High-Level Project Overview: From Contracts to a Knowledge Base

Our project aims to solve a significant challenge: the manual extraction of key information from complex contracts. By transforming static, difficult-to-read PDFs into a dynamic, queryable knowledge base, we can automate the validation of labor rules, saving time and ensuring accuracy.

Establishing a Common Standard for Rules

To ensure the project's success, we've established a shared definition for what constitutes a "rule." Each rule we extract will adhere to the following standards:

- **Source:** The exact location of the rule, including the document, page number, and clause.
- **Rule Excerpt:** The original, raw text of the rule itself, which serves as the **source of truth**.
- **Rule Meaning:** A simple, human-readable summary of the rule for end-users.
- **Machine-Readable Format:** A structured, **JSON-based version** of the rule that our system can interpret and use for validation.

In addition to these core components, we'll need to define a list of common **metadata** for all rules. This could include categories like **labor union, or conditions** etc. to help us organize and query the rules more effectively.

Building a "Rule Graph" Knowledge Base

Instead of a simple database, our knowledge base will be a **"rule graph"**, a system where rules are connected to other relevant data points. This structure allows us to answer complex questions that a simple keyword search can't.

At its core, the knowledge base will be a collection of rule objects. Each object will contain essential information, including its **source**, **page number**, **rule excerpt**, **rule meaning**, and **machine-readable format**.

The key to this system is the ability to create connections:

- **Linking Rules to Entities:** We'll connect rules to the specific people, companies, or equipment mentioned within them. For example, a rule about orientation for Boilermakers could be tagged as applicable to both **Boilermakers** and a specific customer like **NWR**.
- **Connecting Rules to Other Rules:** Our system will link rules that reference or depend on each other, even across different contracts. This allows us to automatically trace a rule's lineage and understand its full context.

Our Collaborative Plan: A Multi-Phase Sprint

We are approaching this project with a phased, collaborative strategy to ensure we build a robust and reliable system.

Phase 1: Defining Ground Truth & Document Preparation

We've already started by manually extracting 24 "ground truth" examples from our contracts to serve as our gold standard. The next crucial step is converting all contract PDFs into a structured text format that a large language model (**LLM**) can understand. We'll use a combination of software libraries to handle this significant challenge, which is a key hurdle due to the complex formatting and tables found in these documents.

Phase 2: LLM Automation and Proof of Concept

This is our primary technical sprint, where we'll use a few key techniques with large language models (LLMs) like Gemini and OpenAI.

- **Few-Shot Prompting**: This is our main approach. We'll provide the LLM with a few perfect examples of the desired output. This teaches the model exactly what we're looking for, significantly improving accuracy.
- **General Prompting**: We'll also use high-level commands to see how the LLM performs on its own.

Our goal is to create a proof of concept where the LLM can automatically generate both the human-readable summary and the structured JSON for a given rule excerpt. We'll start with simple rules and gradually increase the complexity to include full-page rules, tables, and inter-contract dependencies.

Phase 3: Validation and the Human-in-the-Loop

Once we have our automated results, we'll begin the crucial validation process.

- We will compare the **LLM's output** against our 24 manually extracted examples to measure its accuracy.
- **Subject Matter Experts** will act as our "human-in-the-loop," cross-validating the LLM's results. This feedback is essential for continuous improvement and for building a reliable system that everyone on the team can trust.

This plan ensures we are not just building an automated system, but one that is validated, trusted, and constantly improving, a truly active system, not a static list.