Smarter Tariff – Smarter Comparisons

Technical Report

January 2023

Author

This independent report was produced for BEIS by:

Joshua Cooper (Hildebrand Technology)

Contents

Introduction

This document is intended as a reference for the reader who is interested in understanding the technical rationale behind the Smart Tariff Smart Comparison tool.

Up to date reference material is maintained online in the open source repositories plus working examples on ObservableHQ. Links to the relevant urls are contained within the body of the document.

The version of the code referenced is for Proof of Concept 3; Proof of Concept 4 (the final iteration of the tool) and all related documentation will be updated online.

Our Smart Tariff Smart Comparison example

The example front end web site https://smarttariffsmartcomparison.org/ will be used as an illustration to help organisations adopt the smart tariff, smart comparison technology. The full source code and build instructions are available at https://bitbucket.org/LDNSFO/stsc-front/ that accompanies this design narrative.

First Stage - Registration

For registration a chat interface design was used to add personality and increase user-friendliness, particularly for navigating energy data which can be perceived as complicated.

The functional goal is to verify the user's right to a meter and gain permission to access data.

In order to verify the consumer, two pieces of data are used:

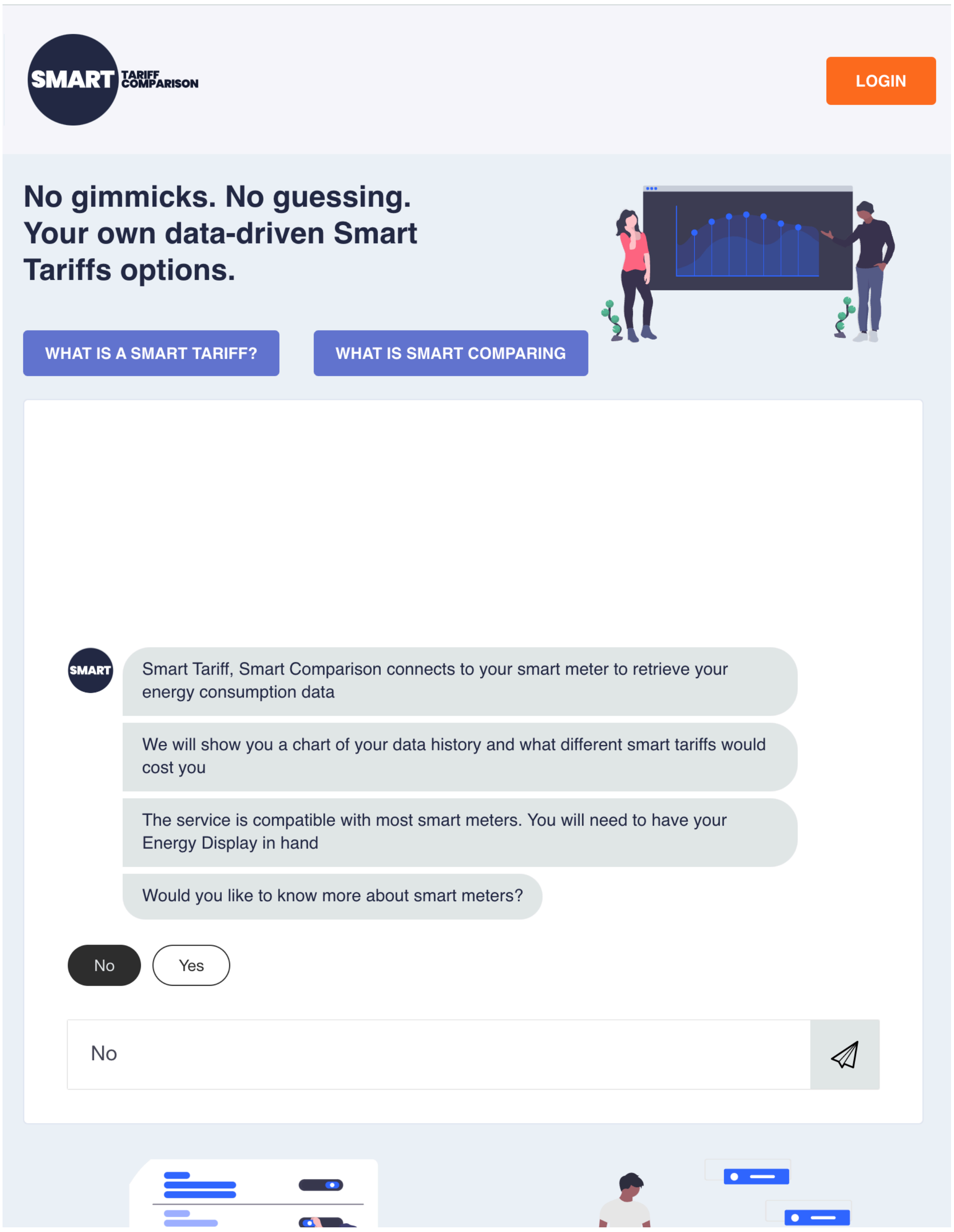
- Postcode - the postcode where the meter has been registered within the smart meter systems
- IHD EUI - or in-home display extended unique identifier, it is a long unique code printed on the in-home display

The nature of the IHD EUI is technical with 16 hexadecimal digits, although alternative verification mechanisms are more complicated. The design challenge is to help people find and easily enter this information.

Next there is a challenge of quickly retrieving the data from the meter. Because the data is traveling over slow mobile data links there is a chance that it may take anywhere from a few minutes to a few hours to request and receive data from the meter.

To identify users that may return after a long delay in retrieving data, a registration with an email address and password is implemented.

Figure 1: Initial chat interface for registration, it is a dialog that is intended to engage and be friendly while directing the user through complex informational input



No gimmicks. No guessing. Your own data-driven Smart Tariffs options.

Smart Tariff, Smart Comparison connects to your smart meter to retrieve your energy consumption data

We will show you a chart of your data history and what different smart tariffs would cost you

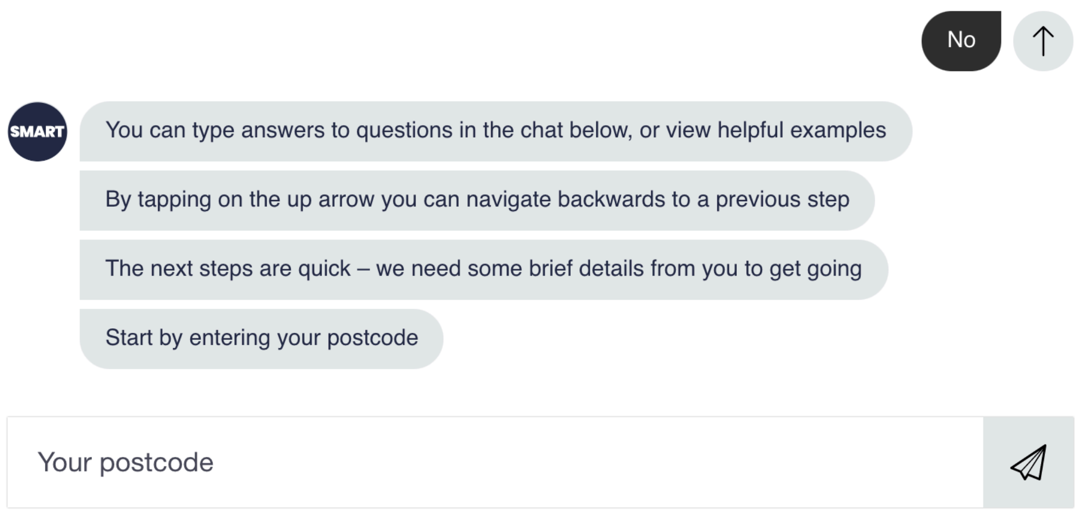
The service is compatible with most smart meters. You will need to have your Energy Display in hand

Would you like to know more about smart meters?

No

Stepping through the dialogue the user is asked to input a postcode which is validated against a web service call. Information about the postcode is returned to give confidence that the system has understood the postcode.

Figure 2: A prompt to enter a valid postcode, an error message is presented if the postcode is not recognised



No

You can type answers to questions in the chat below, or view helpful examples

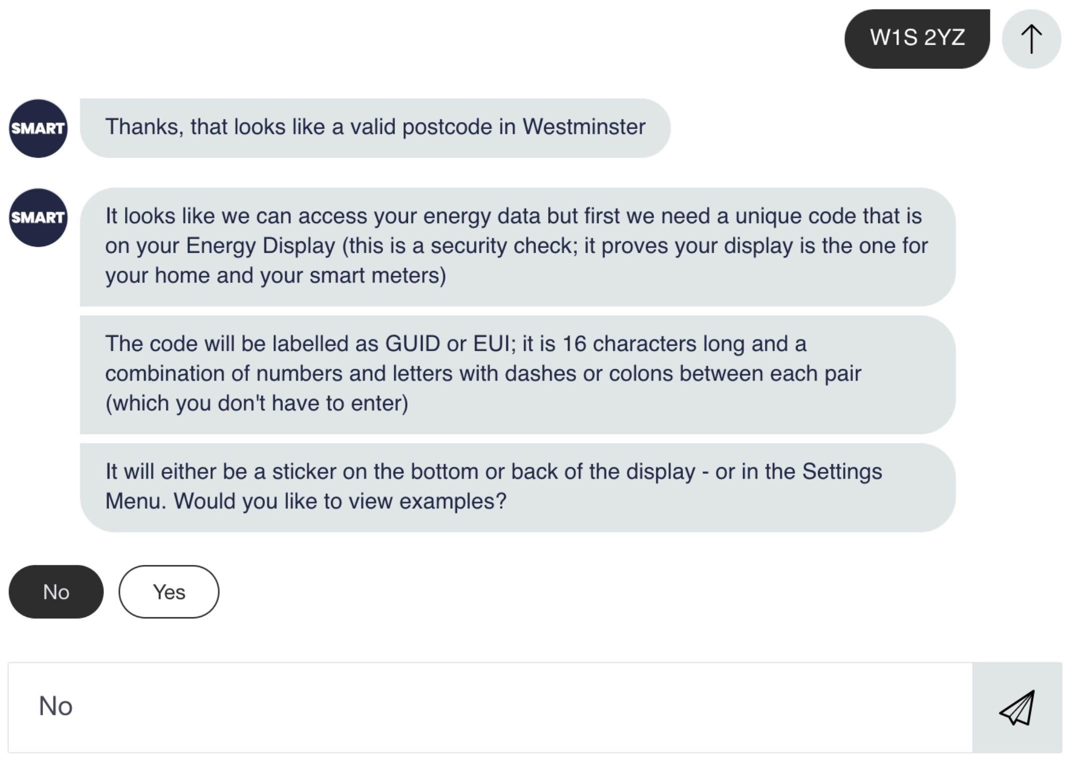
By tapping on the up arrow you can navigate backwards to a previous step

The next steps are quick – we need some brief details from you to get going

Start by entering your postcode

Your postcode

Figure 3: A prompt is shown to explain the identifier from the In Home Display (IHD) that is required to be entered. Through research we found that this was needed with example identifiers shown on devices.



