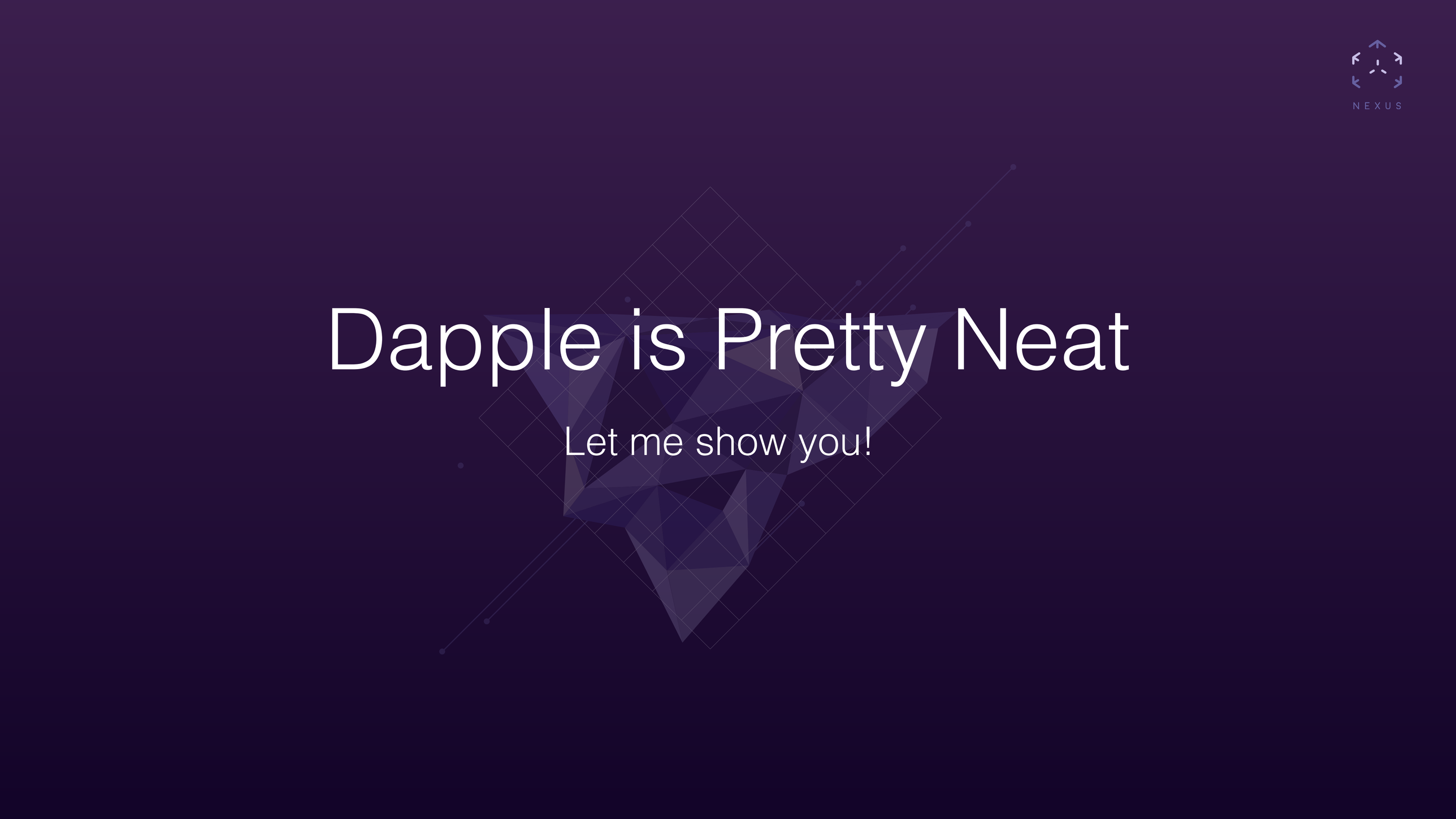

NEXUS
Dapple is Pretty Neat
Let me show you!

