Репозиторий архитектуры

Роман
Цирульников

money

Flow
2022

О компании ЮMoney

>20

лет промышленной
эксплуатации

>30

команд инженеров

>5 250

реализованных проектов
в рамках проектного офиса

Масштаб системы

3

основных направления бизнеса и множество интеграционных решений

>700

прикладных сервисов

>370

технических систем

Наши основные продукты

Электронный кошелёк
и банковские карты

Приём платежей
за товары и услуги

Расчётное обслуживание
и банковские карты
для юридических лиц

Проектный офис

- Задачи развития направлений бизнеса декомпозируются на набор проектов
- Проект представляет собой пакет задач в рамках ограниченных сроков и ресурсов
- Составляется бэклог проектов направления
- Команды формируются под бэклог проектов направления

Эффекты масштаба и времени

- Постоянное накопление сложности систем
- Неуправляемый рост сложности — один из основных рисков для новых проектов
- Знание об устройстве системы не помещается ни в одной отдельно взятой голове
- Дальнейший рост системы невозможен при наличии связей всех со всеми
- Стоимость внесения изменений — это определяющий фактор развития

Практические наблюдения

- Наибольшую сложность представляют знания специфики предметной области
- Значительная часть ошибок в ПО происходит из-за непонимания предметной области
- Сложность — это количество необходимого знания для эффективной работы

Задачи архитектуры

- **Фиксировать и распространять знания**
- Преодолевать разрыв между предметной областью и технической командой
- Предоставлять методики проектирования решений
- Выстраивать матрицу ответственности за сервисы
- Обеспечивать предсказуемость и повторяемость результатов в последующих проектах
- Обеспечивать автономность команд

Информационная система

Информационная система —
система, организующая хранение
и обработку информации
о предметной области

ГОСТ 33707-2016
Информационные технологии. Словарь.
п. 4.452 информационная система

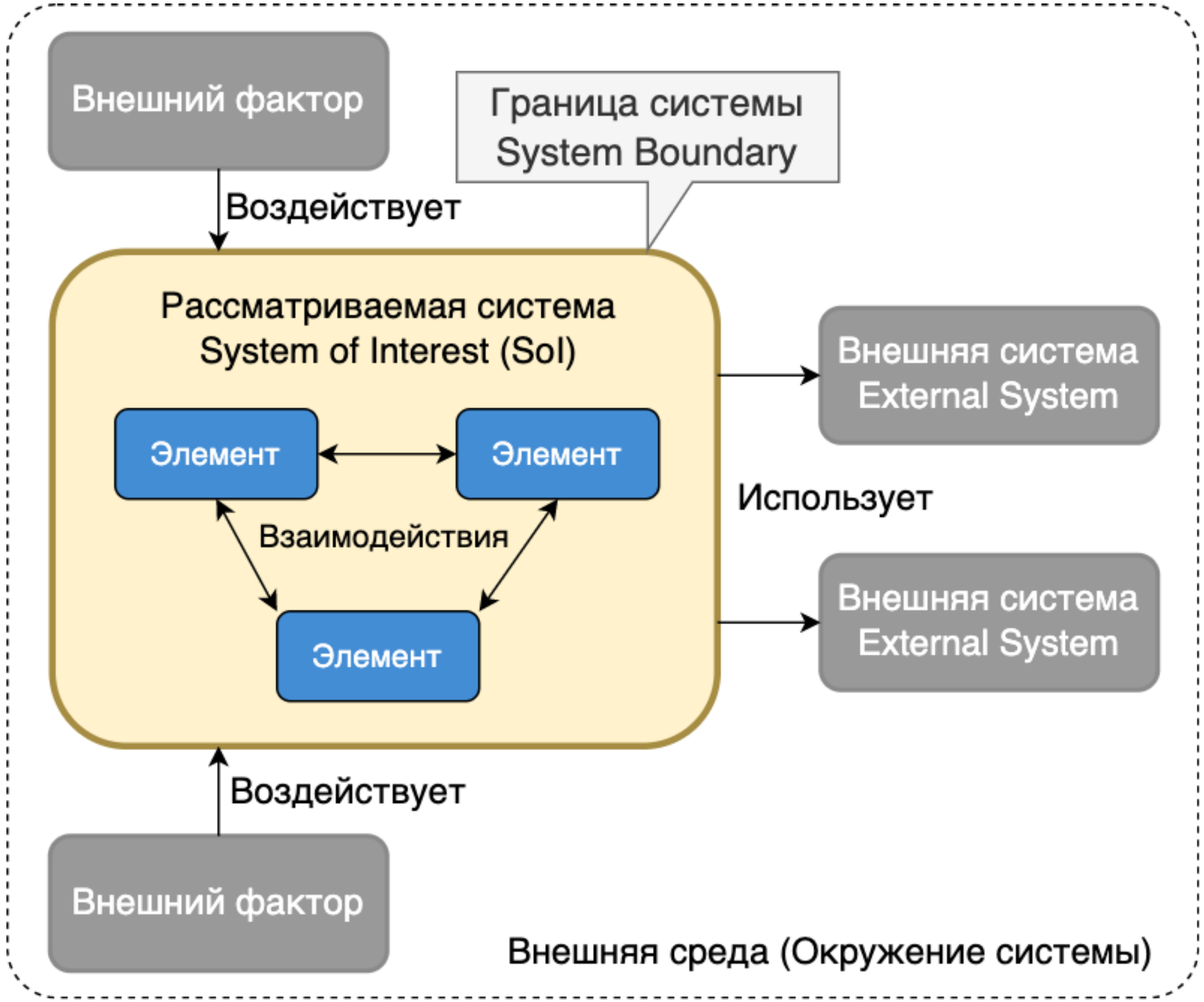
Information system —
information processing system, together with associated organizational resources such as human, technical, and financial resources, that provides and distributes information

