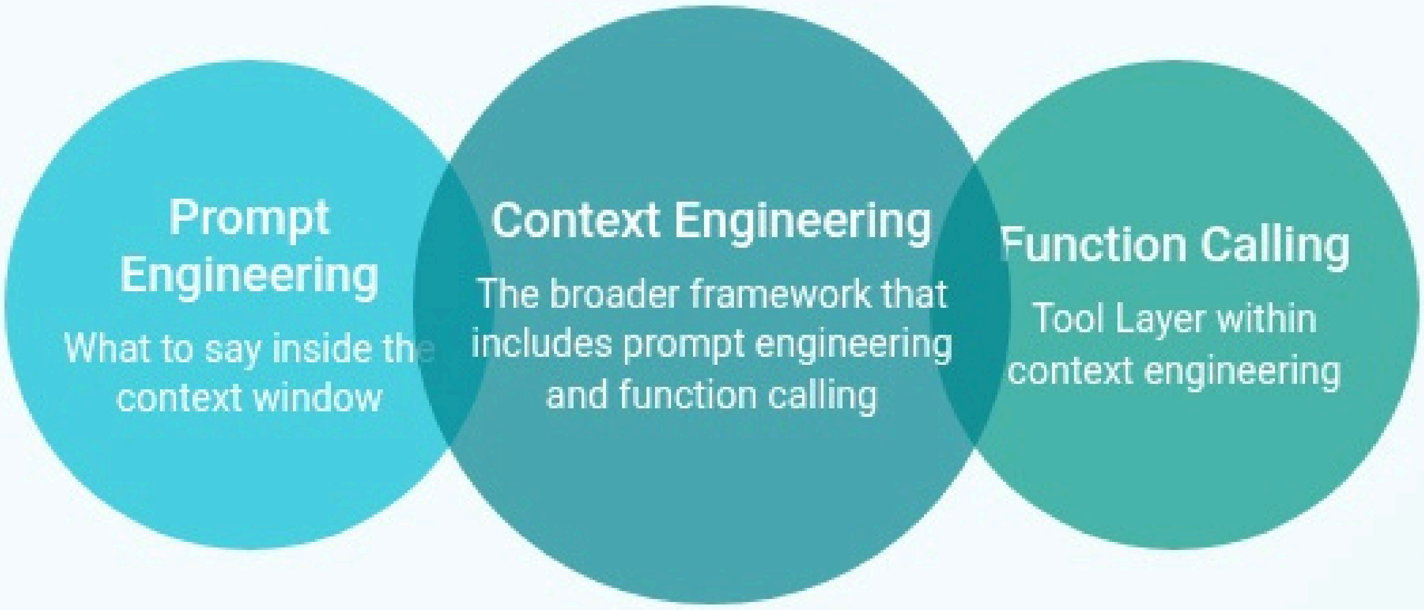
Naresh Edagotti
@Statfusionai

Understanding the Differences: Prompt Engineering, Context Engineering, and Function Calling in Agents

A Comprehensive Guide to AI Agent Development

Relationship Between Concepts



Prompt Engineering
What to say inside the context window

Context Engineering
The broader framework that includes prompt engineering and function calling

Function Calling
Tool Layer within context engineering

Prompt Engineering

Definition

The process of **designing, writing, and optimizing prompts** to guide AI models toward generating desired responses.

Purpose & Focus

Focuses on "**what to say**" to the model at a moment in time. It's about crafting the right instructions within the context window to get a specific response.

