

## Project NoéMI — The Blueprint Compiler

A declarative agent runtime, a visual playground, and an LLM arena

*Build the engine that turns nine domains of AI-agent blueprints into running, multi-provider Virtual Coworkers — and ship it into a live public repository under review from professional engineers.*

Submitted by NewPush · Fall 2026 (Aug 24 – Dec 18) · Contact: Balázs Nagy, [bnagy@newpush.com](mailto:bnagy@newpush.com)

|                             |                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Client</b>               | NewPush — AI acceleration, cybersecurity and cloud-infrastructure company (founded 1999, HQ Baltimore, MD)                                                                                                                                                                                                                                                                                           |
| <b>What you build</b>       | The Declarative Blueprint Compiler & Runtime for project-noemi/agents: an engine that parses NoéMI's structured YAML/JSON domain specifications and dynamically instantiates, executes and supervises multi-provider agents — plus NoéMI Studio (a visual agent playground) and the LLM Arena model bake-off suite                                                                                   |
| <b>Repository</b>           | <a href="https://github.com/project-noemi/agents">github.com/project-noemi/agents</a> — public and inspectable under a Fair Source license (Functional Source License), auto-converting to Apache-2.0 after two years. The commercial-grade terms are what keep it adoptable in the regulated and security-sensitive environments our partners operate in; see Licensing below. Node.js 24 baseline. |
| <b>Preferred team size</b>  | 3–4 students (minimum 3)                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Work location</b>        | Remote-friendly; no on-site requirement, so no mileage. Client meets the team at least once per sprint over video.                                                                                                                                                                                                                                                                                   |
| <b>Core stack</b>           | Mastra (TypeScript) · Node.js 24 · React + Tailwind (pre-styled starter supplied) · Docker Compose · GitHub Actions                                                                                                                                                                                                                                                                                  |
| <b>What you take away</b>   | Production-Grade AI Engineering: schema-driven compiler design, multi-provider LLM orchestration, automated evals, CI/CD and secret hygiene — plus a public commit history in a codebase presented each year to senior decision-makers in Europe and the United States                                                                                                                               |
| <b>Mentorship</b>           | Weekly 30-minute sync + a dedicated Slack/Discord channel for daily async support + pull-request review by NewPush engineers who have onboarded 3,000+ students across 100+ cohorts                                                                                                                                                                                                                  |
| <b>Internship potential</b> | Yes — strong students may be offered a paid internship or continued paid collaboration                                                                                                                                                                                                                                                                                                               |
| <b>NDA required</b>         | No. Nothing you write is hidden.                                                                                                                                                                                                                                                                                                                                                                     |

|                 |                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------|
| Authorship & IP | Public commit history and contributor credit under your own name; no IP assignment required |
|-----------------|---------------------------------------------------------------------------------------------|

### Why this project is worth your semester

Most student AI projects are a wrapper around one vendor’s API. This one is a compiler and a runtime.

NewPush maintains **project-noemi/agents**, a public library of AI-agent blueprints spanning nine domains — coding, communication, education, engineering, guardian, infrastructure, marketing, operations and product — together with reusable skills and Model Context Protocol (MCP) definitions. Every blueprint is a structured YAML/JSON specification: a declaration of what an agent is, what it may do, and which skills it composes.

As the repository README says plainly: **“this is not a runtime — it is the blueprint that orchestrators execute.”** Today those blueprints are executed by hand-wired orchestrators (n8n, Gemini CLI, Claude Code, OpenAI Codex). There is no single engine that reads a NoéMI specification and brings the agent it describes to life across whichever model provider makes sense at that moment.

**You are building that engine.** It is a genuine systems problem — schema parsing, validation, dynamic instantiation, provider abstraction, failure handling, supervision — and it lands in a live repository through reviewed pull requests, under a public license, with your name on the commits.

**And it does not stay in the repository.** NoéMI is presented each year at BAISS, the Budapest AI & Security Symposium that NewPush has run since 2023 — held in 2026 as a side event of the Budapest Energy and Security Talks, a conference whose strategic partner is NATO and whose audience is government, foreign-policy and industry leaders from across Europe and the United States. Work merged this semester becomes part of what is shown and discussed there, with your commits attached to it. Few undergraduate projects put your name in front of a room like that.