W1S 2YZ

Thanks, that looks like a valid postcode in Westminster

It looks like we can access your energy data but first we need a unique code that is on your Energy Display (this is a security check; it proves your display is the one for your home and your smart meters)

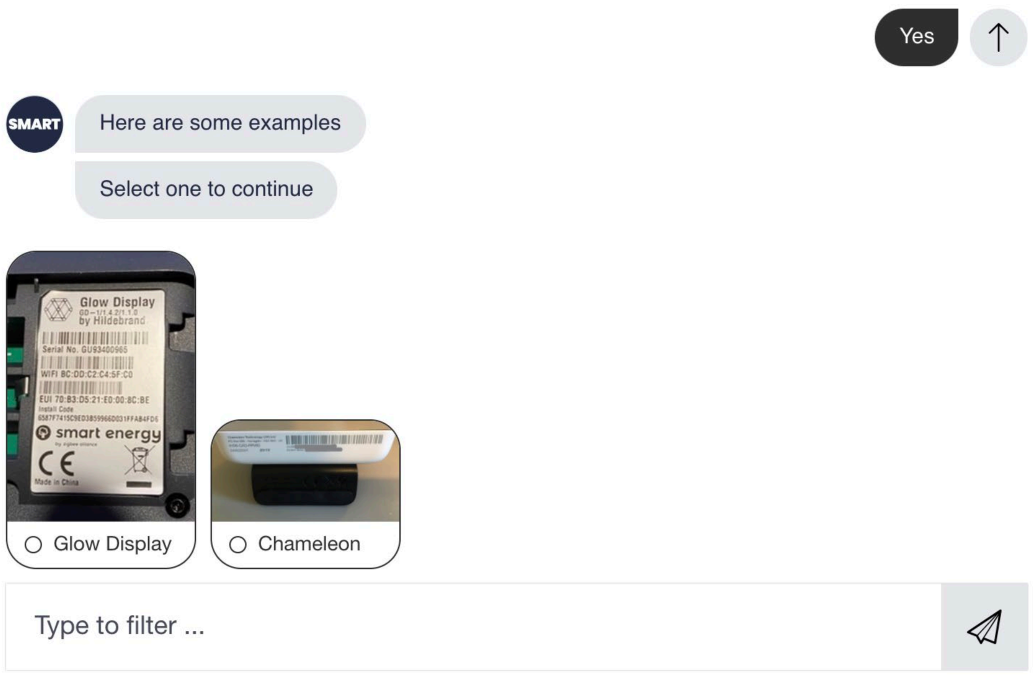
The code will be labelled as GUID or EUI; it is 16 characters long and a combination of numbers and letters with dashes or colons between each pair (which you don't have to enter)

It will either be a sticker on the bottom or back of the display - or in the Settings Menu. Would you like to view examples?

[✓ No] Yes

No

Figure 4: The EUI is entered and validated against the postcode. A web service is used to compare the match between these two pieces of information. At this point we could return some technical details about the meter type, e.g. if it is an electric and gas installation. We opted to just confirm that it was possible to proceed.



Yes

Here are some examples

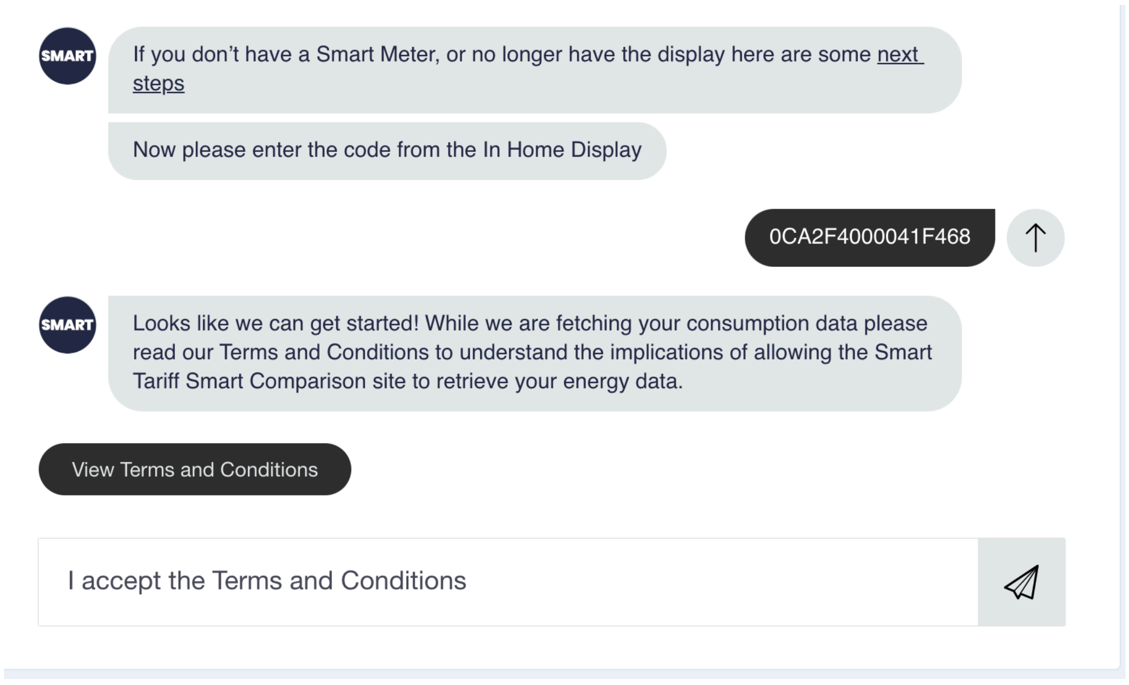
Select one to continue

○ Glow Display

○ Chameleon

Type to filter ...

Figure 5: Terms & Conditions are presented confirming that the person who has access to the two key pieces of information 1) postcode and 2) EUI of the IHD, grants permission to the data. There is a pop-up with all of the details when 'View Terms & Conditions' is selected.



If you don't have a Smart Meter, or no longer have the display here are some next steps

Now please enter the code from the In Home Display

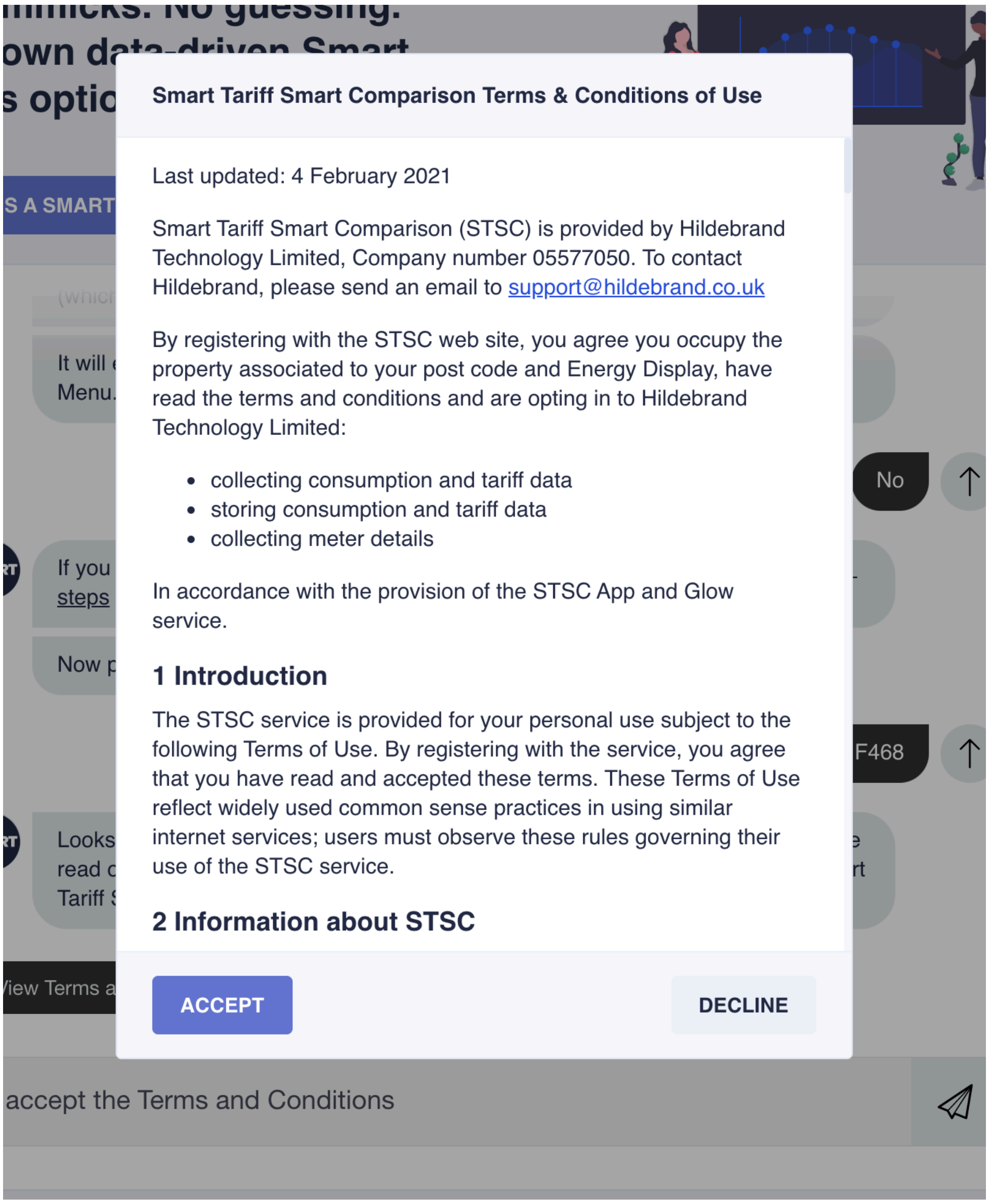
0CA2F4000041F468

Looks like we can get started! While we are fetching your consumption data please read our Terms and Conditions to understand the implications of allowing the Smart Tariff Smart Comparison site to retrieve your energy data.

