Synaptic Four

White Paper

March 2026

Ferrum: Building GA4GH-Compliant Infrastructure as a First-Principles Practice

How Synaptic Four built its own first product — and what that reveals about working at the intersection of precision, standards, and AI-assisted engineering.

Author: Synaptic Four Technical Team · Stuttgart, Germany · synapticfour.com

Abstract

Ferrum is a full-stack implementation of the Global Alliance for Genomics and Health (GA4GH) standards stack, built by Synaptic Four as its own first infrastructure project. This white paper describes the motivation, architecture, development approach, and lessons learned — with particular focus on how AI-assisted engineering was used as a core methodology, and how the independently developed HelixTest conformance suite validates the system. Ferrum demonstrates that a focused team can deliver a real-world, standards-compliant genomic execution environment. It serves as a concrete example of how Synaptic Four works: with precision, transparency, and a commitment to doing things properly from first principles.

1. Origin and Motivation

1.1 The Problem We Saw

The GA4GH has defined an interoperable suite of APIs for genomic data and computation: TES, WES, DRS, TRS, htsget, and Beacon. These specifications are well-documented and widely referenced. In practice, real-world implementations are rare. Most systems implement only subsets. The result is fragmentation: workflows, data, and execution infrastructure remain siloed, limiting reproducibility and federation across institutions.

GA4GH has itself acknowledged this gap — the ecosystem needs demonstrated, working reference implementations that show the standards operating together in real pipelines.

1.2 The Landscape of Existing Work

Several GA4GH component implementations exist and are actively maintained. TESK (EMBL-EBI / ELIXIR) and Funnel (OHSU) implement TES. cwl-WES (ELIXIR Cloud & AAI) implements WES for CWL workflows against TES backends. WESkit (DKFZ / Sanger

Institute) implements WES for Nextflow and Snakemake in HPC environments. The GA4GH Starter Kit provides individual DRS and WES reference implementations. Galaxy integrates TRS, DRS, and TES across its platform. These are valuable tools — they implement individual components well. What none of them provide is TRS + DRS + WES + TES + htsget + Beacon + Crypt4GH in a single runtime with a unified gateway, shared authentication, and a cross-service conformance suite running in CI.

1.3 Building It

Synaptic Four is a small bioinformatics and software consultancy based in Stuttgart. When we founded the company, we had deep technical knowledge but no client portfolio to show. Rather than waiting for a client to commission this work, we decided to become our own first client. Ferrum became our first project, our proof of capability, and our contribution to the community simultaneously.

> *"We went looking for this tool. We did not find it. So we built it. Ferrum is not a demo — it is our answer to a gap we saw, in a language the GA4GH community speaks."*

2. What Ferrum Is

2.1 System Architecture

Ferrum implements all six major GA4GH components — TRS, DRS, WES, TES, htsget, and Beacon — in a single composable runtime implemented in Rust. The gateway routes all requests, handles authentication, and delegates to the appropriate service. All data is backed by PostgreSQL (metadata) and MinIO/S3 (object storage).

