



Tendermint |  ignite



What is Ignite (Tendermint)?

Is a software for **securely** and **consistently replicating** an application on many machines.

- It works even if **up to 1/3 of machines fail** in arbitrary ways.
- Every machine sees the **same transaction log** and computes the **same state**.

The **ability to tolerate machines failing** (also becoming malicious) is called Byzantine fault tolerance (**BFT**).

Byzantine Fault Tolerance

A Byzantine failure is the loss of a system service due to a Byzantine fault in systems that require consensus.

Nodes must reach consensus on the current state of the system.

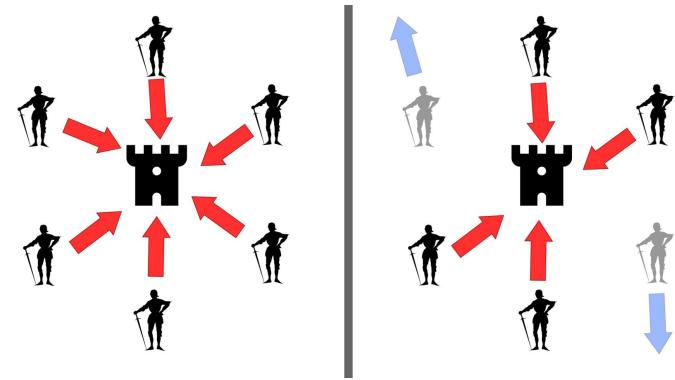
The **majority** of participants in a distributed network **must agree** and perform the same action in order to avoid a complete failure.

To achieve consensus:

There is the need to have at **least $\frac{2}{3}$** or more reliable and **honest** nodes.

If **$>\frac{1}{3}$** of the network is **dishonest**, the system is susceptible to errors and attacks.

By Lord Belbury - Own work. This file was derived from Noun Project - Castle tower.svg|KnightSilhouette2.svg., CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=108234603>

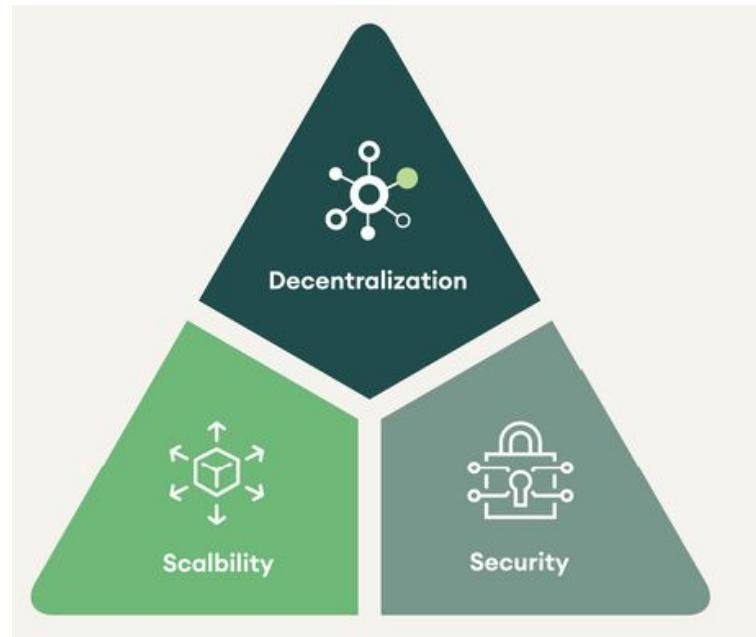


Ignite (Tendermint) goal

Developers tool to skip technical cryptography focusing on higher level development.

Benefits

- speed
- high scalability
- secure
- decentralization



What is Ignite?

Nowadays there are 2 development paradigms:

monolithic and virtual-machine blockchain:

Building dapps on top of existing blockchain as a set of **smart contracts**. SC are good in single-use applications but **limited** in flexibility, performance and sovereignty.

app-specific blockchain:

Customized to operate a single application. All freedom to make design decisions required for the application to **run optimally**. Better security, sovereignty and performance.



What is Ignite?

Ignite allows to **build on top** of the BFT consensus
whatever logic you like:

- key-value store
- cryptocurrency
- e-voting platform
- exchange

You can even build **Ethereum** on top of Ignite. (**Evmos**)



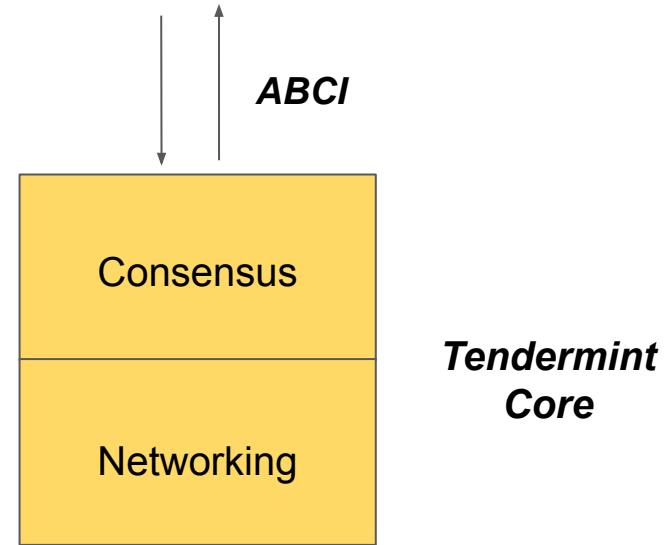
The **Cosmos network** is a clear working **example** of
cryptocurrency applications built on Ignite.

What is Ignite?

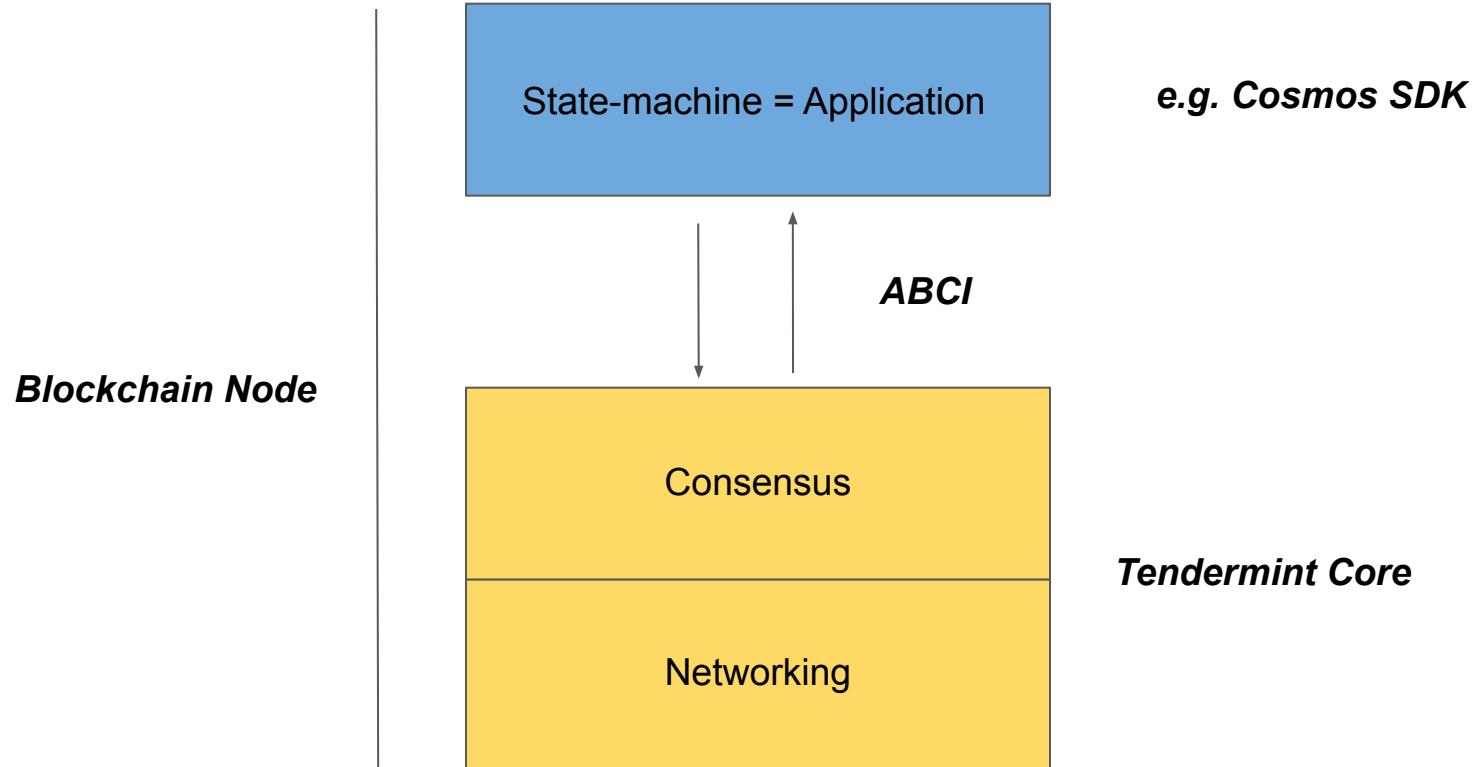
It consists of **three** components:

- a generic application interface (**ABCI**)
- a blockchain consensus engine (**Tendermint Core**)
- a networking layer (p2p layer)

Developers can use Tendermint for BFT state machine replication of applications written in **whatever programming language** and development **environment**.



