DWARVES FOUNDATION

C4 model

Visualising software architecture

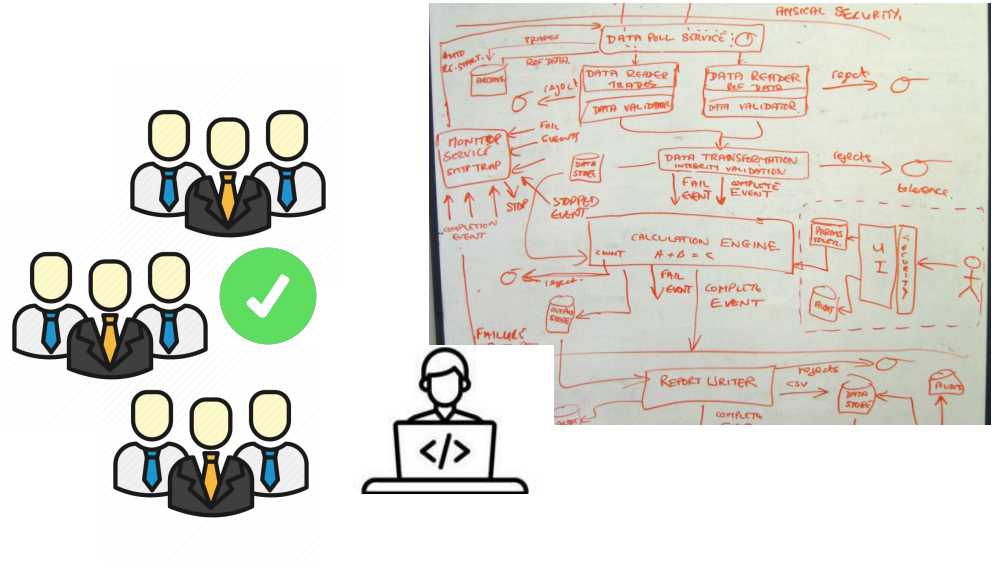
Agenda

Topic summary

1. C4 Model
2. Abstractions
3. Visualization
4. Case by case

Problem statement

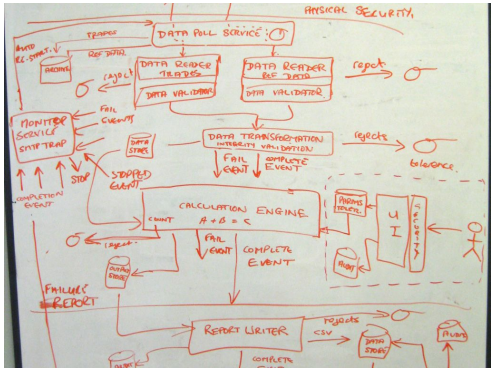
- Developers need to convey the software architecture information to the business people/other teams.

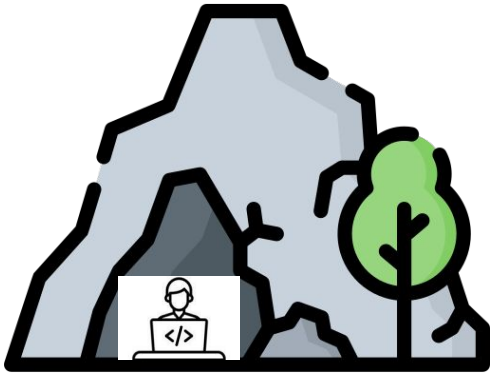


DWARVES FOUNDATION

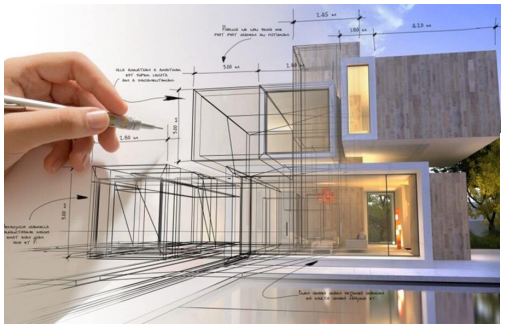
Problem statement

- But Developers are handicapped by:
  - UML complexity or non descriptive box and line drawing.
  - Confusing Technical terms and vocabulary
  - Being introvert





