



Seaglass Control Center™

One download that installs, configures, and operates the Forge suite.

The Seaglass Control Center is a single installer that is also a living dashboard. One download lays down a private Java runtime and the full Forge suite, makes API-key setup trivial, runs the services, manages data, and lets an operator chat with the agent — **all from one window, with no prerequisites on the target machine.** It is built for people who would rather evaluate a running system in minutes than assemble one over an afternoon.

1 installer

A single NSIS package carrying a private JRE 17 and every suite JAR — no prerequisites on the host.

Zero-friction

No JRE to install, no dependency hunt, no config files to hand-edit. Paste one API key and you are running.

3 modes

Install → Configure → Operate, all in one window — light onboarding, dark dashboard.

Self-contained

A private Temurin JRE 17 and all four suite JARs ship inside the installer. Nothing to pre-provision on the host.

1 key

Paste one API key, live-validated and vaulted; the suite is running moments later.

Install → Operate

The same window that installs the suite also runs it: service control, health, live logs, data, and agent chat.

LIFECYCLE

Three modes, one window

The Control Center collapses installation, configuration, and day-to-day operation into a single application. The UI shifts from a light "front of house" onboarding to a dark "engine room" dashboard once the suite goes live, so the mode you are in is always obvious.

PHASE	SCREENS	WHAT HAPPENS
Install	Welcome · Components · Location · Preflight · Install	Pick components, run live system checks, then lay down a private JRE 17 + the suite JARs + a sample DEM.
Configure	Keys	Provider picker (Claude / OpenAI) with a live-validated key stored in the Windows Credential Manager — never written to disk.
Operate	First Light → Dashboard	Start/stop services, health + live logs, a Data gallery, and an "Ask" chat with the agent.

WORKFLOW

From download to first result

- 1

Run one installer

Forge Suite `<version> x64-setup.exe` — an NSIS installer carrying a private Temurin JRE 17, the four suite JARs, the Studio bridge, and a sample DEM. No prerequisites on the machine.
- 2

Pass live preflight

Free-disk, RAM, GPU, port, and write-permission checks run *before* install; a port already in use can be changed inline, so a conflict never dead-ends the install.
- 3

Paste one key

The key is validated live against the provider's `/v1/models` endpoint, then stored in the Credential Manager and injected into child processes at launch.
- 4

Start the suite & ask a question

One click launches ForgeMind, which spawns ForgeGIS and ForgeData as MCP children — health, uptime, live tool count, and a colored log tail appear at once. The streaming Ask tab shows tool-by-tool progress (`calling forgegis.slope...`) and the answer; a bundled offline DEM means elevation/slope work with no first-run network.

ARCHITECTURE

The suite it manages

The Control Center runs ForgeMind (which spawns ForgeGIS and ForgeData as MCP children) and the opt-in Studio, giving each JVM a **RAM-aware heap cap**. Ports are **configurable** and persisted through every launch and health check, and a one-click **data gallery** fronts 13 ingesters (OpenTopography, Copernicus, STAC, PostGIS, and more).

COMPONENT	ROLE	PORT
ForgeMind	Agent orchestrator (HTTP + MCP host); spawns the others	8080 loopback
ForgeGIS	250+ GPU geospatial ops (MCP stdio child)	stdio
ForgeData	Dataset catalog + ingesters (MCP stdio child)	stdio
ForgeGIS Studio (opt-in)	Web IDE for building and running pipelines	8081 loopback
Seaglass Globe (planned)	3D globe viewer	—

SECURITY POSTURE

- Loopback + token-authed.** ForgeMind's HTTP API binds `127.0.0.1` and requires a per-install bearer token; the agent API is never exposed to the LAN.
- Secrets never on disk.** LLM keys, data-source keys, and the bearer token all live in the Windows Credential Manager; the webview runs under a strict CSP.
- Foundation to scale.** A future multi-user tier (Studio behind a reverse proxy with TLS, auth, tenancy) is a distinct shape — env-driven ports and token auth are its groundwork.

Build from source, or evaluate as shipped

Staging assembles the payload (Maven-built JARs, private JRE 17, Studio bridge, offline DEM) and one `npm run tauri build` produces the installer — end users need none of the build toolchain. Run the whole suite on one evaluation host, disconnected if required, exercising install, configure, and operate through a single window.

Evaluate the Forge suite — see it install and run on one machine.
rich@seaglassfoundry.com