UECS

Development of a smart charging management system for EVs

Xiaoying Sun

SCANIA

Supervisor: **Hampus Andersson**
Fjellström Jonatan

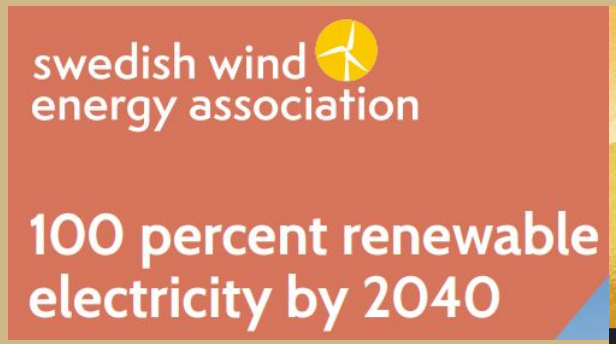
Groupmate: **Christine Arkbo**

SCANIA

Background

What brought us to this research?

Policy Support



Energy Shortage

Flood of new models

Global rapid growth of EVs

What is Smart Charging?

(maximize customer benefits)

Uncontrolled Charging **+**

（immmediate charge）

Electricity Prices

Maximum grid capacity

Sustainable Energy

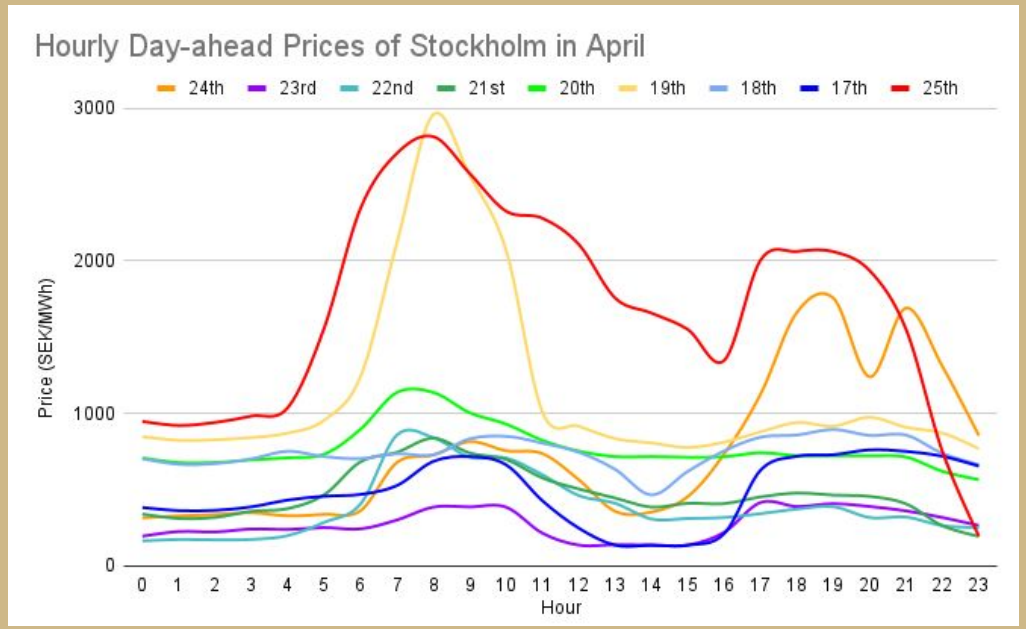
V2G

…

= Smart Charging

The electricity market has considerable potential

From Day to Day

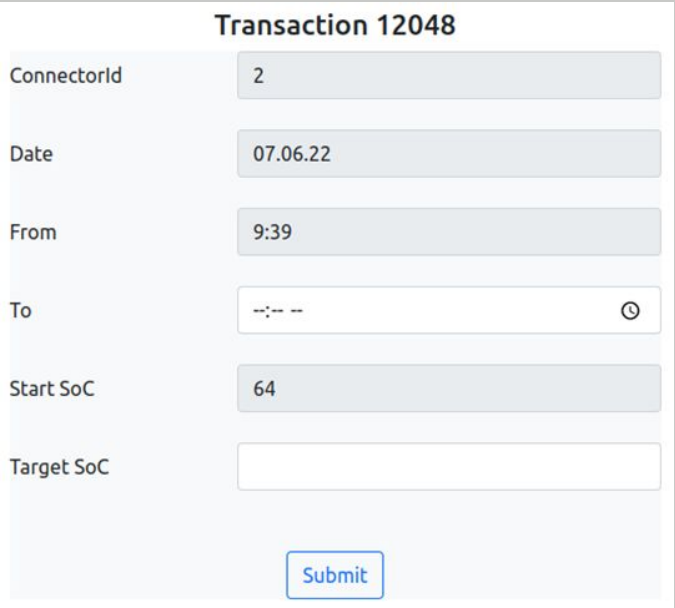


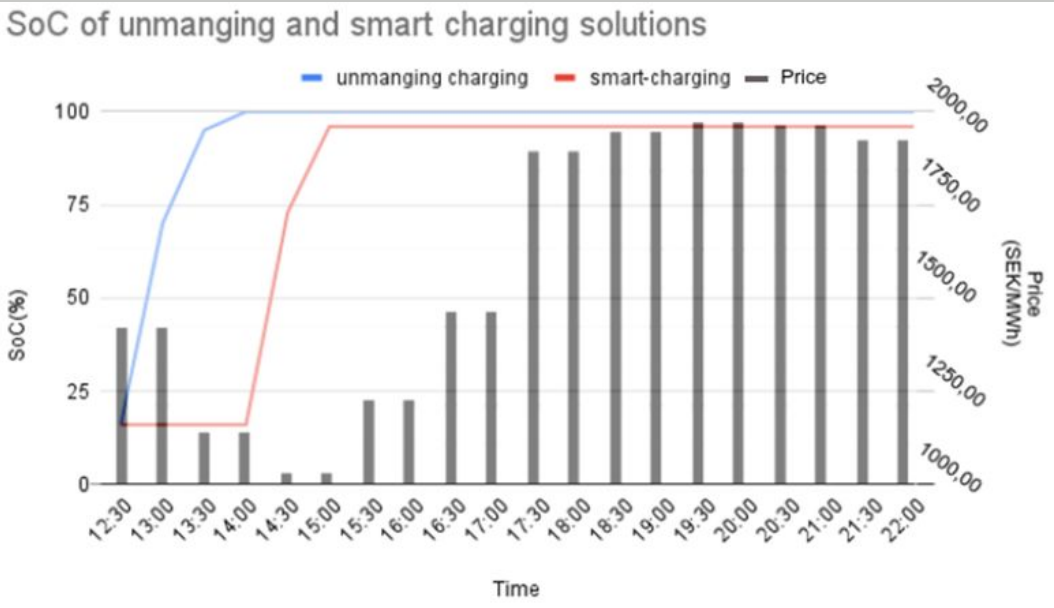
From Hour to Hour

NORD POOL

Problem Decomposition

Picture the final result?



