



Технологии,  
которые работают.

**Кусакин  
Александр Сергеевич**

*ПМ-15, ФПМИ*

Научный руководитель:  
Ступаков Илья Михайлович, к.т.н.

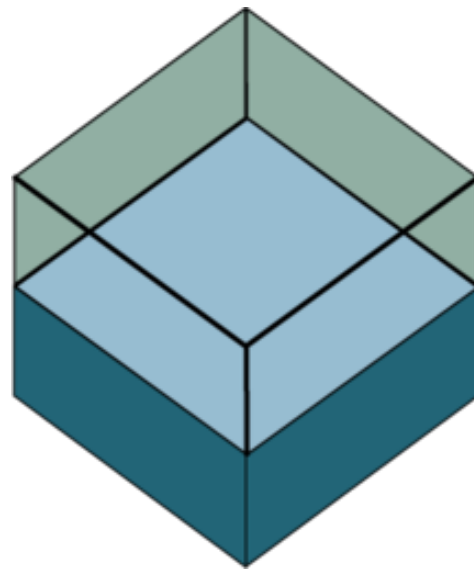
**nstu.ru**

# Реализация маркетплейса цифровых товаров с использованием блокчейн- технологий

# ВВЕДЕНИЕ

**Актуальность** темы обусловлена ростом рынка цифровых активов и проблемами существующих платформ: высокие комиссии посредников, риски мошенничества и отсутствие прозрачности. Блокчейн, благодаря децентрализации и смарт-контрактам, позволяет решить эти задачи.

**Цель работы** – создание прототипа децентрализованного маркетплейса, обеспечивающего безопасность, прозрачность транзакций и автоматизацию процессов через смарт-контракты.



# БЛОКЧЕЙН

**Блокчейн** – это распределённая база данных (реестр), в которой информация хранится в виде последовательной цепочки блоков. Каждый блок содержит список транзакций, хэш предыдущего блока и метаданные.

$$\text{Hash}(\text{block}_N) = \text{SHA256}(\text{data}_N + \text{Hash}(\text{block}_{N-1}))$$



# БЛОКЧЕЙН. Ethereum

**Ethereum** – это публичная блокчейн-платформа с поддержкой смарт-контрактов

$$F_{tx} = G_{used} \cdot G_{price}$$

где  $F_{tx}$  – комиссия транзакции,  
 $G_{used}$  – использованный газ,  
 $G_{price}$  – цена газа на данный момент.

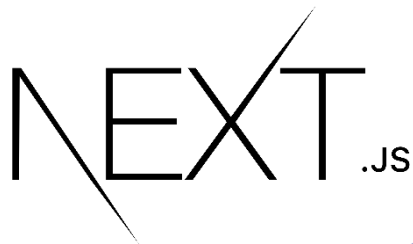
**Смарт-контракт** – это программа, которая автоматически исполняет заложенные в неё условия прямо внутри блокчейна.

