



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

Факультет экономики, менеджмента и  
бизнес-информатики

# ИНТЕГРАЦИЯ IT РЕШЕНИЙ НА БАЗЕ INFLUXDB, ВЗАИМОДЕЙСТВИЕ КОМПОНЕНТОВ ВЕБ-СИСТЕМЫ

Горшков Олег Владимирович  
Аналитик, стажер-исследователь  
НУЛ МЭИ НИУ ВШЭ Пермь

Пермь, 2021



# ВЕБ-СИСТЕМА ИЛИ ВЕБ-ПРИЛОЖЕНИЕ

---

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере в базе данных, обмен информацией происходит по сети.

**Главное – много компонентов!**



# ОСНОВНЫЕ КОМПОНЕНТЫ СИСТЕМЫ

- Клиентское приложение HTML-JS (React)
- Серверное приложение C#
- Серверное аналитическое приложение Python
- Средство моделирования OpenModelica
- Реляционная БД PostgreSQL
- База данных временных рядов InfluxDB
- Средство визуализации Grafana



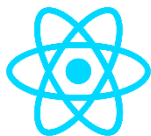
Grafana



*influxdb*



PostgreSQL



React

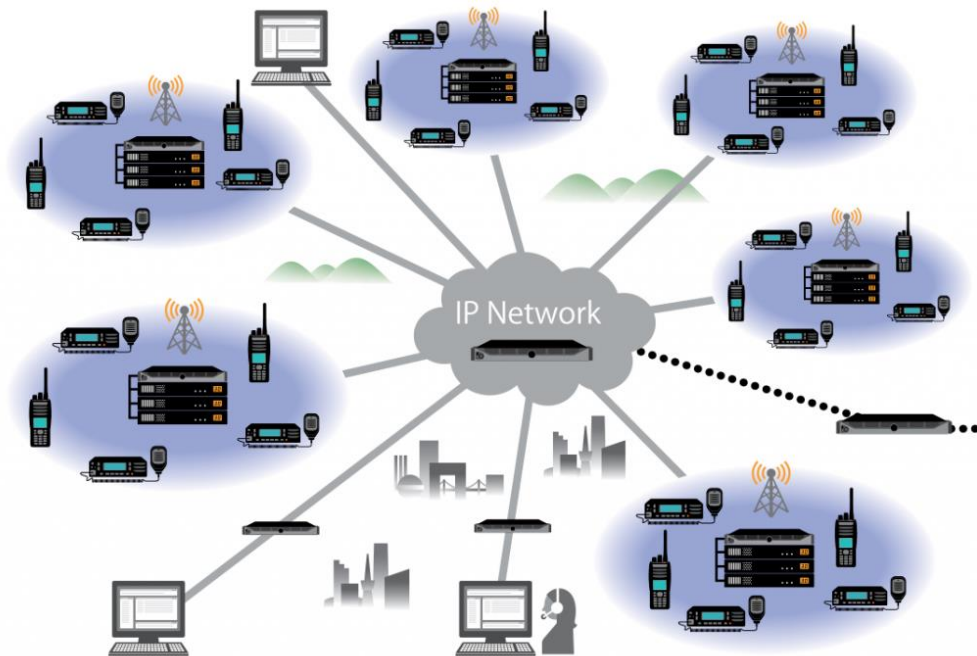


OpenModelica



# ВЗАИМОДЕЙСТВИЕ КОМПОНЕНТОВ

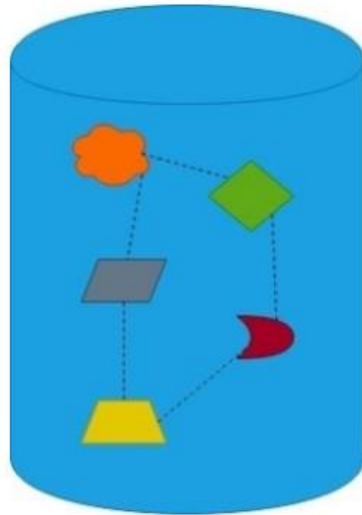
Возникает вопрос: «Как компоненты должны общаться между собой?»



