

The lost art of software architects



Simon Brown



@simonbrown

For every person on Twitter telling
you that teams don't need software
architects, and how technical
leadership should be a shared
responsibility...

just hire good people
and trust them to do
the right thing

...there's another on Reddit describing their absolute tire fire of a team that followed fashion, and has no technical leadership at all. 🙄



43



r/ExperiencedDevs · Posted by u/PoliteRedditUsername 17 hours ago



Is it worth giving a hoot about code smell?

So I just started a new job a few months ago and it's increasingly becoming apparent to me the code base is about 6/10 in terms of overall quality. The whole thing is basically one big band-aid, years of bugfixes on top of bugfixes. Best practice is eschewed in favor of the 'easy and fast' solution - which is leading to major code stink, more bugs, and the code is just a mess and a pain in the ass to do anything in. State all over the place, everything needlessly tightly coupled. I'm sure you all have been there done that.

So mainly I'm wondering - is this sort of situation just a lost cause? I'm just a regular mid-level dev without much say in anything, and the people that made these decisions are senior to me by like 5 years. The part of me that takes pride in my work really wants to do and say something about this - but I don't want to become 'that guy' that always thinks he knows what's best, especially considering my relative lack of experience compared to my seniors. The alternative is to follow the established convention and write code that smells like hot trash in 2x the time it would take otherwise, with high chance of bugs and bullshit down the road.

What do?



Vote



r/ExperiencedDevs · Posted by u/lost__being 1 hour ago



How do teams handle the constant change in leadership

I actually am an inexperienced dev. But i see all the experienced folks in my team changing currently. Both our architect and tech lead left. Few other old folks also left. Our team is left with all people with 1-2 yoe and some newly hired folks with 5-10 yoe. My first question is - Is this common in our industry? I mean the Indian job market is blooming currently, people are being offered very high salaries on switching. But if it happens in all teams how do the teams cope with it? I have a feeling that the new folks would want to rewrite services now. We were already in the middle of creating a whole new product, doing POCs and stuff. I think some of these might be abandoned since the new leadership won't agree with what our old architect had thought. All in all should be worried that my team is heading in a bad direction?



89



 **r/ExperiencedDevs** · Posted by u/AwkwardInstance0 4 days ago



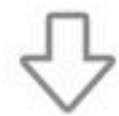
How to cope with staff turnover with lots of microservices

We have about 800 microservices, some of them not touched for years, and never heard of by anyone still employed here - maybe written in a language e.g. Perl that nobody really knows well.

How is this kind of thing not a killer problem elsewhere? How do staff leaving not make their services hard to maintain?



47



r/ExperiencedDevs · Posted by u/phone_dilemma 5 days ago 🤔



How do I navigate a humongous codebase at my new job?

Joined a big tech company. I'm handed tasks/tickets to resolve bugs and enhancements. Following things have tripped me.

1. Individual files with 30k lines, huge classes of 10k lines, and huge functions.
2. Nesting everywhere, nested conditionals, nested loops(5 levels), nested functions, and nested exception blocks (max is 4 levels of nesting),
3. Lot of circular dependencies between multiple files, we do some shenanigans to circumvent it.
4. A very important set of variables (around 400) all of which are in the global namespace.

They are all compressed into an executable(turning the code into an executable takes 30 mins), so each time I make changes I've to wait 30 mins for executables to run, which I've to test on a remote machine. Since it's an executable I'm not sure if I can attach some debugger to it. The only reason this codebase manages to work is due to tens of thousands of tests that take 10 to 12 hours to run. If it fails I'm stuck again.

Anyone who went through this? I've been working around 4 years and hence I guess my team expects I'd get by without too much handholding.

Thanks

Simon Brown

Independent consultant specialising in software architecture,
plus the creator of the C4 model and Structurizr

@simonbrown

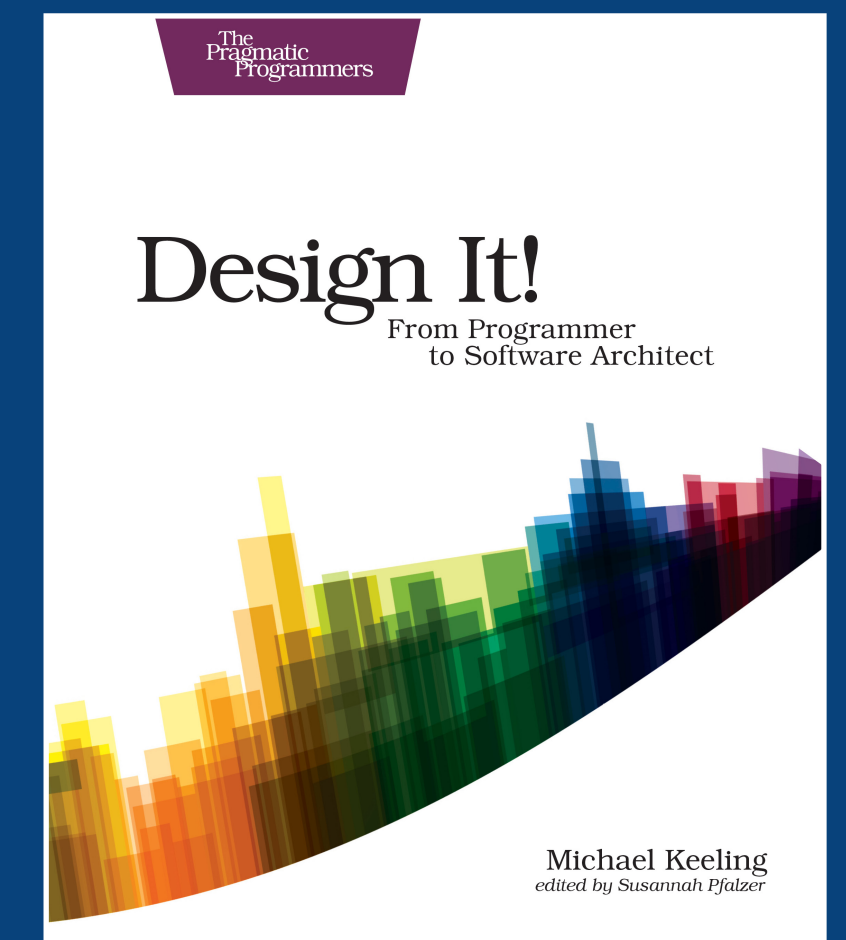
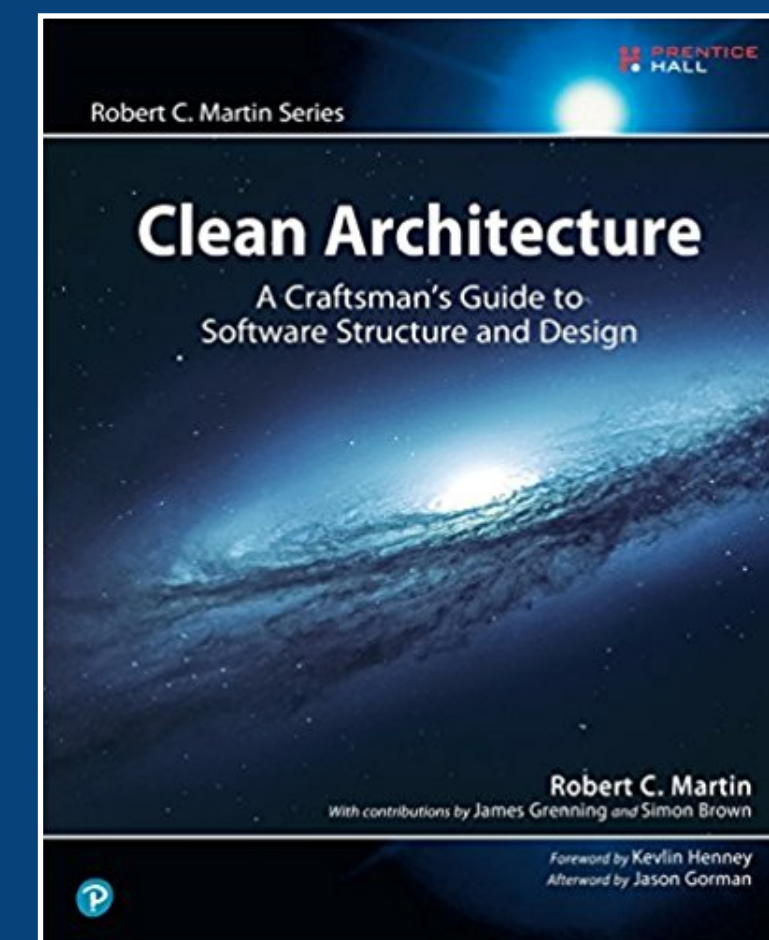
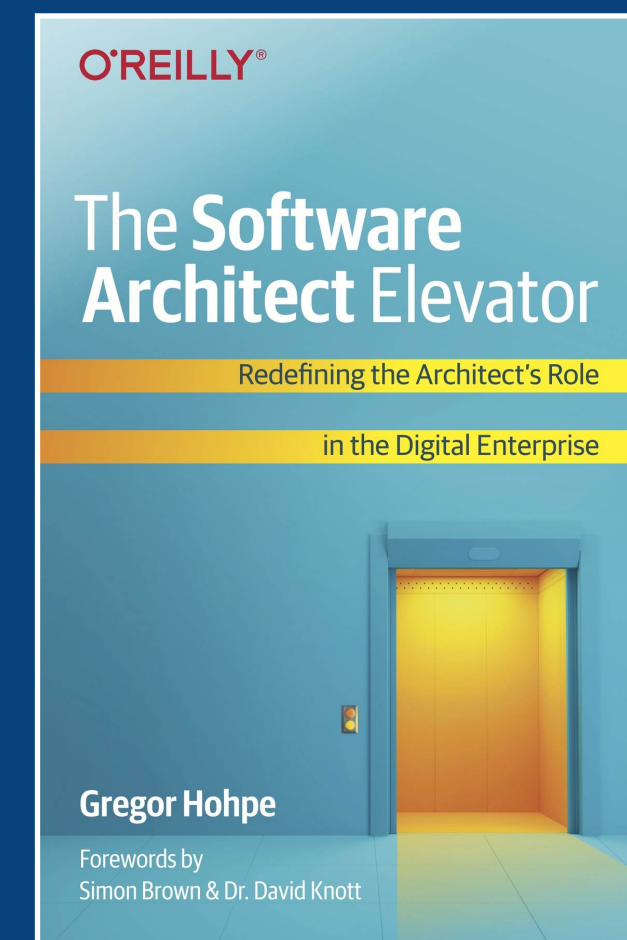
Software
architecture
for developers

