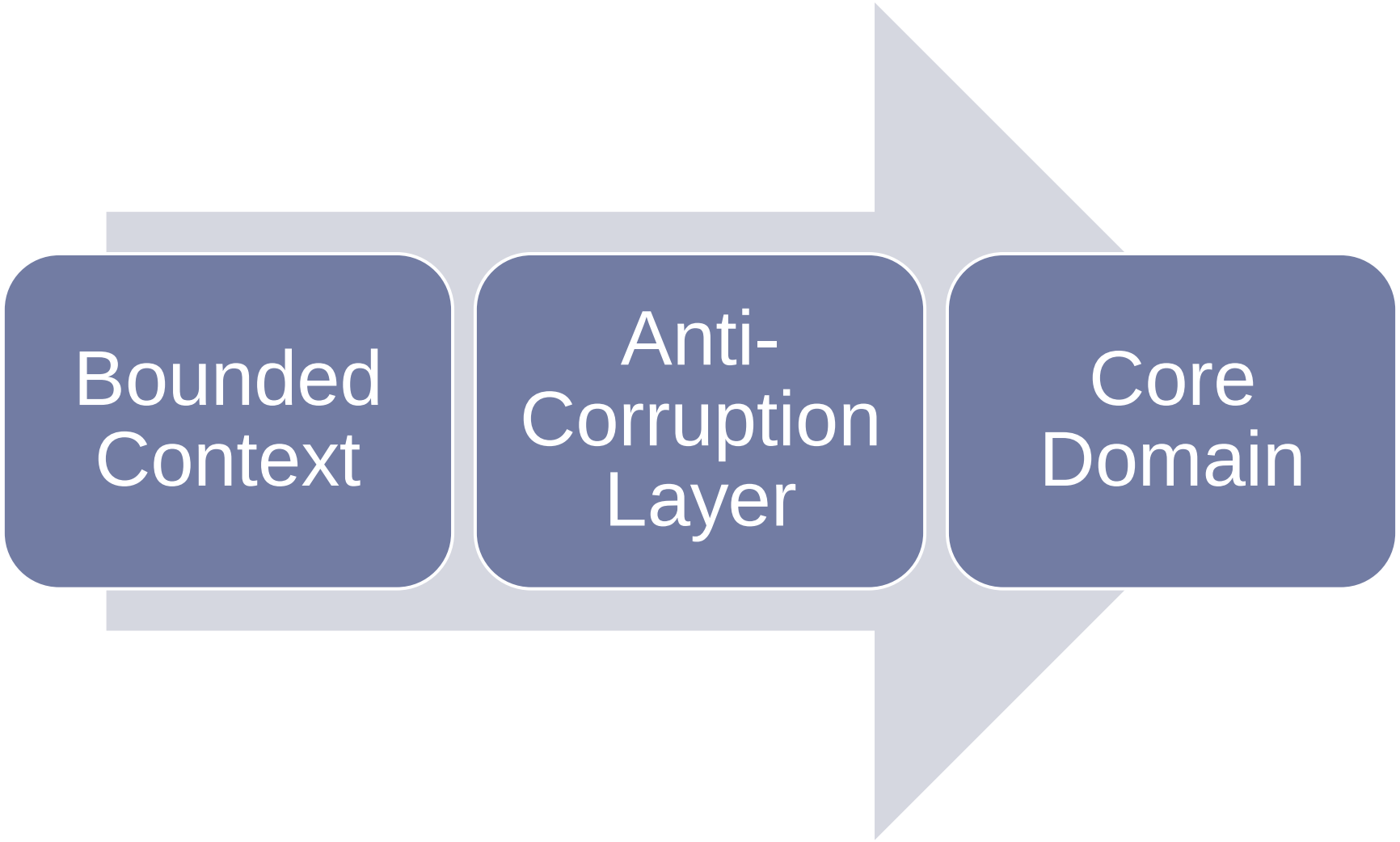
DDD & Performant Systems


DDD
Scalability
Latency
Language
Throughput
Domain Knowledge
Partitioning

Bounded Contexts

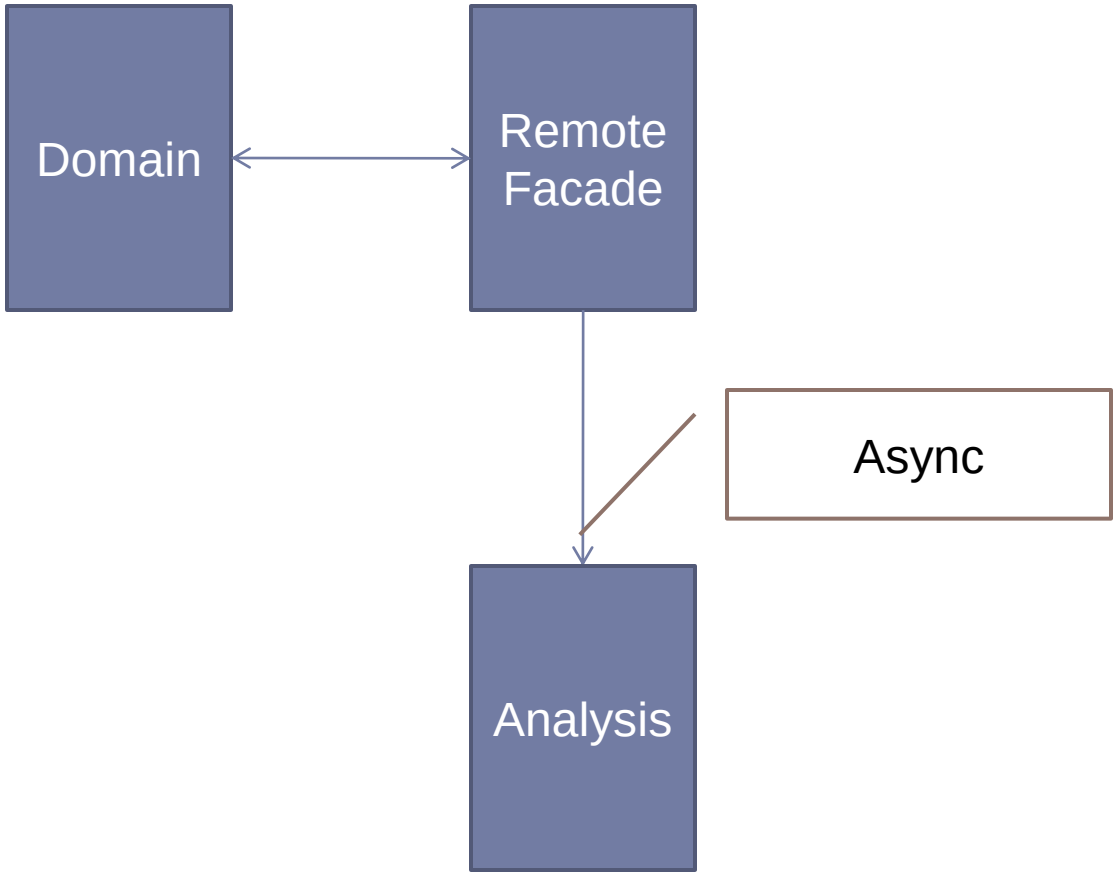
- Language Specialization
- Knowledge Hiding
- Team Partitioning
- Scalability??

Context Mapping



Many Bounded Contexts do not need real time data…

First Attempt



Analysis

Pros

- Easily Scalable
- Simplicity
- “Retrofitability”

