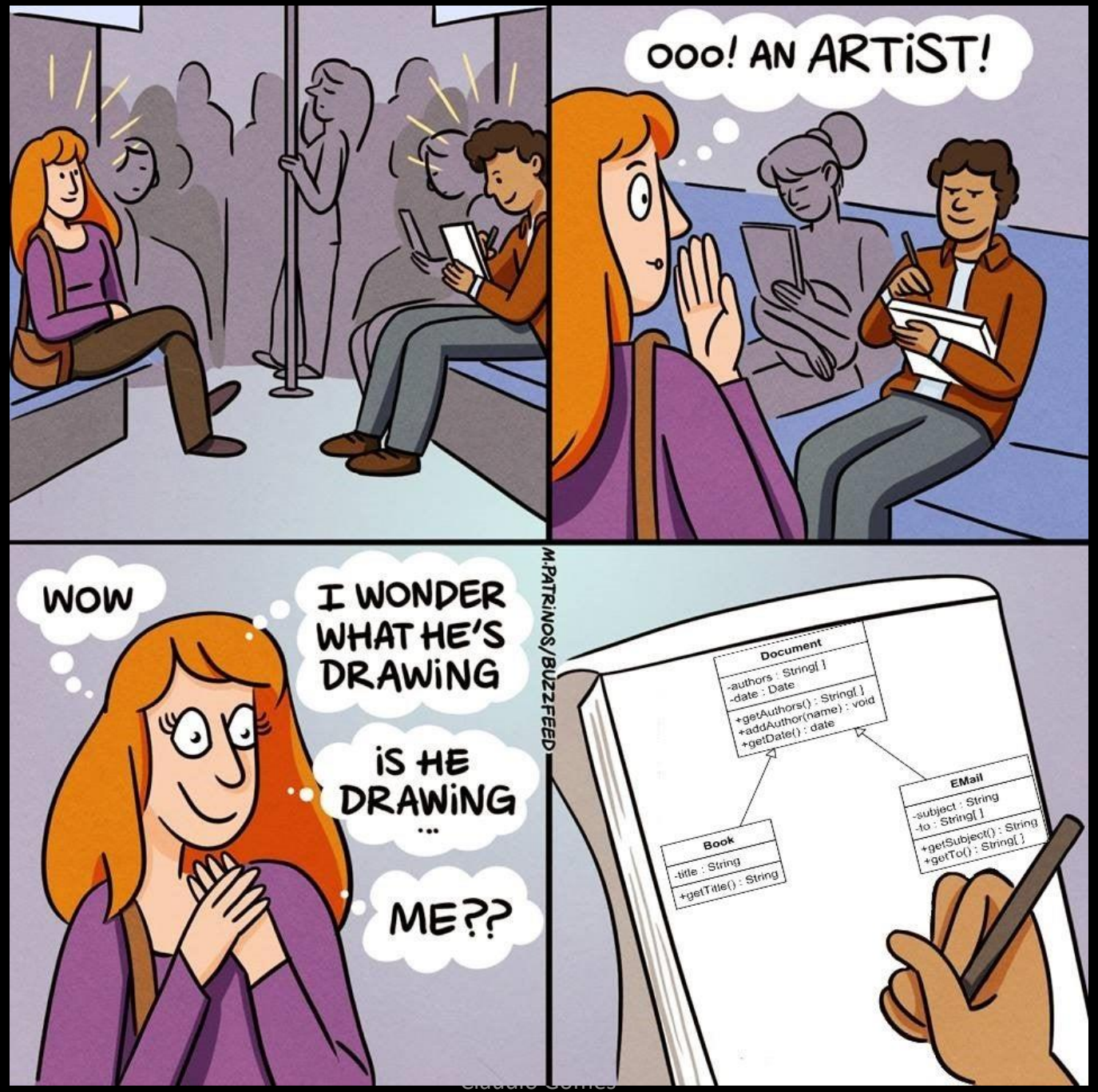

OOO! AN ARTiST!
WOW
I WONDER WHAT HE'S DRAWiNG
iS HE DRAWiNG ...
ME??
M.PATRINOS/BUZZFEED
Document
-authors : String[ ]
-date : Date
+getAuthors() : String[ ]
+addAuthor(name) : void
+getDate() : date
Book
-title : String
+getTitle() : String
EMail
-subject : String
-to : String[ ]
+getSubject() : String
+getTo() : String[ ]

