

DOES YOUR ISV UNDERSTAND PCI DSS COMPLIANCE?

Your ISV can build an elegant payment integration and still hand you a sentence that is impossible to assess: We are PCI certified. Certified as what, for which product or service, under which standard, covering which version and responsibilities?

PRACTITIONER POSITION

My practitioner position is that an ISV's PCI maturity is revealed less by the badge it offers and more by the precision of the boundary it can explain. ISV is a commercial relationship, not a PCI classification. Depending on actual activities, a software vendor may provide payment software, host or manage services, act as a third-party service provider, influence the security of the CDE, or remain outside particular account-data and security-impacting paths. Different PCI programs and evidence can apply to different objects. A Secure Software Framework listing, where applicable, evaluates a defined payment software product and does not automatically make the deploying merchant compliant. A PCI DSS AOC can describe an assessed service provider and still leave customer-controlled configuration, integration, access, operations, and validation obligations. The assessed entity needs exact claims: who was assessed, against what, for which service or product and version, through what deployment model, for what period, with which dependencies, exclusions, and responsibilities. If the ISV cannot answer that without changing nouns three times, the integration is not ready to inherit a scope conclusion.

THE FIELD PATTERN

WHAT THIS LOOKS LIKE IN PRACTICE

A merchant buys a commerce platform from an ISV. Sales says the platform is PCI compliant. Engineering says the card reader encrypts data and the hosted page is handled by a processor. Support says the merchant owns workstation security and user access. The contract names none of those boundaries. The implementation guide mixes hosted and merchant-hosted scripts, the version in production differs from the version on a validation listing, and the provider AOC covers a managed service the merchant did not purchase. During assessment, everyone presents a different true fragment. The problem is not necessarily an insecure product. It is the absence of one attributable, versioned claim that joins product, service, deployment, data flow, responsibility, and evidence.

SIGNALS DETECTED

WHAT SHOULD MAKE YOU STOP

- The answer to Which PCI standard? is PCI compliant, PCI certified, Level 1, or our processor handles it.

- Product validation, service-provider assessment, merchant validation, and payment-brand or compliance-accepting-entity direction are treated as interchangeable.
- The ISV cannot name the exact product version, service SKU, legal entity, deployment model, region, functions, date, and evidence covered by its claim.
- The implementation guide does not distinguish defaults from required secure configuration, customer choices, prohibited combinations, prerequisites, and change-sensitive assumptions.
- The ISV supplies an AOC or listing but no responsibility allocation connecting every relevant activity to provider, customer, processor, integrator, or other party.
- Production uses custom scripts, plugins, reader firmware, APIs, containers, functions, identities, or support paths that do not appear in the vendor's described architecture.
- Patch, vulnerability, end-of-life, logging, incident, remote access, key, credential, and change responsibilities surface only after the assessor asks.
- The ISV promises scope reduction without stating the factual eligibility criteria, customer obligations, exceptions, or evidence needed to sustain it.

WHY IT MATTERS

THE COST OF LEAVING IT UNEXPLAINED

Translate every claim into a record. Capture claimant, legal entity, standard or program, object assessed, product or service name, version, deployment model, period, functions, regions, dependencies, exclusions, evidence identifier, source, and expiration or change trigger. Reject orphan adjectives.

Classify roles from activities. Determine whether the ISV develops payment software, hosts or manages a service, stores, processes, or transmits account data, manages security-impacting components, provides remote support, supplies only software, or performs multiple roles. Preserve fact-specific classifications instead of forcing the company into one label.

Separate product assurance from entity compliance. Use applicable Secure Software Framework evidence to understand a defined software product. Use PCI DSS service-provider evidence to understand assessed services. Use the merchant's own scope, implementation, responsibilities, and validation instructions to determine what the merchant must prove.

Demand a responsibility model that reaches configuration. Map hosting, deployment, identities, access, logging, vulnerability management, patching, code and script integrity, segmentation, keys and tokens, incident response, support, change detection, data retention, secure deletion, and evidence production. Shared does not mean somebody else.

