

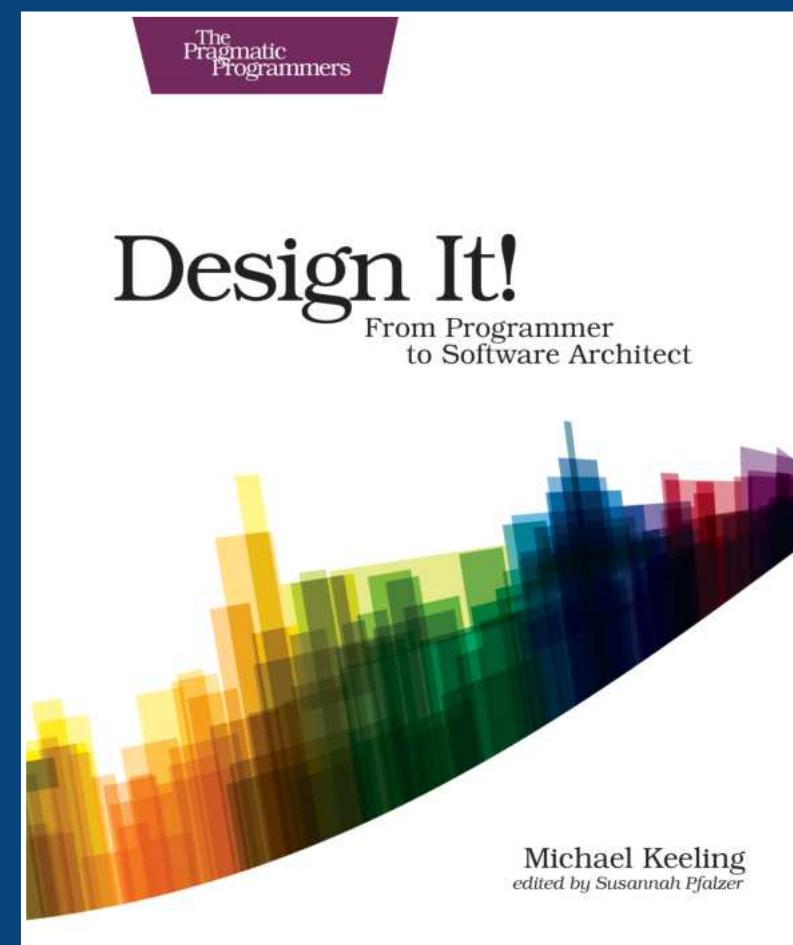
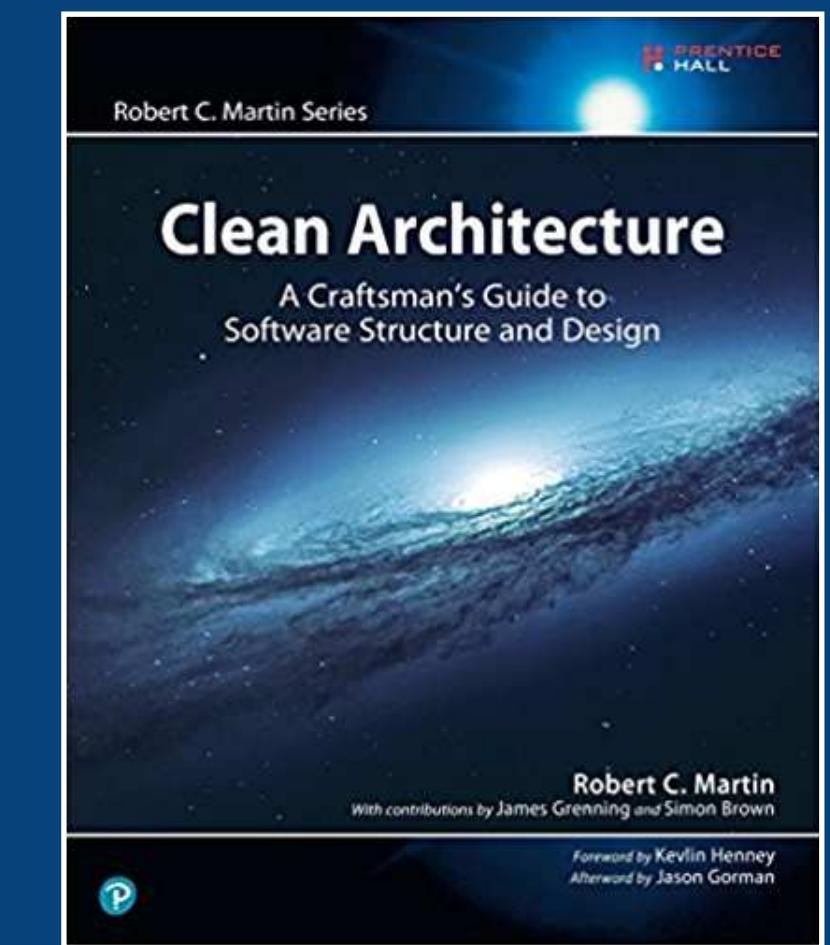
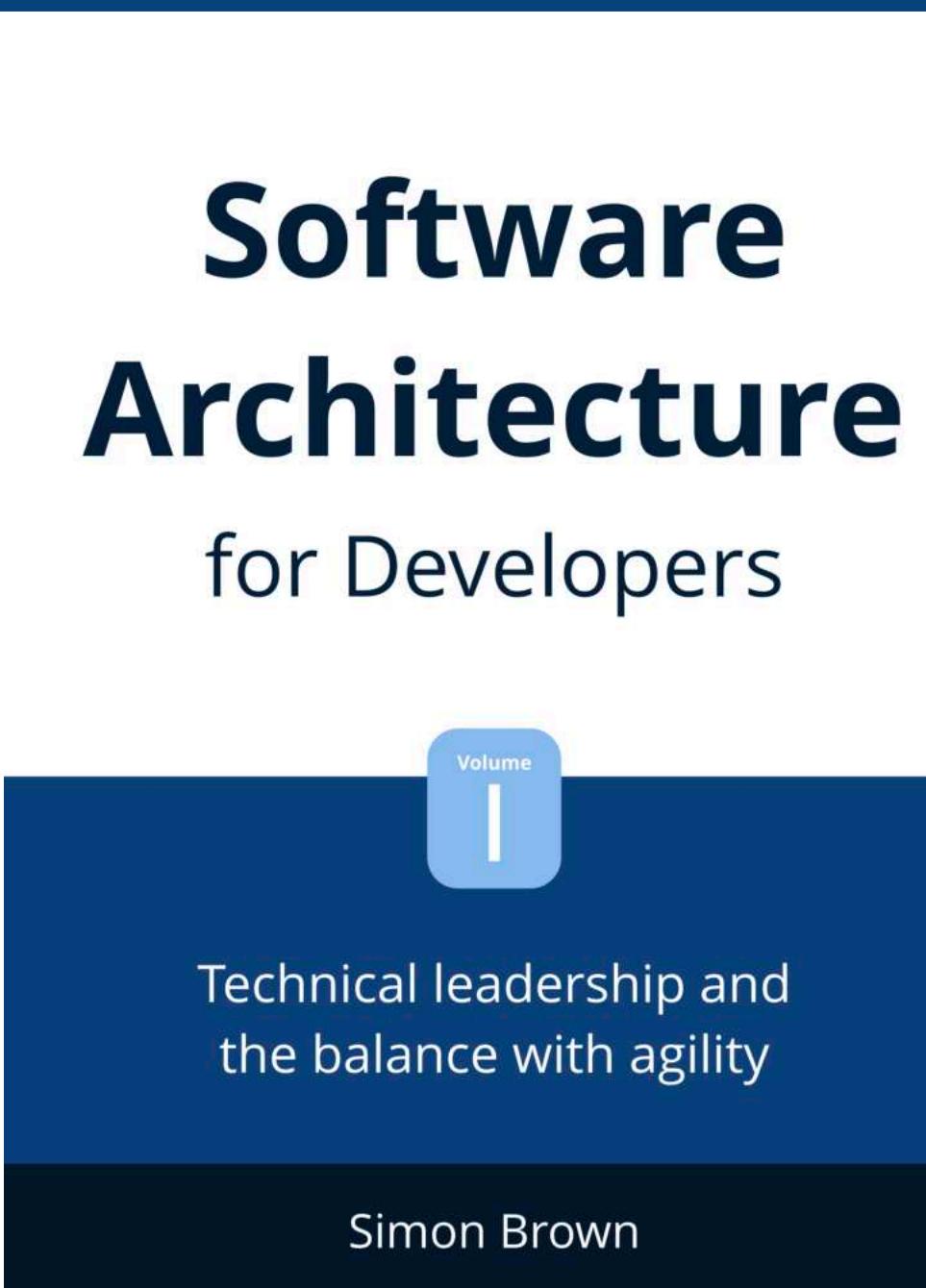
Visualising software architecture with the C4 model

@simonbrown

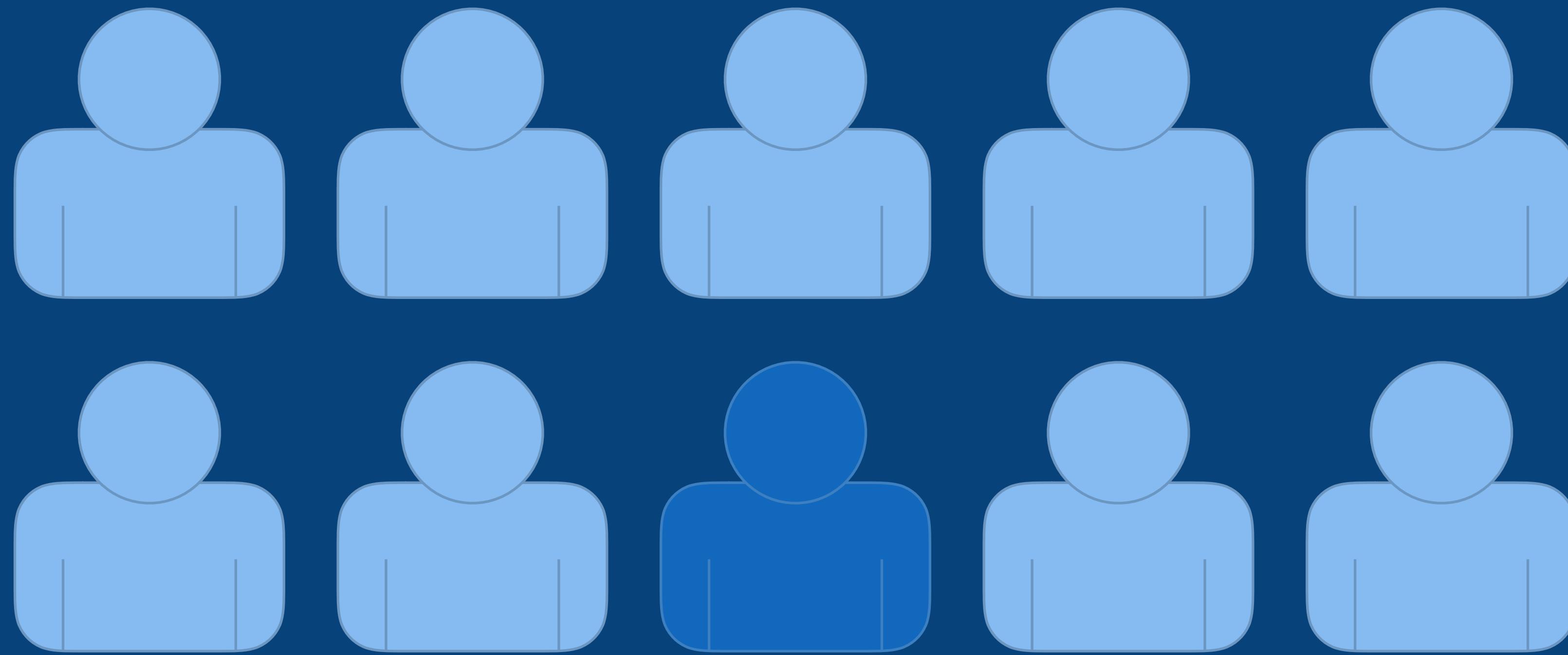
Simon Brown

Independent consultant specialising in software architecture,
plus the creator of the C4 model and Structurizr

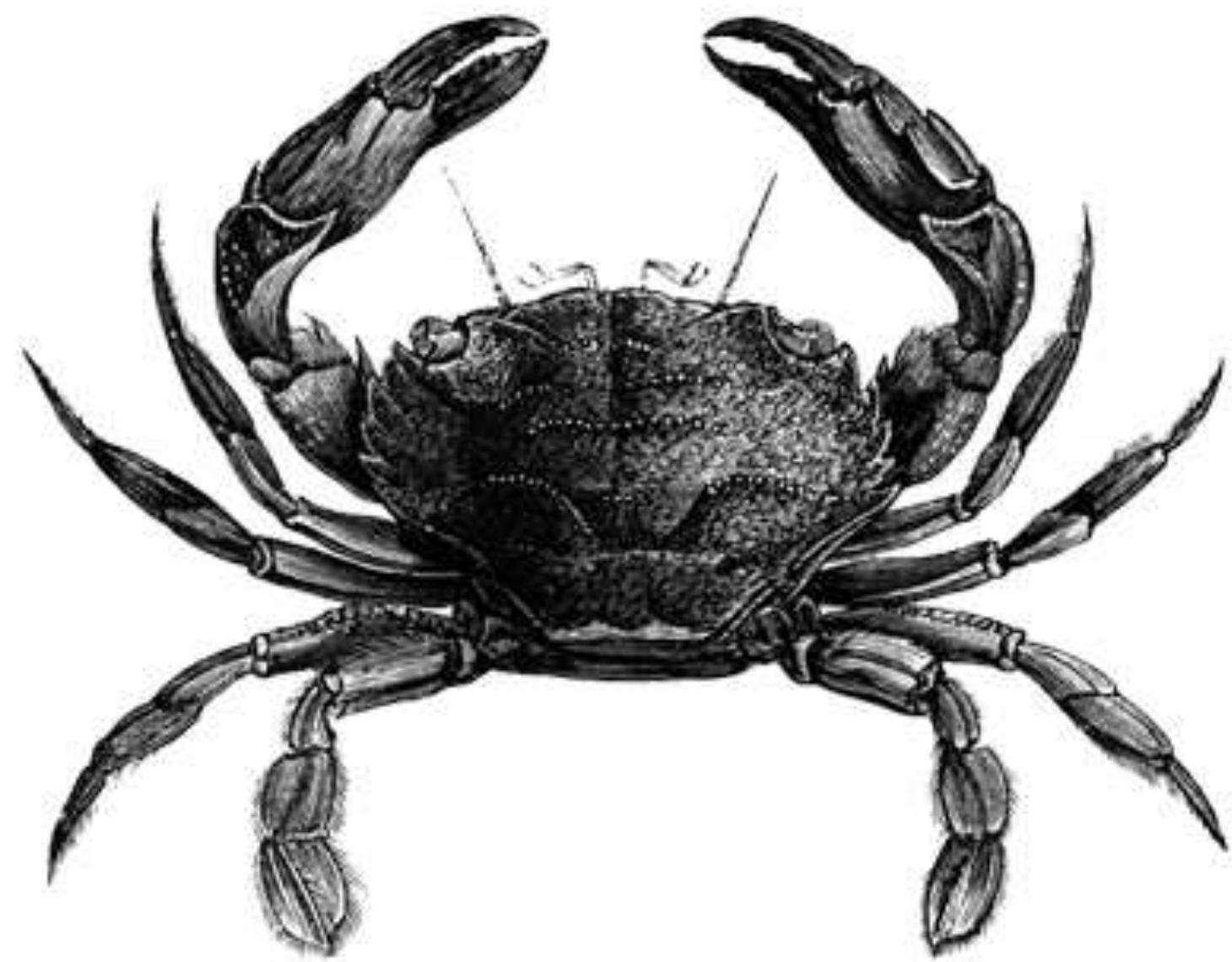
@simonbrown



Do you use UML?



In my experience, optimistically,
1 out of 10 people use UML



97 Ways to Sidestep UML

O RLY?

Knowfa Mallity

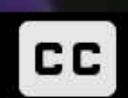
- #2 "Not everybody else on the team knows it."
- #3 "I'm the only person on the team who knows it."
- #36 "You'll be seen as old."
- #37 "You'll be seen as old-fashioned."
- #66 "The tooling sucks."
- #80 "It's too detailed."
- #81 "It's a very elaborate waste of time."
- #92 "It's not expected in agile."
- #97 "The value is in the conversation."

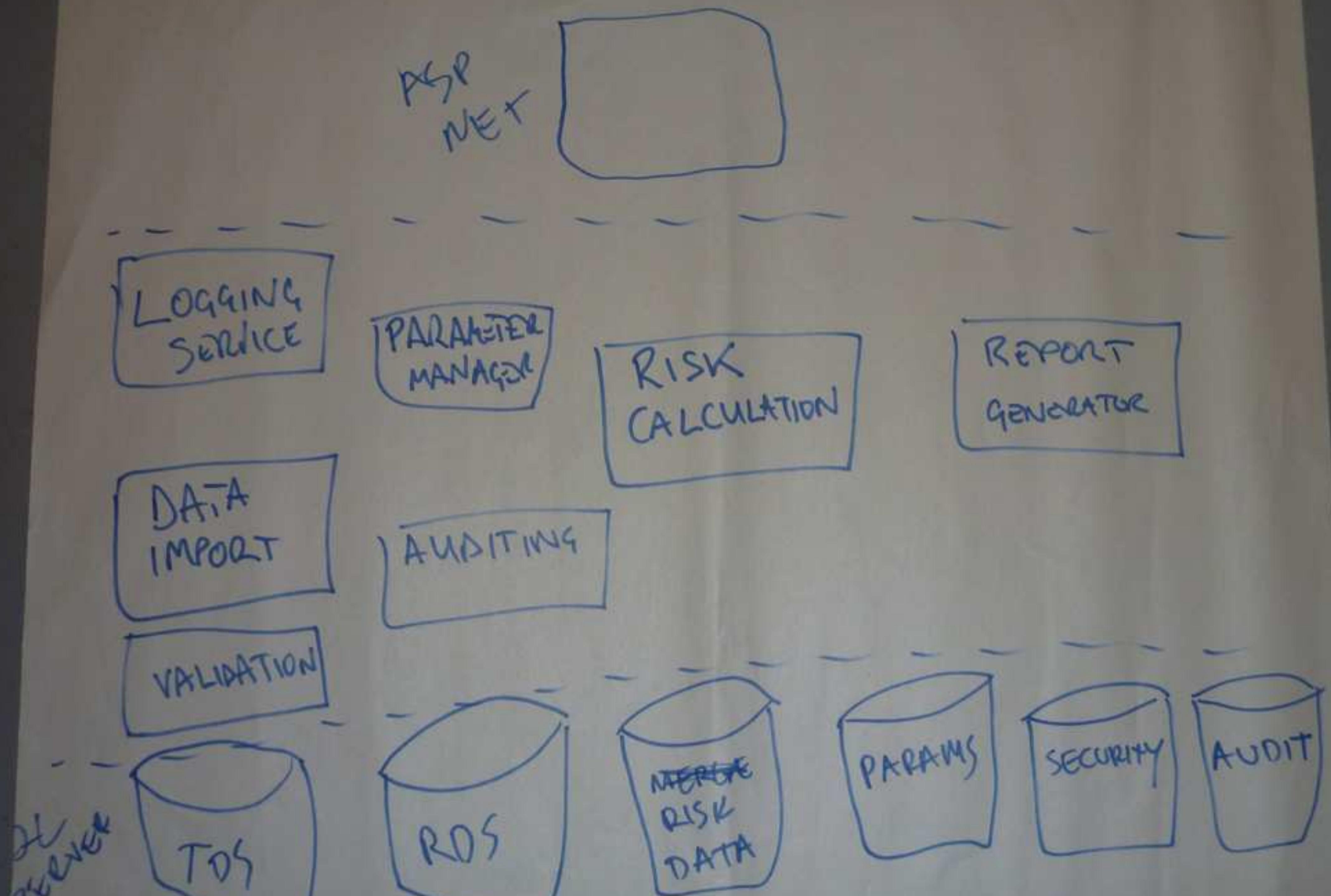


Just use a whiteboard!

