

# Getting started with DDD when surrounded by Legacy Systems

Revisiting a 2013 paper by Eric Evans

Christoph Baudson

# Strategie 1 BUBBLE CONTEXT

Legacy ablösen + DDD  
=> keine gute Idee  
=> inkrementell arbeiten!

☺☺☺  
auch Team-Erfahrungen sind wichtig!

Legacy-Modell schwer änderbar

! Neues Modell könnte Legacy schaden

DDD braucht einen sauberen Bounded Context!

! Kern **ACL**

und dann?  
EXPANDING A BUBBLE

XY → AB → **GEFAHR**  
Es wird ein riesiger Übersetzer gebaut

Neue Daten = Aufwand

• Wenige Daten übernehmen  
• Daten übersetzen  
• Legacy-Erweiterungen schwierig

Wirklich nötig? → Schwieriger als neue Anforderungen

Eine Blase kann platzen  
→ Code ins Legacy  
→ Erfahrungen

Legacy-DB mitnutzen  
→ kein Synchronisationsproblem

Pattern: **ACL-BACKED REPOSITORY**  
kein eigener Datenspeicher  
wenn man nur wenige Daten braucht.

## GETTING STARTED WITH DDD ... when surrounded by Legacy Systems

@ewolff 14.07.2020  
SOFTWAREARCHITEKTUR im STREAM

€ nicht zu viel Geld in den Übersetzer

ACL erweitern = eigener Entwicklungstask

! nur Daten holen, die man wirklich braucht!

Macht man einen guten Job, so schreibt man ein Legacy-System ☺

# Strategie 2 AUTONOMOUS BUBBLE

⌚ Kann einige Zeit unabhängig von anderen Systemen laufen

Evolution des Datenmodells ist loser gekoppelt

mehr Aufwand, mehr Autonomie

Pattern: **SYNCHRONIZING ACL**

☾ nächtlicher Batch

⚡ Domänen-event

Übersetzung

Kann sich auch aus Bubble Context entwickeln

# Strategie 3 EXPOSING LEGACY AS A SERVICE

@teapot4191

Legacy kann das schon gut, warum das nicht nutzen!

€ Du willst in die Core-Domain investieren

Legacy ist nicht gleich schlecht!!!

Spezialisierte Sprache

Published Language

Granularität: mentaler Overhead

Generic & Supporting Subdomains

Pattern: **OPEN HOST SERVICE**

Baut auf Legacy System auf

System ist mit bestimmten Feature-Set implementiert!

↳ Synchronized oder ACL backed  
↳ Side-effect free functions

🌳 Großer Abhängigkeitsbaum

Das Ziel sollte nicht sein, das Allsystem abzulösen!  
bei stabilem Legacy-System

