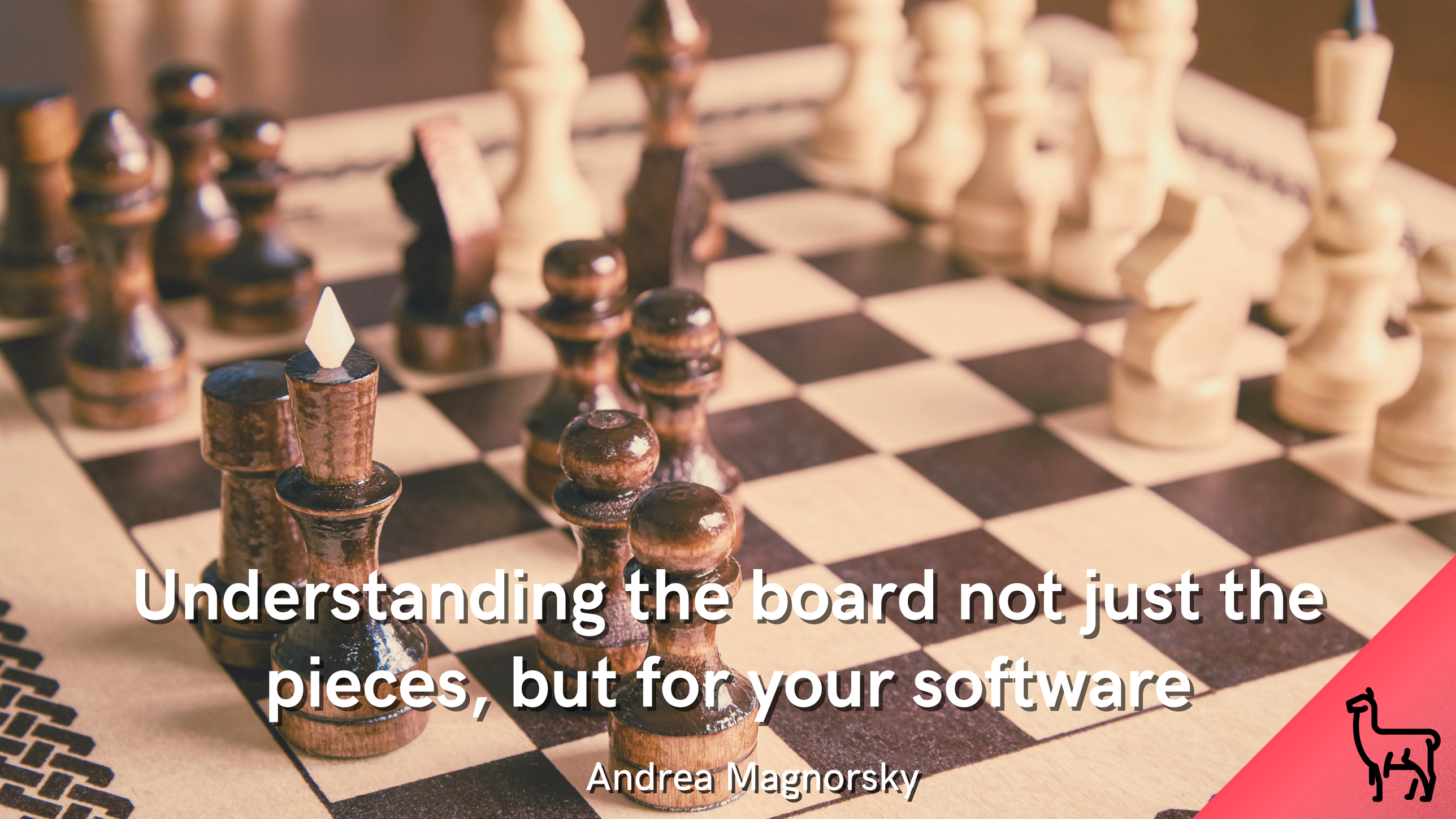


Understanding the board not just the pieces, but for your software

Andrea Magnorsky

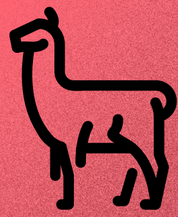
About Andrea

Programming professionally for 20 years

- Finance
- Video Games
- TV, etc

I am a consultant specialising in Functional programming and Software Architecture.

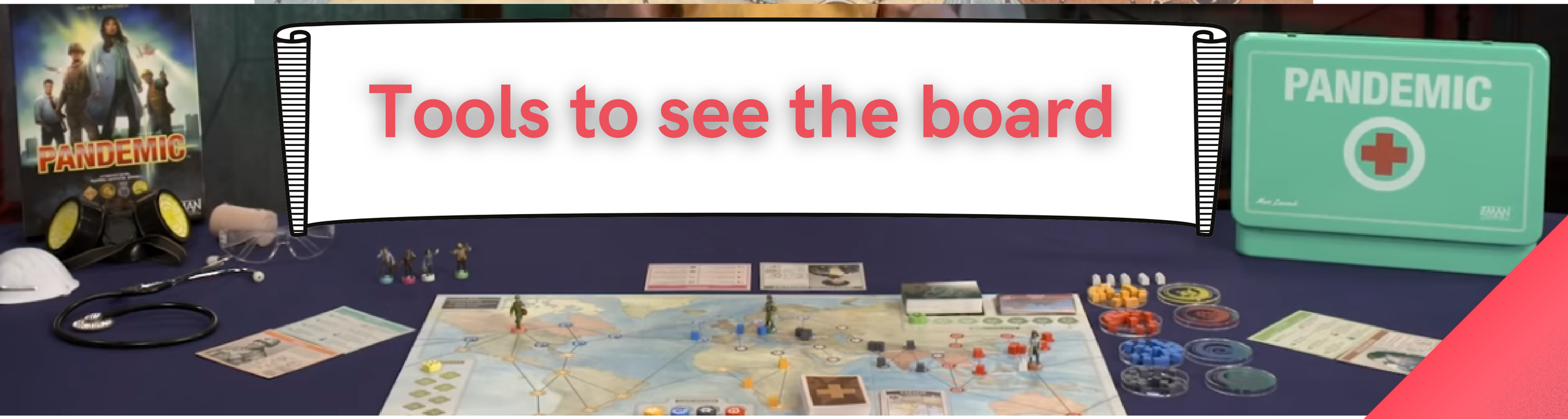
roundcrisis.com



Questions?

