

Курс: Otus Software Architect

Домашнее задание 1

по кате Нила Форда “Lights, Please”

Выполнил: Крошкин Игорь

24.12.2024

Содержание

1. [Бизнес-контекст](#)
2. [Бизнес-цели и драйверы](#)
3. [Стейкхолдеры и их потребности](#)
4. [Пользовательские истории](#)
5. [Атрибуты качества](#)
6. [Контекстная схема](#)
7. [Критические сценарии](#)
8. [Критические характеристики](#)
9. [Архитектурные решения](#)
10. [Система управления конфигурациями](#)
 1. [Бизнес-контекст и цели](#)
 2. [Критические сценарии](#)
 3. [Критические характеристики](#)
 4. [Базовое решение](#)
 5. [Архитектурное решение](#)
 6. [Альтернативные решения](#)

1. Бизнес-контекст

Оригинал: <https://nealford.com/katas/kata?id=LightsPlease>

Гигант в сфере бытовой электроники хочет создать систему для автоматизации дома: включение и выключение света, запирание и отпирание дверей, удаленное наблюдение с помощью камер и неопределенное поведение в будущем.

- Пользователи: каждая система будет продаваться потребителям (небольшим семьям), но компания рассчитывает продать тысячи таких устройств в течение первых трех лет.
- Требования:
 - система должна быть максимально готова к эксплуатации, но при этом продаваться в модульных блоках (камера, замок, термостат и т. д.) для удобства покупки
 - устройства должны быть доступны через Интернет (для удаленного мониторинга и доступа), и предполагается, что у пользователя будет существующая настройка WiFi (маршрутизатор и подключение) для подключения
 - клиенты могут программировать систему для управления различными модулями в соответствии со своими потребностями.
 - электротехникой для блоков займутся другие группы, а программные протоколы для управления модулями будут гибкими в соответствии с потребностями/проектами вашей архитектуры. (Они займутся реализацией модульной части протокола, как только вы им это укажете.)
- Дополнительный контекст:
 - готов инвестировать большую сумму, чтобы запустить это новое направление бизнеса
 - собирает данные от клиентов, которые согласились собирать более широкую статистику
 - международная компания

2. Бизнес-цели и драйверы

- Международный гигант в сфере бытовой электроники хочет запустить новое бизнес-направление и создать систему для автоматизации дома
- Разработка системы предполагается собственными силами, включая наличие штата сервисных инженеров для настройки и обслуживания оборудования, при этом сами устройства будут производиться и поставляться компаниями-партнерами
- Монетизация системы предполагается в виде тарифных планов, доступных по подписке

- Необходима интеграция с текущей системой продажи электроники для управления складскими запасами
- Необходима локализация пользовательского интерфейса системы в регионах присутствия

3. Стейкхолдеры и их потребности

- CUST: пользователь системы
 - Автоматизация процессов жизнеобеспечения, включая системы физического доступа и видеонаблюдения, подачи воды, газа, воздуха, отопления, электричества
 - Группирование устройств по жилым помещениям и типам оборудования
 - Поддержка пользовательских профилей и режимов энергосбережения, а также зависимости от времени суток, температуры окружающей среды и тд
 - Просмотр ключевых показателей через интерактивные аналитические панели с возможностью формирования отчетов за произвольный период
 - Использование популярных сервисов голосового управления для сценариев автоматизации
- MNGR: менеджер продаж
 - Формирование спецификаций оборудования для ключевых клиентов
- SPRT: инженер поддержки
 - Рекомендации по выбору используемого оборудования
 - Создание заявки для сервис-инженера
- SRVC: сервис-инженер
 - Проведение работ по настройке и сервисному обслуживанию оборудования на стороне пользователя системы

