

RUN A DIAGRAM RECONCILIATION DRILL

Do not ask whether the diagram is current. Choose one important path and make the diagram prove it.

PRACTITIONER POSITION

A diagram reconciliation drill compares one bounded architecture path against live configuration, inventories, data-flow knowledge, administrative access, and system ownership, then records the exact corrections and unknowns instead of commissioning another broad redraw.

THE FIELD PATTERN

WHAT THIS LOOKS LIKE IN PRACTICE

The network diagram shows a payment application, database, and segmented boundary. The cloud inventory contains two additional accounts. The identity team knows administrators enter through a shared access service. The logging team receives events through a route that is missing from the drawing. Everyone agrees the diagram is broadly right, but no one has tested whether one complete path is right from end to end.

SIGNALS DETECTED

WHAT SHOULD MAKE YOU STOP

- The diagram has a recent date but no record of what was verified.
- Data, administrative, monitoring, and deployment paths live in separate artifacts that are never reconciled.
- System names on the diagram do not match cloud, inventory, or evidence identifiers.
- Teams correct the picture during assessment interviews without routing those corrections back to the maintained source.
- Unknown connections are removed from the drawing before they are resolved.

WHY IT MATTERS

THE COST OF LEAVING IT UNEXPLAINED

A diagram is a maintained representation, not an oracle. The goal of the drill is to improve the relationship between the representation and observable architecture facts. One verified path produces more useful confidence than a wholesale redraw whose assumptions remain untested.

The drill also creates a repeatable maintenance mechanism. Each discrepancy receives an owner, source, decision, and date so the organization can distinguish a corrected picture from an unresolved architecture question.

QUESTIONS

QUESTIONS THAT EXPOSE THE WACK

- 01 Which business or security path is important enough to verify from origin to destination today?
- 02 What live configuration, inventory, owner knowledge, and telemetry can corroborate each hop?
- 03 Which identifiers or boundaries disagree across the sources, and what fact will resolve each difference?
- 04 Who updates the maintained diagram and who verifies that the correction matches the environment?

THE CLUSTER OF WACK

FIND / MAP / EXPLAIN / REDUCE / PROVE

ESCAPE THE WACK

01 / FIND

Select one bounded path and gather the diagram, inventory, configuration, access, telemetry, and ownership records that describe it.

02 / MAP

Walk every data, administrative, security-service, and control-plane hop; mark its source and last verification date.

03 / EXPLAIN

Record discrepancies as explicit questions rather than silently choosing the most convenient representation.

04 / REDUCE

Retire duplicate diagrams, stale labels, and unnecessary paths revealed during reconciliation.

05 / PROVE

Publish the corrected path with verification evidence, unresolved items, owners, and the next review trigger.

THE FIELD LAW

THE LINE WORTH REMEMBERING

A DIAGRAM BECOMES CURRENT ONE VERIFIED PATH AT A TIME.

Which path would expose the most Diagram Drift if you reconciled it tomorrow?

TECHNICAL GROUNDING

START WITH THE SOURCE MATERIAL

These references ground the practitioner interpretation. They do not turn the metaphor or method into official PCI DSS terminology.

- WACK-SOURCE-0001 / PCI DSS v4.0.1 / PCI Security Standards Council / [Open source](#)

AUTHORITY CLASSIFICATION / PCI_SSC_SOURCE / PRACTITIONER_POSITION / PRACTITIONER_METHOD / AI_ASSISTED

Practitioner reviewed by Scott Norton on 2026-09-09 against canonical digest

6e32a14dff41124824ebd66e721e698b1c21a3fe1877ca5e4b85cf06cf588e69. Educational content, not a compliance determination.

Independent practitioner resource; not affiliated with or endorsed by PCI SSC. Authorship and authority classification do not determine legal intellectual-property ownership.