ISO/IEC 2382:2015
Information technology — Vocabulary

Основы теории систем

Понятие рассматриваемой системы

- Называя что-то **системой** вы определяете её границы
- У системы есть внутренние взаимодействующие элементы
- Система взаимодействует с внешним окружением
- Поведение системы сложнее, чем сумма поведения её элементов



Принцип System Of Systems

Рассмотрение сложности окружающего мира посредством декомпозиции в иерархию:

- Рассматриваемая система (SoI)
- Набор подсистем как элементы рассматриваемой системы
- Надсистема, элементом которой система является

Enabling System

Обеспечивающие системы — это внешние, по отношению к рассматриваемой системы, предоставляющие ресурсы, поведение, данные рассматриваемой системе

Методика декомпозиции

Принципы построения архитектуры

- Ключевой фактор — выбор **стабильных во времени сущностей** в основу декомпозиции
- **Декомпозиция на основе выделения доменов предметных областей бизнеса**
- Элементарная единица декомпозиции — **сервис**
- **Сервис** — это единица прикладного поведения, заключает полный сценарий
- Система — иерархия сервисов, сгруппированных в домены
- Сервис может использовать другие сервисы как части своего сценария

Примеры сервисов

- Регистрация учетной записи
- Перевод с карты на карту
- Проведение упрощенной идентификации владельца кошелька
- Выставление счёта на оплату
- Выплата на банковский счёт
- Получение выписки по счёту клиента

Продукт

Продукт — это совокупность сервисов, совместно реализующих некоторую ценность для конечного пользователя

- Границы продукта отражают некоторую предметную область бизнеса
- Продукты могут использовать общую платформу

Примеры продуктов

- Приём платежей за товары и услуги
- Выплаты физическим лицам
- Электронный кошелёк
- Карты Cyberpunk 2077
- Расчётное обслуживание для юридических лиц

Платформа

- Платформа является частным случаем продукта для внутренних пользователей
- Платформы решают задачи обеспечения бизнес-процессов, используемых множеством продуктов
- Платформа предоставляет набор повторно используемых сервисов
- Платформа может стать самостоятельным продуктом

Примеры платформ

- Профиль учётной записи пользователя YooID
- Проведение платежа за товары и услуги
- Эквайринг банковских карт
- Эмиссия банковских карт

О платформах

- Важно избежать соблазна сделать общее универсальное решение — это невозможно
- Остерегайтесь реализации паттерна Golden Hammer
- Платформа предоставляет набор сервисов, решающих отдельные конкретные задачи
- Клиент имеет возможность использовать отдельно любые из сервисов на выбор

Выбор подхода

Требования

1. **Доступность широкому кругу сотрудников**
2. Онлайн, веб, всегда доступна актуальная версия
3. Гипертекстовая среда, ссылки между документами
4. Коллективная работа по множеству проектов
5. Назначены группы ответственных экспертов за сегменты системы

Ограничения

1. Доступность широкому кругу сотрудников, сложные нотации не годятся

2. Не будет специализированной группы архитекторов как звена производственной цепочки

3. Сложность сопровождения модели это решающий фактор, выделенного ресурса на сопровождение репозитория не будет

