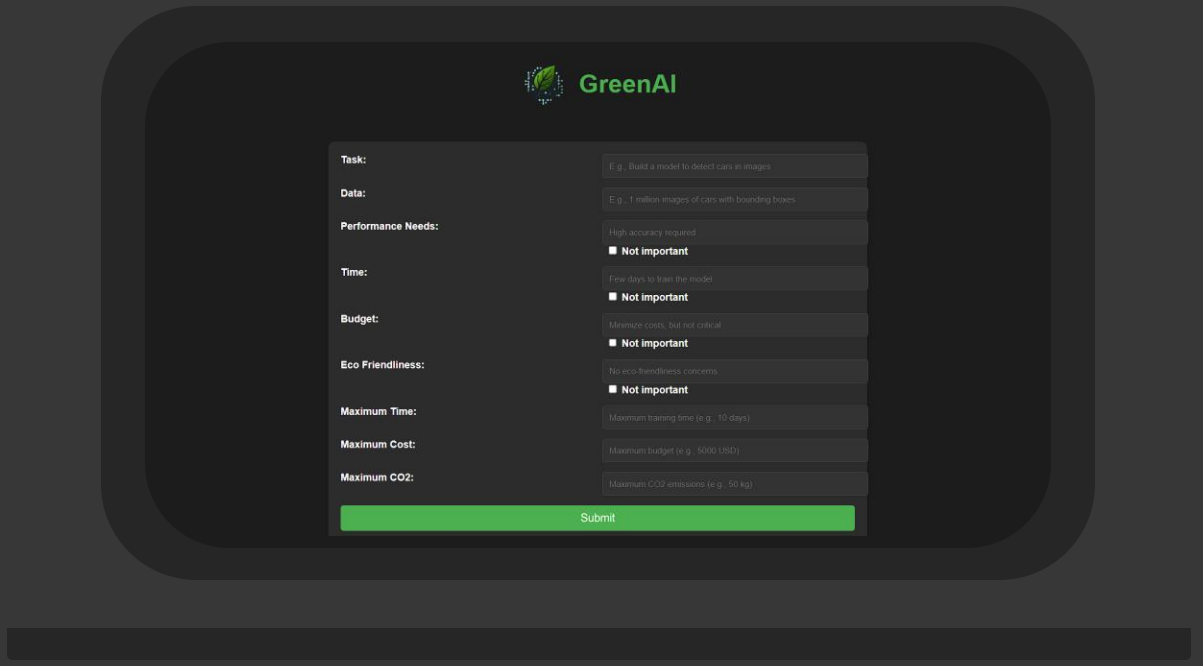
Project Presentation

GreenAI

GreenAI – Optimize your model, minimize your impact



The Team

AI Engineer
Illia Rohalskyi

Web/Software-Dev
Oskar Herkt

Agenda

1. **Problem Statement**

   Defining the challenges in AI training, including performance, cost, and environmental impact.

2. **Solution Overview**

   An outline of how GreenAI optimizes model training by balancing eco-friendliness, time, and cost.

3. **AI Implementation**

   Overview of GreenAI's mechanisms, focusing on input, output, and the roles of nodes, edges, and conditional edges.

4. **UI and User Workflow**

   A quick look at the interface that allows users to interact with and configure GreenAI’s recommendations.

5. **Questions & Answers**

   Open the floor to address any questions and provide clarity on the discussed topics.

