

BLENDING PRODUCT THINKING WITH ARCHITECTURE



Joel Tosi

SLIDES—<https://www.dojoandco.com/speaking>

@joeltosi@mastodon.social

Joel.Tosi@dojoandco.com

ABOUT ME



Last 10 years - easier for people to
learn & do great work

Product - Tech - Process

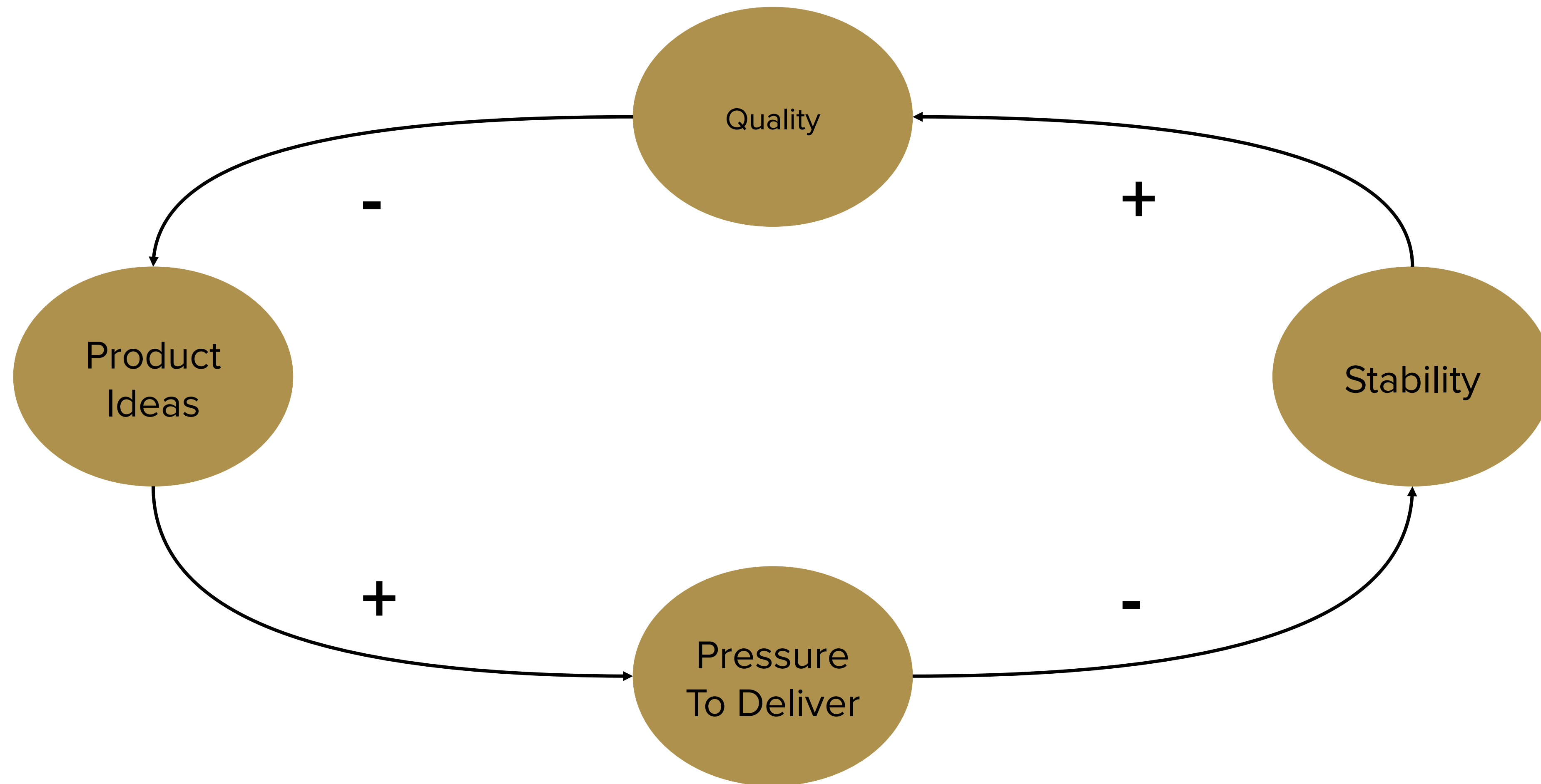
Wrote Books

Past lives - Architect at Red Hat;
Trading Platforms; etc

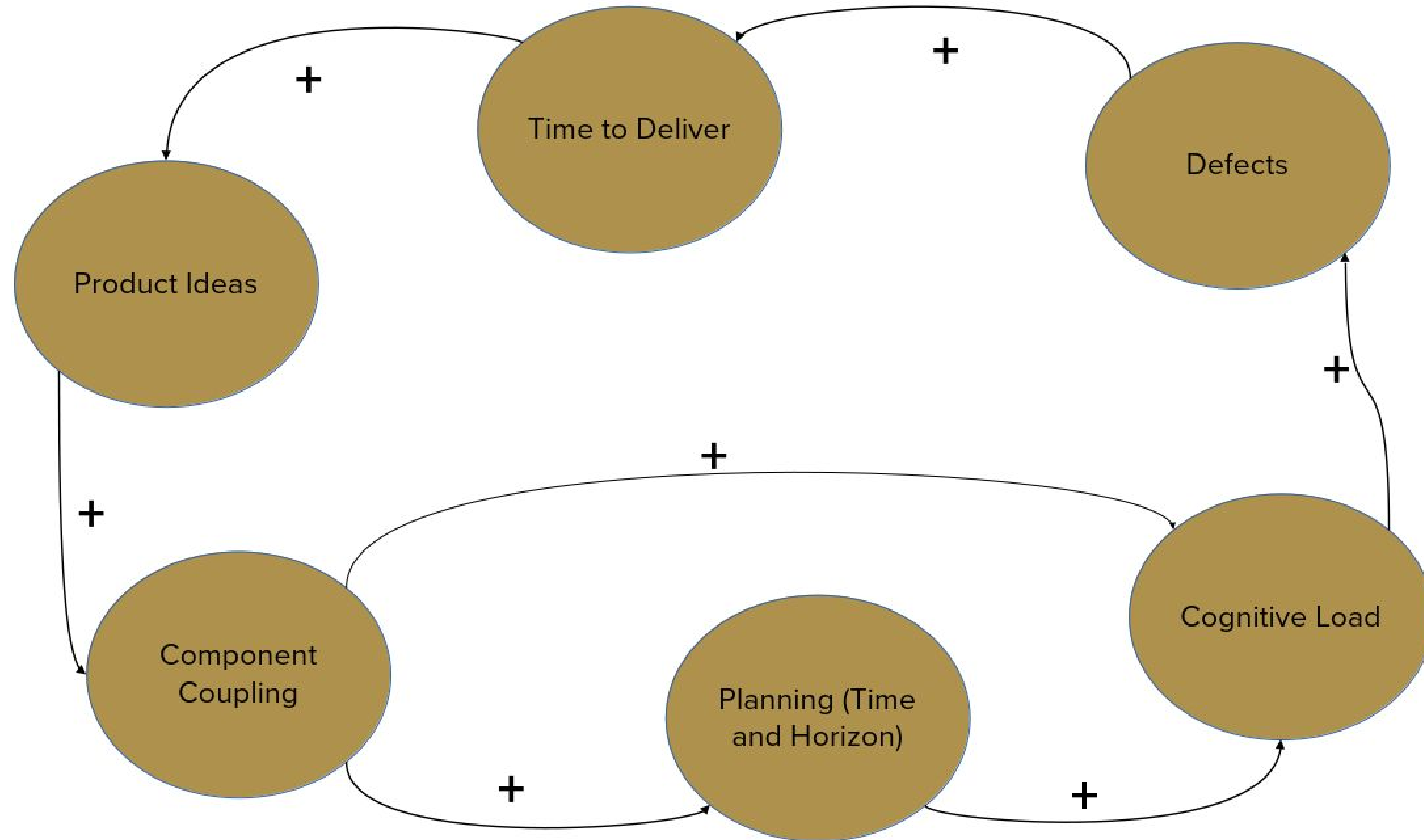


Hit me up if you want to chat

WHY DOES IT TAKE LONGER TO GET STUFF DONE?



A VICIOUS CYCLE





EXPECTED OUTCOMES



Simple Techniques You Can Use Now

- Strategic Domain Driven Design
- C4 Modeling
- Architecture and Planning

Build Better Products

Reduce Planning Nonsense

Take Control of Your Work

["Augsburg, Germany. 2014. Same idea, different outcome."](#) by [Boris Thaser](#) is licensed under [CC BY 2.0](#).

