# 

# 

# 

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

«ОПТИМИЗАЦИЯ ВЫПОЛНЕНИЯ SERVERLESS-ФУНКЦИЙ НА ОБЛАЧНЫХ ИНФРАСТРУКТУРАХ НА ОСНОВЕ ИМПОРТОЗАМЕЩЕНИЯ»

СОГЛАСОВАНО

Руководитель:

Власов Дмитрий Викторович

\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Исполнитель:

Величко Арсений Александрович

\_\_\_\_\_\_\_\_\_\_\_\_

(подпись исполнителя)

# 

# 

Санкт-Петербург

2025

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_9iigqmnfhgqa)

[НАЗНАЧЕНИЕ РАЗРАБОТКИ 4](#_62lf5uv3qjez)

[Цель разработки 4](#_hsymp0gmdt1r)

[Обоснование необходимости разработки 5](#_vxmr6mxvyqwu)

[ОБЩИЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ИЗДЕЛИЮ 6](#_pkj66iue4p7w)

[Требования к функциональным характеристикам 6](#_olxshk2d3vp2)

[Требования к информационной и программной совместимости 6](#_rai4x7gtp32o)

[ТРЕБОВАНИЯ К БЕЗОПАСНОСТИ И ЗАЩИТЕ ИНФОРМАЦИИ 7](#_j034obanukc2)

[ОГРАНИЧЕНИЯ 8](#_v9mdyiqjaxzk)

[СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 9](#_9cpo2al0ah2v)

[Рис. 1 Диаграмма Ганта 10](#_oy8qlqsy0jy0)

[ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 11](#_c7s96rd9w5ji)

[Контроль разработки 11](#_1ailfwk5hewz)

[Порядок приемки 11](#_rogvv461yxvk)

[ПРИЛОЖЕНИЕ А 12](#_hg162ijo2u64)

[ПРИЛОЖЕНИЕ Б 13](#_asm6qoftqe25)

[ПРИЛОЖЕНИЕ В 14](#_ff4g1yp1hmry)

[ПРИЛОЖЕНИЕ Г 15](#_ary9fosixhmg)

[ПРИЛОЖЕНИЕ Д 16](#_2sb6xikaj5jk)

# 

# 

# ВВЕДЕНИЕ

Настоящее техническое задание разработано в целях проведения исследования, направленного на оптимизацию выполнения serverless‑функций в облачных инфраструктурах с учетом принципов импортозамещения. В условиях динамично изменяющейся политической, экономической и регуляторной обстановки в Российской Федерации актуализируется применение отечественных сервисов, что способствует снижению операционных рисков и уменьшению зависимости от зарубежных поставщиков.

Предлагается использовать отечественные сервисы Yandex Cloud Functions и Advanced FunctionGraph в качестве альтернативы зарубежным AWS Lambda и Microsoft Azure Functions. Поскольку тарификация работы раннеров осуществляется посекундно, это обуславливает необходимость оптимизации времени «холодного старта», времени выполнения, потребления оперативной памяти и сетевого трафика. Конечной целью разработки является снижение финансовых затрат и повышение скорости обработки клиентских запросов.

# 

# НАЗНАЧЕНИЕ РАЗРАБОТКИ

## Цель разработки

Цель состоит в разработке методологий и практических рекомендаций, применение которых позволит реализовать комплекс мер по оптимизации serverless‑функций в облачных инфраструктурах с учетом специфики импортозамещения.

Разработка предусматривает реализацию алгоритмов и методик, позволяющих снизить время и ресурсоемкость выполнения функций, что в свою очередь способствует уменьшению операционных расходов и повышению качества обслуживания конечных пользователей.

Предложенные рекомендации должны быть применимыми в контексте типовых сценариев использования serverless-функций, таких как:

1. Web API и микросервисная архитектура: использование serverless-функций для построения масштабируемых API и микросервисов, обеспечивающих гибкое распределение нагрузки и оптимизацию ресурсов.
2. Обработка событий и автоматизация: применение serverless-функций для реактивной обработки событий, таких как изменения в базе данных или загрузка файлов, что способствует автоматизации бизнес-процессов.
3. Пакетная и потоковая обработка данных: использование serverless-функций для выполнения ETL-процессов, трансформации и агрегации данных, позволяющих интегрировать разнородные источники информации.
4. Фоновая обработка и выполнение задач: применение serverless-функций для реализации асинхронных и периодических операций, что снижает нагрузку на основные серверные ресурсы.
5. Обработка данных в IoT-системах: использование serverless-функций для управления, анализа и хранения данных, поступающих с устройств интернета вещей, что обеспечивает оперативное принятие решений и оптимизацию затрат на инфраструктуру.

