

GOTOpia November 2020

Good Enough Architecture

Stefan Tilkov

@stilkov

INOQ

3 Statements About Architecture

**Architecture is not
an upfront activity
performed by
somebody in charge
of telling everyone
else what to do**

**Architecture is a
property of a
system, not a
description of its
intended design**

**There is no "good"
or "bad"
architecture
without context;
architecture needs
to take specific
quality attributes
into account**

Architecture Antipatterns

Unhealthy Ecosystem Affinity

Knowledge Tunnelling

***Misapplied Genericity**

Over-Engineering

Domain Allergy

Centralization Bottleneck

Detail Exposure

***Horizontalism**

Layerism

Pattern Mis-Transfer

Over-Isolation

Network Ignorance

Cargo-Culting

Innovation Addiction

***Failing Rigidity**

Abstraction Bias

Configurability Fallacy

Generic Conway Failure

Backend Fixation

Emotional Misattachment

***Over-Modularization**

Under-Modularization

Golf Course Peer Pressure

Vendor-driven Architecture

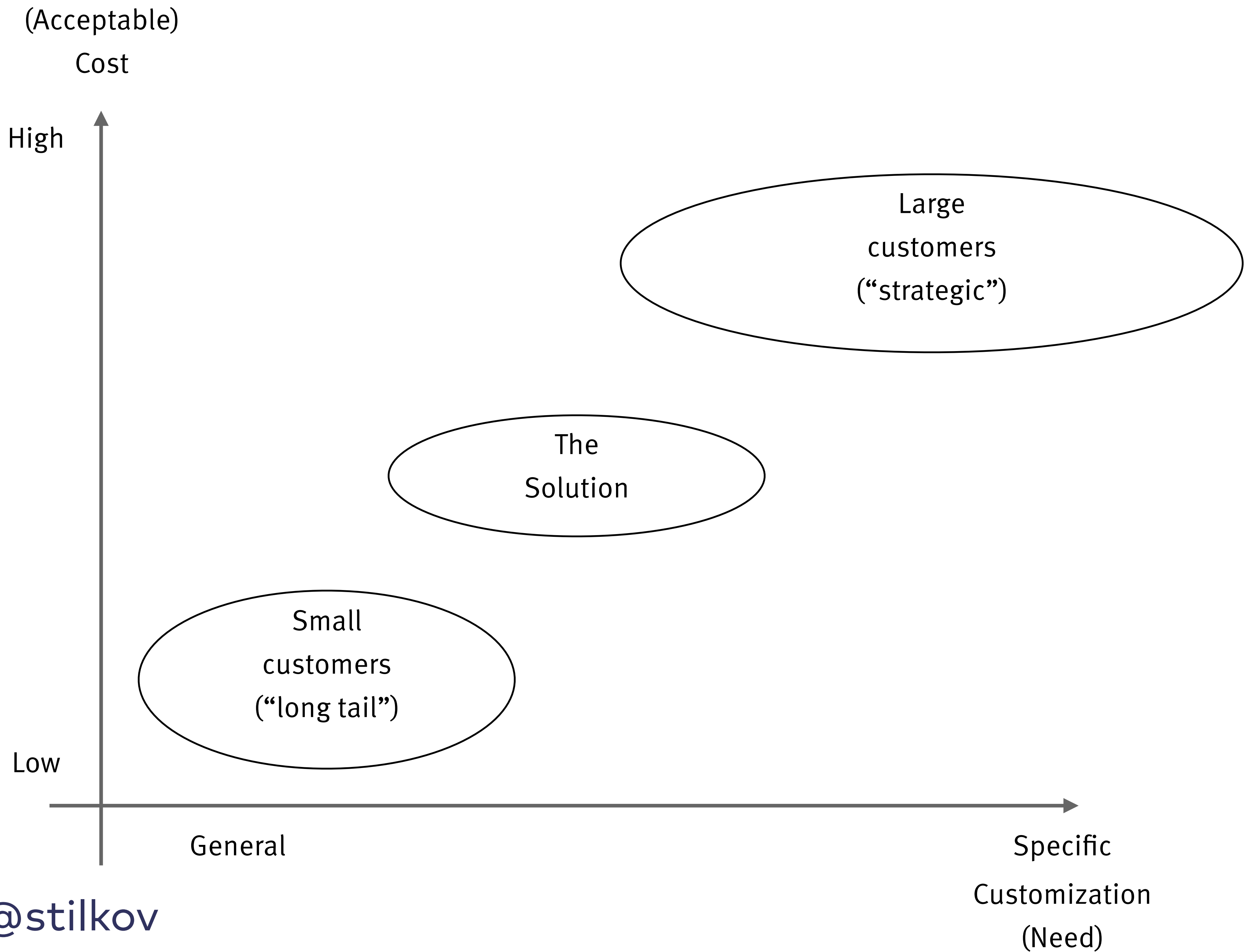
Cases

@stilkov

#1: Non-extensible Extensibility

Context

- **E-Commerce (retail) provider**
- **Global customer base**
- **Catalog/CMS/Shop/Fulfillment**
- **Multi-tenant**
- **Highly customizable**



