Vasic Digital

AI-नेटिव सॉफ़्टवेयर इंजीनियरिंग · साक्ष्य-आधारित पोर्टफोलियो

Helix उत्पाद परिवार, वासिक-डिजिटल उपयोगिता बेड़ा, और Server Factory टूलचेन — AI विकास प्रणालियाँ, जो विश्वसनीयता के लिए निर्मित हैं, एक साझा इंजीनियरिंग Constitution पर।

140+	43	25	439
रिपॉज़िटरी बेड़ा	LLM प्रदाता	प्रावधानित LINUX वितरण	सफल परीक्षण · MAIL SERVER FACTORY

१. अवलोकन — पहले पूरी तस्वीर

यह एक एकीकृत पोर्टफोलियो है जिसका उपयोग vasic.digital और milosvasic.ru दोनों द्वारा किया जाता है। यह अलग-अलग साइड प्रोजेक्ट्स का संग्रह नहीं, बल्कि एक सोच-समझकर तैयार की गई संरचना है: बड़े उत्पाद अनुप्रयोग जो दर्जनों छोटे, अलग-अलग, स्वतंत्र रूप से परीक्षित मॉड्यूल्स पर आधारित हैं। पूरी प्रणाली एक साझा इंजीनियरिंग **Constitution** द्वारा संचालित है और धोखा-रहित गुणवत्ता आश्वासन के अनुशासन से सत्यापित है। यही संरचना इसे अलग बनाती है। अधिकांश पोर्टफोलियो उन चीज़ों की सूची होते हैं जो बनाई गईं; यह एक ऐसी प्रणाली है जिसमें हर उत्पाद सिद्ध, पुनःप्रयोगी घटकों से तैयार किया जाता है, हर घटक एक ही अटल नियमों का पालन करता है, और हर विज्ञापित क्षमता के पीछे ठोस प्रमाण मौजूद होता है। मुख्य भाषा **Go** है, जबकि Kotlin/KMP, TypeScript/React, Python, Swift और Shell का उपयोग वहाँ किया जाता है जहाँ वे वास्तव में सबसे उपयुक्त होते हैं — Go उच्च-प्रवाह सेवाओं और लाइब्रेरियों के लिए, Kotlin प्रोविज़निंग और क्रॉस-प्लेटफ़ॉर्म मोबाइल के लिए, TypeScript टाइप्ड फ्रंटएंड्स के लिए, Python AI/ML को जोड़ने के लिए।

इस कार्य-संग्रह को एकजुट करने वाली बात यह है कि यह अनुशासन आकांक्षात्मक नहीं, बल्कि यांत्रिक है। एक साझा **Constitution** Git सबमॉड्यूल के रूप में शिप होता है और 140 से अधिक रिपॉजिटरीज़ के समूह में विरासत में मिलता है, इसलिए एक नियम में बदलाव हर जगह फैल जाता है; धोखा-रहित गुणवत्ता आश्वासन परत बिना रनटाइम प्रमाण के पास दर्ज नहीं करती। **Helix** परिवार, यूटिलिटी समूह और **Server Factory** टूलचेन — ये तीनों एक ही विचार के अभिव्यक्ति हैं: एक बार बनाओ, हर जगह प्रयोग करो, और इसे पूरा कहने से पहले सिद्ध करो कि यह काम करता है।

नीचे दी गई सभी जानकारी प्राथमिकता क्रम में प्रस्तुत है:

1. शासन और गुणवत्ता आश्वासन के स्तंभ — HelixConstitution, HelixQA (वह अनुशासन जो बाकी सबको विश्वसनीय बनाता है)।
2. **Helix** उत्पाद परिवार — AI-विकास चक्र (HelixTrack पहले)।
3. **LLM** इंफ्रास्ट्रक्चर — प्रदाता अमूर्तीकरण, ऑर्केस्ट्रेशन, सत्यापन।
4. वासिक-डिजिटल यूटिलिटीज़ — उत्पाद-स्तरीय स्वतंत्र उपकरण।
5. **Server Factory** — इंफ्रास्ट्रक्चर-ऑटोमेशन विरासत (अंतिम स्थान पर)।

