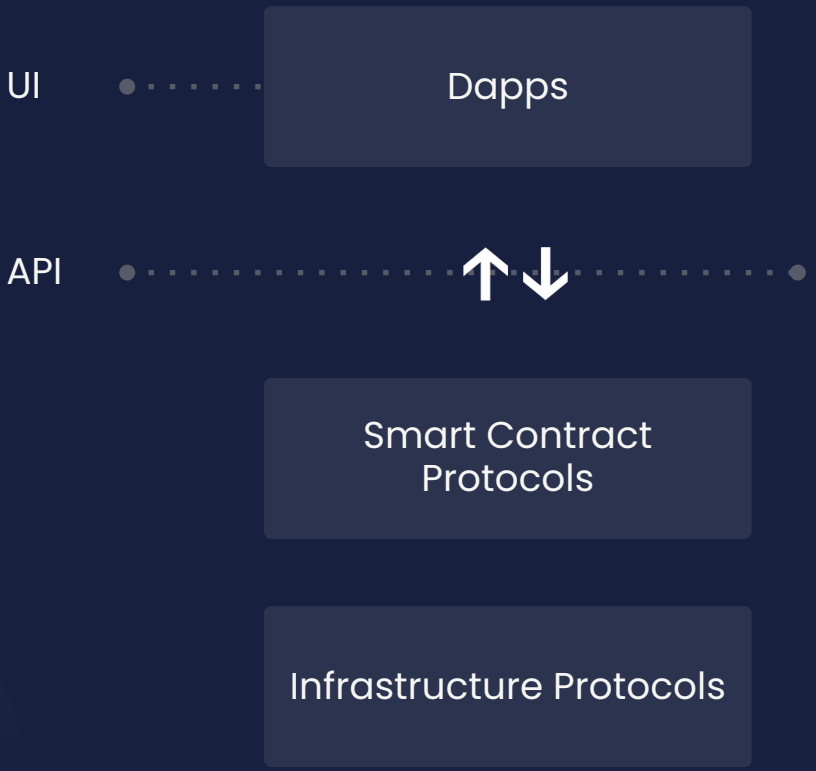
The Graph
如何提高 Dapp 开发效率

什么是Web3?

构建分布式应用的新平台


Web3
无服务器

什么是 Web3?


