



Miloš Vasić

Партфоліа · Сямейства Helix · vasic-digital ·
Server Factory

Аб'яднанае партфоліа на аснове доказаў — сямейства прадуктаў Helix, флот утыліт vasic-digital і інструментальны ланцужок Server Factory на агульнай інжынернай Constitution.

140+

ФЛОТ РЕПАЗІТОРЫЯЎ

43

LLM ПАСТАЎШЧЫКІ

25

РАЗГОРНУТЫЯ
ДЫСТРЫБУТЫВЫ
LINUX

439

ПАСПЯХОВА
ПРАЙШОЎШЫЯ ТЭСТЫ
· MAIL SERVER
FACTORY

1. Агляд — спачатку ўсё ў цэлым

Гэта адзінае, аб'яднанае партфоліё, якое выкарыстоўваецца як vasic.digital, так і milosvasic.ru. Яно апісвае не разрозненыя дадатковыя праекты, а наўмысна створаны **флот: вялікія прадуктовыя дадаткі**, якія грунтуюцца на **дзясятках невялікіх, раз'яднаных, незалежна пратэставаных модуляў**, усё гэта кіруецца агульнай інжынернай **Constitution** і правяраецца **антыблефавай дысцыплінай забеспячэння якасці**. Менавіта такая структура з'яўляецца адметнай рысай.

Большасць партфоліё — гэта проста пералік створаных рэчаў; тут жа — сістэма, у якой кожны прадукт збіраецца з правяраных, шматразова выкарыстоўваемых кампанентаў, кожны кампанент падпарадкоўваецца аднолькавым неабмеркаваным правілам, а за кожнай дэклараванай магчымасцю стаіць зафіксаваная доказынасць. Асноўная мова — **Go**, а таксама Kotlin/KMP, TypeScript/React, Python, Swift і Shell там, дзе кожная сапраўды найлепш падыходзіць: Go — для высоканавантажваных сэрвісаў і бібліятэк, Kotlin — для разгортвання і кросплатформавых мабільных дадаткаў, TypeScript — для тыпізаваных фронтэндаў, Python — для звязвання AI/ML.

Што робіць гэтую працу цэласнай, так гэта тое, што дысцыпліна тут механічная, а не дэкларатыўная. Агульная **Constitution** размяшчаецца як Git-падмодуль і наследуецца ўсім флотам з больш чым 140 рэпазіторыямі, таму адна змена правіла распаўсюджваецца паўсюль; антыблефавы слой забеспячэння якасці адмаўляецца фіксаваць паспяховае праходжанне тэсту без доказу на ўзроўні выканання. Сямейства **Helix**, флот утыліт і інструментальны ланцужок **Server Factory** — гэта тры праявы адной ідэі: ствары адзін раз, выкарыстоўвай паўсюль і дакажы, што гэта працуе, перш чым абвешчаць пра завяршэнне.

Усё ніжэй падаецца ў парадку прыярытэту:

1. **Кіраванне і слупкі забеспячэння якасці** — HelixConstitution, HelixQA (дысцыпліна, якая робіць усё астатняе надзейным).
2. **Сямейства прадуктаў Helix** — жыццёвы цыкл распрацоўкі AI (спачатку HelixTrack).
3. **Інфраструктура LLM** — абстракцыя пастаўшчыкоў, аркестрацыя, праверка.
4. **Утыліты vasic-digital** — самастойныя інструменты прадуктовага ўзроўню.
5. **Server Factory** — спадчына аўтаматызацыі інфраструктуры (апошняе месца ў рэйтынгі).

Аб'ядноўваючы тэзіс: **функцыя лічыцца гатовай толькі тады, калі рэальны карыстальнік можа ёю скарыстацца і ёсць зафіксаваныя доказы гэтага.**

2. Кіраванне і слупкі забеспячэння якасці

Гэтыя два пункты стаяць на першым месцы, бо ўся астатняя частка партфоліё пазычае сваю даверанасць менавіта ў іх. Разам яны ператвараюць «даверцеся мне, гэта працуе» ў правяраны факт: **Constitution** фіксуе правілы, а **HelixQA** даказвае, што іх выканалі.