एकीकृत सिद्धांत: एक सुविधा तब पूरी मानी जाती है जब कोई वास्तविक उपयोगकर्ता उसका उपयोग कर सके और उसके प्रमाण के रूप में दर्ज सबूत मौजूद हो।

---

२. शासन और गुणवत्ता आश्वासन के स्तंभ

ये दोनों सबसे पहले इसलिए आते हैं क्योंकि इस पोर्टफोलियो की बाकी सभी चीज़ें अपनी विश्वसनीयता इन्हीं से उधार लेती हैं। साथ मिलकर ये "विश्वास करो, यह काम करता है" को एक ऑडिट योग्य तथ्य में बदल देते हैं — **Constitution** नियमों को कोडित करता है, **HelixQA** सिद्ध करता है कि उनका पालन हुआ।

- **HelixConstitution** — एक सार्वभौमिक, प्रोजेक्ट-निरपेक्ष इंजीनियरिंग नियम-पुस्तिका, जो Git सबमॉड्यूल के रूप में शिप होती है और 140 से अधिक रिपॉजिटरीज़ के समूह में विरासत में मिलती है। धोखा-रहित प्रमाण गेट्स, झूठी-सकारात्मकता से

प्रतिरक्षा, डेटा/होस्ट सुरक्षा, कवरेज अनुशासन; विस्तार-करो-लेकिन-कमज़ोर-न-करो विरासत; ऐसे प्रसार गेट्स जो पूरे समूह में आवश्यक क्लॉज़ की खोज करते हैं; हर गेट के साथ एक म्यूटेशन टेस्ट जो सिद्ध करता है कि यह दिखावा नहीं है।

- **HelixQA** — धोखा-रहित गुणवत्ता आश्वासन ऑर्केस्ट्रेशन (Go)। लिखित YAML टेस्ट बैंक और पूरी तरह स्वायत्त LLM-और-कंप्यूटर-विज़न आधारित गुणवत्ता आश्वासन सत्र Android, Android TV, वेब और डेस्कटॉप पर; बिना दर्ज प्रमाण (स्क्रीनशॉट, लॉगकैट, वीडियो, स्टैक ट्रेस) के कोई पास नहीं। **Constitution** द्वारा अनिवार्य गुणवत्ता आश्वासन टेस्ट-प्रकार (§11.4.169)।

३. **Helix** उत्पाद परिवार

Helix श्रृंखला प्रमुख है: परस्पर जुड़े उत्पादों का एक ऐसा परिवार जो AI-विकास के संपूर्ण जीवनचक्र को समाहित करता है—योजना और विनिर्देश से लेकर निर्माण, स्मृति, अनुवाद और वितरण तक। प्रत्येक उत्पाद अपने आप में एक पूर्ण उत्पाद है, परंतु डिज़ाइन का उद्देश्य यह है कि वे एक-दूसरे के पूरक बनें—समान शासन प्रणाली, समान पुनःप्रयोगी मॉड्यूल, और सभी के पीछे एक ही प्रमाणन अनुशासन।

