Ethermint 2.0
Cosmos SDK + EVM

Christopher Goes

Software Engineer at Cosmos/Tendermint

In This Presentation

Quick overview of Cosmos

- Features of the Cosmos SDK
- Background on Tendermint consensus

All about Ethermint 2.0

- Architecture outline
- EVM + SDK integration
- DApps on Ethermint
- Q&A

The Power of the Cosmos SDK

Framework for building state machines

Developer Friendly

- Written in Golang
- “Ruby-on-Rails for blockchains”
- Open-source on Github

Secure

- Modules for areas of functionality
- Least-authority permissioning

The Power of the Cosmos SDK

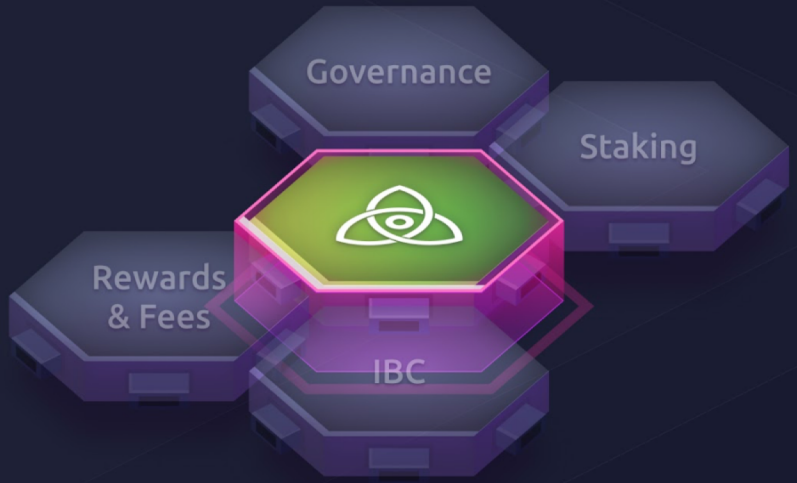
Modular & Extensible

- Modular architecture for plug-and-play development
- Glue together existing modules to build your blockchain
- Share modules you build to contribute downstream

