Lisboa: 02/05/2019

What makes æternity different?

æternity's goal is to provide one of the most **powerful blockchain** platforms in the industry with a very **reliable** on-chain and off-chain **performance**, while still remaining **decentralized**.

Not ignoring important aspects:

- open code
- 200 years
- stable releases
- dev instruments
- readable language
- active community

Technology

Some of the **technology** we use:

You can find a full list on our GitHub: https://github.com/aeternity

What is æternity?

An open-source, blockchain 3.0 æpps platform.

Solving current problems in blockchain, ie:

- scalability
- governance
- usability
- efficiency
- real world data
- contract security

Why æternity?

Scalability.

On-Chain (Layer 1)

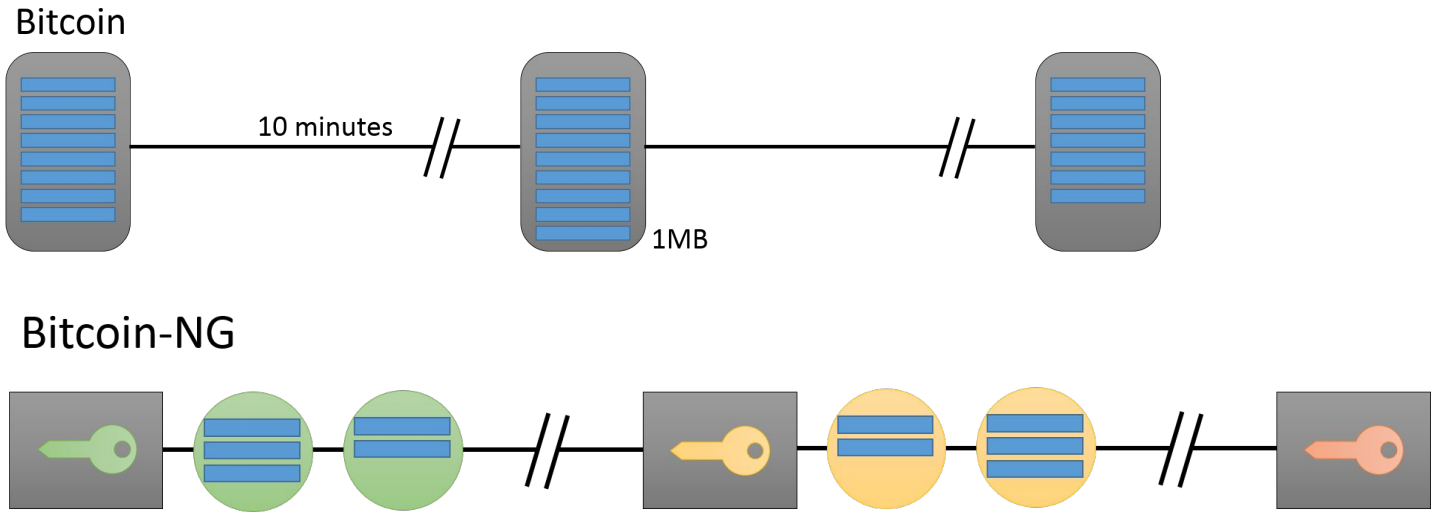
- Bitcoin-NG
  - using the Cuckoo Cycle Proof of Work mining algorithm

Off-Chain (Layer 2)

- State Channels

On-Chain: Bitcoin-NG

Fast, Secure, and Decentralized



Why æternity?

Cuckoo Cycle PoW.

æternity uses the **Cuckoo Cycle**, a memory bound mining algorithm, rather than raw processor speed, in order to achieve the greatest **decentralization** possible.

PoW might use 100% of your CPU (70W) (PoW normally uses GPU for mining instead of CPU or **120W or higher**). The Cuckoo Cycle only uses **54W** of power to fully saturate the memory of your machine.

Why æternity?

