### **Название задачи: Открытие депозитов онлайн (MVP)**

### **Автор: Орлов Илья**

### **Дата: 13.01.2025**

### **Функциональные требования**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действующие лица или системы** | **Use Case** | **Описание** |
| UC1 | Клиент, сайт, кол-центр, менеджер-кол-центра, АБС, менеджер фронт-офиса, менеджер бэк-офиса, СМС-шлюз. | Подача заявки через сайт | 1. Клиент заходит на сайт и видит список доступных депозитов с актуальными ставками. 2. Клиент выбирает депозит и подаёт заявку через форму на сайте, указывая свой номер телефона и Ф.И.О. 3. Заявка передаётся в систему кол-центра. 4. Менеджер кол-центра видит заявку в системе кол-центра и забирает её в работу. 5. Менеджер кол-центра звонит клиенту, оставившему заявку, для уточнения условий депозита и актуализирует данные заявки в системе кол-центра. 6. Система кол-центра передаёт заявку с актуальными условиями в АБС. 7. Клиент приходит в отделение банка для идентификации. Менеджер фронт-офиса вносит соответствующую информацию в АБС. 8. Менеджер бэк-офиса видит заявку клиента, прошедшего идентификацию, в АБС и берёт её в работу. 9. После подтверждения условий депозита по заявке менеджером бэк-офиса, АБС инициирует отправку СМС-уведомления клиенту через СМС-шлюз. |
| UC2 | Клиент, интернет-банк, СМС-шлюз, АБС, менеджер бэк-офиса. | Подача заявки через интернет-банк | 1. Клиент авторизуется в интернет-банке и видит список доступных депозитов с актуальными ставками, а также персонализированные ставки. 2. Клиент, указав счёт и сумму, подаёт заявку на открытие депозита в интернет-банке. 3. Интернет-банк отправляет клиенту через СМС-шлюз СМС-код для подтверждения операции. 4. После подтверждения операции заявка передаётся в АБС. 5. Менеджер бэк-офиса видит заявку клиента в АБС и берёт её в работу. 6. После подтверждения условий депозита по заявке менеджером бэк-офиса, АБС инициирует отправку СМС-уведомления клиенту через СМС-шлюз. |
| UC3 | Менеджер бэк-офиса, АБС, сайт, интернет-банк. | Управление ставками по депозитам | 1. Менеджер бэк-офиса ежедневно актуализирует ставки по депозитам в АБС. 2. АБС передаёт актуальные значения ставок в интернет-банк и на сайт. |

### **Нефункциональные требования**

|  |  |
| --- | --- |
| **№** | **Требование** |
| **R** | **Надёжность (Reliability)** |
| R1 | Все сервисы должны работать 24/7. |
| R2 | Доступность сервисов должна составлять 99,9% |
| **P** | **Производительность (Performance)** |
| P1 | Отклик по всем операциям должен занимать миллисекунды. |
| P2 | Необходимо предусмотреть использование горизонтального масштабирования и балансировку нагрузки для интернет-банка и сайта. |
| **S** | **Поддерживаемость (Supportability)** |
| S1 | Необходимо предусмотреть возможность перевода интернет-банка на микросервисную архитектуру в рамках задачи открытия депозитов. |
| **+R** | **+ Ограничения (Restricitions)** |
| +R1 | Необходимо обеспечить шифрование клиентских данных при передаче и хранении. |
| +R2 | Необходимо использовать технологии, уже имеющиеся в банке, либо совместимые с имеющимися платформами. |
| +R3 | Необходимо использовать технологии, по которым в банке уже имеется экспертиза (СУБД: Oracle, MS SQL; языки программирования и фреймворки: Delphi, PHP, React.js, Java, .Net, SQL). |
| +R4 | В качестве очереди сообщений необходимо использовать Kafka, но текущая версия интернет-банка с ней несовместима. |
| +R5 | Следует избежать прямой работы интернет-банка с API АБС в новом процессе. |
| +R6 | Необходимо предусмотреть возможность только вертикально масштабирования АБС. |

