Legacy application
Large ambitions

A case study: how **DDD** helped deliver at **Sungard Asset Arena**

Cyrille Martraire - @cyriux - arolla

http://cutefunnyanimal.blogspot.in/2012/01/funny-tired-dogs-photos-images-2012.html

LEGACY IS

The New Sexy

Passionate developer

PARIS

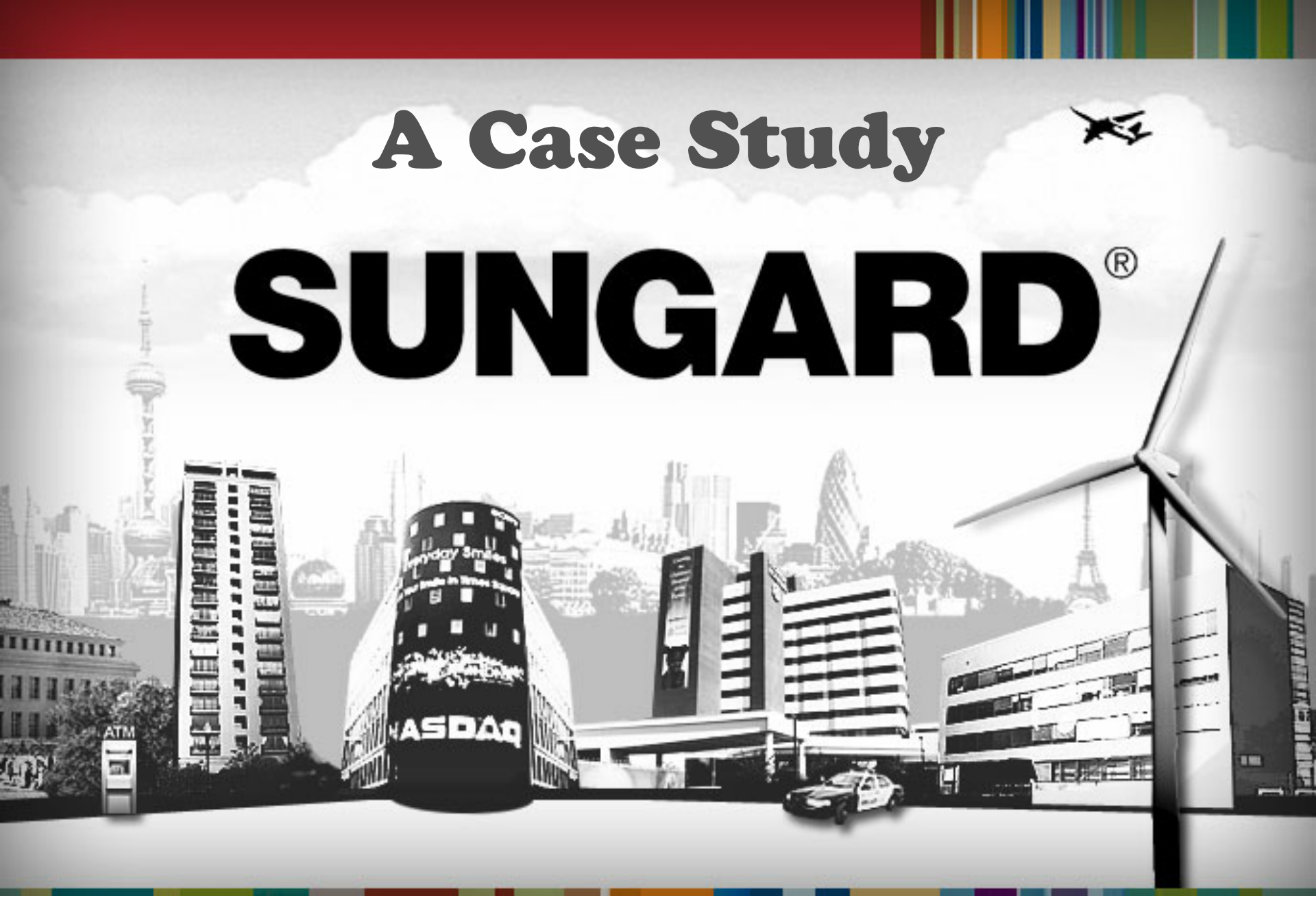
Since 1999

@cyriux


Paris Software
Craftsmanship Community
http://www.meetup.com/paris-software-craftsmanship/


Arolla
Haskell
Erlang
Academic
Design by Contract
Formal Verification
CAP theorem
MBA
Naked Objects
AOP
UML2
EJB
JavaEE
CDI
Servlets
JSP
Scala
Clojure
Groovy
Grails
JAVA
JVM
Play!
Wicket
GWT
SPRING MVC
JSF
SWING
JAXB
JMX
Hibernate
JPA
LINQ
Spring.NET
WCF
WinForms
WPF
ADO'NET
C#
.Net
CLR
VB.NET
ASP.NET
MVVM
Silverlight
ASP.MVC
LEGACY OCEAN
Project management
PL/SQL
SQL
ORACLE
LEGACY
OPEN SOURCE
Cross-team comunication
Short Iterations
Feedback ASAP!
AGILE OCEAN
TDD
SOLID
Clean Code
KISS
YAGNI
Design Patterns
OO Design
HTML5
ANDROID
iOS
Continuous Delivery
Dev-Ops
SCRUM
Agile Games
Specification by example
XP
Kanban
Lean
Software Factory
BUILD tools
Ivy
Maven
Ant
Hudson
Gradle
Sonar
Refactoring
IDE
Continuous Integration
TFS
Cruise Control
Bamboo
TeamCity
ATDD
BDD
DDD
Domain-Driven Design
MDD
Model-driven
Code is model
Ubiquitous Language
Elastic provisioning
Multicore concurrency
SOA
ESB & ServiceBus
WS
SOAP
REST
RAILS
Cloud
Actor
STM
SEDA
Map-Reduce
BigData
Event Sourcing
noSQL
CQRS
Balagrid
Enterprise Architecture
GC Tuning
Profiler
Extreme performances
DOMAIN
AROLLA.fr
2011
arolla.fr

A Case Study



From a recent project (contract)

Every example is made-up to protect the innocent IP.

Once upon a time...

Large Ambition

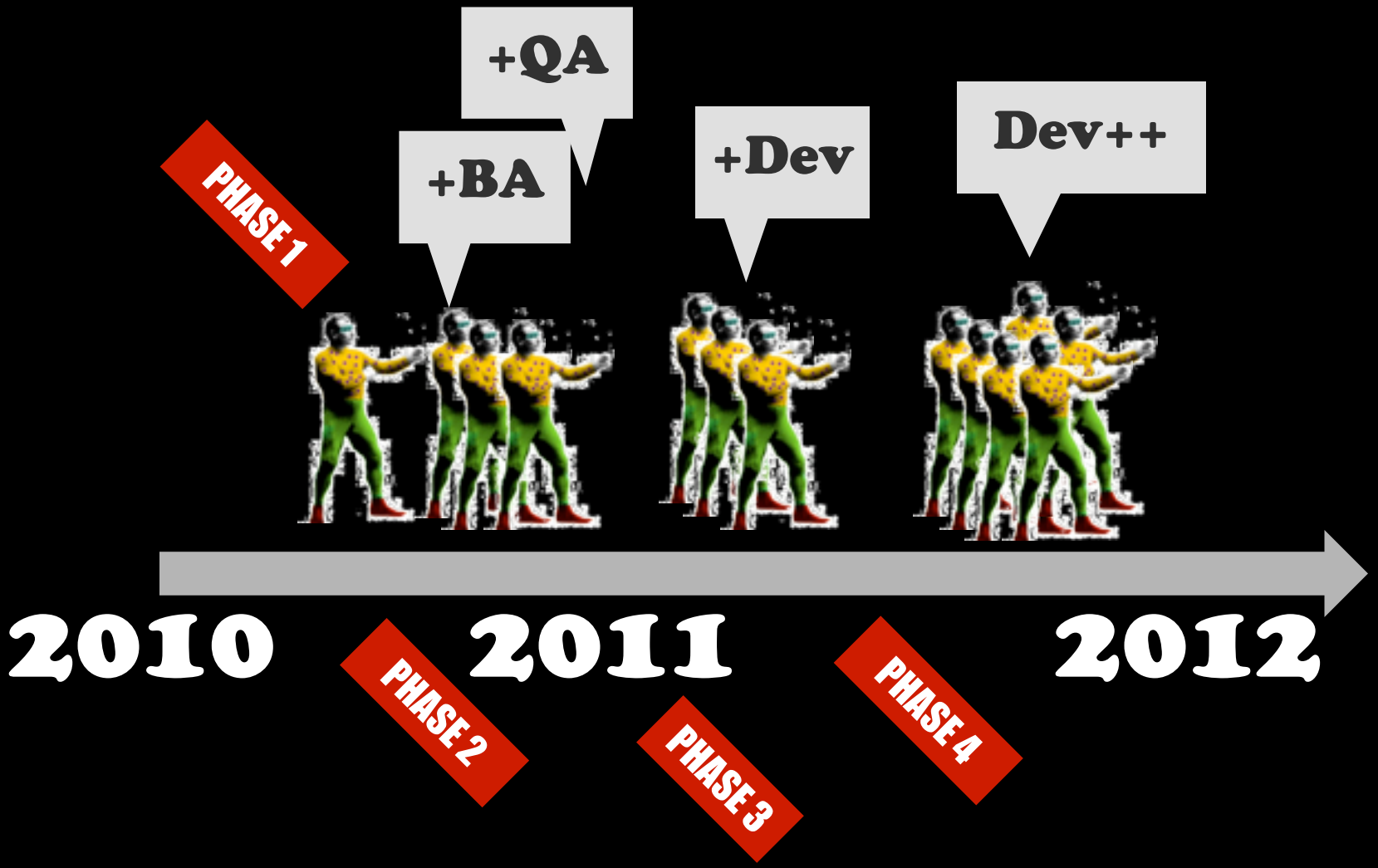
Asset Management


Distributed-Transactions
Legacy
Weblogic
2MLOC
N-tiers
Real-Time
Business
Assset-Management
C++
business-logic
SQL
Java
Swing
cache
Client-Server
EJB2
4GL
High-Volume
Spring
Vendor
spread-over
Application
proprietary
20+years
Batches
Hibernate
Database
Transaction-Scripts

Buy, Partner or Make?

~~Buy, Partner~~ or Make?

Project Life



“You’ve got the right skills
We trust you.
Deliver”

So how do we do that?


We Can Do It!?
WAR PRODUCTION CO-ORDINATING COMMITTEE

PHASE 1

Spike!

3-weeks
Deliberate Discovery in mind

Codeanalysis

[Greg Young]

Within 3 weeks

1. Conversations with experts
2. Code the understanding & note questions
3. Iterate

http://skillsmatter.com/podcast/design-architecture/mystery-ddd/js-4401

Find the Domain Experts

The official expert (former Asset Manager)

Proxy Domain Experts

Proxy Domain Experts

Haz Context?

"Trade" = ?

Sell-Side
(Hedging)

Broker
(A Vs. B)

Buy-Side
(investment)

Accounting
(tax, Basel etc)

+ fees!

Accounting
(tax, accuracy, boring stuff)

Fossilized Domain Expertise

http://newspaper.li/fossil/

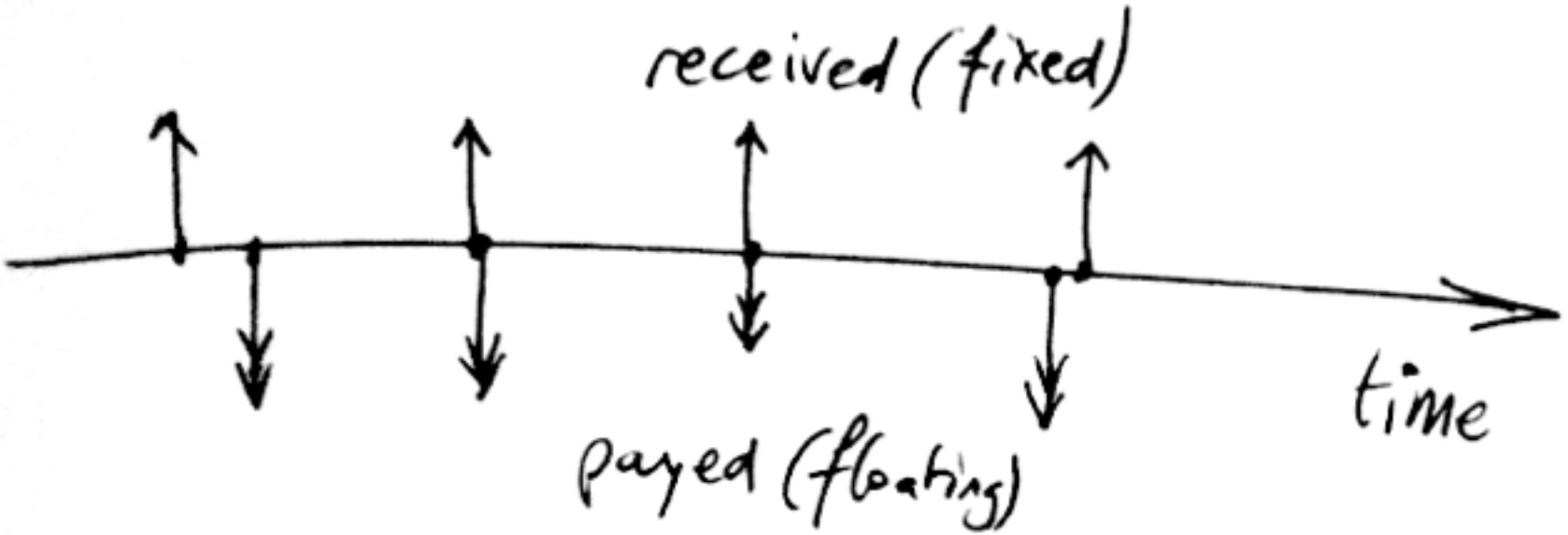
Conversations first

- Ask ***good*** questions
- Build respect and confidence

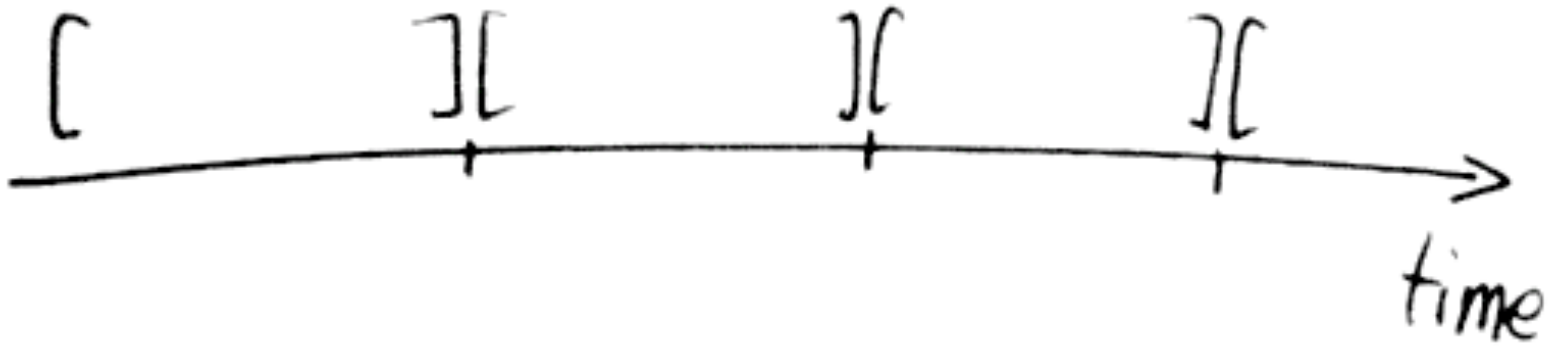
- Beware and care about synonyms and main concepts

