

Emir Hodzic

Principal QA Engineer

emir.hodzich@gmail.com · +387 62 876 923 · emirhodzic.com · [linkedin/emiride](https://linkedin.com/in/emiride) · [github/emiride](https://github.com/emiride)

Sarajevo, Bosnia and Herzegovina · Remote

SUMMARY

SDET and QA lead with over 10 years in test automation, the last four in Web3 across EVM DeFi and Cardano's eUTxO stack. I build test frameworks from an empty repo (Playwright, TypeScript, real wallets in CI) and I specialize in the flows most suites skip because they depend on a wallet and a live chain. Much of my work is checking the money math: liquidation thresholds, health factors, stress test logic, fees, UTxO integrity.

HIGHLIGHTS

- 10+ years in test automation, the last 4 in Web3 (EVM DeFi and Cardano eUTxO).
- Over 200 bugs reported in the past year, roughly 10-20 of them high severity.
- Releases at my current job wait on my suite. Nothing moves to the test phase until it is green.
- Have built Playwright and TypeScript frameworks from an empty repo at two companies, wallet automation included.

EXPERIENCE

Arkis · Principal QA Engineer

Jul 2025 - Present

DeFi prime brokerage for on-chain leveraged trading, lending, and cross-margin across DEX and CEX venues · Contract · Remote

- Own all of the platform's test automation: the admin portal, the borrower and lender web apps, and the REST and WebSocket APIs behind them. The suites are Playwright and TypeScript, and I maintain the CI that runs them.
- Reported over 200 bugs in the past year, 10-20 of them high severity. Each report included the root cause, whether that was a product defect or a test data issue, not just reproduction steps.
- My suite gates every release. A build cannot move to the test phase until it passes, and the team treats a red run as a real problem because false alarms are rare.
- Run real MetaMask E2E tests in CI against Tenderly forked-mainnet testnets, covering wallet onboarding, dApp connect, network switching, and transaction signing. We vendor and pin the extension because LavaMoat lockdown and upstream MetaMask releases used to break the pipeline.
- Ported the API regression suite (150+ tests) from V2 to V3 endpoints without letting it go red in CI during the cutover.
- Test data is seeded through the API rather than the UI: orgs, users, whitelisted wallets, on-chain funding. A reusable seed cache keeps local iteration fast, and every run starts isolated.
- Reconcile the same value across the UI, REST, WebSocket, on-chain state, and ClickHouse analytics (the helpers use ethers.js/viem). The same tests validate health factor, stress test matrix, and liquidation math against the product's own numbers.
- When production incidents happen I dig through distributed traces and logs to find the cause, then backfill a regression test for it.
- Test the centralized exchange integrations (Binance, Bybit, OKX, Hyperliquid): balances, positions and PnL, read-only account safety.
- Also cover RBAC and policy engine behavior across roles and scopes.

Input Output (IOHK) · Software Engineer in Test

Mar 2022 - Jul 2025

Cardano ecosystem: Catalyst Voices, Metadex (DeFi), and the Lace light wallet · Full-time · Remote

- Wrote the Catalyst Voices test framework (Playwright, TypeScript) from the first commit.
- Containerized the whole stack with Docker networks and Nginx (catalyst-gateway, voices, mock-reviews). Runs stopped depending on shared environments after that.
- Third-party wallet releases kept breaking CI, so I started self-hosting pinned wallet builds in AWS S3. That killed the whole class of flake.
- Caught on-chain bugs, including malformed UTxO structures and registration transactions that over-charged fees.
- Worked directly with Cardano tooling (cardano-cli, Lucid, Mesh) for wallet and transaction test flows.
- As QA lead on Metadex (a DeFi project) I managed vendor teams and built the Playwright integration and E2E suites. To test the DEX logic we deployed custom tokens and liquidity pools on testnet, a full mock DeFi environment.
- On the Lace light wallet I cleaned up WebdriverIO onboarding-test debt and wired Slack reporting into CI, which surfaced blockchain sync issues as they happened.

CodeScene · Test Automation Engineer

Jun 2020 - Mar 2022

Concurrent contract alongside System Verification · Contract · Remote

- Modernized an inherited test library to speed up runs, and set up Cypress automation for an internal web application.

System Verification · Test Automation Engineer

Dec 2019 - Mar 2022

QA on a job marketplace product · Full-time

- Set up the QA process and shifted testing left. The automation started as Python, pytest and Selenium, was later rebuilt in TypeScript and Playwright, and ran in GitHub Actions with Slack reporting. Onboarded 3 QAs along the way.

Authority Partners · Full Stack QA Engineer

Apr 2018 - Dec 2019

Regus office-rental platform

- Wrote automated scenarios from feature specs and rebuilt an inherited .NET Framework and Selenium suite on .NET Core and xUnit.

Compusight · QA Engineer

Sep 2016 - Apr 2018

US insurance web applications (Vertafore)

- Regression tested customer-facing web apps and automated an internal tool.

International University of Sarajevo · Teaching Assistant

Sep 2015 - Sep 2016

Computer Science department

SELECTED OPEN SOURCE

TestAutomationBase. A C#/.NET Core UI automation framework I wrote by hand: NUnit and Selenium, Page Object Model, parallel execution, fully Dockerized. It came before my Playwright work and set the patterns I still use.

Open-source tooling. Public repos that run from C#/.NET (Selenium, NUnit) through TypeScript (Playwright, Cypress) to Python, plus CI reporting integrations for Allure, ReportPortal, Zephyr and Slack.

EDUCATION

MSc, Computer Science and Engineering, International University of Sarajevo

2016 - 2019

BEng, Computer Science and Engineering, International University of Sarajevo

2011 - 2015

SKILLS

Primary: TypeScript/Node · Playwright · REST + WebSocket API testing · CI/CD (GitHub Actions) · Docker

Web3 / Blockchain: MetaMask & real-wallet E2E · Tenderly forked mainnet · ethers.js/viem · EVM DeFi (health factor, stress tests, liquidations) · Cardano/eUTxO (cardano-cli, Lucid, Mesh) · Solidity (read/debug) · Transaction & gas validation · CEX integrations (Binance, Bybit, OKX, Hyperliquid)

Working: Python (pytest) · C#/.NET (Selenium, NUnit/xUnit) · Cypress · WebdriverIO · k6 · Dart/Flutter integration tests

Test strategy: API-first test-data seeding · Multi-source data reconciliation · Observability-driven QA (traces, logs, analytics) · RBAC/policy-engine testing · Page Object Model

Tooling: Git · Jira · Postman · Allure · Nginx · AWS S3 · ClickHouse