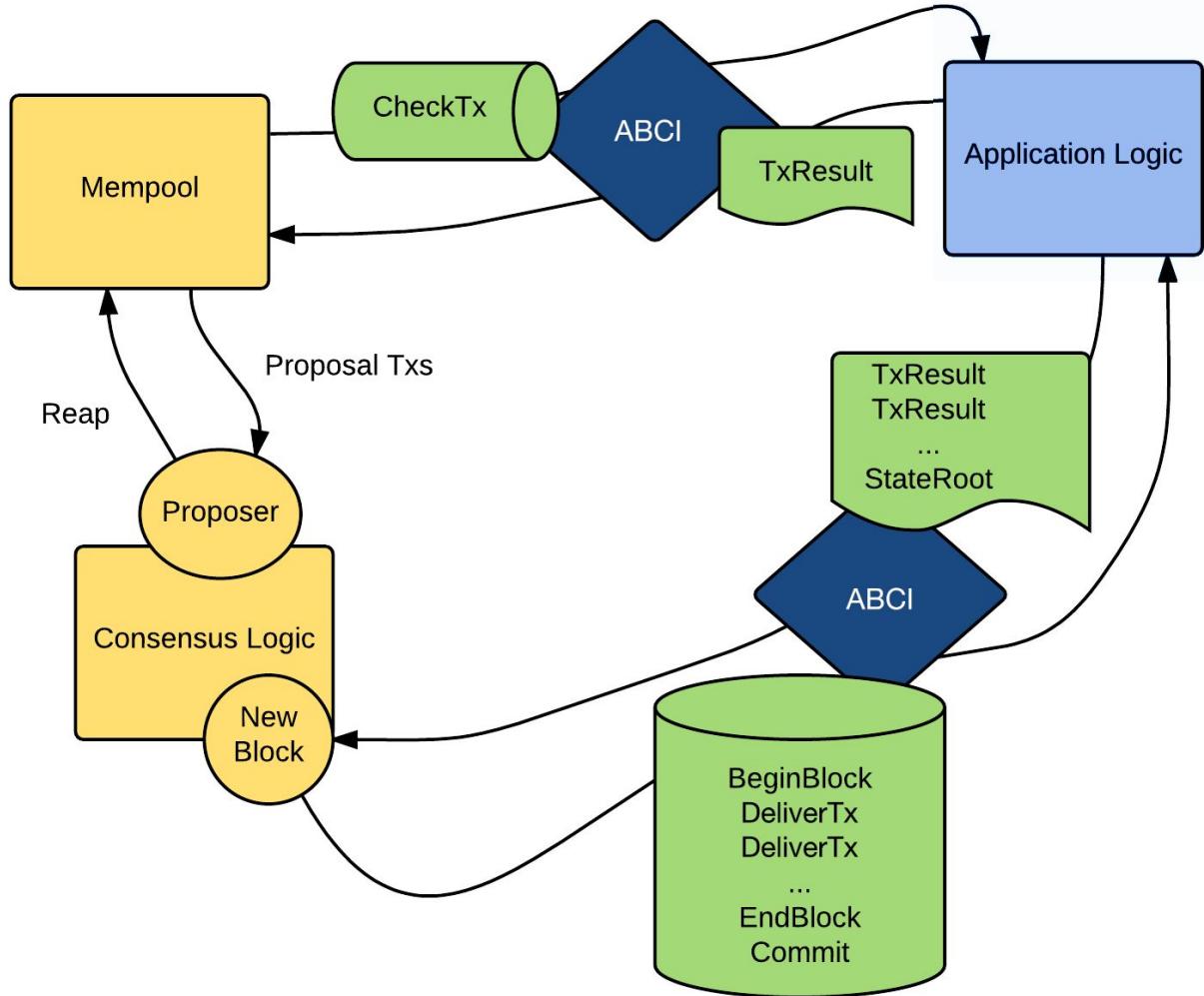
App-specific blockchain



ABCI

ABCI is the **API** between the **application** process and **consensus** process.

- **DeliverTx**: each transaction is delivered with this message.
- **CheckTx**: message to validate tx.
- **Commit**: to compute a cryptographic commitment to the current app state to be placed in next block header.



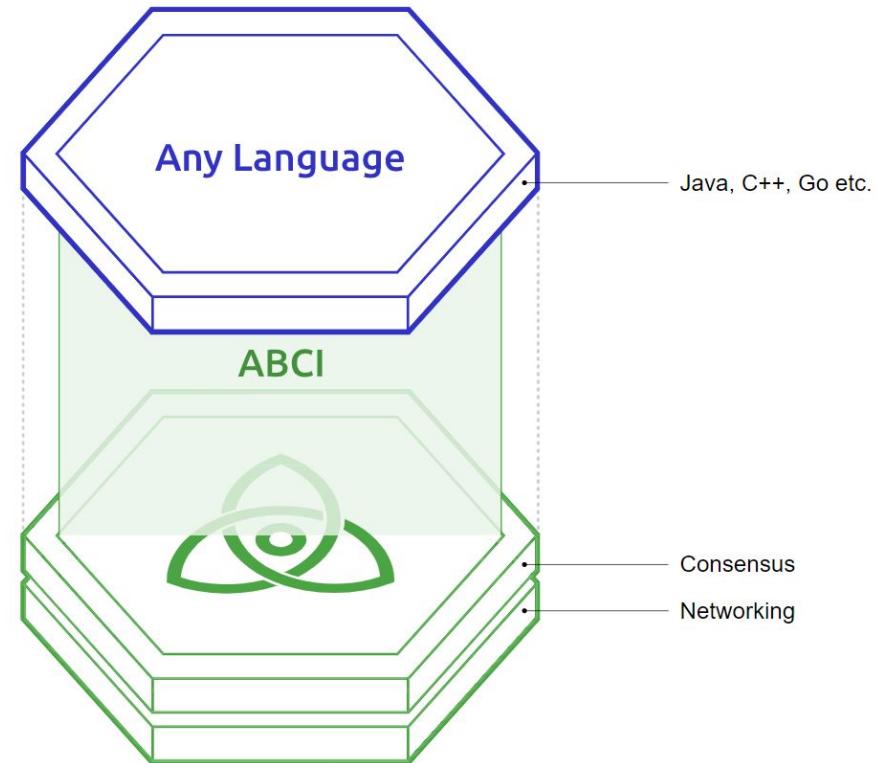
ABCI

3 ABCI connections to the app:

- **mempool connections** - checking if a transaction should be relayed before commit.
- **consensus connections** - for executing transactions already committed.
- **query connections** - for querying the application state.

Aim: decouple consensus engine and P2P layers from details of application state of blockchain app. Allows to program blockchain in any language

Tendermint decompose blockchain offering API between App layer and consensus process.



THE SYBIL RESISTANCE MECHANISM SECURING TODAY'S TOP BLOCKCHAINS

Sources: Global X ETFs with information derived from: CoinMarketCap. (n.d.). Top blockchain assets sorted by market cap Accessed on September 20, 2022.

Proof of Work



Dogecoin



Ethereum
(Pre-merge)



Bitcoin



Ethereum
(Post-merge)

Proof of Stake



Binance
Smart Chain



Cardano



Solana



Polkadot



Polygon

Cryptocurrency Consensus Mechanism - PoW vs PoS

PoW in brief

Energy-intensive process

Miners work to solve for the hash to verify transactions.

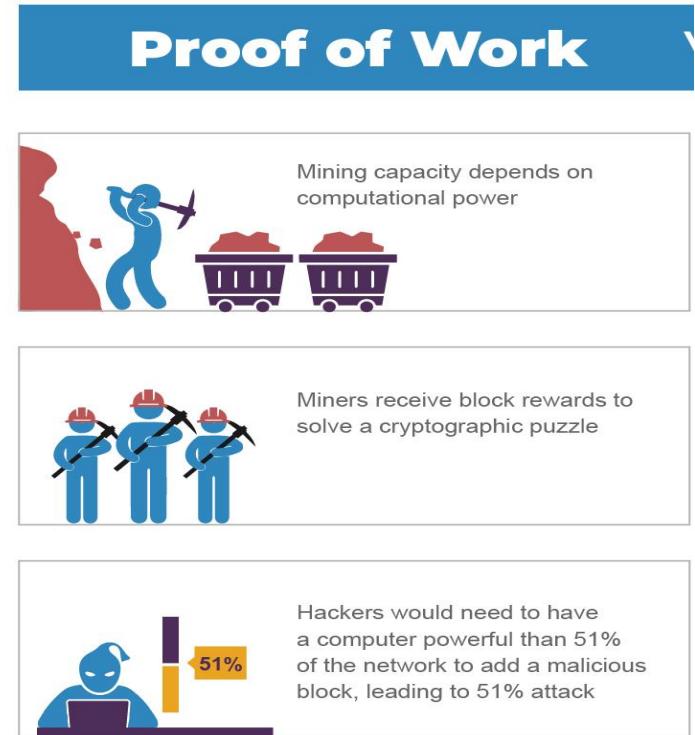
Competitive approach to verifying transactions encouraging people to look to gain an advantage.

Bitcoin miners earn bitcoins by verifying a new block.

PoS Pro

Reducing the amount of computational work needed to verify blocks is more environmentally sustainable.

Compensation structure make an attack less advantageous, decreasing the potential for an attack on the network.



Proof-of-Stake

PoS is a cryptocurrency consensus mechanism for processing transactions and creating new blocks in a blockchain.

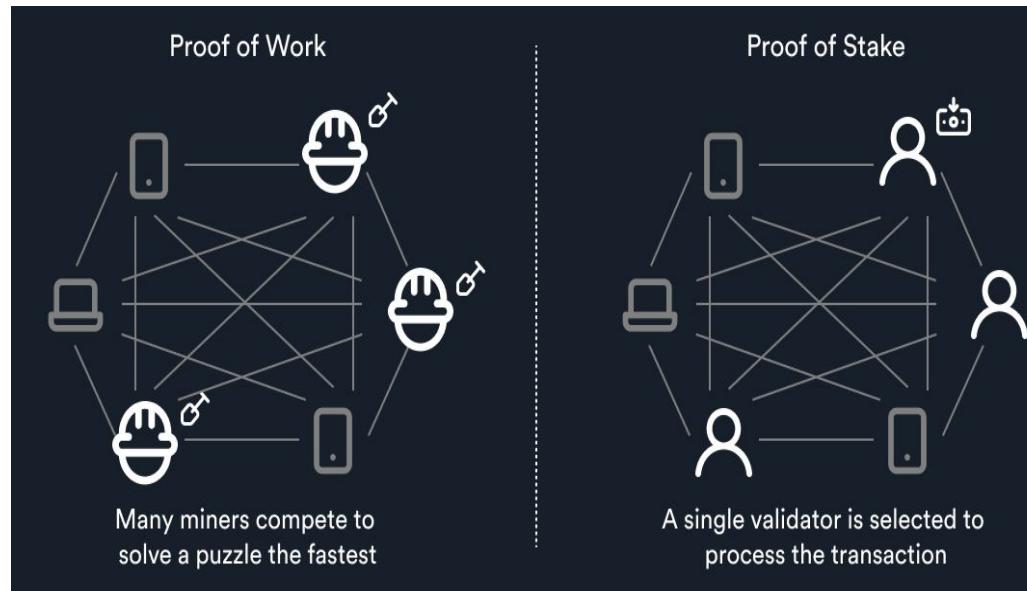
Validators are "staking" an amount of coins and are randomly selected based on the staked amount.

