

ozon банк

**Тестируем UI
банкоматов**

ozon банк



Максим Якубовский

Руководжу отделом тестирования B2C сервисов

Банкоматы в Ozon

600+

банкоматов

30+

Городов присутствия

5000

клиентов ежедневно

6 месяцев

запустили первый банкомат

Банкоматы в Ozon

600+

банкоматов

30+

Городов присутствия

5000

клиентов ежедневно

№1

в России выдаем карты

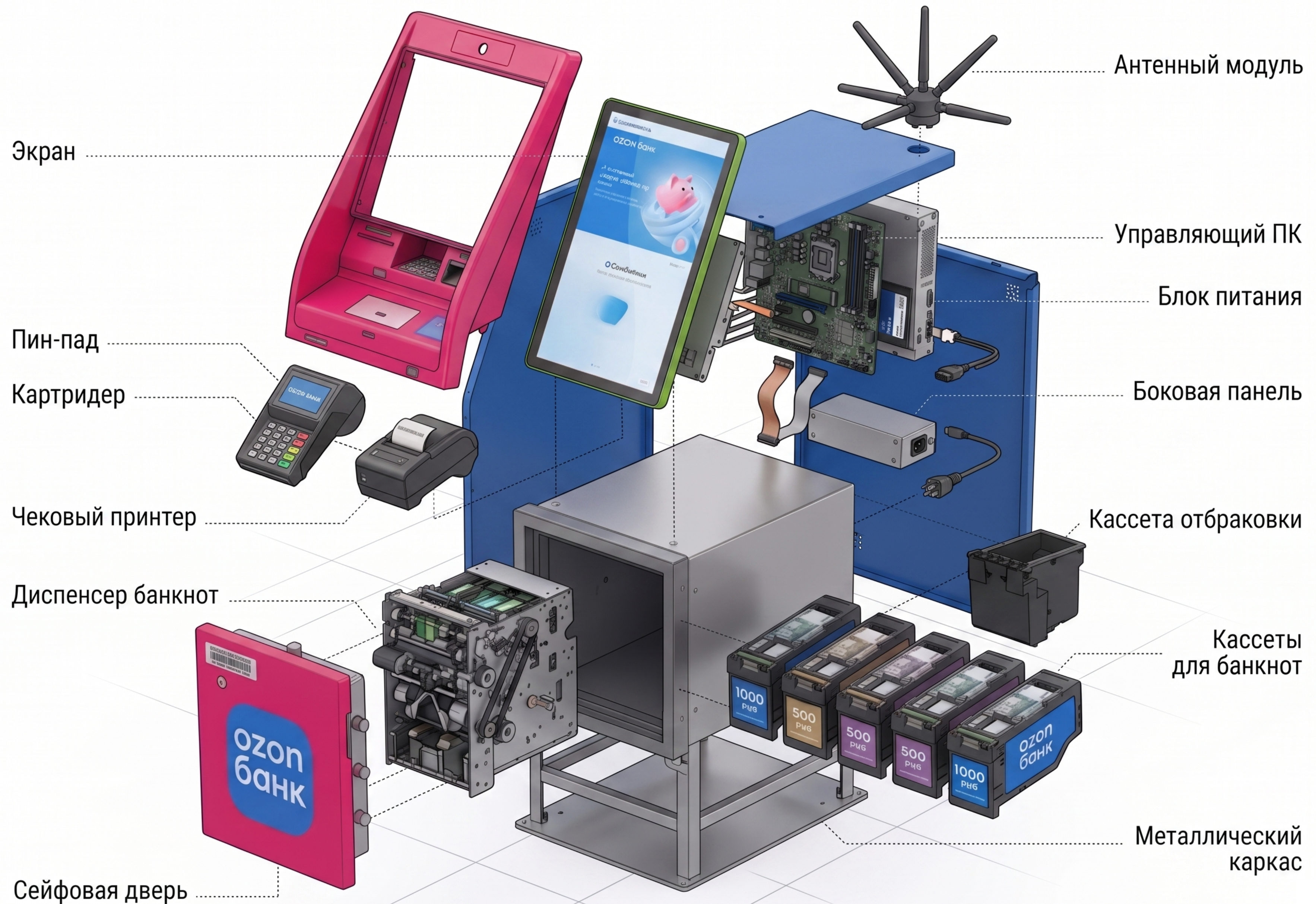
6 месяцев

запустили первый банкомат

2000+

карт выдали

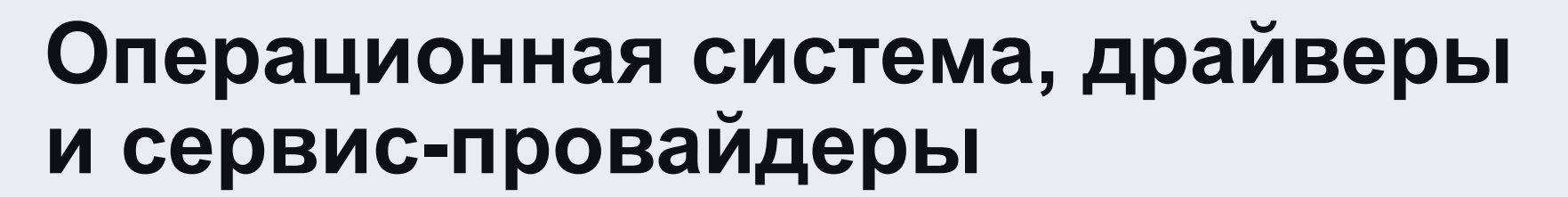




Софт



Драйверы и ОС

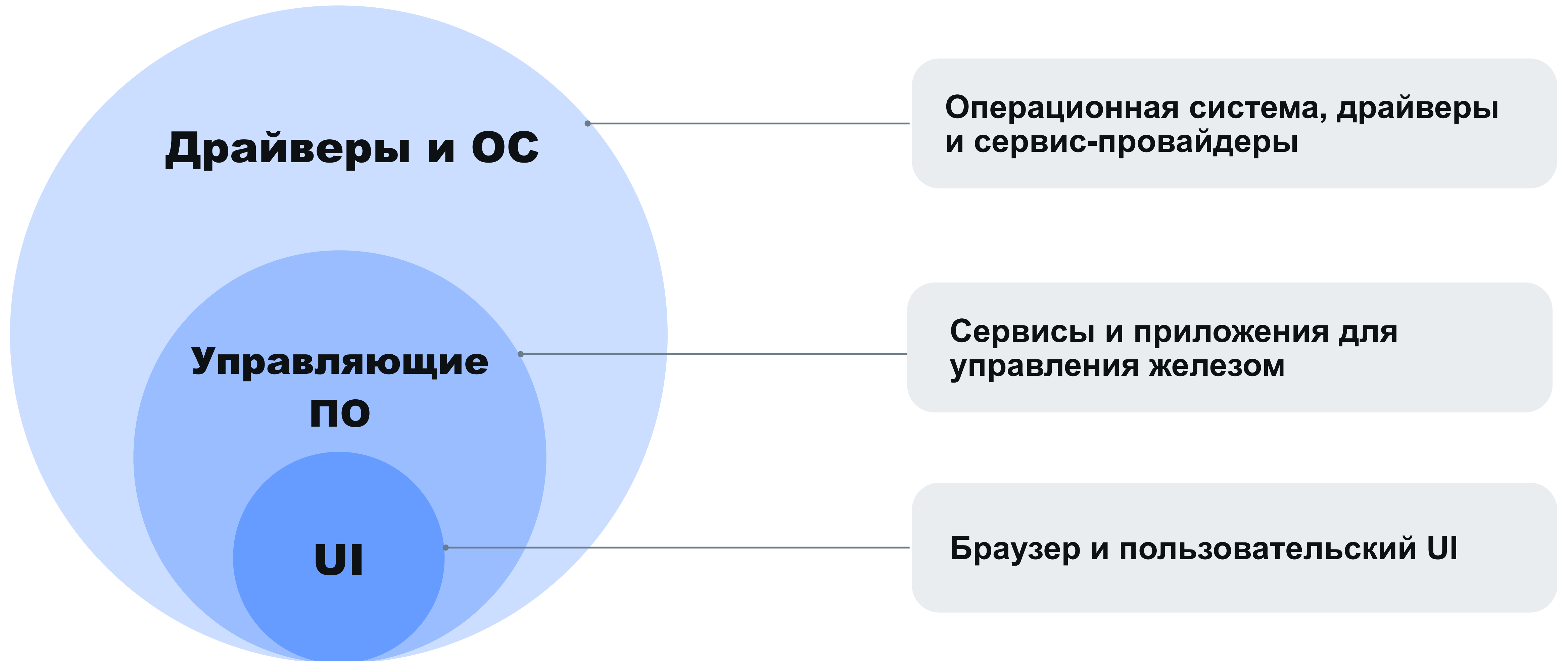


**Операционная система, драйверы
и сервис-провайдеры**

Софт



Софт

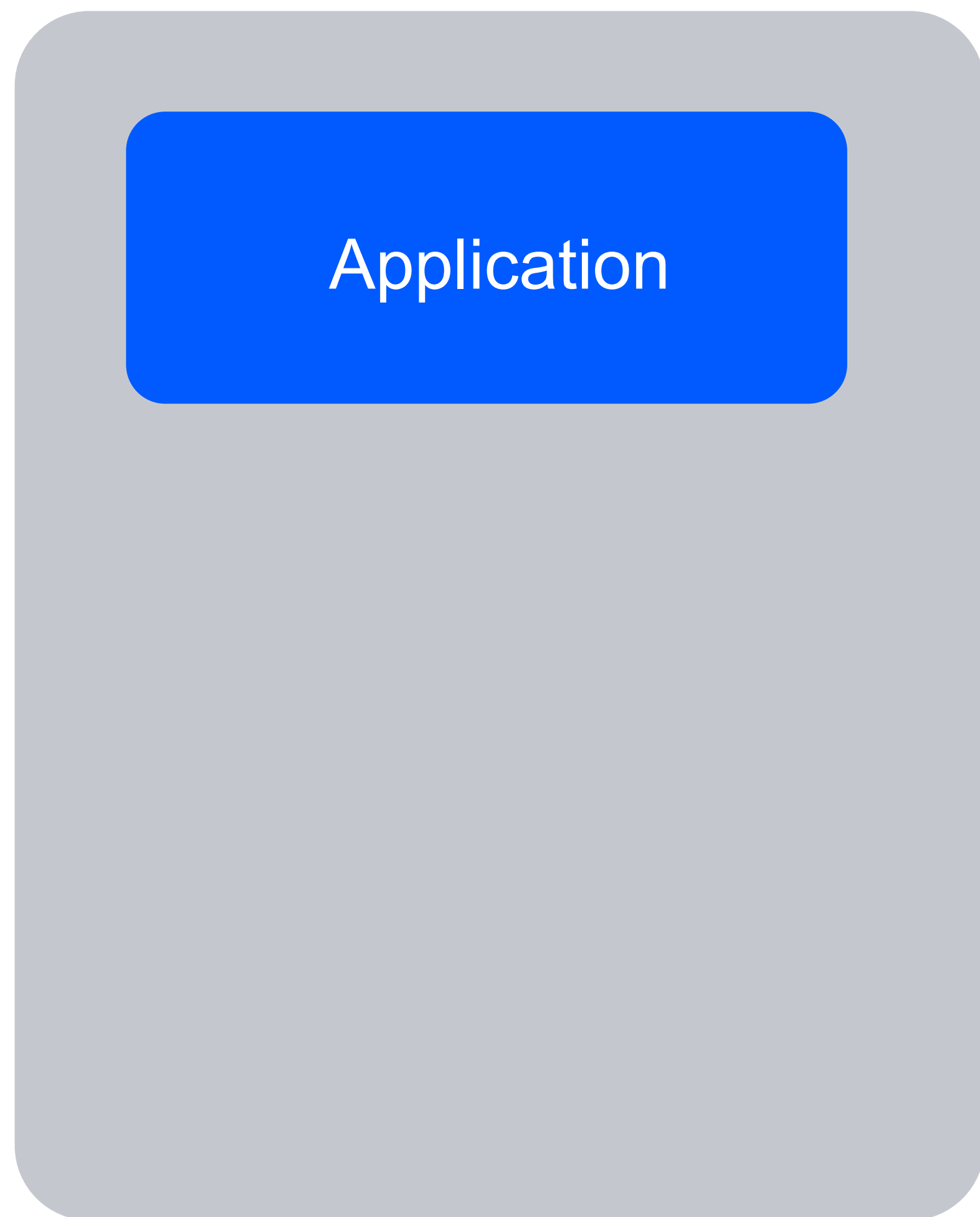




Как работает UI банкомата?

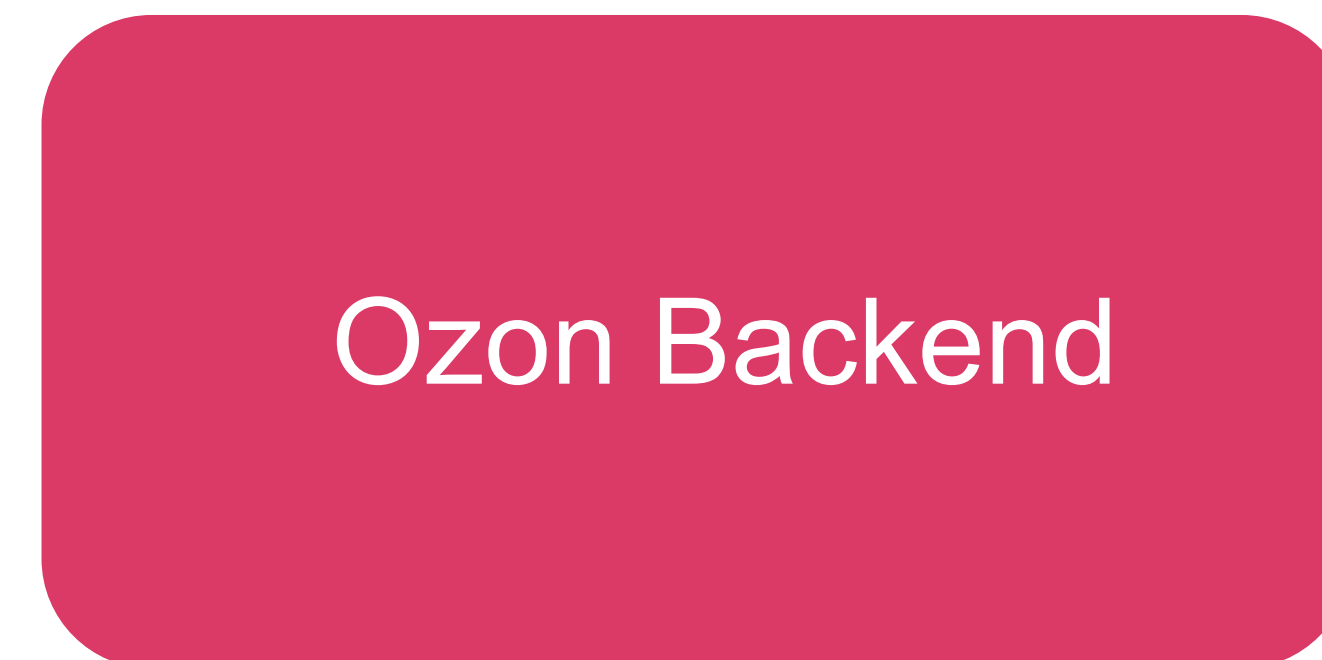
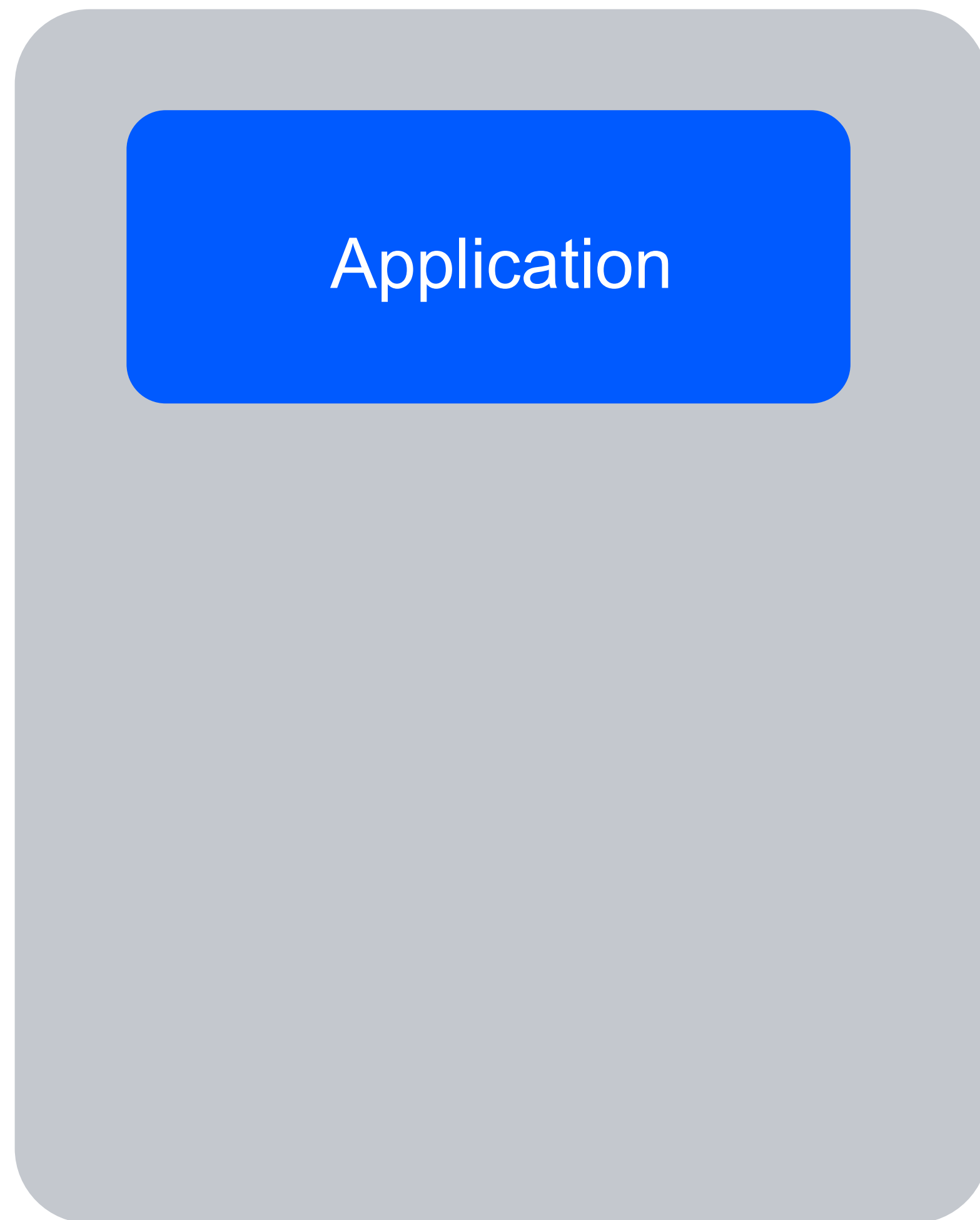
Как работает UI