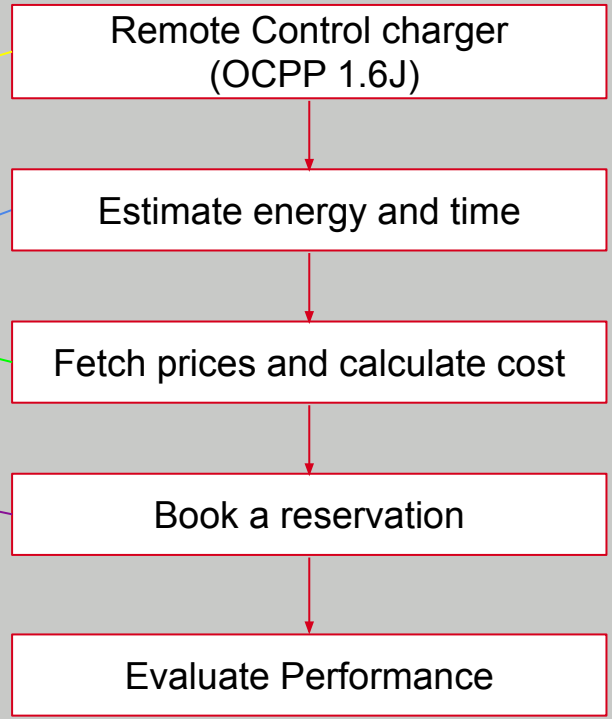


How did we approach the problem?

Central system should control Charger to

Stop charging when price is high,
Start charging while price is low.

Finish charging before departure time



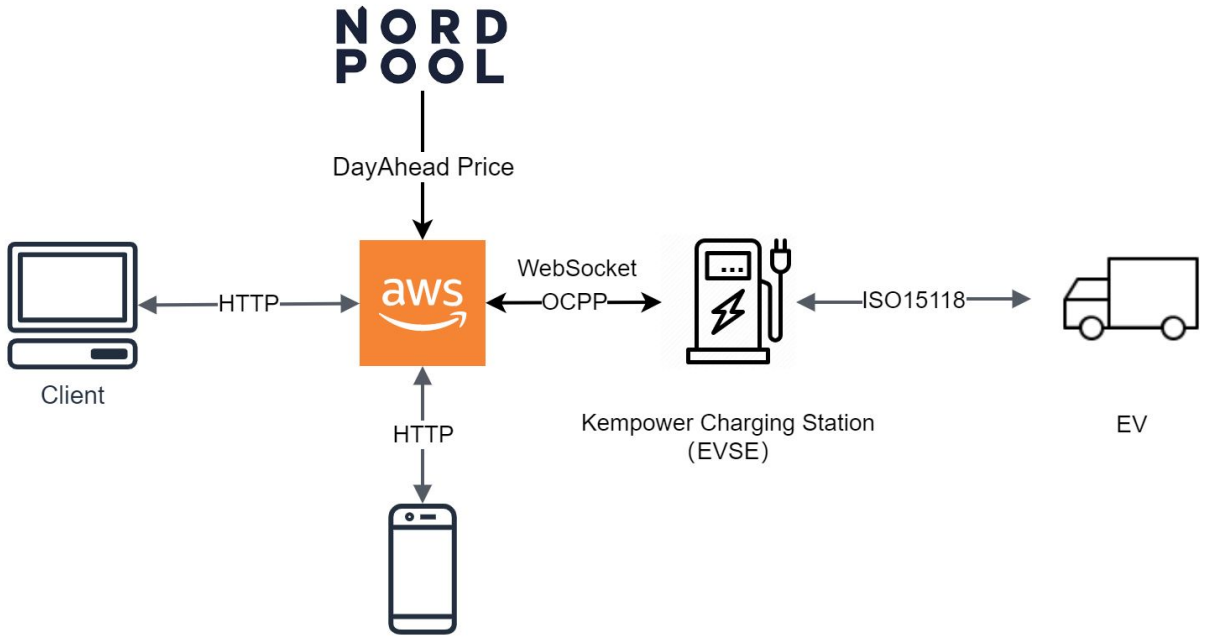
4 Sprints

Sprint 1	**Server Core Handler**	Remote Control charger (OCPP 1.6J)
Sprint 2	**Capacity Estimator**	Estimate energy and time
Sprint 3	**Price Fetcher + Cost Calculator**	Fetch prices and calculate cost
Sprint 4	**Profit Maker + MAPE**	Book a reservation

