Loosely Coupled Location Based Services

Oracle Tech Talk

Erik Wilde, School of Information, UC Berkeley

January 26, 2011

http://dret.net/netdret/publications#oracle11

[http://dret.net/netdret/publications#oracle11]

Contents

- Abstract .................................................................................................................. 2
- 1 Introduction
  - Who Am I? ........................................................................................................... 4
  - Where Do I Work? .............................................................................................. 5
  - What Do I Teach? ............................................................................................... 6
- 2 Location-Based Services (LBS)
  - Precious Snowflakes ........................................................................................... 8
  - LBS as Services ................................................................................................. 9
  - Horizontal Integration ....................................................................................... 10
  - Feeds: Web Containers ..................................................................................... 11
  - Feeds as RESTful Web Plumbing ...................................................................... 12
  - Feed Reader ....................................................................................................... 13
  - What's Missing? ................................................................................................. 14
- 3 Tiled Feeds
  - Linking Spatial Information ............................................................................. 16
  - Tiled Feed Example .......................................................................................... 17
  - Tiled Feed Reader ............................................................................................. 18
  - Implementing Tiled Feeds ............................................................................... 19
  - Proof of Concept ............................................................................................... 20
- 4 Feed Subscription Management (FSM)
  - Managing Decentralization ............................................................................. 22
  - FSM Architecture ............................................................................................. 23
  - OPML "Folder" in FSM .................................................................................... 24
Abstract

Location-Based Services (LBS) are a popular use case for the Web as well as for mobile devices. However, the vast majority of LBS today is built in a vertically integrated way, where the map imagery and all overlays for it are delivered by the same service provider. We propose an architecture and present a prototype where map imagery as well as overlays are managed in a decentralized way, allowing clients to easily combine and repurpose services. Since in such a scenario, clients may want to keep track of the various LBS providers they are using, we also propose and present an architecture for managing subscription information for services. Both areas of work are based on lightweight Web technologies, most importantly Atom feeds and a RESTful service design around those. Using this architecture, subscription information can be easily shared among clients (allowing users to for example use the same LBS providers on their laptop, their smartphone, and in their car), and display and delivery methods can be adjusted according to client preferences and capabilities (display size, pull or push delivery, and similar considerations).
Who Am I?

- Post-Doc at [ICSI, Berkeley](http://www.icsi.berkeley.edu/) (1997/98)
- Various activities in Switzerland (1998-2006)
  - teaching at [ETH Zürich](http://www.ethz.ch/) and [FHNW](http://www.fhnw.ch/)
  - working as independent consultant
- Professor at [UC Berkeley School of Information](http://ischool.berkeley.edu/) (since Fall 2006)
  - Teaching [Web Architecture](http://dret.net/lectures/web-fall10/)
  - Technical Director of the [Information and Service Design (ISD) program](http://isd.ischool.berkeley.edu/)
- Erik’s representations: [Web](http://dret.net/netdret/); [Blog](http://dret.typepad.com/); [Twitter](http://twitter.com/dret)

Where Do I Work?

- Interdisciplinary program for Masters and Ph.D.
  - Diverse Faculty from a wide variety of disciplines
    - Computer Science
    - Economics
    - Law and Public Policy
    - Social Sciences
    - Linguistics
- Managing breadth vs. depth is a problem
- [Information schools](http://en.wikipedia.org/wiki/List_of_I-Schools) are quickly growing
What Do I Teach?

Location-Based Services (LBS)

Precious Snowflakes

- Many applications are not as unique as you might think
  - reusing design patterns and even technologies is a good idea
  - avoiding reuse usually should be justified by good reasons
- Take the Web's HTML as an inspiring example
  - HTML is not all that great as a document format
  - it was good enough as a container for a lot of useful content
  - the network effect far outweighs its functional shortcomings
- could you imagine a Web based on PDF or Word?
- Simplicity and wide applicability are very valuable
  - simplicity means a lower barrier to entry
  - wide applicability means higher chances to create network effects
LBS as Services

- Most LBS platforms today are "vertically integrated"
  - LBS "framework" and imagery and resources from the same provider
  - allows server-side optimizations and data processing
  - no ability for users to freely mix and match LBS providers
- Our goal is a "horizontally integrated" LBS architecture
  - LBS expose their services according to a unified LBS model
  - users can mix and match LBS services in client-side platforms
  - server-side scenarios are still possible through intermediaries
- Various challenges in such an architecture
  - hitting the 80/20 point (avoiding unnecessary complexity/simplicity)
  - scalability (using dozens of LBS with each exposing many resources)
  - abstraction (HTML for documents, no established standard for "spatial resources")
  - interaction (designing for limited resources and various delivery channels)
Feeds as RESTful Web Plumbing

