Generative and Agentic AI in Practice

DS 246 (1:2)

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Generative and Agentic AI in Practice: DS 246 (1:2)

Prof. Sashikumaar Ganesan

Lecture Topics

Phase 2: Advanced Topics & Applications

Oct 8: Introduction to Agentic AI

- Foundations of AI Agents — definition, perception–action loop, and core architecture.
- Tool Use & Function Calling — how agents interact with external systems and APIs.
- Planning & Task Decomposition — reasoning strategies, CoT/ToT, and hierarchical planning.
- Reflection & Introspection — self-evaluation, error detection, and continuous improvement.
- Integrated Agent Lifecycle — combining perception, planning, action, and reflection in practice.
- Challenges & Design Trade-offs — managing hallucination, safety, cost, and efficiency.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Learning Objectives

By the end of this lecture, you will be able to:

- Explain the architecture and core components that define an AI agent.
- Differentiate between chatbots, assistants, and autonomous agents based on behavior and autonomy.
- Describe how agents use tools and function calls to extend reasoning into real-world actions.
- Apply key planning and reasoning frameworks (CoT, ToT, ReAct, Plan-and-Execute) to structured problem solving.
- Evaluate how reflection and introspection improve agent reliability, adaptability, and learning.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

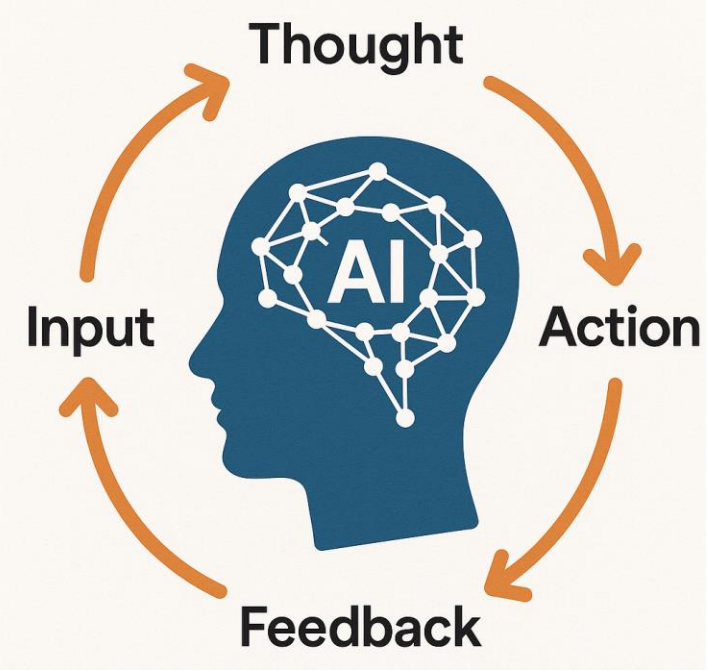
AI Agent

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

What Is an AI Agent? – Definition & Essence

AI Agent

- An autonomous system capable of perceiving, reasoning, acting, and learning from its environment.
- Operates via a closed feedback loop
  - — input → reasoning → action → reflection.
- Designed to achieve goals with minimal human intervention.
- Core idea:
  - LLM + memory + tools + reasoning = intelligent behavior.



© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Agent vs. Chatbot vs. Assistant

Key Insight

- Chatbot → reactive
- Assistant → task executor
- Agent → proactive problem-solver

Type	Characteristics	Example
Chatbot	Scripted dialogue, predefined rules, limited context	FAQ bot
Assistant	Task-oriented, responds to commands	Siri, Alexa
AI Agent	Goal-driven, autonomous planning, tool-using, reflective	AutoGPT, LangGraph Agent

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

The Perception–Action Loop

- The core operating principle of AI agents.
- Steps:
  - Perceive → interpret input/state
  - Plan → decide on next actions
  - Act → use tools or produce outputs
  - Reflect → evaluate results and update memory
- Continuous closed-loop feedback system for improvement.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Core Components of an AI Agent

- LLM Brain – reasoning, language understanding, decision logic.
- Memory – retains context, experiences, and reflections.
- Short-term: current conversation
- Long-term: learned knowledge
- Tools – external actions (APIs, databases, web search).
- Reasoning Engine – decomposes problems, plans, and evaluates outcomes.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Agent Architecture Overview

