# Практическая работа. Лекция 3. Анализ требований и подготовка архитектурному решению

## Общий бизнес-контекст

Компания “Soft Banking” представляет собственное уникальное решение для автоматизации розничного бизнеса на основе программного комплекса 3CR.

Во многих банках, которым компания поставляет свое ПО, существует необходимость отправлять уведомления своим клиентам: по операциям, новостям, акциям и др. Банки-клиенты пользуются сторонними провайдерами услуг для отправки сообщений.

Был проанализирован трафик и стоимость отправки одного сообщения по разным каналам связи: sms, whatsup, telegram. Руководство компании посчитало целесообразным войти на рынок услуг по отправке сообщений.

## Решение А. Собственный продукт. Монолит.

### Бизнес-контекст

На первом этапе было принято решение о создании программного обеспечения по отправке sms-сообщений с возможностью дальнейшего функционального расширения(отправка сообщений в whatsup, telegram), поддержать взаимодействие сервиса с существующими программными продуктами компании.

### Бизнес-цели и бизнес-драйверы

Цель: создать сервис по отправке sms-сообщений, это позволит компании выйти на рынок услуг по отправке сообщений, продать данную услугу всем действующим банкам-клиентам, дифференцировать источники доходов – доля выручки от нового сервиса должна составить к концу 2 года эксплуатации 10% от всех доходов.

Бизнес-драйверы:

* банки-клиенты нуждаются в сервисах по отправке смс-уведомлений о результатах выполненных операций, акциям, новостям и т.п.;
* дифференцировать источники дохода;
* у некоторых банков-клиентов существуют проблемы, связанные с интеграцией их банковских продуктов с провайдерами смс-уведомлений - что-то исправляется руками, нет полной автоматизации;
* расширить свою деятельность за пределами банковской сферы;
* есть заказчик, который хочет включить в свою инфраструктуру сервис по отправке смс-сообщений.

### Стейкхолдеров и их потребности

Клиенты банка – клиенты банка, которые подписались на получение уведомлений от банка;

Банк, который предоставляет своим клиентам подписку на получение уведомлений об операциях, акциях, новостях и т.п;

Клиенты сервиса(помимо банков), должны иметь API с описанием для взаимодействия с сервисом;

Администратор смс-сервиса должен иметь интерфейс для удобного конфигурирования, управлением безопасностью программы;

Служба поддержки клиентов должна иметь инструменты по мониторингу отправленных сообщений;

Операционист – служащий банка, который обслуживает клиента банка.

### Пользовательские истории

UC-1: Клиенты банка оплачивают покупки в магазине, после выполнения транзакции, на телефон клиента должно прийти уведомление о покупке с указанием суммы;

UC-2: При входе клиента в приложение банк-клиент, после того как пользователь указал валидный логин и пароль, банк через смс-сервис отправляет клиенту сообщение с кодом для авторизации его в программе;

UC-3: Регистрация клиентов смс-сервиса через API;

UC-4: Администратор системы может посмотреть список клиентов смс-сервиса, изменит их роль, заблокировать или предоставить доступ к сервису;

UC-5: Банки и другие клиенты смс-сервиса могут получить статусы отправленных сообщений;

UC-6: Служба поддержки банков, клиенты могут получить статистические данные по отправке своих сообщений в виде отчетов, графиков;

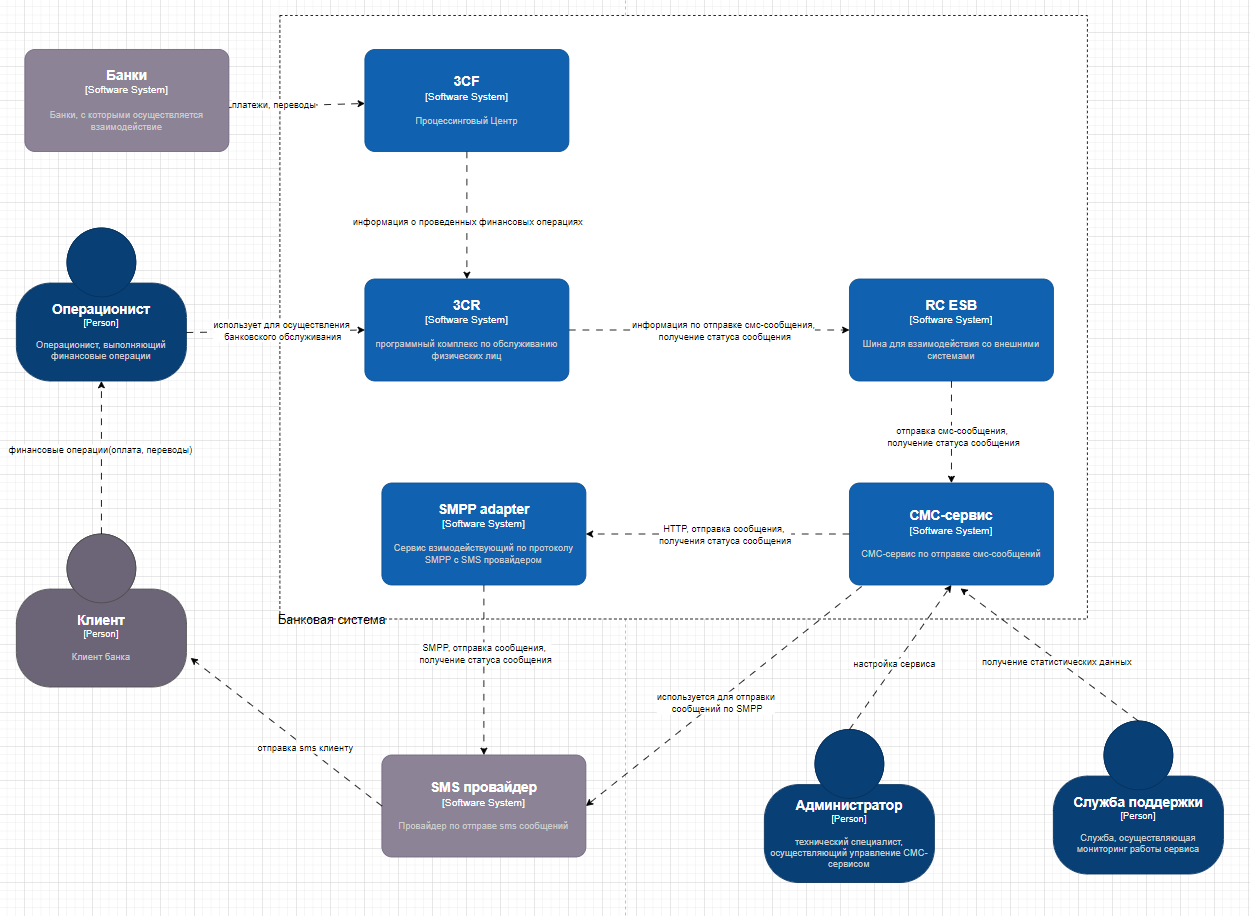
UC-7: Администратор системы через веб-интерфейс может регистрировать, изменять, удалять провайдеров.

