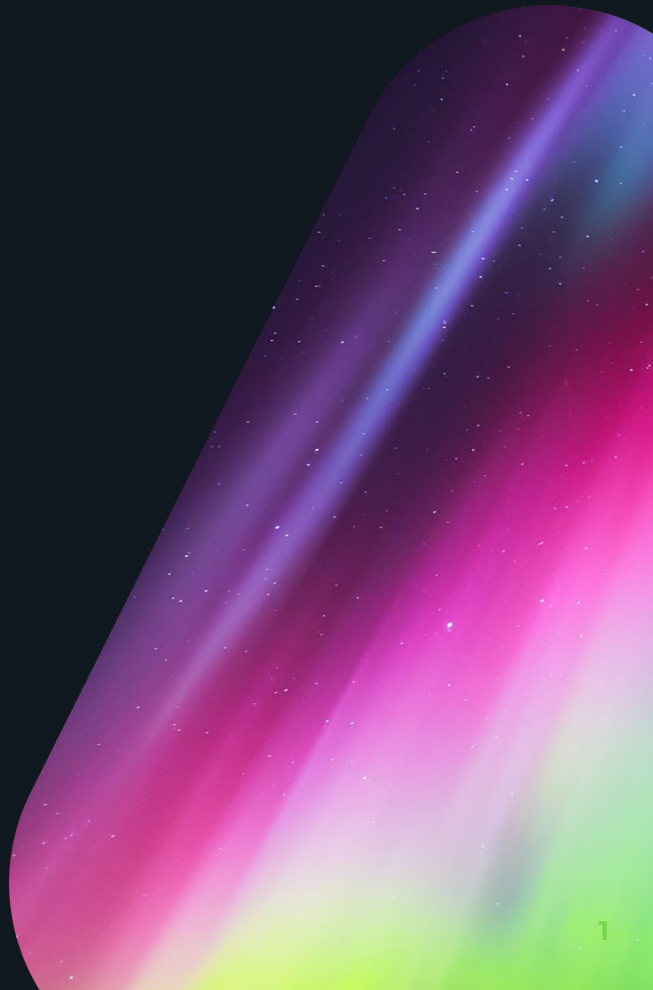
AURORA

The Rainbow Bridge

How to build the ‘right’ bridge

Alex Shevchenko
CEO Aurora Labs

Cyber Academy Meetup,
August 12th 2021



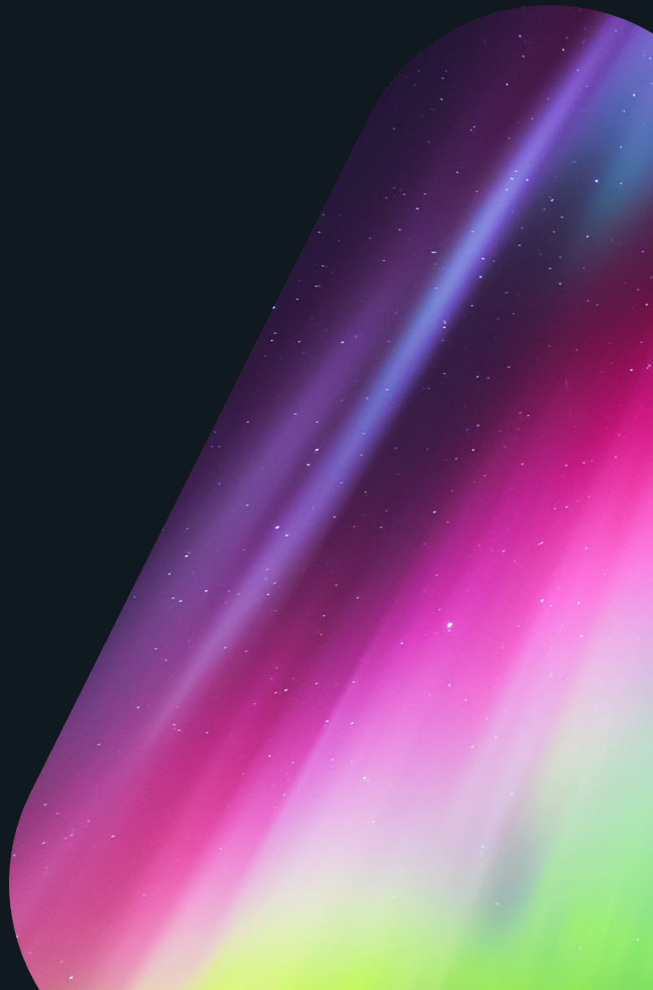
01 Introduction

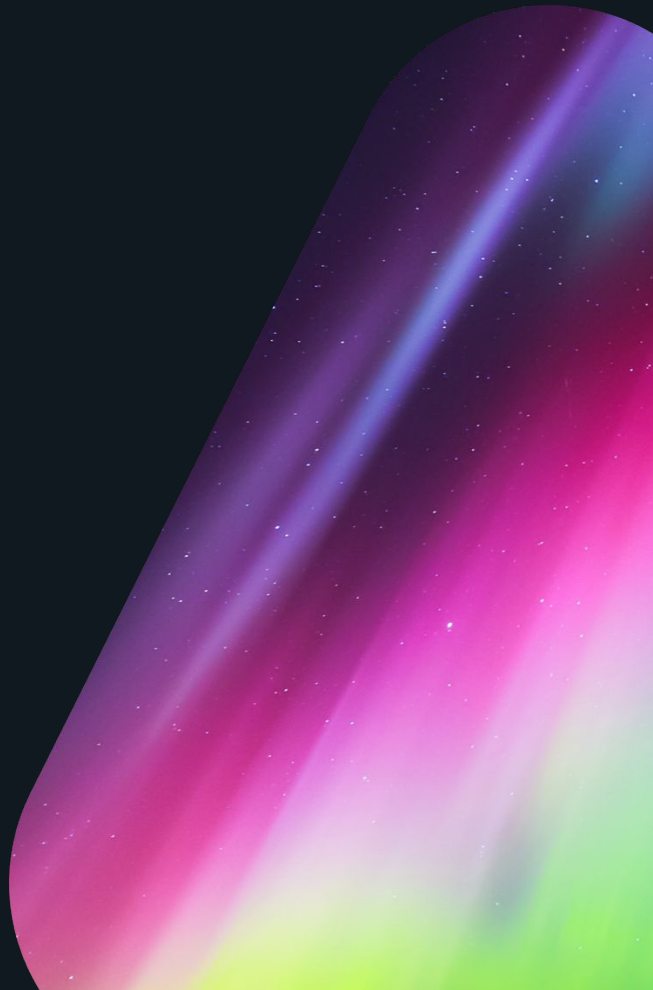
02 Rainbow Bridge architecture

03 Governance

04 Practical aspects

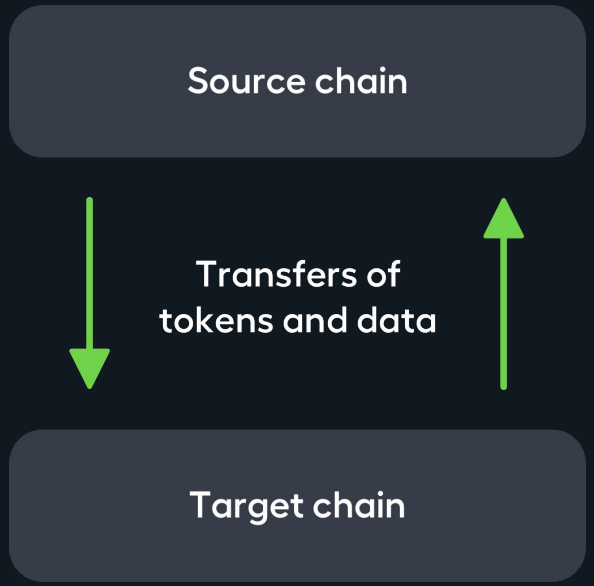
05 Open problems





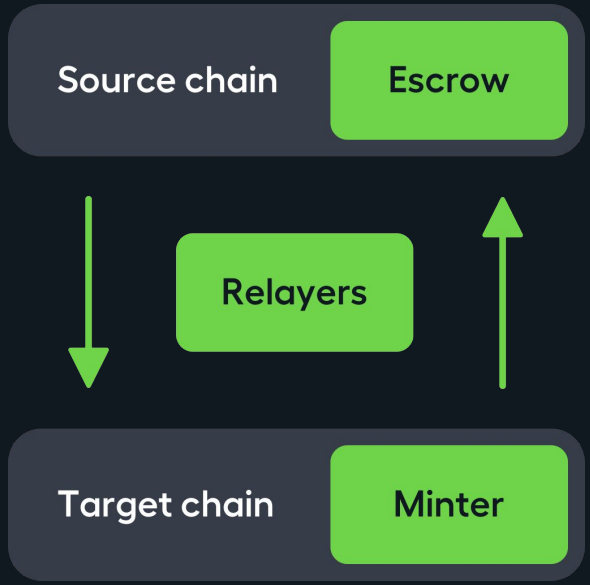
What is a bridge?

- **Sandboxed runtime** — the requirement of data consistency
- **Bridge:** a solution to connect blockchains
  - Token bridges
  - Generic bridges: allow transfers of arbitrary data
