

THE CLOUD SERVICE CHANGES. THE ASSESSMENT QUESTION DOES NOT.

AWS, Azure, and Google Cloud use different names for accounts, subscriptions, projects, identities, policies, logs, inventories, functions, containers, keys, and assurance portals. A service-name crosswalk can still leave the assessment unable to explain the claim.

PRACTITIONER POSITION

Cloud-provider terminology is necessary for operating each platform, but it is a poor primary key for a PCI DSS assessment. My practitioner position is that cross-cloud work should begin with a stable claim contract: the payment journey, exact subject and resource population, account-data and security-influence relationships, responsible parties, customer-controlled configuration, provider-performed activity, authoritative inventory source, identity and authorization path, network and public exposure, change history, logging and evidence period, test population, limitations, and reopening triggers. AWS Config and CloudTrail, Azure Resource Graph and Activity Log, and Google Cloud Asset Inventory and Cloud Audit Logs can answer related questions, but they do not have identical coverage, semantics, defaults, clocks, or retention. The translation must preserve those differences. The goal is not to pretend that cloud services are equivalent. It is to ask equivalent assessment questions, record provider-specific answers, and stop each conclusion where the available evidence stops.

THE FIELD PATTERN

WHAT THIS LOOKS LIKE IN PRACTICE

A company operates payment-adjacent workloads in AWS, Azure, and Google Cloud. The spreadsheet maps Lambda to Azure Functions to Cloud Run functions, IAM roles to Azure RBAC roles to Google Cloud IAM roles, and CloudTrail to Activity Log to Cloud Audit Logs. The rows look complete. Yet one AWS organization excludes two accounts from the Config aggregator; Azure Resource Graph queries only the subscriptions visible to the collector; Google Cloud Data Access logs are not enabled for every relevant service; workload identities are mixed with human administrators; an Internet-facing callback is represented only by a gateway resource; and each provider assurance package covers a different service population and period. The crosswalk translated nouns. It did not prove population, capability, responsibility, time, or claim fit.

SIGNALS DETECTED

WHAT SHOULD MAKE YOU STOP

- The assessment workbook maps provider service names but has no stable subject or claim identifier across platforms.

- A provider console export is called the complete inventory without recording organizations, tenants, subscriptions, accounts, folders, projects, regions, collectors, permissions, query time, unsupported types, or exclusions.
- Human identities, workload identities, service principals, managed identities, service accounts, assumed roles, federated identities, emergency paths, and application authorization are flattened into one user list.
- CloudTrail, Azure Activity Log, and Cloud Audit Logs are treated as interchangeable even though categories, enablement, access, retention, region or hierarchy coverage, and data-event behavior differ.
- The provider AOC and responsibility matrix are attached at the provider level rather than matched to the exact service, feature, region, account or tenant, period, and relied-upon activity.
- Managed compute is considered absent from inventory because instances are ephemeral, scaled to zero, or hidden behind a platform service.
- IaC, control-plane state, deployment identities, runtime configuration, and observed production resources disagree without a declared reconciliation rule.
- The same word-account, policy, role, project, group, function, log, public, managed-means materially different things across teams and providers.
- A dashboard reports framework alignment but cannot identify the exact resource population, configuration predicate, evidence period, exception set, or assessment procedure behind the result.
- A multi-cloud change can alter one payment path without reopening the equivalent inventory, diagram, responsibility, logging, testing, and evidence questions on the other platform paths.

WHY IT MATTERS

THE COST OF LEAVING IT UNEXPLAINED

Begin with the payment and security proposition, not the provider catalog. State the bounded claim in provider-neutral terms: which actor or service does what, to which resource or data, through which relationship, during which period, under whose responsibility, and with what expected result. Only then map the provider objects and evidence that can support that claim.

Use a common translation spine: organization boundary; account, subscription, or project container; resource identity; service and feature; environment and region; data relationship; security-influence relationship; human and workload identity; effective permission; public and private exposure; configuration source; provider and customer responsibility; inventory evidence; change evidence; operating evidence; test population; exception; owner; and reopening trigger. A blank provider cell means unresolved-not equivalent, not not-applicable, and not out of scope.

