



Miloš Vasić

**Portfolio · Helix family · vasic-digital · Server
Factory**

A unified, evidence-based portfolio — the Helix product family, the vasic-digital utility fleet, and the Server Factory toolchain, on a shared engineering Constitution.

140+

REPOSITORY FLEET

43

LLM PROVIDERS

25

LINUX DISTROS
PROVISIONED

439

PASSING TESTS · MAIL
SERVER FACTORY

1. Visión general — el panorama completo primero

Este es un portafolio unificado y único utilizado tanto por vasic.digital como por milosvasic.ru. No describe una colección dispersa de proyectos secundarios, sino una **flota** cuidadosamente diseñada: **aplicaciones de producto a gran escala** construidas sobre **docenas de módulos pequeños, desacoplados y probados de forma independiente**, todo ello regido por una **Constitution** de ingeniería compartida y verificado mediante una disciplina de **control de calidad anti-engaño**. Esa estructura es el elemento diferenciador. La mayoría de los portafolios son listas de cosas que se han construido; este es un sistema en el que cada producto se ensambla a partir de componentes probados y reutilizables, cada componente cumple las mismas reglas innegociables, y cada capacidad anunciada cuenta con pruebas documentadas. El lenguaje predominante es **Go**, con Kotlin/KMP, TypeScript/React, Python, Swift y Shell empleados donde cada uno encaja mejor: Go para servicios y bibliotecas de alto rendimiento, Kotlin para aprovisionamiento y desarrollo multiplataforma en móviles, TypeScript para interfaces tipadas, Python para la integración de AI/ML.

Lo que da coherencia a este conjunto de trabajos es que la disciplina es mecánica, no aspiracional. Una **Constitution** compartida se distribuye como submódulo de Git y se hereda en una flota de más de 140 repositorios, de modo que un único cambio en las reglas se propaga por todas partes; una capa de control de calidad anti-engaño se niega a registrar un aprobado sin pruebas en tiempo de ejecución. La familia de productos **Helix**, la flota de utilidades y el conjunto de herramientas **Server Factory** son tres manifestaciones de una misma idea: construir una vez, reutilizar en todas partes y demostrar que funciona antes de darlo por terminado.

A continuación, todo se presenta en orden de prioridad:

1. **Pilares de gobernanza y control de calidad** — HelixConstitution, HelixQA (la disciplina que hace confiable todo lo demás).
2. **La familia de productos Helix** — el ciclo de vida del desarrollo en AI (HelixTrack primero).
3. **Infraestructura LLM** — abstracción de proveedores, orquestación, verificación.
4. **Utilidades vasic-digital** — herramientas independientes de nivel profesional.
5. **Server Factory** — legado de automatización de infraestructura (último en orden de prioridad).

La tesis unificadora: **una funcionalidad solo está terminada cuando un usuario real puede utilizarla y existe evidencia documentada que lo demuestra.**

2. Pilares de gobernanza y control de calidad

Estos dos elementos van primero porque todo lo demás en este portafolio toma prestada su credibilidad de ellos. Juntos convierten el “confía en mí, funciona” en un hecho auditable: la **Constitution** codifica las reglas, y la **HelixQA** demuestra que se han seguido.

- **HelixConstitution** — un manual de ingeniería universal y agnóstico a proyectos, distribuido como submódulo de Git e heredado en una flota de más de 140 repositorios. Barreras de evidencia anti-engaño, inmunidad a falsos positivos, seguridad de datos y hosts, disciplina de cobertura; herencia de tipo “extender-pero-nunca-debilitar”; barreras de propagación que verifican mediante *grep* la presencia de cláusulas obligatorias en toda la flota; cada barrera acompañada de una prueba de mutación que demuestra que no es un fraude.
- **HelixQA** — orquestación de control de calidad anti-engaño (Go). Bancos de pruebas escritos en YAML, además de sesiones de control de calidad totalmente autónomas con LLM y visión por computadora en Android, Android TV, Web y Escritorio; ningún APROBADO sin evidencia capturada (capturas de pantalla, *logcat*, vídeo, trazas de pila). El tipo de prueba de control de calidad exigido por la **Constitution** (§11.4.169).

3. La familia de productos Helix

La línea Helix es la nave insignia: una familia interconectada de productos que abarca todo el ciclo de desarrollo de AI, desde la planificación y especificación hasta la construcción, la memoria, la traducción y la entrega. Cada uno es un producto real por derecho propio, pero el diseño busca que se integren: el mismo gobierno, los mismos módulos reutilizables, la misma disciplina de evidencia subyacente en todos ellos.

