I help software teams understand

software architecture,
technical leadership and
the balance with agility

I code too

⇧ ; - ⇧ 0
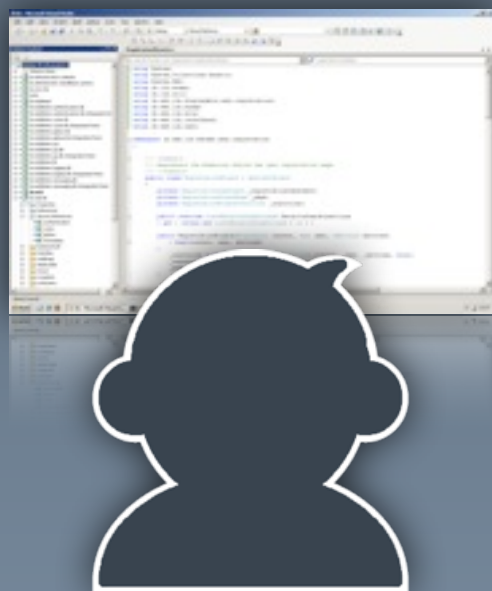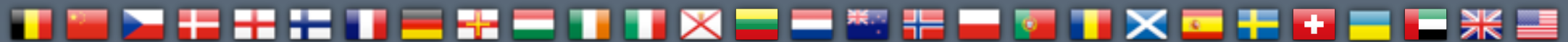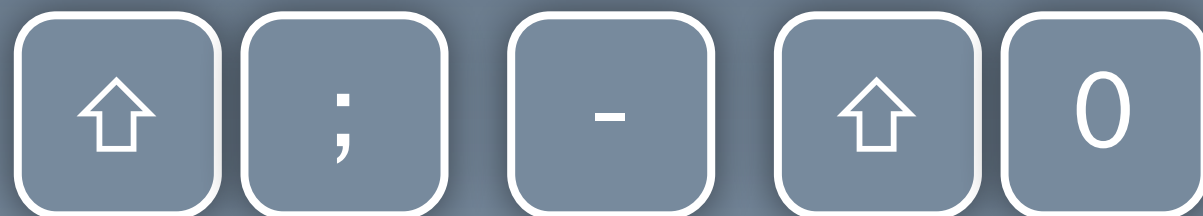
The intersection between

software
architecture
and
code

Software architecture

There's a common misconception that software architecture should be
conceptual
and
exclude technology
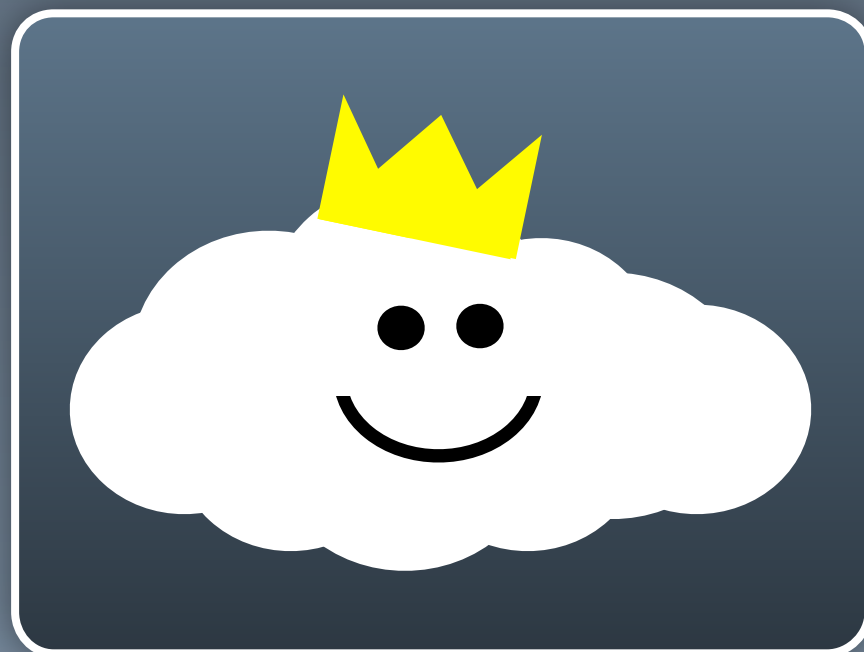
Structure

Vision

How do we
communicate
software architecture?

Who here uses UML on a regular basis?

9 out of 10 people
don't use UML

(in my experience)

NoUML

diagrams?
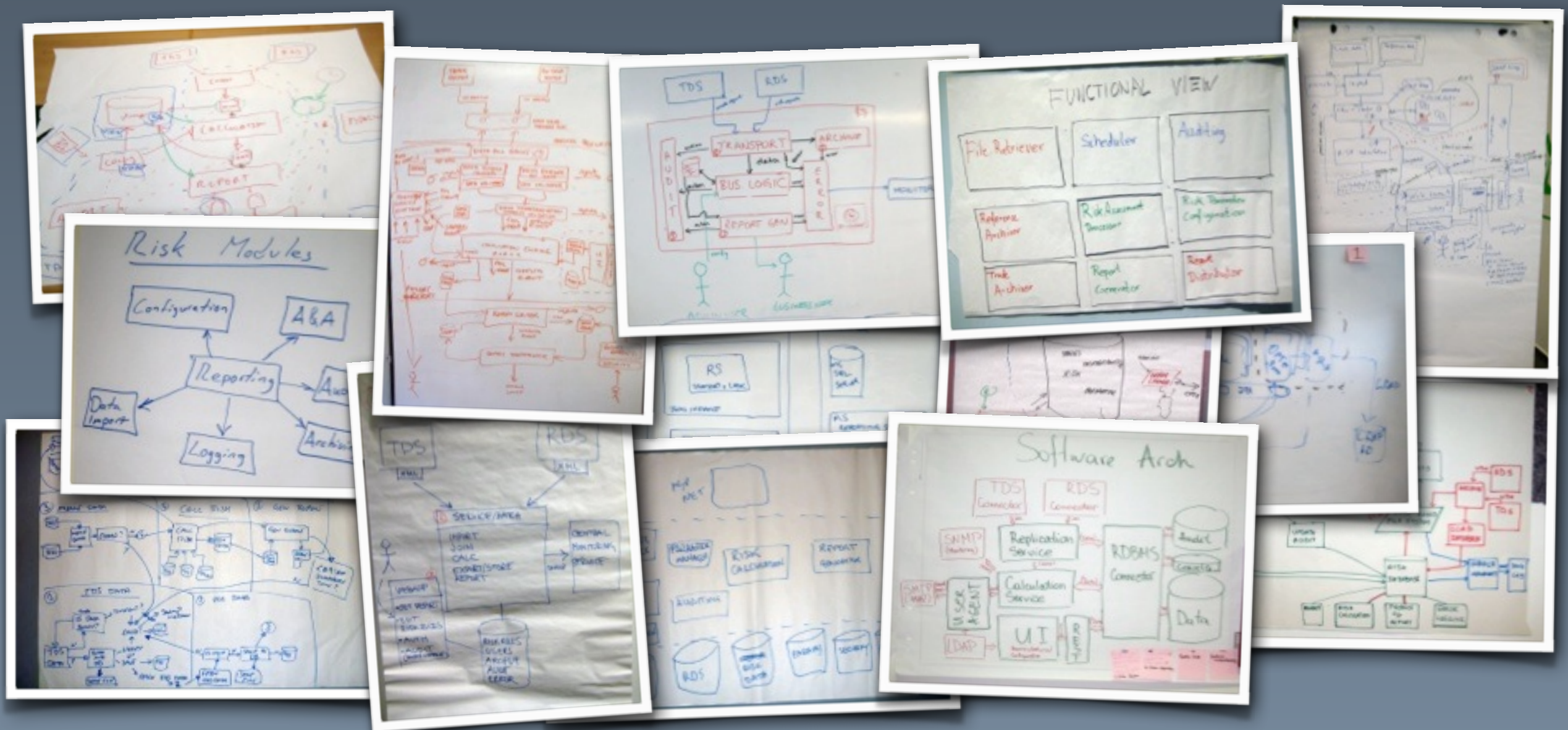
In my experience, software teams aren't able to effectively visualise the software architecture of their systems
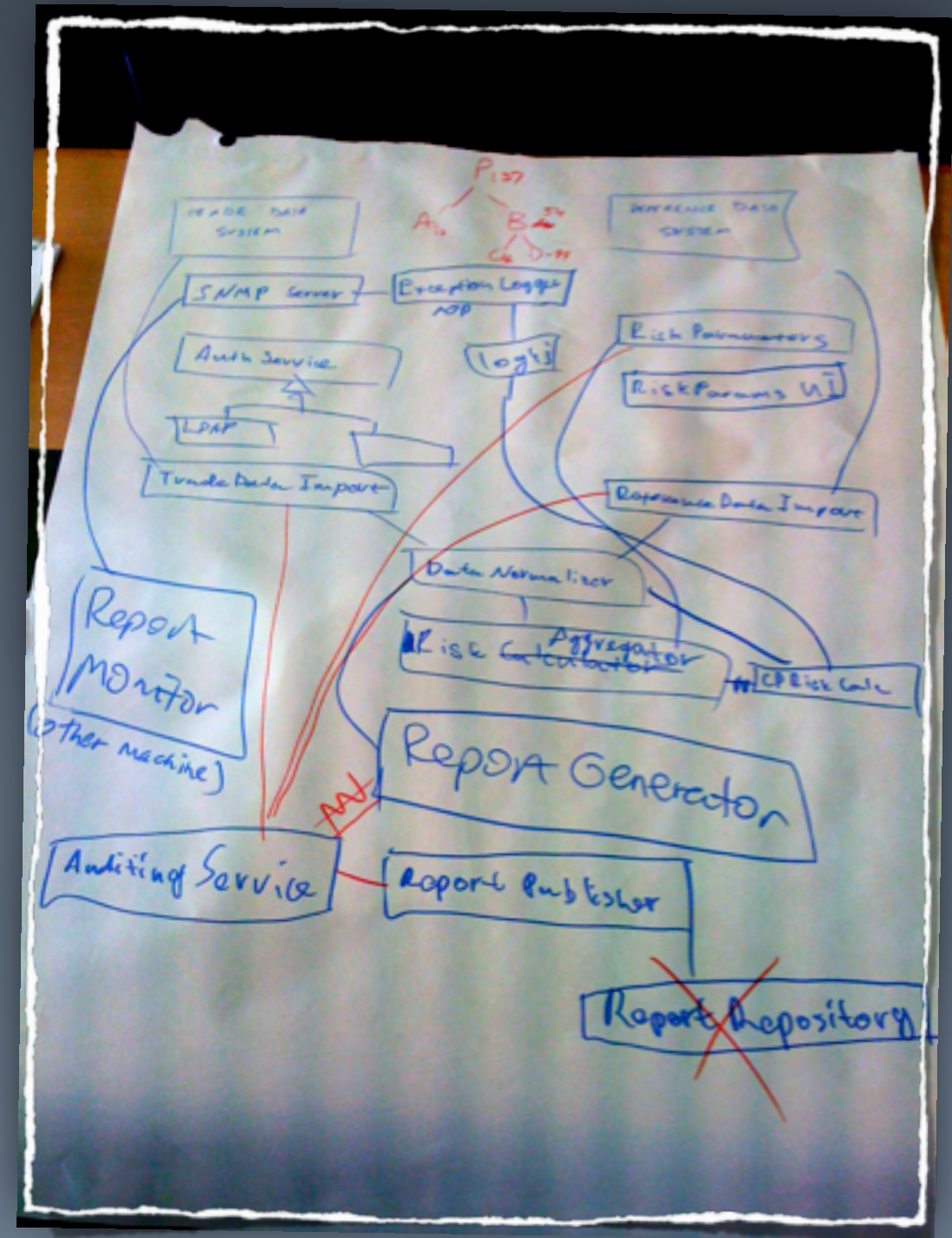
Financial risk system

1. Import data from a Trade Data System (TDS).
2. Import data from a Reference Data System (RDS).
3. Merge the data feeds, perform some risk calculations and generate a Microsoft Excel file of the risk report.
4. Allow a subset of users to modify some parameters used during the calculation process.
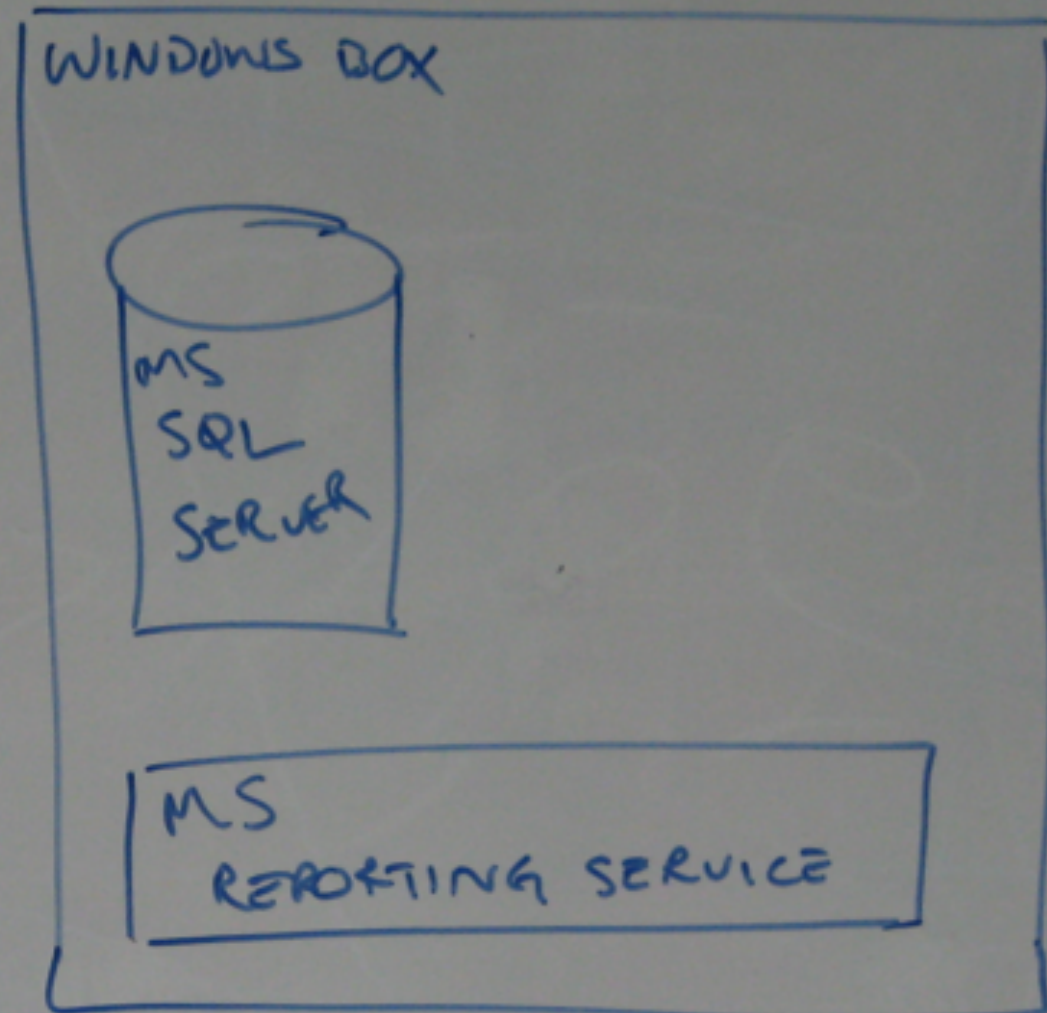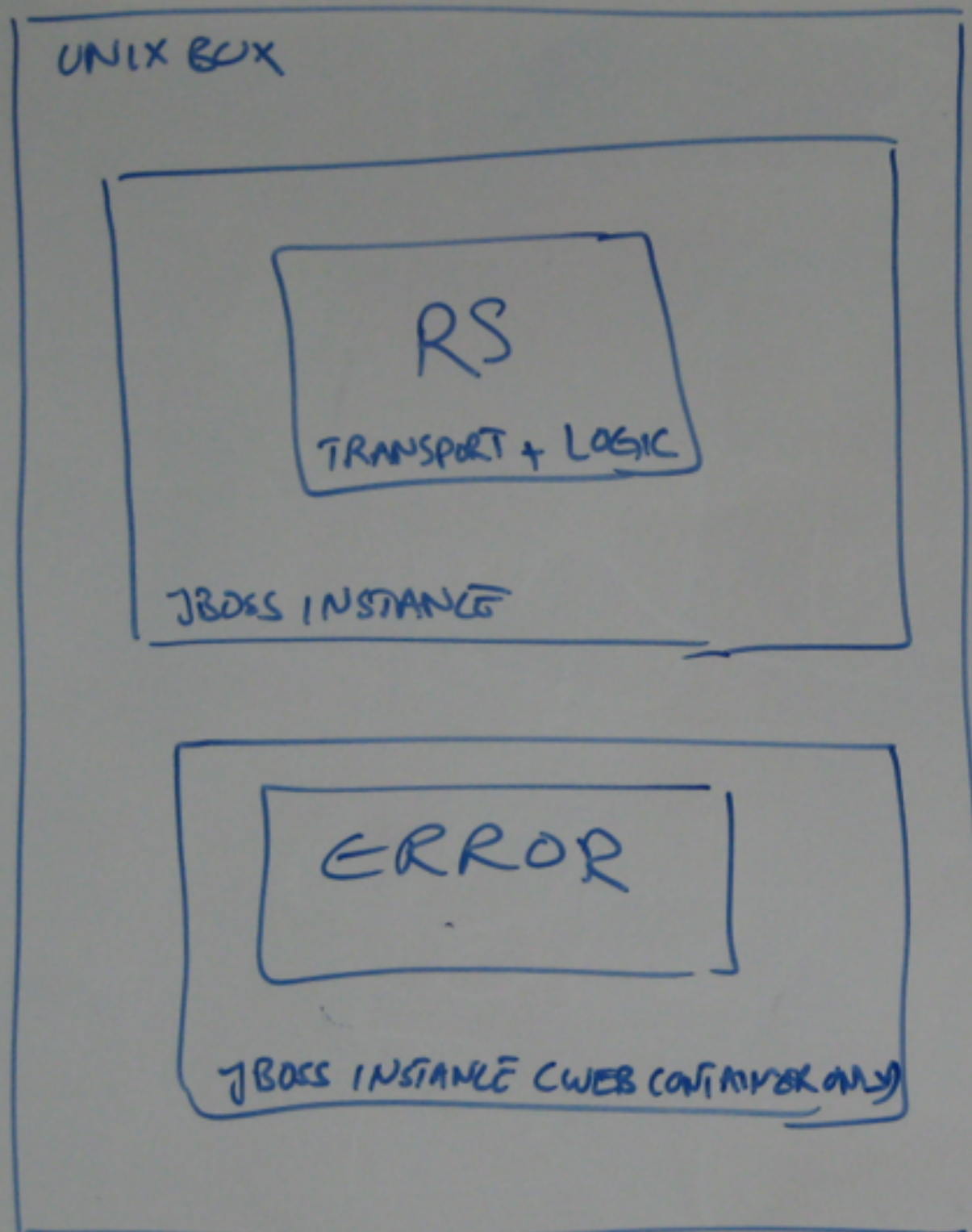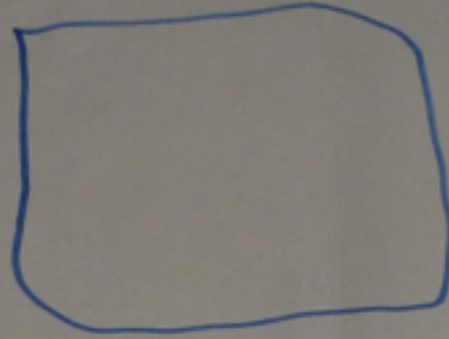
http://bit.ly/sa4d-risksystem

