tenderly

Customer Success Story

How NFTX Reduced Smart Contract Debugging Time by 80% with Tenderly

NFTX

Company: NFTX
Website: nftx.io
Location: Vancouver, British Columbia
Industry: Decentralized Marketplace

Key Challenges

- Countless engineering hours wasted on manual debugging
- Inability to simulate complex scenarios on testnets against accurate Mainnet data
- Siloed development tooling impeding team collaboration

Key Results

- 80% less time spent on manual debugging
- Eliminated guesswork from testing with simulations against real Mainnet data
- Improved customer satisfaction resulting from faster ticket resolution

Creating liquid markets for illiquid NFTs

Launched in 2021, NFTX is a decentralized community-led liquidity protocol for NFTs built on Ethereum. It enables users to deposit NFTs into vaults and receive ERC20 tokens that can be used to create liquidity pools. In numbers, the protocol has $25 million in Total Value Locked, over 40,000 NFTs, and over 3,000 users staking in the protocol. NFTX's mission is to become the primary issuer of NFT vault tokens, allowing anyone to trade and invest in NFT markets.

Wasting countless engineering hours on debugging

NFTX runs on a complex architecture consisting of multiple smart contracts and external contracts that interact with other protocols. At its core, the NFTX smart contract uses an upgradable proxy controlled by community members of the NFTX DAO. Only token holders can change the smart contract upon coming to a consensus.

In developing and maintaining this complex architecture, the NFTX engineering team had to reduce time wasted on manual debugging and focus on improving their release velocity. Manually sifting through console logs consumed countless hours of engineering time.

The team used console logs as the primary debugging method. This approach produced unreliable and incomplete results, leaving the team without the valuable information needed to drive key decisions in terms of security and reliability. As Kiwi, a Solidity developer at NFTX explains, the team resorted to “trial and error,” trying to guess the results of code execution and relying on hunches.

“I would create a fork and then run it locally and then just go with trial and error, run it again, put a different console log here, a console log there… And even after spending an hour or two hours trying that, I still don’t get enough information to figure out the problem,” says Kiwi.

Another challenge the NFTX team faced was the inability to replicate on-chain issues in local environments due to the many moving parts making up the entire system. The error messages generated by third-party tools didn’t provide enough data to understand the issue and implement appropriate fixes. The team needed a solution that would allow them to test different scenarios by simulating the outcome of transactions in a private environment against real Mainnet data.

Siloed tooling and the lack of team collaboration features in other Web3 development tools also caused bottlenecks in testing and customer support. The team needed an integrated Web3 development platform that enables seamless team collaboration during development while addressing customer-facing issues.

Eliminating the pain of debugging in Web3

The Tenderly development platform is vital to enabling NFTX to push error-free code to production faster and reduce engineering hours spent on debugging. Tenderly Debugger is the go-to solution for driving key security and system reliability decisions. In addition, the NFTX team uses Debugger to verify deployments and tackle issues post-deployment.

> "God bless Tenderly for saving my life. Tenderly saved me hours and hours on debugging issues that I just could not wrap my head around."
>
> **Kiwi,**
> Solidity developer at NFTX

80% less time spent on manual debugging

According to Kiwi from NFTX, debugging smart contracts and transactions without Tenderly is "painful and pure torture."

Without Tenderly, getting to the bottom of a single issue could take up to three hours. Tenderly has helped NFTX reduce this time by 80%. By speeding up debugging, NFTX was able to accelerate its “time to solution” and push out products and fixes faster and with more confidence.

"In some cases, it's been an 80% improvement over what we did before during debugging. Without Tenderly, some things could have taken us two or three hours. The fact that I can go into Tenderly to see exactly what happened makes debugging incredibly easy," explains Kiwi.

Removing the guesswork from testing with simulations

Since NFTX interacts with numerous protocols, the engineering team has to simulate transactions against a replica of Mainnet data. In some instances, NFTX needed to test against bugs found on the Mainnet, which is impossible with testnets. Tenderly Forks provide the NFTX team with a copy of the most up-to-date Mainnet data, enabling them to simulate transactions as if they were on the Mainnet.

"Maybe there's a bug on the Mainnet that we want to ensure is solved after an upgrade. We use Tenderly Forks to fork the network with the existing bug, apply the upgrade, and check if the bug is still reproducible after the upgrade. It's a nice way to understand how the fix would actually work in the Mainnet environment beforehand," says Kiwi.

Kiwi explains that “life was painful” for Web3 developers before Tenderly released Forks.

Enhanced team collaboration to improve customer satisfaction

Tenderly’s built-in team collaboration features empower NFTX to “create a better product and improve user experience." Tenderly Debugger, Forks, and Transaction Simulator help NFTX engineers collaborate and share results to address user-reported issues promptly.

"Any time we need to investigate why a transaction is failing that a user brings up to us, the first thing our support team does is check it on Tenderly to investigate the failure; if it's something alarming or a normal thing," says Kiwi.

Tenderly Debugger allows the team to inspect the execution trace in great detail, leave comments on problematic lines of code, make adjustments, and re-simulate the execution against real Mainnet data. Tenderly makes this process streamlined and unified to eliminate the need to use different tools to perform a single action.

Powering product development in Web3

For NFTX, the Tenderly development platform is exceptionally valuable when building Web3 products. For those working on complex projects, especially customer-facing products, Tenderly comes in handy when developers need comprehensive information about why a transaction is failing, or an issue has occurred.

“The Tenderly development platform has extremely powerful tools that show us so much valuable information. It shows us every single thing every step of the way. So when we encounter any kind of issue, our first step is to check it in Tenderly,” says Kiwi.