next conf

C4 model в Structurizr

К архитектуре через код

Евгений Фатеев

Uzum Bank

Евгений Фатеев

Uzum Bank

Руководитель команд разработки. Реализовал Learning To Rank подход на Ruby и Elasticsearch. Автор курсов по Ruby и Rails в Thinknetica. LlamaIndex энтузиаст.

- github.com/psylone
- t.me/psylone

План доклада

Цель - изучить инструменты Structurizr и процесс внедрения подхода «Архитектура через код»

C4 model

Кратко об основных элементах модели, её достоинствах и недостатках

Structurizr

О подходе «Архитектура через код», инструментах Structurizr и DSL

Процессы

О ценности процессов и особенностях их построения

Адаптация

О целевом решении и его пошаговой интеграции

Зачем нам модель?

nextconf

Модель - упрощенное представление о предмете

Абстракции

Без модели абстракции перемешиваются на разных уровнях

Декомпозиция

Желание отобразить все элементы на одной диаграмме приводит к «God Object»

Контекст

Без определенных правил одни и те же элементы обозначают совершенно разные вещи

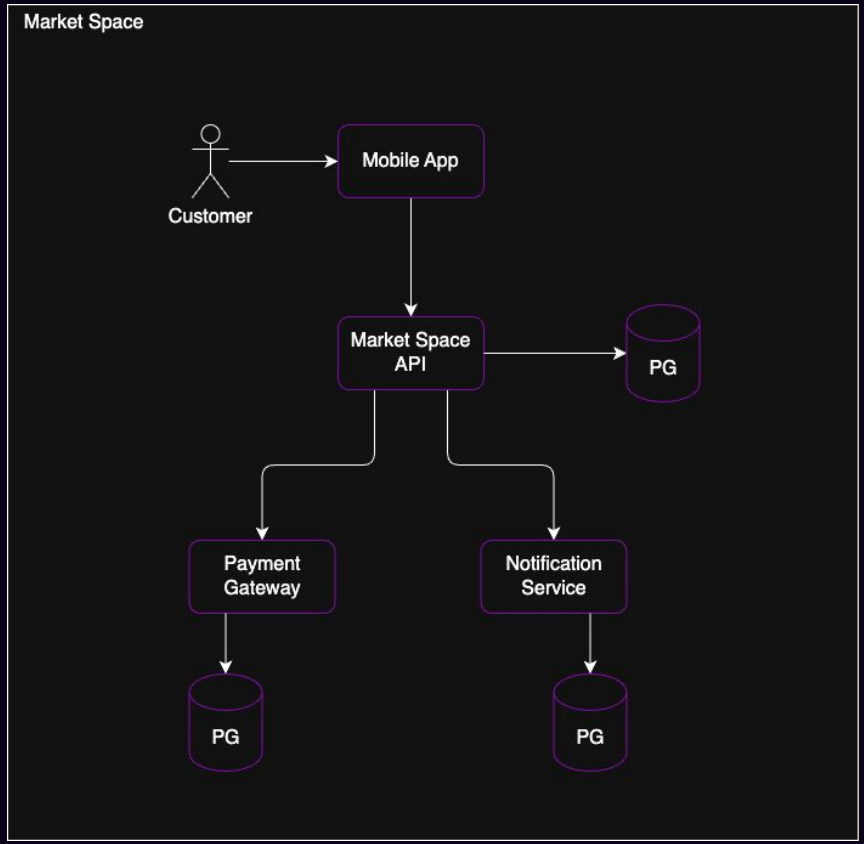
Поддержка

Без инструментов когнитивная нагрузка растет пропорционально времени поддержки и обратно пропорционально желанию

