BLENDING PRODUCT THINKING WITH ARCHITECTURE

EST DOJO 2017

AND Co

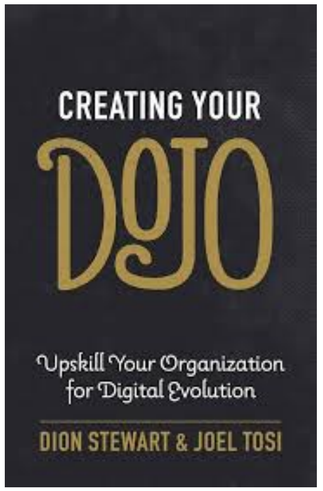
Joel Tosi

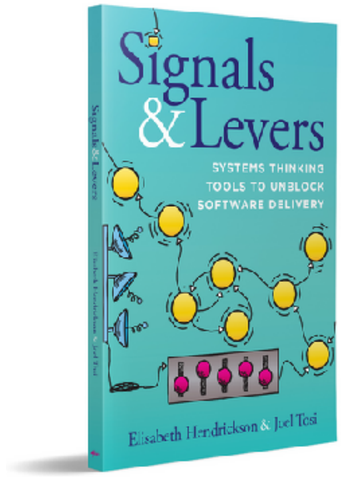
SLIDES – https://www.dojoandco.com/speaking

@joeltosi@mastodon.social

Joel.Tosi@dojoandco.com

ABOUT ME





September 2026
IT Revolution

CME Group

verizon

ATLASSIAN

CENTENE Corporation

Northwestern Mutual

MARK43

Edward Jones

IVA

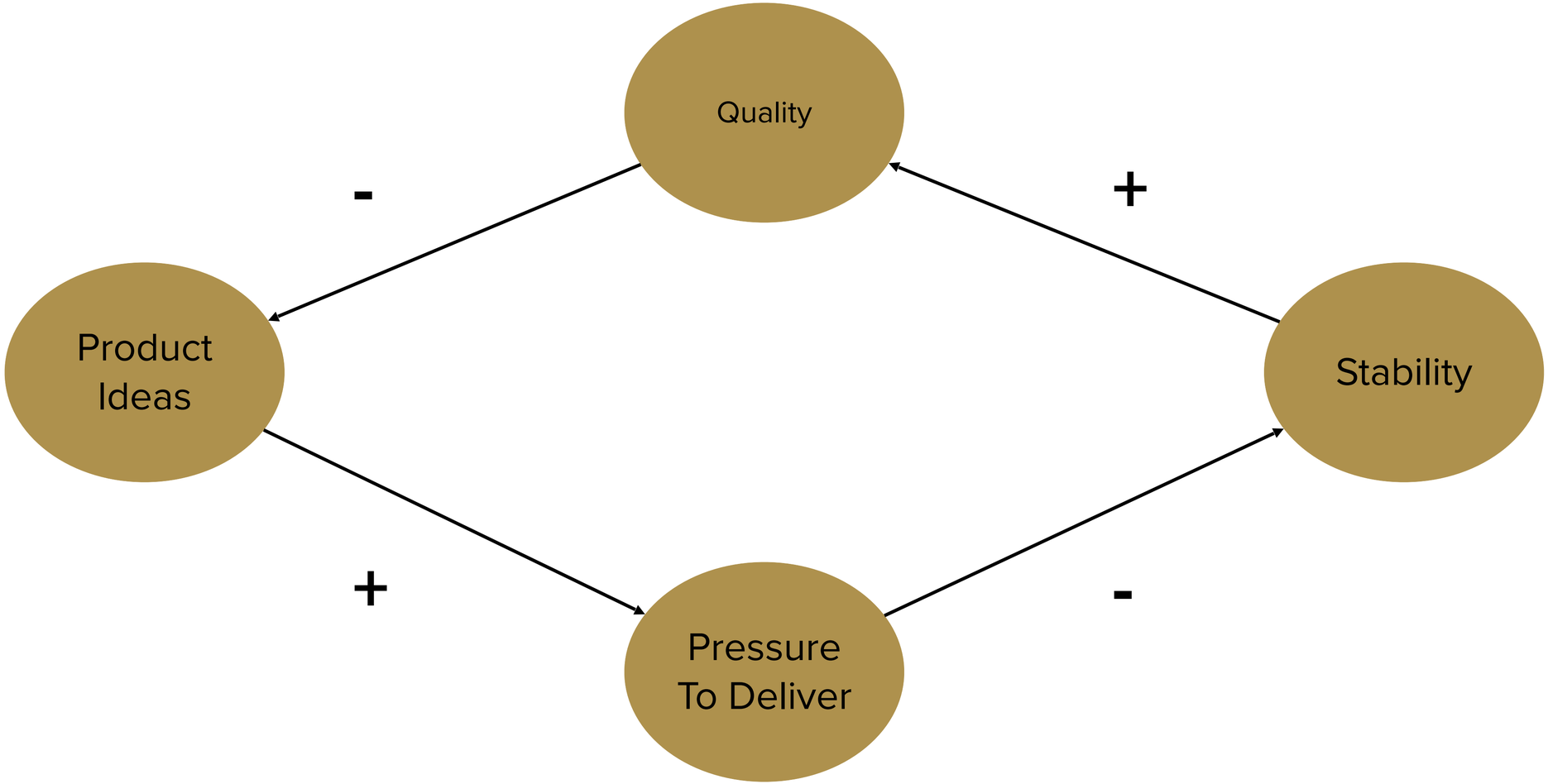
STELLANTIS

FCA
FIAT CHRYSLER AUTOMOBILES

Human Resources and Social Development Canada
Ressources humaines et Développement social Canada

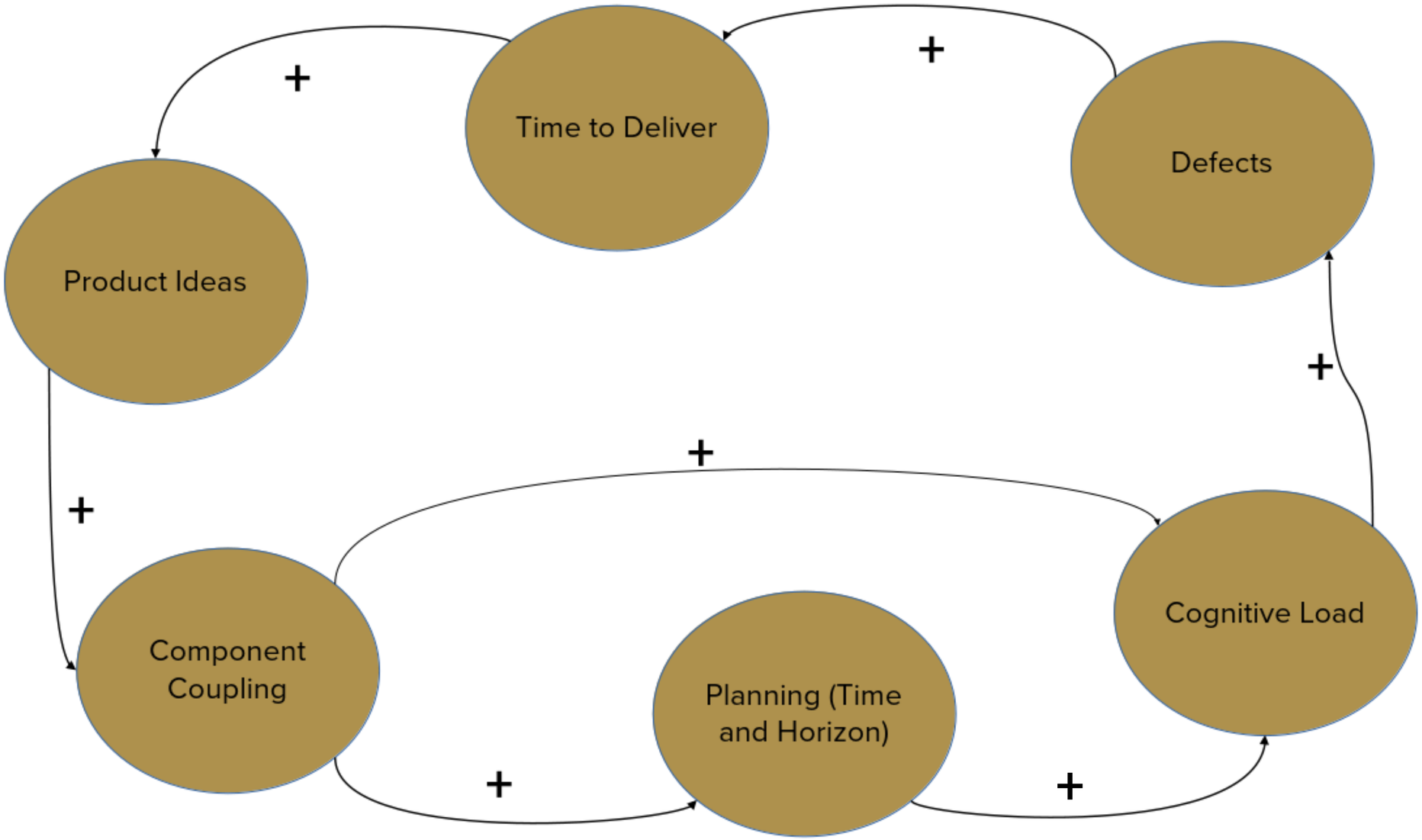
NG

WHY DOES IT TAKE LONGER TO GET STUFF DONE?

