

**PREGRADO**



UNIDAD 1 | OVERVIEW

# APPROACHES



Al finalizar el curso, el estudiante elabora el diseño y la documentación técnica de una aplicación a partir de las especificaciones para las clases y componentes de acuerdo con una arquitectura definida, siguiendo prácticas de diseño adecuadas con patrones GoF.

# AGENDA

INTRO

UML OVERVIEW

DESIGN APPROACHES

C4 MODEL BASICS

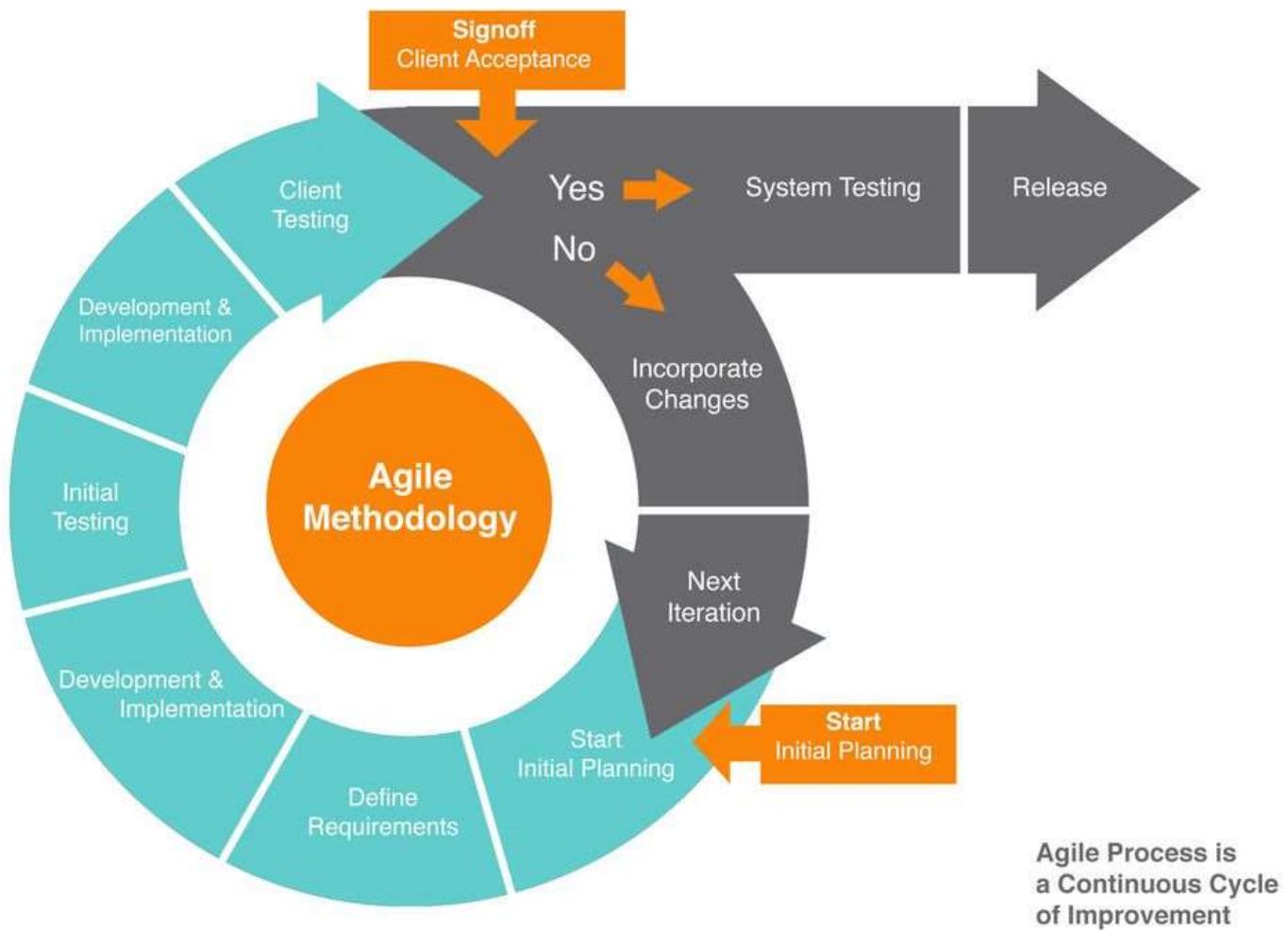
SDD INTRO



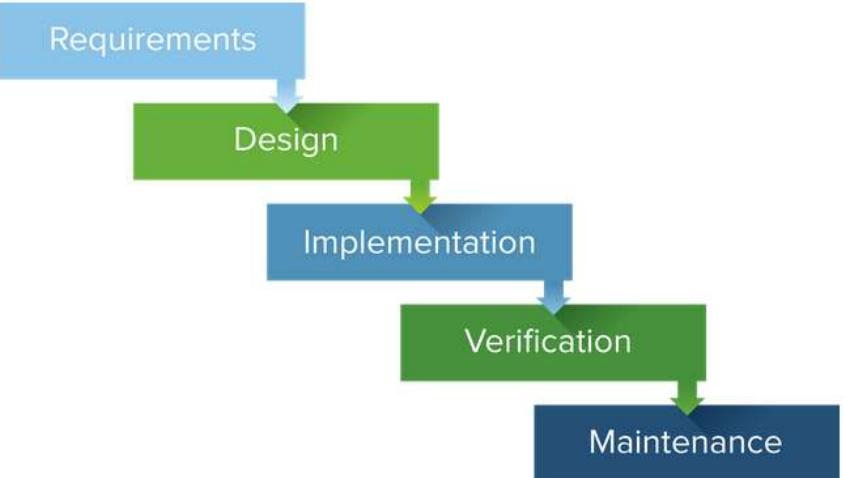
# Agile principles



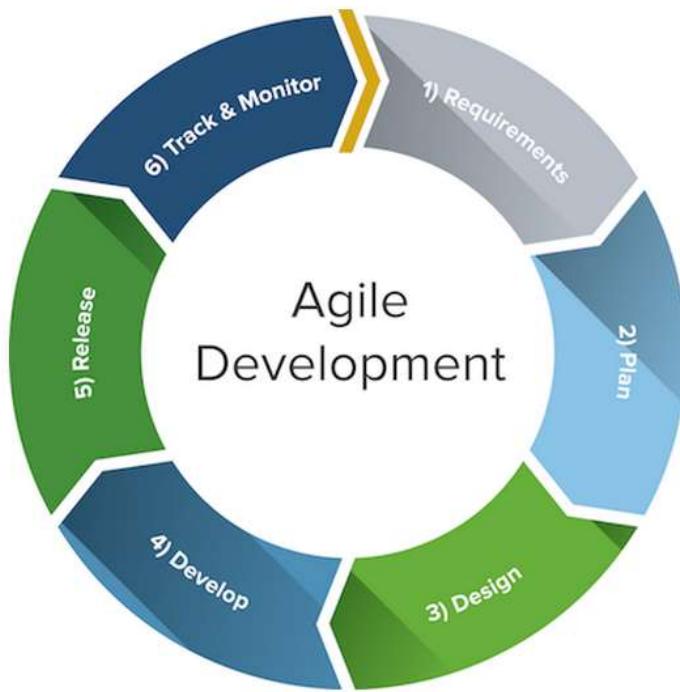
# Agile



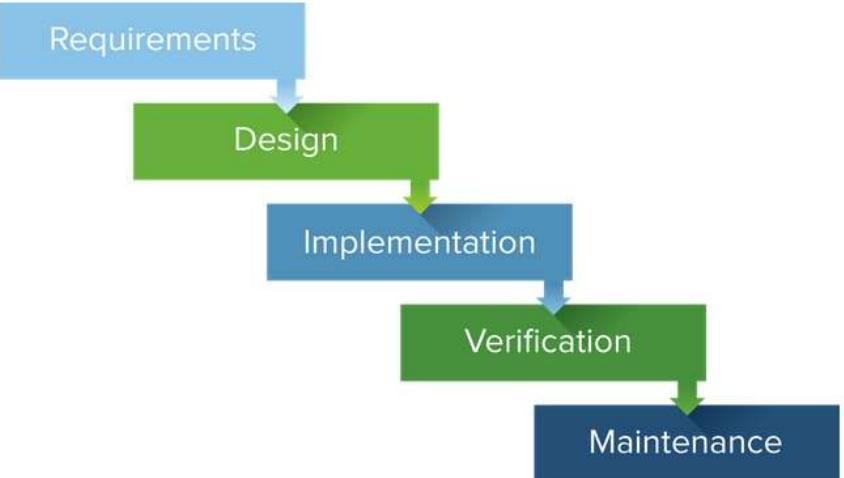
# Agile

Agile	Waterfall
<ul style="list-style-type: none"><li>• Continuous cycles</li><li>• Small, high-functioning, collaborative teams</li><li>• Multiple methodologies</li><li>• Flexible/continuous evolution</li><li>• Customer involvement</li></ul>	 <pre>graph TD; A[Requirements] --&gt; B[Design]; B --&gt; C[Implementation]; C --&gt; D[Verification]; D --&gt; E[Maintenance]</pre> <ul style="list-style-type: none"><li>• Sequential/linear stages</li><li>• Upfront planning and in-depth documentation</li><li>• Contract negotiation</li><li>• Best for simple, unchanging projects</li><li>• Close project manager involvement</li></ul>

## Agile

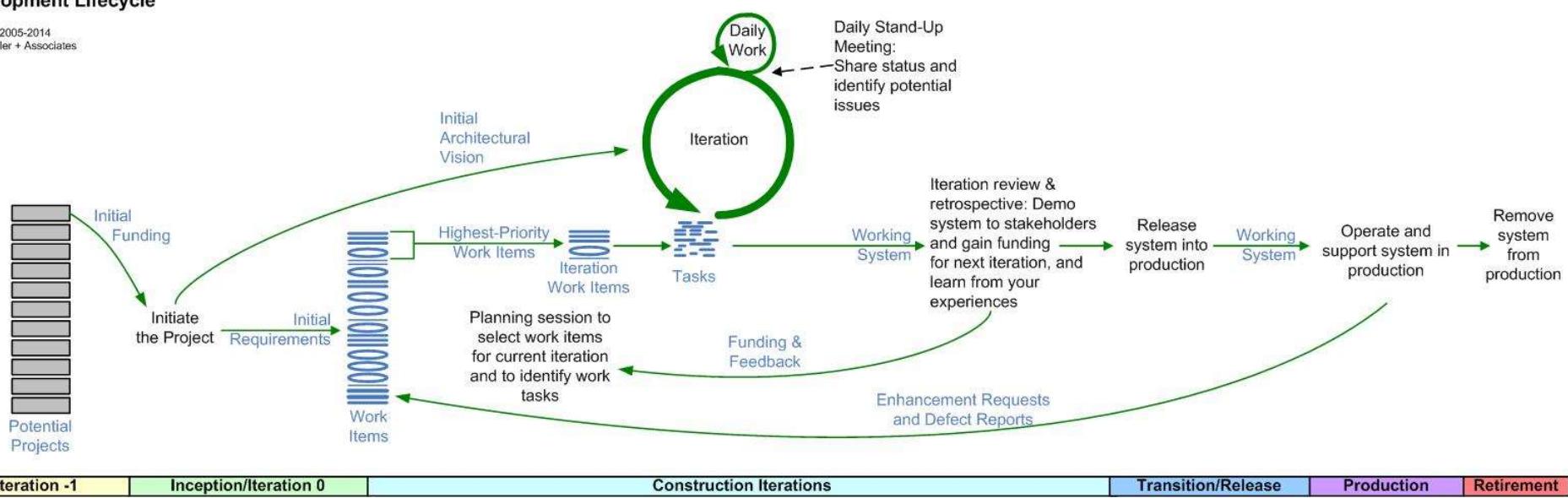


## Waterfall



## Agile System Development Lifecycle

Copyright 2005-2014  
Scott Ambler + Associates



# AGENDA

INTRO

UML OVERVIEW

DESIGN APPROACHES

C4 MODEL BASICS

SDD INTRO



# UML

## Unified Modeling Language (Lenguaje unificado de modelado)



# UML Diagrams

## 14 diagramas, organizados en buckets

### Structure diagrams

- Class diagrams
- Object diagrams
- Package diagrams
- Composite structure diagrams
- Component diagrams
- Deployment diagrams
- Profile diagrams