Browser

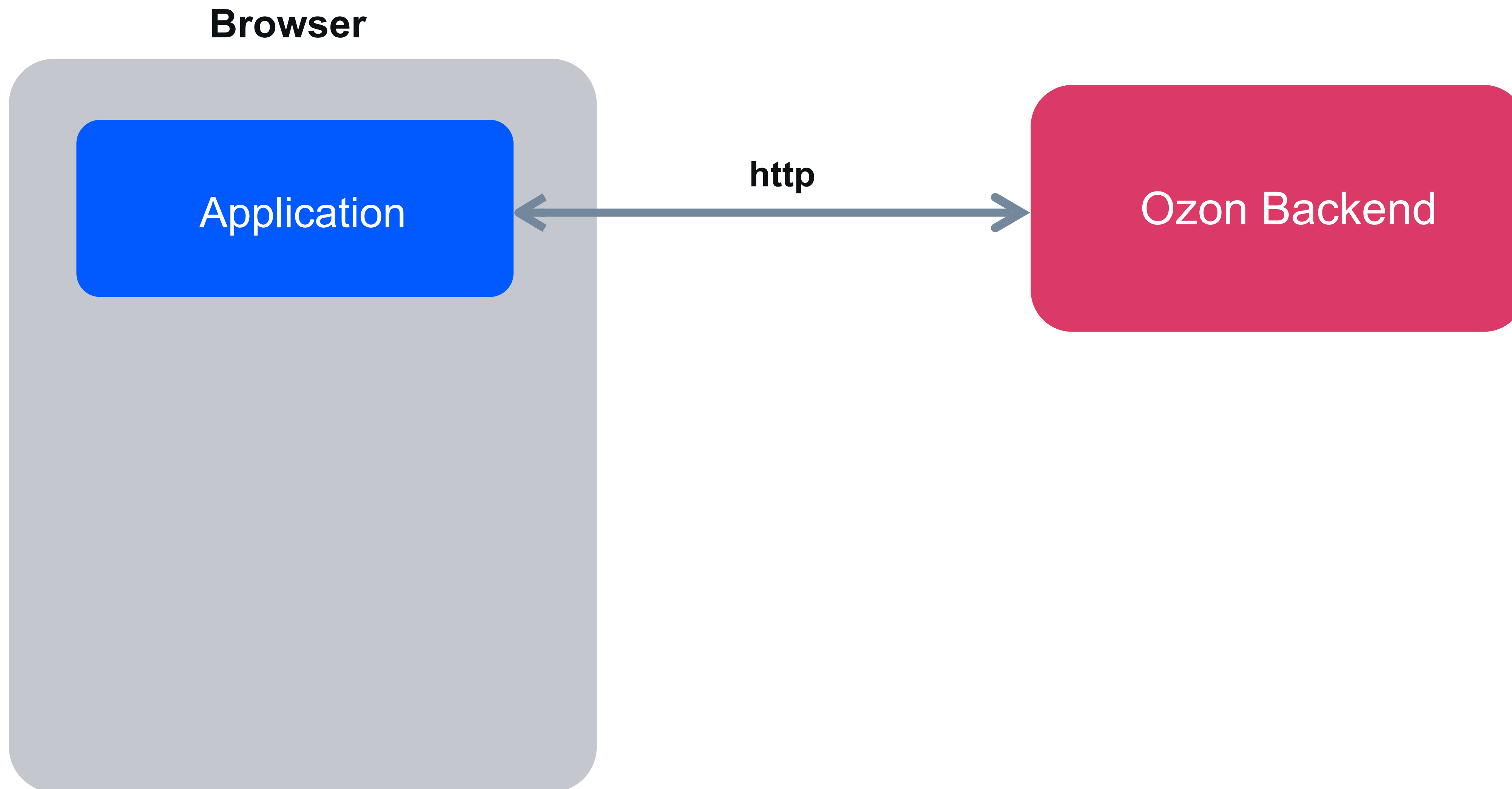


Как работает UI

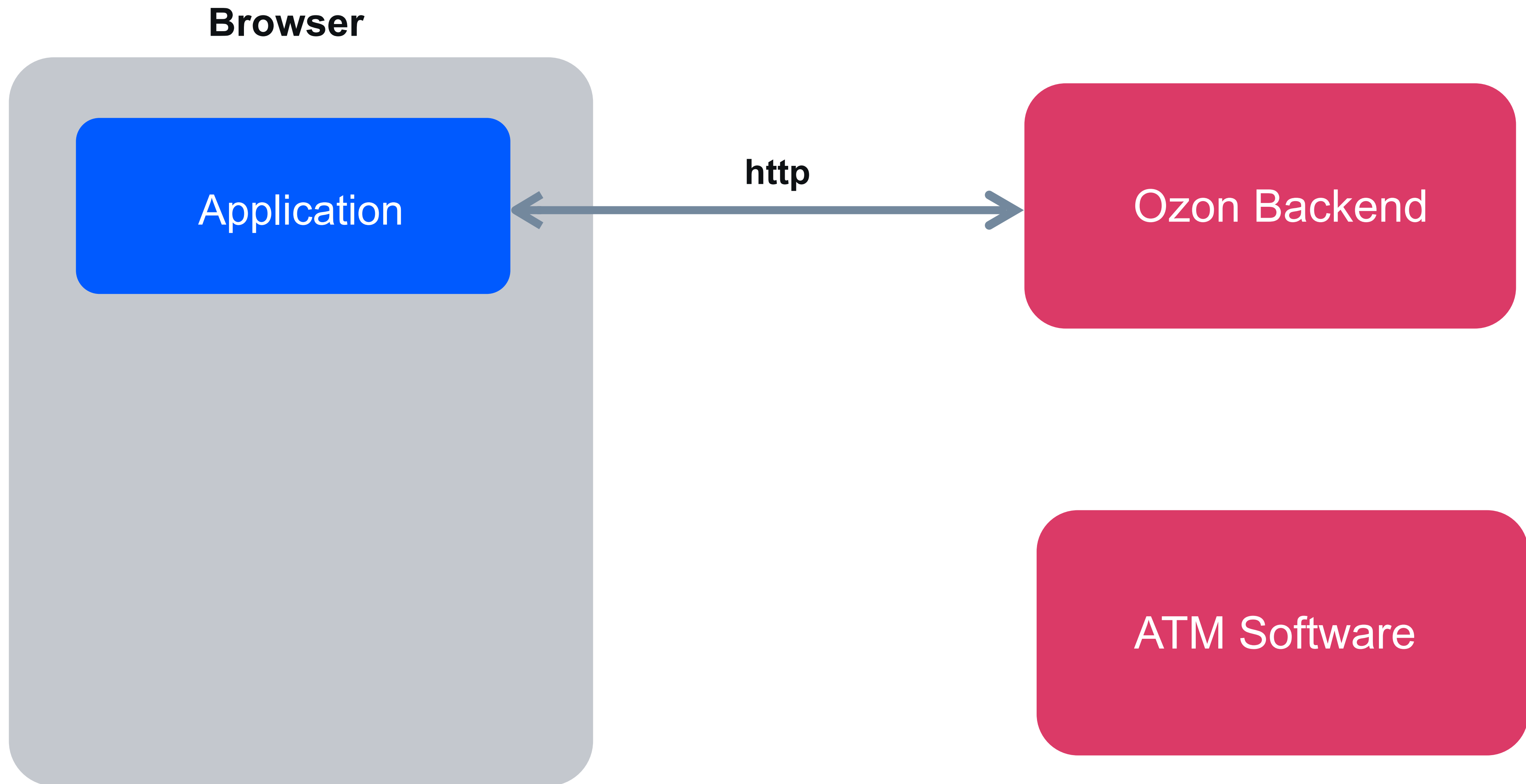
Browser



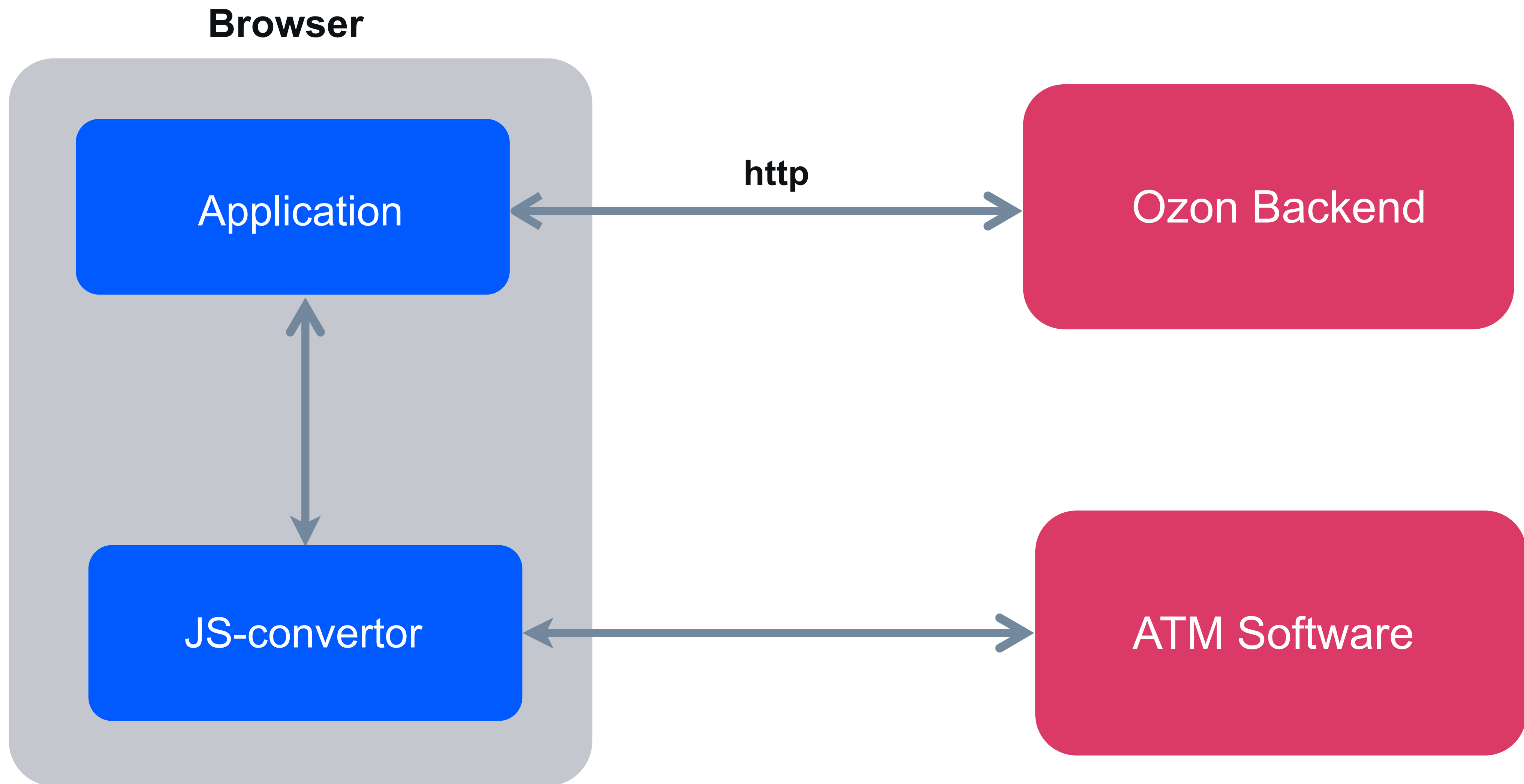
Как работает UI



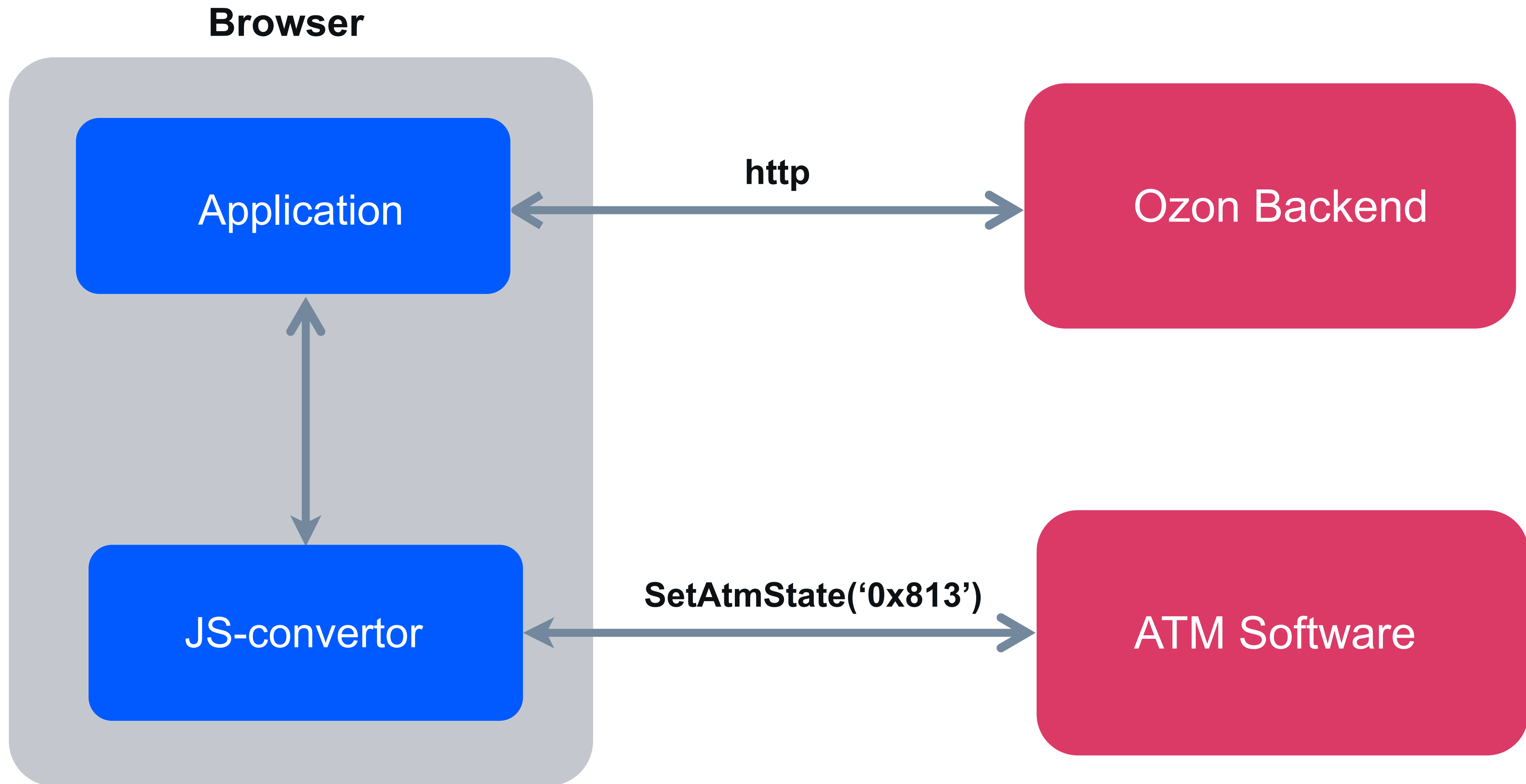
Как работает UI



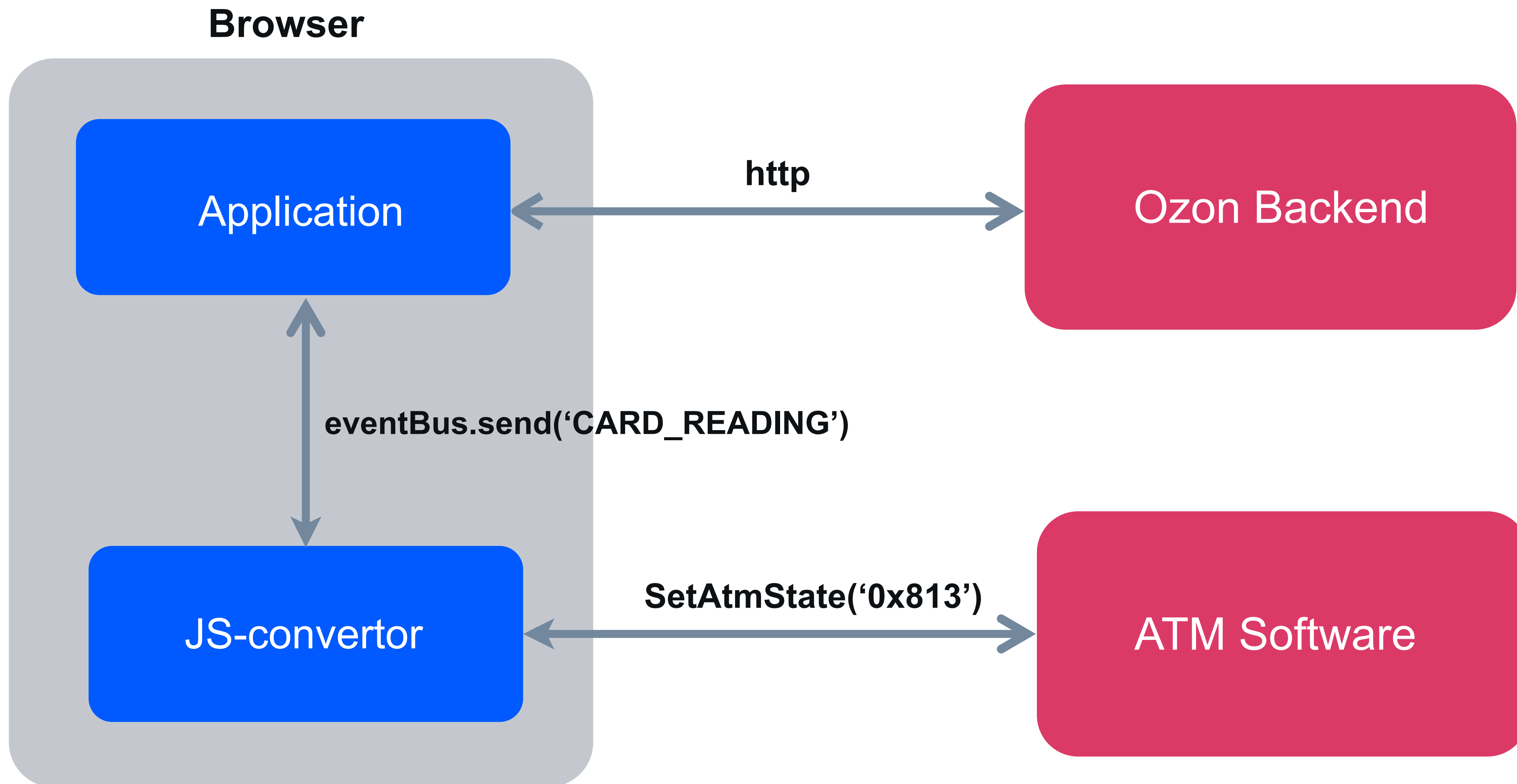
Как работает UI



Как работает UI



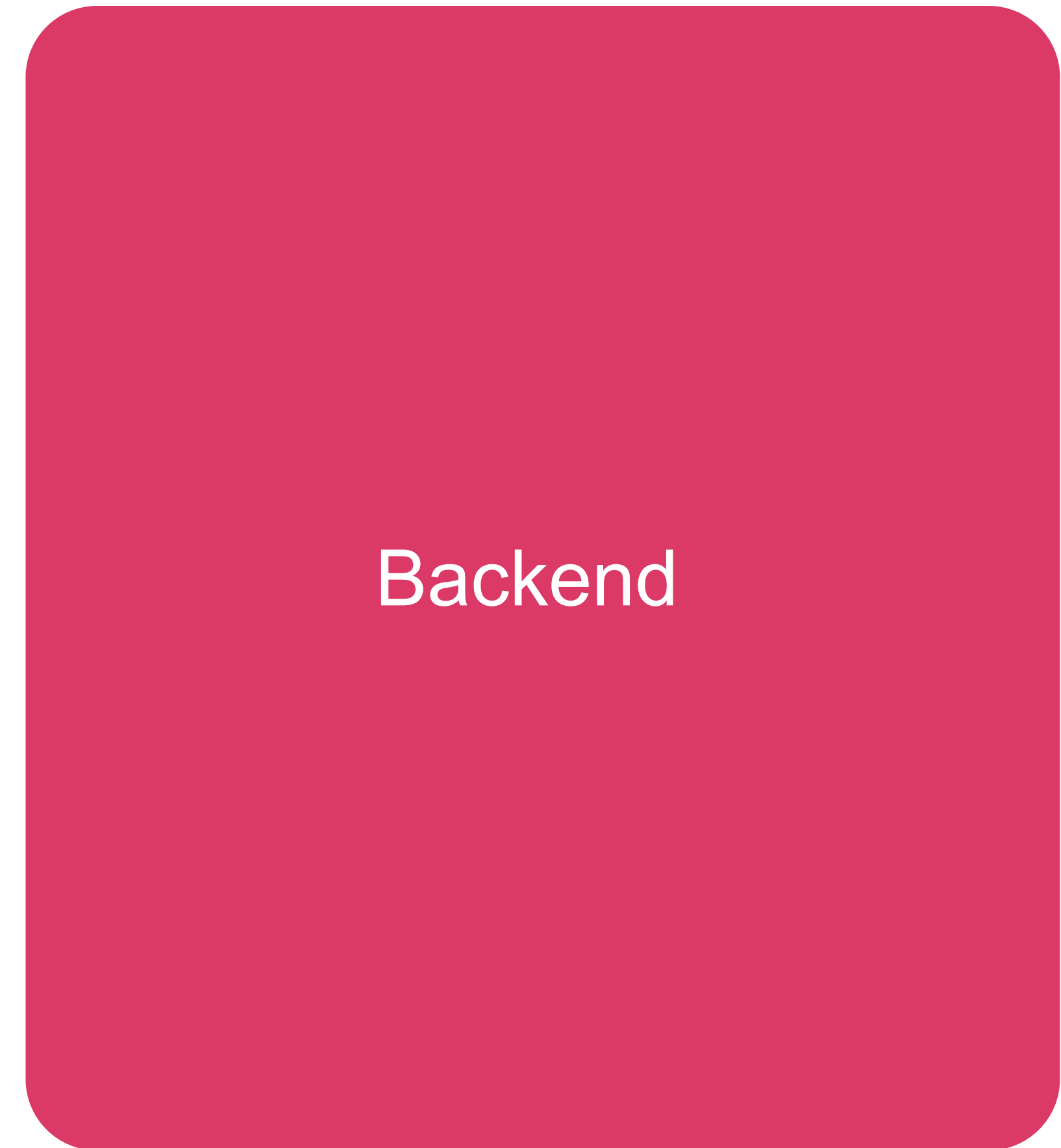
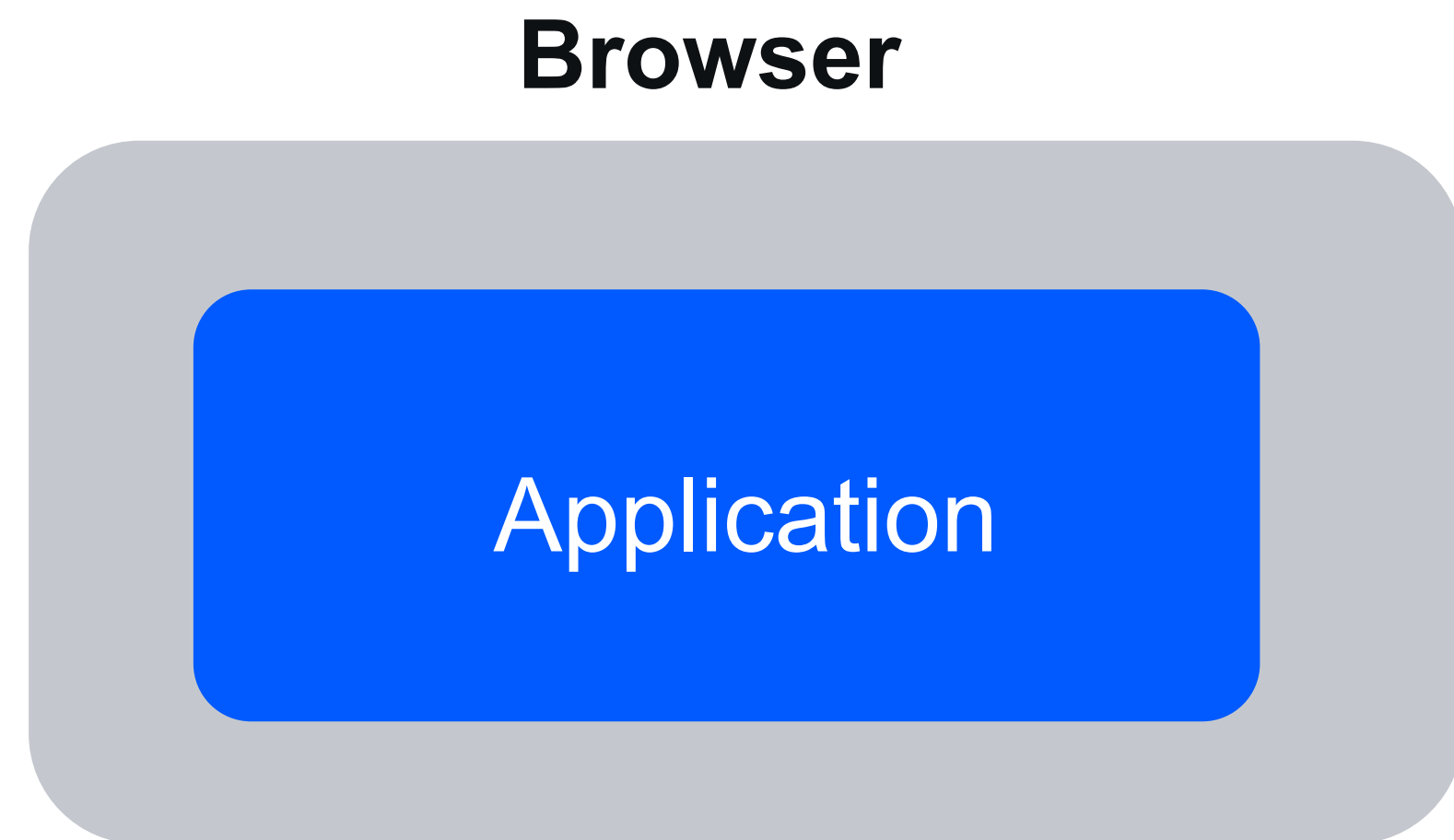
Как работает UI



Где-то мы это видели



Почти как веб-приложение



Тестировать просто, но...

01

**Работа
железа
влияет на UI**

Тестировать просто, но...

01

**Работа
железа
влияет на UI**

02

**Нужны
реальные
деньги**

Тестировать просто, но...

01

**Работа
железа
влияет на UI**

02

**Нужны
реальные
деньги**

03

**Не всегда
стабильное
окружение**

Тестировать просто, но...

