**GeoSocInt – Geo-Spatial Social Intelligence**

|  |
| --- |
| **Jarvis 2.0** |
| **Fall Semester, 2014** |
|  |
| |  |  | | --- | --- | | **Team Members** | **Email address** | | Lizabeth Workman | lworkman@apple.com | | Gaurav Bajaj | gaurav.bajaj@sjsu.edu | | Kovid Dev Reddy Nalla | koviddevreddy.nalla@sjsu.edu | | Sandyarathi Das | sandyarathi.das@sjsu.edu | |

**Table of Contents**

Abstract 3

Introduction 4

Realizing the Future 4

Ecosystem 4

Google Maps 4

Traffic Data 4

Family Location Services 4

Streaming positional updates 4

Event Tracking 4

Insurance GeoMapping 4

Bridging the Gap 5

Realizing the Future 5

Multi Level Mapping Integration 5

Geo-Fencing 5

Radius Calculations 5

Multi Traffic Marker Identification 5

GPS 5

GSM 5

Real Time Processing Algorithms 5

Online aggregation 5

Reliable estimate engine 5

One Pass data processing 5

Robust Real Time scale joining of continuous data streams 5

Database Design 5

SOLAP - GIS – OLAP 5

Fault Tolerance Management 5

Low Latency 5

Consumer Security 5

Application Realization 6

Architectural Approach 6

Data Capture and Processing 6

GUI Integration 6

Wrap Up References: 7

**ABSTRACT:**

This project ventures into the realm of geographical information science to design a location based application that is contextually specific to develop functionality that is accessible and personal through innovating the software processing to improve the user experience. This project will review the boundary processing of large data sets through both hardware and software in order to seamlessly provide the user with personal information to enrich their lives. This investigation is not limited to just location services but the relevant ecosystems that are available to augment the user experience.

This research would encompass the designing of a mobile application that would track the location of devices opted-in a group plan, sync up individual calendar information for group events and provide notifications based on real-time traffic updates.

Based on the location of the event members a push notification will be sent to notify the members about the time required to reach the event location. Critical time to arrive at the event will be updated. Geo-tagging of medical emergency centers on the group map processed based on the user’s Insurance provider would be an added feature.

Provision of a secure, opted-in group location sharing platform, provision of notification and decision enablers based on real-time traffic analysis and event-calendar collaboration forms the backbone requirements of the application. The report will involve the realization of Multi-level Mapping and Traffic Marker Integration from a single level mapping and traffic marking. Multi-level Mapping would involve combining multiple user-maps on a single device, employing geo-fencing methodologies. Whereas multi-level traffic marking would involve processing real time traffic information of multiple users combined with event data and time to event analysis using algorithms.

**Realizing the Future :**

**Eco-System :**

In bringing together different data sets and systems there are specific concerns in integrating the information in a manner that is both scalable and efficient. Conway (Conway,1967) highlighted from his paper that “in sociological observation that systems are designed like the organizations communication structure”. It is an interesting observation and highlights where the database and application landscape has moved to today. There are many examples of systems that have failed over the design and it is important to note that the organizations that tried to implement these applications had several layers of overhead. One example of this would be the New York City payroll system (Kanaracus, 2011) which had an original budget of $63 million but exploded to $232M. The project was finally scraped and several criminal probes were initiated. This occurred because several vendors were involved in the overall design of the project. In order to avoid this type of design mismanagement there needs to be oversight against defined and measureable design goals. A major contributor to success in the ecosystem design will be that the overall architecture has the ability to scale, maintain business process constraints, and have a co-existence to platforms that are already mature.

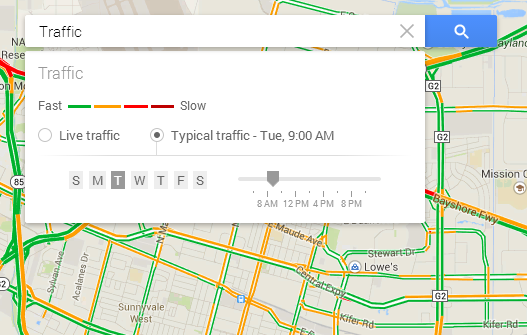
In the development of the Application for this project the team has defined key architectures that will need to be integrated together with the constraints outlined. To do this effectively each of these major areas of architecture will need to be reviewed against the database availability today with a forward look at flexibility for the next technology generation. In order to achieve this a critical eye will need to review designs and capture the key components of the overall architecture.

