



3. NEAR dapp开发常用工具

Daniel



何为dapp

1. **decentralized application**
2. 运行在去中心化网络上面的应用程序
3. 智能合约
4. 但是通常由后端合约、前端界面、存储等共同构成



如何与centralized app交互



User

User Interface



App

HTTP/GraphQL



Web Server



如何与decentralized app交互



User

User Interface



App

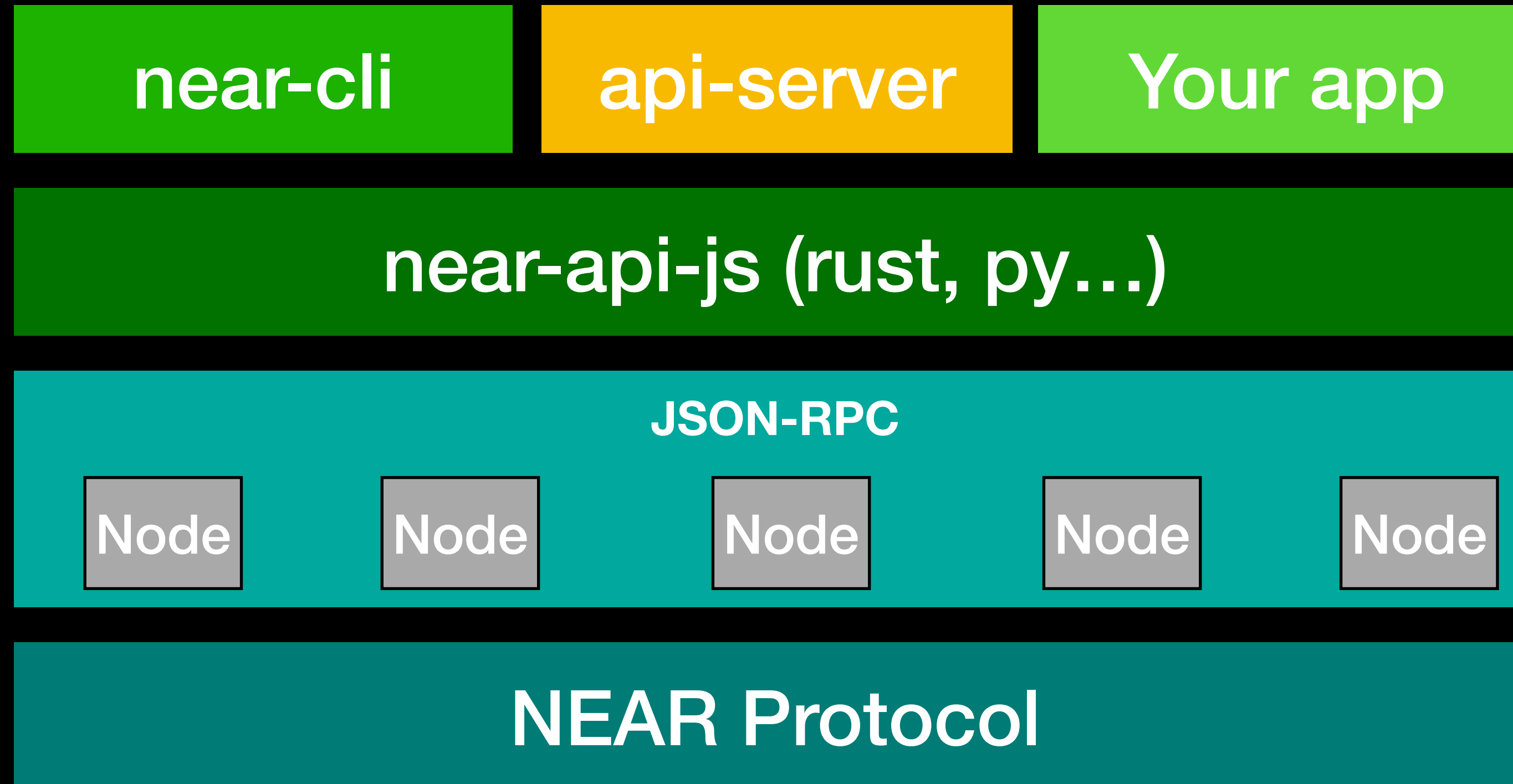
JSON-RPC



NEAR chain



如何与decentralized app交互





JSON-RPC

- <https://docs.near.org/docs/api/rpc>
- 所有方法均用POST
- JSON RPC 2.0
- 接入点
 - Mainnet: <https://rpc.mainnet.near.org>
 - Testnet: <https://rpc.testnet.near.org>



