



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. Обзор — сначала общая картина

Это единое, целостное портфолио, используемое как vasic.digital, так и milosvasic.ru. Оно описывает не разрозненные побочные проекты, а **тщательно спроектированный флот: крупные продуктовые приложения**, построенные на базе **десятков небольших, развязанных, независимо протестированных модулей**, где всё подчинено общей инженерной **Constitution** и проверяется **антиблефовой дисциплиной QA**. Именно эта структура — ключевое отличие. Большинство портфолио — это просто перечень созданных продуктов; здесь же представлена система, в которой каждый продукт собирается из проверенных, переиспользуемых компонентов, каждый компонент подчиняется одним и тем же нерушимым правилам, а за каждой заявленной возможностью стоит подтверждённое доказательство. Основной язык — **Go**, а Kotlin/KMP, TypeScript/React, Python, Swift и Shell применяются там, где они действительно оптимальны: **Go** — для высоконагруженных сервисов и библиотек, Kotlin — для развёртывания и кроссплатформенных мобильных решений, TypeScript — для типизированных фронтендов, Python — для склейки AI/ML.

Целостность этой работы обеспечивается не амбициями, а механической дисциплиной. Общая **Constitution** распространяется как Git-сабмодуль и наследуется во флоте из 140+ репозиториев, так что одно изменение правил мгновенно применяется везде; антиблефовый слой QA отказывается фиксировать прохождение теста без доказательств на уровне исполнения. Семейство продуктов **Helix**, флот утилит и инструментарий **Server Factory** — три воплощения одной идеи: создай один раз, используй везде и докажи, что это работает, прежде чем объявить задачу выполненной.

Далее всё представлено в порядке приоритета:

1. **Принципы управления и QA** — HelixConstitution, HelixQA (дисциплина, делающая всё остальное надёжным).
2. **Семейство продуктов Helix** — жизненный цикл разработки AI (сначала HelixTrack).
3. **Инфраструктура LLM** — абстракция провайдеров, оркестрация, верификация.
4. **Утилиты vasic-digital** — самостоятельные инструменты продуктового уровня.
5. **Server Factory** — наследие автоматизации инфраструктуры (на последнем месте).

Объединяющий тезис: **функция считается готовой только тогда, когда реальный пользователь может ею воспользоваться, и есть зафиксированные доказательства этого.**

2. Принципы управления и QA

Эти два пункта стоят на первом месте, поскольку всё остальное в этом портфолио заимствует у них свою достоверность. Вместе они превращают «доверьтесь, это работает» в проверяемый факт:

Constitution кодифицирует правила, а **HelixQA** доказывает, что они соблюдаются.

- **HelixConstitution** — универсальный, независимый от проекта свод инженерных правил, распространяемый как Git-сабмодуль и наследуемый во флоте из 140+ репозиториях. Антиблефовые проверки, защита от ложноположительных результатов, безопасность данных и хостов, дисциплина покрытия; наследование с возможностью расширения, но без ослабления; механизмы распространения, сканирующие флот на наличие обязательных условий; каждая проверка сопровождается мутационным тестом, доказывающим, что она не фиктивна.
- **HelixQA** — антиблефовая оркестрация QA (на **Go**). Написанные вручную **YAML** тестовые наборы плюс полностью автономные сессии QA с использованием **LLM** и компьютерного зрения на платформах Android, Android TV, Web и Desktop; без зафиксированных доказательств (скриншоты, logcat, видео, трейсы стека) прохождение теста не засчитывается. Обязательный тип тестирования QA, предписанный **Constitution** (§11.4.169).

3. Семейство продуктов Helix

Линейка Helix — это флагманская серия взаимосвязанных продуктов, охватывающая весь жизненный цикл разработки AI: от планирования и спецификаций до сборки, работы с памятью, трансляции и поставки. Каждый продукт самодостаточен, но изначально спроектирован так, чтобы они составляли единое целое — с общей системой управления, повторно используемыми модулями и единой дисциплиной верификации.