UI
Dapps
API
Smart Contract Protocols
Infrastructure Protocols

Web3

基础设施

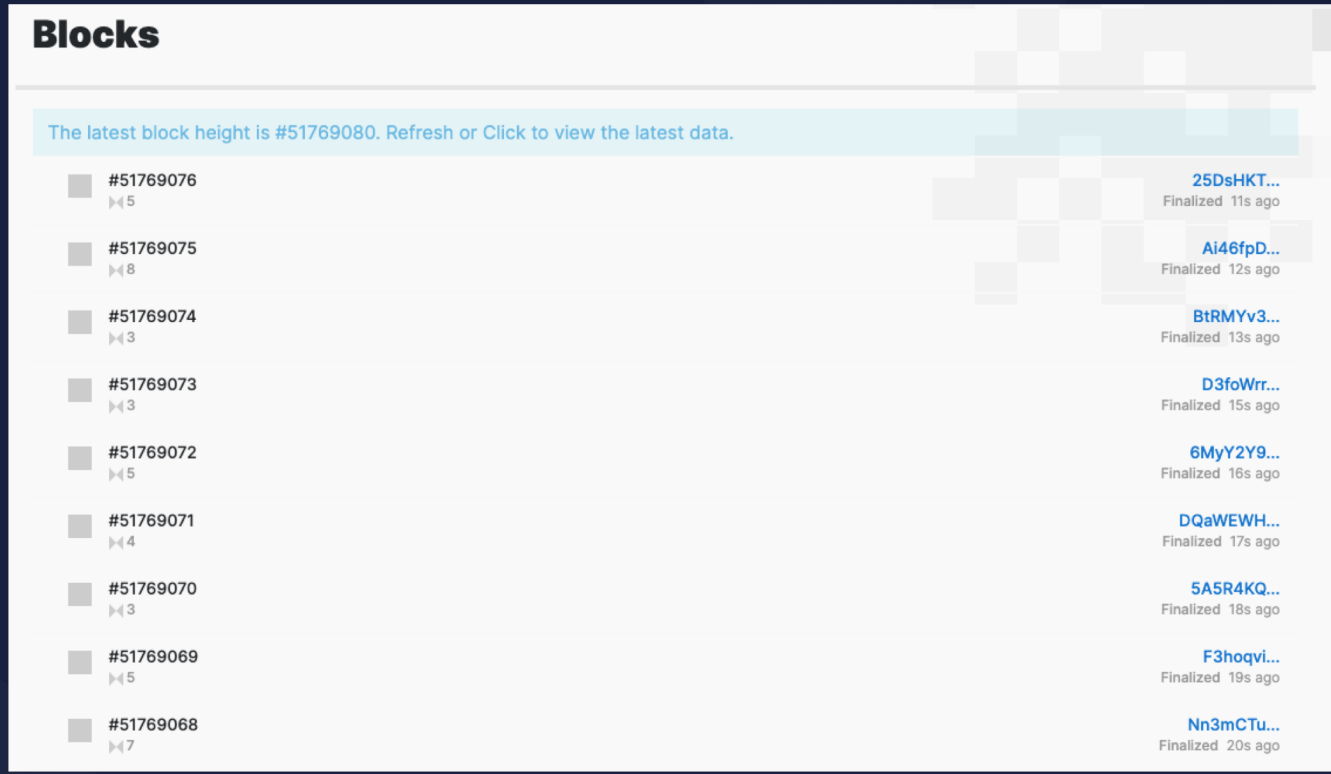
The Graph

Onchain		Offchain	
		Textile	Ceramic
Bridges	L2s	Notifications	IPFS
Ethereum	Other L1s	Storage	Video

Web3

