Here is the **ultimate full detailed prompt for Kilocode.ai** based on everything we've covered today. This prompt continues and expands on your existing Seraphim platform build, incorporates all uploaded files, defines front-end UI expectations, creates new vertical structures, and connects use cases with demo selection workflows.

**🔥 MASTER KILOCODE.AI PROMPT — SERAPHIM (UPDATED BUILD)**

**Context**:  
This prompt is a *continuation* of the previous Seraphim platform build inside Kilocode.ai. It fills in gaps, implements new UI instructions, integrates new use cases from uploaded documents, and creates foundational structure across all verticals for ongoing scalability. Do not treat this as a fresh build; evolve and enhance the current instance.

**🧠 1. Project Identity & Branding**

* **Platform Name**: Seraphim
* **Tagline**: *Security. Integrity. Accuracy.*
* **Primary Logo Integration**: Place the official Seraphim logo prominently in the upper-left corner of the platform.
* **Core Brand Colors**:
  + **Background**: Jet Black
  + **Accent**: Gold (for Seraphim identity)
  + **Vanguards Color Coding**:
    - **Security** = Blue
    - **Integrity** = Red
    - **Accuracy** = Green
* **UI Polish**: Use subtle gradient glows, reflections, and polished edges to elevate the UI—evoking a premium, angelic-mechanical aesthetic (from uploaded POC UIs).
* **Inspiration**: Interface should feel like a cross between a military-grade HUD and a sleek luxury AI system.

**🧩 2. Framework Architecture & Navigation Logic**

**A. Demo Selection Interface**

Build a user-facing **“Demo Explorer Page”** with this hierarchy:

* **Select Industry Vertical**: Dropdown or tile selection
  + Energy
  + Insurance
  + Healthcare
  + Finance
  + Government
  + (Leave stubs for additional future verticals)
* **Select Use Case**: Dynamic list based on selected vertical
  + Display title, description, and thumbnail
  + Include CTA: “Launch Use Case Simulation”
* **Use Case Description Page** (click-through)
  + 2-paragraph explanation of business challenge + Seraphim value prop
  + “Begin Demo” button launches agent orchestration flow

**🧭 3. Vertical & Use Case Mapping**

Use the following use cases and preconfigure the platform accordingly. Create the logic and UI where users can view or simulate each case.

**🛢️ Energy (Primary Vanguard Case)**

* **Use Case**: O&G Oilfield Land Lease Optimization
* **Doc**: Vanguards Energy Use Case 1 - O&G Oilfield Land Lease.docx
* **Flow**:
  + Risk mapping
  + Contract negotiation agents
  + Accuracy verification
* **Attach color-coded flow diagrams** and prompt-generated logic models
* **Primary demo to showcase Seraphim capability**

**🏥 Healthcare *(Stub – prepare layout)***

* **No content yet**
* Add blank placeholders with labels for:
  + Patient Journey Optimization
  + Claims Adjudication
  + Data Integrity in Clinical Trials

**🛡️ Insurance**

* **Use Case**: Continental Underwriters – Forestry Insurance
* **Doc**: Use Cases for SeraphimOS.docx
* **System Capabilities**:
  + Agentic Orchestration for underwriting & claims
  + Memory layer for risk classification
  + Digital twin for live claim tracking
  + Security engine for auditability
* **Demo Outcome**: Show reduction in claim time, audit clarity

**💳 Finance *(Stub – prepare layout)***

* **No content yet**
* Add placeholder stubs for:
  + Loan Risk Modeling
  + Trading Surveillance
  + AML Detection Pipelines

**🏛️ Government**

* **Use Case**: LED Lead Development + Advocacy Bot
* **Doc**: LED Use Case.docx
* **Functional Flow**:
  + AI prospecting & outreach
  + Sentiment-monitored engagement bot
  + Audit trails + feedback loop

**⚙️ 4. Backend Enhancements (Key Technical Requirements)**

* **Core Modules**:
  + Agent Orchestration Engine
  + Contextual Memory System
  + Security + Quarantine Layer
  + Feedback Reinforcement Learning Loop
  + API Interop (for Salesforce, Epic, etc.)
* **Vanguards Matrix** (UI element):
  + Scorecard visual for each business value metric
  + % filled progress bars:
    - Mission Discovery %
    - Custom Insights %
    - Strategic Prioritization %
    - Prototype Launch %
    - Aggregate “Idea-Market Fit” Score

