



1. Компоненты и архитектура системы

Компоненты системы

Менеджер инфраструктуры

управление отказоустойчивостью и балансировкой нагрузки

Конструктор страниц

создание пользовательских web интерфейсов

Активаторы

обработка запросов от корпоративных систем по различным протоколам

Обработчики

запуск сценариев обработки информации

Источники данных

подключение внешних информационных систем в качестве источников данных

Сценарии

управление сценариями обработки информации

Типы источников данных

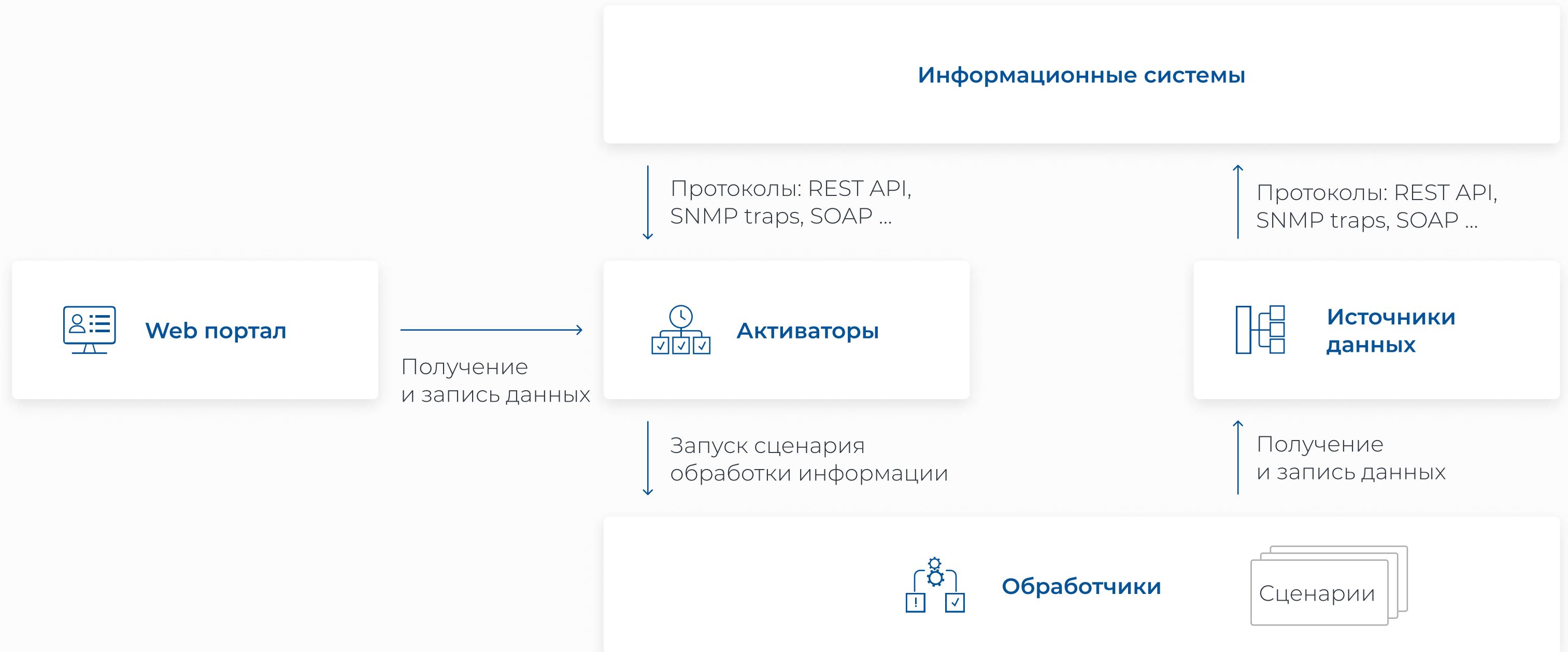
разработка шаблонов для подключения источников данных