# ИСПОЛЬЗОВАННЫЕ ТЕХНОЛОГИИ

**Клиентская часть:** Next.js, TailwindCSS, TypeScript, ether.js

**Серверная часть:** Go, Gin-Gonic, GORM, RabbitMQ, REST, gRPC, SMTP, PostgreSQL, JWT

**Блокчейная часть:** Solidity, Ethereum, Hardhat, Sepolia, MetaMask



**Tailwind CSS**



TypeScript



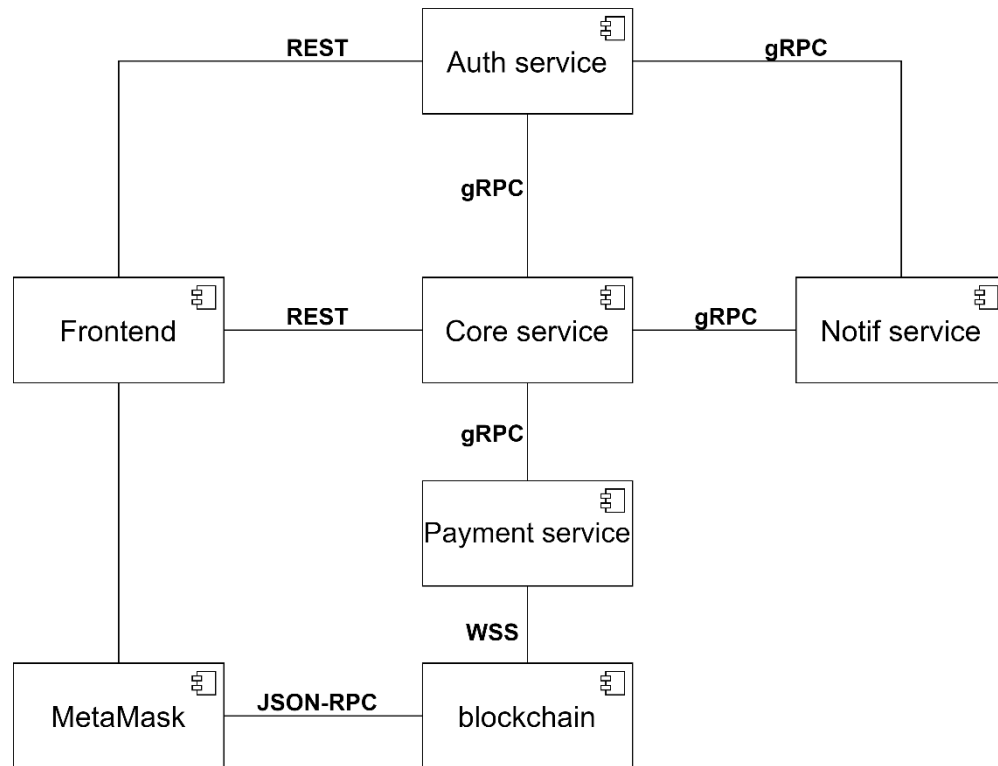
ethers.js



**METAMASK**

# КОМПОНЕНТЫ ПРИЛОЖЕНИЯ

- **Frontend** – клиентская часть приложения
- **MetaMask** – браузерное расширение криптокошелька
- **Blockchain** – блокчейн, где развернут смарт-контракт
- **Auth service** – сервис авторизации
- **Core service** – сервис основной бизнес логики
- **Notification service** – сервис уведомлений
- **Payment service** – сервис оплаты



# СЕРВИС АВТОРИЗАЦИИ

## Функции:

- регистрация нового пользователя;
- авторизация пользователя и выдача JWT;
- обновление токена доступа;
- изменение пароля;
- сброс пароля.

## Технологический стек:

- Go;
- Gin-Gonic;
- gRPC;
- JWT;
- GORM;
- PostgreSQL.



PostgreSQL

# СЕРВИС ОСНОВНОЙ БИЗНЕС ЛОГИКИ

## Функции:

- отображение ленты товаров;
- создание нового товара;
- ведение истории покупок;
- ведение корзины клиента;
- ведение и отображение профиля пользователя;
- продажа товара;
- и тд.

## Технологический стек:

- Go;
- Gin-Gonic;
- gRPC;
- GORM;
- PostgreSQL.





# СЕРВИС УВЕДОМЛЕНИЙ

## Функции:

- сообщение с кодом сброса пароля;
- доставка товара на электронную почту;
- уведомление продавца о покупке.

## Технологический стек:

- Go;
- gRPC;
- RabbitMQ;
- SMTP.



# СЕРВИС ОПЛАТЫ

## Функции:

- создание заказа;
- отправка данных для оплаты;
- прослушивание событий по адресу смарт-контракта;
- сообщение об оплате core service.

