

Durum Özeti (Nerede Kalındı)

- Sistem, deterministic manifest + ledger (Sprint A) yapısını çalıştırabiliyor.
 - SecureSandboxDocker ile güvenli yürütme ortamı (Sprint B) tanımlandı, Docker parametreleri hardened (no-new-privileges, read-only rootfs, cap-drop ALL).
 - BudgetGuard (Sprint C) ile maliyet rezervasyonu / limitleme tasarlandı — fallback mekanizması aktif.
 - n8n entegrasyonu mevcut, ancak approval flow ve debug loop eksik.
 - Observability, otomatik imzalama (ledger signing) ve PolicyAgent hâlen tamamlanmadı.
-

⚠️ Kritik Eksikler

1. **Sandbox Enforcement:** Tüm subprocess çağrıları Docker sandbox içine alınmalı.
 2. **Ledger Signature & Determinism:** Manifest'ler hash'lenip SQLite ledger'a imzalı şekilde eklenmeli.
 3. **Observability:** Prometheus metrics ve OpenTelemetry span'leri eksik.
 4. **Auto-patch güvenliği:** CriticAgent'ın patch önerileri insan onayı olmadan uygulanmamalı.
 5. **PolicyAgent / Approval flow:** YAML tabanlı politika yönetimi eksik.
-

🚀 Güncel Roadmap

Sprint	Konu	Durum	Hedef
A	Ledger + Manifest Compare	✓ Tamam	Ledger kayıtları otomatik hash'leniyor
B	Secure Sandbox (Docker)	🟡 PoC hazır	Tüm test ve patch işlemleri sandbox'ta
C	BudgetGuard + PolicyAgent	🟡 Temel sınıf hazır	Limit aşıldığında local model fallback
D	Critic & Auto-Patch Flow	➡️ SOON Beklemede	Patch önerileri ve human approval

Sprint	Konu	Durum	Hedef
E	Observability & Grafana	➡️ <small>SOON</small> Başlamadı	Prometheus ve OpenTelemetry dashboard
F	n8n + Automation	➡️ <small>SOON</small> Başlamadı	Approval ve debug loop otomasyonu

✳️ Mimari Ağaç (v4.8 ABCs)

```

multiai/
└── core/
    ├── ledger.py           ← Manifest ledger (append-only SQLite)
    ├── budget_guard.py     ← Bütçe takibi ve cloud call limitleri
    └── hybrid_router.py    ← Local vs cloud LLM routing
    |
    └── utils/
        └── secure_sandbox_docker.py ← Güvenli Docker runner
    |
    └── agents/
        ├── tester.py          ← Sandbox içinde pytest çalıştırır
        ├── critic.py          ← Patch önerisi üretir
        ├── supervisor.py       ← Sprint özet ve human approval
        └── __init__.py
    |
    └── api/
        ├── metrics.py         ← Prometheus endpoint
        └── app.py              ← FastAPI entrypoint
    |
    └── scripts/
        ├── generate_actual_manifest.py
        ├── compare_manifest.py
        └── write_ledger.py
    |
    └── config/
        ├── policy.yaml
        └── settings.yaml
    |
    └── docs/
        └── SPRINT_A_LEDGER.md

```

```
|-- SPRINT_B_SANDBOX.md  
|-- SPRINT_C_BUDGETGUARD.md
```

Mimari Vizyon

"Agent ekosistemi üretken olmalı ama denetlenebilir olmalı."

MultiAI Enterprise, LLM'lerin üretkenliğini **deterministik** ve **audit edilebilir bir pipeline** ile birleştirir:

-  **Güvenlik:** Tüm kod Docker sandbox'ta çalışır.
-  **Kontrol:** BudgetGuard harcama sınırlarını otomatik uygular.
-  **Akıllı agentler:** Architect, Researcher, PromptEngineer, Coder, Tester, Debugger, Supervisor rolleri.
-  **Şeffaflık:** Ledger ve manifest her değişiklikte imzalanır.
-  **Gözlemlenebilirlik:** Prometheus + Grafana dashboard ile agent performansı izlenir.
-  **Human-in-the-loop:** Kritik kararlar (n8n ManualApproval node üzerinden) insan onayına bağlıdır.