搜索与查询

没有The Graph



Web3

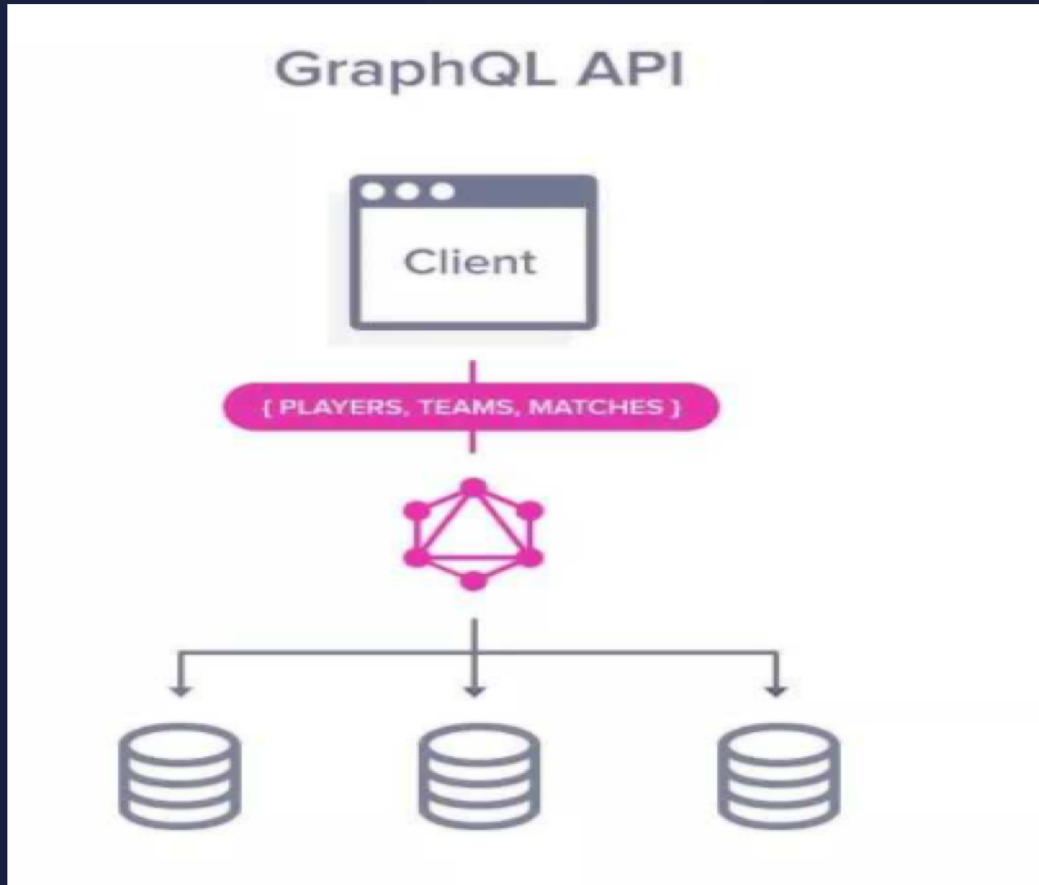
搜索与查询

没有The Graph

Web3

搜索与查询

有了The Graph



Web3

搜索与查询

有了The Graph

```
{
  hero {
    name
  }
}
```

```
{
  "hero": {
    "name": "Luke Skywalker"
  }
}
```