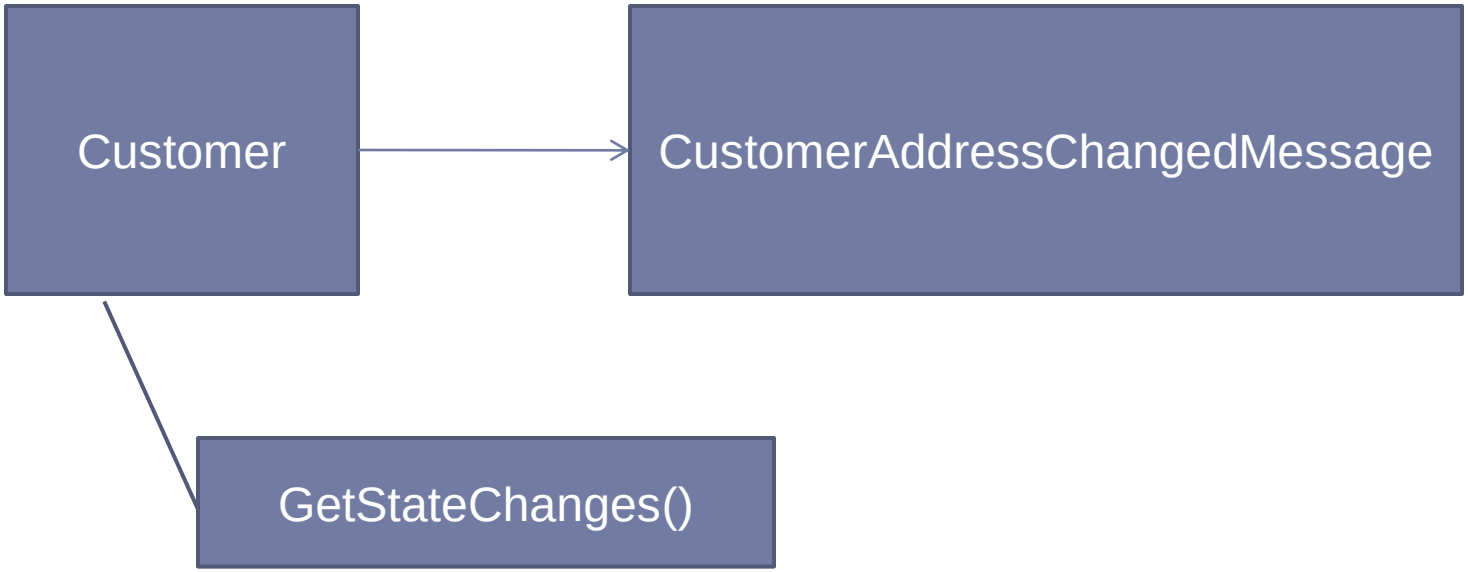
Cons

- Fractured Domain/ Duplicated Logic
- Wrong Language
- Too Much Information

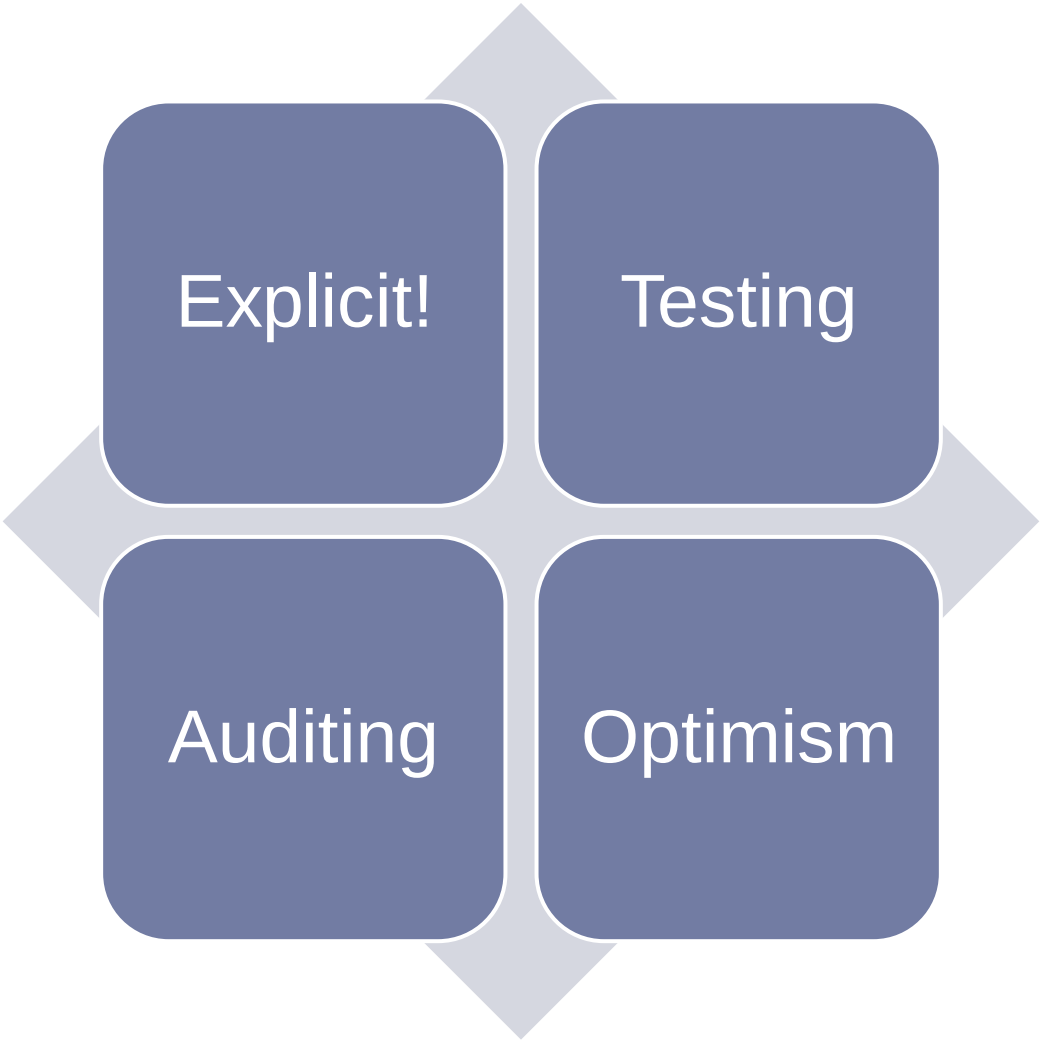
If things seem hard, there is something wrong with the model…

Are state changes domain concepts?

