# Location Based data sharing

**CSE 589 Course Project**

**Supported by Microsoft Research**

**(Hawaii Cloud-Enabled Mobile Computing Project)**

**Naveen Rawat**

**Vikram Sawant**

**Under guidance of**

**Dr. Dimitrios Koutsonikolas**

# Location based data sharing

## Abstract

In today's world information is considered as currency. Information sharing is responsible and will be responsible for many novel ideas to come. In fact it has been main driving force behind human evolution itself. Social networking started with this idea and soon developed into this huge mega monster that is providing lots of new opportunities to creative minds if used properly.

Continuing on same track we developed an information sharing service with respect to location. This idea coupled with technology we have today, can really take advantage of existing ideas and generate a whole new customer base for products that never existed before. We are trying to create services based on information sharing coupled with location awareness.

## Introduction

Consider a scenario, currently, you want to play a football and you have no friends nearby you. So there should be a service that will solve your problem and this thought motivates us to design this system "Location based data sharing".

In this system user provides information to system also it provides metadata about same information. This information will be shared with users having same preferences as metadata, location until some threshold time limit.

We used terms tagging, location-tweets, and threshold time limit for metadata, shared data and TTL respectively.

## Hypothesis

To define this solution we identified following components:

### Users with location device

- The user is a core part of the system. He registers to the LBSTweets service.
- Users will update their location in Central coordinator with specific interval of time (automatically done by application).
- Sending location tweets and accepting tweets

### Location-Tweets

- This is the information shared by or to users.
- This information contains user tweets, his current location, and current timestamp.

### User choices/tags

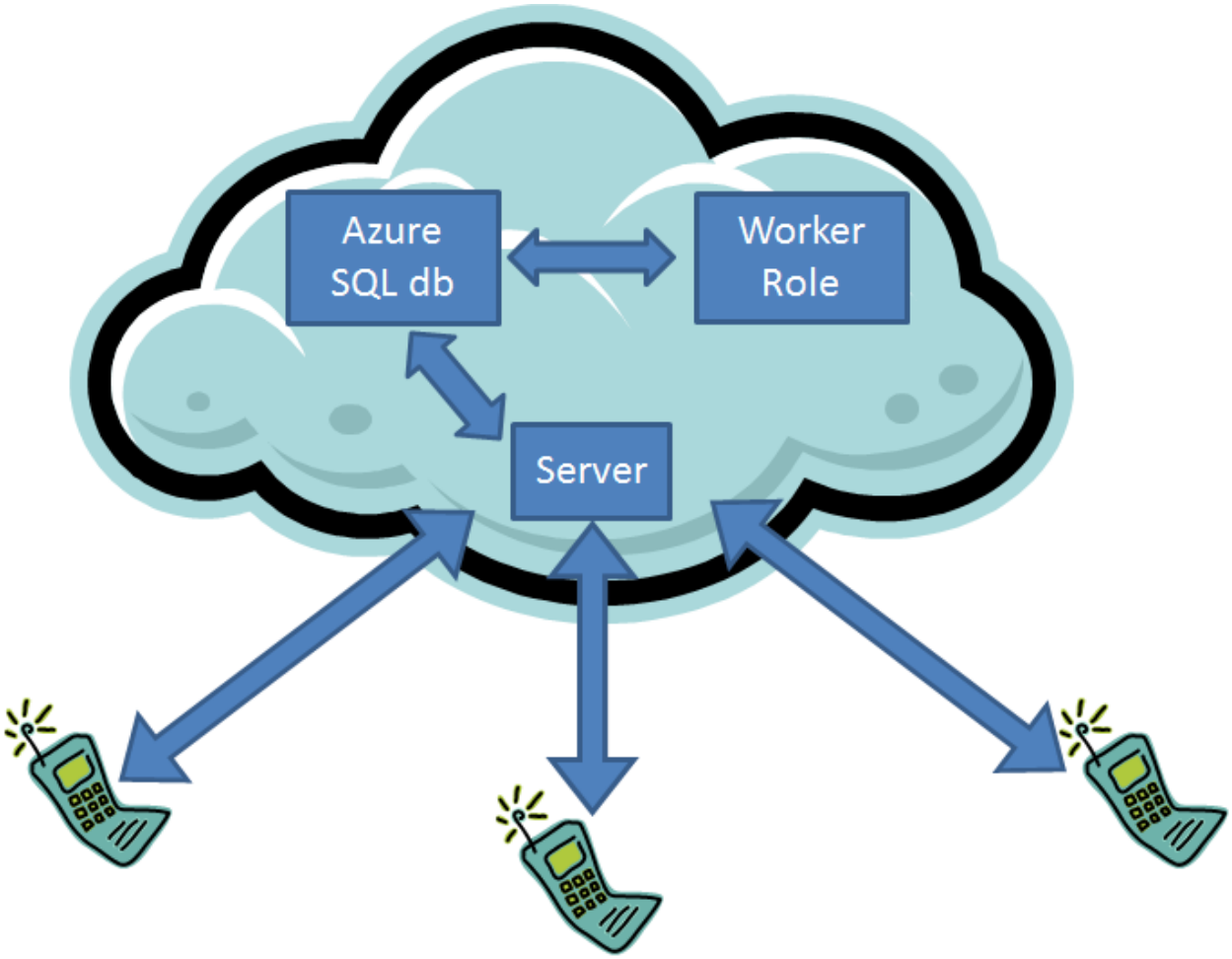
- Users may not want every updates of location from the service.
- So users will get updates/tweets for his/her specific choices.