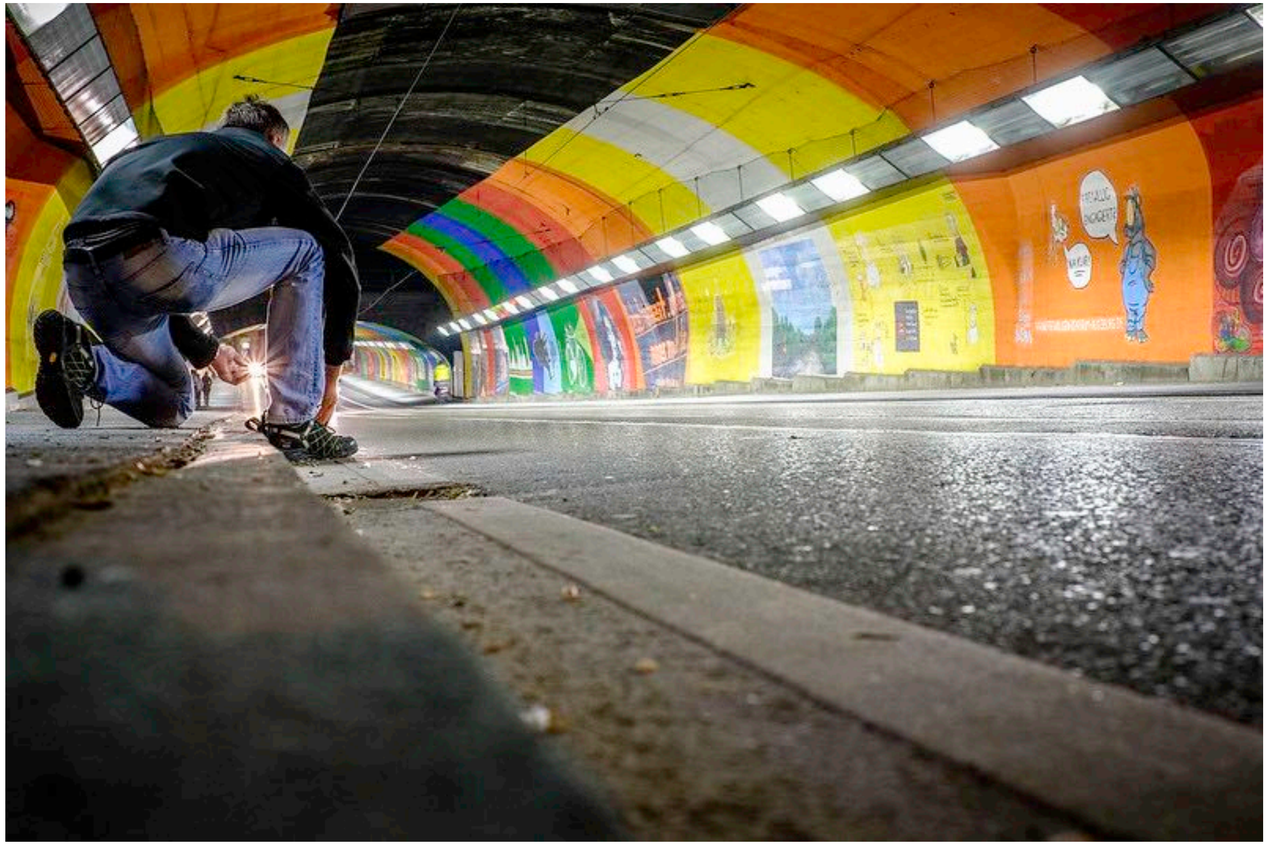


©2026 Dojo and Co

A VICIOUS CYCLE



EXPECTED OUTCOMES



Simple Techniques You Can Use Now

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

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

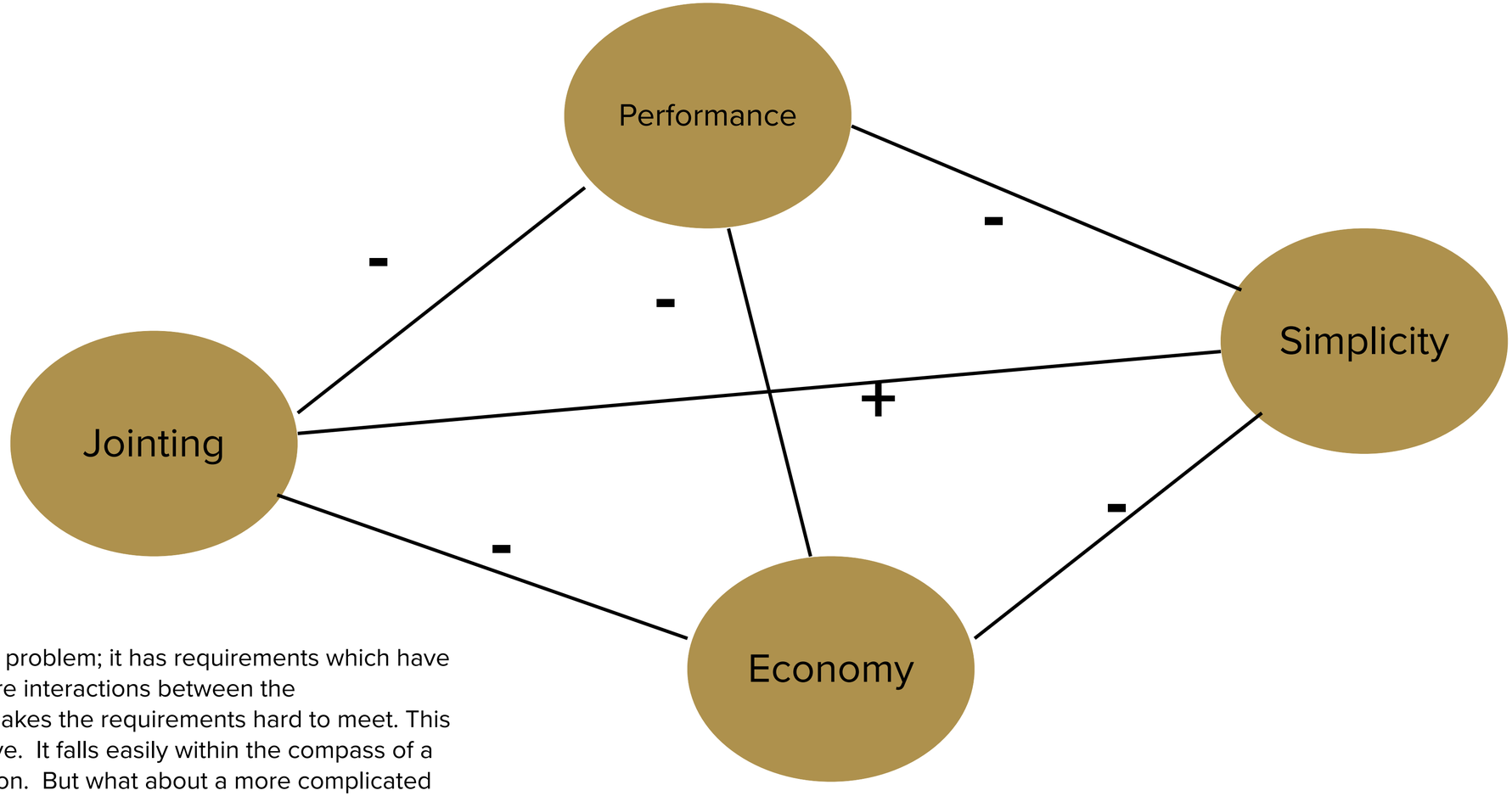
HOW CAN WE APPROPRIATELY EVOLVE
THE ARCHITECTURE
AS THE PRODUCT EVOLVES

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

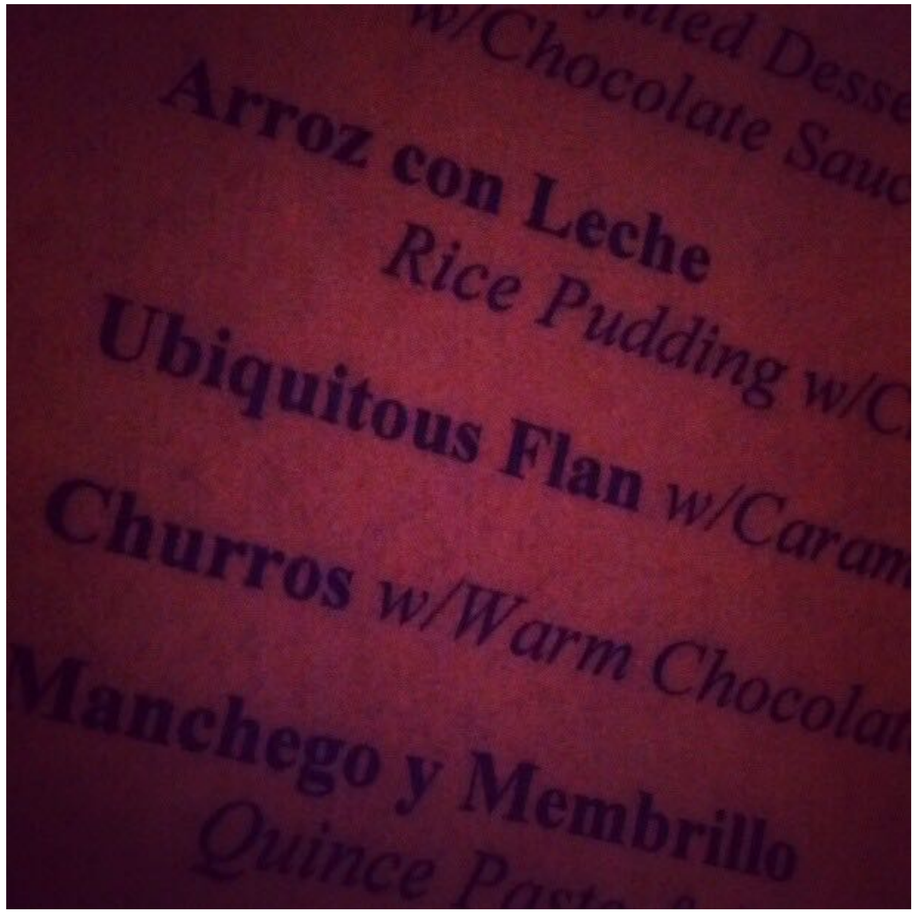
USING PRODUCT CONTEXT TO GROUND ARCHITECTURE