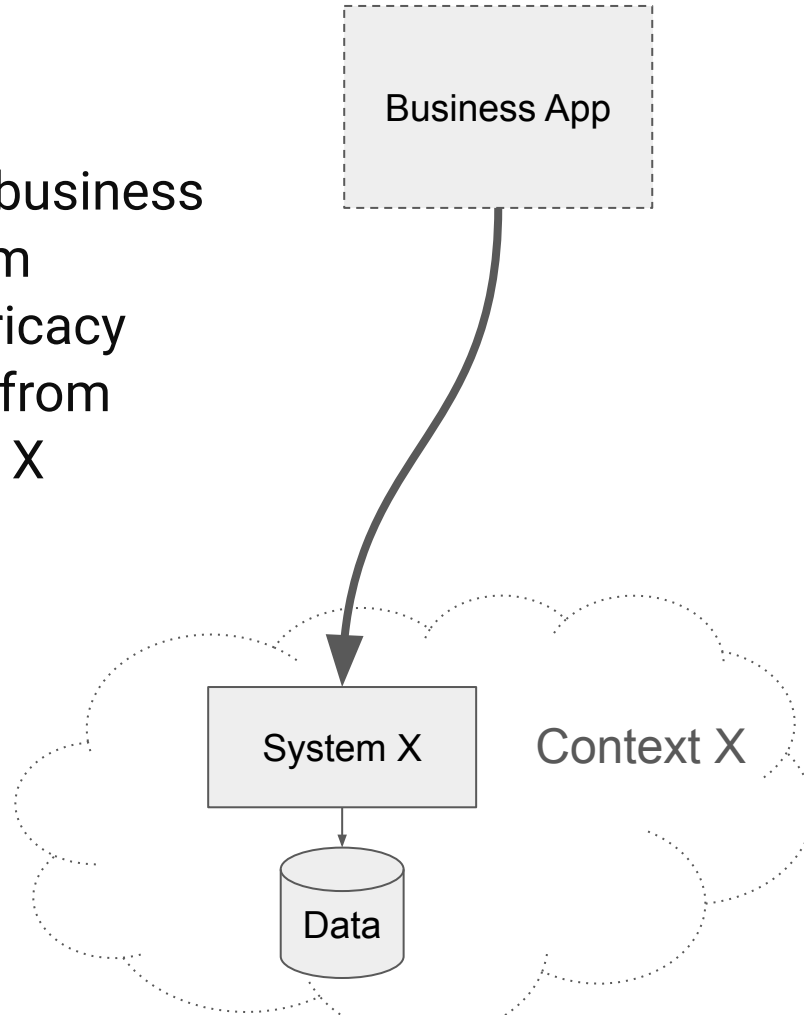
Tragedy of the Commons

# Core ideas

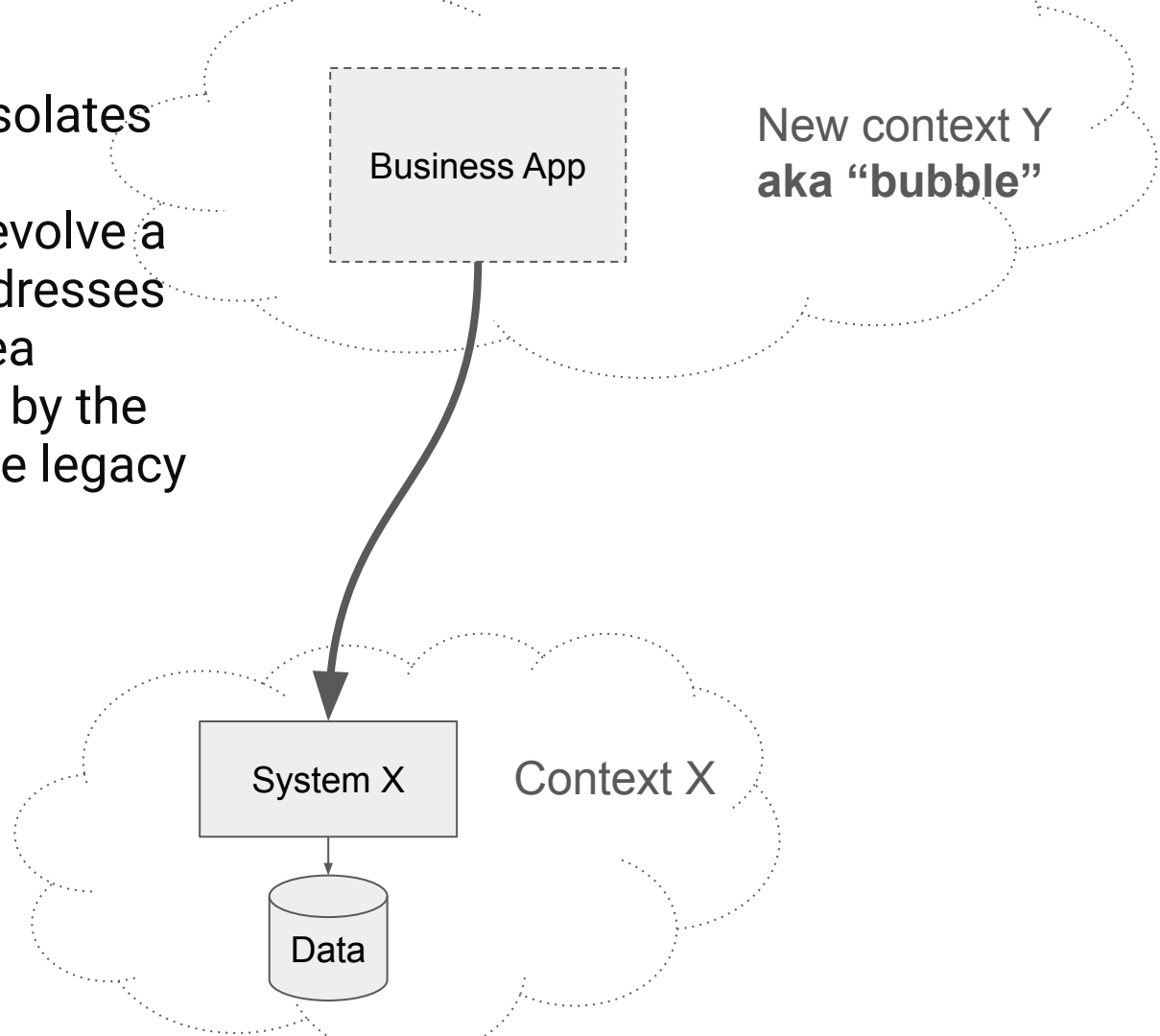
- ❑ Legacy represents value - don't fall into the trap of replacing legacy systems by default!
- ❑ Make it easy for people to apply DDD patterns in brownfield situations!
- ❑ Remember some solutions presented here are temporary or steps in an evolution!