4. Пользовательские истории

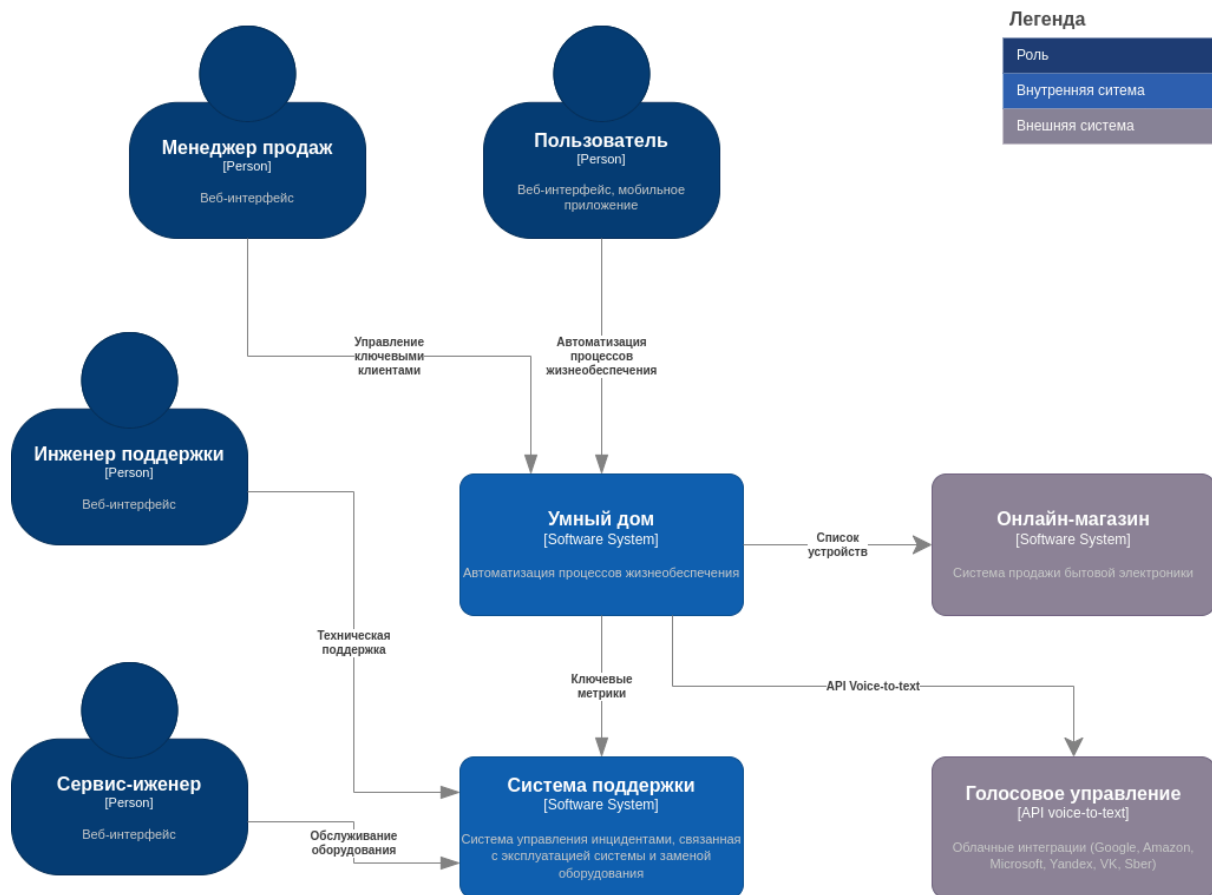
1. CUST-1: пользователь регистрируется в системе для подключения устройств, приобретенных через онлайн-магазин компании либо отдельно
2. CUST-2: пользователь имеет возможность создания профилей для автоматизации процессов жизнеобеспечения, например поддержание определенной температуры, влажности, вкл/откл режимов энергосбережения
3. CUST-3: пользователь использует датчики движения для автоматизации управления светом и камер видеонаблюдения
4. CUST-4: пользователь использует популярные сервисы голосового управления для автоматизации основных процессов

