AN AUTONOMOUS PROCESS OPTIMIZATION SYSTEM

OVERVIEW

As we head towards a more sufficient and productive future, it is also important that we improve our productivity in industrial sectors, making manufacturing and production more effective, accessible, and reliable. This project is a step toward a self-sufficient, productive environment where manufacturing and production can become fully autonomous.

This initiative leverages emerging technologies in **Artificial Intelligence (AI)** and automation to achieve this autonomy within a process plant in a refinery, specifically for a **Crude Distillation Unit (CDU)**.

The system involves the use of **Generative AI** technology, **AI agents**, and **Agentic AI system technology**. These agents will utilize tools such as a **machine learning forecast model**, event-driven software automation tools, and others yet to be developed.

This system will operate in a **simulated environment**, separate from the live plant. This allows engineers and operators alike to safely test ideas, gather better insights, or augment their information on plant processes.

CRUDE DISTILLATION UNIT (CDU) PLANT

A refinery processes crude oil, extracted from the depth of the earth by oil rigs, to produce useful and beneficial products for humans.

The **CDU plant** is the **first and central unit** to receive raw crude oil and break it down into its constituent products through a process called **distillation**. These products are then taken to other units for further refining. The overall running efficiency, productivity, and efficacy of the refinery are mostly dependent on the performance of the CDU, making it the central point as all other plant units take its products as feed for their systems.

This autonomous system aims to address several key challenges faced by CDU plants:

1. Enhancing the **explainability** of unaccounted energy and product losses encountered in plant processes.
2. Reducing **downtime** associated with retrieving product quality data needed for process optimization and adjustments.
3. Enabling **accurate adjustment** of product production rates based on market demand.

4. Identifying more **energy-efficient and safer operational strategies** or philosophies to attain plant goals.

THE PROCESS OPTIMIZATION SYSTEM

The system consists of 5 major components:

1. A **Forecast Model (TFT)**
2. A **Generative AI and Interactive Chatbot**
3. An **Operator Agent**
4. A **Safety Enforcing Agent**
5. A **Market Analysis Agent**

(Enhanced Diagram Description & Mermaid Code Below)

Diagram Interpretation:

The **User Interface** is the entry point for users. The **Generative AI Chatbot** acts as the central orchestrator. It interacts with:

- The **Forecast Model (TFT)** to get predictions.
- The **Plant DB** for historical data and documentation (via RAG).
- The **Agent Memory** for contextual information about ongoing tasks or past agent interactions.
- The **Market Analysis Agent**, **Operator Agent**, and **Safety Enforcing Agent**. These agents then interact with the **Simulated Environment** to perform tasks or gather information. The **Forecast Model** itself might also store/retrieve model files or configurations from a **Predictor File Storage**.

```
 graph TD

UI[User Interface] --> GAC[Generative AI Chatbot]


subgraph Data & Knowledge Sources

  PFS[Predictor File Storage]

  PDB[Plant DB / Documentation Vector DB]
```

```
    AM[Agent Memory]
end

GAC --- TFT[Forecast Model (TFT)]
TFT --- PFS

GAC --- PDB
GAC --- AM

subgraph Core Agents
    MAA[Market Analysis Agent]
    OA[Operator Agent]
    SEA[Safety Enforcing Agent]
end

GAC --> MAA
GAC --> OA
GAC --> SEA

MAA --> SE[Simulated Environment]
OA --> SE
SEA --> SE
OA --- TFT

style GAC fill:#lightblue,stroke:#333,stroke-width:2px
style TFT fill:#lightgreen,stroke:#333,stroke-width:2px
```

```
    style SE fill:#lightyellow,stroke:#333,stroke-width:2px
    style UI fill:#f9f,stroke:#333,stroke-width:2px
```

(You can copy and paste the Mermaid code above into a Mermaid live editor or compatible markdown renderer to see the visual diagram)

Infrastructure:

- **OS:** Windows OS with a few GPUs and sufficient CPUs.
- **Platforms:** AWS or integration with various open and closed platforms.
- **Frameworks (System-wide):** JAX, LangGraph, MCPs (Message Control Programs/Multi-Cloud Platforms - *clarification might be useful*), .NET.
- **Databases:** Vector DB, Unstructured Database.

