

Что под капотом местного радио

Максим Крентовский

DevImpress

У нас есть план!

Постановка задачи

- a. почему FM-радио еще живо
- b. как сделать свою региональную радиостанцию
- c. проблемы и решение
- d. что внутри FM-вещания

Эволюция решения:

- a. прототип
- b. первая версия, ее архитектура и недостатки
- c. вторая версия - что поменялось и почему это лучше
- d. эксплуатация и доработки

Почему FM-радио еще живо

1. есть в каждом автомобиле
2. LTE/4G/xG на скоростях около 100 км/ч работает плохо
3. канал оповещения о ЧС, ГО

Почему FM-радио еще живо

1. есть в каждом автомобиле
2. LTE/4G/xG на скоростях около 100 км/ч работает плохо
3. канал оповещения о ЧС, ГО

Есть федеральные радиостанции (ГПМ, РМГ, ...), но

1. слушатели хотят слушать местные новости, местную погоду и местное время (11 часовых поясов)
2. местная реклама

Как сделать свое региональное радио

1. получить лицензию на вещание

Как сделать свое региональное радио

1. получить лицензию на вещание
2. выкупить частоту на аукционе

Как сделать свое региональное радио

1. получить лицензию на вещание
2. выкупить частоту на аукционе
3. заключить договор с федеральной станцией

Как сделать свое региональное радио

1. получить лицензию на вещание
2. выкупить частоту на аукционе
3. заключить договор с федеральной станцией
4. поставить оборудование (компьютер вещания, спутниковый канал, FM-процессор, передатчик, RDS-генератор)

Как сделать свое региональное радио

1. получить лицензию на вещание
2. выкупить частоту на аукционе
3. заключить договор с федеральной станцией
4. поставить оборудование (компьютер вещания, спутниковый канал, FM-процессор, передатчик, RDS-генератор)
5. закупить и настроить ПО

Как сделать свое региональное радио

1. получить лицензию на вещание
2. выкупить частоту на аукционе
3. заключить договор с федеральной станцией
4. поставить оборудование (компьютер вещания, спутниковый канал, FM-процессор, передатчик, RDS-генератор)
5. закупить и настроить ПО
6. нанять персонал (системный администратор, трафик-менеджер, менеджеры по продажам)

Как сделать свое региональное радио

1. получить лицензию на вещание
2. выкупить частоту на аукционе
3. заключить договор с федеральной станцией
4. поставить оборудование (компьютер вещания, спутниковый канал, FM-процессор, передатчик, RDS-генератор)
5. закупить и настроить ПО
6. нанять персонал (системный администратор, трафик-менеджер, менеджеры по продажам)
7. подготовить “обертку” для эфира (джинглы, влеты, добивки и т.п.)

Как сделать свое региональное радио

1. получить лицензию на вещание
2. выкупить частоту на аукционе
3. заключить договор с федеральной станцией
4. поставить оборудование (компьютер вещания, спутниковый канал, FM-процессор, передатчик, RDS-генератор)
5. закупить и настроить ПО
6. нанять персонал (системный администратор, трафик-менеджер, менеджеры по продажам)
7. подготовить “обертку” для эфира (джинглы, влеты, добивки и т.п.)
8. запустить вещание и начать продажи рекламы

Проблемы

Когда станция одна - еще ничего, но когда их 10, 20, 40?

- отсутствие звука в эфире - катастрофа

Проблемы

Когда станция одна - еще ничего, но когда их 10, 20, 40?

- отсутствие звука в эфире - катастрофа
- оборудование имеет свойство отказывать

Проблемы

Когда станция одна - еще ничего, но когда их 10, 20, 40?

- отсутствие звука в эфире - катастрофа
- оборудование имеет свойство отказывать
- ПО для вещания, ОС, антивирусы - добавляют проблем

Проблемы

Когда станция одна - еще ничего, но когда их 10, 20, 40?

- отсутствие звука в эфире - катастрофа
- оборудование имеет свойство отказывать
- ПО для вещания, ОС, антивирусы - добавляют проблем
- человеческий фактор

Проблемы

Когда станция одна - еще ничего, но когда их 10, 20, 40?

- отсутствие звука в эфире - катастрофа
- оборудование имеет свойство отказывать
- ПО для вещания, ОС, антивирусы - добавляют проблем
- человеческий фактор
- (самое главное) это все в разных регионах и городах!