Прямоугольники с текстом

и стрелками

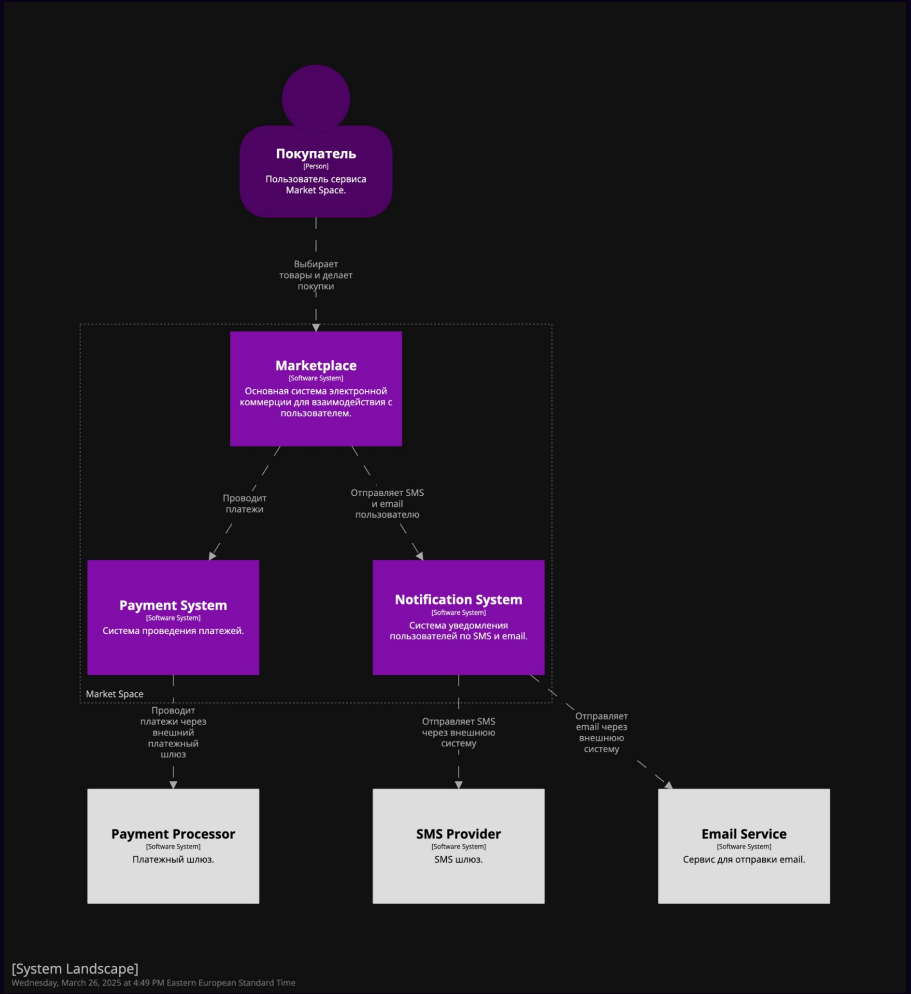
- Границы системы не определены
- Разные уровни абстракции
- Плохо масштабируется



C4 model

Ландшафт систем

- Ясные границы системы
- Один уровень абстракции
- Адекватное масштабирование
- Структура каждого элемента явно определена



C4 model

Абстракции

- **Software system** - логический элемент, обозначающий определенный функционал (e-commerce, ДБО банка, система платежей, система уведомлений)
- **Container** - конкретный сервис (SPA, API, database)
- **Component** - логический элемент, обозначающий компонент внутри сервиса (модуль, пакет, коллекция классов, реализующих интерфейс)
- **Code** - реализация компонента (класс, интерфейс, объект)