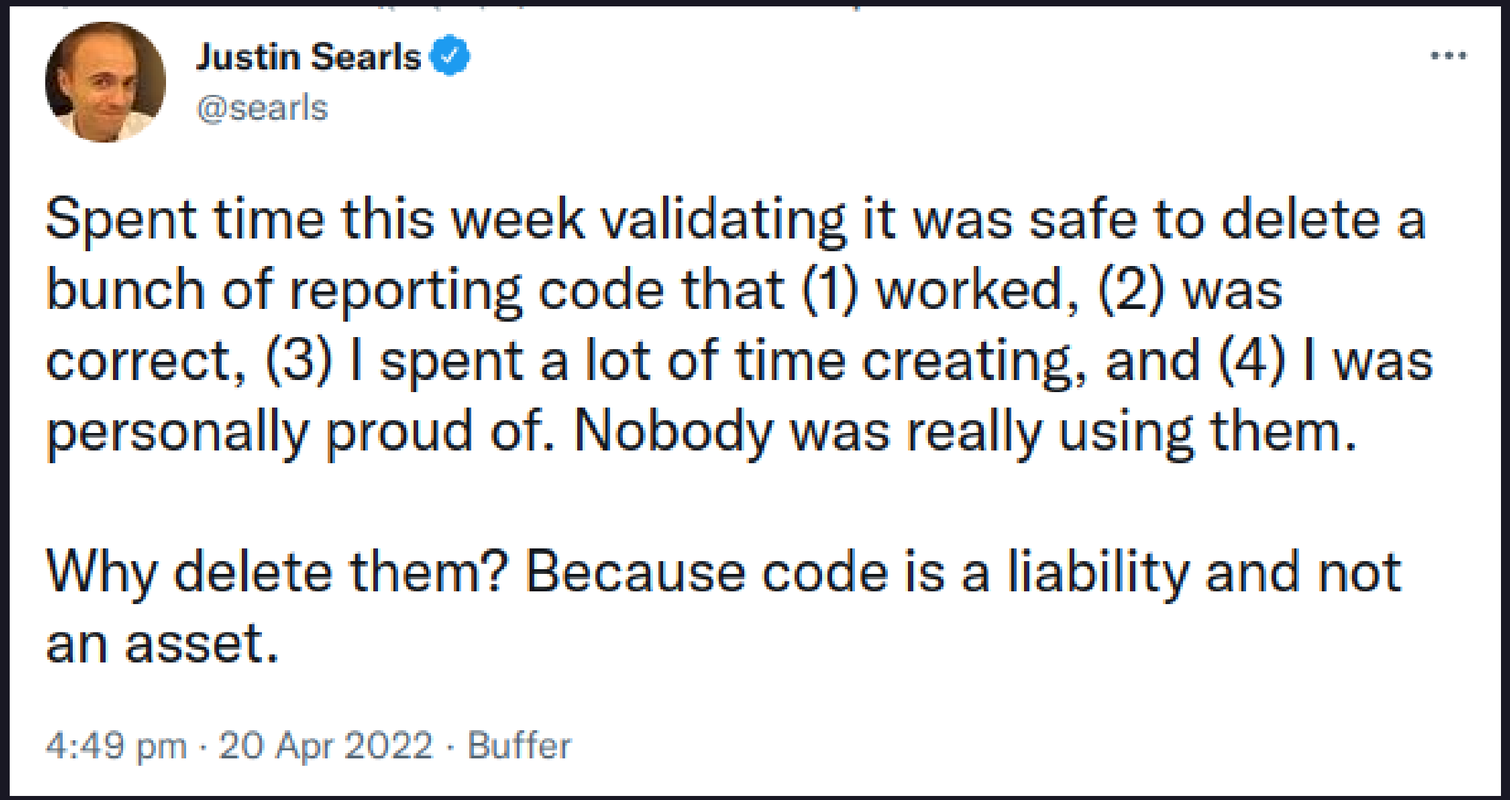
slido.com and enter **#569895**

Tools to see the board



Strategy

Architecture



source: https://twitter.com/searls/status/1516806149940686856

related: https://wiki.c2.com/?SoftwareAsLiability

HATS

Like planning, modelling is it's own reward

- [ ] Analyse high level context

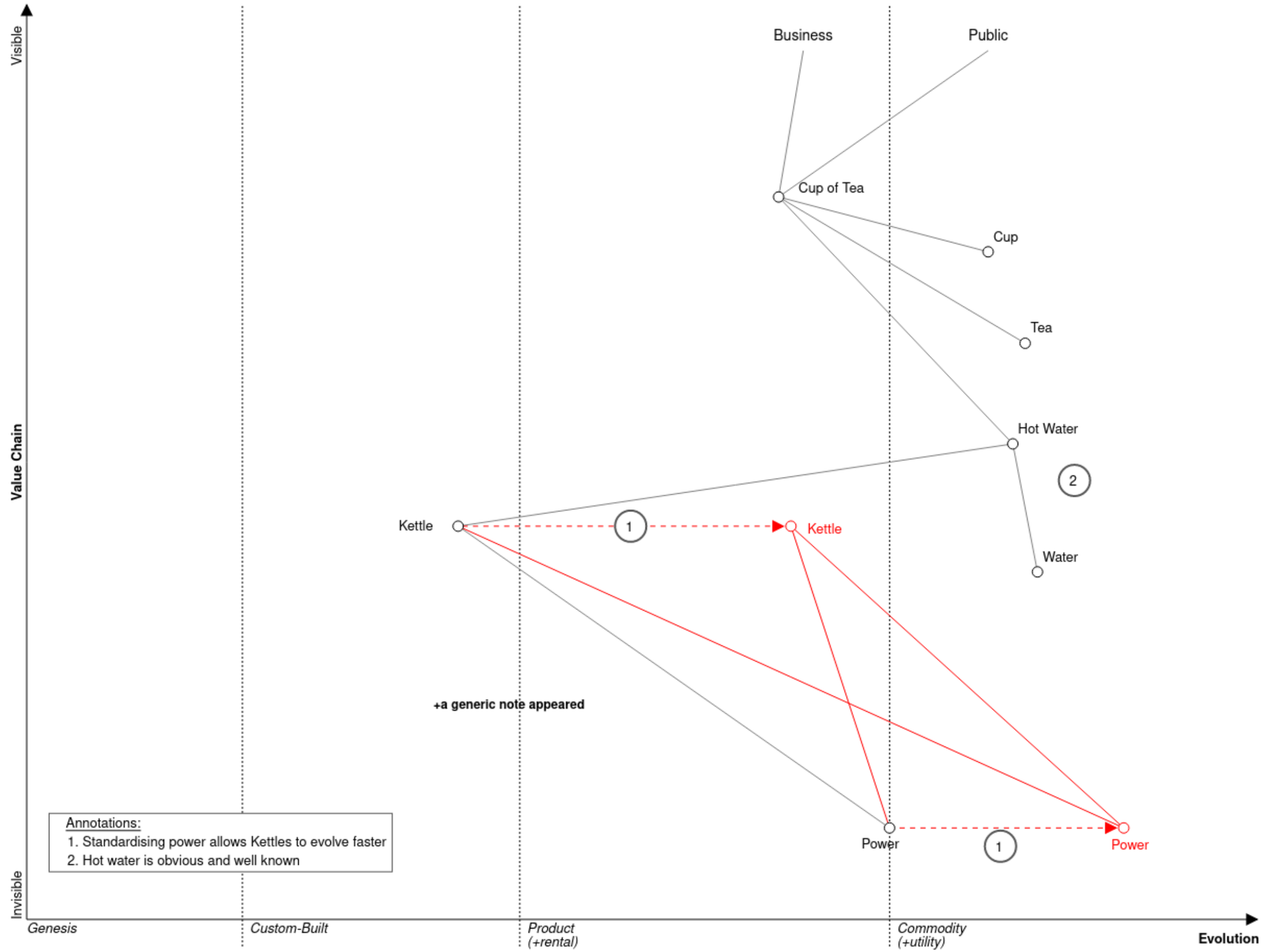
Wardley maps

- [ ] Understand the problem
- [ ] Understand the your systems and their interactions
- [ ] Record expensive decisions
- [ ] Keep doing all the above

