

# Solution Architect Playbook Template

## 1. Header

- **Domain(s) Supported:**
- **Version / Date:**
- **Author / Owner:** Lead Architect
- **Audience:** Solution / Product Architects (5–6 squads each)

## 2. Purpose

- Define the **role expectations** for solution/product architects in our domains.
- Standardize **artifacts, practices, and responsibilities** to ensure quality and consistency.
- Provide **practical guidance** to balance **strategic alignment** with **day-to-day squad support**.

## 3. Roles & Responsibilities

- **Requirements Shaping:** Work with PMs and BAs to identify architecture-significant requirements (security, compliance, performance).
- **Solution Design:** Produce solution design docs aligned to domain target state and enterprise standards.
- **Governance:** Present to Domain Council/ARB when required; log exceptions and decisions.
- **Delivery Oversight:** Support squads during design/build, answer architectural queries, ensure incremental delivery.
- **Strategic vs Tactical Balance:**
  - 50% strategic alignment (roadmaps, reuse, target state).
  - 50% tactical delivery (design reviews, squad support, BAU fixes).

## 4. Squad Engagement Model

- **Coverage:** Each architect supports 5–6 squads.
- **Engagement Rhythm:**
  - PI Planning: participate in shaping epics and enablers.
  - Squad Refinement: attend only when architecture-heavy stories appear.
  - Design Clinics: run weekly office hours for squads.
- **Priority Management:** Focus on **regulatory-impacting work, cross-squad dependencies, and tech debt** before BAU asks.

## 5. Core Artifacts to Produce

- **Solution Design Document (SDD)** – key flows, integrations, security.
- **Integration Maps** – APIs, Kafka topics, batch flows.
- **Security & Compliance View** – data classifications, regulatory mappings.
- **Data Flow Diagrams** – lineage of critical data (shareholder, tax, performance).
- **Architecture Decision Records (ADR-lite)** – log key trade-offs.
- **Tech Debt Log** – items deferred, escalated to Domain Council.

## 6. Practices & Ways of Working

- **With Product Managers:** Align features with architecture runway.
- **With Engineering Leads:** Partner on feasibility, performance, and delivery slices.
- **With BAs:** Ensure requirements trace to design.
- **With Ops:** Validate designs for resilience, SLA, observability.
- **Ceremonies:** PI Planning (mandatory), Refinement (selective), ARB/Council (as needed), Retros (feedback loop).
- **Ways of Working:**
  - Use **reuse-first** mindset (patterns, APIs, data models).
  - Document **exceptions transparently**.

- Maintain visibility across squads via shared dashboards.

## 7. Governance Alignment

- **Domain Level:** Align to target state; track domain-level risks/tech debt.
- **Enterprise Level:** Adopt enterprise patterns (cloud, API-first, lineage).
- **Regulatory Alignment:** Ensure designs generate evidence for audits (lineage reports, reconciliation logs).

## 8. KPIs / Success Measures

- % of solutions aligned to domain/enterprise standards.
- % reduction in exceptions/tech debt.
- Stakeholder feedback (PMs, Ops, Compliance).
- Delivery predictability (architectural enablers delivered on time).
- Audit/regulator findings tied to design = zero.

## 9. Flexibility Guidance

- **Lightweight Documentation:** Use ADR-lite + diagrams for small changes; full SDD for regulator-facing solutions.
- **Domain-Specific Needs:**
  - Shareholder Reporting → emphasize SEC schema compliance.
  - Tax → focus on audit evidence and reconciliations.
  - Performance & Attribution → lineage and benchmark mapping.
  - Alternatives → flexibility for evolving data models.