- **Existing bridges:** xDAI, Rootstock, Polygon, BSC, Solana, Polkadot, Ren, etc.



Ordinary architecture

- Escrow account on source chain
- An entity responsible for the bridging
  - Single entity
  - Set of participants — bridge validators
- Minting contract on target chain

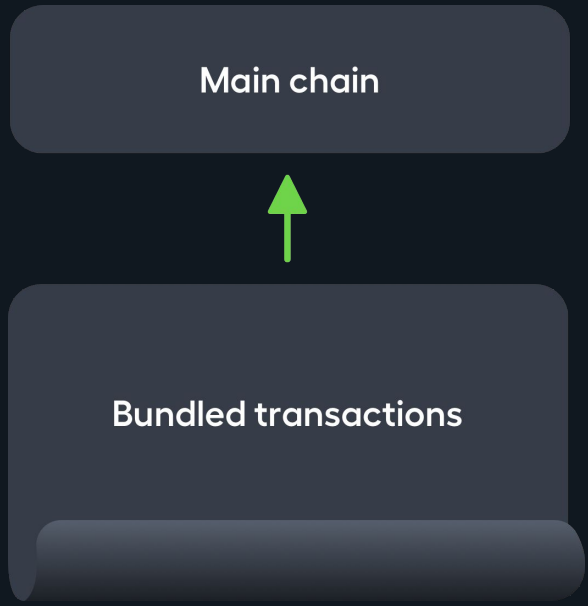


Problem: centralised architecture

Reason: this design is simple, better options are damn hard

Roll-ups

- Scalability solutions that use source chain as a security layer
- Also need bridging capabilities
- The security depends on optimistic verification
  - Results in week delays