Software architecture

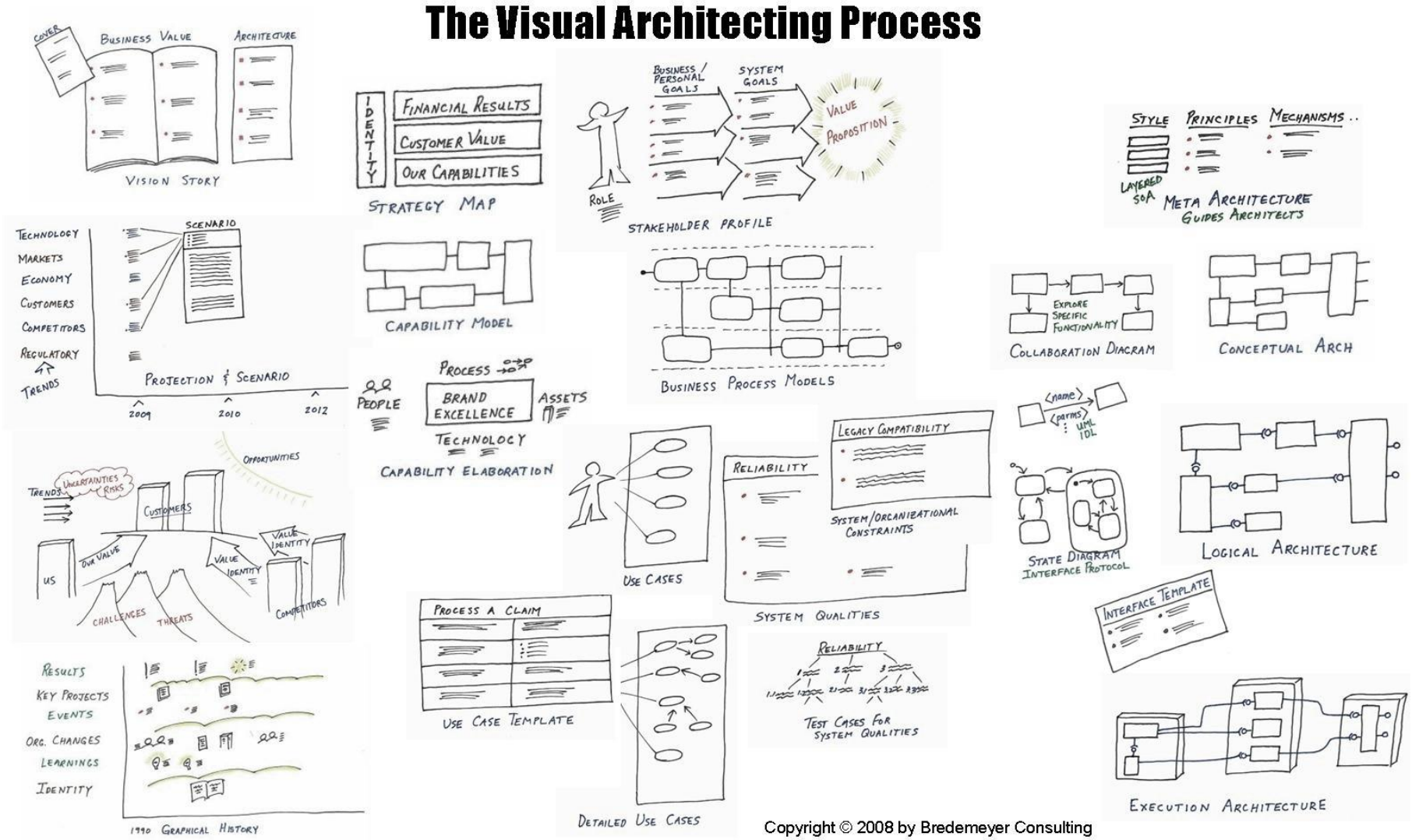
documentation

version: 1.0.1

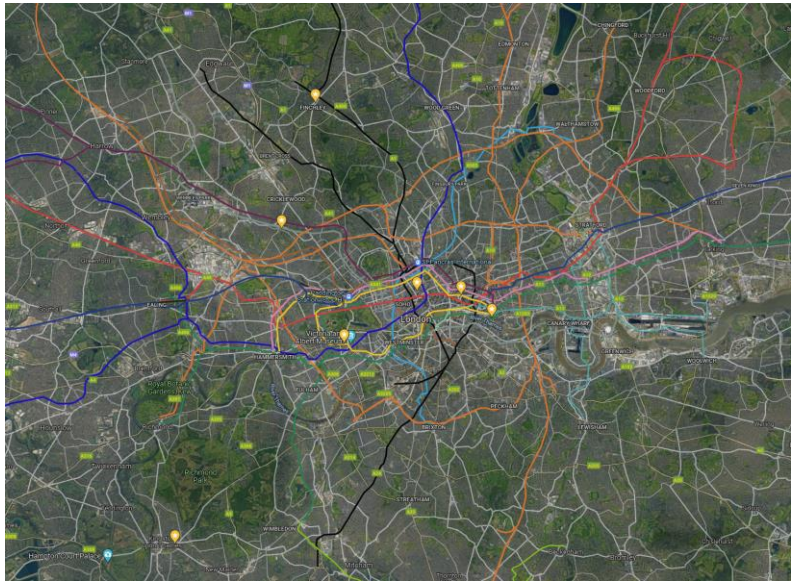
Agenda

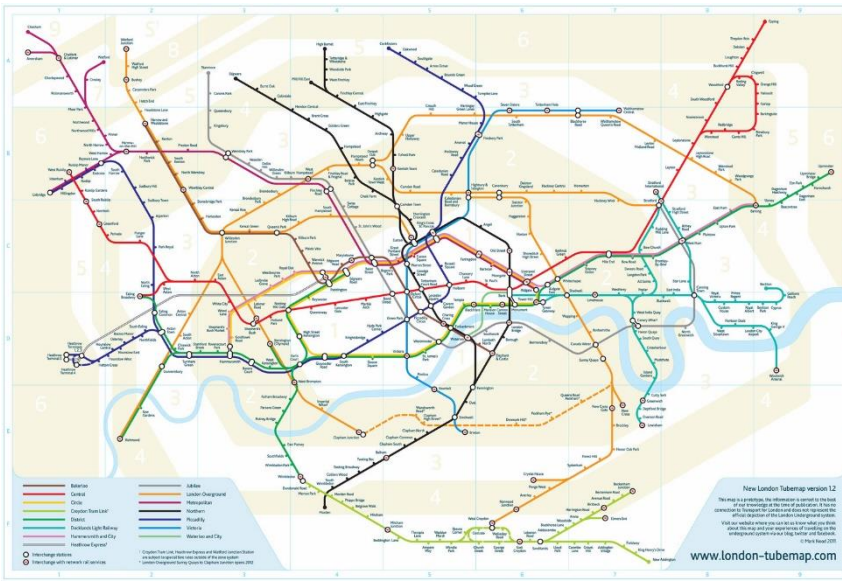
- Software architecture documentation
  - Modelling a system
  - Simon Browns C4 model
- Architecture process
  - Recap
  - Proposal for next steps

