VEHICLE CHARGING COMMUNICATION

KopherV2G Engineering Guide

Architecture, interfaces
and integration evidence

KB-V2G-EG-001

Revision 1.1 · 11 September 2026

English edition

For OEM and Tier 1 engineering teams

Contents

KopherV2G Engineering Guide

Document information

Field	Value
Document ID	KB-V2G-EG-001
Revision / issue date	1.1 / 11 September 2026
Language / audience	English / OEM and Tier 1 system, ECU and integration engineers
Document type	Product architecture and integration guide
Product scope	Vehicle-side EVCC charging communication software

Purpose and use

This guide explains the KopherV2G software boundaries, deployment choices, communication sequence and integration evidence needed to plan a customer project. It turns the product presentation into an engineering reference for architecture reviews and integration workshops.

Read Chapters 1-2 to select the deployment model, Chapters 3-5 to identify network, security and vehicle interfaces, and Chapters 6-7 to define the integration package and acceptance evidence. Chapter 8 records terminology and technical references.

Scope of this edition

The guide describes product-level integration contracts. Concrete API signatures, runnable configuration, supported feature combinations and release-specific limits are supplied with the selected software package and project specification. The message examples illustrate protocol paths; the applicable standards define the normative behavior.

Revision	Change
1.0	Initial engineering edition based on the September 2026 KopherV2G presentation. Clarifies SDP, TLS, payload types and AUTOSAR configuration boundaries.
1.1	Adds six engineering figures for deployment, security, interface responsibilities, workflow, hardware and acceptance evidence.

KopherBit | kopherbit.com | Product documentation

1 Product scope and deployment

KopherV2G is a vehicle-side EVCC communication stack for integration with the vehicle charging application. The core covers session control, message parsing, the V2G data model, EXI encoding and decoding, SDP and V2GTP. The target platform supplies networking, timing, storage and security services.

1.1 Protocol scope

Protocol family	Project definition
DIN SPEC 70121	Define the required charging path and interoperability targets.
ISO 15118-2	Define charging services, authorization mode and certificate requirements.
ISO 15118-20	Define service set, AC/DC and any bidirectional scope, authorization mode and TLS 1.3 support.

Protocol-family coverage does not imply that every optional service, vehicle function or EVSE combination is enabled in every build. Agree the feature matrix before configuration. Bidirectional power transfer also requires compatible power electronics, vehicle control and EVSE support.

1.2 Three deployment paths

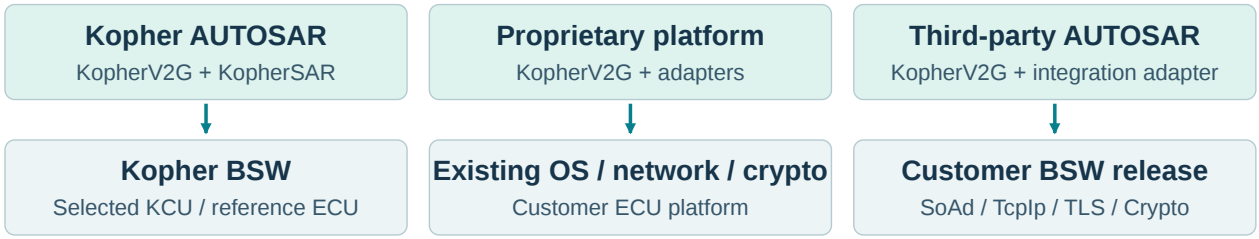

Figure 1. Three integration paths with a shared EVCC protocol core.

1.3 Responsibility boundary

KopherV2G manages charging communication. The vehicle application owns charging permission, operational limits and coordination with BMS, OBC and VCU. The complete ECU integration must connect protocol outcomes to the required vehicle actions and independent safety mechanisms.

2 Software architecture

The following AUTOSAR Classic example locates the protocol services below the RTE in BSW / middleware. It is a logical integration view; module placement and binding details depend on the target platform.