UML-диаграмма гипотетических user-story в рамках каждого сценария предложена в Приложениях А-Д.

## Обоснование необходимости разработки

Обоснование необходимости разработки базируется на тенденциях современной информационной безопасности и необходимости адаптации ИТ‑систем к требованиям национальных стандартов. Приоритет отечественных сервисов становится стратегически важным фактором в условиях изменяющейся внешней среды, что требует разработки специализированного программного обеспечения, способного обеспечить конкурентоспособность и стабильность функционирования облачных вычислений.

# 

# ОБЩИЕ ТРЕБОВАНИЯ К ПРОГРАММНОМУ ИЗДЕЛИЮ

## Требования к функциональным характеристикам

Требования к функциональным характеристикам предусматривают возможность оптимизации основных параметров serverless‑функций, а именно времени холодного старта, времени выполнения, объема потребляемой оперативной памяти и сетевого трафика в контексте типовых сценариев применения. Конечный продукт должен демонстрировать высокую адаптивность к типовым сценариям использования облачных вычислительных ресурсов и обеспечивать возможность масштабирования с учетом динамики нагрузки.

## Требования к информационной и программной совместимости

Требования к информационной и программной совместимости определяют необходимость интеграции разрабатываемого продукта с отечественными облачными сервисами Yandex Cloud Functions и Advanced FunctionGraph, а также соблюдения установленных стандартов обмена данными, что обеспечивает беспрепятственное взаимодействие с существующими информационными системами и повышает устойчивость решения к внешним воздействиям.

# 

# ТРЕБОВАНИЯ К БЕЗОПАСНОСТИ И ЗАЩИТЕ ИНФОРМАЦИИ

Конечный продукт должен соответствовать установленным нормам и требованиям по обеспечению безопасности информации, действующим в Российской Федерации, включая требования федерального закона "О персональных данных" от 27.07.2006 N 152-ФЗ. Решение должно гарантировать защиту данных от несанкционированного доступа, обеспечение их целостности и конфиденциальности, а также реализацию механизмов контроля доступа и аутентификации пользователей. При разработке необходимо учитывать требования по криптографической защите и предотвращению возможных угроз, что позволит минимизировать риски утечки информации и повысить уровень доверия пользователей к системе.

# 

# ОГРАНИЧЕНИЯ

Разработка продукта осуществляется с учетом технологических и тарифных ограничений облачных платформ, используемых в отечественной инфраструктуре. Система должна функционировать в условиях ограниченного объема оперативной памяти, времени выполнения и количества обращений к сервисам, что накладывает требования на оптимизацию алгоритмов и методов обработки данных. Особое внимание уделяется необходимости соблюдения требований нормативных документов, регламентирующих деятельность в области информационных технологий и обеспечения информационной безопасности.

Предложенные методы оптимизации должны быть применимы к типовым сценариям использования serverless-функций в рамках сервисов Yandex Cloud Functions и Advanced FunctionGraph.

