

ТЕСТИРОВАНИЕ ПЛАТЕЖЕЙ В АНДРОИД ПРИЛОЖЕНИИ



Дмитрий Макаренко

Android QA

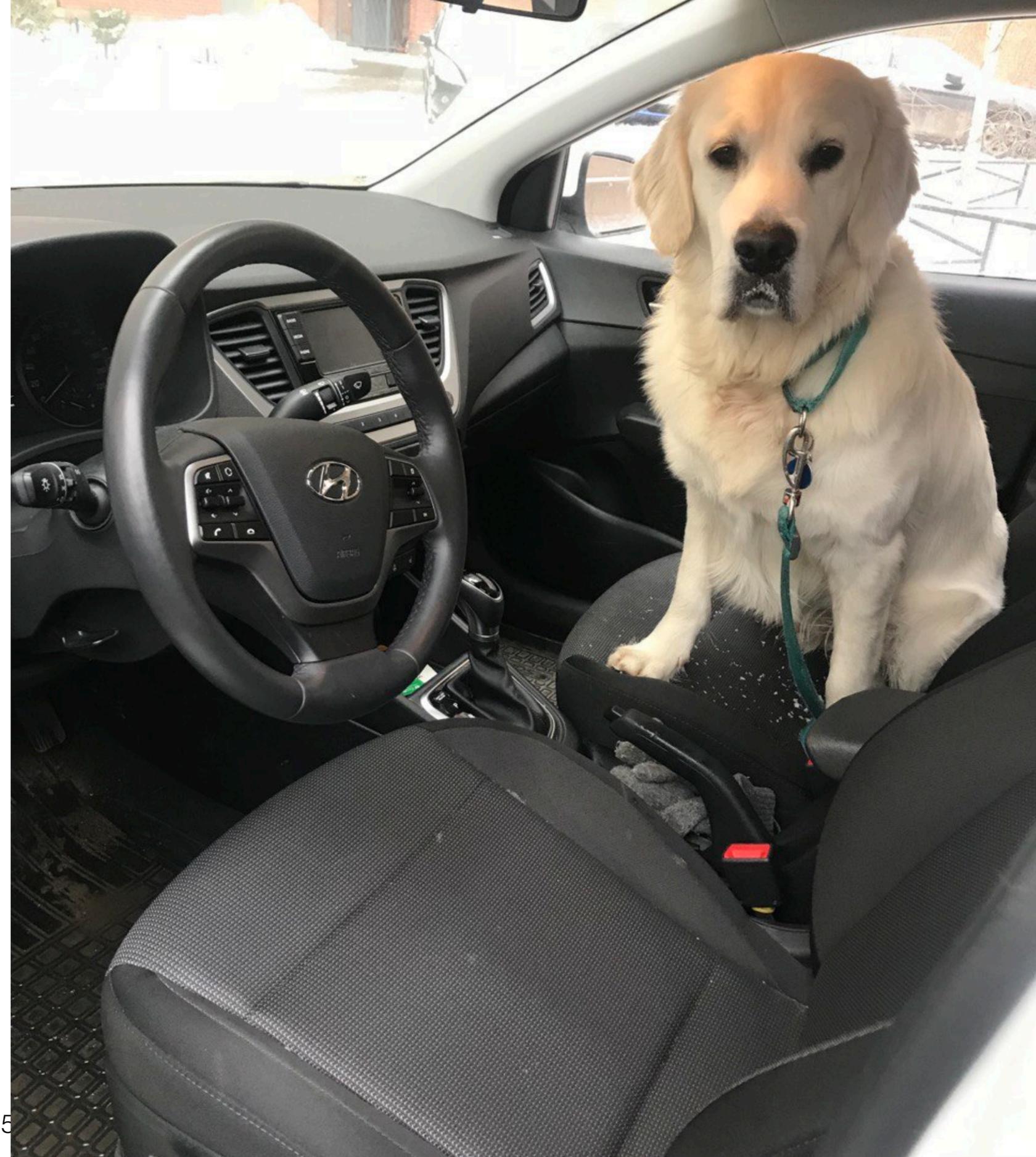
**Учился и начинал
работать в Питере**



**5+ лет в
тестировании**



**2.5 года назад
переехал в Лондон**



Тестирую Android-
приложение Vadoo





A world map in a dark blue color with numerous small pink heart-shaped markers scattered across the continents, representing global users. The map is centered on the Atlantic Ocean.

~500 000 000

людей по всему миру пользуются нашими приложениями

Биллинг в 250 странах



~20
провайдеров
мобильных
платежей



**Помогаем с
тестированием
и интеграцией
SDK**



**Почему я хочу поделиться
с вами этими знаниями?**

ME

Responsibilities





Или так



План доклада

1. Какие платежи тестируем

План доклада

1. Какие платежи тестируем
2. Как их тестируем

Какие платежи тестируем

Что тестировать Платные продукты

Типы платных продуктов

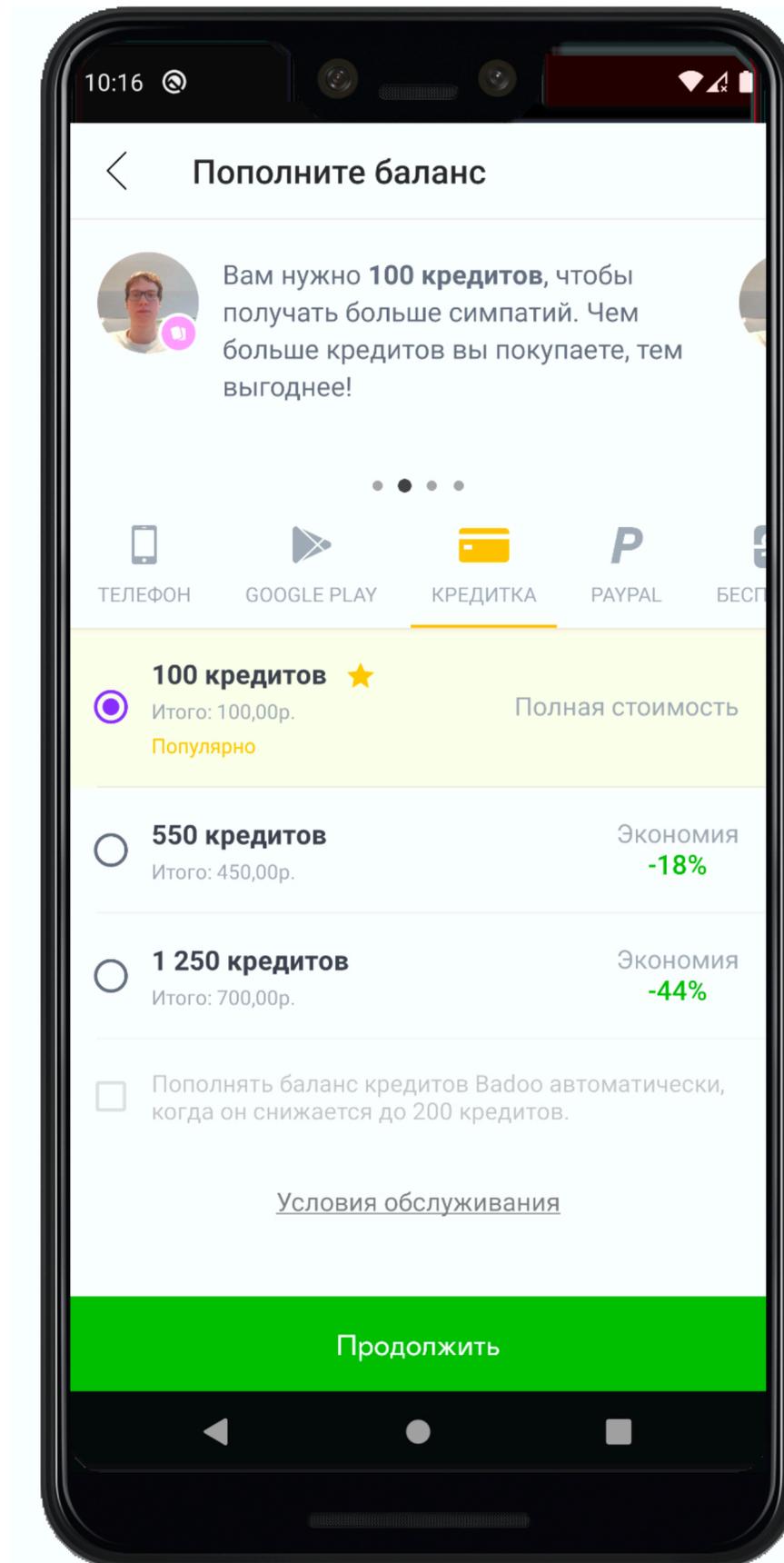
Типы платных продуктов

- Кредиты

Типы платных продуктов

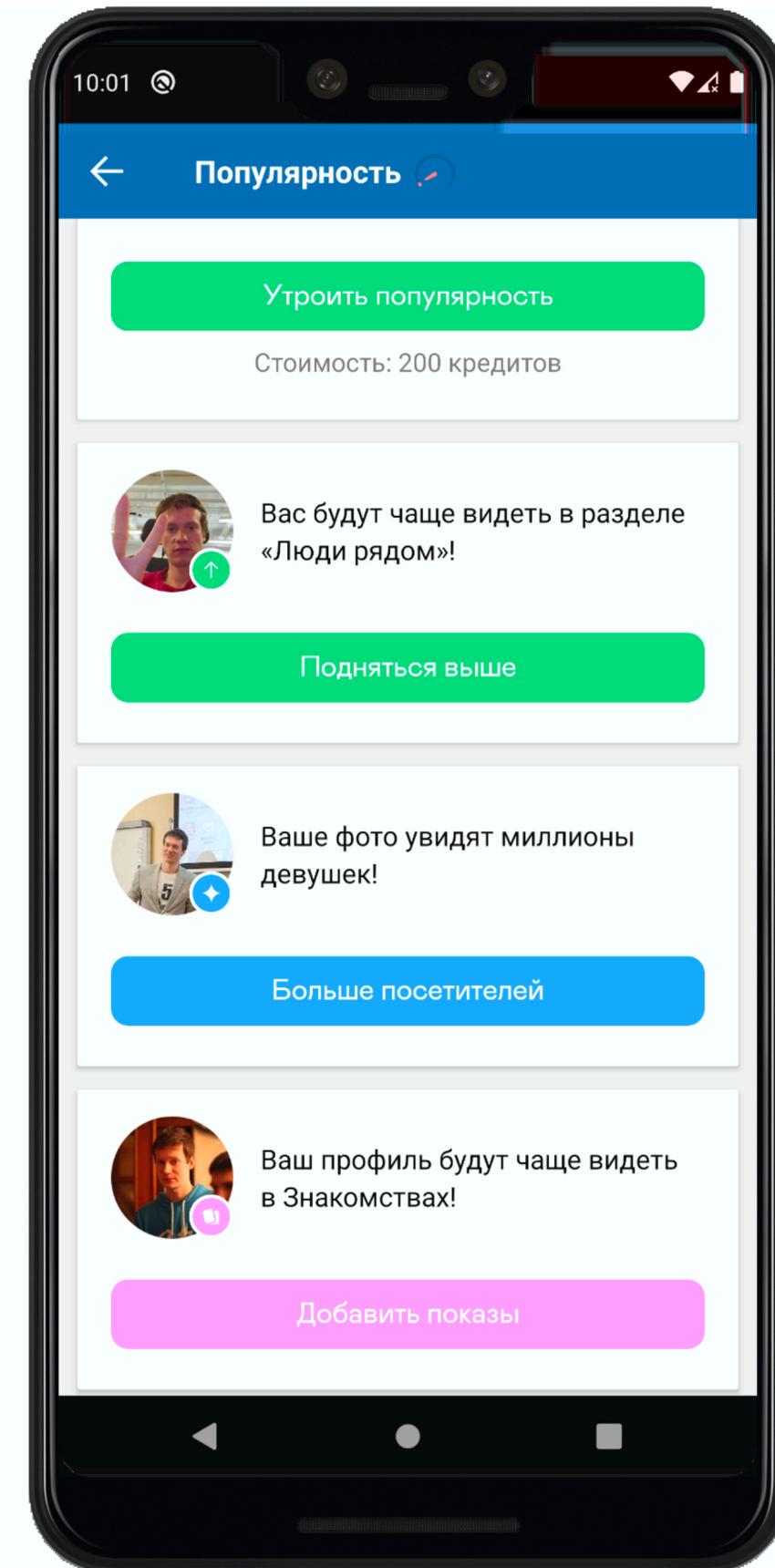
- Кредиты
- Подписки

Кредиты



Что тестировать Платные продукты

Кредиты



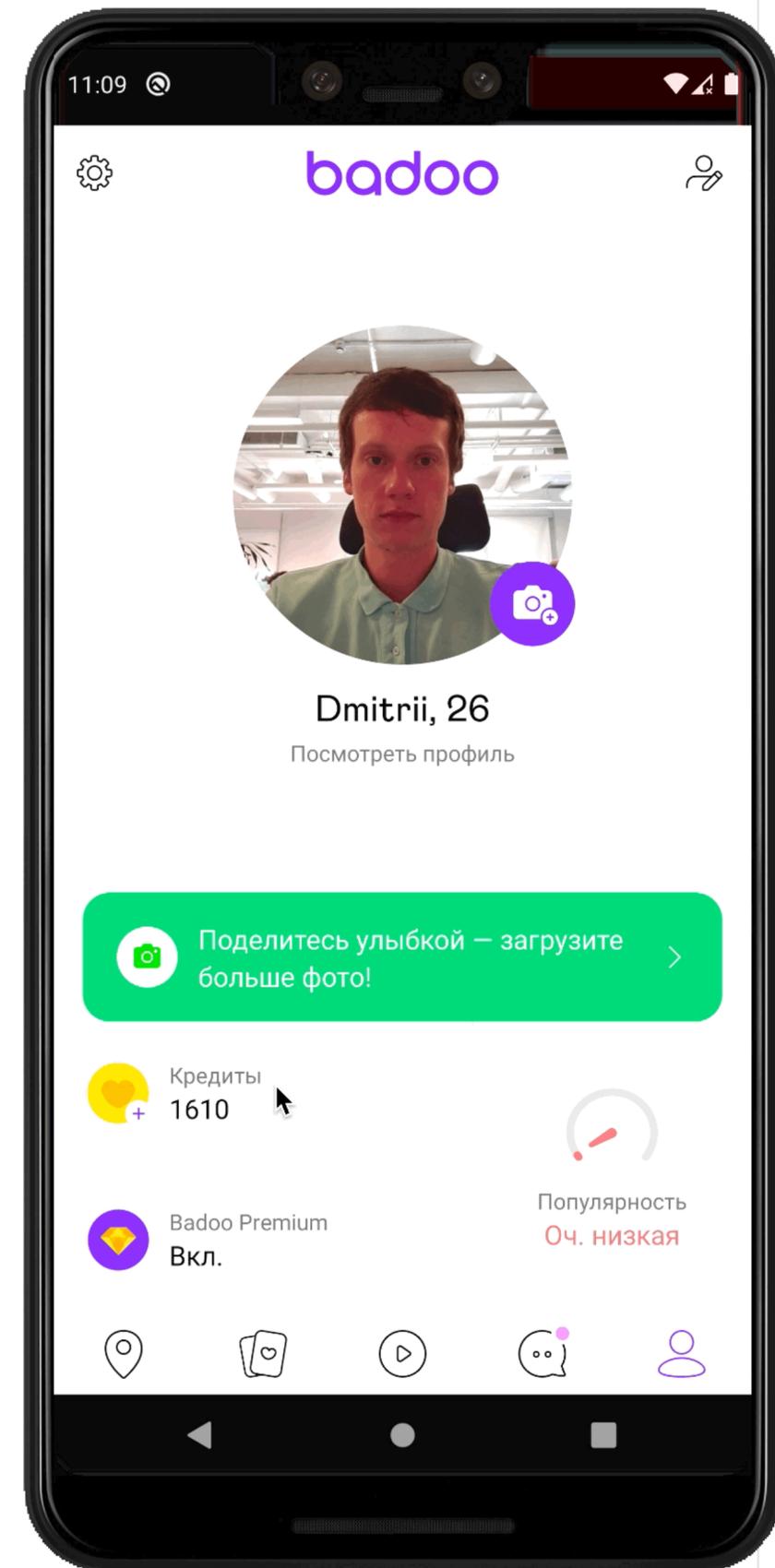
Варианты покупки сервисов

- Достаточно кредитов

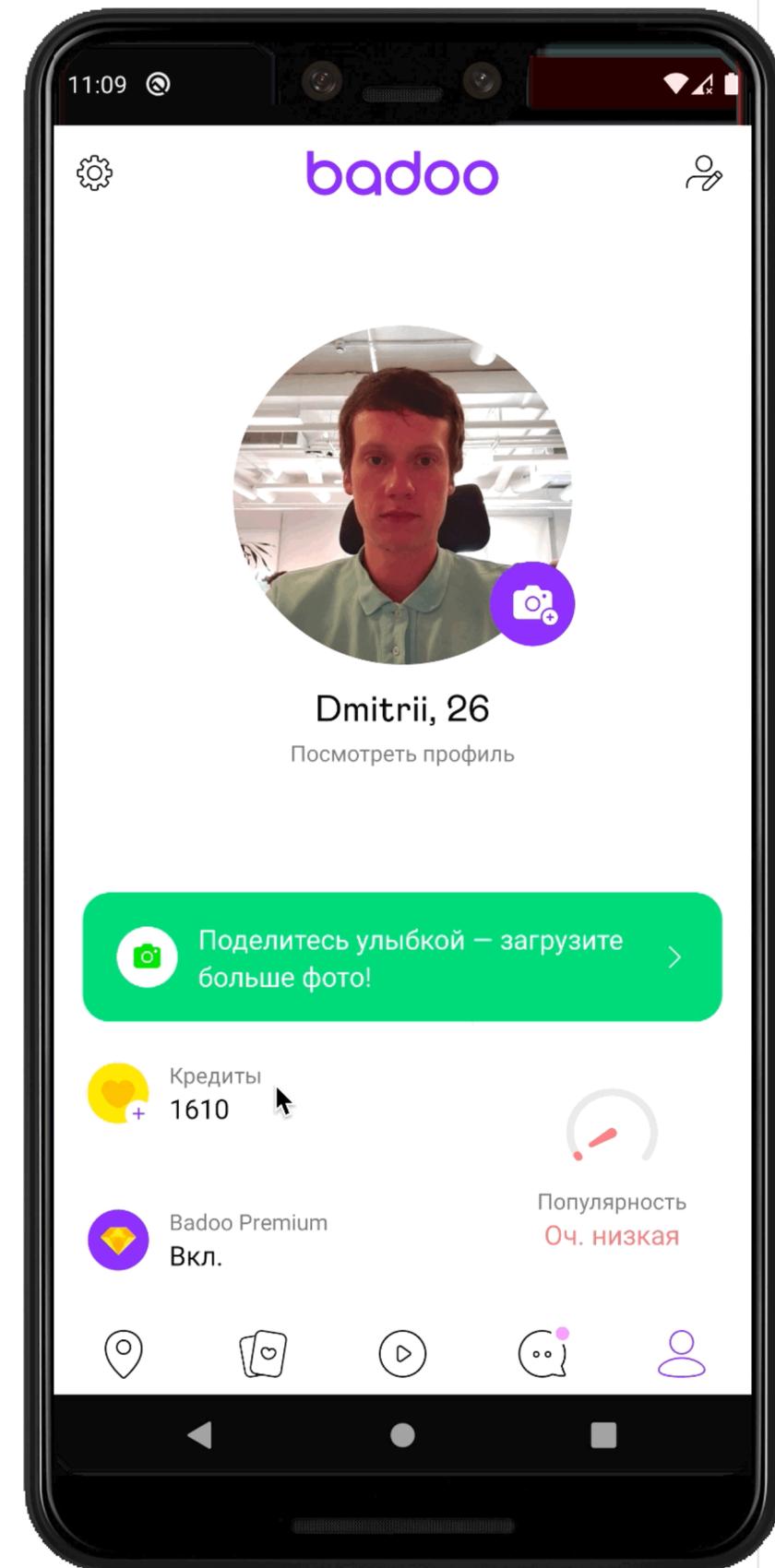
Варианты покупки сервисов

- Достаточно кредитов
- Недостаточно кредитов

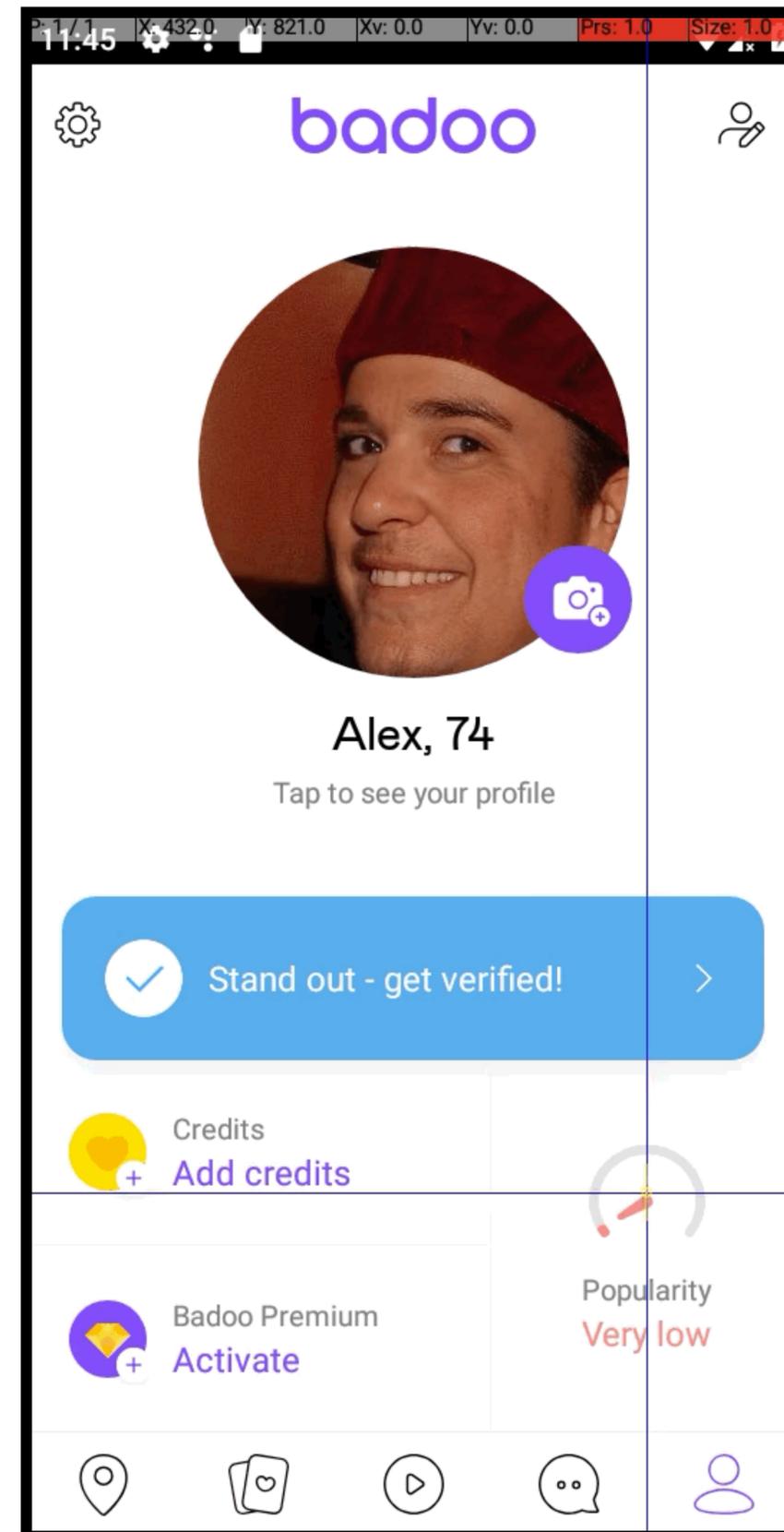
Достаточно кредитов



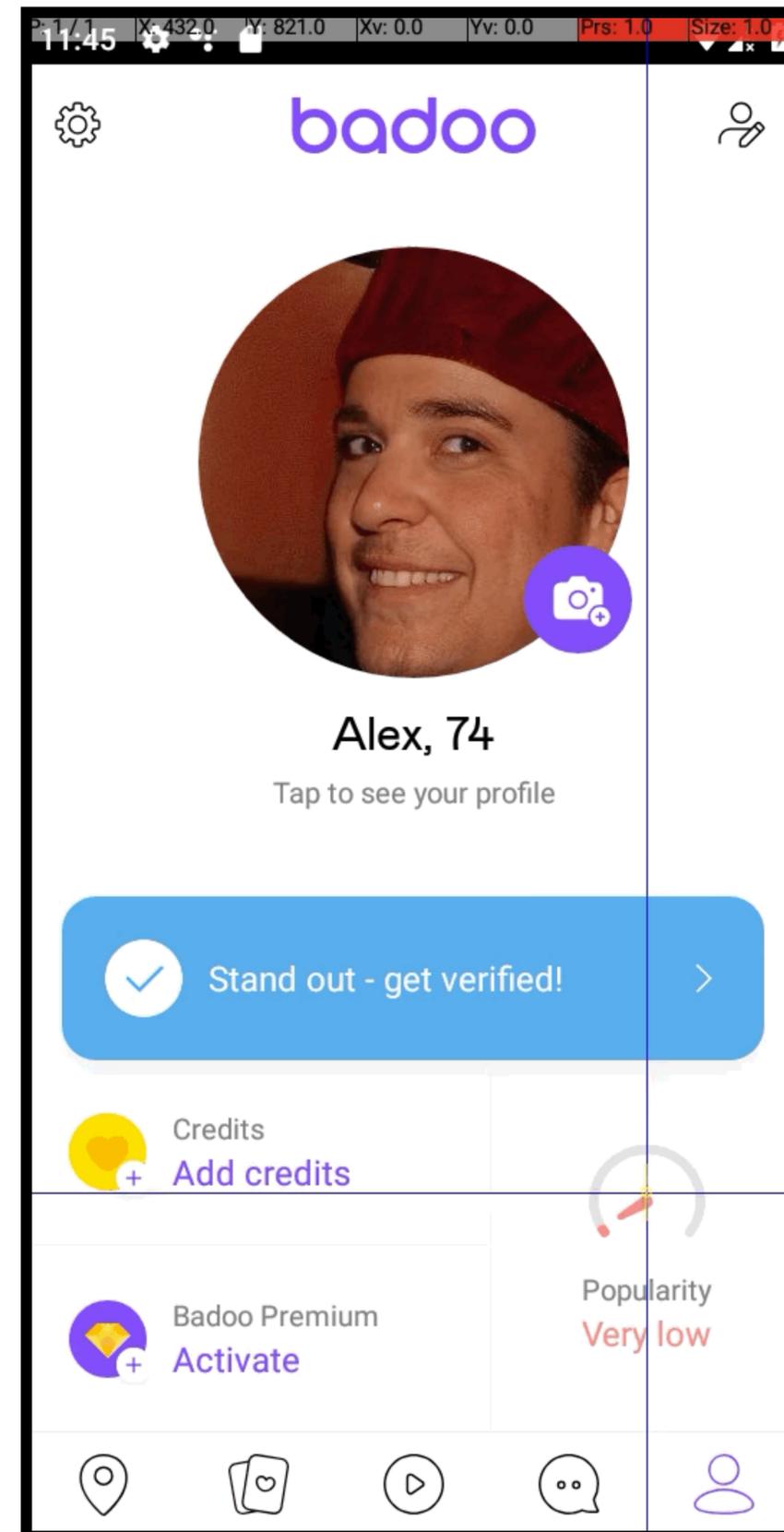
Достаточно кредитов



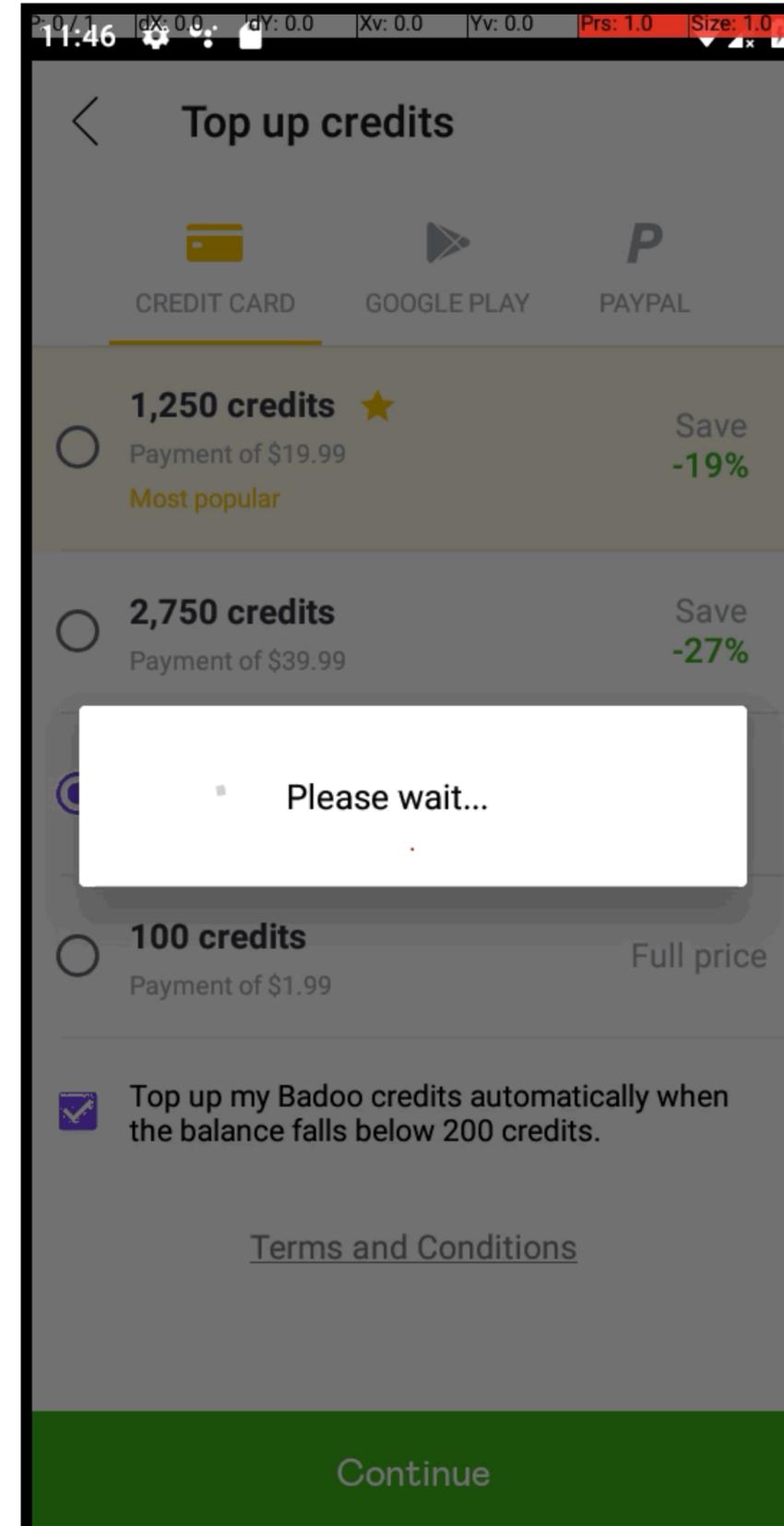
Недостаточно кредитов



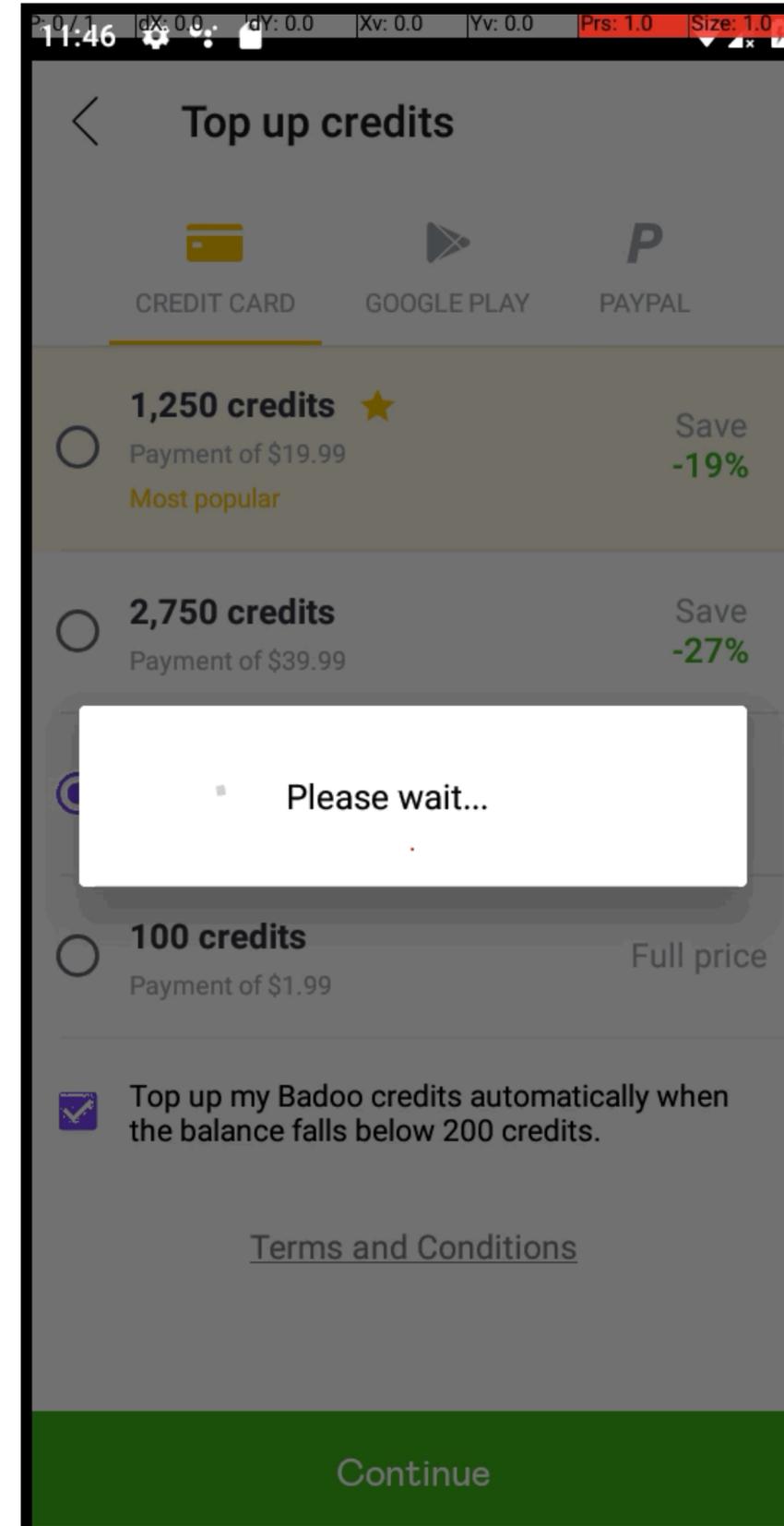
Недостаточно кредитов



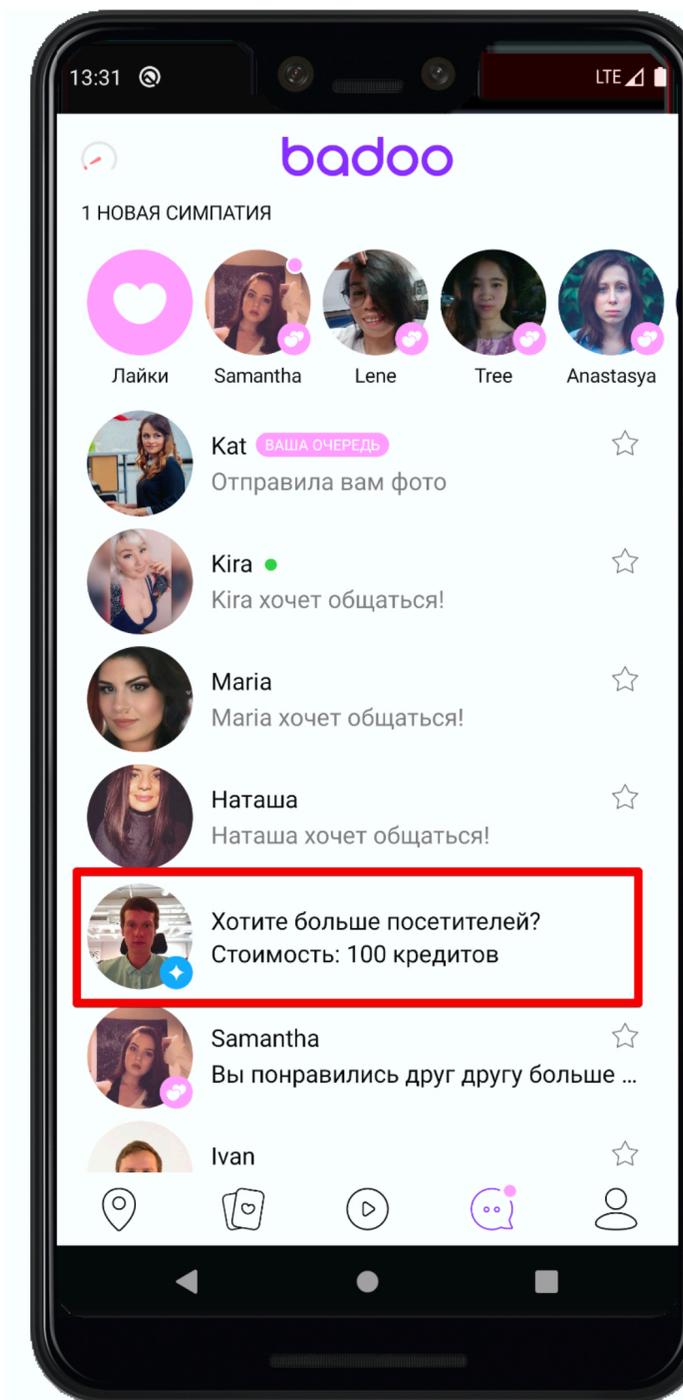
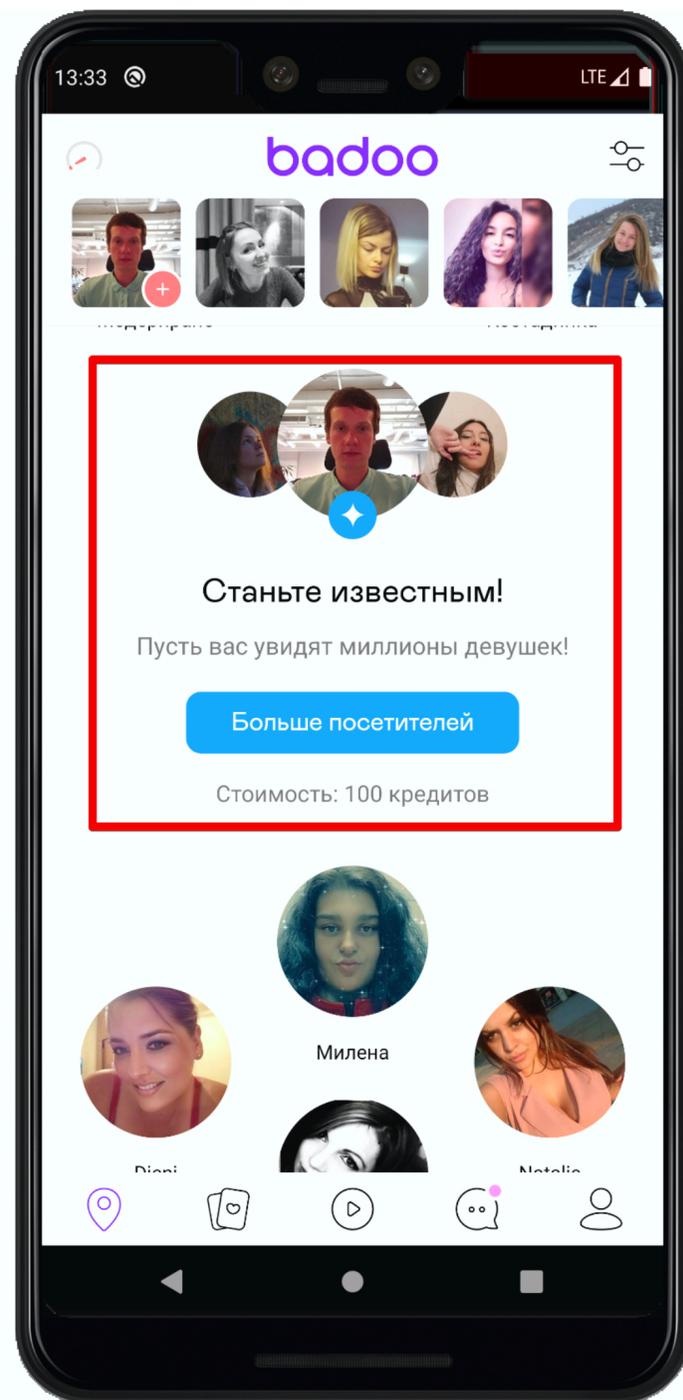
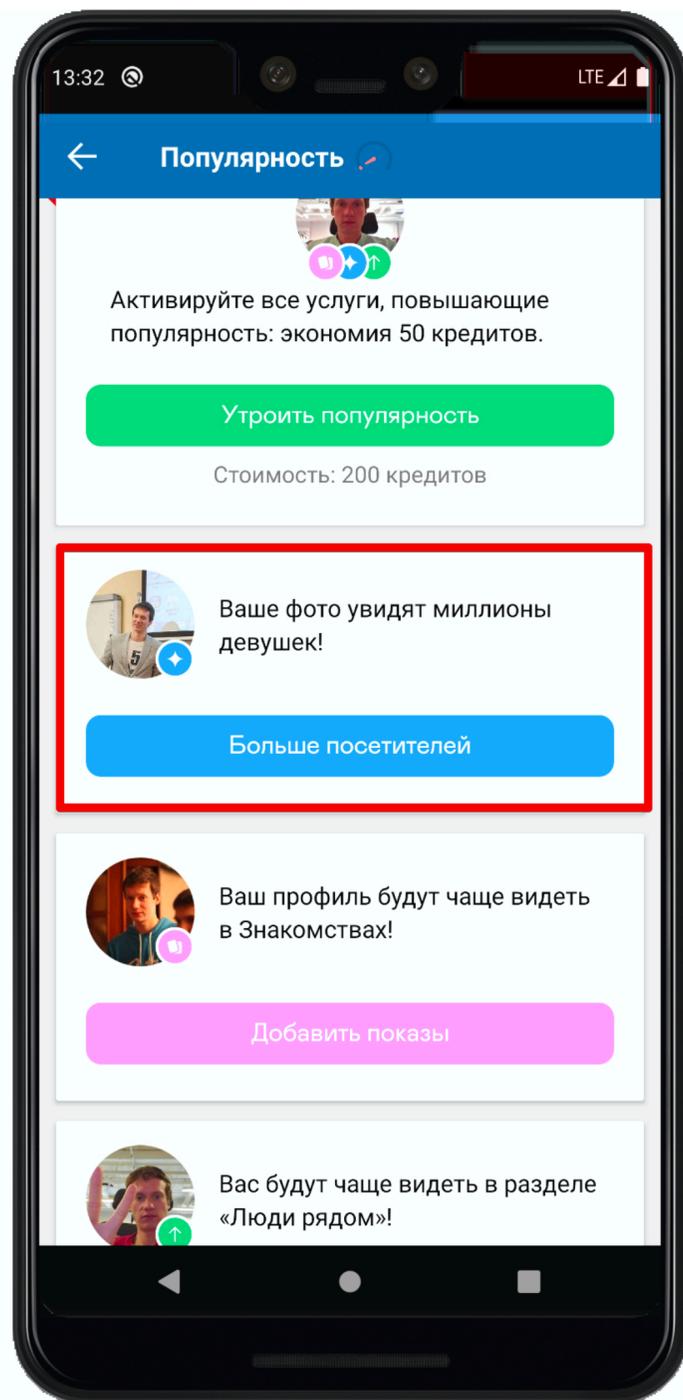
Недостаточно кредитов



Недостаточно кредитов



Что тестировать Покупка сервисов



Что тестировать Платные продукты

Подписки

Что тестировать Платные продукты

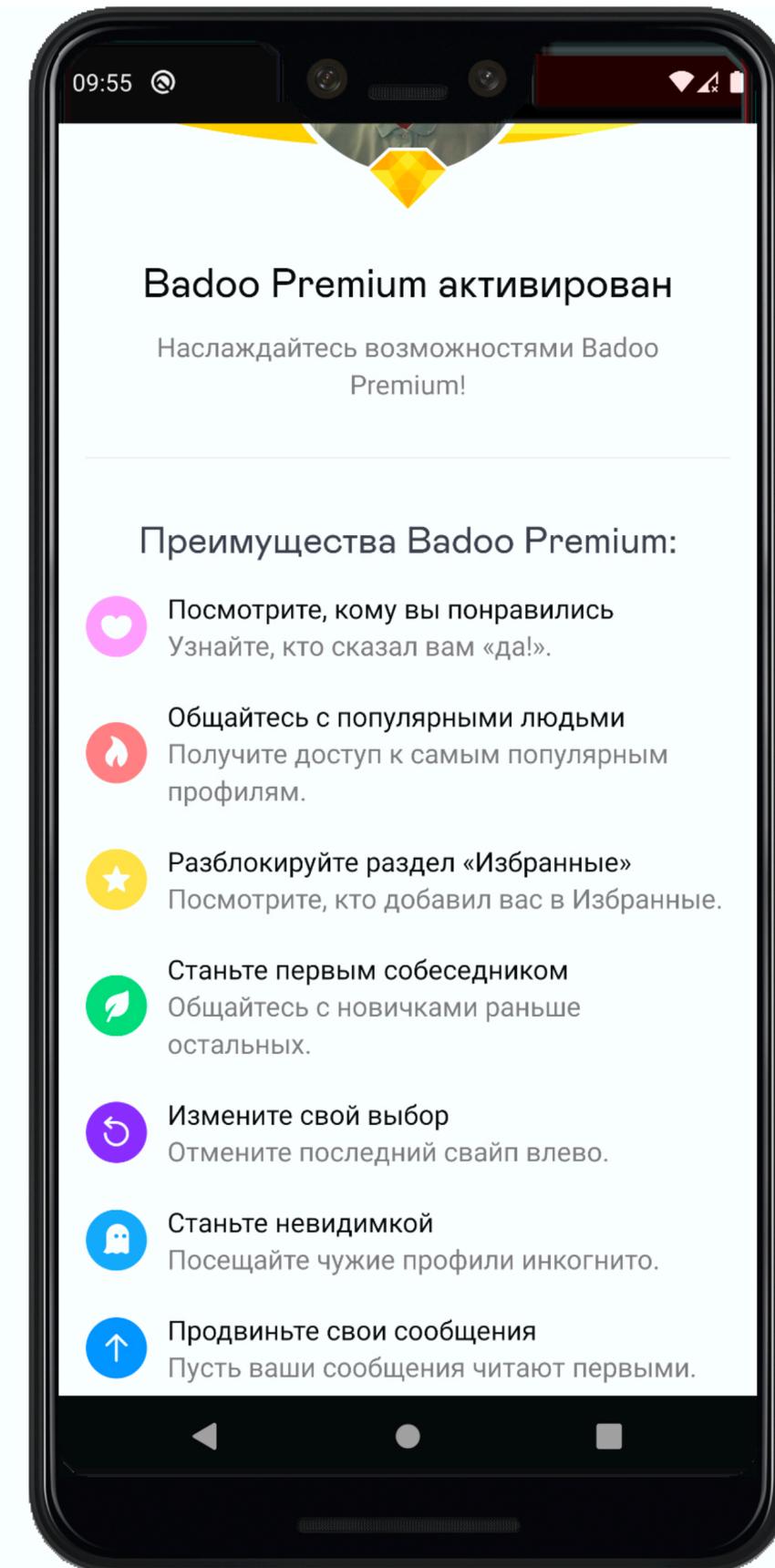
Подписки

- Премиум

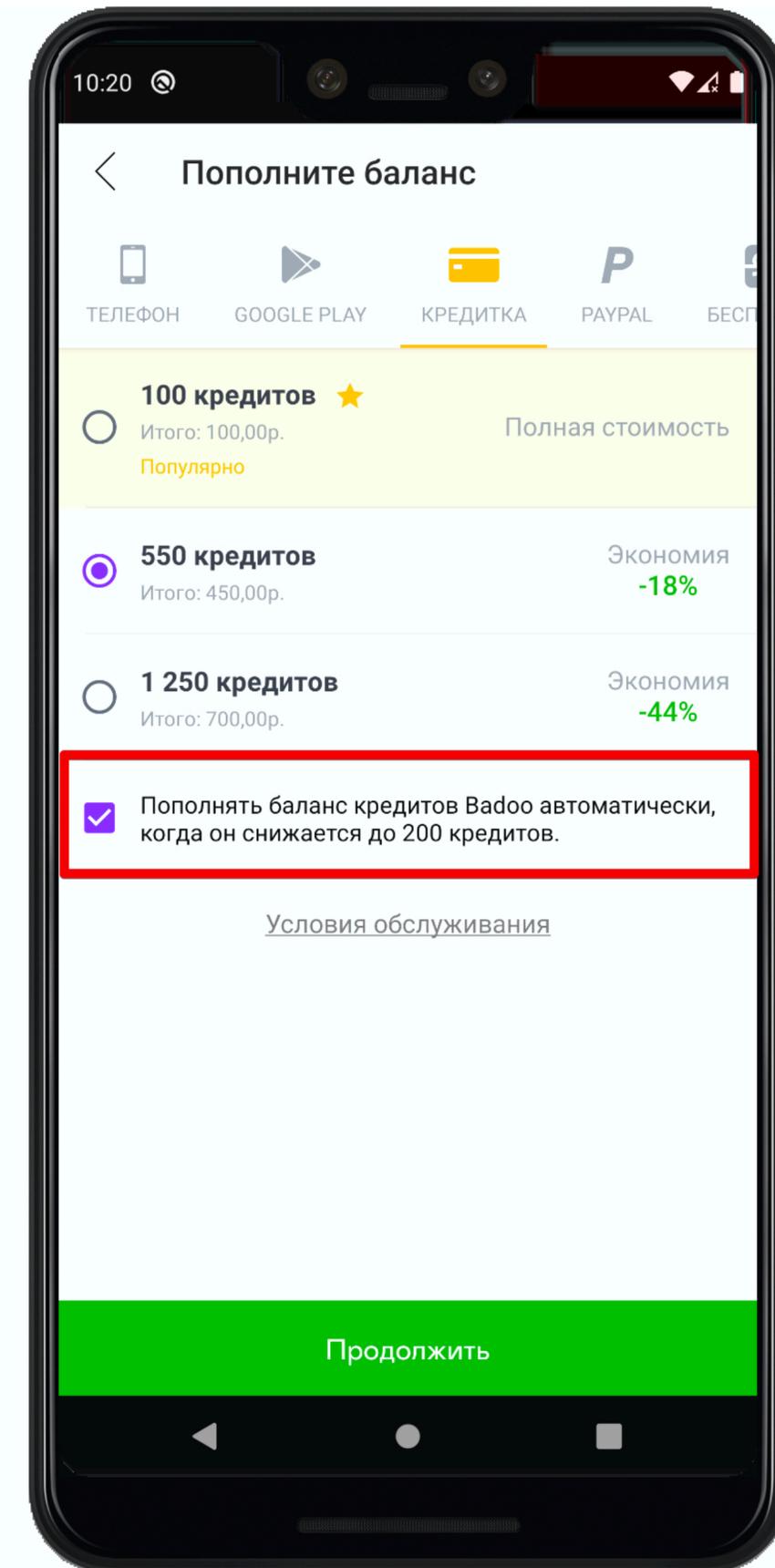
Подписки

- Премиум
- Автопополнение кредитов

Премиум



Автопополнение кредитов



Что тестировать

ЧТО МОЖНО КУПИТЬ

1. Два типа платных продуктов

Что тестировать

ЧТО МОЖНО КУПИТЬ

1. Два типа платных продуктов
2. Разные сервисы за кредиты

ЧТО МОЖНО КУПИТЬ

1. Два типа платных продуктов
2. Разные сервисы за кредиты
 - **Доступны с разных экранов**

ЧТО МОЖНО КУПИТЬ

1. Два типа платных продуктов
2. Разные сервисы за кредиты
 - Доступны с разных экранов
 - **Достаточно кредитов для покупки**

ЧТО МОЖНО КУПИТЬ

1. Два типа платных продуктов
2. Разные сервисы за кредиты
 - Доступны с разных экранов
 - Достаточно кредитов для покупки
 - **Недостаточно кредитов для покупки**

Методы оплаты

- 4 глобальных метода оплаты

Методы оплаты

- 4 глобальных метода оплаты



Методы оплаты

- 4 глобальных метода оплаты



Методы оплаты

- 4 глобальных метода оплаты



Google Play

Методы оплаты

- 4 глобальных метода оплаты



Google Play



Методы оплаты

- 4 глобальных метода оплаты
- **Несколько локальных методов оплаты, доступных в конкретных странах**

Методы оплаты

- 4 глобальных метода оплаты
- Несколько локальных методов оплаты, доступных в конкретных странах
- **Предоставление продуктов/сервисов за рекламу**

Что тестировать

Полная регрессия

> 1 ДНЯ

Что тестировать

Особенности процессов

- Android – 1 релиз в неделю

Что тестировать

Особенности процессов

- Android – 1 релиз в неделю
- Сервер – 2 релиза в день

Особенности процессов

- Android – 1 релиз в неделю
- Сервер – 2 релиза в день
- **Отдельные команды разработки функциональности платежей на клиенте и сервере**

Особенности процессов

- Android – 1 релиз в неделю
- Сервер – 2 релиза в день
- Отдельные команды разработки функциональности платежей на клиенте и сервере
 - **Постоянный поток новых задач**

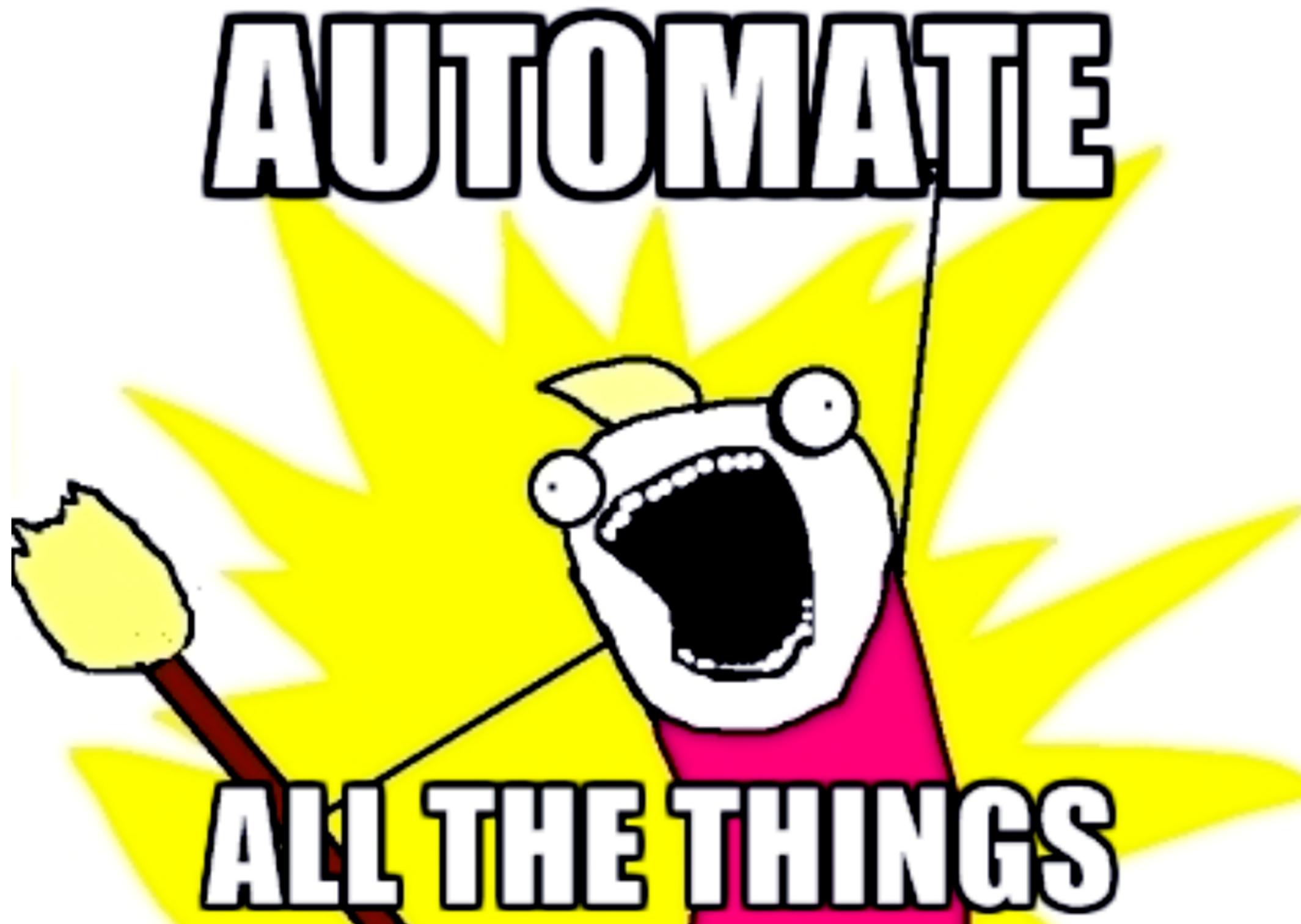
A cartoon illustration of a brown dog with large, expressive eyes and floppy ears, wearing a small brown hat. The dog is sitting on a simple wooden chair at a round wooden table. On the table sits a white mug. The dog is surrounded by large, jagged, yellow and orange flames that appear to be rising from the floor. The background is a light green wall with a dark doorway behind the dog and a window with dark shutters to the right. A dark, scalloped-edged canopy hangs from the top of the scene. A white speech bubble with a black outline is positioned above the dog's head.

THIS IS FINE.

Что тестировать

Или так





Что тестировать

АВТОМАТИЗАЦИЯ

- 4 глобальных метода оплаты

АВТОМАТИЗАЦИЯ

- 4 глобальных метода оплаты
- **Предоставление продуктов/сервисов за рекламу**

Другие виды контроля

- **Мониторинг + детекция аномалий**

Другие виды контроля

- Мониторинг + детекция аномалий
- Тестируем вручную

Как тестируем

Технологии

Calabash



Как тестируем

Глобальные методы оплаты

Тестовые конфигурации

- 1 метод – 1 конфигурация

Тестовые конфигурации

- 1 метод – 1 конфигурация
+ Локализация падений

Тестовые конфигурации

- 1 метод – 1 конфигурация
 - + Локализация падений
 - + Гибкость настройки

Тестовые конфигурации

- 1 метод – 1 конфигурация
 - + Локализация падений
 - + Гибкость настройки
 - **Дополнительное время на подготовку запуска**

План

- Процесс покупки

План

- Процесс покупки
- **Что надо изменить для тестирования**

План

- Процесс покупки
- Что надо изменить для тестирования
- **Особенности автоматизации**

План

- Процесс покупки
- Что надо изменить для тестирования
- Особенности автоматизации
- **Решение проблем**

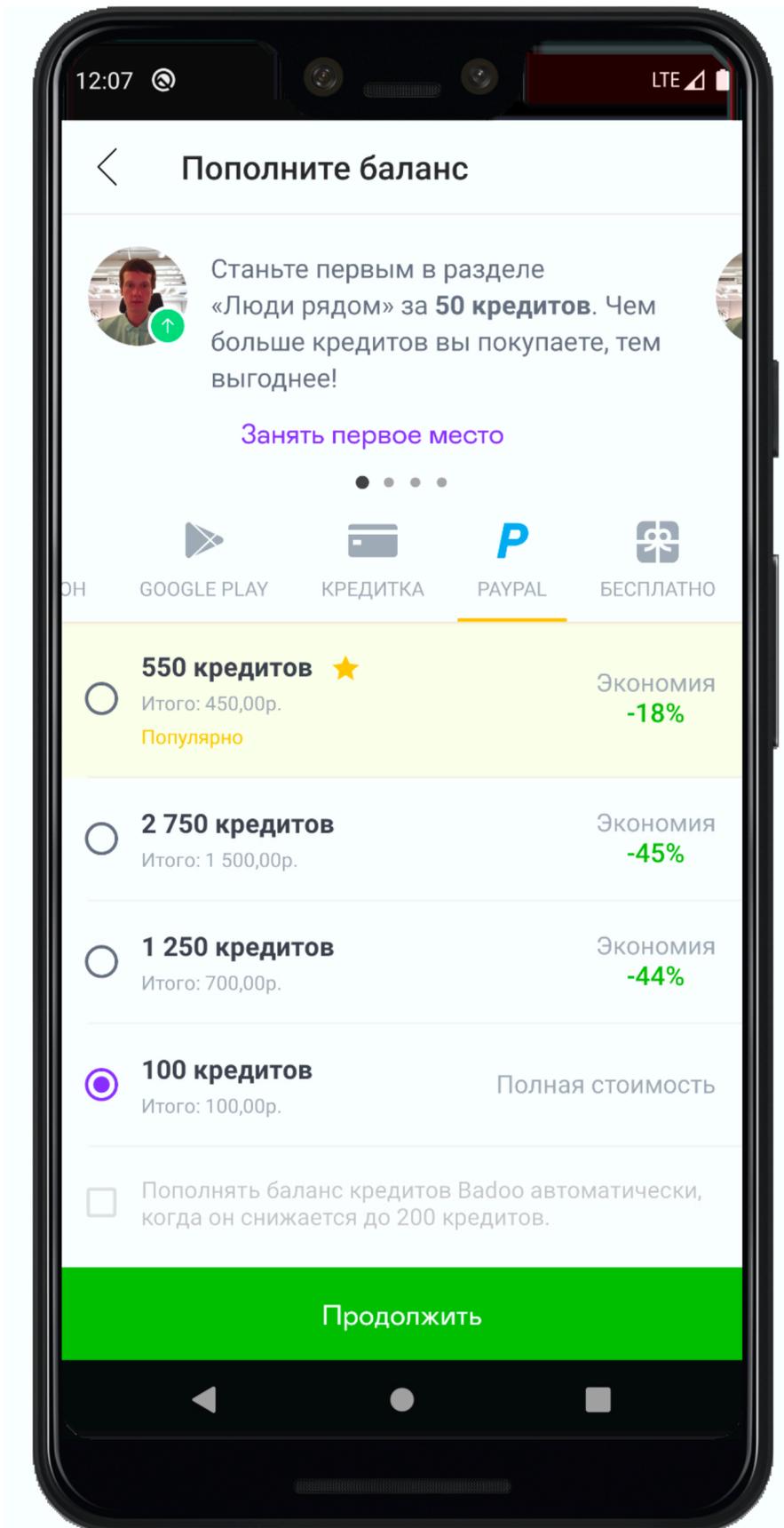
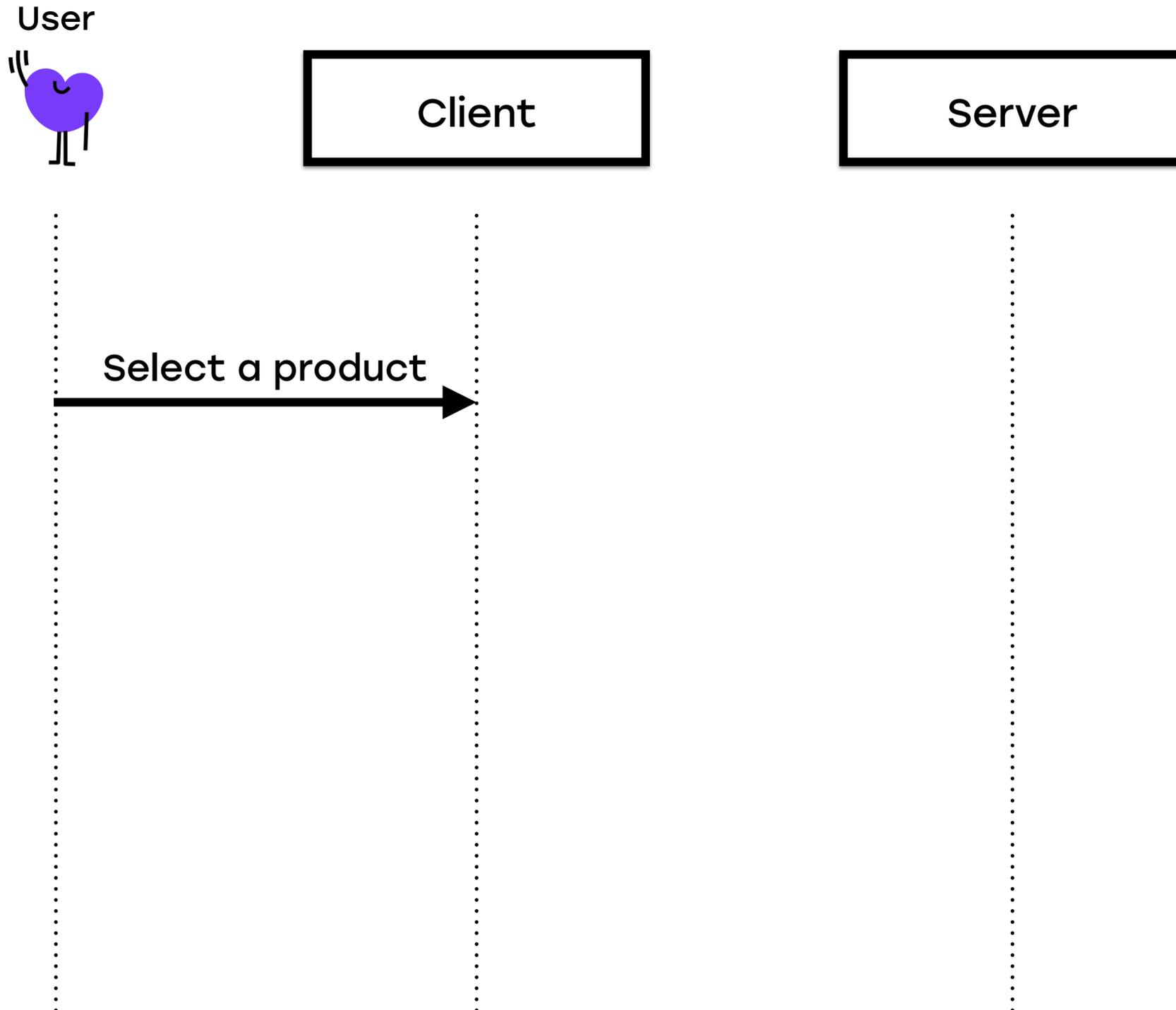
План

- Процесс покупки
- Что надо изменить для тестирования
- Особенности автоматизации
- Решение проблем
- **Специфические сценарии**

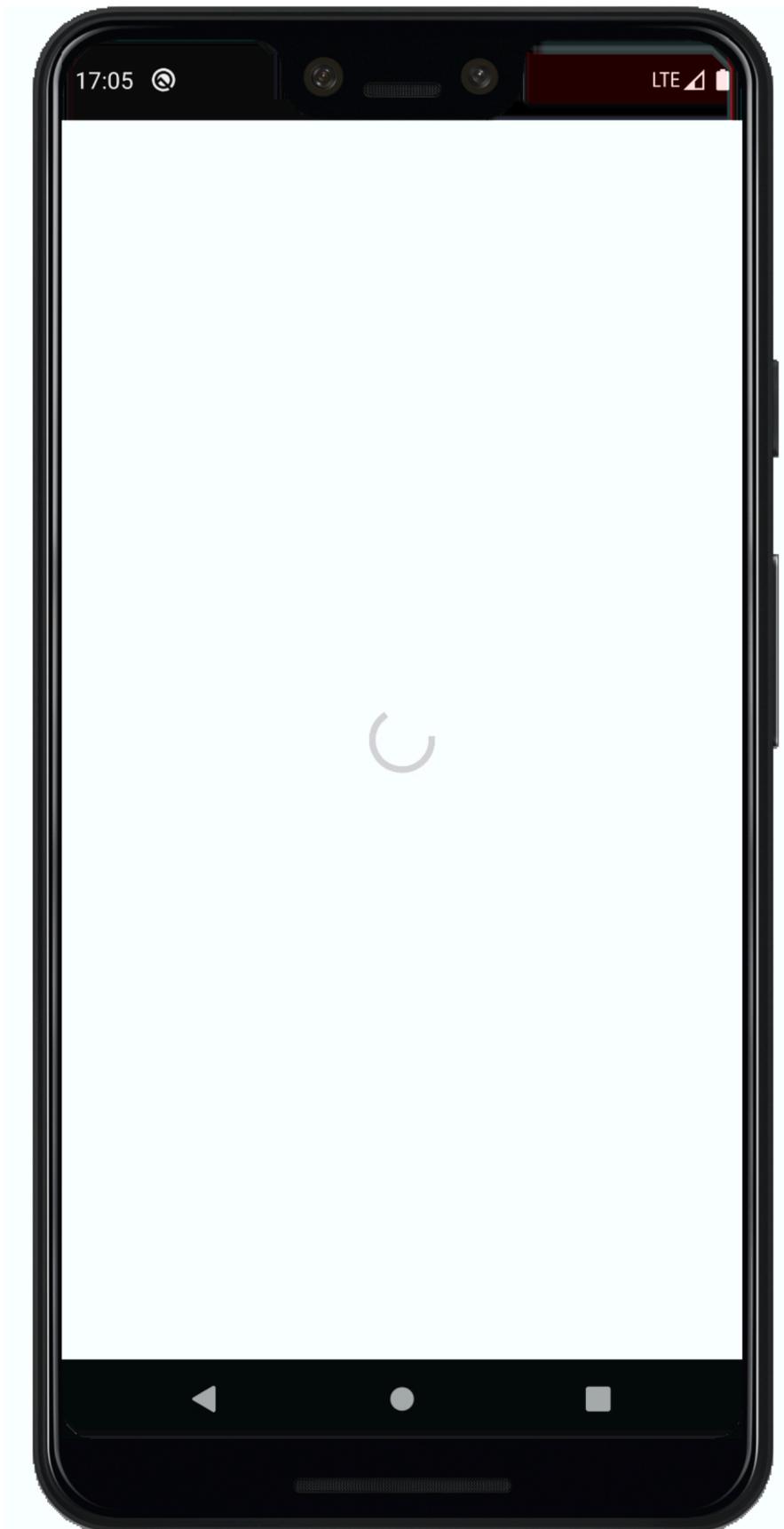
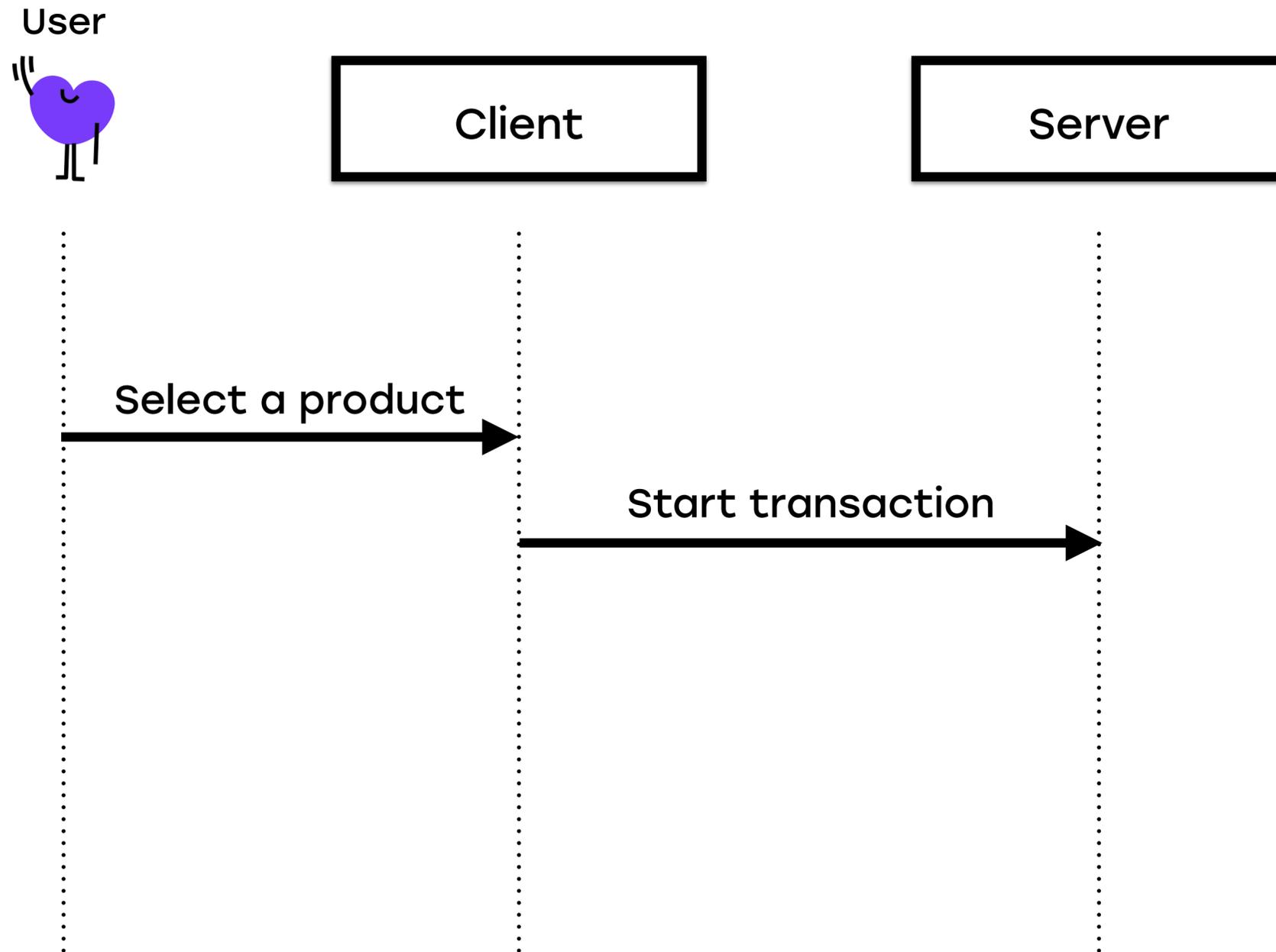
PayPal

