



C4 model на практике

Денис Цветчих

Тинькофф, DevBrothers





Денис Цветцих

Тинькофф, DevBrothers

→ Диаграммы C4 model

→ Инструменты создания диаграмм

→ Как развиваться на каждом уровне



О чем поговорим



Simon Brown



C4 model (2006-2011, 2018 – статья InfoQ)
<https://c4model.com/>

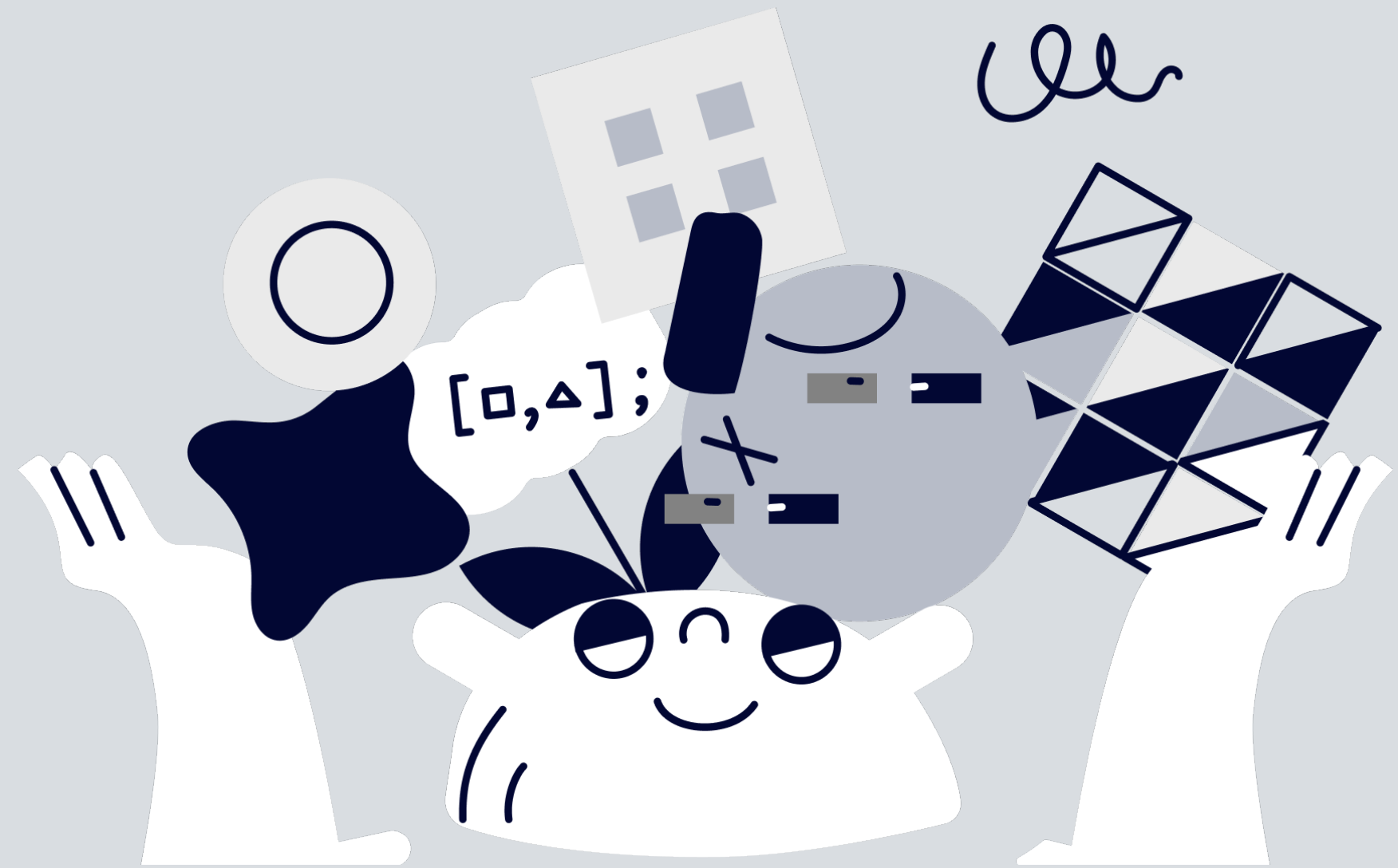


Книга «Software Architecture for Developers»
<https://softwarearchitecturefordevelopers.com/>



Clean Architecture «Недостающая глава»
<https://youtu.be/5OjqD-ow8GE>

Зачем



UML
(150 сущностей)

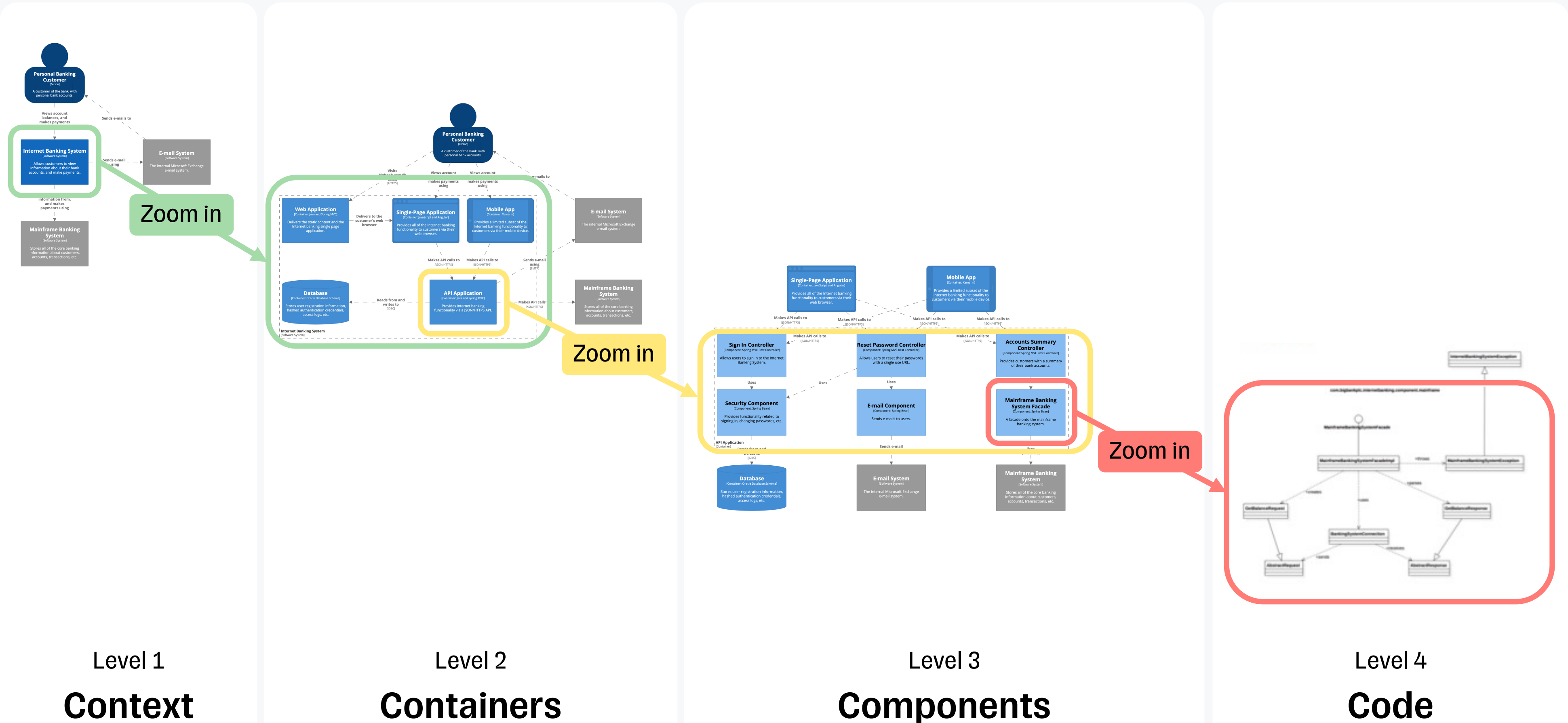


ArchiMate
(40 сущностей)

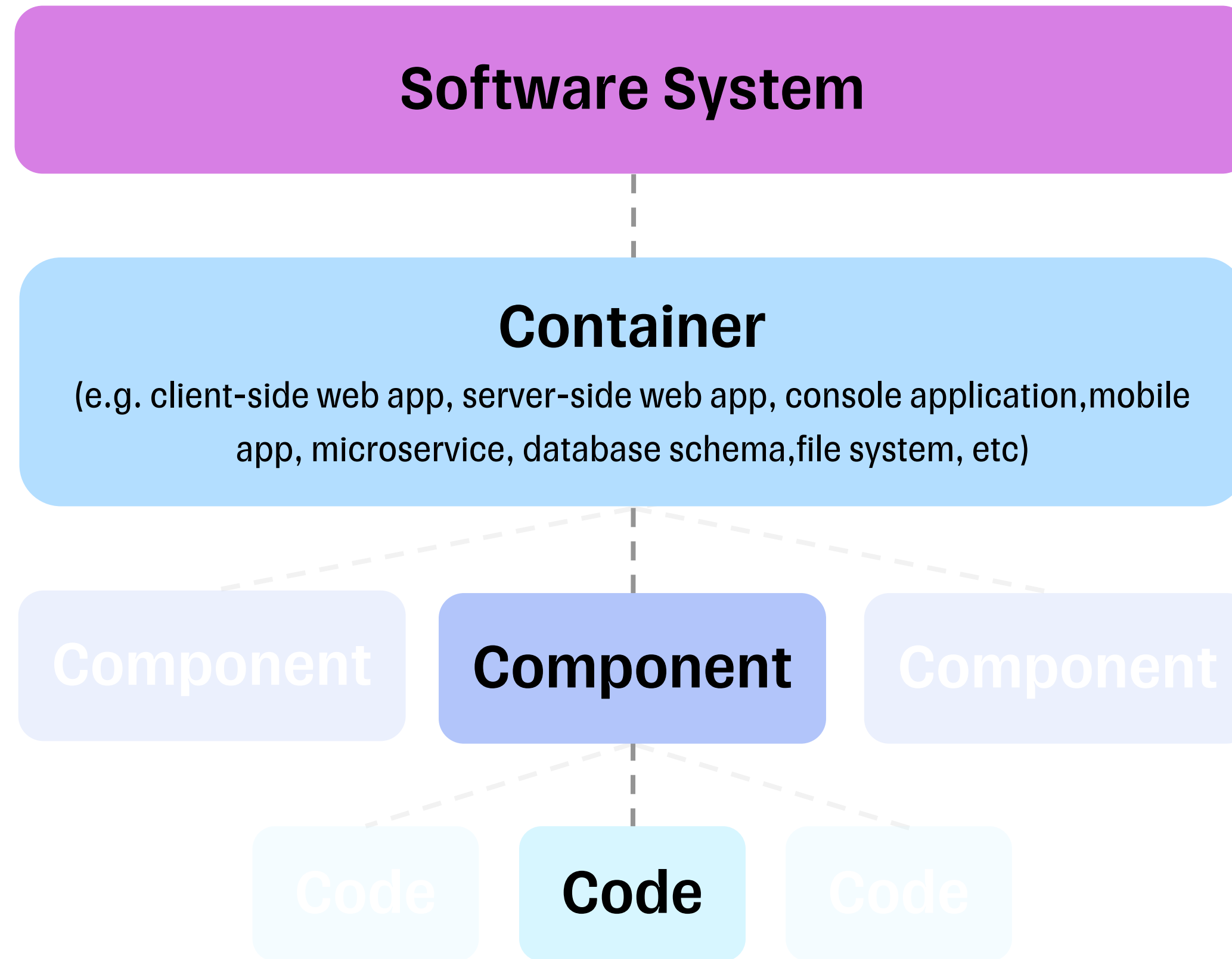


C4 model
(5 сущностей)

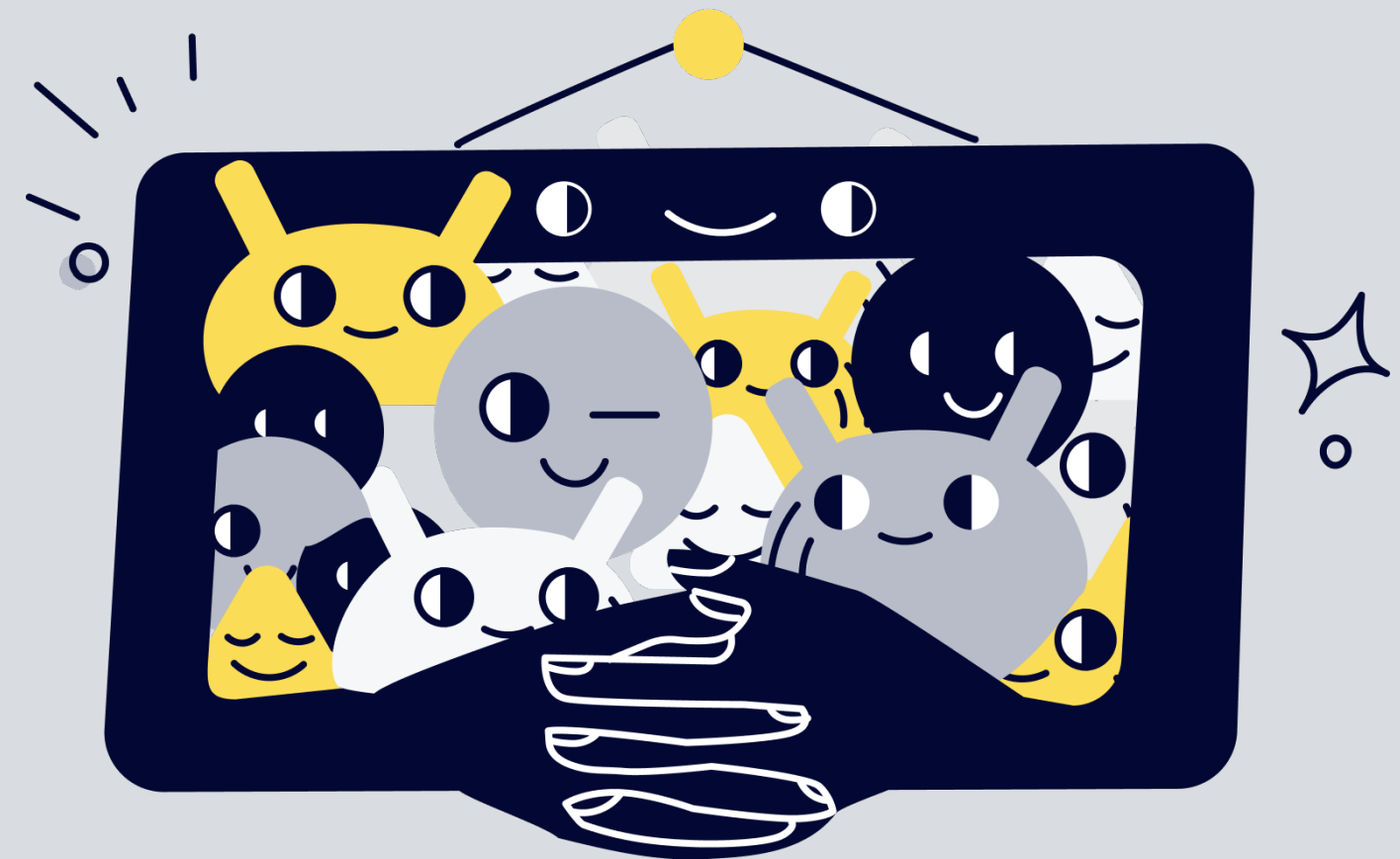
В чем отличия C4 model



Для кого



Когда



Предварительное
проектирование



Ретроспективное
документирование

Абстракции



Человек (Person)



Программная система
(Software System)



Контейнер (Container)



Компонент (Component)



Код (Code)

Человек



Один из пользователей системы (роль)

Программная система

Internet Banking System [Software System]

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

E-mail System [Software System]

The internal Microsoft Exchange e-mail system.