Node activities :

- checks transactions
- verifies activity
- votes on outcomes
- maintains records

Blocks are validated by more than one validator, and when a specific number of validators verify that the block is accurate, it is finalized and closed.

In cosmos-blockchain validators receive block rewards.



ATOM tokens :

to pay fee, as staking tokens, ATOM holders govern the Cosmos hub with their staked Atom by voting on proposals.

Energy consumption

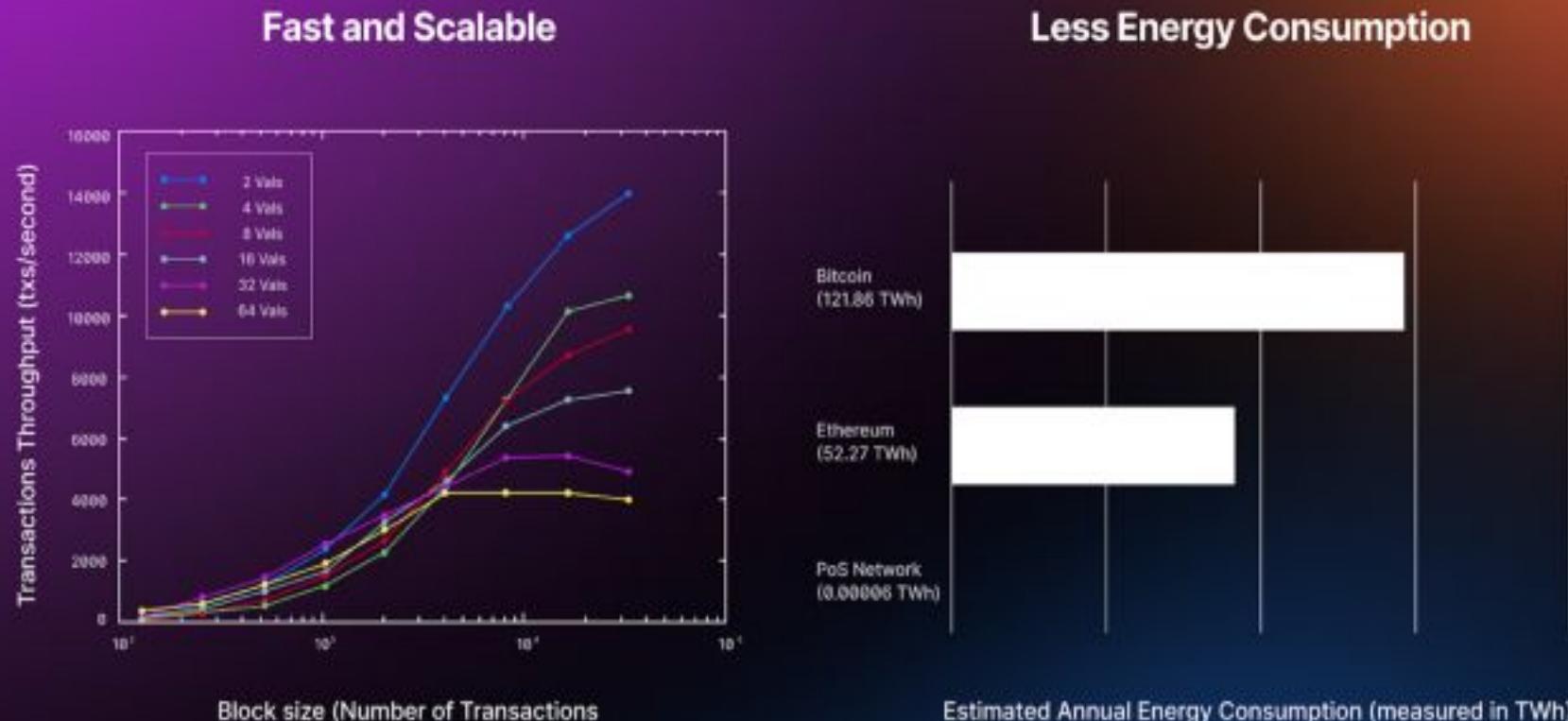
Estimated annual energy consumption:

Bitcoin is 121.86 TWh

Ethereum pre-merge was 52.27 TWh

Cosmos-based blockchain (PoS) is 0.00046647 TWh

Less than the energy consumed by
Bitcoin and Ethereum **in a single day**



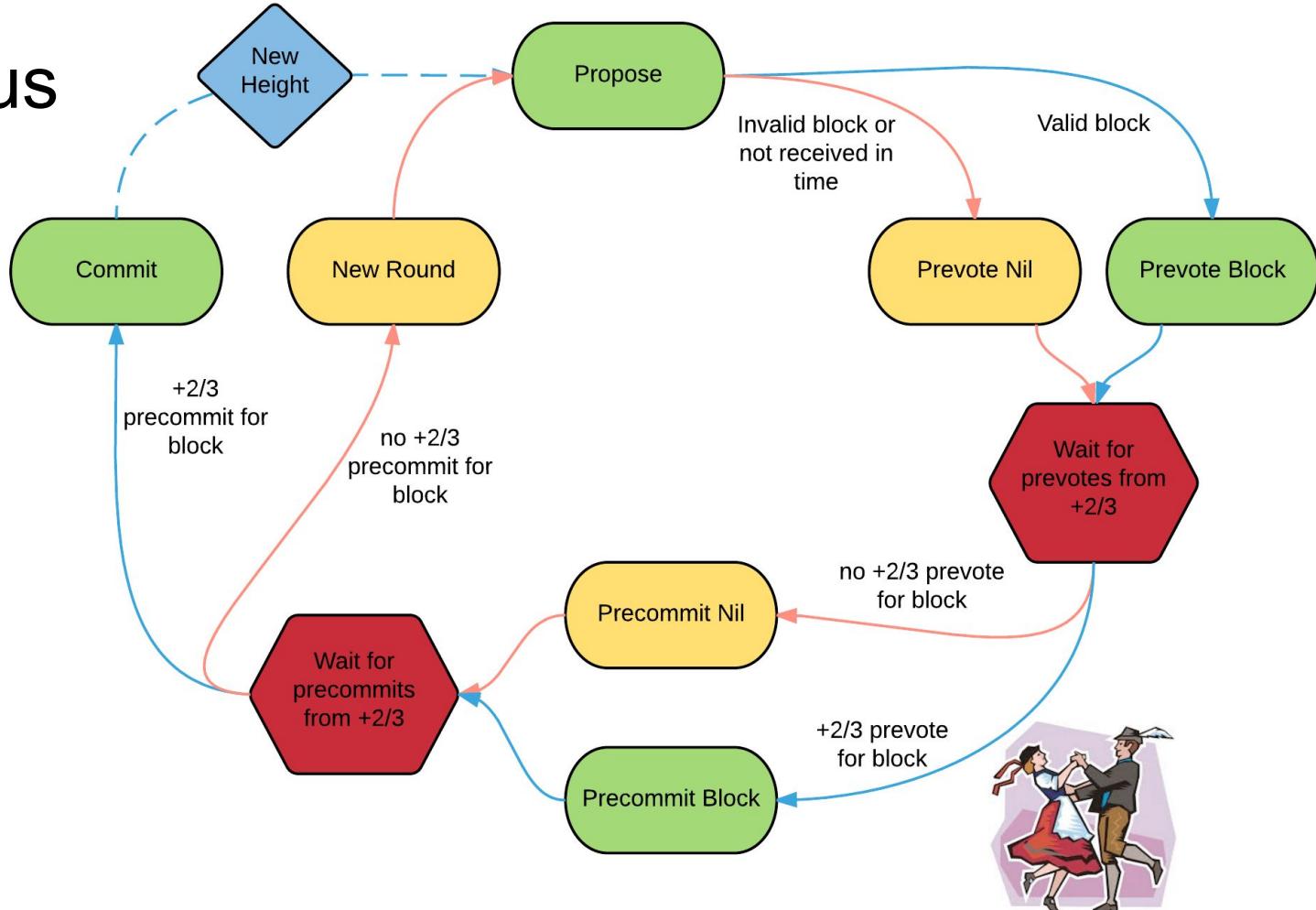
Consensus

Not all validators will have the same "weight" in the consensus protocol.

We are interested in proportions of the total voting power, which may not be uniformly distributed across individual validators.

It is possible to define a currency, and denominate the voting power in that currency. In that case the system is referred to as Proof-of-Stake.

Stake and lock can be implemented by application logic.