BEHIND EVERY ARCHITECTURE DECISION
IS A PRODUCT ASSUMPTION

HOW CAN WE APPROPRIATELY EVOLVE
THE ARCHITECTURE
AS THE PRODUCT EVOLVES

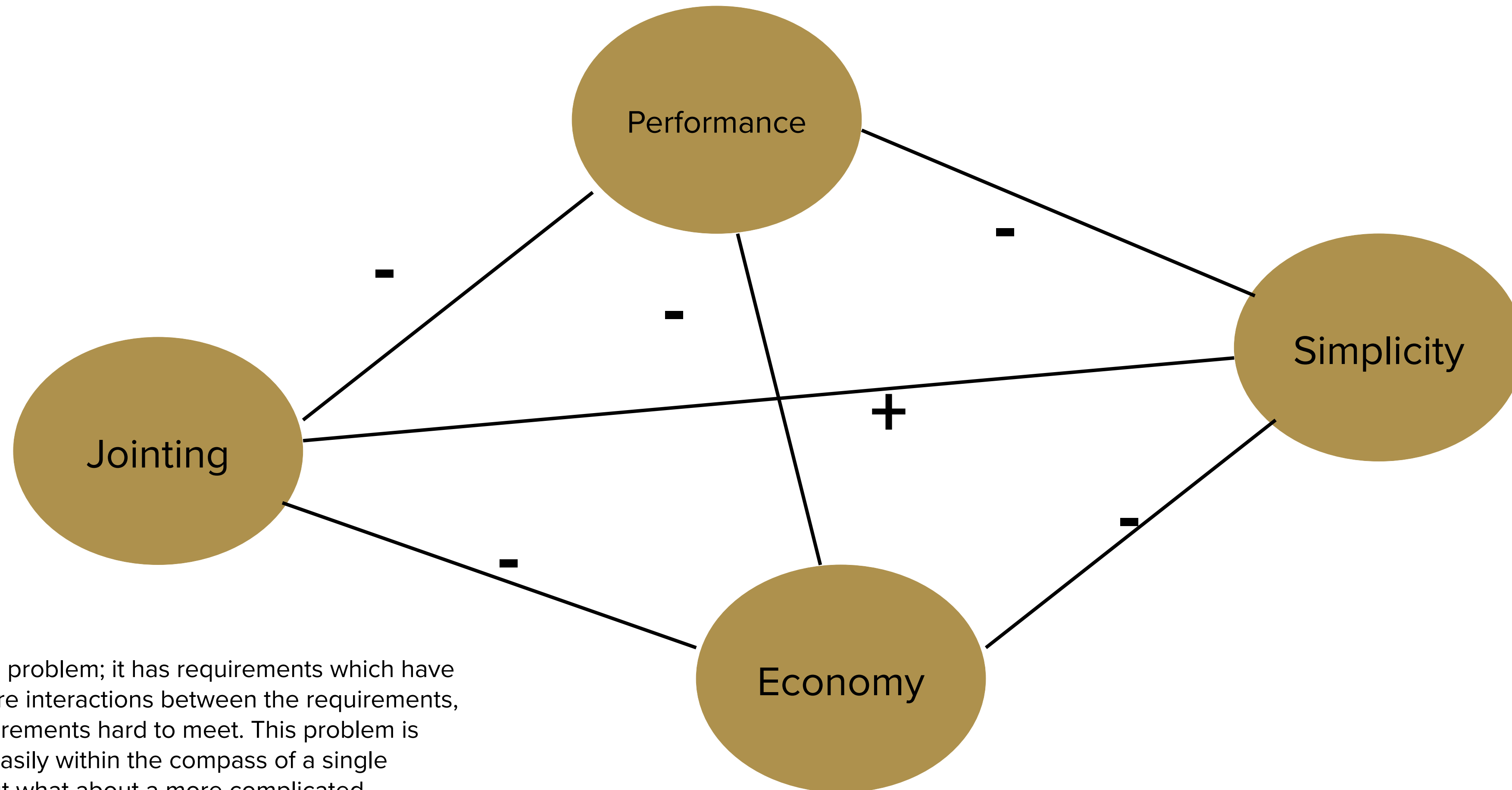
WHAT IS ARCHITECTURE?

CONTEXT AND THE WIKI



"Ward Cunningham" by Joi is licensed under CC BY 2.0

WHAT IS ARCHITECTURE?



This is a typical design problem; it has requirements which have to be met; and there are interactions between the requirements, which makes the requirements hard to meet. This problem is easy to solve. It falls easily within the compass of a single (person)'s intuition. But what about a more complicated problem?

© Notes on the Synthesis of Form – Christopher Alexander

**ARCHITECTURE IS ABOUT FIT
WITHIN CONTEXT**

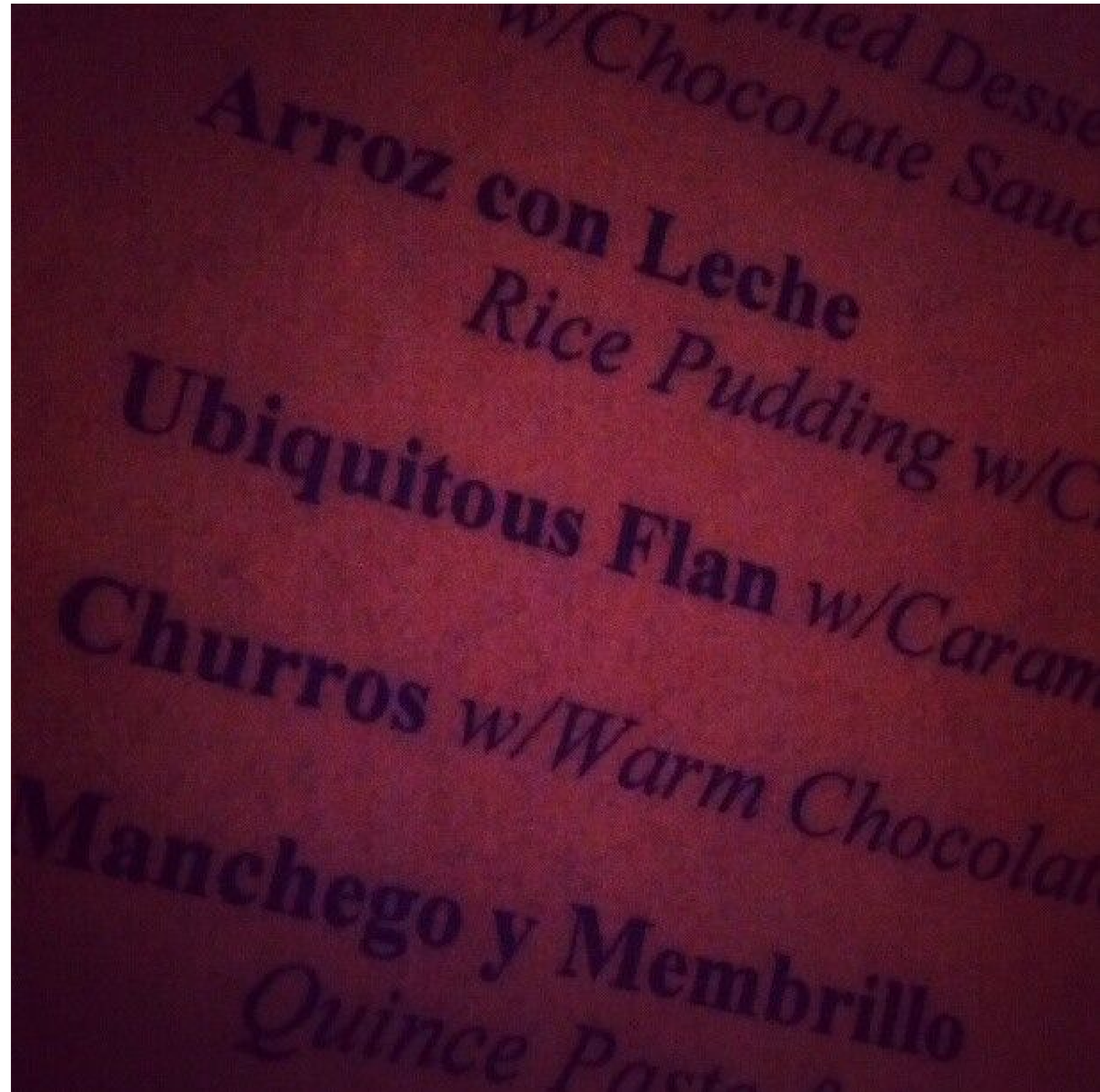
USING PRODUCT CONTEXT
TO GROUND ARCHITECTURE

i.e.

