



中山大學
SUN YAT-SEN UNIVERSITY

Lecture 24

Mashups

SE-805 Web 2.0 Programming (supported by Google)

<http://my.ss.sysu.edu.cn/courses/web2.0/>

School of Software, Sun Yat-sen University

Outline

- **Mashup basics**
- Web APIs
- Google APIs

Mashups = Remixed Data

- A **mashup** is a Web page or application that uses and combines data, presentation or functionality from two or more sources to create new services.

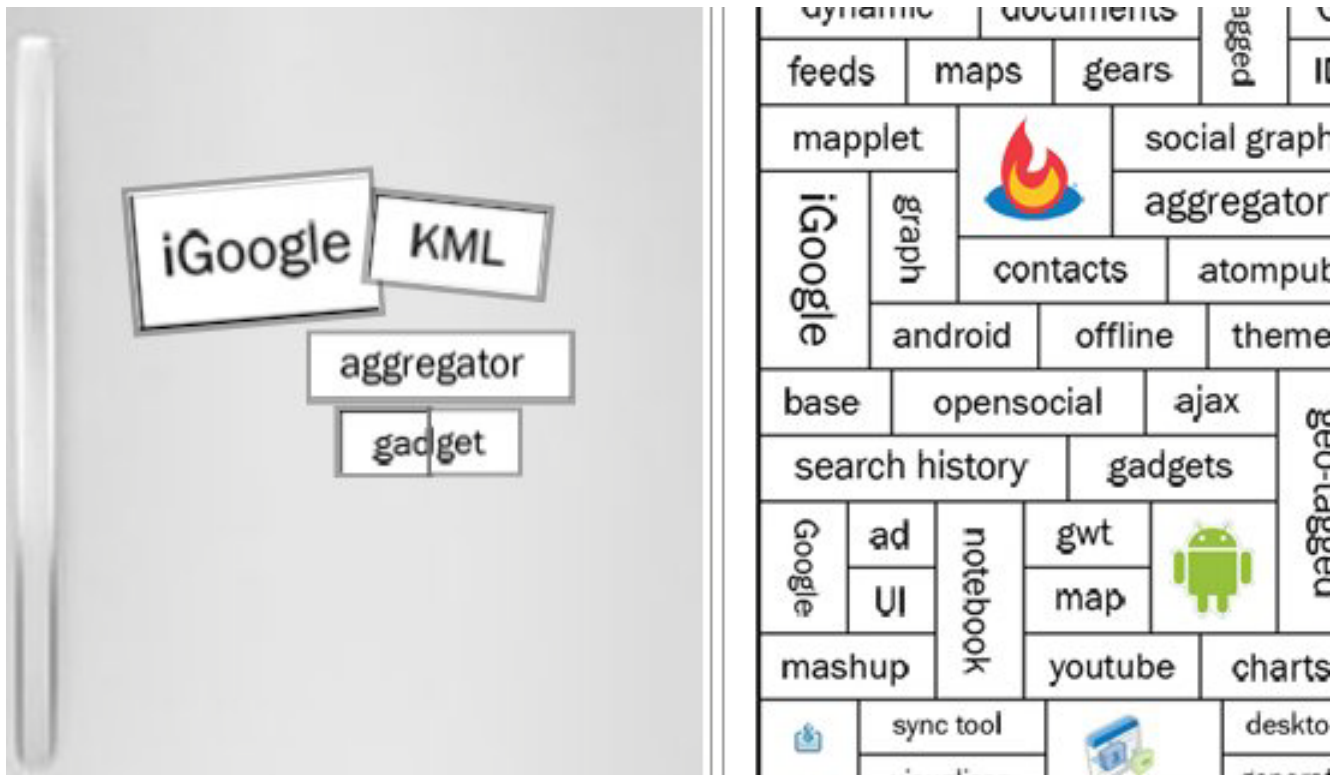
Data + APIs

Data + Other Data

Data + Functionality

Mashups: Why?

