

VASIC DIGITAL · EVIDENCE-BASED PORTFOLIO



# Milos Vasic ✓

**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. 개요 — 전체 그림부터

이 포트폴리오는 vasic.digital와 milosvasic.ru가 공동으로 사용하는 단일 통합 포트폴리오입니다. 여기에는 산발적인 사이드 프로젝트가 아닌, 의도적으로 설계된 **시스템**이 담겨 있습니다. **대규모 제품 애플리케이션**은 수십 개의 작은 **독립 모듈** 위에 구축되며, 각 모듈은 독립적으로 테스트되고, 전체 시스템은 공유된 엔지니어링 **Constitution**에 의해 관리되며, **블러핑 방지 QA** 프로세스로 검증됩니다. 이러한 구조가 바로 차별점입니다. 대부분의 포트폴리오는 단순히 '만들어진 것들의 목록'에 불과하지만, 이 포트폴리오는 검증된 재사용 가능한 부품으로 구성된 시스템입니다. 모든 부품은 동일한 절대적인 규칙을 따르며, 모든 광고된 기능에는 그 증거가 담겨 있습니다.

주요 언어는 **Go**이며, Kotlin/KMP, TypeScript/React, Python, Swift, Shell는 각각의 적합성에 따라 사용됩니다. **Go**는 고처리량 서비스와 라이브러리에, Kotlin은 프로비저닝과 크로스플랫폼 모바일에, TypeScript는 타입 기반 프론트엔드에, Python은 AI/ML 간의 연결에 활용됩니다.

이 작업들이 하나로 묶이는 이유는 그 원칙이 추상적인 목표가 아니라 기계적인 규율에 기반하기 때문입니다. 공유된 **Constitution**는 Git 서브모듈로 배포되어 140개 이상의 저장소에 상속되며, 하나의 규칙 변경이 전체 시스템에 즉시 반영됩니다. 또한 블러핑 방지 QA 레이어는 런타임 증거 없이는 통과를 기록하지 않습니다. **Helix** 제품군, 유틸리티 시스템, **Server Factory** 도구 체인은 하나의 아이디어를 세 가지 방식으로 구현한 것입니다 — 한 번 만들고, 어디서나 재사용하며, 완료되었다고 선언하기 전에 실제로 작동한다는 증거를 확보합니다.

아래 내용은 우선순위 순으로 정리되어 있습니다.

1. **거버넌스 & QA 핵심 요소** — HelixConstitution, HelixQA (나머지 작업의 신뢰성을 보장하는 규율)
2. **Helix 제품군** — AI 개발 라이프사이클 (HelixTrack 우선)
3. **LLM 인프라** — 프로바이더 추상화, 오케스트레이션, 검증
4. **vasic-digital 유틸리티** — 제품급 독립 도구
5. **Server Factory** — 인프라 자동화의 유산 (우선순위 최하위)

통합된 핵심 명제: **실제 사용자가 사용할 수 있고, 그 증거가 확보되었을 때만 기능은 완료된 것이다.**

## 2. 거버넌스 & QA 핵심 요소

이 두 가지는 이 포트폴리오의 나머지 모든 것들이 신뢰성을 얻는 기반이 되기 때문에 가장 먼저 소개됩니다. 이 둘은 “믿어주세요, 잘 작동합니다”라는 말을 검증 가능한 사실로 바꿔줍니다. **Constitution**는 규칙을 codify하고, **HelixQA**는 그 규칙이 준수되었음을 증명합니다.

- **HelixConstitution** — 프로젝트와 무관한 보편적인 엔지니어링 규칙집으로, Git 서브모듈로 배포되어 140개 이상의 저장소에 상속됩니다. 블러핑 방지 증거 게이트, 거짓 양성 방지, 데이터/호스트 안전성, 커버리지 규율, 확장하되 절대 약화되지 않는 상속 구조, 전체 저장소에 걸쳐 필수 조항을 검색하는 전파 게이트, 모든 게이트에는 그것이 허울뿐이 아님을 증명하는 뮤테이션 테스트가 짝을 이룹니다.

- **HelixQA** — 블러핑 방지 QA 오케스트레이션 (Go). 작성된 **YAML** 테스트 बैं크와 안드로이드, 안드로이드 TV, 웹, 데스크톱을 아우르는 완전 자동화된 **LLM** 및 컴퓨터 비전 QA 세션으로 구성되며, 증거(스크린샷, 로그캣, 영상, 스택 트레이스)가 확보되지 않으면 PASS를 기록하지 않습니다. 이는 **Constitution**에서 의무화한 QA 테스트 유형(§ 11.4.169)에 해당합니다.