- **HelixTrack** — स्वतंत्र दुनिया के लिए JIRA का एक विकल्प; Helix-ट्रैक श्रृंखला का प्रमुख उत्पाद।
- **HelixAgent** — समूह LLM सेवा: कई मॉडल आपस में चर्चा करते हैं और जिस उत्तर पर सहमति बनती है, उसे भेजते हैं, साथ ही सत्यापन-आधारित प्रदाता चयन।
- **HelixCode** — उद्यम-स्तरीय वितरित AI विकास मंच; SSH-प्रबंधित श्रमिकों के बीच कार्य विभाजन, स्वचालित चेकपॉइंट/रोलबैक; REST, CLI, TUI, MCP इंटरफेस।
- **HelixCluster** — AI कंप्यूटिंग के लिए एक वितरित ऑपरेटिंग सिस्टम, डेटासेंटर GPU से लेकर एज हैंडहेल्ड डिवाइस तक, एक ही नियंत्रण तंत्र के तहत।
- **HelixBuilder** — अनुप्रयोग निर्माण के लिए AI-संचालित पाइपलाइन, एक समय में एक श्रेणी।
- **HelixSkills** — CLI AI एजेंटों के लिए शासित, संविधान-समर्थित कौशल प्रणाली (कौशल, MCP टूल सर्वर, Claude कोड प्लगइन्स)।
- **HelixSpecifier** — विनिर्देश-आधारित विकास जो कार्य के अनुसार औपचारिकता को समायोजित करता है।
- **HelixMemory** — AI एजेंटों के लिए एकीकृत स्मृति मस्तिष्क, चार सर्वश्रेष्ठ इंजनों का समन्वय।
- **HelixTranslate** — सत्यापित-मॉडल पुस्तक अनुवाद; डिज़ाइन से ही ईमानदार, कभी भी चुपचाप विकल्प नहीं।
- **HelixTerminator** — शून्य-विश्वास टर्मिनल प्लेटफॉर्म: हर SSH सत्र सुरक्षित, साझा और AI-सहायता प्राप्त।
- **HelixGitpx** — एक दर्जन होस्टों पर संघबद्ध Git; एक ही सत्य का स्रोत, हर जगह प्रतिबिंबित।
- **HelixOTA** — सार्वभौमिक, पृथक ओवर-द-एयर अपडेट; डिज़ाइन से ही शून्य-खराबी।
- **HelixPlay** — किसी भी GPU मशीन को अपने क्लाउड-गेमिंग उपकरण में बदलें।
- **Helix-Flow** — Helix-प्लेटफॉर्म अनुमान उत्पाद। असत्यापित / स्रोत पर अवरोधित: इसका सार्वजनिक रिपॉजिटरी में अभी केवल एक-पंक्ति का *README* है; वास्तविक दस्तावेज़ीकरण उपलब्ध होने पर ही उत्पाद गहराई से प्रस्तुत किया जाएगा।

४. **LLM** अवसंरचना

उत्पादों के नीचे वह आधार है जो उन्हें प्रदाता-स्वतंत्र और विश्वसनीय बनाता है: दर्जनों LLM प्रदाताओं पर एक ही इंटरफेस, हेडलेस कोडिंग एजेंटों के लिए एक नियंत्रण तंत्र, और सत्यापन का एक ही स्रोत। यही वह परत है जो ऊपर के सभी उत्पादों को मॉडल बदलने, प्रदाता की विफलताओं से बचने और किसी ऐसे LLM पर भरोसा न करने देती है जो कार्य को समझने का प्रमाण नहीं दे सकता।

- **HelixLLM** — एक बाइनरी, छह मोड: OpenAI- और Anthropic-संगत अनुमान HTTP/3, स्थानीय llama.cpp, स्कोर-आधारित फॉलबैक चेन, RAG पाइपलाइन, ReAct एजेंटों पर।
- **LLMProvider** — एक इंटरफेस, ४३ प्रदाता, जिसमें सर्किट ब्रेकर, पुनःप्रयास और स्वास्थ्य जाँच अंतर्निहित।

- **LLMOrchestrator** — हर हेडलेस CLI कोडिंग एजेंट (OpenCode, Claude Code, Gemini, Junie, Qwen Code) के लिए एक नियंत्रण तंत्र।
- **LLMsVerifier** — सत्यापित करें, निगरानी करें, अनुकूलित करें: LLM/प्रदाता/सत्यापन मेटाडेटा के लिए एकमात्र सत्य का स्रोत, जिसमें अनिवार्य मॉडल समझ गेट शामिल है।