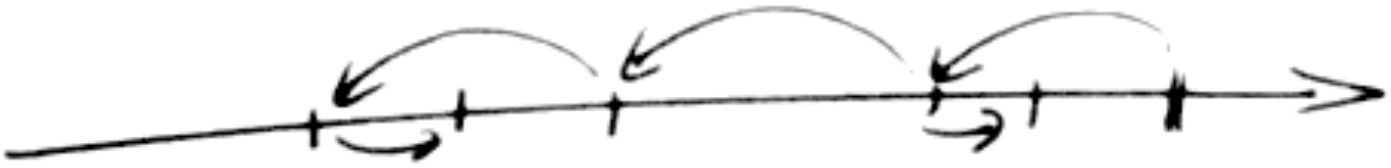
http://journalism.about.com/od/reporting/tp/Finding-And-Developing-Ideas-For-News-Stories-

Sketching

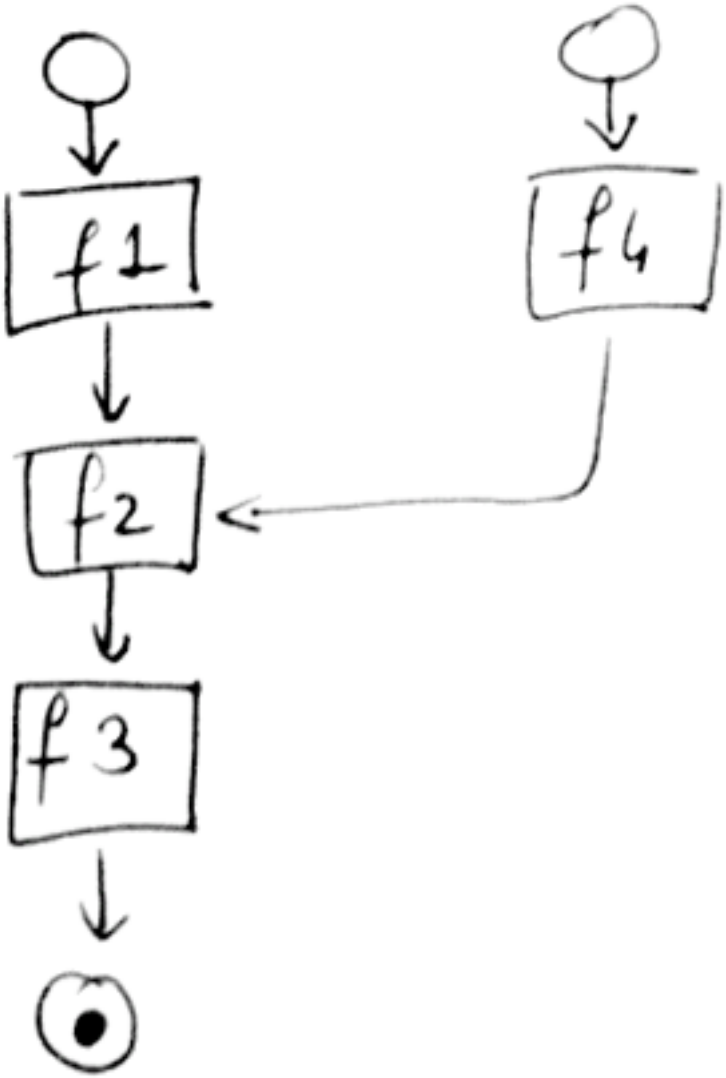


Sketching





Sketching



Growing our Ubiquitous Language

Collecting scenarios

Cash Flow Bank Holiday Interest rate

- An **instrument** is effective for a period of time (its li… is the date range between the effective date and the e… date
- A **swap** is made of **legs**, each of the… g an independ… **instrument** on its own. Each leg def… s two **…ter…cie…**
- A **payoff** is a **cash flow** given by on… …e a… t… other counterparty
- Instruments generate cash flows during their **life**, in particular at **expiry date** (redemption, zero coupon) and at **regular dates**
- There may be a **date offset** for **sett…ments** **…x…s** …d … flows **payments**
- **…loating rate devices** gene… cash flows according to **…bse…at… …ex… ba… o… …ome observable …er…**
- **…x…** …ev…ge… …te… …according to a predefined **coupon**
- **Inflation-linked** devices modulate the coupon, coupon rate or **notional amount** by the variation of some (inflation) observable referen…
- **…egular dates** m… …e… …ne… ex… …a… …t… …t… …epe… ……tes …he… …r… …l a… …s… …ck… …d o… …wa… …w… …t… **…onv…ion** …te **…us… …t… …es** …d a… …di… …s… **…da… …alend…**
- Date adjustment rules are defined by a **day roll conv…ti…** and a **business center**
- A **funding leg** is a fixed rate device or a floating rate device
- A **return leg** is a leg that reproduces the exact cash flows of another instrument

- *Typical credit events are **bankruptcy**, **failure to pay**, **restructuring***
- *Forward and Future means that the **effective date** is in the future. The difference between forward and future is that the former is OTC whereas the later is cleared, more standard, and involves frequent **margin calls** (hence a convexity bias between them)*
- *From a **legal** perspective, an **option** is the **right to exercise** some transaction, typically to **buy/sell** the underlying instrument at a predefined **strike** and volume, or to receive a **payoff**. Since investors are rational, we may also consider options to be essentially instruments with conditional transaction, in general payoffs*
- ***Option styles** are typically **European** (exercise only at expiry date), **American** (exercise anytime), **Bermuda** (exercise at regular dates), **Canary** (exercise at regular dates after some date or period)*
- *An **underlying** instrument is an instrument used to condition the exercise (which may be automatic) and often used to calculate the amount of the payoff or more that is directly the object of an exchange, in the case of an exchange of assets*
- *Some instruments are subject to **conversion**, i.e. they can be replaced with another instrument*
- *Some … …e… …vo… …me… …w… or no **exchange(s) of principal …s)***
- *Compounding swaps capitalize interests earned for each period payment period*

Taking notes, in code

Class, interface names & documentation

Fields

Relationships (if any)

+ Behavior!

Code doesn't lie

Got something wrong?

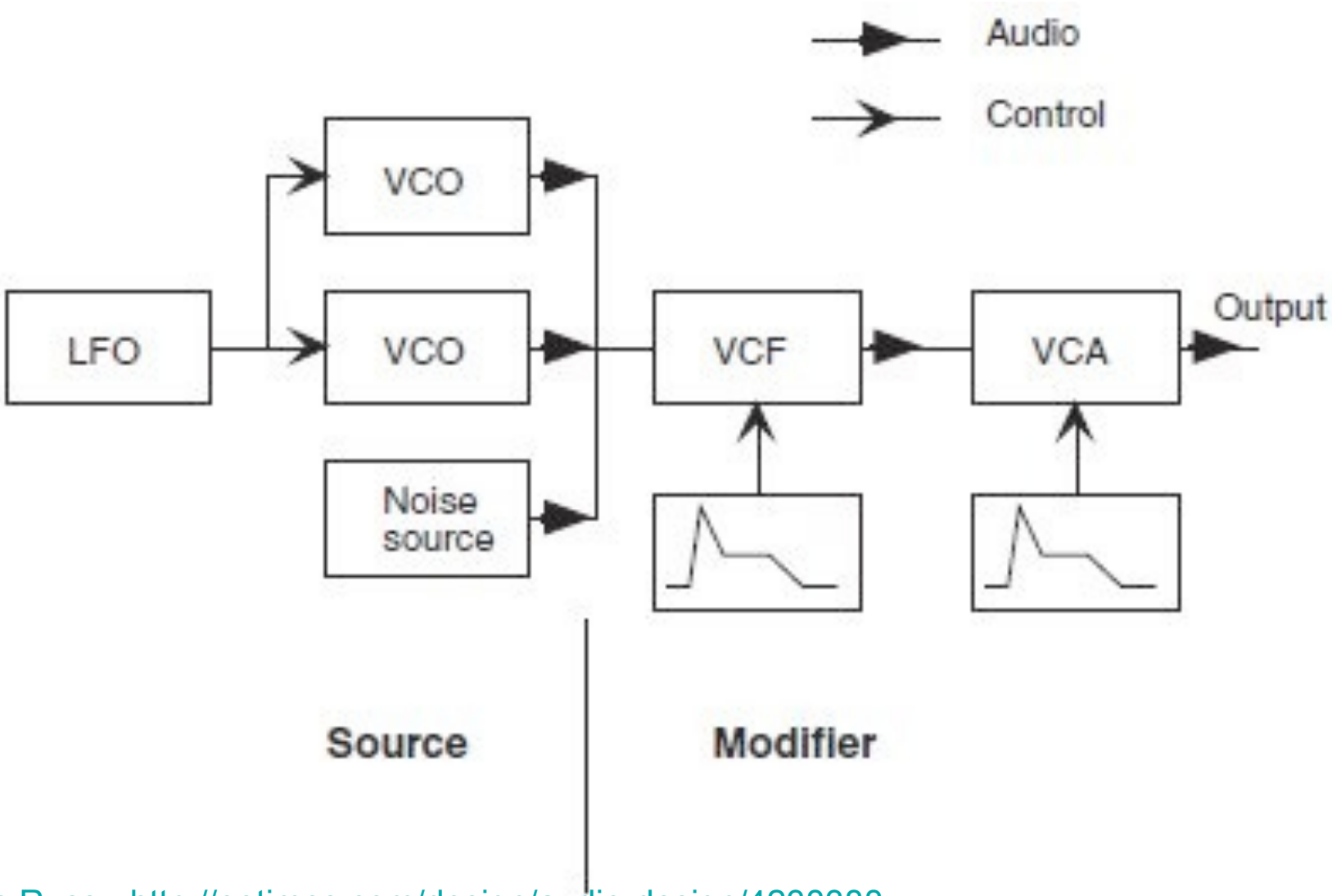
Forgot something?

It shows!