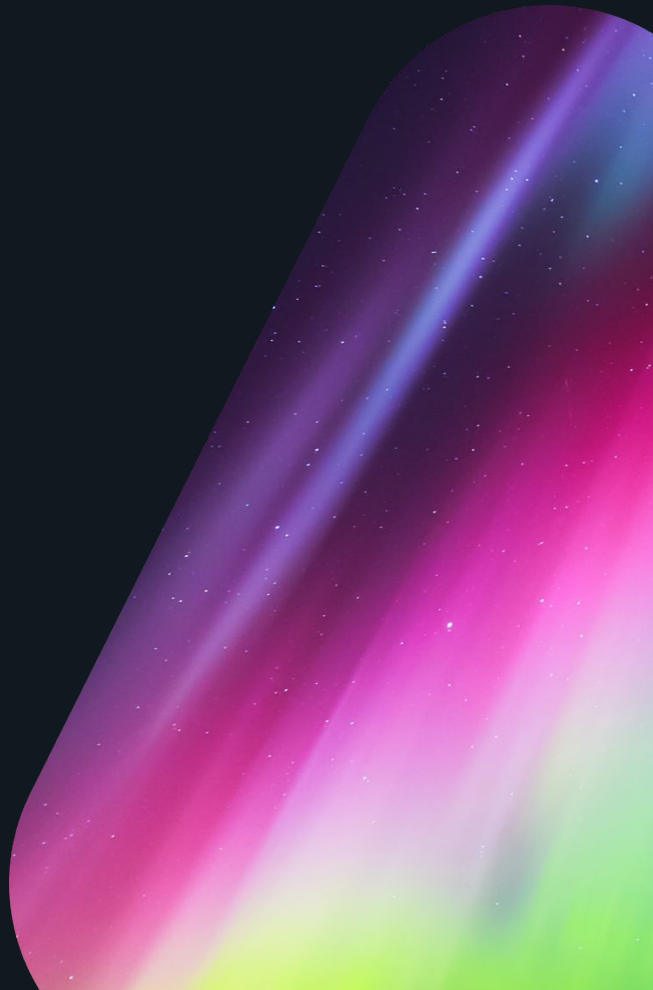
01 Introduction

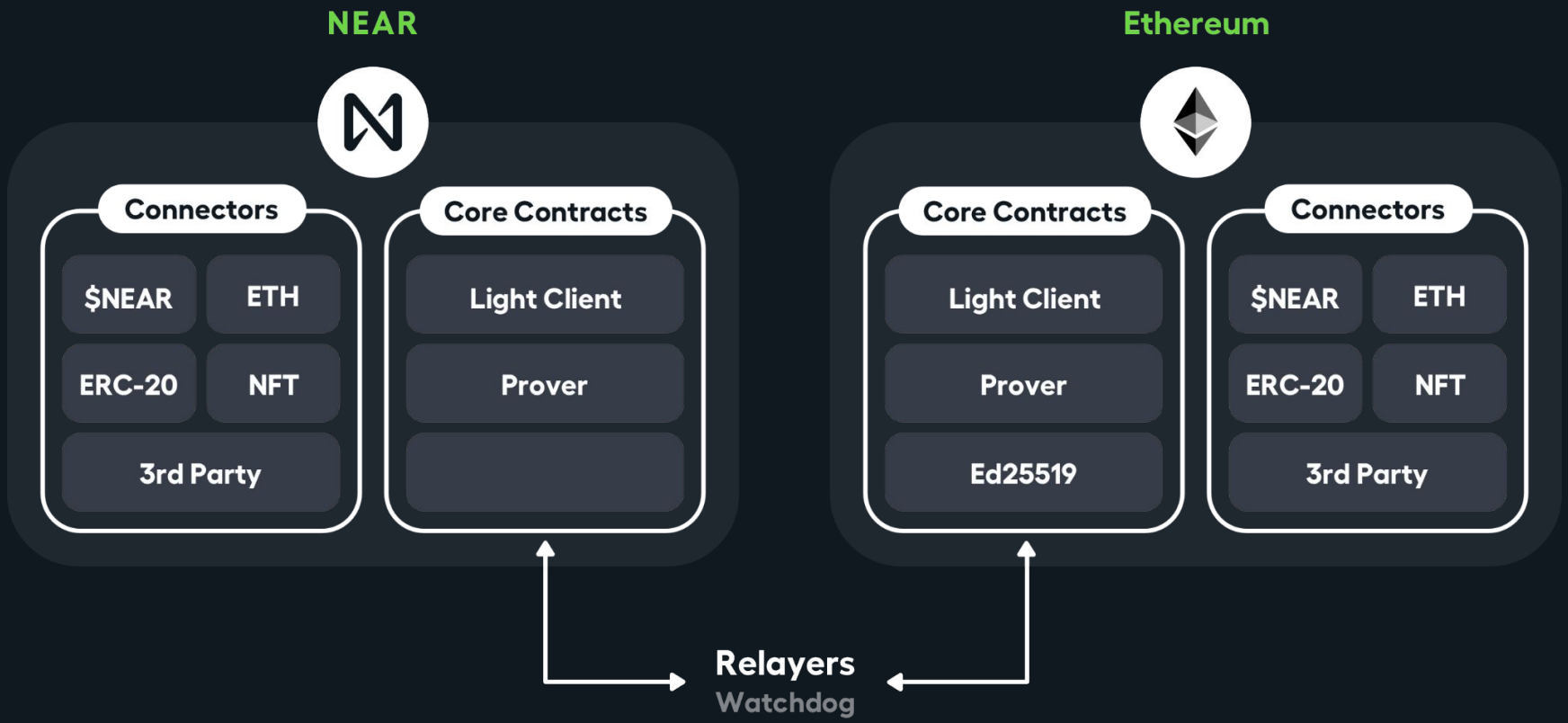
02 Rainbow Bridge architecture

03 Governance

04 Practical aspects

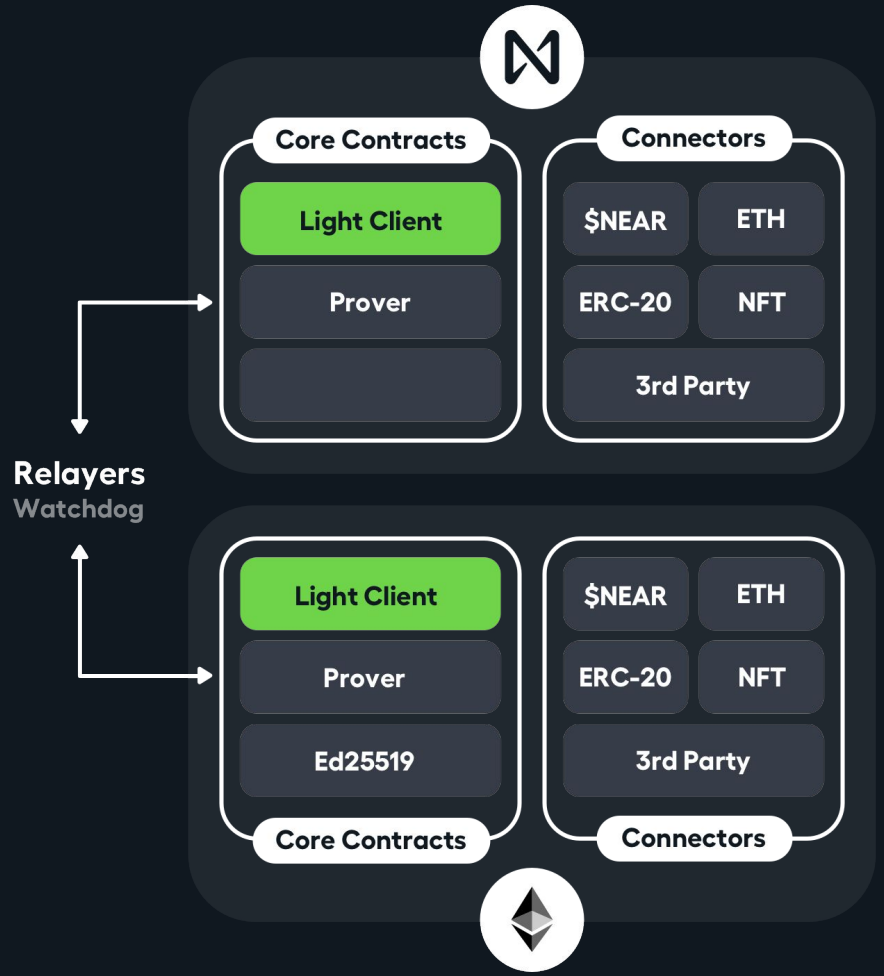
05 Open problems




NEAR
Ethereum
Connectors
$NEAR
ETH
ERC-20
NFT
3rd Party
Core Contracts
Light Client
Prover
Core Contracts
Light Client
Prover
Ed25519
Connectors
$NEAR
ETH
ERC-20
NFT
3rd Party
Relayers
