trevaun

AI-Assisted Architecture Decision Kit

A structured framework for using AI to make better technical decisions, faster.

- The 5-step AI-assisted decision loop
- 10 ready-to-use prompts for architecture reviews
- A structured discovery question bank
- A fill-in-the-blank decision summary template

For developers, architects, and consultants.

THE DECISION LOOP

Use this before every major technical decision. The order matters.

1

Declare the goal and constraints

Tell AI what success looks like and what constraints are real: budget, SLAs, compliance, existing tech stack. This prevents AI from inventing a world that doesn't exist.

2

Ask for questions, not answers

Before solutions, surface what's missing. Ask AI for the top clarifying questions grouped by domain. This upgrades discovery and protects you from premature design.

3

Generate options with explicit tradeoffs

Ask for 2 to 4 viable approaches. Require benefits, risks, operational complexity, cost drivers, and the most likely failure mode. Build a decision record, not a collection of ideas.

4

Stress test the leading option

Pick the most likely path and ask AI to attack it. What could go wrong? What assumptions are unsafe? What should you validate in a POC?

5

Validate reality, then draft deliverables

Only after reasoning is solid, use AI to draft ADRs, implementation plans, runbooks, and risk registers. This is where AI saves time without taking over thinking.

AI can draft outputs. You must own the reasoning.

READY-TO-USE PROMPTS

Copy these directly into your AI tool. Customize the bracketed sections.

Discovery and scoping

Prompt 1: Constraint declaration

"I need [type of solution] that supports [performance target], stays within [budget], and meets [compliance requirements]. The client is on [tech stack]. [Additional constraints]. Before suggesting anything, confirm you understand these constraints and flag any that conflict."

When to use: At the start of every engagement. Forces clarity and prevents hallucinated requirements.

Prompt 2: Clarifying question generator

"Given these constraints: [paste]. What are the top 15 clarifying questions I should ask before recommending an approach? Group by: performance, security, data model, integration, and operations. Flag which are blockers vs. nice-to-know."

When to use: Before discovery workshops. Ensures you don't miss critical questions.

Prompt 3: Assumption surfacer

"Here is my current understanding: [paste context]. List every assumption I am making, explicit and implicit. For each, rate as confirmed, likely, or unverified. For unverified, suggest how to validate quickly."

When to use: After initial discovery, before committing to a design direction.

Architecture and design

Prompt 4: Multi-option comparison

"Given these constraints: [paste]. Give me 3 architecture options for [problem]. For each: benefits, risks, operational complexity, estimated cost range, scaling ceiling, and the single most likely failure mode in production. Present as a comparison table."

When to use: When evaluating design approaches. Builds a decision record.

Prompt 5: Red team / stress test

"Assume we choose [option]. Act like a skeptical staff engineer. List: (1) what could go wrong in the first 90 days, (2) what assumptions are unsafe, (3) what to validate in a POC, and (4) the operational burden this creates."

When to use: After selecting a leading option. Catches blind spots before production.

Prompt 6: STRIDE threat model

"Here is the target architecture: [describe]. Perform a threat model using STRIDE. For each threat category, list specific risks and suggest mitigations that fit a [cloud-native / on-prem / hybrid] setup."

When to use: For any system handling sensitive data or requiring compliance.

Prompt 7: Scaling bottleneck finder

"Current architecture: [describe]. Current load: [X]. Expected growth: [Y] over [timeframe]. Identify the top 5 scaling bottlenecks by likelihood. For each, explain the failure mode and suggest a mitigation that doesn't require re-architecture."

When to use: During architecture reviews or capacity planning.

Deliverables and communication

Prompt 8: ADR generator

"Draft an Architecture Decision Record for: [describe]. Include Context, Decision, Options Considered with pros/cons, Consequences, and Status. Short paragraphs. The audience is technical peers reading this in 6 months."

When to use: After finalizing any significant architecture decision.

Prompt 9: Executive summary

"Draft a one-page executive summary for this decision: [describe]. Include business context, tradeoffs, operational impact, and what we need from leadership. Non-technical but accurate. The reader has 5 minutes."