Changes are First Class Citizens



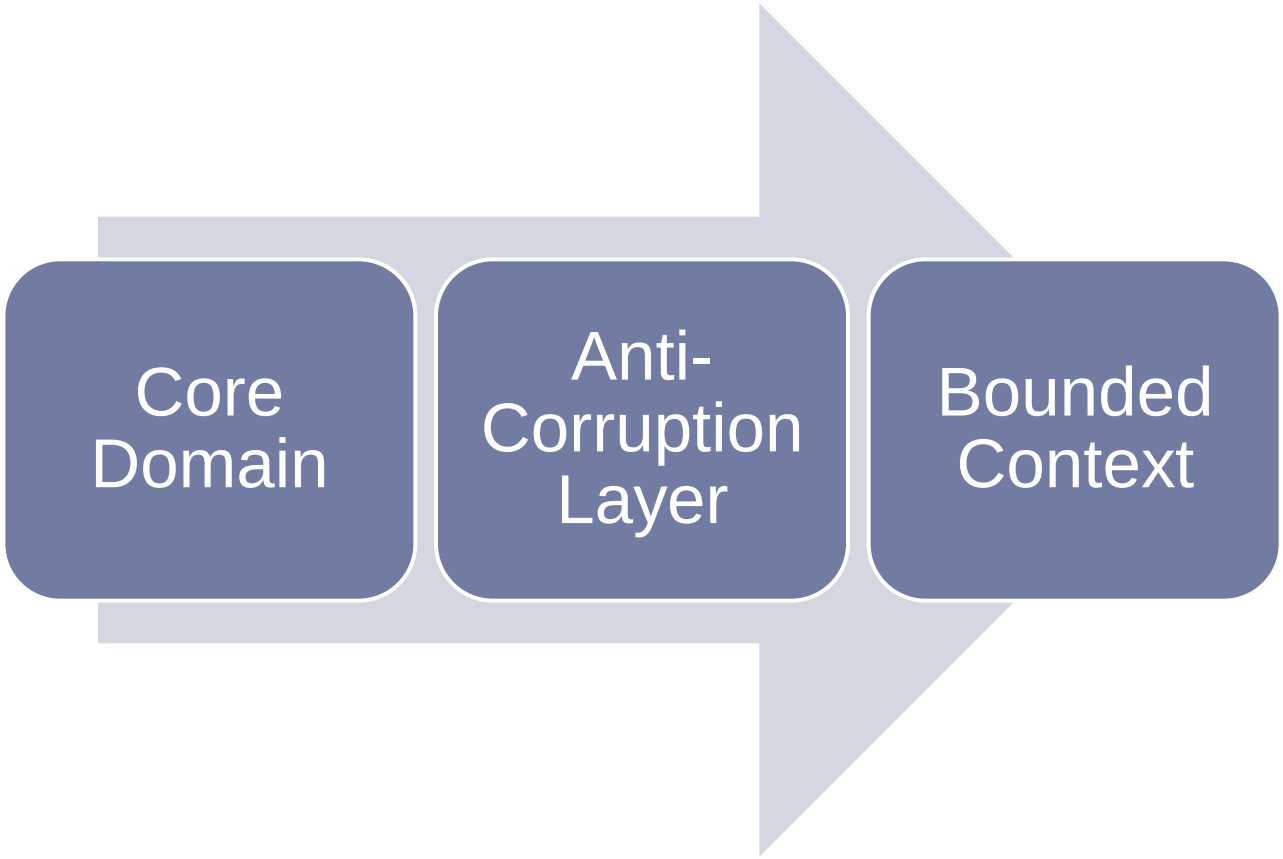
Benefits



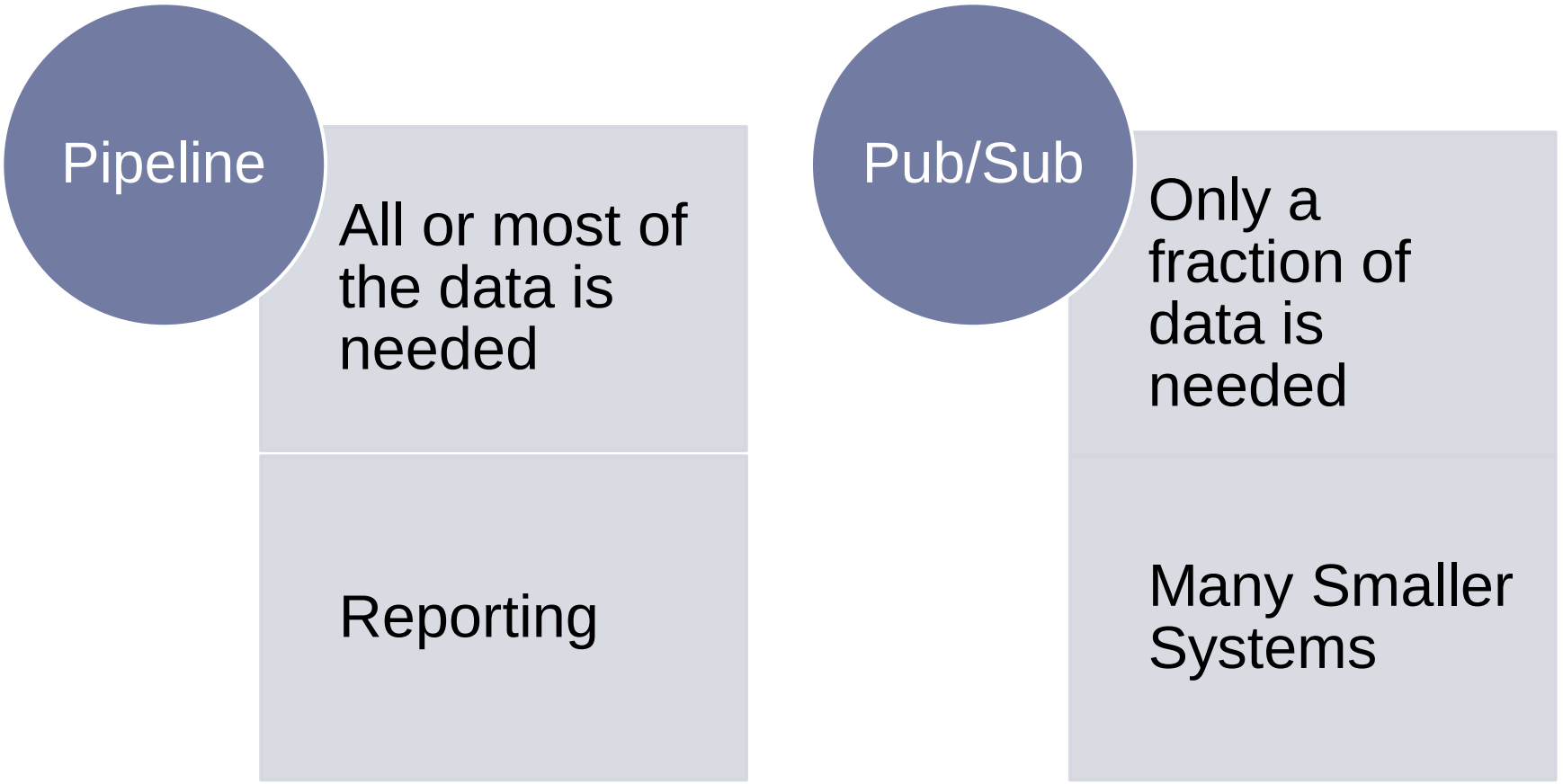
Annoyances

- Creating all of the messages (especially creates)
- Not easily applied to an existing domain

