innoQ

Architecture War Stories

Fun Stories from an Architect's Life

Stefan Tilkov | @stilkov | innoQ

GOTO Berlin

11/07/2014

Ground Rules

No offense – just lessons learned

Most systems are in production

Still, many {managers | users | admins} are happy

Less can be more

Context

Customer care, accounting & billing

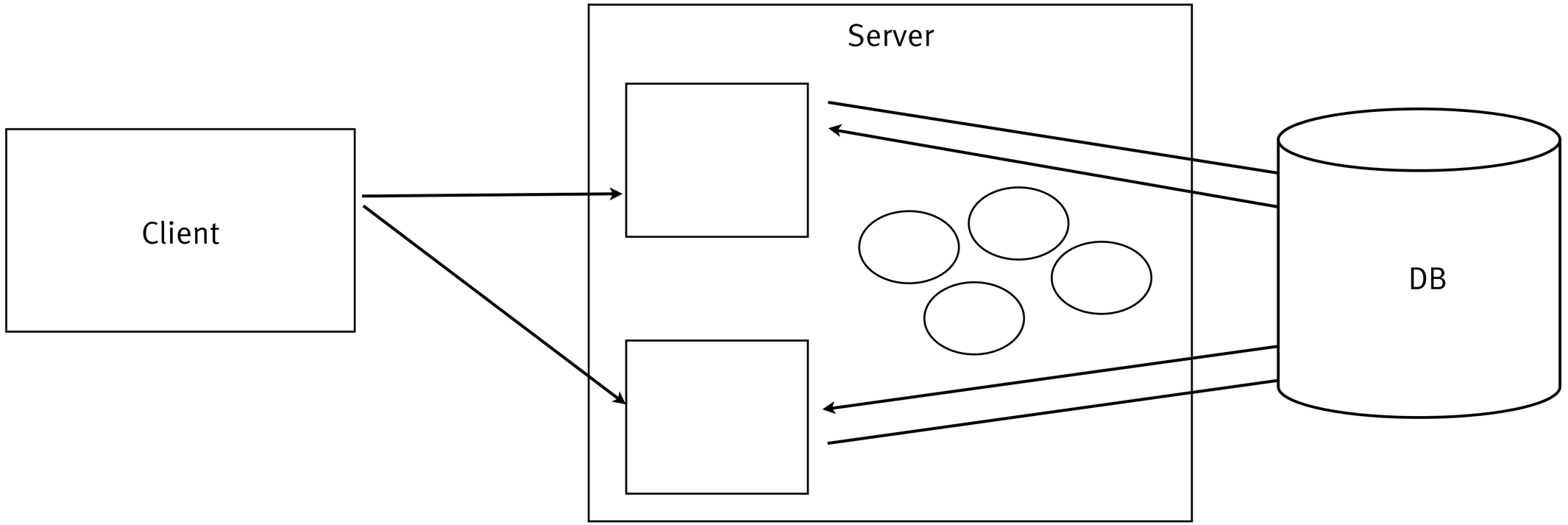
Large-scale big-bang replacement effort

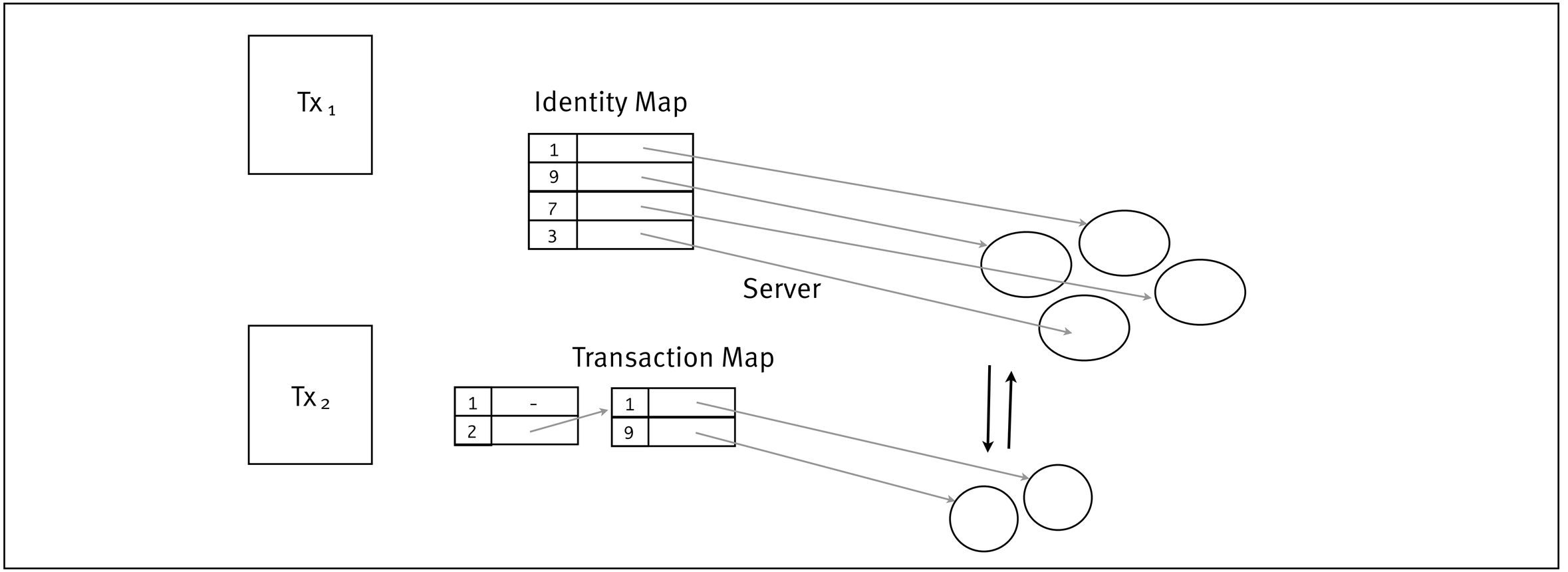
