Introducing Motown.io (13-8-2013 – auke.hoekstra@alliander.com)

Introduction: the future of "talking to charge points"

Regarding charge points, more than 99,99% of the market (real calculation) is ahead of us. We need to plan ahead. In the future, many "secondary actors" (to borrow a term from 15118) will want to communicate with the charge point or the EV that lies behind it:

- CSO's will need to manage charge points at higher quality and against lower costs using better diagnostics.
- E-mobility providers will offer the customer more choice in energy usage and pricing plans based on local and temporal conditions.
- Energy suppliers will say: "please use this excess in solar and wind energy: its cheap".
- DSO's will say: "pay extra or throttle if you are coming close to our maximum network capacity".
- Companies and households will use smart charging commands to lower their utility costs by peak shaving.
- Cars will receive certificates and sign receipts for power consumed.
- Third party application developers will give end-users direct access to available charge points, reviews, diagnostics about their current session and historic usage data.

These are just some examples of many interactions with charge points that are already foreseeable but there will be many more because charge points play a pivotal role in the world wide energy transition, probably the biggest undertaking in the history of humanity. In that transition, charge points are the physical interface between two of the world's largest industries: automotive and utilities. Motown is a way to open up charge point networks to this new business ecosystem. This goes beyond managing charge points.

Why OCPP is not enough

OCPP is a standardized way of talking to charge points. It makes charge points interchangeable. It is needed but it is not enough. We don't want to force/allow secondary actors to talk directly with all those different charge points (with all their different IP addresses, up-times and hardware-related quirks) and we don't want to make the charge points too expensive (by giving them multiple interfaces and a long memory).

So we need to move the problem to a place where intelligence, bandwidth and memory are cheap and where we can offer an interface to many charge points at once. In short: we need to use a server in a datacenter on the Internet.

Why current back office systems are not enough

Current back office systems of CSO's address this problem of moving access from charge points to a place on the Internet. Good! But they don't have a standardized interface for all secondary actors. They seem unaware of the enormous burgeoning business ecosystem in which their primary function is foremost that of an interface towards a multitude of other businesses which size they cannot match and whose whishes they can only partly know and comprehend.

The worst case scenario is that all vendors of back end systems for CSO's develop their own "value added" interface, thereby creating a chaotic and cumbersome situation for all other secondary actors. This chaotic situation would slow down the emerging business ecosystem and would make charge point networks less desirable and profitable.

A step back: it is about commands and an event store

Let's try to make this as simple as possible. At the end of the day, charge points have only a limited set of messages, defined by OCPP.

The first thing these messages enable us to do is to give **commands** to the charge point like "start charging" and "release cable".

Furthermore, a database of all messages to and from the charge point is a treasure trove of information that secondary actors will want to access. We call this database the "**event store**". Accessing it without having to talk to the charge point means the information can be retrieved faster and without incurring data communication costs towards the charge point.

Motown is first and foremost a practical open standard

(Motown = MObility Transition Open source Webservices iNterface)

First we talk to all the secondary actors to find out how they wish to access the commands and event store. All this talk is too broad and time consuming for single CSO vendors but manageable for a consortium of secondary actors and CSO vendors comparable to the OCPP consortium (although a bit more professional based on the lessons learned from OCPP and on a market that is getting larger).

Secondly we translate these wishes into a comprehensive set of web services. The syntax and functionality of the web services is open and downloadable from the web so it is easy for everybody to participate. The focus should be on growing the pie, not on dividing the 0,01% of the pie that we already have.

Motown is a reference implementation of the open standard

Software vendors can decide to build the Motown web services into their applications but we want to go a step further. We also want to implement the standard in software, thereby showing what we mean *exactly* and also *checking* for ourselves that the standard makes sense (the proof of the pudding is in the eating).

[Tech talk: we see this increasingly in software developments. A good example is Java Enterprise Edition that is foremost an open standard but that also knows reference implementations like the application server glassfish (though most developers use alternatives).]

The reference implementation is not a threat to current vendors because it only has very limited functionality on top of the commands and event store. Therefor it is not usable as is. However, the implementation of the commands and the event store is done is such a way that it is extremely easy to add new web services and that all information is easily available. It will be a joy for developers.
[See details about the technical implementation elsewhere.]

Using the reference implementation brings added value.

Because it has very limited functionality on top of the web services, Motown is hardly a threat to current software vendors. The open source version will include a simple set of screens to do basic charge point maintenance in order to be able to evaluate Motown, but on its own it is not a competitor for the commercial suites that are available.

However, the reference implementation can be used as a sound fundament for further software development. Its structure is such that adding interfaces is easily done. Furthermore, software vendors don't have to code the web services and underlying events anymore. Finally, secondary actors can easily add functionality to Motown.

For example: suppose a charge point knows some special tricks that are not (yet) part of OCPP. The charge point manufacturer could try to sell the customer his own software suite but chances are the customer already has software for managing charge points that he wishes to keep. However, it is easy for the charge point manufacturer to write a simple add-on to the Motown open source. If the customer uses Motown, installing this add-on will give him all the extra functionality.