Ключевые решения

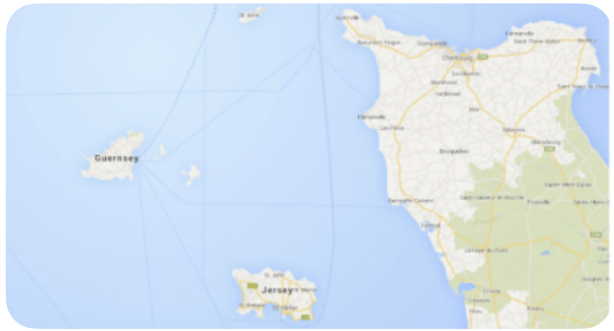
1. Решение для широкого круга сотрудников, а не специальной группы архитекторов
2. Простая нотация (C4 Model) и минимально достаточная метамодель
3. Распределенный репозиторий на разные группы сотрудников
4. Оставить пространство для специфики предметной области
5. Гипертекстовая среда навигации как связующее звено
6. Первичные документы — это спецификации сервисов системы

Нотация C4 Model

Реализует принцип System of Systems, последовательно детализирует систему

Context

zoom in



Container

zoom in

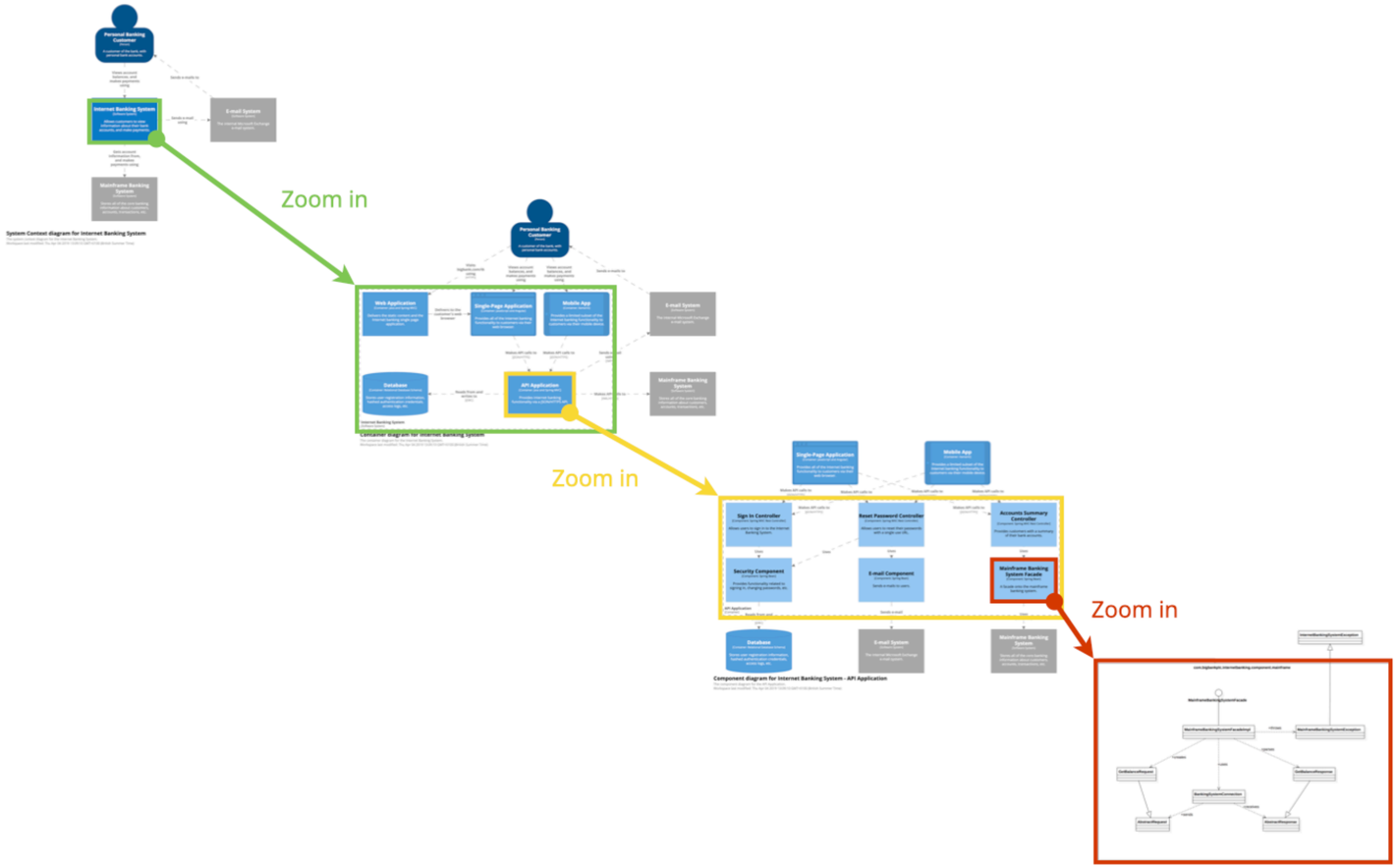