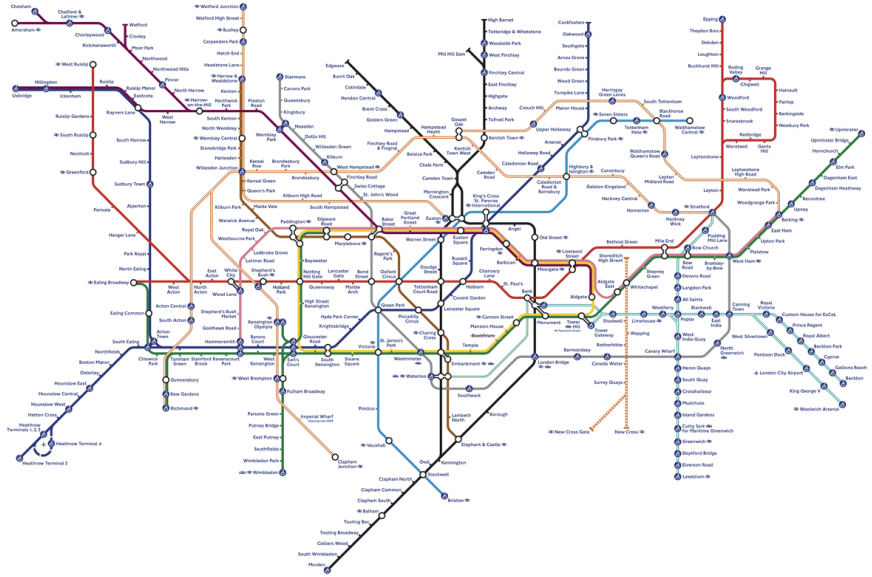
Pick up the whiteboard markers



Find the right level of abstraction…

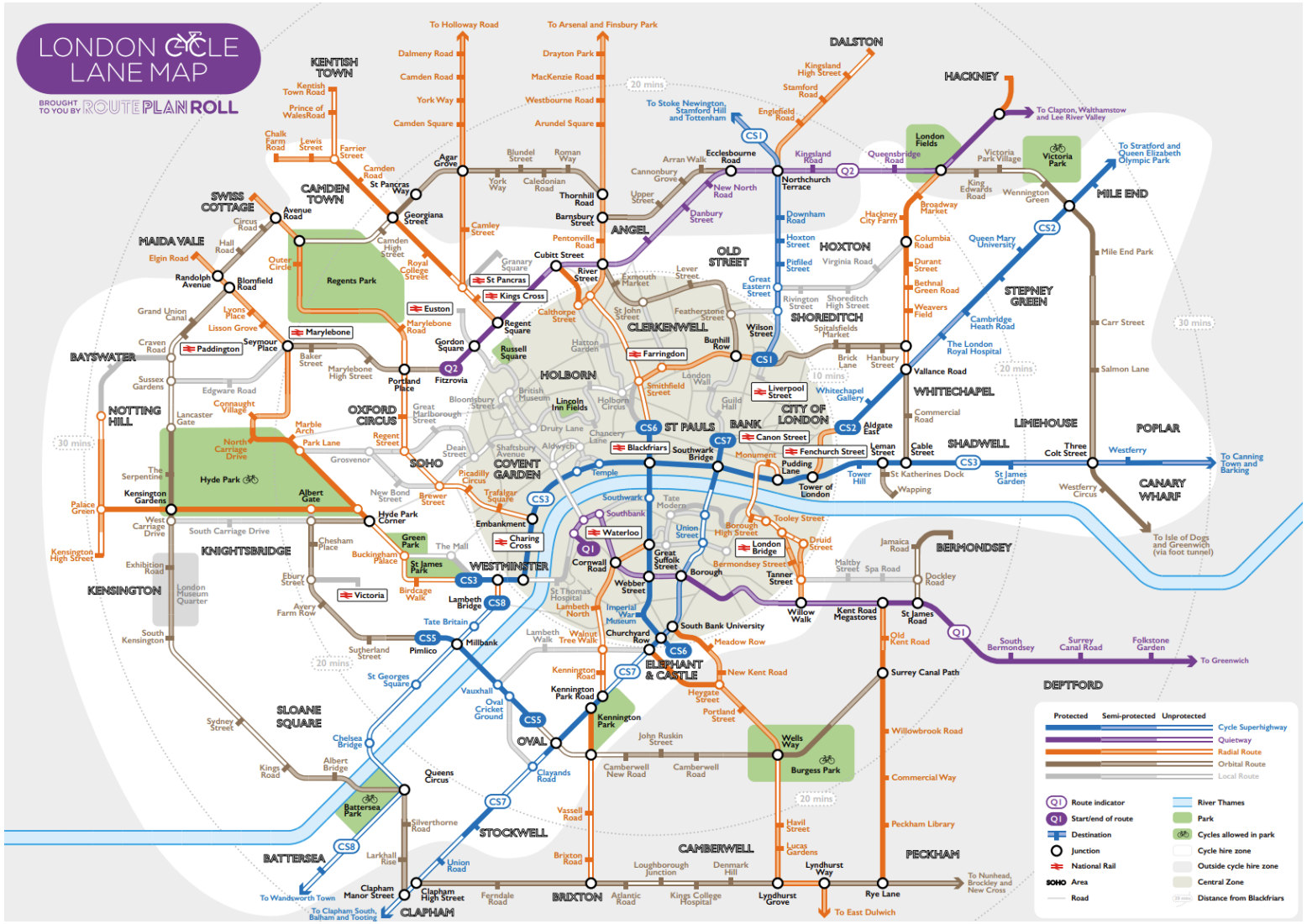






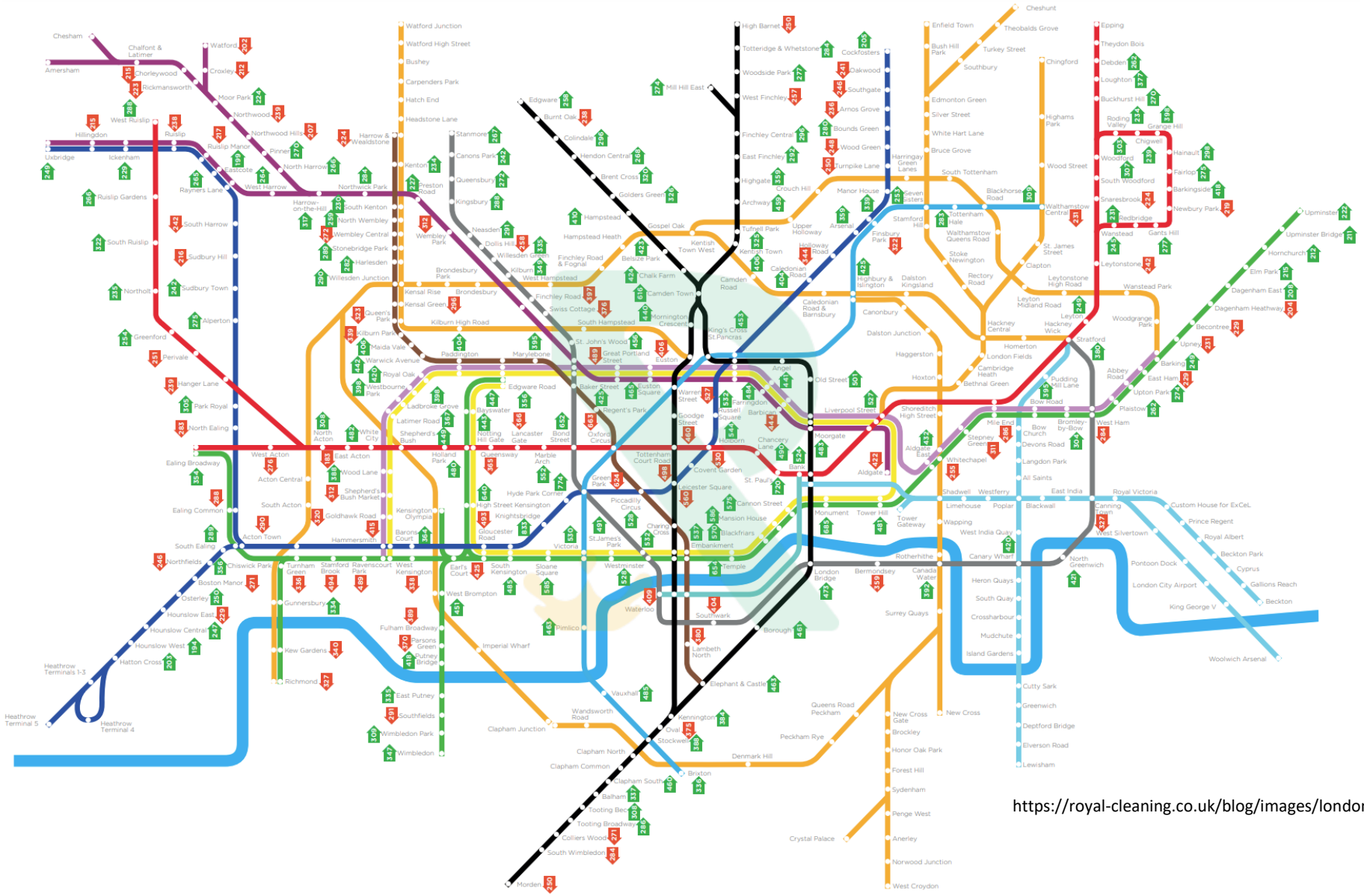
Source: google maps

… and identify the relevant viewpoints



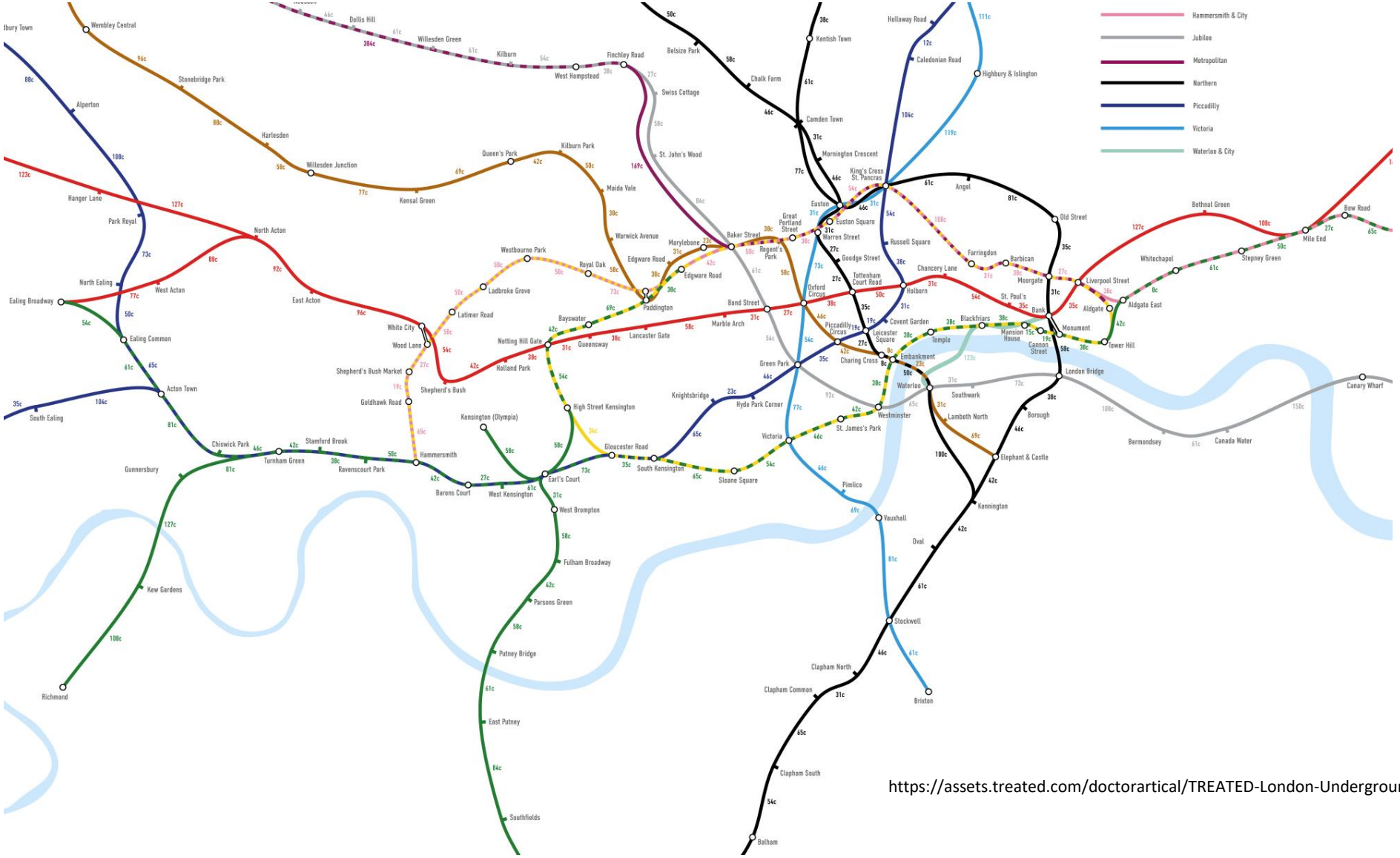
https://londoncyclenetwork.files.wordpress.com/2016/10/london-cycle-map-e28093-printed-map-design-v1-2.pdf

… and identify the relevant viewpoints



https://royal-cleaning.co.uk/blog/images/london-tube-map-rent-prices-2019.pdf