What?

- It’s an EVM Dev Multitool
- Collection of utilities centered around common data model

NEXUS

Key Innovations

- `Dappfile`: package/dapp descriptor format
- Chain Forks (local)
- EVM extensions

NEXUS

Key Innovations

- `Dappfile`: package/dapp descriptor format
  - Shared global runtime environment blurs line between **code packages** and **deployed code-objects**

```
version: 1.0.0
tags: []
layout:
  sol_sources: src
  build_dir: build
  packages_directory: .dapple/packages/
dependencies:
  dappsys: 0.2.6
ignore: []
name: 006_multiarray
environments:
  develop:
    objects:
      mytoken:
        value: '0x3fcc401d928e09043fc1b4939ab9df72f2fe607c'
        type: Token[0087c30be200773ded359e676165fa650deee3b3e58ec24893b39d20ff961ae8]
    type: internal
  morden:
    objects:
      mytoken:
        value: '0x67658a48be1b39db1c111dfca6f8fed09ac0da4e'
        type: Token[bc365dfdfd4b67086a9483f7ce915da522a252b1b0289f1b8e07c16d06a476be]
    type: MORDEN
dapple_version: 0.8.0-dev
```

NEXUS

Key Innovations

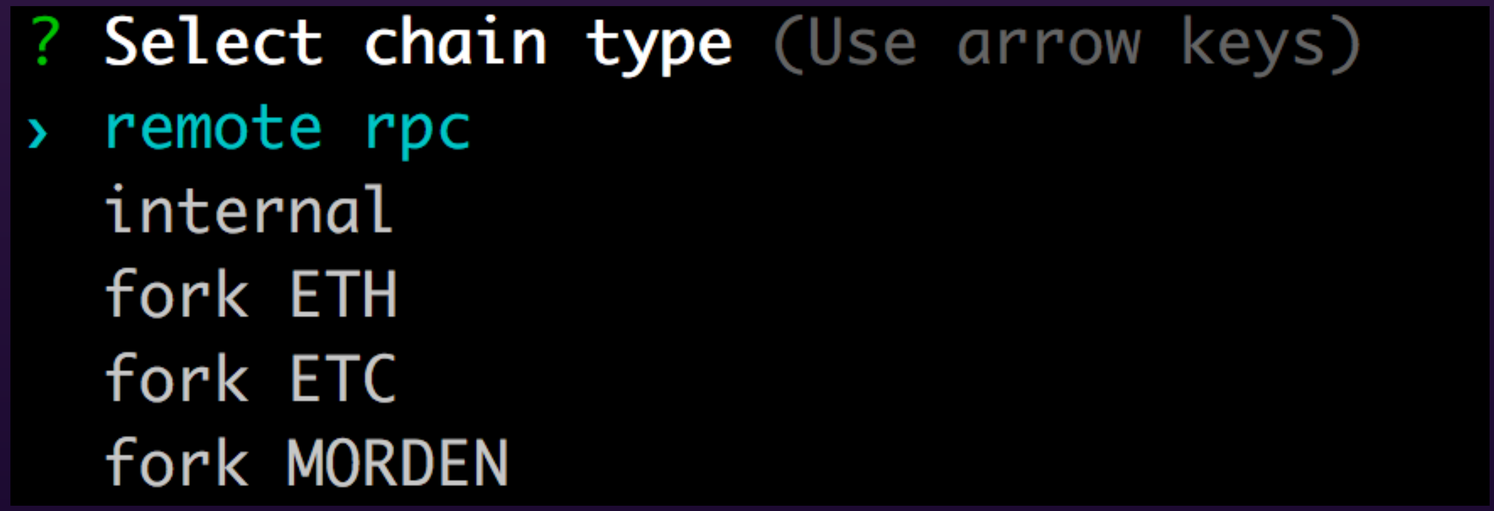
- Chain Forks (local)

```
dapple chain status
dapple chain ls
dapple chain rm <name>
dapple chain fork <name>
dapple chain checkout <name>
dapple chain server
dapple chain fake <address>
dapple chain new [<name>]
```

NEXUS

Key Innovations

- Chain Forks (local)



NEXUS

Key Innovations

- EVM extensions
  - Break the rules… privately

NEXUS

The Workflow

NEXUS

1) Find some Dependencies

- Hint hint: github.com/nexusdev/**dappsys**
  - Auth, proxy actors, tokens, governance, datastores, utils…
  - Contract system building blocks
  - Top candidate for formal verification!