Found a metaphor

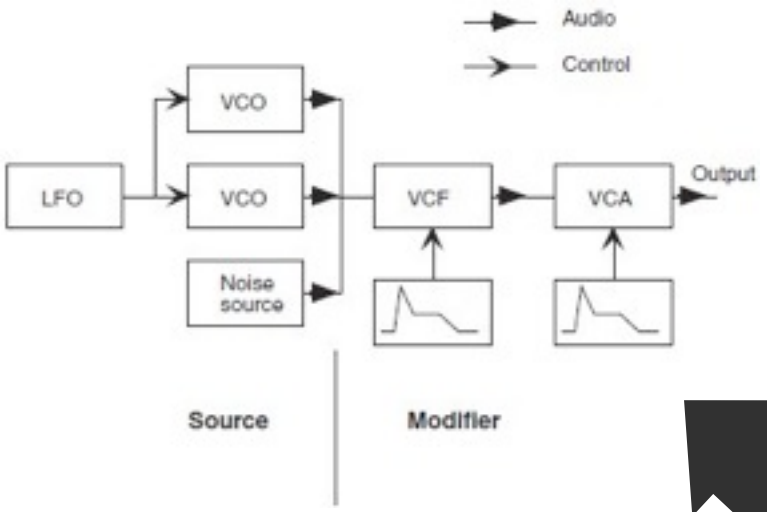
http://www.marksmart.net/gearhack/pogomoog/pogomoog.html

Modular Synthesizer

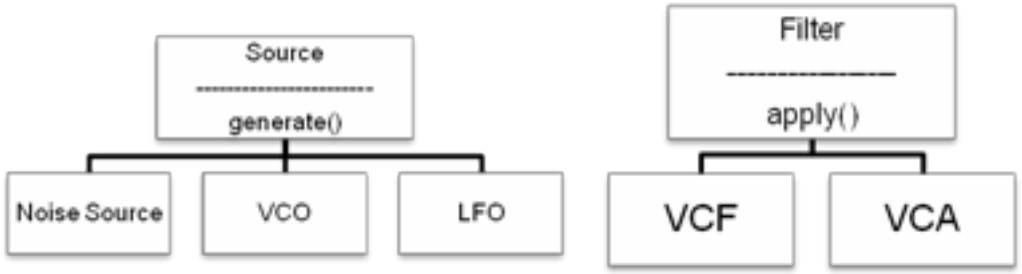


Martin Russ - http://eetimes.com/design/audio-design/4238388

In Java code



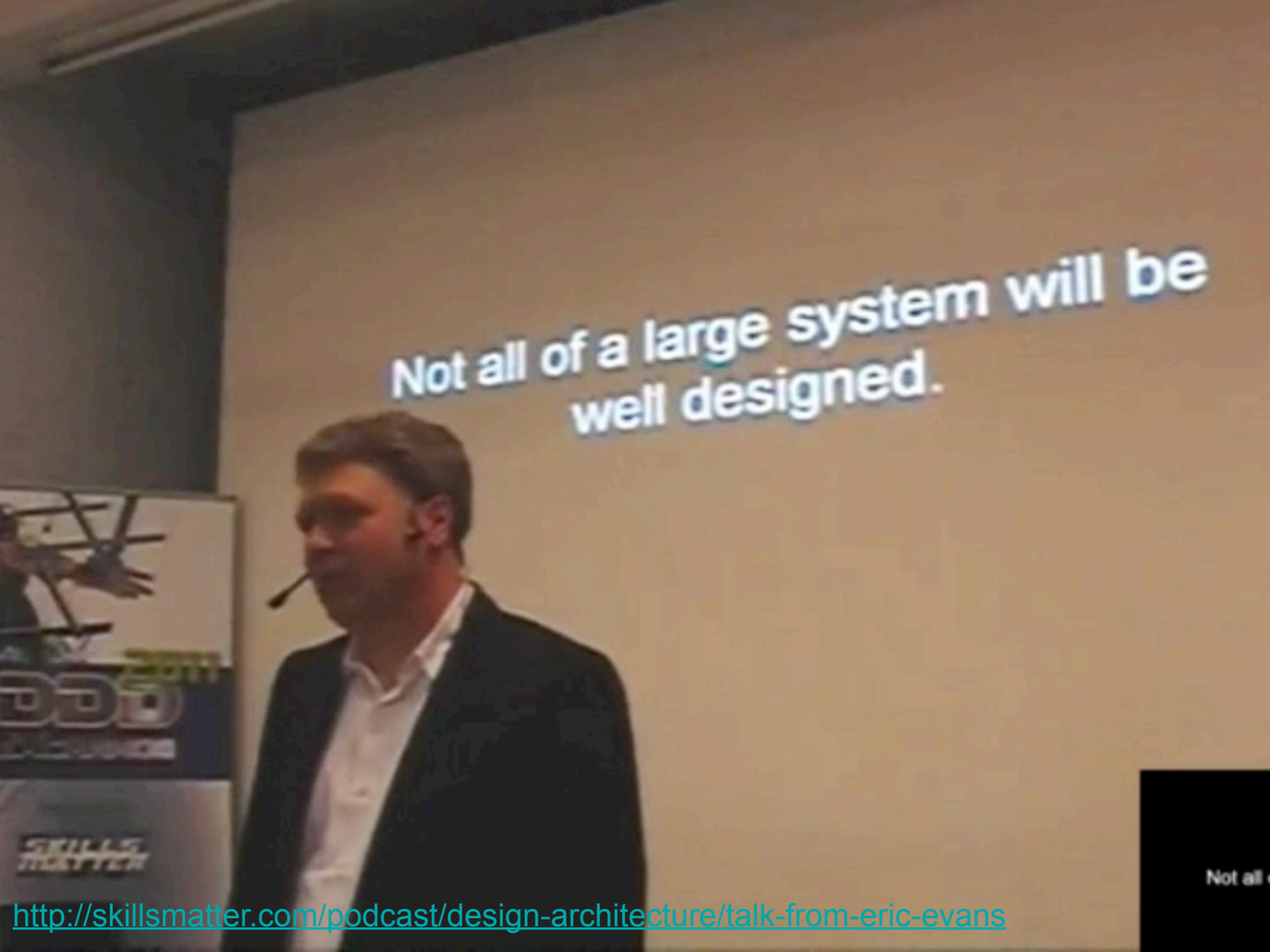
Output of a class/ function is the **input** of another




We Can Do It!
WAR PRODUCTION CO-ORDINATING COMMITTEE

So let's keep on coding and rebuilding everything!

Sounds like a great idea!



http://skillsmatter.com/podcast/design-architecture/talk-from-eric-evans

Plenty of Legacy code working since 1993

Redoing = waste + risk

Still need to redo some parts

How do we integrate with legacy?

Where do we stop redoing?

PHASE 2

Legacy strategy

Know your enemies

Asset Capture & Strangler Application

Bubble Context & ACL

Scream it loud

Know your enemies
[discover the app]

Read the documents

Interview people

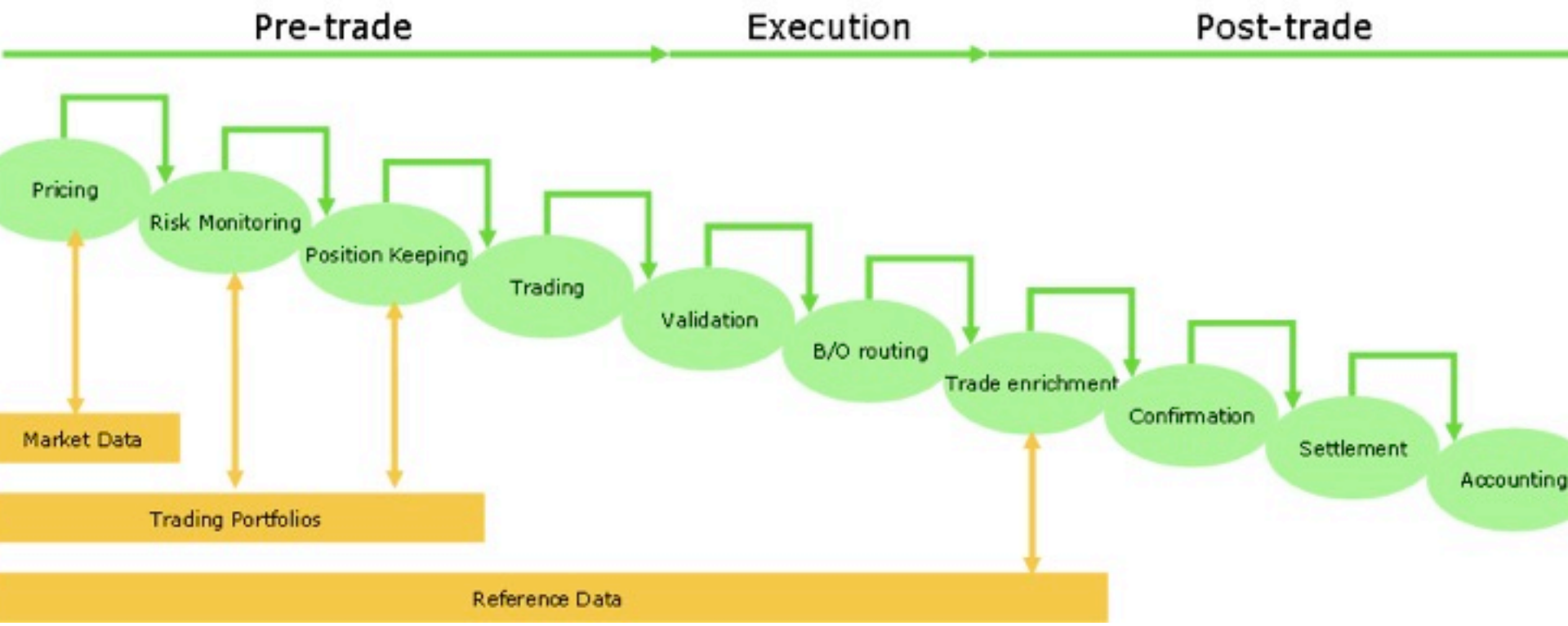
Use the application

Read the code

Fix bugs

Visualize

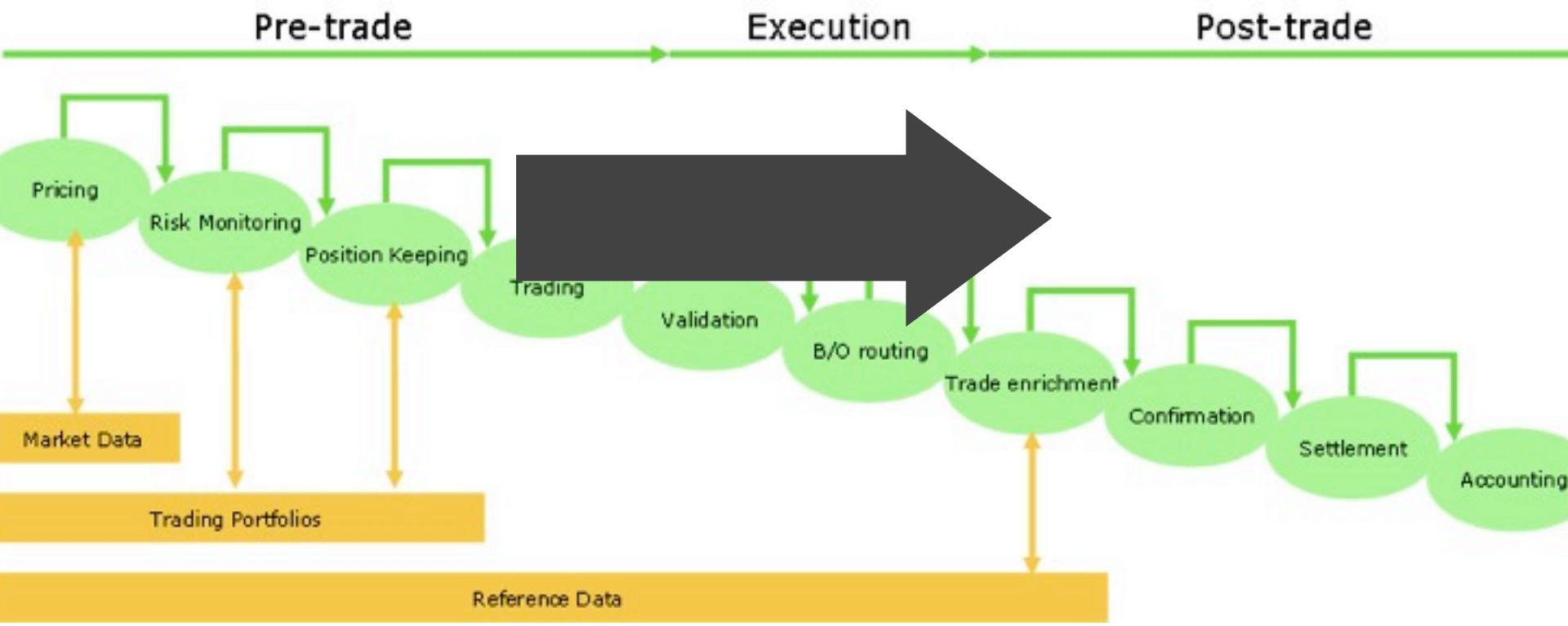
[primary flow & support features]



http://www.fimarkets.com/pagesen/back-middle-front-office.php

Visualize

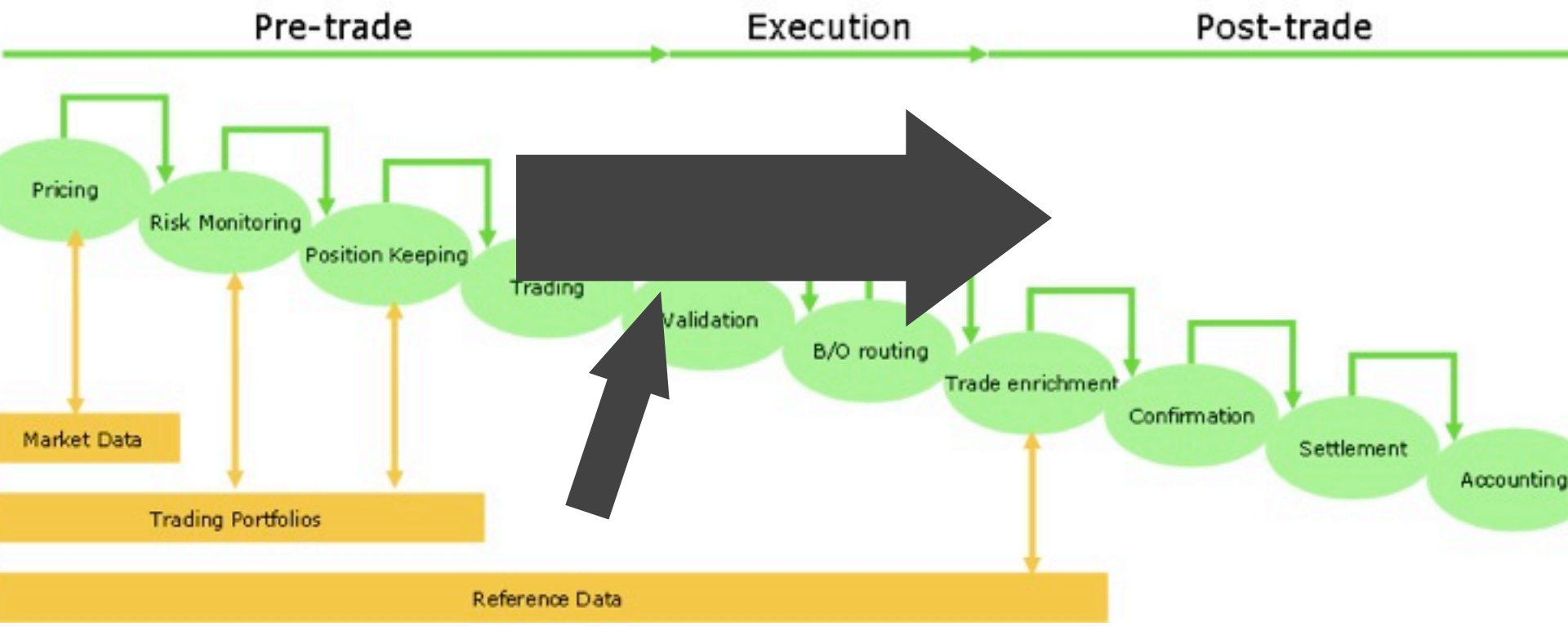
[primary flow & support features]



http://www.fimarkets.com/pagesen/back-middle-front-office.php

Visualize

[primary flow & support features]

