



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. Pregled — slika u celini

Ovo je jedinstven, ujedinjen portfolio koji koriste i vasic.digital i milosvasic.ru. Ne opisuje razbacane sporedne projekte, već namerno projektovanu **flotu**: **velike aplikacije** koje se oslanjaju na **desetine malih, odvojenih, nezavisno testiranih modula**, a sve je podređeno zajedničkoj inženjerskoj **Constitution** i provereno **anti-bluf QA** disciplinom. Ta struktura je ono što ga izdvaja. Većina portfolia je samo spisak onoga što je napravljeno; ovo je sistem u kojem se svaki proizvod sastavlja od dokazanih, višekratno upotrebljivih delova, svaki deo poštuje iste nepregovorive pravila, a iza svake reklamirane mogućnosti stoji prikupljen dokaz. Dominantan jezik je **Go**, uz Kotlin/KMP, TypeScript/React, Python, Swift i Shell tamo gde svaki zaista najbolje pristaje — Go za usluge i biblioteke visokog protoka, Kotlin za provizioniranje i multiplatformske mobilne aplikacije, TypeScript za tipizovane front-endove, Python za povezivanje AI/ML.

Ono što ovom radu daje koherentnost jeste činjenica da je disciplina mehanička, a ne samo željena. Zajednička **Constitution** isporučuje se kao Git podmodul i nasleđuje se u floti od preko 140 repozitorijuma, tako da svaka promena pravila automatski stiže svuda; anti-bluf QA sloj odbija da zabeleži prolaz bez dokaza u vreme izvršavanja. Porodica **Helix**, flota pomoćnih alata i **Server Factory** alati su tri izraza jedne ideje — napravi jednom, koristi svuda i dokaži da radi pre nego što to proglasiš gotovim.

Sve što sledi prikazano je po prioritetu:

1. **Piloni upravljanja i QA** — HelixConstitution, HelixQA (disciplina koja sve ostalo čini pouzdanim).
2. **Porodica proizvoda Helix** — životni ciklus razvoja AI (HelixTrack na prvom mestu).
3. **LLM infrastruktura** — apstrakcija provajdera, orkestracija, verifikacija.
4. **Vasic-digital pomoćni alati** — samostalni alati na nivou proizvoda.
5. **Server Factory** — nasleđe automatizacije infrastrukture (rangirano poslednje).

Ujedinjujuća teza: **funkcionalnost je gotova tek kada je stvarni korisnik može da koristi i kada postoji prikupljen dokaz koji to potvrđuje.**

2. Piloni upravljanja i QA

Ova dva segmenta dolaze na prvo mesto jer sve ostalo u ovom portfoliju pozajmljuje svoju kredibilnost od njih. Zajedno pretvaraju „verujte mi, radi” u proverljivu činjenicu — **Constitution** kodira pravila, a **HelixQA** dokazuje da su poštovana.

- **HelixConstitution** — univerzalni, projektno-agnostički inženjerski pravilnik isporučivan kao Git podmodul i nasleđen u floti od preko 140 repozitorijuma. Anti-bluf kontrolne tačke, imunitet na lažno pozitivne rezultate, sigurnost podataka i hostova, disciplina pokrivenosti; nasleđivanje s mogućnošću proširenja, ali bez slabljenja; kontrolni mehanizmi koji pretražuju sve repozitorijume u floti tražeći obavezne odredbe; svaka kontrolna tačka uparena s mutacionim testom koji dokazuje da nije lažna.

- **HelixQA** — anti-bluf orkestracija QA (Go). Pisani **YAML** testovi plus potpuno autonomne QA sesije na Androidu, Android TV-u, vebu i desktopu, uz korišćenje **LLM** i računarskog vida; nema PROLAZA bez prikupljenih dokaza (snimci ekrana, logcat, video, stack trace-ovi). Obavezan tip QA testa prema **Constitution** (§11.4.169).

3. Porodica proizvoda Helix

Linija Helix predstavlja zastavni deo: porodicu međusobno povezanih proizvoda koja obuhvata ceo ciklus razvoja AI, od planiranja i specifikacije, preko izgradnje, memorije, prevođenja, do isporuke. Svaki od njih je sam po sebi pravi proizvod, ali je osnovna zamisao da se međusobno dopunjuju – sa istom upravljačkom strukturom, istim modulima za ponovnu upotrebu i istom disciplinom dokazivanja koja stoji iza svih njih.

