The Advanced Research Center (ARC) A Technology Research Publication Issue 6/2026

The ARC Journal

Published by the Engineering Association Yerevan, August 21, 2026

ARC-LAB: Physical AI and Smart Manufacturing

IB-ARC Industrial-Based Project

ARC Project ID [ARC-IB-2608-01]

Paper Title	**AI-Based Production Master BOM Analysis, Verification, and Validation Assistant**
Project Leader: Narbeh Nahapetian	Authors (in publication order): Ashot Arzumanian

Project Base Information	
Primary Proposal Submitted Date:	Paper Submission Date:
ARC Category: ☐ EB-ARC ☑ IB-ARC ☐ RB-ARC	Duration:
Sponsors (if any): -	Project Assets Available: ☑ Yes ☐ No
Team Members: Ashot Arzumanian	POC Developed: ☑ Yes ☐ No
	Project Type: ☑ Process Improvement ☐ Automation ☑ AI/ML ☐ Manufacturing ☐ Robotics ☐ Hybrid ☐ Embedded Systems ☑ Procurement

Abstract

Production Master Bills of Materials (BOMs) are fundamental to modern manufacturing, procurement, and inventory management. However, inconsistencies between manufacturer part numbers (MPNs) and their corresponding component descriptions frequently occur due to manual data entry, incomplete documentation, or human error. Such inconsistencies may lead to incorrect component selection, increased engineering verification effort, procurement delays, and reduced data quality throughout the manufacturing process.

This project presents the design, development, and validation of an AI-based verification agent capable of automatically determining whether a component description corresponds to the component identified by its manufacturer part number. The developed system classifies each MPN–description pair into one of three categories: *Match*, *Partial Match*, or *No Match*, enabling engineers to rapidly identify incorrect or incomplete BOM entries while encouraging standardized component descriptions.

The solution was implemented as a proof-of-concept using an AI agent developed in the n8n automation platform, with GPT-based Large Language Models (LLMs) serving as the reasoning engine. A complete workflow was created to retrieve MPN-description pairs, execute AI-based analysis, measure execution performance, and return validation results through an Excel-based user interface. A structured validation dataset consisting of representative components with intentionally modified descriptions was prepared to evaluate the system under realistic manufacturing scenarios.

A significant portion of the research focused on prompt engineering rather than model fine-tuning. Multiple prompt versions were systematically designed, tested, and refined by modifying reasoning instructions, decision rules, output constraints, comparison logic, and model parameters. Experimental evaluation compared different GPT models and

prompt strategies, demonstrating how carefully structured prompts substantially improve consistency, reasoning quality, output reliability, and classification accuracy while reducing response variability.

The completed system demonstrates that modern LLMs can effectively support intelligent BOM verification without requiring domain-specific model training. The project establishes a practical methodology for integrating AI agents into engineering workflows and provides a foundation for future extensions, including automatic description generation, component standardization, MRP (Material Requirements Planning) integration, and broader AI-assisted manufacturing data validation.

Keywords
Artificial Intelligence (AI), Large Language Models (LLMs), GPT, Prompt Engineering, Bill of Materials (BOM), Manufacturer Part Number (MPN), Description Validation, Data Verification, Smart Manufacturing, Manufacturing Automation, AI Agent, n8n, Engineering Data Quality.

Executive Summary

Production Master Bills of Materials (BOMs) serve as the foundation of manufacturing by defining every component required to build a product. The accuracy of BOM data directly influences engineering, procurement, inventory management, production planning, and product quality. One of the most common data quality issues is inconsistency between a Manufacturer Part Number (MPN) and its corresponding component description. Such inconsistencies may occur due to manual data entry, incomplete documentation, copying descriptions from similar components, or simple human error. Although these mistakes may appear minor, they can increase engineering verification effort, complicate procurement processes, and introduce the risk of selecting incorrect components.

The objective of this project was to design, develop, and validate an AI-based verification agent capable of automatically analyzing the consistency between Manufacturer Part Numbers (MPNs) and their corresponding component descriptions within Production Master BOMs. Instead of relying on manually developed verification rules for thousands of different component families, the proposed solution leverages the reasoning capabilities of GPT-based Large Language Models (LLMs) to analyze the specifications associated with a manufacturer part number and compare them with the supplied description.

A proof-of-concept AI verification agent was designed and developed using the n8n automation platform. GPT-based language models were used for the AI agent. The data (case study manufacture part number and description pairs, validation data, notes, history) was stored in Google Sheets. Excel interface intended for engineering use. The system accepts an MPN and its corresponding description as input, determines the specifications of the referenced component, compares those specifications with the provided description, and classifies the pair as *Match***,** *Partial Match***,** or *No Match*.

The research focused primarily on prompt engineering and systematic optimization of AI reasoning. Rather than considering the language model as a fixed solution, multiple prompt versions were designed, experimentally evaluated, and iteratively refined. Each iteration introduced modifications to reasoning instructions, comparison rules, output constraints, decision logic, and model parameters. Performance was measured using a controlled validation dataset containing representative MPN-description pairs ranging from fully correct to intentionally incomplete and incorrect. This iterative methodology made it possible to identify prompt structures that significantly improved reasoning consistency, reduced output variability, and increased classification reliability.

The completed project demonstrates that modern LLMs can effectively assist engineering verification tasks and the overall Smart Manufacturing development without requiring custom model training. Beyond the developed proof-of-concept, the project establishes a practical methodology for designing AI agents capable of validating engineering data in manufacturing environments. The presented approach can serve as a foundation for future integration with Material Requirements Planning (MRP) systems, automatic description generation, BOM standardization, and broader AI-assisted manufacturing data validation.

Industrial Problem Definition

Each BOM entry typically contains a Manufacturer Part Number (MPN), which uniquely identifies a component supplied by a manufacturer, together with a textual description summarizing its most important characteristics. Engineers, procurement specialists, manufacturing personnel, inventory managers, and other stakeholders rely on both pieces of information throughout the product lifecycle.

In practice, the consistency between the MPN and its corresponding description cannot always be guaranteed. Component descriptions are frequently entered manually, copied from previous projects, shortened for convenience, or modified without complete verification. As a result, descriptions may omit important specifications, contain outdated information, or even describe a different component than the referenced part number. These inconsistencies reduce the quality of engineering data, increase the likelihood of human error, and require additional manual verification before procurement or production activities can continue.

Traditional verification methods are largely manual and require engineers to search manufacturer documentation or technical databases in order to compare the specifications associated with an MPN against its written description. This process is repetitive, time-consuming, and becomes increasingly impractical as Production Master BOMs grow to hundreds or thousands of components. Furthermore, conventional rule-based validation software is difficult to develop and maintain because different component categories have different specification structures, naming conventions, and description formats. Developing deterministic validation rules that cover all possible component types requires significant engineering effort and continuous maintenance.

Recent advances in Large Language Models (LLMs) provide an alternative approach. Unlike traditional rule-based systems, LLMs are capable of understanding technical language, extracting relevant specifications from technical sources, comparing contextual information, and reasoning across a wide variety of component categories. This creates an opportunity to automate a significant portion of BOM verification while maintaining flexibility across different manufacturers, component families, and description styles.

The industrial problem addressed by this project is the automatic verification of consistency between Manufacturer Part Numbers and their corresponding descriptions within Production Master BOMs. The desired solution must reliably distinguish between three situations: descriptions that correctly represent the referenced component (*Match*), descriptions that contain no conflicting information but require additional detail (*Partial Match*), and descriptions containing one or more conflicting specifications (*No Match*). In addition to accurate classification, the solution should produce deterministic outputs, minimize response variability, maintain reasonable execution time and operational cost, and remain sufficiently generic to support a broad range of component categories without requiring component-specific programming.

Successfully addressing this problem has the potential to reduce engineering verification effort, improve BOM data quality, support procurement processes, encourage standardized component descriptions, and establish a foundation for more intelligent AI-assisted manufacturing systems.

