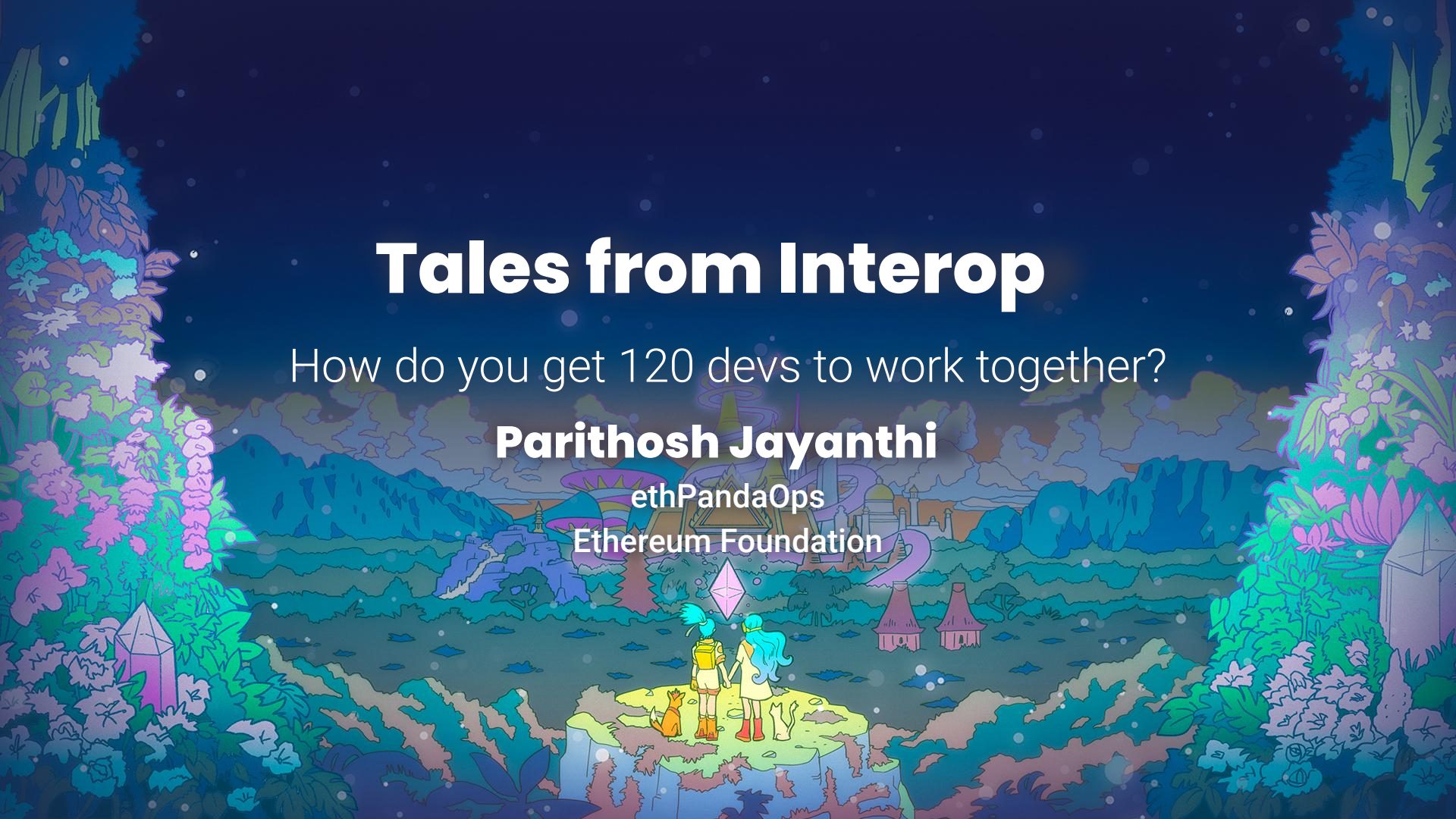

Tales from Interop
How do you get 120 devs to work together?
Parithosh Jayanthi
ethPandaOps
Ethereum Foundation

Section 1

What is Pectra?

The Prague-Electra fork

- The fork in Ethereum after Dencun (shipped March 2024)
- Scoping discussions started in Jan 2024
- By May, we wanted: EIP-3074, MaxEB, EOF, PeerDAS and some misc EIPs included
- SSZ, ePBS, Verkle and Verkle transition were all features under testing
- Different teams are working on different features in parallel
- The feature specifications are all getting updates regularly
- How do you ship a moving target?

Section 2

Nyota interop

Nyota Interop

- Held in Kenya with reps from every client team
- Aim was to work on Pectra + features
- In-person event to reduce communication overhead
- ~120 people working together in total
- Total of 5 workstreams:
    - Pectra (MaxEB, EIP-3074, EIP-6110, etc)
    - EOF
    - PeerDAS
    - Verkle transition
    - Verkle

Challenges?