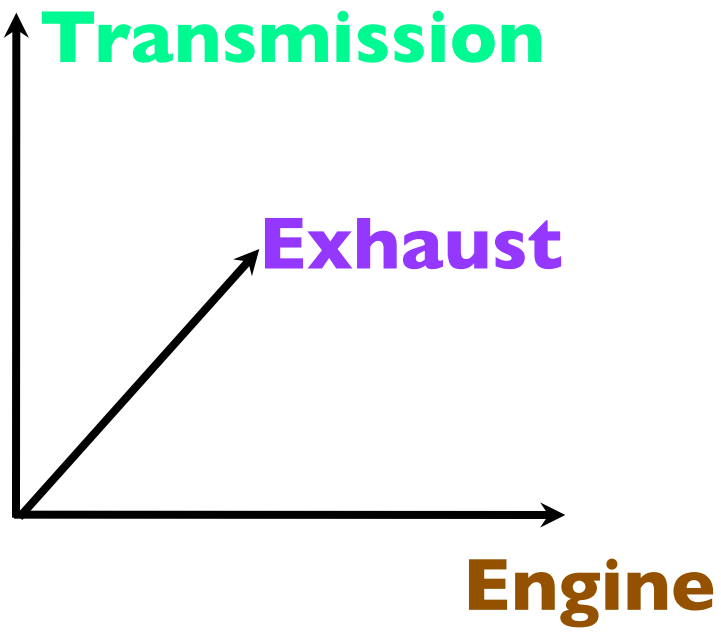


http://www.fimarkets.com/pagesen/back-middle-front-office.php

Asset Inventory

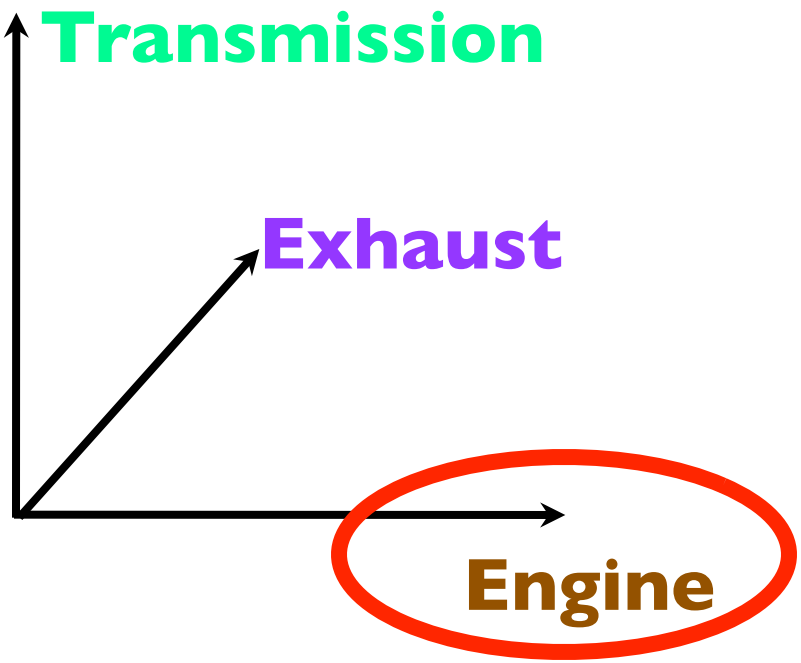
http://staff.bath.ac.uk/enssa/LPDev.htm

Asset Capture



http://martinfowler.com/bliki/**AssetCapture**.html

Asset Capture



Focus on one single asset : the engine

http://martinfowler.com/bliki/**AssetCapture**.html

Strangler application

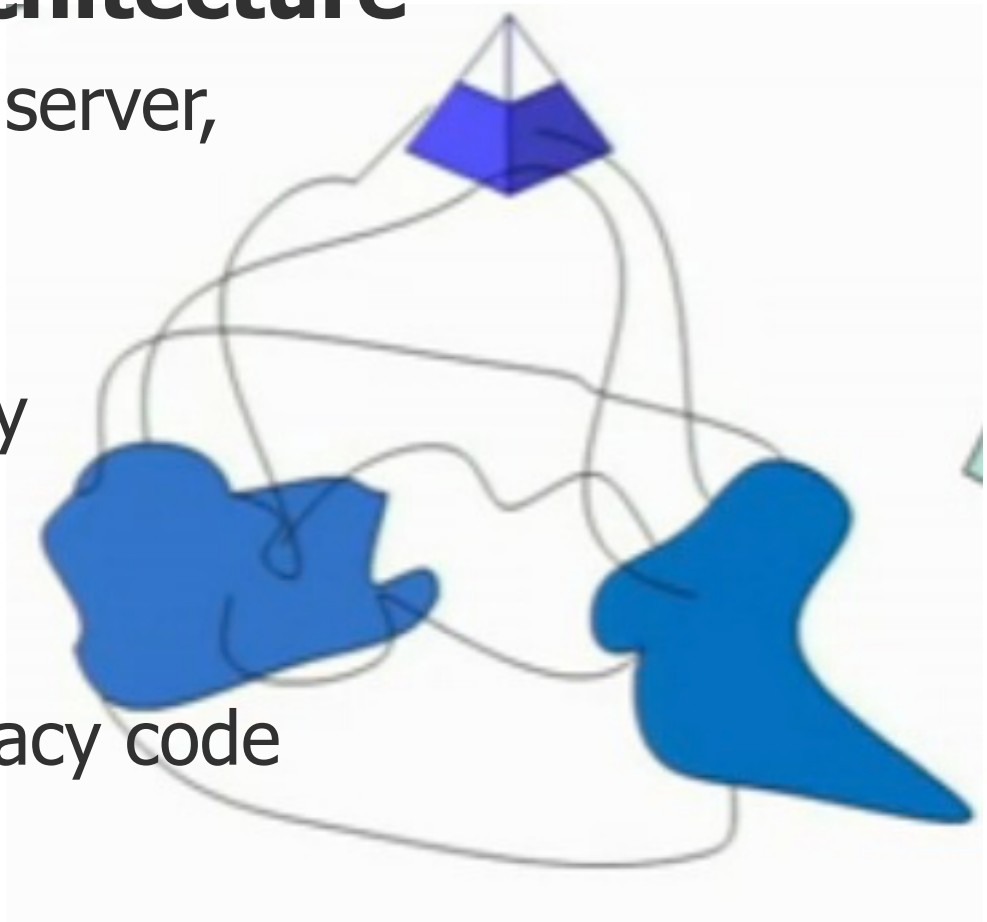
A new module will progressively strangle the former one, for one single asset only

http://martinfowler.com/bliki/**StranglerApplication**.html

http://www.flickr.com/photos/louisfoecy/4114597043

Conservative by default

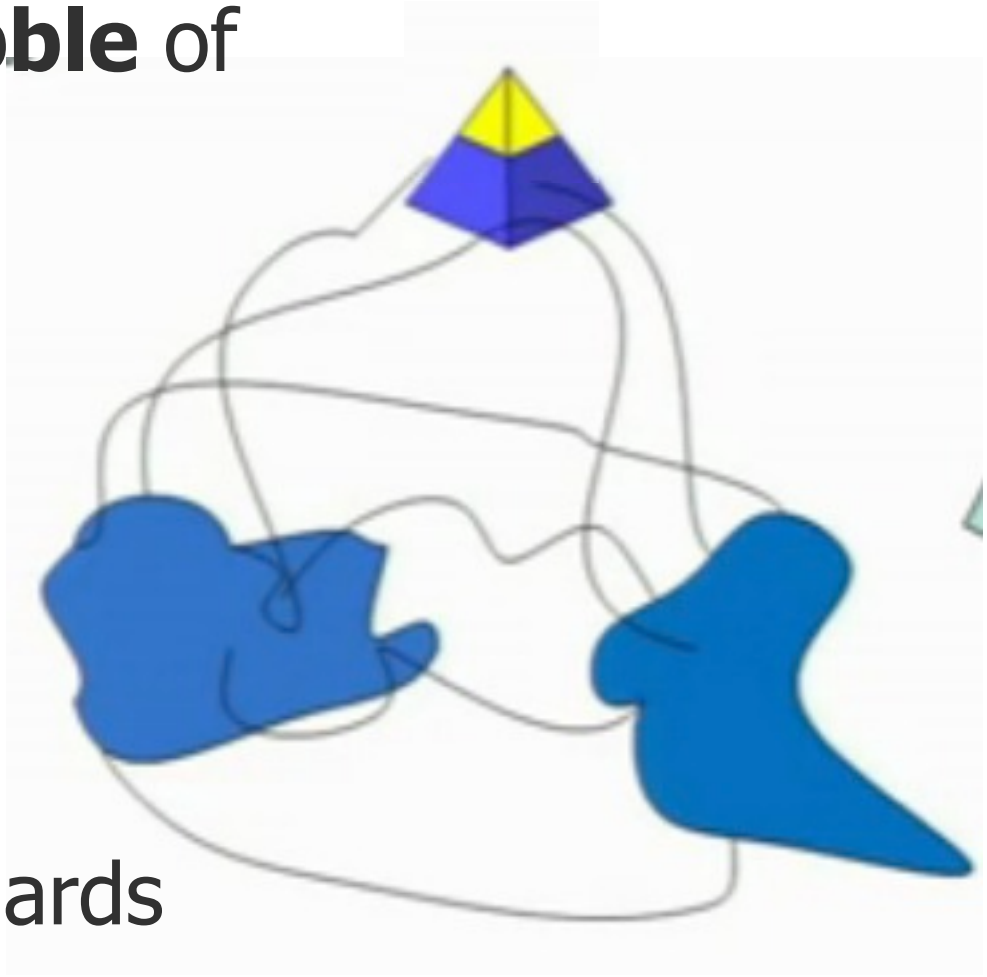
- **Keep existing architecture**
  - N-tiers, application server, middleware
  - Same database
  - Same UI technology
  - (Almost) all the legacy code



Diagrams shamelessly stolen from Eric Evans DDDx 2011 talk

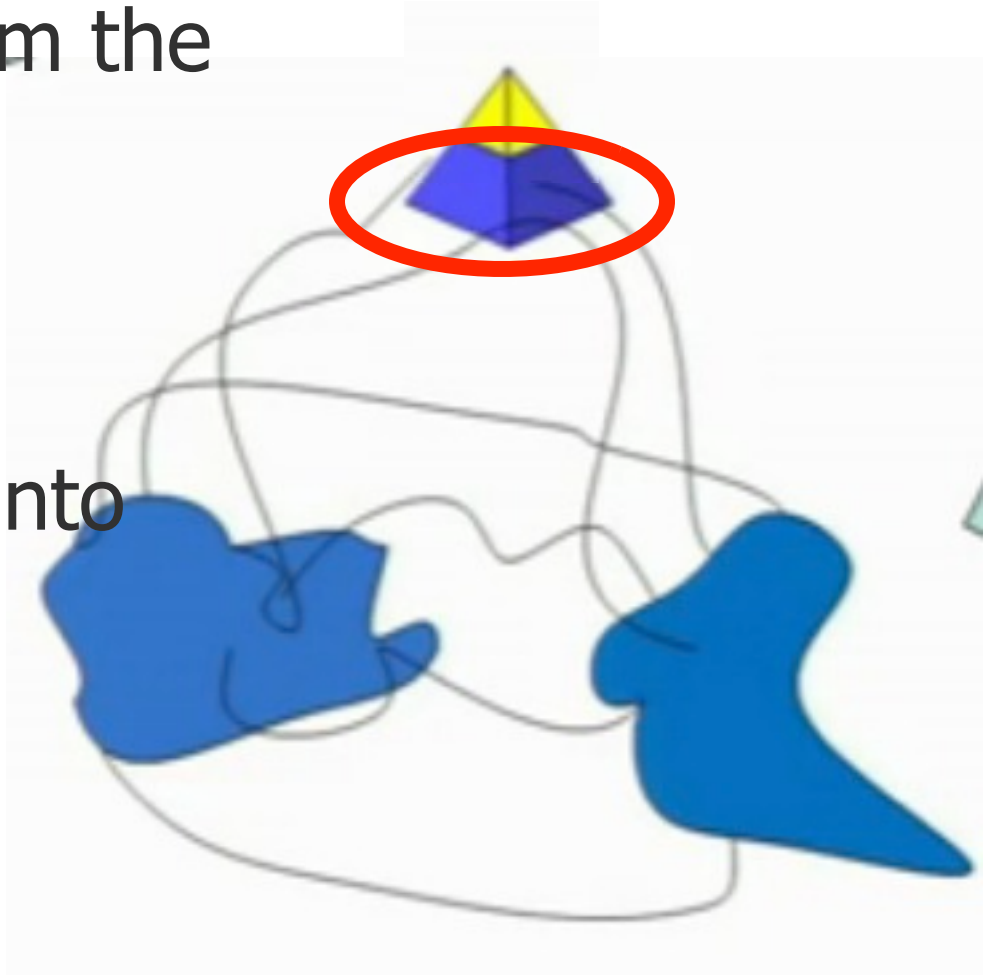
Bubble Context

- Create a **little bubble** of innovation
- Brand **new model**
- As clean as we like
- Highly tested
- High hygiene standards