Simon Brown

The
C4
model

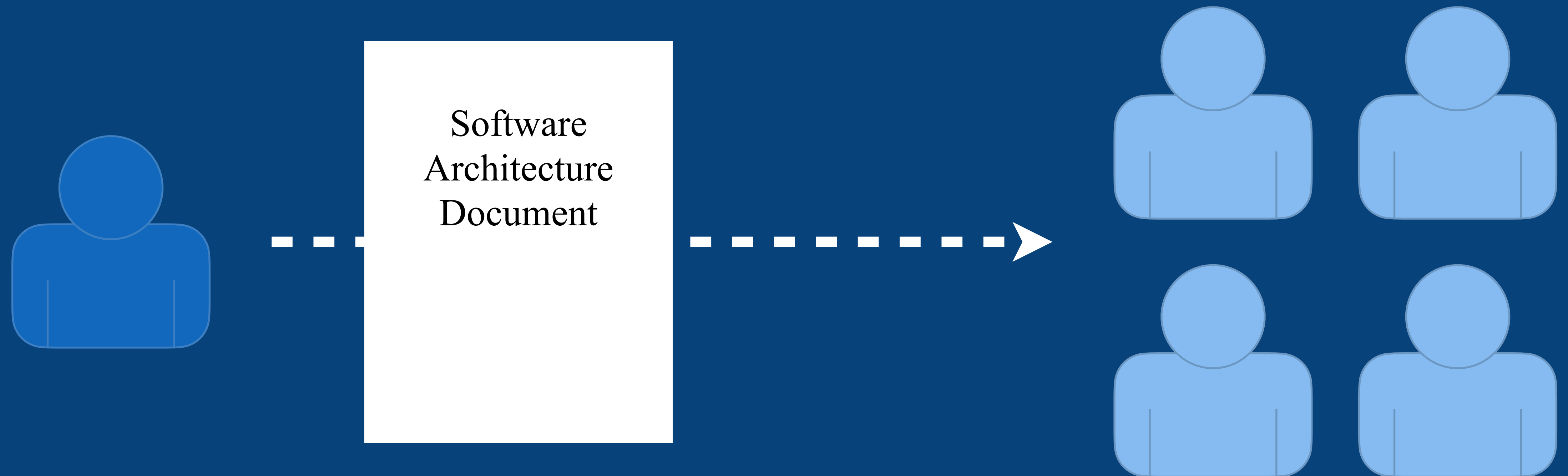
for visualising software architecture

Simon Brown



“Ivory tower architects”
“Architecture astronauts”

Software development is not a relay sport



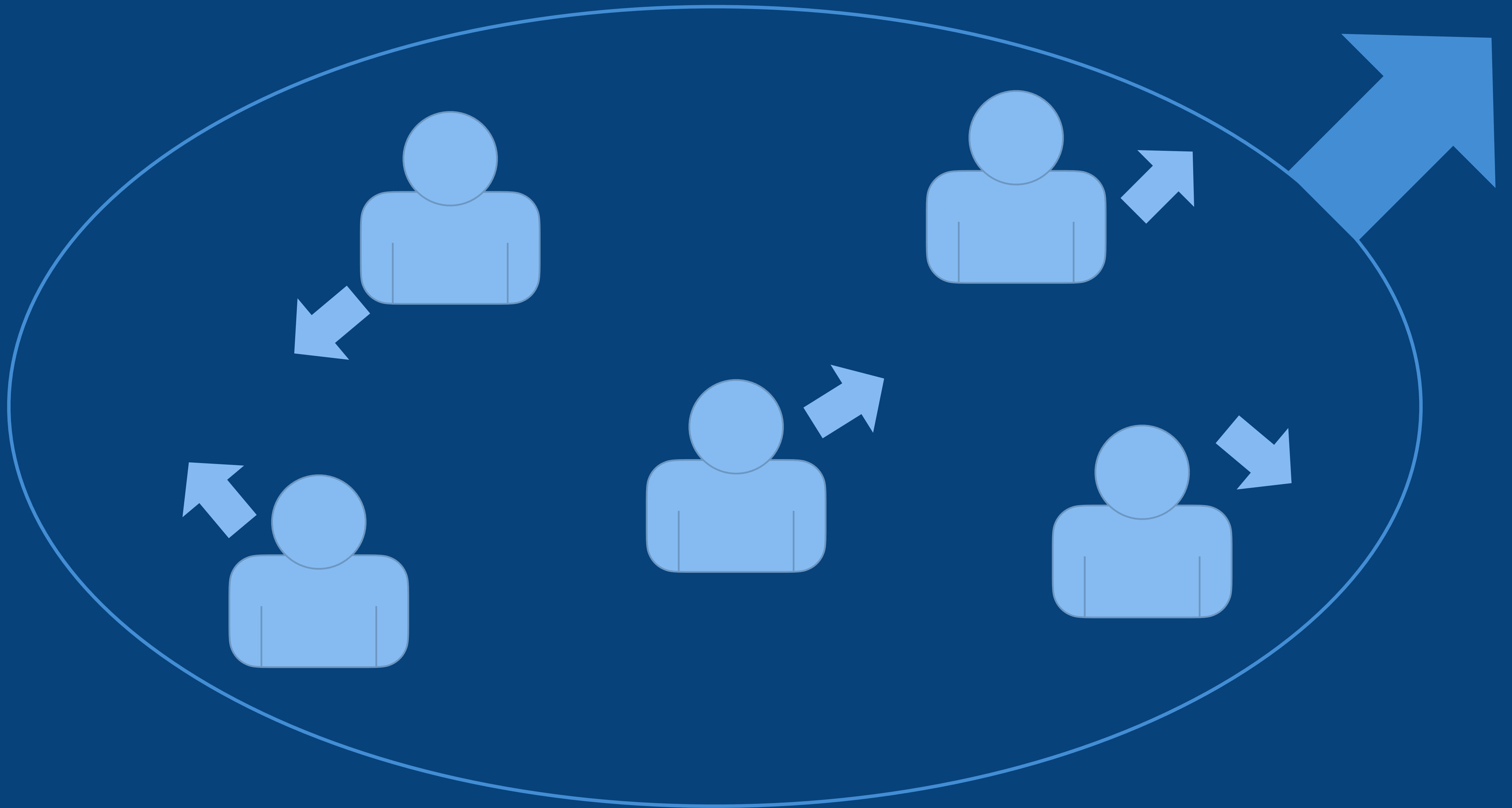
AaaS

Architecture as a Service

Software development teams
don't need "architects"

Software development teams
do need **technical leadership**

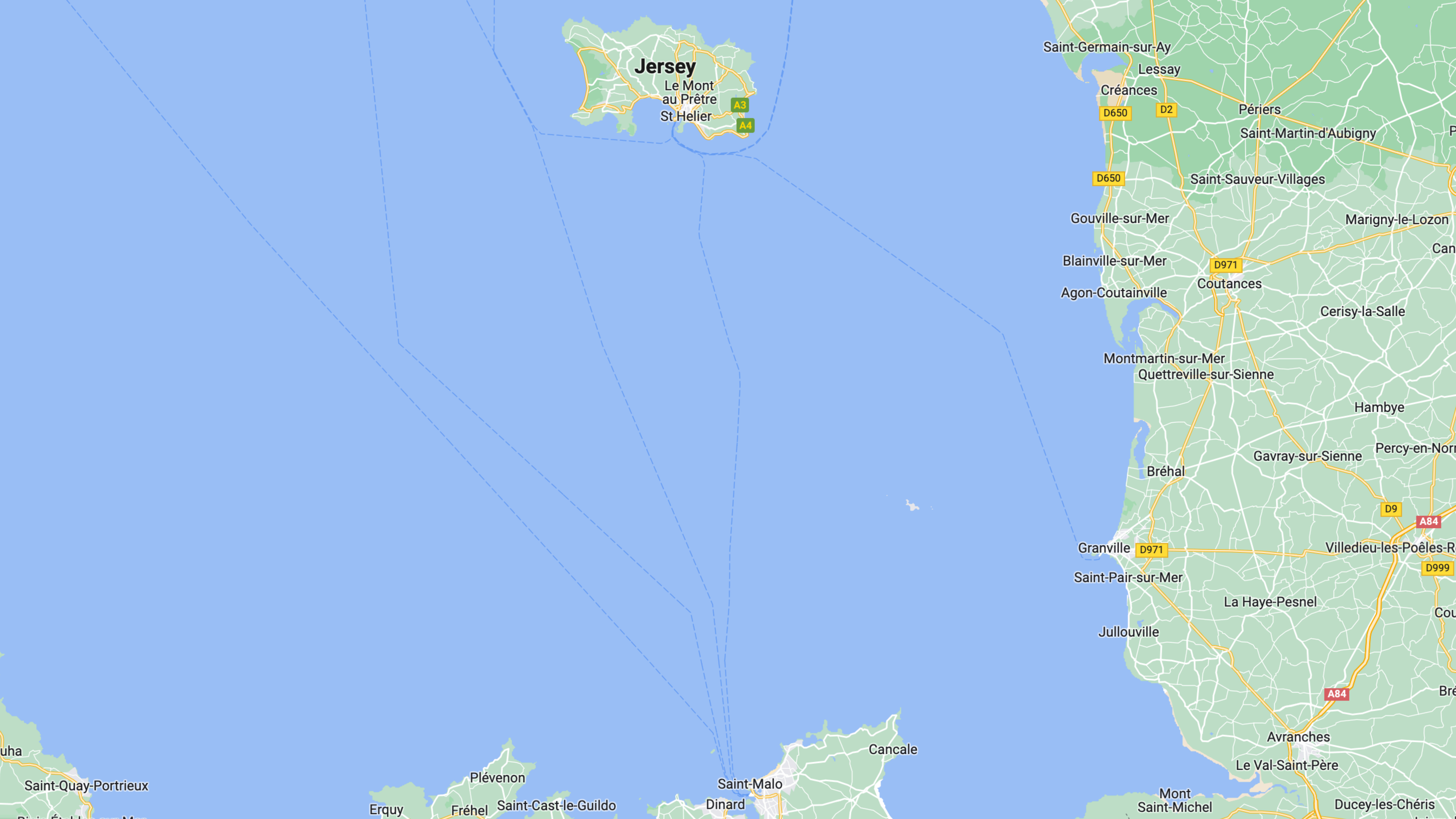
Every team needs
technical leadership



The background is a solid dark blue. On the left and right sides, there are decorative elements consisting of two parallel, slanted bars in a lighter shade of blue, creating a sense of depth or framing.