Типы активаторов

разработка шаблонов используемых при настройке активаторов

1. Компоненты и архитектура системы

Архитектура системы



2. Поддерживаемые типы источников данных

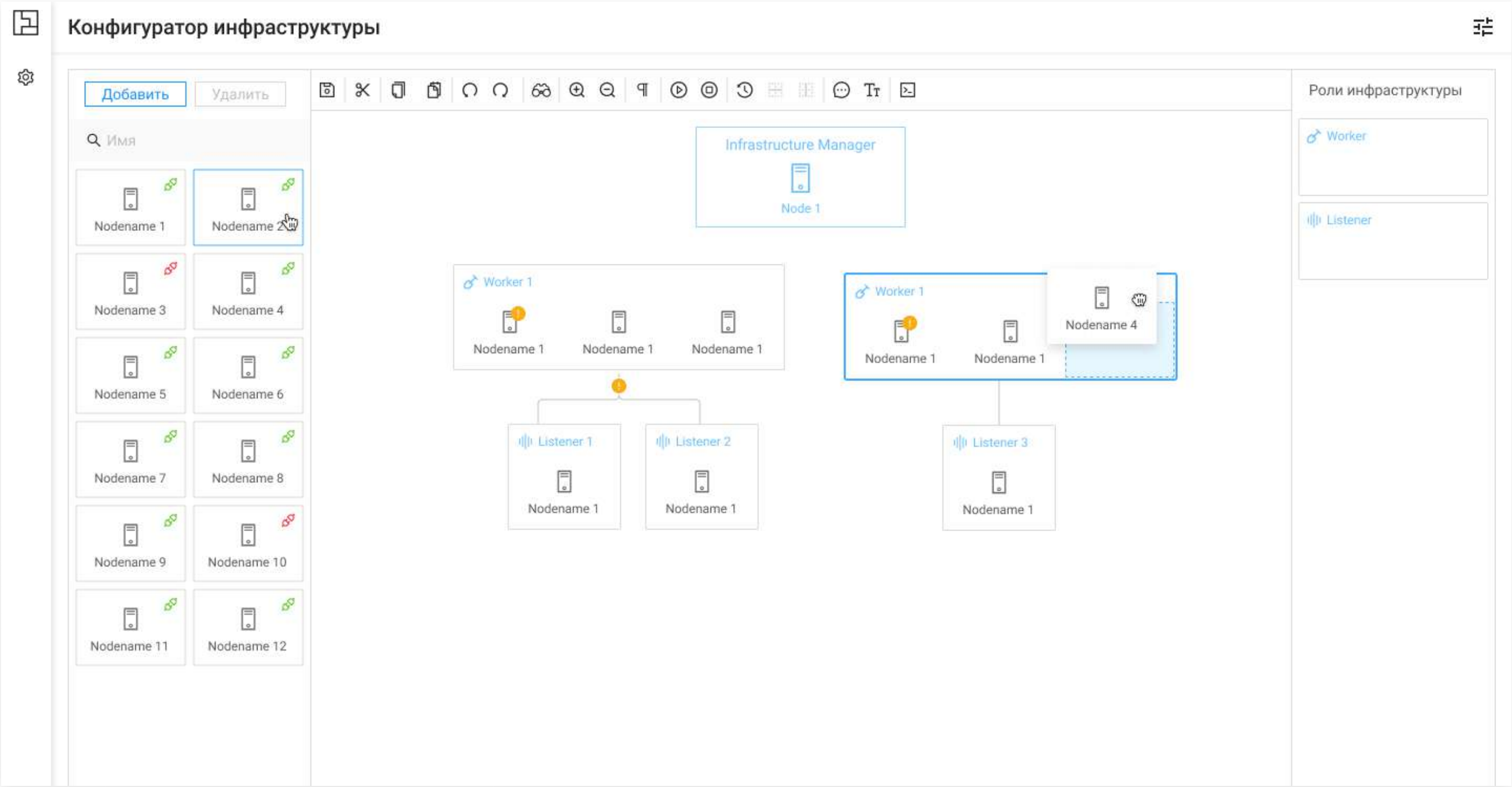
- Базы данных: Oracle, PostgreSQL, MS SQL, DB2, MySQL, MongoDB, Redis
- HTTP REST/SOAP
- SNMP walks
- Command line interfaces SSH/Telnet
- Telegram
- E-mail

3. Поддерживаемые типы активаторов

- Выполнение по расписанию
- HTTP REST/SOAP
- SNMP traps
- Telegram
- E-mail

4. Реализация сценариев применения

4.1. Подготовка вычислительной инфраструктуры



4. Реализация сценариев применения

4.2. Подключение источников данных

Источники данных

Добавить

Удалить

Имя

Тип

Статус

mds_sandbox_psgsql

PostgreSQL

case_4

HTTP REST

telebot_gelarm

Не указан

srt_collector

PostgreSQL

mds_sandbox_psgsql

case_4

telebot_gelarm

srt_collector

Тип источника данных

Тип источника данных

PostgreSQL

Общие

port

5432

hostname

192.168.88.12

db_name

srt_monitoring

Аутентификация

login

srt

password

.....