सामग्री

५. वासिक-डिजिटल उपयोगिताएँ

स्वतंत्र, उत्पाद-स्तरीय उपकरण जो अपने-अपने क्षेत्र में एक कठिन समस्या का समाधान करते हैं — और संयोग से नहीं, पुनःप्रयोगी-मॉड्यूल समूह को व्यावहारिक परीक्षण में भी डालते हैं। इनमें एक लचीला बहु-प्रोटोकॉल मीडिया प्रणाली से लेकर मार्कडाउन-से-वीडियो पाठ्यक्रम पाइपलाइन और सामग्री-हैशित दस्तावेज़/डेटाबेस सिंक इंजन तक शामिल हैं; कई अपनी परिपक्वता के बारे में स्पष्ट हैं, और उन चेतावनियों को छिपाया नहीं गया है।

- **Catalogizer** — बहु-प्रोटोकॉल (SMB/FTP/NFS/WebDAV/स्थानीय), एन्क्रिप्टेड, स्व-होस्ट करने योग्य मीडिया संग्रह प्रबंधन; Go/Gin API + React यूआई; ऑफ़लाइन-सहिष्णु निगरानी; 21 `digital.vasic.*` उप-मॉड्यूल पर निर्मित।
- **Courses-Creator** — मार्कडाउन-से-वीडियो पाठ्यक्रम पाइपलाइन; बहु-LLM संवर्धन, TTS (Bark/SpeechT5), डेस्कटॉप/मोबाइल/वेब प्लेयर; API-कुंजी के बिना भी सुचारू संचालन।
- **VisionEngine** — पृथक Go टूलकिट जो शास्त्रीय कंप्यूटर विज़न को बहु-प्रदाता LLM विज़न के साथ जोड़ता है; नेविगेशन ग्राफ़ BFS + DOT/JSON/मरमेड निर्यात के साथ; OpenCV बिल्ड-टैग-गेटेड।
- **DocProcessor** — दस्तावेज़-से-फीचर-मैपिंग जिसमें सत्यापन-कवरेज ट्रैकिंग; LLM या अनुमानात्मक/ऑफ़लाइन निष्कर्षण; Apache-2.0 लाइसेंस।
- **Docs Chain** — सामग्री-हैशित, द्विदिशात्मक, परमाणु दस्तावेज़/डेटाबेस सिंक (DAG पर साल्सा-शैली वृद्धिशील पुनर्गणना)। चरण १–५ पूर्ण; चरण ६–७ नियोजित।
- **Herald** — विश्वसनीय बहु-चैनल सूचनाएँ प्राकृतिक-भाषा, तीन-स्तरीय इरादा समाधान (कमांड → LLM → स्पष्टीकरण) के साथ; Docs Chain का पहला उपभोक्ता।
- **task_bridge** — पृथक, द्विदिशात्मक कार्य/बोर्ड सिंक (SQLite एकल स्रोत सत्य ? दस्तावेज़ ? ClickUp)। चरण १ ढाँचा — सिंक तर्क अभी लागू नहीं किया गया।
- **Vasic Digital** पुनःप्रयोगी मॉड्यूल सूट — `digital.vasic.*` "मानक पुस्तकालय": आधारभूत संरचना प्रिमिटिव्स, AI निर्माण खंड, और रक्षात्मक-LLM सुरक्षा-रेलिंग, साथ ही Kotlin Multiplatform मिरर। कई संगठन भंडार प्रारंभिक/कार्य-प्रगति में हैं — चिह्नित, प्रकाशित नहीं।

