

How Yield Protocol Uses Tenderly to Reduce Development Friction and Fortify Security Operations



Organization: Yield Protocol

Website: yieldprotocol.com

Industry: Decentralized Finance (DeFi)

Key Challenges

- Overcome the limitations of running tests in unreliable testnet environments
- Ensure the highest levels of protocol security when implementing governance proposals
- Reduce infrastructure management overhead

Key Results

- 40 hours of engineering time saved per week on infrastructure management
- Faster and more secure implementation of protocol changes without manual intervention
- Monitoring 100+ smart contracts deployed on the Mainnet with dozens of custom Tenderly Alerts

Fixed-term, fixed-rate borrowing and lending on Ethereum

Yield is an Ethereum protocol that brings fixed-term, fixed-rate lending, and interest-rate markets to decentralized finance. Yield Protocol partnered with Tenderly to decrease development friction while ensuring the security and transparency of its governance proposals and protocols.

High-stake challenges call for brave solutions

As a global FinTech organization, Yield Protocol manages digital assets worth \$15 million for its nearly 10k users. Operating in the field of high-stakes decentralized finance, the engineering team at Yield Protocol faced three unique challenges:

1. **Unreliable and unstable testnet environments:** The Yield Protocol team spent countless engineering hours developing protocol upgrades and deploying them to a testnet before a big release, only to get unreliable results due to the testnet failing. This has caused the team to waste precious time on fundamental infrastructure operations instead of focusing on developing the core product.
2. **Governance proposals and protocol security:** Even a minor mistake in the governance proposal could cost users millions of dollars if implemented on the protocol. Given the high stakes, the Yield Protocol team had to focus on enhancing security and visibility into its governance proposals. They also had to account for potential insider threats and establish testing, debugging, and deployment procedures to eliminate the risks of a rogue employee escaping with funds.
3. **Infrastructure management overhead:** To run tests against actual Mainnet data, Yield Protocol engineers had to fork the Mainnet on their local machine. This approach was time-consuming and produced inconsistent results. Engineers also faced challenges sharing their local forks with other team members while ensuring a consistent environment during testing.

Tenderly as Yield's development backbone

Tenderly provides the Yield Protocol engineering team with a comprehensive development, testing, and debugging platform designed to support them at critical stages of the development lifecycle. The Yield Protocol team heavily relies on several Tenderly tools, mainly Debugger, Forks, Alerts, and Simulator.

Complete visibility into code & transaction execution

The Yield team uses the Debugger to inspect suspicious or failed transactions. The Execution Trace, State Changes, Gas Profiler, and other Debugger features give the Yield team unmatched visibility into smart contract behavior and transaction failures. Whenever a hack occurs or a user's funds go missing, Yield's customer support turns to the

“We use Tenderly Forks as the backbone for our development. I don't know if you could say that something else could create a bigger business impact than that.”

Alberto Cuesta Cañada,
Lead Engineer at Yield Protocol

Debugger for the most accurate insights into what happened at the code level. This information plays a critical role in Yield's problem-solving process and underpins key decisions in crisis situations.

“Debugging transactions without the Tenderly Debugger is close to impossible.”

Zero time wasted on infrastructure management

Tenderly Forks allow the Yield Protocol team to focus on providing a stellar user experience rather than wasting valuable time managing infrastructure. Working with over a dozen Forks, engineers are able to ensure the integrity of smart contract fixes and changes to the protocol. The Forks provide a simple way to run testing scenarios against real Mainnet data. Forks are also sharable, which allows the entire team to run tests separately in identical environments with accuracy and consistency.

“We use Tenderly Forks as the backbone for our development. I don't know if you could say that something else could create a bigger business impact than that.”

Collaborative testing in reliable & consistent environments

Together with Forks, the Yield Protocol team uses the Tenderly Simulator to test multiple scenarios and dry-run transactions before executing them on-chain. This helps the Yield Protocol team build confidence knowing that code changes and transactions behave as expected before deployment. All simulations are executed in isolated environments which can be customized to fit Yield's specific use case or predefined QA scenarios.

Early vulnerability detection with real-time alerting

The Yield Protocol team uses Tenderly's customizable alert triggers to keep tabs on the stability of the platform and the security of users' funds. The team relies on dozens of custom real-time Alerts for monitoring activity on more than 100 smart contracts deployed on-chain. Example Alerts used by the Yield team include: notify us when a user transaction fails, the user gets liquidated, the governance layer goes through any phase, or when funds appear in unusual places, etc.

We can do this!

Before Tenderly, the Yield Protocol team used testnets for QA and testing. This caused serious productivity and security issues in the team's development pipeline. In addition to unreliable testnet data, the team was also bogged down with managing infrastructure, which prevented them from focusing on developing their core business.

The Yield Protocol engineering team found a solution for these problems in Tenderly Forks.

“When we realized that we could use Tenderly Forks to get rid of testnets, that was one of those 'aha' moments. We were getting bogged down by having to maintain our own infrastructure in testnets. That was a huge sink of time for us because it was unreliable and not working at all. The fact that Tenderly came out with Forks at that time, we were excited and thinking, 'Oh, we can do this.'”

Evolving and growing at the cutting edge

The Tenderly platform provides the Yield Protocol engineering team with critical blockchain development tools that work together seamlessly. Having started using Tenderly when it was only a tool for transaction debugging, Yield's product has evolved and grown in tandem with the Tenderly ecosystem.

“We started using Tenderly when you were doing one thing – transaction debugging. As you released additional features, we started using them. We are very happy with Tenderly, even when we're at the cutting edge using the latest features still under development. I don't think there is any realistic alternative to Tenderly elsewhere.”

The key driver of Tenderly's growth and success is continuous support for community-driven feedback. Along with hundreds of other customers, the Yield Protocol team has been a source of actionable feedback for Tenderly's development team. These valuable insights have helped Tenderly fine-tune its product offering to focus on bringing real value to customers like Yield by explicitly addressing their unique business needs.