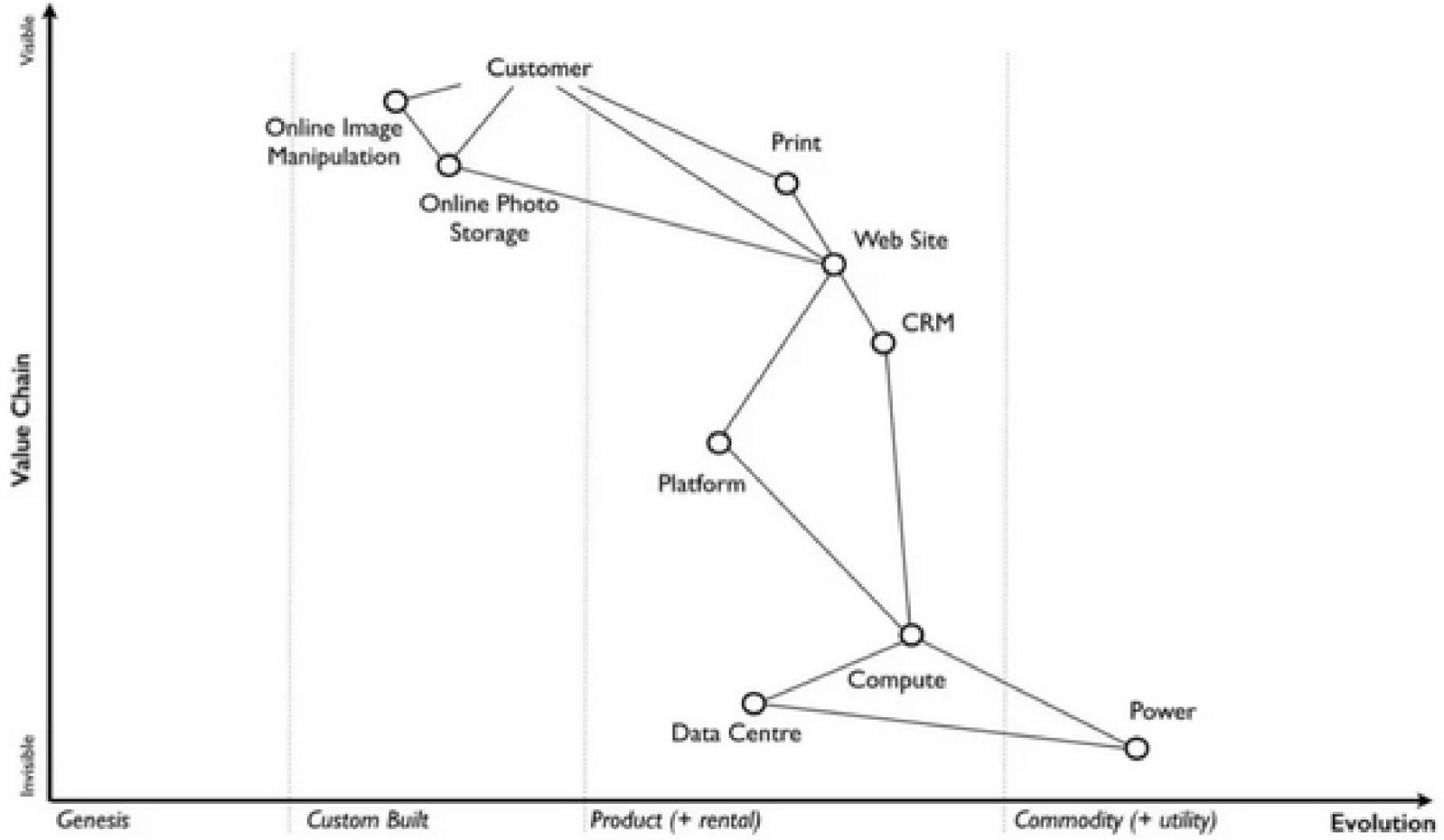
Wardley Maps

- Visualisation of systems
- Patterns of evolution
- For thinking and making decisions

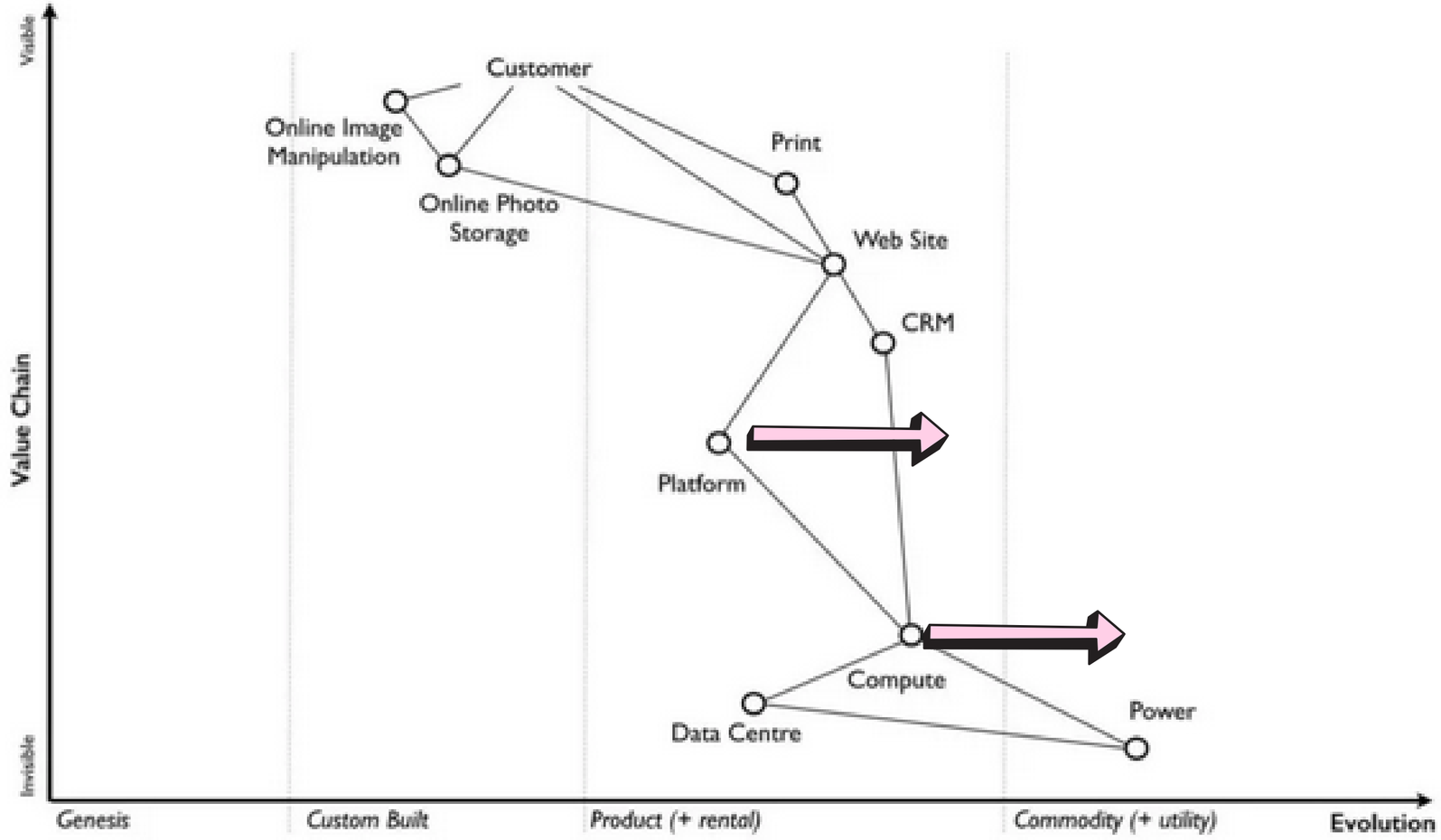
Tea Shop


Visible
Value Chain
Invisible
Business
Public
Cup of Tea
Cup
Tea
Hot Water
2
Kettle
1
Kettle
Water
+a generic note appeared
Power
1
Power
Annotations:
1. Standardising power allows Kettles to evolve faster
2. Hot water is obvious and well known
Genesis
Custom-Built
Product
(+rental)
Commodity
(+utility)
Evolution