01

**Работа
железа
влияет на UI**

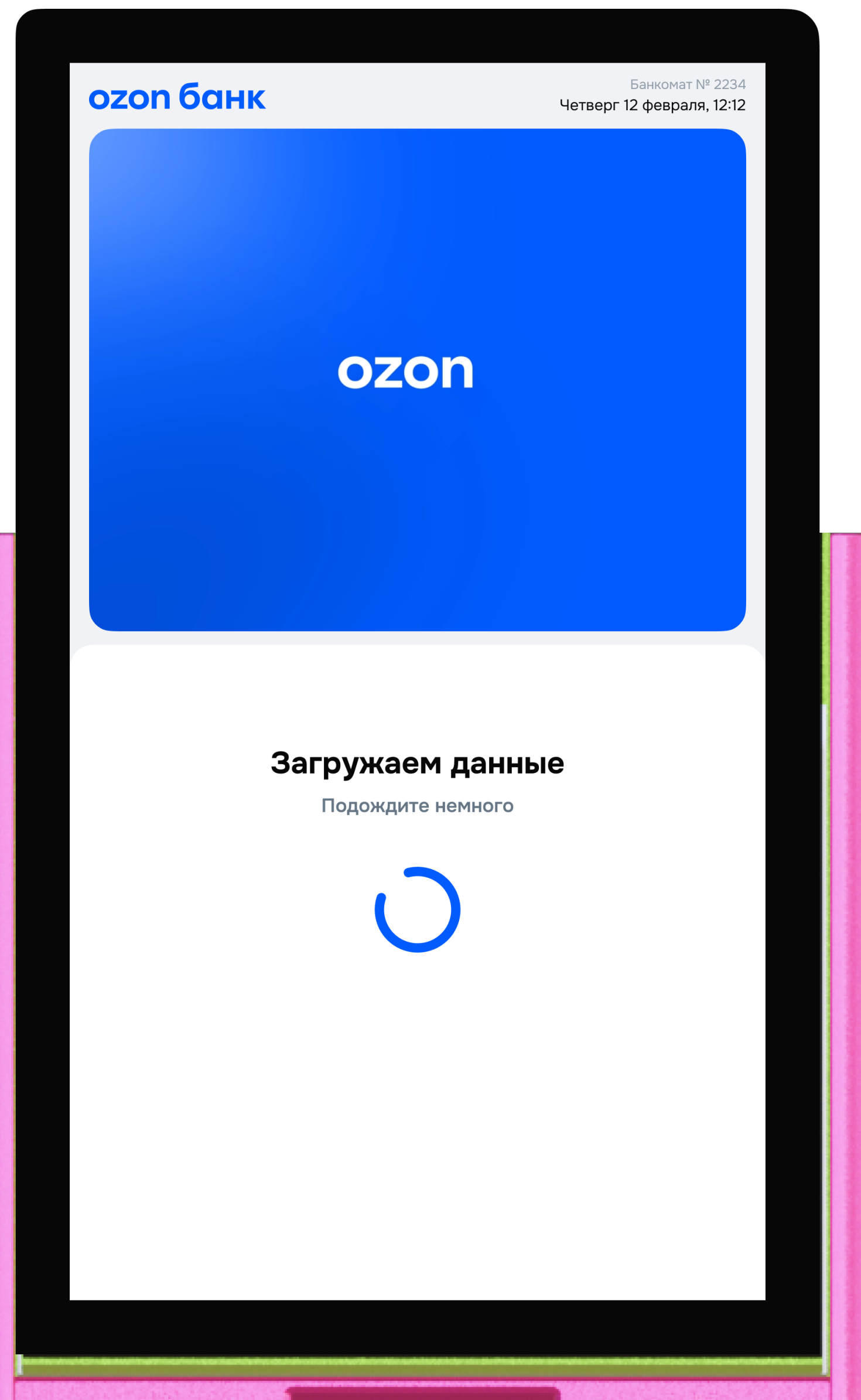
02

**Нужны
реальные
деньги**

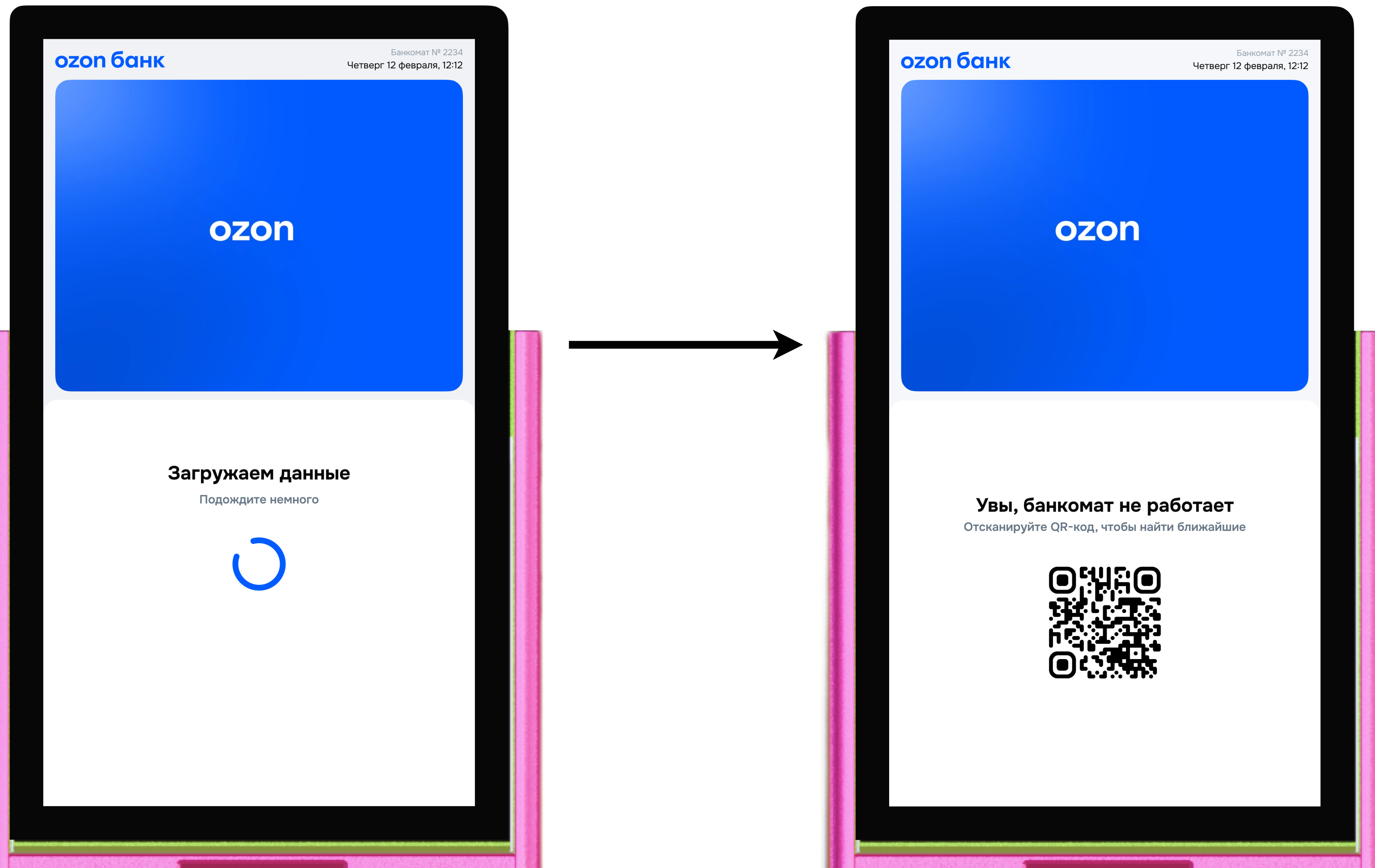
03

**Не всегда
стабильное
окружение**

Запуск приложения



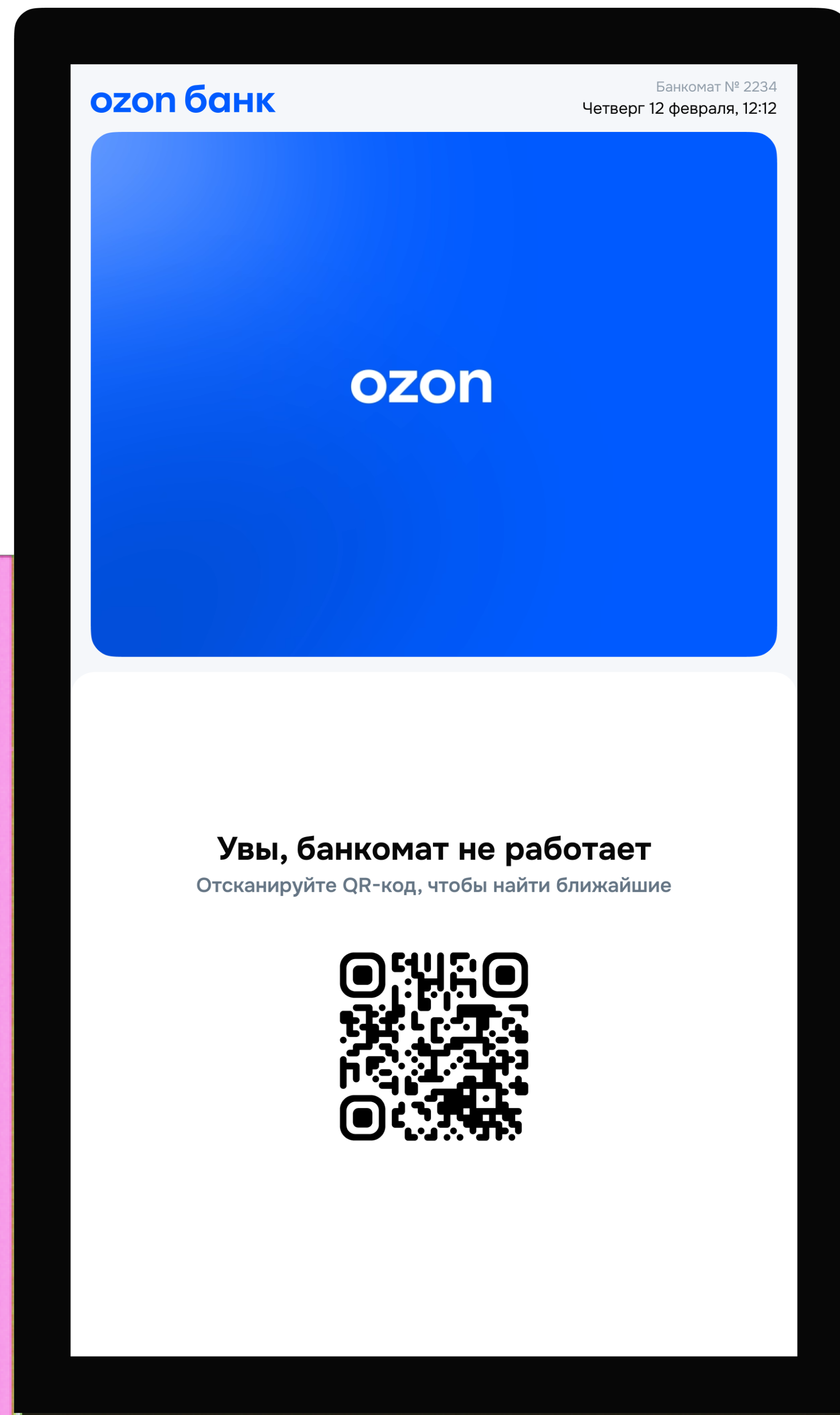
Запуск приложения



Когда случается



Ошибка датчика



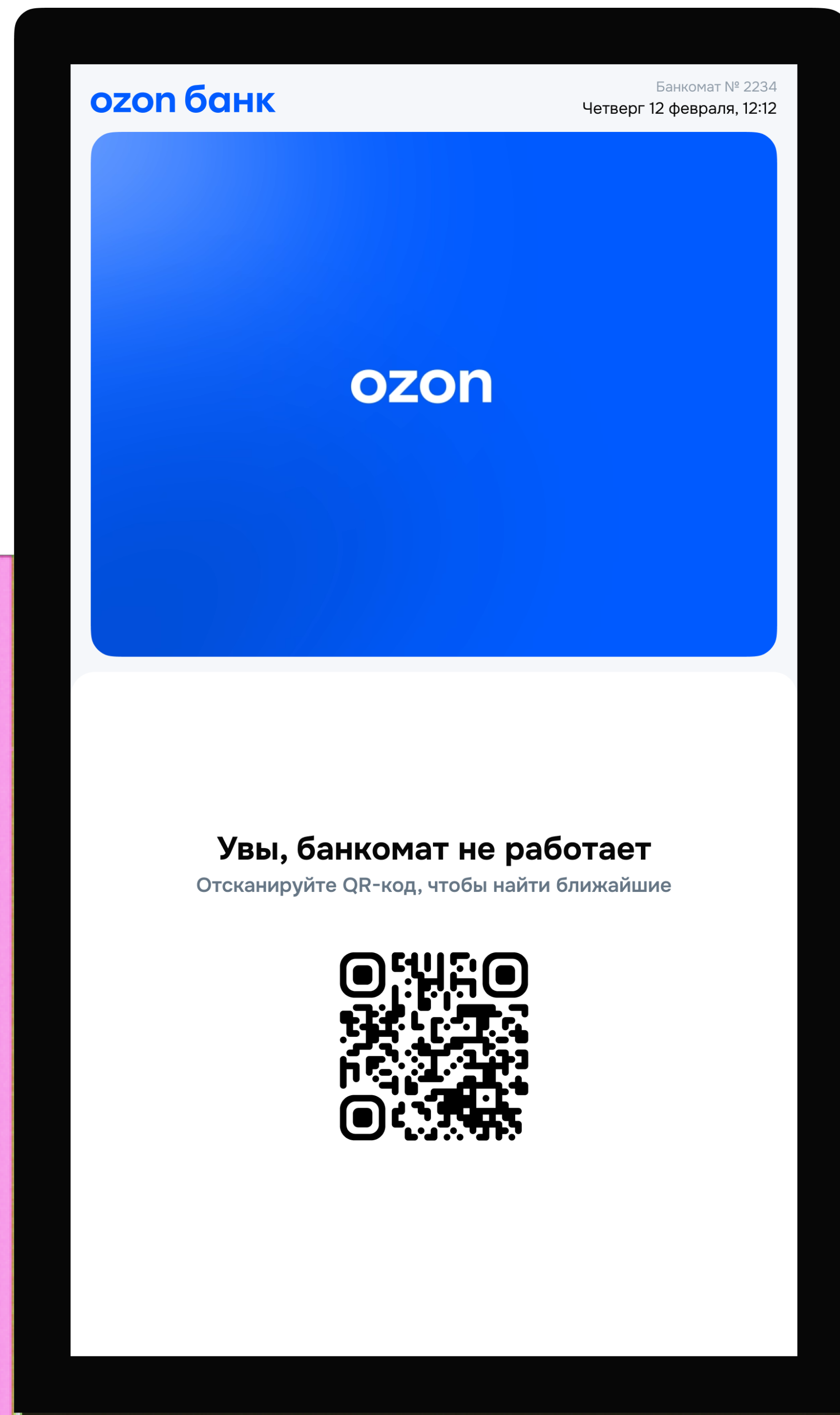
Когда случается



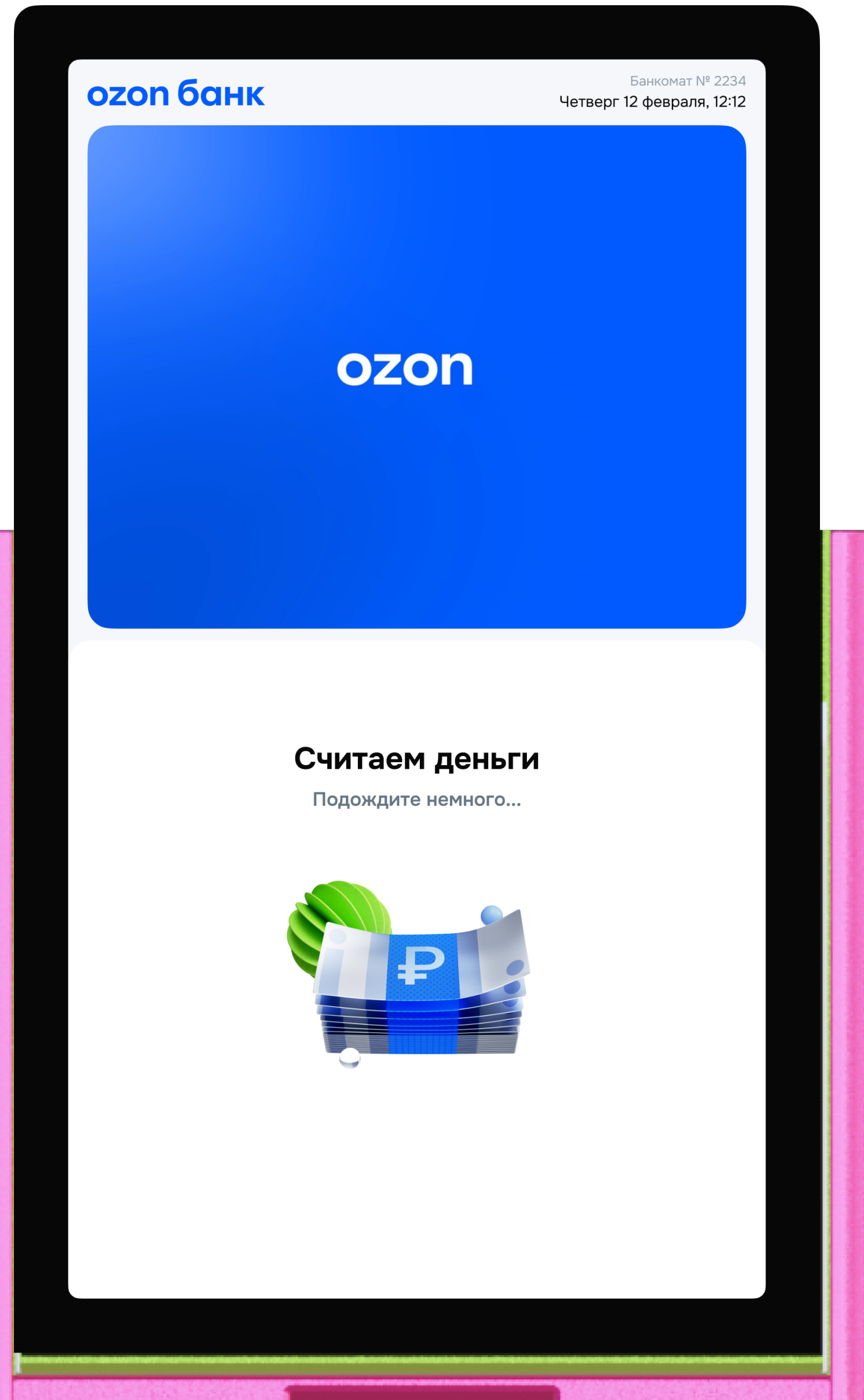
Ошибка датчика



При действии клиента

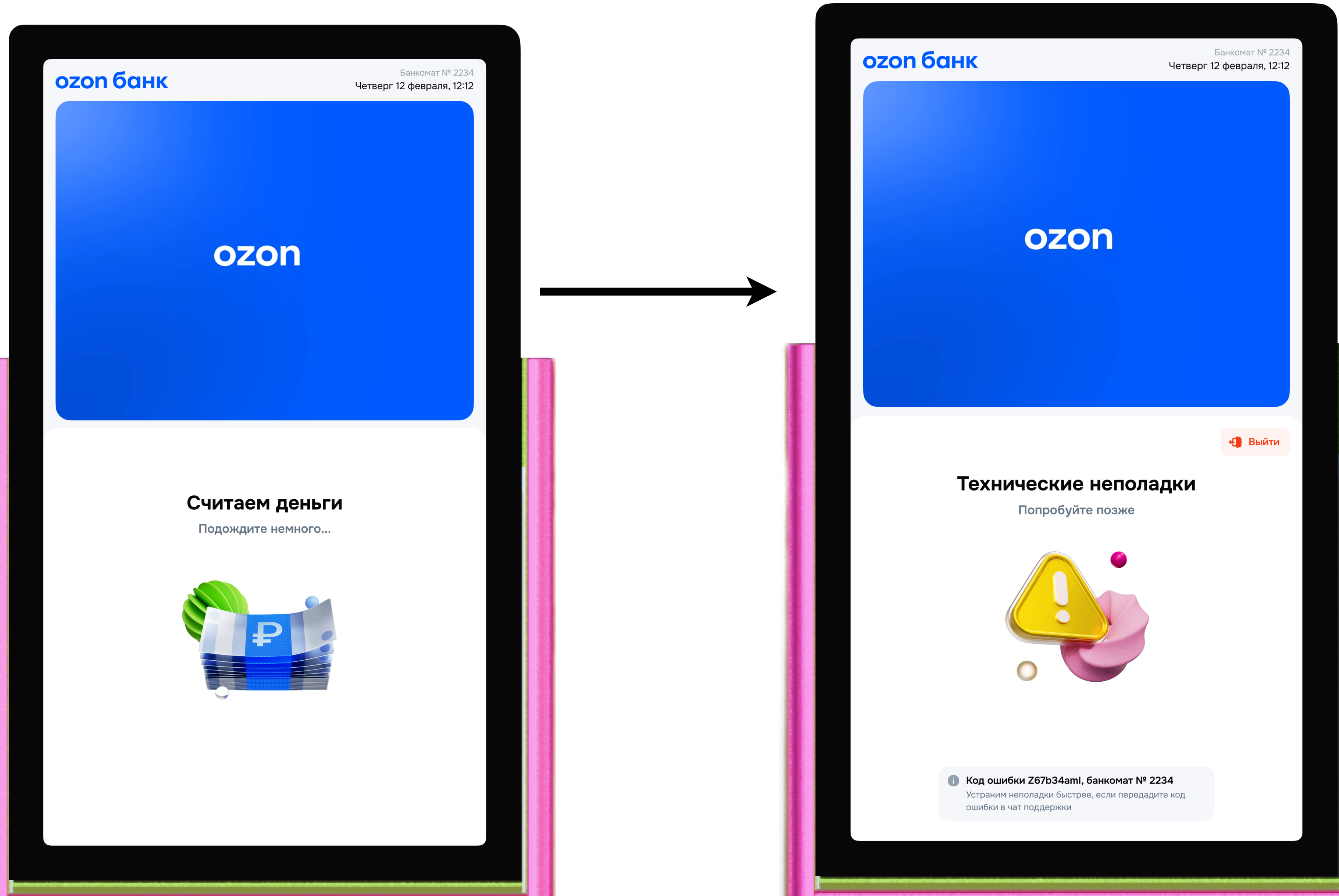


Чаще ломается при пересчете

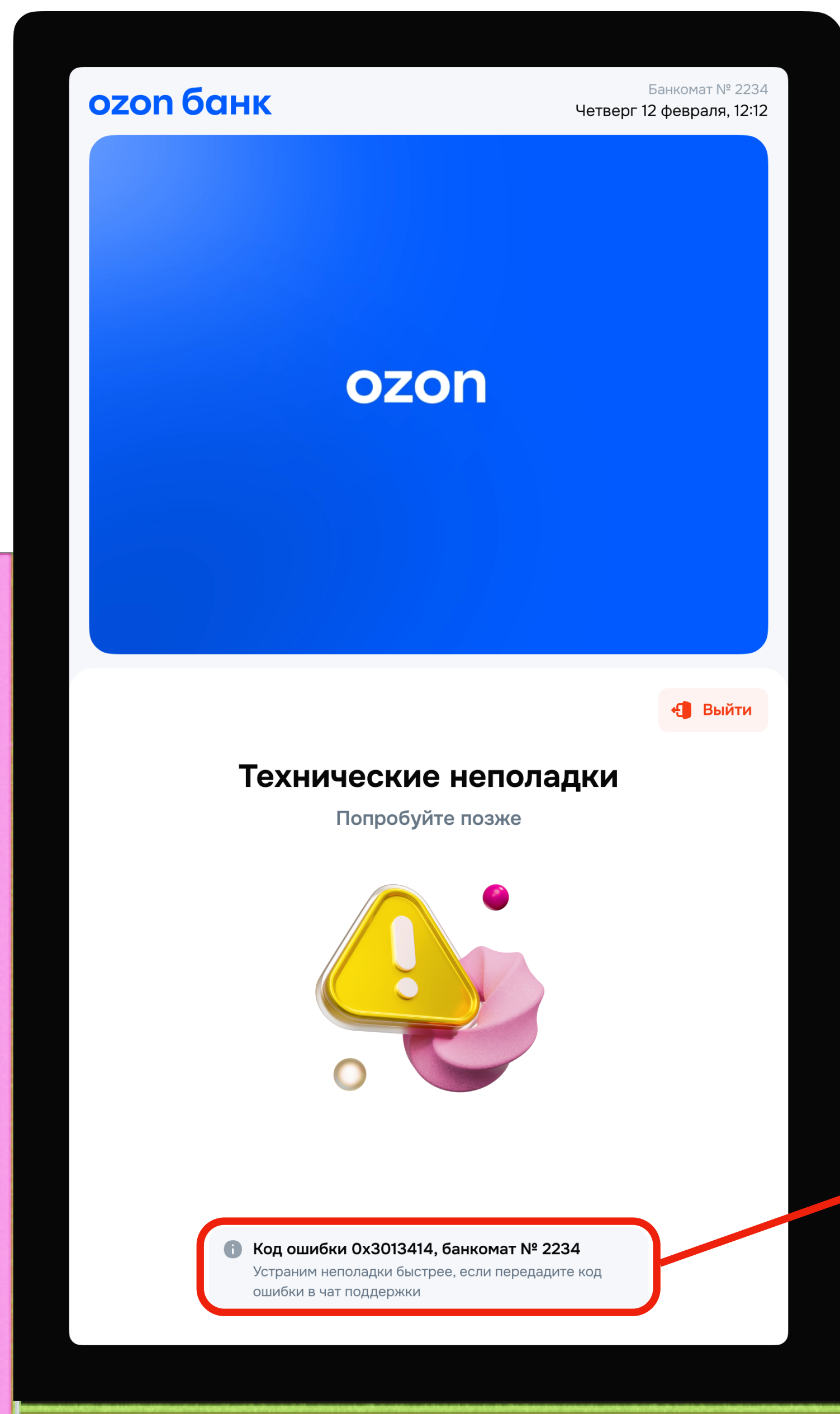


ozon банк

Завершить сессию с ошибкой



Важно описание ошибки

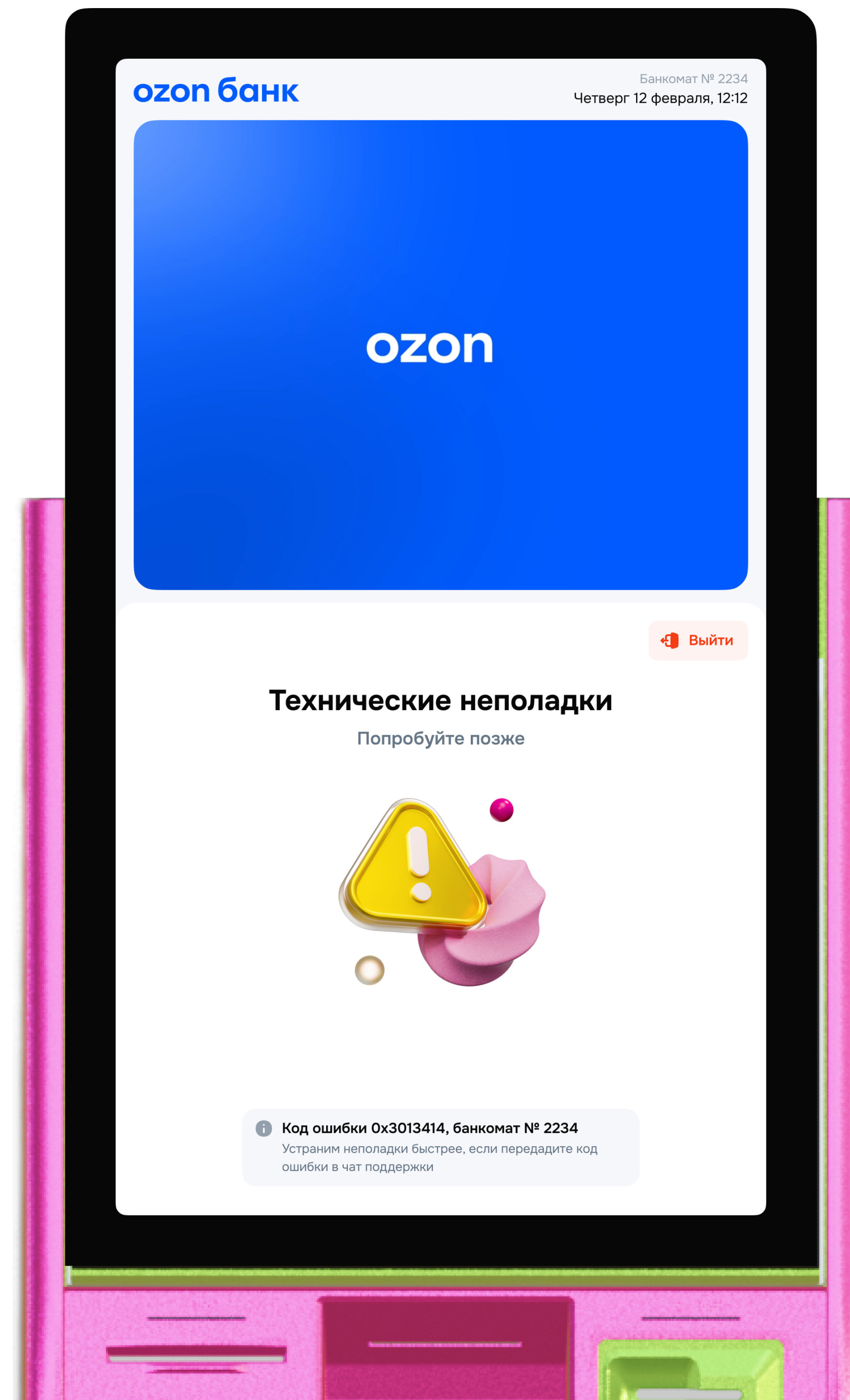


i **Код ошибки 0x3013414, банкомат № 2234**
Устраним неполадки быстрее, если передадите код ошибки в чат поддержки

Как воспроизвести



Отсоединить шлейф



Как воспроизвести




Отсоединить шлейф



Изменить программно



Как воспроизвести

 Отсоединить шлейф

 Изменить программно

 «Обмануть» датчик



Нельзя тестировать UI без
контекста работы железа



Тестировать просто, но...

