



# NeoFS: Хранение объектных данных по своим правилам

NEO Saint Petersburg Competence Center  
Станислав Богатырёв

SmartDataConf  
Декабрь 2020

[nspcc.ru](https://nspcc.ru)



# О докладчике

И Санкт-Петербургском центре компетенций НЕО

- 20 лет в индустрии
- Участвовал в создании первых в РФ облачных хостингов
- “Эксперт года” в ИТ 2012
- Ранее работал в Samsung Research, EMC, Clodo.ru, Intermedia.NET
- Руководит разработкой NeoFS в NSPCC





# Небольшая команда и большие данные

Сырые и не очень

6-10 ТБ сырых фиксированных данных

Не много, но в один макбук не лезет

5-7 человек исследователей

Можно было бы накормить двумя пиццами, но все в разных местах

Исследовательское многообразие

Разные наборы подготовленных данных из одного массива сырых

Соответствие корпоративным правилам/GDPR/152-ФЗ/CCPS/Data Protection Act/etc

# Наивный облачный подход

По ту сторону блокнотика

Просто

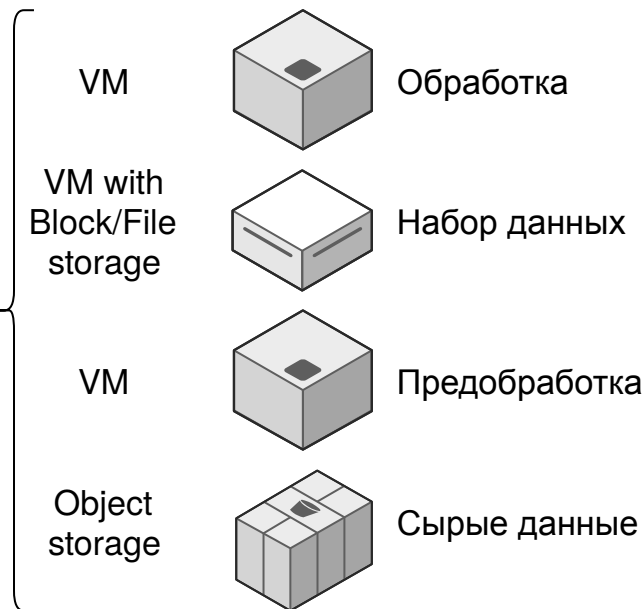
Дорого

Полный Serverless

Все математики



Учёные по данным  
и их  
любимый блокнотик



# Децентрализованный подход

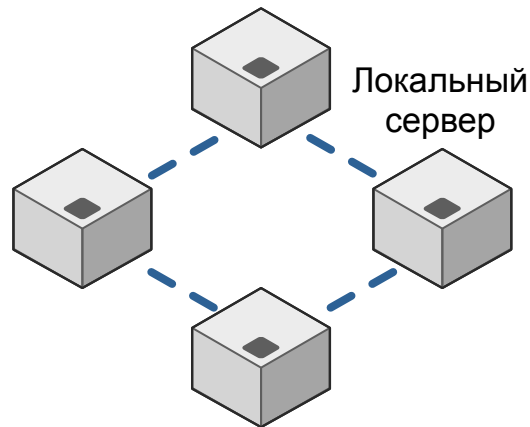
Данные в NeoFS, обработка локальными мощностями