View Terms and Conditions

I accept the Terms and Conditions

Figure 6: Pop-up of the detailed Terms & Conditions



Smart Tariff Smart Comparison Terms & Conditions of Use

Last updated: 4 February 2021

Smart Tariff Smart Comparison (STSC) is provided by Hildebrand Technology Limited, Company number 05577050. To contact Hildebrand, please send an email to support@hildebrand.co.uk

By registering with the STSC web site, you agree you occupy the property associated to your post code and Energy Display, have read the terms and conditions and are opting in to Hildebrand Technology Limited:

- collecting consumption and tariff data
- storing consumption and tariff data
- collecting meter details

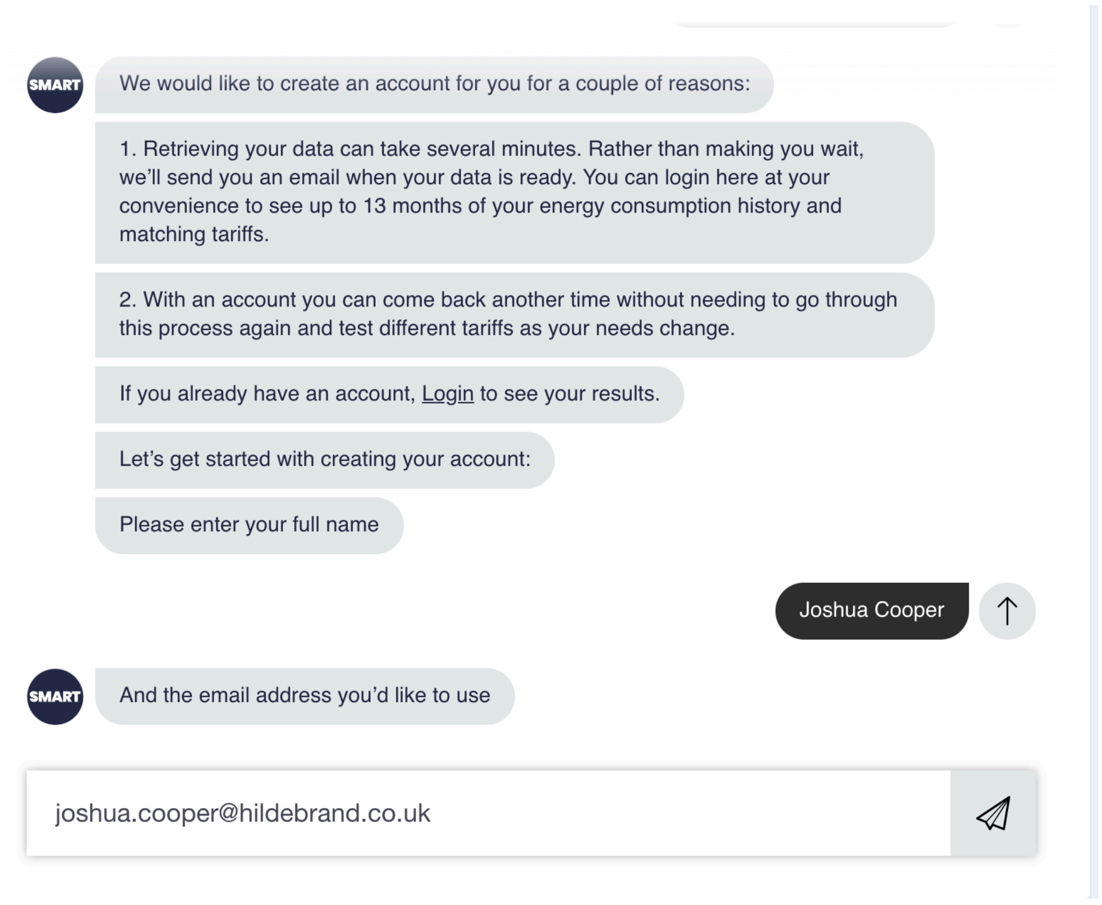
In accordance with the provision of the STSC App and Glow service.

1 Introduction

The STSC service is provided for your personal use subject to the following Terms of Use. By registering with the service, you agree that you have read and accepted these terms. These Terms of Use reflect widely used common sense practices in using similar internet services; users must observe these rules governing their use of the STSC service.

2 Information about STSC

Figure 7: An explanation of why we would like to collect a username and set up an account that can be used to log back in for results.



We would like to create an account for you for a couple of reasons:

1. Retrieving your data can take several minutes. Rather than making you wait, we'll send you an email when your data is ready. You can login here at your convenience to see up to 13 months of your energy consumption history and matching tariffs.

2. With an account you can come back another time without needing to go through this process again and test different tariffs as your needs change.

If you already have an account, Login to see your results.

Let's get started with creating your account:

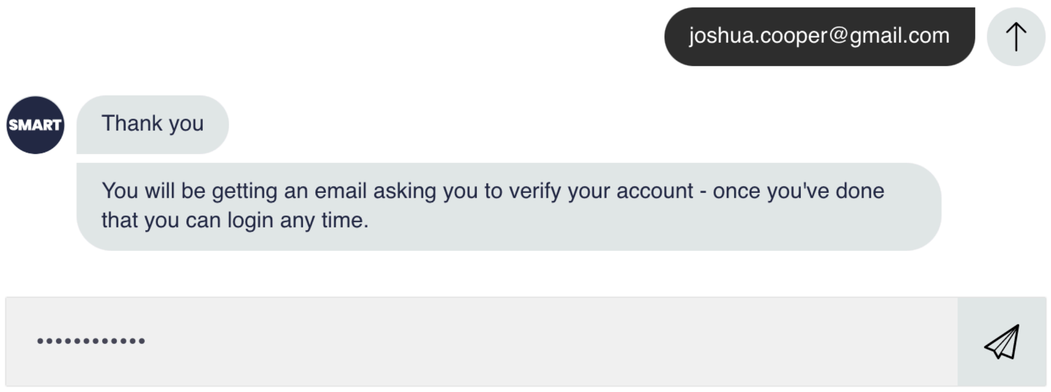
Please enter your full name

Joshua Cooper

And the email address you'd like to use

joshua.cooper@hildebrand.co.uk

Figure 8: Once the password is entered, passing a strength checker, there is a web service call to register the account with the meter technical details. This triggers the send of an email that has a verification link.



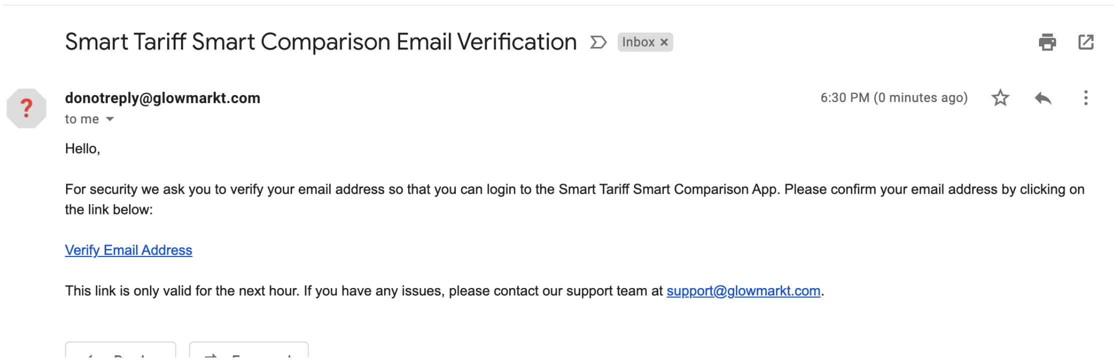
joshua.cooper@gmail.com

Thank you

You will be getting an email asking you to verify your account - once you've done that you can login any time.

••••••••••••

Figure 9: Verification email, sent to the user's email account. It is time limited.



Smart Tariff Smart Comparison Email Verification

donotreply@glowmarkt.com
to me

6:30 PM (0 minutes ago)

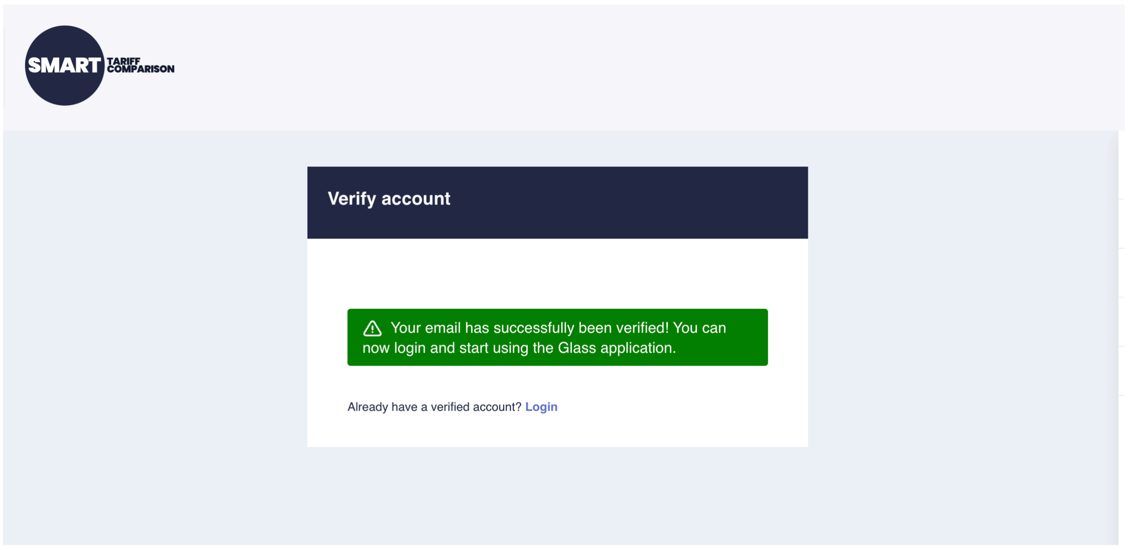
Hello,

For security we ask you to verify your email address so that you can login to the Smart Tariff Smart Comparison App. Please confirm your email address by clicking on the link below:

Verify Email Address

This link is only valid for the next hour. If you have any issues, please contact our support team at support@glowmarkt.com.

Figure 10: Verification of the email account being successful. If the time had expired, the user would be offered the option to resend a link.