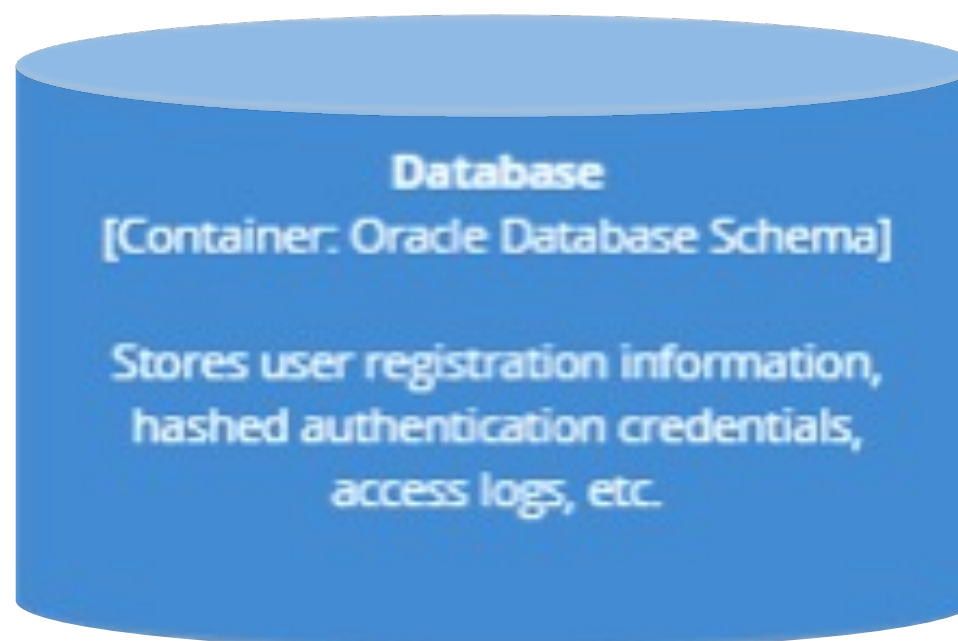
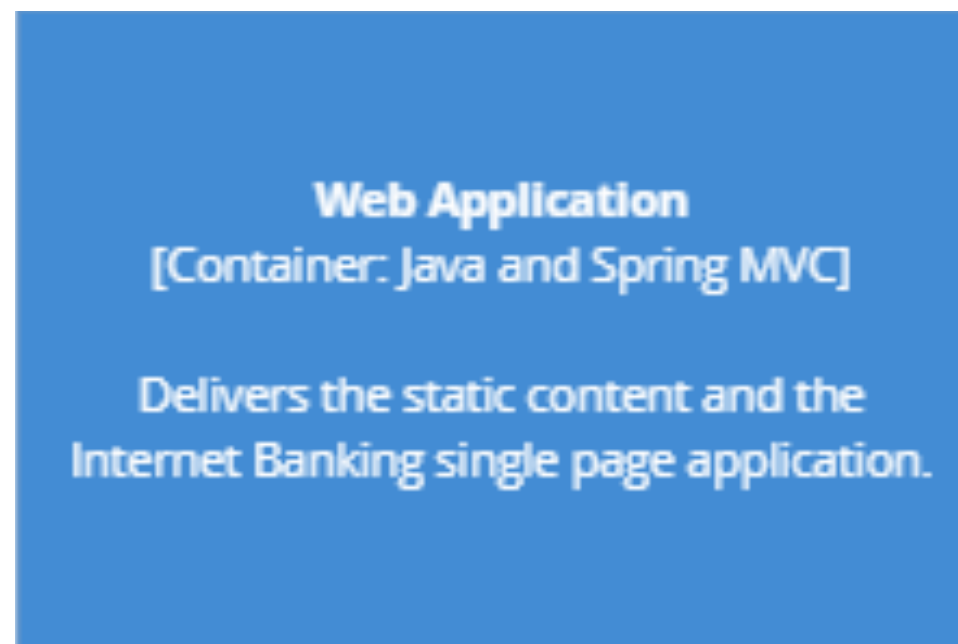
Моделируемая система



Связанные системы

- Зависят от моделируемой
- Моделируемая зависит от них

Контейнер



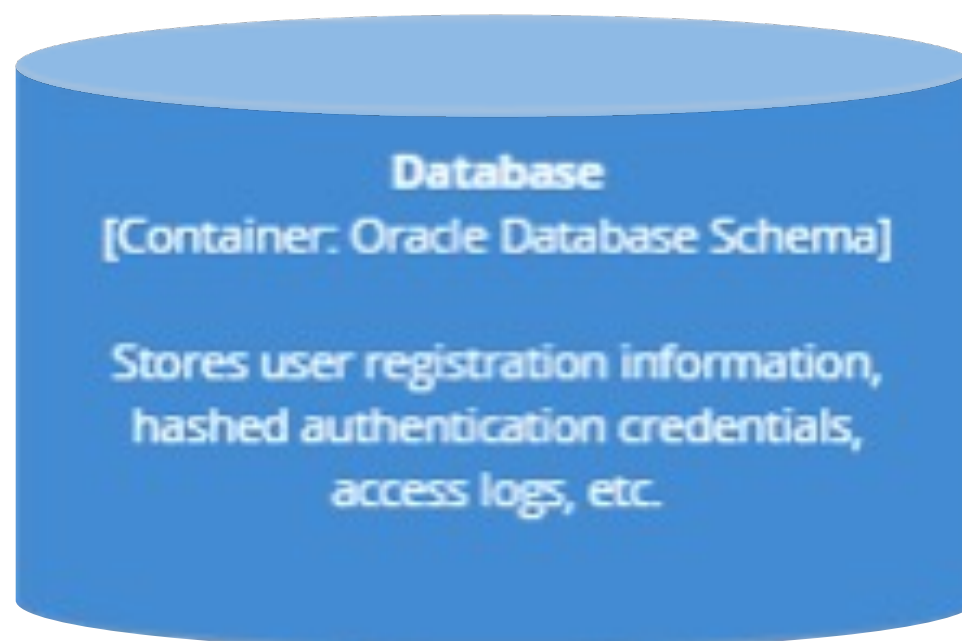
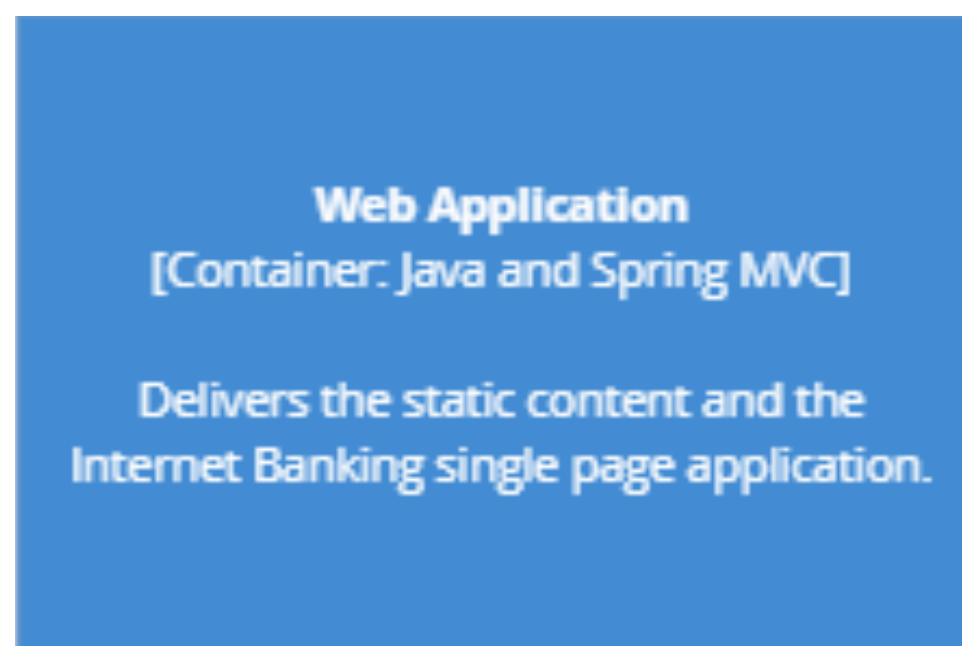
Не Docker 😊



Отдельно запускаемый/развертываемый модуль

- приложение или хранилище данных

Примеры контейнеров



- ✓ Веб-приложение (WebAPI, серверный рендеринг)
- ✓ Фронтенд в браузере
- ✓ Настольное приложение
- ✓ Мобильное приложение
- ✓ База данных
- ✓ Файловая система
- ✓ Blob storage (Amazon S3, Azure Blob)
- ✓ Бессерверная функция (Amazon Lambda, Azure Function)

Компонент

E-mail Component
[Component: Spring Bean]
Sends e-mail to users.

- Группа связанных функций, инкапсулированных за четко определенным интерфейсом
- Набор классов реализации за интерфейсом
- Не является единицей развертывания
- Упаковка в .dll или jar – отдельный вопрос
- На практике – .dll или jar

Код

Account
+ State: AccountState
+ Close()

Table	
PK	<u>Id</u>
	FirstName
	LastName
	MiddleName



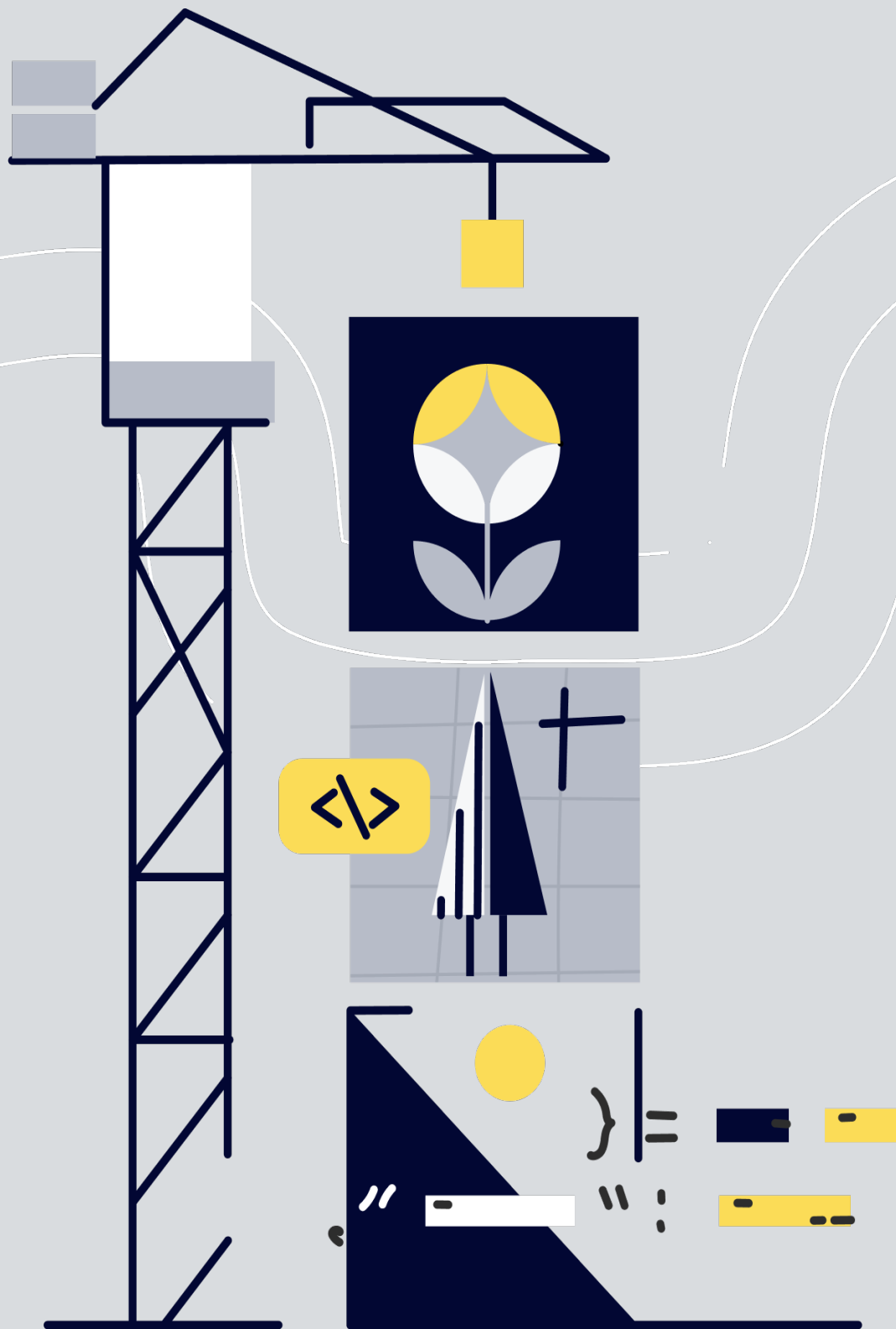
Класс

- Интерфейс
- Перечисление



Таблица БД

Диаграммы



Основные

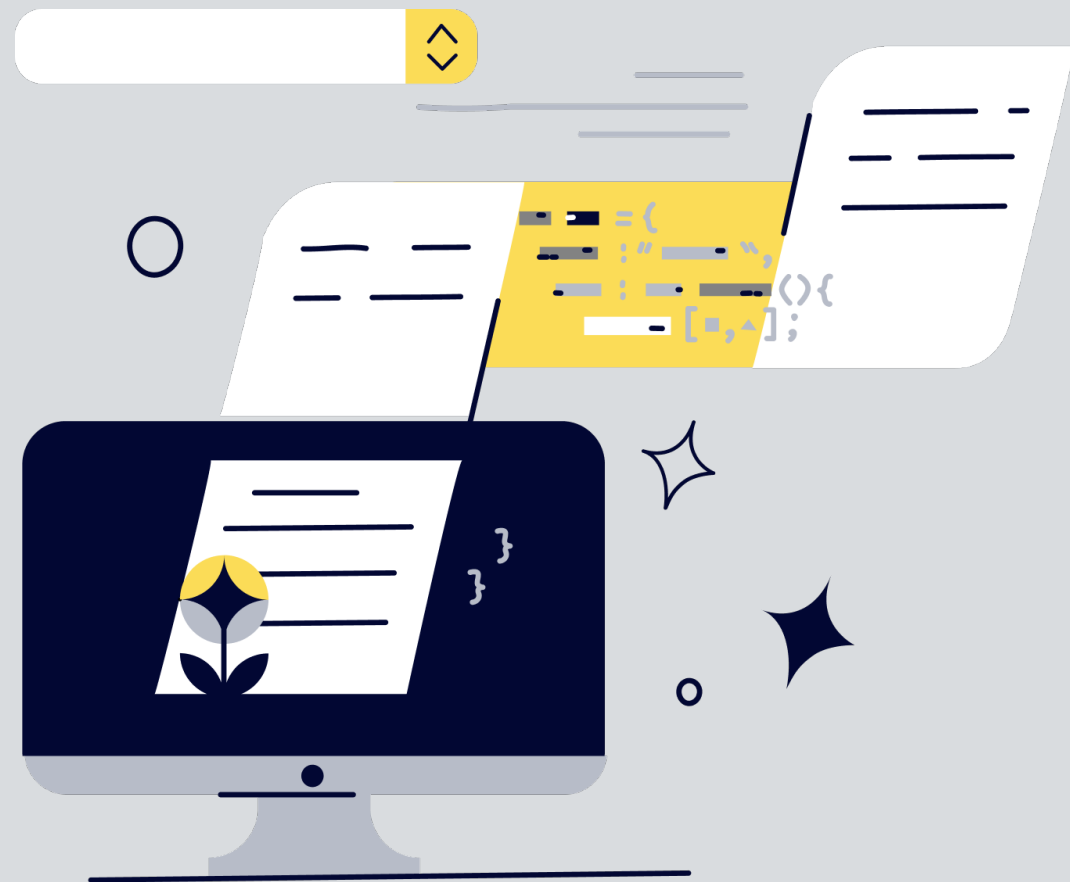
- Контекста (Context)
- Контейнеров (Container)
- Компонентов (Component)
- Кода (Code)



Дополнительные

- Динамическая диаграмма
- Схема развертывания
- Схема ландшафта системы

Диаграмма контекста



Что показывается

- Моделируемая система
- Категории пользователей
- Системы, с которыми взаимодействует моделируемая



Что не показывается

- Технические детали (протоколы, технологии)



Целевая аудитория

- Все

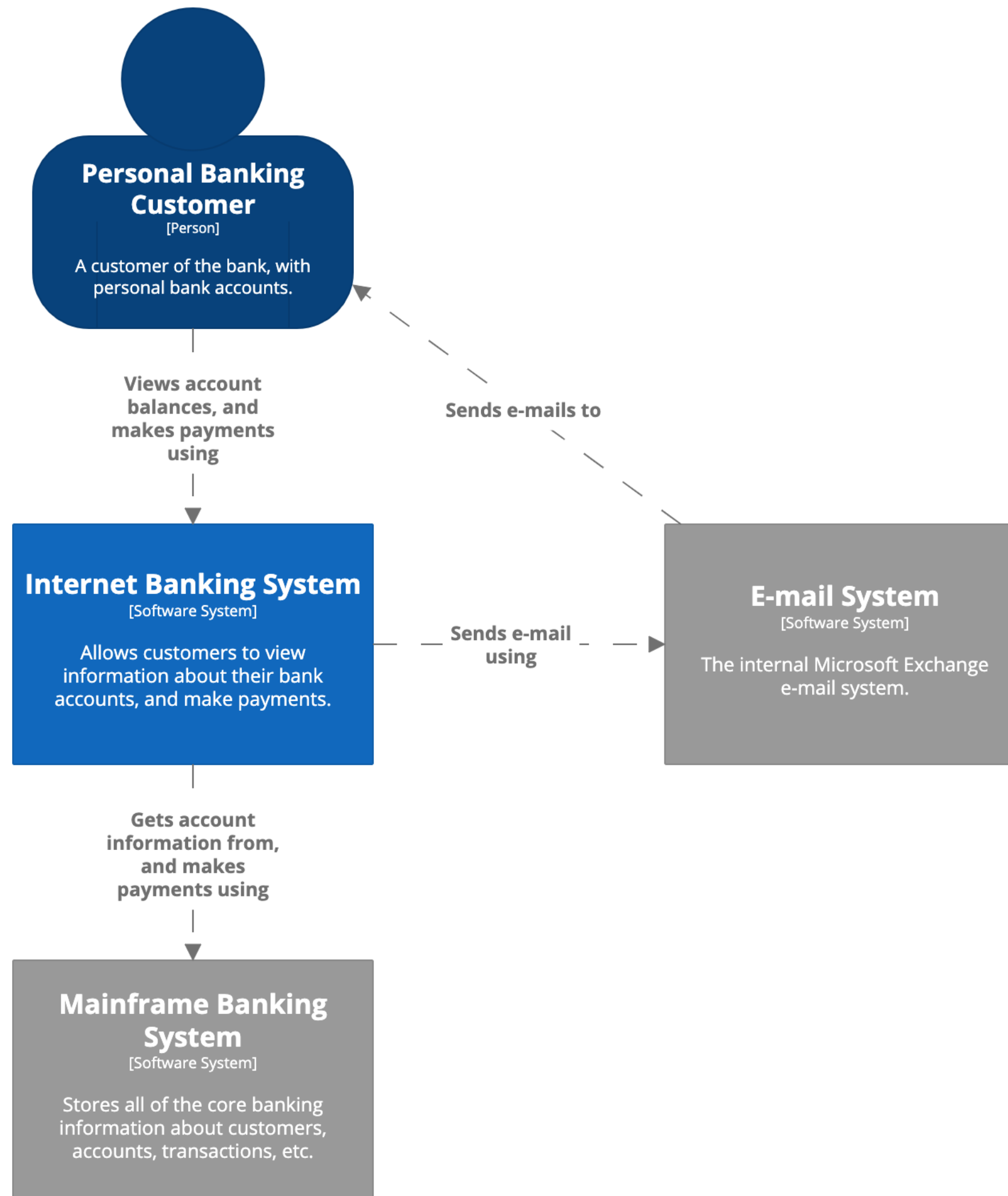


Диаграмма контейнеров



Что показывается

- Контейнеры и их обязанности
- Технологии
- Взаимодействие контейнеров
- [Возможно] Внешние системы и люди



Что не показывается

- Кластеризация
- Репликация, шардинг
- Развертывание (load balancer)



Целевая аудитория

- Технари (разработчики, архитекторы)