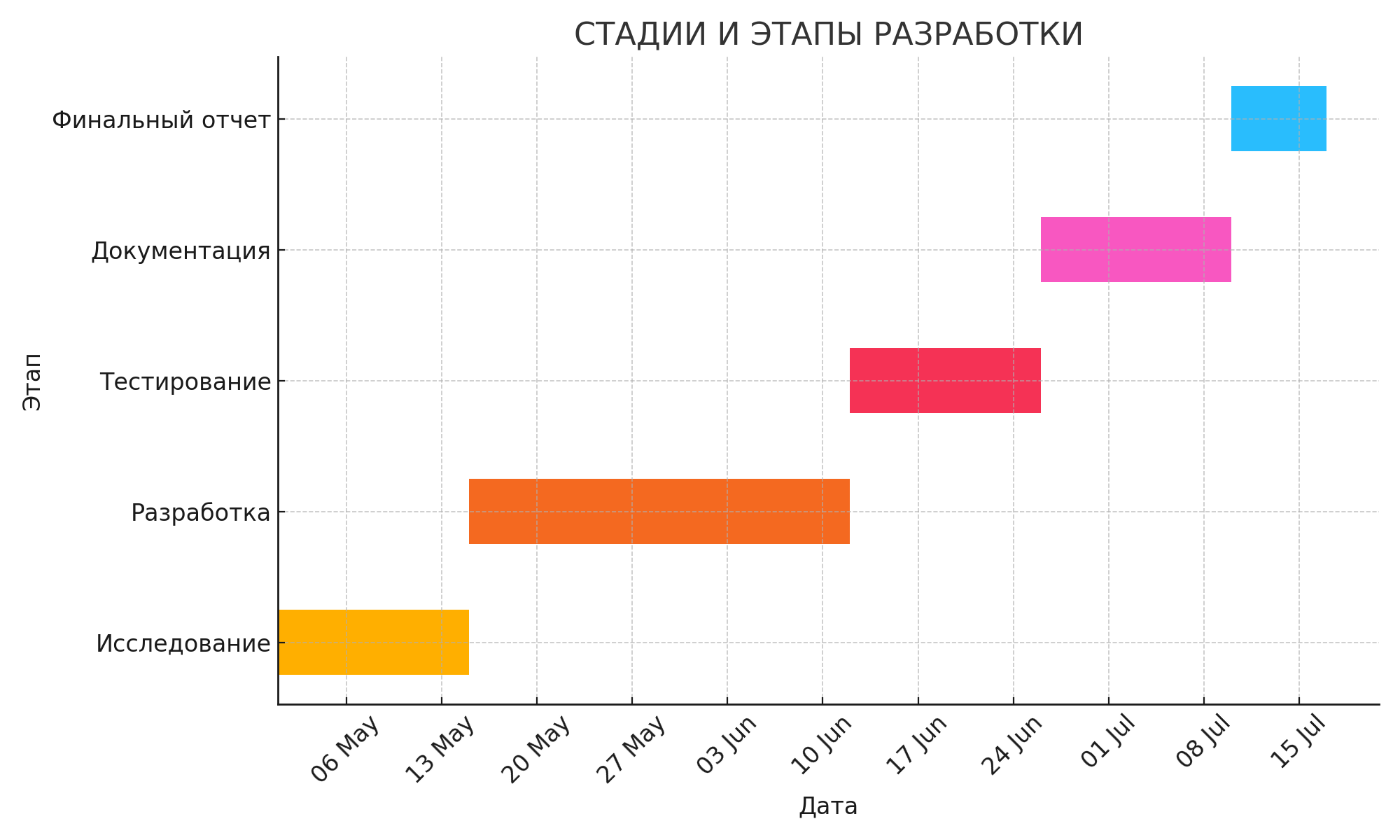
# 

# СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Разработка проекта будет осуществляться в рамках пяти спринтов, каждый из которых включает несколько этапов разработки и тестирования.

| **Стадия** | **Этап** |
| --- | --- |
| Исследование (спринт 1, продолжительность 2 недели) | Анализ типовых сценариев использования serverless-функций |
| Определение основных направлений оптимизации |
| Определение основных критериев сравнения методов |
| Разработка (спринт 2, продолжительность 4 недели) | Разработка методов оптимизации |
| Создание различных фрагментов кода для сравнения методов |
| Создание графиков и отчетов |
| Тестирование (спринт 3, продолжительность 2 недели) | Разработка тестовых сценариев для сравнения эффективности методов оптимизации |
| Тестирование предложенных методов оптимизации |
| Создание документации (спринт 4, продолжительность 2 недели) | Подготовка описания методов и их реализации |
| Описание примеров специфичных для конкретных платформ |
| Подготовка отчета о проделанной работе (спринт 5, продолжительность 1 неделя) | Создание финального отчета |
| Описание основных методов |
| Описание критериев сравнения и проведенных экспериментов |
| Написание выводов и заключения по проведенным исследованиям |

Этапы в виде диаграммы ганта (начало работ - 1 мая 2025):