When to use: When communicating technical decisions to non-technical stakeholders.

Prompt 10: Risk register builder

"Based on this project plan: [describe]. Build a risk register: Risk Description, Likelihood, Impact, Mitigation Strategy, Owner, Status. Include at least 10 risks across technical, operational, organizational, and timeline categories."

When to use: At project kickoff and during phase transitions.

DISCOVERY QUESTION BANK

Use as a workshop guide for CRM and integration projects. These are the questions that prevent scope surprises.

Security and compliance

- What compliance frameworks apply (SOC 2, HIPAA, FedRAMP, GDPR)?
- What is the current identity and access management setup? SSO provider?
- Are there data residency requirements or cross-border transfer restrictions?
- Who owns the security review process, and what is the approval timeline?
- What are the data classification levels? Which fields are sensitive?
- Are there penetration testing or vulnerability scanning requirements?

Integrations and APIs

- What systems need integration? List all, including legacy.
- For each integration: protocol (REST, SOAP, file-based, event-driven)?
- Expected transaction volumes and peak load patterns?
- Who owns and maintains each integration endpoint?
- Existing middleware or iPaaS platforms (MuleSoft, Boomi, etc.)?
- Error handling and retry strategy for failed integrations?
- SLAs on data freshness or synchronization latency?

Data model and migration

- Current data model? Is documentation up to date?
- How much historical data migrates, and what is the cutoff?
- Data quality issues today (duplicates, missing fields, format inconsistencies)?
- Who owns data governance and master data management?
- Field-level security or record-level sharing requirements?
- Rollback plan if migration introduces integrity issues?

Reporting and analytics

- Critical reports and dashboards needed at go-live?
- Who are the primary report consumers? What decisions do they make?
- Existing BI tools (Tableau, Power BI) that need connection?
- Data retention and archival policies?
- Real-time reporting requirements vs. batch/scheduled?

Support operations and change management

- Who is the internal Salesforce admin? What is their capacity?
- Training plan for end users?
- Change management process for post-go-live enhancements?
- Escalation path for production issues?
- What does success look like 90 days after go-live? How will you measure it?

DECISION SUMMARY TEMPLATE

Fill in each section after completing the decision loop. Drop into any ADR or decision document.

Decision title	[One sentence: what was decided]
Date	[Date of decision]
Status	[Proposed / Accepted / Superseded]
Context	[Why this decision is needed. What problem? What triggered it? 2 to 3 sentences.]
Constraints	[Non-negotiable requirements: budget, SLAs, compliance, tech stack, timeline.]
Options considered	[Option A: description + key tradeoff] [Option B: description + key tradeoff] [Option C: description + key tradeoff]
Selected approach	[Which option and why. Reference the constraints it satisfies.]
Key risks and mitigations	[Risk 1: mitigation] [Risk 2: mitigation] [Risk 3: mitigation]
Assumptions	[List assumptions. Flag which are unverified.]
What we will measure	[KPIs or success criteria. Include timeframe.]
Stakeholders	[Who was involved or informed]

Keep it short. If a section takes more than 3 sentences, you're over-explaining. The audience is future-you in 6 months.

ABOUT TREVAUN

Salesforce consulting and architecture, built with AI.

Trevaun is a Salesforce consulting and architecture practice in the Washington DC metro area. We implement and integrate Salesforce, build with AI agents, and specialize in access, security, and dependency governance. Every engagement is led by a certified Salesforce architect.

Our approach pairs deep Salesforce experience with the AI-assisted decision frameworks in this kit, so every recommendation is backed by structured reasoning, not just experience.

Planning a Salesforce implementation, integration project, or architecture review?

Let's talk about how this framework applies to your program. We reply within one to two business days.

Email: hello@trevaun.com
Web: trevaun.com
LinkedIn: linkedin.com/company/aetrum-llc

assess. build. govern.

© 2026 Aetrum LLC. All rights reserved. Trevaun[SM] is a trade name and service mark of Aetrum LLC, a Virginia limited liability company. Ashburn, Virginia.