Предлагаемое решение

1. интернет есть везде

Предлагаемое решение

1. интернет есть везде
2. делаем всё на сервере в ДЦ (включая звукообработку)

Предлагаемое решение

1. интернет есть везде
2. делаем всё на сервере в ДЦ (включая звукообработку)
3. управляем через web-интерфейс из любого региона

Предлагаемое решение

1. интернет есть везде
2. делаем всё на сервере в ДЦ (включая звукообработку)
3. управляем через web-интерфейс из любого региона
4. на передатчик устанавливаем устройство на микроконтроллере, простое и надежное

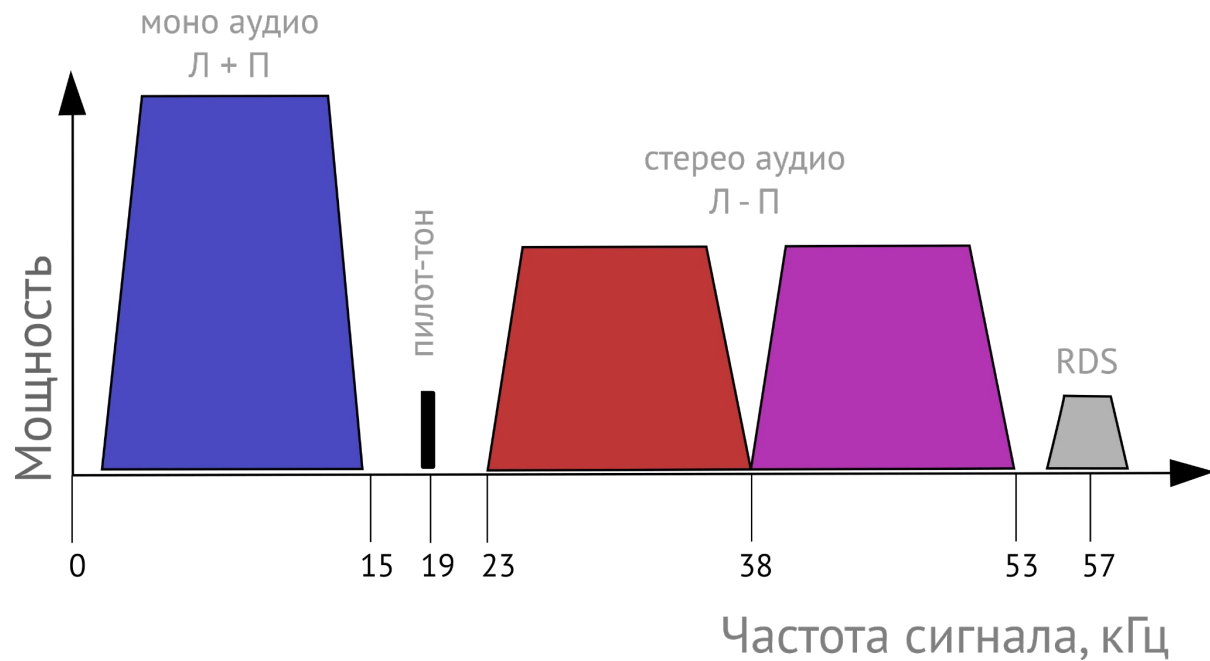
Предлагаемое решение

1. интернет есть везде
2. делаем всё на сервере в ДЦ (включая звукообработку)
3. управляем через web-интерфейс из любого региона
4. на передатчик устанавливаем устройство на микроконтроллере, простое и надежное

Плюсы:

- + меньше оборудования - меньше отказов и затрат на персонал
- + проще и доступнее контроль и управление

Вещание - комплексный стерео-сигнал (КСС)



УКВ (VHF)
ЧМ (FM)
88-108 МГц

Немного о RDS

- вещается на поднесущей 57 кГц

Немного о RDS

- вещается на поднесущей 57 кГц
- бинарные данные в манчестерском (самосинхронизируемом) коде

Немного о RDS

- вещается на поднесущей 57 кГц
- бинарные данные в манчестерском (самосинхронизируемом) коде
- ~1858 бит в секунду

