

PRODUCT THINKING & ARCHITECTURE DISCOVERING TEAMS & BALANCING DECISIONS



Joel Tosi

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

SPONSORS

accenture



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; Blah
blah blah

EXPECTED OUTCOMES



Simple Techniques You Can Use Now

- Strategic Domain Driven Design
- C4 Modeling
- Architectural Decision Records

Embracing Learning - Simply

Leverage Product Context to Ground Decisions

"Augsburg, Germany. 2014. Same idea, different outcome." by Boris Thaser is licensed under CC BY 2.0.

CONTEXT AND THE WIKI



"Ward Cunningham" by Joi 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?

WHAT IS ARCHITECTURE?

ONE VIEW



Web

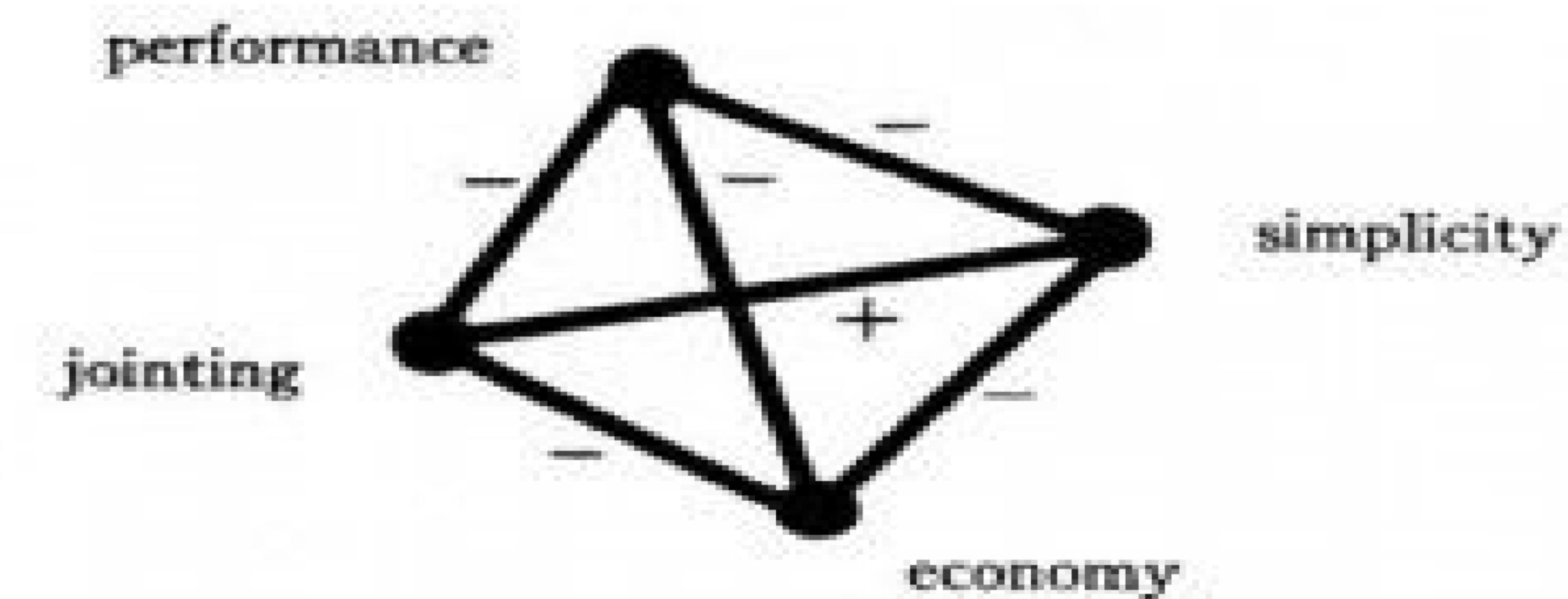
API

Logic / Calculations

Fast Data

Slow Data

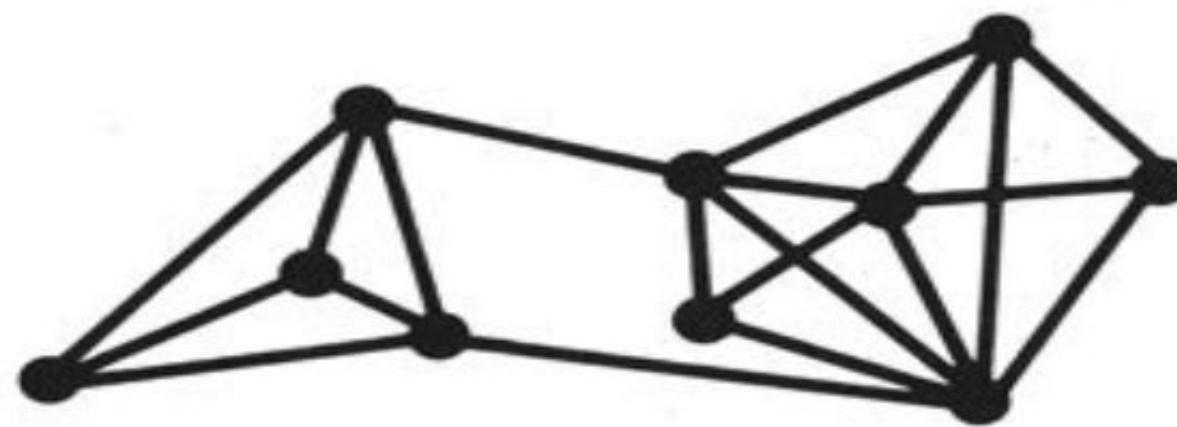
WHAT IS ARCHITECTURE? ANOTHER VIEW



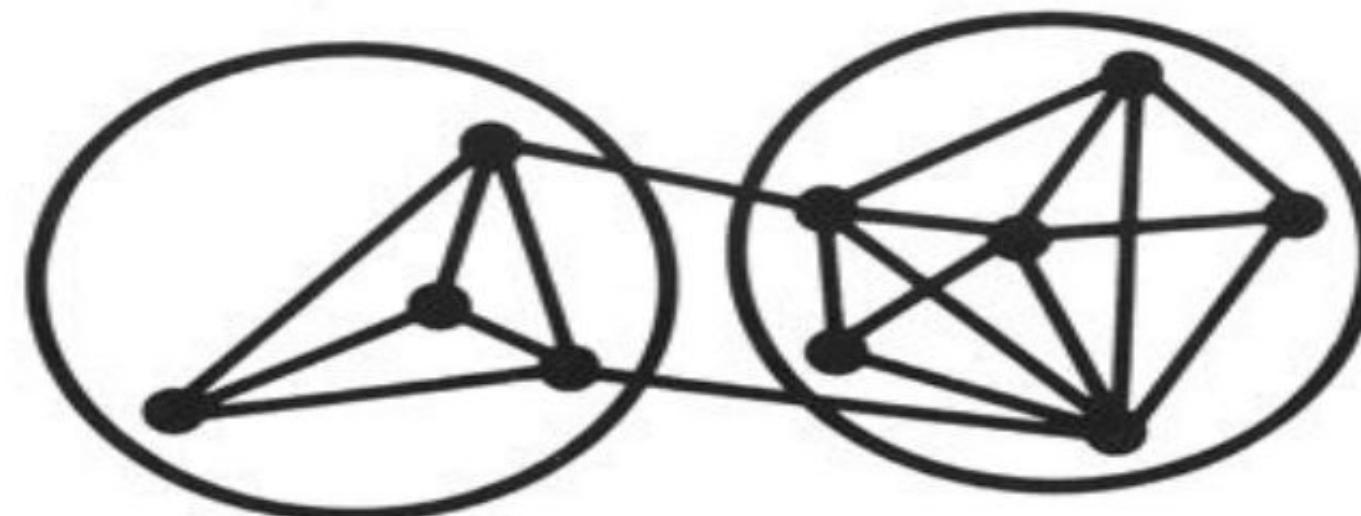
This is a typical design problem; it has requirements which have to be met; and there are interactions between the requirements, which makes the requirements hard to meet. This problem is simple to solve. It falls easily within the compass of a single man's intuition. But what about a more complicated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