5. MNGR-1: менеджер продаж использует информацию о подключенных к системе устройствах для дополнительной продажи оборудования через онлайн-магазин компании
6. MNGR-2: менеджер продаж формирует спецификации оборудования для комплексных сценариев автоматизации
7. SPRT-1: инженер службы поддержки имеет возможность просмотреть историю взаимодействия для выбранного пользователя
8. SPRT-2: инженер службы поддержки использует внутреннюю базу знаний для решения технических вопросов, связанных с эксплуатацией системы
9. SRVC-1: сервис-инженер отвечает за инциденты, связанные с подключением и обслуживанием оборудования на стороне пользователя системы

5. Атрибуты качества

- Безопасность: использование усиленных методов аутентификации (биометрия, одноразовые пароли, сертификаты), протоколов TLS для конфиденциальности и целостности передаваемых данных
- Локализуемость: пользовательский интерфейс системы должен быть локализован в зависимости от региона
- Гетерогенность: поддержка широкого списка устройств от различных поставщиков и производителей
- Модифицируемость: система должна иметь возможность применения пользовательских конфигураций для подключенных устройств
- Настраиваемость: система должна поддерживать настраиваемые пользователем режимы просмотра ключевых метрик
- Обнаруживаемость: при подключении к сети пользователя, устройства должны автоматически отображаться в интерфейсе системы для последующей настройки
- Удобство использования: система должна иметь современный адаптивный пользовательский веб-интерфейс, а также нативные приложения для мобильных платформ iOS и Android
- Масштабируемость: поддержка одновременной работы до 1 миллиона подключенных устройств

6. Контекстная схема



7. Критические сценарии

- Регистрация пользователя в системе через веб-интерфейс либо мобильное приложение iOS/Android
- Подключение к системе устройств от разных поставщиков и производителей
- Обращение в службу поддержки в случае возникновения инцидентов во время эксплуатации системы
- Вызов сервисного инженера для первоначальной настройки и замены оборудования

8. Критические характеристики

- Производительность: время реакции системы на границе public-DMZ не должно превышать 3 сек
- Доступность: система должна иметь доступность 99,99% или допустимое время простоя не должно превышать 53 мин/год

- Отказоустойчивость: система должна использовать резервирование, когда отказ любого из компонентов не влияет на доступность системы в целом
- Совместимость: поддержка MQTT на уровне подключаемых устройств
- Безопасность: персональные данные и данные собираемые с устройств должны быть надежно защищены
- Удовлетворенность работы пользователей: время реакции на инциденты технической поддержки не должно превышать 4 часа

9. Архитектурные решения

Заголовок	Использование REST API для взаимодействия между бэкендом и фронтендом системы
Статус	Утвержден
Контекст	Для реализации взаимодействия между веб-сервисами фронтенда и бэкенда необходим открытый API, который позволит сторонним разработчикам интегрироваться с системой
Принятое решение	REST API выбран в силу своей простоты, универсальности и знакомства с данной технологией большинством разработчиков системы Встроенные средства документирования REST API обеспечивают минимальное время входа для новых разработчиков
Последствия	Использование REST API позволит осуществить необходимые интеграции (онлайн-магазин, системы голосового управления)
Возможные альтернативы	JSON RPC, gRPC, SOAP, GraphQL

Заголовок	Использование InfluxDB для хранения данных устройств и телеметрии
Статус	Предложен
Контекст	Для хранения собранных с устройств данных требуется многоплатформенная СУБД с

	открытым исходным кодом и поддержкой шардирования и репликации
Принятое решение	InfluxDB выбран в силу своей масштабируемости, поддержкой большого объема данных и встроенных возможностей аналитики
Последствия	InfluxDB может работать на нескольких серверах, позволяя быстро и без перебоев распределять нагрузку между репликами
Возможные альтернативы	Amazon DynamoDB, Apache Cassandra, ClickHouse, Redis

10. Система управления конфигурациями

10.1. Бизнес-контекст и цели

- Наличие подсистемы позволит продавать премиумные устройства через онлайн-магазин компании
- Наличие подсистемы позволит обеспечить полный жизненный цикл управления устройствами, включая автообнаружение, настройку, диагностику на ранних этапах эксплуатации и плановую замену оборудования
- Наличие подсистемы обеспечит консистентность пользовательских настроек вне зависимости от производителей и версий прошивок оборудования
- Наличие комплексных сценарии автоматизации позволит увеличить продажи через онлайн-магазин компании