# Рис. 1 Диаграмма Ганта

# 

# ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

## Контроль разработки

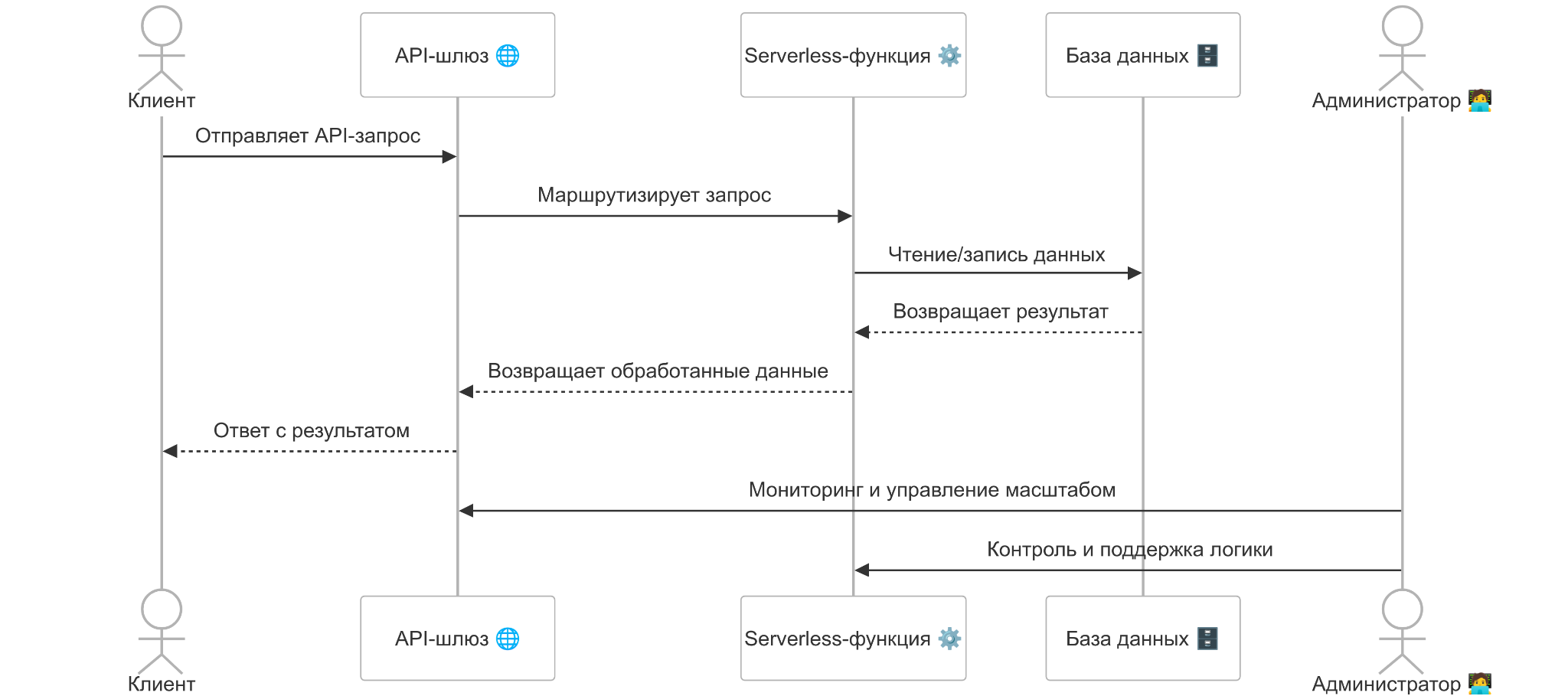
В процессе контроля разработки осуществляется комплексная проверка выполнения задач, которая включает промежуточный контроль, тестирование и проверку безопасности. После завершения каждого спринта проводится оценка выполненных задач, что позволяет оперативно выявлять и устранять возможные несоответствия установленным требованиям. Тестирование функционала охватывает проверку эффективности предложенных рекомендациях в контексте типовых сценариев использования serverless-функций, что обеспечивает всестороннюю оценку эффективности и применимости предложенных оптимизаций в рамках отечественных информационных систем. Наряду с этим проводится проверка безопасности, предусматривающая соответствие решения современным требованиям защиты информации, в том числе защиту данных пользователей.

## Порядок приемки

Порядок приемки реализуется посредством поэтапного контроля, что позволяет на каждом этапе разработки вносить необходимые корректировки. После завершения каждого спринта проводятся промежуточные проверки, на основании которых определяется готовность продукта к переходу на следующий этап. По завершении всех этапов разработки выполняется финальное тестирование, которое подтверждает соответствие конечного продукта всем техническим требованиям, установленным в техническом задании, что обеспечивает уверенность в его надежности и функциональности.

