Open and Decentralized Access across Location-Based Services

Yiming Liu
School of Information
UC Berkeley
yliu@ischool.berkeley.edu

Rui Yang
Department of Information Engineering
The Chinese University of Hong Kong
yr007@ie.cuhk.edu.hk

Erik Wilde
School of Information
UC Berkeley
dret@berkeley.edu

ABSTRACT

Users now interact with multiple *Location-Based Services (LBS)* through a myriad set of location-aware devices and interfaces. However, current LBS tend to be centralized silos with ad-hoc APIs, which limits potential for information sharing and reuse. Further, LBS subscriptions and user experiences are not easily portable across devices. We propose a general architecture for providing open and decentralized access to LBS, based on *Tiled Feeds* — a RESTful protocol for access and interactions with LBS using feeds, and *Feed Subscription Management (FSM)* — a generalized feed-based service management protocol. We describe two client designs, and demonstrate how they enable standardized access to LBS services, promote information sharing and mashup creation, and offer service management across various types of location-enabled devices.

Categories and Subject Descriptors: H.3.5 [**Information Storage and Retrieval**]: Online Information Services — Web-based services, Data sharing

General Terms: Design, Documentation, Languages

Keywords: Location-Based Services, Feeds, Atom, RSS

1. INTRODUCTION

Recent years have seen the proliferation of mobile devices with Internet connectivity and GPS capabilities, and new Web services that exploit these capabilities. Despite this growth, consumption of LBS on user devices remains largely an ad-hoc affair. Most are centralized silos with proprietary APIs, which discourages repurposing and recombination of data. Sharing LBS data across devices is usually supported only on an application-specific basis. For example, a mashup mixing social presence with business recommendations requires ad-hoc integration with at least two LBS APIs.

In this paper, we describe a location-based service platform that allows for standardized access to LBS, and provides service management and data portability across any Web-capable device. The system is based on two technologies: *Tiled Feeds* [2], a system for delivering open LBS via feeds, and *Feed Subscription Management (FSM)*, a feed-centric service management protocol. We demonstrate our system using two prototype clients: a Web-based client that provides a full-featured, read-write interface to the platform, and a mobile client that provides read-only access to stored LBS subscriptions and views. This platform enables applications to consume and mashup location-based services without costly ad-hoc integrations, and deliver uniform, portable experiences to all types of devices.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

2. LOCATION-BASED SERVICES

Many *Location-Based Services (LBS)* available today are either vertically integrated [3], or they are built on location concepts which are specific to one service. In our work towards Web-oriented LBS architectures [1], our focus is on models which allow simple LBS integration and repurposing, so that it becomes simple to aggregate LBS. Some of the current limitations of the *XMLHttpRequest* API make it hard to deploy truly decentralized browser-based LBS, but improvements of the restrictive *Cross-Origin Requests* policy are underway, and non-browser applications can already take full advantage of RESTful LBS architectures.

The LBS architecture is based on two technologies, *Tiled Feeds* [2], and *Feed Subscription Management (FSM)*. Tiled feeds is a RESTful architecture for the representation of and interaction with location-based services. In short, the world is recursively divided into equal-sized tiles. Each tile is represented by a tile feed (an Atom feed with a few geospatial extensions), which contains geospatial features as entries. To retrieve data from such a LBS, a client needs to access the tiles it is interested in. To add new LBS functionality, or to create a mashup, the same tiles can be accessed across multiple LBS. The tiled feed model creates a standardized, Atom- and AtomPub-based interface across various LBS.

For managing the various services a client is interested in, FSM provides an architecture which allows the decentralized management of service subscriptions. Subscribed feeds can be regular feeds (which may contain location-tagged entries) or tiled feeds, in which case clients can interact with them by using the interlinked feeds that are available for various tiles. Since subscription data is managed openly, subscriptions can be shared across multiple clients, which means that a user can for example subscribe to feeds that are relevant to him and his location in a browser (Section 3), and can then reuse the exact same set of LBS in a mobile application (Section 4).

3. WEB CLIENT

The Web Client is a lightweight Web application that reads LBS subscription information from a FSM server, and creates arbitrary mashup views on a Google Maps-based interface using live data from tiled feed services.

