tenderly

Customer Success Story

How Tenderly Improves the Quality of Code and Life for Yearn Finance Developers

yearn

Company: Yearn Finance
Website: https://yearn.finance/
Location: Decentralized
Industry: Decentralized Finance (DeFI)

Key Challenges

- Challenging and limited testing capabilities
- No in-depth insight into complex transactions
- Inefficient debugging with a limited way to simulate transactions prior to deployment

Key Results

- Fewer bugs in production thanks to tx simulations
- Improved testing and code quality with cross-team sharing
- Enhanced developer experience thanks to all-in-one support and integrated Tenderly Web3 Gateway

A DeFi innovator from the start

As one of the veterans in decentralized finance (DeFi), Yearn Finance has played a crucial role in forming the industry. The well-known protocol enables users to participate in otherwise complex processes without requiring advanced proficiency in development or financial analysis. Through automated investment strategies, Yearn Finance truly supports innovation in decentralized finance, allowing anyone to take part and optimize their returns on investment.

The protocol is based on smart contracts that execute elaborate investment strategies. These contracts use yVaults, a type of liquidity pool, to mint yTokens. Then, they use these tokens for further lending, borrowing, and other yield farming strategies.

To ensure their users receive optimal returns, Yearn developers need to create foolproof strategy contracts for automated yield generation. So, Yearn engineers across smart contract, DevOps, and web development teams work together to write these strategies, provide excellent user experience, and ensure the safety of user funds.

Limited testing and simulation capabilities

Writing strategy smart contracts requires a lot of rigorous testing from the Yearn development team to ensure all yield-generating strategies provide optimal results. However, testing initially came with its own set of limitations. It was difficult for Yearn developers to see exactly what was going on once a string of complex transactions was triggered.

Yearn engineers also needed a way to share test results with team members so they could all confirm the correctness of the code. Yet, with fragmented tooling and limited capabilities, it was difficult to keep everyone at the same level of involvement.

Additionally, Yearn developers needed to understand issues once strategy smart contracts were in production. For teams to resolve problems efficiently, they needed to recreate them to get an in-depth insight into executed transactions that usually involved multiple investment strategies.

Tenderly as an all-in-one platform for effective issue resolution

The need for high-performing tooling with comprehensive support brought Yearn Finance to Tenderly. Now, the integration spans different levels and includes Tenderly Infrastructure 3.0 as an integral part of the Yearn team’s development and monitoring processes. While Tenderly Web3 Actions allow Yearn developers to automate responses to on-chain changes, Tenderly Web3 Gateway gives them more than just reliable blockchain access.

“There’s not a lot of competition on the node-as-a-service market, not at the same level. Unlike other node providers who treat the EVM as a black box, the guys from Tenderly know exactly what's going on inside the EVM and they know how to use it for simulations. It's a different level of understanding of the problem”, says @poolpitako, a software engineer at Yearn Finance.

> "There's not a lot of competition on the node-as-a-service market, not at the same level. Unlike other node providers who treat the EVM as a black box, the guys from Tenderly know exactly what's going on inside the EVM and they know how to use it for simulations. It's a different level of understanding of the problem."
>
> **@poolpitako,**
> Software Engineer, Yearn Finance

Additionally, thanks to Transaction Simulator, Debugger, and Gas Profiler, the Yearn development team has a deeper understanding of what went wrong after transaction execution. They can now recreate problems easily and gather in-depth data on transaction execution and gas usage.

What started as a single simulation quickly became a testing ground for complex scenarios thanks to Tenderly Forks. Forks enable the Yearn engineering team to preview the execution outcomes of intricate yield-generating strategies. The Fork environment also allows them to share results across teams, perform white hat-like activities, and even detect exploitable contracts.

Even in time-sensitive situations, Tenderly is there to support Yearn engineers. Yearn Finance has a set of emergency procedures in place to deal with hacks and exploits, assembling a war room with industry experts. "Every time we'd have these war rooms, we'd try to have someone from Tenderly in those processes because it helps us get the answer as fast as possible", explains @poolpitako.

A game-changing development platform

Without siloed tooling, the Yearn Finance team has been able to eliminate inefficient debugging and development processes, reduce infrastructure management overhead, and ensure excellent developer experience. Yearn engineers no longer need to invest valuable resources in developing tools. They can turn to Tenderly for debugging, testing, simulations, and infrastructure and focus on writing advanced yield-generating strategies.

Instead of lengthy debugging and limited insight into complex transactions came fast and highly performant simulations. "After being able to see a simulation in Tenderly, it was night and day because we get more information and it's easier to debug what happened", says @poolpitako.

Additionally, after successfully fixing a bug, Yearn developers can now share solutions with their team members using Tenderly Forks. As a shared development and testing environment, a Fork enables Yearn engineers to batch multiple simulations, test complex yield-generation strategies, and share both the process and the outcome.

This way, Yearn Finance improves collaboration across teams, supports knowledge-sharing, ensures code quality and consistency, and prevents future issues. As @poolpitako explains: "We understand better what's happening in our code and infrastructure when using Tenderly. I'm happy to say that we have been having fewer bugs in production because we have more complex simulations happening before it."

Simulations as an entry point for a deeper understanding of transaction execution

Tenderly is now an important step even in onboarding new Yearn Finance developers. Every new team member gets access to a Tenderly project where they can play around with simulations for a deeper insight into transaction execution, outcomes, and possible issues.

And as @poolpitako explains, Tenderly should be an important entry point for smart contract engineers in general. "You need to start learning how to read Tenderly before being able to read Etherscan. If a transaction hits Etherscan, that means that it's already too late and it's in prod, already executed or minted. I think the best entry point is there, seeing a simulation of a multi-sig transaction, seeing what happens, trying to understand it, and playing with the transaction."

For Yearn Finance, integrating with the Tenderly development platform was an investment worth making. They will continue to implement Tenderly solutions with successful collaboration and support. "If there's something that doesn't work as we expect or we find a bug, it's honestly fixed between 10 and 15 minutes, which is the craziest, fastest type of response that I think I ever had in my life."