### **Решение**

**Диаграмма C4-Context**

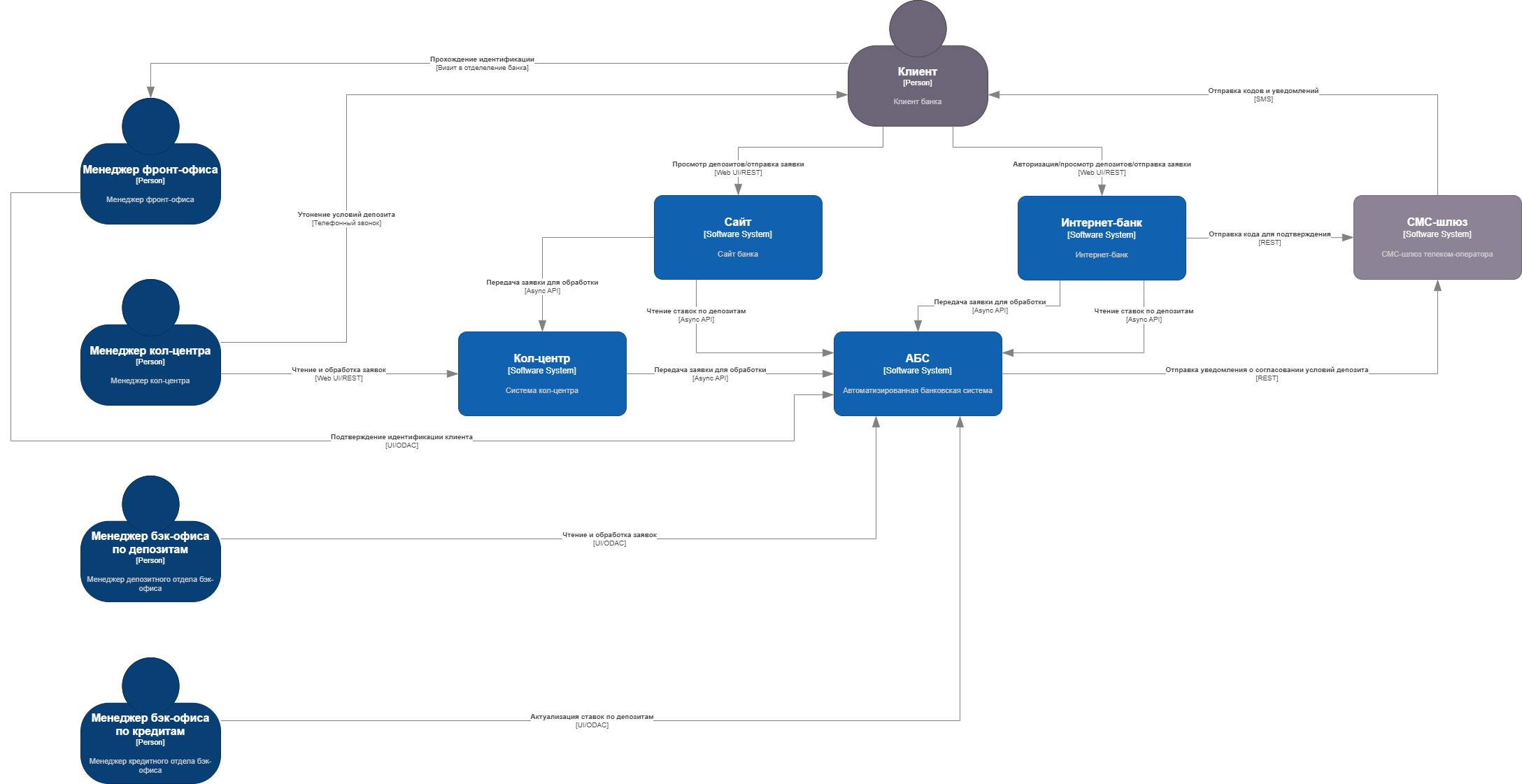


Рис.1 Диаграмма C4-Context

Диаграмма (см. Рис.1) демонстрирует выполнение функциональных требований через реализацию Use Cases с использованием имеющихся IT-систем.