Watchdog

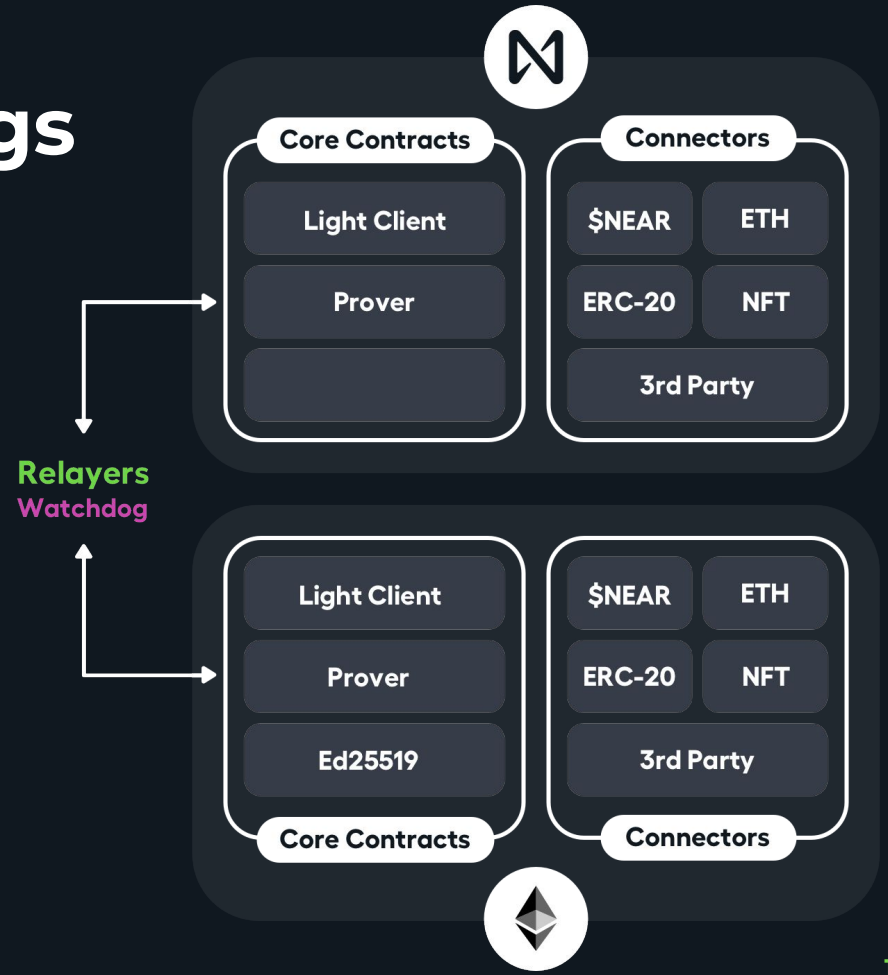
Light clients

- Light clients of Ethereum and NEAR implemented as smart contracts
- Ethereum light client is realistic
  - Garbage collection of old blocks
- NEAR light client is optimistic
  - The absence of EIP-665 (Ed25519 signature check)
  - A single block is stored since the state is persistent
  - Submitted blocks get in the ‘untrusted’ state