© Notes on the Synthesis of Form – Christopher Alexander

WHAT IS ARCHITECTURE? ANOTHER VIEW



Reflection of System Dependencies



Optimal Decomposition / Grouping



Biased Simplification - mereotopology

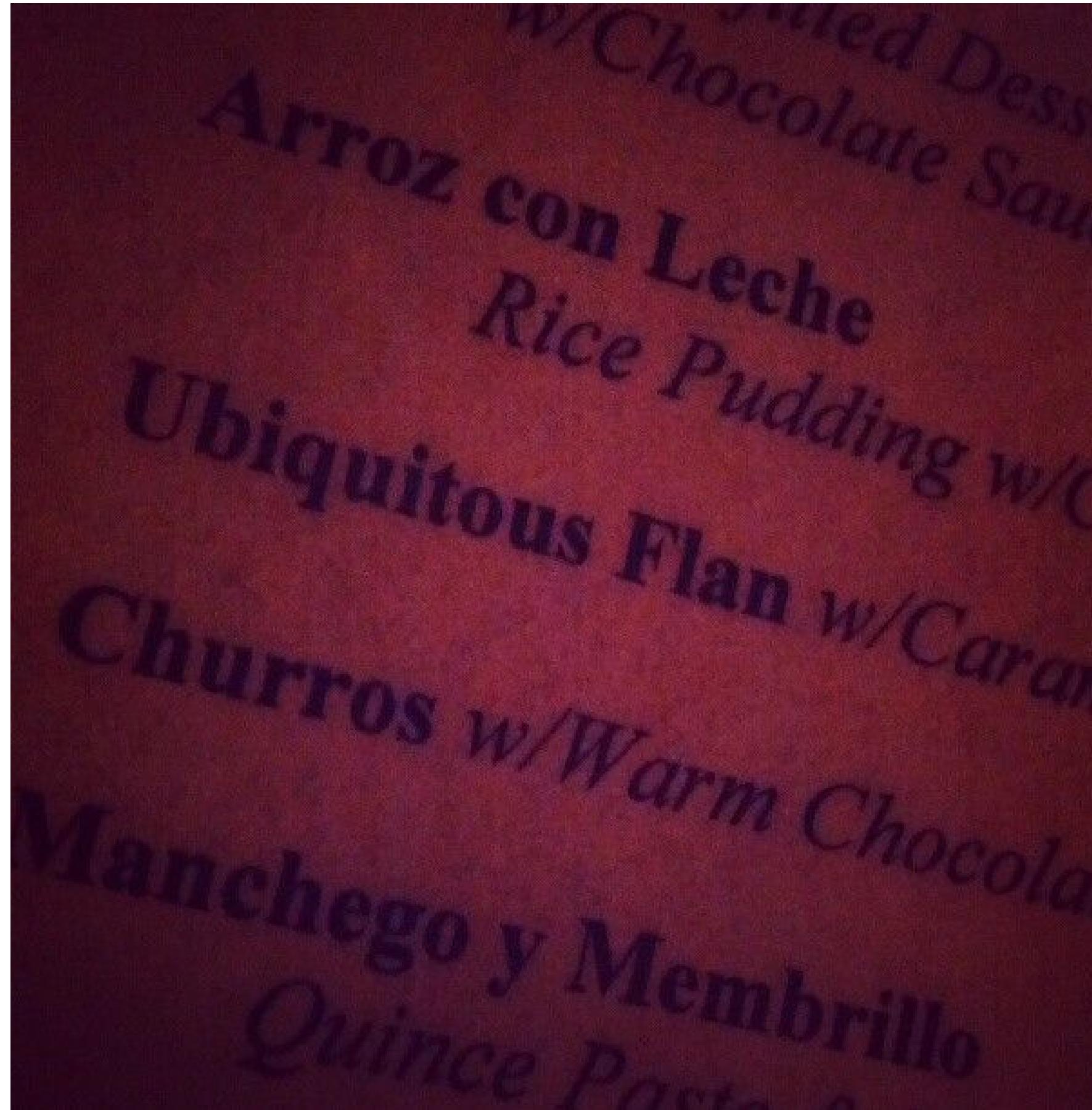
© Notes on the Synthesis of Form - Christopher Alexander

ARCHITECTURE IS ABOUT FIT
WITHIN CONTEXT

USING PRODUCT CONTEXT TO GROUND ARCHITECTURE

**PROBLEM: TEAMS ARE STEPPING
ON EACH OTHERS TOES
OR
INTENTIONAL COUPLING VS ACCIDENTAL**

WHAT IS STRATEGIC DDD? WHY MIGHT WE CONSIDER IT?



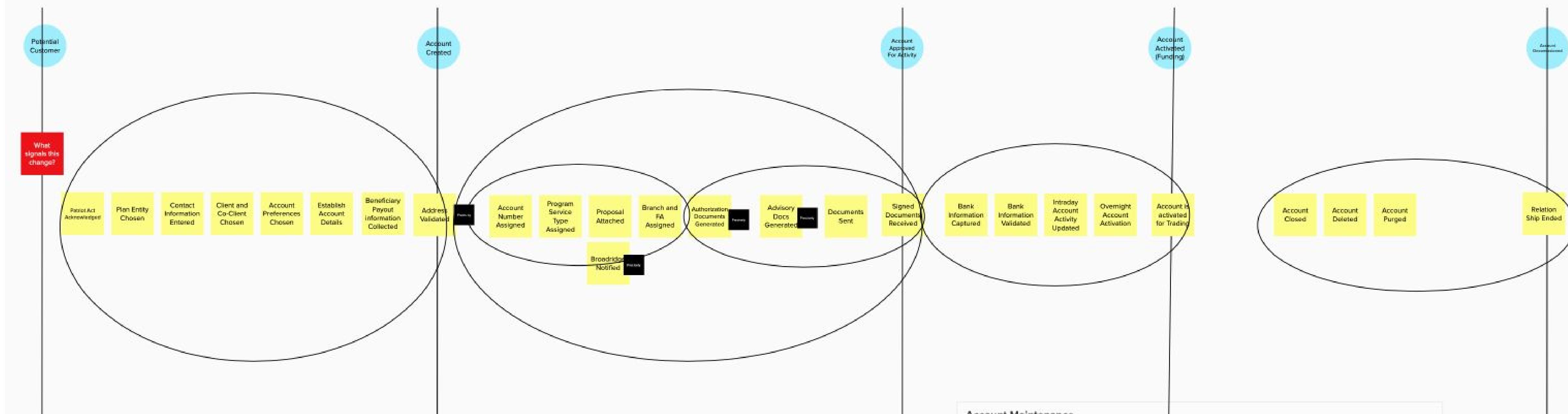
We are trying to model domains / teams closely to product intent

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>.

