

VASIC DIGITAL · EVIDENCE-BASED PORTFOLIO



Milos Vasic ✓

**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. 概要 — 全体像をまず把握する

これはvasic.digitalとmilosvasic.ruの両者が使用する単一の統合ポートフォリオである。散発的なサイドプロジェクトの寄せ集めではなく、意図的に設計された艦隊を描写するものだ。**大規模なプロダクトアプリケーションが、数十の小規模で疎結合、独立テスト済みのモジュールの上に構築され、全体が共通のエンジニアリングConstitutionによって統制され、ブラフ排除のためのQA規律によって検証されている。**この構造こそが差別化要因である。ほとんどのポートフォリオは「作られたもののリスト」に過ぎないが、これは「実証済みの再利用可能な部品から組み立てられたシステム」であり、すべての部品が同じ非妥協的なルールに従い、すべての謳われた機能には裏付けとなるエビデンスが存在する。主な言語はGoで、Kotlin/KMP、TypeScript/React、Python、Swift、Shellはそれぞれ最適な用途に応じて使用される。Goは高スループットのサービスやライブラリに、Kotlinはプロビジョニングやクロスプラットフォームのモバイル開発に、TypeScriptは型付きフロントエンドに、PythonはAI/MLの接着剤として活用される。

この作品群を一貫させているのは、規律が理想論ではなく機械的に運用されている点だ。共通のConstitutionはGitサブモジュールとして提供され、140以上のリポジトリにわたる艦隊全体に継承されるため、ルールの変更は一箇所ですべてに波及する。また、ブラフ排除のためのQAレイヤーは、実行時の証拠がなければ合格を記録しない。Helixファミリー、ユーティリティ艦隊、Server Factoryツールチェーンは、いずれも「一度作ってどこでも再利用し、完了と称する前に動作を証明する」という一つの思想の表れである。

以下の内容は優先順に提示する：

1. **ガバナンスとQAの柱** — HelixConstitution、HelixQA（これらが他のすべての信頼性を支える）
2. **Helixプロダクトファミリー** — AI開発ライフサイクル（HelixTrack優先）
3. **LLMインフラストラクチャ** — プロバイダー抽象化、オーケストレーション、検証
4. **vasic-digitalユーティリティ** — プロダクトレベルのスタンドアロンツール
5. **Server Factory** — インフラ自動化の系譜（優先度最下位）

統一的なテーゼ：「**実際のユーザーが利用可能で、それを証明するエビデンスが記録されて初めて、機能は完了したと言える。**」

2. ガバナンスとQAの柱

これら二つが最初に来るのは、ポートフォリオの他のすべてがその信頼性をこの二つから借りているからだ。合わせて「信じてくれ、動くから」を監査可能な事実へと変える。Constitutionがルールを規定し、HelixQAがその遵守を証明する。

- **HelixConstitution** — プロジェクトに依存しない普遍的なエンジニアリング規約集。Gitサブモジュールとして提供され、140以上のリポジトリにわたる艦隊全体に継承される。ブラフ排除のためのエビデンスゲート、偽陽性耐性、データ/ホストの安全性、カバレッジ規律。拡張は許すが弱体化は許さな

い継承。艦隊全体で必要条項をgrepする伝播ゲート。すべてのゲートには、それが偽物でないことを証明するミューテーションテストが付随する。

- **HelixQA** — ブラフ排除のためのQAオーケストレーション（Go）。記述式のYAMLテストバンクに加え、Android、Android TV、Web、デスクトップにまたがる完全自律型のLLMおよびコンピュータビジョンQAセッション。キャプチャされたエビデンス（スクリーンショット、logcat、動画、スタックトレース）がなければ合格は記録されない。Constitutionで義務付けられたQAテストタイプ（§ 11.4.169）。

3. Helix プロダクトファミリー

Helix ラインは当社のフラッグシップ製品群であり、AI 開発ライフサイクル全体を網羅する相互連携型のプロダクトファミリーです。計画立案から仕様策定、構築、メモリ管理、翻訳、そして納品に至るまで、それぞれが独立した製品として機能しますが、設計思想としては統合を前提としています。共通のガバナンス、再利用可能なモジュール、そして全ての製品を支える統一されたエビデンス管理体制がその基盤です。