Wardley Maps



Wardley Maps



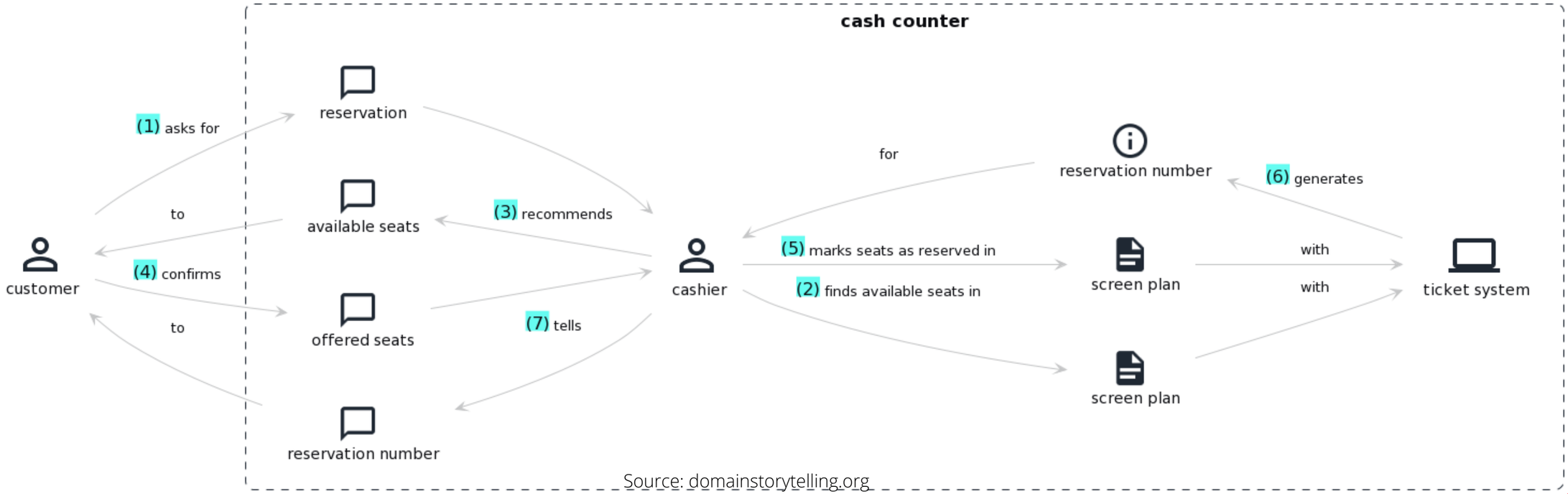
- Analyse high level context:
  Wardley maps
- **Understand the problem:**

Domain storytelling, event storming, etc

- Understand your systems and their interactions
- Record expensive decisions

- Keep doing all the above

DOMAIN STORYTELLING

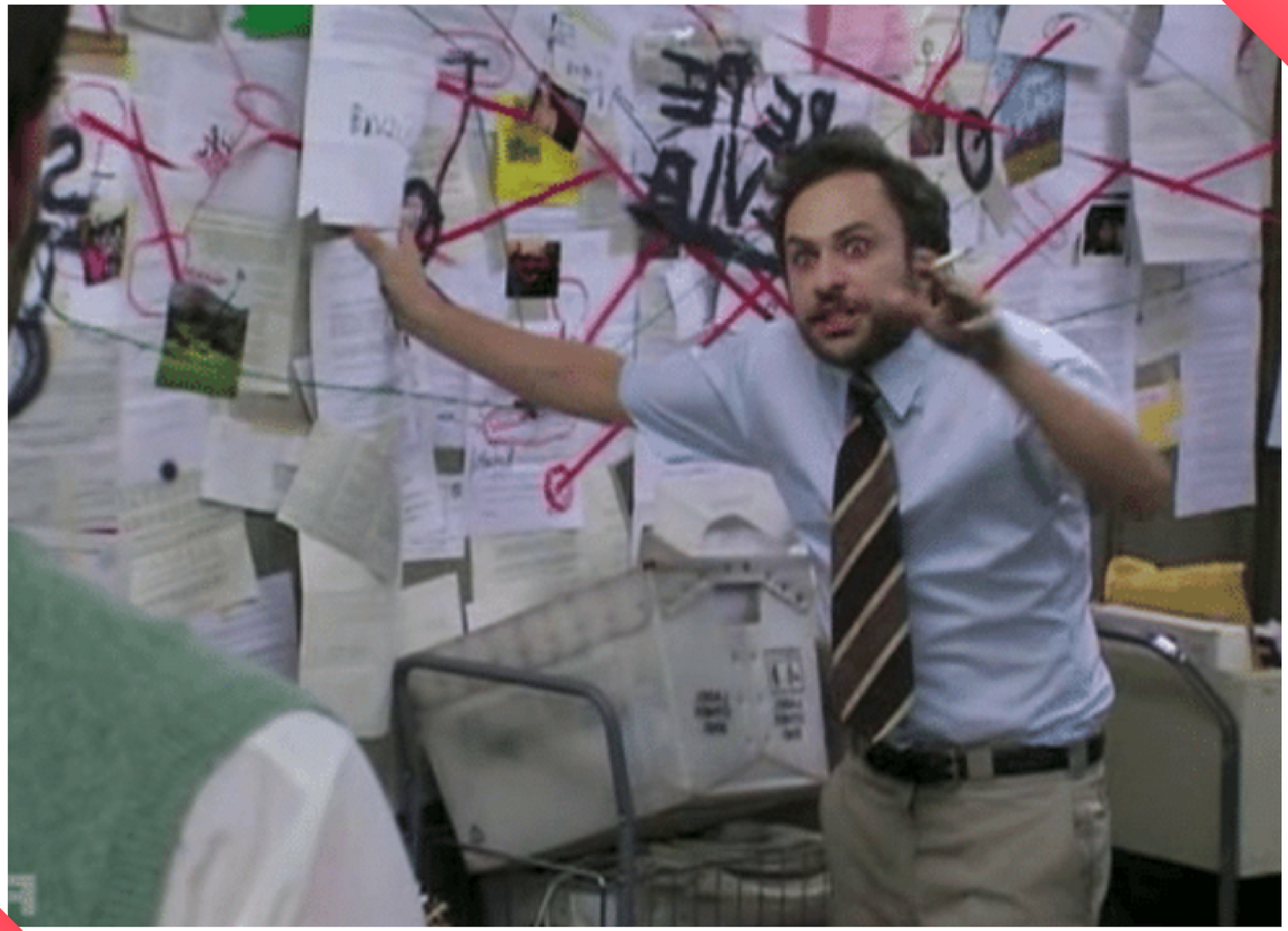


Source: domainstorytelling.org

DOMAIN STORYTELLING

- Fully align all project participants and stakeholders
- Draw clear boundaries
- Get better requirements
- Better visibility

