M2C Mobile App Development High Level Spec

At a meeting between OSNA and M2C (Mick Berry) in January 2014, M2C highlighted the need for a mobile phone app to compliment M2C's Electric Vehicle Smart Charger.

During the meeting the idea was further fleshed out and the following is a high level spec of the desired app functionality as understood by OSNA. This spec will form the basis for the deliverables on the app development project.

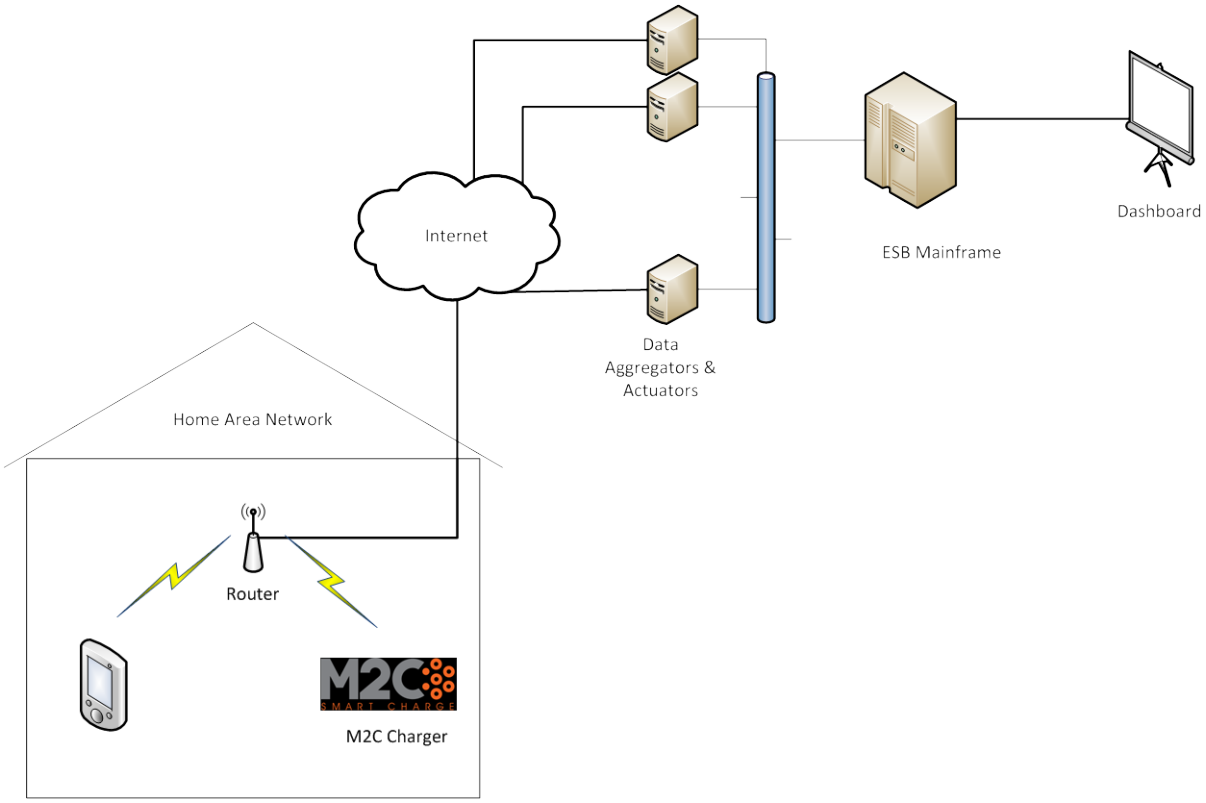
Background

ESB Networks (ESBN) and Eirgrid are responsible for the coordination and distribution of Electricity within the Irish electrical grid. In order for the ESBN to allow greater wind generation capacity to be connected to the grid, it needs to deal effectively with wind variability and wind generation predictability so as to maintain adequate system frequency control. Strategies include; reserve generation capacity during periods of low wind (termed spinning reserve), and preferably, the ability to dynamically change the load to match wind generation capacity.