… and identify the relevant viewpoints



https://assets.treated.com/doctorartical/TREATED-London-Underground-calorie-map-final.jpg

Software architecture documentation is essentially about communication *.

To communicate, we need a shared language:

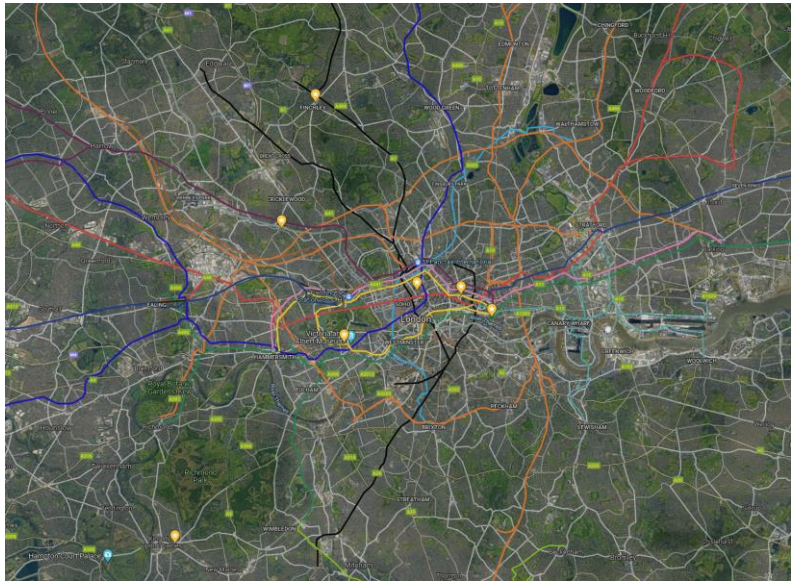
shared vocabulary, shared notation

shared understanding of abstraction level

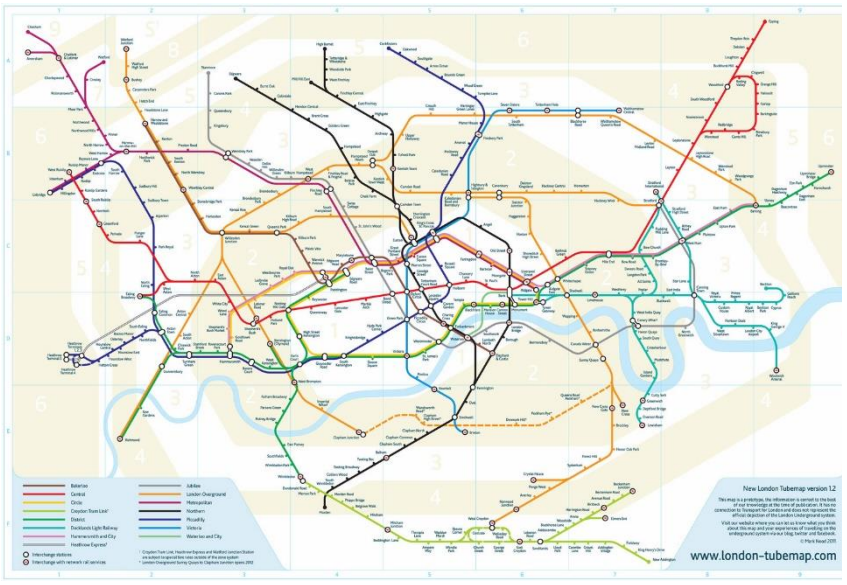
Simon Brown:

Abstractions 1st, notation 2nd.

