

YOUR PCI SCOPE IS A GRAPH. STOP FLATTENING IT.

An asset list can tell you what exists. A scope graph can tell you why it matters, what it can affect, which claim depends on it, and what must reopen when it changes.

PRACTITIONER POSITION

My practitioner position is that PCI scope is best operated as a graph of dated, directional, evidenced relationships. It is not merely a list of systems marked in or out. A useful scope model begins with payment journeys and account-data flows, then connects the systems, services, identities, networks, facilities, provider responsibilities, segmentation boundaries, administrative paths, public points of presence, control occurrences, evidence populations, tests, and assessment claims that make the scope conclusion defensible. Every important edge answers a complete sentence: this subject has this relationship to that object, for this reason, during this period, according to this authority, supported by this evidence, with this owner and review state. The graph does not replace the inventory, diagram, responsibility matrix, evidence repository, or assessor judgment. It gives those artifacts stable objects and shared relationships so they can tell the same story. A spreadsheet remains useful as a view or editing surface. It becomes dangerous when the row is mistaken for the relationship model underneath it.

THE FIELD PATTERN

WHAT THIS LOOKS LIKE IN PRACTICE

A new cloud deployment service is introduced to release code into an application connected to the CDE. The service does not store account data, so the asset owner marks it out of scope. The network diagram does not show it because it is a control-plane service. The identity team can show privileged roles but not which deployment targets they can change. The TPSP inventory lists the cloud provider but not the exact managed service or customer responsibility. The penetration-test scope was copied from last year's host list. The GRC tool retains green controls and last year's evidence because no requirement text changed. Every artifact is locally plausible. Together they hide the new influence path. In a scope graph, the deployment service administers the application, the privileged identities can invoke the service, the provider supplies part of the service, customer configuration governs the remainder, the change affects the CDE-connected application, and those edges reopen the relevant scope decision, evidence populations, tests, and claims. Nothing needs to become universally red. The affected relationships need to become visible.

WHAT SHOULD MAKE YOU STOP

- The scope record has one status field but no relationship stating why the object is included, excluded, separately covered, or unresolved.
- Payment journeys, data flows, administrative paths, and security-influence paths live in different artifacts without shared stable identifiers.
- A provider is listed as one row even though different services have different responsibilities, limitations, data paths, and customer-controlled configurations.
- Identities are reviewed by directory role while the systems, applications, data, and control planes those privileges can affect remain disconnected.
- The network diagram shows traffic paths but not deployment, orchestration, identity, logging, backup, key-management, or terminal-management influence paths.
- A change updates inventory but cannot identify the diagrams, scan targets, penetration-test scope, evidence populations, responsibility statements, and claims that may now be stale.
- Evidence is stored against a requirement or control but not against the bounded subject, population, period, relationship, and claim it supports.
- Excluded objects disappear from the model instead of retaining the tested reason, evidence, authority, review date, and trigger that could invalidate the exclusion.

WHY IT MATTERS

THE COST OF LEAVING IT UNEXPLAINED

Start with node identity. A graph node is an addressable thing whose identity must survive across artifacts: a payment journey, data flow, system component, software product, identity population, network or facility, provider service, segmentation control, public endpoint, control occurrence, evidence population, test, claim, or report conclusion. The node needs a stable identifier, name, type, accountable owner, lifecycle state, valid-from and valid-to dates, and source system. Do not force materially different things into one universal asset type merely because the platform has only one object table.

Then make the edges carry meaning. Uses, stores, processes, transmits, connects to, administers, authenticates to, can affect security of, is segmented by, is provided by, is responsible for, is tested by, produces evidence for, supports, limits, contradicts, and concludes are not interchangeable links. Direction matters. 'Application depends on identity provider' is not the same proposition as 'identity administrator can change application access.' Each edge needs a reason, effective period, state, owner, evidence reference, authority classification, review disposition, and reopening trigger. An unlabeled line is decorative architecture.

Preserve scope as a conclusion over relationships, not a permanent property of the node. The same component may participate in one payment journey, administer another, be separately assessed for a provider service, and remain unrelated to a third. Its relevance can change when routing, privilege, service configuration, responsibility, segmentation, or account-data behavior changes. Record the boundary conclusion for the specific context and time slice. Keep excluded and unresolved candidates visible because

absence from the graph cannot prove exclusion.