**Google Maps :**

A mobile web-mapping service application like Google Maps could be integrated with other data analysis applications ,creating a whole new paradigm of location/navigation based services . Satellite imagery, street maps, and Street View perspectives ,route planners for commuting by foot, a four wheeler or a bicycle, as well as by public transport, makes this mapping application a ubiquitous one-stop shop for any kind of navigation service. The Google Maps API, provides the required platform for embedding maps on third party applications.  The API provides a means to overlay our data on a customized Google map. Interactive and engaging web and mobile applications are developed using this platform which includes elevation profiles, demographics, analytics and an extensive places data base.

**Traffic Data :**

Harnessing the current traffic data , also forms the base for the provision of the traffic based location travel notifications. The different possible routes that could be taken to reach a particular on time.The Google Maps API provides a traffic layer that could be embedded in third party applications , that provides access to the usual traffic on a given route and as well the live traffic data, based on which notifications based on users event calenders and travel time suggestions could be notified.



Source : gis.stackexchange.com/questions/111980/access-traffic-data-using-google-maps-api

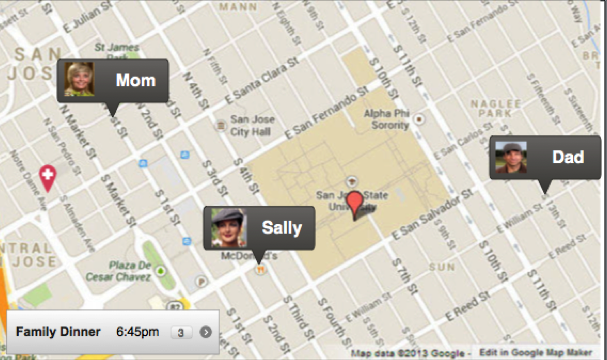
**Family Location Services:**

A Group Opt-In feature will encompass the provision of mapping of the location of all the members in the group plan. Every member , will be able to look up the location of every member on their map.

Streamlining positional updates : One of the core problems of GIS is of how the geometrical information is mapped. Geometrical information in general concaves two main classes ,the first class contains the absolute geometry. Absolute position of an object is referenced to a global reference frame, in the case of a GIS to an official coordinate system. The second class contains the relative geometrical information like local coordinates, distances or angles. This class of information is unique and non-redundant. Relative geometrical information is only considered as a view of the absolute geometrical information. Once the positional information is located, this has to be stream lined to employ the analytics and logistics to integrate it with a fixed map location. There are two ways to display GPS locations on a map using the API:

* Use a GPS layer
* Use a graphics layer and a custom GPS event listener

The first option is best for the simple display of GPS location data, including the current location, previous track points, and a trail connecting the track points. This option includes the ability to set a GPS mode on the layer, such as automatically panning the map to keep the latest GPS point visible. The second option is best if you want to further customize the display of GPS information, for example, to display satellite information on screen or change the GPS symbol based on the status of the GPS device. With either option, you can display the location data either from a connected device or from a text file containing raw NMEA sentences.



**Event Tracking :**

The Event tracking feature would accommodate the coordination, reminding and notification system of events that the user or the members of the opted-in group have added to their calendars. This feature would provide an event pop-up to which the user response about attendance is received and based on this decision provided by the user, timely suggestions based on approximate time to start by analyzing the live traffic data and routes. This would also collate the collective information of all the users in the family plan and provide a view to each member.



Insurance Geo-Mapping :

The application enables the provision of medical centers and hospitals based on one’s insurance policy around the user’s current or stored location. This feature feeds on the insurance data of the users in the group plan and also on the current location or daily commute locations and provides a map indicating the locations of the medical centers and hospitals and as well routes to reach the location in the shortest time from the current location or from the daily commute locations. This could also be used for pre-scheduled appointments made at the medical centers and corresponding reminder system is set up in collaboration with the Event tracking calendar.

Figure : Medical indications with red balloons.

Bridging the Gap :

**Multi Level Mapping Integration :**

Global Positioning System (GPS):

The Global Positioning System is a space based satellite navigation system providing information related to location and time across all weather conditions provided there is unobstructed line of sight to four or more GPS satellites. Our mobile uses the Google Map navigation which uses a internet connection to connect to the GPS navigation system for guided instructions to arrive at a destination.

Our application is leveraging on the top of single level traffic marker identification to multi traffic marker identification. An user could see the locations of all his circles or to whom he/she subscribed on his mobile device i.e., an user could see not only where he is heading towards but he could also view the traveling path of the people he/she follows. If all the users on the map have a common destination point which is mapped to an pre-defined event, then our application would update the real time traffic across the streets they are traveling and provides each of them with the estimated time they would be at the location.

