The Security & Compliance Pack

Answering the questions procurement will ask

Charging platforms hold money, driver identities and remote control of physical infrastructure. Here is how to evidence that you handle all three properly.

FOR Operators facing enterprise or public-sector procurement

UPDATED 16 August 2026

Contents

01 **The threat model**

Unattended devices in public places, money in motion, and remote control of physical equipment. What each implies.

02 **Isolation and access**

Tenant scoping in the data layer, role-based access, territory restrictions and separately permissioned financial actions.

03 **Payments and scope**

Why card data should never touch the platform, and what that removes from your compliance burden.

04 **Audit and evidence**

What must be recorded, for how long, and what an auditor will actually ask to see.

05 **Residency and retention**

Where operational data sits, how long it is kept, and how to fix both contractually.

06 **Answering the questionnaire**

Model responses to the questions that recur — including how to answer honestly when a control is not yet in place.

WHAT YOU GET

- The security questions that come up in every enterprise evaluation
- What tenant isolation means in practice, and how to demonstrate it
- Payment scope: what you hold, and what you should never hold
- Data residency, retention and portability as contract terms

CHAPTER 01

The threat model

Charging platforms are usually assessed with a generic software security questionnaire, which misses what is distinctive about them. Three properties make charging different, and each implies a category of risk that a standard checklist does not ask about.

Unattended devices in public places

A charger is a networked computer bolted to the ground in a car park, reachable by anyone, powered continuously, and rarely inspected. It is the least defensible endpoint most organisations own.

- Physical access to the enclosure means access to whatever is inside it, including any local credentials.
- A charger connects outward to your platform, so a compromised unit is a foothold that is already inside the perimeter.
- Firmware is updated rarely and by third parties, and vulnerabilities persist for years in fielded units.
- The unit identifies itself to the platform, and in many deployments that identity is the only authentication.

THE IMPLICATION PEOPLE MISS

Charge points are not trusted devices and must not be treated as ones. Everything a charger reports is input to be validated, not fact. A platform that acts on charger-supplied data without schema validation is trusting equipment anyone can open with a screwdriver.

Money in motion

Sessions create charges, wallets hold balances, refunds send money out, and settlements pay third parties. Each is a path by which value leaves the system, and each needs a separate control.

PATH	RISK	CONTROL
Refunds	Unlimited outbound value	Separate permission, cap, audit
Wallet credits	Value created from nothing	Separate permission, reconciliation
Settlement approval	Payments to a third party	Separate permission, dual approval at size
Tariff changes	Revenue silently reduced	Effective dates, audit trail

Remote control of physical equipment

The property that has no software analogue. An API call can energise or de-energise equipment attached to a vehicle in a public place. That deserves a different standard of care from an API call that updates a record.

01 Command permissions separated from data permissions, so a reporting integration cannot reset a charger.

02 Every command recorded against the person or key that issued it.

03 Safety-relevant actions — clearing a ground fault, overriding a load limit — behind an additional confirmation that names the risk.

04 Bulk commands treated as more dangerous than single ones, because a mistake scales.

What follows

The rest of this document works through the four areas these imply: isolation between operators, what happens to payment data, what must be auditable, and where data lives. Each is framed as what to ask and what a good answer sounds like, because that is the form the questions will reach you in.

CHAPTER 02

Isolation and access

A platform serving several operators holds several businesses' money, drivers and infrastructure in one system. Isolation between them is the precondition for the product existing, and it fails in a consistent, boring way: somebody writes a query and forgets the filter.

Enforce it once, not everywhere

The pattern that works is a single layer through which all data access passes, applying the tenant scope from request context. Where that exists, a forgotten filter is impossible. Where it does not, correctness depends on every query ever written — including the one added under time pressure on a Friday.

- Every tenant-scoped table carries a tenant identifier, indexed and non-null.
- The scope comes from the authenticated context, not from a parameter the caller supplies.
- Platform-level tables that legitimately span tenants are a short, explicitly enumerated list on a separate code path.
- Any query bypassing the layer is a review finding rather than a shortcut.

The four places it actually leaks

LEAK	WHY IT HAPPENS
Aggregate and reporting queries	Written as raw SQL for performance
Background jobs	No request context, so scope is threaded manually
Lookup by opaque identifier	A UUID feels unguessable, so the check feels redundant
Caches	A key without the tenant serves one operator another's data

UNGUESSABLE IS NOT A CONTROL

An object fetched by identifier must still be checked against the caller's tenant, however random the identifier looks. This is the most common real isolation defect and it is invisible until somebody enumerates.