- **HelixTrack** — フリーな世界向けの JIRA 代替製品。Helix-Track ラインのフラッグシップ。
- **HelixAgent** — アンサンブル型 LLM サービス。複数のモデルが議論し、合意した回答を出力。検証ベースのプロバイダー選定機能付き。
- **HelixCode** — エンタープライズグレードの分散型 AI 開発プラットフォーム。SSH 管理下のワーカーに作業を分散し、自動チェックポイント／ロールバック機能を搭載。REST、CLI、TUI、MCP インターフェース対応。
- **HelixCluster** — データセンターの GPU からエッジ端末まで、AI コンピューティングを統合管理する分散オペレーティングシステム。単一のコントロールプレーンで運用。
- **HelixBuilder** — AI を活用したアプリケーション構築パイプライン。カテゴリーごとに段階的に開発。
- **HelixSkills** — ガバナンスと憲章に基づく CLI AI エージェント向けスキルシステム（スキル、MCP ツールサーバー、Claude Code プラグイン）。
- **HelixSpecifier** — 仕様駆動型開発。作業規模に応じてセレモニーをスケーリング。
- **HelixMemory** — AI エージェント向けの統合メモリブレイン。4つの最先端エンジンを融合。
- **HelixTranslate** — 検証済みモデルによる書籍翻訳。設計上、不正確なフォールバックは一切発生しない。
- **HelixTerminator** — ゼロトラスト端末プラットフォーム。全ての SSH セッションはセキュアに保護され、共有可能かつ AI の支援を受けられる。
- **HelixGitpx** — 十数のホストにまたがるフェデレーテッド Git。単一の真実ソースを全ての環境にミラーリング。
- **HelixOTA** — 汎用的で疎結合な OTA アップデート。設計上、ブリック（文鎮化）のリスクはゼロ。
- **HelixPlay** — 任意の GPU マシンをクラウドゲーミング端末に変えるソリューション。
- **Helix-Flow** — Helix プラットフォーム向け推論製品。未検証／ソース依存：現時点の公開リポジトリには README が1行のみ。正式なドキュメントが整い次第、製品レベルでの詳細を公開予定。

4. LLM インフラストラクチャ

プロダクト群の基盤となるのが、プロバイダーに依存せず信頼性を確保するためのレイヤーです。数十の LLM プロバイダーを統合する単一インターフェース、ヘッドレスコーディングエージェント向けの単一コントロールプレーン、そして検証に基づく唯一の真実ソースがここにあります。このレイヤーにより、上位レイヤーはモデルの切り替え、プロバイダー障害への耐性、そしてタスク理解を証明できない LLM の排除が可能になります。

- **HelixLLM** — 単一バイナリ、6 つの動作モード。OpenAI および Anthropic 互換の推論を HTTP/3、ローカル llama.cpp、スコアベースのフォールバックチェーン、RAG パイプライン、ReAct エージェントで実行。
- **LLMProvider** — 単一インターフェース、43 のプロバイダー対応。サーキットブレーカー、リトライ、ヘルスチェック機能を内蔵。
- **LLMOrchestrator** — 全てのヘッドレス CLI コーディングエージェント（OpenCode、Claude Code、Gemini、Junie、Qwen Code）向けの単一コントロールプレーン。
- **LLMsVerifier** — 検証、監視、最適化を一元管理。LLM／プロバイダー／検証メタデータの唯一の真実ソース。モデル理解度のゲートチェックは必須。

5. vasic-digitalユーティリティ

それぞれが単体で難題を解決するプロダクトレベルのツール群であり、偶然にも再利用可能モジュール群の実戦テストの場ともなっている。レジリエントなマルチプロトコルメディアシステムから、Markdownを動画コースに変換するパイプライン、コンテンツハッシュ化されたドキュメント／データベース同期エンジンまで多岐にわたる。成熟度については率直に示されており、その表示は隠されることなく維持されている。

