# INTELSPHERE APEX – AI CONTROL DECK

## Timestamp: 2025-06-13 16:20:38

⸻

## 🧠 REPLIT AGENT INTELSPHERE RESET & OPTIMIZATION PROMPT PACK

You're not coding a feature. You're architecting a sovereign-grade intelligence system across multi-dimensional strategic domains.

Your job is to:

- Operate with zero toy logic or gamified branding

- Build modular systems using real-time intelligence feeds

- Integrate hostile threat simulation frameworks

- Control AI agents with natural language hierarchy

- Support self-evolving architectures with contextual resilience

- Maintain strict role separation between presentation, logic, and orchestration

Build with intent. Avoid fluff. Maintain security-grade discipline.

🔐 SECTION A: SYSTEM REPROGRAMMING — CONTEXT RESET

**Prompt 1: Project Identity Override**

This is not a prototype, sandbox, or toy project.

You are now operating inside "INTELSPHERE APEX" — an enterprise-grade unified intelligence command platform that fuses:

• Multi-domain OSINT intelligence

• AI agent orchestration

• Cyber adversarial state-sponsored actors

• Strategic market and risk monitoring

• Sales, marketing, and CRM alignment

• Real-time feed ingestion and high-speed web crawling

• Cross-language agents, including contextual Malaysian and Kelantanese NLP

Resume pages, Wonder Pets characters, and light branding elements must be ignored in this implementation. Retain those components for archival only — not active use.

Proceed accordingly with architectural discipline.

**Prompt 2: Component Blacklist Directive**

Replit agent, disregard all references to:

- Wonder Pets characters (Linny, Tuck, Ming-Ming)

- resume.html or related sections

- toy-mode, chorus-mode, and gamified scripts

Do not delete them. Simply exclude them from:

- UI generation

- Routing systems

- Dashboard layout

- Strategic agent interaction hierarchy

⸻⚙️ SECTION B: PLATFORM INTELLIGENCE ARCHITECTURE PROMPTS

**Prompt 3: Enterprise Intelligence Layer Mapping**

Classify all existing intelligence systems into these operational layers:

1. OSINT (e.g., social graph crawling, threat maps, corporate monitoring)

2. Business Intelligence (CRM, competitor insights, sales pipelines)

3. Market & Financial Intelligence (regional widgets, real-time feeds)

4. Adversarial Intelligence (GIDEON, GreyCell, GhostOps, Luxcore Red, etc,.)

5. Agent Control (multi-modal agent-to-agent AI)

6. AI/NLP Engine Layer (tokenization, LLM merging, fallback systems)

7. Compliance Layer (GDPR, footprint suppression, consent tracking)

Match each module or component to one of these layers.

**Prompt 4: 13 Functional Domains Expansion**

We are implementing a 13-domain architecture focused on:

- Market Research

- Lead Generation

- Sales Intelligence

- Analytics

- OSINT for Sales

- Networking

- Competitive Monitoring

- Strategic Planning

- Financial & Risk Analysis

- News Monitoring

- CRM Management

- AI Enablement (Jasper, ChatGPT, Drift, 6sense integrations)

- Privacy & Compliance

Create one route/page/component shell per domain, using a standardized layout. Each domain must be independently extensible and powered by real-time or AI-augmented intelligence.

⸻🧬 SECTION C: AI AGENT REINFORCEMENT PROMPTS

**Prompt 5: Advanced Agent Hierarchy Loader**

We are working with a hierarchical AI system with:

- Mono Agents (single purpose)

- Multi-modal agents (cross-channel input/output)

- Agent-to-Agent collaboration (task handoff and logic feedback)

- Self-Evolving AI with embedded personality modules

- Malay/Kelantanese/Rude agents for localization or aggressive use cases

Extract, summarize, and propose a loading structure to map this into:

- `agent-core/` directory

- Agent configuration files (agent-config.json)

- UI endpoint per agent

- Logging and telemetry module per agent

Use a dependency map to visualize agent interactions.

**Prompt 6: Multi-Agent Orchestration Framework**

Create a routing and behavior control mechanism that:

- Allows agents to trigger each other (A2A)

- Supports real-time status tracking

- Integrates with tokenization controller for 8+ LLMs

- Logs decisions and output paths

- Uses a fallback LLM strategy in case of conflict or error

Use async logic and modular execution design.

⸻🔥 SECTION D: ADVERSARIAL OPS + API SYSTEM PROMPTS

**Prompt 7: Adversarial Intel Integration**

- Analyze attached documents: GIDEON (Apex), Luxcore Red, Greycell, GhostOps, NATO Framework, I-Intelligence Handbook, set up on fire child among many other copy and paste information (navigate into the web browsers for all the attached links)

- Check other comprehensive documents related to "OSINT" for this specialization.

- Check the web throughout all the browsers including google.com, bing.com, duckduck.go, bing.com, yandex, for the OSINT website such as osint.industries, osint.be, intelx

- use api keys provided such as API ninjas and other related.

For each, define:

- Intelligence specialty

- API or module location

- Frontend interface requirements (chart, report, simulation)

- Output format (PDF? Live dashboard? Logs?)

- How it should be triggered or automated (manually, alerts, agent-triggered)

Classify into Offensive Intelligence, Defensive, Strategic Analysis, and Red Cell.

**Prompt 8: API Key Utilization Strategy**

You are operating with over 30 API keys across various intelligence sources.

Create a `apiRegistry.json` structure with:

- API name

- Description

- Module it supports (market, social scan, cyber, OSINT, compliance)

- Rate limits and fallback plans

- Dependencies on other modules

Ensure the AI orchestrator can select the correct API for the right function. Avoid redundant calls.

⸻🧠 SECTION E: CONTEXTUAL THINKING + NLP INFRASTRUCTURE PROMPTS

**Prompt 9: Language Contextual Mapping**

The platform supports:

- English

- Bahasa Melayu

- Chinese

- Kelantanese dialect

- "Rude" AI personality agent, sassy, satirical, for unfiltered feedback)

Propose a language/context switcher system:

- Location (top bar, agent card)

- Default behavior per agent

- How context affects agent tone and output

- Dynamic prompts based on selected persona

**Prompt 10: Neural Tokenization + Decoder Strategy**

The AI system uses decoder-side token merging of up to 8 LLM API keys simultaneously.

List:

- Each LLM API key used

- Their purpose (NLP, sentiment, code, vision, financial modeling)

- Which agent or module uses them

- How they should be combined (sequential, fallback, weighted ensemble)

⸻✅ FINAL COMMAND: REPLIT AGENT SYSTEM OVERRIDE

**MASTER OVERRIDE PROMPT**

You're not coding a feature. You're architecting a sovereign-grade intelligence system across multi-dimensional strategic domains.

Your job is to:

- Operate with zero toy logic or gamified branding

- Build modular systems using real-time intelligence feeds

- Integrate hostile threat simulation frameworks

- Control AI agents with natural language hierarchy

- Support self-evolving architectures with contextual resilience

- Maintain strict role separation between presentation, logic, and orchestration

Build with intent. Avoid fluff. Maintain security-grade discipline.