you're hiring
the wrong people

Continuous technical leadership



**Everybody should
be an architect**

With collective code ownership,
everybody is an architect

Everybody is an architect
... except when they're not



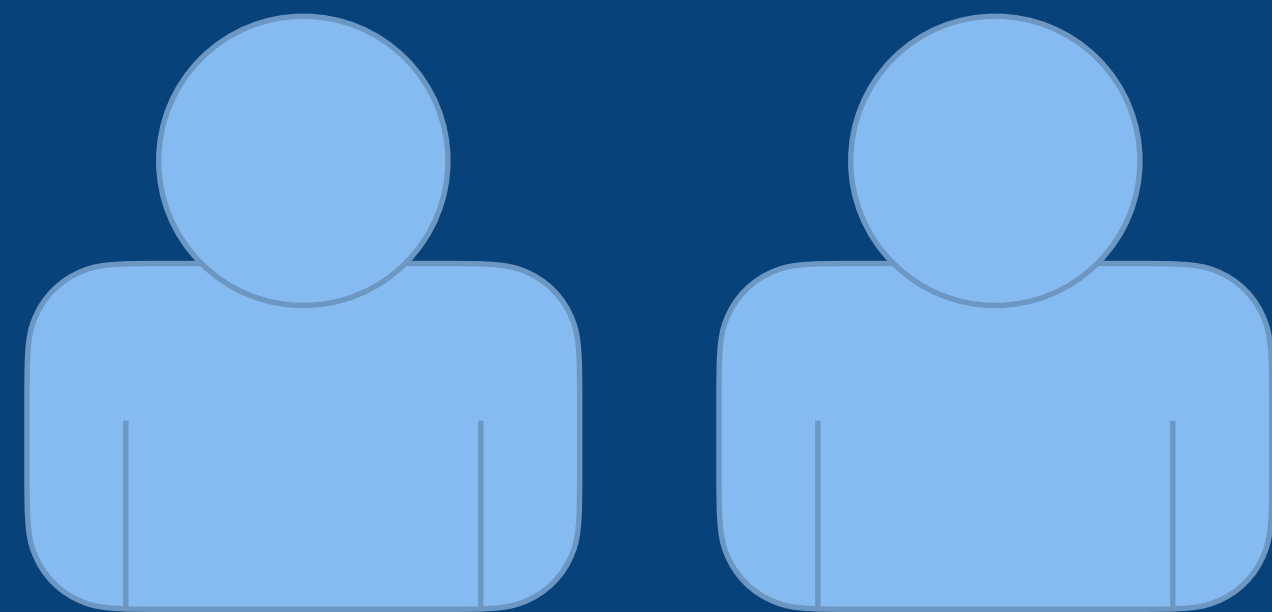
“everybody is responsible for architecture”

!=

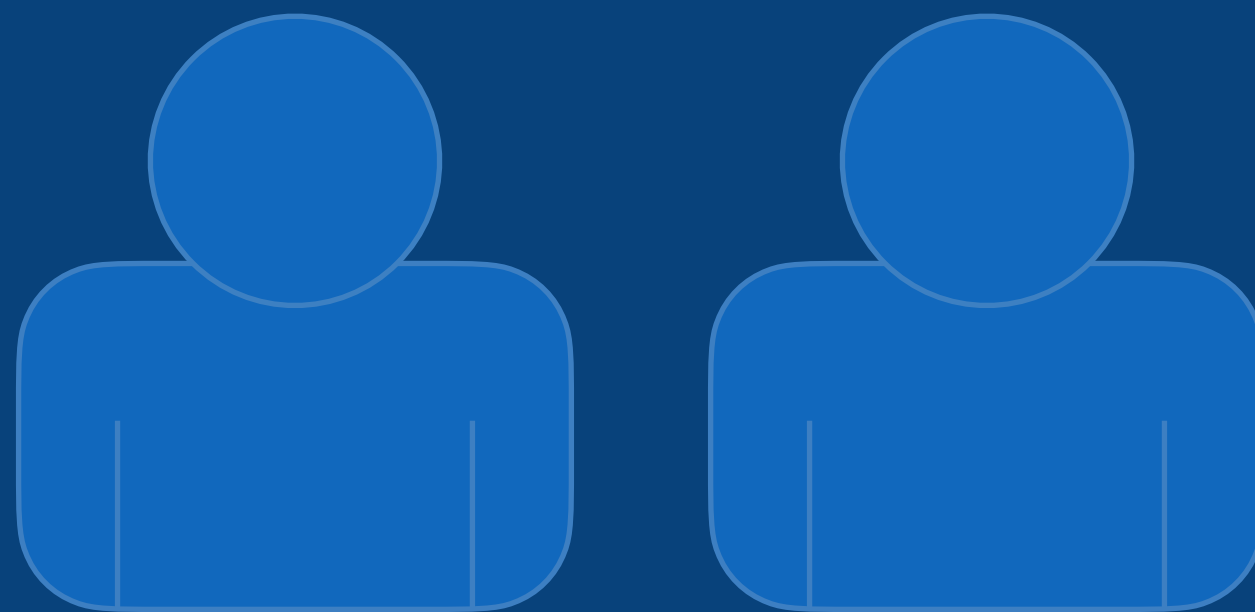
everybody being responsible for architecture



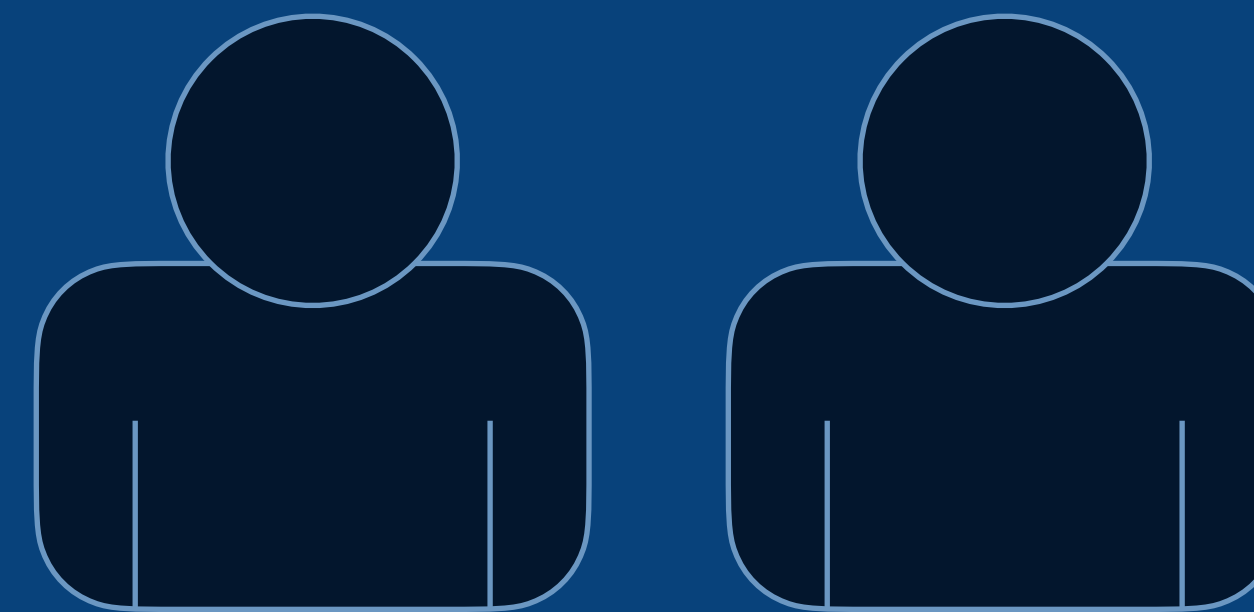
Does everybody have the skills
and motivation to collaborate
on the technical leadership role?



Team A
(original authors)



Team B
(adding code to support business capability 1)



Team C
(adding code to support business capability 2)



Who is providing technical leadership?

[insert name]

(software architects, solution architects,
tech leads, principal engineers, etc)

What about hierarchies of architects
and central architecture groups?

The people performing the technical leadership role have responsibility for the technical success of the project/product

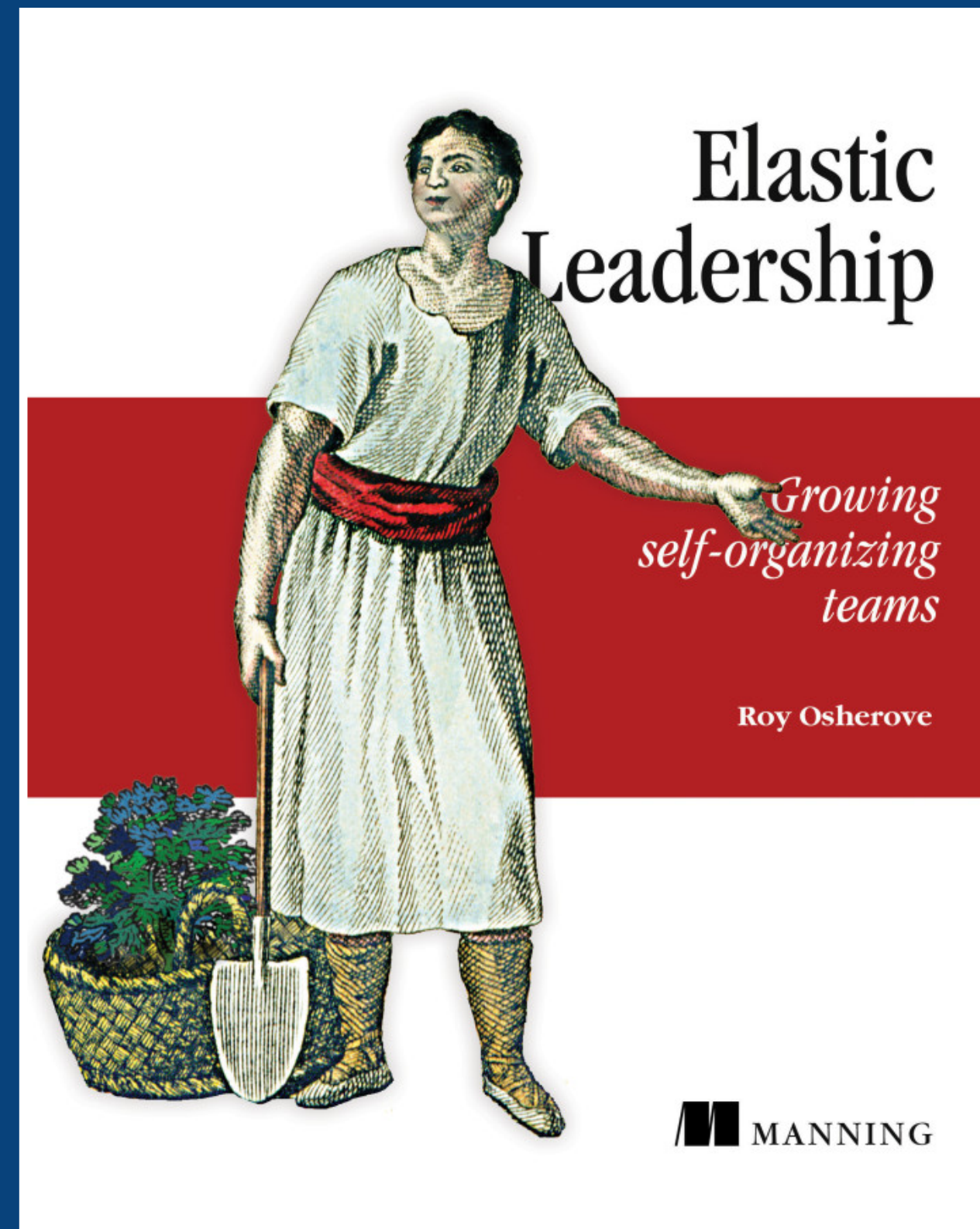
What happens if there's
a lack of technical leadership?