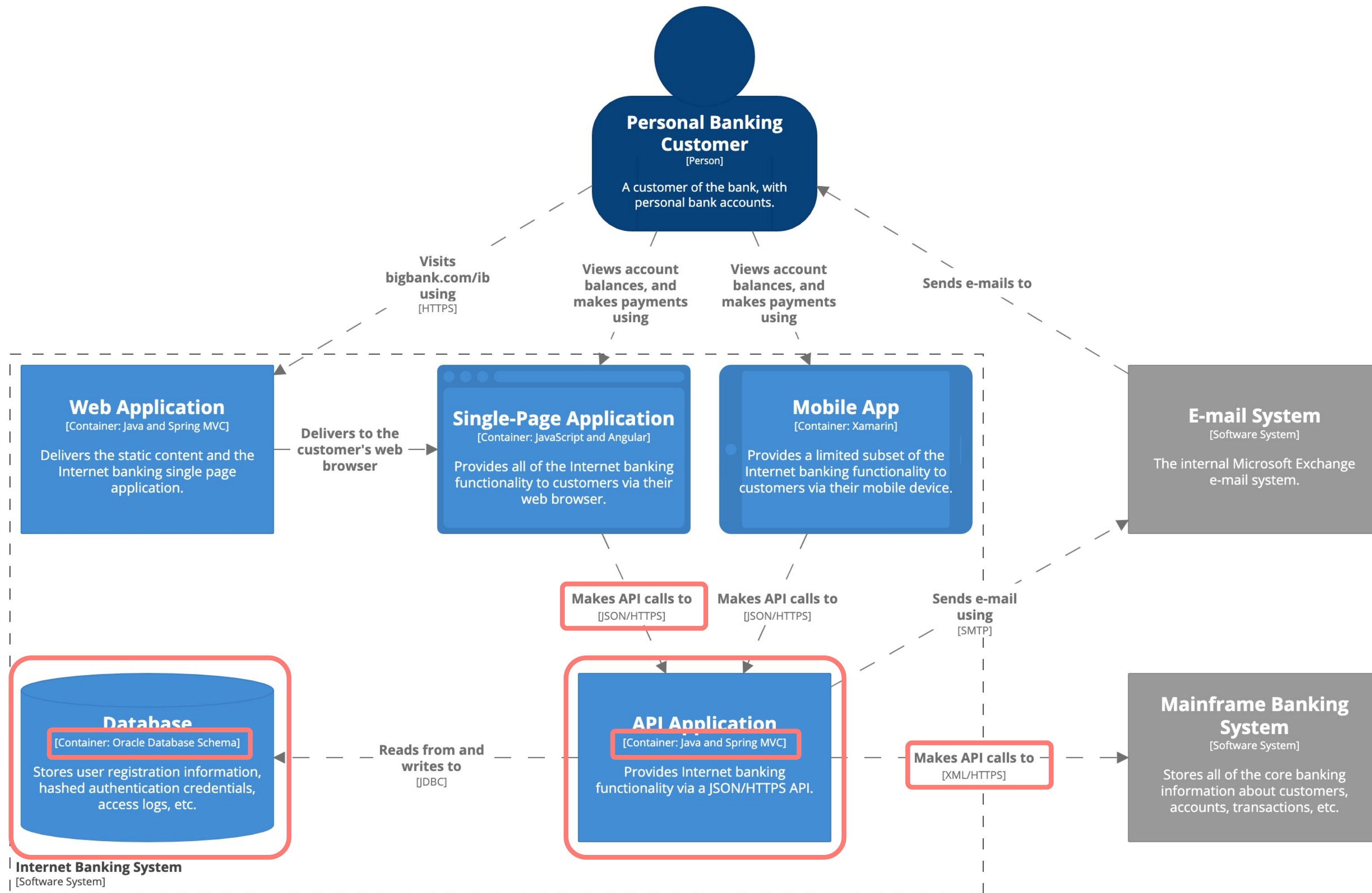


Диаграмма КОМПОНЕНТОВ



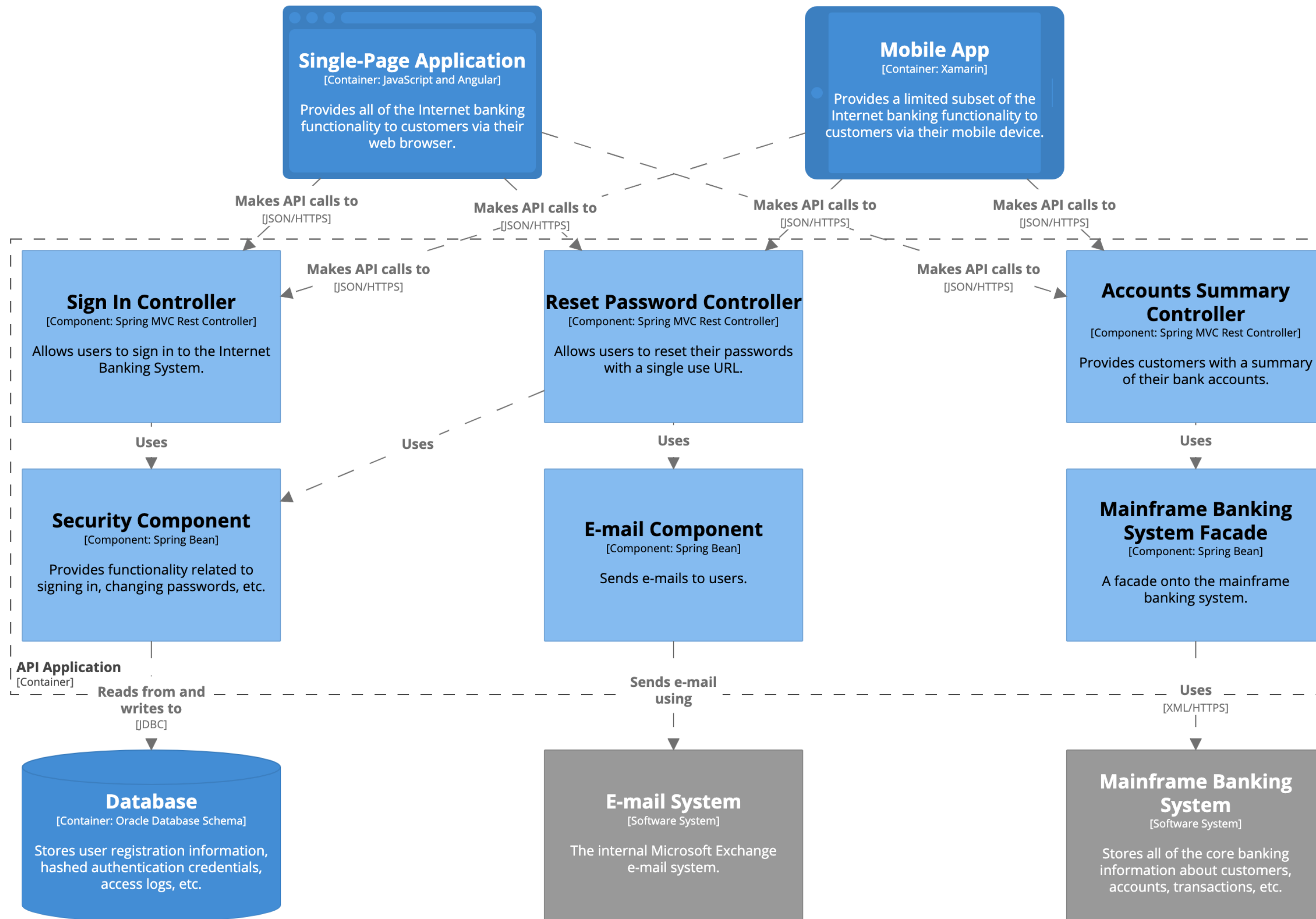
Что показывается

- Компоненты
- Технологии
- [Возможно] Внешние системы и люди



Целевая аудитория

- Технари (разработчики, архитекторы)

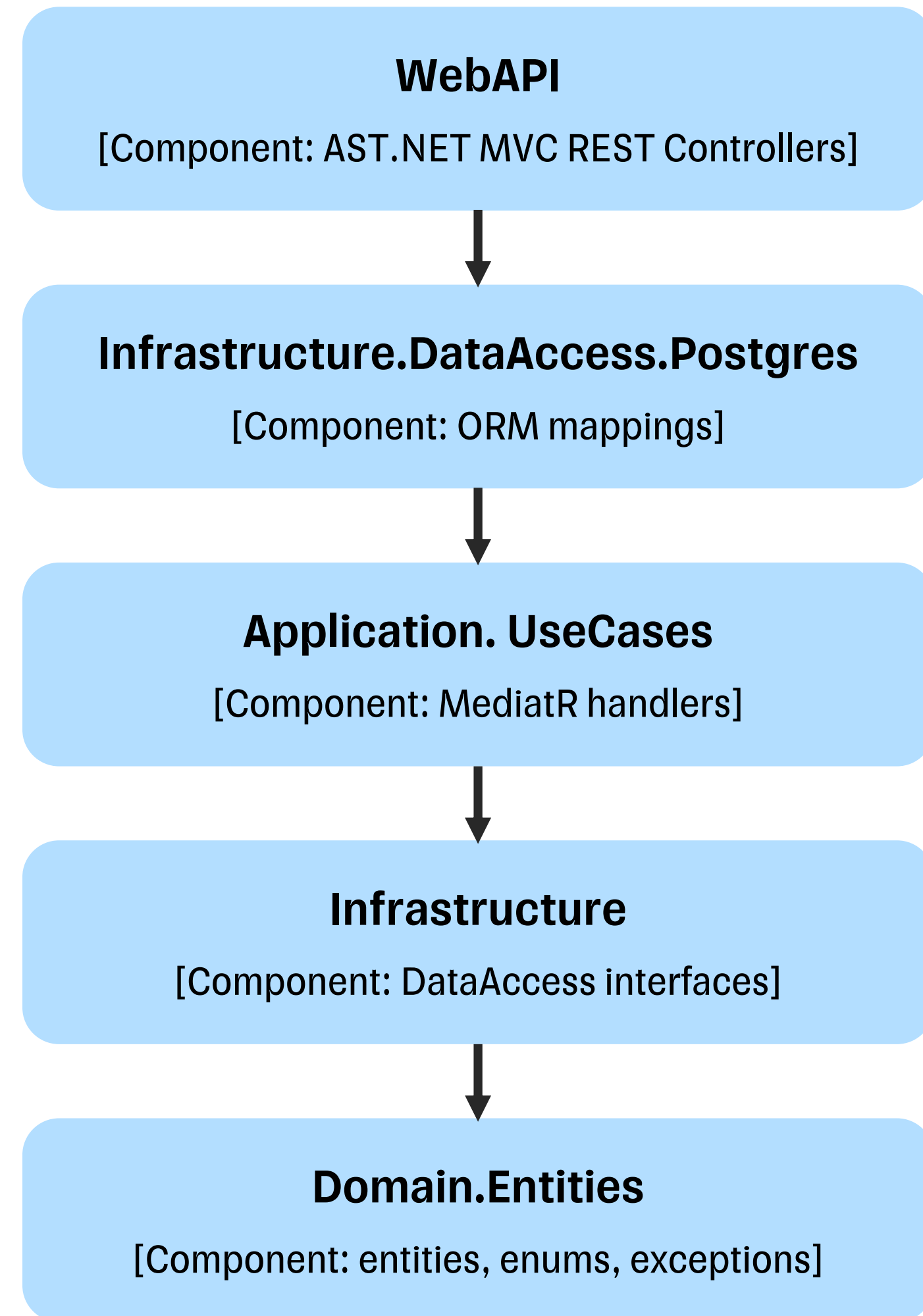


Component diagram for Internet Banking System - API Application

The component diagram for the API Application.
 Workspace last modified: Wed Feb 05 2020 09:33:36 GMT+0100 (Central European Standard Time)

На практике - расслоение

Компонент – dll, jar, ...



IDE строит диаграммы

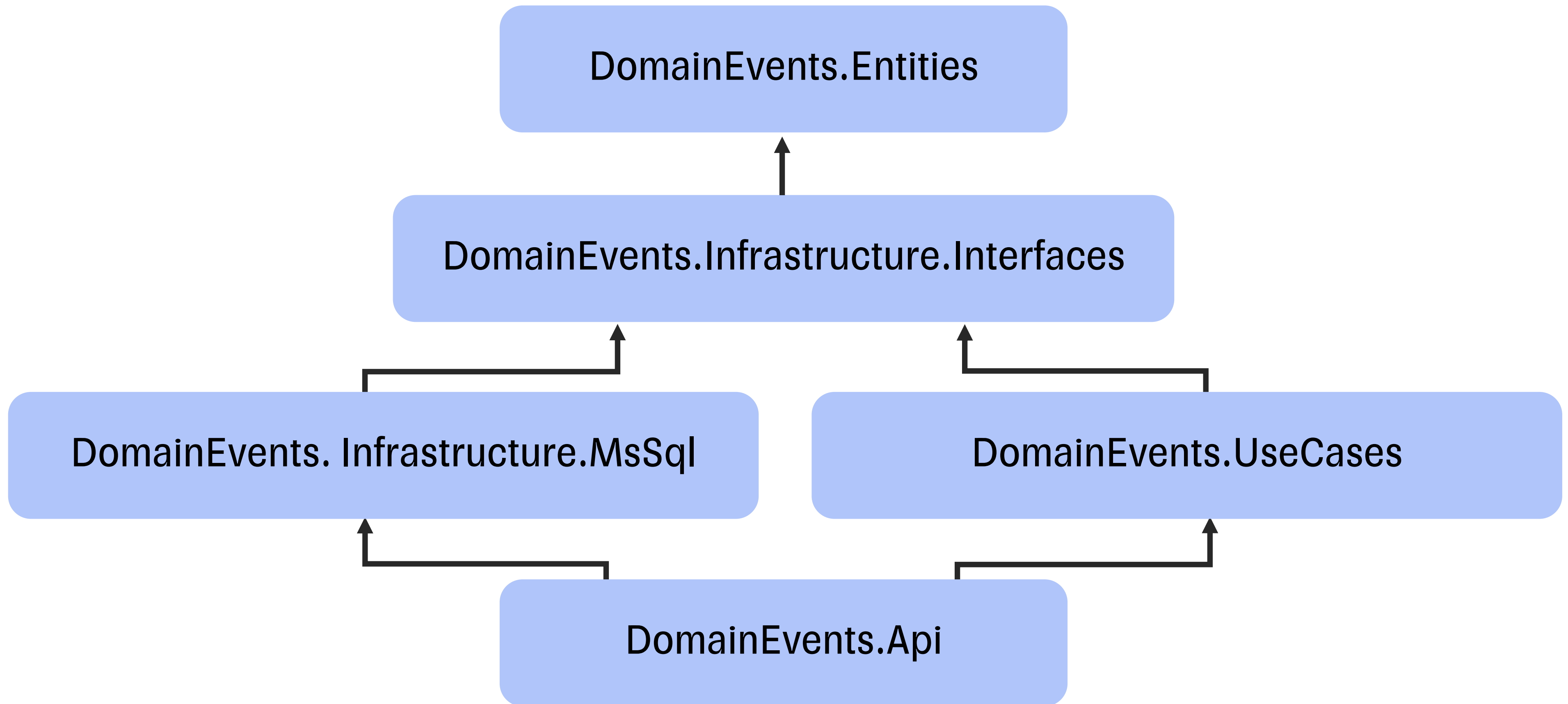
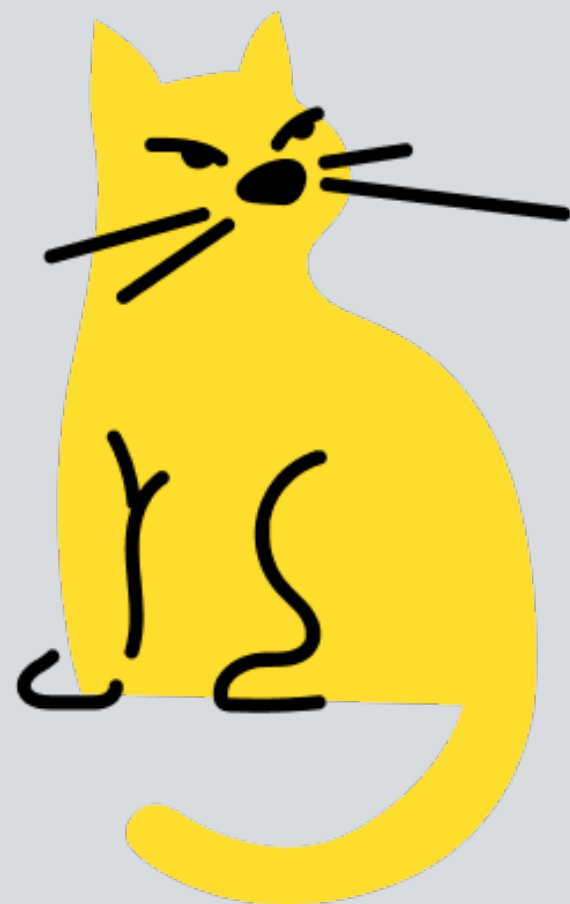


Диаграмма кода



Что показывается

- Классы (диаграмма классов)
- Таблицы БД (Entity Relationship диаграмма)
- Отношения между ними
 - наследование, реализация
 - ассоциация, агрегация, композиция



Генерируется в IDE

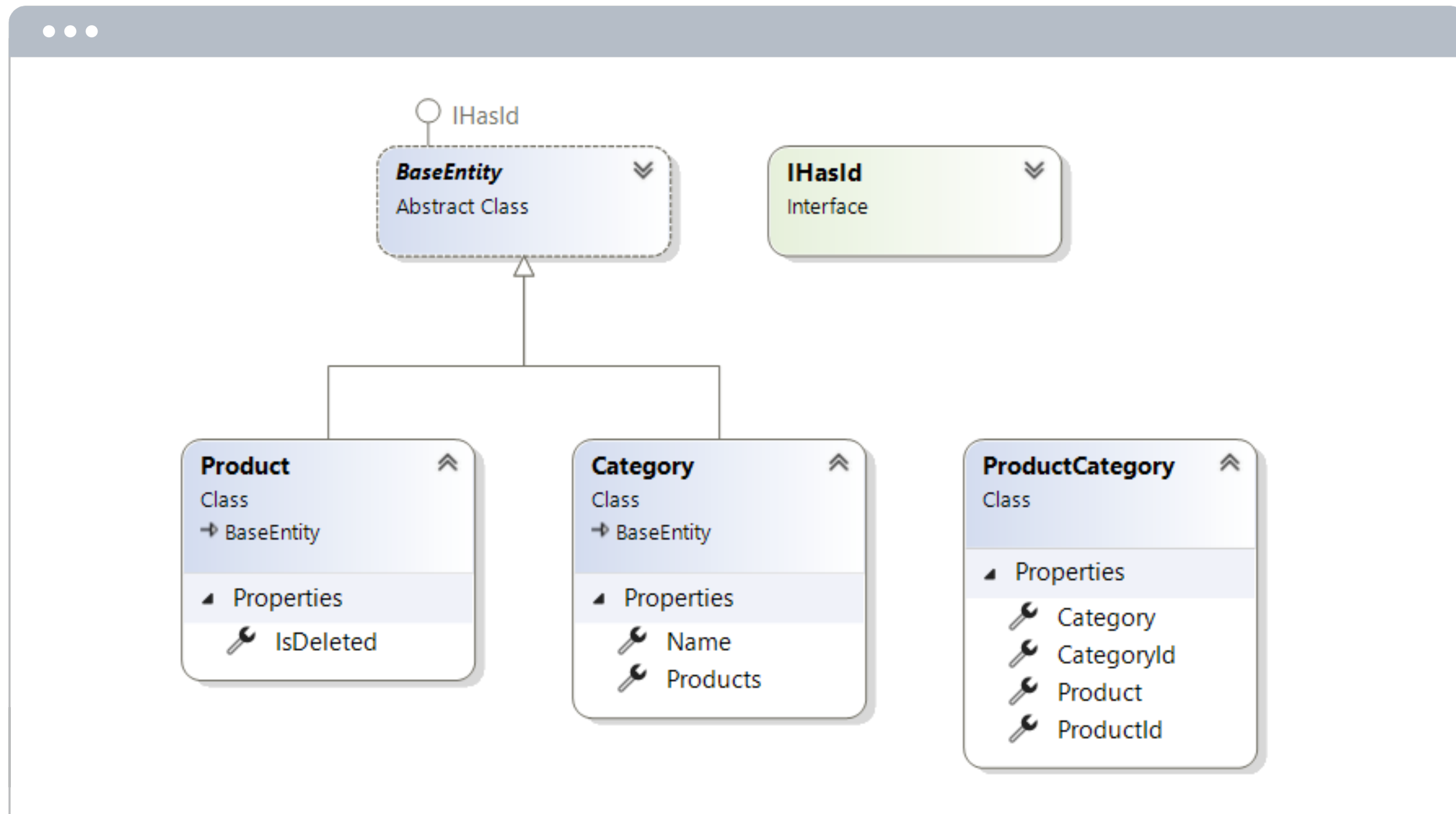
- Или в специальных инструментах EnterpriseArchitect