INTENTIONAL VS ACCIDENTAL
ARCHITECTURE

**PROBLEM: TEAMS ARE STEPPING
ON EACH OTHERS TOES**

WHAT IS STRATEGIC DDD? WHY MIGHT WE CONSIDER IT?



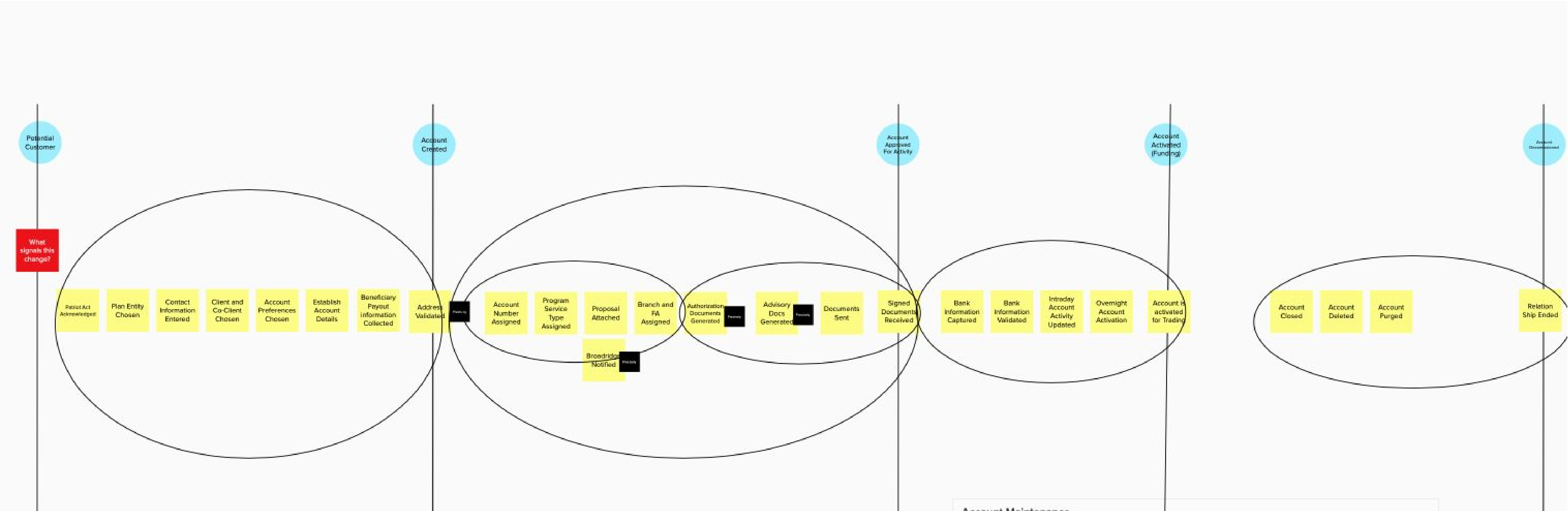
We are trying to model domains / teams closely to product intent

Strategic DDD is a technique for designing systems & boundaries

It is a model that helps us reason and keep product intent close

"Ubiquitous Flan" by lisa.williams is licensed under CC BY 2.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by/2.0/?ref=openverse>.

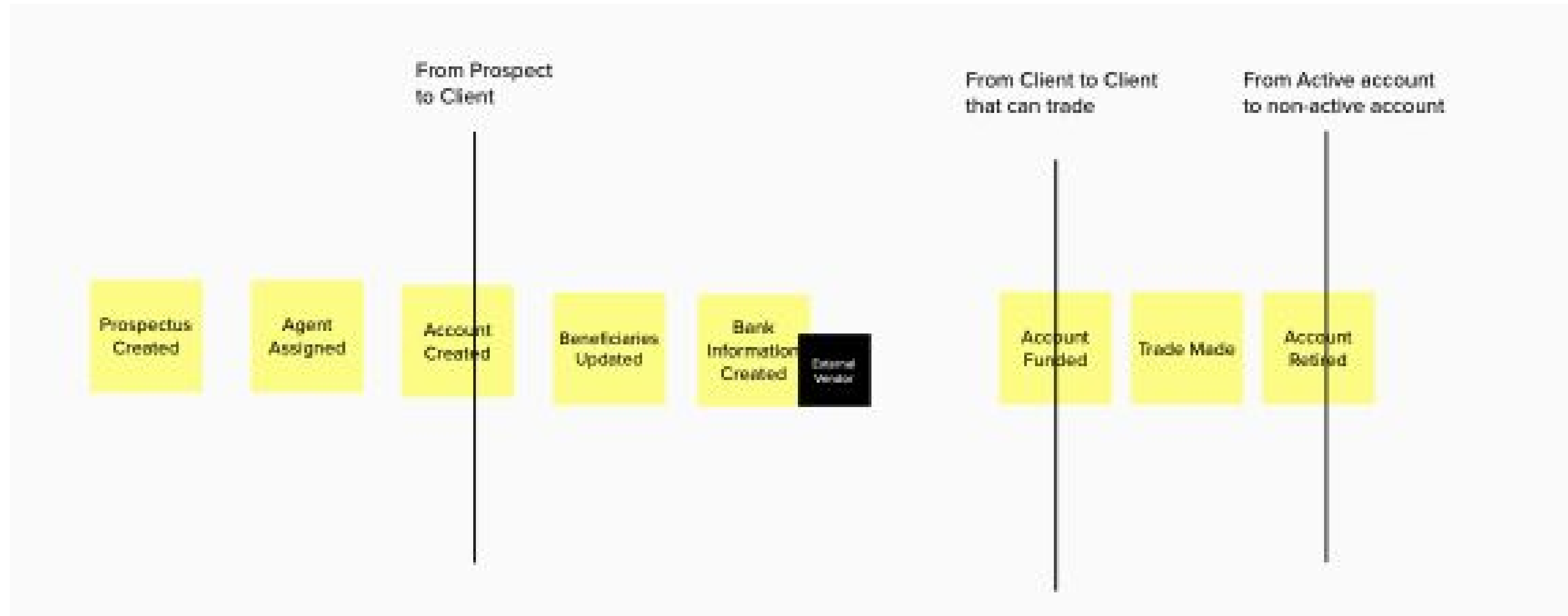
AN EXAMPLE



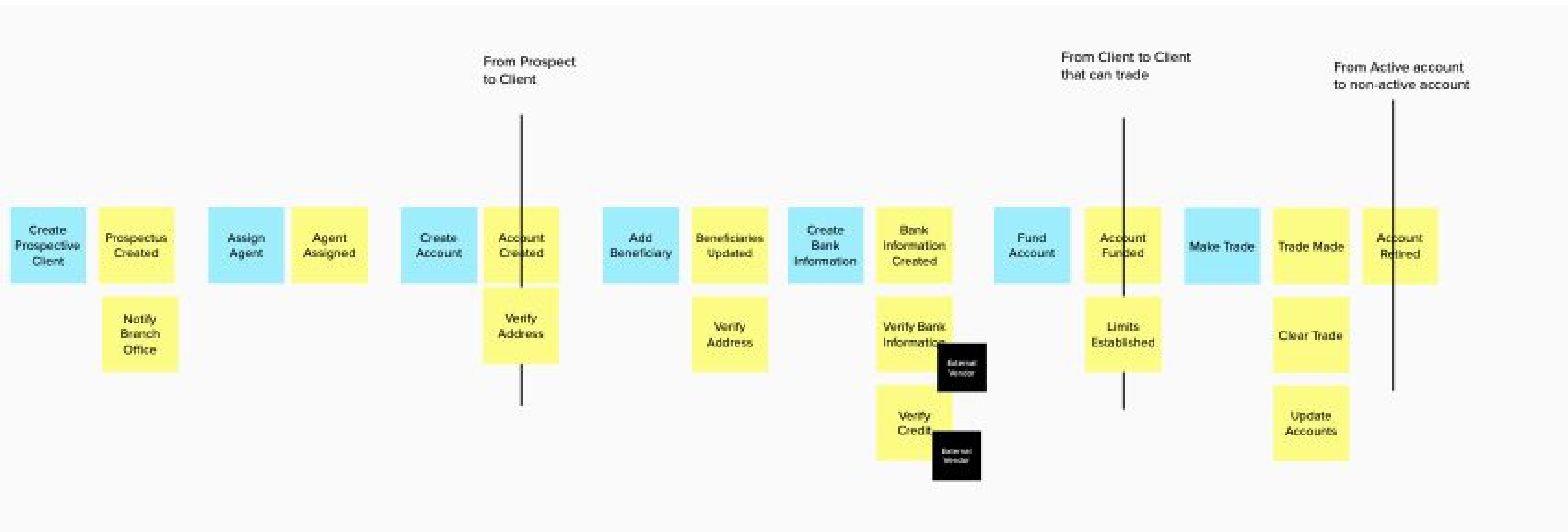
HOW DID WE GET THERE?