Chaos

Big ball of mud, spaghetti code, inconsistent approaches to solving the same problems, quality attributes are ignored, deployment problems, maintenance issues, etc

Introducing control?
Avoiding chaos?

How much control do you need?



Different types of teams need
different leadership styles



Pair architecting

What should the technical
leadership role include?

Managing architectural drivers

Requirements

(use cases, user stories, features, etc)

Requirement

"a thing that is needed or wanted"

(this includes experiments and hypotheses too)

Quality attributes

(also known as non-functional requirements,
cross-cutting concerns, service-level agreements, etc)

Understand how to **capture, refine**
and **challenge** quality attributes

Software lives in the real world,
and the real world has
constraints

Typical constraints include
time and budget, technology,
people and skills, politics, etc

Principles

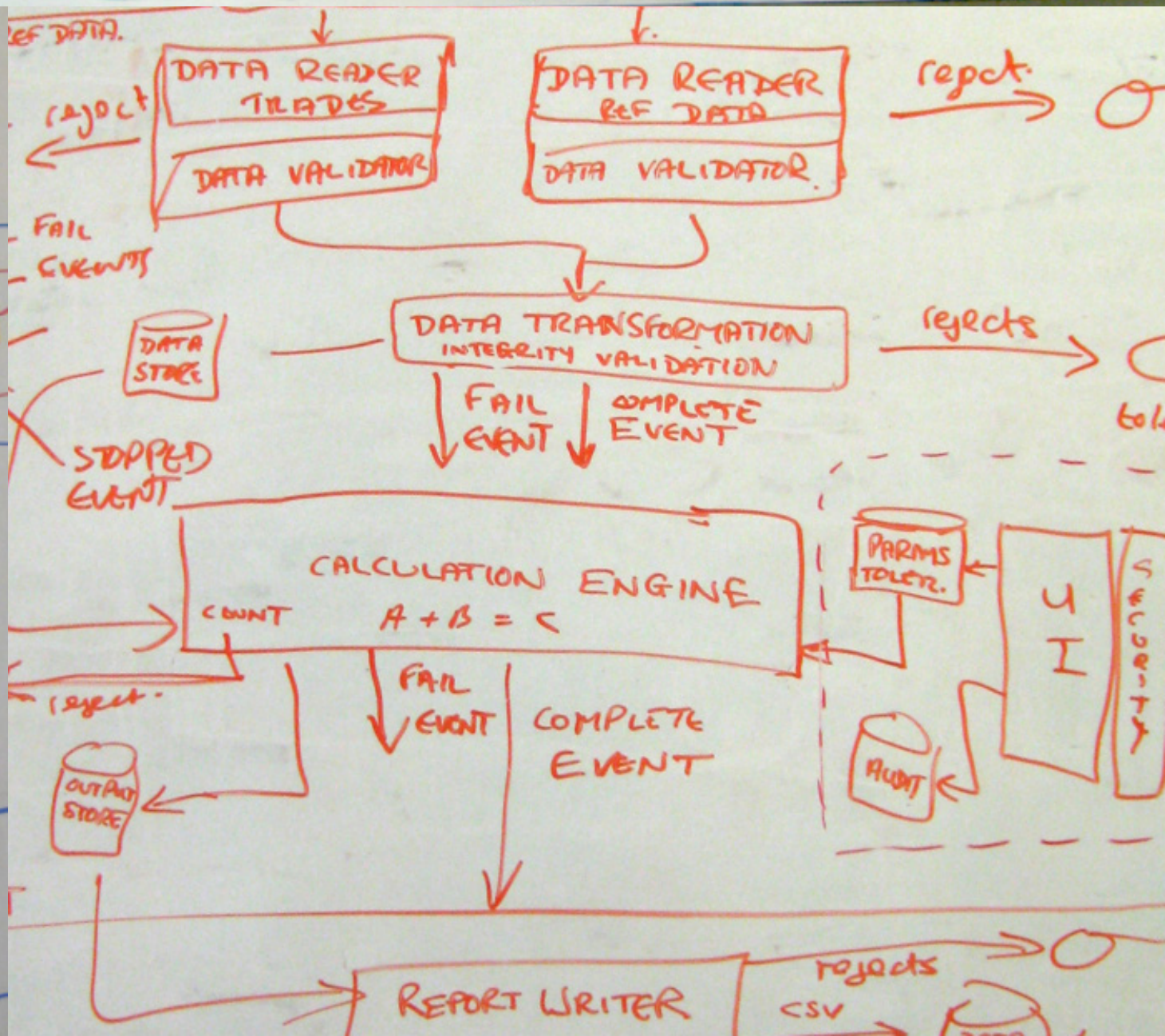
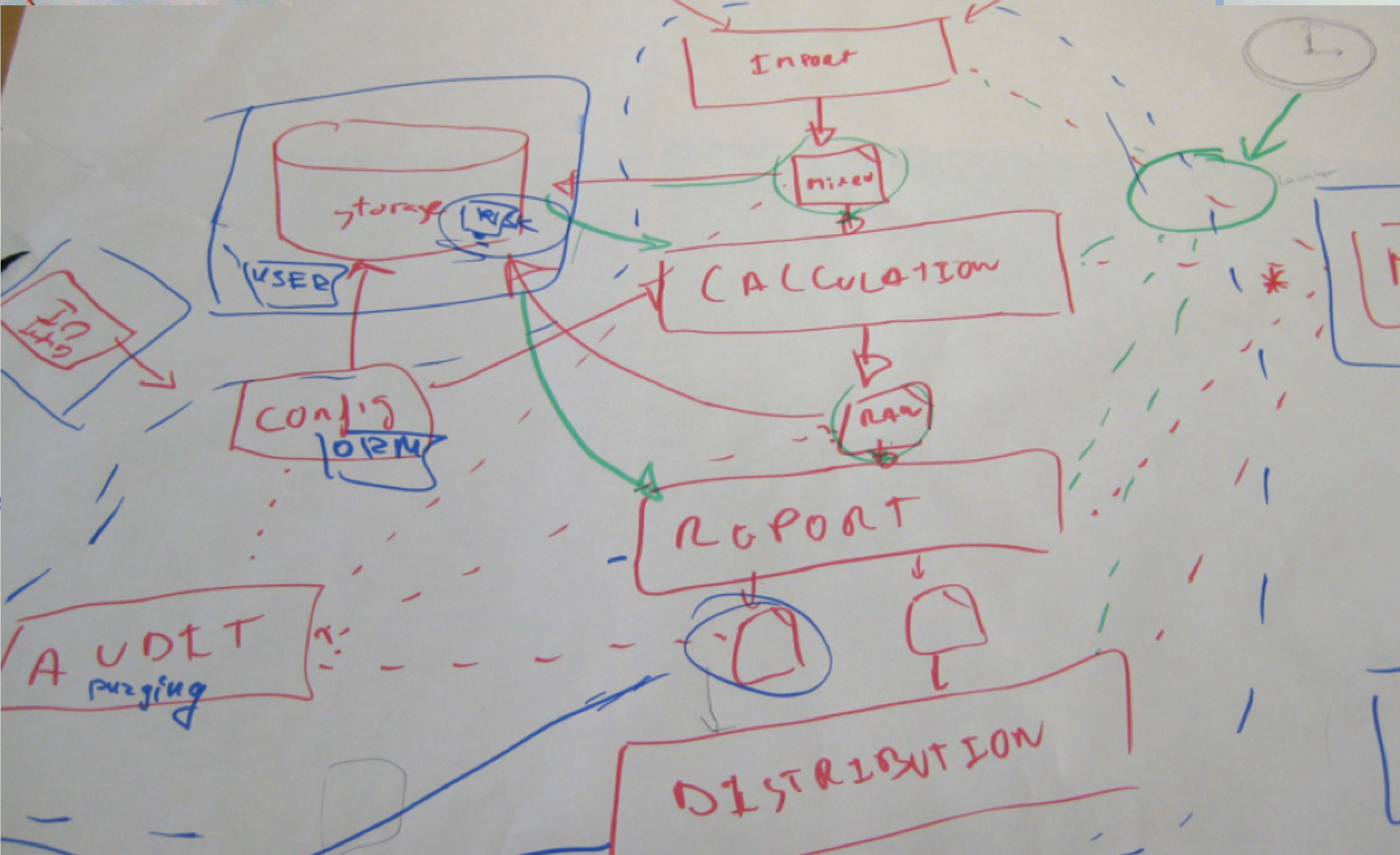
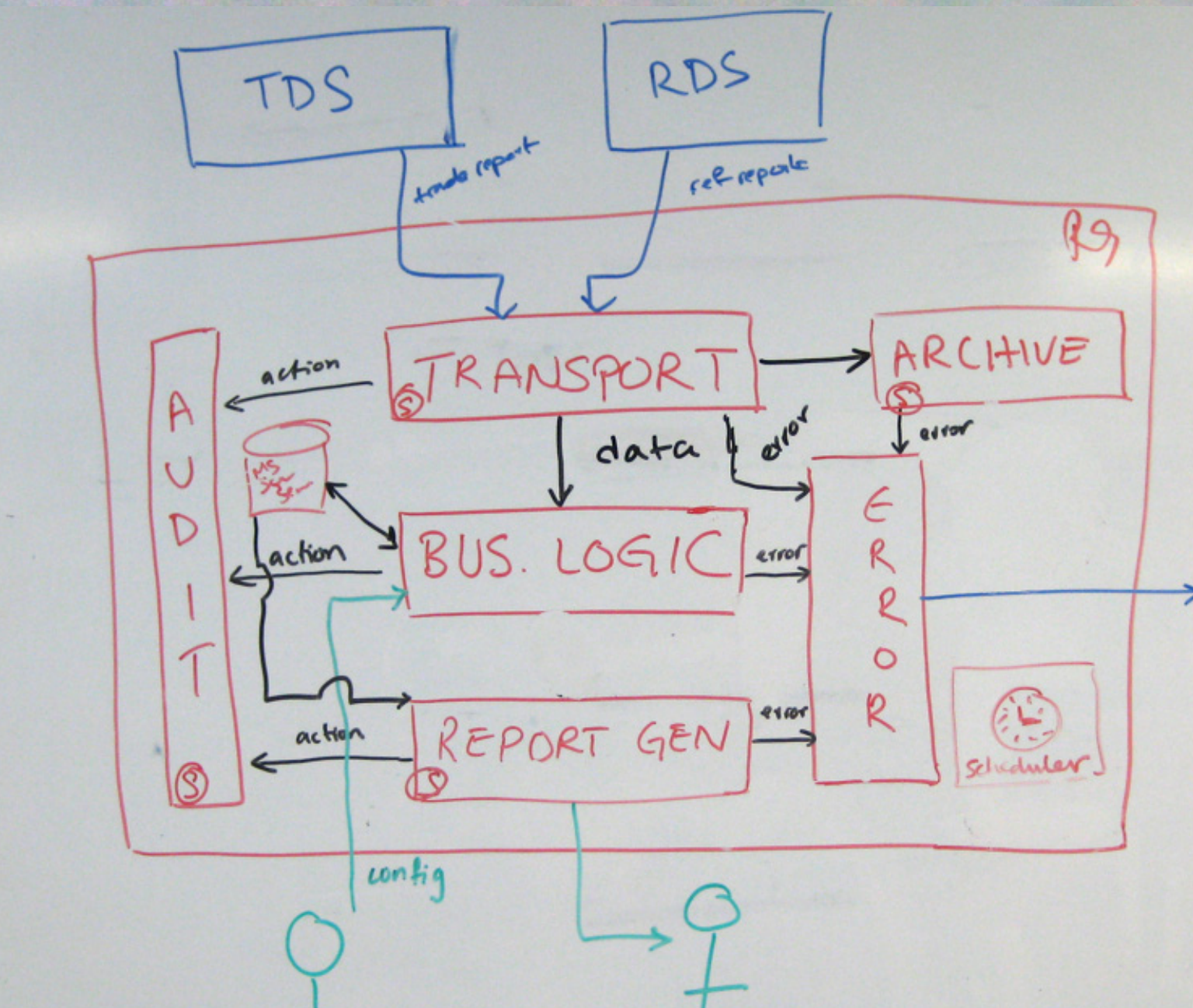
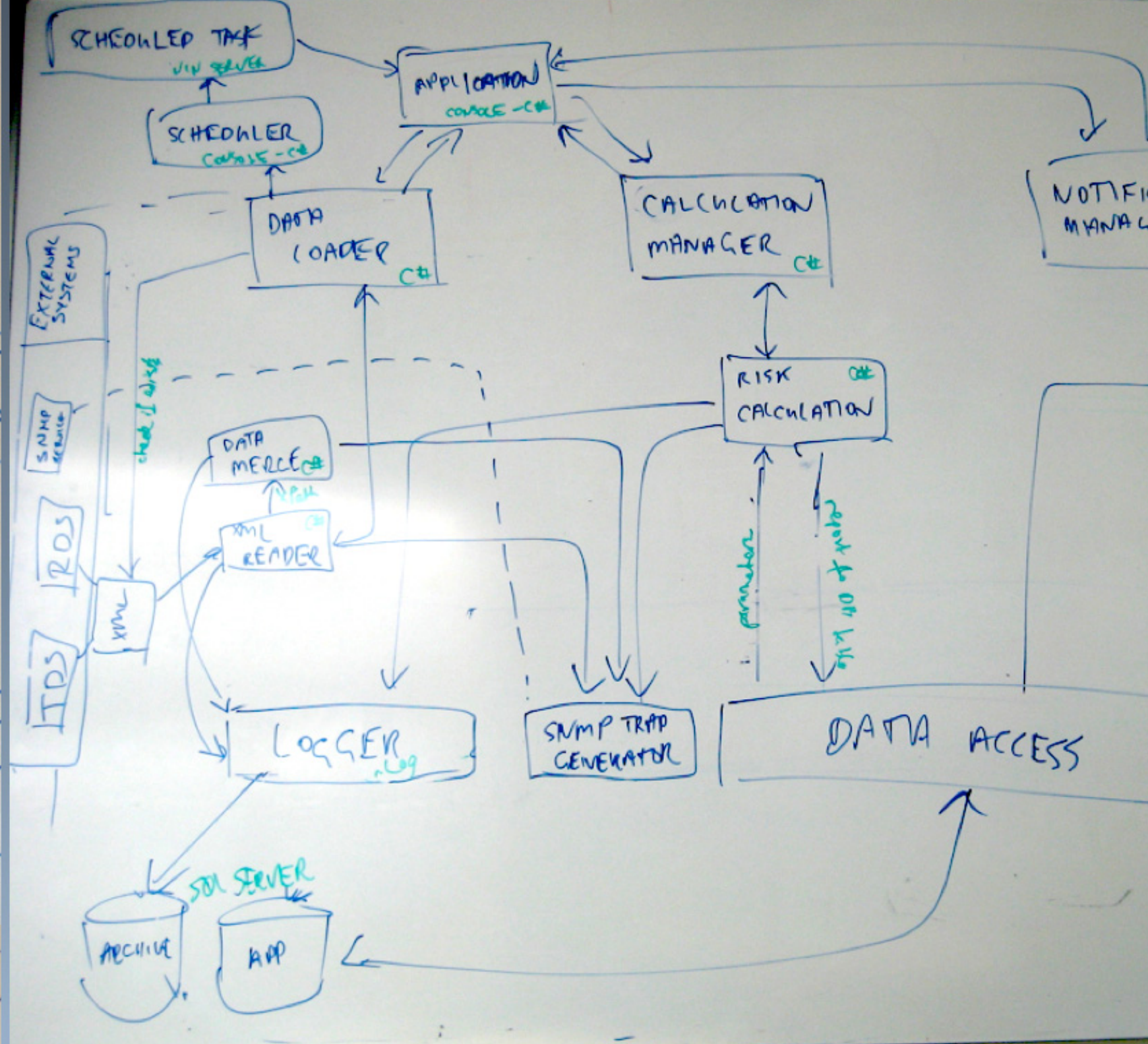
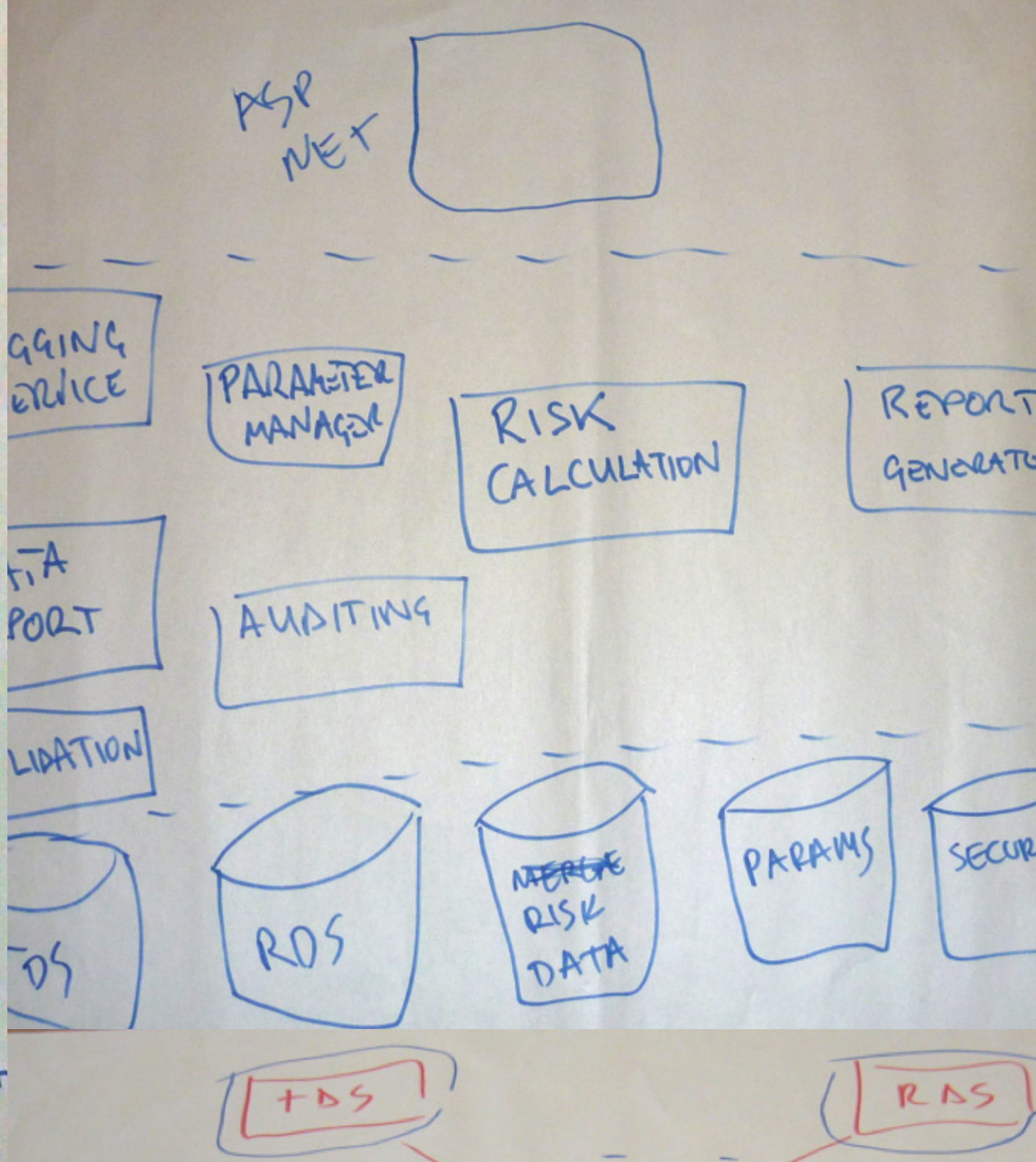
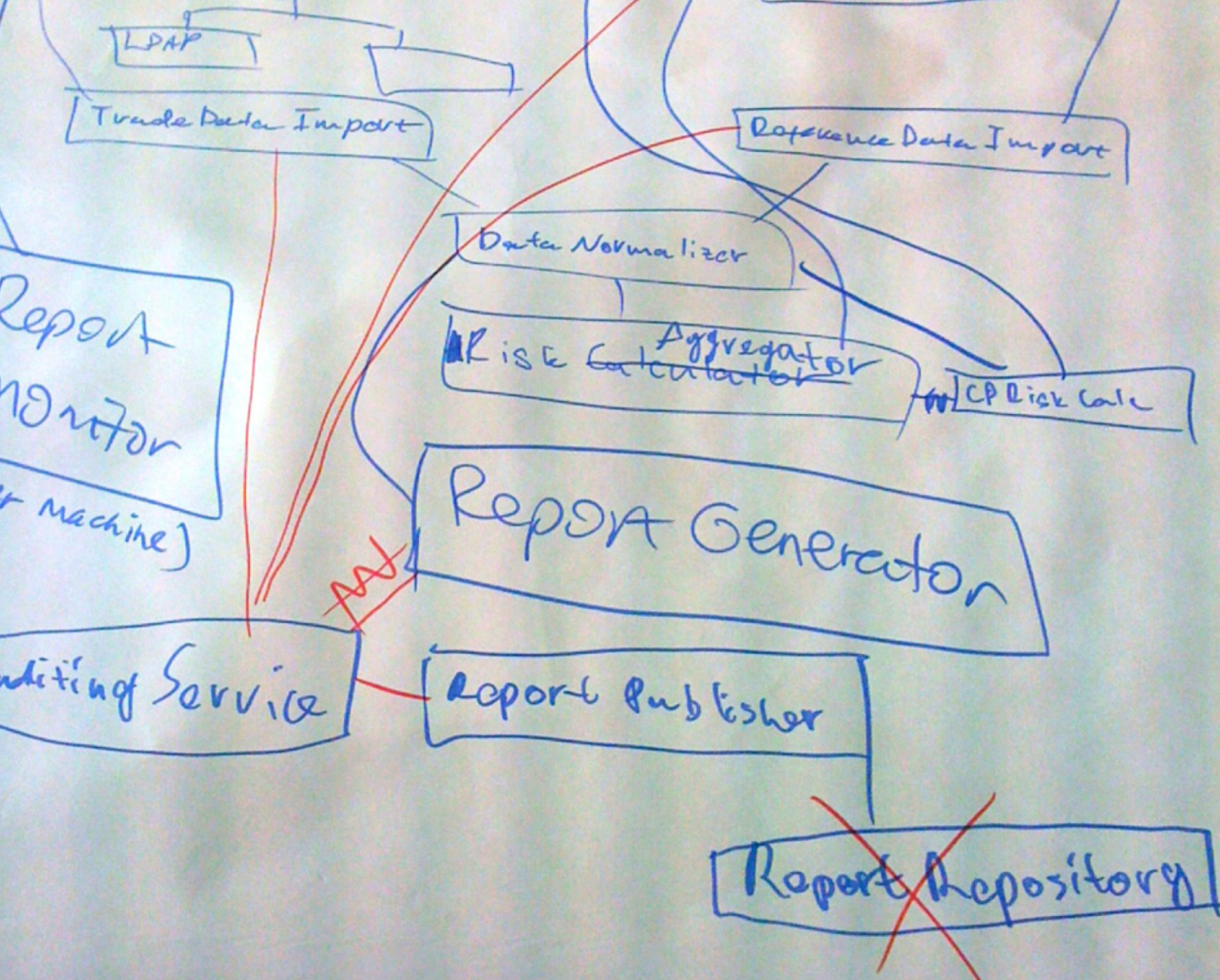
are selected by the team