- **HelixConstitution** — універсальны, незалежны ад праекта збор інжынерных правіл, які размяшчаецца як Git-падмодуль і наследуецца ўсім флотам з больш чым 140 рэпазіторыямі. Антыблефавыя бар'еры доказнасці, імунітэт да ілжыва-пазітыўных вынікаў, бяспека даных і хостаў, дысцыпліна пакрыцця; наследаванне з магчымасцю пашырэння, але без аслаблення; бар'еры распаўсюджвання, якія правяраюць наяўнасць абавязковых пунктаў ва ўсіх рэпазіторыях; кожны бар'ер суправаджаецца мутацыйным тэстам, які даказвае, што ён не фіктыўны.
- **HelixQA** — аркестрацыя антыблефавага забеспячэння якасці (Go). Банкі тэстаў, напісаныя на YAML, а таксама цалкам аўтаномныя сесіі забеспячэння якасці з выкарыстаннем LLM і камп'ютарнага зроку на платформах Android, Android TV, Web і Desktop; паспяховае праходжанне фіксуецца толькі пры наяўнасці зафіксаваных доказаў (здымкі экрана, logcat, відэа, stack traces). Абавязковы тып тэсту забеспячэння якасці згодна з патрабаваннямі Constitution (§11.4.169).

3. Сямейства прадуктаў Helix

Лінія Helix — гэта флагман: узаемазвязанае сямейства прадуктаў, якое ахоплівае ўвесь жыццёвы цыкл распрацоўкі AI — ад планавання і спецыфікацыі да стварэння, памяці, перакладу і дастаўкі. Кожны з іх з'яўляецца самастойным прадуктам, але задума праектавання ў тым, каб яны складаліся ў адзінае цэлае: аднолькавае кіраванне, аднолькавыя паўторна выкарыстоўваемыя модулі, аднолькавая дысцыпліна пацверджанняў пад усімі імі.

- **HelixTrack** — альтэрнатыва JIRA для свабоднага свету; флагман лініі Helix-Track.
- **HelixAgent** — ансамблевы сэрвіс LLM: некалькі мадэляў абмяркоўваюць і адпраўляюць адказ, з якім згодныя, з выбарам пастаўшчыка на аснове верафікацыі.
- **HelixCode** — платформа распрацоўкі AI класа enterprise з размеркаваннем задач паміж працоўнымі вузламі, кіраванымі SSH, з аўтаматычнымі кантрольнымі кропкамі і адкотам; інтэрфейсы REST, CLI, TUI, MCP.
- **HelixCluster** — размеркаваная аперацыйная сістэма для вылічэнняў AI, ад датацэнтравых GPU да мабільных прылад на краі сеткі, пад адзінай панэллю кіравання.
- **HelixBuilder** — канвеер на аснове AI для стварэння праграм па катэгорыях.
- **HelixSkills** — кіраваная сістэма навыкаў для агентаў CLI AI на аснове канстытуцыі (навыкі, MCP-серверы інструментаў, плагіны Claude Code).
- **HelixSpecifier** — распрацоўка на аснове спецыфікацый, якая маштабуе цырымоніі ў адпаведнасці з аб'ёмам працы.
- **HelixMemory** — адзіны мозг памяці для агентаў AI, які аб'ядноўвае чатыры найлепшыя рухавікі.
- **HelixTranslate** — пераклад кніг з верафікацыйнай мадэлі; празрысты паводле канструкцыі, без маўклівых падменаў.
- **HelixTerminator** — платформа тэрмінала з нулявой даверлівасцю: кожная сесія SSH абароненая, падзеленая і падтрыманая AI.

