

BLENDING PRODUCT THINKING WITH ARCHITECTURE



Slides - <https://dojoandco.com/speaking>

BIG THANKS TO THE SPONSORS!

Gold Sponsors

neo



Community Supporters

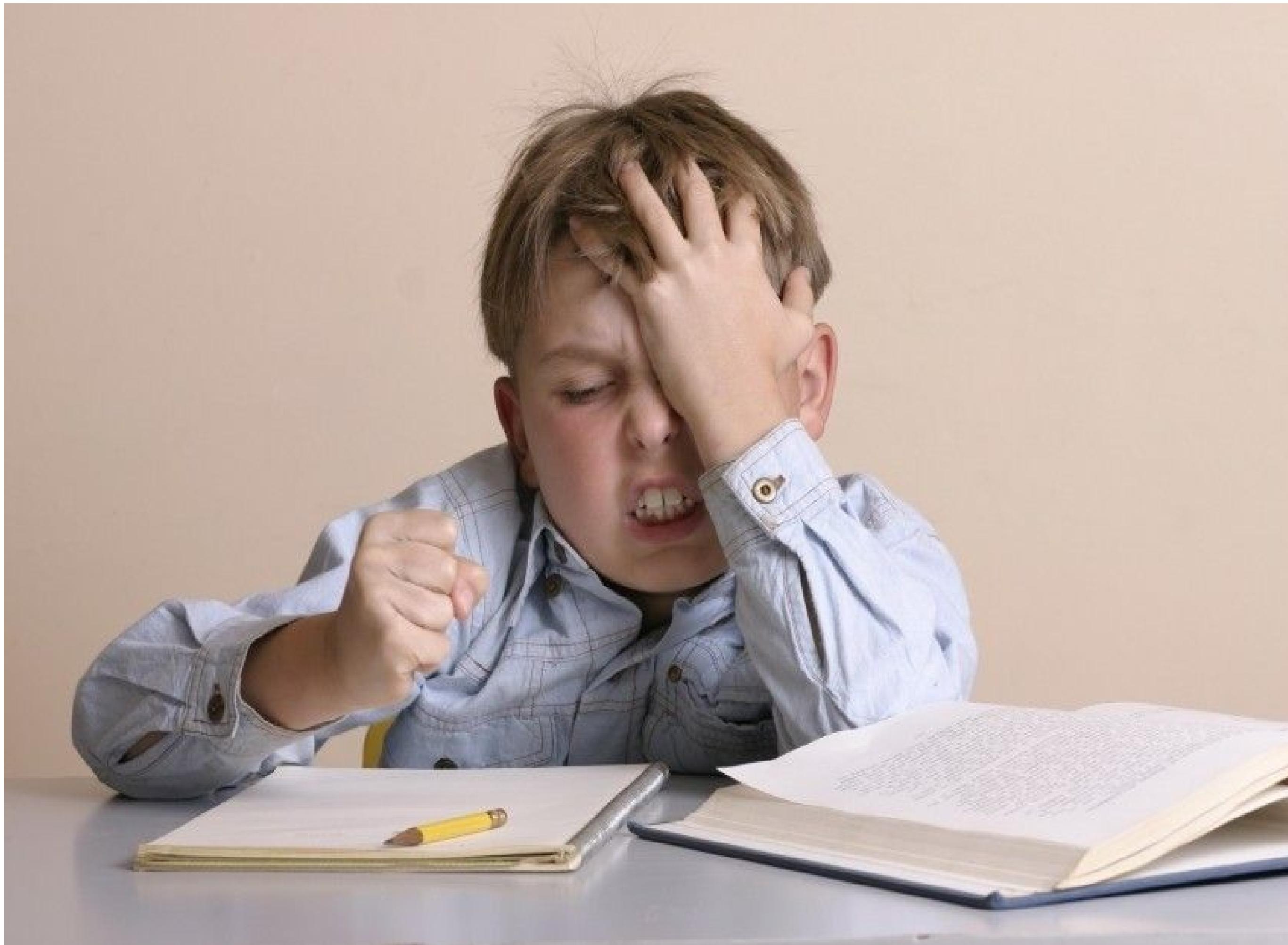


ABOUT ME



Joel Tosi

- Last 9 years – make it easier for people to learn and do great work at orgs – Product → Tech → Process
- Wrote books
- Past lives – Architect at Red Hat; Trading Platforms; Blah blah blah



EXPECTED OUTCOMES

Simple Visuals You Can Use Now

Less ‘What’ Architecture, More
‘When’

Leverage Product Context to
Ground Decisions

Ideas on Documenting Decisions

History

CONTEXT AND THE WIKI



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

BEHIND EVERY ARCHITECTURE DECISION
IS A PRODUCT ASSUMPTION

WHAT IS THE PROBLEM?

What Is Architecture?

ONE VIEW



Web

API

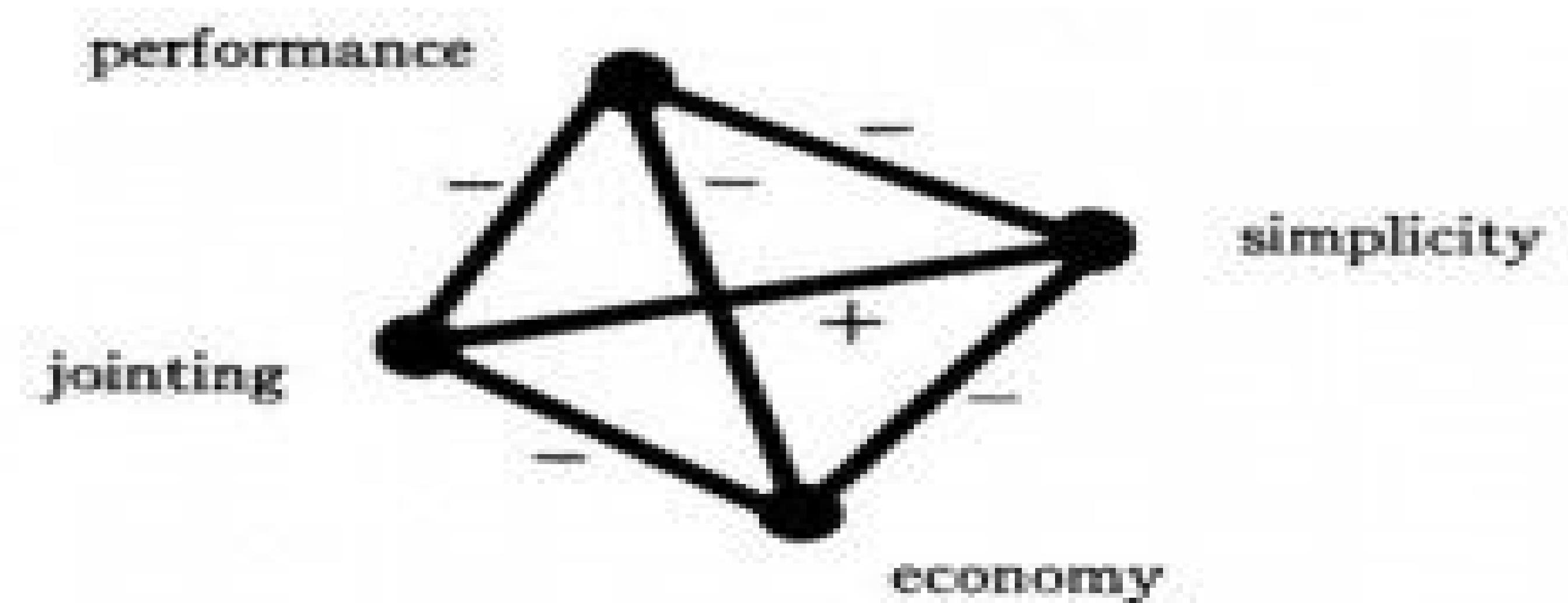
Logic / Calculations

Fast Data

Slow Data

What Is Architecture?