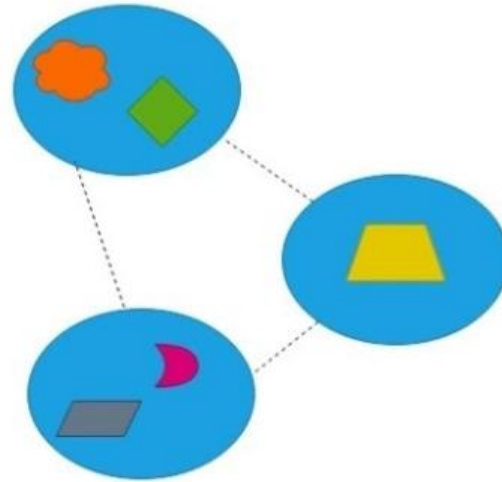


# АРХИТЕКТУРА СИСТЕМ

Два наиболее часто встречаемых типа архитектуры веб-систем



Монолит



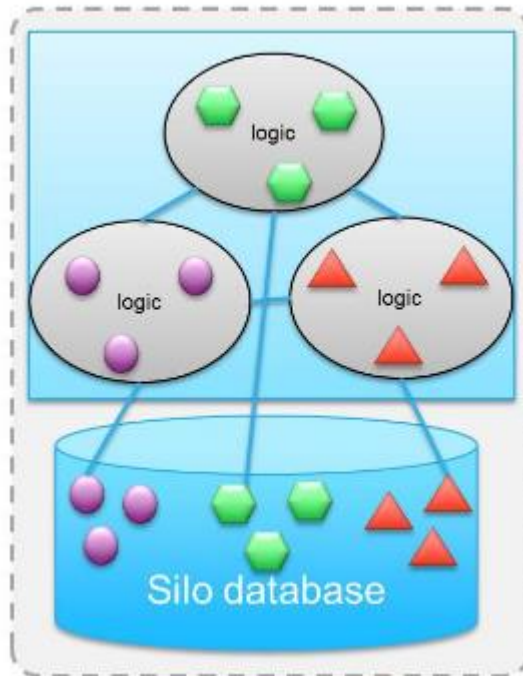
Микросервисы



# АРХИТЕКТУРА СИСТЕМЫ

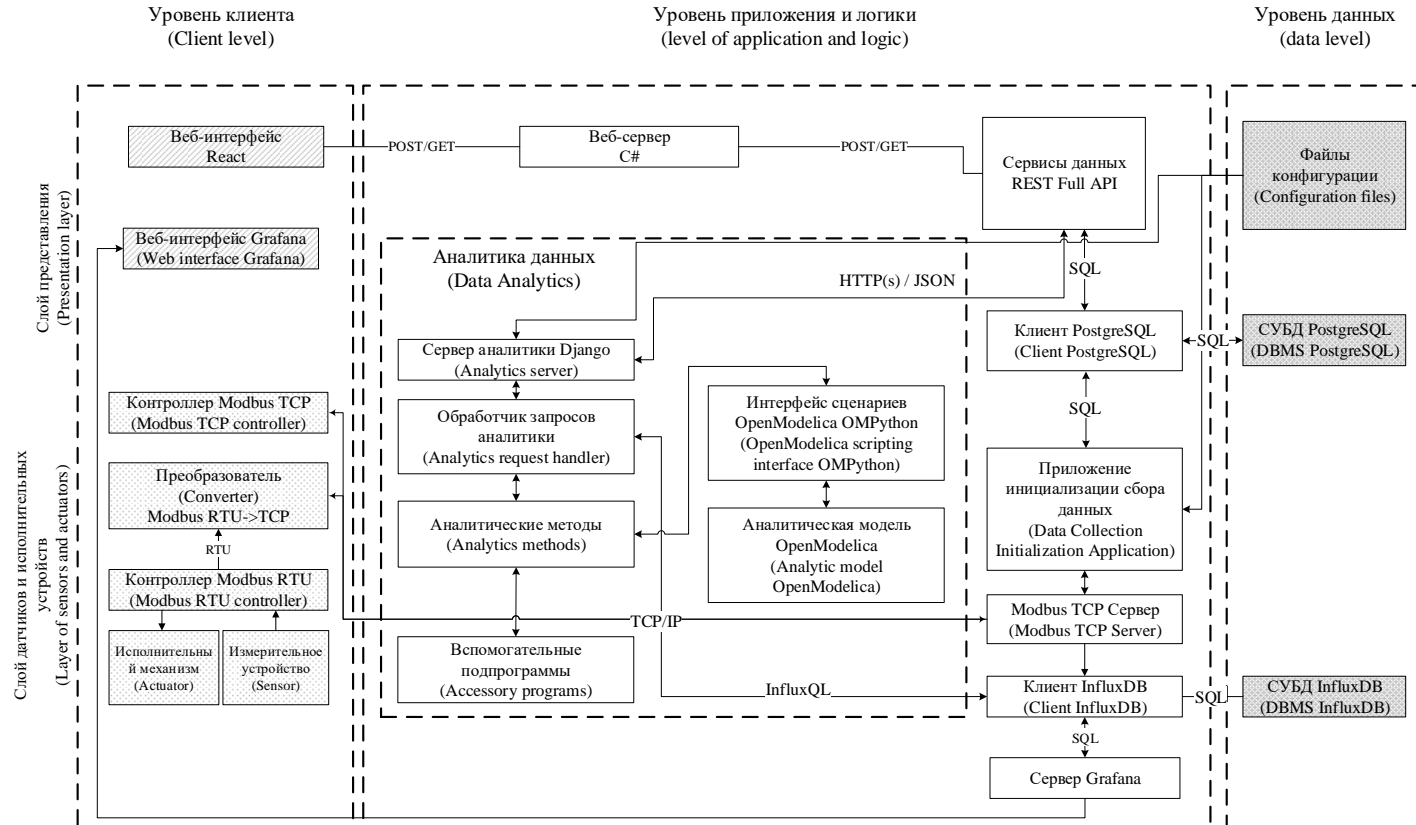
Выбираем микросервисную архитектуру с оговоркой:

Единая база данных временных рядов **InfluxDB**





# ВЗАИМОДЕЙСТВИЕ КОМПОНЕНТОВ





# REST API

REST (от англ. Representational State Transfer — «передача состояния представления») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой веб-системы.



Запросы:

POST

GET

PUT

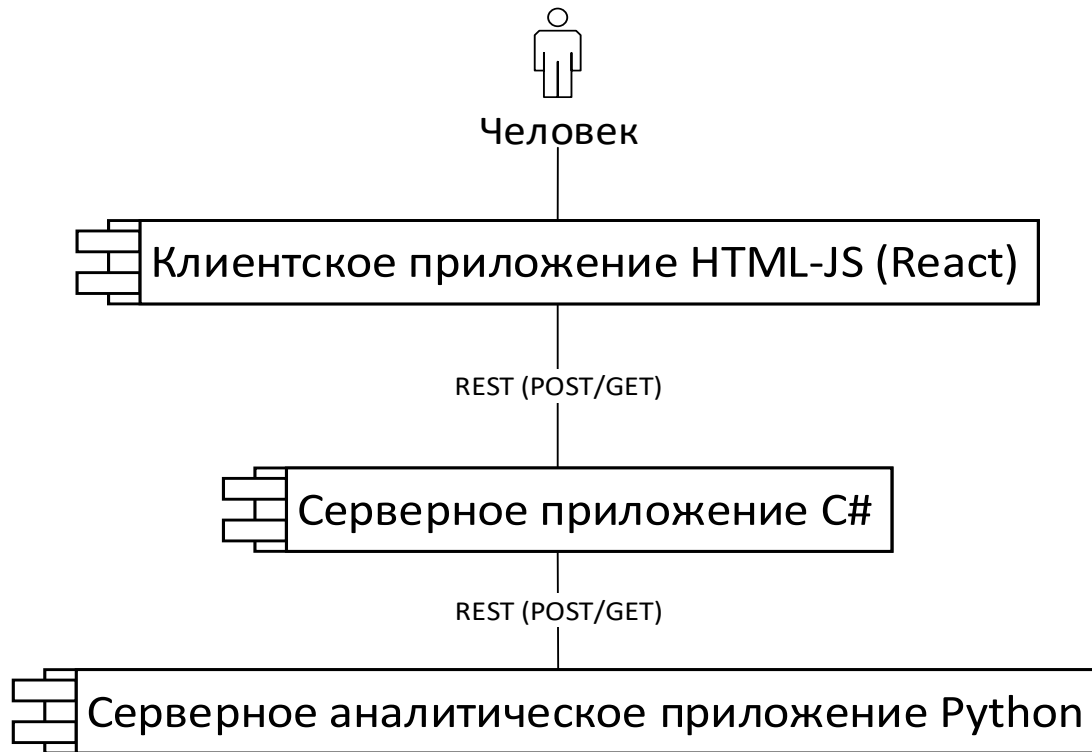
DELETE

```
"db_io_parameters": {  
  "mode": "r",  
  "result_id": [  
    "00000000-0000-0000-0000-000000000012"  
  ],  
  "device_id": [  
    "6859f2d3-f165-43df-be79-e2aa08eb4701"  
  ],  
  "data_source_id": [  
    "442"  
  ],  
  "time_upload": [  
    "2019-12-01_00:00:00+0000",  
    "2019-12-31_23:00:00+0000"  
  ]  
}
```





# ВЗАИМОДЕЙСТВИЕ КОМПОНЕНТОВ ЧЕРЕЗ REST API





# OPENMODELICA И PYTHON

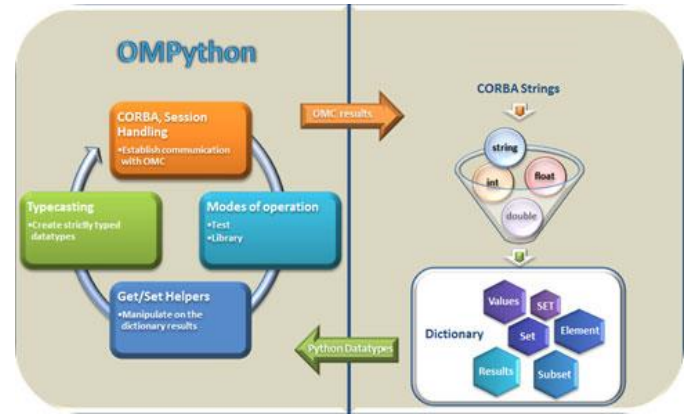
Серверное аналитическое приложение Python