Reconcile evidence to production. Match legal names, service names, versions, regions, deployment patterns, dependencies, dates, and functions. Record every mismatch and either resolve it with authoritative evidence or keep the affected conclusion open.

Test the integration seams. Verify redirects, embedded frames, scripts, APIs, readers, firmware modes, callbacks, webhooks, tokens, logs, error paths, administrative portals, remote support, updates, and custom code. Scope reduction claims often fail at the seam rather than inside the vendor's assessed service.

Make lifecycle duties explicit. Identify security advisories, release support, patch timelines, end-of-life dates, customer notification, emergency changes, incident coordination, evidence refresh, and who re-evaluates the integration when the product or service changes.

Give procurement an acceptance gate. Do not accept a payment technology on the strength of a logo. Require the claim record, current evidence, responsibility matrix, deployment and data-flow model, secure implementation guidance, lifecycle commitments, incident path, and named owners before production approval.

QUESTIONS

QUESTIONS THAT EXPOSE THE WACK

- 01 When the ISV says PCI compliant, which exact entity, product or service, version, deployment, standard or program, period, and functions does it mean?
- 02 What activities make the ISV a software vendor, TPSP, service provider, integrator, or another role for this implementation?
- 03 Does the supplied evidence cover the exact service and deployment we purchased, or merely a related product, parent company, environment, or offering?
- 04 Which customer configurations and responsibilities are necessary for any validation eligibility or scope-reduction claim to remain true?
- 05 Which custom scripts, APIs, readers, firmware, plugins, identities, containers, functions, support paths, and logs sit outside the vendor's described boundary?
- 06 Who owns vulnerability response, patching, end of life, logging, access, remote support, incident coordination, change notification, and evidence refresh?
- 07 How does the vendor prove that production versions, dependencies, regions, data flows, and responsibility statements remain aligned after change?
- 08 What would the ISV have to tell us tomorrow if its evidence, architecture, software, service, subcontractor, or security status changed?

THE CLUSTER OF WACK

FIND / MAP / EXPLAIN / REDUCE / PROVE

ESCAPE THE WACK

01 / FIND

Collect every PCI claim, listing, AOC, responsibility statement, product and service identifier, version, implementation guide, architecture, data flow, contract term, support path, and customer-controlled configuration associated with the ISV.

02 / MAP

Build an ISV PCI Claim and Responsibility Record that connects each claim to the exact object, standard, evidence, production implementation, provider activity, customer duty, integration seam, and lifecycle trigger.

03 / EXPLAIN

Write one bounded conclusion for product assurance, one for assessed provider services, and one for the assessed entity's deployment and validation. State where they support one another and where they do not.

04 / REDUCE

Remove unneeded customization, data paths, plugins, privileges, support access, retained data, unsupported versions, ambiguous shared duties, and undocumented deployment variants. Prefer architectures the ISV can actually explain and support.

05 / PROVE

Select one production payment journey. Reconcile every component and responsibility to current evidence, test each integration seam and failure path, and show how a vendor or customer change reopens the affected claims.

THE FIELD LAW

THE LINE WORTH REMEMBERING

ISV IS A RELATIONSHIP. PCI SCOPE AND RESPONSIBILITY FOLLOW ACTIVITIES, SYSTEMS, SERVICES, AND EVIDENCE.

Download the ISV PCI Claim and Responsibility Record. Translate the phrase PCI compliant into a claimant, exact object, standard, version, period, evidence artifact, deployed boundary, retained responsibility, and change trigger-or stop pretending it is a conclusion.

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-0002 / How is an entity's PCI DSS compliance impacted by using TPSPs? / PCI Security Standards Council / [Open source](#)

- WACK-SOURCE-0023 / FAQ 1065: What is a TPSP expected to provide to customers as evidence of PCI DSS compliance? / PCI Security Standards Council / [Open source](#)

- WACK-SOURCE-0073 / PCI Secure Software Standard / 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

9986b7dff8b521a126e9e2bdb2c05e355919484347edb9ae70cbc87a656b81e9. 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.