i.e.

INTENTIONAL VS ACCIDENTAL ARCHITECTURE

PROBLEM: TEAMS ARE STEPPING ON EACH OTHERS TOES

WHAT IS STRATEGIC DDD? WHY MIGHT WE CONSIDER IT?



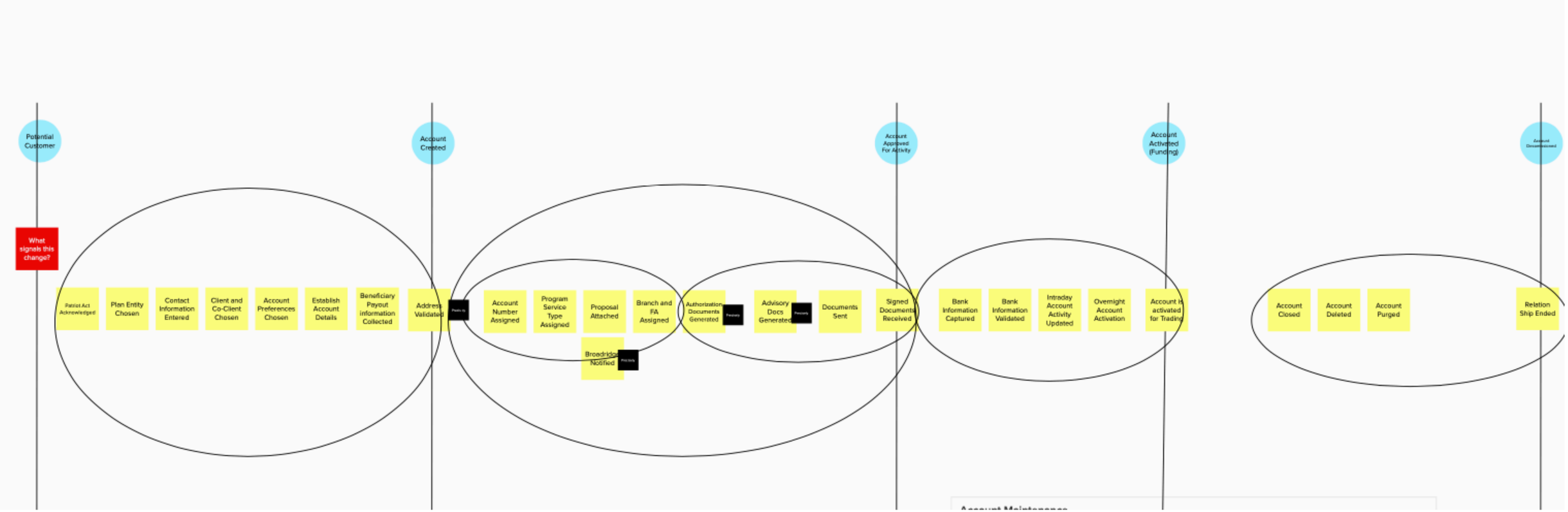
We are trying to model domains / teams closely to product intent

Strategic DDD is a technique for designing systems & boundaries

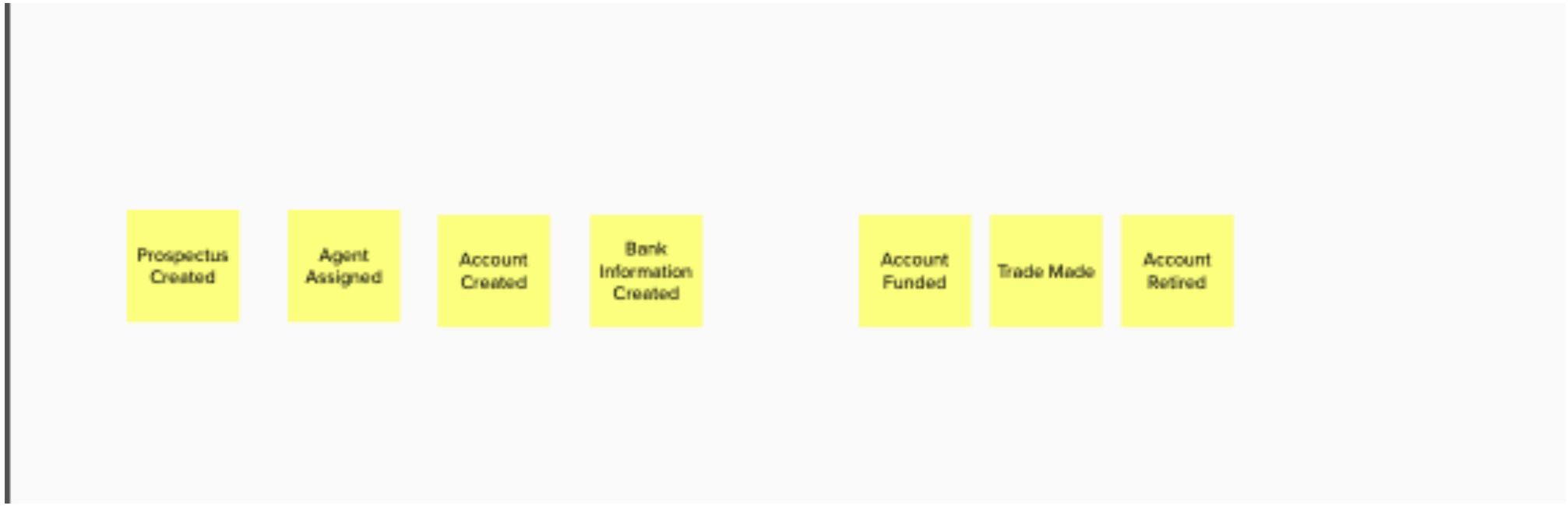
It is a model that helps us reason and keep product intent close

"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.