Немного о RDS

- вещается на поднесущей 57 кГц
- бинарные данные в манчестерском (самосинхронизируемом) коде
- ~1858 бит в секунду
- множество полей и флагов:
 - PS - название радиостанций (8 символов)
 - RT - вспомогательный текст, например, название композиции (64 символа)
 - тип и жанр радио
 - альтернативные частоты (то же радио в соседних регионах для бесшовной миграции)
 - флаги признака информации о пробках
 - и многое другое (расширяемое)

Разметка эфира

- нет возможности выровнять эфир точно по времени

Разметка эфира

- нет возможности выровнять эфир точно по времени
- DTMF-коды как маркер начала блока (влет) и конца блока (вылет)

Разметка эфира

- нет возможности выровнять эфир точно по времени
- DTMF-коды как маркер начала блока (влет) и конца блока (вылет)
- добивки (дополнение блока до регламентного времени в эфире)
 - случайные
 - по расписанию (лежит на FTP федеральной радиостанции вместе со всеми вариантами композиций на разное время)

Разметка эфира

- нет возможности выровнять эфир точно по времени
- DTMF-коды как маркер начала блока (влет) и конца блока (вылет)
- добивки (дополнение блока до регламентного времени в эфире)
 - случайные
 - по расписанию (лежит на FTP федеральной радиостанции вместе со всеми вариантами композиций на разное время)
- плавное переключение между федеральным и региональным эфиром

Разметка эфира

- нет возможности выровнять эфир точно по времени
- DTMF-коды как маркер начала блока (влет) и конца блока (вылет)
- добивки (дополнение блока до регламентного времени в эфире)
 - случайные
 - по расписанию (лежит на FTP федеральной радиостанции вместе со всеми вариантами композиций на разное время)
- плавное переключение между федеральным и региональным эфиром
- есть нюансы формирования регионального плейлиста

Прототип

- прием и выдача - HTTP-вещание (IceCast), традиция

Прототип

- прием и выдача - HTTP-вещание (IceCast), традиция
- управление и контроль (Erlang)

Прототип

- прием и выдача - HTTP-вещание (IceCast), традиция
- управление и контроль (Erlang)
- плавное переключение между двумя потоками (gstreamer)

Прототип

- прием и выдача - HTTP-вещание (IceCast), традиция
- управление и контроль (Erlang)
- плавное переключение между двумя потоками (gstreamer)
- обнаружение DTMF-меток и активация врезок (gstreamer)

Прототип

- прием и выдача - HTTP-вещание (IceCast), традиция
- управление и контроль (Erlang)
- плавное переключение между двумя потоками (gstreamer)
- обнаружение DTMF-меток и активация врезок (gstreamer)
- одна федеральная радиостанция - много региональных
 - забирать поток для каждой (дорого, неэффективно)
 - UDP Multicast (забираем один раз - отдаем во много потребителей)

Прототип

- прием и выдача - HTTP-вещание (IceCast), традиция
- управление и контроль (Erlang)
- плавное переключение между двумя потоками (gstreamer)
- обнаружение DTMF-меток и активация врезок (gstreamer)
- одна федеральная радиостанция - много региональных
 - забирать поток для каждой (дорого, неэффективно)
 - UDP Multicast (забираем один раз - отдаем во много потребителей)
- резервирование входных каналов
 - равноценное
 - приоритетное
 - “основной-резервные”

Прототип - почему gstreamer

1. простое создание конвейеров для обработки данных без написания (почти) собственного ПО

Прототип - почему gstreamer

1. простое создание конвейеров для обработки данных без написания (почти) собственного ПО
2. гибкость и простота изменений цепочек обработки

Прототип - почему gstreamer

1. простое создание конвейеров для обработки данных без написания (почти) собственного ПО
2. гибкость и простота изменений цепочек обработки
3. огромное число элементов обработки и возможность встраивания сторонних фильтров

Прототип - почему gstreamer

1. простое создание конвейеров для обработки данных без написания (почти) собственного ПО
2. гибкость и простота изменений цепочек обработки
3. огромное число элементов обработки и возможность встраивания сторонних фильтров
4. если что-то не хватает - всегда можно дописать свой элемент цепочки

Прототип - почему Erlang

1. платформа создавалась как универсальный клей (control plane) над системами обработки потоковых данных (data plane)

Прототип - почему Erlang

1. платформа создавалась как универсальный клей (control plane) над системами обработки потоковых данных (data plane)
2. система “мягкого” реального времени

