Seamless Mobile Services in the Cloud

Jagan Sankaranarayanan Hakan Hacıgümüş
NEC Laboratories America
10080N Wolfe Rd SW#3-350
Cupertino, CA 95014
Email: {jagan,hakan}@sv.nec-labs.com

Abstract—**The mobility of today is defined by the multitude of apps, which while working in isolation, can achieve a variety of tasks for the mobile user. The mobility of tomorrow is envisioned as one where mobile apps work together by sharing information to create a seamless mobile experience, where the focus is the mobility of the user but not the device. The ultimate goal of mobility is to create a seamless mobile experience, which can be achieved by mobile apps sharing data actively with one another in a cloud ecosystem.**

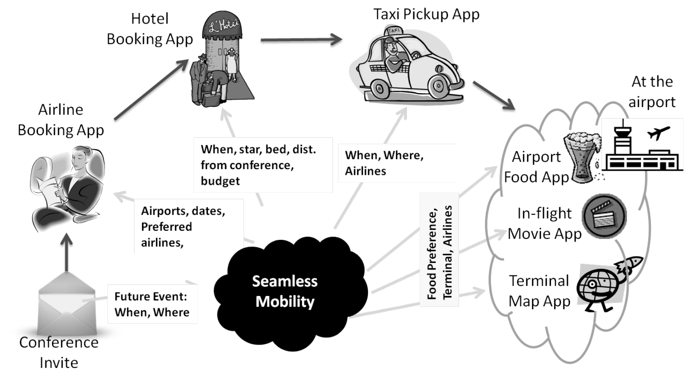

Fig. 1. A case scenario of mobile apps working together to create a seamless mobile experience for a user

I. Introduction

The ultimate goal of mobility is to ensure that the experience of a mobile user is a rich one. This means that individual apps (i.e., mobile services) should interact with the mobile user as if they exist in an ecosystem whose collective objective is to ensure that the mobile user can interact with the digital world in a *seamless* fashion. Mobile users frequently change their context as they navigate in their fast-paced daily lives, as described in an example scenario below. The desire is to ensure that mobile users stay connected to the digital world through mobile services as they move forth from one context to another – regardless of the kind of mobile device and sometimes even without a mobile device, which is largely irrelevant here. In this environment, the main constraints of mobility are the limited time, patience, and attentive span of the mobile user who is *on the go*. Although there have been recent efforts to significantly advance the capabilities of mobile devices to improve the user experience, more substantial improvements in user experience can be achieved if individual apps are cognizant to the limitations of the mobile user. In some sense, we want to move beyond traditional arguments that solely attribute the challenges of mobility to the limitations of the mobile device, but instead focus on how apps can provide a much better user experience. Apps that constantly adapt to the current context of the mobile users are said to exhibit "*seamless mobility*" [1].

The idea of seamless mobility is best described by a use case scenario, given in Figure 1. Our scenario begins with a user receiving an email confirmation of acceptance of a paper to a conference. The email app pulls out the event information and populates the calendar app with the relevant information. Later when the mobile user invokes an airline booking app to purchase tickets, the app has access to the conference details, using it to determine if the user will be traveling, which airport the user is flying to/from etc. The mobile user's interaction with the airline ticket booking app is the first instance of a seamless mobile experience, which happened because the calendar app shared information with the airline booking app to make the airline reservation process intuitive for the user. Of course, such a sharing arrangement between the two apps can never be perfect all the time as the user may want to travel on different dates, or use another airline; all these issues can be resolved by suitable user interface options on the app. Moreover, the goal of this example is not delve into the virtues of incorporating context into apps, which is quite evident, but instead motivate the utility of apps sharing information. Continuing with our example, next if the user invokes a hotel reservation app, it has access to the conference details, dates of travel, air ticket, and other information from the calendar and airline reservation apps, and tries to present the user with hotel options that are proximate to the conference venue as well as being on the same dates as the conference. Subsequently, when the user invokes an app to reserve a taxi ride to the airport, the app has all the required information such as the place to pick up and drop as well as relevant parts of the user's itinerary. When the user is at the airport, and looks for restaurants to dine, the app shows options from the same terminal where the user is currently located, but not showing those that the user did not prefer in the past.

The seamless mobility in our scenario is achieved by apps working together to help the mobile user achieve tasks on the go. The apps in our scenario greatly benefited from sharing information with one another. The mobility as we envision is far from what currently exists. Even though there is a rich variety of apps on mobile devices, they generally do not talk to one another let alone share information to create a seamless mobile experience. The communication between apps is *adhoc* in the sense that there is no real support to enable them.