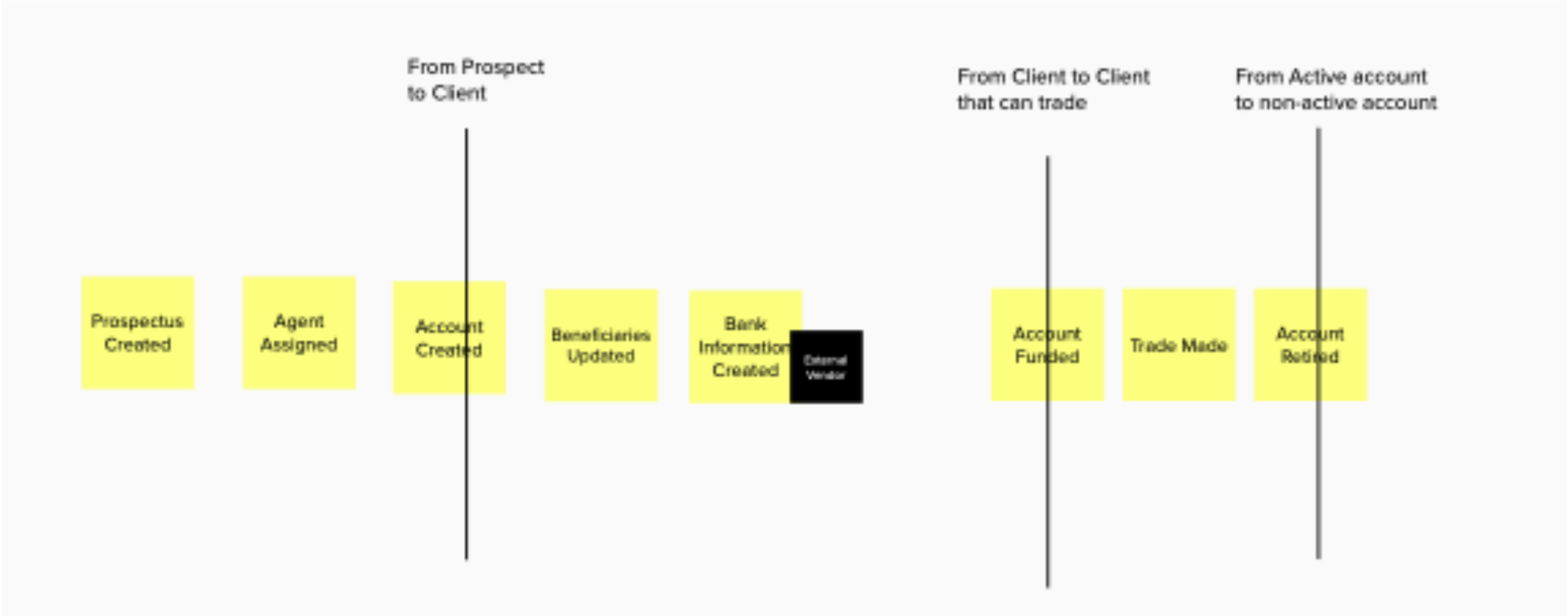
AN EXAMPLE


Potential Customer
Account Created
Account Approved For Activity
Account Activated (Funding)
What signals this change?
Patriot Act Acknowledged
Plan Entity Chosen
Contact Information Entered
Client and Co-Client Chosen
Account Preferences Chosen
Establish Account Details
Beneficiary Payout information Collected
Address Validated
Account Number Assigned
Program Service Type Assigned
Proposal Attached
Branch and FA Assigned
Broadridge Notified
Authorization Documents Generated
Advisory Docs Generated
Documents Sent
Signed Documents Received
Bank Information Captured
Bank Information Validated
Intraday Account Activity Updated
Overnight Account Activation
Account Closed
Account Deleted
Account Purged
Relation Ship Ended

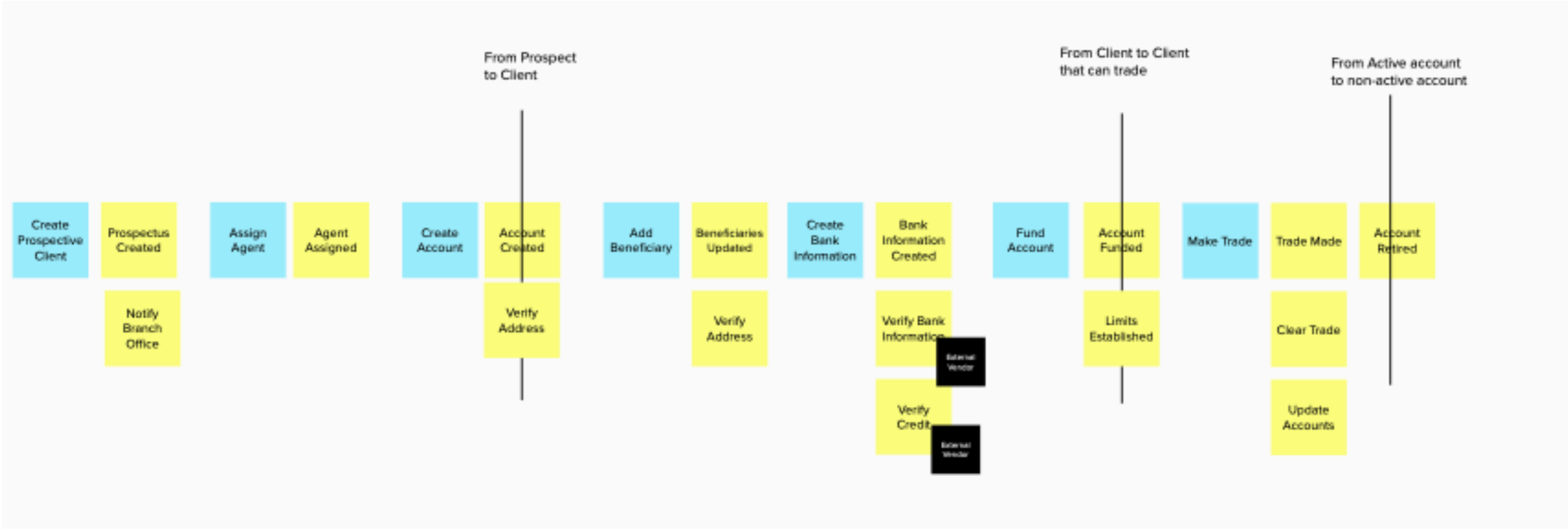
HOW DID WE GET THERE?



HOW DID WE GET THERE?