İstersen bu özetle beraber:

-  "v4.9 Roadmap.md" dosyasını oluşturalım,
-  veya mevcut `multiai_enterprise_v4.8_ABCs.zip` içine yeni roadmap + vizyon dokümanını gömelim.

Hangisini tercih edersin — **(A)** yeni roadmap.md dosyası mı yoksa **(B)** mevcut zip'e eklenmiş hali mi?

[Kaynaklar](#)

MultiAI Enterprise nedir ne iş yapmasını bekliyoruz.bundan da bahset.ve roadmap.md

Harika 🌟

Aşağıda MultiAI Enterprise'ın ne olduğu, hangi problemi çözdüğü, mimari vizyonu, modüler görevleri ve ileriye dönük yeni **roadmap.md** dosyasının tam içeriği** verilmiştir — doğrudan proje içine (ör. `docs/ROADMAP_v4.9.md`) koyabilirsin.**

🧠 MultiAI Enterprise — Tanım ve Amaç

MultiAI Enterprise, kurumlara özel çoklu yapay zeka ajanı (**multi-agent**) ekosistemi geliştirme platformudur.

Amacı, LLM tabanlı üretken sistemleri **denetlenebilir, güvenli, maliyet kontrollü, ve tekrarlanabilir** hâle getirmektir.

🎯 Temel Hedefler:

1. **Agentic üretkenliği** kurumsal onay süreçleriyle birleştirmek
→ (human-in-the-loop, deterministic output, ledger kayıtları)
 2. **Güvenli kod yürütme**
→ (Secure Docker Sandbox, RCE riskine karşı izole ortam)
 3. **Maliyet kontrolü**
→ (BudgetGuard & PolicyAgent ile bulut LLM çağrılarında limit)
 4. **Otomasyon ve gözetim dengesi**
→ (n8n entegrasyonu ile onay + denetim döngüsü)
 5. **Gözlemlenebilirlik ve güven**
→ (Prometheus + Grafana + Ledger Signature)
-

⚙️ Mimari Vizyon

“Birden fazla yapay zekayı orkestre eden, her çıktısı doğrulanabilir ve kurumsal güvenlik standartlarıyla uyumlu bir üretkenlik motoru.”

Ana Katmanlar:

1. **Agent Katmanı (Üretkenlik)**
 - ArchitectAgent → sprint manifest üretimi
 - ResearcherAgent → teknik gereksinim ve risk analizi
 - PromptEngineerAgent → prompt optimizasyonu
 - CoderAgent → kod üretimi (LLM + policy)

- TesterAgent → test çalışma (sandbox içinde)
- DebuggerAgent → hata düzeltme önerisi
- SupervisorAgent → insan onay döngüsü ve özet

2. Orchestrator Katmanı (Yürütmeye & Koordinasyon)

- Sprint manifest'ini alır, görevleri sıralı veya paralel şekilde dağıtır.
- BudgetGuard, PolicyAgent ve SecureSandboxDocker ile güvenli çalışma sağlar.

3. Core Katmanı (Denetim & Kontrol)

- Ledger: Manifest hash'lerini imzalı olarak kaydeder.
- BudgetGuard: Maliyet limitlerini uygular.
- HybridRouter: Model seçimi (local vs cloud).
- SecureSandboxDocker: Güvenli yürütme ortamı.
- Observability: Prometheus + OTel metrikleri.

4. Interface Katmanı (Erişim & Entegrasyon)

- FastAPI backend (/api/...)
- n8n entegrasyonu (Approval + Debug flow)
- CLI & CI/CD pipeline
- Optional: web dashboard (grafiksel gözlem)