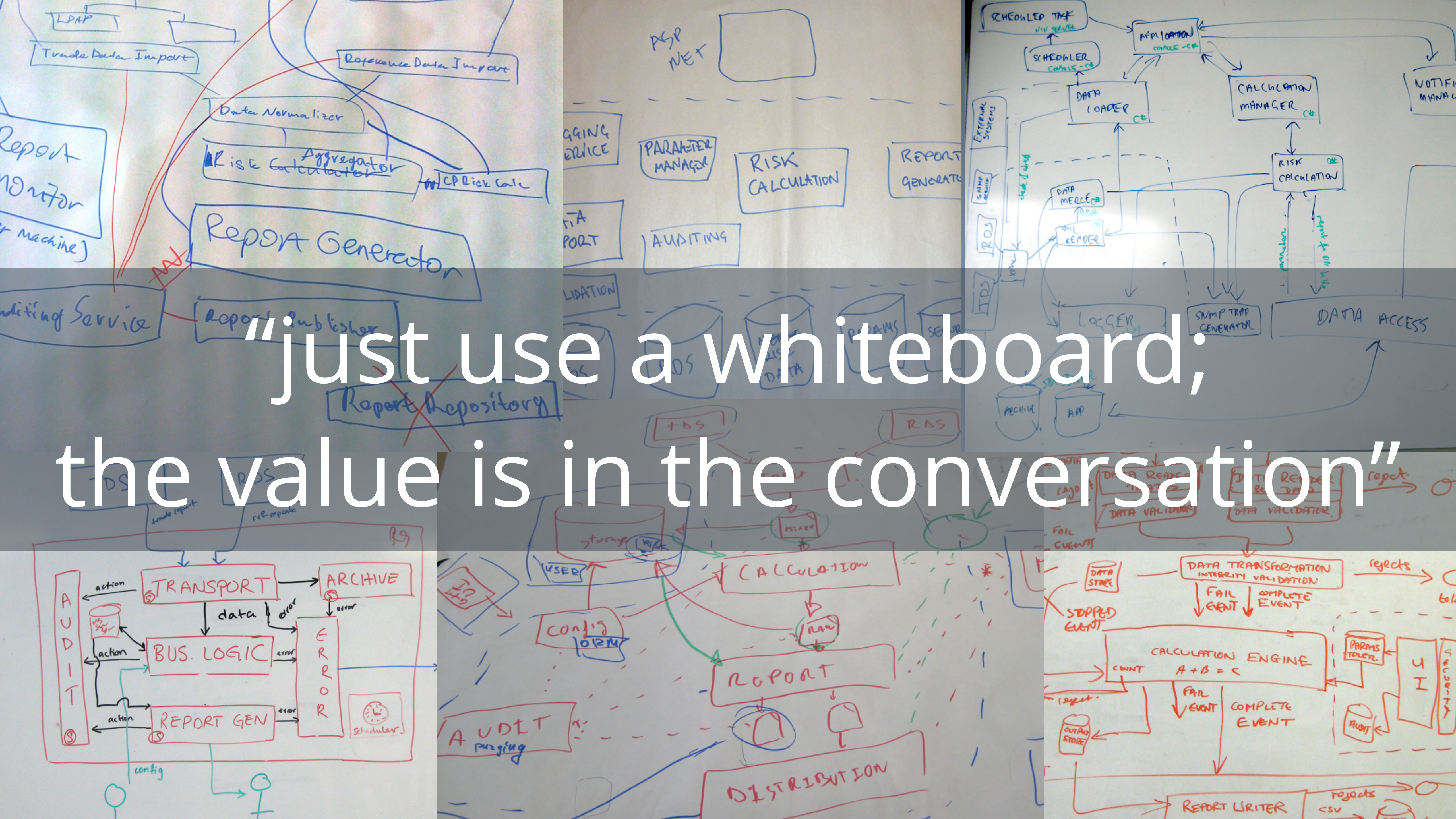
Development principles include
coding conventions, naming
guidelines, testing approaches,
review practices, etc

Architecture and design principles typically relate to modularity or crosscutting concerns

(architectural layering, separation of concerns,
stateless vs stateful, rich vs anaemic domain,
security, error handling, logging, etc)

**Creating and communicating
the team's technical vision**





“just use a whiteboard;
the value is in the conversation”

“the value is in the conversation”
only works if you’re having
the right conversations

No

“What does that arrow mean?”

“Why are some boxes red?”

“Is that a Java application?”

“Is that a monolithic application, or a collection of microservices?”

“How do the users get their reports?”

Yes

“What protocol are your two Java applications using to communicate with each other?”

“Why do you have two separate C# applications instead of one?”

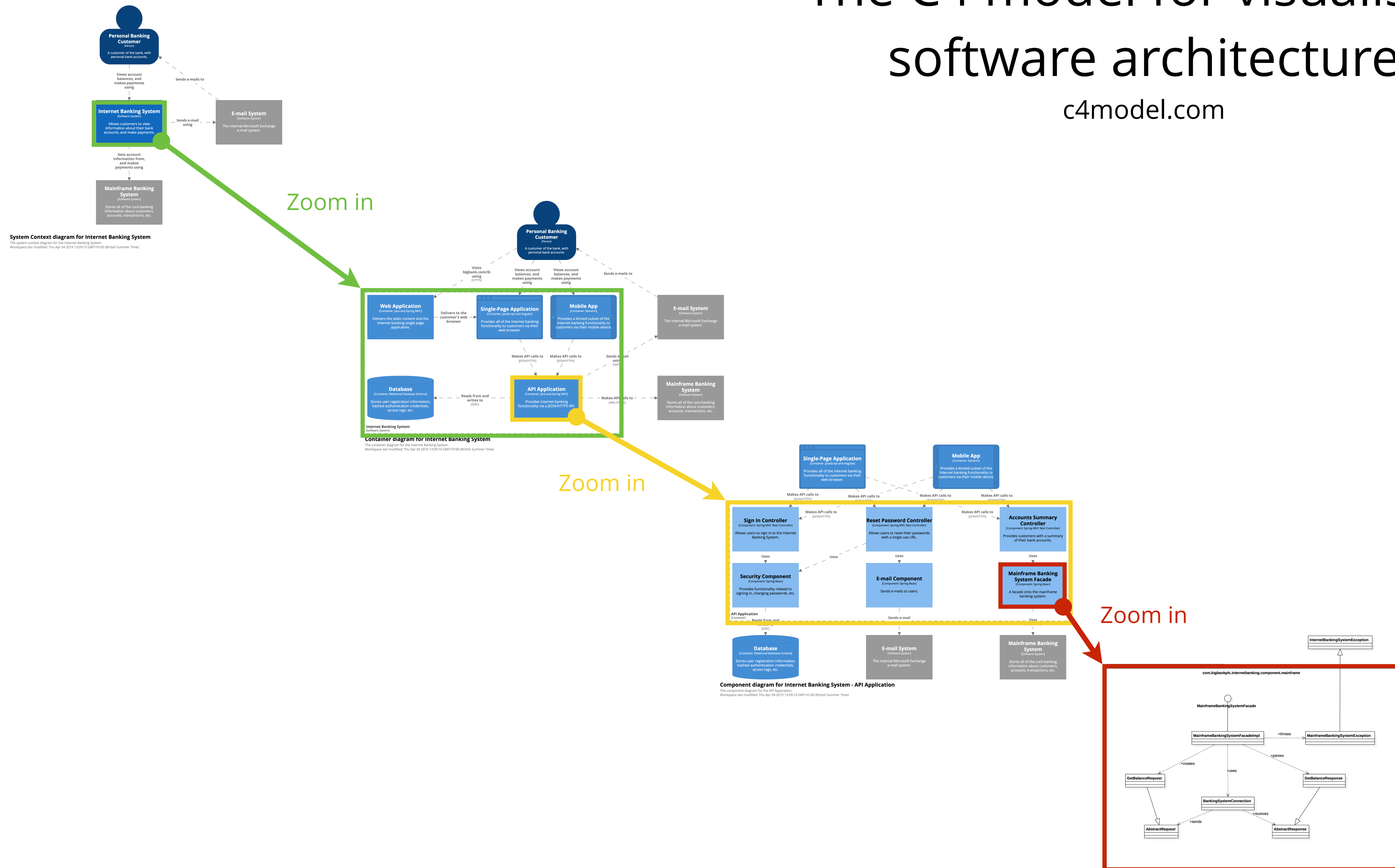
“Why are you using MongoDB?”