### Атрибуты качества (и не функциональные требования)

НФТ:

1. Сервис должен выдерживать увеличение количества запросов с 10 запросов в секунду до 50000.
2. Возможность восстановления системы в течение 1 часа с сохранением всех данных.
3. Обеспечить квантиль пользовательских запросов 90 процентиль время выполнения запросов(latency) 700 миллисекунд.
4. Обеспечить защищенный канал взаимодействия с сервисом.
5. Возможность обнаруживать подозрительный трафик и автоматически блокировать их источники - защищать сайт от DDoS атак.
6. Использовать защищенные технологии аутентификации и авторизации с возможностью проверки подлинности пользователя и разрешенных ему операций.
7. Фиксировать все запросы, ответы при взаимодействии с другими системами.
8. Обеспечить понятную и легкую конфигурируемость системы через внешние настройки.
9. Фиксировать все запросы в журнале.
10. Обеспечить идемпотентность запросов.

### Контекстная схема системы



### Критические сценарии и критические характеристики

Критические сценарии:

* Сервис должен работать, даже если произошел разрыв связи c SMS провайдером. Все поступающие сообщения сохранить и при восстановлении связи с SMS провайдером еще раз отправить полученные сообщения;
* При падении сервиса перенаправлять запросы на резервный экземпляр сервиса;
* В случае прекращения дальнейшего сотрудничества(взаимодействия) с текущим SMS провайдером, иметь возможность в короткие сроки и с минимальными затратами переключить взаимодействие сервиса на другого провайдера.

Критические характеристики:

* Доступность сервисов и БД. Процент ошибочных ответов от сервиса и БД не должен превышать 1 %;
* Время ответа от сервиса 92 процентиль квантиль клиентских запросов время выполнения запросов(latency) не должен превышать 2,5с;
* Обеспечить автоматическую повторную отправку сообщений, в случае неудачной предыдущей попытки;
* Время разработки;
* Стоимость разработки.

### ADR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | status | context | decision | responsibility party | decision date | comments |
| D001 | Принято | Технический | Монолит | Костя | 02.02.2024 | Предложил создать монолитное приложение |
| D002 | Отклонено | Технический | Использовать расширение базового протокола системы 3CR | Марат | 02.02.2024 | Взять за основу существующую систему xsd |
| D003 | Принято | Технический | создать отдельную схему xsd для описания протокола | Леня | 03.02.2024 |  |
| D004 | Отклонено | Технический | Поддержка json | Саша | 04.02.2024 |  |
| D005 | Принято | Технический | Поддержка json и xml | Костя | 04.02.2024 | Формат обмена сообщениями с сервисом |
| D006 | Отклонено | Бизнес | Взаимодействие сервиса с провайдером по HTTP протоколу | Саша | 12.02.2024 | У каждого SMS провайдера свой формат сообщений, который сложно адаптировать в случае смены провайдера |
| D007 | Принято | Бизнес | Взаимодействие сервиса с провайдером по SMPP протоколу | Костя | 12.02.2024 | SMPP протокол – стандарт, описывающий определенный формат обмена сообщениями и который поддерживают многие SMS провайдеры |

## Решение Б. Собственный продукт. Микросервисы.

### Бизнес-контекст

На первом этапе было принято решение о создании программного обеспечения по отправке sms-сообщений с возможностью дальнейшего функционального расширения(отправка сообщений в whatsup, telegram), поддержать взаимодействие сервиса с существующими программными продуктами компании.

Предполагается резкое увеличение количества запросов с подключением к системе юридических лиц, поэтому система должна быть масштабируемой, устойчивой к различным нагрузкам.

Необходимо обеспечивать независимую разработку и развитие отдельных частей, чтобы модули(сервисы) системы можно было разрабатывать и поддерживать независимым образом.

### Бизнес-цели и бизнес-драйверы

Цель: создать сервис по отправке sms-сообщений, это позволит компании выйти на рынок услуг по отправке сообщений, продать данную услугу всем действующим банкам-клиентам, дифференцировать источники доходов – доля выручки от нового сервиса должна составить к концу 2 года эксплуатации 10% от всех доходов.

Бизнес-драйверы:

* Банки-клиенты нуждаются в сервисах по отправке смс-уведомлений о результатах выполненных операций, акциям, новостям и т.п.;
* Дифференцировать источники дохода;
* У некоторых банков-клиентов существуют проблемы, связанные с интеграцией их банковских продуктов с провайдерами смс-уведомлений - что-то исправляется руками, нет полной автоматизации;
* Расширить свою деятельность за пределами банковской сферы;
* есть заказчик, который хочет включить в свою инфраструктуру сервис по отправке смс-сообщений.

### Стейкхолдеров и их потребности

Клиенты банка – клиенты банка, которые подписались на получение уведомлений от банка;

Банк, который предоставляет своим клиентам подписку на получение уведомлений об операциях, акциях, новостях и т.п;

Клиенты сервиса(помимо банков), должны иметь API с описанием для взаимодействия с сервисом;

Администратор смс-сервиса должен иметь интерфейс для удобного конфигурирования, управлением безопасностью программы;

Служба поддержки клиентов должна иметь инструменты по мониторингу отправленных сообщений;

Операционист – служащий банка, который обслуживает клиента банка.

### Пользовательские истории

UC-1: Клиенты банка оплачивают покупки в магазине, после выполнения транзакции, на телефон клиента должно прийти уведомление о покупке с указанием суммы;

UC-2: При входе клиента в приложение банк-клиент, после того как пользователь указал валидный логин и пароль, банк через смс-сервис отправляет клиенту сообщение с кодом для авторизации его в программе;

UC-3: Регистрация клиентов смс-сервиса через API;

UC-4: Администратор системы может посмотреть список клиентов смс-сервиса, изменит их роль, заблокировать или предоставить доступ к сервису;

UC-5: Банки и другие клиенты смс-сервиса могут получить статусы отправленных сообщений;

UC-6: Служба поддержки банков, клиенты могут получить статистические данные по отправке своих сообщений в виде отчетов, графиков;

