Mini-workshop

Zilliqa <> XXX

zilliqa

Agenda

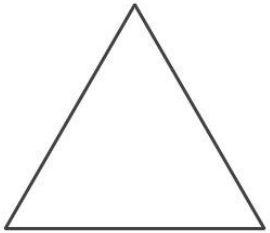
1. Introduction to Zilliqa Network
2. Scilla fundamentals
3. End-to-End Application Development

Introduction to Zilliqa Network

zilliqa

zilliqa

Low Latency Finality
Low Overhead
Small Number of Nodes



Low Latency Finality
High Overhead
Large Number of Nodes

High Latency Finality
Low Overhead
Large Number of Nodes

zilliqa

What Zilliqa aims to achieve?

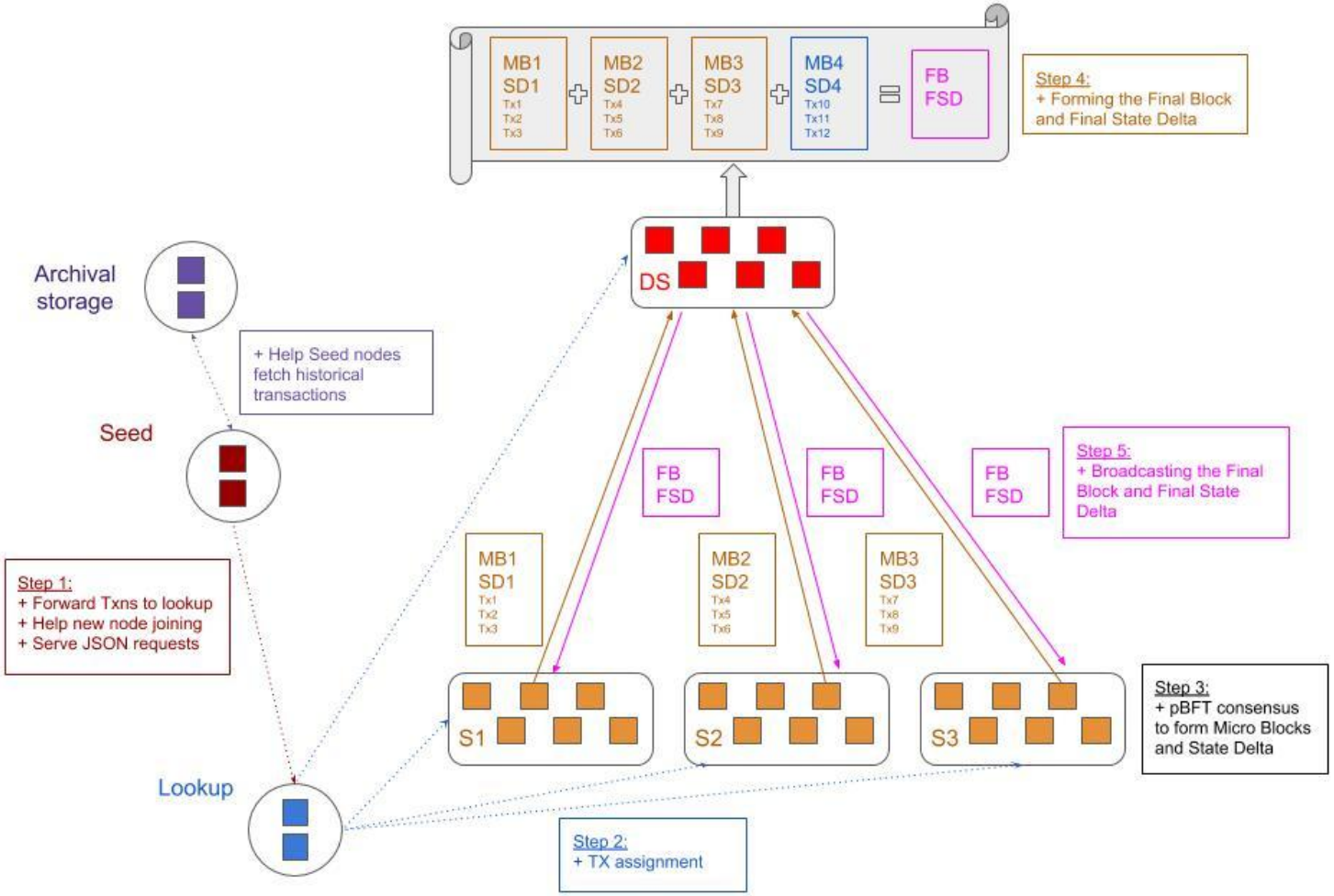
1. Reasonable latency for finality (1~2 mins)
2. Low overhead
3. Large number of nodes