› 150 developers

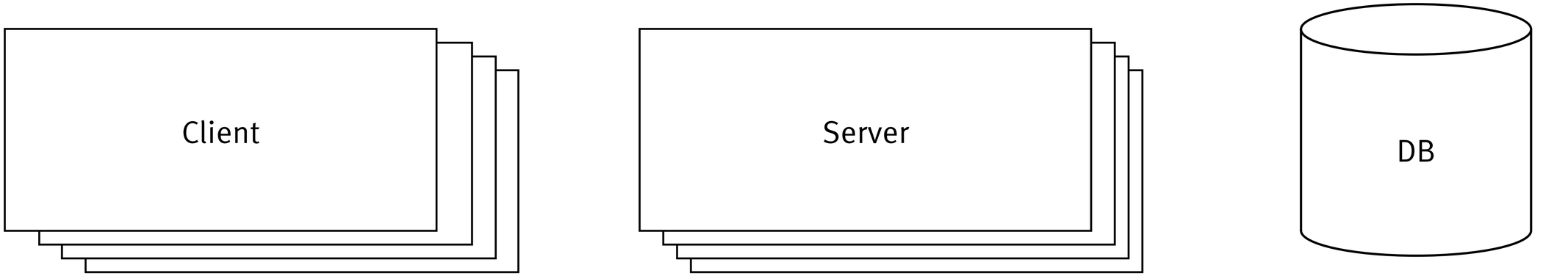
› 20 architects/framework devs

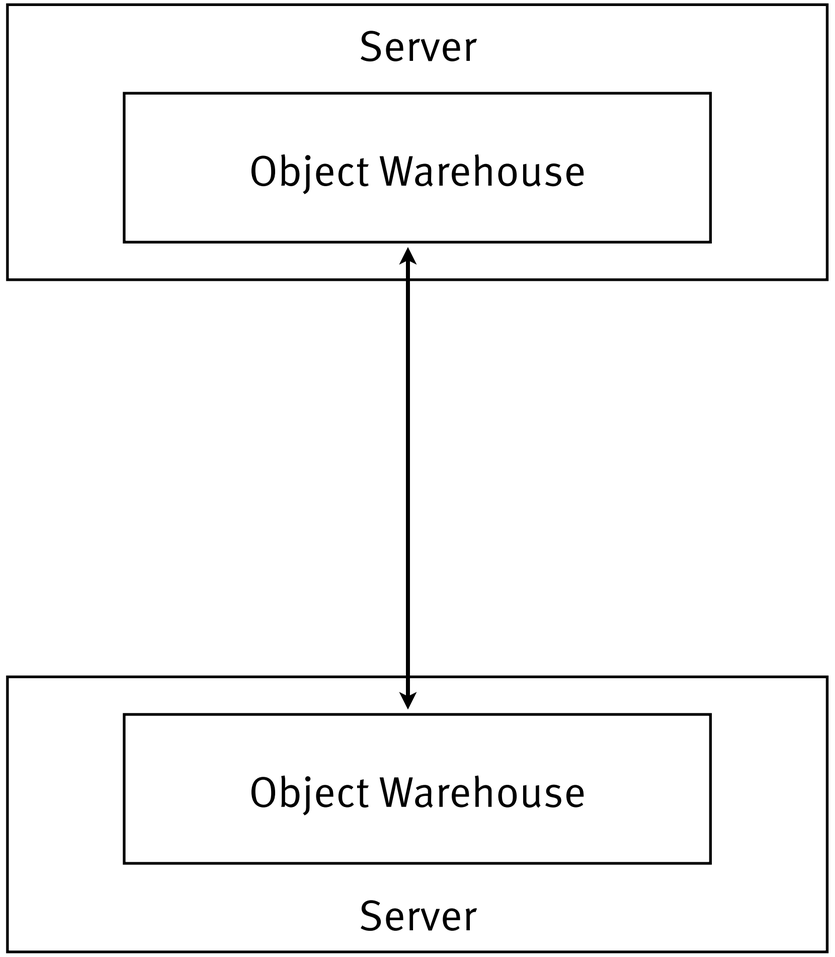
Java, Swing, CORBA, Oracle

Custom persistence framework


Server
Client
DB


Tx 1
Identity Map
1
9
7
3
Server
Tx 2
Transaction Map
1
2
-
1
9

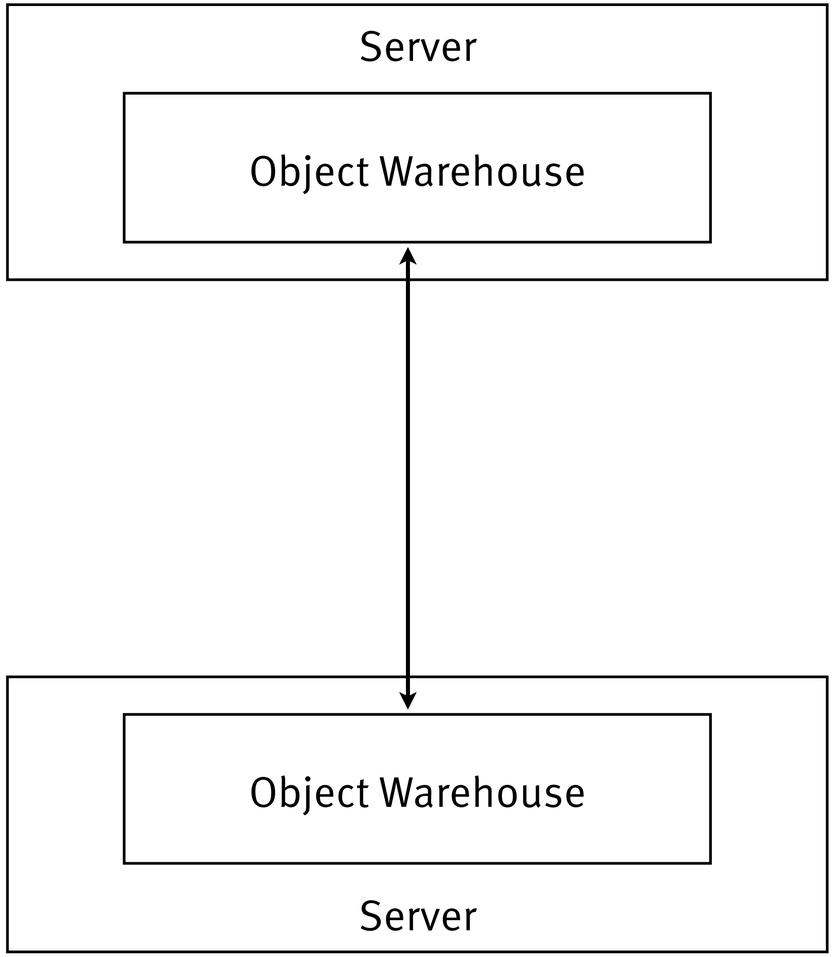

Client
Server
DB

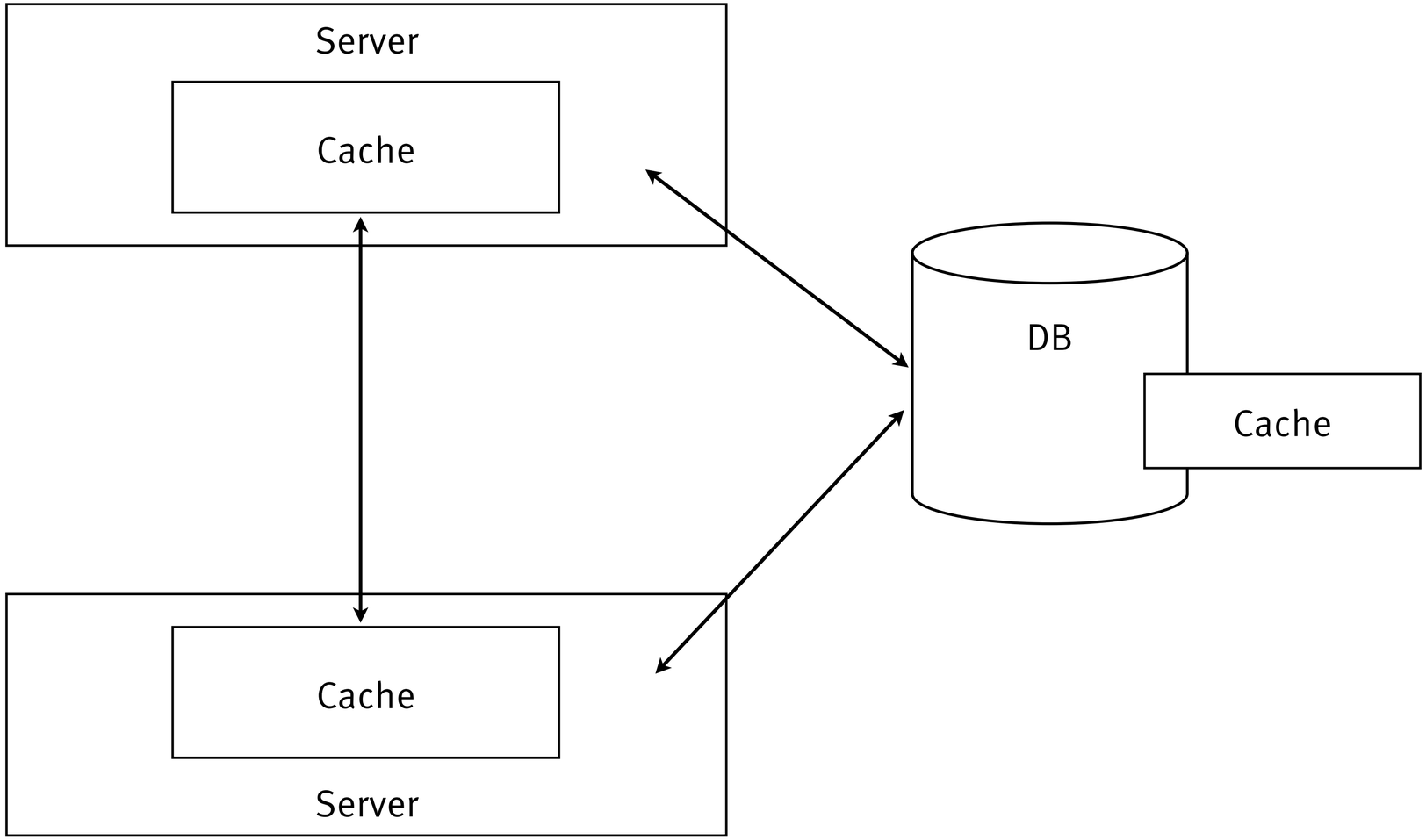

Server
Object Warehouse
Object Warehouse
Server

Global synchronization? 2 PC?