- **HelixTrack** — una alternativa JIRA para el mundo libre; buque insignia de la línea Helix-Track.
- **HelixAgent** — servicio de ensamblaje LLM: múltiples modelos debaten y envían la respuesta en la que coinciden, con selección de proveedores basada en verificación.
- **HelixCode** — plataforma de desarrollo distribuido de AI de nivel empresarial; distribuye el trabajo entre trabajadores gestionados por SSH con puntos de control y retroceso automáticos; interfaces REST, CLI, TUI y MCP.
- **HelixCluster** — un sistema operativo distribuido para cómputo AI, desde GPUs en centros de datos hasta dispositivos de borde, bajo un único plano de control.
- **HelixBuilder** — una canalización impulsada por AI para construir aplicaciones, una categoría a la vez.
- **HelixSkills** — un sistema de habilidades gobernado y respaldado por una constitución para agentes CLI de AI (habilidades, servidores de herramientas MCP, complementos de código Claude).
- **HelixSpecifier** — desarrollo basado en especificaciones que ajusta su ceremonia al trabajo.
- **HelixMemory** — un cerebro de memoria único para agentes AI, que fusiona cuatro motores de primera categoría.
- **HelixTranslate** — traducción de libros con modelos verificados; honesta por diseño, sin retrocesos silenciosos.
- **HelixTerminator** — una plataforma de terminal de confianza cero: cada sesión SSH segura, compartida y asistida por AI.

- **HelixGitpx** — Git federado en una docena de hosts; una única fuente de verdad, replicada en todas partes.
- **HelixOTA** — actualizaciones universales y desacopladas por aire; diseño a prueba de bloqueos.
- **HelixPlay** — convierte cualquier máquina GPU en tu propio dispositivo de *cloud gaming*.
- **Helix-Flow** — producto de inferencia de la plataforma Helix. *SIN VERIFICAR / bloqueado por la fuente: su repositorio público solo tiene un archivo README de una línea; se presentará con profundidad de producto solo cuando exista documentación real.*

4. Infraestructura LLM

Bajo los productos se encuentra el sustrato que los hace independientes del proveedor y confiables: una única interfaz sobre docenas de proveedores LLM, un plano de control para agentes de codificación sin interfaz gráfica y una única fuente de verdad para la verificación. Esta es la capa que permite que todo lo anterior intercambie modelos, sobreviva a fallos de proveedores y se niegue a confiar en un LLM que no pueda demostrar que comprende la tarea.

- **HelixLLM** — un binario, seis modos: inferencia compatible con OpenAI y Anthropic sobre HTTP/3, llama.cpp local, cadena de retroceso puntuada, canalización RAG y agentes ReAct.
- **LLMProvider** — una interfaz, 43 proveedores, con disyuntores, reintentos y monitoreo de salud integrados.
- **LLMOrchestrator** — un plano de control para cada agente de codificación CLI sin interfaz gráfica (OpenCode, Claude Code, Gemini, Junie, Qwen Code).
- **LLMsVerifier** — verificar, monitorear, optimizar: la única fuente de verdad para metadatos de LLM, proveedores y verificación, con una compuerta obligatoria de comprensión del modelo.

5. utilidades vasic-digital

Herramientas independientes de calidad profesional, cada una diseñada para resolver un problema complejo por sí misma —y, no por casualidad, para poner a prueba con rigor el ecosistema de módulos reutilizables—. Van desde un sistema de medios multiprotocolo resiliente hasta un flujo de trabajo de cursos en vídeo a partir de Markdown, pasando por un motor de sincronización de documentos y bases de datos con hash de contenido. Algunas son francas respecto a su grado de madurez, y esas advertencias se mantienen visibles, no se ocultan.

- **Catalogizer** — gestión de colecciones de medios multiprotocolo (SMB/FTP/NFS/WebDAV/local), cifrado y autoalojamiento; Go/Gin API + interfaz React; monitorización resiliente tolerante a desconexiones; construido sobre 21 submódulos `digital.vasic.*`.
- **Courses-Creator** — flujo de trabajo de cursos en vídeo a partir de Markdown; enriquecimiento multiLLM, TTS (Bark/SpeechT5), reproductores para escritorio/móvil/web; modo de funcionamiento elegante sin clave API.
- **VisionEngine** — kit de herramientas Go desacoplado que fusiona visión por computadora clásica con LLM multiproveedor; grafos de navegación con BFS + exportación a DOT/JSON/Mermaid; con puertas de compilación para OpenCV.