- ~120 devs working on different forks with various OSes, capabilities, etc
- 5 workstreams are hard to keep up with, no dedicated DevOps person for each one
- Impossible to keep up with updates from other teams
- Hard to test when a feature works as expected
- Discussion rooms occurring all day
- Most devs unaware of how to compile other clients
- Chaos rules

How do we solve the difficulty of keeping up with workstreams?

Client implementation tracker

Client/EIP	EIP-6110	EIP-7002	EIP-7251	EIP-7549	EIP-2537	EIP-2935
Lighthouse	✅	🛠️	🛠️	🛠️	N/A	N/A
Teku	✅	✅	✅	🛠️	N/A	N/A
Lodestar	✅	✅	✅	🛠️	N/A	N/A
Prysm	🛠️	🛠️	🛠️	🛠️	N/A	N/A
Nimbus	✅	✅	✅	🛠️	N/A	N/A
Grandine	🛠️	🛠️	🛠️	🛠️	N/A	N/A

Add a process

- We had a interop shared landing page
- Each workstream got its own document with details about the workstream
- Each workstream doc contains a specification, details, branches, configs
- Each workstream doc contains a config file that some tools can consume to test interop

We automate building images

- We maintain a github actions CI that can build every Ethereum client
- We added a bot to the interop chat that could trigger a build in the CI
- The dev could then just send a message to the bot with `/build` to trigger a build
- The builds would follow a defined format and always push to defined docker repositories
- Docker ensured client dependencies and build systems were abstracted away

How do we abstract other clients from devs?

/build https://github.com/status-im/nimbus-eth2/tree/kzgpeerdas

Your build was triggered. View run on GitHub
Docker Image(s) once run completed:

```
192.168.45.152:80/dh/ethpandaops/nimbus-eth2:kzgpeerdas
192.168.45.152:80/dh/ethpandaops/nimbus-validator-client:kzgpeerdas
```

How do we abstract other clients from devs?

