

Base Application infrastructure

Матвей Попов

План

- Из каких компонент состоит приложение?
- Аутентификация и авторизация
- Хранение данных
- Как организовывать архитектуру приложения?
- Как тестировать приложение?

**Из чего же состоит наше
приложение?**

- Framework(Ktor)
- Зависимости
- BuildManagementTool(Gradle)
- Наш код

Как наш код выглядит?

World

IN(http)

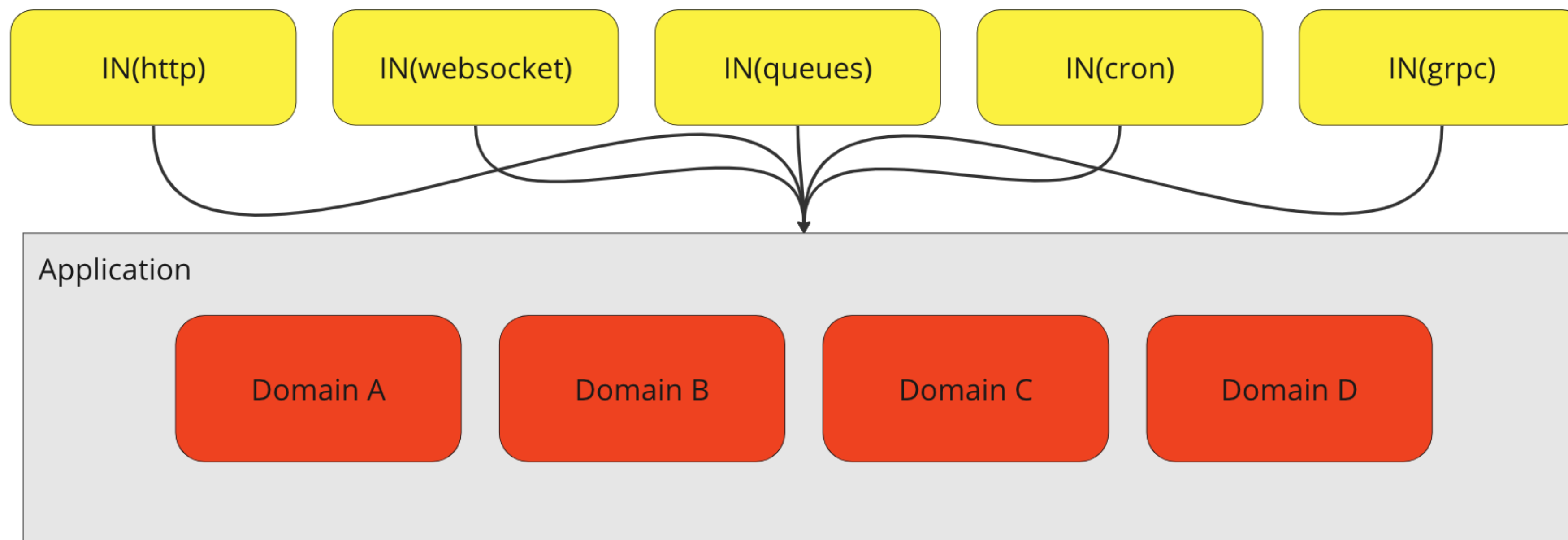
IN(websocket)

IN(queues)