### 3. Helix 제품군

Helix 라인은 AI 개발 라이프사이클 전체를 아우르는 플래그십 제품군입니다. 기획과 스펙 정의부터 구축, 메모리, 번역, 배포에 이르기까지 모든 단계를 연결하는 제품들로 구성되어 있으며, 각각은 독립적인 제품으로서의 가치를 지니지만, 궁극적으로는 하나의 통합 시스템으로 설계되었습니다. 동일한 거버넌스, 재사용 가능한 모듈, 그리고 모든 제품에 적용되는 동일한 증거 기반 원칙이 그 바탕에 있습니다.

- **HelixTrack** — 자유로운 세계를 위한 JIRA 대체 제품. Helix-Track 라인의 플래그십.
- **HelixAgent** — 앙상블 LLM 서비스: 여러 모델이 토론을 거쳐 합의된 답변을 제공하며, 검증 기반 공급자 선택 시스템을 적용.
- **HelixCode** — 엔터프라이즈급 분산 AI 개발 플랫폼. SSH로 관리되는 워커에 작업을 분배하며, 자동 체크포인트/롤백 기능을 지원. REST, CLI, TUI, MCP 인터페이스 제공.
- **HelixCluster** — 데이터센터 GPU부터 엣지 핸드헬드 기기까지, 하나의 제어 플레인 하에서 동작하는 AI 컴퓨팅을 위한 분산 운영체제.
- **HelixBuilder** — AI 기반 애플리케이션 구축 파이프라인. 한 번에 하나의 카테고리씩 구축.
- **HelixSkills** — CLI AI 에이전트를 위한 거버넌스 및 헌법 기반 스킬 시스템(스킬, MCP 톨 서버, Claude Code 플러그인).
- **HelixSpecifier** — 작업 규모에 맞게 절차를 조정하는 스펙 기반 개발.
- **HelixMemory** — AI 에이전트를 위한 통합 메모리 브레인. 네 가지 최고 수준의 엔진을 융합.
- **HelixTranslate** — 검증된 모델 기반 도서 번역. 설계상 투명하며, 묵시적인 대체 수단은 사용하지 않음.
- **HelixTerminator** — 제로 트러스트 터미널 플랫폼: 모든 SSH 세션은 보안 강화, 공유 가능, AI 지원 기능 제공.
- **HelixGitpx** — 수십 개 호스트에 걸친 연합 Git 시스템. 하나의 진실 공급원, 모든 곳에 미러링.
- **HelixOTA** — 보편적이며 디커플링된 무선 업데이트 시스템. 설계상 무결성 보장.
- **HelixPlay** — 어떤 GPU 기기든 클라우드 게이밍 기기로 변환.
- **Helix-Flow** — Helix 플랫폼 추론 제품. 미검증 / 소스 코드 미비: 현재 공개 저장소에는 한 줄짜리 README만 존재. 실제 문서가 준비되는 시점에 제품 수준으로 공개 예정.

### 4. LLM 인프라

제품군 아래에는 이를 공급자 중립적이고 신뢰할 수 있게 만드는 기반 인프라가 자리합니다. 수십 개의 LLM 공급자를 아우르는 단일 인터페이스, 헤드리스 코딩 에이전트를 위한 단일 제어 플레인, 그리고 검증된 단일 진실 공급원이 그 핵심입니다. 이 계층은 상위 시스템이 모델을 교체하고, 공급자 장애를 극복하며, 작업을 제대로 이해하지 못하는 LLM를 신뢰하지 않도록 합니다.

- **HelixLLM** — 하나의 바이너리, 여섯 가지 모드: OpenAI 및 Anthropic 호환 추론(HTTP/3, 로컬 llama.cpp, 점수 기반 대체 체인, RAG 파이프라인, ReAct 에이전트 지원).
- **LLMProvider** — 하나의 인터페이스, 43개 공급자 지원. 서킷 브레이커, 재시도, 상태 모니터링 내장.

- **LLMOrchestrator** — 모든 헤드리스 CLI 코딩 에이전트(OpenCode, Claude Code, Gemini, Junie, Qwen Code)를 위한 단일 제어 플레인.
- **LLMsVerifier** — 검증, 모니터링, 최적화: LLM/공급자/검증 메타데이터를 위한 단일 진실 공급원. 모델 이해도 검증 필수.

## 5. 바식-디지털 유틸리티

각각이 독립적으로 어려운 문제를 해결하는 제품급 도구들로, 우연히도 재사용 가능한 모듈군을 실전 환경에서 검증하는 역할을 합니다. 복원력 있는 멀티 프로토콜 미디어 시스템부터 마크다운-비디오 강의 파이프라인, 콘텐츠 해시 기반 문서/DB 동기화 엔진까지 다양하며, 일부 도구는 성숙도를 솔직하게 밝히고 있으며 이러한 표시들은 숨기지 않고 그대로 유지합니다.