The Shopping List
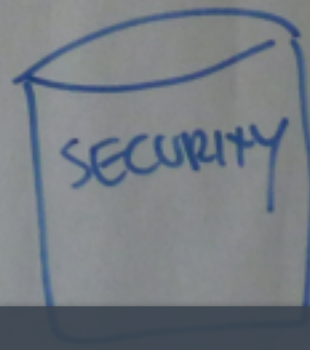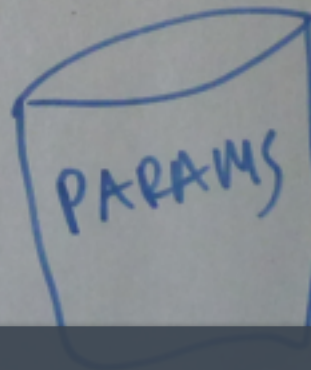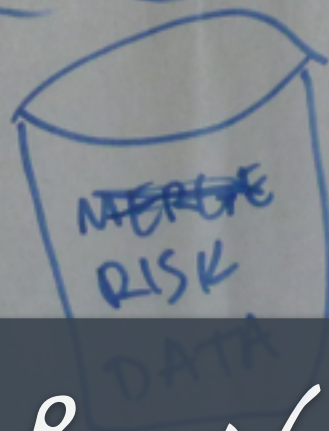
Boxes & No Lines

FUNCTIONAL VIEW

File Retriever	Scheduler	Auditing
Reference Archiver	Risk Assesment Processor	Risk Parameter Configuration
Trade Archiver	Report Generator	Report Distributor

The "functional view

The Airline Route Map
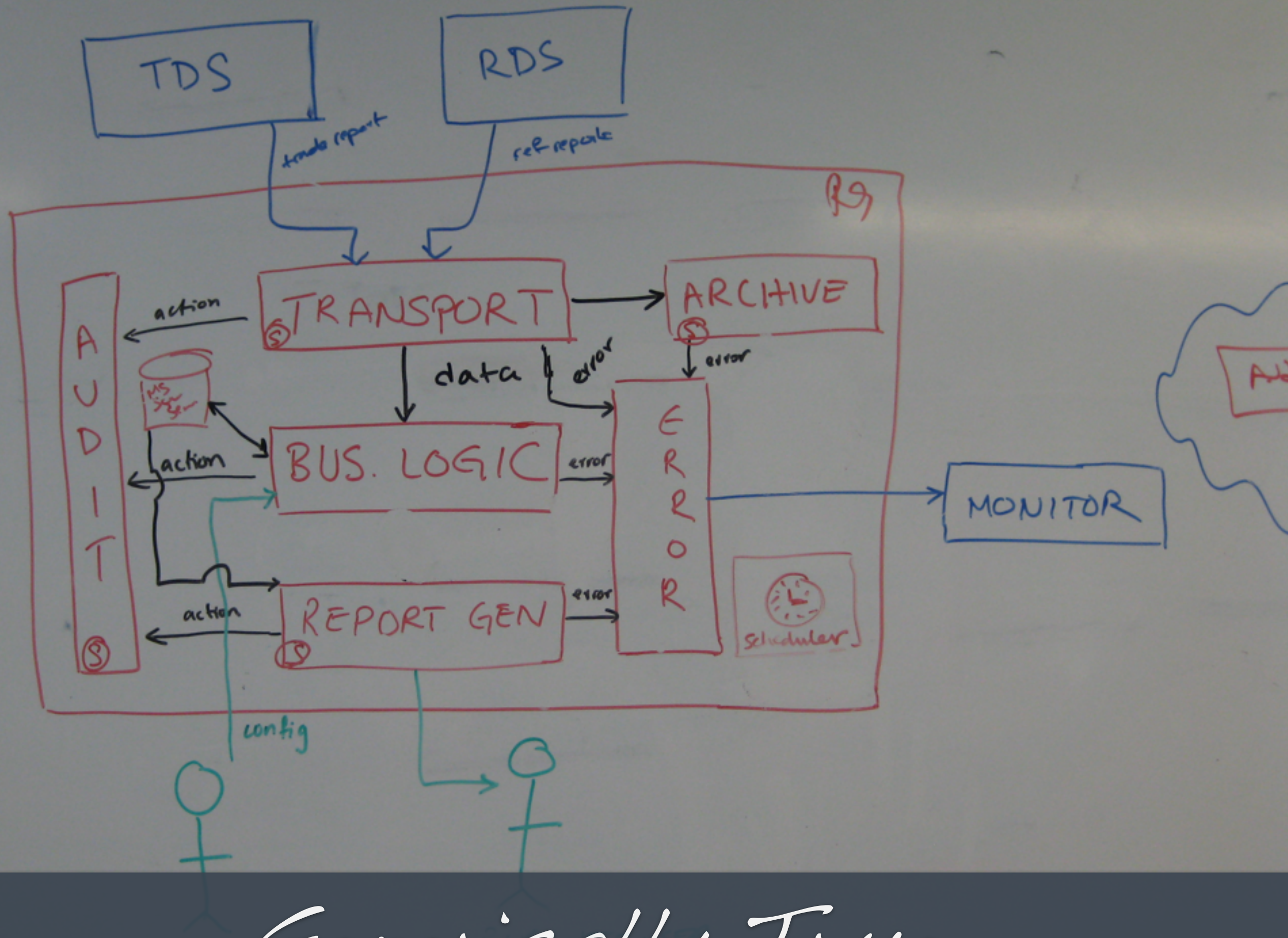
Generically True
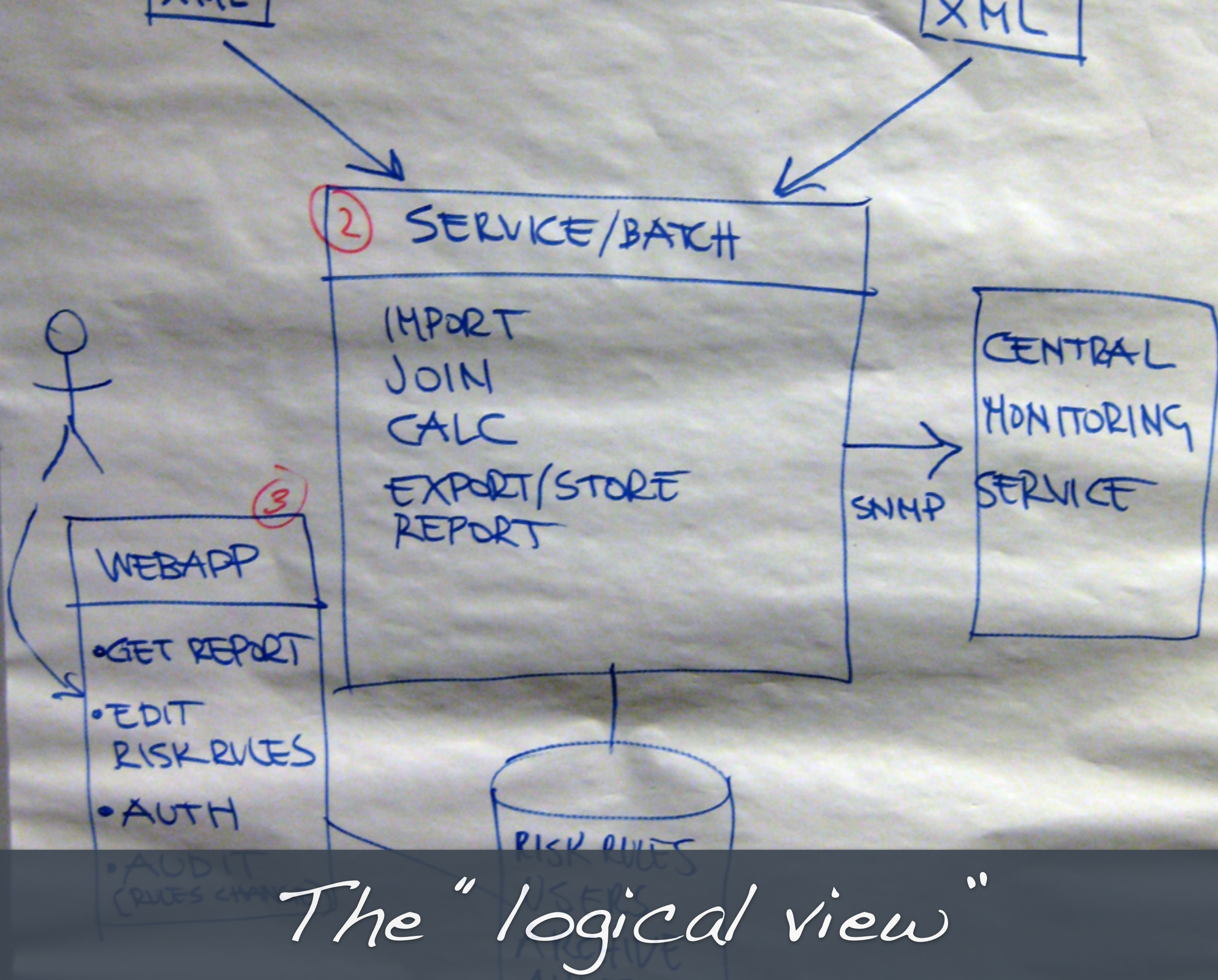
The "logical view"
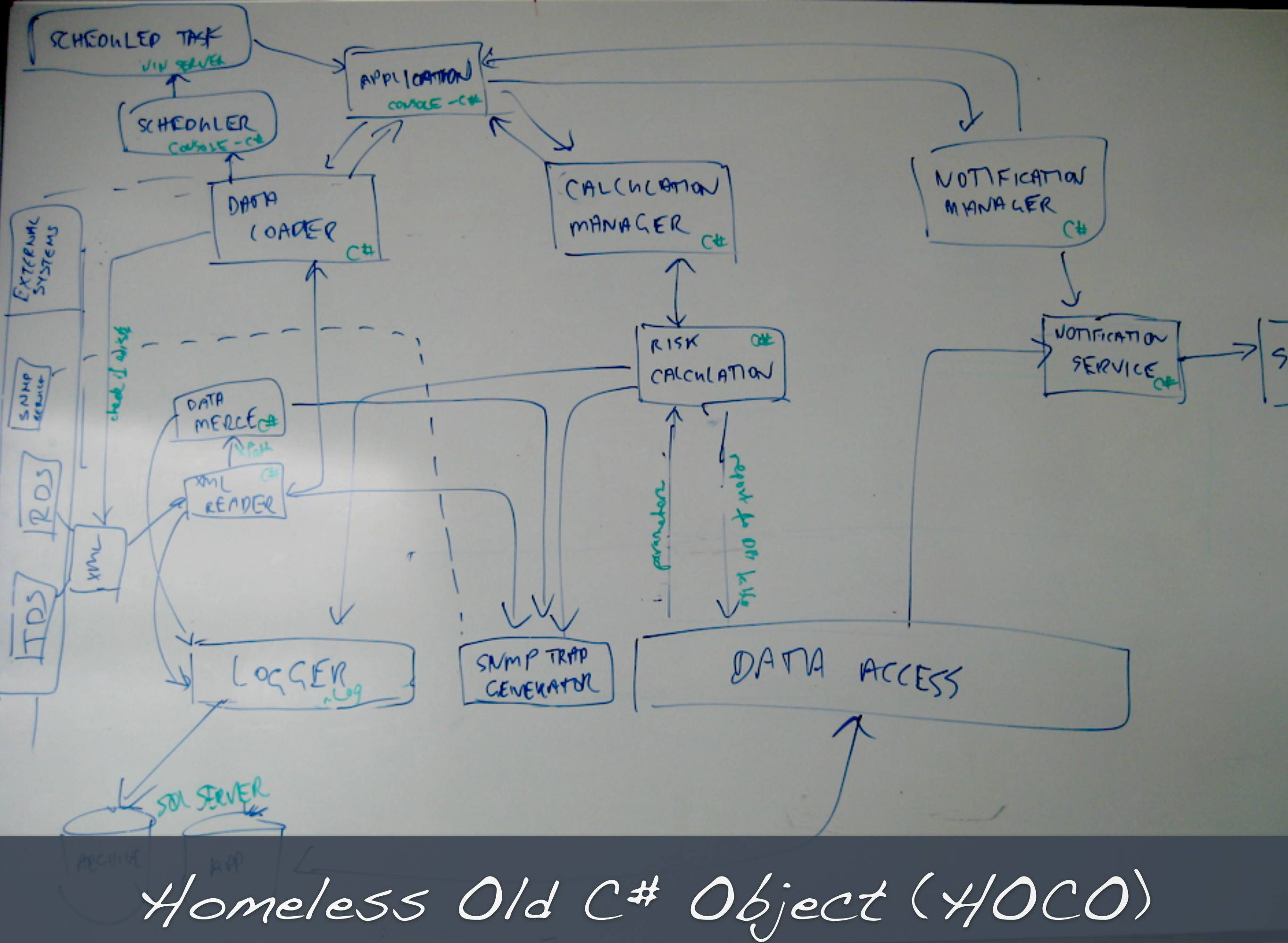
Homeless Old C# Object (HOCO)
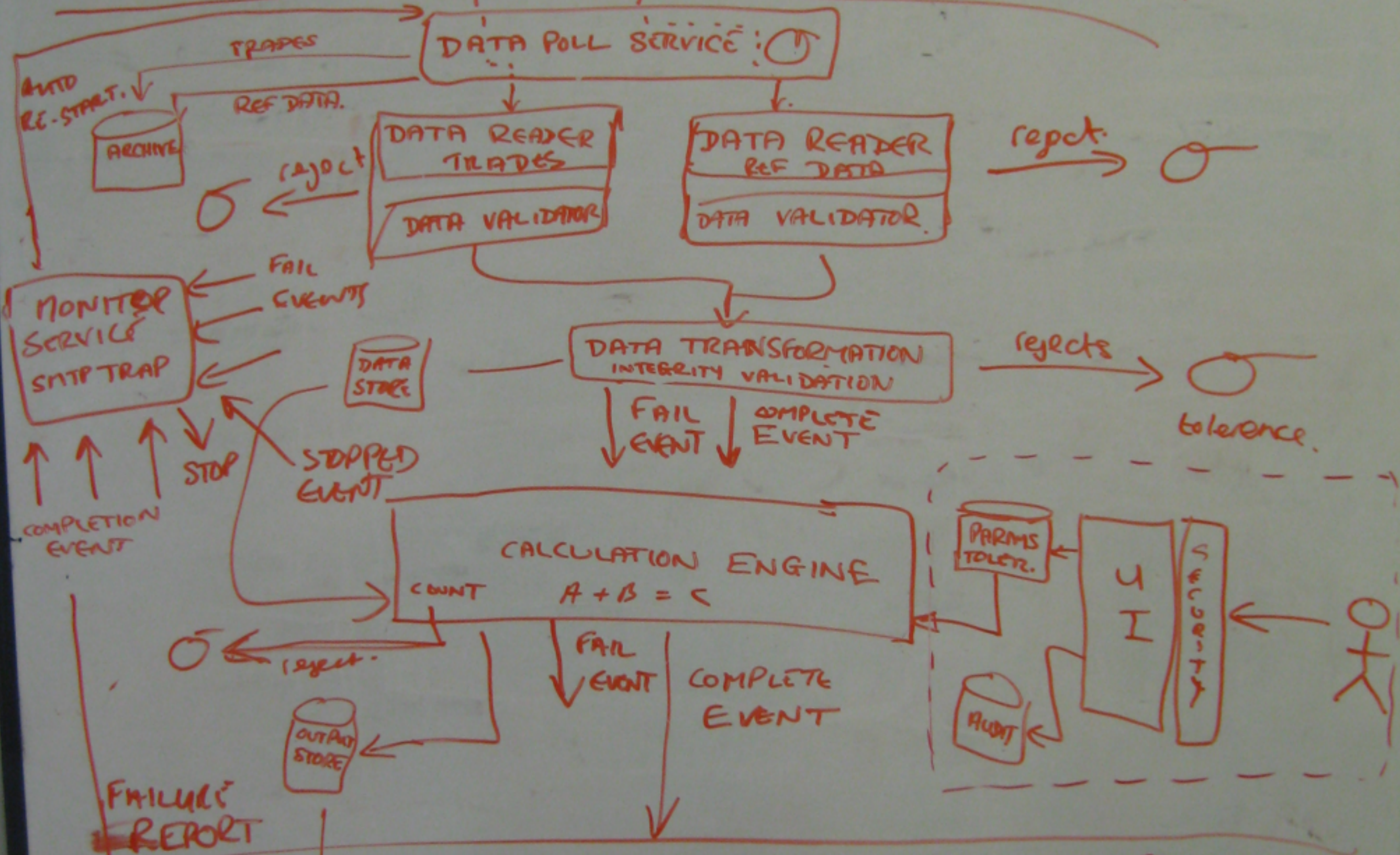
Choose your own adventure

What's been **challenging**
about the exercise so far?