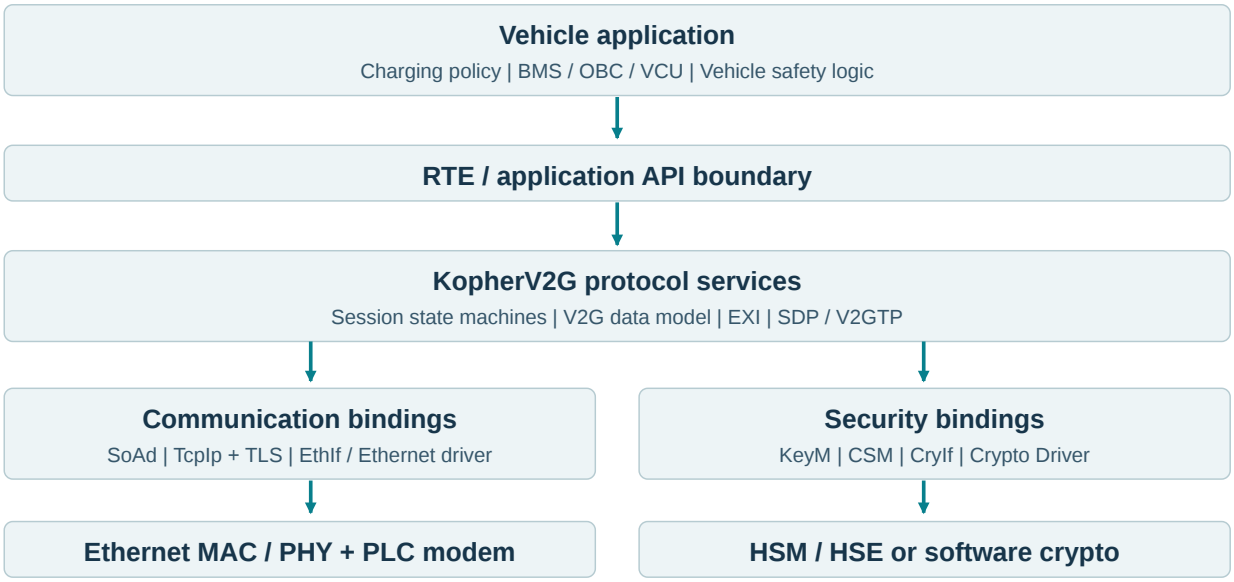

Figure 2. Logical module boundaries. Communication and cryptographic services are separate integration paths; the diagram is not a packet-by-packet call sequence.

2.1 Runtime communication

The application exchanges charging requests, limits and status through the agreed RTE or application API. The protocol core encodes or decodes V2G messages and uses platform bindings for discovery and transport. In AUTOSAR integrations, the communication path connects to SoAd, TcpIp and Ethernet interface services.

2.2 Configuration and security

EcuC PDU identifiers, SoAd routes and socket groups, TcpIp controller settings and security references configure these relationships. EcuC is not a runtime packet-processing hop. TLS consumes certificate/key-management services and cryptographic jobs; KeyM, CSM, CryIf and the Crypto Driver form the target security integration.

SLAC is the PLC link-pairing procedure. Its software, driver and modem responsibilities are defined by the selected platform; it should not be treated as a physical hardware component.

3 Discovery and charging session

Figure 3 shows an illustrative ISO 15118-20 DC session. Authorization, service selection and charging-control branches depend on the negotiated protocol and service; a single fixed stage count is not a conformance criterion.