Here we are integrating our geological location based maps with the event scheduler and real time traffic data as in stream to update our maps and resultant metrics simultaneously. For example, if Sally, Mom and Dad would like to attend an dinner event at 7 PM on a specific day. Each of them who registered for the event could see where the other person is on his/her mobile device. Sally can see where her mom or dad is and she can also retrieve information on the estimated time for her parents to reach to the location based on the real time traffic data. Once she thinks she could reach early and meanwhile until her parents she can do some chores and save time.

Users for example use Google Maps which offers satellite imagery, street maps and view perspectives which tracks with respect to the current user using the device.

From the current state of single level mapping enabled in our geologically location enabled devices, our mobile application implements the concept of multi level mapping integration. Besides tracking the location of single user who owns the device, the user can add his family or friends to his group. Once he adds the people to the group, he could as well view where his family or friends are in a single map along with his current location. Additionally , we are implementing the feature of Geo-Fencing as described below.

Geo-Fencing:

Geo-Fencing is the virtual representation of the physical location on the map. A geo-fence in our context is a predefined set of boundaries set by user to the members in his circle. When the location aware device of the location based service user enters the geo-fence boundary set by his family or friends in the circle, the user who is subscribed will get notifications and updates when ever he/she enters/exits the geo-fence area.

For example, our application provide information to parents on where their children or heading towards. Parents can draw geo-fence like a shape of a circle of certain radius on the map and tag their children for all the circles they define on the map. Multiple circles can be defined and each circle represents the physical representation of the geographical area virtually defined on the device. Once the children enter the geo-fence boundary set by parents, both the parents and children get a notification that they are entering into the

specific boundary area and the custom message tagged to the circle will be displayed like “This is a risky area. Be Cautious”.

Geo-fencing Types:

You can create your own geo-fence type by defining it on the map. Each type of geo-fence can have one of the following shapes like Circle Fence, Polygon Fence or Street-Link Fence.

Circle Fence:

Users would define a boundary on the maps in the form of circle. Our application takes the current position of the device of either the follower/followed person and if the distance between the current longitude/latitude and center of the point is smaller than the radius, then the person is inside the boundary or else outside.

Polygon Fence:

Users can define a geo-fence connecting arbitrary number of points in the shape of a polygon. Current position of the device latitude and longitude is noted and we use the ray casting algorithm to detect if the user is inside or outside the defined boundary. Each device location is also mapped similar to a interactive event which travels on the virtual map and each time it grazes through the boundary and enters , the device is said to entered into the area until it intersects any side of the polygon boundary.

Street Link Fence:

Users can define a geo-fence with respect to any street on the virtual map on the device and each time an user enters the street it would give the custom message set by the user for that street. A street link geo-fence consists of the border of the street and can be used to say whether it is inside or is traveling on the street link.

Real-Time Processing Algorithms

Our application mainly targets ios, android and windows phone users, so we’ll be writing native code in objective-C, java and C# respectively. The application is integrating our social data with geo-spatial data to enrich the user experience. For the social data we’ll be consuming APIs from different social networking web-sites like Facebook, Twitter, LinkedIn, for geo-spatial data we’ll be using Google Maps and for secure authorization, we’ll be using OAuth 2.0 protocol.

Now that we know that Google maps is one of the vital elements in our application, we’ll be consuming Google maps APIs for real time location data. Most of our features will be built on top of these APIs. Google Maps APIs consume and produce JSON message format, so we’ll be writing our client APIs, which will serialize the parameters in the same format. Additionally, we’ll use MongoDB as our database because it is friendly for location-based data; also it provides great support for large chunks of data via replica-sets and sharding features. The MongoDB instances will be further deployed on Amazon’s EC2 engines. Amazon EC2 is part of Amazon web services that enables computing on cloud platform.

Online Aggregation:

The Google APIs provides us with raw data that cannot be used directly in our application. This data requires aggregation before it can be consumed inside the application. But before aggregation, we need to store this data in the database so that we can process it efficiently, so we’ll be using MongoDB; though, frameworks like Apache Spark provide us with features like in-memory streaming data processing but MongoDB is way more friendly for location-based data. Google APIs gives data that contain location information in the form of latitude and longitude. This information is persisted in MongoDB as GeoJSON objects with this coordinate-axis order: longitude, latitude. We can also index the location using 2dsphere index, which is a feature of MongoDB.