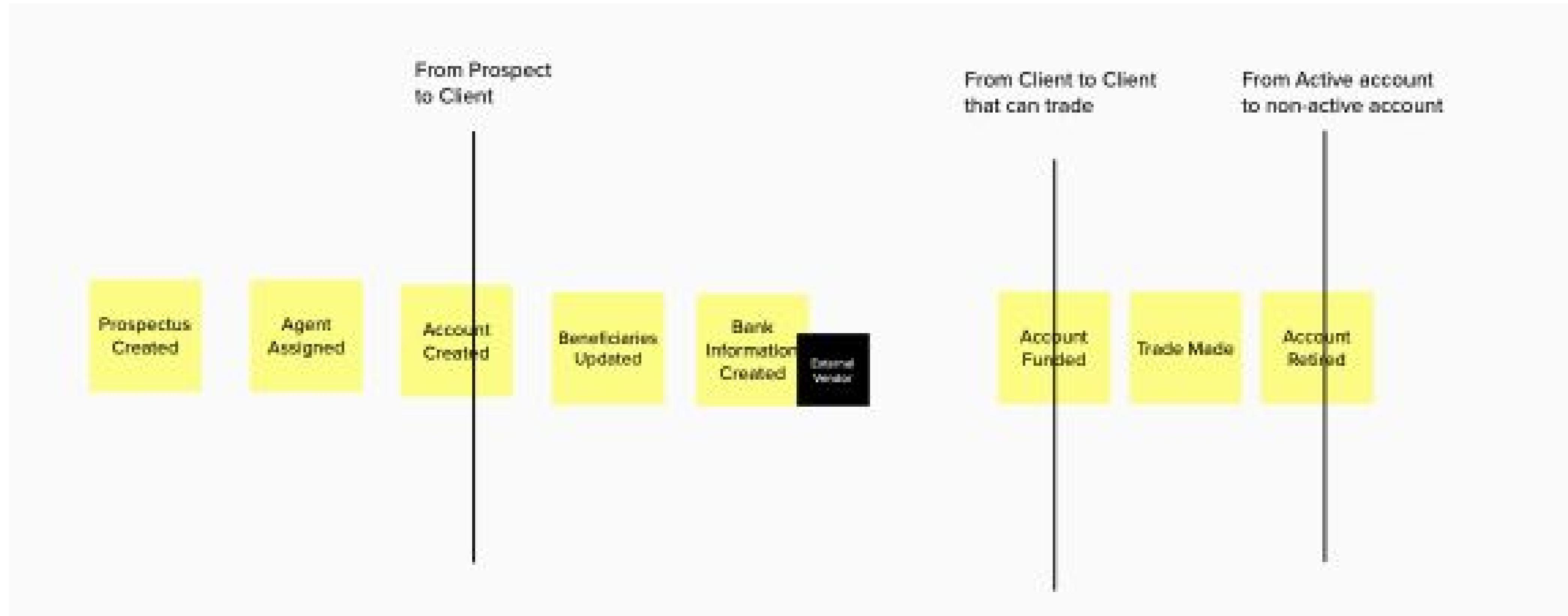
AN EXAMPLE



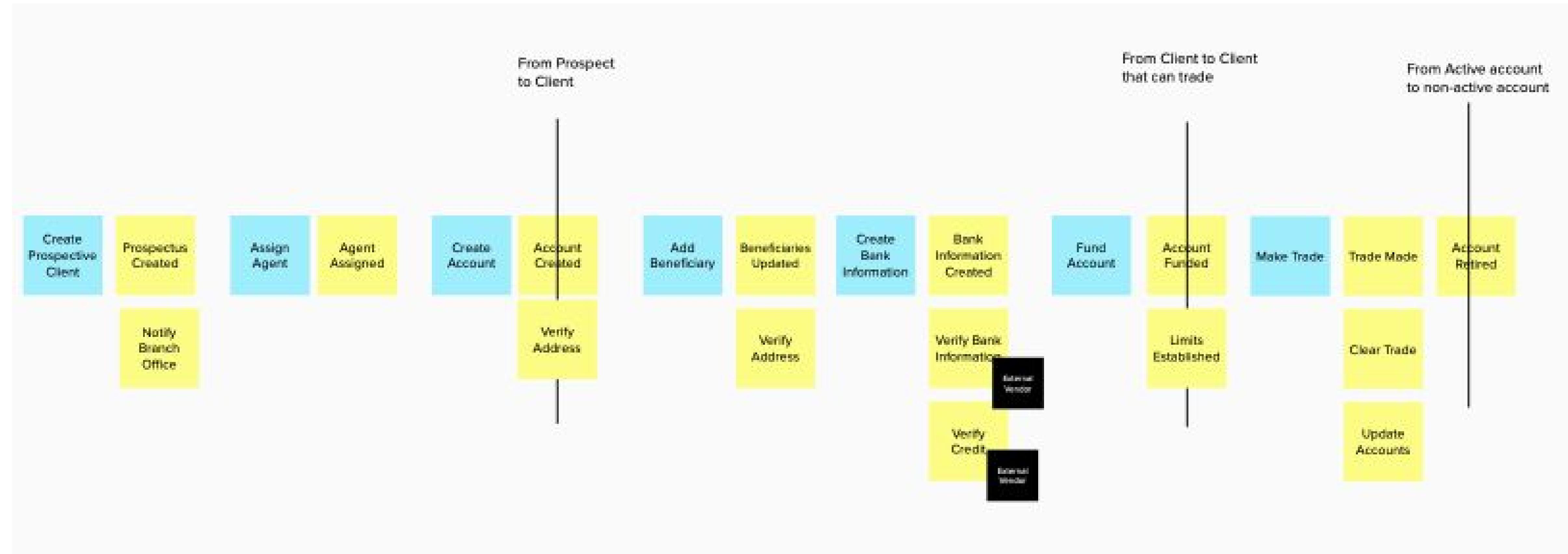
HOW DID WE GET THERE?



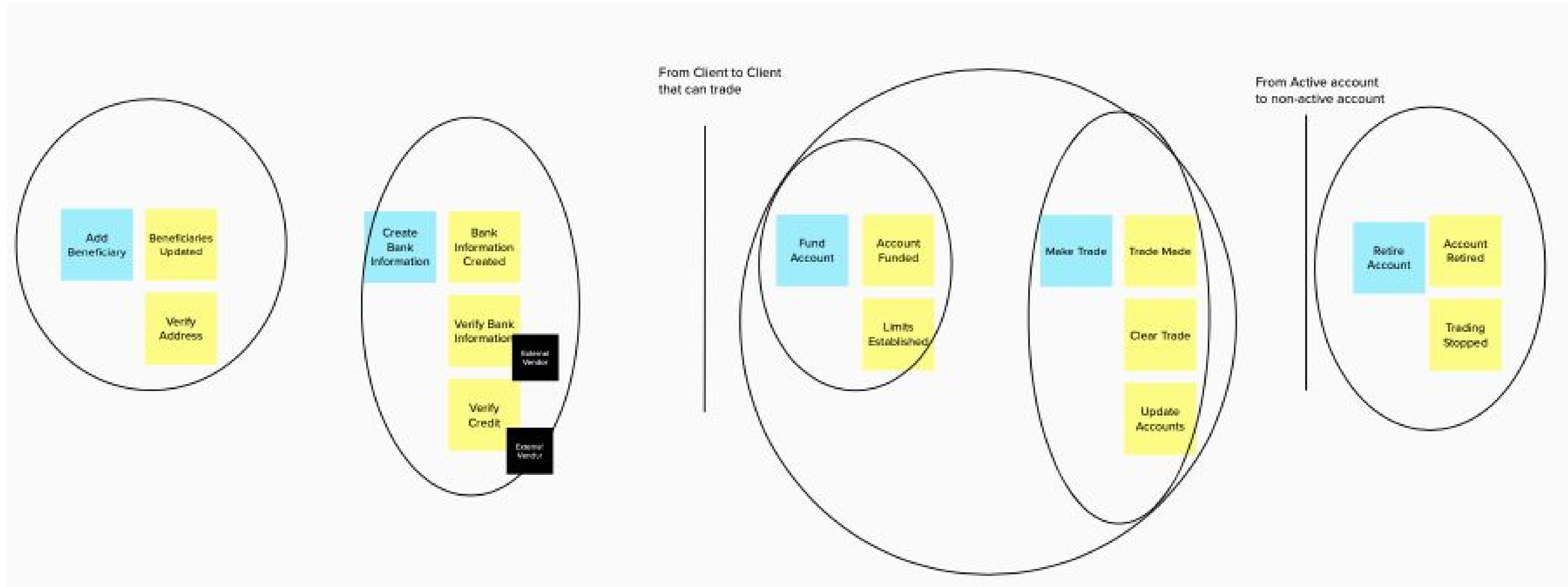
HOW DID WE GET THERE?



