### **Название задачи: Концептуальная архитектура открытия депозитов для MVP**

### **Автор: Челнаков Алексей**

### **Дата: 22.12.2024**

## **Контекст**

Проект направлен на цифровизацию процесса открытия депозитов для клиентов банка. На этапе MVP решено реализовать минимальный набор функциональности, включающий:

* Подачу заявки на депозит через сайт и интернет-банк.
* Обработку заявок менеджерами кол-центра и бэк-офиса.
* Подтверждение условий депозита в АБС.

### **Функциональные требования**

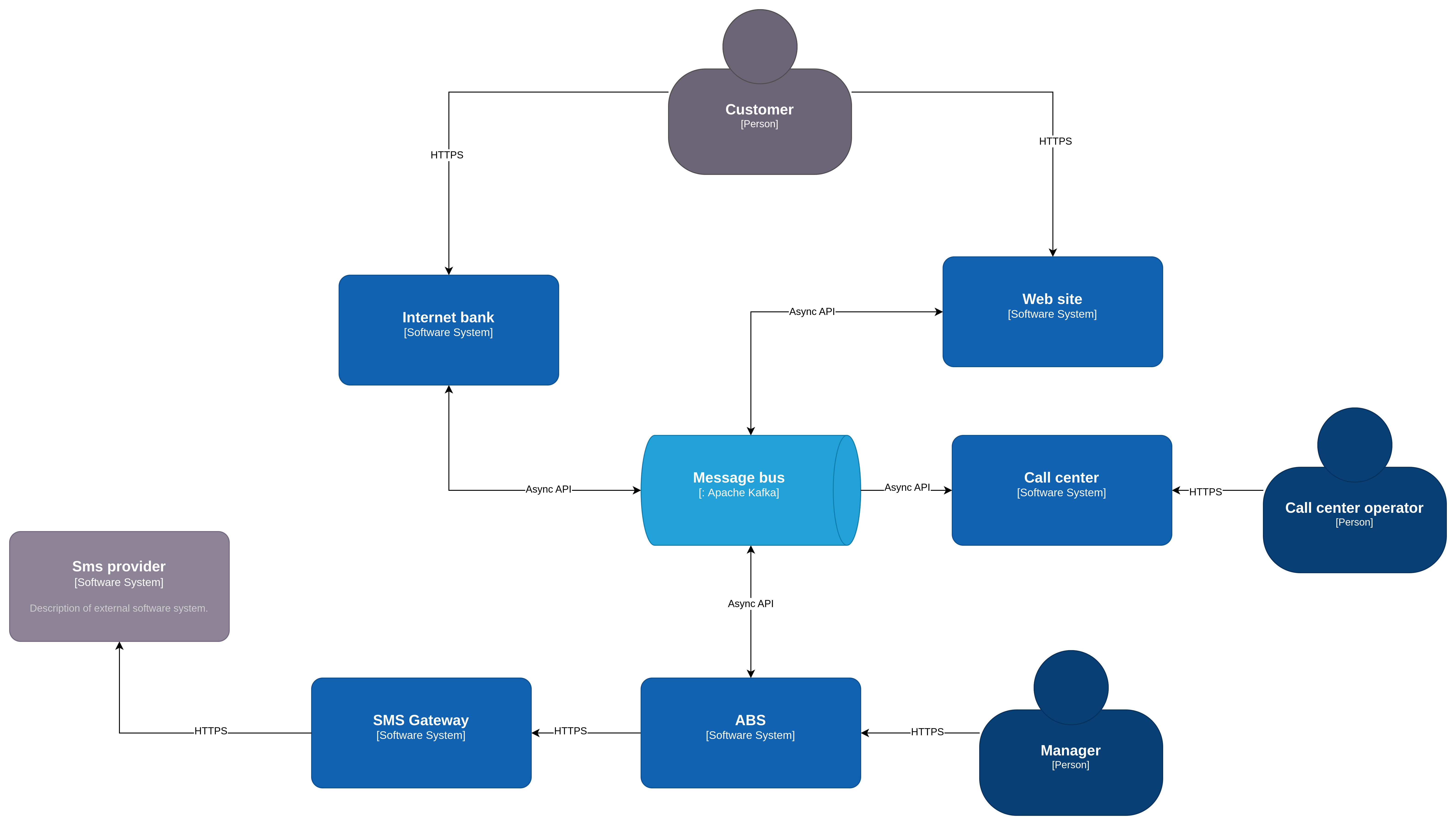
|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Действующие лица или системы** | **Use Case** | **Описание** |
| 1 | Интернет-сайт, Колл-центр, Менеджер кол-центра | Подача заявки через сайт | 1. Клиент выбирает депозит и оставляет заявку. 2. Заявка передается в кол-центр. 3. Менеджер кол-центра связывается с клиентом для уточнения условий. |
| 2 | Интернет-банк, АБС, Смс-шлюз | Подача заявки через интернет-банк | 1. Клиент авторизуется и видит персонализированные ставки. 2. Выбирает депозит, подтверждает сумму и счет. 3. Система отправляет заявку в АБС 4. После обработки клиент получает СМС. |
| 3 | АБС,  Kafka, Менеджер Бэк-офиса, Интернет-банк, Сайт | Управление ставками | 1. Сотрудник бэк-офиса редактирует ставки в хранилище. 2. Обновленные ставки становятся доступными на сайте и в интернет-банке. |