Challenging?

Level of detail
  └ where to stop   different
Who is the audience / backgrounds
Implementation
  — easy to get bogged down
    in detail
Type of diagrams
Notation
Documenting assumptions

⑦ Challenging

Needed to ask questions /
  make assumptions
Temptation to focus on detail
  └ when do we stop?
How much detail?
Talked about more than the diagrams
What notation?   — boxes
                 — arrows

⑩ Challenging?

Verifying our own assumptions
Expressing the solution
  — communicating it in a clear way
  — use of notation
  — easy to mix levels of abstraction
  — how much detail?

What's been challenging about the exercise?

Why has UML fallen out of fashion?

UML

[
is too complex,
is too low-level,
nobody knows it
]

Oh, plus we're

agile

and/or do

TDD

Software architecture provides

boundaries for TDD

It's usually difficult to show the entire design on a **single** diagram

Different **views** of the design can be used to manage complexity and highlight different aspects of the solution
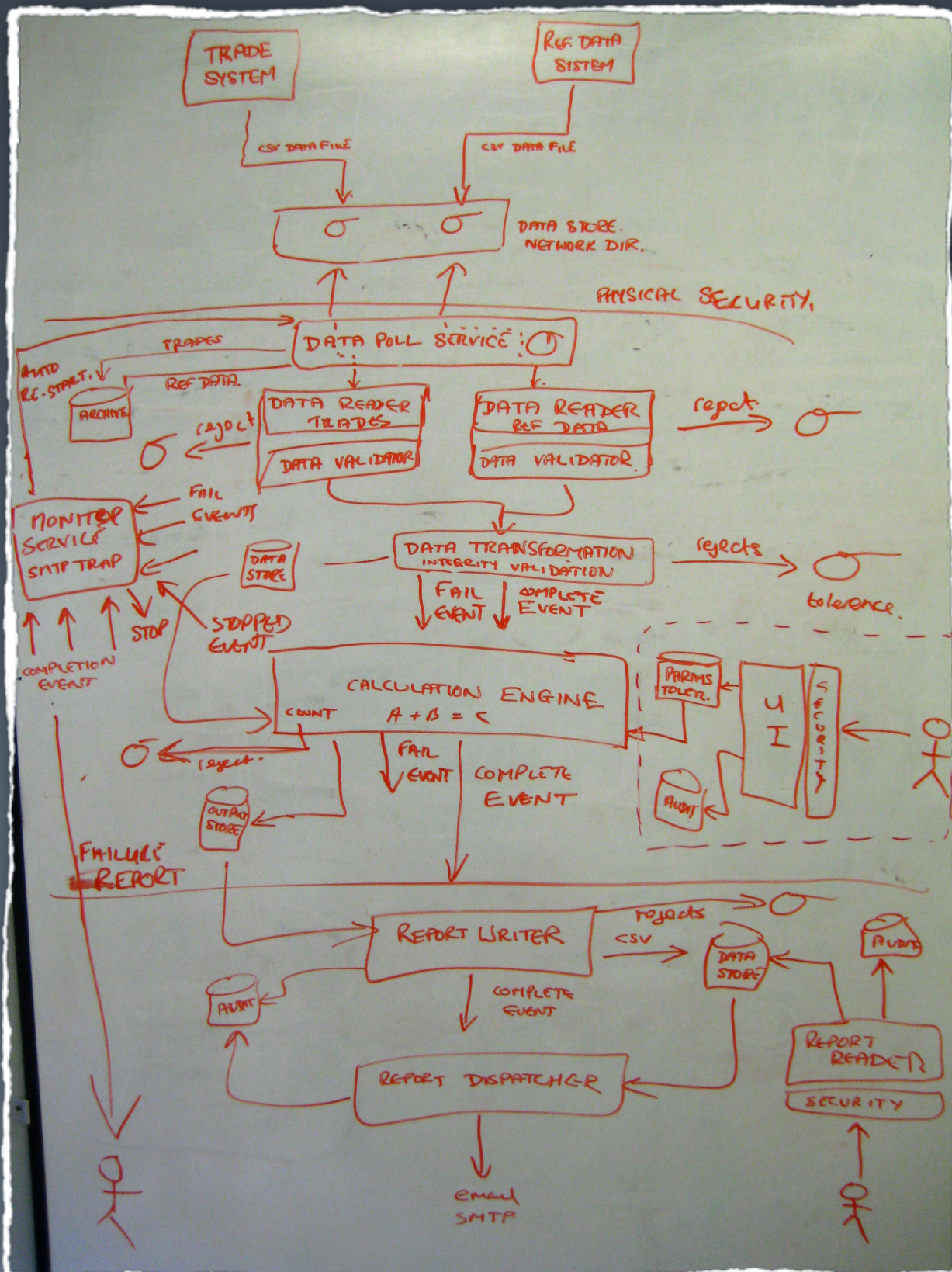
Do the names
of those views make sense?

Conceptual vs Logical

Process vs Functional

Development vs Physical

Development vs Implementation

Physical vs Implementation

Physical vs Deployment

The description of an architecture—the decisions made—can be organized around these four views, and then illustrated by a few selected *use cases*, or *scenarios* which become a fifth view. The architecture is in fact partially evolved from these scenarios as we will see later.

End-user
Functionality

Programmers
Software management

Logical View → Development View

Scenarios

Process View → Physical View

Integrators
Performance
Scalability

System engineers
Topology
Communications

Figure 1 — The "4+1" view model
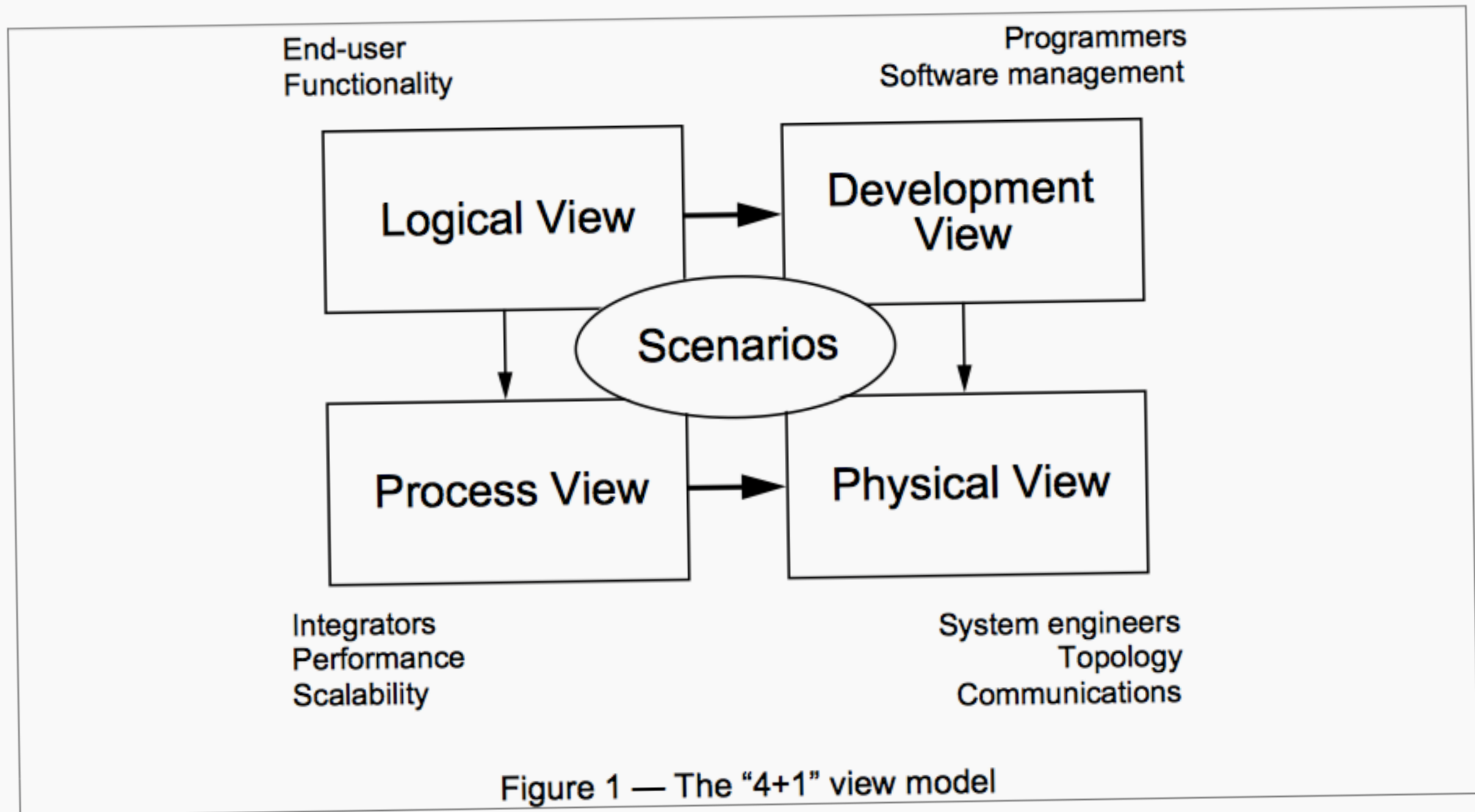
We apply Perry & Wolf's equation independently on each view, i.e., for each view we define the set of elements to use (components, containers, and connectors) , we capture the forms and patterns that work, and we capture the rationale and constraints, connecting the architecture to some of the requirements.

Logical and development views are often separated

BRAIN FREEZE!

software architecture diagram

Video over IP

- Capturing video frames from Camera
- Format conversion and Pre-processing
- H.264 Encoder
- NALU Fragmentation
- Audio/Video sync module
- Format conversion and Post-processing
- H.264 Decoder
- Reconstruction of NALU

RTP/RTCP

Voice over IP

- Acoustic echo cancellation (AEC)
- LEC (G.168/G.165)
- AGC – Gain Control
- VAD
- Speech Encoder

Audio Interface layer — Sound card Interface (PCI / USB)

Telephony Interface layer – Tone detection/generation, CPT/DTMF, MF-R1/R2, and Voice Fax Modem Discrimination Unit

FXS/TDM Interface

V.?

Mod?

Automation

- Administration UI
- Reservation Manager UI
- AutoShell

| Packaging | Process Automation | Configuration Management / Compliance | Help Desk | Reporting (Jasper) | Agent Remote Deployment | Service Response Monitoring |

- CA IT Client Manager
- CA Patch Manager
- CA Process Automation
- CA Configuration Automation
- CA Service Desk
- Chargeback
- Agent Policy Configuration
- Server Monitoring (Local & Remote)

Web Services and Reliable Transport (Active MQ)

- Storage
- Platform Management
- Microsoft HyperV
- Provisioning
- Rapid Server Imaging

Automation Infrastructure Platform

- AOM Data Service
- Policy, Performance, and Resource Management
- Authentication / Security (CA EEM)
- State and Event Management
- Scheduler (Windows)
- Discovery (CA Network Discovery Gateway)
- Management DB
- Performance DB

Service Controller

- Signing
- Authentication
- Authorization
- Rich UI
- Web UI
- Controls
- Service Interface
- Activity
- Business Rules
- Human Workflow
- Workflow
- Service Agents
- DAL
- E-Publish
- EAI
- ECM
- DW
- OLTP
- Monitoring
- Log & Trace
- Exception Management
- Configuration

---

e-AIRS / Scientific Application Service Provider

- Presentation Sublayer
- Portal
- Project Management
- Project Editing Service
- Pr? ?ing Service
- User Profile Management Service
- System Pro? Management S?

Computational Analysis Service

- e-AIRSmesh
- e-AIRSview
- Mesh Converter
- CFD Solver
- Output Converter

Remote Experiment Service

- Exp. Exec.
- Exp. Info.
- Aerodynamic Exp.
- PIV Exp.

Parametric Study

- Validation
- Analysis

Collaboration

- Session Reservation
- Session Management

- Resource Configuration Service
- Resource Monitoring Service
- Authentication

TCP / HTTP

ASP (Scientific Application Service Provider)

Basic Execution Service

- Parameter Sweeper
- Global Scheduler
- Resource Catalog
- GRAM adapter

Plotting Service

- eAIRS Plot
- Conv. Graph

Data Transfer Service

Metadata Management Service

AG + VNC

- Visualization Sharing
- Mobile Service Component
- A/V Conference
- Chat

Application

SCADA

- ASCII File Editor
- ASCII Files
- Commercial DB
- Commercial Develop. tools
- Library
- SCADA Develpt. Environ
- Export
- Import
- Project Editor
- Driver Toolkit
- HMI
- Trending
- Alarm Display
- Log Display
- Active X Controls
- 3rd Party Applic.

Active X Container

Client — Server — Publish / Subscribe — TCP/IP

SCADA Server

- Recipe DB
- Recipe Managt.
- Ref. DB
- Data Processing
- RT & Event Manager
- Report Gener.
- Alarm
- Log
- Archive
- Data R/W
- RT DB
- SQL
- Alarm DB
- Log DB
- Archive DB
- ODBC
- DDE
- API DLL
- Driver
- OPC
- Private Application
- EXCEL

VME PLC PLC

Typical generic software architecture of SCADA systems

- Remote administration interface
- User manager
- Output device manager
- Output devices
- DB-Server
- Analysis

I bought a boat

Code

Would we

code

it that way?

The code is the embodiment of the architecture

Abstraction

is about reducing detail

rather than creating a different representation

Abstractions help us

reason about

a big and/or complex software system

Does your code reflect the **abstractions** that you think about?

We often **think**

in components

but write classes

(usually in layers)

Controller A Presentation layer Controller B

Service A Business layer Service B

Repository A Data layer Repository B
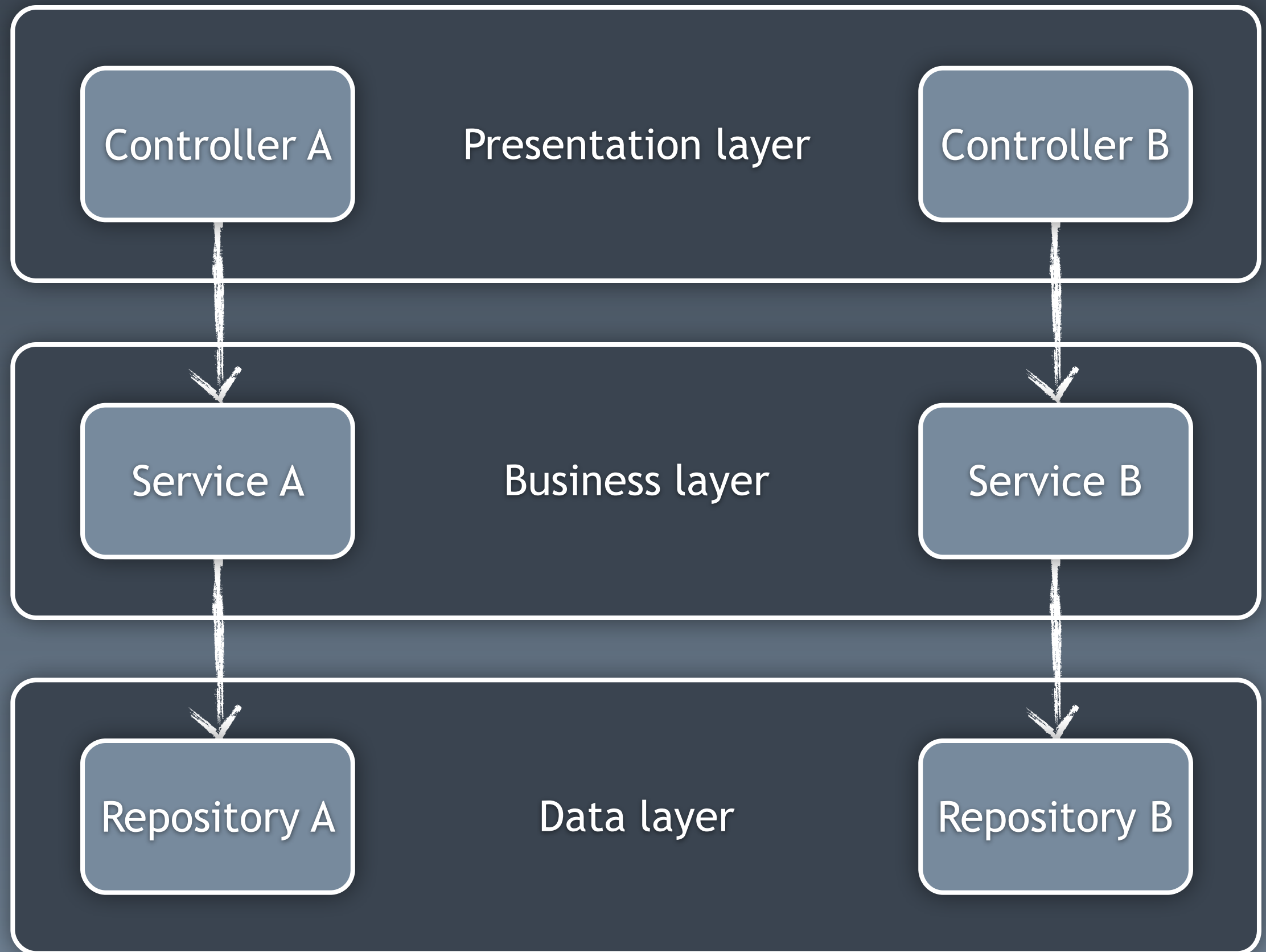
A layered architecture

Package Explorer ✖   ⫶ JUnit

- ▼ springapp
  - ▼ src
    - ▼ springapp.domain
      - ▶ Product.java
    - ▼ springapp.repository
      - ▶ JdbcProductDao.java
      - ▶ ProductDao.java
    - ▼ springapp.service
      - ▶ PriceIncrease.java
      - ▶ PriceIncreaseValidator.java
      - ▶ ProductManager.java
      - ▶ SimpleProductManager.java
    - ▶ springapp.web
  - ▶ test
  - ▶ JRE System Library [JVM 1.5.0 (MacOS X Default)]
  - ▶ Referenced Libraries
  - ▶ db
  - ▼ war
    - ▼ WEB-INF
      - ▼ classes
        - ▶ springapp
        - jdbc.properties
        - messages.properties
      - ▼ jsp
        - hello.jsp
        - include.jsp

Presentation tier

JSPs or other views
Generate HTML

Remote service
"exporters":
*Web services or
other protocols*

Web tier actions
*Process user input, call
service layer, choose
view to display*

Business services layer: *Exposes key functionality.
Manages transaction boundaries, includes business
logic. No knowledge of persistence specifics.
Declarative services typically used here.*

DAO interface layer
*Defines persistence operations, independent of
implementing technology*

DAO interface implementation layer
Retrieves, saves entities using ORM tool, JDBC

Persistent
domain object
(entity)
Core OO model

O/R mapping layer

J
D
B
C

RDBMS

EIS tier Databases,
other transactional
resources

Let's summarize each layer and its responsibilities, beginning closest to the database or other enterprise resources:

- ❑ **Presentation layer:** This is most likely to be a web tier. This layer should be as thin as possible. It should be possible to have alternative presentation layers—such as a web tier or remote web services facade—on a single, well-designed middle tier.