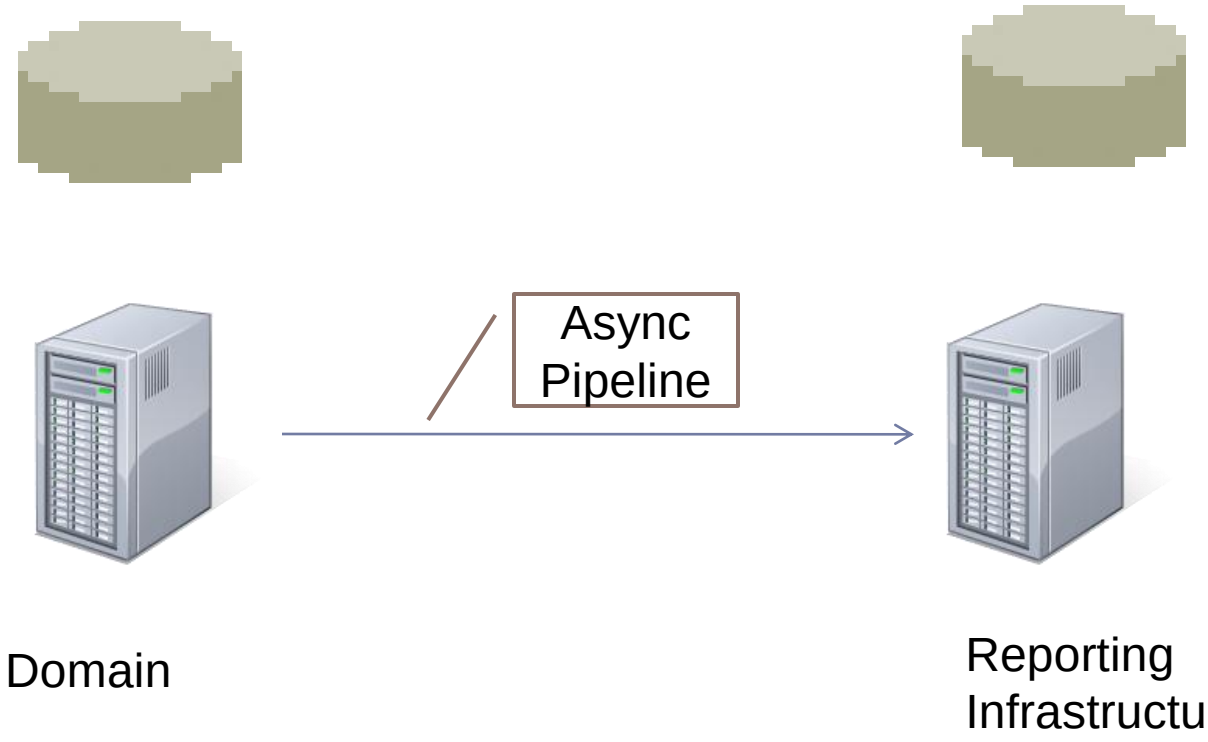
Reversed



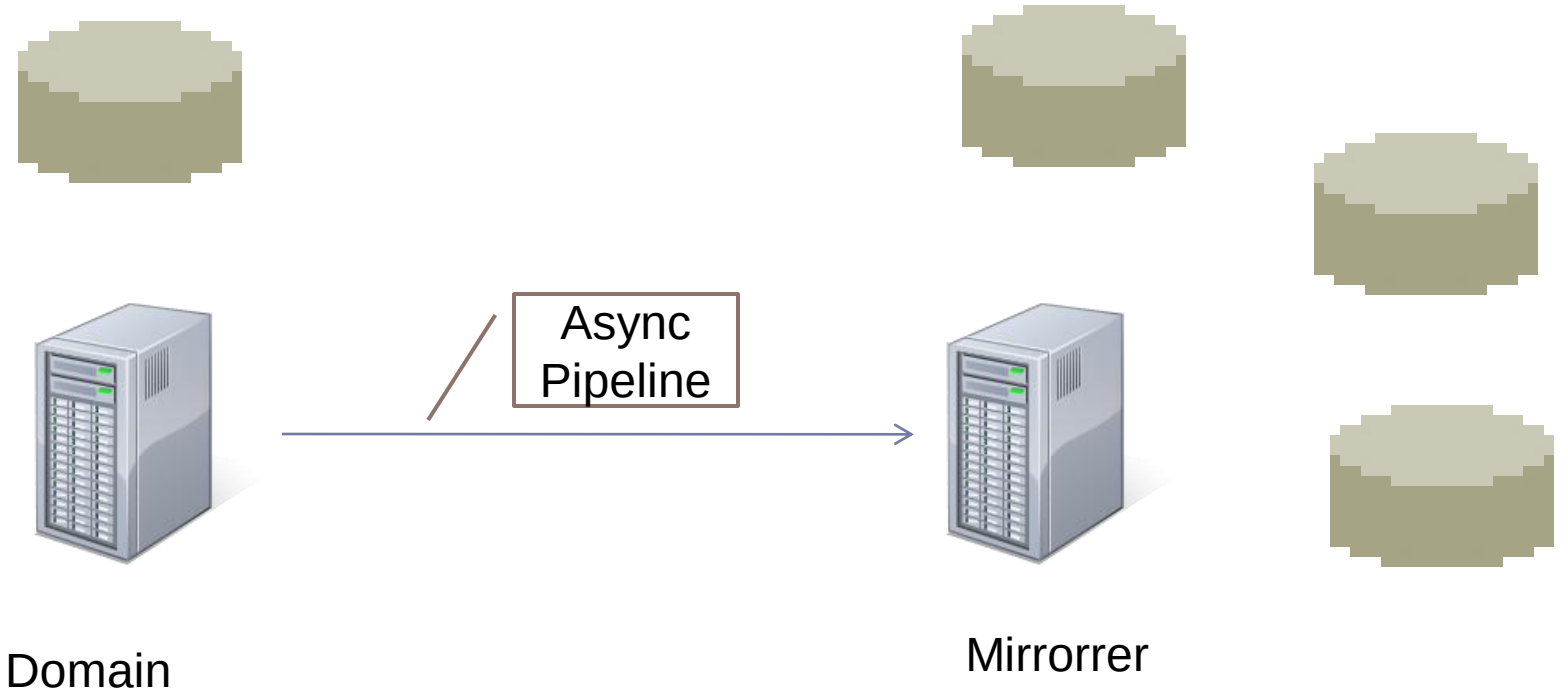
How to send the state changes?