Дополнительные параметры подключения (опционально)

4. Реализация сценариев применения

4.3. Реализация сценариев обработки информации

Сценарии автоматизации

Добавить

Удалить

Имя

test3

snmp_rules

CASE_4

CASE_3

CASE_1

Новый сценарий

Widget_HTTP_API

CASE_2

CASE_3

Widget_HTTP_API

1 import json

2 import re

3

4 print(request_type)

5 print(request_path)

6 print(request_headers)

7 print(request_data)

8

9 return_result_ = None

10 request_path_p = re.split('\\/', request_path)

11 print(f"request_path_p: {len(request_path_p)}")

12 for request_path_ in request_path_p:

13 | print(request_path_)

14

15 if request_type == 'POST':

16 | res = []

17 | print('POST')

18

19 | if request_path.lower().find("/scenario/widget/") == 0:

20 | | print('/scenario/widget/')

21 |

22 | | if len(request_path_p) >= 4:

23 | | | widget_type = request_path_p[3]

24 | | | print(f"widget_type: {widget_type}")

25

Журнал выполнения сценария

4. Реализация сценариев применения

4.4. Настройка активаторов

Активаторы

Добавить

Удалить

Имя

Тип

Состояние

CASE_4

✓

Test_NEW_HTTP

HTTP

✗

HTTP_CASE_2

HTTP

✗

HTTP_CASE_1

HTTP

✗

HTTP_CASE_3

HTTP

✗

test

test

✗

WIDGET_REST_API

HTTP_0.1

✗

snmp - 192.168.88.61:162

snmp_listener

✗

Activator_Cron

✓

CASE_4

Activator_Cron

Тип активатора

Расписание

Сценарий

test3

Таймаут выполнения сценария, сек

60

Интервал запуска, сек

30

☐

Автозапуск

5. Сценарии применения

5.1. Предоставление REST API на базе обработки данных из БД



Сценарий обработки

```
result_data = mds_sandbox_psgsql.execute(sql="select * FROM  
public.faultdb_alertfilter")
```

```
return_result_ = json.dumps(result_data["res"])
```



REST API

5. Сценарии применения

5.2. Запись данных в БД поступивших через REST API



Сценарий обработки

```
sql = f"INSERT INTO public.faultdb_alertfilter (name, description,  
condition) VALUES('{request_data['name']}',  
'{request_data['description']}', '{request_data['condition']}')  
RETURNING id;"
```

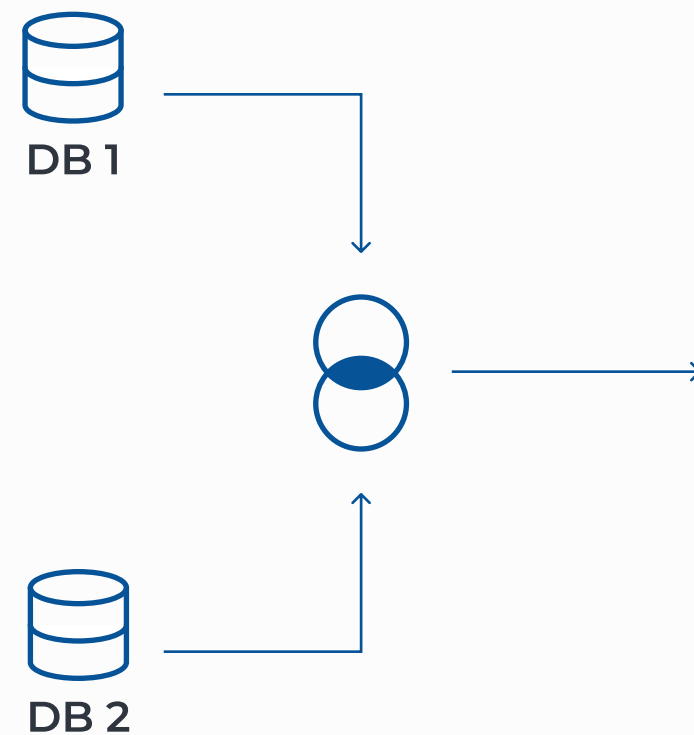
```
mds_data = mds_sandbox_psgsql.execute(sql=sql)
```

```
return_result_ = json.dumps(mds_data["res"])
```