Целевая аудитория

- Технари (разработчики, архитекторы)

Диаграмма классов



Entity Relationship диаграмма

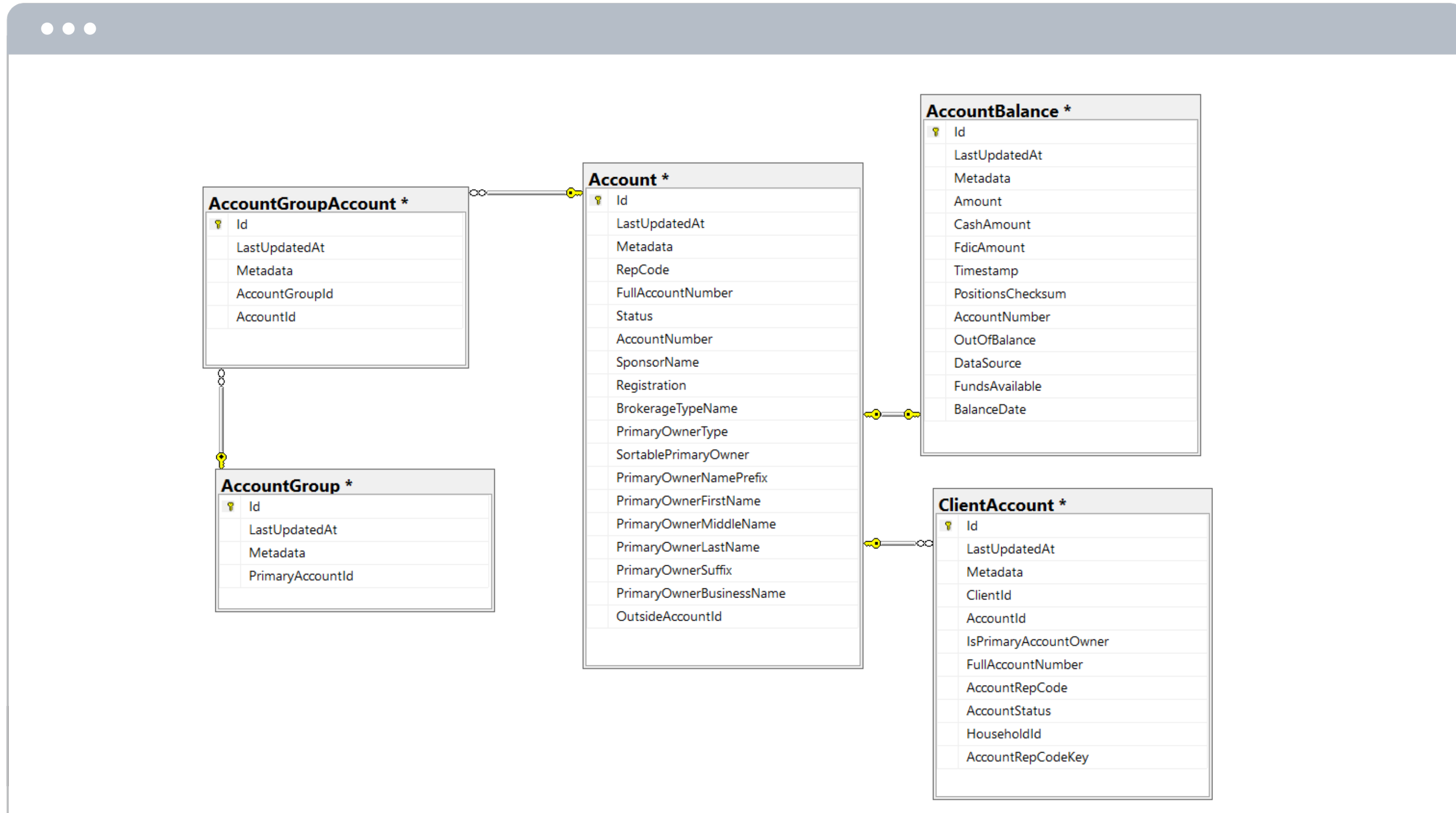
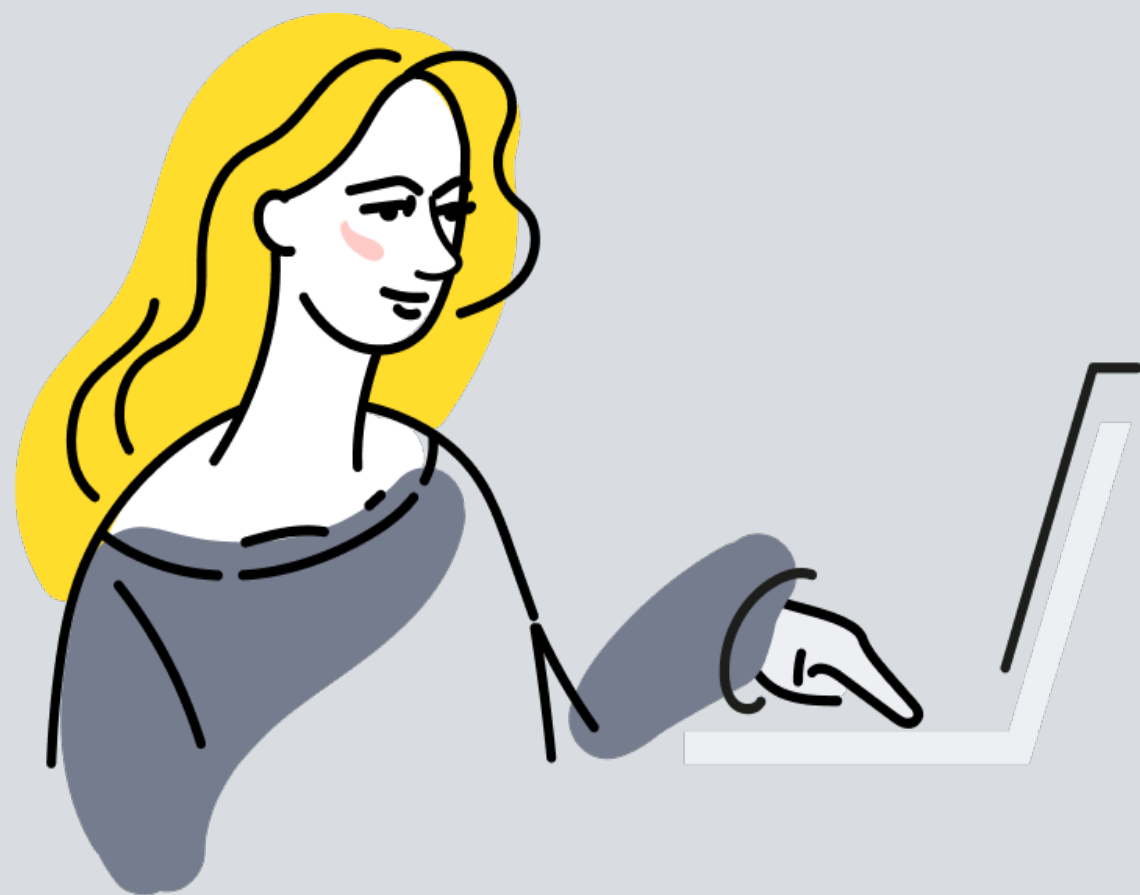


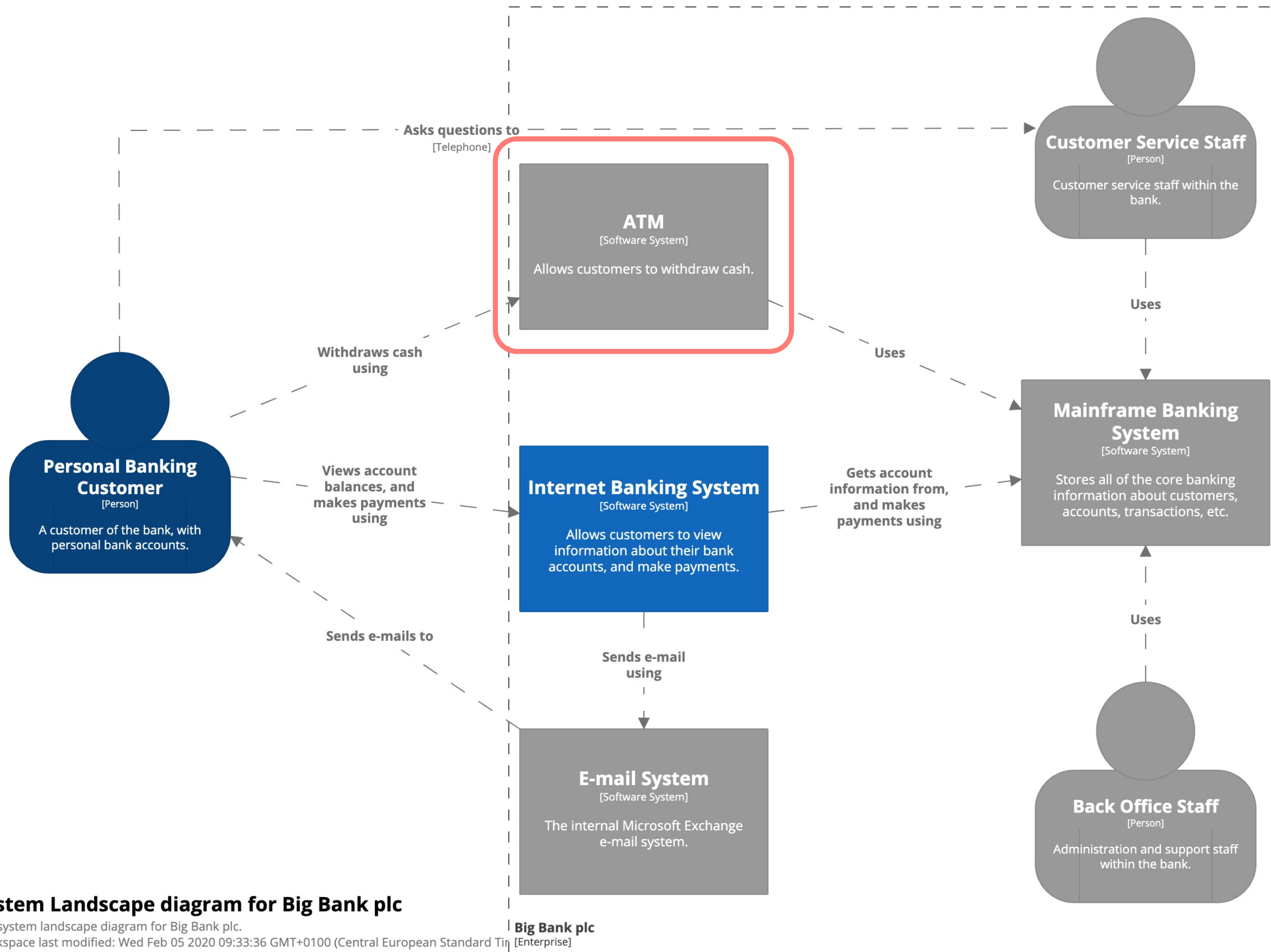
Схема ландшафта системы



Сверху С4



Показывает место системы на уровне
предприятия



System Landscape diagram for Big Bank plc

The system landscape diagram for Big Bank plc.
 Workspace last modified: Wed Feb 05 2020 09:33:36 GMT+0100 (Central European Standard Time)

Динамическая диаграмма



Взаимодействие элементов для реализации UseCase

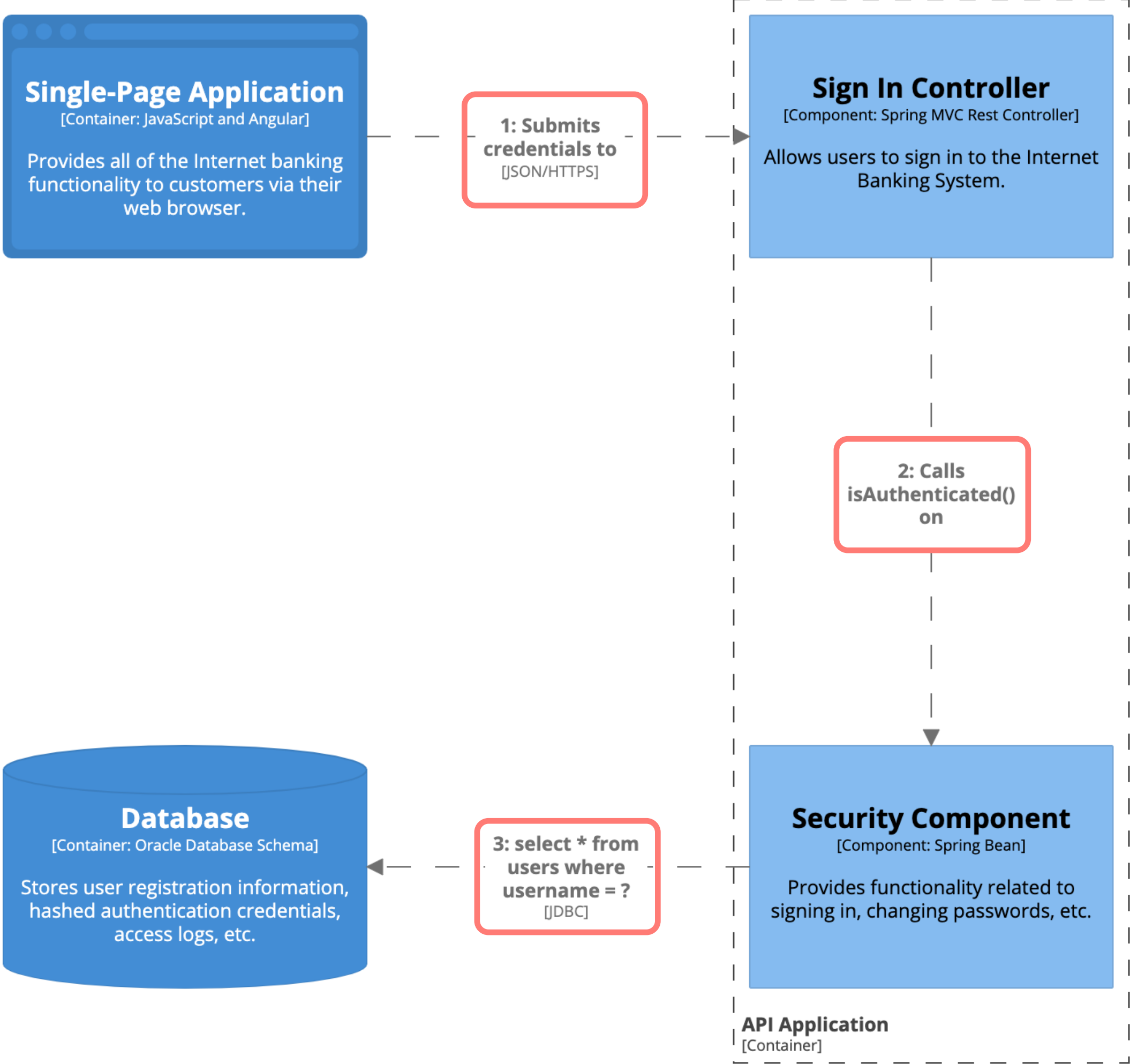


Схема развертывания



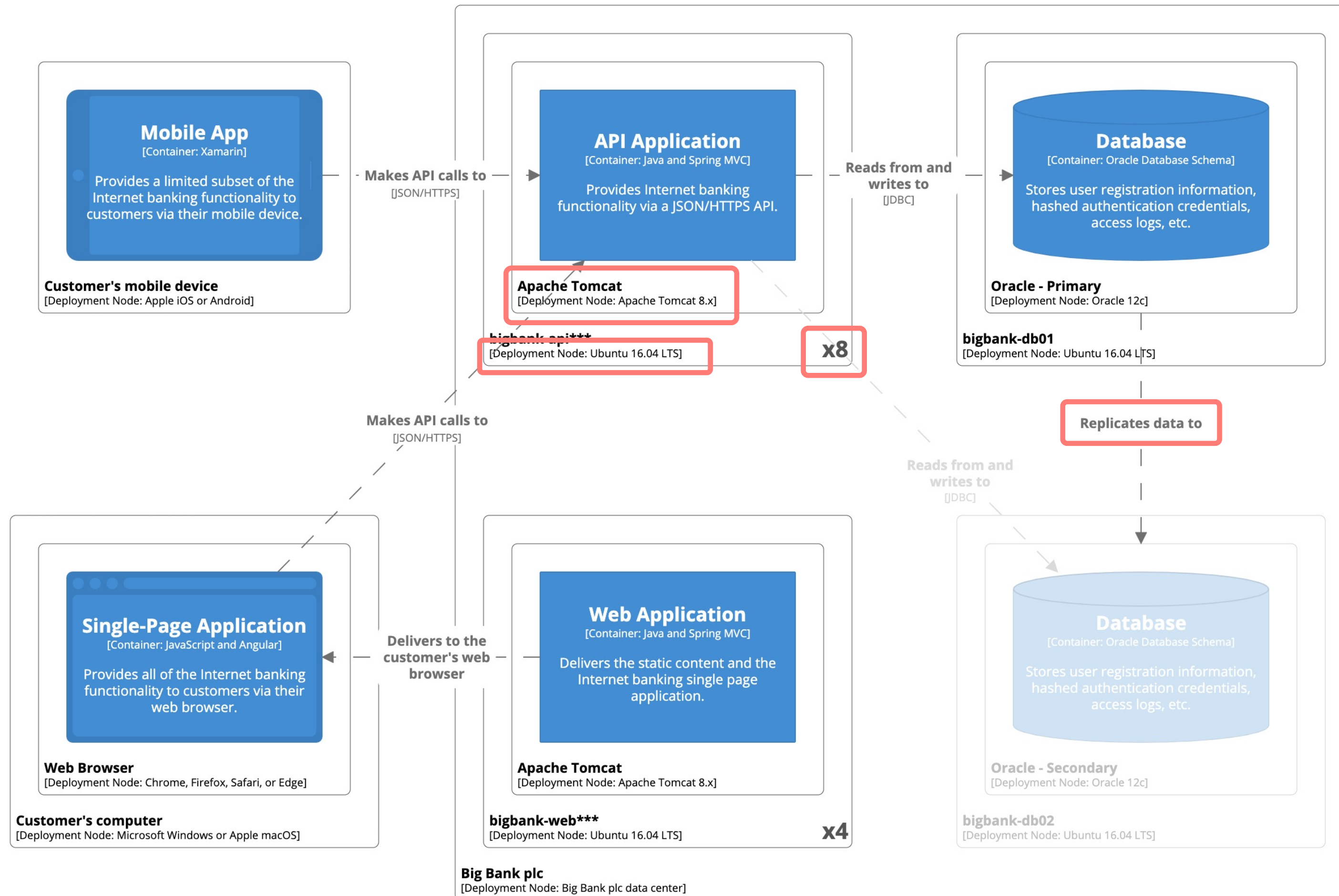
Показывает связь между контейнерами и узлами развертывания

- Физический (сервер)
- Виртуальный (виртуальная машина)
- Контейнерный (Docker контейнер)
- Выполнения (веб-сервер)
- Балансировщик нагрузки, ...