Сложнее

Дешевле

Полный контроль

Есть ответственные



Обработка

Набор данных

Предобработка

Сырые данные



# NeoFS

Объектное хранилище

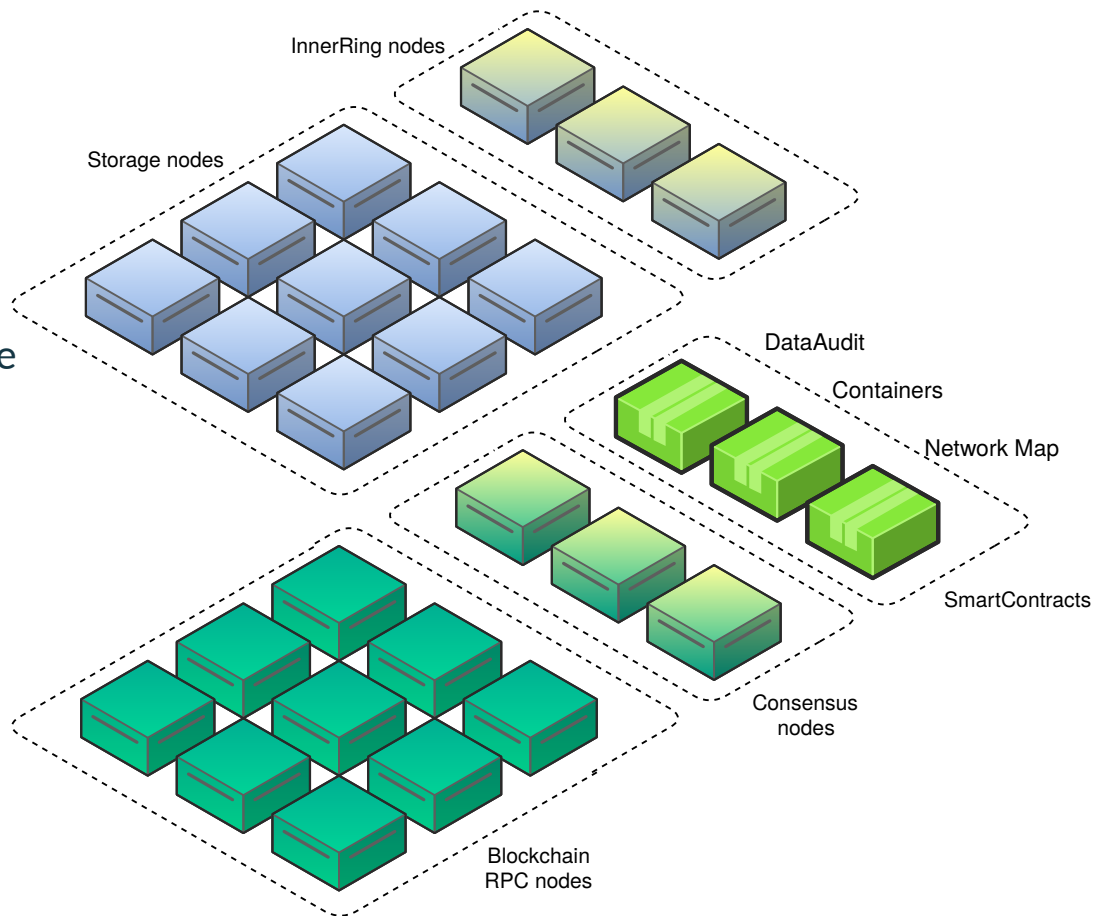
Децентрализованное хранение

Разделение на InnerRing и Storage

Синхронизация через Blockchain

gRPC API, поддержка S3 и HTTP

Политики хранения и ACL





# О чём пойдёт речь

И что мы сегодня узнаем

#зачемздесьблокчейн

Почему объекты хранятся в контейнерах и как правильно их нарезать

Куда сохранить объект и как потом найти

Политики и правила хранения

Аудит данных и авторизация

Что получилось и что будет в будущем



# NeoGo

Blockchain как источник правды

dBFT алгоритм консенсуса

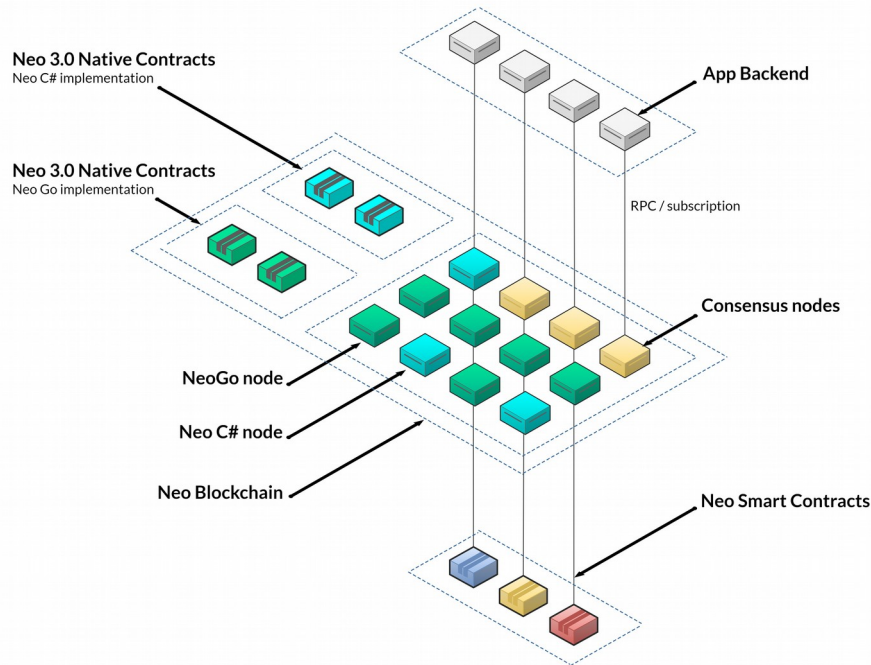
~2000 TPS для 7 CN

Контракты на Go, C#, Python, JavaScript, Java

Свой компилятор Go в NeoVM

API для подписки на события и RPC через WebSocket

Специфичная база данных =)





# Как и где хранятся данные

Контейнеры, объекты, узлы и их атрибуты

Адрес: ContainerID/ObjectID

ContainerID: Hash(Container)

Container: Политика хранения + атрибуты

ObjectID: Hash(Header+Payload)