ANOTHER VIEW

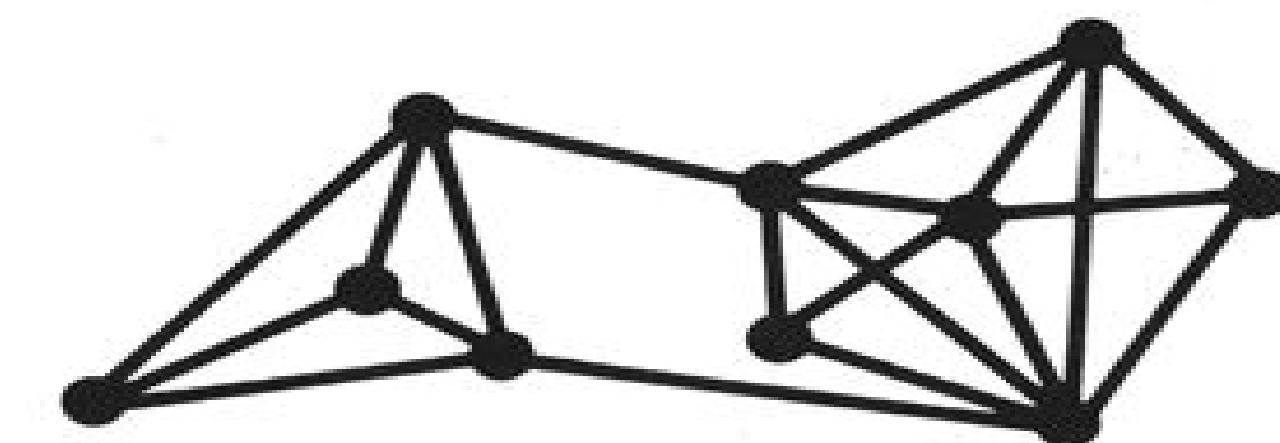


This is a typical design problem; it has requirements which have to be met; and there are interactions between the requirements, which makes the requirements hard to meet. This problem is simple to solve. It falls easily within the compass of a single man's intuition. But what about a more complicated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

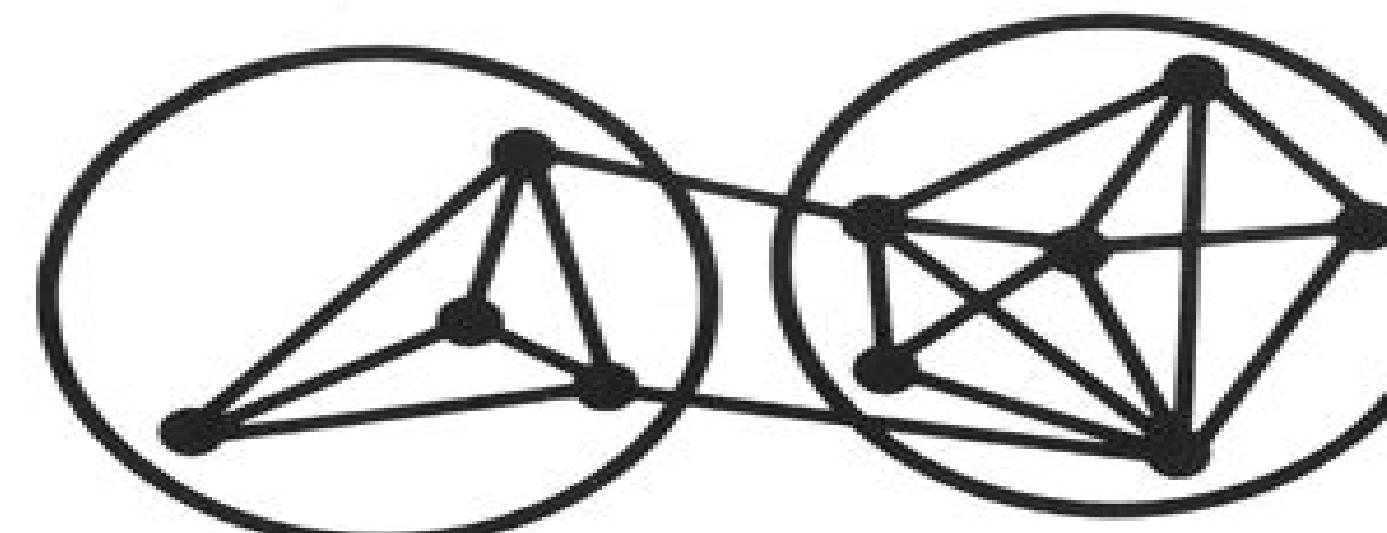
© Notes on the Synthesis of Form – Christopher Alexander

What Is Architecture?