- **Catalogizer** — 멀티 프로토콜(SMB/FTP/NFS/WebDAV/로컬) 지원, 암호화, 자가 호스팅 가능한 미디어 컬렉션 관리 시스템; Go/Gin API + React UI; 오프라인 내성 모니터링; [digital.vasic.\\*](#) 하위 모듈 21개 기반 구축.
- **Courses-Creator** — 마크다운-비디오 강의 파이프라인; 멀티 LLM 강화, TTS(Bark/SpeechT5), 데스크톱/모바일/웹 플레이어 지원; API 키 미지원 모드 제공.
- **VisionEngine** — 고전적 컴퓨터 비전과 멀티 공급자 LLM 비전을 결합한 분리형 Go 툴킷; BFS 기반 내비게이션 그래프 및 DOT/JSON/Mermaid 내보내기 지원; OpenCV 빌드 태그 게이트 적용.
- **DocProcessor** — 문서-기능 맵핑 및 검증 커버리지 추적; LLM 또는 휴리스틱/오프라인 추출 지원; Apache-2.0 라이선스.
- **Docs Chain** — 콘텐츠 해시 기반 양방향 원자적 문서/DB 동기화(Salsa 스타일 DAG 증분 재계산). 1~5단계 GREEN; 6~7단계 PLANNED.
- **Herald** — 신뢰성 높은 멀티 채널 알림 시스템; 자연어 기반 3단계 의도 해석(명령 → LLM → 명확화); Docs Chain의 첫 번째 소비자.
- **task\_bridge** — 분리형 양방향 작업/보드 동기화(SQLite SSoT [?](#) 문서 [?](#) ClickUp). P1 스캐폴드 구축 완료 — 동기화 로직 미구현.
- **Vasic Digital 재사용 모듈 스위트** — [digital.vasic.\\*](#) “표준 라이브러리”: 인프라 기본 요소, AI 빌딩 블록, 방어적 LLM 가드레일, 그리고 Kotlin Multiplatform 미러 포함. 일부 조직 저장소는 SCAFFOLD/WIP 상태 — 표시된 대로 제공, 배포되지 않음.

## 6. Server Factory (인프라 자동화 유산 — 의도적으로 최하위 순위)

품질이 아닌 설계에 따라 최하위에 배치된 도구군입니다. Server Factory 툴체인은 AI 라인보다 앞서 개발되었으며, “한 번 만들고 어디서나 재사용한다”는 철학이 처음 구현된 모습을 보여줍니다. 대표 제품인 JSON로 정의하고 어디서든 프로비저닝할 수 있는 메일 서버는 성숙하고 검증된 제품이며, 지원 도구들은 과장 없이 실제 성숙도에 맞춰 소개됩니다.

- **Mail Server Factory** — 선언형 JSON → 12개 연결 유형 및 25개 리눅스 배포판 지원, 완전 프로비저닝된 도커 기반 메일 서버; 엔터프라이즈 보안; 439개 테스트 통과 및 SonarQube 게이트 클린 상태 보고. Server-Factory 조직의 대표 제품.
- **Server Factory 코어 프레임워크** — 모든 팩토리가 공유하는 Kotlin 엔진.
- **Qemu-Utils** — 아티팩트처럼 관리되는 QEMU VM 이미지: 다운로드/캐시/실행, 압축/게시, 브리지/TAP 네트워킹, ISO 설치 지원; Linux 및 macOS.
- **Parallels-Utils** — Parallels(macOS) VM 이미지 압축, 게시 및 간단한 설정 파일로 검색.

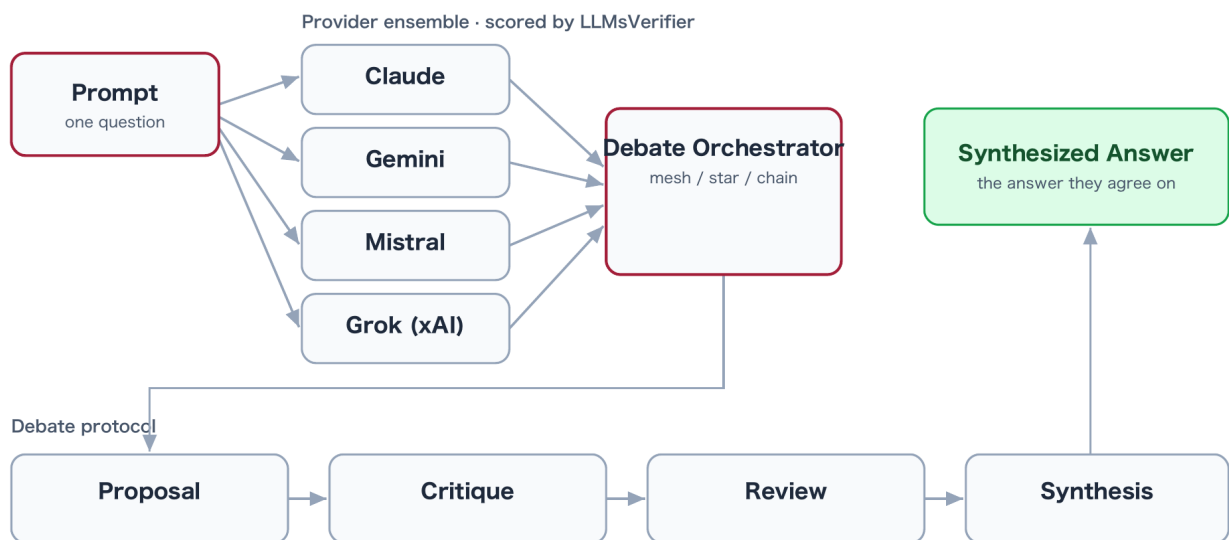
- **Server Factory — 추가 구성 요소** — 서비스 팩토리(Web/SonarQube/캐싱 프록시), 정의 팩, 유틸리티 포함. 서비스 팩토리는 플레이스홀더 문서화 상태 — 검증되지 않음 / 초기 단계.

