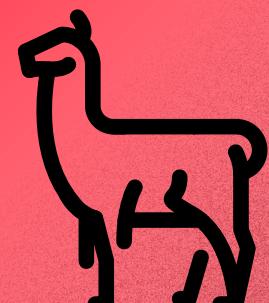




The best life hack,
but
for your software

Andrea Magnorsky @silverspoon



Hose

Tree
trunk

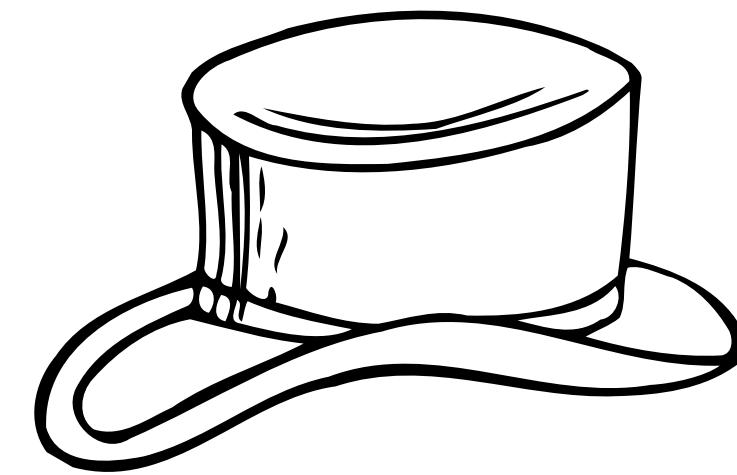
Wall



Blind Men Appraising an
Elephant by Ohara Donshu,
Edo Period (early 19th
century)

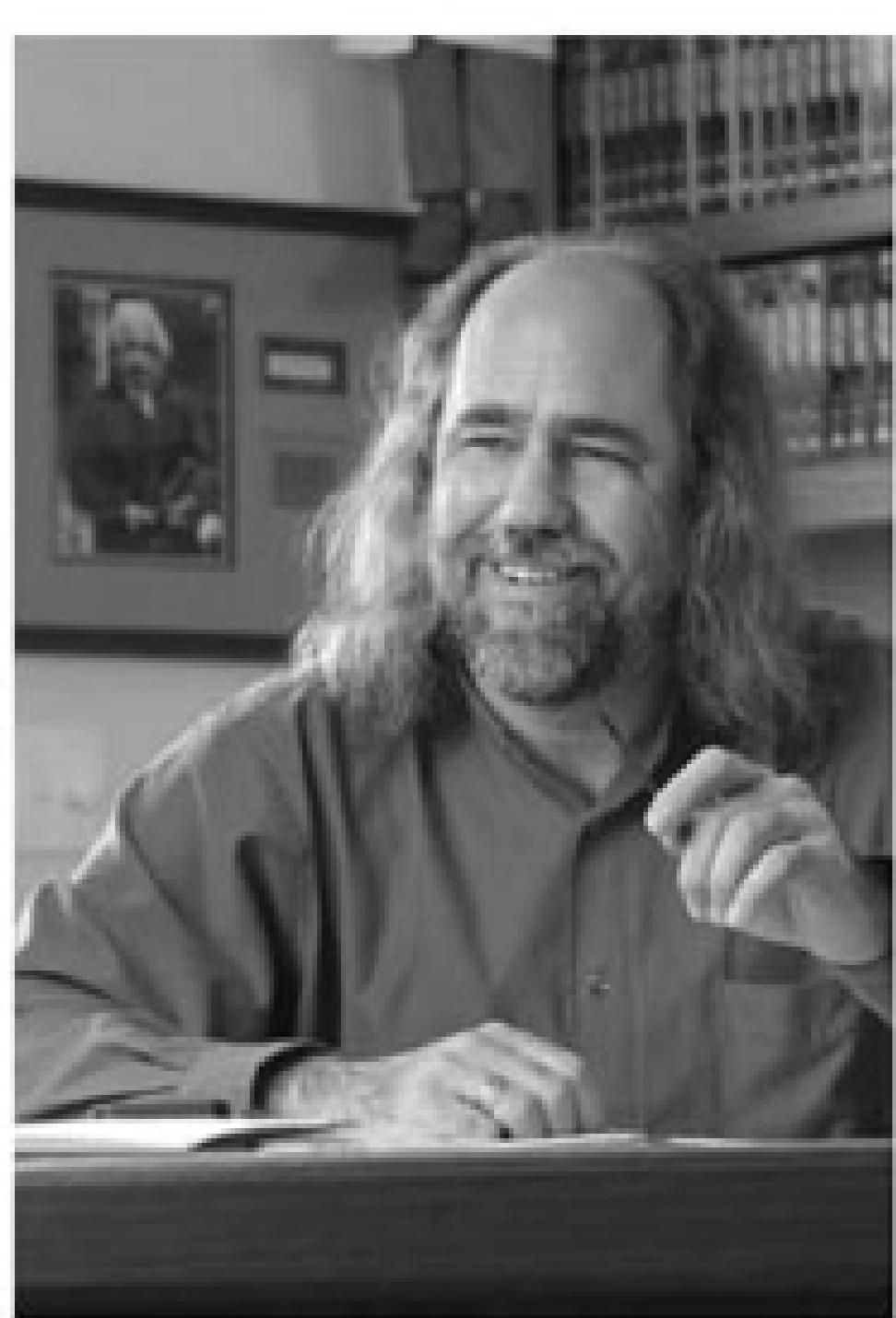
A photograph of a savanna landscape. In the foreground, a herd of elephants is grazing in a grassy field. Some elephants are facing left, while others are facing right. The background features several acacia trees with distinctively shaped, spreading canopies. The sky is clear and blue.

Tools to see the
elephant and then some



"All architecture is **design** but not all design is architecture. Architecture represents the significant **design** decisions that shape a system, where significant is measured by cost of change."

-- Grady Booch, blog post,
March 2, 2006



“A system is an interconnected set of elements that is **coherently** organized in a way that achieves something”