Object: Данные + атрибуты

Контейнер – аналог директории или bucket

Хранится в блокчейне

Объект – аналог файла

Хранится на Storage узлах сети

Address: 2UVmq...3PcdbSv / 8pw9Ma...HhEag5ds7AbDTF5BT



2Uwm...PcdbSv



8pw9...ds7AbDTF5BT

Container
OwnerID
BasicACL
Attributes
...
Placement Policy

Object
OwnerID
ContainerID
Attributes
...
Payload

# Нарезка объектов

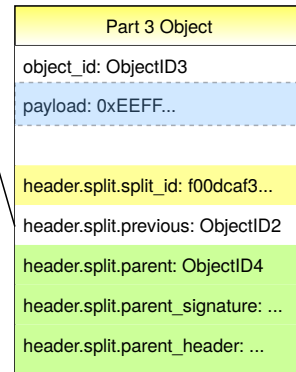
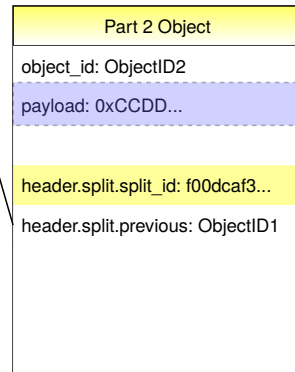
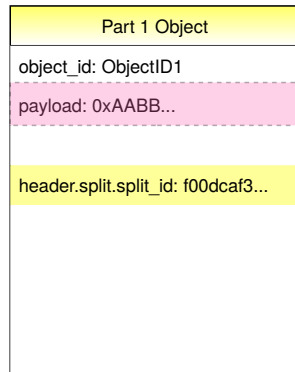
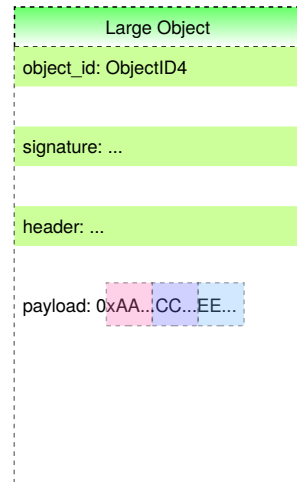
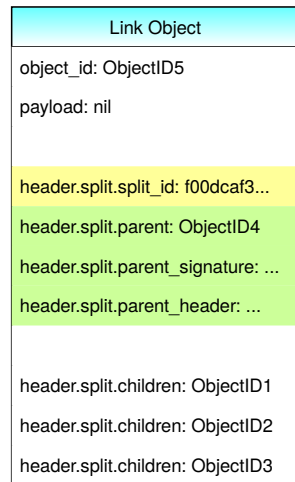
Потоковая колбаса с докачкой

Объект прозрачно делится на части

Доступны как части, так и целое

Потоковая разбивка и сборка  
производятся автоматически

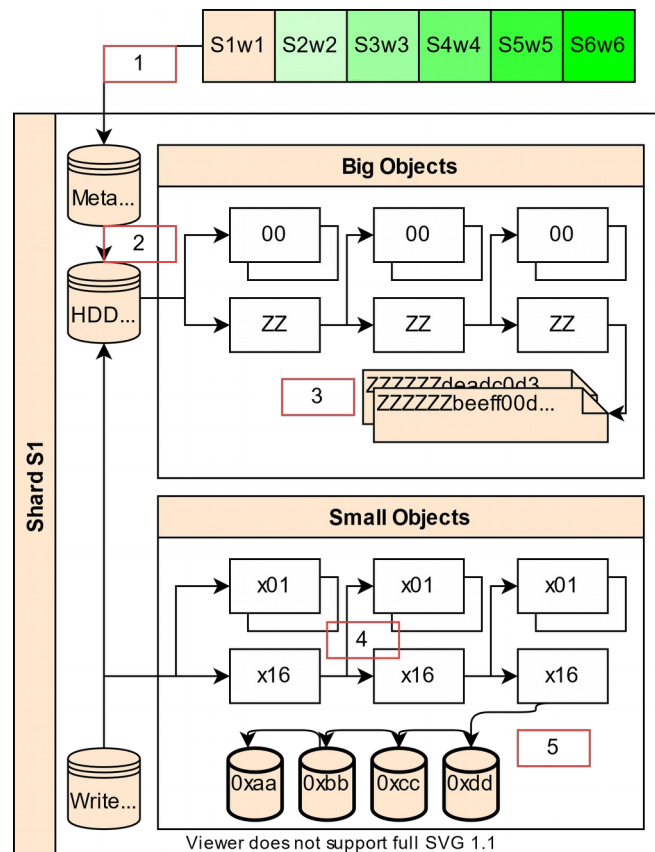
Операционно составной объект  
идентичен единичному