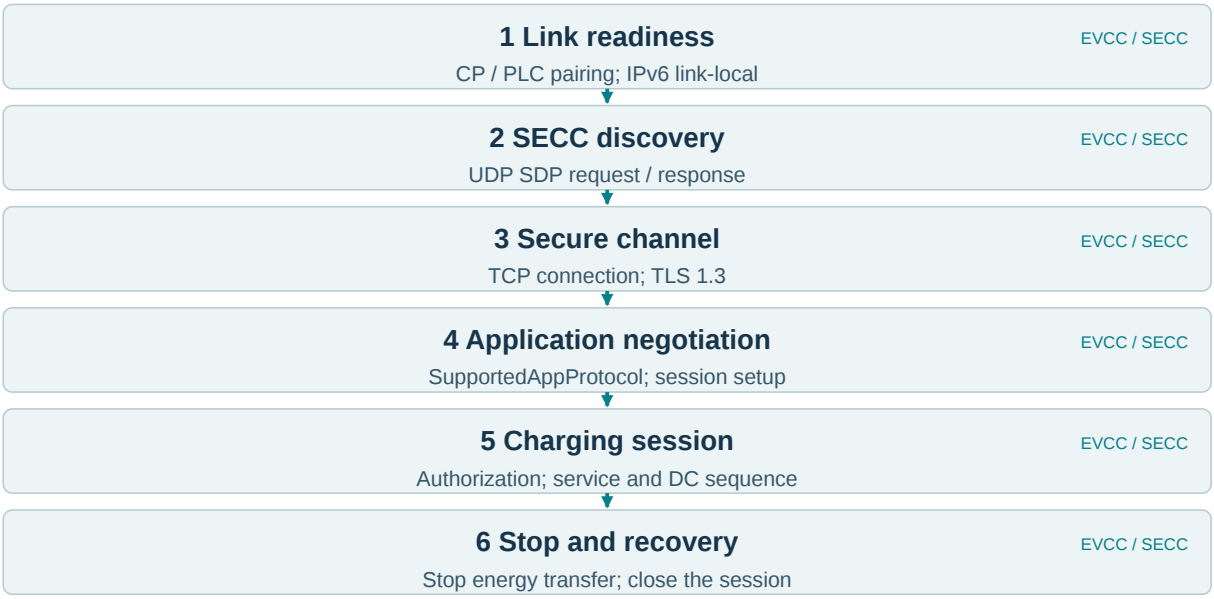

Figure 3. Session-level sequence; each step must be correlated with vehicle state and timing.

3.1 Discovery is endpoint selection

After link readiness and IPv6 link-local configuration, the EVCC sends SDP to multicast ff02::1 on UDP port 15118. The response provides the SECC IPv6 address, service port and security/transport selection. SDP does not assign the EVCC IP address. The EVCC then connects to the returned endpoint.

3.2 V2GTP payload reference

Payload type	Use
0x9000 / 0x9001	SDP request / response
0x8001	SupportedAppProtocol negotiation
0x8002	ISO 15118-20 common messages
0x8004	ISO 15118-20 DC messages

The payload type must match the selected message family. SDP is a discovery payload; it is not an EXI-encoded application message.

4 TLS, Plug & Charge and key services

Transport security and application authorization are distinct. TLS secures the communication channel, while the application protocol performs the selected authorization procedure. ISO 15118-20 requires TLS 1.3 for both EIM and Plug & Charge (PnC) charging communication.

4.1 Target service responsibilities

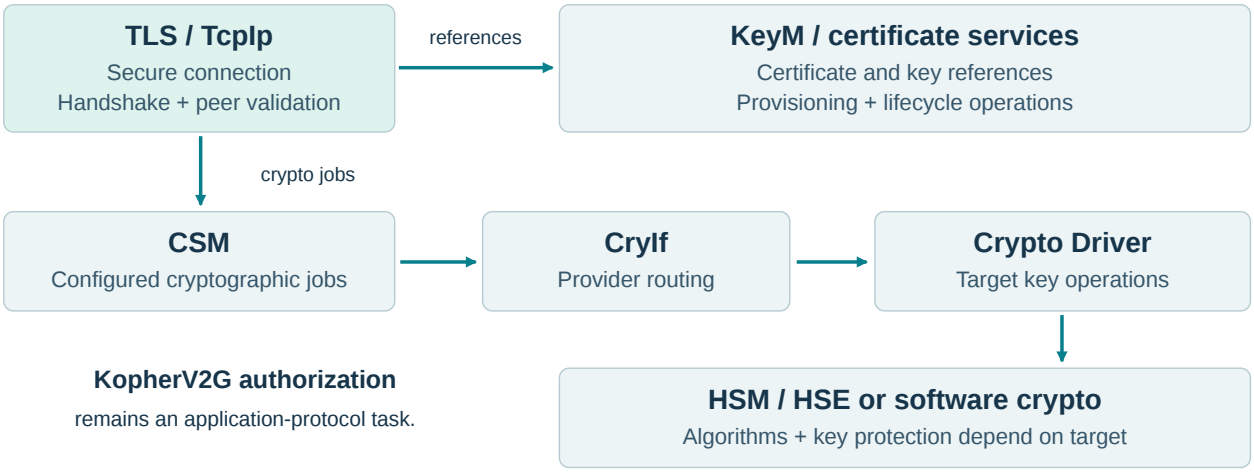

Figure 4. Logical TLS service dependencies. Arrows identify requests, not a complete call sequence.