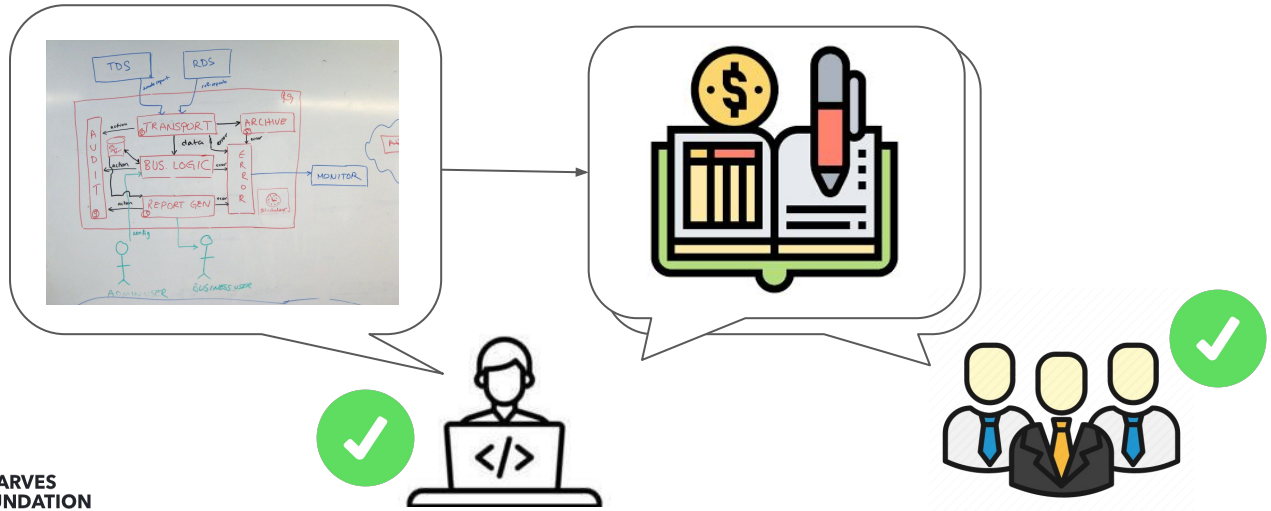
C4 Model



DWARVES FOUNDATION

C4 model

- The C4 model leveraging abstraction as a common language and visualization to describe the structure of the software system



Why using abstractions?

Abstrations

- Business people, even developers themself only know the context of the vocabulary specific to their domain and not the terms in system architecture.

Abstrations

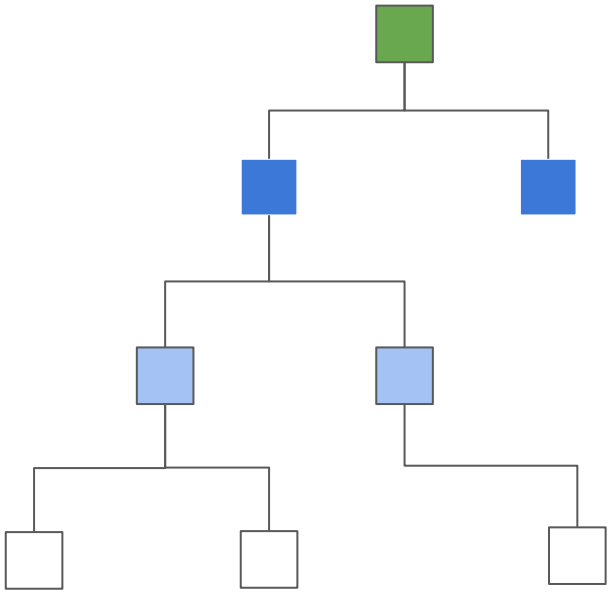
- Involving parties must devise and agree upon abstractions which help define the domain that the software system will take place in.

Abstrations

- Abstractions are set of domain specific. It meaning is defined with the effort, and approval of all involving parties(developer, stakeholder, business analyst).

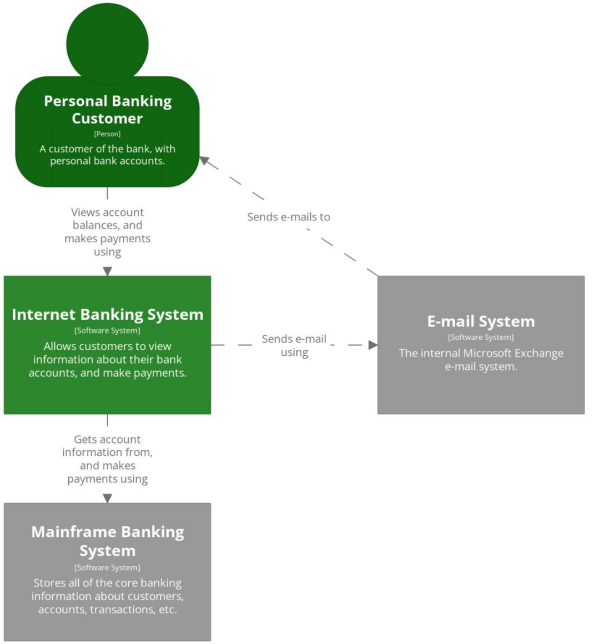
Level of abstractions

- C4 model divides the level of abstractions into:
  - Context of the Software system
    - Container
      - Component
        - Code



Software system context

- The highest level of abstraction and describes the software system and other dependent systems created a whole package that delivers value to its users.