-- Donella Meadows

Thinking in Systems

A Primer

Donella H. Meadows

*Edited by Diana Wright,
Sustainability Institute*

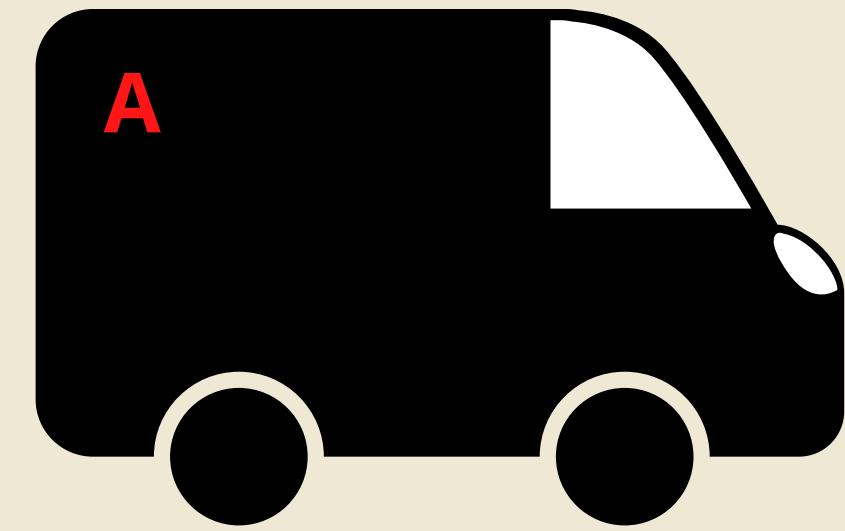




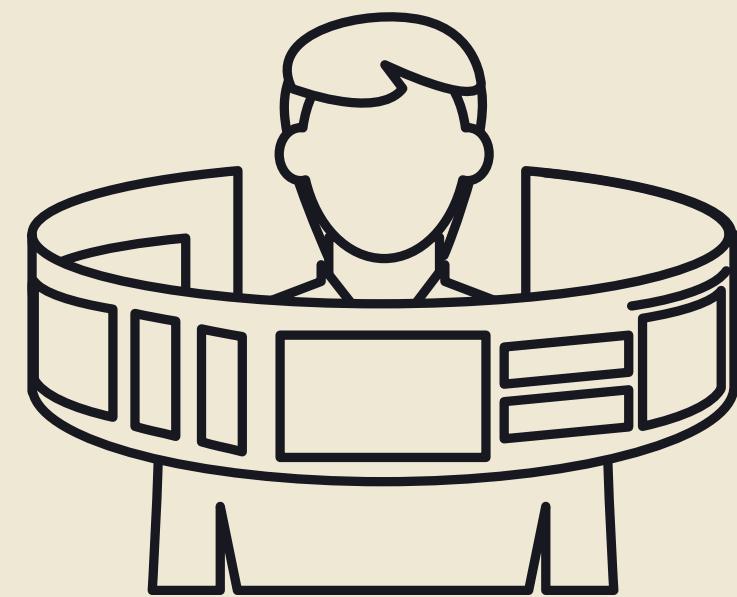
Architecture



Design



Team



Systems



Philosophy

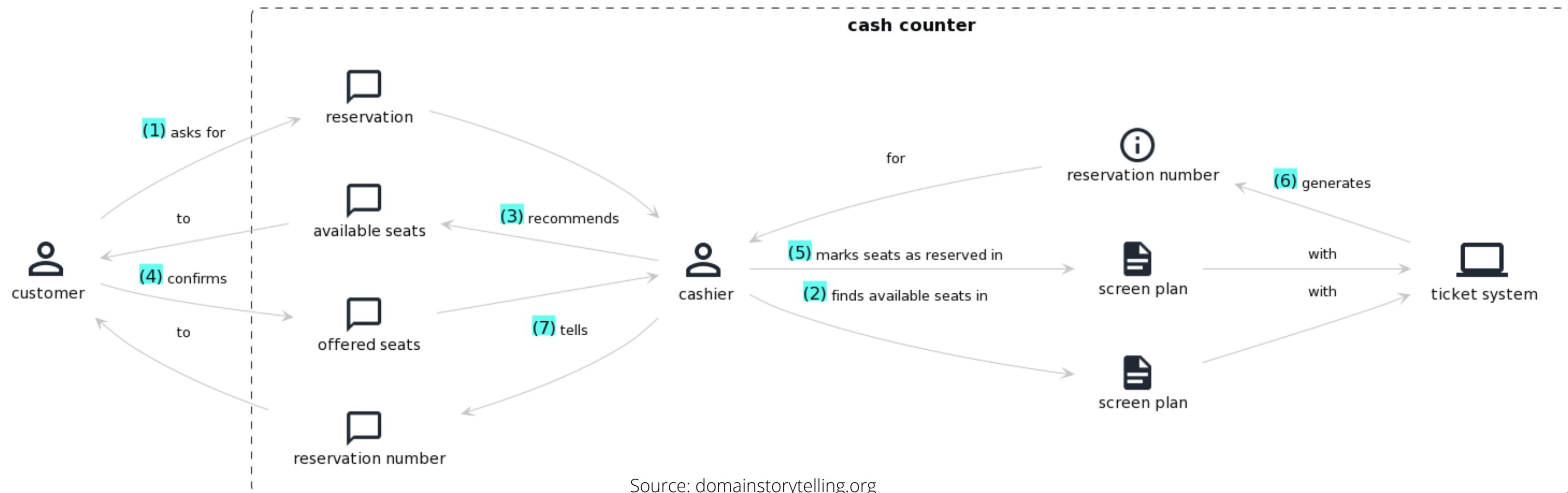
- Understand the problem:
Domain storytelling, event storming, impact mapping, rich picture
- Understand the systems you have
- Record expensive decisions
- Analyse your current context and strategy
- Keep doing all the above