Component

zoom in

Code

Использованы материалы
https://c4model.com

The C4 model for visualising software architecture



Level 1
Context

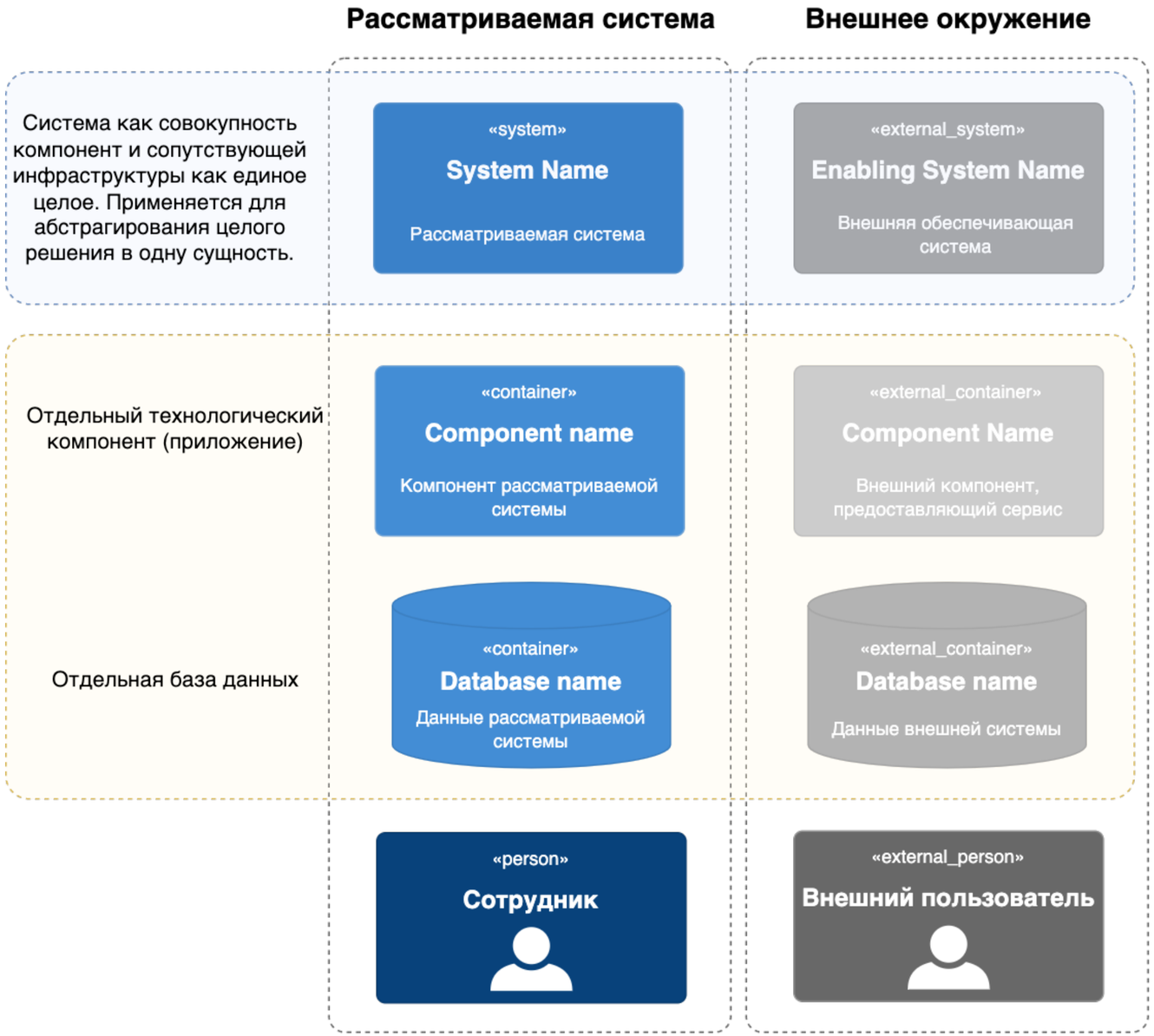
Level 2
Containers

Level 3
Components

Level 4
Code

Использованы материалы
https://c4model.com

Нотация C4 Model



Структура репозитория

Метамодель архитектуры

TOGAF Capability Map

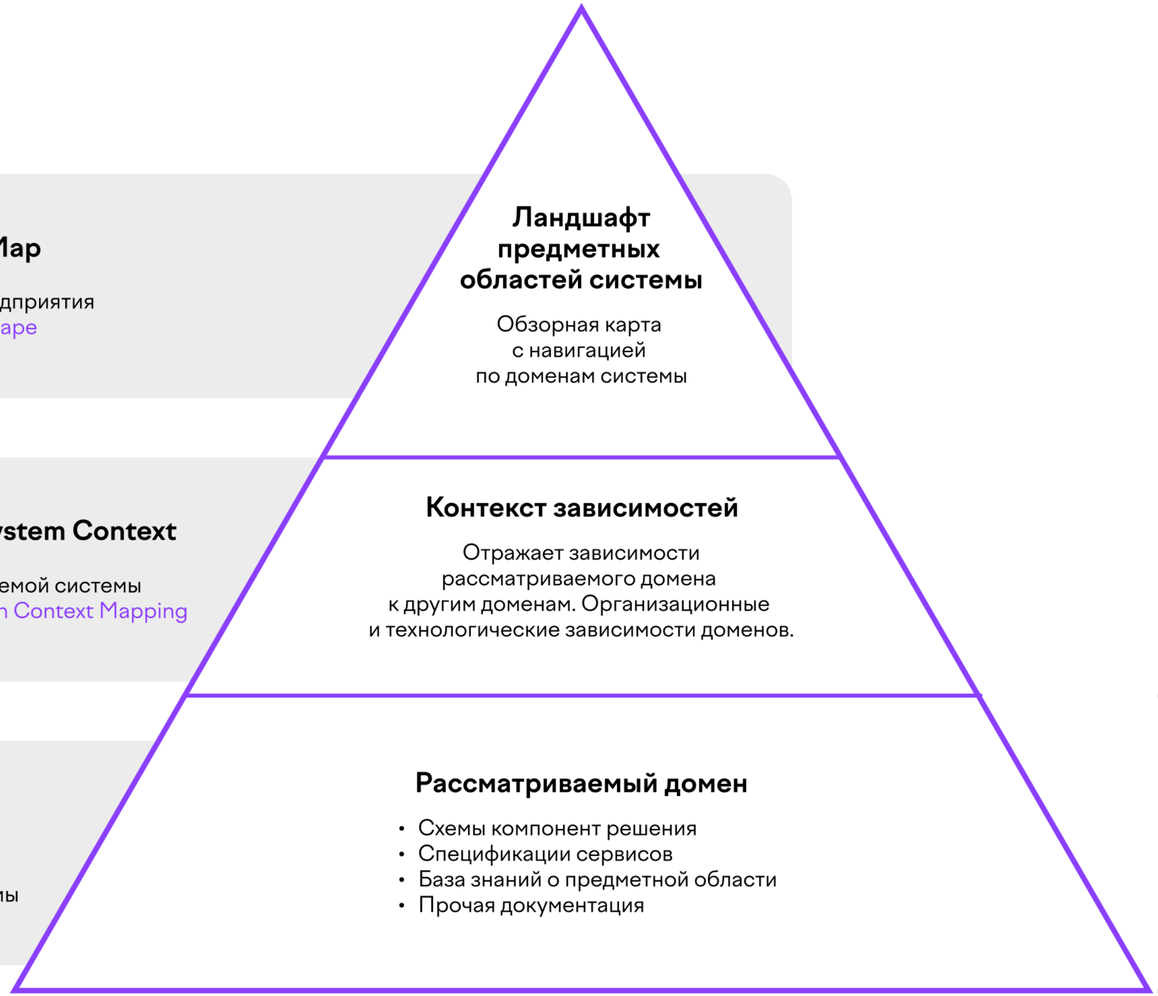
Состав групп систем предприятия
C4 Model: System Landscape

C4 Model Level 1: System Context

Окружение рассматриваемой системы
Domain Driven Design with Context Mapping

C4 Model Level 2

Компоненты
рассматриваемой системы



Репозиторий архитектуры предприятия

Гиперссылки

Репозиторий домена в PRODUCT-SPECS

Configuration Management Database (CMDB)

Сопровождается эксплуатацией по методике ITIL:

- Каталог сервисов (SERV)
- Каталог приложений и компонент их инфраструктуры (COM)
- Каталог внешних интерфейсов (IF)
- Каталог оборудования (CUI)

Слой #1. Ландшафт предметных областей

- На основе TOGAF Capability Map
- Capability — это группа компетенций в некоторой предметной области
- Каждый домен предметной области имеет группу ответственных экспертов