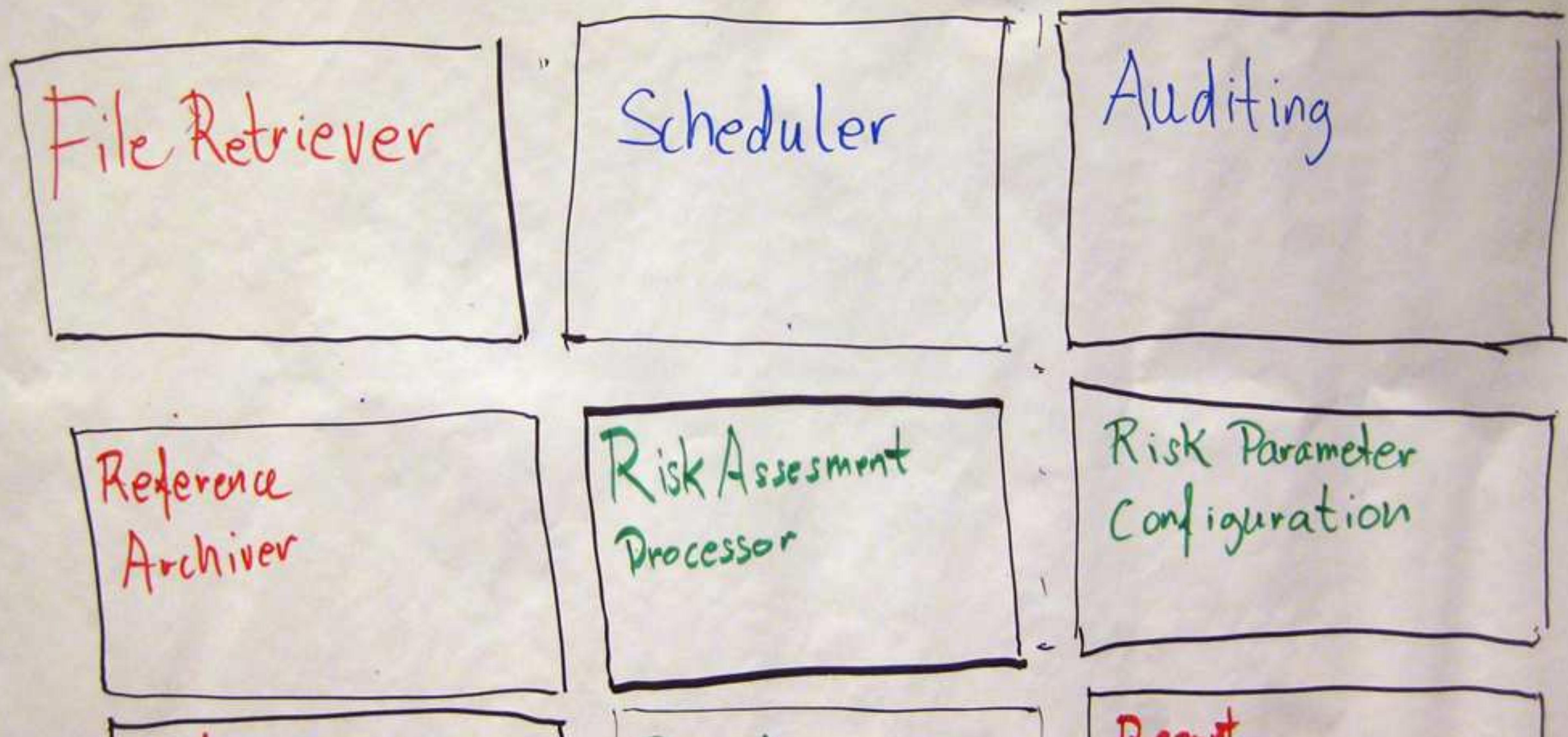


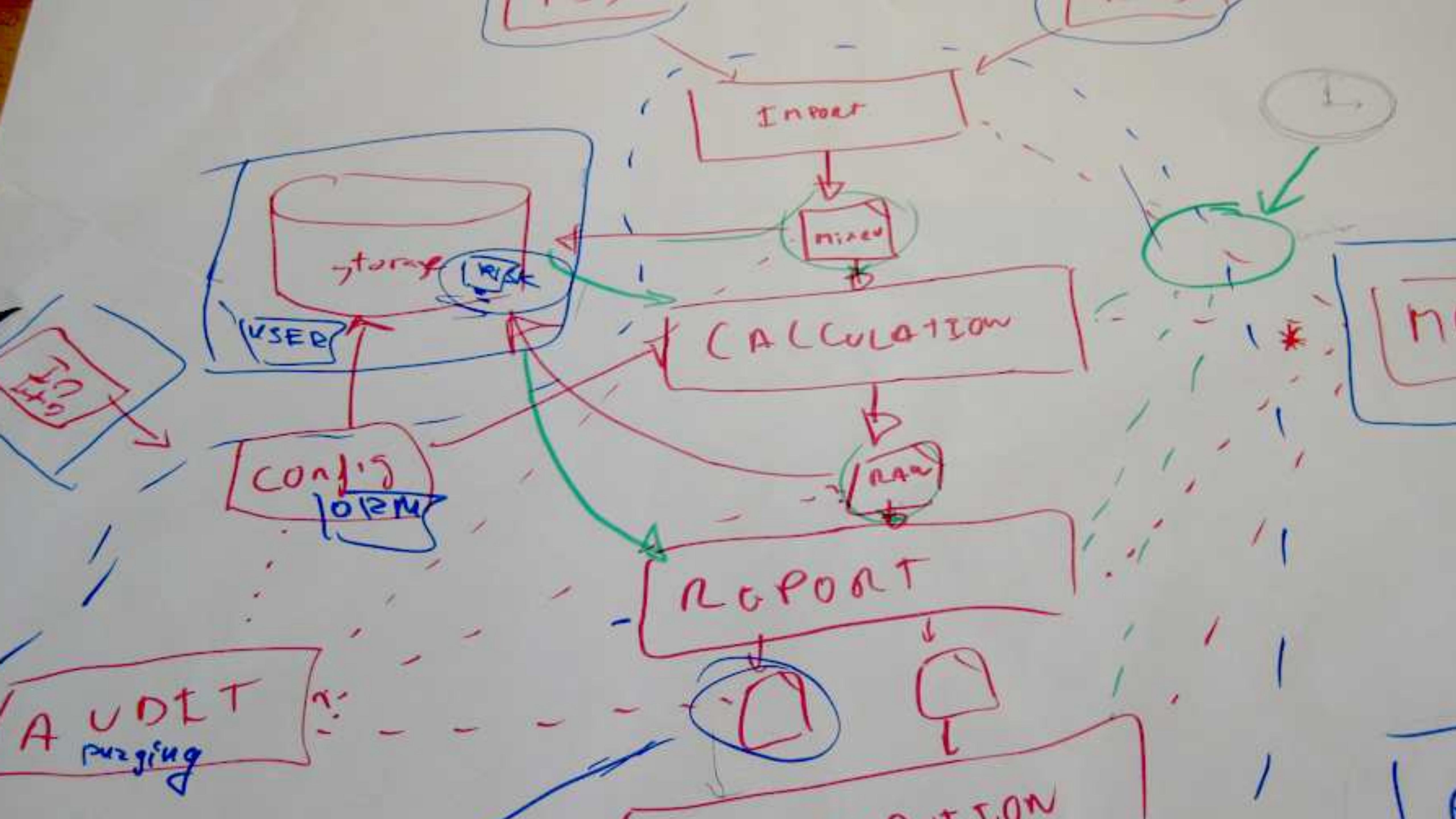
1:42 / 12:48

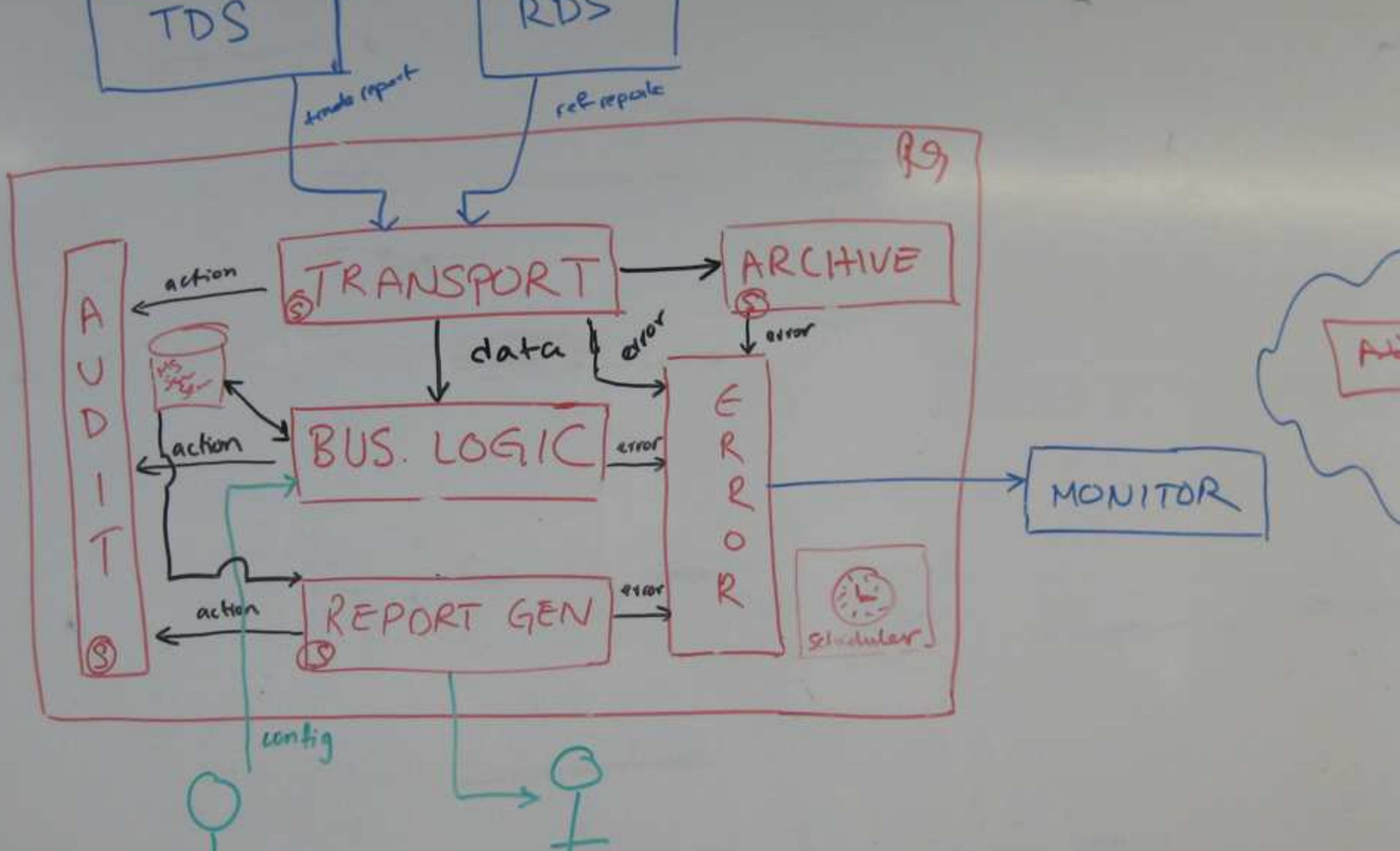


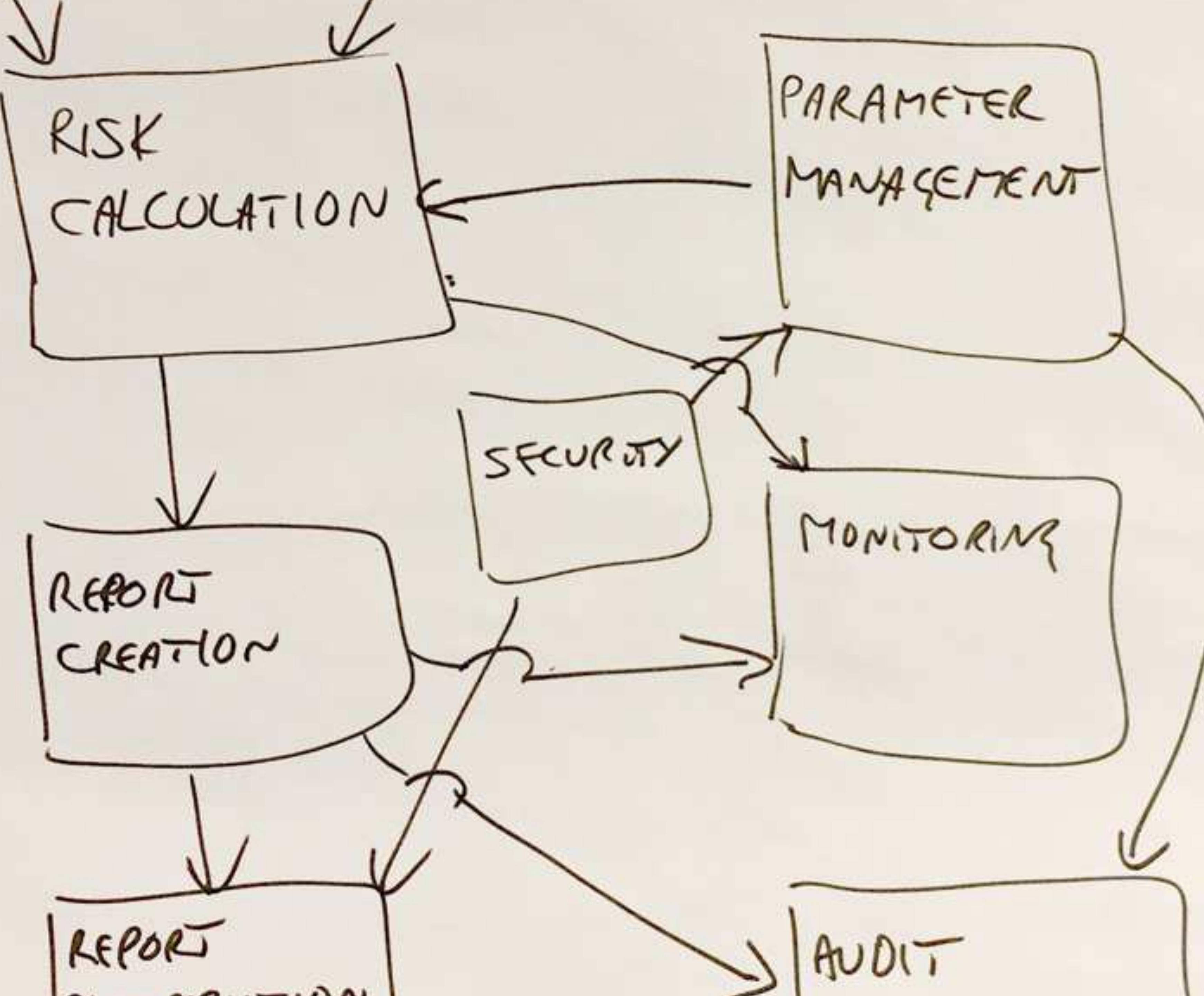


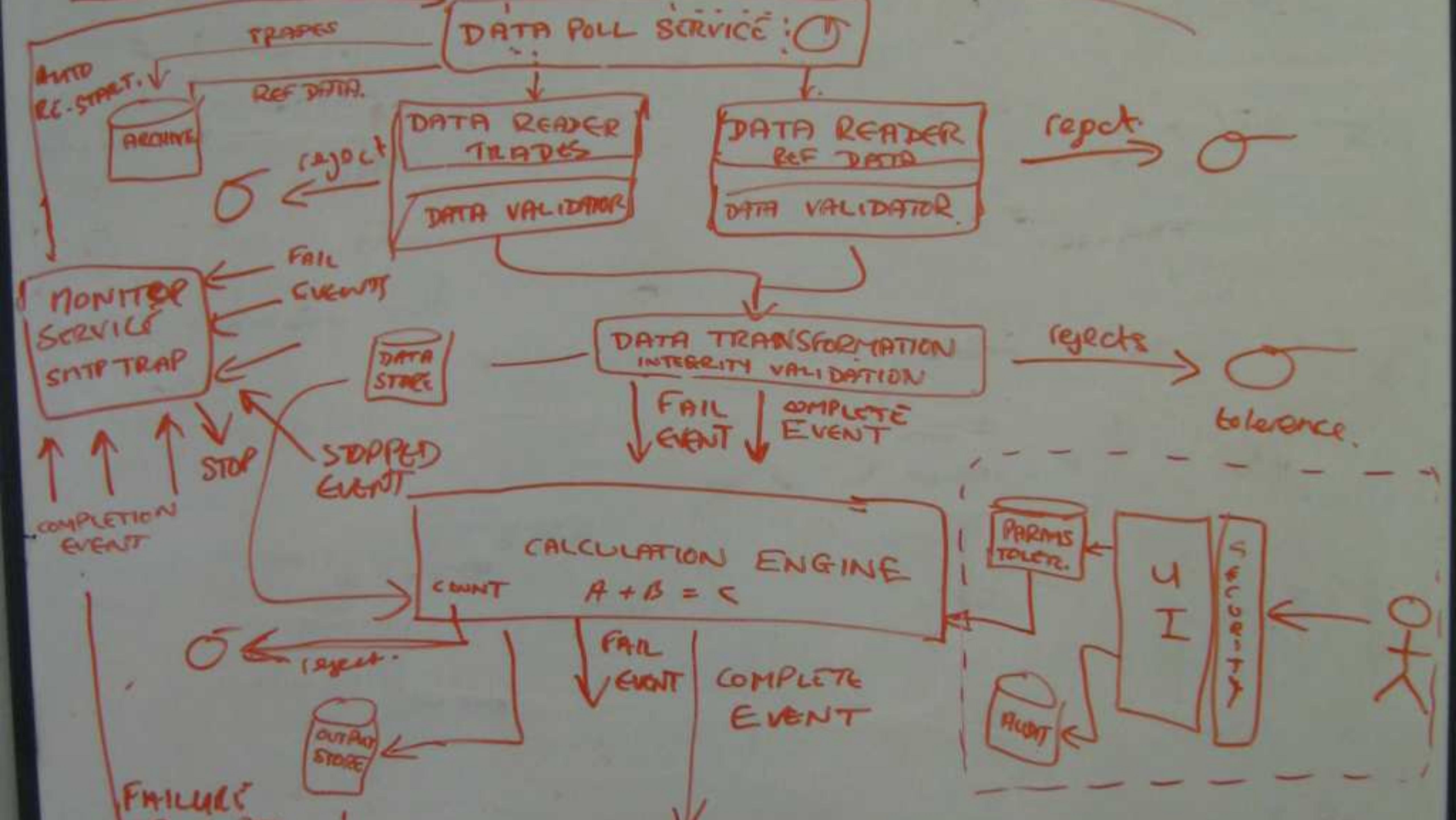
FUNCTIONAL VIEW

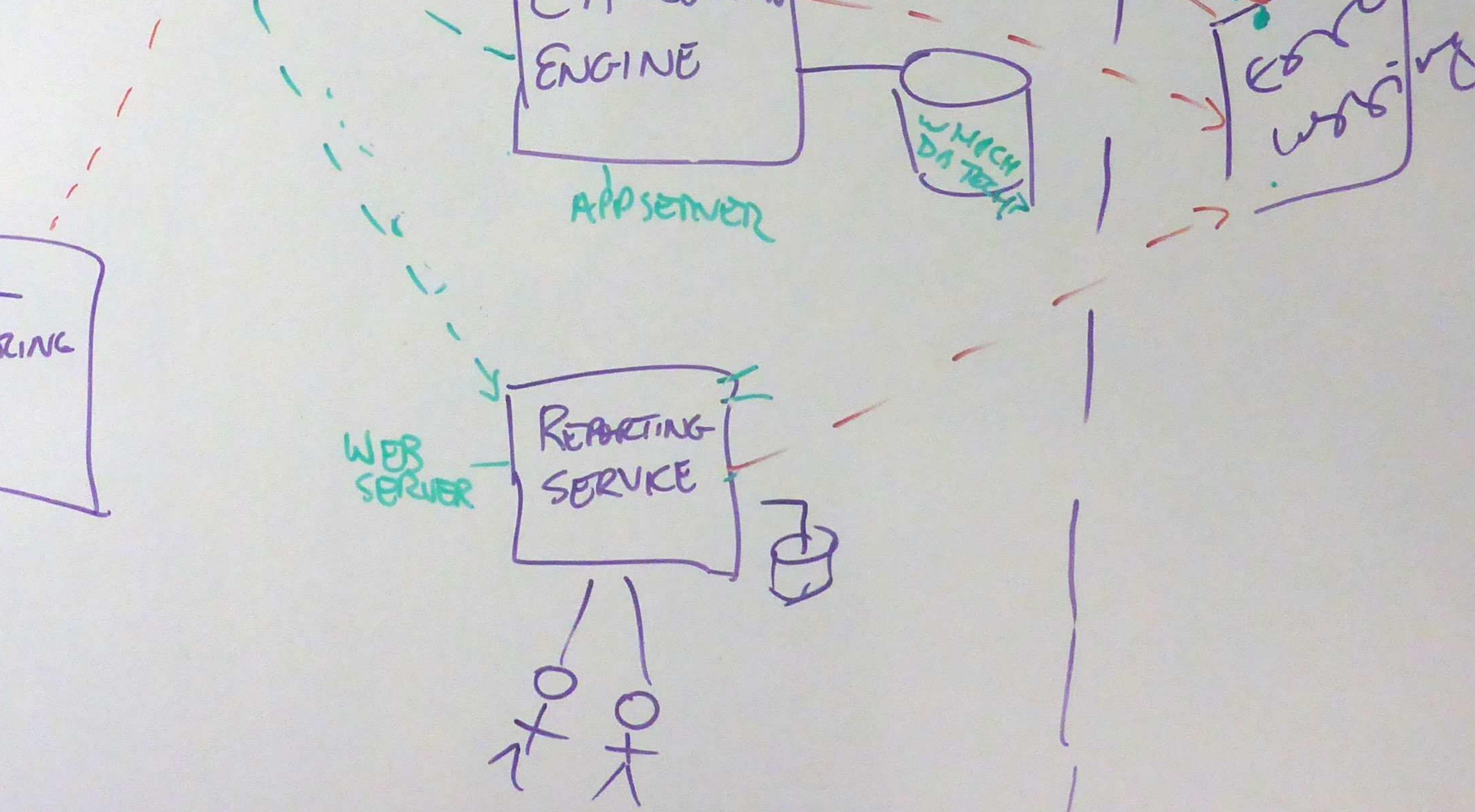






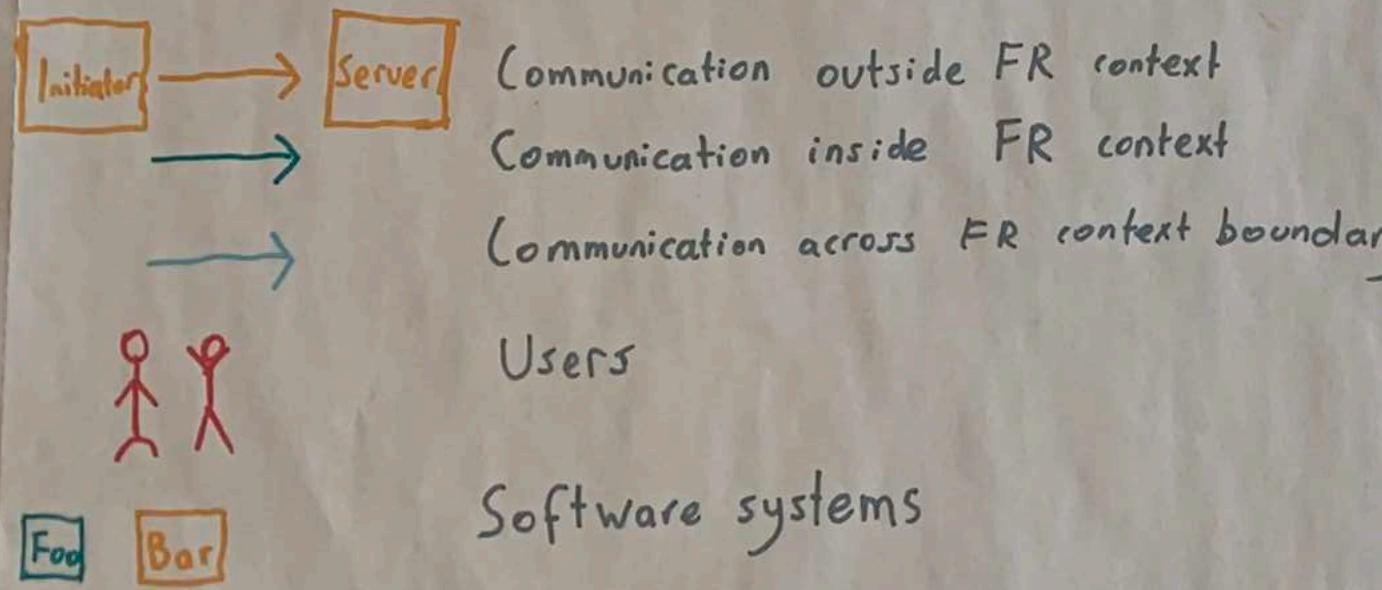
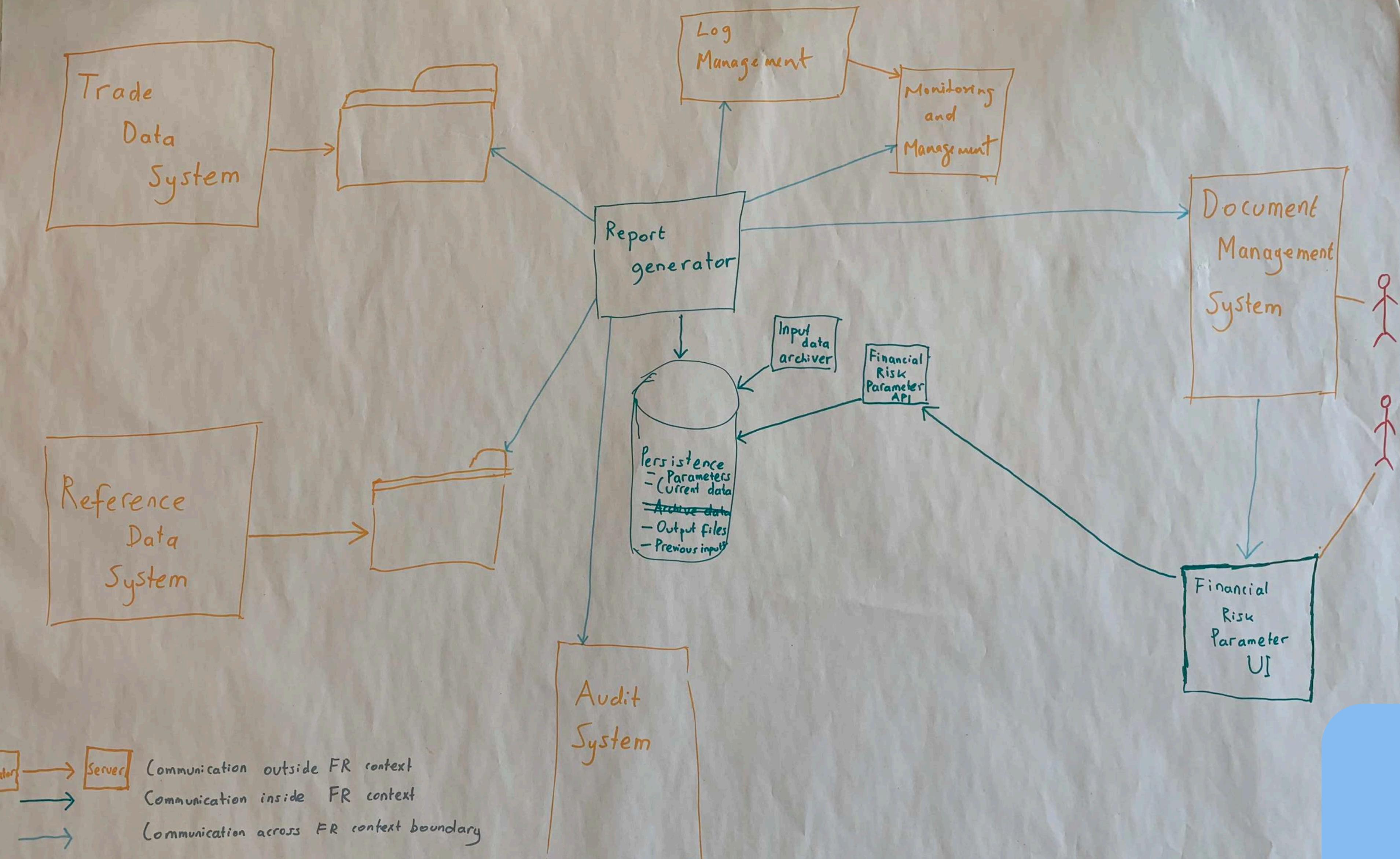


