



Vasic Digital

AI yerel yazılım mühendisliği · Kanıta dayalı portföy

Helix ürün ailesi, vasic-digital hizmet filosu ve Server Factory araç zinciri — güvenilir AI geliştirme sistemleri, ortak bir mühendislik Constitution üzerinde inşa edildi.

140+

DEPO FILOSU

43

LLM SAĞLAYICILARI

25

SAĞLANAN LINUX
DAĞITIMLARI

439

BAŞARILI TESTLER ·
MAIL SERVER
FACTORY

1. Genel Bakış — Önce Bütün Resim

Bu, hem vasic.digital hem de milosvasic.ru tarafından kullanılan tek ve bütünleşik bir portföydür. Dağınık yan projelerden değil, **kasıtlı olarak tasarlanmış bir filodan** bahsediyor: **Düzine dolusu küçük, bağımsız, ayrı ayrı test edilmiş modüllerin** üzerinde yükselen **büyük ürün uygulamaları**; tüm bunlar ortak bir mühendislik **Constitution** tarafından yönetilir ve **blöf karşıtı bir QA** disipliniyle doğrulanır. Bu yapı, ayırt edici özelliktir. Çoğu portföy inşa edilmiş şeylerin bir listesidir; bu ise her ürünün kanıtlanmış, yeniden kullanılabilir parçalardan bir araya getirildiği, her parçanın aynı tavizsiz kurallara tabi tutulduğu ve her ilan edilen yeteneğin arkasında somut kanıtların bulunduğu bir sistemdir. Ana dil **Go** olup, Kotlin/KMP, TypeScript/React, Python, Swift ve Shell, her biri gerçekten en uygun olduğu yerde kullanılır — Go yüksek verimli hizmetler ve kütüphaneler için, Kotlin tedarik ve çok platformlu mobil için, TypeScript tip güvenli ön yüzler için, Python ise AI/ML entegrasyonu için tercih edilir.

Çalışmalar bütünlüğünü sağlayan şey, disiplinin heveskârlıktan değil, mekanik bir yaklaşımla uygulanmasıdır. Ortak bir **Constitution**, Git alt modülü olarak yayınlanır ve 140'tan fazla depodan oluşan bir filoya miras bırakılır; böylece tek bir kural değişikliği her yere yayılır. Blöf karşıtı QA katmanı ise çalışma zamanı kanıtı olmadan geçer notu kaydetmeyi reddeder. **Helix** ailesi, yardımcı filo ve **Server Factory** araç zinciri, tek bir fikrin üç farklı ifadesidir: Bir kez inşa et, her yerde kullan ve bitti diyebilmek için çalıştığını kanıtla.

Aşağıda her şey öncelik sırasına göre sunulmuştur:

1. **Yönetişim ve QA temelleri** — HelixConstitution, HelixQA (gerisini güvenilir kılan disiplin).
2. **Helix ürün ailesi** — AI geliştirme yaşam döngüsü (önce HelixTrack).
3. **LLM altyapısı** — sağlayıcı soyutlaması, orkestrasyon, doğrulama.
4. **vasic-digital yardımcı araçları** — ürün kalitesinde bağımsız araçlar.
5. **Server Factory** — altyapı otomasyonu mirası (sıralamada son sırada).

Birleştirici tez: **Bir özellik, gerçek bir kullanıcı tarafından kullanılabilirdiğinde ve bunu kanıtlayan somut bir delil olduğunda tamamlanmış sayılır.**

2. Yönetişim ve QA Temelleri

Bunlar ilk sırada yer alır çünkü portföydeki diğer her şey güvenilirliğini onlardan alır. Birlikte, “Bana güven, çalışıyor” ifadesini denetlenebilir bir gerçeğe dönüştürürler — **Constitution** kuralları kodlar, **HelixQA** ise bu kurallara uyulduğunu kanıtlar.

- **HelixConstitution** — Projenin bağımsız, evrensel bir mühendislik kural kitabı. Git alt modülü olarak yayınlanır ve 140'tan fazla depodan oluşan bir filoya miras bırakılır. Blöf karşıtı kanıt kapıları, yanlış pozitiflere karşı bağıışıklık, veri/konak güvenliği, kapsam disiplini; genişletilebilir ama asla zayıflatılamaz

miras yapısı; filonun tamamında gerekli maddeleri tarayan yayılım kapıları; her kapının, sahte olmadığını kanıtlayan bir mutasyon testi ile eşleştirilmesi.