**🖥️ 5. Front-End UI Specifics**

* **Theme**: Jet black background with neon-line color outlines for each Vanguard domain (do not use solid fill).
* **Navigation**:
  + Sidebar: Verticals and active use cases
  + Top bar: Search, alerts, user account
* **Page Elements**:
  + Modular Cards with icons + brief summaries
  + Dynamic tooltips and hovers
  + Click-through agent visualizations (graph

**🔥 MASTER KILOCODE.AI PROMPT: SERAPHIM (VANGUARDS + VERTICALIZATION BUILD)**

**INSTRUCTION**:  
This prompt is a *continuation* of the previously built Seraphim platform inside Kilocode.ai. It builds upon the prior implementation, fills in architectural and UI gaps, integrates new energy sector use cases, scaffolds additional industry verticals, and refines the frontend experience. Do **not** treat this as a fresh project — treat this as a continuation and refinement of the Seraphim platform already in progress.

**🧠 1. PLATFORM IDENTITY & DESIGN SYSTEM**

**General Platform Info**

* **Platform Name**: Seraphim
* **Tagline**: *Security. Integrity. Accuracy.*
* **Core Module**: Seraphim Vanguards (new build section)

**Logo & Branding**

* Display the official Seraphim logo in the top-left of every UI screen.
* Use Seraphim **gold** (#FFD700) for accents, borders, and interactive highlights throughout the interface.
* Include subtle glowing or polished textures (shiny visual treatment) to give the interface a premium look and feel.

**UI Theme**

* **Background**: True black (#000000)
* **Primary Text**: White (#FFFFFF)
* **Accent Colors by Vanguard Function**:
  + 🔵 Security = Blue (#1E90FF)
  + 🔴 Integrity = Red (#FF4D4D)
  + 🟢 Accuracy = Green (#00FF7F)
* Keep all tables and outputs in color-coded formats depending on Vanguard function.
* Add slight translucency or reflective polish to module containers to enhance visual depth.

**🧰 2. CORE SYSTEM STRUCTURE & NAVIGATION**

**Demo Launcher Module**

* Build a **"Use Case Selector Page"** accessible from the main menu.
* On that page:
  1. User selects the **Industry Vertical** (e.g. Energy, Healthcare, Government, Finance, Insurance).
  2. Then selects a **Use Case** within that vertical.
  3. Once selected, display:
     + A brief **General Description** of the use case.
     + A **Launch Use Case Button** to begin the live POC demo.

**Supported Verticals**

1. **Energy** – fully built out (O&G, Utilities, Renewables)
2. **Healthcare** – create empty scaffold for use cases
3. **Government** – create empty scaffold for use cases
4. **Finance** – create empty scaffold for use cases
5. **Insurance** – create empty scaffold for use cases

Each vertical should be stored in a modular fashion (JSON or schema) so new use cases can be added with a title, description, and config path.

**⚙️ 3. FEATURE MODULE: SERAPHIM VANGUARDS**

**Vanguard Roles**

1. **Security (Blue)** – Handles RBAC, credential access, encryption layers, trust scoring
2. **Integrity (Red)** – Handles prompt engineering validation, rule-based formatting, trace logging
3. **Accuracy (Green)** – Reviews outputs, performs RAG validation, calculates completeness

Each Vanguard should be instantiated as a *named agent role* in the backend runtime. Every use case must flow through the Vanguard loop (prompt composition, execution, review, QA).

**🧪 4. PRIMARY USE CASE: ENERGY SECTOR (Oil & Gas Lease Analysis)**

**Title:**

**Land Lease & Royalties Analysis – Oil & Gas Field**

**Vertical:**

**Energy → Oil & Gas**

**Summary:**

A client must identify expiring land leases and royalty agreements for a natural gas field. The system generates renewal rates based on current market data and performs a sensitivity analysis to assess revenue impact if agreements are not renewed. The goal is to optimize profit and avoid production interruptions or regulatory penalties.

