

# Extend Conformance Testing of Consensus Framework to Peras and Leios

## Abstract

We propose extending the [Conformance Testing of Consensus](#) (CTC) framework to support emerging consensus features while ensuring its reliability and usability. The CTC framework represents an initial investment toward a shared, implementation-agnostic conformance testing infrastructure for the Cardano consensus protocol; while still maturing, it establishes the foundations for validating consensus behavior across independent node implementations. This proposal consolidates that foundation by integrating the framework into the ecosystem and improving the testing infrastructure to account for the features of Ouroboros Peras and Ouroboros Leios.

## Problem Statement.

The security and resilience of the Cardano network depend on independent node implementations correctly realizing the Ouroboros consensus protocol. As the ecosystem evolves toward multiple implementations, ensuring consistent behavior under realistic network conditions becomes increasingly important. Consensus correctness therefore depends not only on sound protocol design but also on faithful implementation across nodes.

The Conformance Testing of Consensus (CTC) framework was developed to support this objective. It provides a controlled testing environment that evaluates node behavior through black-box testing under simulated yet valid network conditions. By generating extreme but possible chain structures and modeling network interactions, the framework enables systematic validation of consensus behavior in scenarios that are difficult to reproduce in production.

However, the framework must evolve alongside the protocol. Developments such as Ouroboros Peras and Ouroboros Leios introduce mechanisms—including voting rounds and on-chain certificates—that alter how nodes determine the preferred chain. These behaviors are not yet represented in the current testing machinery, limiting the ability to validate consensus correctness under the full range of protocol scenarios.

At its current stage, CTC provides a strong foundation focusing on Ouroboros Praos and Ouroboros Genesis, but does not yet cover the full spectrum of emerging consensus

mechanisms. Without continued development, the framework risks becoming misaligned with the protocol and remaining limited in scope, preventing broader adoption across implementations.

## Context

While the goal is to ensure that multiple nodes agree on the correct chain, the CTC framework is based on two key insights:

1. Identifying the honest chain locally is complex, but identifying it globally is straightforward. By restricting attention to scenarios where a globally best chain exists, correctness can be determined unambiguously.
2. Given a known global best chain, it is unnecessary to simulate multiple nodes. Instead, a single node can be evaluated against the expected outcome.

The framework therefore relies on a coordinated *point schedule* that simulates multiple upstream peers serving blocks from a predetermined block tree rooted at genesis. This tree defines a canonical best chain while including competing branches representing adversarial conditions.

After executing the schedule, the node under test (NUT) is evaluated based on whether it converges to the correct chain. These are the fundamentals of the “node vs environment” (NvE) tests, originally exclusive to the internal test suite of the ouroboros-consensus library. Its development strengthens both external testing capabilities as CTC infrastructure and internal ouroboros-consensus CI processes.

In its current form, CTC acts as a bridge between protocol-level reasoning and executable validation, but remains under active development. Advancing it to support multiple implementations requires extending its coverage and improving its usability as a general-purpose testing framework.

## Key Proposal Deliverables / Milestones

This proposal focuses on extending the CTC framework to support new consensus features while ensuring the long-term maintainability of the testing infrastructure. The extension of CTC to emerging consensus protocols builds on two different knowledge contexts. For Ouroboros Peras, we leverage direct experience from its design and implementation, enabling us to extend both scenario generators and validation properties with a high degree of confidence and precision. In contrast, Ouroboros Leios introduces substantially more complex and less familiar mechanisms, requiring additional preparation, careful analysis, and iterative refinement to ensure that both

generated scenarios and validation properties accurately reflect its consensus dynamics. This distinction informs the effort estimates of the milestones below.

## Milestone 1: Ecosystem Integration

The foundation of the CTC framework has already been established: most of the required infrastructure changes are complete, some already merged upstream,<sup>1</sup> and the core tool MVPs implemented.<sup>2</sup> This milestone is about integrating this framework into the ecosystem by delivering the complete suite of tools—as defined by our [design document](#)—to the ouroboros-consensus repository, including:

1. A serialization format for NvE tests.
2. A *runner* tool capable of executing serialized tests by simulating upstream peers communicating through the node-to-node [ChainSync](#) protocol.
3. A *testgen* tool for generating serialized NvE tests corresponding to the exposed ouroboros-consensus test suite.
4. A *shrinkview* tool for inspecting non-minimal test inputs.
5. Tools integrating the above for the following workflows:
  - a. Running a QuickCheck-style test-and-shrink loop for a specific test.
  - b. Running the complete test suite against a node without performing shrinking.

Together, these tools constitute the full CTC integration into ouroboros-consensus, providing a self-contained infrastructure for generating, executing, and inspecting NvE tests. Their delivery establishes the operational foundation on which Milestones 2 and 3 build.