MongoDB also provide us with aggregation framework, which is used to filter the real-time data according to our requirements. This framework is modeled on the concept of data processing pipelines. For the calculated results, documents will enter a multi-stage pipeline that transforms the document into an aggregated result. The [$geoNear](http://docs.mongodb.org/manual/reference/operator/aggregation/geoNear/#pipe._S_geoNear) pipeline operator takes advantage of a geospatial index.

Reliable Estimate Engine:

We’re consuming Geo-Spatial data in our application; Most of the features require Estimate Time of Arrival (ETA) based on this data; one way to do this by using in-built APIs in the SDKs provided by the vendor. For example, Apple provides an API in MapKit in order to calculate all this information.

Below is the snippet for the same:

MKDirectionsRequest \*request = [[MKDirectionsRequest alloc] init];

[request setSource:[MKMapItem mapItemForCurrentLocation]];

[request setDestination:destination];

[request setTransportType:MKDirectionsTransportTypeAutomobile];

[request setRequestsAlternateRoutes:NO];

MKDirections \*directions = [[MKDirections alloc] initWithRequest:request];

[directions calculateDirectionsWithCompletionHandler:^(MKDirectionsResponse \*response, NSError \*error) {

if ( ! error && [response routes] > 0) {

MKRoute \*route = [[response routes] objectAtIndex:0];

//route.distance = The distance

//route.expectedTravelTime = The ETA

}

}];

But, these calculations are based on the static information; it totally ignores the traffic delays and other delay factors like weather conditions, emergency blockage of routes.

So, when we talk about reliability, nothing can beat Google; hence, we’ll be consuming Google’s directions API for the real-time ETA based on our routes. Although, it’s a time and resource consuming task but reliability is our first priority.

One Pass Data Processing:

As we talked about pipelines in aggregation framework, which is inherited from the Unix world, we’ll be using these pipelines, which will constitute a channel that gives the final computed result. Although there’ll be multi-processing going inside the channel using thread-pools and future objects, but the result will come out in one pass from this channel.

#### Reliable estimate engine

Data capture will be completed through a user UI, sourced mapping data and internal processing algorithms. One aspect is to ensure future flexibility not only with Google Maps but also the ability to integrate real time, high-fidelity spatiotemporal data from transportation networks within the major cities. In order to accommodate this design schema a way to integrate though Google Maps this information will augment the time to event predictions. While Google maps will provide most of the speed predications these will need to be calculated against the event and party information within the application db. This is the predication horizon and will be processed with the highest predication calculation. This will modularized within the native db against the traffic pattern data. The ability to accurately predict the user to event time is important. As it can be seen in Figure x the processing methodology is important because it can change the time to event within minutes based on the right strategy. This accuracy will relate to customer satisfaction on time to event metrics.

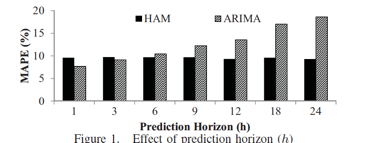


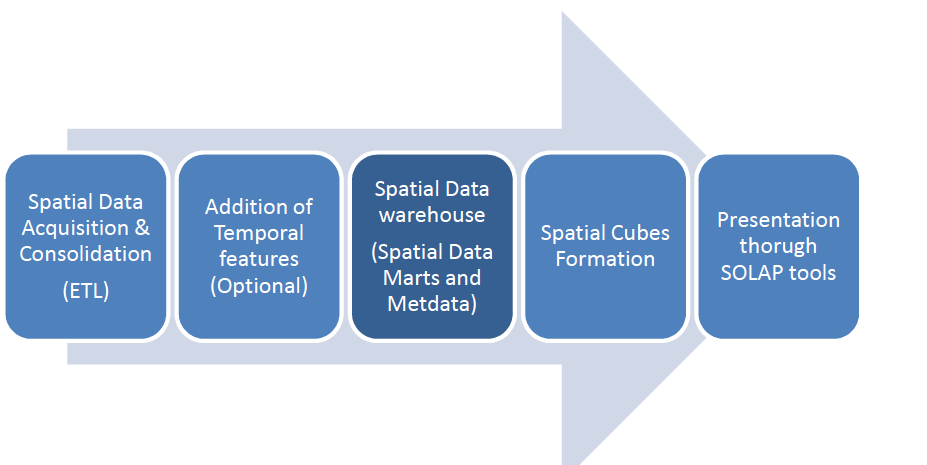
Figure X – Utilizing Real World Transportation Data for Accurate Traffic Prediction. Demiryirek,U. (2013)