10.2. Критические сценарии

- Применение параметров конфигурации устройств, обновление прошивок и перезагрузки без необходимости физического доступа к оборудованию
- Группирование устройств на основе атрибутов, местоположения для централизованного применения конфигураций
- Использование подсистемы мониторинга для увеличения срока жизни устройств за счет проактивного выявления ошибок
- Регистрация применения конфигураций в подсистеме логирования

10.3. Критические характеристики

- Расширяемость: наращивание функционала за счет добавления новых микросервисов

- Обнаруживаемость: автообнаружение устройств и регистрация применения конфигураций
- Идемпотентность: результат применения конфигурации не зависит от количества итераций
- Сходимость: подключенные устройства должны применять изменения в течение минуты
- Автономность: гарантированный результат применения конфигурации вне зависимости от статуса устройства за счет асинхронного характера взаимодействия
- Обслуживаемость: возможность вносить изменения без отключения системы
- Гранулярность: возможность группировать устройства на основе атрибутов, местонахождения, функциональных признаков для централизованного применения конфигураций
- Версионность: возможность отката конфигураций к предыдущим версиям

10.4. Базовое решение

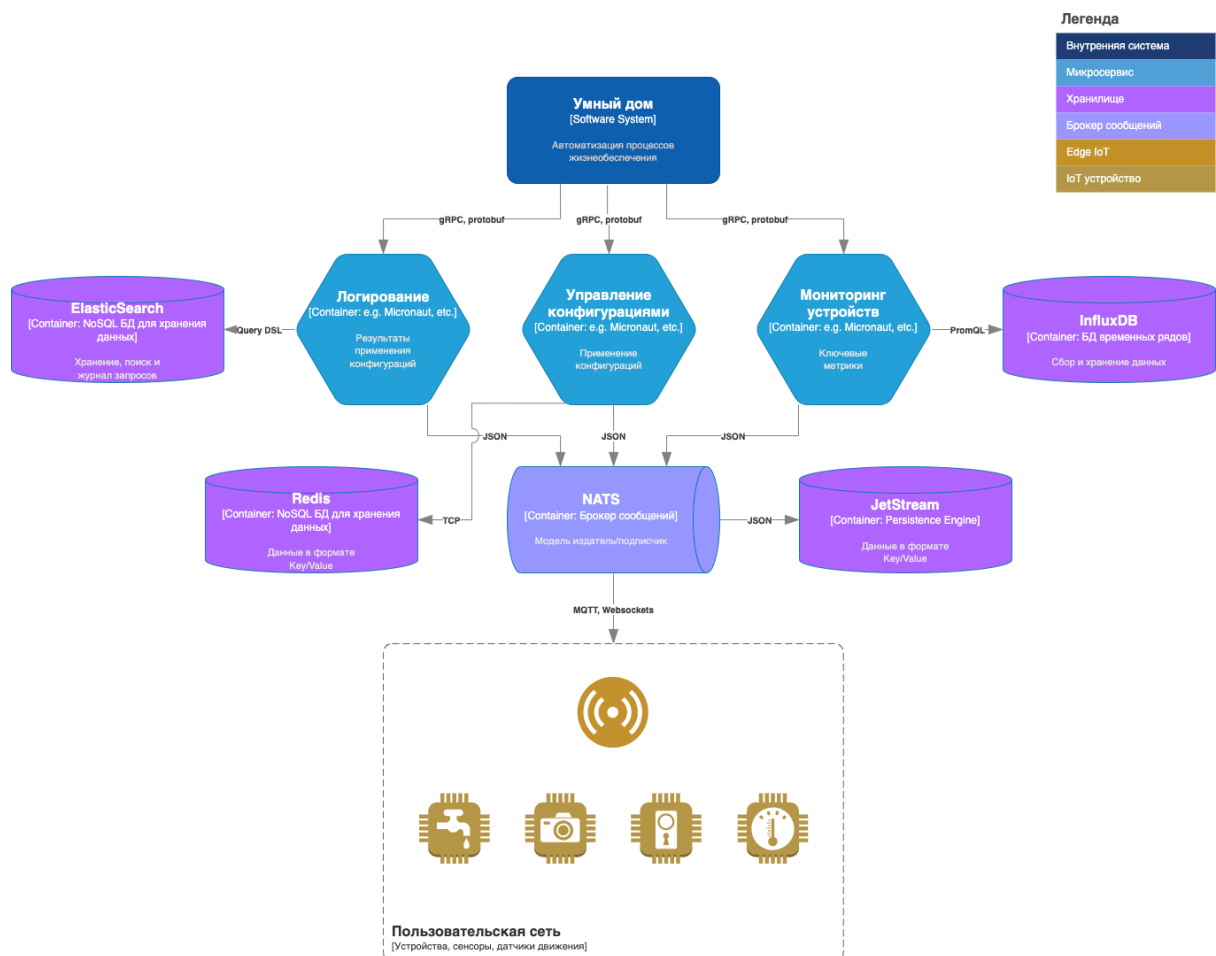


Рис. Контейнер-схема для подсистемы управления конфигурациями

Подсистема управления конфигурациями реализуется с использованием микросервисной архитектуры и включает следующие компоненты:

- Подсистема управления конфигурациями: использует Redis для хранения перманентных данных конфигурации и NATS в качестве брокера для передачи MQTT сообщений подключаемым устройствам
- Подсистема мониторинга: использует БД временных рядов InfluxDB для хранения данных и выявления потенциальных проблем до их появления
- Подсистема логирования: фиксирует применение конфигураций и используется для выявления устройств, которые не соответствуют пользовательским параметрам, в том числе при автообнаружении новых устройств