4.2 Security integration inputs

- Identify trust anchors, certificate chains, key ownership and the provisioning process for the target authorization mode.
- Confirm TLS version, algorithms, certificate formats and key operations against the actual BSW, TLS library, driver and hardware.
- Define storage persistence, update permissions, time validity handling and behavior after a credential or communication failure.
- Agree which traces can be retained. Record certificate identifiers and error categories without disclosing private keys or secrets.

4.3 Verification evidence

Demonstrate a successful secure session on the selected ECU and capture negative cases such as an untrusted peer, an invalid or expired certificate and a failed cryptographic operation. Correlate the error with the application decision and vehicle charging state. Hardware presence alone does not establish that the selected TLS and PnC requirements are supported.

5 Vehicle and platform interfaces

The interfaces below are conceptual integration contracts for project planning. Their concrete signal names, types, units, ranges, update periods and API signatures are defined in the target integration package.

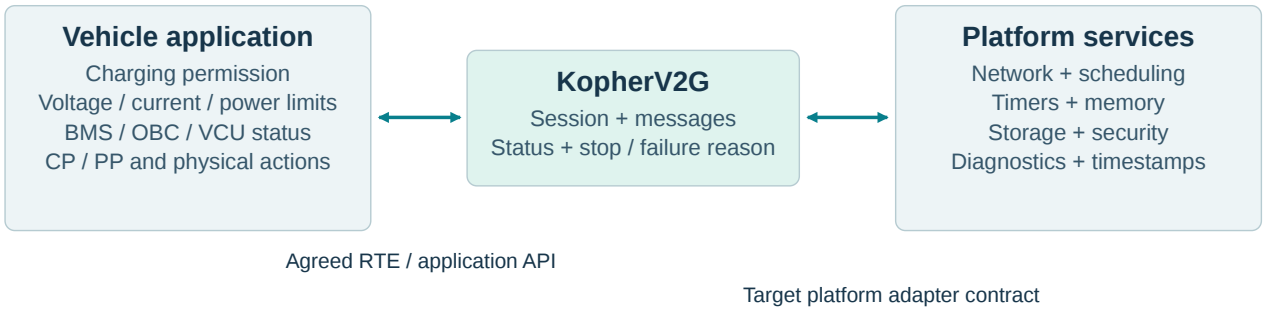

Figure 5. Integration boundaries. Signal types, units, validity and API signatures are project-defined.

5.1 Vehicle application contract

Interface group	Information and behavior to define
Charging permission	Readiness, inhibit reasons, operating mode and when charging may start or continue.
Charging limits	Voltage, current, power and energy-related parameters for the selected service; units, validity and update behavior.
Vehicle status	BMS / OBC / VCU state, fault information and communication freshness.
Session status	Selected protocol and service, authorization result, progress, stop reason and recovery outcome.
Physical interfaces	CP / PP state, PLC readiness and applicable power-path actions; assign the owner of each action.

5.2 Platform adapter contract

Adapter	Items to agree
Network and scheduling	UDP/TCP bindings, socket lifecycle, RX/TX dispatch, task execution and concurrency model.
Timing and memory	Time source, timeout dispatch, buffer ownership, allocation strategy and target resource budget.
Persistent data	Configuration and credential storage, restart behavior and update consistency.
Security	TLS provider, certificate/key references and supported cryptographic operations.
Diagnostics	Events, session identifiers, timestamps, trace correlation and sensitive-data handling.

5.3 Failure and recovery

Define behavior for stale vehicle data, cable removal, transport loss, timeouts, authorization failure and ECU restart. Verify the complete outcome: protocol state, application notification, vehicle action and conditions for starting a new session.