### Robust Real Time scale joining of continuous data streams

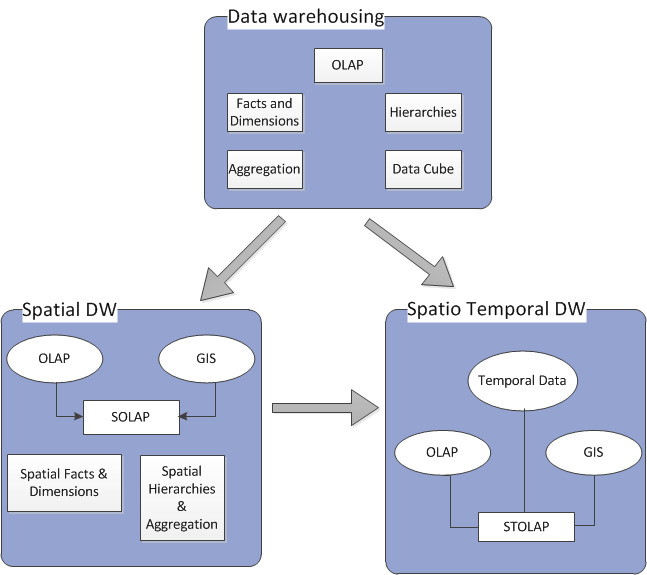
#### **Database Design** : **SOLAP - GIS – OLAP**

*"Spatial OLAP can be defined as a visual platform built especially to support rapid and easy spatiotemporal analysis and exploration of data following a multidimensional approach comprised of aggregation levels available in cartographic displays as well as in tabular and diagram displays."* Bédard, 1997.

OLAP is an approach to swiftly answer multi-dimensional analytical (MDA) queries . It is a category of decision-support tools often used to provide access in an efficient and intuitive manner to a data warehouse. Some of the examples include Cognos Powerplay, Business Objects and Oracle Express. OLAP tools are not robust to analyze spatial and temporal data. GIS tools are also helpful in analyzing spatial data but still are not good enough to make full utilization of spatio temporal datasets. Therefore, a new approach is to couple of OLAP and GIS functionalities. In this way it will be possible to have decision support tools that are bette adapted for spatio temporal exploration and analysis of data. These are called Spatial OLAP systems, or SOLAP.



**Figure** : High level view of the various phases in implementation of a spatial data warehouse



#### Figure : The relationship between spatial, spatiotemporal and a data warehouse.

**Fault Tolerance Management System:**

Our application can also experience failures during real time scale joining of continuos data streams. Failures are common where query operators are distributed across independent processes running on different physical machines in local area network, wide area network or cloud network. These failures can affect our application relying on these queries at the fault location. Our fault tolerance management system can handle queries using different techniques offering various levels of availability as per the application needs.

Our application continuously process data as soon as it becomes available and uses it in multi traffic marker integration coupled with event schedule of users. To support this, our data stream management system is equipped with techniques to handle both node and network failures. Generally, a system replicates the state of its fault transactions on to various nodes which should then coordinate and do a immediate recovery for successful functioning of the application. Two general approaches are State Machine Approach and Rollback Recovery for replication and coordination.

**State Machine Approach:** Assume our fault tolerance techniques can tolerate up to a pre-defined number of k simultaneous failures. In this approach, replicates the computation on k+1>=2 independent nodes and coordinates the replicates by sending the same put in the same order to all these nodes. In case of no fault nodes, all produce the same output result in the same order. In case of a fault , maximum number of nodes differ the result from the output of a single or multiple nodes. The maximum alike output is probably said to be a non-faulty execution and the output result differing from the output of maximum nodes is said to be faulty and the node is not used , restarted and recovered . If there are any queries depending on the current faulty node, the query extractors are re-assigned to other replica nodes to avoid erroneous results.

**Rollback recovery:** In this approach, our application system periodically packages the computation state into a checkpoint copies it on a log, disk or an independent node. The system maintains all the logs in the memory and all the computation input is logged between check points. Once a failure occurs, system refers to the previous checkpoint and rollbacks the transaction to the previous checkpoint. For recovery, it does the execution starting from the previous checkpoint as stored in the memory.

Two types of replicas for handling crash failures are Passive Replication and Active Replication. This involves storing copies of data at multiple node locations and benefits being performance enhancement, increased availability and fault tolerance.