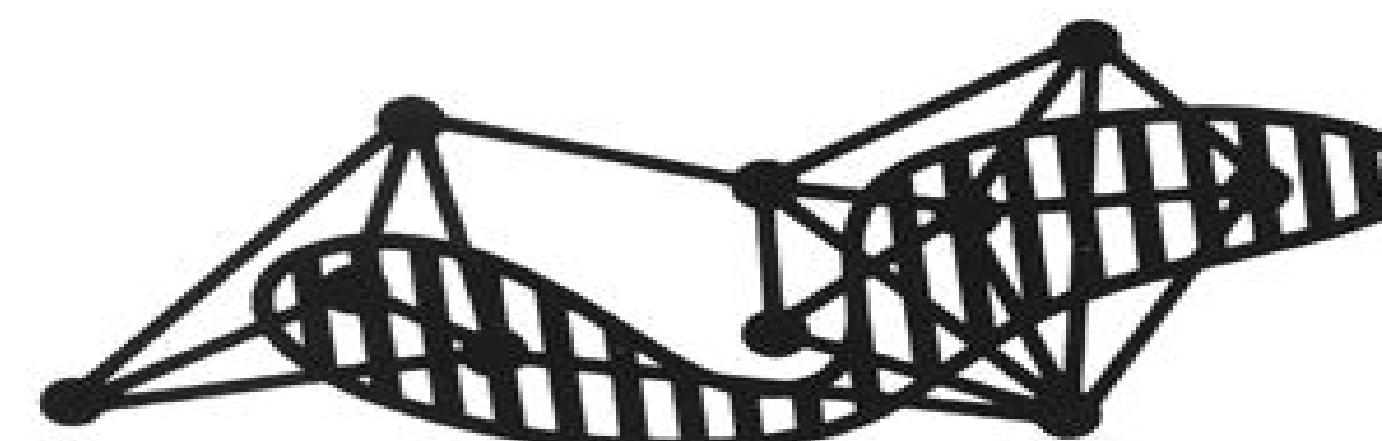
ANOTHER VIEW



Reflection of System Dependencies



Optimal Decomposition / Grouping



Biased Simplification - mereotopology

© Notes on the Synthesis of Form – Christopher Alexander

ARCHITECTURE IS ABOUT FIT
WITHIN CONTEXT

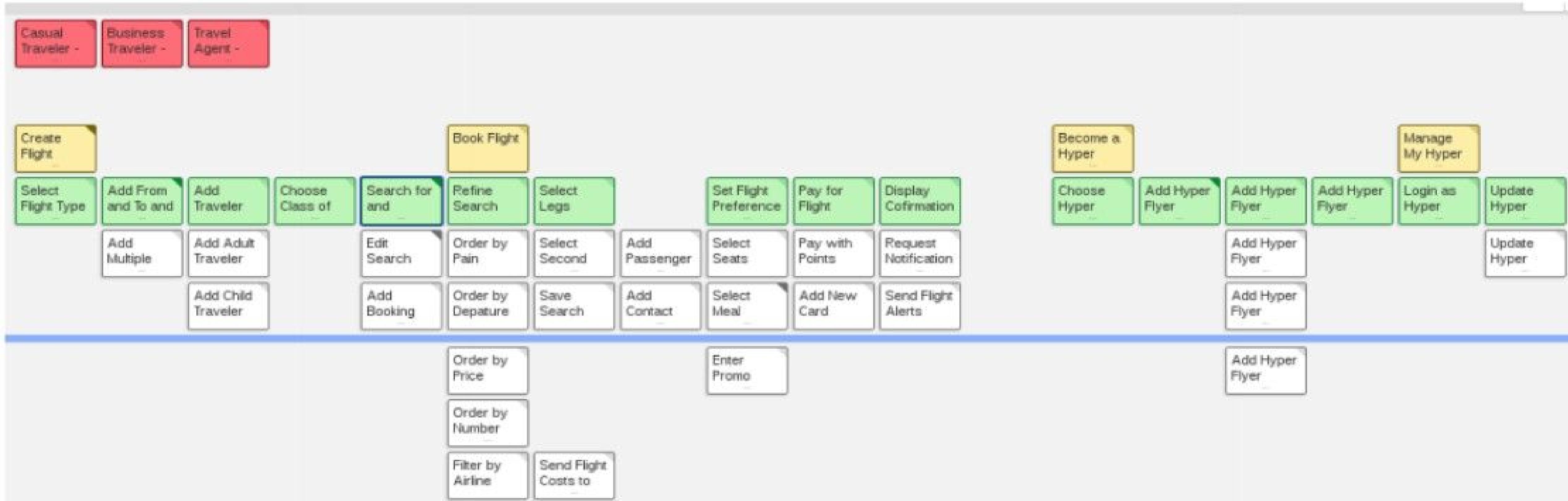
