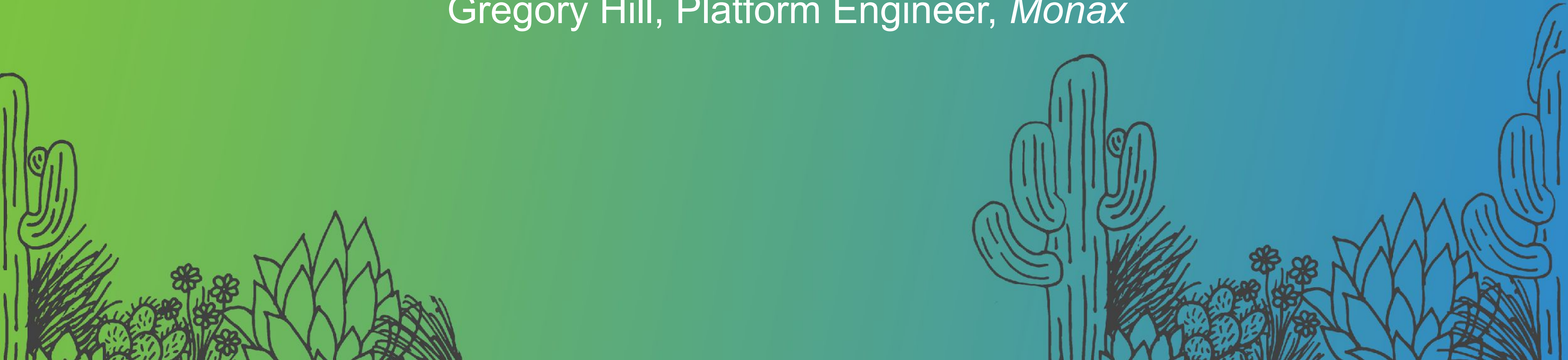
HYPERLEDGER GLOBAL FORUM

PHOENIX, AZ | MARCH 3-6, 2020

HYPERLEDGER BURROW

Gregory Hill, Platform Engineer, *Monax*



Getting Started

https://github.com/hyperledger/burrow/releases/latest

```
~$ burrow --version
~$ burrow spec -v1 > spec.json
~$ burrow configure -s spec.json > config.toml
~$ burrow start -c config.toml
```

Digital Signatures

- Ed25519 [Tendermint]
- Secp256k1 [Bitcoin / Ethereum]

```
~$ burrow keys server &
~$ burrow keys gen -t secp256k1
~$ pkill burrow
```

Genesis

```
[GenesisDoc]
  GenesisTime = 2020-02-23T16:26:58Z
  ChainName = "BurrowChain_365586"

  [[GenesisDoc.Accounts]]
    Address = "*****"
    PublicKey = "*****"
    Amount = 9999999999
    Name = "Validator_0"

    [GenesisDoc.Accounts.Permissions]
      [GenesisDoc.Accounts.Permissions.Base]
        Perms = "bond"
        SetBit = "bond"

  [[GenesisDoc.Validators]]
    Address = "*****"
    PublicKey = "*****"
    Amount = 9999999998
    Name = "Validator_0"
```

Name (Entropy - Genesis Hash)

Account:

- Initial Token
- Default Permissions

Validator (>= 1)

Contracts

- EVM (Solidity & Vyper), eWASM
- Tooling:
  - Deploy:
    - solc
    - solang
  - CLI (tx)
- Web3 (Remix, Truffle)
- Burrow JS / TS

```
~$ cat config.toml

      [RPC.GRPC]
         Enabled = true
         ListenHost = "0.0.0.0"
         ListenPort = "10997"

      [RPC.Web3]
         Enabled = true
         ListenHost = "0.0.0.0"
         ListenPort = "26660"
```

Truffle (1)

```
~$ cd erc20
~$ npm install
~$ npm install -g truffle
```

contracts/ERC20.sol

```
pragma solidity ^0.5.0;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract Mintable is ERC20 {
      function mint(address account, uint256 amount) public {
            _mint(account, amount);
      }
}
```