Целевая аудитория

- Разработчики
- DevOps
- SRE

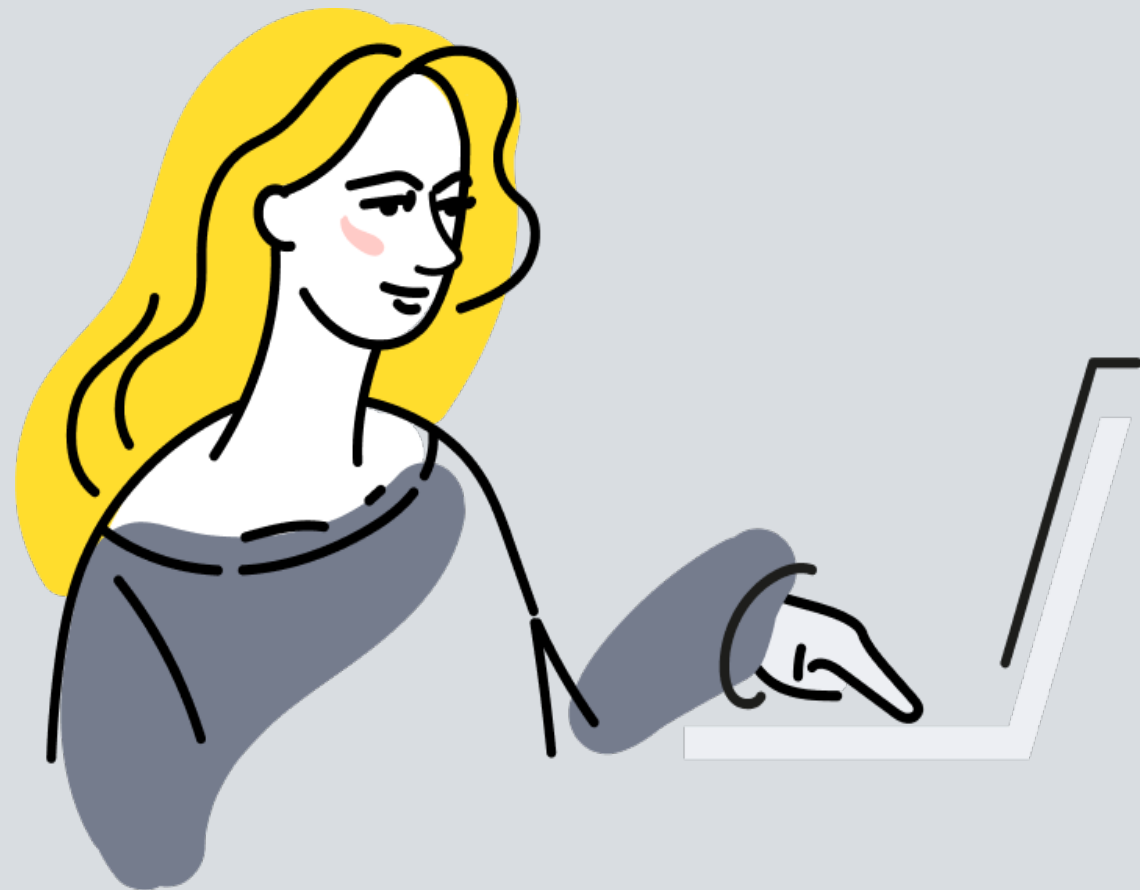


Deployment diagram for Internet Banking System - Live

An example live deployment scenario for the Internet Banking System.

Workspace last modified: Wed Feb 05 2020 09:33:36 GMT+0100 (Central European Standard Time)

Инструменты



- Код – диаграмма классов и Entity Relationship диаграмма
 - IDE
 - Enterprise Architect

- Компонент – зависимости между jar, dll в IDE

- Контейнер

- Контекст

Диаграммы контейнеров и контекстов

The screenshot shows a web page with a list of diagramming tools. The tools are organized into three columns based on their type: Static diagrams, Dynamic diagrams, and Deployment diagrams. Each tool is represented by a button with its name. A 'Recommended' badge is placed under the 'Reuse elements across multiple diagrams' category.

- Static diagrams
(e.g. system context, container, and component diagrams)
- Dynamic diagrams
(e.g. collaboration or sequence diagrams)
- Deployment diagrams
(e.g. diagrams showing deployment and infrastructure concerns)
- Open source
(free, fork/customize, etc)
- Reuse elements across multiple diagrams
(to keep multiple diagrams in sync automatically when you rename elements)
- Rendering tool independent
(to render diagrams with different tools or visualisation formats such as diagrams, graphs, etc)
- Diagrams and models as code
(for easy version control and integration into build pipelines/other tools)

Recommended

Archi Archinsight Archipeg Astah C4-PlantUML c4builder C4Sharp Carbide
CUE4Puml4C4 Diagrams diagrams.net Excalidraw Figma Gaphor Gliffy IcePanel
Lucidchart Microsoft Visio Mermaid Miro Model Mood OmniGraffle Overarch
pumla PyStructurizr Sparx Enterprise Architect RDB modeling Structurizr
Visual Paradigm yEd

Draw.io

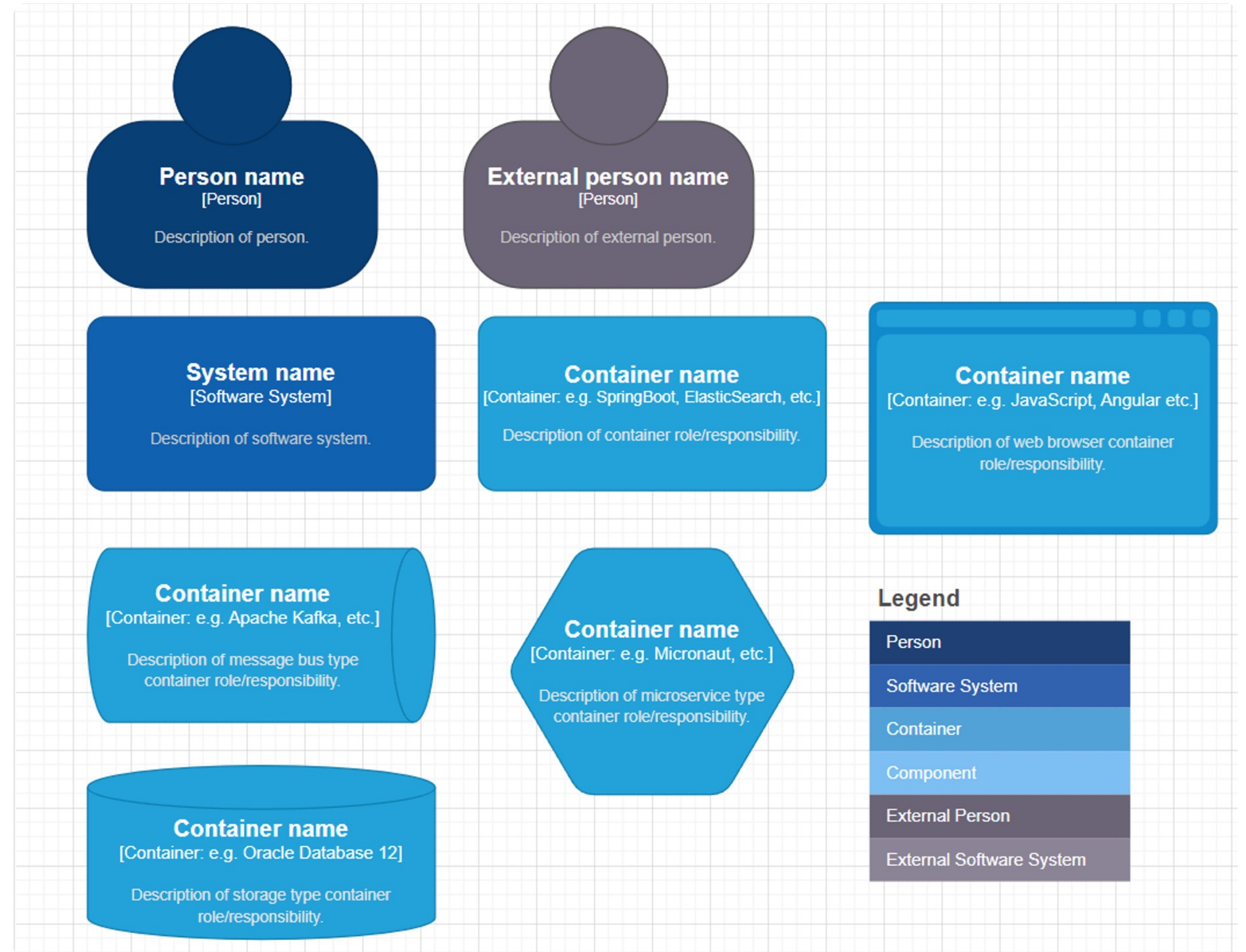


Контейнеры

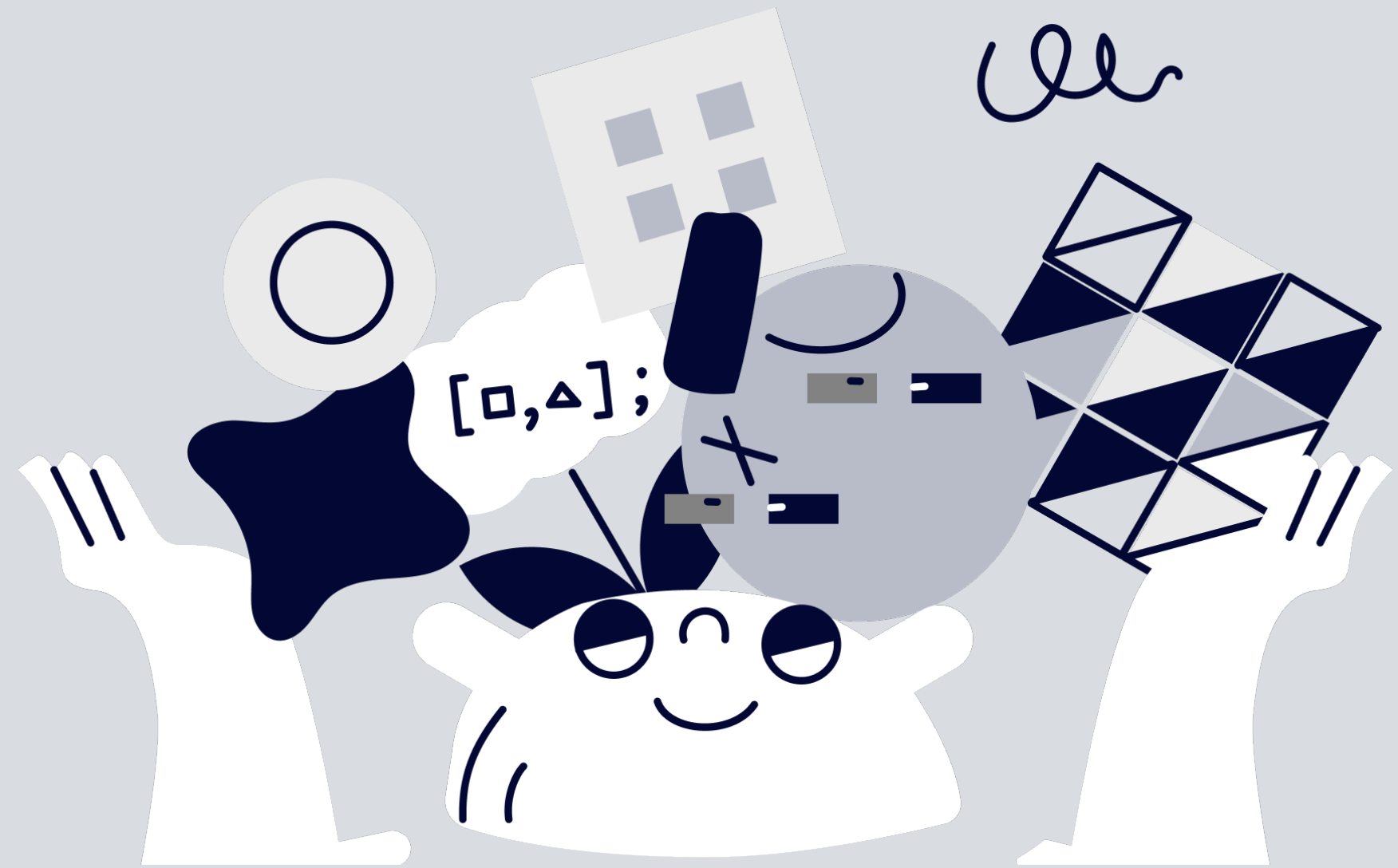
- БД
- MessageBroker
- Микросервис
- Браузер