USING PRODUCT TO GROUND ARCHITECTURE

YOU ARE ALL CERTIFIED



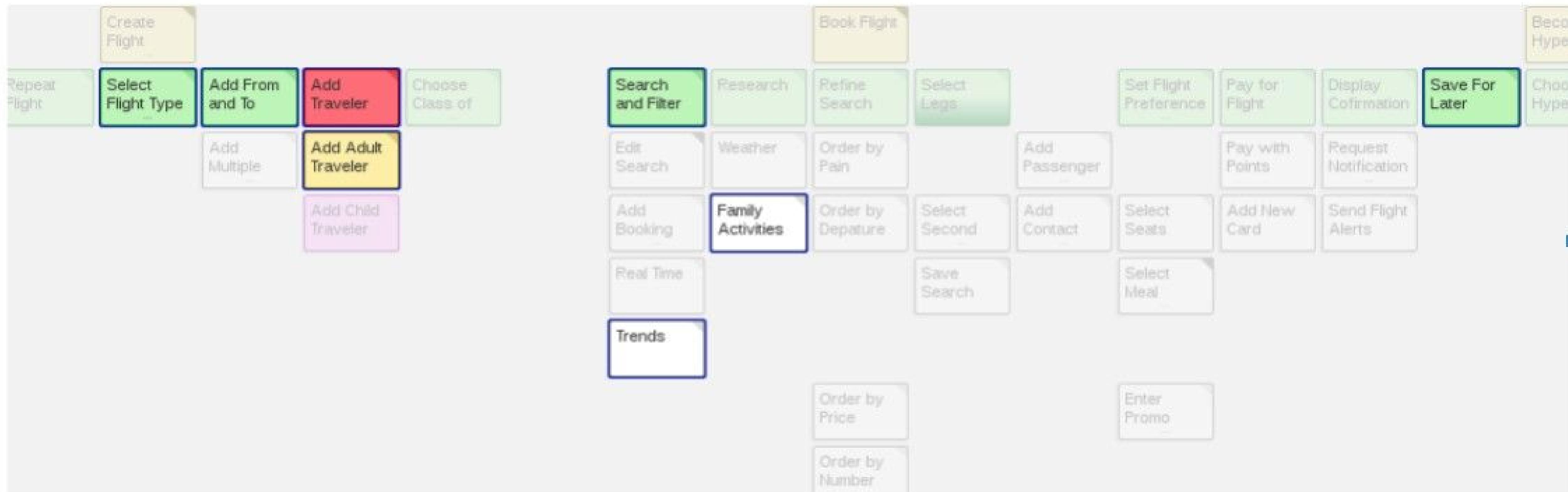
Visuals To Use

ALL THE POSSIBILITIES



Providing Clarity

FAMILY TRAVELING



Providing Clarity

BUSINESS TRAVELER

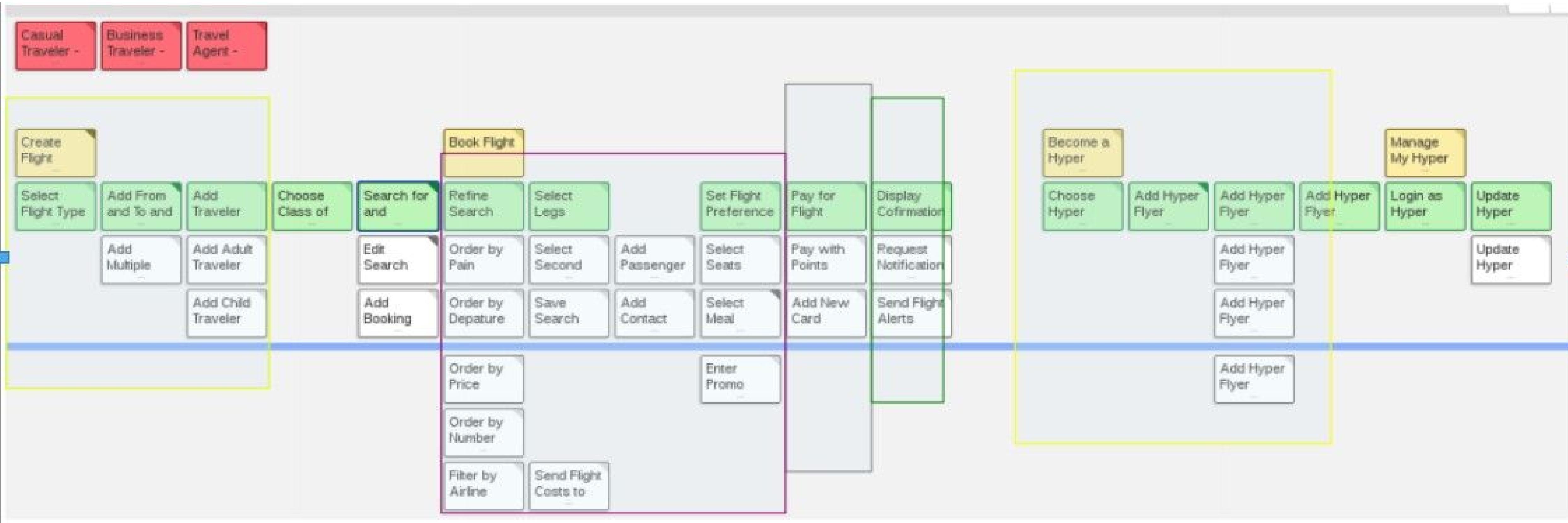