Truffle (2)

```
const Mintable = artifacts.require("Mintable");

contract("Mintable", accounts => {
      it("Should deploy and mint token", async () => {
            const owner = accounts[0];
            const amount = 500;
            const instance = await Mintable.deployed();
            await instance.mint(owner, amount);
            const result = await instance.balanceOf(owner);
            assert(result.toNumber() == amount);
      });
});
```

test/mintable.test.js

```
var Mintable = artifacts.require("Mintable");

module.exports = function(deployer) {
      deployer.deploy(Mintable);
};
```

migrations/2_deploy_contract.js

HYPERLEDGER GLOBAL FORUM PHOENIX, AZ | MARCH 3-6, 2020

Truffle (3)

```
~$ burrow spec -v1 -d2 > spec.json

~$ burrow configure -s spec.json
        --curve-type secp256k1 > burrow.toml

~$ burrow start -c burrow.toml



~$ truffle test --network <id>
```

```
module.exports = {
      networks: {
            development: {
                  host: "127.0.0.1",
                  port: 26660,
                  network_id: "*",
            },
      }
};
```

truffle-config.js

Burrow (1)

```
~$ cd erc721
~$ npm install
~$ npm install -g ts-node
```

```
pragma solidity ^0.5.0;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";

contract Mintable is ERC721 {
	function mint(address account, uint256 id) public {
		_mint(account, id);
	}
}
```

src/contracts/ERC721.sol

Burrow (2)

```
~$ npm run build
```

src/deploy.ts

```
import { Mintable } from "./contracts/ERC721";
import { Client } from "./client";
import { CallTx } from "@hyperledger/burrow/proto/payload_pb";

export async function Deploy(account: string): Promise<Mintable.Contract<CallTx>> {
	const client = new Client('localhost:10997', account);
	const address = await Mintable.Deploy(client);
	return new Mintable.Contract(client, address);
}
```

Burrow (3)

src/test/mintable.test.js

```
import { Deploy } from '../deploy';
import * as assert from 'assert';

describe("Mintable", () => {
        it("Should deploy and mint token", async () => {
                const owner = '<ADDRESS>';
                const id = 1337;
                const contract = await Deploy(owner);
                await contract.mint(owner, id);
                assert.equal(await contract.balanceOf(owner), 1);
                assert.equal(await contract.ownerOf(id), owner);
        });
});
```

Networking (1)

```
~$ burrow spec -v4 | burrow configure -s- --pool --json

~$ burrow start -c burrow001.json &

~$ burrow start -c burrow002.json &

~$ burrow start -c burrow003.json &

~$ burrow start -c burrow004.json &
```

Networking (2)

```
~$ curl -LO https://gist.githubusercontent.com/gregdhill/<ID>/burrow.toml

~$ burrow start --config burrow.toml --address ${ADDRESS}

~$ curl 'http://localhost:26658/account?address=${ADDRESS}'
```

HYPERLEDGER GLOBAL FORUM PHOENIX, AZ | MARCH 3-6, 2020

Networking (3)

```
~$ burrow tx formulate bond --amount ${AMOUNT} --source ${ADDRESS} > tx.json
~$ burrow tx commit --file tx.json
~$ curl http://localhost:26658/network
~$ curl http://localhost:26658/validators
```