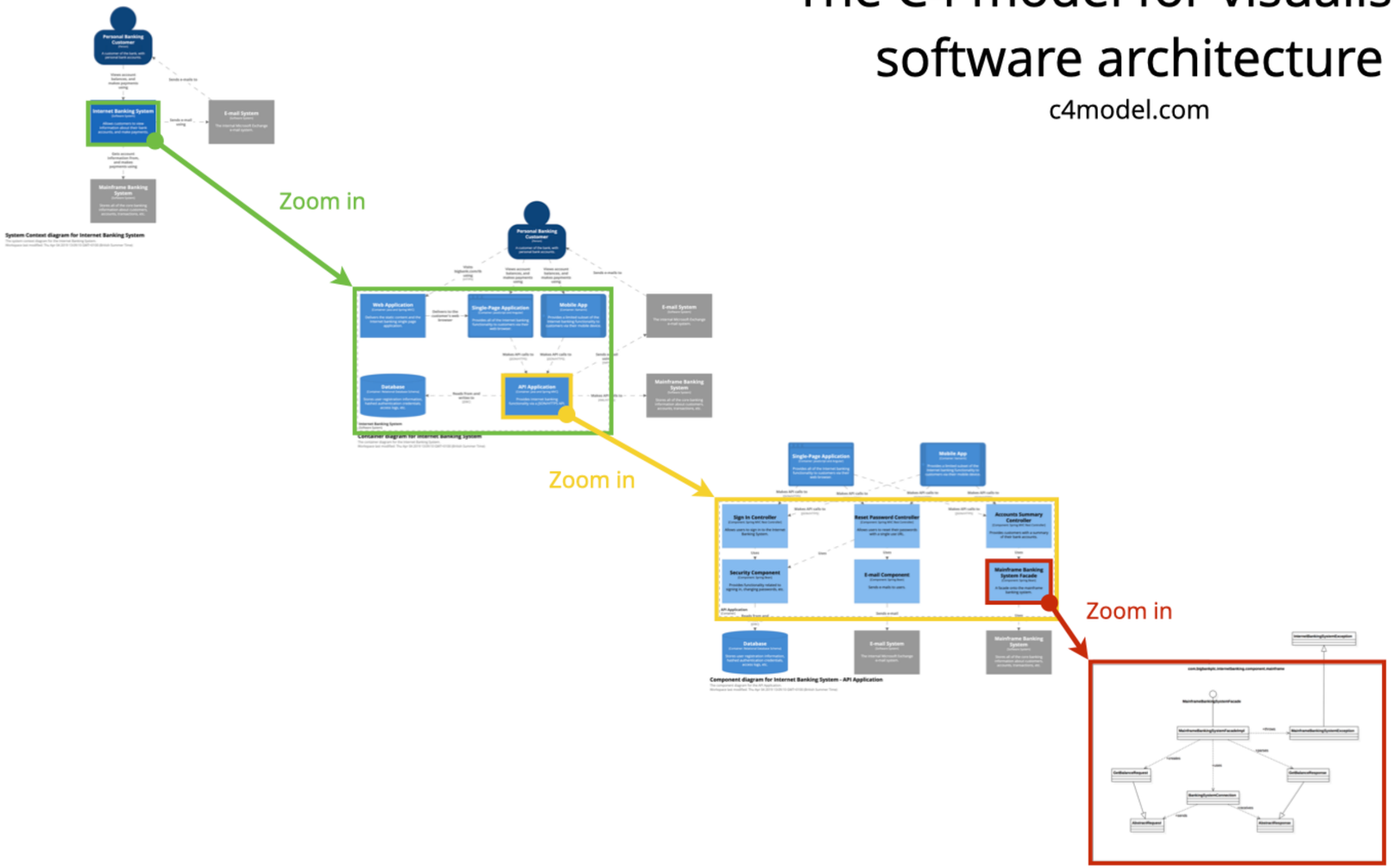
- Analyse high level context : Wardley maps
- Understand the problem: Domain storytelling, event storming, impact mapping, etc
- **Understand your systems and their interactions: C4, UML, sequence diagrams, CRC, etc**
- Record expensive decisions
- Keep doing all the above



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

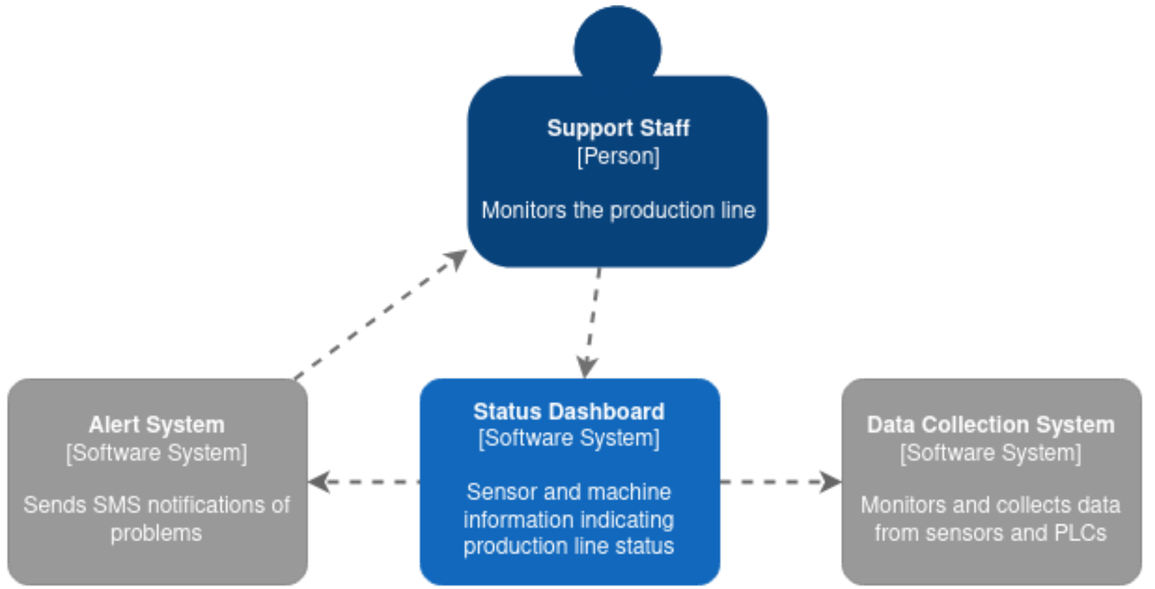
The C4 model for visualising software architecture