NEXUS

2) Build/Link (yawn)

- Problem: Solidity has no namespaces
  - maker-core has objects from 4 different dappsys versions
- Solution: Custom Linker

NEXUS

3) Test

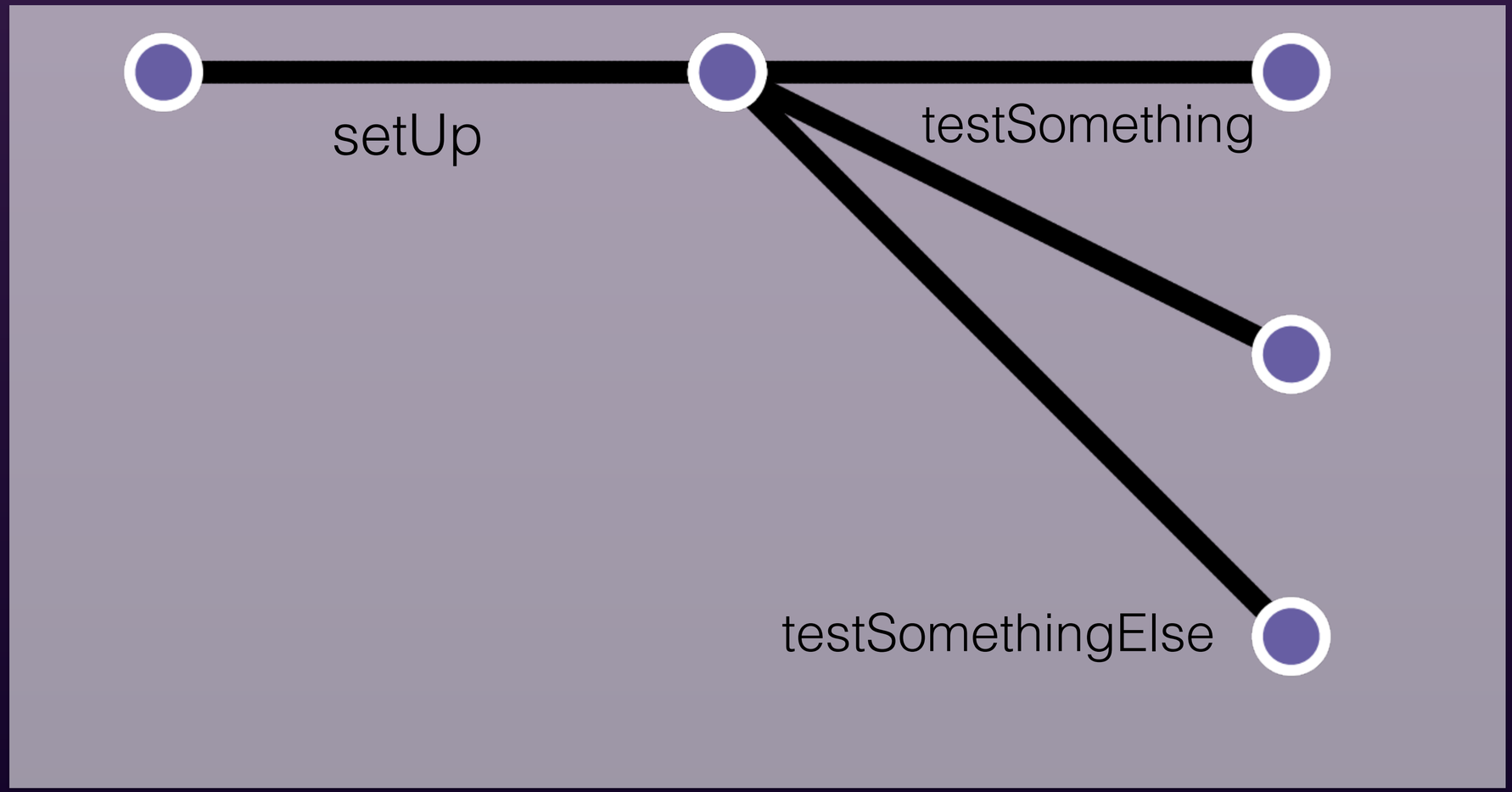
- Dapple perspective: Primary consumer of a contract is *other contracts*
- Magic `Test` contract, harness knows test definition conventions