- "We know we don't have a corner on creativity. There are **creative people** all around the world, hundreds of millions of them, and they are going to think of things to do with our basic platform that we didn't think of."
- – [Vint Cerf](#)

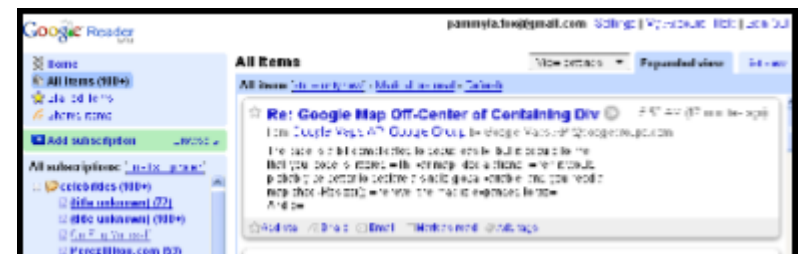


Mashup Type: Aggregators

- Combine feeds/info from various related websites into one site.
- More info + less clicks = happier websurfer

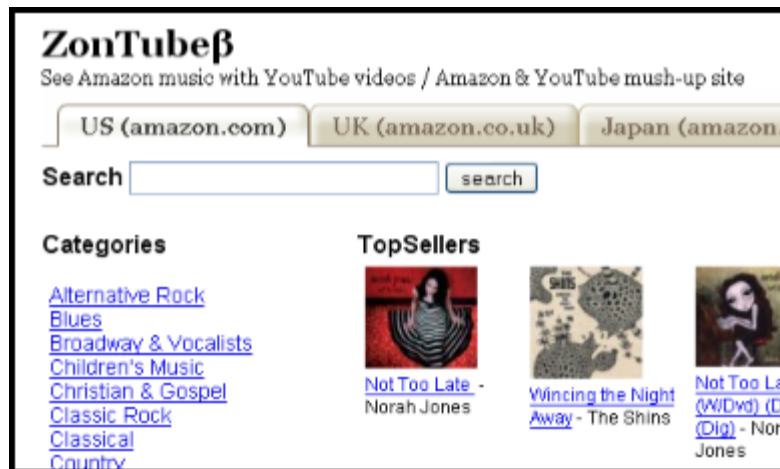


- Examples: <http://www.originalsignal.com>, <http://doggdot.us>, <http://reader.google.com>



Mashup Types: Search/Search Aggregators

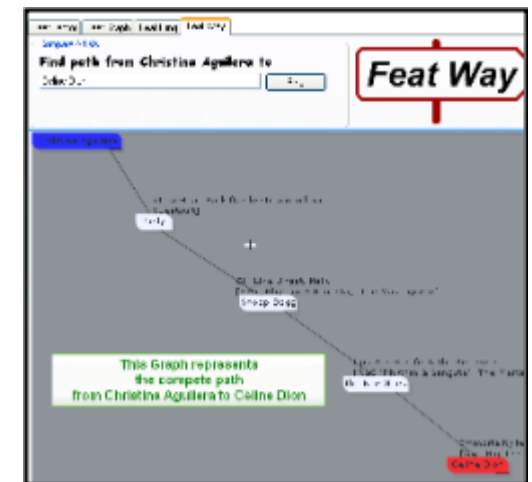
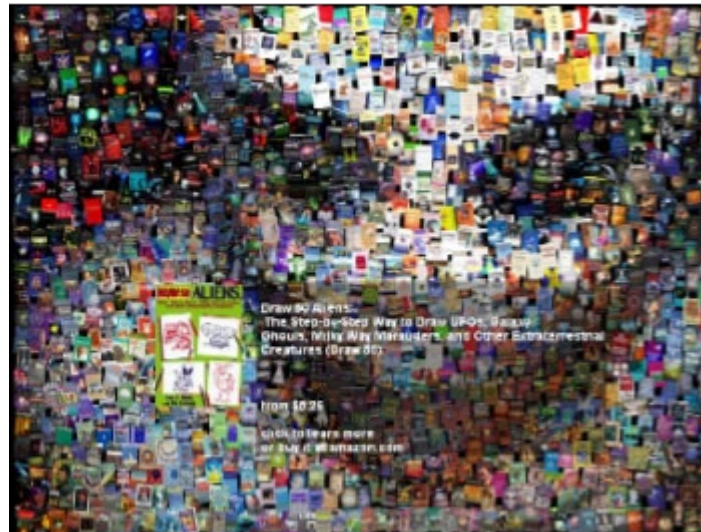
- Search: Let you search an API's data
- Search aggregators: Let you search once, find info from many search engines/APIs at once.