**Passive Replication:** In this model, our client interact with a single primary replica manager. The primary replica manager sends updates to secondary replica managers after responding to requests. If the primary replica manager fails, the secondary replica manager takes its place. The flow of events continues as first the front end issues the request to our primary replica manager which processes them atomically in the order it received and pre-checks for the identifier if it has already processed the request. In case if the request is already processed then it simply sends the stored response or else it executes the oncoming requests and stores the response. If the request is an update or new request , the primary replica sends the updated state and unique identifier to all the secondary replicas or backups. The primary responds to the front end which hands the response back to the client and in case of primary replica failure the secondary replica takes over this as all replicas are supposed to have same record of updates.

**Active Replication:**

In this model of replication, front end issues requests to all replicas. The flow of event continues as the front end assigns a unique identifier to each request and multicasts it to the group of all replica managers. The group of replica managers communication system delivers the request to every correct replica manager in the same order and every manager executes the request. Since all requests are sent in the same order to all replicas and since they are state machines all replica managers should process the requests identically and store a clients Unique request identifiers against its response. Now each replica manager sends its response to the front end. Based on the goals whether to tolerate only crash failures or byzantine failures , the front end waits for number of responses accordingly. If the goal is to tolerate crash failure, then the front end passes the first response to the client and once received it discards the rest provided all the responses at the replica manager are of the same output. In case of byzantine failures , to tolerate f failures the front end waits until it collects f+1 identical responses out of a total of 2f+1 replicas.

**Consumer Security :**

Consumer security in the era of the mobile platform is of pivotal concern. This scrutiny is dependent on geographical location and the laws and guidelines provided by the governments of concern. One aspect of the application is that it will be deployed in the USA and will be subjected to the laws that govern this region.

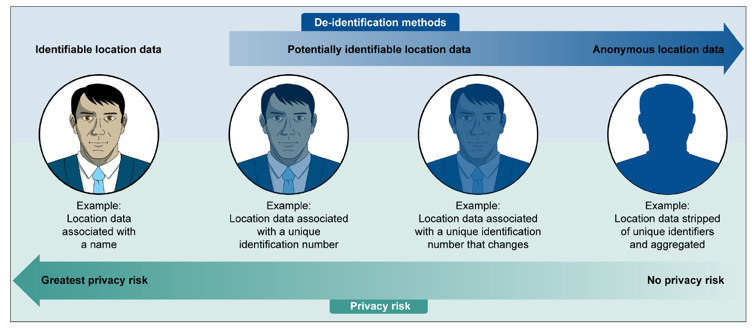
The FTC is taking a role in identifying how a consumer should be directed to the privacy of their information by requiring applications that use geo-location data to required an affirmative consent from the users to allow them to use the data. The guidance is focused on the transparency to consumers to allow informed consent regarding the collection of their information (Pillsbury, 2014). While this type of tracking is only focused on B&M locations it plays out in the mobile traffic arena as well.

The judiciary senate of the USA has put together several recommendations and guidelines to help protect the consumer when it comes to the privacy concerns of data being transmitted to 3rd party companies. The concern here is that the consumer may be unable to judge whether location data shared with these sites are trustworthy. Another aspect is that over time individual behavior can be derived from this information and potentially exploited. Included in review of the legal concerns it the National telecommunications and Information Administration (NTIA) and the Federal Trade Commission (FTA).

The application will ensure consumer security by these guidelines:

* Privacy Policies
* Terms of Service Agreements
* On Screen Notification
* User Controlled methods to end data collection
* User requested deletion of data

One of the main concerns with the data collection methodology is the ability link the information to a specific individual identifiable through personal identifiable information (PII). An example is shown in Figure X which highlights how an individual can be identified and then linked to the data set, vs. anonymized data in which no personal identifiable information can be obtained.

