Eichrecht Whitepaper

OCPP Plugfest @ PIONIX 29-Feb-24

OPEN CHARGE ALLIANCE

www.OpenChargeAlliance.org

Agenda

1. Background
2. Scope of Whitepaper
3. Development process
4. OCPP 1.6 Implementation overview
5. OCPP 2.0.1 Implementation overview
6. Q&A

Background

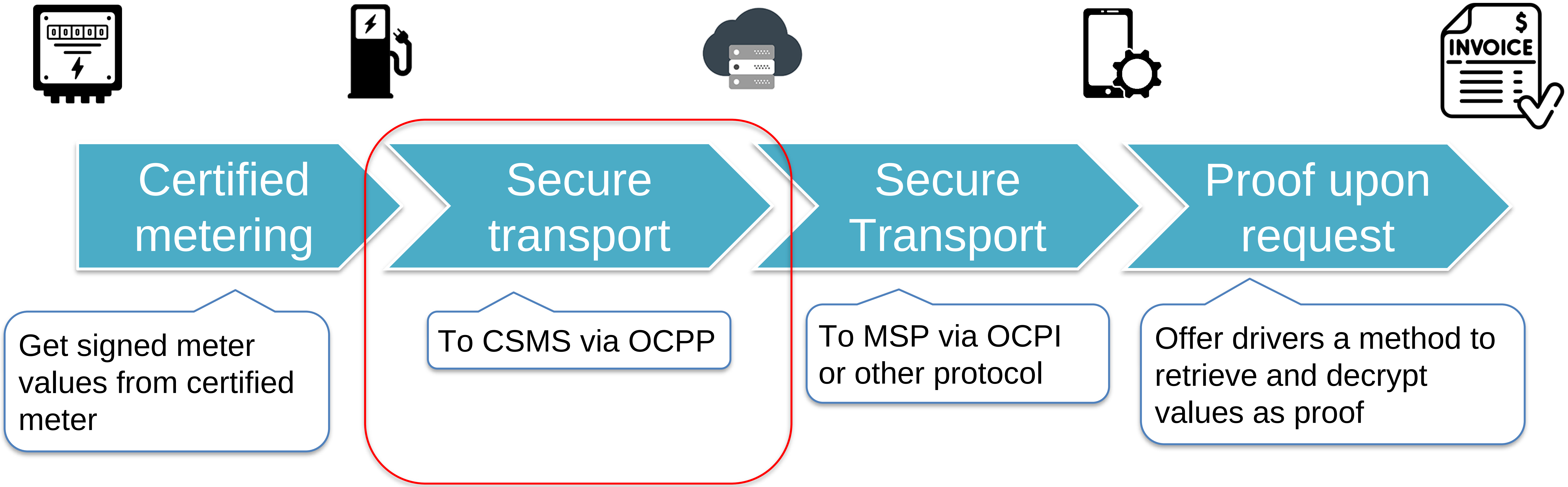
Legislation (Germany) context

- Amendment to MessEG was passed 1st Jan 2015
- Compliance deferred until 1st Jan 2024 where it came into force

Requirements

- Physically measured quantities in the supply of electricity must be collected and processed in such a way to exclude deliberate and erroneous manipulation
- Customer must be able to verify each invoice is valid for the above otherwise cannot be charged for the energy/transaction
- Relevant measurands: (k)Wh and/or Time (s)

Scope of Eichrecht vs Whitepaper



Certified metering

Get signed meter values from certified meter

Secure transport

To CSMS via OCPP

Secure Transport

To MSP via OCPI or other protocol

Proof upon request

Offer drivers a method to retrieve and decrypt values as proof

Scope of the Whitepaper

And therefore should be a method consistent with any future, similar legislation

Development Process

- Approach similar to a standards document – drafts and comment periods
- Lead author – Shell Recharge (inc. ubitricity)
- Supporting & commenting parties include PIONIX, Alfen, OCA, Chargepoint, GraphDefined
- 2x commenting periods – one complete, one about to start