DOMAIN STORYTELLING

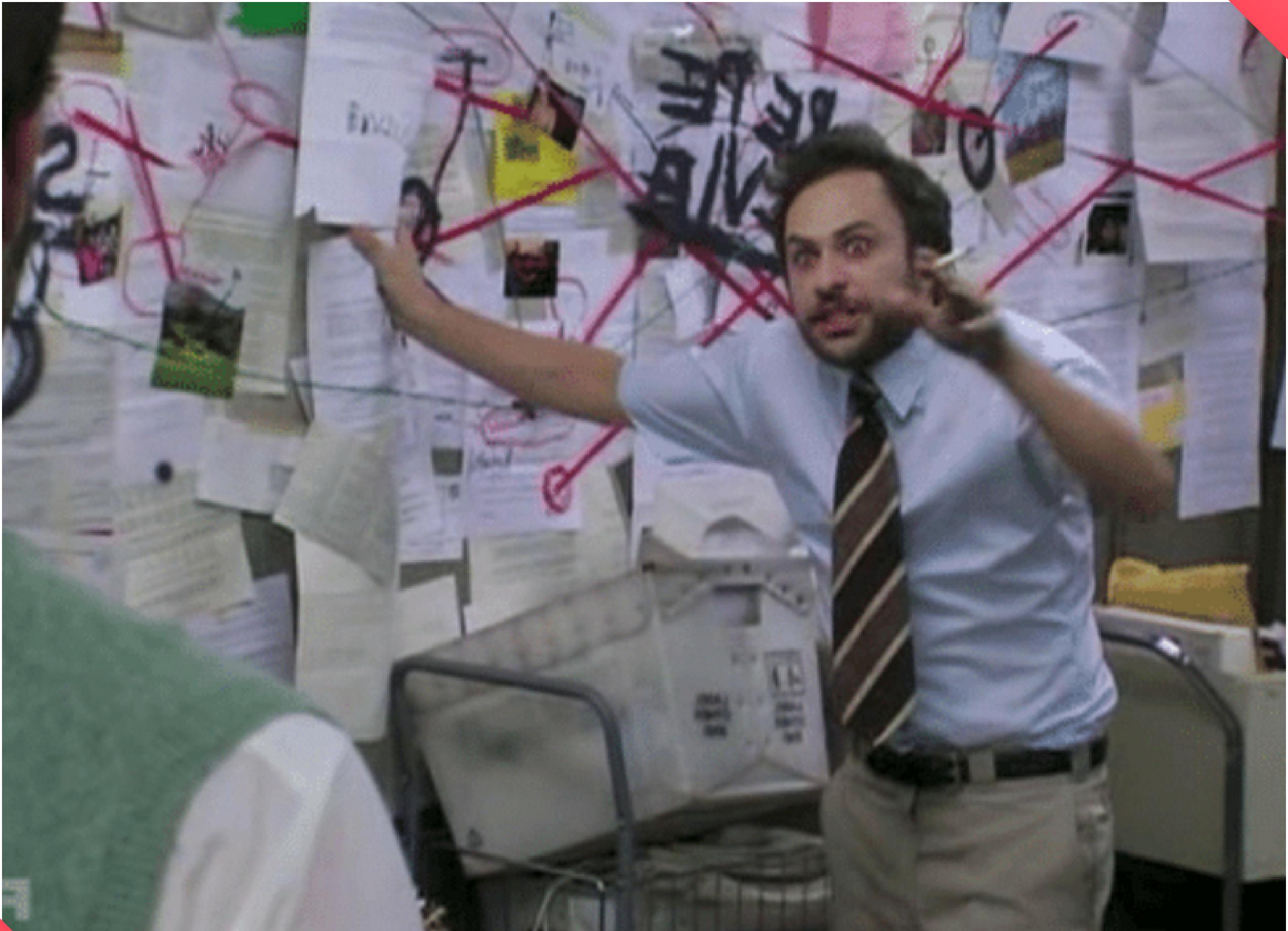
- Fully align all project participants and stakeholders, both technical and business-focused
- Draw clear boundaries to organise your domain, software, and teams
- Transform domain knowledge into requirements
- Gain better visibility of your landscape so you can consolidate it or optimise it

DOMAIN STORYTELLING



- Understand the problem:
Domain storytelling, event storming, impact mapping, etc
- **Understand the systems and their context**
C4, UML, sequence diagrams, CRC, etc
- Record expensive decisions
- Analyse your current context and strategy
- Keep doing all the above

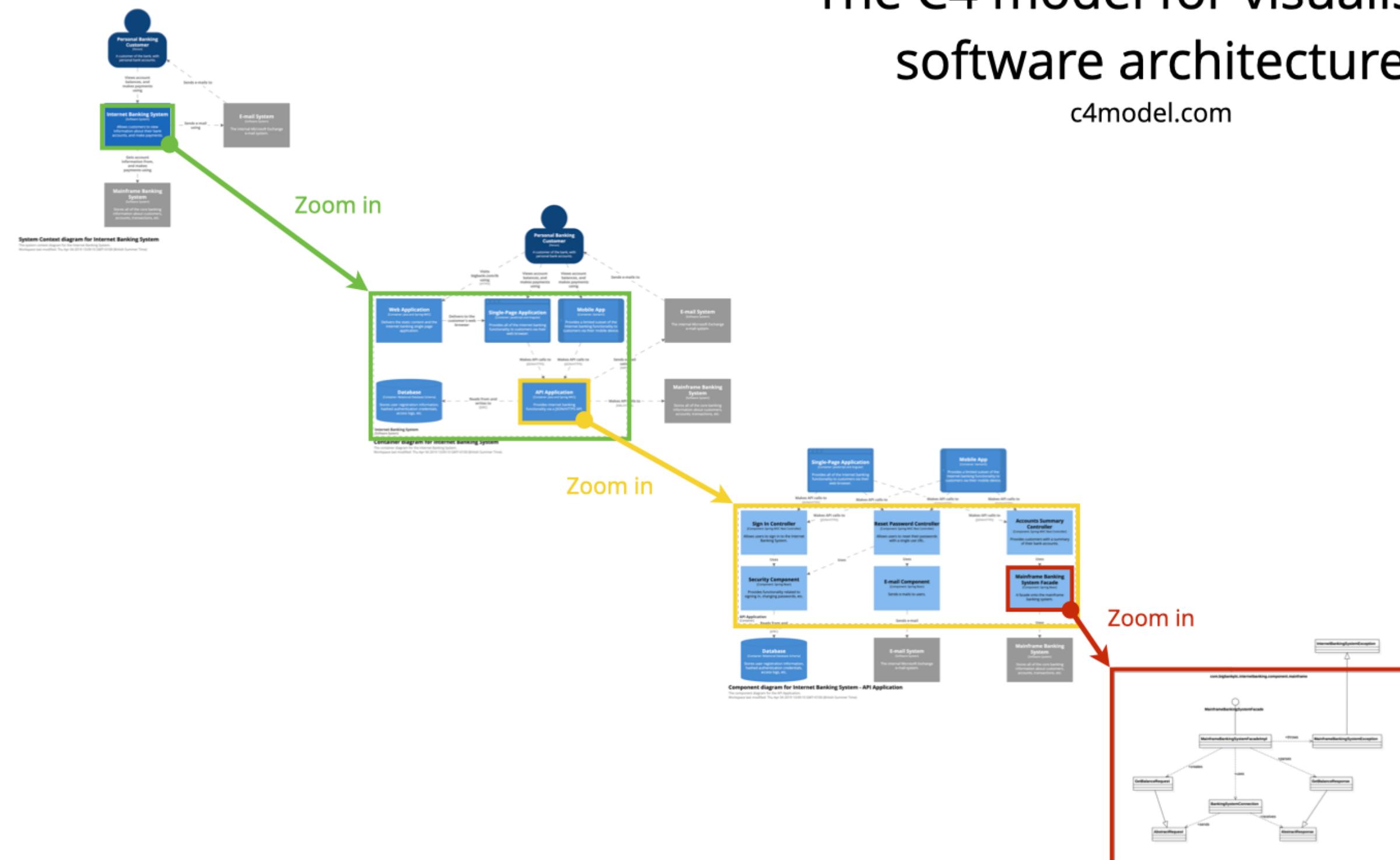




More about languages: <http://m.nautil.us/issue/54/the-unspoken/the-rise-and-fall-of-the-english-sentence>

