

FICTIONAL COMPOSITE / REAL PRACTITIONER PATTERN

THE TEMPORARY FIREWALL RULE THAT BECAME ARCHITECTURE

The exception had an expiration date. The dependencies it attracted did not.

01

**The exception had an expiration date.
Its dependencies did not.**

02

A fourteen-day migration bridge stayed quiet enough to become normal.

03

**New jobs and templates discovered
the path. The exception became
infrastructure.**

04

Annual review changed the date but never reconstructed the current consumers.

05

Nobody knew enough to remove it, so uncertainty became the reason to keep it.

06

**The team mapped actual traffic,
owners, templates, and unknown flows.**

07

**Retirement became reversible
experiments instead of one heroic
outage gamble.**

08

**Track exceptions as changing
architecture, not stationary
documents.**

THE NEXT USEFUL QUESTION

Take the oldest live exception and compare its current consumers, owners, and configuration with the original approval before renewing it again.

FIND / MAP / EXPLAIN / REDUCE / PROVE