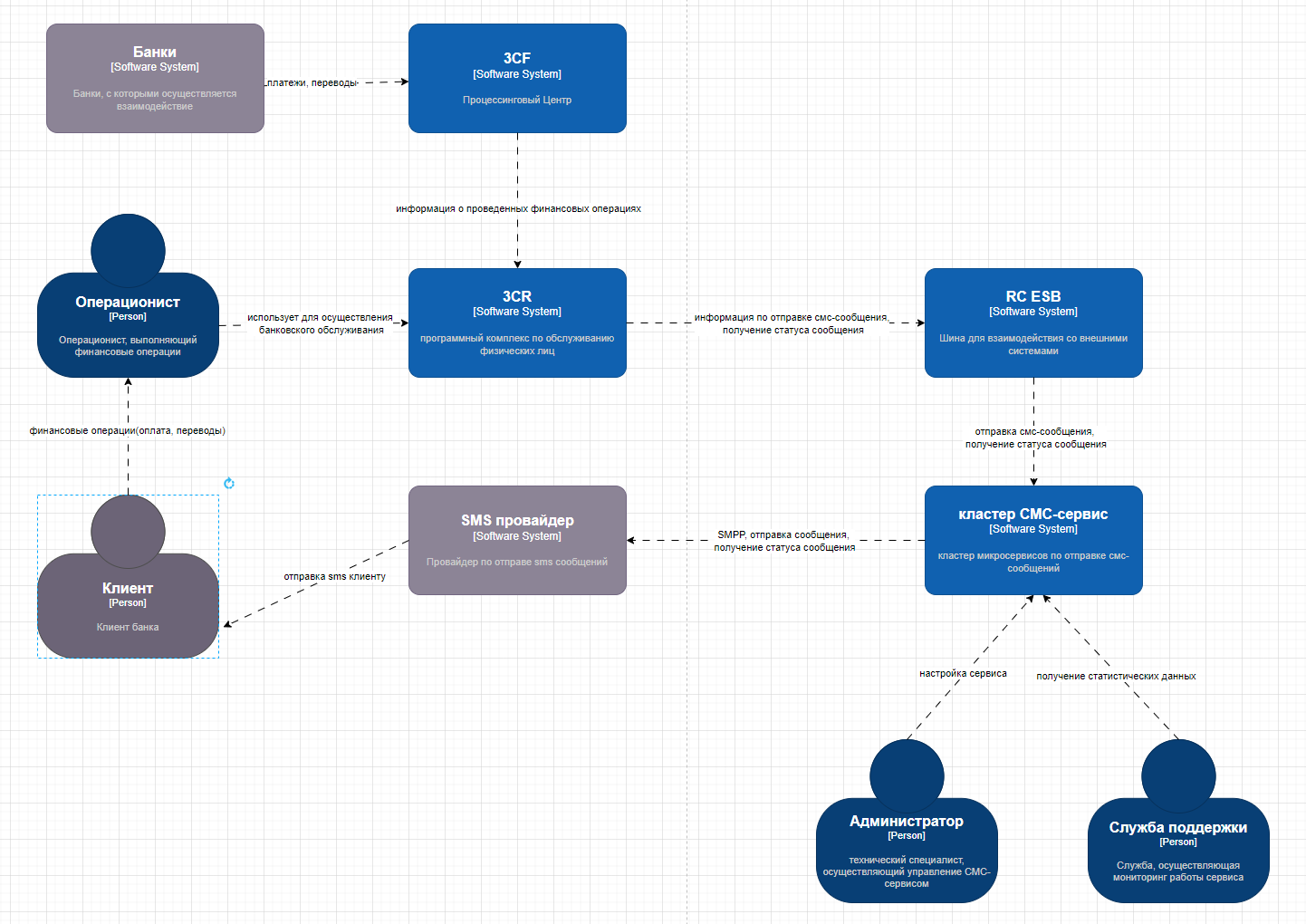
UC-7: Администратор системы через веб-интерфейс может регистрировать, изменять, удалять провайдеров.

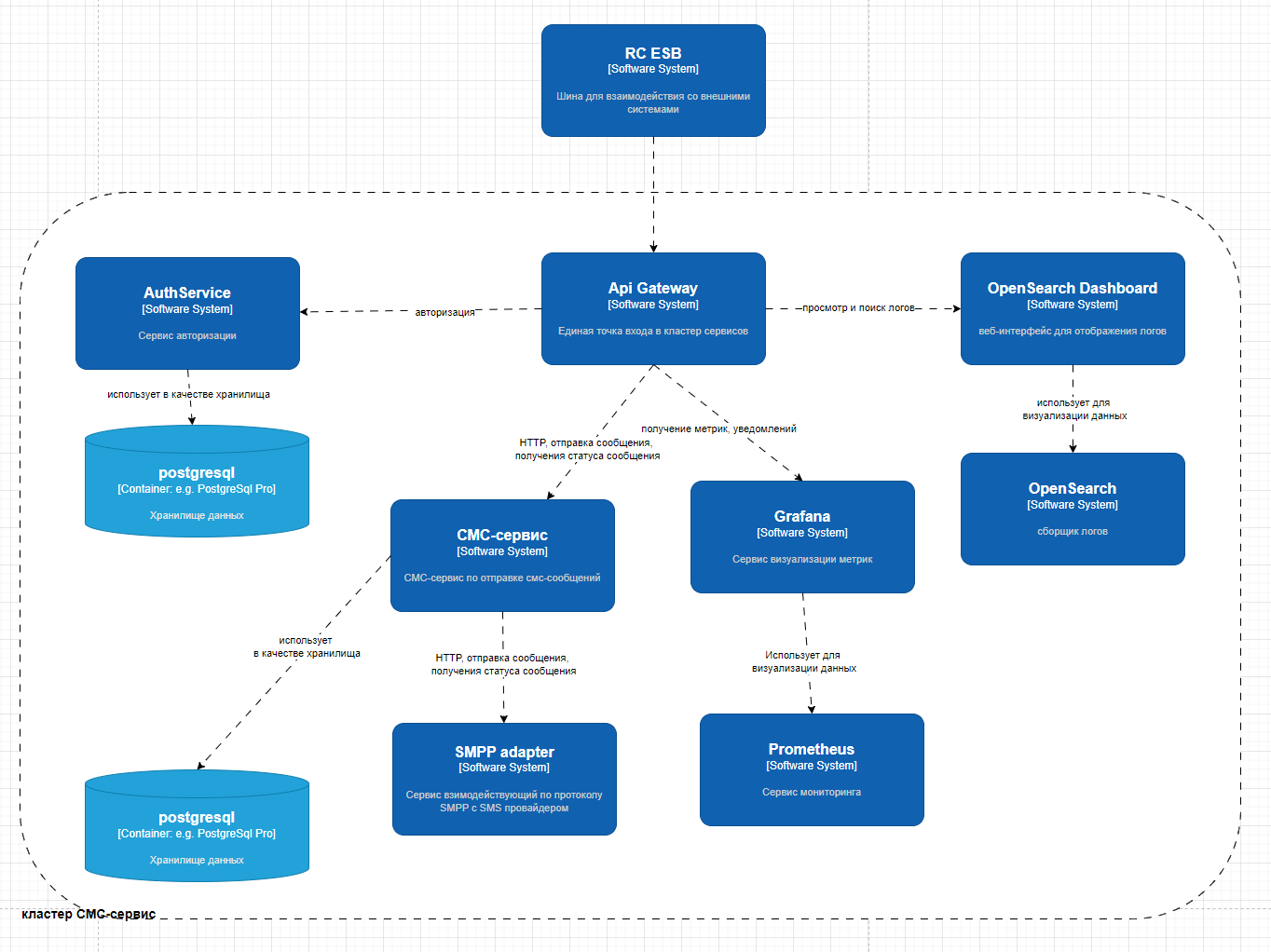
### Атрибуты качества (и не функциональные требования)