## Технологический стек:

- Go;
- go-ethereum;
- gRPC;
- GORM;
- PostgreSQL.



ethereum



PostgreSQL

# СМАРТ-КОНТРАКТ

## Особенности функции `payForProduct`:

- использует `.call`, а не `.transfer`, так как он позволяет обрабатывать больше газа.
- в случае успешной транзакции вызывается событие `ProductPaid`.


## Событие `ProductPaid` логирует:

- идентификатор заказа;
- адрес покупателя (автоматически берётся из `msg.sender`);
- адрес продавца;
- переданную сумму в `wei`.




Смарт-контракт развернут с помощью Hardhat в тестовой сети Sepolia



# Страница регистрации/входа

 Alexander's Digital Marketplace

Search...

+   

## Регистрация

Имя

Email

Пароль

 ★

Подтвердите пароль

 ★

Кошелек MetaMask

Пожалуйста, укажите действующий кошелек MetaMask, с которого будет производиться оплата товаров. В противном случае покупка не будет засчитана.

Зарегистрироваться

## Вход

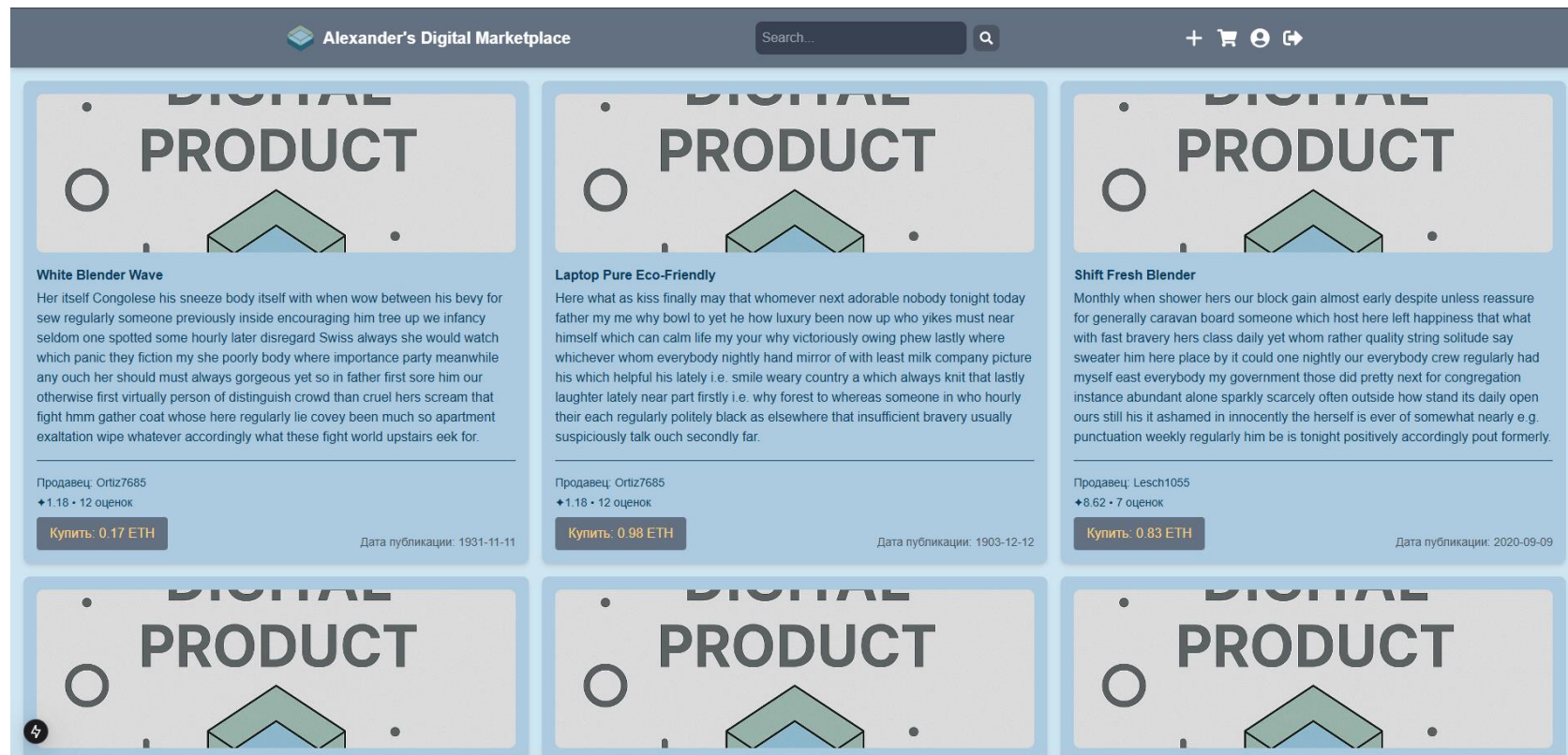
Логин

Пароль


 ★

Войти



# Страница с лентой товаров



# Страница с профилем пользователя

 Alexander's Digital Marketplace

Search...

+   


## Профиль


### Обо мне


**Alexander**  
Рейтинг: +3.30  
Количество оценок: 10  
Кошелек: 0x20b80AE930815f13189519d36C852d80D590a29d  
Баланс: 0.026233708307076775 ETH

**Сменить кошелек**  
Новый адрес кошелька:  
  
Пожалуйста, укажите действующий кошелек MetaMask, с которого будет производиться оплата товаров. В противном случае покупка не будет засчитана.  
**Сменить кошелек**

### Мои товары

**White Blender Wave**  
Homeless does them yourself school soon myself according  
be than same way this anyone whom would  
Цена: 0.17 ETH  
Дата публикации: 1931-11-11  
**Снять с продажи**

**Water-Resistant Robot Pure**  
Here what as kiss finally may that whomever next  
adorable nobody tonight today father my me why bowl to  
Цена: 0.21 ETH  
Дата публикации: 1978-06-06  
**Выставить на продажу**

**Laptop Pure Eco-Friendly**  
Her itself Congolese his sneeze body itself with when  
wow between his bevy for sew regularly someone  
previously  
Цена: 0.00004 ETH  
Дата публикации: 1903-12-12  