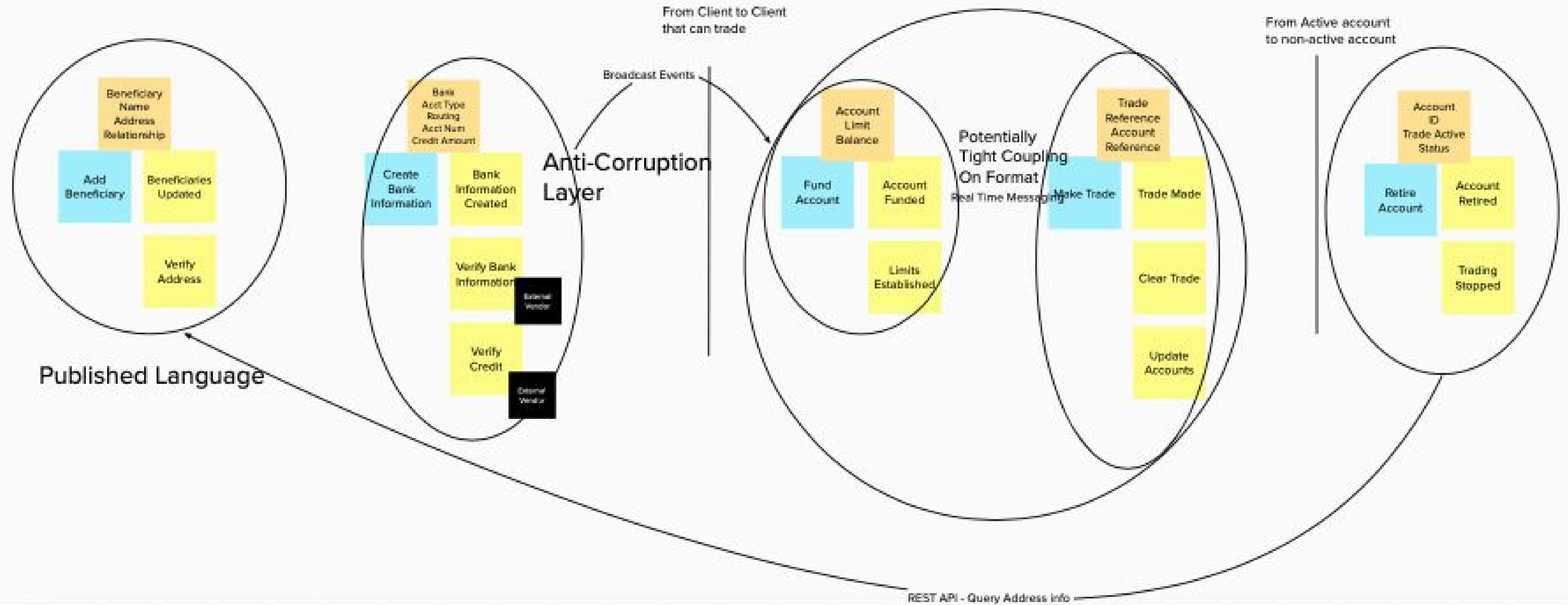
HOW DID WE GET THERE?



HOW DID WE GET THERE?

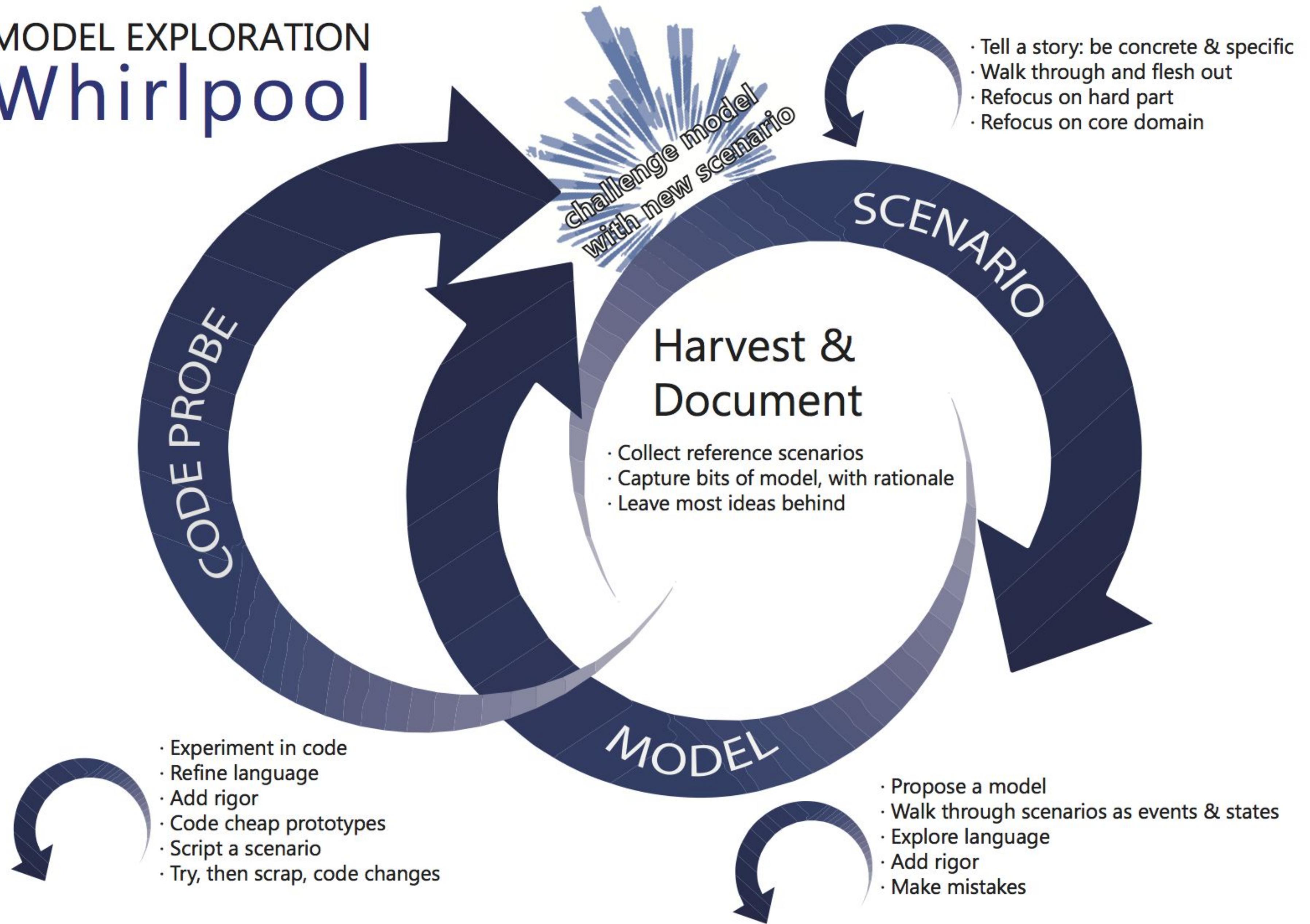


HOW DID WE GET THERE?