६. Server Factory (बुनियादी-स्वचालन विरासत — अंतिम स्थान पर)

गुणवत्ता के आधार पर नहीं, बल्कि डिज़ाइन के अनुसार अंतिम स्थान पर: Server Factory टूलचेन AI श्रृंखला से पहले का है और दिखाता है कि "एक बार बनाओ, हर जगह प्रयोग करो" का दर्शन सबसे पहले कहाँ आकार लेना शुरू हुआ। इसका प्रमुख उत्पाद — एक मेल सर्वर जिसे आप JSON में वर्णित करते हैं और कहीं भी प्रावधान कर सकते हैं — एक परिपक्व, अच्छी तरह परीक्षित उत्पाद है; सहायक उपकरणों को उनकी वास्तविक, विविध परिपक्वता के साथ प्रस्तुत किया गया है, न कि सजावट के साथ।

- **Mail Server Factory** — घोषणात्मक JSON → 12 कनेक्शन प्रकारों और 25 Linux वितरणों पर पूरी तरह प्रावधानित, डॉकरीकृत मेल सर्वर; उद्यम सुरक्षा; 439 सफल परीक्षण और स्वच्छ SonarQube गेट। सर्वर-फैक्ट्री संगठन का प्रमुख उत्पाद।
- **Server Factory** कोर फ्रेमवर्क — वह साझा Kotlin इंजन जिस पर हर फैक्ट्री निर्मित होती है।

- **Qemu-Utils** — QEMU वर्चुअल मशीन छवियों का कलाकृतियों की तरह प्रबंधन: डाउनलोड/कैश/चलाएँ, संपीड़ित/प्रकाशित करें, ब्रिज/TAP नेटवर्किंग, ISO इंस्टॉल; Linux + macOS।
- **Parallels-Utils** — Parallels (macOS) वर्चुअल मशीन छवि संपीड़न, प्रकाशन और पुनर्प्राप्ति सरल सेटिंग फ़ाइलों के माध्यम से।
- **Server Factory —** अतिरिक्त घटक — सेवा फैक्ट्रियाँ (वेब/SonarQube/कैशिंग-प्रॉक्सी), परिभाषा पैक, और उपयोगिताएँ। सेवा फैक्ट्रियाँ प्रलेखन-स्थानधारक हैं — अप्रमाणित/प्रारंभिक चरण।