c4model.com

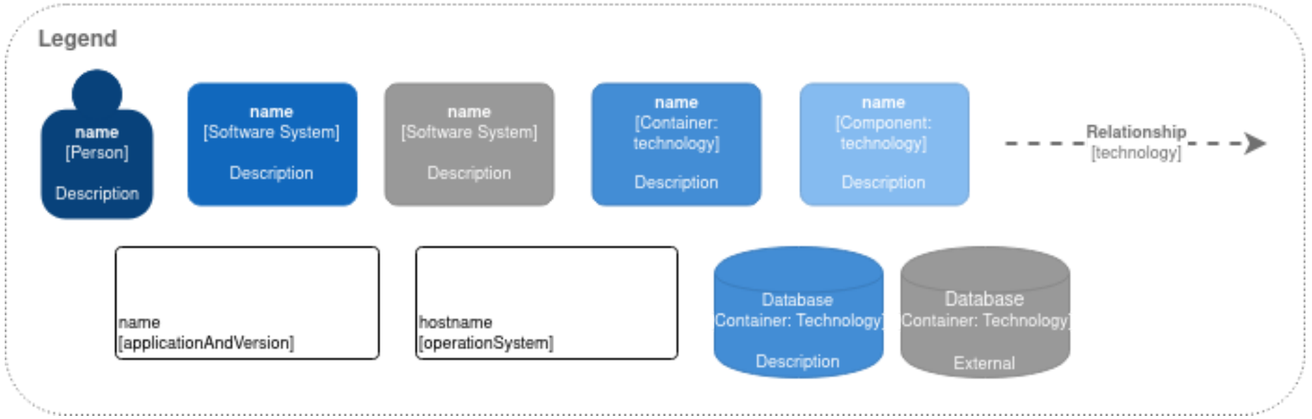


Level 1	Level 2	Level 3	Level 4
Context	Containers	Components	Code

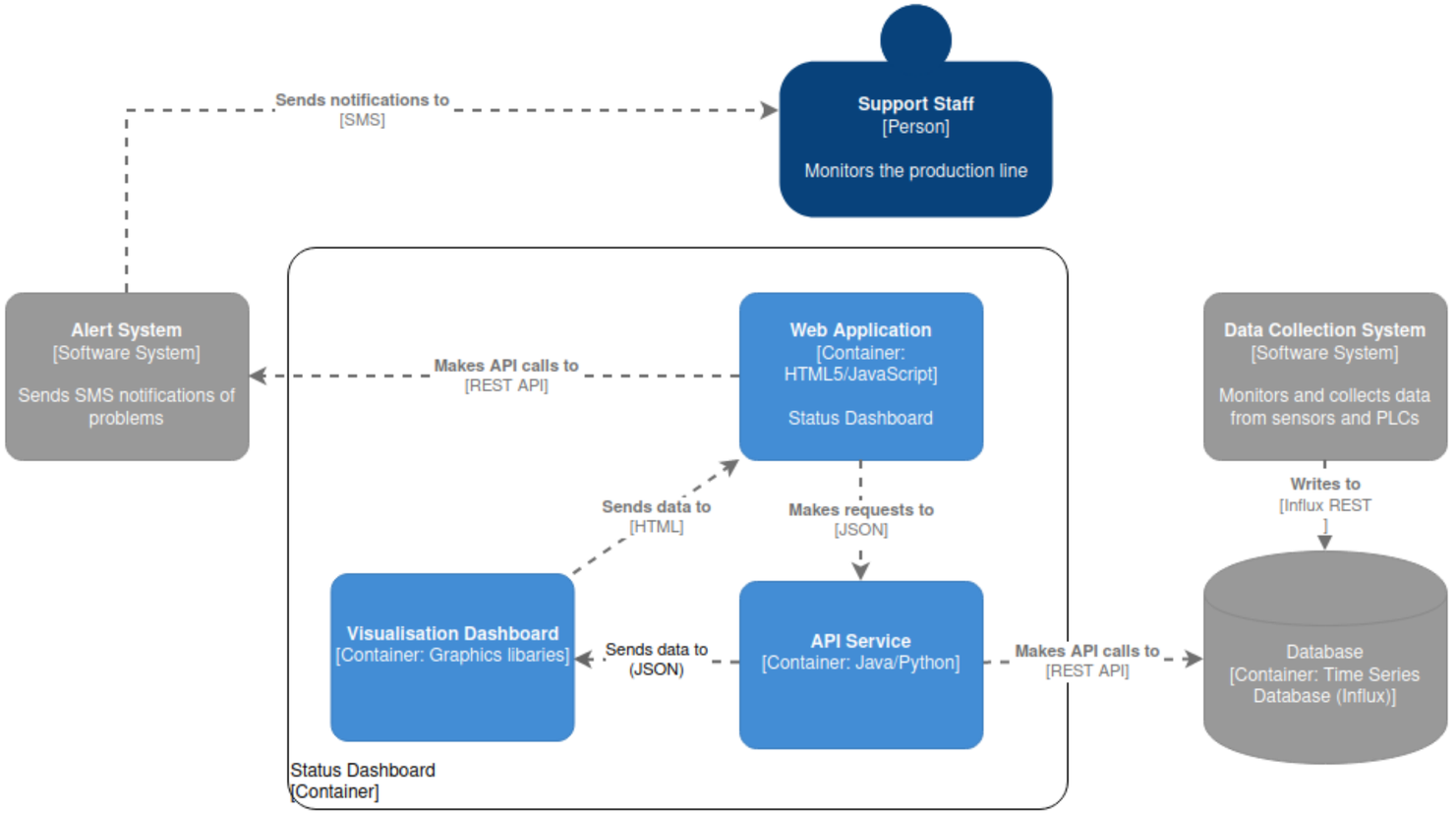
Context



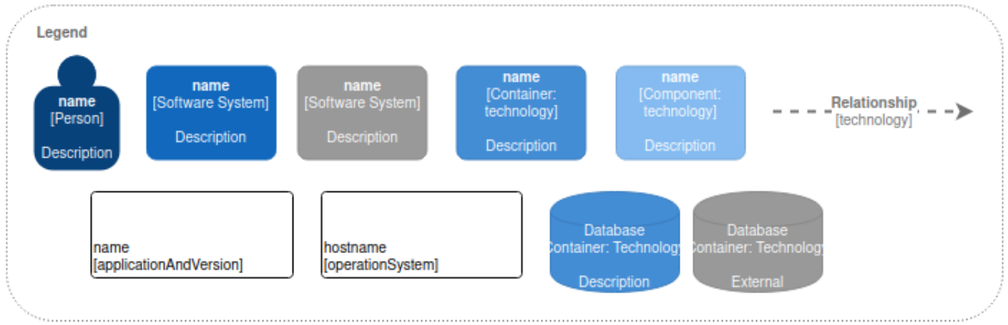
Status Dashboard (Context)

