




# Mental Models for Google ADK Mastery - Complete Guide 2025

---

**Description:** Master Google Agent Development Kit with comprehensive mental frameworks, 34 tutorials, and production-ready examples. Learn AI agent development from first principles to deployment with Google Gemini.

 **Purpose:** A comprehensive mental framework for understanding Google Agent Development Kit (ADK) and Generative AI concepts from first principles.

 **Source of Truth:** [google/adk-python](https://github.com/google/adk-python) (<https://github.com/google/adk-python>) (ADK 1.15) + Official Google Documentation + 28 Tutorials

 **Status:** Complete mental model synthesis covering all ADK patterns and Generative AI fundamentals

---

## [BRAIN] Core Mental Model: The Agent as a System

---

### | The Agent = Human Worker Analogy

Think of an AI agent like a **human office worker**:

## AI AGENT

## [BRAIN] BRAIN (Model)

- Reasoning
- Decision making
- Language understanding

## [MEM] MEMORY (Context)

- Short-term: Session State
- Long-term: Memory Service
- Working memory: temp: state

## [TOOLS] TOOLS (Capabilities)

- Search web
- Execute code
- Call APIs
- File operations

## [INSTR] INSTRUCTIONS (Behavior)

- Personality
- Rules & constraints
- Task guidance
- Examples

## [FLOW] WORKFLOWS (Process)

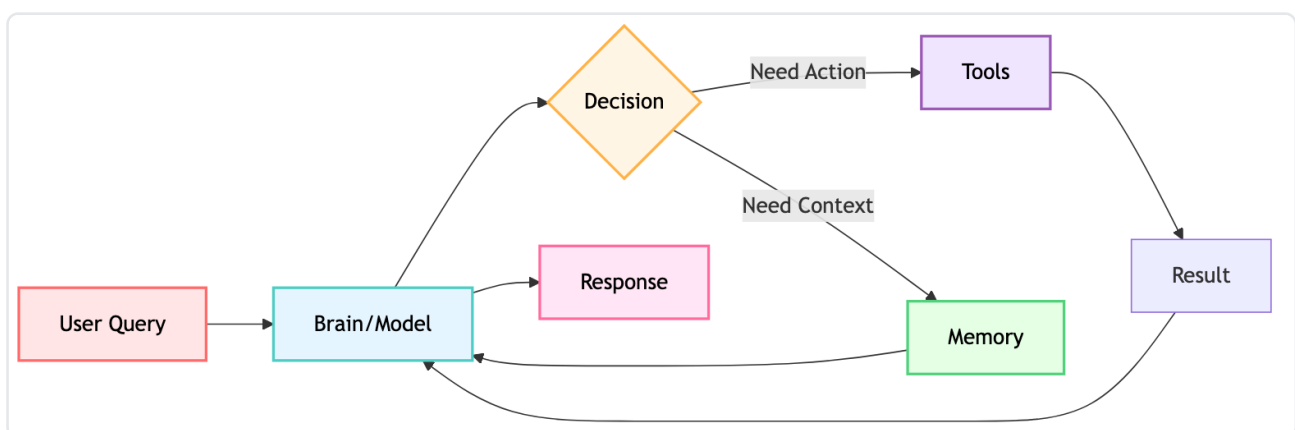
- Sequential steps
- Parallel tasks
- Iterative loops
- Dynamic routing

## [CALLB] CALLBACKS (Supervision)

- Before/after hooks
- Guardrails
- Logging
- Policy enforcement

**Key Insight:** An agent is NOT just an LLM. It's a **complete system** with:

- **Brain** (LLM model) for reasoning
- **Hands** (tools) for taking actions
- **Memory** (state + memory service) for context
- **Instructions** (prompts) for guidance
- **Process** (workflows) for structured execution
- **Supervision** (callbacks) for control



**Source:** [https://github.com/google/adk-python/blob/main/src/google/adk/agents/base\\_agent.py](https://github.com/google/adk-python/blob/main/src/google/adk/agents/base_agent.py)



## Foundational Concepts

### | The Three Types of Agents

**Mental Model:** Agents are like workers with different thinking styles:

AGENT TYPES
<p>[BRAIN] LLM AGENT (Thinker)</p> <p>"I reason and decide dynamically"</p> <ul style="list-style-type: none"><li>- Powered by language model</li><li>- Flexible, creative, adaptive</li><li>- Uses: Conversations, analysis, creative tasks</li></ul> <p>Source: agents/llm_agent.py</p>
<p>[FLOW] WORKFLOW AGENT (Manager)</p> <p>"I follow a strict process"</p> <ul style="list-style-type: none"><li>- Deterministic execution</li><li>- Orchestrates other agents</li><li>- Types: Sequential, Parallel, Loop</li></ul> <p>Uses: Pipelines, coordination, iteration</p> <p>Source: agents/workflow_agents/</p>
<p>[REMOTE] REMOTE AGENT (External Expert)</p> <p>"I'm a specialist from another service"</p> <ul style="list-style-type: none"><li>- HTTP-based agent communication</li><li>- A2A protocol</li><li>- Uses: Microservices, specialized domains</li></ul> <p>Source: agents/remote_a2a_agent.py</p>

**Rule of Thumb:**

- **Use LLM Agent when:** Need reasoning, flexibility, natural language

- **Use Workflow Agent when:** Need predictable, ordered execution
- **Use Remote Agent when:** Need to call external services

---

## Learning Navigation

---

This mental models guide is organized into focused sections for optimal learning:

### **Agent Architecture** → [\(agent-architecture.md\)](#)

- Agent hierarchy and composition patterns
- State vs memory management
- Session and user context handling