- ❑ **Business services layer:** This is responsible for transactional boundaries and providing an entry point for operations on the system as a whole. This layer should have no knowledge of presentation concerns, and should be reusable.

- ❑ **DAO interface layer:** This is a layer of interfaces *independent of any data access technology* that is used to find and persist persistent objects. This layer effectively consists of *Strategy* interfaces for the Business services layer. This layer should not contain business logic. Implementations of these interfaces will normally use an O/R mapping technology or Spring's JDBC abstraction.

- ❑ **Persistent domain objects:** These model real objects or concepts such as a bank account.

- ❑ **Databases and legacy systems:** By far the most common case is a single RDBMS. However, there may be multiple databases, or a mix of databases and other transactional or non-transactional legacy systems or other enterprise resources. The same fundamental architecture is applicable in either case. This is often referred to as the *EIS (Enterprise Information System)* tier.

In a J2EE application, all layers except the EIS tier will run in the application server or web container. Domain objects will typically be passed up to the presentation layer, which will display data they contain, *but not modify them*, which will occur only within the transactional boundaries defined by the business services layer. Thus there is no need for distinct Transfer Objects, as used in traditional J2EE architecture.

In the following sections we'll discuss each of these layers in turn, beginning closest to the database.

> Spring aims to decouple architectural layers, so that each layer can be modified as

Spring aims to decouple architectural layers, so that each layer can be modified as far as possible without impacting other layers. No layer is aware of the concerns of the layer above; as far as possible, dependency is purely on the layer immediately below. Dependency between layers is normally in the form of interfaces, ensuring that coupling is as loose as possible.

1 Answer

active   oldest   **votes**

▲
8
▼
✓

If you are using maven, it's best to follow the standard maven project layout. You can get maven to generate this structure for you by doing,

```
mvn archetype:generate
```

and select spring-mvc-jpa-archetype from the list of choices

This will give you a package structure like,

```
├── pom.xml
└── src
    └── main
        ├── java
        │   └── mygroup
        │       ├── controller
        │       │   ├── HomeController.java
        │       │   └── PersonController.java
        │       ├── dao
        │       │   └── PersonDao.java
        │       └── model
        │           └── Person.java
        ├── resources
        │   ├── db.properties
        │   ├── log4j.xml
        │   └── META-INF
        │       └── persistence.xml
        └── webapp
            ├── index.html
            ├── META-INF
            │   ├── context.xml
            │   └── MANIFEST.MF
            ├── resources
            │   └── css
            │       └── screen.css
            └── WEB-INF
                ├── spring
                │   ├── app
                │   │   ├── controllers.xml
                │   │   └── servlet-context.xml
                │   ├── db.xml
                │   └── root-context.xml
                ├── views
```

Related

2   How do I use a custom authentication mechanism for a Java web application with Spring Security?

0   Is there a sample web project of integrating Spring-MVC and Spring-Data-JPA?

1   Reusing DAOs in another web application

0   Java Web Application Warming

3   Is a parallel Spring-MVC application possible with a non-spring web app?

2   Adding a web interface (Spring MVC) to existing Java application

0   package strucutre for Spring MVC project with multiple sub projects using maven

0   How to add a setup

Quick Launch (Ctrl+Q)

FILE   EDIT   VIEW   PROJECT   BUILD   DEBUG   TEAM   TOOLS   TEST   WINDOW   HELP

▶ Internet Explorer   Debug

HomeController.cs

MvcMovie.Controllers.HomeCont   ▾   Index()

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MvcMovie.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            ViewBag.Message = "Modify this templ

            return View();
        }


        public ActionResult About()
        {
            ViewBag.Message = "Your app descript

