

Architecture Anti-Pattern Catalog

16 patterns • 4 domains • Symptoms • Remediation

Governance

1. Architecture by Committee
2. Ivory Tower Architect
3. Shelfware Blueprint
4. Governance Theater

Integration

5. Integration Spaghetti
6. Golden Hammer ESB
7. Vendor Lock-in Trap
8. Synchronous Quicksand

Data & Modeling

9. God Object
10. Data Lake Swamp
11. Copy-Paste Architecture
12. Phantom Requirements

Infrastructure

13. Lift-and-Shift Cloud
14. Microservice Mania
15. Security as Afterthought
16. Resume-Driven Design



Governance & Integration Anti-Patterns

1. Architecture by Committee

Symptoms: Decision paralysis, 6+ month review cycles, watered-down standards.

Fix: Define clear RACI. Empower domain architects. Reserve committee for strategic decisions only.

2. Ivory Tower Architect

Symptoms: Models no developer reads, target states nobody can reach, growing EA irrelevance.

Fix: Embed architects in delivery teams. Co-create models. Validate with implementation feedback.

3. Shelfware Blueprint

Symptoms: Beautiful 200-page document, zero implementation, next rewrite in 2 years.

Fix: Publish incrementally. Link every architecture decision to a funded initiative.

4. Governance Theater

Symptoms: Every proposal approved, no exceptions recorded, compliance is a checkbox.

Fix: Publish rejection rate. Record architecture decisions (ADRs). Review with teeth.

5. Integration Spaghetti

Symptoms: Any change breaks 5 systems. Nobody knows all the interfaces. Testing is impossible.

Fix: Introduce integration platform. Enforce canonical data models. API-first design.

6. Golden Hammer ESB

Symptoms: ESB is the bottleneck. Simple REST call routed through 7 ESB transformations.

Fix: Right-size integration: direct API for simple, bus for complex orchestration.

7. Vendor Lock-in Trap

Symptoms: Switching vendors requires full rewrite. Vendor controls roadmap and pricing.

Fix: Anti-corruption layers. Standard interfaces (OpenAPI). Evaluate exit cost before signing.

8. Synchronous Quicksand

Symptoms: Cascading timeouts. $99.9\%^{*10} = 99\%$ availability. Peak load causes avalanche failures.

Fix: Default to async/events. Use circuit breakers. Synchronous only where truly required.

Data/Modeling & Infrastructure Anti-Patterns

9. God Object / God Service

Symptoms: Single service owns 40 tables. Every change is high-risk. Cannot scale independently.

Fix: Decompose by bounded context. Apply single responsibility. Domain-driven design.

10. Data Lake Swamp

Symptoms: Nobody trusts the data. Analysts spend 80% cleaning. No lineage or ownership.

Fix: Implement data zones (raw/curated/published). Assign stewards. Enforce metadata standards.

11. Copy-Paste Architecture

Symptoms: Price calculation in 4 systems, each slightly different. Bug fix needed 4 times.

Fix: Identify canonical services. Centralize shared logic. Publish as reusable service/library.

12. Phantom Requirements

Symptoms: Over-engineered abstractions. 10x complexity for hypothetical future needs.

Fix: YAGNI principle. Build for known requirements. Refactor when new needs emerge.

13. Lift-and-Shift Cloud

Symptoms: Cloud costs exceed on-prem. No elasticity, no managed services, no cloud benefits.

Fix: Cloud-native roadmap. Prioritize workloads for re-platform vs re-architect vs retain.

14. Microservice Mania

Symptoms: Distributed monolith. 500 services, 3 developers. Debugging across 12 services.

Fix: Start with well-structured monolith. Extract services when team/scale justifies it.

15. Security as Afterthought

Symptoms: Critical vulnerabilities found too late. Expensive rework or accepted risk.

Fix: Shift-left: threat modeling in design, DevSecOps pipeline, security review at inception.

16. Resume-Driven Design

Symptoms: Kubernetes for a 3-page website. Kafka for 10 events/hour. Team cannot maintain it.

Fix: Technology radar (adopt/trial/assess/hold). Match tech to actual scale and team skill.

Build governance that prevents these anti-patterns

nilus.be/services/togaf-certified-training | nilus.be/services/archimate-training | nilus.be/contact