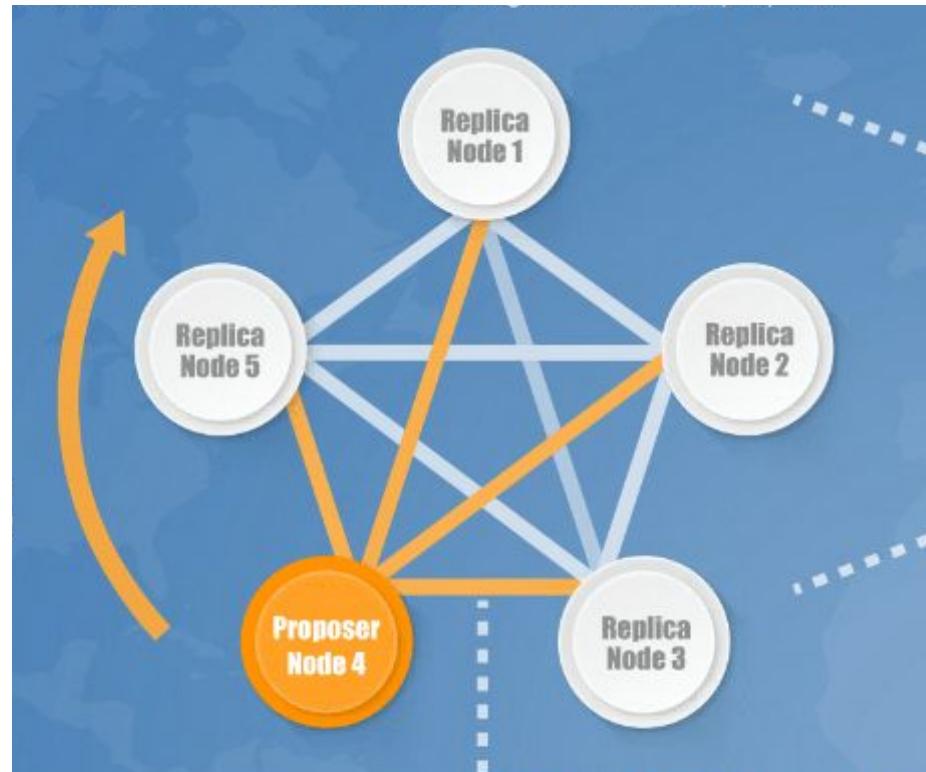


Block Proposer rotation

Each round a new node is designated as Proposer of the new block

Blocks are amended in 1–2 seconds.

Tendermint processes between 4,000 and 16,000 transactions per second (TPS)



Node types

Tendermint consensus leverages the use of these main node types:

Validation nodes:

validate and commit next block

Full nodes:

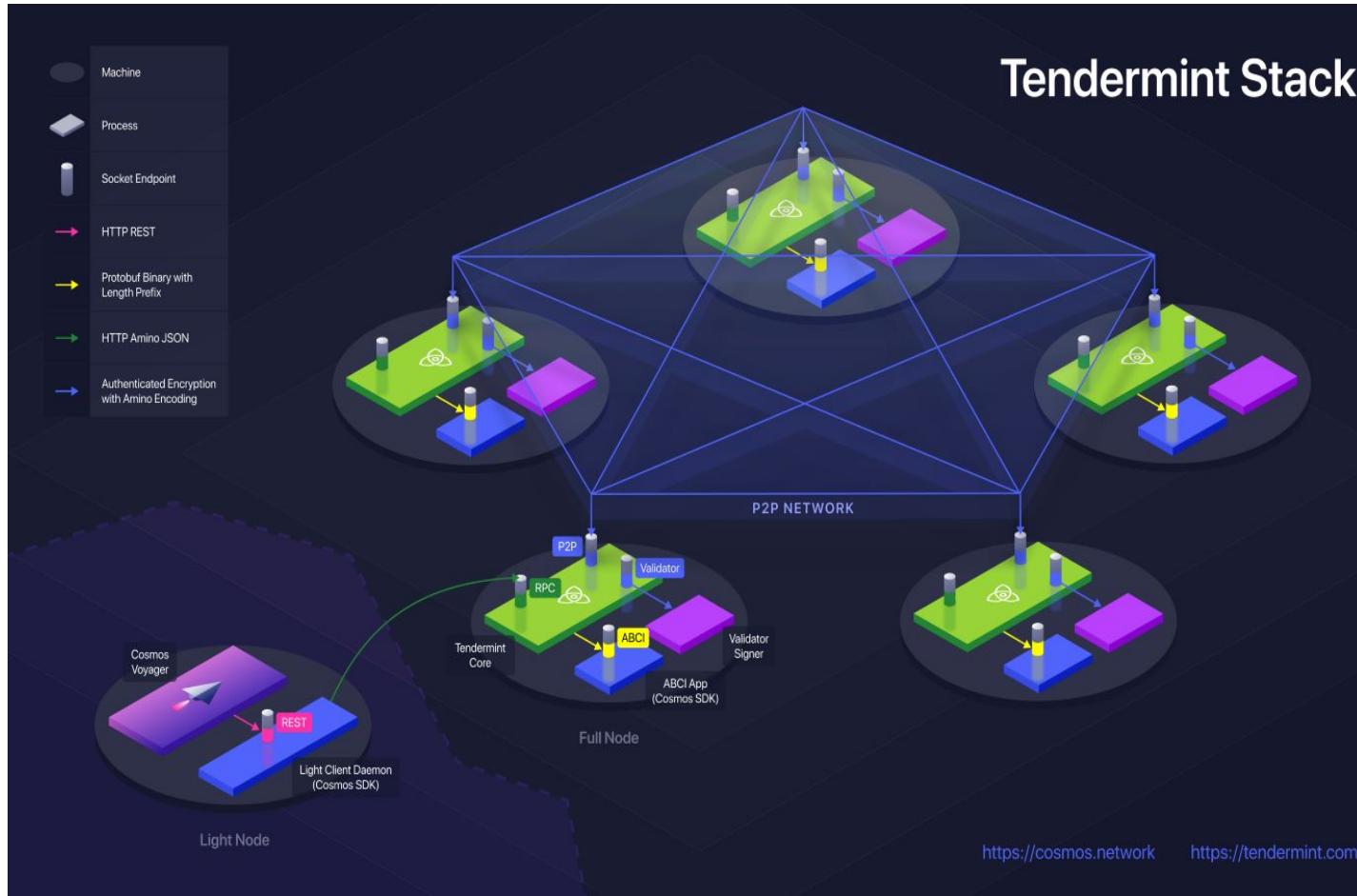
come to consensus as validators and store the state of blockchain

Sentry nodes:

connected to private peers, provide robust security to validators

Seed nodes:

used to provide a list of peers that additional nodes within the network can connect to via seeding.



What is Cosmos SDK

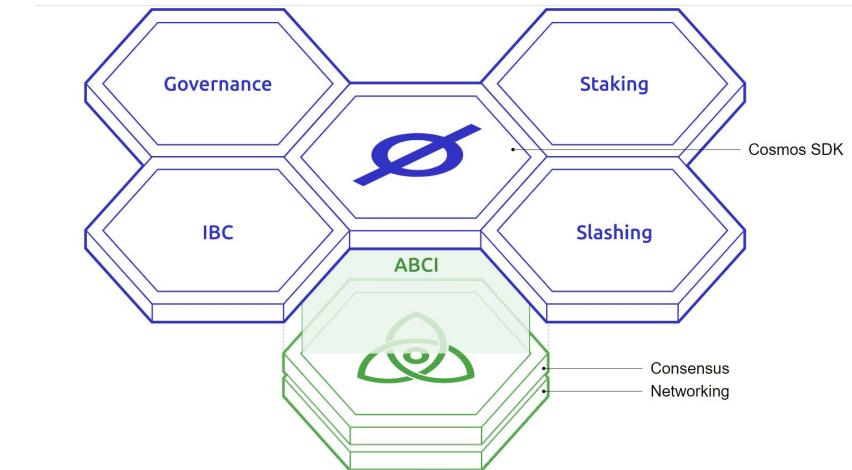
It's an open-source framework that aims to build multi-asset public PoS blockchains.

It is a **npm-like** framework to build secure blockchain applications on top of Tendermint.

SDK-based blockchains are built out of **composable modules**, most of which are open-source and readily available for any developers to use. Allows developers to use predefined or custom modules reducing complexities in building ABCI.

Developers define transaction types, state transition functions, states of app.

Anyone can create a module for the Cosmos SDK.



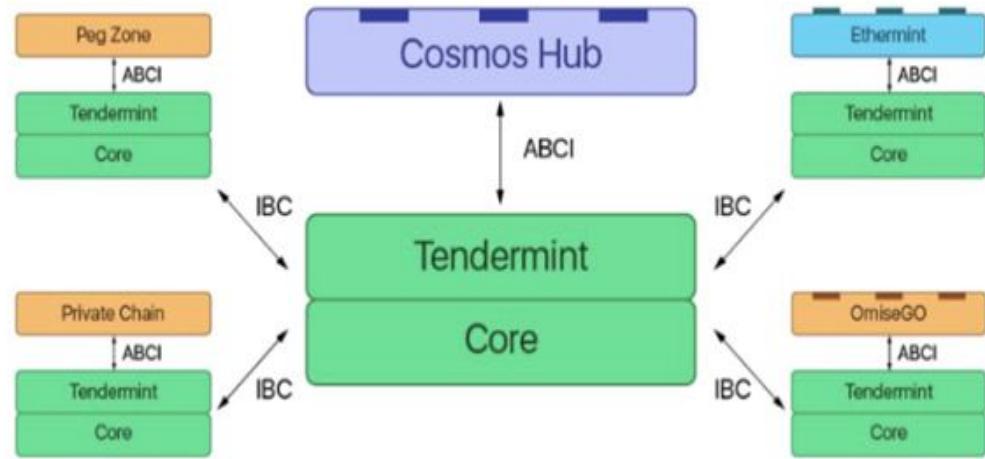
Cosmos Hub

Decentralized network implemented with Cosmos SDK in which each blockchain is powered by consensus algorithm and interconnected by Cosmos network.

Communication network between Cosmos Hub and other blockchain implemented with Cosmos SDK is created via inter-blockchain communication protocol (**IBC**).