01

Работа
железа
влияет на UI

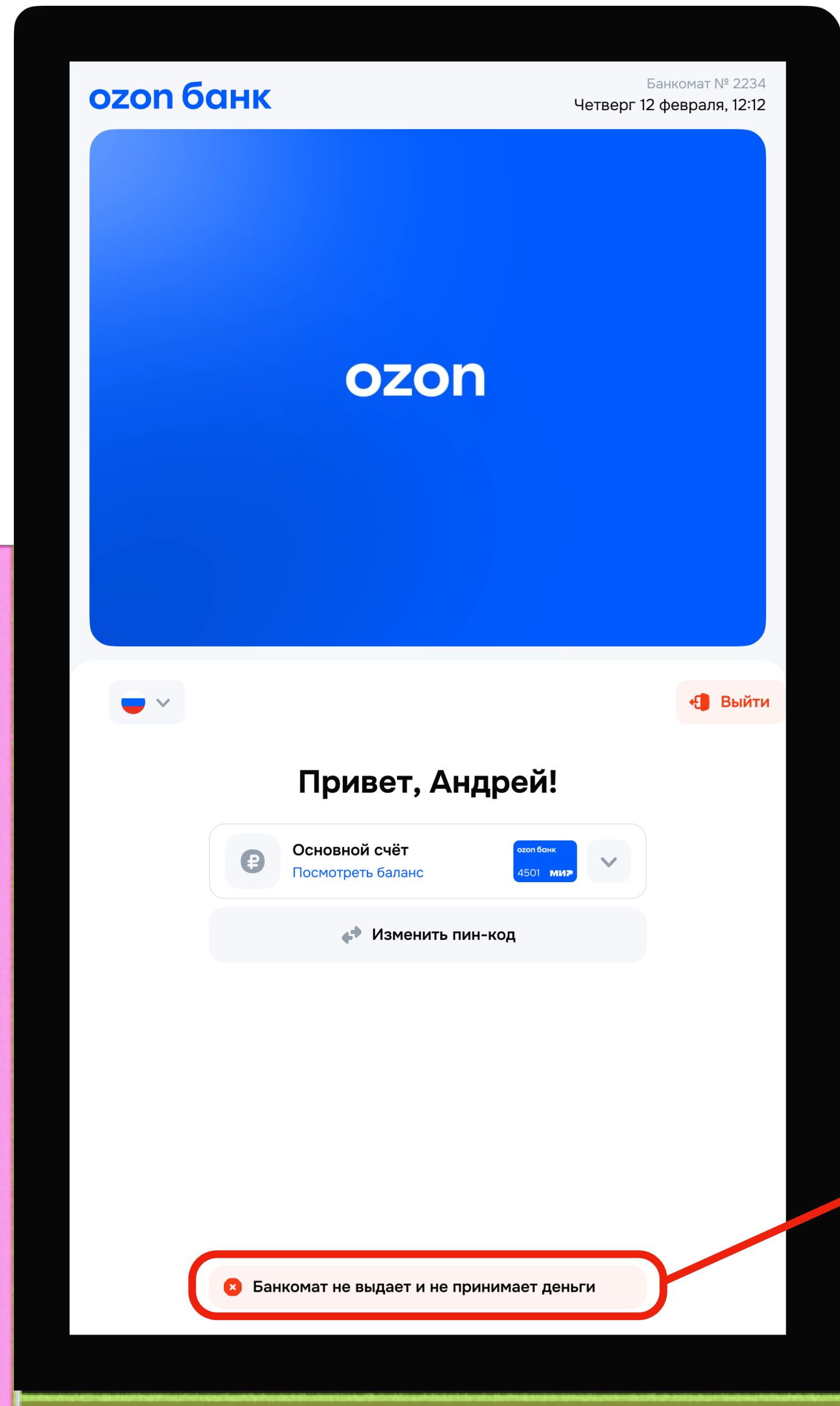
02

Нужны
реальные
деньги

03

Не всегда
стабильное
окружение

Если нет данных из кассет



Банкомат не выдает и не принимает деньги



Банкомат не выдает и не принимает деньги

Нужна инкассация



Проводить перед каждым тестом



Нужна инкассация



Проводить перед каждым тестом



Иногда нужно много денег



Нужна инкассация



Проводить перед каждым тестом



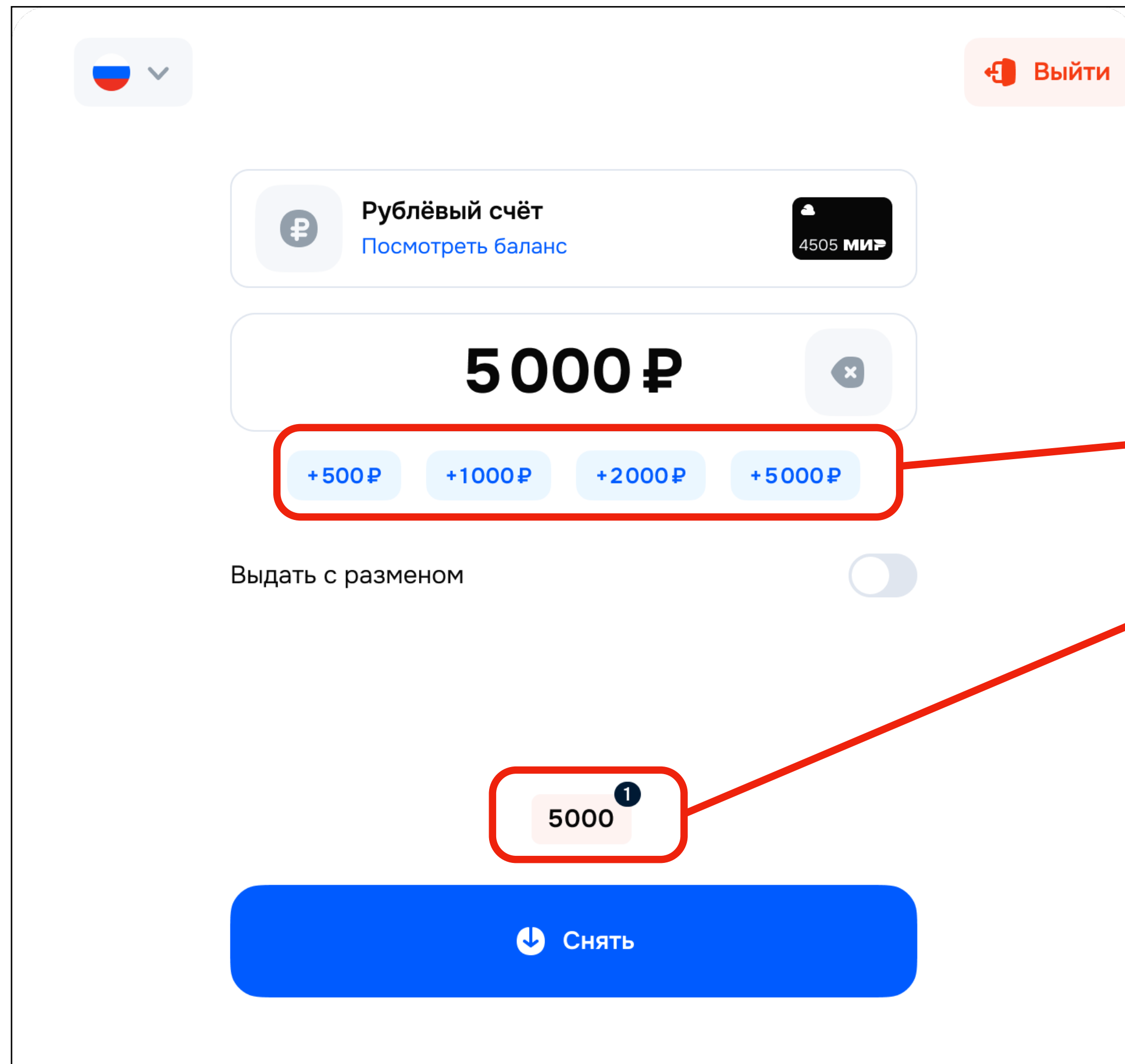
Иногда нужно много денег



Можно использовать моки

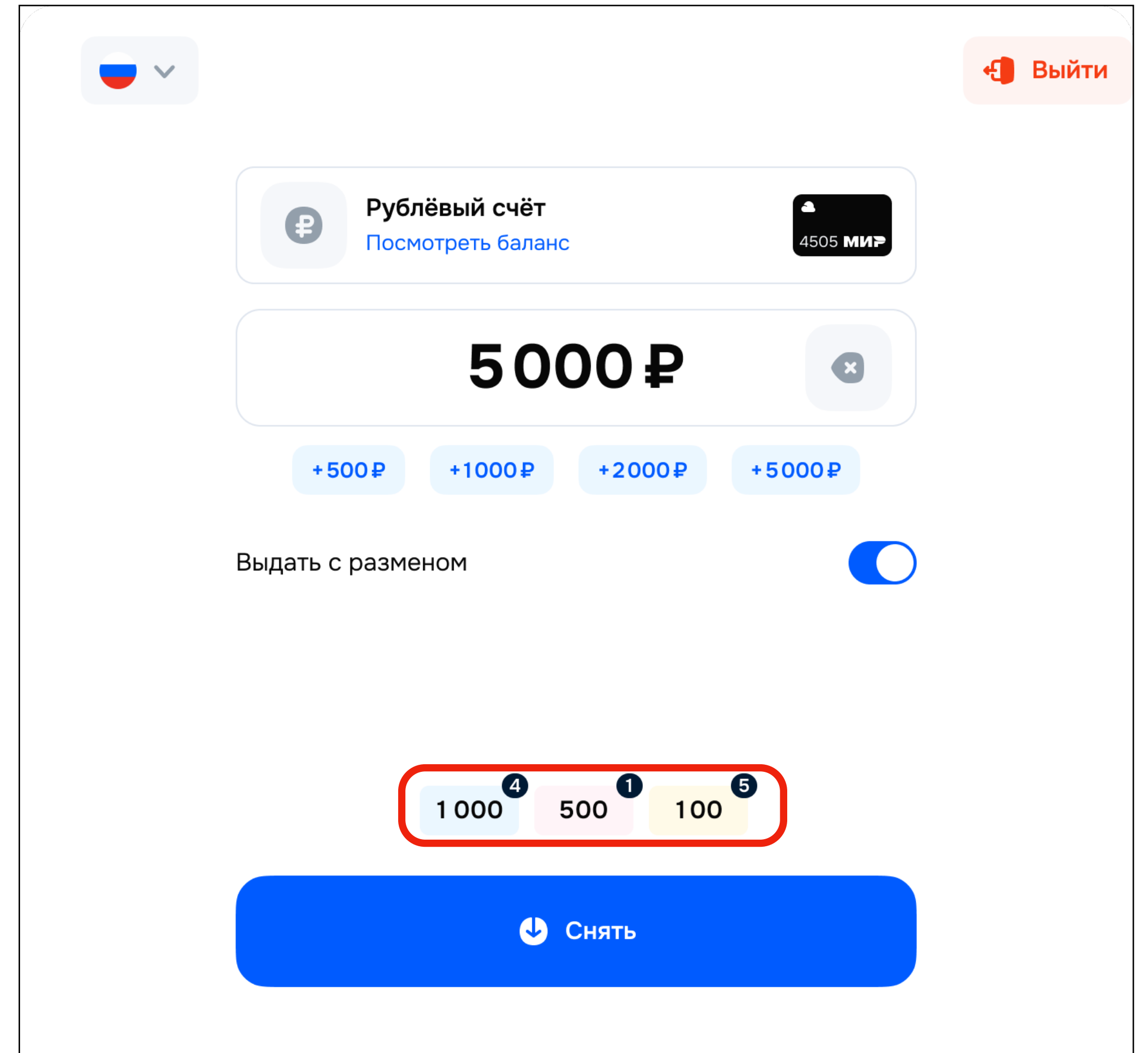
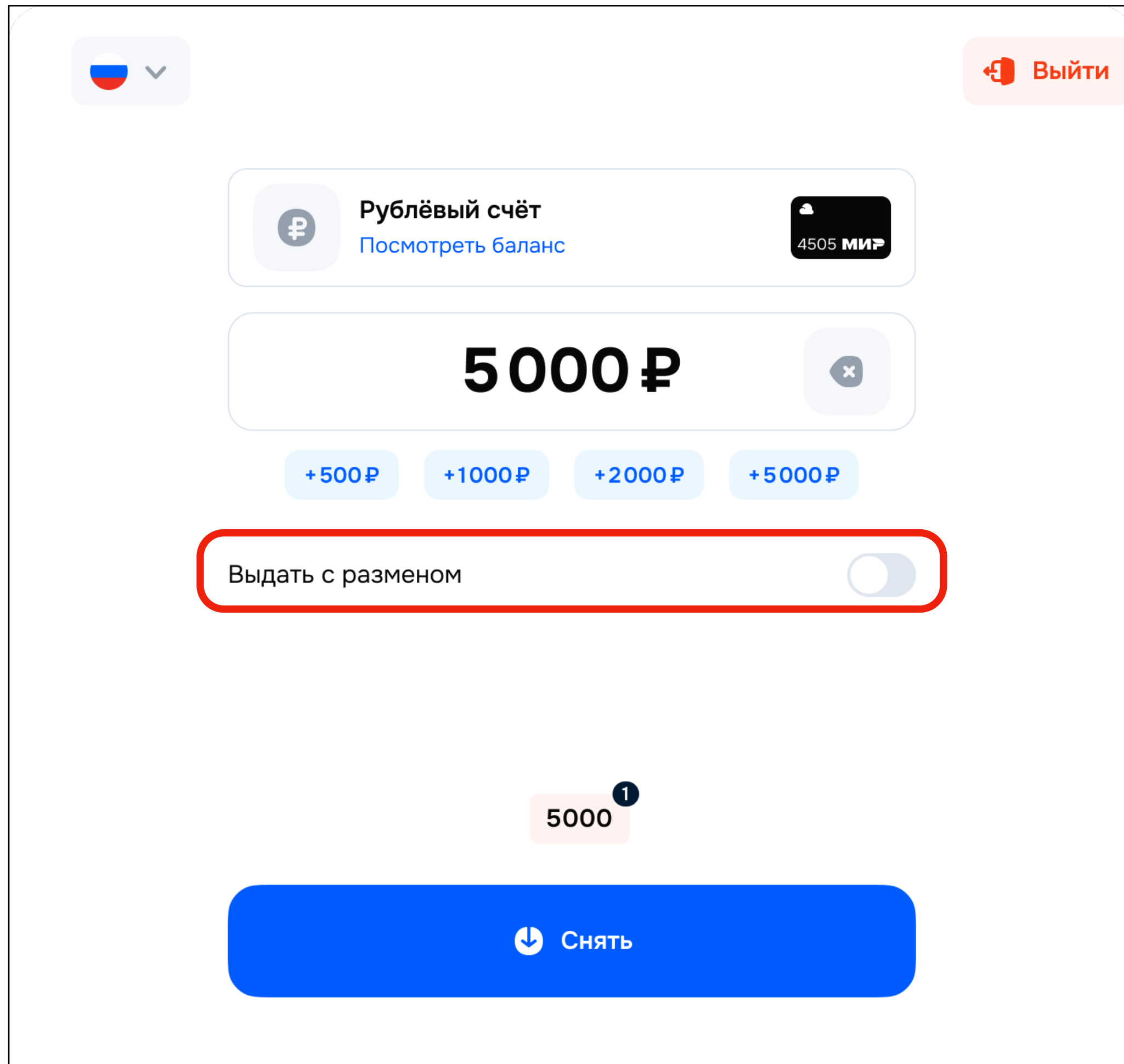


Быстрый набор и выдаваемые купюры



Получаем из кассет

Алгоритмы размена



Для тестирования нужны
реальные деньги, но есть выход



Тестировать просто, но...

