AI AGENT SYSTEMS

FIELD GUIDE V2



A practical premium reference and training product for understanding and implementing AI-agent systems.

Reader: Founders, operators and product teams with basic AI knowledge.

Contents

Chapter 1: Agent Foundations - concepts 01-10

Chapter 2: Prompting and Context - concepts 11-21

Chapter 3: Model Lifecycle - concepts 22-27

Chapter 4: Tool and API Execution - concepts 28-31

Chapter 5: Safety and Evaluation - concepts 32-36

Chapter 6: Operations and Control - concepts 37-46

Chapter 7: Distribution and Deployment - concepts 47-50

Five system diagrams

Three complete examples

Glossary

Bibliography and full-edition information

Introduction

This private guide turns the 50 owner-approved premium AgentiScript icons into a practical control map for real AI-agent work. Each concept page explains what the component is for, how a team might use it, what commonly fails, and the control that should exist before customer-facing release.

Accessibility note: these PDFs are marked ACCESSIBILITY BLOCKED because no dedicated PDF/UA validator is installed in the approved local/free toolchain.

01 / 1. Agent Foundations

Agent Core



PURPOSE
Defines the central runtime that accepts goals, coordinates tools, and returns auditable results.

PRACTICAL EXAMPLE
A team defines which requests the agent may accept, which tools it may call, and which results count as complete.

COMMON FAILURE
The agent becomes a catch-all process with unclear authority and no auditable boundary.

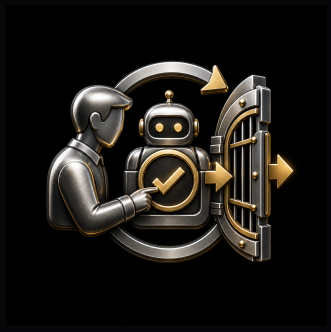
RECOMMENDED CONTROL
Define one clear responsibility and expose every external effect through an auditable boundary.

RELATED CONCEPTS
Agent Planning; Agent Tool Use; Observability

REFERENCES
[NIST-AI-RMF], [MCP-2026]

09 / 1. Agent Foundations

Human in the Loop



PURPOSE
Defines where human judgment must approve, revise, or reject agent output before consequences occur.

PRACTICAL EXAMPLE
A refund assistant drafts an answer, but a human approves any exception or customer-impacting credit.

COMMON FAILURE
Review happens too early or too vaguely to control the actual consequential action.

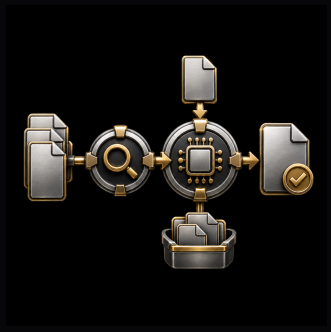
RECOMMENDED CONTROL
Require explicit human approval where judgment, authority, or irreversible effects cannot be delegated safely.

RELATED CONCEPTS
Approval Gate; Access Control; Red Teaming

REFERENCES
[NIST-AI-RMF], [MCP-2026]

19 / 2. Prompting and Context

RAG Pipeline



PURPOSE
Defines the path from source content through retrieval into a generated, citation-backed response.

PRACTICAL EXAMPLE
A field-guide assistant retrieves source sections, drafts the answer, and rejects claims with no citation.

COMMON FAILURE
Generated language sounds sourced, but the cited evidence does not support the claim.

RECOMMENDED CONTROL
Require grounded citations, separate retrieved facts from generated language, and fail closed when evidence is insufficient.

RELATED CONCEPTS
Guardrails; Observability; Checkpoint

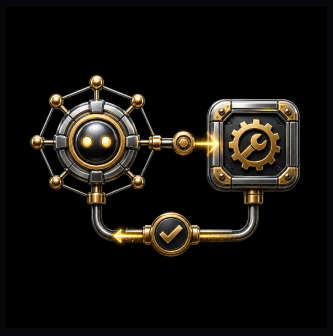
INPUTS
Source records, retrieval query, prompt, citation rules.

OUTPUTS
Grounded response with cited source material.

REFERENCES
[JSON-SCHEMA], [OWASP-GENAI]

28 / 4. Tool and API Execution

Function Calling



PURPOSE
Defines a model-requested function invocation governed by schema validation and runtime authorization.

PRACTICAL EXAMPLE
A model proposes a function call, but the runtime validates arguments before any tool executes.

COMMON FAILURE
The runtime accepts malformed arguments because the model produced plausible JSON.

RECOMMENDED CONTROL
Use strict schemas, allowlisted functions, bounded timeouts, and result validation before accepting tool output.

RELATED CONCEPTS
Guardrails; Observability; Checkpoint

INPUTS
Model request, function schema, tool arguments.

OUTPUTS
Tool call result accepted only after validation.

REFERENCES
[OPENAI-SO], [JSON-SCHEMA], [OAUTH2]

32 / 5. Safety and Evaluation

Guardrails



PURPOSE

Defines blocking and escalation controls that keep unsafe or unauthorized behavior from taking effect.