NEXUS

3) Test

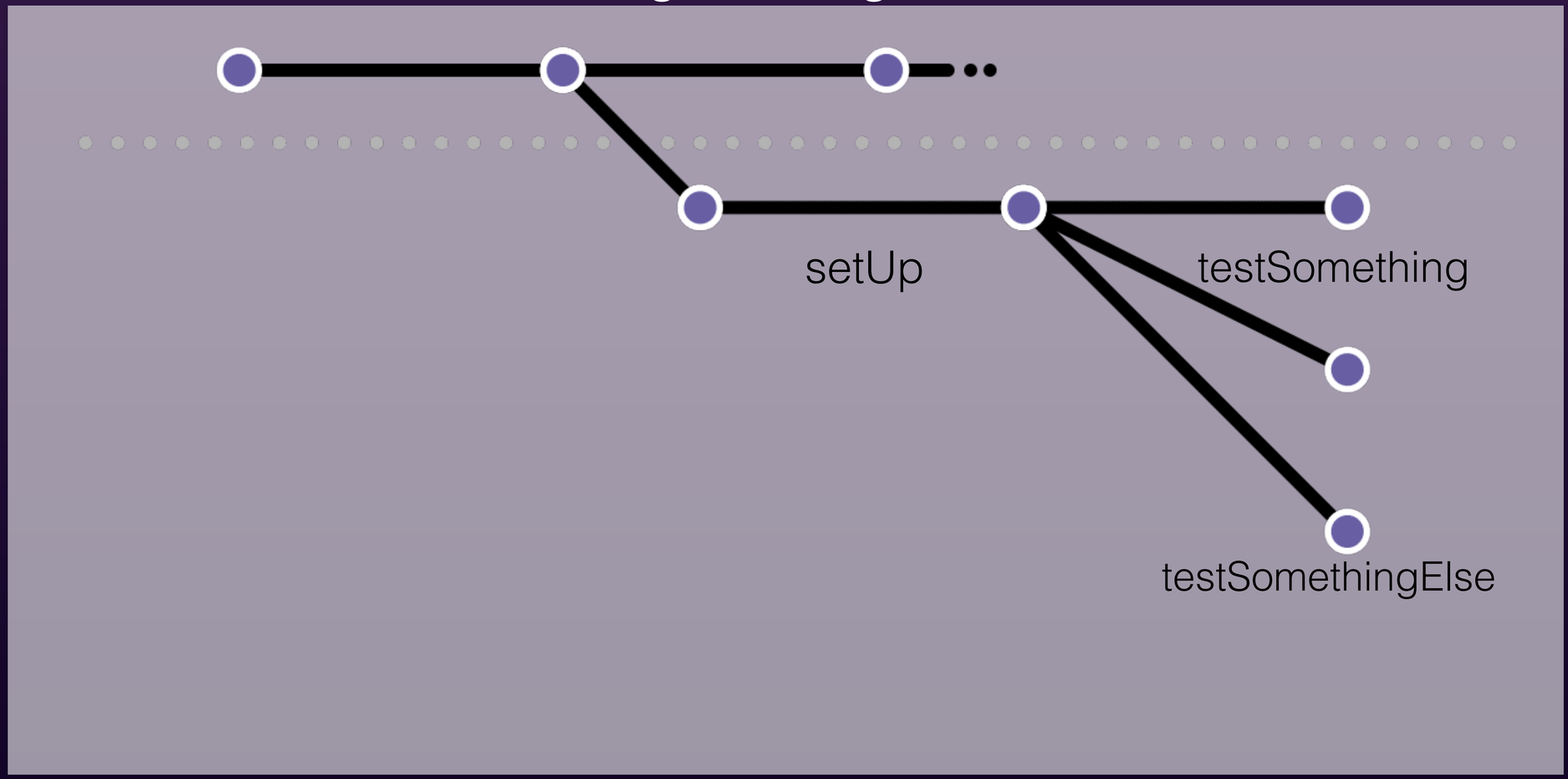
```
1  import "dapple/test.sol";
2  import "./token_events.sol";
3
4  contract Caller is Test, TokenEvents {
5    Token token;
6    function setUp() {
7      token = new Token();
8    }
9    function testEvents() {
10     expectEventsExact(token);
11     token.transfer(0x0, 42); // trigger
12     BalanceUpdate(0x0, 42); // Event
13   }
14
```