Preserve provider-native semantics. AWS accounts and Organizations, Azure tenants and subscriptions, and Google Cloud organizations, folders, and projects are not cosmetic labels for one generic container. Each hierarchy changes inheritance, visibility, policy, identity, aggregation, and evidence collection. The translation record should name the provider object and the stable assessment role it plays without erasing either.

Separate inventory from completeness. AWS Config, Azure Resource Graph, and Google Cloud Asset Inventory are powerful discovery and history sources, but their results depend on enablement, supported

resource types, hierarchy coverage, permissions, timing, consistency, aggregation, query, and exclusions. Record those boundaries beside the export. Reconcile the result with billing, organization hierarchy, IaC, deployment, DNS, public exposure, identity, and service-specific sources as applicable.

Separate change evidence from logging slogans. CloudTrail, Azure Activity Log, and Cloud Audit Logs each expose provider-specific control-plane and data-access signals with different categories and configuration choices. Identify which event class proves the claim, whether it was enabled, where it is retained, how the full period is queried, which identities and resource identifiers survive, and what the source explicitly cannot show.

Model effective capability rather than role names. An AWS assumed role, Azure service principal or managed identity, and Google Cloud service account or workload identity may each authorize automation, but the evidence chain still needs identity source, credential or federation path, trust policy, assignment, scope, conditions, session or token behavior, reachable targets, observed use, owner, and review history. Similar function does not mean identical evidence.

Treat managed services as responsibility reallocations, not disappearing infrastructure. The provider may operate hardware, hypervisor, host OS, control-plane components, or parts of a runtime. The customer may still control code, images, configuration, identities, network exposure, secrets, data, tenancy, logging, deployment, monitoring, testing, incident response, and evidence. Record the exact service and feature because responsibility can change within one provider product family.

Bind provider assurance to consumption facts. For each relied-upon service, match the current AOC or report, in-scope service listing, responsibility matrix, cloud or government environment, region where relevant, assessment period, service feature, customer configuration, and contractual entity. Provider validation is an assurance input. It does not automatically validate the customer's architecture or implementation.

Create evidence contracts, not screenshot parity. The AWS console, Azure portal, and Google Cloud console need not look alike. Require equivalent output fields: subject ID, authoritative container, collection method, collector identity and permissions, query or API, collection time, represented period, inclusion rules, exclusions, raw-object reference, transformation, exceptions, owner, and claim relationship. This makes heterogeneous evidence comparable without pretending it is interchangeable.

Translate tests through propositions and populations. A vulnerability scan, ASV scan, penetration test, access review, ruleset review, key review, log review, or software assessment should derive its targets from the current architecture and provider-specific resources. A generic row labeled cloud does not identify public endpoints, managed gateways, serverless revisions, container images and runtime policies, cluster control planes, workload identities, provider constraints, or alternate administrative paths.

Use provider names at the evidence edge and stable identifiers at the assessment core. Resource ARNs, Azure resource IDs, Google Cloud full resource names, provider event IDs, and assurance-package references should remain intact. Join them to stable internal subject, relationship, claim, obligation, test, and conclusion IDs. That design lets the provider implementation change without forcing the assessment language to forget what question it was answering.

Do not make the cross-cloud matrix decide compliance. It can expose missing populations, contradictory configurations, stale assurance, absent logs, unowned identities, or incomplete tests. The assessed entity

still owns its representations and remediation. The assessor still evaluates applicability, evidence, testing, and conclusions. The compliance-accepting entity still determines how it treats the result.