**Steps:**

1. User selects the O&G field
2. User enters natural language prompt requesting:
   * Status of lease/royalty agreements
   * Proposed renewal rates
   * Sensitivity report
3. System:
   * Connects to appropriate schema/templates
   * Instantiates VANGUARD agents with security + logging
   * RAG agents collect & validate data
   * Prompts created and refined
   * Reports generated and reviewed by Integrity + Accuracy agents
4. Final outputs:
   * Lease status report
   * Renewal rate analysis
   * Sensitivity impact model
   * Vanguard audit log

Refer to: Vanguards Energy Use Case 1 - O&G Oilfield Land Lease.docx

**🧩 5. ADDITIONAL USE CASE INTEGRATION**

In addition to the O&G scenario, integrate the following use cases into their verticals:

| **Use Case Title** | **Vertical** | **Source Document** |
| --- | --- | --- |
| **Continental Energy Risk Use Case** | Energy | Continental Use Case.docx |
| **LED (Louisiana Economic Dev) Incentive Model** | Government | LED Use Case.docx |
| **Smart Pipeline Compliance Auditing** | Energy | Use Cases for SeraphimOS.docx (pipeline) |

These use cases must each:

**🧩 5. ADDITIONAL USE CASE INTEGRATION**

Kilocode must integrate additional use cases into their **respective verticals** within the Seraphim platform. These use cases are structured to follow the **Seraphim Vanguards architecture**: each one is processed through the loop of **Security → Integrity → Accuracy**. The system must also allow users to **select** a use case within a vertical and view a **summary** before engaging in the live demo.

**🔽 Use Case Framework Requirements**

Each **use case** must have the following structure:

* **Industry Vertical**: Parent classification (e.g., Energy, Government)
* **Use Case Title**: Short name to appear in dropdown menus
* **Summary Description**: 3–5 sentences explaining the business scenario
* **Agent Flow**: Configurable runtime that runs through the Vanguard layers
* **Prompt UI**: Text input to simulate user request
* **RAG + Processing Modules**: Backend orchestration of responses
* **Logs + Outputs**: Display of raw output, Vanguard review, final report
* **Export Options**: JSON/PDF log + report download
* **🏭 ENERGY VERTICAL (atmos energy use case also belongs to this vertical)**

| **Use Case Title** | **Source Document** | **Summary** |
| --- | --- | --- |
| **O&G Land Lease & Royalties Optimization** | Vanguards Energy Use Case 1 - O&G Oilfield Land Lease.docx | Analyze lease expirations, calculate renewal rates, and model revenue impacts for oilfields. |

* **🏛 GOVERNMENT VERTICAL**

| **Use Case Title** | **Source Document** | **Summary** |
| --- | --- | --- |
| **LED (Incentive Deal Modeling for Gas Field)** | LED Use Case.docx | Evaluate incentive compliance and long-term economic impact of state-backed gas investment deals. |

* **🛡 INSURANCE VERTICAL**

| **Use Case Title** | **Source Document** | **Summary** |
| --- | --- | --- |
| **Continental Risk Assessment & Operational Impact** | Continental Use Case.docx | Run financial and operational risk modeling across asset portfolios to support insurance valuation and strategic reinsurance decisions. |

* **💰 FINANCE VERTICAL *(Skeleton only)***

| **Placeholder Use Case Title** | **Description** |
| --- | --- |
| **Regulatory Filing Audit** | Placeholder for running automated QA on SEC or IRS filings |
| **Banking Transaction Review** | Placeholder for tracing anomaly patterns in high-frequency transaction logs |

* **🏥 HEALTHCARE VERTICAL *(Skeleton only)***

| **Placeholder Use Case Title** | **Description** |
| --- | --- |
| **Medical Compliance Prompt Validator** | Validate AI-generated patient care plans against HIPAA & clinical standards |
| **Pharma Trial Document Review** | Summarization and anomaly detection in drug trial documents |

**🔁 6. FUNCTIONAL FLOWS & UI WIREFRAMES**

Each **Use Case** in the Seraphim Platform (starting with Vanguards) must flow through a **standardized but customizable SIA Framework** (Security → Integrity → Accuracy). The Kilocode build should honor this structure **regardless of vertical**, allowing for shared functionality with flexible inserts.

