portfolio

A unified, evidence-based portfolio of the Helix family, vasic-digital utilities, and the Server Factory toolchain.

المحفظة

١. نظرة عامة — الصورة الكاملة أولاً

هذه محفظة موحدة واحدة يستخدمها كلّ من milosvasic.ru. وvasic.digital وهي لا تصف مجموعة متفرقة من المشاريع الجانبية، بل مصمماً بعناية **أسطولاً** مبنية على **تطبيقات منتجات كبيرة**: حيث يحكم الكل إطار هندسي مشترك، **عشرات الوحدات الصغيرة المستقلة المختبرة بشكل منفصل**، ويتحقق منه نظام **ضمان الجودة المضاد للادعاء**. هذا الهيكل هو ما يميزها **Constitution**. فمعظم المحافظ هي مجرد قائمة بالأشياء التي بُنيت؛ أما هذه فهي نظام يُجمع فيه كل منتج من أجزاء مجربة وقابلة لإعادة الاستخدام، وكل جزء يخضع لنفس القواعد غير القابلة للتفاوض، وكل قمرة معلن عنها مدعومة بأدلة ملموسة. اللغة السائدة هنا هي **Go**، مع استخدام TypeScript/ وKotlin/KMP للخدمات Go — حيثما تناسب كل منها بشكل حقيقي Shell وSwift وPython وReact للواجهات TypeScript، للتزويد والتطبيقات متعددة المنصات Kotlin، والمكتبات عالية الإنتاجية والأمامية المكتوبة بأنواع البيانات. Python للصق بين AI وML.

ما يجعل هذا العمل متماسكاً هو أن الانضباط هنا ميكانيكي وليس طموحاً. فالإطار المشترك **Constitution** يُنشر كوحدة فرعية في Git ١٤٠ مستودعاً، ويُورث عبر أسطول يضم أكثر من بحيث تنتشر أي تغيير في القاعدة في كل مكان؛ كما يرفض نظام ضمان الجودة المضاد للادعاء تسجيل نجاح أي اختبار دون دليل تشغيلي. عائلة Helix والأسطول الخدمي وأدوات Server Factory هي ثلاث تجليات لفكرة واحدة: ابنِ مرة واحدة، أعد الاستخدام في كل مكان، وأثبت أنه يعمل قبل أن تعتبره مكتملاً.

كل ما يلي يعرض بحسب الأولوية:

١. **أركان الحوكمة وضمان الجودة** — HelixConstitution وHelixQA (الانضباط الذي يجعل الباقي موثوقاً). ٢. **عائلة منتجات Helix** — دورة حياة تطوير AI (HelixTrack أولاً). ٣. **أدوات vasic-digital** — تجريد المزودين والتنسيق والتحقق. ٤ — **البنية التحتية LLM** — إرث أتمتة البنية التحتية (مرتبة أخيراً). ٥ — **المستقلة Server Factory** — أدوات مستقلة على مستوى المنتجات.

الأطروحة الموحدة: **لا يُعتبر الميزة مكتملة إلا عندما يتمكن مستخدم حقيقي من استخدامها، وهناك دليل ملموس يثبت ذلك.**

---

٢. أركان الحوكمة وضمان الجودة

يأتي هذان العنصران أولاً لأن كل ما سواهما في هذه المحفظة يستمد مصداقيته منهما. فهما معاً يحولان عبارة "ثق بي، إنه يعمل" إلى حقيقة قابلة للتدقيق — حيث يُرمّز الإطار **Constitution** القواعد، أنها اتُبعت **HelixQA** بينما يثبت نظام.