FORECAST MODEL (TFT)

This machine learning model is used to predict or forecast two major aspects of CDU plant products: their **product flow/production rates** and **product quality**.

The quality forecast aims to **mitigate optimization downtime** resulting from delays in lab results, while the flow rate forecast provides insights into expected plant output during periods of adjustment.

The model employed is a **Temporal Fusion Transformer (TFT)**. Details on its rationale and usage are provided in the GitHub documentation. It ingests several process parameters to predict product flow rates and properties.

- **Key Technologies:**
    - **Environment:** Python
    - **Frameworks:** Dart (*Note: Dart is typically for UI; if used in backend ML, clarify its role. PyTorch Forecasting & Lightning are standard for TFTs.*), PyTorch Forecasting, PyTorch Lightning
    - **Deployment:** FastAPI
    - **Interfaces:** ETL API (for PoC); Kubernetes and Airflow (for production)

GENERATIVE AI CHATBOT

This is the primary AI model with which the user interacts to retrieve augmented information, query databases, or execute tasks. Possessing reasoning capabilities, it can:

- Invoke the **Forecast Model** for predictions.
- Call on **other agents** to perform required tasks in the **simulated environment**.
- Provide answers to plant-related queries based on retrieved information.
  All interactions are guided by company and industry standards, maintaining professionalism.
- **Key Technologies & Interfaces:**
  - **Environment:** Python
  - **Frameworks/Techniques:** LangChain, RAG (Retrieval Augmented Generation)
  - **Deployment:** FastAPI
  - **Interfaces:** Vector DB, MCP (for agent and automation tool communication), Web User Interface.

AI AGENTS

AI Agents are models trained to autonomously handle tasks using their reasoning abilities. They are invoked by the **Generative AI Chatbot** as needed.

- **Operator Agent:** This agent simulates the process panel operator's tasks during production. It aims to find more effective ways of implementing given tasks while optimizing the control philosophy. It also interacts with the **Forecast Model** to adjust production based on predictions.
- **Safety Enforcing Agent:** This agent handles the safety logic implementation for processes as needed. It collaborates with the **Operator Agent** to enforce safety procedures while executing tasks.
- **Market Analysis Agent:** This agent is exposed to market data (e.g., stock market information relevant to commodities) to analyze and provide feedback to the system on where production focus should shift based on demand, thereby improving the overall efficacy of the plant. It interacts with the **Operator Agent** to guide adjustments in its control procedures to attain products needed in the market.
- **Software Implemented Tools (for Agents in general):**

- **Environment:** Python
- **Frameworks:** CAG (Causal Analysis Graph / Cognitive Agent Graph - *clarification might be useful*), .NET. *Research is still ongoing for optimal frameworks.*
- **Deployment Mode:** FastAPI (for agent services)
- **Interfaces:** MCP. *Research on additional interfaces is ongoing.*

SIMULATED ENVIRONMENT

This is a **simulated environment** where the agents execute their given tasks. It will be **external to the live refinery system** but interfaced with the intelligent autonomous system.
It consists of refinery-like software tools, graphics, and HMIs, potentially leveraging open-source components (e.g., from Schneider Electric, if applicable and open-source). It may also include specific Windows applications and automation tools.

CONCLUSION

Most components are currently undergoing research and development. Updates will be communicated as they become available.

1. The specific **LLM for the AI agents is under consideration**, with **DeepSeek-R1** being a strong candidate due to its open nature and well-detailed post-training and fine-tuning documentation.
2. A **Proof of Concept (PoC) for the Forecast Model is ready**. An improved version with better accuracy and explainability will be developed in due time.
3. The specific **software platform for the overall system is still under evaluation**.
4. The type of **data lake or lakehouse architecture to be used is still uncertain**.

We hope to cover key milestones for this project in the next **4-5 months (approximately May - September)**:

- **Month 1:** Forecast Model ready and integrated.
- **Month 2:** Generative AI Chatbot tested, ready, and integrated.
- **Months 2-3 (overlapping):** AI Agents tested, ready, and integrated.
- **Month 4:** Simulated Environment setup, ready, tested, and interfaced.

- **Final Months:** Overall system testing, documentation, and preparation for production deployment.

Note (NT): We will be flexible in removing or adding features and components to the system as development progresses.

This revision should make your document clearer and more impactful. Let me know if you have any other questions or need further assistance!