# Движок хранения

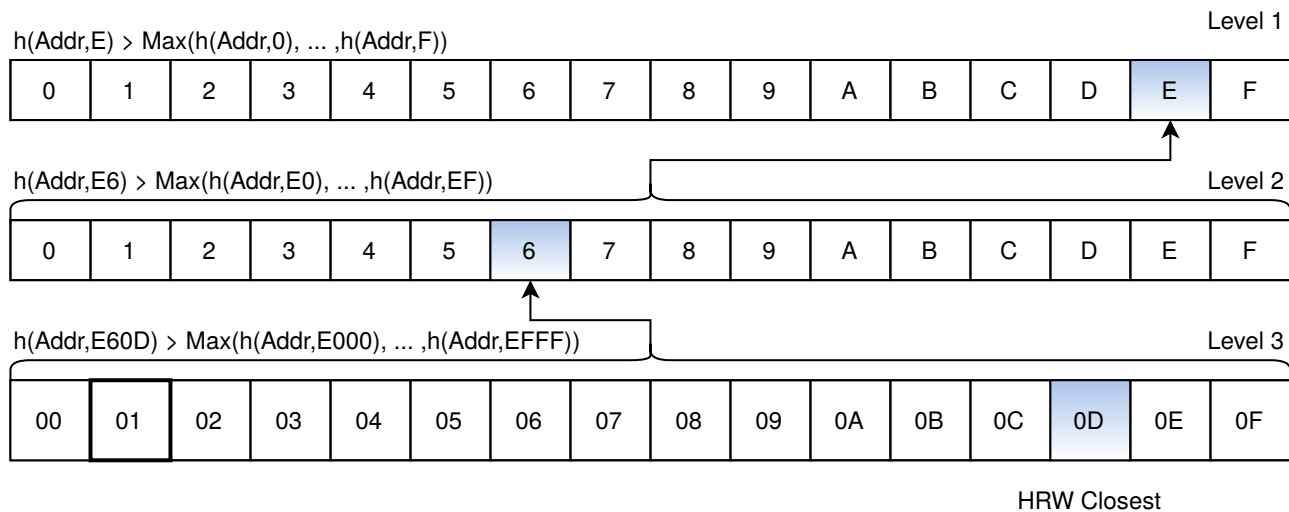
Рандеву с блобовницей

1. Шарды с HRW размещением
2. Кэш локального размещения
3. Размещение по хэшу в shallow dir
4. Размещение по хэшу в блобовницах
5. HRW размещение по блобовницам



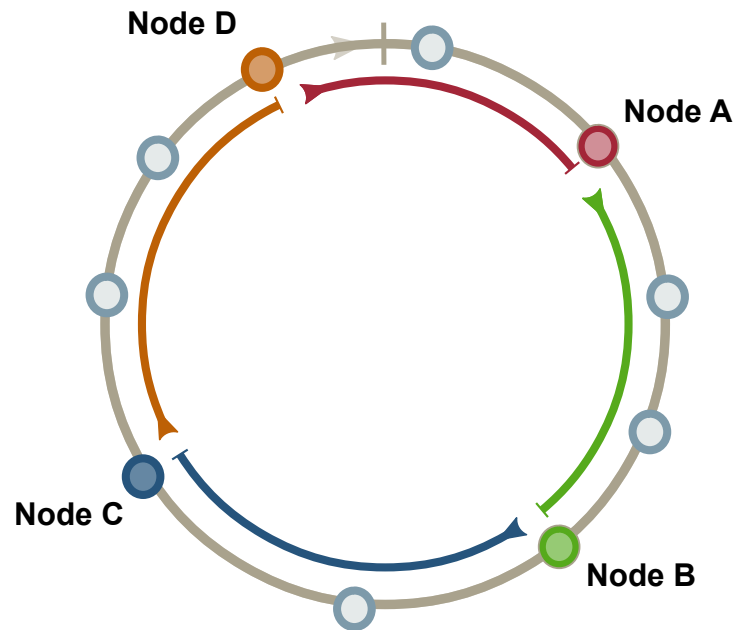
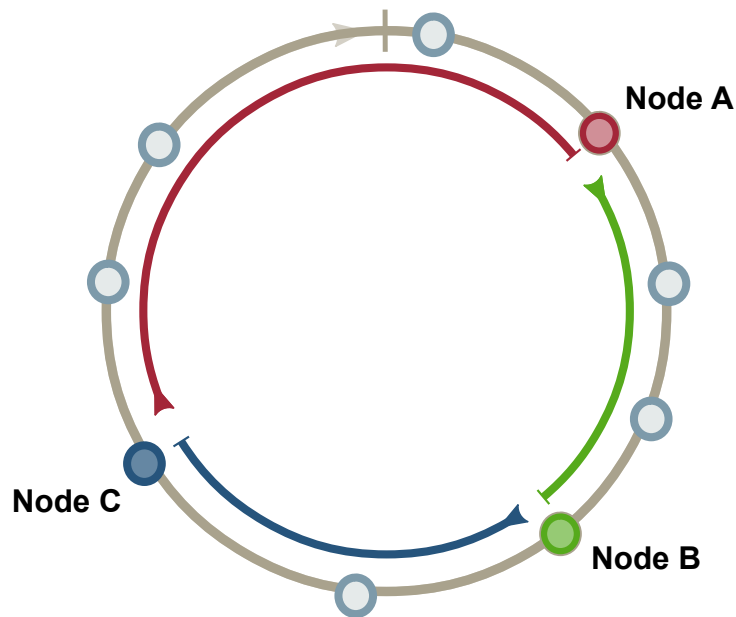
# Rendezvous hashing