kurtosis run

github.com/ethpandaops/ethereum-package

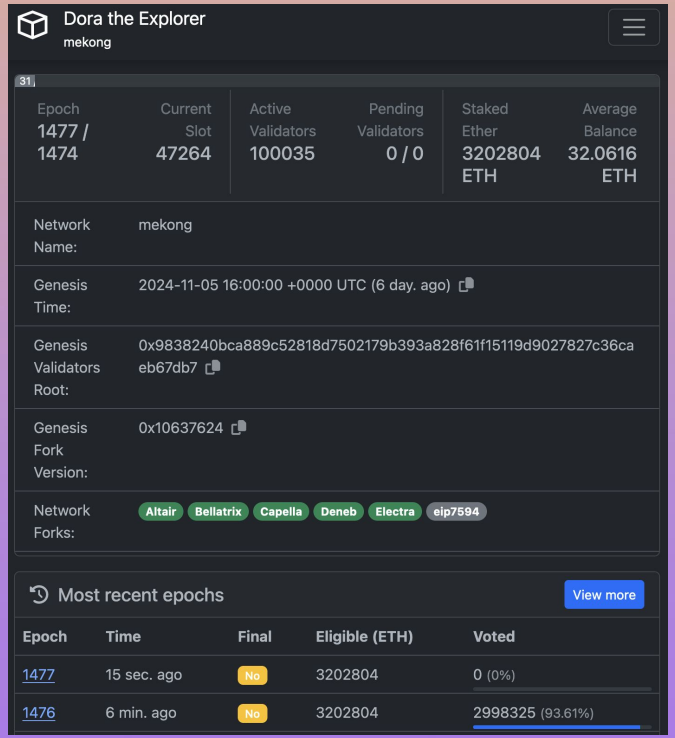
We write tools to run nodes

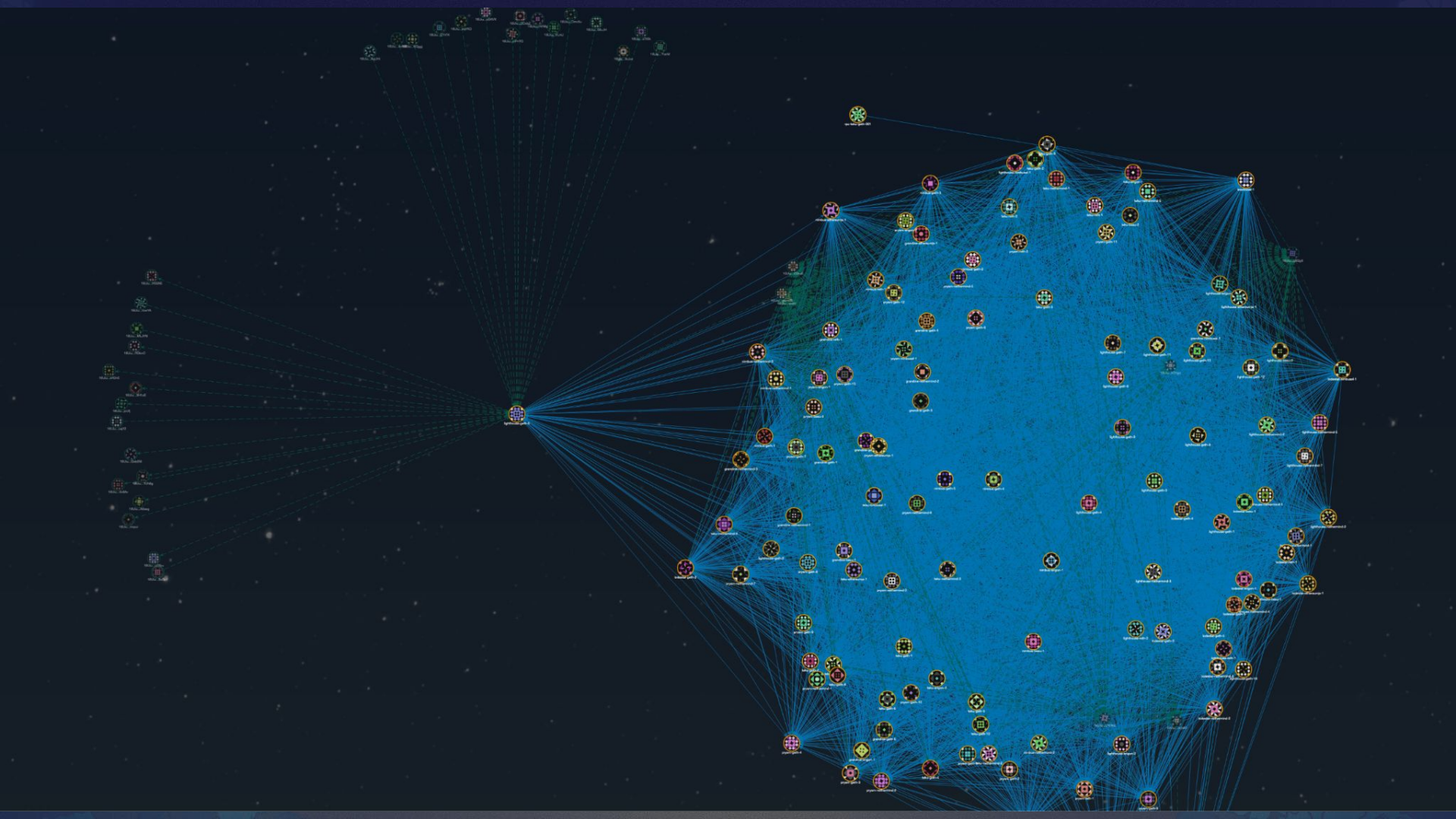
- Kurtosis allows us to specify the configuration of a network as a simple YAML file
- A one line command can then bring up the network in the defined manner
- Devs never need to learn the runtime flags of other clients
- Kurtosis accepts docker images, meaning there are no other dependencies to install
- Kurtosis runs completely locally, so each dev can iterate independently from others

Dora the explorer

- Hard to get a global view from logs
- Custom built explorer with support for forks
- Built using tools/libraries supported by client adjacent teams or testing teams
- Only added features that are useful for client debugging, not for end users
- Accesible database that manual SQL queries can be run on to answer questions
- Extensible when we want to integrate other tools the devops team builds

How do you inspect the chain?





Submit Consolidation

Submit consolidation requests

This tool can be used to create consolidation requests for your validators.

Step 1: Connect your wallet

16.56 ETH 0xf9...F134

Step 2: Select source validator

Select the validator you want to consolidate your funds from. This validator will be exited and its funds will be sent to the target validator.

Select a validator

Step 3: Select target validator

Select the validator you want to consolidate your funds to. This validator will receive the funds from the source validator and gets 0x02 withdrawal credentials assigned.

Select a validator

How do you validate that you can interop with other clients?

https://github.com/ethpandaops/assertoor

Add a test assertion tool

- Introducing Assertoor!
- A test assertion tool that performs an action and asserts the expected outcome
- We wrote basic tests for client teams to use and validate their implementation
- Tests fetched from github => new tests auto-pulled
- E.g: Make a 2000 ETH Deposit and assert that the validator was activated. This test checks EIP-6110, Pectra validator activation logic and MaxEB

Assertoor

Clients

Connected Clients:
CL: 51 / 57 Slot: 175933
EL: 51 / 57 Block: 150483

Test Registrations

All Test Runs

Tests:

Validator Slashing Test

EIP7002 test (el triggerable exits)