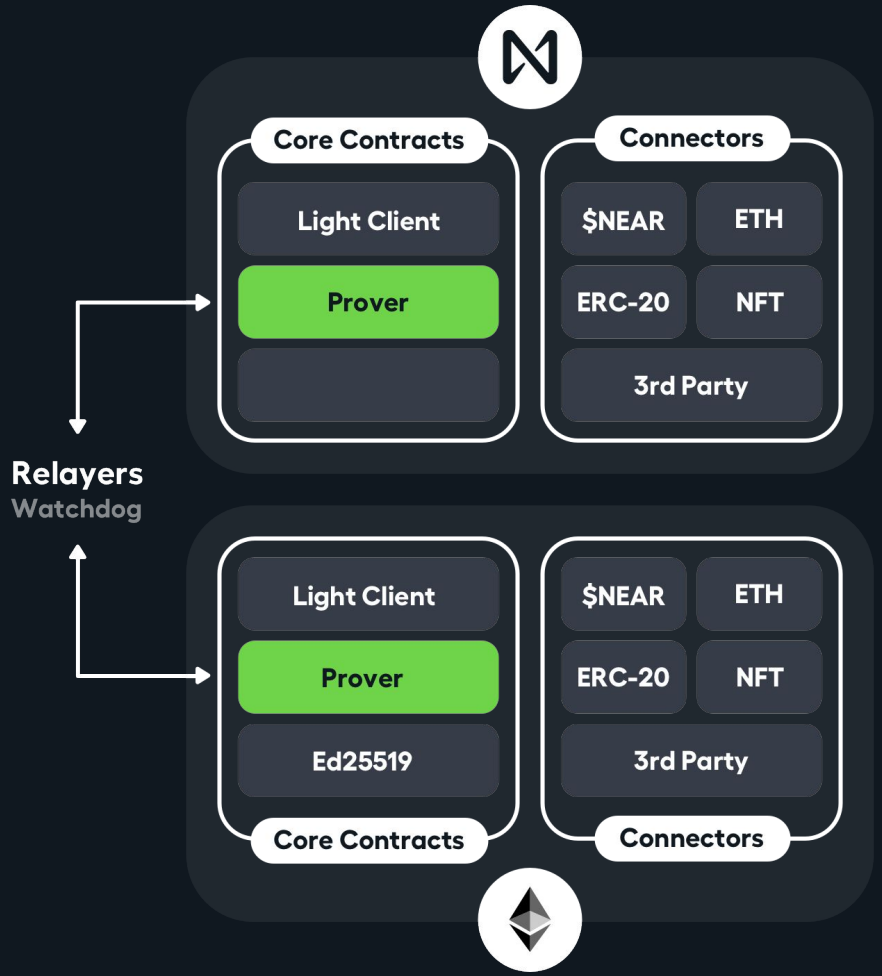
Relayers & watchdogs

- Relayers submit new blocks to light clients
- NEAR light client has specifics:
  - Relayers stake ETH to be able to relay NEAR blocks
  - Watchdog check NEAR blocks submitted and issue a challenge (signature check) if they're incorrect
  - Relayers get slashed in case they submit incorrect blocks



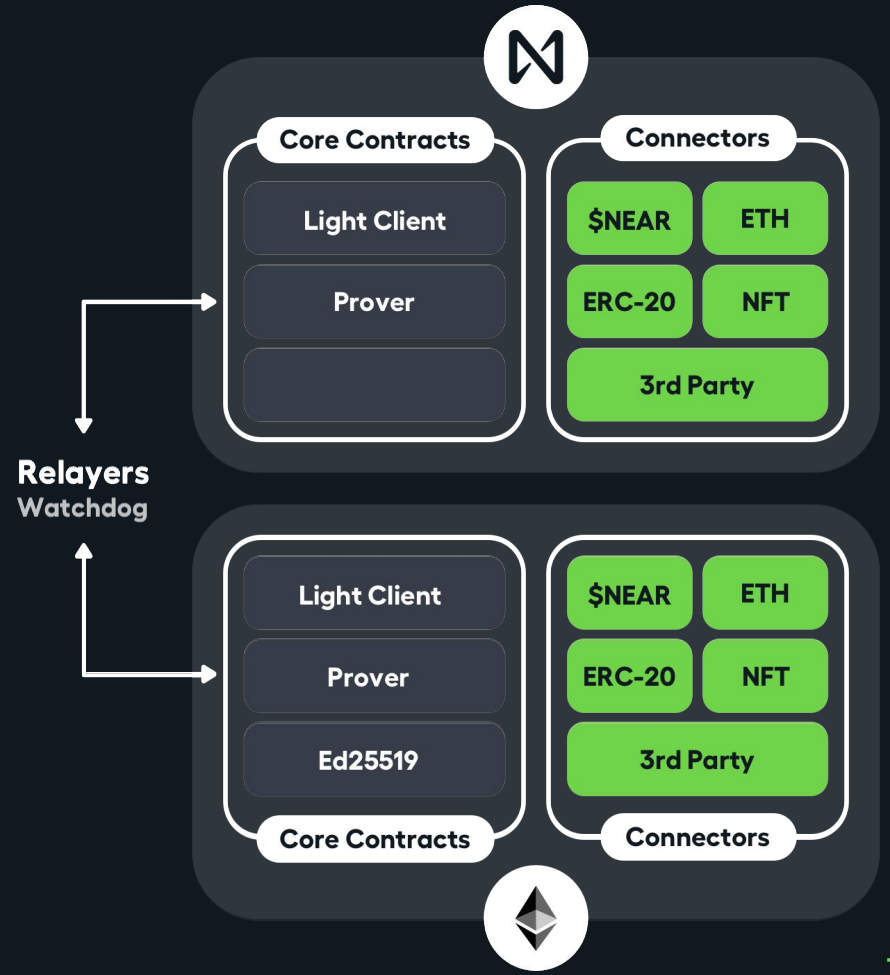
Provers

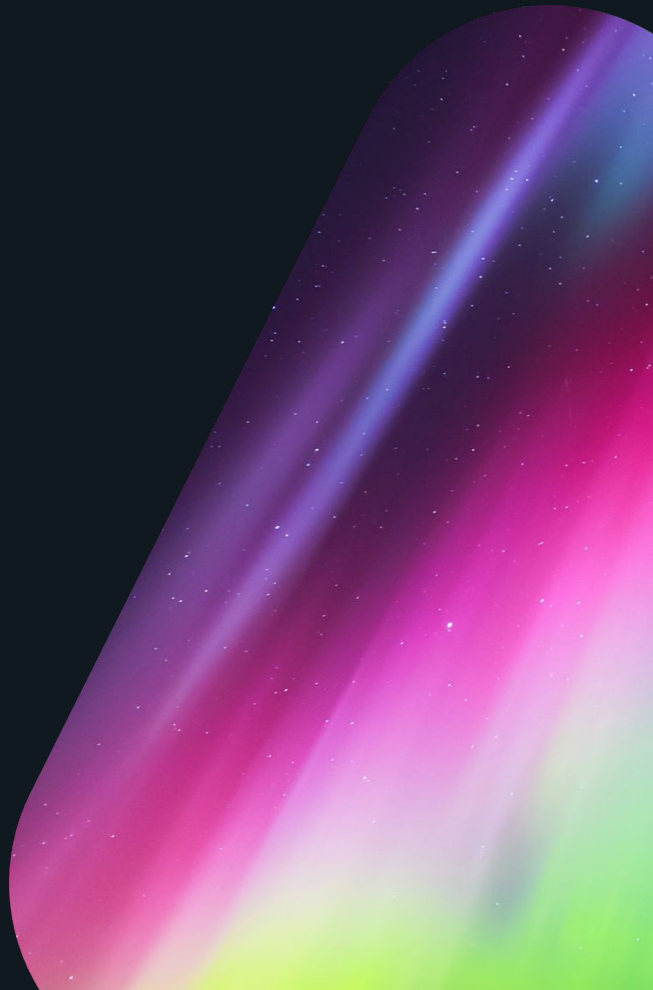
- Provers are the service contracts that provide interface to light clients
- Ethereum prover:
  - Checks Trie inclusion
  - Supports receipts
  - Soon: supports state and transactions
- NEAR prover:
  - Checks Trie inclusion
  - Supports transaction outcomes (similar to receipts)
- Anyone can build his own prover (e.g. the prover for the blockchain height)