- A modular pipeline for perception → planning → action → learning.
- Key layers:
  - **Input Processing:** parses and understands user/environmental input.
  - **Planning Layer:** decides what to do and in what order.
  - **Execution Layer:** interacts with tools or APIs to perform tasks.
  - **Memory System:** stores and retrieves context or learned experiences.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Memory Systems in AI Agents

- Short-term memory (Context): – Holds current conversation or working data.
- Long-term memory: – Stores knowledge and past experiences for future use.
- Episodic memory: – Records sequences of events (e.g., previous task steps).
- Enables learning from past actions and contextual continuity.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Summary – Foundations of AI Agents

- AI Agents ≠ chatbots — they reason, plan, act, and reflect.
- Built around the perception–action–reflection loop.
- **Core modules:** LLM brain, memory, tools, reasoning.
- Architecture integrates input processing, planning, execution, and memory.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

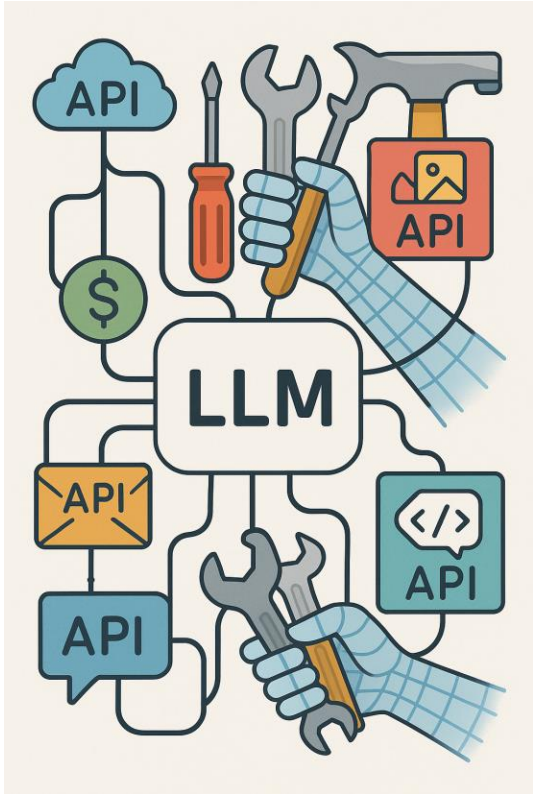
Tool Use & Function Calling

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Introduction to Tool Use

Why Tools Matter in AI Agents

- Tools extend an LLM's reasoning into action — enabling interaction with external systems.
- Transform the model from a language generator → to a problem-solving agent.
- Bridge between reasoning (thinking) and acting (doing).



© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

What Are Tools in AI Agents?

Defining Tools and Their Role

- Tools = external functions or APIs that agents call to perform tasks.
- Examples: search APIs, code runners, data retrieval, file systems.
- Enable real-world interaction, e.g., querying data, executing code, retrieving knowledge.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Function Calling Mechanisms

How Agents Invoke Tools

- Agents use structured JSON calls (function calling / API calling).
- LLM outputs → parsed into function name + parameters.
- Results are re-integrated into reasoning context.
- Common protocols: OpenAI Function Calling, LangChain Tools, MCP, Agent2Agent (A2A).

https://youtu.be/Fbr_Solax1w?si=pKryEmYI6MNtmk2y

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Tool Selection and Orchestration

Choosing and Combining the Right Tools

- Tool selection based on intent classification and context understanding.
- Orchestration involves deciding order and dependencies among multiple tools.
- Can be sequential, parallel, or conditional.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Designing Effective Tool Interfaces

Principles of Tool Design

- Clear input/output definitions — minimize ambiguity.
- Structured function schemas (name, description, args).
- Ensure alignment between model expectations and tool capabilities.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Tool Descriptions and Schemas

How Agents Understand Tools

- Descriptions help LLMs decide when and how to use tools.
- Schema guides argument construction.
  - Example: { "name": "search_web", "description": "Retrieve web results", "parameters": {...} }.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Error Handling & Fallback Strategies

Making Tools Reliable

- Anticipate tool errors, API failures, or invalid responses.
- Implement fallback mechanisms – retry, alternative tools, or manual reasoning.
- Reflection and validation loops enhance reliability.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Multi-Step Tool Chains

Orchestrating Multi-Tool Workflows

- Agents often perform multi-step reasoning involving several tools.
- Each tool’s output can become the next tool’s input.
- Enables complex pipelines (e.g., plan → retrieve → compute → summarize).