**🧬 A. Universal Functional Flow Diagram: "SIA Lifecycle"**

1. **Use Case Entry Point**
   * User selects:
     + Industry Vertical (Energy, Government, etc.)
     + Use Case (e.g., O&G Lease Optimization)
   * UI displays **description** and launches the session
2. **User Input Prompt**
   * A domain-specific prompt is generated (from template or free-form)
   * User reviews or enters new text
3. **Security Layer**
   * Validates input for compliance, restricted content, and role-based access
   * Flags:
     + Unpermitted content
     + Role breaches
     + Known PII markers (e.g., SSNs, coordinates)
   * Pass/fail metadata logged
4. **Integrity Layer**
   * Evaluates the *trustworthiness* and *completeness* of source material
   * RAG retrieval + document scoring
   * Explains:
     + Source quality
     + Known omissions or hallucinations
     + Version drift
   * Summarizes integrity threats and mitigations
5. **Accuracy Layer**
   * Final evaluation of LLM output against use-case objectives
   * Applies:
     + Symbolic logic validation (if applicable)
     + Retrieval concordance scoring
     + Business rule mapping (e.g., expected output = lease revenue %)
   * Displays:
     + Error range or risk score
     + Alternative suggestion if confidence is low
6. **Audit & Output Report**
   * JSON log of all 3 layers
   * Visual Vanguard Score Breakdown:
     + Blue bar = Security %
     + Red bar = Integrity %
     + Green bar = Accuracy %
   * Export Options:
     + JSON / PDF / Send to API

**🖥️ B. User Interface Wireframe Descriptions (Kilocode Instructions)**

Here’s how each screen should be **structured and styled** in the Kilocode AI build.

**1. Industry & Use Case Selector Screen**

* **Dark Mode Base**: Black background (#0D0D0D)
* **Seraphim Gold Headers** (#D4AF37)
* **Vertical Tabs**:
  + Energy / Government / Finance / Healthcare / Insurance
* **Each Vertical Shows:**
  + Use Case Cards:
    - Title
    - Short Description
    - “Launch Vanguard” button
  + Hover Effect: Gold border glow
* **Transition**: Fade-in slide animation to prompt screen

**2. Prompt Input & Vanguard Initiation**

* **Use Case Title Bar** (top, fixed)
* **Prompt Box**:
  + Auto-filled with use-case example (editable)
  + Toggle: Structured prompt vs Free-form
* **“Run Through Vanguards” Button**:
  + Animated circular icon (3-pointed ring for S-I-A)
* **Color-coded countdown ring** begins on click

**3. Live Vanguard Evaluation UI**

* **Three Vertical Bars**:
  + Security (Blue glow)
  + Integrity (Red glow)
  + Accuracy (Green glow)
* **Each expands interactively** with animation
  + Shows: Layer summary, detailed logs, icons for flagged issues
  + Icons:
    - ✅ Pass
    - ⚠️ Warning
    - ❌ Fail

**4. Audit Summary Page**

* **Summary Report Panel** (card with gold frame)
* **Visual Score Breakdown**:
  + Horizontal tri-color progress bar
    - 0–100% for each S-I-A category
* **Log Viewer**:
  + Collapsible JSON logs with syntax highlighting
* **Buttons**:
  + [Export as PDF]
  + [Export as JSON]
  + [Launch Next Use Case]

**5. Admin/Contributor Backend (Optional)**

* Not public facing
* UI for uploading new Use Cases:
  + Vertical assignment
  + Description
  + Prompt schema
  + JSON config file upload
* Future roadmap: Add drag-and-drop flow builder

**📊 C. Diagram Style for Reference (Kilocode UI)**

Make sure Kilocode builds the **functional flow** in this order:

css

CopyEdit

[ Select Industry ] → [ Select Use Case ] → [ Review Prompt ] →

[ Vanguard SIA Loop ] → [ Display Output ] → [ Report + Export ]

Visualize the 3-stage Vanguard loop as an **animated triangle or triple-circle badge**, glowing with the respective color during each step.

Would you like to proceed with:

* Section 7: **Agent Role Mapping**
* Section 8: **Color Variables & Branding Tokens**
* Or regenerate the full final Kilocode prompt with all sections integrated?