Inside one operator

The second boundary matters almost as much, and it is what enterprise procurement asks about.

01 Staff restricted to the stations or territories they are responsible for, enforced in the data layer rather than hidden in the interface.

02 View, operate and manage as separable permissions, so a contractor can reset a charger without reconfiguring it.

03 Site partners seeing only their own sites and revenue.

04 Financial actions — refunds, wallet credits, settlement approval, tariff changes — permissioned independently of day-to-day operations.

The last is not paranoia about employees. A support agent with unlimited refund capability is a control failure an auditor will find, and the finding is against you rather than against them.

Proving it

Claims are cheap. What means something is a test suite asserting that cross-tenant access returns nothing, an audit trail showing who read and changed what, and a willingness to demonstrate both. Ask any platform for them — including this one.

CHAPTER 03

Payments and scope

The most valuable security decision in a charging platform is a negative one: never handle card data. Everything that follows from that is a reduction in what you have to defend, prove and answer for.

How it should work

The driver enters card details into the payment gateway's own hosted interface or SDK. The gateway returns a token — a reference that is useless to anyone else — and the platform stores the token. The platform never sees, transmits or stores a card number.

THE QUESTION TO ASK, AND THE ANSWER TO EXPECT

"Does any card data touch your systems?" The answer should be an immediate no, with an explanation of the tokenisation flow. Hesitation, or an answer about how securely it is stored, is itself the finding.

What it removes

- The bulk of PCI DSS scope, which is expensive and recurring.
- The most damaging single breach scenario available to a charging platform.
- A category of encryption, key management and retention questions that no longer apply.
- Liability that would otherwise sit with you rather than with a payment institution.

What still needs care

Removing card data does not remove everything. Four things remain and are worth being explicit about.

01 Payment tokens. Not card numbers, but they can initiate charges against a saved instrument and must be protected accordingly.

02 Wallet balances. Real money held on behalf of drivers, which is a liability with its own reconciliation and audit expectations.

03 Refund capability, which is an outbound value path regardless of who holds the card data.

04 Gateway credentials. API keys that can move money, and which belong in environment configuration rather than in the database or the repository.

Whose merchant account

A structural question with security consequences. Where driver payments land in the operator’s own merchant account, the platform never holds the operator’s revenue and the operator carries no counterparty exposure to the platform.

Where they land in the platform’s account and are remitted, the platform is holding your money and your drivers’ balances. That may be entirely fine, and it should be a decision rather than a discovery at contract signature.

What to put in a questionnaire response

State plainly: card data is never received, transmitted or stored; payments are processed by named gateways; the platform stores tokens only; wallet balances are held as ledger entries and reconciled against gateway settlements. Four sentences, and they answer most of a payments section.

CHAPTER 04

Audit and evidence

An audit trail is not a log. Logs are for engineers diagnosing behaviour; an audit trail is a durable record of who did what, retained deliberately, that a third party can read. Conflating them is why so many platforms cannot answer an auditor's question.

What must be recorded

CATEGORY	EXAMPLES
Money out	Refunds, wallet credits, settlement approvals
Commercial terms	Tariff changes, contract terms, group membership
Physical control	Remote start, stop, reset, availability, profiles
Access	Logins, permission changes, API key issue and revocation
Data	Exports, bulk reads, deletions
Configuration	Charger configuration changes, firmware pushes

Each entry needs the actor, the action, the target, the time, and — for anything with a before and after — both values. An audit entry saying a tariff was changed, without saying from what to what, answers nothing.

Properties that make it an audit trail

01 Append-only. If it can be edited, it is not evidence.

02 Retained on a stated schedule, longer than your dispute and tax obligations.

03 Readable by someone who is not an engineer, without a query language.

04 Complete for the categories above, rather than for whatever happened to be instrumented.

05 Covering platform staff as well as operator staff, where a vendor can access production data.

THE LAST ONE IS THE ONE TO ASK ABOUT

Who at the vendor can access your production data, under what process, and is that access logged where you can see it? Most questionnaires ask about your own users and never about theirs.

What an auditor actually asks for

Rarely a general review. Usually something specific, and the platform either produces it in minutes or does not.

- Every refund over a threshold in a period, with who authorised each.
- All changes to a particular tariff, with dates and actors.
- Who accessed a named driver’s records.
- The complete history of one session, including any adjustment.
- Permission changes for a named user over a year.

Retention

Session records and their meter values are the evidence in a billing dispute, and disputes arrive late. Audit entries are the evidence in a control review, which arrives later still. Set retention against the longest obligation you actually have rather than against storage cost, which for text records is not the binding constraint.