5. Сценарии применения

5.3. Предоставление REST API на базе обработки данных из нескольких источников



Сценарий обработки

```
res = []
```

```
mds_data = mds_sandbox_psgsql.execute(sql="select * FROM  
public.faultdb_alertfilter")  
srt_collector_data = srt_collector.execute(sql=f"select * FROM  
public.stats_indicator where object_id = {request_data['id'][0]}  
fetch first 100 rows only")
```

```
for mds_data_row in mds_data["res"]:  
    for srt_collector_data_row in srt_collector_data["res"]:  
        if mds_data_row["id"] == srt_collector_data_row["object_id"]:  
            res.append({"id": mds_data_row["id"], "name":  
mds_data_row["name"], "indicator_value":  
srt_collector_data_row["indicator_value"], "ts":  
srt_collector_data_row["ts"]})
```

```
return_result_ = json.dumps(res)
```



5. Сценарии применения

5.4. Синхронизация изменений в двух базах данных через DB Link и REST API



DB 1

Check
changes

Сценарий обработки

```
last_id = get_cache('last_id')
if last_id == None:
    last_id = 0

srt_collector_data = srt_collector.execute(sql=f"select * from
public.faultdb_alertfilter where id > {last_id}")
for row in srt_collector_data["res"]:
    print(case_4.post_json(data = json.dumps(row)))
    if last_id < int(row["id"]):
        last_id = int(row["id"])

set_cache('last_id', last_id)
```



REST API



DB 2

5. Сценарии применения

5.5. Вывод отчета из БД по запросу из телеграм канала



Сценарий обработки

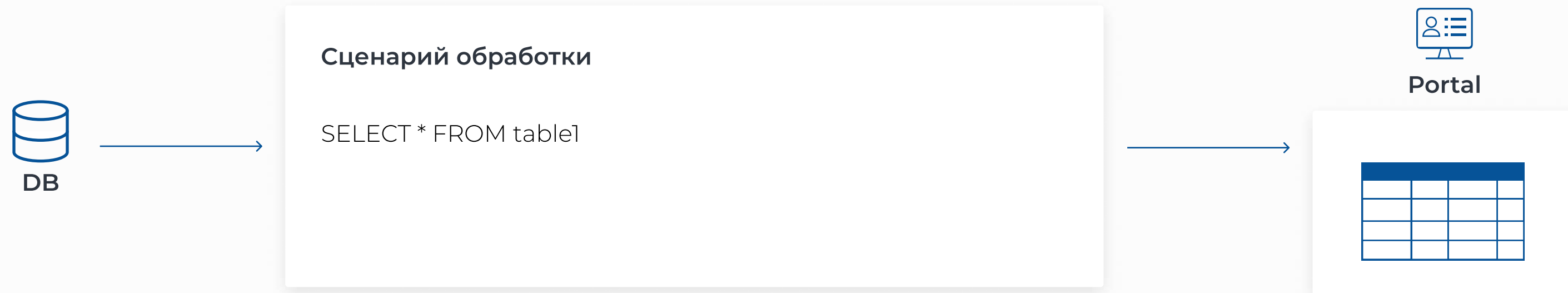
```
srt_collector_data = srt_collector.execute(sql=f"select * from  
public.faultdb_alertfilter")
```

```
return_result_ = json.dumps(srt_collector_data['res'])
```