“Why are you using MySQL when our standard is Oracle?”

“Should we really build new applications with .NET Framework rather than .NET Core?”

The C4 model for visualising software architecture

c4model.com



Level 1
Context

Level 2
Containers

Level 3
Components

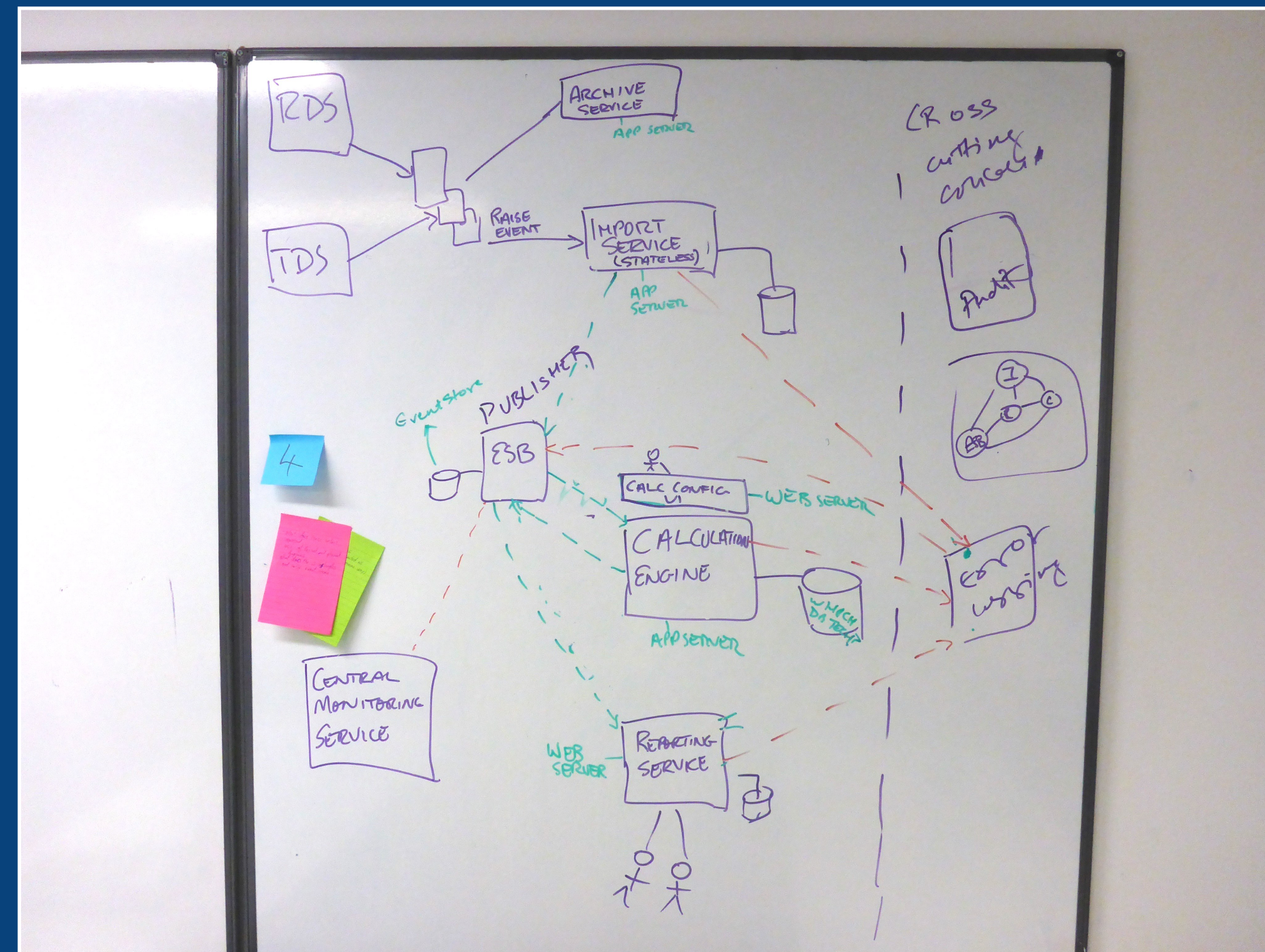
Level 4
Code

Richer diagrams lead to
better communication,
making it easier to scale teams

Richer diagrams lead to
richer design discussions

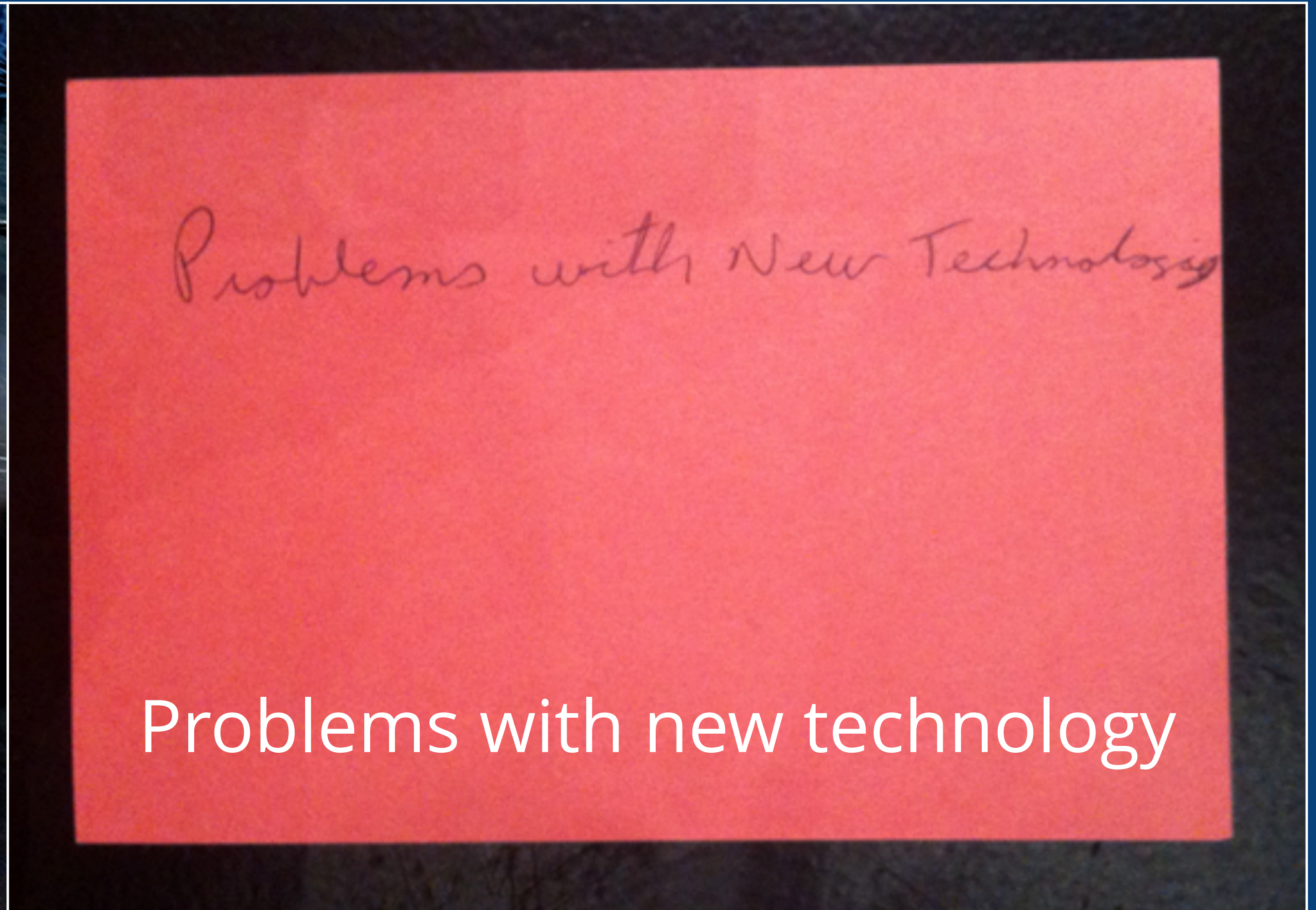
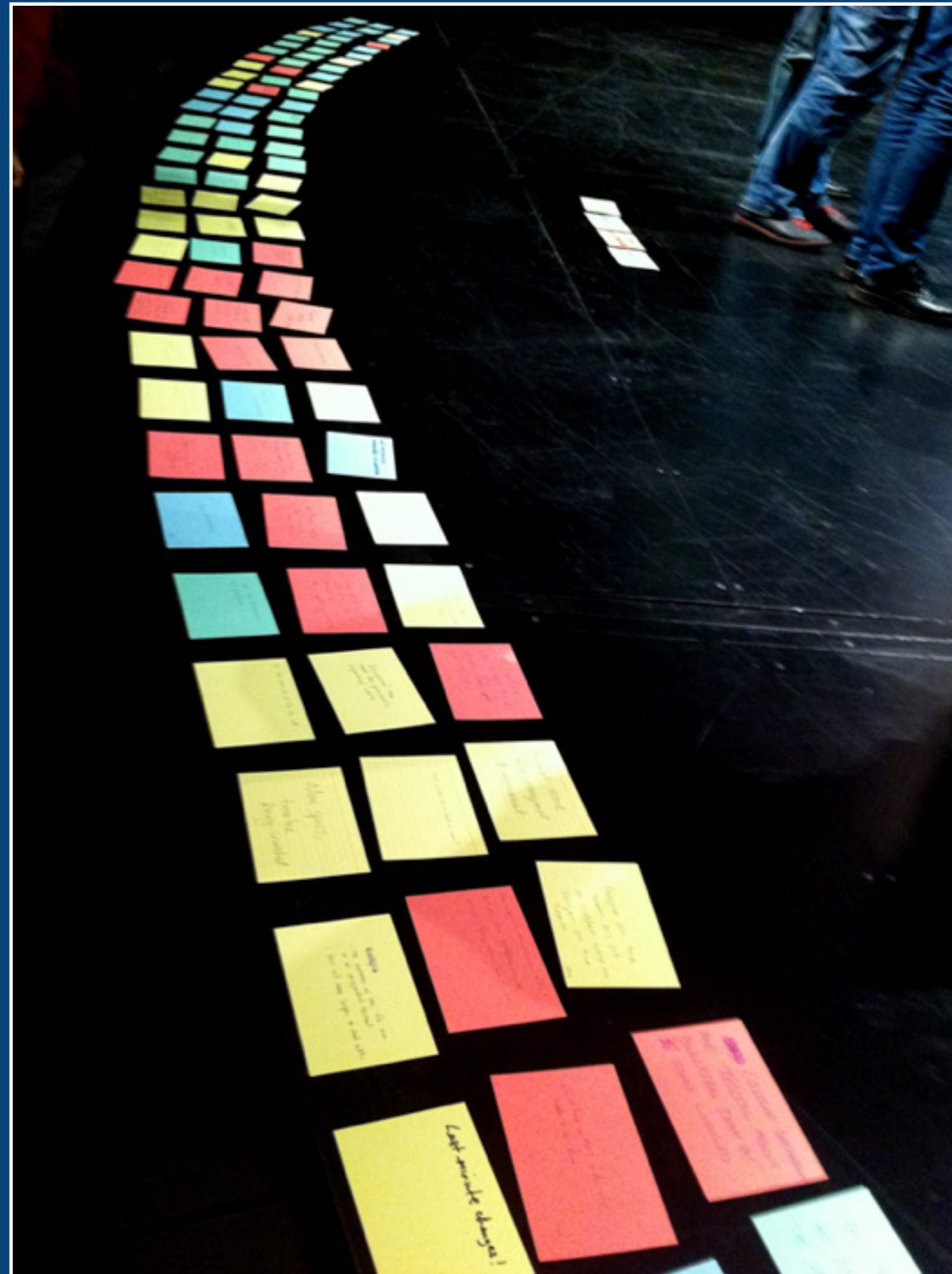
Managing technical risks