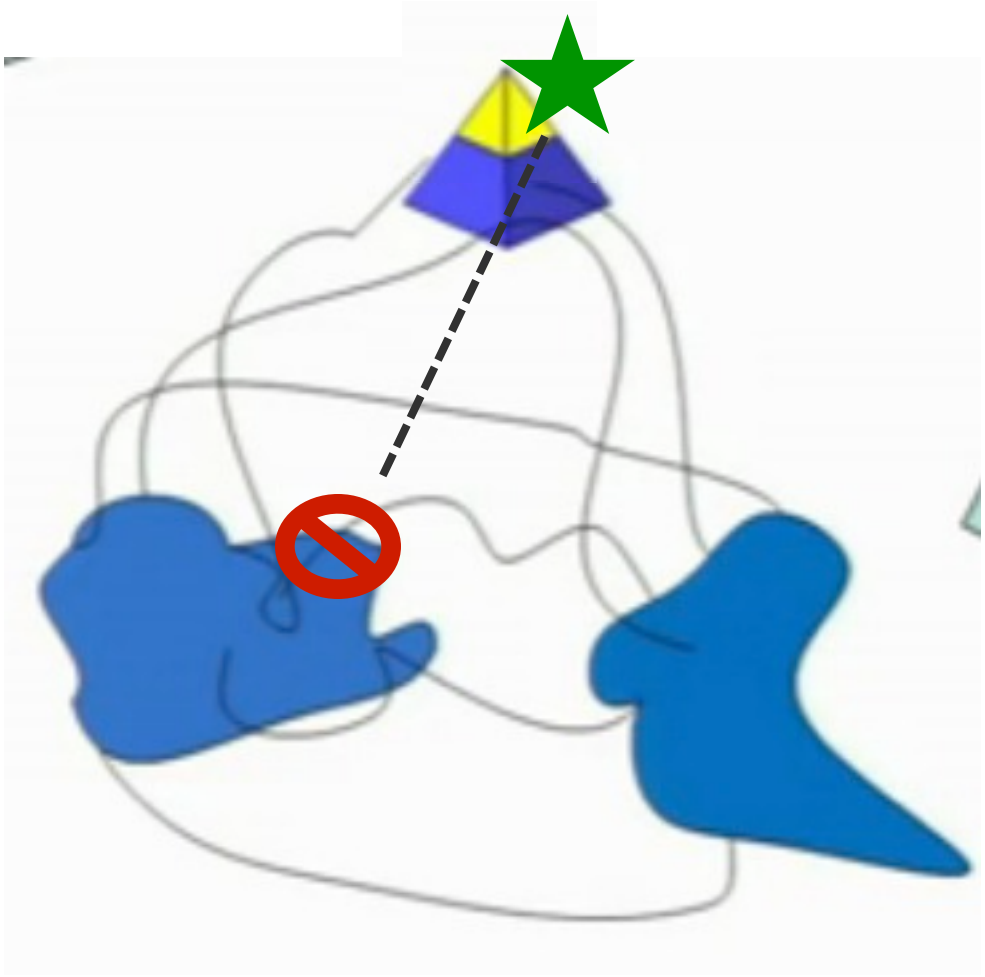
Bubble Context

- ACL to **protect** from the legacy code
- **Translates** legacy objects & services into new, cleaner ones

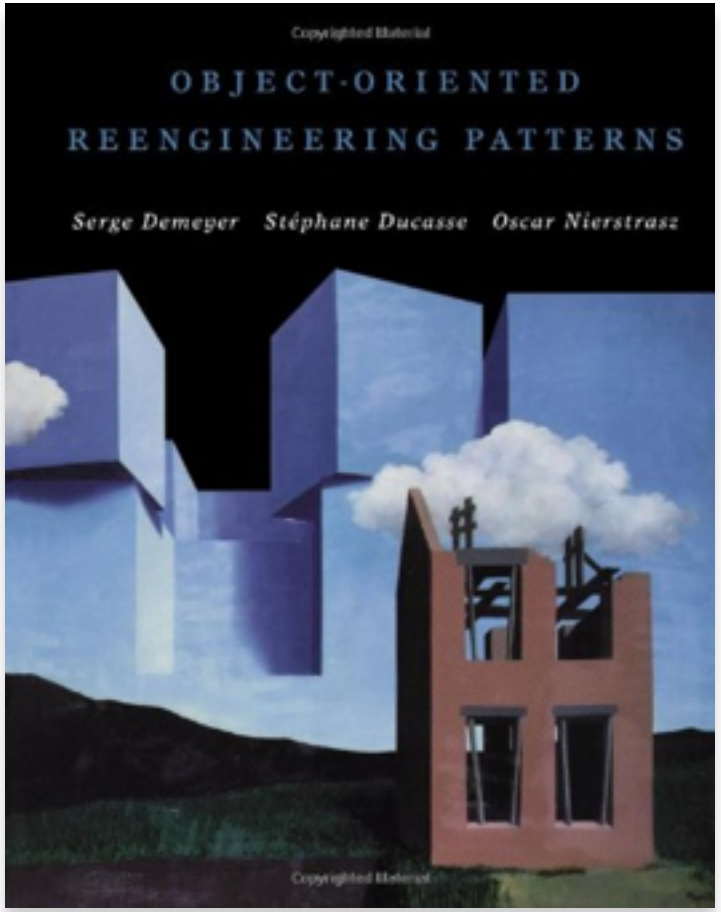


Strangler (cont.)

- Progressively **turn off** **legacy** functionalities & **turn on** the **new** ones



Agree on maxims
[communicate the plan]



- "Only one work site at a time: instruments"
- "When in Rome, do as Romans do"

PHASE 3

Scaling development in the bubble context

Infect the team

Code as SPOT

High hygiene standards

Functional-style

LITTLE MISS CURIOUS

by Roger Hargreaves

LITTLE MISS CURIOUS

by Roger Hargreaves

I only care about Java EE 6

LITTLE MISS CURIOUS

by Roger Hargreaves

Hey! Finance is cool!

LITTLE MISS CURIOUS

by Roger Hargreaves

LITTLE MISS CURIOUS

by Roger Hargreaves

Oh, this domain is really cool!

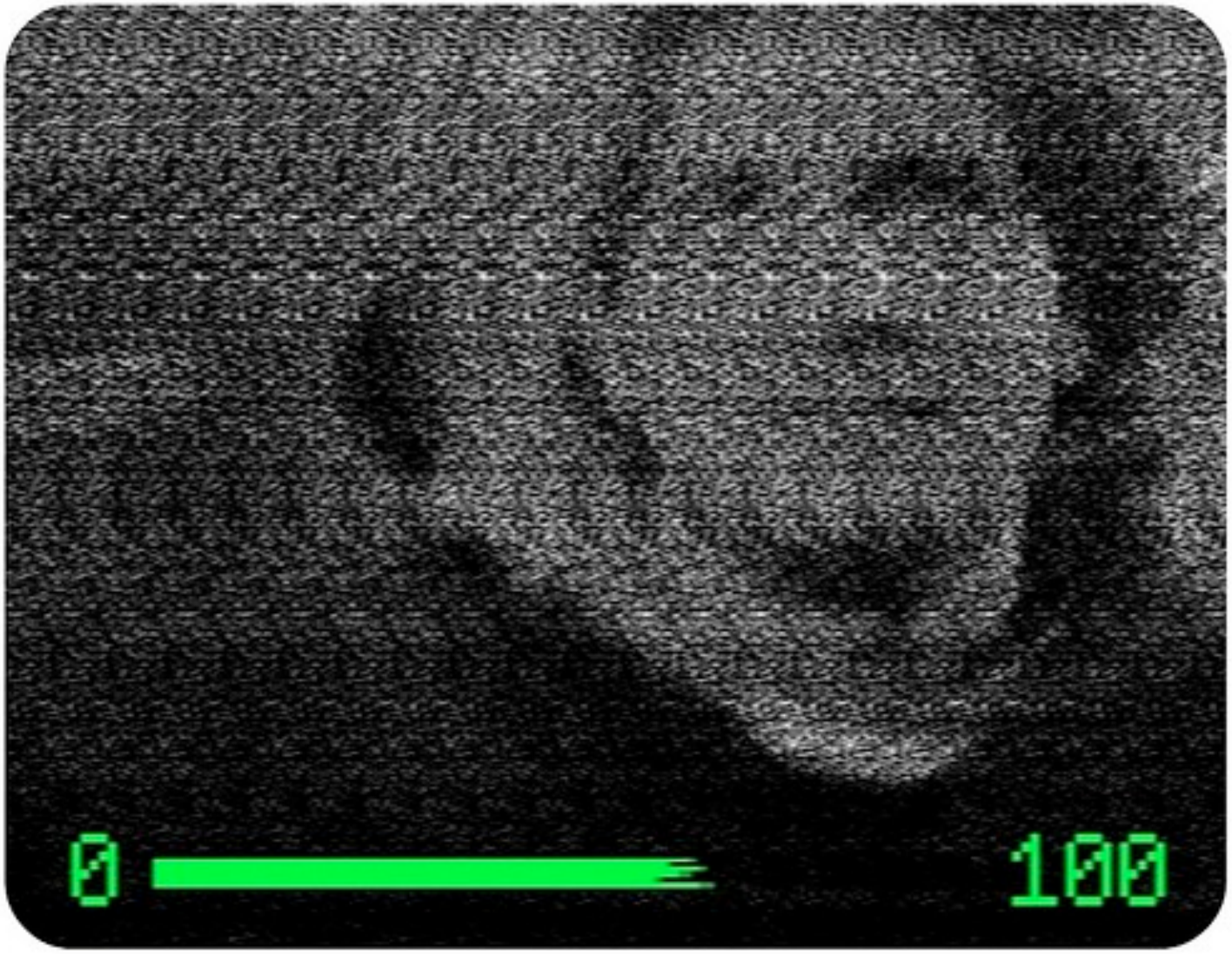
LITTLE MISS CURIOUS

by Roger Hargreaves

Invest in domain knowledge

- Bi-weekly 30mn training sessions

Signal to Noise ratio



http://www.flickr.com/photos/28471130@N07/2666802097

Signal to Noise Ratio

- **SNR ≥ 80 %**
- **Signal**: **`CashFlow, CashFlowSequence, CashFlowEngine, FinancialProduct, BankHolidays, ReferenceData`**
- **Noise**: `CashFlow`**`Builder`**`,` **`Composite`**`Engine, Product`**`Factory`**`,` `BankHolidays`**`Decorator`**

“All what you explained is nice, but what about the actual code and design?”

-client

Code as reference
[export the glossary]

- Annotate domain-relevant classes

```
 */
@Convention
public class DayShift {
```

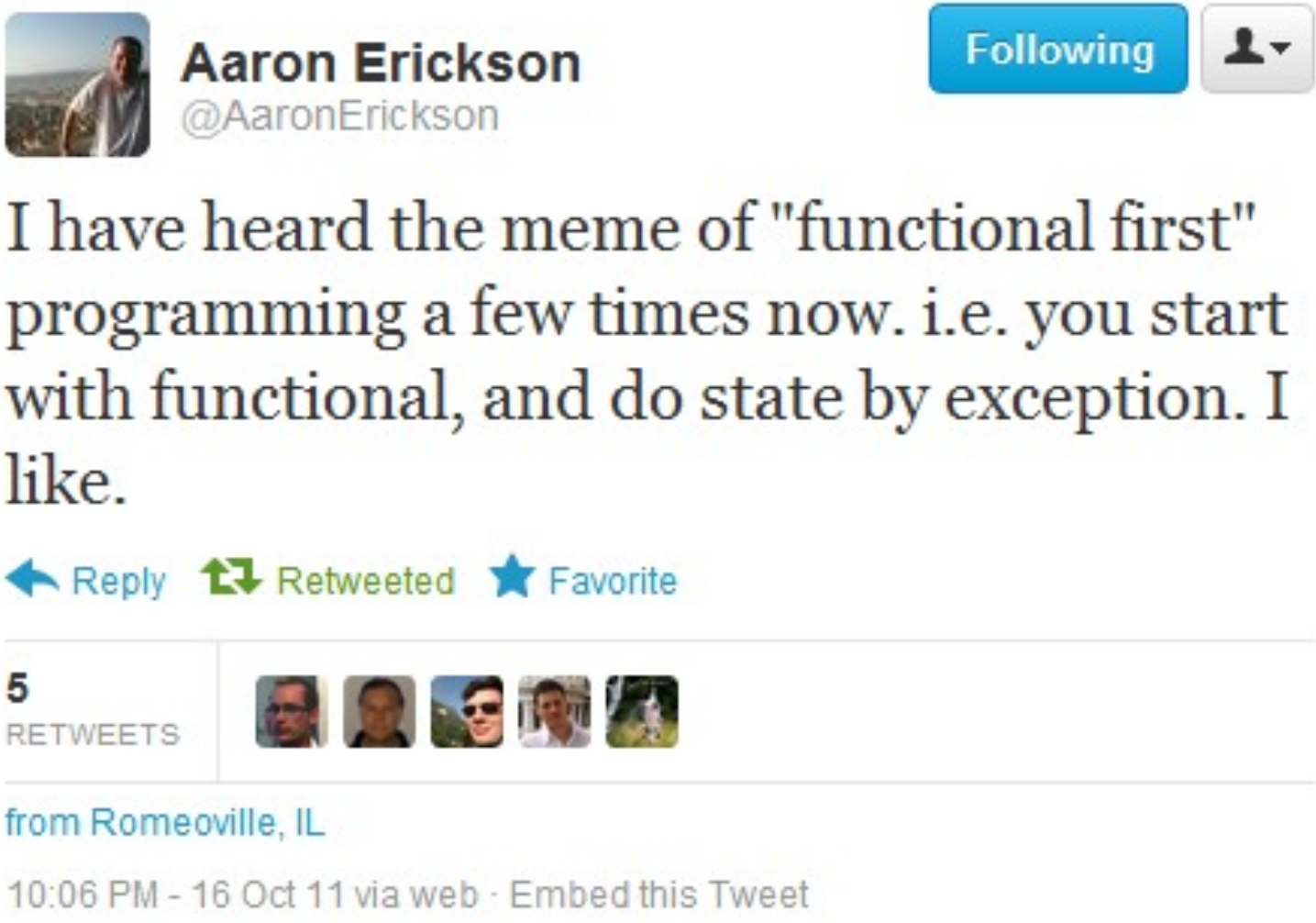
- Custom Doclet to export **Excel-formatted glossary** of every domain concept
- Send directly **to end customers** as reference documentation

High hygiene standards
[in the bubble context]

- Dependencies
  - ~~Legacy~~
  - ~~Middleware~~
  - ~~Frameworks~~
  - ~~Database stuff~~

+ TDD, BDD
+ Clean Code

“Functional-First” style



Aaron Erickson
@AaronErickson

Following

I have heard the meme of "functional first" programming a few times now. i.e. you start with functional, and do state by exception. I like.