Strategy 1:  
Bubble context

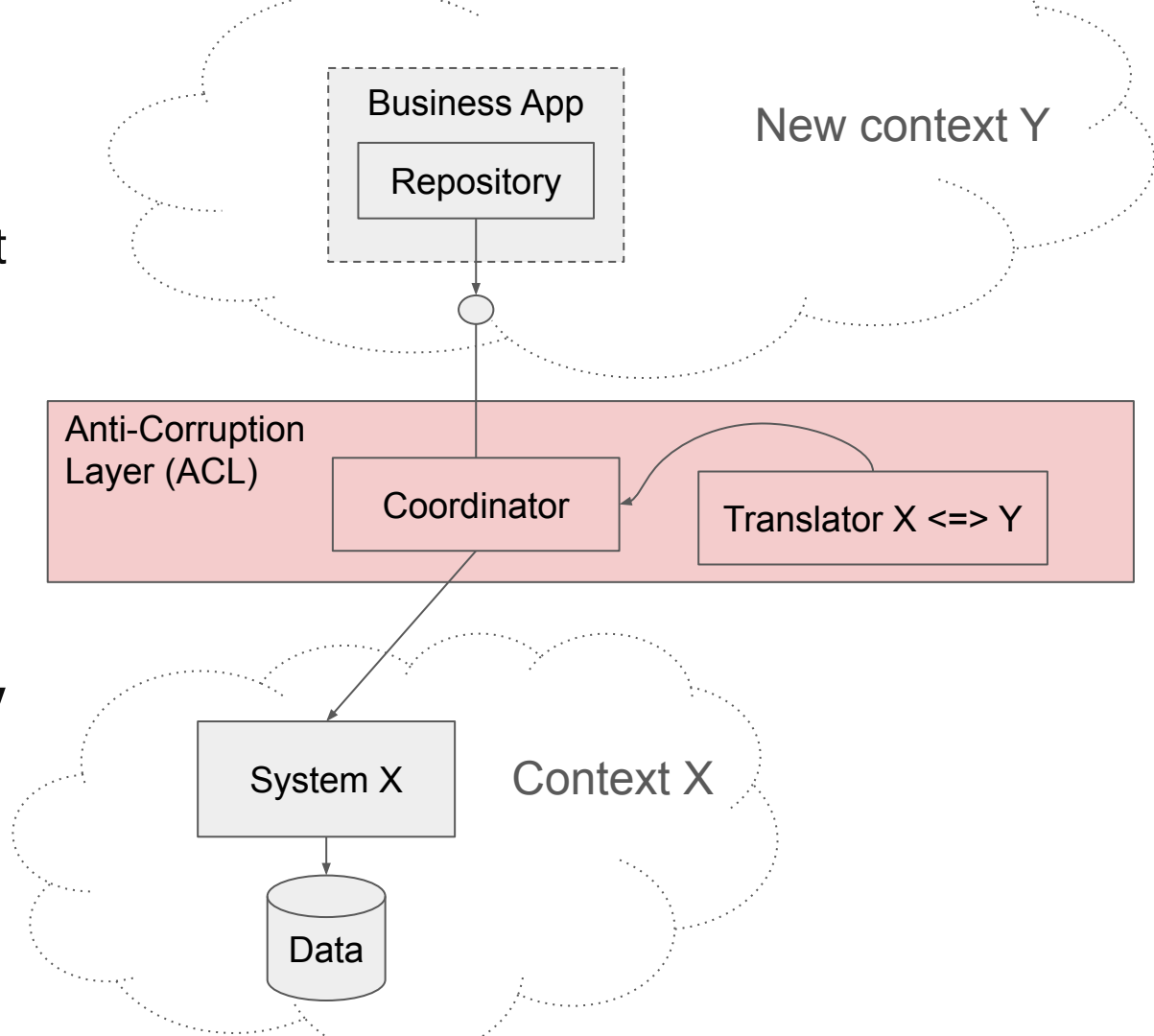
- ❑ Solution to an important, yet modest-sized business related problem
- ❑ With some intricacy
- ❑ Requires data from legacy system X



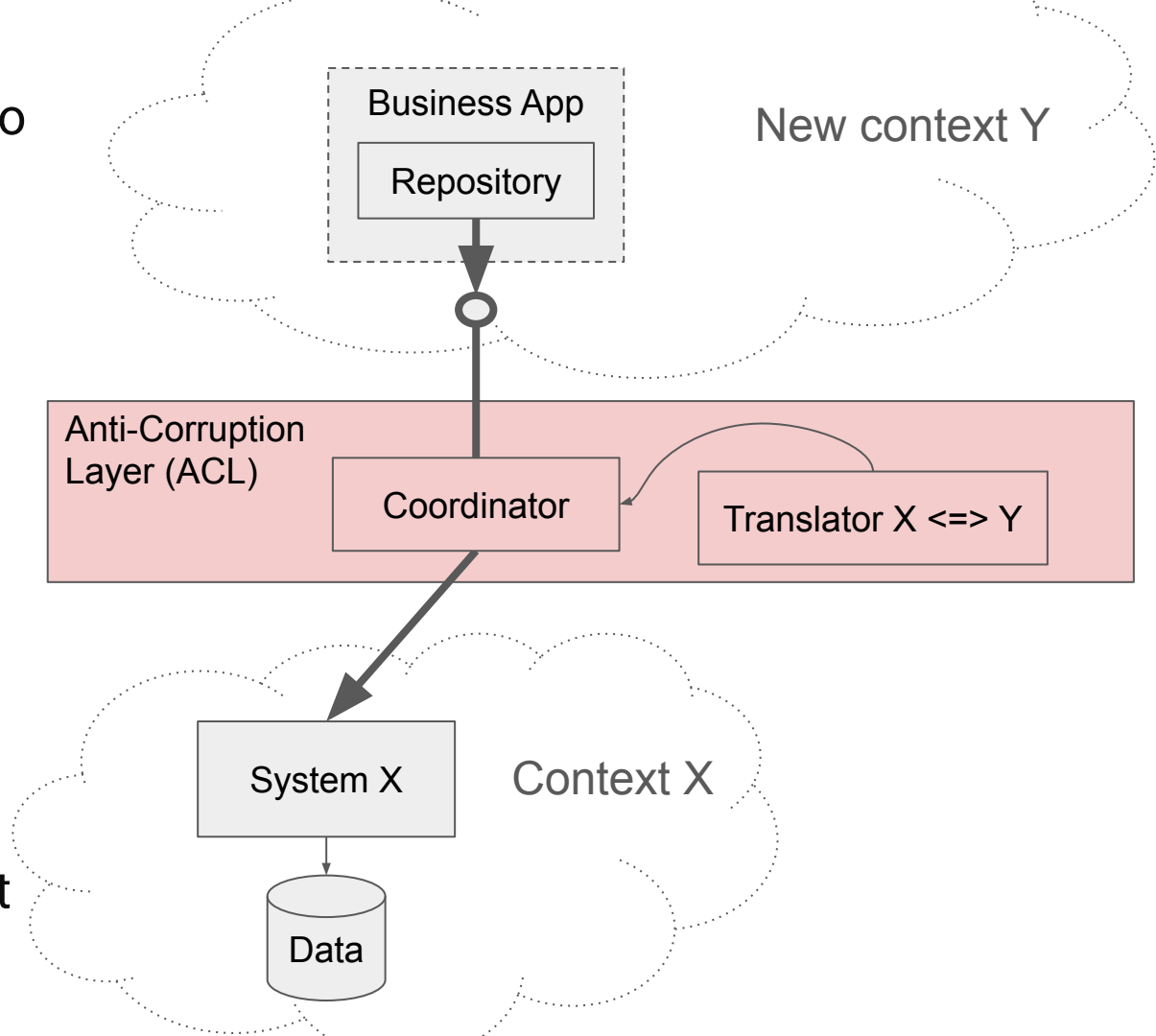
- ❑ The “bubble” isolates that work
- ❑ the team can evolve a model that addresses the chosen area
- ❑ unconstrained by the concepts of the legacy systems.