- Examples: <http://pulp site.net/zontube/>

Mashup Types: Visualizers

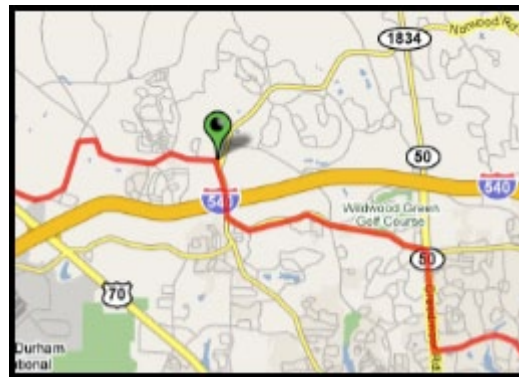
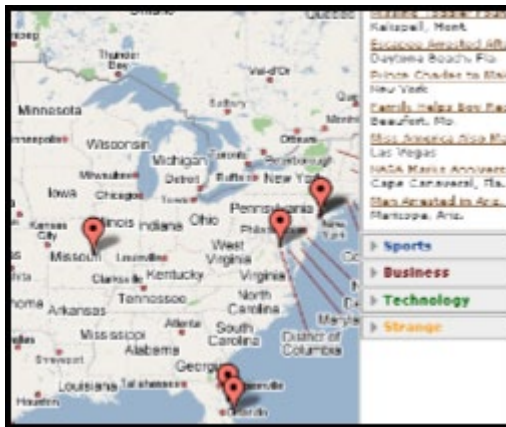
- Visualizers: Take related data and visualize in a new and meaningful way (e.g. clouds, maps)



- Examples: <http://imagine-it.org/amazong/vissimweb.htm>, <http://www.coverpop.com>, <http://mathias.cianci.free.fr/>

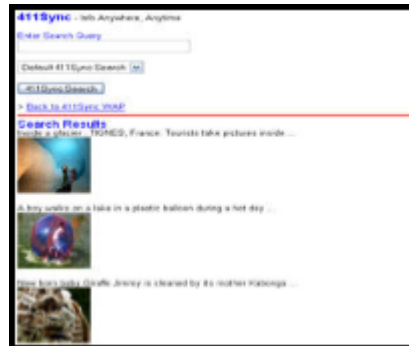
Mashup Types: Maps/Earth

- Take data with geo info from other sources (including users!) and plot on map
- Geographical info could be latitude/longitude, could just be an address, city or place name as geocoding APIs are quite common



- Examples: <http://www.81nassau.com/apnews/>, <http://www.bikely.com>, <http://imagine-it.org/mashplanet/>

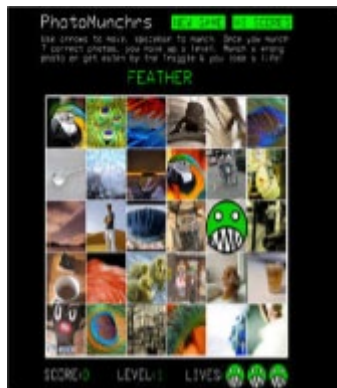
- Take online data from APIs/feeds and put in mobile-friendly format
- Increasingly needed as many fancy web2.0 websites just won't work on cell phones (AJAX) but people still want their information, and quick!



- Examples: http://www.411sync.com/cgi-bin/search_api?query=mydigg+txttester, http://www.411sync.com/cgi-bin/search_api?query=daily, <http://www.frucall.com>