- **HelixConstitution** — كتاب قواعد هندسية شامل وغير مرتبط بمشروع محدد، يُنشر ويُورث عبر أسطول يضم أكثر من ١٤٠ مستودعاً. بوابات أدلة مضادة Git كوحدة فرعية في للادعاء، مناعة ضد الإيجابيات الكاذبة، سلامة البيانات والمضيف، انضباط التغطية؛ وراثة قابلة للتوسيع دون إضعاف؛ بوابات انتشار تبحث عن البنود المطلوبة عبر الأسطول بأكمله؛ وكل بوابة مقترنة باختبار تحور يثبت أنها ليست مجرد خدعة.
- **HelixQA** — تنسيق ضمان الجودة المضاد للادعاء (Go). بنوك اختبارات مكتوبة بلغة YAML والرؤية الحاسوبية LLM بالإضافة إلى جلسات ضمان جودة مستقلة بالكامل تعتمد على والويب وسطح المكتب؛ لا يُسجل نجاح أي اختبار دون Android TV وAndroid عبر أنظمة نوع الاختبار (مقاطع فيديو، تتبع المكدس، logcat لقطات شاشة، سجلات) أدلة ملموسة الإلزامي لضمان الجودة وفقاً للإطار **Constitution** (§١١.٤.١٦٩).

٣. عائلة منتجات **Helix**

بأكملها، بدءاً من AI الرائدة عائلة متكاملة من المنتجات تغطي دورة حياة تطوير Helix تعدّ سلسلة التخطيط والمواصفات مروراً بالبناء والذاكرة والترجمة وصولاً إلى التسليم. كل منتج منها قائم بذاته، إلا أن القصد من التصميم هو تكاملها معاً: نفس الحوكمة، نفس الوحدات القابلة لإعادة الاستخدام، ونفس الانضباط القائم على الأدلة تحت مظلة واحدة.

- **HelixTrack** — بديل JIRA للعالم الحر؛ الرائد في خط منتجات Helix-Track.
- **HelixAgent** — خدمة LLM التجميعية: تتناقش نماذج متعددة وتصدر الإجابة التي تتفق عليها، مع اختيار المزودين بناءً على التحقق.
- **HelixCode** — منصة تطوير AI موزعة على مستوى المؤسسات؛ تقسم المهام بين العاملين TUI وCLI وREST مع نقاط تفتيش واستعادة تلقائية؛ واجهات SSH المدارين بواسطة MCP.
- **HelixCluster** — نظام تشغيل موزع لحوسبة AI، من وحدات معالجة الرسوميات في مراكز البيانات إلى الأجهزة المحمولة الطرفية، تحت لوحة تحكم واحدة.
- **HelixBuilder** — خط إنتاج مدعوم بـ AI لبناء التطبيقات، فئة تلو الأخرى.
- **HelixSkills** — نظام مهارات محكوم ومستند إلى دستور خاص بوكلاء AI من CLI (المهارات، خوادم الأدوات، MCP، إضافات Claude Code).
- **HelixSpecifier** — التطوير القائم على المواصفات الذي يضبط طقوسه وفقاً لحجم العمل.
- **HelixMemory** — دماغ ذاكرة واحد لوكلاء AI، يجمع بين أربعة محركات رائدة في فئتها.
- **HelixTranslate** — مصممة لتكون ترجمة كتب تعتمد على النماذج المُتحقق منها؛ شفافة، ولا تلجأ أبداً إلى التراجع الصامت.
- **HelixTerminator** — منصة طرفية بلا ثقة: كل جلسة SSH مؤمنة ومشتركة ومساعَدة بواسطة AI.
- **HelixGitpx** — Git موحد عبر عشرات المضيفين؛ مصدر واحد للحقيقة، متكرر في كل مكان.
- **HelixOTA** — تحديثات عبر الهواء عالمية ومستقلة؛ مصممة لتكون بلا أعطال.
- **HelixPlay** — حوّل أي جهاز GPU إلى جهاز سحابي للألعاب خاص بك.
- **Helix-Flow** — منتج استدلال لمنصة Helix. غير مُتحقق منه / معلق بانتظار المصدر: بسطر واحد فقط؛ سيتم تقديمه بالعمق *README* مستودعه العام يحتوي حالياً على ملف المنتج عند توفر الوثائق الحقيقية.