- ❑ Create the **illusion** that bubble context Y has its own data store
- ❑ No new DB in the context Y, query the legacy DB **synchronously** through ACL
- ❑ translate data in the ACL into Y's model



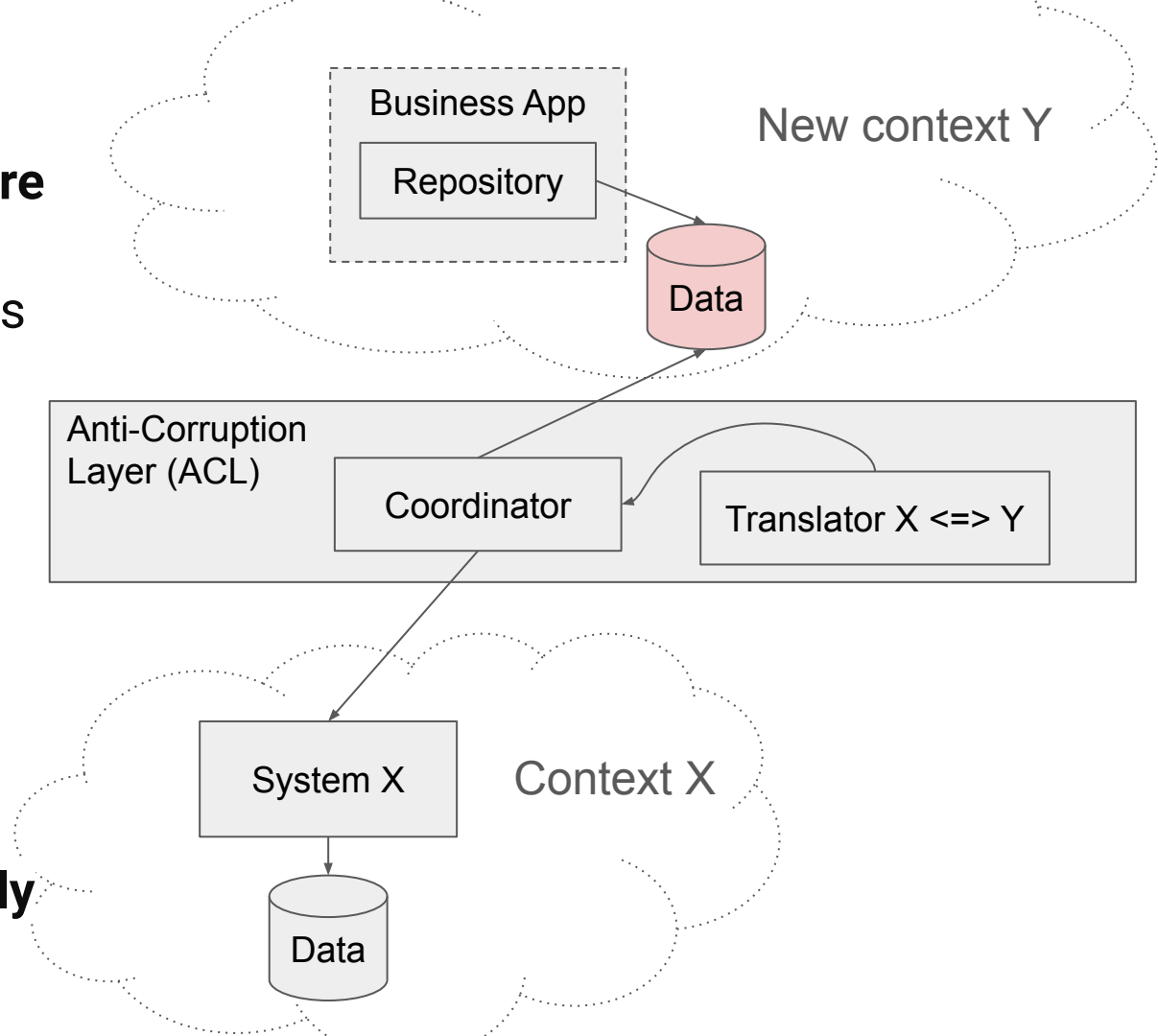
- ❑ ACL is closer to context Y than to context X
- ❑ The bubble context Y is **completely dependent** on the parent context X!
- ❑ ACL acts like an **umbilical cord**
- ❑ Context Y can't survive alone!