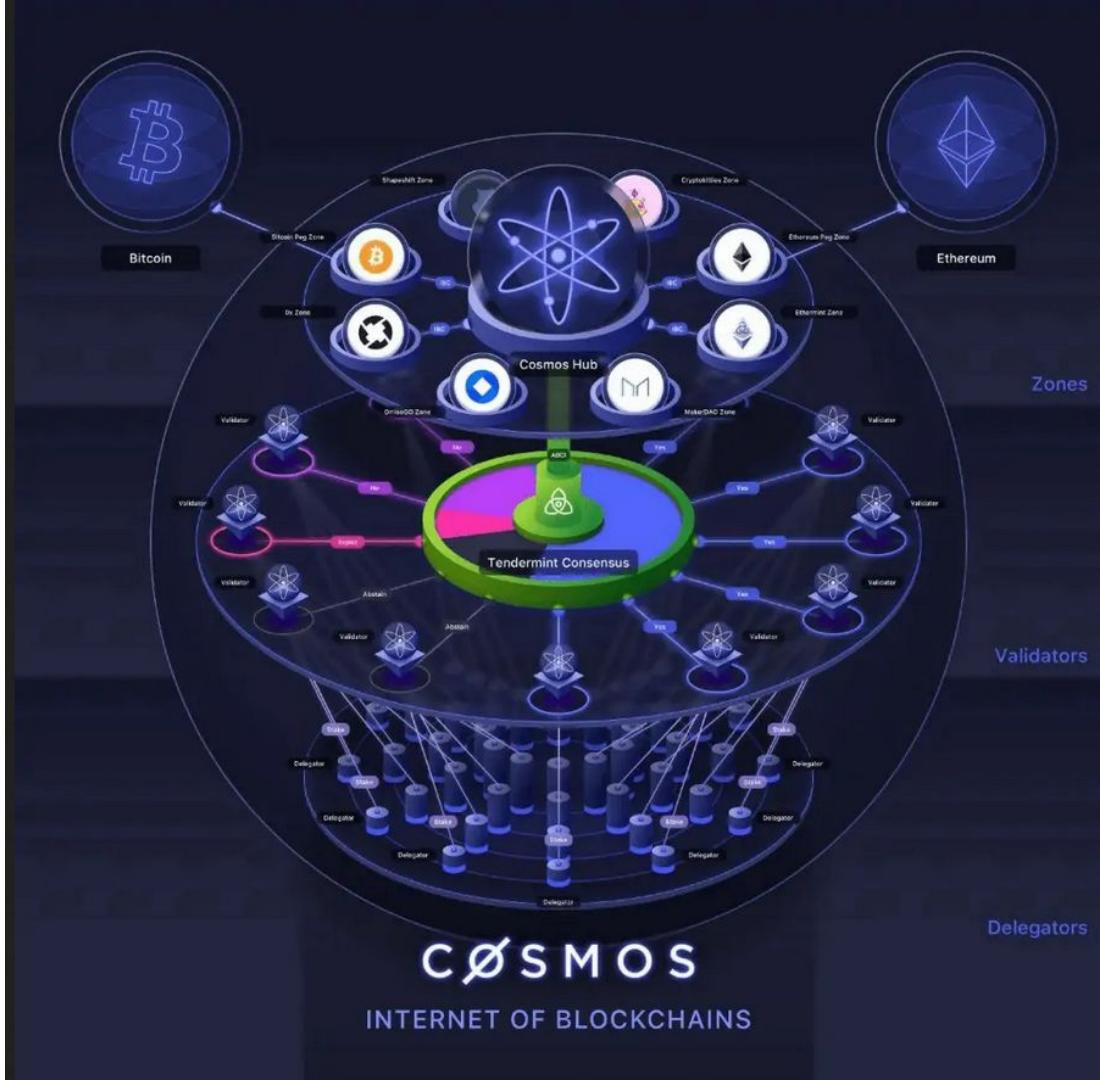
Cosmos network characteristics:

- interoperability
- scalability
- decentralization



Internet of Blockchains

A network of blockchains able to communicate with each other in a decentralized way.



App-specific Cosmos SDK blockchains

Custom Blockchains:

-  BNB Chain
-  Crypto.com Chain
-  Secret Network
-  Evmos

Finance:

-  Osmosis
-  Kujira

Microservices:

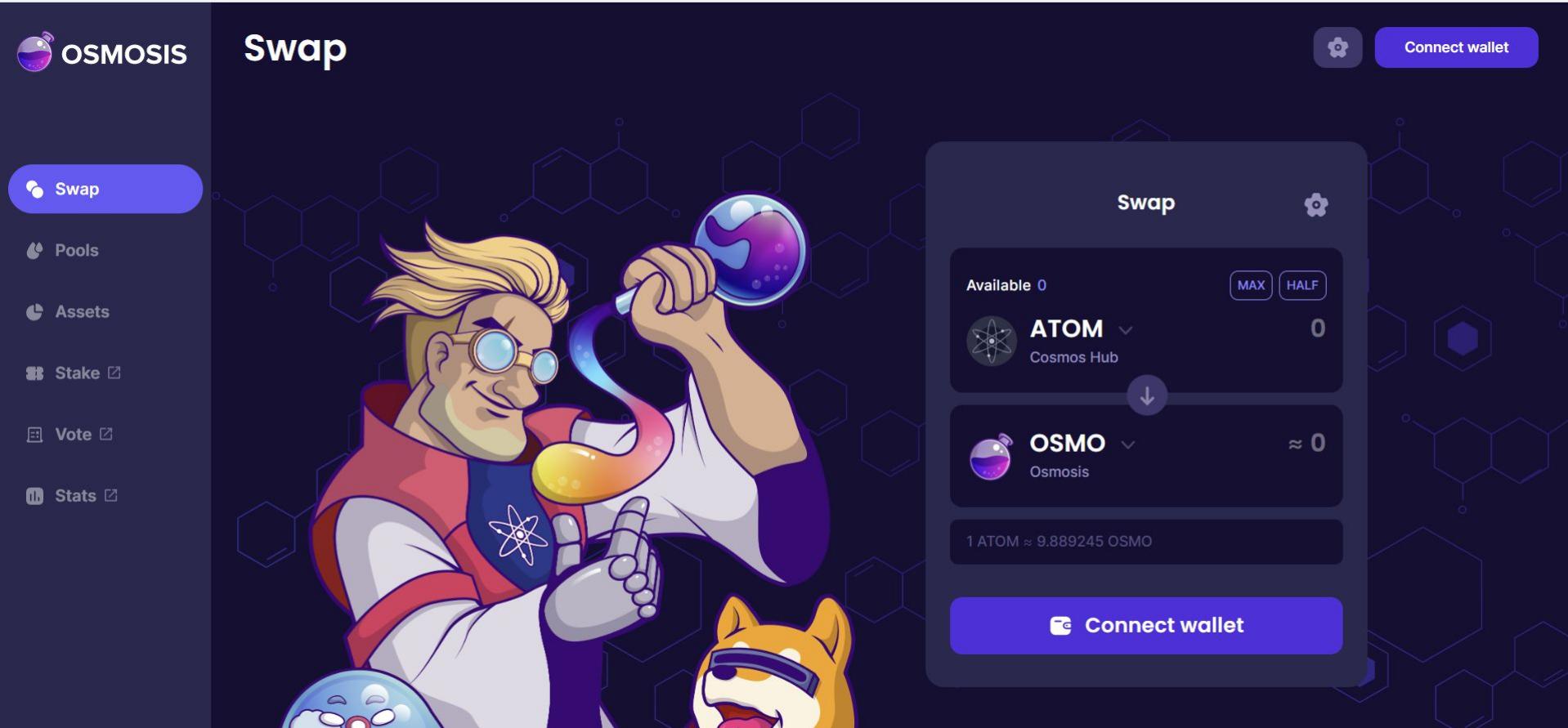
-  Akash Network

Infrastructure:

-  Nomic BTC
-  Band Protocol

Osmosis

<https://app.osmosis.zone/>

The image shows the Osmosis Swap interface. On the left, a vertical sidebar has 'Swap' selected. The main area features a cartoon scientist with blonde hair and blue goggles, holding a test tube with a purple liquid. A small white dog wearing a VR headset is at the bottom. The background is dark with floating chemical structures. A central modal window titled 'Swap' shows two input fields: 'ATOM' (Cosmos Hub) with 'Available 0' and 'OSMO' (Osmosis) with '≈ 0'. Below them is the exchange rate '1 ATOM ≈ 9.889245 OSMO'. At the bottom of the modal is a 'Connect wallet' button.