Existing/Background and Current State

Production Master Bills of Materials (BOMs) are essential engineering documents used throughout the manufacturing lifecycle. They define the components required to produce a product and serve as a common source of information for engineering, procurement, inventory management, production planning, and quality assurance. Each BOM entry typically includes a Manufacturer Part Number (MPN) together with a textual description that summarizes the component's primary characteristics. Maintaining consistency between these two fields is essential for ensuring reliable engineering data and efficient manufacturing operations.

In many manufacturing organizations, verification of BOM data remains a largely manual process. Engineers and procurement specialists compare manufacturer part numbers with supplier catalogs, technical databases, or manufacturer documentation to confirm that the recorded description accurately represents the referenced component. When inconsistencies or incomplete descriptions are identified, they must be corrected manually before the BOM can be confidently used in downstream processes. Although effective for small datasets, this approach becomes increasingly time-consuming and resource-intensive as the size and complexity of Production Master BOMs continue to grow.

Various software solutions have been developed to improve BOM management by providing functions such as duplicate detection, data standardization, lifecycle management, and integration with Enterprise Resource Planning (ERP) and Material Requirements Planning (MRP) systems. However, these systems generally depend on structured databases, predefined validation rules, or exact field matching. While suitable for deterministic verification tasks, they often have limited capability to reason about the semantic relationship between a manufacturer part number and a free-text component description, particularly when descriptions are incomplete, abbreviated, or written using different terminology.

Traditionally, expanding automatic verification has required developing rule-based systems that explicitly define how different component categories should be analyzed. Such systems typically require extensive engineering effort because each component family may contain different specification sets, naming conventions, and description

formats. As new manufacturers, products, and description styles are introduced, these rules must be continuously updated and maintained, making the approach increasingly difficult to scale.

Recent advances in Artificial Intelligence, particularly Large Language Models (LLMs), have introduced new opportunities for engineering data validation. Unlike conventional rule-based approaches, LLMs are capable of interpreting technical language, understanding contextual relationships, extracting relevant specifications, and reasoning about whether two pieces of information describe the same physical component. These capabilities enable AI systems to perform validation tasks that would otherwise require numerous manually defined rules while remaining adaptable to different component categories and description styles.

Despite these advances, applying LLMs within manufacturing environments presents several practical challenges. AI models may produce inconsistent responses, interpret technical specifications differently across executions, hallucinate, or generate outputs that do not conform to the strict formats required by automated workflows. Furthermore, response time, operational cost, prompt complexity, and reasoning reliability become important considerations when integrating LLMs into industrial processes. Consequently, successful implementation requires not only selecting an appropriate language model but also designing deterministic workflows, carefully engineered prompts, and strict validation methodologies to ensure dependable operation.

The current project addresses these challenges by developing an AI-based verification agent specifically designed for Production Master BOM validation. Rather than relying on extensive component-specific programming, the proposed solution employs systematic prompt engineering and controlled workflow design to produce reliable *Match*, *Partial Match*, and *No Match* classifications. The project therefore investigates how modern LLMs can be transformed from general-purpose language models into dependable engineering assistants capable of supporting BOM verification within manufacturing environments.

Project Objectives

The primary objective of this project was to design, develop, and validate an AI-based verification agent capable of automatically analyzing the consistency between Manufacturer Part Numbers (MPNs) and their corresponding component descriptions within Production Master BOMs. The proposed system was intended to replace manual verification effort by identifying incorrect, incomplete, or inconsistent BOM entries while encouraging standardized component descriptions.

To achieve this objective, the project focused on the following goals:

- Design a proof-of-concept AI agent capable of receiving an MPN and its corresponding description as input and automatically determining whether the description accurately represents the referenced component.
- Develop a consistent and reliable classification methodology that categorizes each MPN-description pair into one of three predefined outcomes: *Match*, *Partial Match*, or *No Match*.
- Investigate prompt engineering techniques that improve the reasoning capabilities, consistency, and reliability of GPT-based Large Language Models when performing technical engineering validation tasks.
- Evaluate different prompt versions, reasoning strategies, and model configurations using a controlled validation dataset in order to identify the most reliable solution.
- Throughout the project development process achieve deterministic outputs, while maintaining the acceptable execution time, and practical operational costs, in order to make the application of Large Language Models to engineering data verification appropriate.

The overall objective was not only to develop a functional proof-of-concept, but also to establish a structured methodology for designing AI agents capable of supporting engineering verification processes within Smart Manufacturing environments.

Requirements Definition

The requirements for the proposed AI-based verification agent were derived directly from the industrial problem identified in the previous sections. Since the objective of the project was to develop a proof-of-concept capable of supporting Production Master BOM verification, the system was required not only to perform the desired verification task but also to produce reliable, repeatable, and practically applicable results suitable for engineering environments. The defined requirements guided both the architecture of the solution and the iterative prompt engineering process throughout the project.

Functional Requirements

The system shall:

- Accept a Manufacturer Part Number (MPN) together with its corresponding component description as the primary input.
- Determine the specifications associated with the supplied MPN without modifying or decoding the original manufacturer identifier.
- Compare the extracted component specifications with the supplied description and evaluate their consistency.
- Classify every analyzed MPN-description pair into exactly one of three predefined categories:
    - *Match* – the description correctly represents the referenced component.
    - *Partial Match* – the description contains no conflicting information but requires further completion or additional specification.
    - *No Match* – one or more specifications conflict with the component identified by the MPN.
- Return the classification using a deterministic output format suitable for further automated processing.
- Process multiple MPN-description pairs through an automated workflow without requiring manual intervention during execution.

Non-Functional Requirements

The system should:

In addition to the required functionality, the system was expected to satisfy several quality requirements necessary for practical industrial use.

The verification process should produce consistent and reliable classifications, minimizing unnecessary output variability when identical inputs are analyzed repeatedly. Although Large Language Models are inherently probabilistic, the developed prompt engineering methodology should ensure that the generated classifications remain stable and within predefined engineering-acceptable outcomes.

The solution should maintain reasonable execution time so that verification can be performed efficiently on large Production Master BOMs without introducing excessive delays into engineering workflows.

Operational efficiency was another important consideration. The workflow should minimize token consumption and API costs while maintaining verification quality, making the solution economically suitable for repeated industrial use.

The developed architecture should remain generic and scalable, allowing the same verification methodology to be applied across different component categories without requiring manually developed validation rules for each category.

Finally, the solution should employ a modular workflow architecture, allowing individual components—including prompts, AI models, input sources, and output mechanisms—to be modified or replaced independently as future improvements are introduced.

Research Requirements

In addition to developing a functional verification system, the project aimed to identify the most effective approach for AI-based BOM validation. Therefore, the developed methodology needed to support systematic testing of different prompt versions, reasoning strategies, and GPT models.

Each solution was evaluated using the same validation dataset, allowing its classification accuracy, response consistency, execution time, and operational efficiency to be compared under identical conditions. This iterative evaluation process ensured that the final verification agent was developed through structured optimization and experimental validation rather than relying on a single prompt implementation.

Proposed Solution

The proposed solution is an AI-based verification assistant designed to automatically analyze the consistency between Manufacturer Part Numbers (MPNs) and their corresponding component descriptions within Production Master Bills of Materials (BOMs). The primary purpose of the system is to support engineers during BOM preparation and verification by identifying inconsistent, incomplete, or incorrect descriptions while reducing the amount of manual validation required. This directly addresses the project objective of developing an intelligent assistant for analyzing, verifying, and validating Production Master BOM data.

Unlike conventional rule-based validation systems, the proposed solution utilizes the reasoning capabilities of Large Language Models (LLMs). Instead of manually creating and maintaining validation rules for thousands of different component families, the system interprets technical information associated with a Manufacturer Part Number and determines whether the supplied description accurately represents the referenced component. This enables a single verification methodology to be applied across a broad range of component categories without requiring component-specific programming.

The verification process is implemented as a sequential two-agent AI workflow, where each agent is responsible for a distinct reasoning task. The first AI agent is responsible for extracting and structuring the technical specifications associated with the submitted Manufacturer Part Number. Using reliable technical sources, it generates a canonical representation of the component, which serves as the reference specification set for the verification process.

The second AI agent receives both the extracted specifications and the original component description. Its task is to compare the structured specifications with the provided description and determine whether the description accurately represents the identified component. By separating specification extraction from validation, the system reduces reasoning complexity and improves consistency in classification decisions.