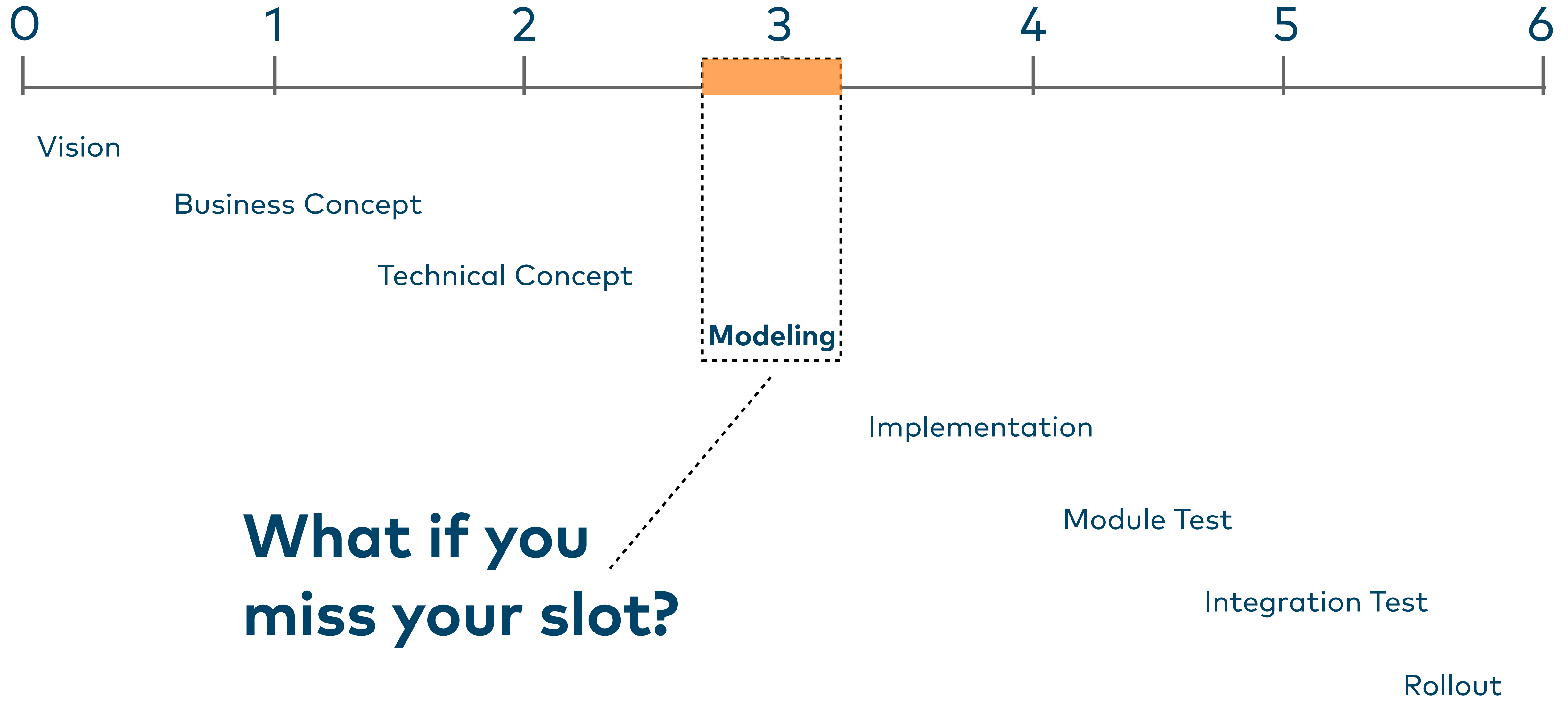
**If your design
attempts to satisfy
everyone, you'll
likely end up
satisfying no one**

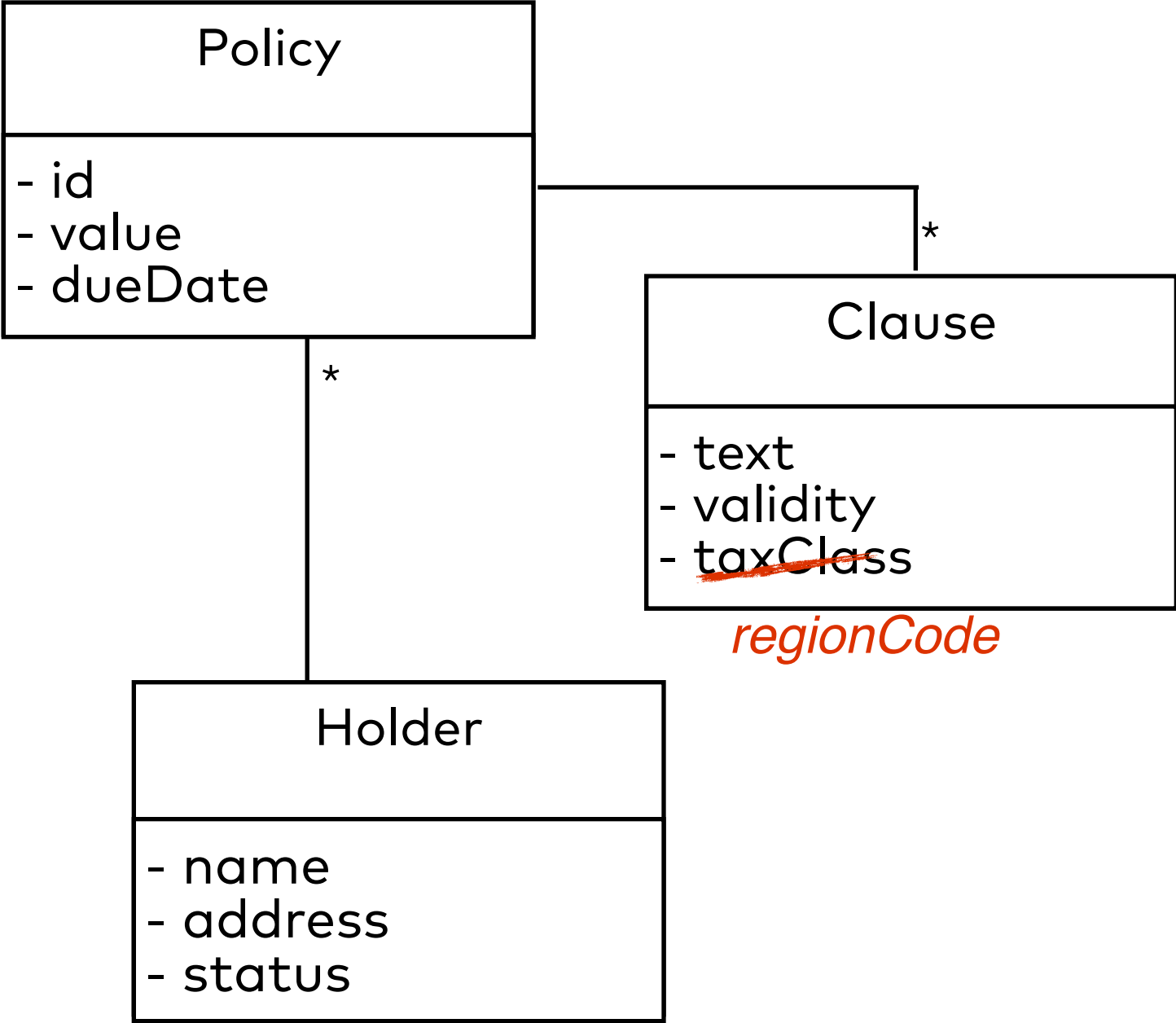
**Highly specific code
is often preferable
to sophisticated
configuration**