Conventional power plants operate most efficiently and effectively at rated capacity so running with reserve under a 'spinning reserve' strategy costs money. There is however a potential to exploit load control. If ESBN could control large loads within the home in real time, across a large percentage of the grid, then they could shed/gain load to match the variability in wind generation. As electric vehicles become more prevalent, the demand they place on the grid will increase. This increase in demand however can be used to the ESB's advantage if the loading can be varied and time shifted to suit generation capacity.

A system could be developed whereby the chargers could use a statistical model to switch on/off based on the frequency of the supply. A statistical model element is required to prevent clusters of chargers switching in/out in a localised area at the same time, putting strain on eth local LV (low-voltage) network. Such a model would require extensive testing, and while it would provide a solution, the ESB would have no visibility or direct control of the 'switchable load'.

The figure overleaf outlines the architecture for a proposed solution which would provide ESBN with both visibility and near real-time control of M2C Chargers. This architecture could be further expanded to include additional devices within the home, including domestic heating and water heaters.



1 - Load Control Architecture Using M2C Vehicle Charger

The App

The mobile app is a key element within the proposed architecture. The mobile app allows the user to visualise various statistics relating to the charger (power usage, car charge rate, etc). More importantly the app also allows a user to set the desired 'Fully Charged Time' for their vehicle. This feature is particularly important as it will determine if the charger can vary the charge rate while the vehicle is attached.

App Operation

When a user first opens the app it will need to pair with the charger. This will need to be completed over an ad-hoc wireless link in order to configure the charger with the SSID and password of the user's home router.

Once this step has been completed the app can re-connect with the home Wi-Fi and communicate with the charger over the home network. When re-connected the app will need to identify the charger and pair with it.

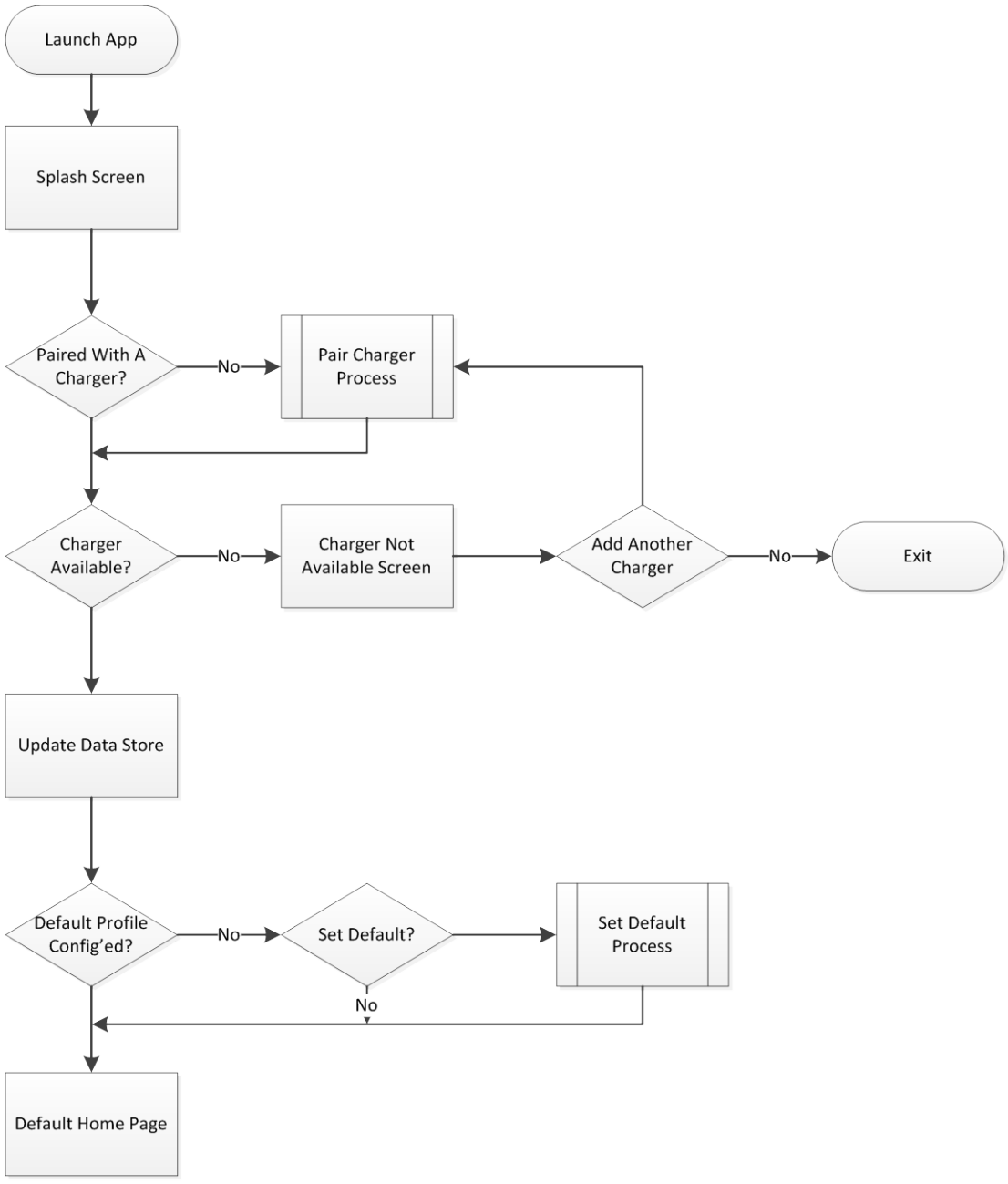
The app is then ready to download the data required; current charge status and any available stored statistics. At this point the first feature of the app is completed.