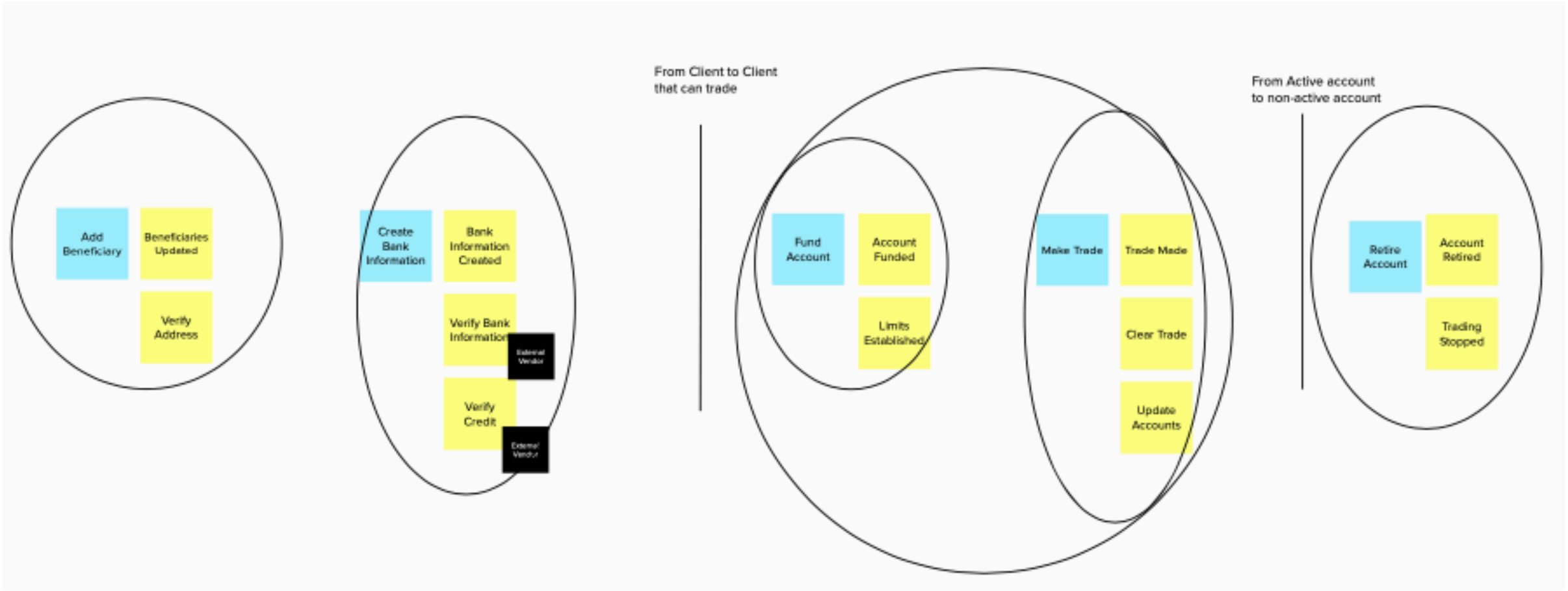


HOW DID WE GET THERE?

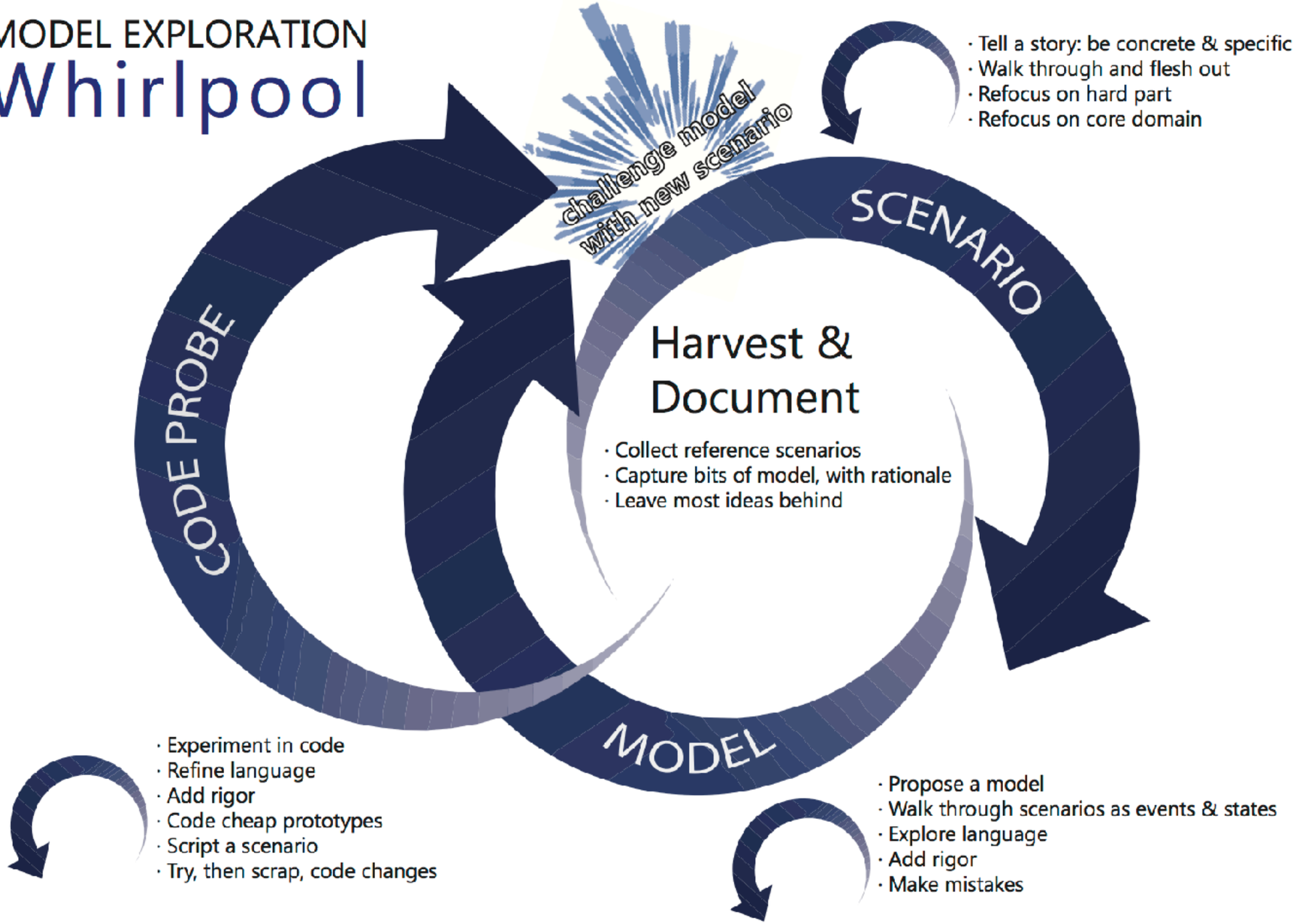


©2026 Dojo and Co

HOW DID WE GET THERE?



MODEL EXPLORATION
Whirlpool



CC BY SA

domainlanguage.com/ddd/whirpool

GETTING STARTED

"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.

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

UBIQUITOUS LANGUAGE AS A SIGNAL

"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.

Listen for words that start to have extra meaning

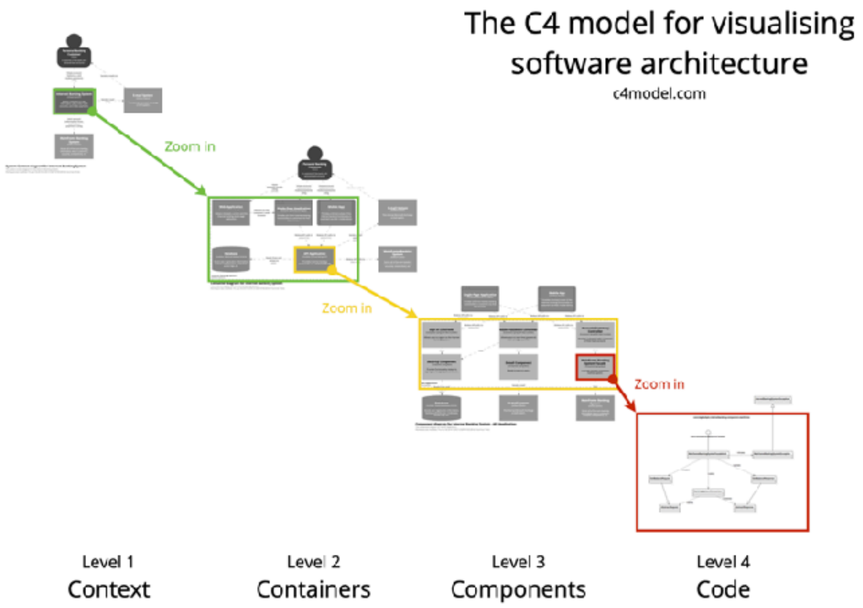
These could be contexts trying to break apart (and rightfully so)