Based on the comparison results, the second AI agent returns one of three predefined engineering classifications:

- *Match* – the description correctly represents the component identified by the Manufacturer Part Number and contains no conflicting specifications.
- *Partial Match* – the description does not contain conflicting information but is incomplete or requires additional specifications to meet engineering documentation standards.
- *No Match* – one or more specifications contained in the description conflict with the specifications of the component identified by the Manufacturer Part Number.

These three classifications were defined at the beginning of the project and were used consistently throughout the prompt development and validation process. For several validation cases, more than one classification was considered acceptable because the descriptions could reasonably be interpreted in different ways without violating the engineering intent. Consequently, the validation methodology allowed predefined alternative acceptable outputs for selected test cases, enabling objective evaluation of the developed AI system.

The proposed solution is intended to operate as an engineering assistant rather than an autonomous decision-making system. The AI workflow performs the verification task and returns the appropriate classification, allowing engineers to quickly identify BOM entries that require review or description improvement. In future industrial deployment, this workflow can be integrated into existing engineering software so that validation becomes part of the normal BOM preparation process. The original project proposal identifies integration with ERP/MRP systems as a possible future development beyond the scope of this proof-of-concept.

To support practical development and evaluation, the proof-of-concept implementation was built using a workflow automation environment, where both AI agents operate sequentially within a single orchestrated process. The validation dataset used for testing was stored in a structured spreadsheet format, enabling systematic execution of experiments and comparison of results. Additionally, a Microsoft Excel interface was used as a user-facing tool to demonstrate how engineers could interact with the system in a future deployment scenario. These tools were used exclusively for prototyping, testing, and evaluation purposes and do not represent mandatory components of the final production architecture.

The proposed solution therefore consists of two tightly connected AI reasoning components operating in sequence: the first agent responsible for specification extraction, and the second agent responsible for validation and classification. Together, they form a structured AI verification workflow that enables consistent, scalable, and engineering-aligned assessment of Production Master BOM data.

System Design

The system was designed as a modular AI-assisted verification solution for validating the consistency between Manufacturer Part Numbers (MPNs) and their corresponding component descriptions within Production Master Bills of Materials (BOMs). The primary objective of the design was to create a scalable verification methodology that does not rely on manually defined rule sets for individual component categories but instead leverages the reasoning capabilities of Large Language Models (LLMs) to evaluate technical consistency between structured and unstructured engineering data.

The final system design is implemented as a two-agent AI workflow orchestrated using the n8n automation platform, which enables structured execution of sequential reasoning steps, API communication with LLM models, and data exchange between system components. This workflow-based design ensures reproducibility of experiments and allows full control over prompt execution, model selection, and data logging during development and evaluation.

Overall System Design

The developed verification system consists of two AI agents executed sequentially within a single automated workflow implemented in n8n. The workflow acts as the central orchestration layer, responsible for managing data flow between system components, triggering API calls to the LLM, and storing intermediate and final results.

The verification process begins when a Manufacturer Part Number (MPN) and its corresponding component description are provided as input. These inputs are retrieved from the validation dataset stored in Google Sheets and passed into the n8n workflow for processing.

AI Agent 1 – Data Extraction

The first AI agent is responsible for retrieving and structuring technical specifications associated with the provided MPN. Its role is not to perform validation, but to find relevant and real specifications in the open sources and transform unstructured or semi-structured information into a standardized specification format.

The output of this agent is a canonical representation of the component, which includes key technical attributes such as electrical, mechanical, or functional properties depending on the component type. This structured output serves as the reference baseline for the second stage of the system.

AI Agent 2 – Description Validating

The second AI agent receives two inputs:

- Structured component specifications generated by AI Agent 1
- The original component description provided in the dataset

Using these inputs, the agent performs a semantic and technical comparison between the extracted specifications and the provided description. The goal is to determine whether the description accurately reflects the real-world component defined by the MPN.

Based on this evaluation, the agent assigns one of three predefined engineering classifications:

- Match – The description fully corresponds to the extracted specifications.
- Partial Match – The description is generally correct but incomplete or missing key attributes, requires improvement.
- No Match – The description contains incorrect or conflicting technical information.

Separating the system into two independent reasoning agents significantly improved reliability, as each agent performs a single well-defined task rather than combining retrieval and validation in one step.

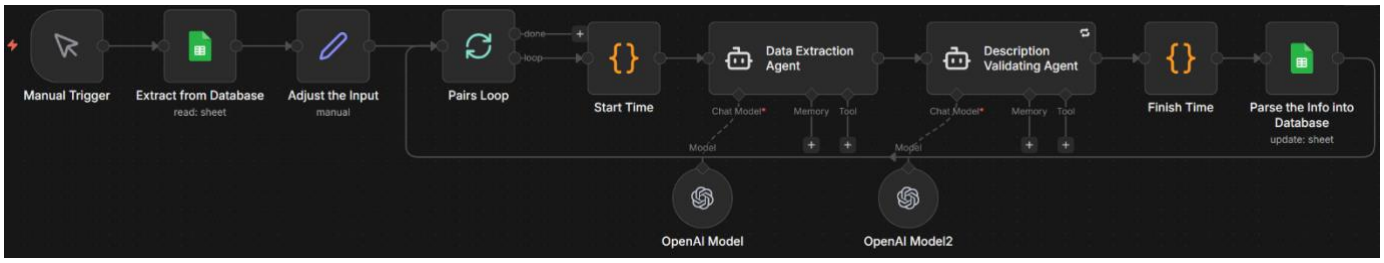
n8n Workflow Execution & System Architecture

The entire two-agent system is implemented and executed using the n8n workflow automation platform, which serves as the operational backbone of the prototype.

This workflow-based implementation enables full traceability of each experiment, including prompt versioning, model configuration, execution time, and token usage. It also allows rapid iteration during development, as modifications to prompts or model settings can be applied directly within the workflow without changing the underlying system architecture.

The logical execution of the system is implemented in n8n and is shown in Figure 1.

Figure 1- n8n workflow implementing the two-agent AI verification system



The workflow visually represents the full execution pipeline, node-by-node:

1. "Manual Trigger" - starts execution when clicking on it.
2. "Extract from Database" - gets the description-part number pairs from the document.
3. "Adjust the Input" - leaves only description-part number pairs for the future processing.
4. "Pairs Loop" - loops over description-part number pairs.
5. "Start Time" - sets a time when the execution started.
6. "Data Extraction Agent" - retrieves technical specification of the component.
    6.1. "OpenAI Model" - the AI model itself and its settings of "Data Extraction Agent".
7. "Description Validating Agent" - compares canonical specs from the previous agent with the description and returns the final label: Match | Partial Match | No Match.
    7.1. "OpenAI Model2" - the AI model itself and its settings of "Description Validating Agent".
8. "Finish Time" - outputs execution time.
9. "Parse the Info to Database" - parses in the database the final label and the time of execution.

This implementation ensures that the theoretical two-agent architecture is directly mapped to a reproducible and testable automation system.

Validation Dataset

The system relies on a structured validation dataset stored in Google Sheets, which serves as the central data repository during development and testing.

Each row in the dataset represents a single verification case consisting of a Manufacturer Part Number and its corresponding component description. Additional metadata fields are included to support evaluation and performance tracking, as it is shown on Figure 2.