CHAPTER 05

Residency and retention

Where data lives has moved from an infrastructure detail to a contractual term, particularly for Indian operators selling into enterprise and public sector. It is worth having a precise answer rather than a reassuring one.

What to be able to state

01 The region where operational data — sessions, drivers, meter values — is stored.

02 Whether backups are held in the same region.

03 Whether any processing happens elsewhere, including third-party services for email, analytics or error reporting.

04 Whether support staff outside the region can access production data.

05 What would be involved in changing any of the above, if a customer required it.

THE THIRD AND FOURTH ARE WHERE HONEST ANSWERS GET COMPLICATED

A platform hosted in India may still send error reports to a service abroad, or have an engineer in another country with production access. Both are usually acceptable if disclosed and controlled. Neither survives being discovered during a customer's audit.

Retention

Different data has different obligations, and a single retention period applied to everything is either too short for the records you need or too long for the personal data you hold.

DATA	DRIVEN BY
Session records and amounts	Tax and dispute obligations
Meter values	Dispute defence — the evidence behind an invoice
Driver personal data	Purpose limitation; keep while the account is active
Audit entries	Control review obligations, usually the longest
Operational logs	Engineering need, and should be the shortest
Payment tokens	Only while the driver wants the instrument saved

Deletion requests

A driver asking for their data to be deleted creates a tension: personal data should go, and the financial record of a transaction must remain. The workable answer is anonymisation — remove the identifying fields, keep the session with its energy and amount, so the accounts still balance and the person is no longer identifiable.

Have that process defined before it is requested, because the first request is a poor time to design it.

Putting it in the contract

01 The region, stated, with notice required before it changes.

02 Retention periods per data category, or a right to set them.

03 A stated period to retrieve data on termination, and the account remaining functional during it.

04 Formats and completeness of export, including whether meter values are included.

05 Sub-processors listed, with notice before additions.

CHAPTER 06

Answering the questionnaire

Enterprise and public-sector procurement arrives as a spreadsheet of a hundred questions, most of them generic. The recurring ones are answerable once and reused, and the value of doing that is not efficiency — it is that a prepared answer is a precise one.

The recurring questions

QUESTION	THE SHAPE OF A GOOD ANSWER
Do you store card data?	No. Tokenisation, named gateways, no card data received or stored
How is data isolated between customers?	The mechanism, plus how it is tested
Is data encrypted in transit and at rest?	Yes, with the specifics
Who at your company can access our data?	Named roles, the process, and that it is logged
Is MFA available and enforceable?	Yes, and who can enforce it
Where is data stored?	The region, and whether backups share it
What is your incident notification commitment?	A number of hours, and where it is written
Do you have a disclosure process?	The address, and the response commitment
Can we export all our data?	Yes, self-service, and what it contains
Which certifications do you hold?	Held, with dates. In progress, labelled as such

Answering honestly when the answer is no

Every platform has controls that are not in place yet. The question is how that is said, and the difference between a finding and a non-issue is entirely in the phrasing.

THE PATTERN THAT WORKS

State what is not in place. State what compensates for it today. State whether it is planned, and if so when. Three sentences. What does not work is a yes that becomes a no under examination, because that converts a gap into a credibility problem — and credibility problems end evaluations that gaps do not.

A worked example: "We do not hold ISO 27001 today. Our controls follow it and were reviewed by an external assessor in the last year, whose summary we can share. Certification is planned but I will not give you a date I cannot commit to."

What to prepare in advance

01 A one-page security overview covering the ten questions above.

02 A sub-processor list, current.

03 A data flow description: what is collected, where it goes, who touches it.

04 Your incident response process, in a form someone outside the team can read.

05 Whatever third-party assessment you have, with the summary separable from the detail.

The questions to ask your own platform

Everything above will be asked of you, and most of the answers depend on your charging platform rather than on you. Put the same questions to them, in writing, before you have to answer for them. A vendor who cannot answer precisely is a vendor whose answers you will be repeating to your own customers.

About this document

Written by the team building zevOS, the charge point management platform. We publish this kind of thing because operators who understand utilisation, demand charges and settlement make better decisions — including about us.

Corrections and disagreements are genuinely welcome. If something here is wrong, or true in most states but not yours, write to support@zevos.ai and we will fix it in the next revision.

White Stripe Innovations Private Limited. Last updated 16 August 2026. This document is general information about running a charging business, not legal, tax or financial advice.