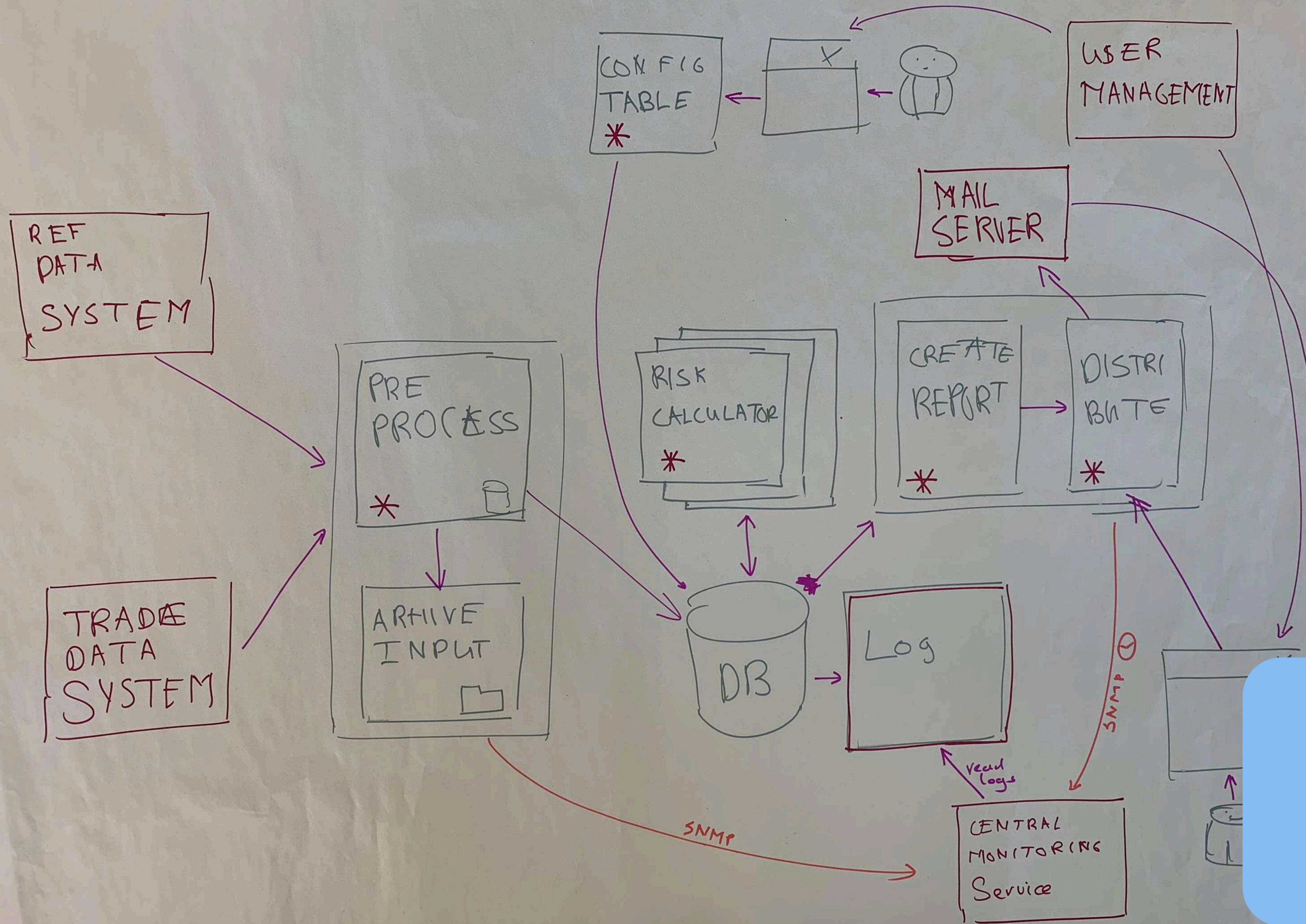


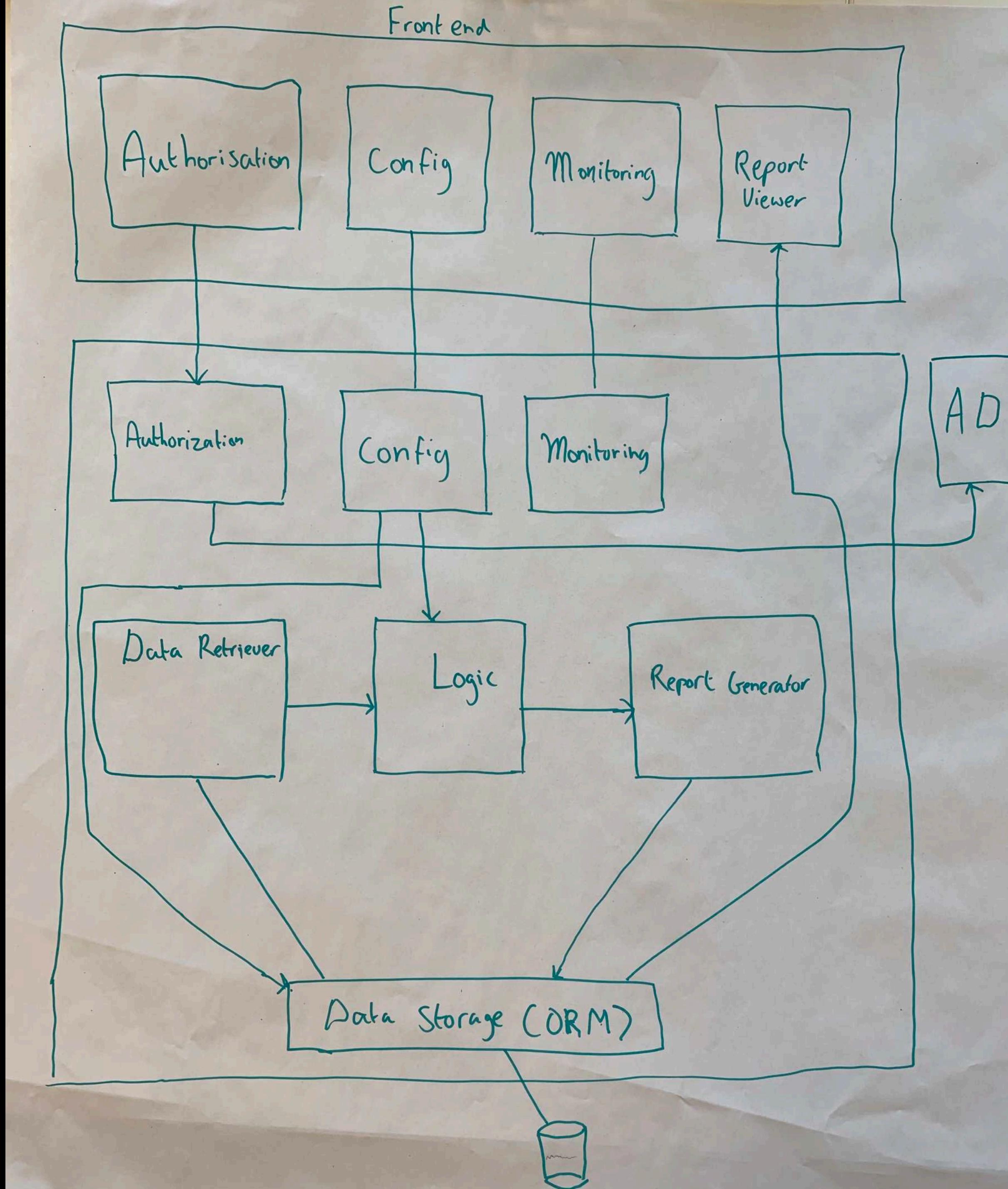






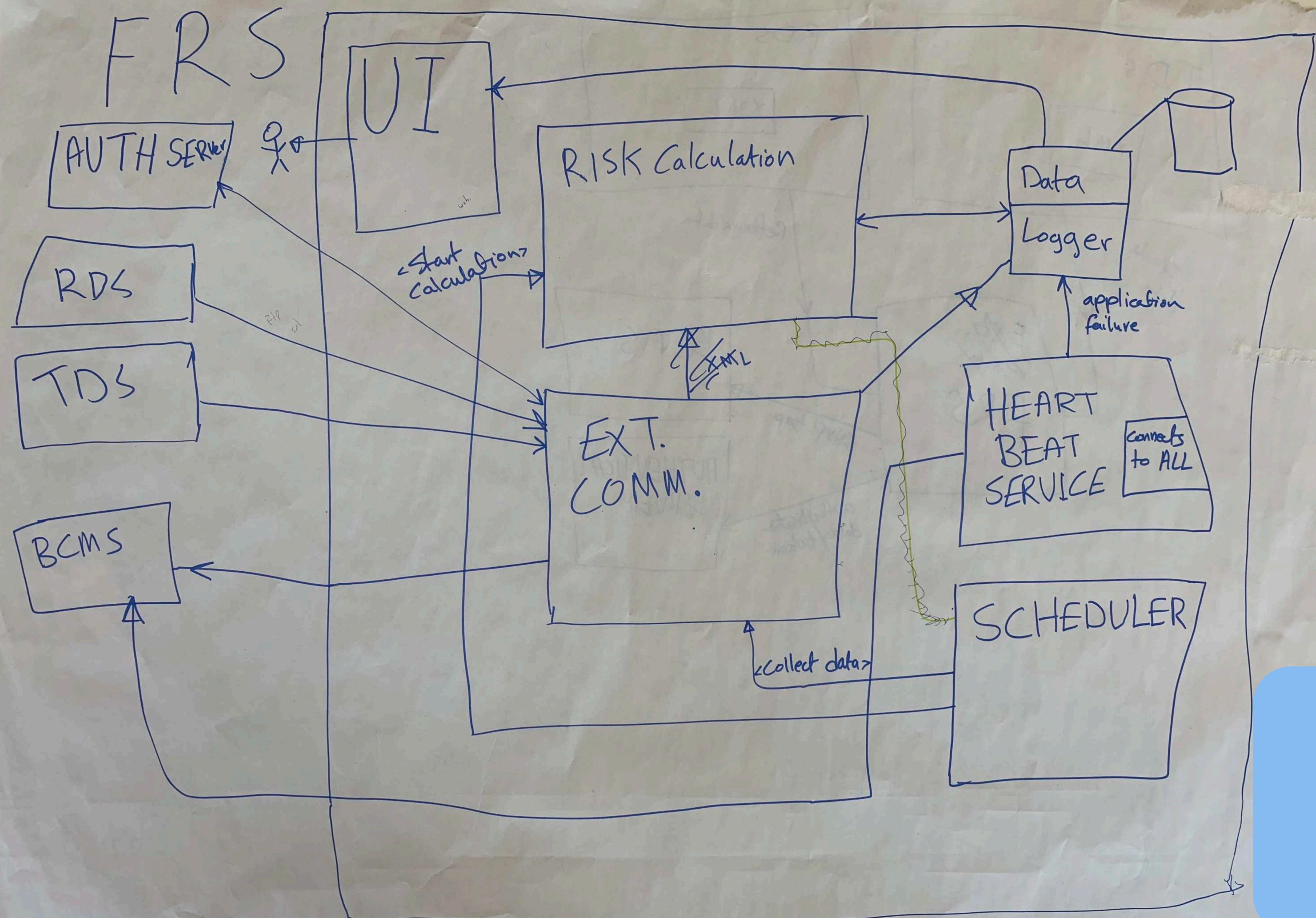


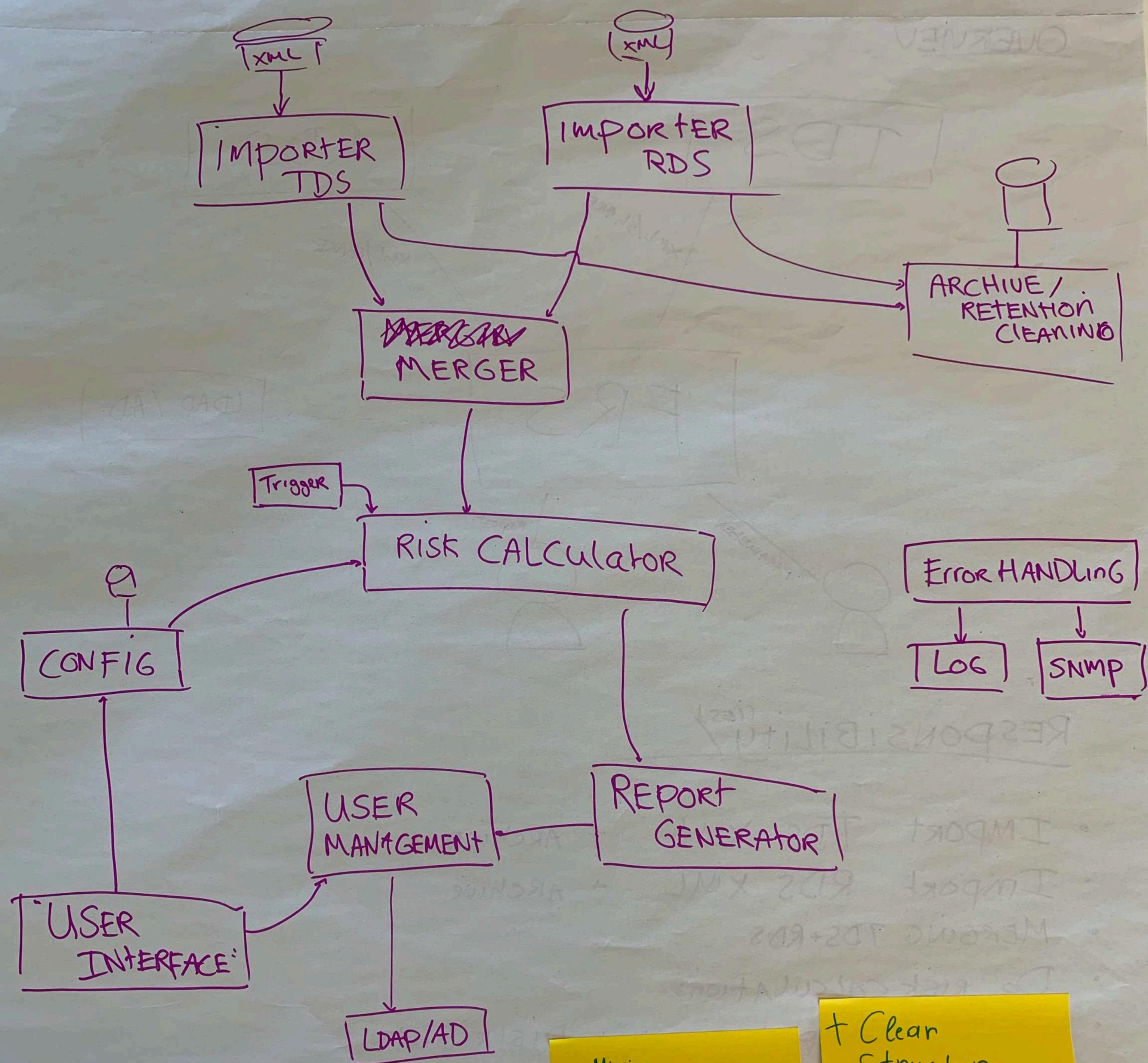


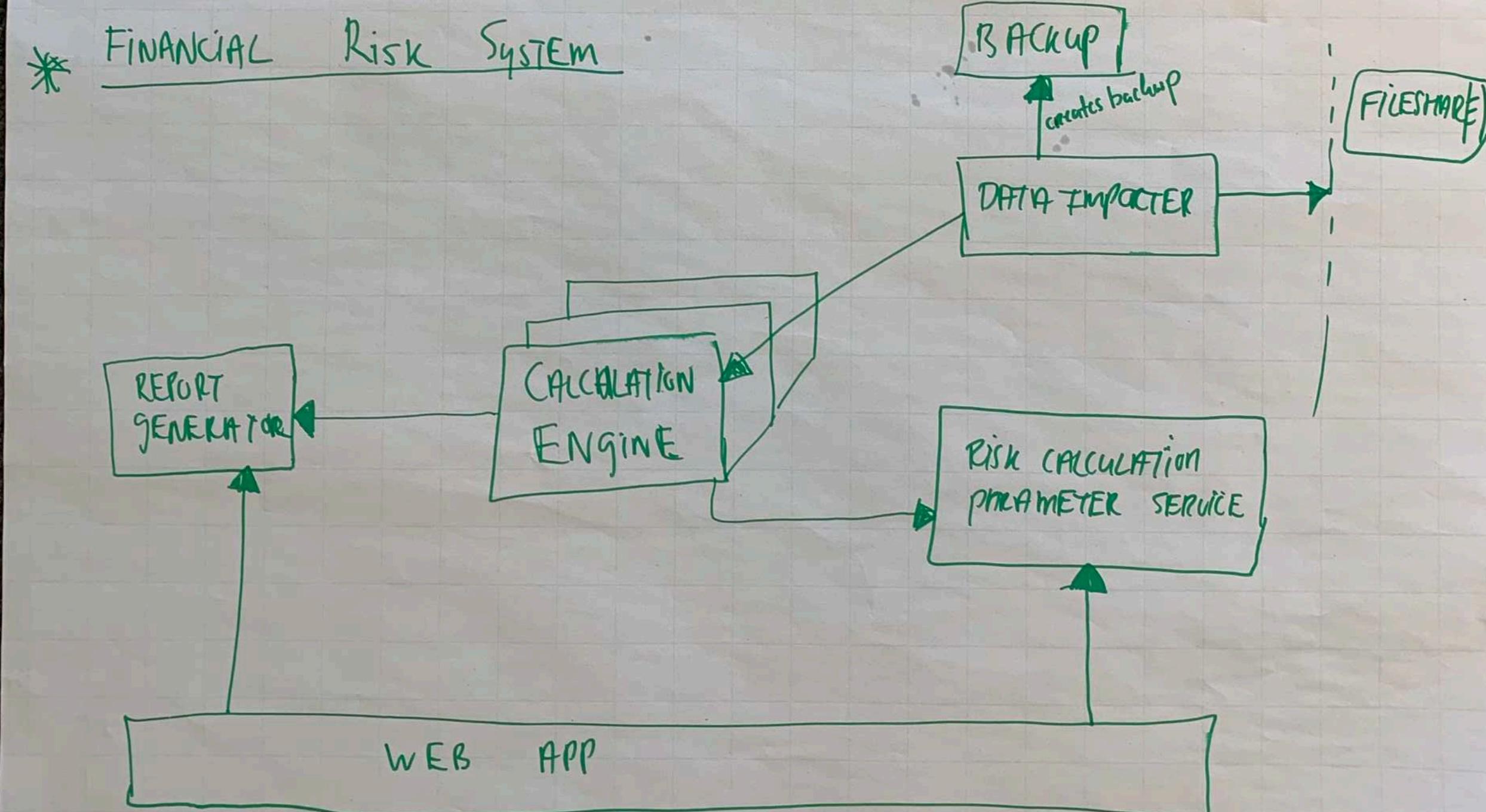
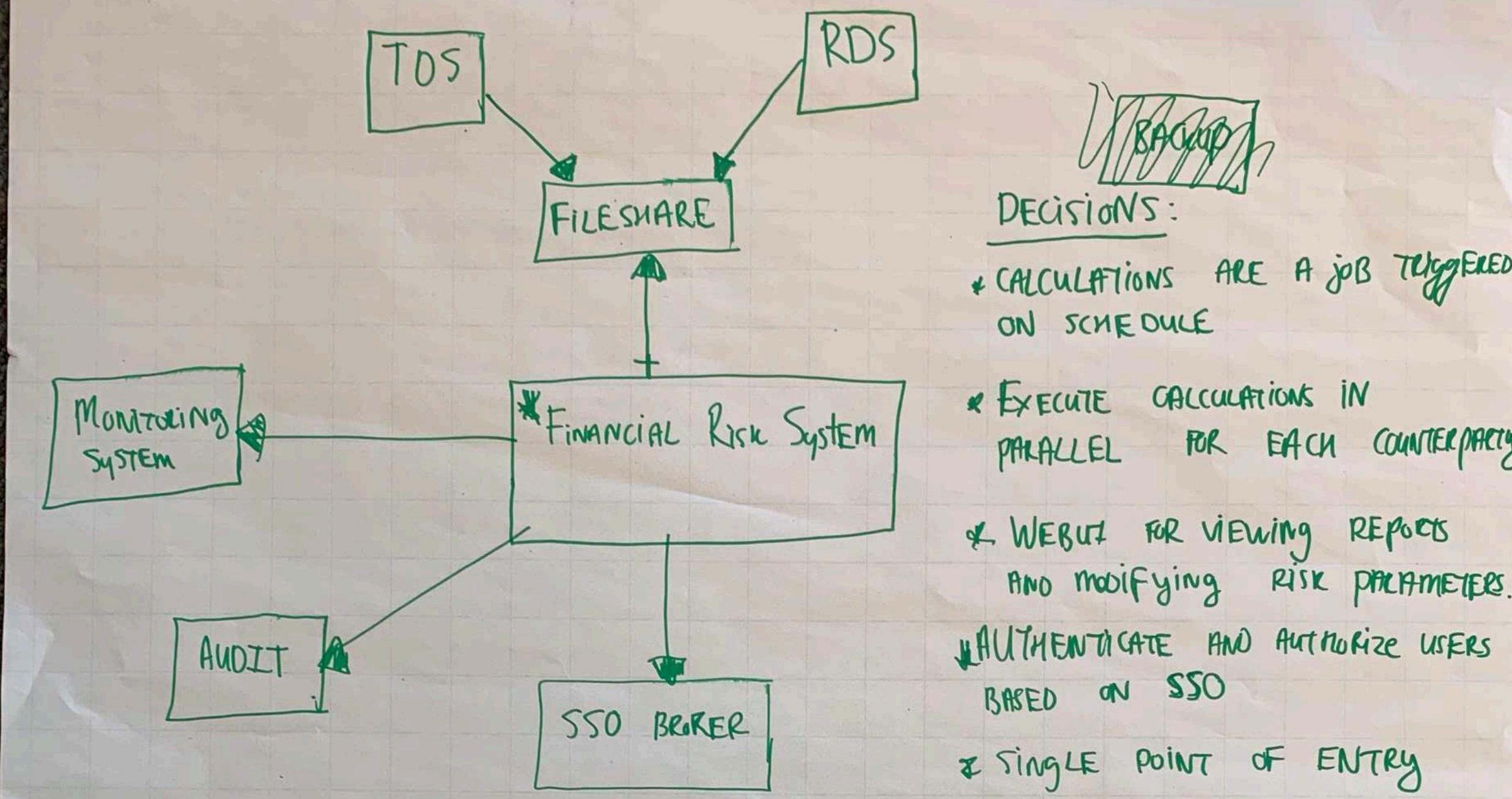


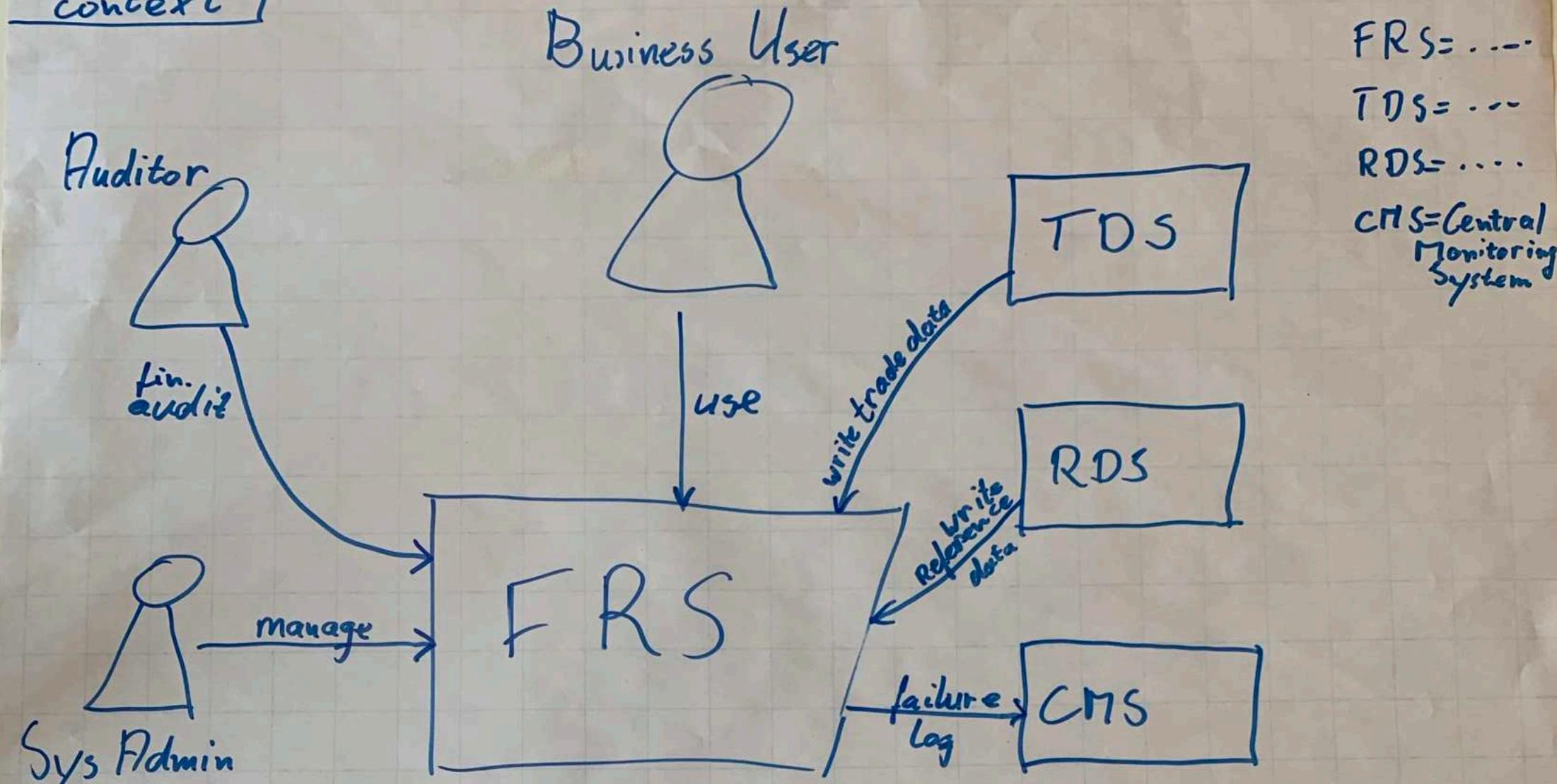
Significant decisions

- F/E < > B/E
- Make use of OS' watchdog mechanism
- Data storage ORM framework: Entity
- ASP .NET B/E
- Angular F/E