The app will then detect the set-up of the charger. The charger will have a default 'vehicle connected' behaviour which the user can then vary.

The user can choose between; available charge rates, delayed available charge rates and dynamic.

- Available Charge Rates: if multiple charge rates are available (e.g.: fast, medium, trickle charge) the user can select one as a default.
- Delayed Available Charge Rates: as with above but start after a fixed period or at a fixed time of day (e.g.: off-peak)
- Dynamic: This is the advanced option. This would allow the user to configure a timetable with 'Fully Charged Times' settings. Once configured, the charger can vary the charge rate to provide a full charge within the given timeframe (i.e.: Vehicle Connected Time -> Fully Charged Time).

As these are default settings the app should allow the user to do 'once off' actions also, i.e.: the user could select an immediate fast charge if they needed the vehicle outside their usual usage pattern.



2 - Sample App Sequence

Dynamic Rate – Why?

The dynamic setting described above underpins the availability of variable load within the grid, controllable by ESBN. If the charger has three charge rates and the vehicle can be fully charged in the period between Vehicle Connected Time and Fully Charged Time on all three, then ideally the charger would start charging at the mid rate. This would allow ESBN to increase or decrease the charge rate thereby increasing and decreasing grid load according to supply availability.

A number of key elements are required to support adoption of the 'dynamic' rate.

- ESBN can only vary the charge rate if it will not impact on availability of the vehicle.
- The 'variable load' data should be anonymized by data aggregators and presented to the ESB as summary data on a dashboard.

Things which will improve the adoption rate of the technology / 'dynamic' rate.

- Incentivised charges
- An expandable platform allowing users to control more devices.