01

**Работа
железа
влияет на UI**

02

**Нужны
реальные
деньги**

03

**Не всегда
стабильное
окружение**

Тестируем в идеальной среде



Пользователь ведет себя как ожидаем

Тестируем в идеальной среде



Пользователь ведет себя как ожидаем



Нет перебоев с электричеством

Тестируем в идеальной среде



Пользователь ведет себя как ожидаем

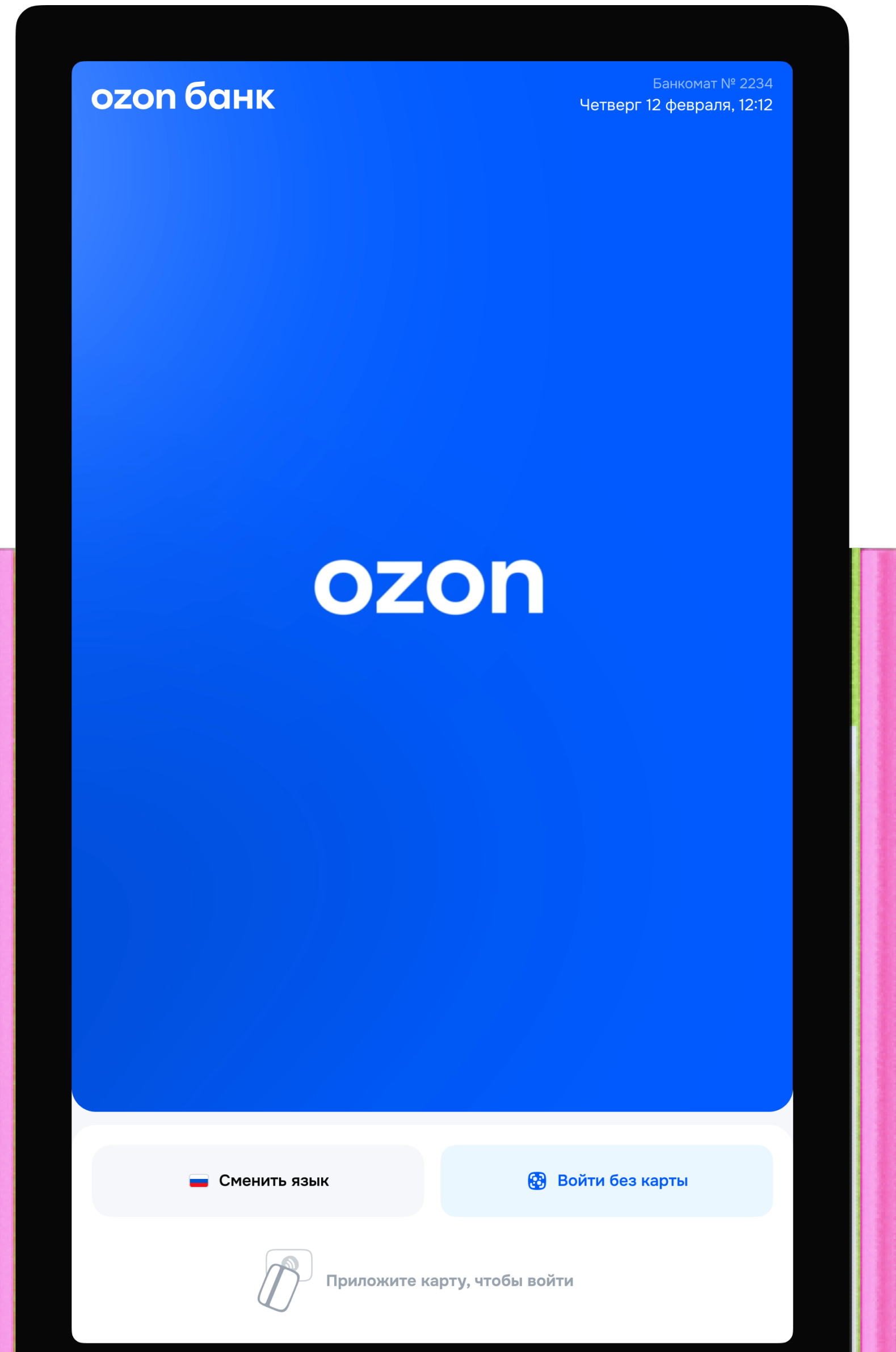


Нет перебоев с электричеством

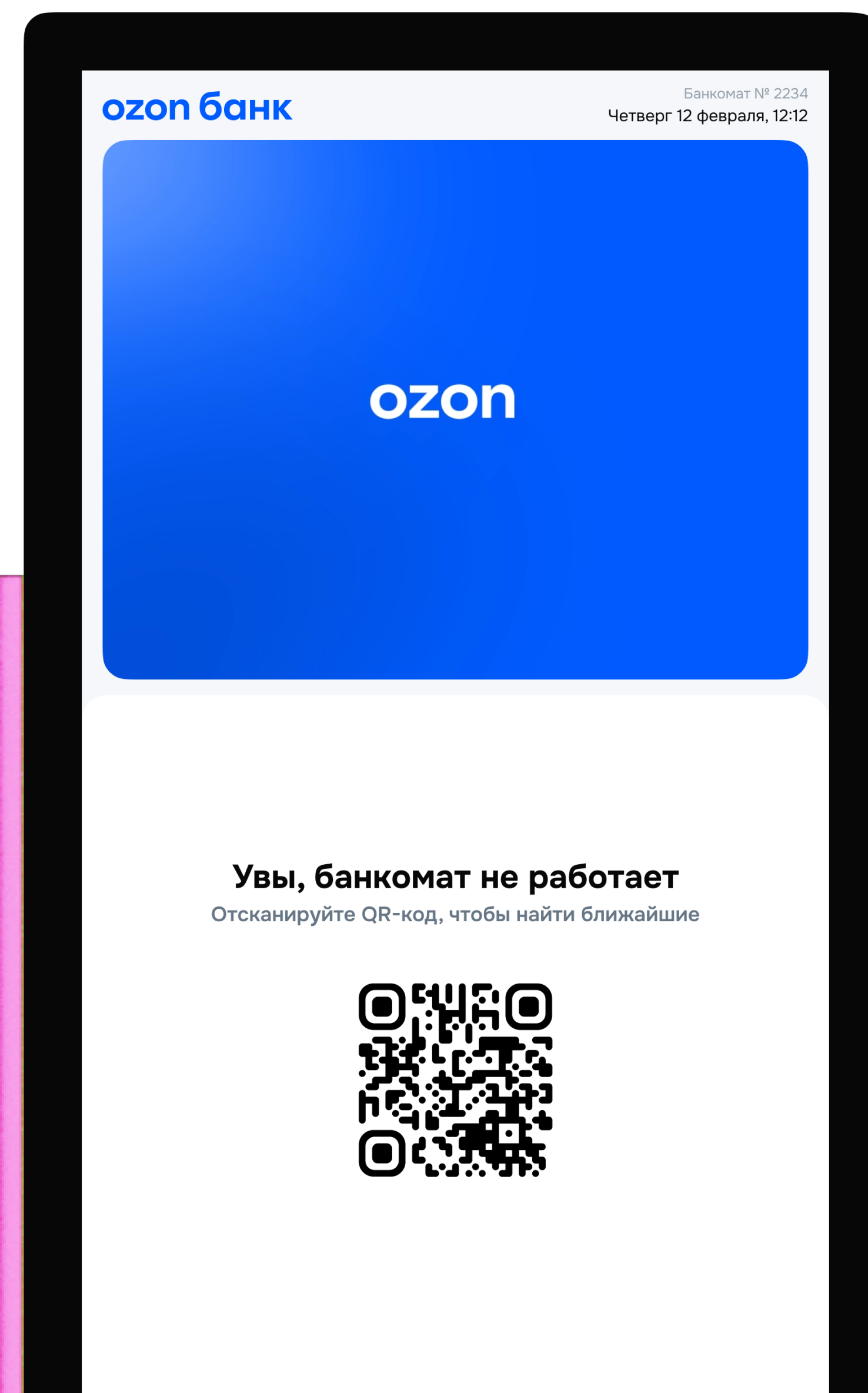
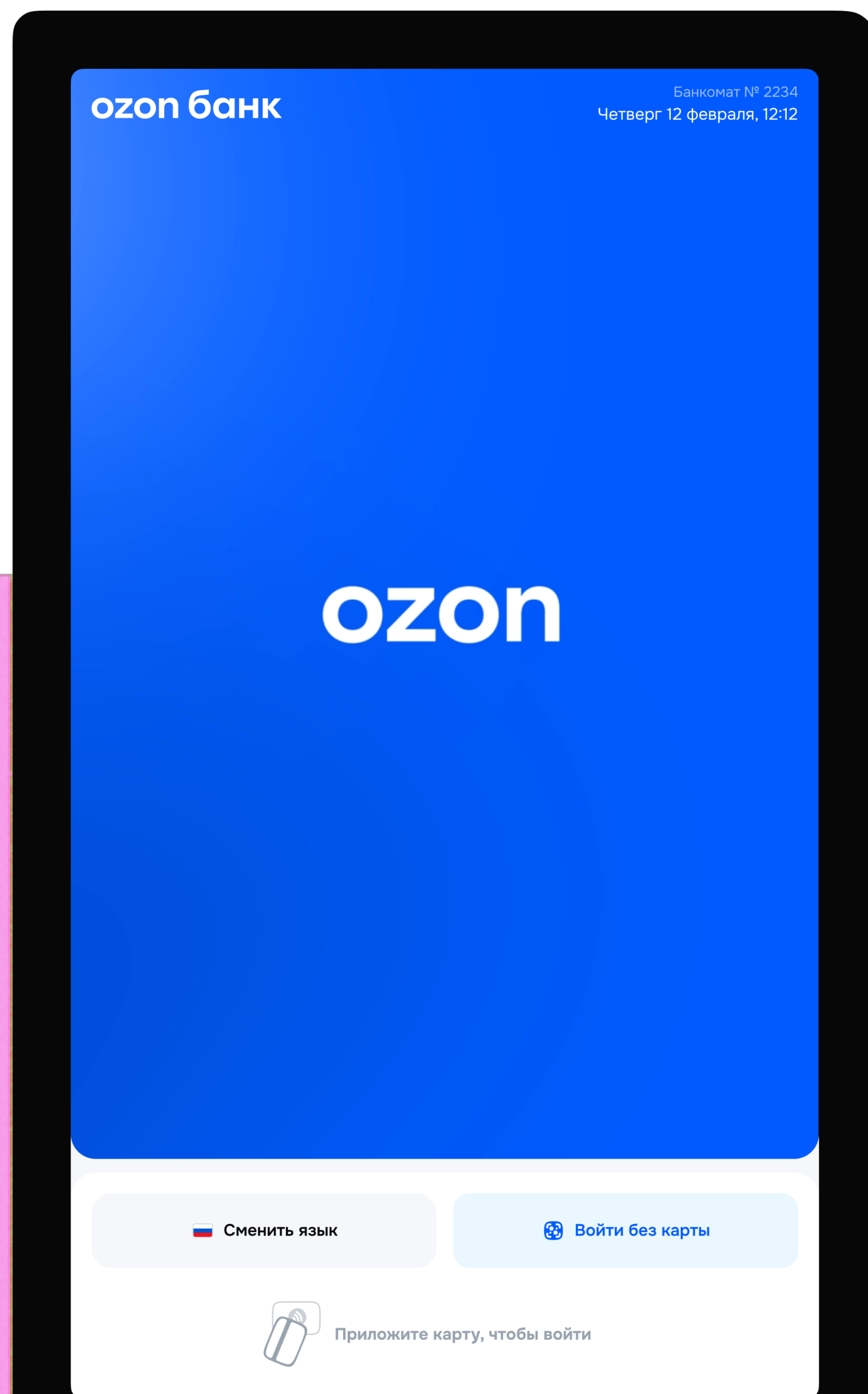


Всегда стабильный интернет

Перебои мобильной сети



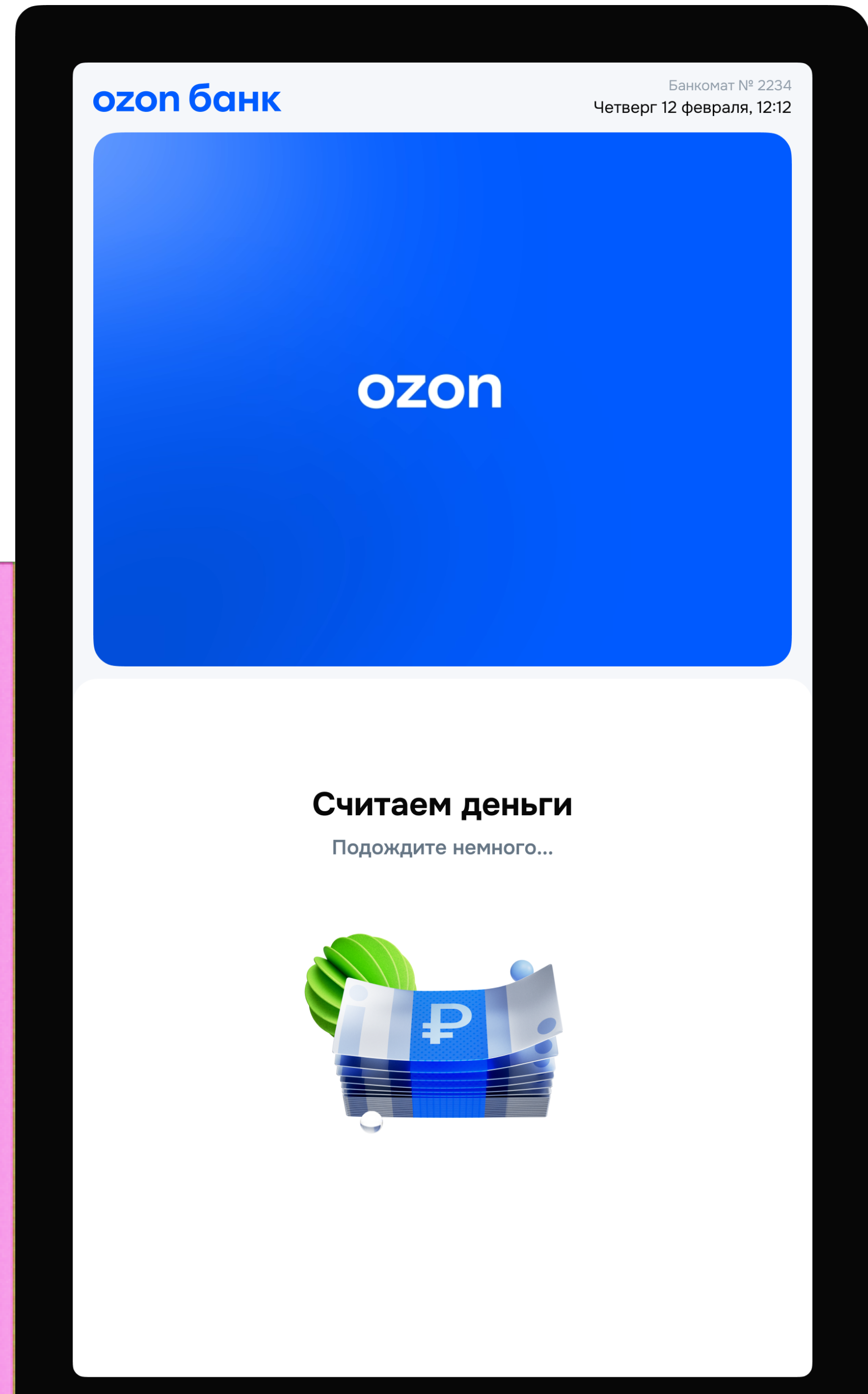
Перебои мобильной сети



Клиентская сессия



Потеря связи при операции



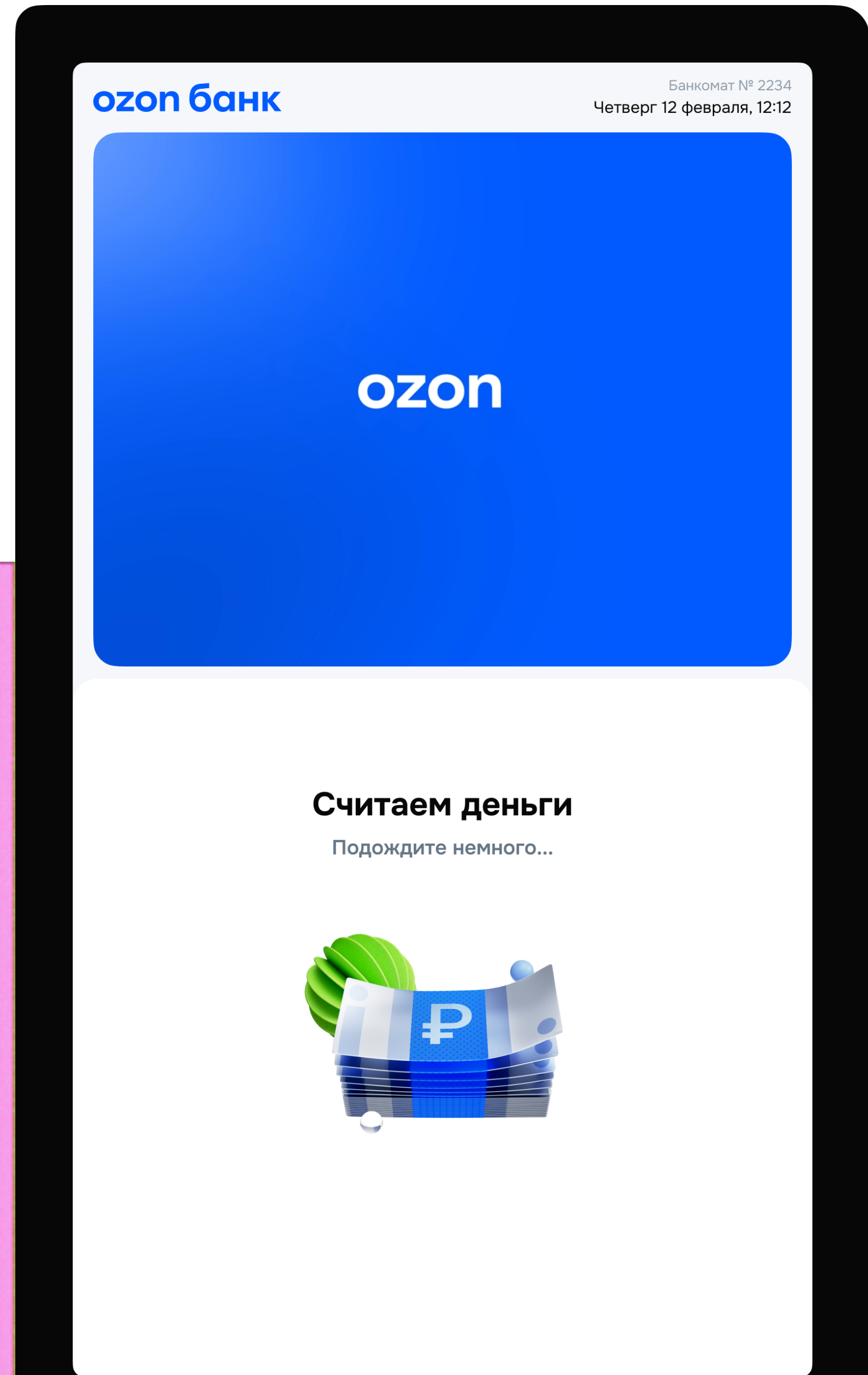
Клиентская сессия



Потеря связи при операции



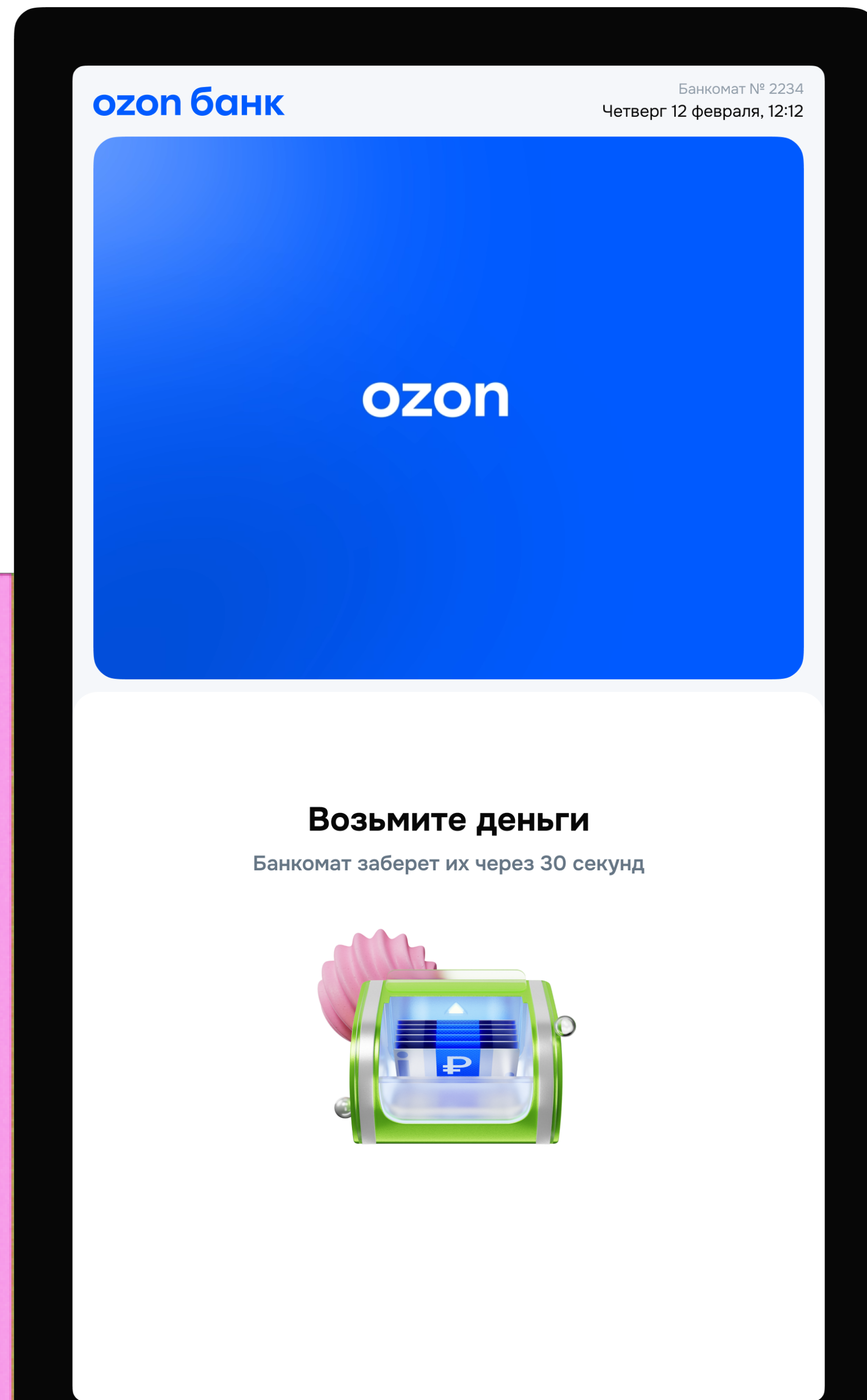
Завершить сессию и вернуть деньги



Как проверить



Отключить модем



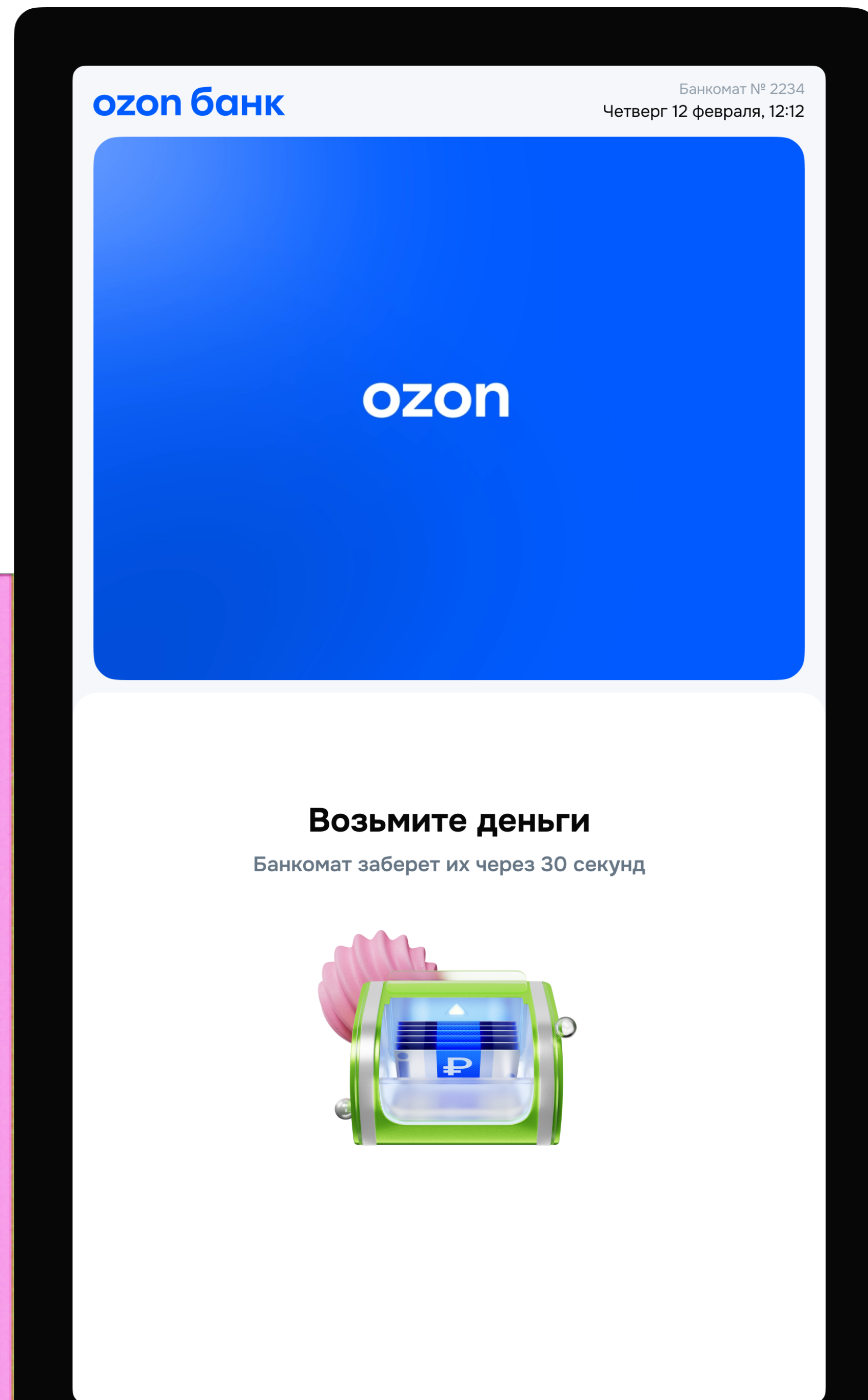
Как проверить



Отключить модем



Использовать прокси или фильтр



Чем ограничить сеть



Мониторить и фильтровать пакеты

Чем ограничить сеть



Мониторить и фильтровать пакеты



<https://github.com/GyulyVGC/sniffnet>

Чем ограничить сеть



Мониторить и фильтровать пакеты



Замедлять и ограничивать соединения

Чем ограничить сеть

- ➔ Мониторить и фильтровать пакеты
- ➔ Замедлять и ограничивать соединения



Toxiproxy

<https://github.com/Shopify/toxiproxy>

Чем ограничить сеть

- Мониторить и фильтровать пакеты
- Замедлять и ограничивать соединения
- Написать что-то свое



Перебои сети могут стать
точкой отказа



Не сможем проверить в регрессе

Не сможем проверить в регрессе



Сигнализацию и безопасность

Не сможем проверить в регрессе



Сигнализацию и безопасность



Источник: Промавто

Не сможем проверить в регрессе

- ✘ Сигнализацию и безопасность
- ✘ Платежные системы и NFC-токены



Не сможем проверить в регрессе

- ✘ Сигнализацию и безопасность
- ✘ Платежные системы и NFC-токены
- ✘ Вендорские решения



Источник: АртМосква



Что будем автоматизировать?

Можем автоматизировать



Пользовательскую логику

Можем автоматизировать



Пользовательскую логику



Доступность железа

Можем автоматизировать

 Пользовательскую логику

 Доступность железа

 Верстку

Можем автоматизировать

→ Пользовательскую логику

→ Доступность железа

→ Верстку

→ Моки

Можем автоматизировать



Пользовательскую логику



Доступность железа



Верстку



Моки



Интеграцию с вендором



E2E

Если не Playwright то?