- **HelixQA** — Blöf karşıtı QA orkestrasyonu (Go). Yazılı **YAML** test bankaları ve Android, Android TV, Web ve Masaüstü platformlarında tamamen otonom **LLM** ve bilgisayarlı görü QA oturumları; kanıt (ekran görüntüleri, logcat, video, yığın izleri) olmadan GEÇER notu yok. **Constitution**'un zorunlu kıldığı QA test türü (§11.4.169).

3. Helix Ürün Ailesi

Helix serisi, amiral gemisi konumunda: Planlama ve şartname belirlemeden inşa etme, bellek yönetimi, çeviri ve teslimata kadar AI geliştirme yaşam döngüsünün tamamını kapsayan, birbiriyle uyumlu bir ürün ailesi. Her biri kendi başına gerçek bir ürün olsa da, tasarım felsefesi bir araya gelmelerini öngörüyor — aynı yönetim, aynı yeniden kullanılabilir modüller, hepsinin altında aynı kanıt disiplini.

- **HelixTrack** — özgür dünyaya yönelik bir JIRA alternatifi; Helix-Track hattının amiral gemisi.
- **HelixAgent** — topluluk tabanlı LLM hizmeti: Birden fazla model tartışır ve üzerinde uzlaştıkları yanıt gönderir, doğrulama temelli sağlayıcı seçimiyle.
- **HelixCode** — kurumsal düzeyde dağıtık AI geliştirme platformu; işleri SSH tarafından yönetilen çalışanlara otomatik kontrol noktası/geri alma ile dağıtır; REST, CLI, TUI, MCP arayüzleri.
- **HelixCluster** — veri merkezi GPU'larından kenar cihazlara kadar AI hesaplamaları için dağıtık bir işletim sistemi; tek bir kontrol düzlemi altında.
- **HelixBuilder** — uygulamaları kategori bazında inşa eden, AI destekli bir boru hattı.
- **HelixSkills** — CLI AI ajanları için anayasal temelli, yönetişimli bir beceri sistemi (beceriler, MCP araç sunucuları, Claude Code eklentileri).
- **HelixSpecifier** — işin gerektirdiği ölçüde tören ölçeklendiren, şartname odaklı geliştirme.
- **HelixMemory** — AI ajanları için tek bellek beyini; dört en iyi sınıf motoru bir araya getirir.
- **HelixTranslate** — doğrulanmış model tabanlı kitap çevirisi; tasarım gereği dürüst, sessiz geri dönüş yok.
- **HelixTerminator** — sıfır güven terminal platformu: Her SSH oturumu güvence altına alınır, paylaşılır ve AI destekli.
- **HelixGitpx** — bir düzine sunucu üzerinde federatif Git; tek gerçeklik kaynağı, her yerde yansıtılır.
- **HelixOTA** — evrensel, bağımsız hava üzerinden güncellemeler; tasarım gereği sıfır tuğla.
- **HelixPlay** — herhangi bir GPU makineyi kendi bulut oyun cihazınıza dönüştürün.
- **Helix-Flow** — Helix platformu çıkarım ürünü. *DOĞRULANMAMIŞ / kaynak engeli: Genel depoda şu an yalnızca tek satırlık bir README dosyası bulunuyor; gerçek belgeler mevcut olduğunda ürün derinliğiyle sunulacak.*

4. LLM Altyapısı

Ürünlerin altında, onları sağlayıcıdan bağımsız ve güvenilir kılan temel katman yer alıyor: Onlarca LLM sağlayıcısı için tek bir arayüz, kodlama ajanları için tek bir kontrol düzlemi ve tek bir doğrulama gerçeklik

kaynağı. Bu katman, üstteki her şeyin modelleri değiştirmesine, sağlayıcı kesintilerine dayanmasına ve görevi anlayamadığını kanıtlayamayan bir LLM'a güvenmeyi reddetmesine olanak tanıyor.