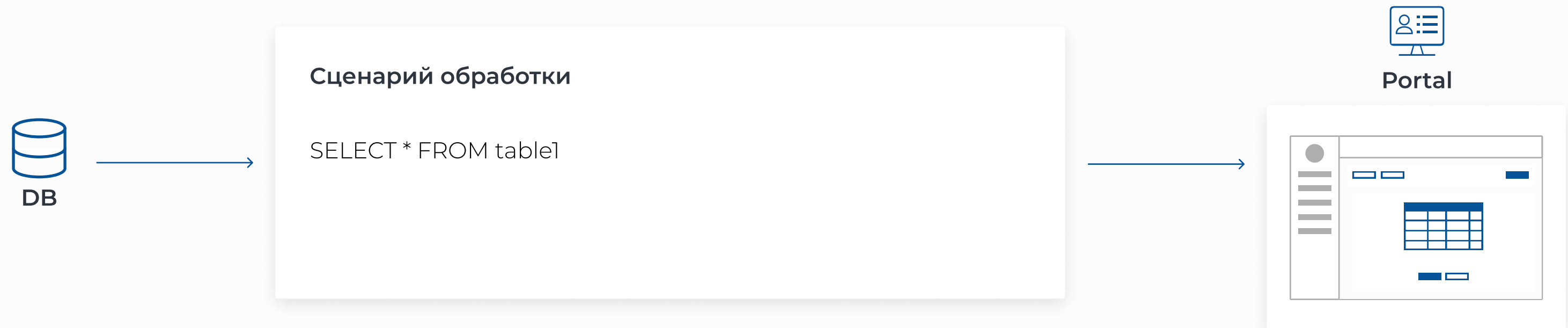
5. Сценарии применения

5.6. Вывод данных из БД в web интерфейсе пользователя в табличном виде



5. Сценарии применения

5.7. Создание управляющих инструментов (кнопки, меню ...) в web интерфейсе пользователя

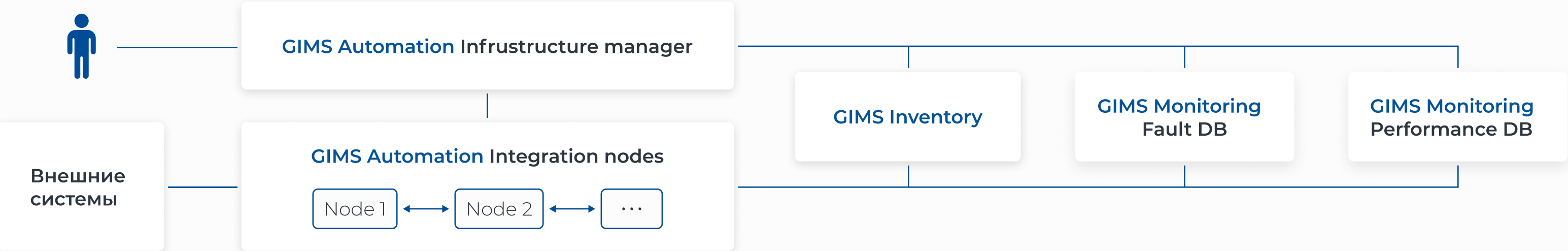


6. Документация

- Руководство пользователя
- Руководство по установке
- Руководство по устранению проблем

Документация доступна на сайте https://gelarm.ru/products/automation_product/

Протоколы системы



Компонент	Порт	Протокол	Шифрование	Назначение
GIMS Automation Integration node	25671/tcp	RabbitMQ cluster protocol	TLS	Взаимодействие узлов кластера RabbitMQ
GIMS Automation Integration node	8443/tcp	HTTPS	TLS	Системные API GIMS Automation
GIMS Automation Integration node	22/tcp	SSH	SSH	Управление
GIMS Automation Infrastructure manager	443/tcp	HTTPS	TLS	Графический интерфейс пользователя
GIMS Automation Infrastructure manager	5432/tcp	PostgreSQL	TLS	Репликация изменений настроек узлов кластера GIMS Automation
GIMS Automation Infrastructure manager	22/tcp	SSH	SSH	Управление
GIMS Automation Integration node	any (опционально)	опционально	опционально	Для взаимодействия с внешними системами
GIMS Inventory	5432/tcp	PostgreSQL	TLS	Сохранение и чтение инвентарных данных
GIMS Monitoring Fault DB	5432/tcp	PostgreSQL	TLS	Сохранение и чтение событий мониторинга
GIMS Monitoring Performance DB	5432/tcp	PostgreSQL	TLS	Сохранение и чтение метрик мониторинга

Используемое программное обеспечение сторонних производителей и библиотеки

Наименование программного обеспечения	Лицензия
PostgreSQL	PostgreSQL License
Celery	BSD
RabbitMQ	Mozilla public license
Nginx	BSD license
Gunicorn	MIT license
Django	BSD license
React	MIT license



gelarm.com