**Диаграмма C4-Container**

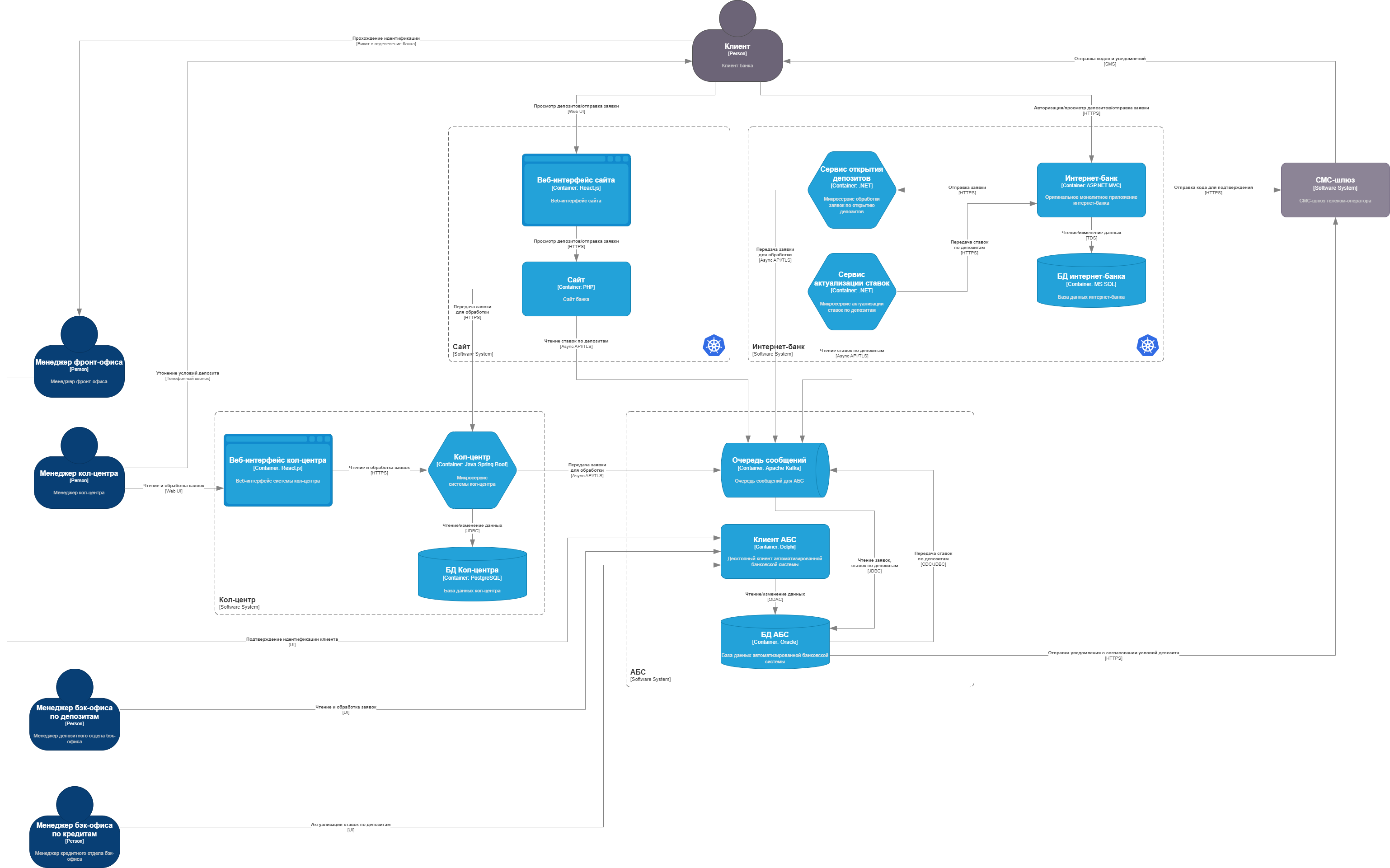


Рис.2 Диаграмма C4-Container

Диаграмма (см. Рис.2) иллюстрирует решения, принятые для удовлетворения нефункциональных требований.