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Practical Tool Examples

Common Tool Categories in Agents

- Web Search: dynamic knowledge retrieval.
- Calculators: numerical reasoning.
- APIs: communication and data exchange.
- Code Execution: logic verification.
- Databases/File Operations: structured data access.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Summary – Tool Use & Function Calling

Connecting Reasoning with Action

- Tools empower LLMs to act beyond text generation.
- Well-designed tools enable robust planning, execution, and adaptation.
- Tool orchestration = key to agent autonomy.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

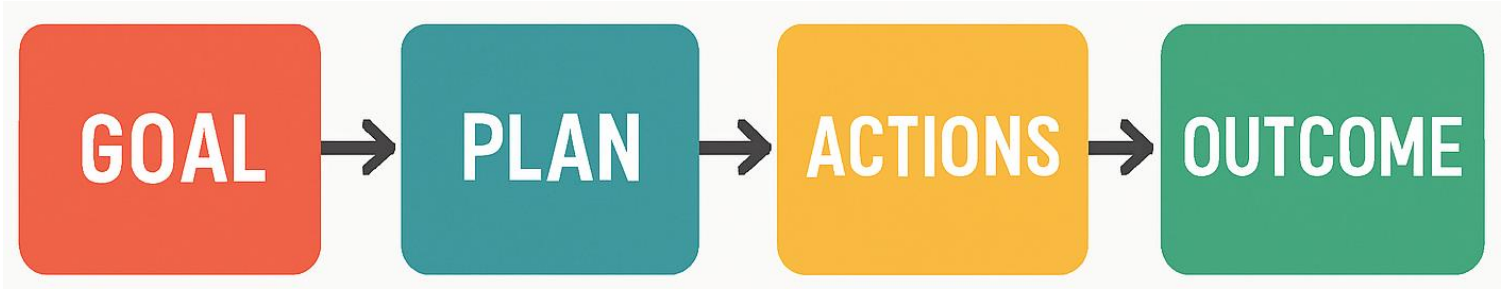
Planning & Task Decomposition

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Introduction to Planning in AI Agents

From Prompt to Plan: The Need for Structure

- AI agents require planning to handle multi-step, goal-driven tasks.
- Planning converts complex objectives → actionable steps.
- Enables autonomy, consistency, and adaptability in reasoning.



© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Zero-shot vs. Few-shot Planning

Different Approaches to Reasoning Structure”

- Zero-shot: Agent plans entirely from context, no examples.
- Few-shot: Uses example patterns to structure the plan.
- Few-shot improves reliability and task generalization.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Chain-of-Thought (CoT) Reasoning

Thinking Step-by-Step

- LLM reveals intermediate reasoning steps before acting.
- CoT = “Let’s reason through this step by step.”
- Enhances transparency, traceability, and accuracy.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Tree-of-Thoughts (ToT) and Graph Planning

Exploring Multiple Reasoning Paths

- Extends CoT by evaluating multiple reasoning branches.
- ToT enables parallel exploration and pruning of weaker paths.
- Graph-based planning handles dependencies and alternatives.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Hierarchical Task Decomposition

Breaking Big Goals into Manageable Steps

- Tasks are decomposed into sub-tasks and micro-actions.
- Enables modular reasoning and scalable planning.
- Often follows “Plan → Execute → Reflect” hierarchy.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

The ReAct Framework

Reasoning + Acting in a Single Loop

- Alternates between thinking (reason) and doing (act).
- Example: ReAct = Reason → Act → Observe → Repeat.
- Prevents “frozen thinking” and keeps reasoning grounded in outcomes.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Plan-and-Execute Frameworks

Separating Planning from Execution

- First, the agent generates a plan end-to-end.
- Then, a second agent or executor carries it out step-by-step.
- Reduces cognitive load and increases control and auditability.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Dynamic Replanning & Adaptation

Handling Uncertainty and Change

- Real-world tasks require adaptive reasoning.
- Agents must replan when tool calls fail or contexts shift.
- Use feedback loops to update partial plans dynamically.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Sequential vs. Parallel Workflows

Choosing the Right Execution Strategy

- Sequential: Tasks depend on previous outputs.
- Parallel: Independent tasks can run simultaneously.
- Balance speed, dependencies, and accuracy.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Resource Allocation & Prioritization