- **HelixGitpx** — федэраваны Git на дзясятку хостаў; адзіная крыніца праўды, дубляваная паўсюль.
- **HelixOTA** — універсальныя, развязаныя абнаўленні праз паветра; бяспечныя паводле канструкцыі.
- **HelixPlay** — ператварыце любую машыну GPU у ўласны прылада для воблачнага геймінгу.
- **Helix-Flow** — інферэнс-прадукт на платформе Helix. *НЕПАЦВЕРДЖАНЫ / заблакаваны праз крыніцу: публічны рэпазіторый змяшчае толькі аднарадковы README; прадстаўлены на ўзроўні прадукта толькі пры наяўнасці рэальнай дакументацыі.*

4. Інфраструктура LLM

Пад прадуктамі знаходзіцца аснова, якая робіць іх незалежнымі ад пастаўшчыкоў і надзейнымі: адзіны інтэрфейс для дзясяткаў правайдараў LLM, адзіная панэль кіравання для бязгалаўковых агентаў-распрацоўшчыкаў і адзіная крыніца праўды для верафікацыі. Гэта той пласт, які дазваляе ўсім сістэмам вышэй мяняць мадэлі, перажываць збоі пастаўшчыкоў і адмаўляцца давяраць LLM, калі ён не можа даказаць разуменне задачы.

- **HelixLLM** — адзін бінарны файл, шэсць рэжымаў: інферэнс, сумяшчальны з OpenAI і Anthropic, праз HTTP/3, лакальны llama.cpp, ланцужок рэзервовых варыянтаў з ацэнкай, канвеер RAG, агенты ReAct.
- **LLMProvider** — адзін інтэрфейс, 43 пастаўшчыкі, з убудаванымі засцерагальнікамі, паўтарамі і маніторынгам стану.
- **LLMOrchestrator** — адзіная панэль кіравання для кожнага бязгалаўковага агента-распрацоўшчыка CLI (OpenCode, Claude Code, Gemini, Junie, Qwen Code).
- **LLMsVerifier** — верафікацыя, маніторынг, аптымізацыя: адзіная крыніца праўды для метаданых LLM/пастаўшчыка/верафікацыі з абавязковай праверкай разумення мадэлі.

5. vasic-digital утыліты

Аўтаномныя інструменты прафесійнага ўзроўню, кожны з якіх вырашае складаную задачу па-свойму — і, не выпадкова, выпрабоўвае бібліятэку паўторна выкарыстоўваемых мадуляў на практыцы. Яны вар'іруюцца ад устойлівай мультыпрагнатычнай сістэмы медыя да канвеера пераўтварэння markdown у відэакурсы і сінхранізатара дакументаў і баз даных з кантэнт-хэшаваннем. Некаторыя адкрыта дэкларуюць сваю стадыю сталасці, і гэтыя пазнакі захоўваюцца, а не хаваюцца.

- **Catalogizer** — кіраванне медыякалекцыямі праз некалькі пратаколаў (SMB/FTP/NFS/WebDAV/лакальна), з шыфраваннем і магчымасцю самастойнага хостынг; Go/Gin API + React UI; устойлівы маніторынг, які трымае адключэнні; пабудаваны на 21 падмадулі `digital.vasic.*`.
- **Courses-Creator** — канвеер пераўтварэння markdown у відэакурсы; мульты-LLM абогачэнне, TTS (Bark/SpeechT5), прайгравальнікі для дэсктопа, мабільных прылад і вэбу; рэжым працы без ключа API.
- **VisionEngine** — дэкапліраваны Go-набор інструментаў, які аб'ядноўвае класічны камп'ютарны зрок з мультыправайдарскай LLM-візуалізацыяй; навігацыйныя графы з экспартам BFS + DOT/JSON/Mermaid; зборка з умоўным уключэннем OpenCV.
- **DocProcessor** — карта дакументацыі на функцыі з адсочваннем пакрыцця праверкі; LLM ці аўрыстычнае/афлайн-выманне; ліцэнзія Apache-2.0.