### Where Mastra ends and your engine begins

A fair question when a framework is named in a proposal: if Mastra already does agents, what is left to build? The honest answer is that Mastra gives you primitives and provider drivers — the low-level plumbing. It does not know anything about NoéMI blueprints. The interesting layer, the one that makes the blueprints executable, does not exist yet, and that layer is your project.

| What Mastra hands you                                                                                                     | What your team designs and builds                                                                                                                |
|---------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Provider drivers and model routing normalized across 40+ providers (Gemini, Claude, OpenAI, Groq/Llama, Mistral, Cohere). | A blueprint compiler: parse and validate NoéMI’s YAML/JSON domain schemas, resolve skill and MCP references, and emit an executable agent graph. |
| Primitives for agents, tools, workflows and MCP servers.                                                                  | A dynamic instantiation layer that constructs Mastra agents at runtime from a specification — not hand-written agent code per blueprint.         |

| What Mastra hands you                             | What your team designs and builds                                                                                                                    |
|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| A built-in evals harness and observability hooks. | A supervision and lifecycle layer: model-selection policy, fallback and retry across providers, timeouts, structured error surfacing.                |
| A TypeScript SDK on Node.js.                      | The developer experience on top: NoéMI Studio for authoring and running blueprints, and the LLM Arena for comparing providers on the same blueprint. |

*Put another way: Mastra is the instruction set. You are writing the compiler and the operating system that sit on top of it.*

### Project background: the multi-provider problem

Teams building with AI must choose among many large-language-model providers — Google Gemini, Anthropic Claude, OpenAI, and open-weight models such as Meta Llama (served via Groq), Mistral, or Cohere. They differ in reasoning quality, cost, latency and privacy posture, and the ranking between them changes every few months.

Most teams hard-wire themselves to one provider, which is both risky and expensive. NoéMI is built on the opposite conviction — **“architecture over tools”** and **“security first”**. Your engine is what makes that conviction real: one declarative blueprint, executable on any provider, with the choice made by configuration rather than by a rewrite.

This is also the reason the blueprints are declarative rather than code. If an agent is data — a specification the engine interprets — then a new provider, a new model generation or a new pricing reality becomes a configuration change instead of a rewrite. That is what lets a single framework serve a small business automating invoice triage and an advanced AI team running a nine-domain virtual workforce, and what lets it keep pace with a field that reorders itself every few months. Designing for that is the architectural heart of this project.

### Company background

NewPush is an AI acceleration, cybersecurity and cloud-infrastructure company founded in 1999, headquartered in Baltimore, Maryland, with operations across the United States, Latin America and Europe. We help organizations adopt emerging technology securely — much of that work in regulated and security-sensitive environments, alongside major enterprise industry players.

Our flagship initiative, **Project NoéMI**, is a global AI-fluency accelerator — academically credentialed by George Mason University — that helps organizations build, govern and deploy a secure “Virtual Workforce” of AI agents. It runs on an open-source stack (NewPush Labs: n8n, Casdoor SSO, Grafana/Loki, Infisical), and the repository you would contribute to is its architectural core.

Teaching is not a side activity for us. **NewPush AI engineers have mentored and onboarded more than 3,000 students across more than 100 cohorts** at universities and private educational institutions worldwide, in cybersecurity, distributed ledger technology and artificial intelligence. The people reviewing your pull requests do this professionally, and they are good at it.

## Core deliverables

Three deliverables make up the minimum viable product. They are deliberately bounded so that a part-time undergraduate team finishes them properly rather than half-finishing five things.

### 1 · The Blueprint Compiler & Runtime

The engine. It loads a NoéMI domain specification, validates it against the schema, resolves the skills and tools it references, and instantiates a running Mastra agent from it — with no blueprint-specific code.

Specifications are loaded from flexible sources, resolved and validated the same way regardless of origin:

- **Local repository files** — the checked-out project-noemi/agents tree, for development and tests.
- **Remote HTTP endpoints** — a specification fetched by URL at runtime.
- **Schema registries** — a versioned registry entry, so a blueprint can be pinned or upgraded without touching the engine.

Model selection is driven by configuration, and the runtime falls back to an alternate provider on failure or timeout. At least three providers must work end to end: for example Gemini, Claude and Llama via Groq.

### 2 · NoéMI Studio — the Visual Agent Playground

The face of the engine: a browser playground where you author a blueprint (name, role, instructions, preferred model, skills), run it, and watch the result stream back. Deliberately scoped as a lean playground rather than a product UI — you receive a pre-styled React + Tailwind starter with the component library, layout and design tokens already in place, so the work is wiring real functionality, not fighting CSS. No custom design system, no bespoke widgets: if a feature needs more than a form, a table and a stream view, it belongs in a stretch goal.

### 3 · The LLM Arena & Model Bake-off Suite

The most fun part, and the reason it is core rather than a stretch goal: once the runtime speaks to several providers, comparing them is a thin layer with an outsized payoff. Send one blueprint and one prompt to Gemini, Claude and Llama simultaneously; show the responses side by side with tokens, latency and cost per provider; keep a simple scoreboard across runs. You end up with a demo that visibly answers a question the whole industry argues about — and a genuinely useful tool for NewPush.