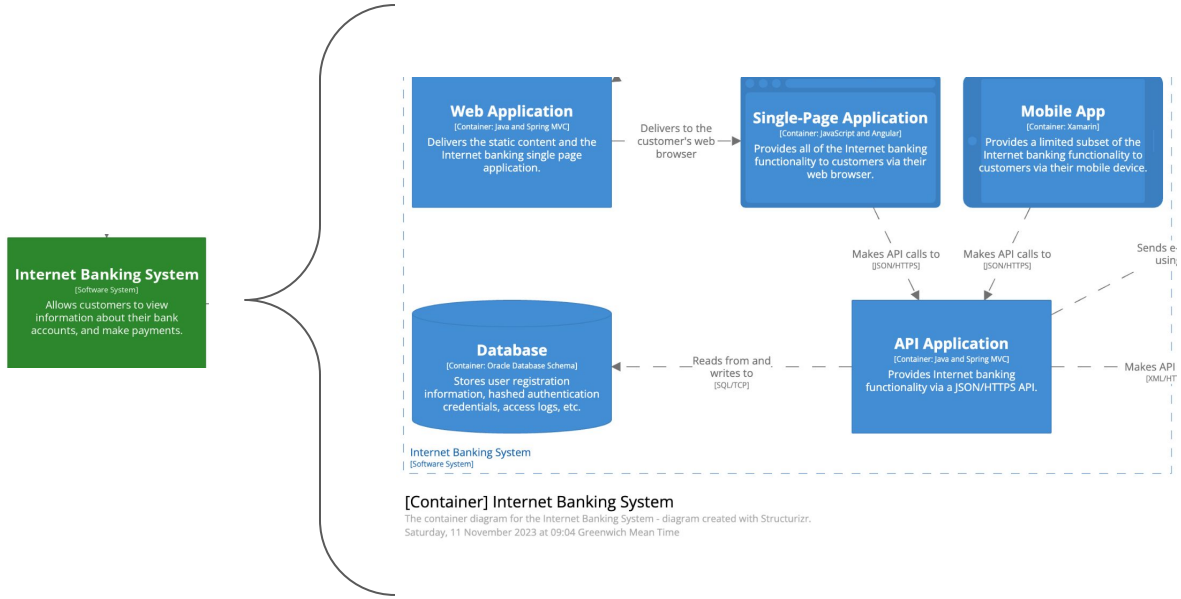
[System Context] Internet Banking System

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

DWARVES FOUNDATION

Container

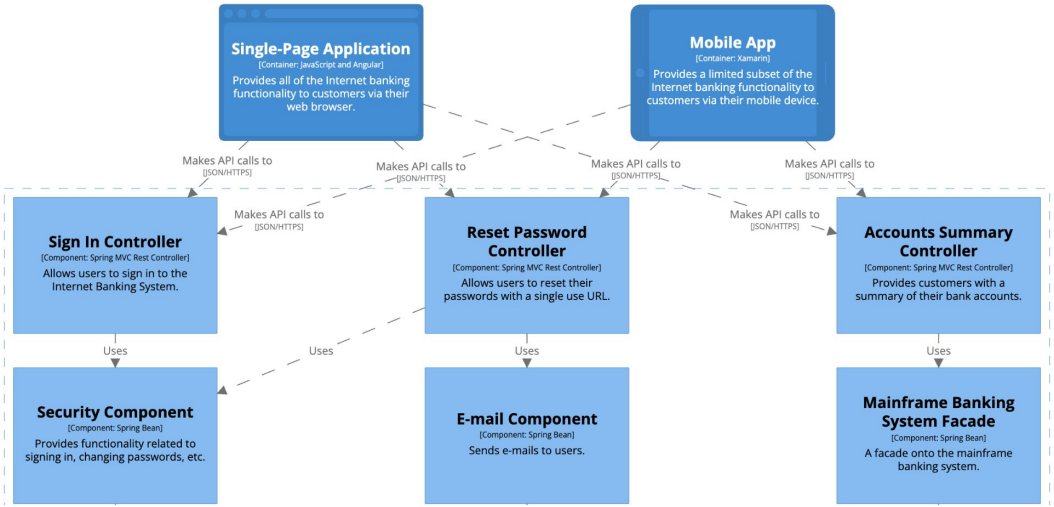
- Container is the abstraction of an application or a datastore. A container is something that needs to be running in order for the overall software system to work