Optimizing Multi-Step Plans

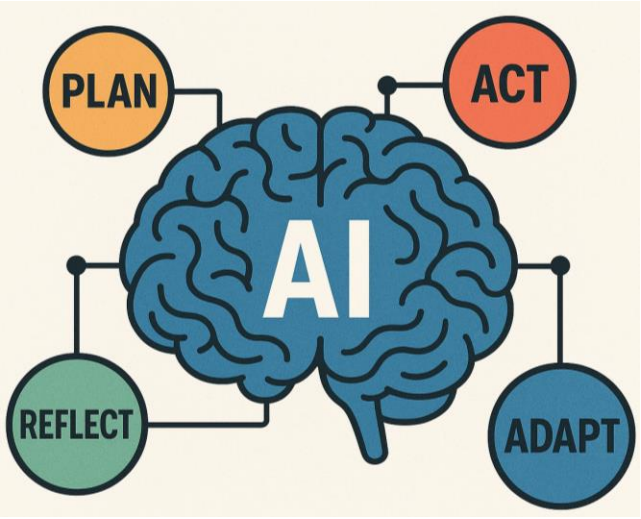
- Agents must allocate compute and tool usage efficiently.
- Prioritize critical tasks and delay low-impact actions.
- Helps minimize cost, latency, and redundancy.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Summary – Planning & Decomposition

Turning Thought into Actionable Strategy

- Planning bridges reasoning → execution.
- CoT, ToT, and hierarchical methods structure complexity.
- Adaptive replanning ensures robustness in real-world tasks.



© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Reflection & Introspection

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Introduction to Reflection in AI Agents

Why Agents Need to Think About Their Thinking

- Reflection = the agent's ability to evaluate and improve its own reasoning.
- Inspired by metacognition in humans ("thinking about thinking").
- Leads to self-correction, learning, and performance improvement.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

The Reflection Loop

Embedding Self-Evaluation into the Reasoning Cycle

- Steps: Generate → Evaluate → Revise → Execute.
- Reflection prompts guide the model to analyze mistakes or biases.
- Enables iterative reasoning refinement after each cycle.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Reflection Prompting Techniques

Designing Prompts for Self-Critique

- Use meta-prompts like “Review your reasoning and identify errors.”
- Self-evaluation improves logical consistency.
- Reflection outputs can update the agent’s confidence score or next step.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Self-Critique & Error Detection

Detecting and Fixing Reasoning Errors

- Agents analyze their own outputs for inconsistencies or missing context.
- Common self-check prompts: “Is my answer complete?”, “What did I miss?”
- Reduces hallucination and logical drift over time.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Confidence Scoring & Uncertainty Estimation

Quantifying How Sure the Agent Is

- Agents estimate confidence in their responses.
- Low confidence triggers reflection or external validation.
- Enhances safety, interpretability, and reliability. Image Suggestion:

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Introspection in AI Agents

Looking Inward: Understanding Internal Reasoning States

- Introspection = analyzing internal reasoning and decision flow.
- Helps explain why certain choices were made.
- Foundation for trustworthy and explainable AI.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Reflexion Framework

Learning from Mistakes Through Experience Replay

- Reflexion = agent self-improvement via reflective feedback.
- Stores previous outcomes → critiques → refines future reasoning.
- Used in autonomous agents like Reflexion (Shinn et al., 2023).

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Self-Ask and Self-Verify Patterns

The Agent Questions and Verifies Its Own Reasoning

- Self-Ask: Agent poses sub-questions to explore reasoning depth.
- Self-Verify: Agent re-checks outputs for factual accuracy.
- Mimics internal Socratic dialogue → improves precision.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Chain-of-Verification

Ensuring Reliability Through Layered Checking

- Multi-stage verification: generate → check → correct → confirm.
- Used in scientific reasoning or multi-step tool use.
- Builds confidence through redundancy and error propagation control.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Memory and Reflection Synergy

How Memory Enables Long-Term Learning

- Episodic Memory: captures reflection events (e.g., success/failure).
- Experience Replay: reuses past learnings to guide new reasoning.
- Over time, reflection builds persistent improvement.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Context & Memory Management

Managing Context Windows in Reflective Agents

- Reflection relies on context continuity.
- Techniques: summarization, retrieval augmentation, long-term recall.
- Balance between relevance and memory efficiency.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Summary – Reflection & Introspection

