

Технические требования к банковской API-интеграции

MVP сервиса «Безопасная сделка» / Trust System

Адресат: АО «Банк ЦентрКредит», Open Banking / Open API

Заявитель: ТОО «МАСШТАБ ARNA», БИН 260740041659

Рынок: Республика Казахстан · Валюта MVP: KZT

Статус документа: запрос на техническую и продуктовую оценку

1. Цель интеграции

Компания запускает сервис безопасной сделки между покупателем и исполнителем при заказе корпусной мебели и ремонтных работ.

Участники сделки:

- **Покупатель** — физическое лицо;
- **Исполнитель** — индивидуальный предприниматель или ТОО;
- **Платформа** — информационная система, фиксирующая договор, статусы сделки и команды на проведение финансовых операций;
- **Банк** — банковский и платёжный партнёр.

Основной принцип. Покупатель оплачивает заказ → средства находятся под контролем банковского механизма → исполнитель выполняет заказ → покупатель подтверждает приёмку → средства перечисляются исполнителю.

До наступления предусмотренного условия исполнитель не должен иметь возможности получить средства. Платформа не хранит денежные средства клиентов на собственном расчётном счёте и не выступает хранилищем денежных средств.

2. Требуемая банковская модель

Для MVP требуется банковский механизм, позволяющий:

1. связать денежные средства с конкретной сделкой;
2. подтвердить платформе факт поступления и обеспечения средств;
3. исключить возможность получения средств исполнителем до наступления условий сделки;
4. перечислить средства исполнителю только после соответствующего основания и команды платформы;
5. вернуть средства покупателю при отмене сделки или соответствующем решении по спору;
6. сохранять средства в защищённом состоянии на период изготовления заказа, который может составлять несколько недель или месяцев.

Юридическая реализация может быть построена через счёт эскроу, специальный или номинальный счёт, либо иной банковский механизм условного перечисления денежных средств. **Выбор юридической и банковской модели остаётся за Банком.** Для платформы критически важна возможность управлять жизненным циклом такой сделки через API.

По данным портала developer.bcc.kz, в каталоге продуктов Банка присутствует **Escrow API** (версии 1.0.1 и 2.0.0). Просим подтвердить его применимость к описанному сценарию либо указать альтернативный продукт.

3. Scope MVP

Параметр	Значение в первой версии
Покупатель	физическое лицо
Исполнитель	проверенный ИП или ТОО
Валюта	KZT
Сделка	один заказ — одна защищённая сумма
Оплата	100% защищаемой суммы одним платежом
Выплата	полная сумма исполнителю после подтверждения приёмки
Возврат	полная сумма покупателю при соответствующем основании
Спор	рассматривается вручную сотрудником платформы

В MVP не требуются частичные выплаты, рассрочка, кредитование, автоматический арбитраж и многоэтапные платежи.

4. Основной сценарий сделки

4.1. Создание сделки

На платформе создаётся сделка с уникальным идентификатором `external_deal_id`. Фиксируются: покупатель, исполнитель, сумма, валюта, предмет заказа, сроки, договор, условия выплаты и возврата. Обе стороны подтверждают договор.

4.2. Создание банковской защищённой сделки

Платформа передаёт в Банк данные, необходимые для создания банковской сущности, связанной с `external_deal_id`. Минимально требуемые данные:

```
{
  "external_deal_id": "DL-100025",
  "amount": 850000,
  "currency": "KZT",
  "buyer_reference": "USR-1001",
  "seller_reference": "USR-5507",
  "contract_reference": "CNT-100025",
  "contract_hash": "...",
  "expires_at": "..."
}
```

Банк возвращает как минимум:

```
{
  "bank_deal_id": "...",
  "external_deal_id": "DL-100025",
  "amount": 850000,
  "currency": "KZT",
  "status": "AWAITING_FUNDS"
}
```

Названия полей и API-методов являются логическими требованиями и могут быть сопоставлены с существующим контрактом Банка.

4.3. Оплата

После создания банковской сделки платформа должна получить способ внесения денежных средств покупателем: платёжная ссылка, банковский перевод, QR, карта либо иной способ, поддерживаемый Банком. После оплаты источником финансового статуса должен являться ответ API или серверное уведомление банка, а не возврат пользователя на Success Page.

4.4. Подтверждение обеспечения средствами

После фактического поступления средств платформа должна получить банковское подтверждение `FUNDS_SECURED`. После этого исполнитель может начинать выполнение заказа. До получения данного подтверждения платформа не считает сделку оплаченной.

4.5. Выполнение и приёмка

После завершения работ покупатель либо подтверждает приёмку, либо открывает спор. При открытом споре выплата не производится до принятия решения.

