

Исходные данные

Есть банковская система, в которой реализованы следующие сервисы:

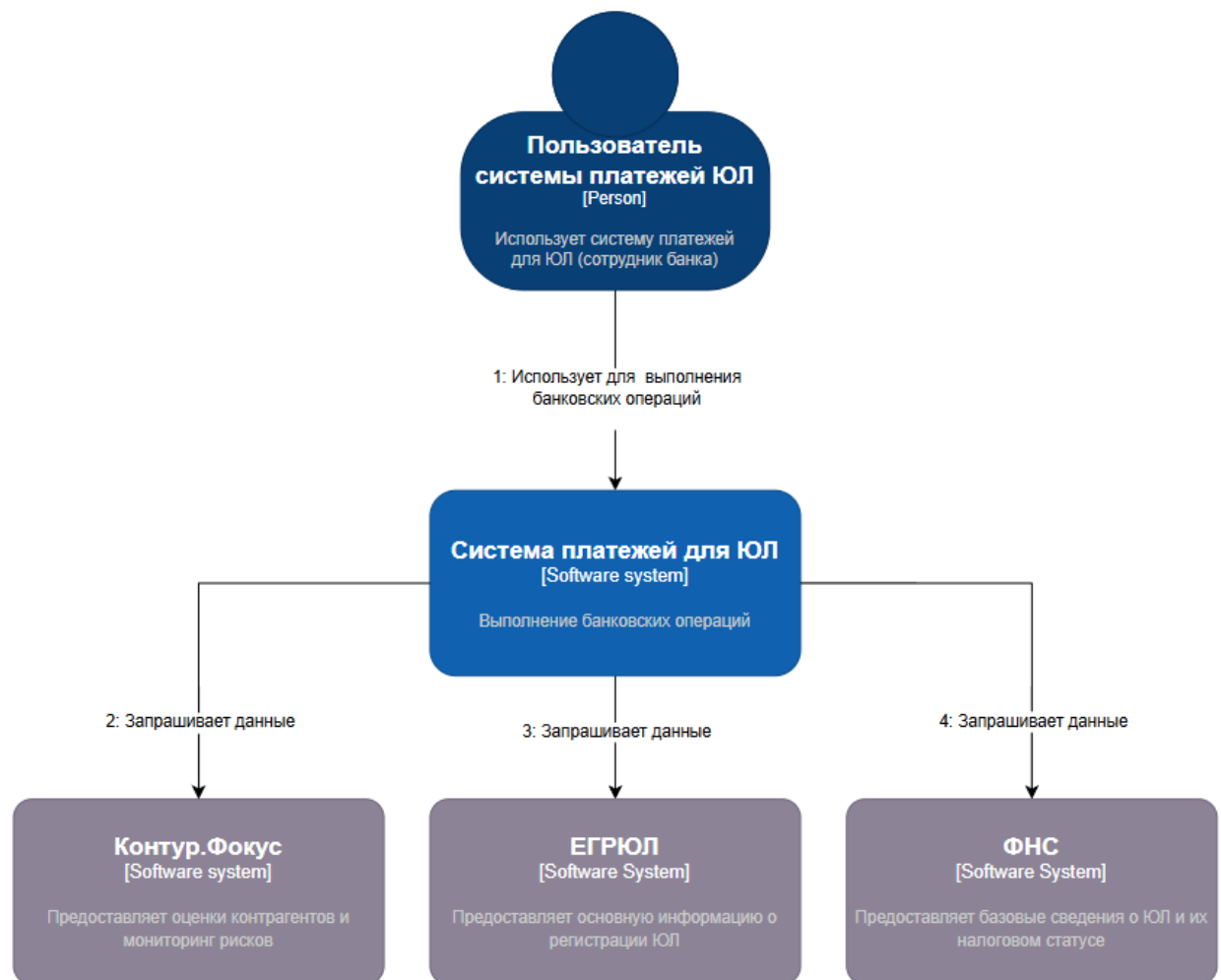
	Элемент	Функции
1	Фронт	Веб-приложение Содержит форму создания платежа.
2	Микросервис авторизации и аутентификации	Проверяет полномочия пользователя на совершение операции. Выдает токен для выполнения запросов к системе.
3	Микросервис «Платежи ЮЛ»	Сервис-оркестратор платежей: <ul style="list-style-type: none">• принимает запросы на создание платежа• проверяет платеж на возможность проведения, получая данные из микросервиса «Проверки ЮЛ»• отправляет платежи на исполнение в сервис «Процессинг платежей»• получает статусы по платежам из сервиса «Процессинг платежей»• отправляет данные по всем платежам в «Озеро данных»
4	Микросервис «Проверки ЮЛ»	Сервис-агрегатор данных по ЮЛ, содержит необходимую для проведения платежа информацию: <ul style="list-style-type: none">• наличие блокировок на транзакции клиента• наличие блокировок на счета клиента• лимиты по операциям <p>Без проверки и получения информации из этого сервиса платеж не может быть совершен!</p> <p>Все необходимые данные микросервис получает из внешних систем, таких как Контур.Фокус, ФНС, ЕГРЮЛ и др.</p>
5	Микросервис «Процессинг платежей»	Сервис-исполнитель платежей: <ul style="list-style-type: none">• получает всю необходимую информацию по платежу от сервиса «Платежи ЮЛ»• переводит деньги непосредственно со счета клиента на счет получателя
6	Микросервис «Озеро данных»	Сервис-агрегатор данных по всем платежам всех систем в банке. Получает данные для формирования отчетов и статистики.