查询架构

Client

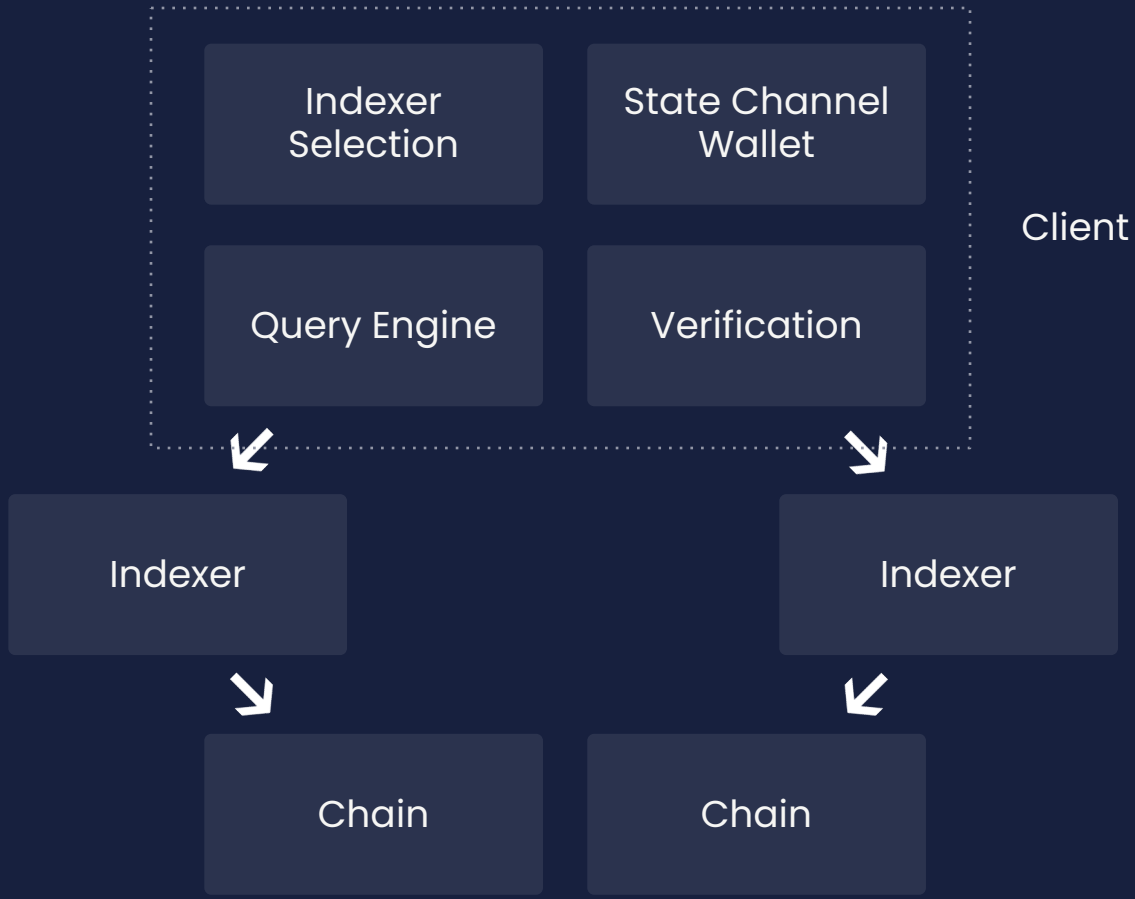
Proof

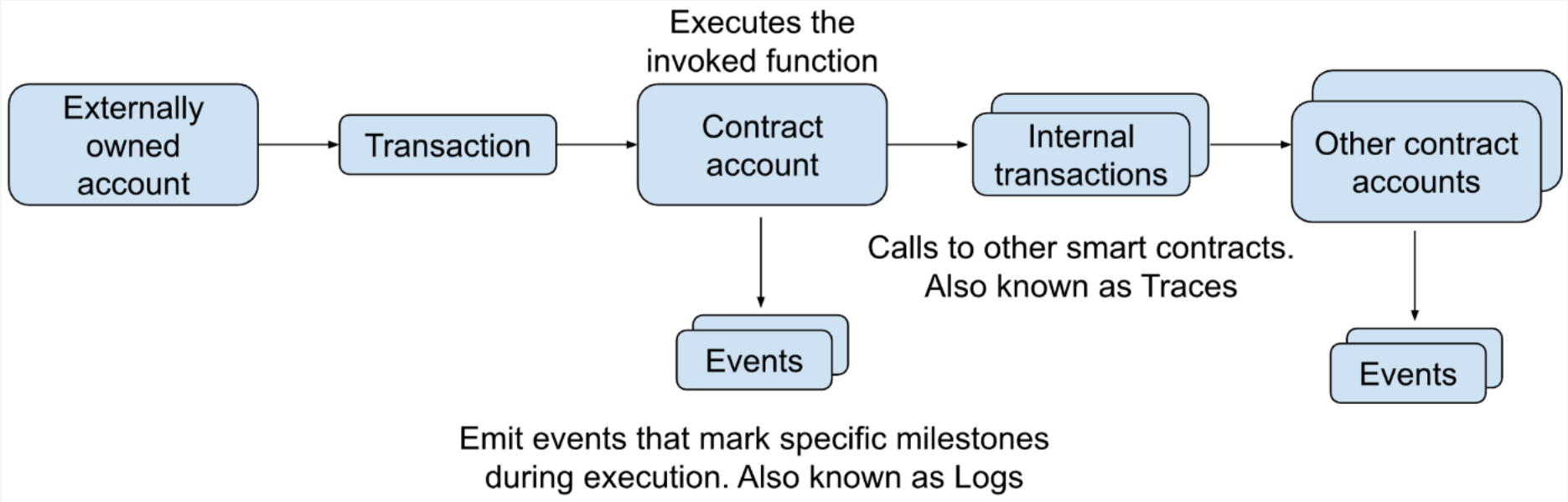
Indexers

Blockchains

Storage Networks

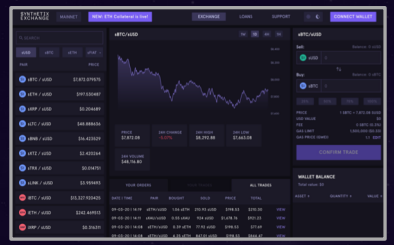
Graph Client



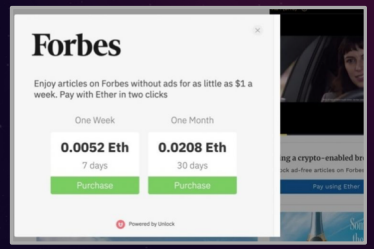

Executes the
invoked function
Externally
owned
account
Transaction
Contract
account
Internal
transactions
Other contract
accounts
Calls to other smart contracts.
Also known as Traces
Events
Events
Emit events that mark specific milestones
during execution. Also known as Logs