- **DocProcessor** — mapa de características a partir de documentación con seguimiento de cobertura de verificación; extracción LLM o heurística/fuera de línea; licencia Apache-2.0.
- **Docs Chain** — sincronización bidireccional y atómica de documentos y bases de datos con hash de contenido (recomputación incremental al estilo Salsa sobre un DAG). *Fases 1–5 VERDES; Fases 6–7 PLANIFICADAS.*
- **Herald** — notificaciones multicanal confiables con resolución de intenciones en tres niveles mediante lenguaje natural (comando → LLM → aclaración); primer consumidor de Docs Chain.
- **task_bridge** — sincronización bidireccional y desacoplada de tareas y tableros (SQLite SSoT [?] documentos [?] ClickUp). *Estructura P1 — lógica de sincronización aún no implementada.*
- **Vasic Digital Suite de Módulos Reutilizables** — la “biblioteca estándar” `digital.vasic.*`: primitivas de infraestructura, bloques de construcción AI y barreras de seguridad con LLM defensivo, además de un espejo Kotlin Multiplatform. *Varios repositorios de la organización están en fase de ESTRUCTURA/EN DESARROLLO — señalizados, no publicados.*

6. Server Factory (herencia de automatización de infraestructura — último en la lista)

Último en la lista por diseño, no por calidad: la cadena de herramientas Server Factory es anterior a la línea AI y muestra dónde comenzó a tomar forma la filosofía de “construir una vez, reutilizar en todas partes”. Su producto estrella —un servidor de correo que se describe en JSON y se aprovisiona en cualquier entorno— es una solución madura y ampliamente probada; el resto de componentes se presentan con su grado real de madurez, sin adornos.

- **Mail Server Factory** — servidor de correo completamente aprovisionado y en contenedores Docker a partir de declaraciones en JSON, compatible con 12 tipos de conexión y 25 distribuciones Linux; seguridad empresarial; informa de 439 pruebas superadas y un paso limpio por la puerta SonarQube. Producto estrella de la organización Server-Factory.
- **Server Factory Marco Central** — el motor Kotlin compartido en el que se basan todas las fábricas.
- **Qemu-Utils** — imágenes de máquinas virtuales QEMU gestionadas como artefactos: descarga/caché/ ejecución, compresión/publicación, redes puente/TAP, instalaciones desde ISO; Linux + macOS.
- **Parallels-Utils** — compresión, publicación y recuperación de imágenes de máquinas virtuales Parallels (macOS) mediante archivos de configuración sencillos.
- **Server Factory — Componentes Adicionales** — fábricas de servicios (Web/SonarQube/Proxy de Caché), paquetes de Definiciones y Utilidades. *Las fábricas de servicios están documentadas como marcadores de posición — SIN VERIFICAR / en fase inicial.*

7. Índice tecnológico (basado en evidencia)

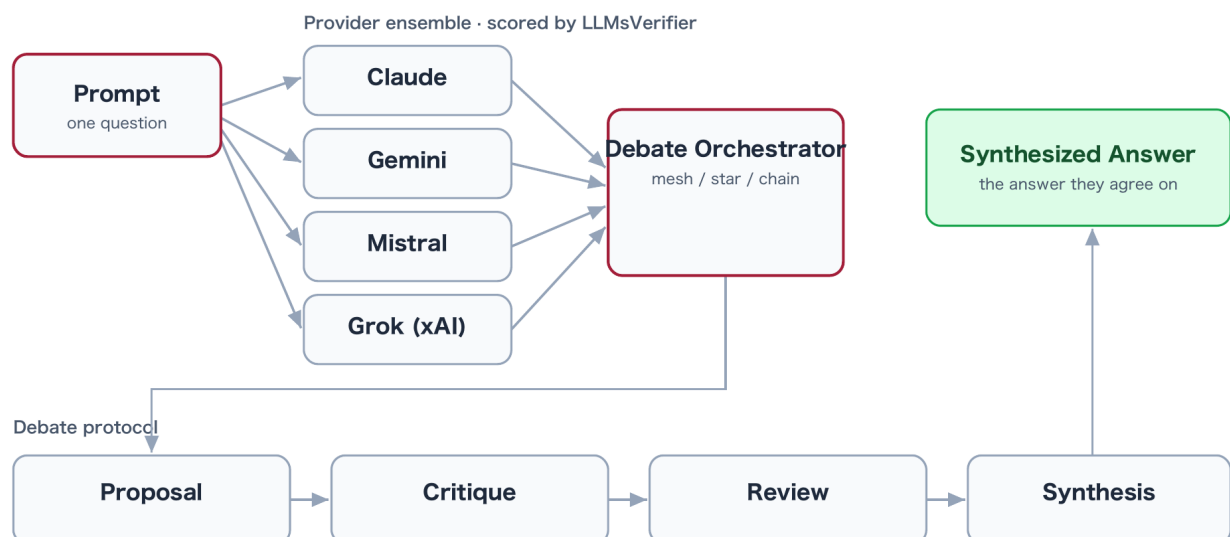
- **Idiomas:** Go (dominante), Kotlin y Kotlin Multiplatform, TypeScript, Python, Swift, Shell; PL/pgSQL; TLA+ (especificaciones formales en *helix_cluster*).
- **AI / LLM:** Acceso a más de 40 proveedores, MCP, RAG, bases de datos vector y embeddings, planificación (HiPlan/MCTS/Árbol-de-Pensamiento), LLMops, evaluación comparativa (SWE-bench/

HumanEval/MMLU), TTS (Bark/SpeechT5), visión por computadora + *LLM-vision*, salvaguardas/*red-team*.