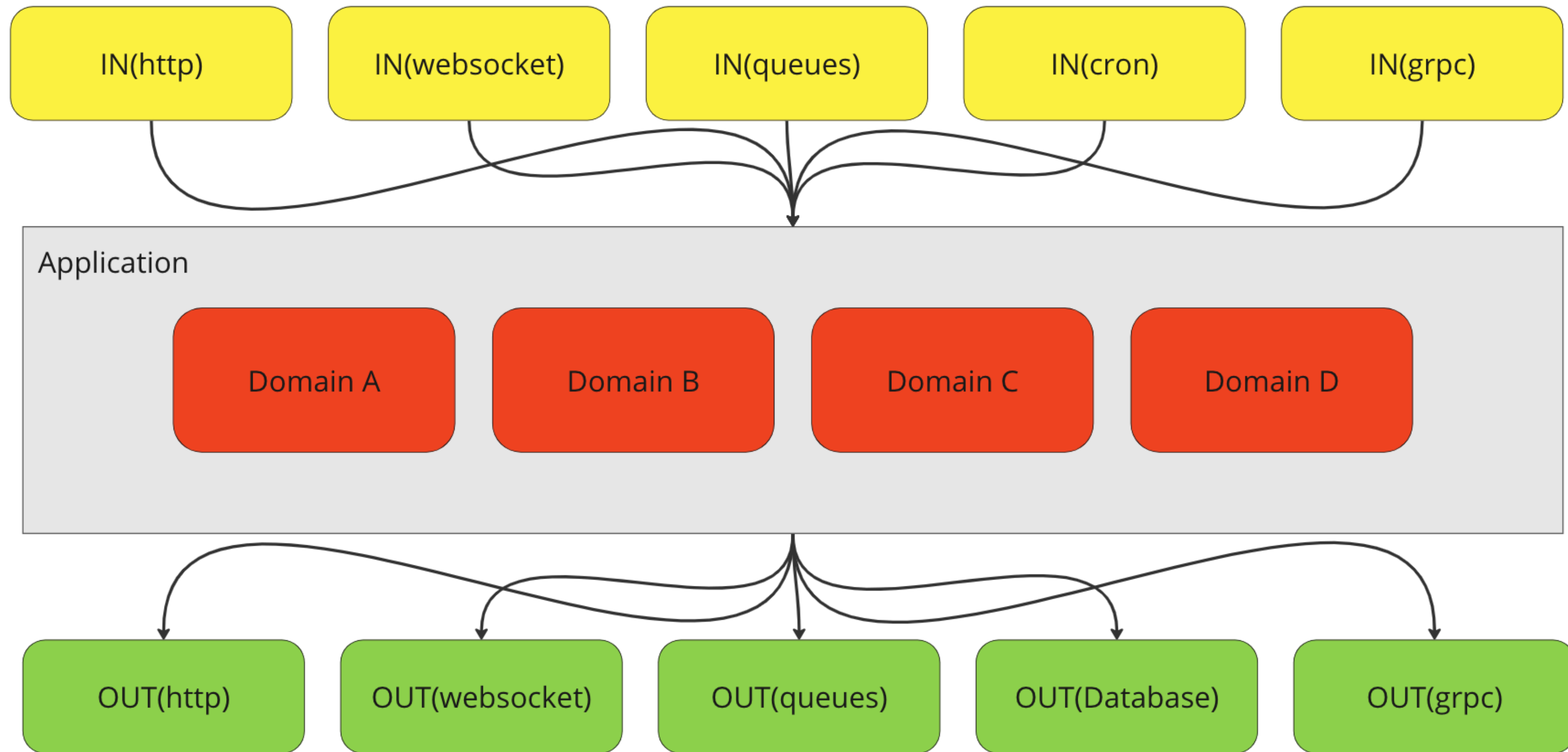
IN(cron)

IN(grpc)

World



World



Что мы всегда делаем?

1. Принимаем данные
2. Манипулируем с данными
3. Отдаем данные

Что мы делаем с входящими данными?

- Преобразуем в необходимый формат
- Валидируем данные
- Идентифицируем источник данных
- Аутентифицируем источник данных
- Авторизуем источник данных

Форматы данных

Виды форматов

- Бинарные
- Не бинарные

- JSON
- XML
- YAML
- Protobuf
- Avro

Валидация данных

- Проверяем соответствие некоторым бизнес требованиям
- Проверяем корректность данных
- Проверяем целостность данных

Идентификация источника

**Цель - понять, кто “стучится в
наше приложение”**

Примеры

- Login
- Email
- Телефон
- Какой-то токен

**Любая сущность по которой
можно однозначно определить
клиента**

Аутентификация

**Цель - проверить клиента на
подлинность**

Необходимо проверить:

- Клиент представлен в нашей системе
- Клиент вводит валидные данные
- Клиент пытается представиться так как мы его знаем

**Существуют 3 фактора,
которые задействуются в
процессе аутентификации**

1. Знание

2. Владение

3. Свойство

Виды аутентификации

- Однофакторная
- Многофакторная

Авторизация

**Цель - наделение
пользователя определенными
правами**

Примеры

- Обычный пользователь с доступом только до своих данных
- Пользователь с доступом ко всем данным(администратор)

**Какие есть способы
реализации?**

Сессии

1. Клиент отправляет идентификационные данные
2. Сервер проверяет данные
3. Сервер создает сессию и генерирует `cookie`
4. Клиент отправляет `cookie` при каждом запросе
5. Сервер валидирует и выдает доступ
6. По истечению времени или каких-то условий `cookie` очищается

Cookie

Небольшой **фрагмент** данных, отправленный клиентом и **хранимый** на стороне клиента.

Токены

1. Клиент отправляет идентификационные данные
2. Сервер проверяет данные
3. Сервер генерирует, подписывает и “зашивает” данные для идентификации пользователя
4. Клиент сохраняет токен на своей стороне и использует его при каждом запросе
5. Сервер проверяет токен

Что мы будем использовать?

JWT - JSON Web Token

JWT

- Открытый стандарт для создания токенов, основан на формате JSON
- Поддерживает симметричную и асимметричную подпись
- Стоит из 3-х частей: **headers**, **payload**, **signature**

**Как мы будем использовать
JWT?**

С помощью Ktor Plugin Authentication

**Теперь мы должны данные
куда-то отдать**

Хранение данных

Архитектура и организация кода

Тестирование

Домашнее задание

- Добавить аутентификацию и авторизацию
- Сделать в отдельном Pull Request
- Добавить меня в reviewers
- Подробное описание в отдельном файле

Материалы

- Про MVC: <https://clck.ru/36AoaC>
- Про аутентификацию: <https://clck.ru/36Aod6>
- Про слой доступа к данным: <https://clck.ru/36AogQ>
- Про гексогональную архитектуру: <https://clck.ru/36Aojc>

- Про unit тесты: <https://clck.ru/36Aon2>

Спасибо за внимание!