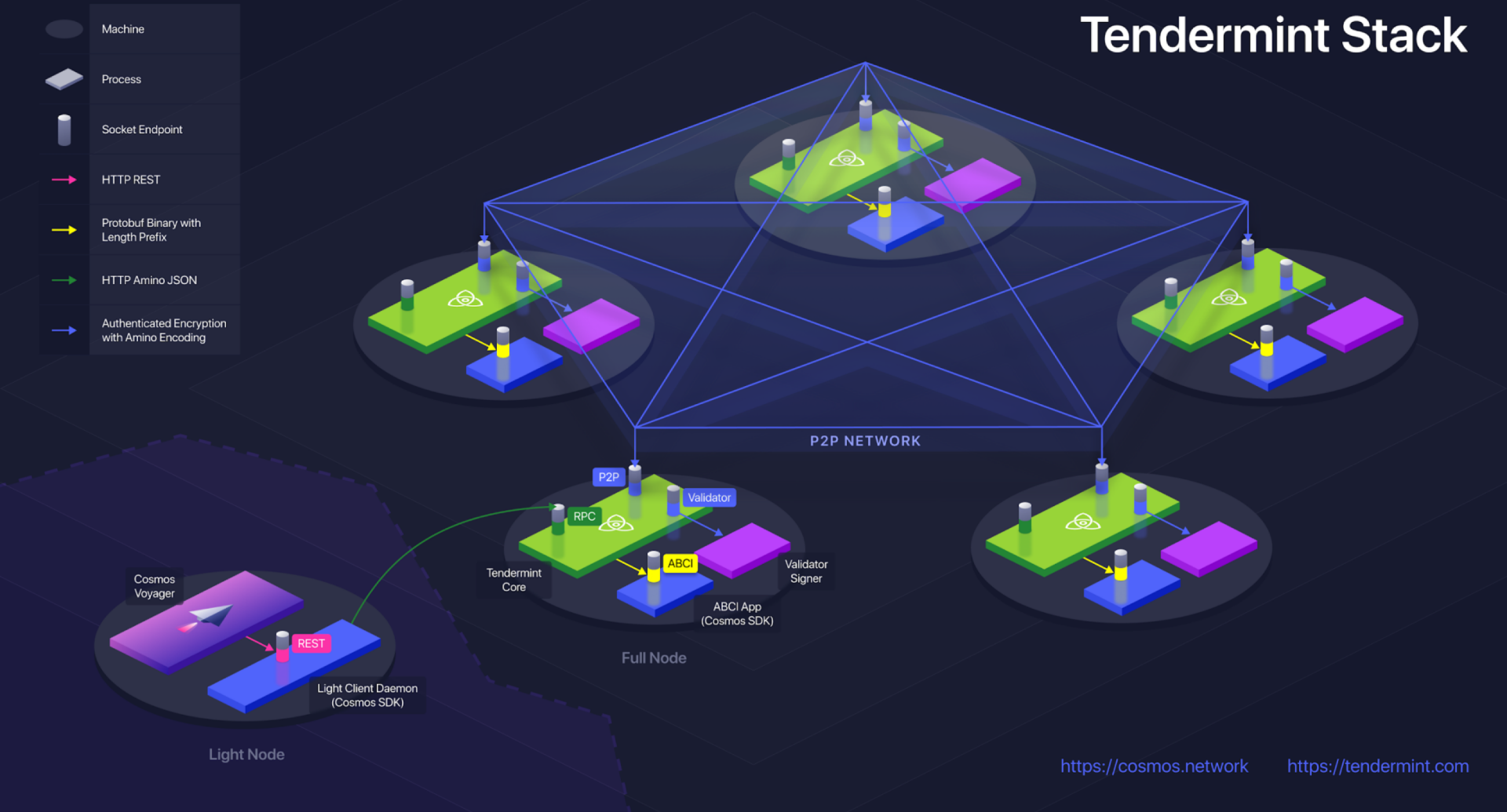
Core Module: Tendermint Consensus

State-of-the-art generic BFT blockchain engine

- Encapsulates consensus & P2P layer
- Vertical layer-1 scaling for your state machine
- One-block finality, peer-reviewed BFT algorithm
- Connects to any state machine



Tendermint Stack



https://cosmos.network https://tendermint.com

Core Module: Bonded Proof-of-Stake

Bonded proof-of-stake (BPoS) as a module

- Use your own staking token
- Delegation with skin in the game
- Addresses “nothing-at-stake” with an unbonding period
- “Batteries included” but extensible

Core Module: Governance

Flexible on-chain governance as a module

- Text proposals (“social consensus”)
- Parameter change proposals
- Software upgrade proposals
- Liquid democracy (delegators/validators)

Core Module: Rewards & Fees

Block rewards and validator fees as a module

- Use your own coin to pay fees when running a transaction - no lock-in.
- Multi-coin support for improved user experience.

Core Module: Inter-blockchain Communication

Inter-blockchain communication as a module

- Move tokens & data between blockchains
- Authenticated exactly-once message delivery
- Basis for horizontal scaling
- “TCP/IP for blockchains”

Module: Ethereum Virtual Machine

EVM in the Cosmos SDK module interface

- Account database, state tree
- EVM module can run Ethereum txs
- EVM module calls into Cosmos SDK modules
- Shared state view - one token

Ethermint: Cosmos SDK + EVM

Best of both worlds

- Scalability of Tendermint
- Power of the Cosmos SDK
- Existing ecosystem of Ethereum contracts, dev tooling

EVM module from TurboGeth

- DB performance improvements
- Flexible SDK module interface

TurboGeth Improvements

Ethermint & Turbo Geth developing in parallel

- Any binary search tree
- Structure-preserving custom serialization
- Same data structure for DB index & Merkle hashing
- Different caching strategies - “benefit of hindsight”

Two Ways to Use EVM Module

Ethermint as a blockchain

- Cosmos PoS chain for smart contracts

EVM as a library

- Deploy your own Cosmos chain with EVM support
- Add in other SDK modules or write your own
- Flexibility in token choice & economic model

Ethermint Hard Spoon

“Hard spoon” for initial staking distribution

- Copy Atom & ETH balances for Photon token
- Shared security with Cosmos Hub
- Approved by Cosmos Hub governance
- Photon may also be a fee token on Hub

Ethermint as a Blockchain

One chain for many EVM applications

- Hard spoon of account balances
- Sovereign chain, own token
- Governance, staking, slashing
- Fully web3 compatible
- IBC connections to other chains

EVM Module for SDK Zones

EVM module for your
Tendermint/Cosmos chain

- Full EVM functionality set
- Include other Cosmos SDK modules
- Utilize existing Solidity contracts
- Gradually port parts of logic to native code

Interoperating with Ethereum

What if you want to move between chains?