The C4 model for visualising software architecture

c4model.com



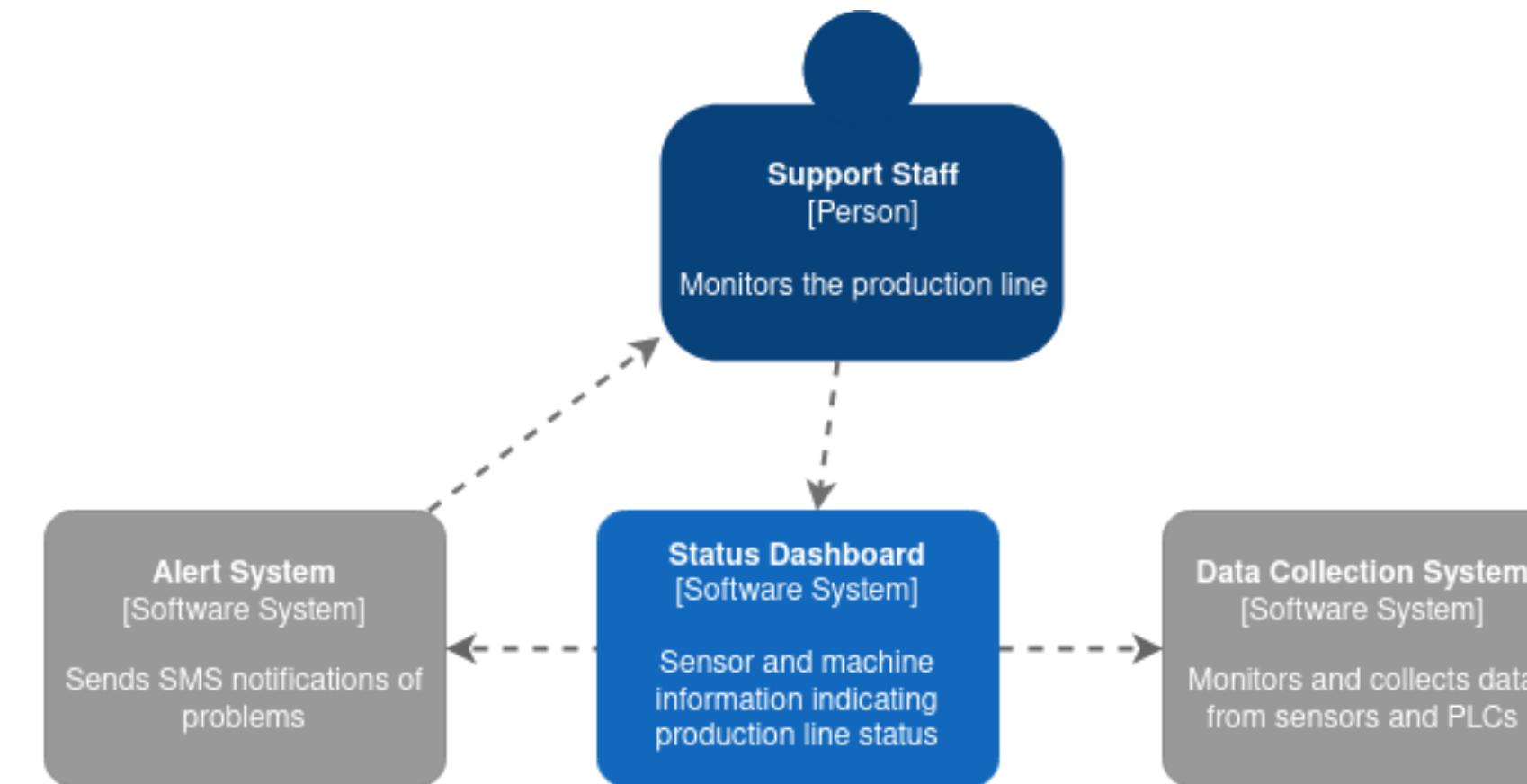
Level 1
Context

Level 2
Containers

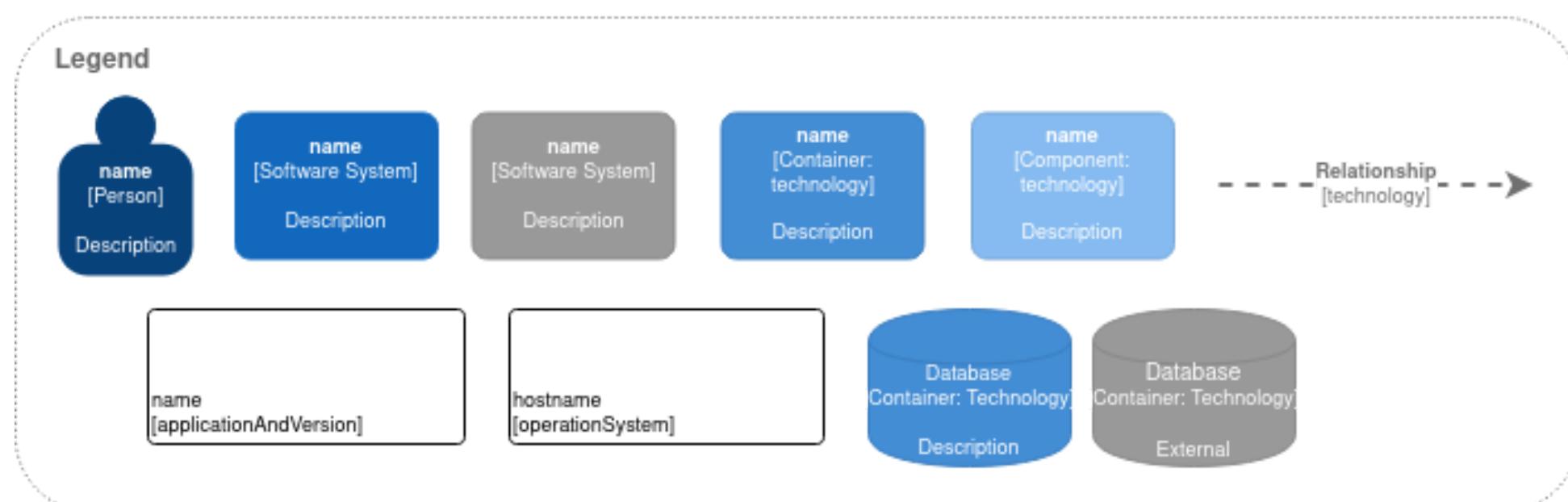
Level 3
