Accessible Federated Learning: A Student-Centered Framework

RIT

AJ Barea
Rochester Institute of Technology, Rochester, NY, USA
ajb6289@rit.edu

Advisor: Dr. Leon Reznik

Motivation

- This project addresses a common barrier in computational research: the steep learning curve of powerful, developer-centric tools.
- The goal was to make a complex federated learning (FL) framework accessible to student researchers new to the field.
- The original "by researchers, for researchers" design created significant accessibility challenges.

Background

- The project focused on an existing PhD research team's FL simulation framework.
- This framework is a robust Python engine with 10 aggregation strategies, 7 datasets, and Byzantine attack mitigation.
- While functionally solid, it lacked the interfaces, documentation, and test coverage needed for broad student use.

Accessibility Gap

- The core gap was the lack of an intuitive user experience, which hindered student onboarding and experimentation.
- There was no comprehensive test suite to ensure users could trust the simulation results.
- The framework was not accessible programmatically via an API or a web-based interface.

Core Project Goals

- **Trust:** Develop comprehensive pytest coverage to ensure users could trust the simulation results.
- **Accessibility:** Create intuitive interfaces and documentation targeting student researchers new to federated learning.
- **Integration:** Develop a RESTful API layer to enable programmatic or browser-based interaction.
- **Education:** Implement a thoughtful UX design to explain complex FL concepts clearly, guiding users without patronizing them.

Methodology

- **Development:** Utilized an iterative, specification-driven development approach with continuous integration.
- **Testing:**
  - Wrote comprehensive test suites concurrently with feature development.
  - Established a pytest suite with 1,136 tests, achieving over 95% coverage on critical logic.
- **Phased Approach:**
  - **Phase 1:** Built testing infrastructure (1,136 tests).
  - **Phase 2:** Developed a FastAPI RESTful API (12 endpoints).
  - **Phase 3:** Created a React 19 web frontend (50+ components).
  - **Phase 4:** Established developer tooling (Ruff, mypy, pytest-xdist)

Objectives

- Implement pytest tests (unit, integration, performance) to validate combinations of strategies, datasets, and attacks.
- Build a FastAPI backend with endpoints for simulation management, status tracking, and result retrieval.
- Design a React frontend with a configuration wizard, live dataset autocomplete, and interactive Recharts visualizations.
- Author comprehensive guides and implement extensive tooltips to explain complex FL concepts.

Results & Impact

- **For Students:**
  - Reduced onboarding time through comprehensive tooltips and validation.
  - Built confidence in results via the extensive test suite.
  - Enabled rapid experimentation with a web UI, removing the need for CLI expertise.
- **For Maintainers:**
  - The test suite prevents regressions by catching bugs early.
  - The API-first design enables future integrations like Jupyter notebooks.
  - Established code quality standards (zero linter violations) for future contributors.

Technologies

- **Backend:** Python, FastAPI, pytest, Flower.ai
- **Frontend:** React, Vite, Recharts, Material Design 3
- **Data & CI/CD:** Hugging Face Datasets, GitHub Actions

Enhancing the IntelliFL Federated Learning Framework

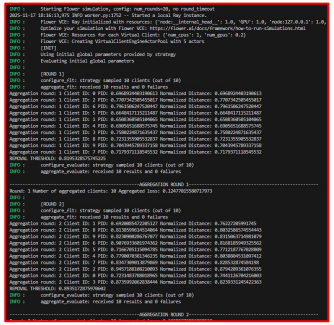
AJ Barea | Rochester Institute of Technology | Advisor: Dr. Leon Reznik

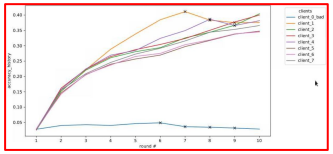
THE CHALLENGE

The IntelliFL framework was a robust Python simulation engine lacking the infrastructure for simple onboarding and reproducible research at scale.

THE TRANSFORMATION

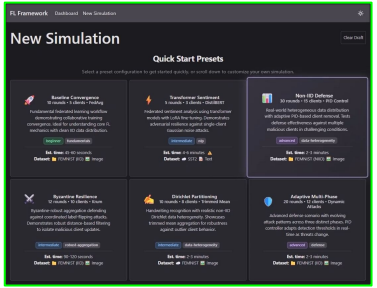
↓ BEFORE: CLI-Only ↓

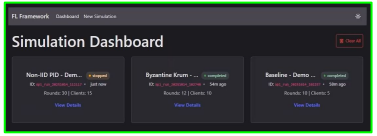


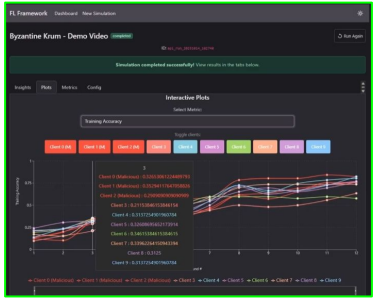


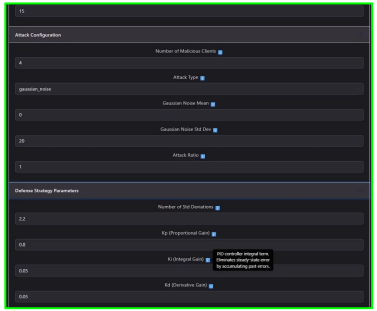
- Manual editing, no pre-validation
- Text logs only
- Static post-analysis review

