Learn EV Charging

ISO 15118 · PLUG & CHARGE

The ISO 15118 & Plug & Charge Field Guide

The car-to-charger conversation: the session flow, how certificate-based Plug & Charge really works, and where the PKI bites.

A practitioner's reference to the protocol that lets you plug in and just charge — no app, no card.

learnevcharging.com

FIELD GUIDE

What's inside

ISO 15118 is the layer most people wave at and few can walk through. This guide is the session, the certificates, and the traps — for the engineer wiring up Plug & Charge.

HOW TO USE THIS

Message names follow ISO 15118-2 (the most-deployed version); where 15118-20 differs, it's flagged. This is a practitioner's map, not the normative standard — the ISO documents remain the source of truth.

01 · THE MENTAL MODEL

EVCC ↔ SECC, and where it sits

ISO 15118 is the digital conversation between the **car** and the **charger** — formally, "Road vehicles — Vehicle-to-Grid Communication Interface." Two controllers do the talking:

EVCC

EV Communication Controller — the car's side. Asserts what the vehicle wants, and (for Plug & Charge) presents its contract certificate.

SECC

Supply Equipment Communication Controller — the charger's side. Offers services, validates certificates, and controls power delivery.

Where it sits in the stack

```
Car  ──ISO 15118 (EXI over PLC)──▶  Charger
                                     Charger ──OCPP──▶ CSMS
                                                      CSMS ──OCPI──▶ eMSP
```

ISO 15118 is **car ↔ charger only**. It does not talk to the back office — that's OCPP's job. A full Plug & Charge authorization actually rides all three protocols (page 06).

> **TWO THINGS IT UNLOCKS**
>
> **Plug & Charge (PnC)** — plug in and charging authorizes automatically via a certificate, no app or RFID. And **smart / bidirectional charging** — the car and charger negotiate schedules and (in -20) send power back to the grid.

On the wire it's **EXI** (Efficient XML Interchange) — a compact binary encoding of XML — carried over **PLC** (power-line communication) on the Control Pilot line, secured with TLS.

02 · THE SESSION

From plug-in to power

A DC session runs roughly this sequence. AC skips the DC-only steps (CableCheck, PreCharge, WeldingDetection).

1. **SLAC → SDP**
   Signal-Level Attenuation Characterization pairs the car and charger over power-line comms; SECC Discovery Protocol finds the SECC and opens a TCP/TLS connection.

2. **supportedAppProtocol**
   The two agree on the ISO 15118 version and schema to speak.

3. **SessionSetup → ServiceDiscovery**
   A session ID is issued; the charger advertises its services (charging, plus value-adds).

4. **PaymentServiceSelection**
   The car picks how it will authorize: `Contract` (Plug & Charge) or `ExternalPayment` (EIM — app/RFID handled outside ISO 15118).

5. **PaymentDetails → Authorization**
   For PnC, the car presents its contract certificate and answers a signed challenge (page 03).

6. **ChargeParameterDiscovery → CableCheck → PreCharge**
   Limits and schedules are exchanged; the charger checks insulation and ramps its output to match the battery voltage before contactors close.

7. **PowerDelivery → CurrentDemand loop → SessionStop**
   Power flows; the car requests current in a loop (ChargingStatus for AC); then stop, WeldingDetection (DC), and session close.

> KEY
>
> **PreCharge** is the safety heart of DC: the charger matches its voltage to the pack *before* the contactors close, so nothing arcs. Skipping or mistiming it is how you damage hardware.

Certificate authorization, step by step

Plug & Charge replaces "tap a card / open an app" with a certificate the car already holds. No user action — the trust is cryptographic.

1. **The car holds a contract certificate**
   Tied to the driver's eMSP contract and identified by an **eMAID** (e-Mobility Account Identifier). It was provisioned into the car (page 04).

2. **PaymentDetails — car sends the cert chain**
   The EVCC sends its contract certificate (and sub-CA chain) to the SECC, plus a fresh challenge value the charger generates.

3. **Authorization — proof of possession**
   The car signs the challenge with its contract private key. The charger verifies the signature *and* the certificate chain up to a trusted **V2G Root**.

4. **The back office confirms**
   The charger relays the identity to its CSMS over OCPP, which checks with the eMSP that the contract is valid and authorized to charge here.

> **GOTCHA — PNC ≠ AUTOCHARGE**
> **Autocharge** is a proprietary shortcut that authorizes on the car's MAC address — convenient, but not secure and not standardized. **Plug & Charge** is the real ISO 15118 method: certificate-based, signed, and interoperable.

Everything above is EXI-encoded and runs over TLS. Without TLS and a valid SECC certificate, there is no Plug & Charge.

04 · THE HARD PART

The Plug & Charge PKI

Most PnC failures are certificate failures. The trust model has several actors, all chaining to a common root.

Certificate	Held by · purpose
V2G Root CA	The trust anchor everything chains to. Cars and chargers ship trusting it.
SECC certificate	The charger's TLS identity (via a CPO sub-CA). Proves the station is genuine.
Contract certificate	In the car — represents the driver's eMSP contract (eMAID). Signed by the Mobility Operator.
OEM provisioning cert	Factory-installed in the car. Bootstraps the *first* contract certificate.
Provisioning / MO certs	Issue and sign contract certificates; delivered via a Certificate Provisioning Service.