Моделирование диаграммы контекста

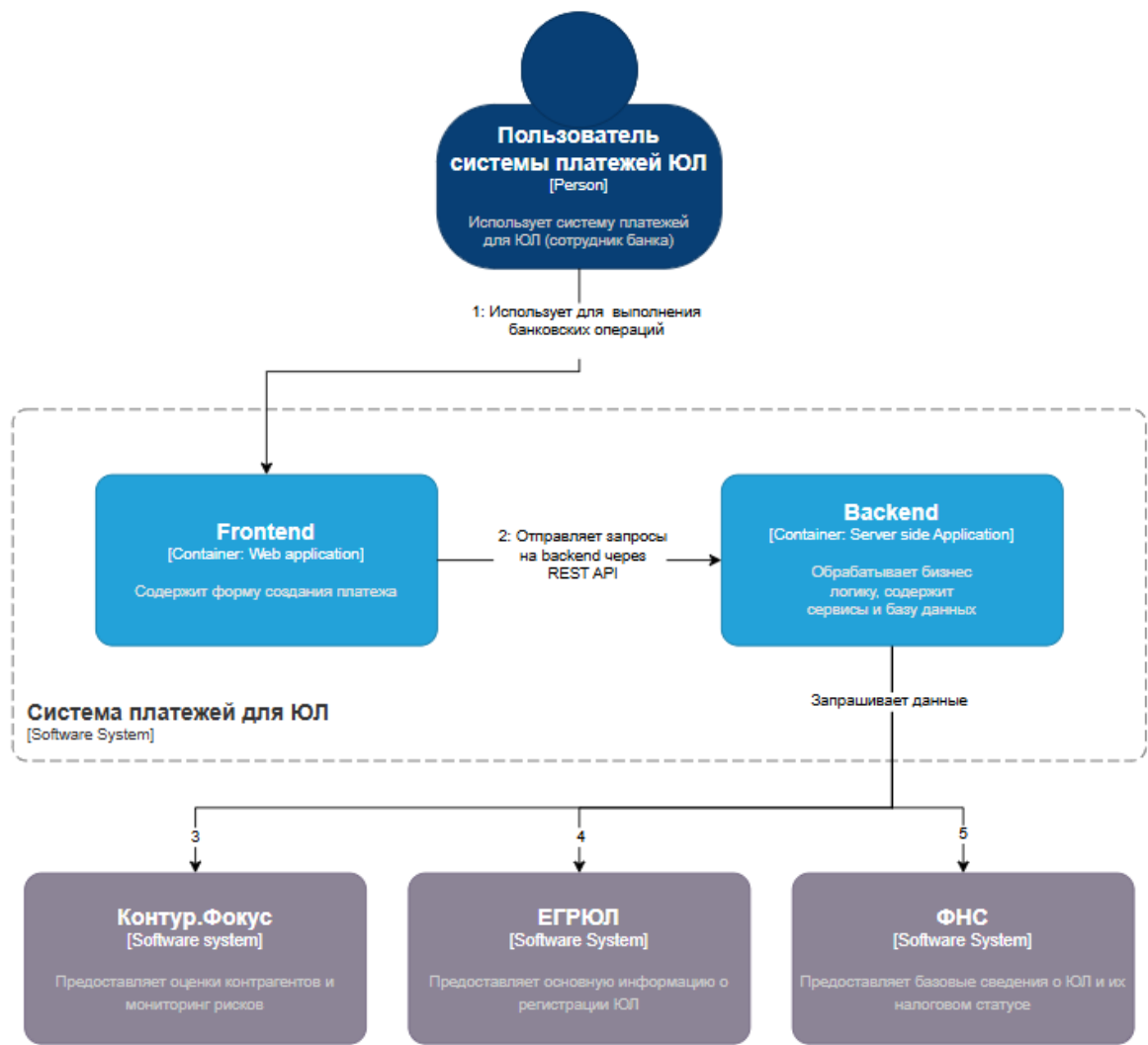
Документируемая система – система платежей для юридических лиц (ЮЛ).