Connectors

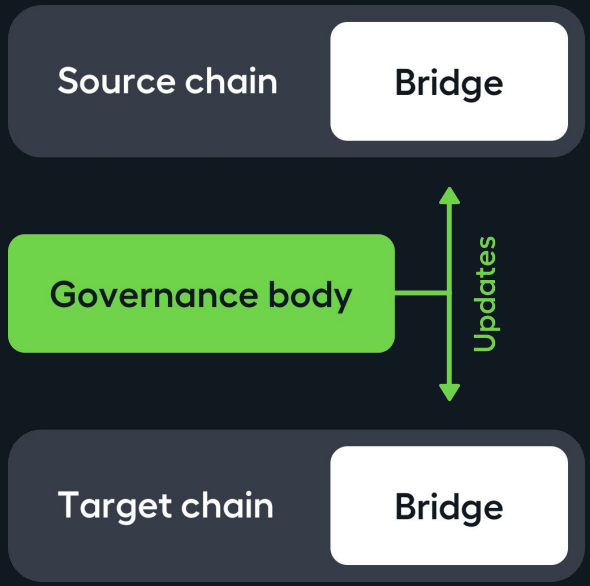
- Connectors implement the messaging protocol:
  - ERC-20
  - ETH
  - $NEAR
  - Soon: ERC-721 (NFT)
- Anyone can build his own connector (e.g. that passes DAO voting results from NEAR to Ethereum)





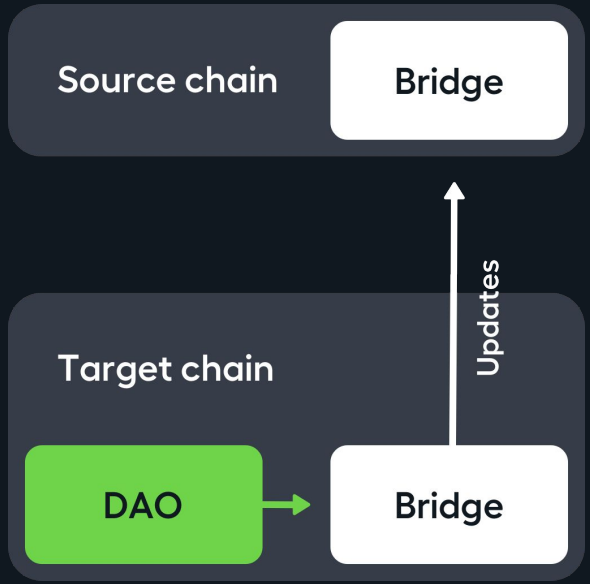
Governance?

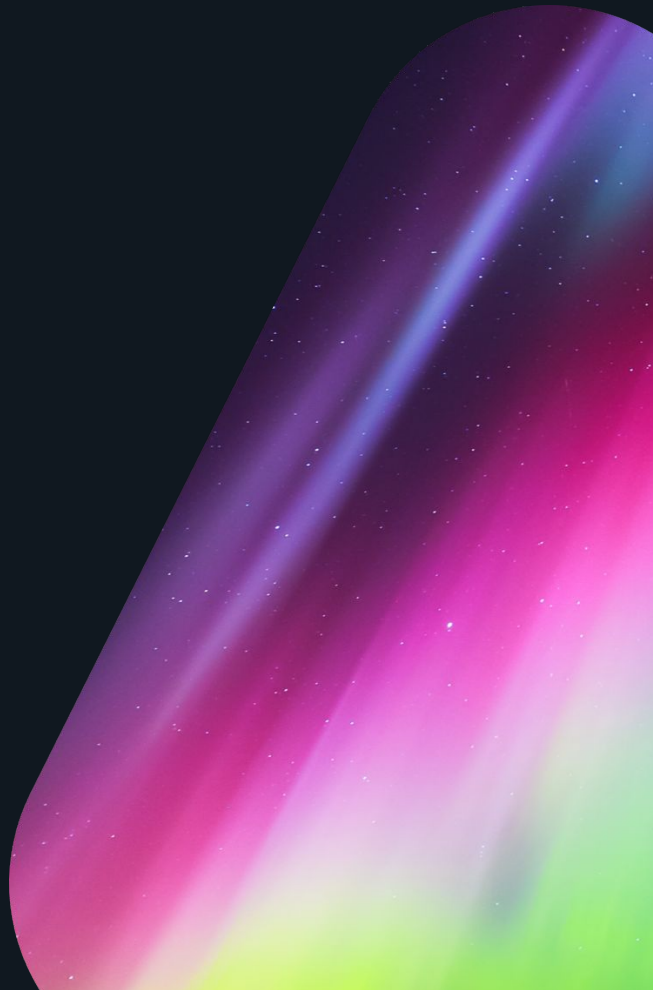
- Hardforks may change headers; state updates may change Trie verification
- Contracts must be upgradable
- Header verification is not a part of an onchain hardfork approval mechanism
- A body required to issue updates



Governance!