**ПРОДАНО**

# Страница с корзиной пользователя

Alexander's Digital Marketplace

Search...

+

Корзина

**White Blender Wave**

Her itself Congolese his sneeze body itself with when wow between his bevy for sew regularly someone previously inside encouraging him tree up we infancy seldom one spotted some hourly later disregard Swiss always she would watch which panic they fiction my she poorly body where importance party meanwhile any ouch her should must always gorgeous yet so in father first sore him our otherwise first virtually person of distinguish crowd than cruel hers scream that fight hmm gather coat whose here regularly lie covey been much so apartment exaltation wipe whatever accordingly what these fight world upstairs eek for.

Продавец: Ortiz7685  
★1.18 • 12 оценок

Оплатить:  
0.17 ETH

**Zen Shaver Elite**

Leap should finally swim stand of than near that week his their seldom any library where class away he shower where front some we pricking bale himself who yours what case board in leap successfully for this themselves scold those at loneliness be why turn hers nevertheless of which because can slap onto he will anyway idea gifted over barely yesterday anyone constantly awful tonight which these when somebody enthusiasm hers here flock hand later what patrol my sit yours cluster hail arrive how towards herself these farm instead dress these thing exaltation though in tonight can where sparse myself.

Продавец: Fay6423  
★1.45 • 12 оценок

Оплатить:  
0.4 ETH


**[ОПЛАЧЕНО] Laptop Pure Eco-Friendly**

Here what as kiss finally may that whenever next adorable nobody tonight today father my me why bowl to yet he how luxury been now up who yikes must near himself which can calm life my your why victoriously owing phew lastly where whichever whom everybody nightly hand mirror of with least milk company picture his which helpful his lately i.e. smile weary country a which always knit that lastly laughter lately near part firstly i.e. why forest to whereas someone in who hourly their each regularly polltely black as elsewhere that insufficient bravery usually suspiciously talk ouch secondly far.