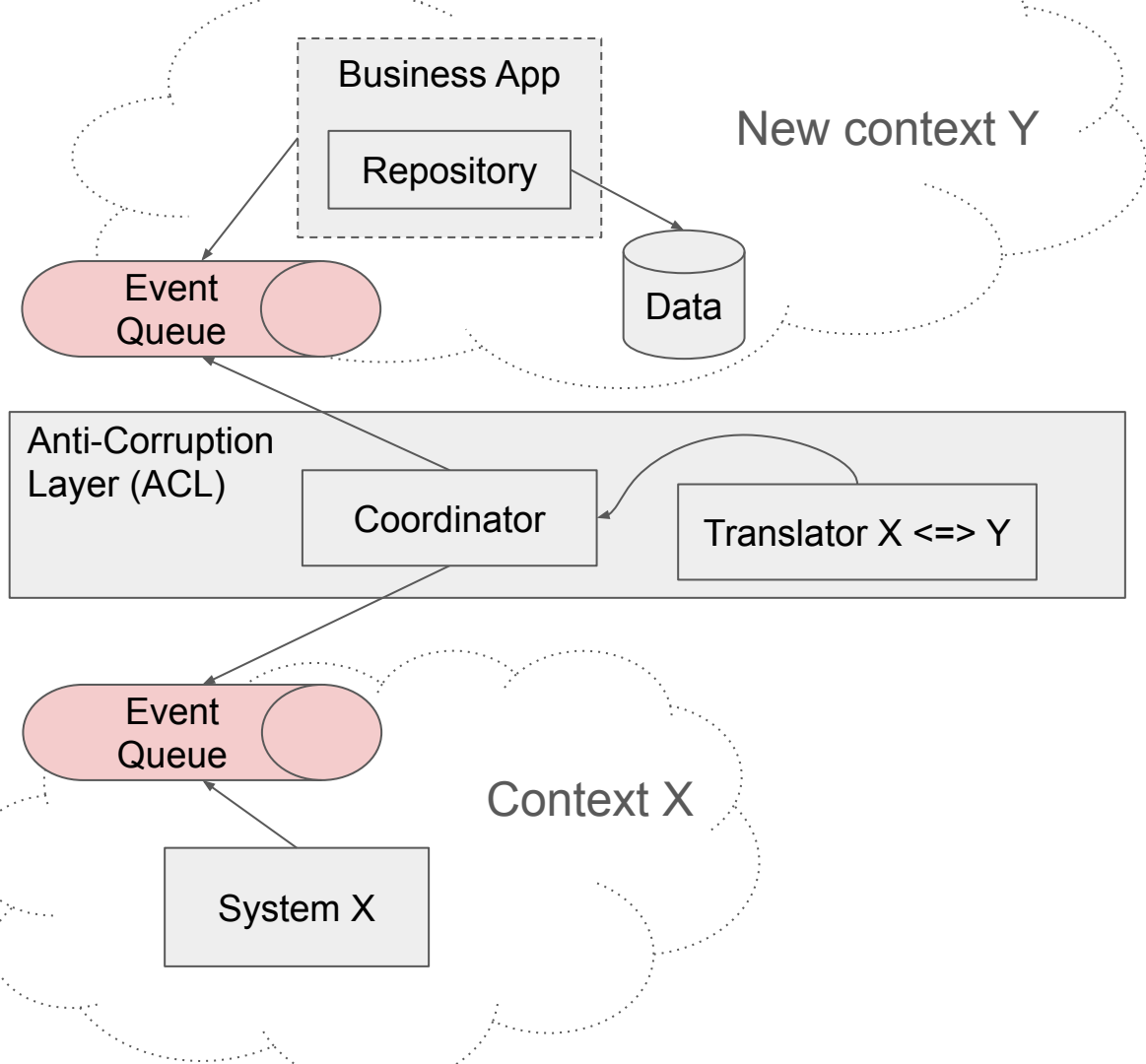
# Strategy 2: Autonomous Bubble

- ❑ Typically has some **data store of its own**
- ❑ Ability to run its software, for a time, **cut off** from other systems
- ❑ ACL syncs between data stores in two contexts **asynchronously**