Прототип - почему Erlang

1. платформа создавалась как универсальный клей (control plane) над системами обработки потоковых данных (data plane)
2. система “мягкого” реального времени
3. горячее обновление (если надо поправить на лету)

Прототип - почему Erlang

1. платформа создавалась как универсальный клей (control plane) над системами обработки потоковых данных (data plane)
2. система “мягкого” реального времени
3. горячее обновление (если надо поправить на лету)
4. отличные возможности быстро и легко реализовать любой бинарный протокол (кодирование RDS, взаимодействие с устройствами)

Первая версия - команда

Состав:

- 2 разработчика на Erlang
- 1 фронтенд-разработчик VueJS

Вне штата со стороны заказчика:

- инженер-системотехник (устройство вещания)
- специалист по звукообработке (автор Linux Studio Plugins Project)

Первая версия - компоненты

- приложение на gstreamer для переключения потоков

Первая версия - компоненты

- приложение на gstreamer для переключения потоков
- gstreamer-цепочки + LSP для получения, обработки, кодирования и передачи звука

Первая версия - компоненты

- приложение на gstreamer для переключения потоков
- gstreamer-цепочки + LSP для получения, обработки, кодирования и передачи звука
- Rocket Stream Audio Server для вещания по HTTP

Первая версия - компоненты

- приложение на gstreamer для переключения потоков
- gstreamer-цепочки + LSP для получения, обработки, кодирования и передачи звука
- Rocket Stream Audio Server для вещания по HTTP
- PostgreSQL для хранения данных

Первая версия - компоненты

- приложение на gstreamer для переключения потоков
- gstreamer-цепочки + LSP для получения, обработки, кодирования и передачи звука
- Rocket Stream Audio Server для вещания по HTTP
- PostgreSQL для хранения данных
- Redis для сессий/кеширования

Первая версия - компоненты

- приложение на gstreamer для переключения потоков
- gstreamer-цепочки + LSP для получения, обработки, кодирования и передачи звука
- Rocket Stream Audio Server для вещания по HTTP
- PostgreSQL для хранения данных
- Redis для сессий/кеширования
- RWA для интерфейса (VueJS)