JSON-RPC

```
{
  "jsonrpc": "2.0",
  "id": "dontcare",
  "method": "query",
  "params": {
    "request_type": "view_account",
    "finality": "final",
    "account_id": "nearkat.testnet"
  }
}
```

```
{
  "jsonrpc": "2.0",
  "result": {
    "amount": "399992611103597728750000000",
    "locked": "0",
    "code_hash": "111111111111111111111111111111111111",
    "storage_usage": 642,
    "storage_paid_at": 0,
    "block_height": 17795474,
    "block_hash": "9MjpcnwW3TSdzGweNfPbkx8M74q1XzUcT1P/"
  },
  "id": "dontcare"
}
```



Archival RPC Node

- 超过 5 epochs 或 ~2.5 days
- 接入点
 - Mainnet: <https://archival-rpc.mainnet.near.org>
 - Testnet: <https://archival-rpc.testnet.near.org>



near-api-js

- 对JSON-RPC接口的封装
- 提供了wallet、account、contract等概念抽象
- 适配多种使用环境（浏览器、命令行、服务器等）
- <https://github.com/near/near-api-js>



创建账号

```
// creates a new account using funds from the account used to create it  
const near = await connect(config);  
const account = await near.account("example-account.testnet");  
await account.createAccount(  
  "example-account2.testnet", // new account name  
  "8hSHprDq2StXwMtNd43wDTXQYsjXcD4MJTXQYsjXcc", // public key for new account  
  "100000000000000000000000" // initial balance for new account in yoctoNEAR  
);
```



合约调用

```
await contract.method_name(  
  {  
    arg_name: "value", // argument name and value - pass empty object if no args required  
  },  
  3000000000000000, // attached GAS (optional)  
  1000000000000000000000000000000 // attached deposit in yoctoNEAR (optional)  
);
```



单位转换

```
// converts yoctoNEAR (10^-24) amount into NEAR  
  
const { utils } = nearAPI;  
const amountInNEAR = utils.format.formatNearAmount("1000000000000000000000000000000");
```



near-cli

- 提供了一个命令行工具，用于和NEAR交互
- 使用node.js编写，基于near-api-js
- <https://github.com/near/near-cli>



near-cli

- 账户

- 创建账户
- 查看状态
- 转账
- ...

- 授权

- 登录
- 添加、删除访问密钥
- ...

- 合约

- 部署
- 调用
- ...



near-api-server

- 基于Node.js实现的用于和NEAR交互的API服务
- 对外提供HTTP API
- 合约部署、调用
- 账户创建、查询
- NFT操作
- <https://github.com/zavodil/near-api-server>



near-api-server

Route	Method	Description
CONTRACTS		
<code>/deploy</code>	POST	Deploys a smart contract on NEAR.
<code>/view</code>	POST	Performs a smart contract view call with no gas burnt.
<code>/call</code>	POST	Performs a smart contract change call that burns gas.
UTILS		
<code>/init</code>	POST	Initializes the master account and updates <code>near-api-server-config.json</code>
<code>/create_user</code>	POST	Creates a NEAR sub-account and stores credentials in <code>/storage</code> .
<code>/parse_seed_phrase</code>	POST	Displays public and private key pair from a given seed phrase.
<code>/balance</code>	GET	Displays account balance.
NFT EXAMPLE		
<code>/mint_nft</code>	POST	Mints an NFT for a given contract.
<code>/transfer_nft</code>	POST	Transfers NFT ownership to a specified account.
<code>/view_nft</code>	POST	Returns owner, metadata, and approved account IDs for a given token ID.



near-api-js demo

- Place (<https://near-examples.github.io/place/>)
- 浏览器中如何初始化账户、合约等对象
- 如何引导用户登录NEAR账户
- 如何调用合约



NEAR Wallet

NEAR

钱包 质押 账户 帮助

hihihiere1.testnet

余额

收藏品

余额

\$875.99

可用余额: \$832.18

发送

接收

购买

Your Portfolio

Token Balance

NEAR

\$4.38

199.99741 NEAR

≈ \$875.99 USD

最近动态

<>

方法已调用

合约 moonlet.pool.f863973.m0 ...

2min

🔑

访问密钥已添加

至 hihihiere1.testnet

5d

👤

新账户已创建

账户 hihihiere1.testnet

5d

↩

已接收 NEAR

来自 testnet

+200 NEAR

5d

🔑

访问密钥已添加

至 hihihiere1.testnet

5d

查看全部



Indexer

- 链上数据原本是流式的 (Stream-based)
- 如果需要查询某一个特定账户当前状态，或者某个特定区块的信息是相对很简单的
- 但是如果查询跨多个区块的、或者历史的信息开销就会很大
- Indexer是一个工具，它监听各种链上事件并将其记录在数据库中，便于后续快速查询



下一步

- 尝试自己部署并调用一个合约
- 写一个简单的UI与其进行交互
- 参加NCD课程的demo以及NEAR黑客松