	2023			2024			
	Oct	Nov	Dec	Jan	Feb	Mar	Apr
First draft authoring	🟦	🟦					
v0.5 Comment period			🟩	🟩			
Second draft authoring				🟦	🟦		
v0.7 Comment period						🟩	
Editorial tidy-up & publication							🟦

OCPP 1.6 Implementation – High Level

Concept: Transaction

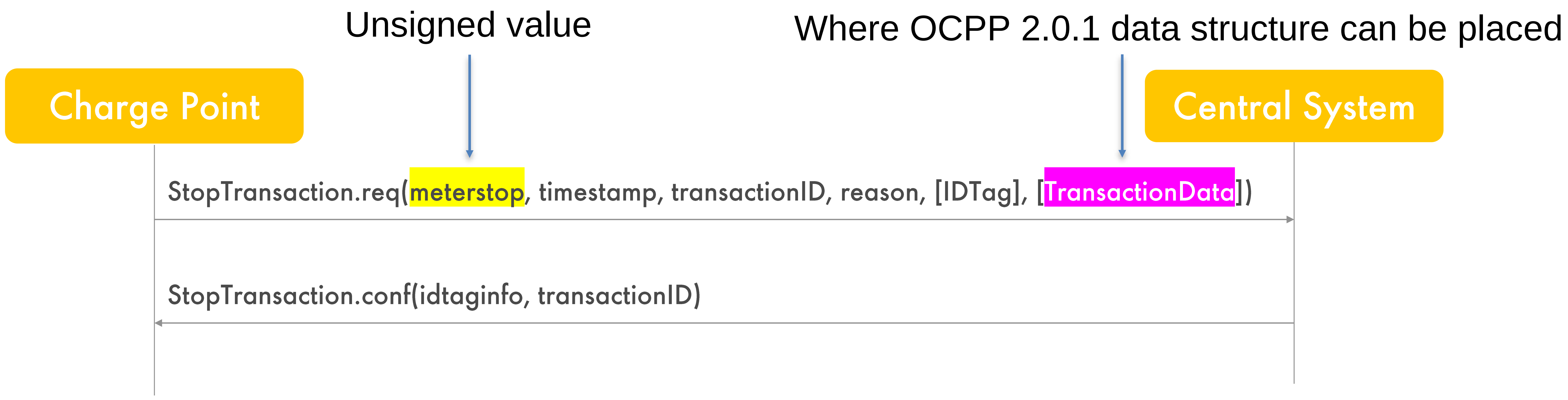
- Pull data structure from OCPP 2.0.1 – *signedmeterValueType* & configuration settings
- Escape payload for compatibility
- Sequence of messages:

“regular” transaction
(timestamp start/stop)

Eichrecht transaction
(timestamp meter values)

1. Pre-requisites include:
   a. Plugin
   b. Authorise (RFID, ISO15118, RemoteStart, integrated payment)
2. Start charging
   a. *StartTransaction*
   b. Power transfer
3. Optional: intermediate meter values (signed or unsigned)
4. Stop charging
   a. Cease power transfer
   b. send *StopTransaction* (inc. start & Final signed meter value)

Example StopTransaction



Unsigned value

Where OCPP 2.0.1 data structure can be placed

Charge Point

Central System

StopTransaction.req(meterstop, timestamp, transactionID, reason, [IDTag], [TransactionData])

StopTransaction.conf(idtaginfo, transactionID)

- TransactionData can have multiple instances of SampledValue with different contexts
- For this whitepaper, utilise context Start & End

Configuration settings

- MeterPublicKey[ConnectorID] – new key
- PublicKeyWithSignedMeterValue – reused key from 2.0.1
- SignedIntermediateMeterValues – new key

OCPP 2.0.1 Implementation

Existing Capability

- The main concern for this Whitepaper is OCPP versions below 2.x
- Concept used for 1.6 is already mostly already possible via 2.0.1, with some variable changes
- New variables to be added to Device Model
- Since signed values can be sent for the beginning transaction message TransactionEvent(Started) as well as in TransactionEvent(Ended) leave up to the implementer to decide since no extra data is required

Q&A