НФТ:

1. Сервис должен быть построен из микросервисов, каждый из которых может быть легко заменен, тем самым обеспечивая гибкость в сборке и поддержке системы
2. Сервис должен выдерживать увеличение количества запросов с 10 запросов в секунду до 250000. Применить горизонтальное масштабирование.
3. На сервисе должна поддерживаться возможность восстановления системы в течение 2 часов с сохранением всех данных.
4. Обеспечить квантиль пользовательских запросов 90 процентиль время выполнения запросов(latency) 500 миллисекунд.
5. Обеспечить защищенный канал взаимодействия с сервисом.
6. Возможность обнаруживать подозрительный трафик и автоматически блокировать их источники - защищать сайт от DDoS атак.
7. Использовать защищенные технологии аутентификации и авторизации с возможностью проверки подлинности пользователя и разрешенных ему операций.
8. Фиксировать все запросы, ответы при взаимодействии с другими системами.
9. В случае падения сервиса или его БД система могла бы переориентировать трафик взаимодействия с другими сервисами так, чтобы пользователь не почувствовал никаких неполадок.
10. Обеспечить понятную и легкую конфигурируемость системы через внешние настройки.
11. Поддержка масштабируемости системы при пиковых нагрузках, например, в период увеличения количества заказов.
12. Фиксировать все запросы в журнале.
13. Обеспечить идемпотентность запросов.

### Контекстная схема системы





### Критические сценарии и критические характеристики

Критические сценарии:

* Сервис должен работать, даже если произошел разрыв связи c SMS провайдером. Все поступающие сообщения сохранить и при восстановлении связи с SMS провайдером еще раз отправить полученные сообщения;
* При падении сервиса автоматически бы поднимался новый экземпляр сервиса;
* В случае прекращения дальнейшего сотрудничества(взаимодействия) с текущим SMS провайдером, иметь возможность в короткие сроки и с минимальными затратами переключить взаимодействие сервиса на другого провайдера;
* При падении БД или сервиса должно происходит переключение трафика на их реплики.

Критические характеристики:

* Доступность сервисов и БД. Процент ошибочных ответов от сервиса и БД не должен превышать 1 %;
* Время ответа от сервиса 92 процентиль квантиль клиентских запросов время выполнения запросов(latency) не должен превышать 2,5с;
* Обеспечить автоматическую повторную отправку сообщений, в случае неудачной предыдущей попытки;
* Время разработки;
* Стоимость разработки.

### ADR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | status | context | decision | responsibility party | decision date | comments |
| D001 | Принято | Технический | Микросервисы | Саша | 07.02.2024 |  |
| D002 | Принято | Бизнес | Достигнуто соглашение с банком о использовании внутреннего сервиса, взаимодействующего по протоколу SMPP | Костя | 07.02.2024 | Интеграция с смс провайдером, развернутого внутри банка |
| D003 | Отклонено | Технический | Авторизация через отдельный микросервис | Саша | 08.02.2024 | Много вопросов как это поддержать в k8s |
| D004 | Принято | Технический | Авторизация через отдельный микросервис с использованием ingress controller nginx | Саша | 08.02.2024 | Управление единой точкой входа |
| D005 | Принято | Бизнес | Развернуть всю инфраструктуру в облаке. Экономически выгодно – затраты на покупку услуг у провайдера оказалось дешевле, чем создавать и поддерживать эту инфраструктуру самим. | Костя | 10.02.2024 | Много уйдет время на поиск штата специалистов для создания и поддержки инфраструктуры k8s. Закупка серверов тоже требует существенных затрат. |
| D006 | Принято | Технический | Использовать собственную инфраструктуру на базе k8s | Саша | 10.02.2024 | Если банк или др.юридическое лицо не хочет использовать сервисы вне своей инфраструктуры, то необходимо развернуть сервисы на их стороне |
| D007 | Принято | Технический, | Использовать облачную инфраструктуру | Саша | 10.02.2024 | Использование облачной инфраструктуры позволит избавиться от покупки железа |
| D008 | Предложено | Бизнес | Предусмотреть в бюджете ресурсы, необходимые для поддержания облачной инфраструктуры | Володя | 11.02.2024 | Средства нужны для оплаты услуг облачного провайдера, труда специалистов, которые бы поддерживали эту инфраструктуру в рабочем состоянии |

## Решение В. Сторонний продукт.

### Бизнес-контекст

Использовать сторонний сервис в виде услуги по отправке смс-сообщения.

### Бизнес-цели и бизнес-драйверы

Цель: создать сервис по отправке sms-сообщений, это позволит компании выйти на рынок услуг по отправке сообщений, продать данную услугу всем действующим банкам-клиентам, дифференцировать источники доходов – доля выручки от нового сервиса должна составить к концу 2 года эксплуатации 10% от всех доходов.

Бизнес-драйверы:

* банки-клиенты нуждаются в сервисах по отправке смс-уведомлений о результатах выполненных операций, акциям, новостям и т.п.;
* дифференцировать источники дохода;
* у некоторых банков-клиентов существуют проблемы, связанные с интеграцией их банковских продуктов с провайдерами смс-уведомлений - что-то исправляется руками, нет полной автоматизации;
* расширить свою деятельность за пределами банковской сферы;
* есть заказчик, который хочет включить в свою инфраструктуру сервис по отправке смс-сообщений.

### Стейкхолдеров и их потребности

Клиенты банка – клиенты банка, которые подписались на получение уведомлений от банка;

Банк, который предоставляет своим клиентам подписку на получение уведомлений об операциях, акциях, новостях и т.п;

Клиенты сервиса(помимо банков), должны иметь API с описанием для взаимодействия с сервисом;

Администратор смс-сервиса должен иметь интерфейс для удобного конфигурирования, управлением безопасностью программы;

Служба поддержки клиентов должна иметь инструменты по мониторингу отправленных сообщений;

Операционист – служащий банка, который обслуживает клиента банка.

