Data Plan Info API

*{aterzis, igrigorik, cbentzel}**@google.com***
*{sicking, mcaceres}**@mozilla.com***
*nrooney**@gsma.com***

Last update: 10/02/14

Many users are price sensitive and have limited cost and data budgets to access online services. This is especially true in developing markets, where data costs are high and data budgets are low - e.g. many users end up juggling multiple plans via multiple SIMs and across different providers. Those same users often end up disabling data access on their devices altogether, because (a) they don't trust the applications to respect their budget, and (b) they don't have visibility into the status of their plan - e.g. bytes remaining, type of contract, etc.

Improving the visibility on the incurred cost and status of the mobile plan can help improve the overall experience for the end users, application developers, and carriers:

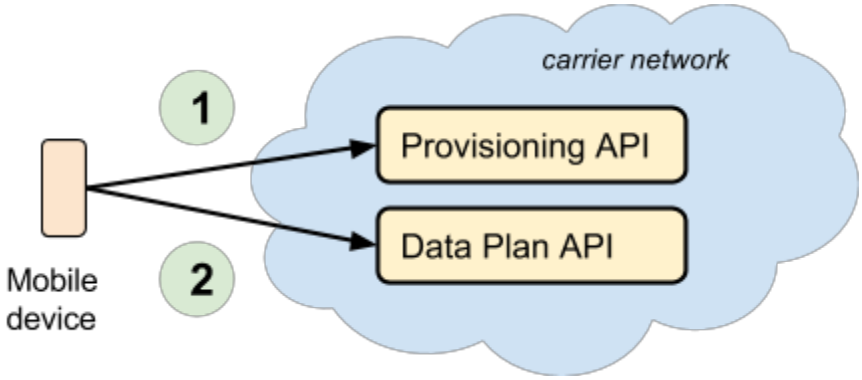
- Improved data usage by the end-users - i.e. reduce manual data management.
- Improved access to mobile data for applications leading to better user experience.
    - Improved trust in mobile applications to respect allotted data and cost budgets.
- Improved plan utilization, user trust with the carrier, and higher revenues.

Most mobile plans impose data limits - e.g. a data cap that is periodically refreshed within a predefined subscription period, a prepaid plan that must be manually "topped up" once the allotted limit has been reached, etc. However, despite the presence of these limits and their importance to the user, today there is no simple or user-friendly way for the mobile user, or the applications running on the mobile device, to identify the current properties and status of their data plan - e.g. is it active, type of contract, remaining bytes, and so on.

In the paragraphs that follow we propose a simple HTTP API that can enable different platforms (e.g., Android, iOS, Firefox OS, etc.) and applications (e.g. Firefox, YouTube, Google Photos, etc.) to learn this information, such that they can improve the user experience: make the data plan information and status visible to the user, adjust fetching logic (e.g. reschedule updates, warn user about large fetches, adjust types of assets being fetched), and surface this information to other applications running on top (web or native).

Mechanism Outline

We propose a standard Data Plan Status API that would be implemented by the carrier that allows the mobile device to learn about its current data plan. As the figure below illustrates, to retrieve we propose the following sequence of steps:



1. The client issues a request to the carrier's mobile provisioning URL[1]. This URL is defined and provided by the operator during the user's network authentication process.
    a. Because the URL resides on the carrier controlled network, we expect the operator to handle authentication, for example through header injection at the PGW, so the mobile device will not store any carrier specific tokens, or implement new authentication methods.
    b. If the operator does not have defined a mobile provisioning URL then the client may fetch a canonical list of per-provider URLs from a well known location. The list is indexed by MCC+MNC which are unique per operator and the browser can query for.
2. The response to the above request includes one or more URLs referencing various service endpoints that may be accessed by the device. One of these endpoints must provide the URL for the Data Plan Status API.
3. The client issues a query for its data plan information against the provided URL contained in the response from the previous step[2].
4. Carrier API responds with a JSON object that contains information about the user's data plan in a defined format. Details of this format needs to be figured out in order to address the range of plans that are out there. But it could look something like:

```
{
  "status": "active",  // values: active, overquota, disabled, roaming
  "type": "recurring", // values: pay-as-you-go, recurring, ...
  "reset_time": "Mon, 1 Sept 2014 00:00:00 GMT", // RFC 1123 Time Format
  "shared": true,                                // e.g. family plan
  "remaining_bytes": 8000000
  …
}
```

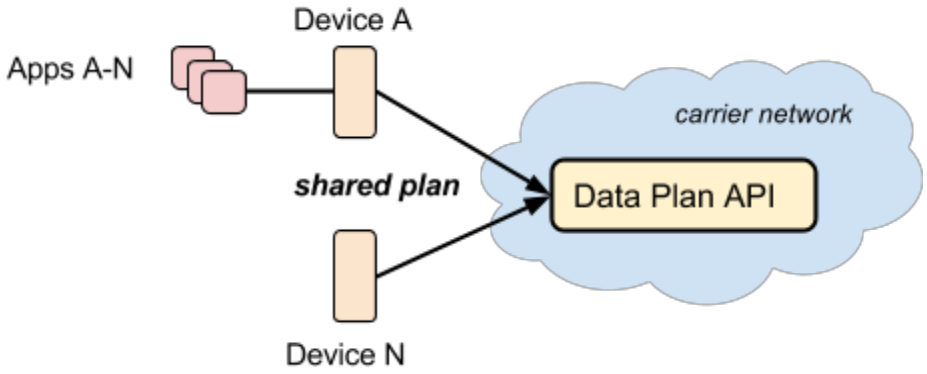
[1] There are existing mechanisms in place that allow the carrier to communicate a URL that can be surfaced to Android users to view their data plan. Our proposal is to reuse and extend this mechanism to provide a list of services to allow for an easily extensible integration beyond the Data Plan Status API.
[2] Client does not have to go through the two-step process each time, it can cache the first response.

The above response would indicate that:

- The user's plan is in good standing ("active")
- Has 8,000,000 bytes left
- Plan resets on September 1st
- Plan may be shared between multiple users or devices

Shared data plans

Shared data plans require special attention because the plan may be shared between multiple devices, which may belong to the same or different users (e.g., family plan), and multiple applications may be consuming data in parallel on each device:



To address this, and to reduce the query load on the Data Plan API, the applications running on the device may delegate the responsibility to query the Data Plan API to a system service, thus eliminating the need for each application to query the service on its own - e.g. an Android service can perform this and provide an API to surface the relevant information to other applications.

Further, in addition to keeping count of consumed bytes of each application on the device, the system service can implement different polling strategies based on type of data plan - e.g. if a shared data plan is used, then the system service may decide to query the Data Plan API more frequently to reflect data consumption between multiple active devices or users. The polling frequency may also be adapted based on number of remaining bytes, rate of data consumption, and other factors.