Reply Retweeted Favorite

5
RETWEETS

from Romeoville, IL

10:06 PM - 16 Oct 11 via web · Embed this Tweet

“Functional-First” style

- **Value Objects** (Quantity, Range…) as much as possible
  - Immutable
- **Side-effect-free** functions
- **Closure of operations**
  - `CashFlow` and `CashFlowsSequence` add(), negate(), multiply()

PHASE 4

Legacy: the bigger picture

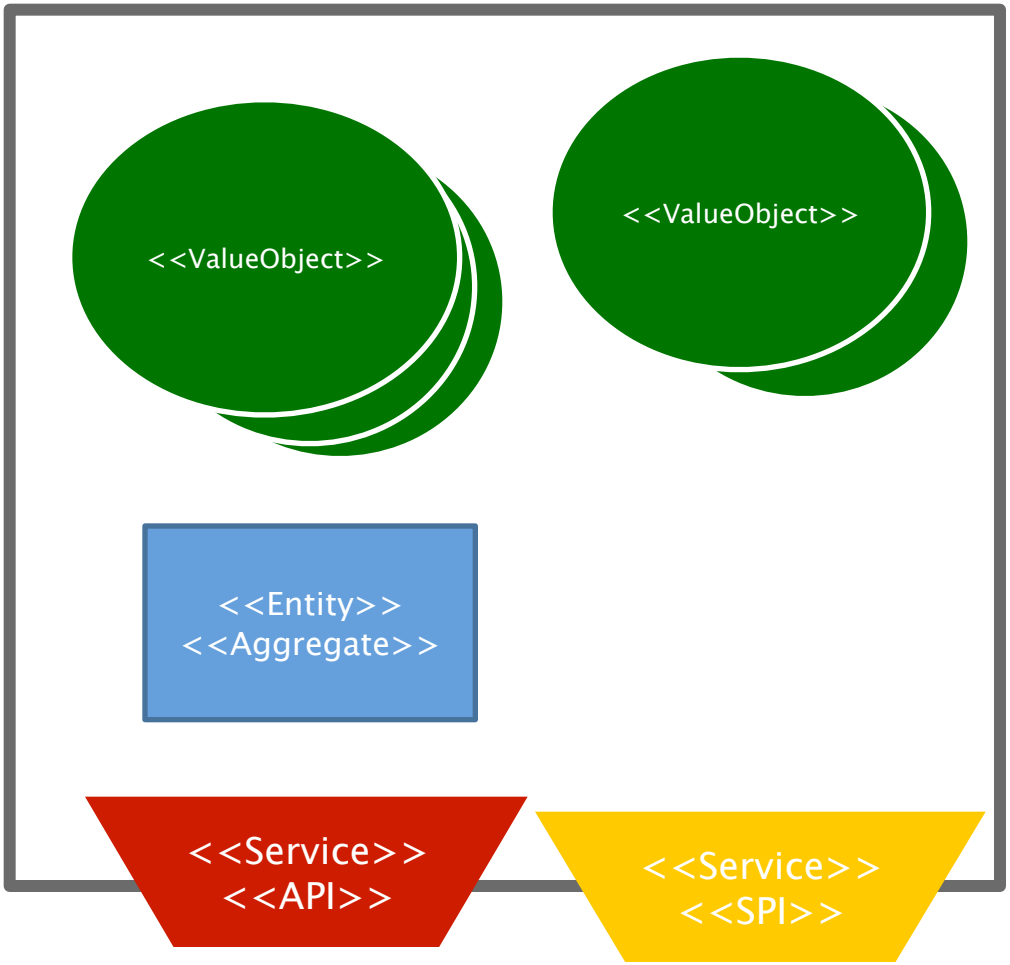
Clarify bubble boundaries

Emerging contexts

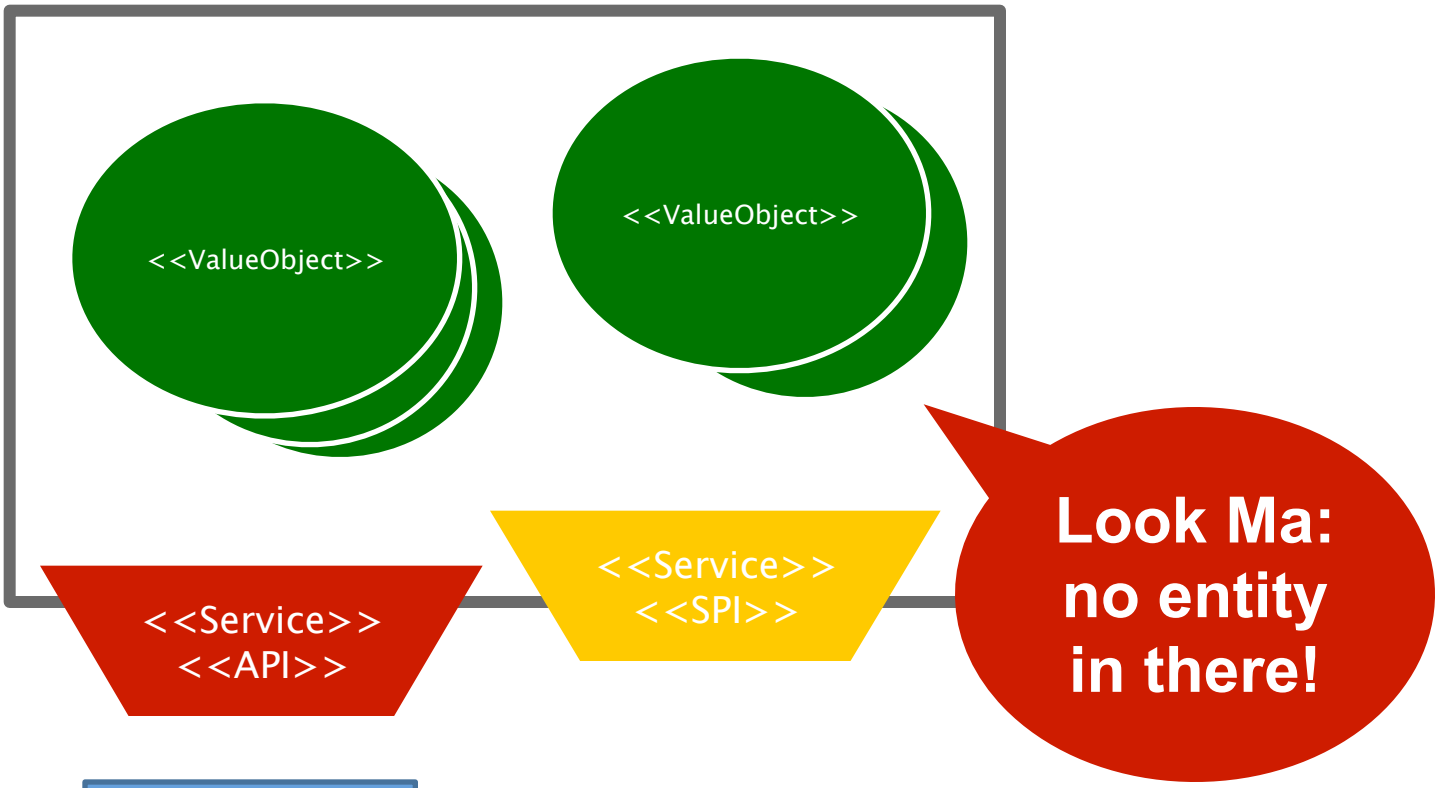
Shock absorber

Responsibility layers

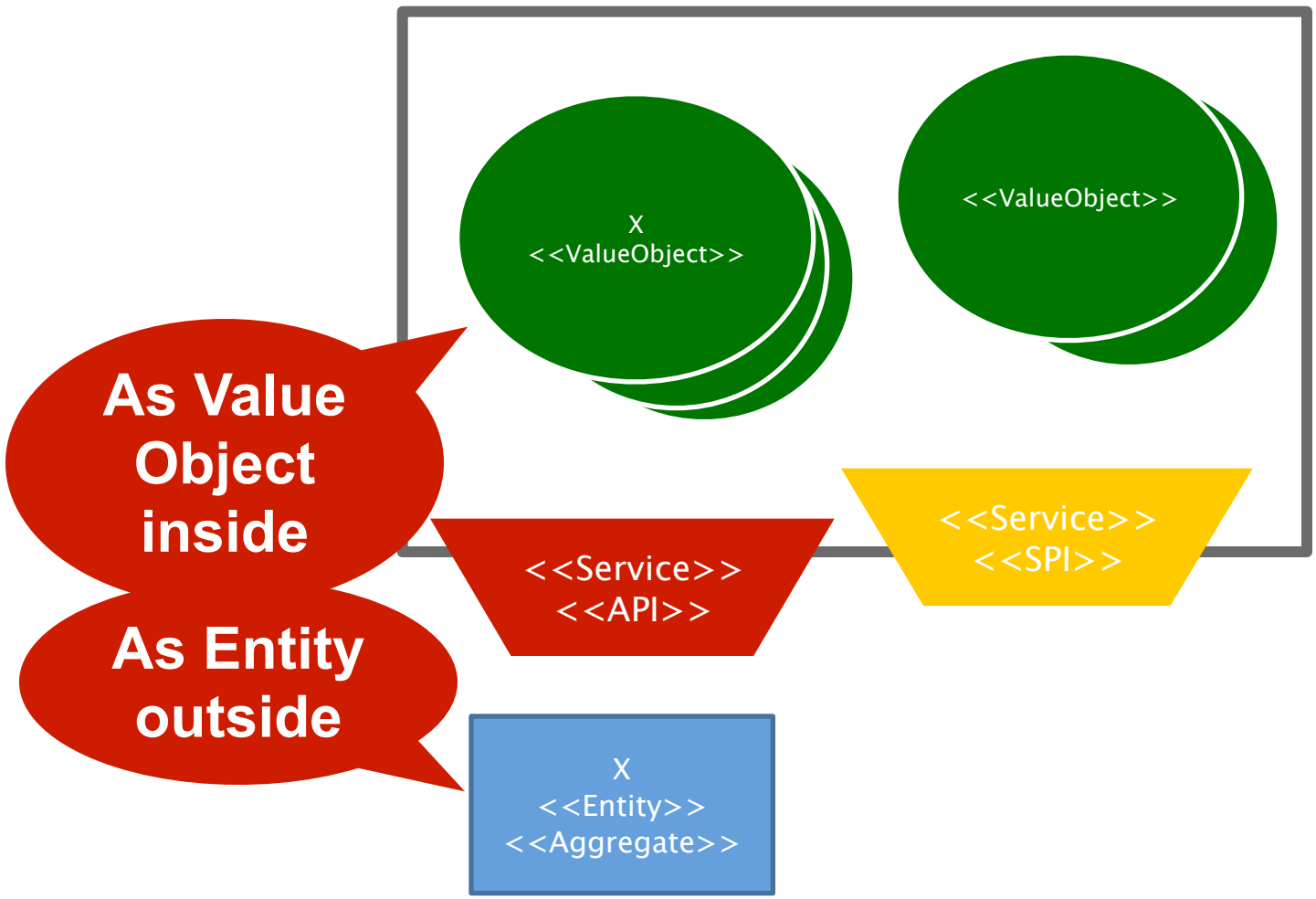
First boundaries



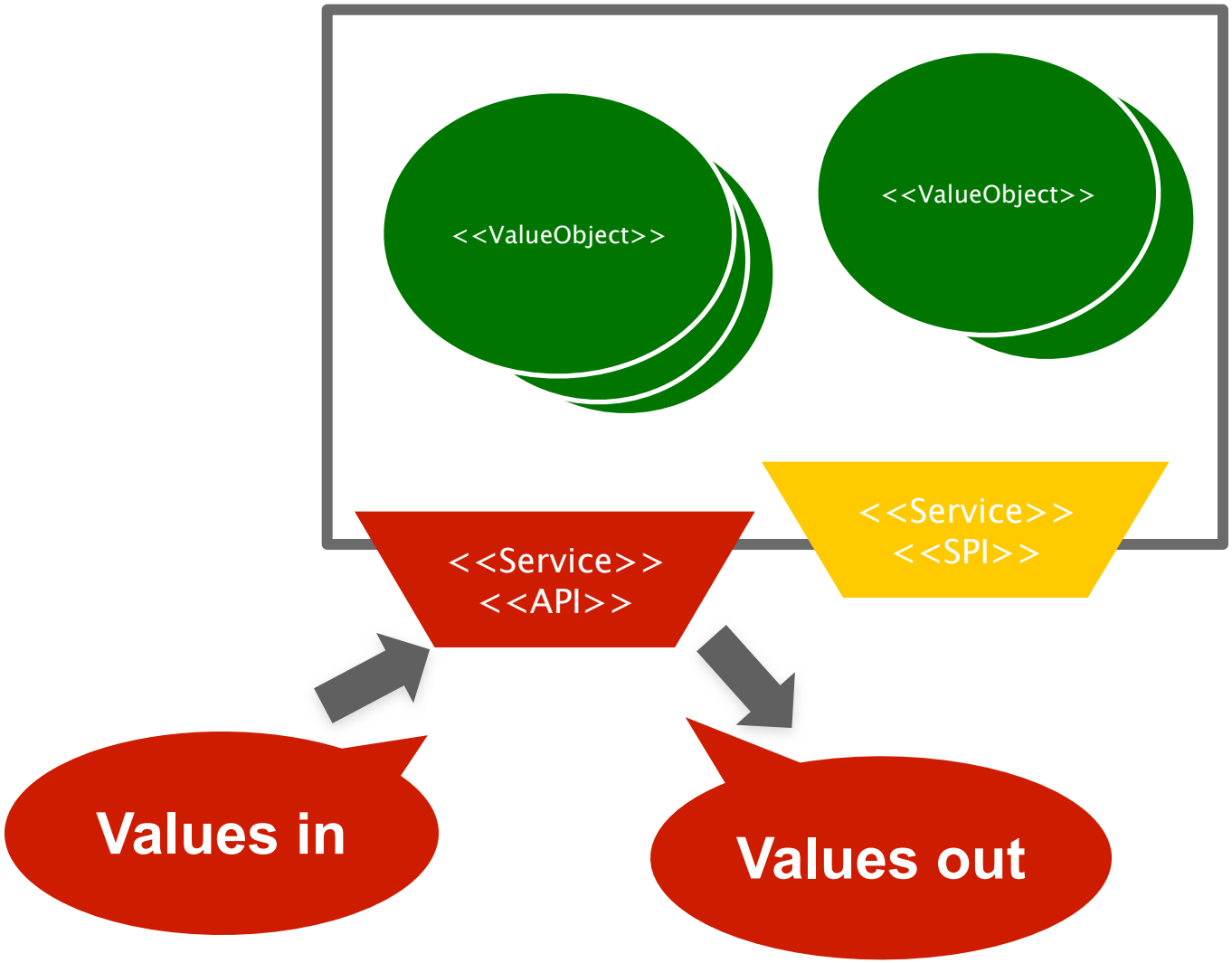
boundaries 2.0



boundaries 2.0



Just a lib & side-effect free!



```java
package com.arollafterwork.ddd.domainmodel;

/**
 * Generates cash-flows for the given instrument, according to the
 * external data provider.
 */
@Service
public interface CashFlowGeneration {

    CashFlows generate(final Instrument instrument, final ExternalData externalData);
}
```

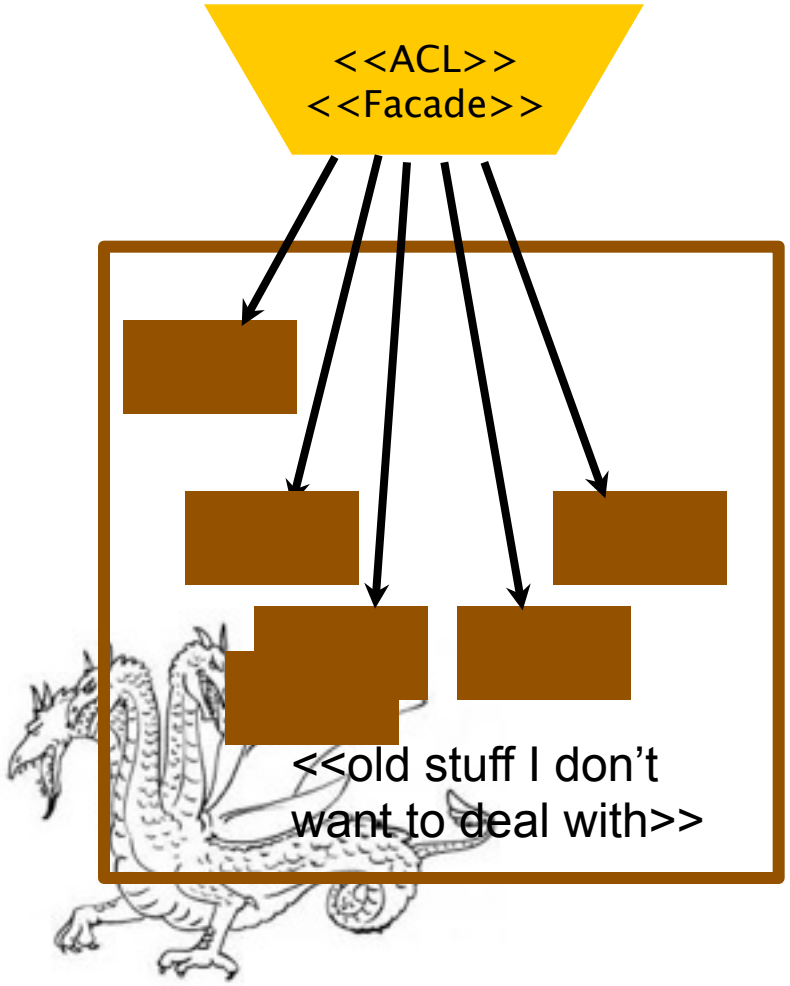
API

```java
package com.arollafterwork.ddd.domainmodel;

