



Pink Phantasm



# Hieda

@pinkphantasm



pinkphantasm/hieda

# Команда разработчиков

Никита

Backend engineer

Иван

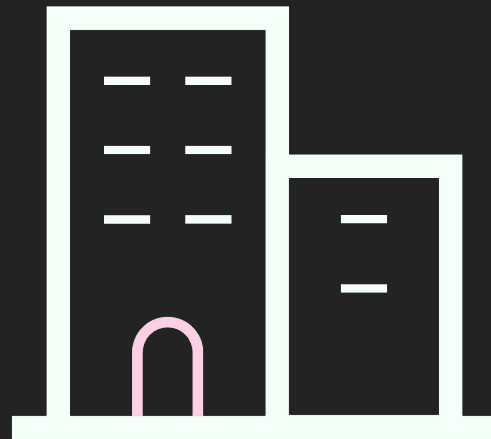
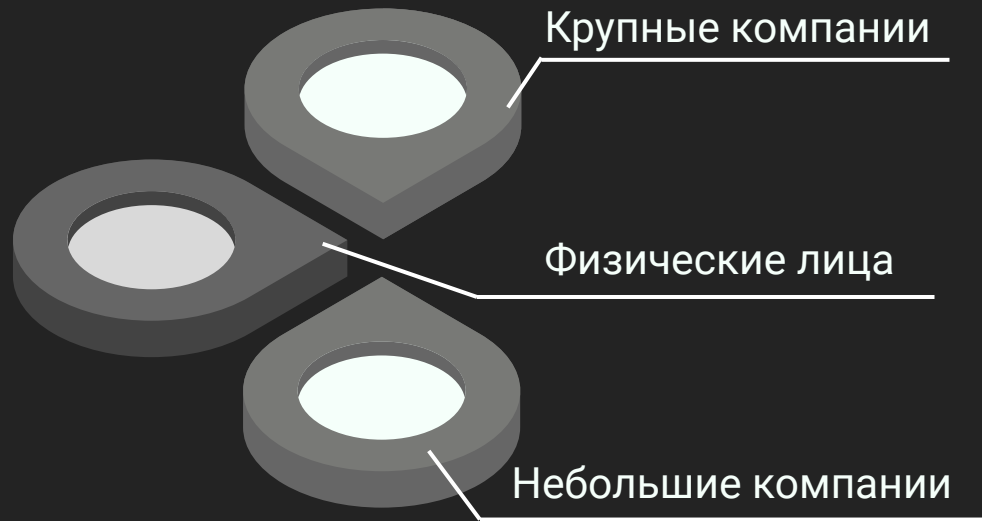
Backend & DevOps  
engineer, Software  
Architect, Team Leader



Даниил

Information security  
engineer, Designer

# Целевая аудитория и компании-конкуренты



- Удостоверяющий центр КОНТУР
- НТПЦ
- F.Doc

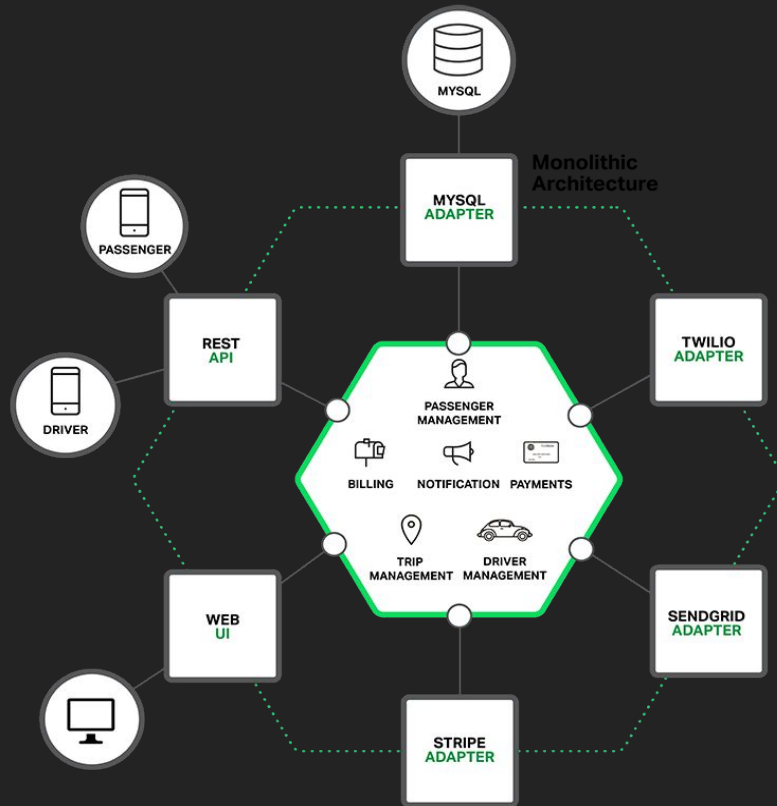
# Архитектура: Монолит

Плюсы:

😊 Простота

Минусы:

- ⚠️ Неудобство при масштабировании
- 🚫 Ограниченный стек технологий
- 🔗 Слабая отказоустойчивость



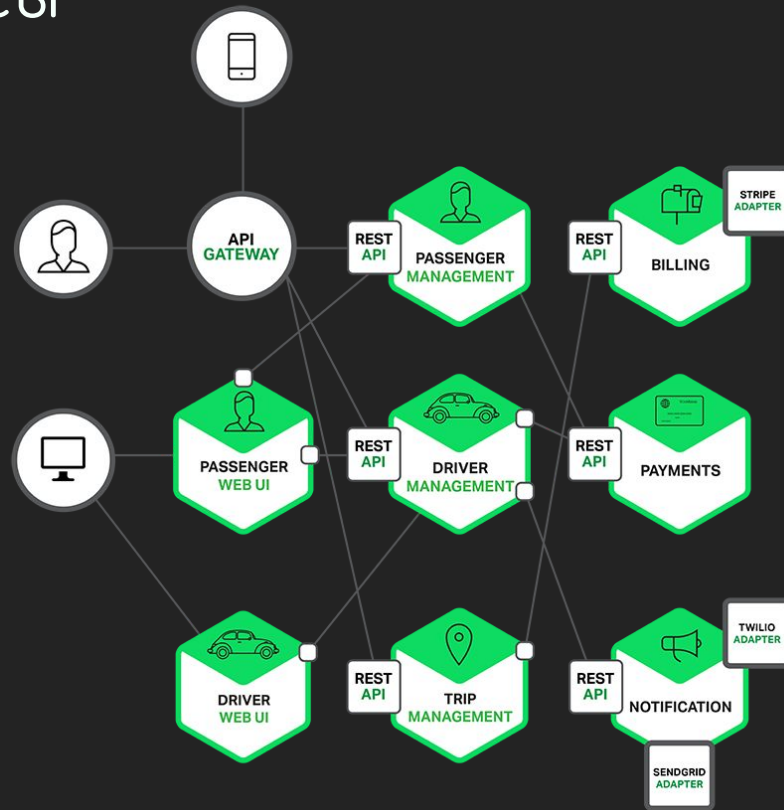
# Архитектура: Микросервисы

## Плюсы:

- 🔗 Отказоустойчивость
- ↻ Гибкость
- 📈 Масштабируемость

## Минусы:

- 😞 Сложность разработки
- 👤 Требования к DevOps



# Ког: S3 Adapter



Сохранение документов в  
объектное хранилище Yandex S3



Ссылка на скачивание документа



Удобная и безопасная передача  
файлов

```
app/internal/s3_adapter.py

from uuid import uuid4

from ..config import s3, config

class S3Adapter:
    def __init__(self, s3, bucket, download_url) → None:
        self.s3 = s3
        self.bucket = bucket
        self.download_url = download_url

    def upload(self, file, filename):
        object_key = f"hieda_{uuid4()}_{filename}"
        self.s3.upload_fileobj(file, self.bucket, object_key)
        return f"{self.download_url}/{object_key}"

def new_s3_adapter():
    return S3Adapter(s3, config.bucket, config.download_url)
```

# Yandex @ Cloud

1. Микросервис обмена документами

# Ког: Crypto Adapter

**RSA**

Удостоверение с использованием пары RSA-ключей



Шифрование с помощью публичного ключа



Верификация с помощью приватного ключа

## 2. Микросервис сертификации

```
internal/pkg/crypto/crypto.go

type Adapter struct {
    privateKey *rsa.PrivateKey
}

func NewAdapter() *Adapter {
    cfg := config.New()

    privateKeyBytes, err := os.ReadFile(cfg.PrivateKey)
    if err != nil {
        panic(err)
    }

    privateKeyBlock, _ := pem.Decode(privateKeyBytes)

    privateKey, err := x509.ParsePKCS8PrivateKey(privateKeyBlock.Bytes)
    if err != nil {
        panic(err)
    }

    return &Adapter{
        privateKey: privateKey.(*rsa.PrivateKey),
    }
}

func (a Adapter) Sign(data []byte) ([]byte, error) {
    return rsa.EncryptPKCS1v15(rand.Reader, &a.privateKey.PublicKey, data)
}

func (a Adapter) Verify(data []byte) ([]byte, error) {
    return rsa.DecryptPKCS1v15(rand.Reader, a.privateKey, data)
}

func (a Adapter) Compare(data, signed []byte) bool {
    verifiedData, err := a.Verify(signed)
    if err != nil {
        return false
    }

    return bytes.Equal(data, verifiedData)
}
```