## 7. 기술 지수 (증거 기반)

- **언어:** Go(주류), Kotlin & Kotlin Multiplatform, TypeScript, Python, Swift, Shell; PL/pgSQL; TLA+ (helix\_cluster 내 공식 사양)
- **AI / LLM:** 40개 이상의 제공업체 접근, MCP, RAG, vector DB 및 embeddings, 기획(HiPlan/MCTS/Tree-of-Thoughts), LLMops, 벤치마킹(SWE-bench/HumanEval/MMLU), TTS(Bark/SpeechT5), 컴퓨터 비전 + LLM 비전, 안전장치/레드팀
- **백엔드:** Gin, gRPC + Protobuf, HTTP/3(QUIC), WebSocket, Angular, React, Kafka/RabbitMQ
- **데이터:** PostgreSQL, SQLite, SQLCipher, Redis, Neo4j, ClickHouse, MinIO/S3/GCS/Azure
- **인프라 / DevOps:** Docker & Compose, Kubernetes + Helm, Prometheus + Grafana, OpenTelemetry, QEMU/Libvirt/Parallels; GitHub Actions, Gradle, Make
- **테스트 / QA:** HelixQA, 변이 게이트가 포함된 챌린지 하네스, `go test -race`, 시각적 회귀 테스트 도구, ADB 기기 테스트, SonarQube, 보안 스캐닝(semgrep/gosec/trivy/snyk/gitleaks/nancy), TLA+ 모델 검증

## 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)**

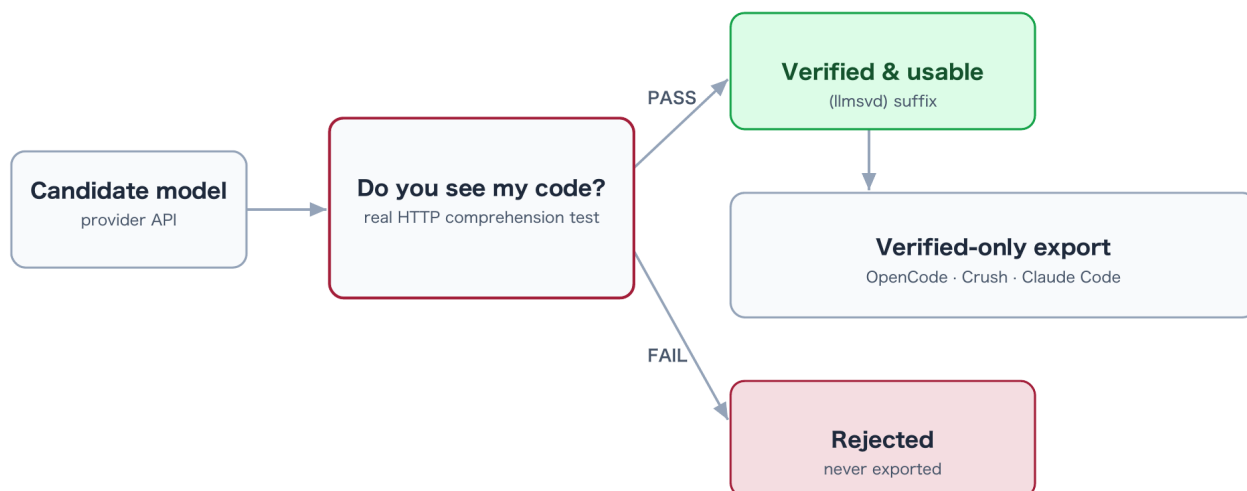
heterogeneous  
nodes T1–T8

**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.