Процесс покупки

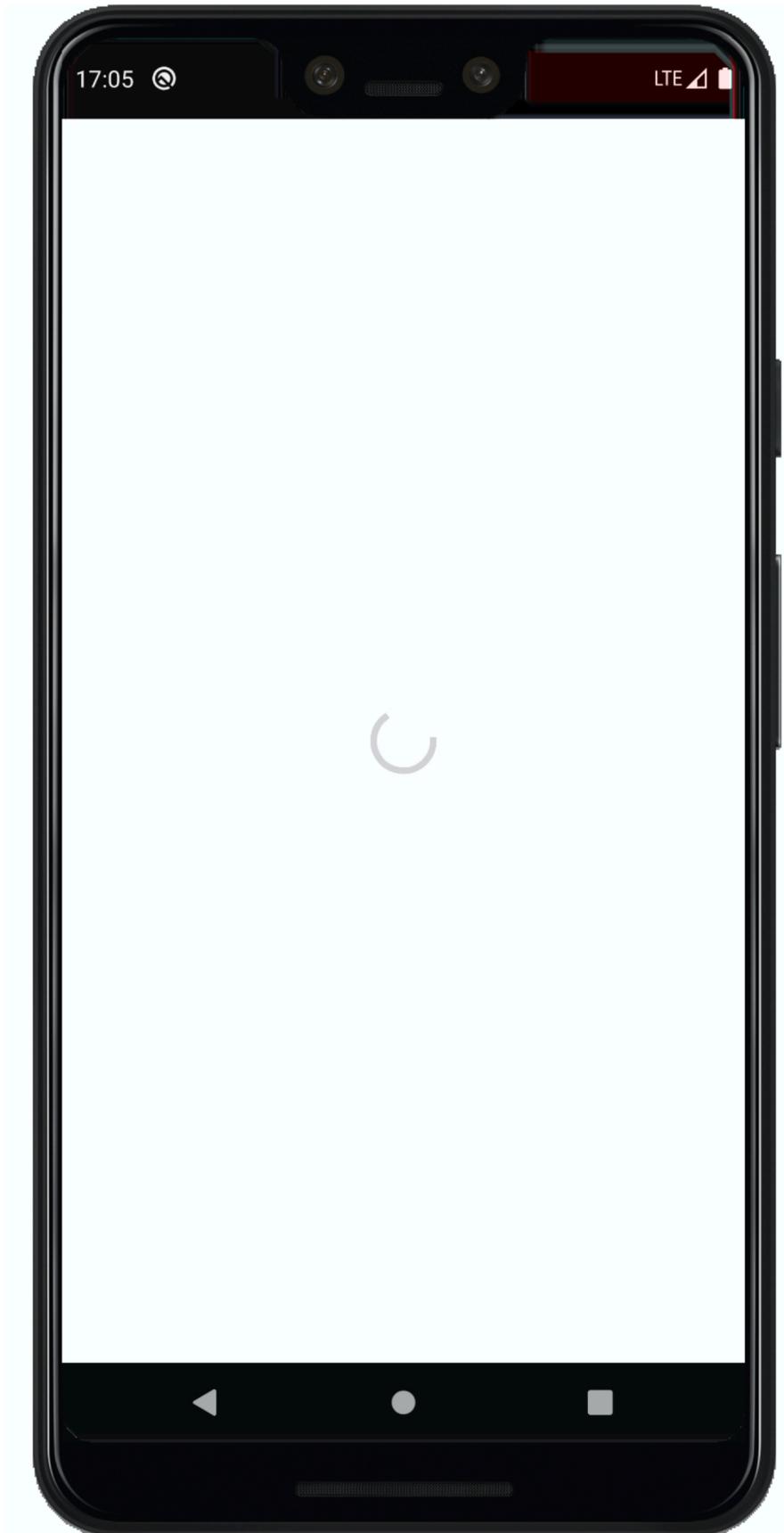
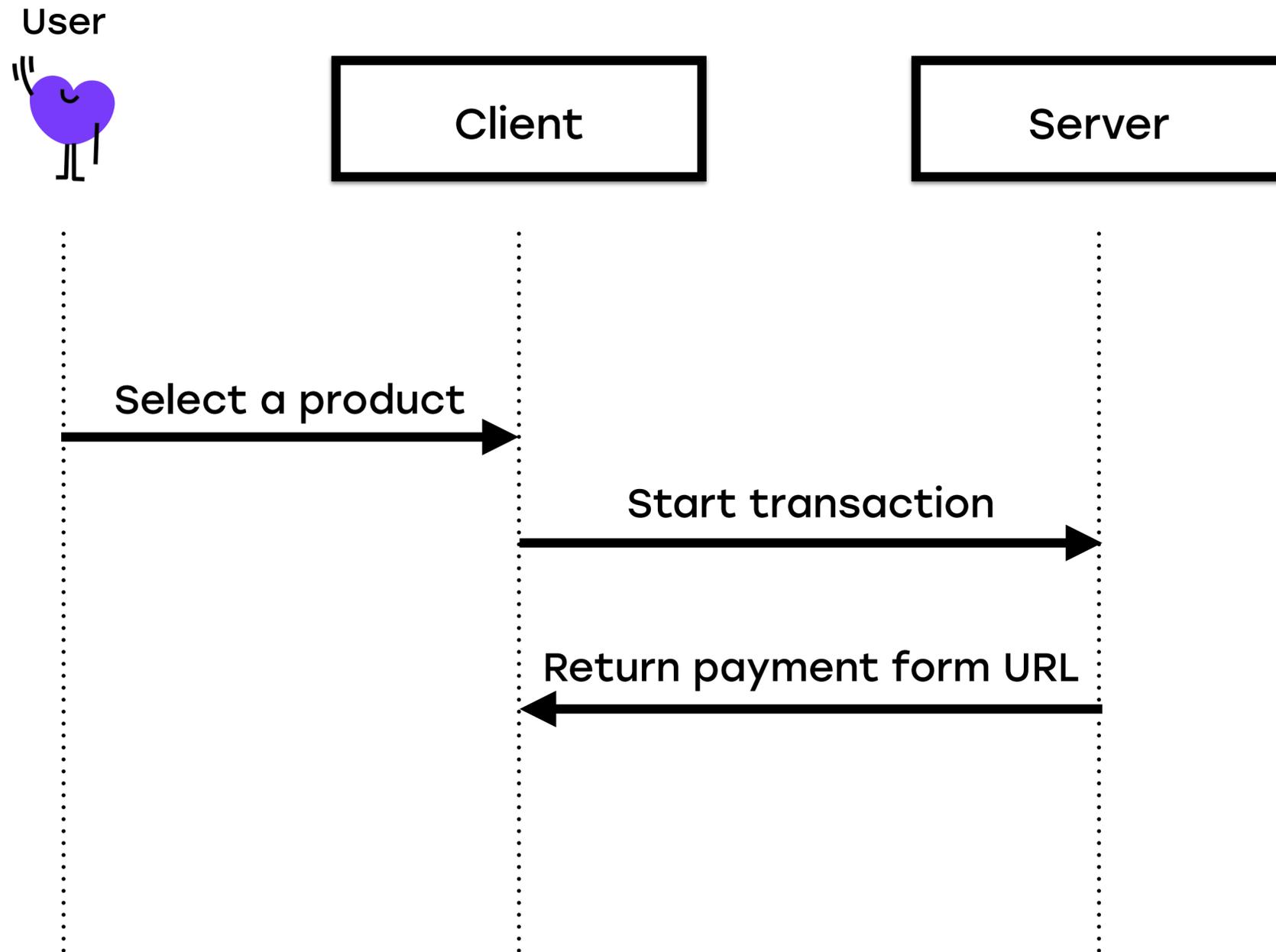
PayPal Процесс покупки



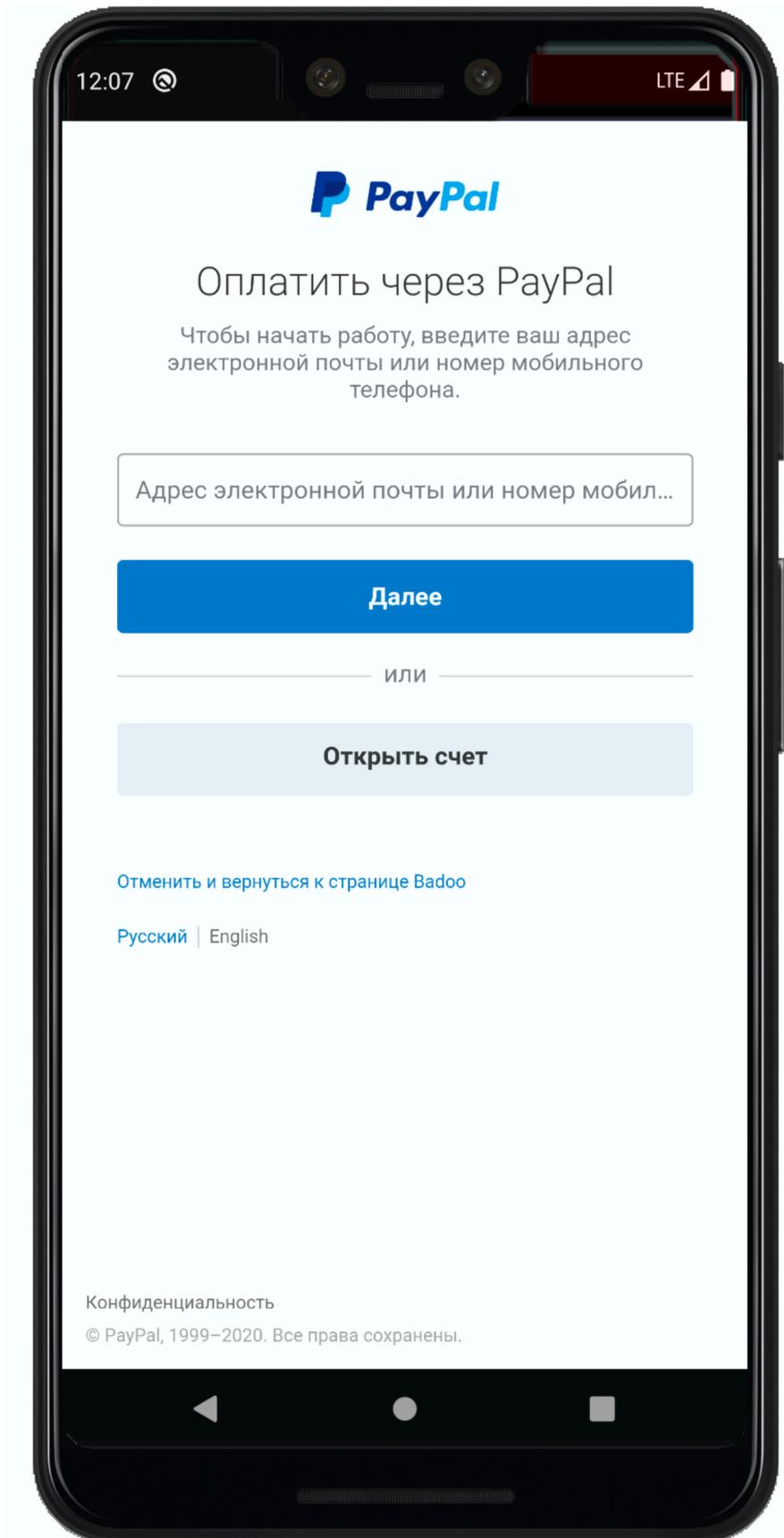
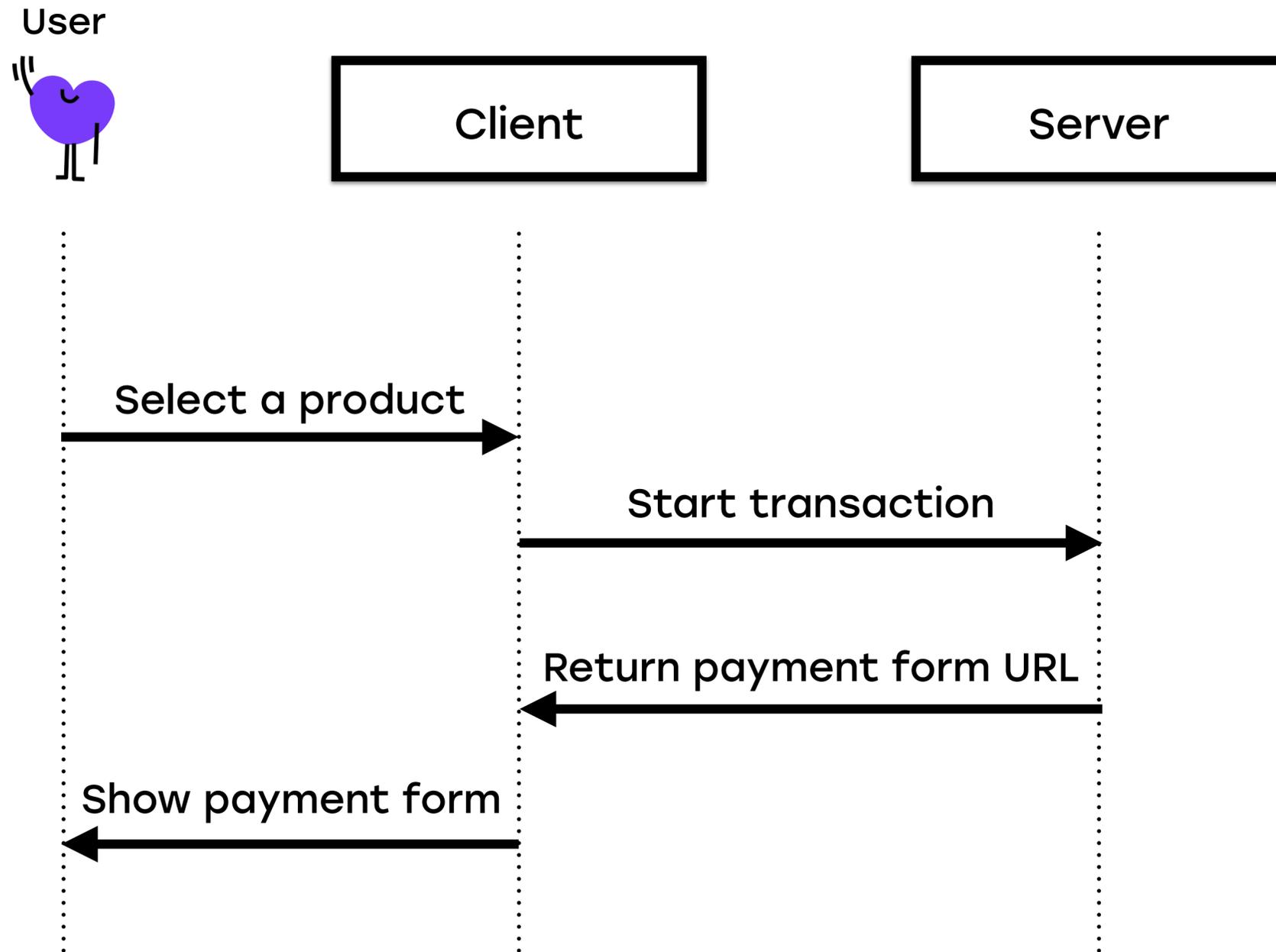
PayPal Процесс покупки



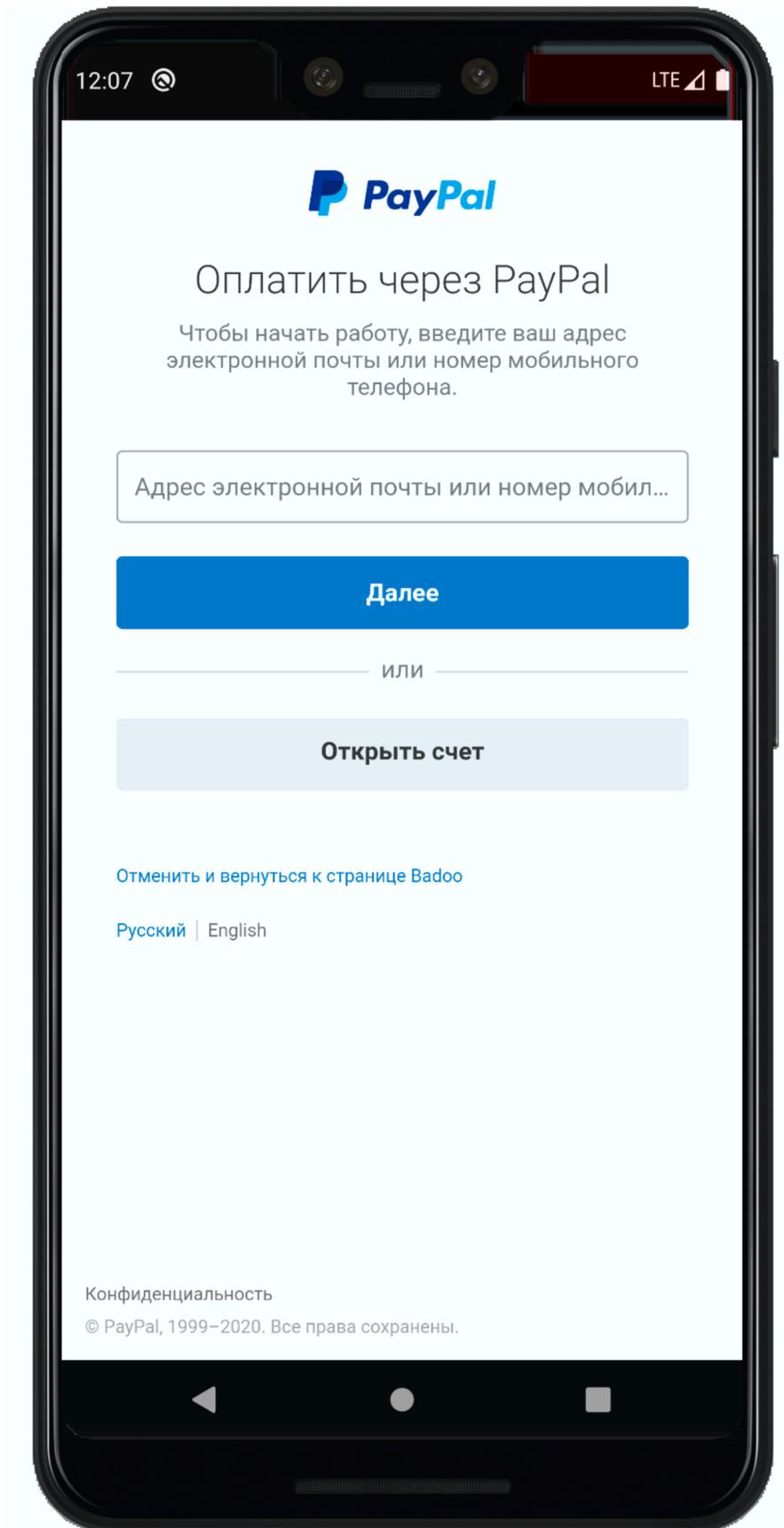
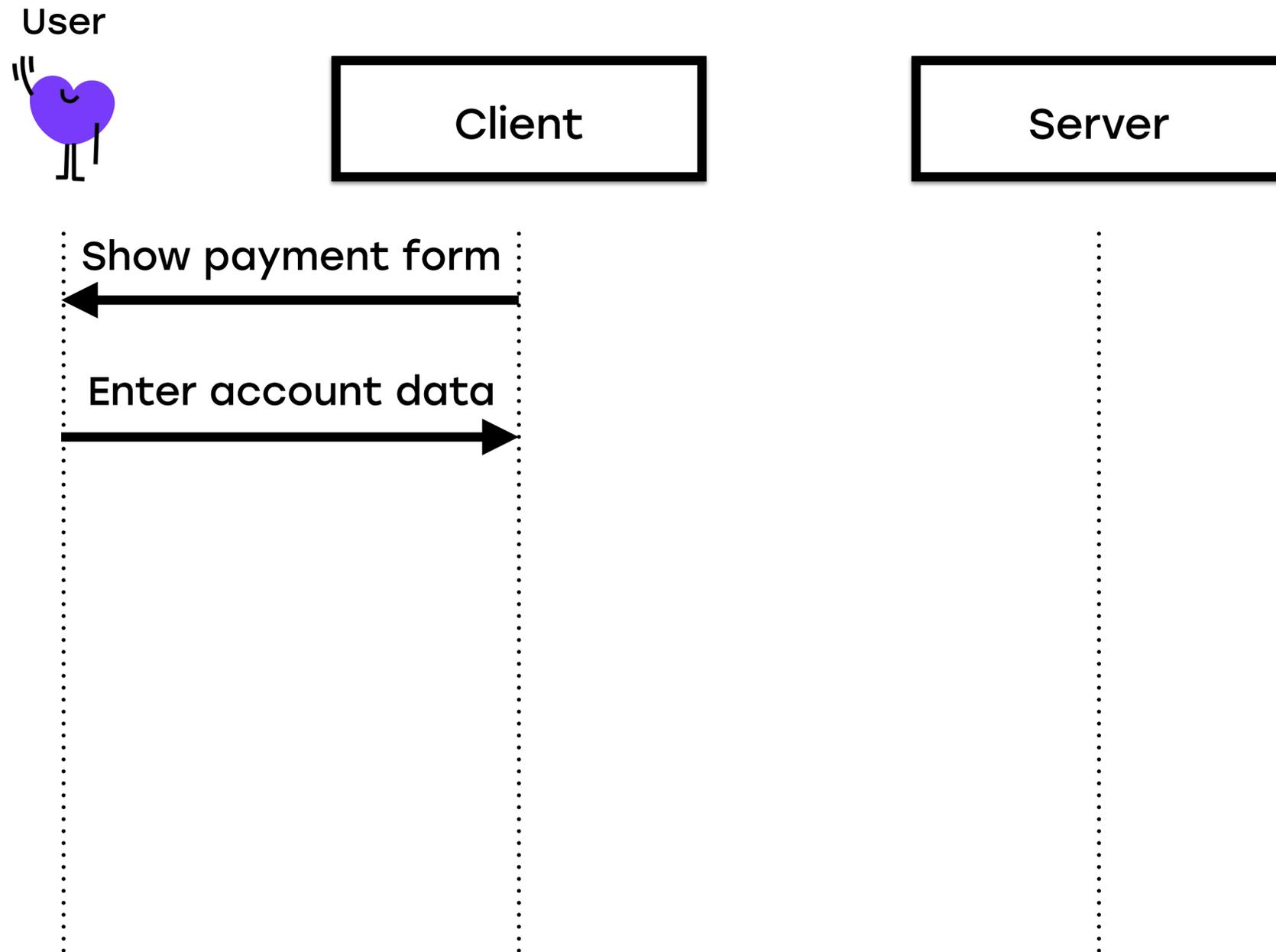
PayPal Процесс покупки



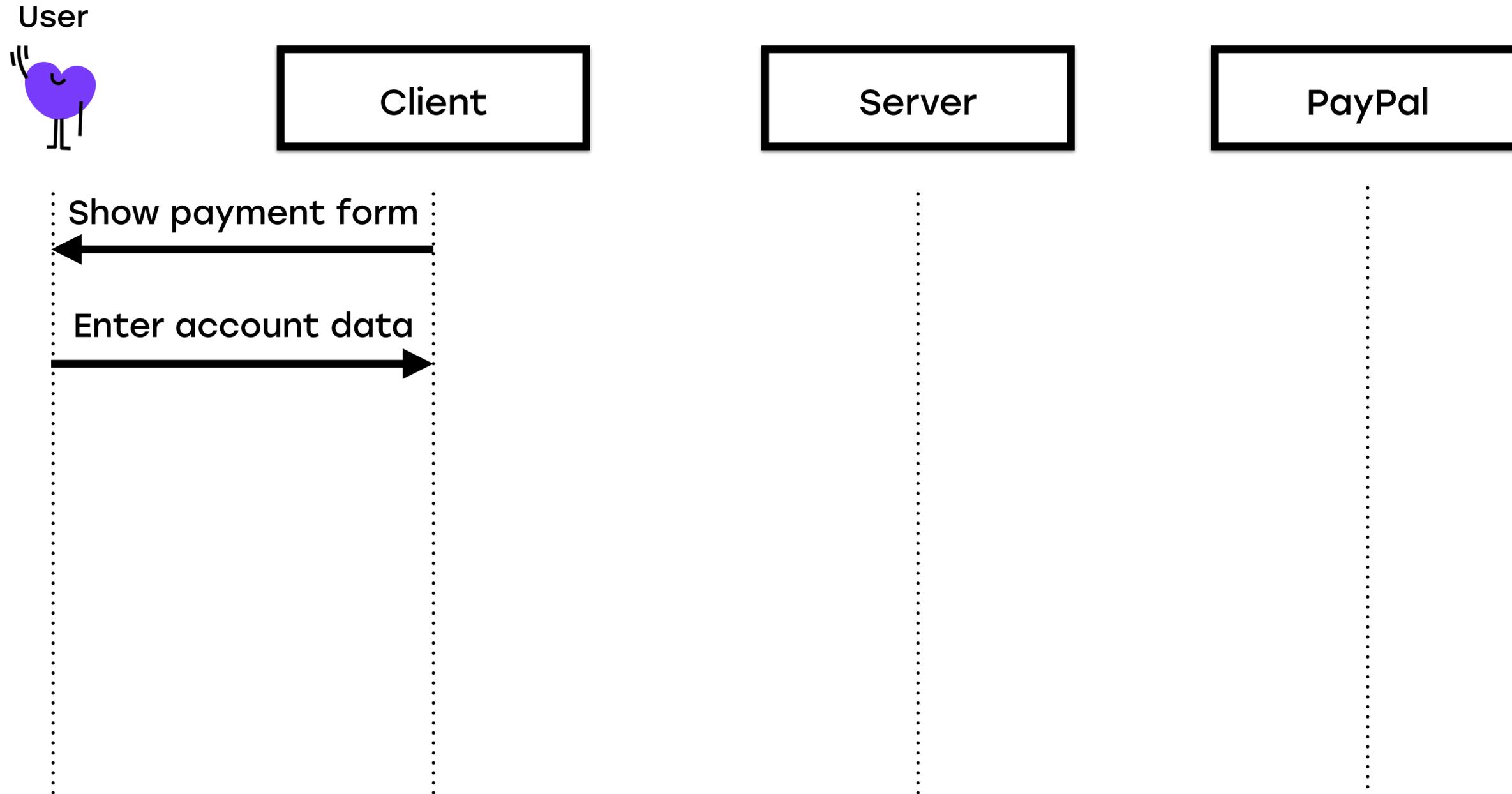
PayPal Процесс покупки



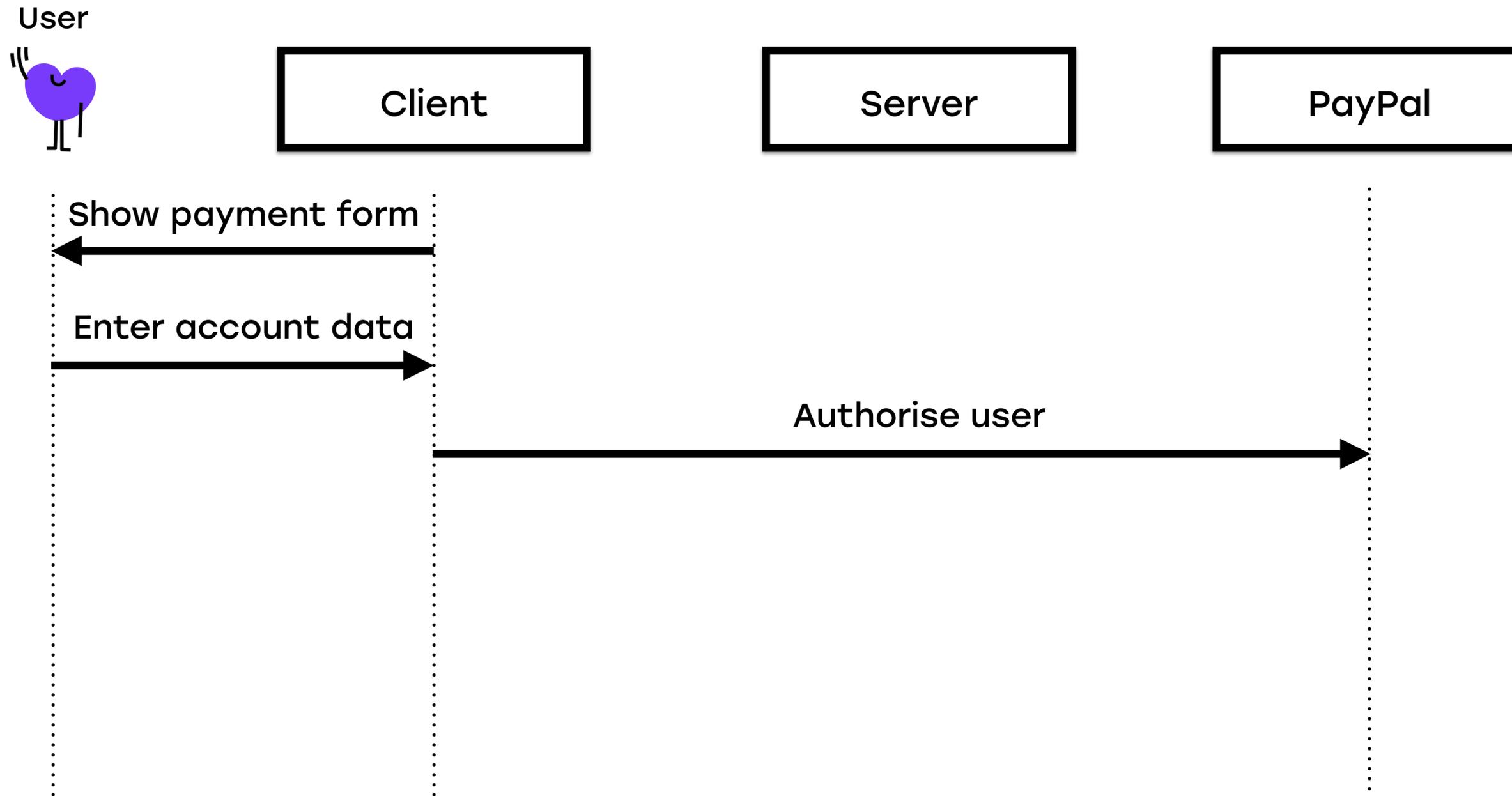
PayPal Процесс покупки



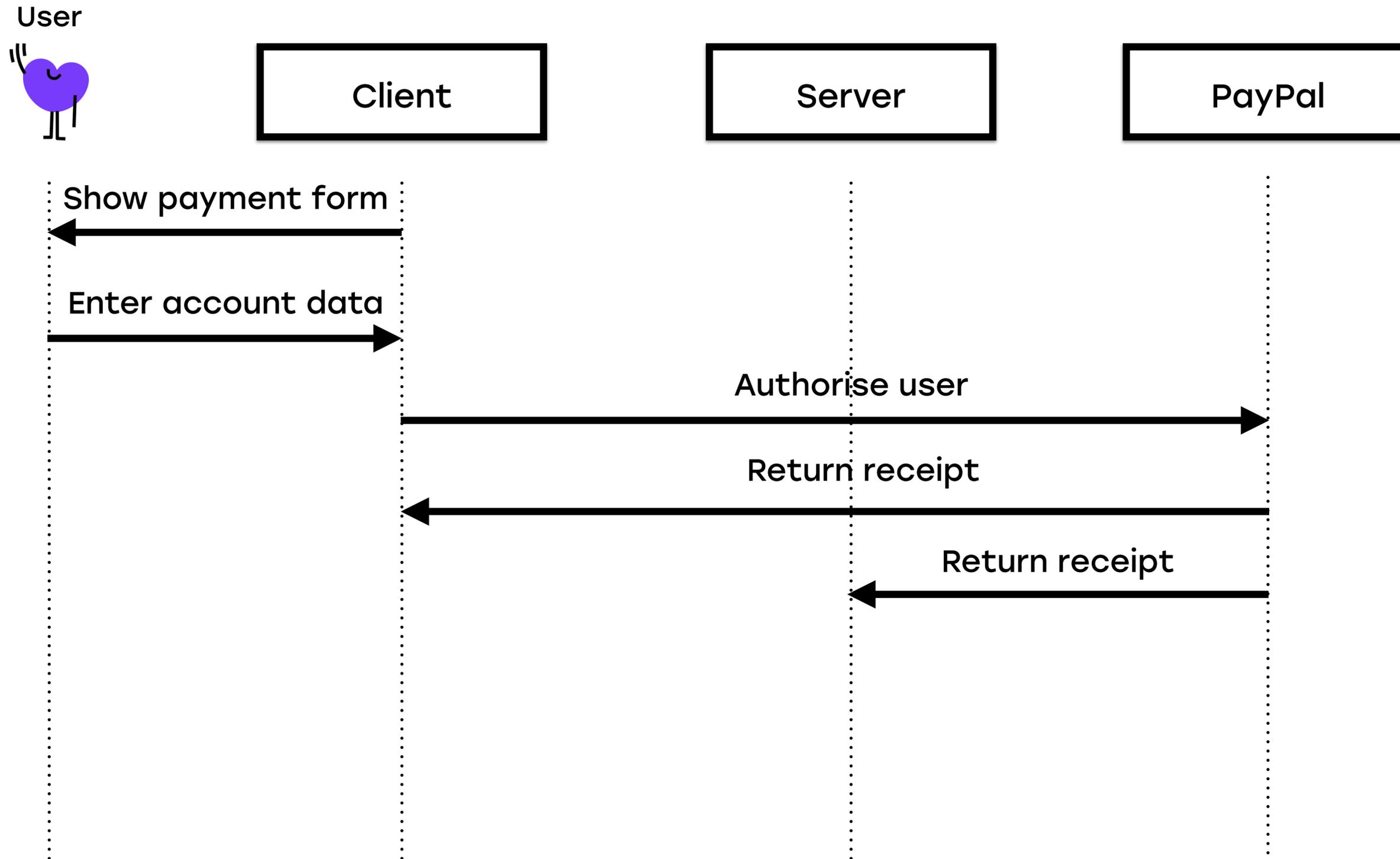
PayPal Процесс покупки



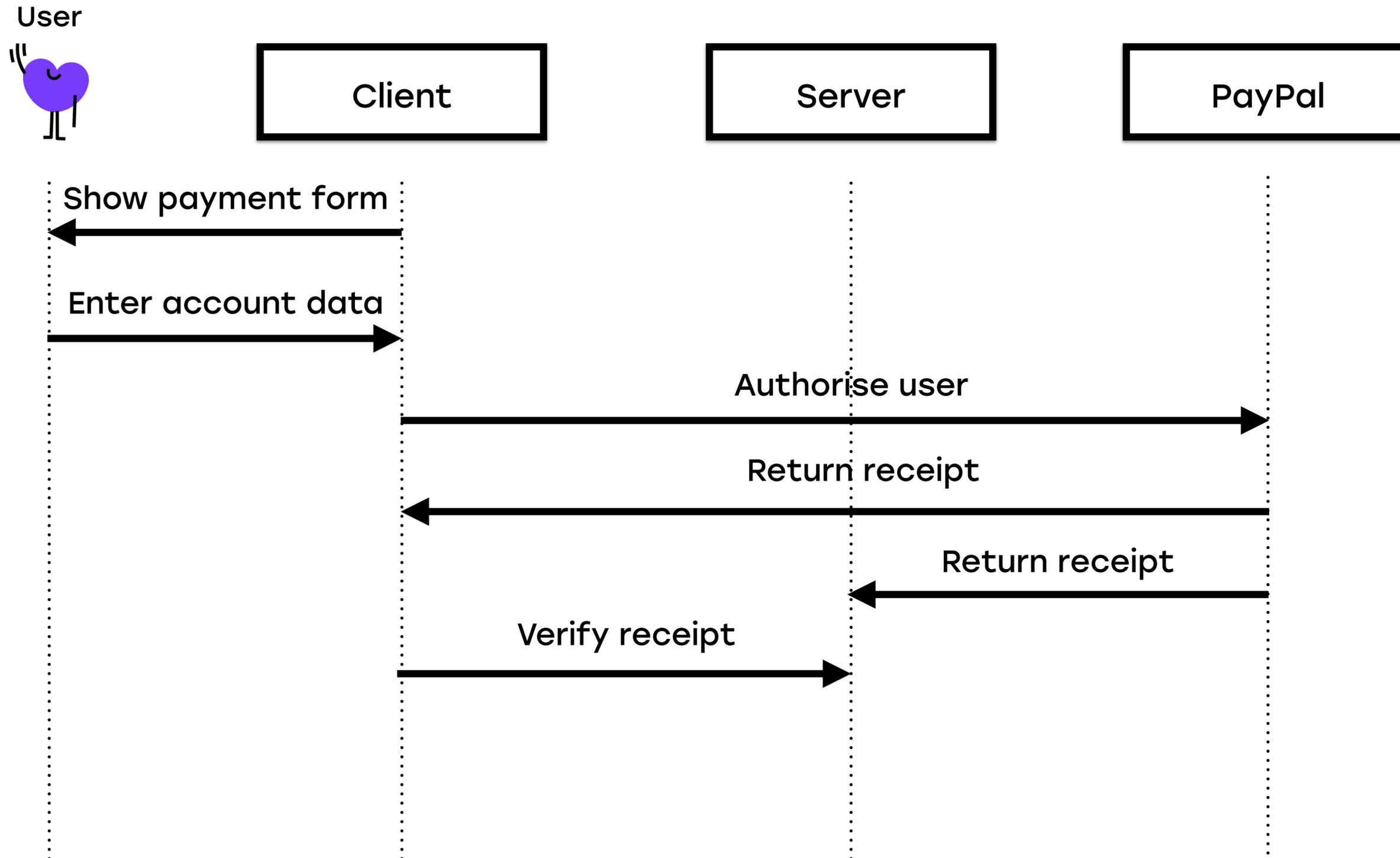
PayPal Процесс покупки



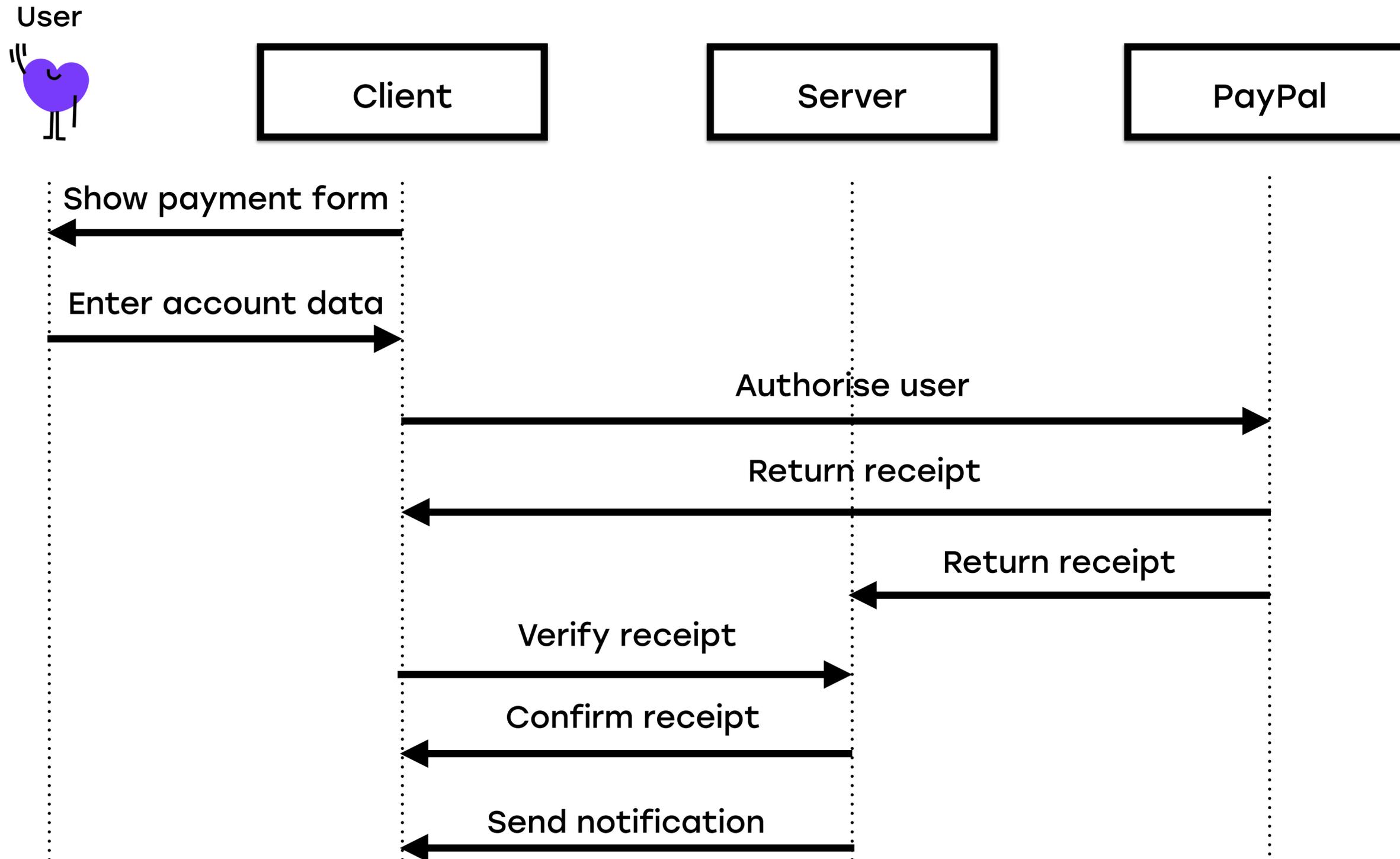
PayPal Процесс покупки



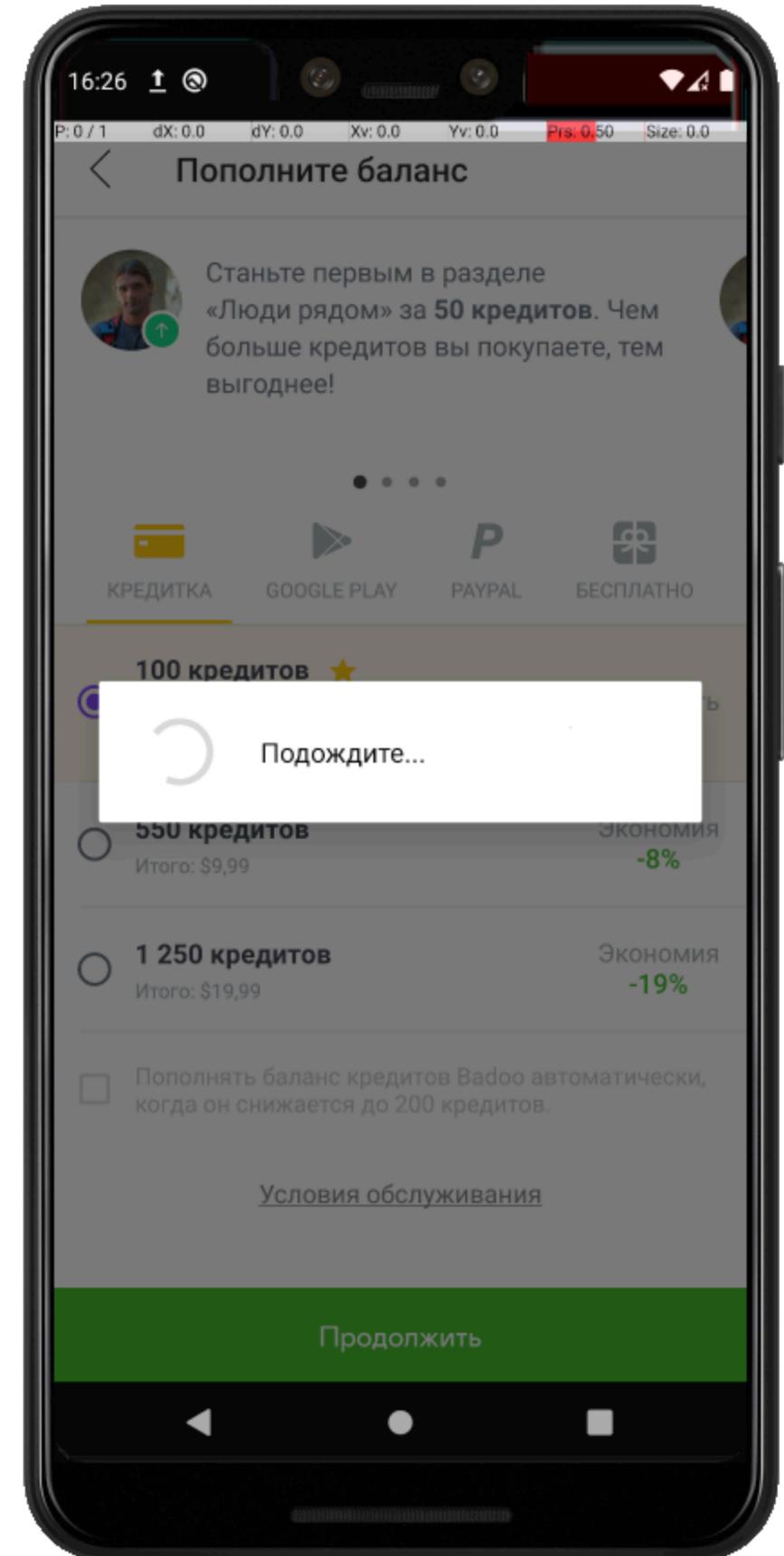
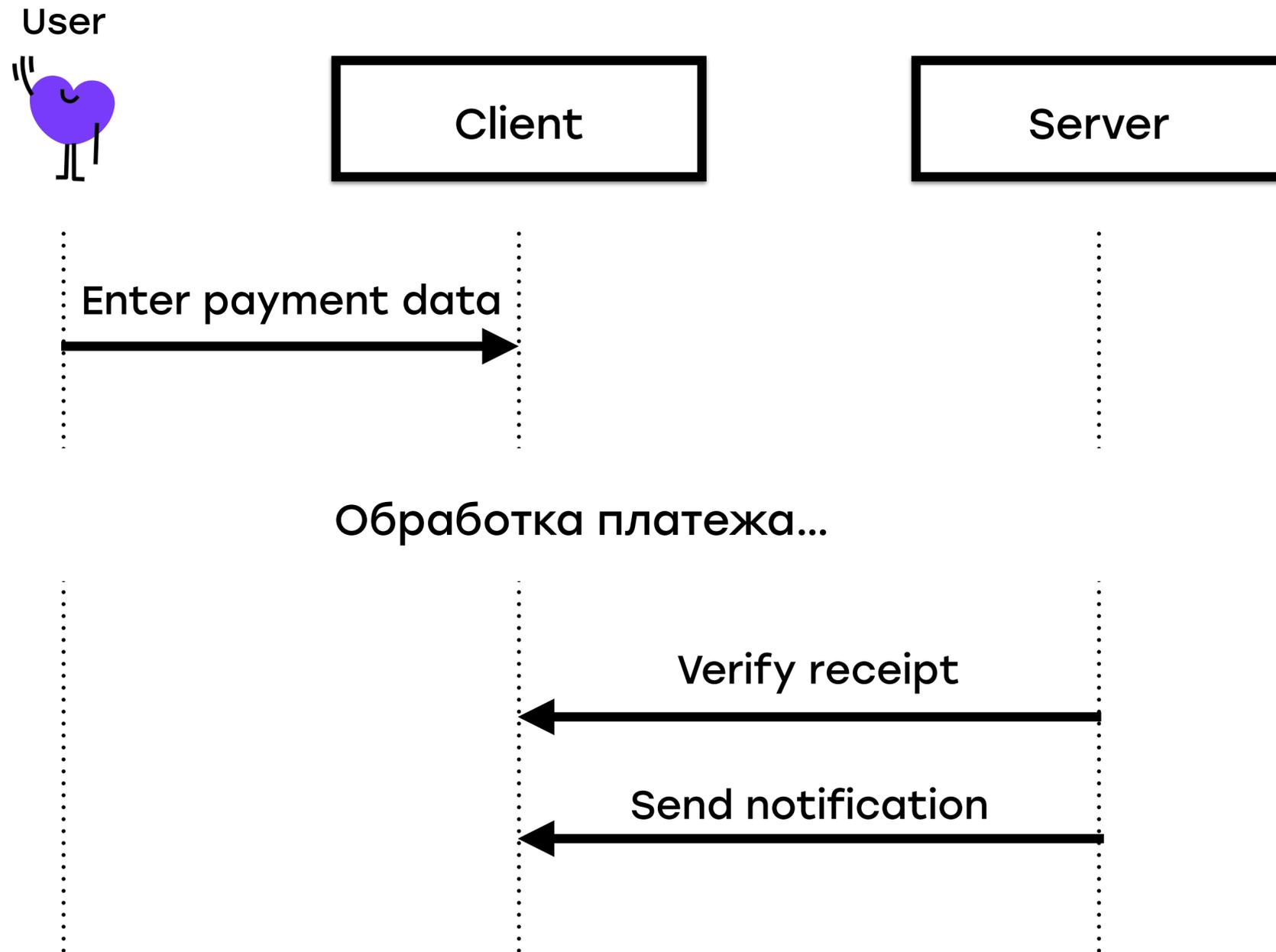
PayPal Процесс покупки



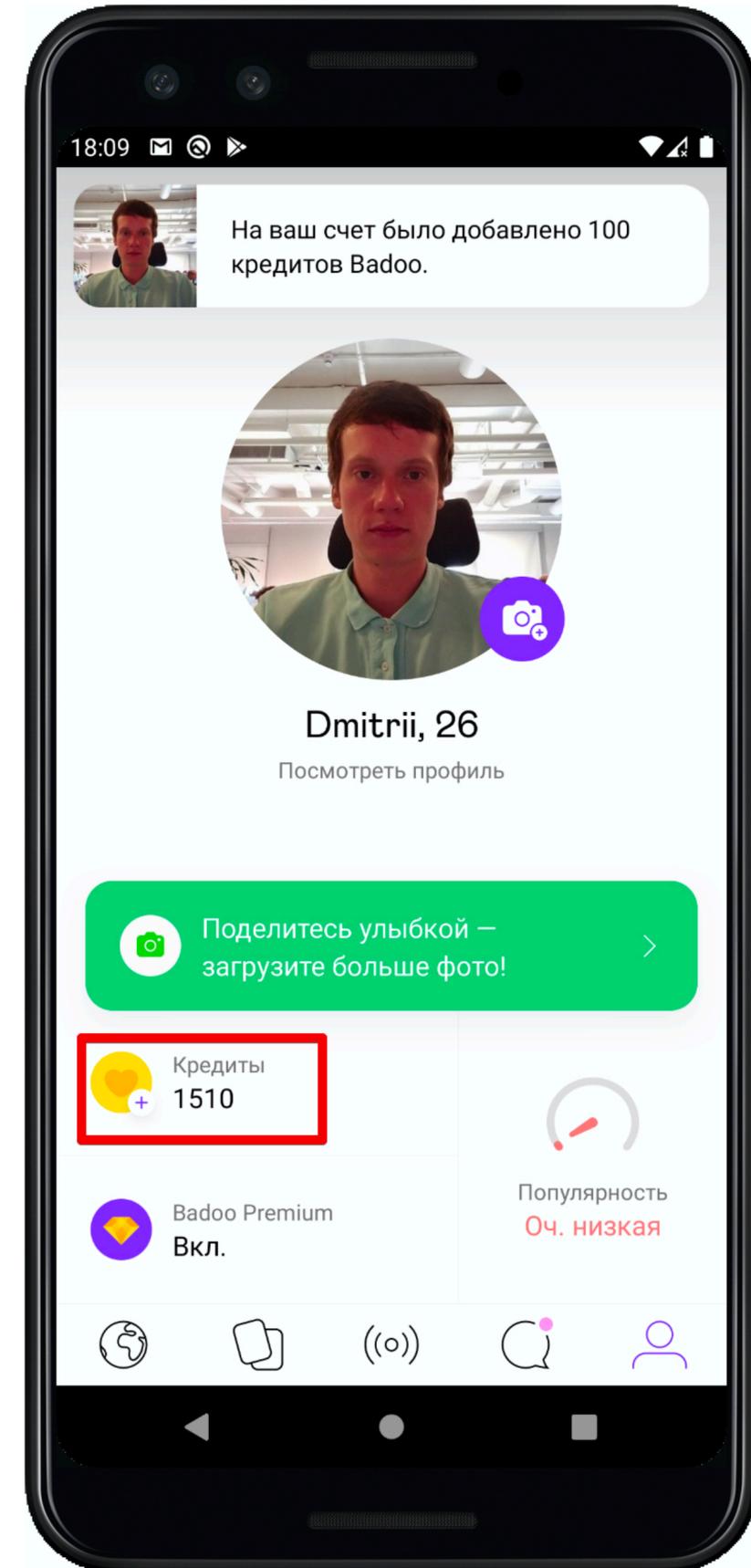
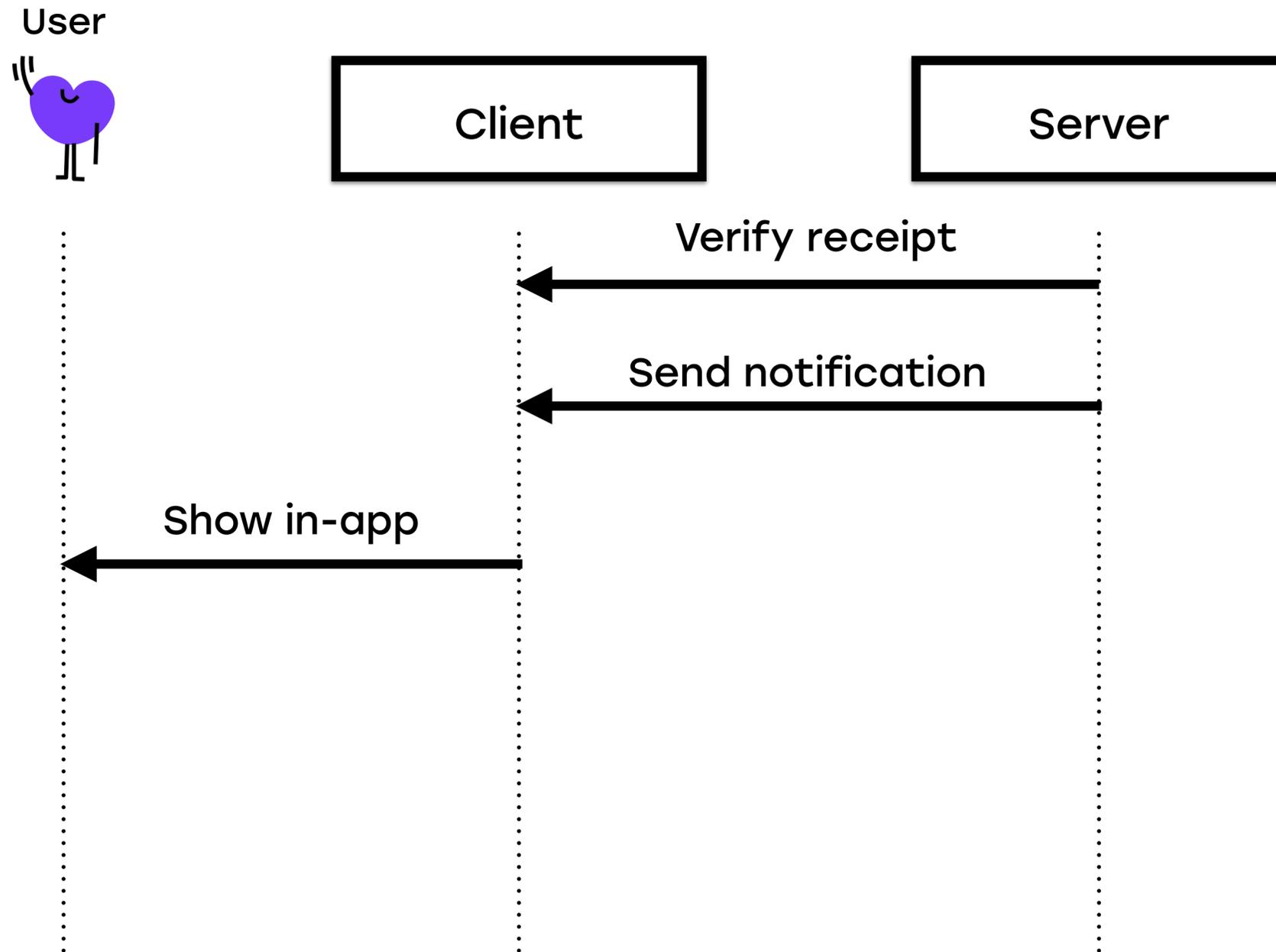
PayPal Процесс покупки



PayPal Процесс покупки



PayPal Процесс покупки



PayPal

Тестовое окружение

Методы тестирования

- Цель end-to-end тестов – проверка **интеграции** с провайдерами платежей

Методы тестирования

- Реальный платеж

Методы тестирования

- Реальный платеж
+ Проверка интеграции

Методы тестирования

- Реальный платеж
 - + Проверка интеграции
 - Дорого

Методы тестирования

- **Исключение внешних зависимостей**

Методы тестирования

- Исключение внешних зависимостей
- **Генерировать ответы от провайдера**

Методы тестирования

- Исключение внешних зависимостей
 - Генерировать ответы от провайдера
 - **Тестовый код, имитирующий работу провайдера**

Методы тестирования

- **Исключение внешних зависимостей**

Методы тестирования

- **Исключение внешних зависимостей**
+ Дешевле (НО нужна разработка)

Методы тестирования

- **Исключение внешних зависимостей**
 - + Дешевле (НО нужна разработка)
 - Нет обращений к самим провайдерам

Песочница (Sandbox)



Песочница (Sandbox)

- Тестовое окружение провайдера платежей

Песочница (Sandbox)

- Тестовое окружение провайдера платежей
- Позволяет провести платеж, используя “МОК” интеграции на стороне провайдера

Песочница (Sandbox)

- Тестовое окружение провайдера платежей
- Позволяет провести платеж, используя “МОК” интеграции на стороне провайдера
- **Ожидаемый ответ на полученные тестовые данные**

Песочница, подходящая для автоматизации

- Необходимые покупки (кредиты и премиум)

Песочница, подходящая для автоматизации

- Необходимые покупки (кредиты и премиум)
- **Негативные ответы на платежи**

Песочница, подходящая для автоматизации

- Необходимые покупки (кредиты и премиум)
- Негативные ответы на платежи
- **Есть документация**

Песочница, подходящая для автоматизации

- Необходимые покупки (кредиты и премиум)
- Негативные ответы на платежи
- Есть документация
- **Стабильная**

Почему песочницы

- Цель end-to-end тестов – проверка **интеграции** с различными методами

Почему песочницы

- Цель end-to-end тестов – проверка **интеграции** с различными методами
- + Процесс покупки максимально близок к реальному

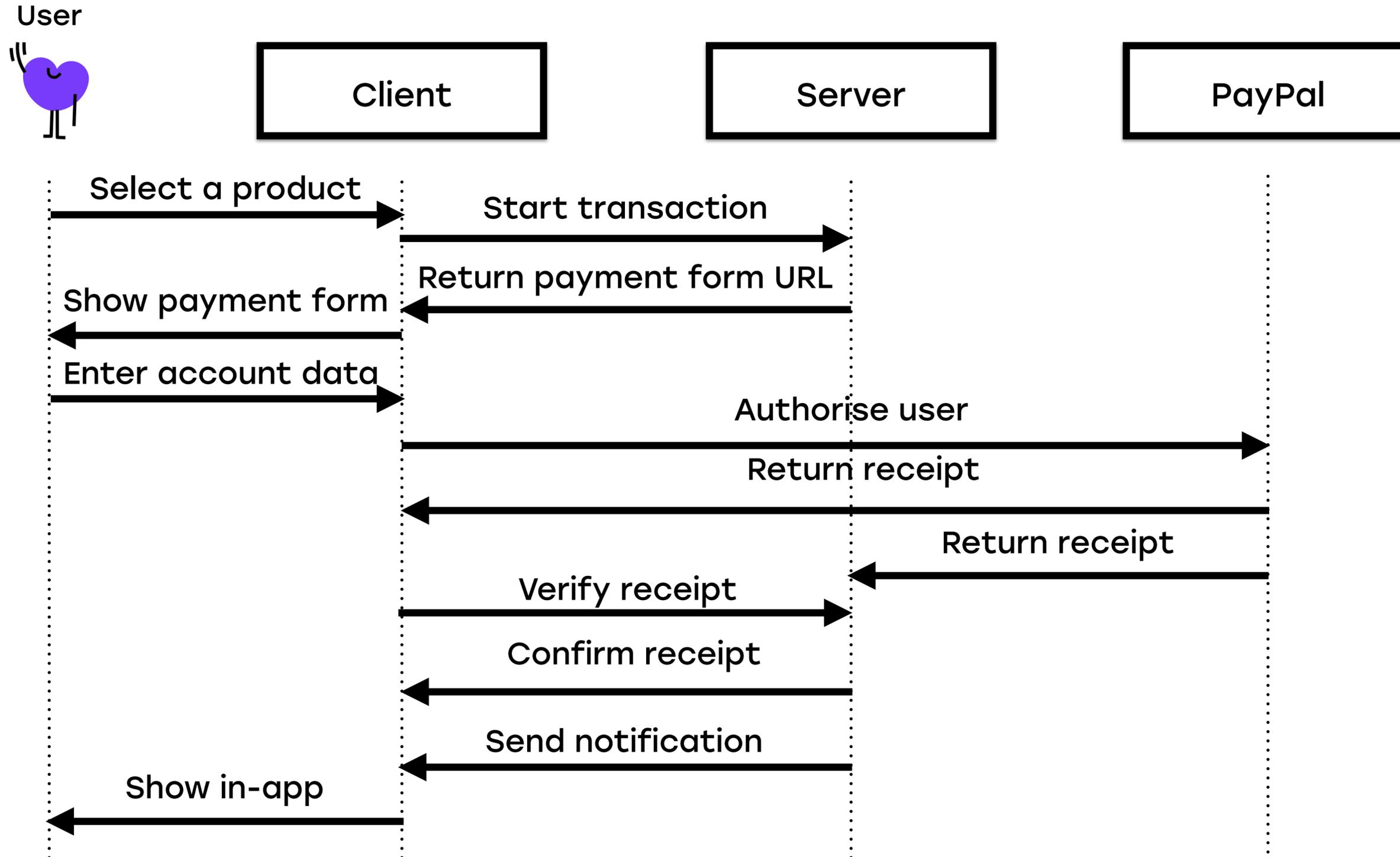
Почему песочницы

- Цель end-to-end тестов – проверка **интеграции** с различными методами
- + Процесс покупки максимально близок к реальному
- + Дешево

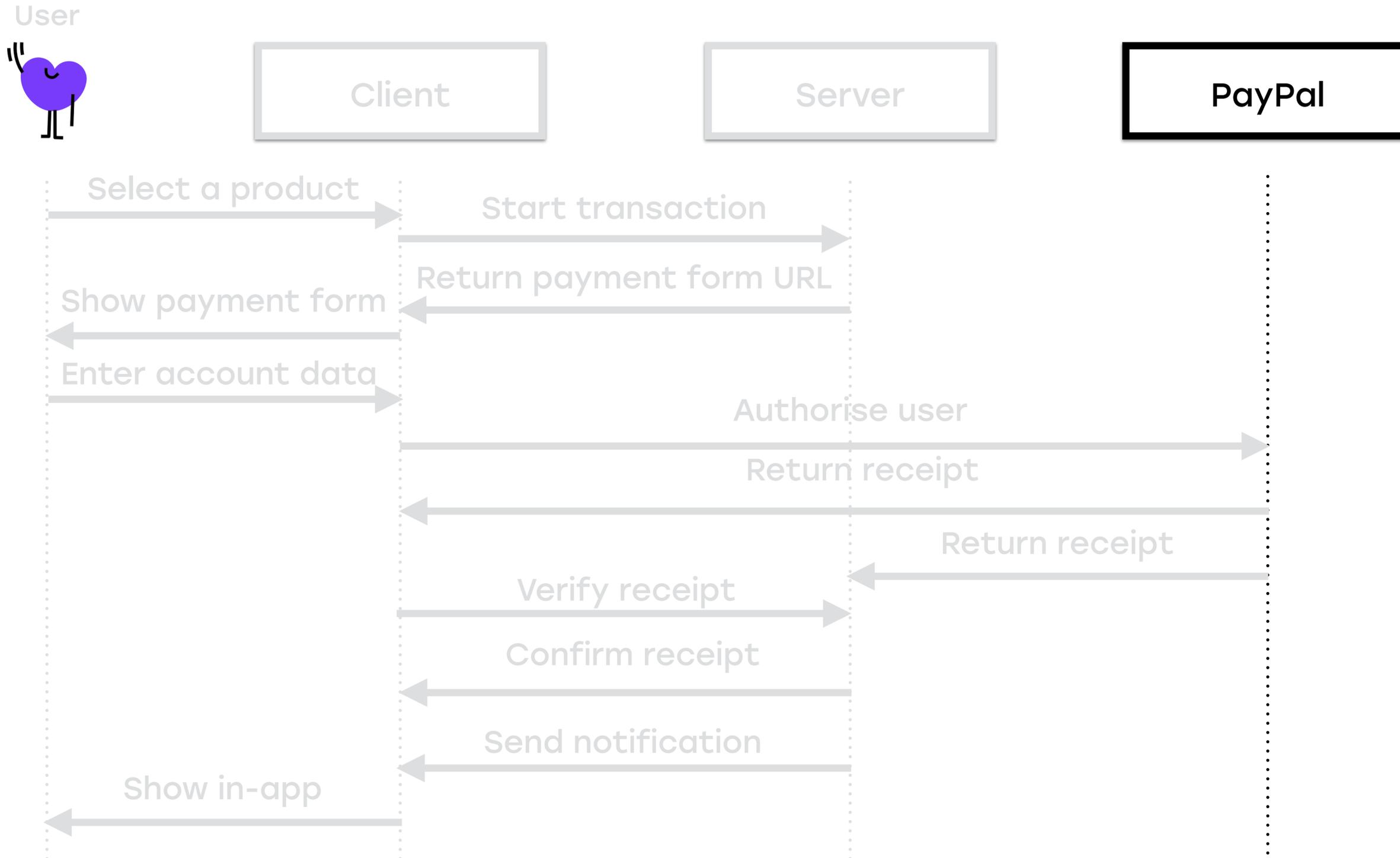
Почему песочницы

- Цель end-to-end тестов – проверка **интеграции** с разными методами
- + Процесс покупки максимально близок к реальному
- + Дешево
- Есть не у всех провайдеров