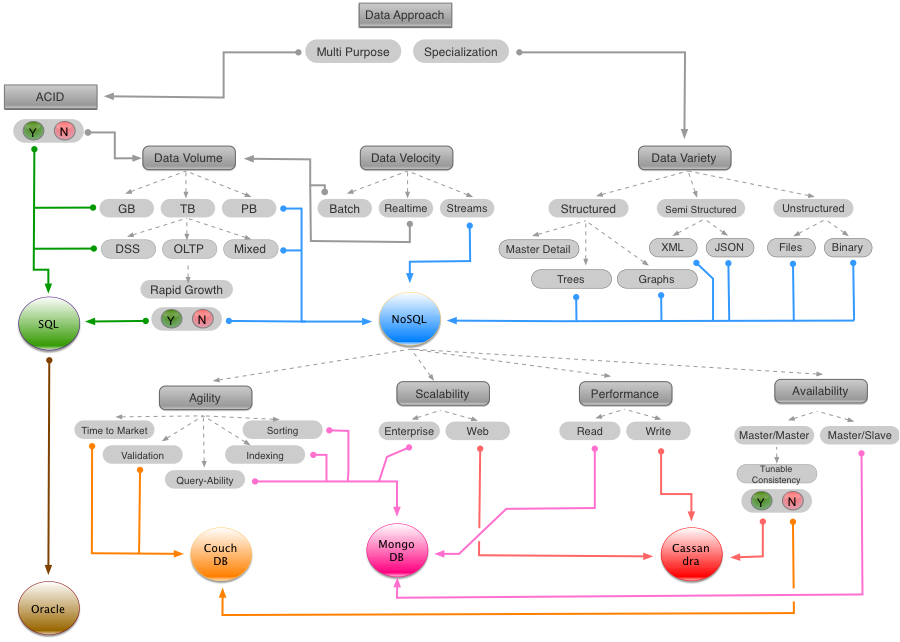
*Figure X – (Consumer Location Data, 2014)*

### Architectural Approach

When developing any system it is important to review the landscape that is available and the technologies that are pressing the envelop to determine if there is any potential points for expansion. In the software development industry there are the “trends” and the meaningful innovations. It is critical to be able to distill this information down into a concise landscape in order to make intelligent decisions on the direction of overall application architecture. For this project the team distilled the technologies available today into a landscape map and by using this determined the direction of the database logic for processing and presenting the information.

Concerns for the architecture design of the processing backbone of the application reside in how the information will be stored and managed. Which data will be used for real time processing, and which data can be moved to secondary systems for analysis and design augmentation. The delta between processing and storing data is not only a design question but one of economics. As a baseline it is expected for US customers that there will be on average of 10 GB of data collected daily for analysis. This is based on crowd-sourced mobile data collection study (Bao,X, 2014) which highlights the scalable data collection methodology of crowdsourcing platforms.

With the database storage requirements identified the structure for the application support needs to be reviewed against the data information coming into the system and the type of real time analytics that need to be driven across it. The system landscape diagram below is referenced in order to determine the processing and reporting backbone (Figure x).



*Figure X – Data Landscape Review Model*

#### Data Capture and Processing

The other concern with the working architecture will be the transactions per second. Currently the process is scoped at 1000 subscribers per the beta testing. The outline of the processing is highlighted in Table 2 (all values average).

Table 2

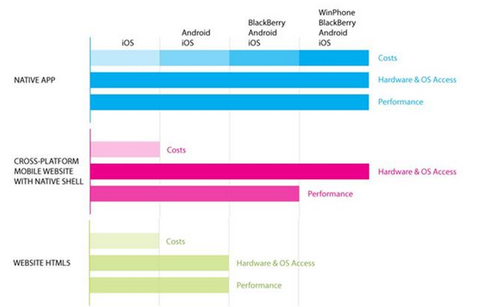
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customers | Transactions per second | Storage per month (Hot) | Storage History | Response SLA |
| (Cold) |
| 1000 | 100 | 10 TB | 50TB | 200-300ms |

The data requirement will be customer data information and data that is combined with the mapping information provider. This composite will be integrated into the device of the customer through a native UI.

Native app: Mobile apps built for specific platforms like iOS, Android, Blackberry, Windows, etc, using their respective development tools and language. Native App will work only on the platform it has been designed for.

HTML5 Web App: Mobile optimized websites that appear to be apps and run on the mobile browser. These are platform independent but lack superior UX and functionality. These apps wont run in the offline mode.   
Hybrid/ Cross- Platform App:

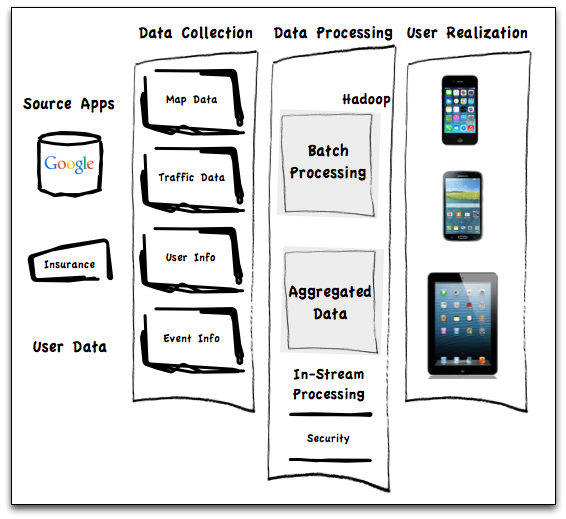
Single mobile app built on HTML5 and optimized for multiple platforms (using native containers). They are downloaded from the app stores like native apps. 