Cache notifications? Events?

Transaction locality? Client Affinity?


Server
Object Warehouse
Object Warehouse
Server


Server
Cache
DB
Cache
Cache
Server

Lessons learned

Don’t cache the cache

In distributed systems, application state is your enemy

If it’s “sophisticated”, you probably shouldn’t be doing it

Development ≠ production

Data Complications

Context

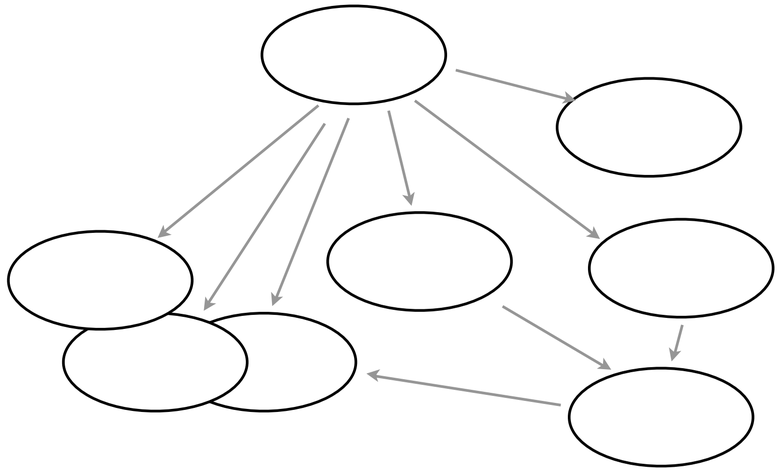
Insurance system

Maintenance 1000 person months/year

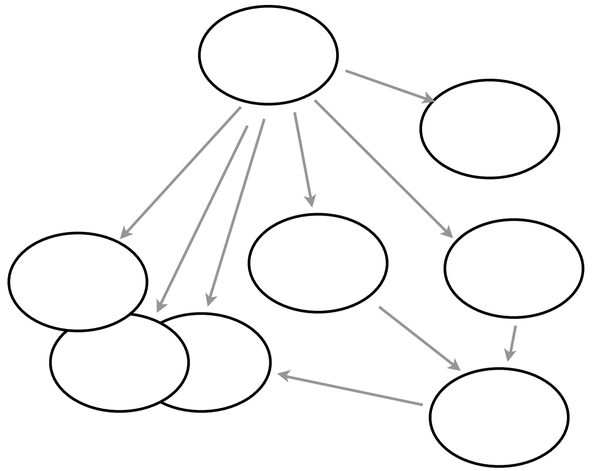
Smalltalk, COBOL, C, IMS, DB2

Homegrown COBOL OO system

Client
(Smalltalk)



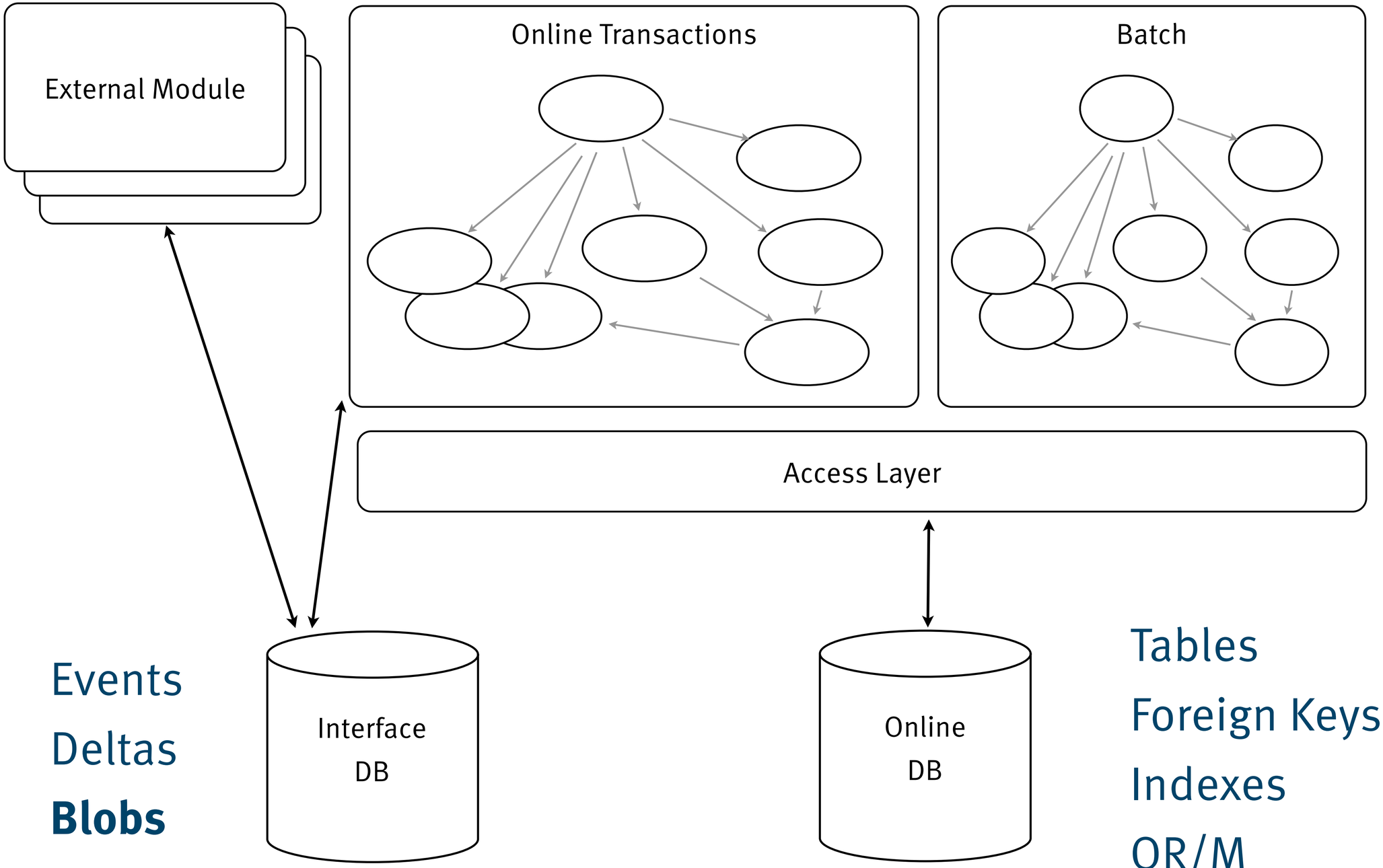
Server
(COBOL)