- **HelixTrack** – alternativa JIRA za slobodan svet; zastavni proizvod linije Helix-Track.
- **HelixAgent** – ansambl usluga LLM: više modela raspravlja i isporučuje odgovor na koji se slože, uz izbor provajdera zasnovan na verifikaciji.
- **HelixCode** – preduzetnička platforma za distribuirani razvoj AI; deli posao na radnike kojima upravlja SSH, sa automatskim checkpointima i povratkom na prethodno stanje; interfejsi REST, CLI, TUI, MCP.
- **HelixCluster** – distribuirani operativni sistem za računanje zasnovano na AI, od GPU-ova u podatkovnim centrima do ručnih uređaja na ivici mreže, pod jednim kontrolnim ravnima.
- **HelixBuilder** – pipeline pokretan AI za izgradnju aplikacija, kategoriju po kategoriju.
- **HelixSkills** – sistem veština za agente CLI zasnovane na AI, regulisan ustavom (veštine, serverski alati MCP, dodaci Claude Code).
- **HelixSpecifier** – razvoj vođen specifikacijama koji prilagođava ceremonijal obimu posla.
- **HelixMemory** – jedan memorijski mozak za agente AI, koji spaja četiri vrhunska motora.
- **HelixTranslate** – prevođenje knjiga zasnovano na verifikovanim modelima; po dizajnu iskreno, nikada sa tihim povratkom na rezervni režim.
- **HelixTerminator** – terminalska platforma sa nultim poverenjem: svaka sesija SSH osigurana, deljena i asistirana od strane AI.
- **HelixGitpx** – federisani Git na desetak hostova; jedan izvor istine, ogledalo svuda.
- **HelixOTA** – univerzalna, odvojena ažuriranja „over-the-air”; dizajnirana da nikada ne blokiraju uređaj.
- **HelixPlay** – pretvorite bilo koju mašinu GPU u sopstveni uređaj za cloud-igranje.
- **Helix-Flow** – proizvod za inferenciju na platformi Helix. *NEVERIFIKOVANO / blokirano zbog izvora: javni repozitorijum trenutno sadrži samo jednolinijski README; prikazano na nivou proizvoda tek kada postoji prava dokumentacija.*

4. Infrastruktura LLM

Ispod proizvoda nalazi se podloga koja ih čini nezavisnim od provajdera i pouzdanim: jedan interfejs preko desetina provajdera LLM, jedna kontrolna ravan za agente za kodiranje bez grafičkog interfejsa i jedan izvor

istine za verifikaciju. Ovo je sloj koji omogućava svemu iznad da menja modele, preživi prekide u radu provajdera i odbije da veruje LLM koji ne može da dokaže da razume zadatak.

- **HelixLLM** – jedan binarni fajl, šest režima: inferencija kompatibilna sa OpenAI i Anthropic preko HTTP/3, lokalni llama.cpp, lanac ocene rezervnih opcija, pipeline RAG, ReAct agenti.
- **LLMProvider** – jedan interfejs, 43 provajdera, sa ugrađenim prekidačima strujnog kola, ponovnim pokušajima i monitoringom zdravlja.
- **LLMOrchestrator** – jedna kontrolna ravan za svakog agenta za kodiranje bez grafičkog interfejsa (OpenCode, Claude Code, Gemini, Junie, Qwen Code).
- **LLMsVerifier** – verifikujte, nadgledajte, optimizujte: jedini izvor istine za metapodatke o LLM/provajderima/verifikaciji, sa obaveznom proverom razumevanja modela.

5. vasic-digital alati

Samostalni alati na nivou gotovog proizvoda, od kojih svaki rešava složen problem na svoj način — a ne slučajno, istovremeno testirajući modularnu flotu u praksi. Raspon im je od otpornog multimedijalnog sistema sa više protokola do toka obrade kursa od markdown-a do videa, pa sve do motora za sinhronizaciju dokumentacije i baza podataka sa heširanim sadržajem; nekoliko njih otvoreno govori o svojoj zrelosti, a te oznake ostaju vidljive, a ne skrivene.