import java.util.Date;

/**
 * The interface to provide on-demand external data services
 */
@Service
public interface ExternalData {

    Object valueFor(final Object key, final Date date);
}
```

SPI

ACL-backed Domain service

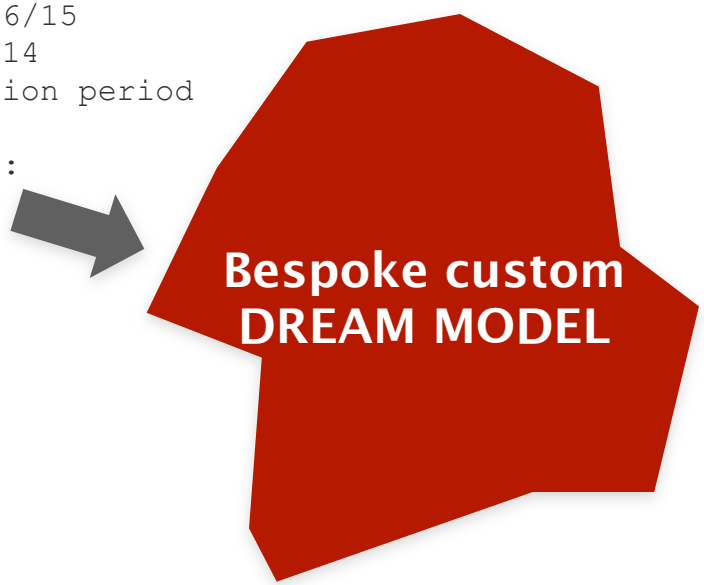
- Unify ***many*** legacy services through **one single & simple interface**
- **Hide** the most unfriendly legacy code



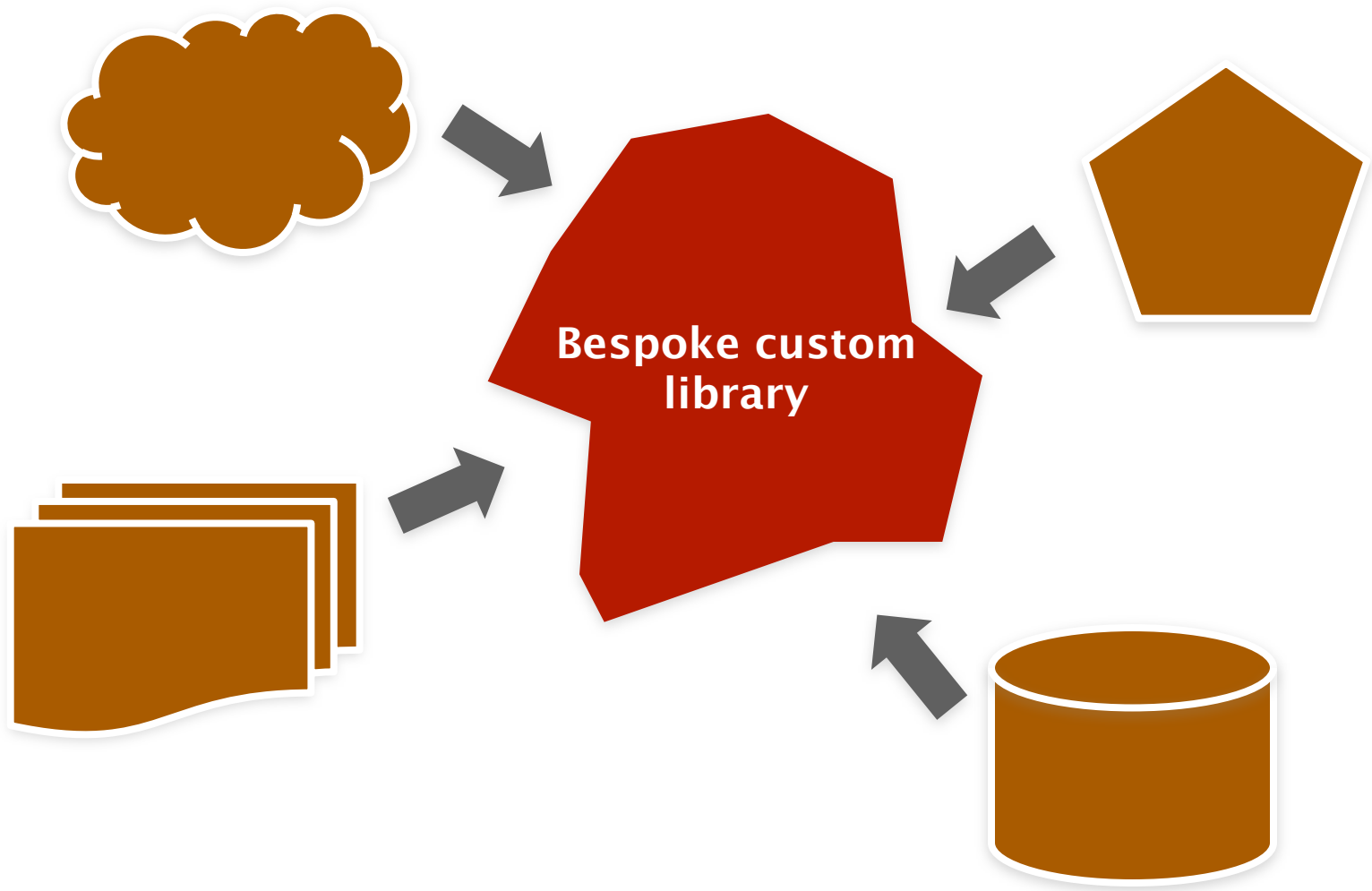
Initial journey toward our dream model

Dream model must pass every scenario

```
Given a floating rate bond on EURIBOR 3M
And a nominal of 15M EUR
And an issue date of 2011/06/15
And an end date of 2012/06/14
And an SEMI_ANNUAL calculation period

When the EURIBOR 3M evolves:
     | 2011/09/15 | 3.5% |
     | 2012/03/15 | 4.0% |

Then the cash-flows are:
     | 2011/12/13 | 23000 |
     | 2012/06/14 | 25500 |
```



Deliberately easier by construction



Conflicting aspects suggest several contexts

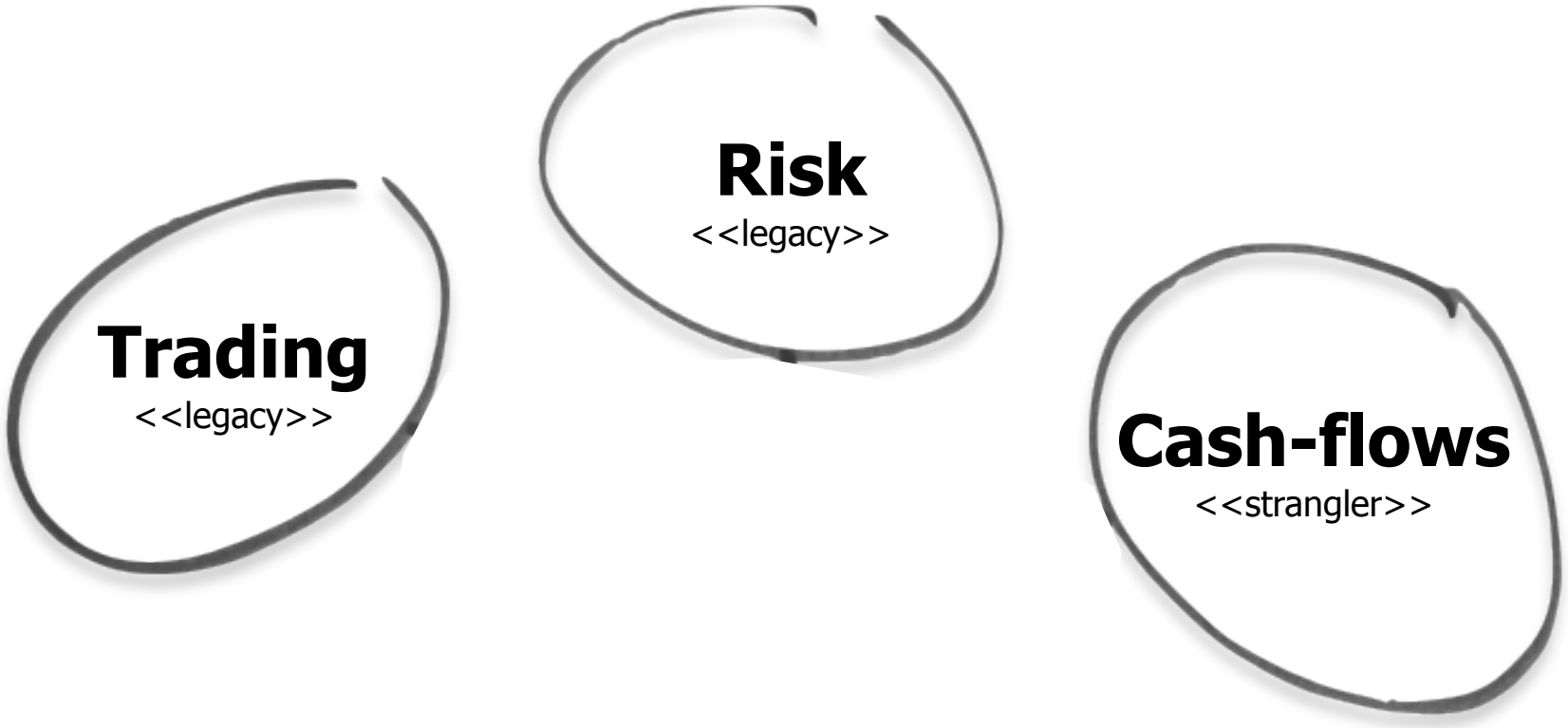
"I don't need all these data to do that, I just need these 3 fields"

Context Game
[Greg Young]

"What's your ideal <instrument>?"