PayPal Тестовое окружение



PayPal Тестовое окружение

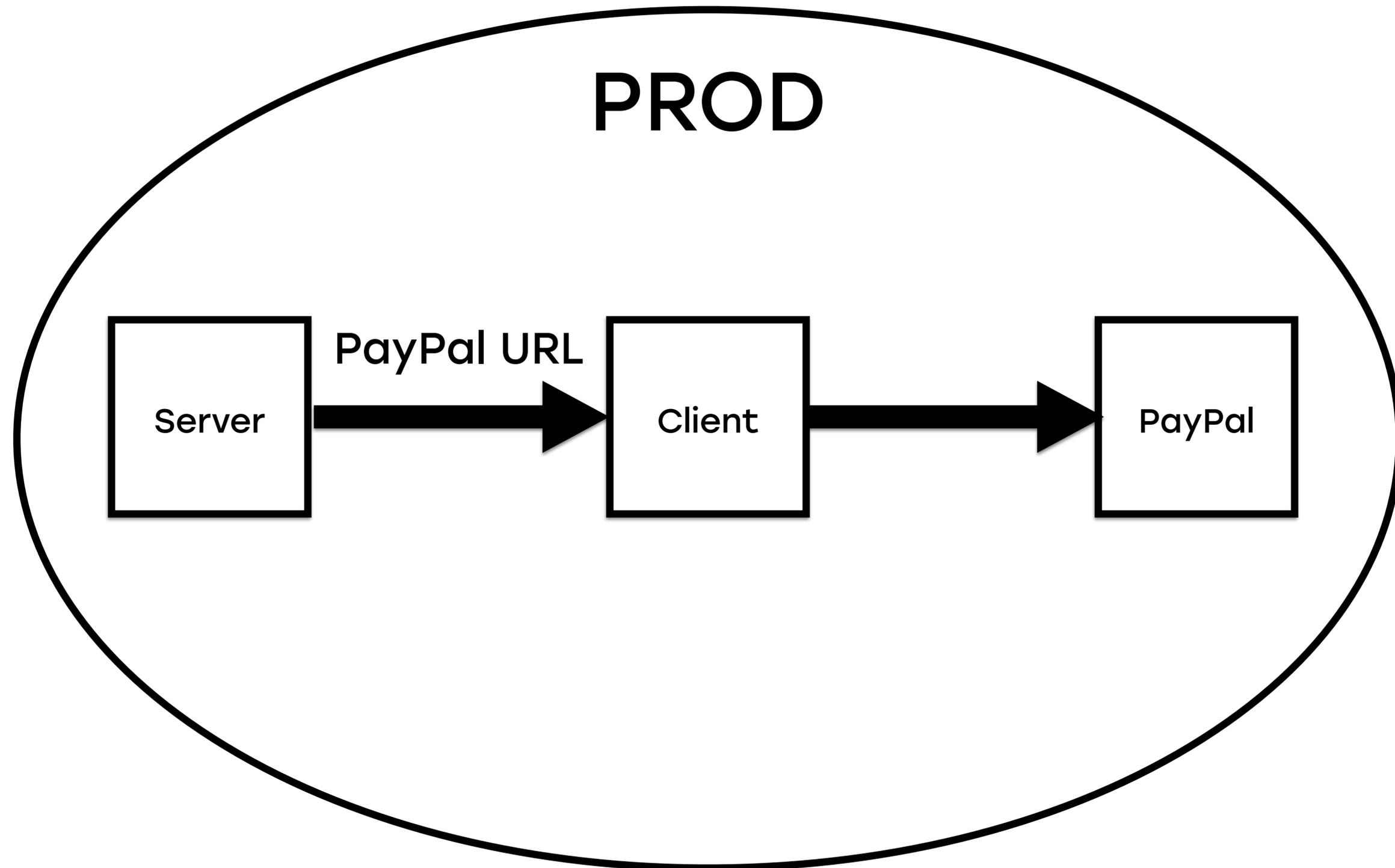


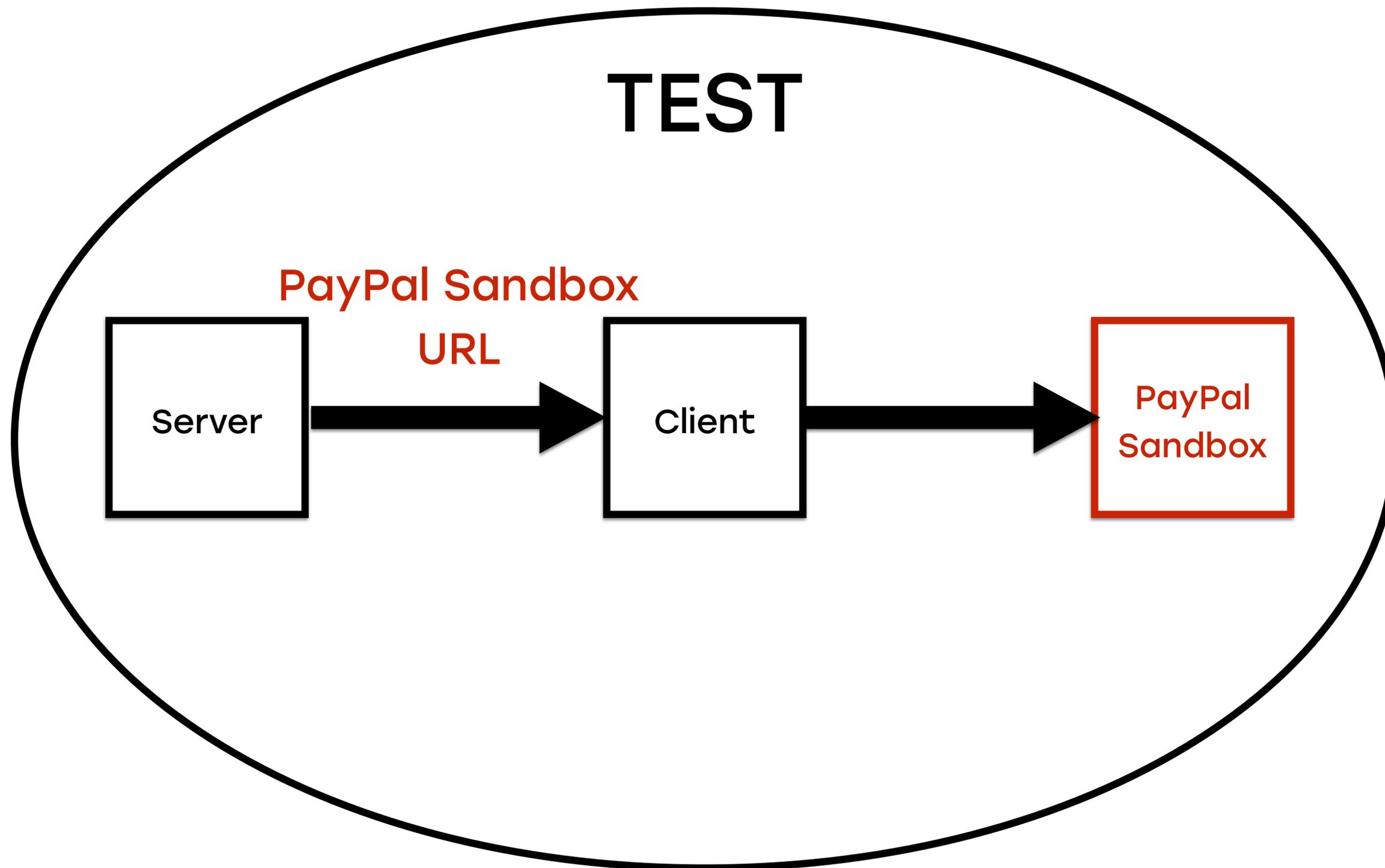


**Что надо
изменить?**

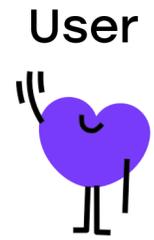


**Что надо
изменить?**





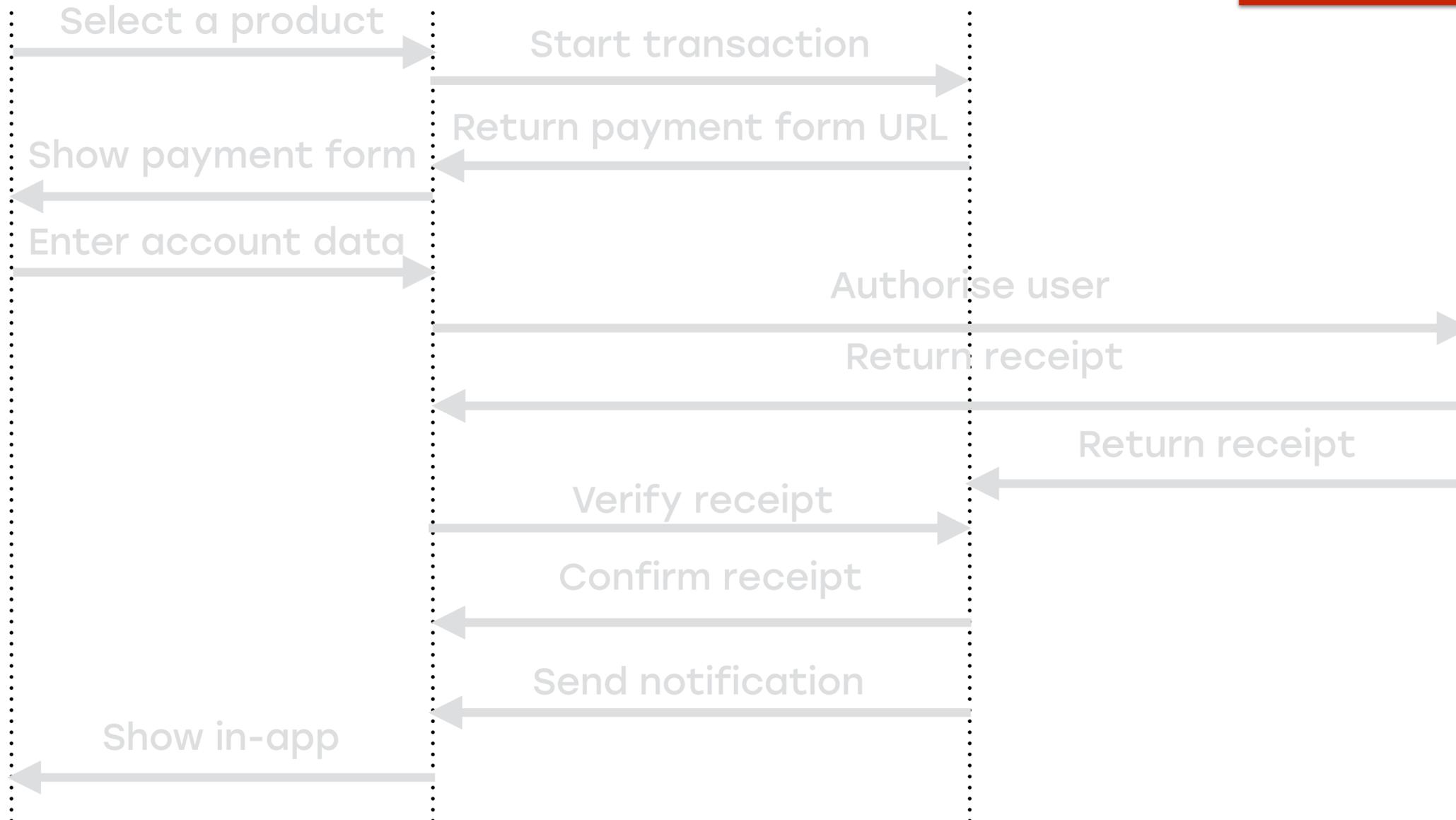
PayPal Тестовое окружение



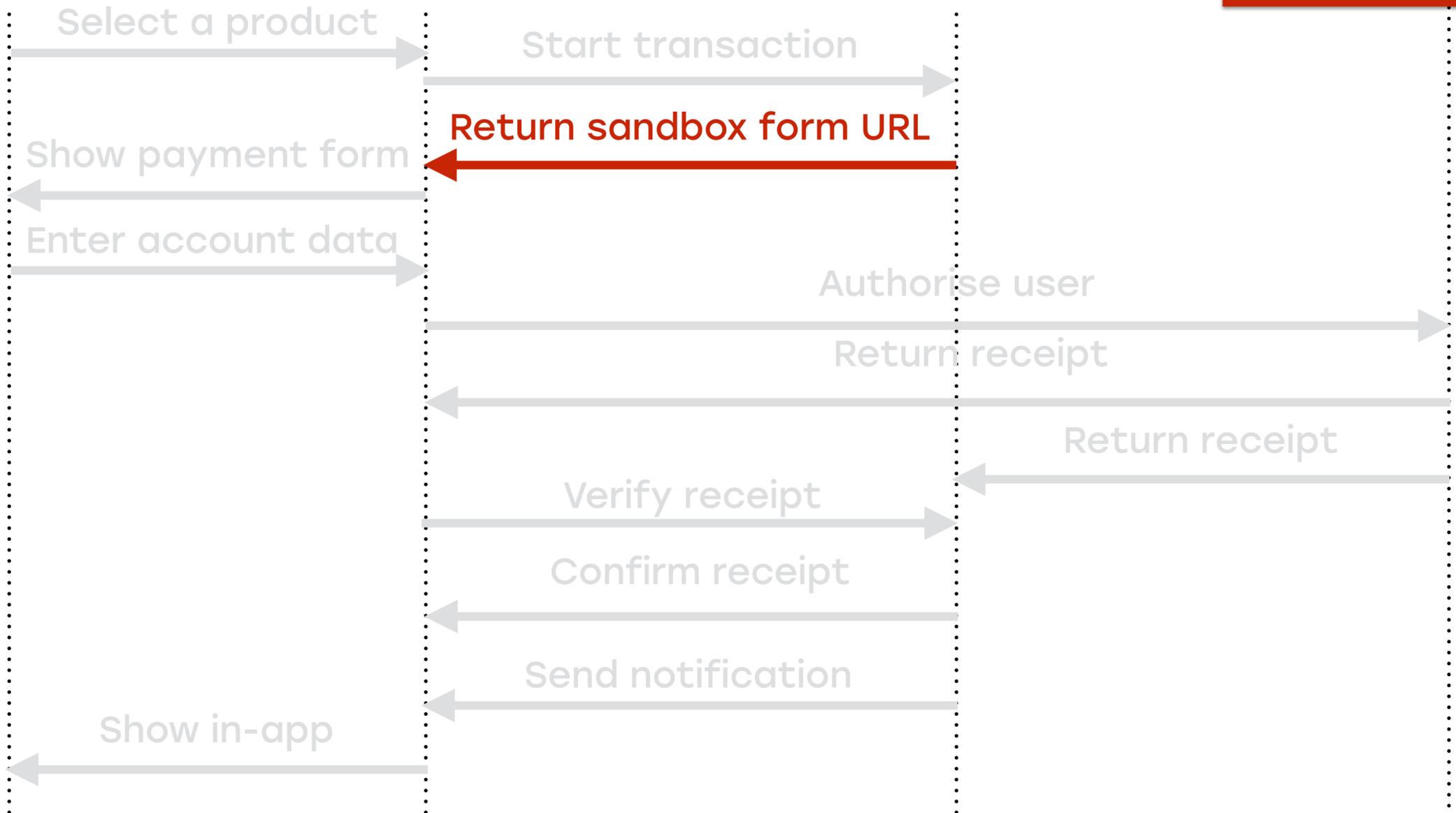
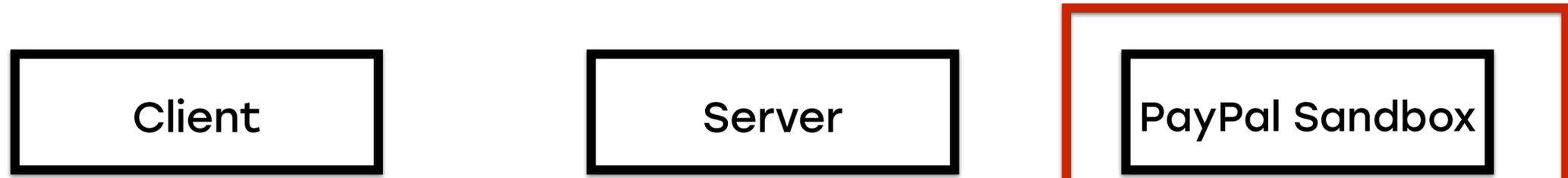
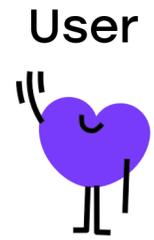
Client

Server

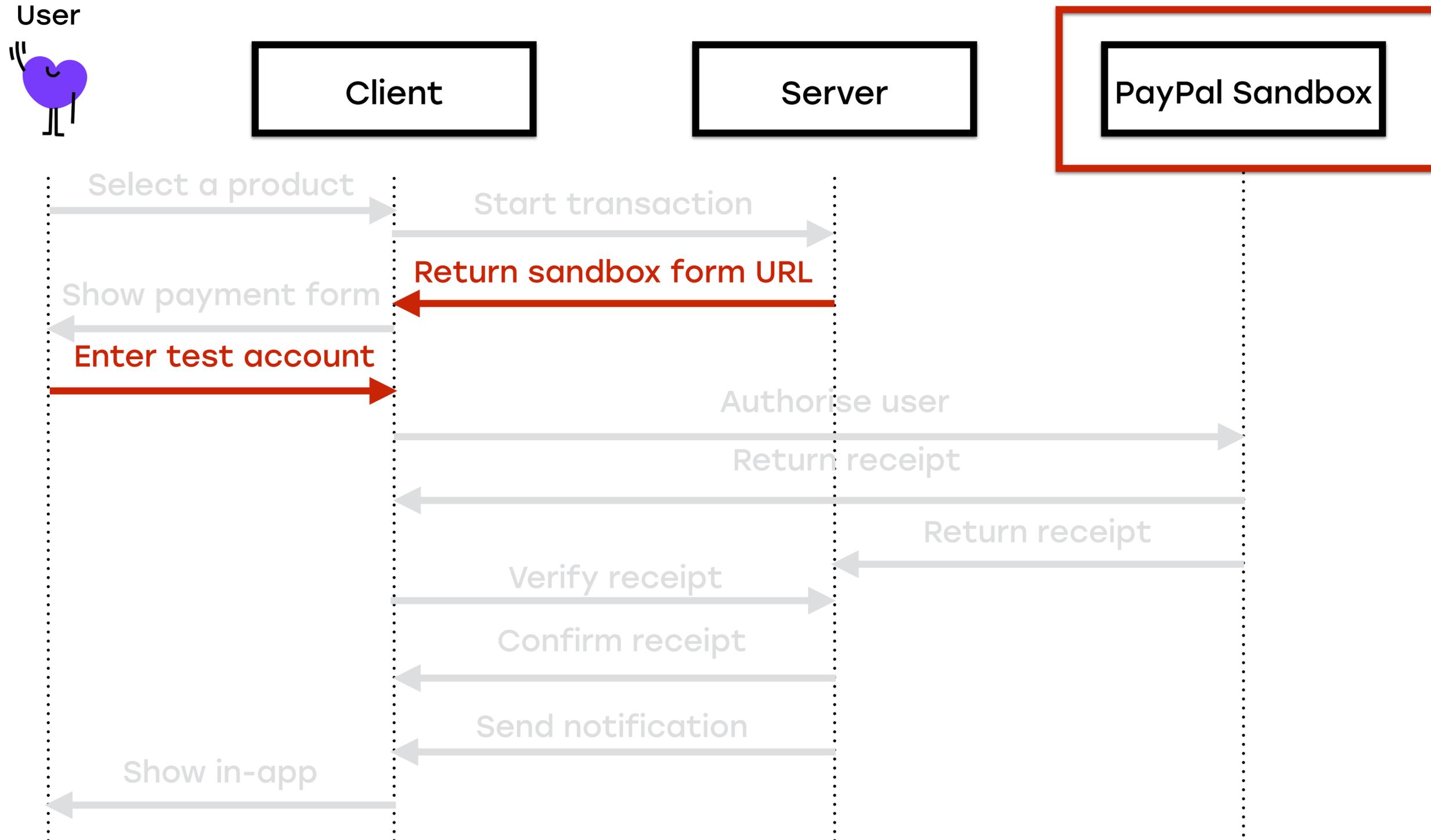
PayPal Sandbox



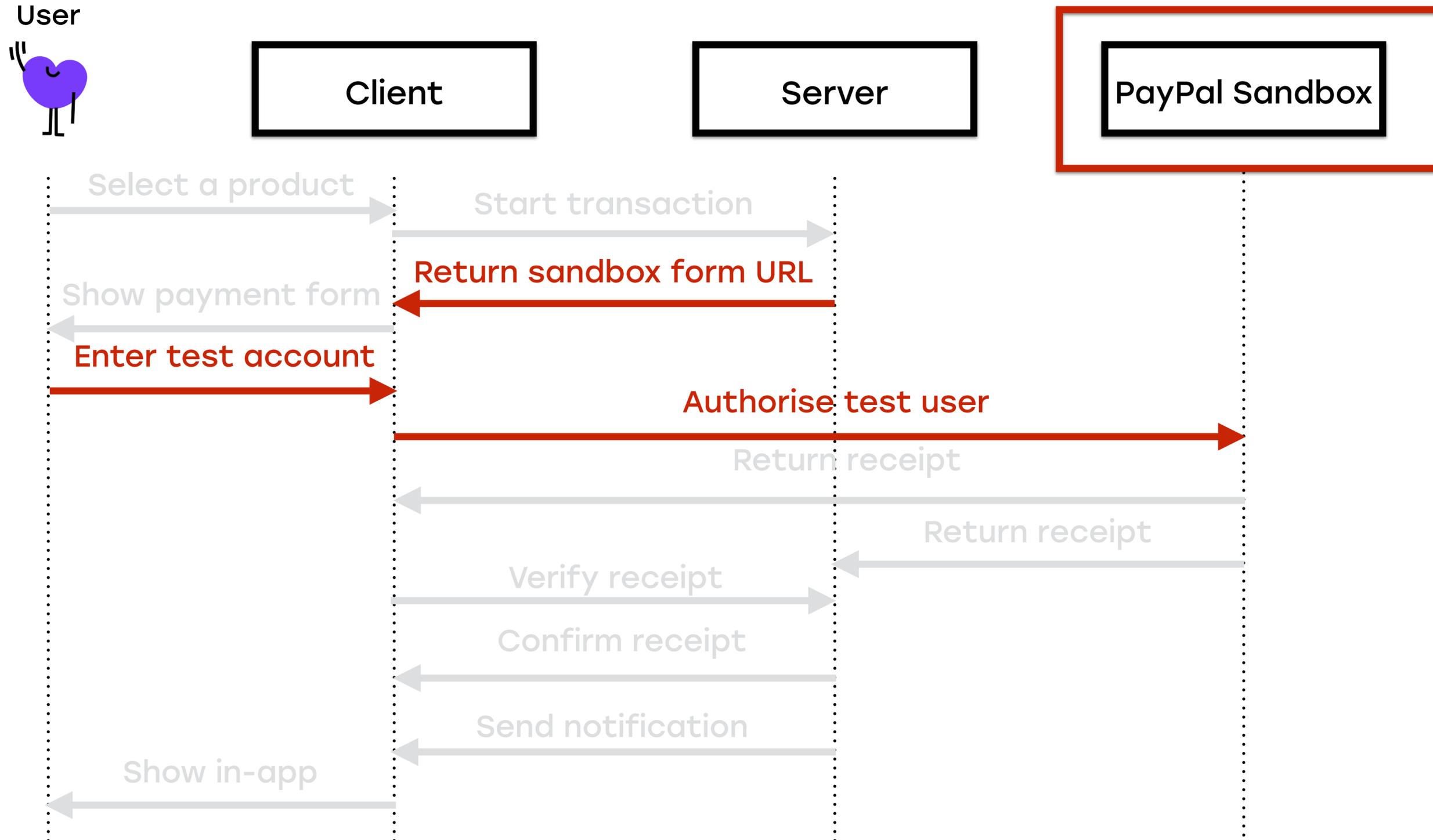
PayPal Тестовое окружение



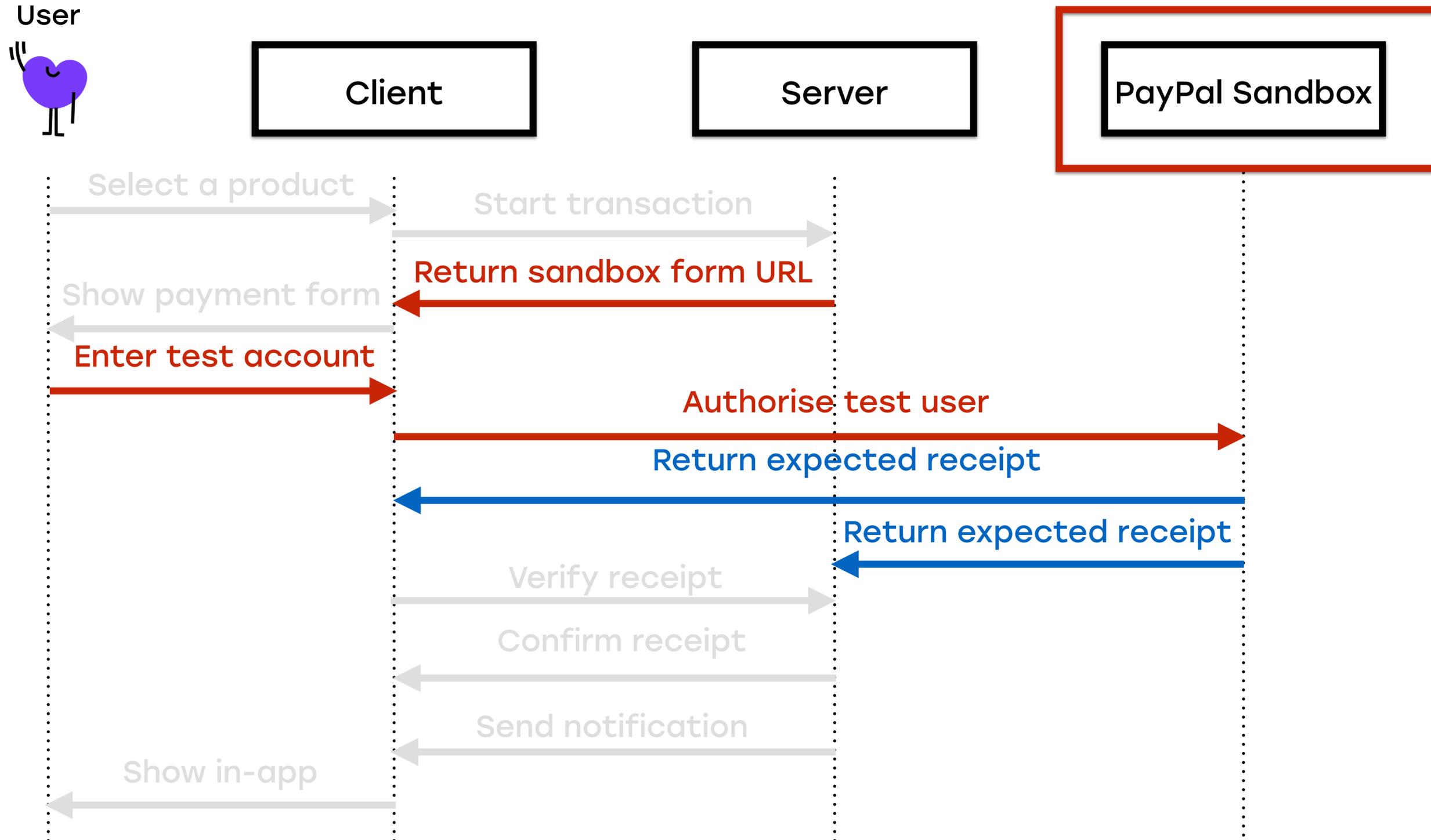
PayPal Тестовое окружение



PayPal Тестовое окружение



PayPal Тестовое окружение

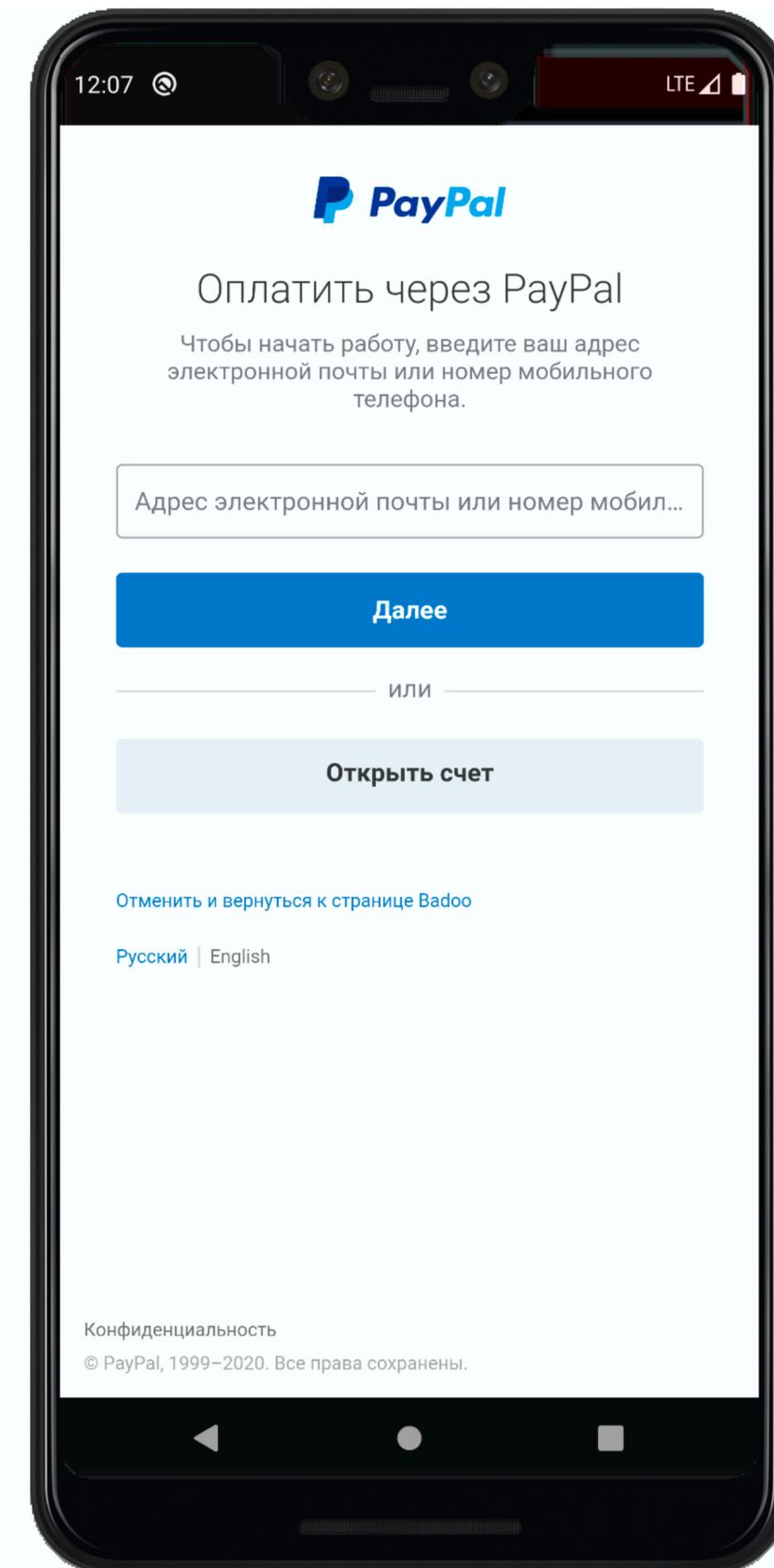


PayPal

АВТОМАТИЗАЦИЯ

Экран оплаты

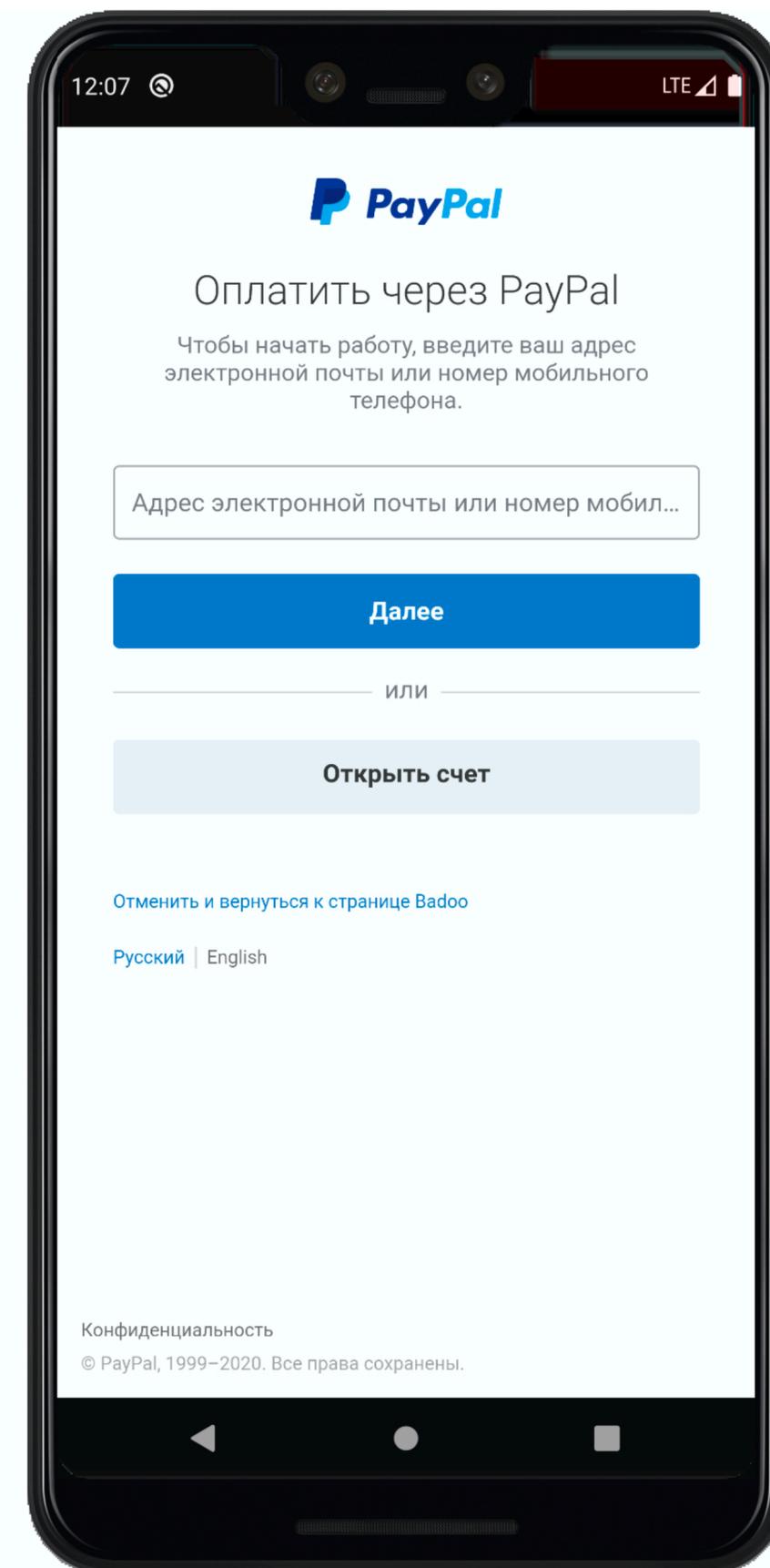
- **WebView**



Экран оплаты

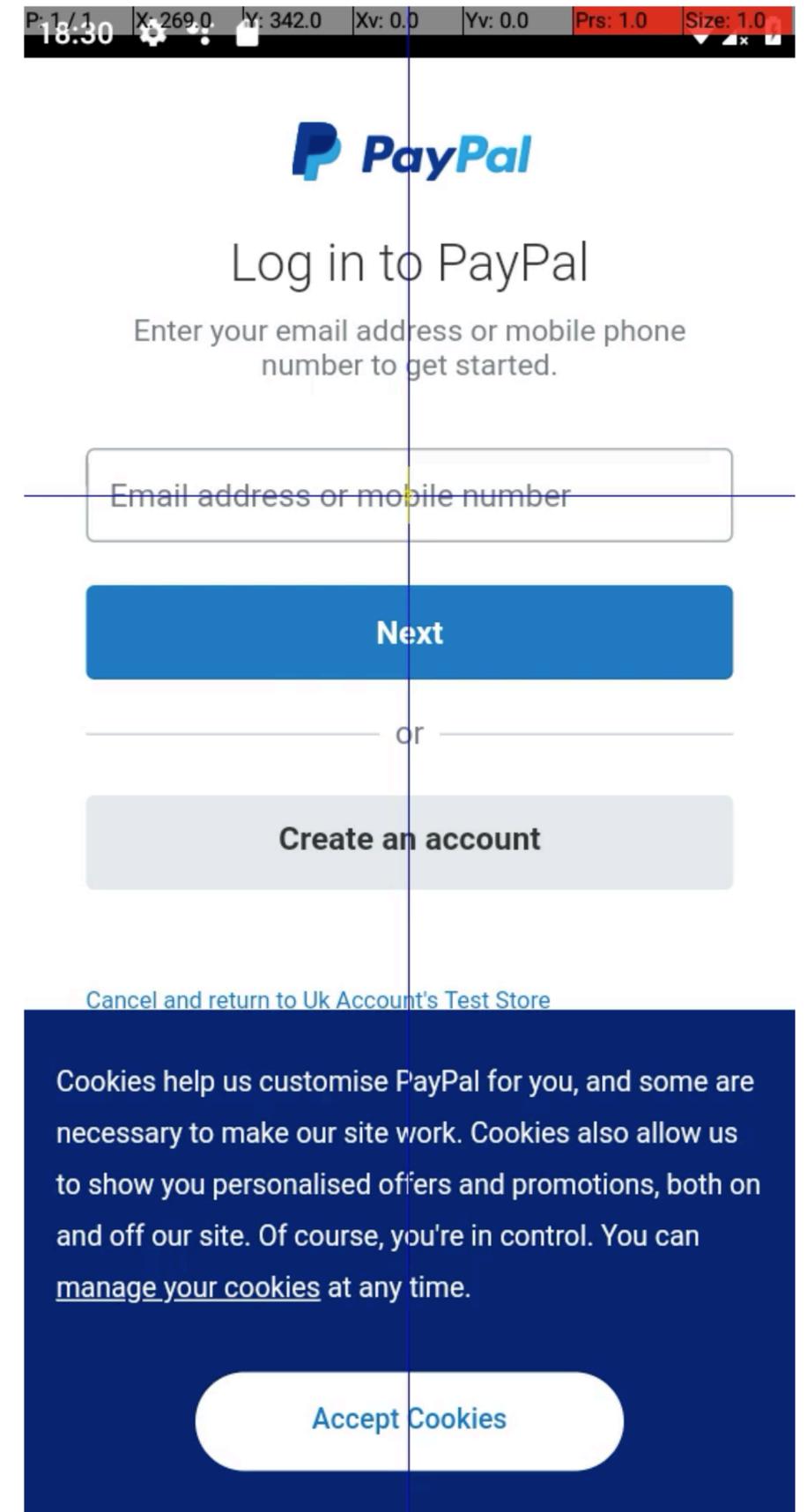
- Есть доступ к элементам

PAYPAL_LOGO = "webview css: '.paypal-logo'"
EMAIL_BOX = "webview css: '#email'"
PASSWORD_BOX = "webview css: '#password'"
NEXT_BUTTON = "webview css: 'button#btnNext'"
LOGIN_BUTTON = "webview css: '#btnLogin'"
CONFIRM = "webview css: '#confirmButtonTop'"



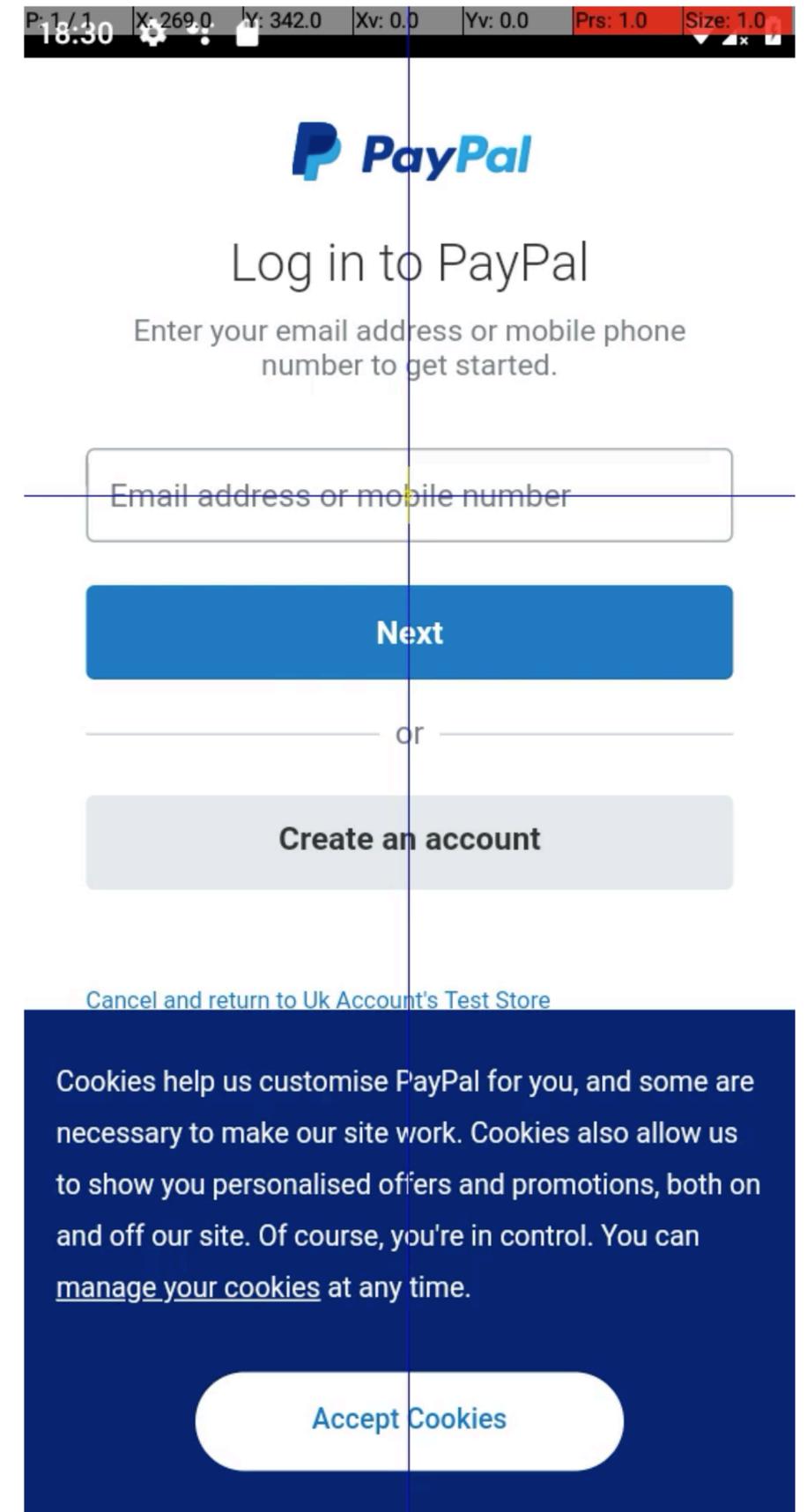
Платеж

```
def submit_payment(payment_details)
  wait_for_elements_displayed(EMAIL_BOX, timeout: Wait::A_MINUTE, timeout_message)
  set_text_in_field(EMAIL_BOX, payment_details[:email])
  unless elements_displayed?(PASSWORD_BOX)
    tap_element(NEXT_BUTTON)
    hide_keyboard
    wait_for_elements_displayed(PASSWORD_BOX, timeout: Wait::A_MINUTE, timeout_message)
  end
  set_text_in_field(PASSWORD_BOX, payment_details[:password])
  hide_keyboard
  tap_element(LOGIN_BUTTON)
  wait_for_elements_displayed(CONFIRM, timeout: Wait::A_MINUTE, timeout_message)
  tap_element(CONFIRM)
end
```



Платеж

```
def submit_payment(payment_details)
  wait_for_elements_displayed(EMAIL_BOX, timeout: Wait::A_MINUTE, timeout_message)
  set_text_in_field(EMAIL_BOX, payment_details[:email])
  unless elements_displayed?(PASSWORD_BOX)
    tap_element(NEXT_BUTTON)
    hide_keyboard
    wait_for_elements_displayed(PASSWORD_BOX, timeout: Wait::A_MINUTE, timeout_message)
  end
  set_text_in_field(PASSWORD_BOX, payment_details[:password])
  hide_keyboard
  tap_element(LOGIN_BUTTON)
  wait_for_elements_displayed(CONFIRM, timeout: Wait::A_MINUTE, timeout_message)
  tap_element(CONFIRM)
end
```



Тестовые данные

- Тестовые аккаунты для песочницы

Дополнительные сценарии

- Сохранение платежных деталей

Дополнительные сценарии

- Сохранение платежных деталей
- Не надо повторно проходить авторизацию

Проблема стабильности

Проблема стабильности

- 8 тестов на Android

Проблема стабильности

- 8 тестов на Android
- **27 тестов на MW**

Проблема стабильности

- 8 тестов на Android
- 27 тестов на MW
- **2 серверных релиза + текущие задачи**

Проблема стабильности

**Error 429 – Too
many requests**

Проблема стабильности

- Анализ упавших тестов занимает время

Проблема стабильности

- Анализ упавших тестов занимает время
- **Перезапуск упавших тестов занимает время**

Проблема стабильности

- Анализ упавших тестов занимает время
- Перезапуск упавших тестов занимает время
- **Потенциальная задержка серверного релиза**

Проблема стабильности. Причина

- Ограничение количества запросов к песочнице

Проблема стабильности. Причина

- Ограничение количества запросов к песочнице
- **50 запросов в минуту**

Проблема стабильности. Причина

- Ограничение количества запросов к песочнице
- 50 запросов в минуту
- **Блокировка IP адреса на 5 минут**

Проблема стабильности. Решение

- Перенастроили PayPal конфигурации

Проблема стабильности. Решение

- Перенастроили PayPal конфигурации
- Ограничили параллельный запуск билдов (<https://www.jetbrains.com/help/teamcity/shared-resources.html>)

Проблема стабильности. Решение

- Перенастроили PayPal конфигурации
- Ограничили параллельный запуск билдов (<https://www.jetbrains.com/help/teamcity/shared-resources.html>)
- **2 потока тестов VS 30-140 потоков в других конфигурациях**

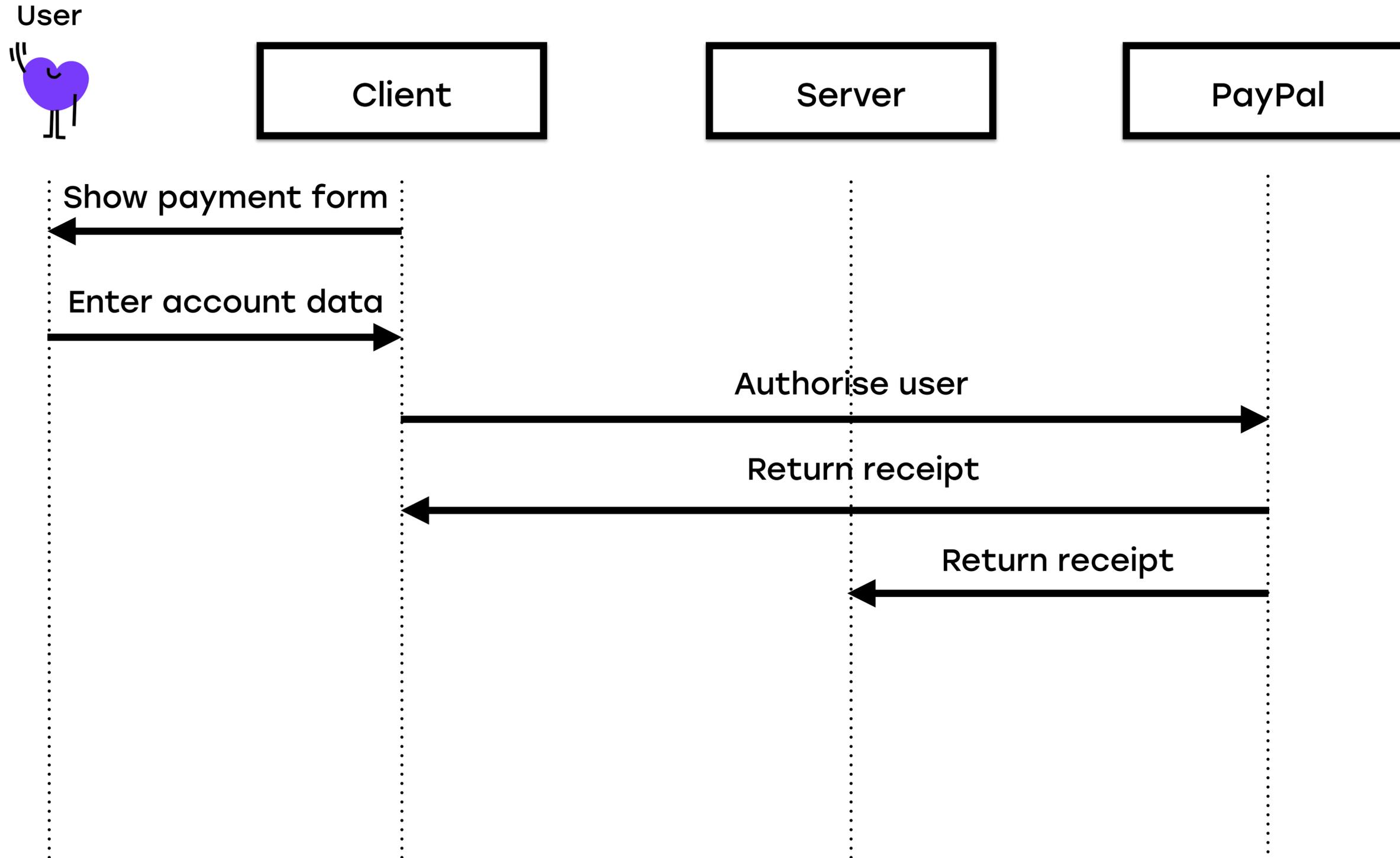
Тестовые конфигурации

- 1 метод - 1 конфигурация
 - + Локализация падений
 - + Гибкость настройки
 - Дополнительное время на подготовку запуска