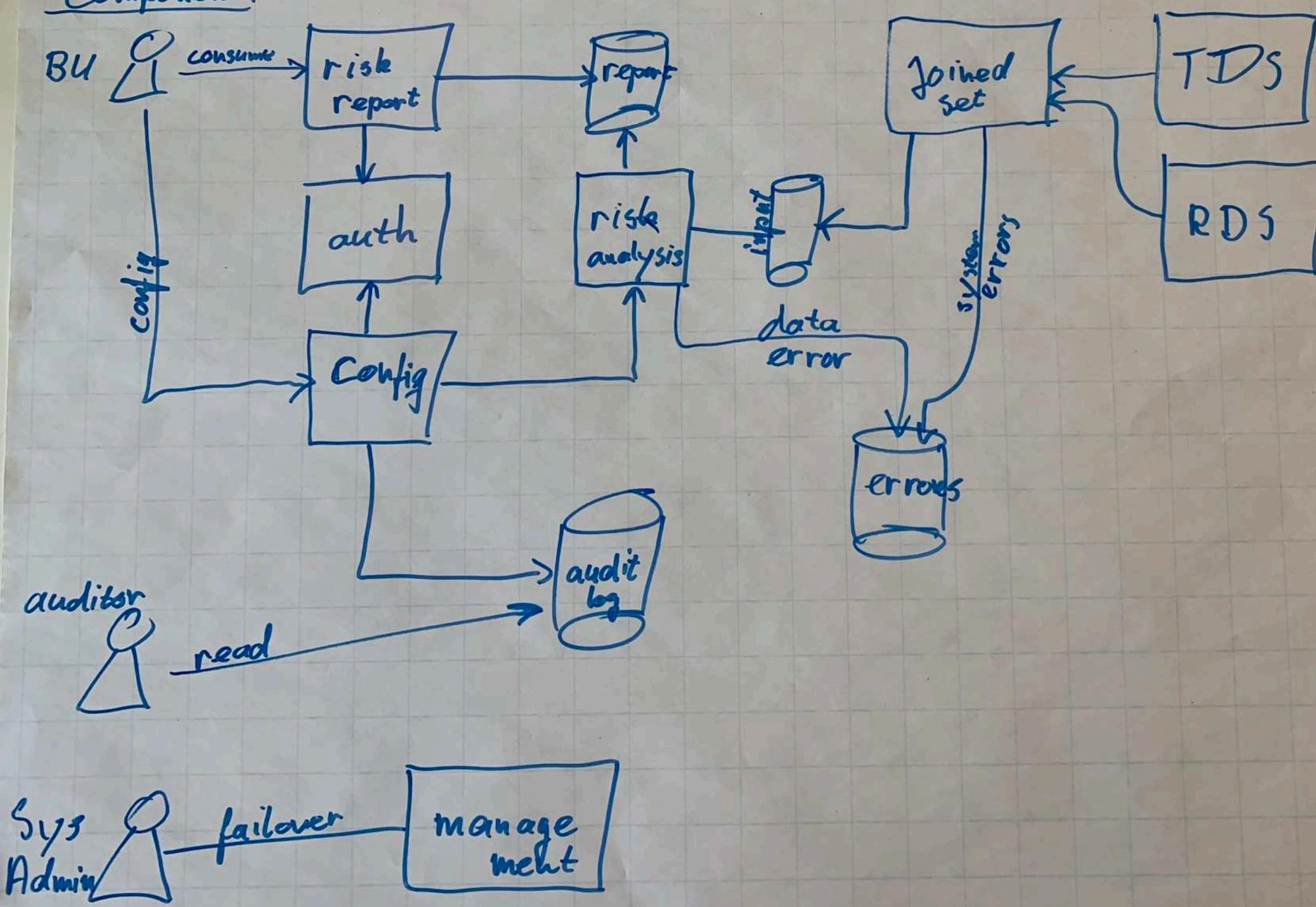




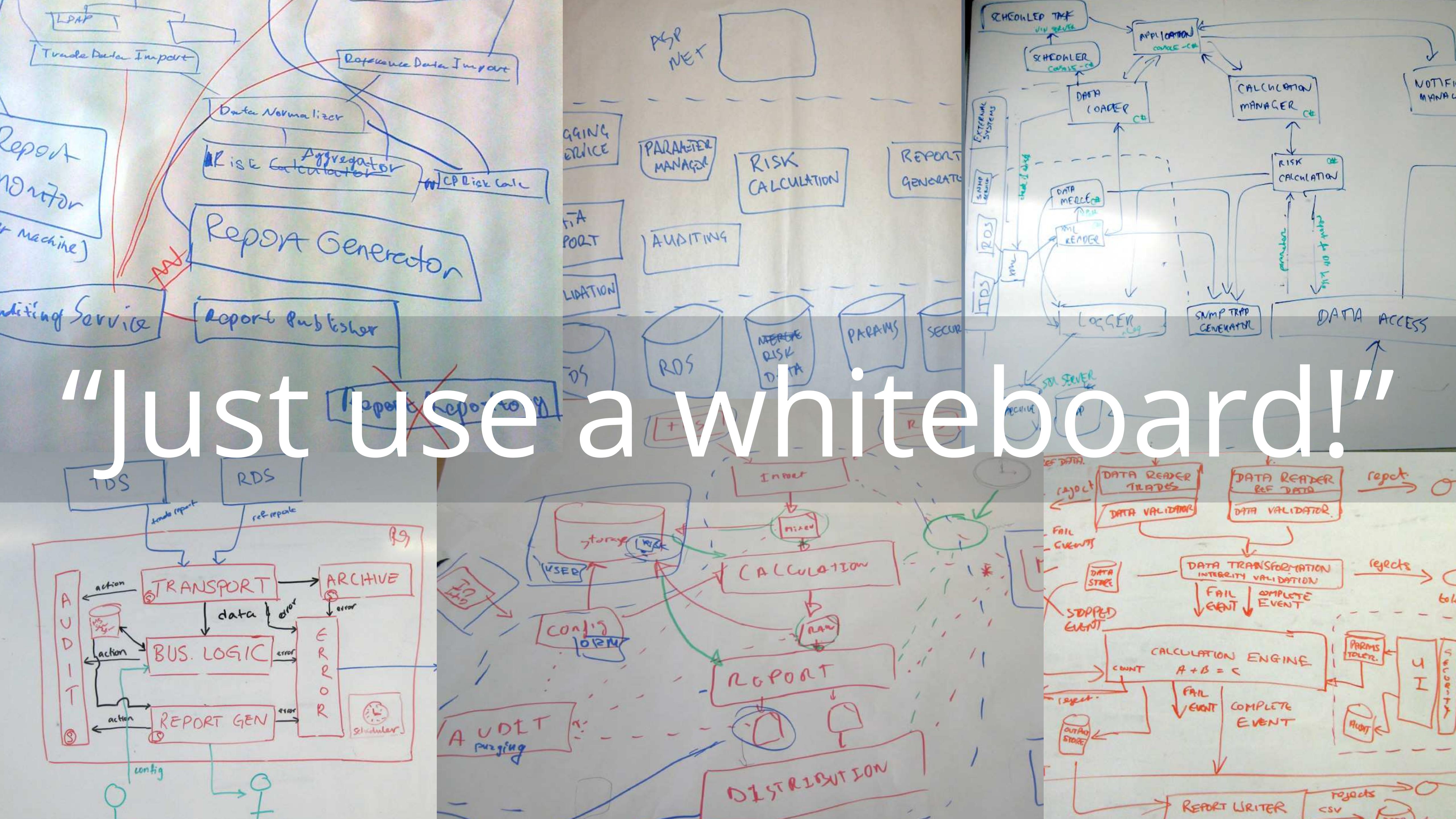


context /

FRS = ...
TDS = ...
RDS = ...
CMS = Central Monitoring System

Component /

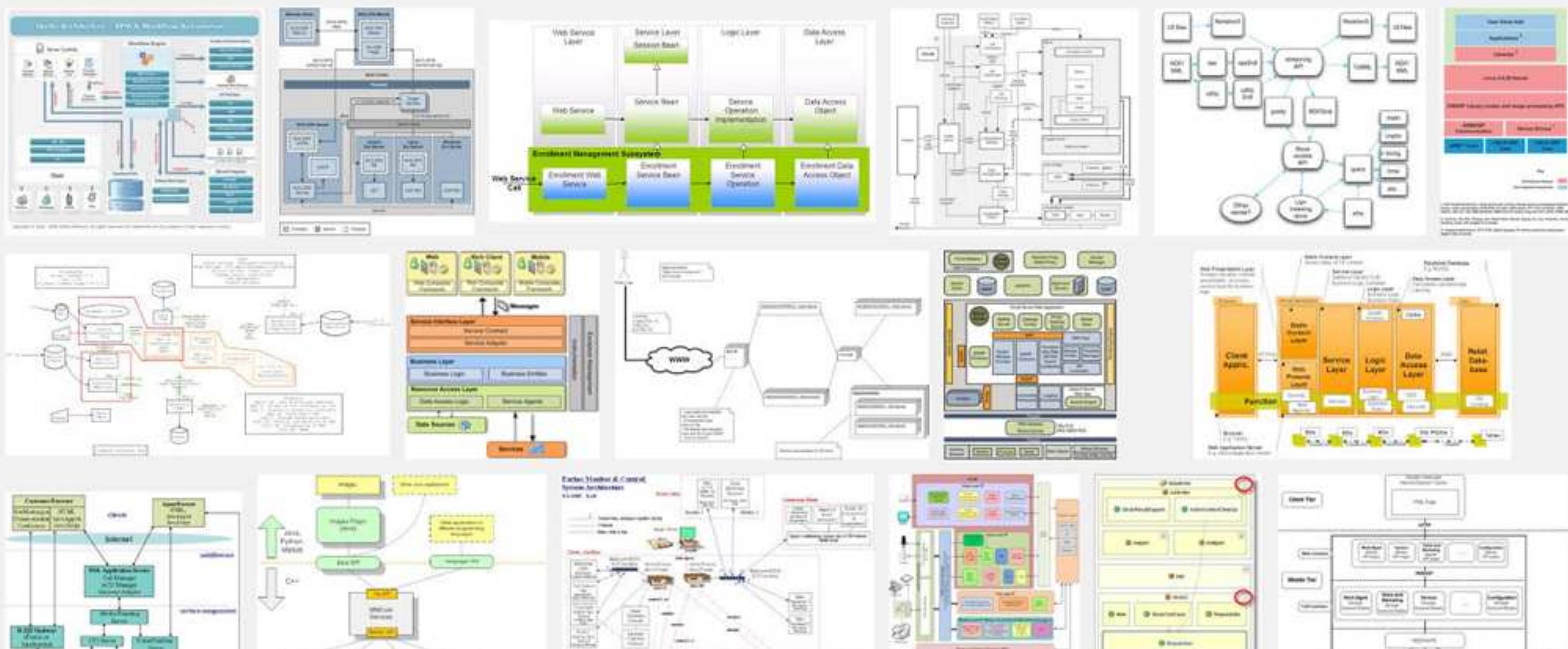
“Just use a whiteboard!”

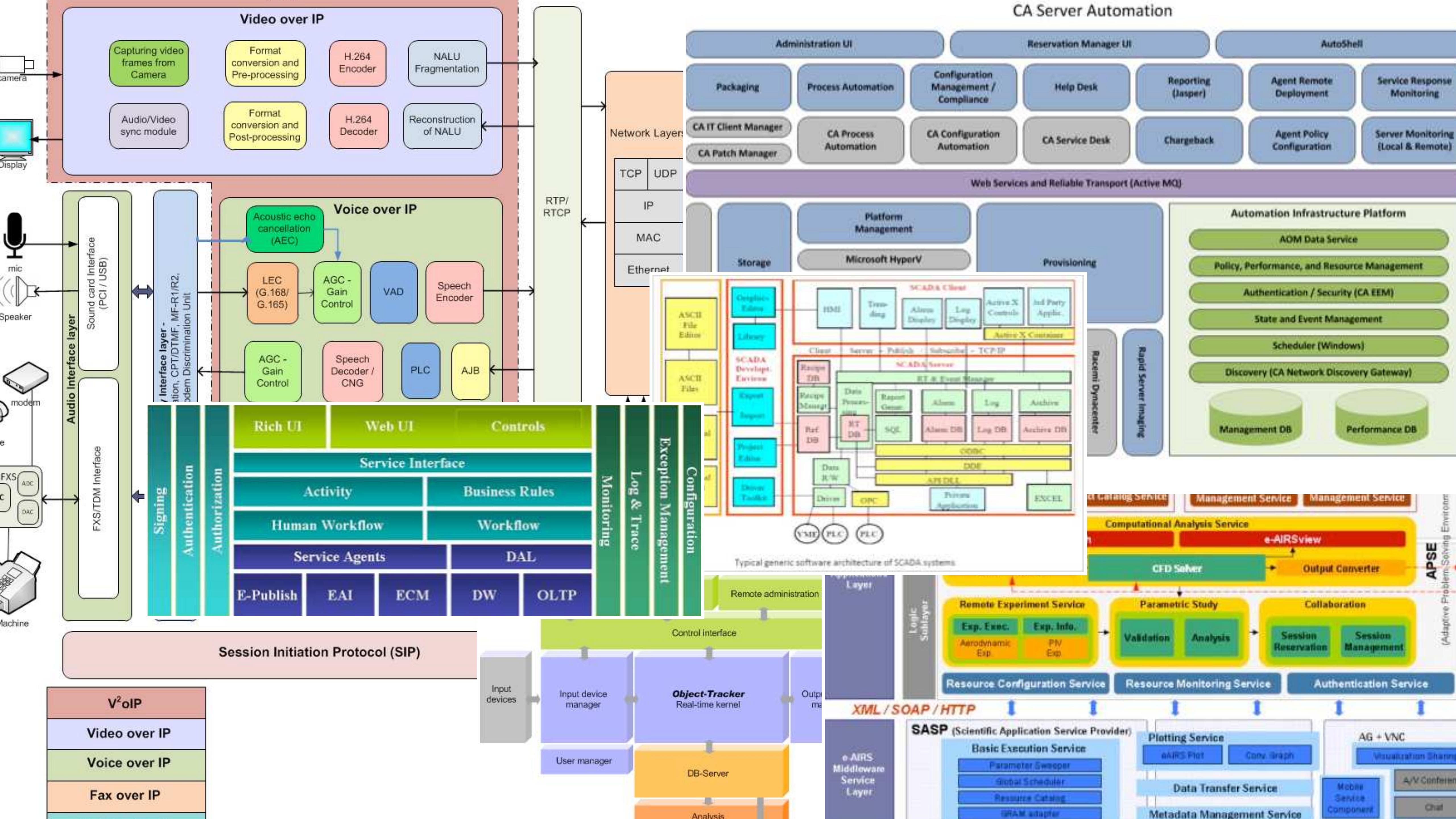


Cookies help us deliver our services. By using our services, you agree to our use of cookies.

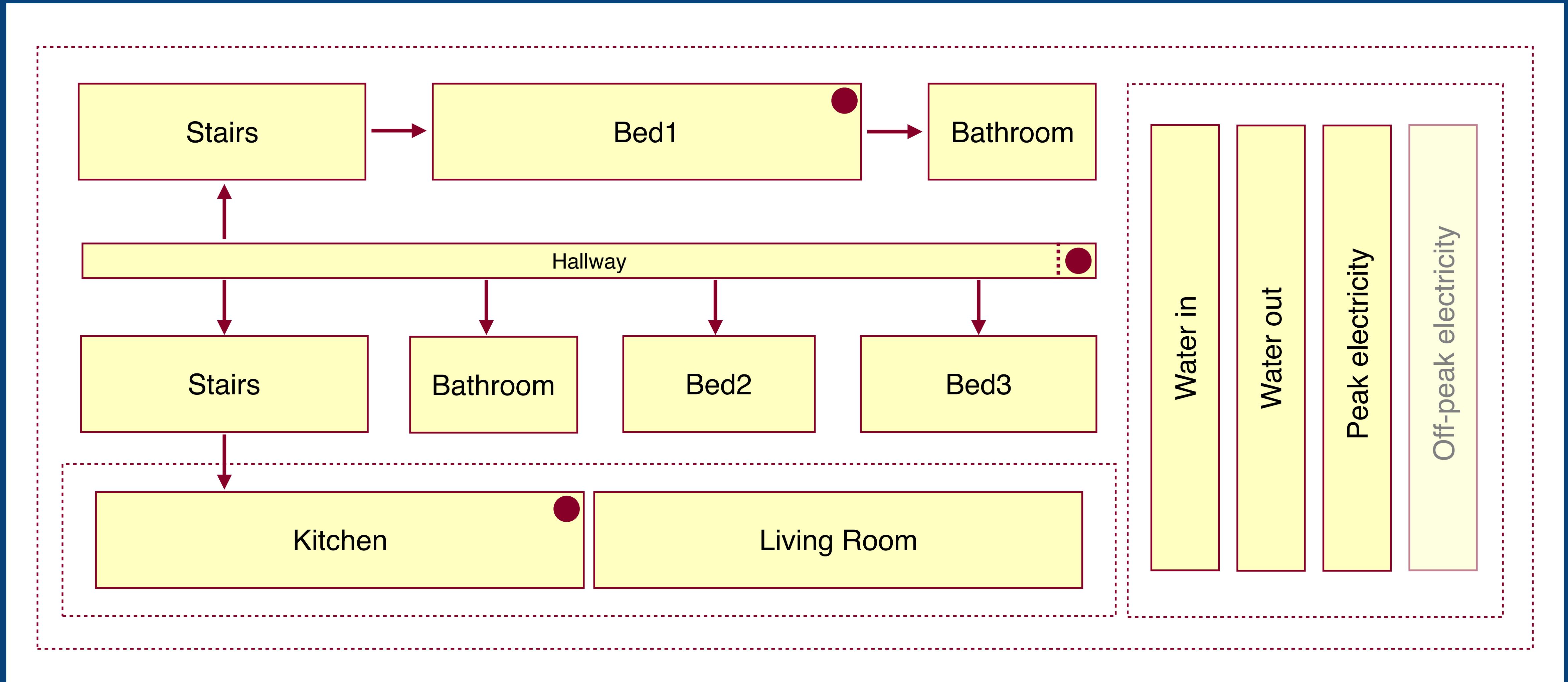
Learn more

Get It





If software developers created building architecture diagrams...



Engineers,
not artists

Moving fast in the same direction
as a team requires
good communication



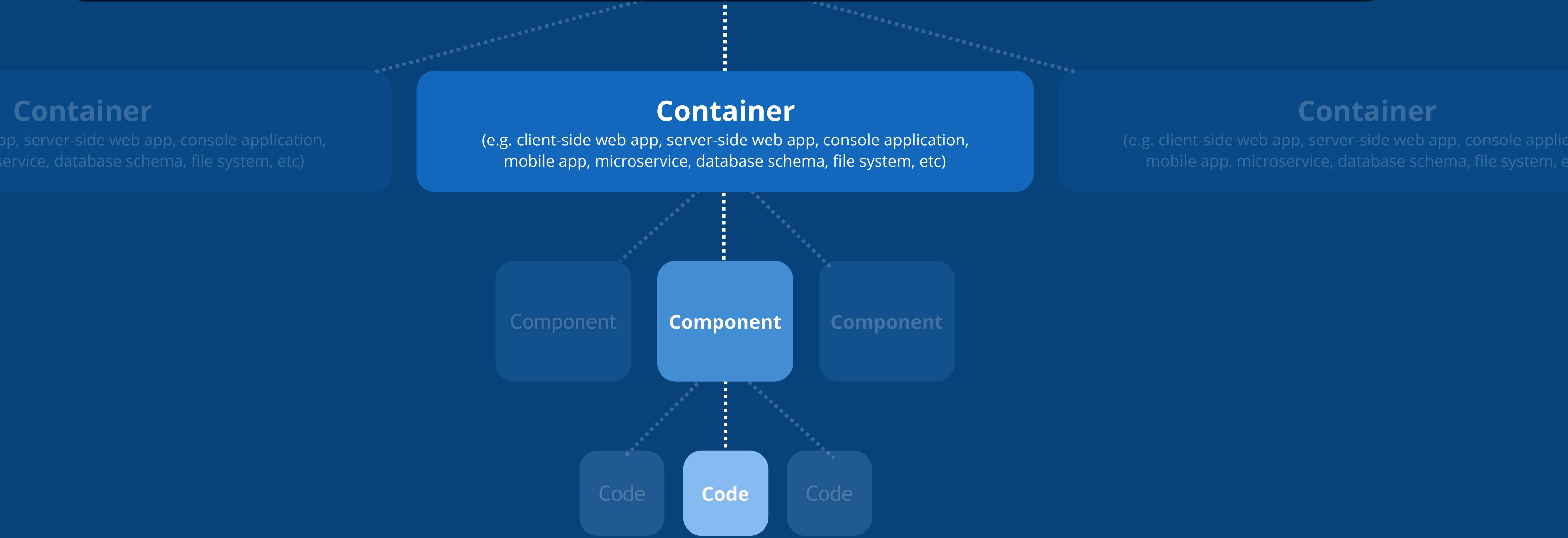
There are many **different audiences** for diagrams
and documentation, all with **different interests**

(software architects, software developers, operations and support staff, testers,
Product Owners, project managers, Scrum Masters, users, management,
business sponsors, potential customers, potential investors, ...)

When drawing software
architecture diagrams,
think like a software developer

A common set of abstractions
is more important
than a common notation

Software System



A **software system** is made up of one or more **containers**,
each of which contains one or more **components**,
which in turn are implemented by one or more **code elements**.

C4

c4model.com

1. System Context

The system plus users and system dependencies.

2. Containers

The overall shape of the architecture and technology choices.

3. Components

Logical components and their interactions within a container.

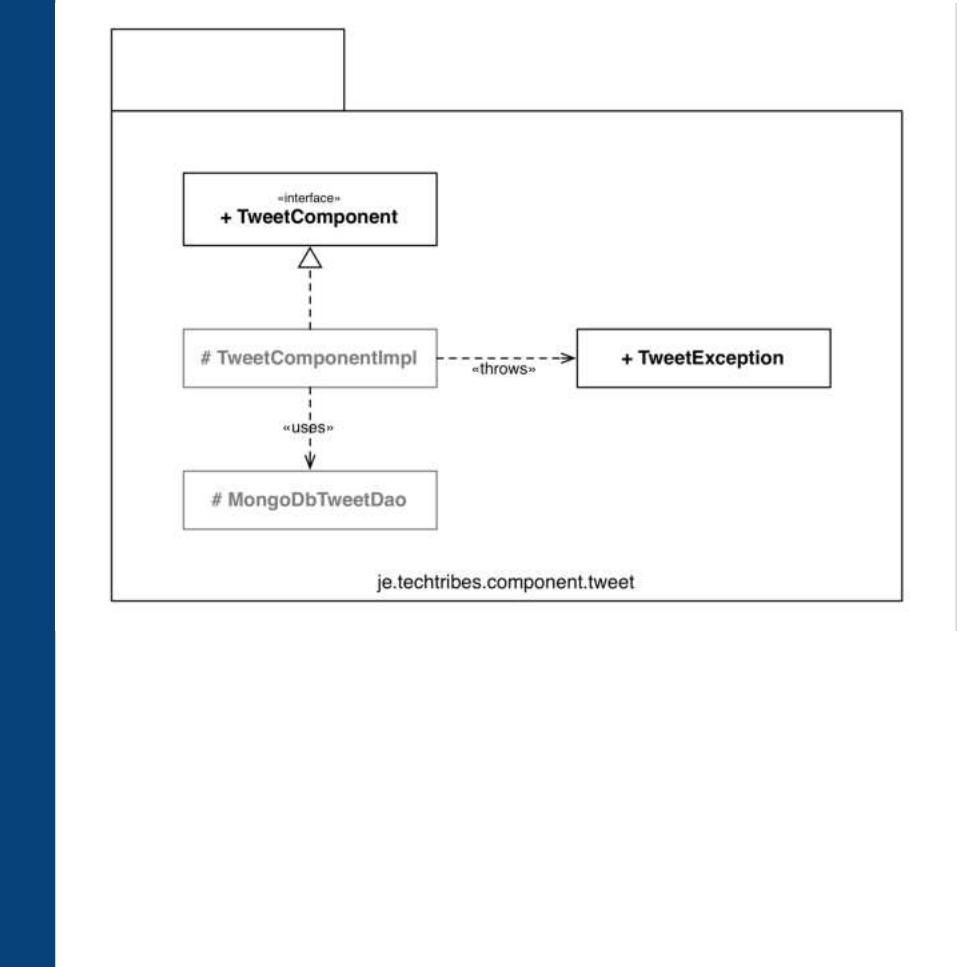
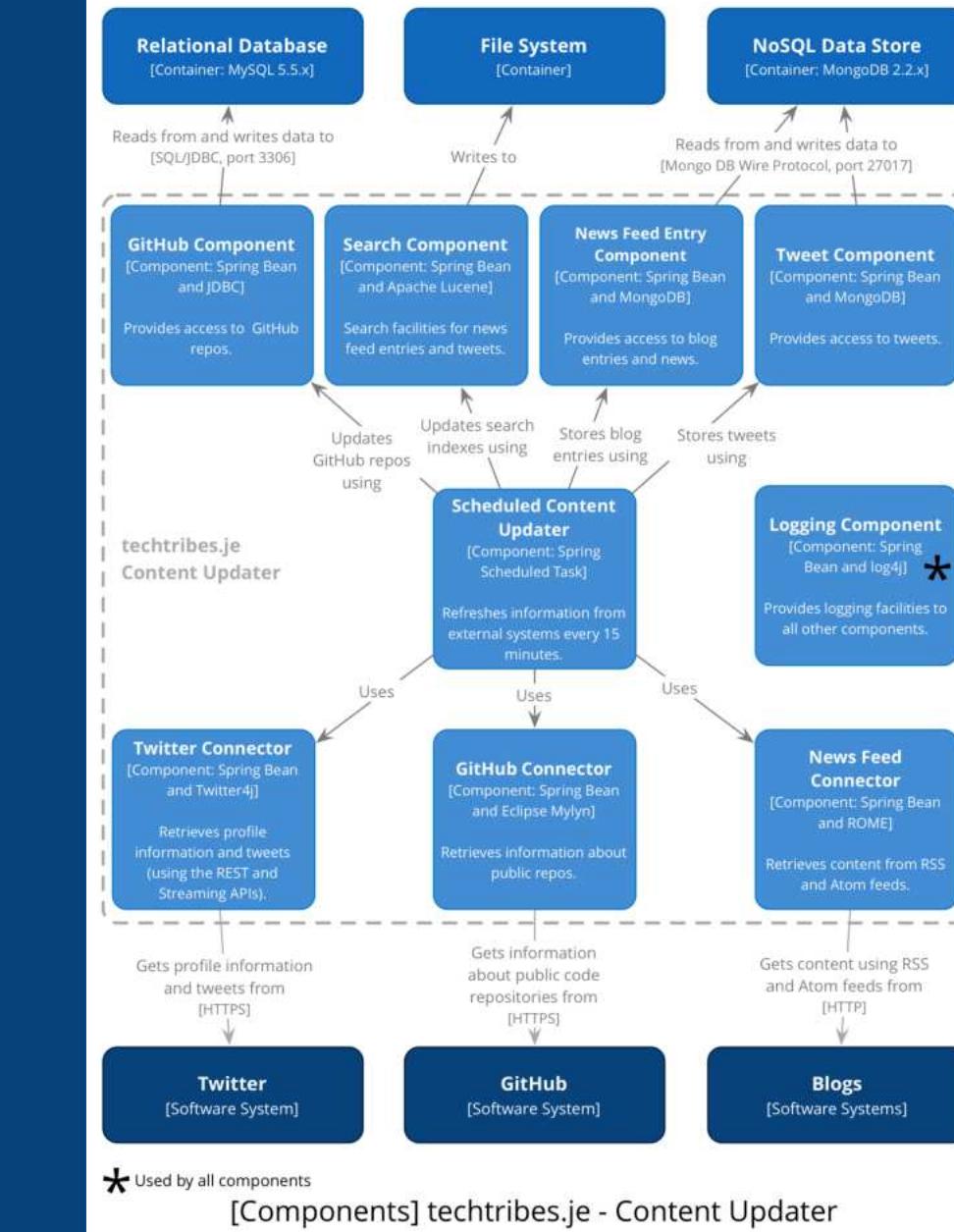
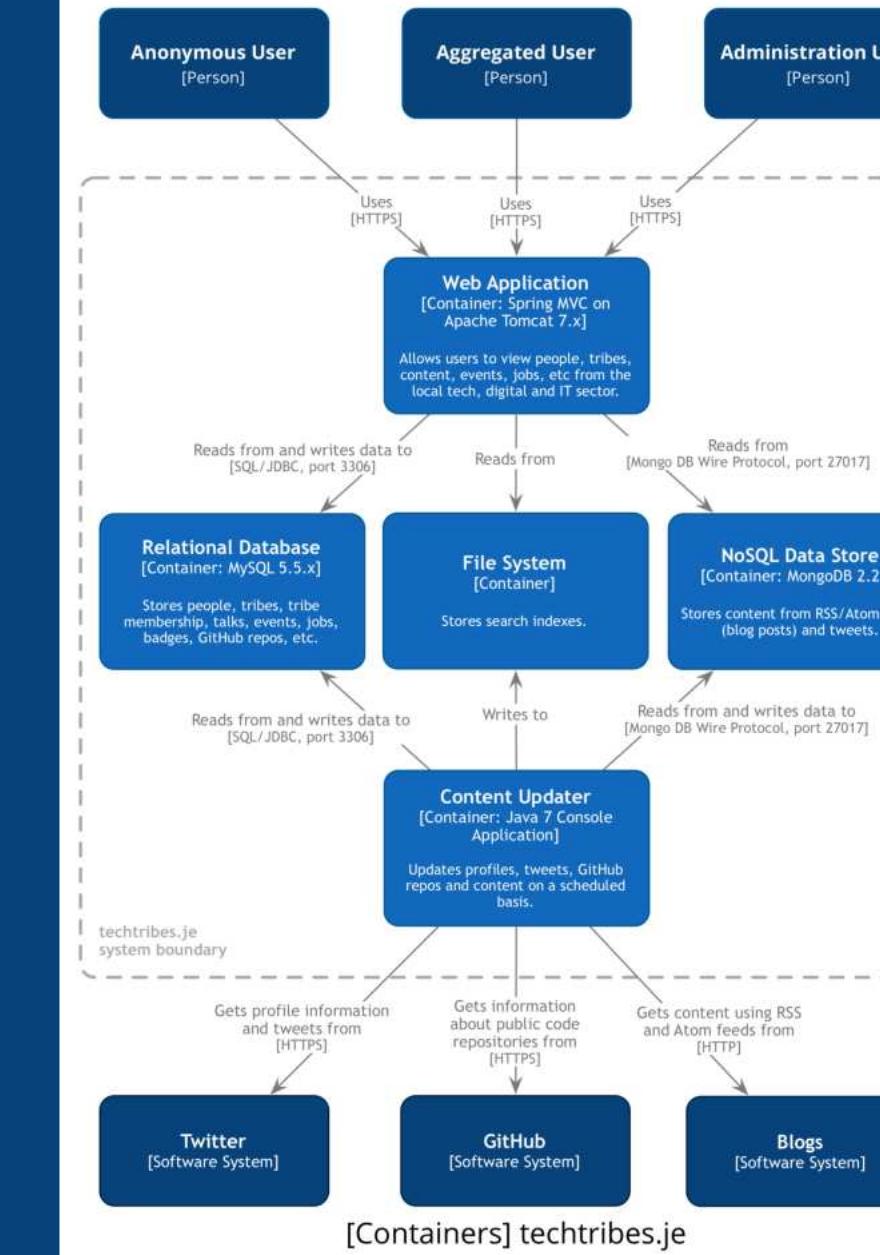
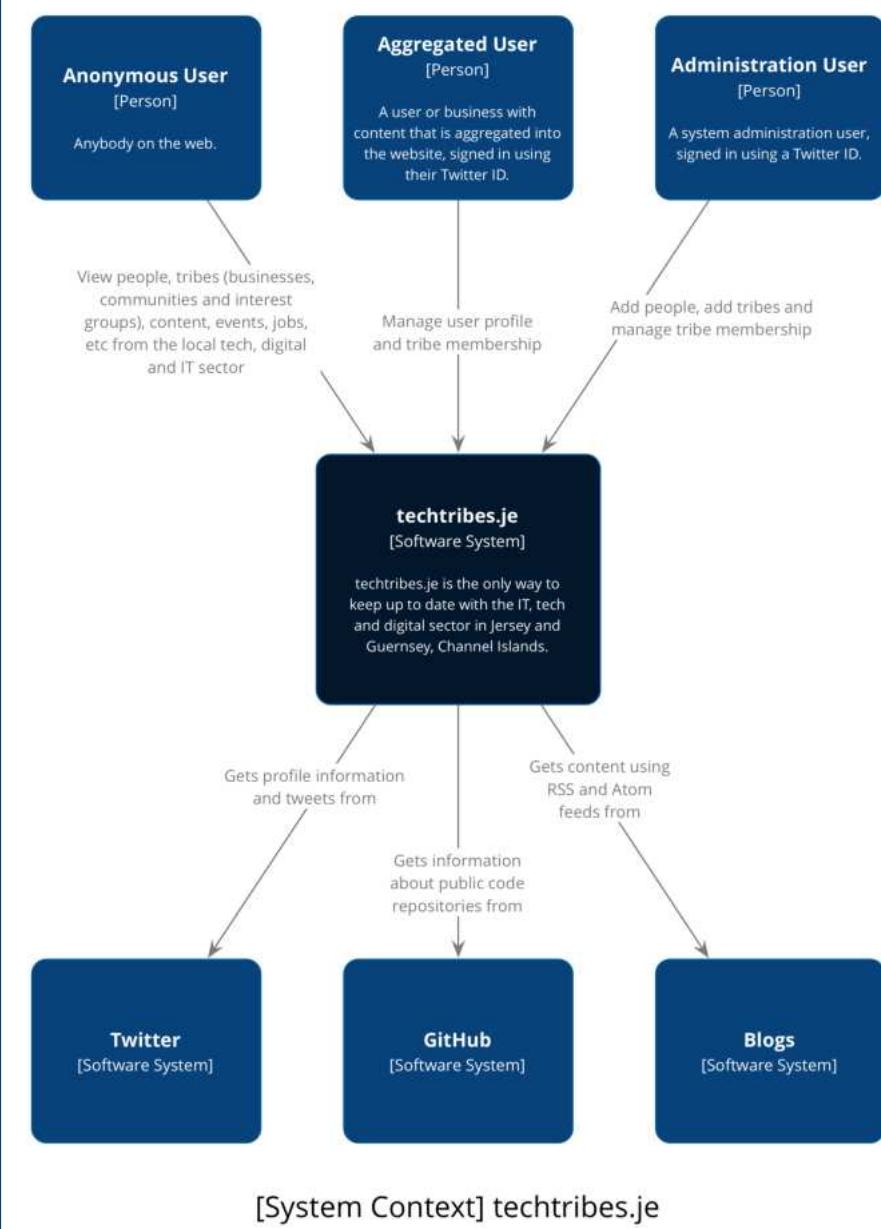
4. Code (e.g. classes)