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

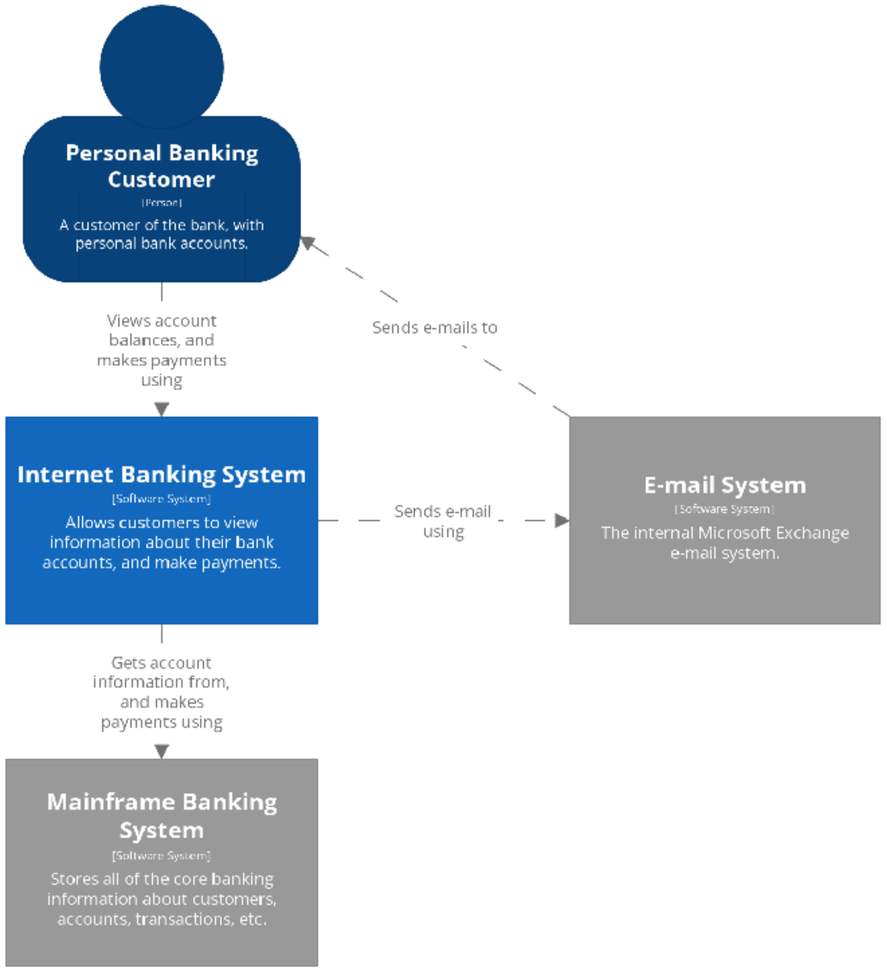
See how / where we fit; Boundaries

Provides context for interfaces

Can reduce accidental coupling

Socializes design

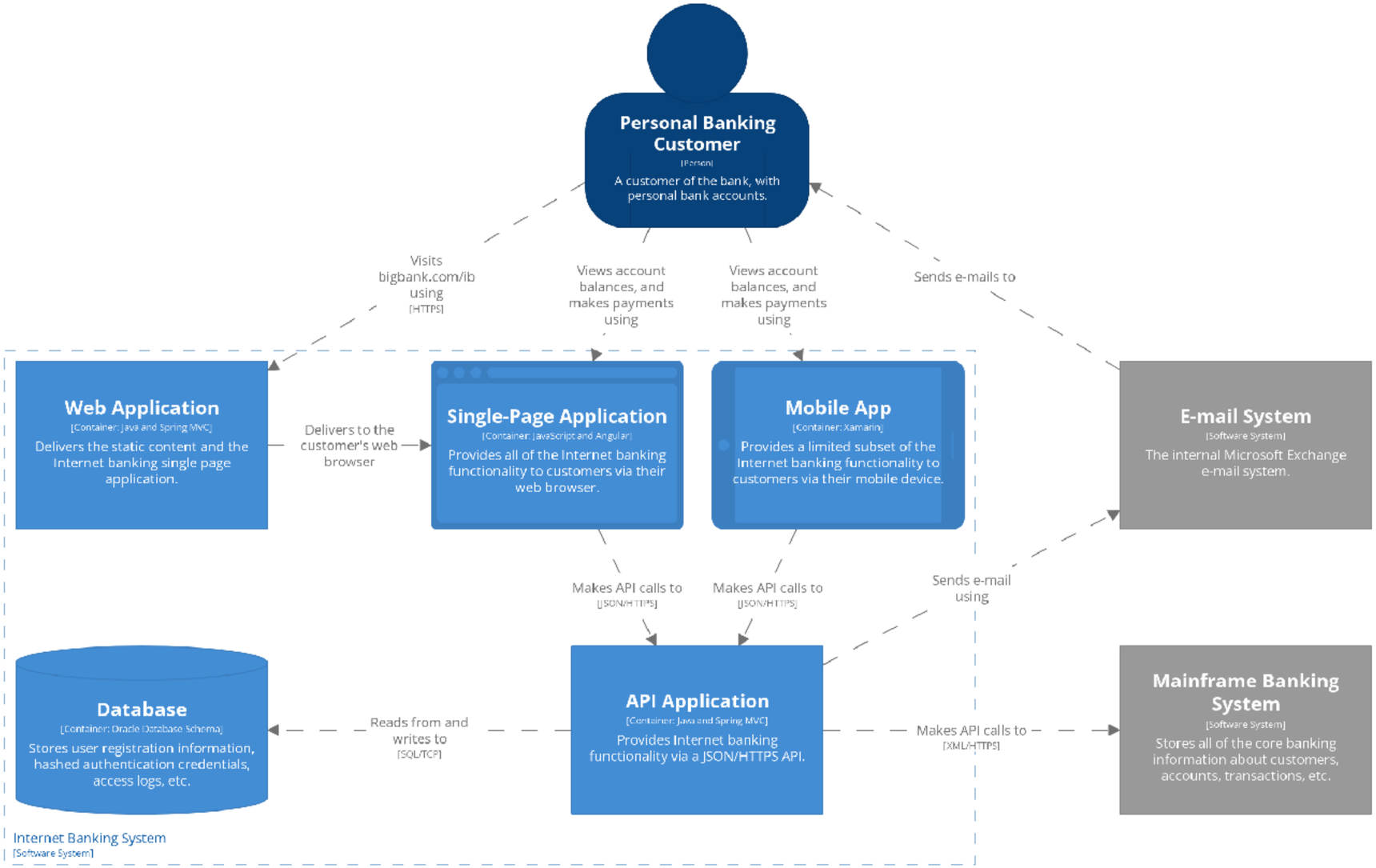
C1 - CONTEXT