**#2: Your system
WILL be dynamic**

Context

- **Large-scale insurance system**
- **Model-driven development**
- **> 100 developers**
- **2 Releases/year**





Model Name	New Name (Meaning)	Description	Release Introduced
taxClass	regionCode	...	10.3
...			

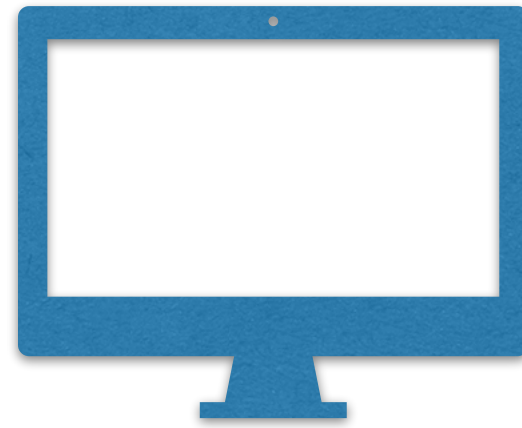
Lessons learned

- **Centralized responsibility hurts**
- **Faced with too much rigidity, developers will find a way around the rules**
- **Just because you're used to it doesn't mean it's acceptable**

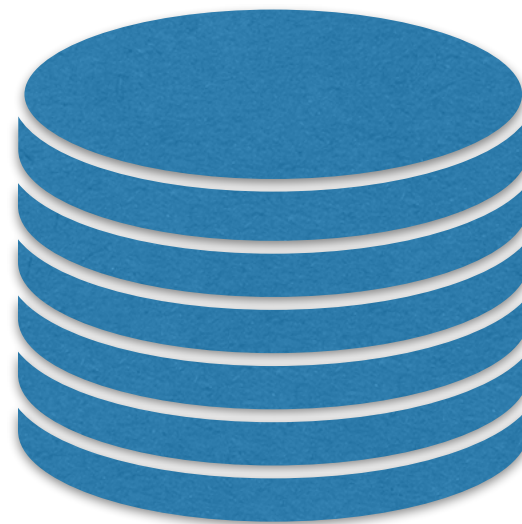
#3: Cancerous Growth

Context

- **Financial services provider with independent brokers as clients**
- **~30 developers**
- **20 years of company history**



