



There Is No Perfect Architecture.

Only trade-offs

“

Every architecture decision
is a silent bet.

Most teams don't know what they've bet
until it fails.

— Gregor Hohpe, *The Software Architect Elevator*

Six Polarities Every Architect Navigates

There is no right answer. There is only the tension which is managed



Each one is a dial, not a switch. Where you set it depends on context, not just best practice.

Business Impact vs. Technology Outputs

BUSINESS IMPACT



TECHNOLOGY OUTPUTS

The Architecture Nobody Uses

A data platform team spends 18 months building a technically elegant lakehouse. Correct medallion layers. Clean lineage. Beautiful metadata.

Nobody queries it. The business still runs off a spreadsheet the finance team built in 2019.

Architecturally excellent. Strategically invisible.

Ask your team:

Can we name the specific business decision this architecture enables?

If the platform disappeared tomorrow, what would the business actually miss?

⚡ AGENTIC AI

Agents optimise for measurable proxies. If the goal isn't clearly defined in the architecture, agents will pursue the wrong thing at speed. Goodhart's Law at machine velocity.

The hardest architectural question isn't 'Is it well-built?', it's 'Does anyone need it?' The agentic version: 'Does the agent know what it's actually for?'

Standardisation vs. Flexibility

STANDARDISATION



FLEXIBILITY

The IS-U Problem — Every Industry Has One

SAP IS-U was the right opinionated choice in 1998. One platform, one truth. Then came deregulation. Rooftop solar. EVs. Time-of-use tariffs.

The system that felt like simplicity is now the reason every downstream migration is a multi-year program.

Simplicity has a compounding interest rate. You don't pay it now your successor does.

Ask your team:

Are we standardising because it creates genuine value or because it makes the platform team's life easier?

Who carries the cost of this standard in 5 years?

⚡ AGENTIC AI

Agents need standardised tool interfaces (APIs, MCP contracts) to act reliably. But over-standardise and agents fail on edge cases. Under-standardise and agent behaviour becomes unpredictable at scale.

Martin Fowler's irreversibility premium: price in the cost of undoing a decision before you make it. For agentic systems, that window is much shorter.

Empowerment vs. Control

EMPOWERMENT



CONTROL

The Self-Service Shadow IT Cycle

You open self-service analytics. Business teams love it. Dashboards multiply. Then the CFO gets three different revenue numbers from three different dashboards built by empowered users with different definitions of 'revenue'.

You tighten control. Business teams route around it. Shadow IT returns.

The tension was never resolved, it was just relocated.

Ask your team:

Where does accountability for data quality live when users define their own metrics?

Are we solving self-service or deferring a governance problem?

⚡ AGENTIC AI

With humans, you can course-correct. When an agent acts autonomously calling tools, spawning sub-agents, writing to systems human-in-the-loop becomes an explicit design decision, not a default. Who is accountable when the agent is wrong?

Empowerment without shared definitions isn't democratization, it's fragmentation. Agentic empowerment without explicit accountability boundaries is something far more dangerous.

Protection vs. Practicality

PROTECTION

The Governance Framework Nobody Follows

The team builds a comprehensive data governance framework. 47-page policy. Classification tiers. Access request workflows. Mandatory steward sign-off.

Six months later: nobody follows it. Not maliciously it's just too heavy to use.

Fit-for-purpose governance means calibrating controls to actual risk. Not applying maximum protection uniformly.

PRACTICALITY

Ask your team:

Are our controls proportionate to actual data sensitivity or to our fear of being blamed?

What is the cost of a governance framework so heavy nobody uses it?

⚡ AGENTIC AI

Agents need broad context to be useful. But broad access creates massive exposure when the agent acts autonomously across systems. Least-privilege becomes nearly impossible when you can't predict in advance what an agent will need.

A governance model that prevents business activity isn't protecting the organisation — it's a different kind of liability. With agents, the liability gap is exponentially wider.

Agility vs. Stability

AGILITY



STABILITY

The SCADA Story

We replicated real-time SCADA telemetry into a cloud lakehouse. Modular. Decoupled. Agile.

Until the fault detection model made network decisions on data 15 minutes stale. We built for agility in the architecture. We forgot we needed stability in the data.

Agility shifts cost from build time to runtime and from architects to operations.

Ask your team:

Are we building for the change we expect next year or the stability we need today?

Who is on-call at 2am when the modular architecture fragments?

⚡ AGENTIC AI

Multi-agent systems produce emergent behaviours no single agent's design predicts. Modular architectures fine for data pipelines create new runtime instability classes when the modules are autonomous actors reasoning and acting independently.

Agility is not free. It shifts cost from build time to runtime and from architects to the people keeping the lights on at 2am.

What Agentic AI Actually Changes

These polarities existed before. Agentic AI doesn't create new trade-offs, it removes the safety net.

Design-time decisions



Runtime governance questions

Human course-correction



Explicit accountability architecture

Periodic audit



Continuous compliance in the loop

Predictable behaviour



Emergent multi-agent dynamics

The meta-shift: your architecture must now encode your organisation's values, not just its data flows.

The Decision Lens

Before you set any of these dials, ask these four questions. They matter more with agents in the system.

01

What is the dominant constraint right now?

Time, cost, talent, regulation. They're not equal, and they change.

02

What does it cost to reverse this in 2 years?

Fowler's irreversibility premium. Agentic decisions compound faster.

03

Who bears the pain of this trade-off?

Builder, operator, or business? Also: which downstream system?

04

Are we making this trade-off explicitly?

Or will we discover it when an agent does something we didn't anticipate?

**Most architectural pain
doesn't come from making
the wrong trade-off.**

***It comes from making one
unconsciously.***

The lakehouse nobody used. The IS-U system nobody planned to outlive. The self-service tool that created shadow governance. The AI platform compliance blocked. The governance framework too heavy to follow. The SCADA pipeline that traded stability for agility.

All wrong silently. Nobody wrote down what they were giving up. With agents in the system, that silence is no longer affordable.



The best architects aren't the ones who find the perfect design.

They're the ones who can say clearly:

"Here's what we're trading away, here's why that's the right sacrifice for now, and here's the signal that will tell us we got it wrong."

References

Hohpe, G.

The Software Architect Elevator

O'Reilly, 2020 — The whiteboard-to-production gap

Fowler, M.

Evolutionary Architecture

martinfowler.com — Reversibility premium and architectural fitness functions

Brewer, E.

Towards Robust Distributed Systems

CAP Theorem — PODC Keynote, 2000

Dehghani, Z.

Data Mesh

O'Reilly, 2022 — Domain ownership, federation, and accountability



Empowering South Australia