[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

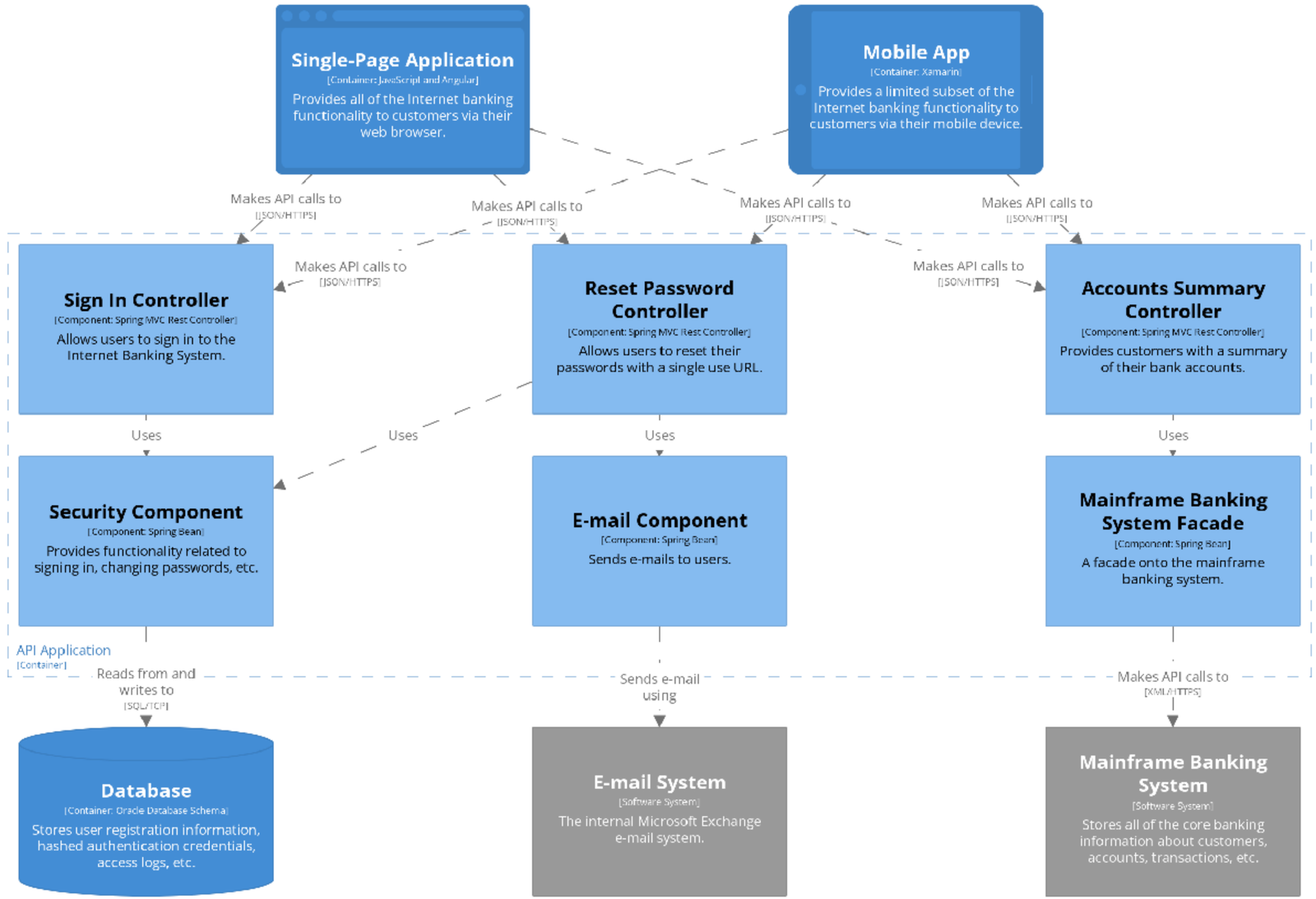
C2 - CONTAINERS



[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

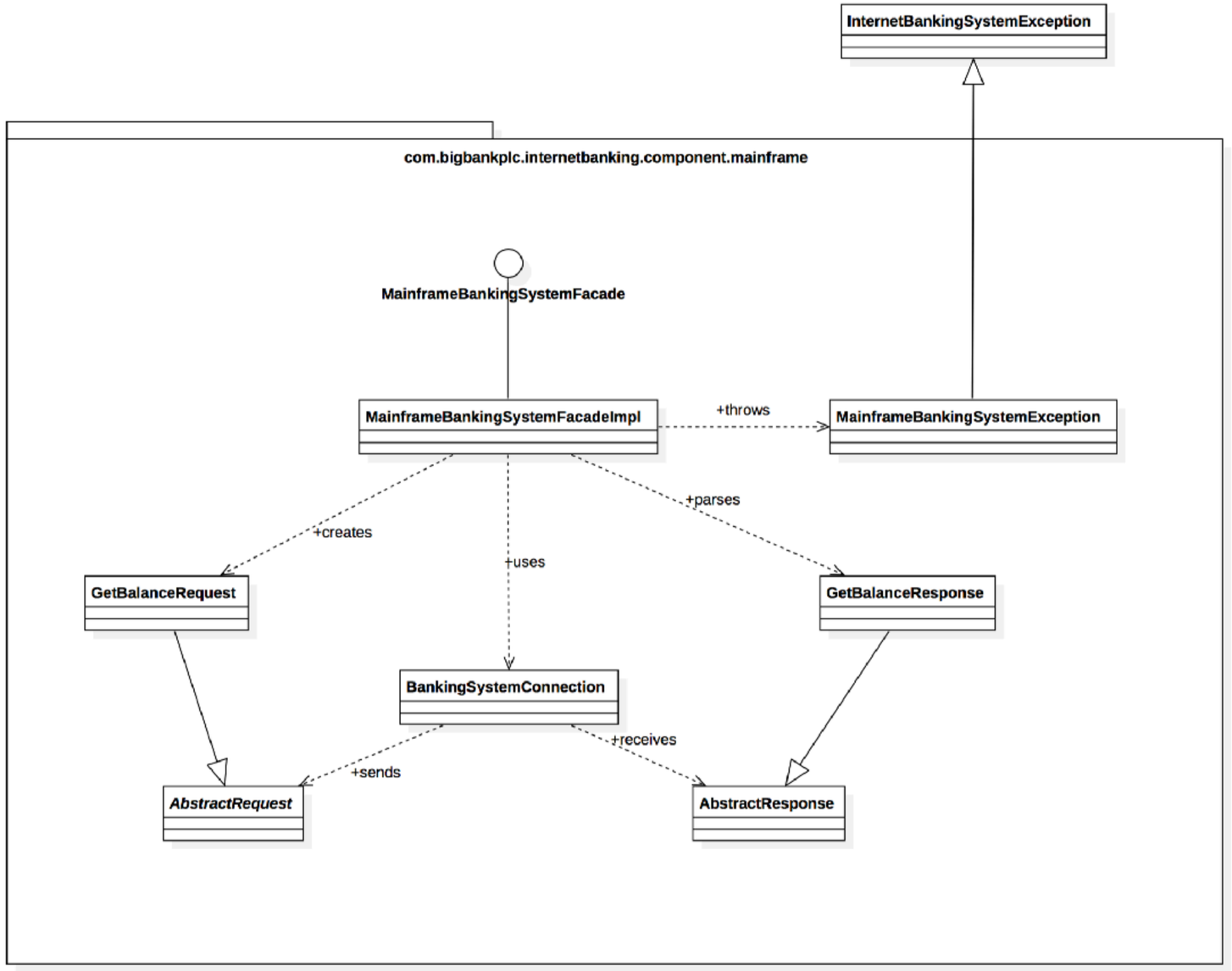
C3 - COMPONENTS



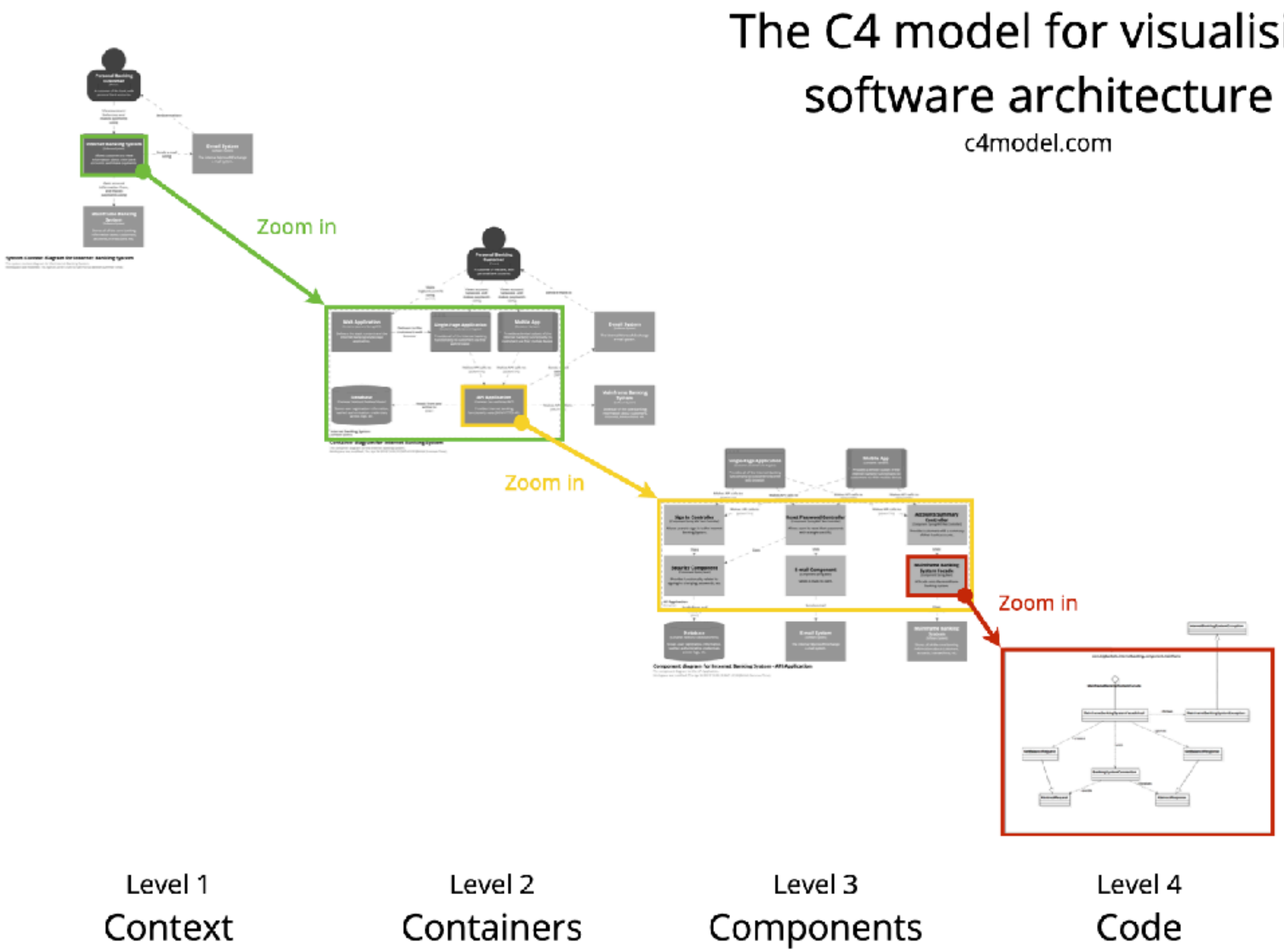
[Component] Internet Banking System - API Application