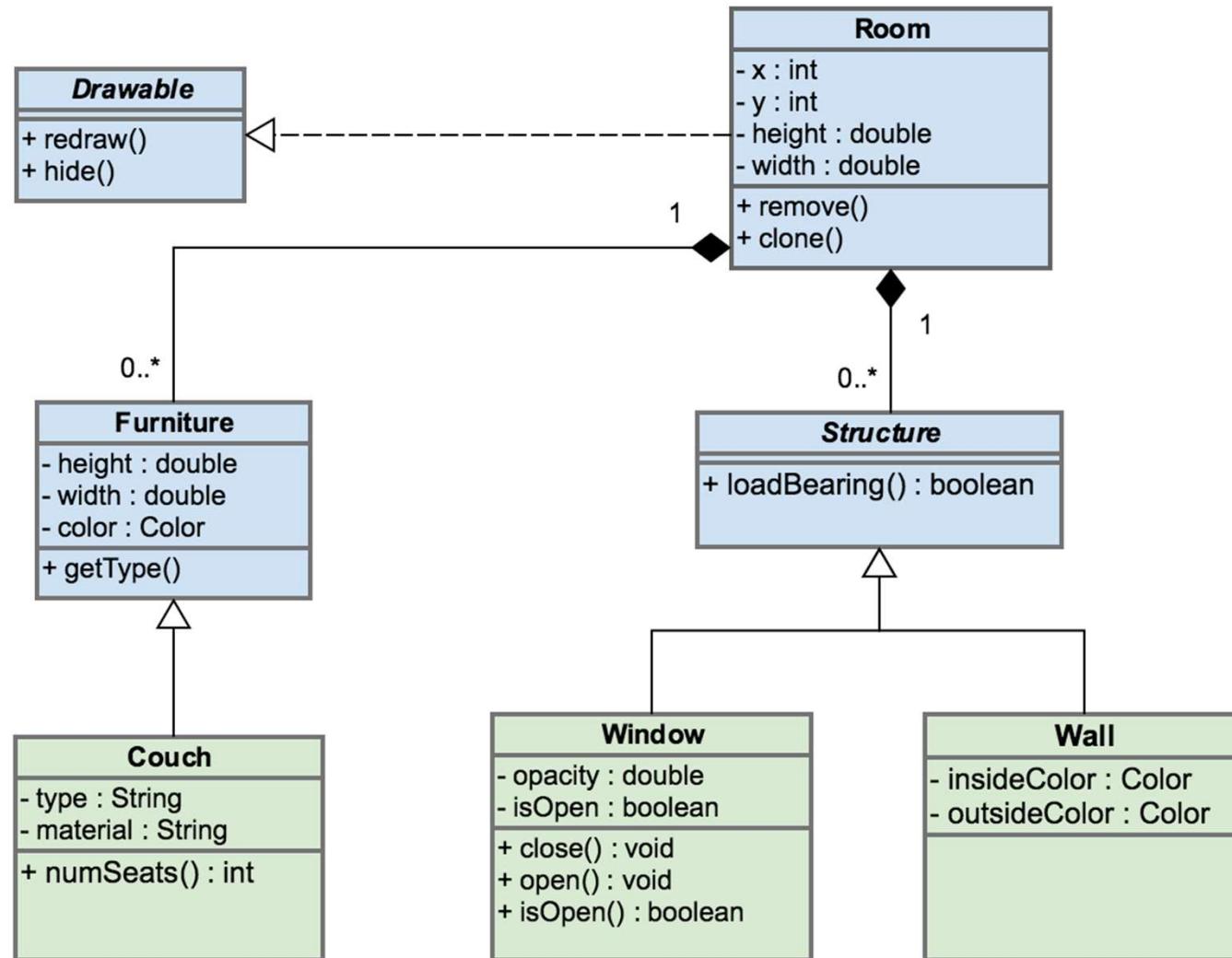
### Behavior diagrams

- Use case diagrams
- Activity diagrams
- State machine diagrams

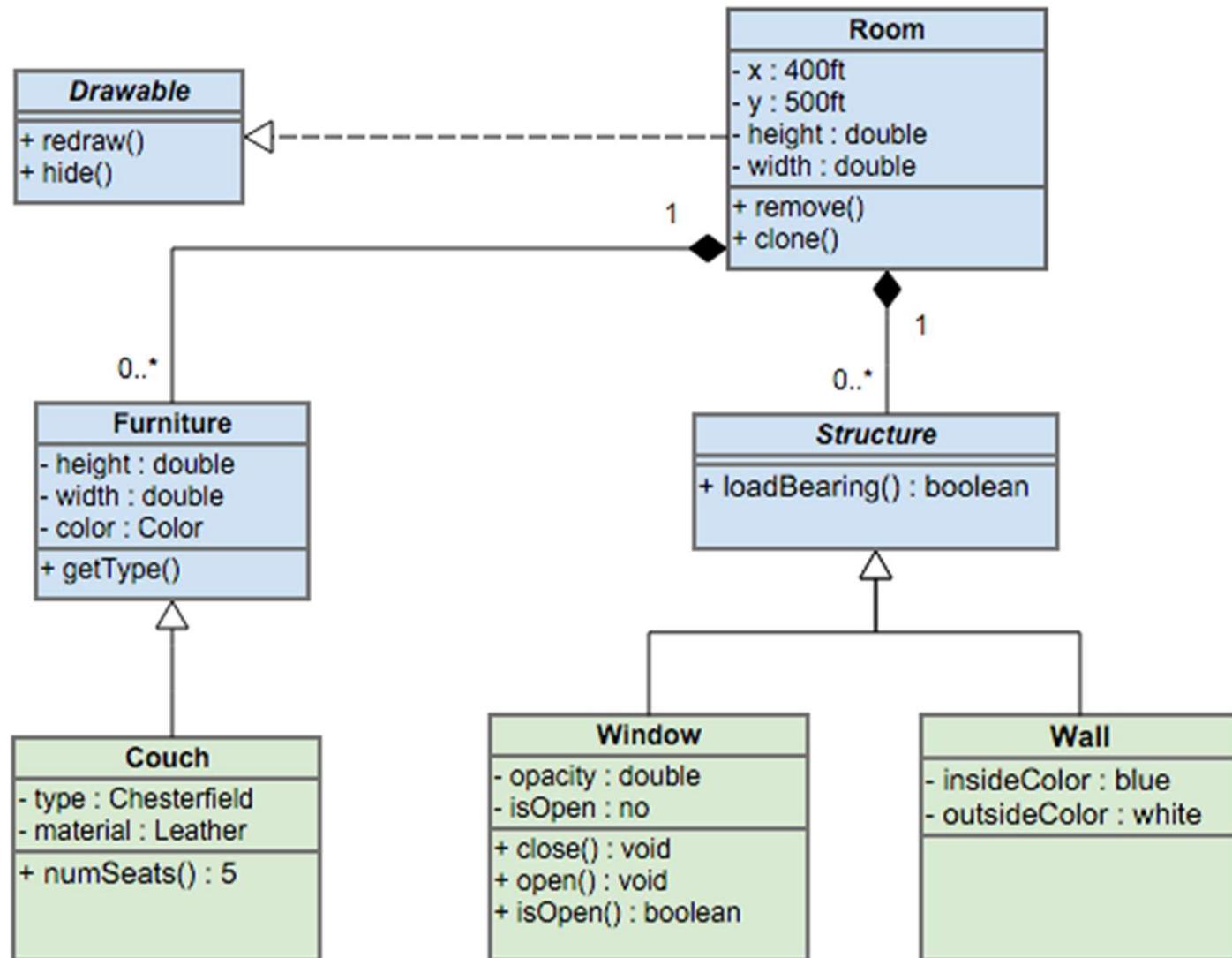
### Interaction diagrams

- Sequence diagrams
- Communication diagrams
- Timing diagrams
- Interaction overview diagrams

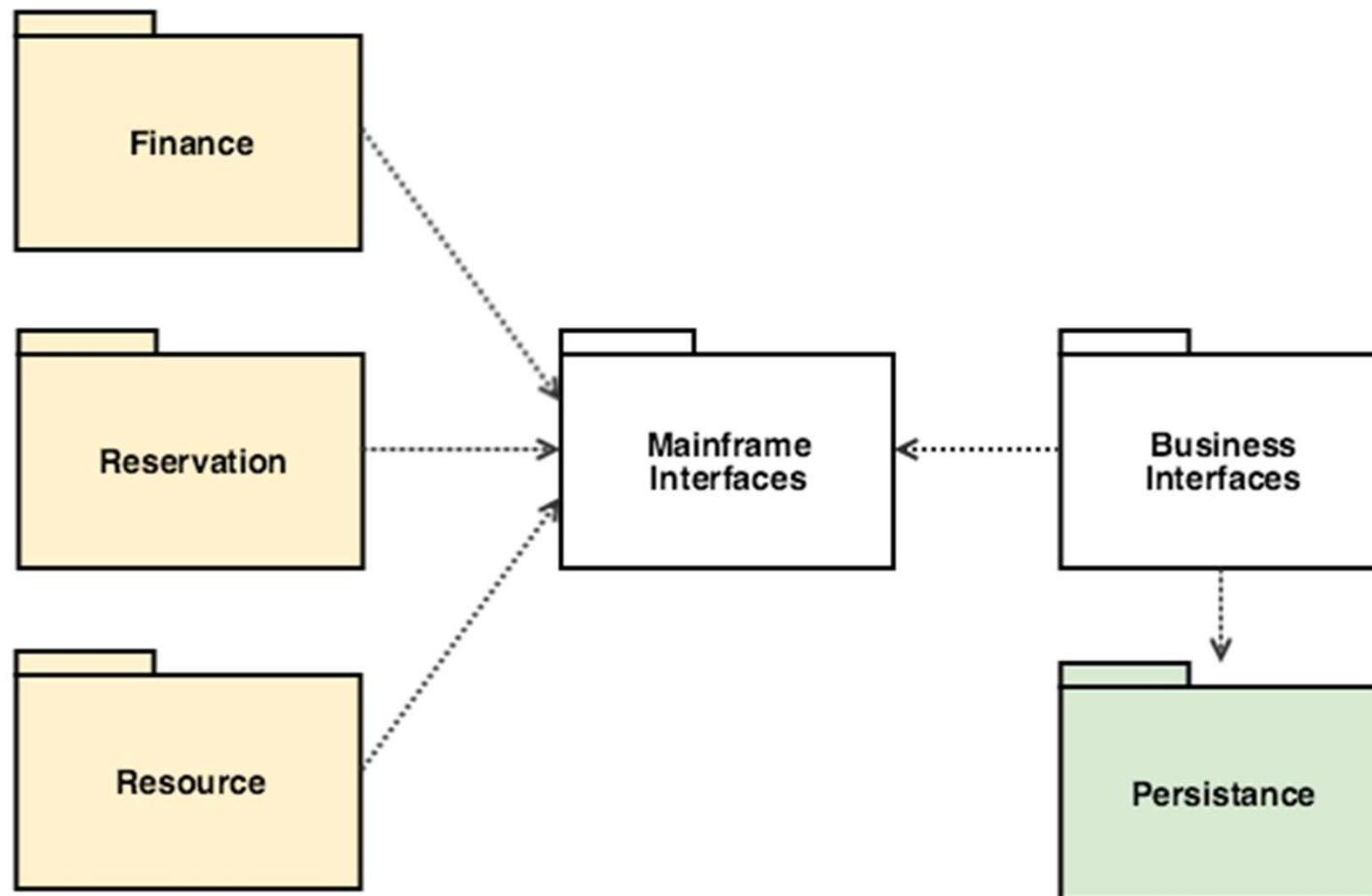
# Class diagrams



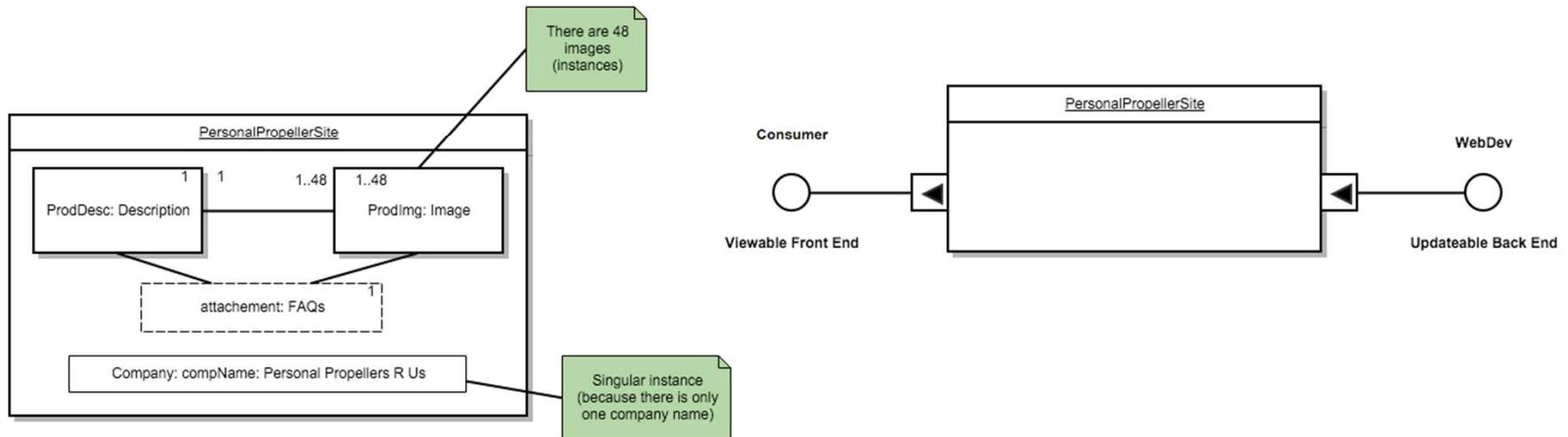
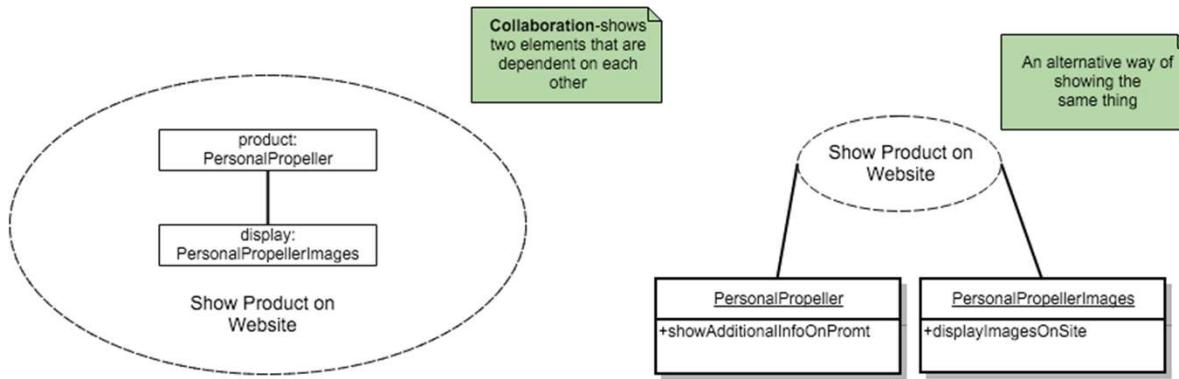
# Object diagrams