Providing Clarity

ADVENTURER

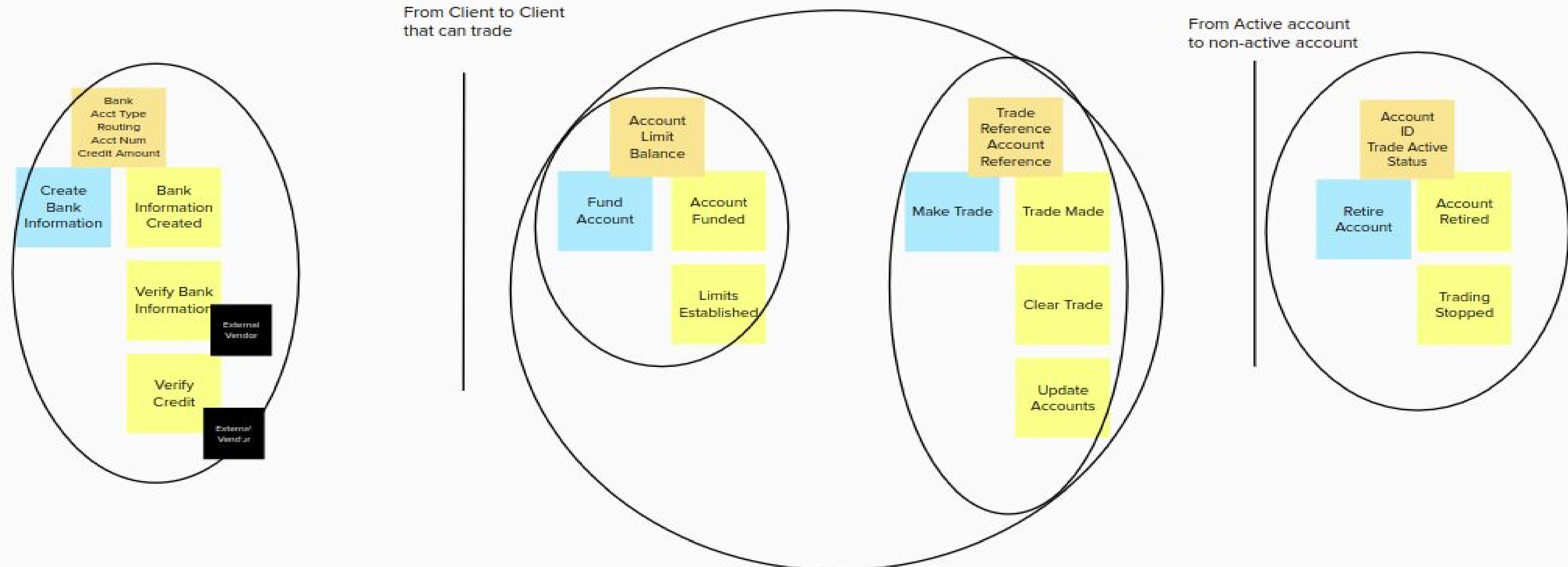


Simulating Architecture MICROSERVICES

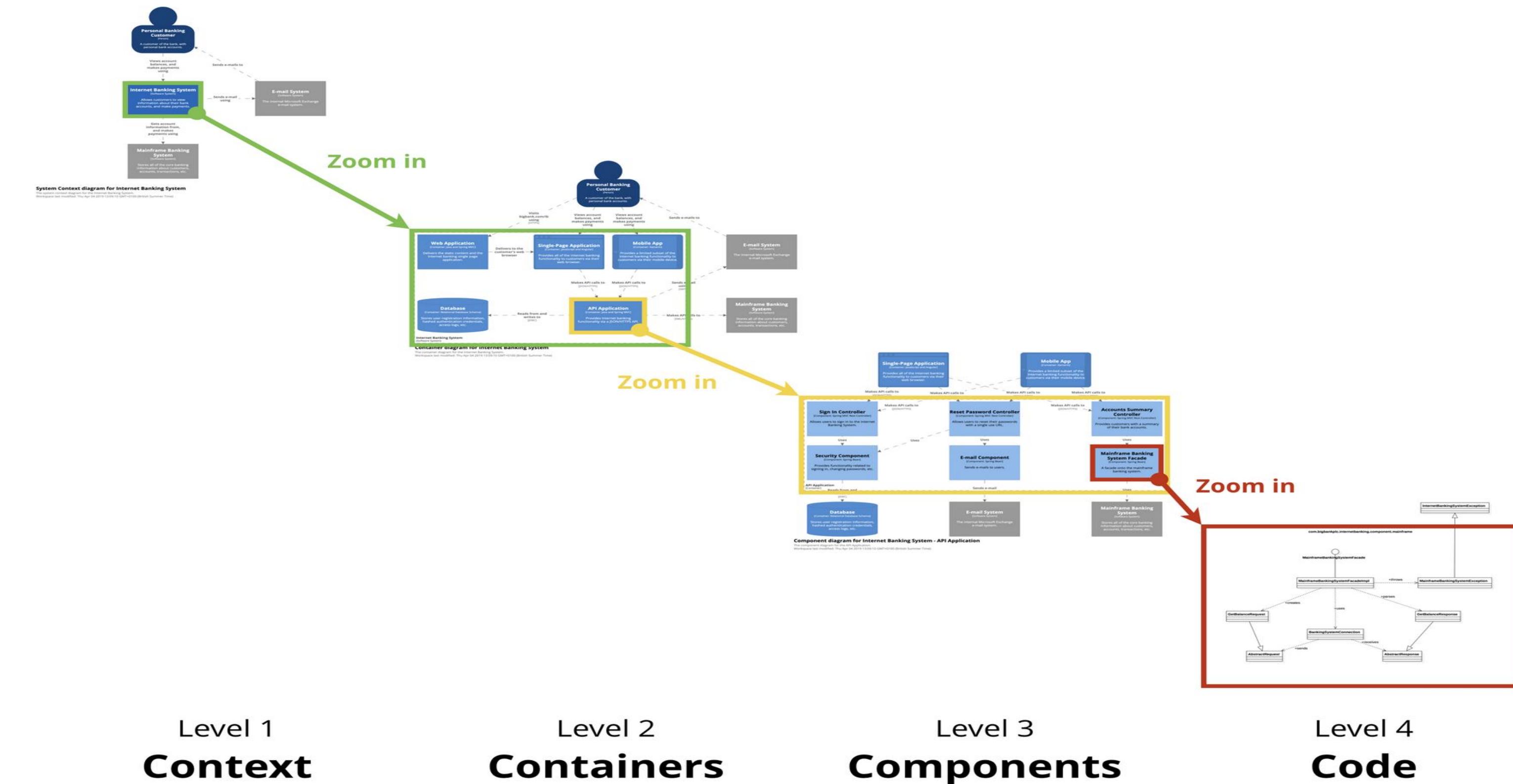


Providing Clarity

CREATING TEAMS AND BOUNDARIES



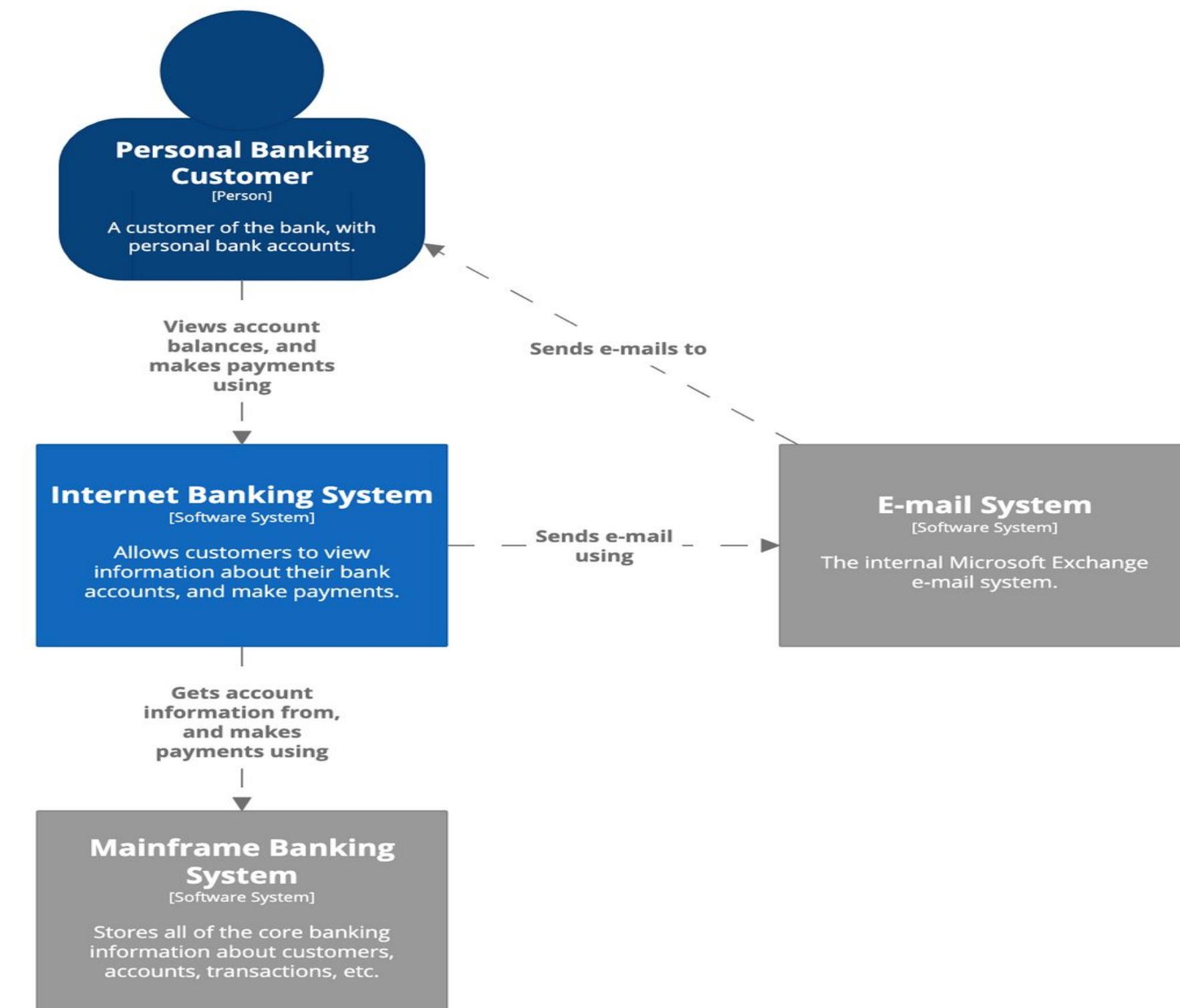
Another Visual DOESN'T FEEL ARCHITECTY ENOUGH