Problem Statement

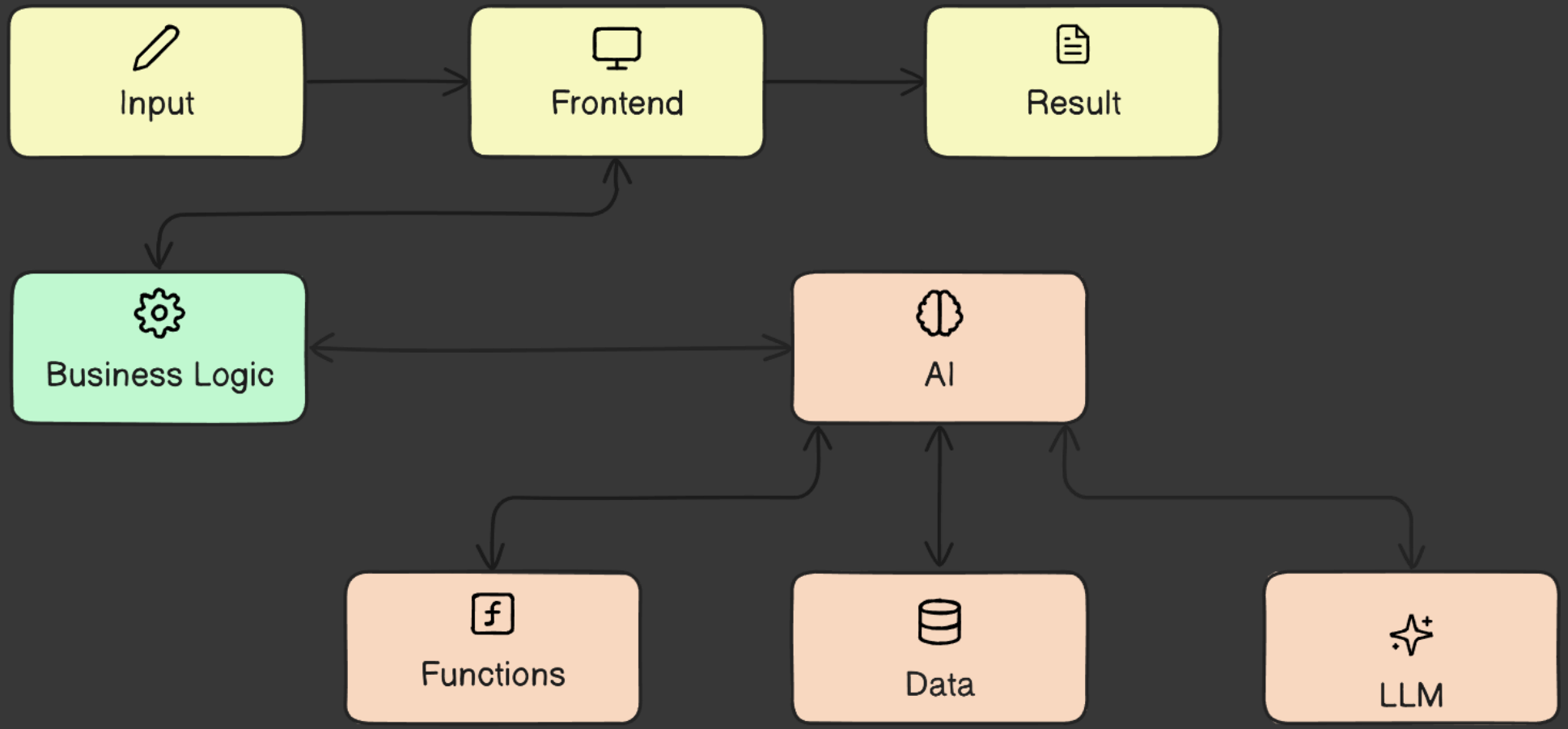
- **Environmental Impact** of AI Training
- **Cost Management** in AI Development
- **Time Constraints** in Model Training
- Lack of **Tailored Recommendations** for AI Model

Training

- Data-Driven Decision Making:

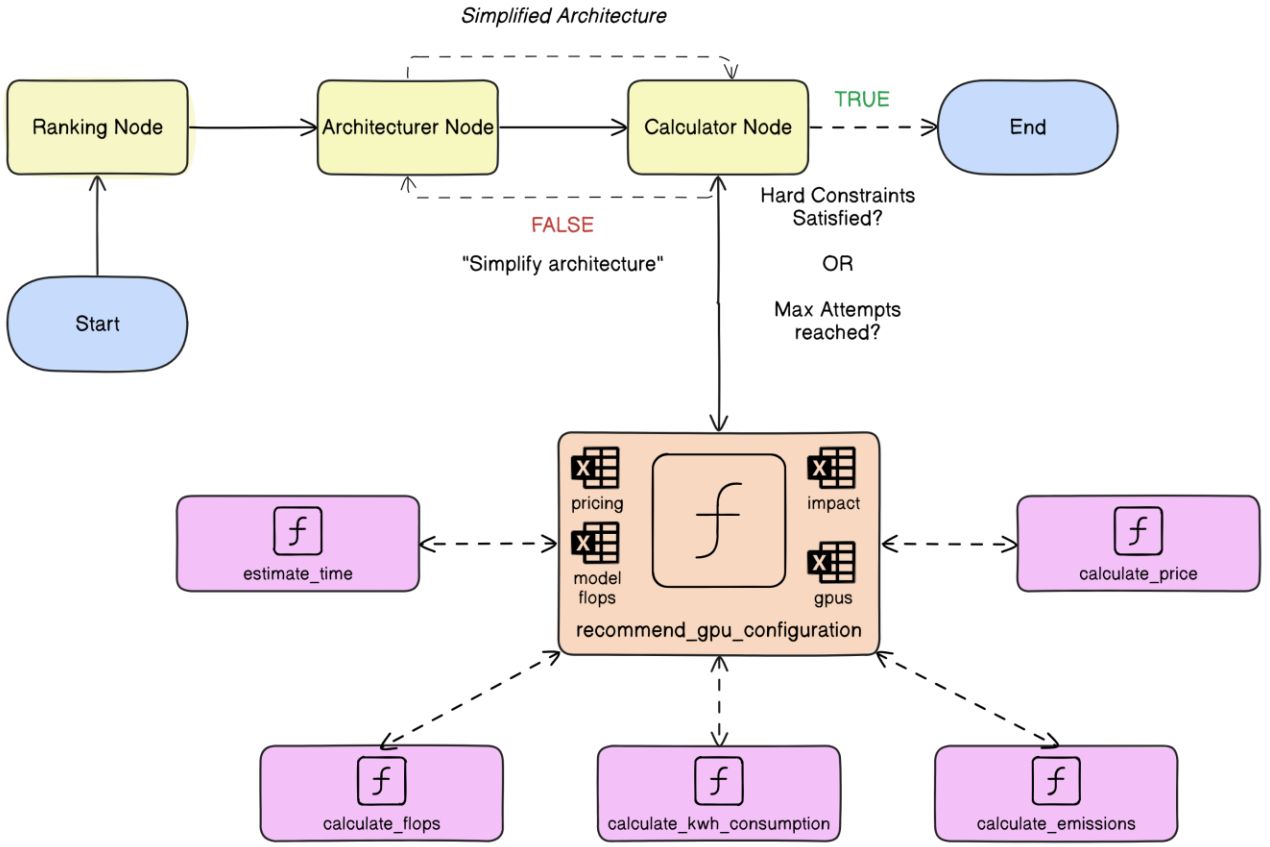
1

Solution Overview



The AI Workflow

- ***Nodes***: Decision points that process input data and generate recommendations.
- ***Edges:*** Connections between nodes that direct the flow of information.
- ***Conditional Edges:*** Allow iterative refinement by revisiting nodes if constraints are not met.
- ***State:*** Represents the current context and information throughout the decision-making process, including tasks, priorities, and constraints.



Soft and Hard Constraints

Soft Constraints:

- ***Task Description:*** Objective for the model/system.
- ***Data Description:*** Dataset size, format, and preprocessing.
- ***Performance Needs:*** Desired accuracy level.
- ***Time Constraint:*** Expected completion time.
- ***Budget Constraint:*** Financial limitations for the solution.
- ***Eco-Friendliness:*** Environmental impact requirements.

Hard Constraints:

- ***Maximum Time:*** Absolute time limit for training.
- ***Maximum Cost:*** Financial limit that must not be exceeded.
- **Maximum CO2 Emissions:** Strict carbon footprint limit.

Data Sources

1. **model_flops.xlsx**

   Information on various AI models and their Floating Point Operations (FLOP) requirements.

2. **GCP gpus pricing.xlsx**

   Pricing details for different GPU configurations on Google Cloud Platform.