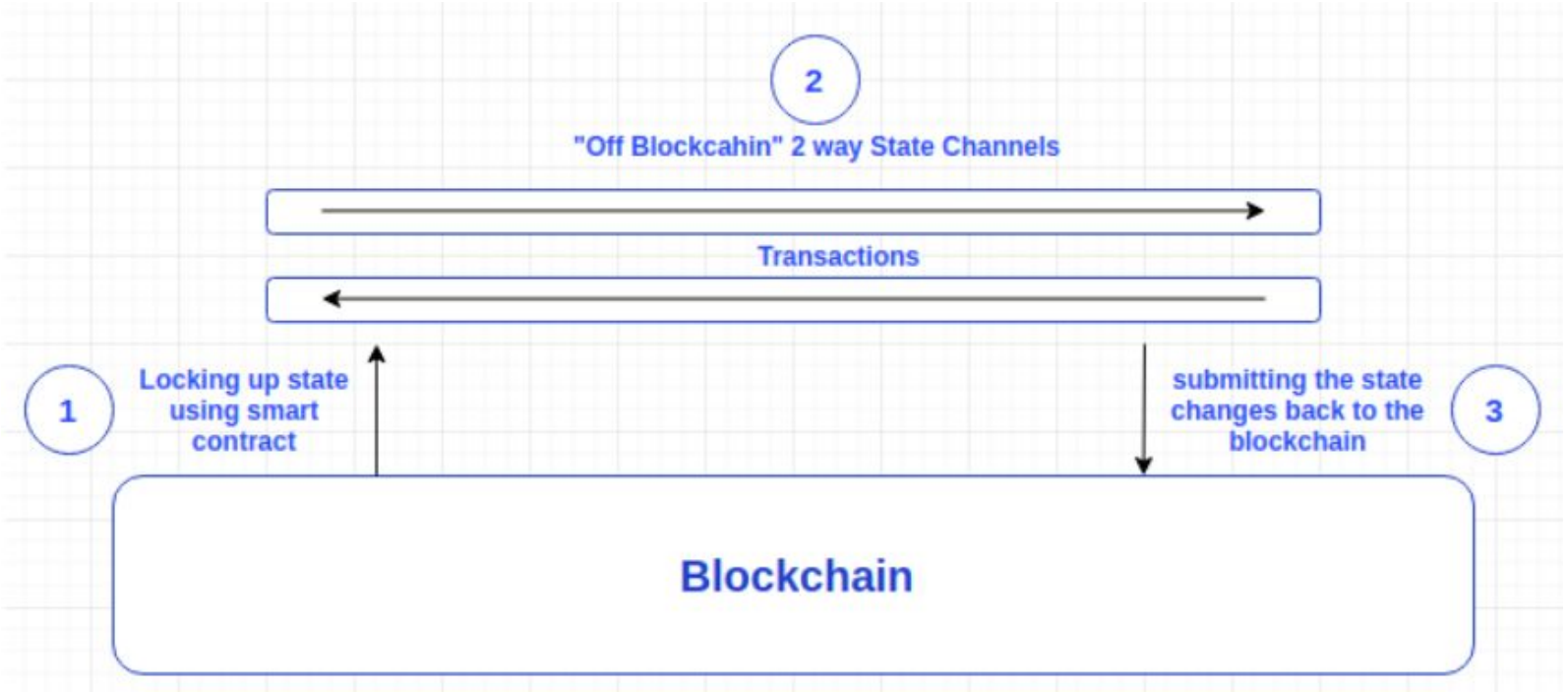
Scalability via State Channels.

State channels provide a method for users to **privately communicate and transact off-chain**. æternity's method for **reducing the on-chain load**.

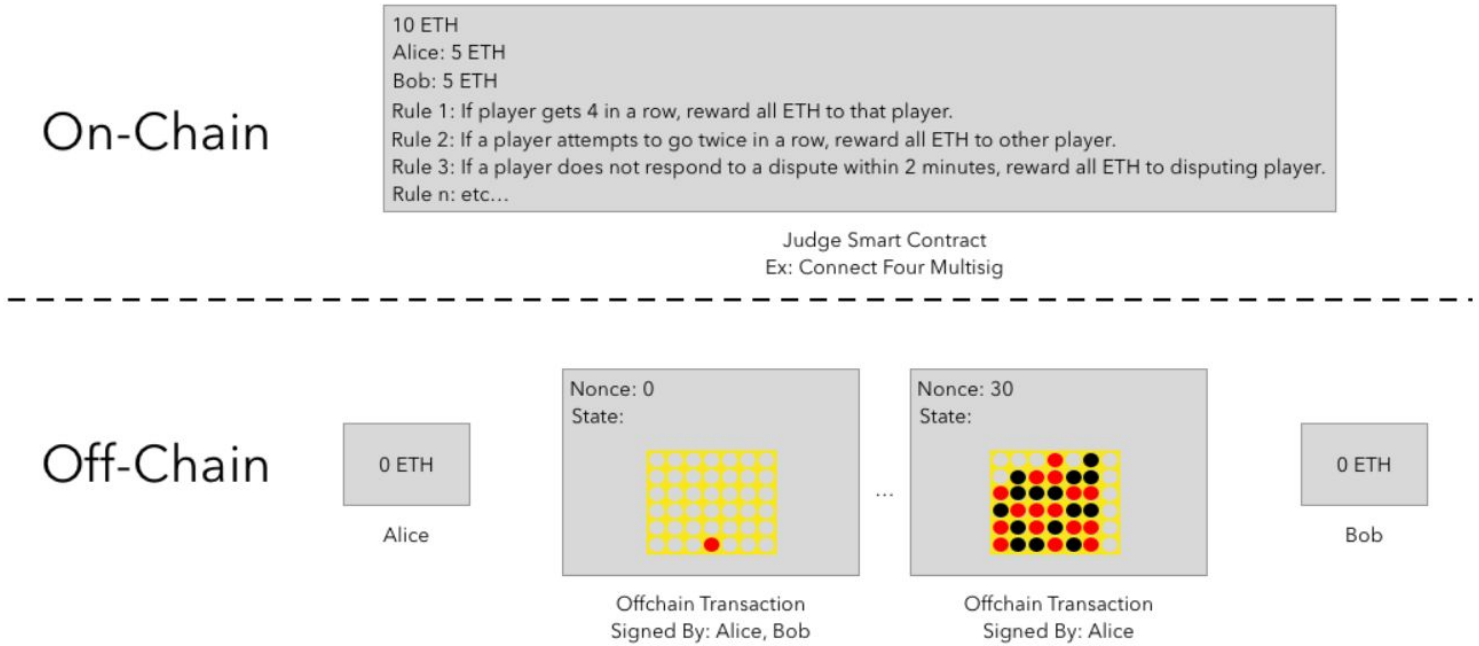
Only in the case of a **disagreement** between the contracting parties does the æternity blockchain **enforce** the smart contract code.