Component implementation details.

Overview first

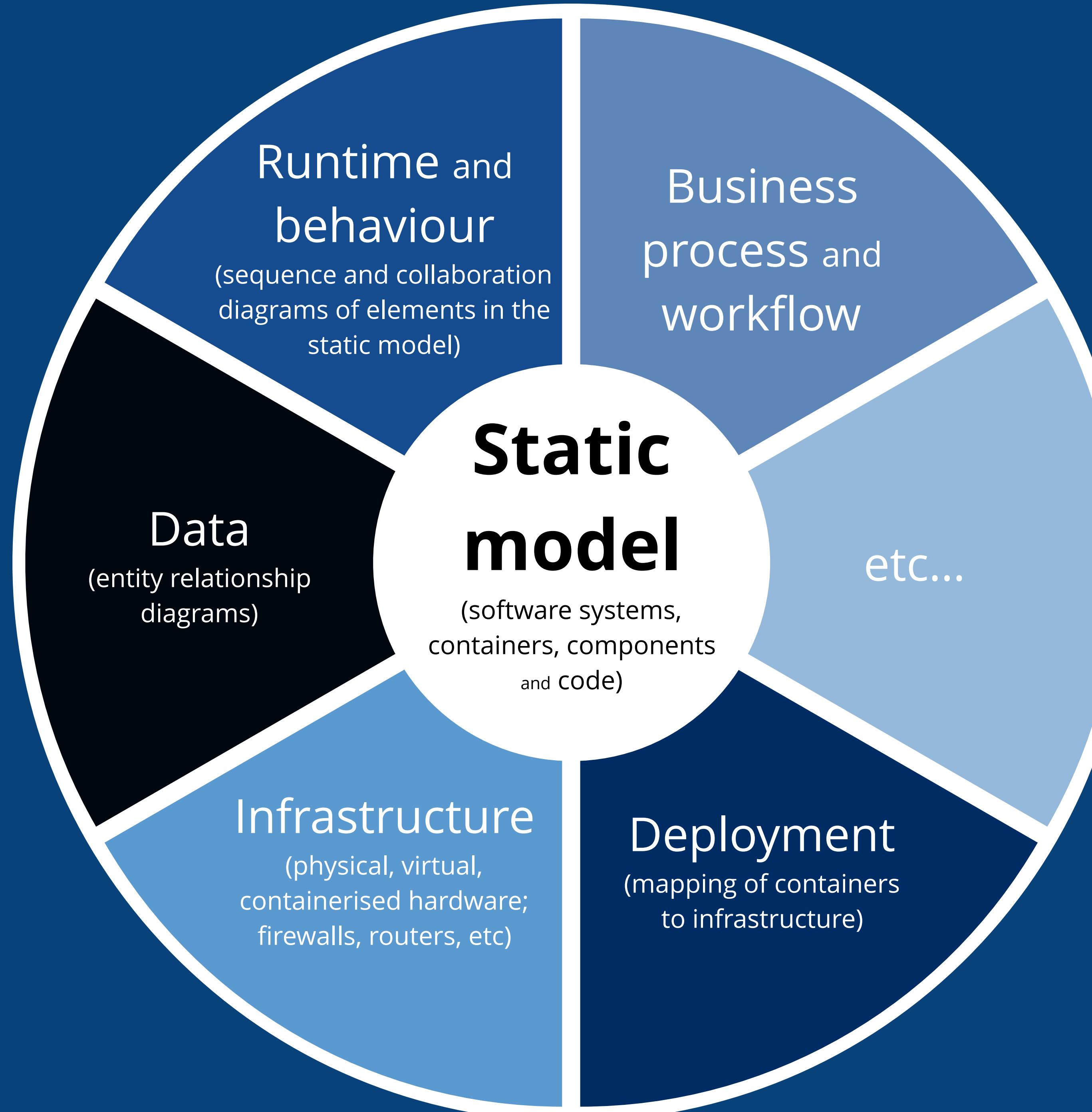
Zoom & filter

Details on demand



Diagrams are maps

that help software developers navigate a large and/or complex codebase



A model of the
static structure
forms the basis
for other views

Example

(Internet Banking System)

Level 1

System Context diagram

Internet Banking System

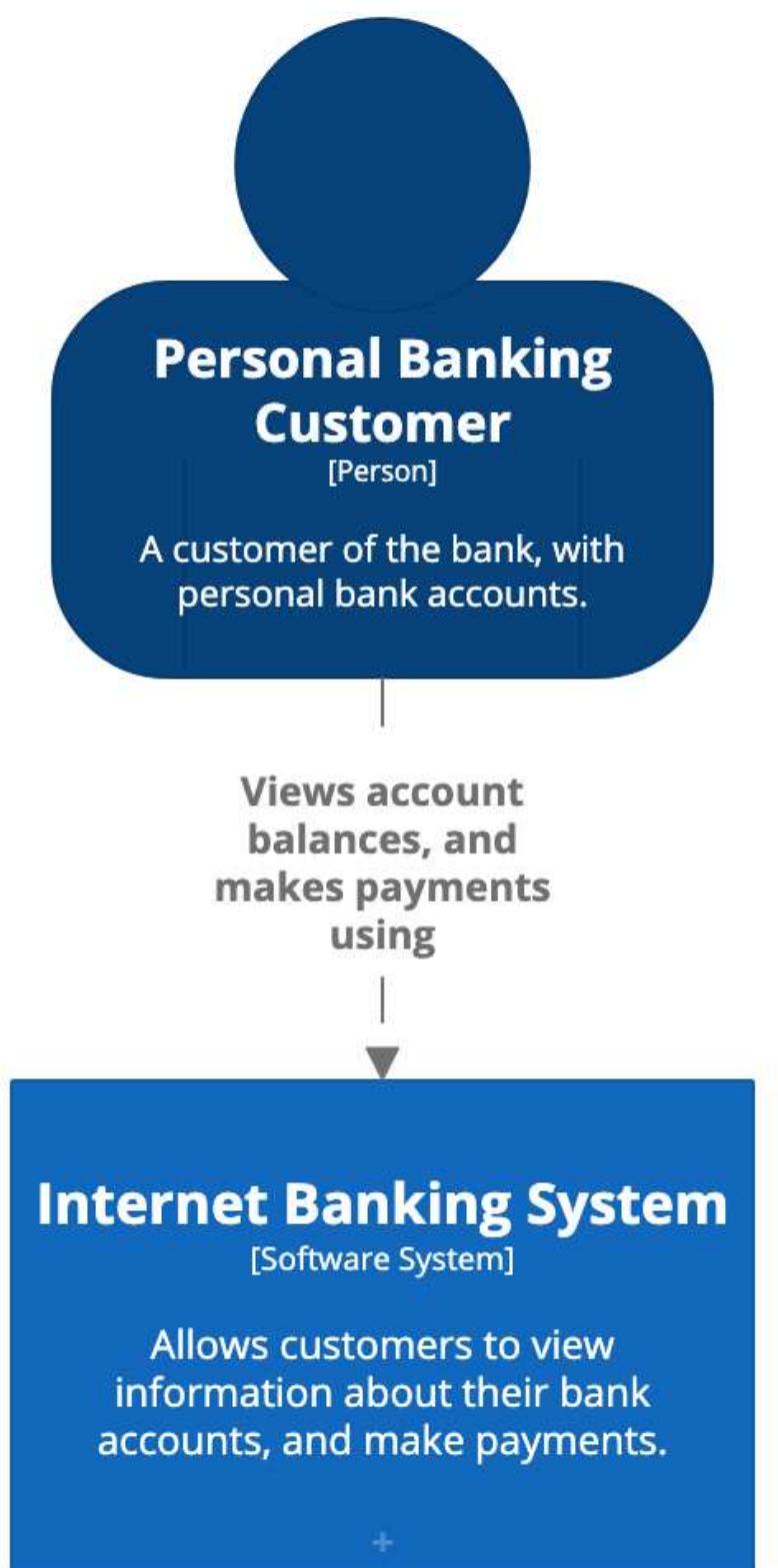
[Software System]

Allows customers to view information about their bank accounts, and make payments.

System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.

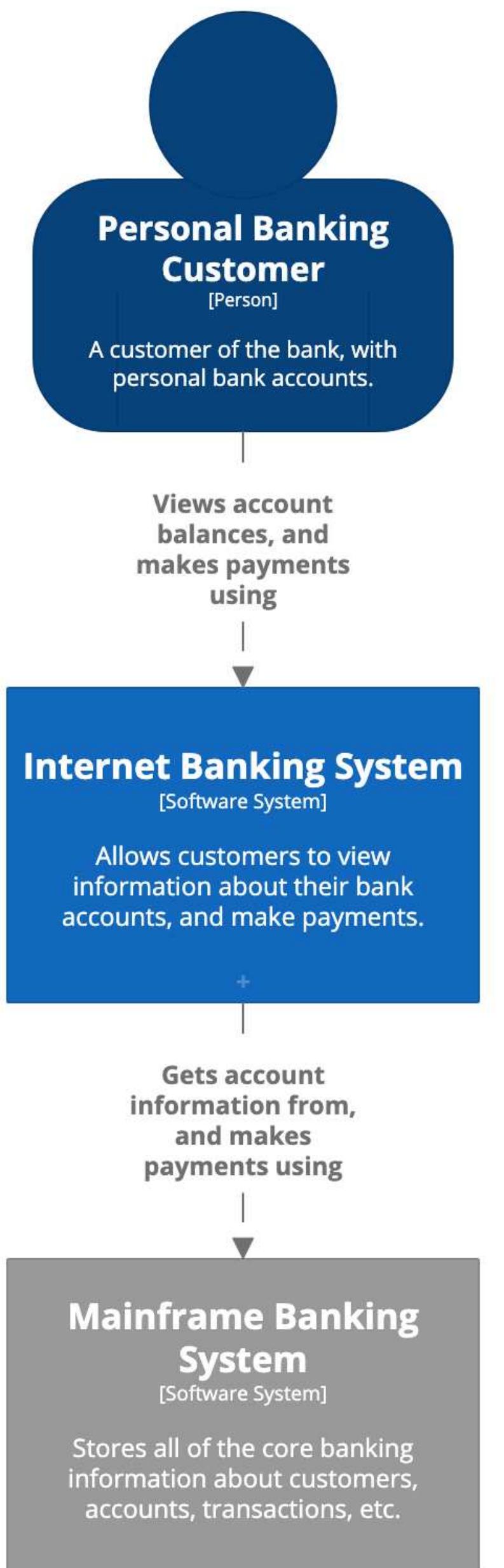
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.

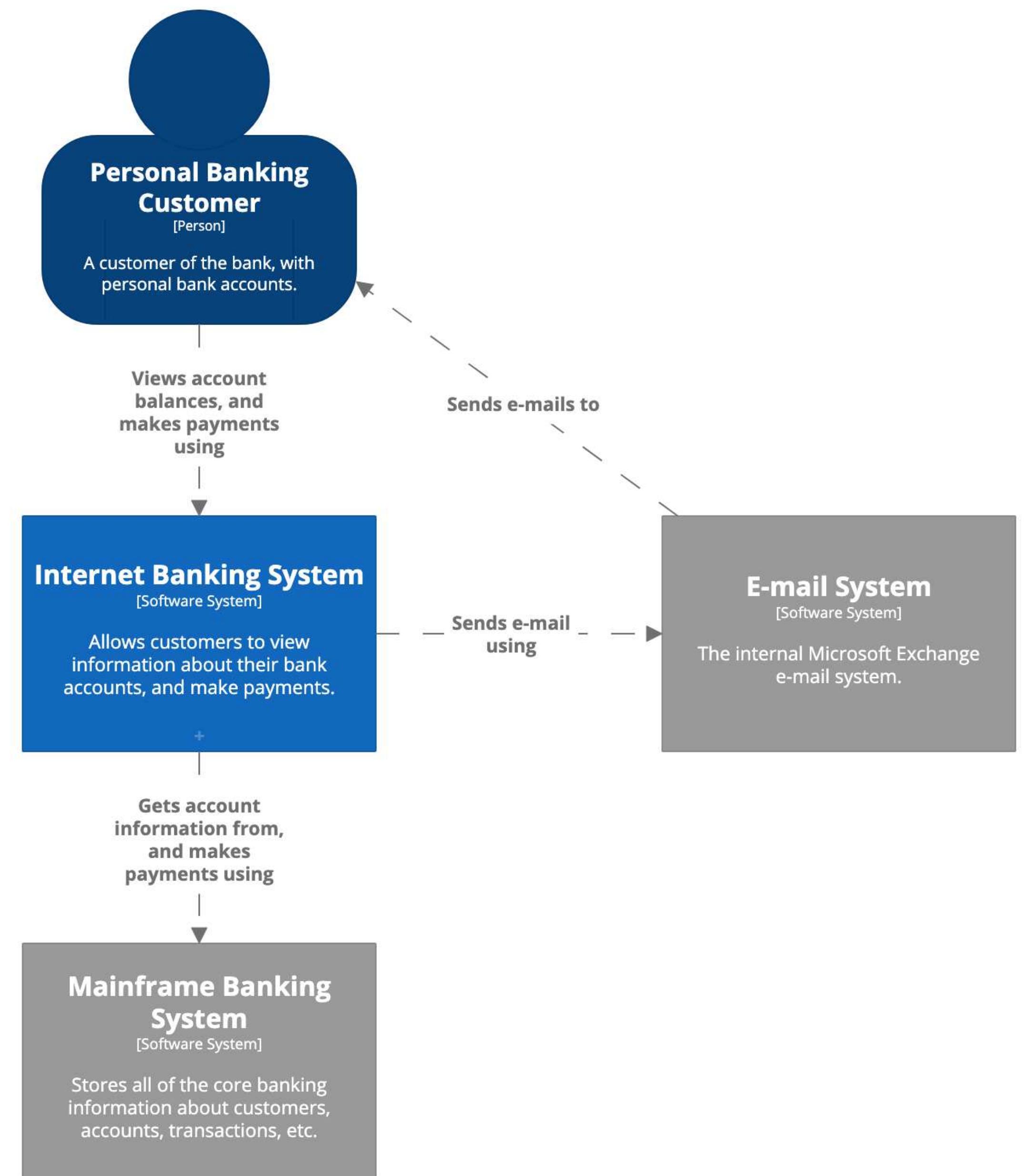
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.

Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



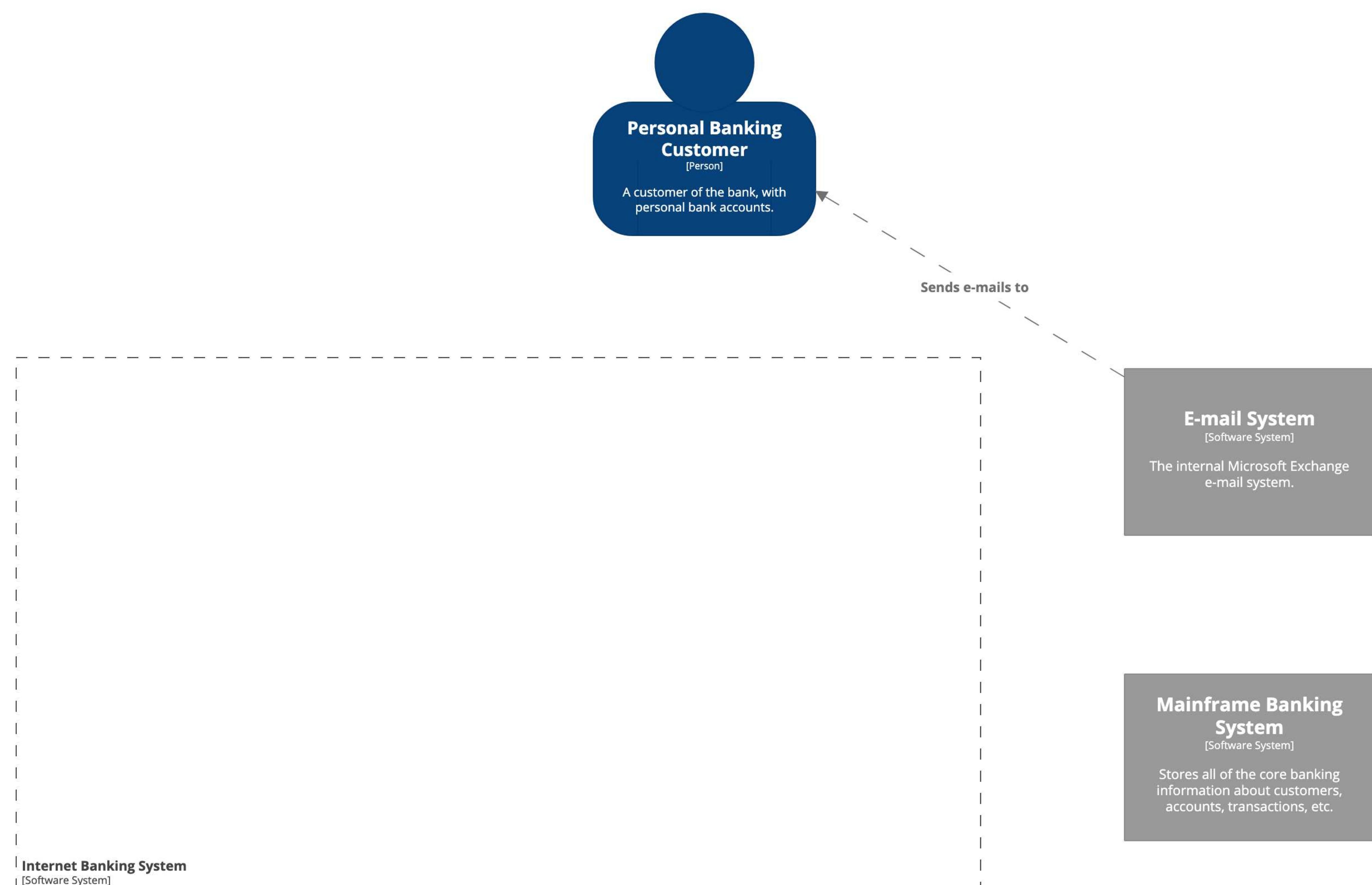
System Context diagram for Internet Banking System

The system context diagram for the Internet Banking System.

Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)

Level 2

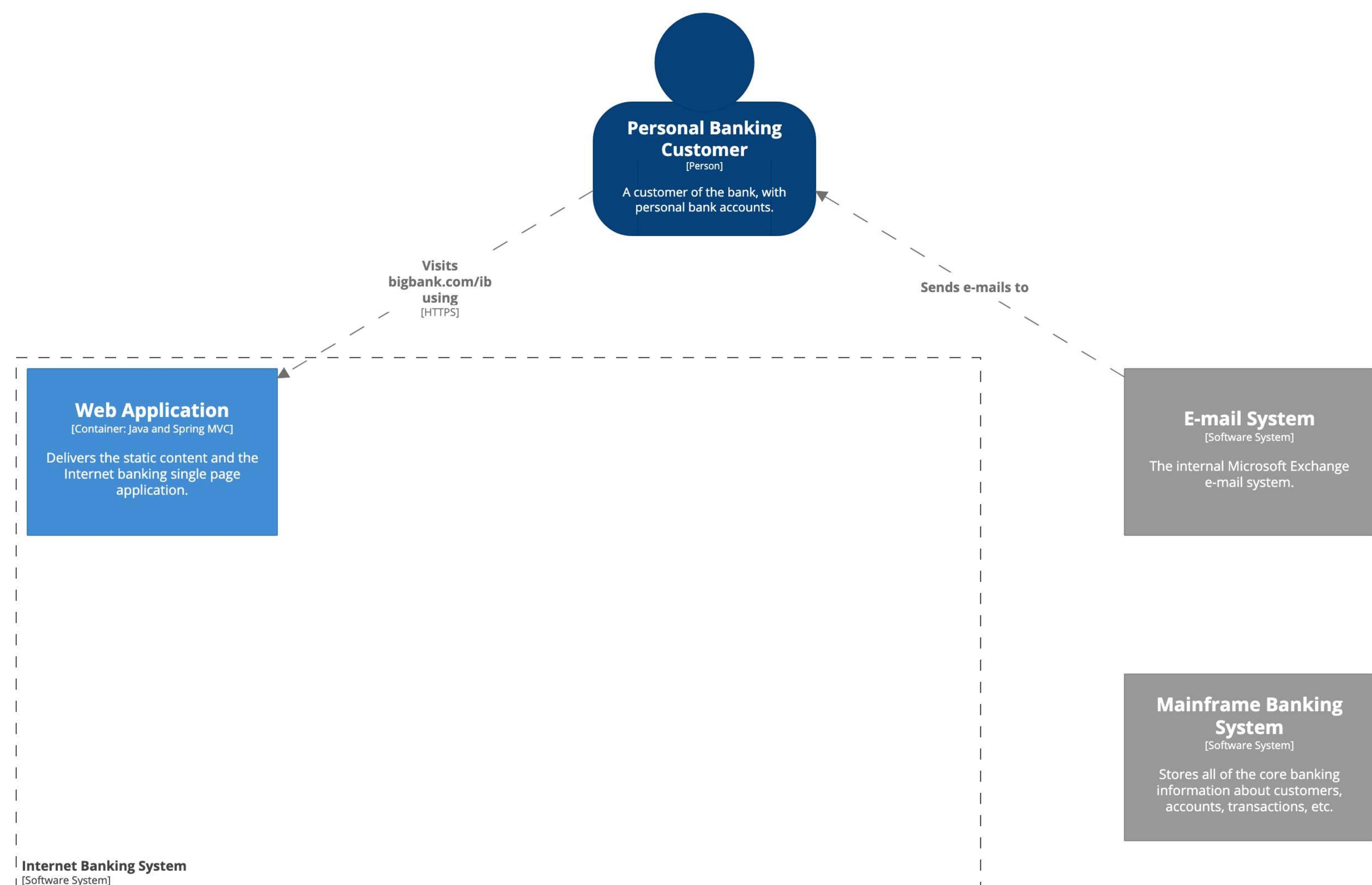
Container diagram



Container diagram for Internet Banking System

The container diagram for the Internet Banking System.