Интерфейс OMPython

Аналитическая модель OpenModelica

Import

CORBA/ZeroMQ





# OPENMODELICA И PYTHON

```
# симуляция моделики
def simulate_modelica(model_path: str, model_name: str, params: list):
    # создаем моделику для симуляции
    mod = ModelicaSystem(model_path, model_name)
    mod.buildModel()
    # устанавливаем параметры симуляции из файла parameters.txt
    mod.setSimulationOptions(params)
    # симуляция моделики и возврат времени начала симуляции
    mod.simulate(resultfile=model_name + ".mat")
```

```
while True:
    # проводим симуляцию моделики и создаем файл данных моделики с расширением .mat
    simulate_modelica(model_path, model_name, params)

    # получаем данные и имена данных из файла
    my_file = dm.DyMatFile(model_name + ".mat")
    data = my_file.names()
    var_names = list()
    for var in data:
        var_names.append(var)

    # отбираем необходимые данные
    fields = [Buffer.var_simulation_time]
    for var in data:
        if re.search(Buffer.parameter_pattern(Buffer.mass_flow_par, Buffer.dens_par, Buffer.sect_par,
                                             Buffer.temp_par, Buffer.press_par), var):
            fields.append(var)

    # время симуляции
    res = dict()
    res[Buffer.var_simulation_time] = my_file.abscissa(var_names[0], valuesOnly=True)

    # добавление к времени симуляции начало симуляции
    for var in res:
        res[var] *= 1000
        for i in range(0, len(res[var])):
            res[var][i] = res[var][i] + timestamp

    # заполнение словаря данными
    max_count = -1
    for varName in var_names:
        if varName in fields:
            res[varName] = my_file.data(varName)
```



# POSTGRESQL & INFLUXDB

---

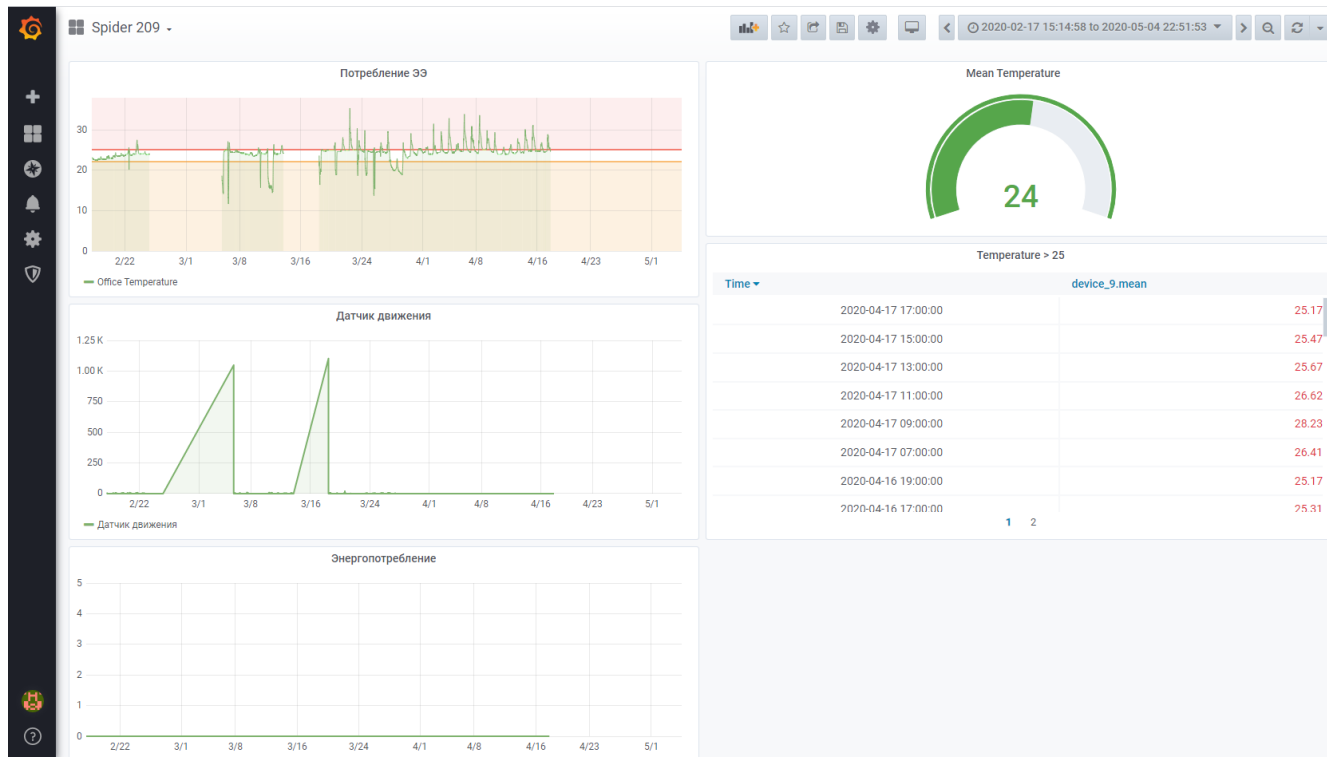
- В PostgreSQL параметры и информация, не имеющая временных меток
- В InfluxDB – измерения, данные, имеющие временную привязку