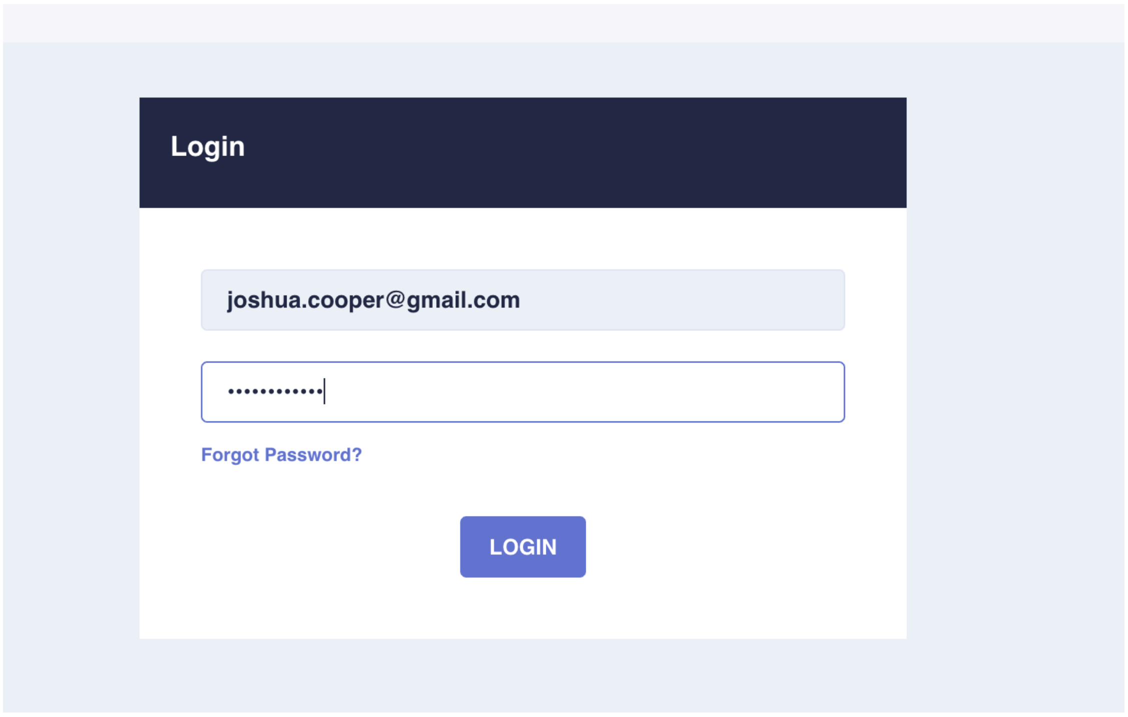
Verify account

Your email has successfully been verified! You can now login and start using the Glass application.

Already have a verified account? Login

At this point in the process the Glow platform is acting as the data controller for the data subject. The user could email from the address and request to be deleted from the system, use the login details to authenticate and get meter data.

Figure 11: Login dialogue box that will authenticate and return the authentication tokens required to access consumption data.



A login is presented. User credentials that have been created in this registration can be used to login or if the user is already registered with Hildebrand's Bright mobile app, existing credentials will grant access. We have made Hildebrand Bright directory services to be able to login to STSC so that existing users did not have to re-enter registration information.

For information, Hildebrand Bright users go through a similar onboarding journey with EUI and MPAN as part of their verification journey within the Bright App. There is also the option in Bright to verify using a photo id with an energy bill showing their address and MPAN; this is known as "manual" verification is a human operator checks the information and onboards the customer. This would be the fall back plan for users that have lost, or do not have access to, their in-home display EUI[1].

Rationale

Following are a few side notes on different topics to explain the underlying rationale.

Data access

A requirement is to make registration easy for the consumer and take away as much friction as possible while balancing the need to prove that the consumer has the right to request data for a particular meter. The rights are granted to the occupier of a property that has an electricity or gas meter that is smart. For example, a home owner living in a property, a renter of a property or even a short term let.

[1] Not all displays include an EUI; it is the installing supplier who specifies the IHD.

It is helpful to also talk about who might not have access to data. An owner of a property or bill payer does not necessarily have the right to see 30-minute smart meter data, as they could use that data to effectively spy on a tenant. The principle is to protect the privacy and security of the person or persons who are living in a property.

If the person that granted permission moves out of a property, known as change of tenancy or COT in the industry, then data must stop being accessible to that person and to any systems that were granted permission by that person.

The COT process is sometimes confused with change of supplier or COS. The smart meter data is still accessible to the consumer when they change suppliers. In fact it is extremely desirable from a consumer protection standpoint that the energy management and monitoring services continue regardless of the supplier. It is only in the case where the consumer is no longer using a property that the data needs to stop.

It is important to understand that the outgoing supplier must stop (and, by the way, there are technical features that enforce this within the DCC) accessing data once they lose a customer unless they happen to be running an energy management service beyond the supply of energy and have permission to continue accessing the data.

Choice of IHD EUI

The in-home display (IHD) is given to consumers as a part of their smart meter install. This is one of the few physical interactions with the consumer. The assumption is the consumer is physically in control of the IHD and that it is securely inside the occupied property. The EUI found on the IHD is unique and was registered, along with the electricity and/or gas meter, in the DCC inventory database against the supply point and therefore the postcode.

Using the meter serial number, meter EUI or any identifier found on the meter itself, was deemed by SECAS[2] to be less secure as some meters are found in common parts of buildings. The meter itself being in dark cupboards or locked metering rooms is usually less accessible to the consumer than their IHD.

Similarly, the meter supply point, known as the MPAN or MPRN is something known to the wider world and therefore does not ensure, on its own, that it is the occupier granting permission.

And finally, by definition, whoever has the IHD in their hand already has access to the smart meter data as it is displayed on the screen.

In order to ensure that an IHD is still in the property it was installed in, the postcode provides that extra verification that the IHD has not been moved into a different property. (IHDs cannot be interchanged easily from one meter set and communications hub to another).

[2] Smart Energy Code Administrator – web site: https://smartenergycodecompany.co.uk/

DCC Data Access

The DCC uses mobile technology to securely communicate with meters. There is a communications hub or comms hub that contains a modem and SIM card providing a private network back to the servers and systems operated by the DCC.

Data is not stored within the DCC. It is stored on the meters themselves or sometimes mirrored on the comms hub, therefore still within the property. Retrieval of data from the DCC takes some time and relies on the DCC's systems being available to make a request for that meter data and a reply, decryption and presentation of the data back to the end consumer.

Typically this is quick when all things are working properly. Our estimation is that within 5-10 minutes most people will have their data retrieved (up to one year's worth) and it will be accessible to servers and the applications in use. Things like tariff information may take longer and there are some specific meters and configurations that cannot be read.

Today's errors and issues are continually being found and fixed. At the time of writing we want to stress that perfection has not been achieved, but in most cases access to data is possible and once received the quality of data is very high.

Glow Platform

The Glow Platform is a commercial service provided by Hildebrand Technology Limited. It is free to use for individual consumers. Hildebrand is an "Other User" of the DCC, meaning not an energy supplier, meter installer or distribution network operator.

Hildebrand facilitates consumers sharing their data with other companies, service providers, devices, etc. as long as that service provider entity has registered with Hildebrand. This is done by signing up as a Glow Developer. You do not have to become a DCC Other User, effectively as a Developer you are able to access data as either a Data Processor or joint Data Controller with Hildebrand providing the role of Data Controller for the consumer.

You can use all of the examples and code and not use the Glow Platform, however you will have to become a DCC Other User yourself and the only available support will be from the open source community.

Second Stage - Comparison Dashboard

Once logged in, the system makes several fetches of server data to populate a tariff comparison dashboard. Each area has a loading state that will show until data has arrived at the browser.

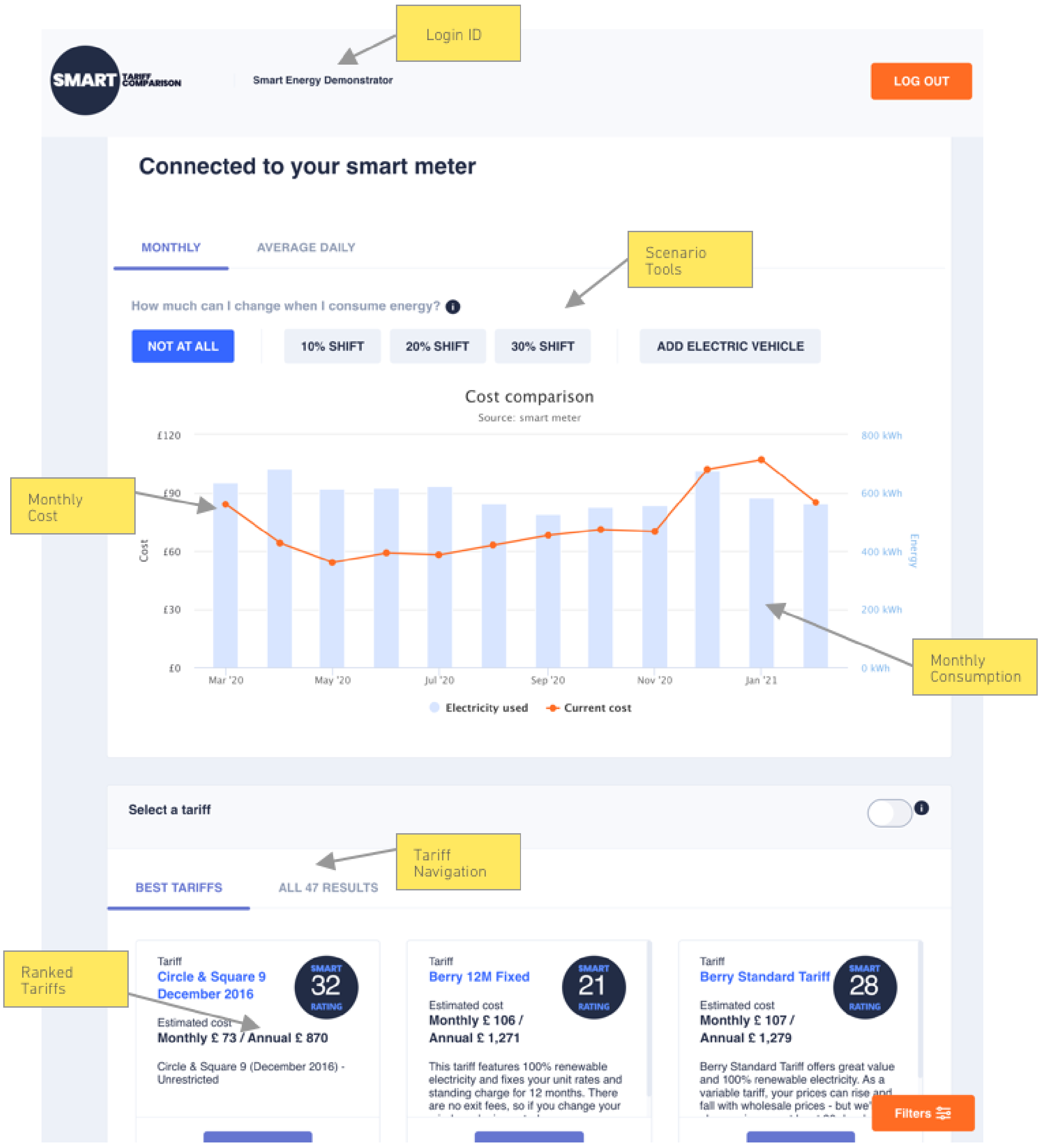
Fetch 1: The first is to populate a monthly summary graph of electricity consumption and cost. This data loads fairly quickly with server side aggregation calculations taking the time for the load. If a user has a dynamic tariff (Octopus' Agile, for example) then load time for the cost line is higher as cost is calculated for each 30 minute period then aggregated. The source of this data is the Glow service, therefore if there were any historical tariff changes while the user was registered, then the different tariffs for the historical time period are captured. If the user is new, then it is only the current tariff rate that is accessed through the meter that is used to calculate the cost line.