- **Catalogizer** — multimedijalni sistem za upravljanje kolekcijama sa više protokola (SMB/FTP/NFS/WebDAV/lokalno), šifrovan, samostalno hostovan; Go/Gin API + React interfejs; otporno praćenje sa podrškom za offline režim; izgrađen na 21 `digital.vasic.*` potmodulu.
- **Courses-Creator** — tok obrade kursa od markdown-a do videa; obogaćivanje sa više LLM, TTS (Bark/SpeechT5), plejeri za desktop/mobilni/web; elegantan režim rada bez ključa API.
- **VisionEngine** — nezavisni Go alat koji spaja klasični računarski vid sa LLM viđenjem više dobavljača; navigacioni grafovi sa BFS + izvozom u DOT/JSON/Mermaid; OpenCV sa uslovnim oznakama za izgradnju.
- **DocProcessor** — mapa funkcionalnosti iz dokumentacije sa praćenjem pokrivenosti verifikacijom; LLM ili heurističko/offline izdvajanje; licenca Apache-2.0.
- **Docs Chain** — dvosmerna, atomska sinhronizacija dokumentacije i baza podataka sa heširanim sadržajem (inkrementalno preračunavanje u stilu Salse preko DAG-a). *Faze 1–5 ZELEN; Faze 6–7 U PLANU.*
- **Herald** — pouzdane obaveštenja preko više kanala sa razrešavanjem namere u tri nivoa (komanda → LLM → pojašnjenje) na prirodnom jeziku; prvi korisnik Docs Chain.
- **task_bridge** — nezavisna, dvosmerna sinhronizacija zadataka i tabela (SQLite jedini izvor istine [?] dokumentacija [?] ClickUp). *Skelet P1 — logika sinhronizacije još nije implementirana.*
- **Vasic Digital Paket ponovljivo upotrebljivih modula** — „standardna biblioteka” `digital.vasic.*`: infrastrukturni primitivi, gradivni blokovi AI i odbrambeni mehanizmi LLM, uz ogledalo Kotlin Multiplatform. *Nekoliko repozitorijuma organizacije je U SKELETU/RAZVOJU — označeno, a ne objavljeno.*

6. Server Factory (nasleđe automatizacije infrastrukture — poslednje rangiran)

Rangiran poslednji po zamisli, a ne po kvalitetu: Server Factory alatni lanac prethodi liniji AI i pokazuje gde je filozofija „izgradi jednom, koristi svuda” prvi put zaživela. Njegov vodeći proizvod — mail server koji opišete u JSON i postavite bilo gde — zreo je, dobro testiran proizvod; prateći alati prikazani su u svojoj stvarnoj, različitoj zrelosti, a ne uljepšani.

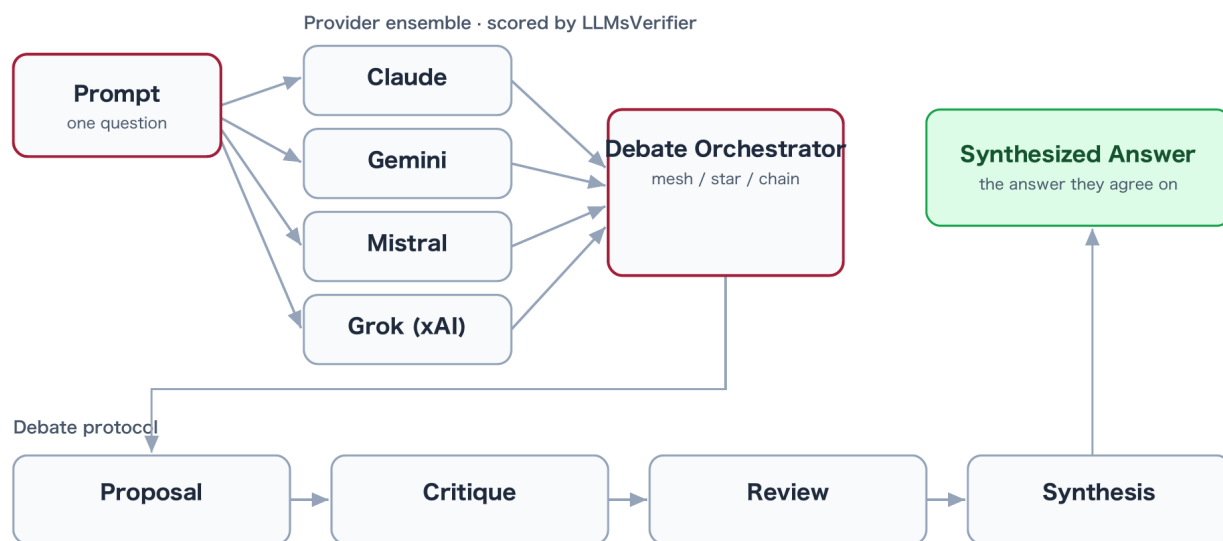
- **Mail Server Factory** — deklarativni JSON → potpuno postavljen, dokeraizovan mail server za 12 vrsta veza i 25 Linux distribucija; preduzetnička sigurnost; izveštava o 439 uspešno položenih testova i čistoj SonarQube kapiji. Vodeći proizvod Server-Factory organizacije.
- **Server Factory Osnovni okvir** — zajednički Kotlin pogon na kojem se grade sve fabrike.
- **Qemu-Utills** — VM slike QEMU upravljane kao artefakti: preuzimanje/keširanje/pokretanje, kompresija/objavljivanje, mrežno povezivanje preko bridge/TAP, ISO instalacije; Linux + macOS.
- **Parallels-Utills** — kompresija, objavljivanje i preuzimanje VM slika Parallels (macOS) putem jednostavnih fajlova sa podešavanjima.
- **Server Factory — Dodatne komponente** — fabrike servisa (Web/SonarQube/Caching-Proxy), paketi definicija i pomoćni alati. *Fabrike servisa su dokumentovane kao placeholderi — NEVERIFIKOVANO / u ranoj fazi.*