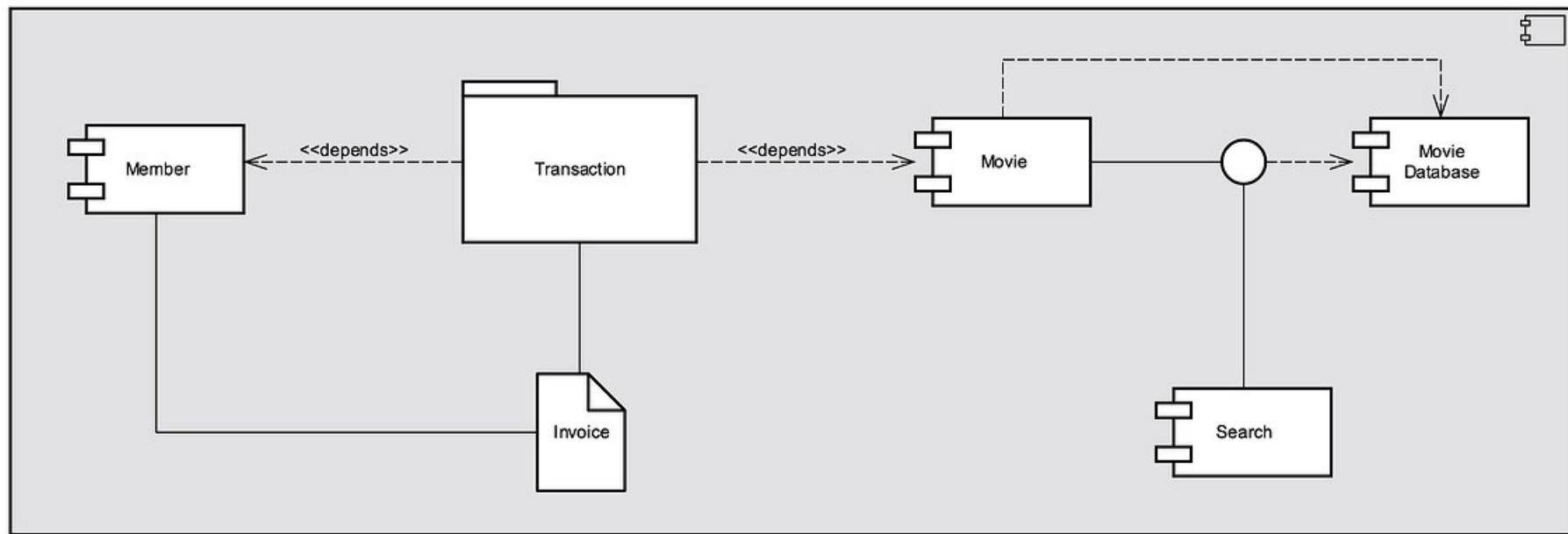
# Package diagrams



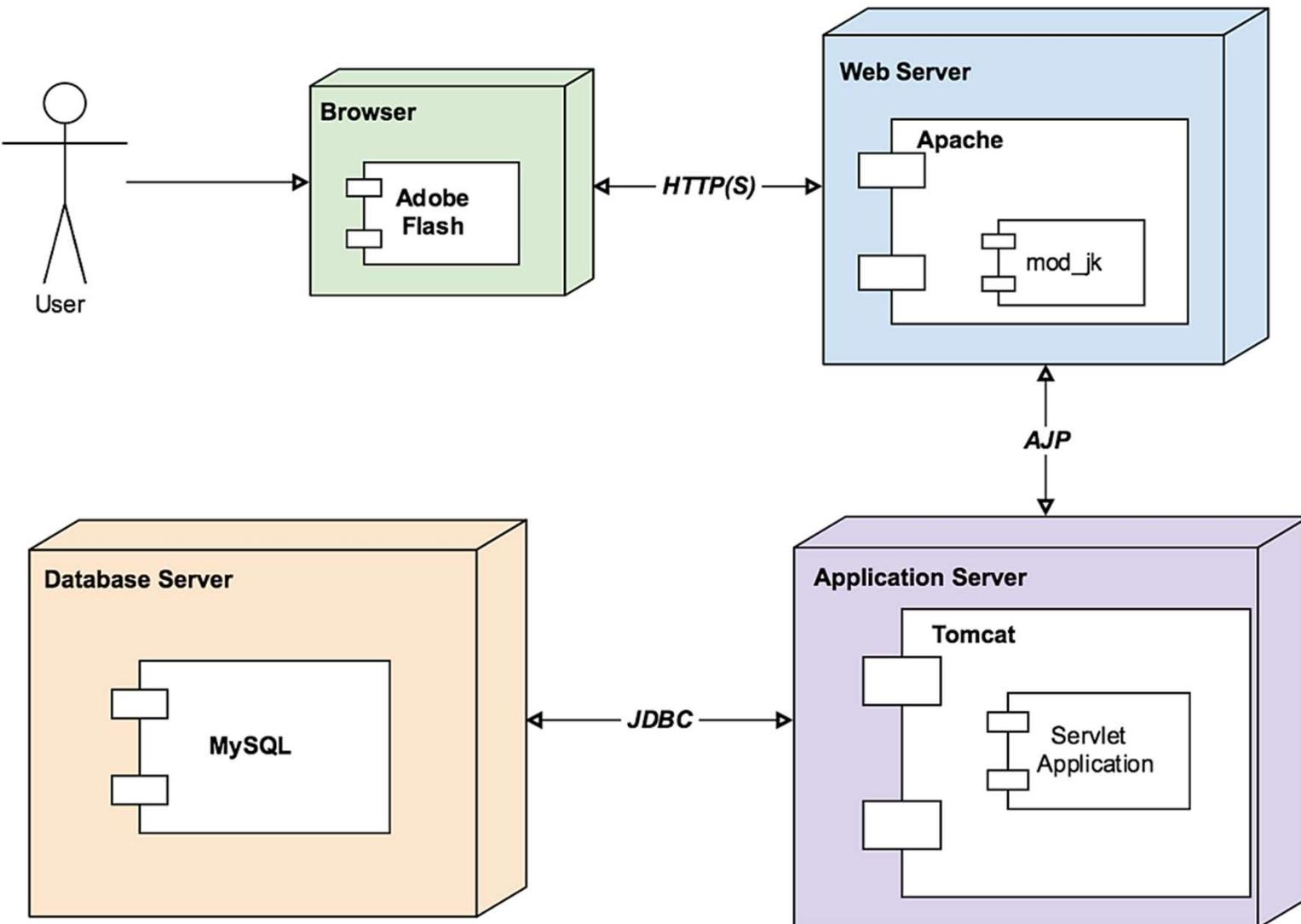
# Composite structure diagrams



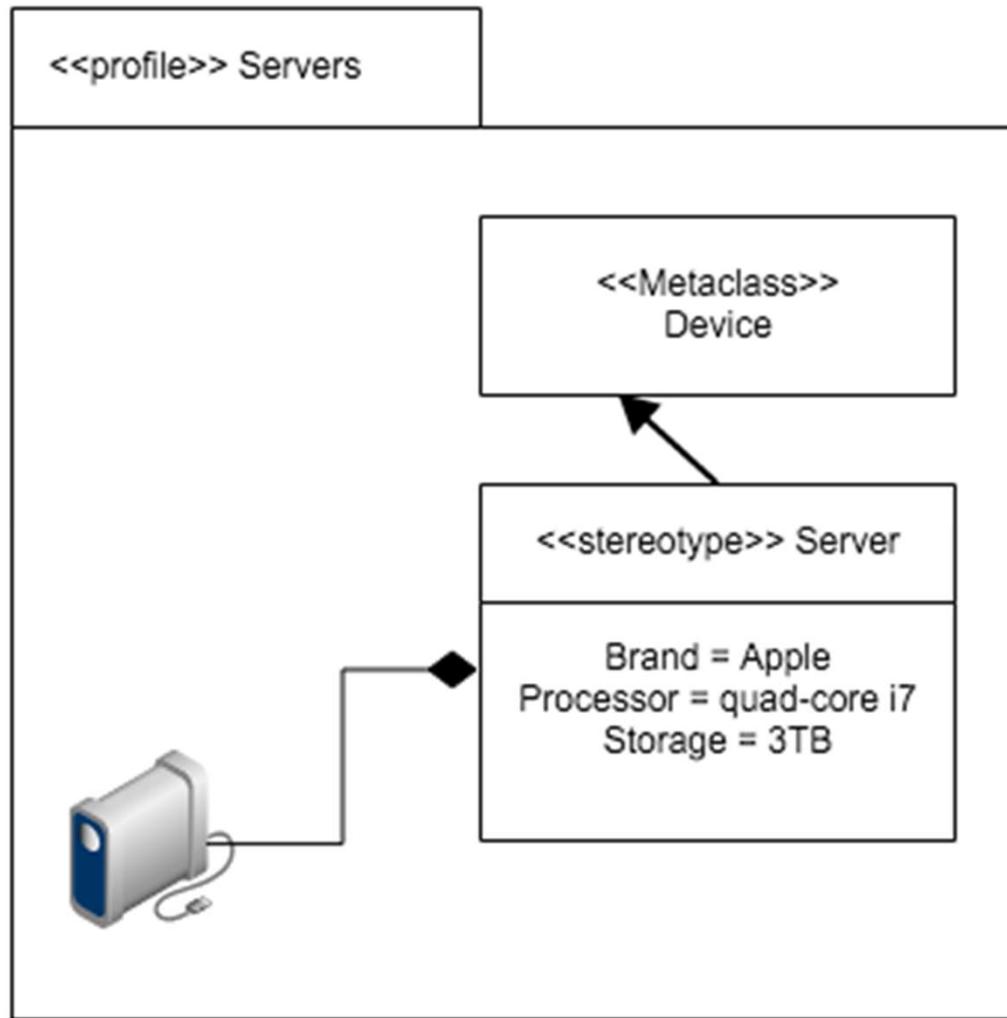
# Component diagrams



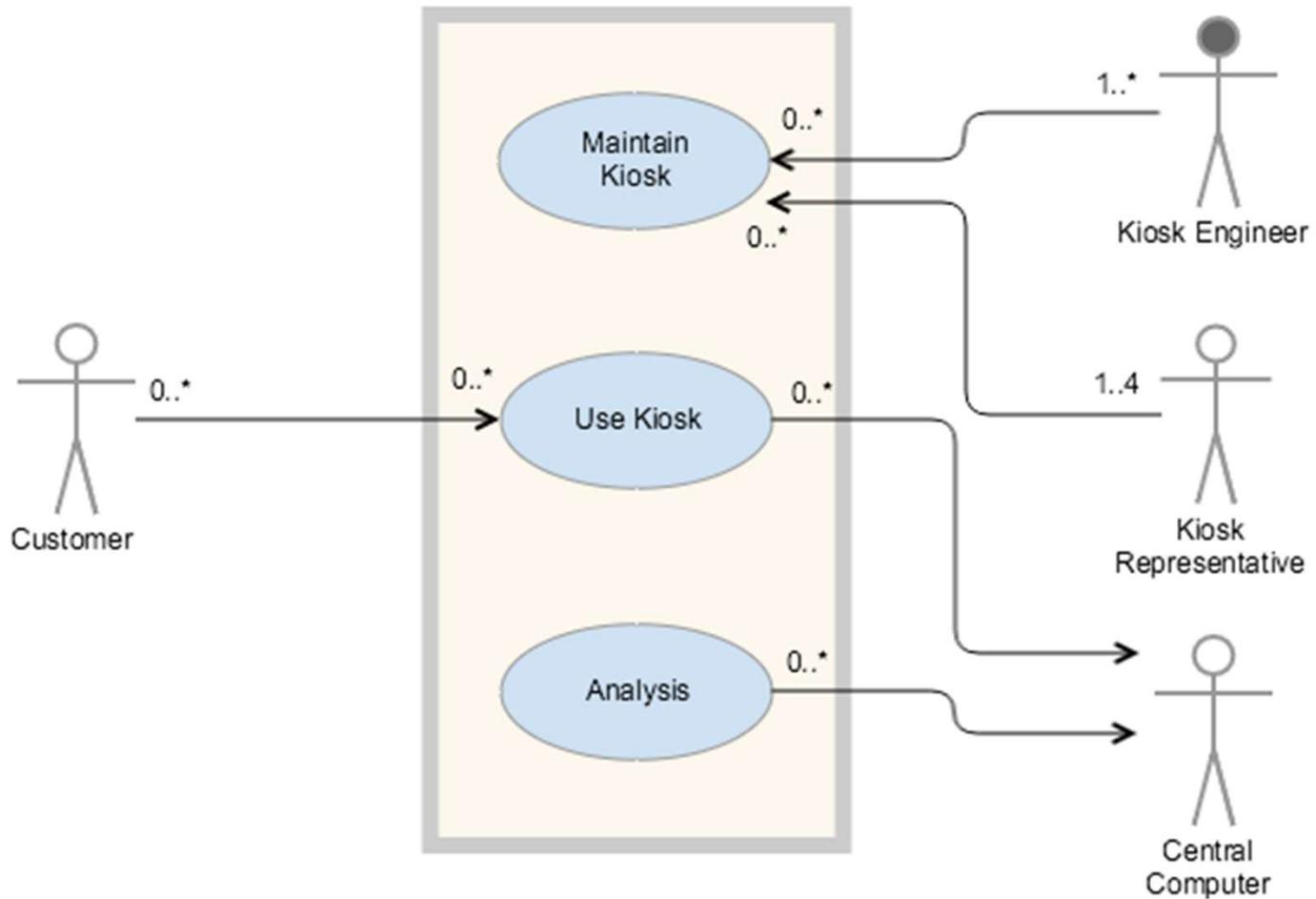
# Deployment diagrams



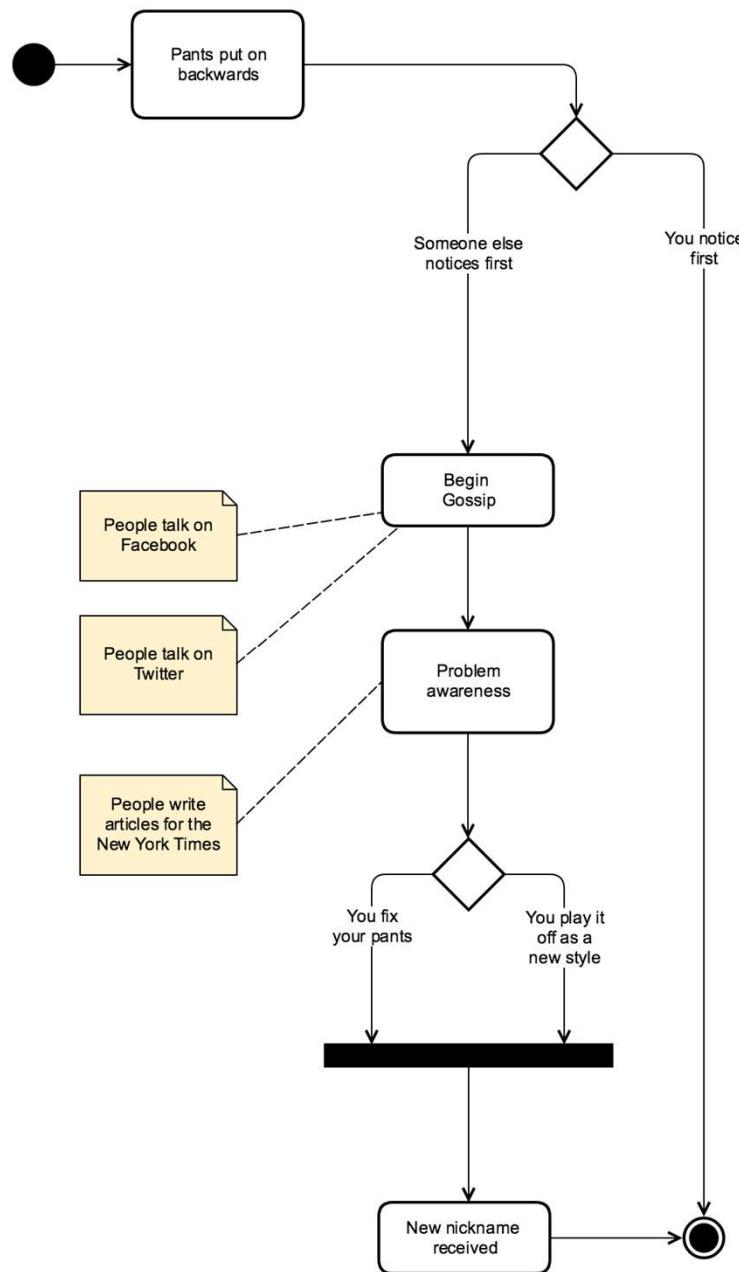
# Profile diagrams



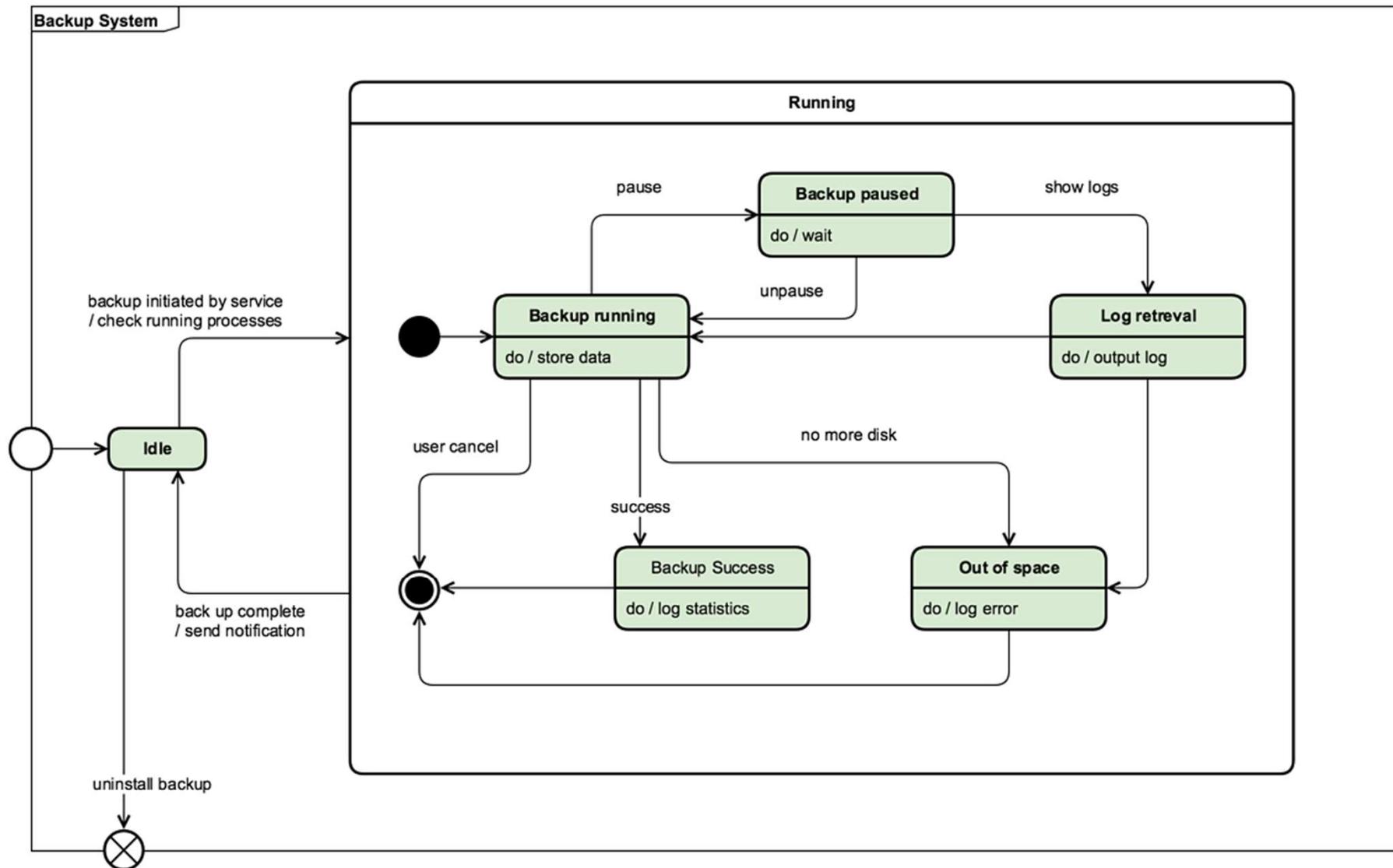
# Use case diagrams



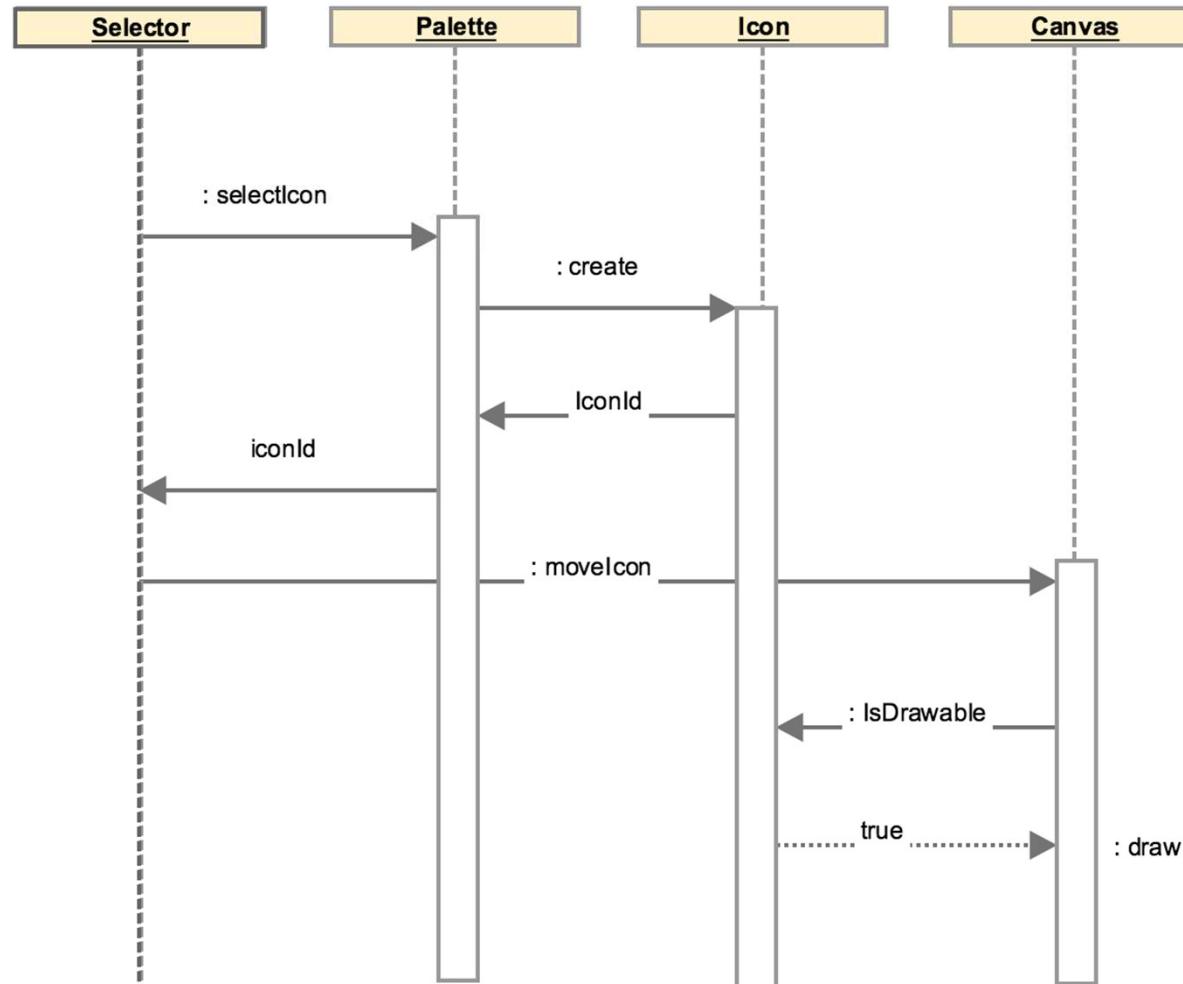
# Activity diagrams



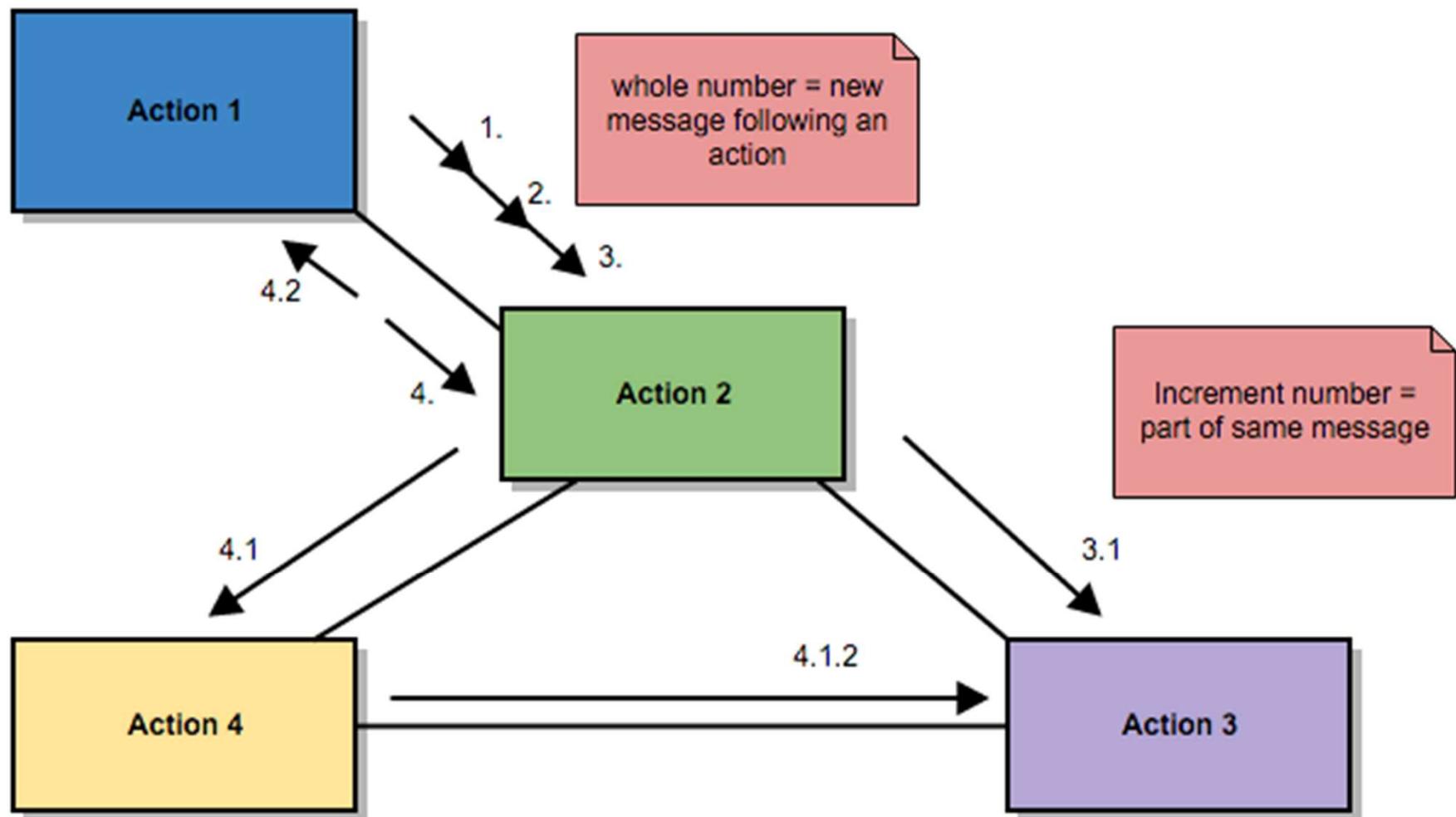
# State machine diagrams



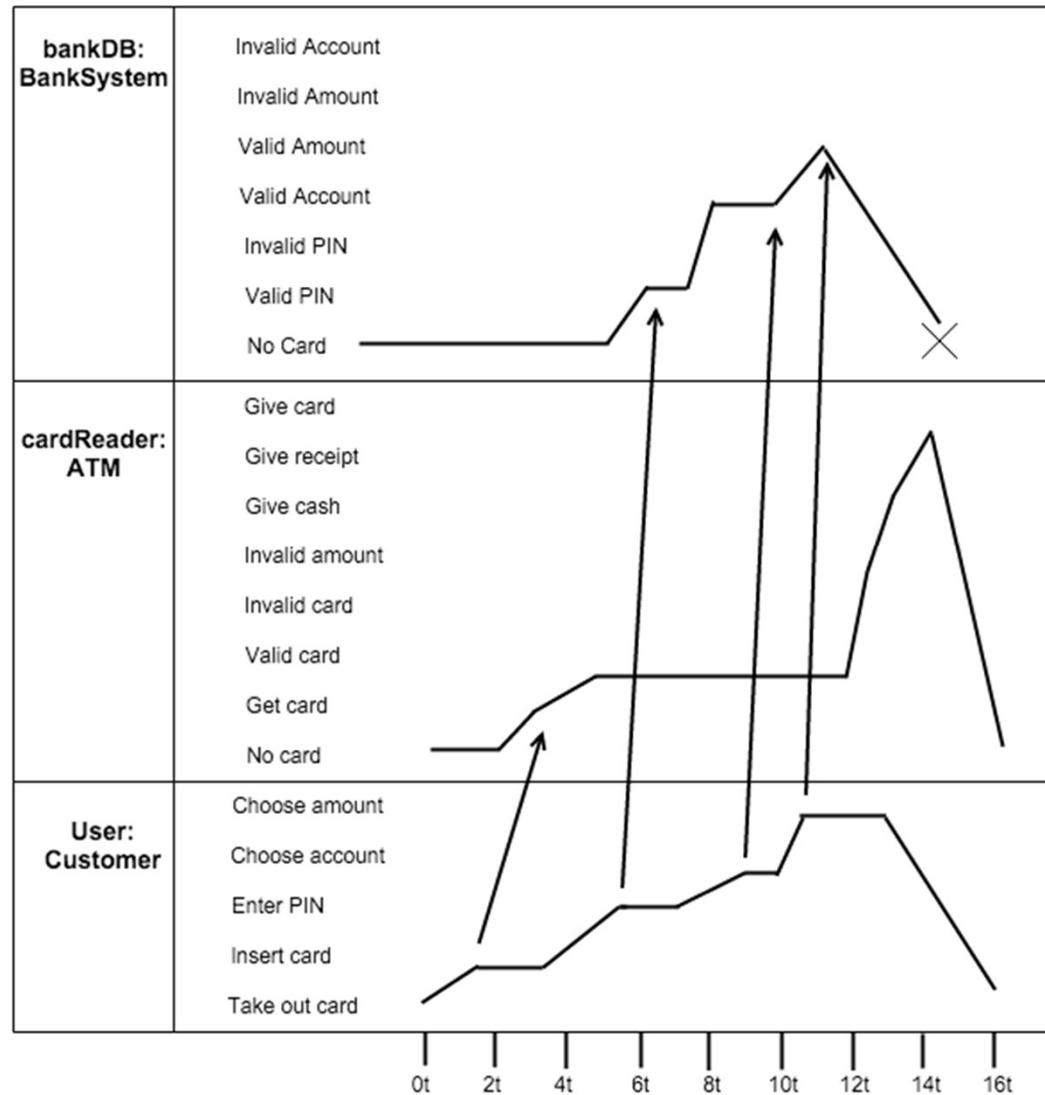
# Sequence diagrams



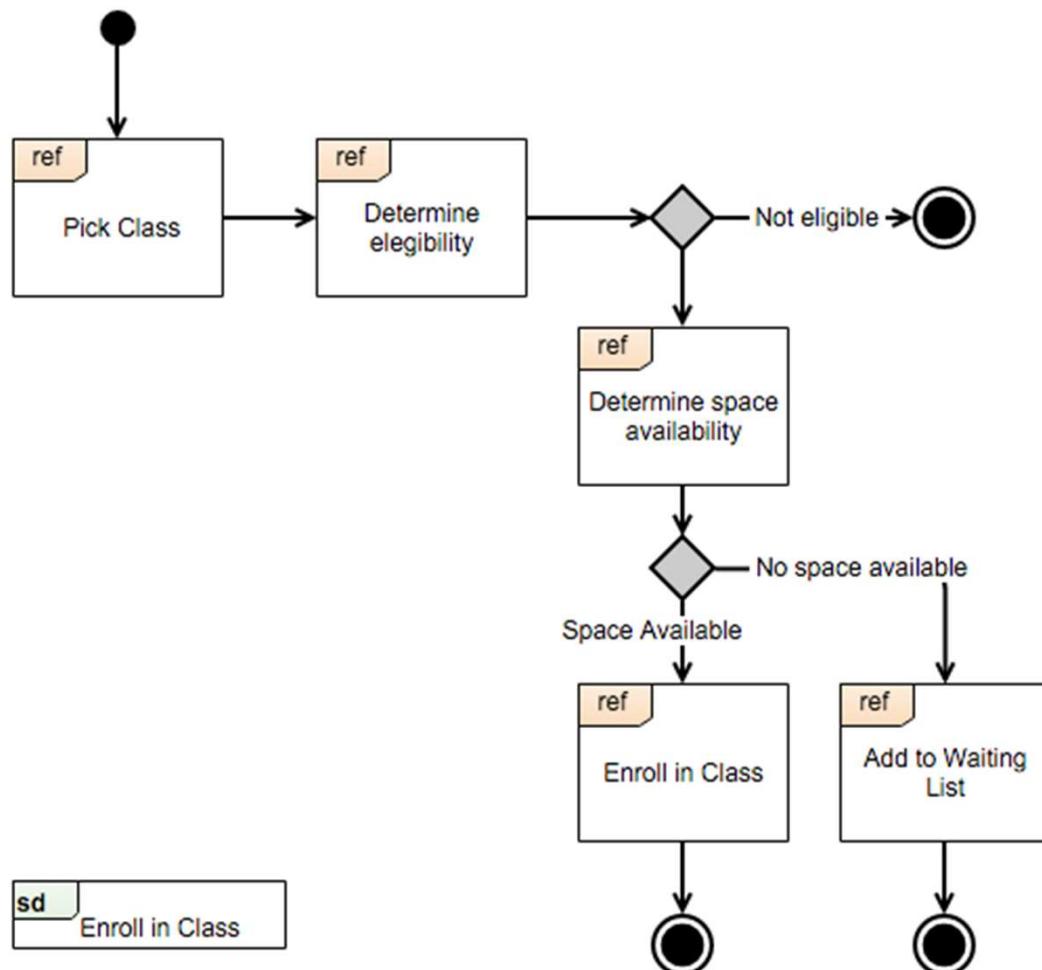