**R1, R2, P1, P2.** Для обеспечения отказоустойчивости и высокой доступности сервисов интернет-банк и сайт развернуты под управлением Kubernetes с возможностью динамического горизонтального масштабирования и георезервирования с использованием резервного ЦОДа. Очередь сообщений Kafka также развёрнута в отказоустойчивом кластере из 3-х узлов. Для ускорения загрузки справочных данных в монолите интернет-банка стоит предусмотреть кэширование на уровне приложения.

**S1.** Для обработки заявок на открытие депозитов, а также для актуализации данных по ставкам в систему интернет-банка добавлены соответствующие микросервисы. В микросервисах также реализована поддержка взаимодействия с Kafka, что решает проблему несовместимости с Kafka оригинального монолитного приложения интернет-банка.

**+R1.** Для защиты клиентских данных предусмотрено их шифрование при передаче между IT-системами (HTTPS, TLS) и хранении.

**+R2, +R3.** Для реализации новых микросервисов использованы технологии, уже имеющиеся в банке, по которым у команды банка уже имеется экспертиза: сервис открытия депозитов (.NET), сервис актуализации ставок (.NET). Выбор в качестве очереди сообщений Kafka также согласуется с имеющейся у команды экспертизой в Java.

**+R4, +R5, +R6.** Для снижения нагрузки на АБС и исключения прямой работы интернет-банка с API АБС предусмотрено использование очереди сообщений. Взаимодействие АБС со внешними системами (интернет-банк, сайт, кол-центр) по передаче заявок на открытие депозитов и актуализации ставок по депозитам сделано асинхронным с использованием Kafka.

### **Альтернативы**

При проектировании архитектуры принят ряд решений, не являющихся прямым следствием описанных ранее требований: использование Kubernetes для интернет-банка и сайта, добавление функционала для подтверждения операций с помощью СМС-кода непосредственно в монолитном приложении интернет-банка, вынесение функционала для актуализации ставок в интернет-банке в отдельный микросервис, использование АБС в качестве мастер-системы для хранения актуальных ставок по депозитам. Далее для обоснования принятых решений будут описаны их ключевые преимущества над альтернативами.

1. **Масштабирование интернет-банка и сайта без использования Kubernetes.**

Описание: Для серверов интернет-банка и сайта выделяются дополнительные инстансы в режиме active-active или active-standby для горизонтального масштабирования. Также предполагается добавление балансировщика нагрузки.

Преимущества:

* Не требуется экспертизы в Kubernetes.
* Проще реализовать.

Недостатки:

* Не подходит для динамически изменяющейся нагрузки.
* Требуется экспертиза в настройке балансировщика нагрузки.

Причина отказа: DevOps-экспертиза всё равно требуется, а относительная простота реализации компенсируется сложностью планирования требуемого количества и мощности инстансов. Также данная альтернатива неэффективна для предполагаемого характера нагрузки на системы.

1. **Добавление микросервиса для подтверждения операций с помощью СМС-кода.**

Описание: Функционал для подтверждения операций с помощью СМС-кода реализуется в отдельном микросервисе, а не в монолитном приложении интернет-банка.

Преимущества:

* Соответствует общему плану по переводу интернет-банка на микросервисную архитектуру.

Недостатки:

* Лишние сетевые взаимодействия из-за зависимости от основного функционала для работы с СМС, находящемся в ядре системы.
* Необходима отдельная база данных для хранения кода подтверждения в случае масштабирования сервиса.

Причина отказа: Вынесение функционала, находящегося в сильной связности с ядром системы, в отдельный микросервис только замедлит работу системы. А более глубокое разделение монолита на этапе MVP не предусмотрено.