OSMOSIS

Swap

Pools

Assets

Stake

Vote

Stats

Swap

Swap

Available 0

ATOM

Cosmos Hub

0

MAX HALF

OSMO

Osmosis

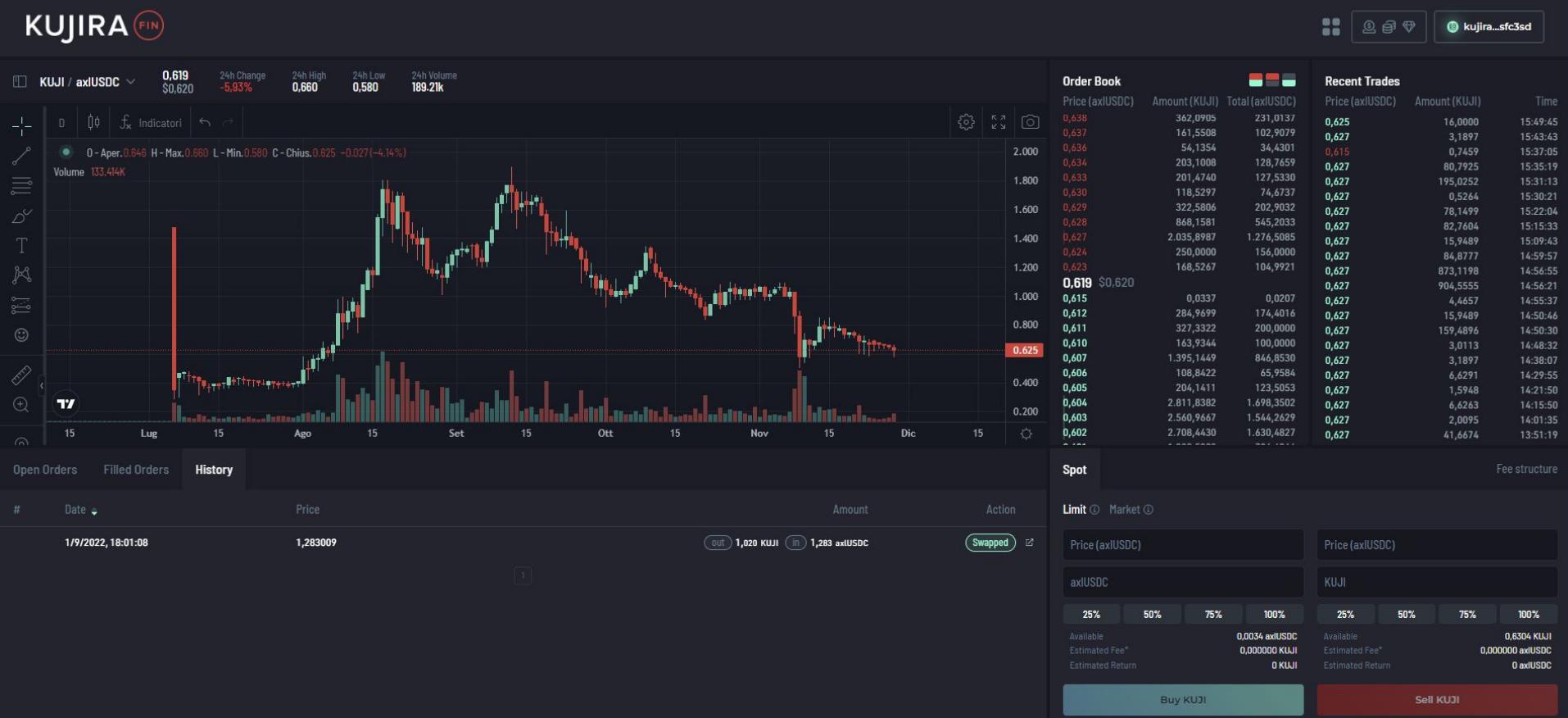
≈ 0

1 ATOM ≈ 9.889245 OSMO

Connect wallet

Kujira Fin

<https://fin.kujira.app/>



SPECS	AKASH	AWS	GCP	AZURE		
vCPUs 1	Memory 1 GB	Storage 1 GB	5,83 USD -83,85%	32,82 USD	36,14 USD	39,37 USD
1 vCPU	1 GB RAM	1 GB Disk	5,83 USD -83,85%	32,82 USD	36,14 USD	39,37 USD
1 vCPU	2 GB RAM	1 GB Disk	7,08 USD -82,16%	36,06 USD	39,74 USD	43,26 USD
2 vCPU	4 GB RAM	1 GB Disk	14,08 USD -82,26%	72,13 USD	79,44 USD	86,52 USD
2 vCPU	8 GB RAM	1 GB Disk	19,08 USD -79,63%	85,12 USD	93,82 USD	102,08 USD
2 vCPU	16 GB RAM	1 GB Disk	29,08 USD -76,22%	111,09 USD	122,59 USD	133,21 USD



ENTER THE APP

THE PROCESS

How it **Works**

Nomic makes it easy to use your Bitcoin on any
IBC-enabled chain



01

Deposit BTC

Send BTC to your deposit address. nBTC will show up in your wallet.



02

Use nBTC in DeFi Protocols

Earn yield on your nBTC on Osmosis, The Cosmos Hub, and more.



03

Withdraw nBTC for BTC

Transfer your gains back to your Bitcoin wallet.

• Mainly Sends, \$ • Mainly Receives, \$



263 Apps & Services built in Cosmos Ecosystem

→ <https://cosmos.network/ecosystem/apps>

<https://mapofzones.com/>

Damaplay

An appchain built on Cosmos SDK



Cosmos SDK

Cosmos

Main concepts recall

Cosmos SDK

Cosmos SDK is a generalized framework for building secure blockchain applications on Tendermint BFT in Go exploiting ready-made modules easy to import, adapt and use.

SOVEREIGNTY

Cosmos gives developers the ability to build a blockchain tailored to the application. Each chain is managed by its own set of validators so there are no limits to app governance

IBC

is the key of interchain interoperability.

Leverage Tendermint's instant finality to enable value transfer (token transfers) and communication between heterogeneous chains. IBC compliant blockchain make blockchain interoperable.



Main Tools

Ignite CLI

Developer-friendly command-line interface tool for application-specific blockchains that builds on the Tendermint BFT and Cosmos SDK.

Creates a modular blockchain written in Go with a single command. Enable inter-chain token transfers using an integrated IBC relayer to send value and data to different chains.

KEPLR

Web wallet of the ecosystem browser extension to interact with blockchain-based Dapps especially developed with Cosmos.

Allows you to manage your cryptocurrencies and tx without having to entrust your private keys to third parties.

It shows your first public address in the form “cosmos1...”



Application Layer - How it works

It is made by integrating composition of modules.

CometBFT passes confirmed tx to ABCI that sends it to APP layer.

The App is a state machine following state transition process.

The transition function is the tx.

Given an initial state, a confirmed transaction, and a set of rules, the machine passes to a new state.

Blockchain are distribute and transactions arrive in batches called block.

Transactions uses several key methods and message types:

- **CheckTx**

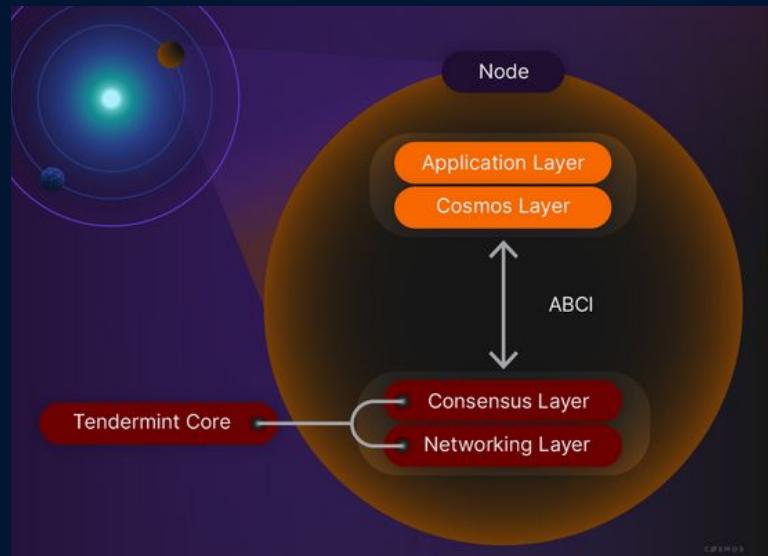
CometBFT uses this method to ask the APP layer if the tx is valid.

- **DeliverTx**

CometBFT calls the DeliverTx method to pass block information to the APP layer for interpretation and state machine transition.

- **BeginBlock, EndBlock**

BeginBlock and EndBlock messages are sent via ABCI even if the blocks contain no transactions.



Application Layer - Concepts

Account

Made by private and public key.

Keys are stored and managed in an object called keyring.

It can manage multiple accounts.

Messages

Each tx contains ≥ 1 messages.

Messages trigger state changes within a module.

Module processes the messages after tx is included into a block by consensus layer.

Modules

Modules can be considered state machines within the larger state machine.

BaseApp decomposes the Messages and routes messages to the appropriate module for processing.

TxBuilder is the interface for the tx object.

- **Msgs** : the array of messages included in the transaction.
- **GasLimit** : chosen by the users for the gas amount they are willing to spend.
- **Memo** : a note or comment.
- **FeeAmount** : the maximum amount the user is willing to pay in fees.
- **TimeoutHeight** : the block height until which the transaction is valid.
- **Signatures** : the array of signatures from all signers of the transaction.

Damaplay

functionalities

Game State Object

- Black player
- Red player
- Board
- Player to play next
- Wager
- Winner
- Unique incrementing ID

Deadline and Auto Expiring game

Games are deleted after the expiring date.
If it is ongoing, all the data are deleted.
If it is ended, the board state is deleted and
winner data is kept.

Create a game

- Validate input
- creates a brand new game
- saves it in storage and returns the ID

Make a move

- Validate input
- Perform the move
- Update the game state

Record Winner

Check for a game winner.

Damaplay

functionalities

Check correct move

To query the validity of a move you need to pass:

- Game ID
- Player color
- Origin board position X and Y
- Target board position X and Y

The information returned is a boolean for whether the move is valid or an error which explains why the move is not valid.

Wager and wager payment

Players choose to wager money or not, and the winner gets it. The forfeiture loses their wager.

Utilizes the bank module.

An external module needs to call the bank module to tell it to move tokens.

FIFO game order

A FIFO data structure to sort games from the least recently played at the top of the list to the most recently played at the bottom, in order to help identify inactive games which may become candidates for forced termination, which reduces undesirable and wasteful data stored on the blockchain.

Playing as Alice and Bob from terminal

- Initialize the blockchain.

Alice and Bob accounts are generated for test purpose.

```
amer@amer-HP-P:~/checkers$ sudo docker exec -it checkers ignite chain serve
Cosmos SDK's version is: stargate - v0.45.4
⠼ Building proto...
⠼ Installing dependencies...
⠼ Building the blockchain...
⠼ Initializing the app...
⠼ Created account "alice" with address "cosmos1kpj8rfv3rgsd0rnw4l50540edfewqj5hnkyxhy" with mnemonic: "success trigger evil fold enroll tennis pluck march afford mystery citizen sketch glow river frozen cabbage unfold deer where hour moment hen opinion"
⠼ Created account "bob" with address "cosmos12flwfhkhgx8ypqn2zc7598yr8kdxu77v54u99" with mnemonic: "peanut game bacon tissue essence blind federal nation dog crumble decline try multiply runway garden pool east obtain buddy identify note focus else vintage"
⠼ Tendermint node: http://0.0.0.0:26657
⠼ Blockchain API: http://0.0.0.0:1317
⠼ Token faucet: http://0.0.0.0:4500
```

- Create a match Alice vs Bob with 1 token as wager.