Пользователь системы – сотрудник банка, который проводит платежные операции клиентов (юридических лиц) через банковские интерфейсы.

Контур.Фокус, ФНС, ЕГРЮЛ – внешние системы, предоставляющие необходимые данные для выполнения банковских операций.

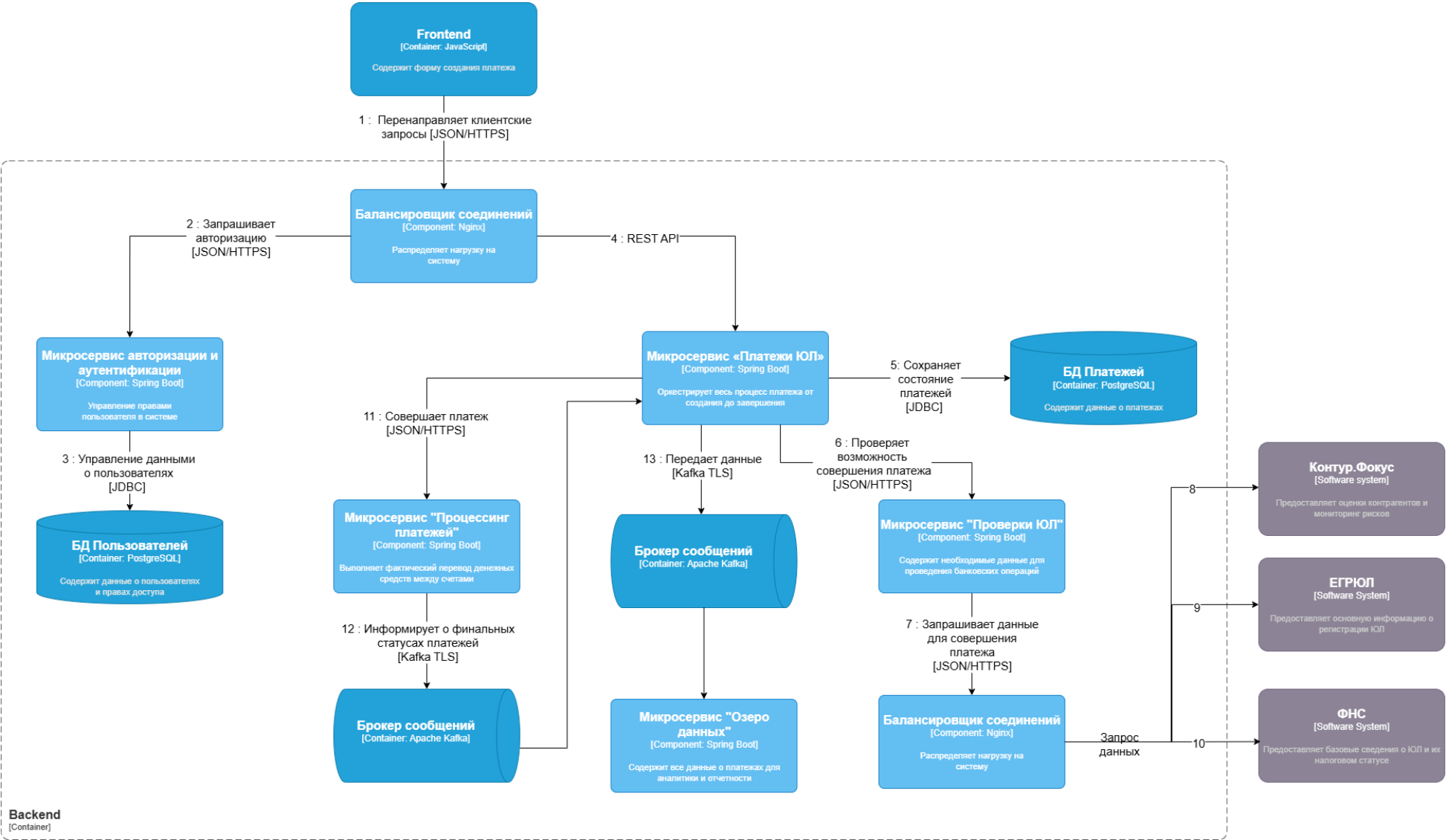


Моделирование диаграммы контейнеров



№ связи	Тип (Синхронный/асинхронный) Технология	Обоснование
1	Синхронный HTTPS, JavaScript	Пользователь ожидает немедленного отклика интерфейса на свои действия (заполнение формы).
2	Синхронный REST API, JSON, HTTPS	Frontend отправляет запросы на Backend для выполнения операций. JSON — легкий и удобный для JavaScript формат. HTTPS обеспечивает безопасность финансовых данных.
3	Синхронный REST API, SOAP, API-ключами/ЭЦП, HTTPS, JSON	Платеж не может быть передан на исполнение, пока не пройдены все проверки. (Нужно исключать ситуацию, когда платеж отправлен в процессинг, но проверки еще не завершены). Коммерческие сервисы (Контур.Фокус) используют REST API с ключами, государственные системы (ФНС, ЕГРЮЛ) — SOAP с электронной подписью.
4		
5		

Моделирование диаграммы компонентов



№ связи	Тип (Синхронный/асинхронный) Технология	Обоснование
1	Синхронный REST API, JSON, HTTPS	Пользователь ожидает немедленного отклика интерфейса на свои действия. JSON — легкий и удобный для JavaScript формат. HTTPS обеспечивает безопасность финансовых данных.