# Communication diagrams



# Timing diagrams



# Interaction overview diagrams



# AGENDA

INTRO

UML OVERVIEW

DESIGN APPROACHES

C4 MODEL BASICS

SDD INTRO



## **Planned Design**

Desde los 70s.  
Diseñadores buscan establecer la foto completa de antemano.  
Luego lo entregan a otro grupo para implementar.

### **PROS/CONS**

## **Evolutionary Design**

Concepto derivado de XP.  
Prácticas como Testing, Refactoring, Continous Integration buscan reducir curva de cambios.  
Arquitectura y Diseño van evolucionando en cada iteración.

### **PROS/CONS**

## Planned Design

Desde los 70s.  
Diseñadores buscan establecer la foto completa de antemano.  
Luego lo entregan a otro grupo para implementar.

**PROS/CONS**

**Reto**

**Encontrar un balance**

## Evolutionary Design

Concepto derivado de XP.  
Prácticas como Testing, Refactoring, Continous Integration buscan reducir curva de cambios.  
Arquitectura y Diseño van evolucionando en cada iteración.

**PROS/CONS**

# AGENDA

INTRO

UML OVERVIEW

DESIGN APPROACHES

C4 MODEL BASICS

SDD INTRO



# The C4 model



## System Context

The system plus users and system dependencies



## Containers

The overall shape of the architecture and technology choices



## Components

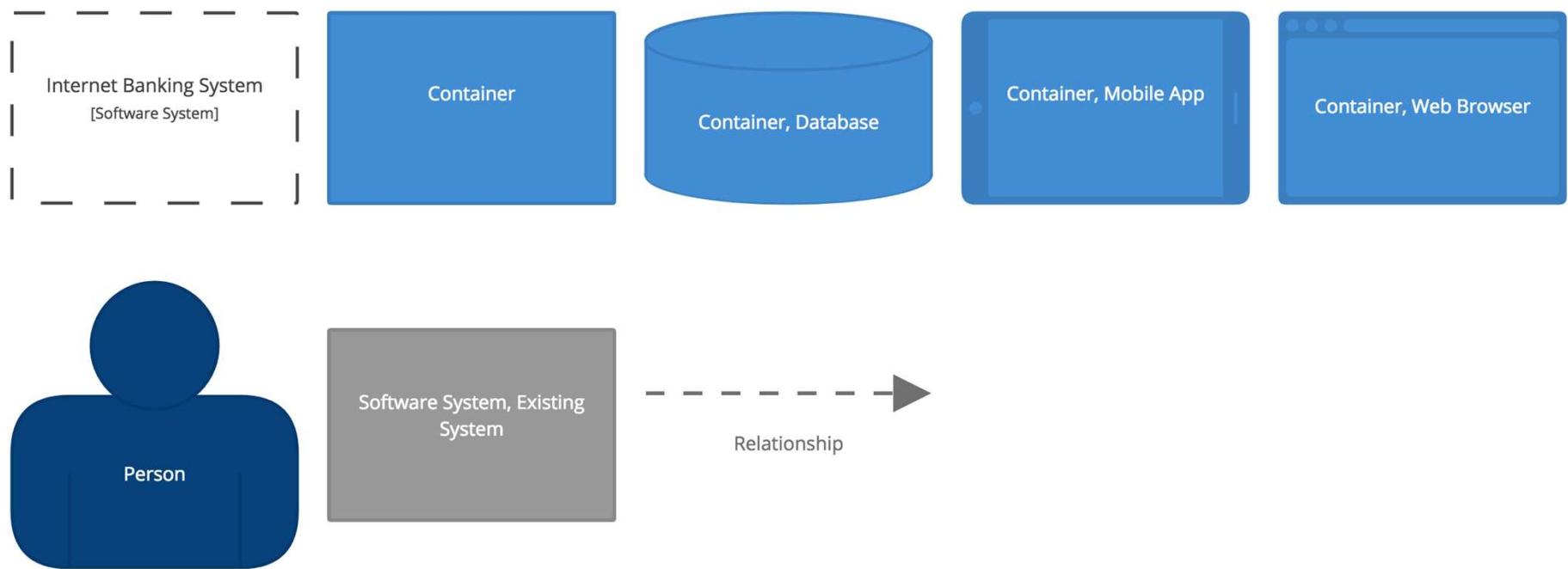
Logical components and their interactions within a container



