

VASIC DIGITAL · EVIDENCE-BASED PORTFOLIO

Miloš

Portfolio · Helix family · vasic-d



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

439	25	43	+140
PASSING TESTS · MAIL SERVER FACTORY	LINUX DISTROS PROVISIONED	LLM PROVIDERS	REPOSITORY FLEET

۱. نمای کلی — تصویر کامل در ابتدا

این یک پورتفولیوی واحد و یکپارچه است که توسط هر دو **vasic.digital** و **milosvasic.ru** مورد استفاده قرار می‌گیرد. این پورتفولیو نه مجموعه‌ای پراکنده از پروژه‌های جانبی، بلکه یک **ناوگان** مهندسی‌شده و هدفمند را توصیف می‌کند: **برنامه‌های کاربردی محصولی بزرگ** که بر روی **ده‌ها مازول کوچک، جداشده و مستقلاً آزمایش‌شده** قرار گرفته‌اند؛ کل این مجموعه توسط یک **چارچوب مهندسی مشترک (Constitution)** اداره می‌شود و توسط یک **فرایند تضمین کیفیت ضد-فریب** تأیید می‌گردد. این ساختار وجه تمایز اصلی است. بیشتر پورتفولیوها فهرستی از چیزهایی هستند که ساخته شده‌اند؛ اما این یک سیستم است که در آن هر محصول از قطعات اثبات‌شده و قابل استفاده مجدد مونتاژ می‌شود، هر قطعه تابع قوانین غیرقابل مذاکره یکسانی است، و پشت هر قابلیت تبلیغ‌شده شواهد ملموس وجود دارد. زبان غالب **Go** است، با استفاده از **Kotlin/TypeScript/React**، **Python**، **Swift**، و **Shell** در مواردی که هر یک واقعاً بهترین گزینه هستند — **Go** برای سرویس‌ها و کتابخانه‌های با توان عملیاتی بالا، **Kotlin** برای تأمین منابع و توسعه موبایل چندسکویی، **TypeScript** برای فرانت‌اندهای تایپ‌شده، و **Python** برای اتصال **AI/ML**.

آنچه این مجموعه کارها را منسجم می‌کند، نظم و انضباط مکانیکی آن است، نه آرمانی. یک **چارچوب مشترک (Constitution)** به‌عنوان زیرماژول گیت منتشر می‌شود و در سراسر ناوگانی با بیش از ۱۴۰ مخزن به ارث می‌رسد، به‌طوری که تغییر یک قانون به‌طور خودکار در همه‌جا اعمال می‌شود؛ لایه تضمین کیفیت ضد-فریب بدون شواهد اجرایی از ثبت تأیید خودداری می‌کند. خانواده **Helix**، ناوگان ابزارهای کاربردی، و زنجیره ابزار **Server Factory** سه جلوه از یک ایده واحد هستند — یک بار بساز، همه‌جا استفاده کن، و پیش از اعلام اتمام کار، اثبات کن که کار می‌کند.

موارد زیر به ترتیب اولویت ارائه شده‌اند:

۱. **ستون‌های حاکمیت و تضمین کیفیت** — **HelixQA**، **HelixConstitution** (انضباطی که بقیه را قابل اعتماد می‌سازد).
۲. **خانواده محصولی Helix** — چرخه حیات توسعه **AI** (ابتدا **HelixTrack**). ۳. **زیرساخت LLM** — انتزاع تأمین‌کننده، هماهنگ‌سازی، تأیید. ۴. **ابزارهای مستقل واسیک-دیجیتال** — ابزارهای مستقل در سطح محصول. ۵. **Server Factory** — میراث خودکارسازی زیرساخت (آخرین اولویت).

تذویر: **یک قابلیت تنها زمانی تمام‌شده تلقی می‌شود که یک کاربر واقعی بتواند از آن استفاده کند و شواهد ملموسی برای اثبات آن وجود داشته باشد.**

۲. ستون‌های حاکمیت و تضمین کیفیت

این دو بخش در اولویت قرار دارند زیرا اعتبار تمام بخش‌های دیگر این پورتفولیو از آن‌ها وام گرفته شده است. این دو با هم عبارت «به من اعتماد کن، کار می‌کند» را به یک واقعیت قابل حسابرسی تبدیل می‌کنند — **Constitution** قوانین را کدگذاری می‌کند و **HelixQA** اثبات می‌کند که این قوانین رعایت شده‌اند.