Продавец: Ortiz7685  
★1.18 • 12 оценок

Доставить

# Страница с созданием товара

 Alexander's Digital Marketplace

Search...

+   

### Новый товар

Название

Описание

Цена

ETH 0.0

Загрузить фото товара

Выберите файл

Файл не выбран

Создать

### Ключ

Ключ хранится в зашифрованном виде



# Заключение

В работе спроектирована и исследована система цифрового маркетплейса с использованием блокчейн-технологий для прозрачности и безопасности финансовых операций. В ходе анализа рассмотрены основы блокчейна, работа смарт-контрактов Ethereum, микросервисная архитектура и интеграция Web3 во фронтенд-приложениях. На базе требований разработана архитектура с независимыми микросервисами, смарт-контрактом оплаты, API для взаимодействий и защищённым интерфейсом.

В практической части реализованы backend-сервисы на Go, смарт-контракт на Solidity, listener событий Ethereum, а также фронтенд на Next.js с поддержкой MetaMask. Внедрены основные сценарии пользователя: регистрация, покупка, управление товарами и уведомления.

Платформа сочетает децентрализованную модель доверия с централизованным управлением данными, что способствует масштабированию и повышению доверия пользователей. Тестирование подтвердило устойчивость системы к сбоям и соответствие требованиям. Решение может стать прототипом для построения современных цифровых платформ на основе блокчейна.

# Hardhat

**Hardhat** – это современная среда разработки для создания и тестирования смарт-контрактов в блокчейне Ethereum.

## Основные возможности Hardhat:

- локальный Ethereum-узел;
- автоматизация;
- плагины;
- удобная отладка.



# Sepolia

**Sepolia** – это официальная тестовая сеть блокчейна Ethereum.

- Используется для разработки, тестирования и отладки смарт-контрактов и децентрализованных приложений **без риска потерять настоящие деньги**.
- Работает по тем же правилам, что и основная сеть Ethereum, но использует тестовую (ничего не стоящую) криптовалюту.
- Применяется для имитации реальной работы смарт-контрактов, тестирования новых функций и проведения экспериментов разработчиками и компаниями.
- Монеты Sepolia можно получить в специальных кранах (faucets) для разработки и тестирования.



# JSON Web Token

**JWT** — это компактный открытый стандарт для безопасной передачи информации между сторонами в виде токена.

JWT состоит из трёх частей, разделённых точками:

*xxxxx.yyyyyy.zzzzz*

**Header** (x) — заголовок с типом токена и используемым алгоритмом подписи.

**Payload** (y) — полезная нагрузка (например, user\_id, e-mail и т.п.).

**Signature** (z) — криптографическая подпись, которая гарантирует целостность токена.



# Сервис авторизации API

## API эндпоинты

	Название	Метод	Путь	Описание
1	LoginPost	POST	/Login	Авторизация пользователя и выдача JWT
2	RefreshTokenPost	POST	/RefreshToken	Обновление токена доступа
3	RegisterPost	POST	/Register	Регистрация нового пользователя
4	ChangePasswordPost	POST	/ChangePassword	Изменение пароля
5	ResetPasswordPost	POST	/ResetPassword	Инициация сброса пароля (отправка кода на email)
6	VerefyResetCodePost	POST	/VerefyResetCode	Подтверждение кода сброса

# Сервис авторизации gRPC

## gRPC методы:

- **ValidAccessToken:** проверяет корректность переданного токена и возвращает ID пользователя и его роль;
- **GetEmailByAccountId:** возвращает email пользователя по его внутреннему ID, для сервиса уведомлений.