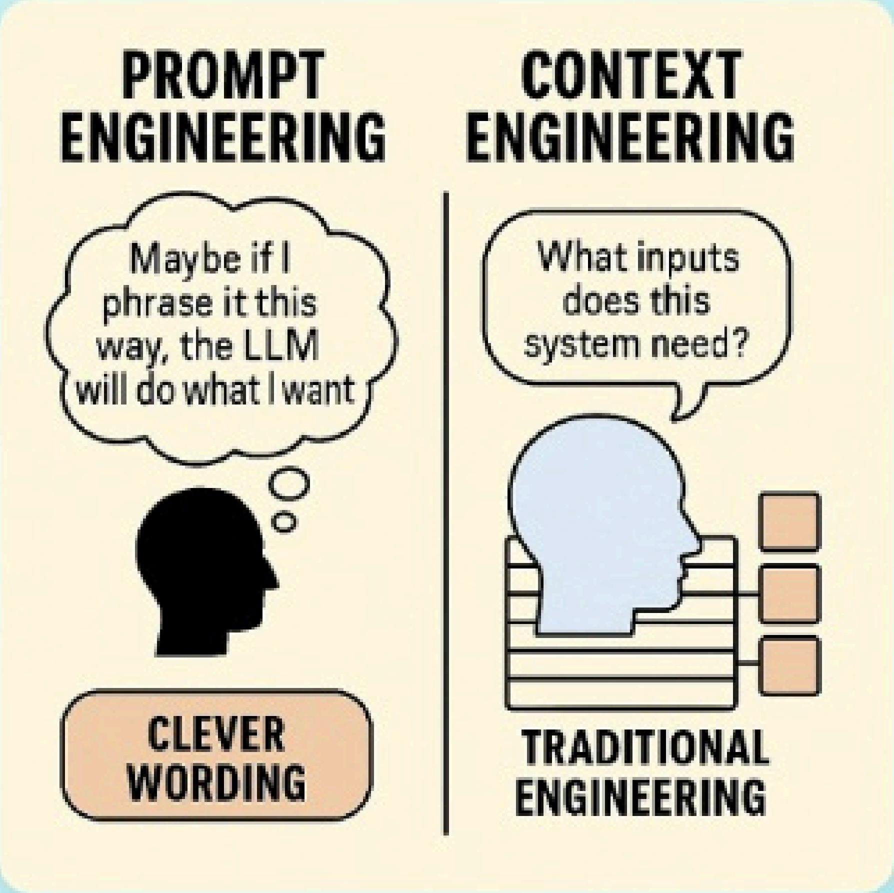
Use Cases

Particularly effective for:

- Copywriting
- One-shot code generation
- Creative writing
- Idea generation
- Quick demos

Key Characteristics

- Like **creative writing** or copy-tweaking
- Relies heavily on **wordsmithing** to get things "just right"
- Operates within a **single input-output pair**
- Can be **hit-or-miss** and often needs manual tweaks
- Gets you the **first good output**
- Great for **short tasks** or bursts of creativity


PROMPT ENGINEERING
Maybe if I phrase it this way, the LLM will do what I want
CLEVER WORDING
CONTEXT ENGINEERING
What inputs does this system need?
TRADITIONAL ENGINEERING

Context Engineering

Definition

The process of **designing, testing, and iterating** on the contextual information provided to AI agents to shape their behavior and improve task performance.

Purpose & Focus

Focuses on **"what the model knows"** when you say something—and why it should care. It's about designing the entire environment in which the LLM operates.

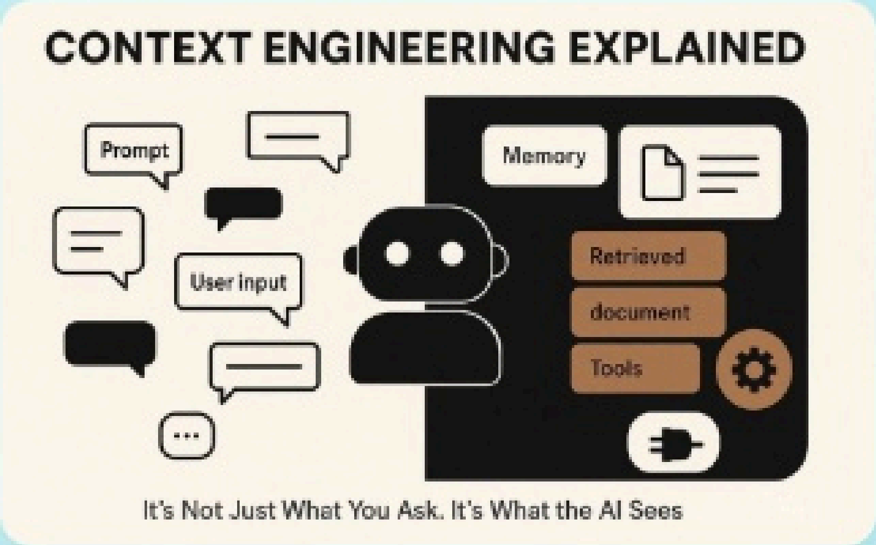
Use Cases

Particularly effective for:

- LLM agents with memory
- Customer support bots
- Multi-turn conversations
- Production systems
- Complex workflows

Key Characteristics

- ✓ Like **systems design** or software architecture
- ✓ Built with **scale in mind** from the beginning
- ✓ Designed for **consistency** and reuse
- ✓ Handles **everything the model sees**—memory, history, tools
- ✓ Makes sure the **1,000th output is still good**
- ✓ Supports **long-running workflows** with complex state



CONTEXT ENGINEERING EXPLAINED

It's Not Just What You Ask. It's What the AI Sees

Function Calling in Agents

Definition

The ability to **reliably connect LLMs to external tools** to enable effective tool usage and interaction with external APIs.