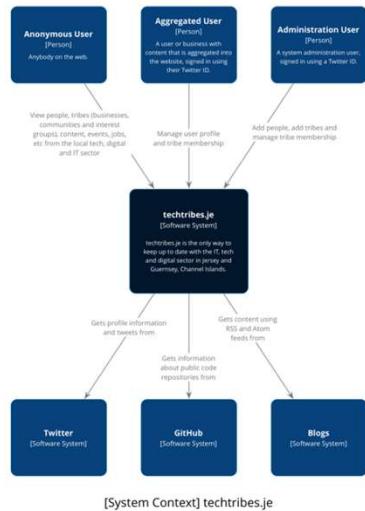
## Classes

Component or pattern implementation details

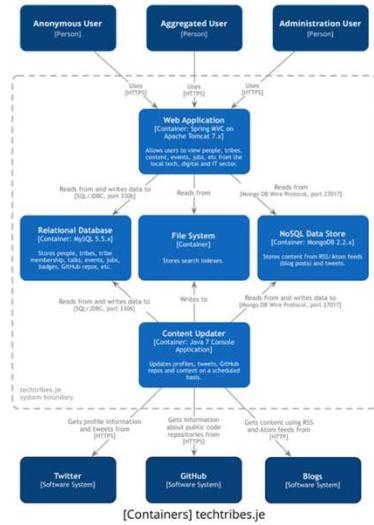
# C4 notations



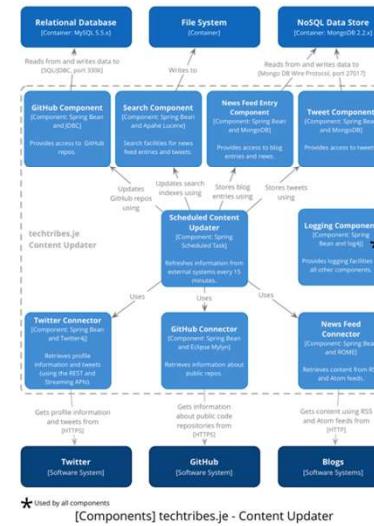
# C4 levels



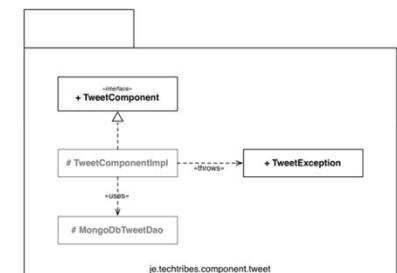
## Level 1 Context



## Level 2 Containers

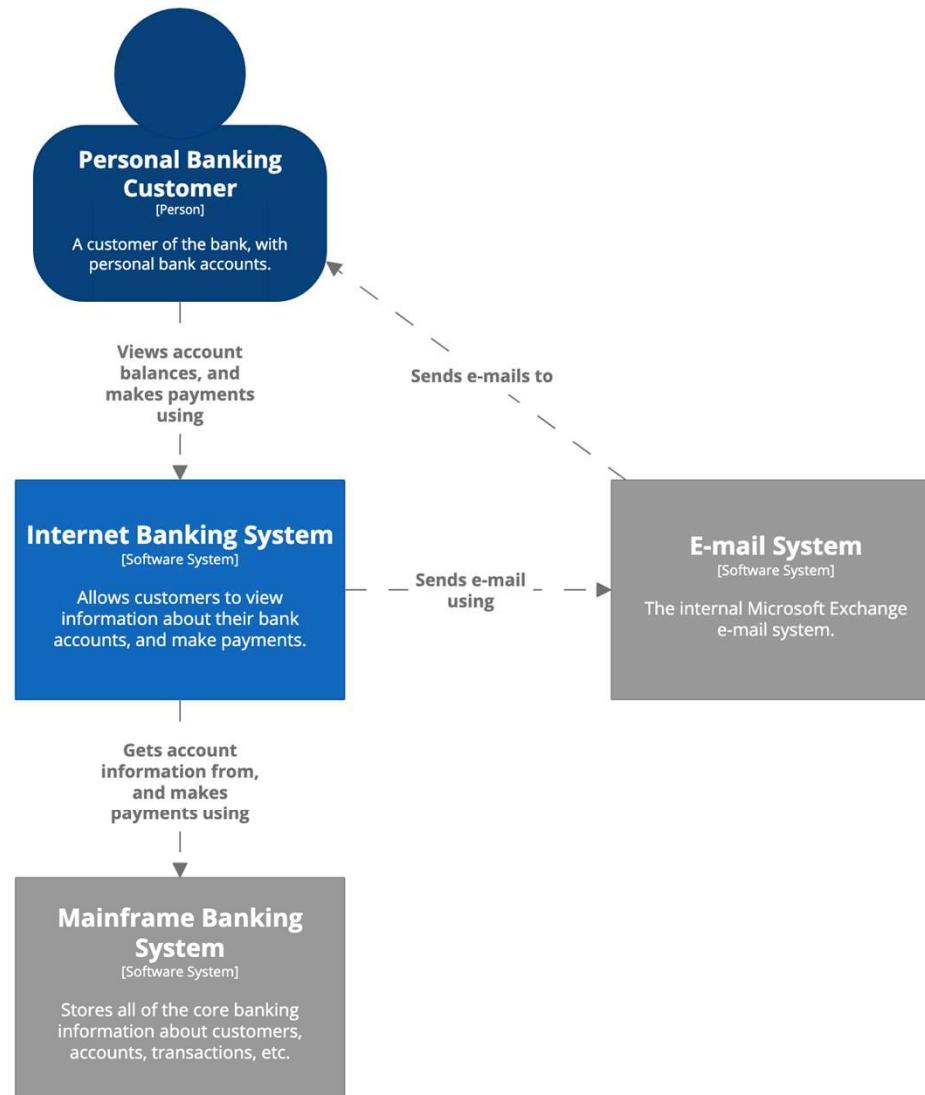


## Level 3 Components



## Level 4 Classes (or Code)

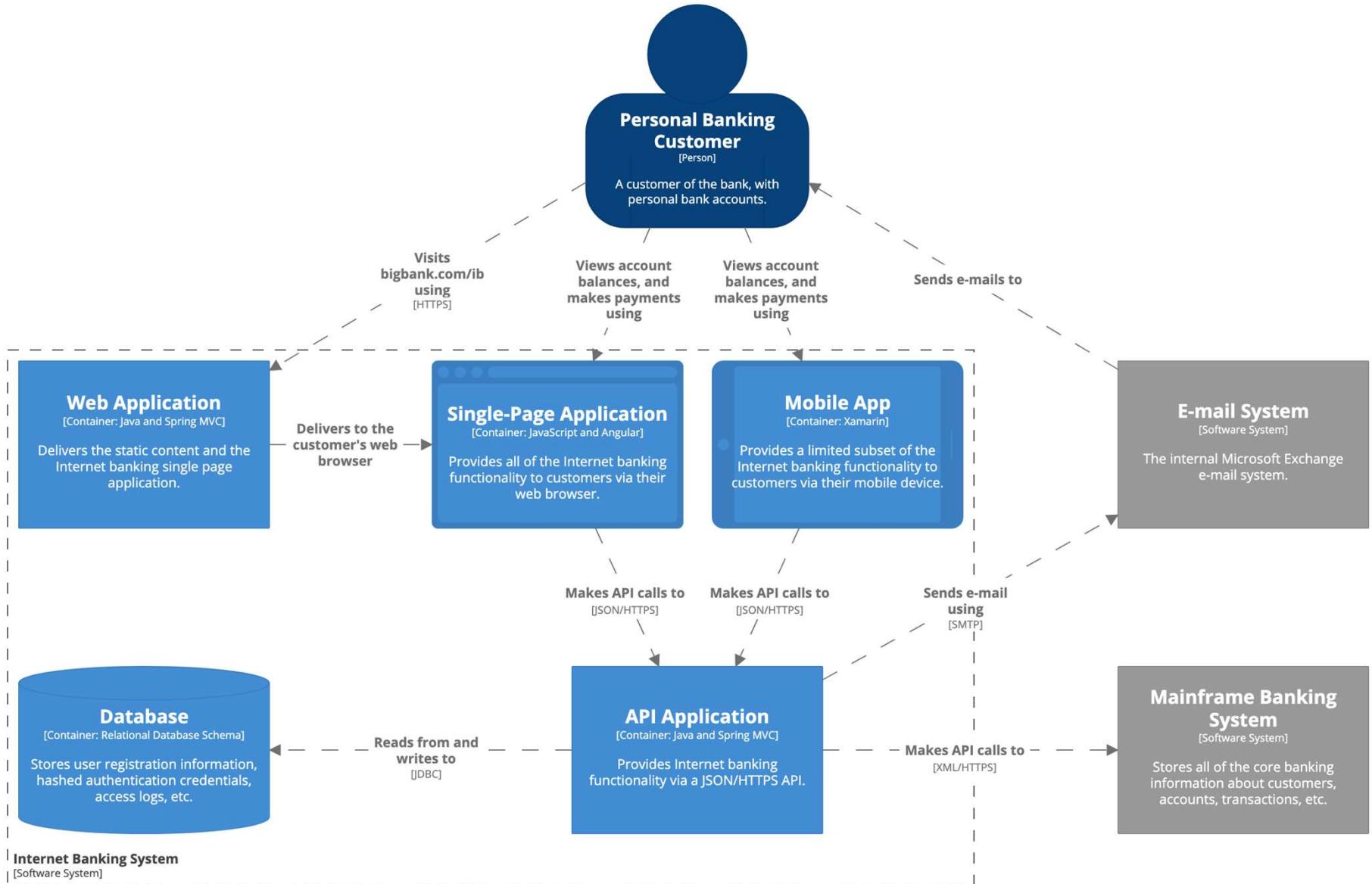
# System Context diagram



## 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)

# Container diagram

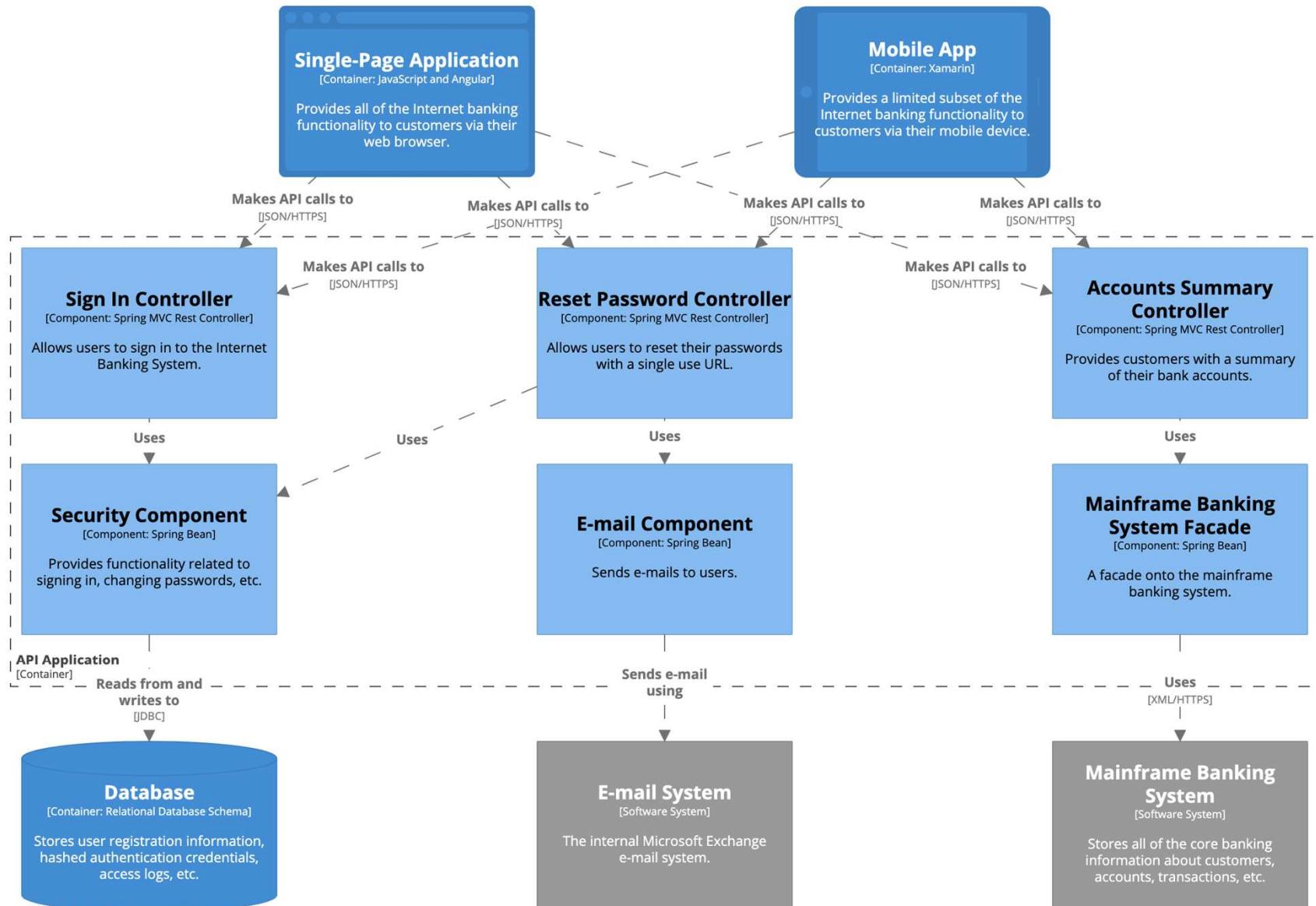


**Container diagram for Internet Banking System**

The container diagram for the Internet Banking System.

Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

# Component diagram

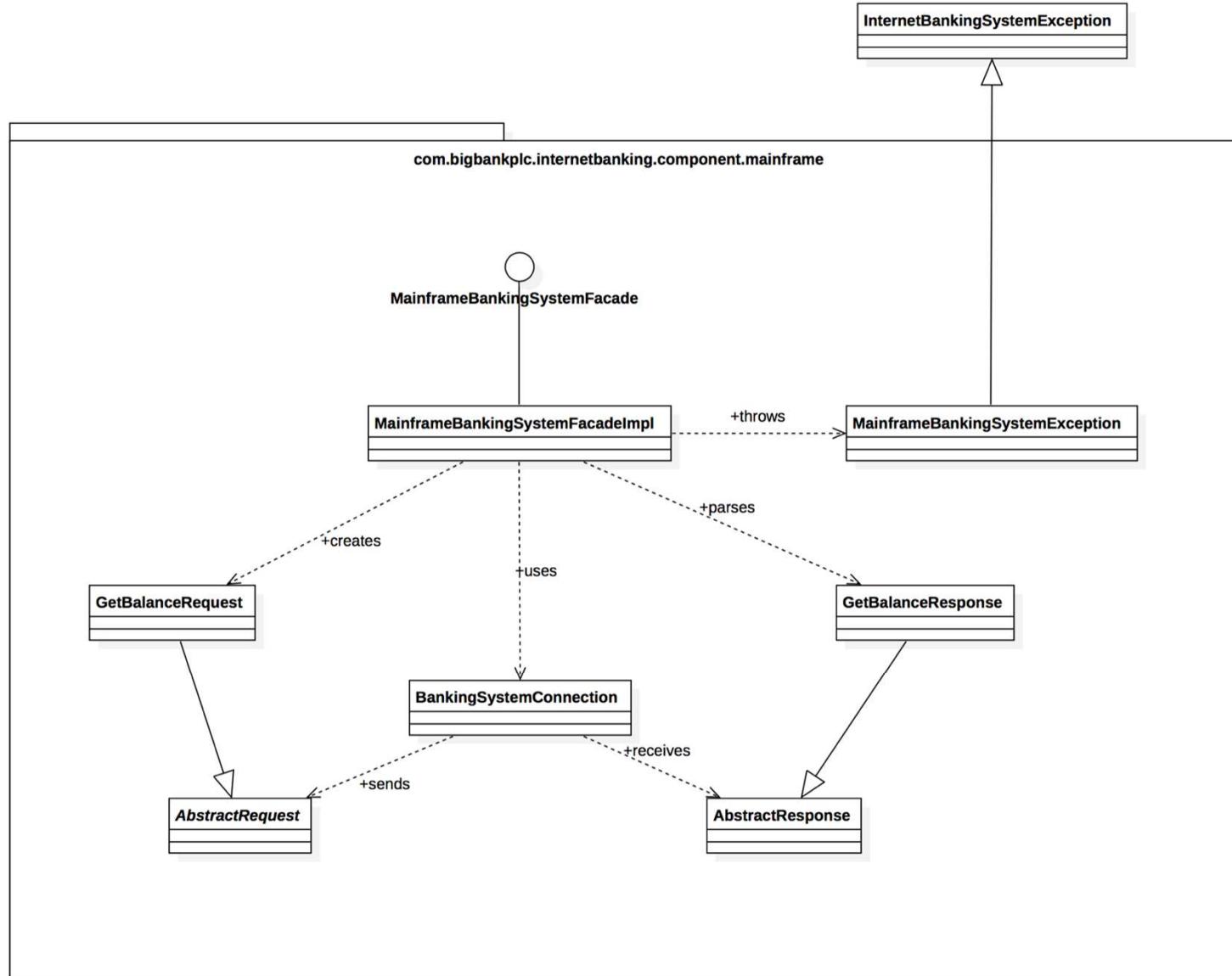


**Component diagram for Internet Banking System - API Application**

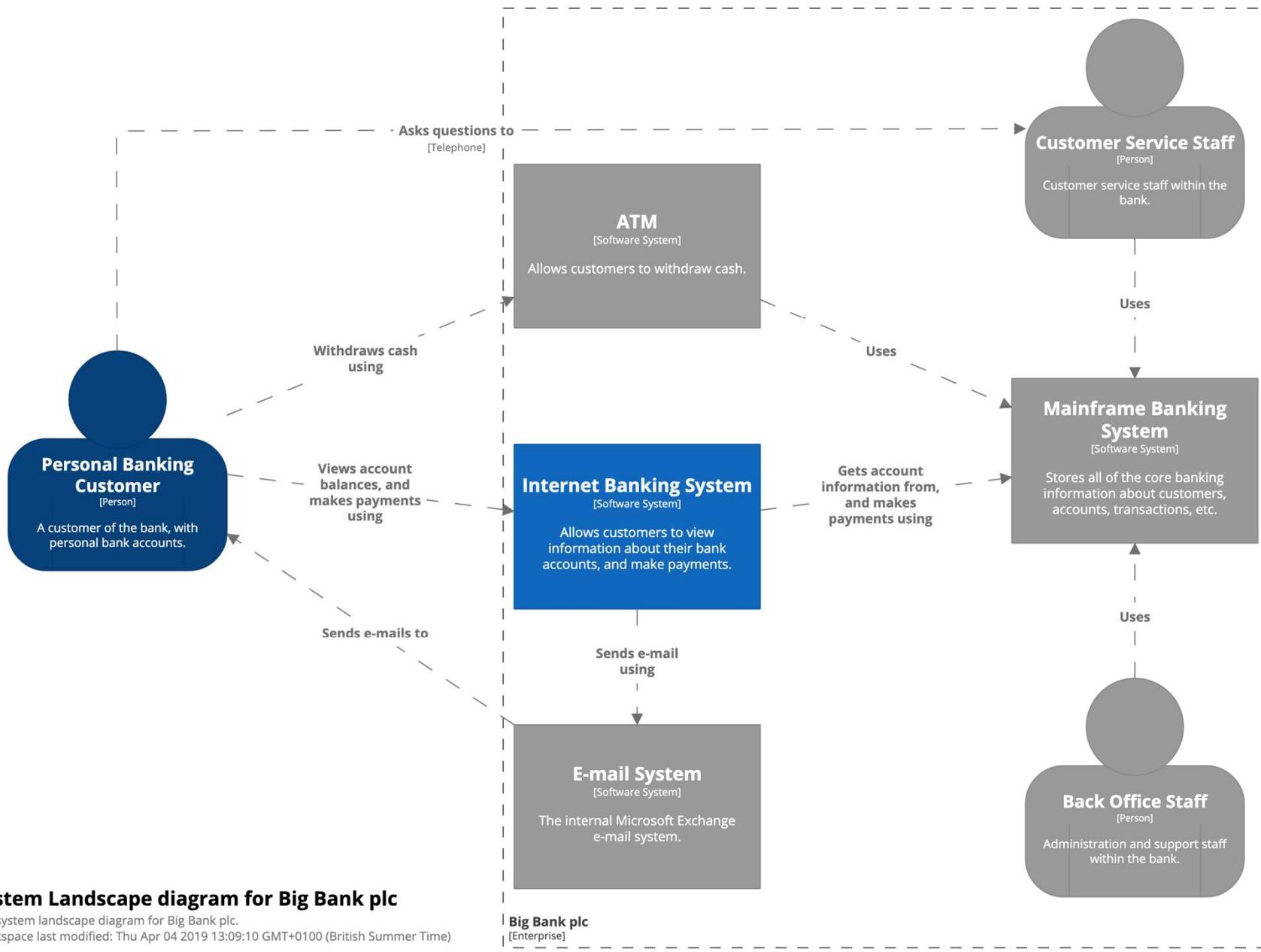
The component diagram for the API Application.

Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

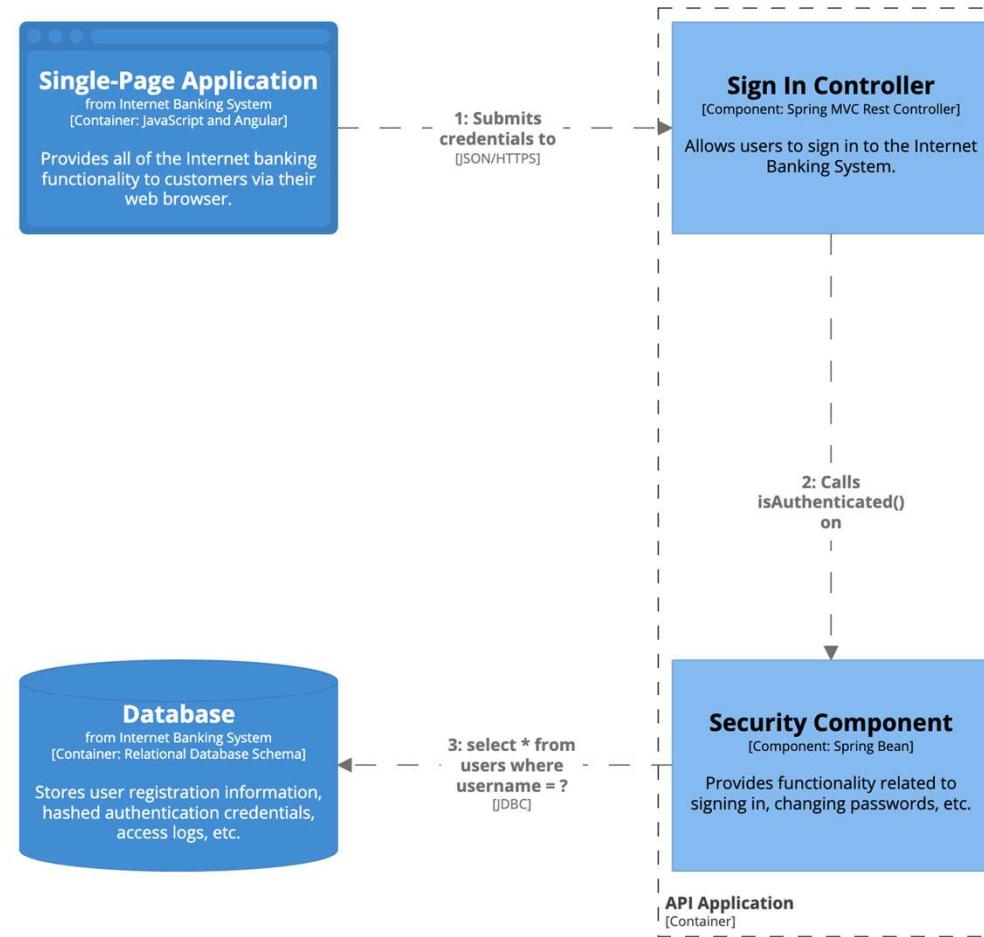
# Class/Code level diagram



# System Landscape diagram



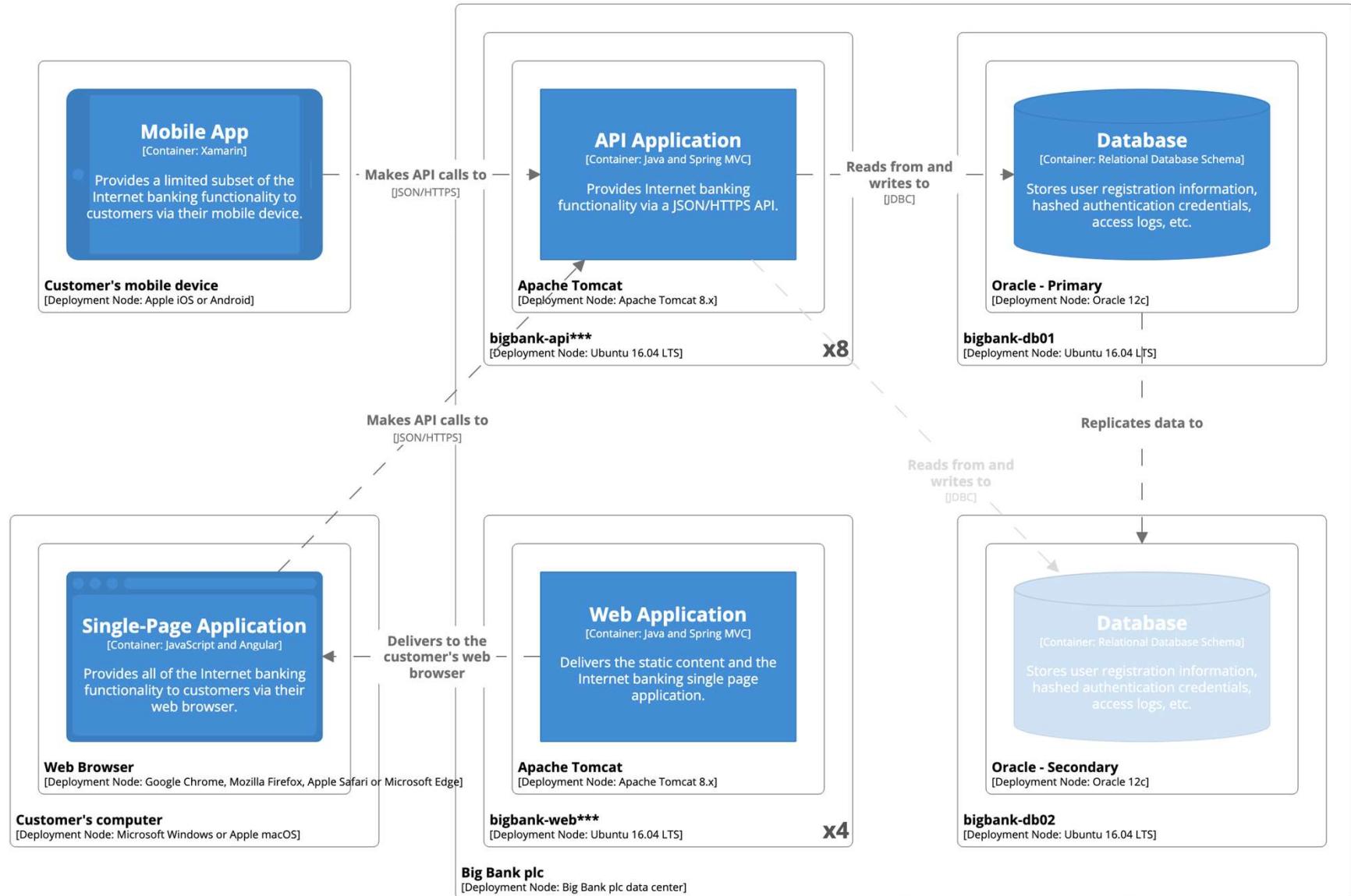
# Dynamic diagram



## Dynamic diagram for API Application

Summarises how the sign in feature works in the single-page application.  
Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

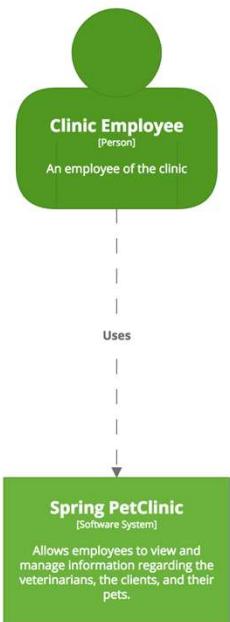
# Deployment diagram



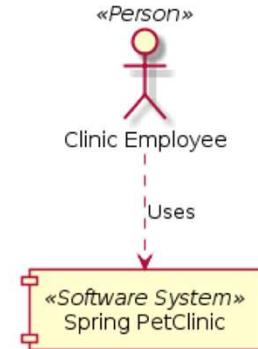
**Deployment diagram for Internet Banking System - Live**

An example live deployment scenario for the Internet Banking System.  
Workspace last modified: Thu Apr 04 2019 13:09:10 GMT+0100 (British Summer Time)

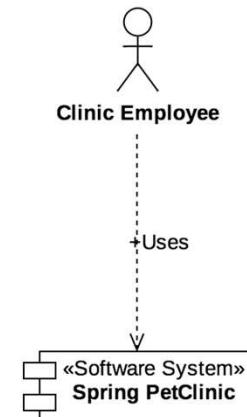
# C4 diagrams & UML



Spring PetClinic - System Context



The System Context diagram for the Spring PetClinic system.