Figure 2 - Google Sheets Validation Dataset Used for System Evaluation

MPN	Description	Optimal Description	Desired output		Actual Output	Time
CL31B225KBHNNNE	2.2	2.2 µF ±10% 50V Ceramic Capacitor X7R 1206 (3216 Metric)	Partial Match	No Match	No Match	4s 730ms
CL31B225KBHNNNE	2.2 µF	2.2 µF ±10% 50V Ceramic Capacitor X7R 1206 (3216 Metric)	Partial Match	-	Partial Match	7s 602ms
CL31B225KBHNNNE	2.2 50V	2.2 µF ±10% 50V Ceramic Capacitor X7R 1206 (3216 Metric)	Partial Match	-	Partial Match	5s 141ms
CL31B225KBHNNNE	2.2 µF ±10% 50V Ceramic Capacitor X7R 1409 (3216 Metric)	2.2 µF ±10% 50V Ceramic Capacitor X7R 1206 (3216 Metric)	No Match	Partial Match	No Match	5s 168ms
PA2512FKF070R001E	RES 0.001 OHM 1% 2W 2512	1 mOhms ±1% 2W Chip Resistor 2512 (6332 Metric) Automotive	Match	Partial Match	Match	4s 730ms
PA2512FKF070R001E	1 mOhms ±1% 2W Chip Resistor 2512 (6332 Metric)	1 mOhms ±1% 2W Chip Resistor 2512 (6332 Metric) Automotive	Match	Partial Match		
PA2512FKF070R001E	1 kOhms ±1% 2W Chip Resistor 2512 (6332 Metric)	1 mOhms ±1% 2W Chip Resistor 2512 (6332 Metric) Automotive	No Match	-		
PA2512FKF070R001E	1	1 mOhms ±1% 2W Chip Resistor 2512 (6332 Metric) Automotive	Partial Match	No Match		

Below, on Table 1, is shown the breakdown per column, from Figure 2:

Table 1 - Google Sheets Dataset Column Breakdown

Column	Description
MPN	Manufacturer Part Number
Description	Input description to be validated
Optimal Description	Reference engineering description
Desired Output	Expected classification result (may include multiple valid labels)
Actual Output	AI-generated classification
Execution Time	Processing time per case

The dataset (Figure 2) was designed to include a wide range of cases, including:

- Fully correct descriptions
- Partially complete descriptions
- Incorrect or conflicting descriptions

This diversity ensures that the system is evaluated across all possible classification outcomes and reflects realistic engineering conditions encountered in BOM validation processes.

In addition, some dataset entries include two acceptable "Desired Output" values. This reflects the fact that certain component descriptions can be interpreted in more than one valid way from an engineering perspective. In such cases, both classifications (e.g., *Match* and *Partial Match*) are considered acceptable depending on interpretation granularity and documentation standards. This approach was introduced to ensure fair evaluation of the AI system in borderline or ambiguous cases.

User Interface

To provide a practical interface for interacting with the developed AI verification agent, a dedicated Microsoft Excel application was developed. Rather than serving solely as a spreadsheet, the workbook contains a complete program implemented using Excel macros, enabling engineers to configure the AI agent, execute the verification process, and review the generated classifications without interacting directly with the underlying implementation workflow.

The application consists of two primary worksheets: Settings and BOM, each performing a specific role within the verification process.

The Settings worksheet (see Figure 3) contains the configurable parameters required for AI execution. These parameters determine how the AI agent performs the verification and include:

- OpenAI API key;
- GPT model selection;
- Maximum token limit;
- Request timeout;
- Verification prompt.

All verification requests executed through the application use the configuration defined in this worksheet. This design enables engineers to modify AI models, adjust execution parameters, or evaluate different prompt versions without changing the underlying program logic, making the application suitable for both experimental evaluation and future system configuration.

Figure 3 - Settings worksheet of the Excel-based AI verification interface.

Setup	
Field	Value
API Key	
Temperature	0.0
Max Tocken	2,000
Time Out	1000
AI Model	gpt-5

System Query (System Prompt)	
Line 1	You are an expert electronic component verification and validation system for analyzing BOMs.
Line 2	
Line 3	Input Data:
Line 4	The input contains two fields which come from BOM
Line 5	Manufacturer Part Number (MPN), Description
Line 6	
Line 7	Main Task:
Line 8	1- Search reliable internet sources for the MPN and collect its official technical features.
Line 9	> Official manufacturer datasheet
Line 10	> Official manufacturer website

The BOM worksheet (see Figure 4) serves as the primary operational interface. It allows the user to enter a Manufacturer Part Number and its corresponding component description that will be validated by the AI agent. After providing the required input, the engineer initiates the verification process by pressing the **RUN** button. The application then executes the AI verification workflow using the settings specified in the Settings worksheet and returns the final engineering classification (Match, Partial Match, or No Match) directly within the workbook.

Figure 4 - BOM worksheet of the Excel-based AI verification interface.

Run

Item	Manufacturer	Manufacturer Part Number	Description	Result
1	ND	BZX84C3V3-TP	3.3 V 350 mW 5 Surface Mount SOT-23	Match
2	Murata Electronics	GRM188R61E225MA12D	2.2 uF ±20% 25V Ceramic Capacitor X5R 0603 (1608 Metric)	
3	Murata Electronics	GRM188D71A106MA73D	10 uF ±20% 10V Ceramic Capacitor X7T 0603 (1608 Metric)	
4	TDK Corporation	C3216X5R1V226M160AC	22 uF ±20% 35V Ceramic Capacitor X5R 1206 (3216 Metric)	
5	TDK Corporation	C2012X5R1A226K125AB	22 uF ±10% 10V Ceramic Capacitor X5R 0805 (2012 Metric)	
	TDK Corporation	C2012X5R1V106K085AC	10 uF 10%	

At the current stage of development, the application supports verification of a single MPN-description pair during each execution. Consequently, only one row within the BOM worksheet is processed when the **RUN** button is pressed. This limitation was considered acceptable for the proof-of-concept implementation, as the primary objective of the project was to validate the AI verification methodology rather than develop a production-ready engineering application. Future development will extend the application to support batch processing of multiple BOM records, allowing complete Production Master BOMs to be validated automatically in a single execution.

Design Characteristics

The proposed AI verification system was designed to be modular, configurable, and scalable. By separating specification extraction and description validation into two independent AI agents, the reasoning process becomes more structured while allowing each agent to be optimized independently.

The system is also configurable through the Excel interface, where parameters such as the GPT model, prompt, maximum token limit, and request timeout can be modified without changing the underlying application. This supports future experimentation and adaptation to new AI models.

Finally, although the current proof-of-concept validates one MPN-description pair at a time, the overall design provides a foundation for future enhancements, including batch validation of complete Production Master BOMs and integration with industrial engineering systems.

Implementation

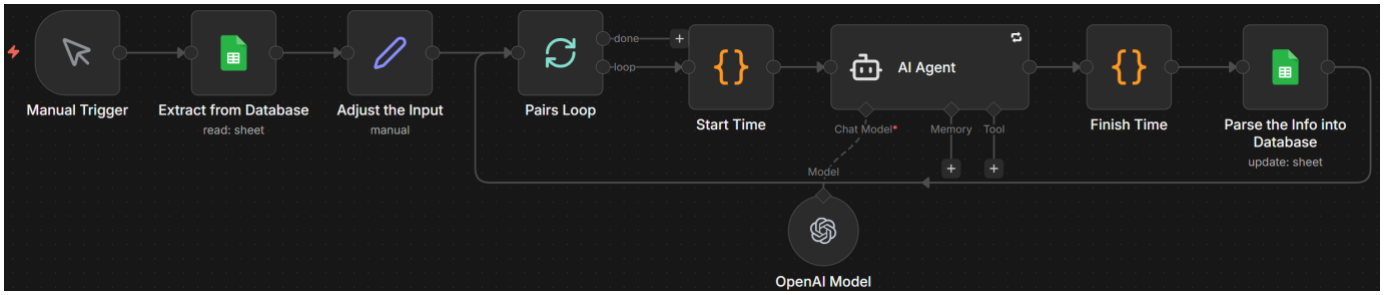
The implementation of the proposed AI verification system followed an extracted development methodology based on continuous experimentation and prompt engineering. Rather than designing a single AI agent and evaluating its performance, the system was progressively improved through multiple implementation stages. Each stage introduced modifications to the workflow, prompt structure, reasoning strategy, or AI model, after which the updated implementation was evaluated using the same validation dataset.

The prototype was developed using the n8n workflow automation platform, which served as the orchestration environment and the execution engine for the AI verification process. The workflow managed data retrieval, AI execution, response processing, execution time measurement, and result storage, allowing every implementation change to be evaluated under identical conditions. Throughout the project, Google Sheets was used to store the validation dataset, while the developed Microsoft Excel application provided the user interface for configuring and executing the AI agent.