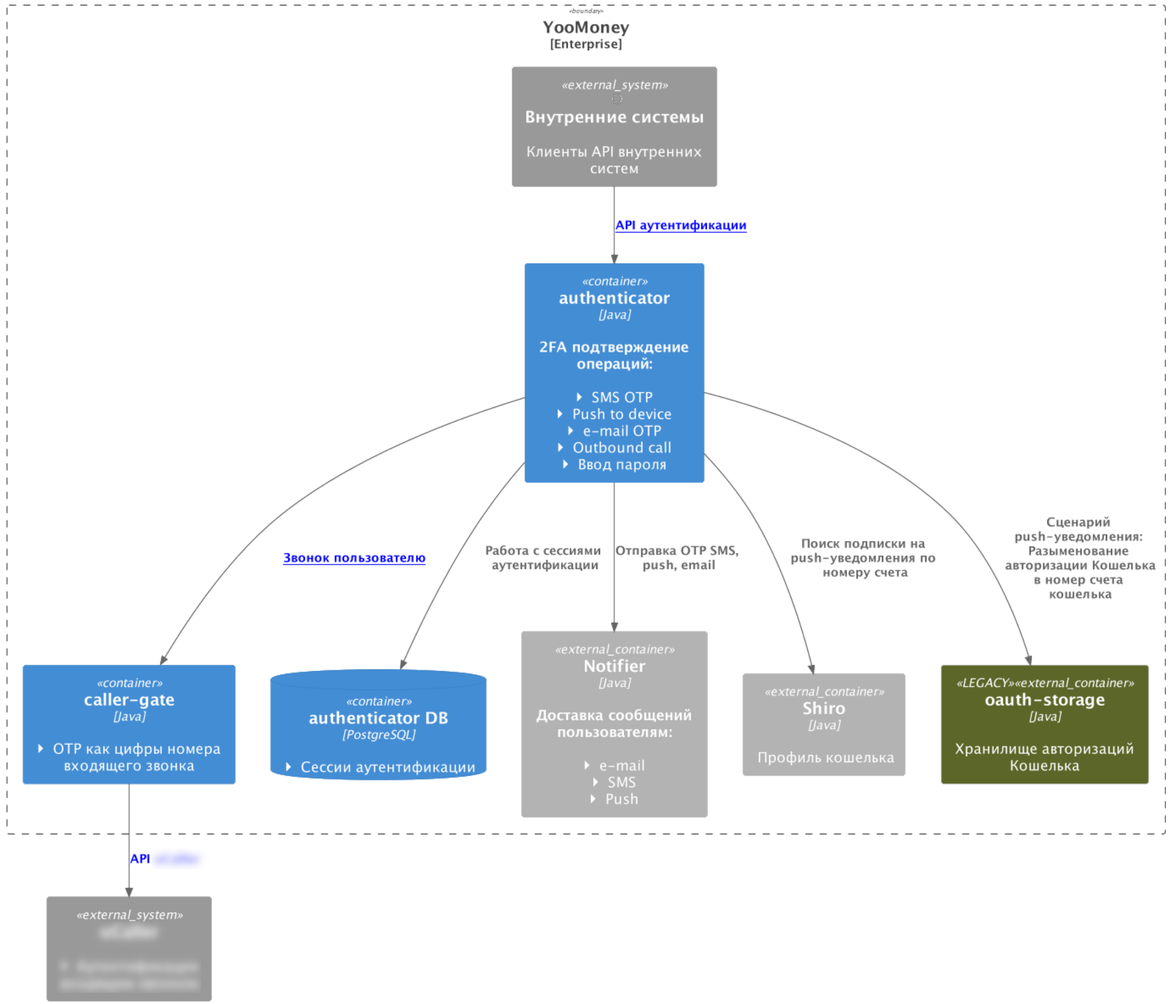
«PLATFORM»«system»
Эквайринг банковских карт