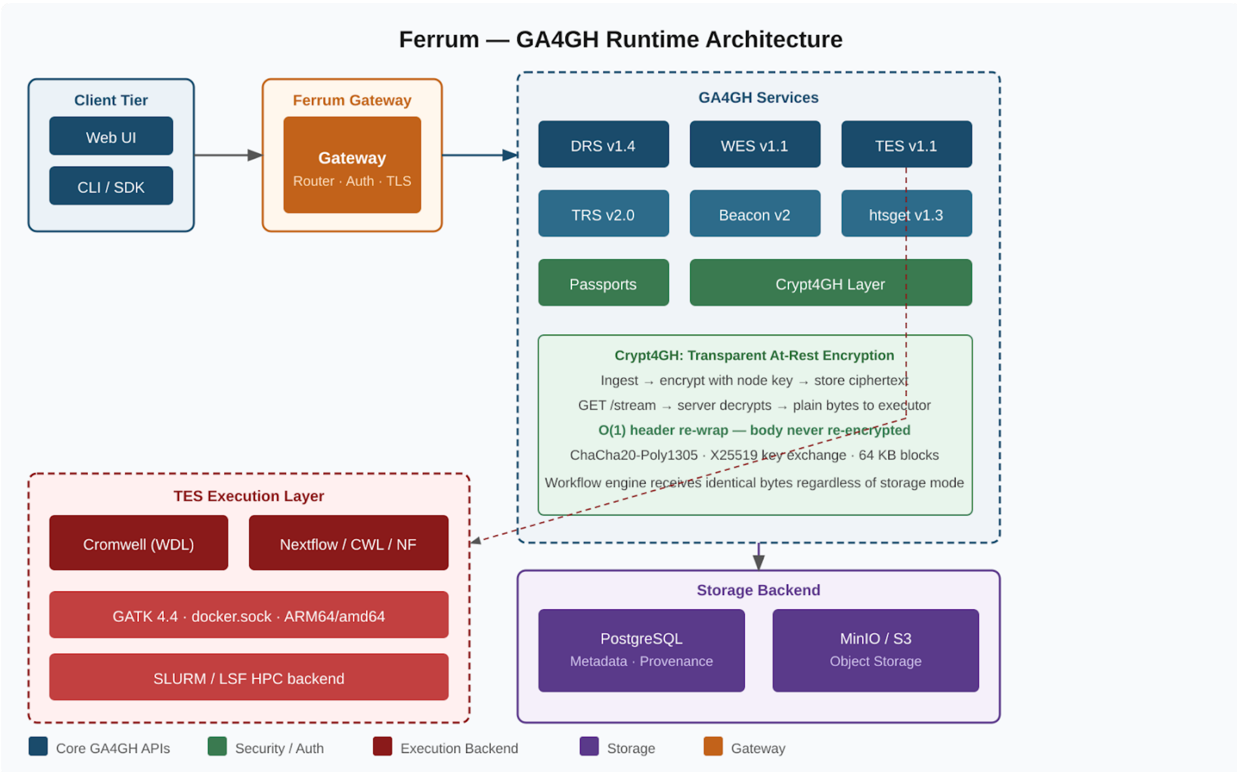

Figure 1. Ferrum full-stack architecture. All six GA4GH API components are integrated under a single gateway, with transparent Crypt4GH encryption at the DRS layer and a pluggable execution backend (Cromwell, Nextflow, CWL, Snakemake, SLURM, LSF).

2.2 Storage Backends

Ferrum supports multiple storage backends. The benchmark runs in this paper use a local MinIO S3-compatible instance. Other supported options include POSIX local filesystem (for on-premises HPC without S3), any S3-compatible provider (AWS, Hetzner, Cloudflare R2, and others), and OpenDAL for Azure Blob, GCS, and additional backends. The storage backend is orthogonal to the GA4GH API layer — workflow executors interact only with DRS endpoints, and the underlying storage is invisible to them.

2.3 Selective Deployment and Ferrum Lab Kit

Ferrum is not all-or-nothing. Each GA4GH service can be enabled or disabled individually. An institution that only needs Beacon v2 for cohort discovery can deploy just Beacon. A lab that needs DRS and WES/TES but not htsget can omit it. Services share the gateway and auth layer when co-deployed but are independently usable.

For smaller labs and ELIXIR node candidates, Synaptic Four maintains Ferrum Lab Kit — a separate deployment and integration layer providing an opinionated on-ramp without requiring the full platform. Lab Kit targets university research groups, GDI national node participants, and GHGA data submitters running SLURM or a single server. It integrates with ELIXIR LS Login out of the box and generates Docker Compose, Helm, or systemd configurations from a single TOML file. Existing infrastructure is not replaced; individual GA4GH services are added on top of it.