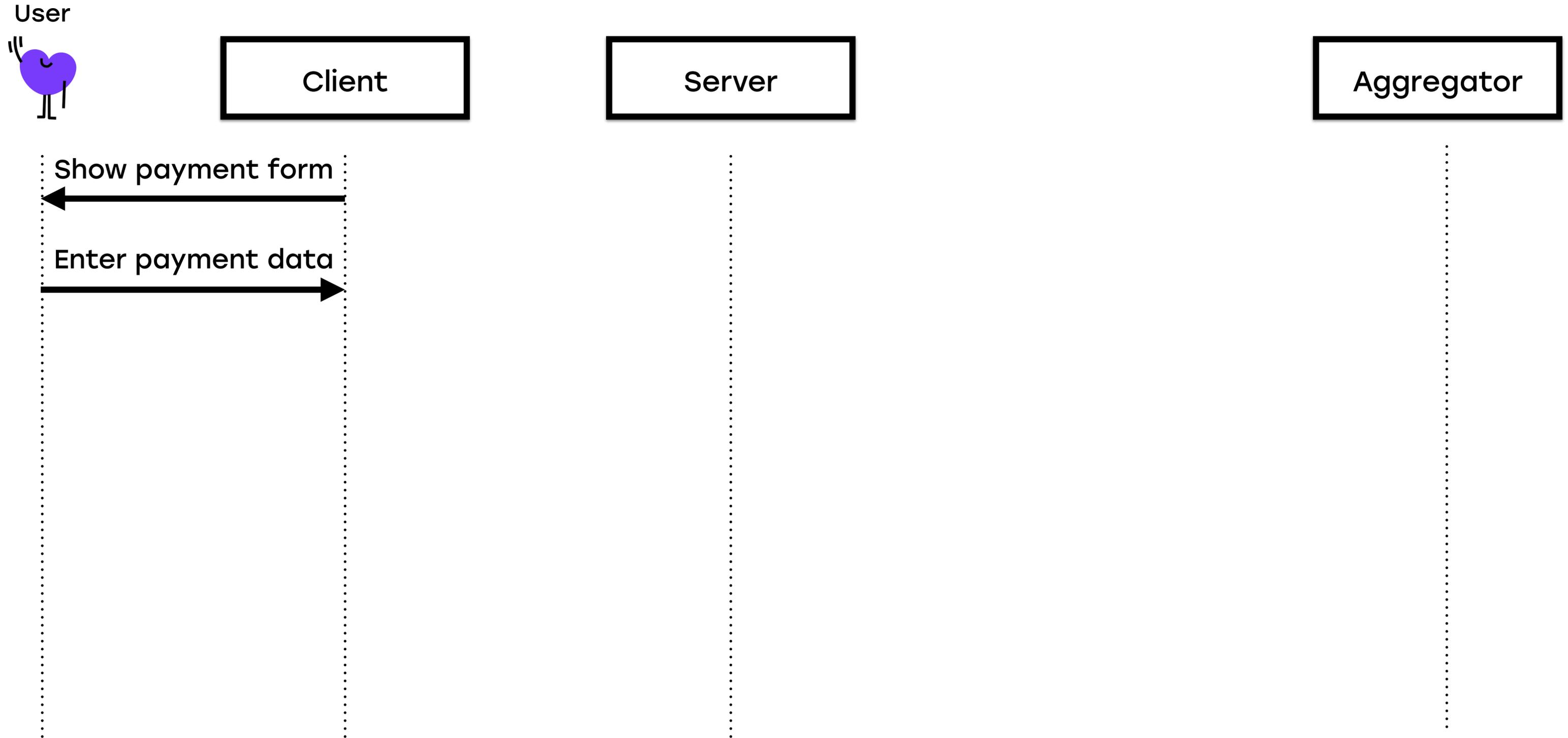
Кредитная карта

Процесс покупки

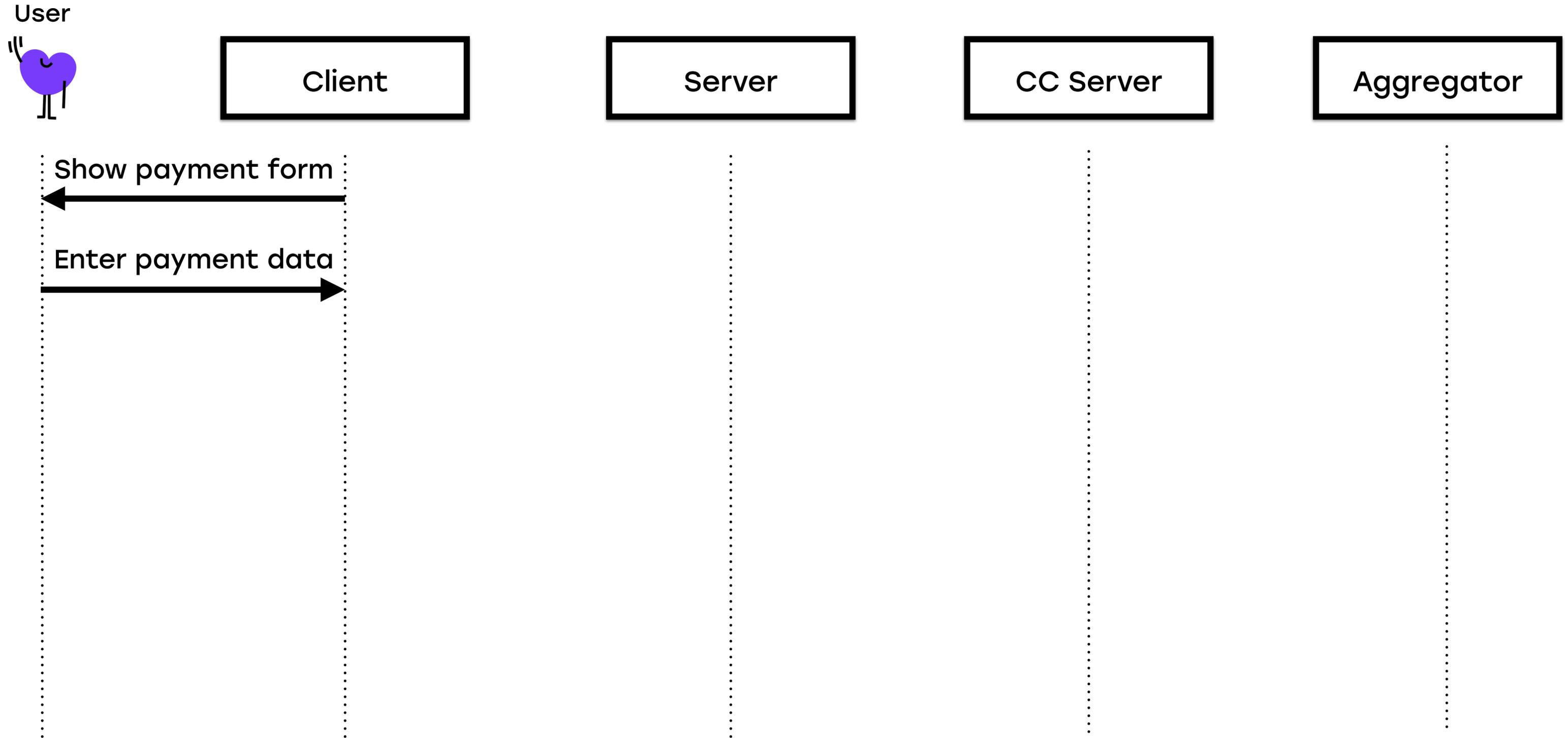
Кредитная карта Процесс покупки



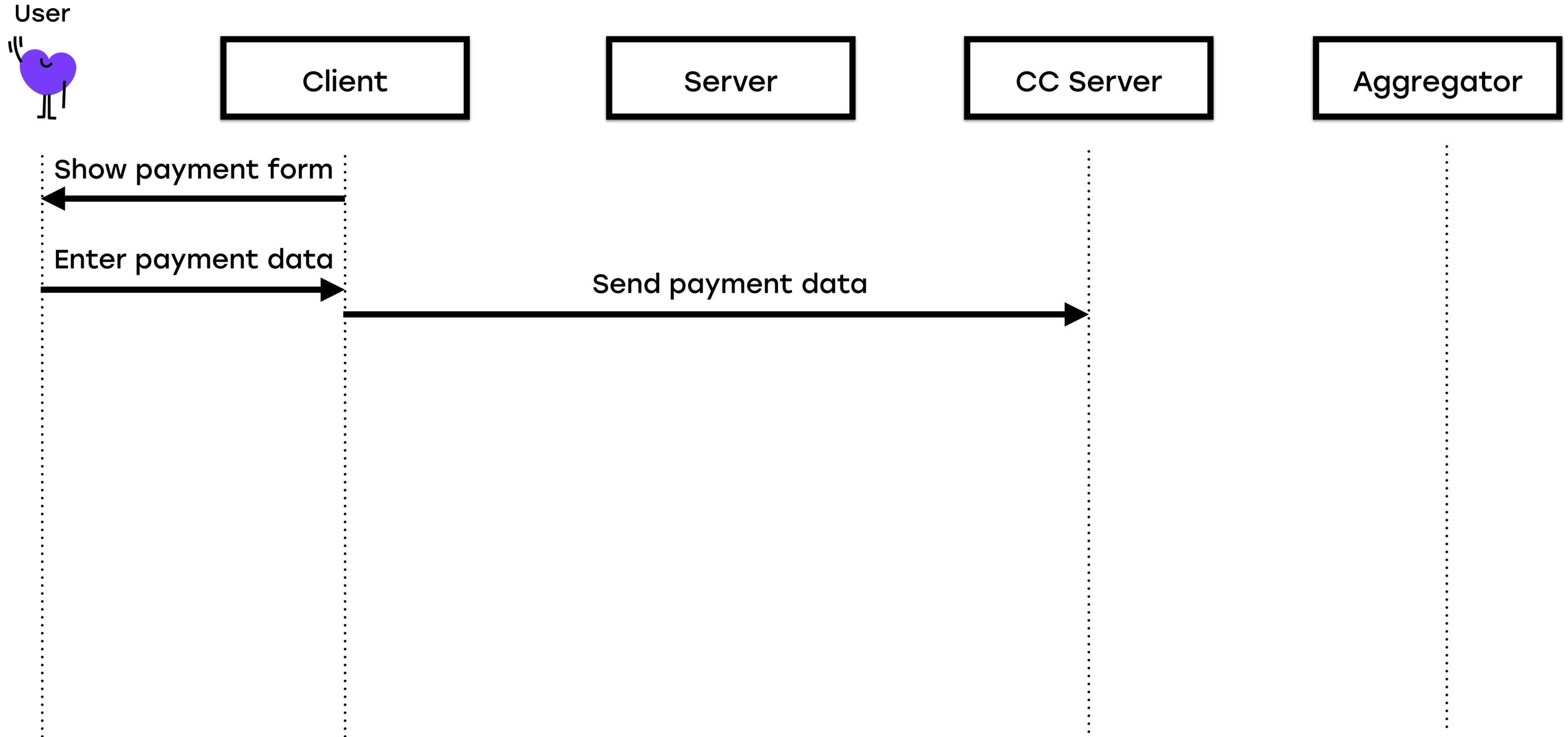
Кредитная карта Процесс покупки



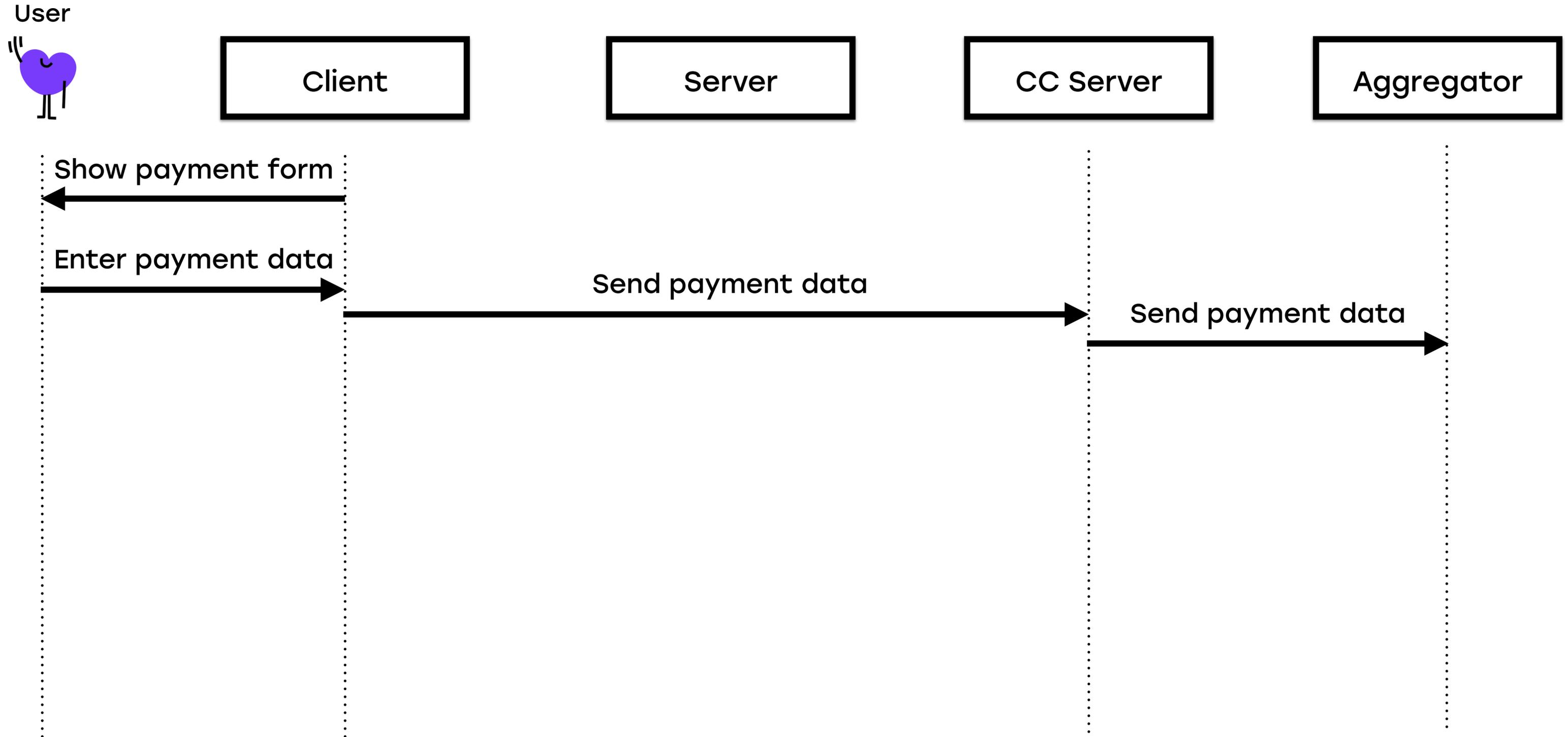
Кредитная карта Процесс покупки



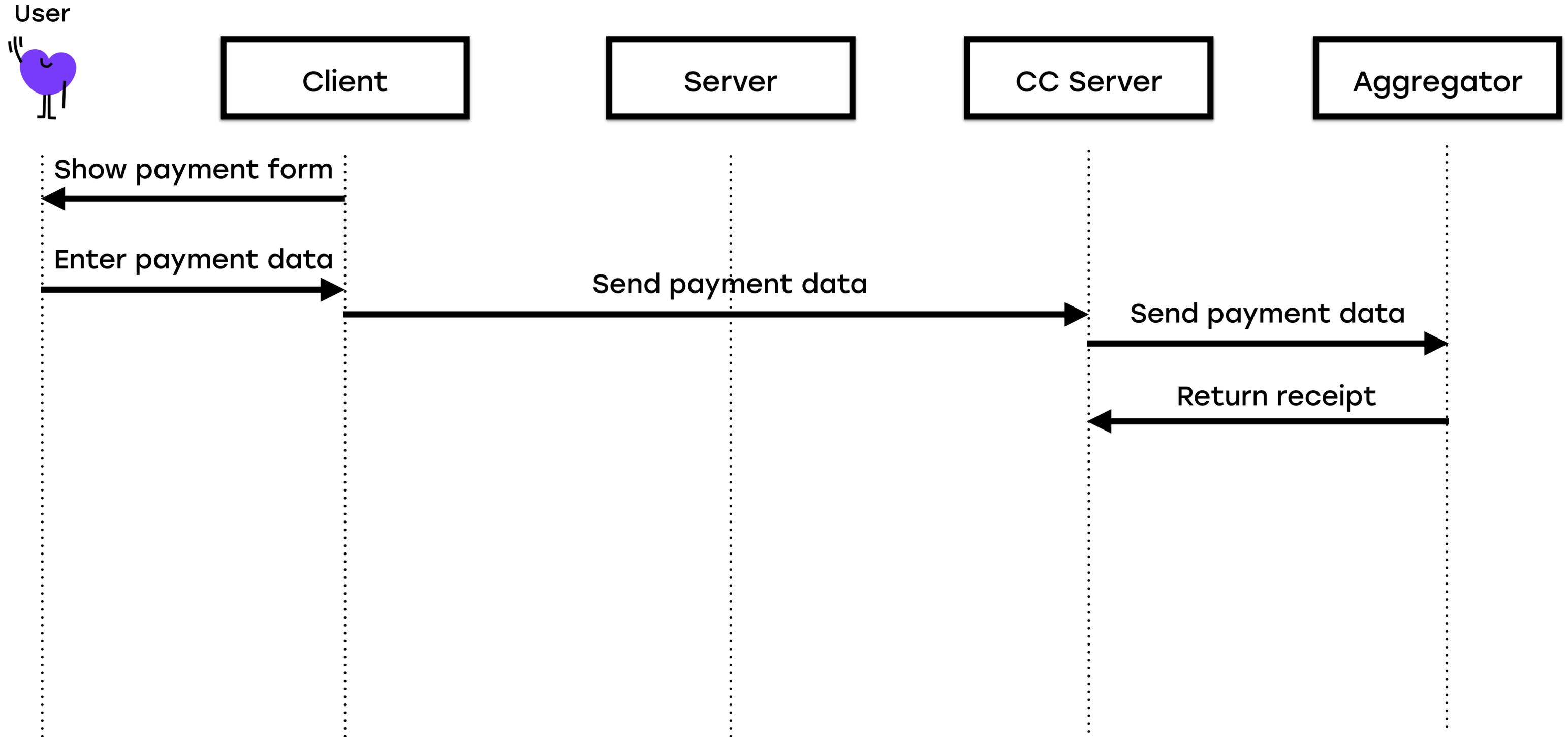
Кредитная карта Процесс покупки



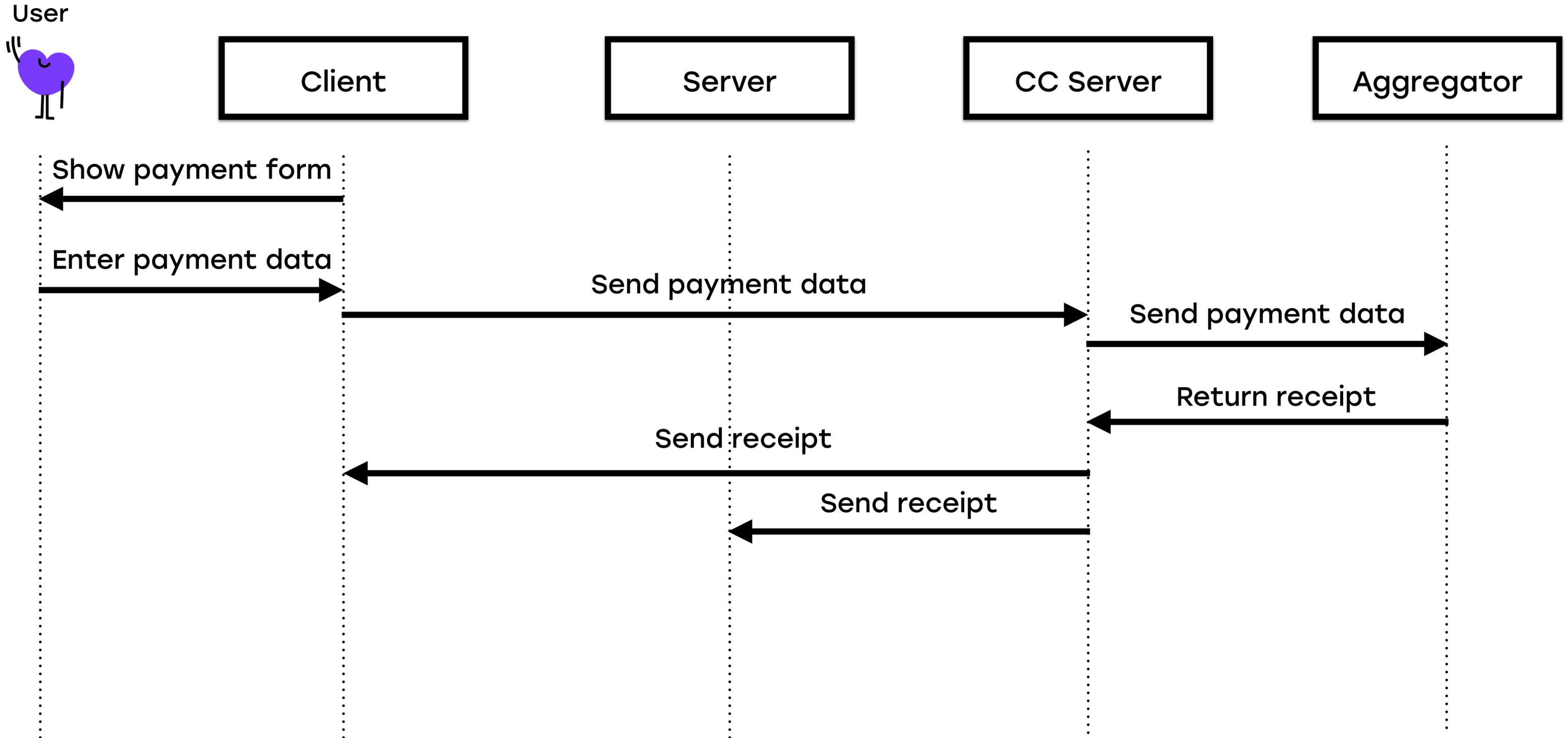
Кредитная карта Процесс покупки



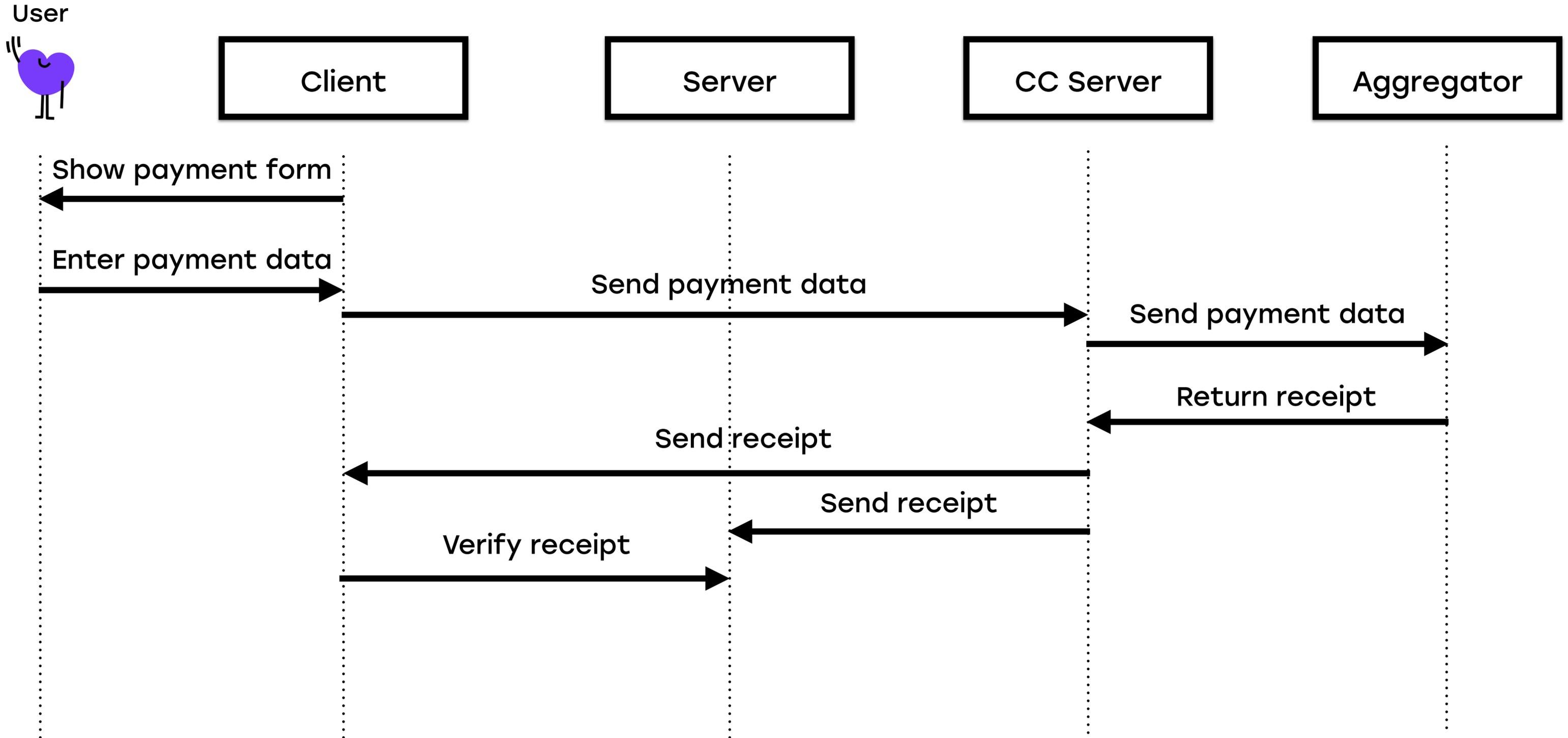
Кредитная карта Процесс покупки



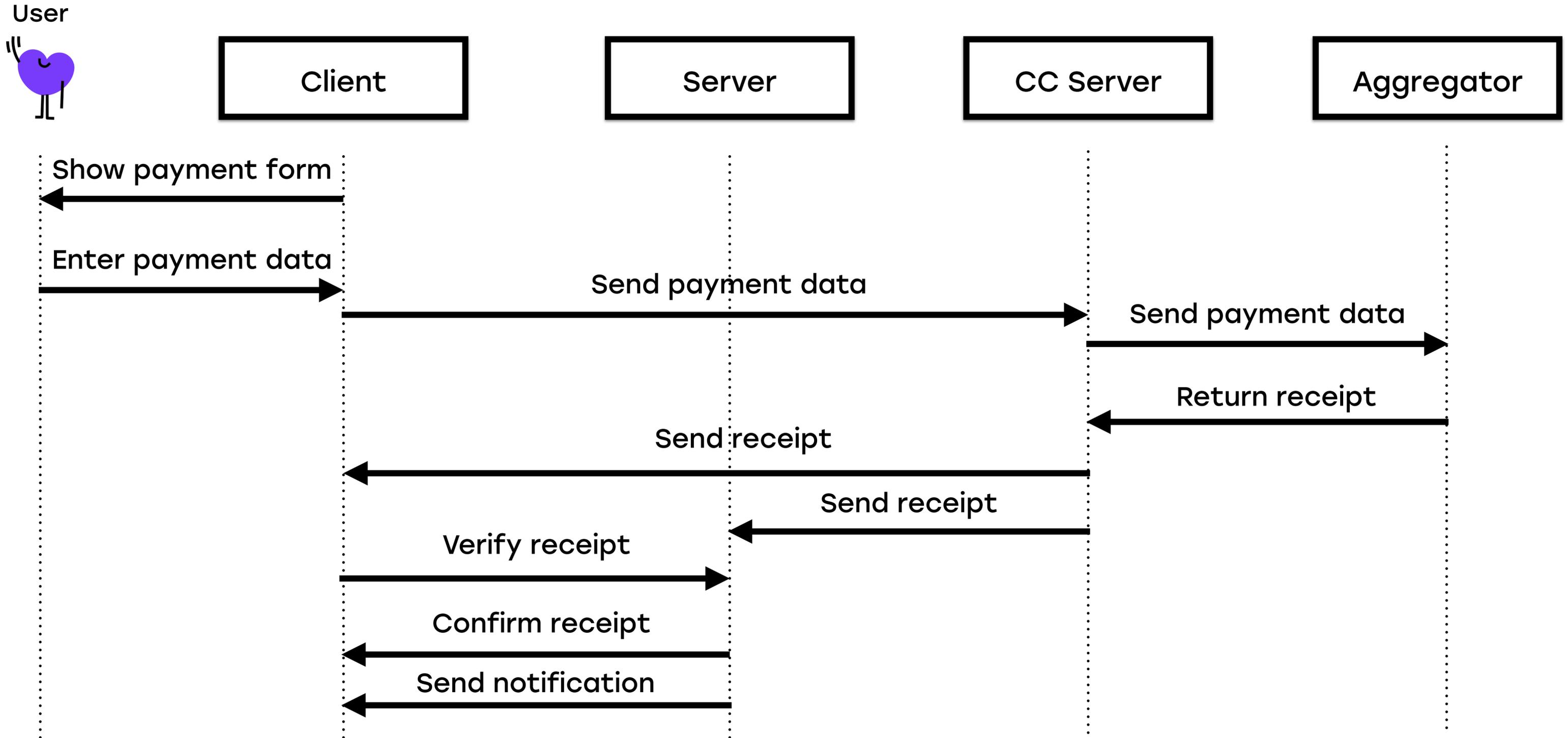
Кредитная карта Процесс покупки



Кредитная карта Процесс покупки



Кредитная карта Процесс покупки



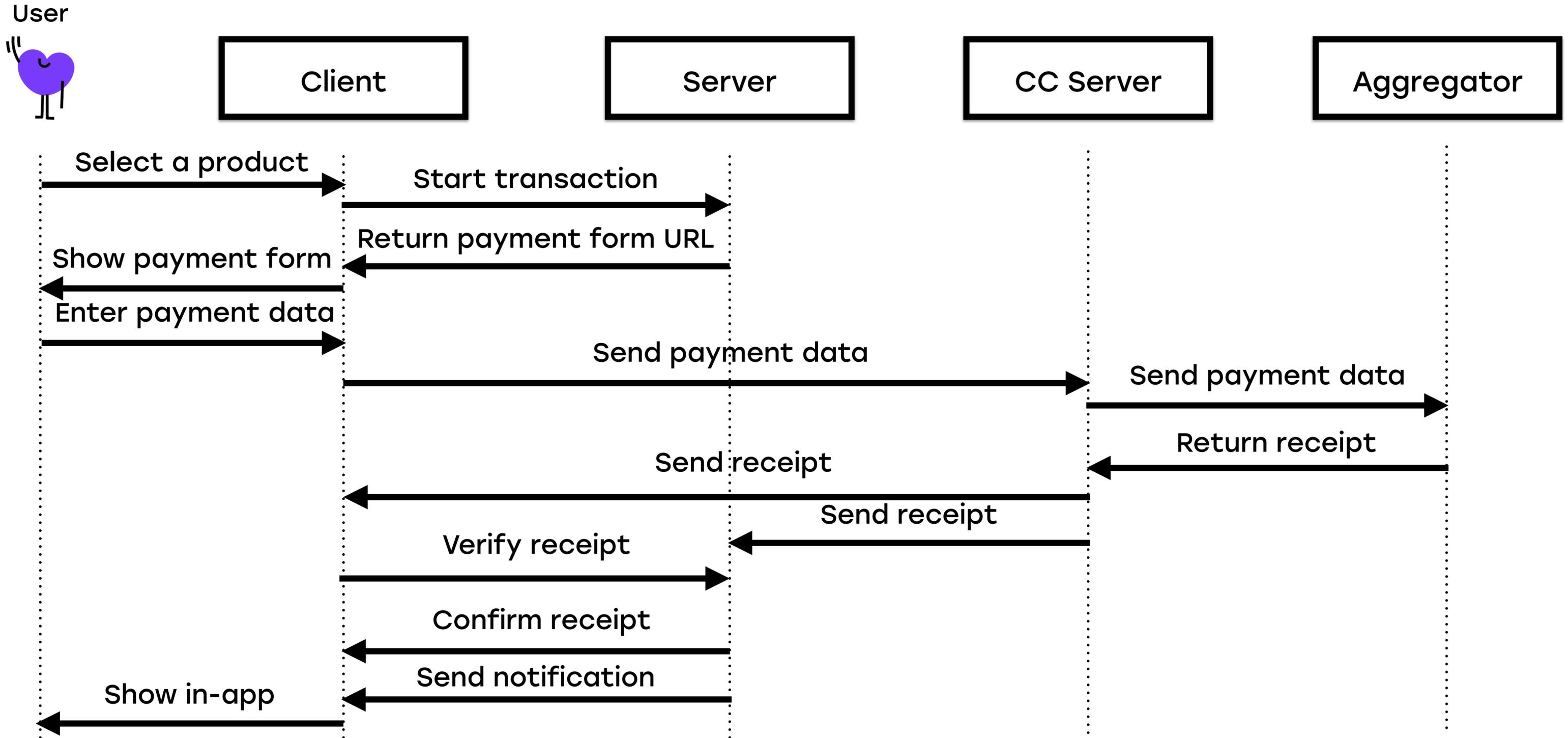
Кредитная карта

Тестовое окружение

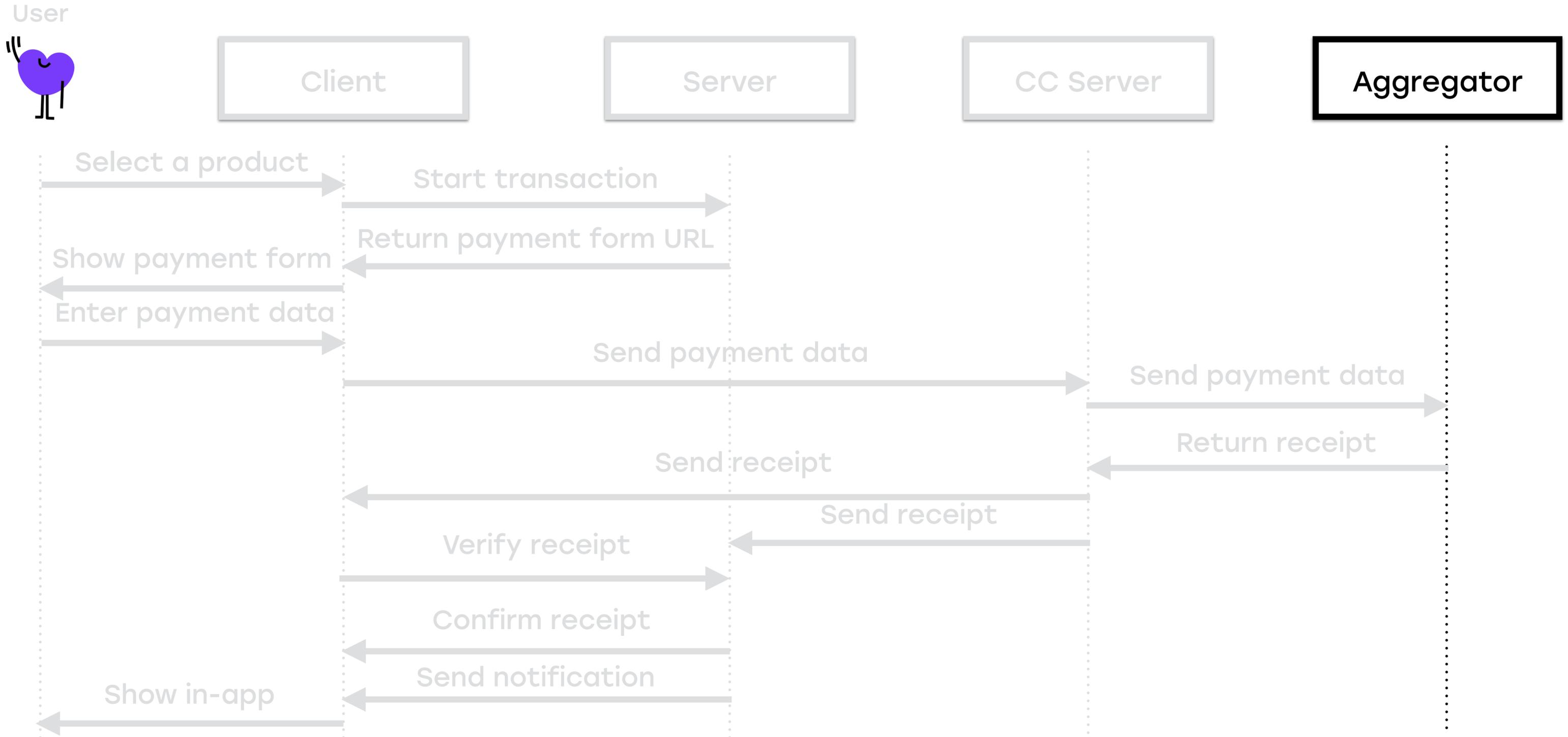
Почему песочницы

- + Процесс покупки максимально близок к реальному
- + Дешево
- Есть не у всех провайдеров

Кредитная карта Тестовое окружение



Кредитная карта Тестовое окружение

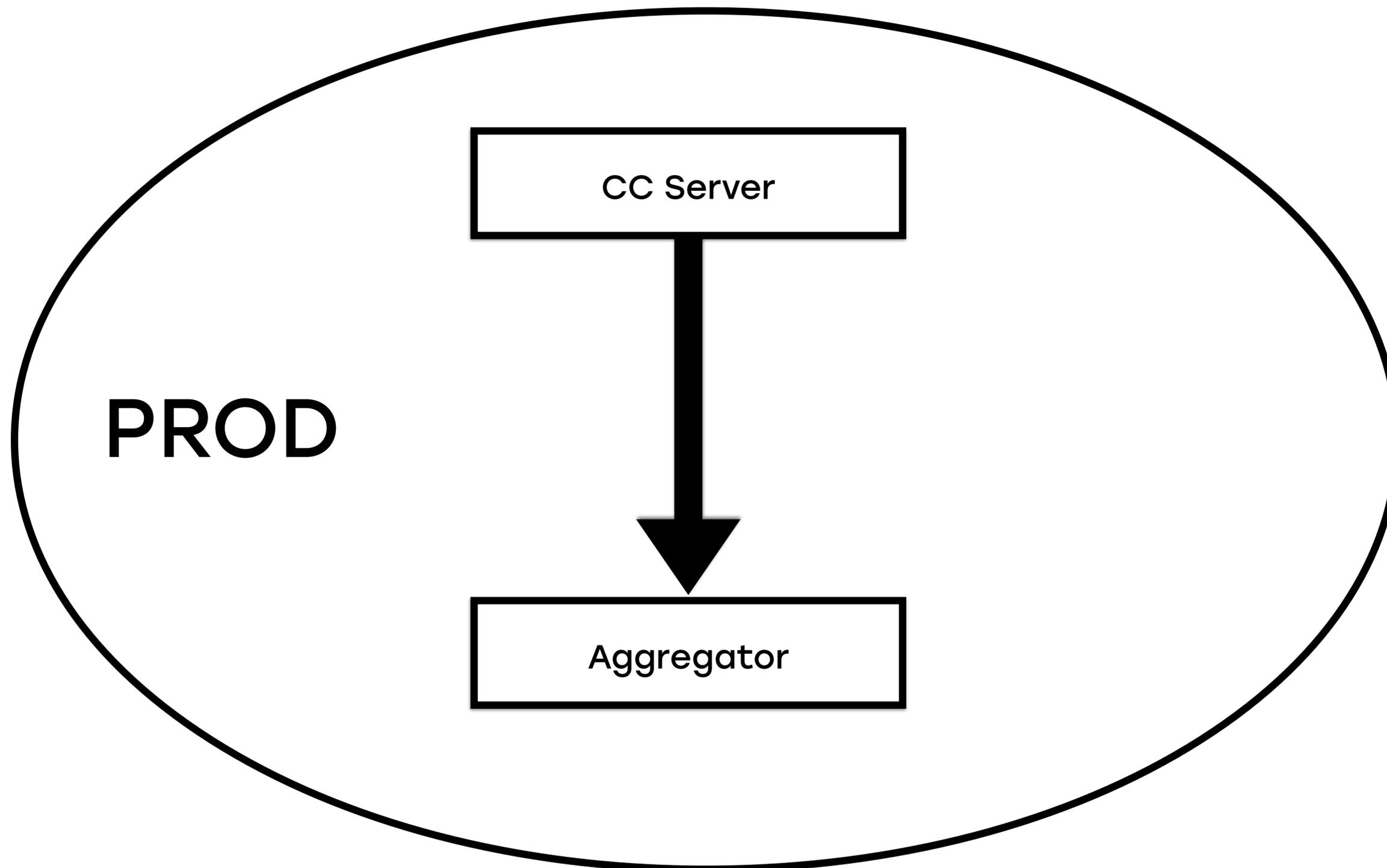


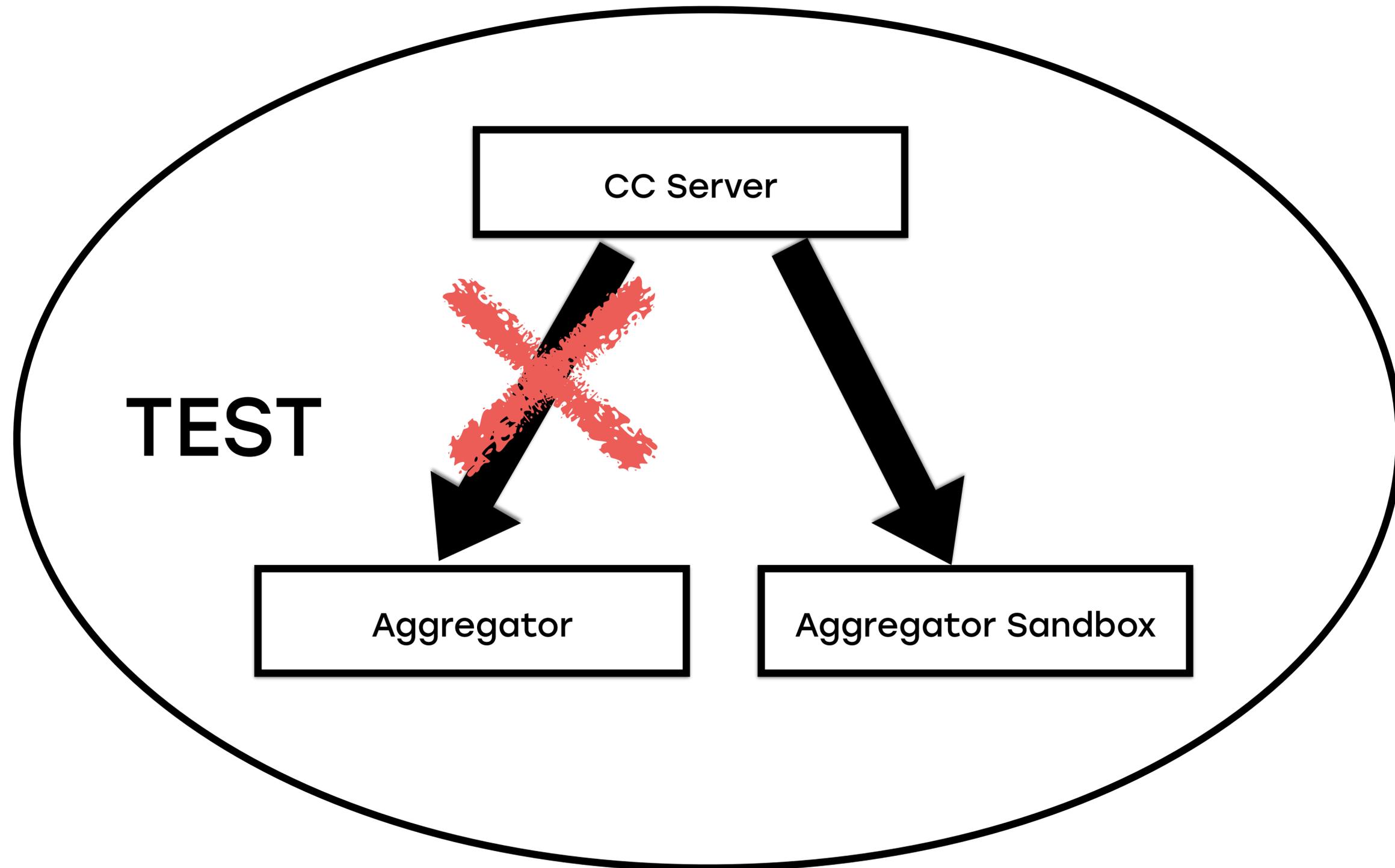
**Что надо
изменить?**



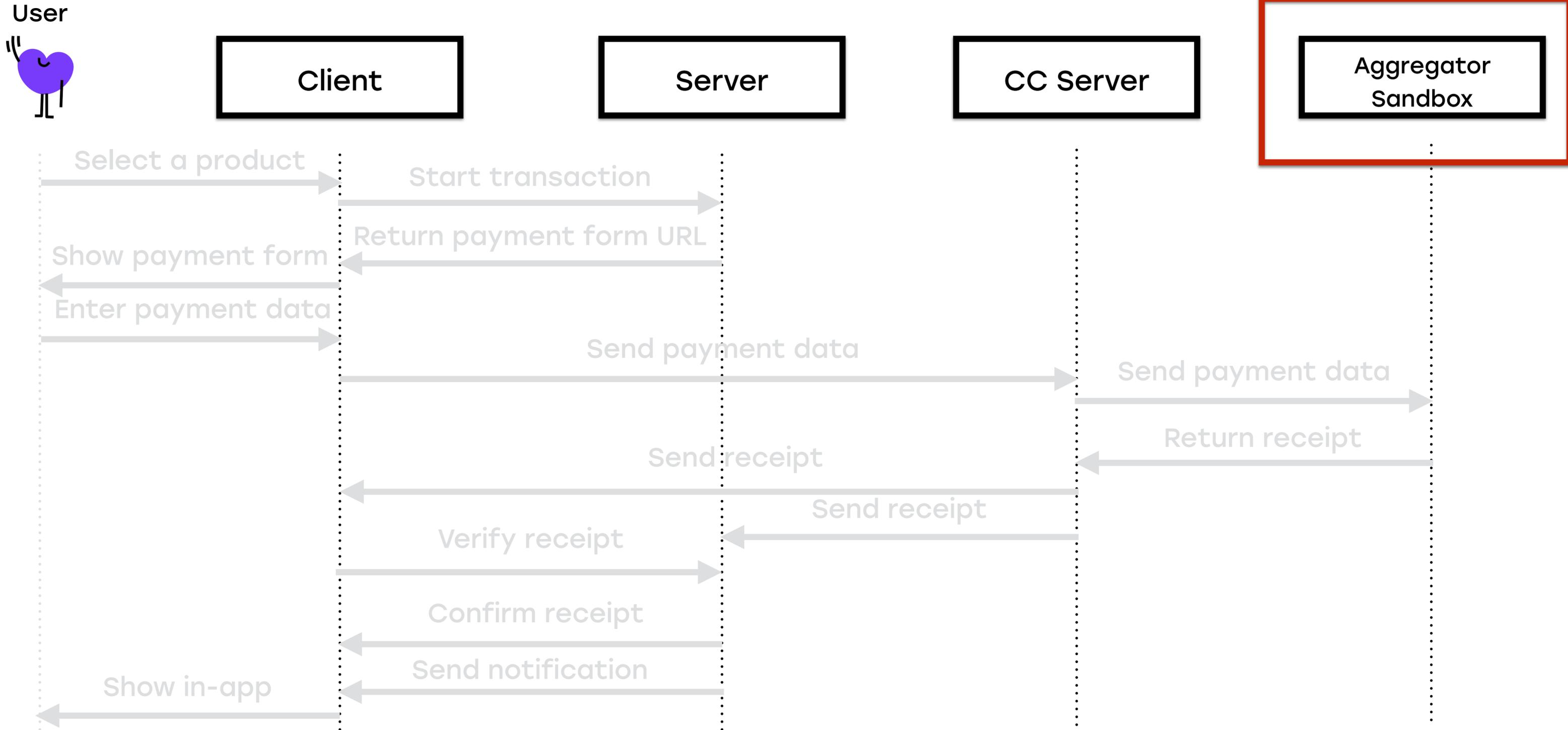
Что надо изменить?



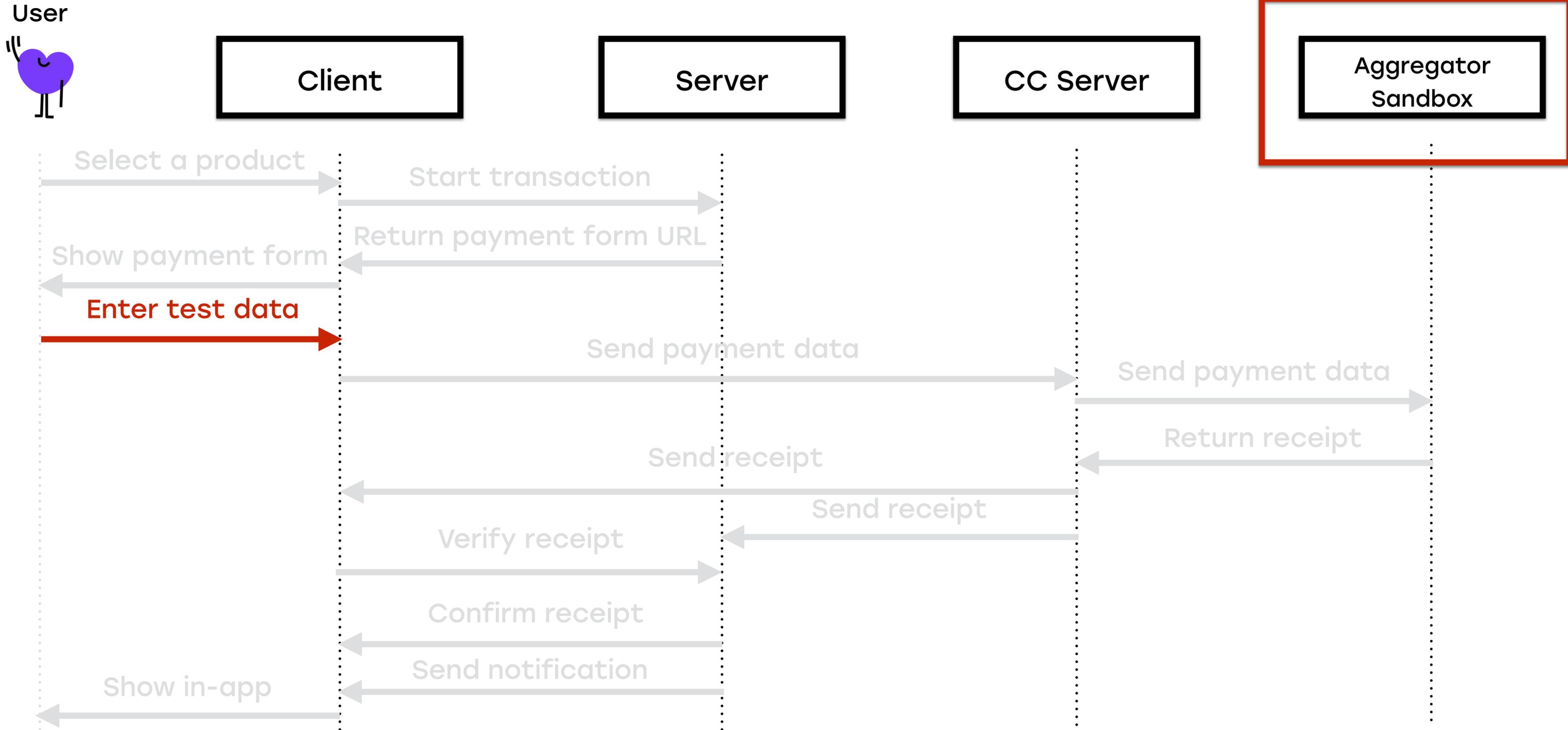




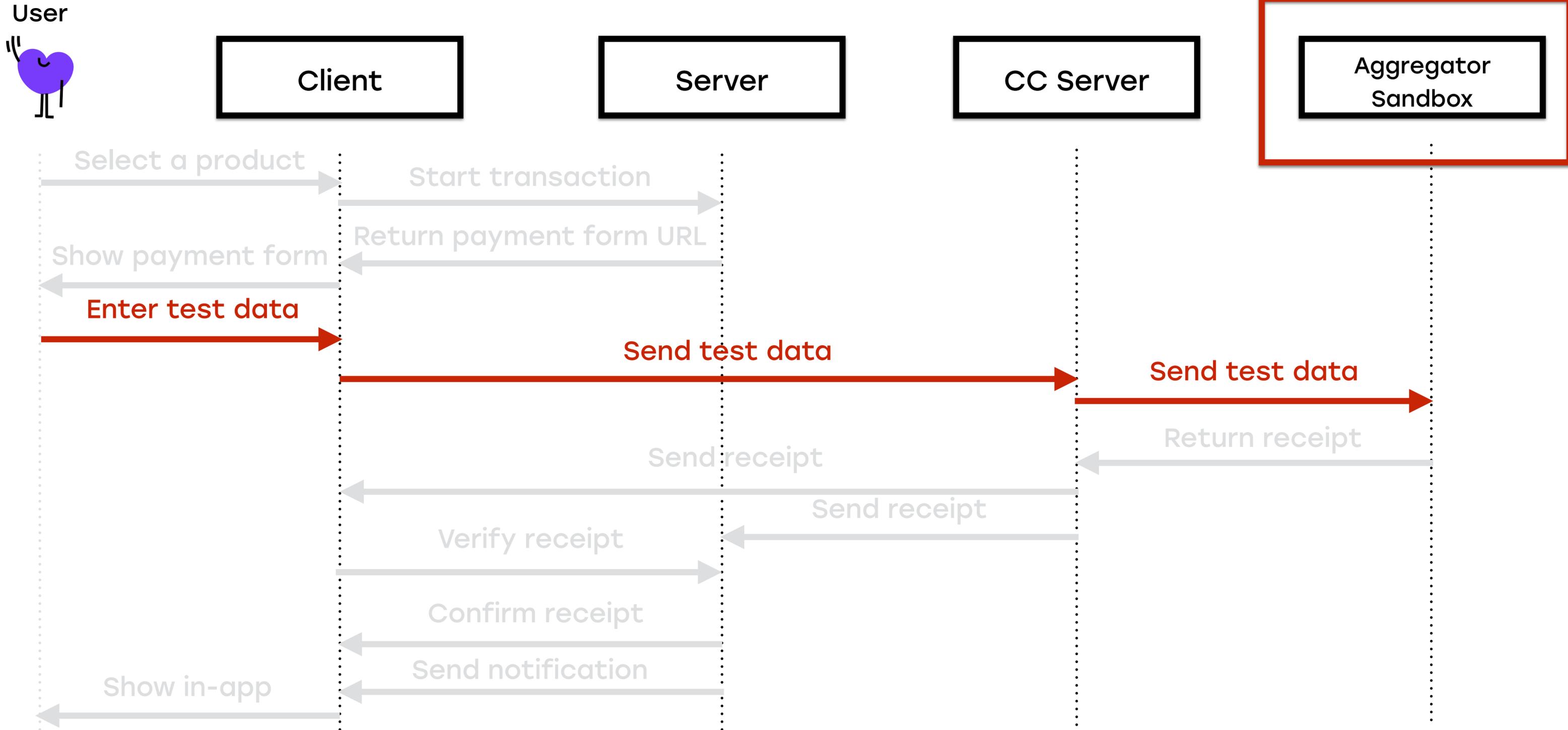
Кредитная карта Тестовое окружение



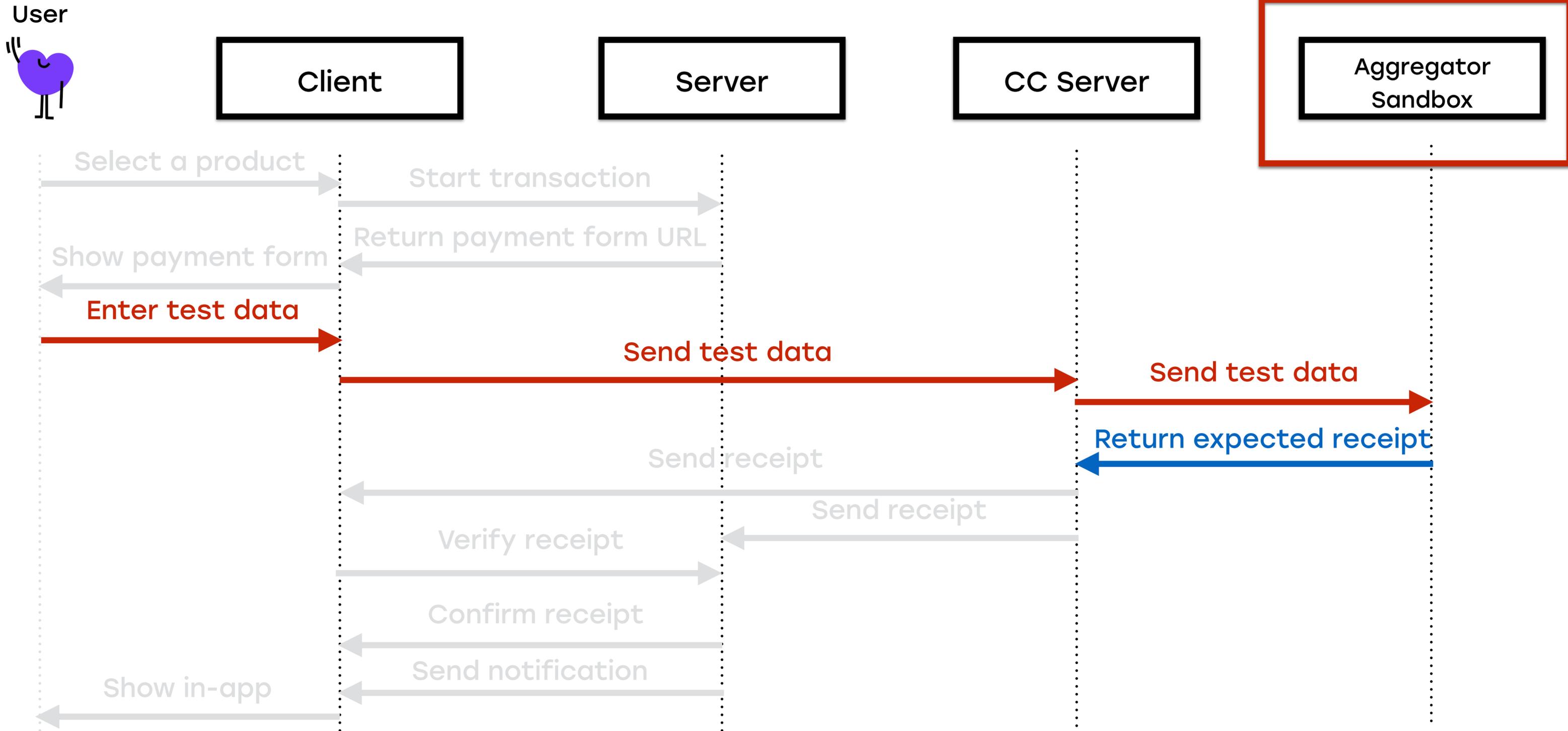
Кредитная карта Тестовое окружение



Кредитная карта Тестовое окружение



Кредитная карта Тестовое окружение



Набор песочниц

- Adyen, WorldPay, Ingenico, Computop

Набор песочниц

- Adyen, WorldPay, Ingenico, CompuTop
- **CC Server выбирает, к какой обратиться**

Набор песочниц

- Adyen, WorldPay, Ingenico, CompuTop
- CC Server выбирает, к какой обратиться
 - **Параметры транзакции**

Набор песочниц

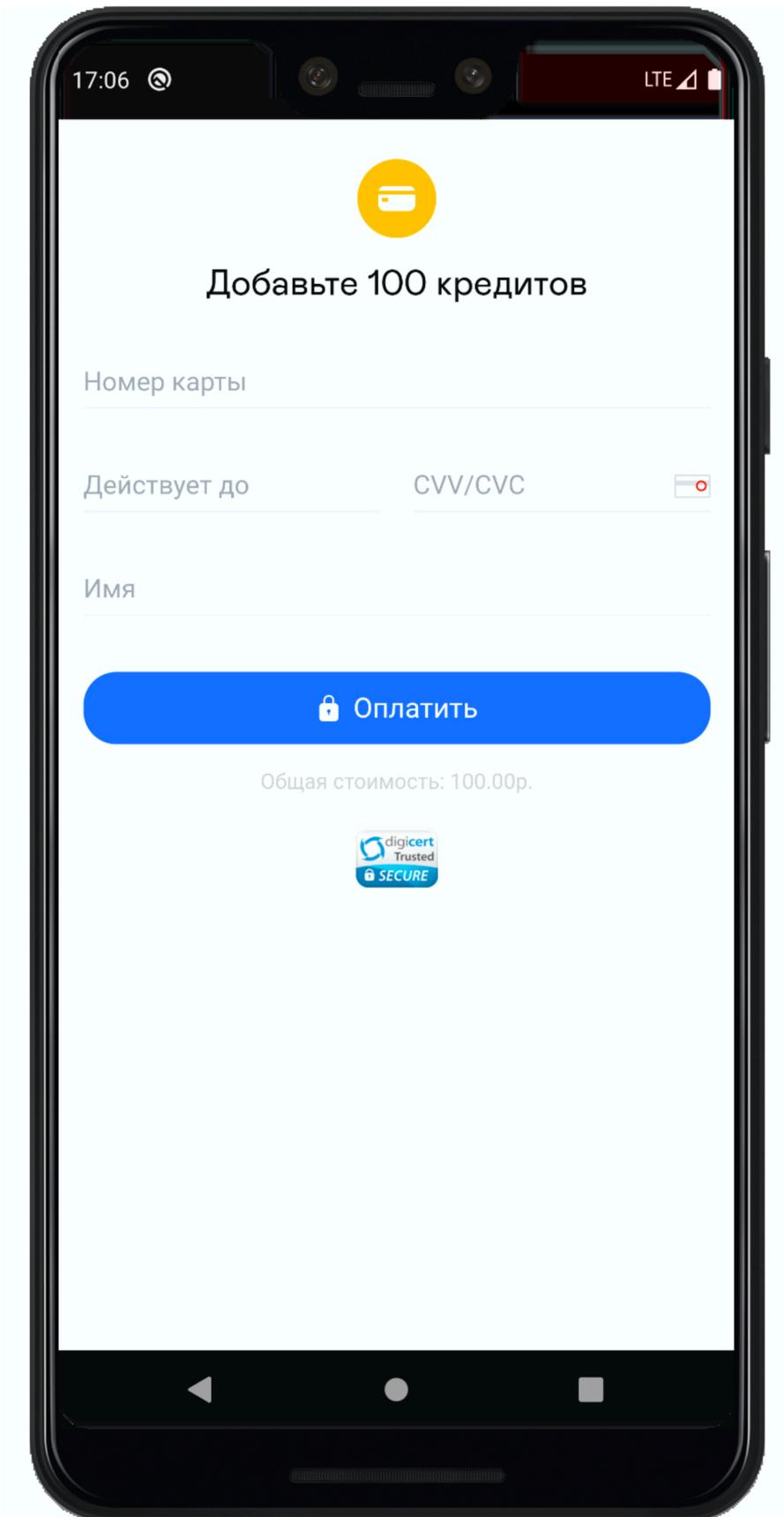
- Adyen, WorldPay, Ingenico, Computop
- CC Server выбирает, к какой обратиться
 - Параметры транзакции
 - **Параметры карты**

Кредитная карта

Автоматизация

Экран оплаты

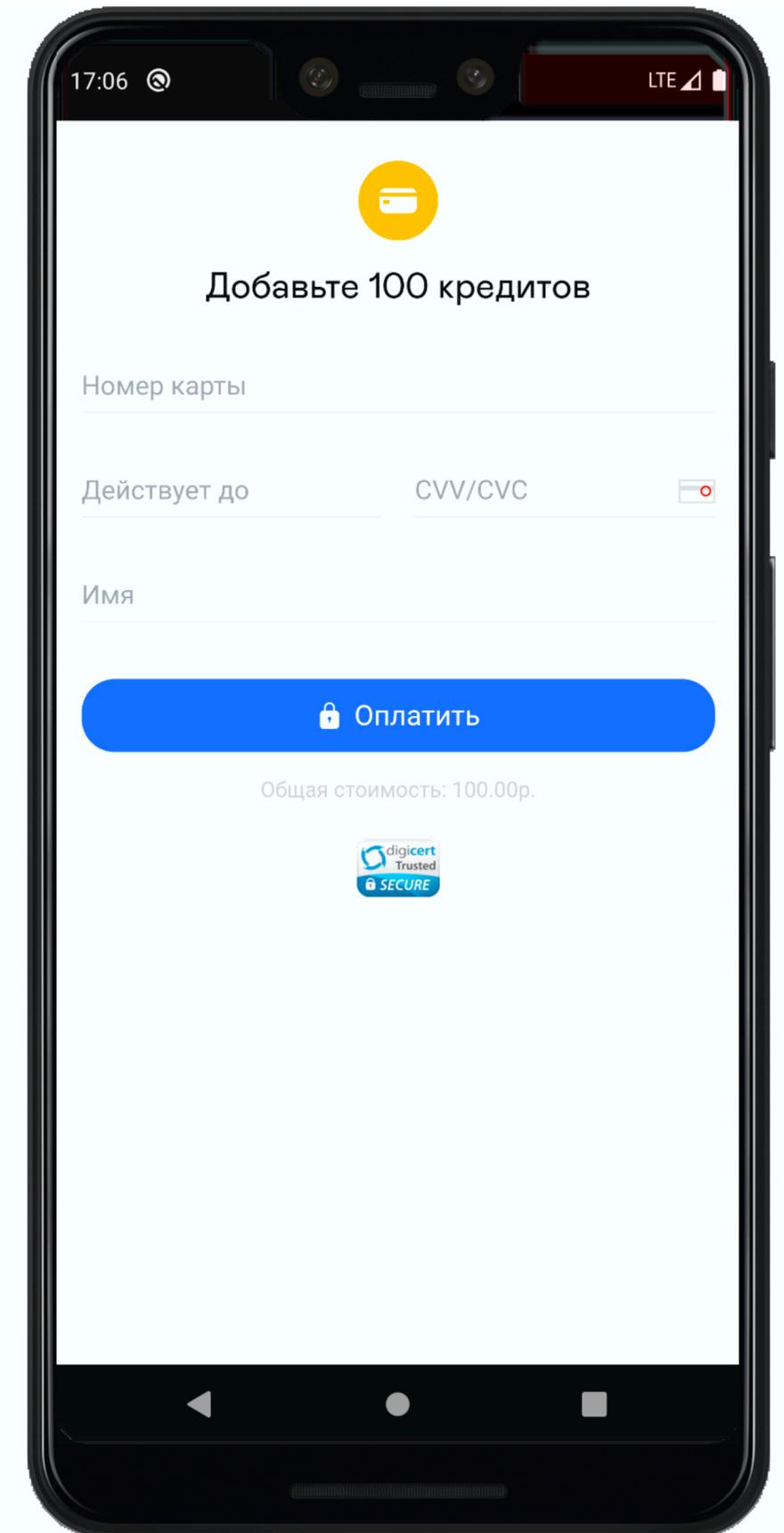
- **WebView**



Экран оплаты

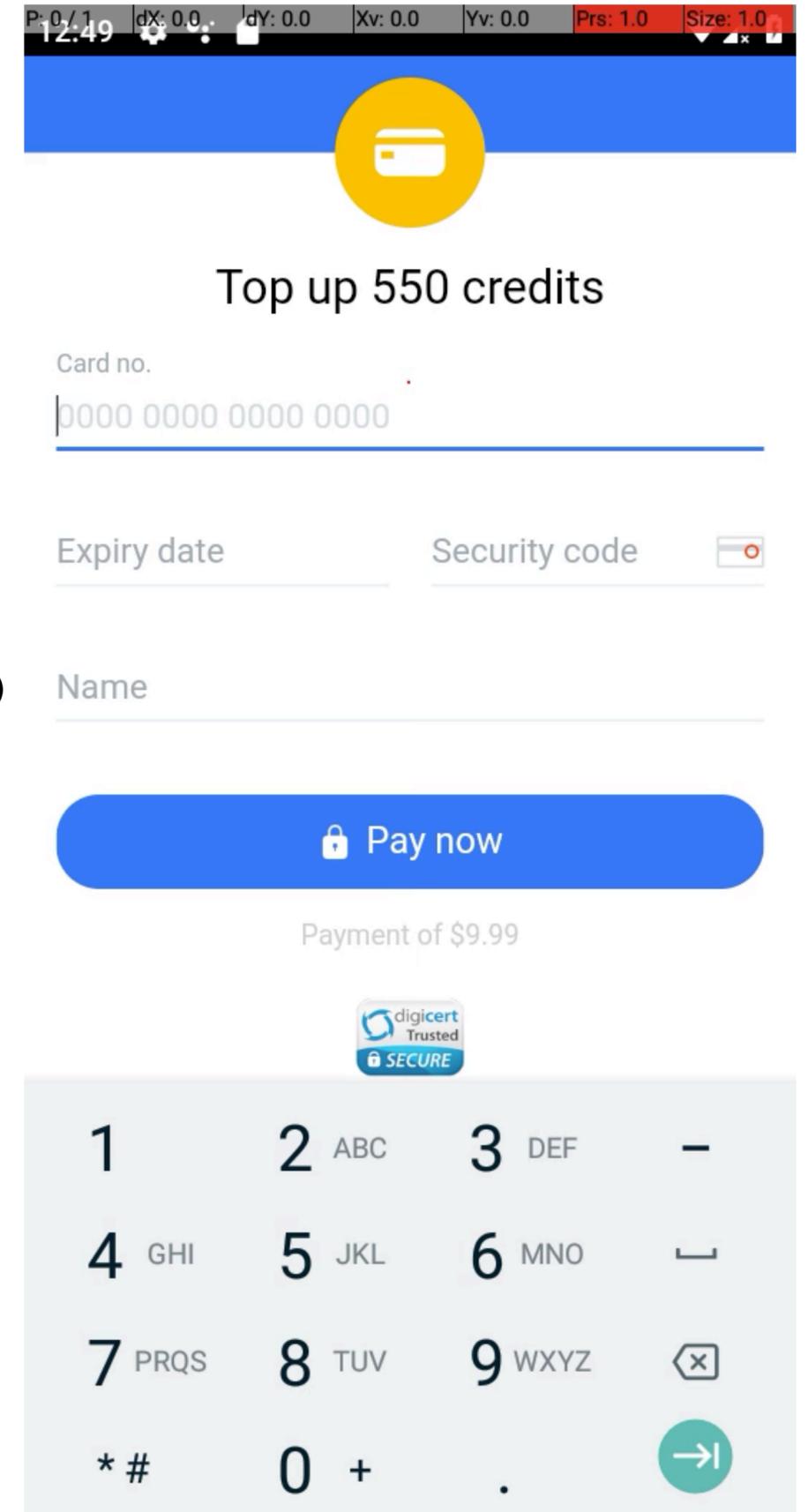
- **WebView**
- **Есть доступ к элементам**

```
CARD_NUMBER = "webView css:'.js-input-cc'"  
EXPIRY_DATE = "webView css:'#expiry_date'"  
SECURITY_CODE = "webView css:'#security_code'"  
NAME_ON_CARD = "webView css:'#name_on_card'"  
SUBMIT_BUTTON = "webView css:'#js-submit-button'"
```



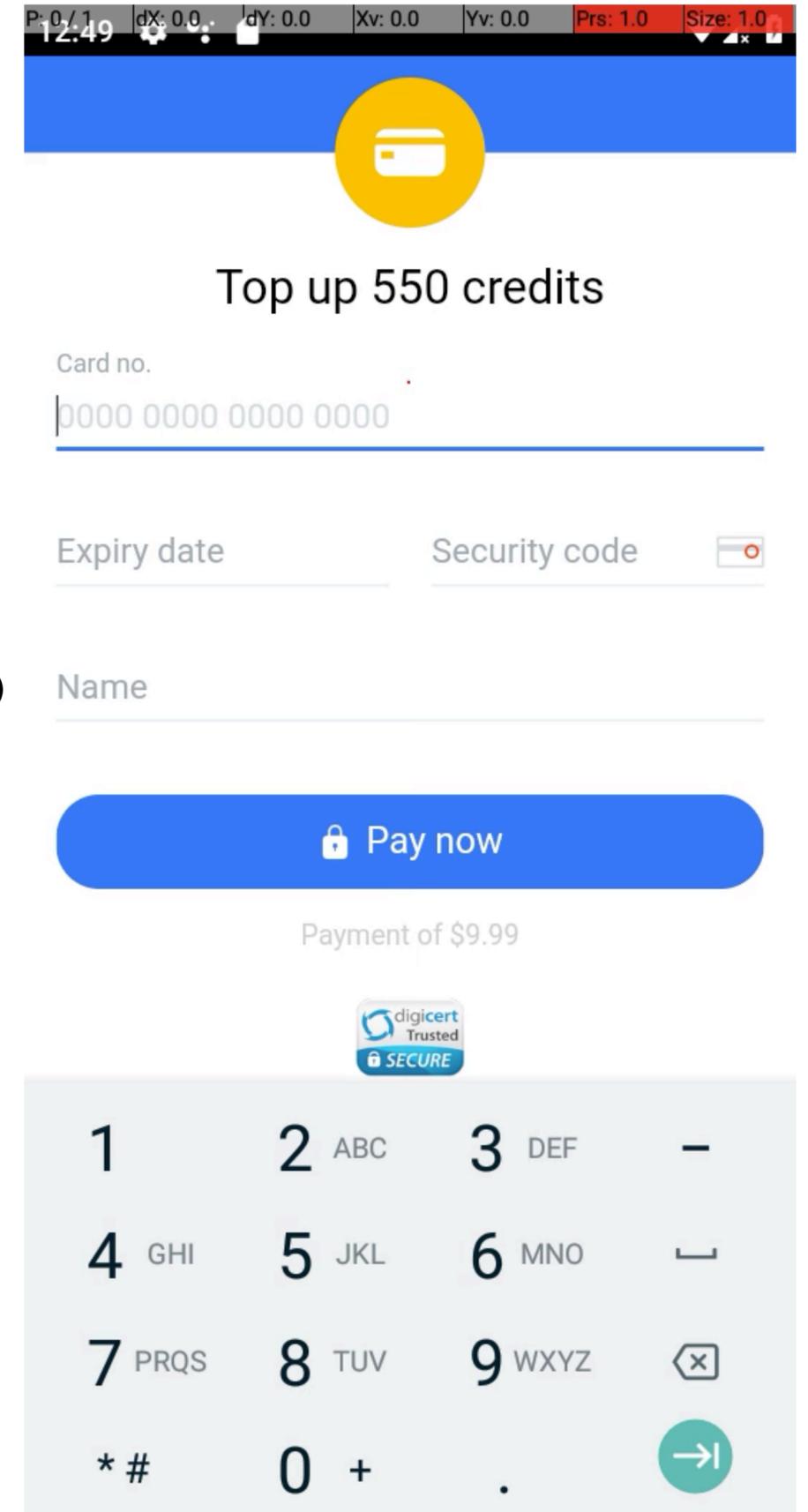
Платеж

```
def submit_payment(payment_details)
  set_text_in_field(CARD_NUMBER, payment_details[:number])
  set_text_in_field(EXPIRY_DATE, payment_details[:expiry_month] + payment_details[:expiry_year][2..3])
  set_text_in_field(SEcurity_CODE, payment_details[:security_code])
  set_text_in_field(NAME_ON_CARD, payment_details[:name])
  hide_keyboard
  tap_element(SUBMIT_BUTTON)
end
```



Платеж

```
def submit_payment(payment_details)
  set_text_in_field(CARD_NUMBER, payment_details[:number])
  set_text_in_field(EXPIRY_DATE, payment_details[:expiry_month] + payment_details[:expiry_year][2..3])
  set_text_in_field(SEcurity_CODE, payment_details[:security_code])
  set_text_in_field(NAME_ON_CARD, payment_details[:name])
  hide_keyboard
  tap_element(SUBMIT_BUTTON)
end
```



Тестовые данные

- Доступны в песочнице провайдера

Тестовые данные

- Доступны в песочнице провайдера
- **Настроить в песочнице самостоятельно**

Дополнительные сценарии

- Сохранение деталей

Дополнительные сценарии

- Сохранение деталей
- 3-D Secure

Сохранение деталей

- Не надо много раз вводить данные карты

Сохранение деталей

- Не надо много раз вводить данные карты
- **НО работает не для всех карт**

Сохранение деталей

- Не надо много раз вводить данные карты
- НО работает не для всех карт
- **Автоматизируем оба случая**

Сохранение деталей

- Не надо много раз вводить данные карты
- НО работает не для всех карт
- Автоматизируем оба случая
 - **Сохранение данных и последующие покупки**

Сохранение деталей

- Не надо много раз вводить данные карты
- НО работает не для всех карт
- Автоматизируем оба случая
 - Сохранение данных и последующие покупки
 - Покупку картой без возможности сохранения

3-D Secure

- **Безопасность онлайн платежей**

3-D Secure

- **Безопасность онлайн платежей**
- **Двухфакторная аутентификация**

3-D Secure. Способы подтверждения

- Код в SMS или email

3-D Secure. Способы подтверждения

- Код в SMS или email
- Логин/пароль

3-D Secure 2.0

- Октябрь 2016

3-D Secure 2.0

- Октябрь 2016
- **Больше данных между продавцом и банком**

3-D Secure 2.0

- Октябрь 2016
- Больше данных между продавцом и банком
- **Биометрия**

3-D Secure. Автоматизация

- 2 тестовые карты

3-D Secure. Автоматизация

- 2 тестовые карты
- 2 страницы в тестах

3-D Secure. Автоматизация

3-D Secure. Автоматизация

```
class ThreeDSPage < Calabash::AndroidBadooBase
  USERNAME      = "webView css:'#username'"
  PASSWORD      = "webView css:'#password'"
  SUBMIT_BUTTON = "webView css:'[type=submit]'"

  def complete_challenge(payment_details)
    enter_text(element: USERNAME, text: payment_details[:user])
    hide_keyboard
    enter_text(element: PASSWORD, text: payment_details[:password])
    hide_keyboard
    tap_element(SUBMIT_BUTTON)
  end
end
```

3-D Secure. АВТОМАТИЗАЦИЯ

```
class ThreeDSPage < Calabash::AndroidBadooBase
  USERNAME      = "webView css:'#username'"
  PASSWORD      = "webView css:'#password'"
  SUBMIT_BUTTON = "webView css:'[type=submit]'"

  def complete_challenge(payment_details)
    enter_text(element: USERNAME, text: payment_details[:user])
    hide_keyboard
    enter_text(element: PASSWORD, text: payment_details[:password])
    hide_keyboard
    tap_element(SUBMIT_BUTTON)
  end
end
```

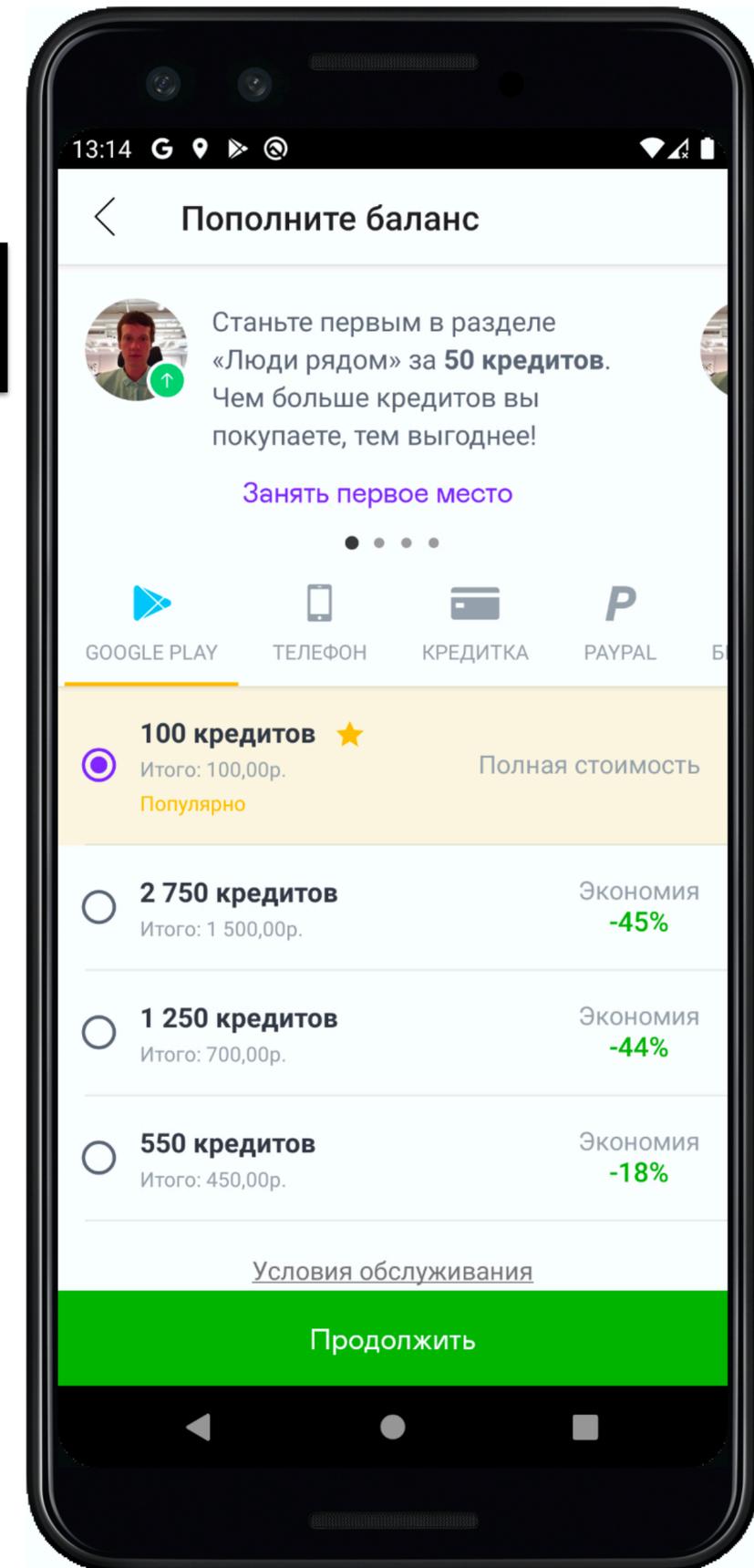
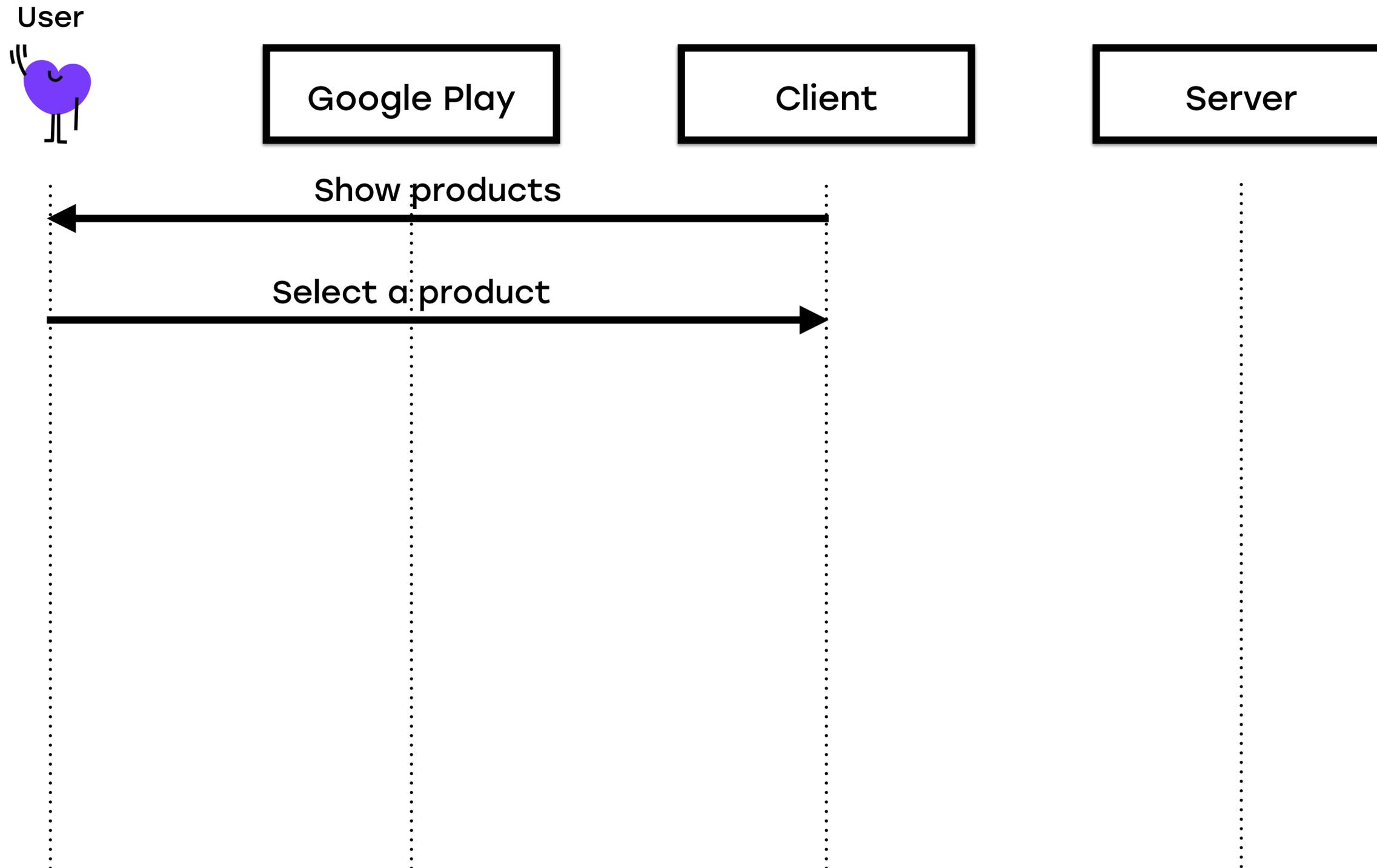
```
class ThreeDSv2Page < Calabash::AndroidBadooBase

  def complete_challenge(payment_details)
    uia_touch_text('Submit') # workaround to put focus on the pass
    enter_text(text: payment_details[:password])
    uia_touch_text('Submit')
  end
end
```

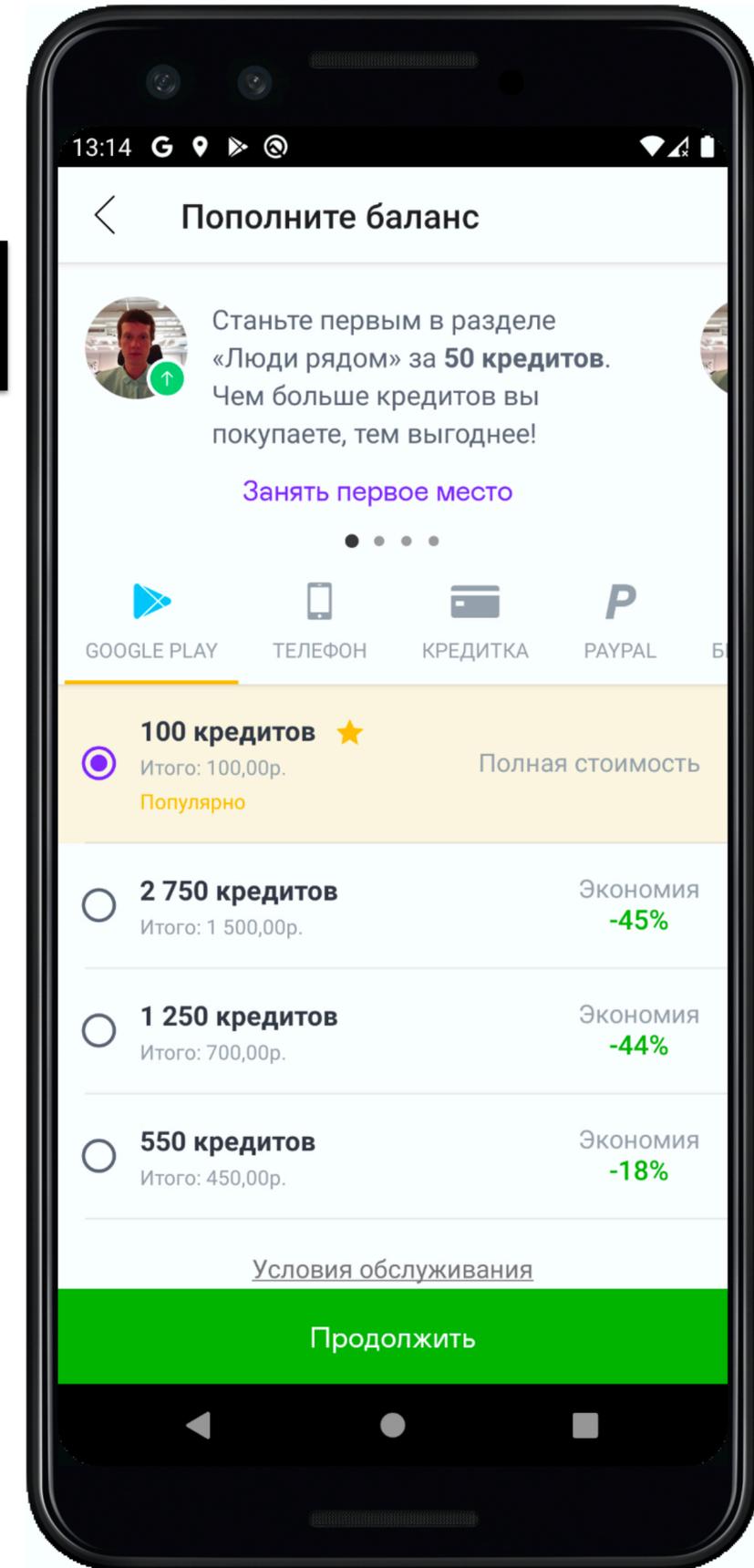
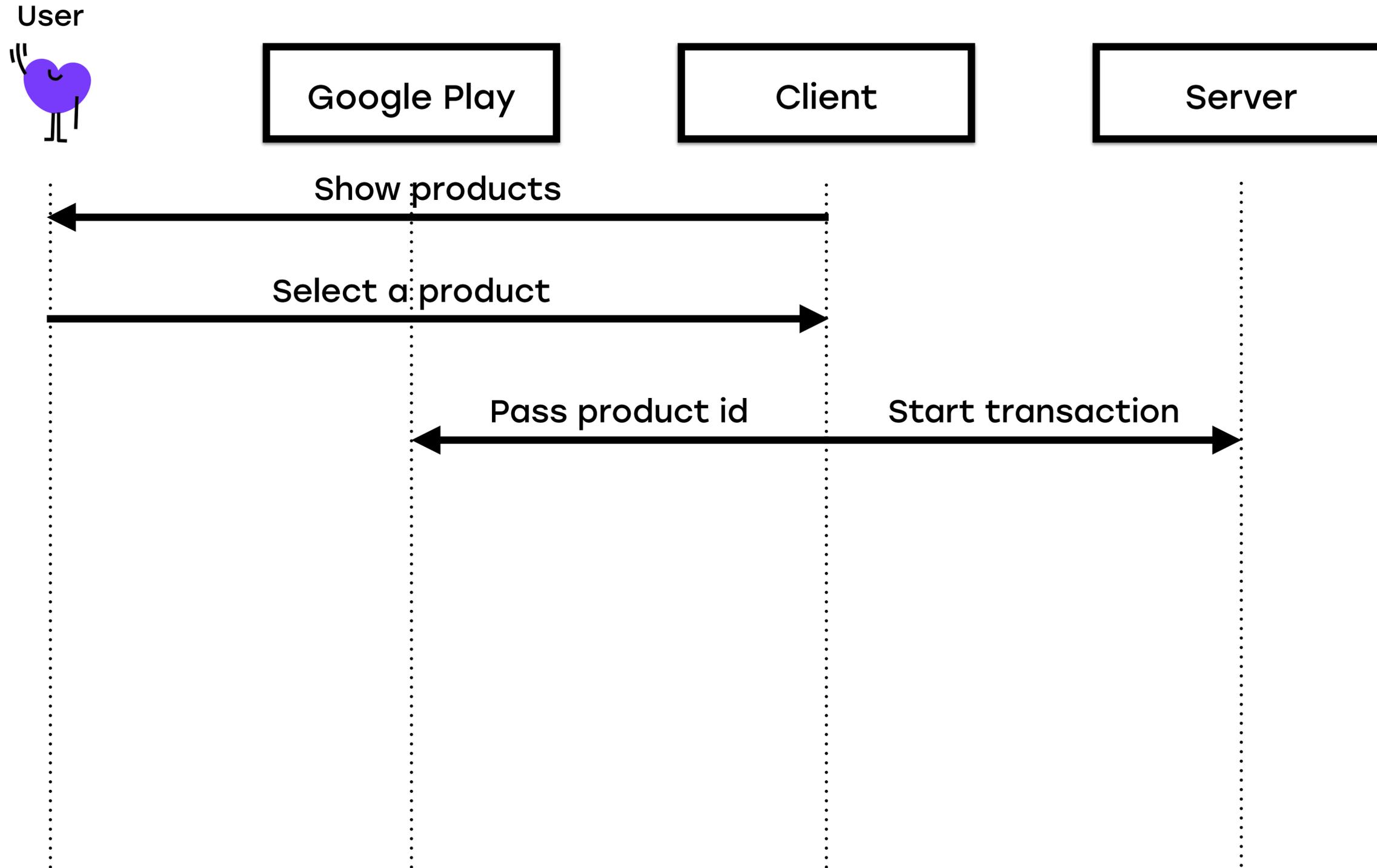
Google Play