Нет диаграммы как кода



Примерно то же самое



Miro

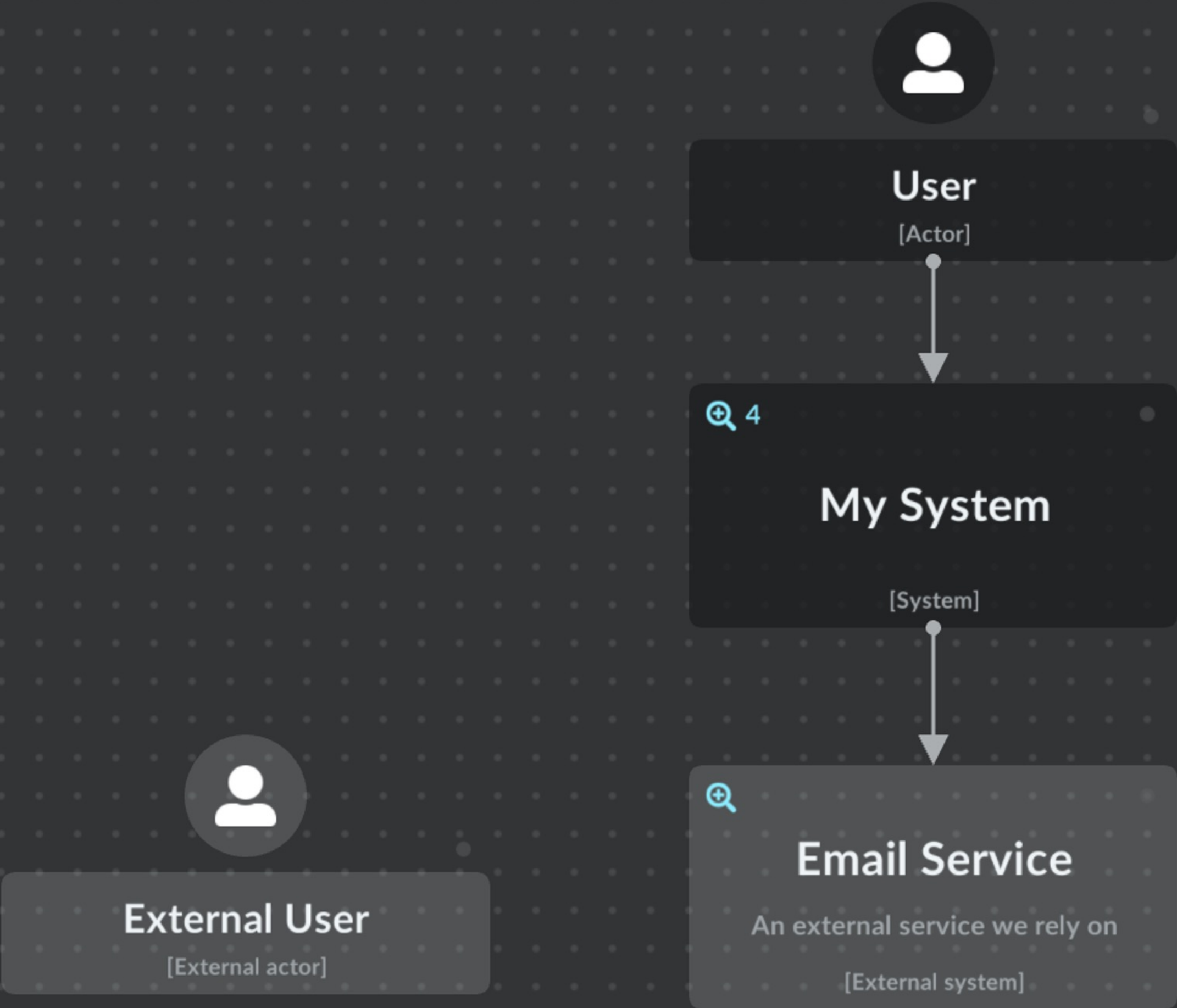


Figma

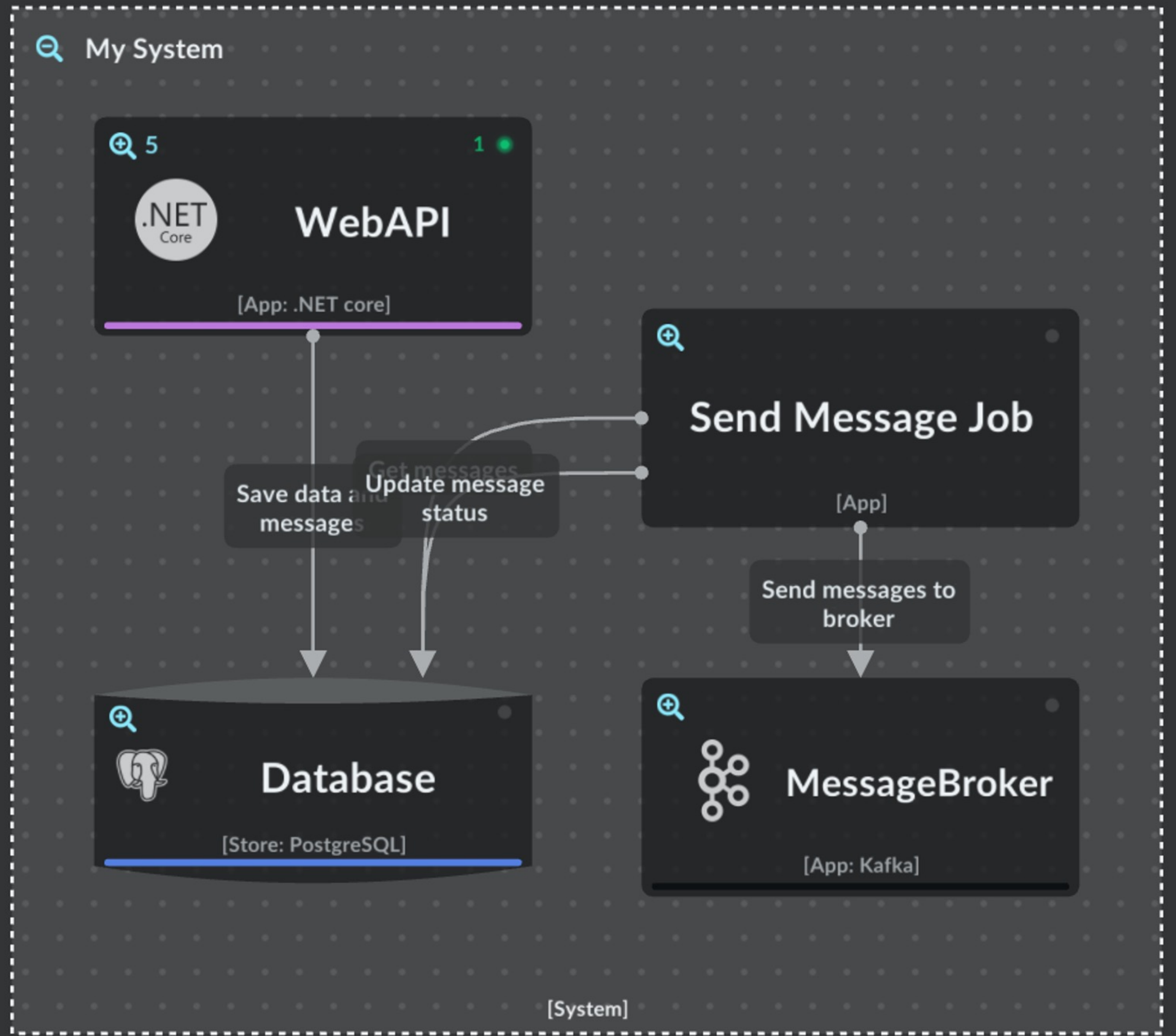


Lucidchart

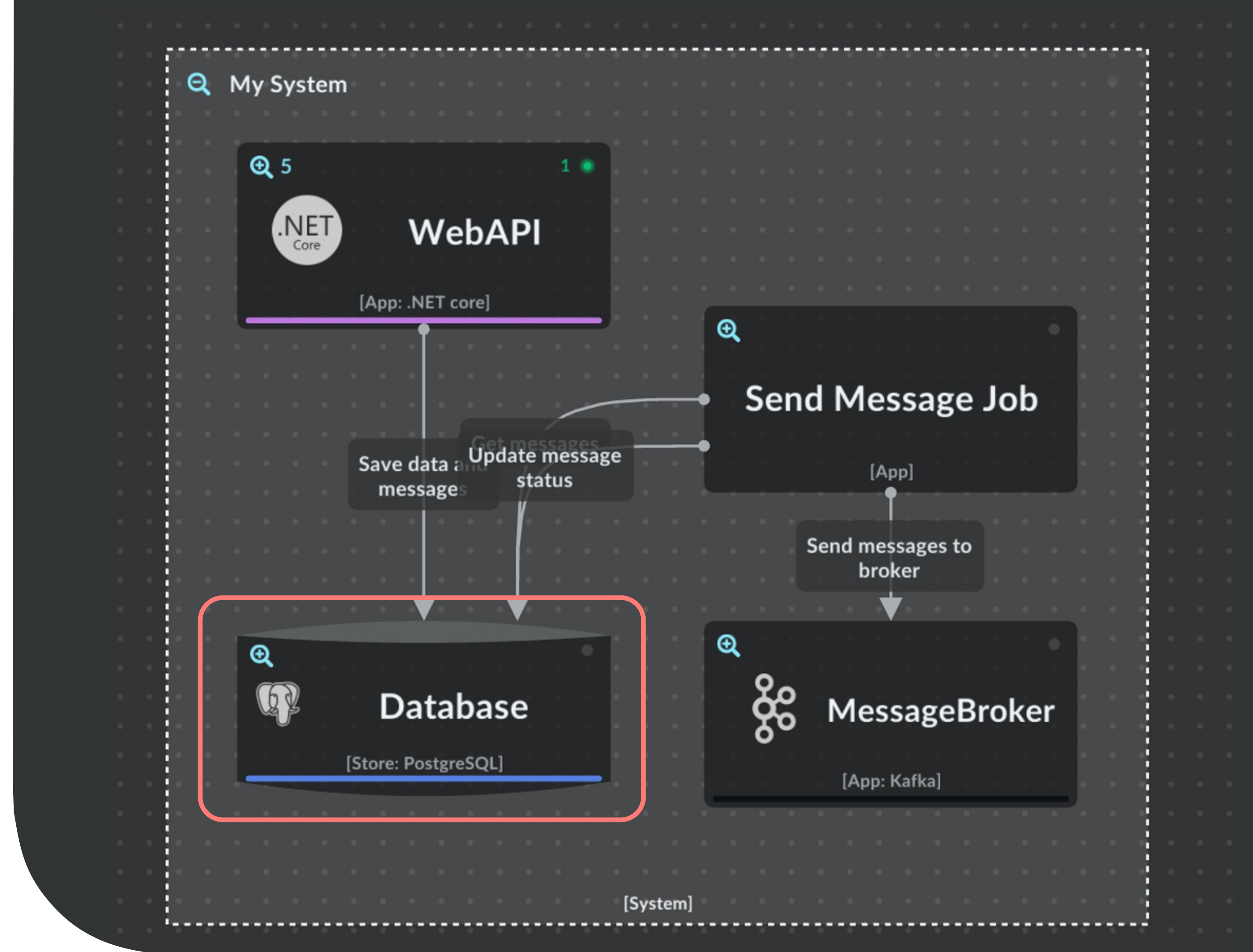
IcePanel – КОНТЕКСТЫ



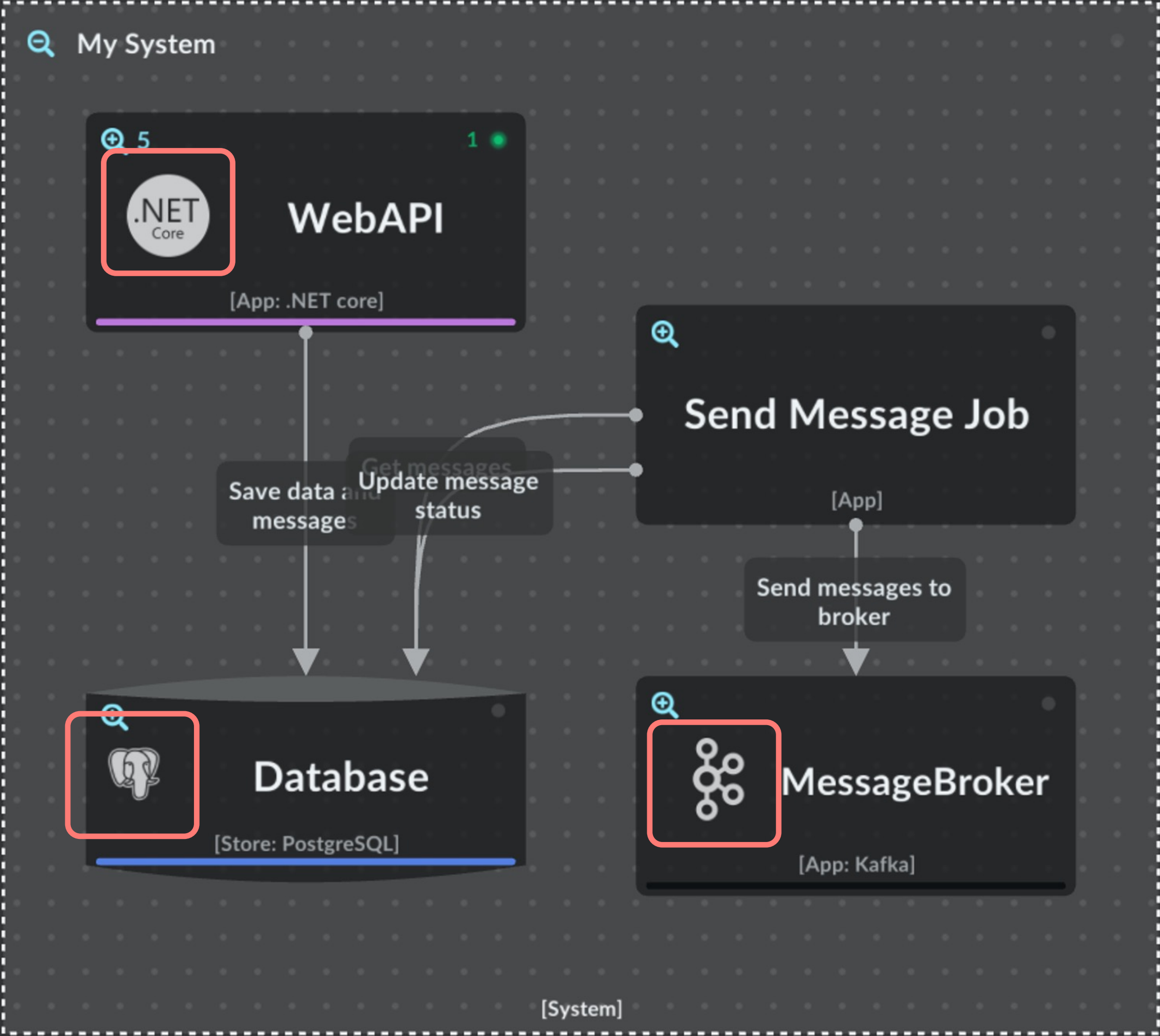
IcePanel – контейнеры



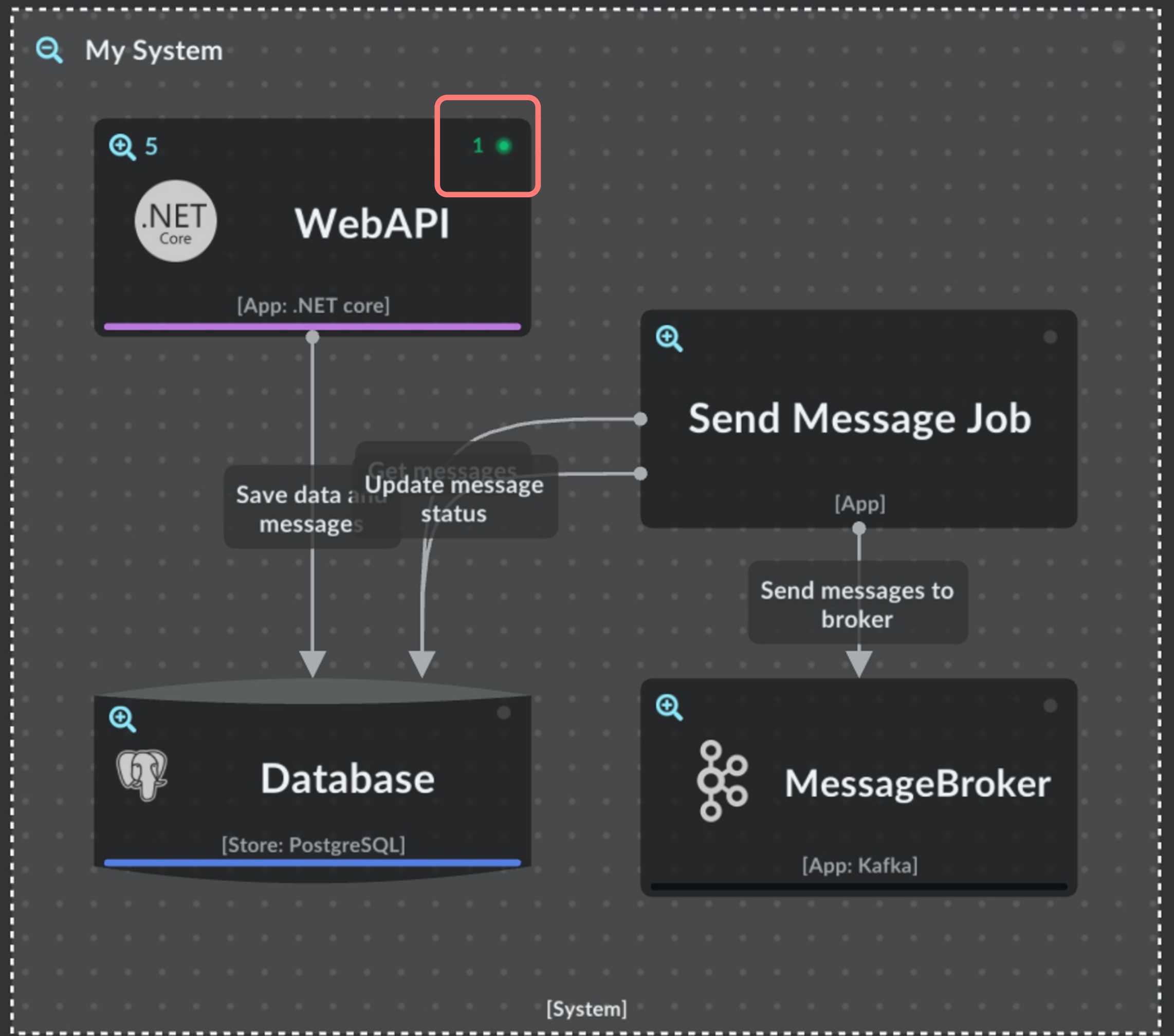
Контейнер для хранилища



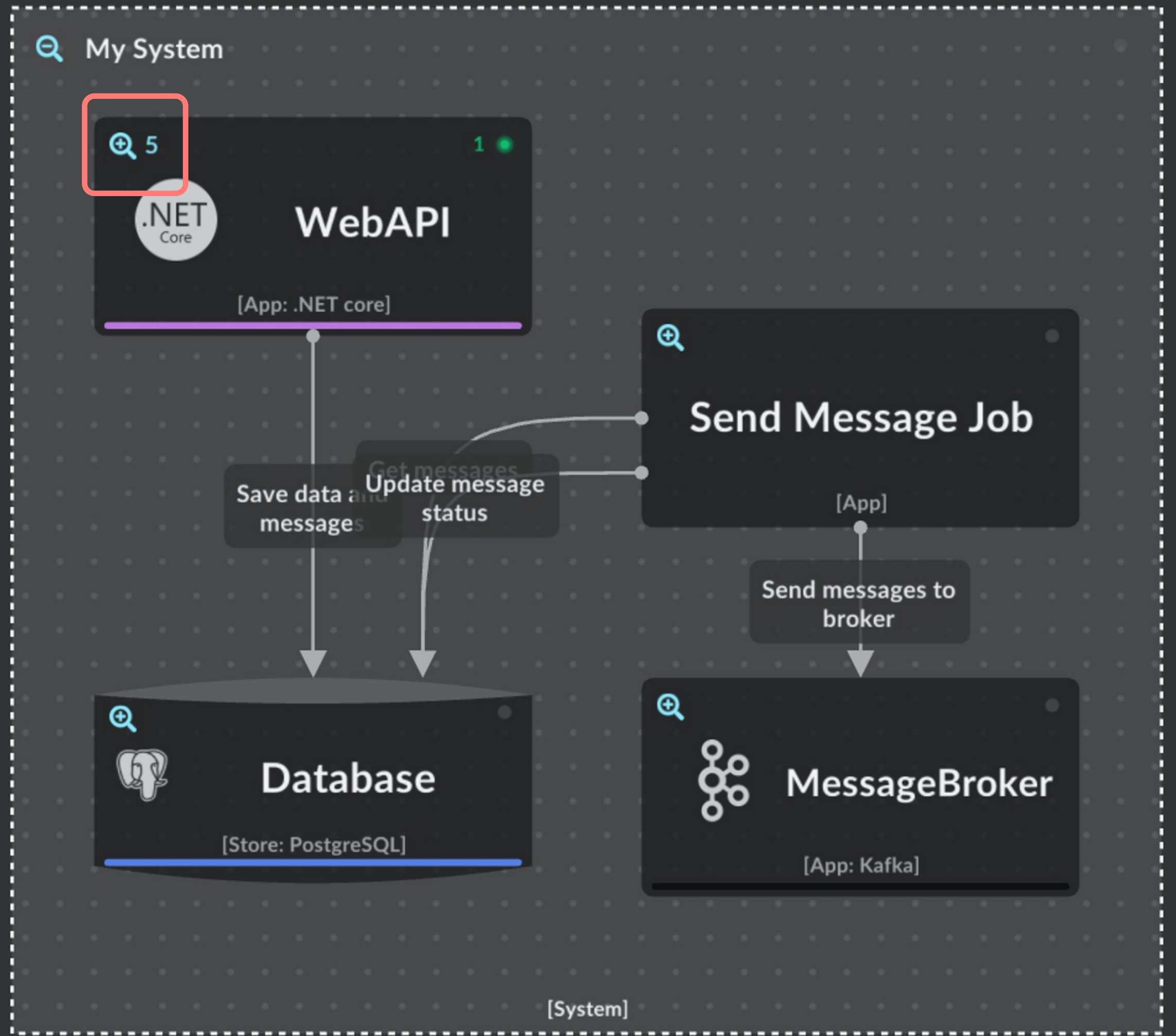
Технологии контейнеров



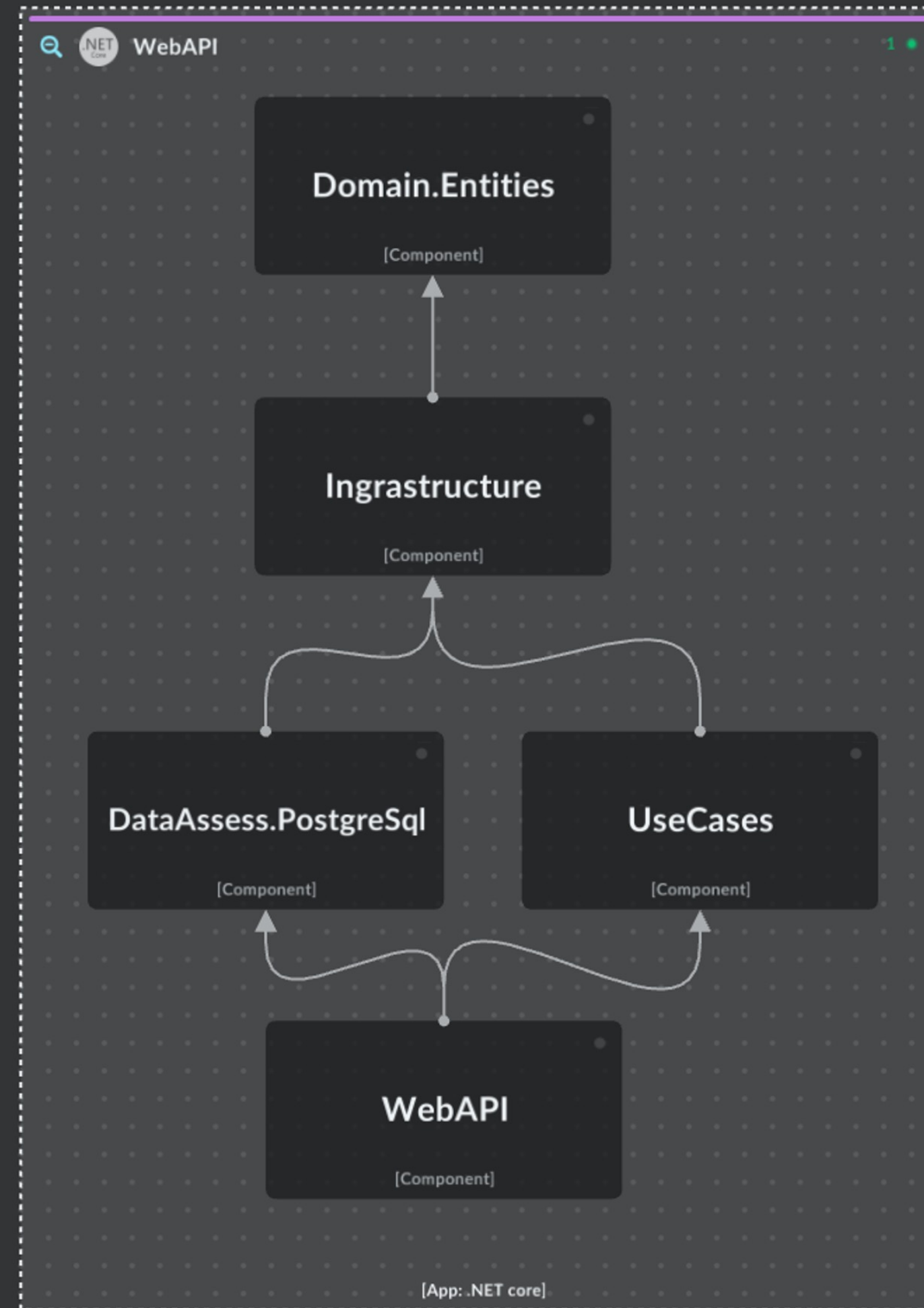
Валидация ССЫЛКИ GitLab, GitHub



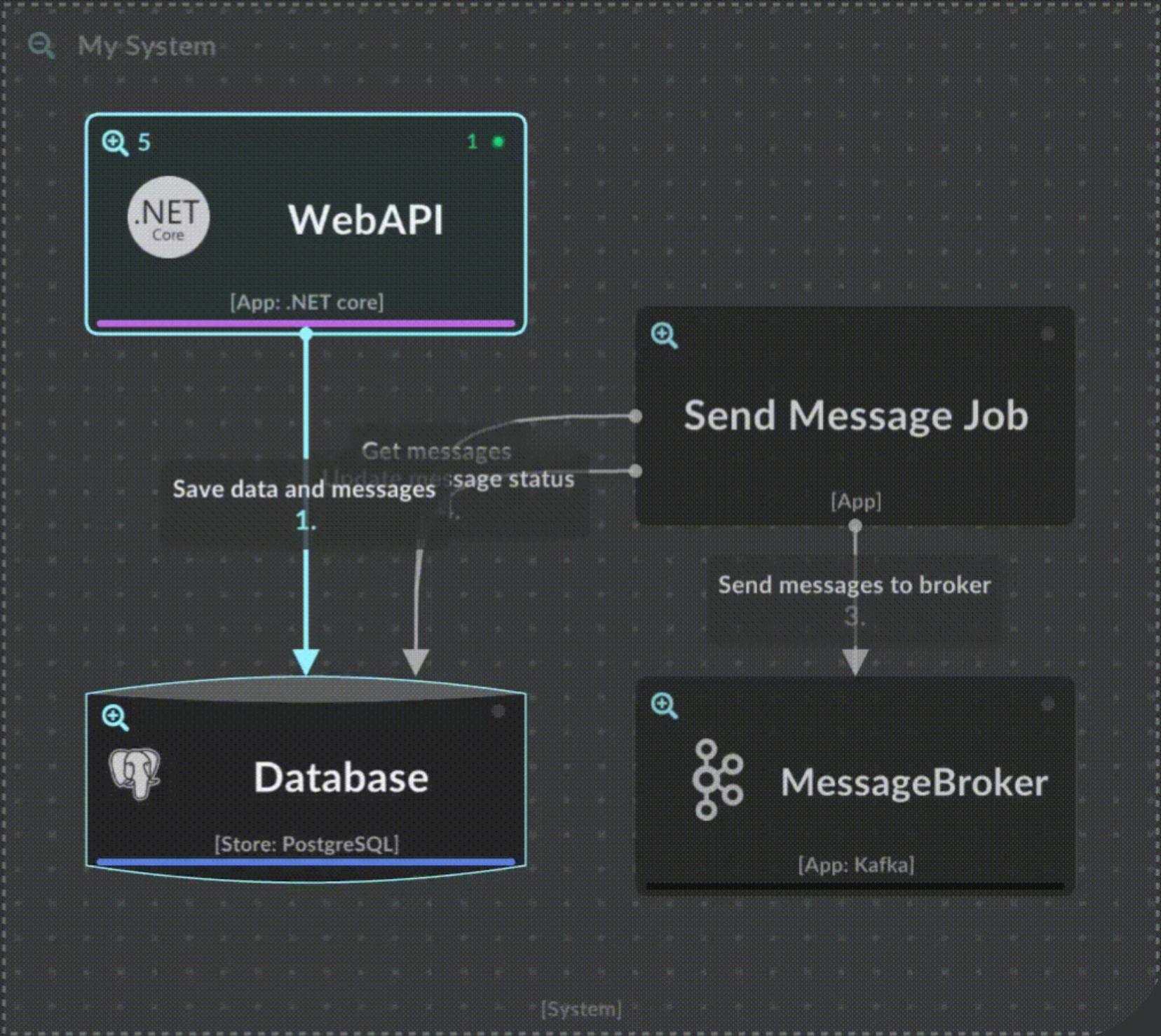
Кликабельная навигация



Компоненты



Flows



IcePanel



IcePanel

<https://icepanel.io/>



Огонь!



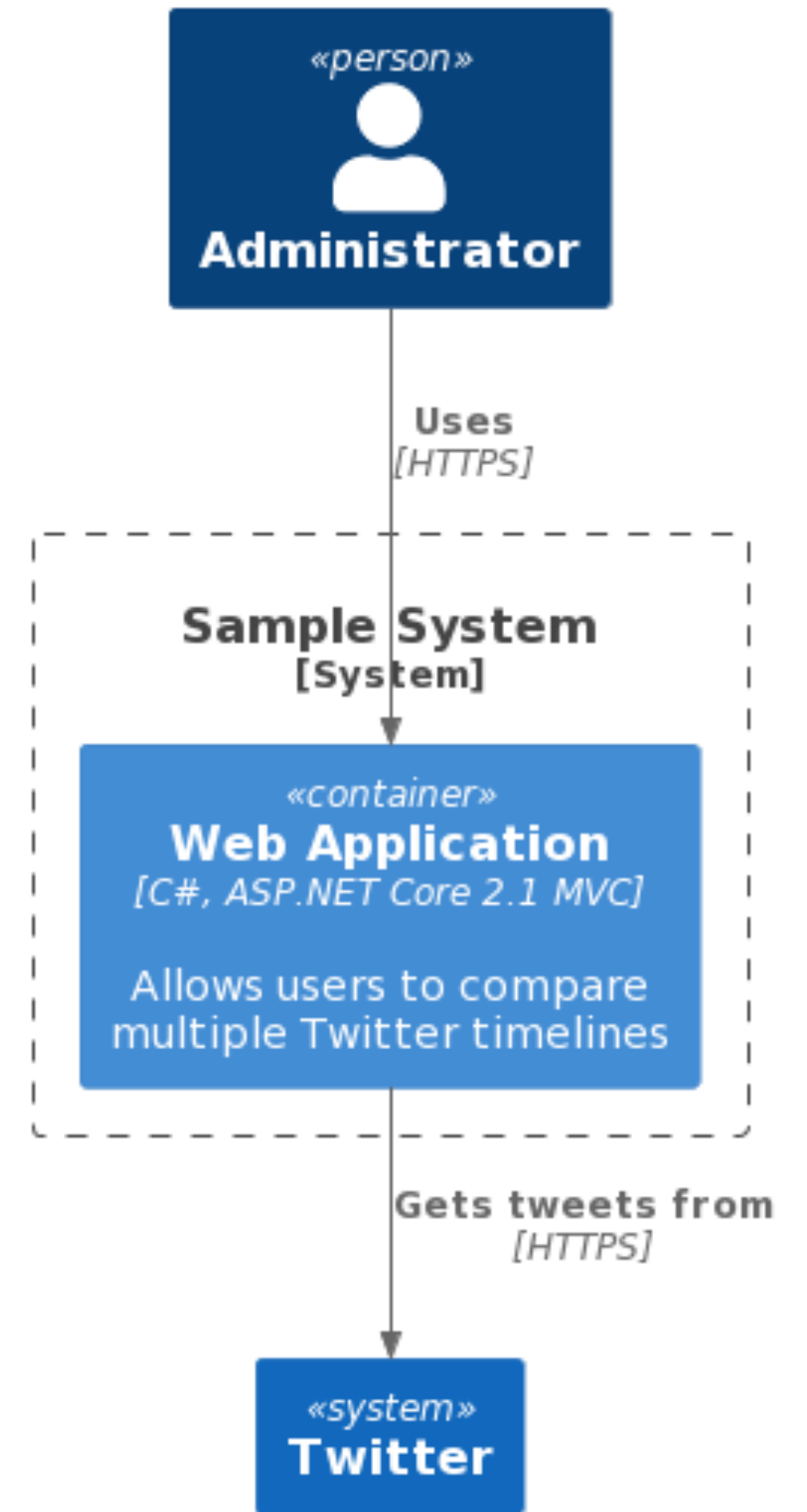
Нет диаграммы как кода

PlantUML

```
@startuml Basic Sample
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4_Container.puml

Person(admin, "Administrator")
System_Boundary(c1, "Sample System") {
    Container(web_app, "Web Application", "C#, ASP.NET Core 2.1 MVC", "Allows users to compare multiple
}
System(twitter, "Twitter")

Rel(admin, web_app, "Uses", "HTTPS")
Rel(web_app, twitter, "Gets tweets from", "HTTPS")
@enduml
```



PlantUML

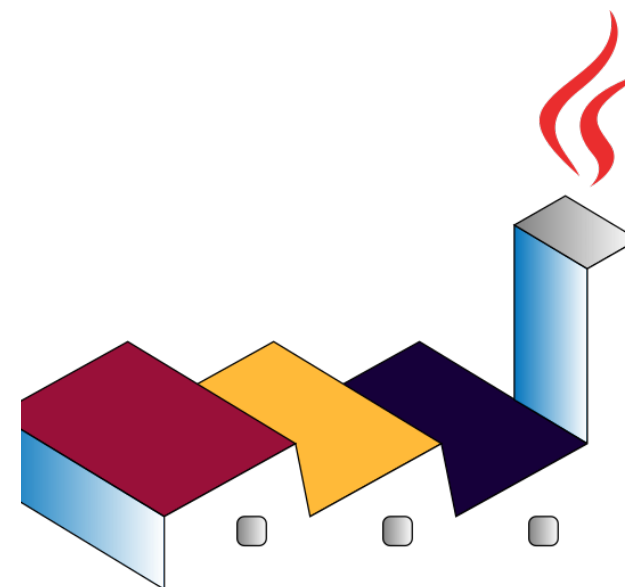


