



Seaglass Globe™

The Agent's Presentation Surface — a Globe That Can Be Flown by Software

A precision WGS-84 virtual globe with a native Model Context Protocol server: one POST /mcp endpoint on the running application, JSON-RPC 2.0 over MCP Streamable HTTP, loopback-bound and bearer-protected in the suite install. Clients attach to a running globe rather than launching one — the user owns the window, and the agent joins it. The surface covers the camera, layers and providers, projection, screenshots, scene entities, reports and raster overlays, and the full compute bridge: ForgeGIS™ operations, pipelines and batch runs; ForgeData™ catalog, vector, elevation, ingest and publish. Results come back as things on the globe, not descriptions of them.

/mcp

one endpoint, one transport

MCP Streamable HTTP (rev. 2025-03-26), JSON-RPC 2.0 envelopes on 127.0.0.1:7900; a stdio-to-HTTP adapter for clients, ForgeMind™ included, that speak stdio

2 engines

spawned as stdio children

forgegis-mcp and forgedata-mcp, supervised with restart and backoff; a missing jar degrades the compute tools and never breaks the globe

30 s

per tool call, bounded

a wedged UI thread or child engine returns an error to the caller instead of hanging the agent; a panicking tool is caught, and the window stays up

Attach, don't spawn

The stdio adapter proxies to a running globe's /mcp; the desktop binary under --studio serves that endpoint itself — the same tools and default port as the shell-less server. An agent-launches-the-app model is wrong for an interactive viewer, so the design is the other way round: the port is bound before the shell and the GPU come up, and the suite's Seaglass Control Center™ waits on the port, not the window. Readiness is observable — tools/list, then list_ops for ForgeGIS, then list_datasets for ForgeData — and a tool called early answers warming, not an error.

Results land on the globe

Every compute result is routed by shape. A prose summary and stats always go to the report panel. A GeoJSON FeatureCollection — contours, extracted features — becomes a scene entity, reprojected on the way in. A raster handle is draped: EPSG:4326 as-is, anything else reprojected by ForgeGIS, never in Rust, and if it cannot be, the drape is skipped with an honest note rather than placed wrongly; the file is written as GeoTIFF and composited over the terrain. Anything else reports "ran — see report". analyze_area is the one-shot best-dataset → extract → compute → drape path.

Answers that survive a restart

publish_scene_entity records an answer geometry the operator can reference by name in later turns — site an HLZ as "Site 1", then "route from Site 1". It is durable two ways: injected into the chat scene context on every subsequent query and, when the ForgeData catalog is up, written through as a tagged one-feature GeoJSON, so the entity comes back after a restart with the same id. add_overlay is the transient counterpart, auto-clearing after a TTL. undo_last undoes the last entity change from any source — the agent, a manual edit, or the entities panel.

WHO THIS IS FOR

Teams building on ForgeMind™

The suite's own agent adopts the Globe as its drawing surface at launch: answers that produce something visual — contours, elevation profiles, draped rasters, scene entities — are drawn instead of only described. Ask from the Control Center, from ForgeGIS Studio™, or from the Globe's own chat panel; it is the same agent, and the panel renders only its narration while the camera and overlay changes arrive separately over MCP; the panel's bearer rides each request, never a client default.

MCP-native agent platforms and orchestrators

Anything that speaks MCP over HTTP can drive the running globe — initialize, ping, tools/list, tools/call; notifications answer 202 Accepted. Tool descriptions are hand-written agent-facing prose, because the description is what decides whether the tool gets picked. Every object-shaped result is also sent as structuredContent, the spec's named machine payload, so a client never has to pick a trailing text block by position.

Builders of geospatial agent workflows

fly_to a position with duration, easing, heading and pitch; fit_bounds to a bbox; set_camera to snap. Load a raster into an engine-side handle, run an operation or a saved pipeline against it, drape the result; browse the catalog, pull a windowed extract, publish an answer back. Batch runs a pipeline across a folder of inputs with a job id you poll. It is the same globe the operator sees, not a copy, and a shape they draw by hand is the same kind of entity the agent places.

ON THE WIRE

T+0.5 s

the MCP port is open — bound before the shell and the GPU come up; a busy port is retried every 250 ms for 10 s, well inside the Control Center's 25 s port wait

64 MiB

maximum request body — agent contour payloads run to megabytes, and axum's 2 MB default was not enough for them

3 restarts

per child engine with a 1 s backoff before the compute surface reports itself unavailable; the globe keeps rendering either way

Fire-and-queue, by design. Compute tools return {"ok": true, "queued": true} immediately; the overlay lands a moment later on the UI thread. An agent that expects a synchronous result will look at an empty globe and conclude failure — poll list_handles for rasters or list_scene_entities for geometry, or read the report. Handles are opaque ids for rasters living in the engine's memory; the handle cache is cleared when a child restarts.

Read-back gaps, stated. Agents can set layer state but not yet read it back (list_layers); pick_at and tour/bookmark tools are not on the surface yet; per-tool allow-listing is deferred — the loopback bind and the bearer (constant-time compare, empty 401) are the boundary today, and a hand-launched globe with no token configured has none. Do not expose the port or put it behind a tunnel without setting a token first.

Build on the Globe

The Technical Brief covers the MCP surface, the compute and catalog bridges, the entity model, the digital-twin wire, and deployment; the generated tool reference ships in the suite documentation. Seaglass Globe is an opt-in component of the Forge suite, installed by the Seaglass Control Center, and is not sold separately; the suite's license is not yet finalized. Ask about evaluation access.

rich@seaglassfoundry.com · seaglassfoundry.com