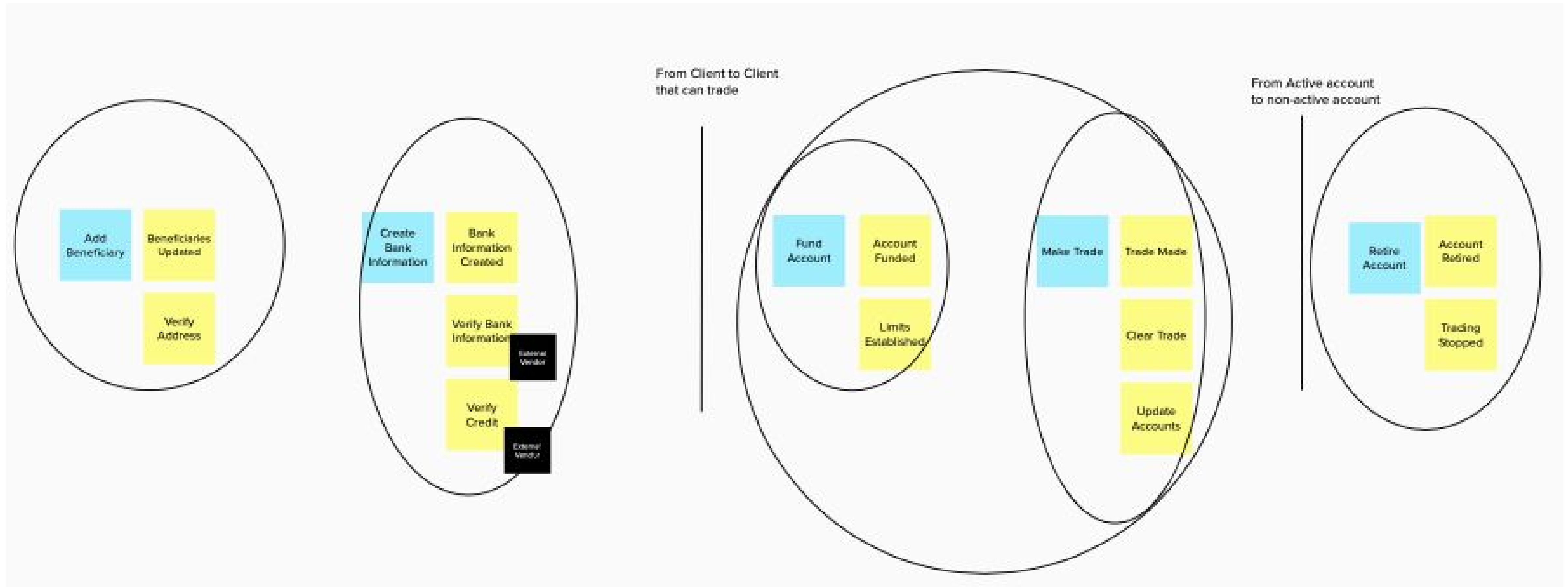
HOW DID WE GET THERE?



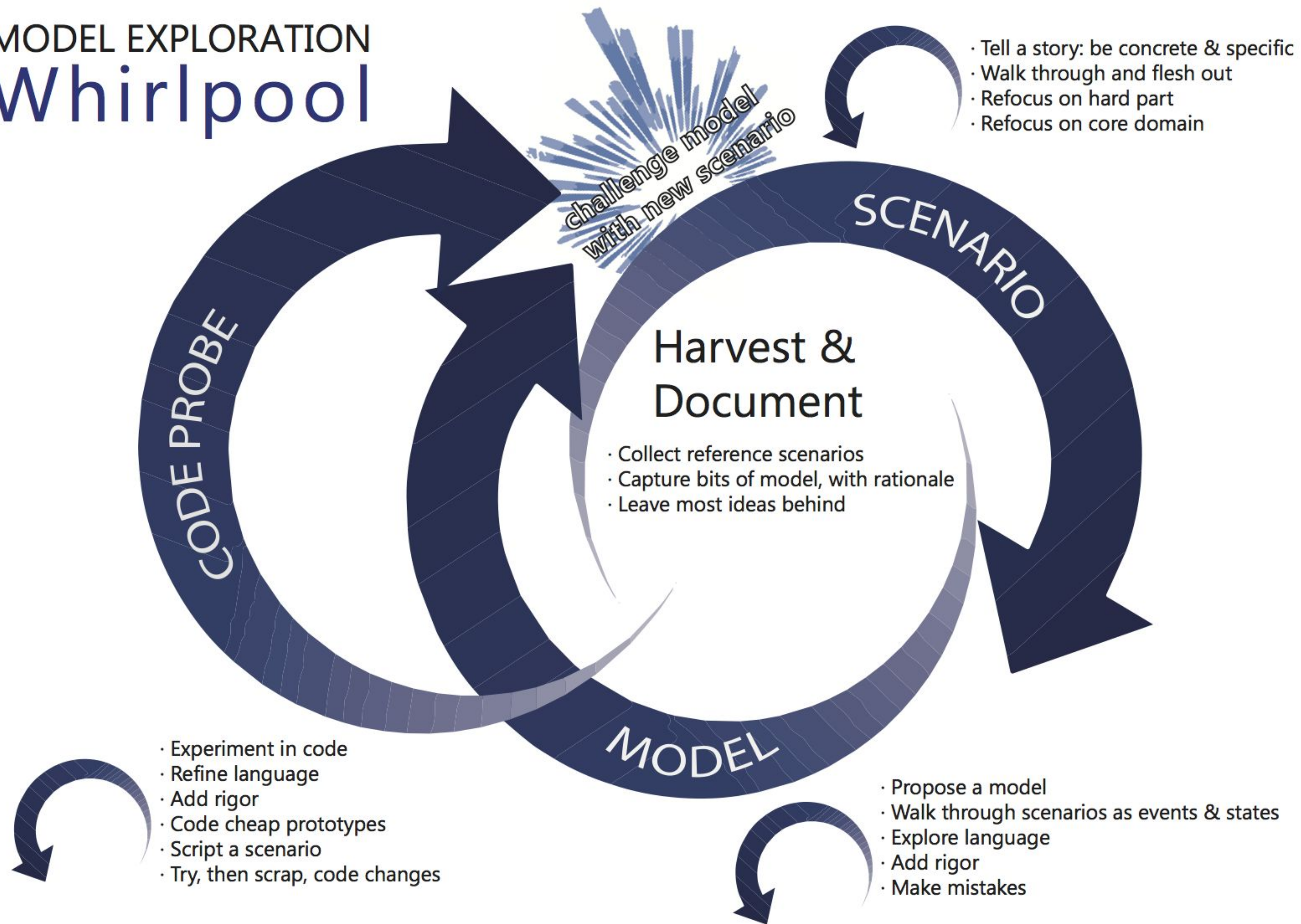
HOW DID WE GET THERE?



HOW DID WE GET THERE?



MODEL EXPLORATION Whirlpool



GETTING STARTED

Event Storming - capture events from a customers perspective
(Story Mapping may help)

Which are significant?

When does context change?

Start to identify boundaries - your first best guess (code to learn)

Teams respond to events or own publishing events

Look out for -

Inventory of systems, reports, APIs

Language that is not context aware

You will need to rethink data



"Advent 2011 Star Wars Recap: Figs" by Bill Ward's Brickpile is licensed under CC BY 2.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by/2.0/?ref=openverse>.

UBIQUITOUS LANGUAGE AS A SIGNAL



Listen for words that start to have extra meaning

These could be contexts trying to break apart (and rightfully so)

"Why actuated signals are bad for pedestrians" by Eric Fischer is licensed under CC BY 2.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by/2.0/?ref=openverse>.

USE STRATEGIC DDD TO REDUCE
COGNITIVE LOAD, VISUALIZE
RELATIONSHIPS, AND MOVE TOWARDS
REDUCED COUPLING

**PROBLEM: SYSTEMS ARE LARGE -
WHERE DO WE FIT / CONSUME / PRODUCE
AND FOR WHO?**

HOW DOES MY TEAM CONTRIBUTE?