3. **gpus.csv**

   Performance data for GPUs, including Tera Floating Point Operations Per Second (TFLOPS) and Thermal Design Power (TDP).

4. **impact.csv**

   Region-specific carbon intensity data for emissions calculations.

Conditional Edge and Iterative Refinement

How It Works:

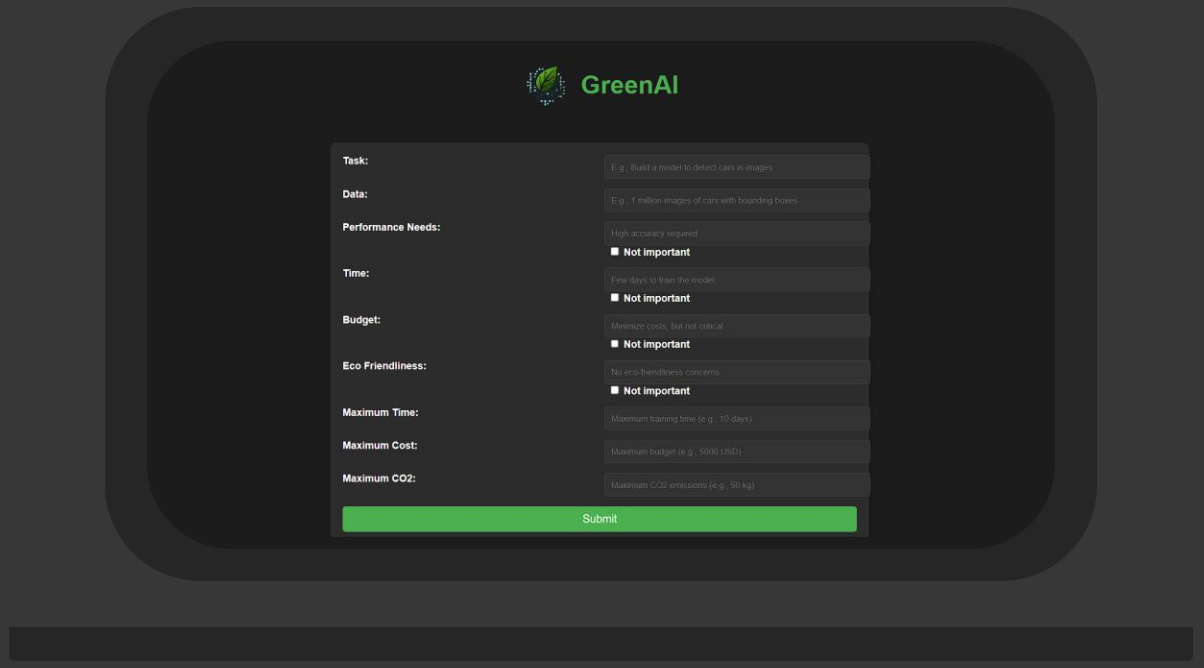
- ***Conditional Edge Mechanism:*** Evaluates if constraints are met and if simplification attempts are within the limit.
- ***Logic:***
    - ***True:*** Revisit architecture recommendation.
    - ***False:*** End the process.

Purpose:

- Re-evaluates architecture recommendations to fit hard constraints (time, cost, CO2-emission).
- Allows up to **three iterations** for a feasible solution.

```
def should_simplify(state: MainState):
  return not state.constraints_met and state.simplification_attempts < 3
builder.add_conditional_edges(
  "calculator",
  should_simplify,
  {
    True: "architecturer",
    False: END
  }
```

Live Demo


GreenAI
Task:
E.g., Build a model to detect cars in images
Data:
E.g., 1 million images of cars with bounding boxes
Performance Needs:
High accuracy required
Not important
Time:
Few days to train the model
Not important
Budget:
Minimize costs, but not critical
Not important
Eco Friendliness:
No eco-friendliness concerns
Not important
Maximum Time:
Maximum training time (e.g., 10 days)
Maximum Cost:
Maximum budget (e.g., 5000 USD)
Maximum CO2:
Maximum CO2 emissions (e.g., 50 kg)
Submit

Questions & Answers