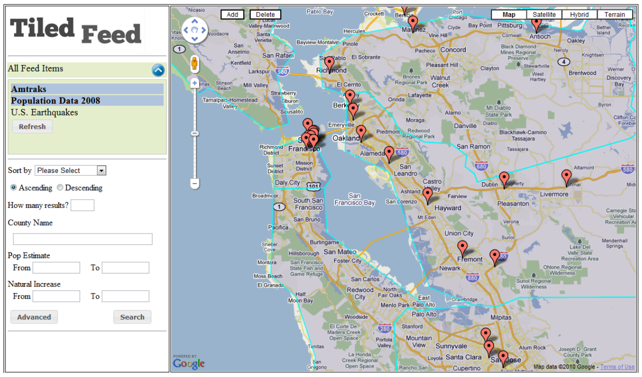

Figure 1: Tiled Feed Web Client

Figure 2: iOS Tiled Feed and FSM Client: Showing LBS List from FSM and View of the East Bay

The client contains two main parts, a tiled feed reader and a Google Map visualization. Besides using a server-side HTTP request proxy to circumvent current cross-site scripting restrictions with *XMLHttpRequest*, it is a purely client-side application written in JavaScript. The client provides most of the functions of a typical feed reader such as Google Reader: it can load subscriptions, read feeds, and manage subscriptions. Being tiled feed aware, it presents features from subscribed LBS on the Google Map visualization. Further, it allows user to query tiled feeds via an automatically generated query form and publish new features to the tiled feed server using AtomPub.

When the client is loaded, it reads the list of subscribed LBS from a designated FSM server, and displays the list of services. After a user clicks on a service, the web client attempts to identify the feed as a tiled feed LBS; if successful, it accesses the feed's query schema (if any) and tile resolution services. It retrieves a list of all the tiles inside the current map view, at the appropriate level of resolution, and reads each feed in that list. The content, typically KML features such as points, polygons, landmarks, and associated metadata, are parsed and displayed on Google Map visualization. As an example, Figure 1 shows a visualization populated by U.S. population census and earthquake LBS for the San Francisco Bay Area. The viewport is freely moveable, and new data is loaded as needed from the selected LBS. Feeds are cached locally so every feed will only be retrieved once. New LBS in tiled feed format can be added to the subscription list via the FSM protocol.

If a feed is query-able, which means the tiled feed provider provides a tiled feed query schema, the web client is able to use the schema to generate a query/filtering interface automatically. This query interface is displayed on the sidebar. A user can choose different query parameters to refine the feeds contents. In addition, if the server supports read/write capabilities via AtomPub, the web client allows users to publish new content to the tiled feed LBS. A user can create new features, edit and delete existing features from each subscribed service using HTTP `POST`, `PUT` and `DELETE` respectively.

4. MOBILE CLIENT

The mobile tiled feed client uses same FSM and tiled feed services to deliver a more in-context, consumption-oriented LBS experience. Built on the iOS platform, the mobile client faces more resource, display, and interaction constraints than the Web client. Therefore, unlike the more free exploring Web client, the mobile client uses the concept of *views*, or preconfigured mashups of LBS data at specific areas of interest. A user can create views at any level of resolution for particular areas, and then specify which LBS should be used to populate the view. As an example, Figure 2 shows the East Bay Area view, which displays a mashup of nearby earthquakes and train stations, both of which are being delivered by subscribed LBS. Views themselves also managed by FSM as view feeds, and can be retrieved or stored on an FSM server using Atom and AtomPub.

Based on this architecture, the mobile client can refresh both its subscribed LBS and its view from an FSM server, and thus could be implemented in a very similar way on devices with very constrained UI capabilities such as car navigation systems.

Functionally, the tiled feed client works very similarly to the Web client. It reads a FSM subscription feed to determine the LBS needed, and uses tiled feed processing to retrieve geospatial data from appropriate tile feeds and displays the contained features on a map visualization. The view feed, also part of FSM, is used to determine the preset areas that are available for display. Some scrolling of the map viewport is available, but due to resource constraints on mobile devices, it is limited in scope compared to the world-exploring capabilities of the Web client.

5. REFERENCES

[1] Martin Kofahl and Erik Wilde. Location Concepts for the Web. In Irwin King and Ricardo Baeza-Yates, editors, *Weaving Services and People on the World Wide Web*, pages 147–168. Springer-Verlag, Heidelberg, Germany, August 2009.

[2] Yiming Liu and Erik Wilde. Scalable and Mashable Location-Oriented Web Services. In Boualem Benatallah, Fabio Casati, Gerti Kappel, and Gustavo Rossi, editors, *10th International Conference on Web Engineering (ICWE 2010)*, volume 6189 of *Lecture Notes in Computer Science*, pages 307–321, Vienna, Austria, July 2010. Springer-Verlag.

[3] Yiming Liu and Erik Wilde. Personalized Location-Based Services. In *iConference 2011*, Seattle, Washington, February 2011.