Близость по хэш-дистанции



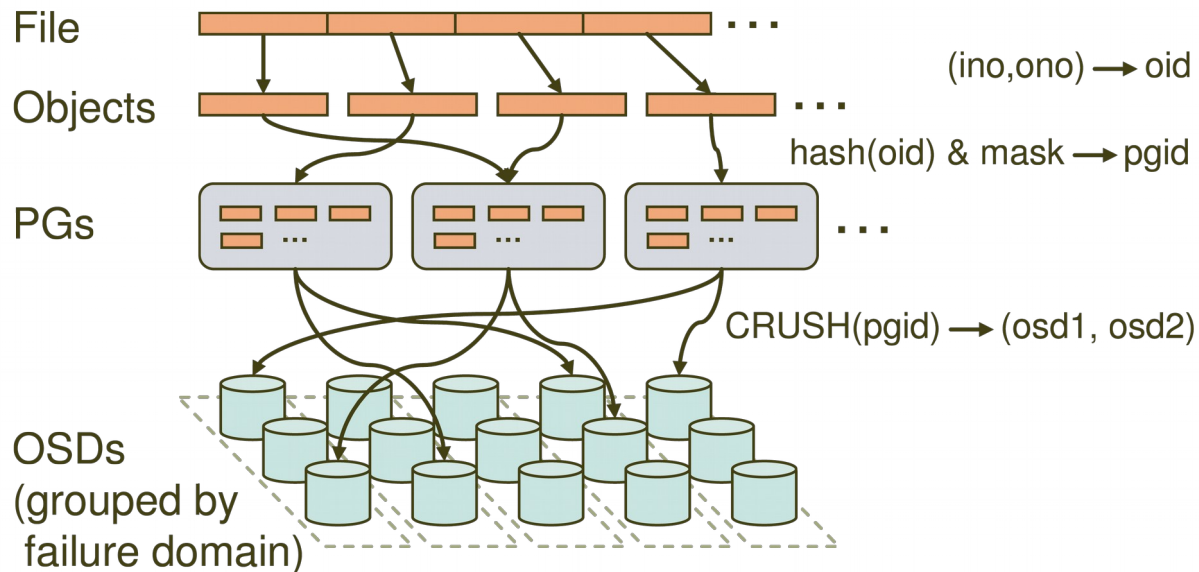
# DHT и HashRing

Балансировка и миграции



# CRUSH в Ceph

Политика на кластер



# Карта сети

CRUSH на весь мир

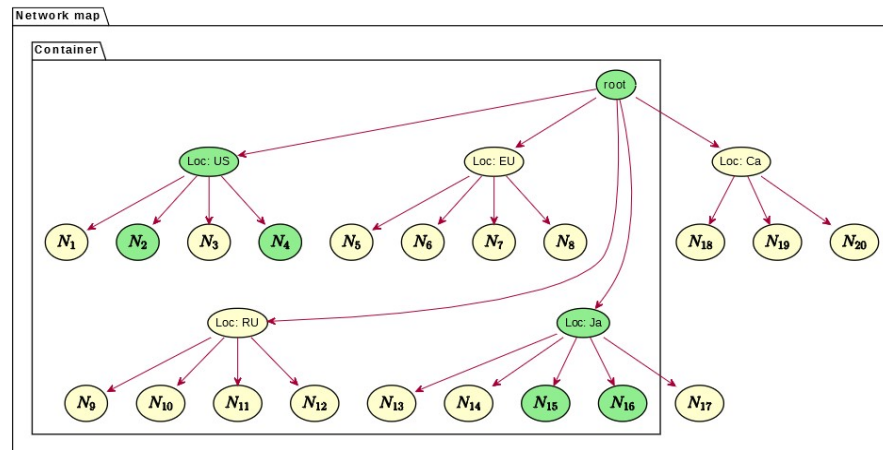
Карта сети содержит активные узлы

Узел сети имеет набор атрибутов

Политика работает с атрибутами по срезам мультиграфа карты сети

HRW заданной глубины для выборки

Утверждается каждую эпоху узлами InnerRing с фиксацией в блокчейне



REP 1 in SPB

REP 2

CBF 4

SELECT 1 Node IN City FROM SPBSSD AS SPB

FILTER SSD EQ true AS IsSSD

FILTER @IsSSD AND Country eq "RU" AND City eq "St.Petersburg" AS SPBSSD



# Политика хранения

## Простые правила

Политика работает с атрибутами по срезам мультиграфа карты сети

REP 1 in CORP

REP 1 in EXT

REP 1 in EMP

Гибкий язык политик с трансляцией в SQL-like, JSON, Blockly

CBF 4

Реплики распределяются по группе узлов

SELECT 4 Node IN CorpSRV as CORP

SELECT 2 Node IN GoodAliens as EXT

SELECT 6 Node IN EmpWKS as EMP

Селекторы определяют группу по фильтрам из карты сети

FILTER CorpProperty EQ ACMECorp AS IsACME

FILTER @IsACME AND NodeType EQ "SRV" AS CorpSRV

FILTER @IsACME AND NodeType EQ "WKS" AS EmpWKS

FILTER Rating GE 5 AS GoodAliens