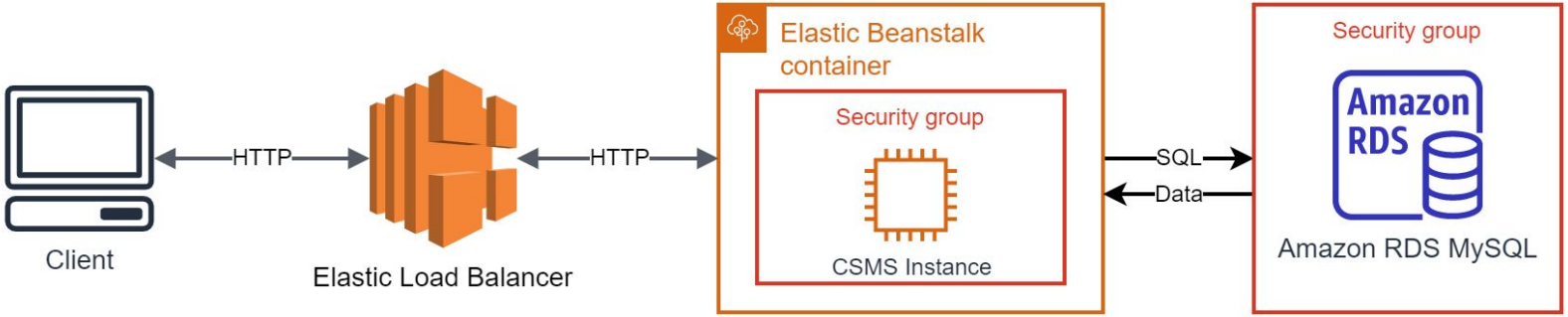
Implementation

Main Framework

Spring Boot + AWS + MySQL



AWS Deployment



Free Tier:

+ Amazon Elastic Beanstalk
+ Amazon RDS MySQL
+ Amazon VPC (Security Group)