### Пользовательские истории

UC-1: Клиенты банка оплачивают покупки в магазине, после выполнения транзакции, на телефон клиента должно прийти уведомление о покупке с указанием суммы;

UC-2: При входе клиента в приложение банк-клиент, после того как пользователь указал валидный логин и пароль, банк через смс-сервис отправляет клиенту сообщение с кодом для авторизации его в программе;

UC-3: Регистрация клиентов смс-сервиса через API;

UC-4: Администратор системы может посмотреть список клиентов смс-сервиса, изменит их роль, заблокировать или предоставить доступ к сервису;

UC-5: Банки и другие клиенты смс-сервиса могут получить статусы отправленных сообщений;

UC-6: Служба поддержки банков, клиенты могут получить статистические данные по отправке своих сообщений в виде отчетов, графиков;

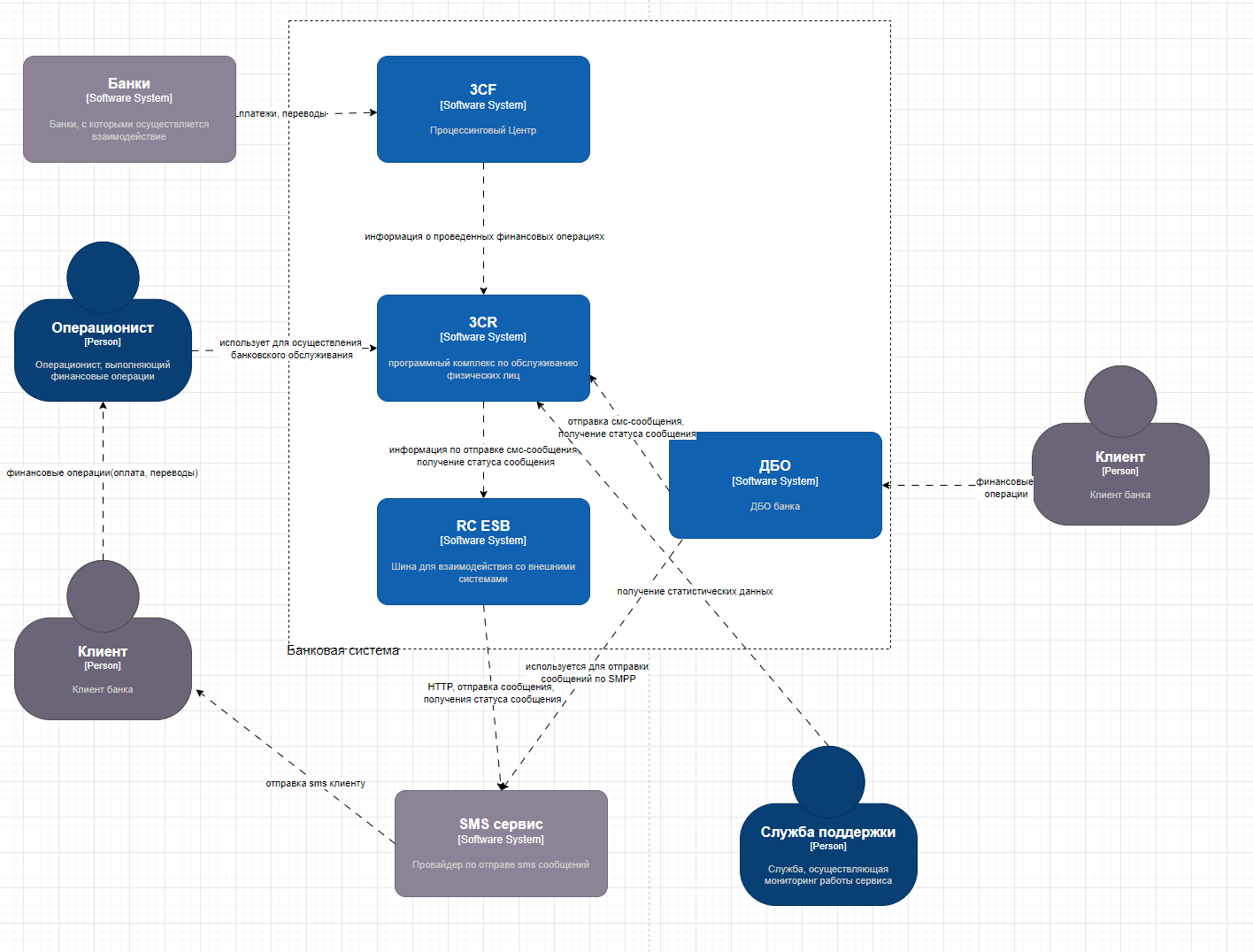
UC-7: Администратор системы через веб-интерфейс может регистрировать, изменять, удалять провайдеров.

### Атрибуты качества (и не функциональные требования)

НФТ:

1. Сервис должен выдерживать увеличение количества запросов с 10 запросов в секунду до 50000.
2. Возможность восстановления системы в течение 1 часа с сохранением всех данных.
3. Обеспечить квантиль пользовательских запросов 90 процентиль время выполнения запросов(latency) 700 миллисекунд.
4. Обеспечить защищенный канал взаимодействия с сервисом.
5. Возможность обнаруживать подозрительный трафик и автоматически блокировать их источники - защищать сайт от DDoS атак.
6. Использовать защищенные технологии аутентификации и авторизации с возможностью проверки подлинности пользователя и разрешенных ему операций.
7. Фиксировать все запросы, ответы при взаимодействии с другими системами.
8. Обеспечить понятную и легкую конфигурируемость системы через внешние настройки.

### Контекстная схема системы



### Критические сценарии и критические характеристики

Критические сценарии:

* Сервис должен работать, даже если произошел разрыв связи c SMS провайдером. Все поступающие сообщения сохранить и при восстановлении связи с SMS провайдером еще раз отправить полученные сообщения;
* При падении сервиса перенаправлять запросы на резервный экземпляр сервиса;
* В случае прекращения дальнейшего сотрудничества(взаимодействия) с текущим SMS провайдером, иметь возможность в короткие сроки и с минимальными затратами переключить взаимодействие сервиса на другого провайдера.

Критические характеристики:

* Доступность сервисов и БД. Процент ошибочных ответов от сервиса и БД не должен превышать 1 %;
* Время ответа от сервиса 92 процентиль квантиль клиентских запросов время выполнения запросов(latency) не должен превышать 2,5с;
* Обеспечить автоматическую повторную отправку сообщений, в случае неудачной предыдущей попытки;
* Изменение стоимости услуг SMS провайдера;
* Зависимость от провайдера.