The black player (Alice) signs the transaction.

```
amer@amer-HP-P:~/checkers$ sudo docker exec -it checkers checkersd tx checkers create-game $alice $bob 1 token --from $alice
{"body": [{"messages": [{"@type": "/alice.checkers.checkers.MsgCreateGame", "creator": "cosmos1djhhgkwucvaqve8gqekrrx2pm9jhs1hm4nefvp", "black": "cosmos1djhhgkwucvaqve8gqekrrx2pm9jhs1hm4nefvp", "red": "cosmos1ye2zr9plgm005fvgttd8n056shfkhs5mg8c4ml2l", "wager": "1", "denom": "token"}, {"memo": "", "timeout_height": "0", "extension_options": {}, "non_critical_extension_options": {}, "auth_info": {"signer_infos": [], "fee": {"amount": [], "gas_limit": "200000", "payer": "", "granter": ""}}, "signatures": []}], "signatures": []}
confirm transaction before signing and broadcasting [y/N]: 
```

- Show the details of the created game (id = 1).

```
storedGame:
  Winner: '*'
  afterIndex: "2"
  beforeIndex: "-1"
  black: cosmos1djhhgkwucvaqve8gqekrrx2pm9jhs1hm4nefvp
  board: '*b*b*b*b|b*b*b*b|*b*b*b*b|*****|*****|r*r*r*r|r*r*r*r|r*r*r*r'
  deadline: 2023-06-02 18:13:47.958028924 +0000 UTC
  denom: token
  index: "1"
  moveCount: "0"
  red: cosmos1ye2zr9plgm005fvgttd8n056shfkhs5mg8c4ml2l
  turn: b
  wager: "1"
```

Playing as Alice and Bob from terminal

- Check player (Alice) credit.

```
amer@amer-HP-P:~$ sudo docker exec -it checkers checkersd query bank balances $alice
balances:
- amount: "100000000"
  denom: stake
- amount: "20000"
  denom: token
pagination:
  next_key: null
  total: "0"
```

- Print the board with the positions of red and black pawns.

```
amer@amer-HP-P:~$ sudo docker exec -it checkers bash -c "checkersd query checkers show-stored-game 1 --output json | jq \".storedGame.board\" | sed 's/\//\n/g' | sed 's/|/\n/g'"
*b*b*b*b
b*b*b*b*
*b*b*b*b
***** 
***** 
r*r*r*r*
*r*r*r*r
r*r*r*r*
```

Playing as Alice and Bob from terminal

- Print of the list of active matches.

```
amer@amer-HP-P:~$ sudo docker exec -it checkers checkersd query checkers list-stored-game
{
  pagination: {
    next_key: null,
    total: "0"
  },
  storedGame: [
    {
      - Winner: "*",
        afterIndex: "2",
        beforeIndex: "-1",
        black: "cosmos1djhhgkwucvaqve8gqeckrrx2pm9jhsllhm4nefvp",
        board: "*b*b*b*b|b*b*b*b*|*b*b*b*b|*****|*****|r*r*r*r*|*r*r*r*r|r*r*r*r*r*",
        deadline: "2023-06-02 18:13:47.958028924 +0000 UTC",
        denom: "token",
        index: "1",
        moveCount: "0",
        red: "cosmos1ye2zr9plgm005fvgt8n056shfkh5mg8c4ml2l",
        turn: "b",
        wager: "1"
    },
    {
      - Winner: "*",
        afterIndex: "-1",
        beforeIndex: "1",
        black: "cosmos1djhhgkwucvaqve8gqeckrrx2pm9jhsllhm4nefvp",
        board: "*b*b*b*b|b*b*b*b*|*b*b*b*b|*****|*****|r*r*r*r*|*r*r*r*r|r*r*r*r*r*",
        deadline: "2023-06-02 18:24:09.686280252 +0000 UTC",
        denom: "token",
        index: "2",
        moveCount: "0",
        red: "cosmos1ye2zr9plgm005fvgt8n056shfkh5mg8c4ml2l",
        turn: "b",
        wager: "1"
    }
  ]
}
```

Playing as Alice and Bob from terminal

- Make a move as Alice.

```
amer@amer-HP-P:~$ sudo docker exec -it checkers checkersd tx checkers play-move 1 1 2 2 3 --from $alice
{"body": {"messages": [{"@type": "/alice.checkers.checkers.MsgPlayMove", "creator": "cosmos1djhhgkwucvaqv8gqekrrx2pm9jhshlm4nefvp", "gameIndex": "1", "fromX": "1", "fromY": "2", "toX": "2", "toY": "3"}], "memo": "", "timeout_height": "0", "extension_options": [], "non_critical_extension_options": [], "auth_info": {"signer_infos": [], "fee": {"amount": [], "gas_limit": "200000", "payer": "", "granter": ""}}, "signatures": []}
confirm transaction before signing and broadcasting [y/N]: y
```

- Visualize the winner of the match.

It returns **null** because the match is ongoing.

```
amer@amer-HP-P:~$ sudo docker exec -it checkers bash -c "checkersd query checkers show-stored-game 1 --output json | jq '.storedGame.winner'"
null
```

Tx inside the Block

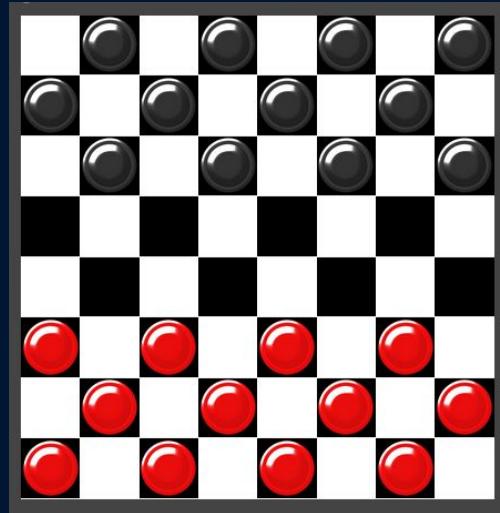
- **Header**
containing the hash of previous block and the hash of the all hash transactions.

- **Data**
depending by commands when creating the tx.

Height variable gives the position of the block into the blockchain.

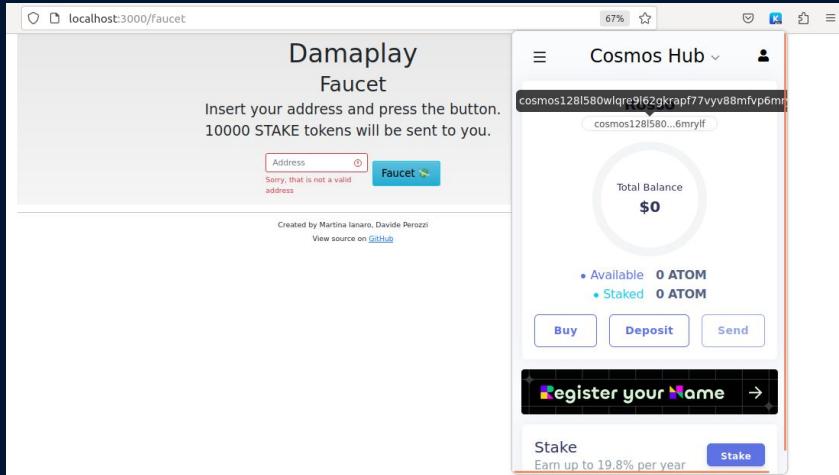
```
amer@amer-HP-P:~$ curl localhost:26657/block?height=13106
{
  "jsonrpc": "2.0",
  "id": -1,
  "result": {
    "block_id": {
      "hash": "5A7048E056B415F7C116C10CC224BC683B37AE29795B6A33B1AB7CEFEAF28963",
      "parts": {
        "total": 1,
        "hash": "6361E70E429174330686244E803A364772006EB3C0A30B160E4F7E98EB402DE9"
      }
    },
    "block": {
      "header": {
        "version": {
          "block": "11"
        },
        "chain_id": "checkers",
        "height": "13106",
        "time": "2023-06-01T18:25:09.582095978Z",
        "last_block_id": {
          "hash": "8ACD73BF7D1A0B6AEF3DBA01D2FF6CE222906AB9ADB3F626468A3AF5AC687173",
          "parts": {
            "total": 1,
            "hash": "EF6135DCCB3A071E9D1E24A266B9EA11CF91CBB131DBC1D6767CA82711322083"
          }
        },
        "last_commit_hash": "547782D5E893CBFEA73154E2C170ACF44B0AC740119EF4EE2C879D3CCF74891D",
        "data_hash": "06C09125E2937D801284B75C69CE4AD10D5C1FC175472B44FB7F704AO0C02E1F",
        "validators_hash": "D9889BD25452CC975619481907CB6DF4768170A363453624353F1948604103",
        "nextValidatorsHash": "D9889BD25452CC975619481907CB6DF4768170A363453624353F1948604103",
        "consensus_hash": "048091BC7DC283F778FBF91D73C44DA58C3DF8A9C867405D887F3DAA22F",
        "app_hash": "41B1ACEC7C2E7C3664C775F4B0363999202F19975A33F2A6E7C3A734EA116",
        "last_results_hash": "E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855",
        "evidence_hash": "E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855",
        "proposer_address": "6925493F3E92C8962371987B3A5xpEc40A40F28B6"
      },
      "data": {
        "txs": [
          "cnQKgokL2FsawN1LmNoZwNrZXJzLnNoZwNrZXJzLk1zBxsYXlnb3zLejoKLWNv21vczFkamhoZz3dWn2YXF2zThncVrcn34MnBt0Wpc2xbTruZWZcB1BMRgBIA
          jAEIgkUApGCh8vY2z0WqzLmNvexB0bz5zZNNWjU2azEuUHLS2V5E1MKrO0nTVlpC9d1rqjPUD9NlR0rxJKV1Ea0Y3557rgqvWRIEcgIIArgDgQ0JoMGkCanSf8h8ZwU906
          ieZckAb6cd0Fe3lsz0rrPqVovJalp9/oSPRg6TA5xpEc40A40F28B6Q0"
        ]
      },
      "evidence": {
        "evidence": []
      },
      "last_commit": {
        "height": "13105",
        "round": 0,
        "block_id": {
          "hash": "8ACD73BF7D1A0B6AEF3DBA01D2FF6CE222906AB9ADB3F626468A3AF5AC687173",
          "parts": {
            "total": 1,
            "hash": "EF6135DCCB3A071E9D1E24A266B9EA11CF91CBB131DBC1D6767CA82711322083"
          }
        },
        "signatures": [
          {
            "block_id_flag": 2,
            "validator_address": "6925493F3E92C8962371987B3D9EC40A40F28B6",
            "timestamp": "2023-06-01T18:25:09.582095978Z",
            "signature": "6zfbrf0j2DufKap2yrm/aRCJuVddfuSAoabY6DUai090B5bL903VDWATkBlyWC/kwZRB1qxprqzXWarI7EA=="
          }
        ]
      }
    }
  }
}
```