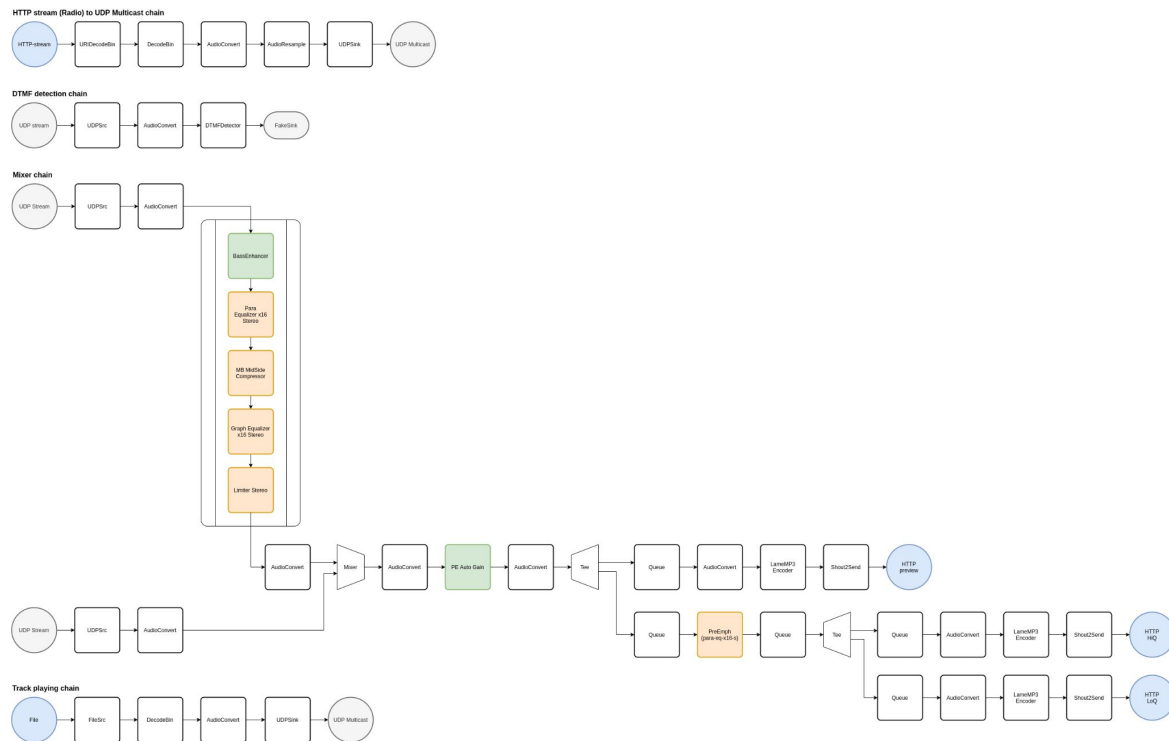
Первая версия - компоненты

- приложение на gstreamer для переключения потоков
- gstreamer-цепочки + LSP для получения, обработки, кодирования и передачи звука
- Rocket Stream Audio Server для вещания по HTTP
- PostgreSQL для хранения данных
- Redis для сессий/кеширования
- RWA для интерфейса (VueJS)
- FFmpeg для обработки входящих медиа-файлов и архивации

Первая версия - компоненты

- приложение на gstreamer для переключения потоков
- gstreamer-цепочки + LSP для получения, обработки, кодирования и передачи звука
- Rocket Stream Audio Server для вещания по HTTP
- PostgreSQL для хранения данных
- Redis для сессий/кеширования
- RWA для интерфейса (VueJS)
- FFmpeg для обработки входящих медиа-файлов и архивации
- Erlang как клей (управляющий слой для всего этого), генератор RDS и подсистема сопряжения с устройствами вещания

Первая версия - цепочки gstreamer-a



Проблемы первой версии

1. излишний оптимизм (планировали 150-200 потоков на сервер, получили 75 максимум)

Проблемы первой версии

1. излишний оптимизм (планировали 150-200 потоков на сервер, получили 75 максимум)
2. коварство радийщиков (1-2 станции на тест, 30-60 радиостанций потом) - горизонтальное масштабирование наступило гораздо раньше

Проблемы первой версии

1. излишний оптимизм (планировали 150-200 потоков на сервер, получили 75 максимум)
2. коварство радийщиков (1-2 станции на тест, 30-60 радиостанций потом) - горизонтальное масштабирование наступило гораздо раньше
3. нерегулярность расписания (как следствие - сложность добивок)

Проблемы первой версии

1. излишний оптимизм (планировали 150-200 потоков на сервер, получили 75 максимум)
2. коварство радильщиков (1-2 станции на тест, 30-60 радиостанций потом) - горизонтальное масштабирование наступило гораздо раньше
3. нерегулярность расписания (как следствие - сложность добивок)
4. новые пожелания заказчика (журналирование всех действий, проигрывание готовых блоков), не предусмотренные в текущей версии

Нужно делать вторую версию.

Вторая версия - команда

Состав:

- 6 бекенд-разработчиков + тим-лид на Python (но одновременно работали 2-3)
- 2 фронтенд-разработчика
- архитектор/разработчик на Erlang/системный администратор

Со стороны заказчика:

- состав не изменился

Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)

Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)
- Фронтенд (VueJS + NuxtJS)

Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)
- Фронтенд (VueJS + NuxtJS)
- PostgreSQL/Redis - ничего не поменялось

Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)
- Фронтенд (VueJS + NuxtJS)
- PostgreSQL/Redis - ничего не поменялось
- MQTT для обмена сообщениями между компонентами

Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)
- Фронтенд (VueJS + NuxtJS)
- PostgreSQL/Redis - ничего не поменялось
- MQTT для обмена сообщениями между компонентами
- GStreamer как в виде цепочек обработки, так и приложения для переключения потоков

Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)
- Фронтенд (VueJS + NuxtJS)
- PostgreSQL/Redis - ничего не поменялось
- MQTT для обмена сообщениями между компонентами
- GStreamer как в виде цепочек обработки, так и приложения для переключения потоков
- FFmpeg для архивации и обработки входящих медиа-файлов