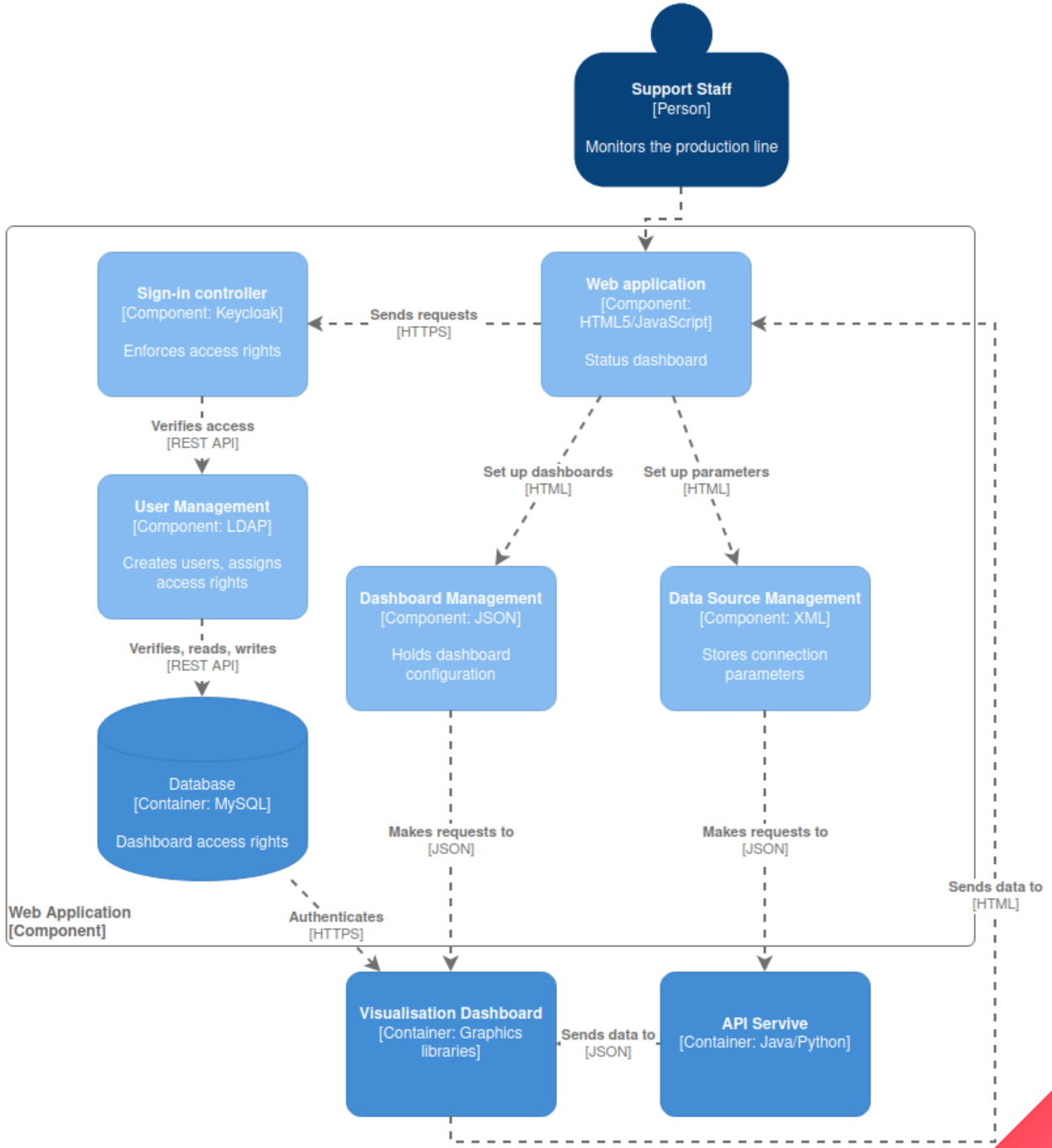


Container

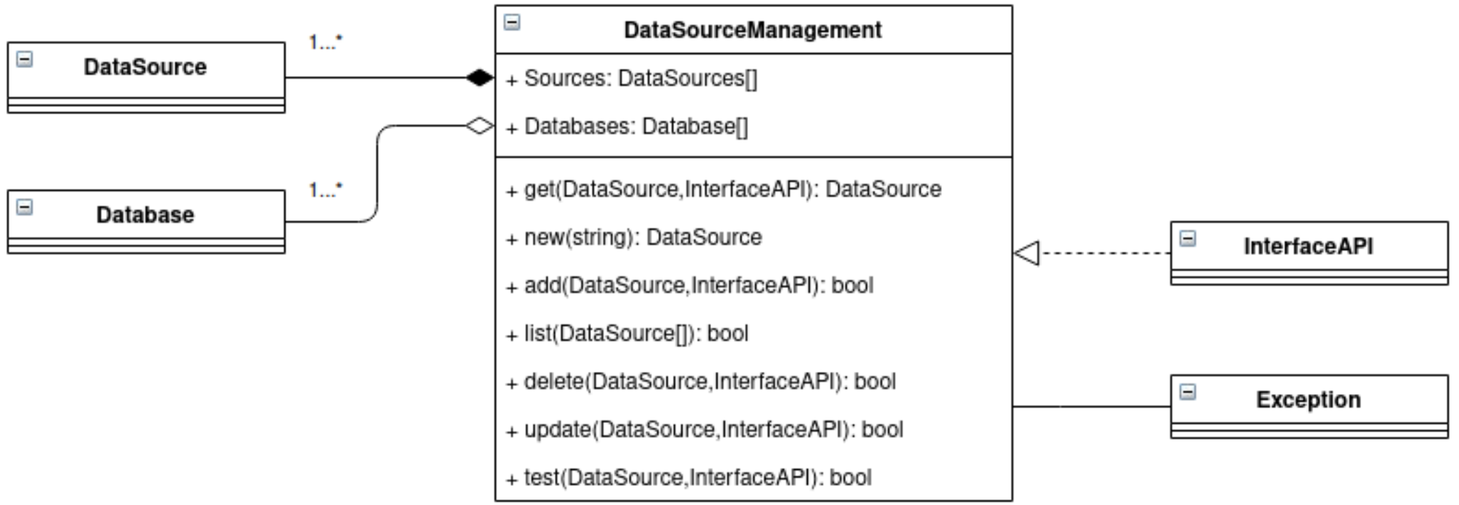

Sends notifications to
[SMS]
Support Staff
[Person]
Monitors the production line
Alert System
[Software System]
Sends SMS notifications of problems
Makes API calls to
[REST API]
Web Application
[Container: HTML5/JavaScript]
Status Dashboard
Data Collection System
[Software System]
Monitors and collects data from sensors and PLCs
Writes to
[Influx REST
]
Sends data to
[HTML]
Makes requests to
[JSON]
Visualisation Dashboard
[Container: Graphics libaries]
Sends data to
(JSON)
API Service
[Container: Java/Python]
Makes API calls to
[REST API]
Database
[Container: Time Series Database (Influx)]
Status Dashboard
[Container]

Component


Legend
name
[Person]
Description
name
[Software System]
Description
name
[Software System]
Description
name
[Container: technology]
Description
name
[Component: technology]
Description
Relationship
[technology]
name
[applicationAndVersion]
hostname
[operationSystem]
Database
Container: Technology
Description
Database
Container: Technology
External


Support Staff
[Person]
Monitors the production line
Sign-in controller
[Component: Keycloak]
Enforces access rights
Sends requests
[HTTPS]
Web application
[Component: HTML5/JavaScript]
Status dashboard
Verifies access
[REST API]
Set up dashboards
[HTML]
Set up parameters
[HTML]
User Management
[Component: LDAP]
Creates users, assigns access rights
Dashboard Management
[Component: JSON]
Holds dashboard configuration
Data Source Management
[Component: XML]
Stores connection parameters
Verifies, reads, writes
[REST API]
Database
[Container: MySQL]
Dashboard access rights
Makes requests to
[JSON]
Makes requests to
[JSON]
Web Application
[Component]
Authenticates
[HTTPS]
Sends data to
[HTML]
Visualisation Dashboard
[Container: Graphics libraries]
Sends data to
[JSON]
API Servive
[Container: Java/Python]

Class Diagram



Data Source Management
(UML Class)

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

- Analyse high level context:

Wardley maps

- Understand the problem:

Domain storytelling, event storming, etc

- Understand your systems and their interactions: C4, UML, sequence diagrams, etc
- Record expensive decisions

Architecture Decision Records

- Keep doing all the above

```
# Decision record template by Michael Nygard

This is the template in [Documenting architecture decisions - Michael Nygard](http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions).
You can use [adr-tools](https://github.com/npryce/adr-tools) for managing the ADR files.

In each ADR file, write these sections:

# Title

## Status

What is the status, such as proposed, accepted, rejected, deprecated, superseded, etc.?

## Context

What is the issue that we're seeing that is motivating this decision or change?

## Decision

What is the change that we're proposing and/or doing?

## Consequences

What becomes easier or more difficult to do because of this change?

```

Secrets Storage example

```
## Summary

### Issue

We need to store secrets, such as passwords, private keys, authentication tokens, etc.

Some of the secrets are user-oriented. For example, our developer wants to be able to use their mobile pho

Some of the secrets are system-oriented. For example, our continuous delivery pipeline needs to be able to

### Decision

Bitwarden for user-oriented secrets

Vault by HashiCorp for system-oriented secrets.

### Status

Decided. We are open to new alternatives as they arise.

## Details

### Assumptions

For this purpose, and our current state, we value user-oriented convenience, such as usable mobile apps.

  * We want to ensure fast easy access on the go, such as for a developer doing on-call system reliability
```

source: https://github.com/joelparkerhenderson/architecture-decision-record/

- Analyse high level context:

Wardley maps

- Understand the problem:

Domain storytelling, event storming, etc

- Understand your systems and their

interactions: C4, UML, sequence diagrams, etc

- Record expensive decisions

ADR

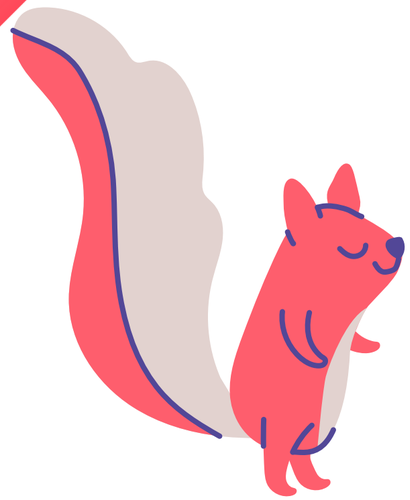
- Keep doing all the above:

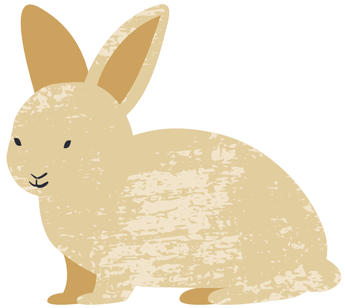
Bytesize architecture sessions

Bytesize Architecture Sessions

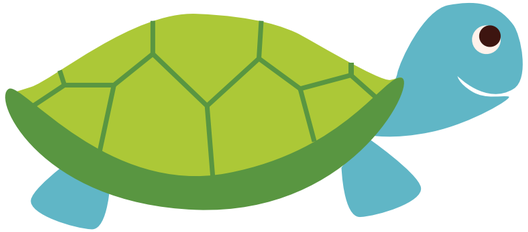
Why do this?