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

"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

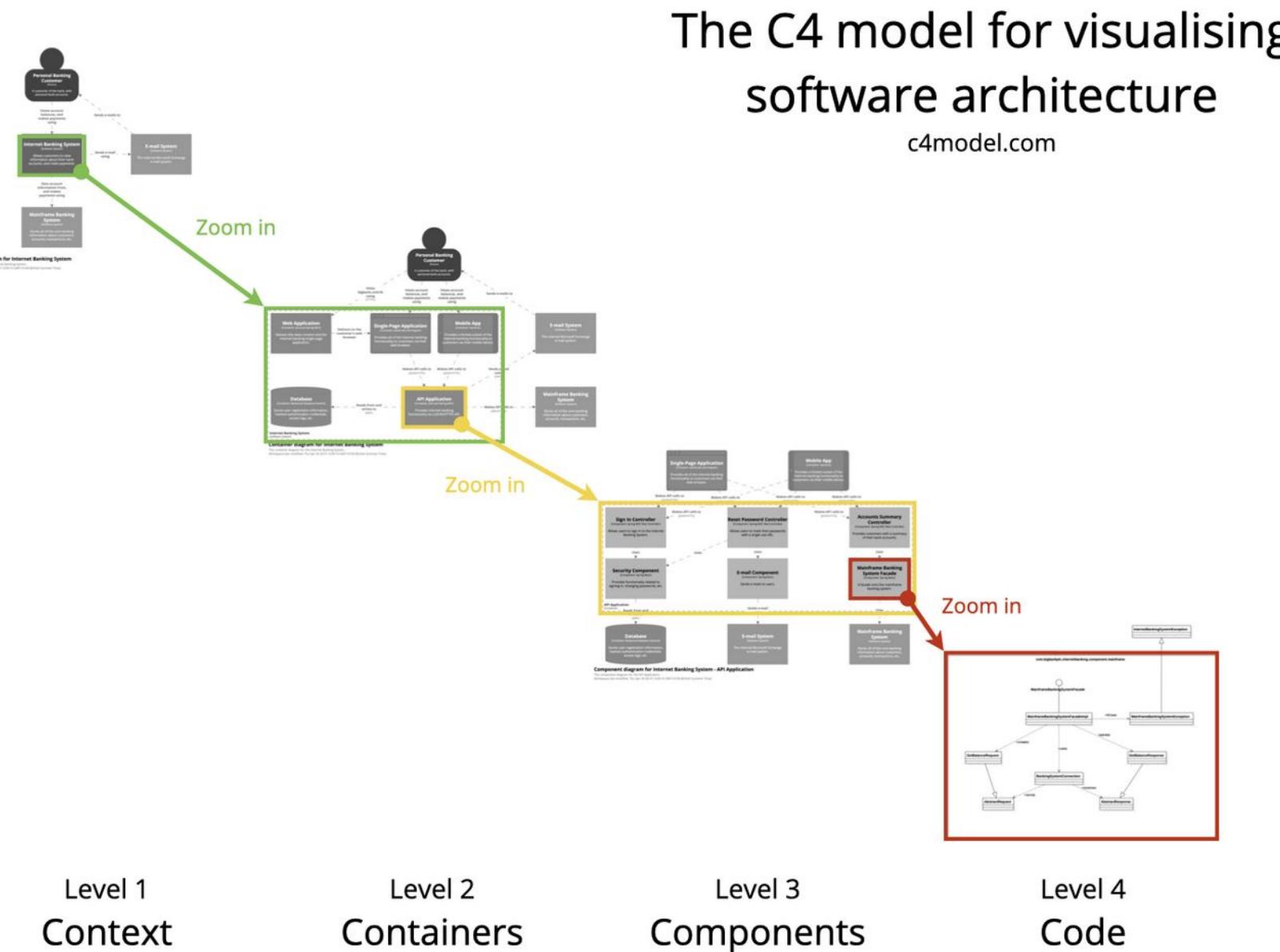
AS YOUR PRODUCT EVOLVES,
SO DOES YOUR MODEL

PROBLEM: SYSTEMS ARE LARGE
WHERE DO WE FIT?

WHAT DO WE CONSUME / PRODUCE
AND FOR WHO?