6 Integration workflow and target hardware

6.1 Engineering workflow

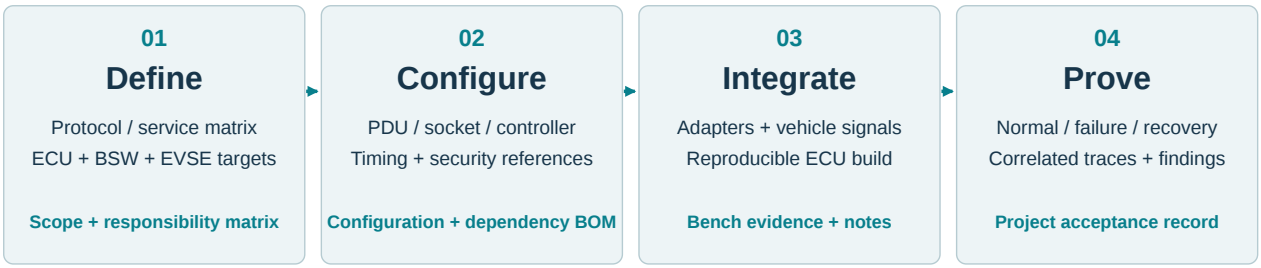

Figure 6. Inputs and outputs across the integration lifecycle.

6.2 Reference platform choices

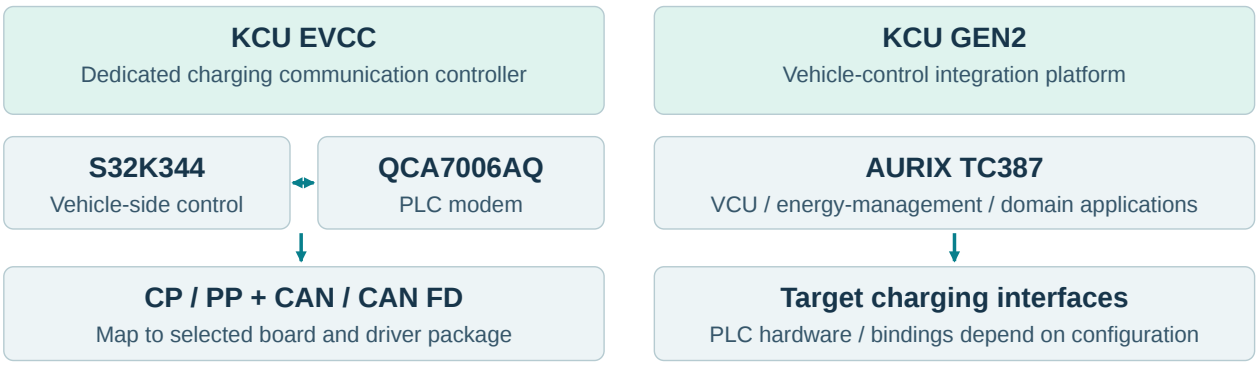

Figure 7. Functional hardware choices. Confirm the board revision, I/O population and electrical specification.

Select the actual board revision, I/O population and driver package before interface mapping. Pin assignments, connector definitions, electrical ratings and production options belong to the selected hardware specification.

6.3 Customer input package

- ECU hardware revision, toolchain, OS / BSW versions and third-party component licenses.
- Vehicle signal dictionary and ownership of charging permission, limits, physical actions and diagnostics.
- Protocol/service matrix, security provisioning concept and representative EVSE / SECC test targets.

Resource use and timing margins must be measured on the configured target. This guide does not substitute generic hardware capacity for measured stack performance.

7 Deliverables and acceptance evidence

Agree acceptance criteria before integration begins. Each criterion should name the target configuration, expected behavior and the evidence needed to decide whether it passed.