Broker

Gateway

Transaction Manager



Events

Deltas

Blobs

Tables

Foreign Keys

Indexes

OR/M

Migration

Lessons learned

Data should be free of code dependencies

The longer you store it, the simpler it needs to be

Beware of smelly blobs

Evaluating Yourself to Death

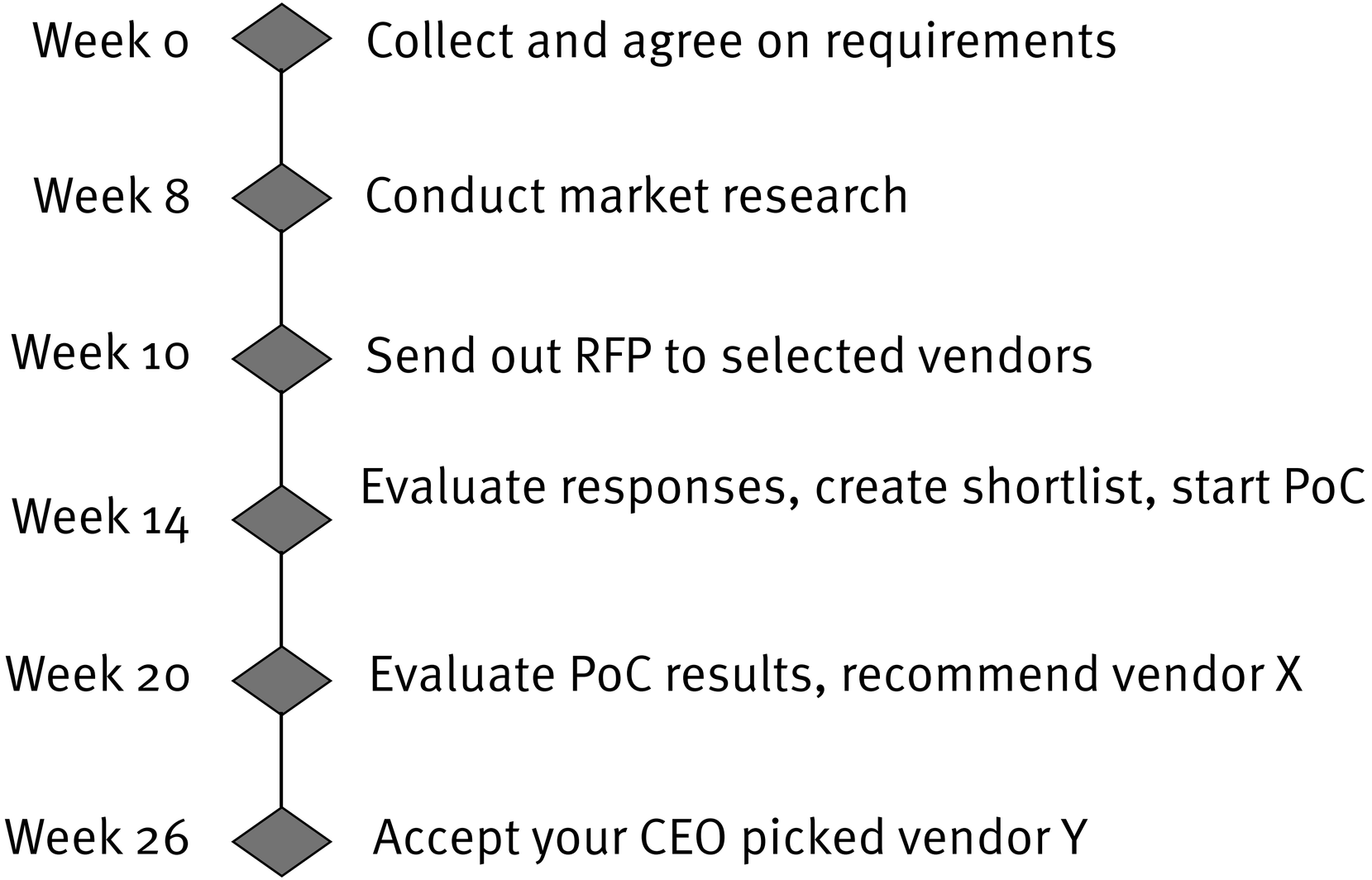
Context

Strategic health care platform

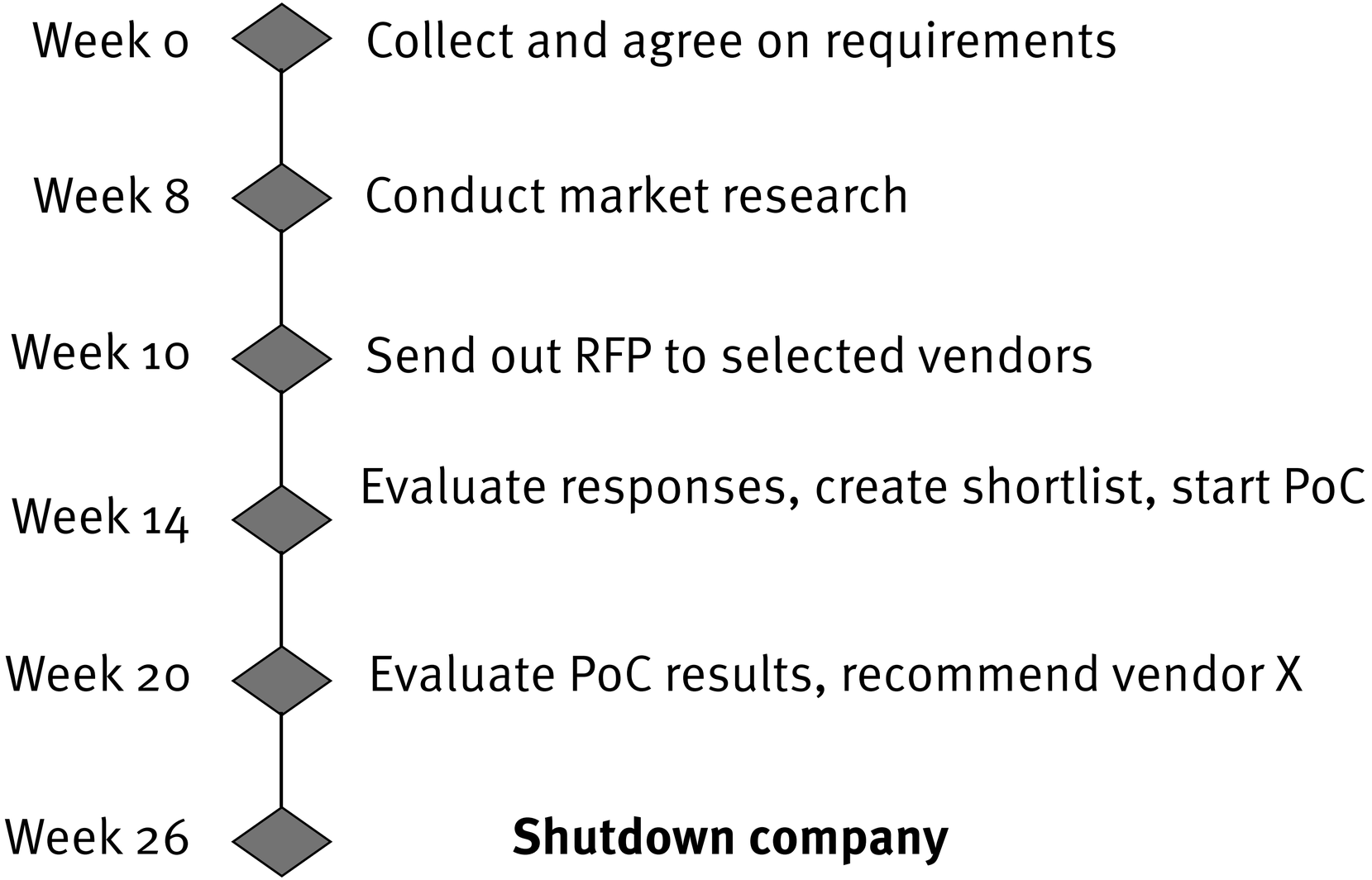
Architecture team 20+10 people

Build foundation for 800+ person year effort

Vendor Selection



Vendor Selection



Lessons learned

Just do it

Better ask for forgiveness than permission

Your system WILL be dynamic

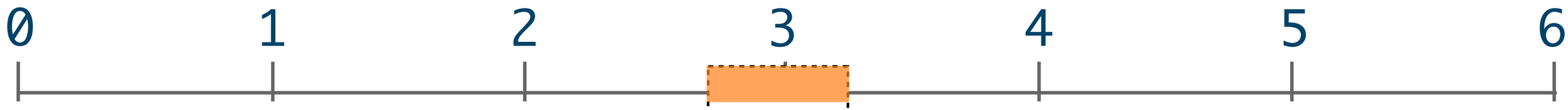
Context

Large-scale insurance system

Model-driven development

› 100 developers

2 Releases/year



Vision

Business Concept

Technical Concept

Modeling

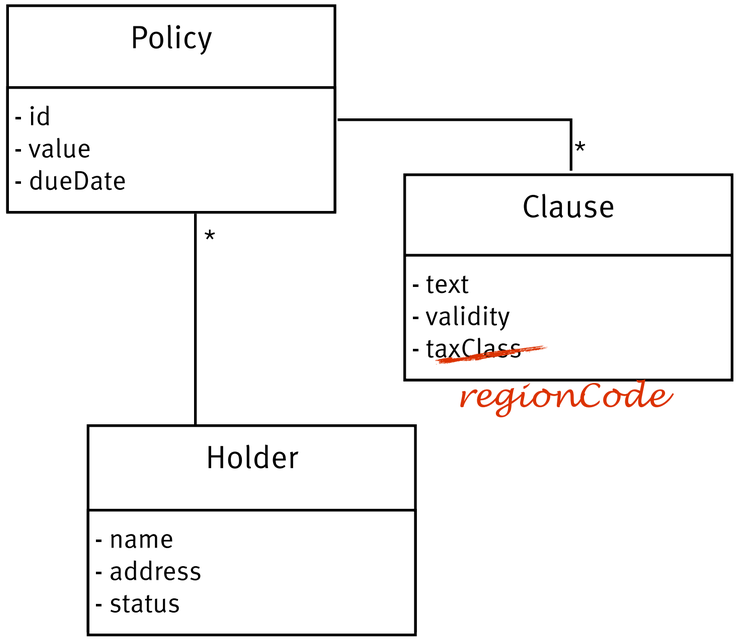
Implementation

Module Test

Integration Test

Rollout