Fetch 2: Supplier, product and tariff objects are loaded in parallel. They are fairly light calls to the server and return quickly. The product and tariff server calls take the grid service point (GSP) of the electricity meter into account to return a filtered list that applies to that GSP code. For instance, there are 47 tariffs displayed for a single GSP whereas the system has 47 products with 658 tariffs in the database. From the user view the higher level product is not directly shown, from a technical view the product information is an efficiency as well as a data structure that can hold higher level information like exit fee costs, minimum contract term and other marketing level information that is useful for filtering. On a technical note the fetch of this information is done through the STSC Javascript library.

Fetch 3: A third type of call is also being made in the background to fetch the 30 minute consumption data. This takes several seconds to load for the first login, but returns as a binary file, appearing to the browser in the same way that an image is loaded. The binary file is then processed and loaded into the stsc-tsc library, or STSCJavascript/Typescript library. It is at this point the library can perform calculations and is available to the user interface javascript to be displayed.

Some of the first calculations performed on the 30 minute consumption data within stsc-tsc are monthly and annual cost of each of the tariffs, an average daily breakdown of consumption and the smart rating. The tariffs are then sorted by the stsc-tsc library in rank order of cost.

Figure 12: Overview of the comparison dashboard for Proof of Concept 3. We have made buttons and areas overly large to test the response to different features with the user segments.



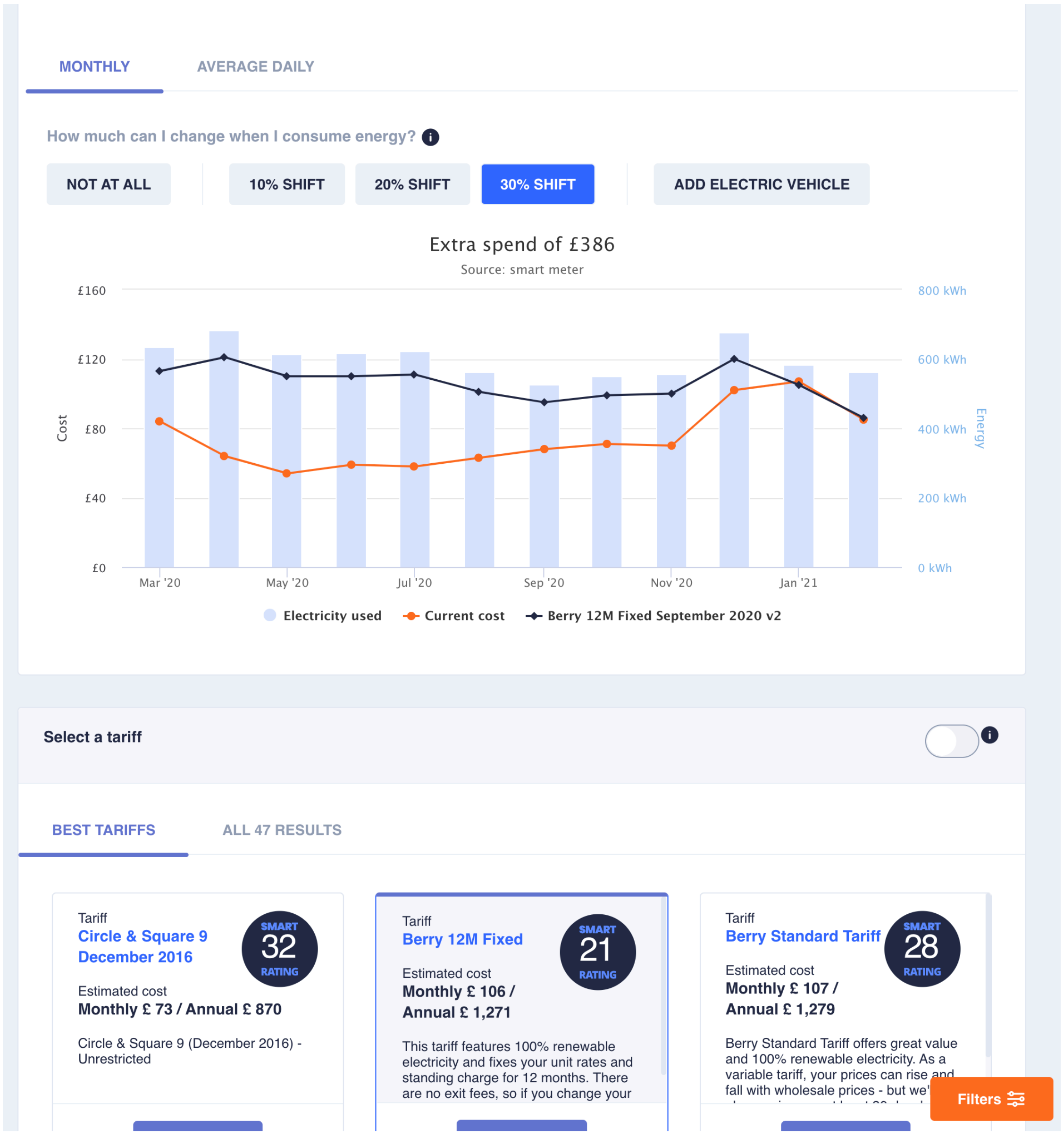
Tariff
Circle & Square 9 December 2016
SMART 32 RATING
Estimated cost
Monthly £ 73 / Annual £ 870
Circle & Square 9 (December 2016) - Unrestricted

Tariff
Berry 12M Fixed
SMART 21 RATING
Estimated cost
Monthly £ 106 / Annual £ 1,271
This tariff features 100% renewable electricity and fixes your unit rates and standing charge for 12 months. There are no exit fees, so if you change your

Tariff
Berry Standard Tariff
SMART 28 RATING
Estimated cost
Monthly £ 107 / Annual £ 1,279
Berry Standard Tariff offers great value and 100% renewable electricity. As a variable tariff, your prices can rise and fall with wholesale prices - but we'

At this point the data is loaded and personalised calculations have been made. By clicking on a tariff the monthly summary of cost for that tariff is overlaid on to the graph with an annual net gain or loss (more or less expensive relative to your current tariff) is shown. The calculations are performed automatically by the library locally on the browser, making the user experience quick and efficient.

Figure 13: Cost for each tariff is calculated and presented on the main graph next to the current cost.



Tariff
Circle & Square 9 December 2016
SMART 32 RATING
Estimated cost
Monthly £ 73 / Annual £ 870
Circle & Square 9 (December 2016) - Unrestricted

Tariff
Berry 12M Fixed
SMART 21 RATING
Estimated cost
Monthly £ 106 / Annual £ 1,271
This tariff features 100% renewable electricity and fixes your unit rates and standing charge for 12 months. There are no exit fees, so if you change your

Tariff
Berry Standard Tariff
SMART 28 RATING
Estimated cost
Monthly £ 107 / Annual £ 1,279
Berry Standard Tariff offers great value and 100% renewable electricity. As a variable tariff, your prices can rise and fall with wholesale prices - but we'

For the prototype a couple of sample interactions show the aspects of the stsc-tsc library's lower level functions. These are presented as scenario tools. First being a shift of peak load to overnight (resembling a battery) and the second being the additional load due to an electric vehicle. Both of these are best illustrated through the Average Daily tab, where clicking on the scenario will change the energy consumption for those periods, then going back to the Monthly graph to show the financial impact.

Figure 14: Scenarios change the energy profile at a 30-minute level for the entire year of data. The average daily view shows shifting a percentage of the peak load to the overnight hours with movement of the time series data.