QUESTIONS

QUESTIONS THAT EXPOSE THE WACK

- 01 What stable claim are we answering before any provider service name appears?
- 02 Which organizations, tenants, accounts, subscriptions, folders, projects, regions, and environments form the complete candidate population?
- 03 Which provider-native resource identifiers join to which internal subject and relationship identifiers?
- 04 What customer-controlled capability remains for each managed service, including code, configuration, identity, network, secrets, logging, deployment, and support?
- 05 Which inventory source is authoritative for each resource type, and what enablement, permission, coverage, time, consistency, and unsupported-type limitations apply?
- 06 Which log category and retained period can show who did what to the relevant resource, and which activity is not captured by that source?
- 07 How do human, workload, emergency, federated, provider-support, and application authorization paths reach the payment environment?
- 08 Does the current provider assurance packet cover the exact service, feature, environment, region, period, and relied-upon activity?
- 09 Which public endpoints, callbacks, gateways, control planes, registries, functions, containers, clusters, and administrative paths belong in scan and penetration-test populations?
- 10 Where do IaC intent, provider control-plane state, runtime observation, diagrams, and inventory disagree?
- 11 What one provider change reopens the equivalent claims, evidence, and tests on the other payment paths?
- 12 Can a cold reader traverse one claim across all three providers without relying on verbal translation from the cloud team?

THE CLUSTER OF WACK

FIND / MAP / EXPLAIN / REDUCE / PROVE

ESCAPE THE WACK

01 / FIND

Choose one payment journey or security-influence path that crosses AWS, Azure, Google Cloud, or a cloud-to-SaaS seam. Recover organization hierarchies, resources, identities, network paths, IaC, runtime state, logs, provider assurance, responsibilities, evidence, tests, exceptions, and current assessment claims.

02 / MAP

Create the provider-neutral claim spine, then attach provider-native objects without renaming them into false equivalents. Join stable subject, relationship, claim, obligation, test, and conclusion IDs to exact provider resource, identity, event, configuration, assurance, and evidence identifiers.

03 / EXPLAIN

For every provider answer, state coverage, collection method, period, permissions, limitations, and non-claims. Distinguish similar function from identical behavior, available telemetry from retained evidence, provider performance from customer responsibility, and provider validation from customer assessment.

04 / REDUCE

Standardize the questions and output contract, not the provider implementation. Reuse governed collectors and references, reconcile source disagreements, remove orphan exports and provider-wide assertions, and preserve the minimum provider-specific facts needed to reproduce each bounded claim.

05 / PROVE

Select one equivalent-looking control across the three providers. Give the records to a cold reviewer and require them to reconstruct the exact population, capability, responsibility, evidence period, test coverage, limitation, and reopening trigger without a cloud-platform interpreter.

THE FIELD LAW

THE LINE WORTH REMEMBERING

TRANSLATE THE ASSESSMENT QUESTION. PRESERVE THE PROVIDER ANSWER. NEVER FLATTEN THE DIFFERENCE THAT CARRIES THE EVIDENCE.

Build one Cross-Cloud Claim Translation Record for a real payment path. Start with the stable claim, map provider-native resources and evidence, and mark every blank as unresolved rather than equivalent.

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-0036 / Information Supplement: PCI SSC Cloud Computing Guidelines v3 / PCI Security Standards Council / [Open source](#)
- WACK-SOURCE-0040 / AWS Services in Scope by Compliance Program: PCI DSS / Amazon Web Services / [Open source](#)
- WACK-SOURCE-0042 / Azure PCI DSS compliance offering / Microsoft / [Open source](#)
- WACK-SOURCE-0043 / PCI DSS compliance in Google Cloud / Google Cloud / [Open source](#)
- WACK-SOURCE-0128 / How AWS Config Works / Amazon Web Services / [Open source](#)

- WACK-SOURCE-0129 / Working with CloudTrail Event History / Amazon Web Services / [Open source](#)
- WACK-SOURCE-0130 / Overview of Azure Resource Graph / Microsoft / [Open source](#)
- WACK-SOURCE-0131 / Azure Monitor Activity Log / Microsoft / [Open source](#)
- WACK-SOURCE-0132 / Cloud Asset Inventory Overview / Google Cloud / [Open source](#)
- WACK-SOURCE-0133 / Cloud Audit Logs Overview / Google Cloud / [Open source](#)

AUTHORITY CLASSIFICATION / PCI_SSC_SOURCE / ASSESSOR_JUDGMENT / PRACTITIONER_POSITION / PRACTITIONER_METHOD / AI_ASSISTED

Practitioner reviewed by Scott Norton on 2026-09-20 against canonical digest

a924cb8ee0bc8aa27b83ececfa97af486516439af465ef34fb95c2e8d678464. 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.