- **Docs Chain** — сінхранізацыя дакументаў і баз даных з кантэнт-хэшаваннем, двухбаковая, атамарная (інкрэментнае пералічэнне па DAG у стылі Salsa). *Стадыі 1–5 ЗАВЕРШАНЫ; стадыі 6–7 ПЛАНУЮЦА.*
- **Herald** — надзейныя мультыканальныя апавяшчэнні з вырашэннем намеры на трох узроўнях праз натуральную мову (каманда → LLM → удакладненне); першы спажывец Docs Chain.
- **task_bridge** — дэкапліраваная, двухбаковая сінхранізацыя задач і дошак (SQLite SSoT ↔ дакументы ↔ ClickUp). *Каркас стадыі P1 — логіка сінхранізацыі яшчэ не рэалізавана.*
- **Vasic Digital Бібліятэка паўторна выкарыстоўваемых мадуляў** — “стандартная бібліятэка” `digital.vasic.*`: інфраструктурныя прымітывы, будаўнічыя блокі AI і засцерагальныя механізмы з абаронай LLM, а таксама люстэрка Kotlin Multiplatform. *Некаторыя рэпазіторыі арганізацыі знаходзяцца на стадыі КАРКАС/РАСПРАЦОЎКА — пазначаны, але не выпушчаны.*

6. Server Factory (спадчына інфраструктурнай аўтаматызацыі — апошні паводле рэйтыngu)

Апошні паводле рэйтыngu не па якасці, а паводле задумы: набор інструментаў Server Factory з’явіўся раней за лінейку AI і паказвае, дзе ўпершыню ўвасобілася філасофія “ствары адзін раз, выкарыстоўвай усюды”. Яго флагман — поштавы сервер, які апісваецца ў JSON і разгортваецца дзе заўгодна, — гэта сталая, добра правяраная праграма; астатнія кампаненты пададзены ў сваім сапраўдным, няроўным стане сталасці, а не прыхарашаныя.