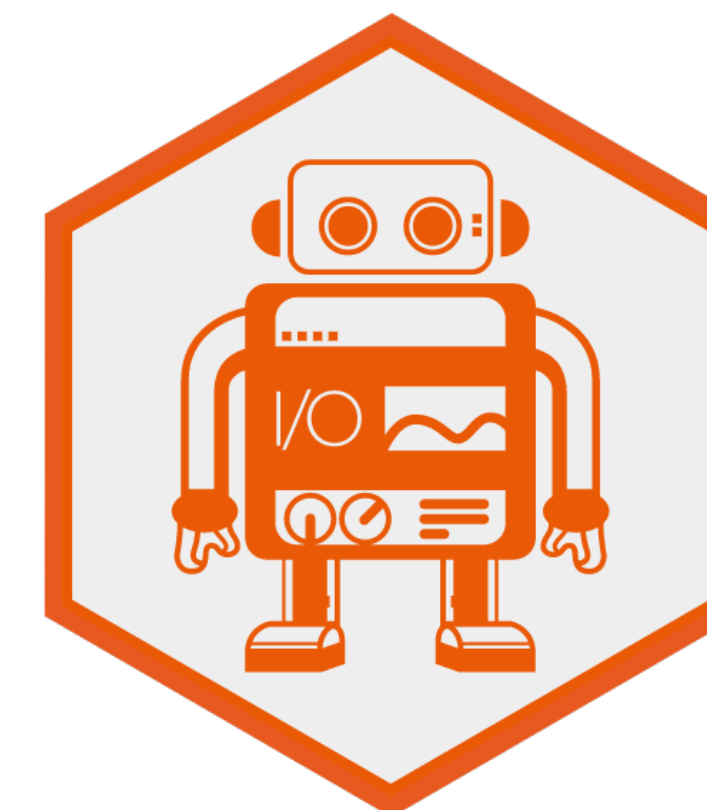
Если не Playwright то?

- Auto-wait из коробки
- Tracing и UI mode
- Работает в банкоматной среде



Если не Playwright то?

- Auto-wait из коробки
- Tracing и UI mode
- Работает в банкоматной среде

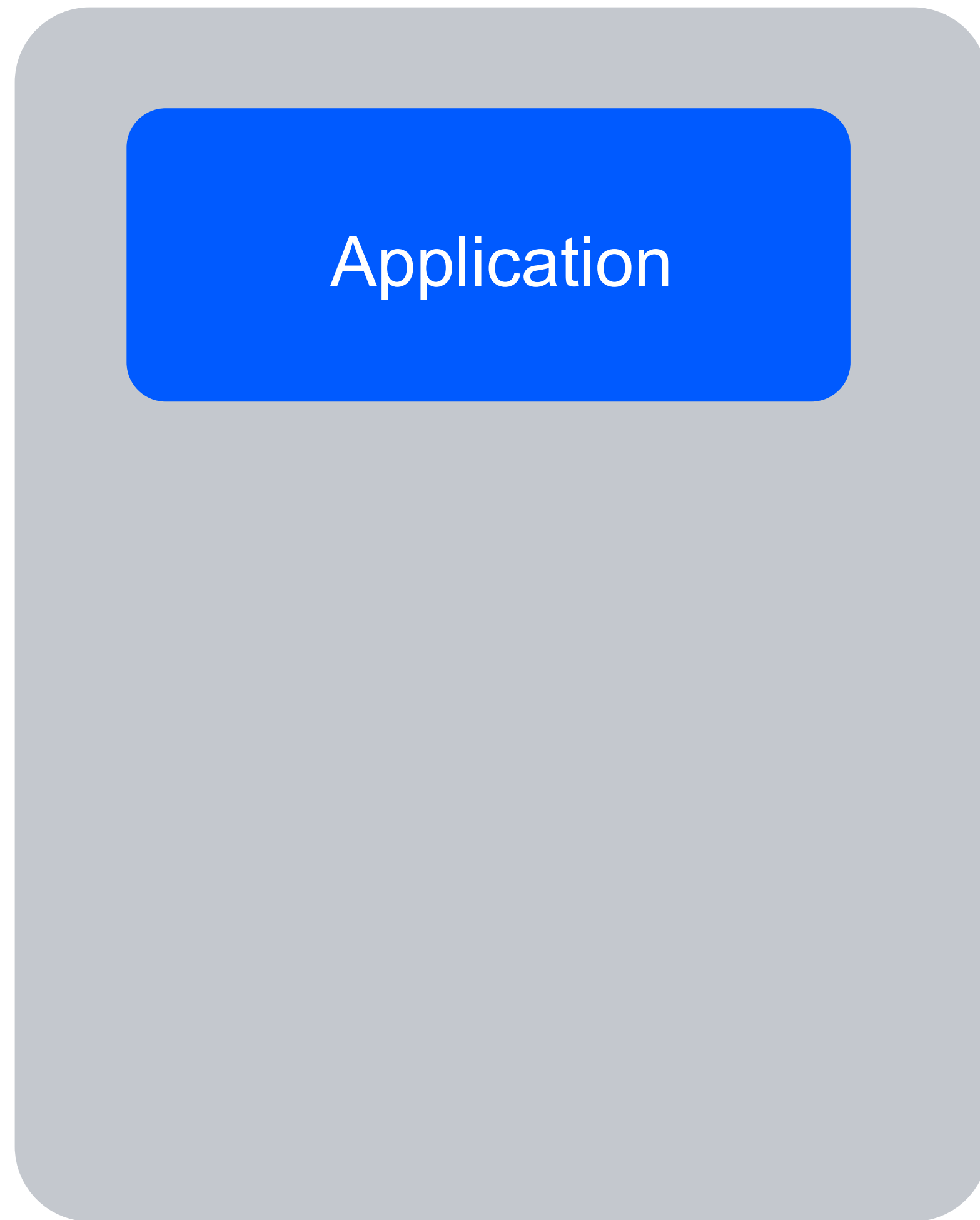




Моки бизнес логики и событий

Что будем мокировать

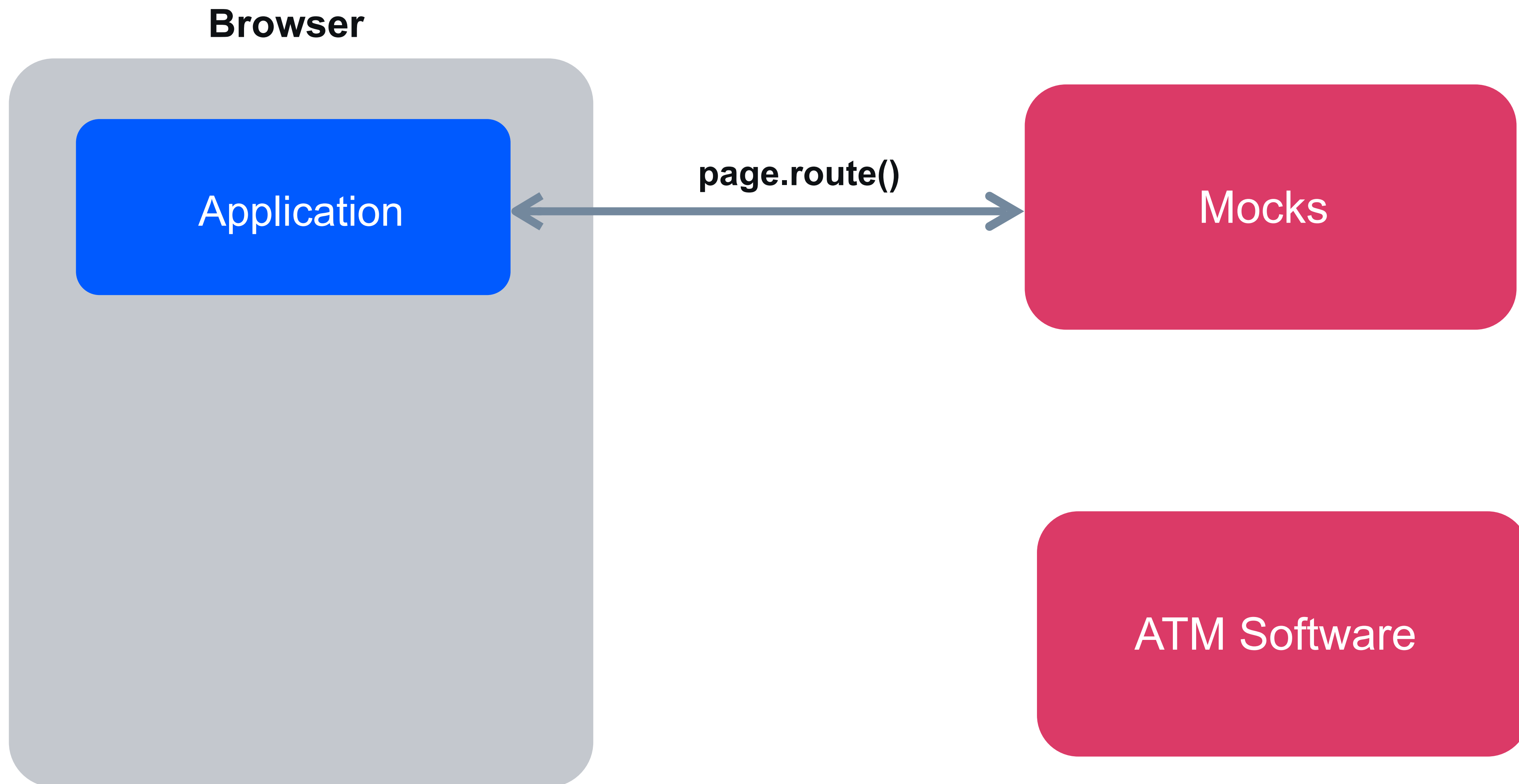
Browser



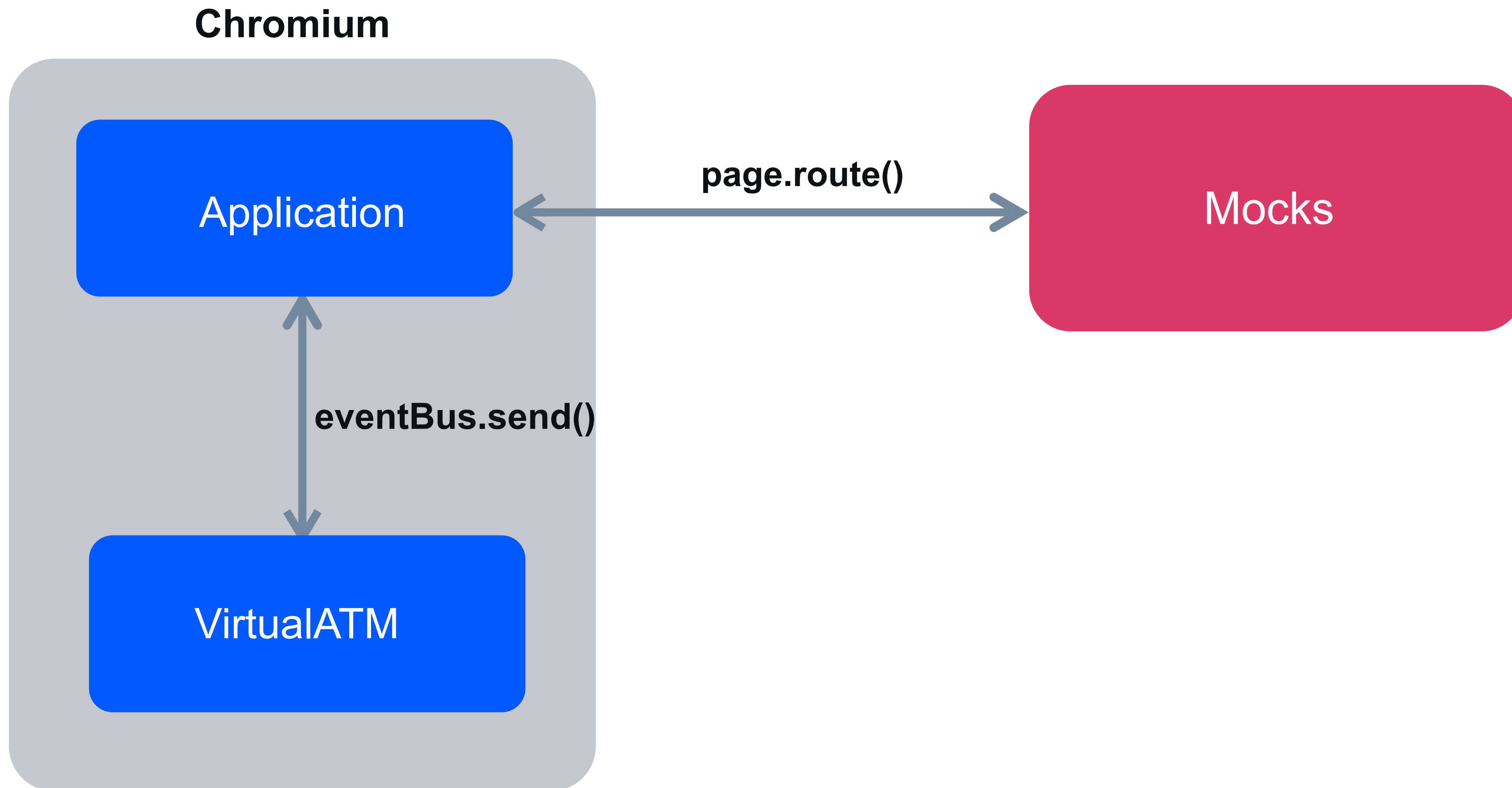
Ozon Backend

ATM Software

Что будем мокировать



Что будем мокировать



Виртуальный АТМ

```
export const virtualATM = () => {
  return {
    hardwareStatus: {
      cardReader: true,
      nfcReader: true,
      acceptor: true,
      dispenser: true,
      printer: true,
      videoCamera: true,
      cardDispenser: true,
    },
    cassettes: [
      {
        denomination: '100',
        count: 10,
      },
    ],
    features: {
      cardIssue: true,
    },
  };
};
```

Виртуальный АТМ

```
export const virtualATM = () => {
  return {
    hardwareStatus: {
      cardReader: true,
      nfcReader: true,
      acceptor: true,
      dispenser: true,
      printer: true,
      videoCamera: true,
      cardDispenser: true,
    },
    cassettes: [
      {
        denomination: '100',
        count: 10,
      },
    ],
    features: {
      cardIssue: true,
    },
  };
};
```

← **Статус железа**

Виртуальный АТМ

```
export const virtualATM = () => {
  return {
    hardwareStatus: {
      cardReader: true,
      nfcReader: true,
      acceptor: true,
      dispenser: true,
      printer: true,
      videoCamera: true,
      cardDispenser: true,
    },
    cassettes: [
      {
        denomination: '100',
        count: 10,
      },
    ],
    features: {
      cardIssue: true,
    },
  };
};
```

← **Деньги в кассетах**

Виртуальный АТМ

```
export const virtualATM = () => {
  return {
    hardwareStatus: {
      cardReader: true,
      nfcReader: true,
      acceptor: true,
      dispenser: true,
      printer: true,
      videoCamera: true,
      cardDispenser: true,
    },
    cassettes: [
      {
        denomination: '100',
        count: 10,
      },
    ],
    features: {
      cardIssue: true,
    },
  };
};
```

← **Полезные метаданные**

Добавляем в код приложения

```
layout.ts

const AppLayout = () => {
  return (
    <Application>
      <Header />

      {/* Логика приложения */}
      <Main />

      {/* Шина событий */}
      window.eventBus.init()

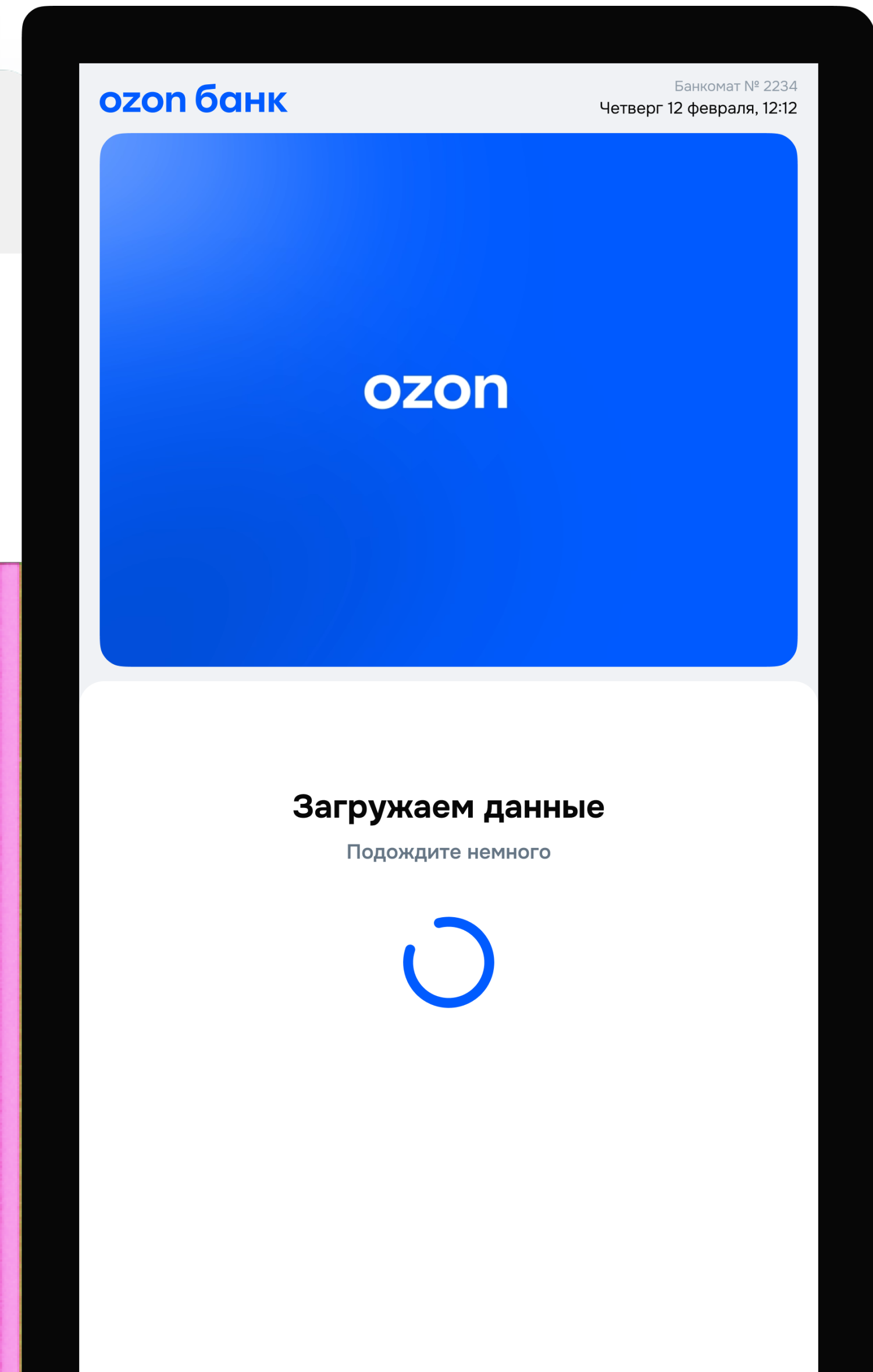
      {/* Виртуальный ATM */}
      window.virtualAtm = {}

      <Footer />
    </Application>
  );
};
```

**Делаем доступным
для компонентов**

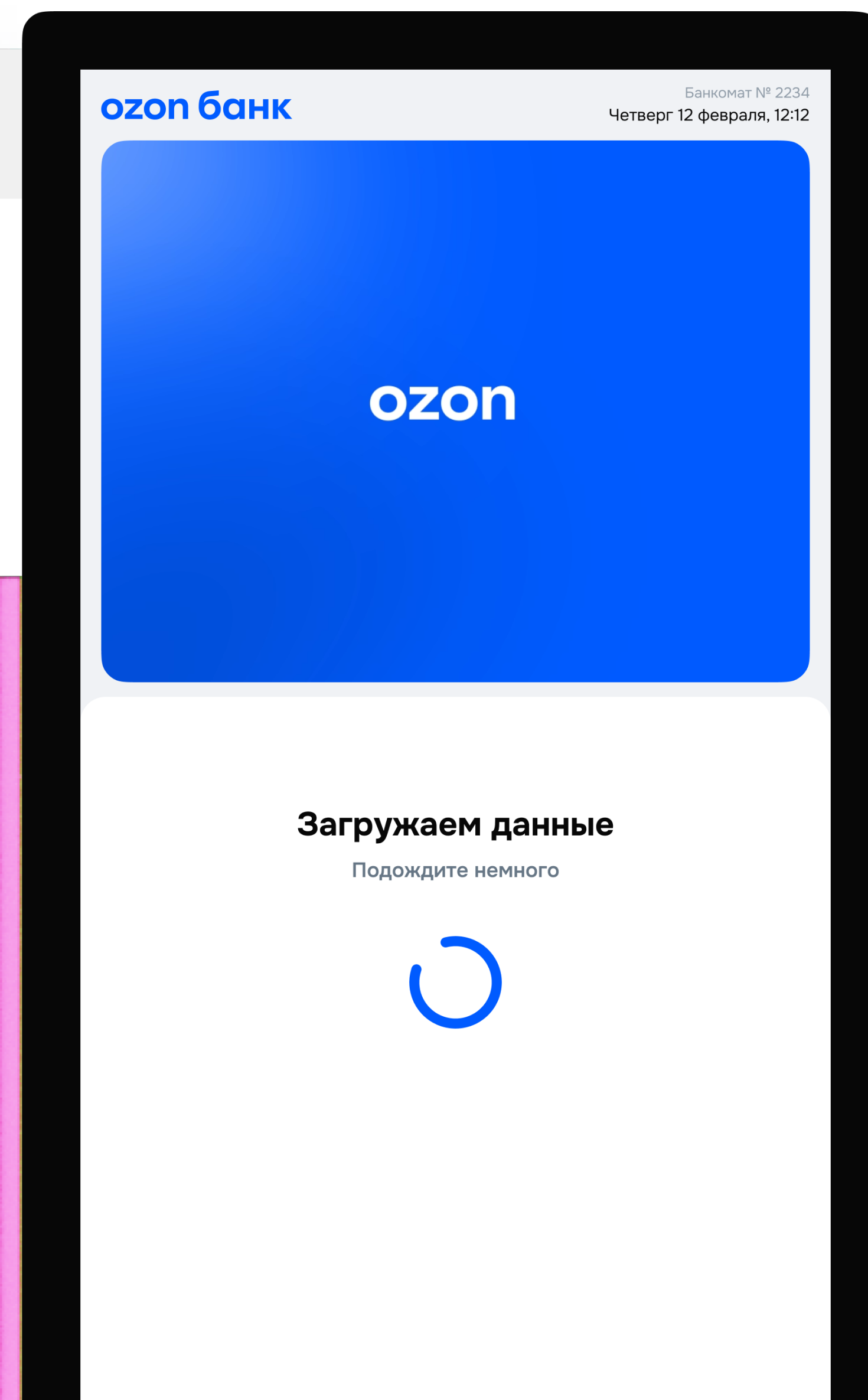
Моки. Проверяем логику событий

```
test('Реакция на события', async ({ page }) => {  
  // 1. Инициализируем virtualAtm  
  await page.addInitScript(() => {  
    window.virtualAtm = new VirtualATM();  
  });  
});
```