### **Нефункциональные требования**

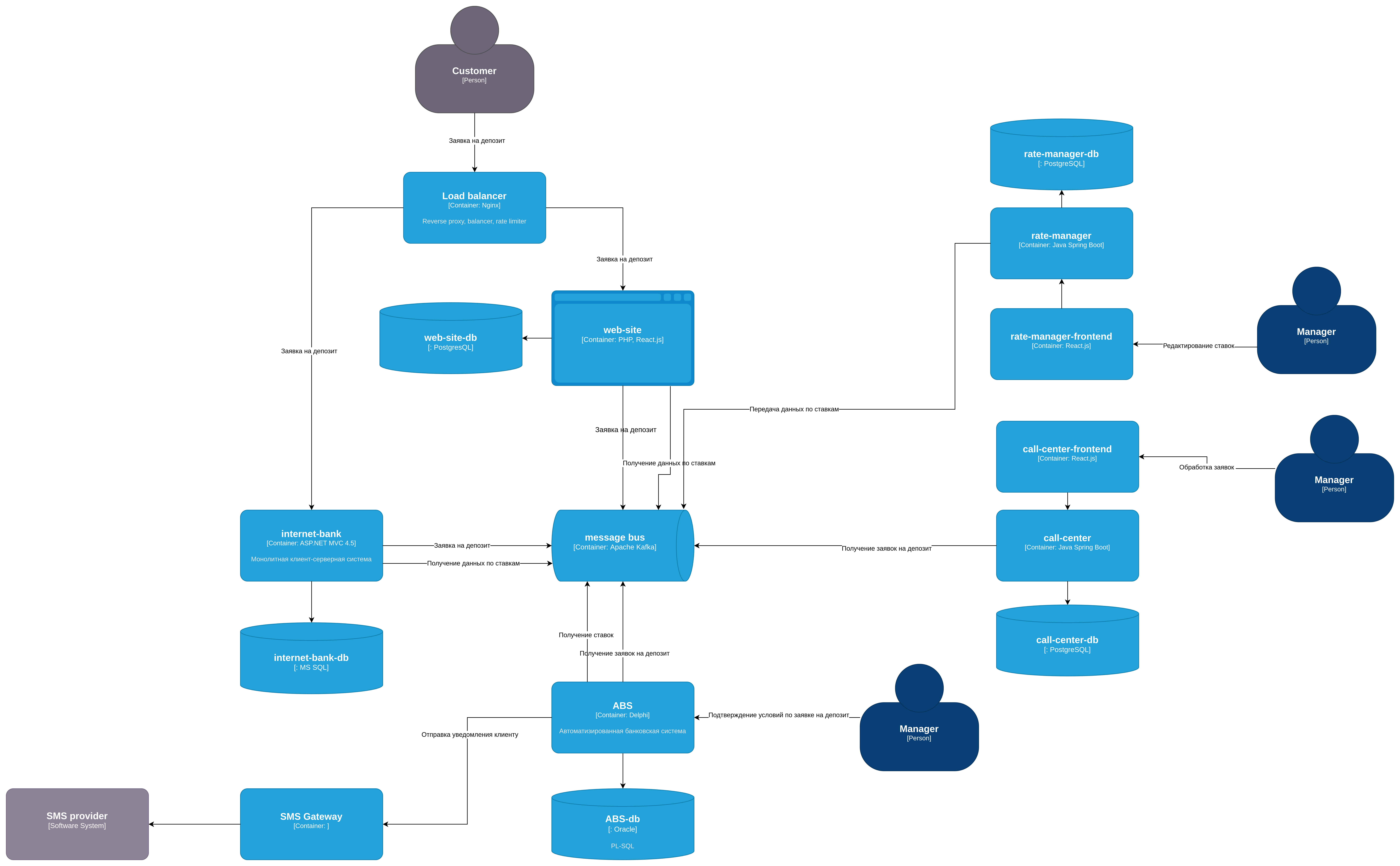
|  |  |
| --- | --- |
| **№** | **Требование** |
| 1 | **Функциональность**: Защита данных, персонализированные ставки, интеграция с существующими системами. |
| 2 | **Производительность**: Минимальные задержки отклика. |
| 3 | **Надежность**: Доступность 99.9%, возможность переключения на резервный ЦОД. |
| 4 | **Технологии**: Использование MS SQL, Oracle, существующих технологий банка. |