# Задание со звездочкой

### Бизнес-контекст и цели

В виду того, что компания предоставляет международные услуги и сотрудничает с большим количеством компаний, находящихся в разных точках нашей планеты, она не раз сталкивалась с санкционной политикой и другими ограничениями, - встал вопрос о возможности быстрого внесения необходимых изменений в приложения компании для обхода санкций и сохранения клиентов. Доля иностранных клиентов компании, находящихся в юрисдикции других стран, составляет 82%.

Цель: иметь возможность быстро перенастраивать систему в виду меняющихся условий работы сервисов, которые могут быть сопряжены с политикой, экономическими условиями сотрудничества связанных внешних сервисов, техническими условиями ограничения использования инфраструктуры провайдера и внешних сервисов.

### Критические сценарии и критические характеристики

Критические сценарии:

* Введения ограничения обмена информацией по финансовым транзакциям с определенной группой стран не должно прекращать взаимодействие между сервисами компании и зарубежными клиентами;
* При падении БД или сервиса должно происходит переключение трафика на их реплики;
* В случае прекращения дальнейшего сотрудничества(взаимодействия) с внешними сервисами, иметь возможность в короткие сроки и с минимальными затратами переключить взаимодействие сервиса через другие сервисы;
* При падении бизнес-сервиса или сервиса обновлений автоматически бы поднимался новый экземпляр сервиса;
* Через веб-интерфейс системы управления параметрами можно было всегда просмотреть статус обновления параметров каждого сервиса, даже, если один из сервисов упал.

Критические характеристики:

* Доступность сервисов и БД. Процент ошибочных ответов от сервиса и БД не должен превышать 1 %;
* Время ответа от сервиса о получении обновлений должно составлять не более 90 секунд;
* Обеспечить автоматическую повторную отправку обновленных параметров, в случае неудачной предыдущей попытки;
* Время разработки;
* Стоимость разработки.
* Ошибки в аналитике по обходу ограничений

### Решение 1

#### Контекстная схема

СonfigService – сервис по изменению конфигурационных параметров. Конфигурационные параметры считываются с БД Redis. Конфигурационные данные хранятся в файле в формате yaml.

Периодически раз в минуту сервисы Service 1 и Service 2 считывают конфигурационные данные с БД Redis.

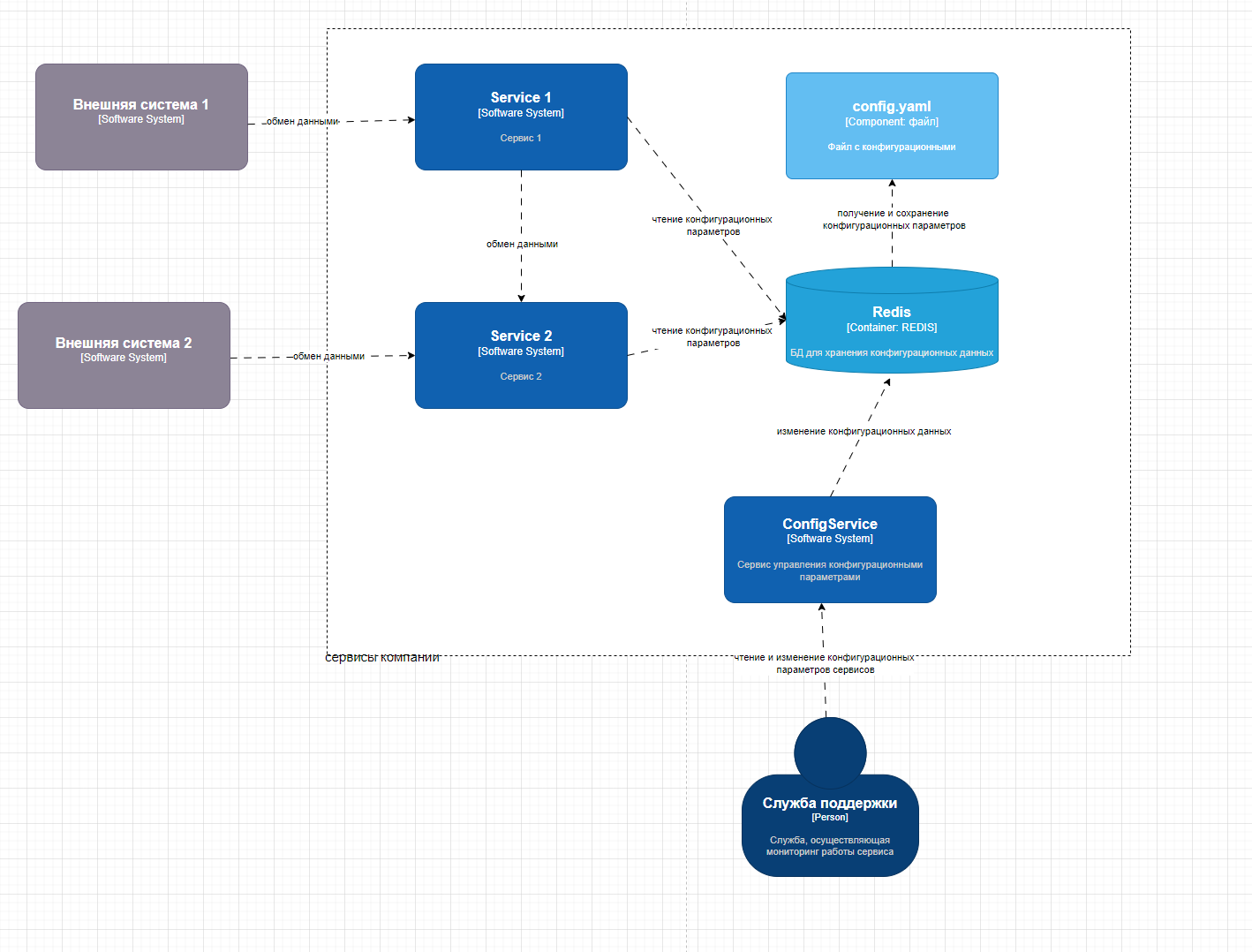


Рисунок - Контекстная схема решешния 1

#### Риски

Возможны следующие риски:

* Злонамеренный взлом – применение невалидных конфигурационных параметров. Доступ из внутренней сети к сервису никак не контролируется и ничем не ограничен.
* Возможны ошибки при работе с конфигурационным файлом;
* Если упадет БД, то данные могут потеряться;
* Нагрузка на сеть - Service 1 и Service 2 создают дополнительный трафик при запросе получения новых конфигурационных данных;
* Можем не знать о результатах применения новых конфигурационных параметров.

#### Возможности

Возможности:

* Требует небольших затрат на реализацию;
* Не требует много ресурсов для поддержания.

### Решение 2

#### Контекстная схема решения 2

ConfigService – сервис по изменению конфигурационных параметров. Конфигурационные параметры, данные необходимые для осуществления контроля доступа к сервису управления конфигурационными параметрами хранятся в БД PostgreSQL.