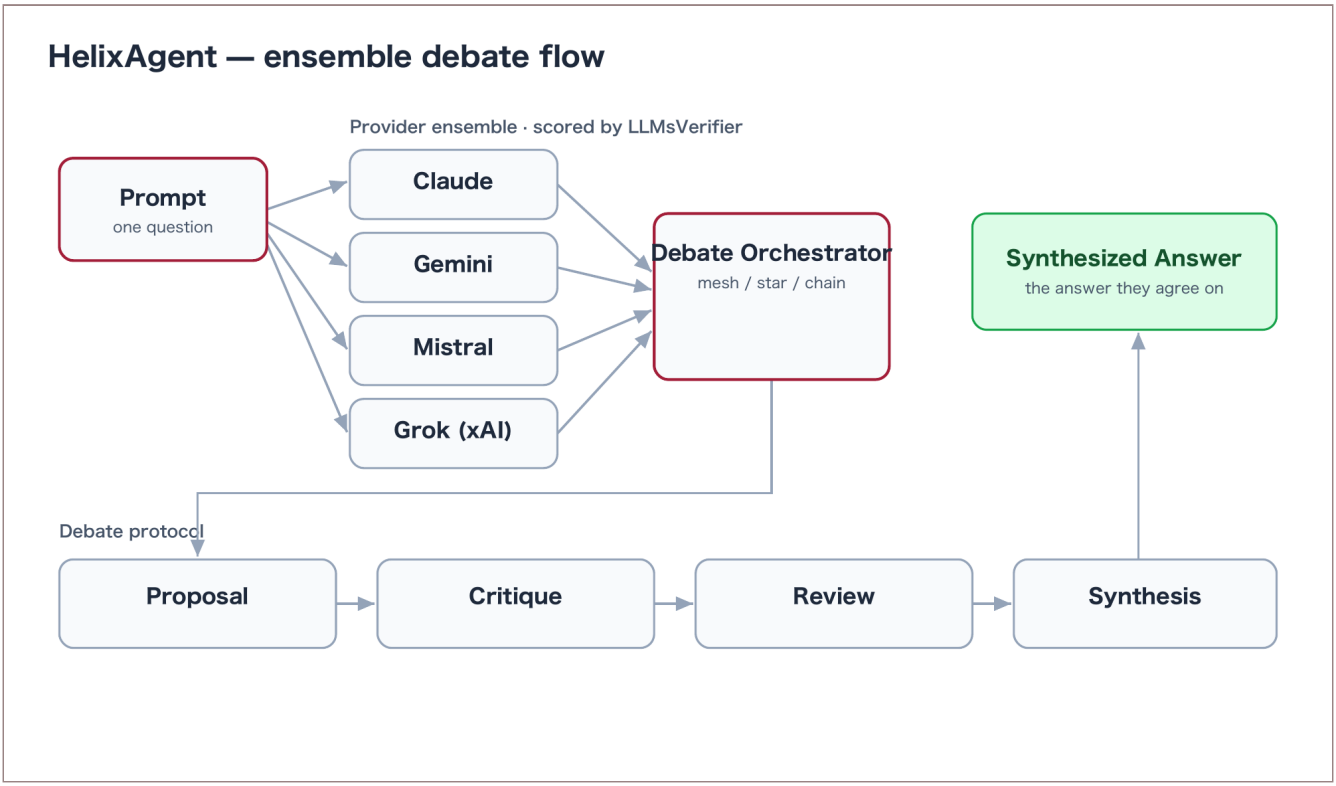
सामग्री

---

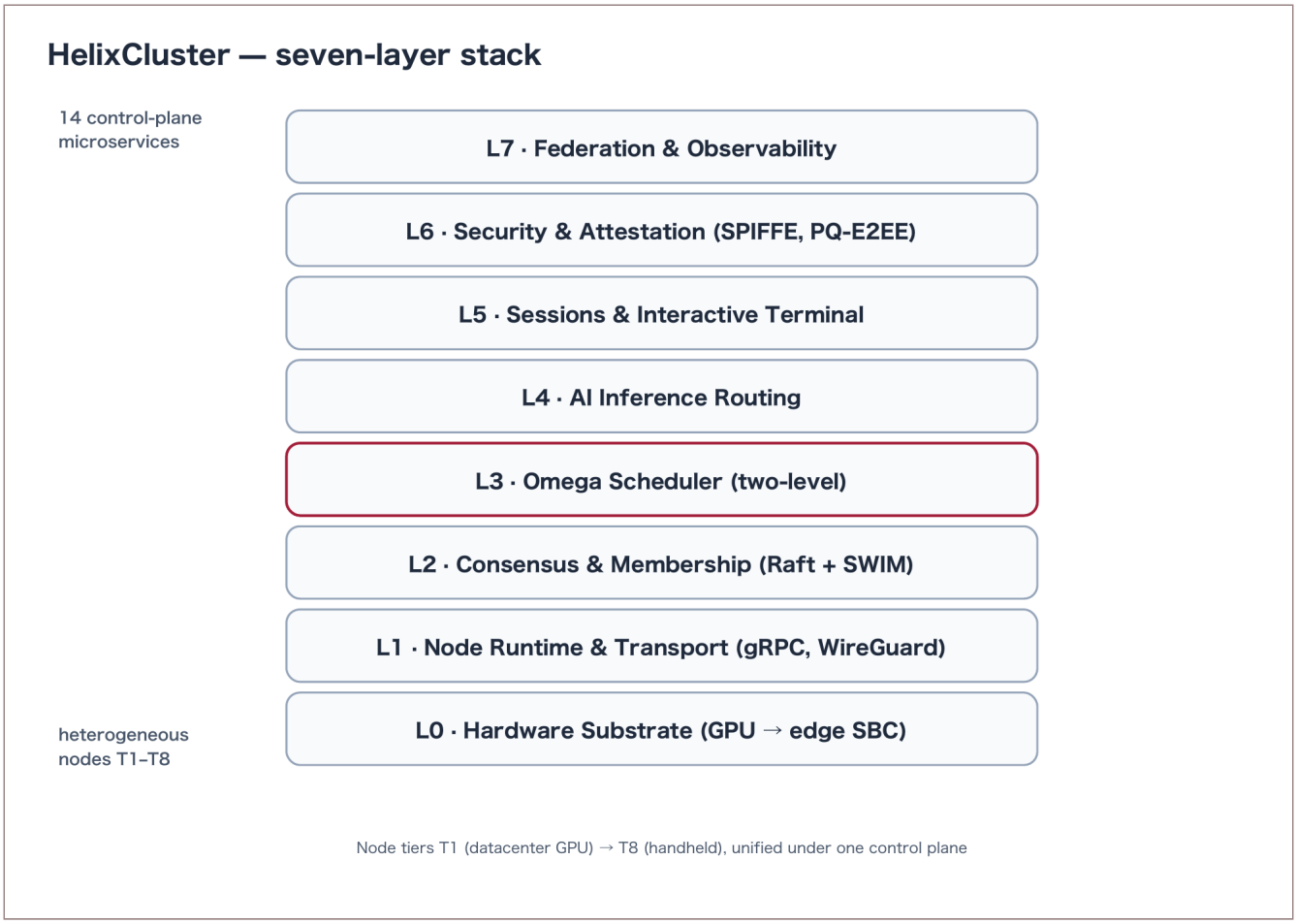
७. प्रौद्योगिकी सूचकांक (साक्ष्य-आधारित)

