Theoretical Introduction to Agentic AI

A guide to understanding AI Agents, their core concepts, and practical applications in automation.

Agenda: Navigating the World of AI Agents

1 **Introduction & Core Concepts**

Setting the foundation for understanding AI Agents.

2 **Agentic AI vs. LLMs & Workflows**

Distinguishing key AI paradigms.

3 **Workflow Design & Automation Types**

Exploring structured execution patterns.

4 **Technical Foundations & Models**

Understanding the underlying infrastructure and AI models.

5 **Development & Implementation**

Platforms, benefits, and challenges in deploying AI Agents.

What is an AI Agent?

An AI Agent is more than just an AI model; it's a digital worker capable of understanding, deciding, and acting autonomously to achieve specific goals.

Key Capabilities:

- Understands human language and requests.
- Makes autonomous decisions based on environmental feedback.
- Takes actions to achieve goals.
- Interacts with external systems and tools.
- Adapts dynamically to changing conditions.

> **Analogy:** If an LLM is a knowledgeable advisor, an AI Agent is an employee who can advise **and** take action.

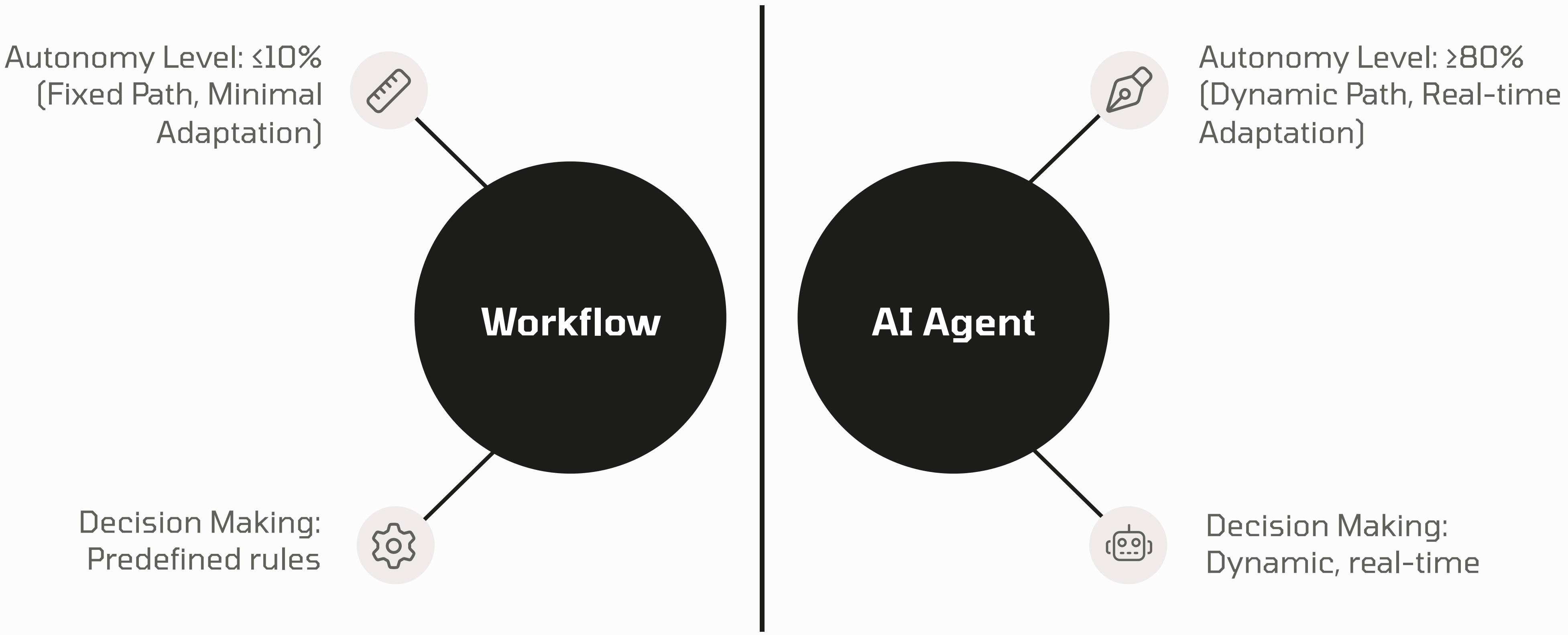
AI Agents vs. Large Language Models (LLMs)

While LLMs are the "brain," AI Agents leverage LLMs with external tools and feedback mechanisms to achieve true autonomy.