* Except on safety, traceability, and external assessment scenarios.



Source: google maps



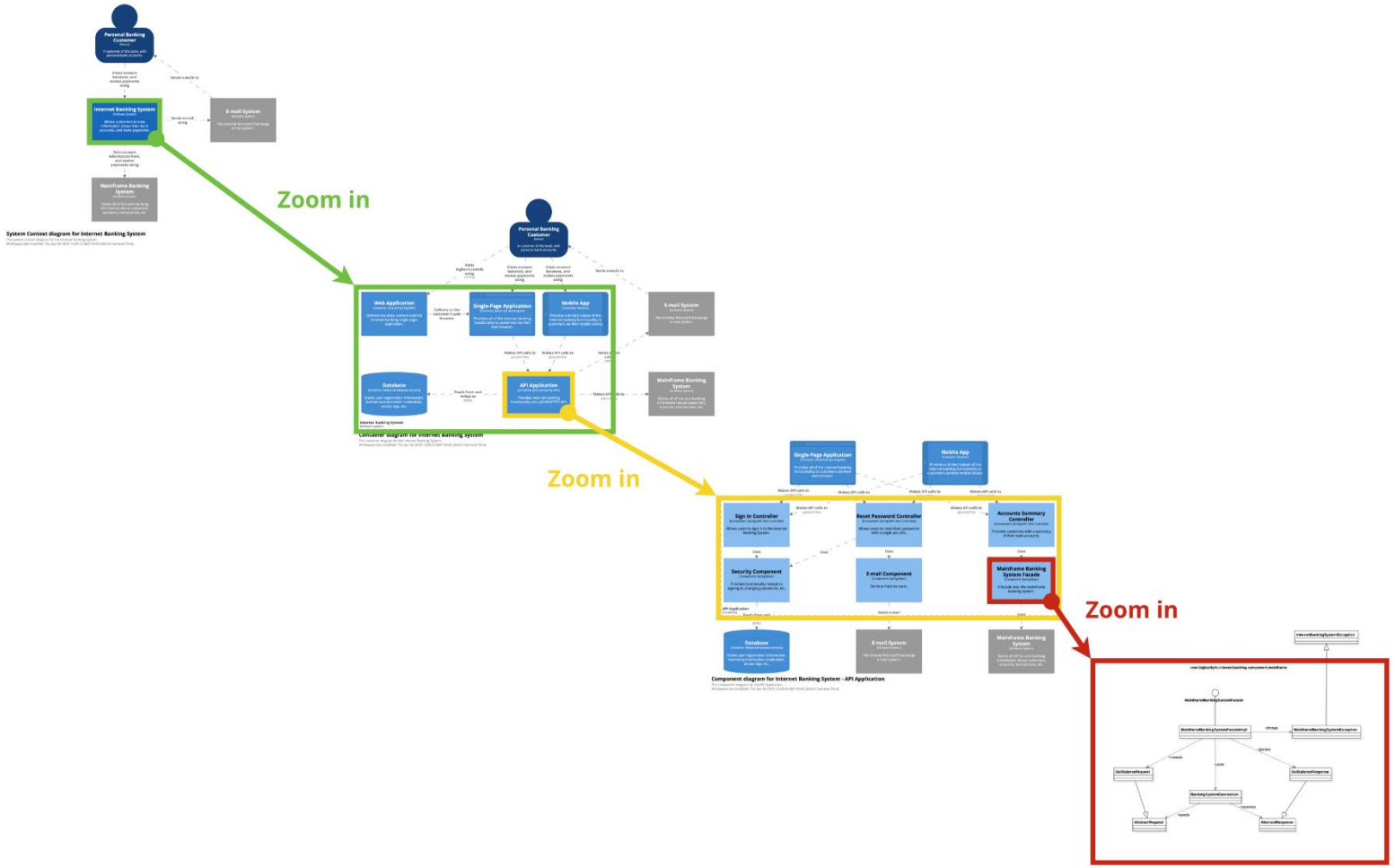
Documenting Abstraction Levels

Using the C4 model

Next lecture

And then: documenting different viewpoints using 4+1 View Model

C4 model by Simon Brown



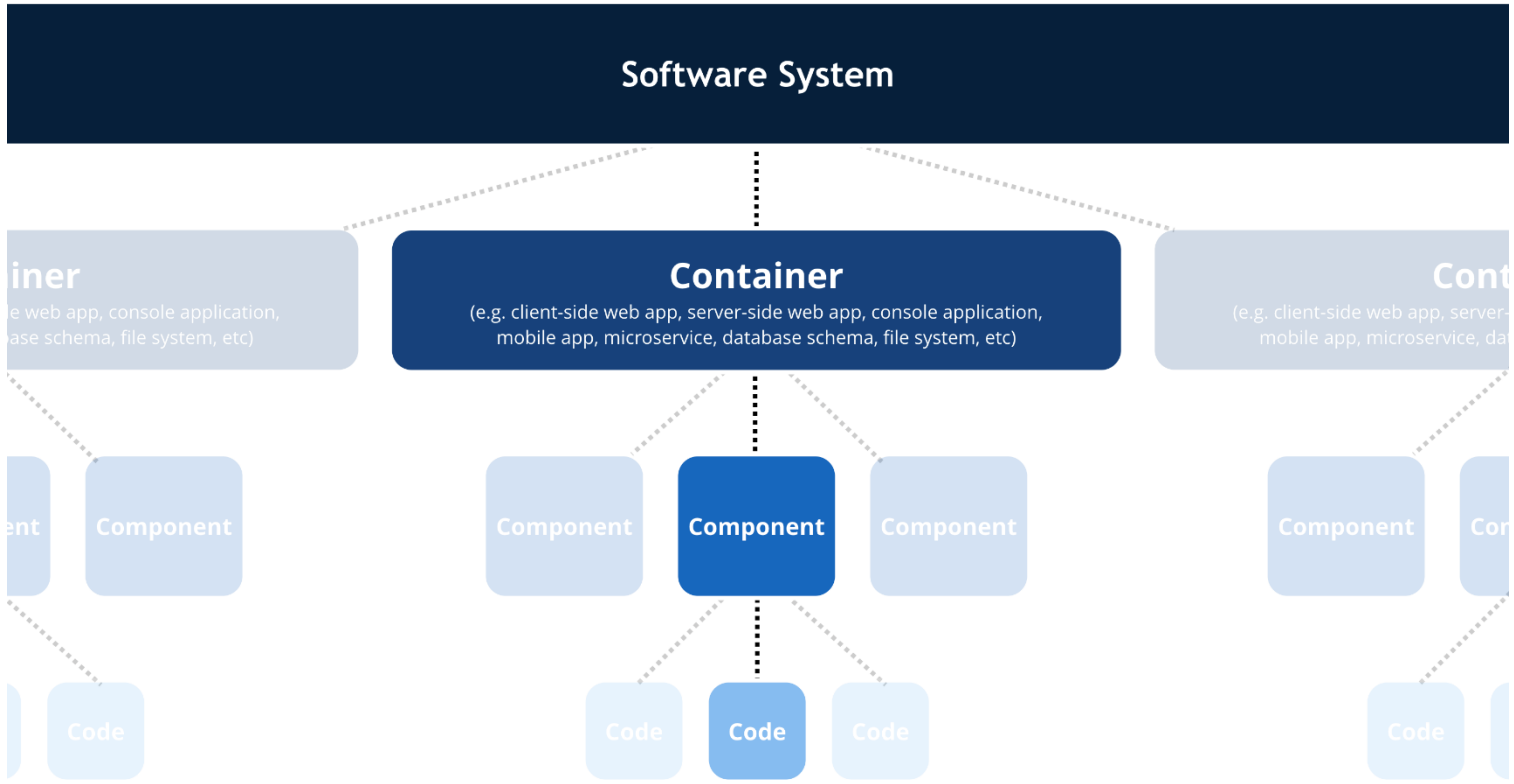