What if you miss your slot?



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

Lessons learned

Centralized responsibility hurts

Faced with too much rigidity, a way around the rules will be found

Just because you’re used to it doesn’t mean it’s acceptable

Formalism
and Common Sense
as Orthogonal Concepts

Context

Telecommunications OSS platform

Support new devices without code changes

› 100 developers

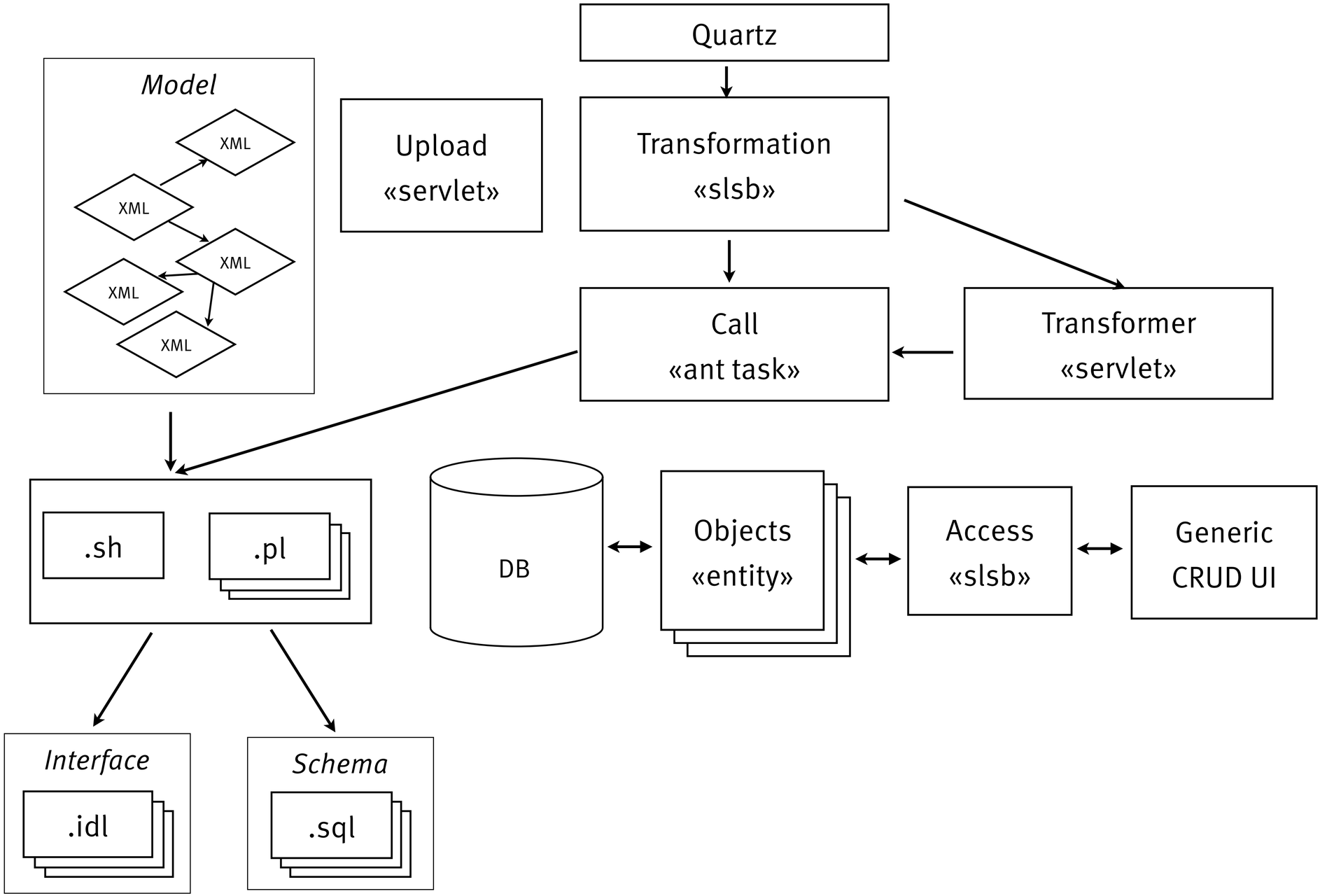
› 3 countries

Warning

((Meta) meta) models) are a perfect means to waste time, on a highly inspiring intellectual level, without any results for years

Highly complex methodology, tool support, agreement processes

Consultants with *exclusive* pricing strategies


Quartz
Model
XML
XML
XML
XML
XML
Upload
«servlet»
Transformation
«slsb»
Call
«ant task»
Transformer
«servlet»
.sh
.pl
DB
Objects
«entity»
Access
«slsb»
Generic
CRUD UI
Interface
.idl
Schema
.sql