支持各种复杂的去中心化应用

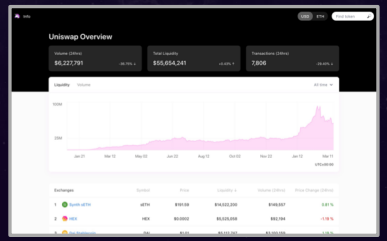
交易数据



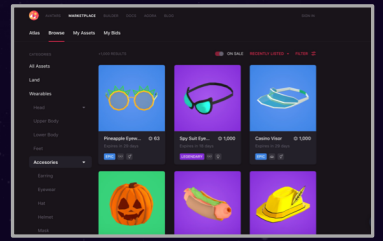
Forbes 订阅数据



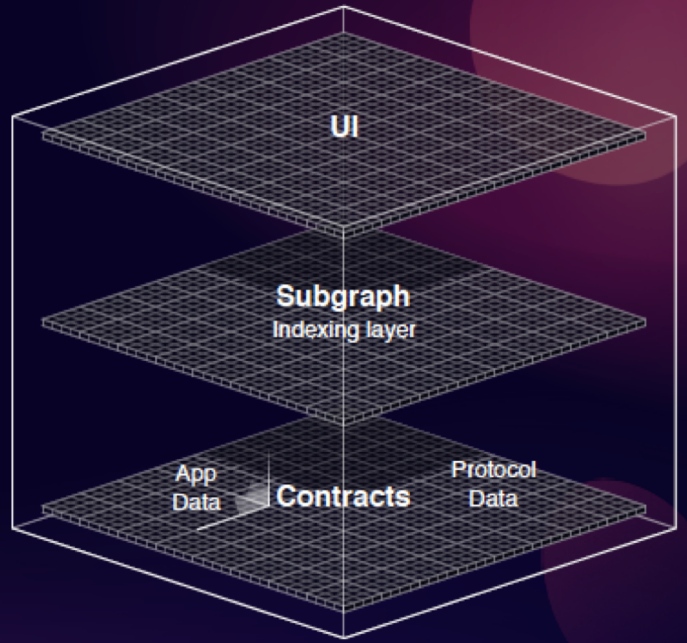
去中心化交易平台



虚拟资产

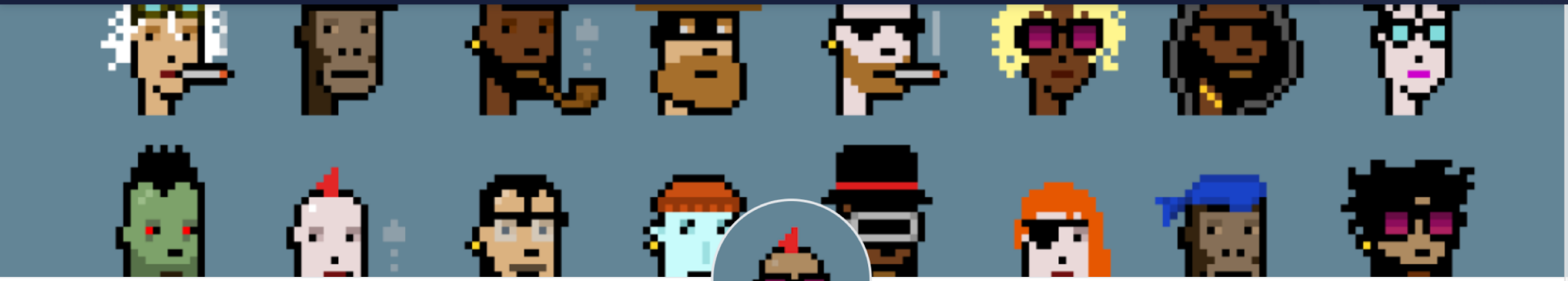


成为web3技术栈不可或缺的基础设施



如何构建Subgraph?

分布式应用



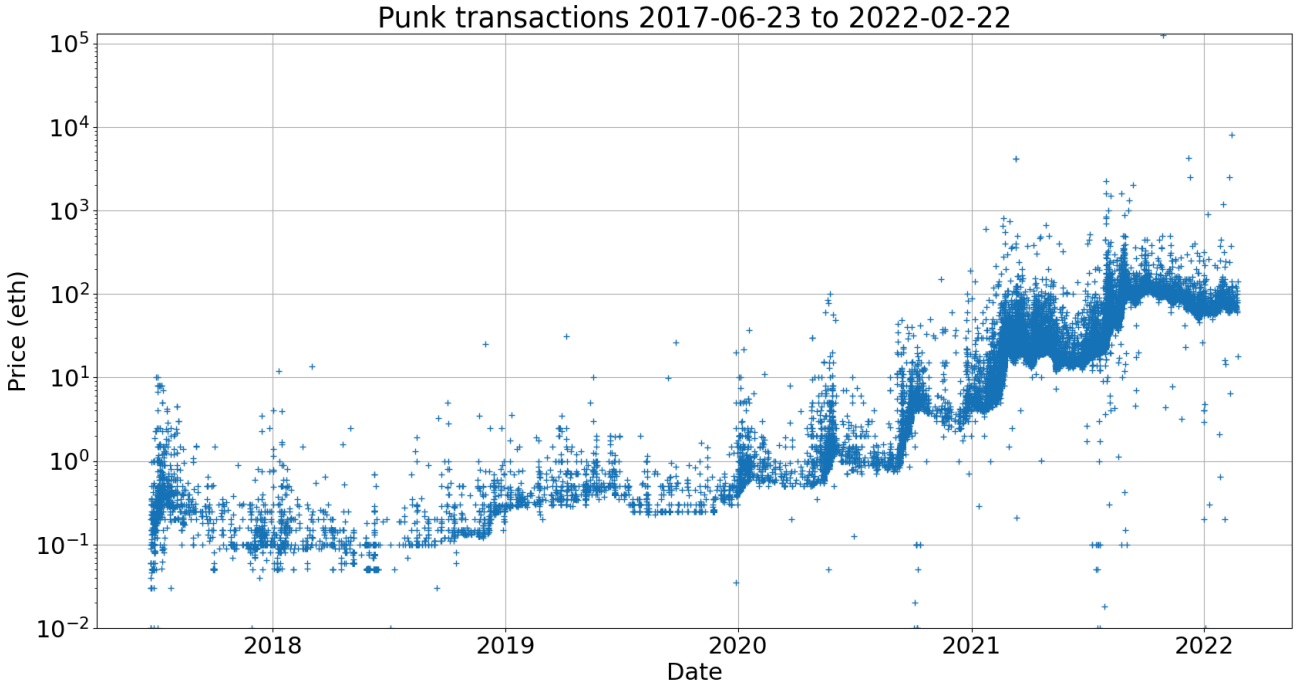
+ Add to watchlist

CryptoPunks

10.0K	3.4K	---	845.0K
items	owners	floor price	volume traded

如何构建Subgraph?