Периодически раз в минуту сервисы Service 1 и Service 2 считывают конфигурационные данные с БД PostgreSQL.

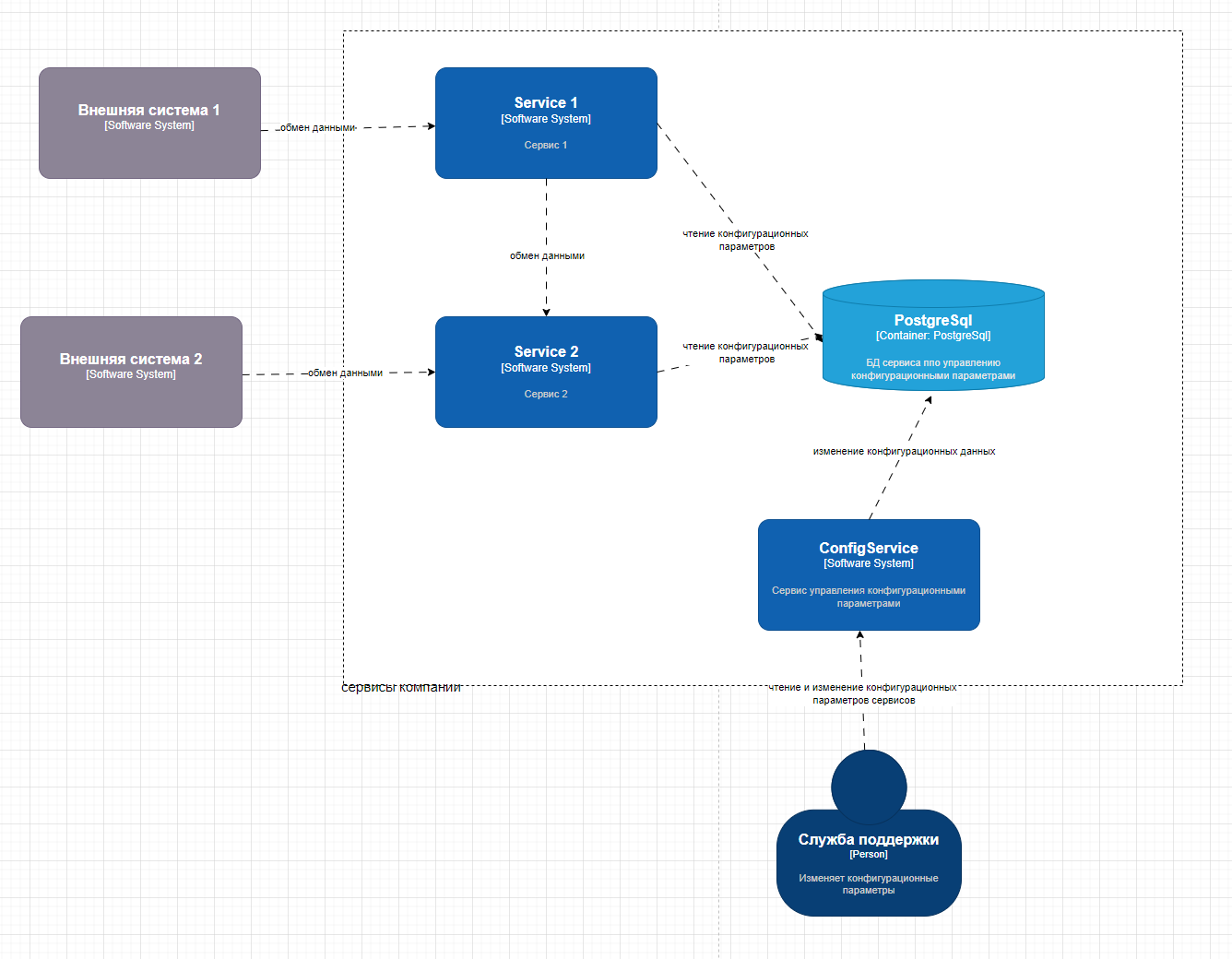


Рисунок - Контекстная схема решешния 2

#### Риски

Возможны следующие риски:

* Нагрузка на сеть - Service 1 и Service 2 создают дополнительный трафик при запросе получения новых конфигурационных данных;
* Можем не знать о результатах применения новых конфигурационных параметров.
* Может упасть БД;
* Сложно восстановить старые конфигурационные данные;

#### Возможности

Возможности:

* Доступ к сервису контролируется;
* Прост в реализации;
* Небольшая вероятность потерять данные(поднять данные из бэкапа)

### Решение 3

#### Контекстная схема решения 3

ConfigService – сервис по изменению конфигурационных параметров.

Конфигурационные параметры, данные необходимые для осуществления контроля доступа к сервису управления конфигурационными параметрами хранятся в БД PostgreSQL.

Последовательность применения новых конфигурационных данных:

* Служба поддержки входит в систему управления конфигурационными параметрами;
* Считывает текущие изменения конфигурационных параметров Service 1 и Service 2, вносит изменения и сохраняет данные;
* При внесении изменений в БД создается событие - конфигурационные данные для каждого из типов сервисов отправляются в kafka-у.
* Соответствующие сервисы слушают topic брокера сообщений, считывают и применяют новые конфигурационные данные и в другой топик отправляют ответ, сообщая успешно или неуспешно обработали новые параметры.
* Сервис ConfigService считывает ответы с kafka-и, записывает результат применения в БД;
* Если один из сервисов не смог обработать, принять новые конфигурационные данные, то на почту отправляется соответствующее уведомление

