Protec and Attac: Programmatic Execution Layer Consensus Tests

ethereum/execution-spec-tests

danceratopz

Protocol Tester, Ethereum Foundation









Client Diversity and Testing

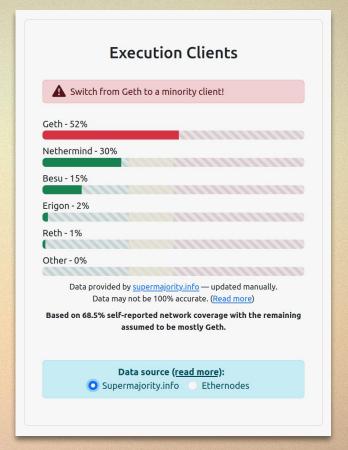
Client diversity is a foundation of Ethereum protocol development.

Client diversity makes the network more resilient against attacks and software bugs.

Ethereum has ≥9 EVM implementations - in ≥8 different languages!

..but it does add complexity to testing.

The ethereum/execution-specs-tests (EEST) team maintains a set of client-independent Execution Layer (EL) consensus tests.



Source: clientdiversity.org, 2024-11-07



Execution Layer Consensus Tests

Every client must handle state changes identically.

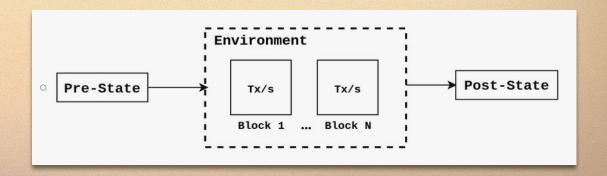
State tests validate

- Opcodes,
- Gas usage,
- Smart contract interactions,
- Transaction rejections.

Blockchain tests validate

- Block progression,
- Block rejections,
- Fork transitions.









Section 2

Status Quo at The Merge





Generating Tests for forks ≥Frontier from ethereum/tests

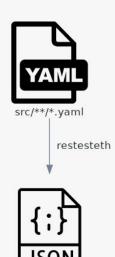
ethereum/tests contains 4697 tests for Cancun.

Test cases are defined in YAML files, hand-written.

ethereum/retesteth generates JSON test fixtures from the YAML files using a reference EVM implementation.

The process of generating the JSON is known as "filling the tests".





fixtures/**/*.json



Generating Tests for forks ≥Frontier from ethereum/tests

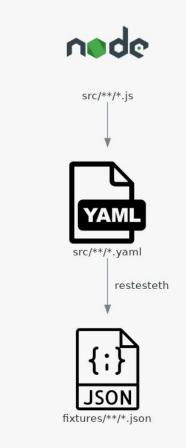
ethereum/tests contains 4697 tests for Cancun.

Test cases are defined in YAML files, hand-written.

2021: Partial programmatic generation!

ethereum/retesteth generates JSON test fixtures from the YAML files using a reference EVM implementation.

The process of generating the JSON is known as "filling the tests".







A Framework for Programmatic Tests

Section 3



The new framework: ethereum/execution-spec-te sts

2021-10: interop greece, how to test AA?

→ programmatic test generation!

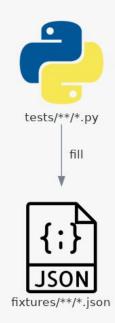
2021-10: cd testing-tools && git init

2021-12: sleep 22222

2022-10: devcon 6: "testing-tools"!?

2022-10: → ethereum/execution-spec-tests

2023-03: EEST Shanghai release: 194 tests



Pytest and Cancun

Shanghai showed how easy it can be to implement test cases in Python.

We then re-wrote the framework to use, pytest:

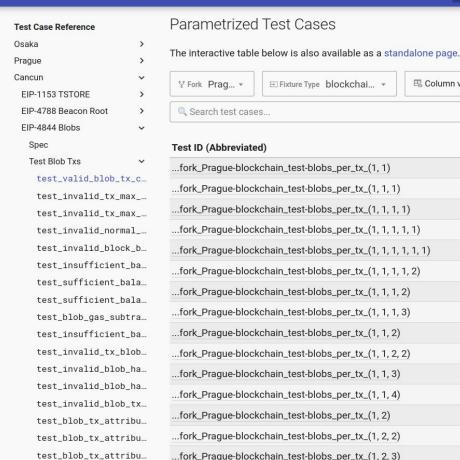
- Extendable via plugins.
 - ...large existing ecosystem.
 - ...allowing a custom modular architecture.
- Easier debugging
- Test cases easily parameterizable

This has greatly accelerated test development.