Level 1
Context

Level 2
Containers

Level 3
Components

Level 4
Code

Abstractions in C4



*A **software system** is made up of one or more*

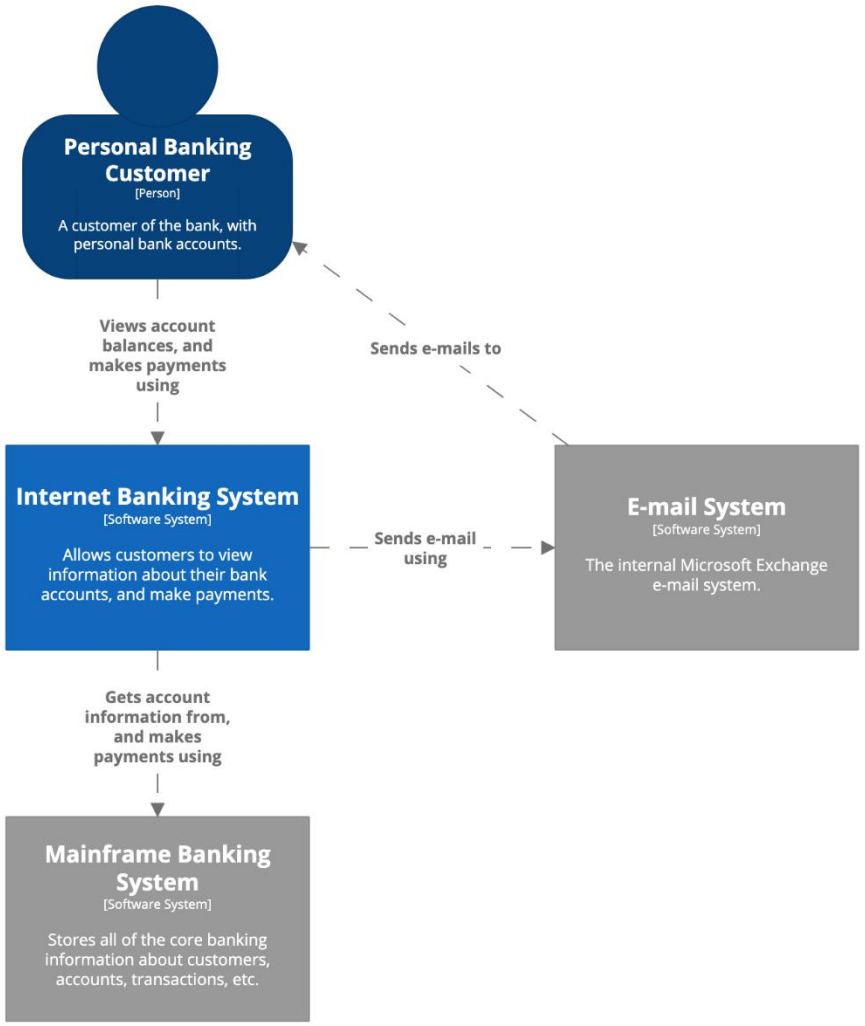
***containers** (web applications, mobile apps, desktop applications, databases, file systems, etc), each of which contains one or more*