Процесс покупки

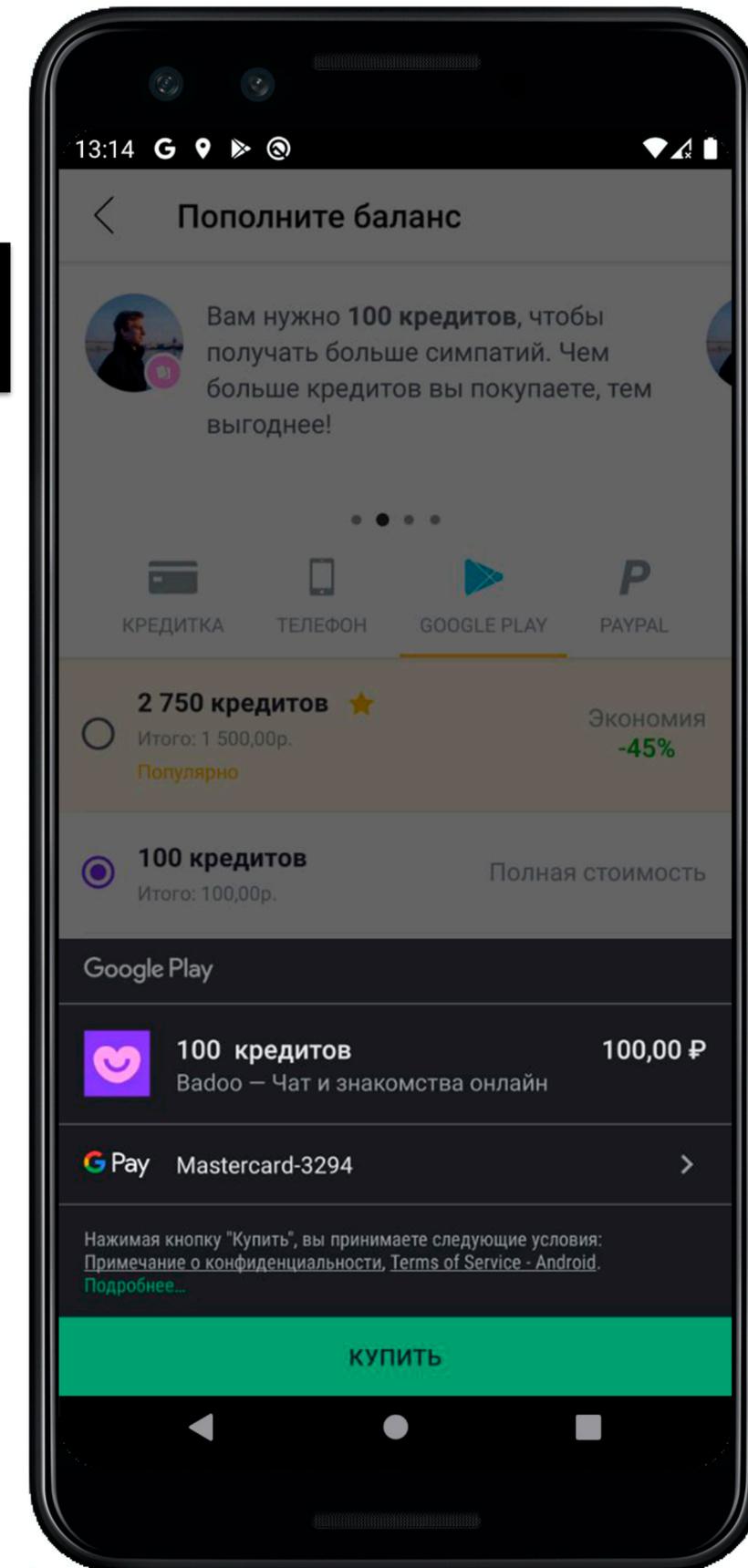
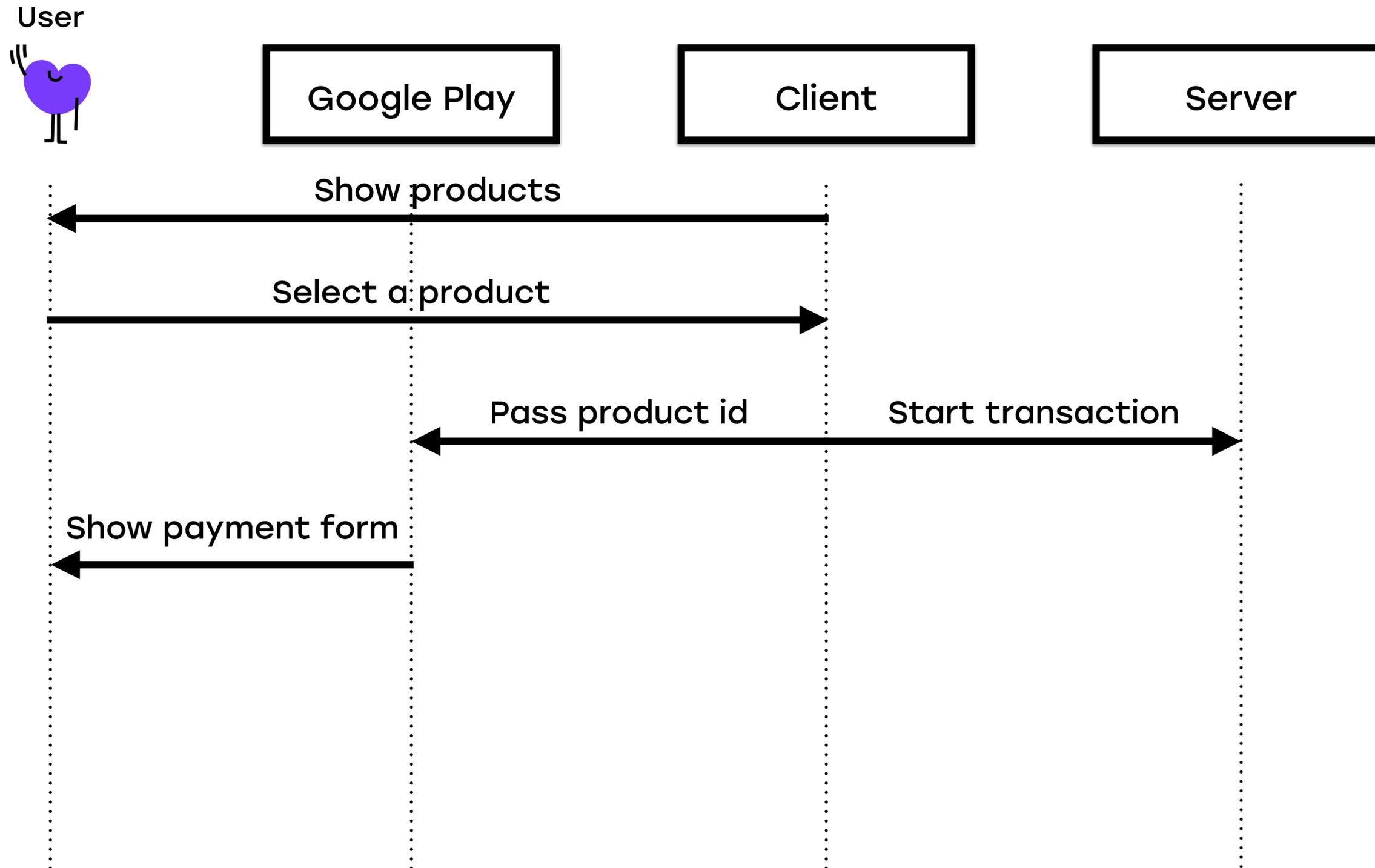
Google Play Процесс покупки



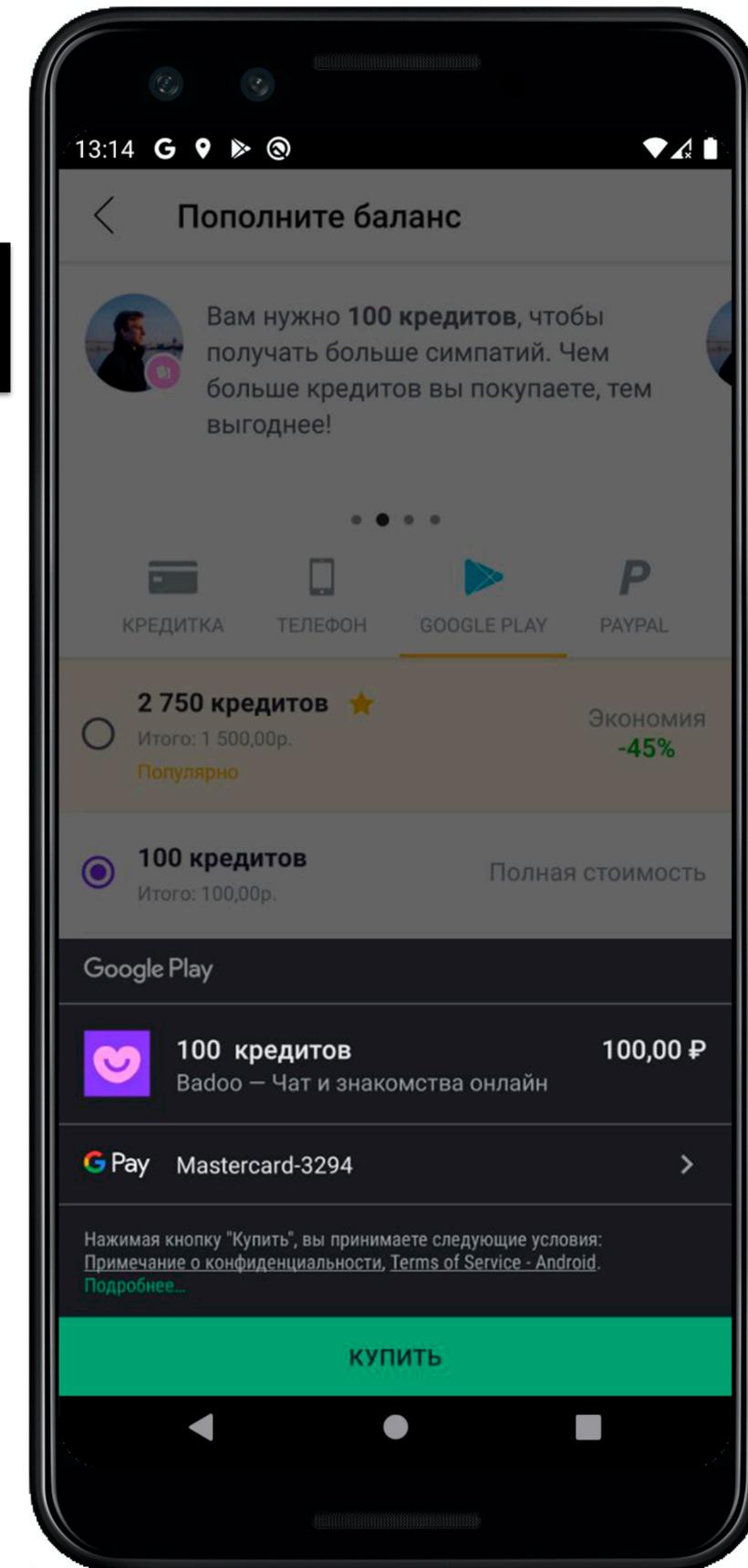
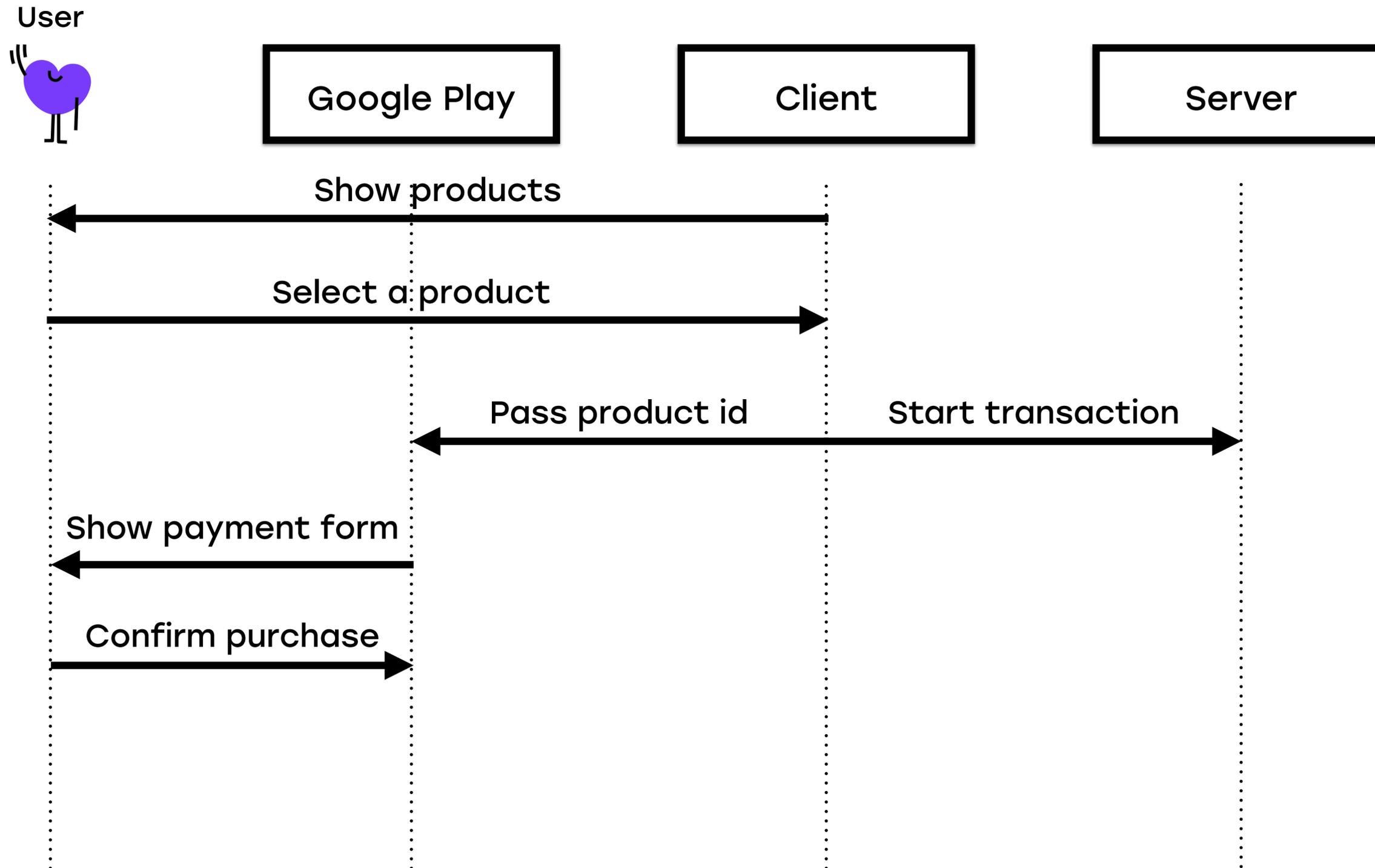
Google Play Процесс покупки



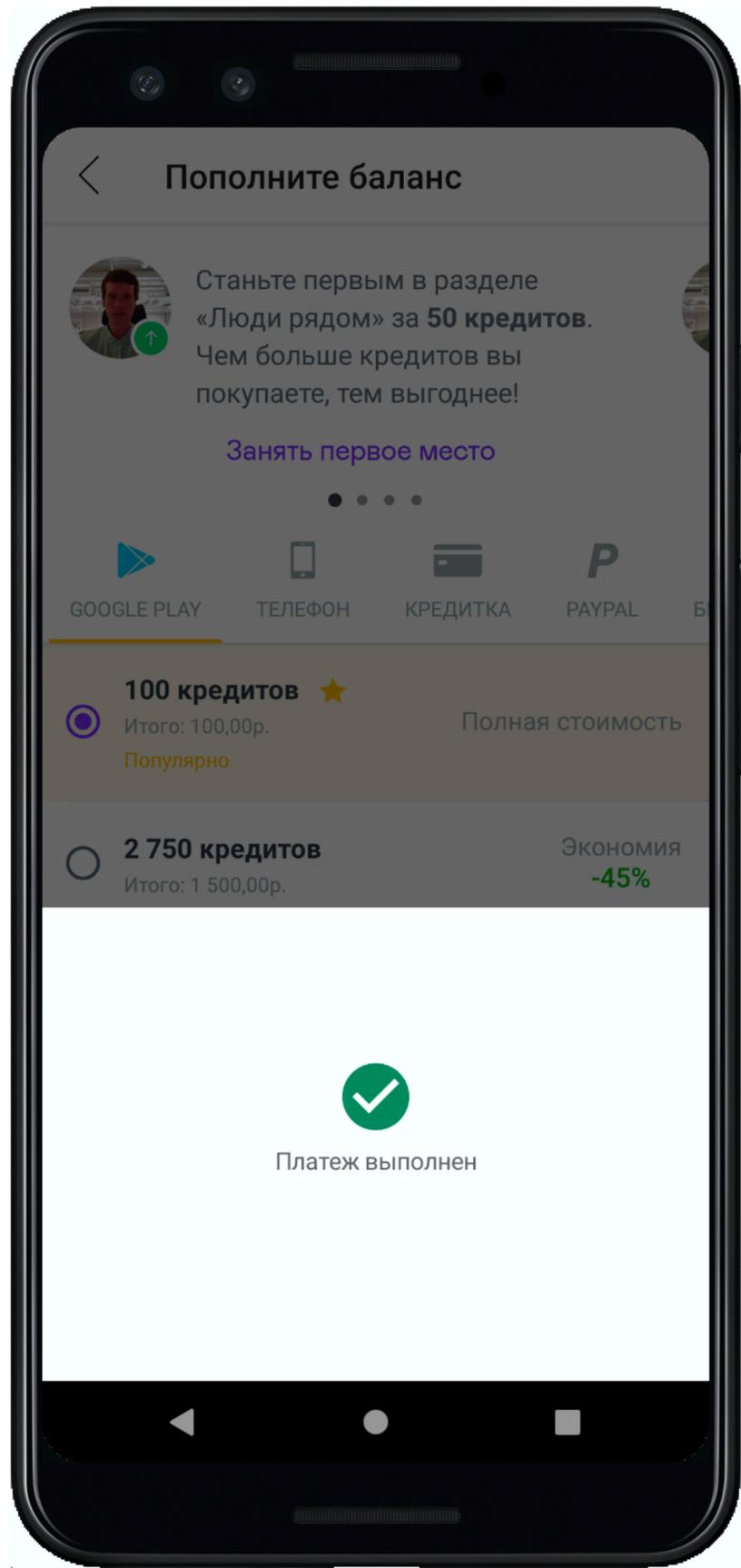
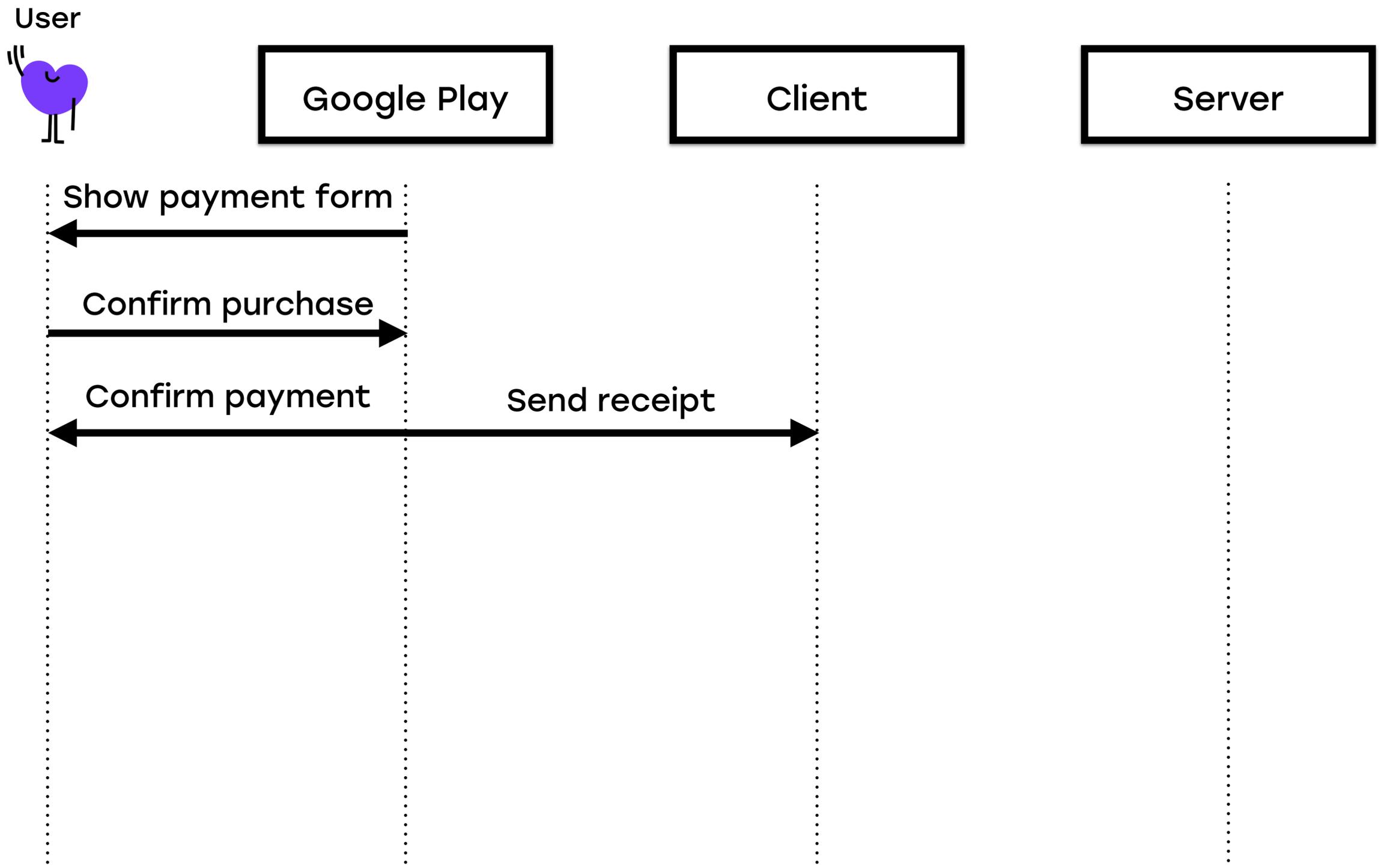
Google Play Процесс покупки



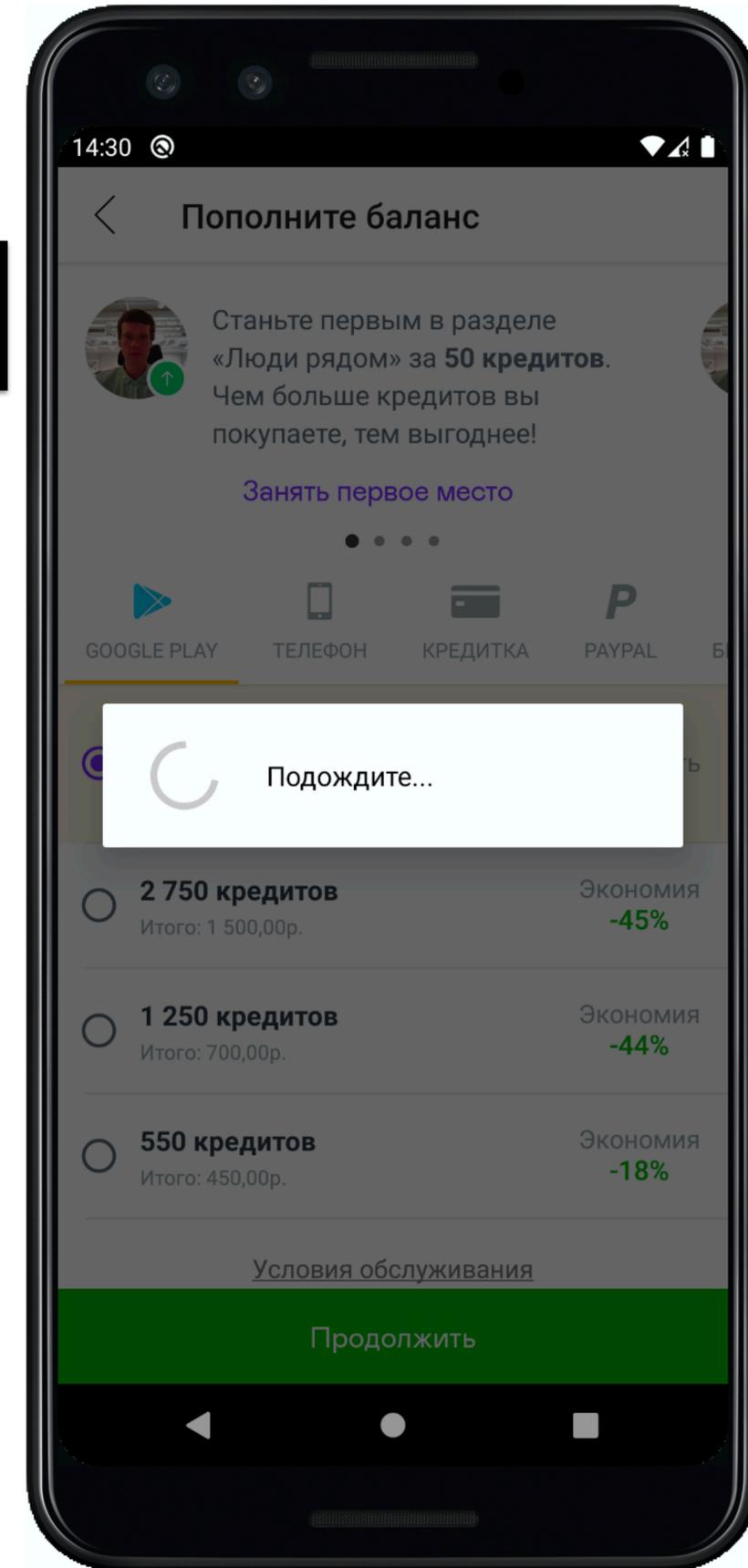
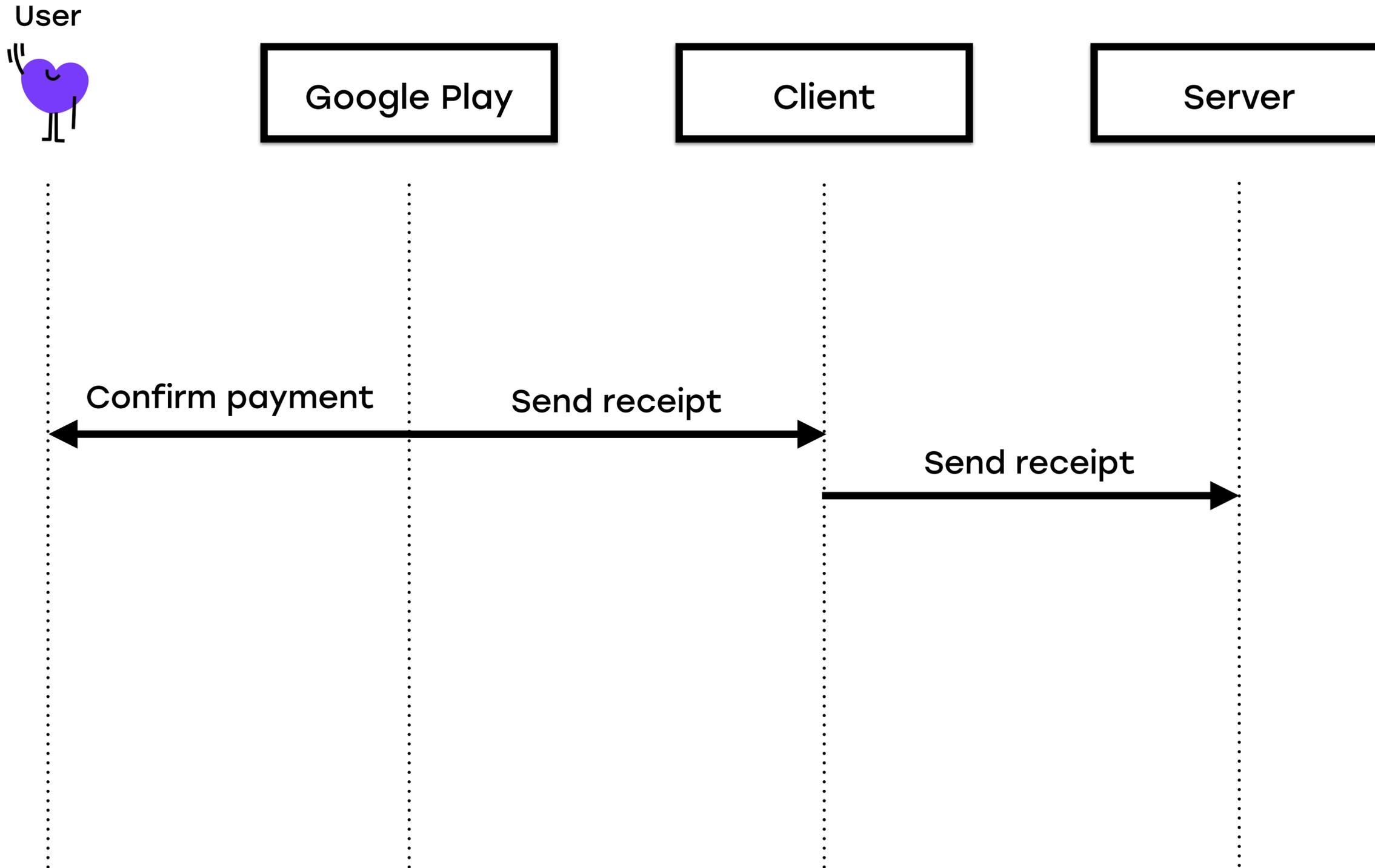
Google Play Процесс покупки



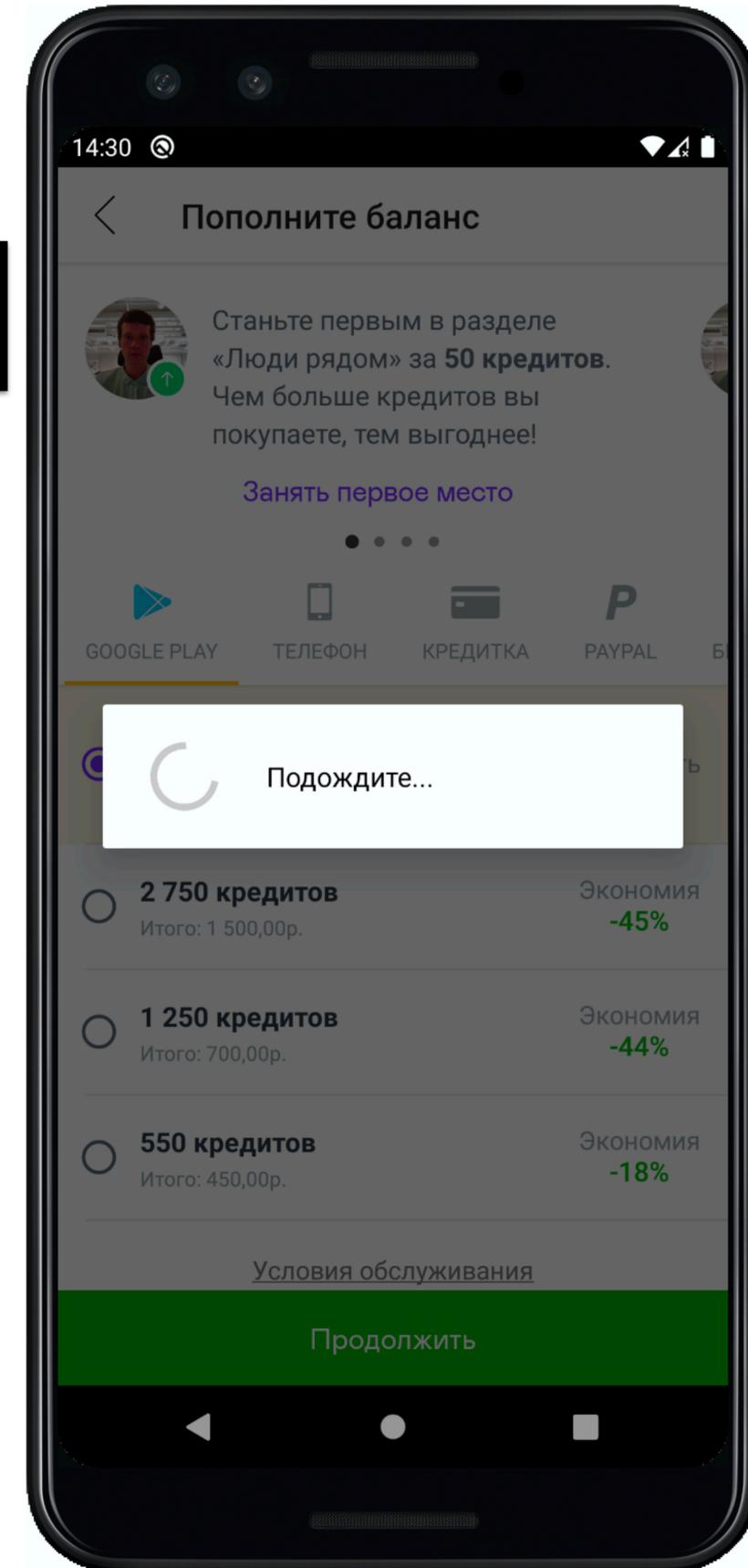
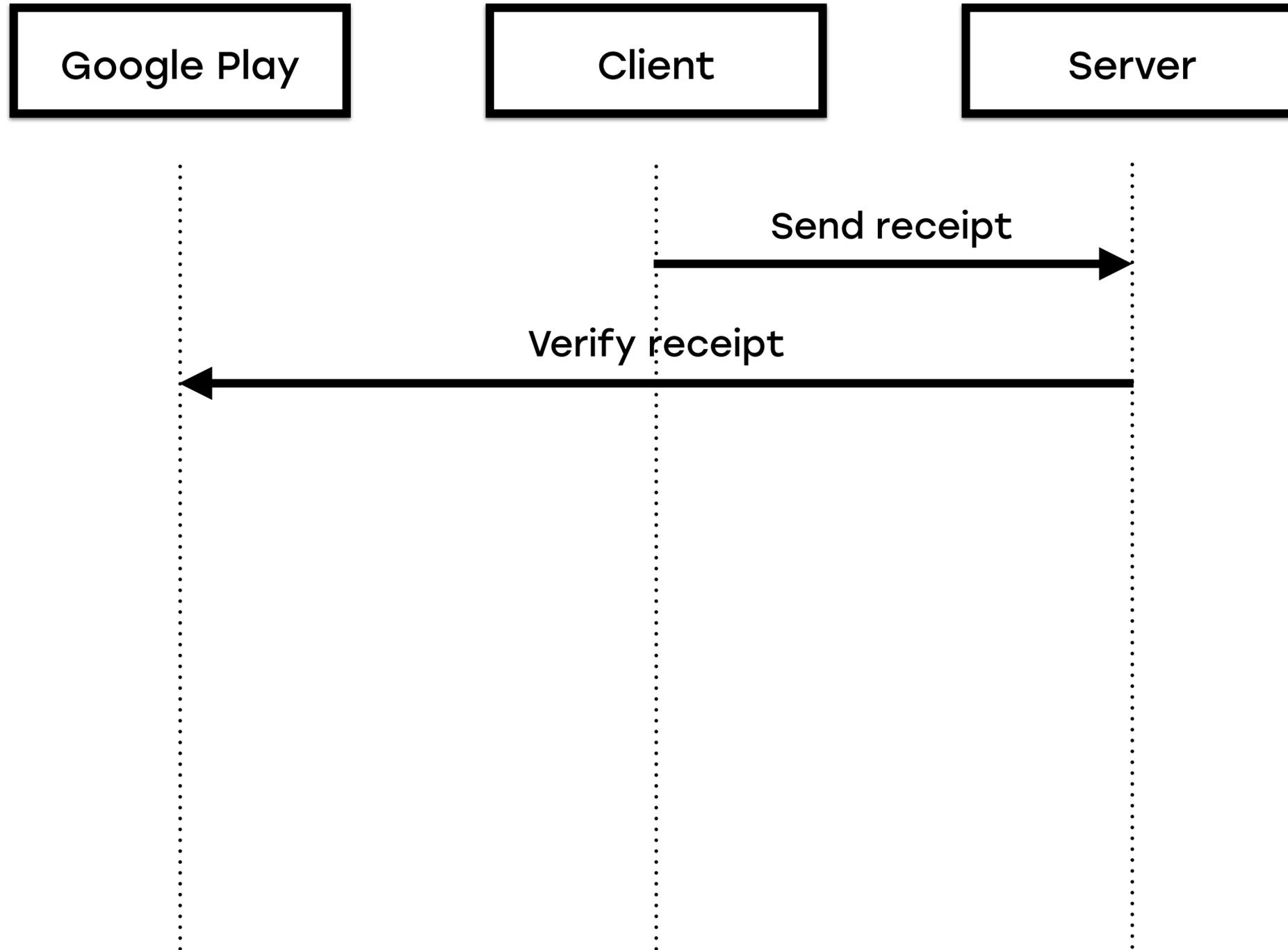
Google Play Процесс покупки



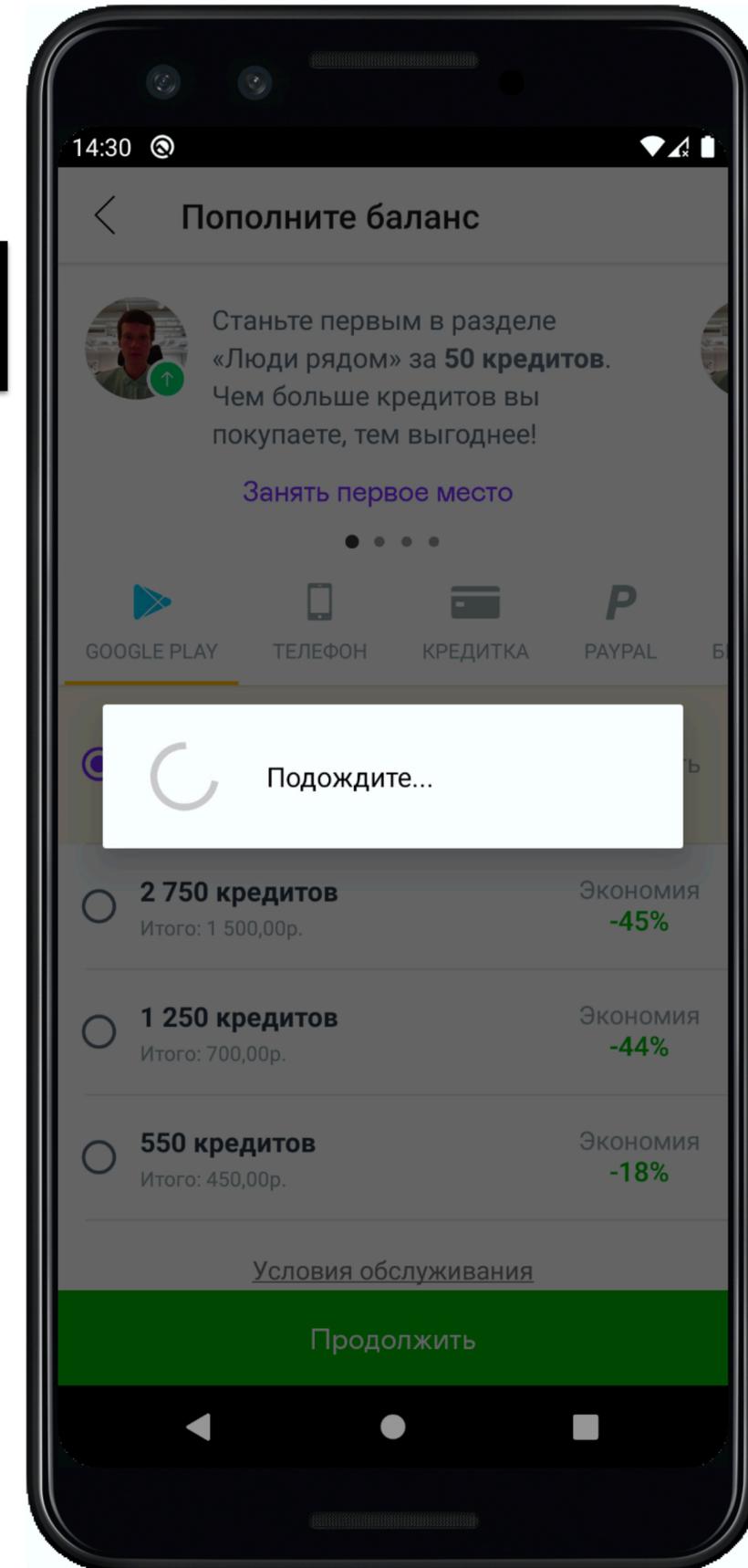
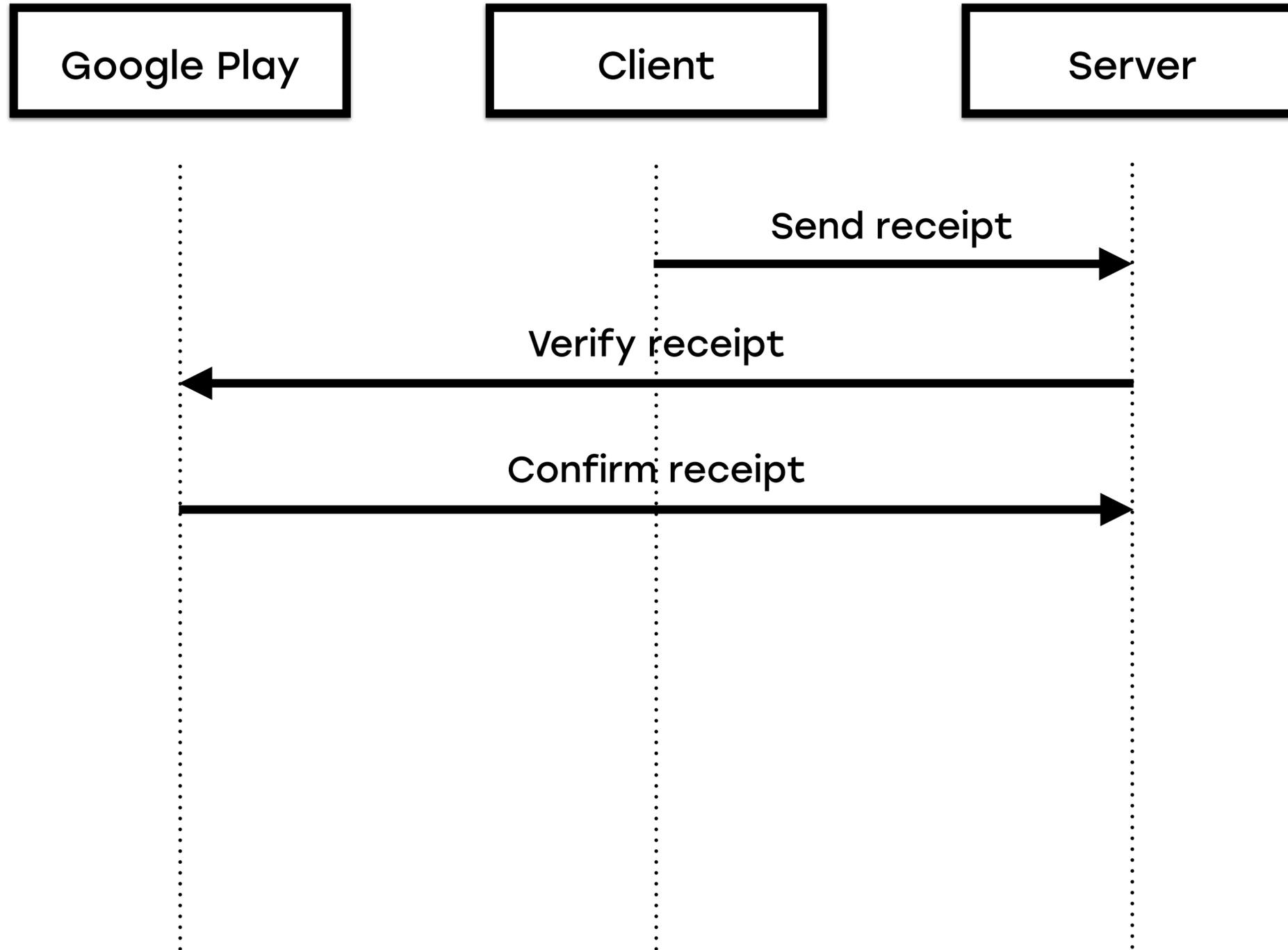
Google Play Процесс покупки



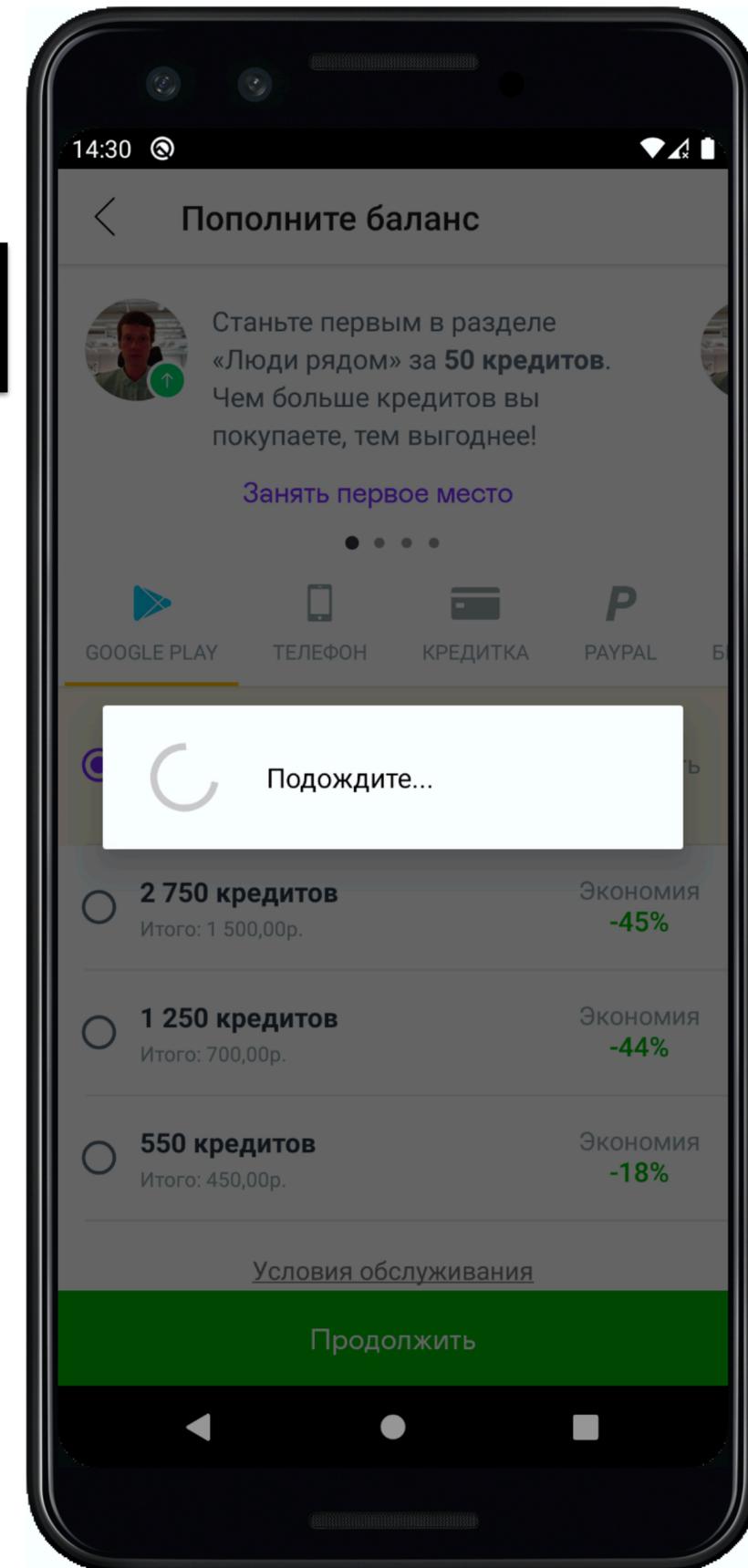
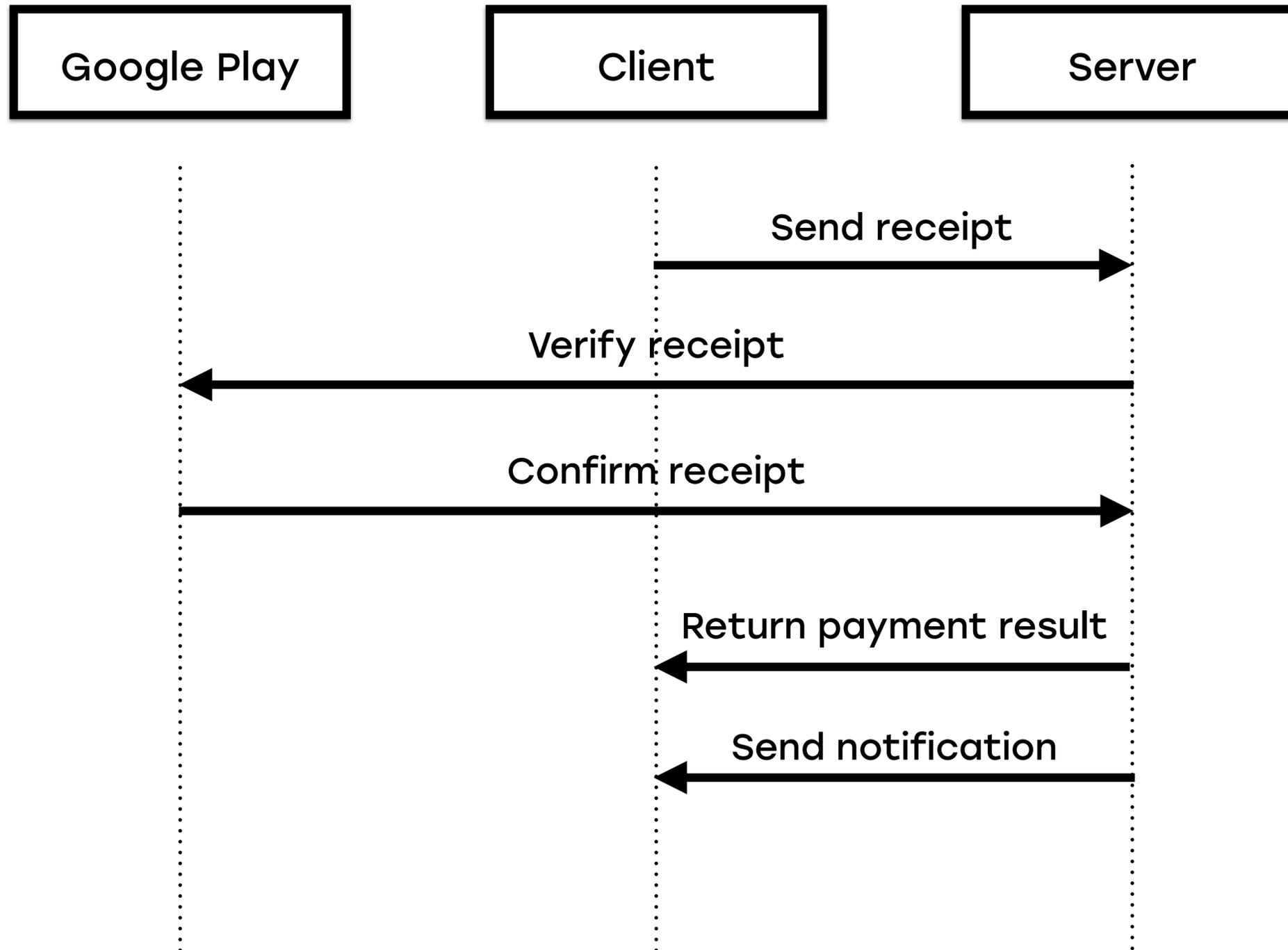
Google Play Процесс покупки



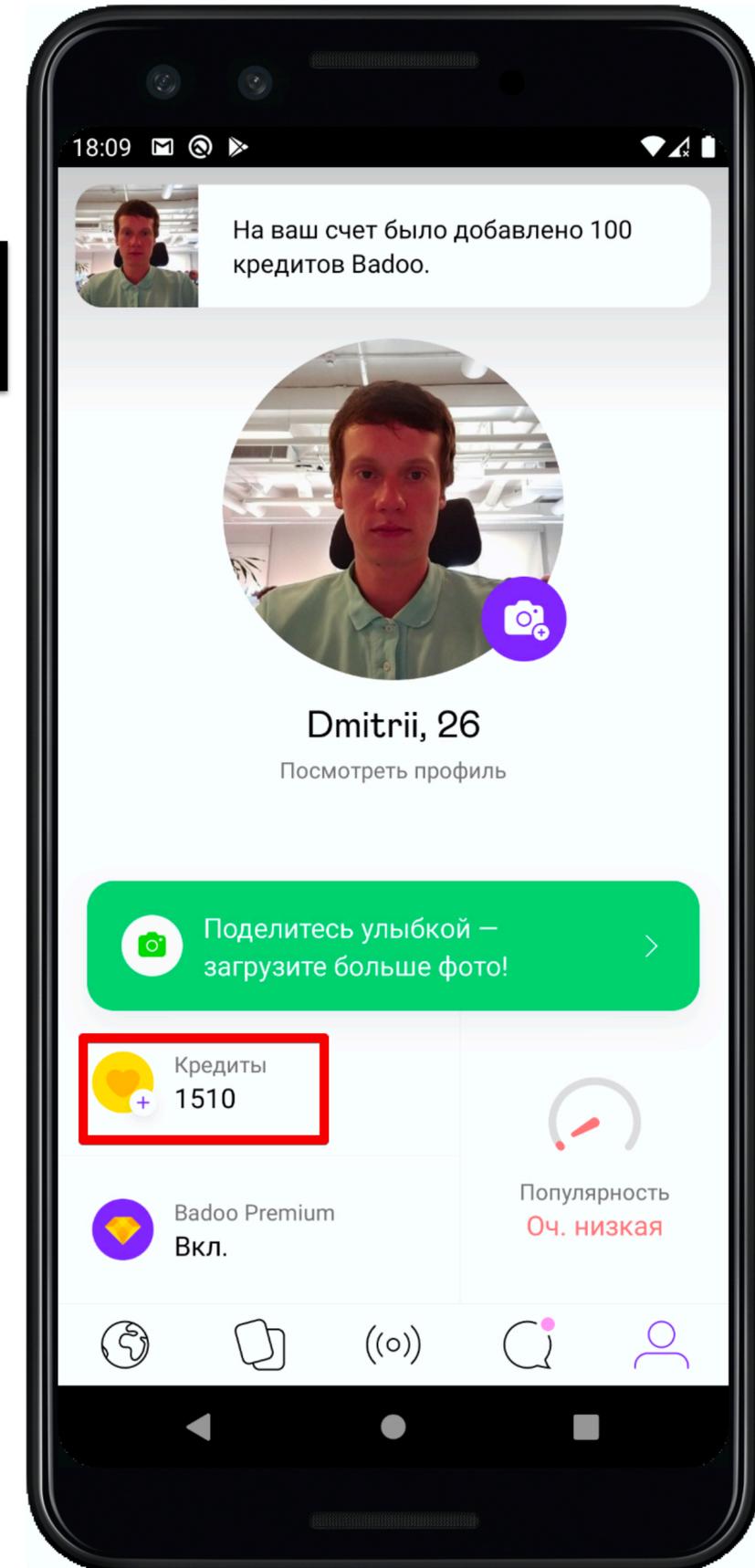
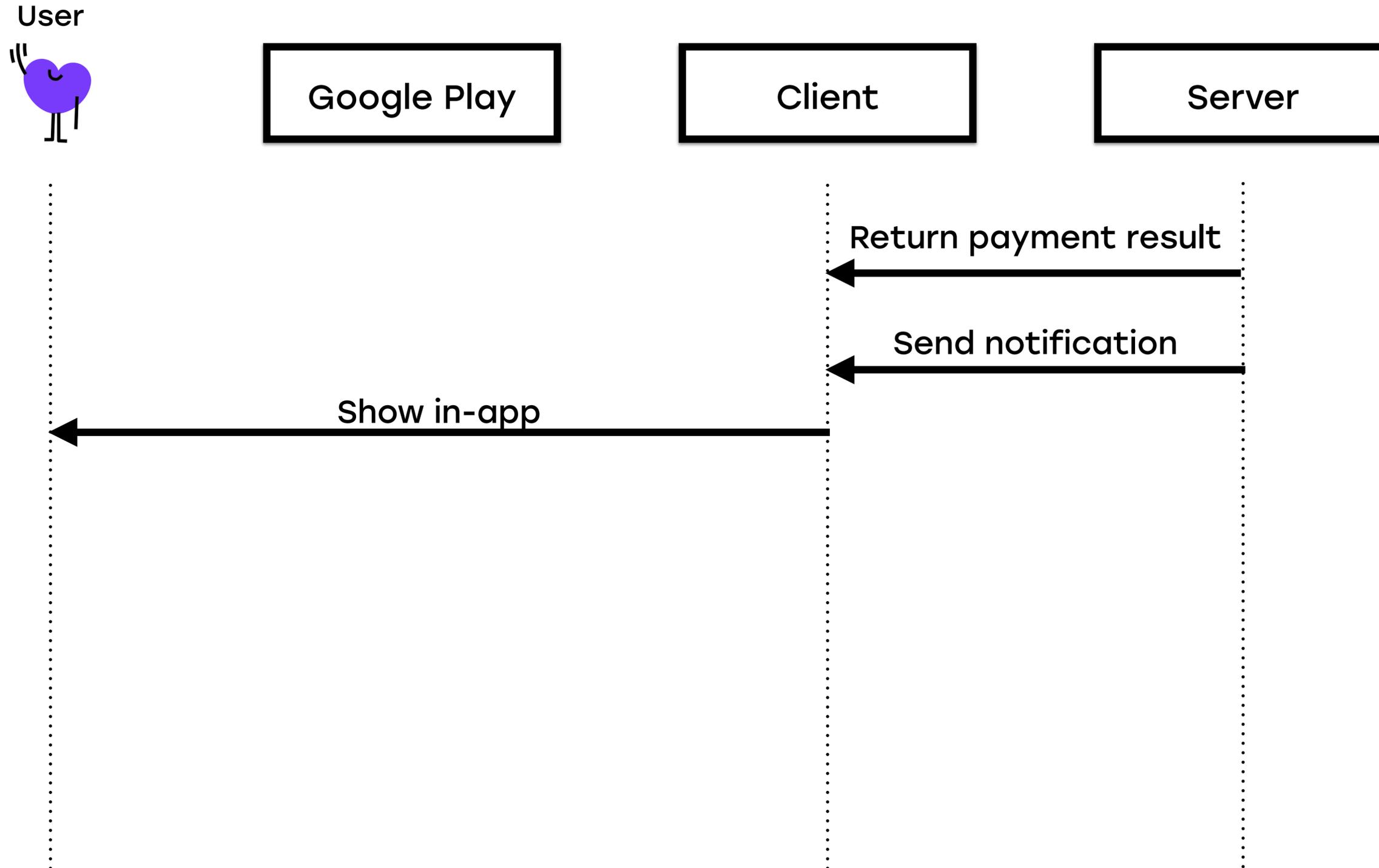
Google Play Процесс покупки



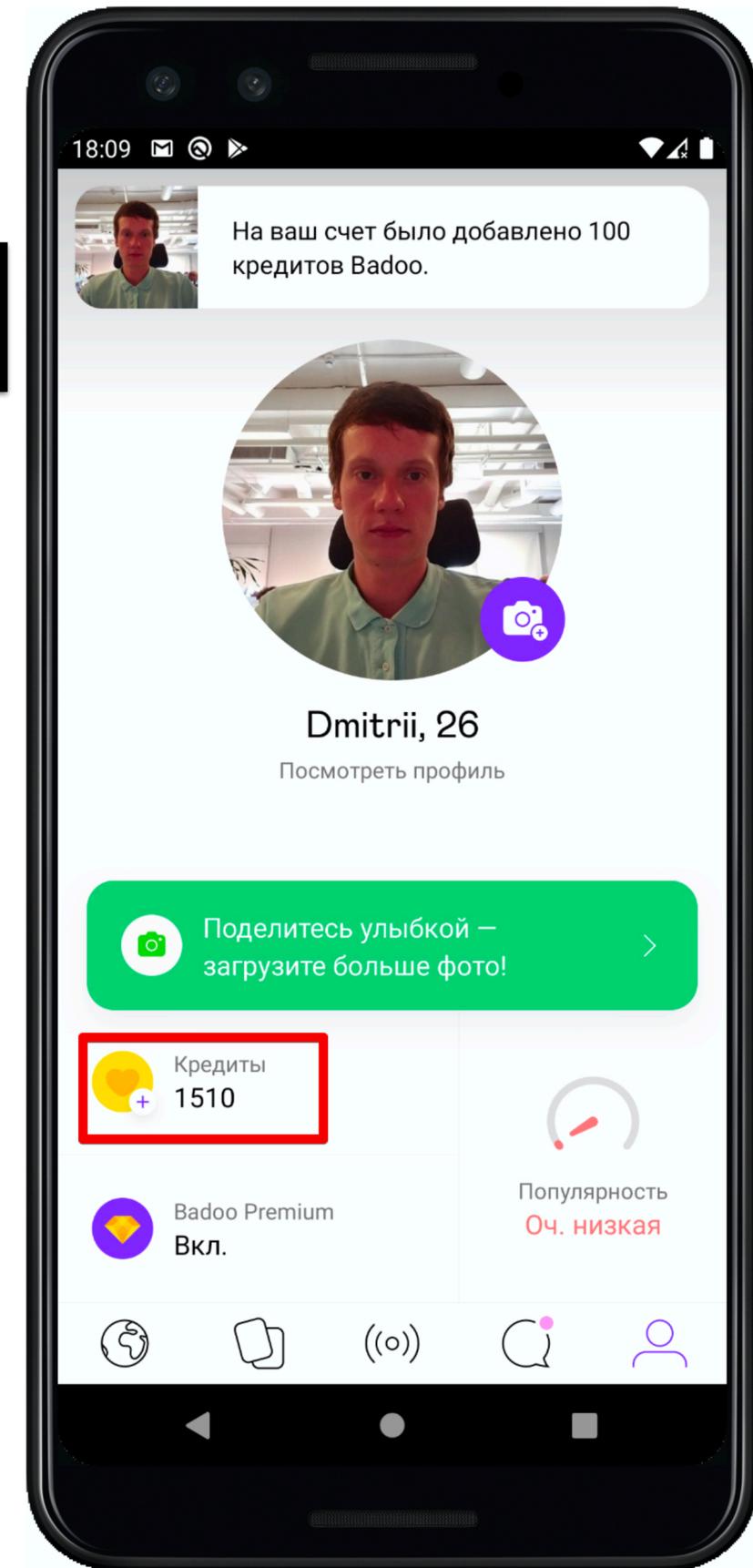
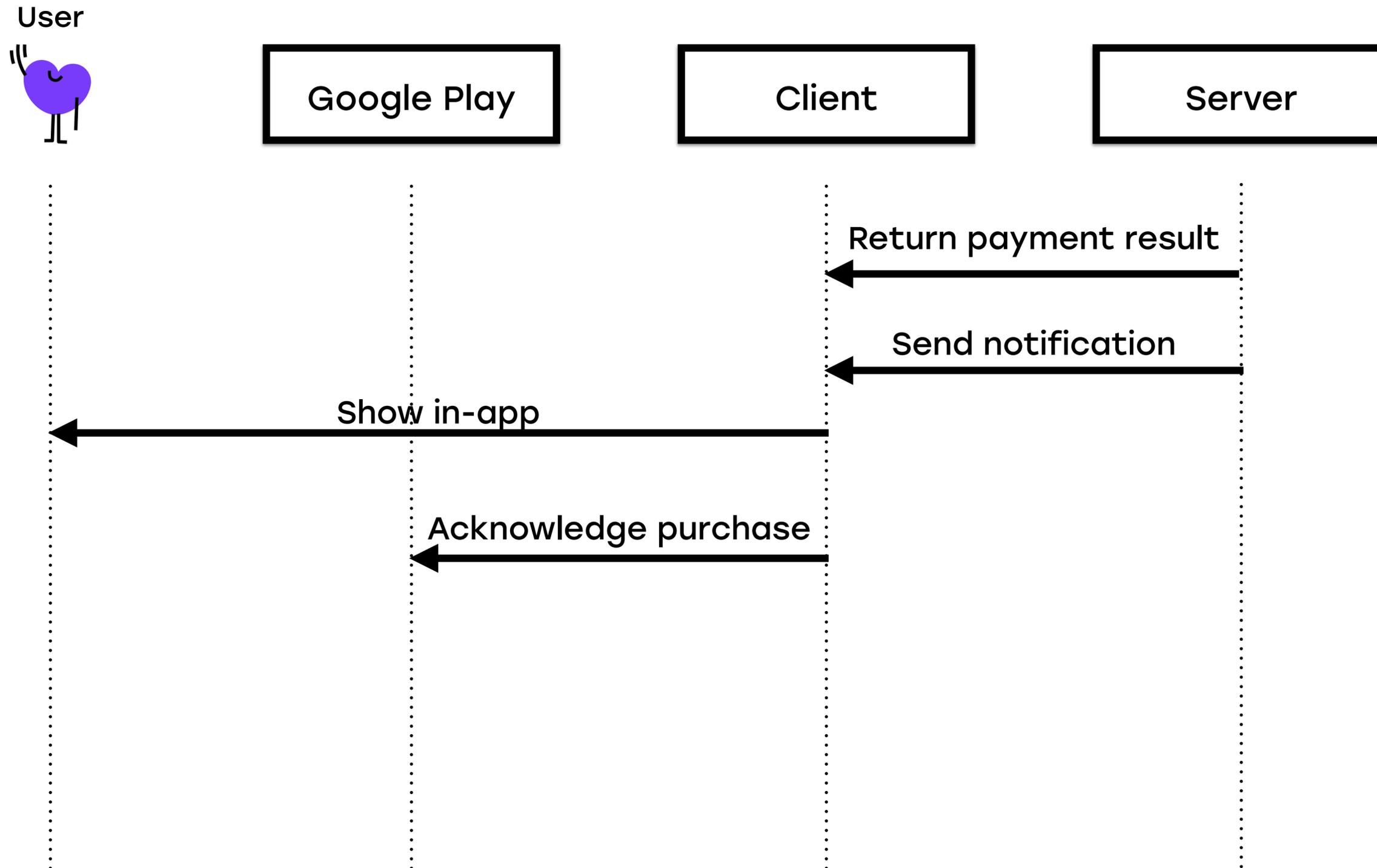
Google Play Процесс покупки



Google Play Процесс покупки



Google Play Процесс покупки



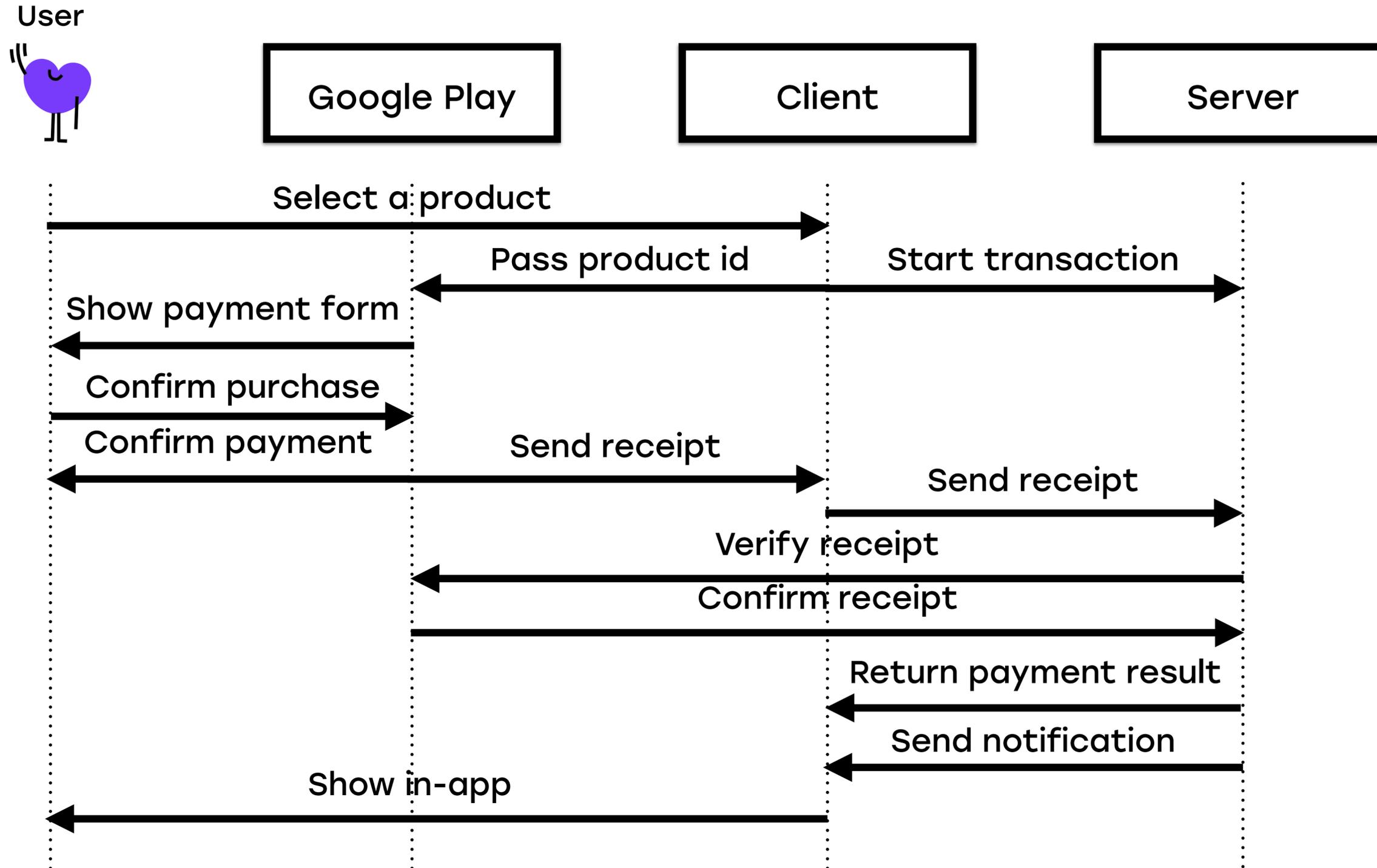
Google Play

Тестовое окружение

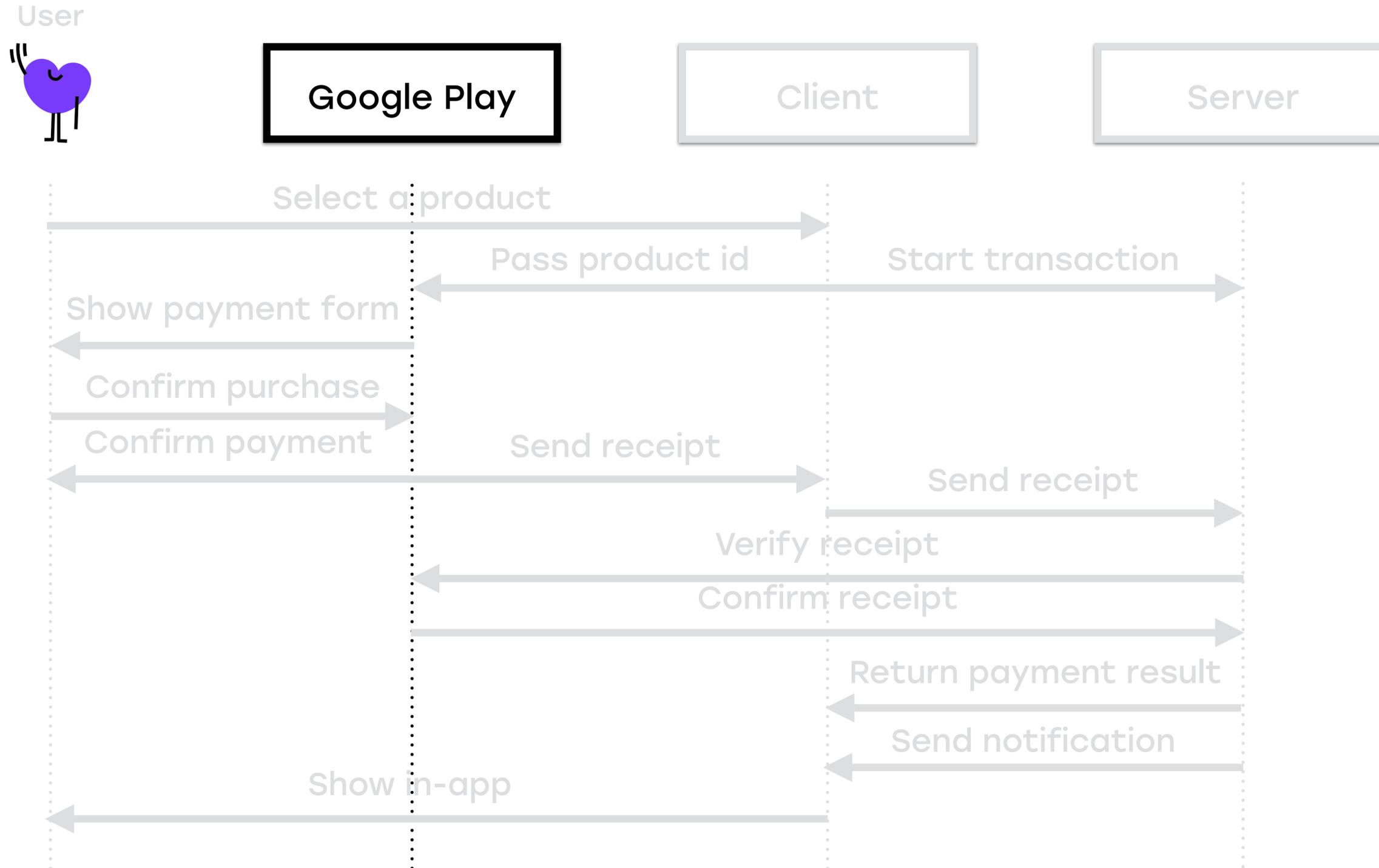
Почему песочницы

- + Процесс покупки максимально близок к реальному
- + Дешево
- Есть не у всех провайдеров