            return View();
        }
```
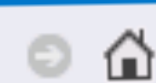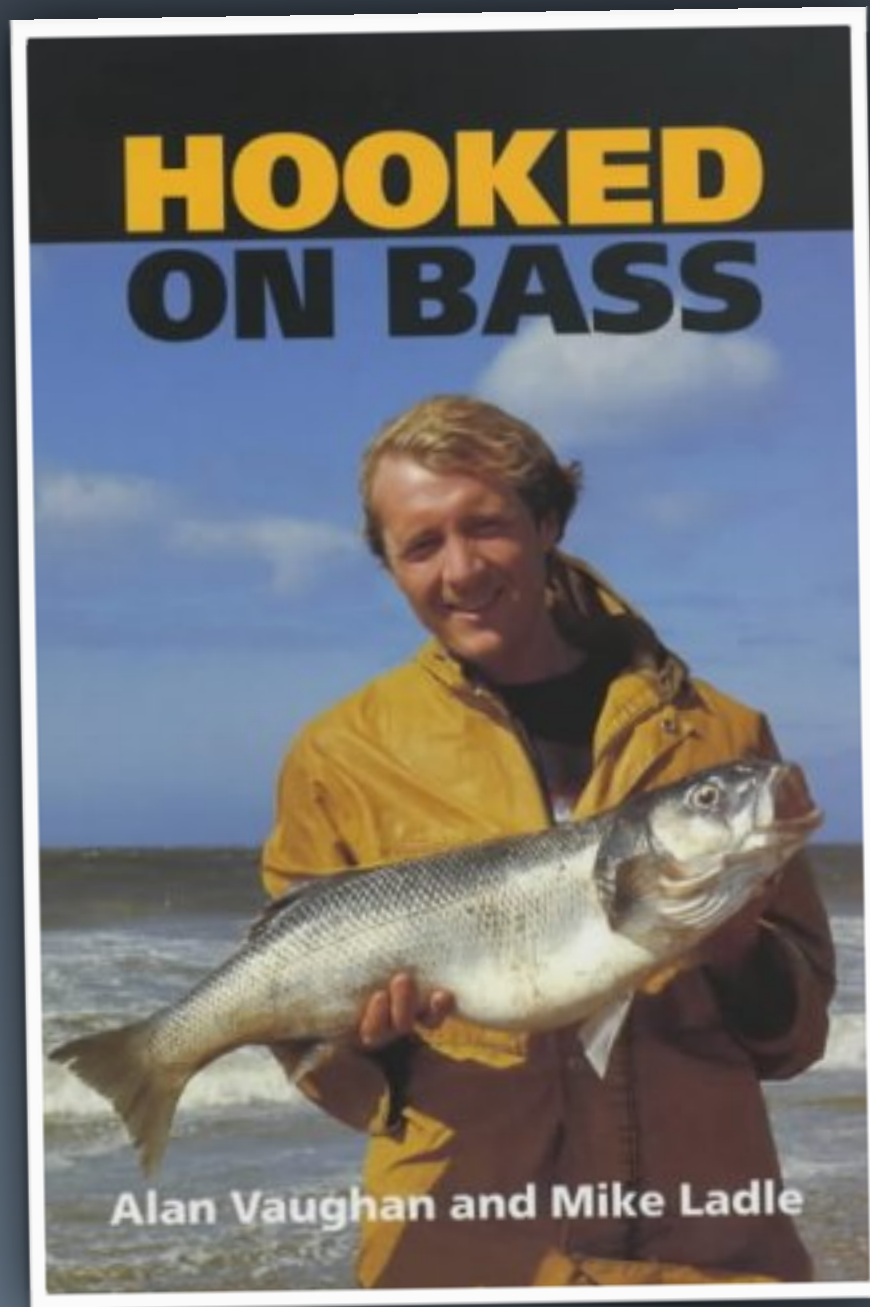
Solution Explorer

Search Solution Explorer (Ctrl+;)
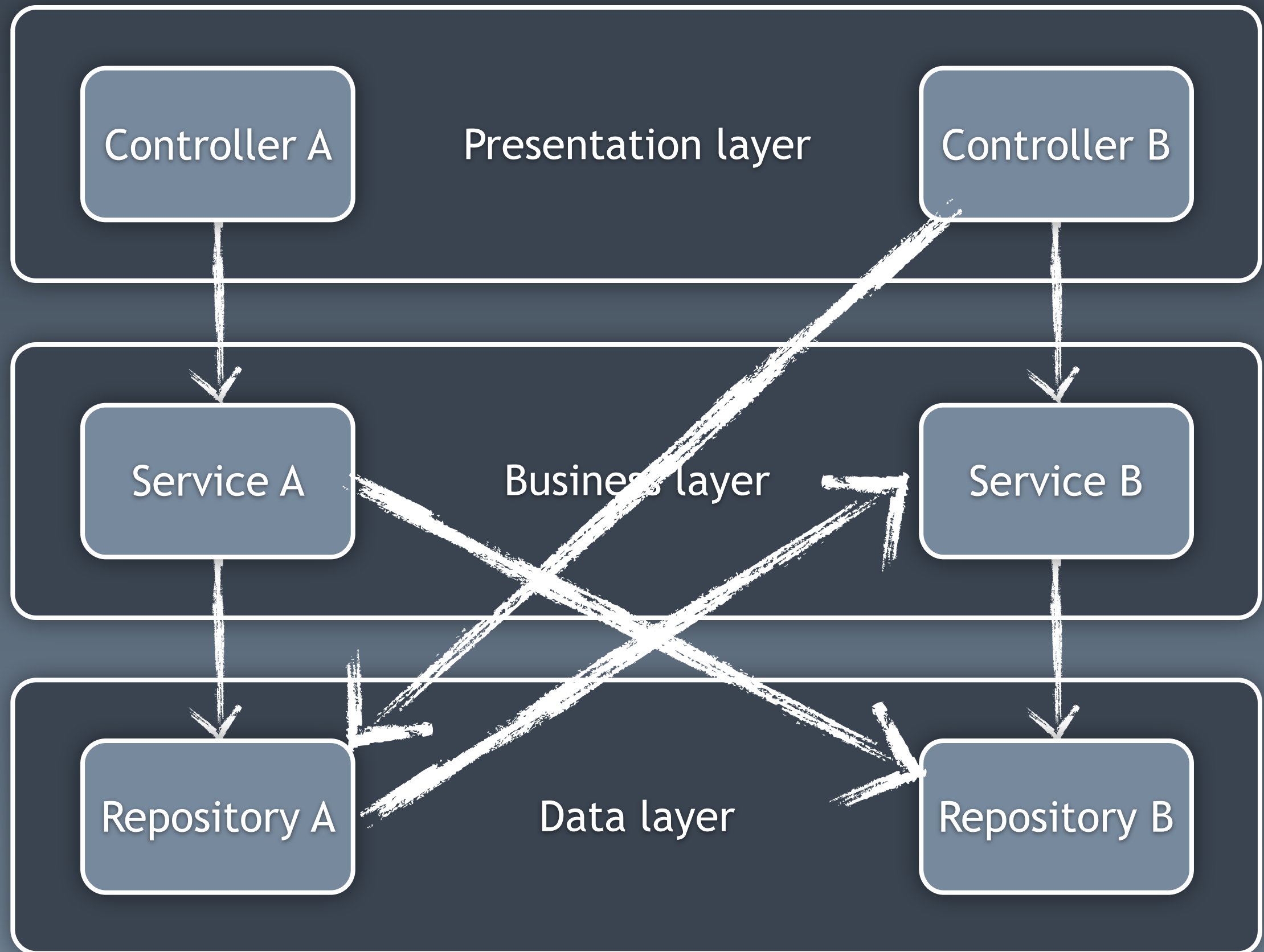
Solution 'MvcMovie' (1 project)
  ▲   MvcMovie
      ▷   Properties
      ▷   References
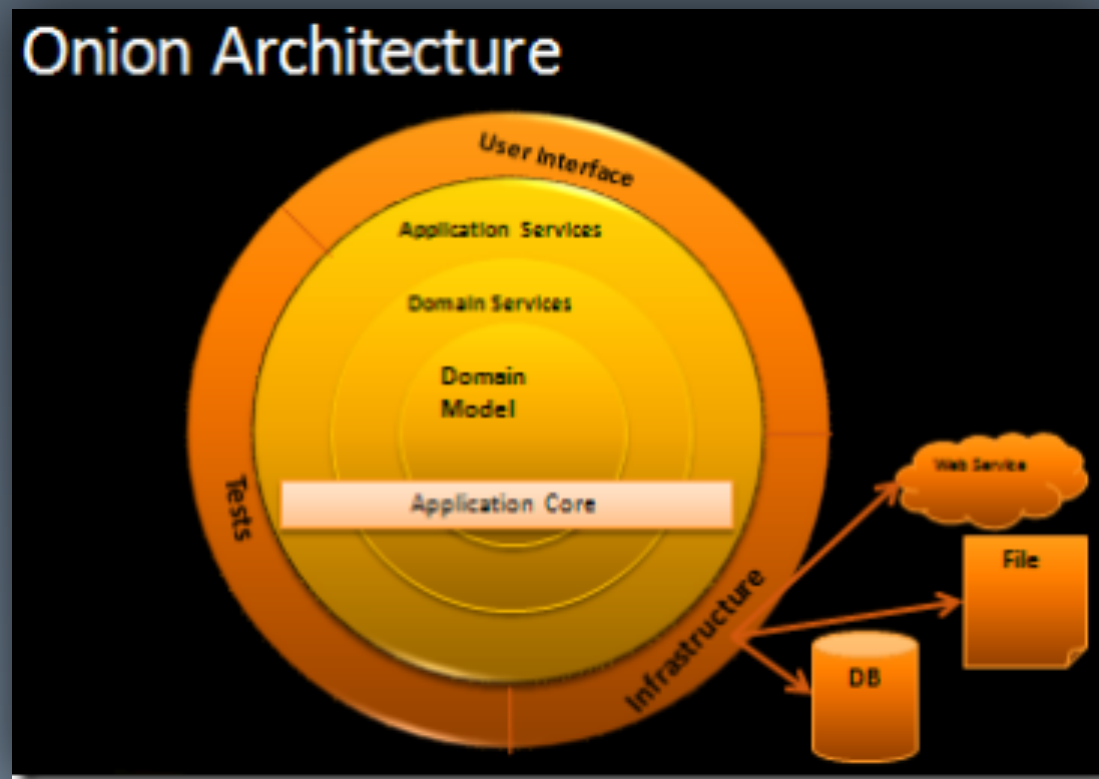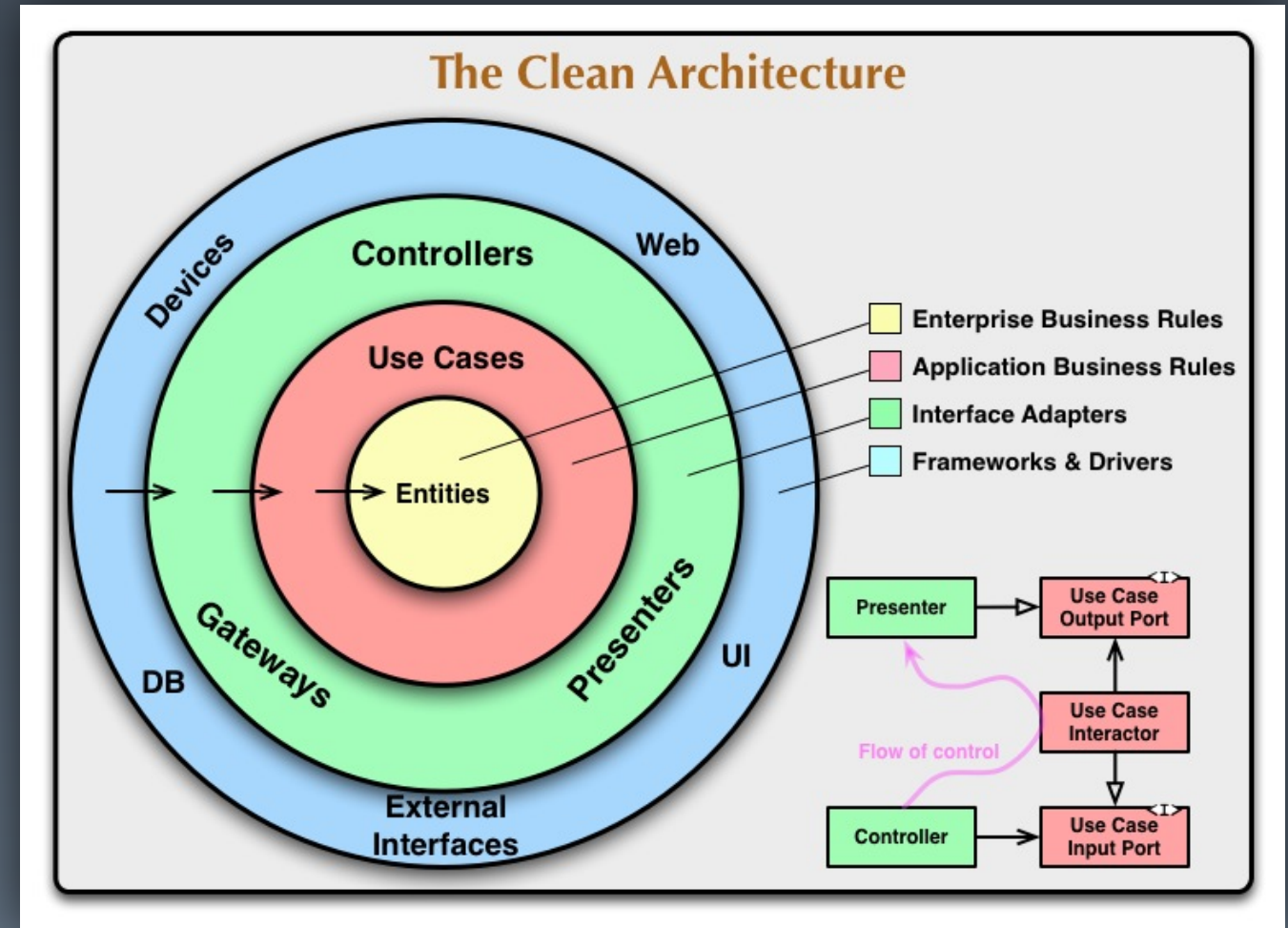   App_Data
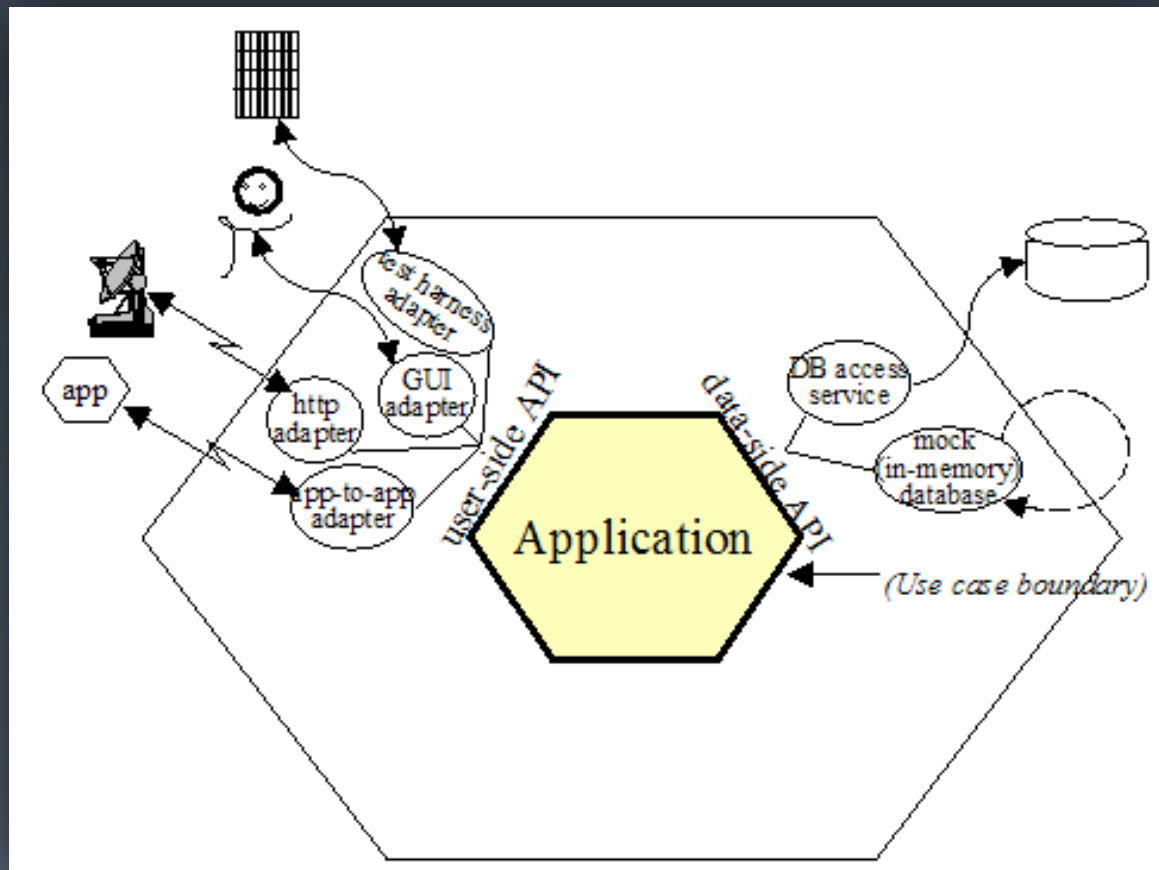   ▷   App_Start
      ▷   Content
      ▷   Controllers
      ▷   Filters
      ▷   Images
      ▷   Models
      ▷   Scripts
      ▷   Views
          favicon.ico
      ▷   Global.asax
          packages.config
      ▷   Web.config

100 %

Ready   Ln 1   Col 1   Ch 1   INS

Find the fish and
catch the fish you will

A typical layered architecture :-)

The Clean Architecture

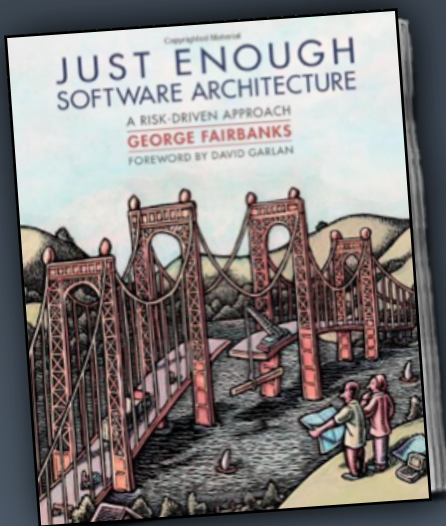Onion Architecture

Hexagons and onions
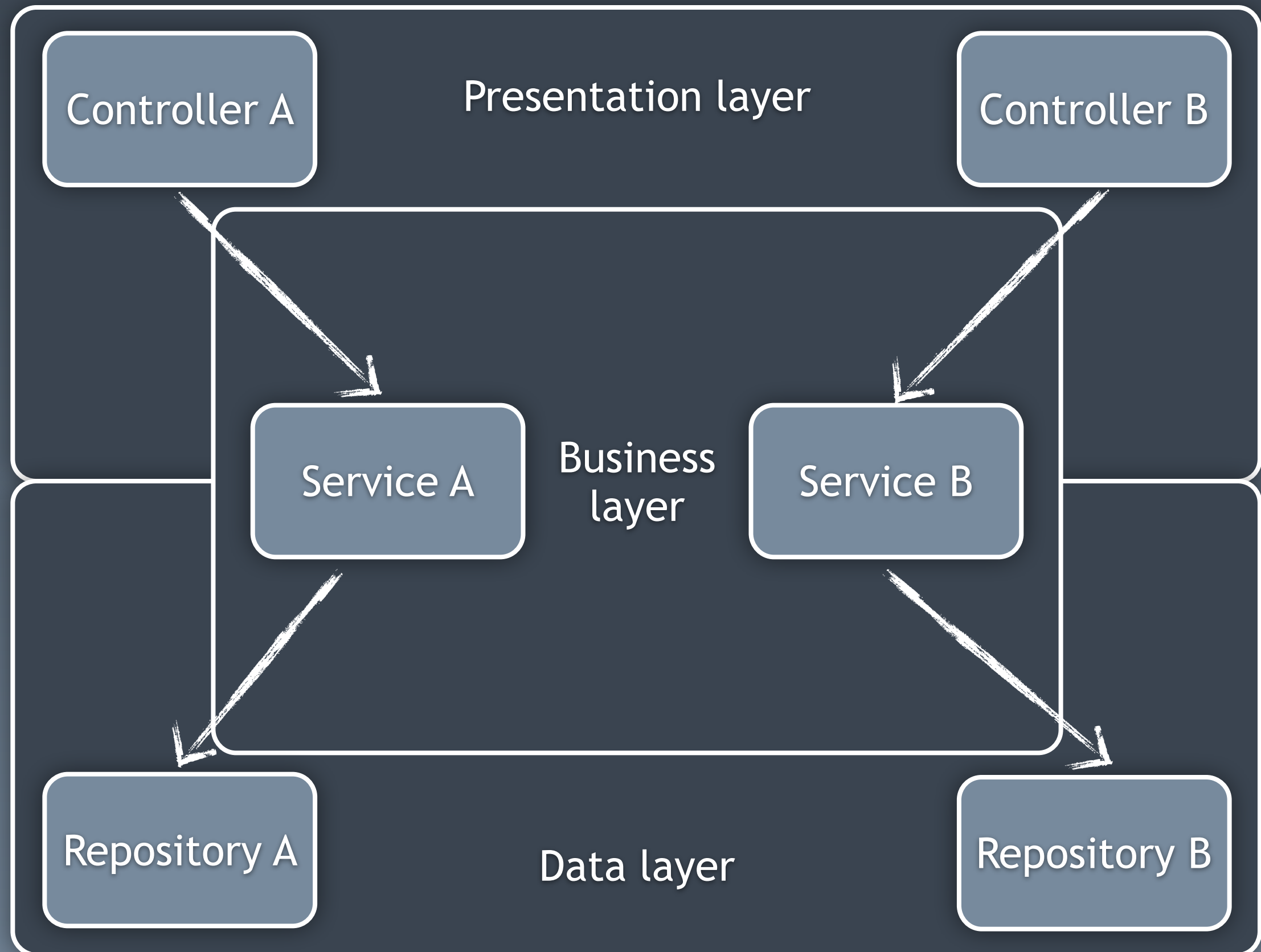
Should layers be
considered
harmful?

Are layers significant

structural
elements

or just an

implementation
detail?

"the model-code gap"

Model-code gap. Your architecture models and your source code will not show the same things. The difference between them is the *model-code gap*. Your architecture models include some abstract concepts, like components, that your programming language does not, but could. Beyond that, architecture models include intensional elements, like design decisions and constraints, that cannot be expressed in procedural source code at all.

Consequently, the relationship between the architecture model and source code is complicated. It is mostly a refinement relationship, where the extensional elements in the architecture model are refined into extensional elements in source code. This is shown in Figure 10.3. However, intensional elements are not refined into corresponding elements in source code.

Upon learning about the model-code gap, your first instinct may be to avoid it. But reflecting on the origins of the gap gives little hope of a general solution in the short term: architecture models help you reason about complexity and scale because they are abstract and intensional; source code executes on machines because it is concrete and extensional.

Organisation
of code
vs
the architectural views
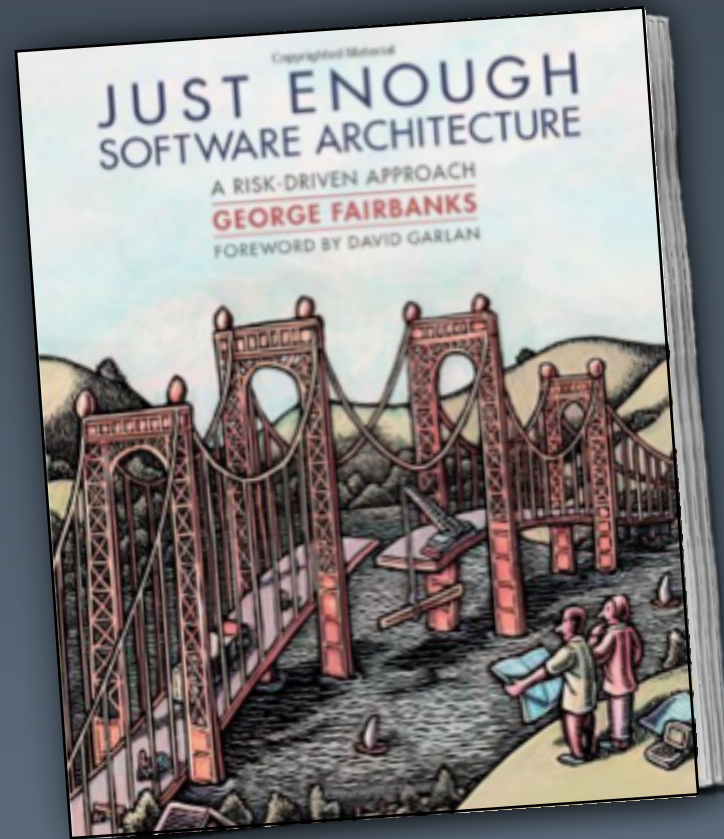
Software architecture vs code

	Business Model	Domain Model	Design Model		Code Model
			Boundary Model	Internals Model	
Bosch			System context	Component design	Code
Cheesman & Daniels		Business concept	Type specs	Component architecture	Code
D'Souza (MAp)	Business architec- ture	Domain	Blackbox	Whitebox	Code
Software Engineering Institute (SEI)			Requirements	Architecture	Code
Jackson		Domain	Domain + machine	Machine	
RUP	Business modeling	Business modeling	Requirements	Analysis & design	Code

Merge

the code and the model

(not model-driven architecture!)

"architecturally-evident coding style"

(subclassing, naming conventions, module dependencies, package structure, ...)

It's 2014, why can't we
auto-generate
a useful set of software
architecture diagrams
from code?!

Diagramming tools see

packages

and classes

rather than components

Architecture

config

aspectj   scope

framework

interceptor   target

support

org.springframework.aop

Diagrams & rules

Define your desired architecture with Structure101's semantically rich, intuitive, layered block architecture diagrams, present it to your developers through our IDE plugin and enforce it by integrating into your build.
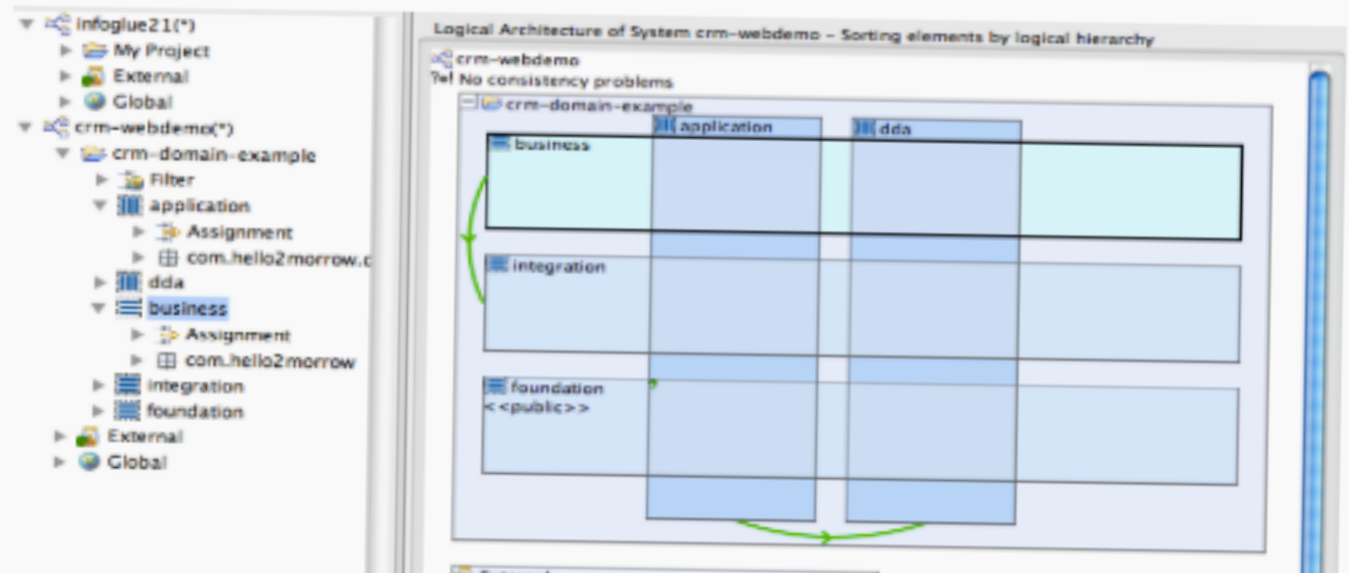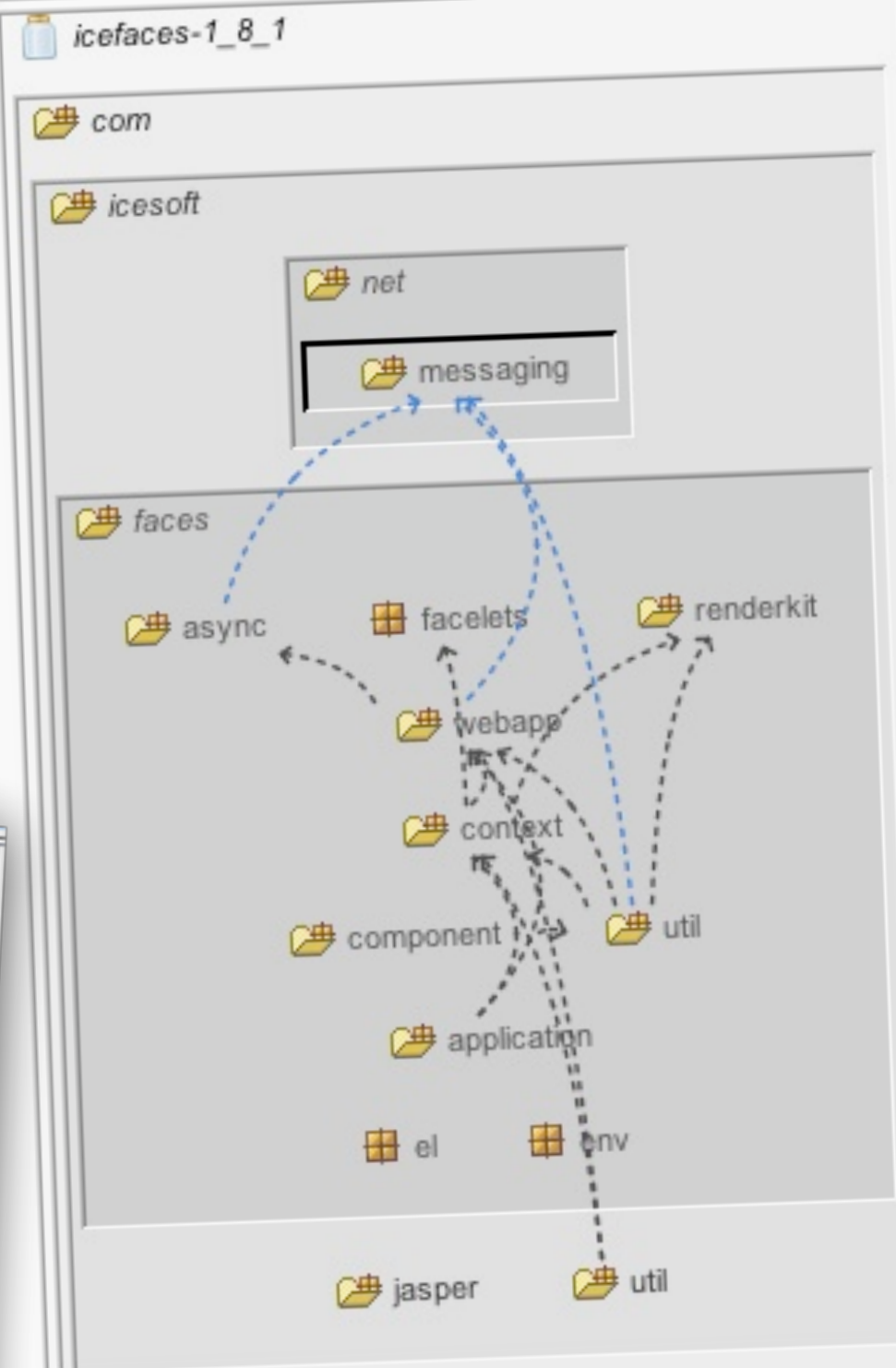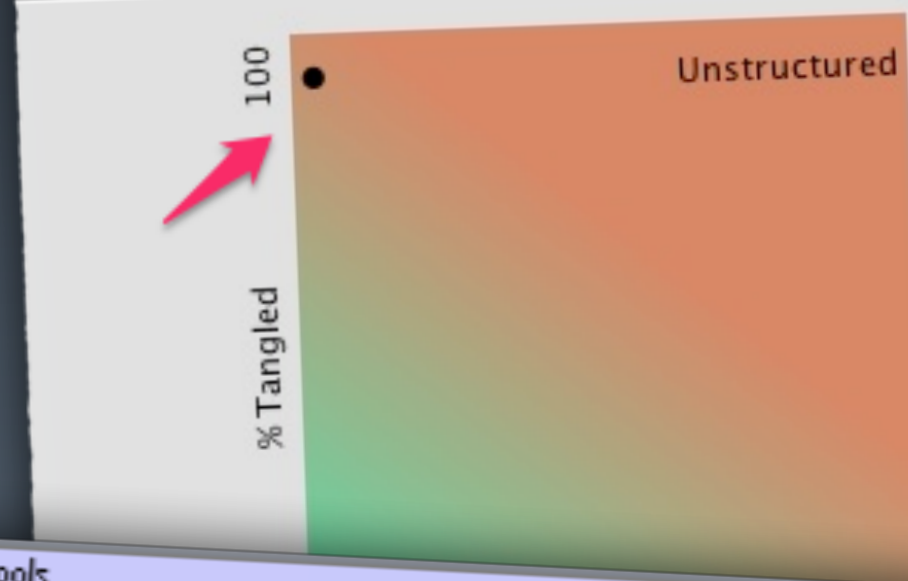
Design

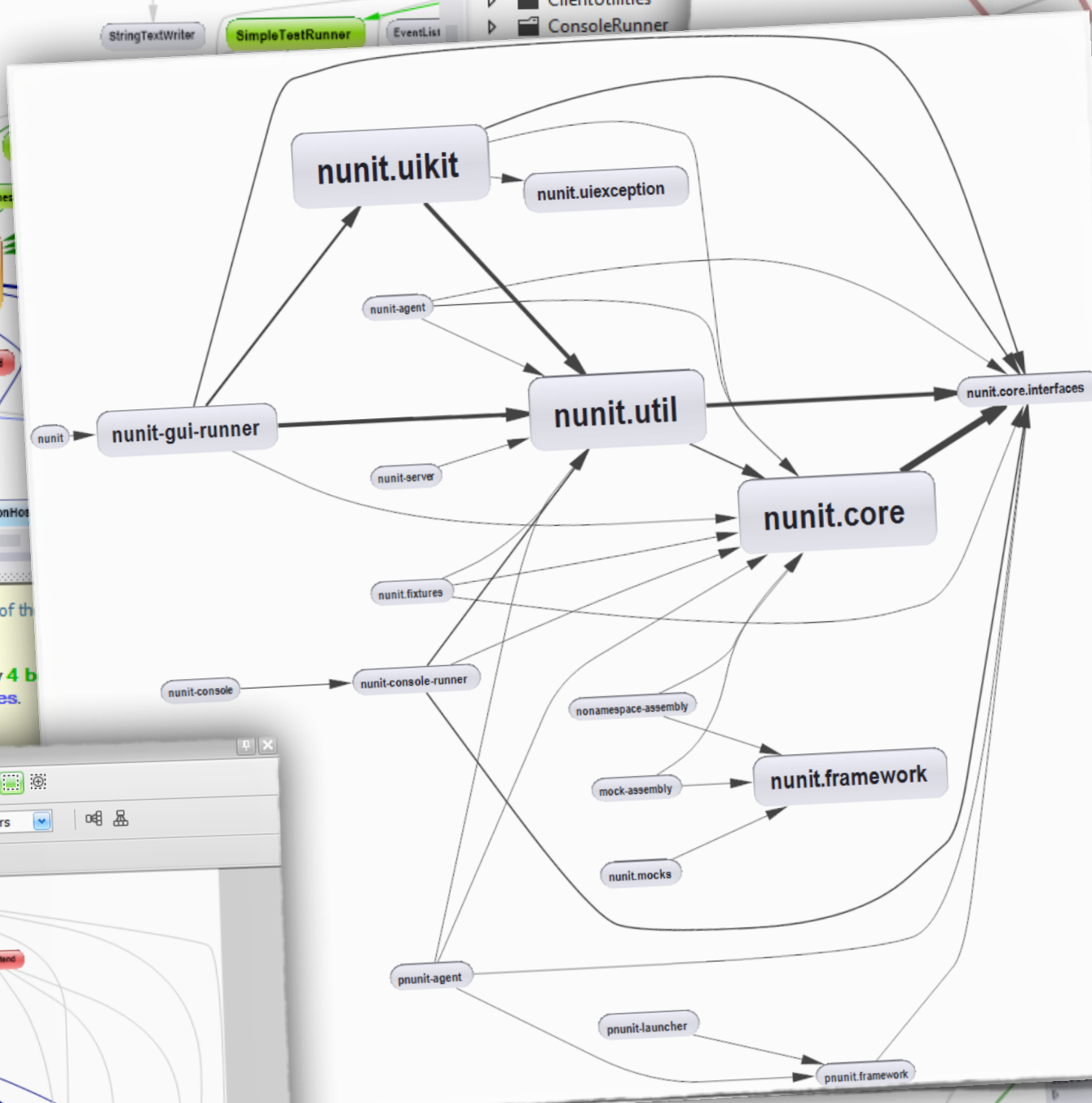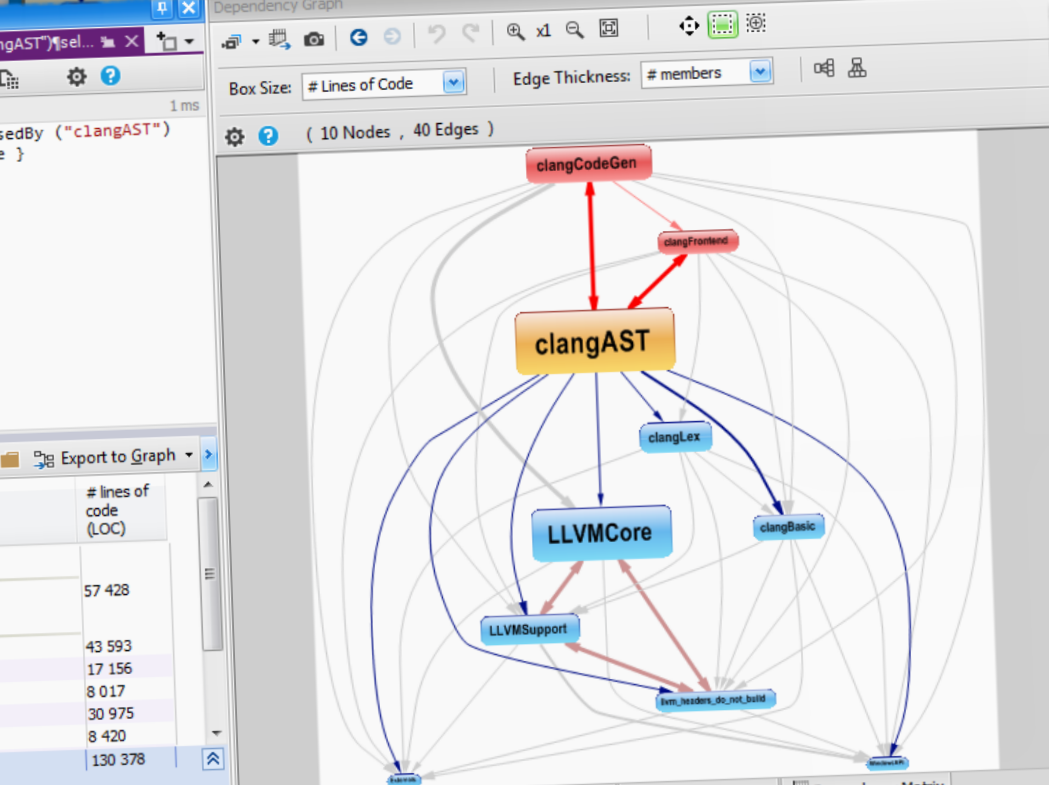### Define and Enforce Architecture