> *Ferrum Lab Kit: github.com/SynapticFour/Ferrum-Lab-Kit · On-ramp for labs and ELIXIR node candidates · BUSL-1.1*

2.4 Crypt4GH Integration

Ferrum includes transparent Crypt4GH support at the DRS layer. Files can be ingested with at-rest encryption; the gateway performs server-side decryption on GET /stream, returning plain bytes to any standard DRS client. Encrypted and unencrypted data are indistinguishable from the workflow engine's perspective — encryption is fully transparent to executors.

Header re-wrapping (for per-requester re-encryption on download) operates in O(1) time: only the Crypt4GH header is re-wrapped regardless of file size. A 500 GB BAM and a 1 KB reference file are re-wrapped in the same time.

2.5 Multi-Engine Execution

Ferrum supports Cromwell (WDL), Nextflow, CWL, and Snakemake as WES backends, routed through TES. The same workflow logic, data, and benchmarking infrastructure runs on all engines. HPC scheduling via SLURM and LSF is also supported for on-premises deployments.

3. Validation

3.1 End-to-End Execution Flow

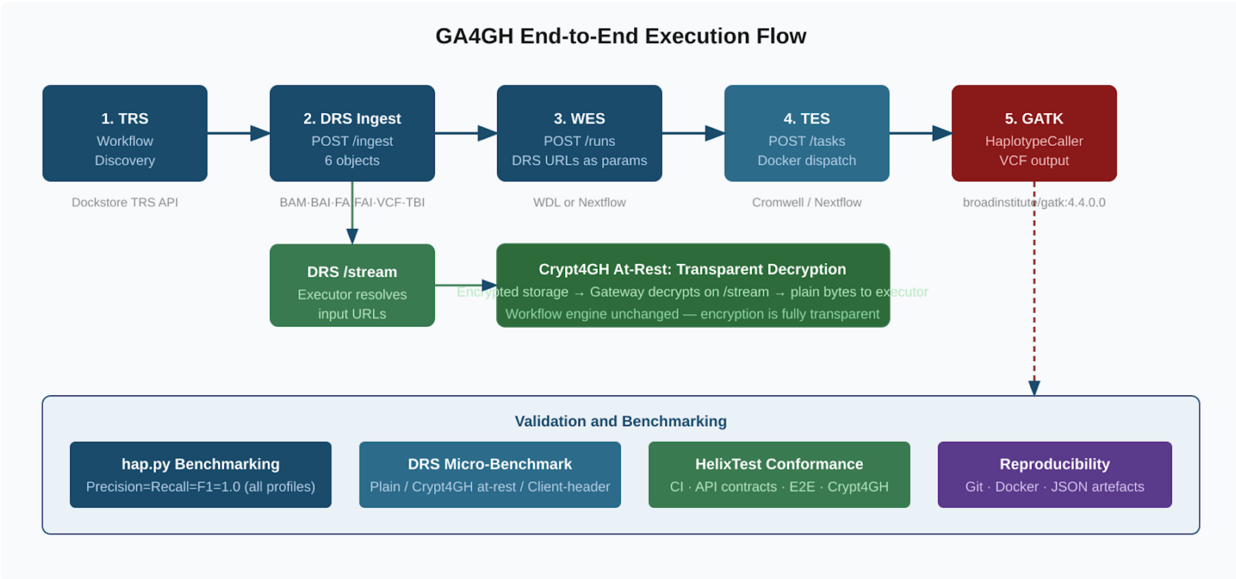

Figure 2. Complete GA4GH execution flow in Ferrum, from TRS workflow retrieval through WES/TES execution to hap.py validation. Crypt4GH decryption is transparent to the workflow engine.

3.2 Benchmark Results

We executed a GATK HaplotypeCaller workflow on a synthetic GIAB-style dataset, benchmarked with hap.py. Results across all profiles (plain ingest, Crypt4GH at-rest, Cromwell, Nextflow) show Precision=Recall=F1=1.0. The Crypt4GH profile adds approximately 10% overhead at the macro level, attributable to ingest-time encryption rather than streaming-time decryption.

