Trust Relationships

The proposal is to secure the midata V2 API using OAuth2. In discussion about how to apply this to our solution we would welcome feedback from this group. What follows is an attempt to describe the proposals in a non-technical way to seek input from the wider group on the preferred way forward. We feel neither approach is wrong they just balance the needs of security and convenience differently.

The starting point for this discussion is that the purpose of midata v2 is to make it easier for the consumer to use their data for their benefit. Also, the goal is to address the widest possible audience and not be limited to consumers who have online accounts or are overly familiar with online technology.

Through the programme Consumers will get value by having access to multiple Applications (e.g. Switching Sites) and granting them access to the data held by Providers (e.g. Energy Suppliers).

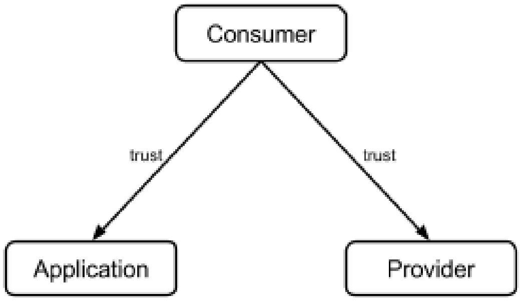
Our question relates to how we build a security model that is secure enough to protect the “not very” sensitive data proposed for the V2 API whilst maintaining a level of convenience for all parties to ensure that security does not act as a barrier to adoption.

Below are two options that we would like feedback on. Again the context here is sharing non-sensitive data (annualised consumption in kwh, current tariff, etc.). The description below uses 4 key terms to describe participants of the ecosystem:

- Consumer - an individual wanting to share their data to gain a service of value
- Developer - an entity or individual wanting to create a service of value
- Application - a product delivering a service to a Consumer
- Provider - an entity sharing data on behalf of a Consumer to an Application

Option A: Trust is derived solely from the Consumer.

In this model the Consumer chooses the Application and authorise the Provider to send data directly to the Application. The Application and the Provider do not need to trust each other or “know” each other. The data exchange between them is based on the consumers trust of both.



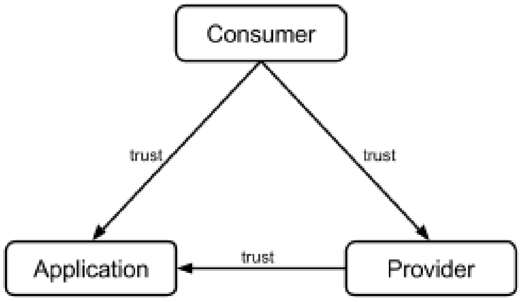
We believe that having a situation where the Application and the Provider do not have a trusted relationship allows the overall solution to be much more convenient for the Consumer, the Developers and the Provider.

For the Consumer it will mean there are more Applications that can add-value and using them will be simpler. For Developers, it facilitates innovation of Applications by removing a barrier to create new solutions of value. Finally, it reduces the cost of implementation to Providers by removing the need for them to each individually register Applications.

At a technical level it means that the midata API will be public. Anyone will be able to access it. Security however will still be maintained by the requirement for the Consumer to authenticate with the Provider each time the API is used by an Application.

Option B: Extends Option A by also having the Provider trust the Application

Option B adds to the security model a "trust relationship" between the Provider and Application. Each Developer who wishes to build an Application will have to register with every Provider to obtain a key. Everytime the Application uses the API on behalf of a Consumer the Application will have to provide the key in ADDITION to the Consumer providing their authorisation.



The benefit of this model over Option A is that each Provider now "knows" each Application and when it uses the API. Providers can revoke permissions of individual Applications or Developers if Applications are deemed harmful or are no longer useful. The downside of this model is that adoption of the API will be limited by how quickly Providers can make available this registration scheme. The registration scheme would be individual to each Provider as would be the criteria for which Developers would qualify to have a key. Both of these may limit the rate of deployment by Providers and adoption by Developers.

Request for Input

We would welcome views on which approach should be considered further and whether there are other viable options. For those that want to know the specific technical details of which each option may mean please contact the midata team at First Utility, ideally by responding to this discussion on Basecamp.

Option A and option B would look the same from the perspective of the Consumer. Option B could be extended further to increase security but this would undermine convenience for the Consumer. Although we want to provide sight such options exist we do not recommend these are pursued. Either way, obtaining the answer to the above A/B question may then open up the requirement to discuss derivatives of B further.

We believe that the more open approach in Option A is possible for midata IF we restrict the API to non-sensitive consumer data (non-personally identifiable), low-risk operations (read-only) and ensure any trust inferred is short-lived (enough time for the transaction being authorised to complete).

Option A has the benefit of being simpler to implement and public APIs tend to get adopted more rapidly. However we are trading convenience for security.

Option A can be extended to Option B at a later stage when it is deemed new APIs are likely to carry more sensitive data or provide access to "riskier" operations (updates, payments, etc.)

Importantly, as the midata API evolves into areas that do not fit this profile stronger security models will be required.