Lessons learned

If it makes you want to scratch your eyes out, you might not want to do it

Sometimes constraints need to be fought

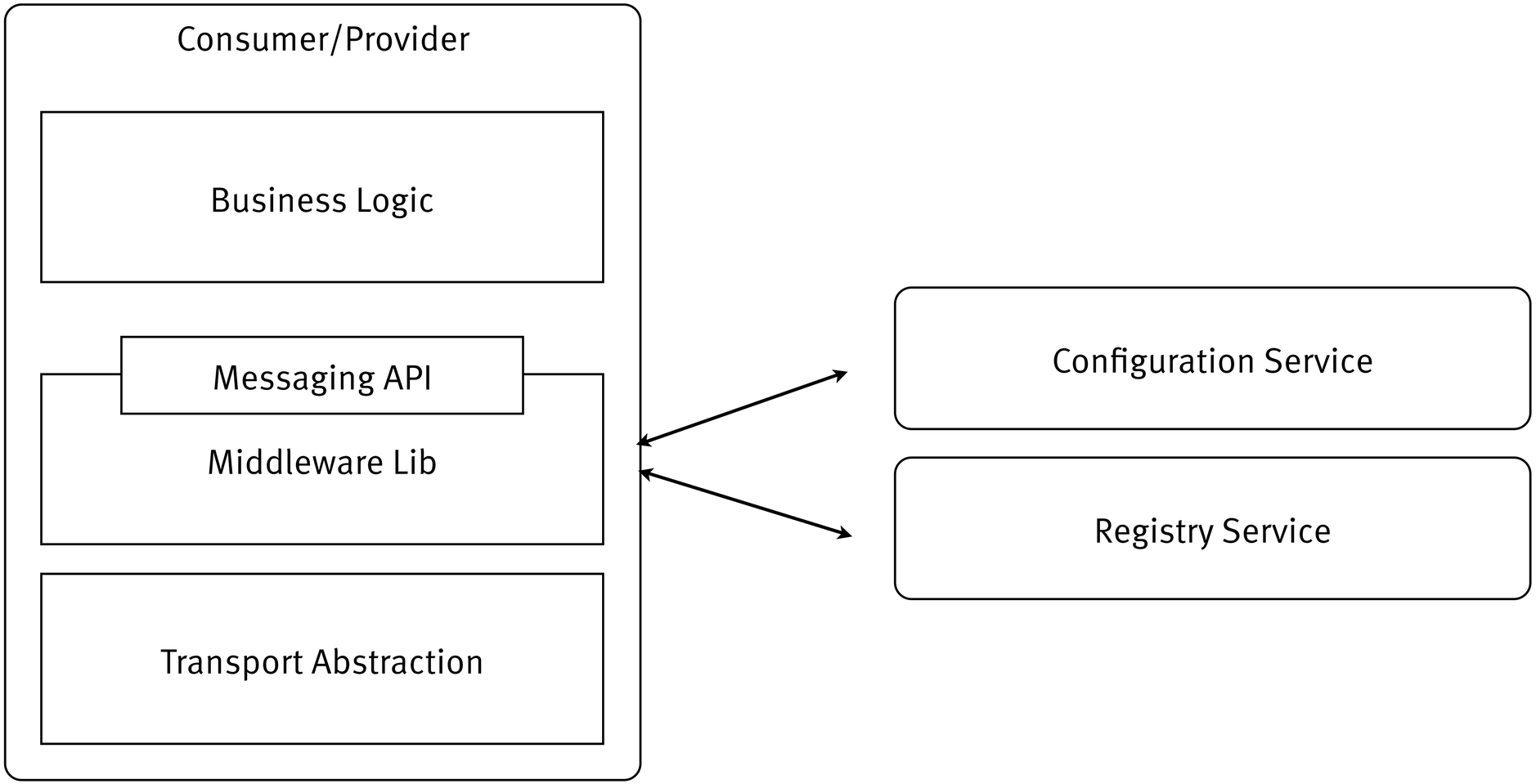
Beware all things “meta”