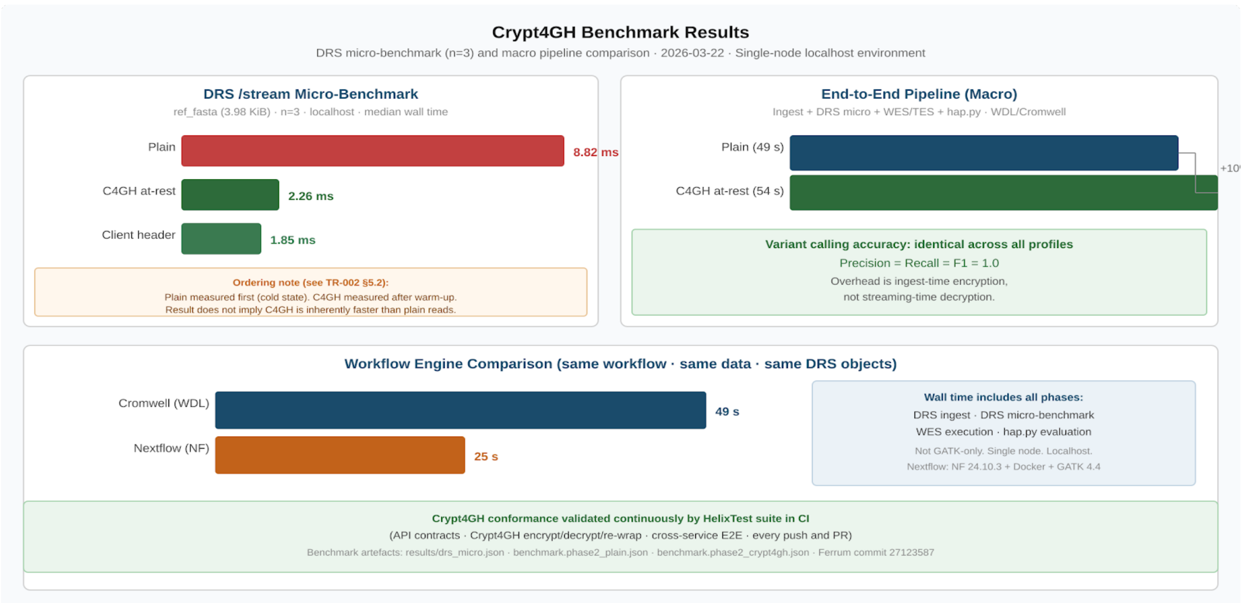

Figure 3. Benchmark results summary. DRS micro-benchmark (left): all three modes measured with n=3 repetitions. Macro pipeline (right): end-to-end comparison, plain vs Crypt4GH at-rest. Engine comparison (bottom): Cromwell vs Nextflow on same workflow and data.

Profile	Engine	Wall Time (s)	Precision	Recall	F1
Plain ingest	Cromwell/WDL	49	1.0	1.0	1.0
Crypt4GH at-rest	Cromwell/WDL	54	1.0	1.0	1.0
Plain ingest	Nextflow	25	1.0	1.0	1.0

4. HelixTest: Conformance as a First-Class Concern

A distinguishing aspect of Ferrum's development is the existence of HelixTest — an independently developed, open-source Rust-based GA4GH conformance suite maintained by Synaptic Four. HelixTest runs on every push and pull request in CI, validating Ferrum against a live stack that includes Postgres, MinIO, Keycloak, and seeded test data.