Oracle Forms App

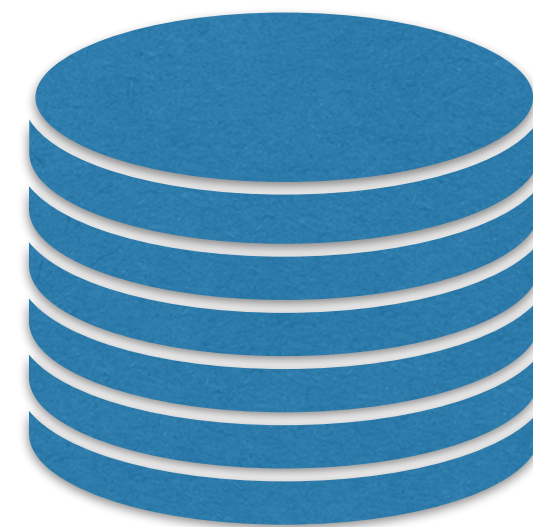


Oracle DB

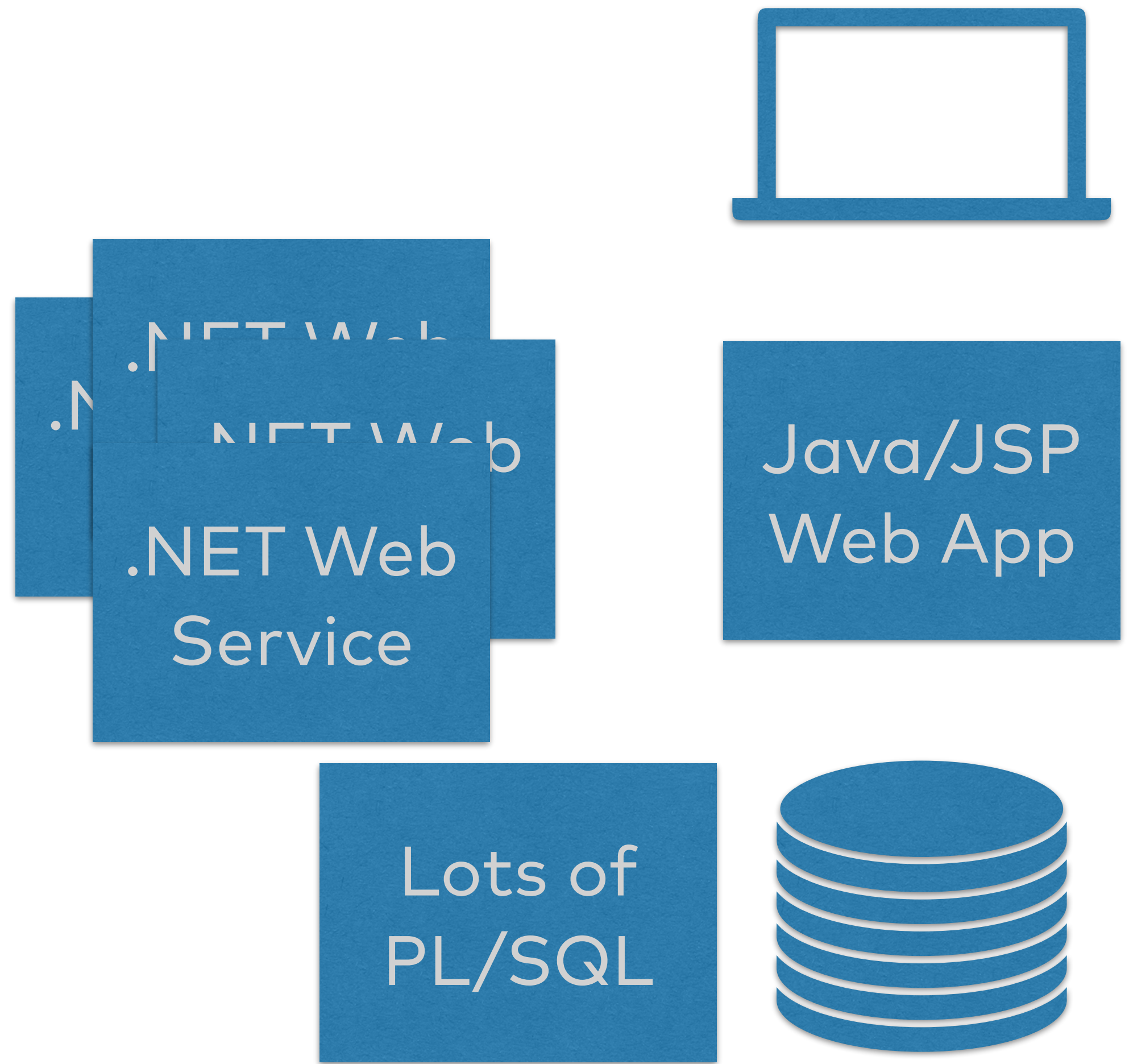


Java/JSP
Web App

Lots of
PL/SQL



Oracle DB



Oracle DB

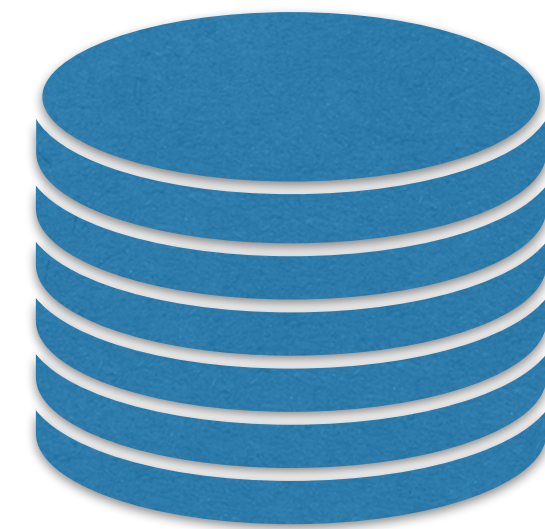
Company A



Java/JSP
Web App

.NET Web
Service

Lots of
PL/SQL



Oracle DB

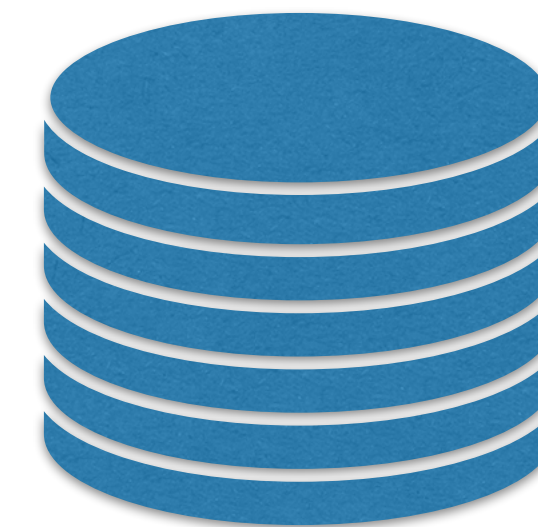