<https://c4model.com/>

Another Visual

MORE ARCHITECTY

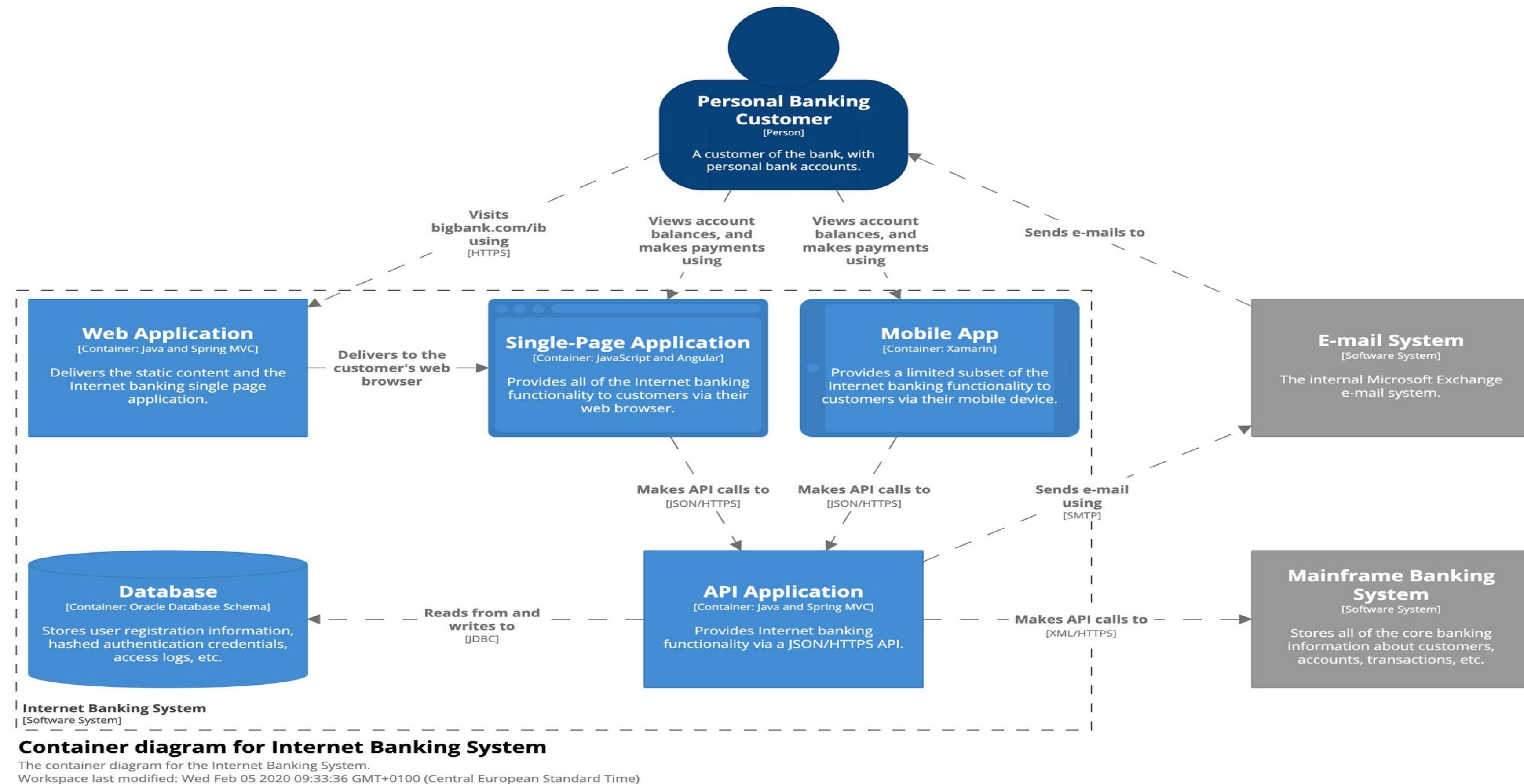


System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.
Workspace last modified: Wed Feb 05 2020 09:33:36 GMT+0100 (Central European Standard Time)