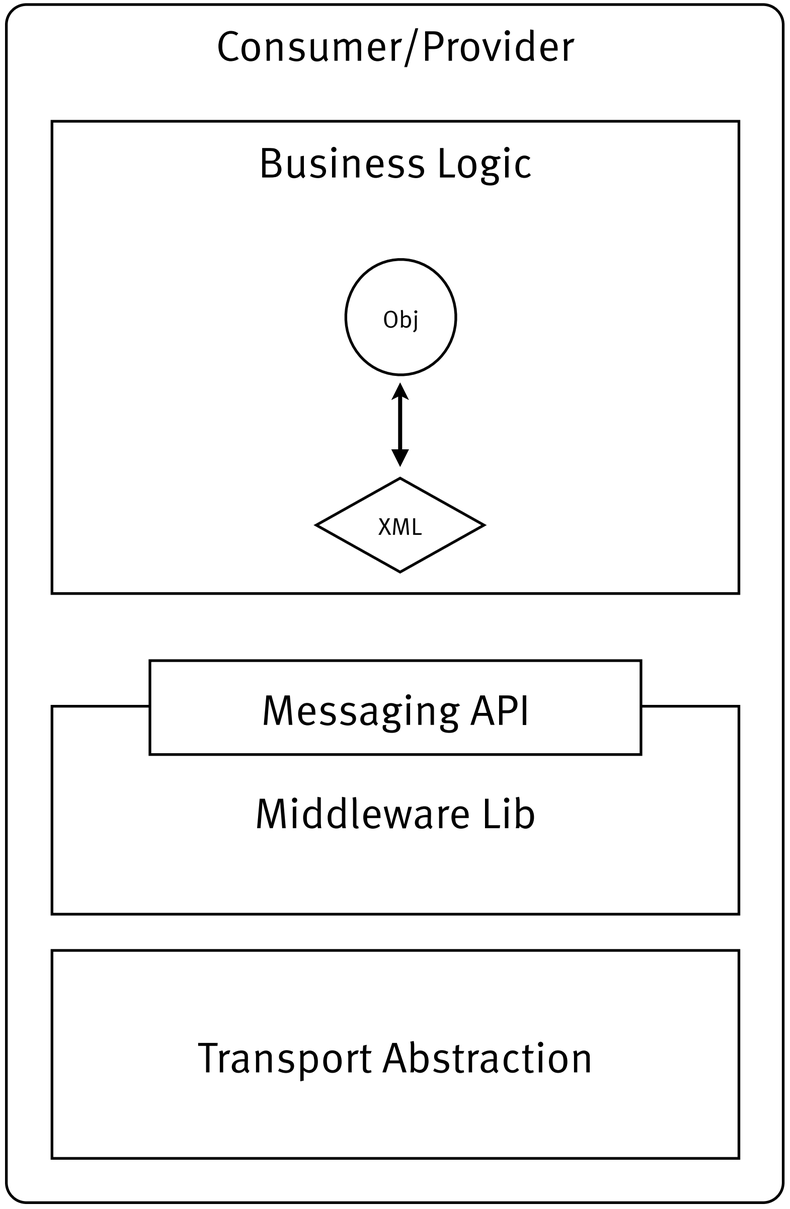
The Perils of Being Smart

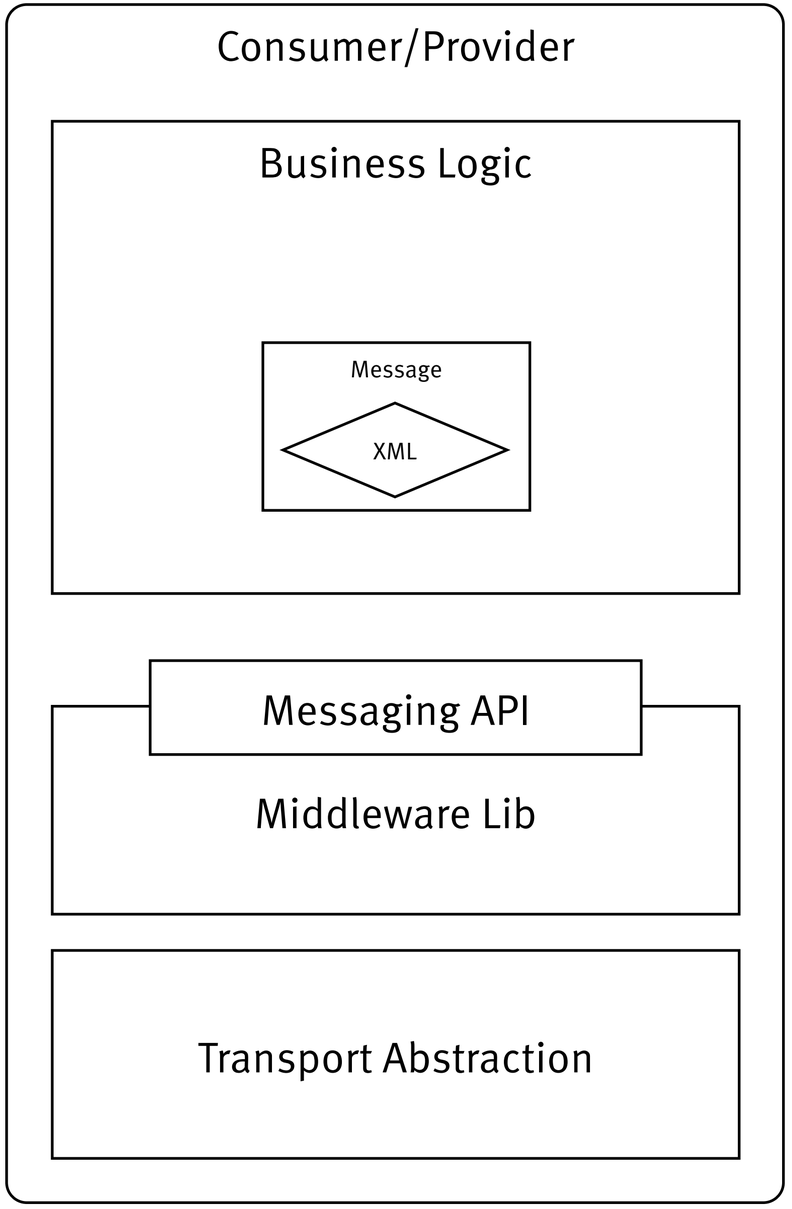
Context

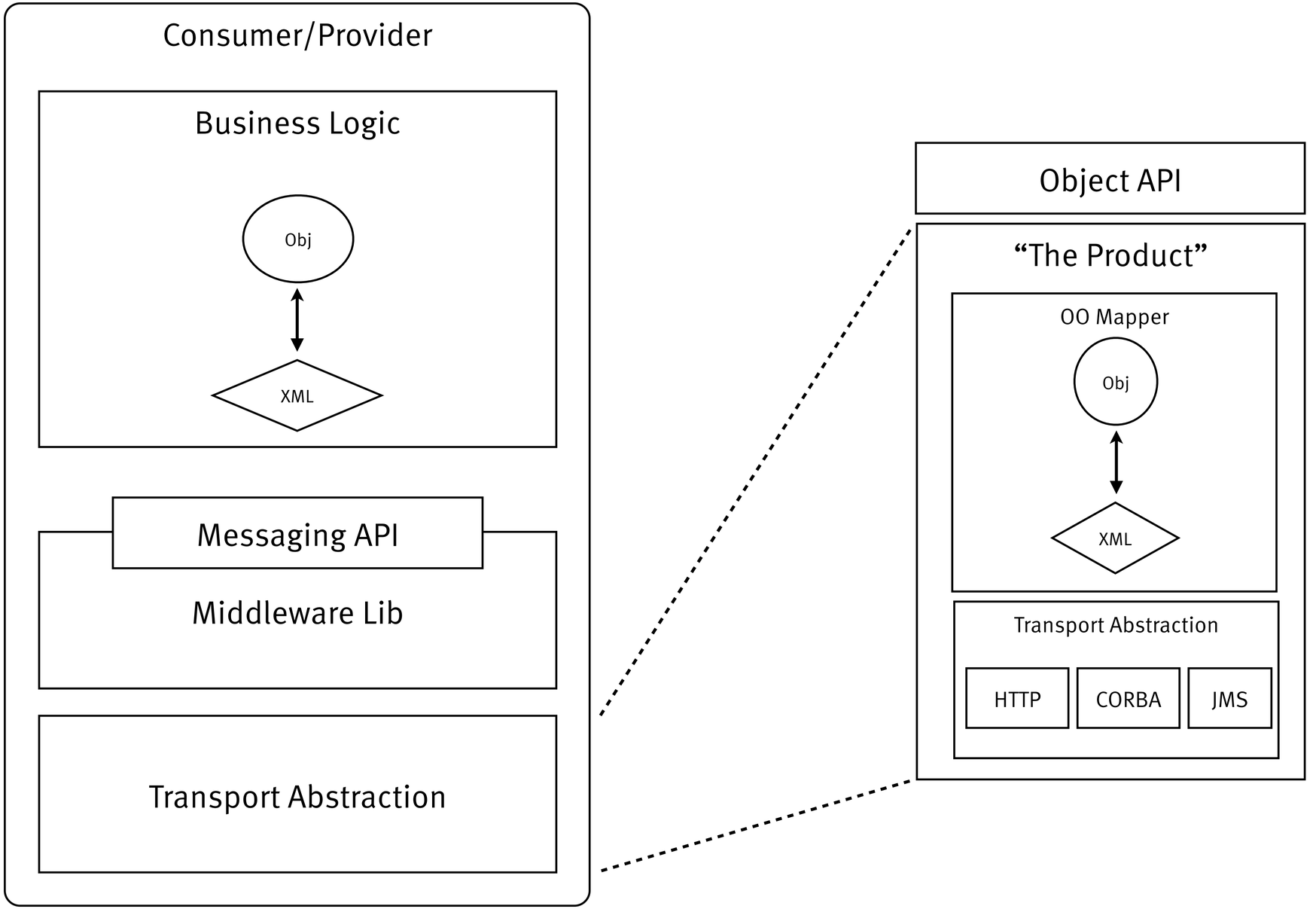
Complex enterprise-wide SOA effort

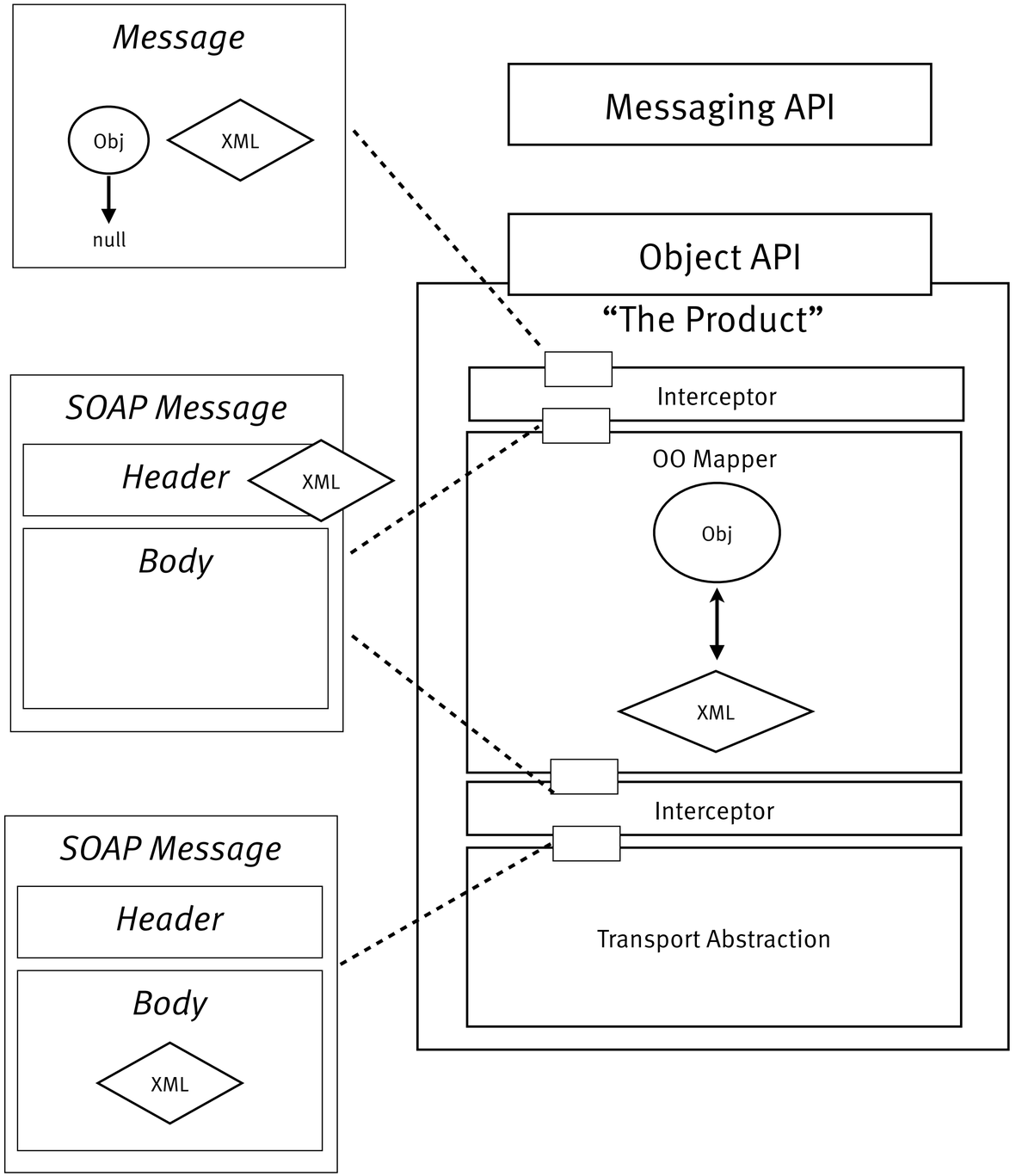
Custom-built “lightweight” middleware


Consumer/Provider
Business Logic
Messaging API
Middleware Lib
Transport Abstraction
Configuration Service
Registry Service


Consumer/Provider
Business Logic
Obj
XML
Messaging API
Middleware Lib
Transport Abstraction


Consumer/Provider
Business Logic
Message
XML
Messaging API
Middleware Lib
Transport Abstraction


Consumer/Provider
Business Logic
Obj
XML
Messaging API
Middleware Lib
Transport Abstraction
Object API
“The Product”
OO Mapper
Obj
XML
Transport Abstraction
HTTP
CORBA
JMS


Message
Obj
XML
null
Messaging API
Object API
“The Product”
Interceptor
OO Mapper
Obj
XML
Interceptor
Transport Abstraction
SOAP Message
Header
XML
Body
SOAP Message
Header
Body
XML

Lessons learned

Fight the madness

Tunneling is the root of all evil