• **HelixConstitution** — یک کتابچه قوانین مهندسی جهانی و مستقل از پروژه که به‌عنوان زیرماژول گیت منتشر می‌شود و در سراسر ناوگانی با بیش از ۱۴۰ مخزن به ارث می‌رسد. دروازه‌های شواهد ضد-فریب، مصونیت در برابر مثبت‌های کاذب، ایمنی داده‌ها/میزبان، انضباط پوشش؛ ارث‌بری با امکان گسترش اما بدون تضعیف؛ دروازه‌های انتشار که به‌طور خودکار در کل ناوگان به دنبال بندهای الزامی می‌گردند؛ هر دروازه با یک تست جهش همراه است که اثبات می‌کند جعلی نیست.

• **HelixQA** — هماهنگ‌سازی تضمین کیفیت ضد-فریب (Go). بانک‌های تست **YAML** به‌صورت مکتوب به‌علاوه جلسات تضمین کیفیت کاملاً خودکار با استفاده از **LLM** و بینایی کامپیوتر در پلتفرم‌های اندروید، اندروید تی‌وی، وب و دسکتاپ؛ بدون تأیید بدون شواهد ملموس (تصاویر، لاگ‌ت، ویدئو، ردپای پشته). نوع تست تضمین کیفیت الزامی چارچوب (**Constitution**)، بند ۱۱.۴.۱۶۹.

بخش

۳. خانواده محصولات Helix

خط تولید Helix پرچمدار ماست: مجموعه‌ای درهم‌تنیده از محصولات که کل چرخه توسعه AI را از برنامه‌ریزی و تعیین مشخصات گرفته تا ساخت، حافظه، ترجمه و تحویل در بر می‌گیرد. هر یک از این محصولات به خودی خود یک محصول واقعی است، اما هدف طراحی این بوده که در کنار هم کار کنند — با یک نظام حاکمیتی واحد، ماژول‌های قابل استفاده مجدد یکسان، و یک نظم مستندسازی مشترک در زیرساخت همه آنها.

- **HelixTrack** — جایگزین JIRA برای دنیای آزاد؛ پرچمدار خط تولید Helix-Track.
- **HelixAgent** — سرویس LLM گروهی: چندین مدل با هم بحث می‌کنند و پاسخی را که بر سر آن توافق دارند ارسال می‌کنند، با انتخاب ارائه‌دهنده بر اساس تأیید.
- **HelixCode** — پلتفرم توسعه توزیع‌شده AI در سطح سازمانی؛ کار را بین کارگران مدیریت‌شده توسط SSH تقسیم می‌کند با قابلیت خودکار چک‌پوینت/بازگشت؛ رابط‌های TUI، CLI، REST، MCP.
- **HelixCluster** — سیستم عامل توزیع‌شده برای محاسبات AI، از پردازنده‌های گرافیکی دیتاسنتر تا دستگاه‌های لبه‌ای، تحت یک صفحه کنترل واحد.
- **HelixBuilder** — خط لوله مبتنی بر AI برای ساخت برنامه‌ها، یک دسته‌بندی در هر بار.
- **HelixSkills** — سیستم مهارت‌های حاکمیتی و مبتنی بر قانون اساسی برای عوامل CLI و AI (مهارت‌ها، سرورهای ابزار MCP، افزونه‌های Claude).
- **HelixSpecifier** — توسعه مبتنی بر مشخصات که تشریفات آن با حجم کار مقیاس می‌یابد.
- **HelixMemory** — یک حافظه واحد برای عوامل AI که چهار موتور برتر را در هم ادغام می‌کند.
- **HelixTranslate** — ترجمه کتاب با مدل تأییدشده؛ صادقانه از طراحی، هرگز با بازگشت خاموش.
- **HelixTerminator** — پلتفرم ترمینال با اعتماد صفر: هر جلسه SSH امن، اشتراک‌گذاری‌شده و با کمک AI.
- **HelixGitpx** — گیت فدرال در چندین میزبان؛ یک منبع حقیقت، منعکس‌شده در همه جا.
- **HelixOTA** — به‌روزرسانی‌های جهانی و جداشده از طریق هوا؛ طراحی‌شده برای عدم ایجاد اختلال.
- **HelixPlay** — تبدیل هر دستگاه GPU به یک ابزار ابری بازی خودتان.
- **Helix-Flow** — محصول استنتاج پلتفرم Helix. تأییدنشده / مسدودشده به دلیل منبع: مخزن عمومی آن در حال حاضر تنها یک فایل README یک‌خطی دارد؛ تنها زمانی با جزئیات محصول ارائه می‌شود که مستندات واقعی وجود داشته باشد.