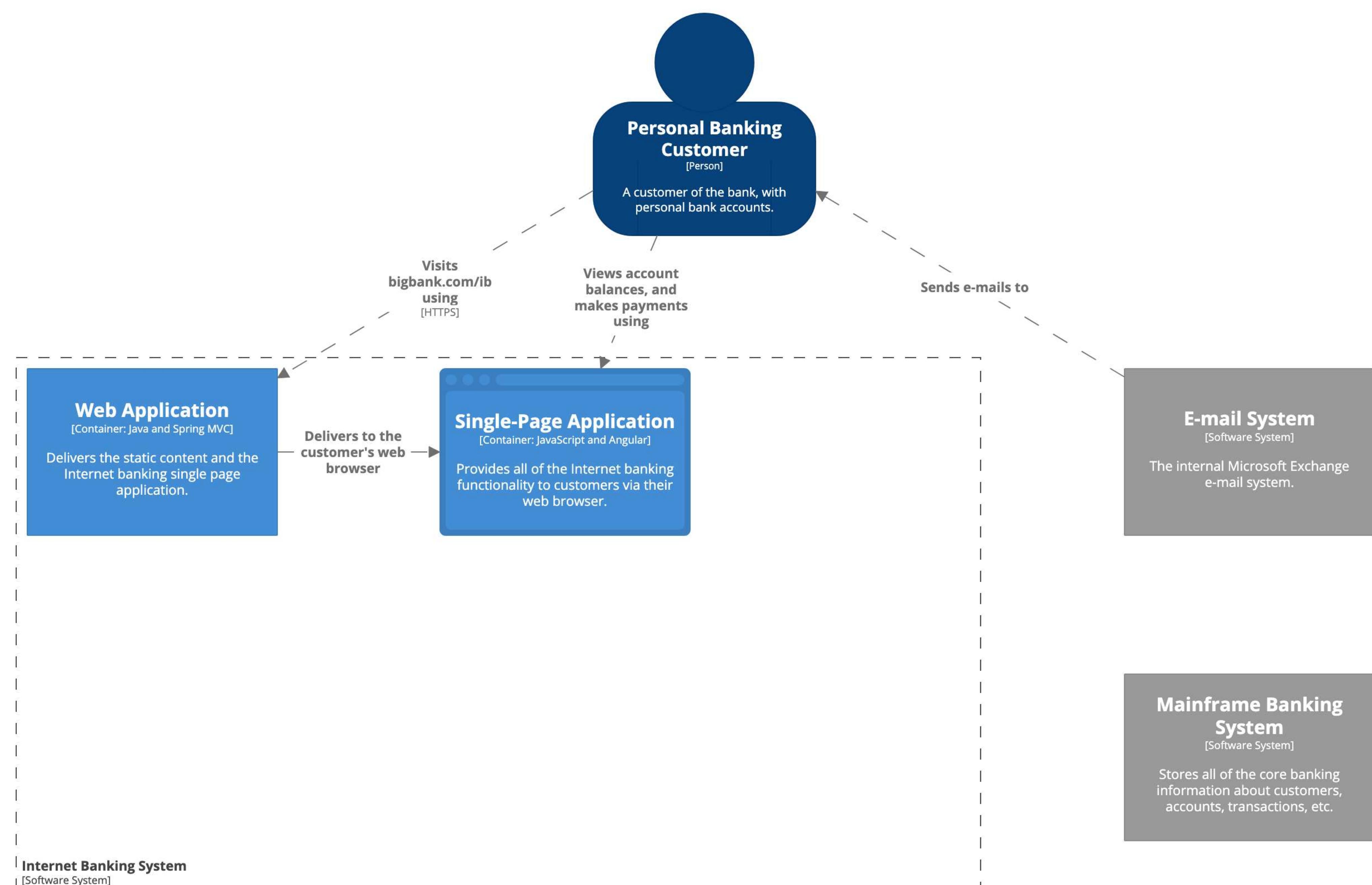
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



Container diagram for Internet Banking System

The container diagram for the Internet Banking System.

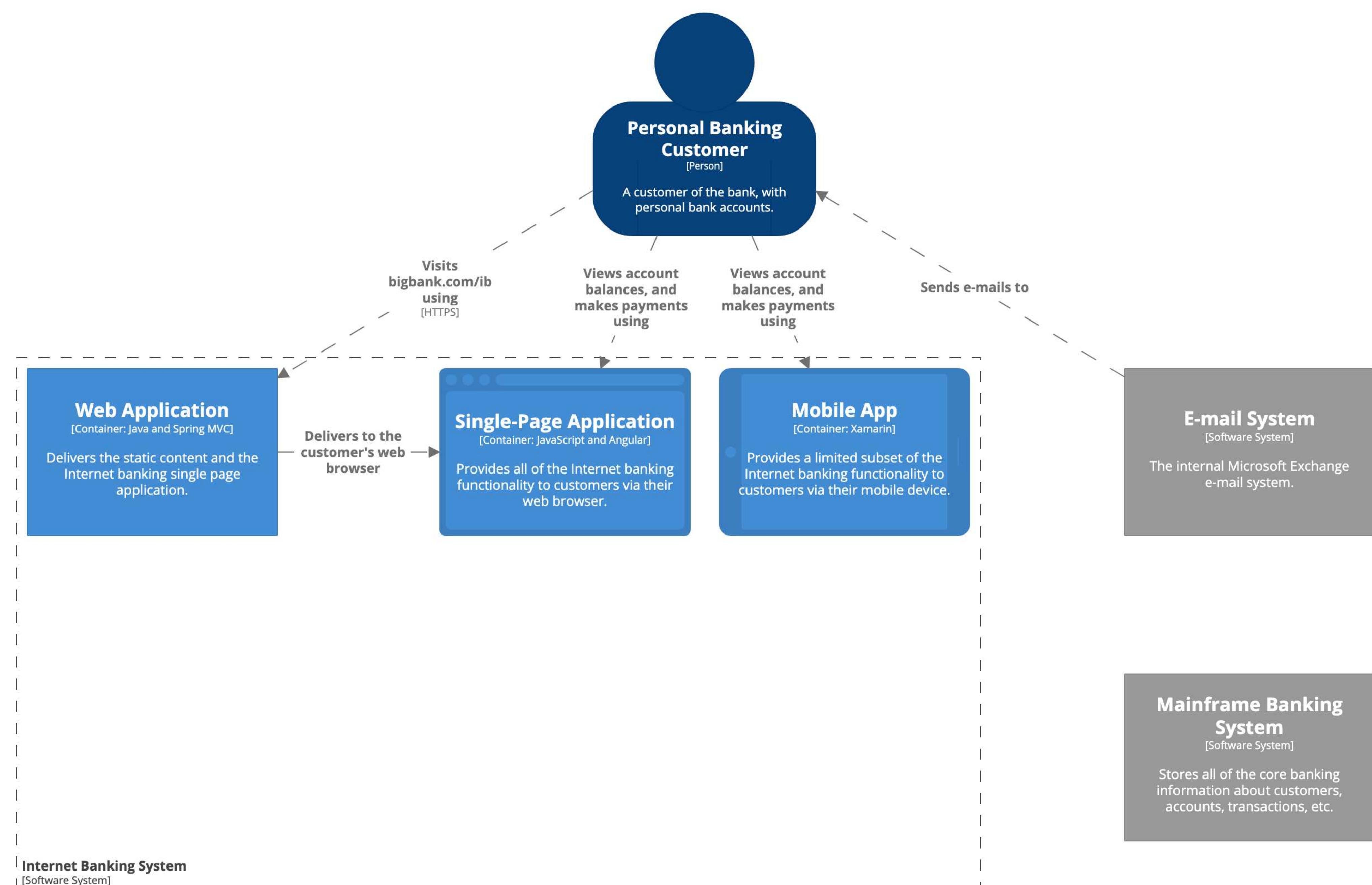
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



Container diagram for Internet Banking System

The container diagram for the Internet Banking System.

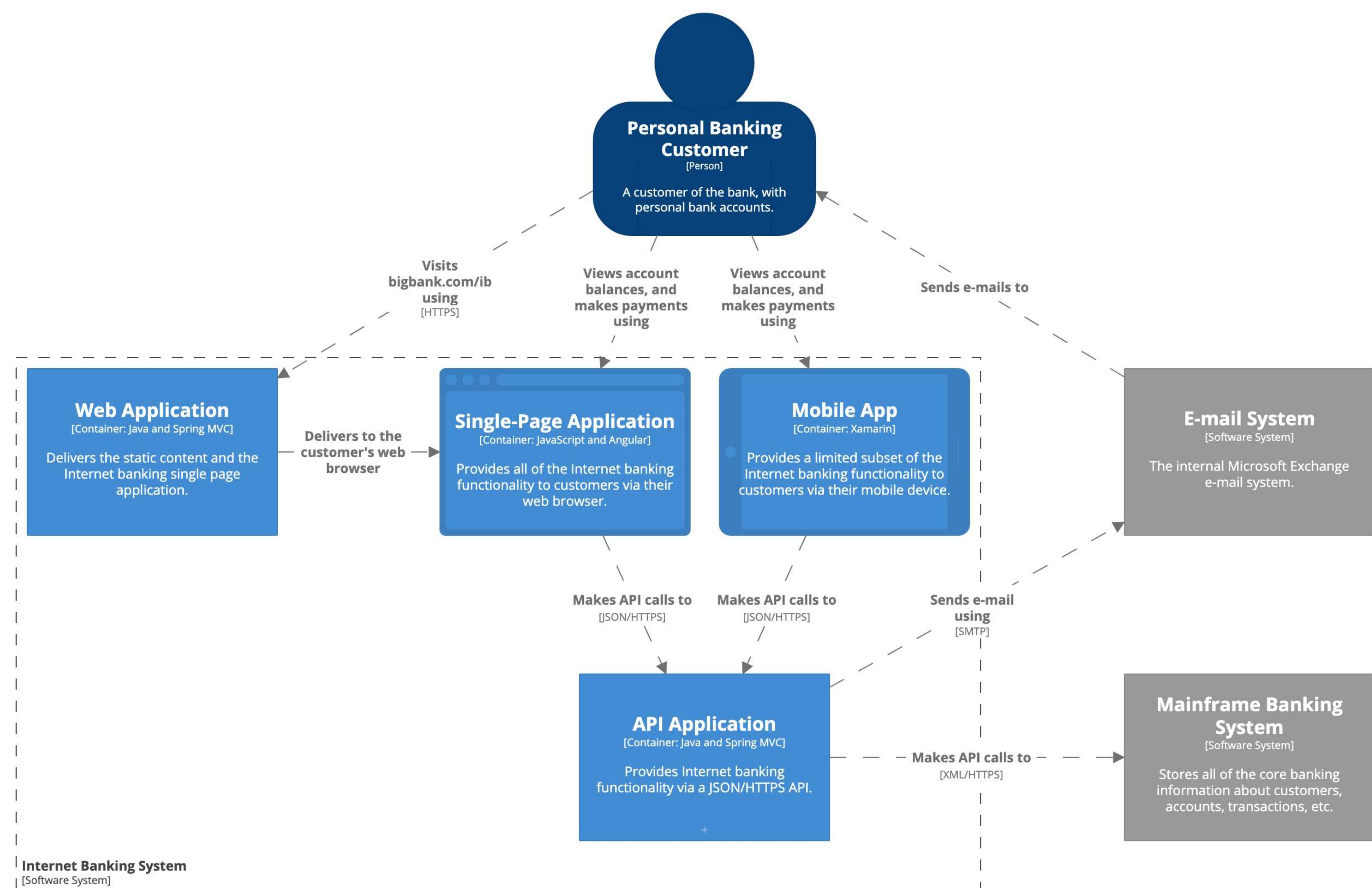
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



Container diagram for Internet Banking System

The container diagram for the Internet Banking System.

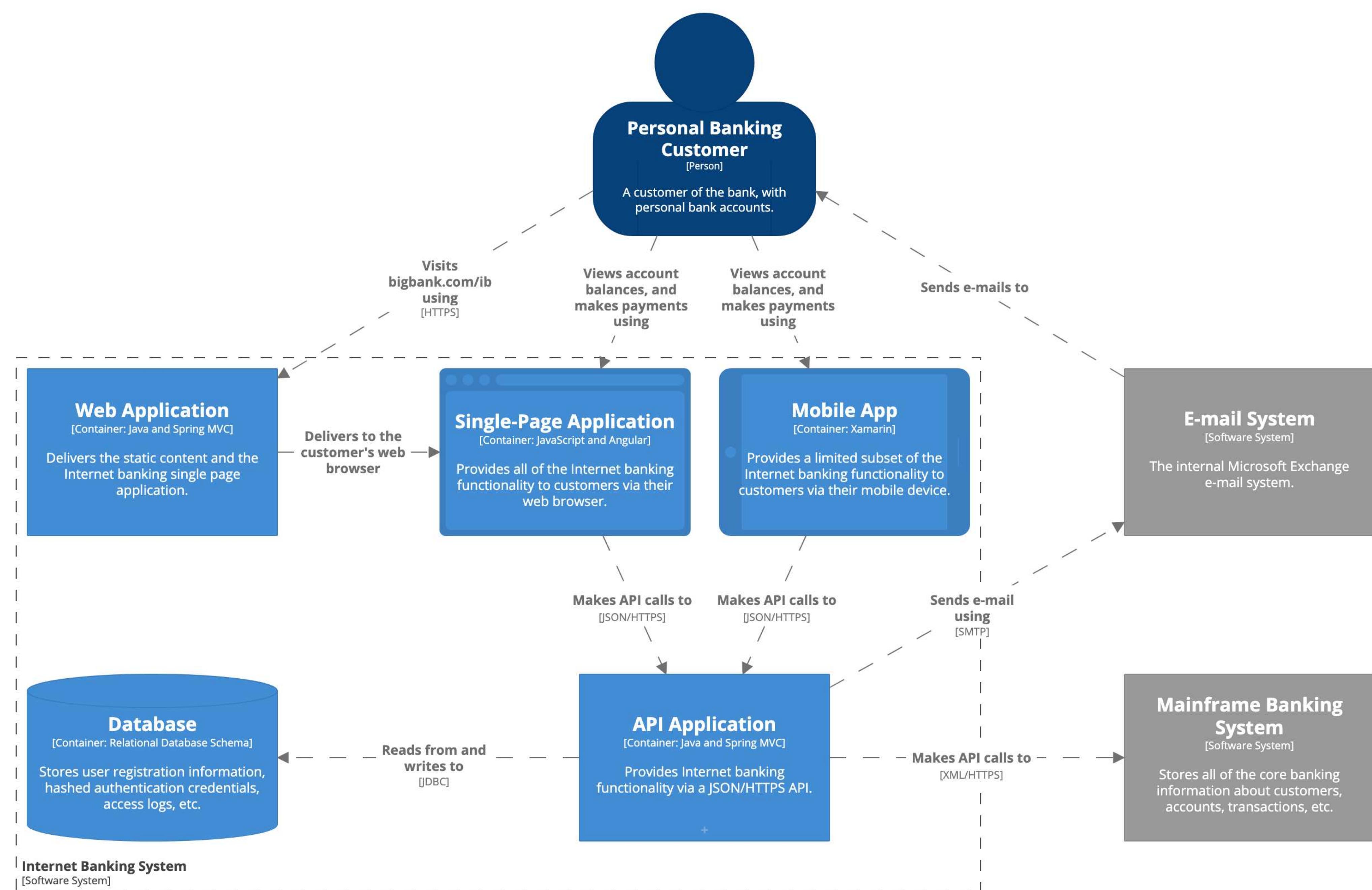
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



Container diagram for Internet Banking System

The container diagram for the Internet Banking System.

Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



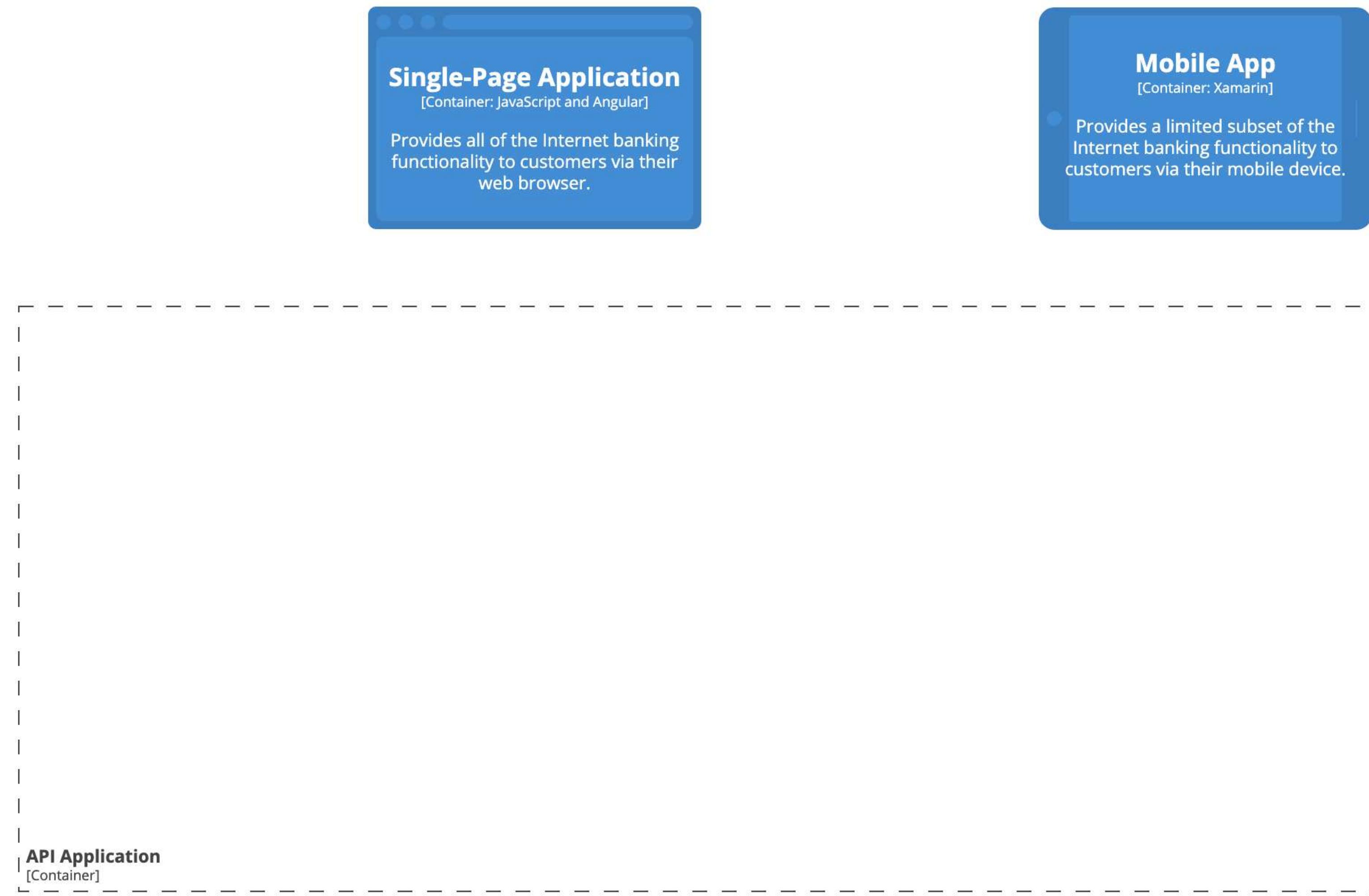
Container diagram for Internet Banking System

The container diagram for the Internet Banking System.

Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)

Level 3

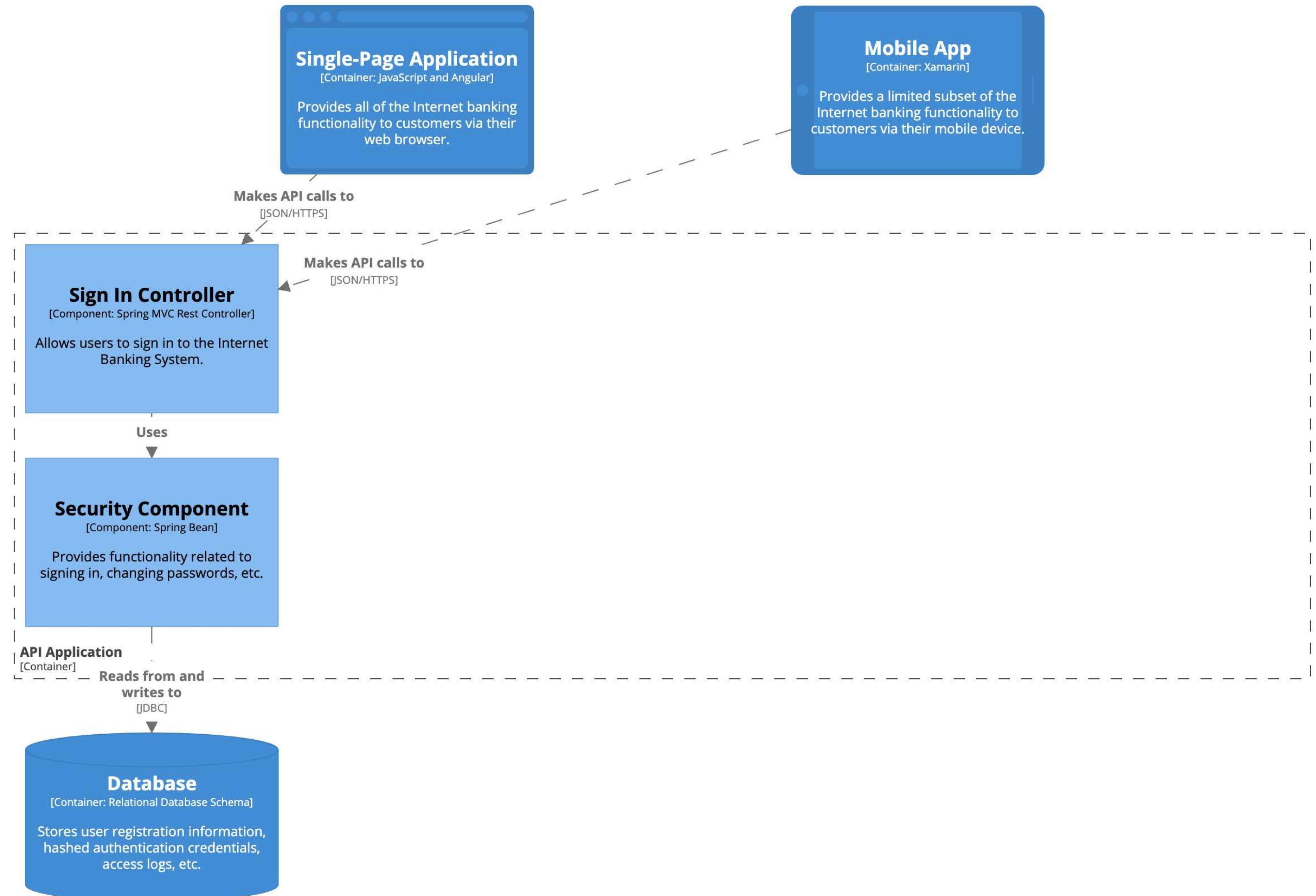
Component diagram



Component diagram for Internet Banking System - API Application

The component diagram for the API Application.