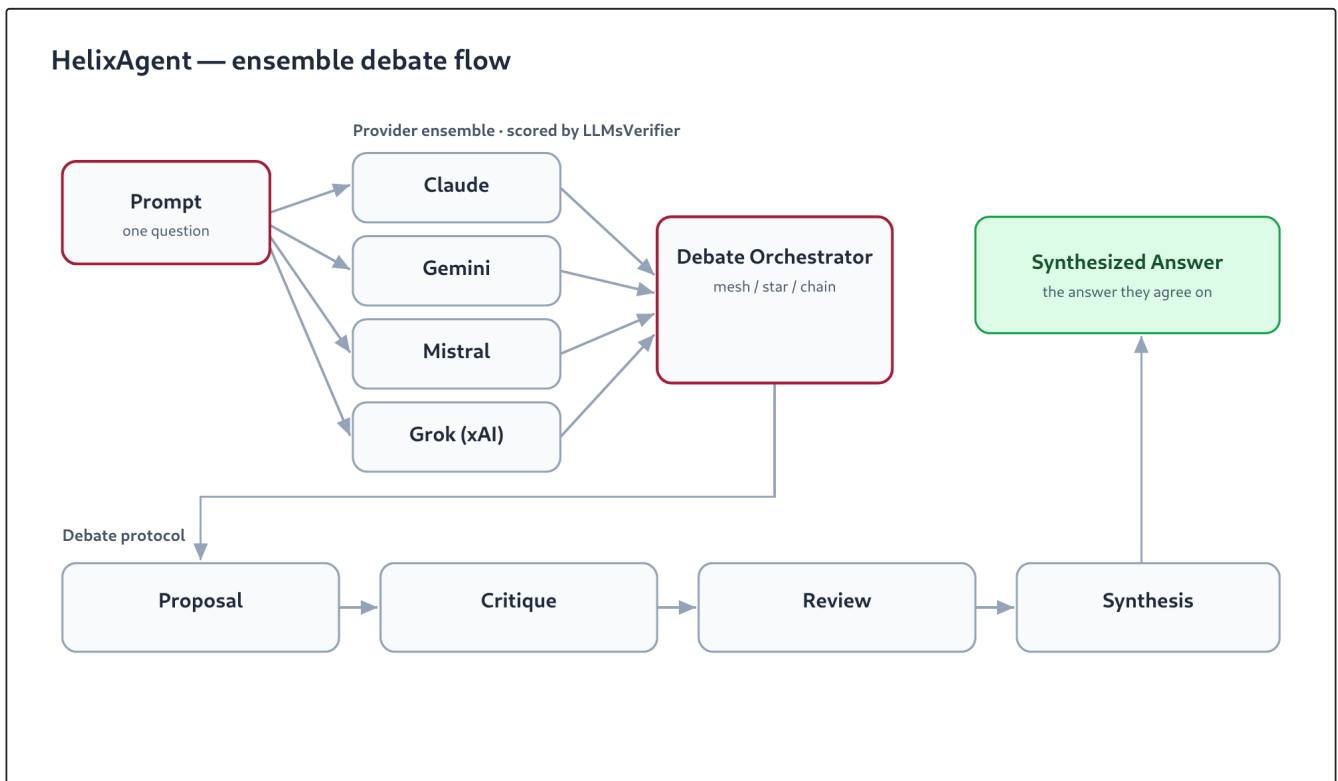
- **Mail Server Factory** — дэкларатыўны JSON → поўнасцю разгорнуты, дакераваны поштавы сервер для 12 тыпаў злучэнняў і 25 дыстрыбутываў Linux; карпаратыўная бяспека; паведамляе аб 439 прайшоўшых тэстах і чыстым кантрольным пункце SonarQube. Флагман арганізацыі Server-Factory.
- **Server Factory Асноўны фрэймворк** — агульны Kotlin-рухавік, на якім будуюцца ўсе фабрыкі.
- **Qemu-Utills** — вобразы QEMU VM, якія кіруюцца як артэфакты: спампоўка/кэшаванне/запуск, сціск/публікацыя, масткі/TAP-сетка, усталяванне з ISO; Linux + macOS.
- **Parallels-Utills** — сціск, публікацыя і атрыманне вобразаў Parallels (macOS) VM праз простыя файлы наладак.
- **Server Factory — Дадатковыя кампаненты** — фабрыкі сэрвісаў (Web/SonarQube/Caching-Proxy), пакеты азначэнняў і ўтыліты. *Фабрыкі сэрвісаў маюць толькі апісанне-запаўняльнік — НЕПРАВЕРНЫЯ / ранняя стадыя.*

7. Індэкс тэхналогій (на аснове фактычных дадзеных)

- **Мовы:** Go (дамінірующая), Kotlin і Kotlin Multiplatform, TypeScript, Python, Swift, Shell; PL/pgSQL; TLA+ (фармальныя спецыфікацыі ў helix_cluster).
- **AI / LLM:** доступ да 40+ пастаўшчыкоў, MCP, RAG, базы дадзеных vector і embeddings, планаванне (HiPlan/MCTS/Tree-of-Thoughts), LLMops, бенчмаркінг (SWE-bench/HumanEval/MMLU), TTS (Bark/SpeechT5), камп’ютарны зрок + LLM-vision, засцерагальныя механізмы/рэд-тымінг.
- **Бэкэнд:** Gin, gRPC + Protobuf, HTTP/3 (QUIC), WebSockets, Angular, React, Kafka/RabbitMQ.
- **Дадзеныя:** PostgreSQL, SQLite, SQLCipher, Redis, Neo4j, ClickHouse, MinIO/S3/GCS/Azure.

- **Інфраструктура / DevOps:** Docker і Compose, Kubernetes + Helm, Prometheus + Grafana, OpenTelemetry, QEMU/Libvirt/Parallels; GitHub Actions, Gradle, Make.
- **Тэсціраванне / Кантроль якасці:** HelixQA, тэставыя каркасы з мутацыйнымі шлюзамі, `go test -race`, інструменты візуальнай рэгрэсіі, тэсціраванне на прыладах праз ADB, SonarQube, сканіраванне бяспекі (semgrep/gosec/trivy/snyk/gitleaks/nancy), праверка мадэляў TLA+.