### **Решение**

**C4 context**



**C4 Container**



### **Альтернативы**

Поиск и внедрение готовых решений для управления ставками. Позволит ускорить этап с разработкой этой части системы, но из минусов завязка на стороннее решение, не всегда оправдано в долгосрочной перспективе. На этом этапе в целом это не требуется и можно в том числе оставить управление ставками в АБС. Но стоит иметь ввиду что АБС работает на устаревших тенологиях (Delphi) и испытывает сложности с горизонтальным масштабированием.

Альтернативы Event-Driven архитектуре, предложенной в данном ADR может быть Service-Oriented Architecture (SOA):

* Система состоит из сервисов, которые взаимодействуют через протоколы (например, SOAP, REST).
* Каждый сервис отвечает за конкретную бизнес-функцию.

#### Преимущества:

* Высокая реиспользуемость сервисов.
* Модульность упрощает внесение изменений.
* Подходит для больших корпоративных систем.

#### Недостатки:

* Высокая сложность управления взаимодействием между сервисами.
* Зависимость от стандартов и контрактов.

**Недостатки и ограничения решения в ADR**

* Требуется переработка существующих систем (например, Интернет Банк, АБС).
* Брокер сообщений становится критически важной частью системы.
* Возможна потеря событий при сбое брокера сообщений или неправильной настройке системы.
* Обеспечение согласованности данных при асинхронной обработке становится сложной задачей.
* Логика распределена между несколькими компонентами, что затрудняет диагностику и исправление ошибок.

**Риски**

* **Технические:** Возможные задержки в интеграции новых компонентов с существующими системами.
* **Организационные**: Недостаточная экспертиза сотрудников в новых технологиях.
* **Производственные**: Проблемы с производительностью при росте числа пользователей.
* **Безопасность**: Необходимость обеспечения шифрования данных и защиты информации клиентов