*Effort estimate: 2 engineer (FTE) months*

## Milestone 2: Node vs Environment Property Tests for Peras

Extend the CTC scenario generation machinery to incorporate the mechanisms introduced by Ouroboros Peras, following the specification in [CIP-0140](#), then define and implement node-versus-environment property tests for these consensus mechanisms, building on the extended machinery.

This includes updating point schedule generators and related tooling so that testing scenarios represent:

---

<sup>1</sup> Such as the [generalization of the testing infrastructure](#) and the creation of a new [data structure to expose the existing test suite](#).

<sup>2</sup> Currently living in our [cardano-node fork](#).

- Voting rounds and committee selection
- Vote diffusion and quorum formation
- Certificate creation, propagation, and expiration
- Chain selection based on certificate-augmented weight
- Cool-down periods and recovery to Praos-like behavior

while preserving existing correctness properties, such as rollback behavior within the security parameter.

Extending the generators is non-trivial because CTC relies on carefully constructed block trees and schedules that guarantee a well-defined global best chain. In Peras, chain selection depends on both chain length and certificate-derived boosts, where a quorum of votes increases the weight of a block and its ancestors. Additionally, voting is governed by round structure and conditional rules, including cool-down periods when quorums are not achieved.

This ensures that the testing infrastructure can generate representative and valid scenarios for Peras specific behavior, providing the necessary foundation for the development of NvE tests that involve:

- Translating Peras protocol guarantees into executable validation properties
- Defining expected node behavior under voting rounds, vote diffusion, and quorum formation
- Verifying correct handling of certificates, including inclusion conditions and expiration
- Validating chain selection based on certificate-augmented weight
- Ensuring correct transitions into and out of cool-down periods

Designing these properties requires mapping protocol-level specifications into precise, testable assertions. In particular, properties must capture that certificates arise from quorums of votes and increase chain weight, thereby influencing the preferred chain, while also accounting for cases where voting fails and the protocol reverts to Praos-like behavior.

*Effort estimate: 6 engineer (FTE) months*

### Milestone 3: Node vs Environment Property Tests for Leios

Extend the CTC scenario generation machinery to incorporate the mechanisms introduced by Ouroboros Leios, following the specification in [CIP-0164](#), then define and implement NvE property tests for these consensus mechanisms, building on the extended machinery.

In particular, generated scenarios must capture:

- Opportunistic inclusion of endorser blocks
- Timing-constrained certification windows
- Chain evolution under stochastic availability of certified payloads

Unlike Peras, where additional signals deterministically influence chain selection, Leios introduces variability through timing: whether certified data is incorporated into the chain depends on the alignment between endorsement, certification delays, and block production. This requires extending scenario generation to model timing-sensitive behaviors and failure modes, such as missed certification windows and discarded endorsements.

This requires expressing properties that account for timing-sensitive execution, where valid behavior depends not only on structural correctness but also on whether protocol timing constraints are satisfied. Careful alignment with the CIP specification is necessary to ensure that these properties accurately reflect expected node behavior.

*Effort estimate: 12 engineer (FTE) months*

## Resourcing & Duration Estimates

This work is estimated assuming a team of 2 Haskell engineers, resulting in a total duration of 10 consecutive months.

### Milestones

Milestone	Deliverables	Acceptance criteria	Weeks
T6.1 CTC Integration into Cardano Ecosystem	<p>Integration of the Conformance Testing of Consensus (CTC) framework into Cardano repositories.</p> <p>Alignment with existing development workflows and tooling.</p> <p>Documentation describing how CTC is used within the ecosystem.</p>	<p>CTC framework is merged into relevant Cardano repositories.</p> <p>Tests can be executed within the standard development or CI workflow.</p> <p>Documentation is available describing usage and integration.</p>	4

<p>T6.2 Executable Consensus Tests for Peras</p>	<p>Extended CTC generators capable of modelling Ouroboros Peras features.</p> <p>Executable properties validating Peras consensus behavior, ensuring correct handling of voting, certificates, chain selection, and protocol fallback conditions.</p>	<p>Code is reviewed and accepted for merge into the relevant repositories.</p> <p>Tests execute successfully and validate expected consensus properties.</p> <p>Public demo demonstrates test scenarios and results.</p>	<p>12</p>
<p>T6.3 Executable Consensus Tests for Leios</p>	<p>Extended CTC generators capable of modelling Ouroboros Leios features.</p> <p>Executable properties validating Leios consensus under its timing constraints, ensuring correct behavior under stochastic certification outcomes and protocol-specific dynamics.</p>	<p>Code is reviewed and accepted for merge into the relevant repositories.</p> <p>Tests execute successfully under defined scenarios.</p> <p>The public demo demonstrates correctness and behavior under test conditions.</p>	<p>24</p>