For example, Yelp, which is a restaurant review app, allows users to post their reviews of restaurants using the Facebook app, which is a popular social networking service. As far as we can tell, the communication between Yelp and Facebook apps is facilitated by Yelp accessing the Facebook API. The problem with this setup is that it is unidirectional and not scalable in the sense that every app needs to implement the API individually. In general, APIs are expensive to create and maintain, not to mention that they may not be expressive enough for most sharing needs between apps. Moreover, as apps are often hosted in the same cloud infrastructure, there are other less expensive ways of enabling sharing of data between apps hosted in the cloud. While acknowledging the enormous security and privacy implications of apps sharing personal information of the mobile user, these issues are beyond the scope of this paper.

II. Sharing Service in the Cloud

Mobile apps can be viewed as front ends driven from remote services, which are typically hosted on the cloud. For example, a weather app on a mobile device is essentially a front end that queries a data store on the cloud infrastructure for the weather conditions at a certain zip code. Platform as a Service (PaaS) [2] provides hosting, processing and querying of data for any mobile app that wishes to use its services. A PaaS is the most appropriate place to build a service for sharing as it typically hosts data from several other apps. Sharing between the apps can be provided as a service with little or no overhead to the apps that use it. Sharing, in our context, adds value to all the parties involved by providing access to richer information on the mobile user. In the PaaS setting, mobile apps that use the PaaS to host their databases are referred to as *tenants*. Usually a *PaaS provider* hosts several tenants in the same cloud infrastructure. In other words, the PaaS provider usually resorts to *multitenancy* for good resource usage and spreading of the operation cost among several tenants.

There are several ways of enabling sharing between tenants in the cloud. The sharing in the cloud is usually between a tenant t, who is the *owner* of the data and another tenant, referred to as a *consumer*, who wants access to t's data. Consider a scenario of two apps, say App-A and App-B, hosted on the same cloud infrastructure that agree to share data. In particular, let us only consider the case of App-A, who is the data owner, agreeing to share data with App-B who is the consumer. For instance, App-A could be a calendar service, while App-B could be an airline ticket booking service that wants to query calendar appointments to determine if the user is traveling in the near future.

If App-A wants to share some of its data with App-B, in a traditional scenario, App-A would create an API and share the details of the API with App-B. The advantage of this model is that App-A is *loosely coupled* with App-B in the sense that App-A is free to change its data layout without really affecting App-B as long as the API is suitably updated. However, App-A must setup the necessary infrastructure to create the API as well as keep updating it whenever its data layout changes. An extreme solution would be if App-A allows App-B to access its data directly. The drawback of this arrangement is that it leads to a tight coupling between App-A and App-B. Moreover, if App-B is a *hard-hitter* (i.e., issues queries at a high rate) of App-A's data, which would lead App-A to have poor access on its own data. The PaaS provider can setup a *materialized* shared space for App-B, which ensures that App-B's access on the shared space will not significantly affect App-A's queries. Of course, materialized shared space takes up storage and is expensive to maintain so this solution, while attractive, must be used intelligently.

We therefore need a sharing service offering for mobile apps (e.g., COSMOS [1]) with the goal of supporting seamless mobility by enabling wide scale sharing between apps. To ensure that all the tenants get an acceptable level of service, in spite of sharing the infrastructure with several others, tenants will negotiate *Service Level Agreements* (SLA) with the PaaS provider. SLA is a contract that describes the level of service a tenant requires on the data hosted with the PaaS. For example, an SLA could specify that the tenant would pay 10 cents for queries responded within 300ms, while the tenant would penalize the PaaS $1 if the execution time for the query exceeds 300ms. The PaaS provider, whose objective is to maximize profits, enables sharing between apps while ensure that the tenants do not miss their SLA deadlines too often as that results in a loss of revenue.

III. Concluding Remarks

This paper laid out a vision of mobility, where apps collaborate to create seamless mobility for the mobile user. As mobile users have a common identity across all the mobile apps, sharing information between apps can lead to the development of interesting services as well as a much richer experience for the mobile user. From an intuitive point of view, sharing creates rich data and the utility of rich data can be readily seen by considering the following three classes of apps:

- Free apps: These apps are usually supported by advertisements. A richer data on the users means more targeted advertisements.
- Paid apps: Richer data in turns means more compelling features, which could be a good incentive for users to pay for these apps.
- Enterprise apps: Sharing support in the enterprise cloud means reduced enterprise silos.

Therefore, there is a natural incentive for mobile apps to share information with one another, even with the current limited availability of services to aid sharing. We believe that by building support for large-scale sharing between mobile apps, we can usher in an era of seamless mobility.

References

[1] J. Sankaranarayanan, H. Hacigumus, and J. Tatemura, "COSMOS: A platform for seamless mobile services in the cloud," in *Proceedings of the International Conference on Mobile Data Management (MDM)*, vol. 1, Jun. 2011, pp. 303–312.

[2] Hacıgümüş, J. Tatemura, W. Hsiung, H. J. Moon, O. Po, A. Sawires, Y. Chi, and H. Jafarpour, "CloudDB: One size fits all revived," in *IEEE World Congress on Services (SERVICES)*, Miami, FL, Jul. 2010.