7. Tehnološki indeks (baziran na dokazima)

- **Jezici:** Go (dominantan), Kotlin i Kotlin Multiplatform, TypeScript, Python, Swift, Shell; PL/pgSQL; TLA+ (formalne specifikacije u *helix_cluster*).
- **AI / LLM:** pristup preko 40 provajdera, MCP, RAG, baze podataka vector i embeddings, planiranje (HiPlan/MCTS/*Tree-of-Thoughts*), LLMOps, benchmarking (SWE-bench/HumanEval/MMLU), TTS (Bark/SpeechT5), računarski vid + LLM-vizija, zaštitne mere/red-tim.
- **Backend:** Gin, gRPC + Protobuf, HTTP/3 (QUIC), WebSockets, Angular, React, Kafka/RabbitMQ.
- **Podaci:** PostgreSQL, SQLite, SQLCipher, Redis, Neo4j, ClickHouse, MinIO/S3/GCS/Azure.
- **Infra / DevOps:** Docker i Compose, Kubernetes + Helm, Prometheus + Grafana, OpenTelemetry, QEMU/Libvirt/Parallels; GitHub Actions, Gradle, Make.
- **Testiranje / KA:** HelixQA, okviri za izazove sa mutacionim kapijama, `go test -race`, alati za vizuelnu regresiju, testiranje na ADB uređajima, SonarQube, skeniranje sigurnosti (semgrep/gosec/trivy/snyk/gitleaks/nancy), provera modela 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)

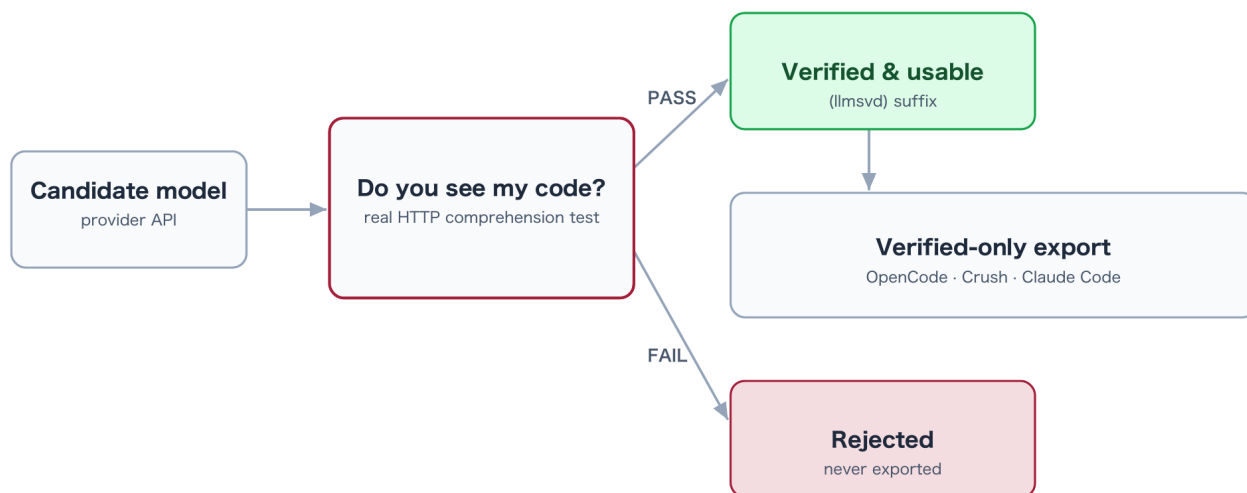
heterogeneous
nodes T1–T8

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.