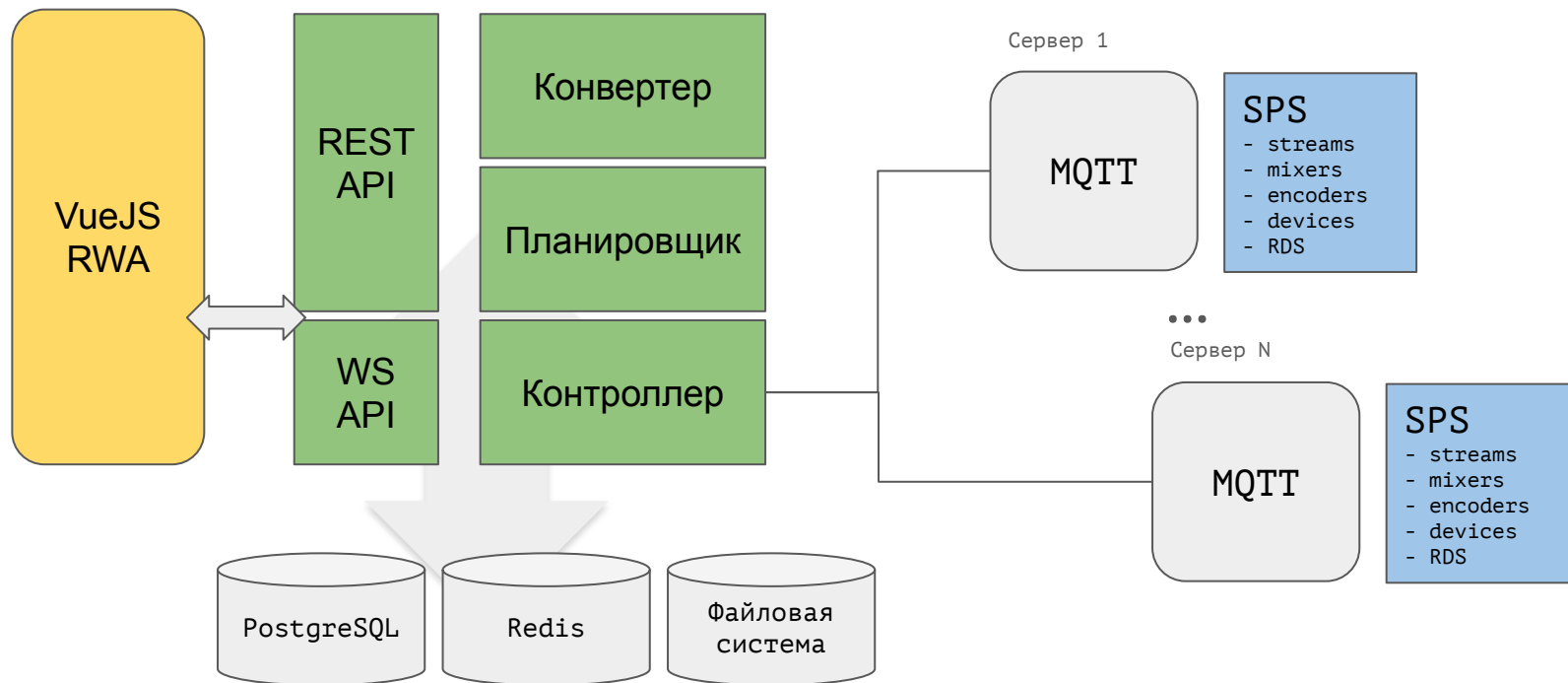
Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)
- Фронтенд (VueJS + NuxtJS)
- PostgreSQL/Redis - ничего не поменялось
- MQTT для обмена сообщениями между компонентами
- GStreamer как в виде цепочек обработки, так и приложения для переключения потоков
- FFmpeg для архивации и обработки входящих медиа-файлов
- Erlang для управления цепочками GStreamer-a

Вторая версия - компоненты

- Бекенд (Python + FastAPI + asyncio)
- Фронтенд (VueJS + NuxtJS)
- PostgreSQL/Redis - ничего не поменялось
- MQTT для обмена сообщениями между компонентами
- GStreamer как в виде цепочек обработки, так и приложения для переключения потоков
- FFmpeg для архивации и обработки входящих медиа-файлов
- Erlang для управления цепочками GStreamer-а
- ClickHouse для журналирования событий (глупость, заменили через год на Postgres по причине неиспользования возможностей)

Вторая версия - архитектура



Вторая версия - улучшения

- гибкая гетерогенная архитектура

Вторая версия - улучшения

- гибкая гетерогенная архитектура
- управление процессом обработки с бекенда

Вторая версия - улучшения

- гибкая гетерогенная архитектура
- управление процессом обработки с бекенда
- горизонтальное масштабирование