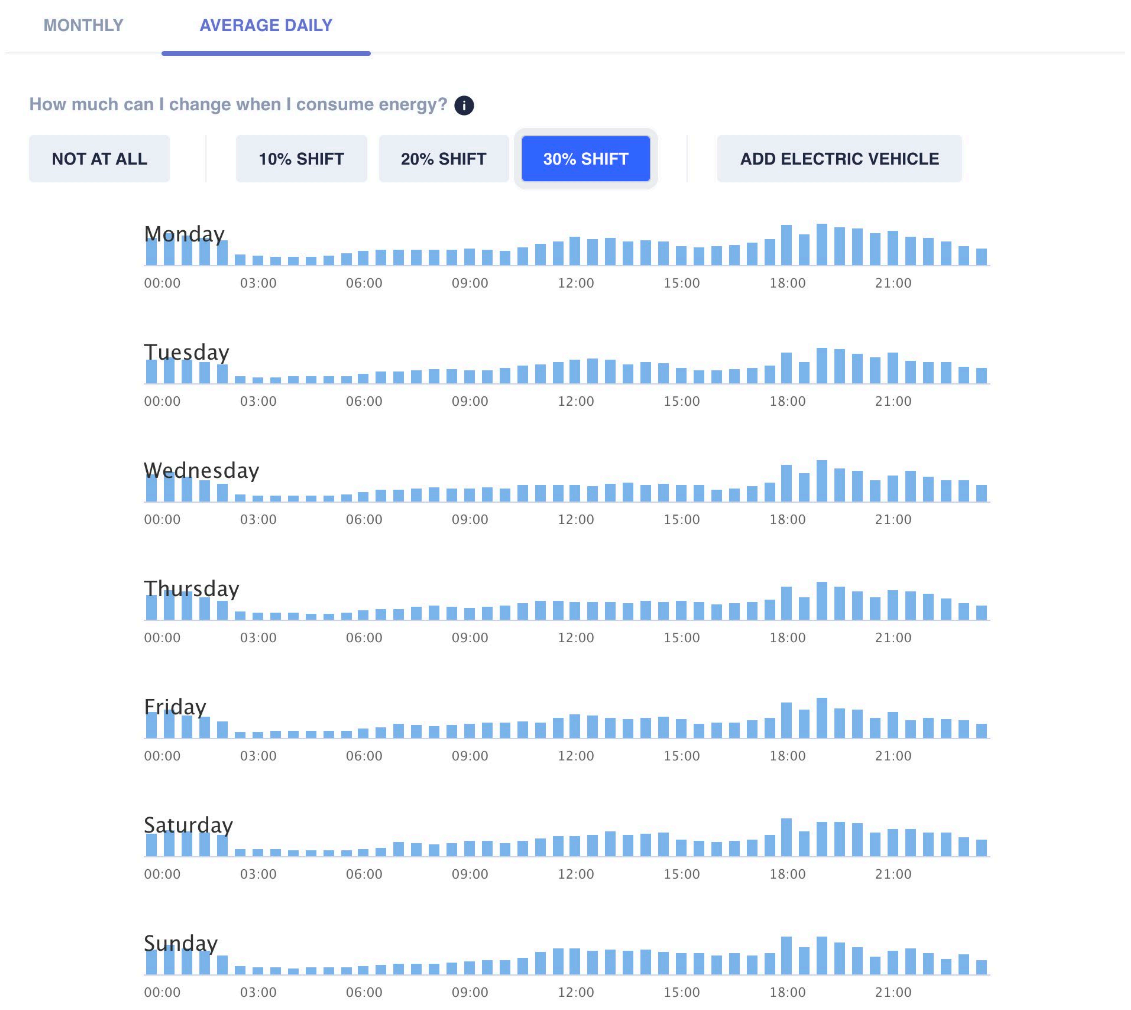
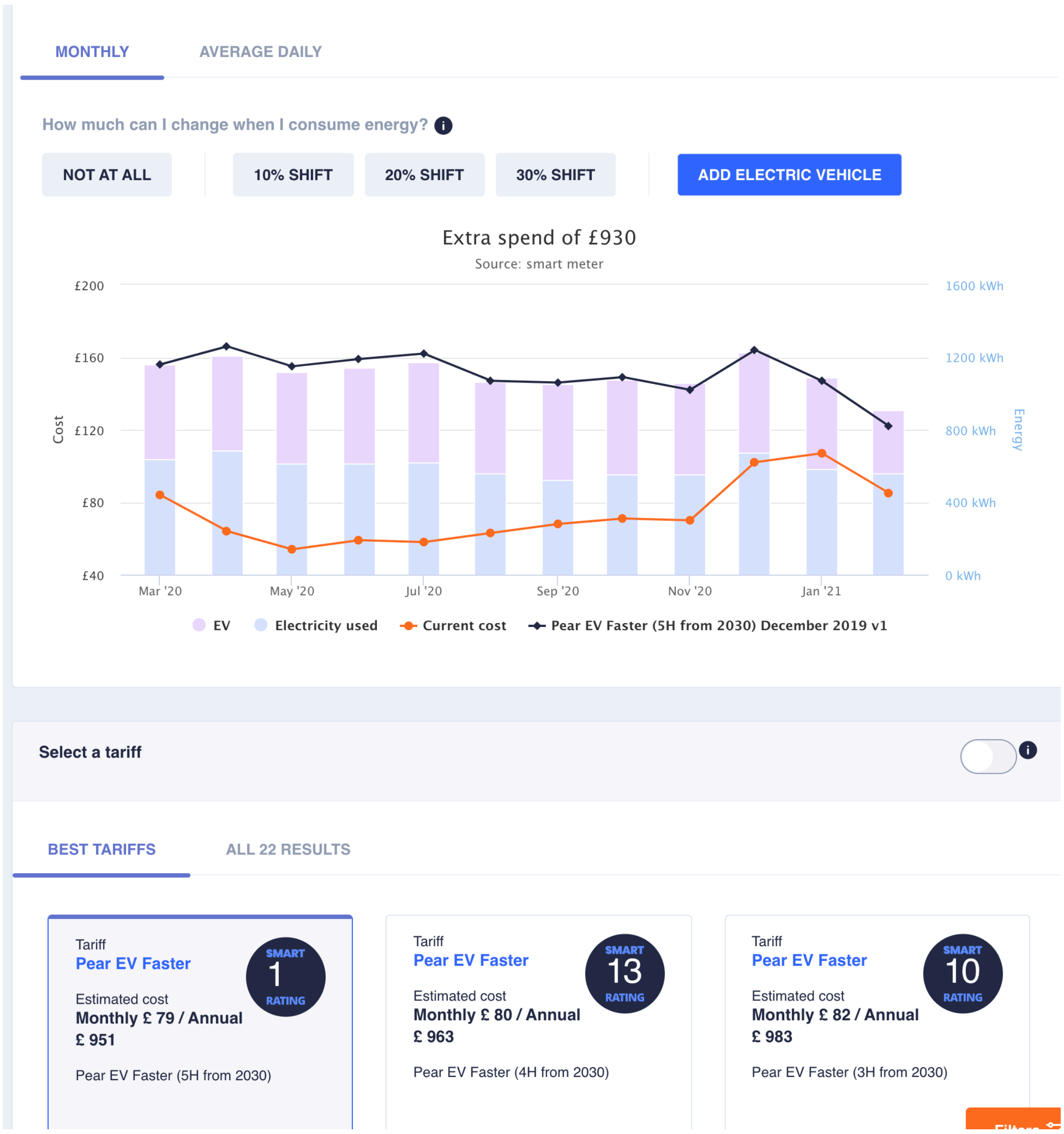


Figure 15: On a monthly and annual basis, the scenarios impact consumption and cost, and this illustration, EV increases consumption and makes overnight time of use tariffs more attractive.



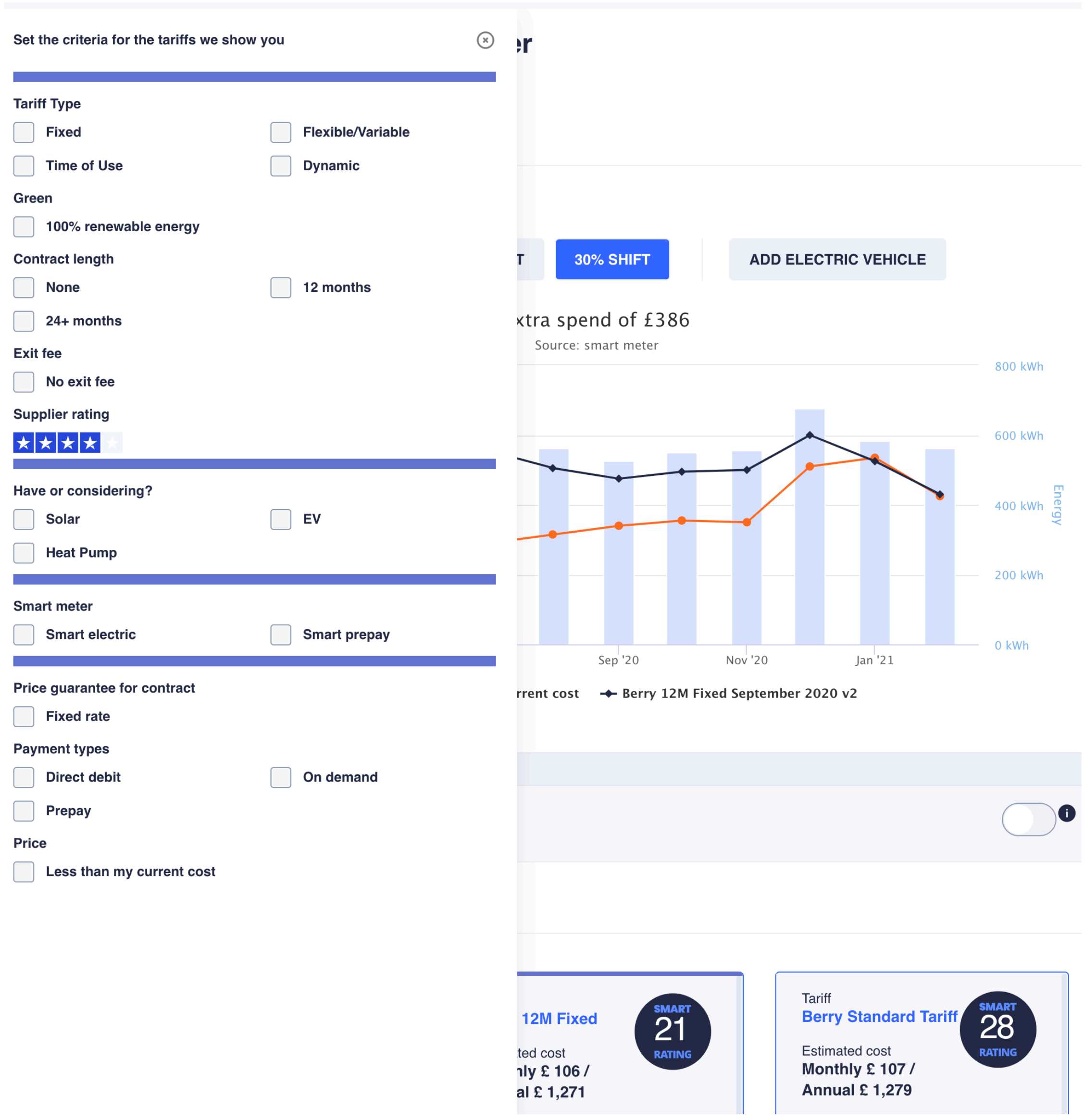
Tariff
Pear EV Faster
SMART 1 RATING
Estimated cost
Monthly £ 79 / Annual £ 951
Pear EV Faster (5H from 2030)

Tariff
Pear EV Faster
SMART 13 RATING
Estimated cost
Monthly £ 80 / Annual £ 963
Pear EV Faster (4H from 2030)

Tariff
Pear EV Faster
SMART 10 RATING
Estimated cost
Monthly £ 82 / Annual £ 983
Pear EV Faster (3H from 2030)

The user interface includes the ability to set product and supplier preferences that filter the tariffs based on features. The features change the list of tariffs presented, however they are still ordered by cost (lowest to highest). Likewise there is a table view of all of the tariff results that can be sorted by other criteria, clicking on a tariff row will then change the "predicted cost" and update the monthly graph.