### Central coordinator

We have setup following two components which constitutes the central core of server:-

- Server is setup using Internet Information Services (IIS) 7 Manager on a Windows Server 2008 R2 Datacenter, virtual machine running on cloud. Server is responsible for:-
    - Accepting tweets from all registered users.
    - Processing and sending tweets.
- Azure SQL Database is responsible for:-
    - For storage according to location, timestamp and user choices/tags.
    - Performing distance calculation and filtered query based on distance from user's location.
- Worker Role is responsible for:-
    - Perform optimization and maintenance job on database to remove obsolete tweets.

## High-level diagram




The proposed solution to this mainly consists of use of cloud services from client devices such as cell phone and other mobile devices. Cloud basically provides infrastructure as service, platform as service and software as service. We are using infrastructure as service from cloud for virtual machine and Azure SQL. On the client side we used windows phone 7 devices. User will have to install our application and create a user profile to be able to use this service.

## System overview

### Installation

- User installs client side application on his phone i.e. windows phone 7.
- Using this application user creates account by providing his mobile number and password. This is one time step.
- User logs into the application by providing his valid credentials.
- He can set his preferences according to choices like Entertainment, sports, shopping, campus update. Also user sets radius
- This system provides following services to user sets radius up to which he wants to get updates.
- To get the updated tweets user selects refresh button.
- These tweets are stored in persistent store of device.
- User send location tweets by selecting compose button.
- In this user sends location tweets by attaching various tags.
    - Entertainment: In this service user gets the updates related to painting exhibition, movies, plays, etc.
    - Sports: In this service user gets the updates related to different sports event
    - Shopping: In this service user gets the updates from shopping center about deals, coupons, special discount, etc.
    - Campus update: In this service user gets updates related to campus like faculty candidate talk.

## Design Decisions

### Device Limitation

Like all mobile devices windows phone 7 has limitation like store, process large amount of data. So we shifted this functionality at server side at client side we used HTTP protocol to communicate with the server. Also parsing of HTTP response done at client side to provide result to user. Windows phone 7 not provide multitasking feature for 3rd party applications. So we used user initiated approach. In this approach user explicitly refresh the application to get the latest tweets.

As we cannot run 3rd party application in background we stored the application state in persistent store of the device. Also location APIs in WP7 uses asynchronous calls. So we used threading model for fetching location.[4]

### Communication model limitation

Maximum size provided by Queue Service REST API is 64kb which will be not be sufficient for sending all tweets as number of users increase and as subscription list grows. Just to keep things simple and avoid multiple packets, instead of Queue Service REST API we used web services interface to provide communication between server and client modules.

For any message the client hits a URL and gets a SOAP based XML document in return. Then using simple XML parsing, client can fetch the message. This makes the communication model very simple, without needing to have to agree on a complicated communication mechanism. This also makes the use of our services very generic so that we can easily build up other applications using same web services.

#### Garbage Collection

As database size will increase with more number of users and high volume traffic, removal of old tweets is needed to keep database size low. Otherwise this will prove to be a major scalability bottleneck. SQL azure does not provide SQL Server Agent functionality for configuration and maintenance jobs. Hence to do this job, we have configured an Azure worker role which will at regular intervals scan the database for obsolete tweets and discard them.

#### Optimization

Another challenge as tweets in database go into hundreds of thousands is that of optimization. Fetching these tweets and calculating distance at server will be a heavy task at server side. To rectify this we moved all calculation oriented task to database by creating procedures and functions and performing query directly based on distance. Performing heavy computational tasks at database side is always advisable rather doing them on server which should not be highly available for serving client application’s request.

## Related work

Twitter - The famous micro-blogging site also provide tweet services. It is used to follow the people not location. Also tweets in the twitter remains persistent while in our case we defined TTL to remove the tweets from system automatically.
http://synrg.ee.duke.edu/ppts/microblog-slides.pdf. This application presented here also builds on the idea of geo-tagging. However our build this application with different target in mind. This application is basically used for answering query whereas our app is used to share generic information. Also we plan to provide users with only tweets from relevant category that he/she has subscribed to. Going forward with geo-tagging and time-stamping we can generate lot more features for applications.

## Future work

## Location verification

Success of this system solely depends on user location. And it is not difficult to fake user location. So we have to identify such events in the system. The possible solution may be use reference system with certification. Also poll based location authentication. In poll based authentication device will collect votes related to location from authenticated devices near its vicinity.[1]

### Intelligent tagging

Going further as we need to find ways to make our application more users friendly and intuitive, we can add feature of intelligently tagging tweets for the categories they should belong to, rather than asking user for manually tagging tweets. This will simplify the interface and spare users of hassle of deciding tags for tweets, thus speeding up the process and increasing the probability and volume of tweets generated.

### User feedbacks on tweets for better service

Success of the application mainly depends on how useful does the user finds these tweets received through our application and how well does an average user contributes in tweet generation. For this it's very important that users do not consider tweeting as worthless task but rather something that benefits him/her. For this it is logical to develop a feedback based incentive system just like P2P applications. If providing better and correct tweets can somehow benefit end user, he will be more inclined to generate more and accurate tweets. For this we can develop a feedback system, where end users will rate tweets provided by end users. Each tweet can be

rated by receiver, “thumbs up” or “thumbs down”. This way user receiving more and more positive feedback will build a good reputation, which then can be used by system to provide with better tweets to such users, like better shopping deals and such other incentives.

## References


**[1] Stefan Saroiu, Alec Wolman** - Enabling New Mobile Applications with Location Proofs.

[2] http://research.microsoft.com/en-us/projects/hawaii/

[3] https://www.windowsazure.com/en-us/develop/

[4] http://msdn.microsoft.com/en-us/library/gg490770.aspx

## Appendix

Screenshots