source: quora: blogs

GUI Integration :

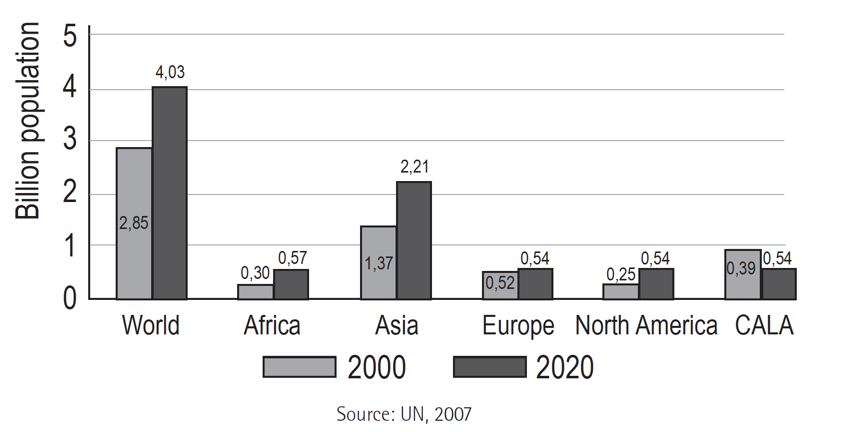
In order to design successfully one thing that will need to be done is to work out the data model for the application. The key to success here is to keep the data model simple and sketch out the working schema without going into much detail until working architecture is identified and then determine an integration strategy.

The other concerns with data capture are any abuse of service. One way to maximize usability is to ensure the system has an accurate way to identify valid requests and a throttle any burst in incoming requests that could deadlock the application. This will ensure that any fraudsters or DoS attacks can be blocked without impacting customer experience.

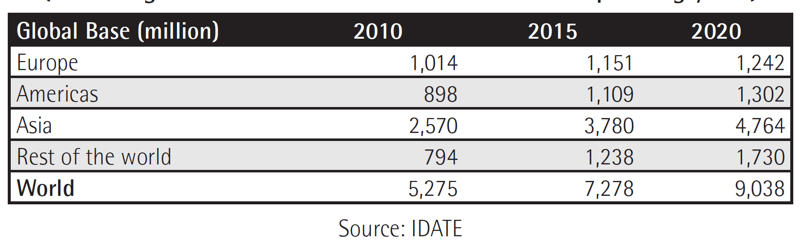


*Figure X – GUI Integration*

Wrap Up :  
  
  
 In the application development cycle one thing that is of importance is the understanding the critical nature of mobile traffic information going forward. It is estimated that in the future the urban population will increase significantly. This increase will only put more demand on the current traffic modeling systems as people try to find faster and more effective means of moving from one location to another. These mega cities will offer challenges for the modeling teams that attempt to derive meaning from the collection of information gathered by the users and the static data capture devices (UMTS Forum, 2011 Jan).



With these advances more users of mobile devices will continue to onboard to the platform and this in effect will increase the overall mobile density of the data collection. This increase will reflect itself in the amount of data stored and captured for real time application support and ongoing model improvement.



References:

Bao,X.,Tapia,M.,Welbourne,E.,Wu,P., (2014) Crowdsourced Mobile Data Collection: Lessons Learned from a New Study Methodology. (Abstract) Retrieved from : http://evan-welbourne.com/CSDC-HotMobile-2014-camera.pdf

Conway, M. (1967) How do Committees Invent? Retrieved from: <http://www.melconway.com/Home/Conways_Law.html>

Kanaracus, C. (2011, Dec 20th) 10 Biggest ERP Software Failures of 2011. (PC World). Retrieved from : http://www.pcworld.com/article/246647/10\_biggest\_erp\_software\_failures\_of\_2011.html

<https://developers.arcgis.com/java/guide/show-device-location-gps-.htm>

<https://developers.google.com/maps/documentation/javascript/examples/layer-traffic>

<http://spatialolap.scg.ulaval.ca/datastructure.asp>