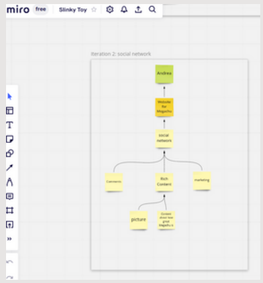
- Avoid silos of knowledge
- Avoid design by "boiling the ocean"



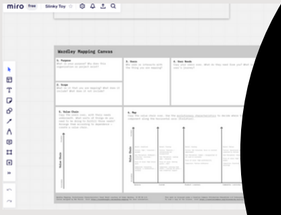


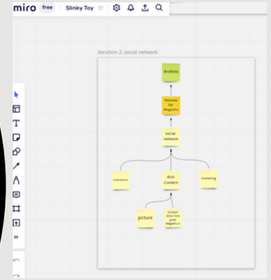
The team

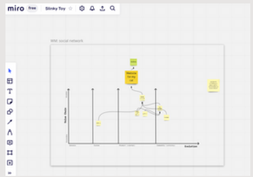



miro free Slinky Toy
Iteration 2: social network
Andrea
Welcome Your Magazine
social network
Comments
Rich Content
Marketing
picture
Contact area and grab support

miro free Slinky Toy
WestDay Mapping Canvas

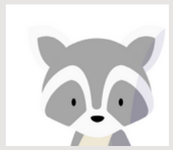

miro free Slinky Toy
WestDay Mapping Canvas

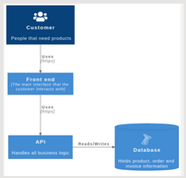

miro free Slinky Toy
Iteration 2: social network
Andrea
Welcome Your Magazine
social network
Comments
Rich Content
Marketing
picture
Contact area and grab support


miro free Slinky Toy

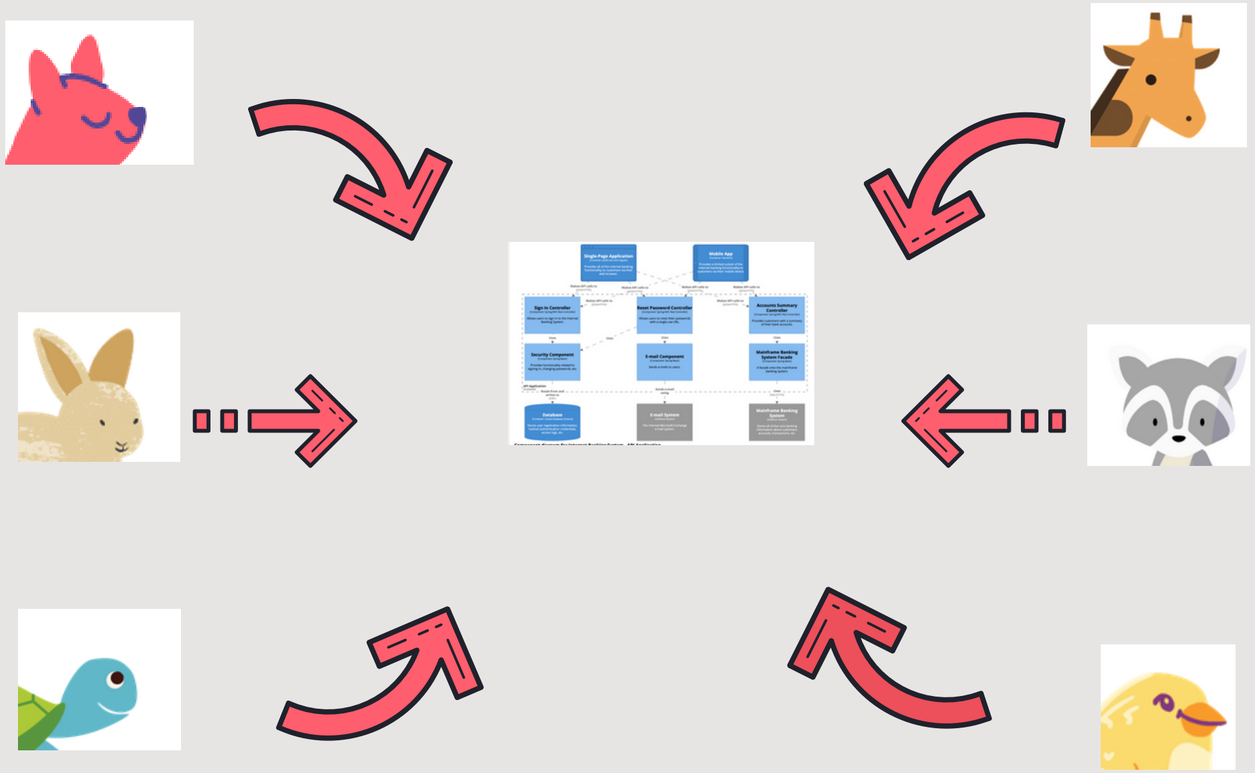
miro free Slinky Toy

Consensus




Customer
People that need products
Front end
API
Handles all business logic
Reads/Writes
Database
Holds product, order and invoice information

Iteration


Single-Page Application
Mobile App
Sign In Controller
Reset Password Controller
Accounts Summary Controller
Security Component
E-mail Component
Mainframe Banking System Facade
Database
E-mail System
Mainframe Banking System

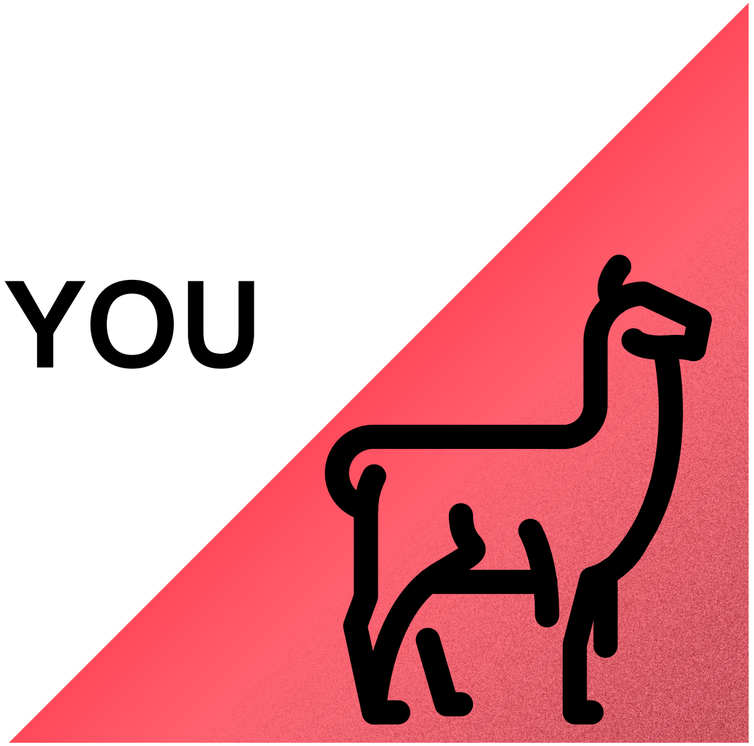
Bytesize Architecture Sessions

How does it work?

- Short meeting
- Recurrent
- Alone together
- Consensus

- Analyse high level context
- Understand the problem
- Understand your systems and their interactions
- Record expensive decisions
- Keep doing all the above

THANK YOU



Questions?

slido.com and enter **#569895**

Resources

- Domaing Storytelling: https://domainstorytelling.org/quick-start-guide
  - 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
- The C4 Model for Software Architecture https://c4model.com/
- Bytesize Architecture Sessions http://www.roundcrisis.com/2021/09/28/bytesize-architecture-sessions/
- ADR resources and tooling https://adr.github.io/
- Key characteristics of systems Russel Ackoff: https://www.youtube.com/watch?v=OqEeIG8aPPk
- What is architecture https://www.bredemeyer.com/whatis.htm

Tools

- Online Wardley maps https://onlinewardleymaps.com/
- Domain Storytelling: https://www.wps.de/modeler/
- PlantUML for C4 Diagrams : https://github.com/plantuml-stdlib/C4-PlantUML