Эксперты
Документация
Зависимости
Схемы компонент

Слой #2. Контекст рассматриваемой системы

- Отражает зависимости этого домена и других доменов

Слой #3. Компоненты рассматриваемой системы



Гиперссылки

«PLATFORM»«system»
Authenticator: 2FA подтверждение операций

Эксперты
Документация
Зависимости
Схемы компонент

- SMS OTP, push, звонок пользователю, платежный пароль

Схема компонент

Open API

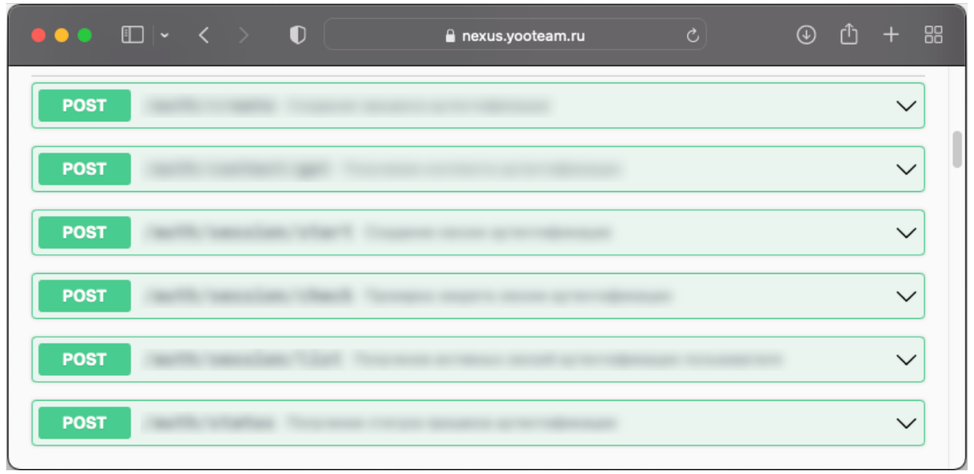
API аутентификации
[HTTP API]

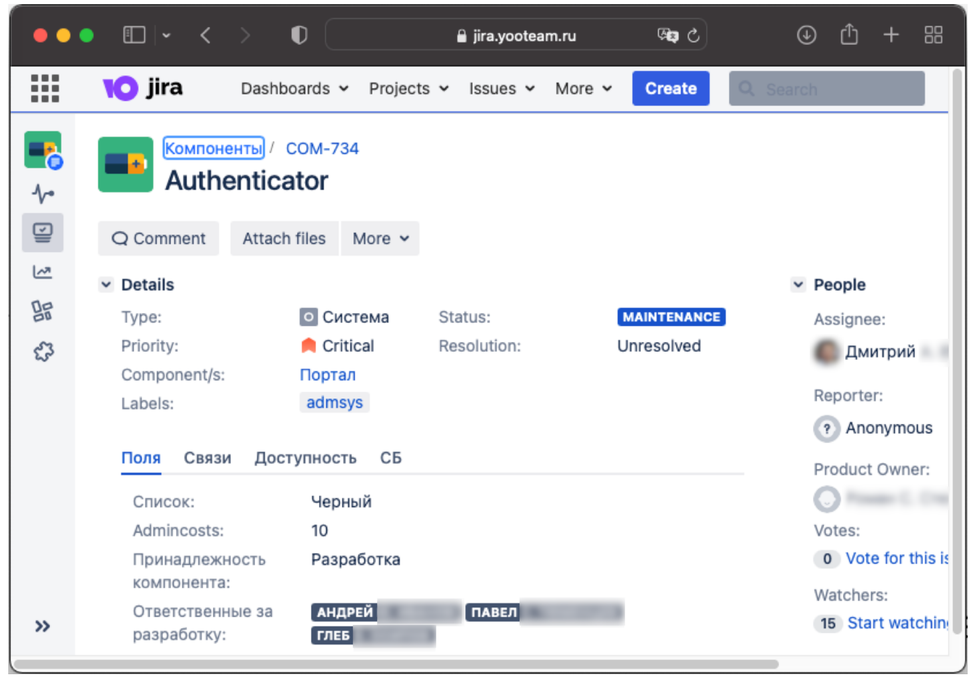
CMDB

«container»
authenticator
[Java]

2FA подтверждение операций:

- SMS OTP
- Push to device
- e-mail OTP
- Outbound call
- Ввод пароля





Инструменты

- Подход Docs as Code
- Подход Design Review
- Atlassian BitBucket Pull-Requests

Docs as Code
Основы подхода

Особенности работы с MS Word

- Работа в режиме правки и обмен документами по почте
- На каждой итерации множатся копии документа
- На каждой итерации приходится вычитывать весь текст
- Сложный документ в режиме редактирования становится нечитаемый
- Работа по нескольким параллельным проектам порождает копии документа и трудоёмкое ручное их сведение
- Трудоёмкость подготовки схем, выравнивание стрелочек
- Невозможно выделить повторно используемые блоки — приходится делать копипаст

✓ Готовый документ для передачи партнёрам

Особенности работы с Confluence

✓ Онлайн, всегда актуальная версия страницы

✓ Есть diff, просмотр истории правок

✓ Можно выделить повторно используемые блоки

- Работа по нескольким параллельным проектам порождает копии страницы и трудоёмкое ручное их сведение
- Отсутствуют механизмы контроля и приёмки правок

– Невозможность подготовки публичного документа для передачи партнёрам

- Низкое качество форматирования контента

Специфические проблемы старых проектов

- **Документ — это описание задачи (проекта), а не функционального блока системы (сервиса)**
- Отсутствует жизненный цикл документа после сдачи проекта
- Отсутствует актуальное представление описания системы
- Отсутствует структура навигации по документам, поиск нужной информации