Вторая версия - улучшения

- гибкая гетерогенная архитектура
- управление процессом обработки с бекенда
- горизонтальное масштабирование
- расширение функциональности:
 - генерация плейлистов на каждый блок врезки
 - пользователи и ограничения доступа
 - настраиваемая сетка вещания
 - мини CRM-система и создание документов

Эксплуатация и доработки

Состав:

- один бекенд/фронтенд/Erlang-разработчик, он же сисадмин, он же служба технической поддержки

Со стороны заказчика:

- все те же.

Что удалось добиться на этом этапе

- все таки сделали добивки! (и дополнительный модуль синхронизации с медиатекой и плейлистами с FTP)

Что удалось добиться на этом этапе

- все таки сделали добивки! (и дополнительный модуль синхронизации с медиатекой и плейлистами с FTP)
- сервер резервного вещания для устройств (отдается чистый поток без вставок, если ДЦ/основные сервера легли)

Что удалось добиться на этом этапе

- все таки сделали добивки! (и дополнительный модуль синхронизации с медиатекой и плейлистами с FTP)
- сервер резервного вещания для устройств (отдается чистый поток без вставок, если ДЦ/основные сервера легли)
- добавили возможность врезки прямых эфиров

Что удалось добиться на этом этапе

- все таки сделали добивки! (и дополнительный модуль синхронизации с медиатекой и плейлистами с FTP)
- сервер резервного вещания для устройств (отдается чистый поток без вставок, если ДЦ/основные сервера легли)
- добавили возможность врезки прямых эфиров
- сделали формирование потоков радиостанций по плейлистам

Что удалось добиться на этом этапе

- все таки сделали добивки! (и дополнительный модуль синхронизации с медиатекой и плейлистами с FTP)
- сервер резервного вещания для устройств (отдается чистый поток без вставок, если ДЦ/основные сервера легли)
- добавили возможность врезки прямых эфиров
- сделали формирование потоков радиостанций по плейлистам
- сделали телеграм-боты для мониторинга и уведомлений

Что удалось добиться на этом этапе

- все таки сделали добивки! (и дополнительный модуль синхронизации с медиатекой и плейлистами с FTP)
- сервер резервного вещания для устройств (отдается чистый поток без вставок, если ДЦ/основные сервера легли)
- добавили возможность врезки прямых эфиров
- сделали формирование потоков радиостанций по плейлистам
- сделали телеграм-боты для мониторинга и уведомлений
- добавили подсистему генерации роликов и прогнозов погоды на основе сервисов TTS, сократив человеческие затраты на подготовку и запись, при этом увеличив точность и оперативность прогнозов

Итого на данный момент

1. стоимость эксплуатации сокращается примерно на 20% относительно традиционного метода создания, а время развертывания радиостанции - до нескольких часов

Итого на данный момент

1. стоимость эксплуатации сокращается примерно на 20% относительно традиционного метода создания, а время развертывания радиостанции - до нескольких часов
2. сервис имеет коммерческий потенциал и продается

Итого на данный момент

1. стоимость эксплуатации сокращается примерно на 20% относительно традиционного метода создания, а время развертывания радиостанции - до нескольких часов
2. сервис имеет коммерческий потенциал и продается
3. заказчик доволен и избавлен от большого количества головной боли

Итого на данный момент

1. стоимость эксплуатации сокращается примерно на 20% относительно традиционного метода создания, а время развертывания радиостанции - до нескольких часов
2. сервис имеет коммерческий потенциал и продается
3. заказчик доволен и избавлен от большого количества головной боли
4. `gstreamer` - удобный инструмент для создания конвейеров обработки данных

Итого на данный момент

1. стоимость эксплуатации сокращается примерно на 20% относительно традиционного метода создания, а время развертывания радиостанции - до нескольких часов
2. сервис имеет коммерческий потенциал и продается
3. заказчик доволен и избавлен от большого количества головной боли
4. gstreamer - удобный инструмент для создания конвейеров обработки данных
5. Erlang/OTP - работает как и должно

Вопросы?

Немного персональных данных

<https://devimpress.ru/>

max@devimpress.ru

<https://t.me/mkrentovskiy>