Provisioning, in one line

The car proves itself with the **OEM provisioning certificate** → a **contract certificate** is installed (over the air or at first PnC session via the provisioning service) → from then on the car authorizes with that contract cert until it expires and is renewed.

> **GOTCHA**
>
> Contract certificates **expire**. Renewal/update happens through the provisioning service and is relayed via OCPP's ISO 15118 certificate messages. A stale or unrenewed contract cert is a very common "PnC just stopped working" cause.

05 · VERSIONS

-2 vs -20

Two generations are in play. Most vehicles and chargers today speak **-2**; **-20** is rolling out for bidirectional and heavy-duty.

	ISO 15118-2 (2014)	ISO 15118-20 (2022)
Power flow	Unidirectional (charge only)	Adds **bidirectional (BPT / V2G)** — send power back
Security	TLS 1.2	**TLS 1.3**, stronger crypto
Auth	EIM + Plug & Charge	Improved Plug & Charge, multiple contract certs
New modes	AC & DC	Wireless (WPT), automatic connection (ACD / pantograph), MCS foundations
Scheduling	Basic	Scheduled & dynamic control modes

> **GOTCHA**
> -2 and -20 use different schemas, message sets, and TLS versions. A -2 stack does *not* automatically speak -20 — the `supportedAppProtocol` handshake is what negotiates which one a given session uses.

Bidirectional charging (V2G / V2H) is the headline reason -20 matters: it's the standardized path to a car powering a home or feeding the grid.

06 · THE FULL PICTURE

One tap-free charge, three protocols

A single Plug & Charge session that also roams touches all three protocols. Here's who does what.

THE END-TO-END PATH

```
1  ISO 15118  car → charger: "here's my contract certificate (eMAID), signed"
2  OCPP       charger → CSMS: Authorize + Get15118EVCertificate / cert status
3  OCPI       CSMS → eMSP: is this token/contract authorized to charge here?
4  OCPI       eMSP → CSMS: authorized ✓
5  OCPP       CSMS → charger: start
6  ISO 15118  charger → car: PowerDelivery — energy flows
7  OCPP       charger → CSMS: TransactionEvent (kWh)
8  OCPI       CSMS → eMSP: Session, then the CDR to settle
```

> **KEY**
>
> ISO 15118 proves *who the car is*; OCPP relays that to the back office and carries the certificate housekeeping; OCPI checks authorization and settles the money across networks. Plug & Charge is the one feature that genuinely needs all three working together.

This is exactly why a real CPMS has to speak all three — and why building one is the fastest way to actually understand the stack.

Top 10 gotchas

#	The trap	What's actually true
1	Confusing PnC with Autocharge	Autocharge = MAC-address hack; PnC = signed certificates.
2	Reading EXI as plain XML	EXI is binary — you need the codec + schema to decode the wire.
3	Underestimating the PKI	Most PnC bugs are certificate-chain or provisioning bugs.
4	Assuming -2 speaks -20	Different schemas/TLS; `supportedAppProtocol` negotiates it.
5	Thinking ISO 15118 hits the CSMS	It's car↔charger only; OCPP carries it onward.
6	Skipping TLS for PnC	No TLS + valid SECC cert → no Plug & Charge, full stop.
7	Ignoring contract-cert expiry	Certs expire; renewal rides the provisioning service + OCPP.
8	Mistiming PreCharge (DC)	Match voltage before contactors close, or you arc/damage.
9	Treating AC and DC flows as identical	CableCheck / PreCharge / WeldingDetection are DC-only.
10	Blaming the car for SLAC/PLC issues	Physical-layer pairing problems are a common field failure.

THE PATTERN

ISO 15118 fails in two places: the *physical layer* (SLAC/PLC) and the *certificates* (the PKI). Master those two and most of Plug & Charge falls into place.

One-page cheat sheet

Session messages (-2)

```
supportedAppProtocol
SessionSetup
ServiceDiscovery / ServiceDetail
PaymentServiceSelection
PaymentDetails → Authorization
ChargeParameterDiscovery
CableCheck → PreCharge      (DC)
PowerDelivery
CurrentDemand / ChargingStatus
WeldingDetection (DC) → SessionStop
```

Two authorization modes

PnC	Contract certificate
EIM	External (RFID / app)

The certificates

V2G Root	trust anchor
SECC	charger TLS ID
Contract	eMSP contract (eMAID)
OEM prov.	bootstraps 1st contract

Acronyms

```
EVCC  car controller
SECC  charger controller
EXI   binary XML on the wire
SLAC  PLC pairing
eMAID contract identifier
BPT   bidirectional power (–20)
EIM   external identification
```

Stack: car —ISO 15118→ charger —OCPP→ CSMS —OCPI→ eMSP.

Learn EV Charging

KEEP GOING

You've got the map. Now go deeper.

Every part of this guide has a full walkthrough on the site (tap to read):

- OCPI vs OCPP vs ISO 15118 — how the three layers fit together
- What is OCPP — the layer that carries Plug & Charge to the back office
- More free guides — Plug & Charge, the PKI, V2G, and the rest of the stack

Free, vendor-neutral guides and tools for the people building EV charging — engineers, PMs, and operators.

learnevcharging.com/blog