Company B



Java/JSP
Web App

.NET Web
Service

Lots of
PL/SQL



Oracle DB

Company A



Java/JSP
Web App

.NET Web
Service

Lots of
PL/SQL

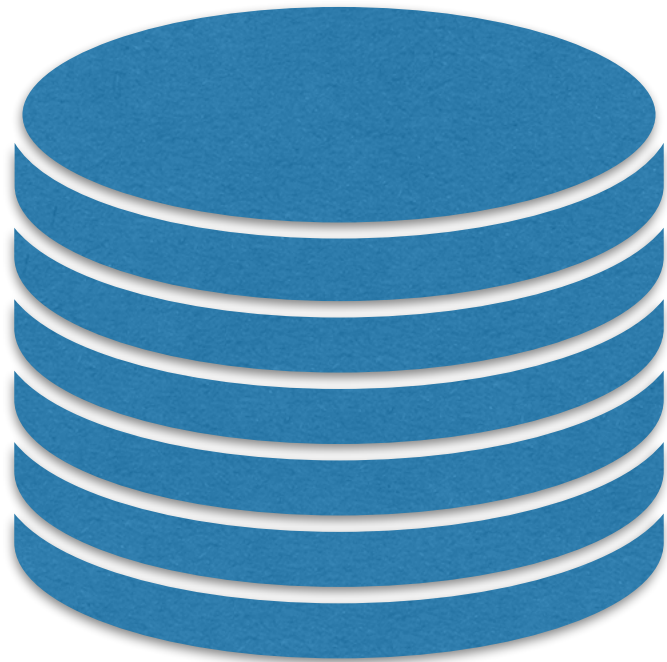
Company B



Java/JSP
Web App

.NET Web
Service

Lots of
PL/SQL

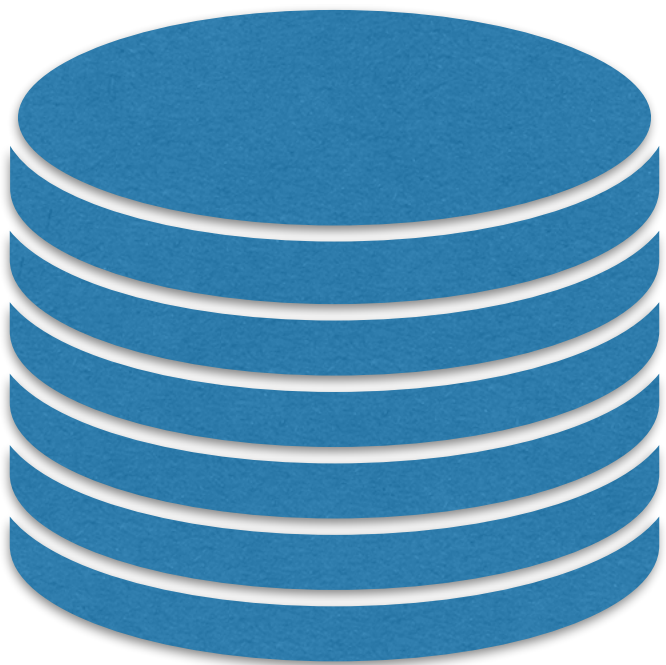
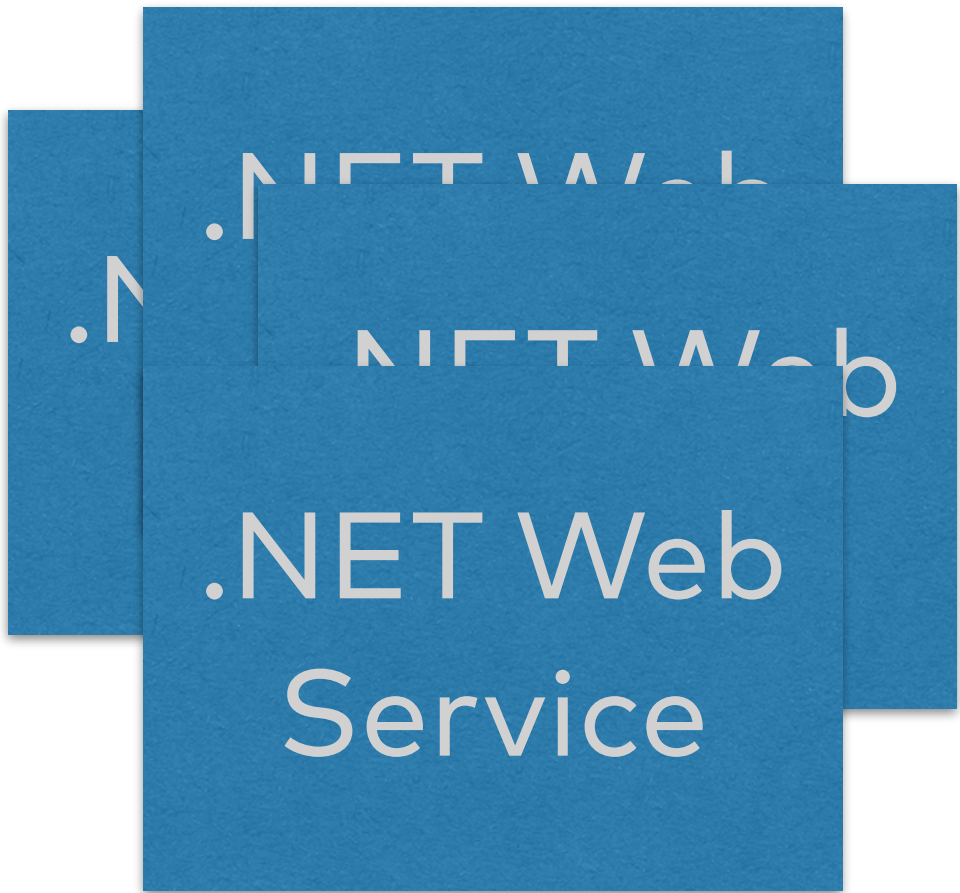