Prototype Implementation

The initial implementation consisted of a single AI agent responsible for performing the complete verification process (see Figure 5). During execution, the workflow retrieved an MPN-description pair from the validation dataset, passed it to the AI agent together with the verification prompt, measured the execution time, and stored the returned classification for further analysis.

Figure 5 – Initial 1-Agent n8n Workflow



The n8n workflow automated the complete experimental process, ensuring that every prompt version and AI model was evaluated using identical input data and execution conditions. This eliminated manual intervention during testing and provided a consistent environment for comparing implementation results.

Prompt Development

During the implementation process, prompt design and model behavior tuning were key focus areas. Iterative improvements focused on refining the instructions provided by the language model and controlling its response stability.

The first prompt (see Figure 6) established the basic verification methodology by instructing the AI to extract component specifications from the Manufacturer Part Number, compare them with the supplied description, and return one of the three predefined classifications. Experimental evaluation showed that this implementation suffered from three principal limitations:

- incorrect output formatting;
- poor reasoning quality;
- variable responses between repeated executions.

The first prompt implementation achieved a success rate of only 18.75% using GPT-4o-mini and 34.375% using GPT-4o, demonstrating that substantial prompt refinement was required.

Figure 6 - First Prompt

You are a Manufacturer part number analyzer and validation expert

All actions before should be done internally

Order of Actions:
1. Extract specs for the part number: {{ $('Loop Over Items').item.json.MPN }}. Do not elaborate

2. Compare each spec of the description with specs extracted from the part number.
Description: {{ $('Loop Over Items').item.json.Description }}

-If the pair of specs is exactly the same - return TRUE
-If any little mismatch in the pair of specs occurs - return FALSE
-If the spec is not in the description - return Unsure

Rules:
-Be strict to numbers, any number difference leads to FALSE
-Pay attention to each spec

Output rules:
-Output No Match ONLY if at least 1 spec is FALSE
-If NO FALSE specs exist:
--if key specs are Unsure - output Partial Match
--if all key specs are TRUE - output Match

Output format:
Output only:
-No Match
-Match
-Partial Match

Figure 7 - Eighth Prompt

You are a Manufacturer Part Number and description comparing machine

Step 1 — Extract specs of the component determined by the part number:
Part Number: {{ $('Loop Over Items').item.json.MPN }}

Step 2 — Compare each spec extracted from part number with spec in description:
Description: {{ $('Loop Over Items').item.json.Description }}

Map each spec with TRUE/FALSE/NOT STATED:
- TRUE: Spec is explicitly mentioned in description AND matches exactly
- FALSE: Spec contradicts the extracted value, the number is written in another form
OR there is dual spec representation (e.g. imperial package size is incorrect, while metric is correct)
- NOT STATED: Spec is NOT mentioned in the description at all OR no measurement unit

Convert measurement units (e.g. 1m = 100cm)
Focus on numerical specs

Step 3 — Choose final label:
- If ANY spec is FALSE → No Match
- If no FALSE specs exist:
- If all key specs are TRUE → Match
- If key specs are NOT STATED → Partial Match (means description needs be improved)
Do not improvise or ask or consider any "if" conditions

Notes:
-Any missed measurement unit = Partial Match

INSTRUCTION FOR OUTPUT FORMAT:
Your response must consist entirely of exactly one of the following literal strings:
Match
No Match
Partial Match

In addition to prompt modifications, the temperature parameter of the GPT-4o model was systematically adjusted during experimentation. Lower temperature values were used to improve determinism and reduce variability in classification outputs, particularly in cases where inconsistent reasoning was observed. This tuning contributed to improved stability of results during later stages of development.

Subsequent prompt versions focused on progressively improving the reasoning process. The implemented changes included:

- replacing the ambiguous UNSURE state with NOT STATED;
- introducing explicit TRUE/FALSE/NOT STATED comparison logic;
- strengthening output format restrictions;
- defining quantitative decision rules for Partial Match;
- introducing additional reasoning rules to reduce ambiguity;
- configuring the sampling temperature to zero to improve response consistency.

Each prompt modification was evaluated using the complete validation dataset before additional changes were introduced. This iterative process enabled objective assessment of the impact of every implementation decision.

In the middle of the project we achieved 96.875% accuracy with the eighth prompt (see Figure 7). However, it worked with this accuracy rate only in collaboration with GPT-5 model, which made the use too costly, and increased execution time significantly. If with the first prompt average time of 1 MPN-description pair execution with GPT-4o model was ~1s 346ms, with eighth prompt with GPT-5 took ~16s 651ms. Therefore, after this prompt improvement was focused on designing a prompt which will work as good as this one, but with the cheaper and less complex model.

At the end we had 2 prompts for 2 AI agents: Data Extraction Agent and Description Validating Agent (see Figures 8.1 and 8.2).

Figure 8.1 – Final Prompt (Data Extraction Agent)

Extract specs of the component determined by the part number:
Part Number: {{ $('Loop Over Items').item.json.MPN }}
Only search:
> Official manufacturer datasheet
> Official manufacturer website
> Authorized distributors (Mouser, DigiKey)
> Trusted component databases

Do not decode the part number

Output form:
""Name"" - ""Value:
""Name"" - ...

-No extra text
- Include only specs useful for validating a short product description.
- Maximum 8 specs

Figure 8.2 – Final Prompt (Description Validating Agent)

You are a strict electronic component description validator.
Your task:
Compare the provided canonical manufacturer specs against the provided component description and return the final validation label.

Canonical Specs:
{{ $json.output }}

Description:
{{ $('Loop Over Items').item.json.Description }}

Return only one of these exact labels:
Match
No Match
Partial Match

Definitions:
No Match:
Return No Match if at least one spec that is stated in the description clearly conflicts with the canonical specs OR all specs are not stated
If zero valid specs are stated in the description, return No Match. A unitless malformed value is not a valid spec.

Partial Match:
Return Partial Match if there is no clear conflict, but the description is incomplete, ambiguous, missing important units, missing component type, or needs improvement.

Match:
Return Match only if the description contains enough correct information to identify the component and no stated spec conflicts with the canonical specs.

If uncertain:
- Do not guess.
- If there is a clear conflict, return No Match.
- If there is no clear conflict but information is insufficient, return Partial Match.
- Return Match only when the Match requirements are clearly satisfied.

Final output:
Return only the label.
No explanation.
No punctuation

Here Data Extraction Agent used GPT-4.1-mini model and Description Validating Agent used GPT-5.4-mini.

Experimental evaluation showed that combining specification extraction and description validation within a single prompt often produced incorrect or incomplete canonical specifications, which subsequently affected the final classification. Separating these responsibilities into two specialized AI agents simplified the reasoning performed by each model and enabled independent optimization of both tasks. This architectural change proved to be one of the most significant improvements introduced during the implementation process. And this idea allowed us to achieve 96.875% accuracy rate, with cheaper and simpler models, reducing both the cost and time of execution, which got to ~8s 528ms.

AI Model Evaluation

The evaluation of different AI models was not performed as a separate stage after prompt development, but rather as an integrated part of the iterative optimization process. Throughout the implementation, both the prompt structure and the selected GPT model were continuously adjusted and evaluated together in order to identify the most effective combination for the verification task.

Multiple OpenAI models were investigated during development, including GPT-4o-mini, GPT-4o, GPT-5, GPT-5.4-mini, and hybrid configurations combining different models for specification extraction and validation. Each configuration was tested using the same validation dataset, allowing direct comparison of classification accuracy, execution time, token consumption, and response consistency.

In practice, prompt modifications were often introduced in response to observed model behavior. For example, weaker models required more strictly constrained prompts to reduce reasoning ambiguity, while stronger models allowed for more flexible instruction structures. Similarly, model changes were sometimes introduced to compensate for limitations identified during prompt testing, such as inconsistent reasoning or insufficient specification extraction. This tight coupling between prompt design and model selection was a key aspect of the development process.

During later experimental stages, web search functionality was also enabled for selected model configurations to evaluate whether external information retrieval could improve specification accuracy. While this occasionally improved

factual completeness, it also introduced variability in response structure and increased execution time, which reduced suitability for a controlled verification workflow.