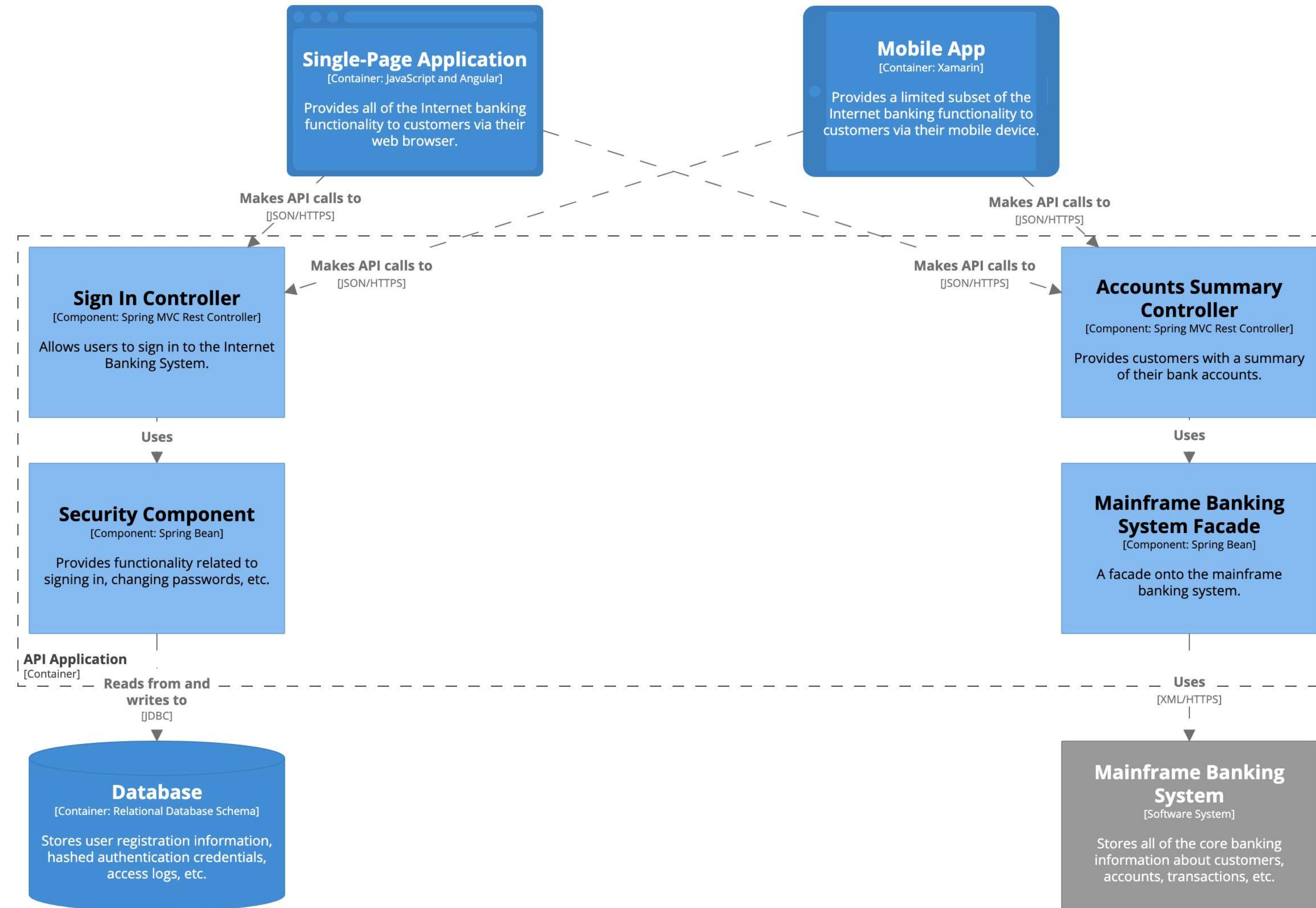
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



Component diagram for Internet Banking System - API Application

The component diagram for the API Application.

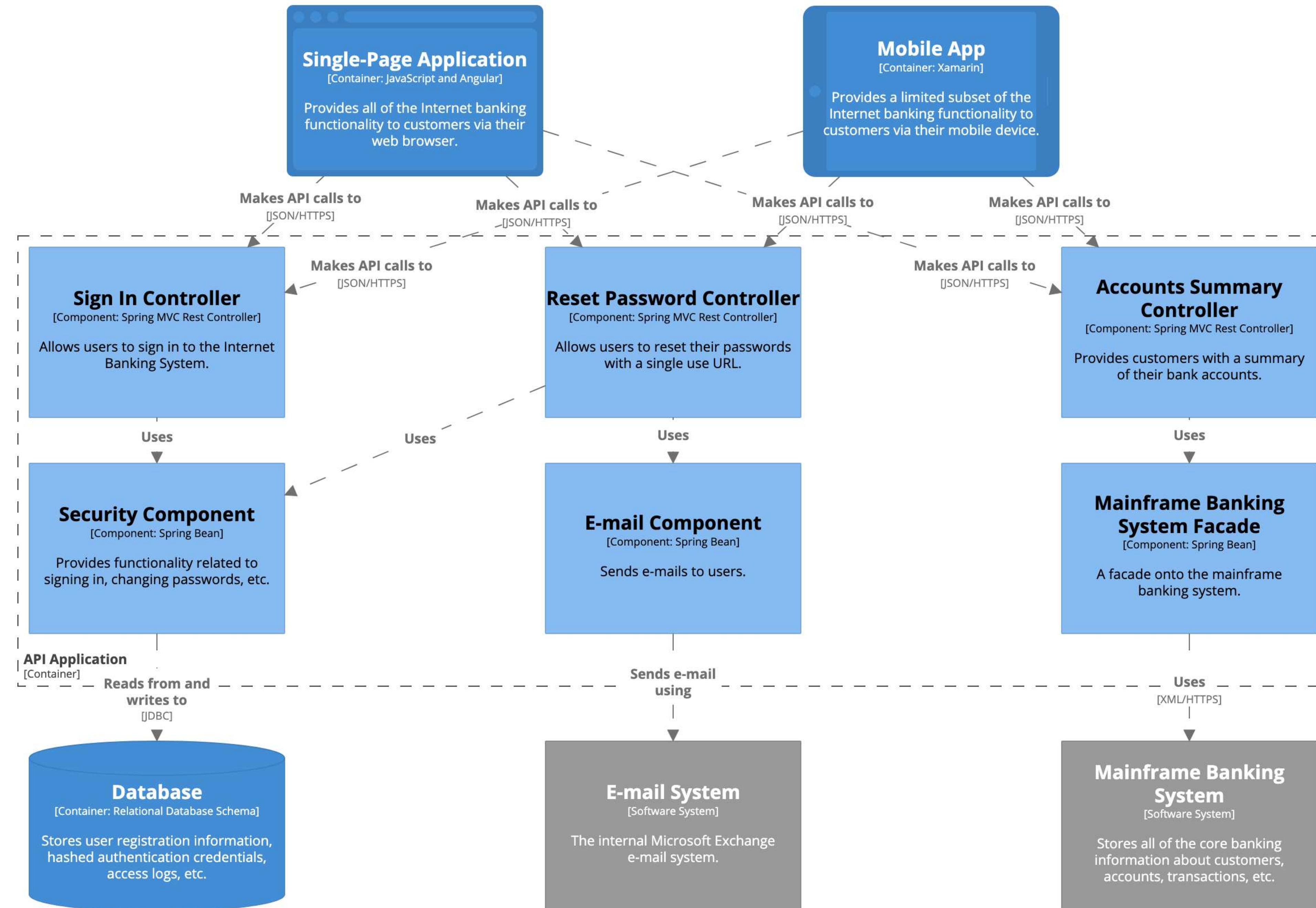
Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



Component diagram for Internet Banking System - API Application

The component diagram for the API Application.

Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)



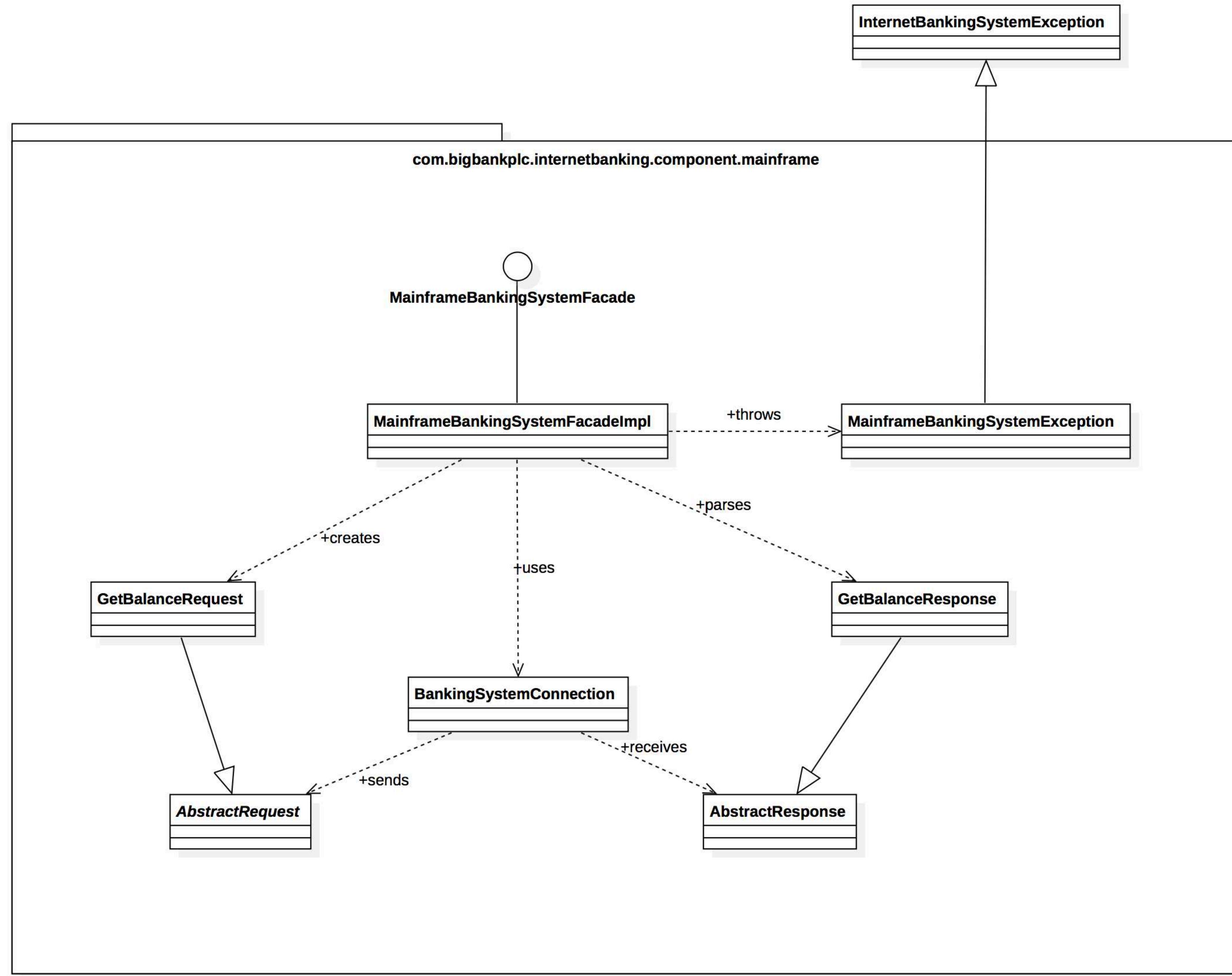
Component diagram for Internet Banking System - API Application

The component diagram for the API Application.

Workspace last modified: Thu Apr 04 2019 13:10:49 GMT+0100 (British Summer Time)

Level 4

Class diagram



Notation

Titles

Short and meaningful, include the **diagram type**, numbered if diagram order is important; for example:

System Context diagram for Financial Risk System
[System Context] Financial Risk System

Layout

Sticky notes and index cards (e.g. CRC cards)
make a great substitute for hand-drawn boxes,
especially if you don't have a whiteboard

Visual consistency

Try to be consistent with notation
and element positioning across diagrams

Acronyms

Be wary of using acronyms, especially those related
to the business/domain that you work in

Elements

Start with simple boxes containing the element name, type, technology (if appropriate) and a description/responsibilities

Anonymous User

[Person]

Anybody on the web.

Web Application

[Container: Java + Spring MVC]

Allows users to view people, tribes, content, events, jobs, etc from the local tech, digital and IT sector.

techtribes.je

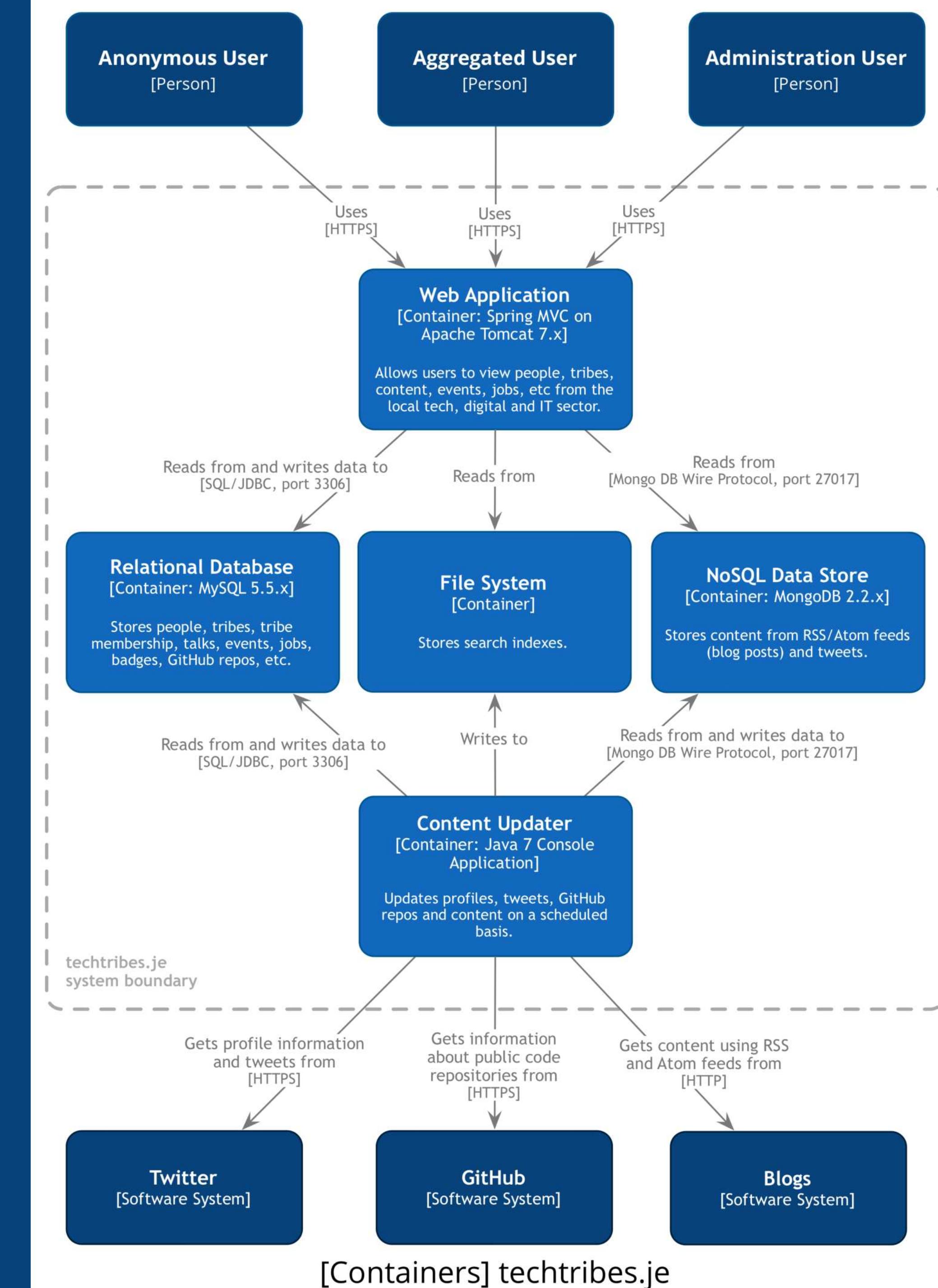
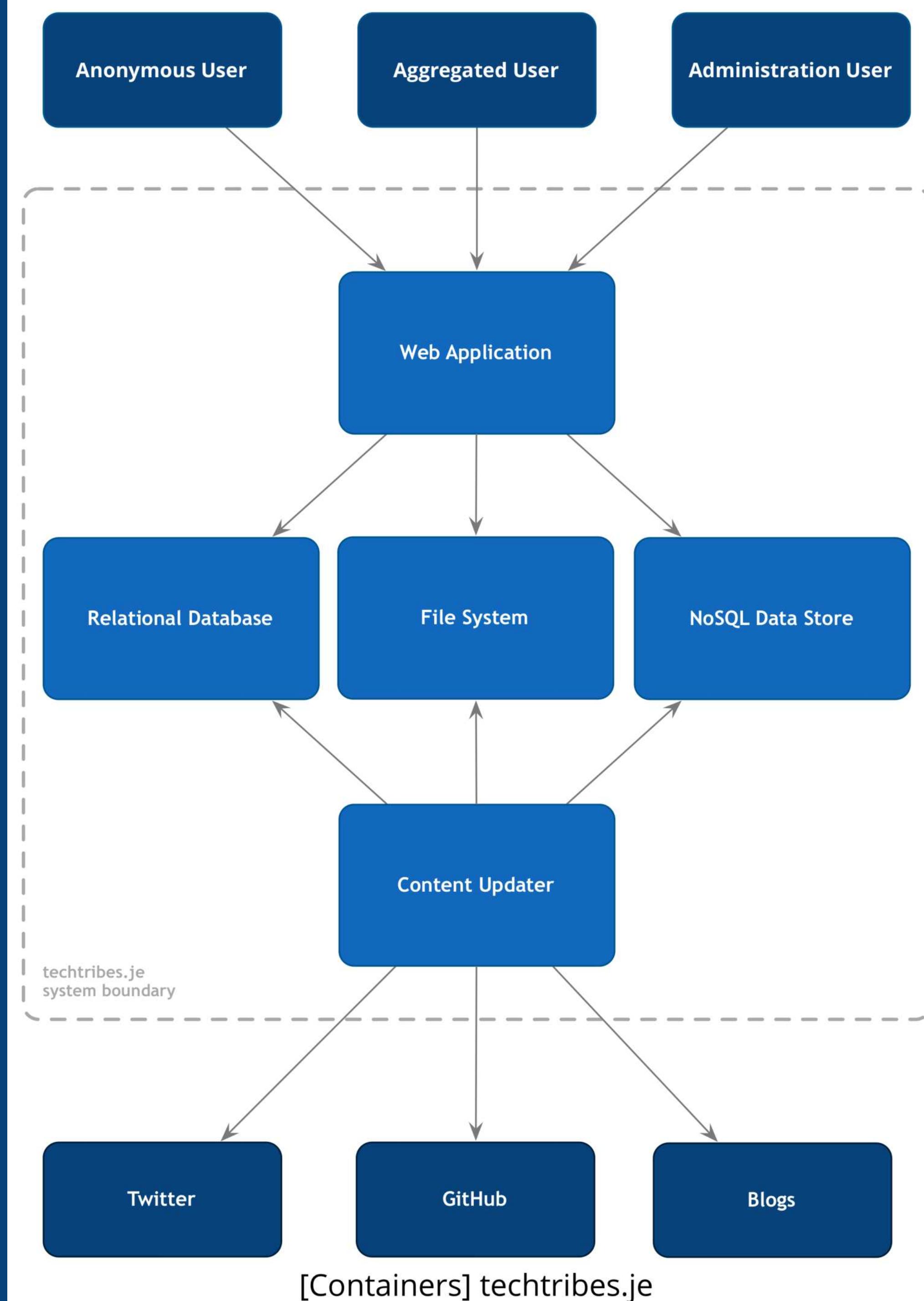
[Software System]

techtribes.je is the only way to keep up to date with the IT, tech and digital sector in Jersey and Guernsey, Channel Islands.

Twitter Connector

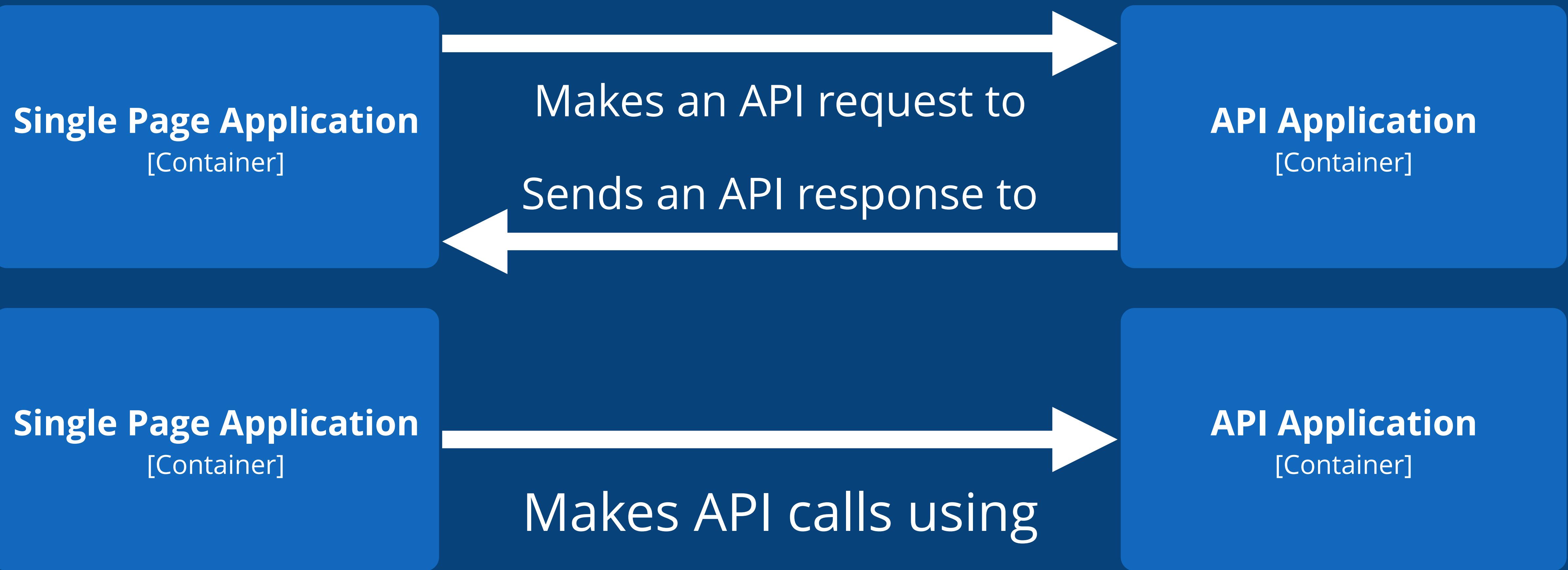
[Component: Spring Bean + Twitter4j]

Retrieves profile information and tweets (using the REST and Streaming APIs).



Lines

Favour uni-directional lines showing the most important dependencies or data flow, with an annotation to be explicit about the purpose of the line and direction



Summarise the intent of the relationship



Summarise, yet be specific

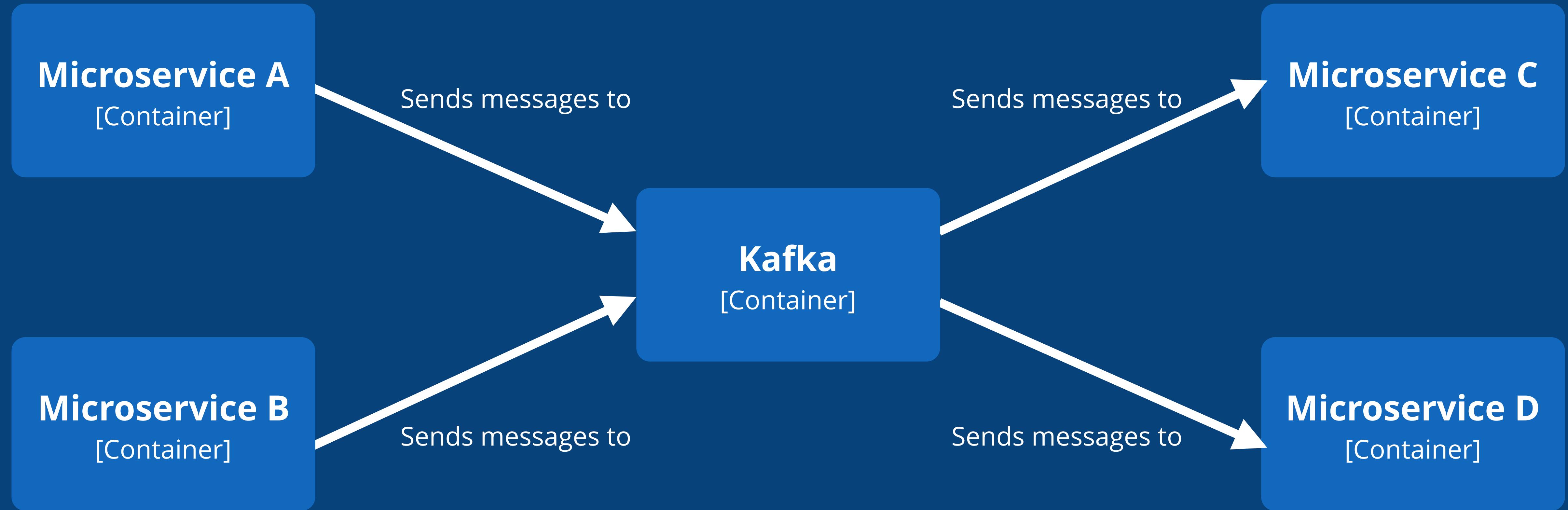
Microservice A
[Container]

Requests a list of customers from
[JSON/HTTPS]

Microservice B
[Container]

Sends new customers to
[Kafka topic]

Show both directions when
the intents are different



Beware of hiding the true story



Sends customer update messages to
[via Kafka topic X]



Sends order creation messages to
[via Kafka topic Y]



Beware of hiding the true story

Trade Data System

[Software System]

Financial Risk System

[Software System]

Trade data

Trade Data System

[Software System]