DWARVES FOUNDATION

Component

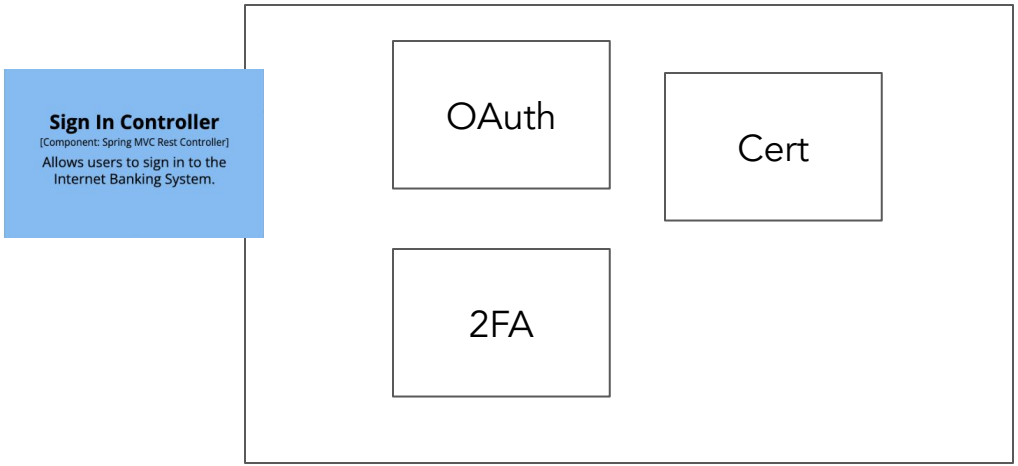
- Component is a set of functions and classes bounded behind an interface.