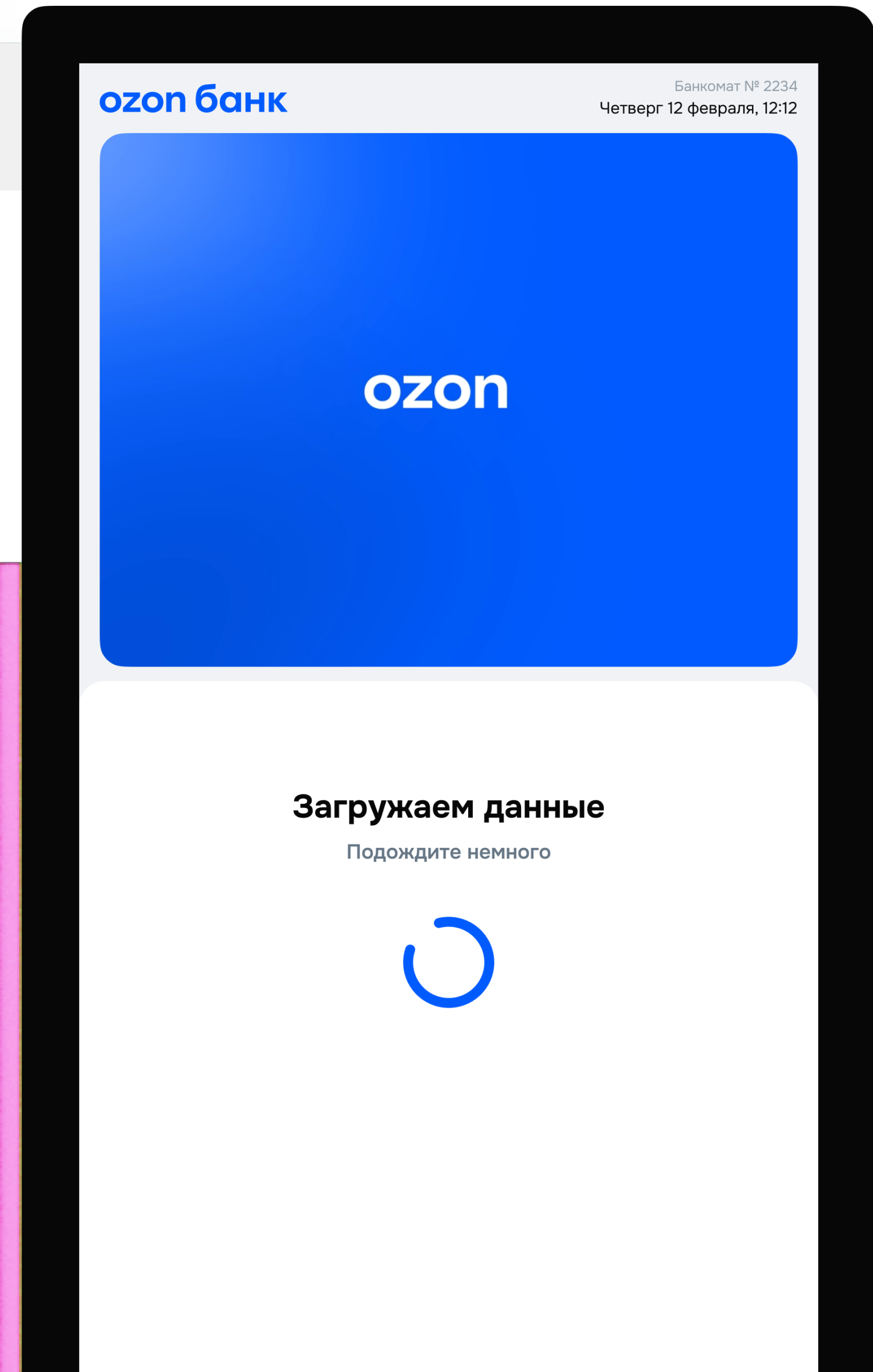
Моки. Проверяем логику событий

```
test('Реакция на события', async ({ page }) => {  
  // 2. Мокаем запросы на backend  
  await page.route('**/api/atm/register', async (route) => {  
    const json = {  
      success: true,  
      status: 'ATM_STATUS_ONLINE'  
    };  
    await route.fulfill({ json });  
  });  
});
```



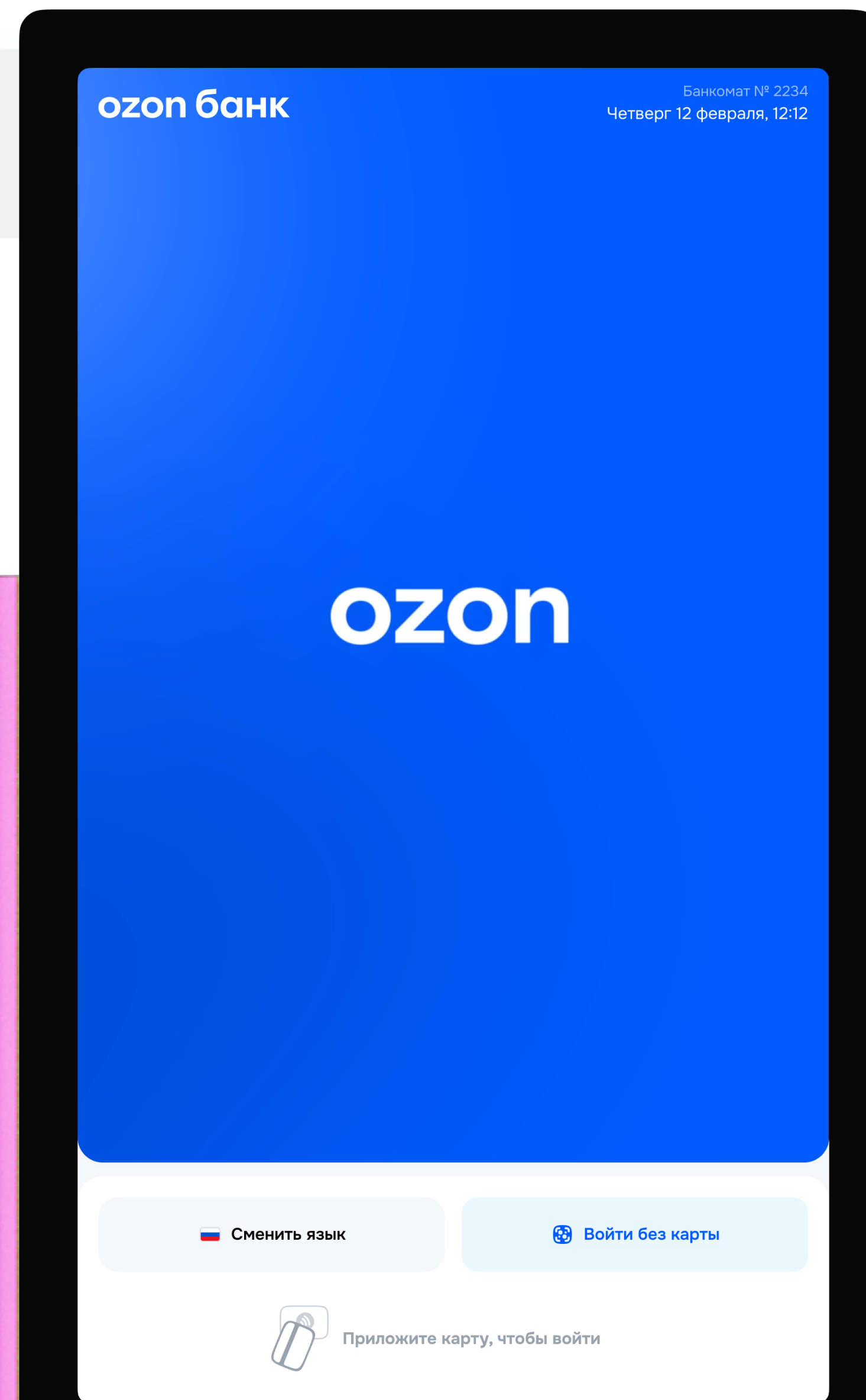
Проверяем логику событий

```
test('Реакция на события', async ({ page }) => {  
  // 2. Мокаем запросы на backend  
  await page.route('**/api/atm/register', async (route) => {  
    const json = {  
      success: true,  
      status: 'ATM_STATUS_ONLINE'  
    };  
    await route.fulfill({ json });  
  });  
});
```



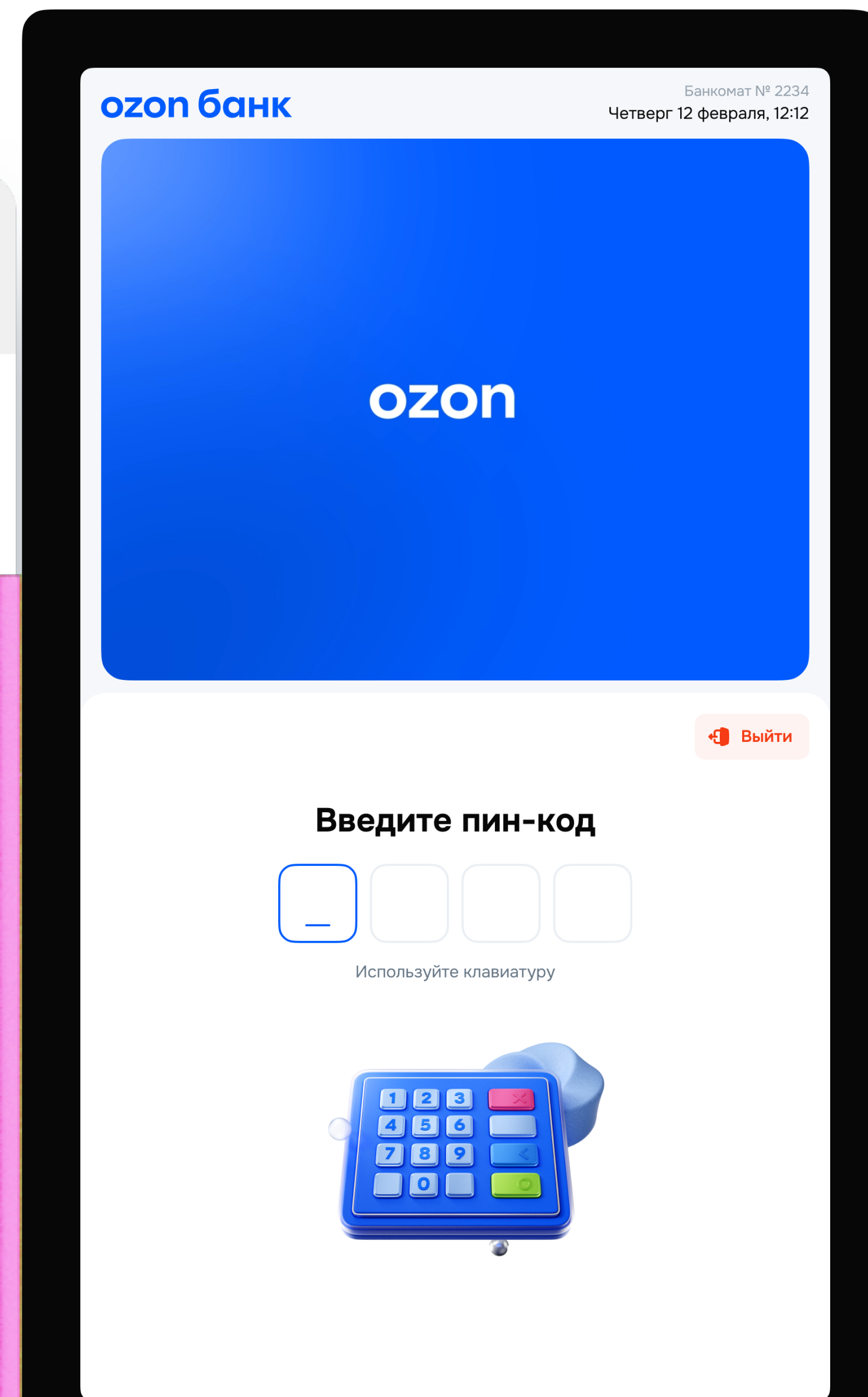
Проверяем логику событий

```
test('Реакция на события', async ({ page }) => {  
  // 3. Открываем страницу приложения  
  await page.goto('/');  
  
  // 4. Вызываем событие  
  await page.evaluate(() => {  
    window.eventBus.send('cardInserted');  
  });  
});
```



Проверяем логику событий

```
test('Реакция на события', async ({ page }) => {  
  // 4. Проверяем результат  
  await expect(  
    page.getByText('Введите ПИН-код'))  
    .toBeVisible();  
});
```



Проверяем верстку



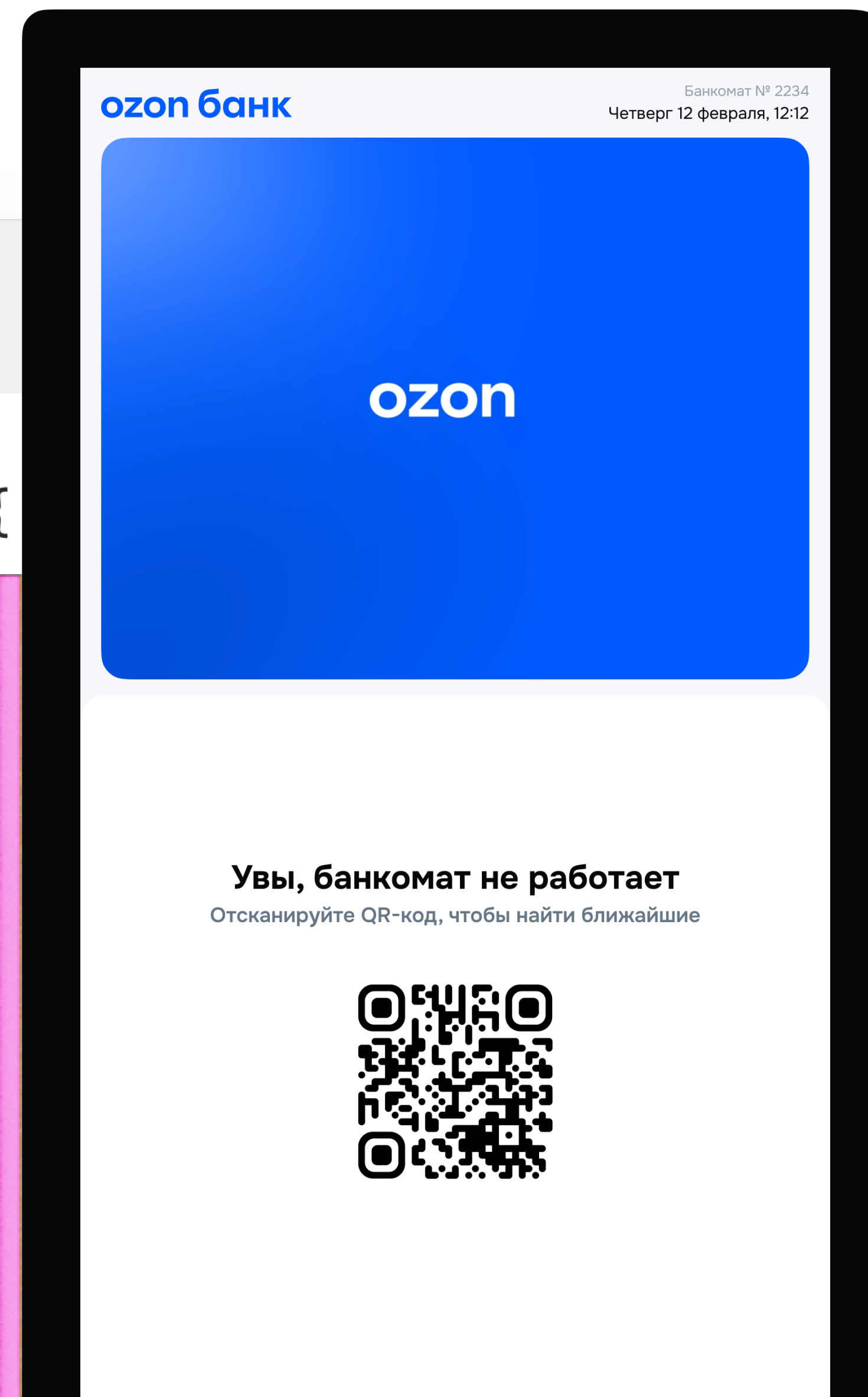
```
test('Верстка экрана', async ({ page }) => {  
  // Можно изменить проверку  
  await expect(page.getByTestId('pin-code-container'))  
    .toBeVisible();  
  await expect(page.getByTestId('all-page-container'))  
    .toHaveScreenshot();  
});
```

Проверяем доступность железа

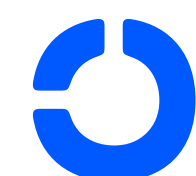
```
test('Реакция на статус железа', async ({ page }) => {  
  // 1. Изменяем состояние модуля  
  await page.addInitScript(() => {  
    window.virtualAtm = {  
      hardwareStatus: {  
        nfcReader: false,  
      },  
    };  
  });  
});
```

Проверяем доступность железа

```
test('Реакция на статус железа', async ({ page }) => {  
  // 2. Переход в состояние "вне обслуживания"  
  await expect(page.getByTestId('oos-overlay'))  
    .toBeVisible();  
});
```



Моки в цифрах



Всего

>400



Время прогона

~5 мин.



Стабильность

99%



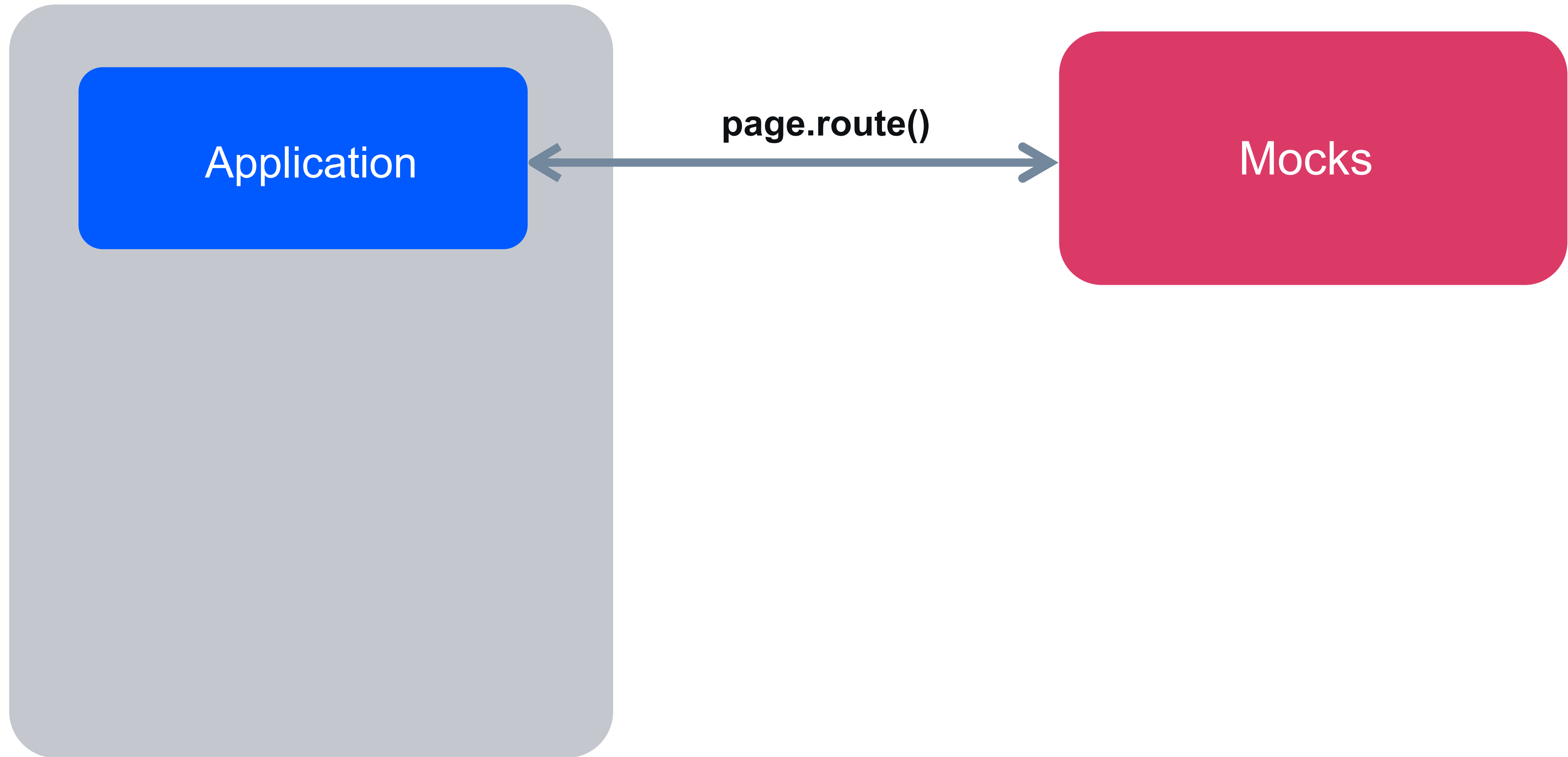
Поддержка

Просто

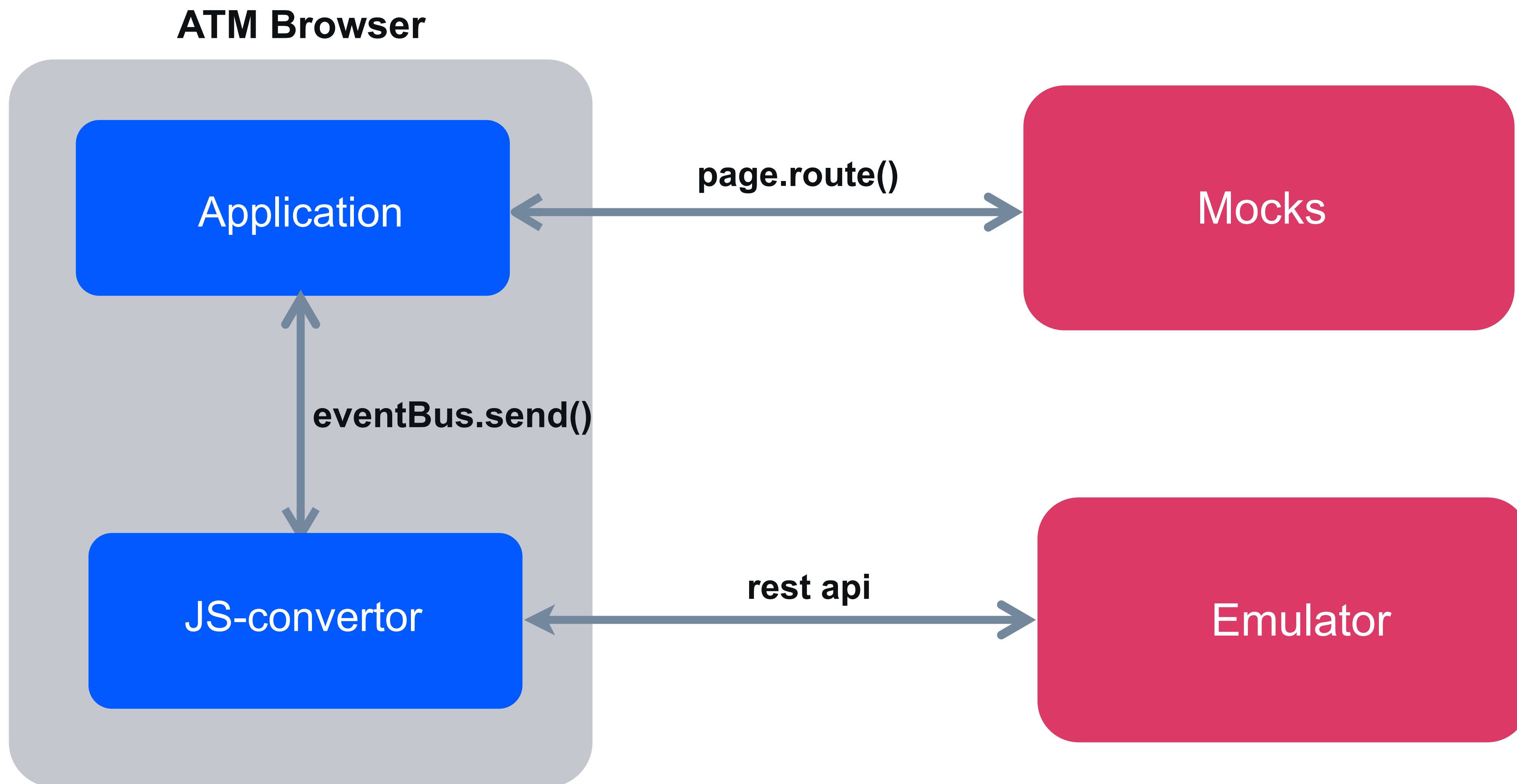


Е2Е тесты через эмульсию железа

Что будем эмулировать



Что будем эмулировать



Что такое эмулятор

```
12:27:49.015 [INFO] Starting ATM Software Emulator... version=v2.1.4-beta commit=a7f8c9b
12:27:49.017 [INFO] Loading configuration path=./config/emulator.yaml
12:27:49.045 [INFO] Initializing hardware modules
12:27:49.102 [INFO] [OK] Card Reader (EMV & Magstripe) interface=/dev/mock_ttyS1
12:27:49.120 [INFO] [OK] EPP (Encrypted PIN Pad) crypto=3DES/DUKPT
12:27:49.155 [INFO] [OK] Cash Dispenser cassettes=4 currency=RUB
12:27:49.180 [INFO] [OK] Receipt Printer paper_roll=100%
12:27:50.005 [INFO] Connecting to Core Banking Host host=10.0.12.44:9000 proto=ISO8583
12:27:50.186 [INFO] Host connection established and active
12:27:50.201 [INFO] Starting gRPC service for UI port=50051
12:27:50.205 [INFO] Starting HTTP Admin API port=8080
12:27:50.210 [INFO] ATM Emulator is successfully running. press_to_stop=Ctrl+C
```

Про Chromium Embedded Framework (CEF)

Про Chromium Embedded Framework (CEF)

01 Умеет рендерить веб-контент



Про Chromium Embedded Framework (CEF)

01 Умеет рендерить веб-контент

02 Ограниченный Web API



Про Chromium Embedded Framework (CEF)