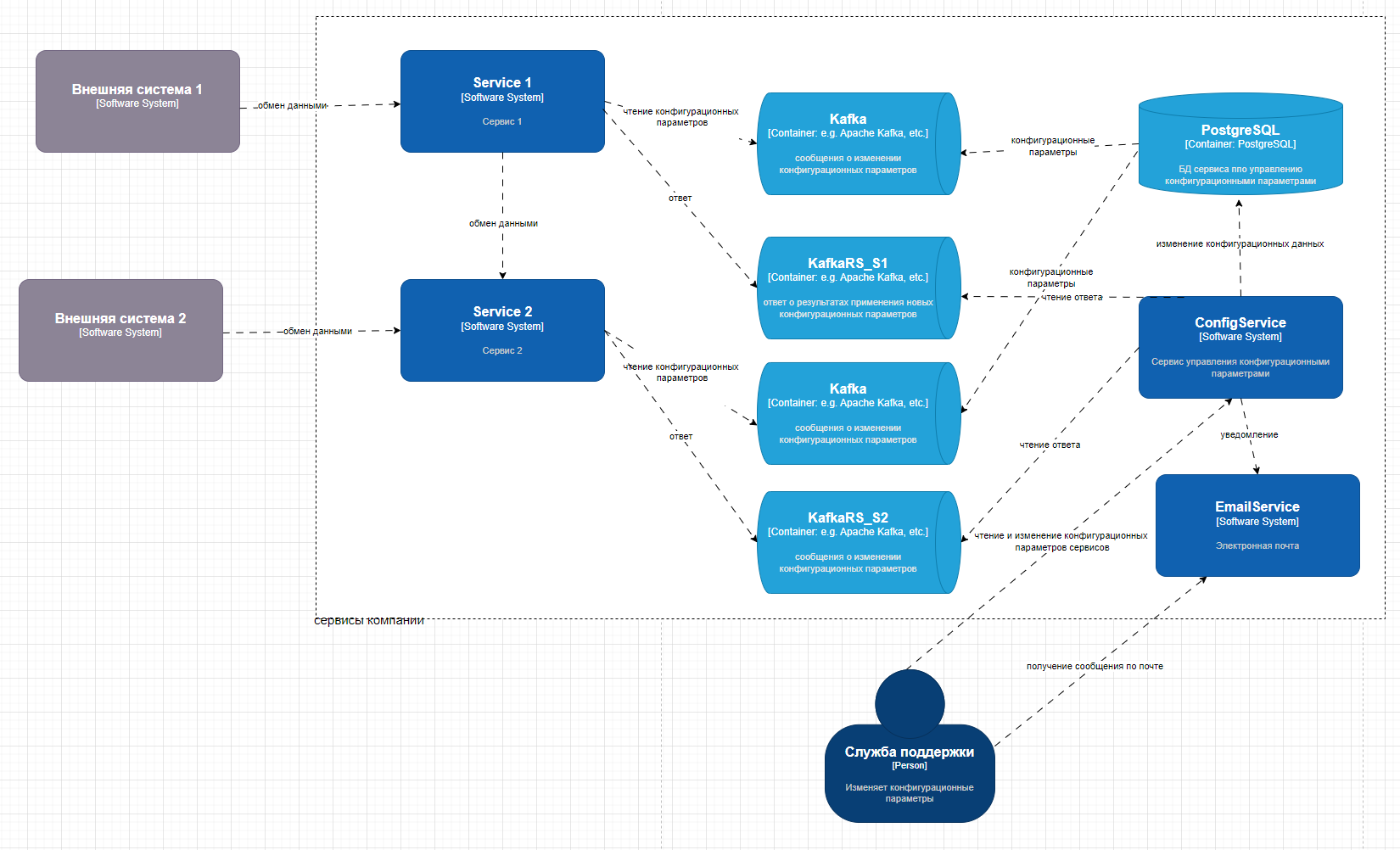


Рисунок - Контекстная схема решешния 3

#### Риски

Возможны следующие риски:

* Более сложная реализация;
* Может потребоваться больше ресурсов;

#### Возможности

Возможности:

* В случае падения брокера сообщений, сервиса, система восстановиться и продолжит работать;
* Очень маленькая вероятность потерять данные;
* Позволяет быть проинформированным о результатах применения новых параметров;
* Не создается бесполезный трафик для получения новых параметров конфигурации;

### Сравнительная оценка решений

Решение 1 в силу того, что данные работают с файлом конфигурации, несут ряд ограничений: сложно масштабируется, в случае порчи файла или падения сервиса данные могут быть безвозвратно потеряны.

Решение 2 является промежуточным вариантом между решениями 1 и 2.

Решение 3 можно развернуть в полноценной микросервисной архитектуре. Данное решение обеспечивает также возможность отслеживать ошибки в системе и быстро реагировать на них.

Учитывая, что решения 2 и 3 более сложные, то их сопровождение наиболее сложно и затратно: поиск и оплата труда специалистов, поддержка, закупка или оплаты инфраструктуры и т.п.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Характеристика | Решение 1 | Решение 2 | Решение 3 |
| 1 | Безопасность(контроль доступа) | - | + | + |
| 2 | Масштабируемость | - | + | + |
| 3 | Гибкость | + | - | + |
| 4 | Надежность | - | +- | + |
| 5 | Восстанавливаиемость | - | +- | + |
| 6 | Мониторинг | - | - | + |
| 7 | Отказоустойчивость | \_ | \_ | + |
| 8 | Удобство сопровождения | + | - | - |

### ADR решения 3

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | status | context | decision | responsibility party | decision date | comments |
| D001 | Принято | Технический | Микросервисы | Саша | 07.02.2024 |  |
| D002 | Принято | Бизнес | Важно быстро реагировать на ошибки сервиса, связанные с принятием новых обновлений | Костя | 07.02.2024 | Не должно быть простоя сервисов. Это отражается на доходах компании. |
| D003 | Отклонено | Технический | Использовать MongoDB в качестве хранилища | Саша | 08.02.2024 | Не целесообразно использовать NoSql БД, т.к. структура данных ясна и зависима от сервисов |
| D004 | Принято | Технический | Авторизация через отдельный микросервис с использованием ingress controller nginx | Саша | 08.02.2024 | Управление единой точкой входа |
| D005 | Принято | Бизнес | Развернуть всю инфраструктуру в облаке. | Костя | 10.02.2024 | Экономически выгодно – затраты на покупку услуг у провайдера оказалось дешевле, чем создавать и поддерживать эту инфраструктуру самим. Много уйдет время на поиск штата специалистов для создания и поддержки инфраструктуры k8s. Закупка серверов тоже требует существенных затрат. |
| D006 | Принято | Технический | Использовать в качестве хранилища PostgreSql | Саша | 10.02.2024 |  |
| D007 | На рассмотрении | Технический | Отправлять уведомления на телефон в случае ,если один из сервисов не принял изменения | Саша | 10.02.2024 | Никто не хочет получать спам на свой номер |
| D008 | Принято | Технический | Отправлять уведомления на почту в случае, если один из сервисов не принял изменения | Артем | 10.02.2024 | Важная часть обратного отклика системы |
| D009 | Предложено | Бизнес | Предусмотреть в бюджете ресурсы, необходимые для поддержания облачной инфраструктуры | Володя | 11.02.2024 | Средства нужны для оплаты услуг облачного провайдера, труда специалистов, которые бы поддерживали эту инфраструктуру в рабочем состоянии |
| D010 | Принято | Бизнес | В случае, если изменения не приняты, то система должна иметь возможность автоматически вернуться к предыдущим рабочим параметрам | Костя | 11.02.2024 | Сервисы не должны простаивать |
| D011 | Отконено | Технический | Включить систему мониторинга на базе ELK | Саша | 11.02.2024 | Последние версии стали платными |
| D012 | принято | Технический | Включить систему мониторинга на базе Opensearch | Саша | 11.02.2024 | Opensource, хорошая замена ELK |