C4 model

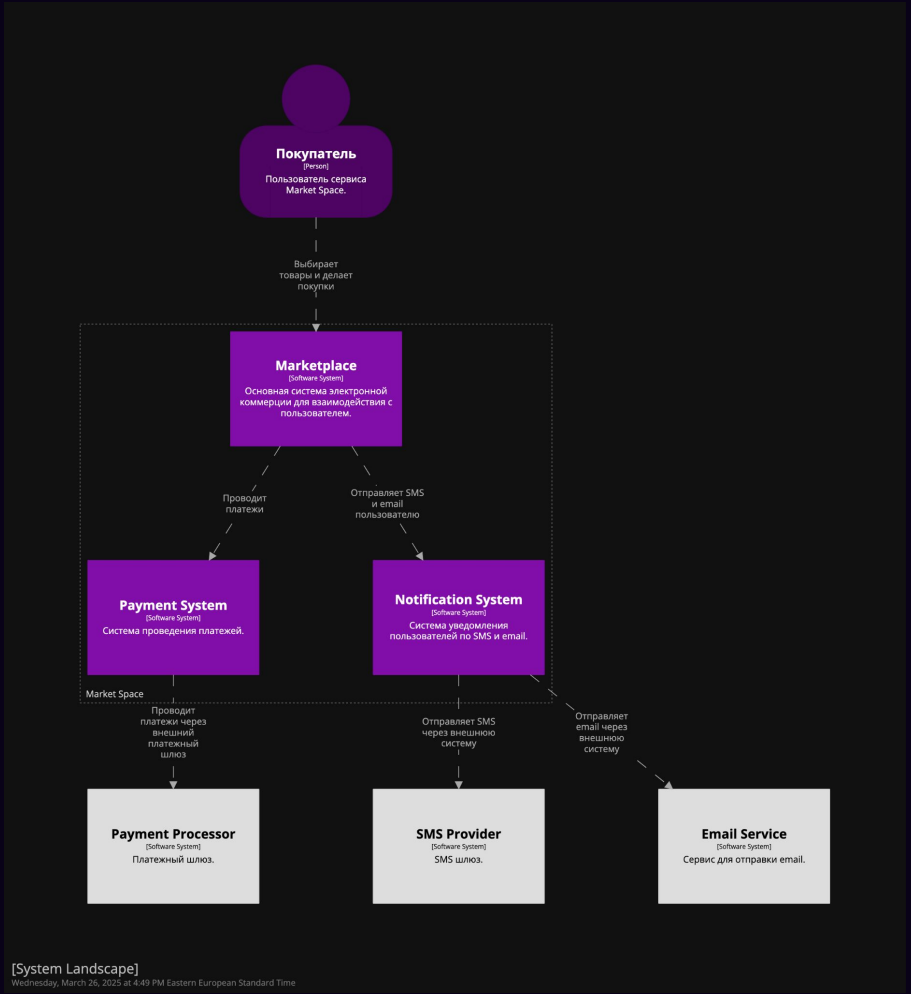
Элементы модели

- Person
- Software System
- Container
- Component
- Relationship

C4 model Диаграммы

System landscape

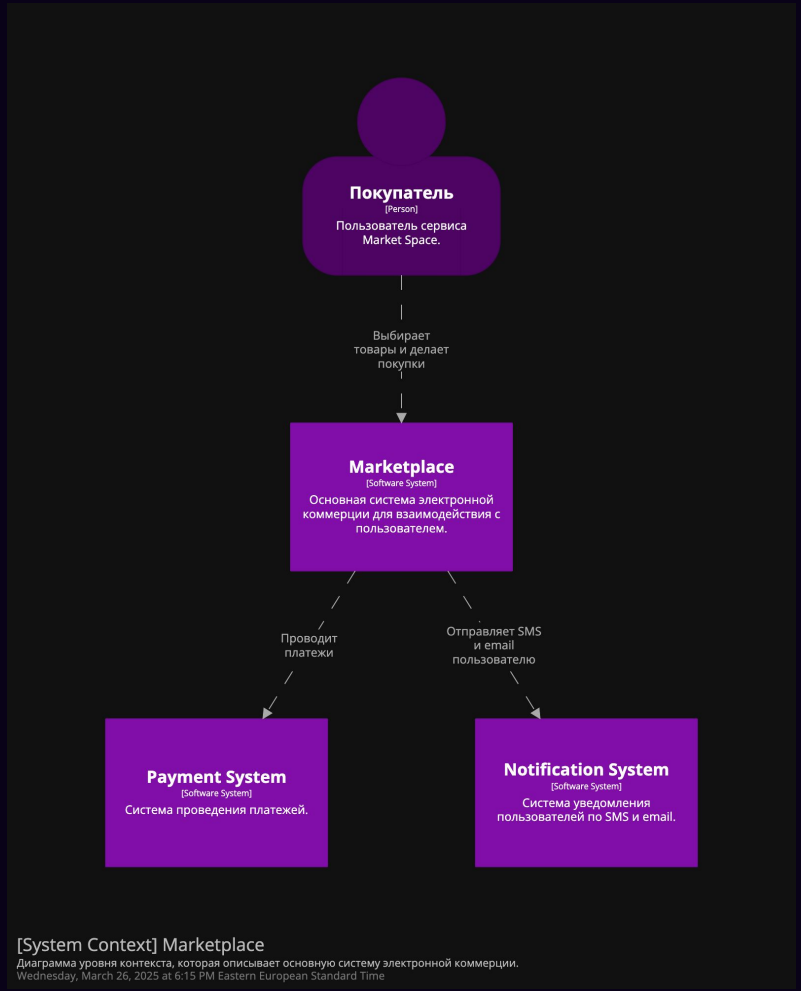
Отображает несколько элементов Software System. Позволяет посмотреть на архитектуру организации в целом.



C4 model Диаграммы

System context

Отображает конкретный элемент Software System и связанные с ним элементы.