Increases **scalability** by reducing the number of transactions on chain.

Off-Chain: State Channels



Off-Chain: State Channels



Why æternity?

Oracles.

Oracles are source of information which can be accessed on the blockchain. **Anyone can be an oracle provider**, their reputation determines whether or not they are seen as a reliable source.

Structured data, that can be anything from a simple Boolean to the complete work of Shakespeare.

Oracle **responses** can be used by **smart contracts** to perform certain actions based on the result

Why æternity?

This is not an Oracle:

Oracle characteristics

- æternity provides an **oracle state tree** which contains objects and queries
- an oracle defines:
  - an **address** that should be registered as an oracle
  - a **request** and a **response format**
  - a **TTL** (how long the oracle will be available)
  - a **fee** that a caller has to pay to get a response
- operations on oracles:
  - register the oracle
  - subscribe to the oracle
  - answer queries from clients
  - possibly extend the lifespan of the oracle, by increasing its ttl
  - or expire
- Sample Oracle: https://dev.aepps.com/aepp-sdk-docs/Oracle-Python.html

Why æternity?

Governance.

The **governance** of the blockchain is done by **voting on the chain** on proposals.

Delegated voting + technical tools.

Important aspects such as: **reputation and cooperation**.

Why æternity?
æternity smart contract approach.

Contract execution should be **safe, efficient, cheap, scalable +** easily **migrated.**

Sophia is a new reason like programming language invented for the **æternity virtual machine**. Developers write smart contracts that run on the æternity blockchain in Sophia. Sophia's **syntax** is derived from (OCaml/ReasonML). **Varna** is a high level language, similar to Script in Bitcoin. Varna contracts **do not contain any loops** and the gas cost for a call is decided at **compile time**.

Solidity vs Sophia comparison

```
pragma solidity >=0.4.0 <0.6.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

```
contract SimpleStorage =

    record state = { storedData: int }

    public function init() : state = { storedData = 0 }

    public stateful function set(x: int) =
        put(state{ storedData = x })

