

THE SEVEN PLANES OF PCI DSS

A requirement number is an address in the Standard. It is not a complete model of the environment, the data, the control, the people, the evidence, the test, or the reason the work exists.

PRACTITIONER POSITION

My practitioner position is that a useful PCI DSS analysis can be projected through seven orthogonal planes: architecture and scope; account data; control; governance; evidence; assessment; and risk. Time is not an eighth plane. It is the dimension running through every plane, because the asset, data flow, control state, owner, evidence, test result, and risk conclusion can each be true at one moment and false or incomplete at another. PCI DSS remains organized into twelve principal requirements with sub-requirements, testing procedures, applicability notes, guidance, and reporting instructions. The seven-plane model is not PCI SSC terminology, a substitute reporting structure, or a compliance determination. It is a practitioner method for asking seven different questions of the same environment and then examining the intersections. A flat requirement-to-document link can say where evidence was filed. A plane-aware graph can show what existed, what data did, which mechanism acted, who was accountable, what fact was observed, how the assertion was independently tested, why the assurance mattered, and when each statement was valid.

THE FIELD PATTERN

WHAT THIS LOOKS LIKE IN PRACTICE

An assessor asks for support for one network-security-control requirement. The entity supplies a rule export and a screenshot from its compliance platform. Architecture says the rule belongs to a shared cloud boundary. The data plane says two payment journeys cross it, but a support path does not appear in the diagram. The control plane says the rule is generated by policy-as-code. Governance says Platform Engineering owns the template while the payment team owns allowed services. Evidence says the export came from one account yesterday. Assessment says the population spans six accounts and three material variants. Risk says an unrestricted management path changes the assurance needed. Time says the template changed halfway through the assessment period. None of those facts competes with the requirement. Together they make the requirement assessable.

SIGNALS DETECTED

WHAT SHOULD MAKE YOU STOP

- A requirement row points directly to a document without naming the architecture, data state, control behavior, owner, population, test, or period the document supports.
- Scope, diagrams, inventories, access, vulnerability evidence, provider responsibilities, and test results use different names for the same system component.

- Evidence produced by a tool is treated as the assessor's test rather than an input the assessor must independently evaluate.
- A control owner is named, but the operator, reviewer, approver, change authority, provider, and compliance-accepting entity are indistinguishable.
- Risk appears only in an annual register and cannot be traced to the control objective, implementation choice, frequency, exception, or testing depth it informs.
- A current diagram is used to support a historical period without showing when the architecture, data path, responsibility, or control state changed.
- Provider responsibility is attached to a requirement number but not to the exact service, activity, customer configuration, evidence, or retained duty.
- A dashboard can answer whether a test passed but cannot traverse from the result to the actual population, variants, limitations, and bounded assessment conclusion.

WHY IT MATTERS

THE COST OF LEAVING IT UNEXPLAINED

Plane one is architecture and scope: the components, workloads, services, identities, locations, network and trust relationships, provider seams, connections, and security-impact paths that answer what is being assessed and why.

Plane two is account data: where PAN, cardholder data, sensitive authentication data, tokens, encrypted or masked values, payment messages, and related transaction data originate, move, transform, persist, disappear, or reappear on normal and exception paths.

Plane three is control: the mechanisms that prevent, detect, respond, recover, constrain, authorize, record, or verify-from network security controls and IAM to encryption, secure development, vulnerability management, logging, change control, and incident response.

Plane four is governance: who owns, operates, reviews, approves, attests, accepts, funds, changes, and maintains each proposition. Provider execution does not erase entity governance; it changes the responsibility relationship that must be evidenced.

Plane five is evidence: attributable observations emitted by the preceding planes. Configuration, API response, log, ticket, scan, report, inventory, approval, and attestation each need subject, source, time, collection method, transformation, custody, and claim lineage.

Plane six is assessment: the independent examination, observation, interview, population analysis, selection, technical validation, contradiction handling, limitation, and conclusion that tests an entity assertion. Evidence and assessment are related, but they are not synonyms.

Plane seven is risk: the threat, exposure, likelihood and impact factors, control objective, residual condition, frequency rationale, exception pressure, and decision authority that explain why assurance is needed without silently redefining applicability or compliance.

Run time through all seven planes. Record valid-from and valid-to state, expected occurrences, actual events, changes, exceptions, evidence collection, testing dates, and reopening triggers. A current-state

snapshot cannot silently prove an earlier interval.

Study intersections instead of viewing each plane alone. Architecture × data informs scope; architecture × control describes implementation; control × governance exposes accountability; control × evidence produces observability; evidence × assessment produces assurance; provider × activity × requirement allocates responsibility; and risk × control × time informs a bounded operating frequency.

Treat a PCI DSS sub-requirement as a coordinate across the graph, not the center of the universe. Preserve its normative text and testing procedure, then connect the relevant subjects, data states, mechanisms, people, evidence, tests, risk rationale, and time interval.

QUESTIONS

QUESTIONS THAT EXPOSE THE WACK

- 01 Can one requirement claim be projected through all seven planes without changing the subject, population, period, or implementation story?
- 02 Which plane is currently carrying facts that belong somewhere else—for example, a tool result pretending to be an assessor conclusion?
- 03 Where do stable identifiers break between architecture, data flows, controls, responsibilities, evidence, tests, and risk records?
- 04 Which plane has the weakest history through time, and what conclusions are currently borrowing a current snapshot to fill that gap?
- 05 At which intersections do provider services change the data path, control operation, evidence source, or retained customer responsibility?
- 06 Can the assessment plane identify the full population and material variants represented by each evidence artifact?
- 07 Which risk decisions inform frequency or implementation, and which have been allowed to masquerade as scope or applicability decisions?
- 08 If one architecture component changes tomorrow, which plane records and bounded claims automatically reopen?

THE CLUSTER OF WACK

FIND / MAP / EXPLAIN / REDUCE / PROVE

ESCAPE THE WACK

01 / FIND

Select one consequential requirement claim and collect its named subjects, data states, controls, people, evidence, tests, risk rationale, and time boundaries without assuming the existing links are correct.

02 / MAP

Place each fact on its proper plane, assign durable identifiers, and draw the intersections and time-valid relationships that make the complete proposition traversable.

03 / EXPLAIN

Label every statement by authority and role: PCI SSC source, entity assertion, provider fact, organizational position, assessor judgment, practitioner method, evidence observation, or compliance-accepting-entity direction.

04 / REDUCE

Remove duplicate names, orphan evidence, control-to-document shortcuts, provider hand-waves, stale snapshots, and plane-crossing conclusions that cannot preserve their subject, population, period, and authority.

05 / PROVE

Walk one requirement coordinate through all seven planes and two points in time. Reconcile every contradiction and state the exact conclusion that the resulting chain can-and cannot-support.

THE FIELD LAW

THE LINE WORTH REMEMBERING

A REQUIREMENT NUMBER TELLS YOU WHERE TO LOOK. THE PLANES TELL YOU WHAT MUST AGREE WHEN YOU GET THERE.

Download the PCI DSS Plane Intersection Matrix. Choose one requirement claim, map it through all seven planes and two points in time, then mark every missing or contradictory intersection W/?.

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-0034 / Best Practices for Maintaining PCI DSS Compliance / PCI Security Standards Council / [Open source](#)

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

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

59a82876525abcc6d2ea88d83995c2c5a7277d99fafa1d5bf9c38ad640d8752f. 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.