Components

Level 4
Code

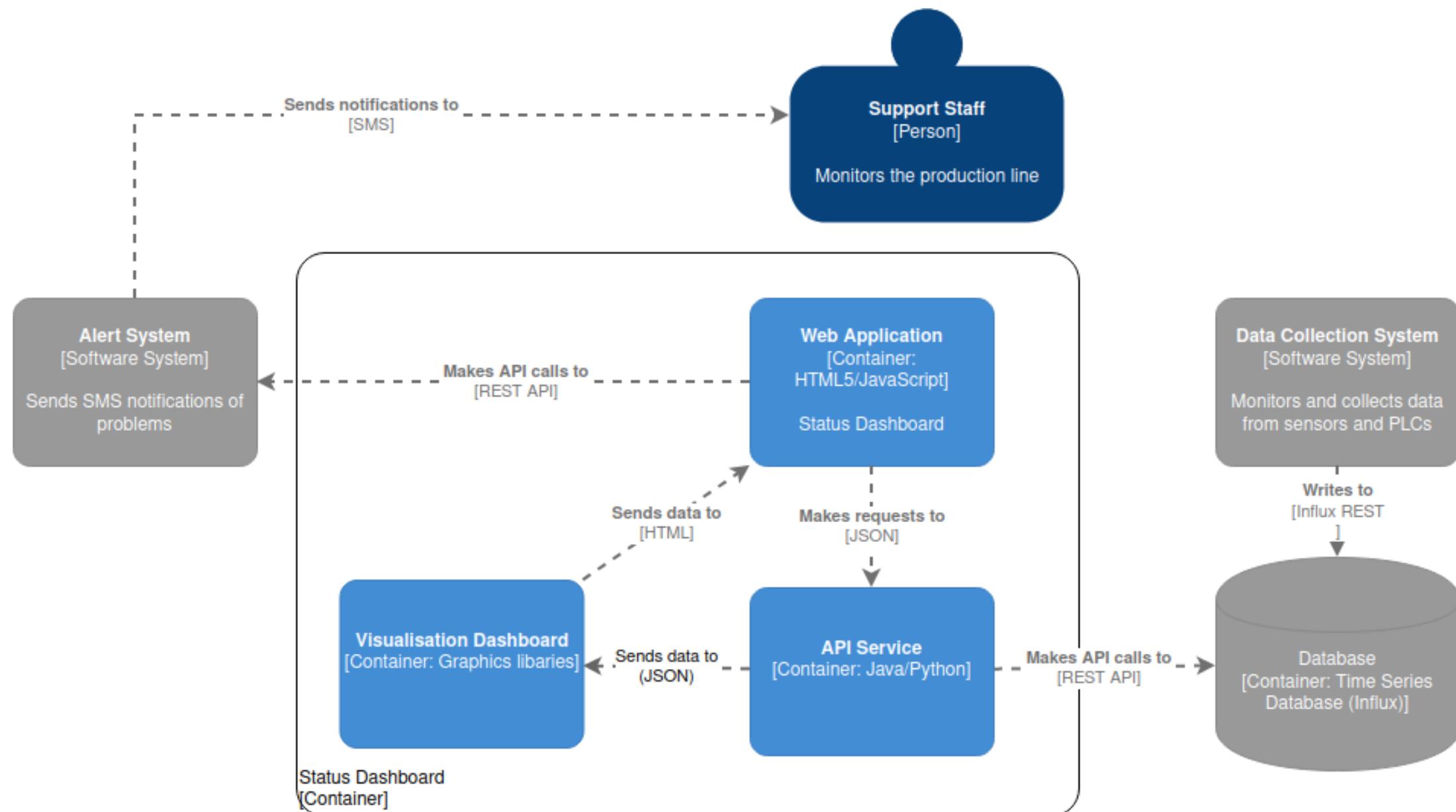
Context



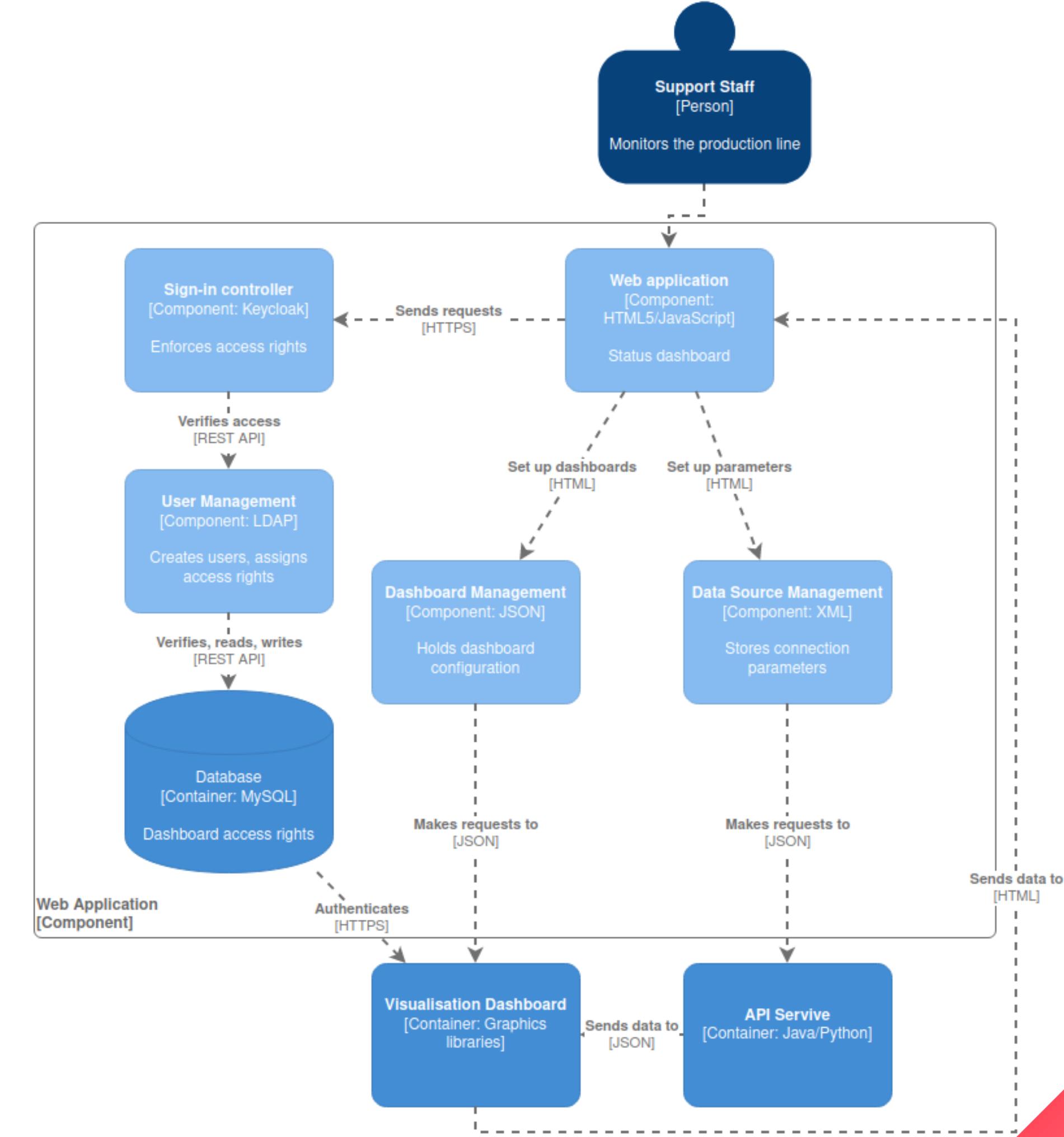
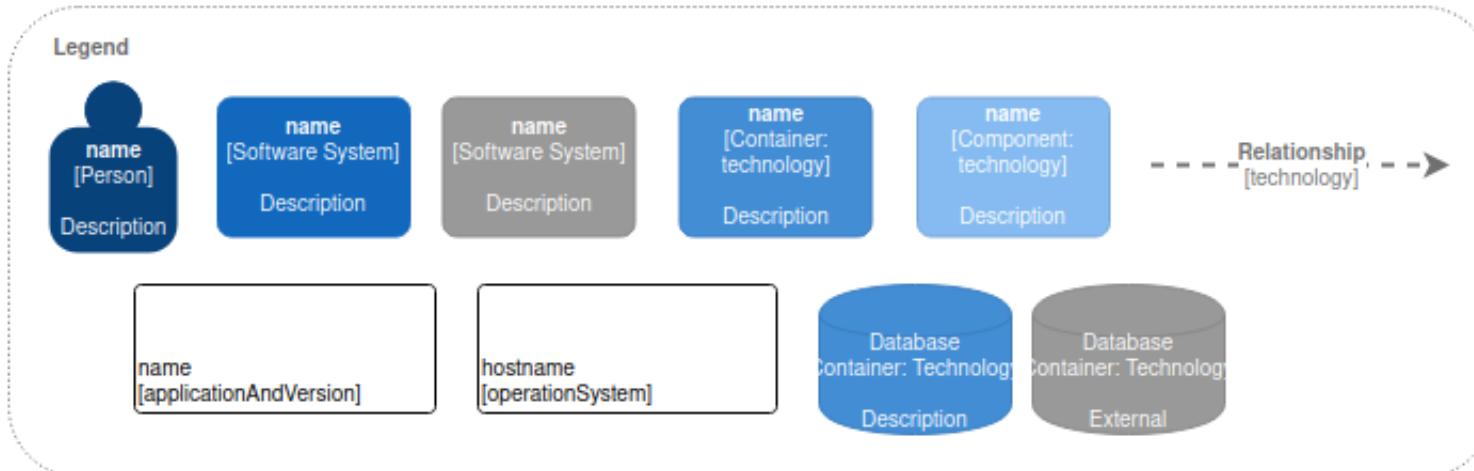
Status Dashboard
(Context)