4.6. Выплата исполнителю

После подтверждения приёмки платформа отправляет в банк команду на перечисление средств исполнителю — условная операция `ReleaseFunds` :

```
{
  "bank_deal_id": "...",
  "amount": 850000,
  "request_id": "...",
  "reason": "BUYER_ACCEPTED"
}
```

Банк возвращает идентификатор финансовой операции. Факт успешного приёма API-запроса не должен автоматически означать завершение сделки: платформа переводит сделку в состояние «Завершена» только после банковского подтверждения фактического перечисления `FUNDS_RELEASED`.

4.7. Возврат покупателю

При отмене сделки либо соответствующем решении по спору платформа должна иметь возможность инициировать возврат защищённых средств покупателю — условная операция `RefundFunds` :

```
{
  "bank_deal_id": "...",
  "amount": 850000,
  "request_id": "...",
  "reason": "DEAL_CANCELLED"
}
```

После фактического возврата банк передаёт подтверждение `FUNDS_REFUNDED`.

5. Минимально необходимые банковские API

№	Возможность	Назначение
1	Create Protected Deal	Создание банковской сущности безопасной сделки
2	Create / Get Funding Method	Получение способа внесения средств покупателем
3	Get Deal Status	Получение актуального финансового состояния сделки
4	Release Funds	Перечисление защищённых средств исполнителю
5	Refund Funds	Возврат защищённых средств покупателю
6	Get Operation Status	Проверка результата release / refund
7	Webhooks / Callbacks	Асинхронные уведомления об изменениях финансового состояния
8	Reconciliation	Сверка состояний платформы и банка

Если исполнителю необходимо предварительно пройти банковский onboarding или проверку реквизитов, дополнительно требуется механизм Verify / Create Beneficiary либо существующий аналог Банка. Конкретные REST URL, форматы запросов и названия методов определяются на основании существующей архитектуры Банка.

6. Минимальные финансовые статусы

AWAITING_FUNDS

Ожидание оплаты

FUNDS_SECURED

Средства поступили и защищены

RELEASE_PROCESSING

Выплата исполнителю выполняется

FUNDS_RELEASED

Средства фактически перечислены исполнителю

REFUND_PROCESSING

Выполняется возврат

FUNDS_REFUNDED

Средства фактически возвращены покупателю

FAILED

Финансовая операция завершилась ошибкой

CANCELLED / EXPIRED

Банковская сделка отменена либо истёк срок её действия

Банк может использовать собственную модель статусов при условии возможности однозначного сопоставления с указанными состояниями.

7. Webhooks и callbacks

Для MVP необходимы серверные уведомления об изменениях финансового состояния. Минимально требуются события: средства успешно обеспечены; платёж неуспешен; начата выплата; выплата завершена; начат возврат; возврат завершён; банковская сделка отменена или истекла.

Каждое событие должно содержать как минимум:

```
{
  "event_id": "...",
  "event_type": "FUNDS_SECURED",
  "bank_deal_id": "...",
  "external_deal_id": "DL-100025",
  "amount": 850000,
  "currency": "KZT",
  "created_at": "..."
}
```

Для webhook и callback необходимо предусмотреть: уникальный `event_id` ; защиту и подпись сообщения; timestamp; возможность повторной доставки; корректную обработку повторного события; возможность восстановить состояние через API, если callback не был получен.

8. Идемпотентность

Все команды, способные изменить финансовое состояние, должны защищаться от повторного исполнения — в первую очередь создание банковской сделки, `release` и `refund`. Каждая команда передаётся с уникальным `request_id` / `Idempotency-Key` либо аналогичным механизмом банка. Повторная отправка одной и той же команды не должна приводить к повторному списанию, выплате или возврату.

9. Источник финансовой истины

Платформа отвечает за бизнес-состояние заказа, Банк — за финансовое состояние денежных средств. Например:

Платформа: заказ принят покупателем

Банк: FUNDS_SECURED

Платформа отправляет `ReleaseFunds`

Банк: RELEASE_PROCESSING

Только после фактического подтверждения — Банк: FUNDS_RELEASED

Платформа переводит сделку в итоговый статус: COMPLETED

Клиентский frontend не должен самостоятельно определять финансовое состояние сделки.

10. Требования к безопасности

Для интеграции необходимо предоставить требования Банка к: server-to-server авторизации; подписанию запросов; TLS / mTLS; IP whitelist, если применяется; управлению ключами; разграничению Sandbox и Production; rate limits; журналированию финансовых операций.

Платформа не планирует получать или хранить полный PAN, CVV и иные платёжные реквизиты, если это не является обязательным требованием банковской схемы.