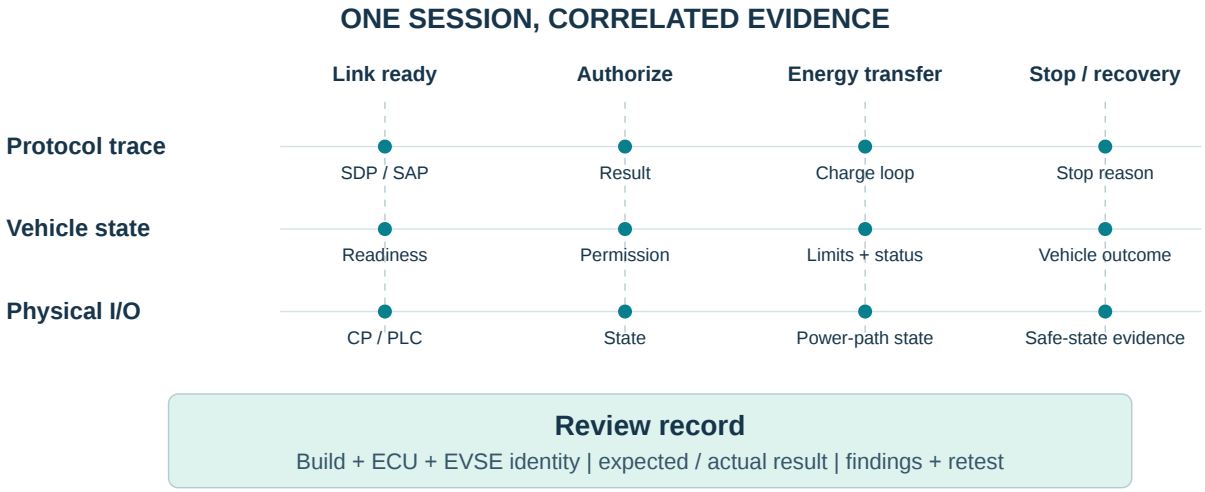

Figure 8. Align protocol, vehicle and I/O evidence on an agreed time base.

Deliverable	Review or acceptance evidence
Protocol core package	Selected protocol and service flows, version identification, message traces and state/timer review.
Platform bindings	Network, timing, memory, persistence and security tests on the target ECU / BSW release.
Vehicle integration	Bench or HIL evidence that protocol states match BMS / OBC / VCU and CP / PP / I/O behavior.
PnC / security package	Provisioning assumptions, certificate-chain checks and secure-session positive and negative cases.
Interoperability package	Tested EVSE / SECC and versions, normal sessions, faults, stopping and recovery.
Integration documentation	Build and configuration instructions, dependency BOM, release notes and known limits.

7.1 Minimum session evidence record

- Record the software build, configuration revision, ECU/board revision, BSW/TLS/driver versions and EVSE identity.
- Correlate network traces, protocol state transitions, vehicle signals and relevant I/O on an agreed time base.
- Record the scenario, expected result, actual result, stop/failure reason and reproducibility information.
- Close findings with retest evidence or document an explicitly agreed release limitation.

7.2 Delivery and licensing boundary

A project package may include the software core, reference configuration, platform adapters, integration guide, test cases and traces. Source-code or binary rights, adaptation work, maintenance, support and third-party licenses are defined by the project agreement and dependency BOM.

Successful testing against selected chargers is evidence for those configurations. Broader protocol conformance or certification claims require their own agreed scope and supporting evidence.

8 Terminology and references

Term	Meaning
EVCC / SECC	Electric Vehicle / Supply Equipment Communication Controller
EIM / PnC	External Identification Means / Plug & Charge
SDP / SAP	SECC Discovery Protocol / SupportedAppProtocol negotiation
V2GTP / EXI	Vehicle-to-Grid Transport Protocol / Efficient XML Interchange
SLAC	Signal Level Attenuation Characterization for PLC link pairing
RTE / BSW	AUTOSAR Runtime Environment / Basic Software
KeyM / CSM / CryIf	Key Manager / Crypto Service Manager / Crypto Interface
HSM / HSE	Hardware Security Module / Hardware Security Engine

8.1 Source material

[1] KopherBit, KopherV2G English product presentation, message-flow final edition, September 2026. Product boundaries and delivery model are derived from the supplied slides 9-19. Slide numbering refers to file order.

8.2 Technical references

[2] CharIN, DC BPT interoperability guide for ISO 15118-20, V1.0

Used to check the Part 20 TLS 1.3 and message payload distinctions. The guide is an interoperability reference, not a substitute for the applicable ISO standards.

[3] AUTOSAR Classic Platform R24-11, Specification of TCP/IP Stack

Reference for AUTOSAR network/security integration. The project must use the specifications and interfaces corresponding to its selected BSW release.

8.3 Project-controlled specifications

The applicable editions of DIN SPEC 70121, ISO 15118-2 and ISO 15118-20, target hardware documents, software release notes and customer interface specifications complete the project reference set. Confirm editions and amendments at project kickoff.

For configuration-specific integration documentation, contact the KopherBit project team through the product website.

KopherV2G product information