GUI demo

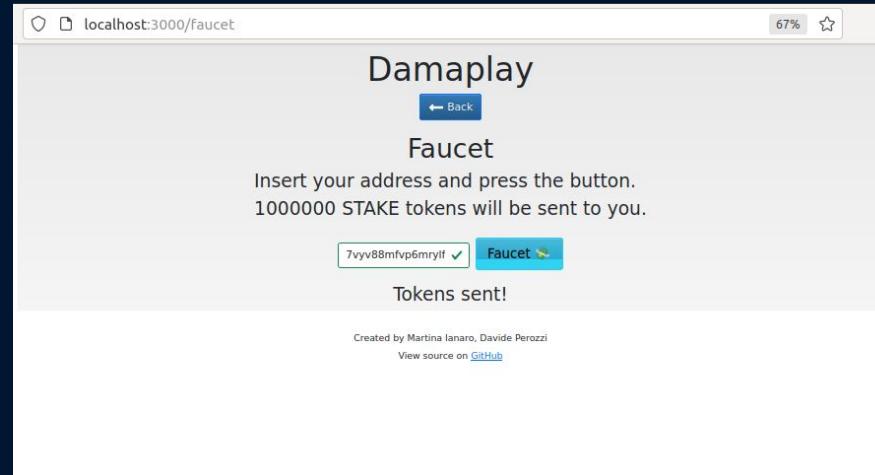


<http://localhost:3000/menu>

Receive STAKE tokens from Faucet



Copy and insert your address in the input



Press the button and wait for the confirmation message.

Also possible from the CLI

```
amer@amer-HP-P:~$ sudo docker exec -it checkers checkersd query bank balances co
smos128l580wlqre9l62gkrapf77vyv88mfvp6mrylf
balances:
- amount: "1000000"
  denom: stake
  pagination:
    next_key: null
    total: "0"
```

Create A New Game

Damaplay

You have no saved games

Created by Martina Ianaro, Davide Perozzi
View source on [GitHub](#)

Create A New Game

Set the addresses of the players

Player 1(Black) ✓

Player 2(Red) ✓

Game wager:

Play Game! Cancel

Insert your address as Player 1 (BLACK) and the opponent as Player 2 (RED)

Damaplay

Updated list of active games

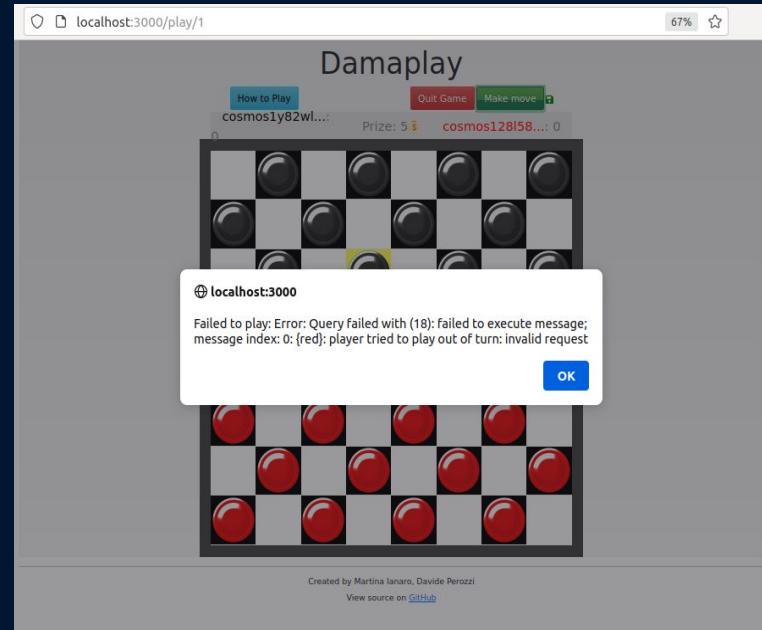
cosmos1y82wlu6ph9na4yfzgx2aanjl625shcvdh4s8la vs
cosmos1281580wlqre9l62gkrapf77vyv88mfvp6mrylf

Score: 0:0

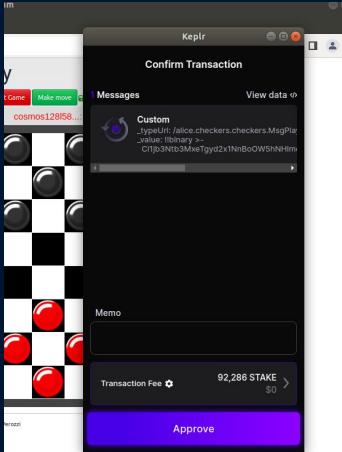
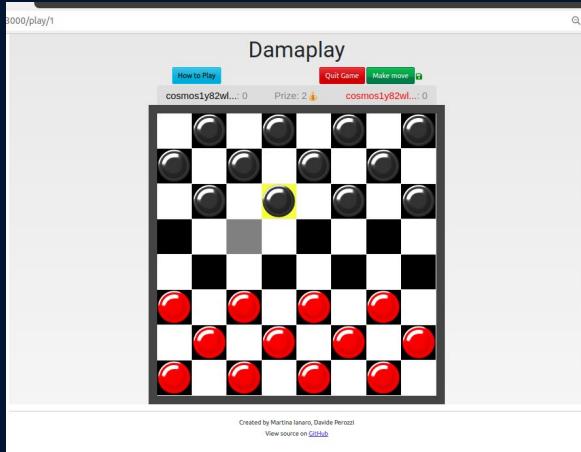
Created: 06/05/23 at 06:35pm

Created by Martina Ianaro, Davide Perozzi
View source on [GitHub](#)

Trying to make a move out of your turn

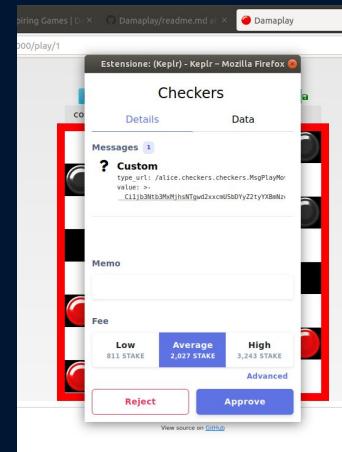
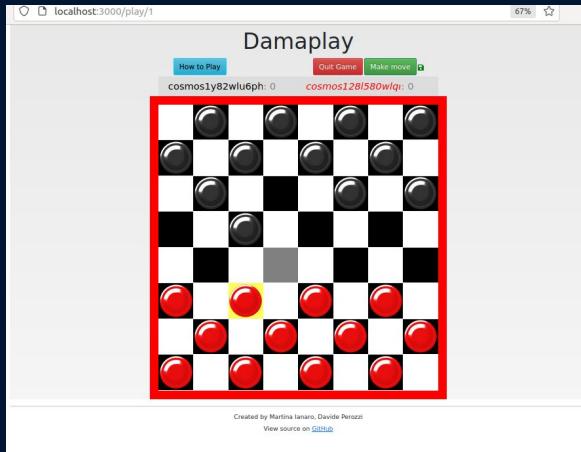


If you try to make a move when it's not your turn, you get a warning.



Making a move

- Select a pawn and the desired cell destination
- Click on Make move
- Firm transaction with Keplr



Board printed via CLI

```
amer@amer-HP-P:~$ sudo docker exec -it checkers bash -c "checkersd query checkers show-stored-game 1 --output json | jq \".\".storedGame.board\" | sed 's/'//g' | sed 's/|/\\n/g'"  
*b*b*b*b  
b*b*b*b  
*b***b*b  
**b*****  
***r****  
r****r*r*  
*r*r*r*r  
r****r*r*
```

Github repo

- ❖ <https://github.com/martinianaro99/Damaplay>

References

- ❖ <https://docs.tendermint.com/v0.34/introduction/what-is-tendermint.html>
- ❖ <https://tutorials.cosmos.network/academy/1-what-is-cosmos/>
- ❖ <https://docs.cosmos.network/main>
- ❖ https://en.wikipedia.org/wiki/Byzantine_fault
- ❖ <https://blog.cosmos.network/why-blockchains-need-cosmos-proof-of-stake-for-a-sustainable-environment-878b3edd2e85>
- ❖ <https://learn.bybit.com/blockchain/tendermint/>

References

- ❖ <https://cosmos.network/learn/staking/>
- ❖ <https://www.gemini.com/cryptopedia/cosmos-ignite-crypto-consensus-tendermint-ecosystem>
- ❖ <https://academy.binance.com/it/articles/tendermint-explained>
- ❖ <https://www.coinbase.com/learn/crypto-basics/what-is-proof-of-work-or-proof-of-stake>
- ❖ https://www.blockchain4innovation.it/news/cosmos-il-network-che-facilita-linteroperabilita-tra-blockchain/#Quali_sono_i_vantaggi_di_Cosmos
- ❖ <https://docs.tendermint.com/v0.33/app-dev/app-development.html>

THANK YOU FOR YOUR ATTENTION