Agentic AI Systems: Design, Implementation, and Real-World Applications

Invited Lecture, Central European University — 4 March 2026

Reda El Makroum

Energy Economics Group (EEG), Technische Universität Wien

elmakroum@eeg.tuwien.ac.at

Website: redaelmakroum.dev

TU WIEN TECHNISCHE UNIVERSITÄT WIEN

EEG Energy Economics Group

About Me

Reda El Makroum

PhD Researcher, Energy Economics Group (EEG), Technische Universität Wien, Austria

Background

- MSc in Sustainable Energy Management, Al Akhawayn University, Morocco.
- Started PhD in 2024 at TU Wien, Austria.

Research Focus

- Agentic AI and LLMs for autonomous decision-making.
- Demand-side flexibility and load scheduling in energy systems.
- Applying AI agent architectures to real-world operational problems.
- Recently published Agentic AI for Home Energy Management

Agenda

1. **Foundations**: What AI means here, what makes it agentic, and how it changed the way we build software.
2. **Why now**: The cost, scale, and infrastructure shifts making agentic systems viable.
3. **Use cases**: Where agentic AI is deployed today, from software engineering to manufacturing to energy.
4. **Implications**: Engineering challenges, the shifting skillset, and what it takes to build reliable systems.
5. **Live demo**: An agentic data analysis tool in action.

What Do We Mean by AI?

AI has developed in three distinct waves, each expanding what software can do autonomously.

Wave 1
Rule-based systems

Humans write explicit logic: if this condition, then that action. Expert systems, decision trees, business rules engines.

Wave 2
Machine learning

Instead of writing rules, the model learns patterns from data. Classification, regression, clustering.

Wave 3
LLMs

A single model trained on broad data that can reason across tasks it was never explicitly designed for.

Sheth & Thirunarayan, "The Duality of Data and Knowledge Across the Three Waves of AI," *IT Professional*, 23(3), 2021

What Makes an AI System “Agentic”?

An LLM on its own is a **prompt-in, response-out** system. It generates outputs, but it does not act. An **agentic** system adds three capabilities:

- **Tool use**: the model can call APIs, execute code, and interact with external systems.
- **Reasoning**: it decides what to do next based on what it has observed so far, not just what it was asked.
- **Iterative execution**: it operates in a loop: act, observe the result, adjust, repeat, until the task is complete.

The difference between a chatbot and an agent is the difference between **answering a question** and **completing a task**.

Components of an AI Agent

Reasoning
Reassessment of action plans with self-correction

Planning
Task decomposition into subtasks for objectives

Memory
Past interactions enabling adaptive responses

Perception
Information gathering from the environment

Communication
Exchange between tools, APIs, and other agents

Tool Calling
Backend integration with databases and code execution

Learning
Autonomous knowledge expansion through experience and feedback, enabling self-improvement across all components.

IBM, “What Are AI Agents?,” IBM Think, 2025

How Software Engineering Changed

A **paradigm shift** in how we build software: from writing every rule by hand to describing what we want and letting the system figure out how.

Before

Software required explicit rules for every case, assembled into deterministic pipelines. Humans sat in every decision loop, and adding a new capability took months of development.

Now

You describe the goal and the system figures out how to get there. It reasons, calls tools, and iterates autonomously across multiple steps. New capabilities are added through prompts, not code.

Why Is This Accelerating Now?

1,000×
inference cost drop in 3 years[1]

142×
fewer parameters for same performance[2]

$0.15/M
GPT-4o Mini input tokens today[3]

- GPT-3 equivalent quality: from **$60/M tokens** (2021) to **$0.06/M tokens** (2024).[1]
- Gartner: <5% of enterprise apps use task-specific AI agents today; projected **40% by end of 2026**.[4]

[1] Appenzeller, "LLMflation," Andreessen Horowitz, Nov. 2024
[2] Stanford HAI, AI Index Report, 2025
[3] OpenAI GPT-4o Mini pricing, 2025
[4] Gartner Press Release, "40% of Enterprise Apps Will Feature Task-Specific AI Agents by 2026," Aug. 2025

How Agents Are Orchestrated

Single agents handle isolated tasks. Real-world complexity requires **multi-agent orchestration**: a coordinator that activates the right agent at the right time.

Centralized (hierarchical)

One orchestrator delegates tasks to specialist agents. Predictable, easy to debug. Most agentic systems today use this pattern.

Decentralized (federated)

Agents communicate as peers with no central controller. More resilient and privacy-preserving, but harder to coordinate.

