

FOUR INCIDENTS. FOUR CAUSES. ONE OUTCOME.

Why blast radius — not root cause — is the property nobody's architecting for

THE ARCHITECT · RACK2CLOUD

FOUR INCIDENTS, FOUR CAUSES

Different systems. Different teams. Different postmortems.



COOLING FAILURE

Northern Virginia · May 2026

Chiller hardware failure, loss of cooling



THIRD-PARTY NETWORK

June 2026

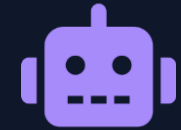
External peer dependency, not internal AWS



ROUTING FAULT

AWS US-WEST-2 · Jul 24

Region-boundary networking hardware



AUTOMATION SCOPE

Azure West US · Jul 23

Isolation job scoped too broadly

Different causes.
Same customer experience.



PROCEDURAL VS. ARCHITECTURAL

Two different questions after every incident.

PROCEDURAL	ARCHITECTURAL
What failed?	What allowed this failure to matter?
How do we stop this cause?	How do we constrain impact regardless of cause?
Fix the trigger	Bound the blast radius
Incident-specific	Failure-mode-independent

The guardrail worked.

The outage happened anyway.



THE CENTRAL OBSERVATION

The failure wasn't that nobody checked.

The failure was that the architecture allowed a passing check to leave the system exposed.

WHY GUARDRAILS AREN'T ARCHITECTURE



PROCEDURAL

Guardrails reduce the chance of a known mistake.



ARCHITECTURAL

Architecture limits the damage of mistakes you haven't imagined yet.

DIAGNOSTIC QUESTION

After the second incident, what changed?

If the answer is a document, a process, or a review board — the third incident is still waiting for you.

If the answer is system behavior, blast-radius boundaries, or authority distribution — you're probably changing architecture.

The recurring pattern isn't the failure.

It's the blast radius.

Full breakdown: The Third Incident Is the One That Should Worry You

rack2cloud.com/cloud-blast-radius