### **Tools & Capabilities** → [\(tools-capabilities.md\)](#)

- Tool ecosystem (Function, OpenAPI, MCP, Built-in)
- Tool selection and implementation patterns
- Parallel tool execution

🎓 **Tutorials:** [02](#) ([02\\_function\\_tools.md](#)), [03](#) ([03\\_openapi\\_tools.md](#)), [11](#) ([11\\_built\\_in\\_tools\\_grounding.md](#)), [16](#) ([16\\_mcp\\_integration.md](#))

### **Workflows & Orchestration** → [\(workflows-orchestration.md\)](#)

- Sequential, parallel, and loop workflow patterns
- Complex pipeline construction
- Performance optimization

🎓 **Tutorials:** [04](#) ([04\\_sequential\\_workflows.md](#)), [05](#) ([05\\_parallel\\_processing.md](#)), [06](#) ([06\\_multi\\_agent\\_systems.md](#)), [07](#) ([07\\_loop\\_agents.md](#))

### **LLM Integration** → [\(llm-integration.md\)](#)

- Prompt engineering and instruction patterns
- Grounding and real-world connection

- Thinking and reasoning frameworks

🎓 **Tutorials:** [12](#) ([12\\_planners\\_thinking.md](#)), [22](#) ([22\\_model\\_selection.md](#)), [28](#) ([28\\_using\\_other\\_llms.md](#))

## **Production & Deployment** → ([production-deployment.md](#))

- Deployment environments and strategies
- Observability and monitoring
- Service configuration

🎓 **Tutorials:** [23](#) ([23\\_production\\_deployment.md](#)), [24](#) ([24\\_advanced\\_observability.md](#)), [25](#) ([25\\_best\\_practices.md](#))

## **Advanced Patterns** → ([advanced-patterns.md](#))

- Streaming and real-time interaction
- MCP protocol and standardization
- Agent-to-agent communication

🎓 **Tutorials:** [13](#) ([13\\_code\\_execution.md](#)), [14](#) ([14\\_streaming\\_sse.md](#)), [15](#) ([15\\_live\\_api\\_audio.md](#)), [17](#) ([17\\_agent\\_to\\_agent.md](#))

## **Decision Frameworks** → ([decision-frameworks.md](#))

- When to use each pattern
- Cost optimization strategies
- Pattern selection guides

## **Learning Paths** → ([learning-paths.md](#))

- Structured learning approaches
- Tutorial sequences
- Skill progression

## **Reference Guide** → ([reference-guide.md](#))

- Source code navigation

- Quick reference tables
- API and configuration guides

## **Glossary** → [\(glossary.md\)](#)

- ADK terms and concepts
- Terminology definitions
- Quick reference tables

## **ADK Cheat Sheet** → [\(adk-cheat-sheet.md\)](#)

- Quick reference guide
- Commands, patterns, and best practices
- Troubleshooting and common issues

# **Key Principles & Rules of Thumb**

---

## **The 10 Commandments of ADK Development**

1. **Agent = System, Not Just LLM**
2. Always think: Model + Tools + State + Instructions + Workflows
3. **State for Short-term, Memory for Long-term**
4. Session state = this conversation
5. Memory service = all conversations
6. **Sequential When Order Matters, Parallel When Speed Matters**
7. Dependencies → Sequential
8. Independent → Parallel
9. **Loop for Quality, Not for Logic**
10. Use LoopAgent for refinement
11. Use SequentialAgent for ordered steps
12. **Ground Everything That Needs to Be True**

13. Facts → google\_search
  14. Data → database tools
  15. Locations → google\_maps
  16. **Tools Are Capabilities, Not Afterthoughts**
  17. Design tools with agents in mind
  18. Return structured data (dicts)
  19. Include clear docstrings
  20. **Callbacks for Control, Not Core Logic**
  21. Use for guardrails, logging, monitoring
  22. Don't put business logic in callbacks
  23. **Start Simple, Add Complexity When Needed**
  24. Single agent → Multi-agent
  25. Sequential → Add parallel
  26. No thinking → Add planner
  27. **Evaluate Early, Evaluate Often**
  28. Create test sets from day one
  29. Run evals with every major change
  30. Use Trace view for debugging
  31. **Production ≠ Development**
    1. Local: InMemory services
    2. Production: Persistent services (PostgreSQL, GCS, Vertex)
- 

## Getting Started

---

**Recommended Path:** Start with this overview, then follow the [Foundation Learning Path](#) ([learning-paths#path-1-foundation-start-here](#)) for a structured approach to mastering ADK.

**Quick Start:** If you're new to ADK, begin with [Tutorial 01: Hello World Agent](#) ([01\\_hello\\_world\\_agent.md](#)) after reading this overview.

**Reference:** Check the [Glossary](#) ([glossary.md](#)) for definitions of key ADK terms and concepts.

**Source Code:** All mental models are derived from the official ADK source code in <https://github.com/google/adk-python>. When in doubt, refer to the source code for truth.

---

## Document Metadata

---

**Created:** 2025-01-26

**Version:** 1.0

**Source:** Research from <https://github.com/google/adk-python> + 28 comprehensive tutorials

**Purpose:** Mental models for mastering Google ADK and Generative AI

**Audience:** Developers learning ADK from beginner to advanced

**Maintenance:** Update as ADK evolves (weekly releases)

 **You now have the foundation to build exceptional AI agents with Google ADK!**

---

Generated on 2025-10-19 17:57:43 from overview.md

Source: Google ADK Training Hub