分布式应用



如何构建Subgraph?

工具

@graphprotocol/graph-cli

@graphprotocol/graph-ts

如何构建Subgraph?

命令

$ graph codegen

$ graph build

如何构建Subgraph?

代码

subgraph.yaml

schema.graphql

AssemblyScript映射

subgraph.yaml

```yaml
specVersion: 0.0.2
schema:
  file: ./schema.graphql
#graft:
#  base: Qmf9kah4dfYmaEBrRiqXEuHXMGRUCKNHcytYSasCygAF9K
#  block: 10821737
dataSources:
  - kind: ethereum/contract
    name: cryptopunks
    network: mainnet
    source:
      address: "0xb47e3cd837dDF8e4c57F05d70Ab865de6e193BBB"
      abi: cryptopunks
      startBlock: 3914495
    mapping:
      kind: ethereum/events
      apiVersion: 0.0.5
      language: wasm/assemblyscript
      entities:
        - Account
        - Punk
        - NFT
        - Transfer
        - MetaData
        - Trait
        - WrappedPunk
        - Assign
        - Offer
        - Transfer
        - Sale
      abis:
        - name: cryptopunks
          file: ./abis/cryptopunks.json
        - name: CryptopunksMetadata
          file: ./abis/CryptopunksMetadata.json
```

subgraph.yaml

```
36        eventHandlers:
37          - event: Assign(indexed address,uint256)
38            handler: handleAssign
39          - event: PunkTransfer(indexed address,indexed address,uint256)
40            handler: handlePunkTransfer
41          - event: PunkOffered(indexed uint256,uint256,indexed address)
42            handler: handlePunkOffered
43          - event: PunkBidEntered(indexed uint256,uint256,indexed address)
44            handler: handlePunkBidEntered
45          - event: PunkBidWithdrawn(indexed uint256,uint256,indexed address)
46            handler: handlePunkBidWithdrawn
47          - event: PunkBought(indexed uint256,uint256,indexed address,indexed address)
48            handler: handlePunkBought
49          - event: PunkNoLongerForSale(indexed uint256)
50            handler: handlePunkNoLongerForSale
51        file: ./src/mapping.ts
```

Schema

```
type Account @entity {
  "Ethereum Address"
  id: ID!

  "All Punks owned by Account"
  punksOwned: [Punk!] @derivedFrom(field: "owner")

  "Purchases by Account"
  bought: [Sale!]! @derivedFrom(field: "to")

  "All Punks owned by Account"
  nftsOwned: [NFT!]! @derivedFrom(field: "owner")

  "Punks assigned to account (if any)"
  assigned: [Assign!]! @derivedFrom(field: "to")

  "Punk transfer by Account"
  sent: [Transfer!]! @derivedFrom(field: "from")

  received: [Transfer!]! @derivedFrom(field: "to")

  "Query bids to Account or by Account"
  bids: [Bid!]! @derivedFrom(field: "from")

  "Punks offered for sale by Account"
  asks: [Ask!]! @derivedFrom(field: "from")

  numberOfPunksOwned: BigInt!
}
```