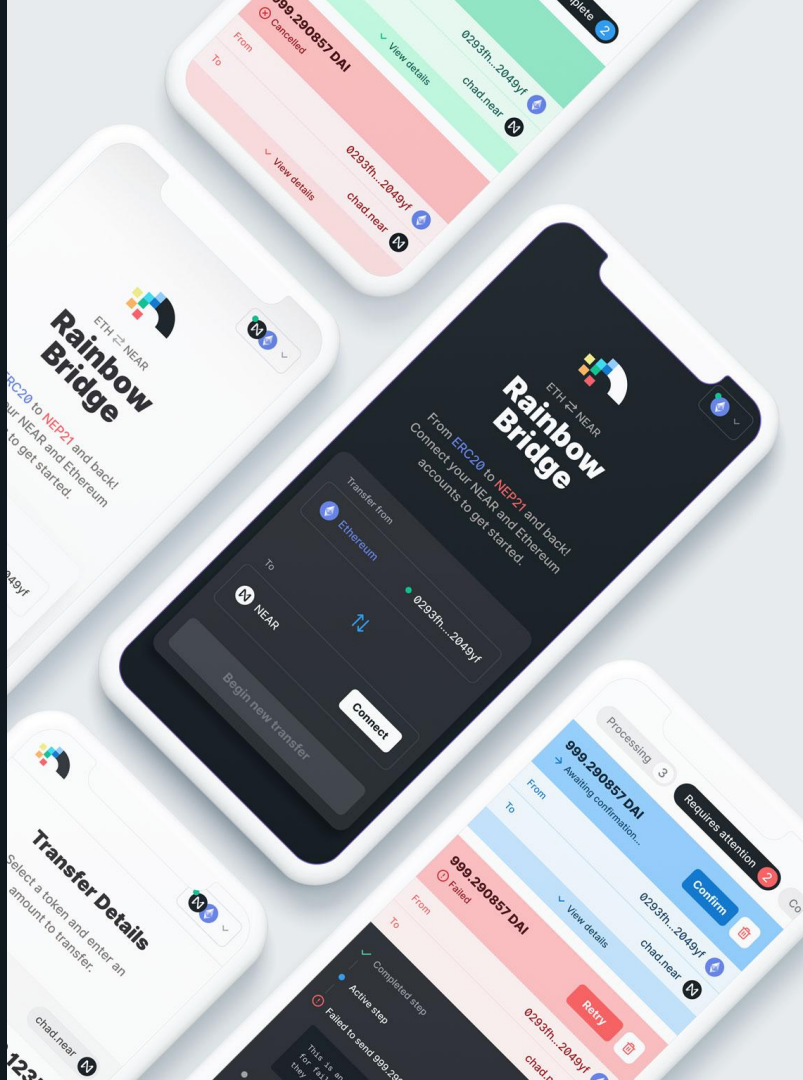
- Governance roadmap
- Ultimate setup:
  - DAO on target chain
  - Voting results applied on target chain directly
  - Voting results passed over the bridge to the source chain
  - DAO voting may be moved to the protocol level of the Target chain (validators DAO)





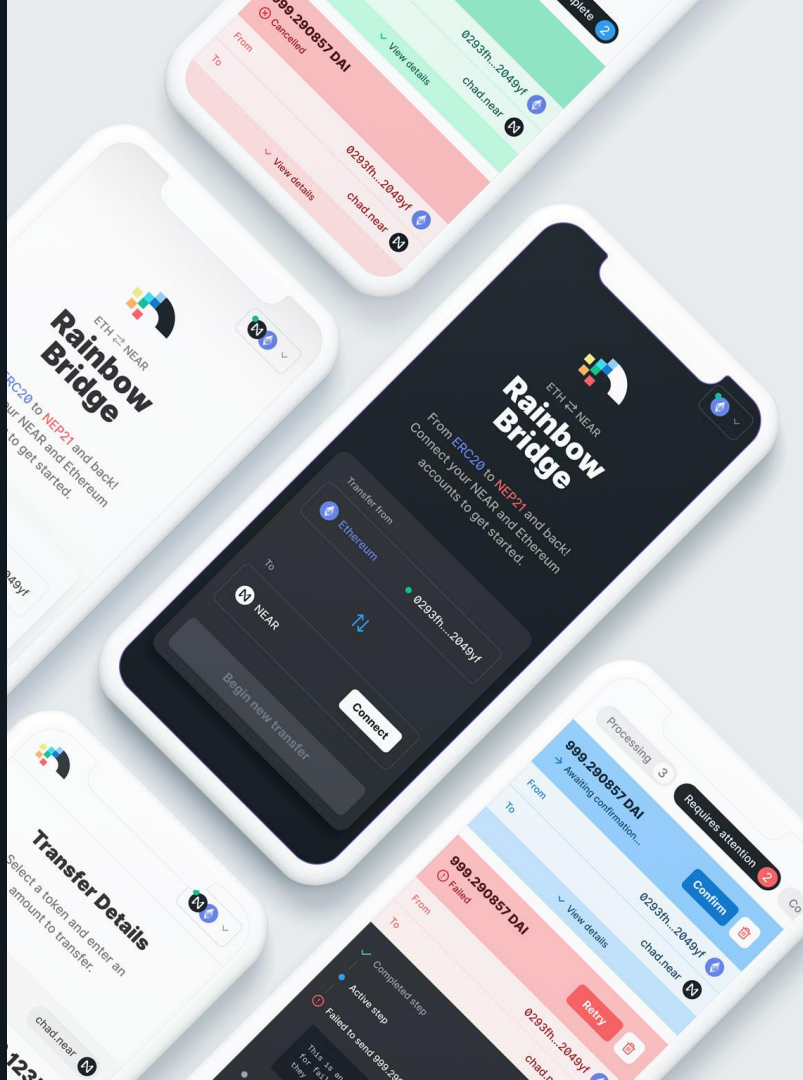
- Bridge Launch: 15th of May 2021
- Zero bridge fees
- Transacted volume: $7M
- Audited in 2020, undergoing next audit now
- Costs:
  - Ethereum -> NEAR: 40k Eth Gas
  - NEAR -> Ethereum: 330k Eth Gas
- Transfer time:
  - Ethereum -> NEAR: 5 min
  - NEAR -> Ethereum: 4h-8h
- ERC-721 connector under review
- Ethereum state and transaction provers are in the development

duneanalytics.com/zavodil/rainbow-bridge



Sugar

- Recipient field implements a protocol:
  - `lockToken(<ERC-20 address>, <amount>, <recipient>)`
  - `<recipient>` may be `'<contract>:<message>'`
- Fast finalization option for tokens
  - Requires LPs
  - Will incur costs for the user
- NEAR blockchain becomes as secure as Ethereum against retrospective attacks



