**Full System Flow for the Natural Language Query**

**Query:**  
*"Analyse recent procurement activity in the categories, to understand if there are any differences in the invoice discounts versus the contracted discounts."*

**🧩 1. Natural Language Query Received**

**UI → Prompt Engine**

* **Input**: User submits the question in natural language via UI
* **Captured Metadata**: User ID, context session ID, category filters (if any)

**🔄 2. Prompt Engine Activated**

* **Identifies Intent**: discount\_discrepancy\_detection
* **Extracts Entities**: category, invoice, contract, discount, timeframe
* **Calls → Context Manager** to fetch prior procurement session context, user permissions, and policy profile

**🧠 3. Context Manager**

| **Function** | **Action Taken** |
| --- | --- |
| Retrieve prior queries | Checks if this is part of a sourcing or supplier cycle |
| Load contract metadata context | Fetches relevant supplier contract discount terms |
| Apply user role filters | Ensures appropriate agent visibility based on user rights |
| Return context bundle to Orchestrator | Category filters, supplier list, historical engagement logs |

**🧠 4. Orchestrator Coordinates System Workflow**

| **Step** | **Agent/Module** | **Description** |
| --- | --- | --- |
| 1 | **Discrepancy Detection Agent** | Run logic on PO/invoice vs contract discount fields |
| 2 | **Opportunity Miner Agent** | Score the financial value of any mismatches |
| 3 | **Spend Analytics Agent** | Provide category-wise aggregated view of volumes and pricing |
| 4 | **Query Engine** | Generate dynamic graph query to fetch contract/invoice discount linkage |
| 5 | **Policy Engine Agent** | Run decision thresholds, scoring, and escalation policy |

**🧠 5. Query Engine Interacts with Knowledge Database**

| **Component** | **Functionality** |
| --- | --- |
| **Query Engine** | Generates and executes SPARQL/GraphQL queries to retrieve: |
|  | - Contracted discounts (from supplier contract entities) |
|  | - Invoice discount rates (from invoice nodes) |
|  | - Linked PO metadata and timestamps |
| **Procurement Knowledge Graph (Database)** | Semantic model with linked entities: Contracts, Suppliers, Invoices, Categories, Discounts |

**🔍 Sample Query Constructed**:

query {

discountDiscrepancies(category: "all") {

supplier

item

invoiceDiscount

contractDiscount

volume

priceVariance

contractId

invoiceId

}

}

Returned results are normalized and sent back to Opportunity Miner and Policy Engine.

**🧠 6. Policy Engine Applies Business Logic**

* Uses stored YAML policy rules
* Applies:
  + Contract Discount Enforcement Policy
  + Systematic Overpayment Policy
* Calculates:
  + Score = 8.3
  + Action = Escalate to buyer (Human-in-the-Loop Agent)
  + Estimated lost savings = £18,000

**🤖 7. Agent Actions Executed**

| **Agent** | **Action Taken** |
| --- | --- |
| **LLM Drafting Agent** | Drafts supplier pricing review email |
| **Human-in-the-Loop Agent** | Provides buyer with context, score, and options for action |
| **Audit Trail Agent** | Logs discrepancy details, agent response, and score |

**🖥️ 8. System Output to User**

* 📊 Visual comparison: contract discounts vs invoiced
* 💰 Missed savings totals
* 🧠 Opportunity score with “Action Required” flag
* 📄 Supplier email preview
* ✅ Buyer decision toggle (escalate, hold, reject)

**🧱 Summary of Core Module Roles**

| **Module** | **Role** |
| --- | --- |
| **Prompt Engine** | Translates NLQ into structured system instructions |
| **Context Manager** | Loads prior data context, permissions, filters |
| **Orchestrator** | Routes tasks, coordinates agents and services |
| **Query Engine** | Dynamically builds and executes queries to fetch structured facts |
| **Knowledge Graph** | Core data model linking contracts, invoices, suppliers, POs, pricing |
| **Policy Engine** | Applies decision rules, scores opportunities, determines actions |
| **Agents** | Detect, evaluate, escalate, communicate, and log insights |