WHAT IS C4 MODELING? WHY MIGHT WE CONSIDER IT?

Way of representing systems and architecture

Show context at various levels; across and within systems
‘The right story at the right level’

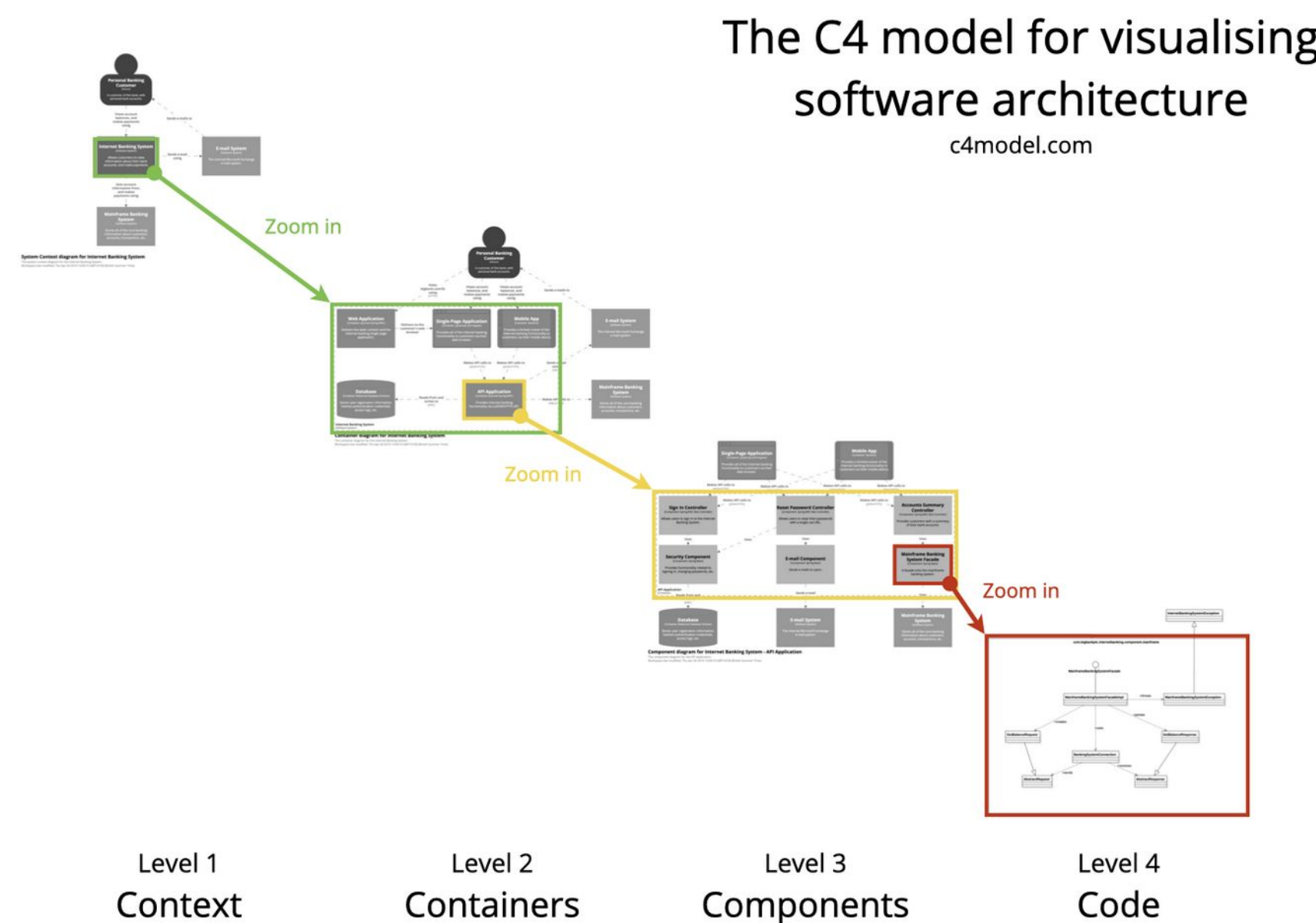
Pinch / Zoom perspective

See how / where we fit; Boundaries

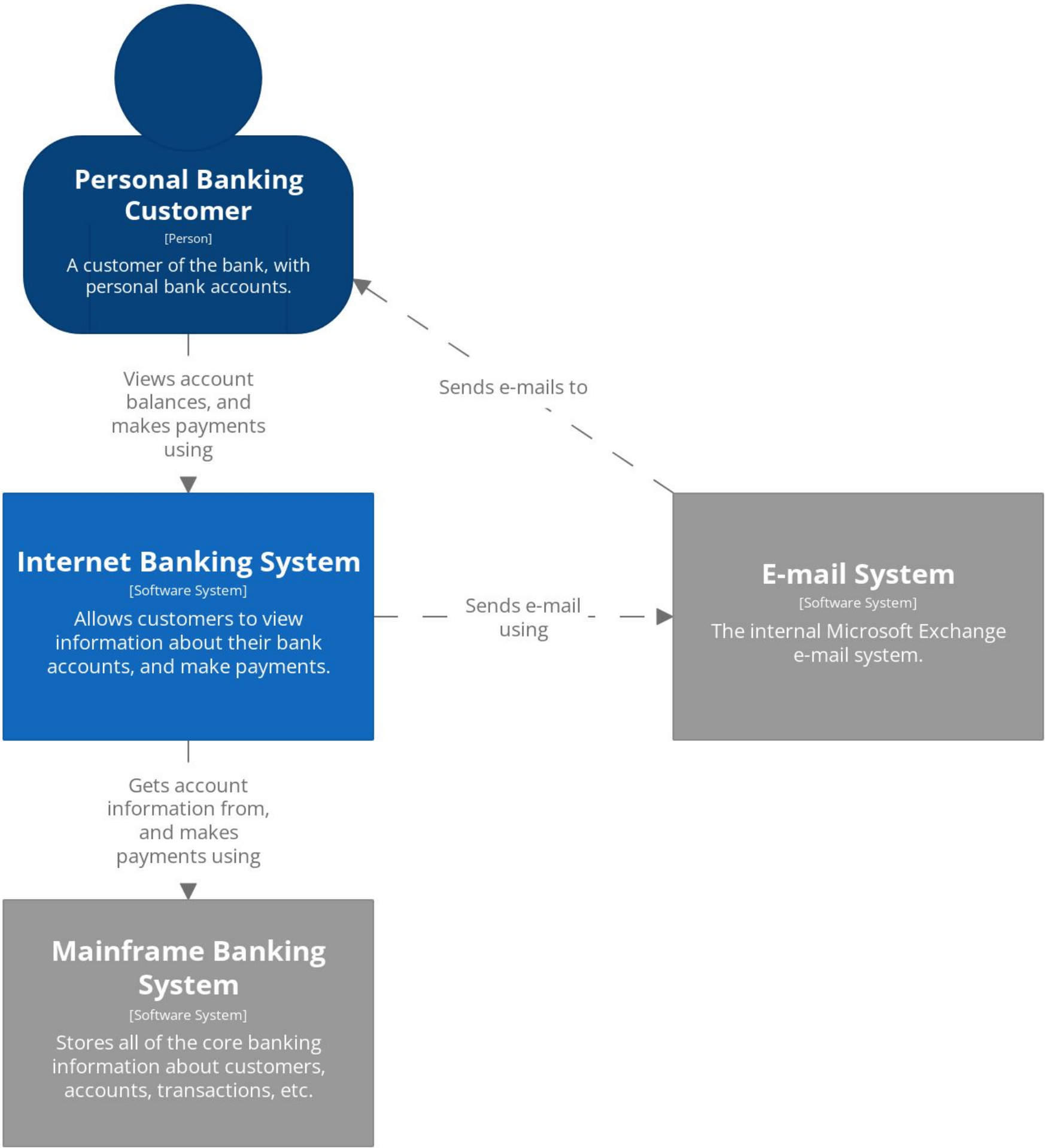