2	Синхронный REST API, JSON, HTTPS	Проверка авторизации должна завершиться до обработки основного запроса.
3	Синхронный JDBC, PostgreSQL	PostgreSQL выбран как надежная реляционная СУБД. JDBC обеспечивает единый интерфейс для работы с СУБД.
4	Синхронный REST API	Пользователь ожидает подтверждения создания платежа в реальном времени.
5	Синхронный JDBC, PostgreSQL	PostgreSQL выбран как надежная реляционная СУБД. JDBC обеспечивает единый интерфейс для работы с СУБД.
6	Синхронный REST API, JSON, HTTPS	Решение о возможности проведения платежа должно быть принято до его передачи на исполнение.
7	Синхронный REST API, SOAP, API-ключами/ЭЦП, HTTPS, JSON	Платеж не может быть передан на исполнение, пока не пройдены все проверки. (Нужно исключать ситуацию, когда платеж отправлен в процессинг, но проверки еще не завершены). Коммерческие сервисы (Контур.Фокус) используют REST API с ключами, государственные системы (ФНС, ЕГРЮЛ) — SOAP с электронной подписью.
8		
9		
10		
11	Синхронный REST API, JSON, HTTPS	Немедленное подтверждение, что платеж принят к исполнению.
12	Асинхронный Kafka TLS	Исполнение платежа в банковских системах может занимать от нескольких минут до нескольких дней. Результат сообщается автоматически, как только он становится известен.
13	Асинхронный Kafka TLS	Отправка данных в Озеро Данных не должна блокировать основной процесс проведения платежей.