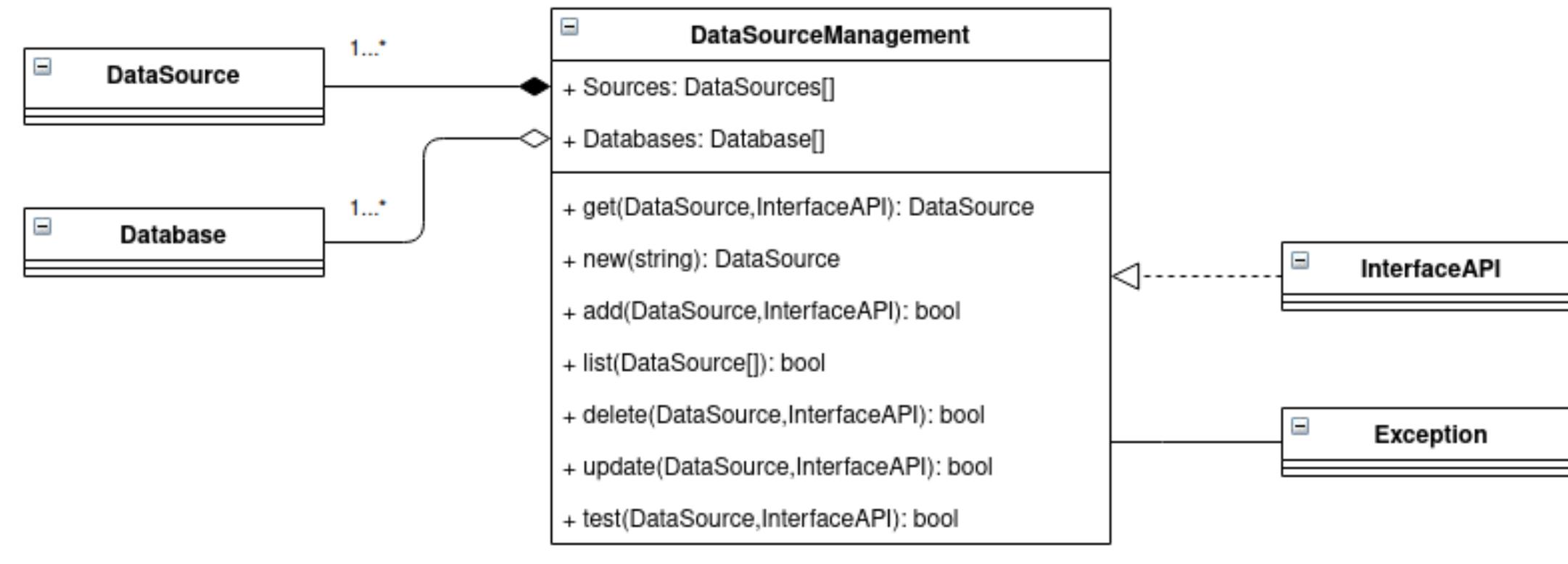
Container



Component



Class Diagram



Data Source Management
(UML Class)

Source: <https://www.diagrams.net/blog/c4-modelling>

- Understand the problem:
Domain storytelling, event storming, impact mapping etc
- Understand the systems and their context
C4, UML, sequence diagrams, CRC, etc
- **Record expensive decisions**