***components**, which in turn are implemented by one or more*

***code elements** (e.g. classes, interfaces, objects, functions, etc).*

Source: https://c4model.com/

Level 1: System Context diagram

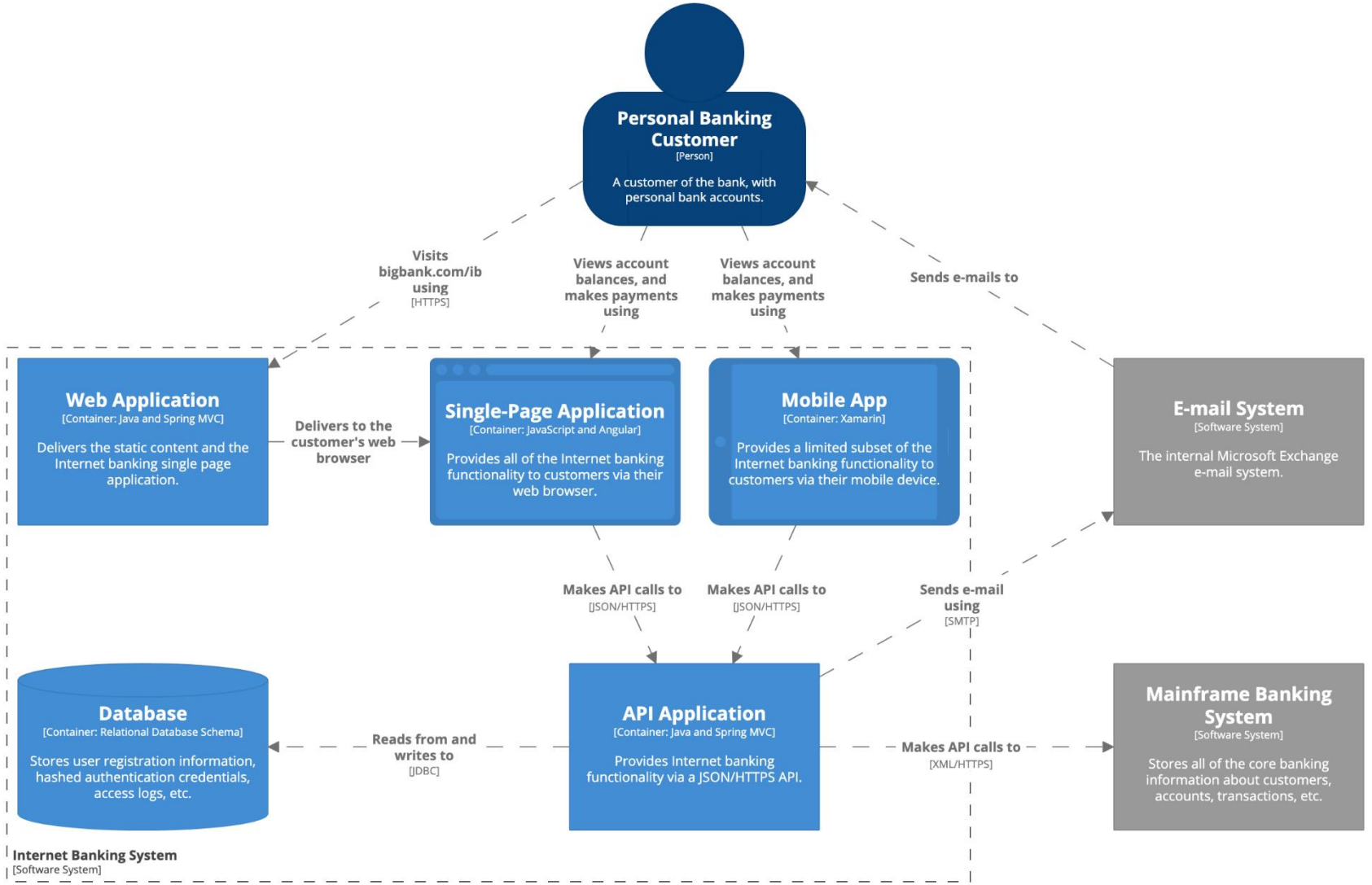


System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.
Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

- A zoomed out view showing a big picture of the system landscape.
- The focus should be on people (actors, roles, personas, etc) and software systems rather than technologies, protocols and other low-level details.
- It's the sort of diagram that you could show to non-technical people.

Level 2: Container diagram



Container diagram for Internet Banking System

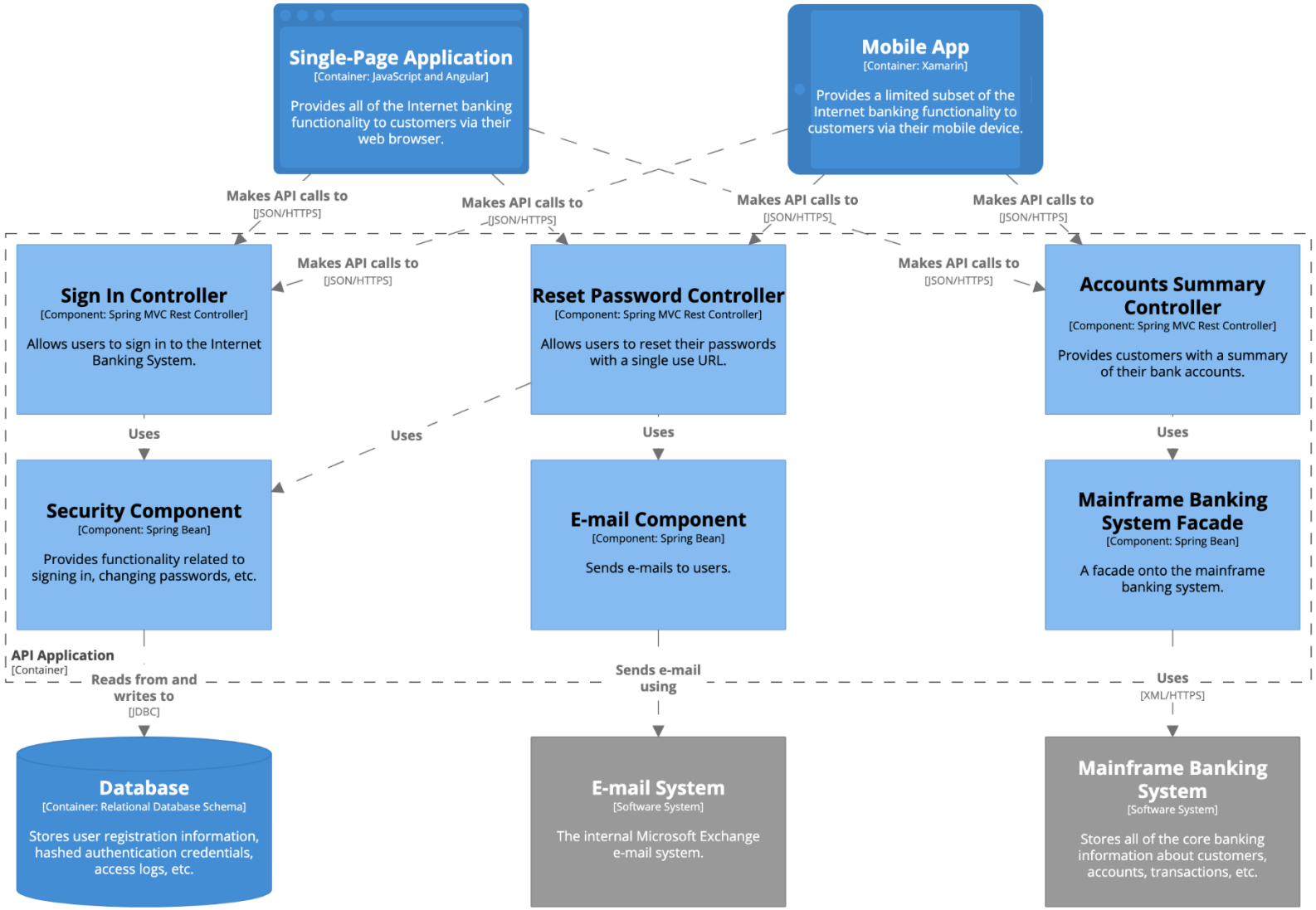
The container diagram for the Internet Banking System.
Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

A container is a separately runnable/deployable unit (e.g. a separate process space) that executes code or stores data.

The Container diagram shows the high-level shape of the software architecture and how responsibilities are distributed across it.

It also shows the major technology choices and how the containers communicate with one another.

Level 3: Component diagram



Component diagram for Internet Banking System - API Application

The component diagram for the API Application.
Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

Zoom in and decompose each container further to identify the major structural building blocks and their interactions.

The Component diagram shows how a container is made up of a number of "components", what each of those components are, their responsibilities and the technology/implementation details.

Component?

The word "component" is a hugely overloaded term in the software development industry.

In this context a component is simply **a grouping of related functionality encapsulated behind a well-defined interface**.