Google Play Тестовое окружение



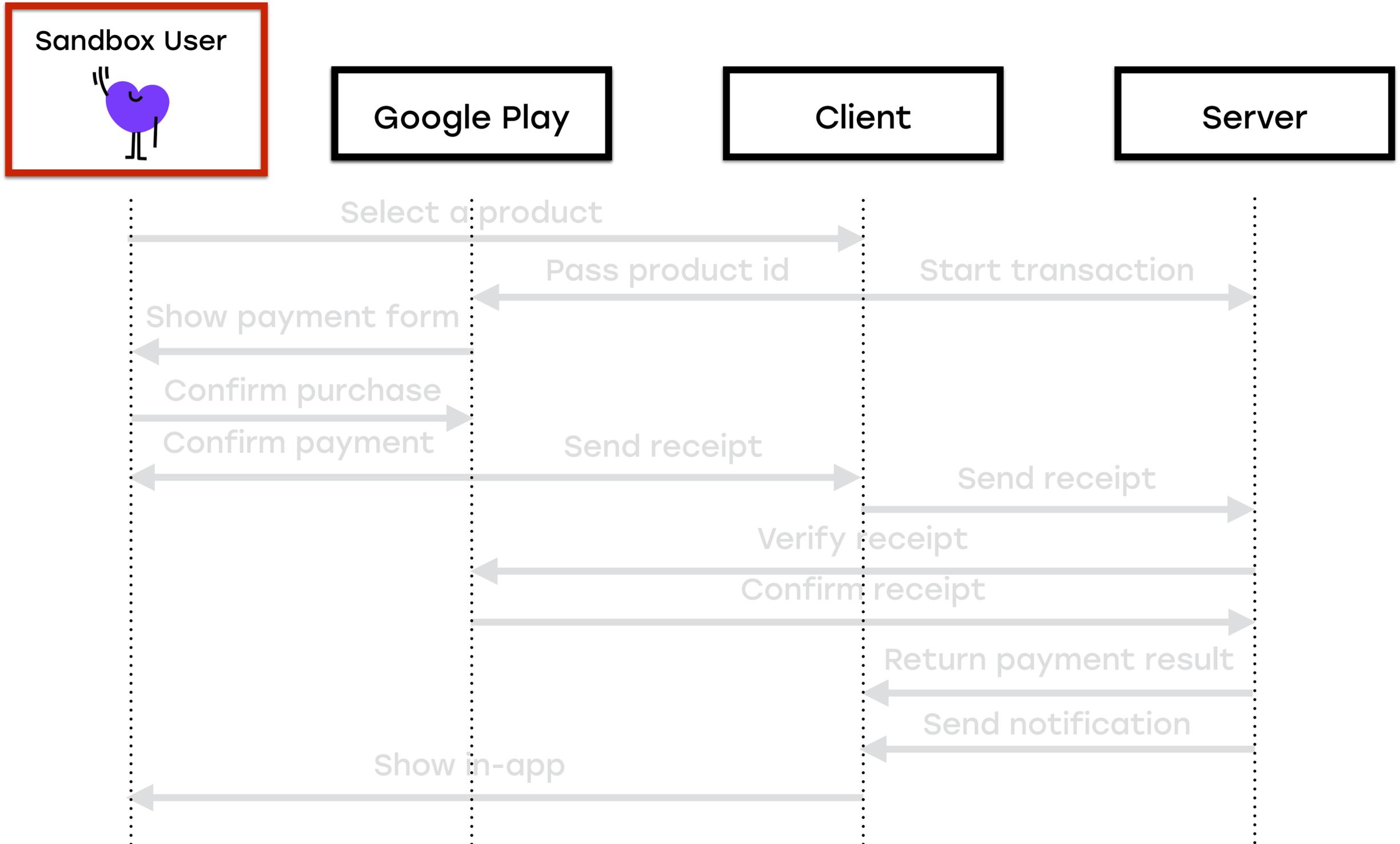
Google Play Тестовое окружение



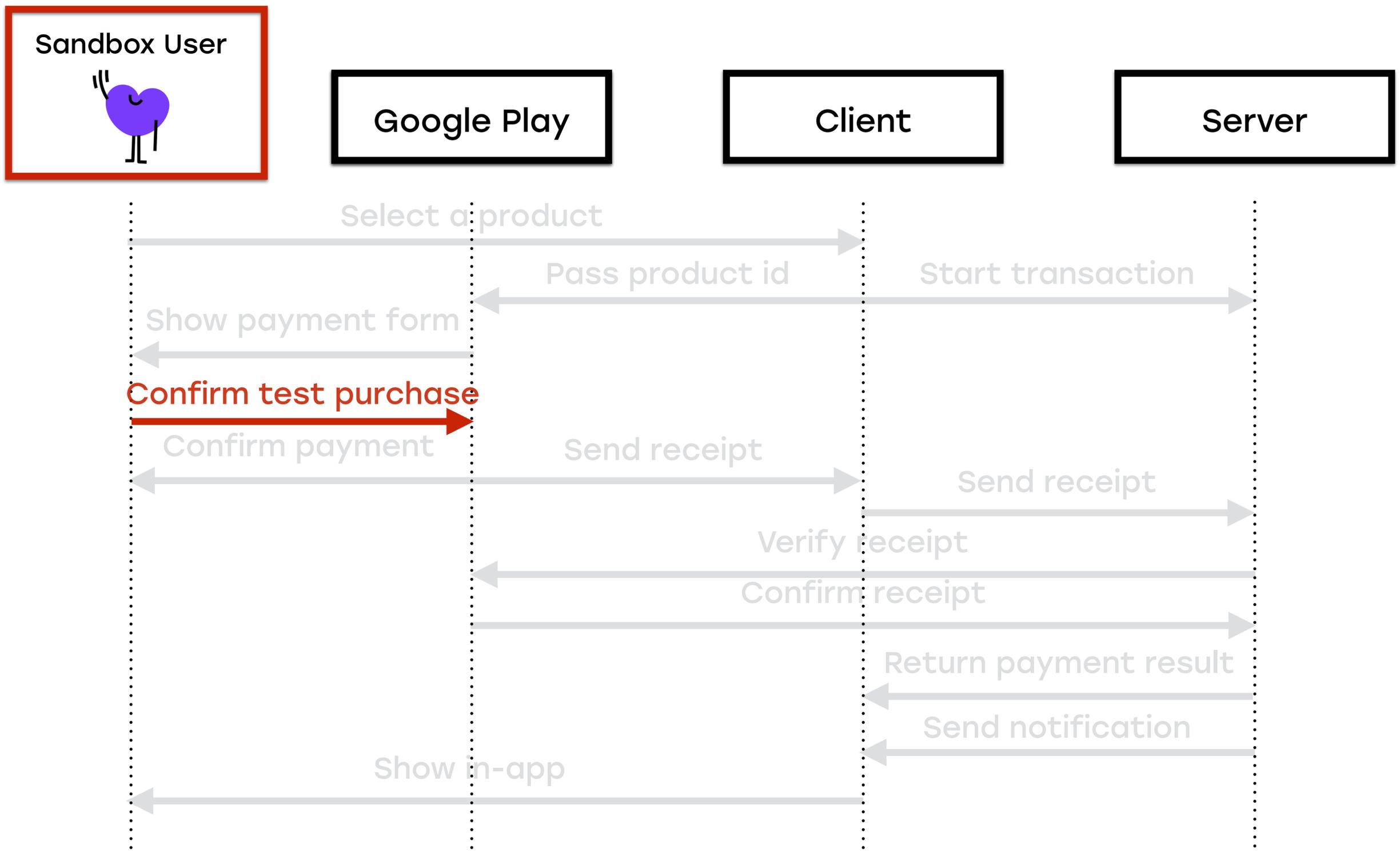
Google Play Sandbox

- **Добавить Google account в список тестовых через Google Play Developer Console**

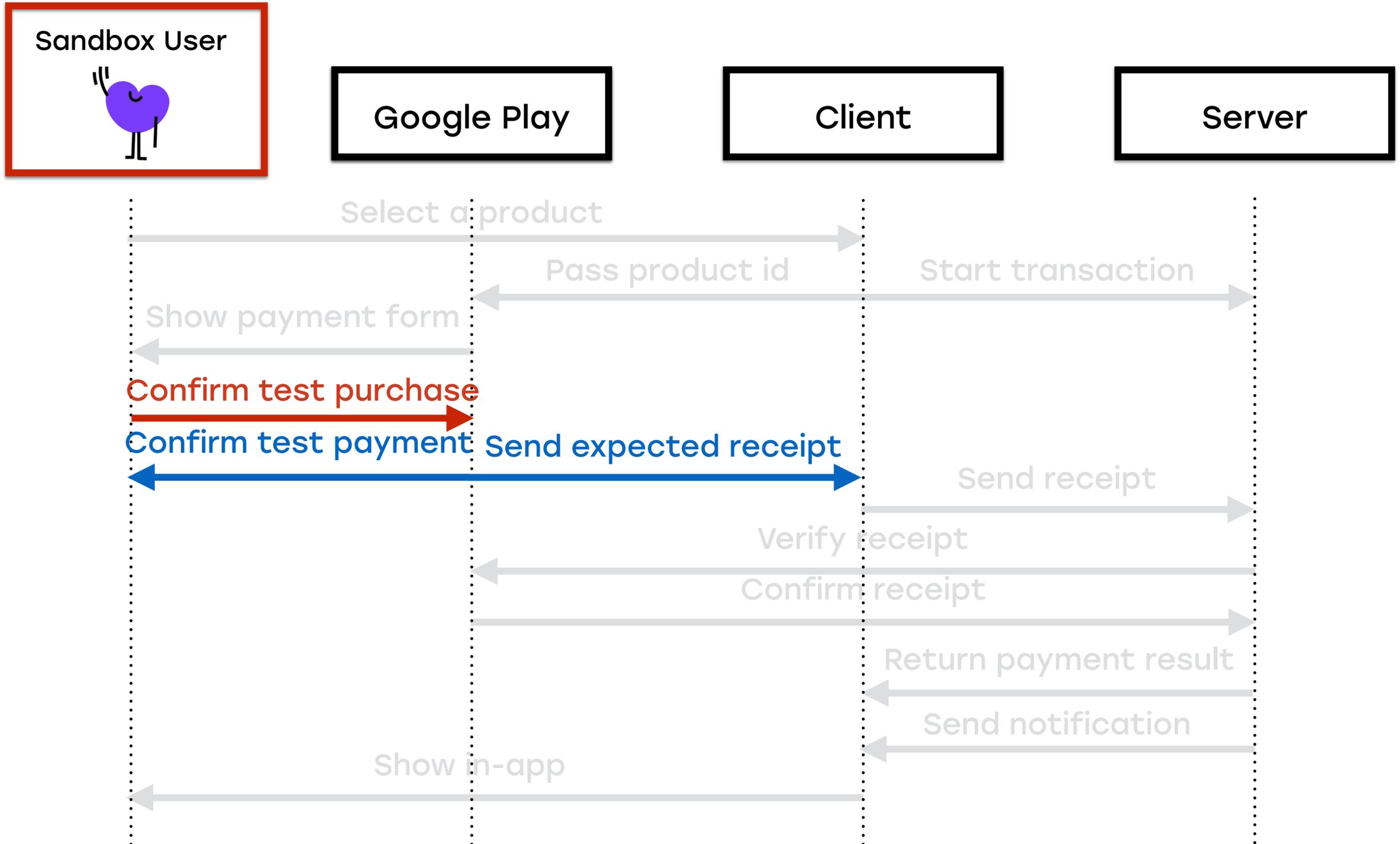
Google Play Тестовое окружение



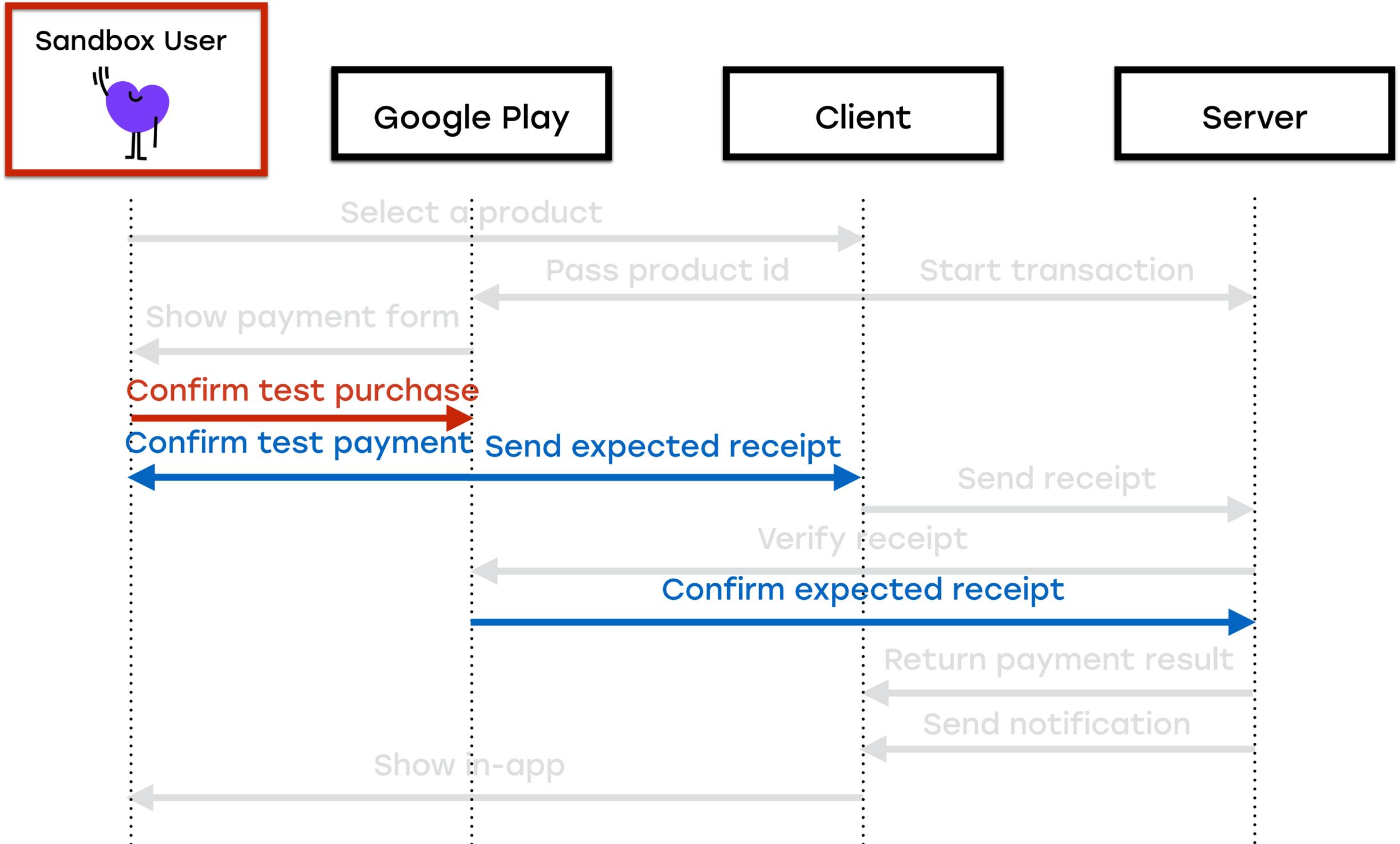
Google Play Тестовое окружение



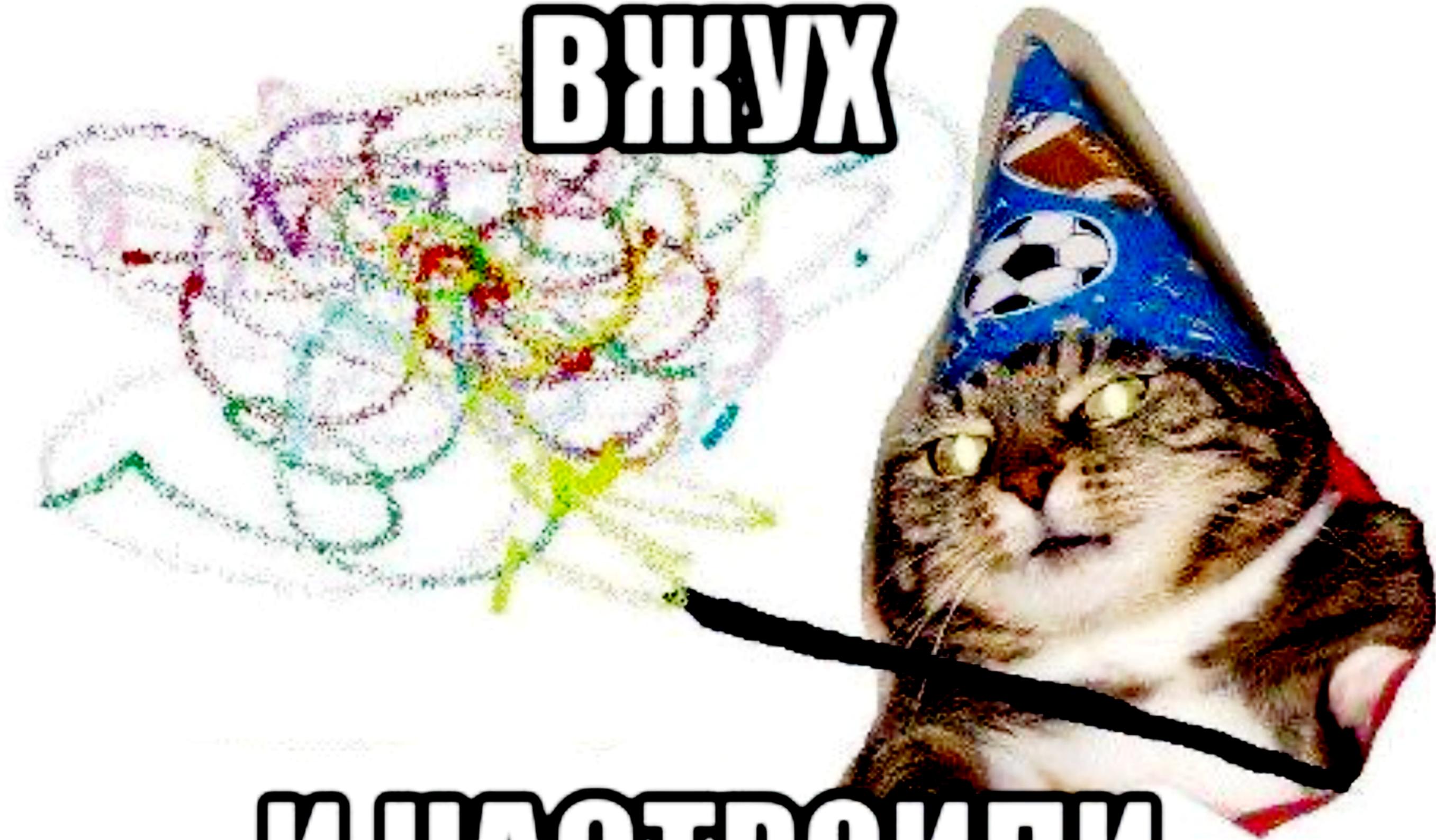
Google Play Тестовое окружение



Google Play Тестовое окружение



ВЖУХ



И НАСТРОИЛИ

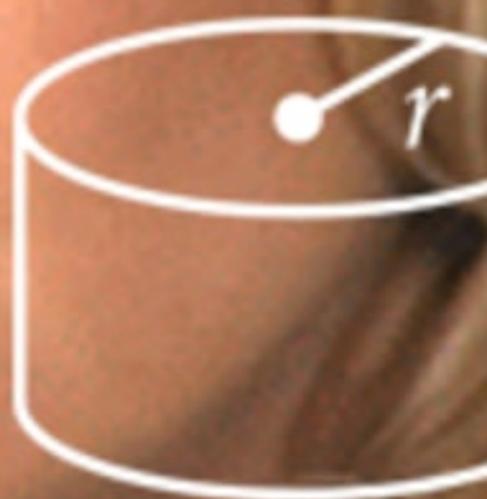
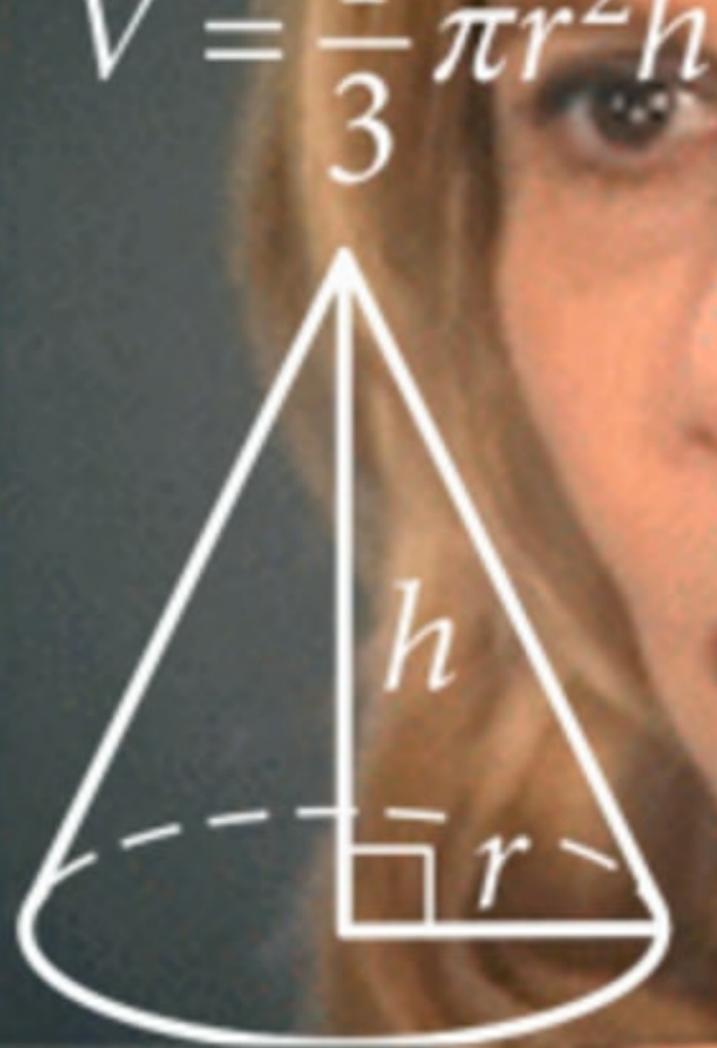
Google Play

АВТОМАТИЗАЦИЯ



$$A = \pi r^2$$

$$C = 2\pi r$$



$$V = \pi r^2 h$$

	30°	45°	60°
sin	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$
cos	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$
tan	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$

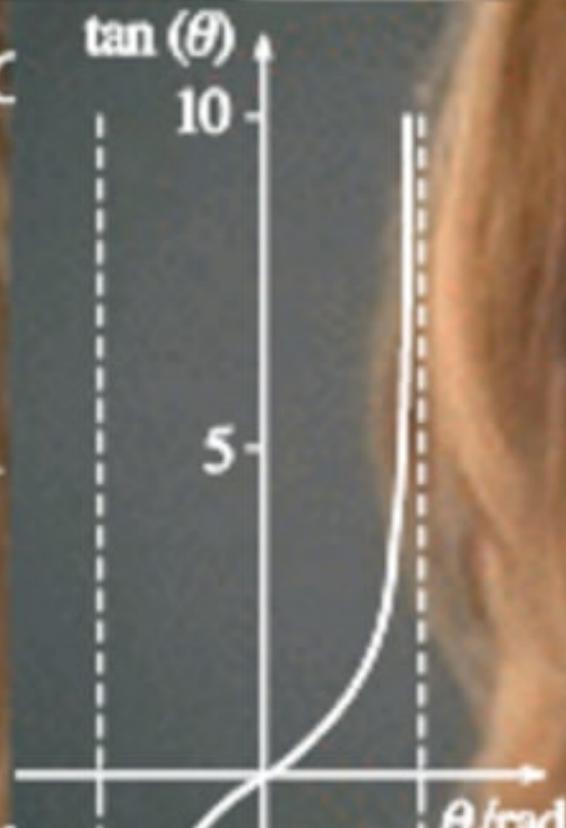


$$\int \sin x dx = -\cos x + C$$

$$\int \frac{dx}{\cos^2 x} = \tan x + C$$

$$\int \tan x dx = -\ln|\cos x| + C$$

$$\int \frac{dx}{\sin x} = \ln\left|\tan \frac{x}{2}\right| + C$$



$$ax^2 + bx + c = 0$$

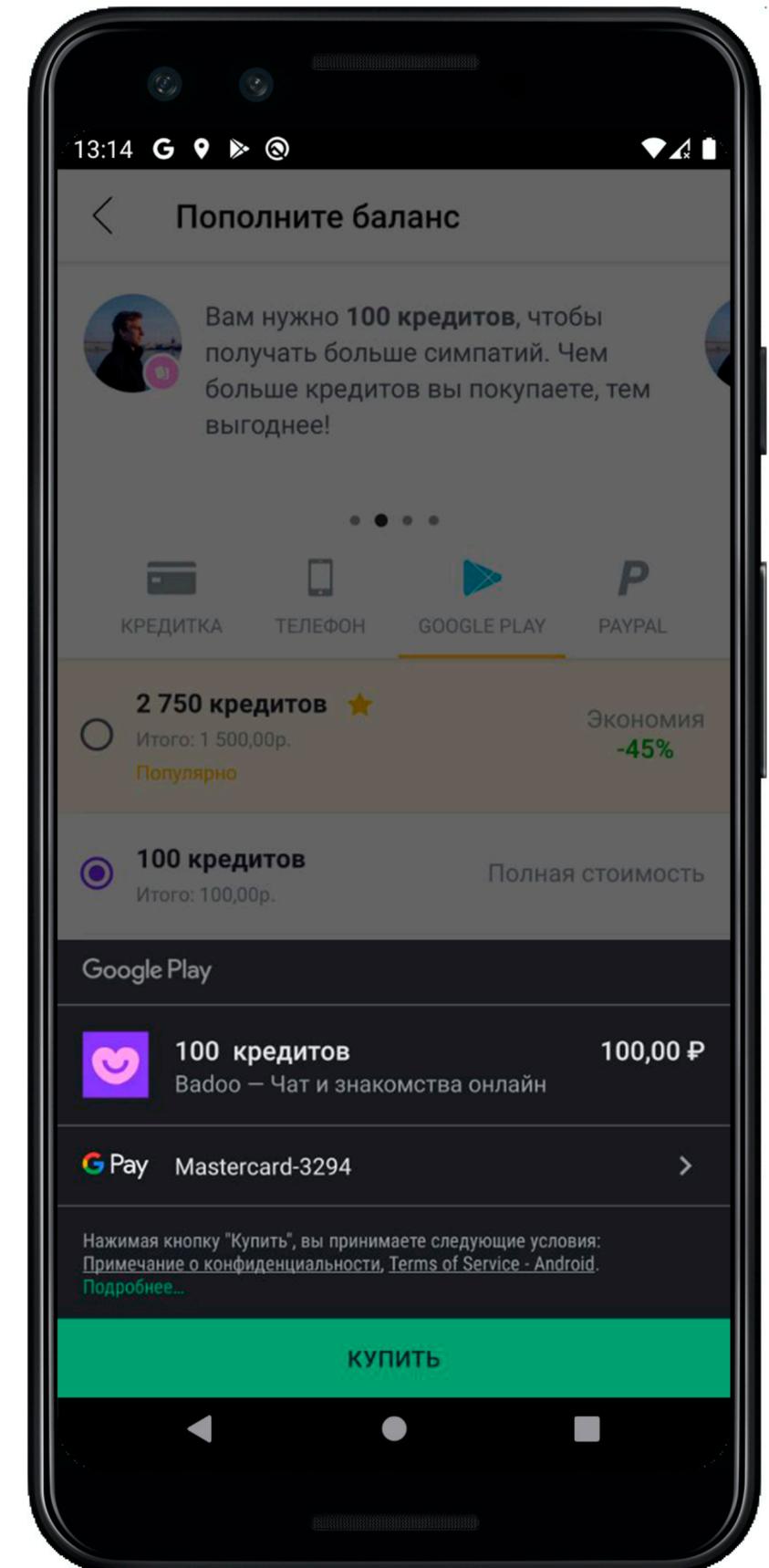
$$a\left(x^2 + \frac{b}{a}x + \frac{c}{a}\right) = 0$$

Путь к стабильным Google Play тестам



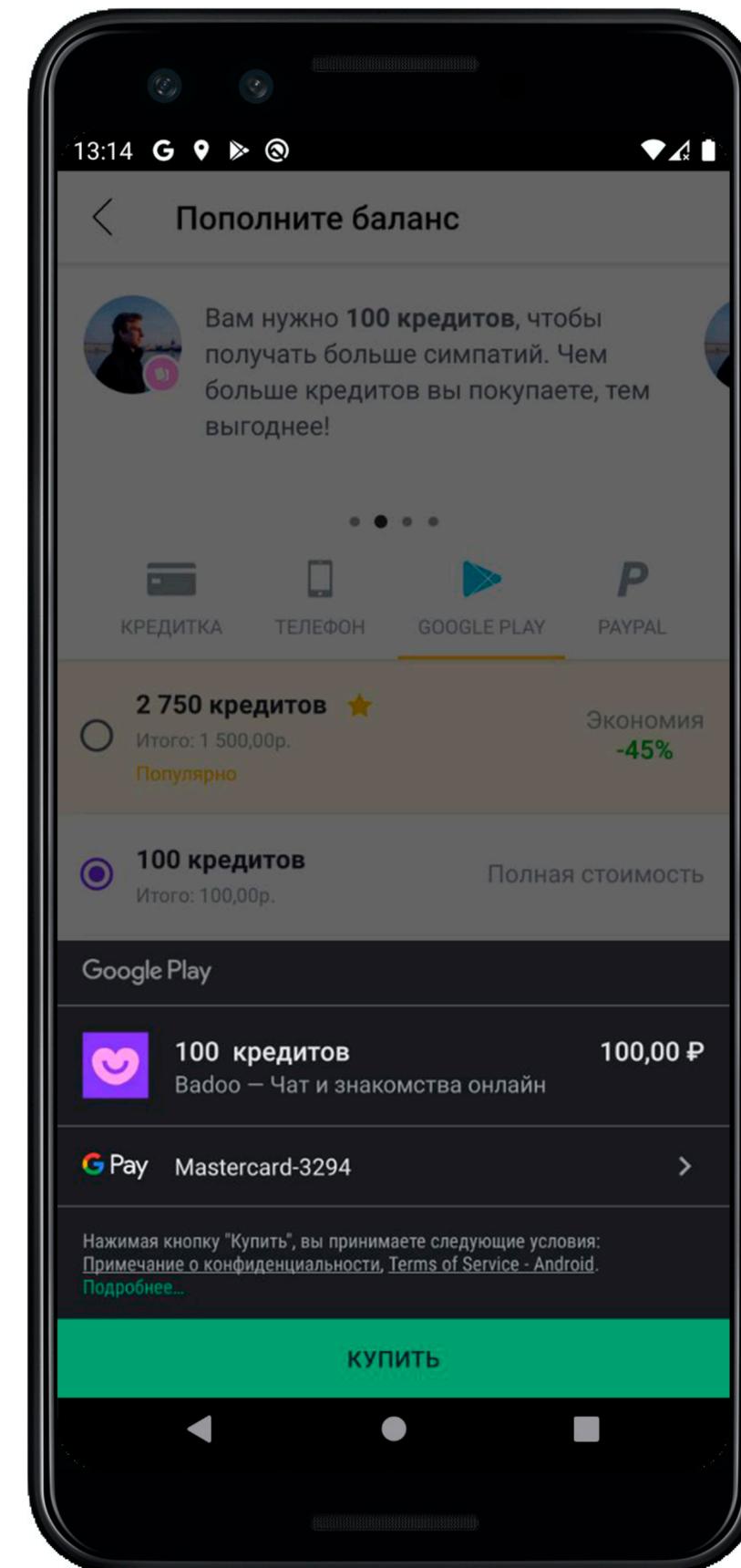
Внешние диалоги

- Диалоги оплаты от Google Play SDK



Внешние диалоги

- Диалоги оплаты от Google Play SDK
- Нет доступа через Calabash



UI Automator

UI testing framework suitable for
cross-app
functional UI testing across system
and installed apps

<https://developer.android.com/training/testing/ui-automator>

Calabash vs Appium

	Calabash	Appium
Внешние диалоги	-	+

Calabash vs Appium

	Calabash	Appium
Внешние диалоги	-	+
Стабильность	+	-

Calabash vs Appium

	Calabash	Appium
Внешние диалоги	-	+
Стабильность	+	-
Бэkdоры	+	-

Внешние диалоги. Решение

Calabash

+

UIAutomator2 driver

<https://github.com/badoo/calabash-android-server/tree/uiautomator2>

Calabash + UIAutomator2

+ **HEISENBUG**
// 2017 Moscow

Rajdeep Varma
Badoo

Automate the impossible:
Blending the best
of Android drivers



<https://www.youtube.com/watch?v=dKHudzto4TM>

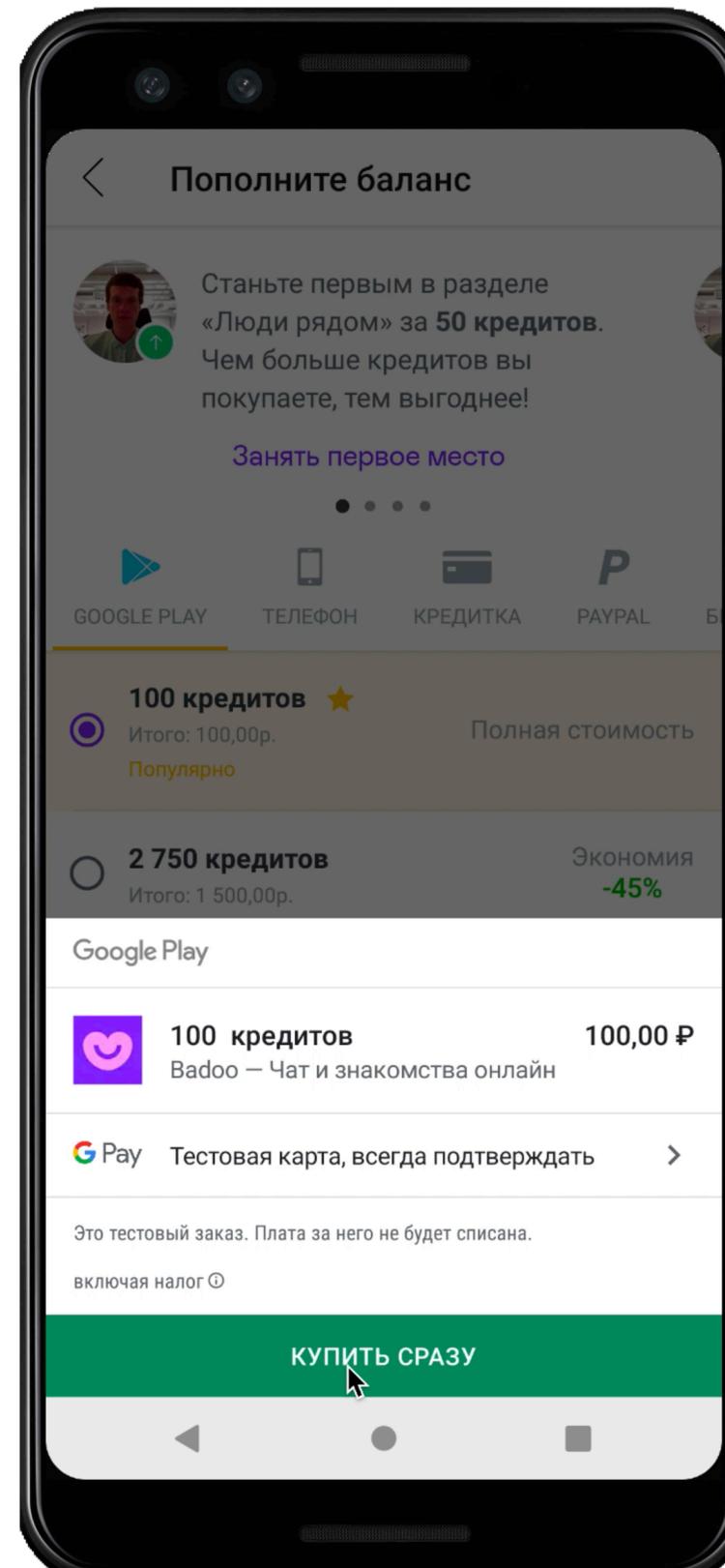
Внешние диалоги

- “uia_” методы

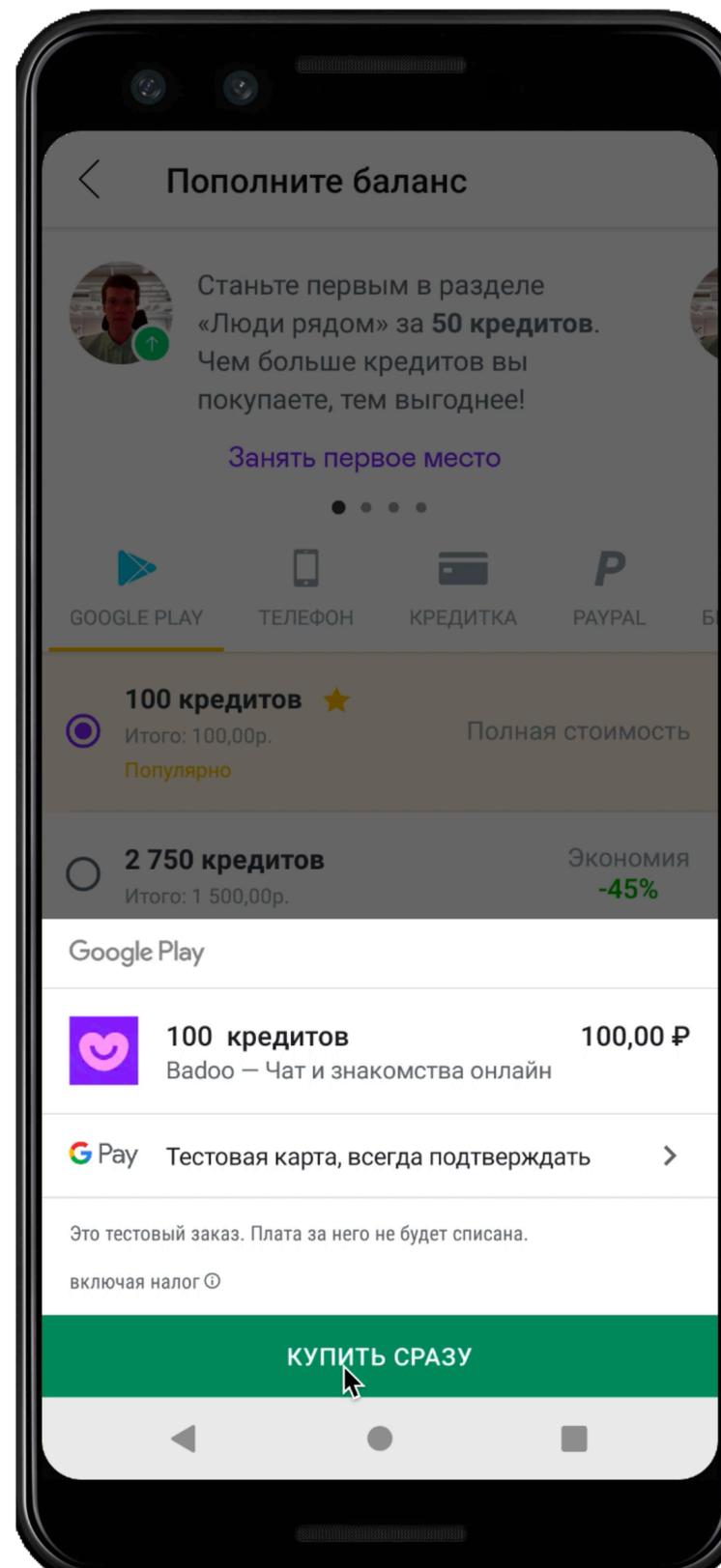
Внешние диалоги

- “uia_” методы
- Выполняются с помощью UIAutomator

Платеж



Платеж



Платеж

```
def current_page?  
  screen_texts.any? { |text| text.include?('will not be charged') || text.include?('VISA xxx-FAKE') }  
end
```

Платеж

```
def current_page?  
  screen_texts.any? { |text| text.include?('will not be charged') || text.include?('VISA xxx-FAKE') }  
end
```

ACTION_BUTTON = 'com.android.vending:id/footer1'

```
def tap_on_submit  
  uia_touch_by_id(ACTION_BUTTON)  
end
```

Отмена подписки

- Подписка привязана к Google аккаунту

Отмена подписки

- Подписка привязана к Google аккаунту
- **Остается активной после завершения теста**

Отмена подписки

- Подписка привязана к Google аккаунту
- Остается активной после завершения теста
- При наличии действующей подписки нельзя купить новую

Отмена подписки. Термины

- [QaApi](#) – API для изменения внутреннего состояния системы в процессе тестирования приложения

Отмена подписки. Термины

- [QaApi](#) – API для изменения внутреннего состояния системы в процессе тестирования приложения
- Добавить кредитов

Отмена подписки. Термины

- [QaApi](#) – API для изменения внутреннего состояния системы в процессе тестирования приложения
 - Добавить кредитов
 - Включить/отключить Премиум

Отмена подписки. Термины

- [QaApi](#) – API для изменения внутреннего состояния системы в процессе тестирования приложения
 - Добавить кредитов
 - Включить/отключить Премиум
 - Привязать детали оплаты на тестовом окружении

Отмена подписки. Термины

- [QaApi](#) – API для изменения внутреннего состояния системы в процессе тестирования приложения
 - Добавить кредитов
 - Включить/отключить Премиум
 - Привязать детали оплаты на тестовом окружении
 - To be continued...

Отмена подписки. Термины

- Before/After hooks

Отмена подписки. Термины

- Before/After hooks
 - **Before** – методы для подготовки тестового окружения

Отмена подписки. Термины

- Before/After hooks
 - Before – методы для подготовки тестового окружения
 - **After – методы для приведения тестового окружения в первоначальное состояние**

Отмена подписки. Реализация

- After hook

Отмена подписки. Реализация

- *After hook*
- QaApi “userCancelGoogleWalletSubscriptions”

Отмена подписки. Реализация

- After hook
- QaApi “userCancelGoogleWalletSubscriptions”

```
if !@user_id.nil? && FeatureName.tagged?( '@google_play_subscription' )  
  QaApi::Billing.user_cancel_google_wallet_subscriptions(user_id: @user_id)  
end
```

Отмена подписки. Реализация

- QaApi “userCancelGoogleWalletSubscriptions”
- [Revoke subscription](#) запрос Google Play Developer API

Home > Products > Google Play Developer API > Reference



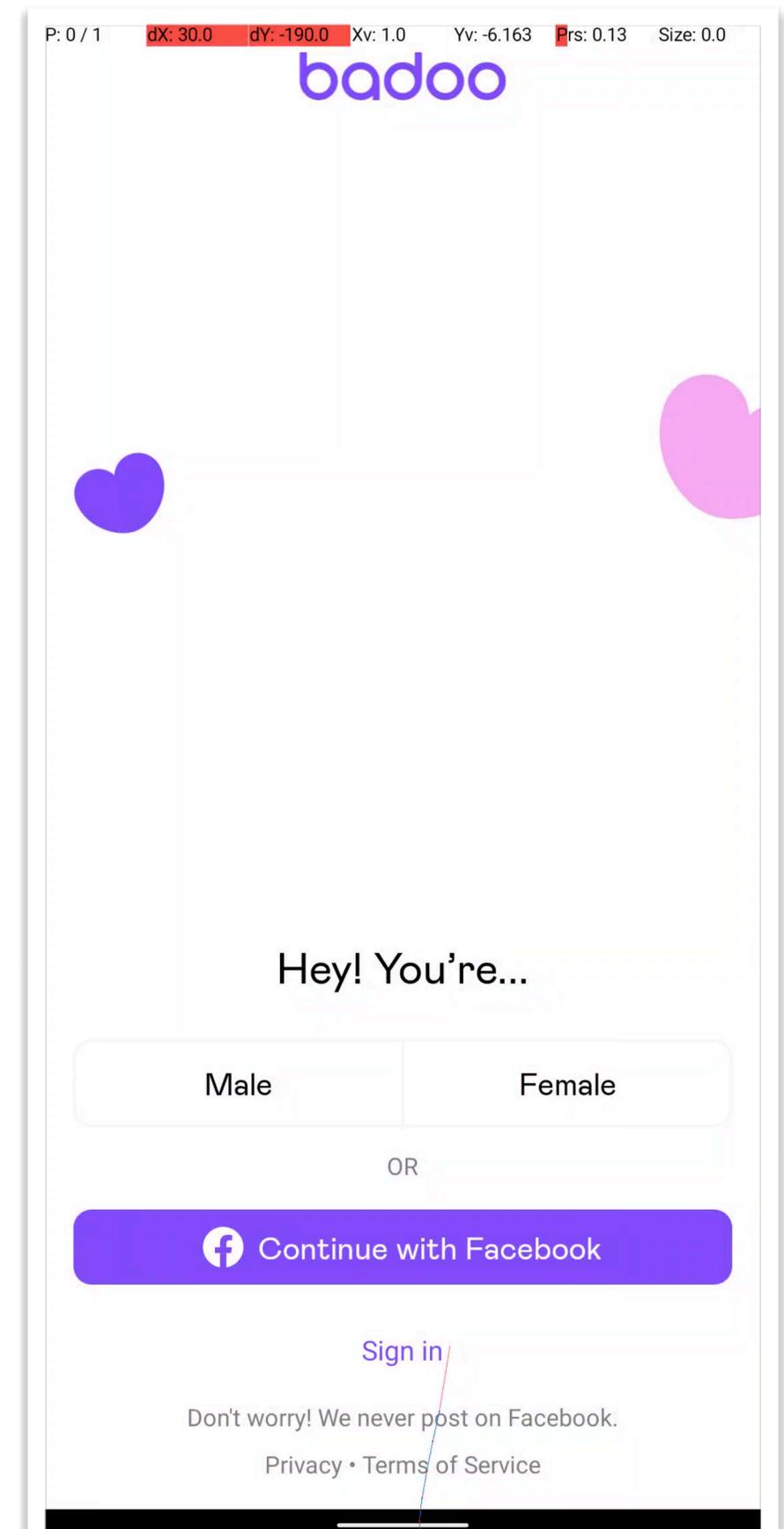
Purchases.subscriptions: revoke

★ **Note:** Requires [authorization](#).

Refunds and immediately revokes a user's subscription purchase. Access to the subscription will be terminated immediately and it will stop recurring.

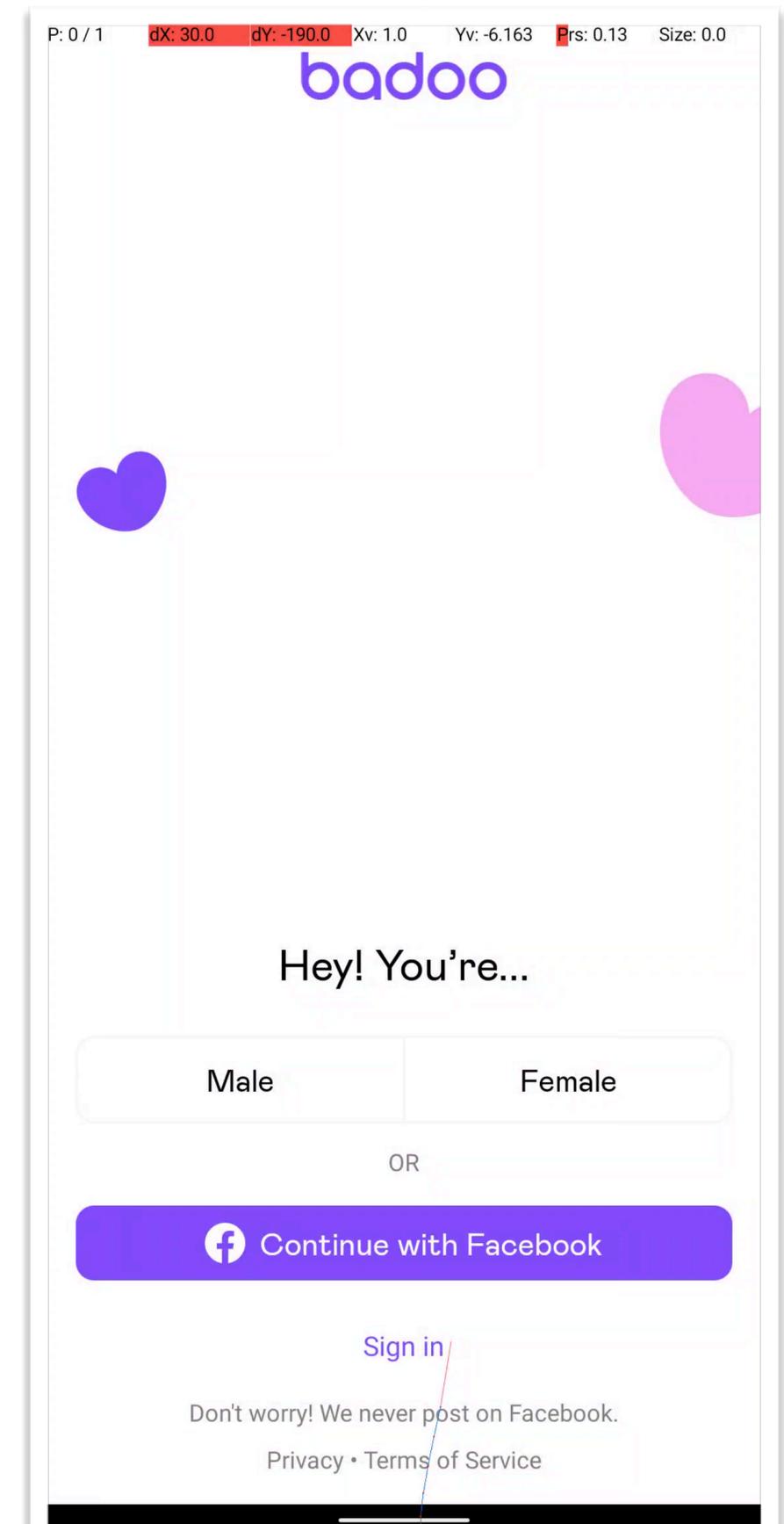
Отмена подписки. Подтверждение

- Before hook



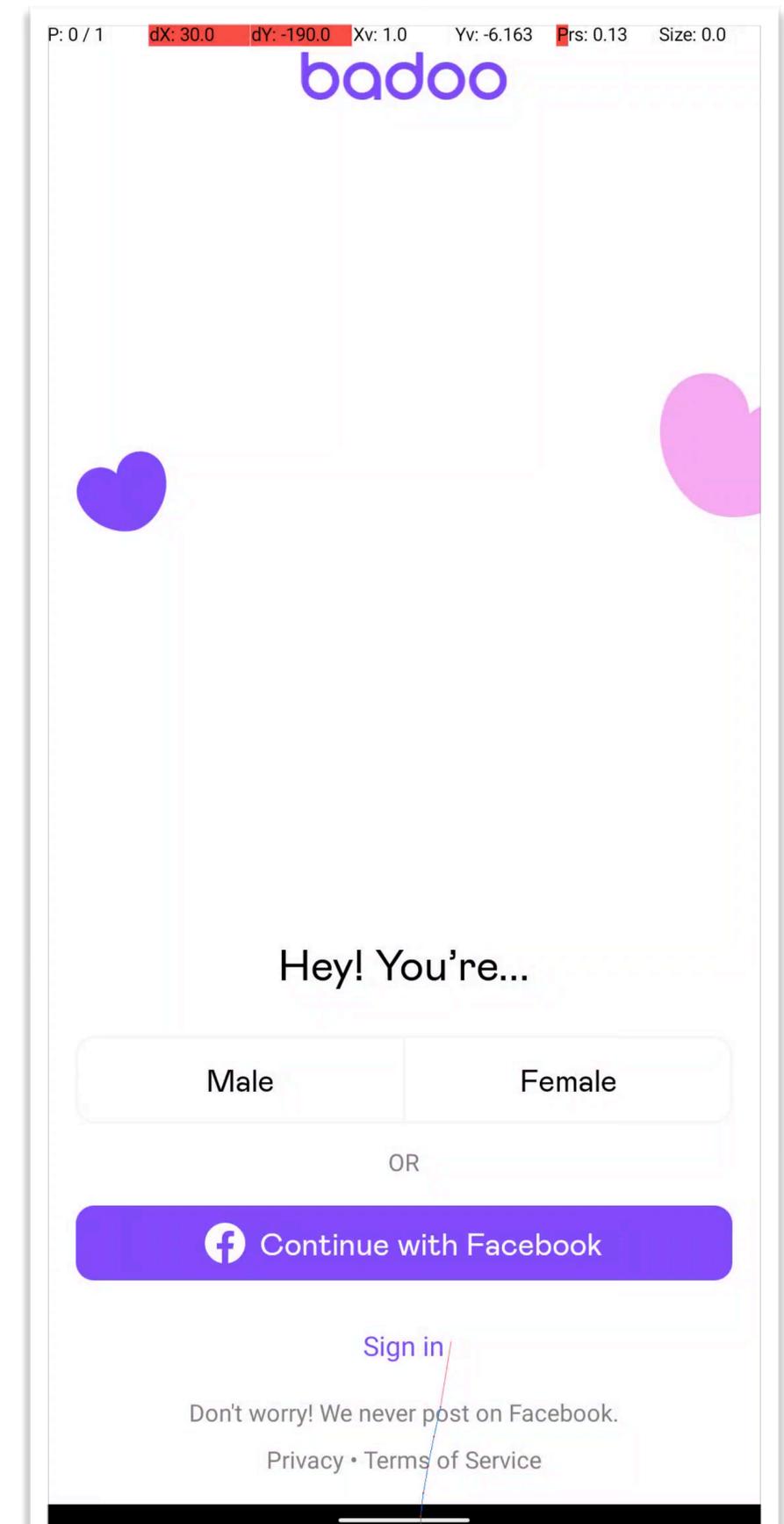
Отмена подписки. Подтверждение

- Before hook



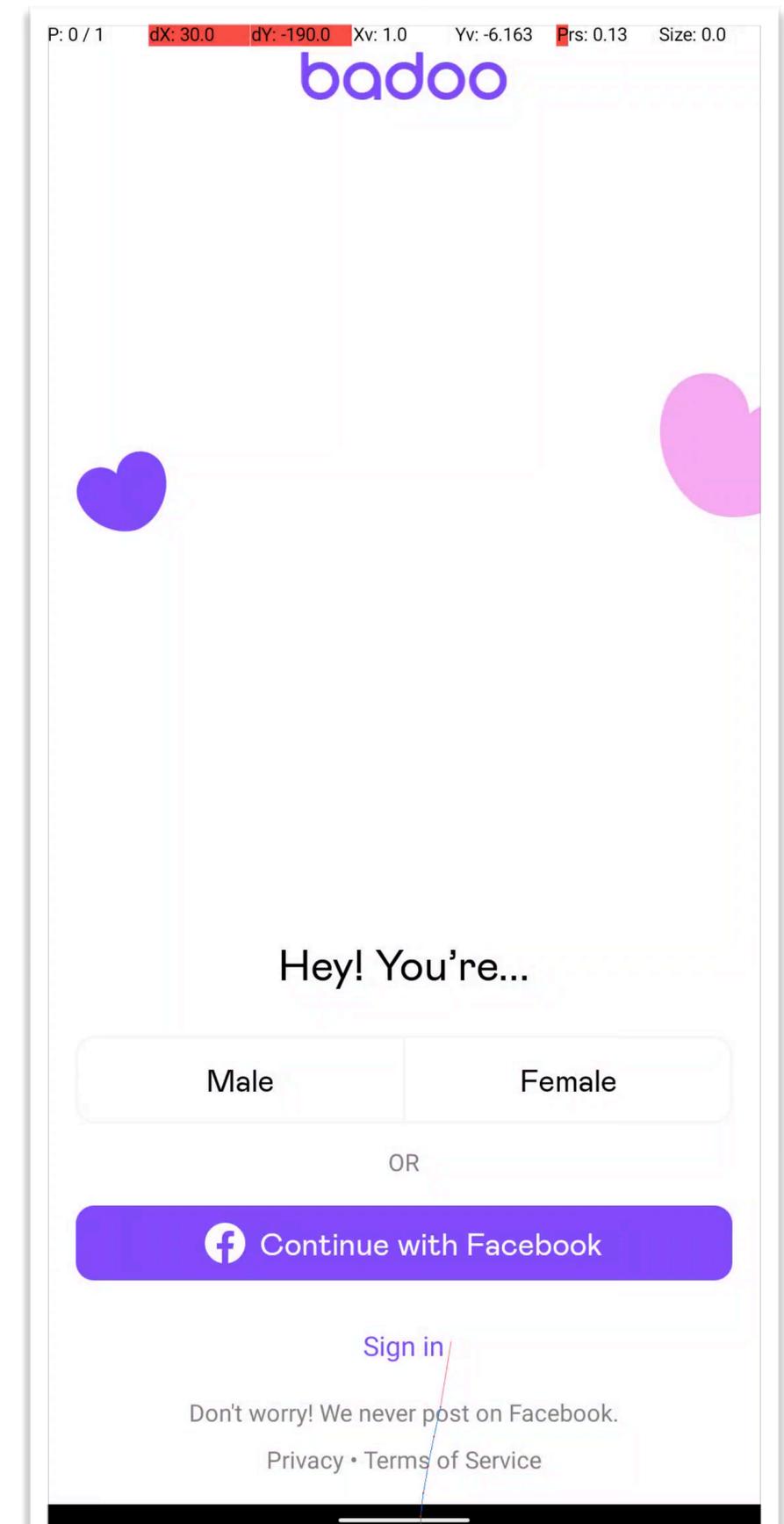
Отмена подписки. Подтверждение

- Before hook
- Проверка подписки в Play Store



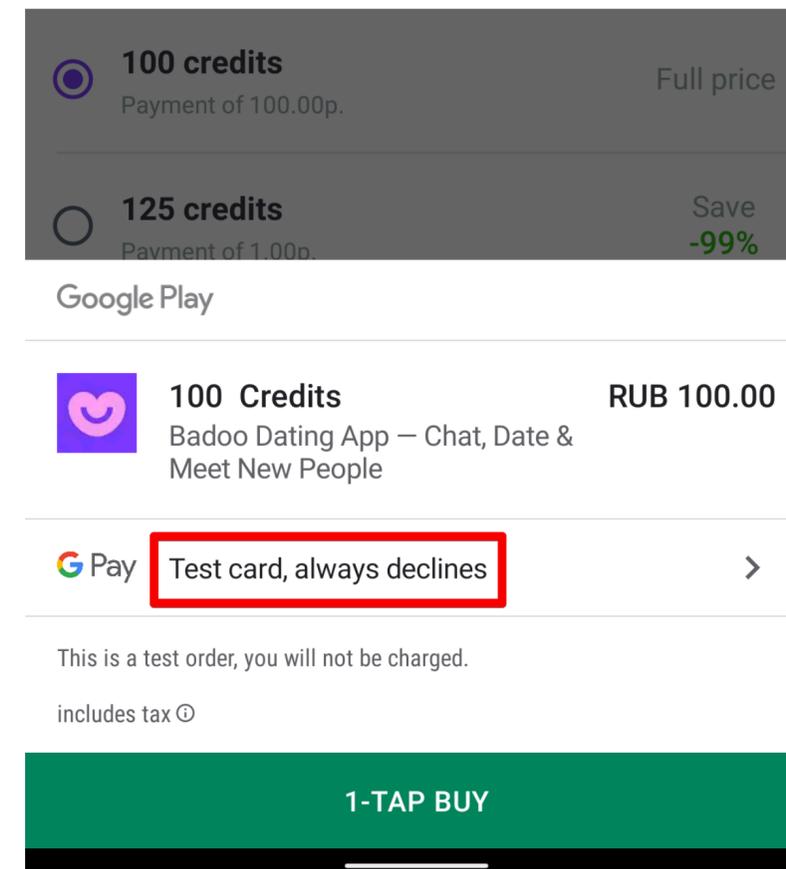
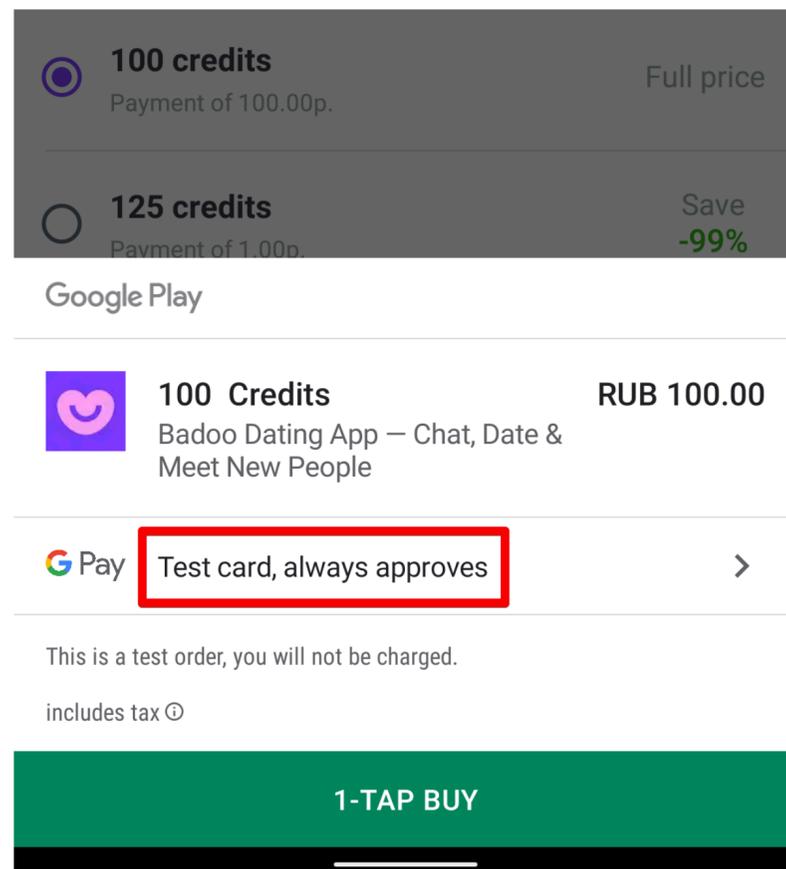
Отмена подписки. Подтверждение

- Before hook
- Проверка подписки в Play Store
- **Отмена при наличии**



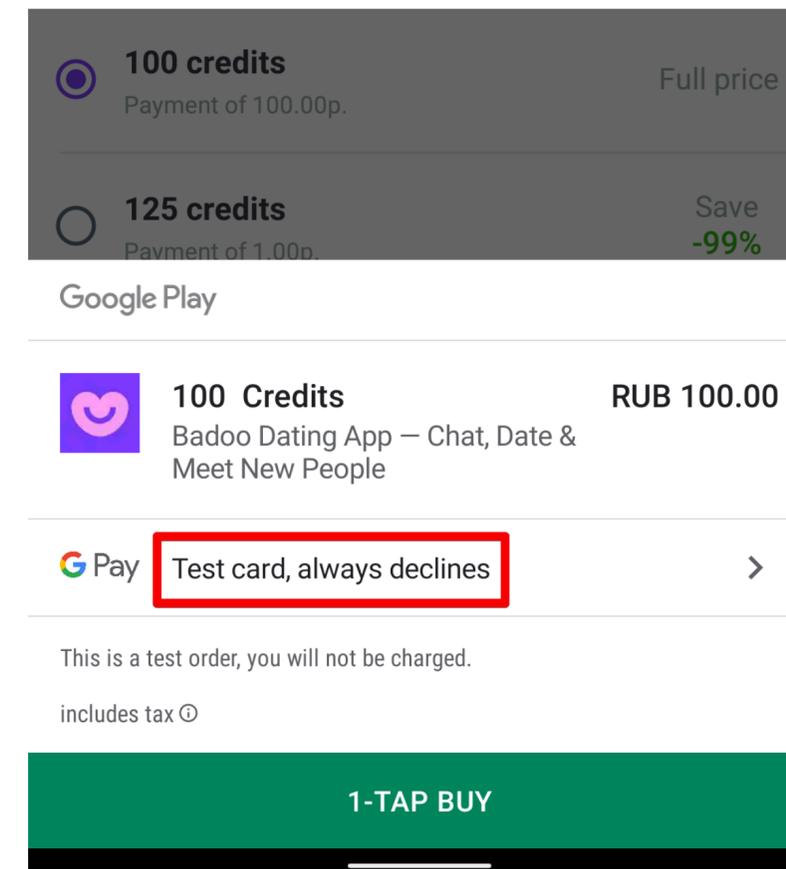
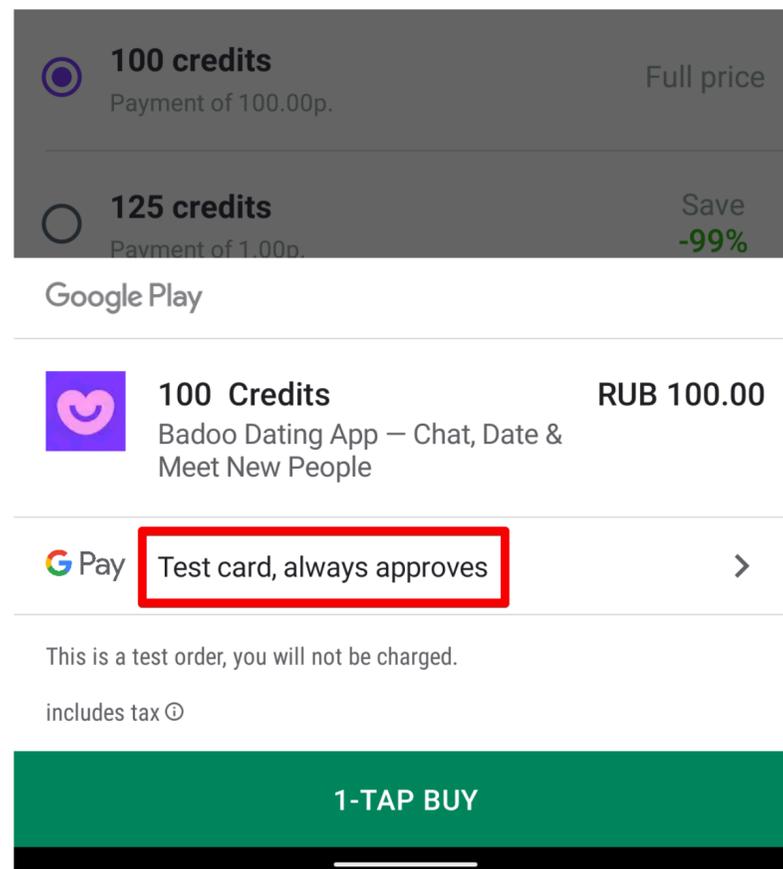
Успешная/неуспешная оплата

- Простой выбор



Успешная/неуспешная оплата

- Простой выбор
- Значение сохраняется



Успешная/неуспешная оплата

- Выставляем желаемый результат

```
def submit_payment(payment_details)
  required_card = payment_details[:required_card] ||= APPROVED_CARD
  change_test_card(required_card)
  tap_on_submit
  enter_password if password_asked?
end
```

Успешная/неуспешная оплата

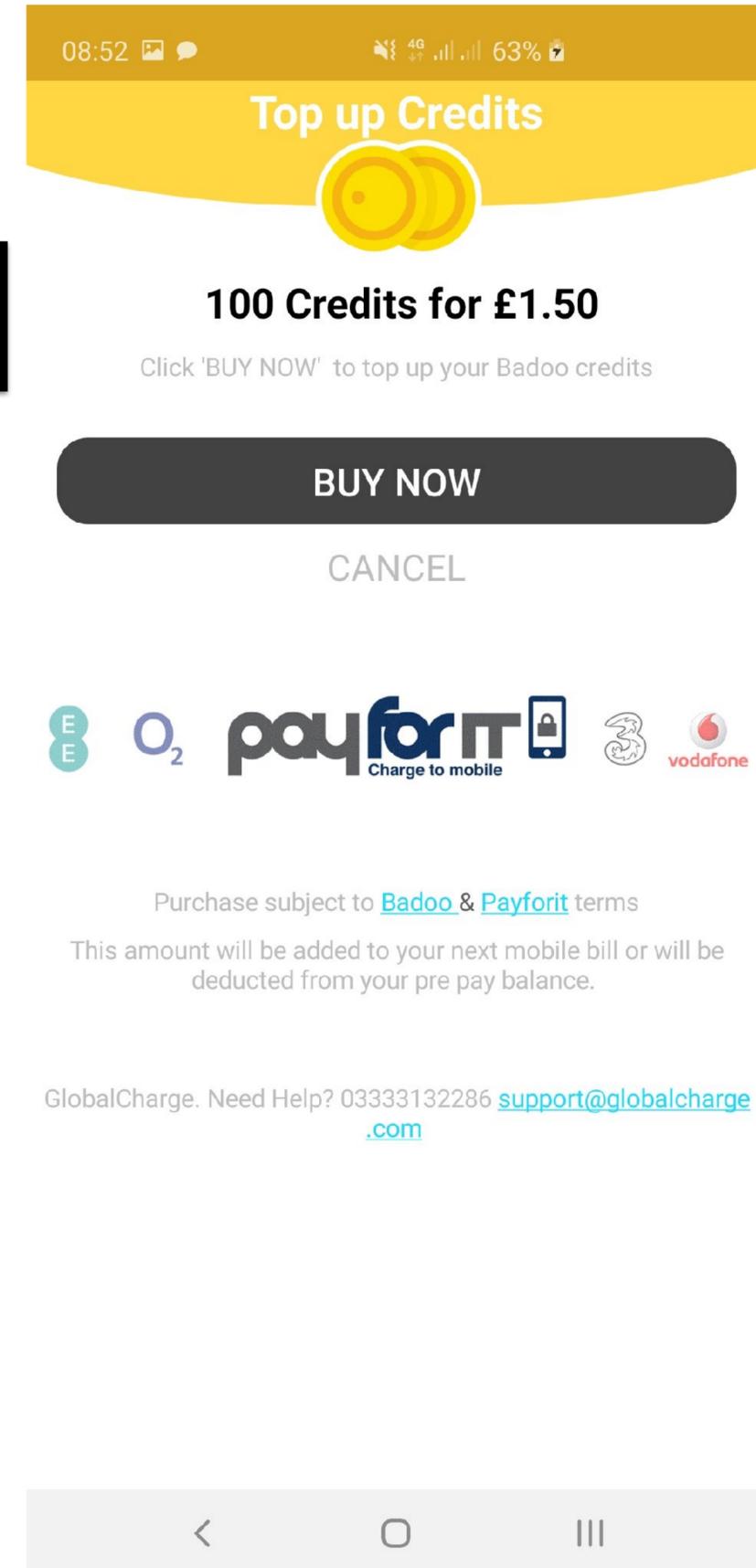
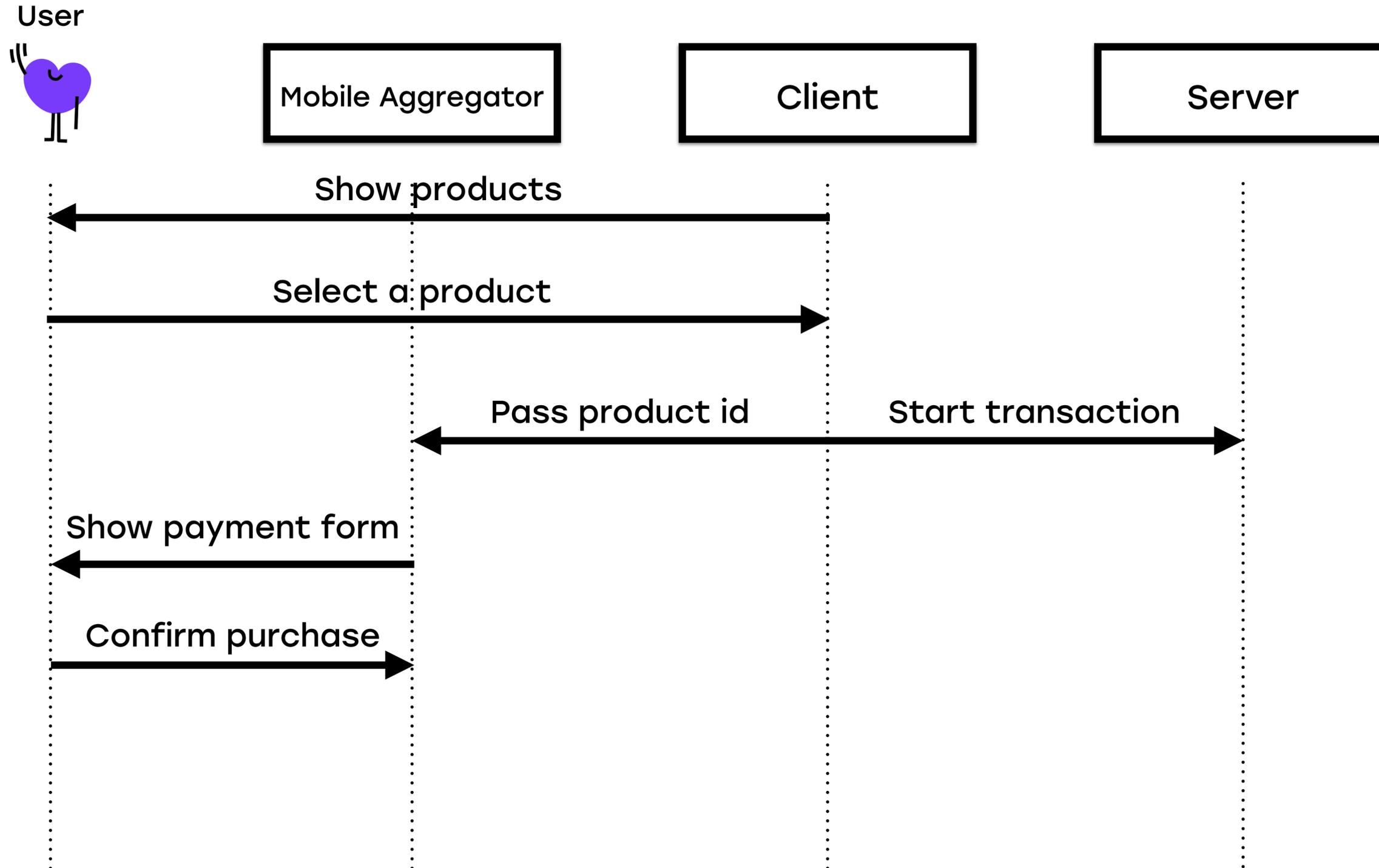
- Выставляем желаемый результат

```
def change_test_card(required_card)
  selected_card = ui.uia_await_text_from_list([APPROVED_CARD, DECLINED_CARD])
  unless required_card == selected_card
    ui.uia_touch_text(selected_card)
    ui.uia_await_text_from_list([APPROVED_CARD, DECLINED_CARD])
    ui.uia_touch_text(required_card)
  end
end
```