zilliqa

Types of nodes

1. Directory Service (DS) nodes
2. Shard nodes
3. Seed nodes
4. Lookup nodes
5. Archival nodes

Zilliqa's Network Topology



Scilla fundamentals

zilliqa

zilliqa

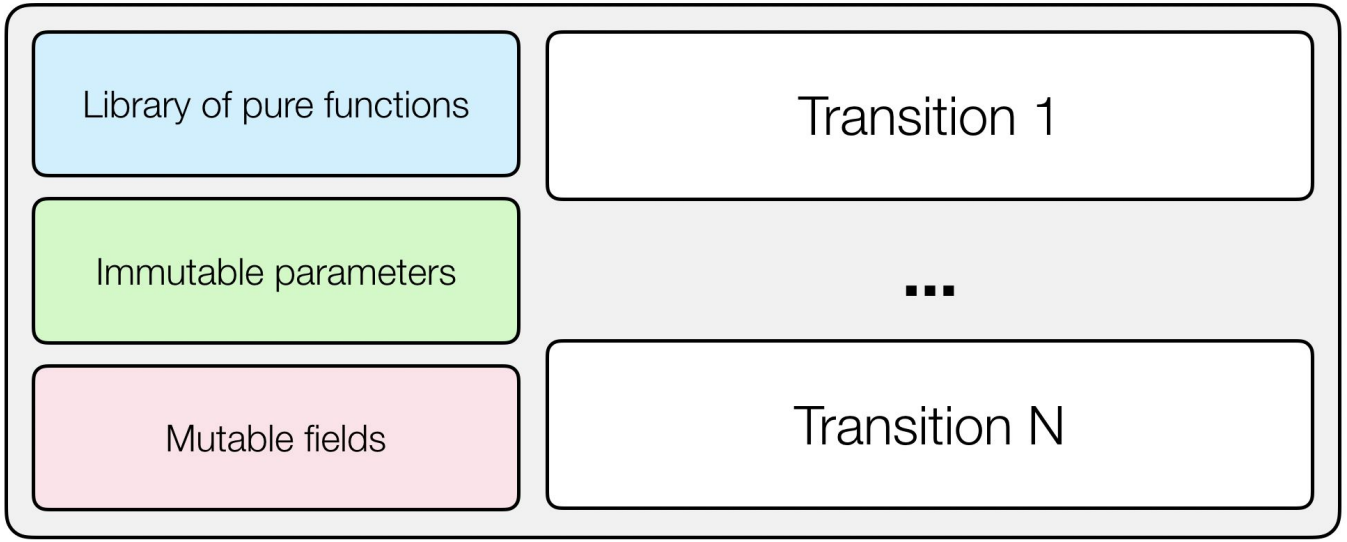
Scilla Features

- NON-TURING COMPLETE
- DECIDABLE CONTRACTS

- AMENABLE TO FORMAL VERIFICATION
- CLEAN SEPARATION: COMMUNICATION VS COMPUTATION

zilliqa

Scilla Basic Structure



zilliqa

Scilla Basic Structure

```
scilla_version 0

library SampleContract

let one = Uint128 1

contract SampleContract
(owner : ByStr20)

field counter : Uint128 = Uint128 0

transition add ()
    (* Only contract owner can increment by 1 *)
    c <- counter;
    c = builtin add c one;
    counter := c
end
```