- **Backend:** Gin, gRPC + Protobuf, HTTP/3 (QUIC), WebSockets, Angular, React, Kafka/RabbitMQ.
- **Datos:** PostgreSQL, SQLite, SQLCipher, Redis, Neo4j, ClickHouse, MinIO/S3/GCS/Azure.
- **Infraestructura / DevOps:** Docker y Compose, Kubernetes + Helm, Prometheus + Grafana, OpenTelemetry, QEMU/Libvirt/Parallels; GitHub Actions, Gradle, Make.
- **Pruebas / Control de calidad:** HelixQA, arneses de desafío con compuertas de mutación, `go test -race`, herramientas de regresión visual, pruebas en dispositivos ADB, SonarQube, escaneo de seguridad (semgrep/gosec/trivy/snyk/gitleaks/nancy), verificación de modelos TLA+.

Selected architecture

HelixAgent — ensemble debate flow



HelixAgent — an ensemble LLM service where providers debate under a four-stage protocol and ship the answer they agree on, scored by LLMsVerifier.

HelixCluster — seven-layer stack

14 control-plane
microservices

L7 · Federation & Observability

L6 · Security & Attestation (SPIFFE, PQ-E2EE)

L5 · Sessions & Interactive Terminal

L4 · AI Inference Routing

L3 · Omega Scheduler (two-level)

L2 · Consensus & Membership (Raft + SWIM)

L1 · Node Runtime & Transport (gRPC, WireGuard)

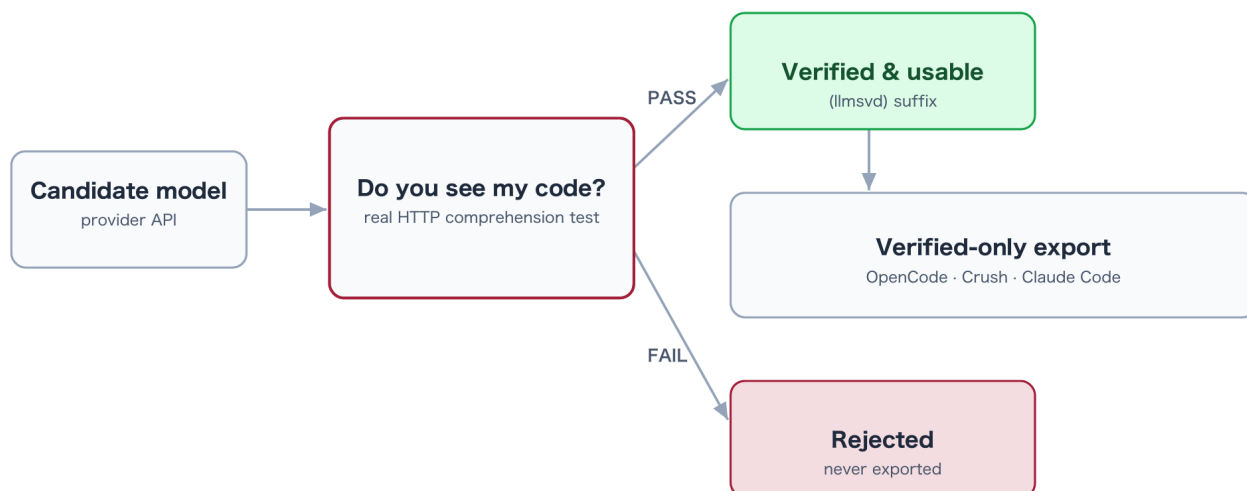
heterogeneous
nodes T1–T8

L0 · Hardware Substrate (GPU → edge SBC)

Node tiers T1 (datacenter GPU) → T8 (handheld), unified under one control plane

HelixCluster — a distributed operating system for AI compute, from datacenter GPUs to edge handhelds, under one control plane.

LLMsVerifier — mandatory verification gate



LLMsVerifier — verify, monitor, optimize: the single source of truth for LLM/provider/verification metadata, gated by a mandatory model-comprehension check.