Provides context for interfaces

Can reduce accidental coupling

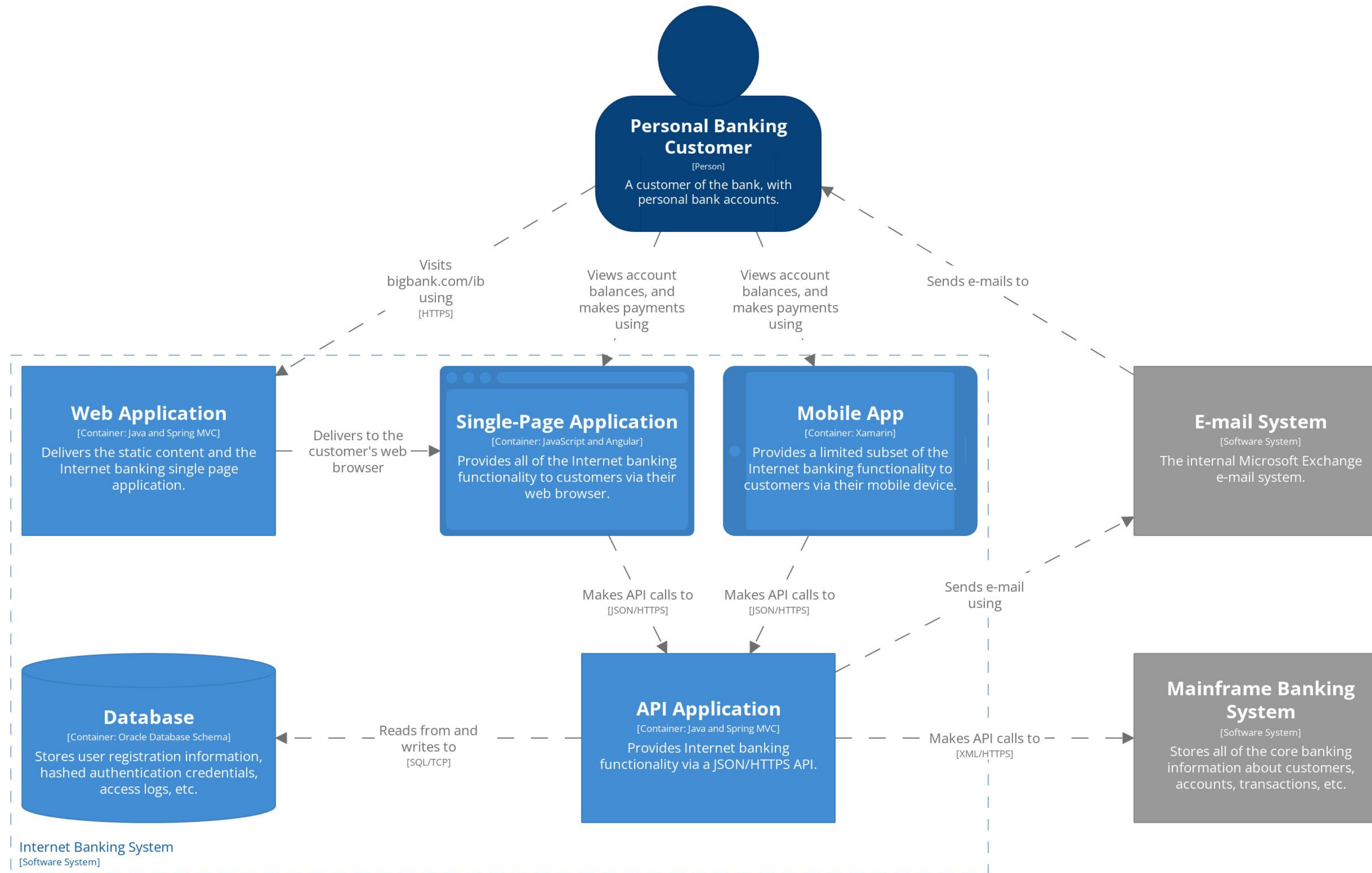
Socializes design



C1 - CONTEXT



C2 - CONTAINERS

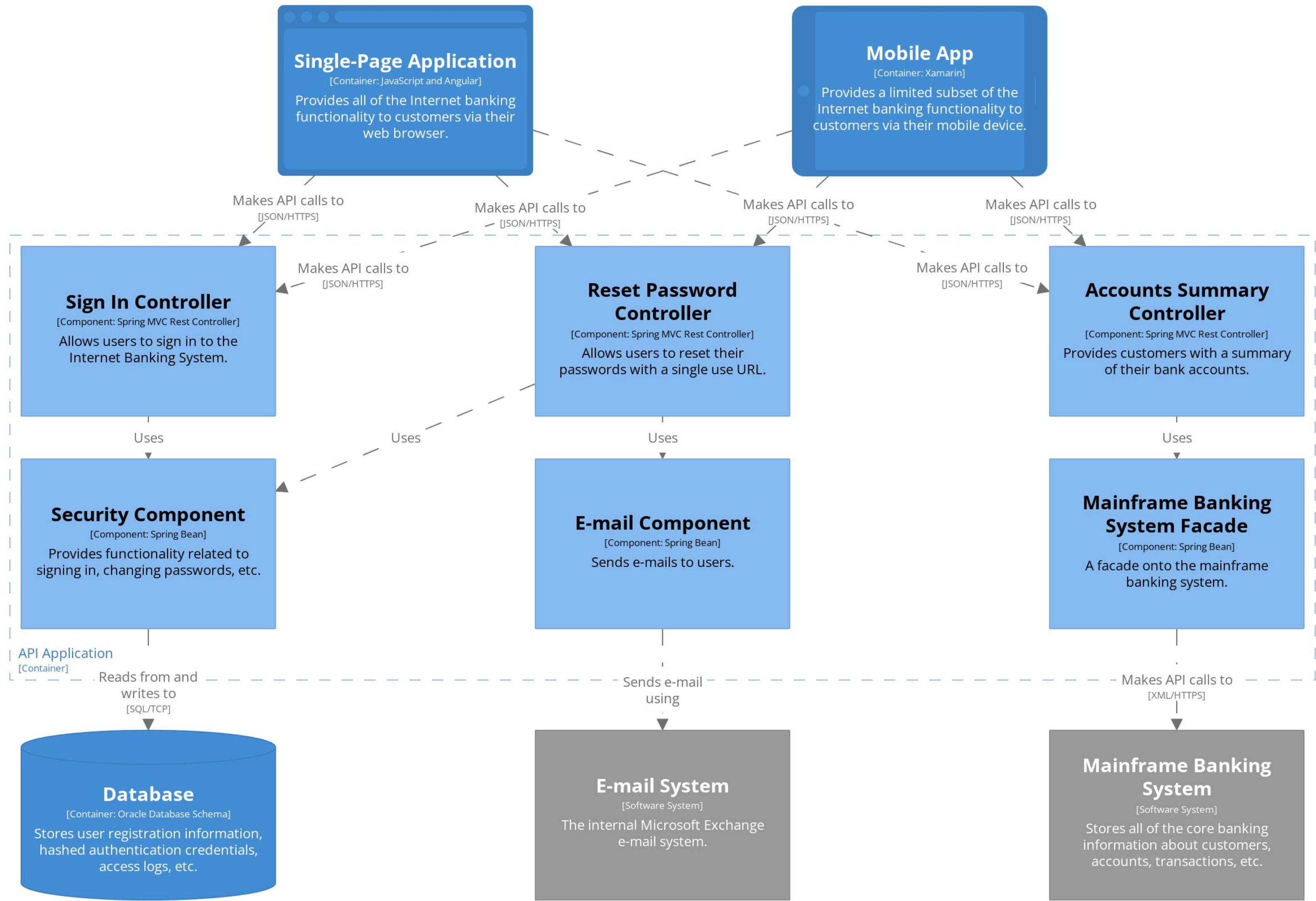


[Container] Internet Banking System

The container diagram for the Internet Banking System - diagram created with Structurizr.