1. Is that what we're going to **build**?



2. Is it going to **work**?

Teams need to explicitly
manage technical risk



Problems with new technology

An example timeline from "Beyond Retrospectives"

Linda Rising, GOTO Aarhus 2011

**Identify and mitigate
your highest priority risks**

Probability

Impact

	Low 1	Medium 2	High 3
Low 1	1	2	3
Medium 2	2	4	6
High 3	3	6	9





Threat modelling

(STRIDE, LINDDUN, Attack Trees, etc)

The person/people performing the
technical leadership role
should own the technical risks

**Assuring technical quality
and conformance to the
vision**

Are team members writing code
that helps the quality attributes,
rather than hindering them?

Are team members following the principles we agreed upon?

The technical leadership role
is about the “**big picture**”
and, sometimes, this means
stepping away from the code

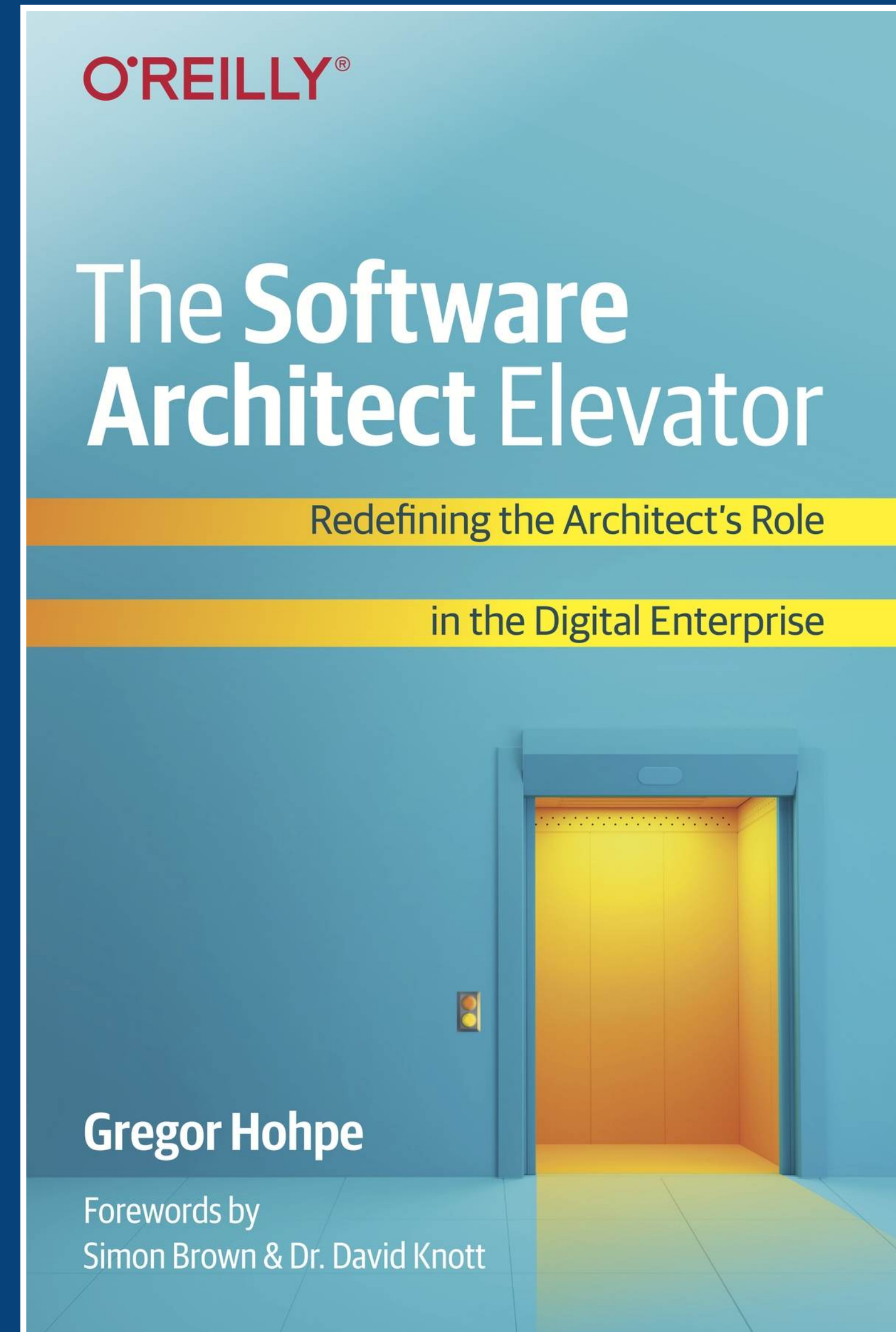
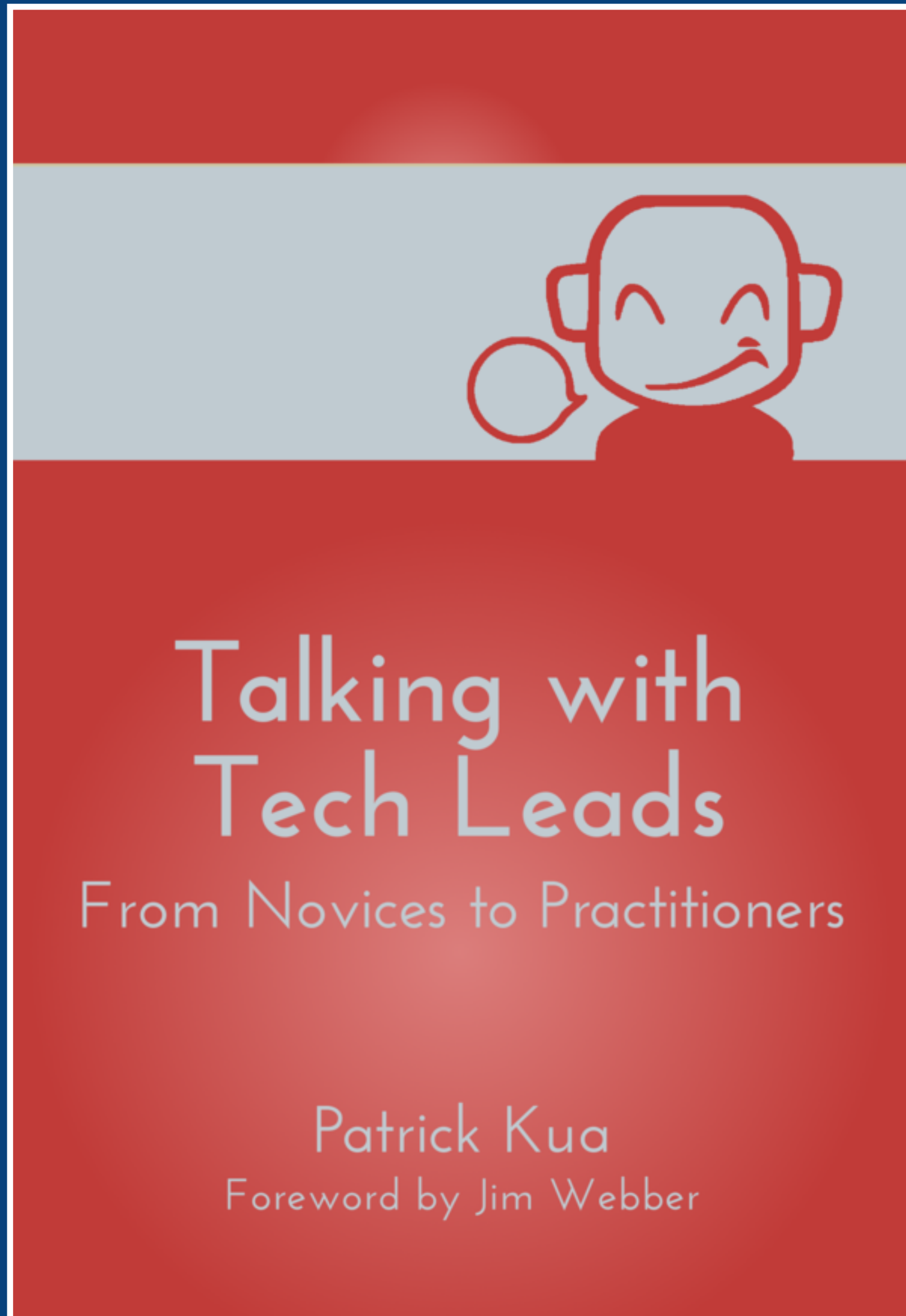
Should software architects
write code?

Production code, prototypes,
frameworks, foundations, code
reviews, experimenting, etc

It's easier to lead a team of
developers if you are a developer

“Soft” skills

(leadership, communication, presentation, influencing, negotiation, collaboration, coaching and mentoring, motivation, facilitation, political, etc)



Reinventing the wheel
... again

From “software architects” to...

“Full-stack engineers”

(technical specialists are an anti-pattern, and full-stack engineers lets teams deliver features faster)

“DevOps”

(but developers and infrastructure/operations
are not collaborating well enough)

“DevSecOps”

(cloud is hard, so perhaps you want to also talk to us security experts)

“DevTestSecOps”

(why is the test team being excluded from the conversation?)

“DevTestDocSecOps”

(why does nobody write documentation these days?)

DevEntArchSecTestDocProductBizOps

(hey, don't forget about us too!)

Collaborate
or fail

Thank you!



Simon Brown

 @simonbrown