Architecture Decision Records

- Analyse your current context and strategy
- Keep doing all the above



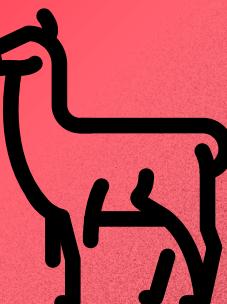
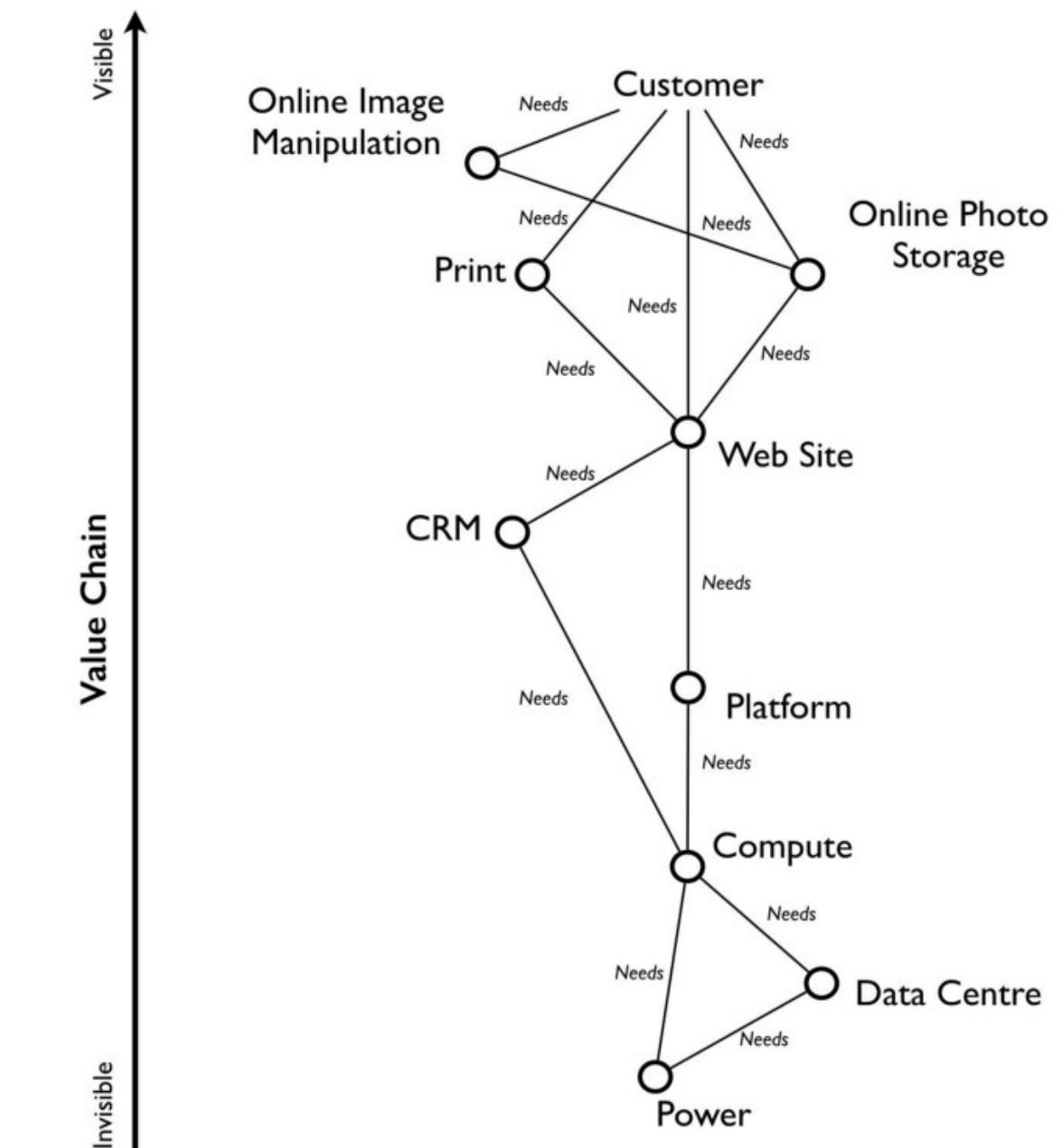
```
1 # Decision record template by Michael Nygard
2
3 This is the template in \[Documenting architecture decisions - Michael Nygard\]\(http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions\).
4 You can use \[adr-tools\]\(https://github.com/npryce/adr-tools\) for managing the ADR files.
5
6 In each ADR file, write these sections:
7
8 # Title
9
10 ## Status
11
12 What is the status, such as proposed, accepted, rejected, deprecated, superseded, etc.?
13
14 ## Context
15
16 What is the issue that we're seeing that is motivating this decision or change?
17
18 ## Decision
19
20 What is the change that we're proposing and/or doing?
21
22 ## Consequences
23
24 What becomes easier or more difficult to do because of this change?
25
```

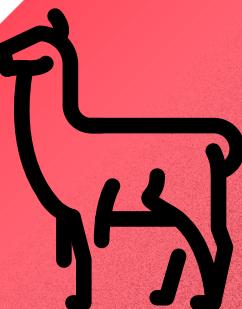
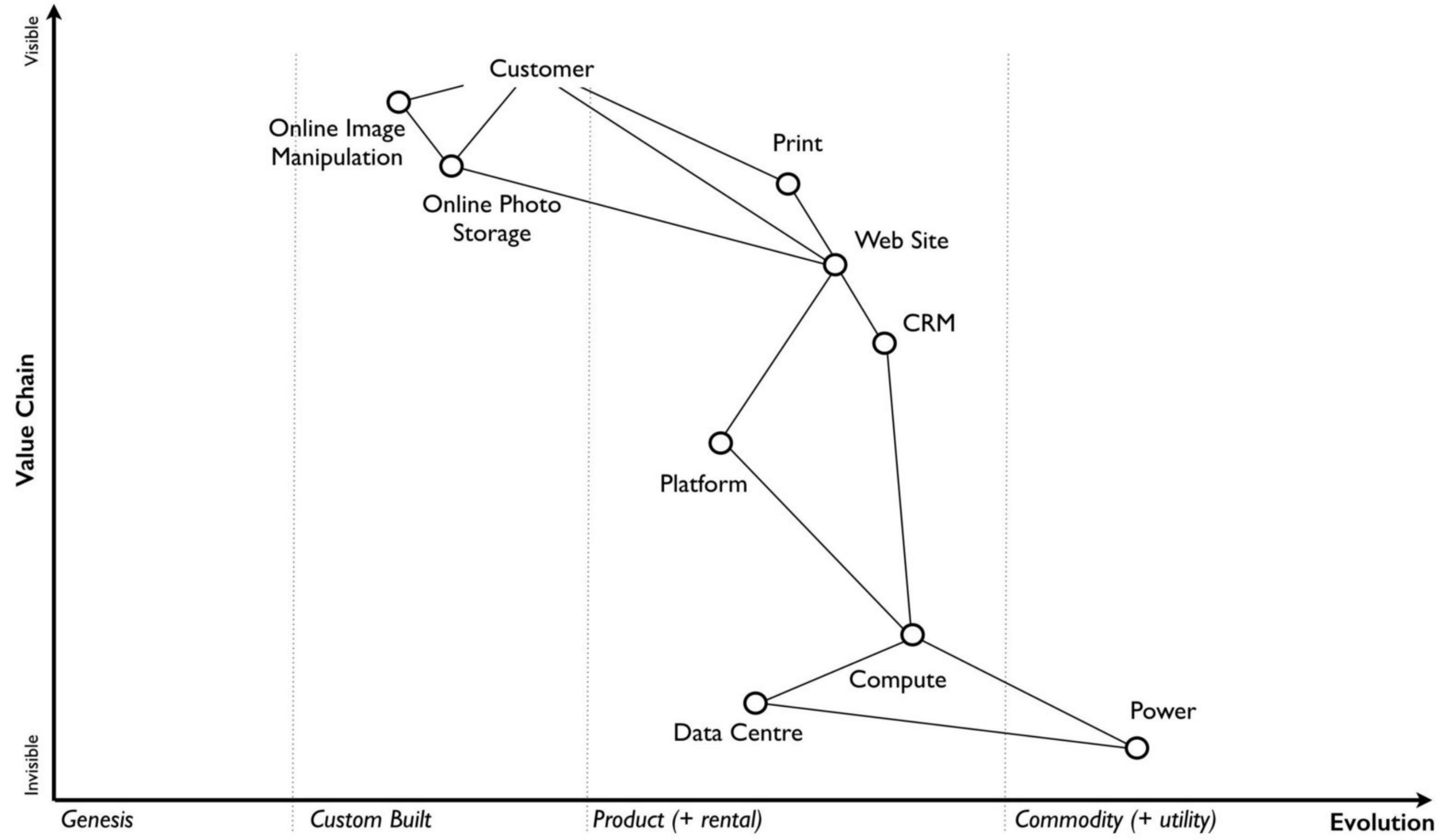
- Understand the problem:
Domain storytelling, event storming, impact mapping, etc
 - Understand the systems and their context
C4, UML, sequence diagrams, CRC, etc
 - Record expensive decisions
ADR
 - Analyse your current context and strategy
- ## Wardley maps
- Keep doing all the above



