# SIMON AI Agent Studio – Master Proje Dokümanı

*v3.1 (AI Agent Sistemi Uyumlu) • 2025-12-27 • TR*

| | |
|---|---|
| **Doküman Tipi** | Master Proje Dokümanı (Canonical) |
| **Sürüm** | v3.1 |
| **Kapsam** | MVP-1 (Browser Sandbox) + Hibrit LLM Gateway + Approval/Audit |
| **Hedef** | Kurumsal, güvenli ve ölçeklenebilir AI Agent geliştirme stüdyosu |
| **Uptime Hedefi** | >95% (MVP-1) |
| **Maliyet Hedefi** | İlk ay ~ $68 • Normal kullanım ~ $41/ay (hedef) |
| **Zaman Planı** | 18 gün (MVP-1) |

# 1. Yönetici Özeti

Simon AI Agent Studio, çoklu LLM sağlayıcısı ve yerel model desteği ile çalışan, kurumsal güvenlik kontrolleri (Approval Gate, Egress Allowlist, Audit Ledger) bulunan bir AI Agent çalışma ortamıdır. v3.1 revizyonu; üretim hatalarını azaltmak ve güvenliği kurumsal seviyeye çıkarmak için 10 kritik düzeltmeyi standarda bağlar.

- Canonical API endpoint'i tekleştirildi: POST /api/tasks.
- Credential izolasyonu: API anahtarları yalnızca Orchestrator'da tutulur.
- Egress Proxy (Squid) ile domain allowlist zorunlu hale getirildi.
- UI Runner ölçeklenebilir yapıya geçirildi: Celery/Redis job queue.
- Idempotency: her UI action için unique key ile tekrar çalıştırma güvenliği.
- Screenshot policy: TTL 30 gün + PII maskeleme.
- Audit & Telemetry MVP-1'de zorunlu: structured logs + cost ledger + approval ledger.
- Ollama çalışma modları standardize edildi: Development / Production / BYOK.

# 2. Kapsam

## 2.1 MVP-1 Kapsamı (18 Gün)

- Orchestrator (FastAPI): görev analizi, alt-görev parçalama, onay akışı, durum yönetimi.
- LiteLLM Gateway: routing/failover/budget (iş mantığı içermez).
- UI Runner (Browser Sandbox): Playwright tabanlı, job queue ile çalışan Computer Use loop.
- Approval Gate: LOW/MEDIUM/HIGH risk matrisi + manuel onay ekranı (HIGH).
- Audit & Telemetry: log/metric/cost/approval ledger; MVP-1'de kapatılamaz.
- Egress Proxy (Squid): allowlist enforced; dış çağrılar sadece izinli domainlere.

## 2.2 MVP-2 (Sonraki Faz) – Desktop VM

MVP-2'de OS seviyesinde Desktop VM (VNC/RDP) entegrasyonu planlanır. MVP-1'de kapsam dışıdır.

# 3. Mimari v3.1

v3.1 mimari 6 katman üzerinden standardize edilmiştir:

| Katman | Bileşen | Sorumluluk |
|---|---|---|
| 1 | Orchestrator | Görev analizi, planlama, state yönetimi, approval akışı, credential yönetimi |
| 2 | LLM Gateway (LiteLLM) | Model routing, failover, budget, rate limit, telemetry |

| | | |
|---|---|---|
| | | (LLM seviyesinde) |
| 3 | UI Runner Service | Computer Use loop, Playwright sandbox, job queue worker'ları |
| 4 | Egress Proxy | Allowlist enforcement, outbound kontrol, loglama |
| 5 | Approval Gate | Risk sınıflandırma, manuel onay (HIGH), policy yönetimi |
| 6 | Audit & Telemetry | Structured logs, cost ledger, approval ledger, metrikler/alarmlar |

### 3.1 Canonical API

- POST /api/tasks (canonical)
- GET /api/health (orchestrator healthcheck)
- GET /api/tasks/{id} (task status/result)

### 3.2 Computer Use Teknik Standardı

- Beta header: computer-use-2025-01-24
- Tool version: computer_20250124
- UI Runner loop: screenshot → model → tool_use → tool_result → tekrar; hedefe ulaşıldığında DONE.
- Idempotency: her action için action_id + idempotency_key.