IBM, “AI Agent Orchestration,” IBM Think, 2025

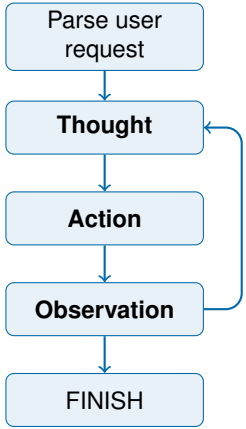
ReAct: Reasoning + Acting

A popular agentic orchestration approach is the **ReAct pattern**[1]: alternating between reasoning about the current state and taking action.

Each iteration produces:

- A **Thought** (what should I do next?)
- An **Action** (call a tool, query an API)
- An **Observation** (what came back?)

The loop repeats until the task is complete or the agent determines it cannot proceed.

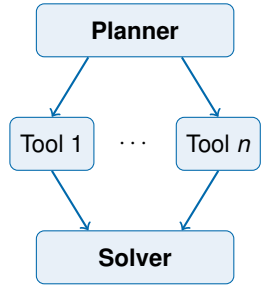


[1] Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," *arXiv:2210.03629*, 2023

ReWOO: Planning Before Acting

ReAct thinks one step at a time, waiting after each action. **ReWOO**[1], which stands for reasoning without observation, plans everything upfront:

1. **Planner**: figures out all the steps and tools needed before starting.
2. **Worker**: runs all tool calls at once (in parallel).
3. **Solver**: combines the results into a final answer.



Faster, cheaper, and easier to scale.

[1] Xu et al., "ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models," *arXiv:2305.18323*, 2023

Where Does Agentic AI Fit Better?

Not every problem needs an agentic system. They excel when:

- **The user is non-technical**: users provide only the core information and describe what needs to happen. The system handles the technical execution.
- **The environment is dynamic**: conditions change faster than a static pipeline can adapt, which applies to most real-world settings. The system needs to observe, reason, and re-plan.
- **The problem is unstructured**: the task involves ambiguous inputs, incomplete information, or decisions that resist clean formalization.

Russell & Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., 2020

Where Is Agentic AI Showing Up Today?

Software Engineering

Agents that read codebases, write code, run tests, and debug autonomously.[1]

Customer Support

Klarna's AI assistant handles two-thirds of customer service conversations.[2]

Data Analysis

Tools like Julius AI enable non-technical users to analyze datasets through conversation.

Manufacturing & Logistics

Multi-agent systems that reformulate scheduling problems on the fly when priorities change.[3]

[1] GitHub, "Copilot Agent Mode," 2025
[2] Klarna, "AI Assistant Handles Two-Thirds of Customer Chats," 2024
[3] Li et al., A4PS, *J. Manuf. Syst.*, 2026

Case Study 1: Warehouse Layout Optimization

An agentic system redesigns a real warehouse layout to reduce forklift congestion.[1]

- A real warehouse with 17 aisles and $\sim$2,400 products. The problem: where to place each product to minimize forklift travel time.
- An LLM agent writes the optimization model, a human checks it, and a simulation tests it. This loop repeats $\sim$10 times until the solution works.
- Result: forklift blocking down **33.7%**, throughput up **7.8%** vs. the baseline.

[1] El Baz et al., "Agentic Optimization Framework," *Comput. Ind. Eng.*, 214, 111884, 2026

Case Study 1: Agentic Optimization Framework

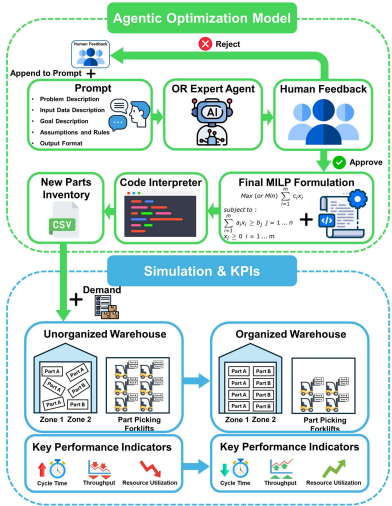


Fig. 1: Agentic optimization framework for warehouse slotting with human-in-the-loop validation.

Case Study 2: Smart Manufacturing Scheduling

An agentic system autonomously rewrites a factory's production schedule when a rush order arrives.[1]

- A VIP rush order arrives at a factory. The production schedule needs to change, which normally takes weeks of back-and-forth between managers, modellers, and developers.
- A4PS replaces this with **8 LLM agents**, each responsible for one step: understand the request, build the model, write the solver code, and debug it.
- Result: modelling success **50%** $\rightarrow$ **75.6%**, code executability **57%** $\rightarrow$ **90%** on complex cases.