- **Catalogizer** — マルチプロトコル（SMB／FTP／NFS／WebDAV／ローカル）対応、暗号化、セルフホスティング可能なメディアコレクション管理システム。Go／Gin API + React UI。オフライン耐性のあるレジリエントな監視機能。21の `digital.vasic.*` サブモジュールを基盤に構築。
- **Courses-Creator** — Markdownから動画コースへのパイプライン。マルチLLMによるエンリッチメント、TTS（Bark／SpeechT5）、デスクトップ／モバイル／ウェブプレイヤー。APIキーなしモードでも優雅に動作。
- **VisionEngine** — 従来のコンピュータビジョンとマルチプロバイダーLLMビジョンを融合した、分離型Goツールキット。BFS + DOT／JSON／Mermaid形式でのナビゲーショングラフエクスポート。OpenCVはビルドタグで制御。
- **DocProcessor** — ドキュメントから機能マップを生成し、検証カバレッジを追跡。LLMまたはヒューリスティック／オフライン抽出に対応。Apache-2.0ライセンス。
- **Docs Chain** — コンテンツハッシュ化された双方向アトミックドキュメント／データベース同期（DAG上でのSalsaスタイルのインクリメンタル再計算）。フェーズ1〜5：GREEN／フェーズ6〜7：計画中。
- **Herald** — 自然言語による3層_intent解決（コマンド→LLM→明確化）を備えた、信頼性の高いマルチチャネル通知システム。Docs Chainの最初の利用者。
- **task_bridge** — 分離型双方向タスク／ボード同期（SQLite SSoT ☐ ドキュメント ☐ ClickUp）。P1 スキャフォールド段階 — 同期ロジックは未実装。

- **Vasic Digital 再利用可能モジュールスイート** — [digital.vasic.*](#) 「標準ライブラリ」：インフラ基盤、AIビルディングブロック、防御的LLMガードレール、およびKotlin Multiplatformミラー。複数の組織リポジトリはScaffold/WIP段階 — 表示あり、未リリース。

6. Server Factory（インフラストラクチャ自動化の系譜 — 意図的に最後尾）

品質ではなく、意図的に最後に配置されている。Server FactoryツールチェーンはAIラインよりも前に誕生し、「一度作って、どこでも再利用」という哲学が初めて形になった場所を示している。その旗艦製品である、JSONで記述しどこでもプロビジョニング可能なメールサーバーは、成熟し十分にテストされたプロダクトである。サポートツール群も、装飾されることなく、それぞれの成熟度に応じて提示されている。

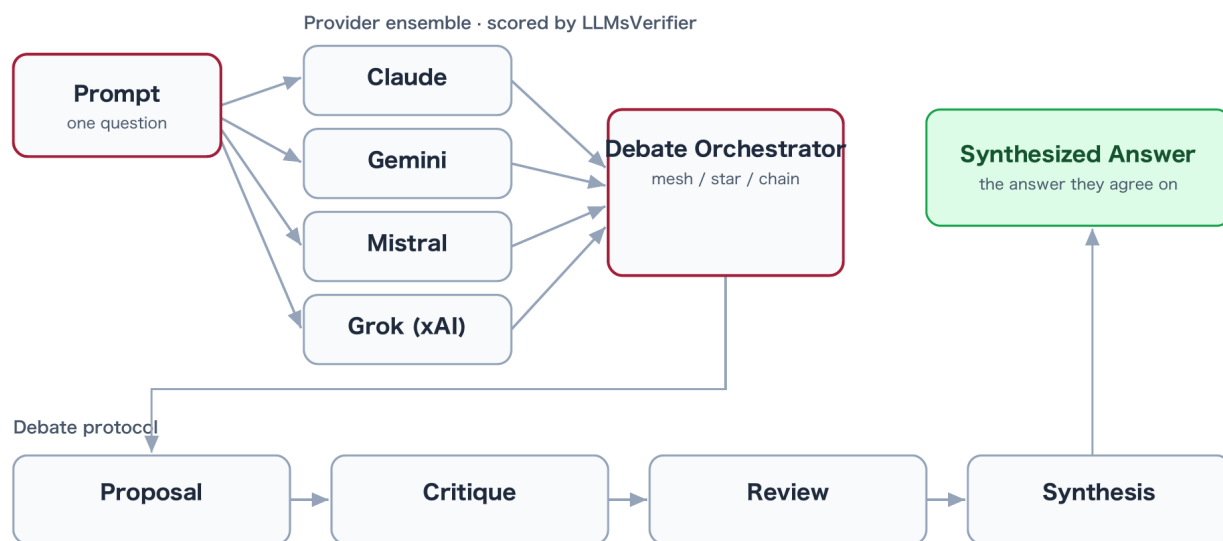
- **Mail Server Factory** — 宣言型JSONから、12種類の接続方式と25のLinuxディストリビューションに対応した、完全プロビジョニング済みDocker化メールサーバーを構築。エンタープライズセキュリティ対応。439件のテスト合格とSonarQubeゲートクリアを報告。Server-Factory組織の旗艦製品。
- **Server Factory コアフレームワーク** — すべてのファクトリーが基盤とする共有Kotlinエンジン。
- **Qemu-Utills** — アーティファクトとして管理されるQEMU VMイメージ：ダウンロード／キャッシュ／実行、圧縮／公開、ブリッジ／TAPネットワークング、ISOインストール。Linux + macOS対応。
- **Parallels-Utills** — Parallels (macOS) VMイメージの圧縮、公開、および設定ファイルによる簡易取得。
- **Server Factory — 追加コンポーネント** — サービスファクトリー (Web/SonarQube/キャッシングプロキシ)、定義パック、ユーティリティ。サービスファクトリーはブレースホルダーとして文書化済み — 未検証／初期段階。