Purpose & Focus

Allows AI agents to **take actions on their environment** by converting natural language into API calls and executing functions with appropriate parameters.

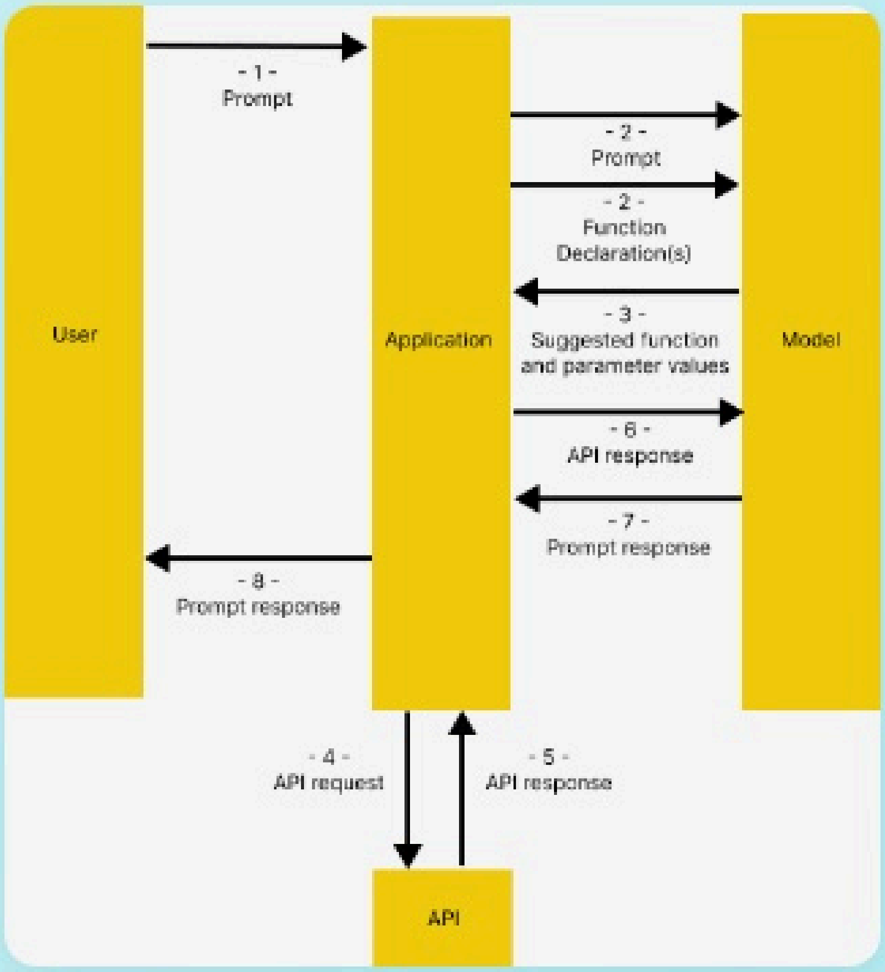
Use Cases

Particularly effective for:

- Conversational agents
- NLP tasks
- API integration
- Math problems
- Data extraction

Key Characteristics

- ✓ LLMs detect when a function needs to be called
- ✓ Outputs **structured JSON** containing arguments
- ✓ Enables **real-time data access** beyond training cutoff
- ✓ Functions act as **tools** in AI applications
- ✓ Supports **multiple functions** in a single request
- ✓ Part of the **Tool Layer** in context engineering