Oracle DB

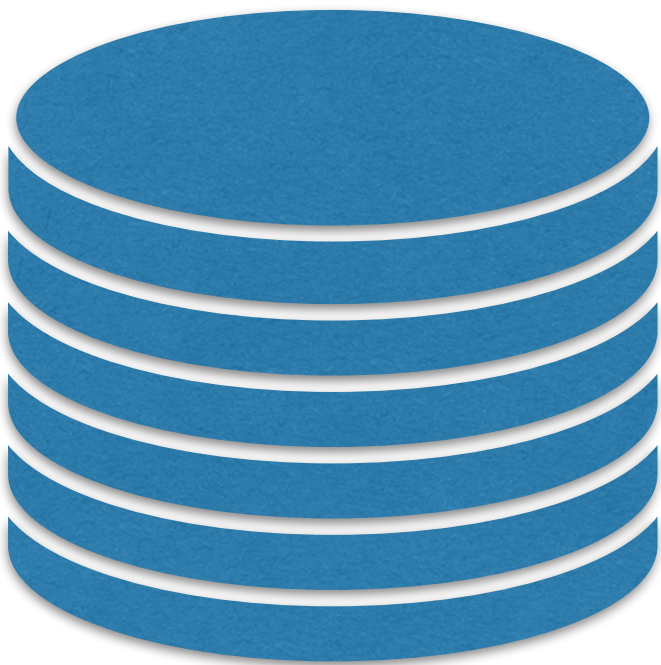
Company A



Java/JSP
Web App



Couch/Pouch
@stilkov



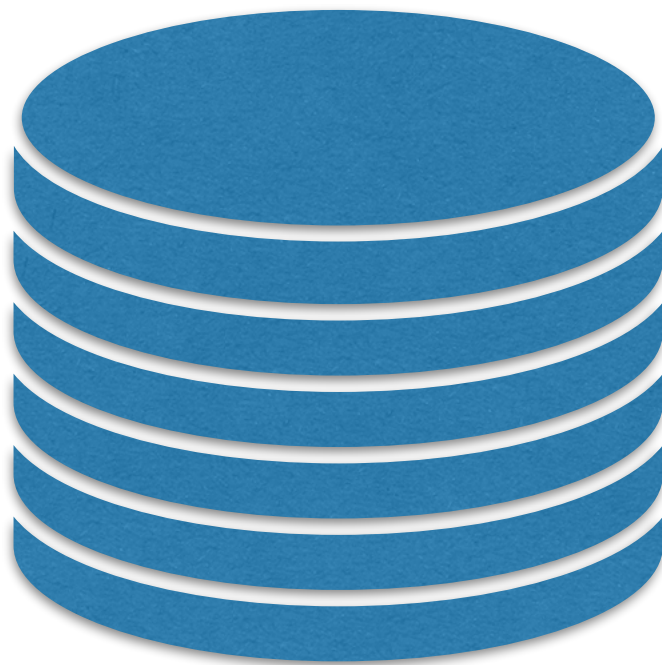
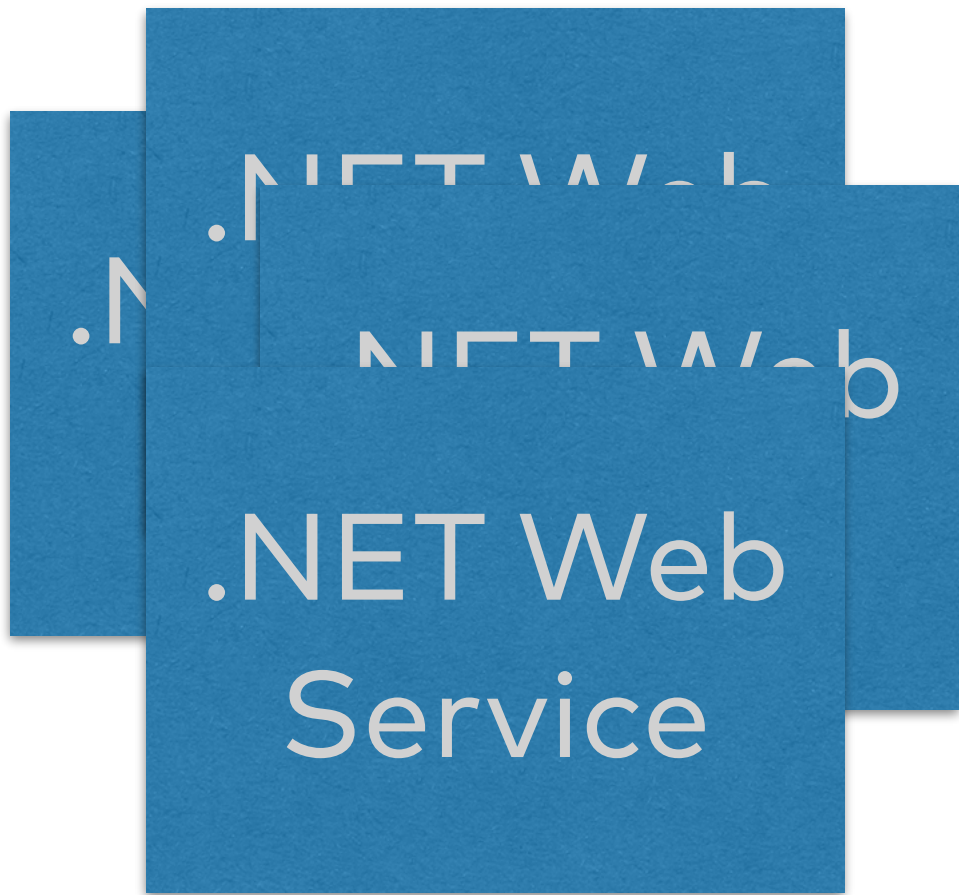
Mongo