- Atom is a format and also an evolving landscape [http://dret.typepad.com/dretblog/atom-landscape.html]
  - Atom [http://dret.net/netdret/docs/rest-icwe2010/practice#atom], the feed format
  - AtomPub [http://dret.net/netdret/docs/rest-icwe2010/practice#atompub], write access to Atom-oriented collections
  - support for threaded discussions in feeds
  - feed licensing
  - feed paging and archiving
  - many related ongoing developments [http://dret.typepad.com/dretblog/atom-landscape.html]
- A landscape of RESTful interactions with "collections of things"
- Some parts of the map are still under construction
Feed Reader

Spatial Information
- LBS expose information that has an important spatial component
- feeds often are ordered by time, but ranking/ordering is up to the provider
- LBS need to say where to put a resource and what to put there

Subscription Management
- decentralized scenarios need to keep track of the aggregated sources/services
- OPML (https://www.google.com/reader/subscriptions/export) is very limited but better than nothing
- subscription management should be a service in itself (API for changing subscriptions)

Query Capabilities
- not covered in this talk (but essential for many scenarios)
Tiled Feeds

Linking Spatial Information

- Partition earth into tiles and provide various zoom levels
  - tiles are linked and thus clients can follow links to request tile feeds
  - links are to neighbors on the same zoom level and for zooming in and out
- LBS providers decide what to publish in tile feeds
  - populating the "world feed" is useful in some scenarios, useless in others
  - smart LBS providers can cluster resources and publish aggregated resources
- Possible improvements for the future
  - describing coverage (avoiding unnecessary requests)
  - support for clustering (allowing clients to "understand" clusters)

Tiled Feed Example

```xml
<feed xmlns:fh="http://purl.org/syndication/history/1.0" xml:lang="en-US" xmlns="http://www.w3.org/2005/Atom">
  <id>http://tiledfeeds.yimingliu.com/amtraks/0230102122.xml</id>
  <link type="text/html" href="http://tiledfeeds.yimingliu.com/amtraks/0230102122.xml" rel="alternate"/>
  <link type="application/atom+xml" href="http://tiledfeeds.yimingliu.com/amtraks/0230102122.xml" rel="self"/>
  <title type="text">Tile 0230102122</title>
  <updated>2011-01-25T18:08:48-08:00</updated>
  <author>
    <name>TileFeed Generator</name>
  </author>
  <link type="application/vnd.google-earth.kml+xml" href="/amtraks/0230102122.kml" rel="alternate"/>
</feed>
```

<entry>
  <id>http://localhost:3000/items/345</id>
  <link type="text/html" href="http://localhost:3000/items/345" rel="alternate"/>
  <title>BKY</title>
  <updated>2011-01-25T18:08:48-08:00</updated>
  <content type="application/vnd.google-earth.kml+xml">
    <kml xmlns="http://www.opengis.net/kml/2.2">
      <Placemark>
        <name>BKY</name>
        <description>University Ave. and Third St.</description>
        <Point>
          <coordinates>-122.299994999871,37.8676030004272</coordinates>
        </Point>
      </Placemark>
    </kml>
  </content>
</entry>

<entry>
  <id>http://localhost:3000/items/111</id>
  <link type="text/html" href="http://localhost:3000/items/111" rel="alternate"/>
  <title>EMY</title>
  <updated>2011-01-25T18:08:48-08:00</updated>
  <content type="application/vnd.google-earth.kml+xml">
    <kml xmlns="http://www.opengis.net/kml/2.2">
      <Placemark>
        <name>EMY</name>
        <description>5885 Horton Street</description>
        <Point>
          <coordinates>-122.290680000357,37.8406790003872</coordinates>
        </Point>
      </Placemark>
    </kml>
  </content>
</entry>

<entry>
  <id>http://localhost:3000/items/509</id>
  <link type="text/html" href="http://localhost:3000/items/509" rel="alternate"/>
  <title>OAC</title>
  <updated>2011-01-25T18:08:48-08:00</updated>
  <content type="application/vnd.google-earth.kml+xml">
    <kml xmlns="http://www.opengis.net/kml/2.2">
      <Placemark>
        <name>OAC</name>
        <description>700 73rd Avenue</description>
        <Point>
          <coordinates>-122.197740496936,37.7518812571236</coordinates>
        </Point>
      </Placemark>
    </kml>
  </content>
</entry>