## Production-Grade AI Engineering

These are how the team works from Sprint 1, not chores saved for December. They are also the difference between a class project and an engineering portfolio:

- **Ship through pull requests.** Work lands as small, reviewable PRs following the repository's CONTRIBUTING.md and agent/skill standards. NewPush engineers review every one — real code review from working professionals, on your public commit history.
- **Automated quality gate.** A GitHub Actions pipeline runs linting, unit tests and an automated secret scan (Gitleaks) on every pull request. Green pipeline or no merge. It is pre-wired in the Day 1 boilerplate, so it is live from your first commit.
- **Secret hygiene.** API keys live in environment variables or an ignored secrets file, never in the repository. You will learn why by watching the scanner catch it if you slip.
- **Evals, not vibes.** Agent behaviour is checked with Mastra's built-in evals alongside conventional unit tests, written with each feature. Testing non-deterministic systems is a distinct skill, and this is where you pick it up.

### Stretch goals (once the core is solid)

Each is incremental and ships with its own tests. The security-sensitive ones integrate established frameworks — you are never asked to invent security primitives:

- **Autonomous Guardian Guardrails (AI-monitoring-AI).** Implement the repository's guardian domain as a second agent that reviews the primary agent's output for errors and policy violations before it is returned, using Mastra agents and evals. This is NoéMI's core safety principle, built by you.
- **PII redaction with Microsoft Presidio.** Run Presidio as a containerized sidecar and call it from the runtime to detect and redact sensitive data before any request leaves for an external provider. Integrate the service — do not build detection from scratch.
- **Observability dashboard.** Surface Mastra's traces, token usage, cost, latency and error rates per model and per agent.
- **MCP server.** Expose the compiled agents and tools over the Model Context Protocol so other clients can reuse them, aligning with the repository's existing MCP definitions.
- **NewPush Labs integration.** Wire the runtime into the open-source Labs stack — n8n for triggers, Casdoor for SSO, Grafana/Loki for logs, Infisical for secrets.
- **Multi-agent workflows.** Blueprint-to-blueprint delegation with loop protection, or simple retrieval-augmented generation over uploaded documents.

### Desired skill set

Helpful, not required — students are expected to learn new languages and tools as part of the course, and we plan for that:

- **TypeScript / Node.js** — the primary environment (Mastra, Node.js 24 baseline).

- **React** for NoéMI Studio. The starter is pre-styled, so basic React is plenty; no design skill needed.
- **Comfort with structured data** — YAML/JSON, schemas and validation. If you have enjoyed a compilers or programming-languages course, the blueprint compiler is your kind of problem.
- **Git and pull-request workflows**, REST APIs. Docker and GitHub Actions familiarity is a plus; the boilerplate makes it approachable for newcomers.
- Some Python is useful for the optional Presidio sidecar. Interest in LLMs, agents and secure engineering matters more than any of the above.

No prior experience with Mastra, the provider APIs, Presidio or MCP is expected. NewPush supplies guidance, examples and a working starting point.

## Approximate 15-week plan

The Fall session runs in roughly two-week sprints. Sprint 1 is deliberately modest — getting a real environment green is a genuine milestone, and rushing it is how projects fail in week 9. Scope is confirmed with the team in Sprint 1 and adjusted as the work reveals itself.

| Sprint (≈2 weeks)           | Focus and deliverables                                                                                                                                                                                                                                                                           |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Sprint 1 · Wks 1–2</b>   | Get set up and get green. Clone the Day 1 boilerplate, run a local “Hello World” Mastra execution against the mock provider, and land one small pull request that passes the automated pipeline. Read CONTRIBUTING.md and the domain schemas; agree the architecture with NewPush. Nothing else. |
| <b>Sprint 2 · Wks 3–4</b>   | First real provider calls: multi-provider routing through Mastra with configuration-driven selection (≥2 providers live, mock still available). Unit tests alongside; PRs reviewed by NewPush.                                                                                                   |
| <b>Sprint 3 · Wks 5–6</b>   | The compiler core: parse and validate NoéMI YAML/JSON specifications from local files and remote sources, resolve skills, and instantiate a Mastra agent dynamically from a blueprint. Third provider live; fallback and retry.                                                                  |
| <b>Sprint 4 · Wks 7–8</b>   | NoéMI Studio on the pre-styled starter — author, save and run a blueprint from the browser. Component tests. Mid-project demo of a blueprint compiled and executed end to end.                                                                                                                   |
| <b>Sprint 5 · Wks 9–10</b>  | LLM Arena: one blueprint fanned out across three providers with side-by-side responses, tokens, latency and cost. Persist blueprints and runs. Security review. Core complete and merged.                                                                                                        |
| <b>Sprint 6 · Wks 11–12</b> | Selected stretch goals — typically Autonomous Guardian Guardrails and/or the Presidio sidecar — each with tests. Pipeline stays green.                                                                                                                                                           |
| <b>Sprint 7 · Wks 13–14</b> | Continue stretch goals (observability, MCP server, Labs integration). End-to-end testing, documentation including the repository docs for the new runtime, polish.                                                                                                                               |

| Sprint (≈2 weeks) | Focus and deliverables                                                                                                 |
|-------------------|------------------------------------------------------------------------------------------------------------------------|
| Week 15           | Final delivery and presentation by Dec 18: PRs merged, pipeline green, documentation complete, live demo of the Arena. |