- **HelixLLM** — tek ikili, altı mod: OpenAI ve Anthropic uyumlu çıkarım, HTTP/3 üzerinden; yerel llama.cpp, puanlı geri dönüş zinciri, RAG boru hattı, ReAct ajanları.
- **LLMProvider** — tek arayüz, 43 sağlayıcı; devre kesiciler, yeniden denemeler ve sağlık kontrolleri entegre.
- **LLMOrchestrator** — her bağımsız CLI kodlama ajanı için tek kontrol düzlemi (OpenCode, Claude Code, Gemini, Junie, Qwen Code).
- **LLMsVerifier** — doğrula, izle, optimize et: LLM/sağlayıcı/doğrulama meta verileri için tek gerçeklik kaynağı; zorunlu model kavrama kapısıyla.

5. vasic-digital araçları

Kendi başlarına zorlu bir problemi çözen, ürün kalitesinde bağımsız araçlar — ve tesadüf değil, yeniden kullanılabilir modül filosunu gerçek dünya koşullarında sınavan örnekler. Dayanıklı çok protokollü bir medya sisteminden, Markdown'dan videoya dönüşen kurs boru hattına, içerik tabanlı hash'li doküman/veritabanı senkronizasyon motoruna kadar uzanıyorlar; bazıları olgunluk seviyelerini açıkça belirtirken, bu uyarılar gizlenmiyor, olduğu gibi bırakılıyor.

- **Catalogizer** — çok protokollü (SMB/FTP/NFS/WebDAV/yerel), şifreli, kendi sunucunuzda barındırılabilir medya koleksiyonu yönetimi; Go/Gin API + React arayüzü; çevrimdışı dayanıklı izleme; 21 `digital.vasic.*` alt modül üzerine inşa edilmiştir.
- **Courses-Creator** — Markdown'dan videoya dönüşen kurs boru hattı; çoklu LLM zenginleştirme, TTS (Bark/SpeechT5), masaüstü/mobil/web oynatıcılar; API anahtarı olmadan da sorunsuz çalışma modu.
- **VisionEngine** — klasik bilgisayarlı görü ile çok sağlayıcılı LLM görüntü işlemeyi birleştiren bağımsız Go araç seti; BFS ve DOT/JSON/Mermaid ihracı destekli gezinme grafikleri; OpenCV derleme etiketiyle sınırlı.
- **DocProcessor** — doğrulama kapsamı izlemeli dokümantasyon-özellik haritası; LLM veya sezgisel/çevrimdışı çıkarım; Apache-2.0 lisansı.
- **Docs Chain** — içerik tabanlı hash'li, çift yönlü, atomik doküman/veritabanı senkronizasyonu (DAG üzerinde Salsa tarzı artımlı yeniden hesaplama). *1-5. Aşamalar TAMAMLANDI; 6-7. Aşamalar PLANLANDI.*
- **Herald** — doğal dil desteğiyle üç aşamalı niyet çözümleme (komut → LLM → netleştirme) içeren güvenilir çok kanallı bildirim sistemi; Docs Chain'in ilk kullanıcısı.
- **task_bridge** — bağımsız, çift yönlü görev/panolar senkronizasyonu (SQLite Tek Doğru Kaynak [?](#) dokümanlar [?](#) ClickUp). *P1 iskelet — senkronizasyon mantığı henüz uygulanmadı.*
- **Vasic Digital Yeniden Kullanılabilir Modül Paketi** — `digital.vasic.*` “standart kütüphanesi”: altyapı temel bileşenleri, AI yapı taşları ve savunma amaçlı LLM koruma mekanizmaları, ayrıca bir Kotlin Multiplatform yansıması. *Bazı kurumsal depolar İSKELET/ÇALIŞMA AŞAMASINDA — uyarı var, yayınlanmadı.*

6. Server Factory (altyapı otomasyonu mirası — sıralamada son sırada)

Kalitesi nedeniyle değil, tasarım gereği son sırada: Server Factory araç zinciri, AI serisinden önce geliştirildi ve “bir kere inşa et, her yerde kullan” felsefesinin ilk kez şekillendiği yeri gösteriyor. JSON ile tanımlayıp herhangi bir yere kurabileceğiniz posta sunucusu, olgun ve iyi test edilmiş bir ürün; destekleyici araçlar ise gerçek olgunluk seviyeleriyle sunuluyor, süslenmiyor.