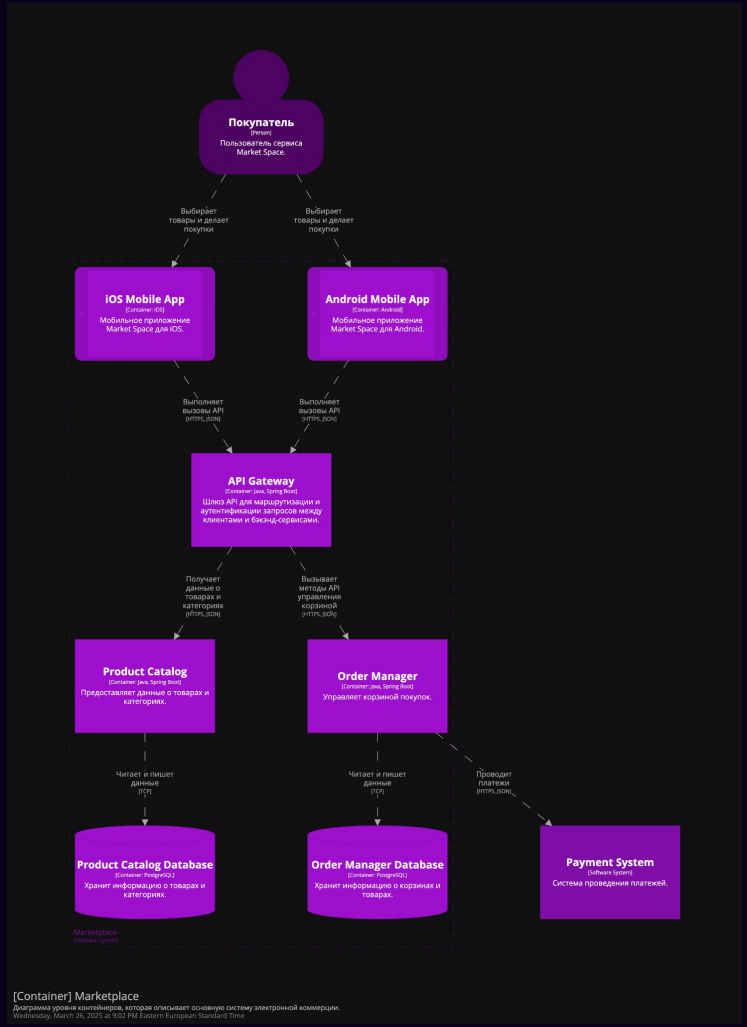


C4 model
Диаграммы

Container

Отображает сервисы для выбранного элемента Software System (mobile app, backend API, database).

Сервис может взаимодействовать с другим элементом Software System.



[Container] Marketplace
Диаграмма уровня контейнеров, которая описывает основную систему электронной коммерции.
Wednesday, March 26, 2025 at 9:02 PM Eastern European Standard Time

C4 model Диаграммы

Component

Отображает реализацию для выбранного элемента Container (модуль, пакет, реализация интерфейса).

Может генерироваться из исходного кода.

[Component] Marketplace - API Gateway

Диаграмма уровня компонентов, которая описывает сервис API Gateway.

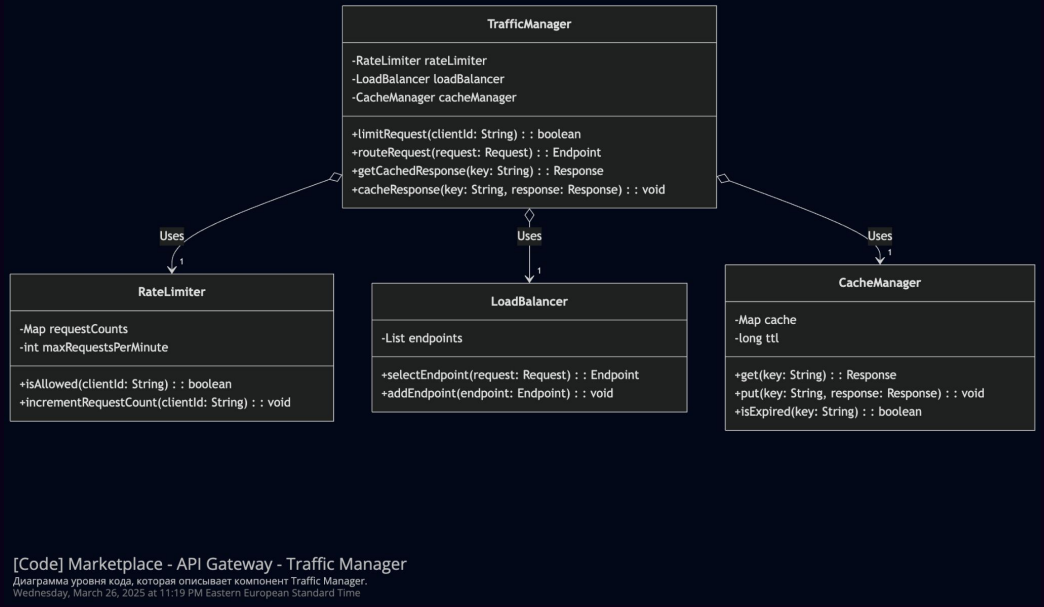
Wednesday, March 26, 2025 at 9:28 PM Eastern European Standard Time