PRACTICAL EXAMPLE

The system blocks unsafe, unauthorized, or out-of-scope actions before any tool or publication step.

COMMON FAILURE

Controls are advisory only and do not actually block execution.

RECOMMENDED CONTROL

Fail closed at trust boundaries and log why an action was allowed, rejected, or escalated.

RELATED CONCEPTS

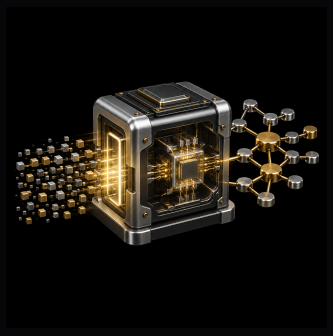
Guardrails; Observability; Checkpoint

REFERENCES

[NIST-AI-RMF], [OWASP-GENAI]

47 / 7. Distribution and Deployment

Machine Readability



PURPOSE
Defines structured product or system data that software can parse and validate reliably.

PRACTICAL EXAMPLE
The package includes a JSON manifest so scripts can verify files, hashes, editions, and status.

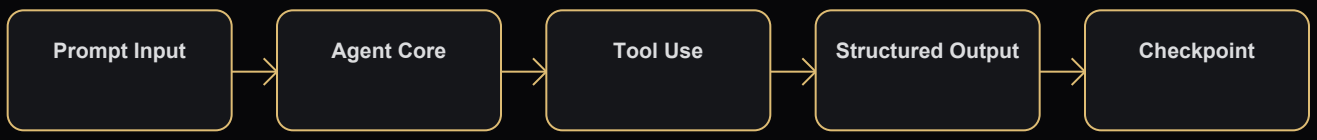
COMMON FAILURE
Humans understand the product, but scripts cannot verify the package.

RECOMMENDED CONTROL
Publish stable schemas, explicit types, and versioned semantics that parsers can validate reliably.

RELATED CONCEPTS
Guardrails; Observability; Checkpoint

REFERENCES
[MCP-2026], [JSON-SCHEMA]

Diagram 1. Controlled Agent Loop



Flow: Prompt Input -> Agent Core -> Tool Use -> Structured Output -> Checkpoint

Explanation: Shows how a request becomes a bounded action only after the agent core, tool boundary, structured response, and saved state all line up.

References: [NIST-AI-RMF], [JSON-SCHEMA]

Example 1. Customer support agent with approval gates

Situation: A SaaS company wants an agent to draft account-specific support replies without changing customer records automatically.

Goal: Reduce response time while keeping customer-impacting actions behind review.

System: Prompt rules, knowledge base retrieval, structured answer schema, tool allowlist, approval gate, execution trace.

Flow: Prompt Input -> Retrieval -> RAG Pipeline -> Structured Output -> Human in the Loop -> Checkpoint.

Controls: Cited sources, no unaudited account mutation, signed tool requests, reviewer decision log.

Failure response: Reject unsupported answers, keep the draft, preserve trace evidence, and route to a human support owner.

Evidence: Source links, retrieved passages, output schema result, approval record, and final customer message hash.

References: [NIST-AI-RMF], [OWASP-GENAI], [JSON-SCHEMA]

Sources + Full-Edition Information

[NIST-AI-RMF] NIST AI Risk Management Framework 1.0: https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10

[NIST-CSF] NIST Cybersecurity Framework 2.0: https://www.nist.gov/publications/nist-cybersecurity-framework-csf-20

[OWASP-GENAI] OWASP GenAI LLM Top 10 2026: https://owasp.org/www-project-top-10-for-large-language-model-applications/

[MCP-2026] Model Context Protocol 2026-07-28: https://blog.modelcontextprotocol.io/posts/2026-07-28/

[JSON-SCHEMA] JSON Schema 2020-12: https://json-schema.org/specification

[OTEL] OpenTelemetry Specification 1.60.0: https://opentelemetry.io/docs/specs/otel/

[OAUTH2] OAuth 2.0 RFC 6749: https://datatracker.ietf.org/doc/rfc6749/

[WCAG22] Web Content Accessibility Guidelines 2.2: https://www.w3.org/TR/WCAG22/

[OPENAI-SO] OpenAI Structured Outputs: https://openai.com/index/introducing-structured-outputs-in-the-api/

[NIST-PRIVACY] NIST Privacy Framework 1.0: https://www.nist.gov/privacy-framework/privacy-framework

[OWASP-API] OWASP API Security Top 10 2023: https://owasp.org/API-Security/editions/2023/en/0x11-t10/

[HTTP-SEMANTICS] HTTP Semantics RFC 9110: https://datatracker.ietf.org/doc/rfc9110/

[WAI-ALT] W3C WAI Images Tutorial: https://www.w3.org/WAI/tutorials/images/

The full private edition includes all 50 concept pages, five diagrams, three examples, glossary, and bibliography. No publication is authorized.