Диаграмма как код!

<https://github.com/plantuml-stdlib/C4-PlantUML>

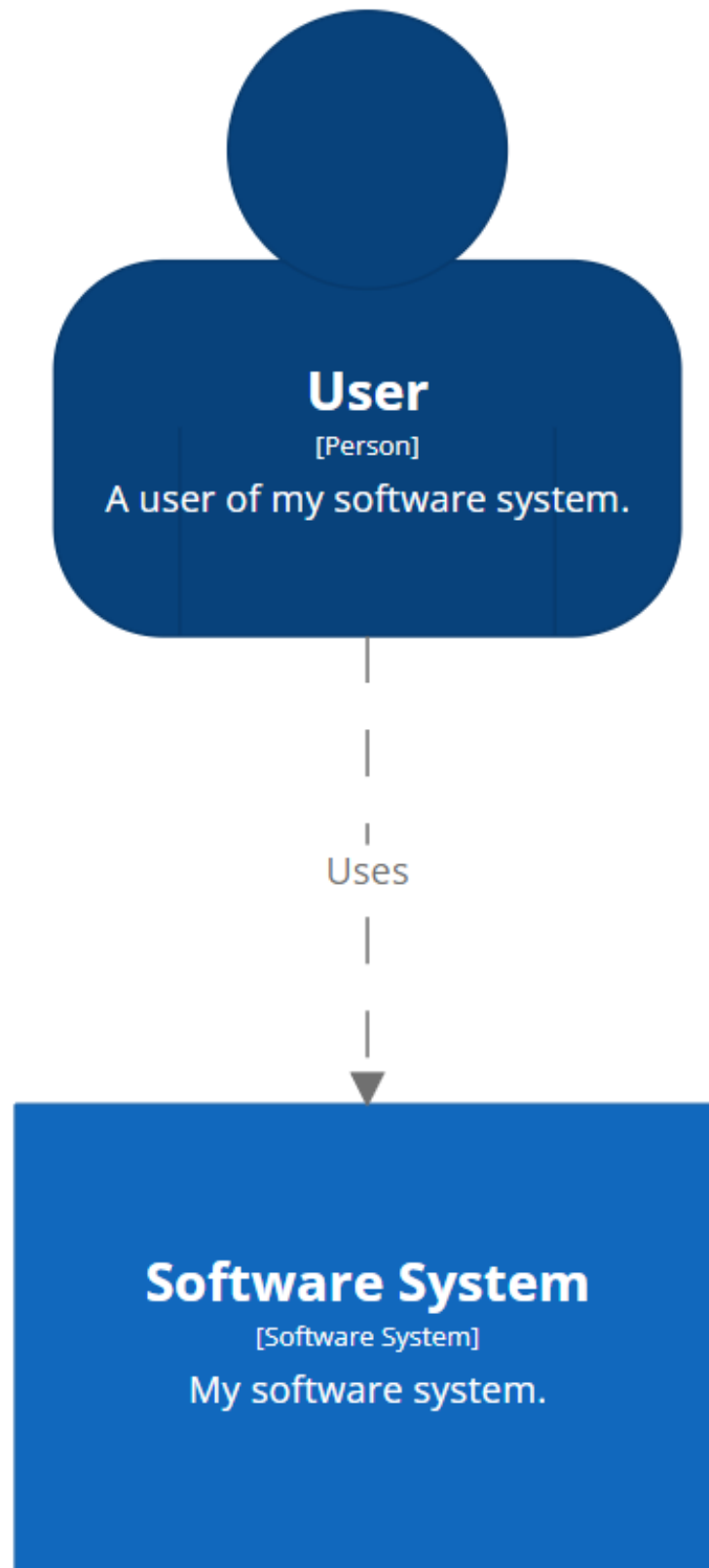


Плагин для Visual Studio Code

<https://marketplace.visualstudio.com/items?itemName=jebbs.plantuml>

Structurizr (Simon Brown)

```
1 workspace {
2
3   model {
4     user = person "User" "A user of my software system."
5     softwareSystem = softwareSystem "Software System" "My software system."
6
7     user -> softwareSystem "Uses"
8   }
9
10  views {
11    systemContext softwareSystem "SystemContext" {
12      include *
13      autoLayout
14    }
15
16    styles {
17      element "Software System" {
18        background #1168bd
19        color #ffffff
20      }
21      element "Person" {
22        shape person
23        background #08427b
24        color #ffffff
25      }
26    }
27  }
28 }
29 }
```



[System Context] Software System

Thursday, August 3, 2023 at 1:47 PM Novosibirsk Standard Time

Зоны ответственности



- Код – разработчик
- Компонент – TechLead / Software Architect
- Контейнер – Solution Architect
- Контекст – Enterprise Architect

Что знать разработчику



Матчасть

- свой стек C#, Java, ...