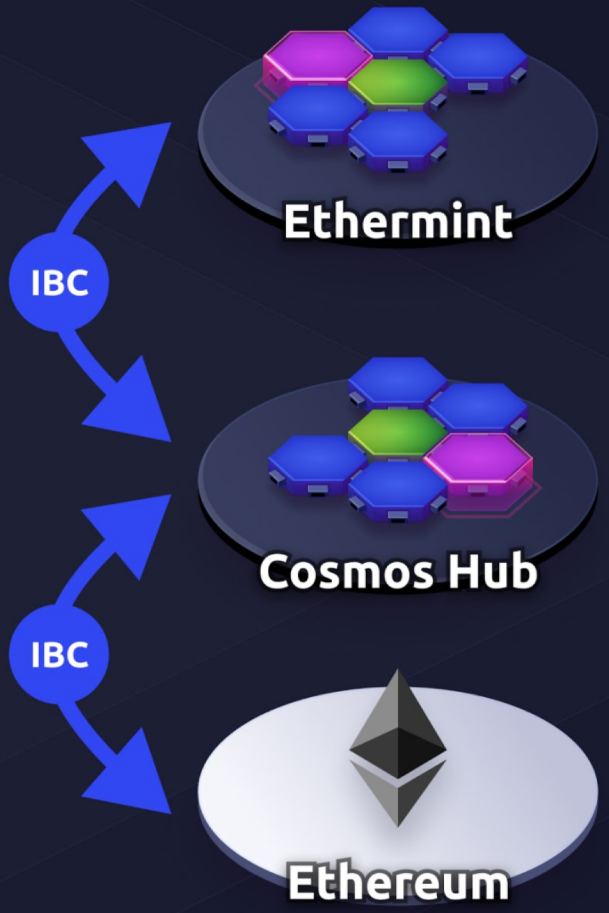
- Trade ERC20 tokens on an Ethermint DEX
- Keep your cold wallet on Ethereum
- Split a Dapp between chains

Necessary: two-way state/asset transfer bridge

IBC Bridge: Prerequisites

Prerequisites

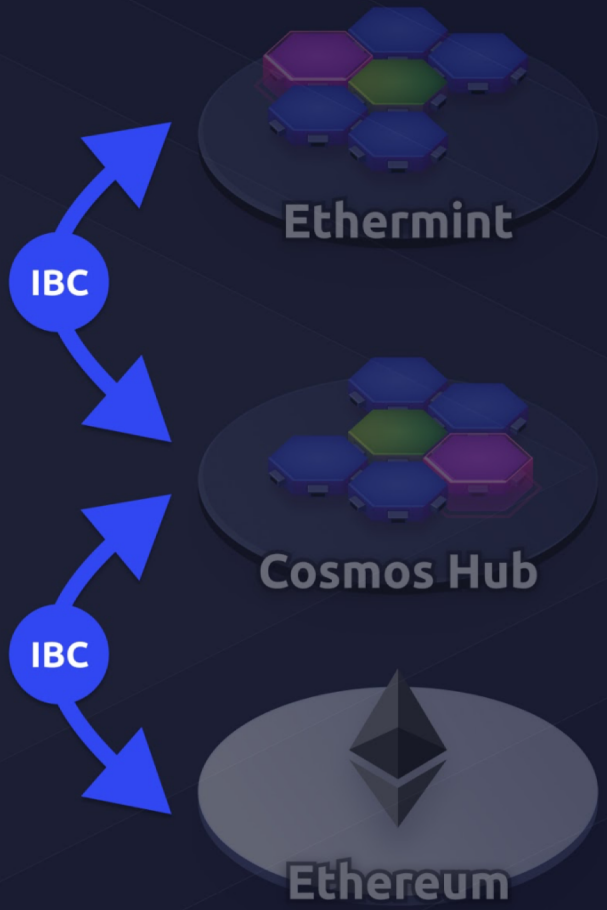
- Functional IBC on Cosmos Hub
- Ethereum main-chain ⇔ Hub IBC
- Hub ⇔ Ethermint zone IBC