Sonargraph-Architect is designed to simplify software architecture and dependency management. You begin by defining the intended logical architecture of your system and map it to your code. A logical architecture is a set of rules designating allowed and forbidden dependencies in your code. Typical rules would be "the UI layer is not allowed to directly access the DAO layer" or "the UI layer cannot use JDBC directly".

Once defined, our IDE plugins (for Eclipse or IntelliJ IDEA) check every code change for rule compliance. Violating code lines will be marked with error markers so that developers are able to fix rule violations even before they commit their changes to the version control system. If you are not using Eclipse or IntelliJ, you can always use Sonargraph-Architect standalone for immediate rule checking. Moreover Sonargraph-Architect integrates with ANT and Maven so that architecture rules can also be enforced in your build process.

Structural over-complexity

100

% Tangled

Unstructured

Levelized Structure Map (LSM)

icefaces-1_8_1

com

icesoft

net

messaging

faces

async   facelets   renderkit

webapp

context

component   util

application

el   env

jasper   util

— org.apache.tools
+ ant.taskdefs
— ant
+ load..   + input   + listener   + help..   + filters
+ util
+ types
+ *
+ UTILITIES

$root

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+ cvslib	1														
+ compilers	2					4									
+ rmic	3					10									
+ condition	4					47						6	1	2	
+ email	5					3		23							
+ *	6	50	61	41	6		9							33	
+ loader	7														
+ listener	8														
+ helper	9					4									
+ input	10					7							1	2	
+ filters	11													10	
+ util	12					25						16	48		
+ types	13	3	13	5	3	1	284	3	2	18	2	33		52	64
+ *	14	5	297	67	4	8	1378					88	95		49
UTILITIES	15	66	79	59	69	75	2800	5	18	300	18	60	175	585	
						14	123		10					10	

org.apache.tools   ant.taskde...   ant

nunit - Microsoft Visual Studio

FILE   EDIT   VIEW   PROJECT   BUILD   DEBUG   TEAM   SQL   NDEPEND   TOOLS   TEST   ARCHITECTURE   ANALYZE   WINDOW   HELP

Start   Debug   webtoc.html

Dependency Graph   Metrics

Box Size: # Lines of Code

Edge Thickness: # members   (83 Nodes , 159 Edges)

BufferedStringTextWriter

StringTextWriter   SimpleTestRunner   EventList

Solution Explorer

Search Solution Explorer (Ct...

Solution 'nunit' (29 pr...
ClientUtilities
ConsoleRunner

ProjectRootSuite

dMethodSuite   ParameterizedFixtureSuite   Names

TestSuite

TypeHelper   TestDecorator   TestFixtureBuilder   TestSuiteThread

TestMethod

ectedExceptionProcesso   NullListener   TestMethodThread   ExtensionHos

Queries and Rules Explorer

Context-Sensitive Help   Show description of th

The box pointed represents the class **TestSuite**.

- It is used directly by **13 boxes**, and indirectly by **4 b**
- It uses directly **9 boxes**, and indirectly **22 boxes**.
- It is directly mutually dependent with **1 box**.

Dependency Graph

Box Size: # Lines of Code   Edge Thickness: # members

(10 Nodes , 40 Edges)

clangCodeGen

clangFrontend

clangAST

clangLex

LLVMCore   clangBasic

LLVMSupport

llvm_headers_do_not_build

lines of code (LOC)

57 428
43 593
17 156
8 017
30 975
8 420
130 378

nunit.uikit   nunit.uiexception

nunit-agent

nunit-gui-runner   nunit.util   nunit.core.interfaces

nunit-server

nunit.core

nunit.fixtures

nunit-console   nunit-console-runner

nonamespace-assembly

mock-assembly   nunit.framework

nunit.mocks

pnunit-agent

pnunit-launcher

pnunit.framework

Dictionary<TKey,TValue>
+ValueCollection
+Enumerator

Mscorlib_DictionaryDebugV
iew<K,V>

Dictionary<TKey,TValue>
+Enumerator

Dictionary<TKey,TValue>
+Entry

Mscorlib_DictionaryValueC
ollectionDebugView<TKey
,TValue>

Mscorlib_DictionaryKeyCol
