SAGAI

Strategy, Architecture, Governance and AI

Sagai Architecture: build your architecture with AI

A method for designing, building, testing, using and improving your architecture framework. AI agents prepare drafts and proposals. People decide.

7	10	6	7	4	10	120
stages	templates	gates	controls	tests	areas	types

THE LIFECYCLE | seven stages, in one order, with templates for each and a decision gate that closes six of them

1 Assess

Assessment report: drivers, constraints, findings with IDs, overlaps and gaps. Landscape started, as documented and as it runs

Gate AG1

Current Architecture Assessment Report
Landscape and Dependency Register

2 Choose

Adopt, adapt or create for each part, with reasons. A decision record where the threshold is met

Gate AG2

Adopt, Adapt or Create Decision Record
Architecture Decision Record

3 Design

Six dimensions, decision rights, the cost-to-reverse threshold. Each dimension has its legal angle

Gate AG3

Architecture Framework Design Brief

4 Build

Agents draft, under the seven controls. People edit and accept

Gate AG4, at the start

Agent Task Brief and Controls Check

5 Test

Tabletop, walk-through, evidence check, consistency check. Register compared with what runs

Gate AG5

Framework Test Plan and Results

6 Put into use

Rollout through the organisation's own route. Review route and register working from day one

Gate AG6

Rollout Record

7 Improve

Review on schedule. Outside reviewers' answers recorded. Open items return to Assess

No gate of its own

Improvement and Review Log

Boundaries. A method, not answers: you give your AI agents principles, standards, decision rights, target-state views and landscape records. The scope is to build your architecture framework with your AI agents, and not to design any one system. The choice of vendors, products and platforms stays with you. The framework takes you through its gates, helps you agree autonomy levels and records escalations, and enforces none of them. It works by itself, and no other Sagai framework is needed. Nothing here is legal advice or architecture advice.

NINE DECISIONS PEOPLE KEEP

1. **Which areas are in scope**
2. **The drivers and constraints**
3. **Which principles and standards bind**
4. **The target state, and what is kept or retired**
5. **Who decides what, and who reviews**
6. **The cost-to-reverse threshold**
7. **Whether a deviation is granted**
8. **Whether a finding is accepted as a risk**
9. **When it goes into use**

A proposal has no standing until a person accepts or declines it.

WHAT AGENTS PREPARE

Well suited

Lists of what is documented, application and technology lists, dependency maps, the register compared with records of what runs, contradictions between principles, standards and decisions, tabletop scenarios, first drafts for people to edit.

Much less so

Target-state and trade-off work. An agent lays out options with reasons, records what was decided, and stops.

Agents never edit their own controls. The drafter is never the tester.

SEVEN CONTROLS FOR THE AGENTS IN USE

1. A named owner
2. A written task list
3. Read by default
4. Marked as draft
5. Prompts and outputs kept
6. Approved tools only
7. A person can switch it off

The minimum, not the limit. For tighter control, use a framework built for AI agents, such as ADLC.

FOUR TESTS

1. **Tabletop exercise**
   Walk a scenario through with the people involved
2. **Control walk-through**
   Follow one control from the rule to the evidence
3. **Evidence check**
   Does the evidence exist, current and findable?
4. **Consistency check**
   Contradictions, gaps and overlaps. Register against what runs

Before use: a tabletop, a consistency check and the register comparison.

TEN CRITERIA FOR A FINISHED FRAMEWORK

Blue: shared Sagai core. Teal: added by Architecture.

- Every part has an accountable role
- Effort fits the stakes
- Decisions recorded with reasons
- Reviewed on a schedule
- Roles, not people
- Traceable to drivers
- Current and target both recorded
- Every standard has a deviation route
- Decisions state what they cost to reverse
- The landscape matches what actually runs

GATES, AUTONOMY LEVELS, ESCALATION

6 DECISION GATES each passed when a person records a decision

AG1 | AG2 | AG3 | AG4 | AG5 | AG6

Assess, Choose, Design, Start of Build, Test, Put into use. AG3 and AG6 need leadership by default. Improve has no gate.

3 AUTONOMY LEVELS limits, not a ladder to climb

A Advisory | B Drafting | C Checking

At every level an agent only prepares. It cannot decide, approve, publish or send, or change a live document or register.

7 ESCALATION TRIGGERS The agent stops and hands the item to a person in a set role. All recorded in the Gates and Escalation Record.

AREAS, 120 REFERENCE TYPES

1. Enterprise
2. Business
3. Application and solution
4. Data and information
5. Technology and infrastructure
6. Integration and API
7. Cybersecurity and privacy
8. AI and intelligent automation
9. Software engineering, delivery and operations
10. Resilience and continuity

Ten cards, each with its types, copied into the organisation's own register and changed to suit. Not 120 disciplines. Single-system types are listed for categorising only.

THE LANDSCAPE, AND OTHER FRAMEWORKS

Register. Items, dependencies, current and target state, steps between, debt. Started in Assess, compared with what runs at Test, kept current.

Threshold. What a decision costs to reverse sets the review depth. A decision record where it is met.

Strategy | Governance | ADLC

Meets other frameworks at set points. Strategy comes in as drivers and constraints. Governance decides who is accountable and how it is checked. ADLC runs the life of each agent. None is needed to use Architecture.

 CC BY 4.0, (c) 2026 Aquasila Pty Ltd