## 4. Model Stratejisi (Hibrit)

MVP-1'de önerilen çalışma modu:

| Rol | Model | Kullanım | Not |
|---|---|---|---|
| Primary | Claude Sonnet 4.5 | Planlama + kod + agent orchestration | Ana iş yükü |
| Failover/Vision | OpenAI GPT-4o | Rate limit/downtime + görsel analiz | İkinci görüş / vision |
| Local | Ollama (Qwen2.5-coder vb.) | Basit işler, log parse, özet | Maliyet $0; 3 mod |
| Premium (Ops.) | Claude Opus 4.5 | Nadir – premium ihtiyaçlar | Opsiyonel |
| Reasoning (Nadir) | OpenAI o1 | Karmaşık reasoning | Pahalı; limitli |

## 5. Güvenlik ve Uyum

### 5.1 Credential İzolasyonu

- API anahtarları sadece Orchestrator container'ında environment olarak bulunur.
- UI Runner, Web ve diğer servislerde anahtar bulunmaz.
- Egress proxy üzerinden çıkış zorunlu; doğrudan internet erişimi kısıtlanır.

### 5.2 Egress Allowlist

- Squid container allowlist ile sadece izinli domainlere outbound.
- LLM sağlayıcı domainleri + gerekli CDN'ler allowlist'e eklenir.
- Tüm outbound loglanır (audit).

### 5.3 Screenshot Policy

- TTL: 30 gün (varsayılan).
- PII maskeleme: e-posta/telefon/ad soyad gibi alanlar maskelenir.
- HIGH risk kategorilerinde screenshot retention isteğe bağlı azaltılır.

## 6. Operasyon ve Gözlemlenebilirlik

- Audit zorunlu: structured logs (JSON), cost ledger, approval ledger.
- Budget default: MVP-1 için $100/ay (Pro: $300 ayrı).
- Alert threshold: %80 bütçede uyarı; %100'de hard-stop (config).
- UI Runner: Celery worker'ları ile yatay ölçeklenebilir.

## 7. Zaman Planı – 18 Gün

Önerilen teslim planı (MVP-1):

1. Gün 1-2: Kritik altyapı (Egress Proxy, Credential izolasyonu, Celery/Redis job queue temeli)
2. Gün 3: Canonical API /api/tasks + Screenshot policy + Audit zorunlu altyapısı
3. Gün 4: Idempotency + Ollama modları + doküman iç tutarlılık temizliği
4. Gün 5-10: Orchestrator core akışları + LiteLLM routing/failover + temel UI Runner loop
5. Gün 11-14: Approval Gate UI + policy entegrasyonu + uçtan uca senaryo testleri
6. Gün 15-18: Stabilizasyon, güvenlik checklist, performans, release tag + runbook

## 8. Maliyet Planı

- İlk ay hedef: ~$68 (proxy + sandbox dahil).
- Normal kullanım hedefi: ~$41/ay.
- Maliyet kontrol araçları: prompt caching, batch (uygun ise), local Ollama agresif kullanım, budget manager hard-limit.

## 9. DevOps ve Sürümleme

- Tek depo disiplin: tek repo, tek compose entrypoint.
- Rollback: her release öncesi git tag; compose down + tag checkout.
- Dev ortam: docker compose (base + dev override).
- Durum yönetimi: docs/PROJECT_STATE.md + docs/RUN_LOG.md (model/sohbet geçişlerinde ortak hafıza).

## 10. Ekler

### 10.1 Çalıştırma Komutları (Windows PowerShell)

Repo kökünde:

```
docker compose -f docker-compose.yml -f docker-compose.dev.yml up -d --build
```

Durum:

```
docker compose -f docker-compose.yml -f docker-compose.dev.yml ps
```