- भाषाएँ: Go (प्रमुख), Kotlin और Kotlin Multiplatform, TypeScript, Python, Swift, Shell; PL/pgSQL; TLA+ (हेलिक्स_क्लस्टर में औपचारिक विनिर्देश)।
- **AI / LLM:** ४०+ प्रदाता पहुँच, MCP, RAG, vector डेटाबेस और embeddings, योजना (HiPlan/MCTS/ट्री-ऑफ-थॉट्स), LLMOps, बेंचमार्किंग (SWE-bench/HumanEval/MMLU), TTS (Bark/SpeechT5), कंप्यूटर-विज़न + LLM-विज़न, सुरक्षा उपाय/रेड-टीम।
- बैकएंड: Gin, gRPC + Protobuf, HTTP/3 (QUIC), वेबसॉकेट्स, Angular, React, Kafka/RabbitMQ।
- डेटा: PostgreSQL, SQLite, SQLCipher, Redis, Neo4j, ClickHouse, MinIO/S3/GCS/Azure।
- इन्फ्रा / डेवऑप्स: Docker और कंपोज़, Kubernetes + Helm, Prometheus + Grafana, OpenTelemetry, QEMU/Libvirt/Parallels; GitHub Actions, Gradle, Make।
- परीक्षण / गुणवत्ता आश्वासन: HelixQA, म्यूटेशन गेट्स के साथ चुनौती संरचना, `go test -race`, दृश्य-प्रतिगमन उपकरण, ADB डिवाइस परीक्षण, SonarQube, सुरक्षा स्कैनिंग (semgrep/gosec/trivy/snyk/gitleaks/nancy), TLA+ मॉडल जाँच।