Financial Risk System

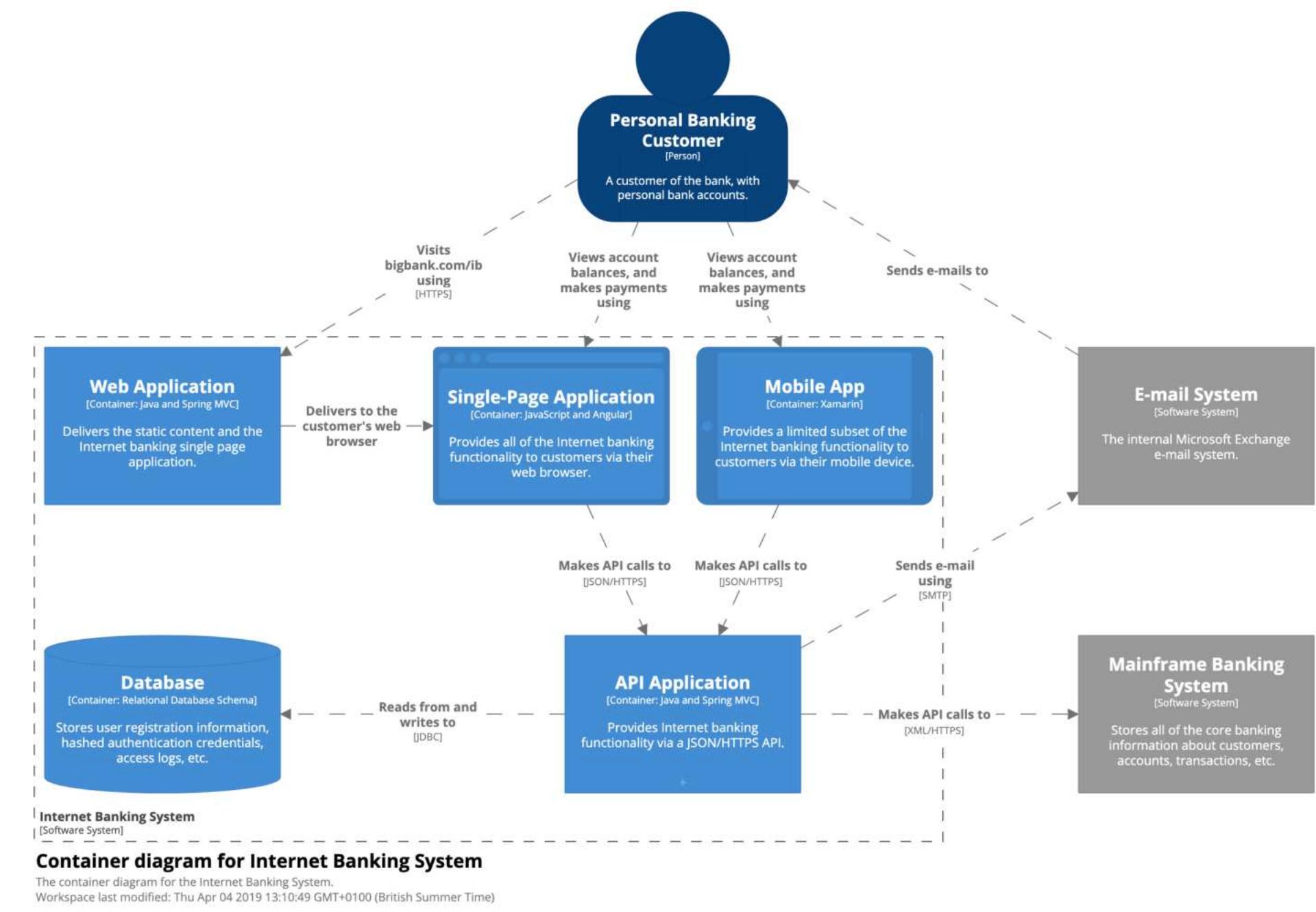
[Software System]

Sends trade data to

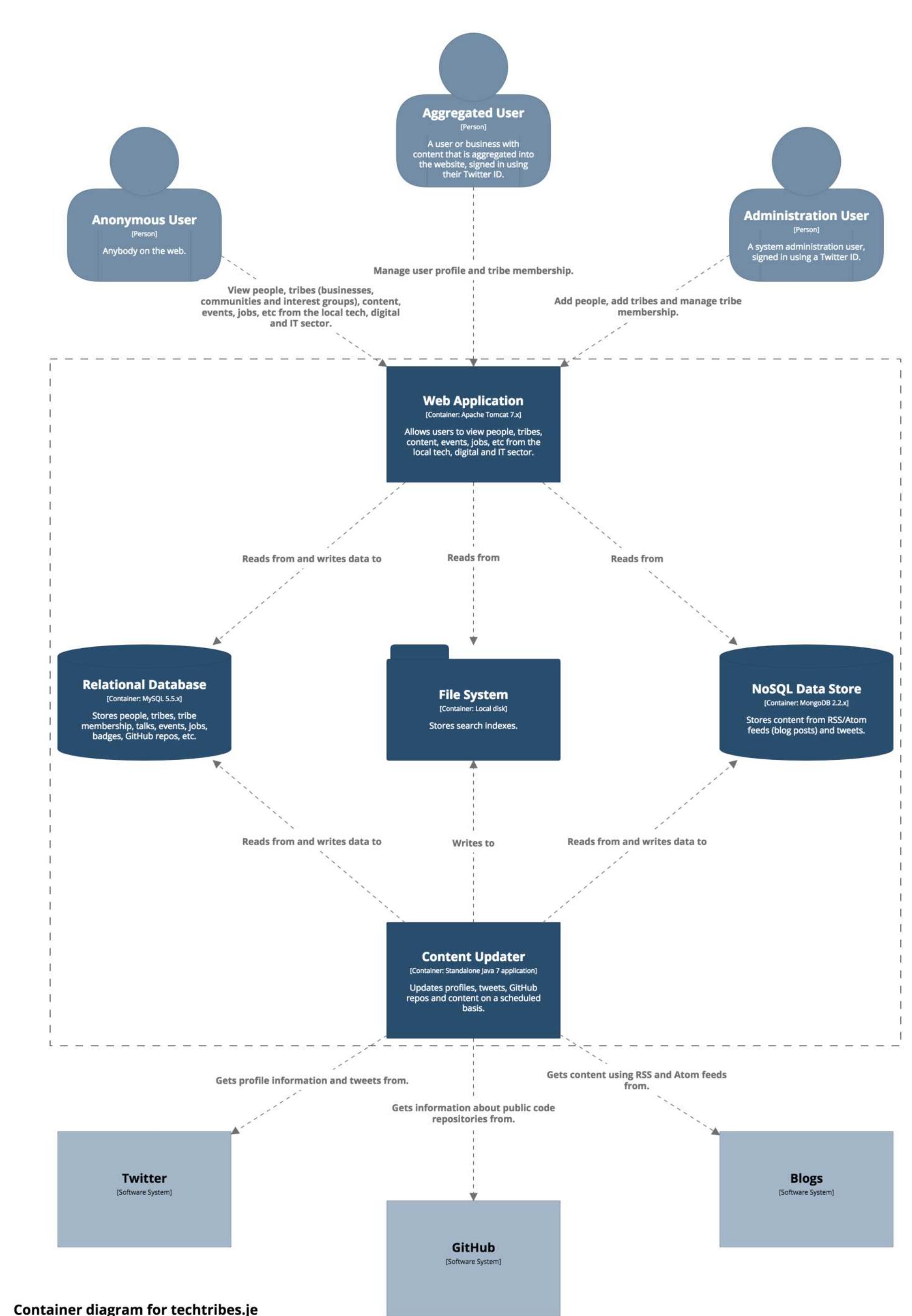
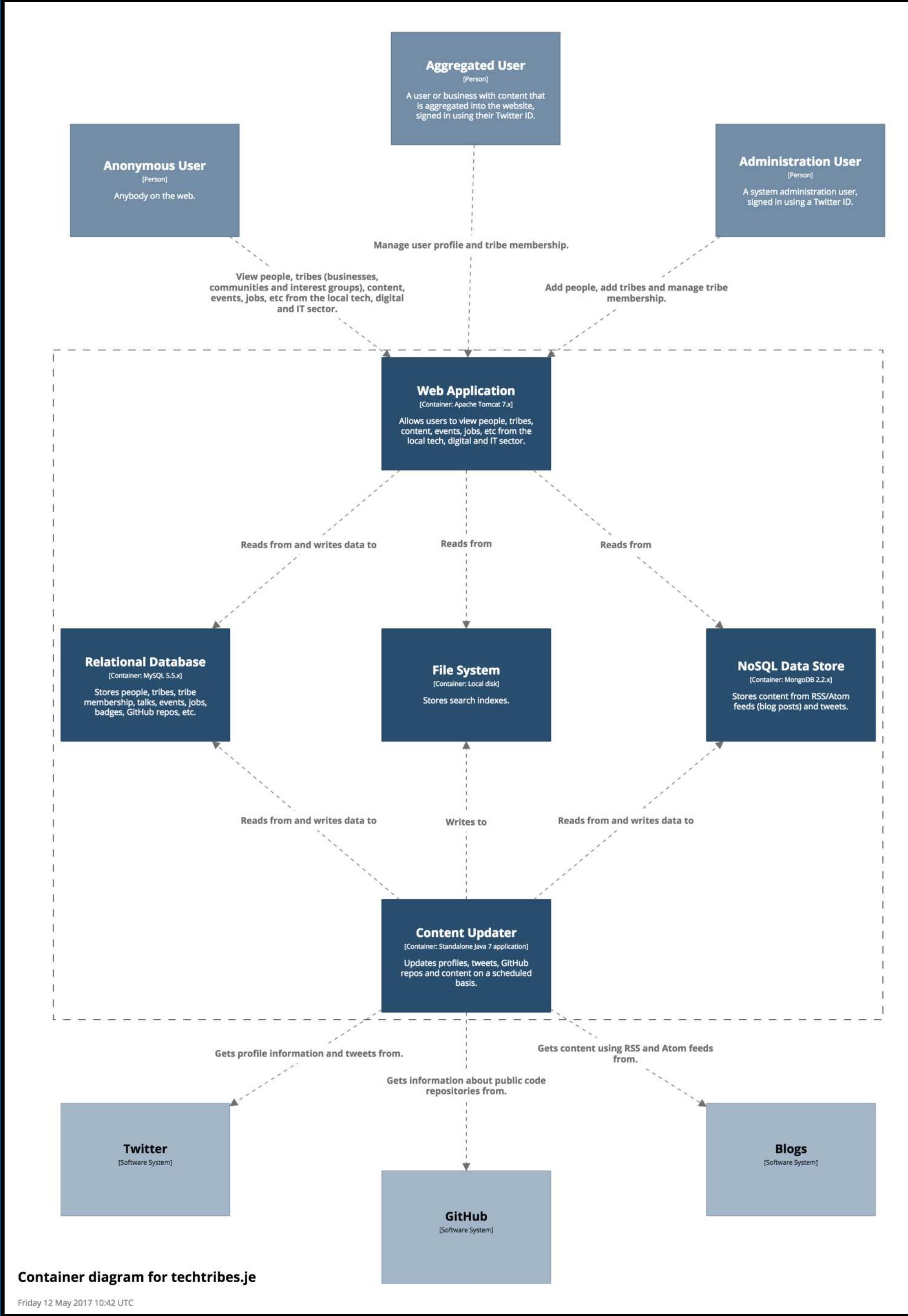
Add more words to make the intent explicit

Key/legend

Explain shapes, line styles, colours, borders, acronyms, etc
... even if your notation seems obvious!



Use shape, colour and size
to complement a diagram
that already makes sense

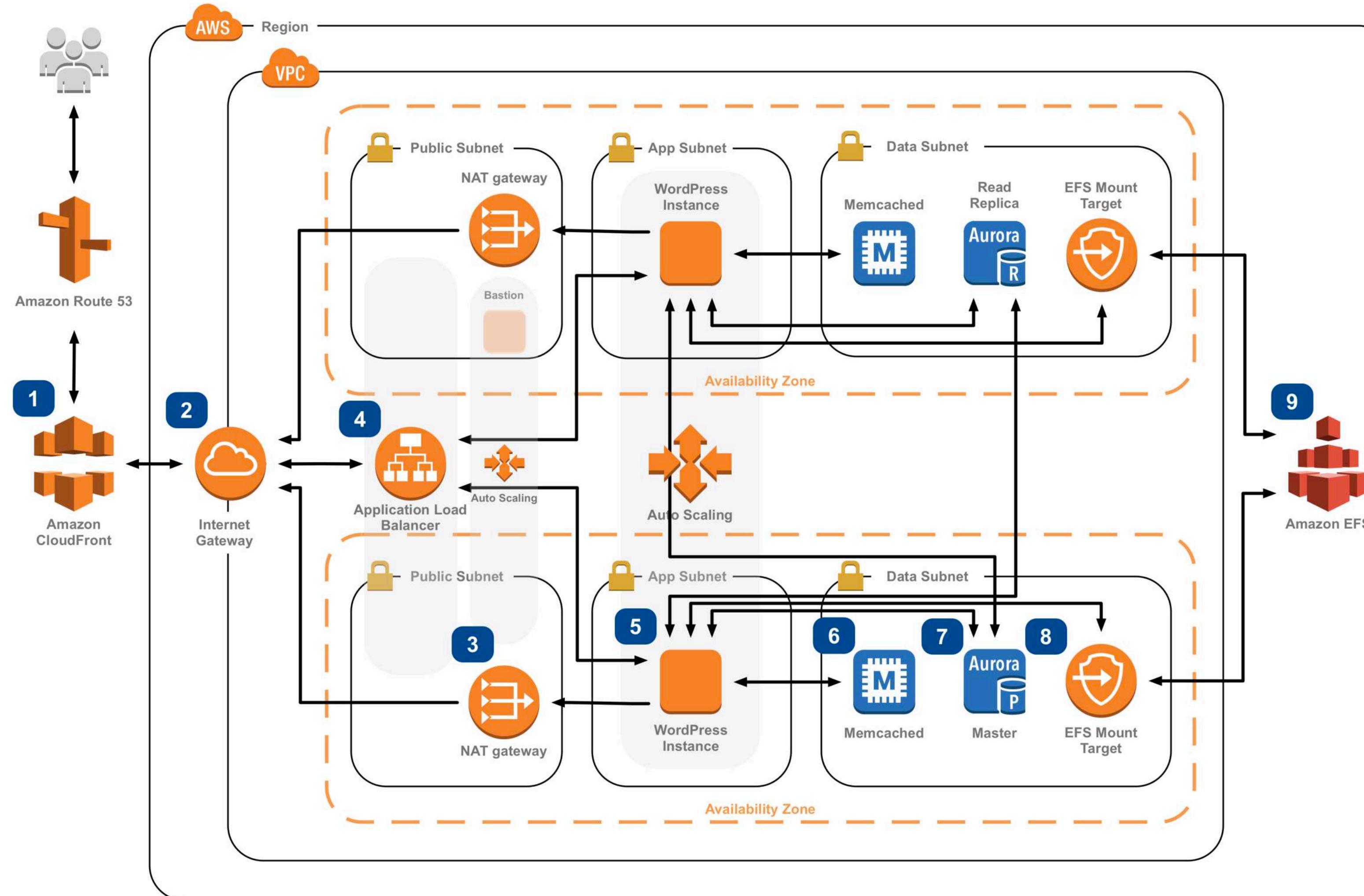


Use icons to supplement text,
not replace it

WordPress Hosting

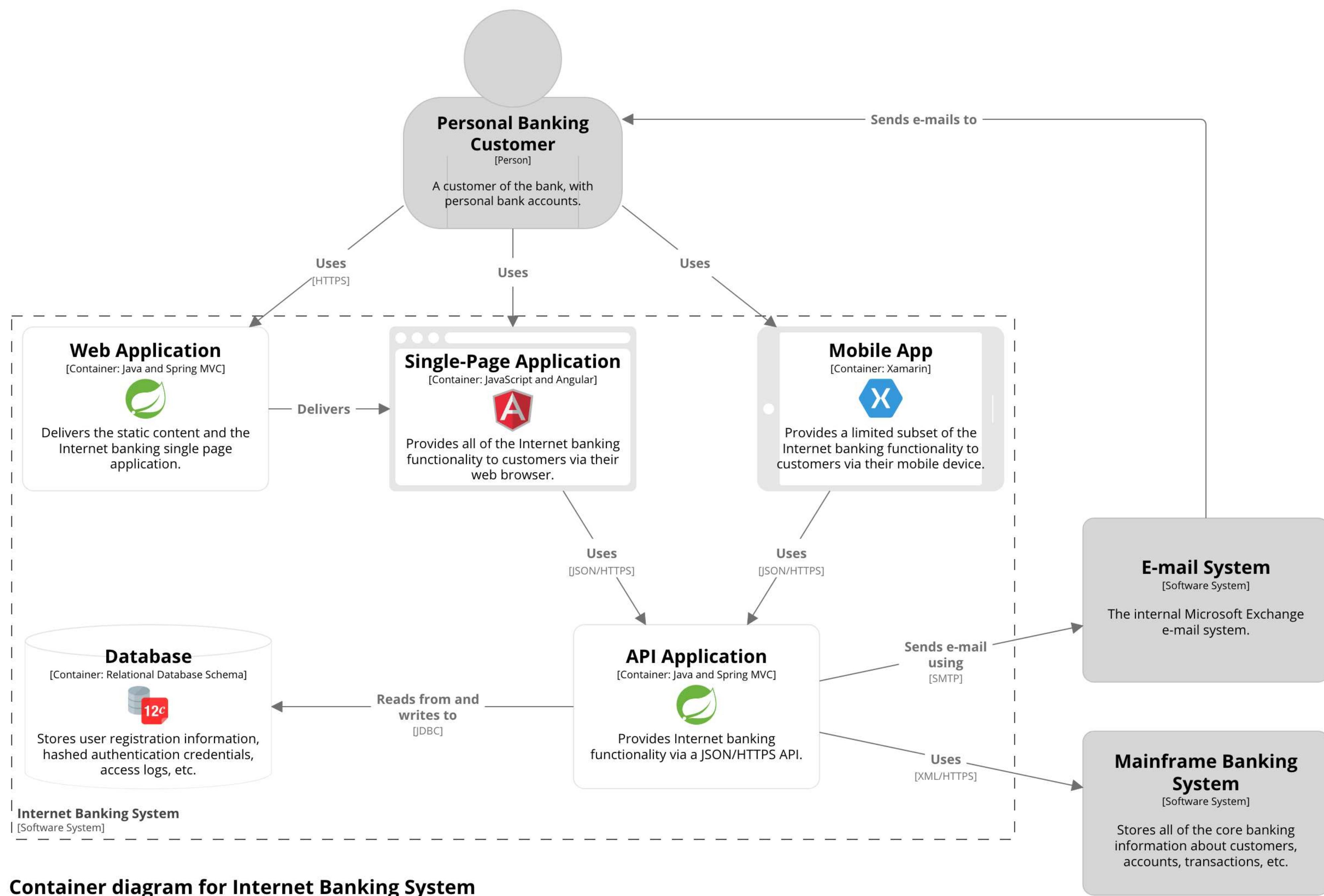
How to run WordPress on AWS

WordPress is one of the world's most popular web publishing platforms, being used to publish 27% of all websites, from personal blogs to some of the biggest news sites. This reference architecture simplifies the complexity of deploying a scalable and highly available WordPress site on AWS.



- 1 Static and dynamic content is delivered by **Amazon CloudFront**.
- 2 An **Internet gateway** allows communication between instances in your VPC and the Internet.
- 3 **NAT gateways** in each public subnet enable Amazon EC2 instances in private subnets (App & Data) to access the Internet.
- 4 Use an **Application Load Balancer** to distribute web traffic across an Auto Scaling Group of Amazon EC2 instances in multiple AZs.
- 5 Run your WordPress site using an **Auto Scaling group of Amazon EC2 instances**. Install the latest versions of WordPress, Apache web server, PHP 7, and OPCache and build an Amazon Machine Image that will be used by the Auto Scaling group launch configuration to launch new instances in the Auto Scaling group.
- 6 If database access patterns are read-heavy, consider using a WordPress plugin that takes advantage of a caching layer like **Amazon ElastiCache (Memcached)** in front of the database layer to cache frequently accessed data.
- 7 Simplify your database administration by running your database layer in **Amazon RDS** using either Aurora or MySQL.
- 8 Amazon EC2 instances access shared WordPress data in an Amazon EFS file system using **Mount Targets** in each AZ in your VPC.
- 9 Use **Amazon EFS**, a simple, highly available, and scalable network file system so WordPress instances have access to your shared, unstructured WordPress data, like php files, config, themes, plugins, etc.





Container diagram for Internet Banking System

The container diagram for the Internet Banking System.

Workspace last modified: Wed Nov 21 2018 15:02:17 GMT+0000 (Greenwich Mean Time)

Increase the readability of
software architecture diagrams,
so they can stand alone

Any narrative should complement
the diagram rather than explain it

Notation, notation, notation

A software architecture diagram review checklist

General

Does the diagram have a title?	Yes	No
Do you understand what the diagram type is?	Yes	No
Do you understand what the diagram scope is?	Yes	No
Does the diagram have a key/legend?	Yes	No

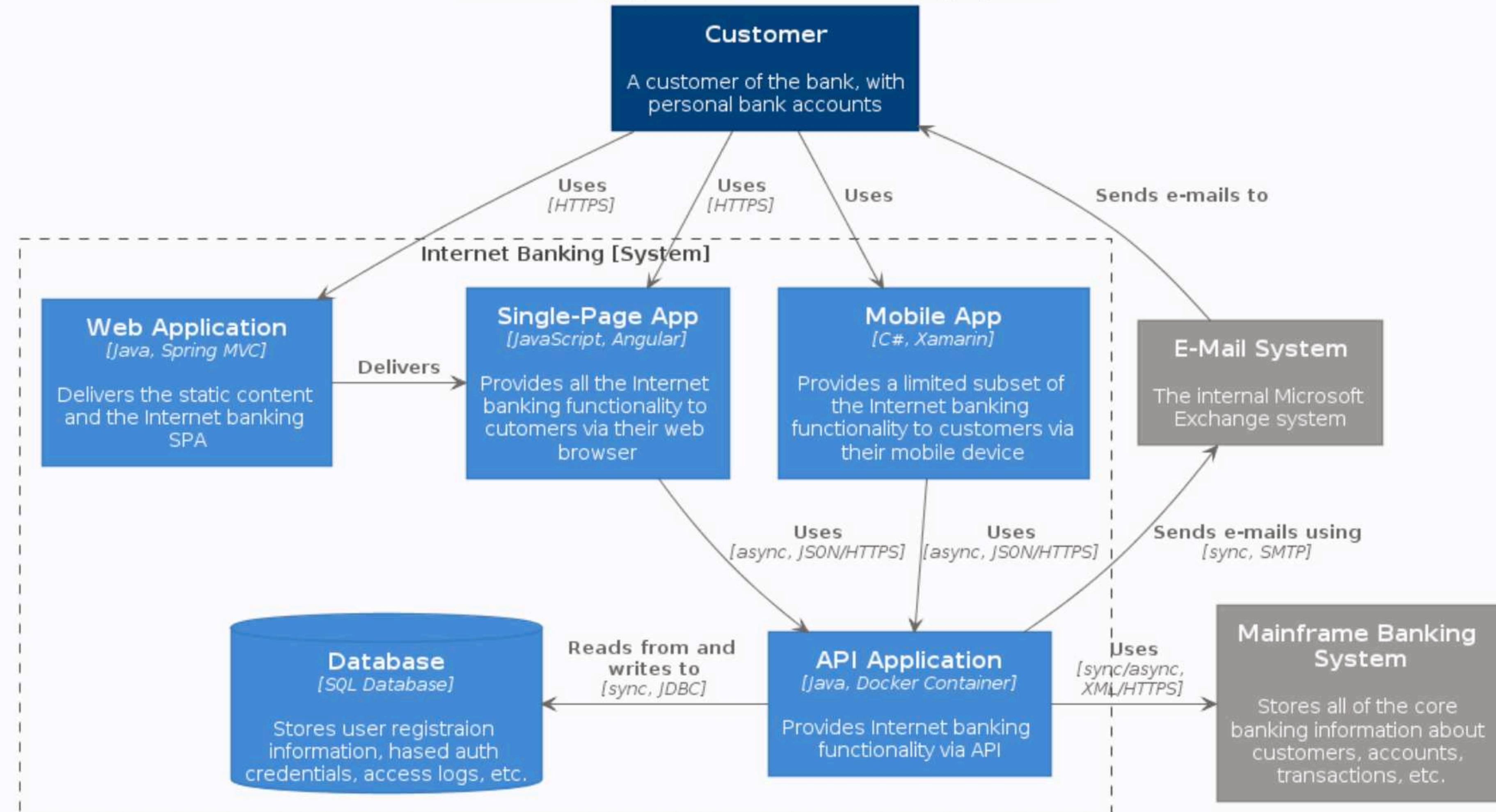
Elements

Does every element have a name?	Yes	No
Do you understand the type of every element? (i.e. the level of abstraction; e.g. software system, container, etc)	Yes	No
Do you understand what every element does?	Yes	No
Where applicable, do you understand the technology choices associated with every element?	Yes	No
Do you understand the meaning of all acronyms and abbreviations used?	Yes	No
Do you understand the meaning of all colours used?	Yes	No

What tools do you
recommend?

C4-PlantUML

Container diagram for Internet Banking System



	Type
	person
	external person
	system
	external system
	container

Home About Tour Getting Started C4 model Express Products and Pricing Examples Help

Sign up Sign in



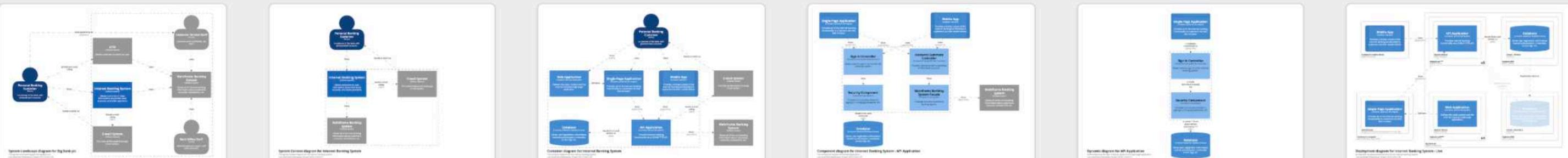
Structurizr

Your software architecture documentation hub

[Demo](#) [Sign up](#) [Cloud service](#) [On-premises installation](#)

The quickest way to create software architecture diagrams

Have you ever spent *hours* trying to make a software architecture diagram look pretty? It's 2019, and we shouldn't be using general purpose diagramming tools to create software architecture diagrams. We're supposed to be engineers, not artists! Structurizr is a lightweight, web-based modelling tool that lets you quickly create diagrams based upon the [C4 model for software architecture](#).



System Landscape diagrams System Context diagrams Container diagrams Component diagrams Dynamic diagrams Deployment diagrams

A lightweight software architecture modelling tool, specifically designed to support the C4 model; supplemented with Markdown/AsciiDoc documentation, and architecture decision records (ADRs)

Abstractions first,
notation second

Ensure that your team has a ubiquitous
language to describe software architecture

Thank you!

simon@architectis.je

@simonbrown