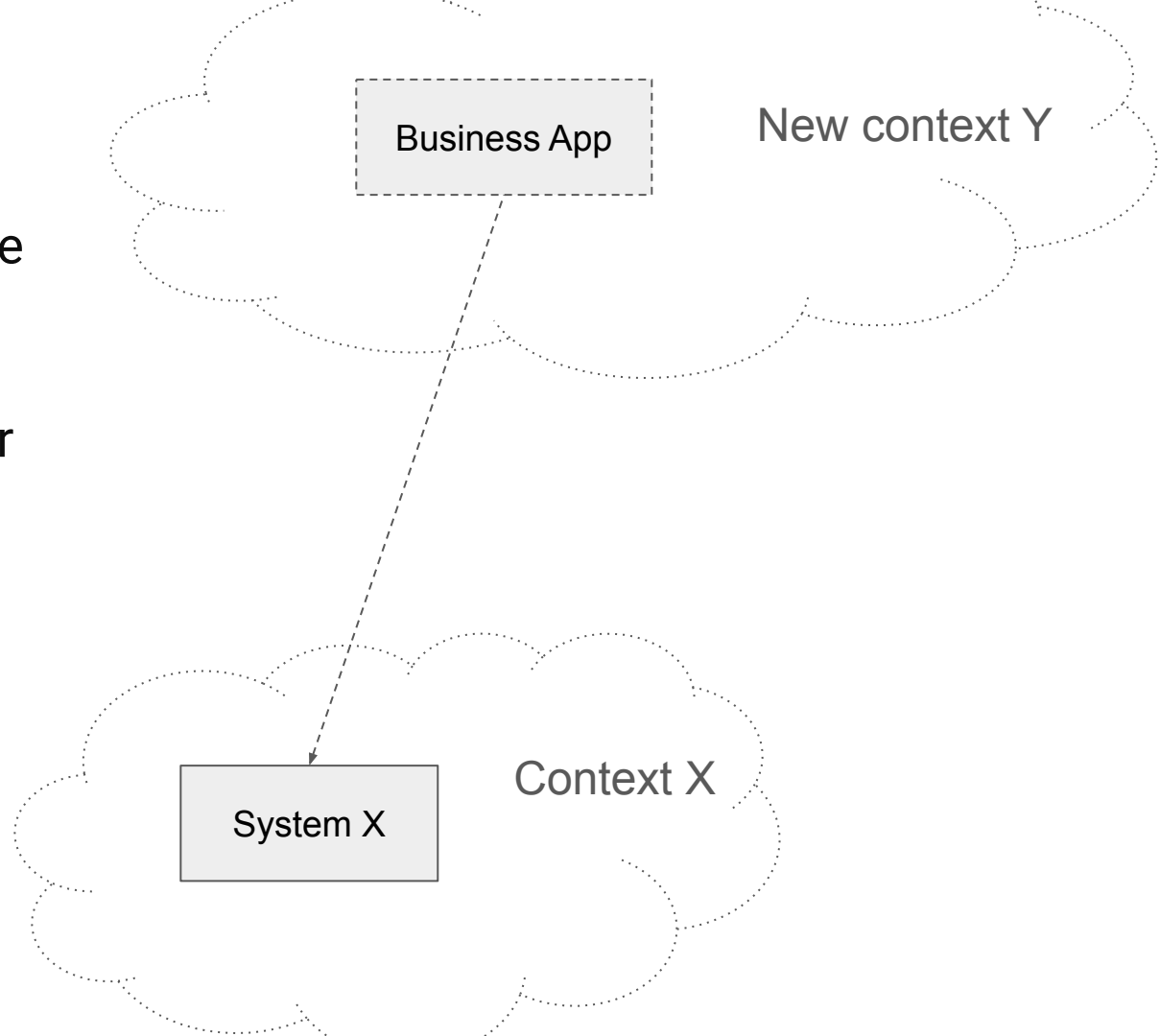
Strategy 2b:  
Autonomous Bubble  
with Domain Events

- ❑ Fully asynchronous
- ❑ **Messages are not neutral!**  
They are always expressed in some language based on a model
- ❑ **Don't let them enter a context that uses a different model**