2024-03: EEST Cancun release 2412



test_valid_blob_tx_combinations



Contributor First Repository

EIP authors and implementers can use EEST to:

- Write tests,
- Generate tests (from a devnet tx hash),
- Check coverage.

Prague:

- ~2500 new test cases

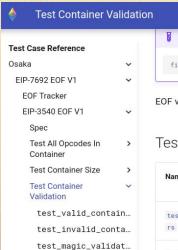
EVM Object Format (EOF):

- Added eof test format.
- ~9000 new test cases for EOF.

Verkle:

- Existing tests (until Shanghai) can be filled for the Verkle Transition.
 - Added the witness to the test fixtures.





test version valid...

test_single_code_s...

Generate fixtures for these test cases for Osaka with:

fill -v tests/osaka/eip7692_eof_v1/eip3540_eof_v1/test_container_validat:

EOF validation tests for EIP-3540 container format

Test Functions Overview

ners

Name	Туре	Cases (Osaka)	Description
test_valid_containe	eof_test	8	Test creating vario contracts using leg
test_invalid_contai	eof_test	107	Test creating vario

contracts using led

Section 4

Running the Tests

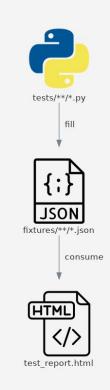
Closing the Loop: consume (protec)

Running a test against a client is a two-step process:

- 1. Generate/fill the tests: JSON.
- 2. Run the tests against a client

We added three ways to run the tests within EEST:

- 1. Consume direct (client's native consumer).
- 2. Consume rlp.
- 3. Consume engine.



There is no Loop: execute (attac)

The next logical step was to extend the framework to allow execution of tests on live networks.

The execute command executes a test case by sending transactions to a clients RPC endpoint.

There is no filling and no JSON is generated.

It unlocks a completely new use case for the execution-spec-tests framework by allowing execution of the tests in a production(-like) setting!





The job's not finished

Improve the lives of client developers:

- Merge ethereum/tests into EEST.
- Provide tools to directly integrate into client team test flows.
- Faster test execution.
- Easier test debugging.

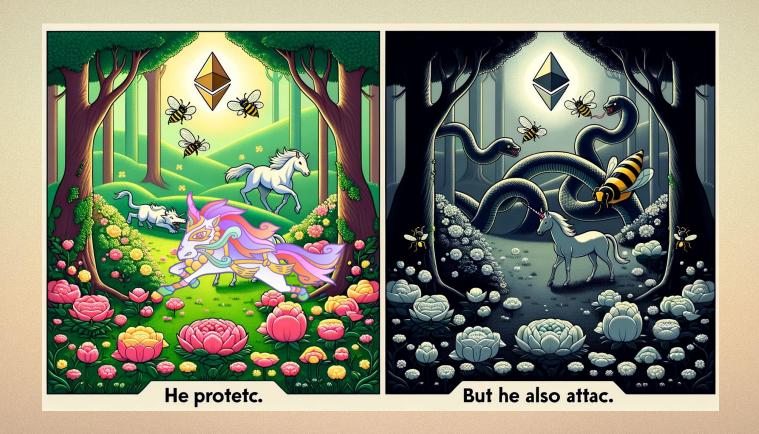
Improve the development cycle:

- Add EIP implementations to the spec first: EELS!
- Add EIP versioning to track implementations/definitions: <u>EIP-7577</u>.

Improve coverage:

- Ability to hook a fuzzer into test execution.
- Better documentation, more eyes on existing test coverage.





Thank you!









Protocol Testers, Ethereum Foundation

ethereum/execution-spec-tests



contact us: repo → docs → getting_started → getting_help