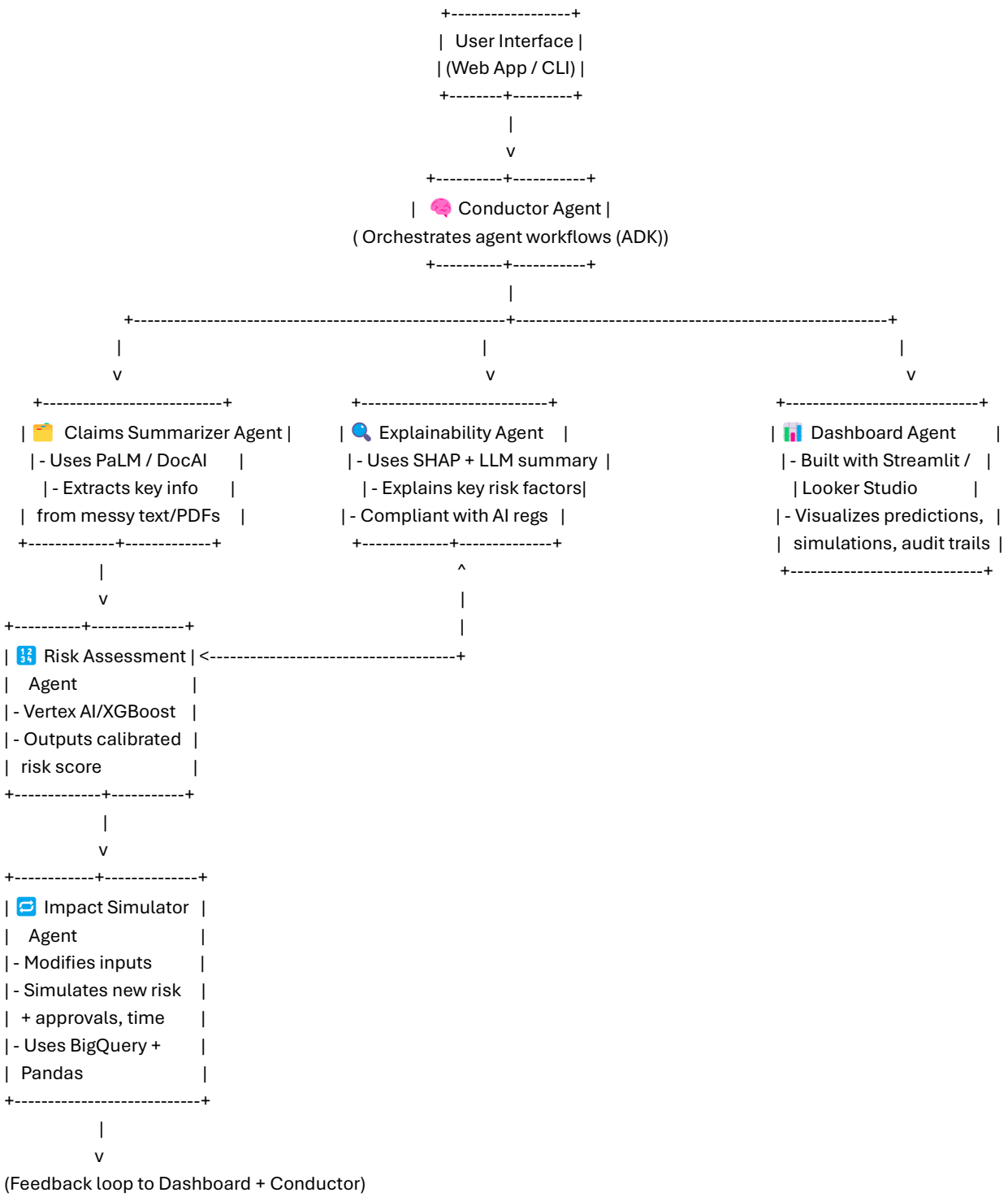
X-RAIL* (Explainable Risk Assessment & Insights Loop)



Component	Value Highlighted
Conductor Agent	Core to ADK – proves orchestration logic and autonomy
Summarizer Agent	Handles unstructured data (PDFs, text) – adds NLP strength
Risk Agent	Central risk prediction – ML model shows data-driven logic
Explainability Agent	Focus on **regulatory transparency**, SHAP, and UX via LLM
Simulator Agent	Demonstrates **autonomous scenario testing** – key differentiator
Dashboard Agent	Gives **control, transparency, and insight** to user

:

🧠 **Full Workflow: Trigger & Flow Logic**

Step 1: User Initiates Claim Assessment

- **Input:** A new insurance claim (structured form + unstructured text or PDF documents)
- **Triggered Agent:** 🧠 Conductor Agent
- **Role:** Central orchestrator. Determines which agents to activate and in what sequence.

Step 2: Extracting Claim Information

- **Triggered Agent:** 🗂️ Claims Summarizer Agent
- **Input:** Unstructured claim description (e.g., messy form text, scanned documents)
- **Processing:** Uses **DocAI / LLM** to extract key details like:
  - Incident type
  - Medical history
  - Geo information
  - Prior claim behavior
- **Output:** Clean, structured claim data
- **Sent To:** 🔢 Risk Assessment Agent

Step 3: Calculating Risk Score

- **Triggered Agent:** 🔢 Risk Assessment Agent
- **Input:** Structured data from summarizer (or direct user inputs)
- **Processing:** Uses **XGBoost / Vertex AI model** to predict:
  - Risk score (0 to 1)
  - Confidence level
- **Output:** { risk_score, confidence, features }
- **Sent To:**
  - 🔍 Explainability Agent for transparency
  - 📊 Dashboard Agent for real-time visualization

Step 4: Explaining the Risk

- **Triggered Agent:** 🔍 Explainability Agent
- **Input:** Processed data + risk score
- **Processing:** Uses **SHAP** to identify key features driving the risk score
  - Outputs top 3 factors

    - Also creates a **visual plot**
    - Adds compliance metadata (e.g., audit ID)
- **Output:** Plain-language explanation + visual + audit log
- **Sent To:**
    - 📊 Dashboard Agent
    - 🧠 Conductor Agent (for state tracking)

Step 5: Simulating Scenarios

- **Triggered Agent:** 🔁 Impact Simulator Agent (manual or auto-triggered)
- **Input:** Modified features (e.g., income increase, age, home location change)
- **Processing:** Runs **what-if analysis** on model + policy thresholds:
    - Recalculates approval rates
    - Updates portfolio risk metrics
- **Output:** Updated predictions, approval %, processing time impact
- **Sent To:** 📊 Dashboard Agent

Step 6: Real-Time Visualization

- **Triggered Agent:** 📊 Dashboard Agent
- **Input:** Risk score, explanation, simulation results
- **Processing:** Presents:
    - Risk score gauge
    - Top factors
    - SHAP plot
    - Approval simulator
    - Audit trail
- **Platform: Streamlit / Looker Studio** on GCP

🔁 **Optional Feedback Loop:**

- If simulation shows **high approval risk**, the user or Conductor can:
    - Modify inputs
    - Trigger the flow again via Summarizer → Risk Agent → Explanation

Why This:

- **Multi-agent orchestration** via Conductor (core ADK idea)
- **Explainability + simulation** stand out for judges
- **Clear separation of agents + intelligent collaboration**
- Shows both **user autonomy** and **regulatory accountability**