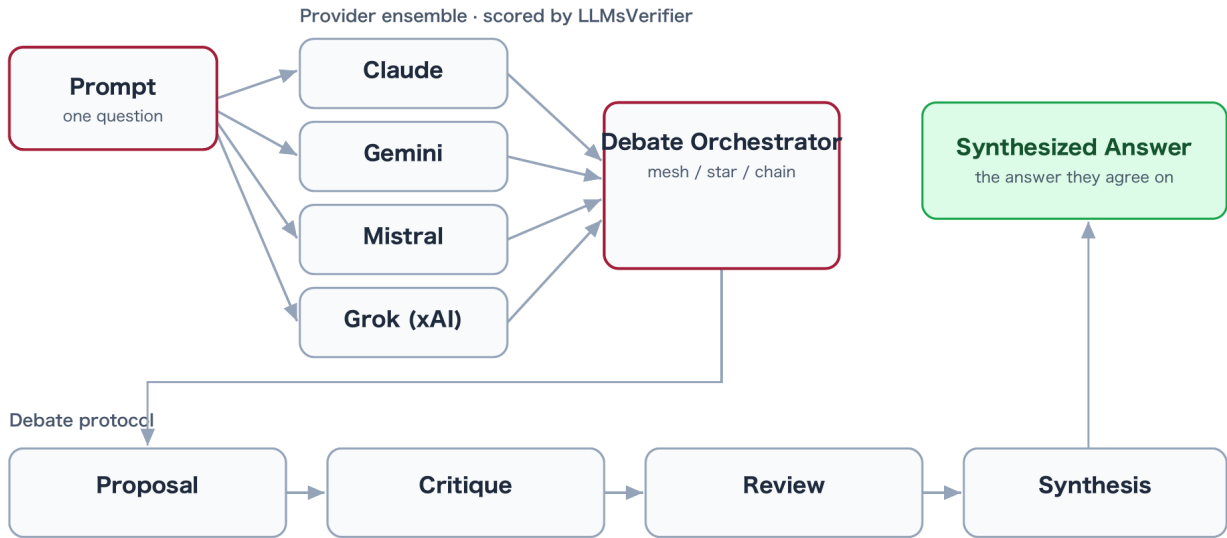
- **Mail Server Factory** — JSON ile tanımlanmış → 12 bağlantı türü ve 25 Linux dağıtımı üzerinde tam otomatik, Docker tabanlı posta sunucusu; kurumsal güvenlik; 439 geçer test ve temiz bir SonarQube geçidi. Server-Factory organizasyonunun amiral gemisi.
- **Server Factory Çekirdek Çerçevesi** — her fabrikanın üzerinde inşa edildiği ortak Kotlin motoru.
- **Qemu-Utils** — QEMU sanal makine görüntüleri, birer yapı ürünü gibi yönetilir: indirme/önbellege alma/çalıştırma, sıkıştırma/yayınlama, köprü/TAP ağ bağlantısı, ISO kurulumları; Linux + macOS.
- **Parallels-Utils** — Parallels (macOS) sanal makine görüntülerinin basit ayar dosyaları üzerinden sıkıştırılması, yayınlanması ve alınması.
- **Server Factory — Ek Bileşenler** — hizmet fabrikaları (Web/SonarQube/Önbellek Vekil Sunucusu), Tanımlar paketleri ve Yardımcı Araçlar. *Hizmet fabrikaları yer tutucu belgelerle sunuluyor — DOĞRULANMADI / erken aşama.*