Figure 16: Tariff and supplier filter criteria



Set the criteria for the tariffs we show you

Tariff Type

☐ Fixed ☐ Flexible/Variable

☐ Time of Use ☐ Dynamic

Green

☐ 100% renewable energy

Contract length

☐ None ☐ 12 months

☐ 24+ months

Exit fee

☐ No exit fee

Supplier rating

★★★★☆

Have or considering?

☐ Solar ☐ EV

☐ Heat Pump

Smart meter

☐ Smart electric ☐ Smart prepay

Price guarantee for contract

☐ Fixed rate

Payment types

☐ Direct debit ☐ On demand

☐ Prepay

Price

☐ Less than my current cost

The smart rating is an attempt to calculate a non-financial metric to prioritise tariffs on some other characteristic. By sliding the toggle in the "Select Tariff" header, the sort order is changed to the smart rating (best to worst).

Likewise sliders have been used as an experimental input mechanism for preferences to see if that was more appealing than a checkbox and filter functional pattern. By sliding the preferences there are filters that display smaller or different sets of tariff options.

Use Case and Scope

The overall goal of the STSC tool is to match the best tariff with a consumer's needs.

The consumer needs are based on volume and pattern of energy consumption, contractual needs/eligibility and increasingly additional energy product features such as smart control, bundled low carbon technology or advanced energy as a service concepts.

Smart tariffs and smart comparisons are challenging are for a number of reasons:

- Nature of the forward consumption and tariffs;
- Volume of data required;
- Complex structures due to new time dependencies;
- Scenarios that require capital;
- Complex optimisation due to reflexive choices;
- Perceived benefits/value.

Crystal ball challenge

Switching tariffs is based on the promise that future costs will be lower or product features will be better in the future as a result of adopting a new energy supplier or tariff. These future costs are based on future consumption volume and patterns as well as unpredictable future prices of energy.

Energy suppliers play a role in making the future price predictable, which is evident in the way single rate, long term contracts are structured and presented to customers. The energy supplier takes bumps out of the energy price road by aggregating demand, balancing a portfolio, purchasing its own future wholesale supply and assuming the financial risk for predictions. It is a legitimate argument that the "smartest" tariff is the one that figures it out for consumers as a single time invariant rate. There are even economic predictions that say flat rates on the wholesale market will be achieved with large scale battery storage and control.

Electricity usage can be difficult for some consumers to understand, particularly future consumption. It turns out that at an annualised level, most electricity usage is fairly stable and predictable, but this is certainly going to be challenged in the coming years with electrification solving our carbon emission issues. It is the adoption and integration of this new or radically changed demand that will present the largest challenges and opportunities for STSC. A new tariff must be chosen BEFORE the consumption has occurred; in effect we need to be able to predict new consumption patterns to gain the financial benefit.

This translates into a few requirements

- Use of historical data to inform the future consumption is very important, but should not necessarily be used verbatim or at the very least make sure the user understands how historical data is being used;
- Dynamics of the present and past consumption should be understood: has the consumer significantly changed consumption due to the addition of a low carbon technology somewhere in the past data stream;
- Ability to work with future estimates and get user input to inform future personalised energy system model;
- Tariffs that have longer term contracts can be applied as future rates, however there needs to be adequate provision for future price risks in dynamic tariffs.

Bigger is better ... sometimes

Dynamic tariffs, exported energy, self-generated energy, data streams associated with storage not to mention the basic imported energy increase the breadth of data sources that need to be considered in STSC. Resolution at 30-minute levels also becomes necessary, further increasing the volume of data used in describing the personalised energy system.

This data is used in two aspects best abstracted into accounting and modelling functions. Accounting is important as it is the final cost of the energy and forms the basis of the purchase transaction with the energy supplier or how you may view return on investment for low carbon technology. Modelling tries to make sense of the data to explain the why and how of energy use, it is useful for prediction and understanding what technology may benefit carbon savings.

Bigger data is great for modelling and a hassle for accounting. In fact, processing and administrating large accounting data sets is a real cost that increases the delivery cost of the energy. Large, granular data sources need an analytic approach to calculate the model features, typically these are statistical features like daily average and variance; with a large data set the options for analysis increase.

Under this topic, we have requirements, both functional and non-functional:

- Performance, particularly load time, needs to be considered with large data sets not to overload or delay results for the end user;
- We are relying on accounting style calculations when presenting cost information, however we need to be able to change model parameters that will then translate quickly into accounting terms - i.e. ranking tariffs based on annual cost;
- Metrics need to be easy to follow and relatable;
- Visual presentation of trends and changes in model data in particular need to be clear with cues and summaries of the changes, e.g. an additional spend of £300 per year showing the current in relation to the new spend;

- Precision should be reduced, but accuracy should still be high; don't overwhelm the user with many decimal places or small unmanageable units of time but make sure aggregated results day, month, year are accurate and correspond to comparable billing periods they would find on a printed bill;
- Bring out the relationships within the data in a modelling context and in real time show the accounting impact.

Time as a new dimension

In flat rate tariffs, which dominate the current energy market, a unit rate is sufficient information to compare tariffs whereas in smart tariffs time becomes a major factor in the overall cost. Low carbon technologies further challenge the need to use time as they have time based constraints such as when you want to charge an electric vehicle or when the sun is shining to produce solar energy. Some requirements include:

- Underlying software capability needs to support time alignment, resampling and aggregation of time series data;
- Relationships between energy sources, demands and prices should be able to be represented through operations between time series such as addition and multiplication and joining independent data sources in time, such as weather data;
- Visual representation of time is important with user input controls helping navigation in time, familiar labels and units should be used when interacting, such as "weekend", "weekday", "morning", "evening", etc. so as not to overwhelm users with time precision;
- Data quality is important, if there is missing data it should not invalidate overall results, models and methods should be sympathetic to missing or erroneous data points;
- Future times should be able to be represented for at least the purposes of prediction.

Free energy?

Smart tariffs are likely to complement investments in renewables and storage. When analysing a new tariff or low carbon technology the capital cost of solar panels or other distributed generation technology needs to be considered. The accounting for payback and operating costs is difficult historically and very difficult to project forward, however it doesn't mean that the unit rate of self-generation is zero or that demand would have occurred if there was a high price on the use. To avoid making renewables appear to be free the following requirements should be kept in mind:

- When displaying renewable energy and its impact on cost it should be framed as either a unit rate expressing the cost of plant, materials and operation for the unit of energy or allocated for the time period being considered e.g. annual cost to run solar panels;
- When analysing self-produced energy, make sure to allocate the avoided expense of grid energy at the market rate for that time period;

- When estimating potential for savings be careful not to use an inflated future rate, including issues with using average rates or double counting consumption and export as avoided expense and income;
- In an ideal analysis stored energy would remember its unit costs when the energy was "consumed" to charge a battery;
- Make sure to consider the efficiency and capacity of production and storage technologies, they don't perform at 100% and have upper and lower bounds of operation including time to charge, discharge;
- Procurement of energy may be outside of the billing current energy bill, for instance charging an EV at work, it should be clear if this is considered as free or as a cost.

Trade offs

Optimisation of the best tariff is complicated by market prices or costs that change based on demand levels. Within the scope of this project we are not going to directly consider these effects. We will assume that the energy market works externally and there is sufficient volume in the market that individual actions will have a meaningful impact. In the future when scheduled loads are present, this may be a large factor.

True value

Understanding the value proposition beyond cost is partly subjective and for some consumers they want to be able to put a value on the benefit of an energy product. This may go well beyond loyalty or brand preference - for instance referral fees.

Our approach has been to present the calculated cost of a particular product and let the consumer discount these benefits themselves.

Design of JavaScript Library

The code is implemented as a mix of pure JavaScript with TypeScript extensions.

STSC JavaScript Library

The smart tariff, smart comparison (STSC) project uses smart meter data along with tariffs to calculate cost and transform data to be used in presentation to user interfaces. The objective is to help consumers select the best tariff, track a current tariff forward and understand how in changing their energy consumption may effect the cost of energy.

This library provides JavaScript and TypeScript support for data access, calculations and general framework elements. It is intended to be used in browser and node.js environments - although the current demonstration of the library is through an Angular web application executing stsc-tsc code in the browser. This means the testing and structure are more mature for use in the browser, however both environments should run identically.

Example code and use as a stand alone library is available on ObservableHQ @joshuacooper. https://observablehq.com/collection/@joshuacooper/smart-tariff-smart-comparison There are several tutorials on how to use the library in the collections found there.

A simple server-side example to demonstrate how the library can be used on a server is on Runkit https://runkit.com/ijosh/stsc-energy-example/2.0.2 This is not meant to be exhaustive as through the front end code we are trying to demonstrate the use of the library.

As a general direction of travel, smart meter data access is through Glowmarkt, a service platform run by Hildebrand. It is not necessary to use this data access route to use the library, however you may need to put more effort in the registration and management of your users, as well as a number of the supporting data elements like GSP codes.

Detailed Design Documents

More detailed documentation, including diagrams for object interaction are found in the stsc-tsc source code release. They are in mark down, although can be viewed online through the Bitbucket code browser.

README.md – file with up to date release notes and introductory text

ModelDesign.md – file explaining object interactions and considerations

Timezone Considerations.md – how timezone including daylight savings is addressed within the code as well as wider smart meter implications

README_Coding.md – coding standards and instructions to follow to join the open source contributions.

Release POC4

The library works with a fourth proof of concept (POC4), supporting the data models for presentation within a larger STSC front end website. The JavaScript libraries could be used to embed calculations within your own website and offer some insight into how the backend data is structured.

For this release the use cases are around using half hourly smart meter data from the DCC and a number of hypothetical tariffs taken from public sources or from energy suppliers. The source of data is Hildebrand's Glowmarkt data service, providing a binary file of consumption data.