1. **Реализация функционала для актуализации ставок в монолите интернет-банка.**

Описание: Функционал для чтения актуальных ставок по депозитам из очереди сообщений реализуется в монолитном приложении интернет-банка, а не в отдельном микросервисе.

Преимущества:

* Не требуется поднимать отдельный сервис.

Недостатки:

* Не соответствует общему плану по переводу интернет-банка на микросервисную архитектуру.
* Платформа монолита интернет-банка не совместима с Kafka, используемой в качестве очереди сообщений.

Причина отказа: Из-за несовместимости платформы монолита с Kafka решение в любом случает будет слабо связанным с кодом монолита и оставлять его в монолите не имеет смысла.

1. **Добавление сервиса для хранения актуальных ставок по депозитам.**

Описание: Менеджеры бэк-офиса вносят актуальные данные по ставкам в отдельный сервис, относящийся к системе интернет-банка, сайта, откуда эти данные передаются в остальные системы.

Преимущества:

* Снижается нагрузка на АБС.

Недостатки:

* Создание дополнительного сервиса даже на основе имеющихся систем требует реализации отдельных интерфейсов и контроля доступа для различных групп пользователей.

Причина отказа: АБС уже имеет необходимые интерфейсы и механизмы разделения доступа. Поэтому использовать её в качестве мастер-системы для хранения актуальных ставок удобней и эффективней. Тем более, актуальные ставки рассчитываются менеджером бэк-офиса с применением данных как раз из АБС. Нагрузка на АБС, обусловленная ежедневным изменением ставок, минимальна, чтение же данных по ставкам остальными системами можно выполнять не из АБС, а из соответствующего топика Kafka.

**Недостатки, ограничения, риски**

Недостатки:

* Необходима экспертиза в Kubernetes.
* Необходимо развернуть очередь сообщений, предусмотрев её высокую доступность и отказоустойчивость.
* Необходимо георезервирование для интернет-банка и сайта.

Ограничения:

* Основной функционал интернет-банка остаётся в монолитном приложении.
* АБС остаётся узким местом системы, не предполагающим горизонтального масштабирования.
* Для СУБД АБС не предусмотрено георезервирование.
* Обмен данными между менеджерами бэк-офиса отделов депозитов и кредитов через АБС не допустим по требованиям безопасности.

Риски:

* АБС не выдерживает нагрузки (замедление работы, отказ в обслуживании).

Митигация: заблаговременно подготовить ресурсы для вертикального масштабирования, обеспечить мониторинг состояния СУБД Oracle (CPU, RAM, дисковое пространство) и алертинг, запланировать миграцию на новую масштабируемую СУБД.

* Интернет-банк не выдерживает нагрузки (замедление работы, отказ в обслуживании).

Митигация: обеспечить мониторинг состояния кластера интернет-банка и алертинг, подготовить возможности перераспределения нагрузки на резервный ЦОД в Kubernetes, предусмотреть альтернативные способы подачи заявки на открытие депозита (через сайт, кол-центр и визит в отделение).

* Сайт не выдерживает нагрузки (замедление работы, отказ в обслуживании).

Митигация: обеспечить мониторинг состояния кластера сайта и алертинг, предусмотреть альтернативные способы подачи заявки на открытие депозита (через кол-центр и визит в отделение).

* Недоступность СМС-шлюза при отправке кодов подтверждения и уведомлений.

Митигация: обеспечить мониторинг доступности СМС-шлюза и алертинг, запланировать реализацию альтернативных способов отправки кодов подтверждения и уведомлений (резервный СМС-шлюз, e-mail, push-уведомления в интернет-банке), предусмотреть возможность автоматических повторных попыток отправки уведомлений.

* Утечка данных клиентов с сайта, интернет-банка

Митигация: обеспечить шифрование данных при передаче, исключить хранение данных клиентов на серверах сайта, использовать шифрование и псевдонимизацию для данных клиентов в интернет-банке, предусмотреть ограничение и контроль доступа к клиентским данным в контуре интернет-банка, проводить регулярный аудит безопасности.