Join the graph to operating time. Recurring activities, inventories, scans, penetration tests, access reviews, provider validations, configuration reviews, and evidence collections should point to the population they were expected to cover at that occurrence. When the graph changes, the platform can compare the effective population before and after the event. That comparison identifies exactly which evidence remains representative, which test scope needs reconsideration, and which claim has lost support. This is more useful than resetting every control or allowing every green status to survive.

Join the graph to claims. A scope conclusion should be traversable from assessed-entity representation through the relevant nodes and edges to supporting evidence, practitioner analysis, assessor testing, limitations, and report treatment. This does not automate professional judgment. It makes the inputs and consequences of judgment visible. The assessor can explain why a system is relevant; the entity can challenge a mistaken edge; and the compliance-accepting entity can see the bounded conclusion without being asked to infer the architecture from a spreadsheet attachment.

Test the graph with change. Add a deployment service, remove a segmentation control, expire a provider validation, change an identity role, discover an undocumented public endpoint, or fail a sampled occurrence. The product should traverse only the affected relationships and present a review queue: boundary decisions, inventory views, diagrams, responsibility seams, occurrence populations, scan targets, penetration-test assumptions, evidence requests, tests, and report claims that may require action. The desired output is controlled reopening-not silent survival and not project-wide panic.

QUESTIONS

QUESTIONS THAT EXPOSE THE WACK

- 01 What are the stable node types in the PCI operating model, and which real objects are currently flattened into a generic asset or control row?
- 02 Which relationship predicates express account-data movement, connectivity, administration, security influence, provider delivery, responsibility, testing, evidence, and claim support?
- 03 Can every scope-relevant edge state why it exists, when it was valid, who owns it, what evidence supports it, and what change would reopen it?
- 04 Can one component have different scope relationships for different payment journeys, services, and time periods without duplicating or overwriting its identity?
- 05 Where are excluded and unresolved candidates retained, and which tested proposition supports each current disposition?
- 06 Can an assessor traverse from a report conclusion backward through claim, test, population, evidence, control occurrence, responsibility, component, and payment journey?
- 07 When one node or edge changes, can the system identify the exact diagrams, inventories, scans, tests, evidence requests, and conclusions that may no longer be representative?
- 08 Which relationships still exist only in oral history, a diagram legend, a spreadsheet formula, or a sentence beginning with 'everybody knows'?

ESCAPE THE WACK

01 / FIND

Inventory the distinct things and propositions hidden inside current asset lists, diagrams, provider inventories, control rows, evidence folders, and assessment workpapers. Give every durable object and every material scope assertion a candidate identity.

02 / MAP

Build the first graph from one payment journey. Add account-data and influence paths, components, identities, networks, provider services, responsibilities, boundary controls, occurrences, evidence populations, tests, and claims. Use directional predicates rather than generic links.

03 / EXPLAIN

For every material edge, record reason, effective period, owner, evidence, authority, disposition, and reopening trigger. Publish which views are inventory, architecture, scope, responsibility, assessment, and reporting projections of the same graph.

04 / REDUCE

Remove duplicate nodes, ambiguous predicates, orphan evidence, permanent scope labels, and invisible exclusions. Prefer one stable relationship with several governed views over several artifacts that silently disagree.

05 / PROVE

Run a change-impact drill. Introduce one new security-influencing service and traverse the graph to the exact inventory entries, diagrams, provider duties, populations, scans, tests, evidence requests, reviews, and report claims that must be confirmed or reopened.

THE LINE WORTH REMEMBERING

SCOPE IS NOT A COLOR ON A ROW. SCOPE IS A CONCLUSION OVER RELATIONSHIPS THAT MUST SURVIVE TIME, EVIDENCE, AND CHANGE.

Download the Scope Graph Node and Edge Ledger. Model one payment journey, preserve exclusions, and inject one change. If downstream claims cannot be traced, the graph is still a picture.

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](#)

- WACK-SOURCE-0005 / PCI DSS Scoping and Segmentation Guidance for Modern Network Architectures / PCI Security Standards Council / [Open source](#)

- WACK-SOURCE-0008 / FAQ 1473: Role of compliance-accepting entities and assessors in determining applicability / 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

865e4c6a47f6f135be33a7044a9aab66c69a79494e6654a6b510bcbbf6888c87. 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.