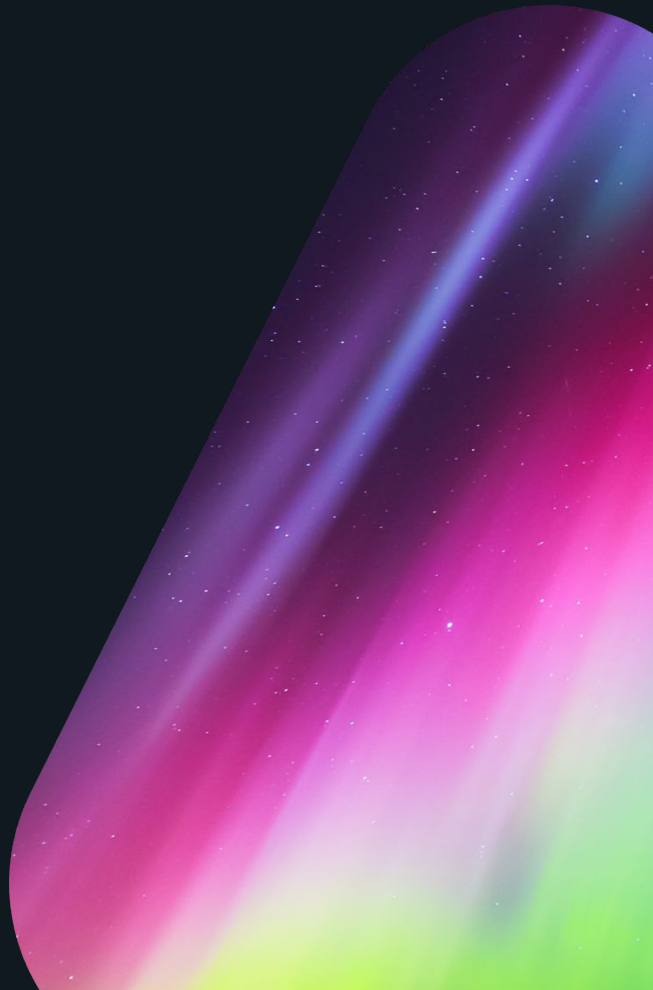
01 Introduction

02 Rainbow Bridge architecture

03 Governance

04 Practical aspects

05 Open problems



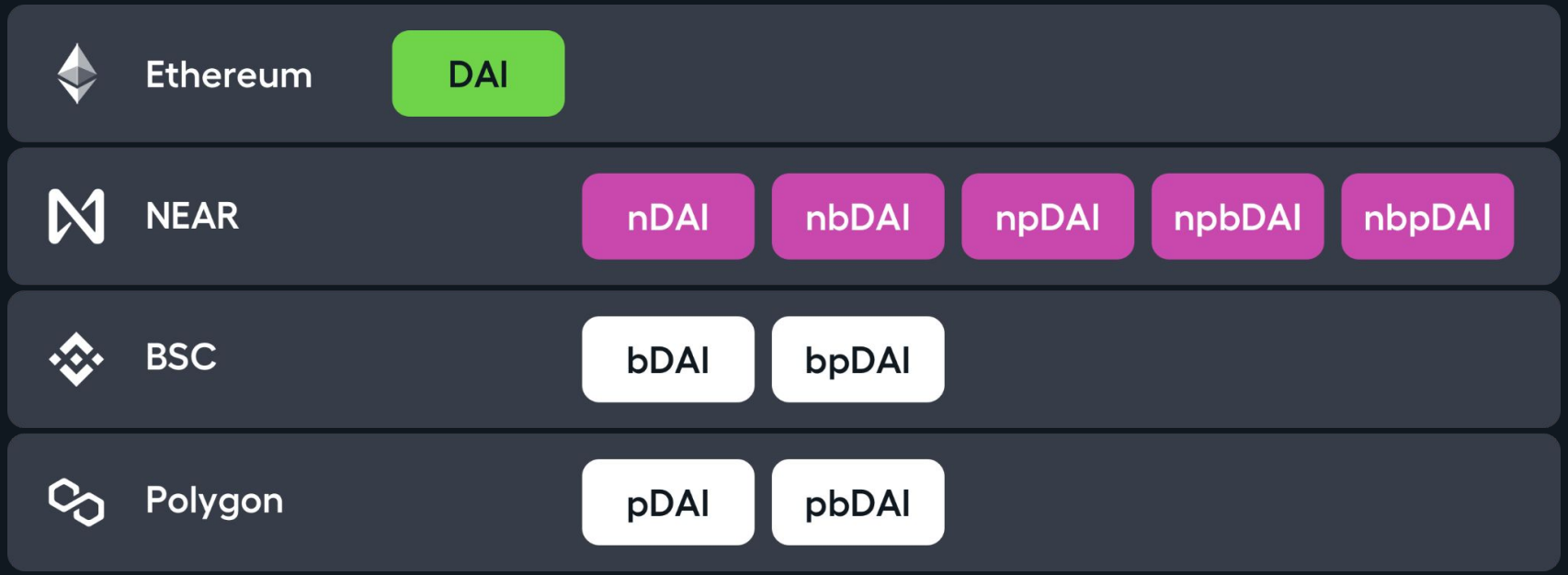
Token consistency

Ethereum	DAI	
NEAR		nDAI
BSC		bDAI
Polygon		pDAI

Token consistency

Ethereum	DAI	
NEAR	nDAI	nbDAI, npDAI
BSC	bDAI	
Polygon	pDAI	

Token consistency



Are the clones of DAI different? Is it convenient to the user?
What to do with metadata and its updates?

Other problems

- How to remove the need for the governance entity?
  Moving header description and verification algorithm to blockchain?
- Security incidents:
  - How to implement escape hatches?
  - Do we need to restrict bridged assets movement within freeze period?
- How to make fast transfers for arbitrary data?
  EIP-665? Secp256k1 keys as a standard on every chain?

Thanks

Alex Shevchenko
CEO Aurora Labs

alex@aurora.dev
@AlexAuroraDev

https://aurora.dev/