7. テクノロジーインデックス（エビデンスベース）

- **言語**：Go（主流）、Kotlin & Kotlin Multiplatform、TypeScript、Python、Swift、Shell；PL/pgSQL；TLA+（helix_clusterにおける正式仕様）
- **AI / LLM**：40以上のプロバイダーアクセス、MCP、RAG、vectorデータベース & embeddings、プランニング（HiPlan/MCTS/Tree-of-Thoughts）、LLMOps、ベンチマーキング（SWE-bench/HumanEval/MMLU）、TTS（Bark/SpeechT5）、コンピュータビジョン + LLM-vision、ガードレール/レッドチーム
- **バックエンド**：Gin、gRPC + Protobuf、HTTP/3 (QUIC)、WebSocket、Angular、React、Kafka/RabbitMQ
- **データ**：PostgreSQL、SQLite、SQLCipher、Redis、Neo4j、ClickHouse、MinIO/S3/GCS/Azure
- **インフラ / DevOps**：Docker & Compose、Kubernetes + Helm、Prometheus + Grafana、OpenTelemetry、QEMU/Libvirt/Parallels；GitHub Actions、Gradle、Make
- **テスト / QA**：HelixQA、ミューテーションゲート付きチャレンジハーネス、`go test -race`、ビジュアルリグレッションツール、ADBデバイステスト、SonarQube、セキュリティスキャン（semgrep/gosec/trivy/snyk/gitleaks/nancy）、TLA+モデルチェック

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)

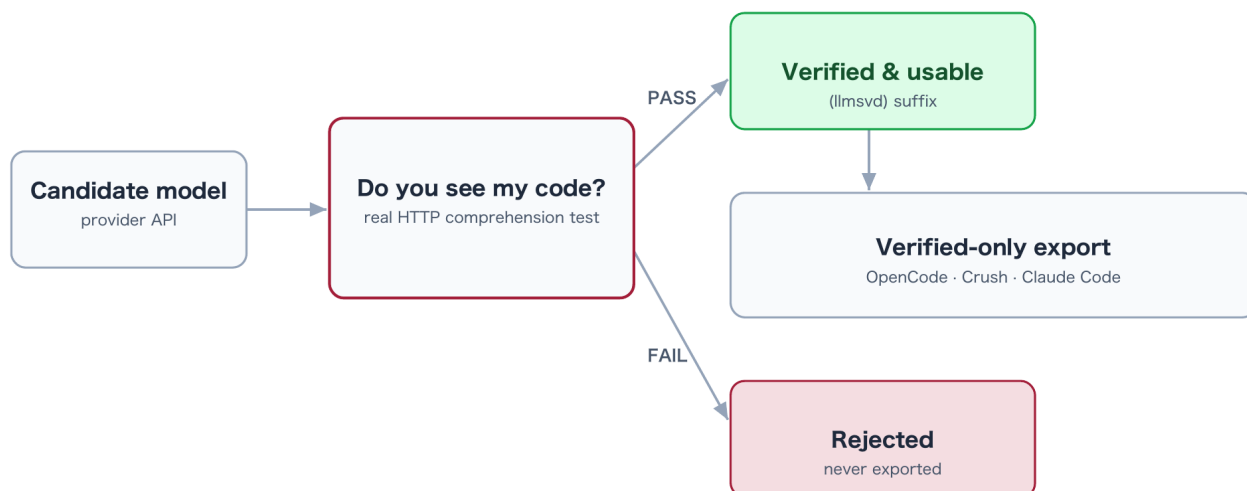
heterogeneous
nodes T1–T8

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.