# Аудит данных

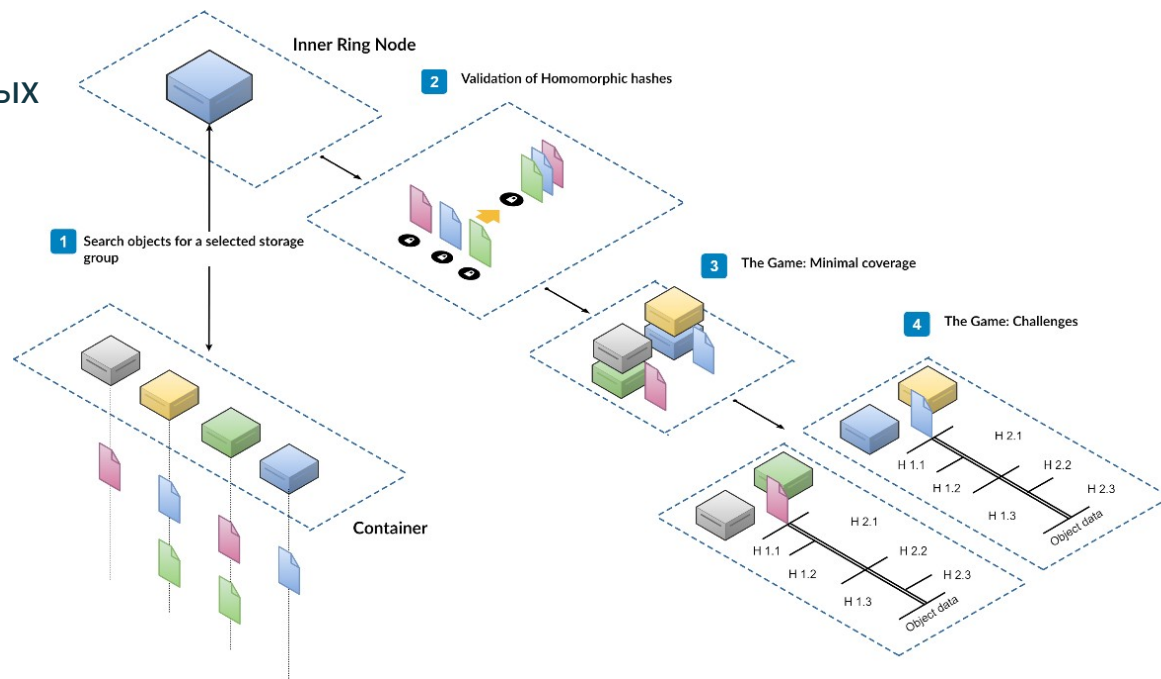
## Многоходовочка

Проверка без раскрытия данных

ZK Proof на гомоморфном  
хэшировании

Работа в недоверенной среде

Сохранение результатов



# Access Control List

## BasicACL

Базовый ACL для каждого контейнера

Неизменяем,  
как и сам контейнер

Можно непротиворечиво  
расширять

0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	3	3	
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
1				2				3				4				5				6				7				8			
0	0	0	1	1	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0
				U	S	O	B	U	S	O	B	U	S	O	B	U	S	O	B	U	S	O	B	U	S	O	B	U	S	O	B
RSRV		X	F	GetRangeHas h				GetRange				Search				Delete				Put				Head				Get			

Похоже на POSIX права доступа



# Access Control List

## ExtendedACL

Можно менять, но нельзя  
противоречить BasicACL

Фиксируется в блокчейне

Фильтры по полям объекта и по  
полям запроса

```
{
  "records": [
    {
      "operation": "GET",
      "action": "DENY",
      "filters": [
        {
          "headerType": "OBJECT",
          "matchType": "STRING_NOT_EQUAL",
          "key": "Classification",
          "value": "Public"
        }
      ],
      "targets": [
        {
          "role": "OTHERS"
        }
      ]
    }
  ]
}
```