lectionDebugView<TKey
,TValue>

Dictionary<TKey,TValue>

Mscorlib_CollectionDebugV
iew<T>

List<T>

ObjectEqualityComparer<T>

GenericEqualityComparer<T>

ByteEqualityComparer

NullableEqualityComparer<
T>

EqualityComparer<T>

Comparer<T>

ExecutionEngine
.getConstantValue
(constIlvm::Constant*)

Do OO programming languages provide the wrong abstractions for building modern software systems?

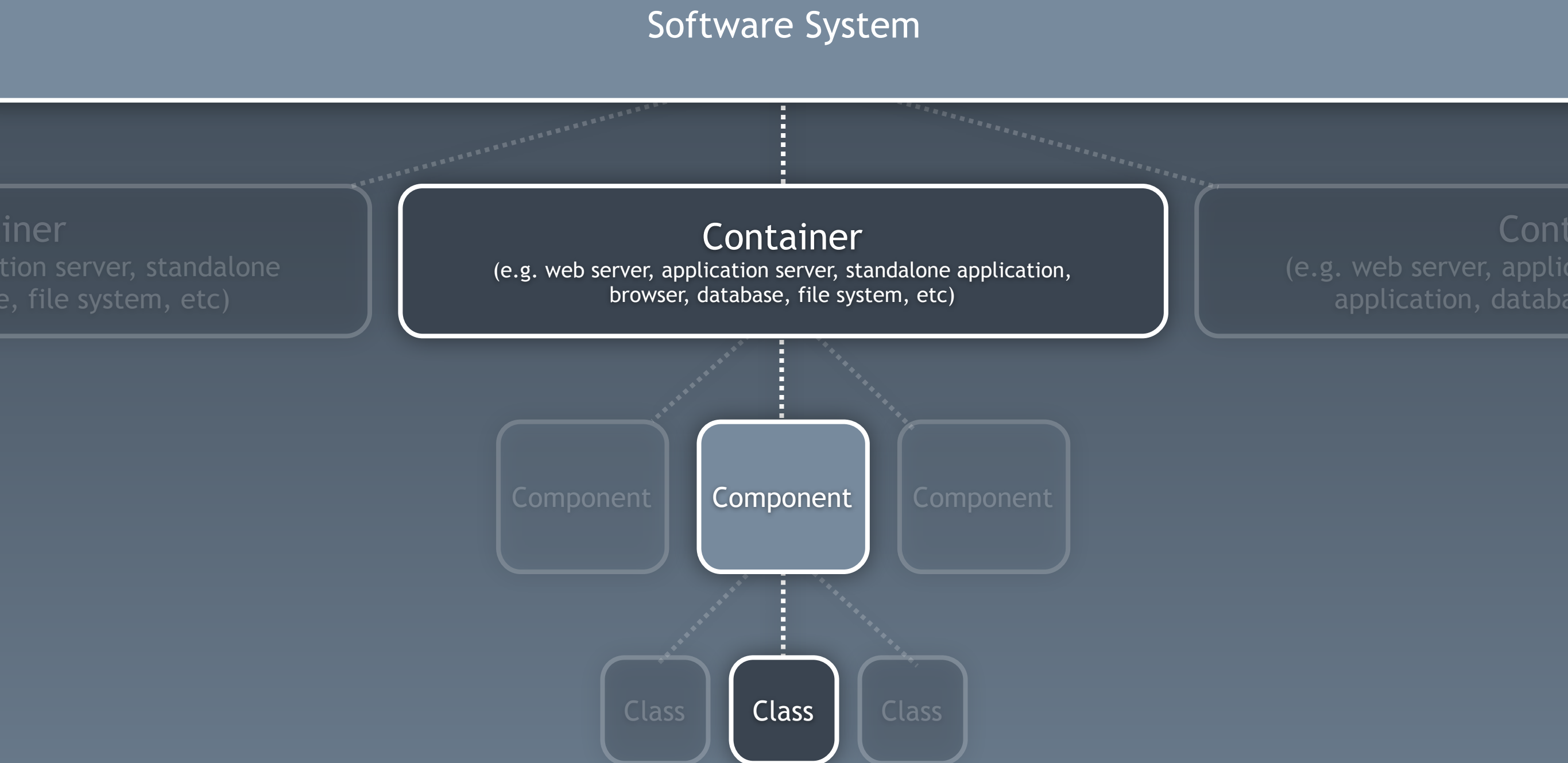The **intersection** of software architecture and code

Abstractions

on diagrams
should reflect the

code

A common set of abstractions

is more important than

a common notation

Software System

Container
(e.g. web server, application server, standalone application,
browser, database, file system, etc)

Container
(e.g. web server, standalone
..., file system, etc)

Container
(e.g. web server, appli...
application, databa...

Component

Component

Component

Class

Class

Class

Agree on a simple set of abstractions
that the whole team can use to communicate

The C4 model

System Context
The system plus users and system dependencies

Containers
The overall shape of the architecture and technology choices
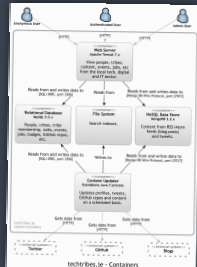
Components
Logical components and their interactions within a container
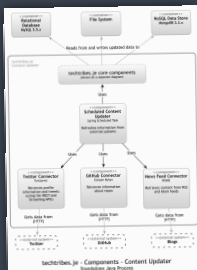
Classes
Component or pattern implementation details
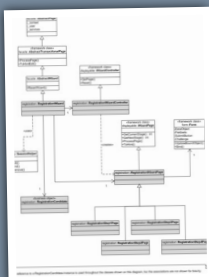
This **isn't** about creating a standard

I do use UML

(activity, class, sequence, collaboration, state)

techtribes.je

Context

- What are we building?

- Who is using it?
(users, actors, roles, personas, etc)

- How does it fit into the existing IT environment?
(systems, services, etc)

Anonymous User

Authenticated User
(sign-in with Twitter ID)

Admin User
(sign-in with Twitter ID)

Manage user profile and tribe membership.

View people, tribes (businesses, communities and interest groups), content, events, jobs, etc from the local tech, digital and IT sector.

Add people, add tribes and manage tribe membership.

techtribes.je

Gets profile information and tweets from

Gets information about code repositories from

Gets content from RSS and Atom feeds from

<<external system>>
Twitter

<<external system>>
GitHub

<<external system>>
Blogs

techtribes.je - Context

Containers

- What are the high-level technology decisions? (including responsibilities)

- How do containers communicate with one another?

- As a developer, where do I need to write code?

techtribes.je - Containers
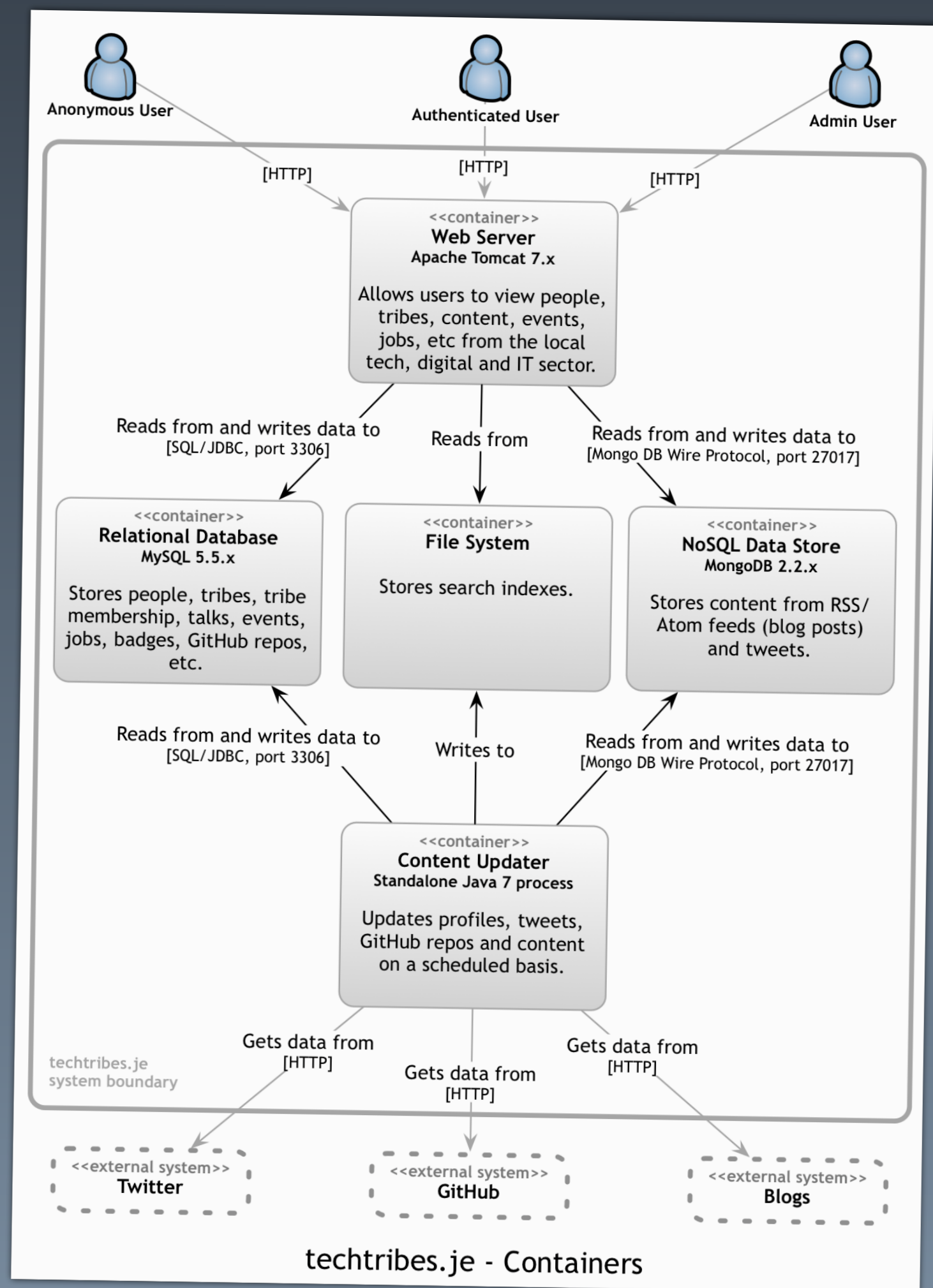
Components

- What components/ services is the container made up of?

- Are the technology choices and responsibilities clear?

techtribes.je - Components - Content Updater
Standalone Java Process

Components

techtribes.je core components (Java)
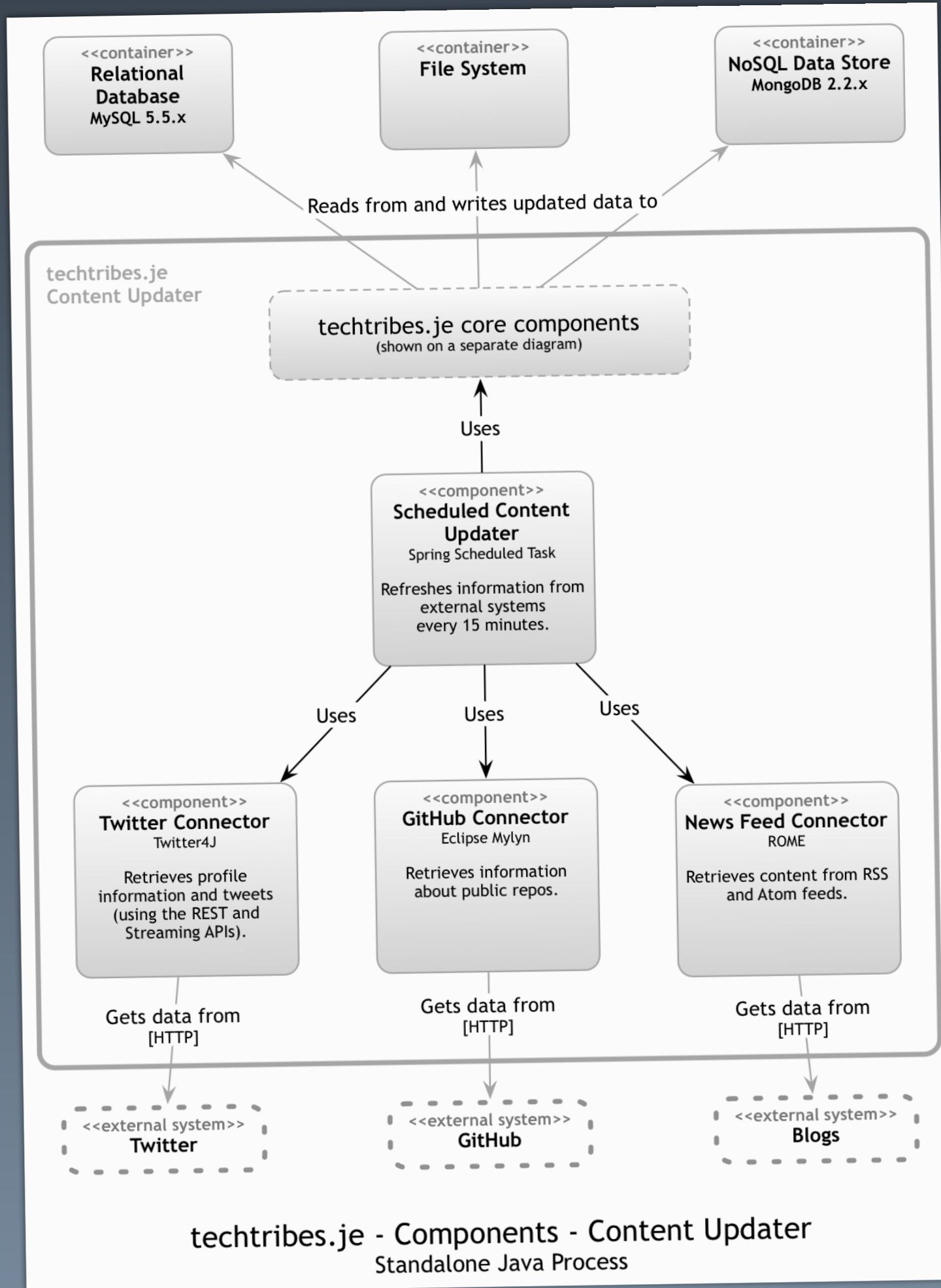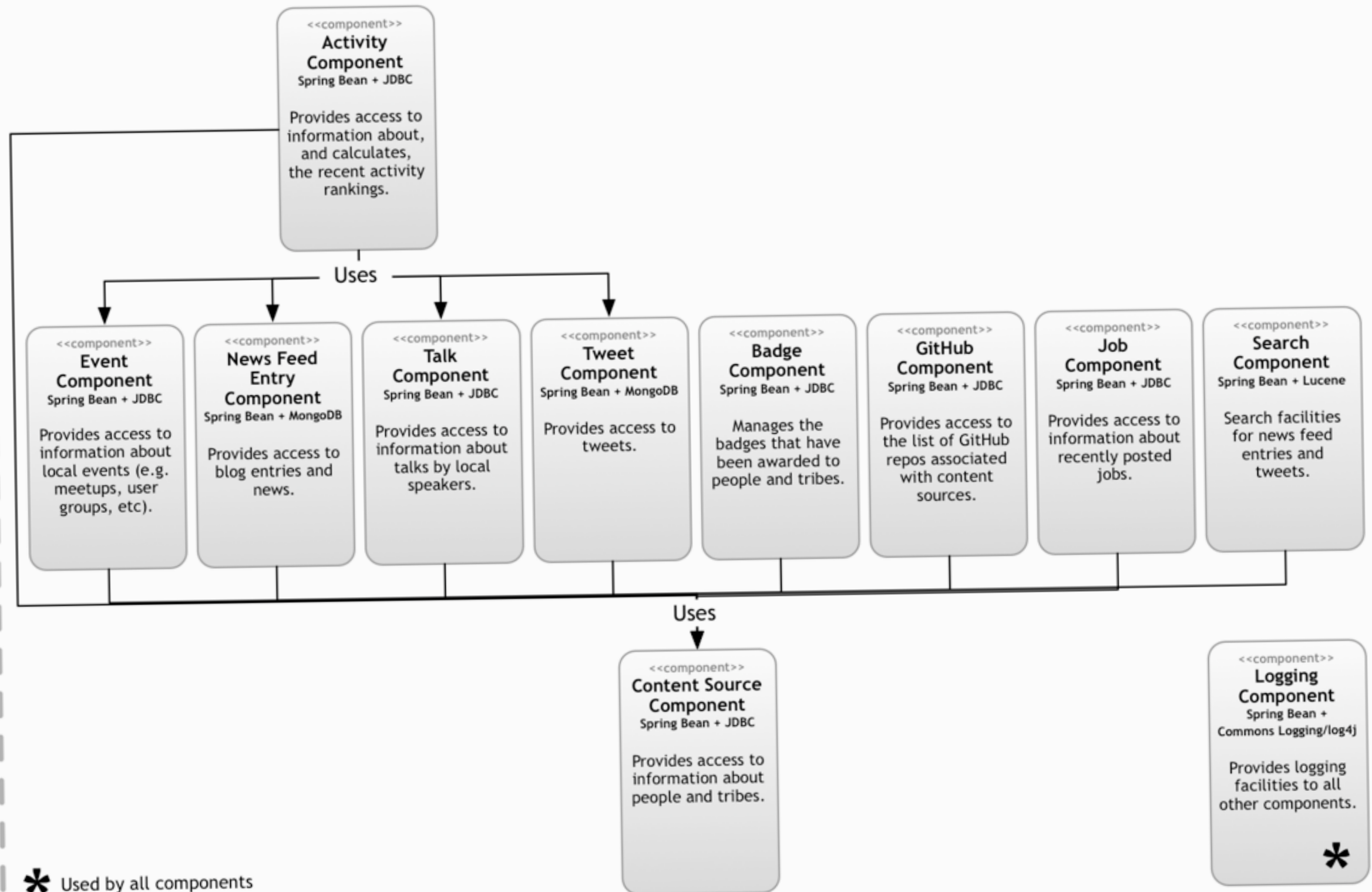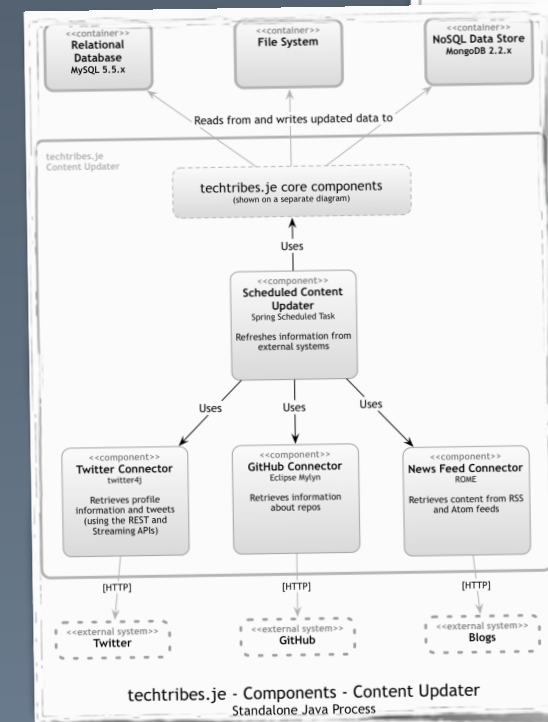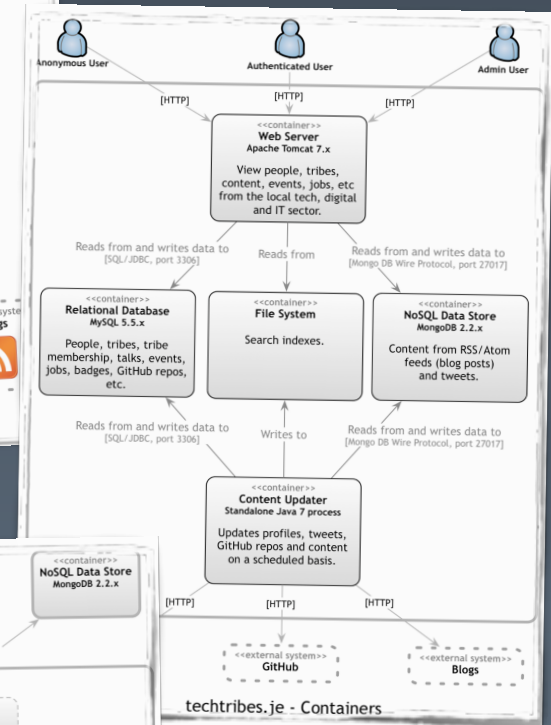
Sketches are maps

that help a team navigate a complex codebase
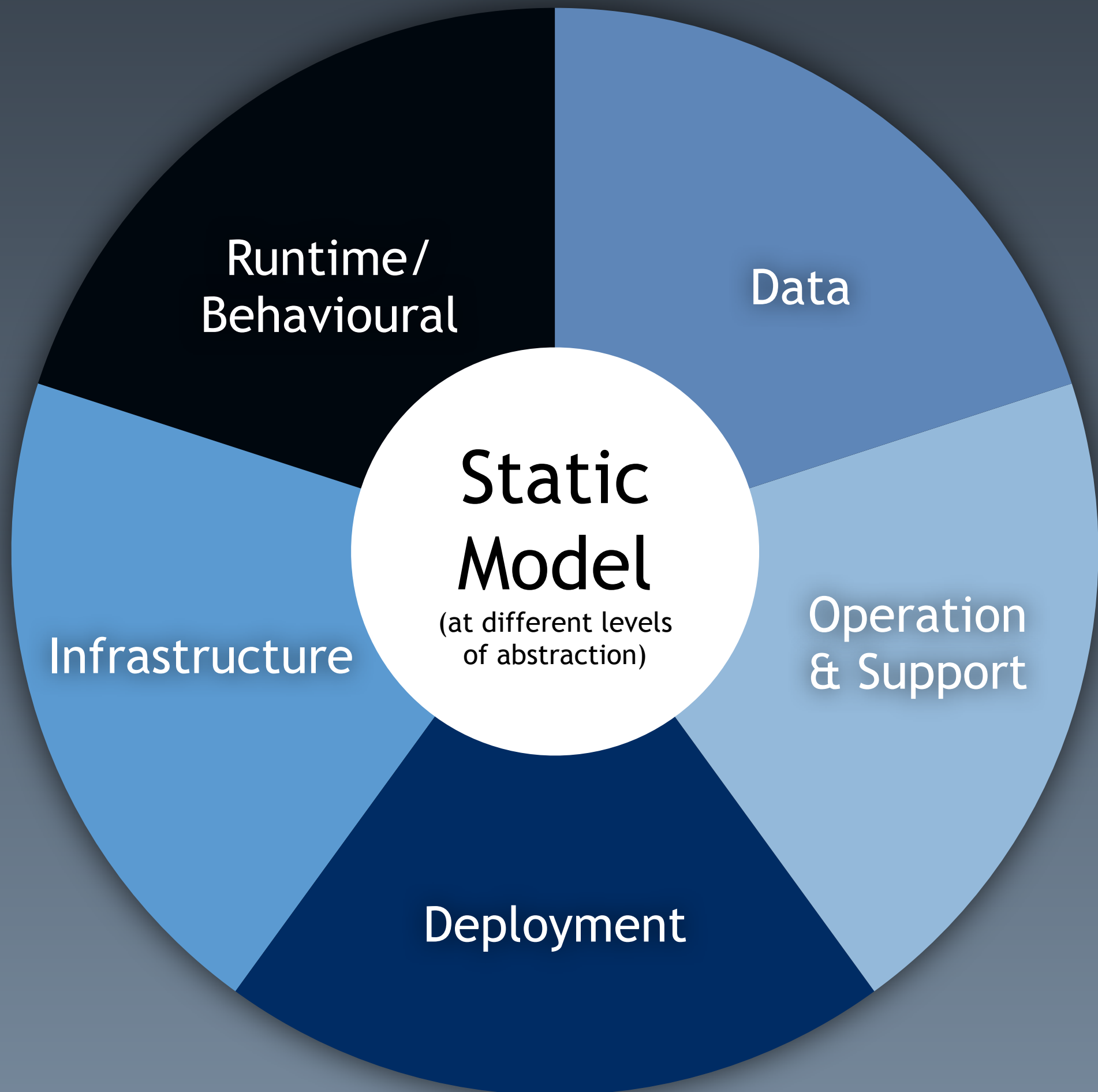
Software developers are the most important stakeholders of software architecture

C4 is about the static structure of software, which is ultimately about code

Does your code reflect the abstractions that appear on your software architecture diagrams?
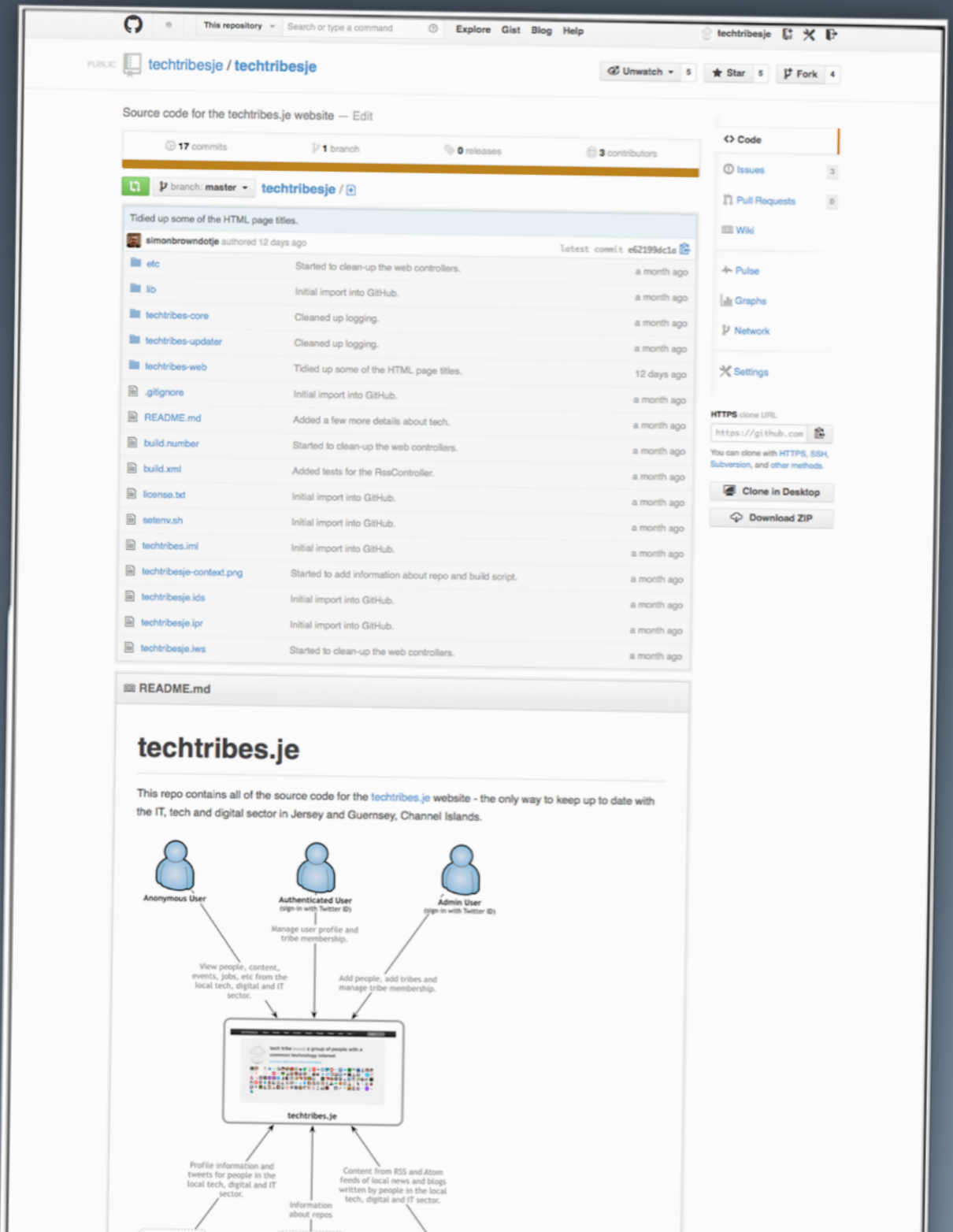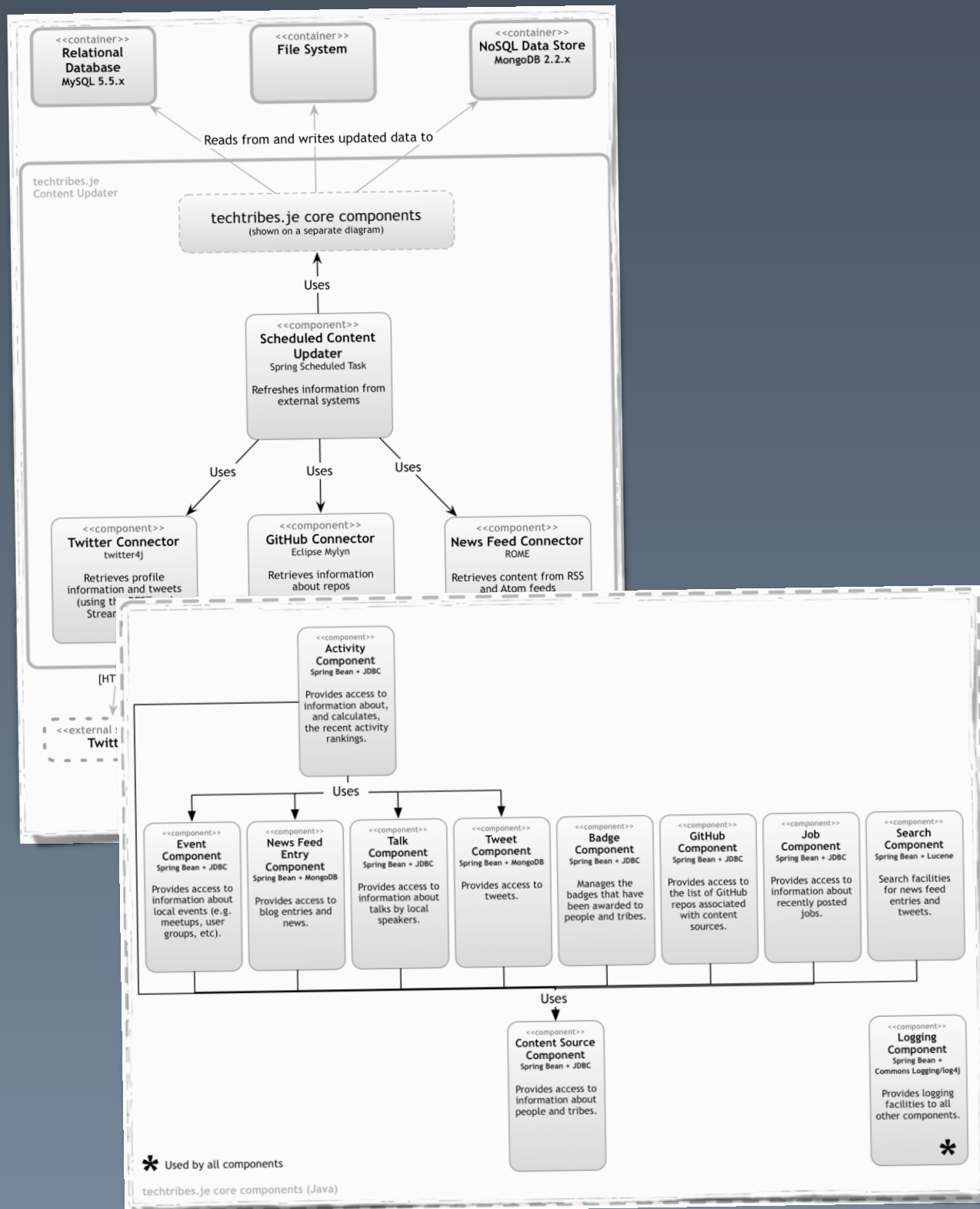
If the answer is "no" ... are the diagrams actually useful?

Does the code for
techtribes.je
reflect the abstractions
on the software
architecture diagrams?

https://github.com/techtribesje/techtribesje

Tomcat 7.0

techtribesje | techtribes-core | src | je | techtribes | component | tweet

Project

I TweetComponent.java | C TweetComponentImpl.java | C MongoDbTweetDao.java