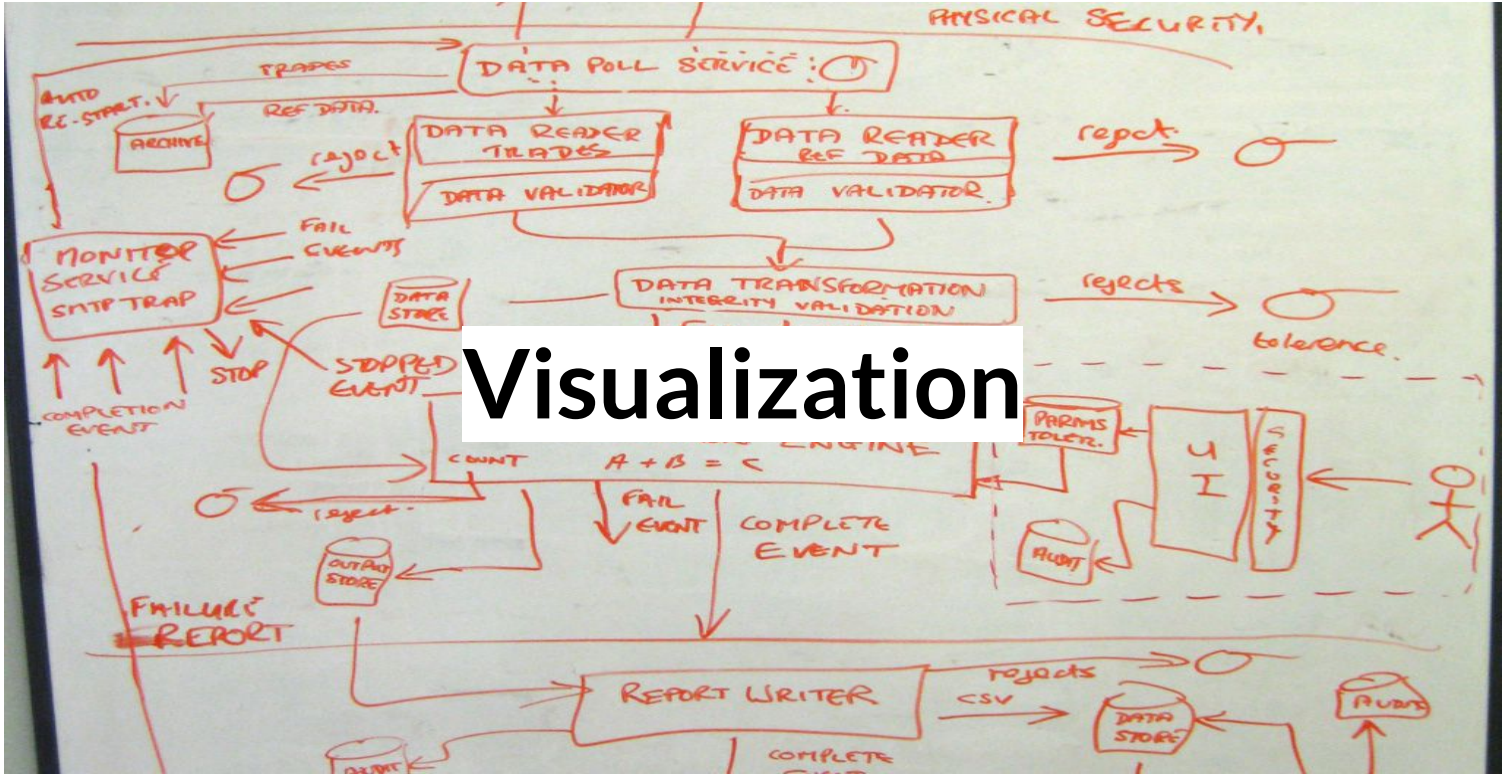


DWARVES FOUNDATION

Code

- Code is the specific implementation that built the component.

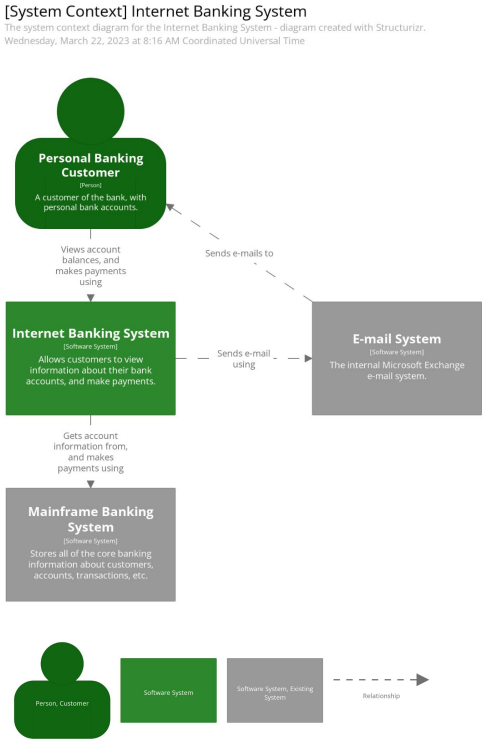




Visualization

Visualization

- We can use UML components or box and line diagram to visualize the software architecture abstractions.
- The diagram must convey the solution meant for that specific domain.
- The diagram elements must be clear, uniform, meaningful and have a legend to explain the usage of each element.



Visualization

Two-way arrows	No diagram title	Writing upside-down	Unexplained box colours	Incomplete key/legend
A box labelled "Security"	Mixed abstractions	A question mark	Unlabelled arrows	A clock symbol
A box labelled "Business Logic"	Unexplained numbers	C4 model for visualising software architecture	Unexplained line styles	Unexplained symbols/icons
Unexplained text colours	Horizontal and vertical "layers"	A cloud shape	Unreadable handwriting	Unexplained arrowheads
Unexplained line colours	Missing technology choices	Unexplained shapes	Ambiguous abstractions	Crossed out boxes/lines

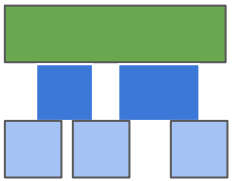
C4 model diagram bingo

Case by case

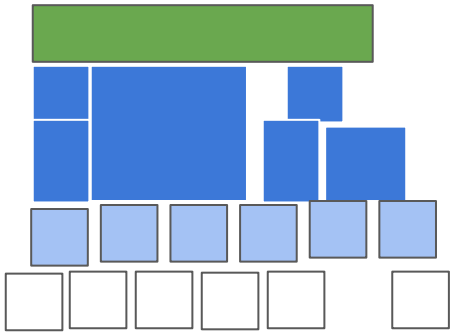
Case by case

- The C4 model's diagram most likely will be the source of truth for other design document, diagram, feature discussions and as result, level of abstractions and how we visualize it should focus on the value we want to deliver

Ticket box system



Banking system



Reference

Resources & Reference links

- https://c4model.com/

DWARVES FOUNDATION

Thank You

DWARVES FOUNDATION

Q&A