Tariffs in scope are of three types:

- Flat rate tariffs - these are typical single rate tariffs.
- Time of Use (TOU) tariffs - rates (up to 48 per day) that change throughout the day, but are the same for every day. This mirrors many of the ECO7, ECO10 rates as well as rates targeted at electric vehicle owners. Natural progression would be for different day of the week rates until they begin to look more like truly dynamic rates.
- Dynamic - each half hour has a specific rate that changes, the rate may only be known a day or two ahead, therefore there is little predictability in the price.

Each of these usually has an additional standing charge expressed in pence per day that is locked in for a longer contract period.

Calculating cost for various time periods is a key output of the library. To do this, it is a relatively simple application of a tariff structure to energy consumption data. This can be used for a number of purposes:

- Optimisation (at least historically) of tariff based on lowest cost; a brute force calculation is fairly quick within a browser however becomes more challenging as the number and complexity of the tariff (see calculation complexity for tariff types below);
- Tracking/monitoring - possibly save off the results and then come back to track if the savings happened in reality;
- Scenarios - change the energy consumption and see what happens to the costs.

if you would like to dive in and have a look at some examples in code, the ObservableHQ notebook https://observablehq.com/@joshuacooper/tariff-calculations works through some examples of the three different tariff structures.

Compute complexity and memory analysis

The implementation of the library is interesting to discuss. Tariff types are sets that will use the same energy input to calculate cost. For instance, there may be 10 flat rate tariffs, 20 TOU and 30 dynamic tariffs to evaluate in order to choose the best tariff for the consumer. If the energy profile is changing, say "I would like to consider what a smart EV charger may do to my cost given the same set of tariffs to evaluate," a fast calculation becomes desirable if not necessary. Very complex and interacting scenarios challenge this further.

Although the scenarios themselves may not be fully coded in the library, the structure of the library is considerate of the computational challenges and it is a goal to accommodate software like this on top of this library.

Calculating a flat rate tariff usually involves summarising data for a time period, such as one year or for 12 calendar months. The summary operation is an O(n) computation with n probably maximal for 13 months (396 days * 48 values = 19,008). Once the summary is complete, then new tariff calculations are O(1). For 10 flat rates, the complexity is 9 x O(1) + O(n).

TOU is fairly similar in complexity with a first pass of the data to put it into time buckets and then rates applied. In order to be as flexible as possible, we've opted for summarising the time period of interest into 48 half hourly buckets. For example, a year's worth of data is summed within its half hourly slot. This allows for simple 48 x O(1) calculations for any TOU tariff.

Dynamic is more challenging in that every half hour has a different rate. There are not really any short cuts for calculation. Each tariff to be evaluated has an O(n) computational cost. The main approach has been to calculate all of the results at once for each tariff half hourly slot; this saves iterating for each tariff.

STSC

At the highest application level the STSC class provides a log in mechanism to the Glow platform service and retrieval of a token that can be used for further calls on classes. Tokens are Java Web Tokens or JWTs as a structure.

Timeseries

This is a base class that is used for storing and aggregating time series data. It assumes "perfect" data as higher level classes will know how to fill in the blanks. There are utility functions that calculate statistics within the time buckets so that when the aggregate function is called it knows how to reduce the data.

The timeseries class is used as a base class for many of the STSC classes as they are time series data. Even within the timeseries, when range is called, it returns a Timeseries, so then the timeseries functions are all available on a new instance of the data rather than mutating the underlying raw data that may be relied on by other parts of an application.

Full documentation of Timeseries will be done later as it is more important to see how the Timeseries is used in the higher level objects, therefore the concept is introduced here.

Loader functions for large timeseries, specifically Energy and Dynamic Tariffs are implemented in their respective classes as the data types and validation assumptions are specific to that higher level semantic.

At the moment to keep the API understandable and testable, each aggregate function is called atomically. This is a known inefficiency in the structure of the code, albeit very minor. A guture implementation may actually call a number of the aggregation functions as the main work is the loop, not the calculation. Presenting the aggregates back (sum, count, mean, variance) as an object is more efficient in CPU cycles only if those statistics are actually required together as a group.

Energy

Energy is a Timeseries that is populated from a binary object sourced from Hildebrand's Glow service. The binary is used because of the size of JSON that would be required, plus benefits in being portable and helping the backend systems service fewer requests.

The Energy object takes has a token property that is passed through to a server request. For testing and simplicity a resource_test token can be used that will return a known, but real smart meter data set. This populates data that methods can then act on. A simple example:

```
// require through local npm install - make sure your project has npm i stsc-tsc in package.json

var mystsc = require("stsc-tsc");

let energy = new mystsc.Energy();

energy._token = 'resource_test';

await energy.load();

// time stamped values, note the default aggregation is being used.

console.log(energy.byMonth());

console.log(energy.max());
```

The async call to load is to make sure the data has been loaded from the server request before any calculations are attempted. The other timeseries and energy methods are then available.

For example, max, min and mean are all against the complete set of energy data that has been loaded.

As an example (following on using the energy object made above), the consumption method helps extract and aggregate for displaying into a graph.

```
let  from = '2020-05-01T00:00:00+00:00';

let  to = '2020-06-01T00:00:00+00:00';

let newTS = energy.consumption({

   from:  from,

   to:  to,

   interval:  'PT30M',

   aggregateFunction:  mystsc.Timeseries.AGG_SUM,

});

console.log(newTS.export());
```

From and to are used to pass in start and end dates for processing. They are ISO formatted strings with timezone information. When the API uses start and end semantics those are expressed in milliseconds since epoch. Slot numbers if referring to half hours are 0 - 47 with 0 equal to midnight, the beginning of the day. Days are indicated by a daySlot, 0-6 with Monday as zero.

You can set different intervals to aggregate with using the ISO standards. In reality, use the PT not the P type ISO durations as localtime is used with P designators, whereas PT are relative to the initial timestamp and in UTC (this is Luxon not handling the P intervals correctly). Also make sure to keep the timezone information on the ISO date passed in to the object as you will get strange behaviour with local day light savings without it.

Aggregation functions are defined in the Timeseries base class and are referred to through the instance name of the library, the Timeseries class and then the aggregate function macro name e.g. mystsc.Timeseries.AGG_SUM.

Build

The build of the project is using Typescript and ES5 syntax in coding as classes. The package.json file has a couple of scripts, make sure to run npm install so they are available.

The index.ts exports the classes, with transpiling to index.js for ES5 with the following commands:

```
git clone https://ijosh@bitbucket.org/LDNSFO/stsc-tsc.git

cd stsc-tsc

npm install
npm run build
```

This should build source code into a distribution folder, lib, that can be included in your projects or used in a bundled single file. For bundling we are using an external service, https://bundle.run/stsc-tsc@1.0.9 for instance. This uses rollup and browserify at the heart of its implementation. Just pass the relavent version number into Bundlerun to get a neat version for use.

Jest tests

Jest is used as the test framework with tests found by class in src/__tests__/

Each of the classes has regression tests associated with it. They are run as a part of the release scripts in npm. If tests fail, then a release can not be made.

To run the jest tests use the script in npm (look in package.json for the detailed command) by running:

```
npm run test
```

Transpile

The project uses the tsc transpiler to Javascript. The typescript configuration is basic, but targets a module style that can be used across a browser and server environment.

```
{
  "compilerOptions": {
    "allowJs": true,
    "target": "es2015",
    "module": "commonjs",
    "moduleResolution" : "node",
    "declaration": true,
    "outDir": "./lib",
    "strict": true
  },
  "include": ["src"],
  "exclude": ["node_modules", "**/__tests__/*"]

}
```

NPM availability

Code is released for use in npm. Semantic versioning is used with POC4 being 1.0.x and patched updates replacing the x.

https://www.npmjs.com/package/stsc-tsc

Publication

This is published as a part of the BEIS sponsored Smarter Tariff Smarter Comparison programme.

Copyright

Glow Service

Please contact Hildebrand to gain access to smart meter data on a B2B basis. As a consumer you should be able to use the public data service to get access to your data.

Glow Forums

If there are particularly interesting discussion points, etc. please join the forum so we can keep questions and answers documented for everyone.

https://forum.glowmarkt.com/

Release notes of the Angular Front-end

StscFront

This project was generated with Angular CLI version 9.0.5.

Development server

Run ng serve for a dev server. Navigate to http://localhost:4200/. The app will automatically reload if you change any of the source files.

Code scaffolding

Run ng generate component component-name to generate a new component. You can also use ng generate directive|pipe|service|class|guard|interface|enum|module.

Build

Run ng build to build the project. The build artifacts will be stored in the dist/ directory. Use the --prod flag for a production build.

Running unit tests

Run ng test to execute the unit tests via Karma.

Running end-to-end tests

Run ng e2e to execute the end-to-end tests via Protractor.

Further help

To get more help on the Angular CLI use ng help or go check out the Angular CLI README.

Source availability

Will be working on making all available via Bitbucket at stsc-front.

Publication

This is published as a part of the BEIS sponsored STSC programme.

Copyright

Binary File Types

We use a binary file ending with a extension stsc to encode and transmit the timeseries data relating to 30 minute consumption readings, dynamic tariffs and other 30 minute data such as carbon intensity. This is equivalent to downloading a jpeg or image data with decoding being done on the browser.

This file type will be progressed with documentation being updated, to include a full dictionary of file type magic numbers, encryption and signed files. This helps to overcome some of the risks with data security, but also helps with the traceability of the provenance of the data, tariffs and other data that may form a part of the commercial considerations for energy suppliers, switching services and the like.

Further extensions of the file formats can be used for estimators, new profiles, exported energy, solar generation and other numeric timeseries.

Consumption

A 2-byte, unsigned integer value. The header identifier is 0x0F with the interval defined in the next byte and four more header bytes representing the start time of the timeseries. The length of the file then corresponds to the number of entries representing a 30-minute interval offset. For instance, from and including byte 8 of the file.

Tariff

The tariff is a 4 byte signed value. The header identifier is 0x0A. The length of the file then corresponds to the number of entries representing a 30-minute interval offset. For instance, from and including byte 8 of the file.

Carbon Intensity

The tariff is a 4 byte signed value. The header identifier is 0x0B. The length of the file then corresponds to the number of entries representing a 30-minute interval offset. For instance, from and including byte 8 of the file.

Live Examples, Documentation and Code

The implementation of the site is live at:

https://smarttariffsmartcomparison.org

Follow the process to create an account, this will give you access to your data.

ObservableHQ

A notebook environment that is linked to JavaScript, it takes care of setting up an environment and allows for easy sharing or you can keep your notebooks private

https://observablehq.com/@joshuacooper