Fillup deposit queue

All Test Runs

Run ID	Test Name	Start Time	Run Time	Status	Actions
10	Fillup deposit queue	06-11-2024 14:27:44	38m23.185s / 1h0m0s		
9	Fillup deposit queue	06-11-2024 12:56:07	37m52.459s / 1h0m0s		
8	Fillup deposit queue		? / 1h0m0s		
7	Fillup deposit queue		?		
6	Fillup deposit queue		?		
5	EIP7002 test (el triggerable exits)	25-10-2024 12:41:55	5h59m35.923s / 38h0m0s		
4	EIP7002 test (el triggerable exits)		?		
3	pectra-dev-all	21-10-2024 07:20:33	1m45.754s		
2	pectra-dev-all	21-10-2024 06:42:08	30m2.936s		
1	pectra-dev-all	21-10-2024 06:40:44	1m24.342s		

Selected Test Runs

Showing test runs 0 to 9 of 10

First 1 of 1 Last

25

Connected Clients:
CL: 46 / 57 Slot: 176748
EL: 46 / 57 Block: 151255

Test Registrations

All Test Runs

Tests:

Validator Slashing Test

EIP7002 test (el triggerable exits)

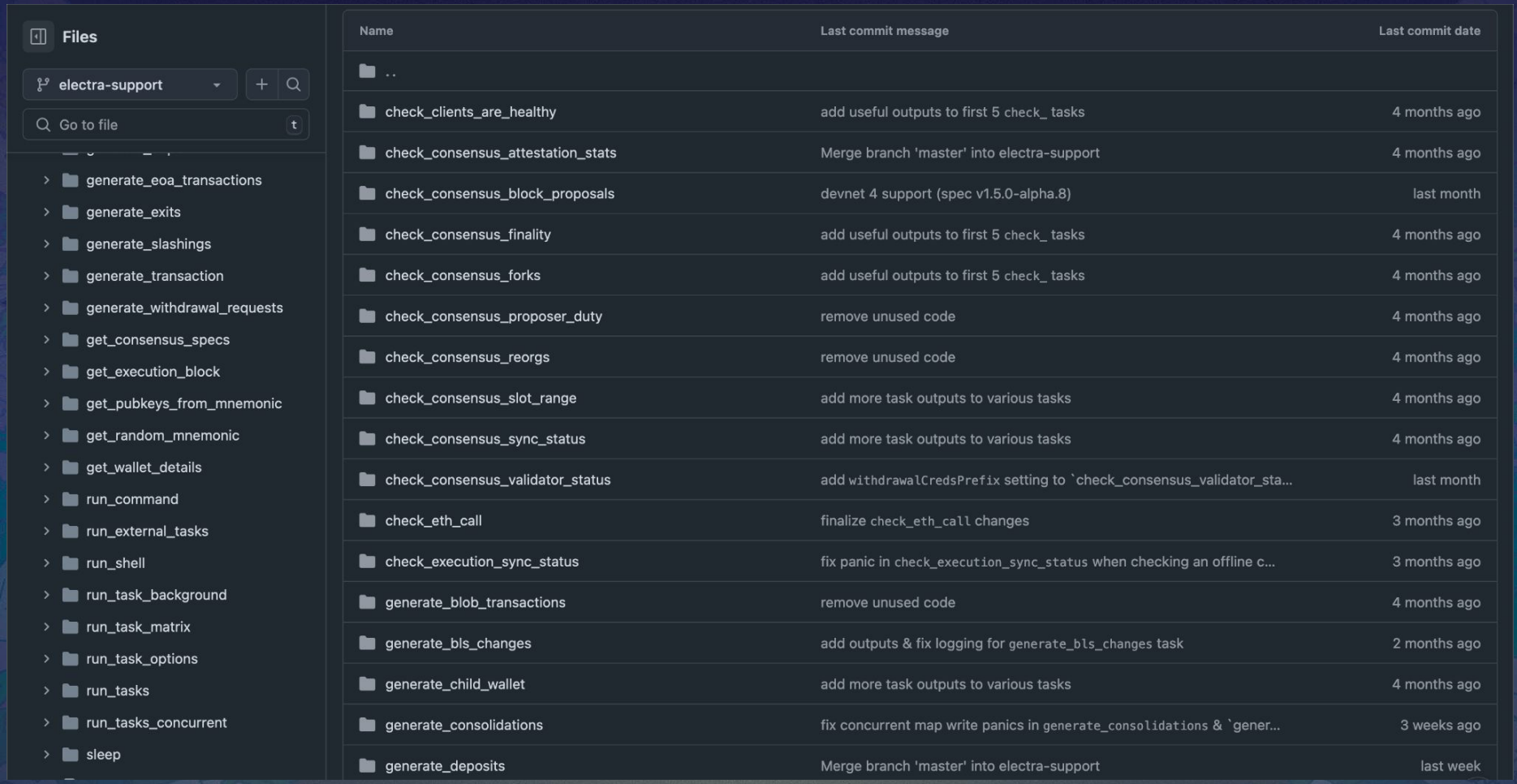
Fillup deposit queue

Test Run 4: EIP7002 test (el triggerable exits)