# Код: Мониторинг статуса инфраструктуры

Hieda

## Hieda Status

Static Operational (expand)

Index Operational (expand)

Upload Operational (expand)

Certification Operational (hide)

Details: All systems operational  
Internal address: http://cert:3004  
Timestamp: 2024-04-14 09:42:40.876566038 +0000 UTC  
m=+3.016607591

## 3. Микросервис статусов

```
internal/pkg/health/adapter.gi

func (a Adapter) CheckHealth(services []*service.Service) []*Health {
    var responses []*Health

    for _, srv := range services {
        apiUrl, err := url.JoinPath(srv.Addr, "api", "health")
        if err != nil {
            continue
        }

        r, _ := http.NewRequest(http.MethodGet, apiUrl, nil)
        res, err := a.c.Do(r)

        srvHealth := New(
            srv.Addr,
            srv.Name,
            StatusOperational,
            DetailsOperational,
        )

        if err != nil {
            srvHealth.Status = StatusNotRunning
            srvHealth.Details = err.Error()
            responses = append(responses, srvHealth)
            continue
        }
        defer res.Body.Close()

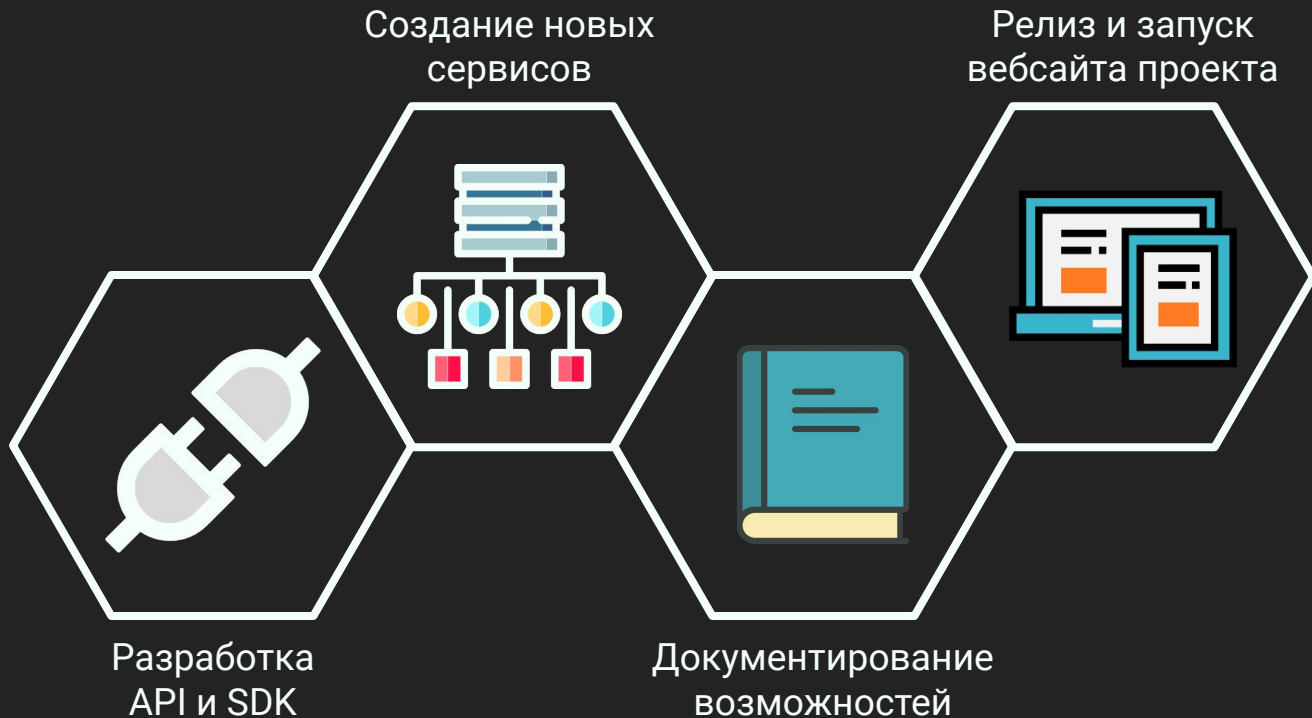
        if res.StatusCode != 200 {
            srvHealth.Status = StatusDegraded
            srvHealth.Details = NoDetailsProvided
        }

        healthResponse := &ServiceResponse{
        if err := json.NewDecoder(res.Body).Decode(healthResponse); err == nil {
            srvHealth.Details = healthResponse.Details
        }
        responses = append(responses, srvHealth)
    }

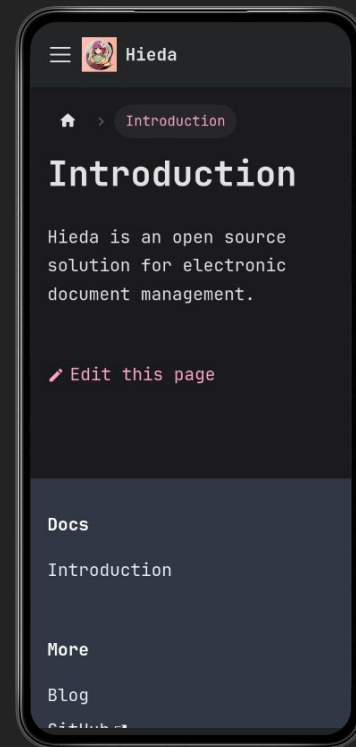
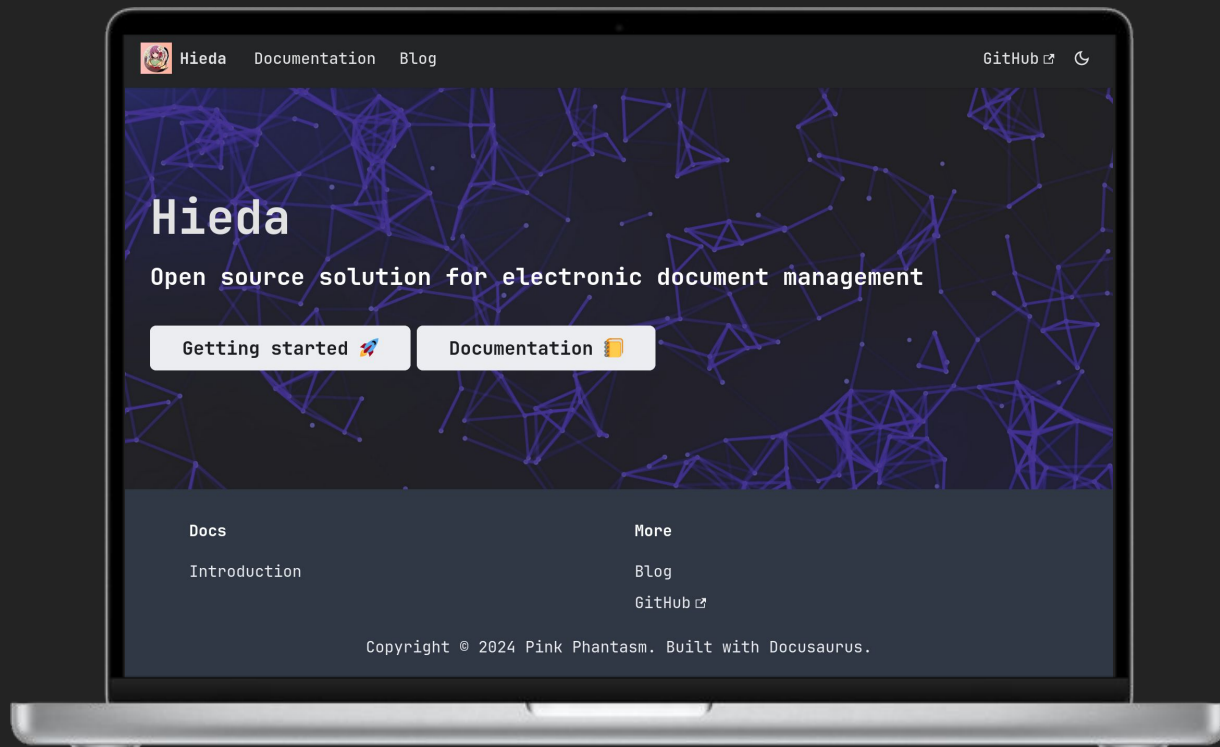
    return responses
}
```



# Планы на будущее



# Сайт + документация





Pink Phantasm



# Hieda

@pinkphantasm



pinkphantasm/hieda