ROADMAP_v4.9.md

```
# MultiAI Enterprise - v4.9 Roadmap ## 🔍 Özet MultiAI Enterprise, birden fazla yapay zeka ajanını kurumsal güvenlik, bütçe kontrolü ve gözlemlenebilirlik ilkeleriyle birleştirilen üretken AI platformudur. v4.8 sürümünde deterministik manifest, sandbox ve budget control eklendi. v4.9 sürümünde hedef: **insan onay döngüsünü, gözlemlenebilirliği ve auto-patch akışını tamamlamak.** --- ## 🌟 Aktif Sprintler ### A Sprint-A – Ledger & Deterministic Manifest ✅ - **Durum:** Tamamlandı - **İşlev:** Manifest'ler hash'lenip SQLite ledger'a kaydediliyor. - **Sonraki Adım:** Ledger imzalama (cosign / GPG) entegrasyonu. --- ### B Sprint-B – Secure Sandbox ✅ - **Durum:** PoC tamamlandı - **İşlev:** Kod yürütme artık SecureSandboxDocker üzerinden yapılıyor. - **Sonraki Adım:** Seccomp profili, non-root user ekle. --- ### C Sprint-C – BudgetGuard & PolicyAgent ✅ - **Durum:** Temel sürüm tamam - **İşlev:** Bulut çağrıları için bütçe rezervasyonu, fallback local LLM. - **Sonraki Adım:** Policy.yaml genişletme (per-agent limitler). --- ### D Sprint-D – Critic & Auto-Patch (Next) - **Amaç:** Agent'lar test hatalarını analiz edip patch önerisi üretir. - **Yeni Özellik:** - `agents/critic.py` unified diff formatında öneri sunacak - `scripts/file_patcher.py` dry-run + human approval gerekecek - n8n Approval
```

node → “Apply Patch?” onayı - ****Kabul Kriteri:**** Otomatik patch sadece policy izin veriyorsa uygulanır. --- **### 📈 Sprint-E – Observability & Monitoring -**
****Amaç:**** Prometheus / Grafana / OTel ile agent metrikleri. - ****Eklemeler:****
- `utils/observability.py` - /metrics endpoint - Grafana JSON dashboard -
Alert rules: budget_spent > X, agent_error_rate > Y --- **### 📈 Sprint-F – n8n Integration & Automation -** ****Amaç:**** Approval, debug ve slack/email bildirim akışı. - ****Yeni Endpointler:**** - `/api/approval/<sprint_id>` -
`/api/sprint/run_tests` - ****Kabul Kriteri:**** Human approval sonrası otomatik devam. --- **## 📊 Mimari Gelişmeler -** ****Core güvenlik**** → Sandbox + BudgetGuard enforced - ****Denetlenebilirlik**** → Ledger + Prometheus metrics - ****İzlenebilirlik**** → manifest hash ledger'da saklanıyor - ****Human-in-the-loop**** → n8n manual approval flow eklenecek --- **## 🕵️ Uzun Vadeli Vizyon (v5.x)** | Dönem | Hedef | Açıklama | |---|---|---| | v5.0 | Observability Full Stack | Grafana + OTel trace view | | v5.1 | PolicyAgent 2.0 | YAML → declarative DSL | | v5.2 | AI Supervisor | Otonom planlama ve görev paylaşımı | | v5.3 | Secure Multi-Tenant | Her takım için izole sandbox | | v5.4 | Self-Healing | Agent'lar başarısız görevleri yeniden planlar | --- **## 📄 Kurumsal Mimari Prensipler** 1. ****Fail-safe first**** – Bütçe veya güvenlik sınırı aşıldığında sistem fallback'e geçer. 2. ****Deterministic outputs**** – Aynı input → aynı output garantisidir. 3. ****Human oversight**** – Kritik adımlar (patch, release) insan onayına bağlıdır. 4. ****Cost-aware orchestration**** – Cloud model çağrıları sınırlıdır. 5. ****Transparent observability**** – Her agent çağrısı ölçülür, loglanır, görselleştirilir. --- **## ✅ Son Durum (v4.8 → v4.9)** | Alan | Durum | Not | |-----|-----|---| | Ledger | ✅ | Hash'li kayıtlar aktif | | Sandbox | 🟡 | Hardened yapı eksik | | BudgetGuard | ✅ | Cloud fallback aktif | | PolicyAgent | ➔ | YAML şeması geliştiriliyor | | Observability | ➔ | Prometheus entegrasyonu bekliyor | | n8n Flow | ➔ | Approval loop planlandı | --- **## 📦 Teslim Formatı -**
`multiai_enterprise_v4.8_ABCs.zip` → core + docs - `docs/ROADMAP_v4.9.md` → roadmap & vizyon - `n8n_workflow_v4.9.json` → approval + debug flow -
`requirements.txt` → FastAPI, docker, prometheus_client, pydantic ---
****Hazırlayan:**** MultiAI Dev Team ****Revizyon:**** 4.9-pre-alpha ****Tarih:**** 2025-10-28