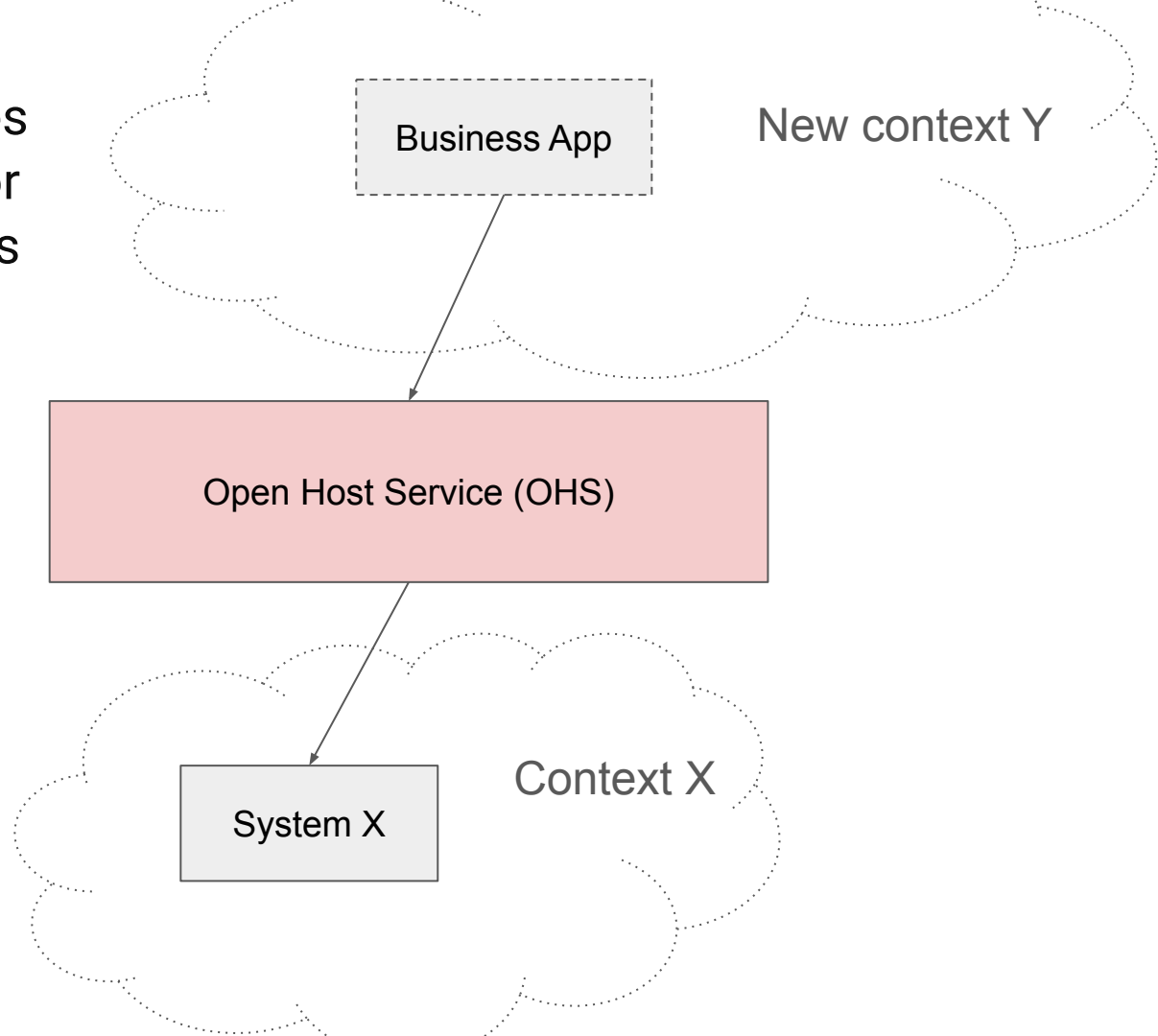


Strategy 3:  
Exposing Legacy Assets  
as a service

- ❑ New context needs data from a valuable legacy system
- ❑ Possibly complicated or inaccessible interfaces
- ❑ No value in replacing the legacy system



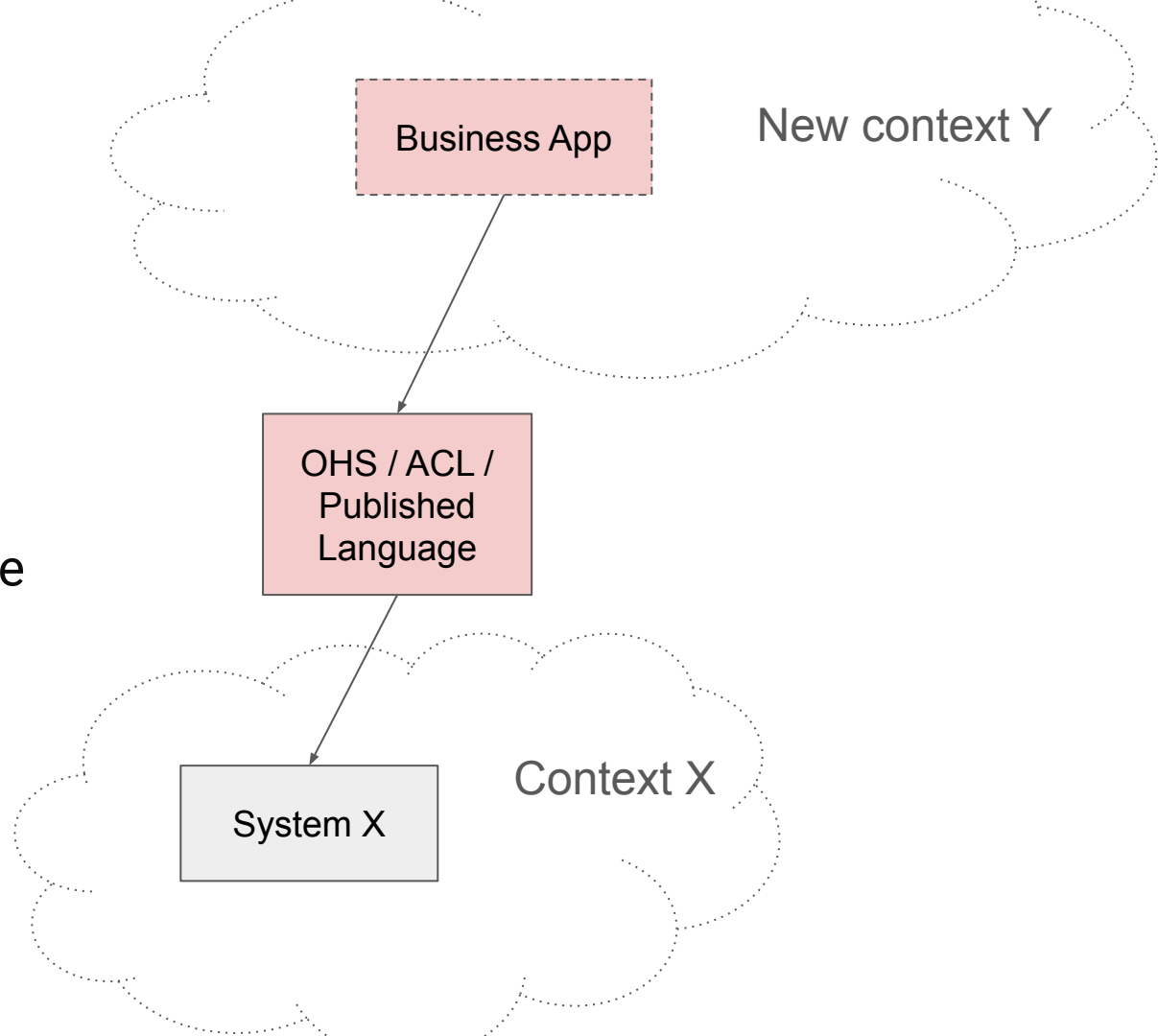
- ❑ OHS introduces an interface for newer contexts
- ❑ In an OHS, the model is more **general**
- ❑ It is closer to context X than to context Y
- ❑ Potentially introduces a **Published Language**