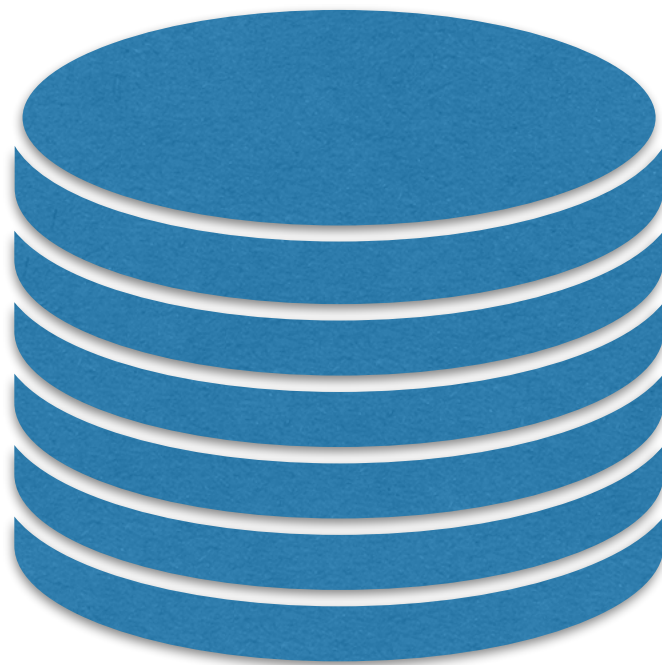
Company B



Java/JSP
Web App



Mongo



MySQL

Oracle DB

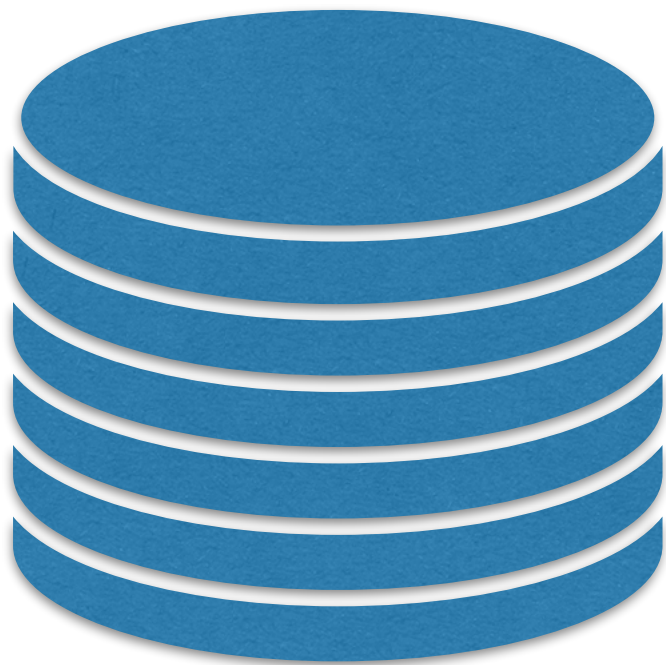
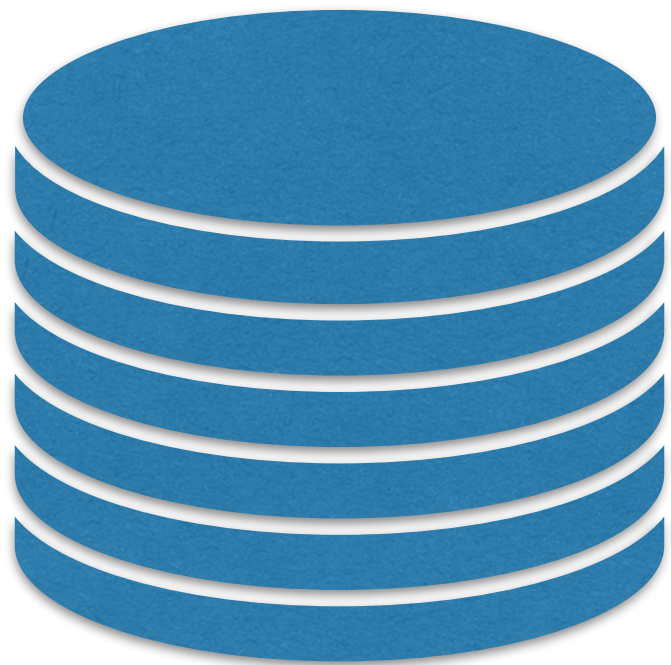
Company A