Sprint 1 Server Core Handler

CS Central system
CP Charging point

CP -> CS

```
client.connect(url, new ClientEvents() {...})
```

CS -> CP

```
server.send(currentSessionIndex, request);
```

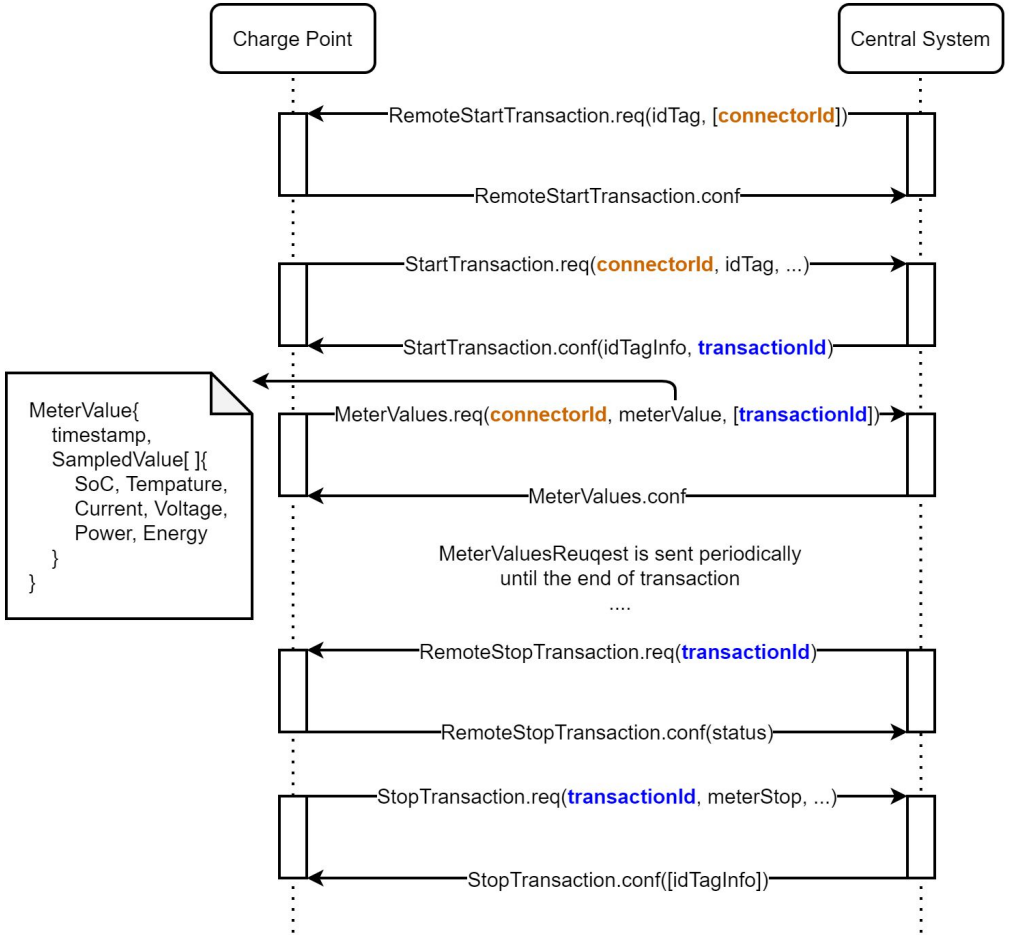
Sprint 1 Server Core Handler

AWS log:

```
StartTransactionRequest{connectorId=1, idTag=, meterStart=0, reservationId=null,
timestamp="2022-03-23T15:02:10.116Z", isValid=true}

MeterValuesRequest{connectorId=1, transactionId=42, meterValue.length=1,
isValid=true}

MeterValuesRequest{connectorId=1, transactionId=42, meterValue.length=1,
isValid=true}

StopTransactionRequest{idTag=null, meterStop=412,
timestamp="2022-03-23T15:45:58.202Z", transactionId=42, reason=Remote,
transactionData.length=1, isValid=true}
```

Sprint 1

Server Core Handler



How could we get requested Energy and Time?

OCPP **does not send requested energy as an absolute value**

Table 3.3.1: Measurements included in *MeterValues* from KemPower Charger.

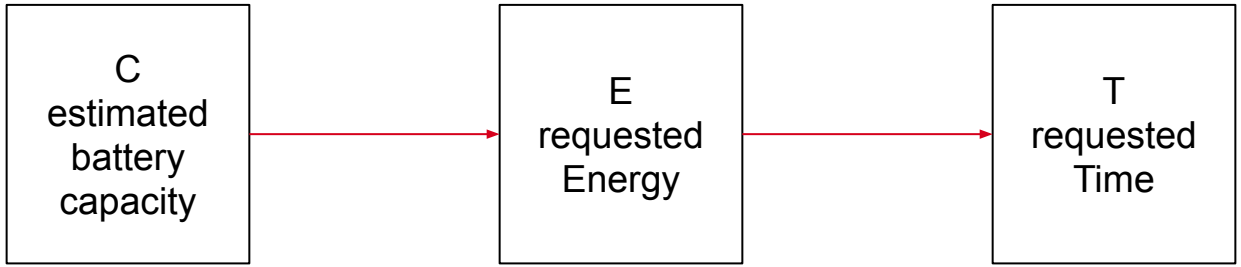
Measurement	**Notation**
SoC	EV Battery State of Charge %, if known
Temperature	Temperature (Celsius) inside the Charger or Satellite housing
Current.Import	Charging current from Charger to EV (amps)
Voltage	Charging voltage (volts)
Power.Active.Import	Charging power (watts)
Energy.Active.Import.Regsiter	Energy outputted from Charger to EV (watt-hours)

Sprint 2 Capacity Estimator

We need Capacity for requested Energy and Time.