Wednesday, March 22, 2023 at 8:16 AM Coordinated Universal Time

C3 - COMPONENTS

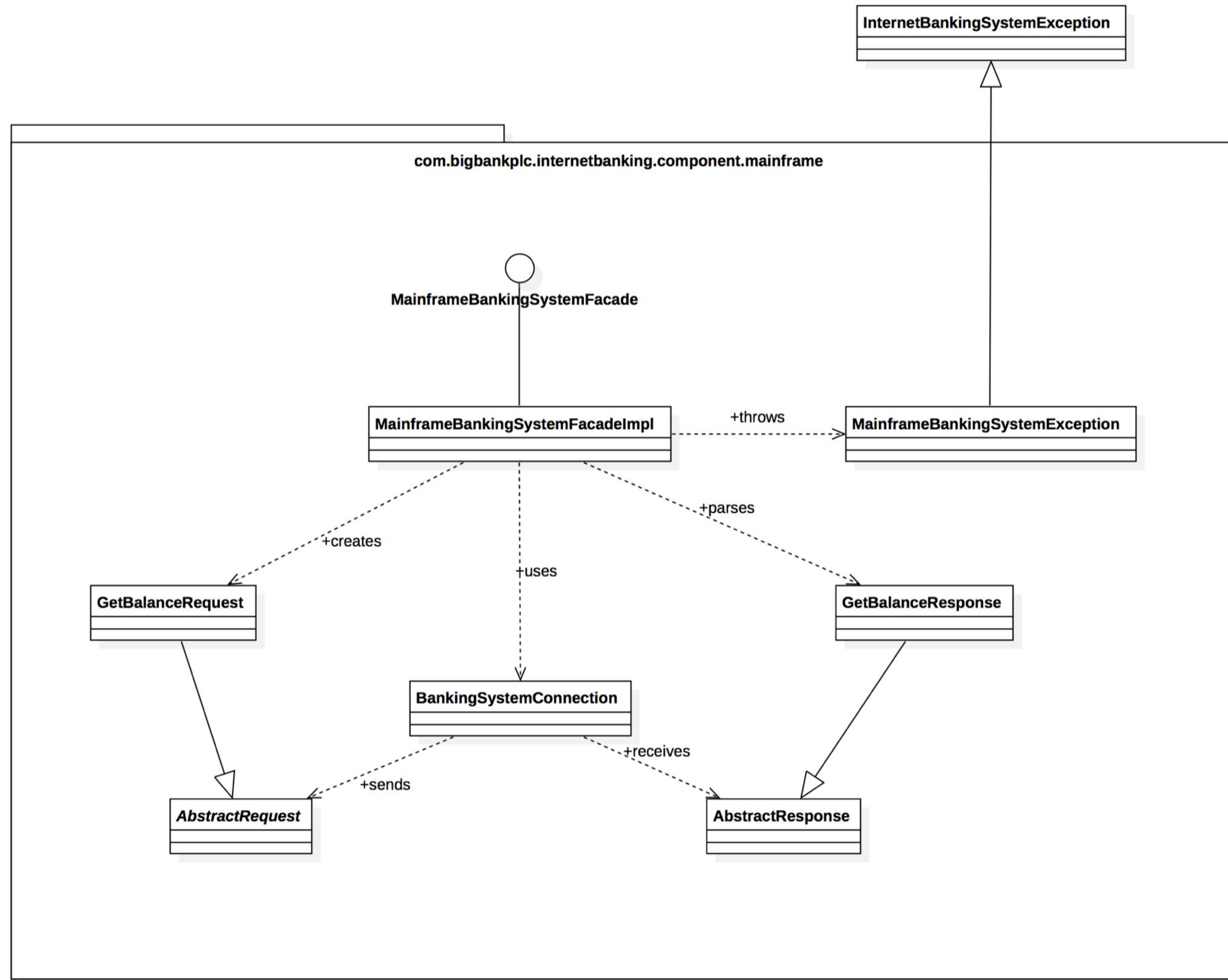


[Component] Internet Banking System - API Application

The component diagram for the API Application - diagram created with Structurizr.

Wednesday, March 22, 2023 at 8:16 AM Coordinated Universal Time

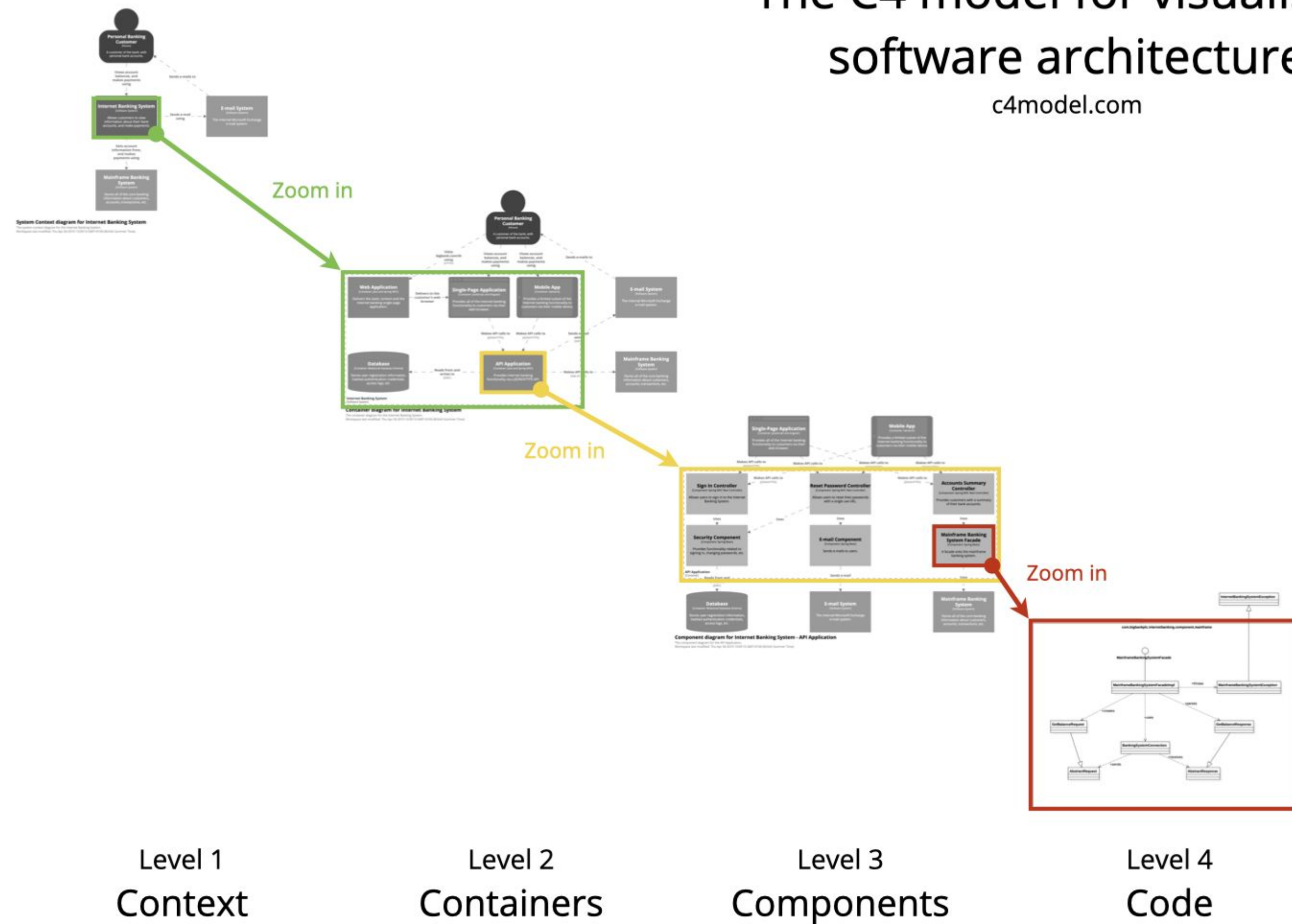
C4 - CODE / CLASS



TESTING!