# 

# ПРИЛОЖЕНИЕ А



UML диаграмма гипотетической user-story в рамках сценария применения “Web API и микросервисная архитектура”

## 

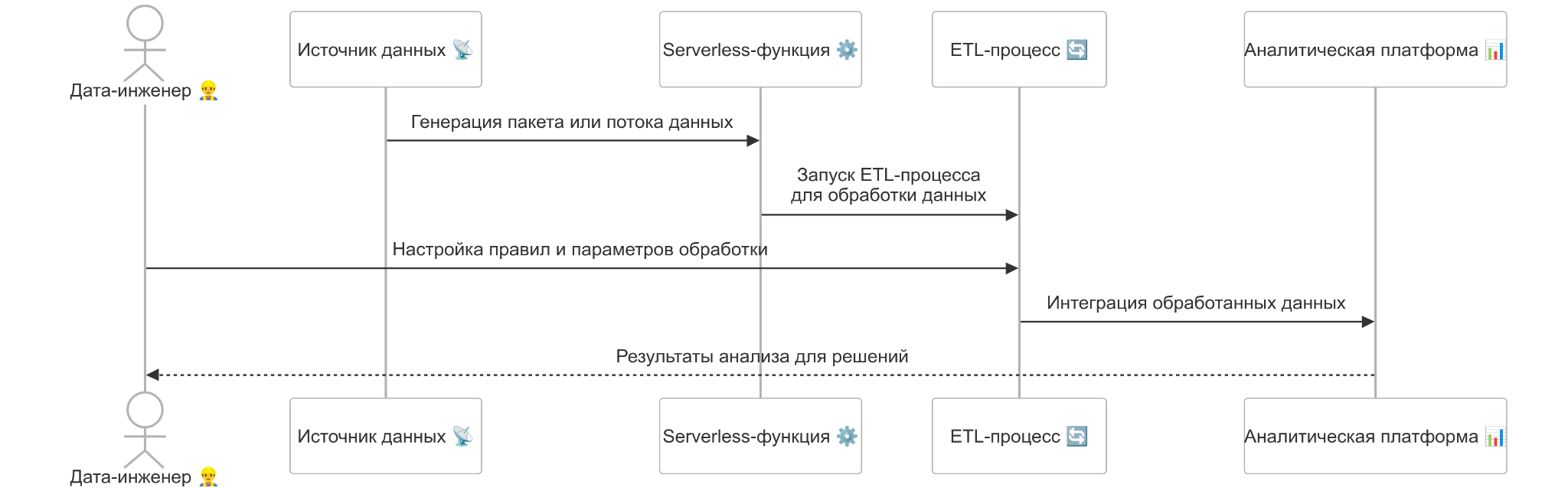
# ПРИЛОЖЕНИЕ Б



UML диаграмма гипотетической user-story в рамках сценария применения “Обработка событий и автоматизация”