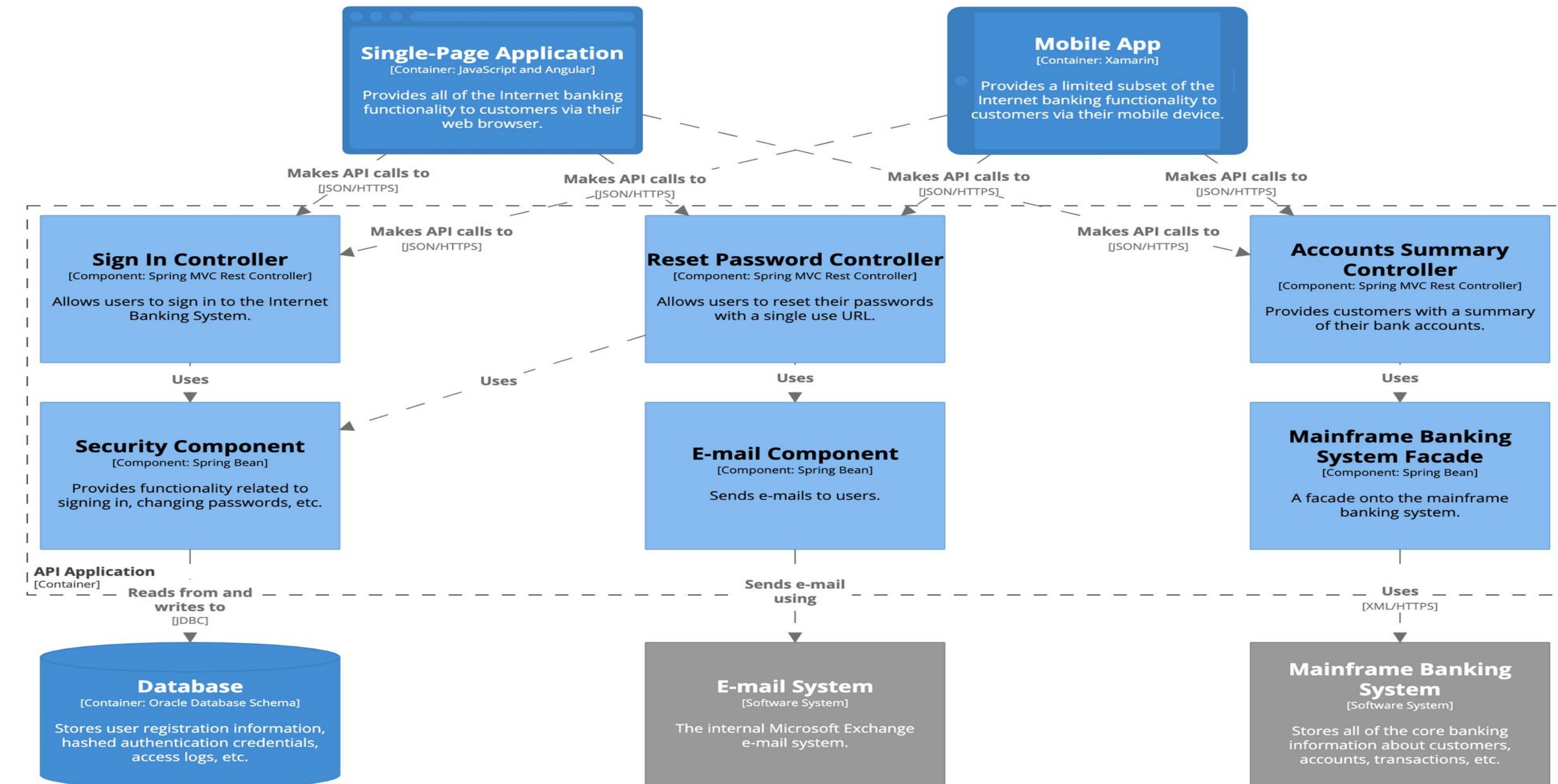
<https://c4model.com/>

Another Visual MORE ARCHITECTY



<https://c4model.com/>

Another Visual MORE ARCHITECTY



<https://c4model.com/>

STICK FIGURES AND BOXES IS
NOT YOUR JOB

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

KNOW WHERE YOU ARE

Agree

CONTEXT SHIFTS FOCUS

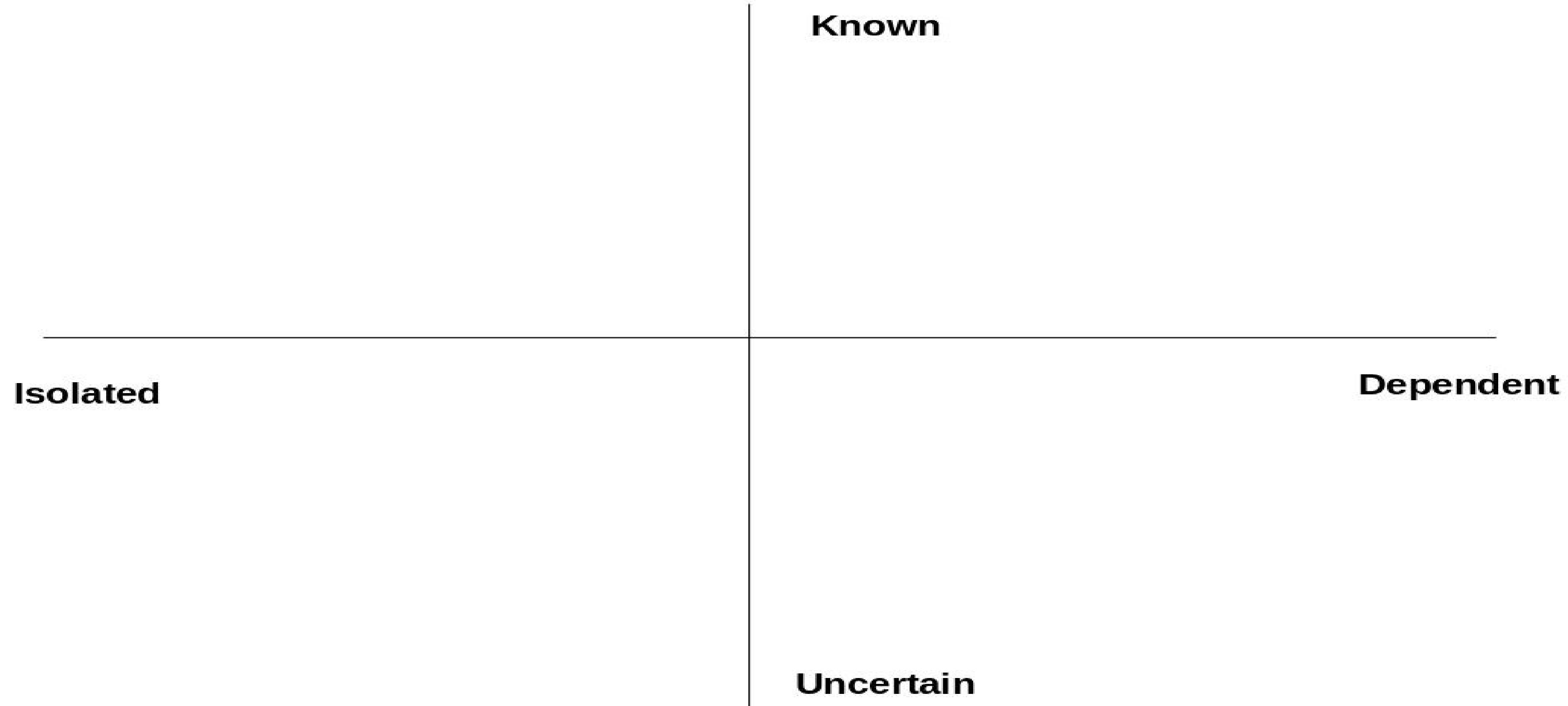
EXPLORE EXPAND EXTRACT

Credit – Kent Beck

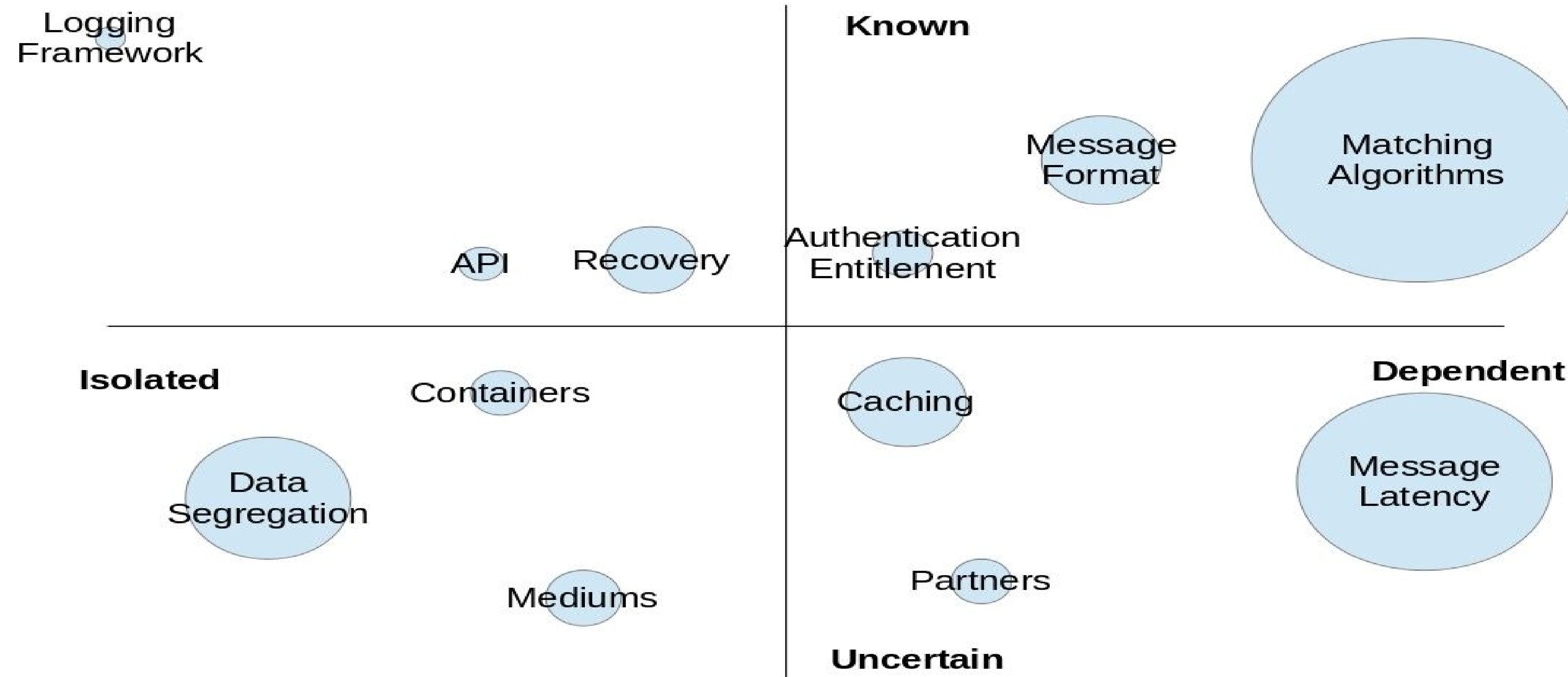
EXPLORE WHAT TO MOVE

Another Visual

DETERMINING DECISION POINTS



DETERMINING DECISION POINTS



DOCUMENTING THE ASSUMPTIONS IN YOUR ARCHITECTURE

Don't Forget Ward

ARCHITECTURE DECISION RECORDS

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

<> Raw Blame

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.