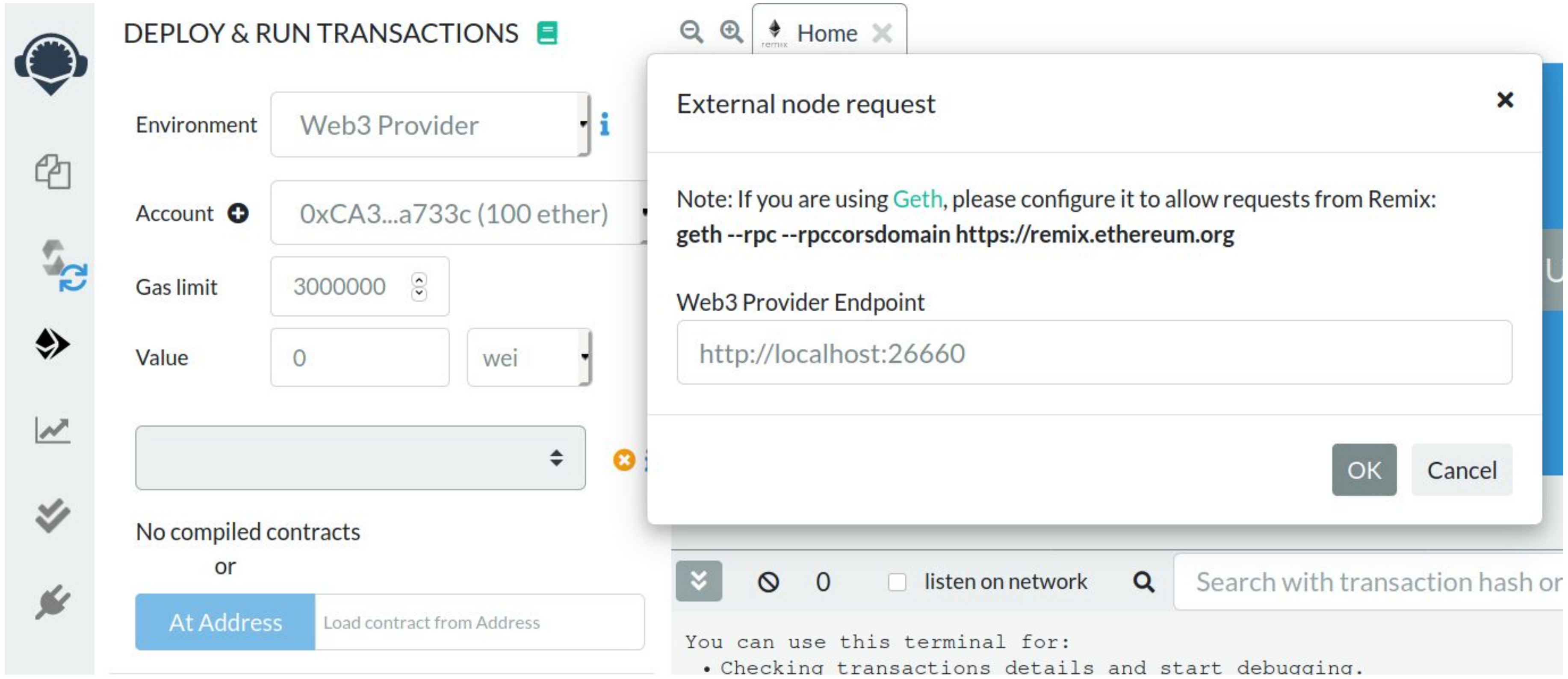
CryptoMarmots

- Fungibility: ERC20 / ERC721

- Auction
- Breed
- Collect

HYPERLEDGER GLOBAL FORUM PHOENIX, AZ | MARCH 3-6, 2020

http://remix.ethereum.org


DEPLOY & RUN TRANSACTIONS
Home
OK
Cancel
At Address
Load contract from Address
listen on network
Search with transaction hash or

Environment: Web3 Provider
Account: 0xCA3...a733c (100 ether)
Gas limit: 3000000
Value: 0 wei

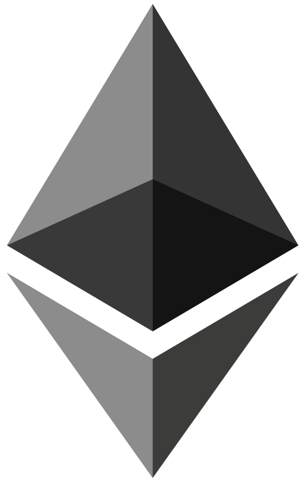
No compiled contracts
or

External node request

Note: If you are using Geth, please configure it to allow requests from Remix:
geth --rpc --rpccorsdomain https://remix.ethereum.org

Web3 Provider Endpoint
http://localhost:26660

```
You can use this terminal for:
 • Checking transactions details and start debugging.
```



Interact

```
pragma solidity >=0.4.22 <0.7.0;
contract Marmots {
        function transfer ( address _to, uint256 _tokenId ) external;
        function tokensOfOwner ( address _owner ) external view returns ( uint256[] memory ownerTokens );
        function getMarmot ( uint256 _tokenId ) external view
                returns ( bool isGestating, bool isReady, uint256 cooldownIndex, uint256 nextActionAt,
                        uint256 siringWithId, uint256 birthTime, uint256 matronId, uint256 sireId,
                        uint256 generation, uint256 genes );
        function breedWithAuto ( uint256 _matronId, uint256 _sireId ) external payable;
        function createPromoMarmot ( uint256 _genes, address _owner ) external;
}
```

HYPERLEDGER GLOBAL FORUM

PHOENIX, AZ | MARCH 3-6, 2020