映射文件

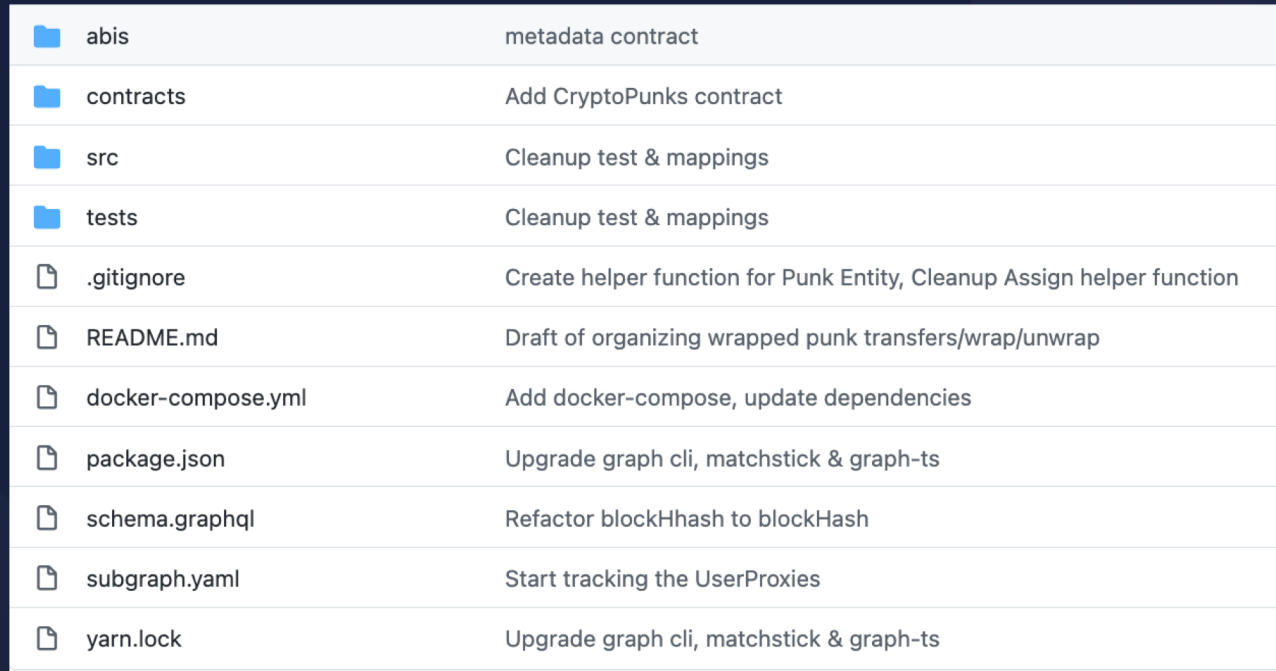
```
export function handlePunkBought(event: PunkBought): void {
  log.debug("handlePunkBought", []);

  let sale = getOrCreateSale(
    event.params.toAddress,
    event.params.fromAddress,
    event.params.punkIndex,
    event
  );
  let punk = Punk.load(event.params.punkIndex.toString())!;
  let contract = getOrCreateCryptoPunkContract(event.address);
  let toAccount = getOrCreateAccount(event.params.toAddress);
  let fromAccount = getOrCreateAccount(event.params.fromAddress);

  sale.amount = event.params.value;

  contract.totalAmountTraded = contract.totalAmountTraded.plus(
    event.params.value
  );
  contract.totalSales = contract.totalSales.plus(BigInt.fromI32(1));
```

映射文件

```
  // Note: buyPunk() does not emit a PunkTransfer event, so we need to keep track
  toAccount.numberOfPunksOwned = toAccount.numberOfPunksOwned.plus(
    BigInt.fromI32(1)
  );

  fromAccount.numberOfPunksOwned = fromAccount.numberOfPunksOwned.minus(
    BigInt.fromI32(1)
  );

  punk.purchasedBy = toAccount.id;
  punk.owner = toAccount.id;

  punk.save();
  fromAccount.save();
  toAccount.save();
  contract.save();
  sale.save();
}
```