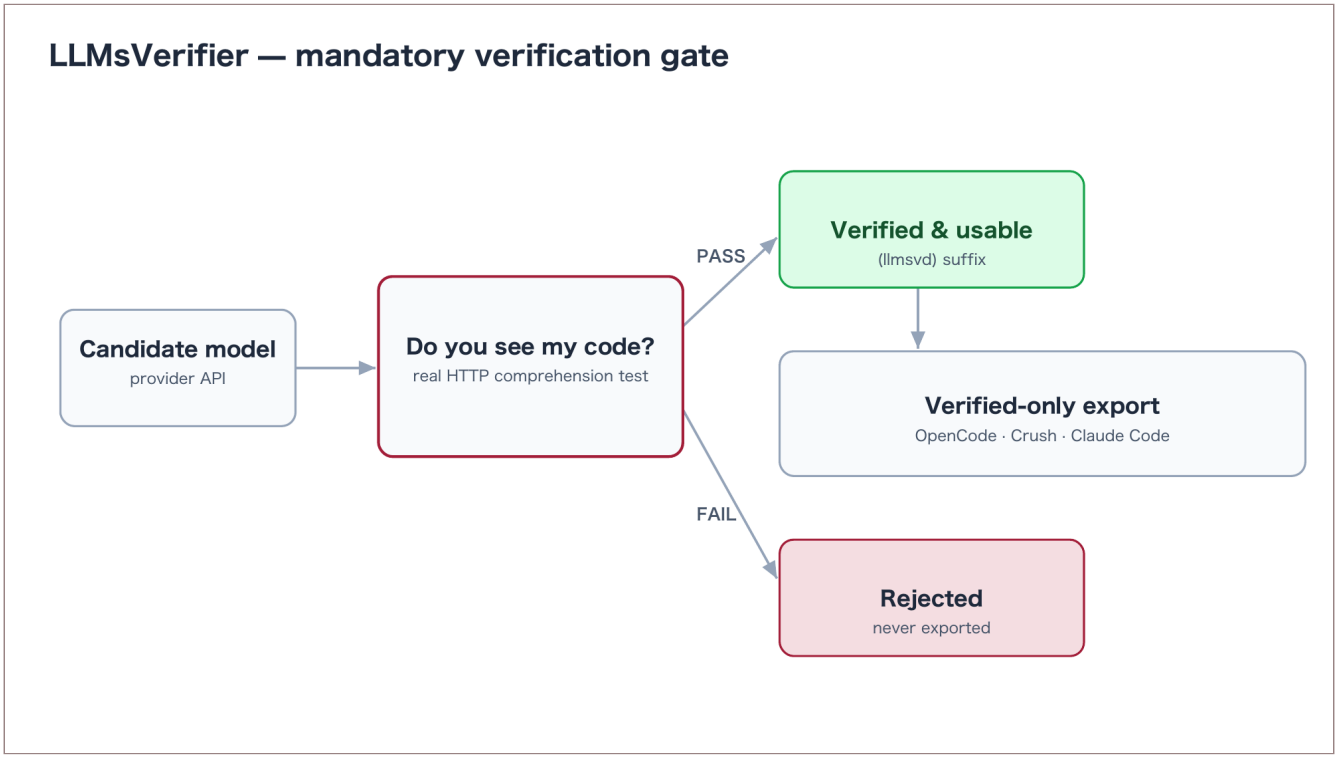
चयनित वास्तुकला



HelixAgent — एक समूह LLM सेवा जहाँ प्रदाता चार-चरणीय प्रोटोकॉल के तहत बहस करते हैं और जिस उत्तर पर सहमति बनती है, उसे LLMsVerifier द्वारा स्कोर करके भेजते हैं।



HelixCluster — AI कंप्यूटिंग के लिए एक वितरित ऑपरेटिंग सिस्टम, डेटासेंटर जीपीयू से लेकर एज हैंडहेल्ड डिवाइस तक, एक ही नियंत्रण प्लेन के तहत।



LLMsVerifier — सत्यापन, निगरानी, अनुकूलन: LLM/प्रदाता/सत्यापन मेटाडेटा के लिए एकल स्रोत, जिसे अनिवार्य मॉडल-समझ जाँच द्वारा नियंत्रित किया जाता है।