    public function get() : int = state.storedData
```

hack

Source: https://hack.bg/blog/meetups/0x07-aeternity-sophia-ml-workshop-blockchain-developers-meetup/

Adoption at æternity?

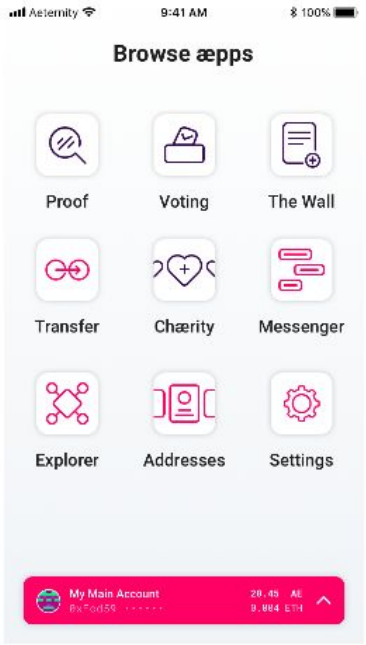
Product / UI / Mobility / Simplicity.

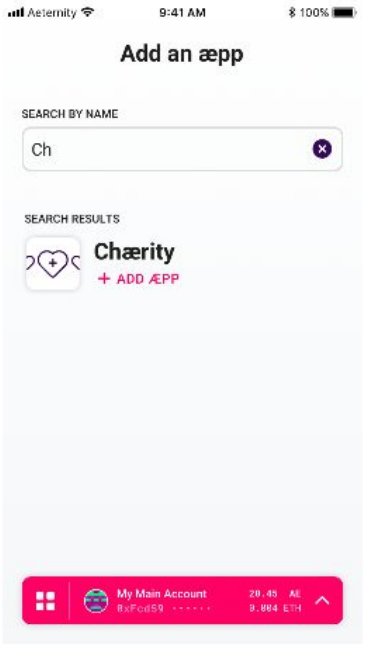
æternity enjoys **simplicity** of delivery of **open source** product to the **end user**. For example: dev portal, æpp portal, explorer that actually provides **information in one place**.

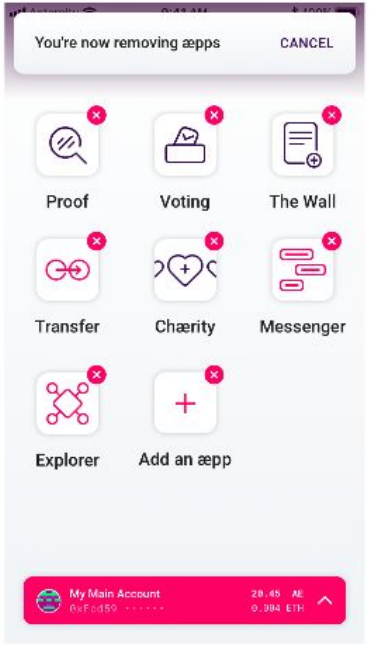
æternity is **mobile friendly** in its essence. No unnecessary and complicated information, which goes to prove that **simplicity and product go hand in hand**.

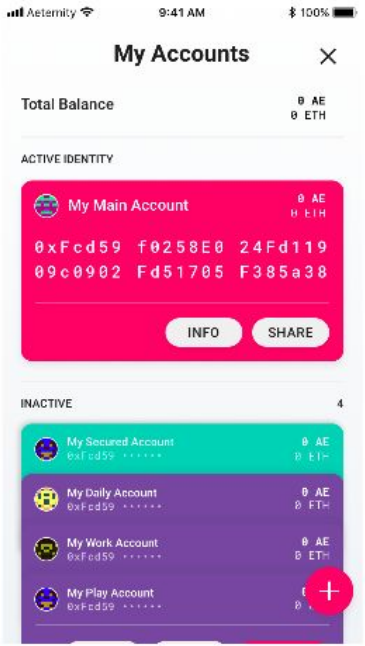
æpps Example

Base æpp



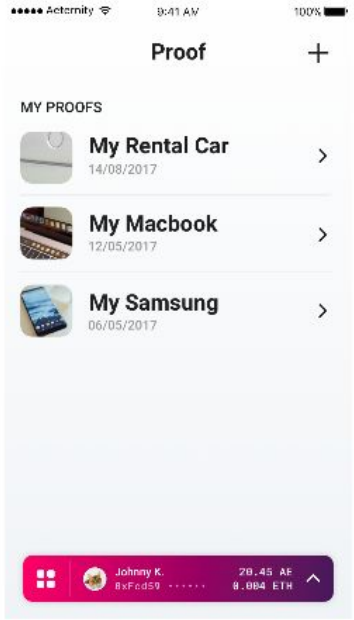


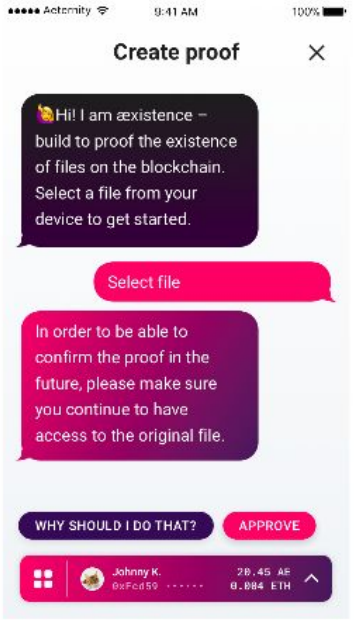




æpps Example

Proof of existence æpp





Mainnet is live!

Build your own æpp.

Some of the first æpps are dedicated to groundbreaking use cases, such as:

- Tokenized trust infrastructure based on machine learning
- Electricity network governance for supply companies and cooperatives
- Intellectual property management for artists and creators
- IoT data search and filtering engine + AI analysis tools
- First aid services for public motorbike transport
- Proof of provenance for agroecological products

æternity ventures.

Be a part of æternity.

Incubator for Blockchain startups building applications on æternity platform:

- Up to $250K
- Up to 6 months incubation
- Global workplace
- Mentoring
- Connect with devs
- Already in action

Join the local community on telegram

https://t.me/aeternitypt

or scan the QR-Code