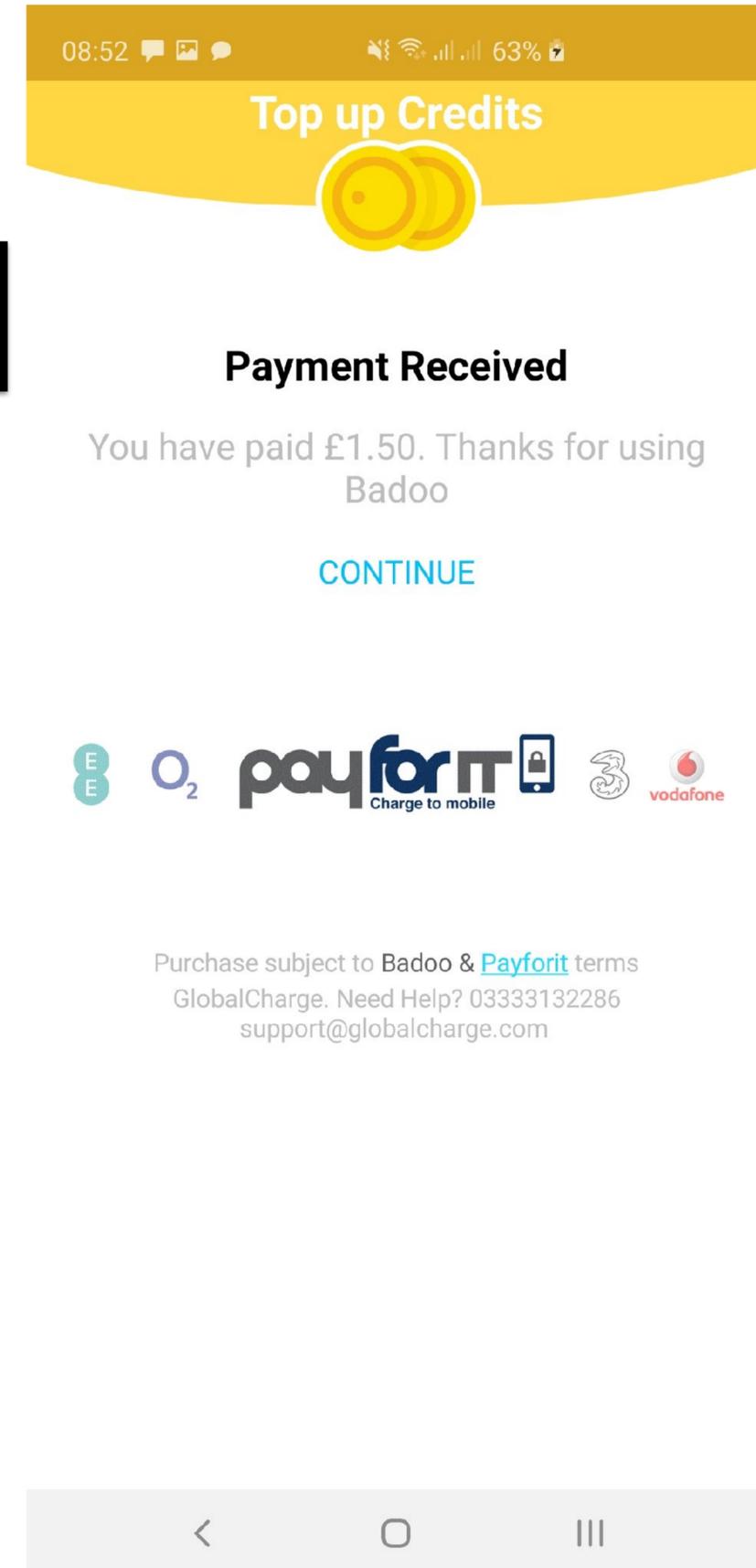
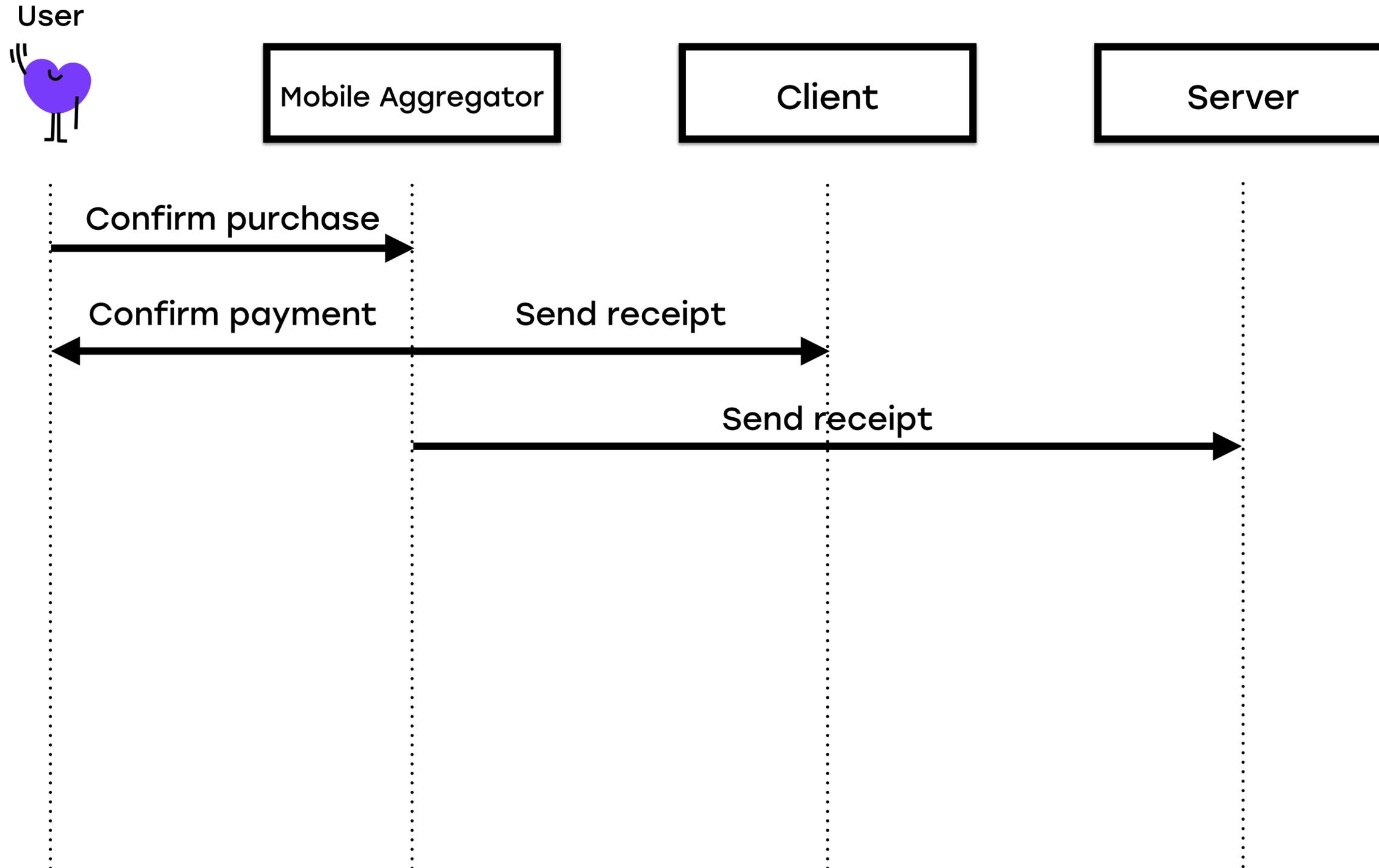
Mobile

Процесс покупки

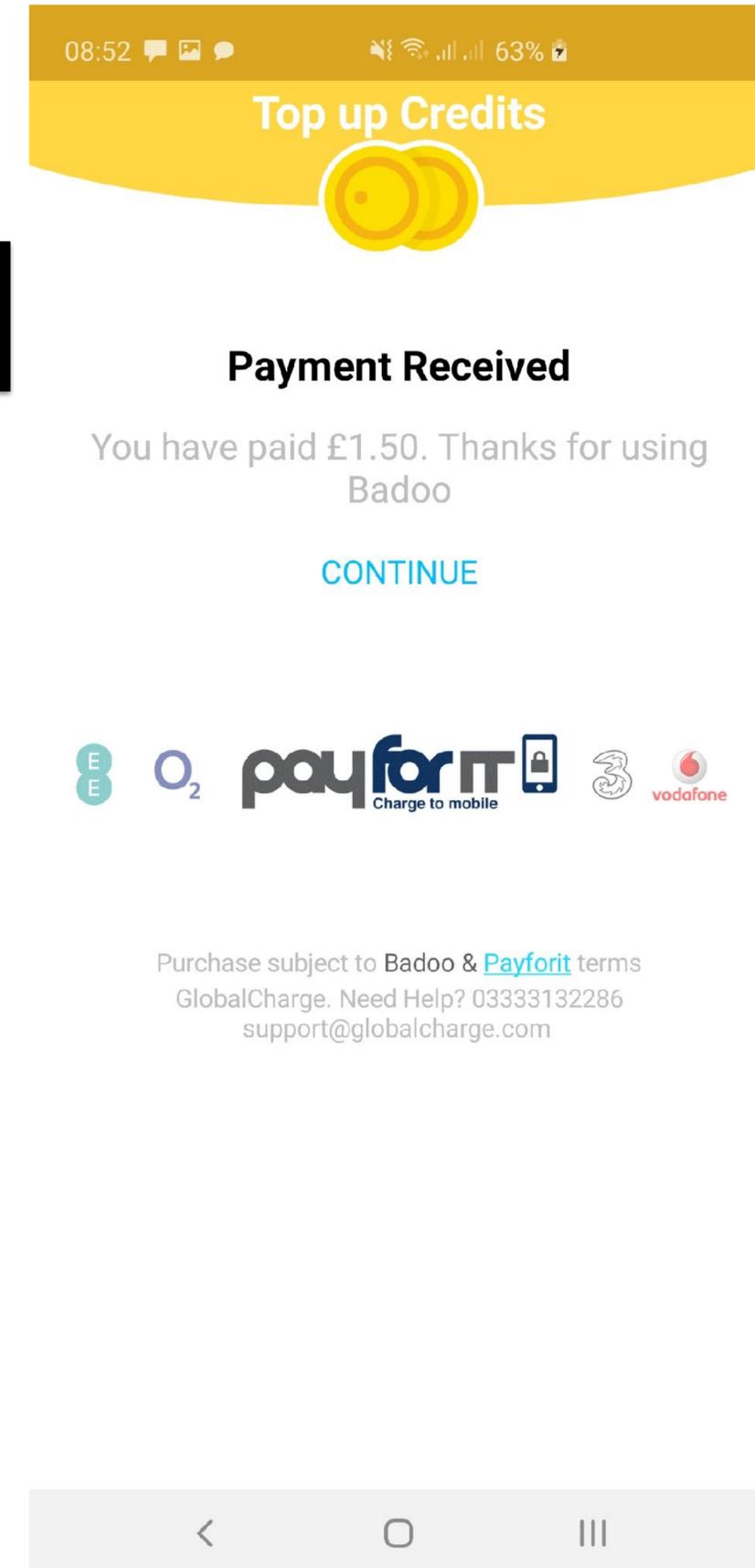
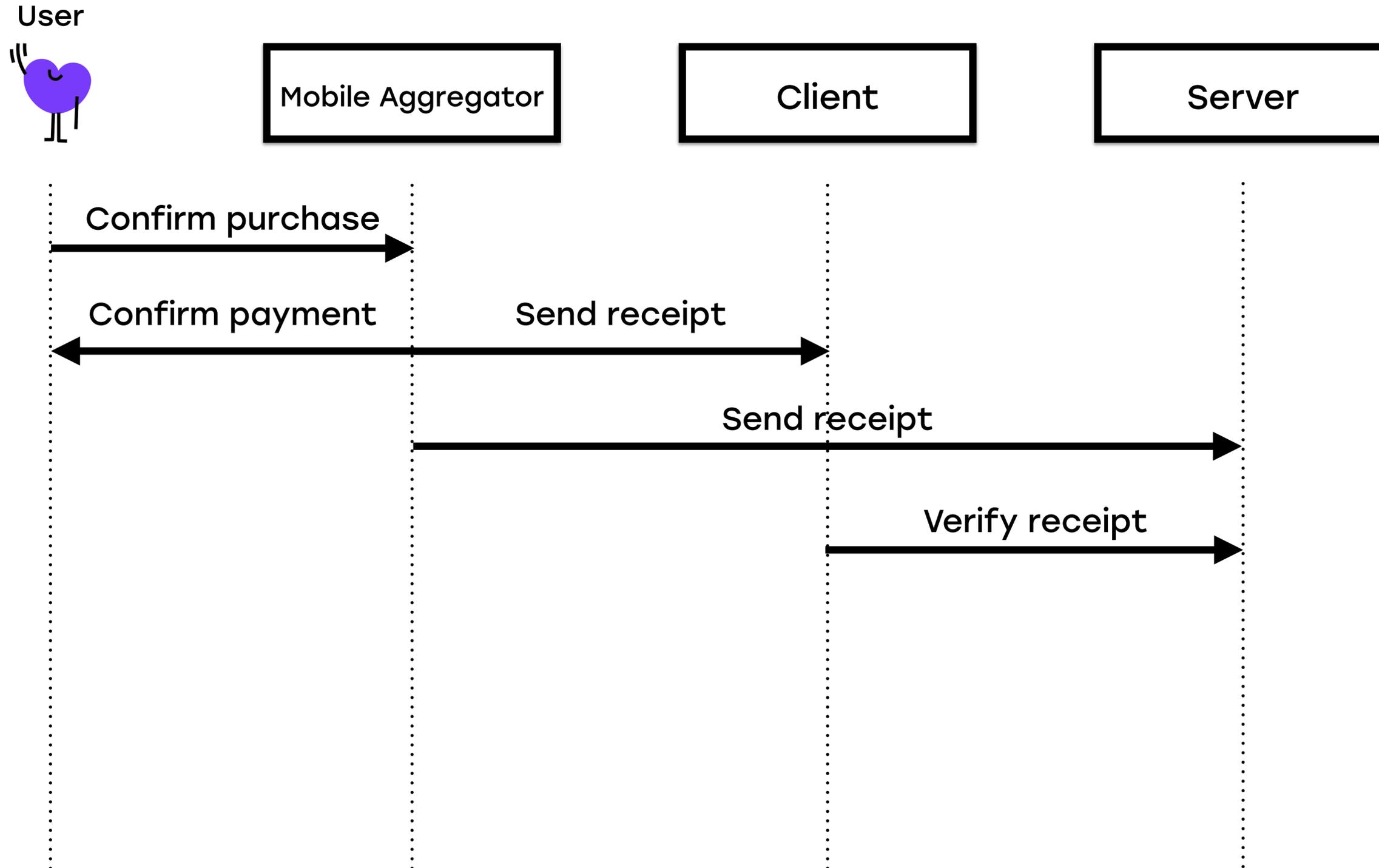
Mobile Процесс покупки



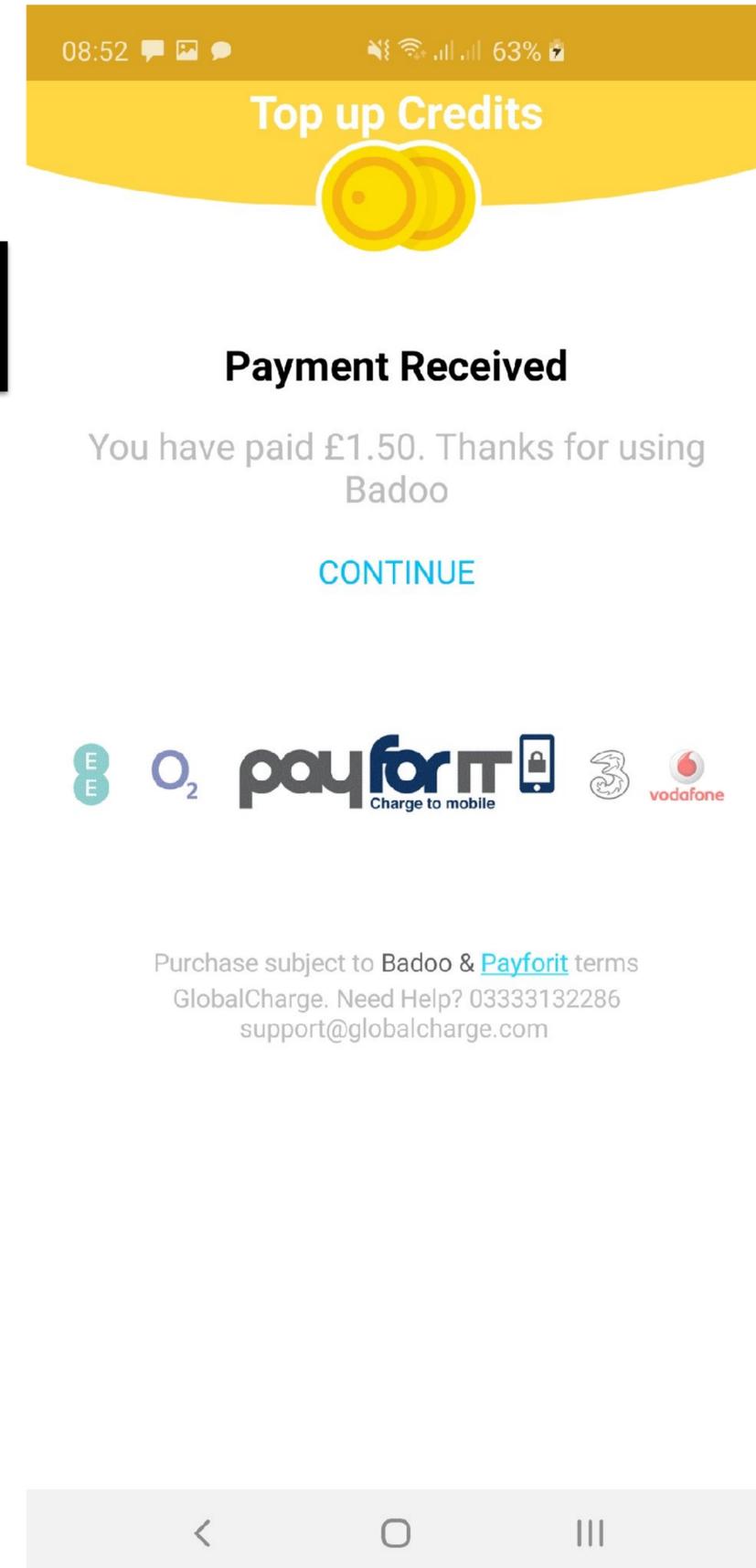
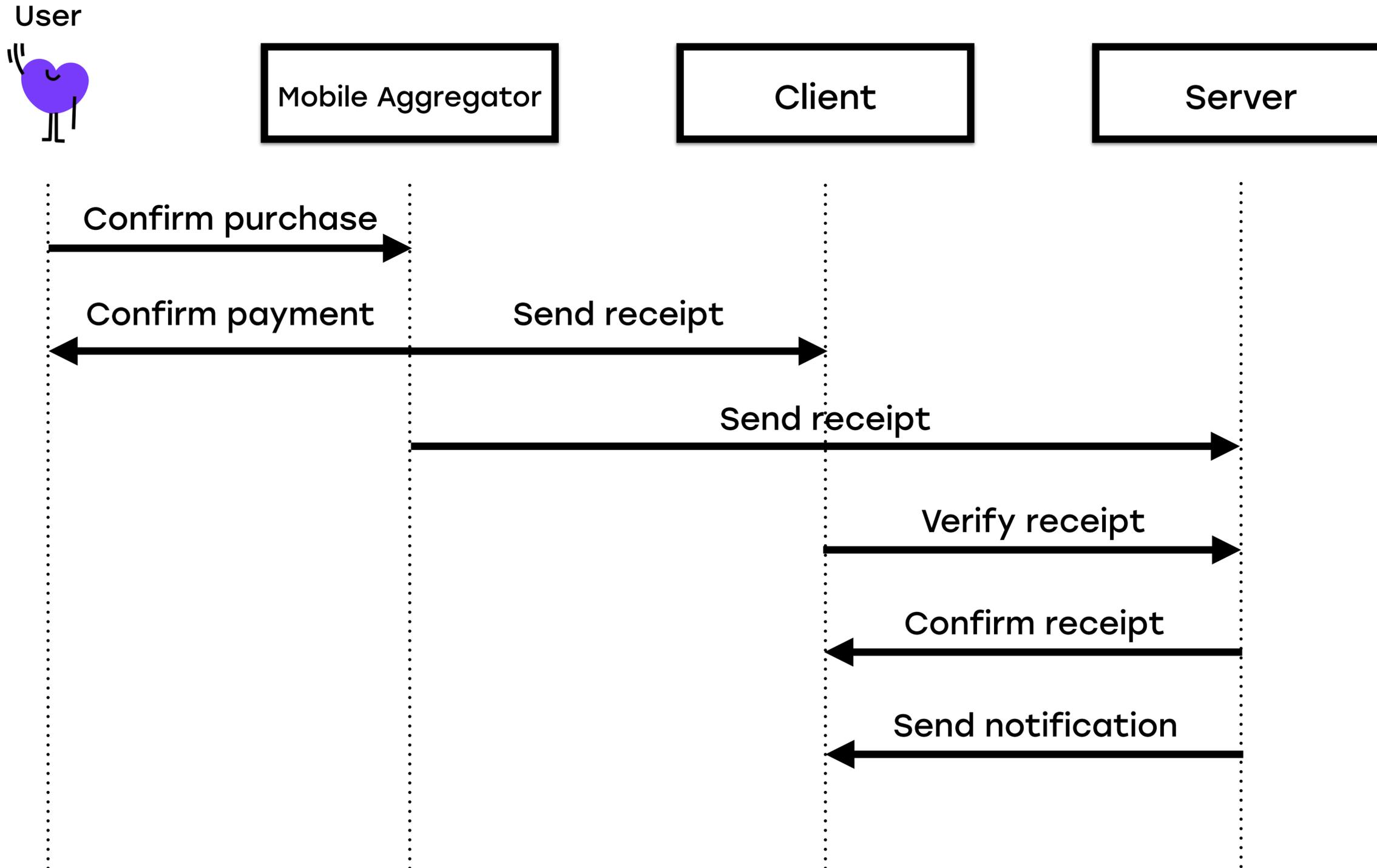
Mobile Процесс покупки



Mobile Процесс покупки



Mobile Процесс покупки



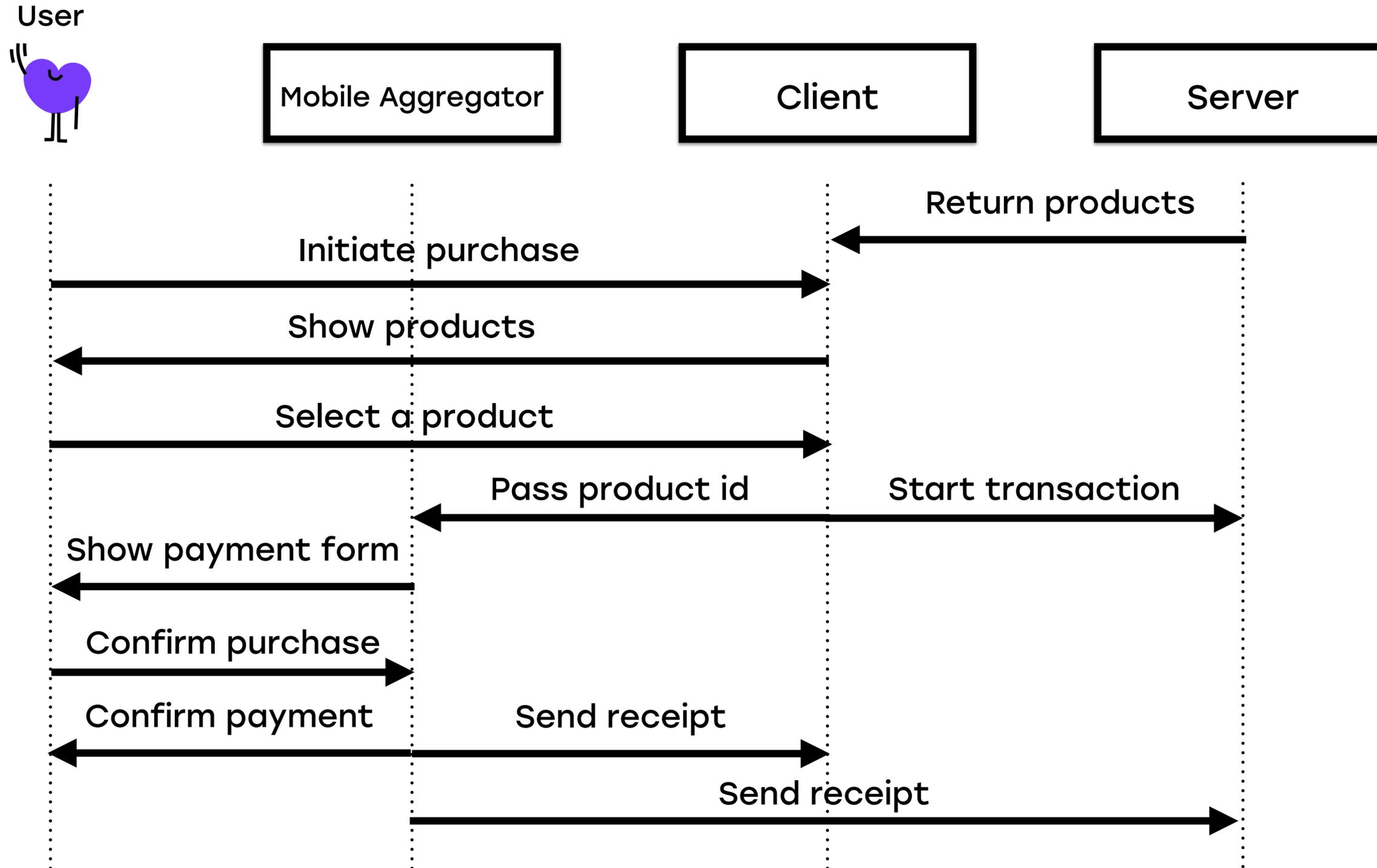
Mobile

Тестовое окружение

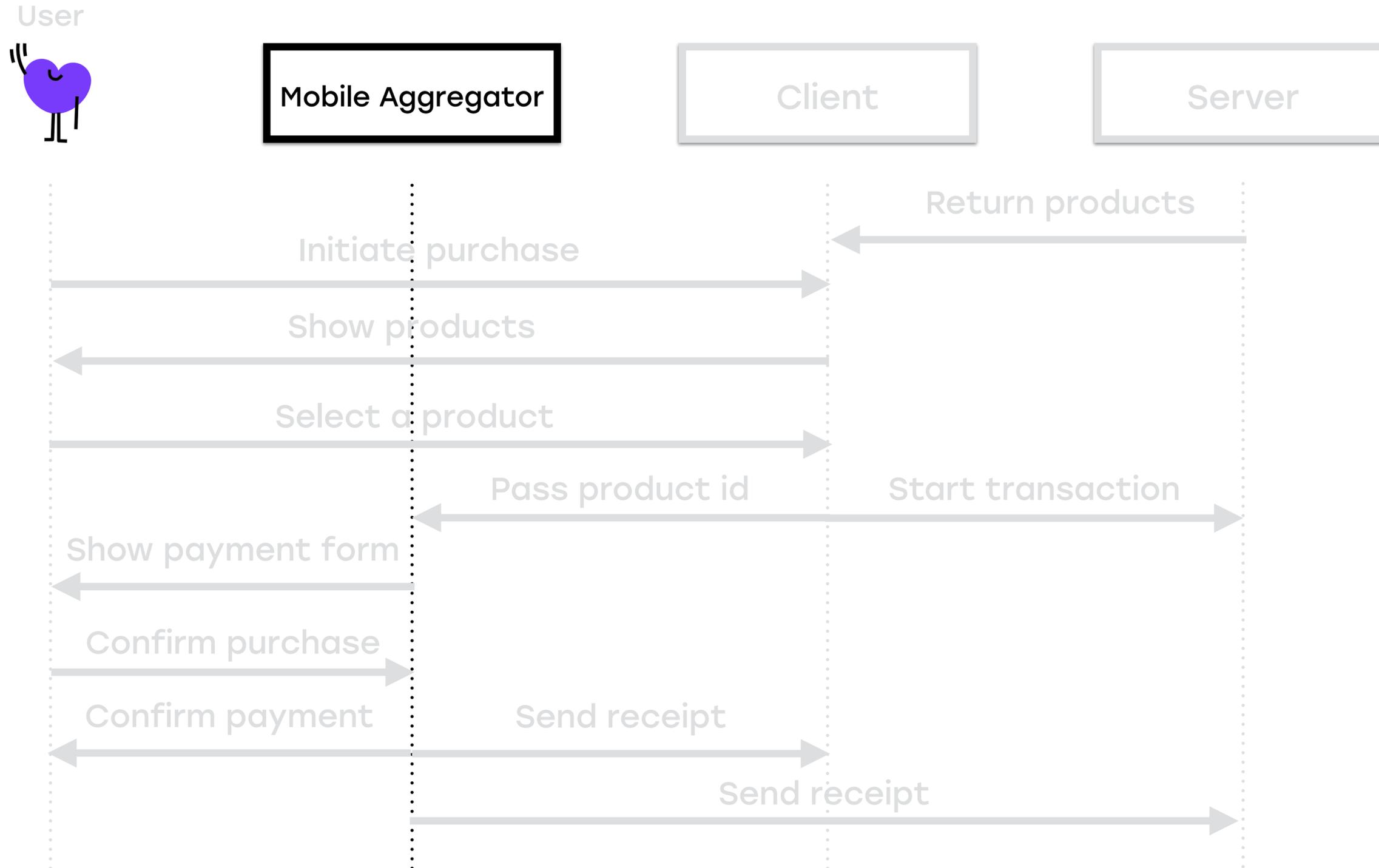
Почему песочницы

- + Процесс покупки максимально близок к реальному
- + Дешево
- Есть не у всех провайдеров

Mobile Тестовое окружение



Mobile Тестовое окружение



Почему песочницы

- + Процесс покупки максимально близок к реальному
- + Дешево
- Есть не у всех провайдеров

Проблема песочниц

- У Fortumo – есть

Проблема песочниц

- У Fortumo – есть
- У остальных – нет

Fortumo Sandbox

- **Добавить тестовые продукты на сервере**

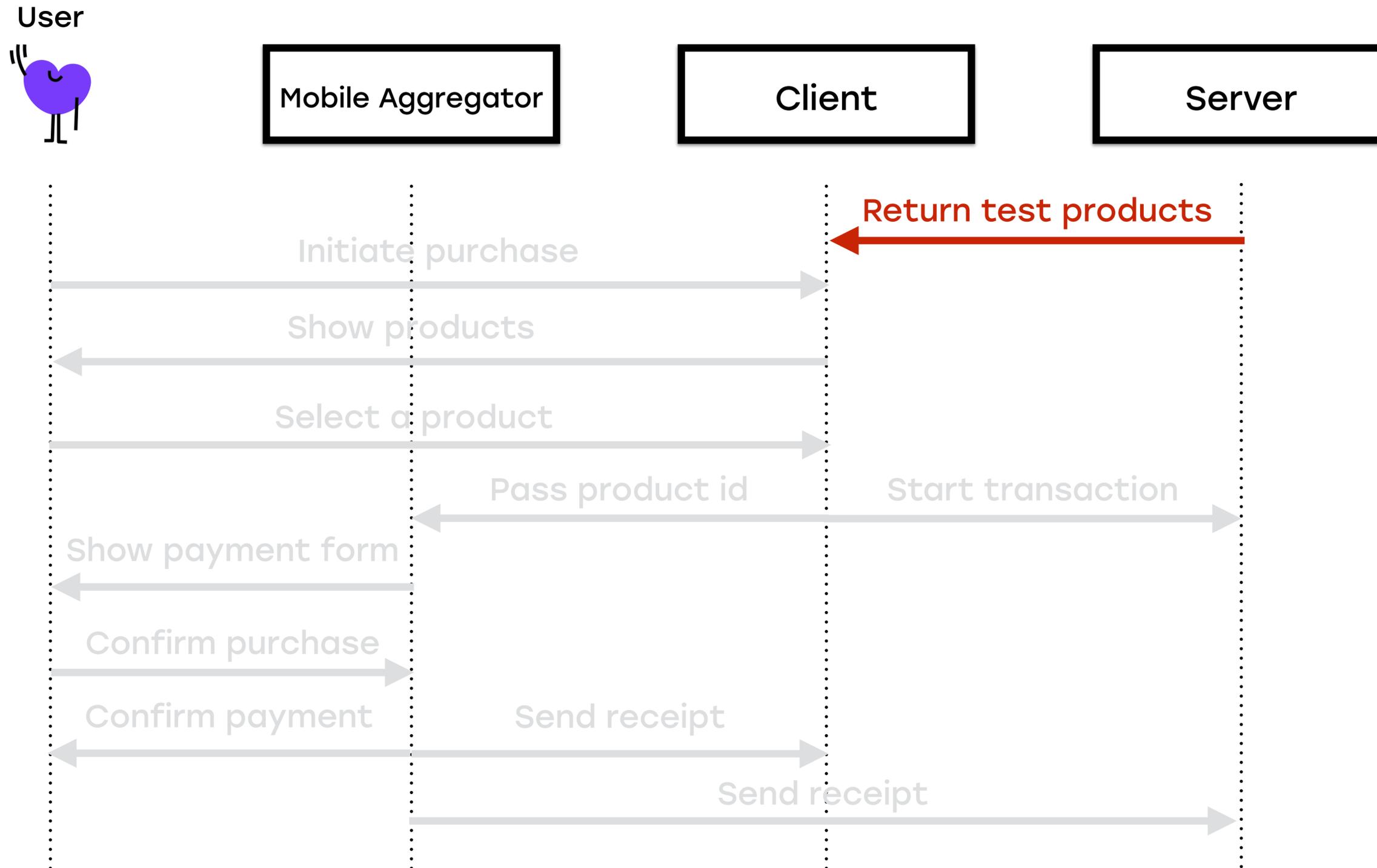
Fortumo Sandbox

- Добавить тестовые продукты на сервере
- **Настроить условия для их получения**

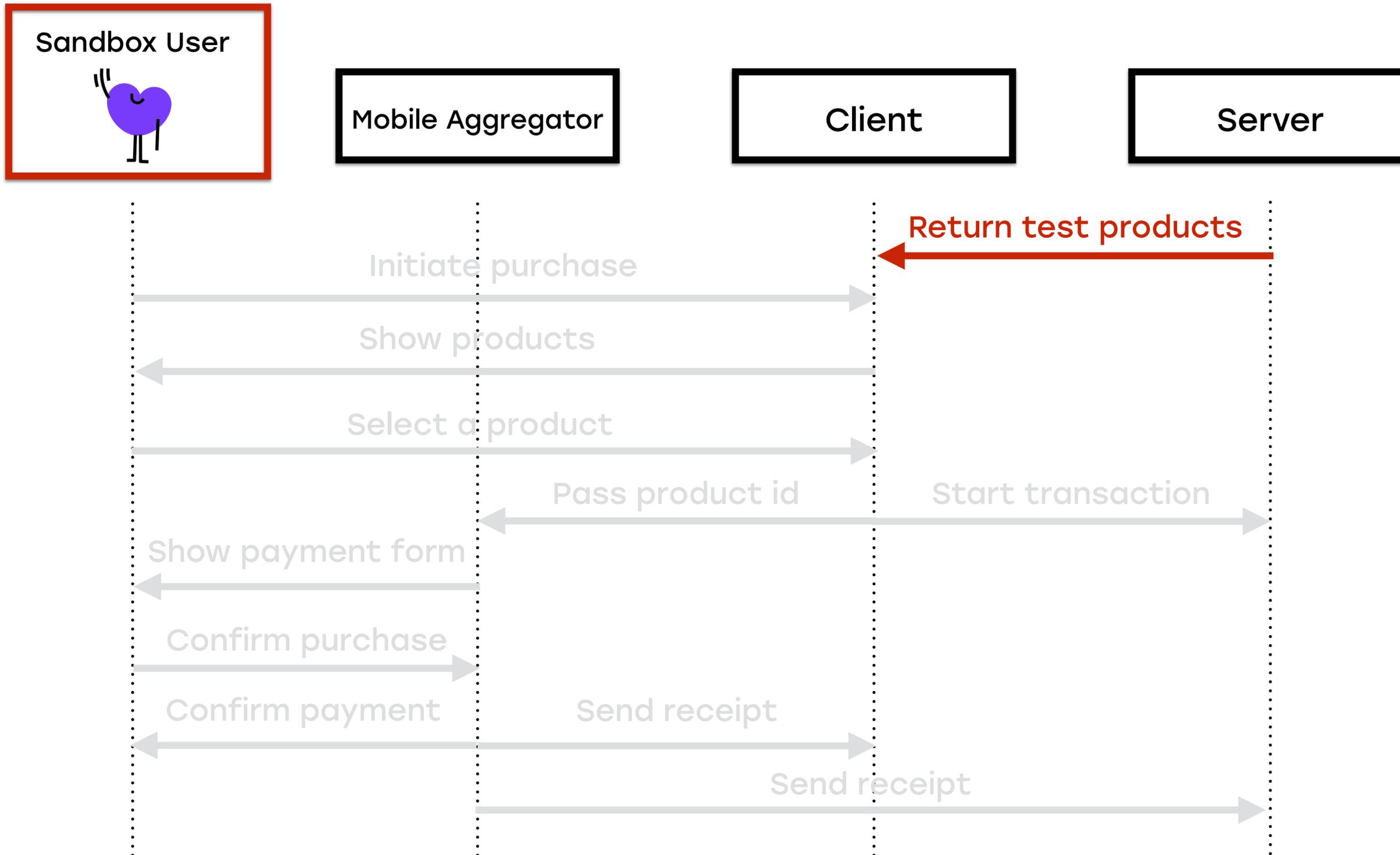
Fortumo Sandbox

- Добавить тестовые продукты на сервере
- Настроить условия для их получения
- **Тестовые пользователи удовлетворяют этим условиям**

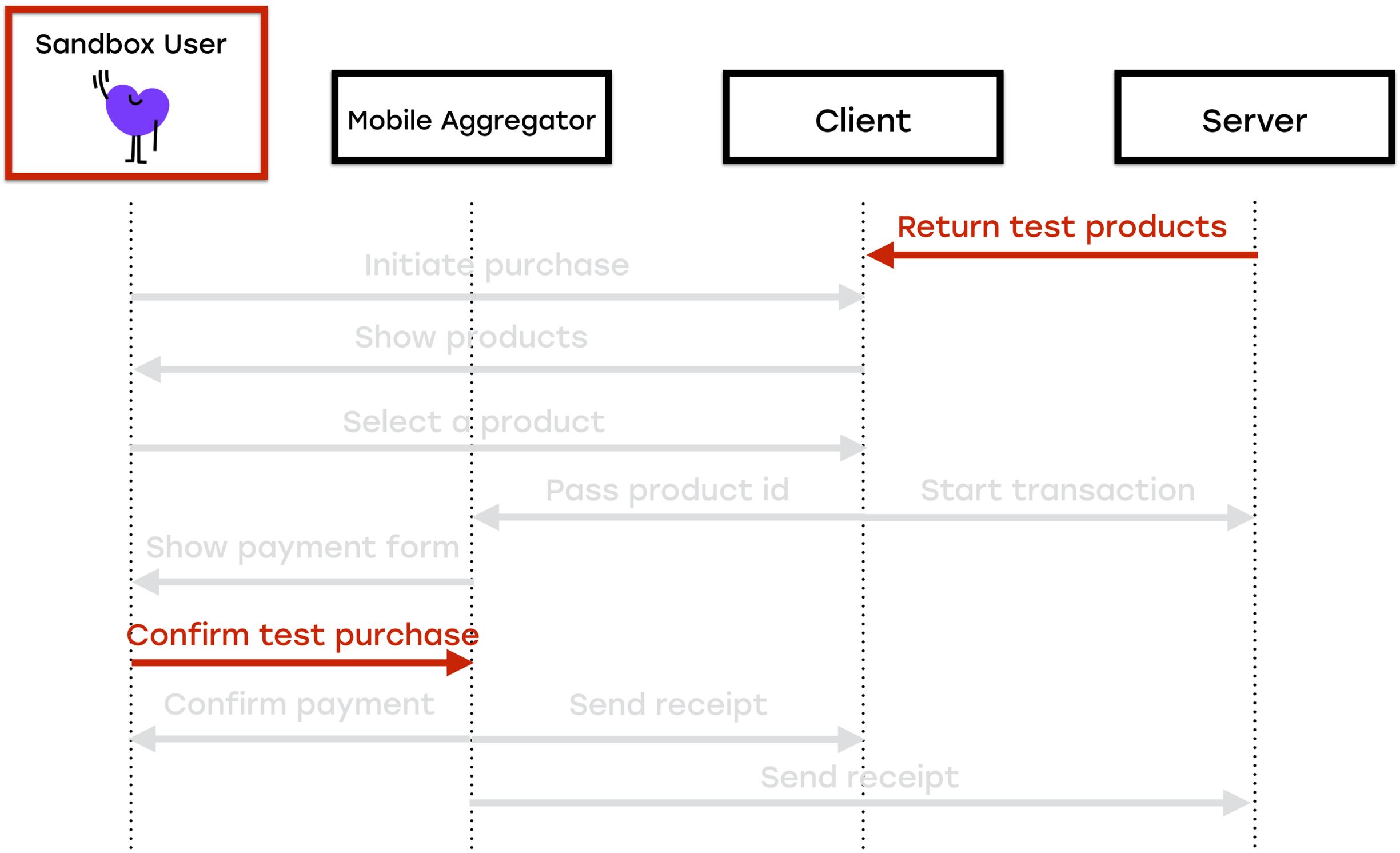
Mobile Тестовое окружение



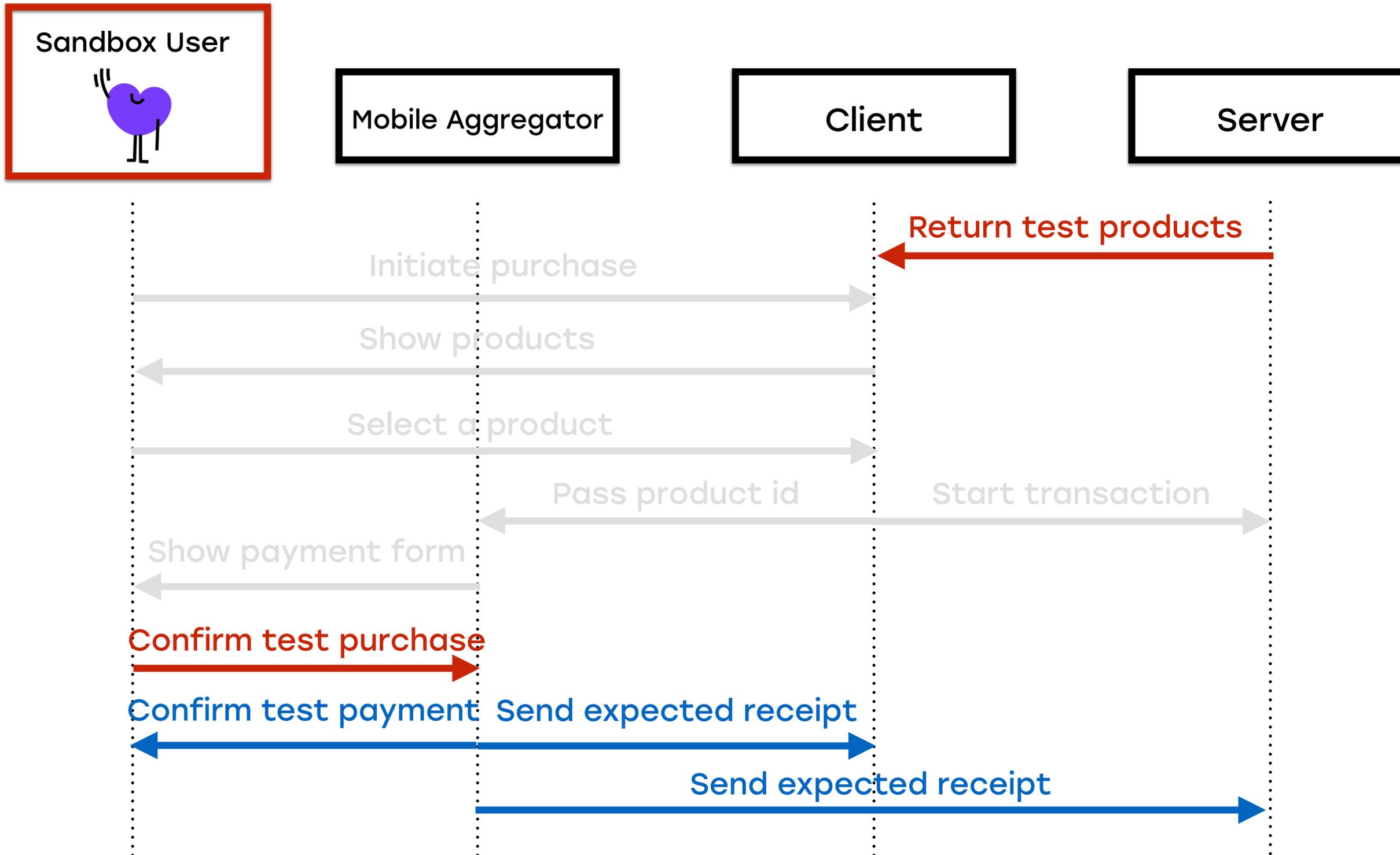
Mobile Тестовое окружение



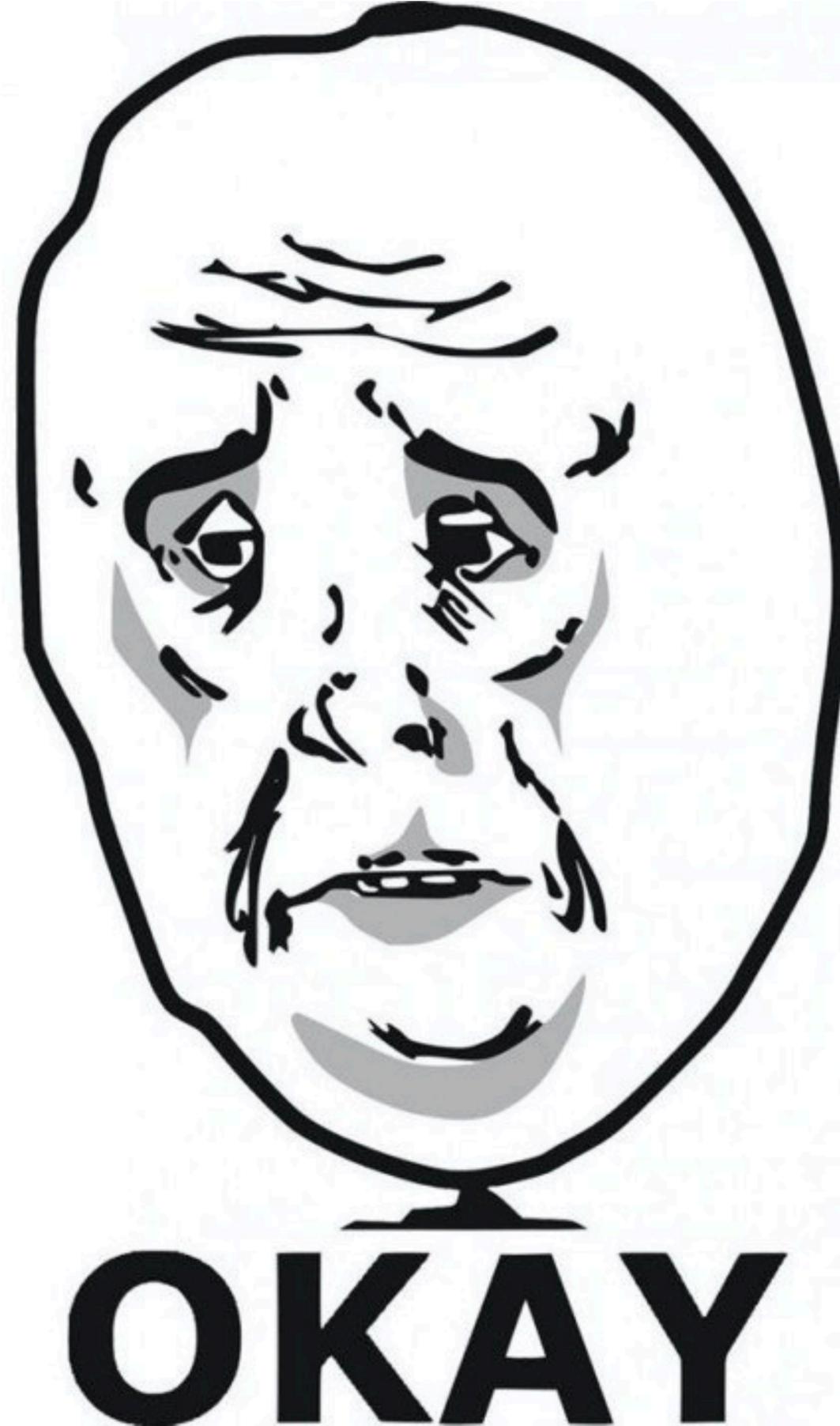
Mobile Тестовое окружение



Mobile Тестовое окружение



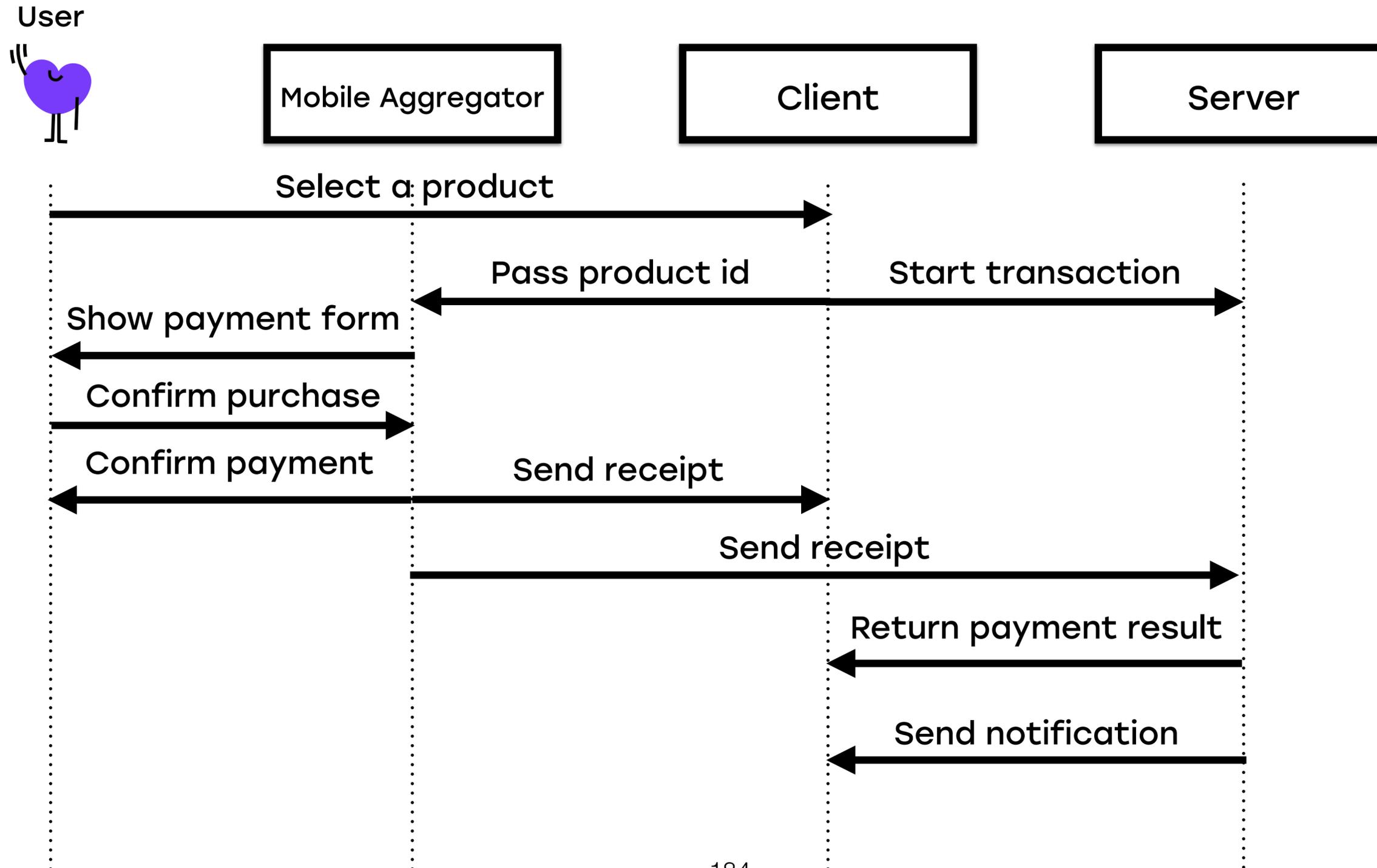
Песочницы других провайдеров



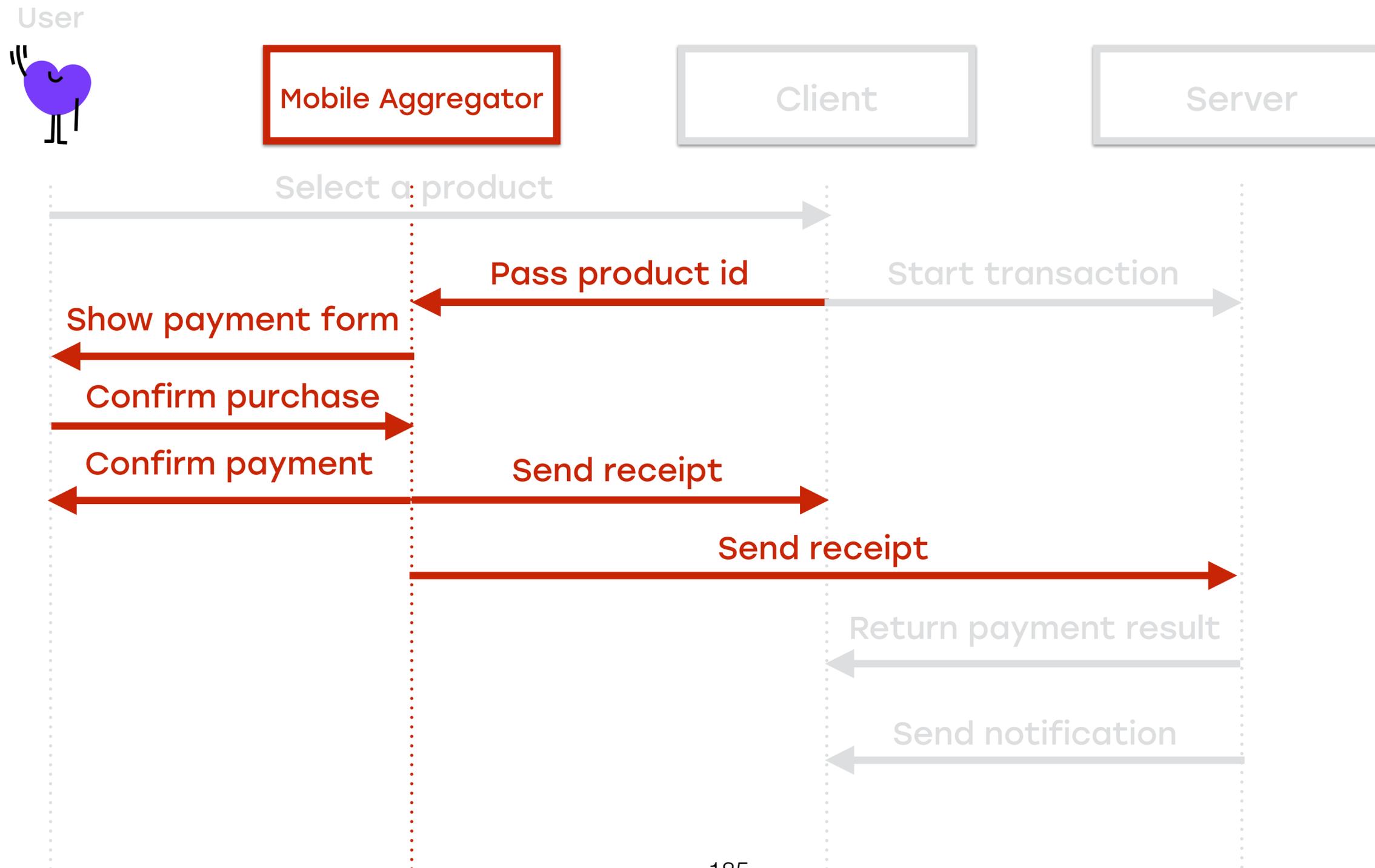
Другие провайдеры

- Будем использовать метод исключения внешних зависимостей

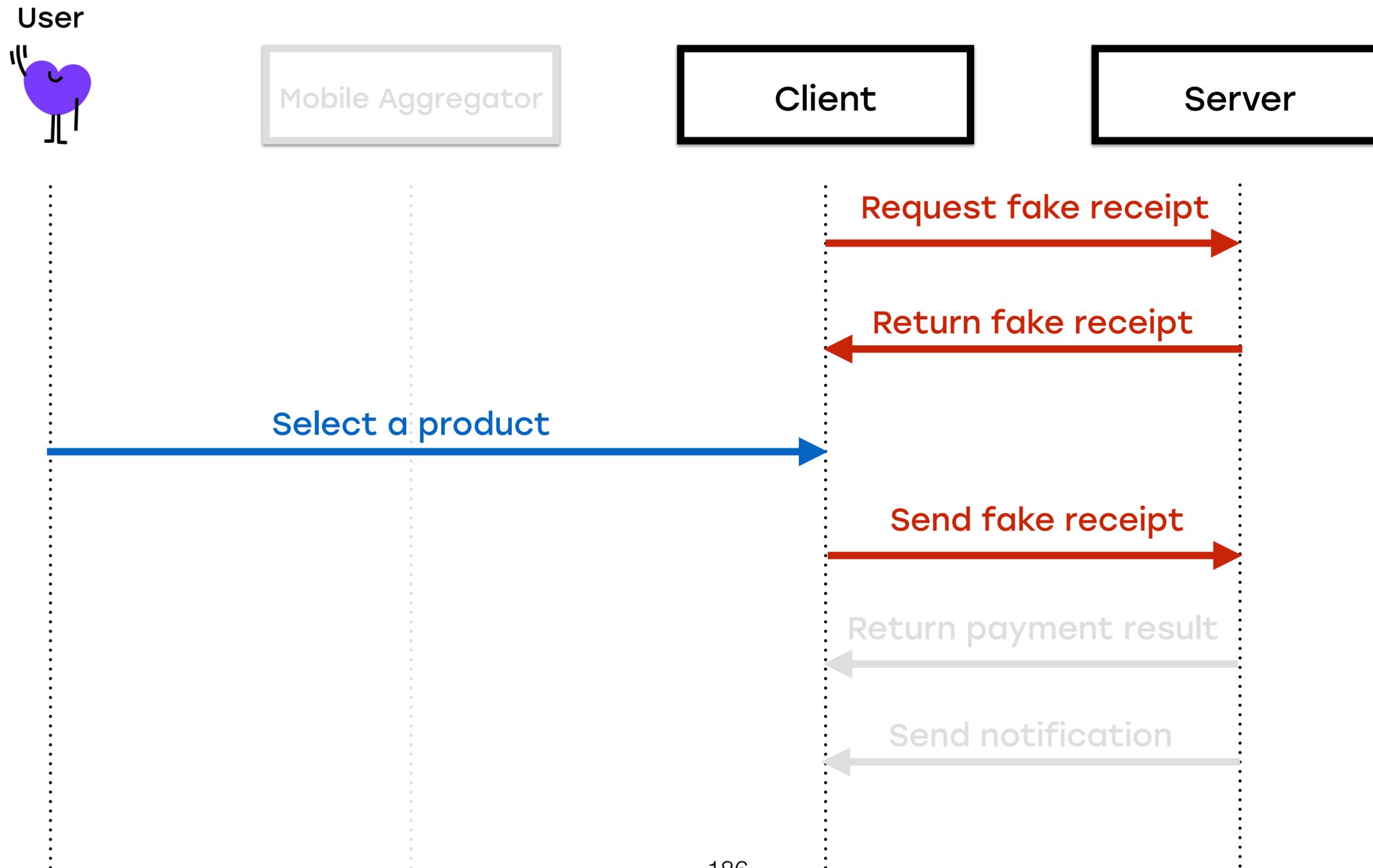
ИСКЛЮЧЕНИЕ ВНЕШНИХ ЗАВИСИМОСТЕЙ



ИСКЛЮЧЕНИЕ ВНЕШНИХ ЗАВИСИМОСТЕЙ



ИСКЛЮЧЕНИЕ ВНЕШНИХ ЗАВИСИМОСТЕЙ



Исключение внешних зависимостей



HEISENBUG
2019 Piter

Виктор Короневич
Владимир Солодов
Badoo

Тестируем платные сервисы:
как перестать запускать
самолет, чтобы проверить
лампочку на приборной
панели

The image is a YouTube video thumbnail. It features a white background with orange 'x' marks scattered around. On the left, there is a plus sign followed by the text 'HEISENBUG' in large, bold, black and orange letters, with '2019 Piter' underneath. Below this, the names 'Виктор Короневич' and 'Владимир Солодов' are listed in bold black text, with 'Badoo' in smaller orange text below them. A large orange banner at the bottom left contains white text describing the video's content. On the right side, there are two portrait photos: one of a man with a beard and sunglasses, and another of a man with short dark hair. The entire thumbnail is framed by a light gray border with orange 'x' marks at the corners and bottom.