## 

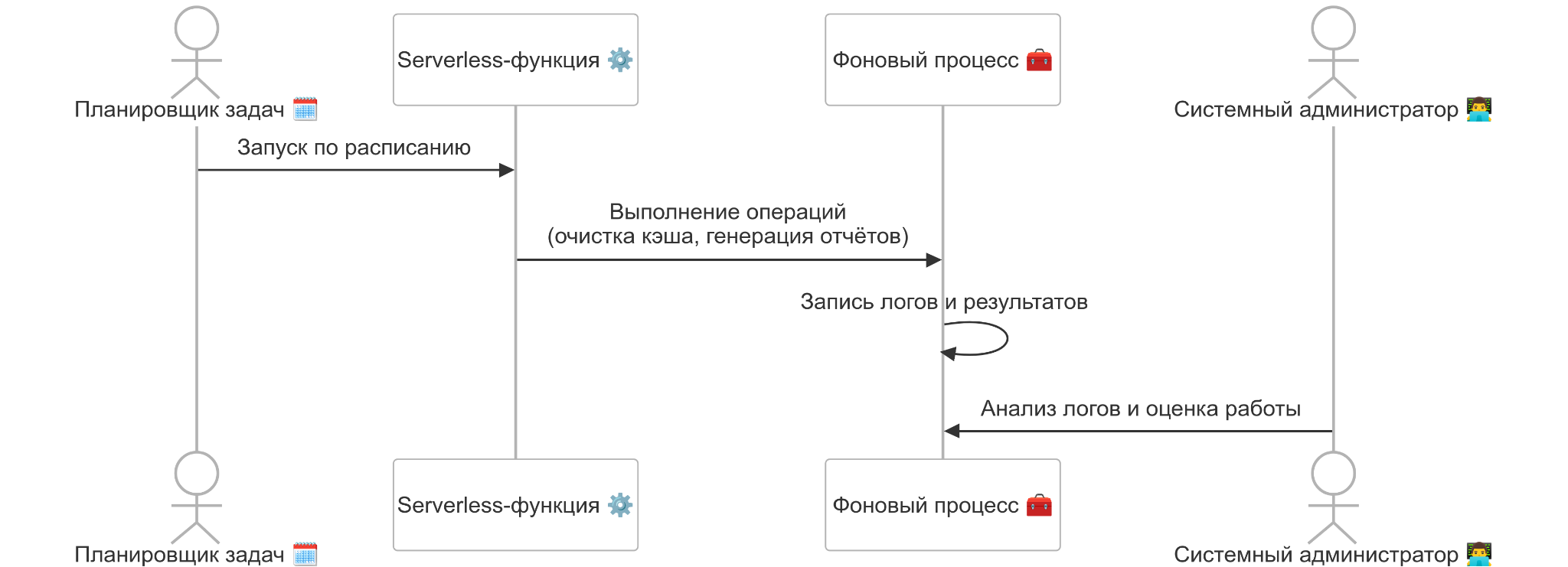
# ПРИЛОЖЕНИЕ В



UML диаграмма гипотетической user-story в рамках сценария применения “Пакетная и потоковая обработка данных”

## 

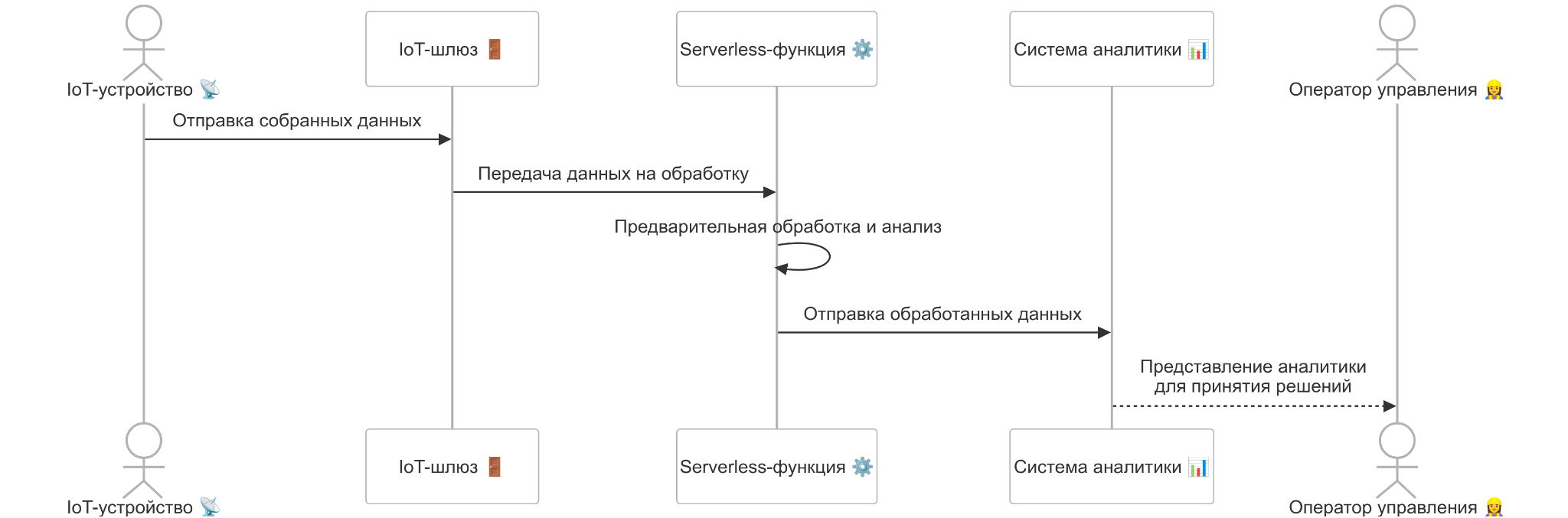
# ПРИЛОЖЕНИЕ Г



UML диаграмма гипотетической user-story в рамках сценария применения “Фоновая обработка и выполнение задач”

## 

# ПРИЛОЖЕНИЕ Д



UML диаграмма гипотетической user-story в рамках сценария применения “Обработка данных в IoT-системах”