Building Self-Aware and Reliable Agents

- Reflection improves reasoning quality and reduces hallucination.
- Introspection enhances transparency, self-correction, and learning.
- Together, they form the “self-awareness loop” in AI agents.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

The Complete AI Agent Lifecycle

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

The Complete AI Agent Lifecycle

From Task Reception to Self-Improvement

- The AI agent lifecycle integrates all components:
  - Perception → Planning → Action → Reflection → Learning.
- A continuous loop of reasoning, acting, and adapting.
- Mimics intelligent human behavior through feedback and iteration.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Task Reception & Understanding

Interpreting the User's Goal

- The agent parses user input or system signal.
- Determines intent, constraints, and goal clarity.
- Sets up initial context for reasoning and planning.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Planning & Tool Selection

Designing the Path to the Goal

- Agent creates a multi-step plan using internal reasoning.
- Selects tools based on task requirements (search, code, data, etc.).
- Balances accuracy, efficiency, and reliability.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Execution with Reflection Loops

Acting and Thinking in Tandem

- Executes the plan through tool use and reasoning updates.
- Embeds mini-reflection cycles after each step to verify outcomes.
- Integrates feedback from environment or tool outputs.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Self-Correction & Adaptation

Learning from Mistakes in Real Time

- Agents detect errors or suboptimal outcomes.
- Adjusts future reasoning or tool strategies dynamically.
- Reflection + memory → experience-informed improvement.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Common Challenges in Agent Systems

Pitfalls in the Agent Lifecycle

- **Hallucination:** generating plausible but false reasoning.
- **Tool Misuse:** incorrect tool selection or parameters.
- **Infinite Loops:** reflection cycles without convergence.
- **Latency & Cost:** inefficiencies in planning and execution.
- **Safety:** sandboxing, permissions, and guardrails.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Optimizing Cost and Latency

Balancing Performance and Efficiency

- Use caching, result reuse, and asynchronous workflows.
- Prune unnecessary reasoning or redundant tool calls.
- **Trade-off:** deeper reasoning = better quality but higher cost.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Safety and Sandboxing Mechanisms

Keeping Agents Under Control

- Restrict access through sandboxed tool environments.
- Define execution boundaries and safety checks.
- Essential for real-world deployment and compliance.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Integrated Agent Architecture Overview

Bringing It All Together

- Unified pipeline view:
  - Input Understanding
  - Planning
  - Tool Use
  - Reflection
  - Memory Update
- Each module feeds the next, forming a closed adaptive loop.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Summary – Putting It All Together

The Intelligent Agent Loop

- True AI agents: Perceive → Plan → Act → Reflect → Learn.
- Combine reasoning, tool use, and introspection for autonomy.
- Reflection ensures continuous improvement and goal alignment.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Recap – The Anatomy of an AI Agent

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Recap – The Anatomy of an AI Agent

From Perception to Reflection

- AI Agents combine reasoning, memory, and tools into one coherent system.
- Lifecycle summary:
  - Perception → Planning → Action → Reflection → Learning.
- Each component reinforces adaptability and autonomy.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

The Four Pillars of Agent Intelligence

Core Building Blocks of Intelligent Agents

- **Perception:** Understanding input/context.
- **Planning:** Structuring reasoning and goal pursuit.
- **Action:** Executing through tools and environments.
- **Reflection:** Learning and improving over time.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Agent Design Trade-offs

Balancing Intelligence, Efficiency, and Safety

- **Autonomy vs. Control:** More independence = more risk.
- **Performance vs. Cost:** Deep reasoning vs. efficiency.
- **Complexity vs. Interpretability:** Richer logic vs. harder debugging.
- Intelligent design is about balance, not extremes.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Bridge to Part 2 – Beyond the Basics

Where We Go Next (Tutorial Session)

- Next tutorial session dives into Agent Frameworks and Interoperability.
- Explore how agents communicate via MCP (Model Context Protocol).
- Hands-on with frameworks: LangGraph, CrewAI, AutoGen.
- Building multi-agent ecosystems and collaborative workflows.

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore

Key Takeaways & Closing Thoughts

From Tools to Thinking: The Future of Agents

- AI agents are evolving toward adaptive, self-reflective intelligence.
- The synergy of planning, reasoning, and reflection defines their power.
- “Next-gen agents will not just answer — they will think, act, and improve.”

© 2025: Prof. Sashikumaar Ganesan, IISc Bangalore