[1] Li et al., "A4PS," *J. Manuf. Syst.*, 85, 207–226, 2026

Case Study 2: A4PS Workflow

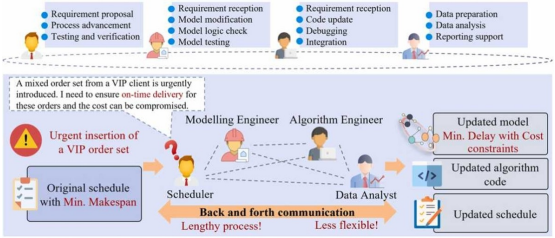


Fig. 2: A4PS multi-agent workflow for autonomous manufacturing scheduling.

Case Study 3: Agentic Home Energy Management

- A household has a washing machine, dishwasher, and EV charger. Electricity prices change every hour. The user says *“charge my EV to 80%, minimize cost”* and the system handles the rest.
- A central orchestrator coordinates three specialist agents, one per appliance, each using the ReAct loop to find the cheapest time slot.
- Tested across 75 trials with real Austrian electricity prices, benchmarked against mathematically optimal schedules.
- Result: Llama-3.3-70B achieves **100% optimality** across all scenarios.

Open source: github.com/RedaElMakroum/agentic-ai-hems Paper: ScienceDirect

El Makroum et al., “Agentic AI HEMS,” *Results in Engineering*, 2026

Case Study 3: System Architecture

A central orchestrator delegates to three specialist agents, each handling a single appliance type.

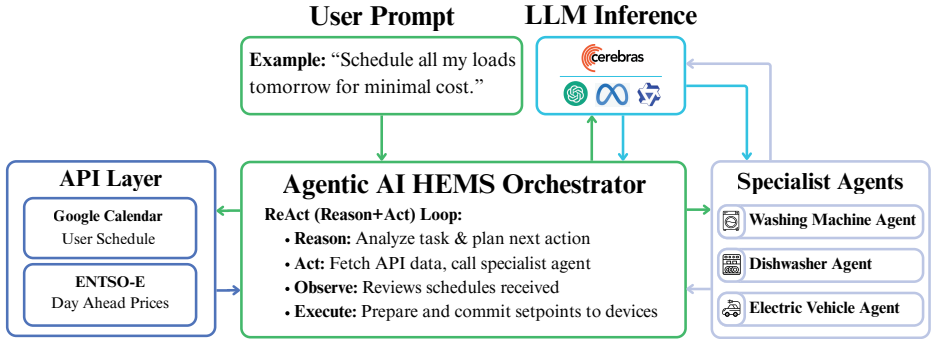
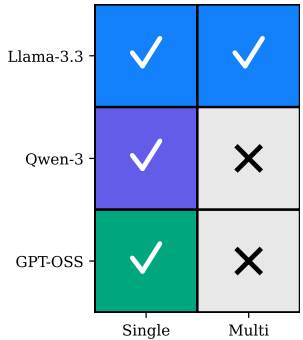


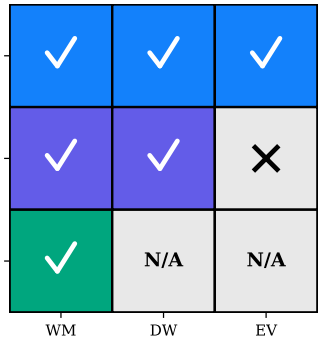
Fig. 3: Agentic AI HEMS architecture with ReAct-based orchestration and specialist agent delegation.

How Does the Agentic System Perform?

Was the orchestrator able to schedule the WM only?



Was the orchestrator successful in scheduling all the loads?



Single-appliance: all models achieve **100% optimality** with similar computational requirements.

Multi-appliance: only Llama-3.3-70B maintains **100% optimality**. Other models fail to coordinate all three appliances.

Engineering Agentic Systems

Building agentic systems that work reliably requires looking beyond the model.

System Prompt Engineering

Without explicit workflow guidance, models fail at multi-step tasks. Our queries showed 0% autonomous success.

Context Engineering

What information to provide, how to structure it, and when to retrieve dynamically vs. pre-load within token budgets.

Token Optimization

Caching, compact prompts, and single-turn specialists reduced token usage by $\sim$**40%**. Direct cost impact at scale.