C4 model
Диаграммы

Code

Отображает реализацию для выбранного элемента Component. Представляется в виде UML class diagram или entity relationship diagram.

Рекомендуется генерировать из исходного кода.



C4 model

Недостатки

- Все не так просто как изначально может показаться
- Больше подходит для статического отображения
- Без процессов модель устаревает и превращается в прямоугольники с текстом

Достоинства

- Минимальное количество сущностей
- Структура и системный подход из коробки
- Возможность разговаривать с разной аудиторией
- Возможность обсуждения на одном уровне абстракции
- Следование принципу «Анализ и синтез»

Structurizr

Архитектура через код

- Текстовый интерфейс
- Быстрота изменений
- Возможность использовать Source Code Manager (SCM)
- Относительно простая программная генерация
- Более независимые инструменты
- Возможность описывать валидации и создавать запросы (diagramming vs modelling)

Structurizr

Инструментарий

- Structurizr for Java - Java плагин, позволяющий генерировать описание модели
- DSL - Надстройка над Java плагином, предоставляющая DSL
- Lite - Приложение для рендеринга и просмотра моделей в браузере
- On-premises - Энтерпрайз решение, которое можно развернуть в локальной среде
- Cloud - Облачное решение
- Static site generator - Генератор статического сайта с диаграммами
- CLI - Команда для работы из терминала

Structurizr

DSL

- `workspace` - корневой элемент описания модели
- `model` - элементы модели (Software System, Container, Component, Code)
- `views` - отображение диаграмм
- Возможно добавить динамическую часть в виде скриптов на Groovy, Kotlin, Ruby, JS
- Есть возможность наследования от других `workspace`

```
workspace "Market Space" "Workspace описывает архитектуру проекта Market Space - сервиса электронной коммерции." {
  !identifiers hierarchical

  model {
    group "Market Space" {
      marketplace = softwareSystem Marketplace "Основная система электронной коммерции для взаимодействия с пользователем."
      paymentSystem = softwareSystem "Payment System" "Система проведения платежей."
    }

    // Relationships

    marketplace -> paymentSystem "Проводит платежи"
  }

  views {
    systemLandscape MarketSpaceLandscape {
      include *
      autolayout tb
    }

    styles {
      element "Software System" {
        background #820ea9
        color #ffffff
      }
    }
  }
}
```

Structurizr

Использование

- Редактирование
  - Текстовый редактор
  - On-premises DSL editor
  - Программная генерация (Java, .NET, Python, Go, PHP, TypeScript)
- Рендеринг
  - Lite
  - On-premises
  - Cloud
  - CLI (экспорт в PlantUML, Mermaid)
- VS Code extension for C4 DSL Models
- IntelliJ plugin for the Structurizr DSL

Structurizr

Использование

- `!docs` - документация сервисов тесно связана с архитектурой
- `!adrs` - записи об изменениях архитектуры - Architectural Decision Records (ADR) позволяют отслеживать историю и контекст
- Все это не будет работать без явно определенных процессов

Процессы

nextconf

Процесс — это карта без которой все идут наугад, часто в разные стороны

Синхронизация

Позволяет командам работать в едином стиле

Изменения

Фиксируется в виде текста и схем, которые можно положить в Git

Метрики

Понимание процесса позволяет выставить метрики

Поиск проблем

Метрики позволяют оперативно находить бутылочные горлышки

Процессы

Диаграммы в виде изображений

- Простой формат работы
- Простое внедрение
- Нет валидаций элементов
- Отсутствует возможность делать запросы к диаграмме
- Сложнее поддержка, например - управление ссылками между элементами, хранение истории изменений
- С высокой вероятностью vendor lock-in
- Только UI, только хардкор