- **HelixTrack** — альтернатива JIRA для свободного мира; флагман линейки Helix-Track.
- **HelixAgent** — ансамблевый сервис LLM: несколько моделей обсуждают и выдают согласованный ответ с верификационным отбором провайдеров.
- **HelixCode** — корпоративная распределённая платформа разработки AI; распределяет задачи между управляемыми SSH воркерами с автоматическим созданием контрольных точек и откатом; интерфейсы REST, CLI, TUI, MCP.
- **HelixCluster** — распределённая операционная система для вычислений AI, от датацентровых GPU до периферийных мобильных устройств, под единой панелью управления.
- **HelixBuilder** — конвейер на базе AI для сборки приложений, по одной категории за раз.
- **HelixSkills** — управляемая система навыков для CLI-агентов AI с конституционной основой (навыки, серверы инструментов MCP, плагины Claude Code).
- **HelixSpecifier** — разработка на основе спецификаций, масштабирующая церемониал под объём работы.
- **HelixMemory** — единый мозг памяти для агентов AI, объединяющий четыре лучших в своём классе движка.
- **HelixTranslate** — верифицированный перевод книг на базе моделей; изначально честный, без скрытых откатов.

- **HelixTerminator** — платформа терминала с нулевым доверием: каждая сессия SSH защищена, доступна для совместной работы и поддерживается AI.
- **HelixGitpx** — федеративный Git на базе десятка хостов; единый источник истины, зеркалируемый повсеместно.
- **HelixOTA** — универсальные бесшовные обновления по воздуху; изначально защищены от «окирпичивания».
- **HelixPlay** — превращает любую машину GPU в облачный игровой терминал.
- **Helix-Flow** — продукт для инференса на платформе Helix. *НЕВЕРИФИЦИРОВАН / заблокирован до появления исходников: в публичном репозитории пока только однострочный README; представлен на уровне продукта только при наличии полноценной документации.*

4. Инфраструктура LLM

Под продуктами лежит основа, делающая их независимыми от провайдеров и надёжными: единый интерфейс для десятков провайдеров LLM, единая панель управления для бесголовых агентов-кодеров и единый источник истины для верификации. Этот слой позволяет всему, что выше, подменять модели, переживать сбои провайдеров и отказываться доверять LLM, если она не может доказать понимание задачи.

- **HelixLLM** — один бинарник, шесть режимов: инференс, совместимый с OpenAI и Anthropic, поверх HTTP/3, локальный llama.cpp, цепочка откатов с оценкой, конвейер RAG, агенты ReAct.
- **LLMProvider** — единый интерфейс для 43 провайдеров с встроенными предохранителями, повторными попытками и мониторингом состояния.
- **LLMOrchestrator** — единая панель управления для всех бесголовых агентов-кодеров CLI (OpenCode, Claude Code, Gemini, Junie, Qwen Code).
- **LLMsVerifier** — верификация, мониторинг, оптимизация: единый источник истины для метаданных LLM/провайдеров/верификации с обязательным тестом на понимание модели.

5. vasic-digital: утилиты

Автономные инструменты промышленного уровня, каждый из которых решает сложную задачу по своему — и заодно испытывает на прочность библиотеку переиспользуемых модулей. В их числе — отказоустойчивая мультипротокольная медиасистема, конвейер преобразования markdown в видеоуроки, синхронизатор документов и баз данных с контентным хешированием. Некоторые честно признают свой уровень зрелости, и эти предупреждения не прячутся, а остаются на виду.

- **Catalogizer** — управление медиаколлекциями с поддержкой нескольких протоколов (SMB/FTP/NFS/WebDAV/локальные хранилища), шифрованием и возможностью самостоятельного хостинга; Go/Gin + API и интерфейс React; мониторинг с устойчивостью к офлайн-режиму; построено на базе 21 подмодуля `digital.vasic.*`.
- **Courses-Creator** — конвейер преобразования markdown в видеоуроки; обогащение с помощью нескольких LLM, TTS (Bark/SpeechT5), плееры для настольных, мобильных и веб-платформ; режим работы без ключа API.