Абраная архітэктур

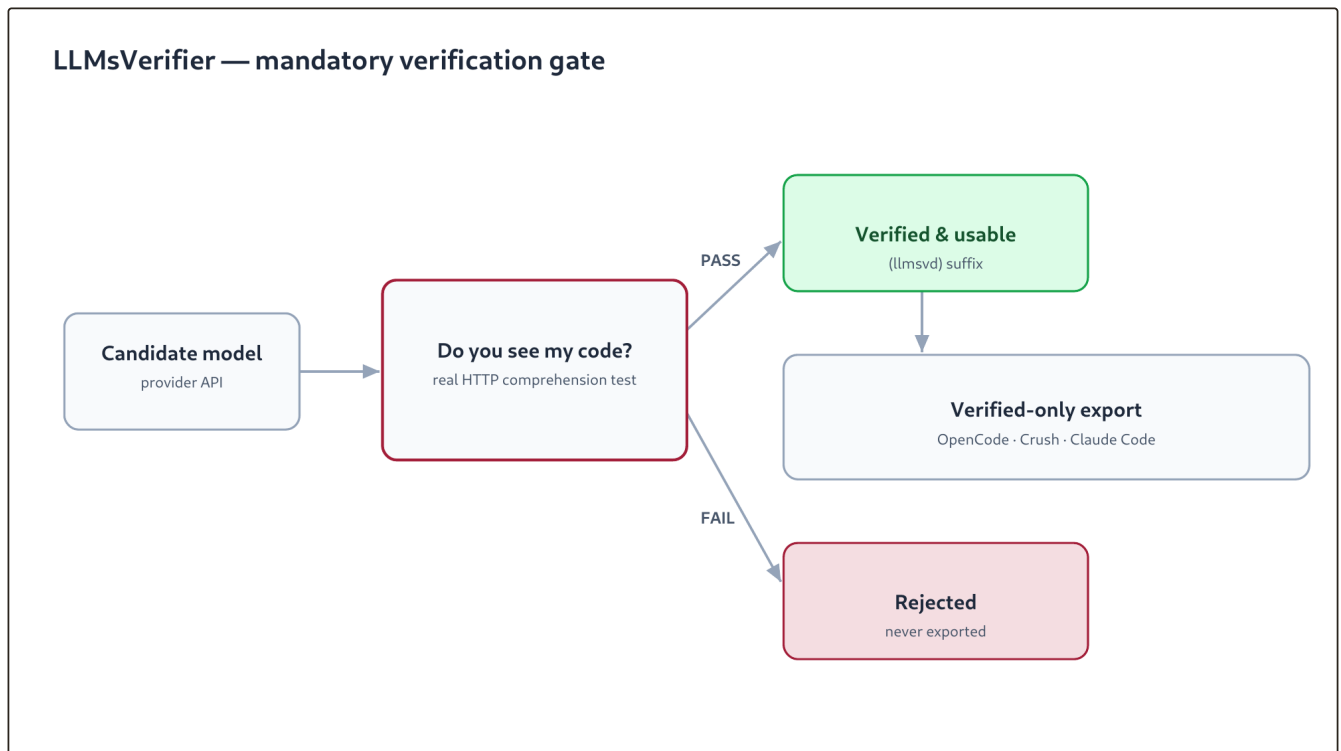


HelixAgent — ансамблевы LLM-сэрвіс, дзе пастаўшчыкі абмяркоўваюць пытанні ў рамках чатырохэтапнага пратаколу і адпраўляюць узгоднены адказ, ацэнены па LLMsVerifier.

HelixCluster — seven-layer stack14 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 — розрахована операційна система для AI-вирішення — ад GPU у датацентрах до мобільних пристроїв на краю мережі, під адзіним контрольним центром.



LLMsVerifier — перевірка, моніторинг, аптимізація: адзіная крыніца праўды для метаданых LLM/пастаўшчык/правірка, якая кантралюецца абавязковай правіркай разумення мадэлі.