Test ID: eip7002-all
Test Status: Cancelled
Timeout: 0s

Tasks

ID	Action Name	Task Title	Run Time	Status
1	check_clients_are_healthy	Check if at least one client is ready	14ms / 5m0s	✓
2	get_consensus_specs	Get consensus chain specs	8ms	✓
3	check_consensus_slot_range	Wait for electra activation	9ms / 1h0m0s	✓
4	get_wallet_details		9ms	✓
5	run_tasks	Generate test validators & wait for activation	1h7m21.386s	✓
10	get_random_mnemonic	Generate random mnemonic	13ms	✓
11	run_shell	Use generated random mnemonic for tests	28ms	✓
12	generate_child_wallet	Generate wallet for new deposits	8.475s	✓
13	generate_deposits	Generate 10 deposits with 32 ETH each and 0x00 withdrawal credentials	13.718s	✓
14	run_tasks	Wait for validators to be active	1h6m59.077s	✓
15	check_consensus_validator_status	Wait for validators to be active	1h6m59.042s / 2h0m0s	✓
6	get_pubkeys_from_mnemonic	Get test validator pubkeys	63ms	✓
7	run_tasks	Wait for validators to be exitable (2 epochs)	20h33m29.65s	✗


Files
electra-support
Go to file
t
generate_eoa_transactions
generate_exits
generate_slashings
generate_transaction
generate_withdrawal_requests
get_consensus_specs
get_execution_block
get_pubkeys_from_mnemonic
get_random_mnemonic
get_wallet_details
run_command
run_external_tasks
run_shell
run_task_background
run_task_matrix
run_task_options
run_tasks
run_tasks_concurrent
sleep
Name
Last commit message
Last commit date
..
check_clients_are_healthy
add useful outputs to first 5 check_ tasks
4 months ago
check_consensus_attestation_stats
Merge branch 'master' into electra-support
4 months ago
check_consensus_block_proposals
devnet 4 support (spec v1.5.0-alpha.8)
last month
check_consensus_finality
add useful outputs to first 5 check_ tasks
4 months ago
check_consensus_forks
add useful outputs to first 5 check_ tasks
4 months ago
check_consensus_proposer_duty
remove unused code
4 months ago
check_consensus_reorgs
remove unused code
4 months ago
check_consensus_slot_range
add more task outputs to various tasks
4 months ago
check_consensus_sync_status
add more task outputs to various tasks
4 months ago
check_consensus_validator_status
add withdrawalCredsPrefix setting to `check_consensus_validator_sta...
last month
check_eth_call
finalize check_eth_call changes
3 months ago
check_execution_sync_status
fix panic in check_execution_sync_status when checking an offline c...
3 months ago
generate_blob_transactions
remove unused code
4 months ago
generate_bls_changes
add outputs & fix logging for generate_bls_changes task
2 months ago
generate_child_wallet
add more task outputs to various tasks
4 months ago
generate_consolidations
fix concurrent map write panics in generate_consolidations & `gener...
3 weeks ago
generate_deposits
Merge branch 'master' into electra-support
last week

Ugh, more test frameworks...

Test Run 1: Run execution spec tests

Test ID: local-1
Test Status: Success
Start Time: 13-09-2024 15:47:43
Finish Time: 13-09-2024 15:55:11
Timeout: 0s

Tasks

ID	Action Name	Task Title	Run Time	Status
1	run_shell	Install dependencies	28.2s	
2	get_consensus_specs		0s	
3	run_external_tasks	Run execution spec tests: eip7702_set_code_tx	6m59.848s	
4	run_shell	Execute tests: https://github.com/ethereum/execution-spec-tests.git@fill-execute-modes [./tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code]	6m59.615s	
5	run_task_matrix	Show test results	228ms	
6	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-stop-tx_value_0]	208ms (48.279076018s)	
7	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-stop-tx_value_1]	216ms (36.18391912s)	
8	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-return-tx_value_0]	190ms (35.270073179s)	
9	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-return-tx_value_1]	212ms (36.210444958s)	
10	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-revert-tx_value_0]	220ms (36.184561675s)	
11	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-revert-tx_value_1]	208ms (36.184712793s)	
12	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-invalid-tx_value_0]	212ms (36.172050411s)	
13	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-invalid-tx_value_1]	212ms (36.238466791s)	
14	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-out-of-gas-tx_value_0]	203ms (35.222100138s)	
15	run_shell	tests/prague/eip7702_set_code_tx/test_set_code_txs.py::test_self_sponsored_set_code[fork_Prague-transaction_post-out-of-gas-tx_value_1]	211ms (48.240314629s)	