Overall, the experiments demonstrated that model performance and prompt design were strongly interdependent. While more advanced models generally improved reasoning quality, the final system performance was achieved only through careful alignment of model capabilities with appropriately structured prompts. So, prompt engineering and model selection must be treated as a unified optimization problem rather than independent design steps. Below see Table 2 which shows how AI models were switched along to the prompt improvement, and the results of these changes.

Table 2 – Result Table of Iterative Changes in Prompt and AI Model

Prompt	Model	Success Rate	~Prompt Tokens	Average Time	Notes
First	Gpt-4o-mini	18.75%	~220	~1s 373ms	• Incorrect form of output • Poor reasoning
Second	Gpt-4o-mini	25%	~260	~1s 285ms	• Poor reasoning
First	Gpt-4o	34.375%	~220	~1s 346ms	• Incorrect form of output • Poor reasoning • Variable responses
Second	Gpt-4o	62.5%	~280	~1s 278ms	• Poor reasoning
Third	Gpt-4o	65.625%	~280	~0s 935ms	• Poor reasoning
Fourth	Gpt-4o	71.875%	~260	~1s 231ms	• Better but still relatively poor reasoning • Provides less variable responses
Fifth	Gpt-4o	56.25%	~340	~0s 994ms	• Does not vary the response at all • Poor reasoning
Sixth	Gpt-4o	59.375%	~280	~1s 169ms	• Does not vary the response at all • Poor reasoning
Sixth: With Dual-Desired Output	Gpt-4o	81.25%	~280	~1s 169ms	• Does not vary the response at all • Poor reasoning
Seventh	Gpt-4o	68.75%	~330	~1s 8ms	• Poor reasoning
Eighth	Gpt-5	96.875%	~280	~16s 651ms	• Too costly • Too long
Ninth	Gpt-5.4-mini	81.25%	~380	~1s 72ms	• Works stricter • Better, but still relatively poor decision
Tenth	Gpt-5.4-mini	84.375%	~600	~2s 674ms	• Works stricter • Better, but still relatively poor decision
Eleventh	2 agents, gpt-5.4-mini	81.25%	~1900	~16s 690ms	• Too heavy data extraction • Too long execution
Twelfth	2 agents, gpt-5.4-mini	84.375%	~580	~17s 554ms	• Poor reasoning • Too long execution

Thirteenth	2 agents, gpt-5.4-mini	96,875%	~580	~9s 286ms	• Good reasoning • Costly
Thirteenth	2 agents: gpt-4.1-mini & gpt-5-mini	96,875%	~580	~8s 528ms	• Good reasoning

Implementation Summary

The implementation evolved from a simple single-agent verification workflow into a modular two-agent AI system through systematic prompt engineering and continuous experimental evaluation. Rather than relying on isolated modifications, every implementation change was validated using the same dataset and compared using quantitative performance metrics. This iterative approach enabled gradual improvements in reasoning quality, response consistency, execution time, and computational efficiency, ultimately resulting in a robust proof-of-concept suitable for AI-assisted Production Master BOM verification.

Validation and Testing

The developed AI verification system was validated through a structured experimental methodology designed to evaluate both the correctness of the generated classifications and the operational performance of the implemented solution. Every implementation change, including prompt modifications, AI model selection, and architectural improvements, was assessed using the same validation dataset, ensuring that results from different implementation stages could be compared objectively.

Validation Methodology

The validation process was based on a predefined dataset consisting of representative Manufacturer Part Number (MPN) and component description pairs. Each record included an expected engineering classification established before experimentation, allowing the outputs generated by the AI verification agent to be compared with the desired results.

For several validation cases, two different classifications were accepted as correct. These cases represented descriptions that could reasonably be interpreted as either *Match* or *Partial Match* without violating the intended engineering requirements. Consequently, the evaluation methodology considered both predefined outputs as valid, ensuring that the assessment reflected practical engineering judgement rather than enforcing unnecessarily strict deterministic behavior.

To improve confidence in the obtained results, each implementation was executed twice using the complete validation dataset. Repeated execution enabled not only the measurement of classification accuracy but also the evaluation of response consistency, revealing whether identical inputs produced identical outputs under the same execution conditions.

Validation Metrics

The performance of each implementation was evaluated using four quantitative metrics.

Classification Success Rate measured the percentage of validation records for which the generated output matched one of the predefined acceptable classifications.

Execution Time measured the average time required to validate a single MPN-description pair, providing an indication of the practical usability of the developed system.

Prompt Token Consumption represents the approximate number of prompt tokens processed during each execution. Since communication with Large Language Models contributes directly to operational cost, token usage was monitored throughout the project as an important optimization metric.

Response Consistency evaluated whether repeated executions of the same validation dataset produced identical classifications. This metric was particularly important during the early stages of prompt development, where some AI models exhibited non-deterministic behavior.

Experimental Evaluation

Validation was performed continuously throughout the implementation process. After every significant modification to the prompt, workflow, or AI model, the complete validation dataset was processed, and the obtained metrics were recorded.

The experimental evaluation investigated several aspects of the developed solution, including:

- comparison of different GPT models;
- influence of prompt engineering on reasoning quality;
- effect of sampling temperature on response consistency;
- comparison between the single-agent and two-agent architectures;
- influence of prompt size on execution time and token consumption.

Using the same dataset for every experiment ensured that observed performance differences resulted from implementation changes rather than variations in the test data. The results of iterations with different models and prompts can be seen above on Table 2.

Validation Results

The experimental results demonstrate a clear improvement throughout the implementation process. The initial implementation, based on a single AI agent and the first prompt version, achieved a classification success rate of 18.75% using GPT-4o-mini and 34.375% using GPT-4o, indicating that both prompt design and reasoning methodology required significant refinement.

Systematic prompt engineering, refinement of the reasoning logic, evaluation of multiple GPT models, and the introduction of the two-agent architecture progressively improved the performance of the verification system. The final implementation, employing separate Data Extraction and Description Validation agents together with GPT-4.1-mini and GPT-5.4-mini, achieved a validation accuracy of 96.875%, while reducing prompt size to approximately 580 tokens and requiring an average execution time of approximately 8.5 seconds per validation.

The results also demonstrate that increasing model capability alone did not guarantee improved overall performance. Several intermediate implementations produced higher computational cost or longer execution times without achieving proportional improvements in classification accuracy. The best overall performance was obtained only after combining optimized prompts, an appropriate model selection, and the two-agent verification methodology.

Validation Summary

The validation process confirmed that the developed AI verification methodology can accurately determine the consistency between Manufacturer Part Numbers and component descriptions across a representative set of engineering examples. Furthermore, the experimental results demonstrate that systematic prompt engineering, careful AI model selection, and architectural refinement can significantly improve both classification accuracy and operational efficiency. The achieved validation accuracy of 96.875% indicates that the proposed approach provides a strong foundation for future industrial deployment and further development.

Deployment Considerations

Although the developed AI verification system was implemented as a proof-of-concept, its architecture was designed with future industrial deployment in mind. The proposed solution can be integrated into existing engineering workflows to assist engineers during the preparation and validation of Production Master Bills of Materials (BOMs), reducing the time required for manual verification while improving the consistency of component descriptions.

A practical deployment would require integration with existing engineering software, such as ERP, MRP, or Product Lifecycle Management (PLM) systems. Rather than manually entering Manufacturer Part Numbers and descriptions, the AI agent could receive this information directly from the organization's engineering database and return the generated classification to the corresponding system. Such integration would enable automated verification during BOM creation or revision without significantly changing the current engineering workflow.

Since the developed solution relies on Large Language Models accessed through the OpenAI API, deployment also requires appropriate management of API credentials, network connectivity, and access permissions. API keys should be securely stored and managed, ensuring that only authorized users and applications can access the AI verification service.

The validation results demonstrate that the developed system is capable of verifying a single MPN-description pair in approximately 8.5 seconds while achieving a classification accuracy of 96.875%. Although this performance is suitable for engineering validation tasks, deployment within large manufacturing environments may require batch processing, which is going to be introduced in the future, to efficiently validate complete Production Master BOMs containing hundreds or thousands of components.