Capabilities	Text generation and understanding	Text generation + external actions
Tools	None (standalone)	APIs, databases, external systems
Autonomy	Responds to prompts	Makes independent decisions
Goal Achievement	Provides information/responses	Executes tasks to completion
Environmental Interaction	None	Dynamic interaction with feedback

AI Agents vs. Workflows: The Autonomy Spectrum

The key distinction lies in dynamic decision-making and real-time adaptation.



Autonomy Level: ≤10% (Fixed Path, Minimal Adaptation)

Decision Making: Predefined rules

Workflow

AI Agent

Autonomy Level: ≥80% (Dynamic Path, Real-time Adaptation)

Decision Making: Dynamic, real-time

Workflows

- **Decision Making:** Predefined rules
- **Flexibility:** Low (fixed steps)
- **Example:** A factory production line.

AI Agents

- **Decision Making:** Dynamic reasoning
- **Flexibility:** High (adaptive)
- **Example:** A chef adjusting a recipe.

Workflow Design Patterns: Structuring LLM Interactions

These patterns provide frameworks for breaking down complex tasks and managing how LLMs collaborate.

1

Prompt Chaining

Large tasks broken into smaller, sequential subtasks handled by separate LLMs. Output of one becomes input for the next.

2

Routing (LM Router)

A central LLM directs tasks to specialised LLMs based on task type (e.g., legal questions to a legal LLM).

3

Parallelization

Tasks are divided and processed simultaneously by multiple LLMs, with results then aggregated.

4

Evaluator-Optimizer

A generator LLM creates solutions, which an evaluator LLM then iteratively refines against specific criteria.

Automation Types: From Simple to Agentic

The evolution of automation reflects increasing levels of intelligence and autonomy.

Traditional Automation

Trigger → Action

No intelligence: Rule-based, fixed responses.

Example: Motion sensor turns on a light.

AI Automation

Trigger → LLM → Action

LLM analysis: Limited autonomy (~10%), fixed sequence (workflow).

Example: Customer review → sentiment analysis → store result.

AI Agent Automation

Trigger → LLM with Tools → Action

Dynamic decision-making: High autonomy (>80%), feedback loops.

Example: Telegram message → process request → send email via API.

Technical Infrastructure: The Agent's Anatomy

APIs are the connectors, and the LLM acts as the central processing unit.

APIs: Interacting with the World

Application Programming Interfaces enable AI Agents to communicate with external systems like Gmail, databases, or weather services.

- AI Agent sends requests, servers process and respond.
- Crucial for expanding an agent's capabilities beyond text.
- Cost often credit-based, depending on requests processed.

LLM as CPU: A Core Analogy

- **CPU** = LLM (core reasoning)
- **RAM** = Context Window (active memory)
- **Hard Disk** = File System (long-term data)
- **Programs** = Tools/APIs (external capabilities)

Function Calling: The mechanism allowing LLMs to invoke and incorporate results from these external tools.

AI Models for Agent Development: Choosing the Right "Brain"

Selecting between standard and reasoning models depends on the task's complexity and required accuracy.

Response Time	Fast	Slower ("think time compute")
Computational Cost	Low	Higher
Accuracy	Moderate	High (for complex tasks)
Use Cases	General assistance, creative writing	Mathematical calculations, code debugging, scientific analysis

Benefits & Challenges of AI Agents

Benefits

- **Efficiency:** Automated task execution, reduced wait times.
- **Accuracy:** Minimises human errors, near 1% error rate.
- **Scalability:** Handles increased workload without proportional staff increases.
- **24/7 Availability:** Continuous operation without fatigue.

Challenges

- **Unpredictable Execution:** Dynamic paths make exact steps/outcomes hard to predict.
- **Variable Output Quality:** Results can vary based on reasoning and data.
- **Unpredictable Costs:** Dynamic API usage complicates cost estimation.
- **Safety Concerns:** Risk of unintended actions requires robust safeguards.