完整代码

https://github.com/itsjerryokolo/CryptoPunks

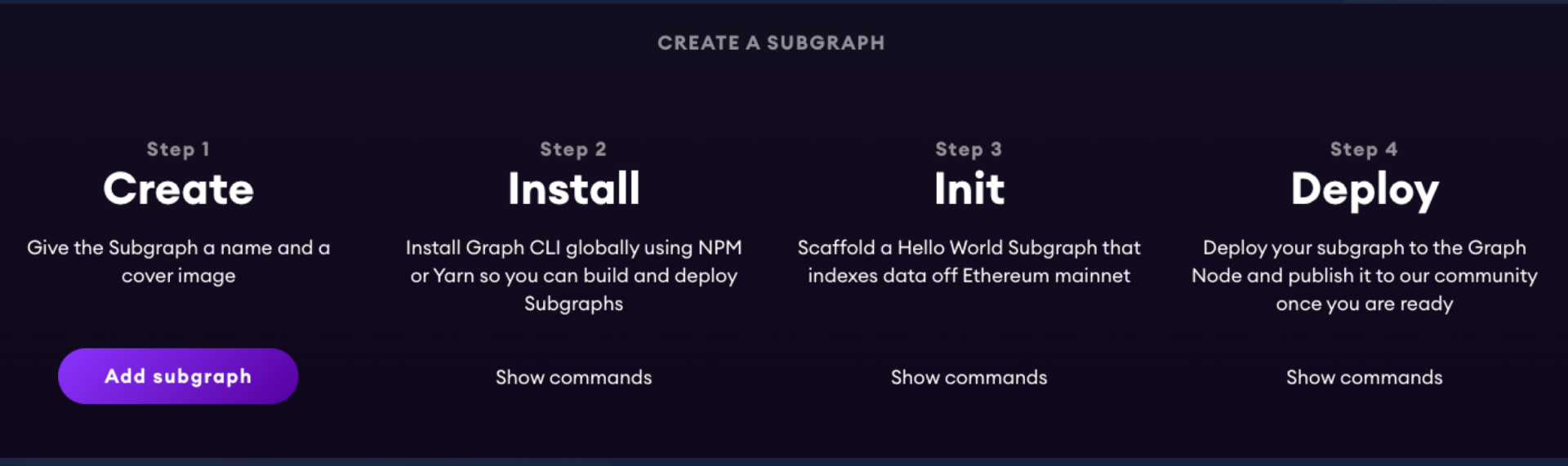


abis	metadata contract
contracts	Add CryptoPunks contract
src	Cleanup test & mappings
tests	Cleanup test & mappings
.gitignore	Create helper function for Punk Entity, Cleanup Assign helper function
README.md	Draft of organizing wrapped punk transfers/wrap/unwrap
docker-compose.yml	Add docker-compose, update dependencies
package.json	Upgrade graph cli, matchstick & graph-ts
schema.graphql	Refactor blockHhash to blockHash
subgraph.yaml	Start tracking the UserProxies
yarn.lock	Upgrade graph cli, matchstick & graph-ts

如何在Near上构建Subgraph?

部署

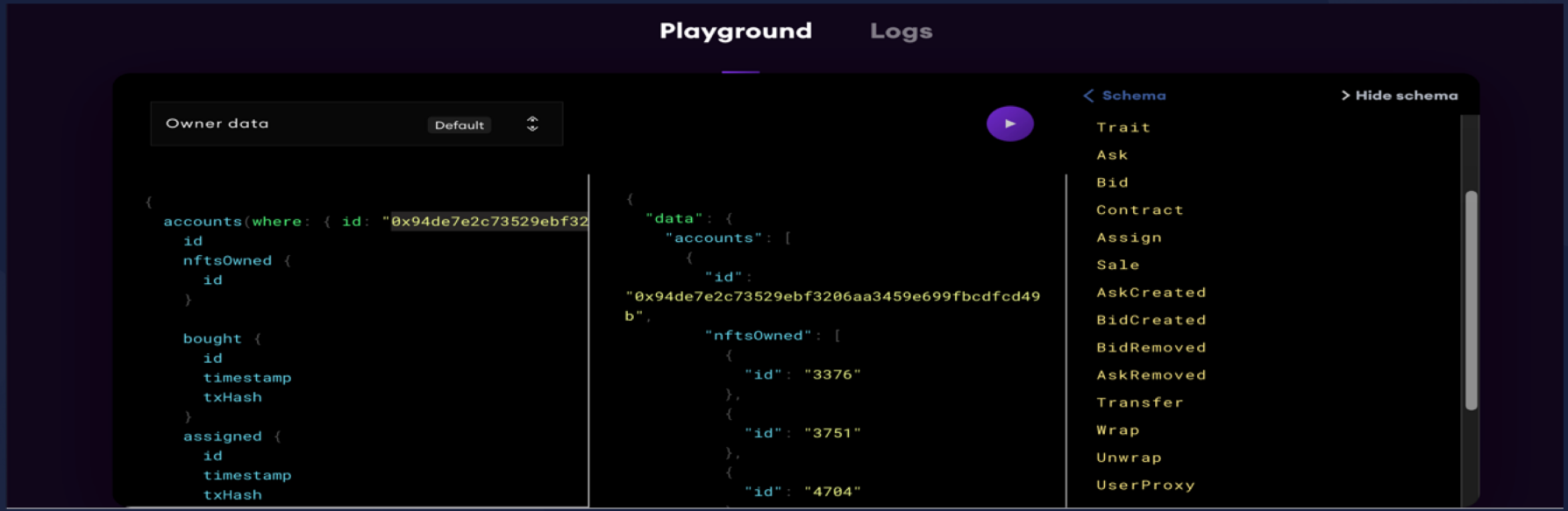
https://thegraph.com/hosted-service/dashboard



如何查询Subgraph?

查询

https://thegraph.com/hosted-service/subgraph/itsjerryokolo/cryptopunks



更多信息

https://thegraph.com

微信公众号

graphprotocol中文

Twitter

@TheGraphCN

Telegram

https://t.me/GraphProtocolCN