۴. زیرساخت LLM

در زیر محصولات، بستری قرار دارد که آنها را مستقل از ارائه‌دهنده و قابل اعتماد می‌سازد: یک رابط واحد بر روی دهه ارائه‌دهنده LLM، یک صفحه کنترل برای عوامل کدنویسی بدون رابط کاربری، و یک منبع حقیقت تأیید. این لایه است که به همه چیزهای بالاتر اجازه می‌دهد مدل‌ها را تعویض کنند، از قطعی ارائه‌دهندگان جان سالم به در ببرند و از پذیرش LLM ای که نتواند ثابت کند وظیفه را درک کرده است، خودداری کنند.

- **HelixLLM** — یک باینری، شش حالت: استنتاج سازگار با OpenAI و Anthropic بر روی llama.cpp، HTTP/3، محلی، زنجیره بازگشت امتیازدهی‌شده، خط لوله RAG، عوامل ReAct.

- **LLMProvider** — یک رابط، ۴۳ ارائه‌دهنده، با قطع‌کننده‌های مدار، تلاش‌های مجدد و سلامت داخلی.
- **LLMOrchestrator** — یک صفحه کنترل برای هر عامل کدنویسی بدون رابط کاربری (OpenCode، Claude CLI (Code، Gemini، Junie، Qwen Code).
- **LLMsVerifier** — تأیید، نظارت، بهینه‌سازی: منبع واحد حقیقت برای ابر داده‌های LLM/ارائه‌دهنده/تأیید، با دروازه اجباری درک مدل.

محتوا

۵. ابزارهای واسیک-دیجیتال

ابزارهای مستقل و آماده‌ی محصول که هر یک به تنهایی مسئله‌ای دشوار را حل می‌کنند — و البته، به طور غیرمستقیم، مجموعه‌ی ماژول‌های قابل‌استفاده را در شرایط واقعی به کار می‌گیرند. این ابزارها از یک سامانه‌ی چندپروتکله‌ی مقاوم برای رسانه‌ها تا خط لوله تبدیل مارک‌داون به ویدیو و موتوری برای همگام‌سازی سند/پایگاه‌داده با هش محتوا را شامل می‌شوند؛ برخی صریحاً به سطح بلوغ خود اشاره می‌کنند و این نشان‌ها پنهان نمی‌شوند.

- **Catalogizer** — مدیریت مجموعه رسانه‌ای چندپروتکله (SMB/FTP/NFS/WebDAV/محلی)، رمزنگاری شده و خودمیزبان؛ رابط کاربری React + Gin API؛ نظارت مقاوم در برابر قطعی آفلاین؛ بر اساس ۲۱ زیرماژول `*.digital.vasic`.

- **Courses-Creator** — خط لوله تبدیل مارک‌داون به ویدیو؛ غنی‌سازی چند LLM، TTS (Bark/SpeechT5)، پخش‌کننده‌های دسکتاپ/موبایل/وب؛ حالت بدون کلید API با عملکرد روان.

- **VisionEngine** — جعبه‌ابزار Go مستقل که بینایی کلاسیک کامپیوتر را با بینایی چند LLM ادغام می‌کند؛ گراف‌های ناوبری با جستجوی اول سطح (BFS) و خروجی DOT/JSON/Mermaid؛ وابسته به برچسب ساخت OpenCV.

- **DocProcessor** — نگاشت سند به قابلیت با ردیابی پوشش تأیید؛ استخراج LLM یا اکتشافی/آفلاین؛ تحت مجوز Apache-2.0.

- **Docs Chain** — همگام‌سازی سند و پایگاه‌داده‌ی دوطرفه و اتمی با هش محتوا (بازمحاسبه‌ی افزایشی به سبک Salsa بر روی گراف جهت‌دار غیرمدور). فازهای ۱ تا ۵ تأیید شده؛ فازهای ۶ و ۷ در برنامه‌ریزی.

- **Herald** — اعلان‌های چندکاناله‌ی قابل‌اعتماد با تفسیر سه‌سطحی قصد به زبان طبیعی (دستور → LLM → شفاف‌سازی)؛ اولین مصرف‌کننده‌ی Docs Chain.