- Edge IoT: используется для подключения к Интернет, сетевой настройки и сбора первичных данных
- gRPC, protobuf: используется для обмена данными между основной системой и подсистемой управления конфигурациями для сериализации передаваемых данных

10.5. Архитектурное решение

Заголовок	Использование NATS в качестве брокера сообщений для подсистемы управления конфигурациями
Статус	Утвержден
Контекст	Для асинхронного взаимодействия между подсистемой управления конфигурациями и подключенными устройствами необходим брокер сообщений с поддержкой протокола MQTT
Принятое решение	NATS является высокопроизводительным распределенным брокером сообщений с поддержкой механизмов Publish/Subscribe, Request/Reply, Streaming, Key/Value, MQTT
Последствия	NATS включает встроенную поддержку различных клиентских устройств, гарантии доставки сообщений и возможности для интеграции

10.6. Альтернативные решения

Apache Kafka	<p>Плюсы:</p> <ul style="list-style-type: none">• Производительность и отказоустойчивость• Высокая пропускная способность до 1 миллиона сообщений в секунду• Поддержка механизма Publish/Subscribe <p>Минусы:</p>
---------------------	---

	<ul style="list-style-type: none"> • Отсутствие встроенной поддержки протокола MQTT • Требователен к ресурсам • Отсутствие механизма Request/Reply
RabbitMQ	<p>Плюсы:</p> <ul style="list-style-type: none"> • Простота использования • Кластеризация и федерация • Поддержка очереди сообщений <p>Минусы:</p> <ul style="list-style-type: none"> • Низкая пропускная способность • Отсутствуют встроенные возможности по распределению данных между репликами • Push-схема взаимодействия, что затрудняет возможность подключения к устройствам
Системы управления конфигурациями (Ansible, Puppet, Chef)	<p>Плюсы:</p> <ul style="list-style-type: none"> • Простота развертывания и использования • Возможность отката к предыдущим версиям конфигураций • Большое количество шаблонов <p>Минусы:</p> <ul style="list-style-type: none"> • Не предназначены для работы с большим количеством клиентов • Отсутствует поддержка асинхронного взаимодействия • Слабые возможности по интеграции