<https://www.youtube.com/watch?v=PWxE0KZLU8Q>

Mobile

АВТОМАТИЗАЦИЯ

Идентификация оператора

- **IMSI** (International Mobile Subscriber Identity) – международный идентификатор мобильного абонента

Идентификация оператора

- **IMSI** (International Mobile Subscriber Identity) – международный идентификатор мобильного абонента
- **MCC** (Mobile Country Code) – мобильный код страны

Идентификация оператора

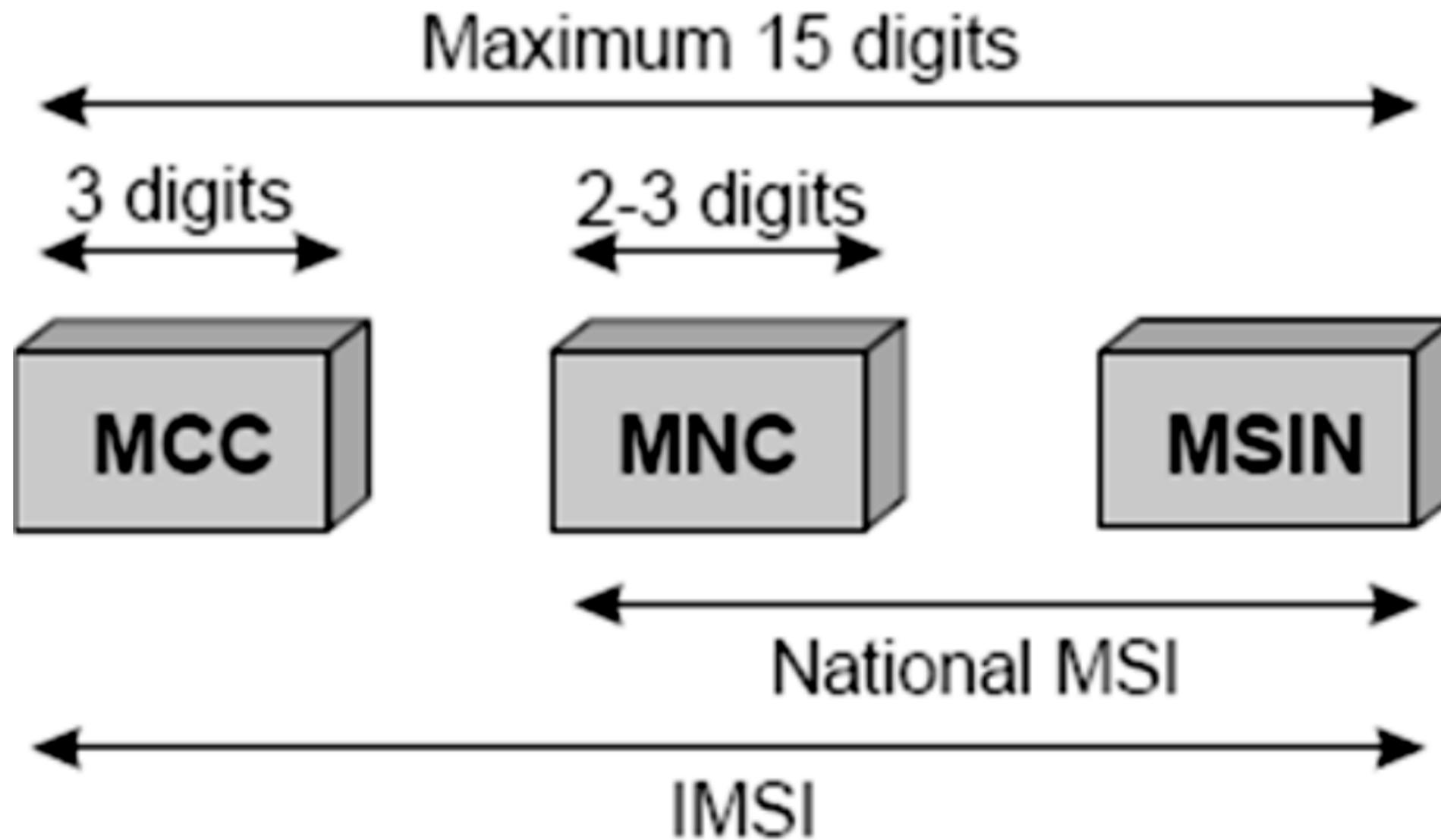
- **IMSI** (International Mobile Subscriber Identity) – международный идентификатор мобильного абонента
- **MCC** (Mobile Country Code) – мобильный код страны
- **MNC** (Mobile Network Code) – код мобильной сети

Идентификация оператора

- **IMSI** (International Mobile Subscriber Identity) – международный идентификатор мобильного абонента
- **MCC** (Mobile Country Code) – мобильный код страны
- **MNC** (Mobile Network Code) – код мобильной сети

MCC + MNC Используется для идентификации мобильного оператора

Идентификация оператора



Получение разных провайдеров

- Настроены на комбинацию оператора (MCC + MNC) и страны

Получение разных провайдеров

- Настроены на комбинацию оператора (MCC + MNC) и страны
- При выборе продуктов приоритет у мобильного оператора

Получение разных провайдеров

- Настроены на комбинацию оператора (MCC + MNC) и страны
- При выборе продуктов приоритет у мобильного оператора
- **Нужно менять MCC + MNC для тестирования**

Термины

- **Бэкдор (Backdoor) – метод, определенный в приложении, который можно вызвать из кода тестов**

Термины

- Бэкдор (Backdoor) – метод, определенный в приложении, который можно вызвать из кода тестов
- Подмена деталей SIM-карты (MCC и MNC)

Термины

- Бэкдор (Backdoor) – метод, определенный в приложении, который можно вызвать из кода тестов
- Подмена деталей SIM-карты (MCC и MNC)
- **Установка вариантов А/Б-тестов**

Термины

- Бэкдор (Backdoor) – метод, определенный в приложении, который можно вызвать из кода тестов
- Подмена деталей SIM-карты (MCC и MNC)
- Установка вариантов А/Б-тестов
- **Отключение показа тултипов**

Термины

- Бэкдор (Backdoor) – метод, определенный в приложении, который можно вызвать из кода тестов
- Подмена деталей SIM-карты (MCC и MNC)
- Установка вариантов A/B-тестов
- Отключение показа тултипов
- **To be continued...**

Работа с бекдорами

+ HEISENBUG
// 2017 Moscow

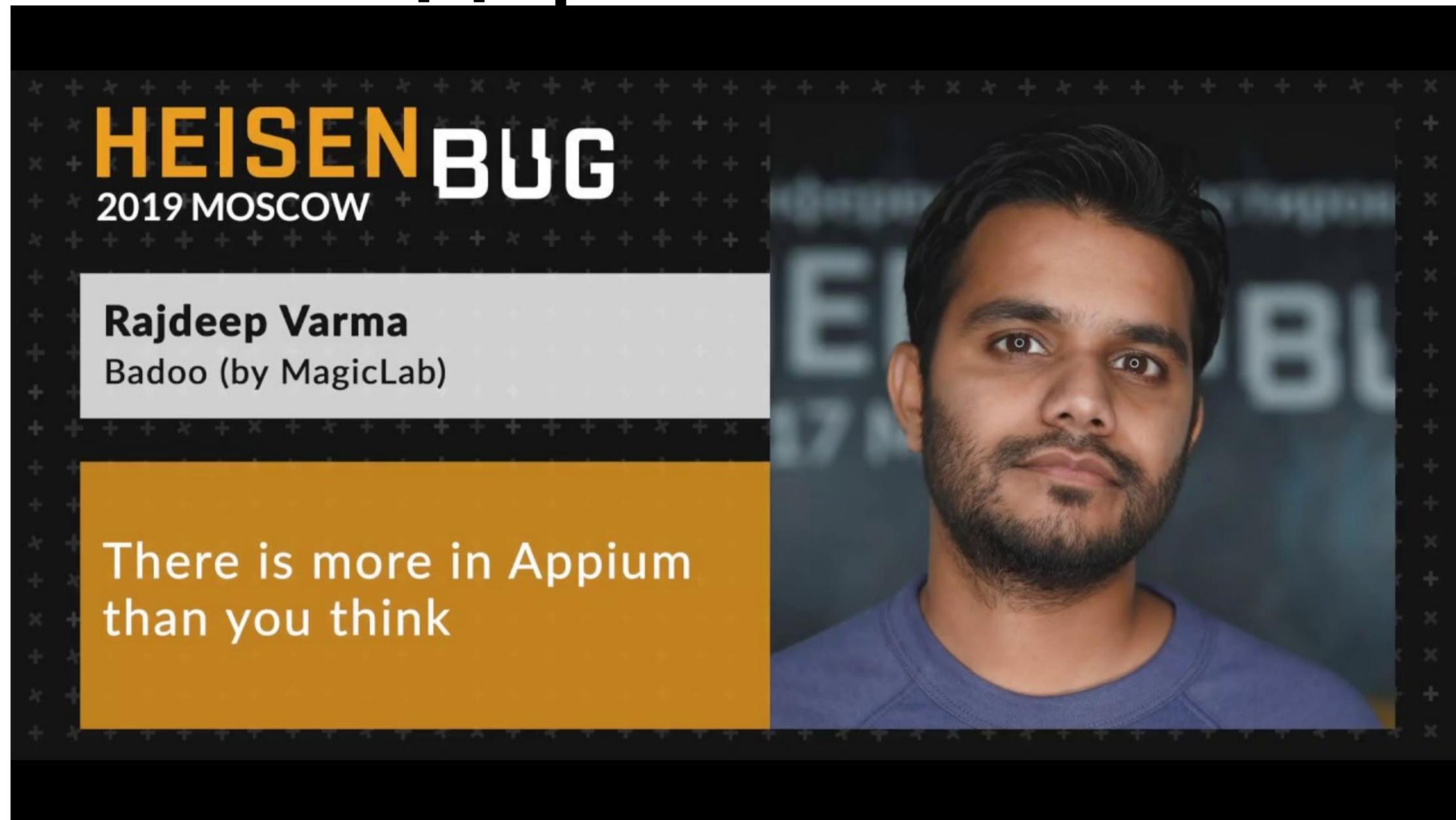
Rajdeep Varma
Badoo

Automate the impossible:
Blending the best
of Android drivers



<https://www.youtube.com/watch?v=dKHudzto4TM>

Работа с бекдорами



<https://www.youtube.com/watch?v=chfjUWbOHek>

Получение разных провайдеров

- Бэкдор для выставления MCC и MNC

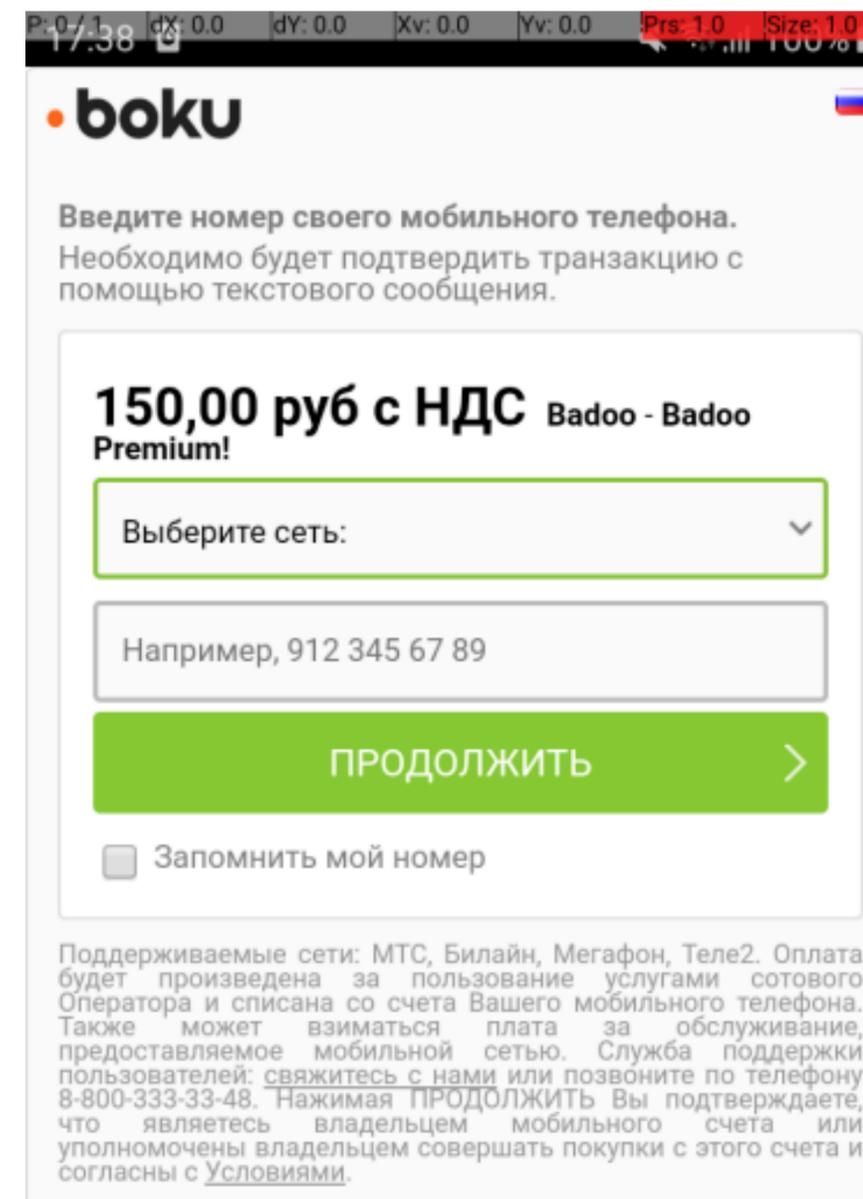
```
def set_mnc_mcc_imsi(mnc, mcc, imsi)
  qa_link(method_name: :startQAContent, arguments: ['SET_SIM_CODES', ['mcc', mcc, 'mnc', mnc, 'imsi', imsi]])
end
```

Получение разных провайдеров

- Бекдор для выставления MCC и MNC
- Android SDK эмулятор – MCC/MNC коды 310/260 (T-Mobile USA)

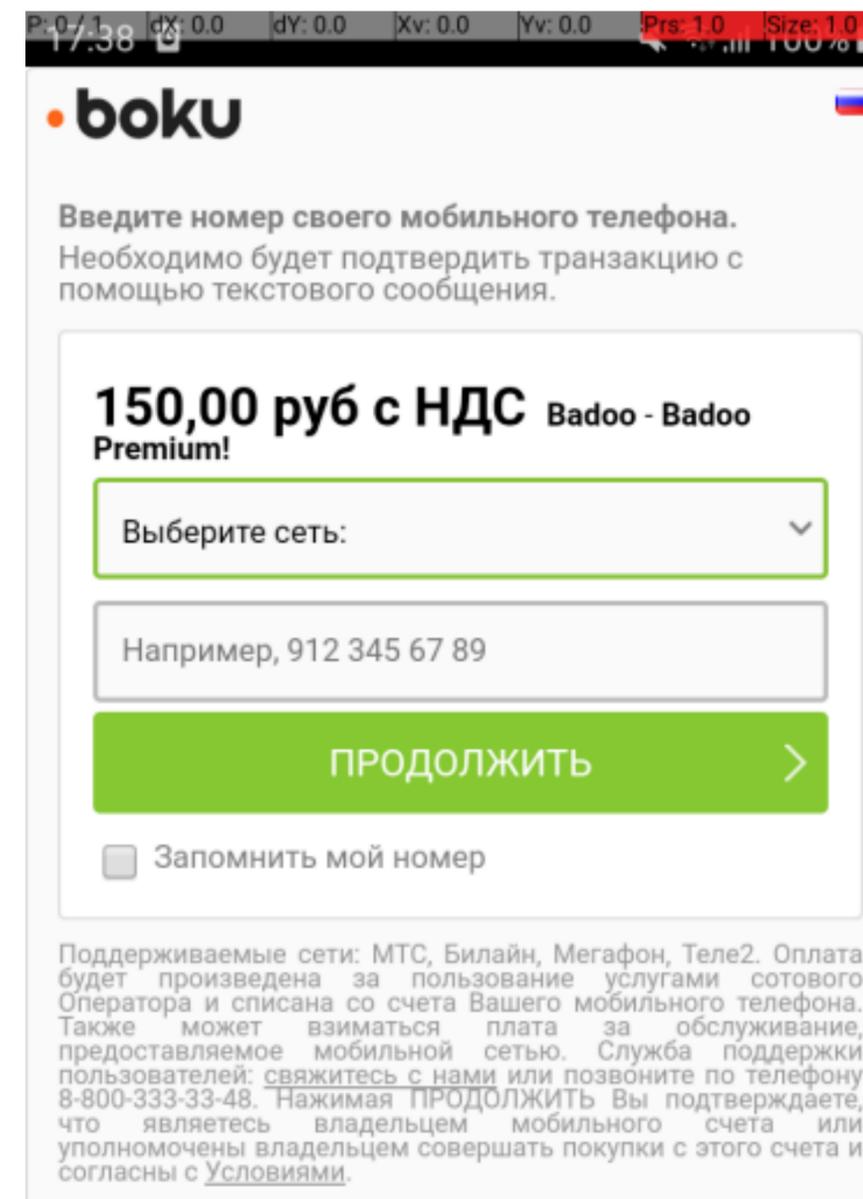
Экран оплаты

- WebView/“нативный” экран



Экран оплаты

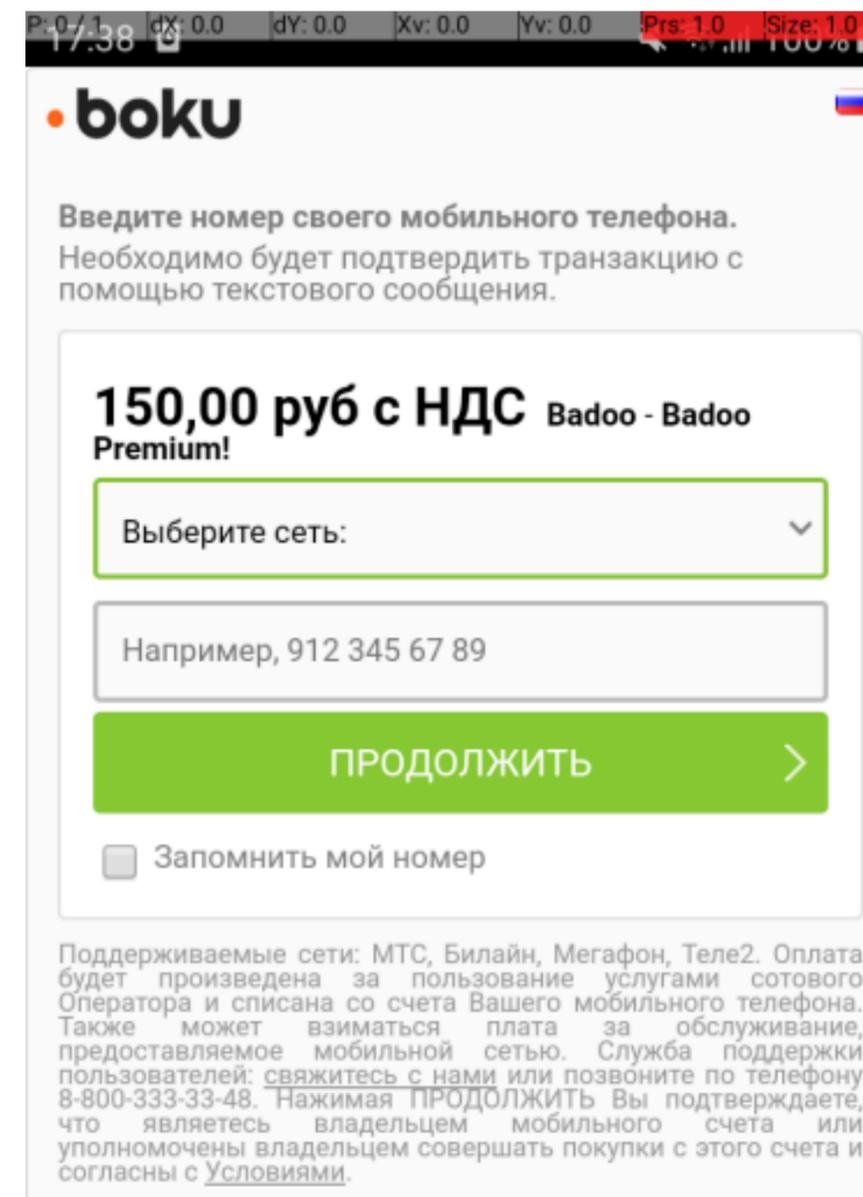
- WebView/“нативный” экран
- Есть доступ к элементам



Экран оплаты

- WebView/“нативный” экран
- Есть доступ к элементам

DESCRIPTION = "webview css: '#purchase_item'"
PRICE_CONFIRMATION = "webview css: '#purchase_price_desc'"
PHONE_INPUT = "webview css: '.ActionInput.TextInput'"
SUBSCRIBE_BUTTON = "webview css: '.ActionButton[type=\\\"submit\\\"]'"

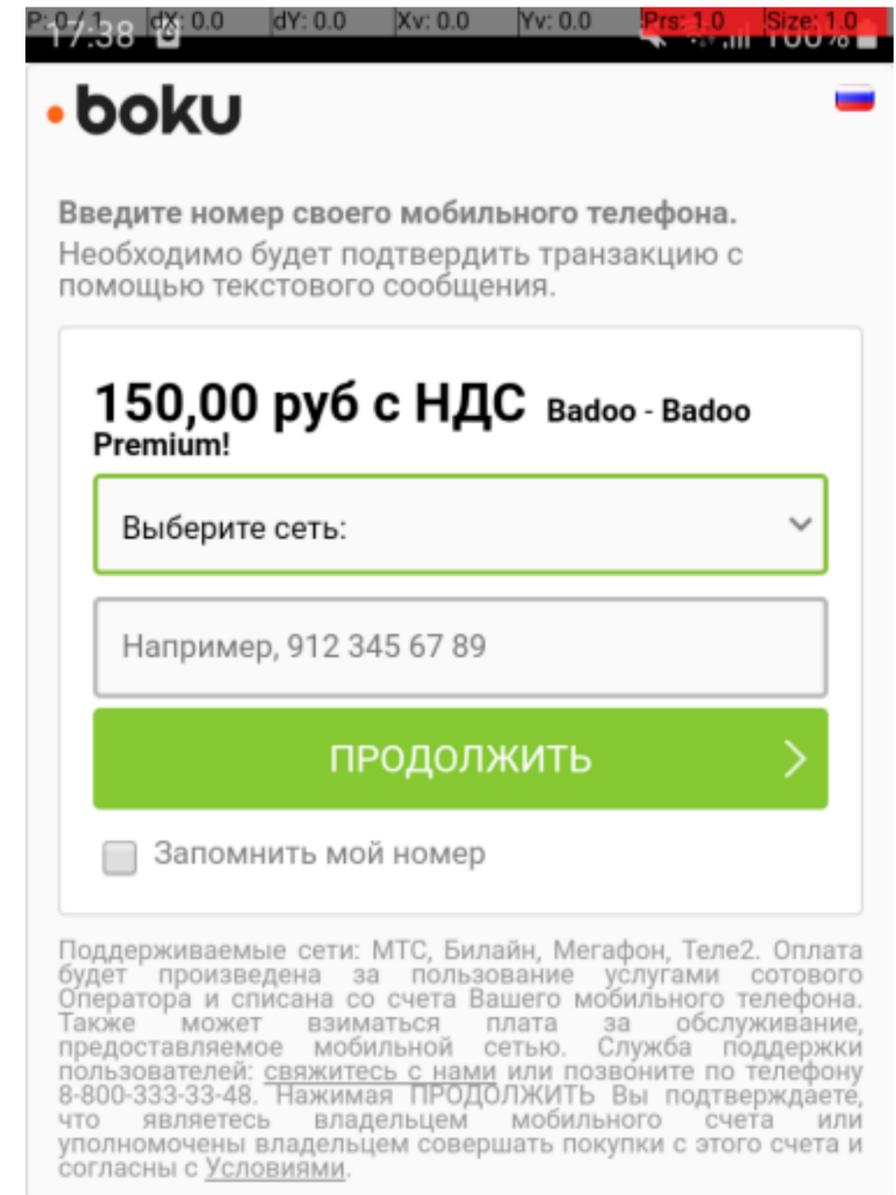


Экран оплаты

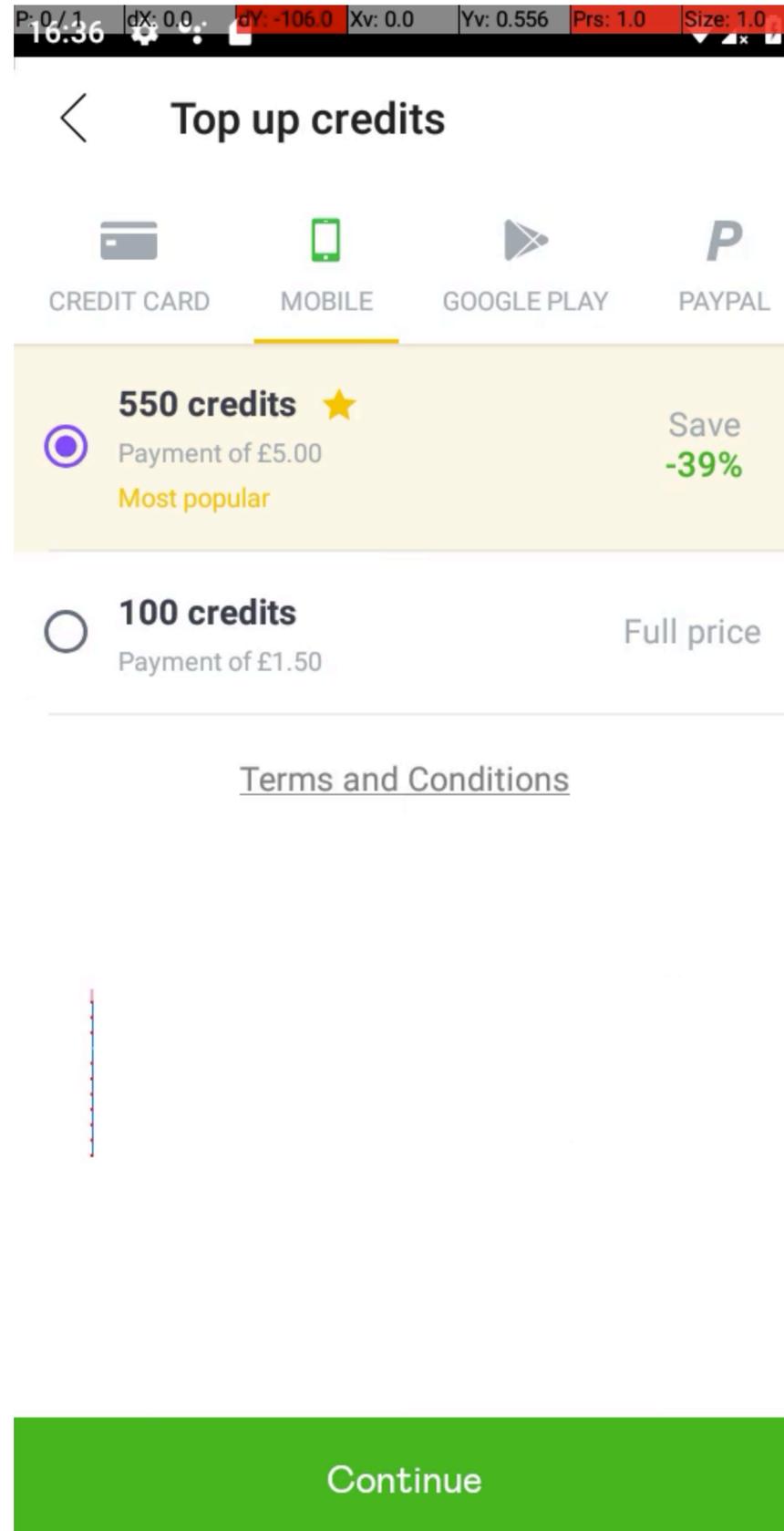
- WebView/“нативный” экран
- Есть доступ к элементам

```
DESCRIPTION = "webview css: '#purchase_item'"
PRICE_CONFIRMATION = "webview css: '#purchase_price_desc'"
PHONE_INPUT = "webview css: '.ActionInput.TextInput'"
SUBSCRIBE_BUTTON = "webview css: '.ActionButton[type=\\\"submit\\\"]'"

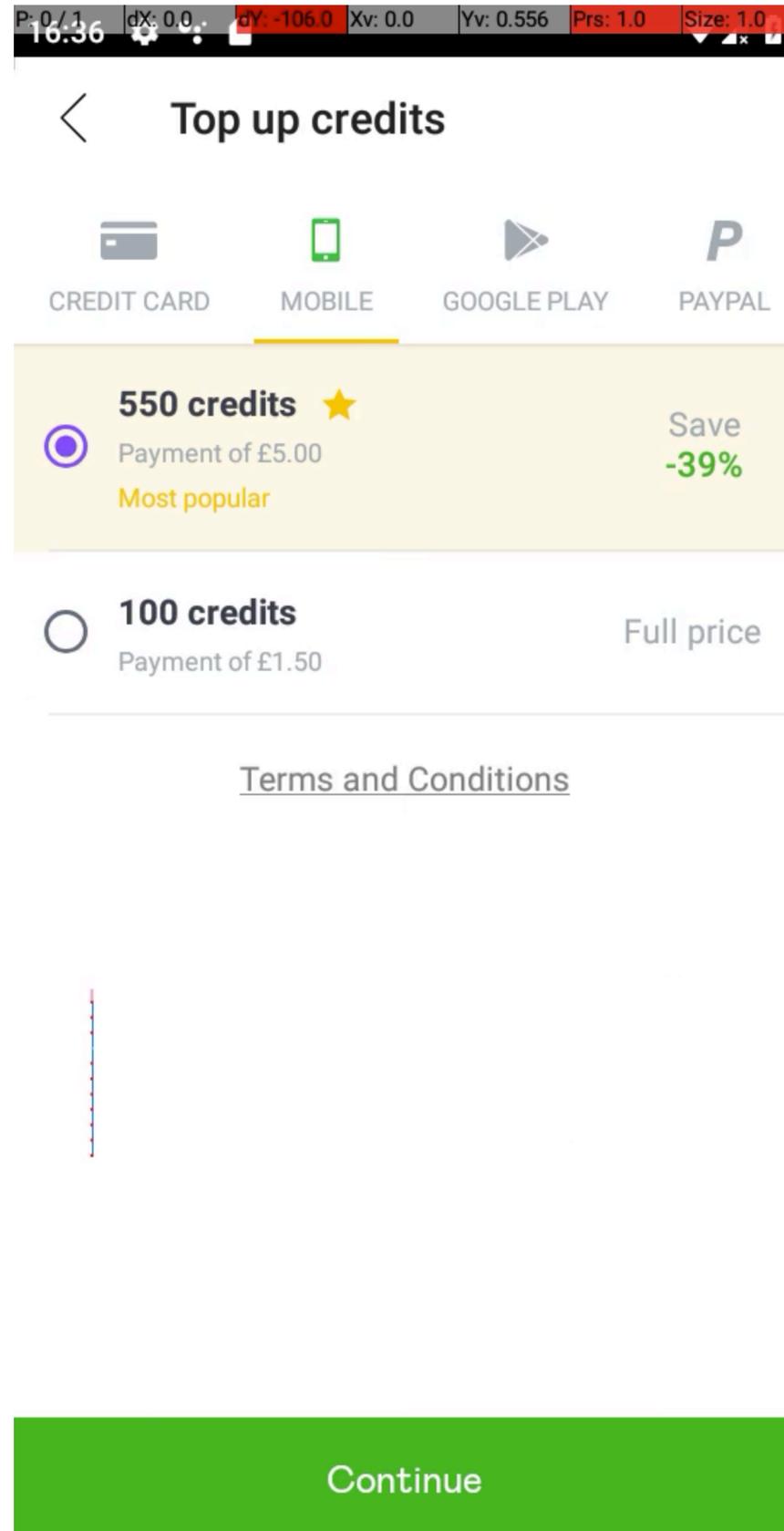
TITLE = "* contentDescription: 'display_string'"
SUBTITLE = "* contentDescription: 'payment_text'"
EXPLANATION = "* contentDescription: 'explanation'"
SUBMIT_BUTTON = "* contentDescription: 'accept_button'"
FOOTER_TEXT = "* contentDescription: 'footer_text'"
```



Платеж



Платеж



Платеж

FORTUMO_SANDBOX_MESSAGE = 'The payment is done in sandbox mode, you will not be charged.'

```
def await
```

```
  wait_for_elements_displayed([TITLE, SUBTITLE, EXPLANATION, SUBMIT_BUTTON, FOOTER_TEXT])
```

```
  assert_equal(FortumoPageLexemes::FORTUMO_SANDBOX_MESSAGE, ui.element_text(FOOTER_TEXT))
```

```
  message 'Fortumo sandbox message is not as expected'
```

```
  assert_match(FortumoPageLexemes::FORTUMO_EXPLANATION_MESSAGE, ui.element_text(EXPLANATION), message 'Explanation text is not as expected')
```

```
end
```

```
def submit_payment(_)
```

```
  tap_element(SUBMIT_BUTTON)
```

```
end
```

Исключение внешних зависимостей

- Генерация подмены receipt'ов на сервере

Исключение внешних зависимостей

- Генерация подмены receipt'ов на сервере
- **Обработка подмененного receipt'а на клиенте**

Базовая интеграция

- Открытие + наличие элементов

Базовая интеграция

- Открытие + наличие элементов

```
def await
  wait_for_elements_displayed([DESCRIPTION, PRICE_CONFIRMATION, PHONE_INPUT, SUBSCRIBE_BUTTON], timeout_message: 'Boku payment ove
  assert_match( pattern /\d+, \d+[\d]/, query(PRICE_CONFIRMATION).first['textContent'], message "Price description doesn't match t
end
```

Базовая интеграция

- Открытие + наличие элементов

```
def await
  wait for elements displayed([DESCRIPTION, PRICE_CONFIRMATION, PHONE_INPUT, SUBSCRIBE_BUTTON], timeout_message: 'Boku payment
  assert_match( pattern /^\d+, \d+[^d]/, query(PRICE_CONFIRMATION).first['textContent'], message "Price description doesn't mat
end
```

Дополнительные меры

- Unit/Integration тесты на стороне разработчиков клиента

Дополнительные меры

- Unit/Integration тесты на стороне разработчиков клиента
- **Тесты на стороне сервера**

Дополнительные меры

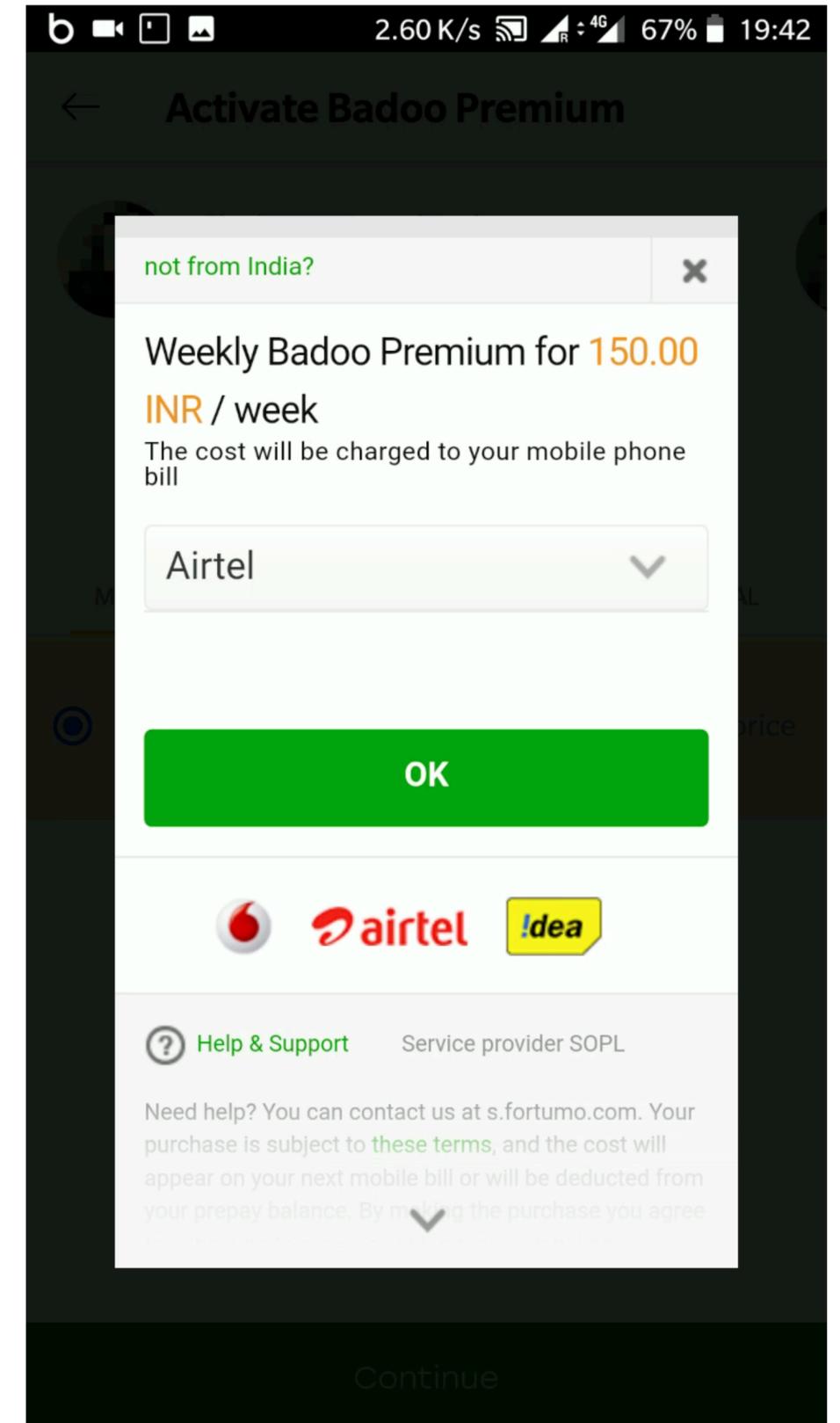
- Unit/Integration тесты на стороне разработчиков клиента
- Тесты на стороне сервера
- **Мониторинг + детекция аномалий**

Дополнительные меры

- Unit/Integration тесты на стороне разработчиков клиента
- Тесты на стороне сервера
- Мониторинг + детекция аномалий
- **Тестируем вручную**

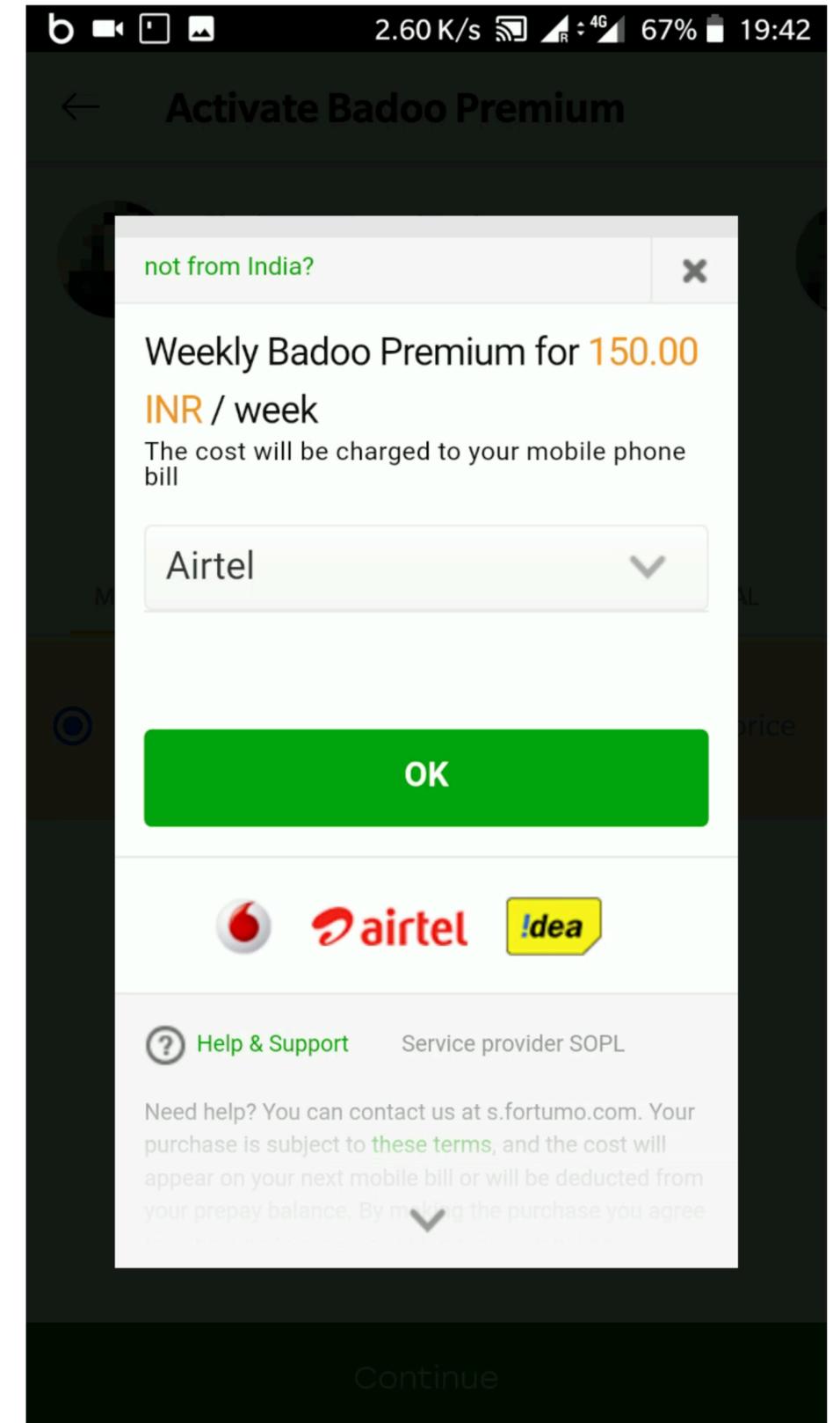
Непроходимая капча

Даже стилус
не помог



Непроходимая капча

Даже стилус
не помог





Как тестируем

Реклама

Почему это важно?

Популярна у
пользователей



Виды рекламы

- Offerwall

Виды рекламы

- Offerwall
- Rewarded video

Offerwall

- Выполнить действие

EARN MORE

IN PROGRESS

Earn



Coin Master
Complete third Village

1,105

TODAY'S BEST!



Stars™ Slots Casino - Play With
Complete level 130

3,188



Stars™ Slots Casino - Play With
Complete level 110

2,338



Huuuge Stars™ Slots Casino Games
Complete level 100

1,913



Solitaire - Grand Harvest
Finish crop 6!

510



Board Kings
Install and Finish level 3

128

[Privacy policy](#) [Terms of use](#)

Offerwall

- Выполнить действие
- Скачать приложение

EARN MORE

IN PROGRESS

Earn

	Coin Master Complete third Village	1,105
---	---------------------------------------	-------

TODAY'S BEST!



	Stars™ Slots Casino - Play With Complete level 130	3,188
---	---	-------

	Stars™ Slots Casino - Play With Complete level 110	2,338
--	---	-------

	Huuuge Stars™ Slots Casino Games Complete level 100	1,913
---	--	-------

	Solitaire - Grand Harvest Finish crop 6!	510
---	---	-----

	Board Kings Install and Finish level 3	128
---	---	-----

Offerwall

- Выполнить действие
- Скачать приложение
- Пройти уровни

EARN MORE

IN PROGRESS

Earn

	Coin Master Complete third Village	1,105
---	---------------------------------------	-------

TODAY'S BEST!



	Stars™ Slots Casino - Play With Complete level 130	3,188
---	---	-------

	Stars™ Slots Casino - Play With Complete level 110	2,338
--	---	-------

	Huuuge Stars™ Slots Casino Games Complete level 100	1,913
---	--	-------

	Solitaire - Grand Harvest Finish crop 6!	510
---	---	-----

	Board Kings Install and Finish level 3	128
---	---	-----

Offerwall

- Открываем и верифицируем

```
When I select Download an app Freebie option on Credits Generic page
And I tap on buy button
Then I am on Offerwall page
And I get 100 credits via fake Offerwall provider notification
When I go back from Offerwall page
And I go back from Credits Generic page
And I wait until credit balance is 100
```

Offerwall

- Генерируем подмененную нотификацию (QaApi)

When I select **Download an app** Freebie option on **Credits Generic** page

And I tap on buy button

Then I am on **Offerwall** page

And I get **100 credits** via fake Offerwall provider notification

When I go back from **Offerwall** page

And I go back from **Credits Generic** page

And I wait until credit balance is **100**

Offerwall

- Проверяем выдачу сервиса

When I select **Download an app** Freebie option on **Credits Generic** page

And I tap on buy button

Then I am on **Offerwall** page

And I get **100 credits** via fake Offerwall provider notification

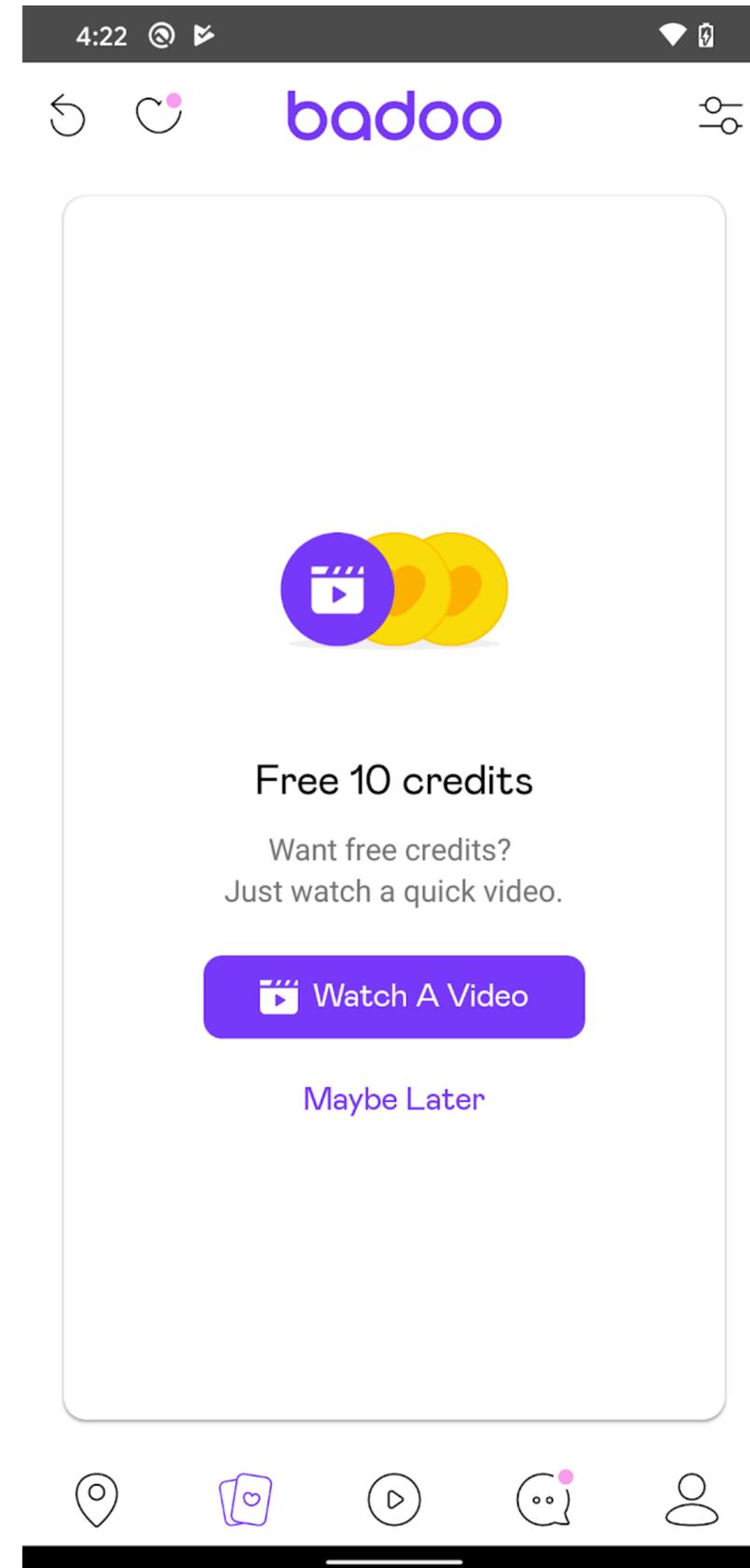
When I go back from **Offerwall** page

And I go back from **Credits Generic** page

And I wait until credit balance is **100**

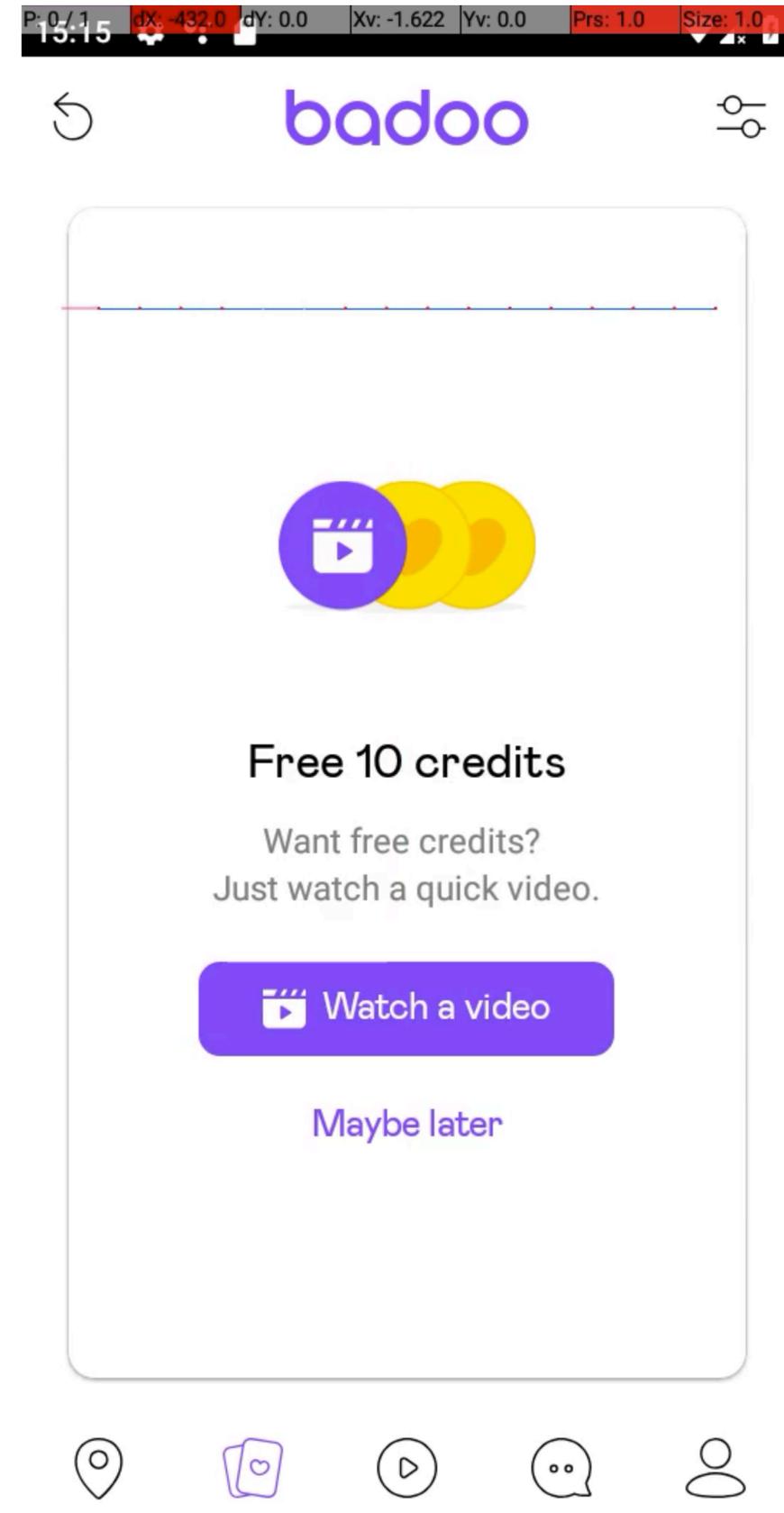
Rewarded video

Просмотр ВИДЕО



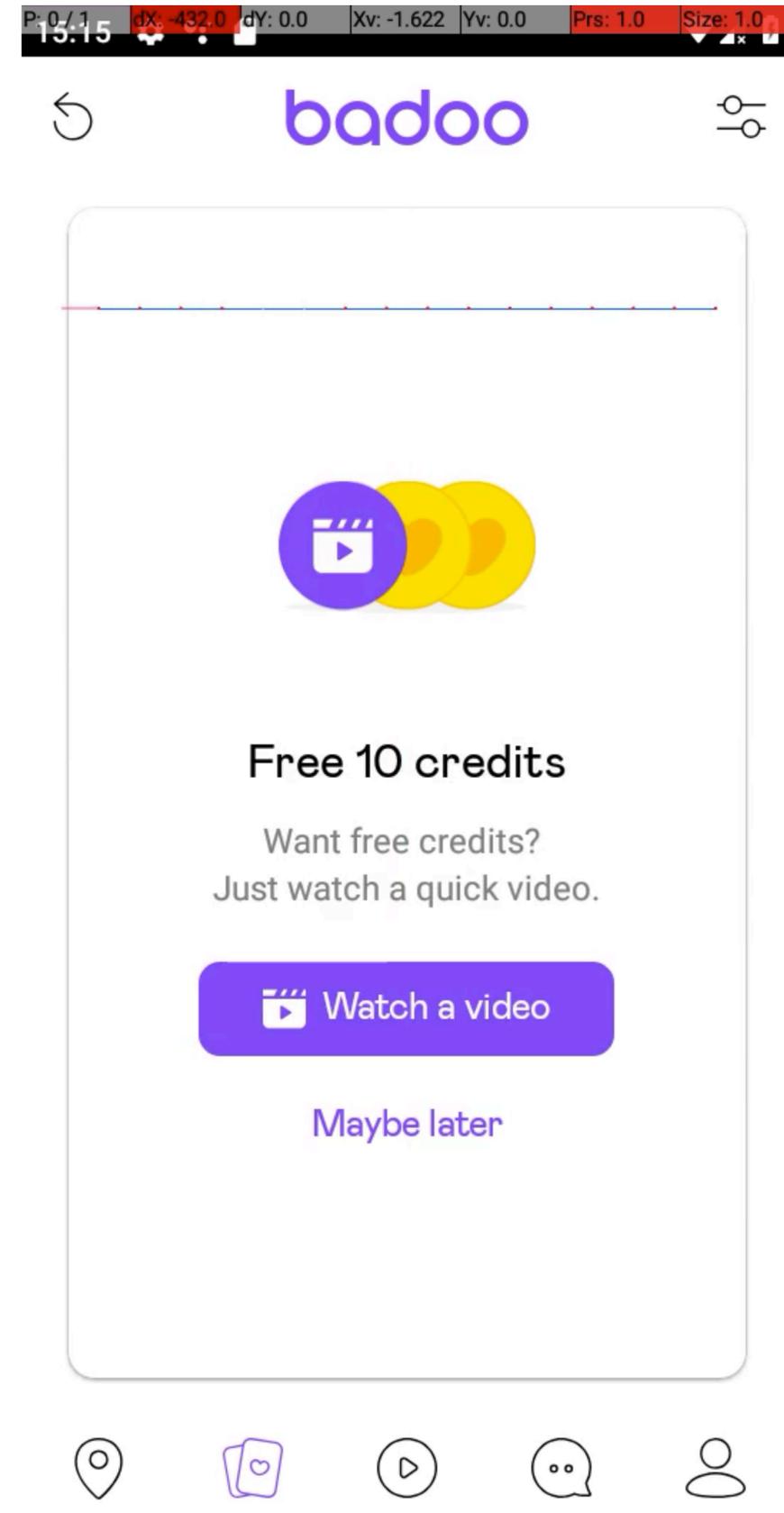
Rewarded video

- Получить видео



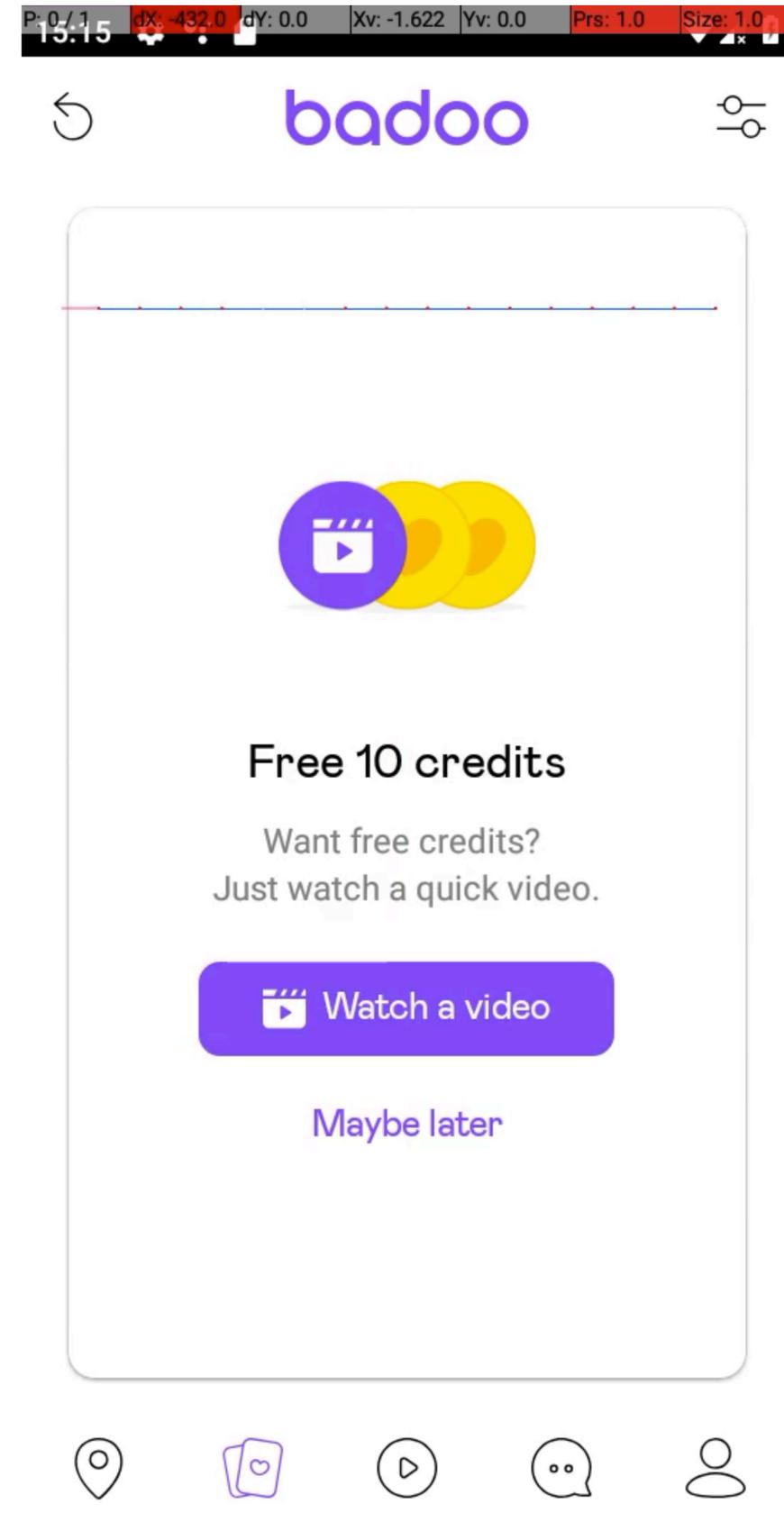
Rewarded video

- Получить видео



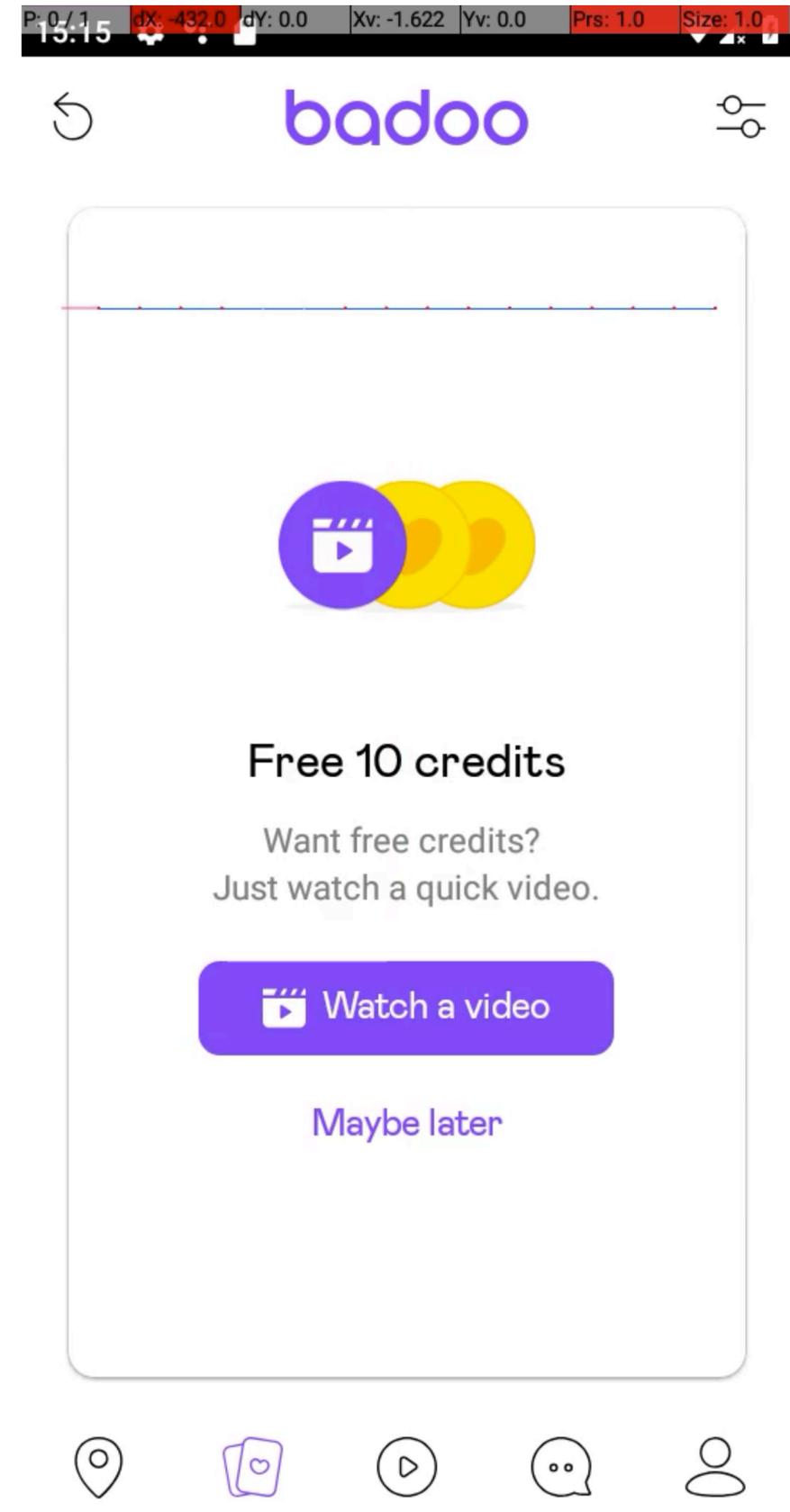
Rewarded video

- Получить видео
- Открыть видео



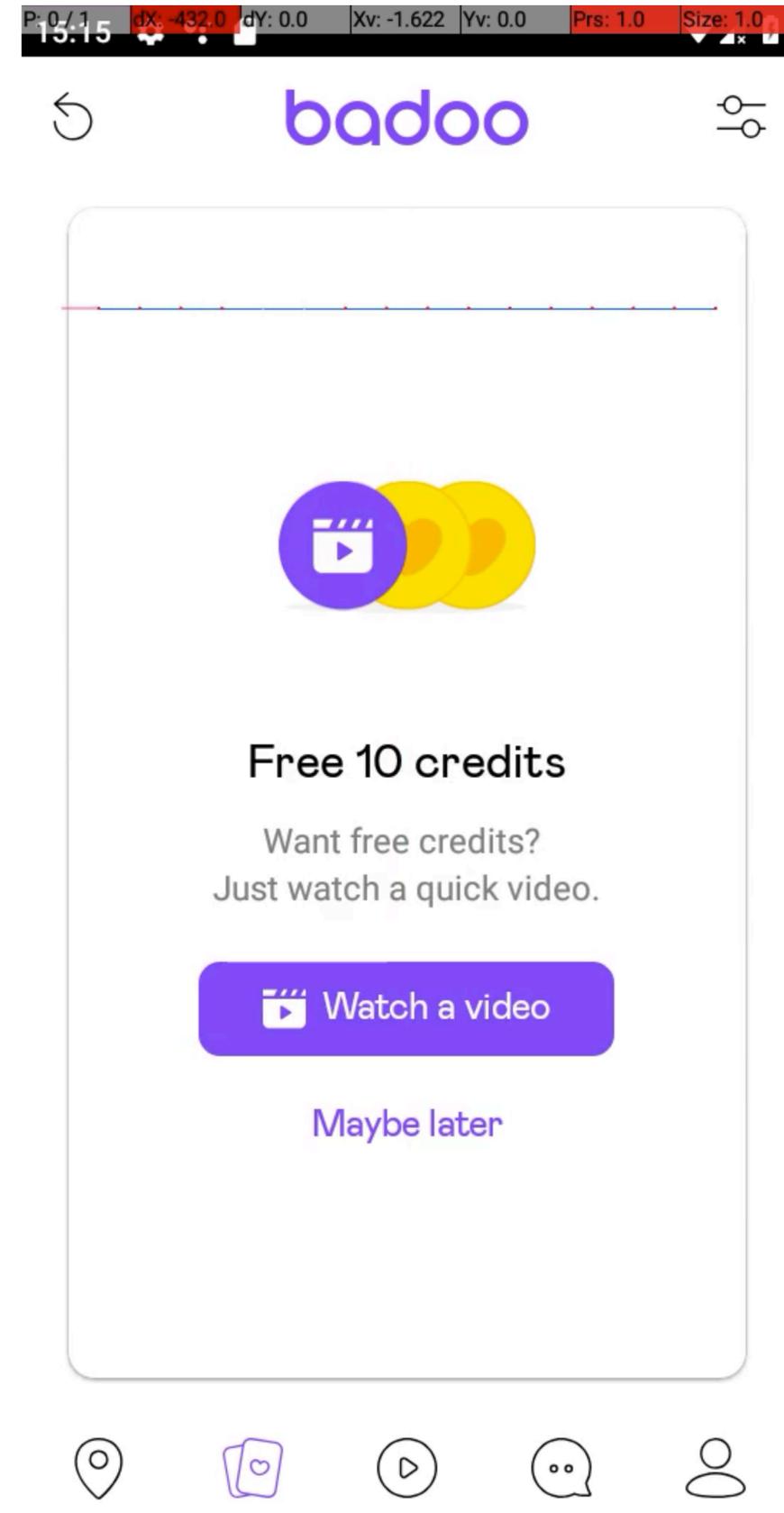
Rewarded video

- Получить видео
- Открыть видео
- **Заккрыть видео после завершения**



Rewarded video

- Получить видео
- Открыть видео
- Закрывать видео после завершения
- Проверить выдачу сервиса



Rewarded video. Проблемы

- **Получить видео**
- Открыть видео
- **Закрыть видео после завершения**
- Проверить выдачу сервиса

Rewarded video. Доступность

- SDK не возвращает видео

Rewarded video. Доступность

- SDK не возвращает видео
- SDK не успевает инициализироваться

Rewarded video. Доступность

- Google Advertising ID – рекламный идентификатор телефона

Rewarded video. Доступность

- Google Advertising ID

Scenario: Credits are added after watching rewarded video on Encounters

Given I reset Ads id on Android in system settings

```
def reset_google_advertising_id
  device.launch_adsettings_activity
  Poll.for_true(timeout_message: 'Timed out waiting for "Ads" text on screen') { screen_texts.include?('Ads') }
  uia_touch_text('Reset advertising ID')
  uia_touch_text('OK')
end
```

Rewarded video. Доступность

- Бэкдоры

Rewarded video. Доступность

- Бэkdоры
 - `isRewardedVideoInitialized()`

Rewarded video. Доступность

- Бэkdоры
 - `isRewardedVideoInitialized()`
 - **`isRewardedVideoAvailable()`**

Rewarded video. Доступность

- Бэkdоры
 - `isRewardedVideoInitialized()`
 - `isRewardedVideoAvailable()`
 - **`isRewardedVideoInitialized() && isRewardedVideoAvailable()`**

Rewarded video. Доступность

- Бэкдоры
 - `isRewardedVideoInitialized()`
 - `isRewardedVideoAvailable()`
 - `isRewardedVideoInitialized() && isRewardedVideoAvailable()`

Scenario: Credits are added after watching rewarded video on Encounters

Given I reset Ads id on Android in system settings

And I authenticate a new **male** user with photo

And I wait for Rewarded video to be ready via backdoor

Rewarded video. Заккрытие

- Когда-то имеем доступ к элементам

Rewarded video. Заккрытие

- Когда-то имеем доступ к элементам
- **Когда-то не имеем**

Rewarded video. Заккрытие

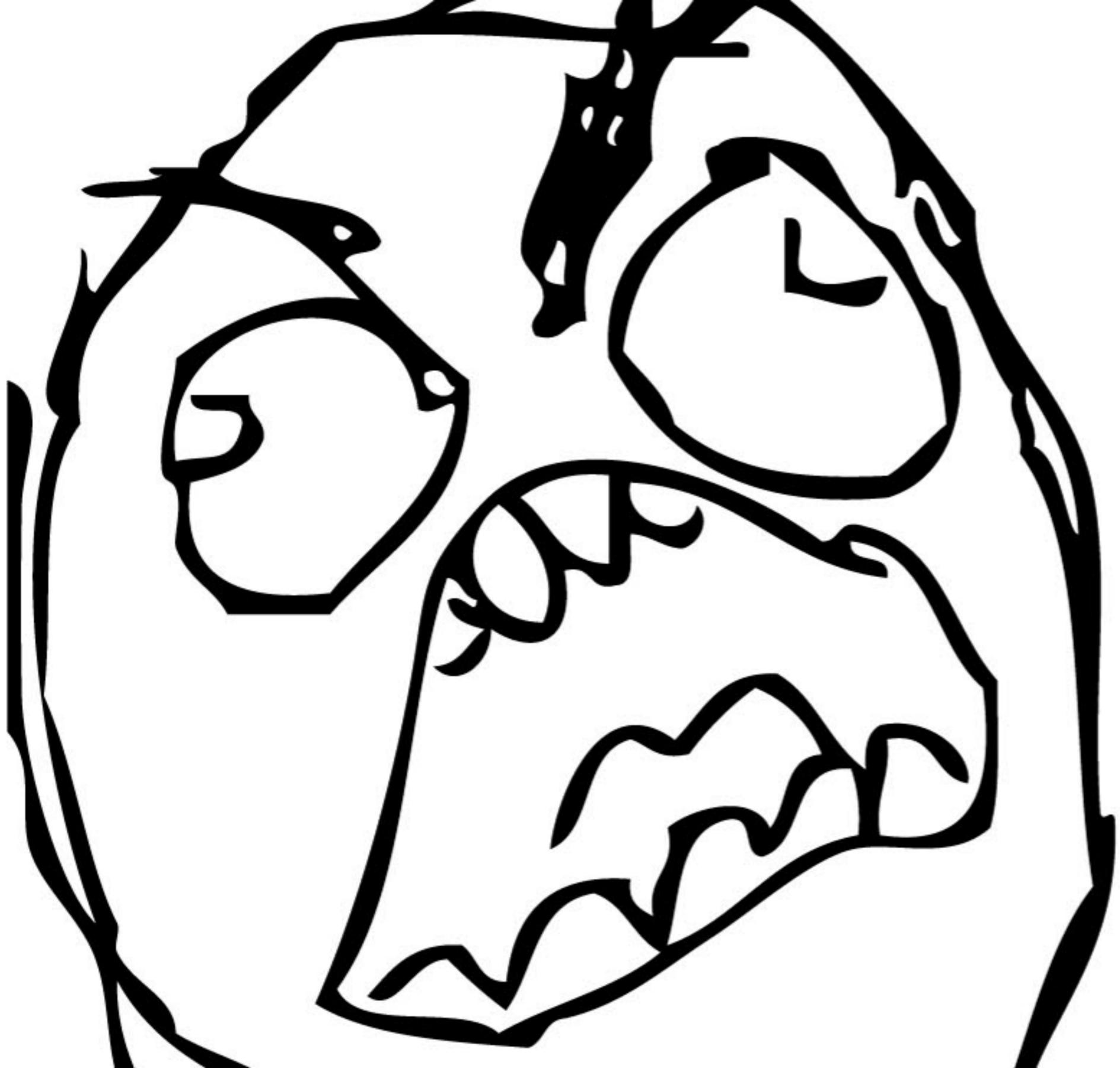
- Когда-то имеем доступ к элементам
- Когда-то не имеем
- **Когда-то закрывается автоматически**

Rewarded video. Заккрытие

- Когда-то имеем доступ к элементам
- Когда-то не имеем
- Когда-то закрывается автоматически
- **Когда-то можно закрыть, не досмотрев**

Rewarded video. Заккрытие

- Когда-то имеем доступ к элементам
- Когда-то не имеем
- Когда-то закрывается автоматически
- Когда-то можно закрыть, не досмотрев
- **Когда-то нужно закрывать диалоги**



FFFFFFF

FFFFFFF

FFFFFFF

FFFUU

UUUU

UUUU

UUUU

UUUU

UUUU-

And I tap on button to watch Rewarded video on **Encounters** screen

And I accept GDPR alert if present

And I wait until Rewarded video finishes and close it

```
def accept_gdpr_alert_if_present
  if gdpr_alert_displayed?
    if elements_displayed?(GDPR_ALERT_BUTTON)
      tap_element(GDPR_ALERT_BUTTON)
    else
      acknowledge_alert_if_present
    end
  end
end
end
```

And I tap on button to watch Rewarded video on **Encounters** screen

And I accept GDPR alert if present

And I wait until Rewarded video finishes and close it

```
def close_video
```

```
  wait_for_elements_not_displayed([VIDEO_CONTAINER, COUNT_DOWN_1, COUNT_DOWN_2, 'ProgressBar'])
```

```
  Poll.for(timeout: WAIT_A_MINUTE, timeout_message: 'Timed out waiting for video to be closed') do
```

```
    press_hard_back_button if elements_displayed?('android.webkit.WebView') # some adv. are autoclose
```

```
    if elements_displayed?(RESUME_VIDEO)
```

```
      tap_element(RESUME_VIDEO)
```

```
      sleep(WAIT_MEDIUM)
```

```
    next
```

```
  end
```

```
  Poll.for(timeout: WAIT_SMALL, return_on_timeout: true) { !elements_displayed?('android.webkit.WebView') }
```

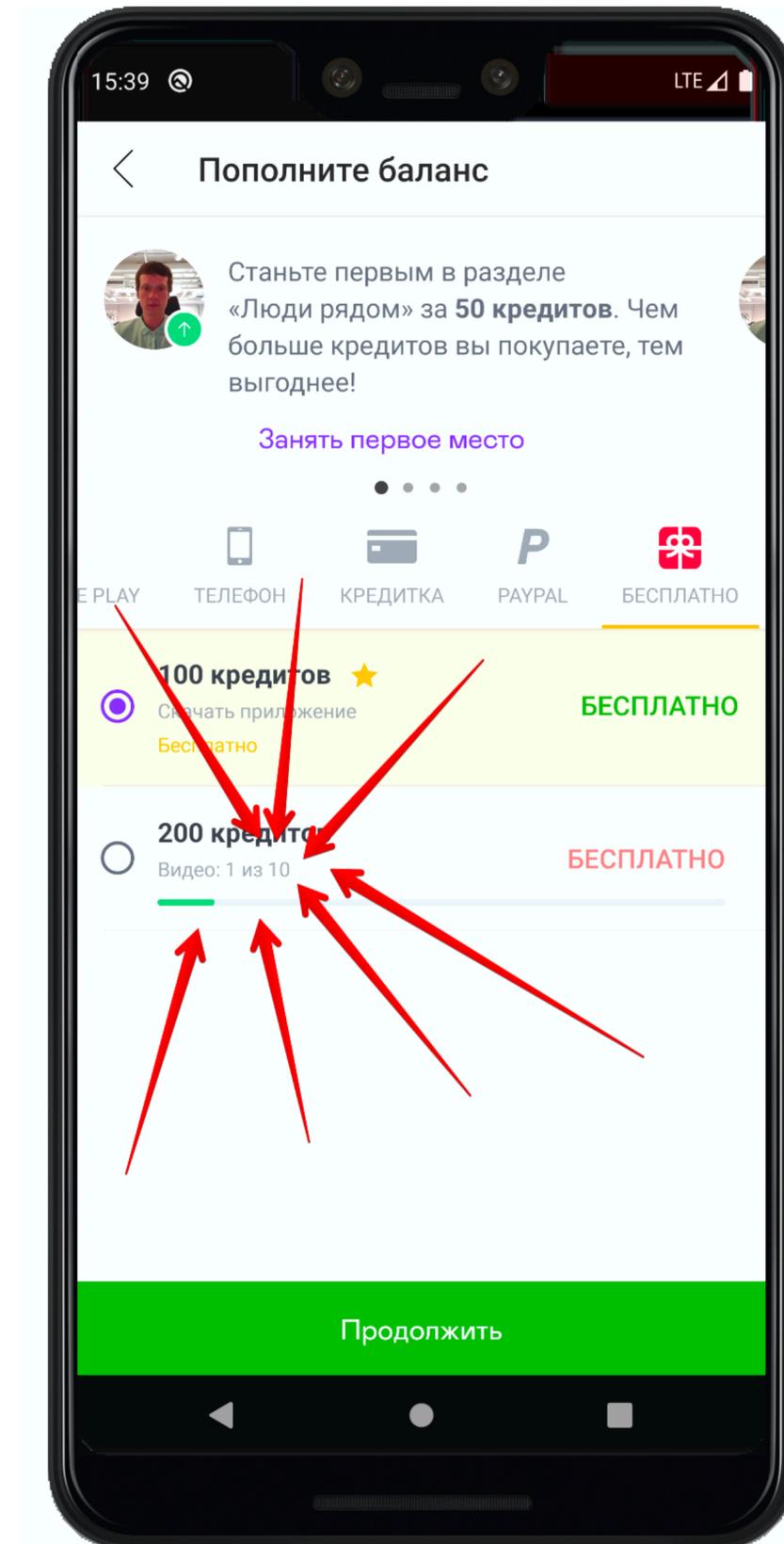
```
end
```

```
end
```



И ТАК
СОИДЕТ!

Несколько видео



ИТОГИ

Полезные идеи

- Комбинировать методы тестирования

Полезные идеи

- Комбинировать методы тестирования
- **1 интеграция – 1 тестовая конфигурация**

Полезные идеи

- Комбинировать методы тестирования
- 1 интеграция – 1 тестовая конфигурация
- **Ограничения фреймворка могут быть преодолимы**

Полезные идеи

- Комбинировать методы тестирования
- 1 интеграция – 1 тестовая конфигурация
- Ограничения фреймворка могут быть преодолимы
- **Автоматизация рекламы нужна, но не всегда стабильна**





CHEERS!

dmitrii.makarenko@magiclab.co
tech.badoo.com



ССЫЛКИ



© 2020