The modular architecture adopted during the project also supports future maintenance and system evolution. Changes to AI models, prompt design, or verification criteria can be implemented without fundamentally altering the overall verification methodology. This flexibility allows the system to adapt to future developments in Large Language Models and changing engineering requirements while preserving the established verification workflow.

Overall, the developed proof-of-concept demonstrates that AI-assisted verification of Manufacturer Part Numbers and component descriptions is technically feasible and provides a practical foundation for future industrial implementation. Successful deployment will primarily depend on integration with existing engineering systems, secure management of AI services, and continued optimization of performance for large-scale validation tasks.

Results and Benefits

The project successfully demonstrated that Large Language Models can be applied as intelligent engineering assistants for the verification of Manufacturer Part Number (MPN) and component description consistency within Production Master Bills of Materials (BOMs). Through systematic prompt engineering, AI model evaluation, and architectural refinement, the developed proof-of-concept evolved from an initial implementation with limited performance into a solid verification system capable of achieving high classification accuracy.

Project Results

One of the most significant outcomes of the project was the improvement in classification performance throughout the development process. The initial single-agent implementation achieved success rates of 18.75% using GPT-4o-mini and 34.375% using GPT-4o. Successive prompt refinements, model evaluation, and the introduction of the two-agent architecture progressively increased the system's performance. The final implementation, consisting of a dedicated Data Extraction Agent and Description Validation Agent, achieved a validation accuracy of 96.875% while maintaining an average execution time of approximately 8.5 seconds per verification and reducing prompt size to approximately 580 tokens.

In addition to the quantitative improvements, the project demonstrated the effectiveness of prompt engineering as a systematic optimization process. Rather than relying solely on increasingly capable language models, significant performance gains were achieved through careful refinement of prompt structure, reasoning logic, and workflow

design. The experiments confirmed that prompt design had a substantial influence on both classification accuracy and response consistency.

Another important outcome was the development of a modular two-agent verification methodology. Separating specification extraction from description validation simplified the reasoning performed by each AI agent and enabled both tasks to be independently optimized. This architectural approach proved to be more effective than the original single-agent implementation and provides a flexible foundation for future system enhancements.

Table 3 – Comparison of Initial and Final Results

Performance Metric	Initial Value	Final Value
Architecture	Single Agent	2-Agentic System
AI Models	gpt-4o-mini/gpt-4o	gpt-4.1-mini + gpt-5.4-mini
Classification Accuracy	18.75–34.375%	96.875%

Moreover, the average execution time was reduced from ~16s 651ms to ~8s 528ms, compared with the use of eighth prompt with gpt-5 model, which was giving the same success rate. Although current version requires more prompt tokens than eighth prompt with gpt5, the actual cost is now lower, since the models that are currently used are much cheaper. So, while the project was in process we significantly reduced the execution time and cost, without any loss on accuracy.

Project Benefits

The developed AI verification system offers several practical benefits for engineering organizations responsible for maintaining Production Master BOMs.

First, the system reduces the amount of manual effort required to verify component descriptions. Instead of individually comparing Manufacturer Part Numbers with technical documentation, engineers receive an automated assessment indicating whether the supplied description matches the referenced component or requires further review. This can significantly reduce the time spent on repetitive validation activities.

Second, the proposed methodology promotes greater consistency in engineering documentation. By applying the same verification logic to every MPN-description pair, the AI agent reduces subjective differences that may occur when multiple engineers perform manual verification. This contributes to improved quality and standardization of BOM data.

The project also demonstrates that Large Language Models can be successfully applied to engineering verification tasks that require interpretation of technical specifications rather than simple keyword matching. Unlike conventional rule-based validation systems, the developed solution can reason about a wide variety of component descriptions without requiring manually defined validation rules for each component category.

Finally, the modular architecture developed during the project provides a foundation for future industrial deployment. The AI verification methodology can be integrated with existing engineering software and further extended to support additional validation capabilities, larger datasets, and batch processing of complete Production Master BOMs. Consequently, the project establishes a practical basis for future AI-assisted engineering verification systems within industrial environments.

Challenges and Lessons Learned

The development of the AI verification system involved several technical and methodological challenges that significantly influenced the final solution. Addressing these challenges required continuous experimentation with

prompt engineering, AI model selection, workflow design, and system architecture. The lessons learned throughout the project played a crucial role in achieving the final implementation.

Challenges

One of the primary challenges encountered during the project was achieving reliable and consistent AI reasoning. Although the initial implementation successfully established the verification workflow, the generated classifications were often inconsistent and did not always follow the intended decision logic. Repeated executions occasionally produced different outputs for identical inputs, making the system unsuitable for engineering verification tasks where reproducibility is essential.

Another significant challenge was prompt engineering. Designing prompts that consistently guided the language model to interpret technical specifications correctly proved to be considerably more complex than initially anticipated. Small modifications to the prompt wording frequently produced noticeable changes in the quality of the generated classifications. Consequently, prompt development became an iterative optimization process requiring extensive experimentation and objective evaluation.

A further difficulty was making the prompt sufficiently strict, structured, and generalizable to handle a wide variety of components while still maintaining accuracy. Since the scope of the project was intentionally broad, the system had to be capable of processing very different types of items, ranging from simple mechanical parts to more complex electronic components. This created a tension between flexibility and strictness in the prompt design. For example, early versions of the prompt allowed the model too much interpretative freedom, leading to cases where a resistor was incorrectly validated as a capacitor due to overly generic reasoning. Conversely, when the prompt was made too restrictive by overemphasizing specific keywords or patterns, such as forcing exact matches on terms like “voltage rating” or “material type,” the system became less adaptable and failed to generalize to components that did not explicitly contain those attributes. Achieving a balance required careful identification of key specifications that are universally relevant across component types, while avoiding over-constraining the model with domain-specific assumptions.

An additional challenge emerged from the reliability of external data sources used for specification extraction. In several cases, different reputable sources provided conflicting or inconsistent technical values for the same Manufacturer Part Number. This led to situations where the Data Extraction Agent retrieved incorrect or ambiguous specifications, which subsequently affected the final validation outcome. These inconsistencies were not caused by the AI model itself but by variability in the underlying reference data. While this issue was partially mitigated through prompt refinement and cross-checking logic, it remains an inherent limitation of relying on different external data sources. It is expected that this challenge will be further addressed in future stages of the project through improved source prioritization and confidence scoring mechanisms.

Selecting appropriate AI models also presented a challenge. Throughout the project, multiple GPT models were evaluated to determine the most suitable balance between reasoning capability, execution time, and computational cost. The experiments demonstrated that newer or larger language models did not automatically produce better verification results. Instead, overall performance depended on the interaction between the selected model, prompt design, and workflow architecture.

The transition from a single-agent implementation to a two-agent architecture introduced additional implementation complexity. Separating the workflow into independent Data Extraction and Description Validation agents required redesigning prompts, restructuring the workflow, and defining a reliable information output format to be transferred between both reasoning stages. However, this architectural change ultimately became one of the key factors contributing to the significant improvement in verification accuracy.

Lessons Learned

The project demonstrated that prompt engineering is a fundamental aspect of developing AI-assisted engineering applications. The quality of the instructions provided to the language model had a great influence on verification performance and the output turned out to be more responsive to slight changes in the prompt than initially expected, and systematic prompt refinement proved essential for achieving reliable results.

Another important lesson was that effective AI systems should divide complex tasks into smaller, well-defined reasoning steps whenever possible. Separating specification extraction from description validation reduced the

complexity assigned to each AI agent, simplified prompt optimization, and resulted in substantially more accurate and consistent classifications than the original single-agent approach.

The project also highlighted the importance of objective experimental evaluation. By validating every implementation change using the same dataset and predefined success criteria, improvements could be measured quantitatively rather than relying on subjective observations. This iterative methodology enabled informed engineering decisions throughout the development process and ensured that each modification contributed positively to the overall system performance.

Finally, the project demonstrated that developing AI-based engineering tools requires balancing multiple performance factors. While the final implementation introduced higher computational cost and longer execution times than the initial prototype, these trade-offs were justified by the substantial improvement in classification accuracy and verification reliability. This experience emphasized that successful AI system design should consider accuracy, consistency, efficiency, and practical usability as complementary objectives rather than optimizing a single performance metric.