$$E = C \times (SOC_{target} - SOC_{start})$$

$$T = \frac{E}{mid(Power)}$$

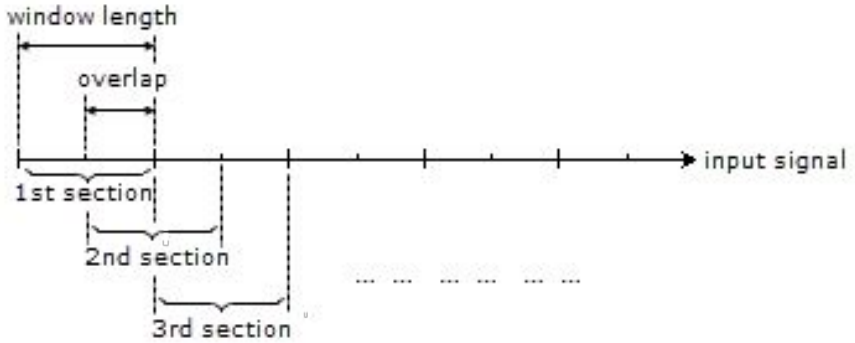


Sprint 2 Capacity Estimator

Move fast! Or the battery is full.

MA (Moving average) is a technical indicator that market analysts may use to determine the direction of a trend.

$$C = \frac{\sum_{i=0}^{N-n} \frac{E_{i+n} - E_i}{SoC_{i+n} - SoC_i}}{N - n}$$



Sprint 3 Price Fetcher + Cost Calculator

Nord Pool Market Data API:
ftp://studentnordic:nordic_2020@ftp.nordpoolgroup.com

Cost (SEK) = **Time** (h) * Hourly Price (kWh/SEK)

Profit = Uncontrolled Cost - Smart Cost

How to evaluate algorithm's performance?

MAPE (Mean absolute percentage error) is a measure of how accurate a forecast system is.

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

The less MAPE is, the better prediction is.

Sprint 4 The first MAPE is 12.547%

Table 5.2.1: Prediction accuracy of estimated time.

Transaction ID	Actual Time(h)	Estimated Time(h)	APE (%)
16122	1.351	1.633	20.873
17817	0.233	0.250	7.296
11201	0.543	0.480	11.602
11941	0.182	0.183	0.549
12368	0.525	0.400	23.810
18957	0.226	0.250	10.619
12706	1.323	1.150	13.076

Why some PEs are so large?

P1: When SoC=100%, charging transaction didn’t stop

SOC<100%		SOC=100%	
11111 (42-67)	0.882%	17060 (23-100)	8.932%
18106 (46-99)	0.074%	13188 (61-100)	0.781%
14414 (52-75)	0.111%	19959 (58-100)	28.404%
15038 (82-90)	0.069%	15328 (61-100)	26.651%
16056 (67-75)	0.259%	10367 (59-100)	2.119%

S1: Manual interruption

Actual time is always bigger than estimation

1. Cut off transaction when **SoC=100%**
2. Delete transaction when **size(Samples) < 5**

P2: Downsides for capacity estimation

- Energy loss in the process.
- Battery self-discharge.

S2:

Time is not be the best measurement.

Estimated capacity VS. Real capacity

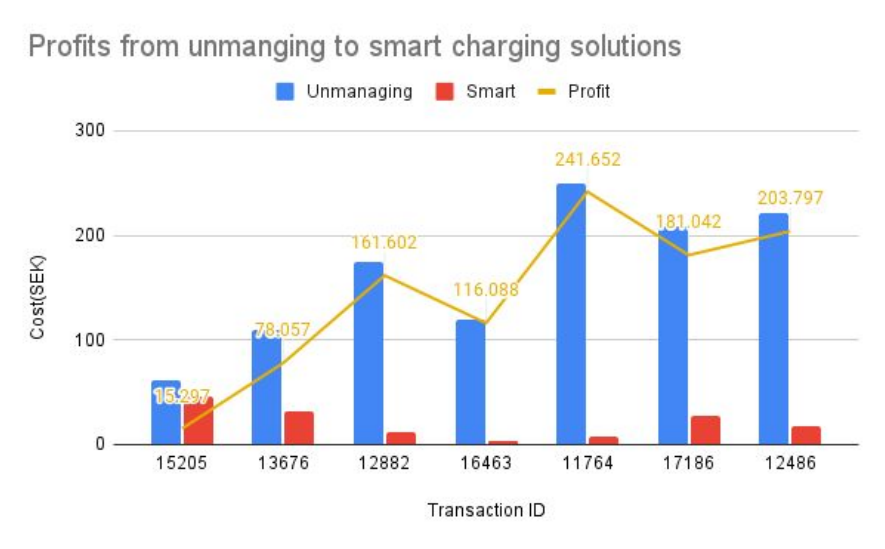
Pre-build a database for all registered trucks (MAC address, battery capacity)