IBC Bridge: Features

Ethereum ⇔ Ethermint IBC

- Assets (Ethereum, Photons, ERC20/721) can be transferred chain-to-chain
- Split contract logic between chains
- Take advantage of speed/lower-fees of Ethermint



IBC Bridge: For Existing DApps

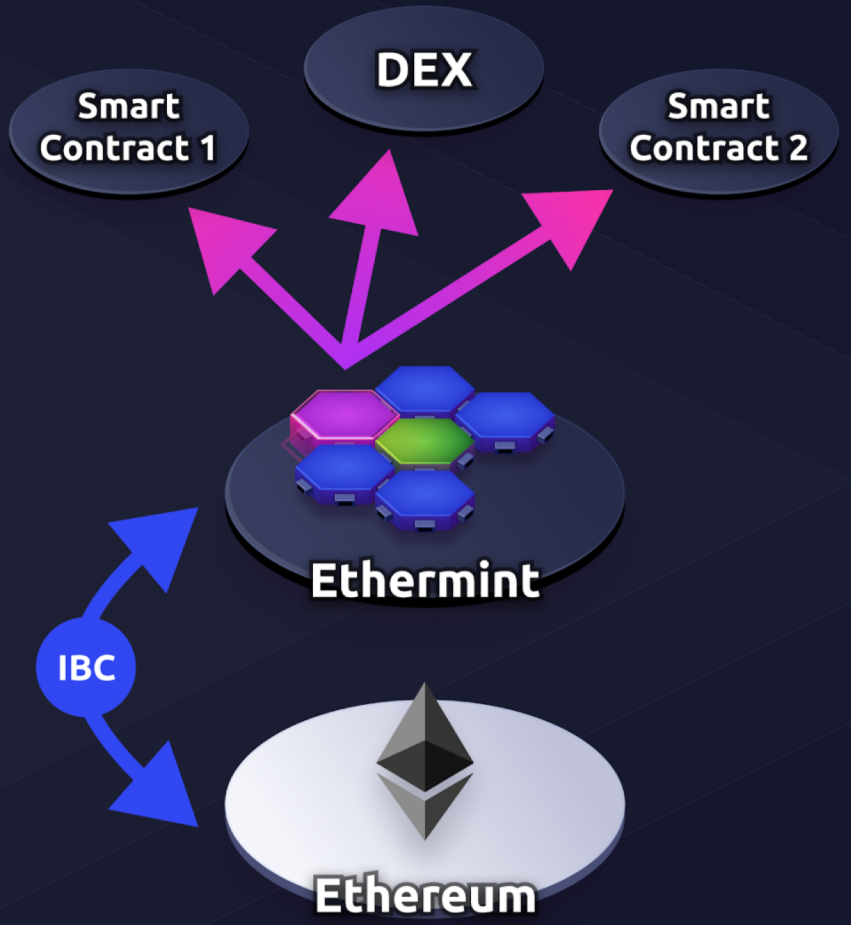
Gradual/partial transition to Ethermint

- Move parts of contract logic over time
- Users transfer assets to Ethermint for expensive transactions
- Users can settle back to Ethereum when desired
- Seamless UX switch