Процессы

Диаграммы в виде кода

- Нужно учить DSL
- Сокращение времени на редактирование
- Более широкий набор независимых инструментов
- Возможности программной генерации, валидации, запросов к диаграмме
- Системный подход с самого начала

Процессы

nextconf

То что мы не измеряем - тем мы не управляем

Просмотры

Количество просмотров страниц с архитектурными диаграммами

Запросы

Количество запросов с вопросами об архитектуре

Lead Time

Среднее время выполнения этапа построения архитектуры

NPS-опрос

Индекс Net Promoter Score для оценки лояльности пользователей

Процессы

Баланс во всех вещах

- При разработке и внедрении процесса важно поддерживать баланс между гибкостью и детализацией процесса
- Слишком сложные процессы труднее фиксировать, внедрять и поддерживать
- Отсутствие деталей ведет к потере ценности

Адаптация

nextconf

Зачем нам еще один инструмент?

Сопротивление

Отсутствие понимания ценности со стороны команды. DSL может напугать тех кто не привык к коду

Интеграция

Отсутствие времени на обсуждение, внедрение и поддержку процесса

ADR и Docs

Систематизированный текст отлично дополнит диаграммы

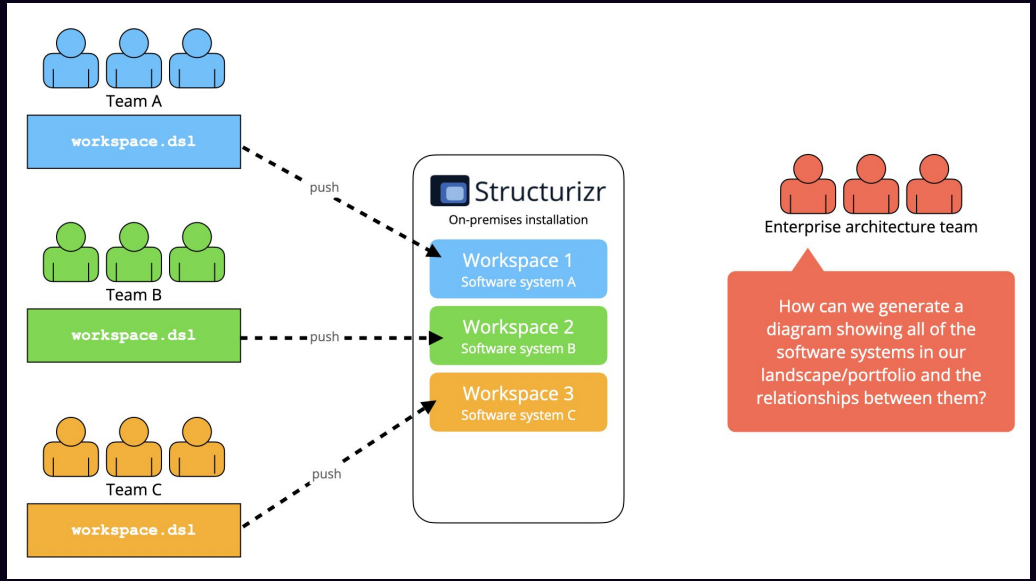
Цель

У каждого действия должна быть цель, поэтому необходимо ясно представлять для чего внедряется процесс

Адаптация

Целевое решение

- Structurizr on-premises
- Общий `workspace.dsl` с описанием System landscape
- Отдельный `workspace.dsl` с наследованием от общего в каждом проекте (сервисе)
- Встроенный iframe на странице в Confluence
- ADR и, возможно, документация сервисов



Адаптация

Пошаговая интеграция

1. Воркшоп для команды
2. Structurizr Lite + PlantUML экспорт в Confluence
3. ADRs в Confluence
4. Развертывание Structurizer on-premises
5. Общий `workspace.dsl`
6. Автогенерация `workspace.dsl` в сервисах на CI
7. Встраивание iframe на страницы в Confluence

Инструменты Structurizr

nextconf

К архитектуре через код

Системный подход к созданию архитектурных диаграмм значительно упрощает разработку и поддержку сервисов. В докладе мы кратко рассмотрели C4 model, познакомились с инструментами Structurizr и DSL. Поговорили о процессе внедрения подхода «Архитектура через код» и его адаптации в команде.

C4 model

Structurizr

Related repo

Спасибо
Вопросы?