Mitigation: Implement monitoring, guardrails, step limits, permission controls, and budget controls.

Standard vs. Reasoning Models: When to Use Which

Understanding the distinctions between model types is crucial for optimising AI Agent performance and cost-effectiveness.

Standard Models: The Quick Helper

- **Fast Response:** Ideal for real-time interactions.
- **Low Cost:** Economical for high-volume, less critical tasks.
- **General Use:** Excellent for open-ended creative tasks or casual conversations.
- **Example:** Generating social media captions or drafting initial email responses.

Reasoning Models: The Thinker

- **Slower ("Think Time"):** Prioritises accuracy over speed for complex problems.
- **Higher Cost:** Justified for tasks requiring precision and deep analysis.
- **Systematic Problem-Solving:** Best for tasks with a definitive correct answer.
- **Example:** Debugging complex code, performing scientific simulations, or financial analysis.

Choosing the right model type aligns the agent's capabilities with the task's demands, balancing speed, accuracy, and resource usage.

Development Platforms

Choosing the right platform is crucial for efficient AI agent development, balancing ease of use with customisation capabilities. Each platform offers unique advantages and considerations for building agentic solutions.

n8n	Open-source, self-hostable, full control	Manual setup/updates	Free (self-hosted)	Custom AI Agents
Make.com	Visual builder, 1500+ integrations	Limited code flexibility	From $9/month	Simple automations
Bubble	No-code app building	Complex logic challenges	From $25/month	Web applications
Voiceflow	Voice/chat specialisation	Limited scope	From $40/month	Conversational AI

n8n Platform Details

n8n offers flexible deployment and robust features for building custom AI agents and workflows.

Deployment: Cloud Hosting

Managed service by n8n:

- Easy OAuth setup ("Sign in with Google")
- Automatic updates
- 14-day free trial (1k executions)
- Pricing from £20/month (2.5k executions)

Deployment: Self-Hosting

For full control and privacy:

- Free, unlimited usage
- Full control over data
- Requires manual updates
- Node.js installation necessary

Install n8n with these commands:

Method	Commands	Description
Temporary (npx)	`npx n8n`	Downloads and runs latest version; resets on shutdown.

Permanent (npm)	`npm install -g n8n`
	`n8n`
	`npm update -g n8n`
		Local storage, persistent sessions; requires manual updates.

Prerequisites	`node --version`	Node.js download required for both methods.

Key Interface Components Of n8n

Workspace

Canvas for building workflows and agents.

Workflows

List of automations and pre-built templates.

Credentials

Securely store API keys and authentication details.

Executions

View runtime logs and historical data.

Templates

Marketplace for ready-to-use workflows.

Building AI Agents in n8n: A Workflow Approach

n8n's visual workflow builder empowers users to design and deploy sophisticated AI agents with ease, connecting language models to real-world actions.

01

Identify the Use Case

Clearly define the problem the AI agent needs to solve, such as automating customer service replies or data extraction from documents.

02

Configure the Trigger

Set up the initial event that starts the agent's workflow (e.g., a new email, a scheduled time, or an incoming webhook request).

03

Integrate LLM & Tool Nodes

Drag and drop LLM nodes (e.g., OpenAI, Anthropic) and connect them to various tool nodes like databases, CRM systems, or messaging platforms.

04

Design Agentic Logic

Utilise n8n's conditional logic and loop capabilities to enable the LLM to make decisions, iterate on tasks, and adapt to different scenarios.

05

Test & Deploy

Thoroughly test the agent's behavior with various inputs and then activate the workflow for continuous operation, monitoring its performance.

Key Takeaways

We've explored the foundational concepts and practical approaches to building AI agents. Here are the key insights to remember:

Agentic AI: Beyond LLMs

AI agents extend large language models with tools, memory, and planning capabilities, enabling autonomous task execution and complex problem-solving.

Strategic Model & Platform Choice

Optimising agent performance and cost involves careful selection between standard and reasoning models, alongside powerful development platforms like n8n.

Visual Workflow Empowerment

Platforms such as n8n streamline agent design, development, and deployment through intuitive visual workflows, connecting LLMs to real-world actions.

Thank You

I hope this presentation on AI Agents and n8n has been useful.