- 01** Умеет рендерить веб-контент
- 02** Ограниченный Web API
- 03** Не умеет создавать контексты и загружать файлы

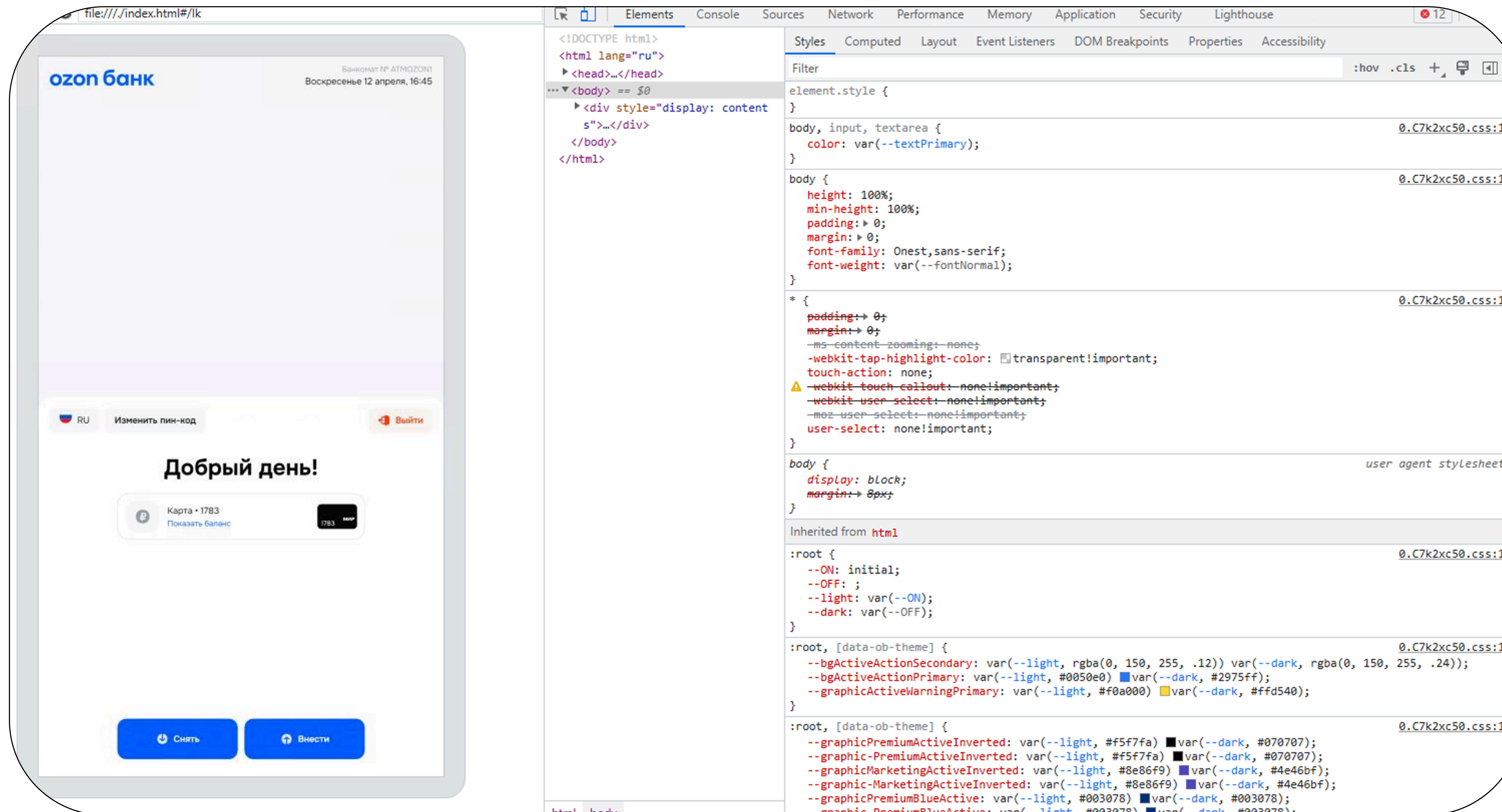


Не забыть запустить



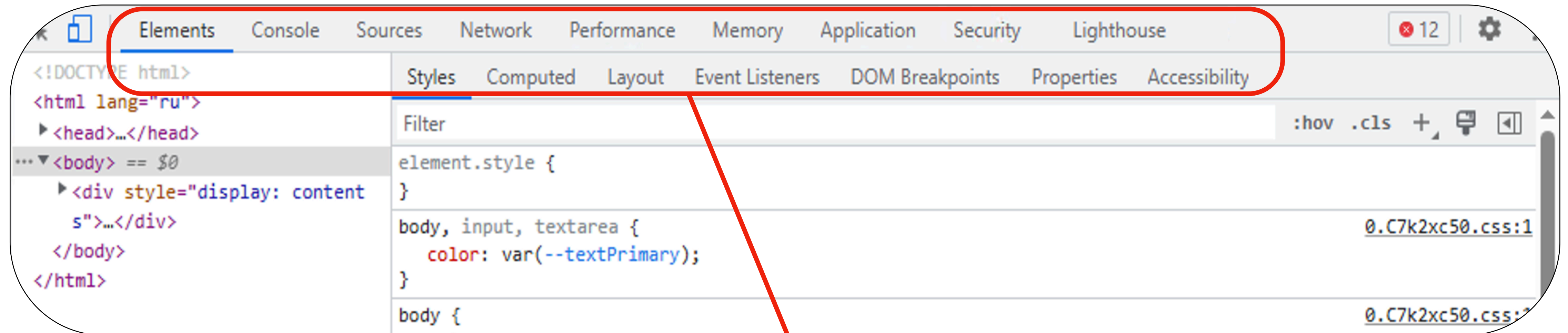
```
cef_browser --url=index.html --remote-debugging-port=9222
```

Доступен Chrome Devtools Protocol (CDP)



<http://127.0.0.1:9222>

Доступен Chrome Devtools Protocol (CDP)



Все привычные вкладки

Патч для запуска по CDP

```
309   async _initialize() {
310     (0, import_assert.assert)(!Array.from(this._browser._crPages.values()).some((page) => page._browserContext === this))
311     const promises = [super._initialize()];
312     // if (this._browser.options.name !== "clank" && this._options.acceptDownloads !== "internal-browser-default") {
313     //   promises.push(this._browser._session.send("Browser.setDownloadBehavior", {
314     //     behavior: this._options.acceptDownloads === "accept" ? "allowAndName" : "deny",
315     //     browserContextId: this._browserContextId,
316     //     downloadPath: this._browser.options.downloadsPath,
317     //     eventsEnabled: true
318     //   }));
319     // }
320     await Promise.all(promises);
321   }
```

<https://github.com/microsoft/playwright/issues/10927>

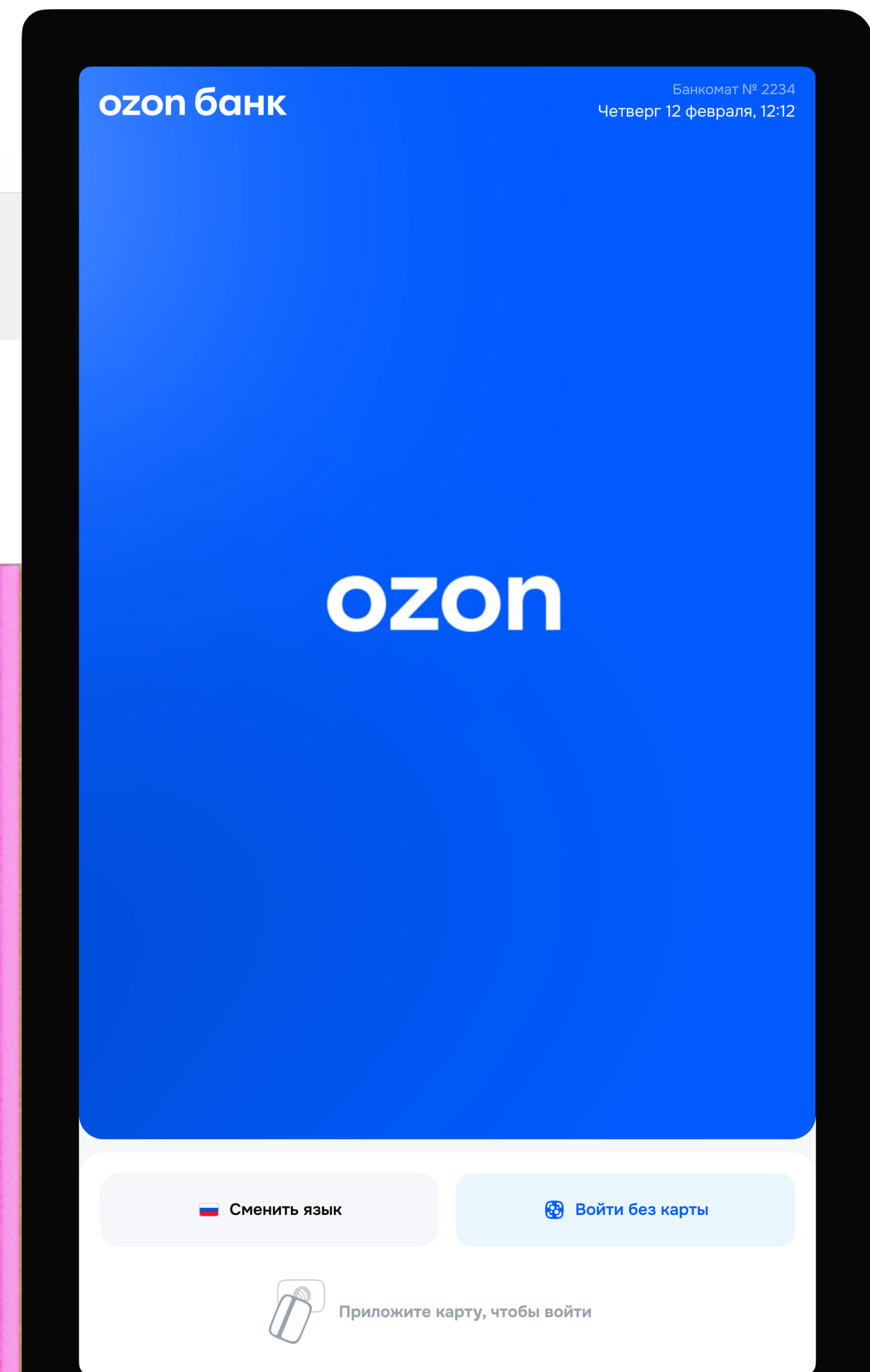
E2E. Проверяем обработку команд



```
test('События нажатия клавиш', async ({ browser }) => {  
  // 1. Подключаемся по CDP к Chromium Framework  
  const browser = await chromium  
    .connectOverCDP('http://127.0.0.1:9222');  
  const defaultContext = browser.contexts()[0];  
  const page = defaultContext.pages()[0];  
  
  // Переходим в приложение  
  await page.goto('/');
```

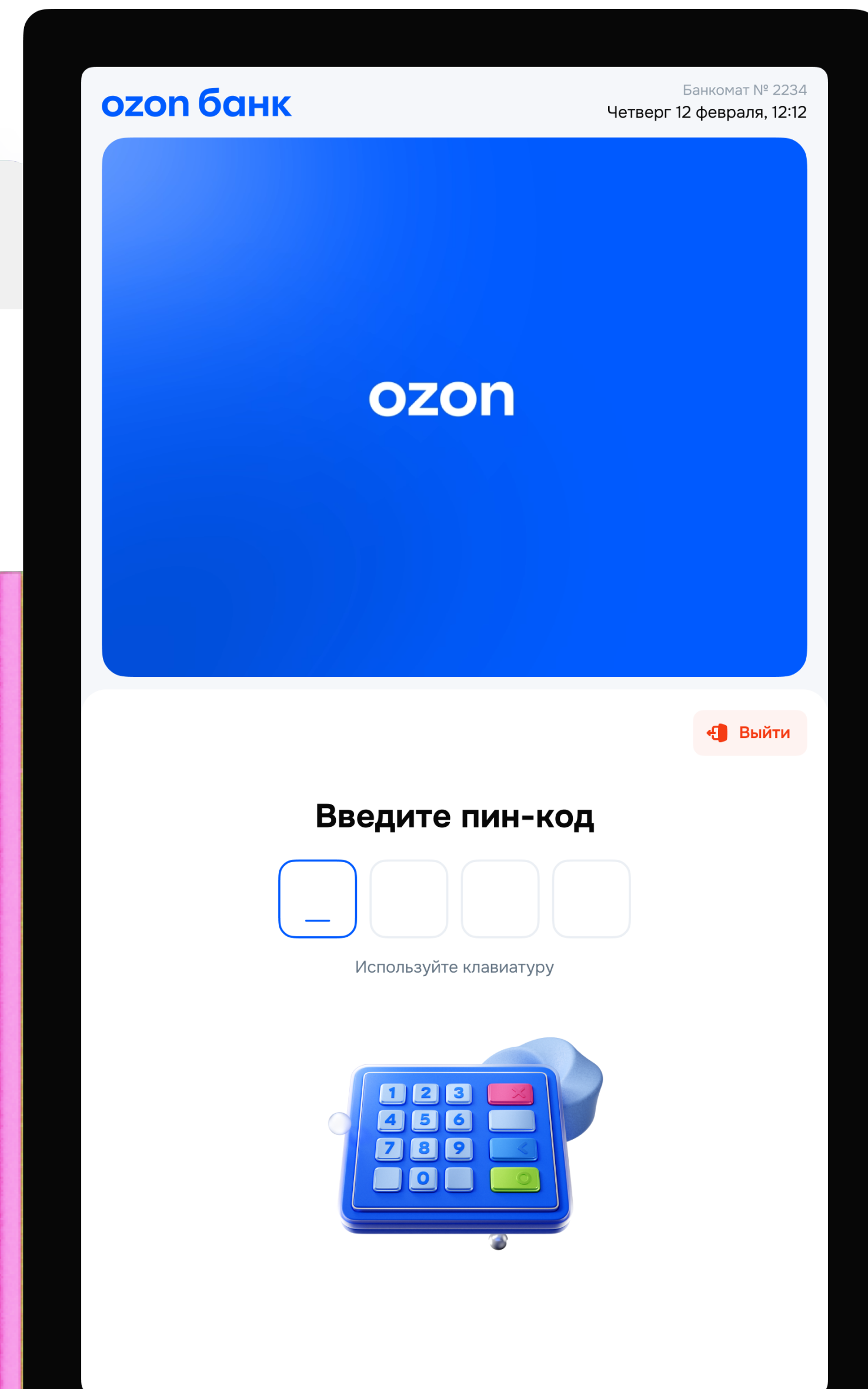
E2E. Проверяем обработку команд

```
test('События нажатия клавиш', async ({ page, request }) => {  
  // 2. Вызываем событие чтения карты  
  await request.post('/emulator/event', {  
    data: {  
      state: 'card_09211',  
    },  
  });  
});
```



E2E. Проверяем обработку команд

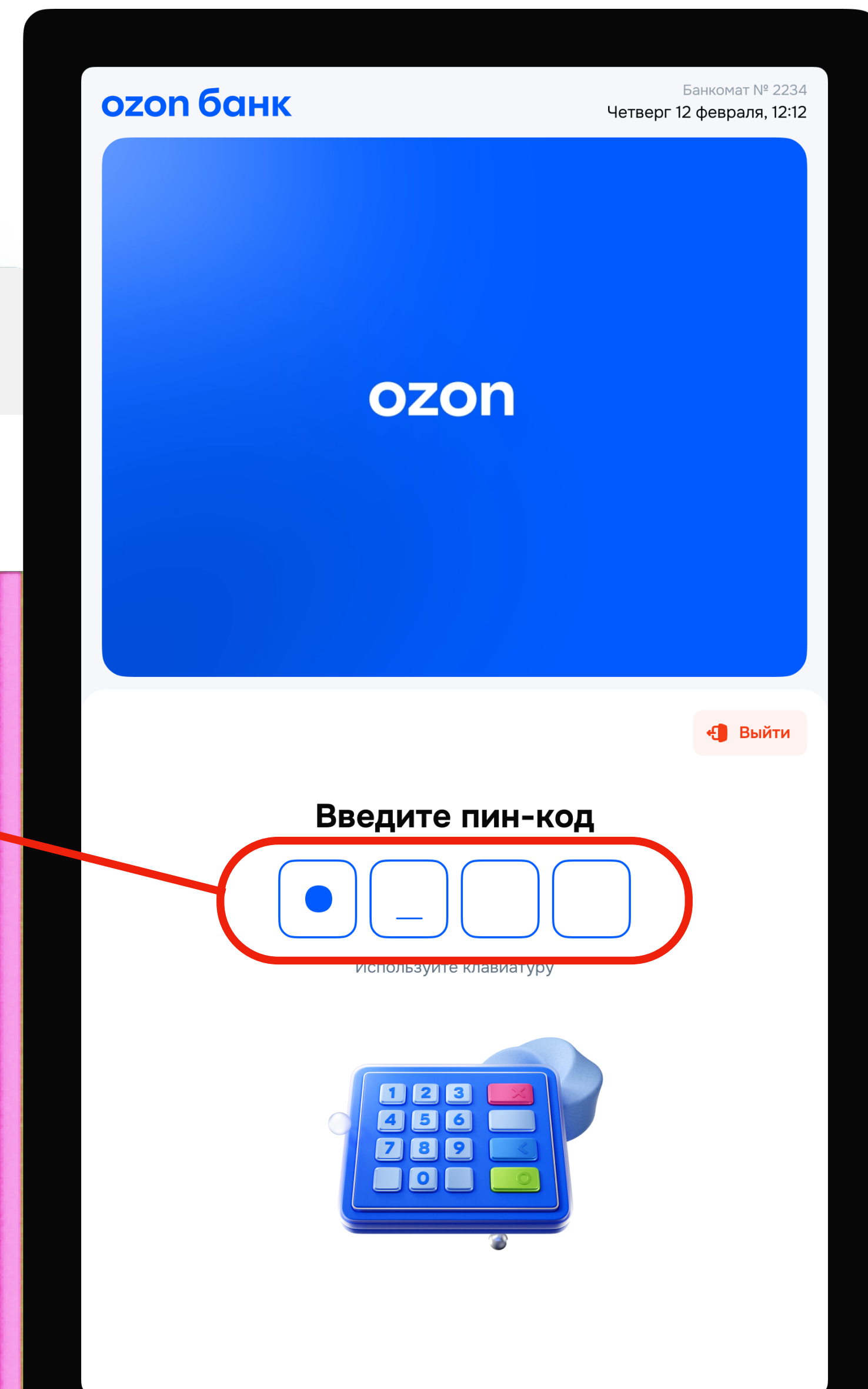
```
test('События нажатия клавиш', async ({ page, request }) => {  
  // 3. Вызываем событие клавиатуры  
  await request.post('/emulator/event', {  
    data: {  
      source: 'pin-pad',  
      value: '5',  
    },  
  });  
});
```



E2E. Проверяем обработку команд



```
test('События нажатия клавиш', async ({ page, request }) => {  
  // 4. Проверяем результат  
  await expect(page.getByTestId('pin-value'))  
    .toBeVisible();  
});
```



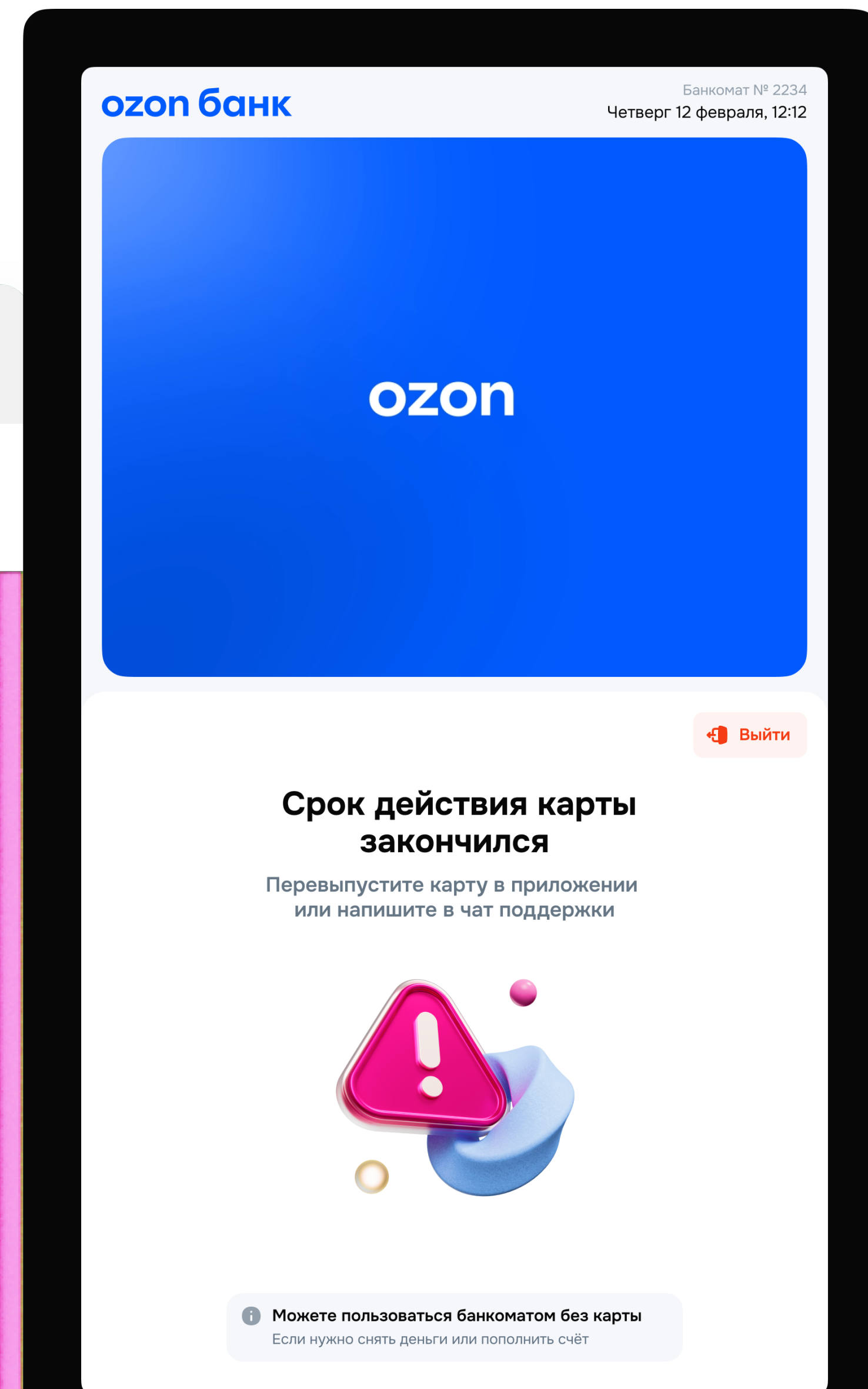
E2E. Проверяем обработку ошибок

```
test('Ошибка с управляющего ПО', async ({ page, request }) => {  
  // Вызываем ошибку чтения карты  
  await request.post('/emulator/error', {  
    data: {  
      lastError: {  
        code: '0x0421f',  
        source: 'cardReader',  
      },  
    },  
  });  
});
```

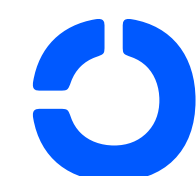
E2E. Проверяем обработку ошибок



```
test('Ошибка с управляющего ПО', async ({ page, request }) => {  
  // Проверяем результат  
  await expect(page.getByText('Срок действия карты закончился'))  
    .toBeVisible();  
});
```



E2E в цифрах



Всего

>25



Время прогона

~15 мин.



Стабильность





80%



Поддержка

Сложно

Сравним метрики

	E2E	Моки
 Всего	>25	>400
 Время прогона	~15 мин.	~5 мин.
 Стабильность	80%	99%
 Поддержка	Сложно	Просто

Вывод

Вывод



UI банкомата можно тестировать как веб-приложение

Вывод



UI банкомата можно тестировать как веб-приложение



Но важно помнить его особенности

Вывод

- UI банкомата можно тестировать как веб-приложение
- Но важно помнить его особенности
- Логику пользователя автоматизировать просто

Вывод

- UI банкомата можно тестировать как веб-приложение
- Но важно помнить его особенности
- Логику пользователя автоматизировать просто
- А интеграцию с железом - нет

ozon банк

**Спасибо за
внимание!**