# Access Control List

## BearerToken

Можно заместить eACL для  
конкретного запроса

Примерно как JWT в мире веба

Можно ограничить предъявителей

Можно транслировать из HTTP и  
других протоколов

Удобно для сложной авторизации  
(фотобанк, временный доступ,  
игры)

```
{
  "body": {
    "eacTable": {
      "version": {...},
      "containerID": {
        "value": "Ab1X...UeZTz8ZAVtzH5NwfKbC"
      },
      "records": [...],
      "ownerID": null,
      "lifetime": {
        "exp": "100500",
        "nbf": "2",
        "iat": "1"
      }
    },
    "signature": {
      "key": "A/ZjE3Ys1tvs6vVP7wPTxwqHYFigoDhRnoeNBuWtq9Tq",
      "signature": "BCDU1mdfKmo9xVv/8H8XOmSQYN...uvtGNXTuDCWgHuSHZUcwqHtvXjsltBbl4gA0="
    }
  }
}
```

# Что получилось?

Для ответов на вопросы

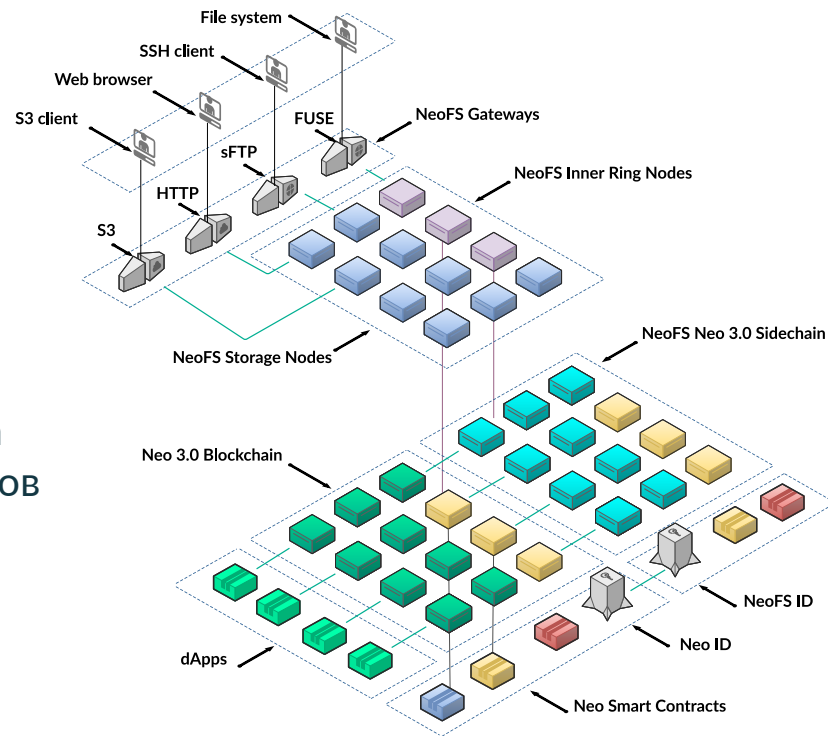
Данные хранятся близко к обработчикам

Узлы на потребительском железе (тоже ServerLess)

Можно работать с привычным ПО на машинах исследователей и инженеров

Можно производить обработку на привычном ПО и фреймворках

Уверенность в соблюдении правил





# Как это работает на самом деле?

Демо!