DEX on Ethermint Zone

Ethermint speed + scale, existing ERC20s

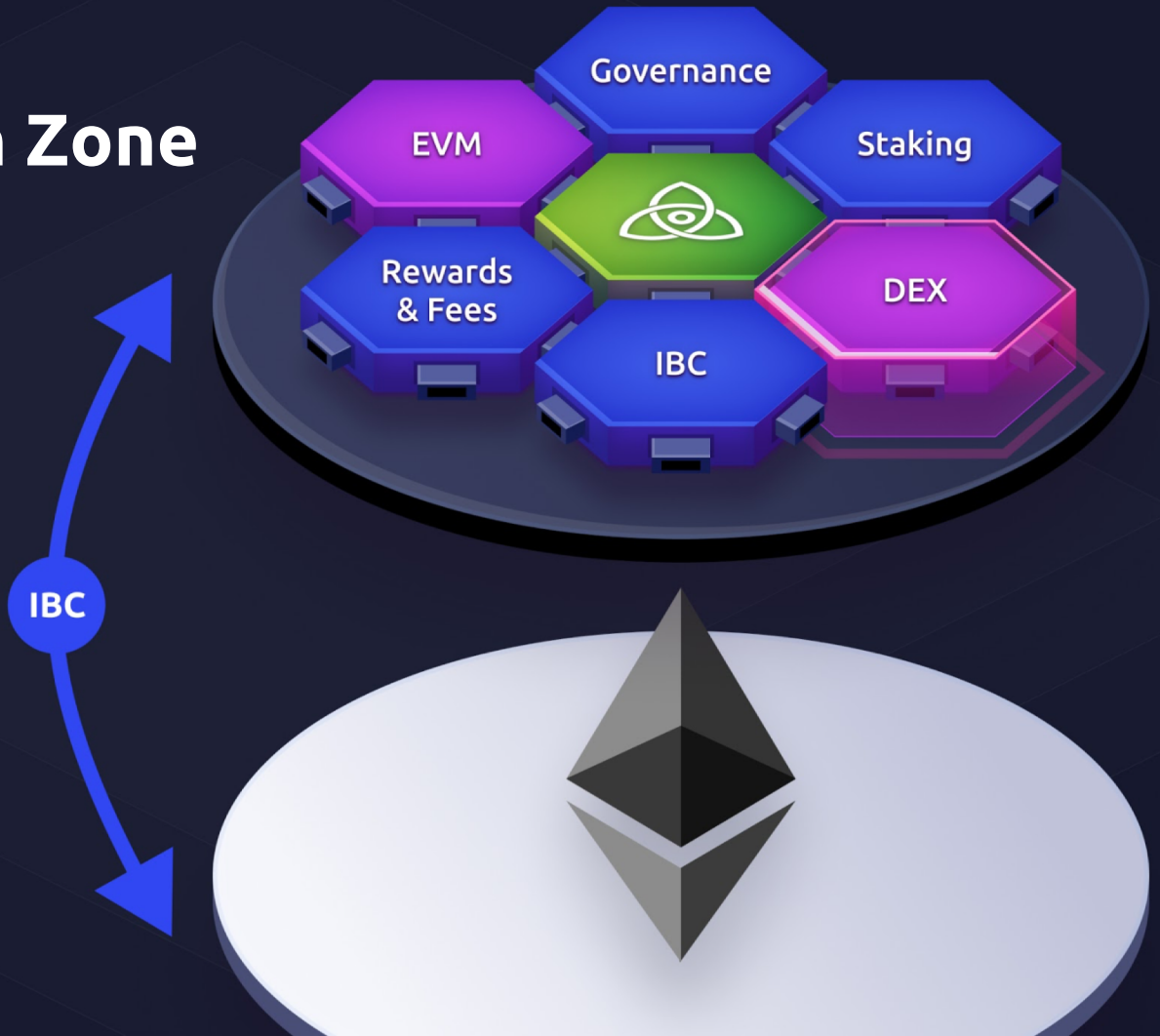
- Deploy protocol (0x, Wyvern) on Ethermint
- Users send tokens over IBC to deposit
- All trades settled on Ethermint
- Withdraw back over IBC to Ethereum



DEX on Sovereign Zone

Advantages of a sovereign chain

- No competition for block space
- Write parts of protocol as native code
- Alter staking mechanics
- Alter state machine



Current State

State of Ethermint Development

- Development in progress: github.com/cosmos/ethermint
- Developer Preview Release - End of Q4
- Play around with smart contracts
- Mostly functional web3 API

Open Questions

Contract balances in PoS

- Delegation / slashing?

Inflation & fee distribution

- No need for PoW rewards
- Experimental monetary policy?

Ethermint ⇔ Ethereum 2.0

- Finality gadget = faster IBC

Q&A

Do the survey for a chance to win a Cosmos hoodie

cosmos.network/devcon4