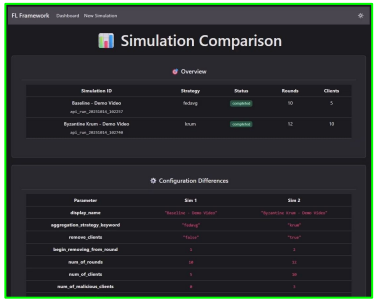
↓ AFTER: Full-Stack Infrastructure ↓

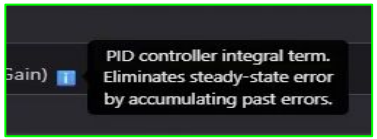


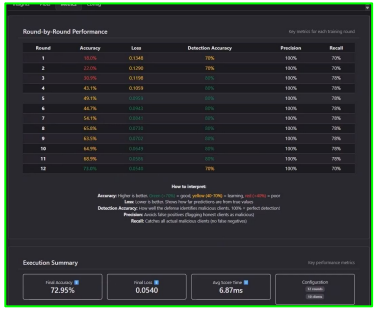












IMPACT

1,136 Tests
Comprehensive pytest suite

95% Coverage
Mission-critical code validated

12 Endpoints
RESTful API + async I/O

50+ Components
React 19 SPA with Recharts

Extensibility
API design enables cloud or automation integrations

Velocity
Reduced test runtime to under 90 seconds

Reproducibility
Ensures trustworthy simulation results

Enhancing the IntelliFL Federated Learning Framework

AJ Barea | Rochester Institute of Technology | Advisor: Dr. Leon Reznik

THE CHALLENGE

The IntelliFL framework is a robust Python simulation engine with **10 aggregation strategies** and **Byzantine attack mitigation**–but it was CLI- only, lacked documentation, and had no testing infrastructure for reproducible research.

THE SOLUTION

Trust: Comprehensive pytest suite ensuring reproducible simulation results

Integration: FastAPI backend enabling programmatic & browser-based access

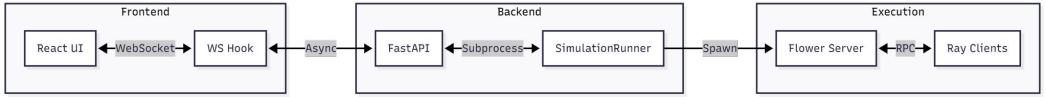
Accessibility: React 19 frontend with guided configuration wizard

Velocity: Modern DX toolchain with parallel test execution

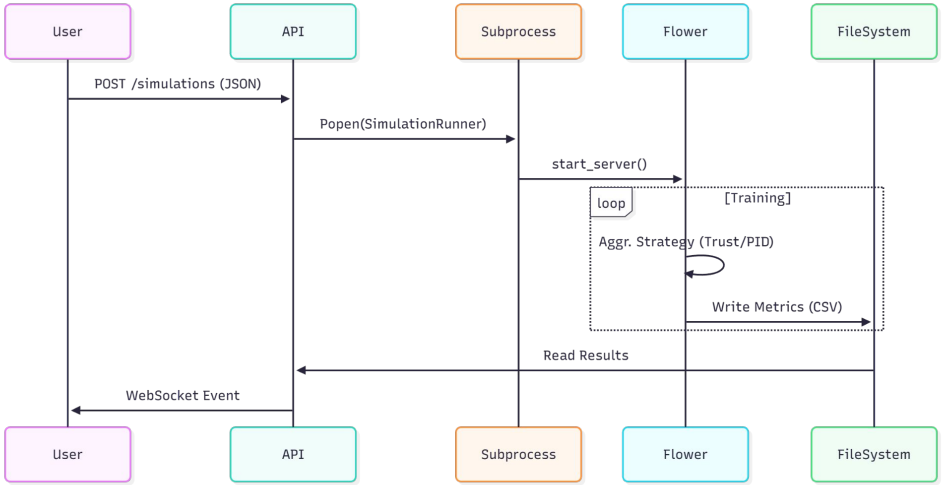
TECHNOLOGIES

Backend: Python, FastAPI, pytest, Flower ai
Frontend: React 19, Vite, Recharts
Data: Hugging Face Datasets
CI/CD: GitHub Actions, CodeCov

SYSTEM ARCHITECTURE



SIMULATION WORKFLOW



IMPACT

1,136 Tests
Comprehensive pytest suite

95% Coverage
Mission-critical code validated

12 Endpoints
RESTful API + async I/O

50+ Components
React 19 SPA with Recharts

Extensibility
API design enables cloud or automation integrations

Velocity
Reduced test runtime to under 90 seconds

Reproducibility
Ensures trustworthy simulation results