٤. البنية التحتية لـ LLM

تحت المنتجات تكمن الطبقة الأساسية التي تجعلها مستقلة عن المزودين وموثوقة: واجهة واحدة فوق ولوحة تحكم واحدة لوكلاء الترميز غير المرئيين، ومصدر واحد للحقيقة القائم على LLM، عشرات مزودي التحقق. هذه هي الطبقة التي تسمح لكل ما فوقها بتبديل النماذج، والبقاء على قيد العمل أثناء انقطاعات المزودين، ورفض الثقة بأي LLM لا يستطيع إثبات فهمه للمهمة.

- **HelixLLM** — Anthropic وOpenAI ثنائي واحد بستة أوضاع: استدلال متوافق مع وكلاء، RAG محلي، سلسلة تراجع مُقيّمة، خط أنابيب llama.cpp، HTTP/3 عبر ReAct.
- **LLMProvider** — واجهة واحدة لـ٤٣ مزوداً، مع قواطع دوائر وإعادة محاولة وآليات صحية مدمجة.
- **LLMOrchestrator** — لوحة تحكم واحدة لكل وكلاء الترميز غير المرئيين من CLI (OpenCode، Claude Code، Gemini، Junie، Qwen Code).
- **LLMsVerifier** — LLM تحقق وراقب وحسّن: المصدر الوحيد للحقيقة الخاصة ببيانات المزودين والتحقق، مع بوابة إلزامية لفهم النموذج.

المحتوى

٥. أدوات فاسيك الرقمية

أدوات مستقلة ذات مستوى إنتاجي، كل منها يحل مشكلة معقدة بذاتها — وهي، ليس من قبيل الصدفة، تختبر أسطول الوحدات القابلة لإعادة الاستخدام بكثافة. تتراوح بين نظام وسائط متعدد البروتوكولات مرن إلى خط إنتاج دورات تعليمية من ماركداون إلى فيديو، وصولاً إلى محرك مزامنة الوثائق/قواعد البيانات المحتوى-المهشّم؛ بعضها صريح بشأن مستوى نضجه، وهذه العلامات تُحفظ ولا تُدفن.

- **Catalogizer** — إدارة مجموعات الوسائط متعددة البروتوكولات (SMB/FTP/NFS/WebDAV/واجهة: API + React مع Go/Gin محلي)، مشفرة وقابلة للاستضافة الذاتية؛ مراقبة مرنة تتحمل الانقطاع؛ مبنية على ٢١ وحدة فرعية من `digital.vasic.*`.
- **Courses-Creator** — خط إنتاج دورات تعليمية من ماركداون إلى فيديو؛ إثراء متعدد LLM، TTS (Bark/SpeechT5)، ومشغلات سطح المكتب/الهواتف/الويب؛ وضع تشغيل سلس بدون مفتاح API.
- **VisionEngine** — مجموعة أدوات Go الكلاسيكية المفككة التي تدمج الرؤية الحاسوبية متعددة المزودين؛ رسوم بيانية للملاحة مع LLM وتصدير إلى BFS DOT/JSON/Mermaid ؛ بوابة بناء محددة بالعلامات OpenCV.
- **DocProcessor** — خريطة الميزات من الوثائق مع تتبع تغطية التحقق؛ استخراج LLM أو استخراج قائم على الاستدلال/غير متصل؛ ترخيص Apache-2.0.
- **Docs Chain** — مزامنة وثائق/قواعد بيانات ثنائية الاتجاه ذرية المحتوى-المهشّم (إعادة حساب تدريجية على نمط Salsa عبر DAG). المراحل ١–٥ مكتملة؛ المراحل ٦–٧ مخطط لها.
- **Herald** — إشعارات متعددة القنوات موثوقة مع تحليل نية ثلاثي المستويات بلغة طبيعية (أمر ← LLM ← توضيح)؛ أول مستهلك لـDocs Chain.
- **task_bridge** — مزامنة مهام/لوحات ثنائية الاتجاه مفككة (مصدر وحيد للحقيقة SQLite ↔ الوثائق ↔ ClickUp). هيكل المرحلة الأولى — منطق المزامنة لم يُنفذ بعد.
- **Vasic Digital مجموعة الوحدات القابلة لإعادة الاستخدام** — المكتبة القياسية" لـ LLM، وحاجز دفاعي لـ AI، وأساسيات البنية التحتية، لبنات بناء: `digital.vasic.*`. عدة مستودعات مؤسسية هي في مرحلة السقالات/قيد التطوير — مُعلّمة، وغير منشورة بالإضافة إلى مرآة Kotlin Multiplatform.