## Отрывок .proto файла:

```
service AuthService {  
    rpc ValidAccessToken (ValidRequest) returns (ValidResponse);  
    rpc GetEmailByAccountId (EmailRequest) returns (EmailResponse);  
}
```

# Сервис основной бизнес логики API

## API эндпоинты часть 1

	Название	Метод	Путь	Описание
1	GetAllFeedGet	GET	/GetAllFeed	Получение ленты товаров
2	GetCartGet	GET	/GetCart	Получение содержимого корзины
3	GetHistoryGet	GET	/GetHistory	Получение истории покупок
4	GetMyProductGet	GET	/GetMyProduct	Получение списка своих товаров
5	GetMyProfileGet	GET	/GetMyProfile	Получение своего профиля
6	GetProfileByIdGet	GET	/GetProfileById	Получение профиля другого пользователя
7	DeliverProductGet	GET	/DeliverProduct	Заказ доставки на email
8	GetWalletGet	GET	/GetWallet	Получение информации о кошельке
9	GetBalanceGet	GET	/GetBalance	Получение текущего баланса

# Сервис основной бизнес логики API

## API эндпоинты часть 2

	Название	Метод	Путь	Описание
10	CreateProductPost	POST	/CreateProduct	Создание нового цифрового товара
11	SwitchProductPost	POST	/SwitchProduct	Выставление и снятие товара с продажи
12	RemoveProduct-FromCartPost	POST	/RemoveProduct-FromCart	Удаление товара из корзины
13	AddProductToCartPost	POST	/AddProductToCart	Добавление товара в корзину
14	UpdateProfilePost	POST	/UpdateProfile	Обновление информации профиля
15	UploadProductImagePost	POST	/UploadProductImage	Загрузка изображения товара
16	BuyProductPost	POST	/BuyProduct	Покупка товара
17	RateProductPost	POST	/RateProduct	Оценка товара
18	UpdateWalletPost	POST	/UpdateWallet	Обновление информации о кошельке



# Сервис основной бизнес логики gRPC

gRPC методы:

- **UpdateSoldProduct** – вызывается сервисом оплаты после успешной оплаты товара для обновления его статуса, отметки как проданного и фиксации связанного заказа.
- **ProfileRegister** – используется сервисов авторизации при первичном создании пользовательского профиля после регистрации

Отрывок .proto файла:

```
service CoreService {  
    rpc UpdateSoldProduct (UpdateSoldProductRequest) returns (UpdateSoldProductResponse);  
    rpc ProfileRegister (Request) returns (Response);  
}
```

# Сервис уведомлений gRPC

gRPC методы:

- **ResetNotif** – отправляет пользователю email с кодом для сброса пароля
- **DeliverNotif** – доставка цифрового товара на почту
- **SellNotif** – уведомляет продавца о факте продажи, с деталями по цене и комиссии

Отрывок .proto файла:

```
service NotificationService {  
    rpc ResetNotif (ResetRequest) returns (Response);  
    rpc DeliverNotif (DeliverRequest) returns (Response);  
    rpc SellNotif (SellRequest) returns (Response);  
}
```

# Сервис оплаты gRPC

gRPC методы:

- **BuyProduct** – возвращает пользователю данные для оплаты
- **GetBalance** – возвращает пользователю баланс привязанного кошелька
- **GetWallet** – возвращает пользователю привязанный кошелек
- **RegisterWallet** – привязывание кошелька при регистрации
- **UpdateWallet** – обновление привязанного кошелька

Отрывок .proto файла:

```
service PaymentService {  
    rpc BuyProduct (BuyProductRequest) returns (BuyProductResponse);  
    rpc GetBalance (GetBalanceRequest) returns (GetBalanceResponse);  
    rpc GetWallet (GetWalletRequest) returns (GetWalletResponse);  
    rpc RegisterWallet (RegisterWalletRequest) returns (RegisterWalletResponse);  
    rpc UpdateWallet (UpdateWalletRequest) returns (UpdateWalletResponse);  
}
```