3) Test



3) Test

- Now add chain forking: Test against live chain



NEXUS

3) Test

- Now add chain forking: Test against live chain

```
1  import "dapple/test.sol";
2
3  contract MyTest is Test {
4    function testBurning() {
5      address burned = 0x0;
6      uint balance = (burned.balance/ 1 ether);
7      //@log there are `uint balance` eth burned.
8      assertTrue(burned.balance > 0);
9    }
10 }
```

NEXUS

3) Test

- Now add chain forking: Test against live chain

```
MyTest
  test burning
  LOG:  there are 1.000000001000000000001397e+24 eth burned.
  Passed!
```

NEXUS

4) Deploy: Wallet-side Scripting

- Magic `Script` contract
  - Hijacks `CREATE` and `CALL` opcodes - makes a cross-chain **transaction** instead of **message call** and **blocks execution** until confirmed
  - combine with chain forking to simulate deployments and perform consistency checks on result of live system

NEXUS

4) Deploy: Wallet-side Scripting

“nikolai: Please confirm proposal 66 ASAP,
it is a critical bugfix!”

NEXUS

4) Deploy: Wallet-side Scripting

```
> dapple chain new
? Chain name proposal66
? Select chain type fork ETH
> dapple chain fake 0x1234 # other admin
> dapple run ConsistencyCheck
   TXR   Token(0x1111).confirm()
   |     GAS   21423

   ACC  0x1234

   TXR   Token(0x1111).confirm()
   |     GAS   21423

   LOG   is consistent
   |     consistent: false
```

NEXUS

4) Deploy: Wallet-side Scripting

Notice the hoisted `env` object

```
1  import "dapple/script.sol";
2
3  contract ConsistencyCheck is Script {
4    event logConsistency(bool consistent);
5    function ConsistencyCheck () {
6      env.token.confirm();
7      setOrigin(0x1234);
8      env.token.confirm();
9      logConsistentcy(env.token.checkConsistency());
10   }
11 }
```

NEXUS

5) Publish your Dappfile

Rinse and Repeat

NEXUS

Sneak Peek

- On-chain registry and power user block explorer
  - dapphub.io
    - GUI for chain forks, dry runs, simulations, etc

NEXUS

Sneak Peek

- More wallet-side EVM extensions:
  - Thin clients - DIY sharding
  - Consensus in VM
  - System calls: Private Sidechains bridge old web and EVM proofs

NEXUS

Sneak Peek

```
1  import "dapple/script.sol";
2  import "./caller.sol";
3
4  contract InstallCallback is Script {
5
6    function InstallCallback () {
7      exportObject("caller", env.caller);
8      on(caller, "pong", "onPong");
9    }
10
11   function onPong(Caller factory, uint value) {
12     sms.send("+49160942282297", "This is easy");
13   }
14 }
```

Thank you!

nexusdev.us

github.com/nexusdev/dapple

github.com/nexusdev/dappsys