٦. Server Factory (إرث أتمتة البنية التحتية — مُصنّف أخيراً)

سبقت خط AI Server Factory مُصنّف أخيراً عن قصد، وليس لجودة الأدوات: سلسلة أدوات وتُظهر أين تشكلت فلسفة "ابنِ مرةً، أعد الاستخدام في كل مكان" لأول مرة. منتجها الرائد — خادم بريد وتُنشره في أي مكان — هو منتج ناضج ومُختبر جيداً؛ أما بقية الأدوات المصاحبة JSON تُصفه في فتُقدّم بمستويات نضجها الحقيقية دون تجميل.

- **Mail Server Factory** — خادم بريد Docker مُعد بالكامل عبر JSON التقريرية ؛ أمان مؤسسي؛ يُبلغ عن اجتياز ٤٣٩ اختباراً وبوابةLinux عبر ١٢ نوع اتصال و٢٥ توزيعة Server-Factory. المنتج الرائد لمنظمة SonarQube نظيفة.
- **Server Factory الإطار الأساسي** — محرك Kotlin المشترك الذي تبنى عليه كل المصانع.
- **Qemu-Utils** — صور آلات افتراضية QEMU تُدار كقطع أثرية: تحميل/تخزين مؤقت/تشغيل، ضغط/نشر، شبكات جسرية/TAP، تثبيت عبر ISO؛ Linux وماك.
- **Parallels-Utils** — ضغط صور الآلات الافتراضية Parallels (ماك)، نشر واسترجاع عبر ملفات إعدادات بسيطة.
- **Server Factory — المكونات الإضافية** — مصانع خدمات (ويب/SonarQube/ — مصانع الخدمات موثقة بشكل مؤقت ،وكيل التخزين المؤقت)، حزم التعريفات، وأدوات مساعدة غير مُتحقق منها/في مراحل مبكرة.

المحتوى

---

٧. مؤشر التكنولوجيا (قائم على الأدلة)

- **اللغات:** Go (السائدة)، Kotlin وKotlin Multiplatform، TypeScript، Python، Swift، Shell؛ PL/pgSQL؛ TLA+ (المواصفات الرسمية في helix_cluster).
- **AI / LLM:** وصول إلى أكثر من ٤٠ مزودًا، MCP، RAG، قواعد البيانات vector وembeddings، التخطيط (HiPlan/MCTS/شجرة الأفكار)، LLMOps، المقارنة المعيارية (SWE-bench/HumanEval/MMLU)، TTS (Bark/SpeechT5)، الرؤية الحاسوبية + LLM-vision، الضوابط/الفريق الأحمر.
- **الواجهة الخلفية:** Gin، gRPC + Protobuf، HTTP/3 (QUIC)، WebSockets، Angular، React، Kafka/RabbitMQ.
- **البيانات:** PostgreSQL، SQLite، SQLCipher، Redis، Neo4j، ClickHouse، MinIO/S3/GCS/Azure.
- **البنية التحتية/DevOps:** Docker وCompose، Kubernetes + Helm، Prometheus + Grafana، OpenTelemetry، QEMU/Libvirt/Parallels؛ GitHub Actions، Gradle، Make.
- **الاختبار/ضمان الجودة:** HelixQA، أطر التحدي مع بوابات التحور، `go test -race`، فحص الأمان، SonarQube، ADB أدوات اختبار الانحدار البصري، اختبار الأجهزة عبر (semgrep/gosec/trivy/snyk/gitleaks/nancy)، التحقق من النماذج عبر TLA+.