## Mentorship, tooling and guardrails

This is a guided contribution to a real codebase, not a hand-off. We remove the failure modes we can predict:

- **A mature environment, not a science project.** NewPush Labs is a battle-tested platform we run in production, with every stack component you need already configured. You are not the first people to stand it up, and the sharp edges have already been filed off.
- **Day 1 boilerplate.** A working starter on day one: Docker Compose environment, Mastra skeleton on Node.js 24, the pre-styled React + Tailwind Studio shell, and a green CI workflow with secret scanning. You build features in week 1 instead of fighting configuration.
- **Onboarding is our specialty.** Our engineers have brought 3,000+ students across 100+ cohorts up to speed on unfamiliar stacks in cybersecurity, DLT and AI. Nobody is going to lose three days to a broken build while someone shrugs — getting developers productive fast is a thing we are measured on.
- **You cannot burn our budget.** The boilerplate ships with local mock provider endpoints for development and tests, request throttling, and hard daily spending caps per key. Run an accidental infinite loop against a mock and it costs nothing — which means you can experiment without anxiety, and we are not the client who takes the keys away.
- **Weekly 30-minute sync.** A short, focused video check-in each week to set priorities and unblock — at minimum once per sprint, as the course requires.
- **Daily async support.** A dedicated Slack or Discord channel with same-business-day answers from the NewPush team.
- **Pull-request review.** Our engineers review your PRs and mentor you through the repository's standards — architecture, testing and security feedback of the kind you normally only get after being hired.
- **Everything else provided.** API keys and usage credits for all providers, access to the NewPush Labs environment, example blueprints, repository access.
- **Evaluation.** NewPush attends the final presentation if possible and provides end-of-session feedback on the team and on individual students via the course survey.

## Licensing, authorship and confidentiality

Worth reading carefully, because it differs from a standard open-source project — and understanding why is itself part of what you get out of this.

project-noemi/agents is published under a **Fair Source** license (the Functional Source License model), not an immediately OSI-approved open-source license.

### Why not plain open source?

The constraint comes from where this code has to be able to go. NewPush builds for organizations with about the most conservative procurement and compliance rules that exist — large enterprises, and the regulated and security-sensitive programs they and their partners serve. Organizations of that kind are deeply cautious about pure open-source licenses: the concerns are compliance, liability exposure, provenance, and the commercial usability of anything they build on top. A permissive license with no commercial guardrails is frequently not approvable inside them at all. Designing the license around that reality is a requirement rather than a preference.

Fair Source resolves the tension in both directions:

- **Today it functions as a commercial-grade license.** Partners in those environments get the legal certainty and clear commercial terms their procurement and legal teams require — which is what keeps the codebase usable where the stakes are highest.
- **Tomorrow it becomes Apache-2.0 automatically.** A time-delayed trigger written into the license relicenses the code to Apache-2.0 two years after publication. It is automatic — not a promise someone has to remember to keep.

Without this structure the repository would either be closed, and you could not put it on your résumé, or permissively licensed, and unusable by precisely the organizations that make the work consequential. Fair Source is what lets both be true at once.

### What it means for you

- **The code is public and inspectable.** Anyone can read it, learn from it, and use it under fair terms; the restriction is narrow and aimed at direct commercial competition, not at users, students or the curious.
- **No NDA, ever.** Nothing you write is confidential. You can discuss it, blog about it, present it and put it on your résumé from day one.
- **You keep 100% authorship.** Your commits, under your own name, in a public repository, permanently — plus credit in the CONTRIBUTORS file. Your commit history is unencumbered. Where AI assistance is used, the models involved are attributed too.
- **No IP assignment.** NewPush does not ask students to sign over intellectual-property rights.
- **Rare, transferable knowledge.** Fair Source and delayed-open-source licensing are how a growing share of commercial and security-sensitive software is released, and most engineers first encounter them as a surprise on the job. You will have worked inside one and be able to explain the model — including why a company operating under those constraints chooses it — which is a considerably more interesting interview answer than “our repo was MIT”.

### Contact

**Balázs Nagy** — CEO & President, NewPush · [bnagy@newpush.com](mailto:bnagy@newpush.com) · +1.303.423.4500

Happy to talk through scope, adjust the balance between core and stretch, or answer questions from students or faculty before team selection.