C++ Encryption Lib



.NET Web Service

Java/JSP Web App



Couch/Pouch
@stilkov

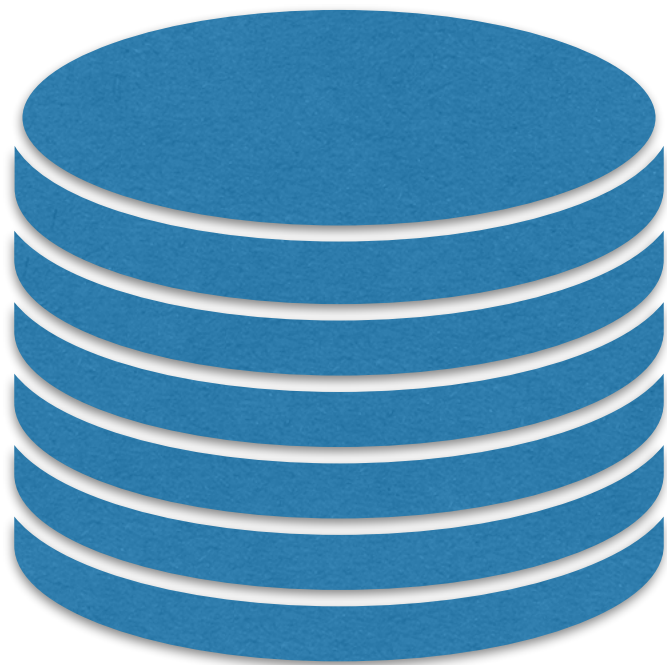
Mongo

Company B

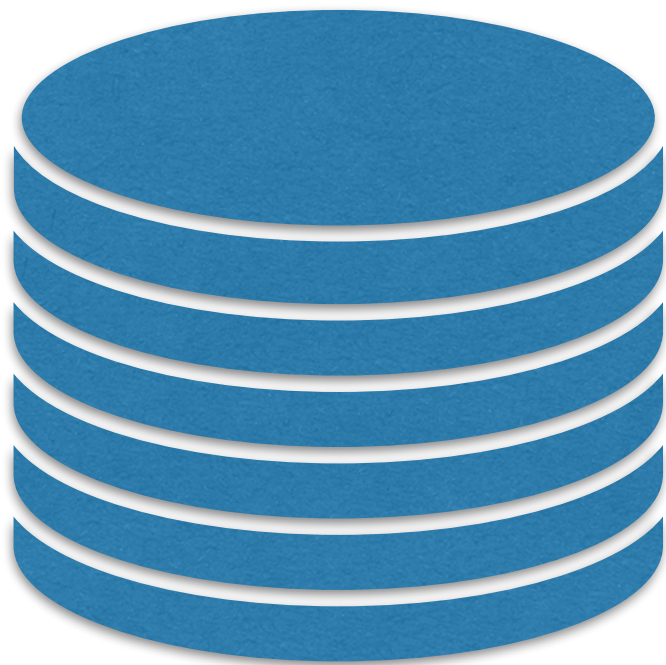


Java/JSP Web App

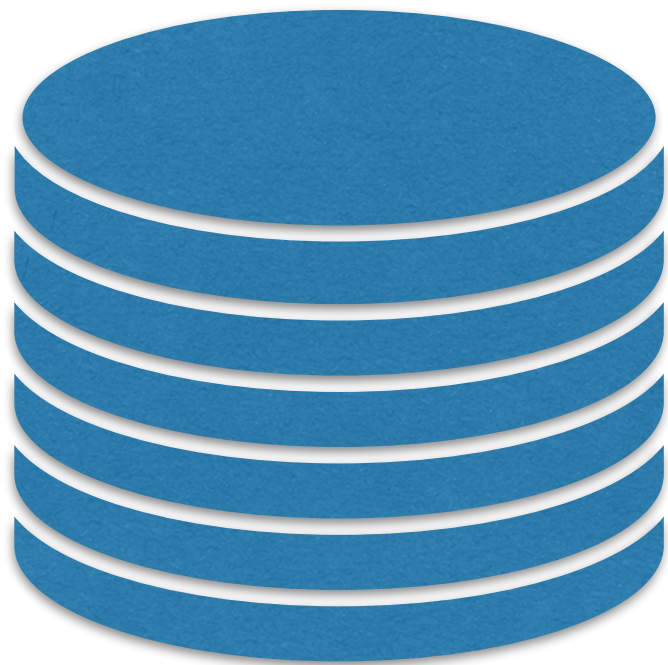
.NET Web Service



Oracle DB



Mongo



MySQL

**Successful systems
often end up with
the worst
architecture**

**Unmanaged
evolution will lead
to complete chaos**

**Don't be afraid of
some light
architectural
governance**

Takeaways

1.

**Don't be afraid of
architecture**

2.

**Choose the
simplest thing that
will work**

3.

**Create evolvable
structures**

4.

Manage your system's architectural evolution

5.

**Don't build road
blocks – create
value and get out of
the way**

That's all I have.

Thanks for listening!

Stefan Tilkov, @stilkov
stefan.tilkov@innoq.com
Phone: +49 170 471 2625



www.innoq.com

SERVICES

Strategy & technology consulting
Digital business models
Software architecture & development
Digital platforms & infrastructures
Knowledge transfer, coaching & trainings

FACTS

~150 employees
Privately owned
Vendor-independent

OFFICES

Monheim
Berlin
Offenbach
Munich
Hamburg
Zurich

CLIENTS

Finance
Telecommunications
Logistics
E-commerce
Fortune 500
SMBs
Startups