7. Teknoloji endeksi (kanıta dayalı)

- **Diller:** Go (baskın), Kotlin & Kotlin Multiplatform, TypeScript, Python, Swift, Shell; PL/pgSQL; TLA+ (helix_cluster’da resmi spesifikasyonlar).
- **AI / LLM:** 40+ sağlayıcı erişimi, MCP, RAG, vector veritabanları ve embeddings, planlama (HiPlan/MCTS/Düşünce-Ağacı), LLMOps, kıyaslama (SWE-bench/HumanEval/MMLU), TTS (Bark/SpeechT5), bilgisayarla görü + LLM-vizyonu, güvenlik önlemleri/kırmızı takım.
- **Arka uç:** Gin, gRPC + Protobuf, HTTP/3 (QUIC), WebSockets, Angular, React, Kafka/RabbitMQ.
- **Veri:** PostgreSQL, SQLite, SQLCipher, Redis, Neo4j, ClickHouse, MinIO/S3/GCS/Azure.
- **Altyapı / DevOps:** Docker & Compose, Kubernetes + Helm, Prometheus + Grafana, OpenTelemetry, QEMU/Libvirt/Parallels; GitHub Actions, Gradle, Make.
- **Test / Kalite Güvencesi:** HelixQA, mutasyon kapıları içeren zorluk donanımları, `go test -race`, görsel regresyon araçları, ADB cihaz testleri, SonarQube, güvenlik taraması (semgrep/gosec/trivy/snyk/gitLeaks/nancy), TLA+ model denetimi.

Seçilen mimari

HelixAgent — ensemble debate flow



HelixAgent — Sağlayıcıların dört aşamalı bir protokol altında tartıştığı ve üzerinde uzlaştıkları yanıtı gönderdiği, LLMsVerifier ile puanlanan bir LLM hizmet topluluğu.

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)

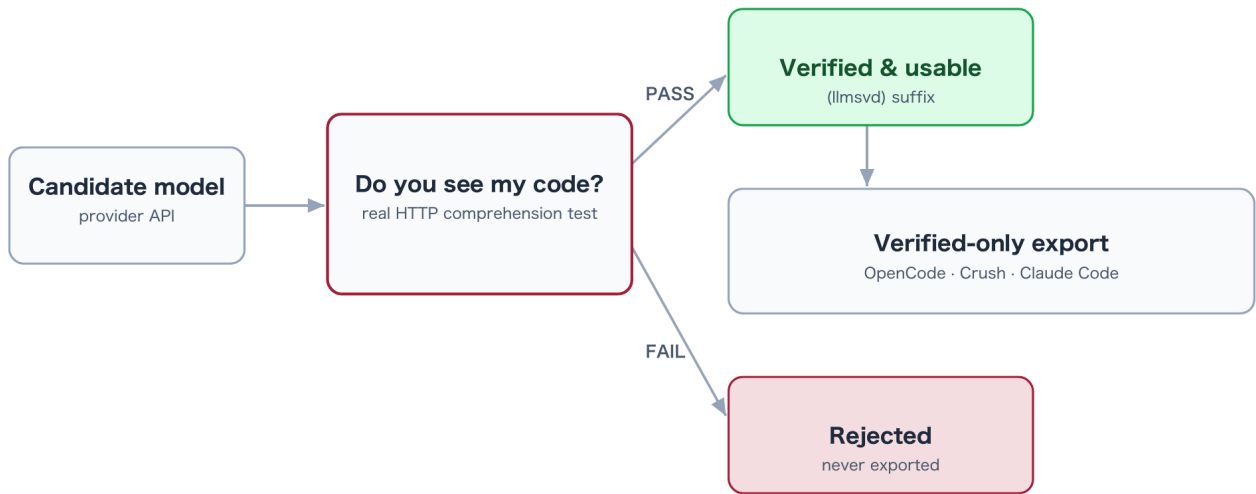
L1 · Node Runtime & Transport (gRPC, WireGuard)

heterogeneous
nodes T1-T8

L0 · Hardware Substrate (GPU → edge SBC)

Node tiers T1 (datacenter GPU) → T8 (handheld), unified under one control plane

HelixCluster — Veri merkezi GPU'larından uç cihazlara kadar AI hesaplamaları için dağıtık bir işletim sistemi; tek bir kontrol düzlemi altında.

LLMsVerifier — mandatory verification gate

LLMsVerifier — Doğrula, izle, optimize et: LLM/sağlayıcı/doğrulama meta verileri için tek gerçek kaynak; zorunlu model-anlama kontrolü ile erişim sağlanıyor.