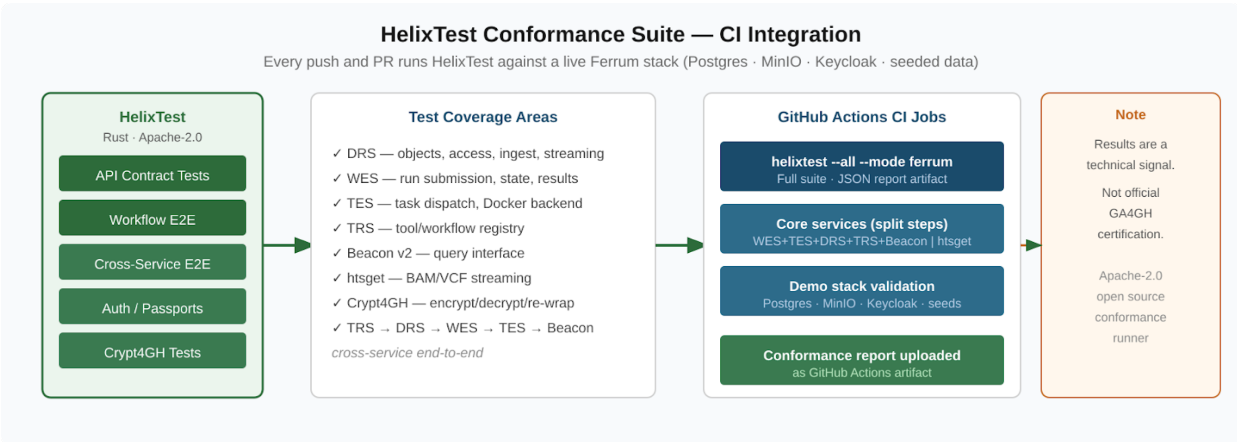

Figure 4. HelixTest CI pipeline. Every commit triggers the full conformance suite covering API contracts, workflow E2E, cross-service integration, authentication, and Crypt4GH. Results are uploaded as GitHub Actions artefacts.

HelixTest covers all major GA4GH API areas: DRS, WES, TES, TRS, Beacon v2, htsget, GA4GH Passports, and Crypt4GH cryptographic tests. The cross-service end-to-end tests span the complete TRS → DRS → WES → TES → Beacon chain — exactly the integration gap that Ferrum exists to address.

HelixTest is released under Apache-2.0. It is a standalone tool usable by any GA4GH-compatible platform, not just Ferrum. This is a substantive contribution to the broader ecosystem: a reusable conformance runner that makes GA4GH compliance testable in CI by any team.

> *Note: HelixTest results are a technical signal, not official GA4GH certification. Certification requires separate assessment by GA4GH.*

5. How Synaptic Four Works

5.1 AI as a Tool, Not a Shortcut

Synaptic Four uses AI-assisted engineering as a core part of its development practice. We believe transparency about methodology is a professional obligation. AI tools accelerate scaffolding of standards-compliant components, help navigate large specification documents, and assist with code generation for repetitive but precise tasks — such as implementing GA4GH API endpoints that must conform exactly to a schema.

What this does not mean: AI makes architectural decisions. Every integration point, every design choice, every validation step in Ferrum was explicitly reasoned through and verified. The AI is a fast colleague that writes code; the engineering judgment is ours.

> *"AI is a tool. The responsibility for correctness, design, and consequences remains with the engineer. We hold that line clearly."*

5.2 Precision and Transparency

Our public benchmark results are honest about scope: Ferrum's validation uses a small synthetic dataset. We do not extrapolate beyond what we have measured. We document limitations explicitly. We believe this is the only credible way to operate in a field where reproducibility is a core value.

5.3 Open Source and Licensing

Ferrum is released under BUSL-1.1, permitting non-commercial research use. HelixTest is Apache-2.0. Our benchmarking suite and demo infrastructure are openly available on GitHub, including the full reproducibility bundle with Git commit hashes, Docker image versions, and structured JSON artefacts.

6. Conclusion

Ferrum demonstrates that a unified GA4GH-compliant genomic execution runtime is achievable by a small, focused team working with precision and the right tools. HelixTest provides the conformance infrastructure to validate this continuously. Together, they represent both a contribution to the GA4GH ecosystem and a demonstration of how Synaptic Four approaches complex infrastructure work.

> *We are open to collaboration, deployment partnerships, and research engagements. Contact us at contact@synapticfour.com or visit synapticfour.com.*

Synaptic Four · Stuttgart, Germany · contact@synapticfour.com · synapticfour.com