INTRODUCTION
We are moving from Desktop to “Webtop” in the view that the Web is being used not only as a portal for information, but also for application to application communication. An emerging development paradigm called “Mashups” is an excellent example of this revolution in the Web. In a nutshell, Mashups are a kind of application that take information from many places and mix it together. Mashups gives the ability to take photos, RSS feeds, and many other kinds of information and combine them to create your own personalized view of the web. Put together, Mashups are an enterprising breed of synergistic web applications that pull up content from external data sources to create entirely new and innovative services. They are the characteristic of the second generation of web applications colloquially known as Web 2.0. The review of this groundbreaking technology highlights the development of Mashups as an emerging epitome of the success of Web 2.0 and future that is Web 3.0.

MOTIVATION
Mashups play an important role in integrating business and data services, as mashup technologies give the capability to create new unified services rapidly and merge internal services with external or individualized information, in order to make these services concrete to the business user through a set of user interfaces.

The general benefits of innovation are shared in a wide fashion. Each Web service or mashup may have new features and abilities that distinguish it from others previously available. Thus, individual users draw the profits of this interoperability characteristic of a Mashup, every time they use it. Furthermore, public service Mashups almost innately give benefits to the society altogether. For example, enabling a level of political clarity in the fields of media and telecommunications in the US that otherwise would be extremely costly if not impossible. Having distributed development of new applications through practical Web services also nurtures a more interactive cultivation of user-driven innovation. This lies on the very fact that can be drawn from this discussion that users are not just the passive consumers or reviewers of
Web 2.0, but also potential developers possessing with the very code that regulates the opportunities available on the Internet and thus to push the limits of online culture. In economic terms, Mashups could not only stimulate current market players to innovate in ways that are open to further discovery, but also break down the barriers so that the potential industry of the Internet is greatly expanded to include individual users as well. By making it easier to innovate, open APIs and similar technologies could create not only innovation, but innovators.

THE ARCHITECTURE BEHIND “LIVE” MASHUPS

In the world of mashable API, a different approach needs to be taken as opposed to the traditional object remoting method used in various Java and .NET applications. With XML becoming the data structure standard for integration, XML HTTP APIs will seemingly look like an obvious alternative to object remoting. There has been debate about object serialization and deserialization (i.e. translating an object into XML messages and vice versa) impacts on performances, but this won’t be discussed here. For lightweight web applications that are based on widgets or Mashup components, XML over HTTP is definitely a better choice.

The interfaces can be as simple as invoking SOAP based Web Service, or a REST based web service. Information syndication technology such as RSS and ATOM can also be the source point of data. Normally Web 2.0 elements such as Blogs and Wiki provide this sort of syndication to allow content to be accessed outside of the website. Access to structural data stored within databases can be accessed via various well defined APIs for security reasons. The response from the SOA services can be in the format of XML or JSON.

The general architecture of a mashup application has three levels or tiers. The services or data sources used in the mashup constitute the first level. This may include one or more resources depending on the application requirements, and one common data source is the actual user base that enters data directly in the application. The next tier is the actual mashup application that brings together these resources into a hybrid. It may include its own functionality or simply present the data sources. The final piece of the puzzle is the user client used to access the site. Client-side scripting may be utilized to enhance the user experience. The following diagram illustrates this:
“LIVE " MASHUPS

MASHUP #1: Location based Live Feeds

In this mashup, we will be developing an application that will take advantage of GeoRSS data format to display live feeds for a user on his blog. The implementation will consist of a mashup built over Virtual Earth that will show live feeds of a user based on the locations where he travels to during his trips/visits. For example, a person travelling to different parts of the world wants to update his blog with feeds showing the motive of his visit to that location or the some event/conference that is been conducted at that location. This will help the user show a trail of his trip on the map along with any information that he wants to post for all the possible locations of his trip. In order to achieve this, all the user has to do is write some comments/text...
about his visit to a location (while he is currently present at that location) and the mashup will automatically get his location information and display a pushpin on the map indicating his presence along with the comments/text that the user wishes to be displayed on the map. Hence, the mashup keeps creating a trail of locations as the user keeps visiting various places around the world and updates the map accordingly as live feeds for others visiting the blog to view.

**MASHUP #2: Live Weather Information**

In this mashup, we will be developing an application that will combine 2 sources: Weather services and Virtual Earth to create a weather portal using the APIs from these 2 sources. The mashup will allow users to view the weather of any place in the world without knowing its zip code. The entire world map will be shown in the form of heat map, where regions with high temperatures will be more on the reddish shade. Also places with snow and rain can be highlighted with respective images overlaid on the map. The user will be able to zoom in to get detailed weather report. Also when the user hovers over the map the weather of the place can be shown as the tool tip. The user can also mark places of his interest. The weather for these places can always be superimposed on the map.

**MASHUP #3: Visitors on Map**

The purpose of this mashup is for marketing entities and businesses to determine the popularity of their website based on hits they get from different locations. Users access web services and websites from different regions around the world. We plan to show this information on a Virtual Earth map. Areas on the map will be highlighted as users hit the website. If a particular area is not shown to be popular but the business believes there is potential in it, it can be targeted for aggressive marketing campaigns. On the other hand, if a company or business knows where the majority of the hits are coming from, they can target those areas for a sustained marketing effort in order to keep the user base growing. This mashup is an example of how a company can efficiently use location-based information and data to improve its productivity in different parts around the world. This mashup can be linked to the other mashups in our application. As an example, a company planning a field marketing effort can check their schedules against the integrated weather mashup to see potential disruptions. This can save a lot of resources and help a company effectively plan out their marketing strategies. Also, if the company serves individual users with their products, like a bank, then this mashup can be linked with the location tracking mashup. A subscribed user will then get information on the closest branch or kiosk of the business or company, as his locations are tracked on Virtual Earth.
ENVIRONMENT
Our system comprises of 3 Mashups which will be developed in the following environment and technologies:

- Windows Vista with .NET 3.5 Framework
- Ajax Control Toolkit 3.5, C#, Ajax
- Microsoft Virtual Earth API
- XML, JSON and GeoRSS Data formats
- Weather web services like Weather.com

PLAN OF ACTION

| Week 1: Research on GeoRSS, JSON, and Web Services |
| Week 2: Design Mashup #1 |
| Week 3: Design Mashup #2 |
| Week 4: Design Mashup #3 |
| Week 5: Implement Mashup #1 |
| Week 6: Implement Mashup #2 |
| Week 7: Implement Mashup #3 |
| Week 8: Testing and Evaluation |
| Week 9: Demonstration |

DELIVERABLES
Deliverables include the following:

1. Working implementation of all 3 Mashups
2. In-class demonstration
3. Final Project Report
REFERENCES