Алгоритмы и структуры данных

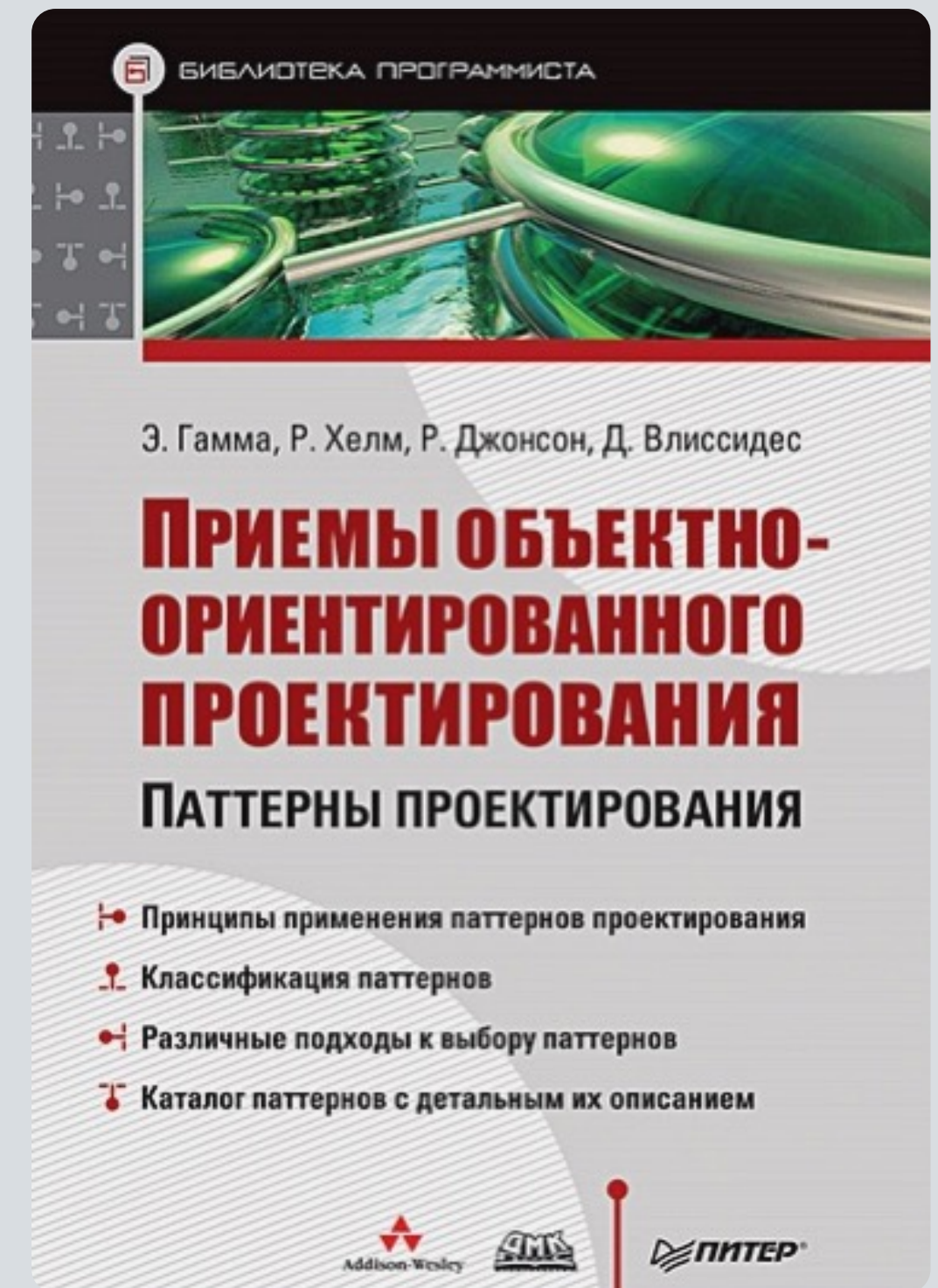


Базы данных

SOLID



Паттерны проектирования



Компоненты



- Выделенной роли Software Architect/Техлид обычно нет
- Команда / Тимлид / Лидер в раздевалке (техлид)
- Database Administrator (DBA)

Что нужно знать техлиду



Расслоение

- Чистая архитектура, Луковая архитектура, Порты и Адаптеры
- Архитектурные тесты и метрики



Best Practices для каждого слоя

- Domain: Rich Model, Anemic Model, Domain Events
- Infrastructure: DataAccess (Repository over ORM)
- Application: Vertical Slice Architecture, Aspect Oriented Programming



Модульный монолит



Enterprise паттерны

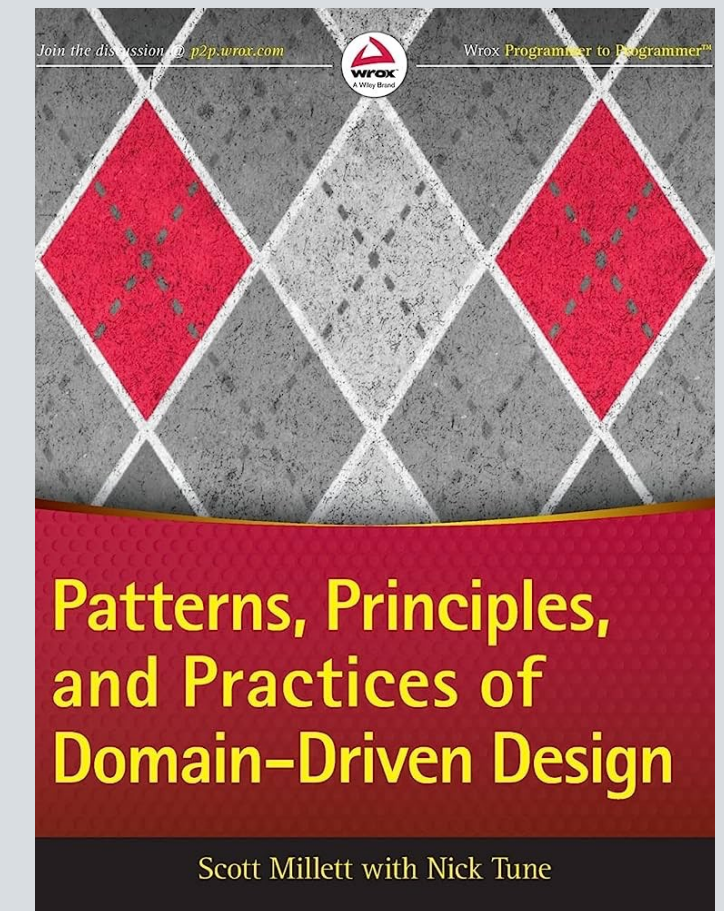
- Оптимистическая и пессимистическая блокировка
- Repository, Unit of Work
- Спецификация

Материалы по теме КОМПОНЕНТЫ



<https://clck.ru/345J3Z>

Для устойчивой психики



Контейнеры - Solution Architect

Соответствие системы своему целевому назначению (SAFE)

Один стек технологий (Java, .NET, Python, ...)

Распределенные отказоустойчивые системы

Хранилища данных (один представитель каждого класса)

- SQL (Postgres)
- Document (Mongo)
- KeyValue (Redis)
- Колоночные (ClickHouse)
- Elastic, ...

Message Brokers

- RabbitMQ, Kafka

Протоколы

- HTTP, WebSocket, gRPC

Event Sourcing, CQRS

Паттерны интеграции

Solution Architect



Александр Поломодов,
Как прокачаться в проектировании ПО
<https://tellmeabout.tech/software-design-books-743be52e4c71>

Enterprise Architect



Разработка технологической концепции, стратегии и дорожной карты портфеля (SAFe)



Опыт в конкретной компании

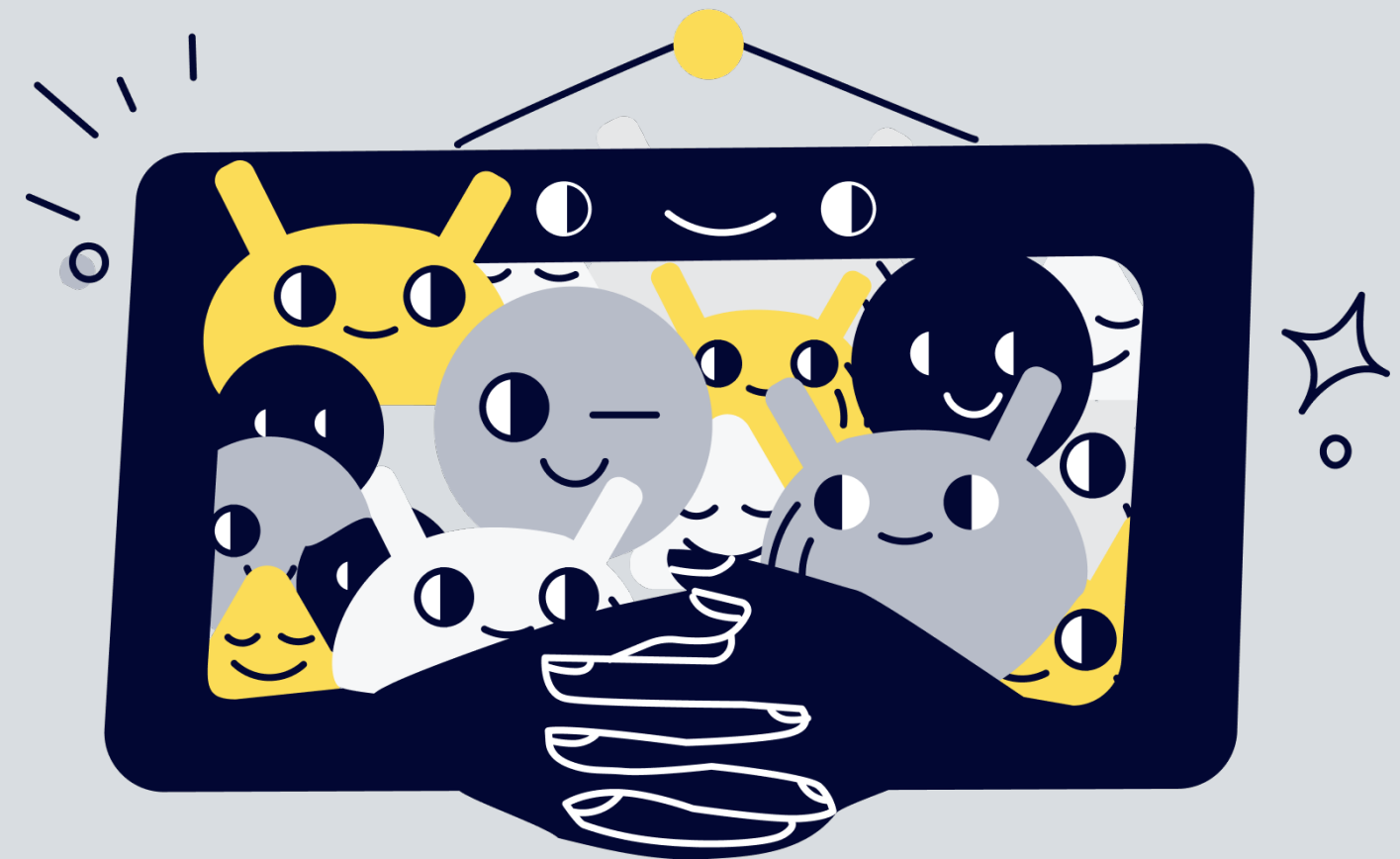


Опыт в предметной области (страхование, логистика, ...)



https://otus.ru/lessons/arch_ent/

О чем поговорили



Основные диаграммы C4

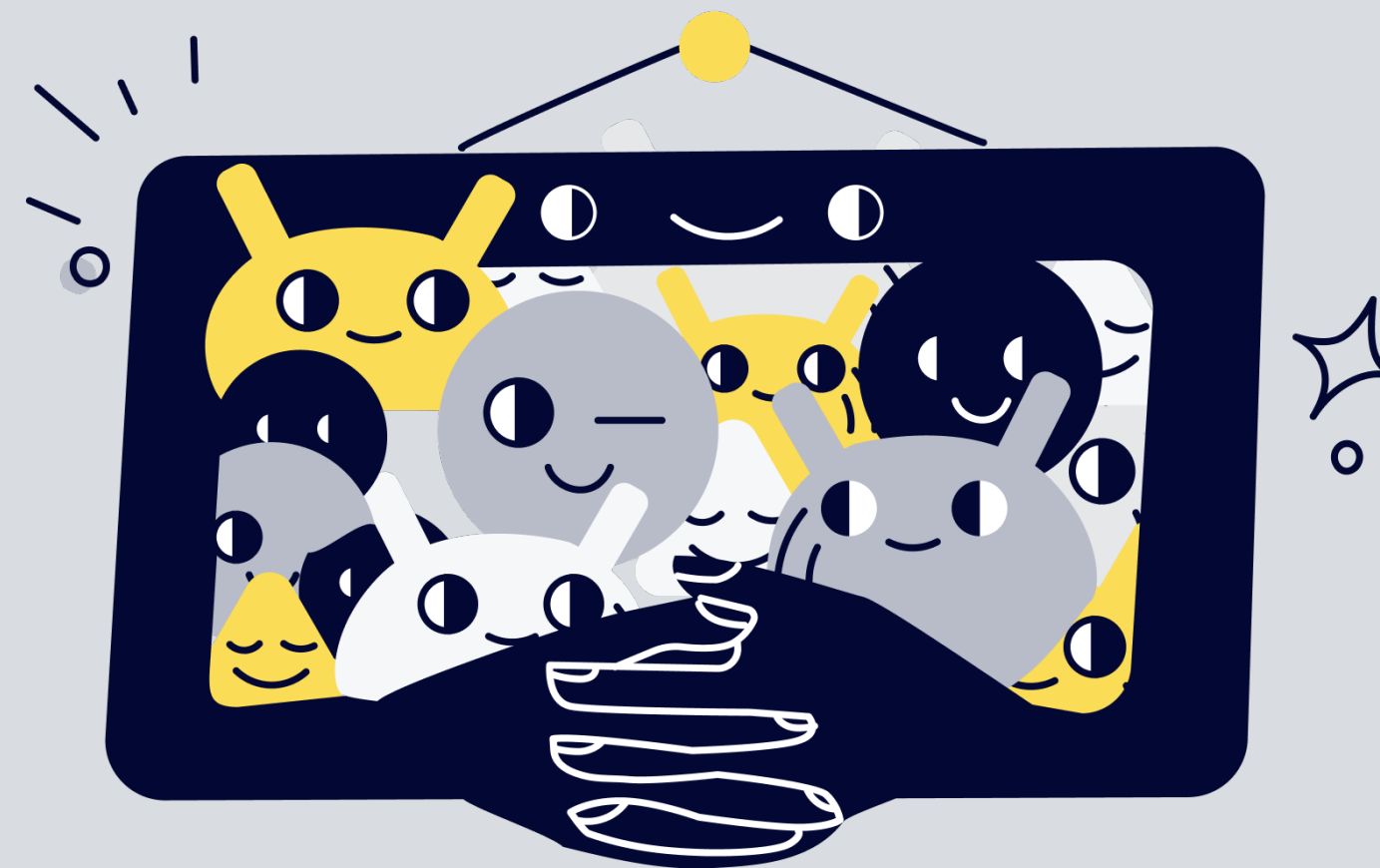
- Контекстов
- Контейнеров
- Компонентов
- Кода



Дополнительные диаграммы C4

- Ландшафта
- Развертывания
- Динамическая диаграмма

О чем поговорили



Инструменты

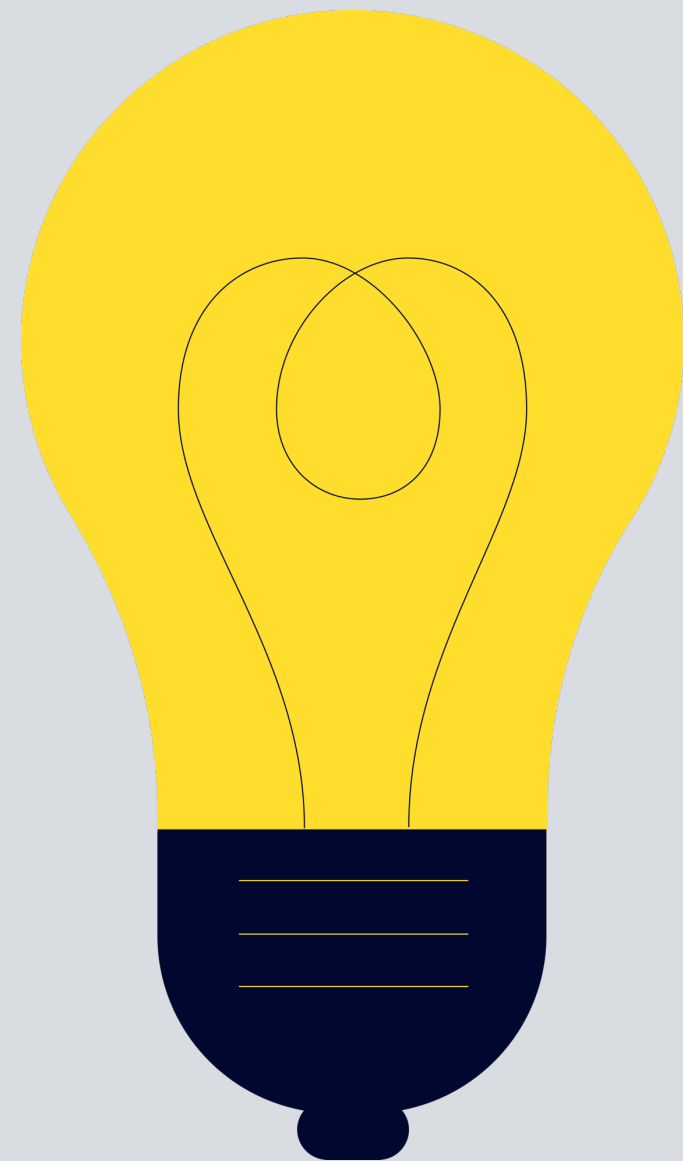
- Draw.io
- IcePanel
- PlantUML, Structurizr



Развитие

- Код – разработчик
- Компонент – TechLead, Software Architect
- Контейнер – Solution Architect
- Контекст – Enterprise Architect

Главная мысль



С4 – не только диаграммы, но и:

- Разделение зон ответственности
- Грейды
- Планы развития

Solution Architect



Отправная точка
<https://c4model.com/>



Архитектурный репозиторий C4 model
<https://youtu.be/cTM6M6BbuMs>



Денис Цветцих

Тинькофф | DevBrothers

 @den_tsvettsikh

 den.tsvettsih@yandex.ru

 <https://github.com/denis-tsv>

**Спасибо
за внимание!**

Есть вопросы?