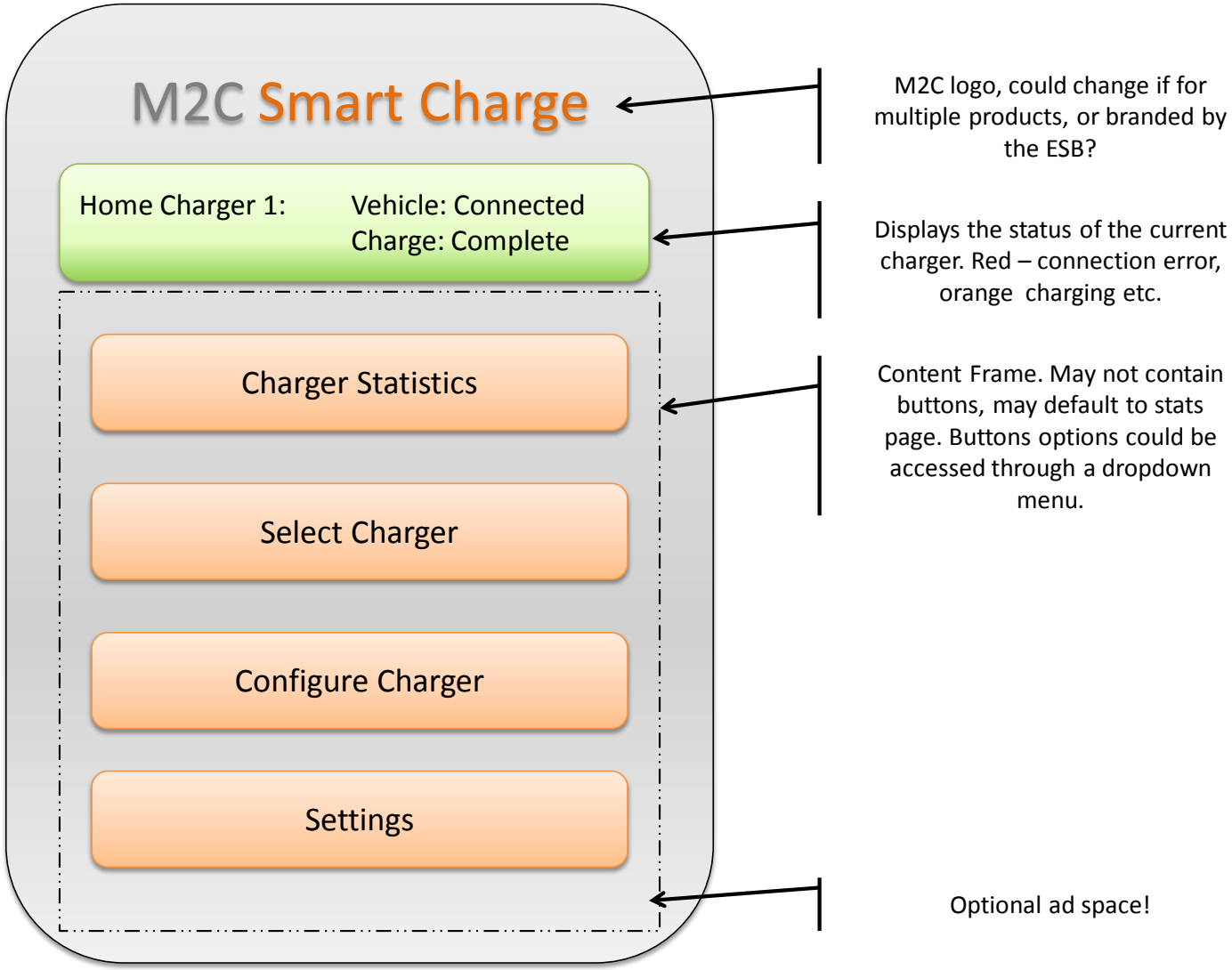
While this solution is dependent on the end user providing broadband access, this is unlikely to hinder the project. Areas where load variability will have the greatest impact are within built up urban areas where broadband penetration/reliability is higher. The customer most likely to make use of such technology is also likely to be more technology aware and therefore also have broadband access.

App Appearance

The core of the app will consist of a main frame with a status bar and a dynamic sub-frame.

The contents of the sub frame will vary as the user navigates through the app.

See overleaf for a mock up of how the app could look.



3 - App Screen Mock Up