11. Спор

Спор в рамках MVP является бизнес-процессом платформы. После открытия спора средства должны оставаться защищёнными и не перечисляться исполнителю автоматически. Решение принимает уполномоченный сотрудник платформы. Результат в MVP: `ReleaseFunds` — перечислить 100% исполнителю, либо `RefundFunds` — вернуть 100% покупателю.

Отдельный API freeze / unfreeze не является обязательным, если банковская модель изначально исключает автоматическое перечисление средств без команды `ReleaseFunds` .

12. Критерий готовности MVP

Интеграция считается достаточной для MVP, если в Sandbox возможно пройти два полных сценария.

Сценарий 1 – успешная сделка:

создание сделки → создание банковской защищённой сделки → оплата → FUNDS_SECURED → выполнение заказа → подтверждение покупателем → ReleaseFunds → FUNDS_RELEASED

Сценарий 2 – возврат:

создание сделки → оплата → FUNDS_SECURED → отмена или решение по спору → RefundFunds → FUNDS_REFUNDED

13. Вопросы к Банку

Банковская модель

1. Какой существующий продукт Банка может использоваться для описанного сценария? Применим ли Escrow API, представленный в каталоге developer.bcc.kz?
2. Может ли существующий механизм эскроу применяться для сделки между физическим лицом и ИП/ООО при заказе товаров и работ, или он ограничен сделками с недвижимостью?
3. На каком банковском счёте и в каком юридическом режиме находятся средства до завершения сделки?
4. Какой максимальный срок средства могут находиться в таком состоянии?
5. Может ли исполнитель получить средства каким-либо способом до наступления условия выплаты?
6. Может ли выплата происходить исключительно после разрешённой команды платформы?

Участники

7. Необходимо ли покупателю быть клиентом Банка?
8. Необходимо ли исполнителю иметь счёт в Банке?
9. Возможна ли выплата исполнителю на IBAN другого банка?
10. Какие процедуры KYC и KYB необходимы покупателю и исполнителю?

API

11. Возможно ли создавать или регистрировать защищённую сделку через API?
12. Возможно ли через API получать подтверждение фактического обеспечения сделки средствами?
13. Возможно ли инициировать выплату исполнителю через API?
14. Возможно ли инициировать полный возврат покупателю через API?
15. Какие API позволяют получить актуальный статус сделки и финансовой операции?
16. Какие callbacks и webhooks доступны?
17. Предоставляется ли Sandbox и каков порядок получения тестовых ключей?
18. Какие механизмы авторизации и подписи запросов используются?
19. Какие ограничения по количеству запросов и SLA применяются?

Ограничения продукта

20. Какие минимальные и максимальные суммы сделки поддерживаются?
21. Какой максимальный срок действия защищённой сделки?
22. Какие комиссии применяются к приёму средств, хранению и эскроу, выплате и возврату? Возможна ли модель «один счёт платформы с субсчётами по сделкам» вместо отдельного счёта на каждую сделку?

23. Как обрабатываются комиссии при полном возврате?
24. Какие способы пополнения защищённой сделки доступны покупателю?

14. Возможности следующих этапов

Следующие функции не входят в MVP, однако просим указать, поддерживаются ли они банковской архитектурой: частичная выплата исполнителю; частичный возврат покупателю; несколько этапов финансирования; несколько этапов выплаты; изменение суммы сделки; продление срока сделки. Их отсутствие не является блокером для запуска первой версии.

15. Ожидаемый результат

По результатам рассмотрения документа просим предоставить сопоставление требований с существующими банковскими продуктами и API.

Требование	Продукт / API Банка	Доступность	Основные ограничения
Создание защищённой сделки			
Внесение средств			
Получение статуса			
Подтверждение защищённых средств			
Выплата исполнителю			
Возврат покупателю			
Статус финансовой операции			
Webhooks / callbacks			
Reconciliation			
Sandbox			

Также просим предоставить: техническую документацию API; описание банковского продукта; требования к подключению; Sandbox credentials либо порядок их получения; тарифы и лимиты; требования к участникам сделки.

16. Ключевой вопрос интеграции

Компании необходим банковский механизм, при котором покупатель перечисляет средства по конкретной сделке → банк подтверждает их наличие и защищённое состояние → исполнитель видит факт обеспечения сделки, но не может получить деньги → после приёмки платформа через API инициирует выплату исполнителю → при отмене или соответствующем решении средства могут быть возвращены покупателю.

Просим Банк определить, какой существующий банковский продукт и набор API позволяют реализовать данный жизненный цикл сделки и какие ограничения необходимо учитывать при запуске MVP.

ТОО «МАСШТАБ ARNA» · БИН 260740041659 · г. Астана

Каныбек Мадияр, основатель · +7 701 556 99 97 · madiyarknbk@gmail.com · демо: escrowkz-web.vercel.app