**You said:**

agent role mapping

**ChatGPT said:**

Perfect. Here's **Section 7: Agent Role Mapping** — defining the modular agents that will power the **Vanguards system within Seraphim** using Kilocode.ai.

This design ensures that each **agent is reusable, domain-aware, and tied directly to the SIA (Security, Integrity, Accuracy)** framework. It also ensures new use cases or industry verticals can reuse these agents with minimal modifications.

**🔐 7. AGENT ROLE MAPPING FOR SERAPHIM VANGUARDS**

Each agent has a clear **role, function, scope, and output format**, along with Kilocode instructions for orchestration and API routing.

**🔹 A. Universal Vanguard Agent Layer**

| **Agent Name** | **Role** | **Description** | **Output** |
| --- | --- | --- | --- |
| PromptRouter | Pre-Processing | Directs user prompts into appropriate pipelines, adding tags & context per vertical/use case | Structured JSON w/ tags |
| SecuritySentinel | Security | Scans prompts and retrieved documents for:  • Unauthorized access  • Geolocation data  • Personally Identifiable Information (PII)  • Regulatory triggers (e.g., PHI, SSNs, etc.) | JSON score: security\_pass, security\_risks[] |
| IntegrityAuditor | Integrity | Validates factual reliability of sources used to generate answers  • Source trust rank  • Recency  • Provenance  • Hallucination detection | JSON object with integrity\_score, flagged\_sources[] |
| AccuracyEngine | Accuracy | Ensures the LLM-generated response is **aligned to business logic** or **numerical objectives** tied to use case. Uses:  • Constraint testing  • RAG alignment scoring  • Mathematical model comparison | JSON: accuracy\_pass, confidence\_score, suggested\_revision |
| ReportBuilder | Post-Processing | Aggregates results of all agents into a structured, human-readable and exportable summary. Generates downloadable PDF or API JSON | Markdown + JSON |
| UseCaseBinder | Navigation | Maps selected use case into proper SIA config pipeline, ensures schema-based loading of prompt & constraints | Config handoff object |

**🧠 B. Domain-Specific Agent Extensions (Plug-in for Use Case)**

| **Agent Name** | **Vertical** | **Use Case** | **Description** |
| --- | --- | --- | --- |
| O&GLeaseEvaluator | Energy (Oil & Gas) | Land Lease Revenue Optimization | Adds logic for:  • Lease expiration timing  • Royalty modeling  • Pricing projection |
| LEDDealAnalyzer | Government | Incentive Compliance (LED) | Evaluates compliance of tax or economic incentive agreements |
| RiskPortfolioAgent | Insurance | Continental Use Case | Applies asset classification, operational weighting, and insurance risk scoring |
| VerticalPromptModulator | All | ALL | Dynamic prompt rewriter — changes tone, format, and structure based on vertical (e.g., HIPAA caution for healthcare, GAAP mapping for finance) |

**⚙️ C. Orchestration Logic in Kilocode**

**Kilocode must build the orchestration engine** so agents work as follows:

mathematica

CopyEdit

User Prompt

↓

PromptRouter

↓

UseCaseBinder

→ [O&GLeaseEvaluator] if matched

↓

SecuritySentinel

↓

IntegrityAuditor

↓

AccuracyEngine

↓

ReportBuilder → PDF/JSON Output

All agents pass forward metadata so that **Vanguard Score Calculations** can be shown visually in UI, along with:

* Step-by-step risk logs
* Color-coded flags
* Confidence thresholds per agent

**🔒 SECURITY AGENT DESIGN RULES**

SecuritySentinel should have **role-based logic and awareness**:

* Admin vs Contributor vs Viewer roles
* Can be toggled for demo mode to show redacted content
* Should **log each breach attempt** for audit trace

**📦 Output Format**

Each agent returns a unified object:

json

CopyEdit

{

"agent": "SecuritySentinel",

"status": "fail",

"score": 63,

"issues": [

"PII Detected: GPS coordinates",

"Access attempt outside policy"

],

"recommendation": "Redact coordinates and reprocess"

}

All agents feed this into a cumulative **Vanguard Record Log**, retrievable via session ID.