- **VisionEngine** — автономный инструментарий Go, объединяющий классическое компьютерное зрение с мультипровайдерными LLM; графы навигации с экспортом в BFS + DOT/JSON/Mermaid; сборка с условной компиляцией OpenCV.
- **DocProcessor** — карта функций на основе документации с отслеживанием покрытия верификацией; извлечение данных с помощью LLM или эвристик/офлайн-режима; лицензия Apache-2.0.
- **Docs Chain** — двунаправленная атомарная синхронизация документов и баз данных с контентным хешированием (инкрементальные пересчёты по принципу Salsa на основе DAG). *Этапы 1–5 — ЗЕЛЁНЫЙ; этапы 6–7 — В ПЛАНАХ.*
- **Herald** — надёжные мультиканальные уведомления с разрешением интенгов на трёх уровнях (команда → LLM → уточнение) на естественном языке; первый потребитель Docs Chain.
- **task_bridge** — автономная двунаправленная синхронизация задач и досок (SQLite как единый источник истины [?] документы [?] ClickUp). *Каркас P1 — логика синхронизации пока не реализована.*
- **Vasic Digital: Набор переиспользуемых модулей** — «стандартная библиотека» `digital.vasic.*`: инфраструктурные примитивы, строительные блоки AI и защитные механизмы на основе LLM, а также зеркало Kotlin Multiplatform. *Некоторые репозитории организации находятся на стадии КАРКАСА/РАЗРАБОТКИ — помечены, но не выпущены.*

6. Server Factory (наследие инфраструктурной автоматизации — на последнем месте по замыслу)

На последнем месте не из-за качества, а по замыслу: инструментарий Server Factory появился раньше линейки AI и демонстрирует, где впервые сформировалась философия «создай один раз, используй везде». Его флагман — почтовый сервер, который описывается в JSON и развёртывается где угодно, — это зрелый, хорошо протестированный продукт; остальные компоненты представлены в своём истинном, разноуровневом состоянии зрелости, без приукрашиваний.

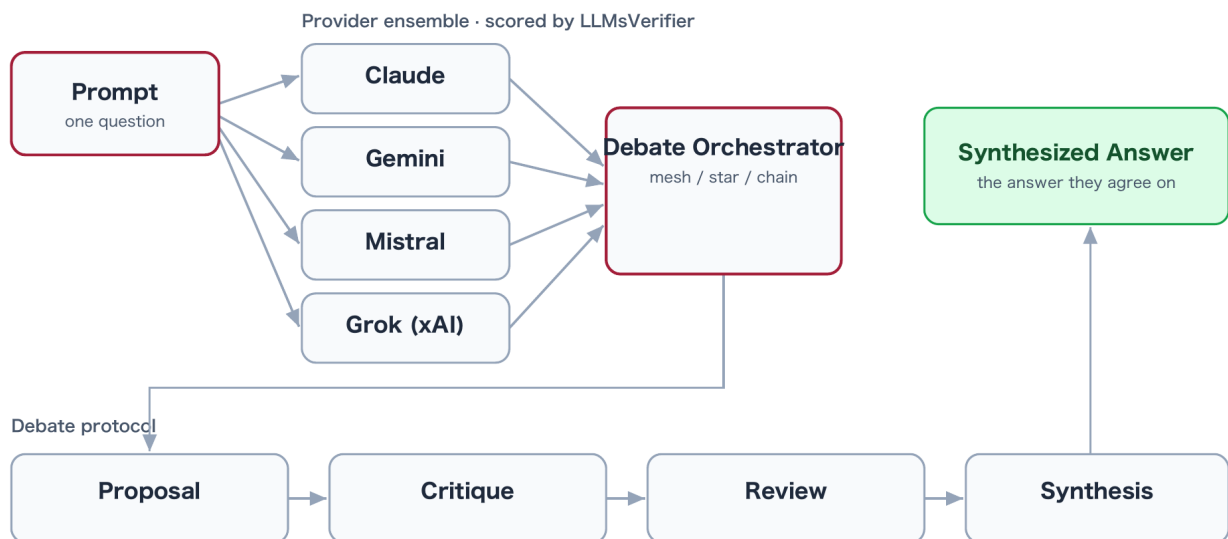
- **Mail Server Factory** — декларативное описание в JSON → полностью развёрнутый почтовый сервер в Docker для 12 типов подключений и 25 дистрибутивов Linux; корпоративная безопасность; 439 пройденных тестов и чистая проверка SonarQube. Флагман организации Server-Factory.
- **Server Factory: Базовый фреймворк** — общий движок Kotlin, на котором строятся все фабрики.
- **Qemu-Utils** — управление образами виртуальных машин QEMU как артефактами: загрузка/кеширование/запуск, сжатие/публикация, сетевые мосты/TAP, установка с ISO; Linux и macOS.
- **Parallels-Utils** — сжатие, публикация и загрузка образов виртуальных машин Parallels (macOS) через простые конфигурационные файлы.
- **Server Factory: Дополнительные компоненты** — фабрики сервисов (веб/SonarQube/кеширующий прокси), пакеты определений и утилиты. *Фабрики сервисов документированы как заглушки — НЕПРОВЕРЕНЫ / на ранней стадии.*