Don’t be too smart

Non-extensible Extensibility

Context

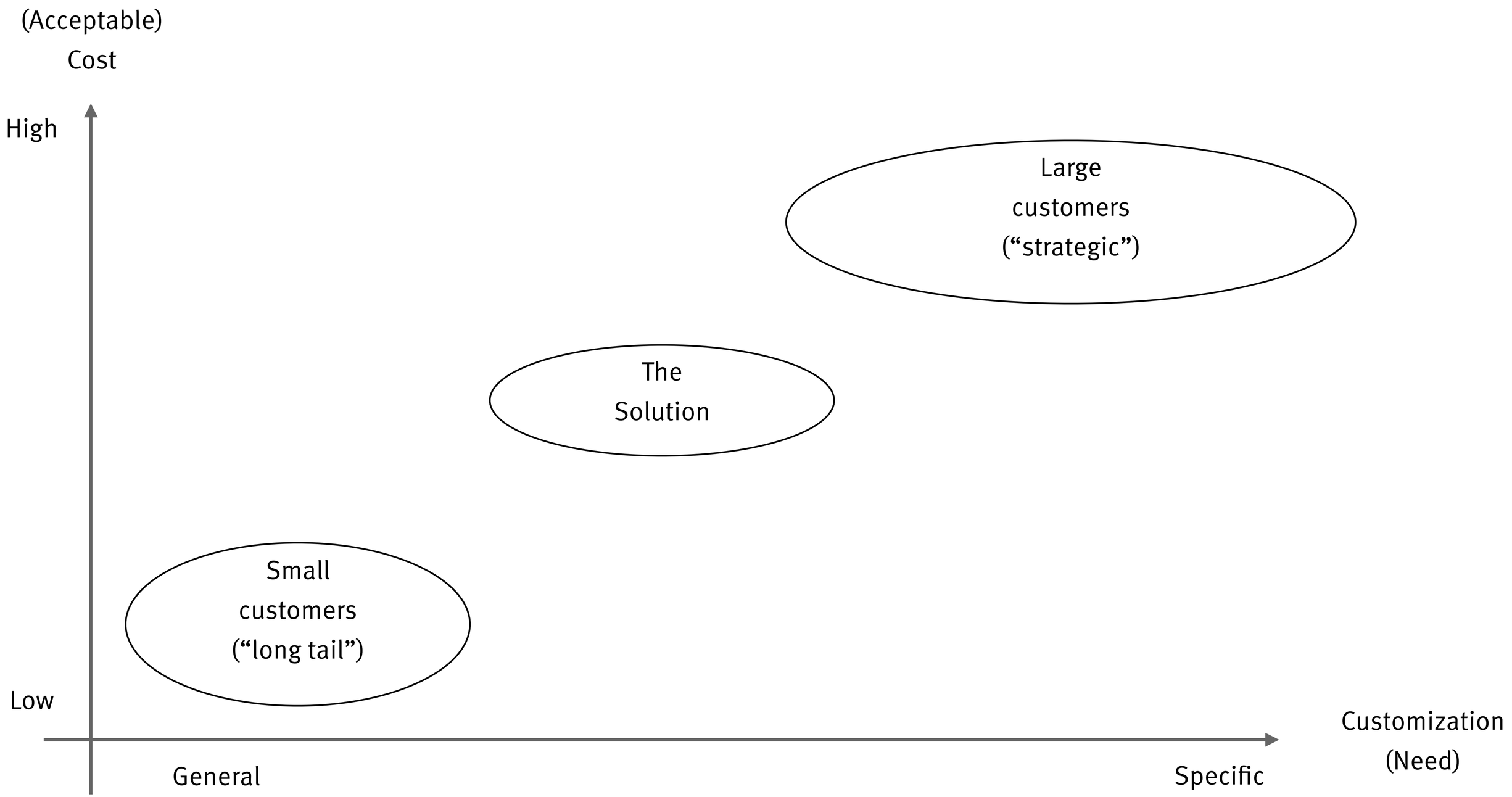
E-Commerce (retail) provider

Global customer base

Catalog/CMS/Shop/Fulfillment

Multi-tenant

Highly customizable


(Acceptable)
Cost
High
Large
customers
(“strategic”)
The
Solution
Small
customers
(“long tail”)
Low
General
Specific
Customization
(Need)

Lessons learned

Don’t be all things to all people

Sometimes specific code is preferable

Saying “no” might help

Summary

Never-ending supply of
{war stories | catastrophes | disasters}
seems guaranteed

Advice:

1. Give Feedback

2. Reflect

3. Iterate

Q&A

Stefan Tilkov, @stilkov

stefan.tilkov@innoq.com

Phone: +49 170 471 2625

www.innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Ohlauer Straße 43
10999 Berlin
Germany
Phone: +49 2173 3366-0

Robert-Bosch-Straße 7
64293 Darmstadt
Germany
Phone: +49 2173 3366-0

Radlkoferstraße 2
D-81373 München
Germany
Telefon +49 (0) 89 741185-270

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116

info@innoq.com