Comparison of Concepts

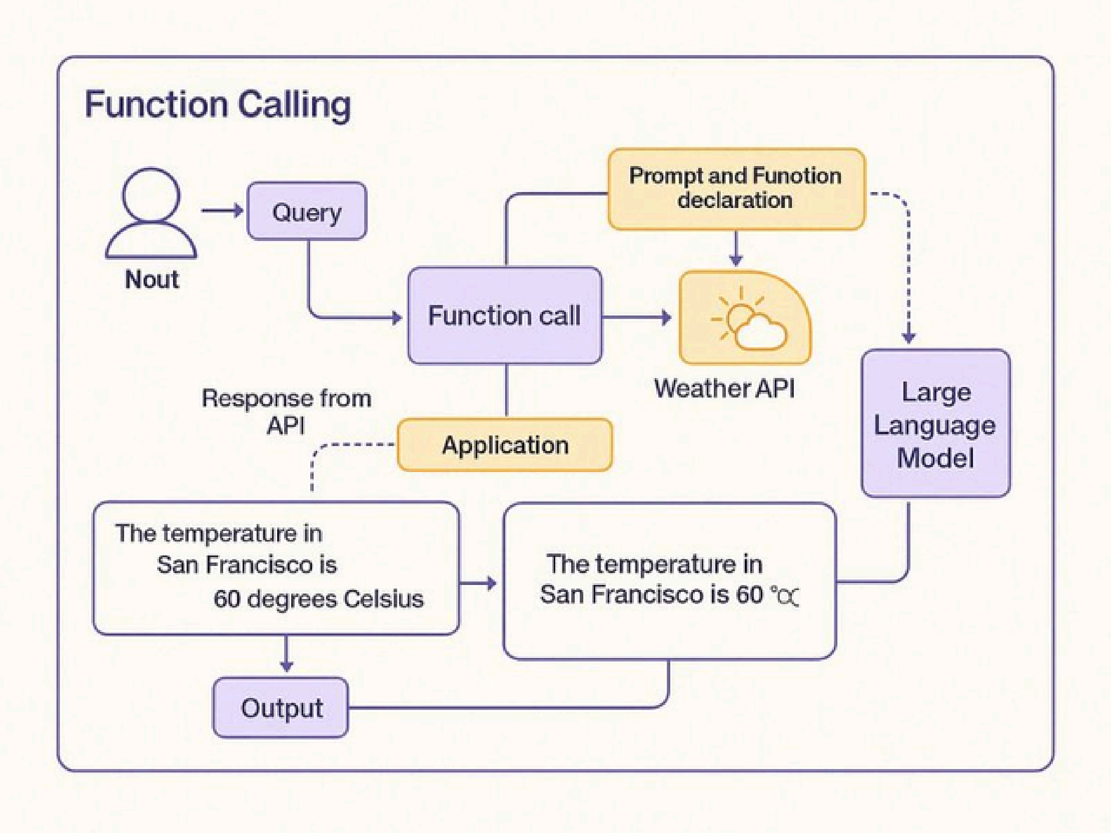
Dimension	Prompt Engineering	Context Engineering	Function Calling
Definition	Designing and optimizing prompts to guide AI models	Designing systems that decide what information an AI model sees	Connecting LLMs to external tools for effective tool usage
Focus	**"What to say"** to the model at a moment in time	**"What the model knows"** and the environment it operates in	**Connecting LLMs** to external tools and APIs
Scope	Single input-output pair	Entire environment including memory, history, tools, system prompts	Tool interaction and API integration
Use Cases	Creative writing, one-shot code generation, copywriting	LLM agents with memory, customer support bots, production systems	Conversational agents, API integration, data extraction
Characteristics	Creative, wordsmithing-focused, good for short tasks	Systems design, scalable, consistent, handles long workflows	Structured output, tool integration, real-time data access

How They Work Together

Relationship Overview

These three concepts form a **hierarchical framework** for building effective AI agents:

- **Context Engineering** is the broad framework that encompasses the entire environment in which an AI agent operates
- **Prompt Engineering** is a subset of Context Engineering, focusing on what to say inside the context window
- **Function Calling** fits into the "Tool Layer" of Context Engineering, enabling agents to interact with external systems


Function Calling
Nout
Query
Prompt and Funotion declaration
Function call
Weather API
Large Language Model
Response from API
Application
The temperature in San Francisco is 60 degrees Celsius
The temperature in San Francisco is 60 °cc
Output

Layered Architecture

Context Engineering organizes AI agent components into a hierarchical structure:

- **System Layer:** Core agent identity and capabilities
- **Task Layer:** Specific instructions for the current task
- **Tool Layer:** Descriptions and usage guidelines for functions
- **Memory Layer:** Relevant historical context and learnings

Practical Example

Consider a customer support AI agent that needs to check order status:

1. **Context Engineering** provides the framework: system prompt, memory of previous interactions, and available tools
2. **Prompt Engineering** crafts the instruction: "You are a helpful customer service agent. Check the order status for the customer."
3. **Function Calling** enables the agent to call the check_order_status function with appropriate parameters