How do we combine these things?

```
participants:
  el:
    - el_type: reth
      el_image: ethpandaops/reth:main-dd18af1
  cl:
    - cl_type: teku
      cl_image: consensys/teku:24.10.3
network_params:
  electra_fork_epoch: 1
  min_validator_withdrawability_delay: 1
additional_services:
  - dora
  - assertoor
snooper_enabled: true
assertoor_params:
  tests:
    # ALL ELECTRA
    - { file:
"https://raw.githubusercontent.com/ethpandaops/
assertoor/refs/heads/electra-support/playbooks/
pectra-dev/all.yaml"}
```

```
optimism_package:
  chains:
      - network_params:
      name: op-rollup-one
      network_id: '3151909'
      - network_params:
      name: op-rollup-two
      network_id: '3151910'
```

- Easy to integrate into the testing workflows!
- We’re hoping more L2s will adopt this setup

Sidenote, what else does this workflow enable?

Section 3

Where is Pectra now?

Close...

- Pectra as a fork was split into 3 “smaller” forks in August
- Pectra will include: MaxEB, EIP-7702, Staking workflow related EIPs and more
- EOF & PeerDAS will move to the next fork
- This is to speed up the process of delivering EIPs to Mainnet and to ensure adequate testing
- Major changes are all finalized, minor tweaks might still happen
- Client teams slowly prepare to merge in the code
- Edge case bug hunting underway

How close are we to ship pectra?

https://eips.ethereum.org/EIPS/eip-7600

Section 4

What’s left?

The community!

https://mekong.ethpandaops.io

Use the testnet!

- As a Wallet dev, Add support for EIP-7702
- As a validator, look into EL triggered exits or MaxEB
- As a node operator, run a node!
- As a tooling dev, support MaxEB features
- As a user, try sending transactions or using a EIP-7702 wallet
- Help us break it!

How can i take part?

https://devcon-mekong-faucet.pk910.de/

Here’s the timeline.

Mekong testnet

- Live now!
- Testing underway
- Minor changes will be made based on findings
- Time to start getting involved as a commuity

Holesky/Sepolia testnet

- Will be scheduled once changes are final (early 2025)
- Last chance to try changes out before Mainnet

Mainnet

- Mainnet!

Description

Demonstrate how we collect data from the Ethereum network and how it's used for upgrades, research, and analytics. We'll then run through some examples of how to use the tools and public datasets yourself.

Nov 14th — 10:30 AM - 11:50 AM

Workshop - Classroom D

Attend Session | Mark as interesting | Export to Calendar

Tell me about similar sessions

Speakers

Andrew Davis
@savid

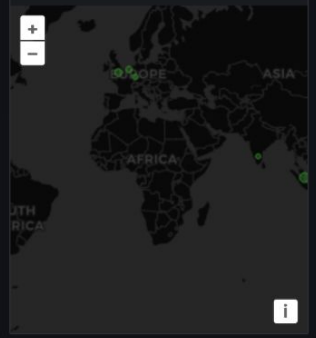
Sam Calder-Mason
@samcmau

Summary

Active sentries

95

Active Sentries



Chain reorgs

0

Blocks ingested

24.840

Attestations ingested

19.915.463

Summary per client

Active sentries (teku)

23

Active sentries (prysm)

26

Active sentries (nimbus)

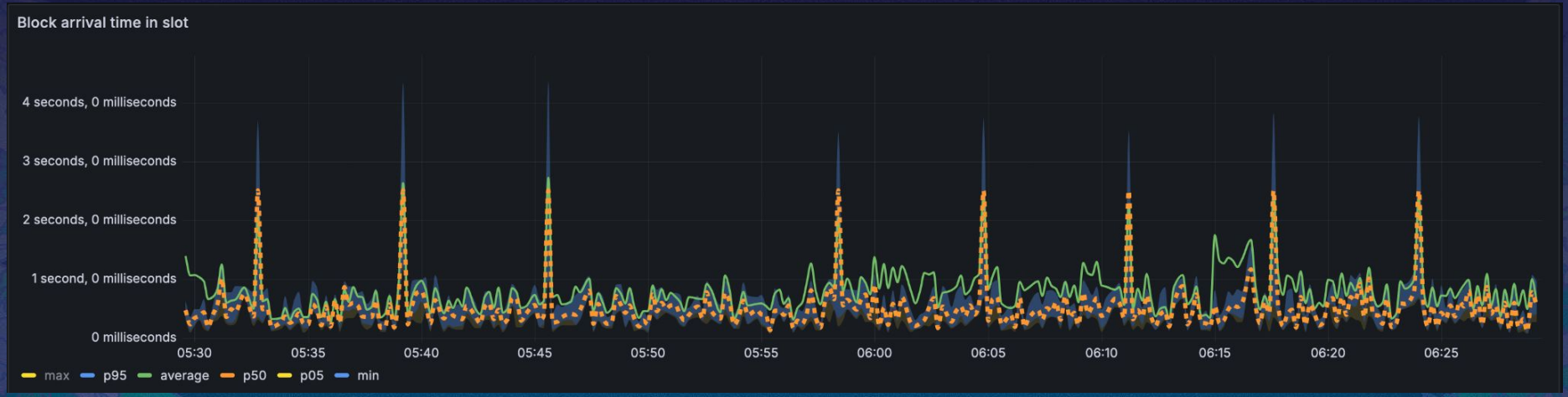
12

Active sentries (lodestar)