The C4 model for visualising software architecture

c4model.com



GETTING STARTED

Start with a product context, an example

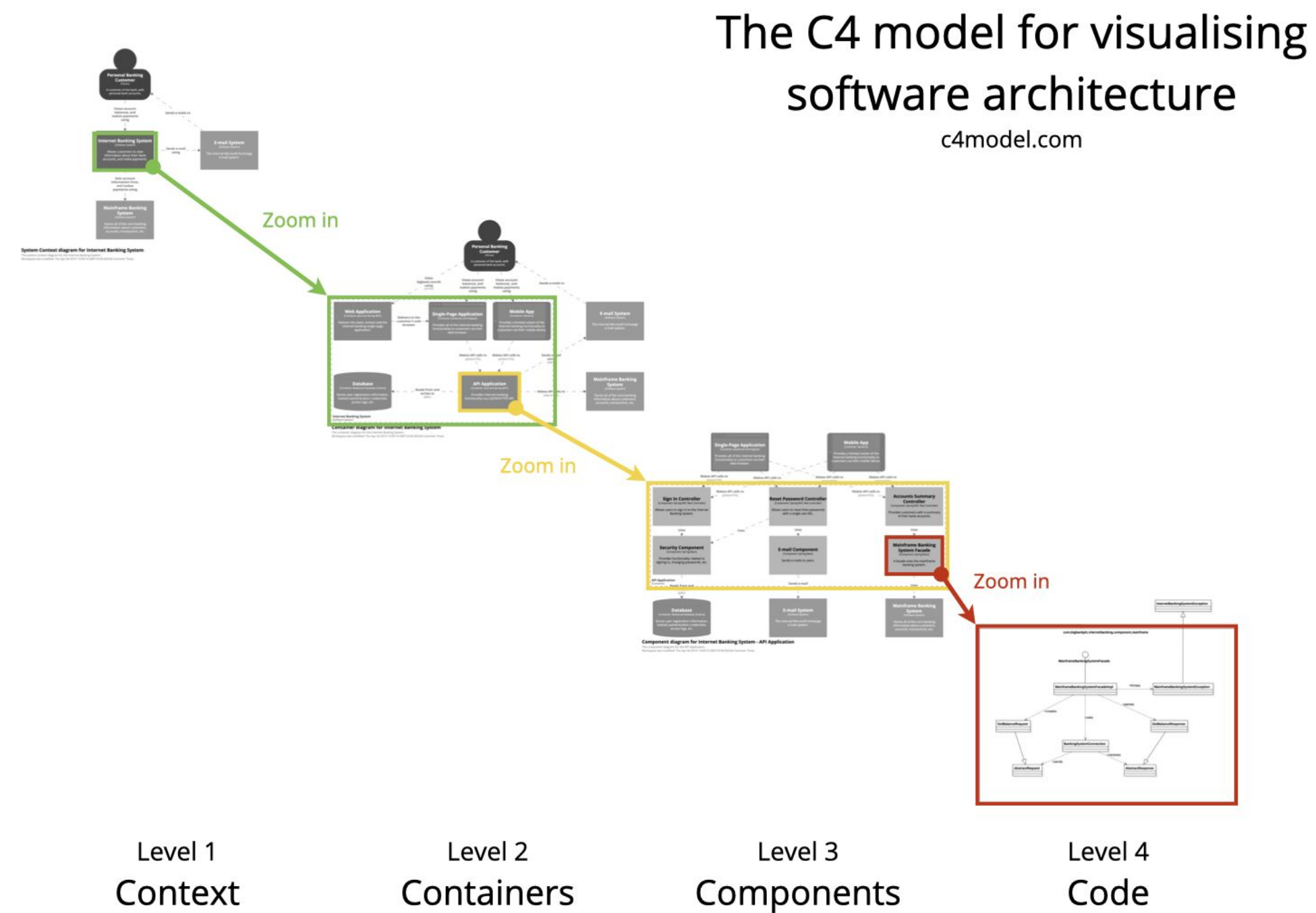
Discuss actions (think verbs)

Drill down, add fidelity

Add ideas

Socialize

Update as you learn



LEARNING AND INNOVATION WILL HAPPEN



"Learning is Hanging Out" by cogdogblog is licensed under CC BY 2.0. To view a copy of this license, visit <https://creativecommons.org/licenses/by/2.0/?ref=openverse>.

Look out for loops - simplify

Consider what changing data models or ownership may do

Piece wise and iterative

USE C4 MODELING SOCIALIZE
ARCHITECTURE AND MOVE APPROPRIATE
CONTEXT AND DECISIONS CLOSER
TO THE WORK

CONTEXT MAPPING - VISUALIZING RELATIONSHIPS

WHAT THESE PATTERNS TELL US

When they are intentional...

Cooperation (Partnership, Shared Kernel)

- High collaboration needed (but not necessarily high planning)

Customer - Supplier (Conformist, Anti-corruption Layer , Open-host service)

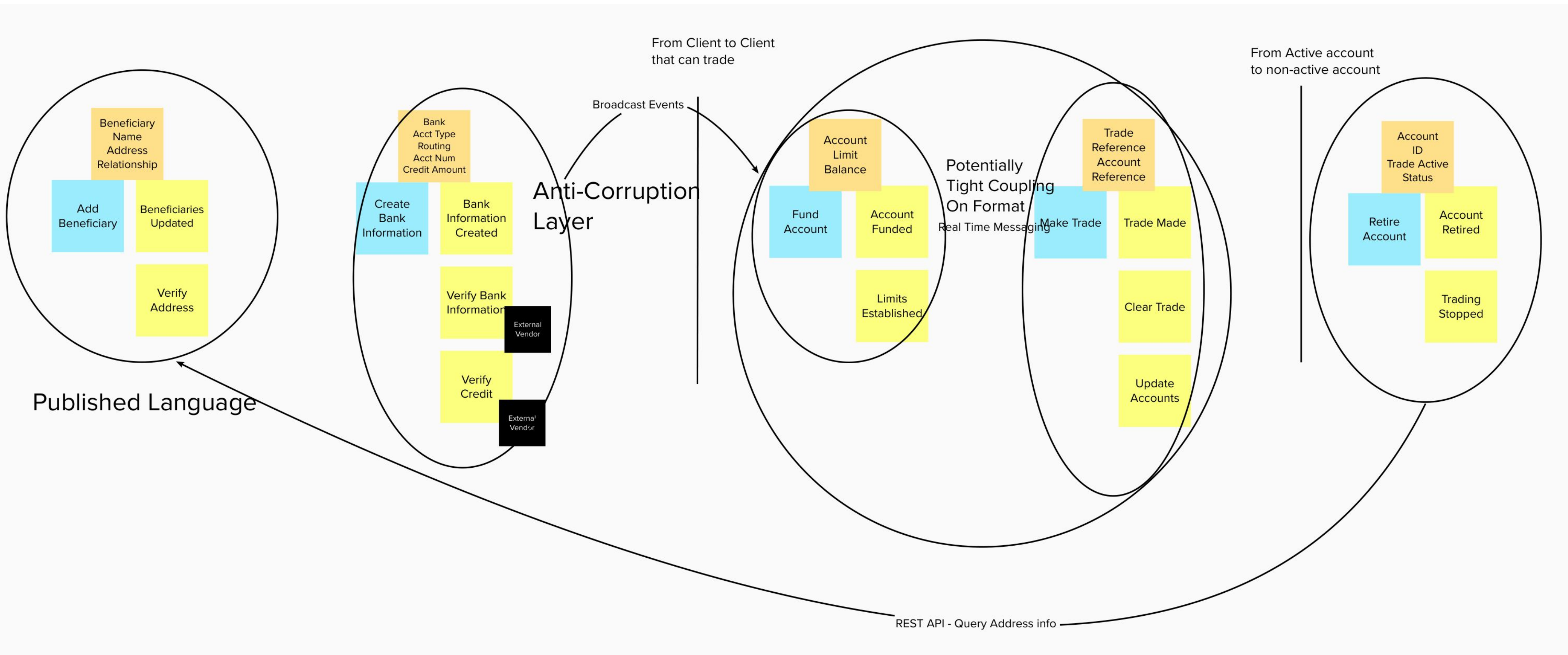
- Good ways of decoupling and allowing teams to be autonomous as much as possible

Separate Ways (Communication Issues, Generic Subdomains , Model Differences)

- Possible organizational issues - teams don't want to talk, can't agree, or are spending too much time on the wrong problem

When they are not intentional....there be dragons

OUR SAMPLE



USE CONTEXT MAPPING TO STOP
'ALL TEAMS ARE AUTONOMOUS...
JUST AS LONG AS THEY ALL FOLLOW
THE SAME PROCESS...'

CREATING STICK FIGURES AND BOXES
IS NOT YOUR JOB

PROVIDE MODELING BASED IN CONTEXT
- HELP PEOPLE REASON, WRITE LESS

OUR JOB...

TAKE A WORLD OF POSSIBILITIES...

APPLY PROPER PRODUCT UNDERSTANDING & CONTEXT...

TO DELIVER MANEUVERABLE SYSTEMS THAT BRING JOY

RECAP

Product Context Guides Architecture and Teams (Conway's Law)

Leverage Strategic DDD, C4, and Context Mapping to Reduce Cognitive Load,
Simplify Systems, and Keep Context

Appreciate Technical Debt, Refactor Our Models as We Learn
(And we will)

RECOMMENDED READING

Are Your Lights On? - Jerry Weinberg

Notes on the Synthesis of Form - Christopher Alexander

Domain Driven Design Distilled - Vaughn Vernon

Event Storming Handbook - Paul Rayner

What Is Domain-Driven Design? – Vladik Khononov

WHAT QUESTIONS DO YOU HAVE?



Joel Tosi

SLIDES—<https://www.dojoandco.com/speaking>

@joeltosi@mastodon.social

Joel.Tosi@dojoandco.com