<entry>
  <id>http://localhost:3000/items/515</id>
  <link type="text/html" href="http://localhost:3000/items/515" rel="alternate"/>
  <title>OKJ</title>
  <updated>2011-01-25T18:08:48-08:00</updated>
  <content type="application/vnd.google-earth.kml+xml">
    <kml xmlns="http://www.opengis.net/kml/2.2">
      <Placemark>
        <name>OKJ</name>
        <description>245 Second Street</description>
        <Point>
          <coordinates>-122.197740496936,37.7518812571236</coordinates>
        </Point>
      </Placemark>
    </kml>
  </content>
</entry>
Implementing Tiled Feeds

Spatial database managing LBS data
- PostgreSQL [http://www.postgresql.org/] relational database for data management
- PostGIS [http://postgis.refractions.net/] adding support for geographic objects to PostgreSQL

Existing datasets with various geographical features
- point-based data (Amtrak, earthquakes) as well as polygons (U.S. counties)

Query features exposed by the tiled feeds and mapped to PostgreSQL
- mapping from URI query parameters to SQL queries is done manually
- processing spatial and/or query requests can become fairly expensive
Feed Subscription Management (FSM)

Managing Decentralization

- LBS can be exposed and used by anyone
  - access control and personalization can be implemented if required
  - clients need to keep track of what and how they aggregate

- Feed Subscription Management (FSM) is a decentralized service itself
  - subscriptions to feeds are managed in feeds ("subscription feeds")
  - subscription feeds can be published and reused as bundles
  - views are defined by map regions and services for that region

- FSM allows seamless switching between (spatial) feed clients
  - subscription and view management uses a RESTful API (based on Atom/AtomPub)
  - FSM represents feed subscriptions, query parameters, and views

FSM Architecture
OPML “Folder” in FSM

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fsm="http://dret.net/fsm/1.0">
<title>delicious REST</title>
<generator>$Id: opml2fsm.xslt 1407 2010-10-27 19:15:56Z dret $</generator>
<entry>
<title>Delicious/cesare.pautasso/rest</title>
<fsm:subtype>subscription</fsm:subtype>
<link rel="alternate" href="http://feeds.delicious.com/v2/rss/cesare.pautasso/rest"/>
<link rel="http://dret.net/fsm/1.0/subscription_to" href="http://feeds.delicious.com/v2/rss/cesare.pautasso/rest"/>
</entry>
<entry>
<title>Delicious/dret/REST</title>
<fsm:subtype>subscription</fsm:subtype>
<link rel="alternate" href="http://feeds.delicious.com/v2/rss/dret/REST"/>
<link rel="http://dret.net/fsm/1.0/subscription_to" href="http://feeds.delicious.com/v2/rss/dret/REST"/>
</entry>
</feed>

OPML Structure in FSM

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fsm="http://dret.net/fsm/1.0">
<title>dret subscriptions in Google Reader</title>
<generator>$Id: opml2fsm.xslt 1407 2010-10-27 19:15:56Z dret $</generator>
<entry>
<title>xkcd.com</title>
<fsm:subtype>subscription</fsm:subtype>
<link rel="alternate" href="http://xkcd.com/atom.xml"/>
<link rel="http://dret.net/fsm/1.0/subscription_to" href="http://xkcd.com/atom.xml"/>
</entry>
<entry>
<title>delicious REST</title>
<fsm:subtype>bundle</fsm:subtype>
<link rel="alternate" type="application/atom+xml"/>
<link rel="http://dret.net/fsm/1.0/bundle" href="fsm-67.xml"/>
</entry>
</feed>
FSM in the Web Client

FSM in the iOS Client
Conclusions

**LBS Need Loose Coupling**

- Cross-platform LBS allow loose coupling between LBS and clients
- **Tiled Feeds** [Tiled Feeds (1)] provide a scalable platform for LBS
- Query-enabled feeds allow a higher level of sophistication for feed-based scenarios
- **FSM** [Feed Subscription Management (FSM) (1)] allows LBS consumption to be client-independent

---

**Future Work**

- Looking at a wider range of feed query scenarios
- Using FSM as the management part of feed-oriented push (such as PuSH [http://code.google.com/p/pushubbub/])
- Exploring tiled feed additions such as clustering and coverage descriptions
Thanks! Questions?