7. Технологический индекс (на основе фактических данных)

- **Языки:** Go (доминирующий), Kotlin и Kotlin Multiplatform, TypeScript, Python, Swift, Shell; PL/pgSQL; TLA+ (формальные спецификации в helix_cluster).
- **AI / LLM:** доступ к 40+ провайдерам, MCP, RAG, базы данных vector и embeddings, планирование (HiPlan/MCTS/Tree-of-Thoughts), LLMOps, бенчмаркинг (SWE-bench/HumanEval/MMLU), TTS (Bark/SpeechT5), компьютерное зрение + LLM-vision, защитные механизмы/красная команда.
- **Бэкэнд:** Gin, gRPC + Protobuf, HTTP/3 (QUIC), WebSockets, Angular, React, Kafka/RabbitMQ.
- **Данные:** PostgreSQL, SQLite, SQLCipher, Redis, Neo4j, ClickHouse, MinIO/S3/GCS/Azure.
- **Инфраструктура / DevOps:** Docker и Compose, Kubernetes + Helm, Prometheus + Grafana, OpenTelemetry, QEMU/Libvirt/Parallels; GitHub Actions, Gradle, Make.
- **Тестирование / Контроль качества:** HelixQA, тестовые стенды с мутационными шлюзами, `go test -race`, инструменты визуальной регрессии, тестирование на устройствах через ADB, SonarQube, сканирование безопасности (semgrep/gosec/trivy/snyk/gitleaks/nancy), проверка моделей 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)

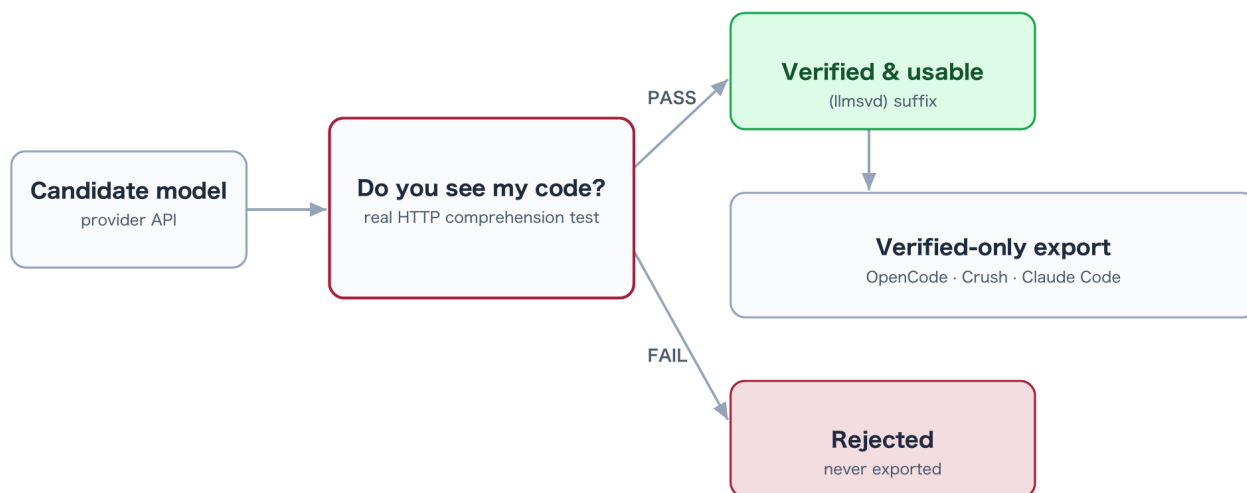
heterogeneous
nodes T1–T8

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.