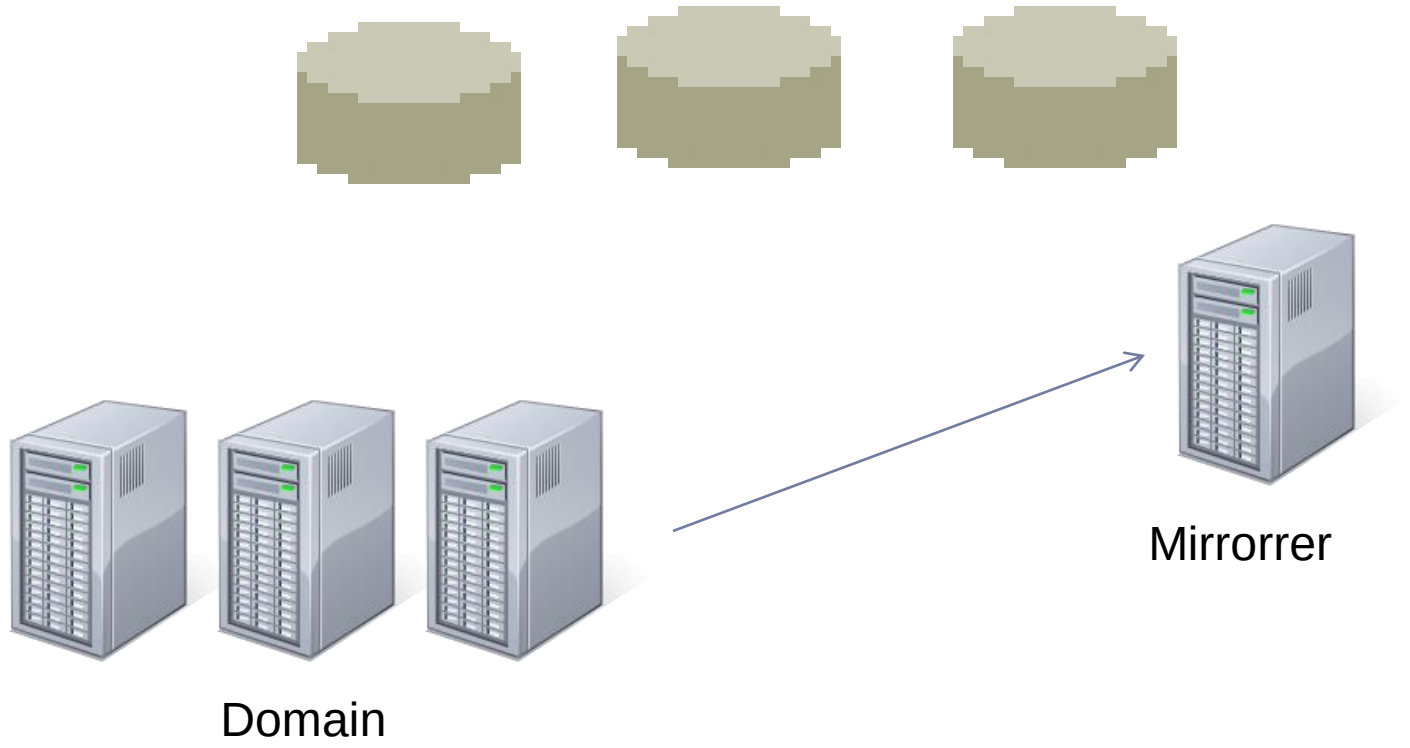
Reporting



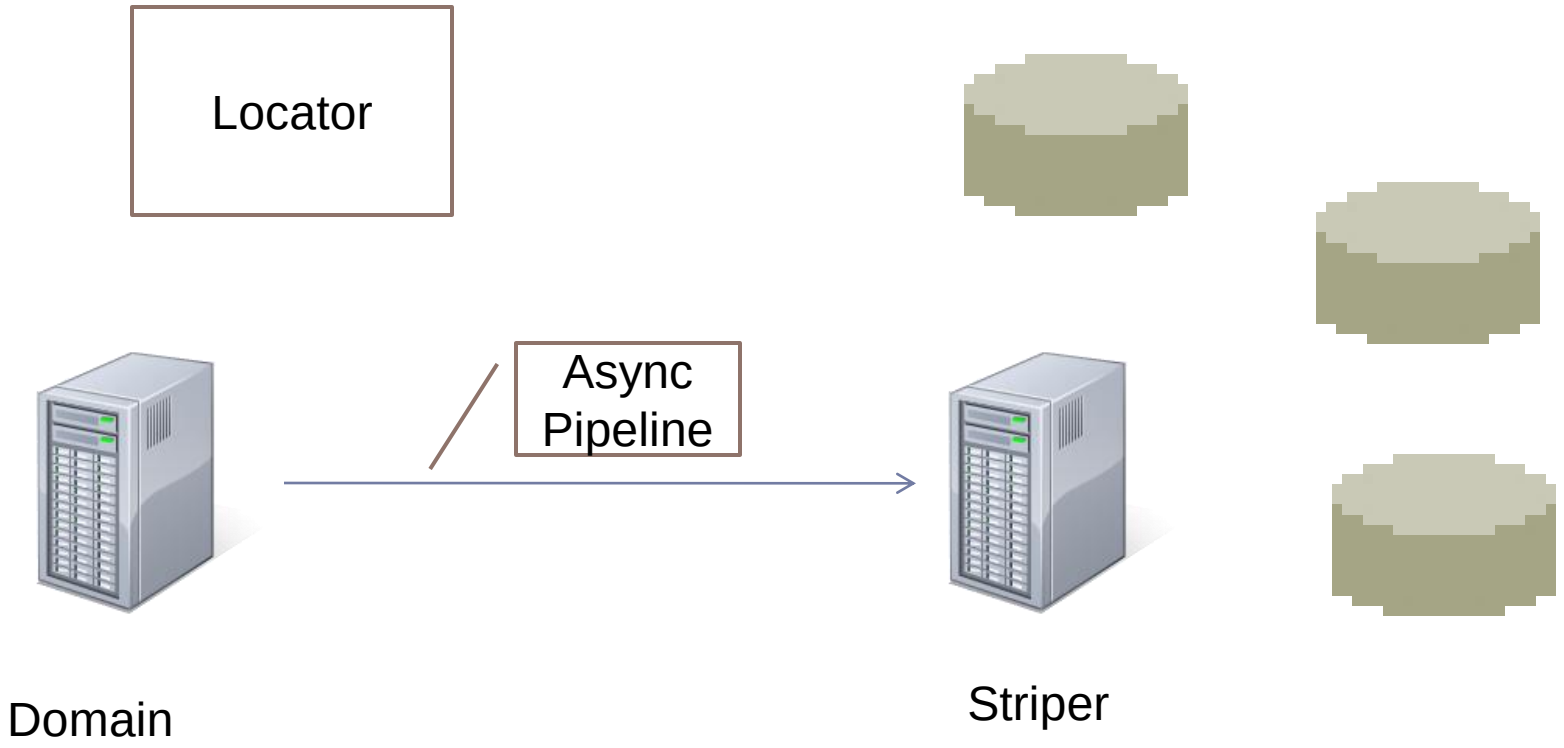
Warm/Hot Replication



Load Balancing w/ Optimistic Locking



Striping (Partitioning)



Summary

Summary

- Bounded Contexts are Also for Scalability
- Most Context Mapping Need not be Synchronous
- State Changes are First Class Citizens

Questions