- **task_bridge** — همگام‌سازی دوطرفه و مستقل وظایف/تخته (SQLite منبع حقیقت واحد [?] اسناد [?] ClickUp). چارچوب فاز ۱ — منطق همگام‌سازی هنوز پیاده‌سازی نشده است.

- **مجموعه‌ی ماژول‌های قابل‌استفاده‌ی Vasic Digital** — «کتابخانه‌ی استاندارد» `*.digital.vasic`: زیرساخت‌های پایه، بلوک‌های سازنده‌ی AI و محافظ‌های دفاعی LLM، به‌علاوه‌ی یک آینه‌ی Kotlin Multiplatform. چندین مخزن سازمانی در مرحله‌ی چارچوب/در حال توسعه هستند — مشخص شده، اما منتشر نشده.

۶. Server Factory (میراث زیرساخت-اتوماسیون — در رتبه‌ی آخر)

به عمد در رتبه‌ی آخر قرار گرفته، نه به دلیل کیفیت: زنجیره‌ی ابزار Server Factory پیش از خط AI شکل گرفته و نشان می‌دهد که فلسفه‌ی «یک بار بساز، همه‌جا استفاده کن» ابتدا چگونه متولد شد. محصول شاخص آن — یک سرور ایمیل که در JSON توصیف می‌شود و در هر جایی قابل استقرار است — محصولی بالغ و کاملاً آزمایش‌شده است؛ سایر اجزای همراه نیز با سطح بلوغ واقعی و متنوع خود ارائه می‌شوند، نه با ظاهری آراسته.

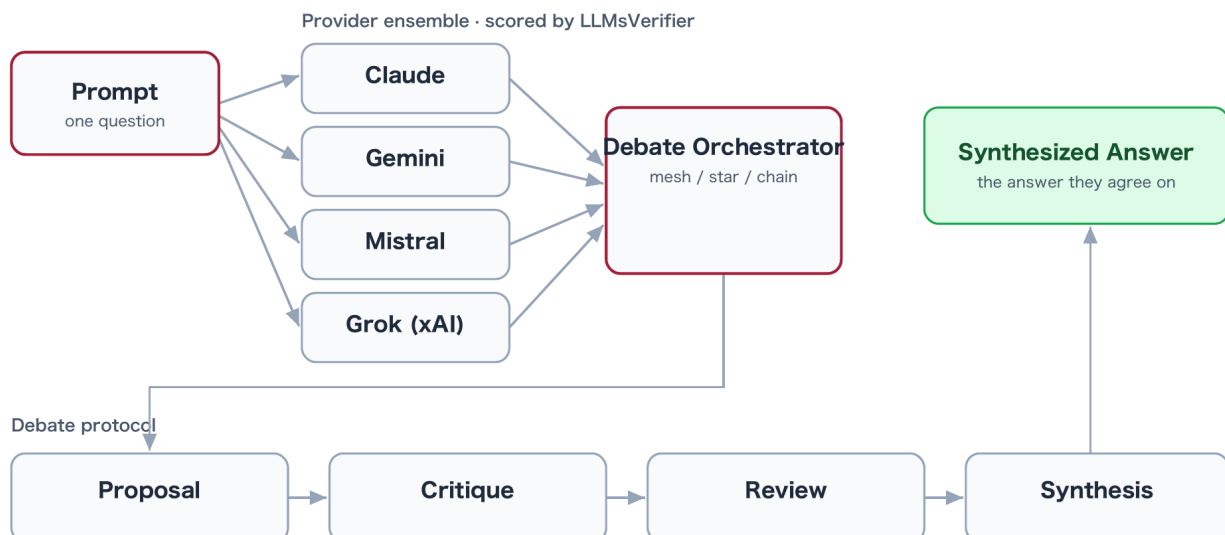
- **Mail Server Factory** — سرور ایمیل داکرایزشده و کاملاً مستقر با توصیف JSON در ۱۲ نوع اتصال و ۲۵ توزیع لینوکس؛ امنیت سازمانی؛ گزارش ۴۳۹ تست موفق و دروازه‌ی SonarQube پاک. محصول شاخص سازمان Server-Factory.

- **چارچوب اصلی Server Factory** — موتور Kotlin مشترک که همه‌ی کارخانه‌ها بر اساس آن ساخته می‌شوند.