Security

Prompt injection is a real threat. 3 defense layers: input validation, pattern detection, and privilege separation, all **before** the LLM.

The AI Engineer vs. The ML Engineer

Across all three case studies, the core challenge was the same: not training a model, but designing the system around it.

	ML Engineer	AI Engineer
Core task	Train models	Orchestrate models
Key skill	Feature engineering	Prompt and tool design
Output	A model (.pt, .onnx)	A system (agents, tools, prompts)
Failure mode	Overfitting, data drift	Hallucination, tool misuse

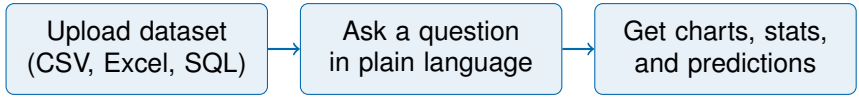
The emerging role of the AI engineer is less about training models and more about **orchestrating them**: designing systems where multiple models, tools, and workflows interact reliably.

Key Takeaways

1. **We moved from programming computers to instructing them.** Agentic AI is not a model improvement. It is a new way of building software, where systems reason, use tools, and adapt autonomously.
2. **Agentic AI is already deployed at scale.** From software engineering to customer support to operations research, the pattern is the same: describe the goal, let the system coordinate execution.
3. **The required skillset is shifting.** Designing agent architectures, evaluation frameworks, and orchestration workflows is becoming as important as training models.

Live Demo: Julius AI

Julius is an AI-powered data analysis platform. Users upload a dataset, ask questions in plain language, and can receive visualizations, statistical tests, and forecasts without writing code.



Feel free to tag along: julius.ai (free tier available)

References

What Do We Mean by AI	Sheth & Thirunarayan, “Three Waves of AI,” *IT Professional*, 23(3), 2021.
Components	IBM, “What Are AI Agents?,” IBM Think, 2025.
Why Now	Appenzeller, “LLMflation,” Andreessen Horowitz, 2024.
	Stanford HAI, AI Index Report, 2025. / OpenAI, GPT-4o Mini Pricing, 2025.
	Gartner, “40% of Enterprise Apps Will Feature AI Agents by 2026,” 2025.
Orchestration	IBM, “AI Agent Orchestration,” IBM Think, 2025.
ReAct	Yao et al., arXiv:2210.03629, 2023.
ReWOO	Xu et al., arXiv:2305.18323, 2023.
Where It Fits	Russell & Norvig, *AI: A Modern Approach*, 4th ed., 2020.
Showing Up Today	GitHub, “Copilot Agent Mode,” 2025. / Klarna, “AI Assistant,” 2024.
Case Study 1	El Baz et al., *Comput. Ind. Eng.*, 214, 111884, 2026.
Case Study 2	Li et al., “A4PS,” *J. Manuf. Syst.*, 85, 207–226, 2026.
Case Study 3	El Makroum et al., *Results in Engineering*, 2026.

Thank You

Reda El Makroum

Energy Economics Group (EEG), Technische Universität Wien

Email: elmakroum@eeg.tuwien.ac.at

redaelmakroum.dev

github.com/RedaElMakroum/agentic-ai-hems

TECHNISCHE UNIVERSITÄT WIEN

Energy Economics Group

The Protocol Layer: How Agents Communicate

As agentic systems multiply, they need standardized ways to discover, authenticate, and exchange information with each other and with external tools.

Protocol	Origin	Purpose
MCP	Anthropic	Standardizes how models obtain context and access tools (JSON-RPC 2.0)
A2A	Google	Agent-to-agent discovery, authentication, and task delegation over HTTPS
ACP	IBM	RESTful API supporting multimodal messages (text, audio, images, video)
ANP	Open	Peer-to-peer with end-to-end encryption and decentralized identity (W3C DID)

Most protocols are still early-stage. MCP and A2A are emerging as the leading standards.

Types of AI Agents

AI agents range from simple reactive systems to fully autonomous learners.

Type	Capability	Example
Simple Reflex	Condition-action rules, no memory	Thermostat
Model-Based Reflex	Internal world model, partial observability	Robot vacuum
Goal-Based	Explicit goals, action sequence planning	Navigation system
Utility-Based	Multi-objective optimization	Route optimizer
Learning	Autonomous improvement through experience	Recommendation engine

Most agentic AI systems today operate at the **learning** level, combining all lower capabilities with continuous self-improvement.