zilliqa

Commonly used types

```
field OptionSample : Option Bool = None

field MapSample : Map ByStr20 Uint128 = Emp ByStr20 Uint128

field AddressSample : ByStr20 = "0x5fC7409B4b41e06e73BA1aA7f3127d93c76bD557"

field UintSample: Uint128 = Uint128 1000

field ListSample : List ByStr20 = let nil_address = Nil {ByStr20} in Cons {ByStr20} _this_address nil_address

field PairSample : Pair ByStr20 Uint128 = Pair {ByStr20 Uint128} _this_address _amount
```

Custom ADTs (Structs on steroids)

```
library Sample_ADT


(* Create a new type that includes sender address, recipient address, and amount sent *)
type Txn_info =
| Info of ByStr20 ByStr20 Uint128
| NoInfo
```

zilliqa

Fungible token (ERC20) contract

Immutable variables

```
contract FungibleToken
(owner : ByStr20,
total_tokens : Uint128)
```

Mutable variables

```
field balances : Map ByStr20 Uint128
field allowed : Map ByStr20 (Map ByStr20 Uint128)
```

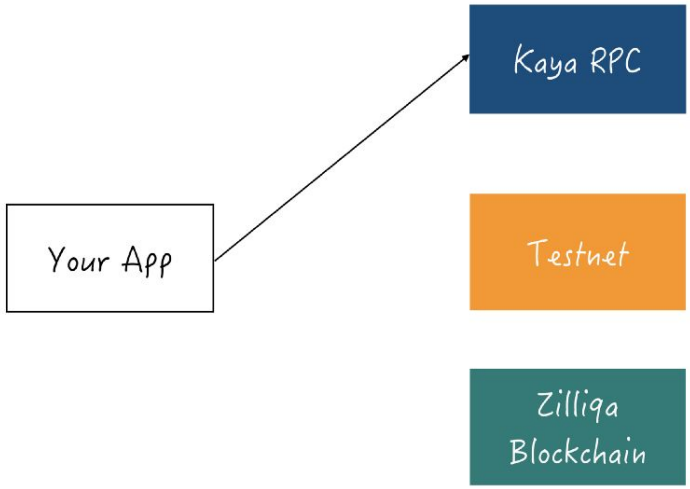
Transitions

```
transition BalanceOf ()
transition TotalSupply ()
transition Transfer (to : ByStr20, tokens : Uint128)
transition TransferFrom (from : ByStr20, to : ByStr20, tokens : Uint128)
transition Approve (spender : ByStr20, tokens : Uint128)
transition Allowance (tokenOwner : ByStr20, spender : ByStr20)
```