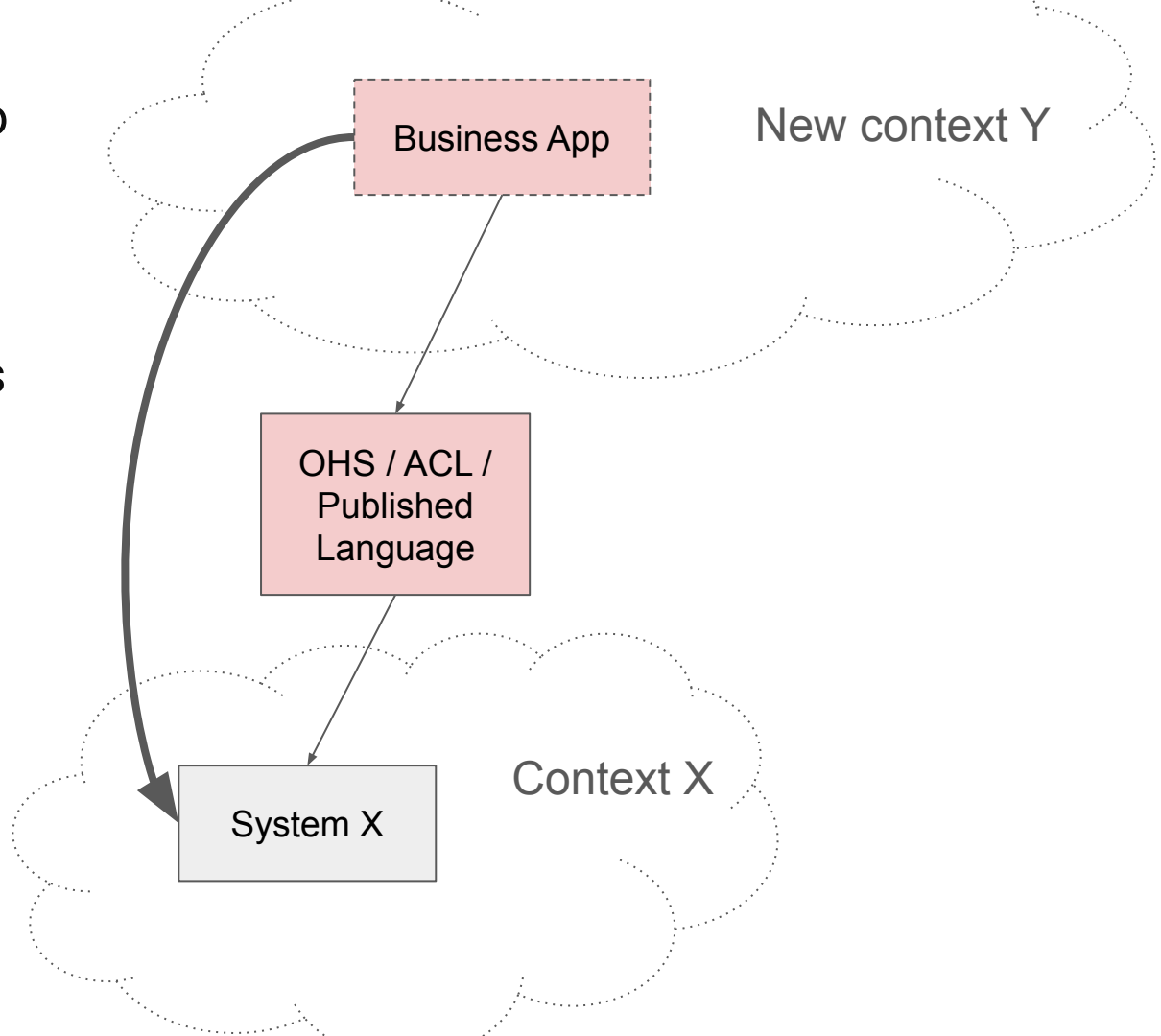
# Strategy 4: Expanding the bubble



- ❑ New context and OHS/ACL both need to evolve
- ❑ Often, the OHS/ACL is neglected and new feature are prioritised
- ❑ It isn't just "a new field" - do serious modeling



- ❑ If you fail to do this, people might take shortcuts
- ❑ this sabotages the OHS / ACL



- ❑ it also leads to eroding context boundaries