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

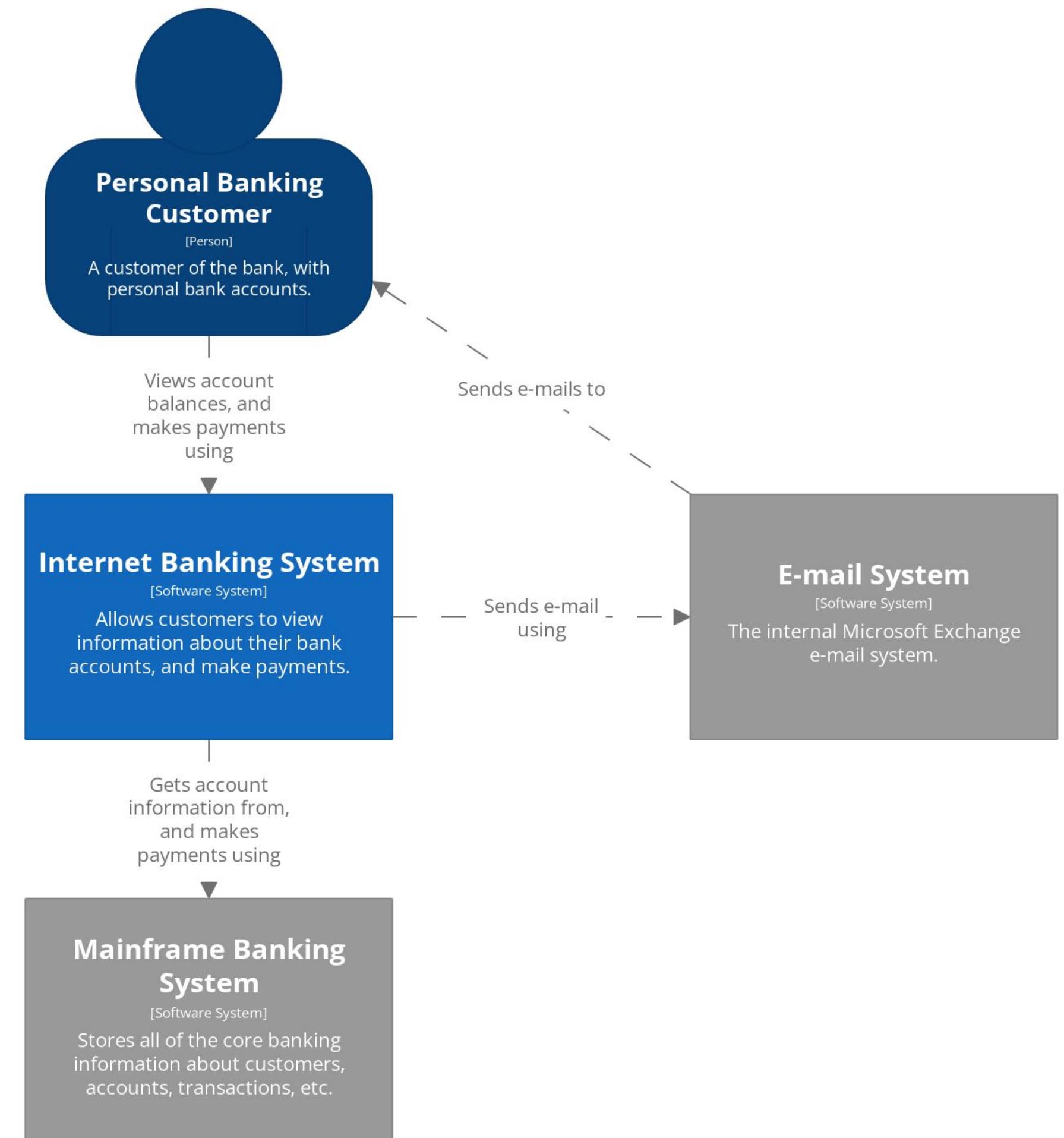
See how / where we fit; Boundaries

Provides context for interfaces

Can reduce accidental coupling

Socializes design

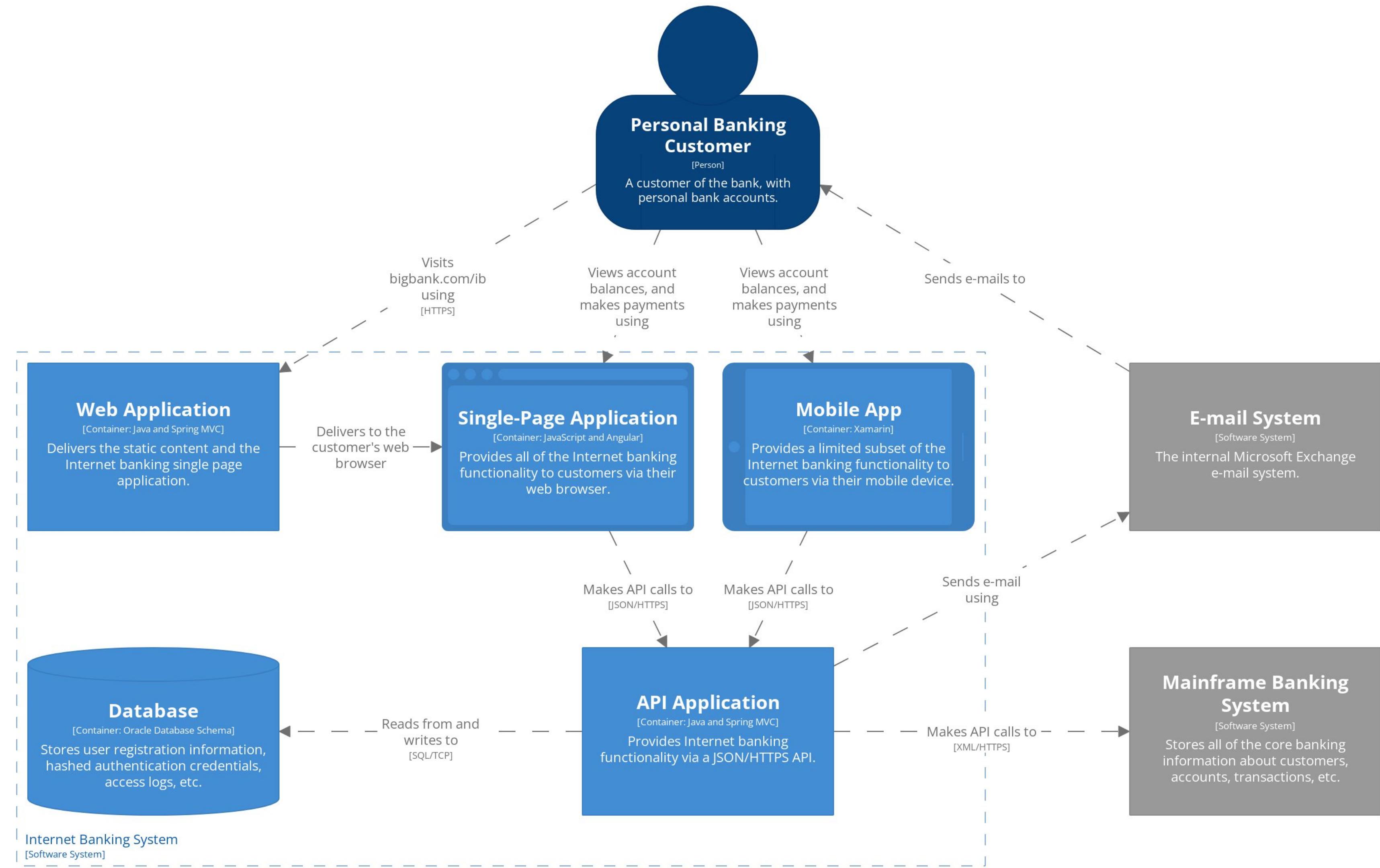
C1 - CONTEXT



[System Context] Internet Banking System

The system context diagram for the Internet Banking System - diagram created with Structurizr.
Wednesday, March 22, 2023 at 8:16 AM Coordinated Universal Time