End-to-End Application Development

zilliqa

Process: From Conceptualization to Production



Kaya RPC: TestRPC for development and testing

```
edison@edison-zil ~/workspace/master-zilliqa/kaya  hotfix npm run debug:fixtur
es

> kaya-cli@0.2.5 debug:fixtures /Users/edison/workspace/master-zilliqa/kaya
> node src/server.js -f test/account-fixtures.json -v

ZILLIQA KAYA RPC SERVER (ver: 0.2.5)
Server listening on 127.0.0.1:4200
Running from local interpreter
================================================================================
[App.js]         : Bootstrapping from account fixture files: test/account-fixtures.jso
n
[Wallet]         : Valid accounts file
[Wallet]         : 10 wallets bootstrapped from file
Available Accounts
================================================================================
(1) 7bb3b0e8a59f3f61d9bff038f4aeb42cae2ecce8    (1000000 ZILs)  (Nonce: 0)
(2) d90f2e538ce0df89c8273cad3b63ec44a3c4ed82    (1000000 ZILs)  (Nonce: 0)
(3) 381f4008505e940ad7681ec3468a719060caf796    (1000000 ZILs)  (Nonce: 0)
(4) b028055ea3bc78d759d10663da40d171dec992aa    (1000000 ZILs)  (Nonce: 0)
(5) f6dad9e193fa2959a849b81caf9cb6ecde466771    (1000000 ZILs)  (Nonce: 0)
(6) 10200e3da08ee88729469d6eabc055cb225821e7    (1000000 ZILs)  (Nonce: 0)
(7) ac941274c3b6a50203cc5e7939b7dad9f32a0c12    (1000000 ZILs)  (Nonce: 0)
(8) ec902fe17d90203d0bddd943d97b29576ece3177    (1000000 ZILs)  (Nonce: 0)
(9) c2035715831ab100ec42e562ce341b834bed1f4c    (1000000 ZILs)  (Nonce: 0)
(10) 6cd3667ba79310837e33f0aecbe13688a6cbca32   (1000000 ZILs)  (Nonce: 0)

 Private Keys
================================================================================
(1) db11cfa086b92497c8ed5a4cc6edb3a5bfe3a640c43ffb9fc6aa0873c56f2ee3
(2) e53d1c3edaffc7a7bab5418eb836cf75819a82872b4a1a0f1c7fcf5c3e020b89
(3) d96e9eb5b782a80ea153c937fa83e5948485fbfc8b7e7c069d7b914dbc350aba
(4) e7f59a4beb997a02a13e0d5e025b39a6f0adc64d37bb1e6a849a4863b4680411
(5) 589417286a3213dceb37f8f89bd164c3505a4cec9200c61f7c6db13a30a71b45
(6) 5430365143ce0154b682301d0ab7318972219O6a7054bbf5bd83c7663a6cbc40
(7) 1080d2cca18ace8225354ac021f9977404cee46f1d12e9981af8c36322eac1a4
(8) 254d9924fc1dcdca44ce92d80255c6a0bb690f867abde80e626fbfef4d357004
(9) b8fc4e270594d87d3f728d0873a38fb0896ea83bd6f96b4f3c9ff0a29122efe4
(10) b87f4ba7dcd6e60f2cca8352c89904e3993c5b2b0b608d255002edcda6374de4
================================================================================
```

- Easy debugging tool with error traces
- Account management
- Save and resume state
- Supports most of the blockchain RPC calls

https://github.com/Zilliqa/kaya

Or

Npm install -g kaya-cli

zilliqa

Useful JavaScript functions

package	description	Examples
`@zilliqa-js/core`	Core abstractions and base classes, such as `HTTPProvider` and network logic for interfacing with the Zilliqa JSON-RPC.	none
`@zilliqa-js/account`	Classes for managing accounts and account-related actions.	`Wallet`, `SignTransactions`
`@zilliqa-js/blockchain`	Main interface to the Zilliqa `JSON-RPC`.	`GetBlockchainInfo`, `CreateTransaction`
`@zilliqa-js/contract`	Classes for managing Scilla smart contracts on the Zilliqa blockchain.	`deploy`, `call`, `getState`
`@zilliqa-js/crypto`	Exposes several loosely-coupled cryptographic convenience functions for working with the Zilliqa blockchain and its cryptographic primitives, such as Schnorr signatures.	`generatePrivateKey`, `getAddressFromPrivateKey`
`@zilliqa-js/proto`	Protobuf source files and corresponding generated JS modules.	
`@zilliqa-js/util`	Miscellaneous functions that take care of serialisation/deserialisation and validation.	Verify addresses, private keys, etc

zilliqa

Getting Started with JS

https://github.com/edisonljh/zilliqajs-starter

zilliqa

Join our Developer community on Gitter!

Have interesting ideas? Apply for the innovation grant!


zilliqa

zilliqa

Challenge (~2 hrs)

- Create a simple Tic-tac-toe Scilla contract that allows Users:
  - To play a single game of TTT
  - Keep track of which player’s turn in the ledger