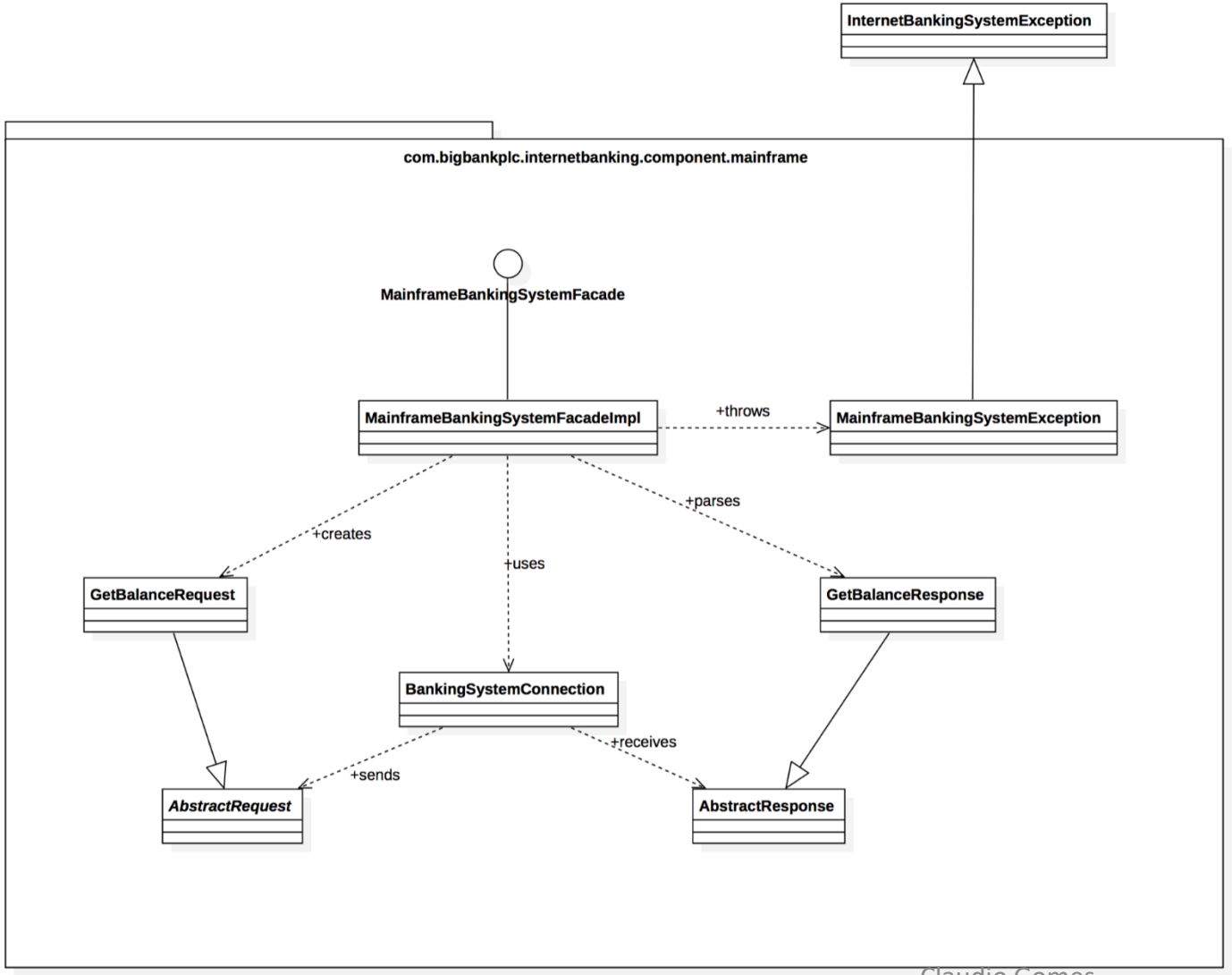
If you're using a language like Java or C#, the simplest way to think of a component is that it's a collection of implementation classes behind an interface.

Aspects such as how those components are packaged (e.g. one component vs many components per JAR file, DLL, shared library, etc) is a separate and orthogonal concern.

An important point to note here is that **all components inside a container typically execute in the same process space**.

Source: https://c4model.com/

Level 4: Code (Optional)



You can zoom in to each component to show how it is implemented as code; using UML class diagrams, entity relationship diagrams or similar.

You should consider showing only those attributes and methods that allow you to tell the story that you want to tell.

Diagram guidelines

Diagrams should be self explaning.

Specify the legend (explain the notation used).

Add descriptive text.

Use color and shapes to optimize readability (but make sure these are just optimizations).

A narrative should complement a diagram, not explain it [Simon Brown].

You should be able to read the diagram aloud and understand it.

The C4 model is a way to document structure.

The C4 model does not mandate any specific notation.

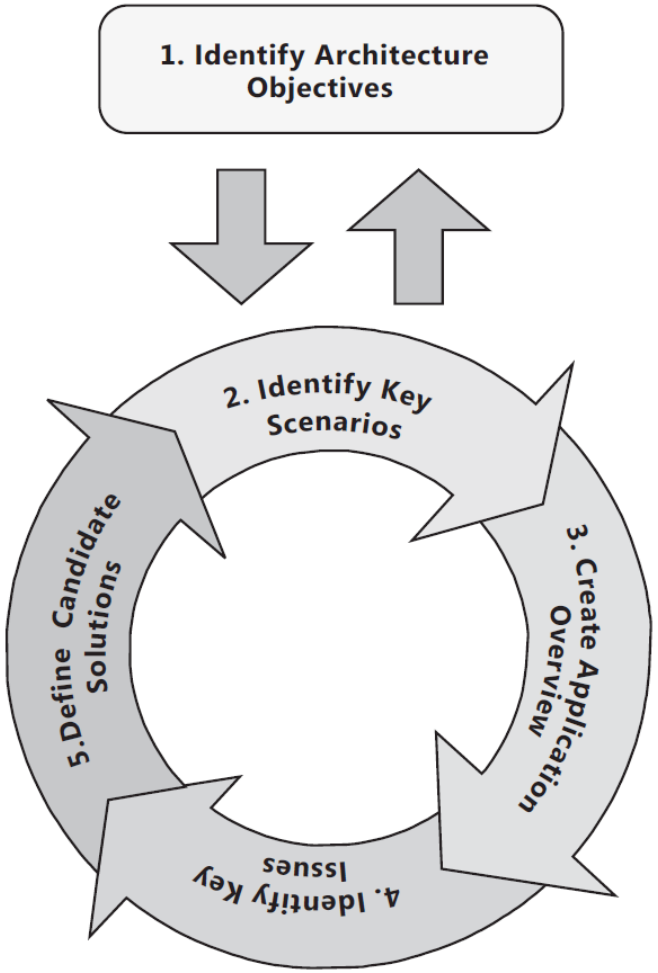
- Simon Brown has a suggestion.
- UML can also be used (se the c4model website)

And remember that behavior is equally as important as structure!

Next lecture

So maybe you need some other viewpoints (supported by other diagrams)!

Recap: An architecture process (from Microsoft)



Iterative and incremental.

Test against:

- requirements
- known constraints
- quality attributes

Source: Microsoft Application Architecture Guide, 2nd Edition

Continue with the process

Your turn again!

Draw the architecture

Pair with someone, who is not working
on the same project as you.

You have 10 minutes to
explain the software architecture
of your project to the other person.
Then switch roles.

Use paper and pencil (or a whiteboard).
PC's and pre-made diagrams are forbidden!

AARHUS UNIVERSITY

References and image sources

Images (s):

C4 model: c4model.com