# Будущее NeoFS

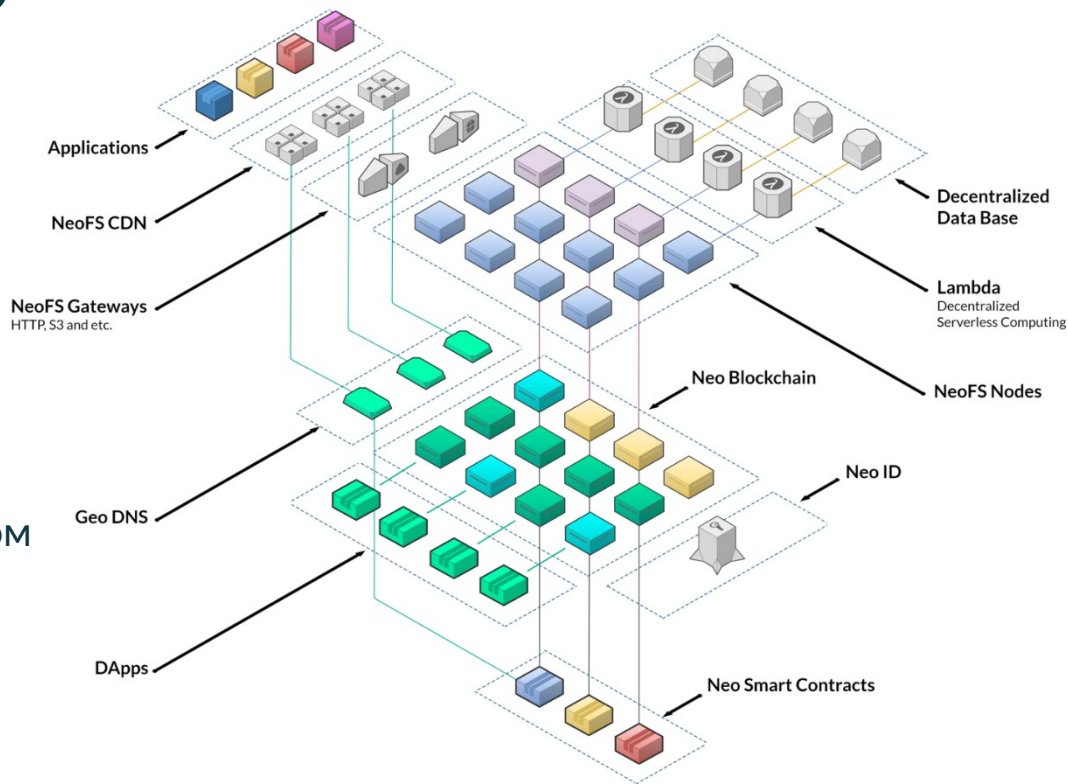
Настоящее облако

NeoFS как хранение

Lambda как Serverless

Децентрализованная БД

Возможность решить любую задачу в децентрализованном подходе с высоким уровнем надежности





# Спасибо!

## Q&A

**E-mail:** *stanislav@nspcc.ru*

**NeoFS.Send (demo)** <https://send.fs.neo.org>

**Neo SPCC:** <https://nspcc.ru>

**NeoFS:** <https://fs.neo.org>

**NeoGo:** <https://github.com/nspcc-dev/neo-go>

**GitHub:** <https://github.com/nspcc-dev/>

**Medium:** <https://medium.com/@neospcc/>



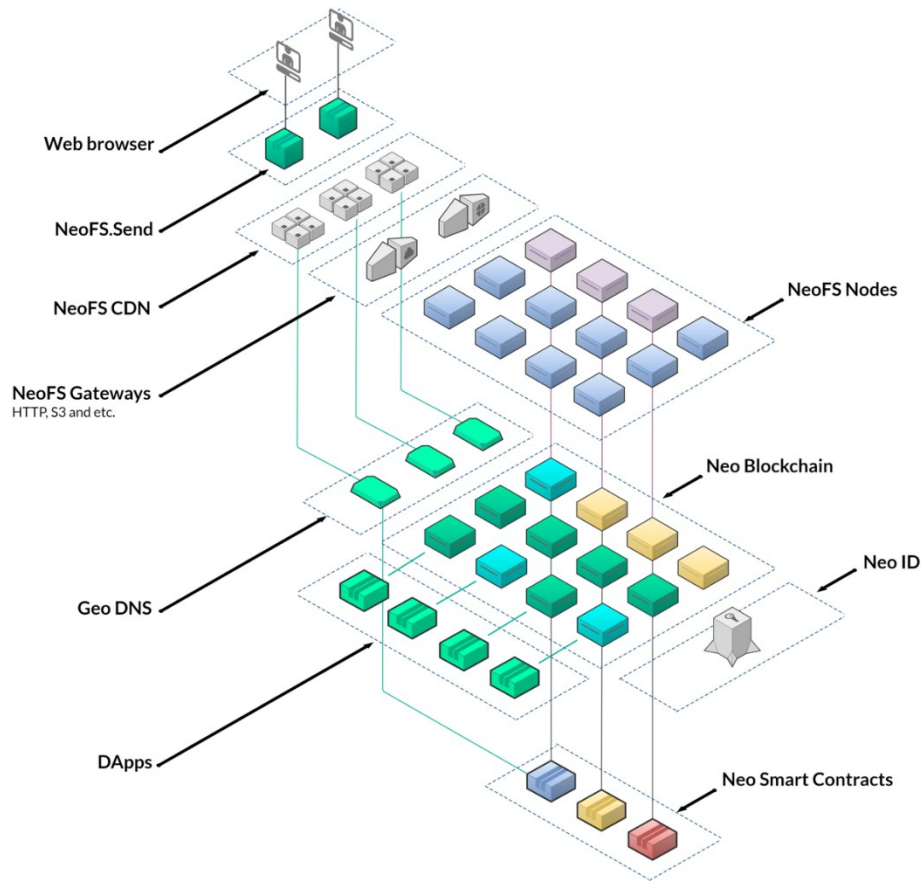
# NeoFS.Send

Пример веб-стека

Полноценное веб-приложение

Простой пример CDN

Гео-привязка по IP клиента



# NeoFS.CDN

GeoDNS + SC

Подключение через контракты

Оптимизация размещения и кеширования

Глубокая интеграция с биллинговой системой

Управление политикой доставки контента из панели управления