Value chain

- User(s)
- Situation





- Understand the problem:
Domain storytelling, event storming, impact mapping, etc
 - Understand the systems and their context
C4, UML, sequence diagrams, CRC, etc
 - Record expensive decisions
ADR
 - Analyse your current context and strategy
Wardley maps
- **Keep doing all the above:**
- Bytesize architecture sessions**

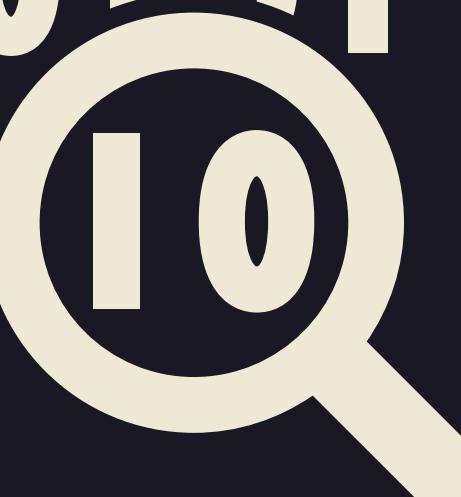




Bytesize Architecture Sessions

- Short
- Recurrent
- Alone together
- Consensus

I 00 I I 0
0 I 0 I 0 I
I 0 I 0



- Understand the problem:

Domain storytelling, event storming, impact mapping, etc

- Understand the systems and their context

C4, UML, sequence diagrams, CRC, etc

- Record expensive decisions

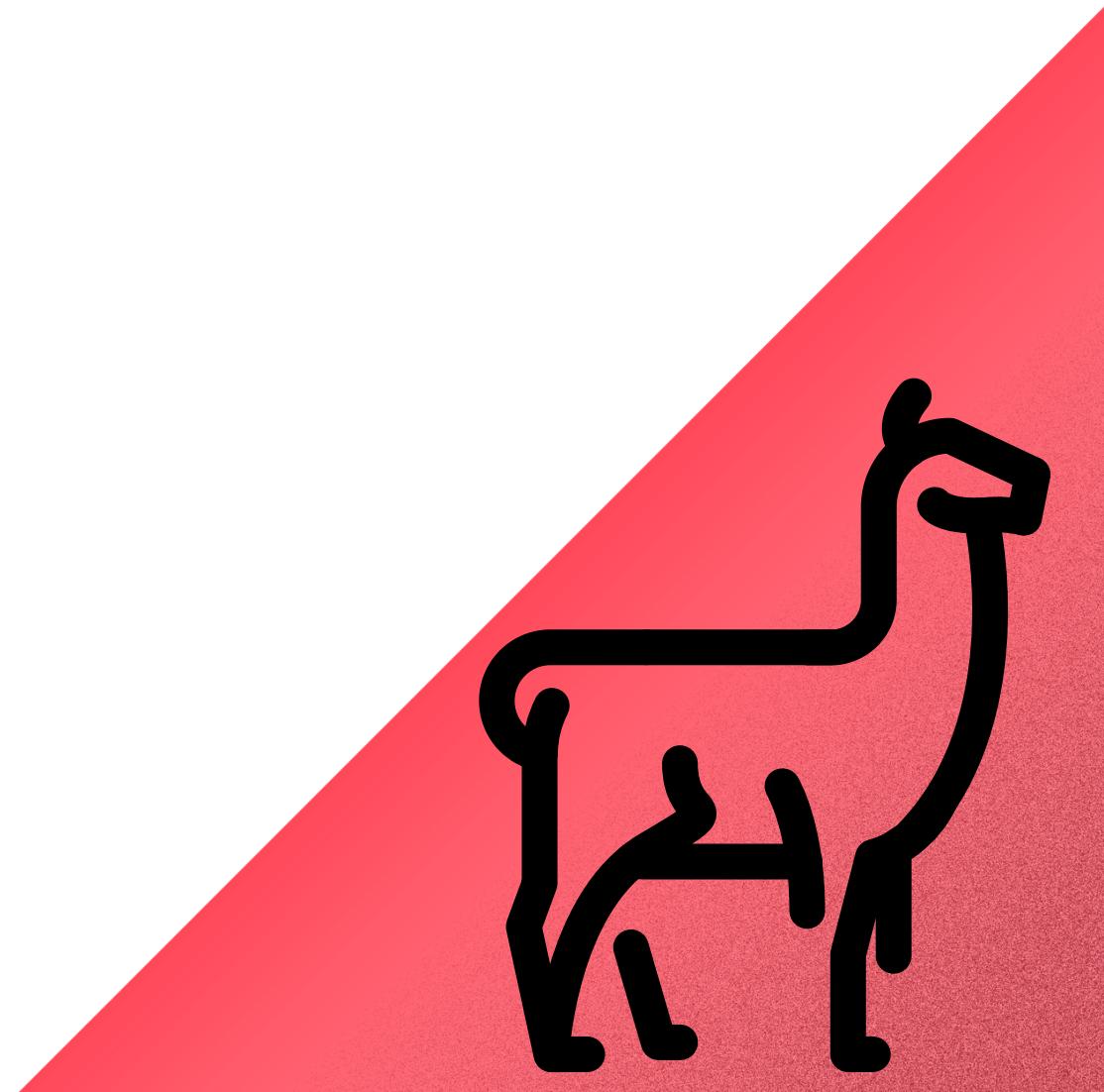
ADR

- Analyse your current context and strategy

Wardley maps

- Keep doing all the above:

Bytesize architecture sessions



THANKS TO YOU

SPECIAL THANKS TO:

- Alvaro Videla

Resources

- ADR resources and tooling <https://adr.github.io/>
- The C4 Model for Software Architecture <https://www.infoq.com/articles/C4-architecture-model/>
- Domain Storytelling: <https://domainstorytelling.org/>
 - Awesome Domain storytelling: <https://github.com/hofstef/awesome-domain-storytelling>
 - Domain Storytelling - Facilitator's Guide to Enhance Learning in Your Organization by Mufrid Krilic - <https://www.youtube.com/watch?v=ANfYEt16vRI>
- Bytesize Architecture Sessions <http://www.roundcrisis.com/2021/09/28/bytesize-architecture-sessions/>
- Key characteristics of systems Russel Ackoff: <https://www.youtube.com/watch?v=OqEeIG8aPPk>
- What is architecture <https://www.bredemeyer.com/whatis.htm>