```java
package je.techtribes.component.tweet;

import ...

@Component(responsibility = "Provides access to tweets.")
@Collaboration(target = ContentSourceComponent.class, responsibility = "Enriches with")
public interface TweetComponent {

    /**
     * Gets the most recent tweets by page number.
     */
    List<Tweet> getRecentTweets(int page, int pageSize);
```

- techtribesje [techtribes] (~/Documents/sandbox/techtribesje)
  - docs
  - lib
  - techtribes-core
    - sql
    - src
      - je.techtribes
        - component
          - activity
          - badge
          - contentsource
          - event
          - github
          - job
          - log
          - newsfeedentry
          - search
          - talk
          - tweet
            - component.xml
            - C MongoDbTweetDao
            - I TweetComponent
            - C TweetComponentImpl
            - TweetException
        - domain
        - util
        - config.xml
    - test
    - techtribes-core.iml
  - techtribes-updater
  - techtribes-web
  - .gitignore
  - build.number
  - build.xml
  - license.txt
  - README.md
  - setenv.sh
  - software-architecture-for-developers.png
  - techtribes.iml

6: TODO   9: Changes   Application Servers

<<component>>
Activity Component
Spring Bean + JDBC

Provides access to information about, and calculates, the recent activity rankings.

Uses

<<component>>
Event Component
Spring Bean + JDBC

Provides access to information about local events (e.g. meetups, user groups, etc).

<<component>>
News Feed Entry Component
Spring Bean + MongoDB

Provides access to blog entries and news.

<<component>>
Talk Component
Spring Bean + JDBC

Provides access to information about talks by local speakers.

<<component>>
Tweet Component
Spring Bean + MongoDB

Provides access to tweets.

<<component>>
Badge Component
Spring Bean + JDBC

Manages the badges that have been awarded to people and tribes.

<<component>>
GitHub Component
Spring Bean + JDBC

Provides access to the list of GitHub repos associated with content sources.

<<component>>
Job Component
Spring Bean + JDBC

Provides access to information about recently posted jobs.

Uses

<<component>>
Content Source Component
Spring Bean + JDBC

Provides access to information about people and tribes.

Did it start out that way?

Package by layer

A component is often a combination of a number of classes in different layers.

What's a "component"?

Package by layer

je.techtribes.service

<<interface>>
TweetService

DefaultTweetService

<<uses>>

<<interface>>
TweetDao

MongoDBTweetDao

je.techtribes.data

A component is often a combination of a number of classes in different layers.

Package by component

je.techtribes.component.tweet

<<interface>>
TweetComponent

TweetComponentImpl

<<uses>>

MongoDBTweetDao
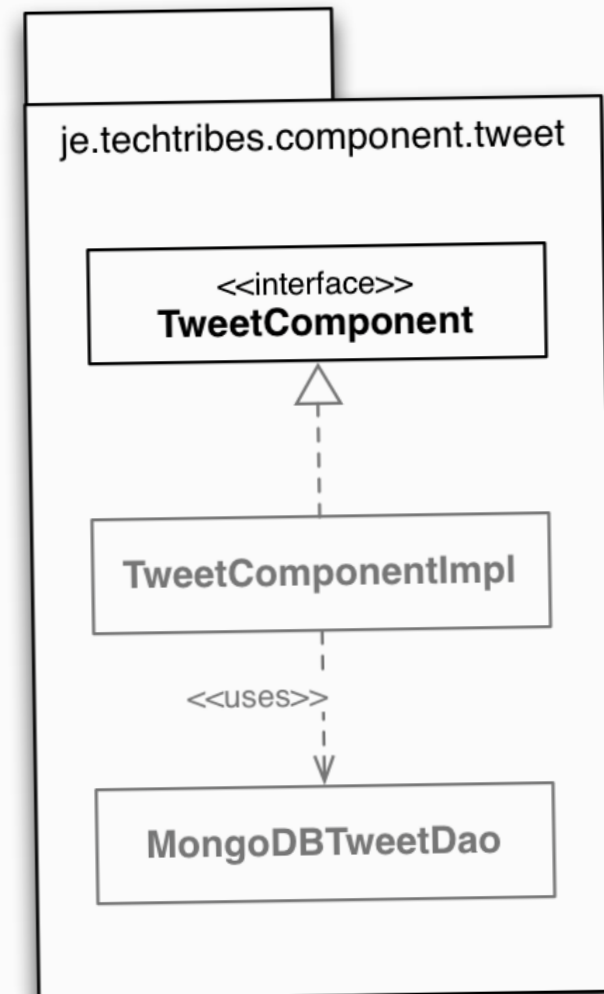
What's a "component"?

Don't do unit testing!

"In the early days of computing when computers were slow, unit tests gave the developer more immediate feedback about whether a change broke the code instead of waiting for system tests to run. Today, with cheaper and more powerful computers, that argument is less persuasive."

RBCS

TIME TESTED.
TESTING IMPROVED.
www.RBCS-US.com

Why Most Unit Testing is Waste

By James O Coplien

1.1 Into Modern Times

Unit testing was a staple of the FORTRAN days, when a function was a function and was sometimes worthy of functional testing. Computers computed, and functions and procedures

"If your coders have more lines of unit tests than of code, it probably means one of several things.
... Or the problem may be at the other end:
developers don't have adequately refined design skills, or the process doesn't encourage architectural thinking and conscientious design."

"do not let your tests drive your design"

Test-induced design damage

By David Heinemeier Hansson on April 29, 2014

"Code that's hard to test in isolation is poorly designed", goes a common TDD maxim. Isolation meaning free of dependent context and separated from collaborators, especially "slow" ones like database or file IO. The prevalent definition of "unit" in unit testing (though not everyone agrees with this).

This is some times true. Some times finding it difficult to test points to a design smell. It may be tangled responsibilities or whatever. But that's a far ways off from declaring that hard-to-unit-test code is always poorly designed, and always in need of repair. That you cannot have well-designed code that is hard to unit test.

Instead of unit testing everything, what about testing your significant structural elements as black boxes?

Don't blindly copy what everybody else seems to be doing

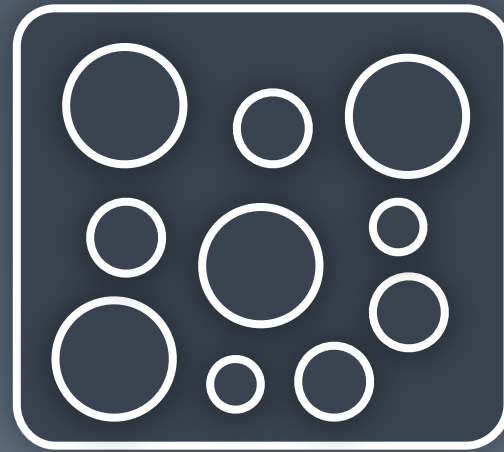To teach something,
you need to understand it

The point of this?

A good architecture enables

agility

Monolithic architecture

Service-based architecture
(SOA, micro-services, etc)

Something in between
(components)

The structure of your software and the decomposition strategy you use to get there are important

Inspect

and

Adapt

Think about how to align the

software

architecture

and the

code

A simple
and explicit
mapping assists
on many levels

Until we find the one true solution that works in all contexts, you'll need to think

If your software system is hard to work with,

change it!