Требования

1. Доступность широкому кругу сотрудников
2. Онлайн, веб, всегда доступна актуальная версия
3. Гипертекстовая среда, ссылки между документами
4. **Коллективная работа по множеству проектов**
5. **Назначены группы ответственных экспертов за сегменты системы**
6. Возможность ссылаться на старые документы
7. Возможность подготовки публичного документа для передачи партнёрам

Основы Docs as Code

- Система контроля версий (VCS) как у разработчиков
- Управление изменениями на основе механизмов VCS
- Человеко-читаемый текстовый формат исходного документа
- Инструмент преобразования исходного документа в конечное представление (html, pdf, etc)

Структура документов

- Документ отражает спецификацию сервиса
- Жизненный цикл документа соответствует жизненному циклу сервиса
- Группа документов отражает домен предметной области (продукт)
- Доменом предметной области владеет группа ответственных экспертов
- Дерево документов домена отражает естественную структуру его функциональных блоков

Спецификация сервиса

1. Информация о назначении сервиса, о решаемых задачах
2. Функциональные и нефункциональные требования, принятые решения
3. Описание прикладного (бизнес) процесса, схемы и пояснения
4. Детали технологической реализации, сценарии, схемы, API
5. Инструкции по сопровождению, настройки

Структура репозитория

- Один документ — это спецификация одного сервиса
- Один репозиторий отражает спецификацию продукта (домена)
- Master branch отражает текущее состояние боевой системы
- Feature branch + Pull-Request содержат работу над проектом или задачей
- Design review, приемка изменений через Pull-Request
- Review группой ответственных экспертов

Наш стек инструментов

- Формат Asciidoctor
- Atlassian BitBucket как хранилище документов, поиск, управление изменениями
- PlantUML для подготовки схем
- Результирующий документ формируется Asciidoctor.js и kroki.io
- Просмотр документов непосредственно из Atlassian BitBucket, при помощи плагина

Выбор формата

01 Markdown	02 Asciidoctor	03 reStructuredText	04 TeX/LaTeX
Функционально бедный	Широкая функциональность	Широкая функциональность	Широкая функциональность
Широкий выбор инструментов	Широкий выбор инструментов	Специфические инструменты (Python)	Специфические инструменты (Emacs, Vi, Unix)
Стандартизация отсутствует, несовместимости	Стандартизован	Стандартизован	Стандартизован
Очень простой синтаксис	Простой синтаксис	Простой синтаксис	Сложный синтаксис
Только HTML	HTML, PDF, ePUB и др.	HTML, PDF, ePUB и др.	HTML, PDF, ePUB и др.

Управление экспертизой

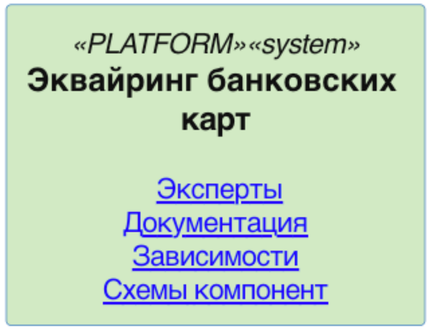
Решаемые задачи

- Накопление и распространение экспертизы в предметных областях
- Матрица поиска экспертов
- Design review решений

Design Review это

- Ревью документов группой ответственных экспертов
- Внутренний контроль качества технических решений
- Эффективный способ взаимного обучения коллег
- Возможность увидеть альтернативные пути решения задачи
- Формальная процедура согласования изменений
- Техническая реализация — BitBucket Pull-Request

Domain это



- Определение границ контекста
- **Группа компетенций в некоторой предметной области**
- Терминология
- Соответствующий им набор технических систем
- В DDD определено понятие Domain Context (Bounded Context)
- В C4 model — System Context

Роли

- **Domain Expert** (Ответственный эксперт) — этой ролью может обладать аналитик, разработчик, тестировщик, PM, PO. Является держателем знаний о домене, в том числе участвует в design review.

- **Analyst Lead** (Ответственный аналитик) — это один из Domain Experts, ответственный за репозиторий документации домена

- **Tech Lead** (Технологический лидер домена) — это один из Domain Experts, обладающий решающим правом голоса в случае спора в группе экспертов, а так же обязательный участник архитектурных комитетов по темам домена

- В группе экспертов важно сочетать компетенции аналитика (что нужно) и разработчика (как устроено)

Благодарю за внимание

Роман Цирульников,
архитектор

romanvt@yoomoney.ru
@romanvt

Благодарю за внимание

Роман Цирульников,
архитектор

romanvt@yoomoney.ru
@romanvt