C2 - CONTAINERS

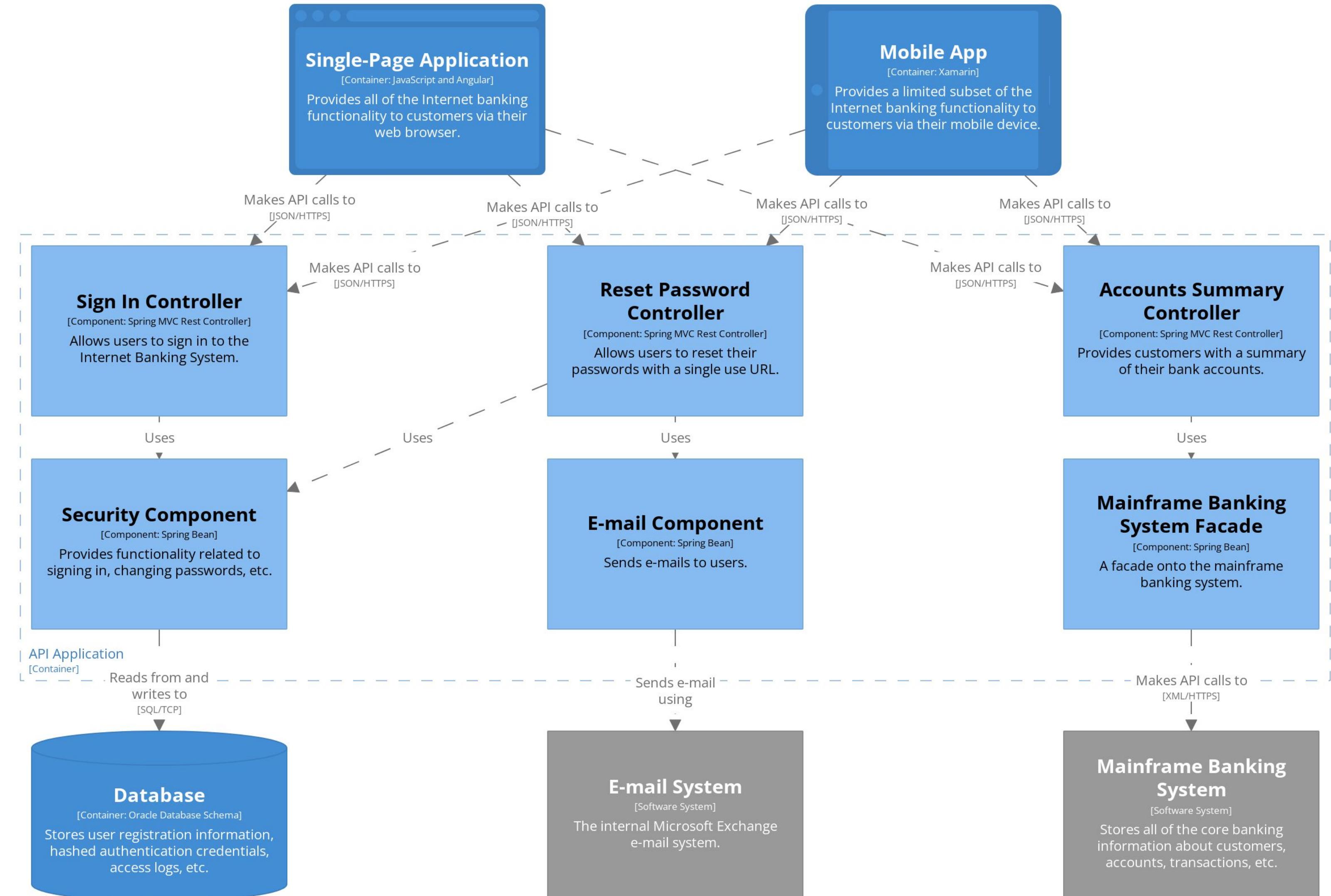


[Container] Internet Banking System

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

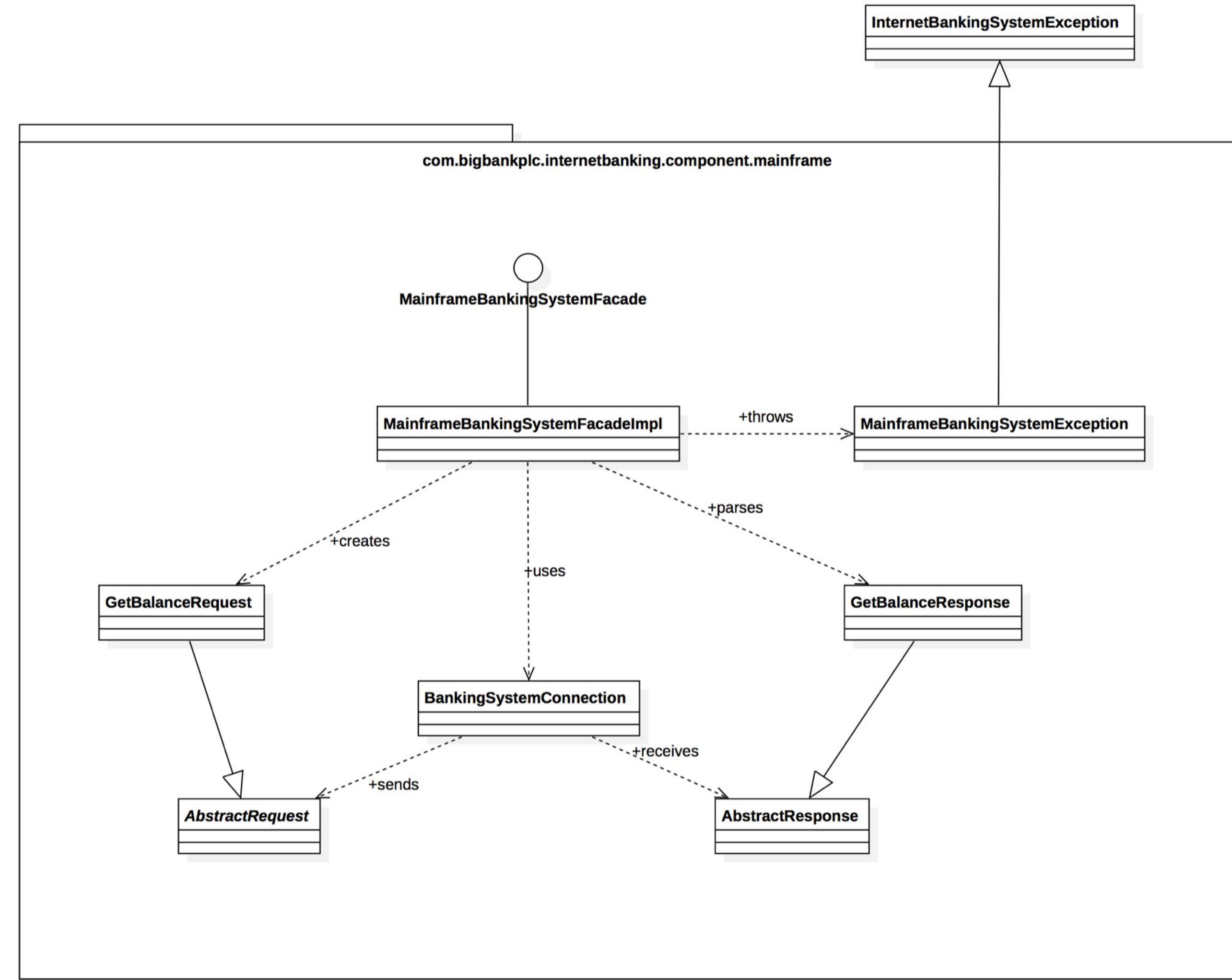


[Component] Internet Banking System - API Application

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

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

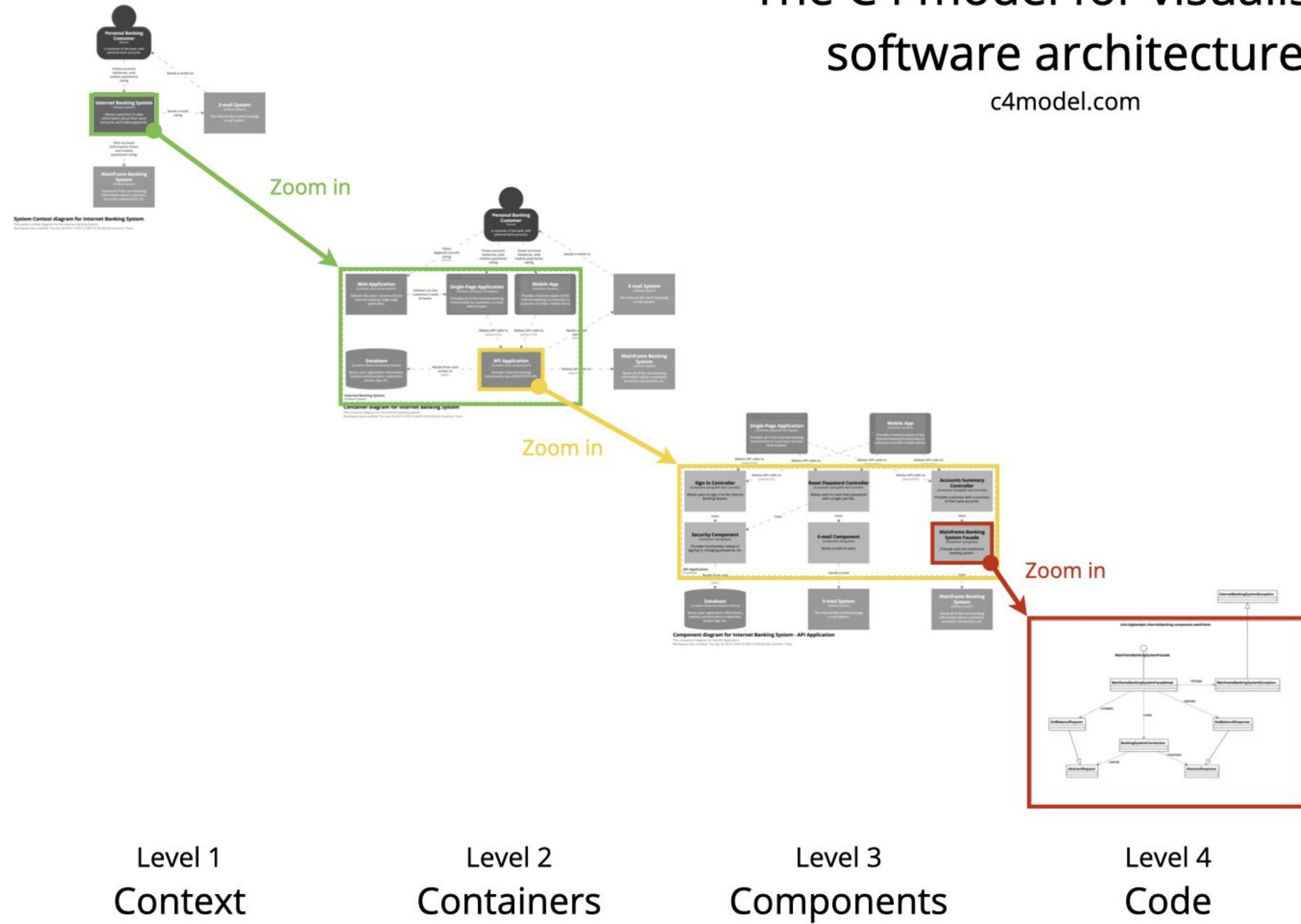
C4 - CODE / CLASS



TESTING!

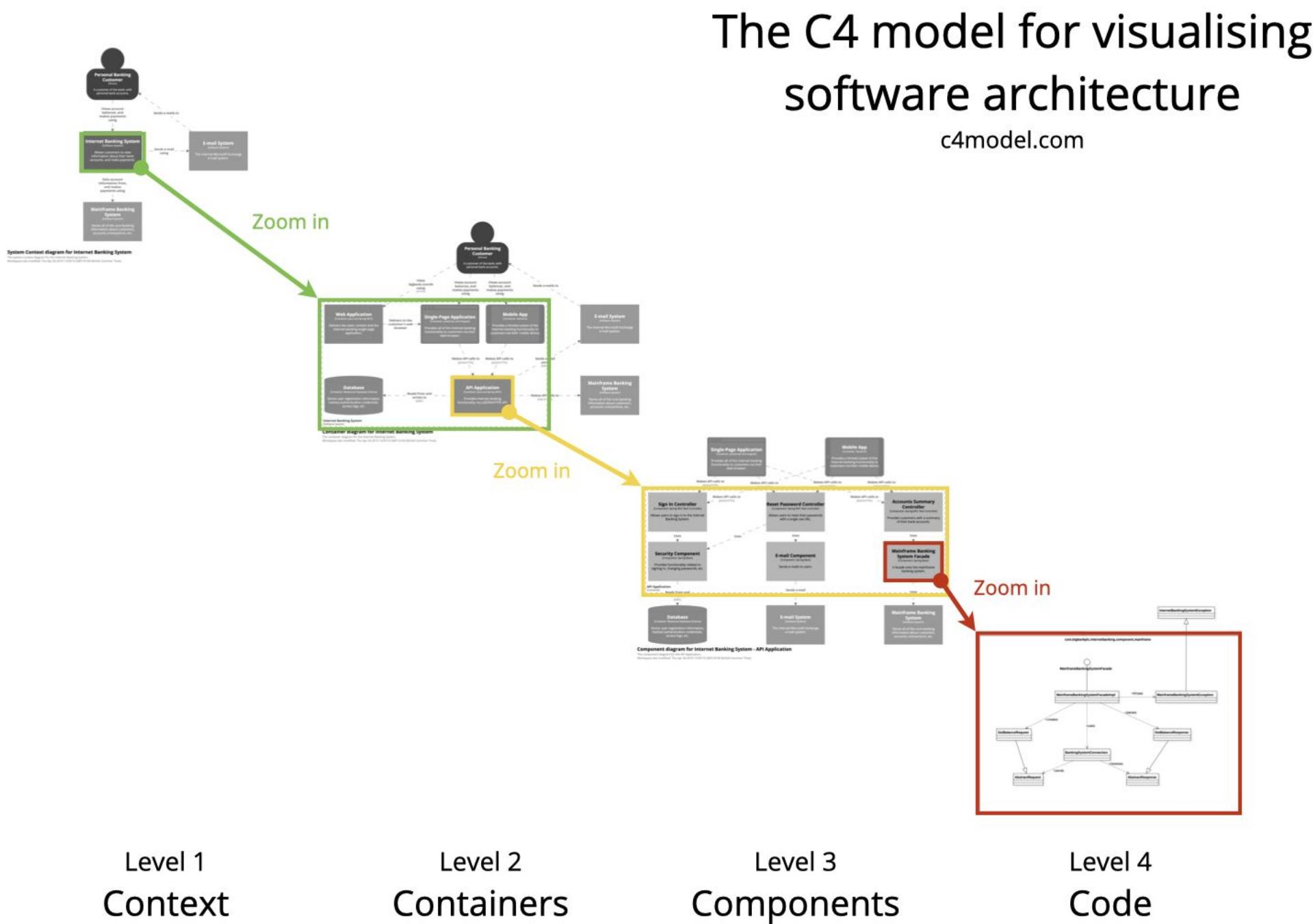
The C4 model for visualising
software architecture

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



Look out for loops - simplify

Consider what changing data models or ownership may do

Piece wise and iterative

"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>.

**PROBLEM: WHY DECISIONS ARE MADE,
AND TRADEOFFS, ARE QUICKLY LOST**

WHAT ARE ADRs? WHY MIGHT WE CONSIDER THEM?

Architecture Decision Records are a way of recording, updating, and sharing decisions made

We are trying to keep a record close to the code that ties decisions with context, purpose, and trade-offs

Lives in repo, updated as we learn

Remember Ward!

GETTING STARTED

Nat Pryce upgrade date format in the project's own ADRs

Latest commit edb7175 on Dec 7, 2017 History

0 contributors

28 lines (17 sloc) | 645 Bytes

2. Implement as shell scripts

Date: 2016-02-12

Status
Accepted

Context
ADRs are plain text files stored in a subdirectory of the project.
The tool needs to create new files and apply small edits to the Status section of existing files.

Decision
The tool is implemented as shell scripts that use standard Unix tools -- grep, sed, awk, etc.

Consequences
The tool won't support Windows. Being plain text files, ADRs can be created by hand and edited in any text editor. This tool just makes the process more convenient.
Development will have to cope with differences between Unix variants, particularly Linux and MacOS X.

For a decision -
Set the context

Call out the decision

Call out the tradeoffs

Date it

Revisit it

<https://cognitect.com/blog/2011/11/15/documenting-architecture-decisions>

ON BEING WRONG - TRY THIS

- 1) Document Assumptions You Are Making
- 2) Document Technical Assumptions You Are Making
- 3) Think of 3 Ways You Might Be Wrong
- 4) For Each Learning Opportunity, What Would You Change?

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 ADR 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 QUESTIONS DO YOU HAVE?



Joel Tosi

SLIDES - <https://www dojoandco com/speaking>

@joeltosi@mastodon.social

Joel.Tosi@dojoandco.com