The component diagram for the API Application - diagram created with Structurizr.
Wednesday, March 22, 2023 at 8:16 AM Coordinated Universal Time

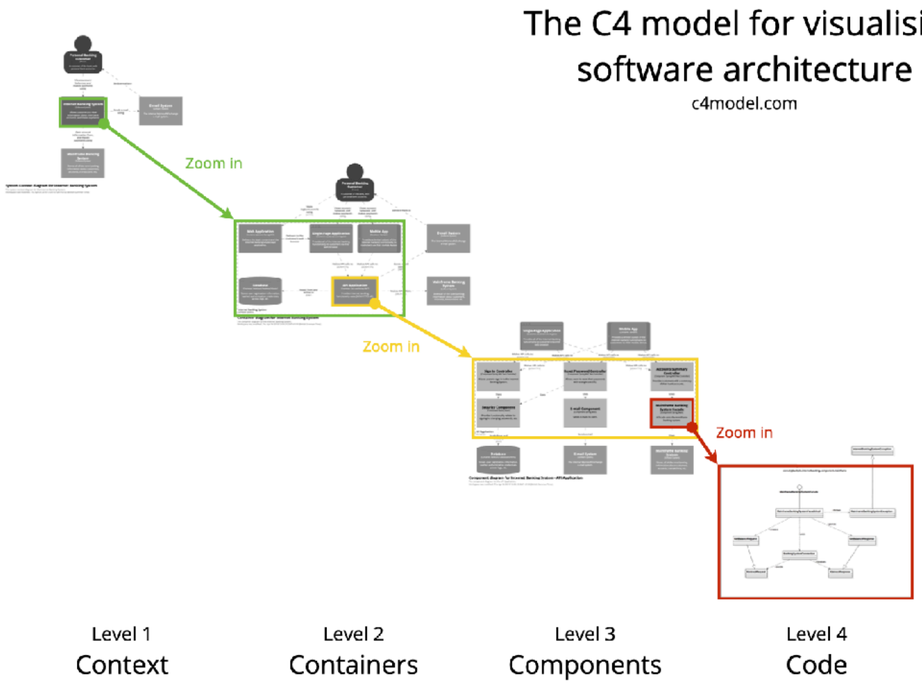
C4 - CODE / CLASS



TESTING!


The C4 model for visualising software architecture
c4model.com
Zoom in
Zoom in
Zoom in
Level 1
Context
Level 2
Containers
Level 3
Components
Level 4
Code

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...

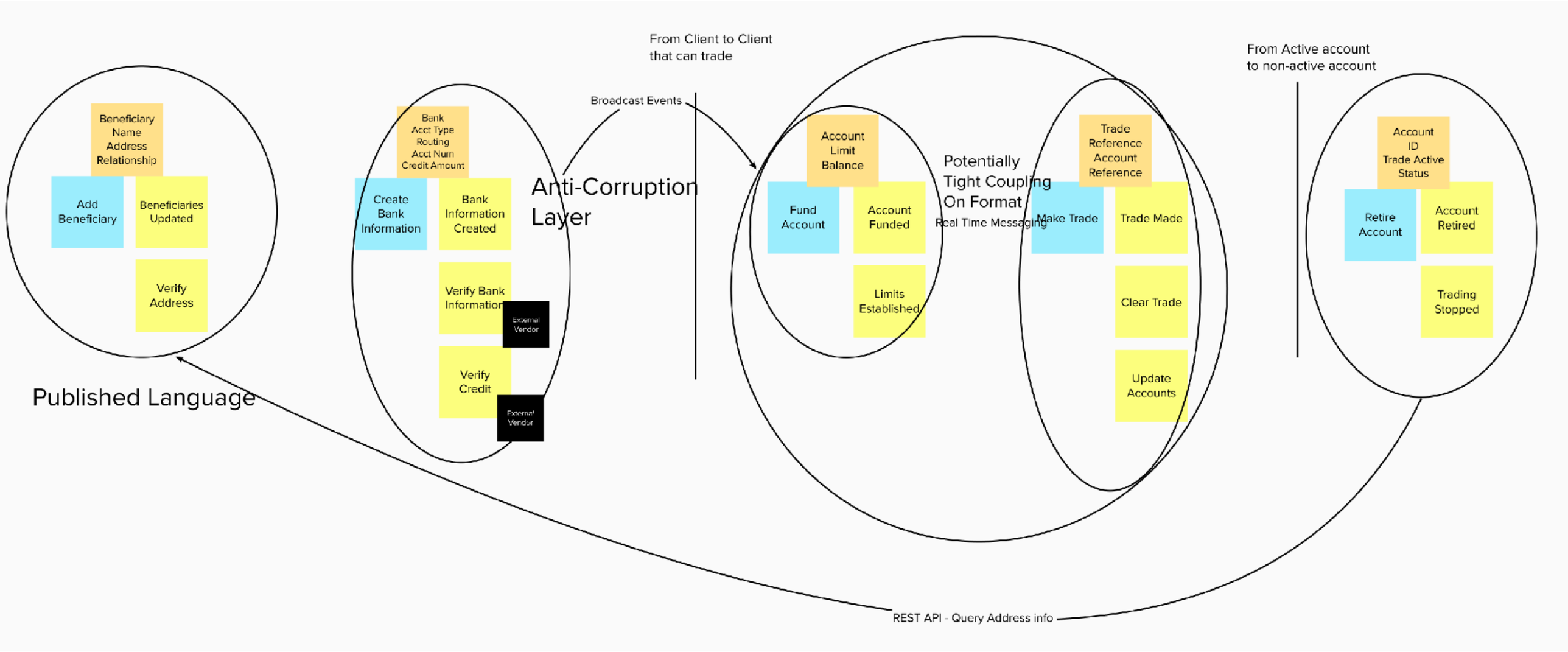
Cooperation (Partnership, Shared Kernel)
- High collaboration needed (but not necessarily high planning)

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


Beneficiary
Name
Address
Relationship
Add Beneficiary
Beneficiaries Updated
Verify Address
Published Language
Bank
Acct Type
Routing
Acct Num
Credit Amount
Create Bank Information
Bank Information Created
Verify Bank Information
External Vendor
Verify Credit
External Vendor
Anti-Corruption Layer
Broadcast Events
From Client to Client that can trade
Account
Limit
Balance
Fund Account
Account Funded
Limits Established
Potentially Tight Coupling On Format
Real Time Messaging
Trade Reference
Account
Reference
Make Trade
Trade Made
Clear Trade
Update Accounts
From Active account to non-active account
Account
ID
Trade Active
Status
Retire Account
Account Retired
Trading Stopped
REST API - Query Address info

USE CONTEXT MAPPING TO STOP ‘ALL TEAMS ARE AUTONOMOUS… JUST AS LONG AS THEY ALL FOLLOW THE SAME PROCESS…’

YOU WILL MAKE DECISIONS
CAPTURE THEM IN CONTEXT

ARCHITECTURE DECISION RECORDS

Title – In memory database for documents
Context – We have a series of highly static rules that determine what documents apply to an account. The current enterprise storage solution has had ripple effects on these documents accidentally.
Decision – Use in memory database.
Status – Approved
Last Updated Timestamp - Jan 1, 2026
Consequences – This decision makes testing easier and availability of data higher. If other teams / domains need the data, it will be harder for them to access. The team will have to manage security of the data as opposed to the enterprise solution. Scaling the service will have data duplication which may be costly if the amount of data grows substantially.

What is easier?
What is harder?

Team making the decision are the ones managing the ADR

ALWAYS START WITH A PROBLEM

NOT THE SOLUTION

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, ADRs, 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?

EST DOJO 2017
AND Co

Joel Tosi

SLIDES – https://www.dojoandco.com/speaking

@joeltosi@mastodon.social

Joel.Tosi@dojoandco.com