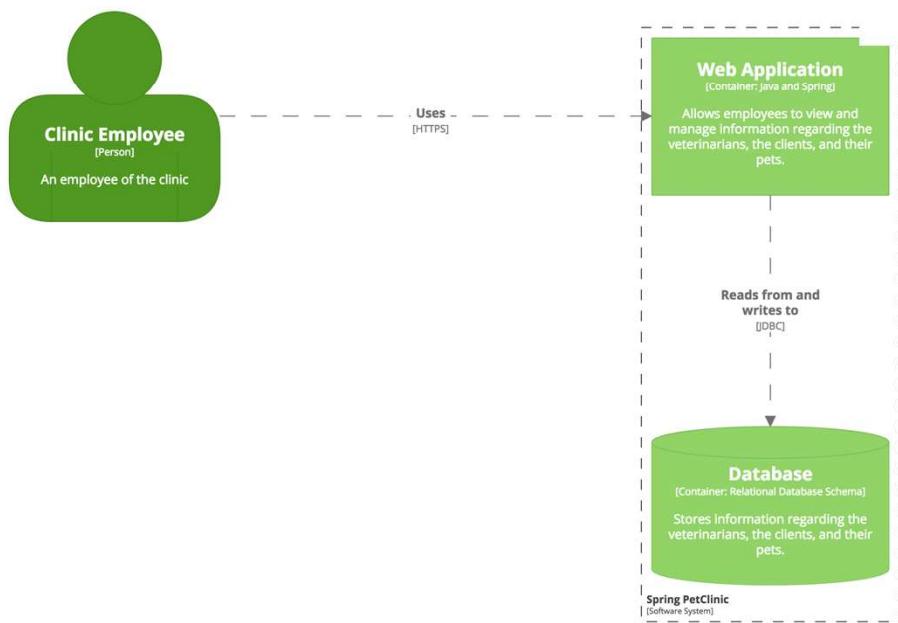


## System Context diagram for Spring PetClinic

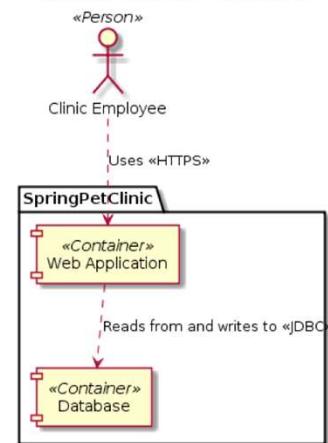
The System Context diagram for the Spring PetClinic system.

Last modified: Thursday 17 August 2017 10:15 UTC | Version: 95de1d9f8bf63560915331664b27a4a75ce1f1f6

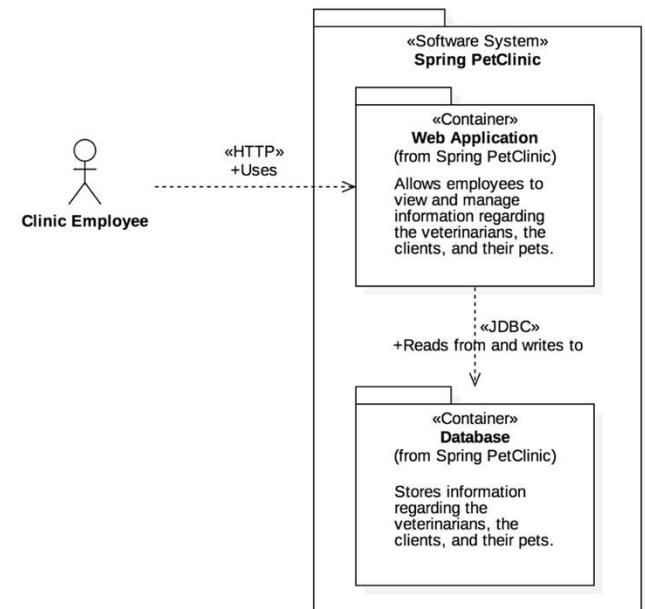
# C4 diagrams & UML



Spring PetClinic - Containers



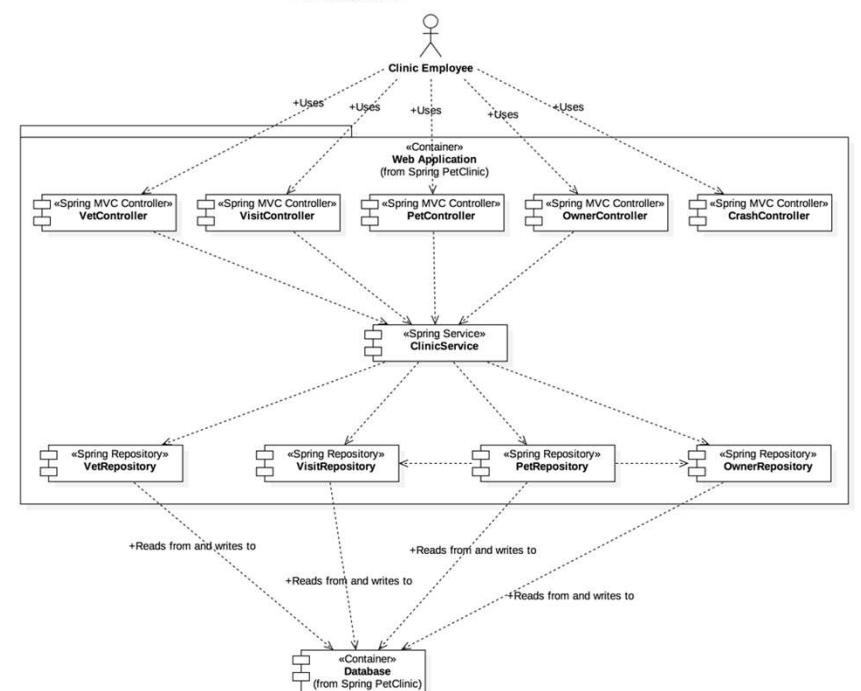
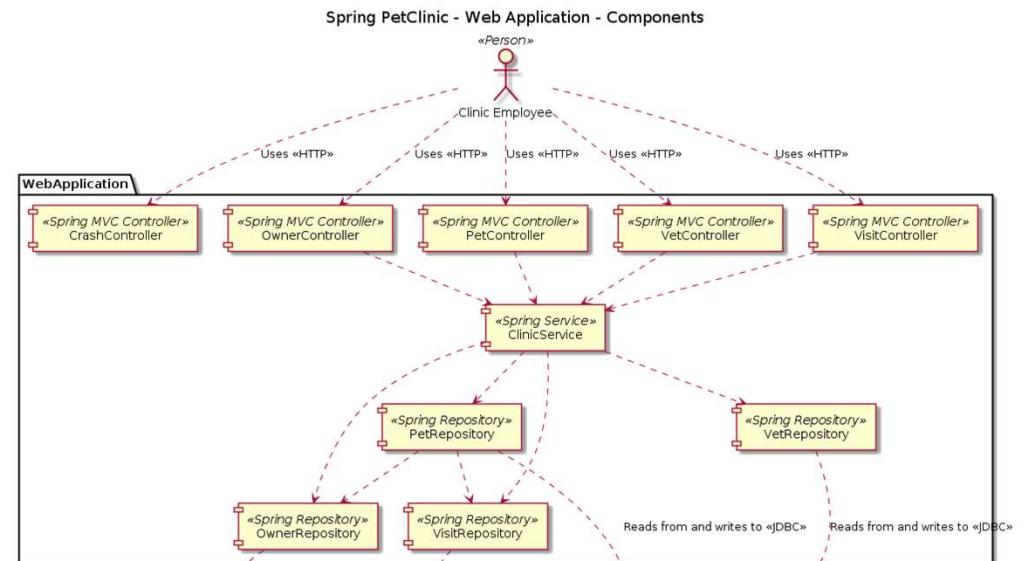
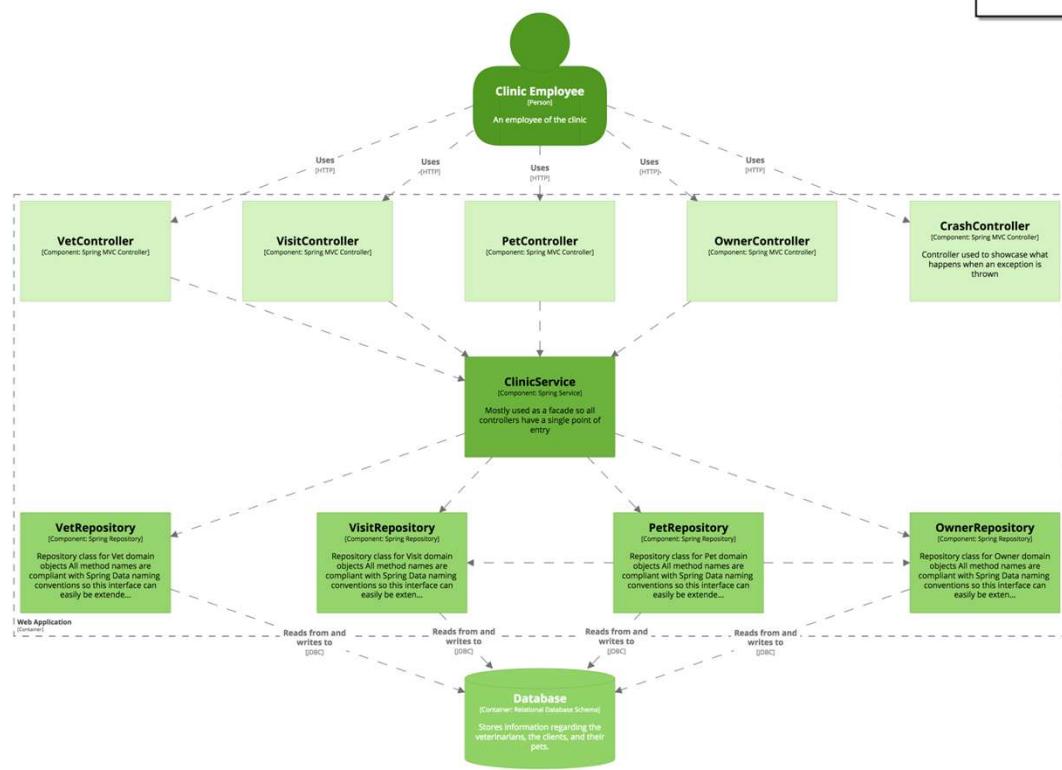
The Container diagram for the Spring PetClinic system.



Container diagram for Spring PetClinic

The Containers diagram for the Spring PetClinic system.  
Last modified: Thursday 18 August 2017 10:15 UTC | Version: 95de1d9f8bf63560915331664b27a4a75ce1f1f6

# C4 diagrams & UML



Component diagram for Spring PetClinic - Web Application

The Component diagram for the SpringPetClinic web application.

Last modified: Thursday 17 August 2017 10:15 UTC | Version: 95de1d9ff8f63560915331664b27a475e1ff16

# AGENDA

INTRO

UML OVERVIEW

DESIGN APPROACHES

C4 MODEL BASICS

SDD INTRO



# Software Design Description (SDD)

Descripción escrita de un producto de software, que un diseñador de software elabora para entregar al equipo de Desarrollo de software una guía general de la arquitectura y diseño del producto de software.

# Software Design Description (SDD)

Según el modelo conceptual de IEEE 1016, SDD se organiza en una o más vistas de diseño.

IEEE 1016 define los siguientes viewpoints:

Context viewpoint  
Composition viewpoint  
Logical viewpoint  
Dependency viewpoint  
Information viewpoint  
Patterns use viewpoint

Interface viewpoint  
Structure viewpoint  
Interaction viewpoint  
State dynamics viewpoint  
Algorithm viewpoint  
Resource viewpoint

# C4 and SDD: Software guidebook

<b>Context</b> A system context diagram, plus some narrative text to "set the scene".	<b>Functional Overview</b> An overview of the software system; perhaps including wireframes, UI mockups, screenshots, workflow diagrams, business process diagrams, etc.	<b>Quality Attributes</b> A list of the quality attributes (non-functional requirements; e.g. performance, scalability, security, etc).	<b>Constraints</b> A list of the environmental constraints (e.g. timescales, budget, technology, team size/skills, etc).	<b>Principles</b> A list of the development and architecture principles (e.g. coding conventions, separation of concerns, patterns, etc).
<b>Software Architecture</b> A description of the software architecture, including static structure (e.g. containers and components) and dynamic/runtime behaviour.	<b>Code</b> A description of important or complicated component implementation details, patterns, frameworks, etc.	<b>Data</b> Data models, entity relationship diagrams, security, data volumes, archiving strategies, backup strategies, etc.	This is a <b>starting point</b> ; add and remove sections as necessary.	
<b>Infrastructure Architecture</b> A description of the infrastructure available to run the software system.	<b>Deployment</b> The mapping of software (e.g. containers) to infrastructure.	<b>Development Environment</b> A description of how a new developer gets started.	<b>Operation and Support</b> An overview of how the software system is operated, supported, monitored, etc.	<b>Decision Log</b> A log of the major decisions made; e.g. as free format text or a collection of "Architecture Decision Records".

---

# RESUMEN

Recordemos

Agile

UML

Design Approaches

C4 Model

SDD



---

# REFERENCIAS

Para profundizar

<https://c4model.com>

<https://martinfowler.com/tags/evolutionary%20design.html>



# PREGRADO

## Ingeniería de Software

Escuela de Ingeniería de Sistemas y Computación | Facultad de Ingeniería



**UPC**

Universidad Peruana  
de Ciencias Aplicadas

Prolongación Primavera 2390,  
Monterrico, Santiago de Surco  
Lima 33 - Perú  
T 511 313 3333  
<https://www.upc.edu.pe>

**exígete, innova**