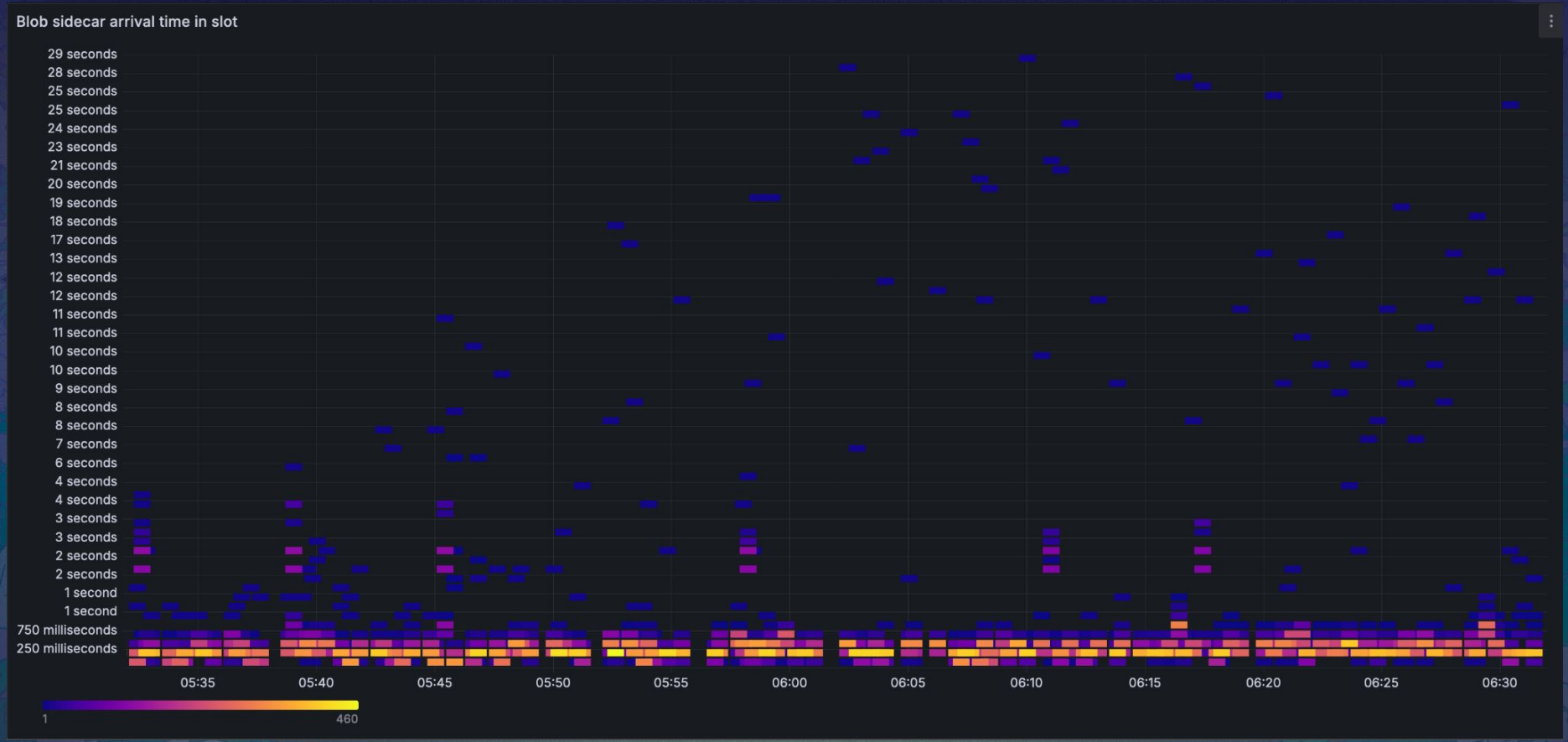
12

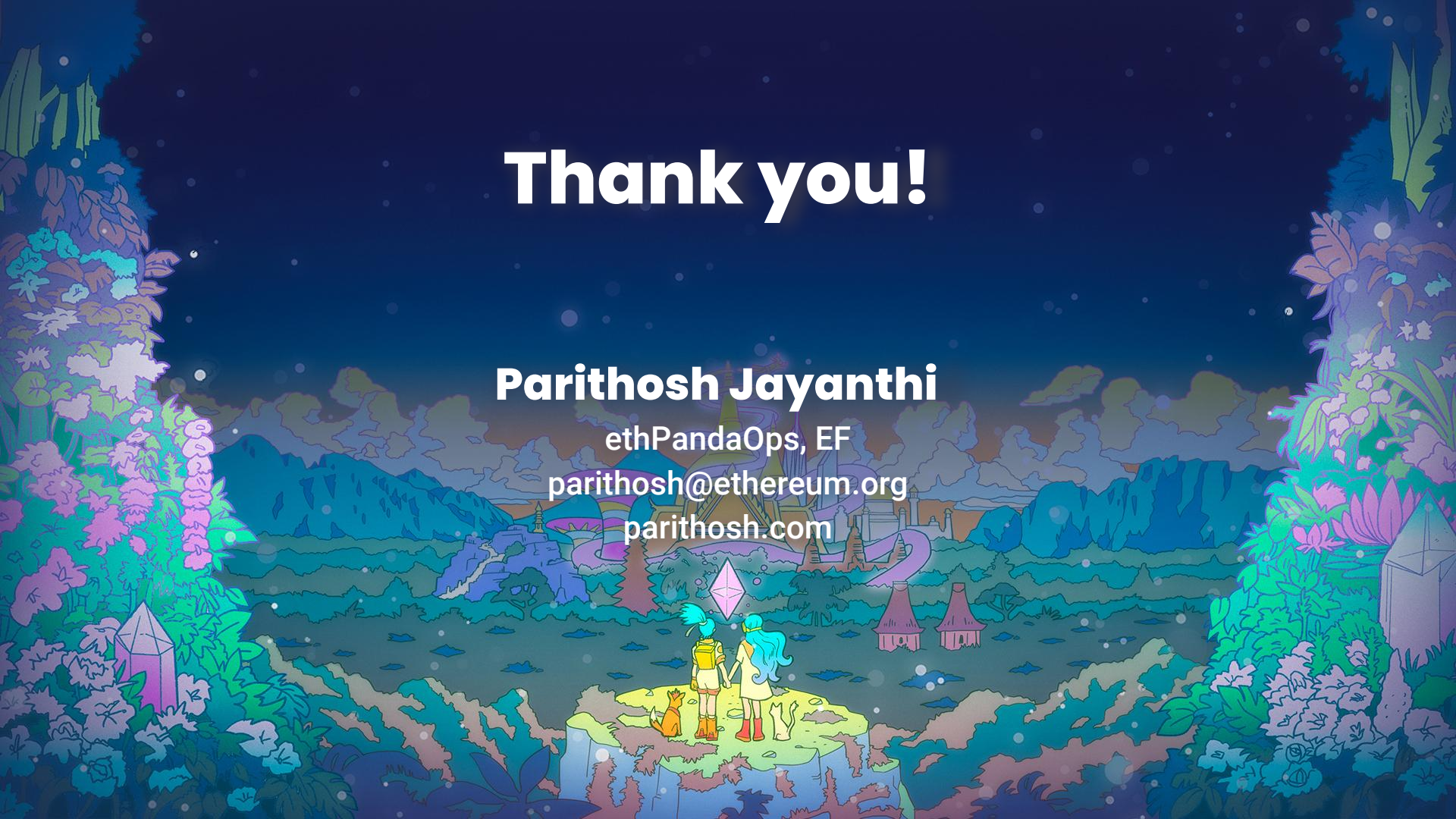
Active sentries (lighthouse)

22


Block arrival time in slot
4 seconds, 0 milliseconds
3 seconds, 0 milliseconds
2 seconds, 0 milliseconds
1 second, 0 milliseconds
0 milliseconds
05:30
05:35
05:40
05:45
05:50
05:55
06:00
06:05
06:10
06:15
06:20
06:25
max
p95
average
p50
p05
min


Block arrival time spread
3 seconds, 0 milliseconds
2 seconds, 0 milliseconds
1 second, 0 milliseconds
0 milliseconds
05:35
05:40
05:45
05:50
05:55
06:00
06:05
06:10
06:15
06:20
06:25
06:30
std dev population
p90 - p10


Blob sidecar arrival time in slot
29 seconds
28 seconds
25 seconds
25 seconds
24 seconds
23 seconds
21 seconds
20 seconds
19 seconds
18 seconds
17 seconds
13 seconds
12 seconds
12 seconds
11 seconds
11 seconds
10 seconds
10 seconds
9 seconds
8 seconds
8 seconds
7 seconds
6 seconds
4 seconds
4 seconds
3 seconds
3 seconds
2 seconds
2 seconds
1 second
1 second
750 milliseconds
250 milliseconds
05:35
05:40
05:45
05:50
05:55
06:00
06:05
06:10
06:15
06:20
06:25
06:30
1
460


Thank you!
Parithosh Jayanthi
ethPandaOps, EF
parithosh@ethereum.org
parithosh.com