- **Qemu-Utils** — تصاویر ماشین مجازی QEMU که مانند مصنوعات مدیریت می‌شوند: دانلود/ذخیره‌سازی/اجرا، فشردن/انتشار، شبکه‌ی پل/TAP، نصب از طریق ISO؛ لینوکس + macOS.
- **Parallels-Utils** — فشردن/انتشار، بازبازی تصاویر ماشین مجازی (macOS) Parallels از طریق فایل‌های تنظیمات ساده.
- **Server Factory — اجزای اضافی** — کارخانه‌های سرویس (وب/SonarQube/پروکسی کش)، بسته‌های تعاریف و ابزارها. کارخانه‌های سرویس مستندسازی اولیه دارند — تأیید نشده/در مراحل ابتدایی.
- فهرست فناوری (مبتنی بر شواهد)
- **زبان‌ها:** Go (غالب)، Kotlin و Swift، Shell، Python، TypeScript، Kotlin Multiplatform، PostgreSQL، PL/TLA+ (مشخصات رسمی در helix_cluster).
- **AI / LLM:** دسترسی به بیش از ۴۰ ارائه‌دهنده، MCP، RAG، پایگاه‌های داده vector و embeddings، برنامه‌ریزی (HiPlan/MCTS/درخت فکر)، LLMops، معیارسنجی (SWE-bench/HumanEval/MMLU)، TTS (Bark)، SpeechT5، بینایی کامپیوتری + بینایی LLM، محافظت/تیم قرمز.
- **بک‌اند:** Gin، gRPC + Protobuf، HTTP/3 (QUIC)، WebSockets، Angular، React، Kafka/RabbitMQ.
- **داده:** PostgreSQL، SQLite، SQLCipher، Redis، Neo4j، ClickHouse، MinIO/S3/GCS/Azure.
- **زیرساخت / DevOps:** Docker و Prometheus + Grafana، Kubernetes + Helm، Compose، GitHub Actions، Gradle، Make، OpenTelemetry، QEMU/Libvirt/Parallels.
- **آزمون / تضمین کیفیت:** HelixQA، چارچوب‌های چالش با دروازه‌های جهش، `go test -race`، ابزارهای رگرسیون بصری، آزمون دستگاه ADB، SonarQube، اسکن امنیتی (semgrep/gosec/trivy/snyk/gitLeaks/nancy)، بررسی مدل TLA+.

Selected architecture

HelixAgent — ensemble debate flow



an ensemble LLM service where providers debate under a four-stage protocol and ship the answer they — **HelixAgent** 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)

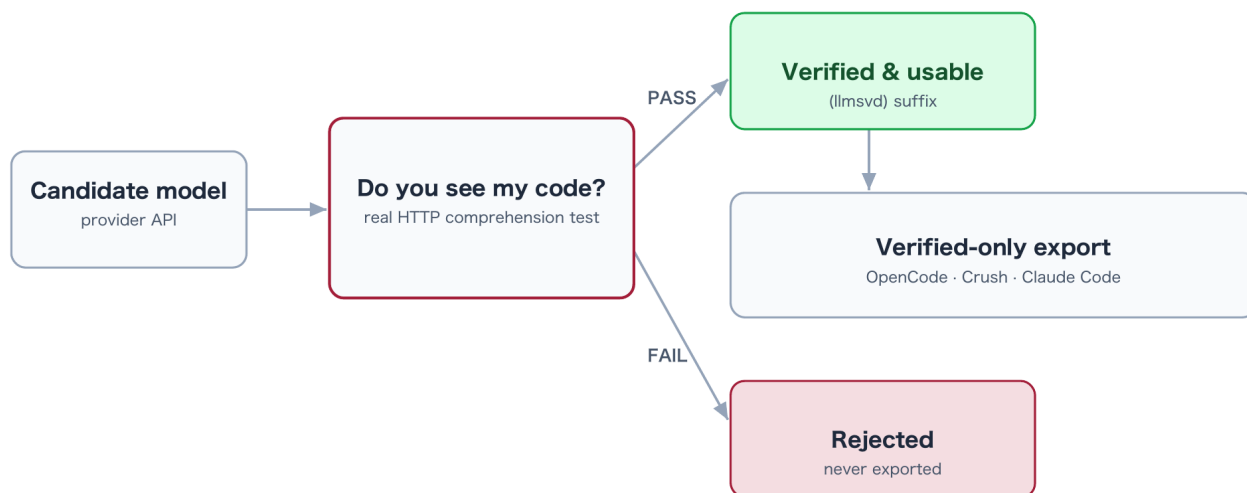
heterogeneous
nodes T1–T8

L0 · Hardware Substrate (GPU → edge SBC)

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

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

LLMsVerifier — mandatory verification gate



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