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

On Being Wrong

TRY THIS

Document Product Assumptions You Are Making

Document Technical Assumptions You Are Making

Think of 3 Ways You Might Be Wrong

For Each Learning Opportunity, What Would Change?

OUR JOB...
TAKE A WORLD OF POSSIBILITIES...

APPLY PROPER PRODUCT UNDERSTANDING &
CONTEXT...

TO DELIVER MANEUVERABLE SYSTEMS
THAT BRING JOY

Where We Have Been

RECAP

Be aware of models and dependencies

Use Product Discussions to avoid architecture bias

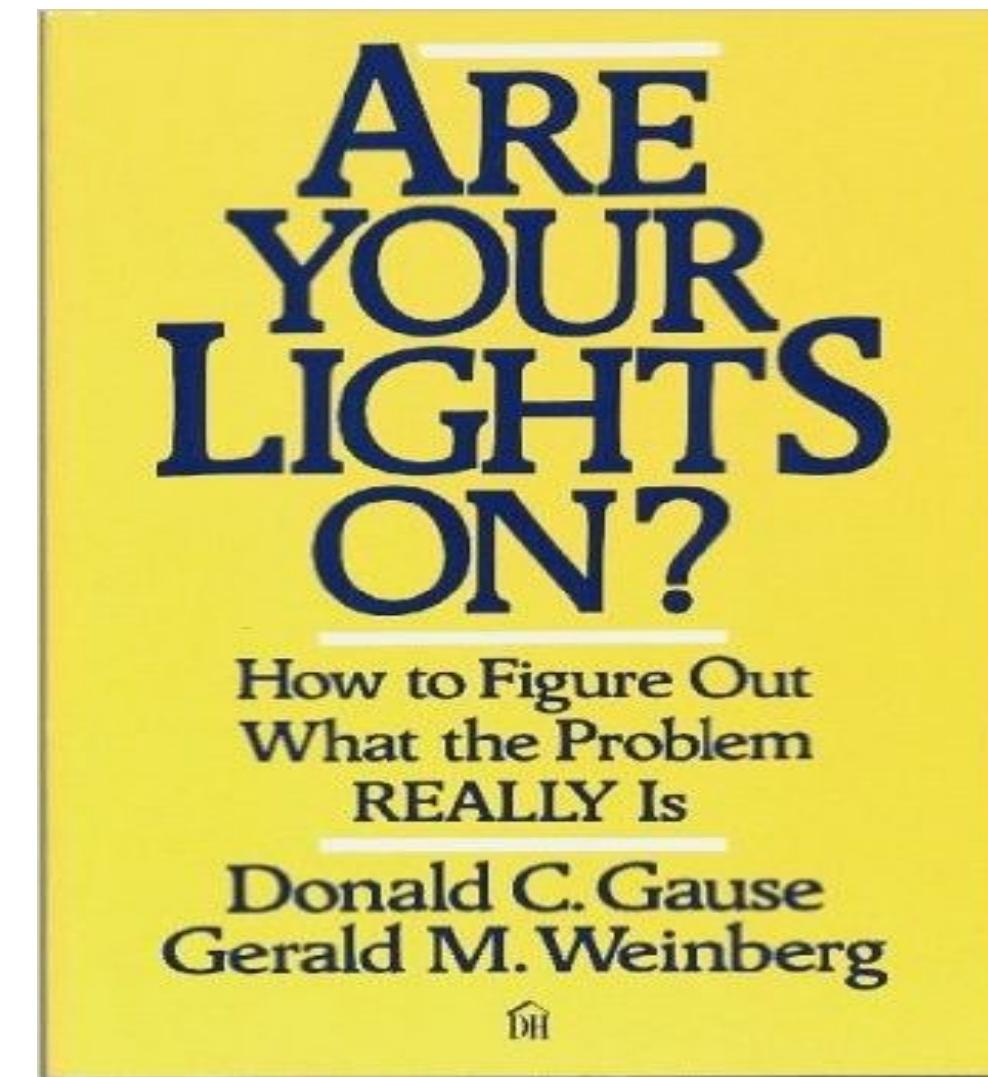
Understand parts of the system need to be right / emerge

Appreciate Technical Debt, Refactor Our Models

RECOMMENDED READING

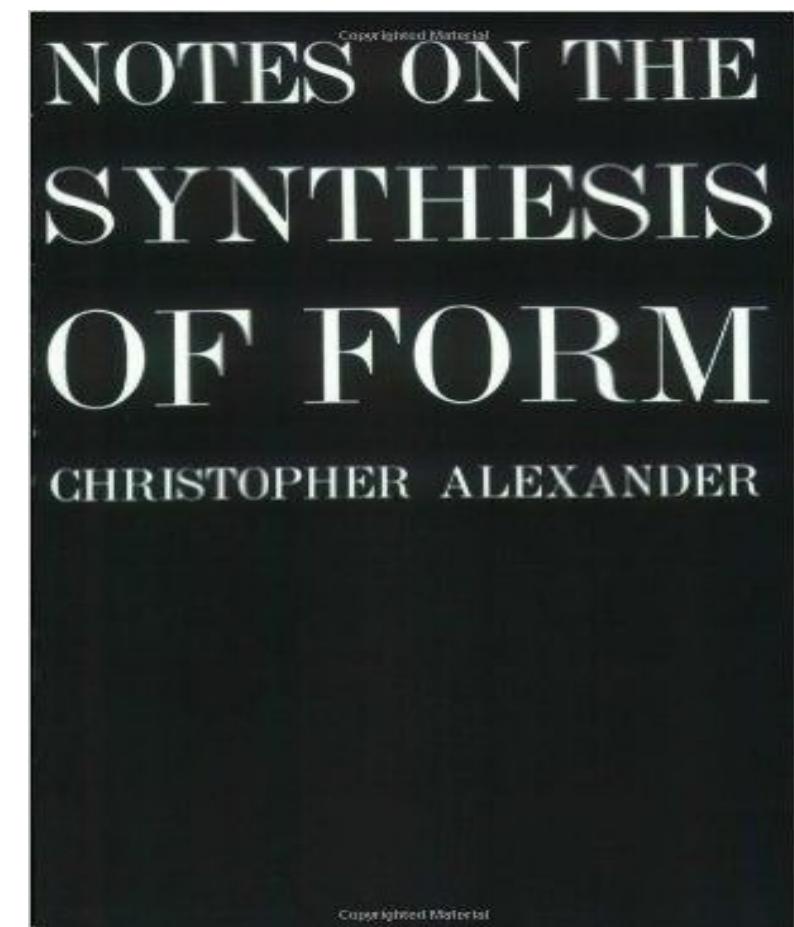
Jerry Weinberg

Are Your Lights On?



Christopher Alexander

Notes on the Synthesis of Form



WHAT ARE YOUR QUESTIONS?

BLENDING PRODUCT THINKING WITH ARCHITETCURE



Slides - <https://dojoandco.com/speaking>