Economic and Industrial Impact

Although the developed AI verification system is currently a proof-of-concept, the project demonstrates considerable potential for application in industrial engineering environments. By automating the verification of Manufacturer Part Number (MPN) and component description consistency, the proposed solution can improve the efficiency and quality of Production Master Bill of Materials (BOM) preparation while reducing the amount of repetitive manual verification performed by engineers.

The developed AI agent is intended to support, rather than replace, engineering decision-making. By automatically identifying descriptions that are inconsistent, incomplete, or potentially incorrect, the system enables engineers to focus their attention on records requiring review instead of manually verifying every MPN-description pair. This approach has the potential to improve engineering productivity, particularly when working with large Production Master BOMs containing hundreds or thousands of components.

The proposed verification methodology also contributes to improved consistency of engineering documentation. Since every MPN-description pair is evaluated using the same AI-driven verification process, subjective differences between individual engineers can be reduced, resulting in more standardized and reliable component descriptions. More consistent BOM data can improve communication across engineering teams and reduce the likelihood of documentation errors propagating to subsequent stages of product development and manufacturing.

From an economic perspective, automating repetitive verification activities has the potential to reduce the engineering effort required for BOM validation and decrease the need for manual rework caused by inconsistent or inaccurate descriptions. Although the scope of this project did not include a quantitative cost-benefit analysis, the achieved verification accuracy of 96.875% demonstrates that the developed methodology is capable of providing reliable engineering support and establishes a strong basis for evaluating its economic impact in future industrial deployments. Reduced engineering effort and saved time allow to reduce the time to market of a new product, therefore contributing to higher competitiveness of an engineering company.

Another important outcome of the project is the demonstration that Large Language Models can be successfully applied to practical engineering verification tasks. Unlike conventional rule-based systems that require manually developed validation rules for different component categories, the developed AI agent applies a generic reasoning methodology that can be adapted to a wide range of components. This flexibility reduces the effort required to extend the system and increases its applicability across different engineering domains.

Finally, the modular design of the developed solution supports future integration with existing engineering software environments, such as ERP, MRP, or Product Lifecycle Management (PLM) systems. Together with the configurable Excel interface and AI-based verification methodology, the project provides a practical foundation for future industrial implementation and demonstrates the potential of AI-assisted verification to improve the efficiency, consistency, and quality of Production Master BOM management.

Innovation

The innovation of this project lies in the application of Large Language Models (LLMs) as intelligent engineering assistants for verifying the consistency between Manufacturer Part Numbers (MPNs) and component descriptions within Production Master Bills of Materials (BOMs). While LLMs are widely used for general conversational tasks, this project demonstrates their application to a specialized engineering verification problem that traditionally relies on manual review and engineering expertise.

A key innovation is the development of a two-agent verification architecture. Instead of assigning all verification tasks to a single AI model, the developed methodology separates the process into two specialized AI agents. The first agent is responsible for extracting the canonical specifications associated with a Manufacturer Part Number, while the second independently validates the supplied component description against those specifications. This modular approach simplifies the reasoning process, improves verification accuracy, and enables each stage to be optimized independently.

Another innovative aspect of the project is the systematic application of prompt engineering as an engineering optimization methodology. Rather than treating prompts as static instructions, the project followed an iterative development process in which prompt modifications were designed, implemented, and objectively evaluated using a predefined validation dataset. This structured approach demonstrated that careful prompt design and workflow architecture can have a greater influence on system performance than changing the underlying AI model alone.

Finally, the project demonstrates a practical methodology for integrating AI into existing engineering workflows. Rather than replacing engineering judgement, the developed system functions as an intelligent decision-support tool that assists engineers by automatically identifying inconsistent or incomplete component descriptions. This human-centered approach illustrates how AI can be incorporated into industrial engineering processes to improve efficiency while preserving engineer oversight and responsibility.

Conclusions

This project investigated the application of Large Language Models (LLMs) for the automated verification of Manufacturer Part Number (MPN) and component description consistency within Production Master Bills of Materials (BOMs). The primary objective was to develop an AI-based verification agent capable of assisting engineers by identifying whether a supplied component description correctly corresponds to the component referenced by its Manufacturer Part Number.

The developed solution evolved through an iterative engineering process involving prompt engineering, evaluation of multiple AI models, workflow refinement, and architectural improvements. The final implementation adopted a two-agent architecture in which one AI agent extracts canonical component specifications, while a second AI agent validates the supplied description against those specifications. This approach significantly improved the reliability and consistency of the verification process compared with the initial single-agent implementation.

Experimental validation demonstrated that the final system achieved a classification accuracy of 96.875% on the developed validation dataset. The project also showed that prompt engineering and workflow architecture had a substantial influence on system performance and that carefully designed AI workflows can achieve significant improvements without relying solely on increasingly capable language models.

Beyond the technical achievements, the project demonstrates the potential of Artificial Intelligence to support engineering activities that traditionally require manual verification and expert knowledge. By automating repetitive validation tasks while maintaining engineers in the decision-making process, the developed methodology provides a practical foundation for future AI-assisted verification of Production Master BOMs.

Overall, the project successfully achieved its objectives and demonstrated that Large Language Models can be effectively applied to engineering verification tasks. The developed proof-of-concept establishes a basis for further research and future industrial implementation, including batch processing of complete BOMs, integration with existing engineering software, and continued optimization of AI models and verification methodologies.

Future Improvements

Although the developed AI verification system successfully demonstrates the feasibility of AI-assisted verification of Manufacturer Part Number (MPN) and component description consistency, several opportunities exist for further development.

One of the primary improvements is the extension of the Microsoft Excel application to support batch processing of multiple MPN-description pairs. The current proof-of-concept validates a single record during each execution, whereas a production-ready implementation should process complete Production Master Bills of Materials (BOMs) automatically, allowing engineers to verify hundreds of components within a single operation.

Another area for future work is the continued optimization of the AI verification workflow. Although the final implementation achieved a classification accuracy of 96.875%, further research could investigate alternative prompt engineering strategies, newer Large Language Models, and additional workflow optimizations to reduce execution time, token consumption, and operational cost while maintaining or improving verification accuracy.

The validation dataset can also be expanded to include a substantially larger number of MPN-description pairs representing a wider variety of component types and description styles. A more comprehensive dataset would enable broader evaluation of the developed methodology and provide greater confidence in its performance across diverse engineering applications.

A further area for future development is improving the reliability of the specification extraction process. In practice, the same Manufacturer Part Number may have different information available across manufacturer datasheets, distributor catalogues, and other trusted technical sources. Future research should focus on developing a methodology that identifies and prioritizes the most reliable information before it is used for validation. This could involve evaluating multiple trusted sources, assigning different levels of confidence to each source, and selecting the most accurate specifications. Such an approach would increase the reliability and strictness of the AI verification process by ensuring that validation is based on the most trustworthy technical information available.

Future development should also focus on integration with industrial engineering systems. Rather than operating as a standalone application, the AI verification methodology could be incorporated into ERP, MRP, or Product Lifecycle Management (PLM) systems, enabling automatic verification during BOM creation, modification, or approval processes.

Finally, an important enhancement would be enabling the system to generate improved component descriptions when a "*Partial Match*" result is detected. In such cases, the AI would not only identify inconsistencies but also rewrite or refine the description to make it fully consistent with the verified specifications. This would transform the system from a purely verification-based tool into a semi-automated documentation assistant, helping engineers correct and standardize BOM entries more efficiently. These enhancements would further extend the system's role as a comprehensive AI-assisted engineering tool for Production Master BOM management.

Appendices

AI agent step-by-step development is in the file below:
https://docs.google.com/document/d/15FG3hX0K5stU9fB41bmJZ9hU0-MyUobprgoXJlrZqjE/edit?usp=sharing

Project Assets

MPN-Description case study is in the document below:
https://docs.google.com/spreadsheets/d/1xoq9UxTsZMycjPUqZesr9N1aT1dujryVbpAI2hhDfpc/edit?usp=sharing

The n8n workflow is in the json file below, copy and paste in your n8n:



Part Number Checking AI Agent __ YEA Engineering.json