Demo

Central System

Strengths

Profitable



The proposed smart charging method

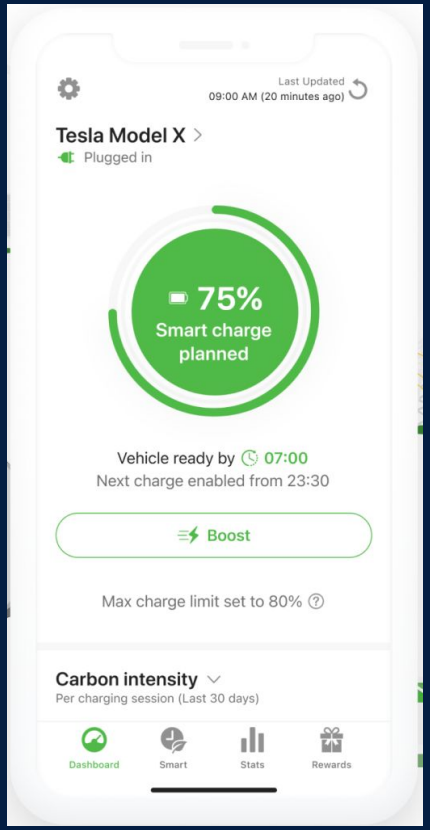
- indeed leads to considerable benefits.
- went beyond the call of duty.

Scalability

MVP (Minimum viable product)

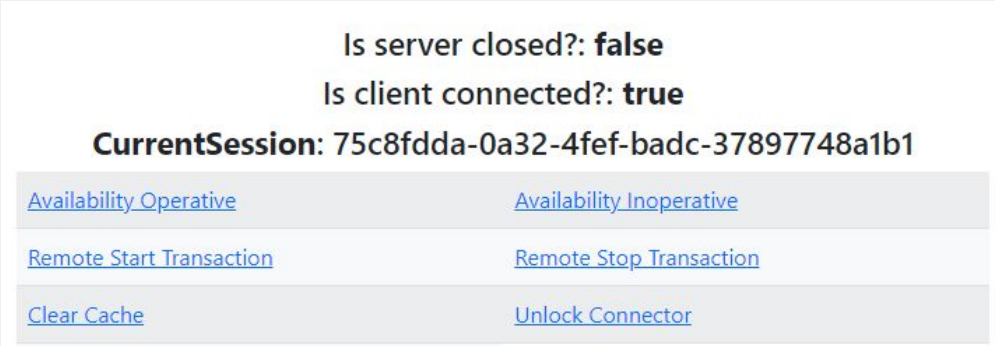
+ **Front-end Integration.**
  (Mobile App)
+ **Fine-tuning database**.
  (Pre-build a truck database

  cut off the calculated time)
+ **Charging fleet**
+ **V2G**



Future Work

From single to multi-thread



Come last, Serve first

Risk to lose current truck's track

From OCPP 1.6 to 2.0.1

Better handling of **numerous transaction-related requests**.

StartTransaction
StopTransaction
MeterValue
StatusNotification

TransactionEvent

More possibilities for future expansion of smart charging

Amazon Security Group

The current inbound rules are **too broad**, which expose the server in potential danger.

Security group rule...	IP version	Type	Protocol	Port range
sgr-078c7673dae22ba...	IPv4	All TCP	TCP	0 - 65535
sgr-0d03250745f4781...	IPv6	All TCP	TCP	0 - 65535
sgr-07afda79b428c9b63	–	HTTP	TCP	80

Also From HTTP to HTTPs

Thank you all for attending

Q&A