



# Реализация платформы для сбора необработанных данных в реальном времени и мониторинга состояния производственных линий

**Автор:** Мирзаянов Глеб Романович, 18.Б11-мм  
**Научный руководитель:** доцент кафедры системного программирования Д.В. Луцив  
**Консультант:** ведущий разработчик Grid Dynamics  
В.М. Черногор

Санкт-Петербургский государственный университет

# Внедрение IoT на производстве

## Плюсы внедрения:

- Решения основанные на данных
- Предотвращение ошибок ведущих к простою производства
- Повышение качества продукции
- Уменьшение гарантийных расходов

## Сложность:

- Сбор и хранения большого количества данных
- Обработка сырых непреобразованных данных

Сбор и хранение информации происходит на стороне фабрики:

- Отсутствие общей инфраструктуры компании на фабриках-поставщиках для унифицированного сбора данных
- Формат данных не удовлетворяет стандарту корпорации
- Отсутствует связь между деталями и продуктом
- Для анализа данных для каждой фабрики нужно поддерживать отдельное соединение

Аналоги:

- Amazon QuickSight
- SAP Business Objects

**Цель работы** — реализация приложения, позволяющего собирать данные в реальном времени, обрабатывать их и выдавать статистику о состоянии фабричных конвейерных лент. Для достижения цели были поставлены следующие задачи:

- Обзор аналогов и существующих решений
- Проектирование архитектуры приложения
- Реализация основной функциональности сервиса
- Покрытие приложения модульными, интеграционными и нагрузочными тестами
- Апробация созданного продукта

# Архитектура приложения



Рис.: Основные компоненты

# Пользовательский сценарий

## Навигация пользователя

Владелец данных



Пользователь данных



Общий доступ



Рис.: Последовательность действий пользователя

# Особенности реализация

- Реализовано около 60 запросов
- Бизнес логика обработки запросов находится в классах сервисов
- Логика работы с данными вынесена в классы репозитории
- Функциональность взаимодействия с API сторонних приложений реализовано в классах клиентах:
  - ▶ DruidClient
  - ▶ TableauClient
  - ▶ AWSClient
  - ▶ AirflowClient

# Тестирование и апробация

- Интеграционное тестирование
  - ▶ Покрыты наиболее популярные запросы
  - ▶ Инфраструктура внедрена в отдельный пайплайн для удаленного исполнения
- Сервис нагрузочного тестирования
  - ▶ Найдены и исправленные узкие места в удалении - увеличение производительности в 1.5 раза
  - ▶ Состояние гонки данных при больше чем 10 одновременных запросов
  - ▶ Результаты развертываются в виде статического контента на GitHub Pages
- Апробация среди 30 уникальных пользователей

# Результаты

В течение выполнения дипломной работы были получены следующие результаты:

- Проведен обзор аналогов и существующих решений
- Спроектирована архитектура приложения. Приложение разбито на несколько подсистем, что снижает сложность разработки
- Разработана основная функциональность приложения, что позволяет пользователю проводить агрегацию и анализ данных
- Сформированы наборы тестовых данных для модульного и интеграционного тестирования. Разработано приложение для нагрузочного тестирования
- Приложение развернуто и апробировано среди 30 уникальных пользователей