PostgreSQL

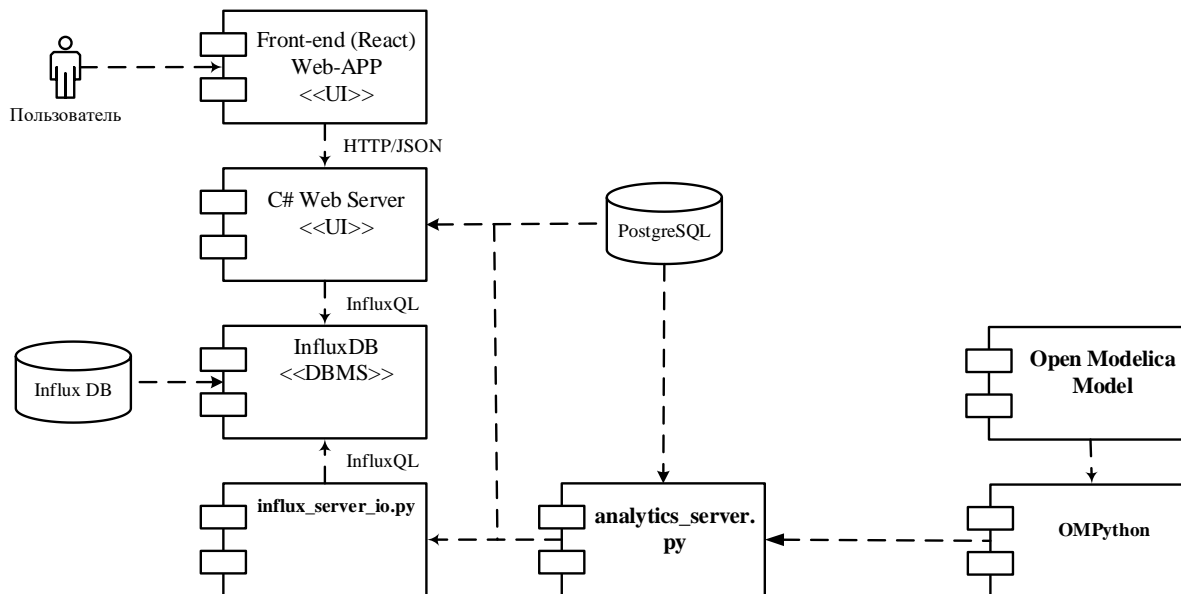


- SQL запросы
- Развертывание БД в отдельных контейнерах и доступ для C# и Python
- InfluxDBClient





# ДИАГРАММА КОМПОНЕНТОВ





# РЕАЛИЗАЦИЯ

Библиотеки Python:

- Datetime
- Numpy
- Pandas
- Sklearn
- Tensorflow
- Statsmodels





НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

Факультет экономики, менеджмента и  
бизнес-информатики

# ИНТЕГРАЦИЯ ИТ РЕШЕНИЙ НА БАЗЕ INFLUXDB, ВЗАИМОДЕЙСТВИЕ КОМПОНЕНТОВ ВЕБ-СИСТЕМЫ

Горшков Олег Владимирович  
+7 912 985 5859  
gorshkovoleg97@gmail.com  
<https://gorshkov.tech>

Пермь, 2021