- **Pre-trade:** very **liquid** with a narrow bid – offer spread
- **Trading:** unique **identifier** easy to key in
- **Risk:** has risk on a **single risk factor**
- **Back-office:** very **few cash-flows** that are **certain** and **known** long beforehand

Contexts that emerged



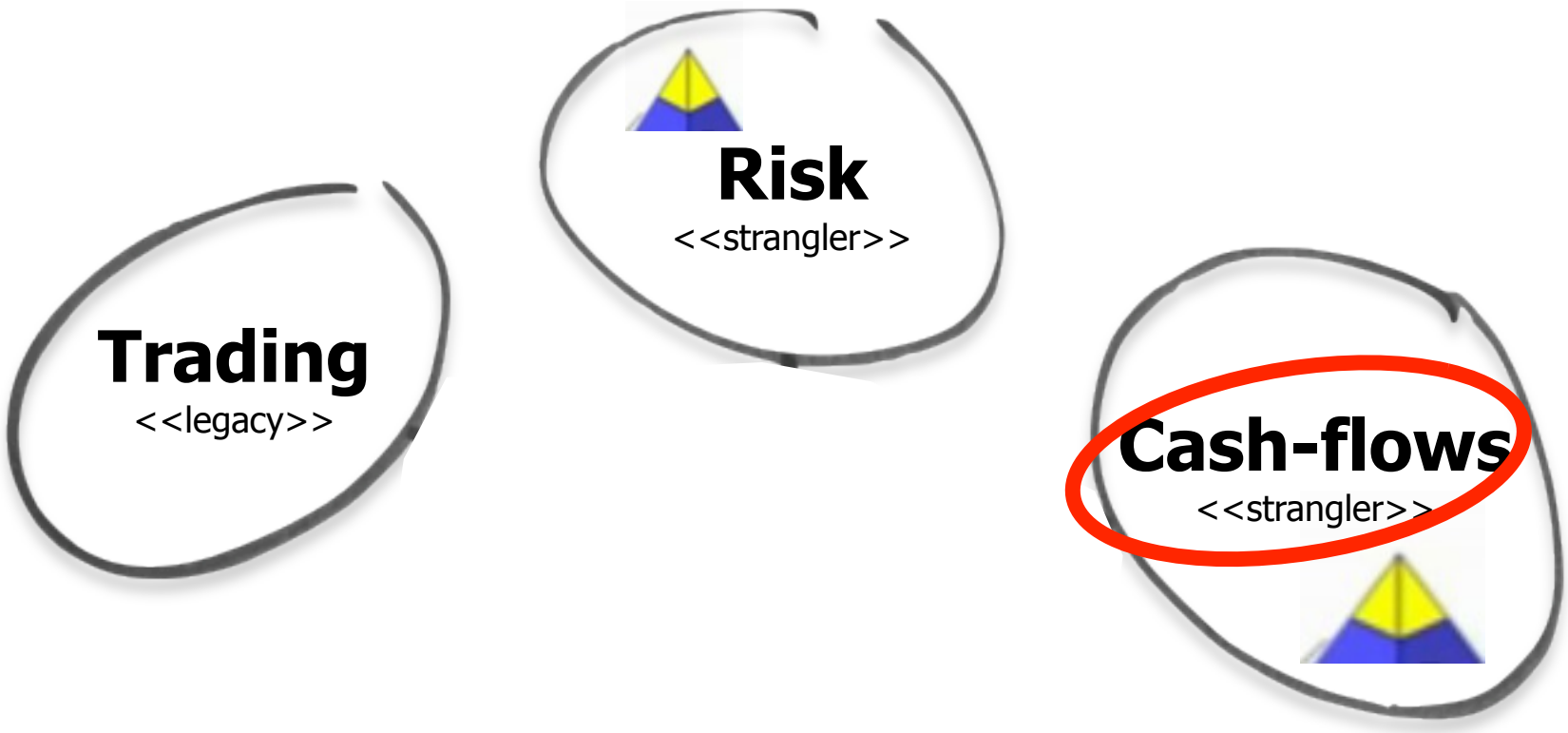
Contexts that emerged

Risk
<<legacy>>

Trading
<<legacy>>

Cash-flows
<<strangler>>

Contexts that emerged



Team /manager invoke DRY

"I don't want to maintain more models, it's already hard enough!"

Team /manager invoke DRY

"I don't want to maintain more models, it's already hard enough!"

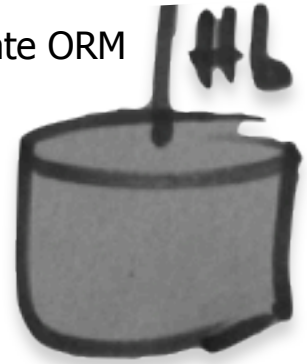
How do we integrate?

Risk
<<strangler>>

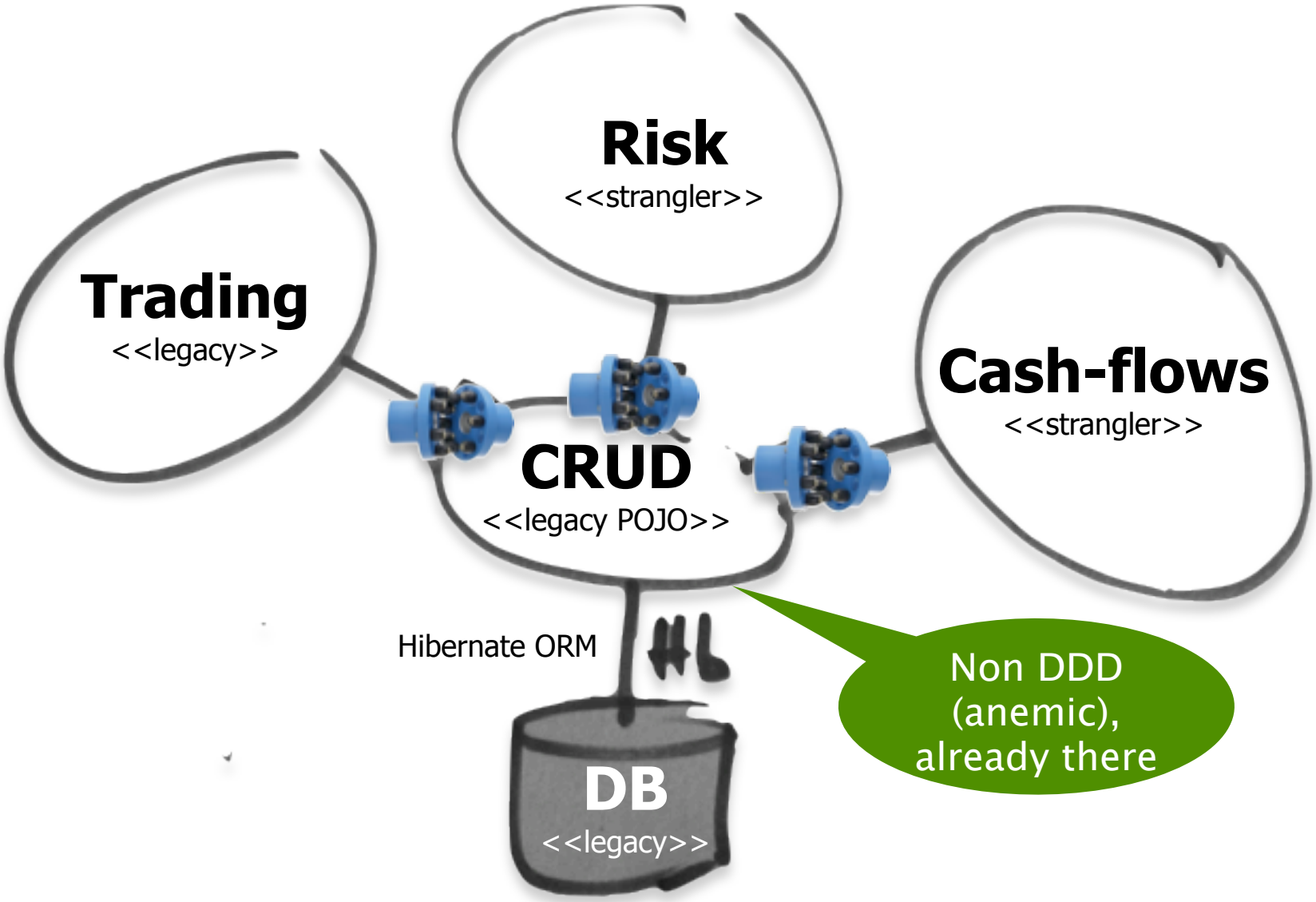
Trading
<<legacy>>

Cash-flows
<<strangler>>

Hibernate ORM



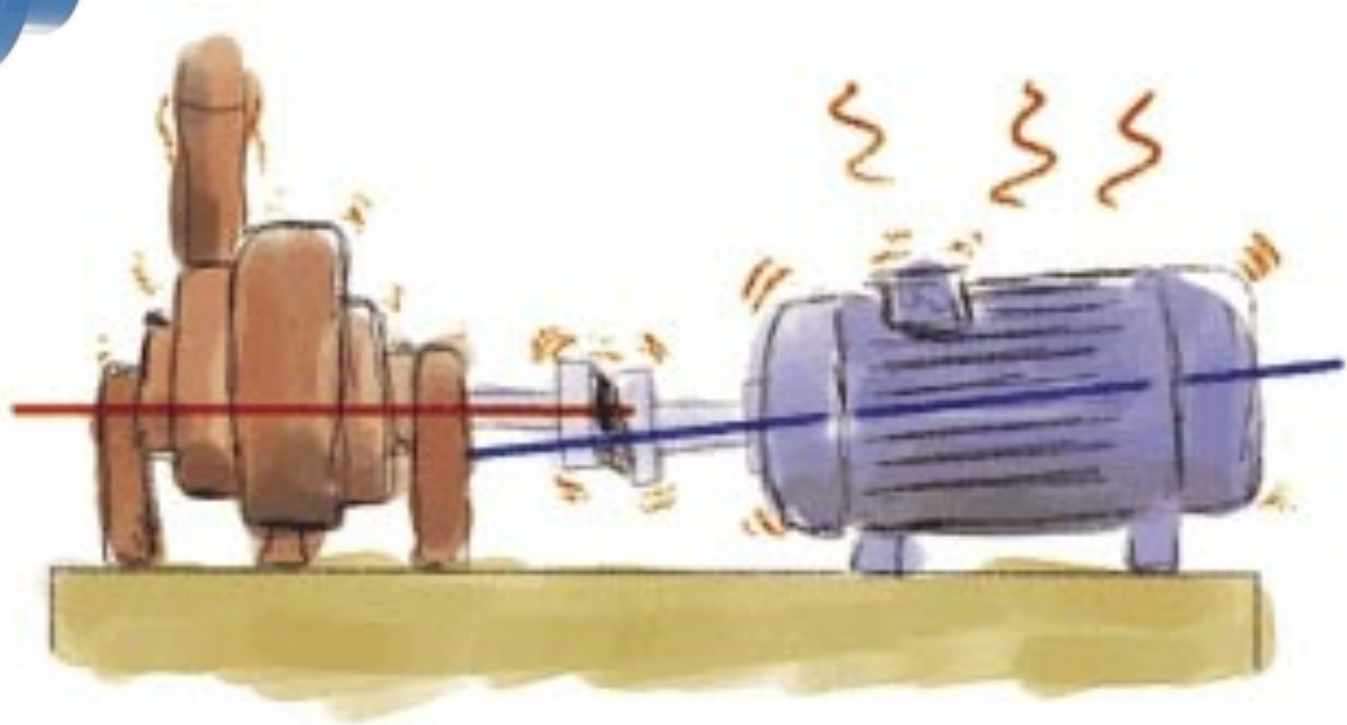
Contexts Mappings



ACL as Shock Absorbers

http://www.directindustry.com/prod/herwarth-reich/flexible-couplings-elastic-couplings-19991-374565.html

ACL as Shock Absorbers



http://www.monition.com/images/shaft-alignment-training.jpg

“That’s useless code!”

No, ACL's are good!

“Ok so I will automate”

No, hand-crafted code is good!

CONCLUSION

Happy End

Project delivered

On required scope

On time

Client satisfaction

Some remarks

- "**DDD + R&D potential**": my projects selection criteria
- **DDD**: subtle, often not quite ***sure***
- Don't preach, **tell the benefits**

The future

- Keep on growing the model in the bubble context
- **Reuse in another context** (e.g. accounting)? - To what extent?
- The **bubble may break** at some point

Questions?

You can also ask me later!


Merci
FUNCTIONAL PROGRAMMING
JAVA / JVM
.NET / CLR
TDD
BDD
Craftsmanship
Red / Green / Refactor

arolla