Mashup Types: Games

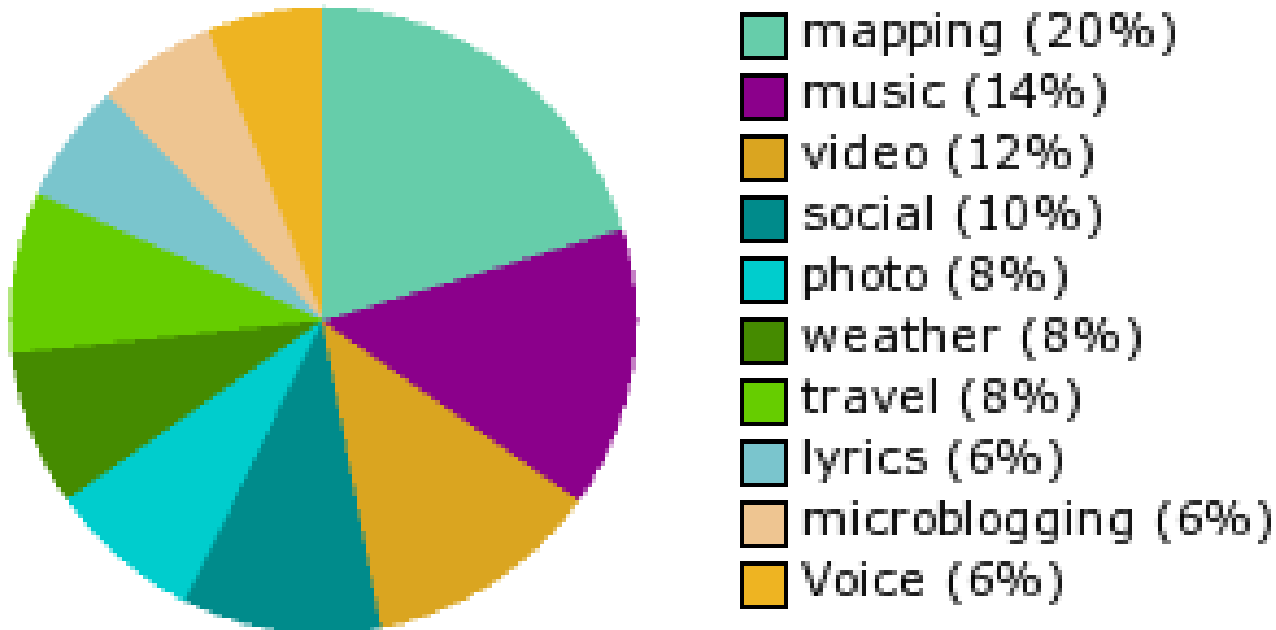
- Make players guess more data about a web object (photo, friend, map clue)
- My personal favorite type ☺



- Examples here: <http://imagine-it.org/flickr/PhotoMunchrs.html>, <http://imagine-it.org/google/wordhunter.htm>, <http://imagine-it.org/flickr/flicktionary.htm>, <http://www.facebook.com>

Mashup Types: Other!

- There are tons of other ways of mashing up data and APIs.
- Browse <http://programmableweb.com> to see others.



ProgrammableWeb.com 08/04/08

Outline

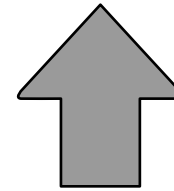
- Mashup basics
- **Web APIs**
- Google APIs

Definition: API ... Web API

Application
Programming
Interface

Web APIs =

- APIs use **http(s)** as transportation



API Types

HTTP
REST | RPC

Visual

Plugin

API Types: HTTP :: RPC

fooInstance->addNumbers(2, 3);



```
<?xml version="1.0"?>
<methodCall>
<methodName>Foo.addNumbers</methodName>
<params>
<param><value><int>2</int></value></param>
<param><value><int>3</int></value></param>
</params>
</methodCall>
```



fooInstance.addNumbers(2, 3);

PHP

XML
(Network)

C++

API Types: HTTP :: RPC

<http://api.flickr.com/services/rest/?method=flickr.photos.search&text=pamela+fox>

```
<rsp stat="ok">
<photos page="1" pages="2" perpage="100" total="159">
<photo id="3461223826" owner="37370984@N07"
secret="6d0bbbbfa3" server="3512" farm="4" title="Pamela
Fox - mapping, red dot fever" ispublic="1" isfriend="0"
isfamily="0" />
<photo id="3461224220" owner="37370984@N07"
secret="7365fecf34" server="3605" farm="4" title="Pam pam
pam" ispublic="1" isfriend="0" isfamily="0" />
<photo id="3459126604" owner="44124396772@N01"
secret="c54c15ee4b" server="3608" farm="4" title="pamela"
ispublic="1" isfriend="0" isfamily="0" />
</photos>
</rsp>
```

API Types: HTTP :: SOAP

<http://www.flickr.com/services/rest/?method=flickr.test.echo&format=soap&foo=bar>

```
<?xml version="1.0" encoding="utf-8" ?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-
envelope" xmlns:xsi="http://www.w3.org/1999/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/1999/XMLSchema"
>
<s:Body>
<x:FlickrResponse xmlns:x="urn:flickr">
[escaped-xml-payload]
</x:FlickrResponse>
</s:Body>
</s:Envelope>
```

API Types: HTTP :: REST

Application state and functionality is abstracted into discrete resources.



Resources are accessible via URLs.

/blog/posts/1234

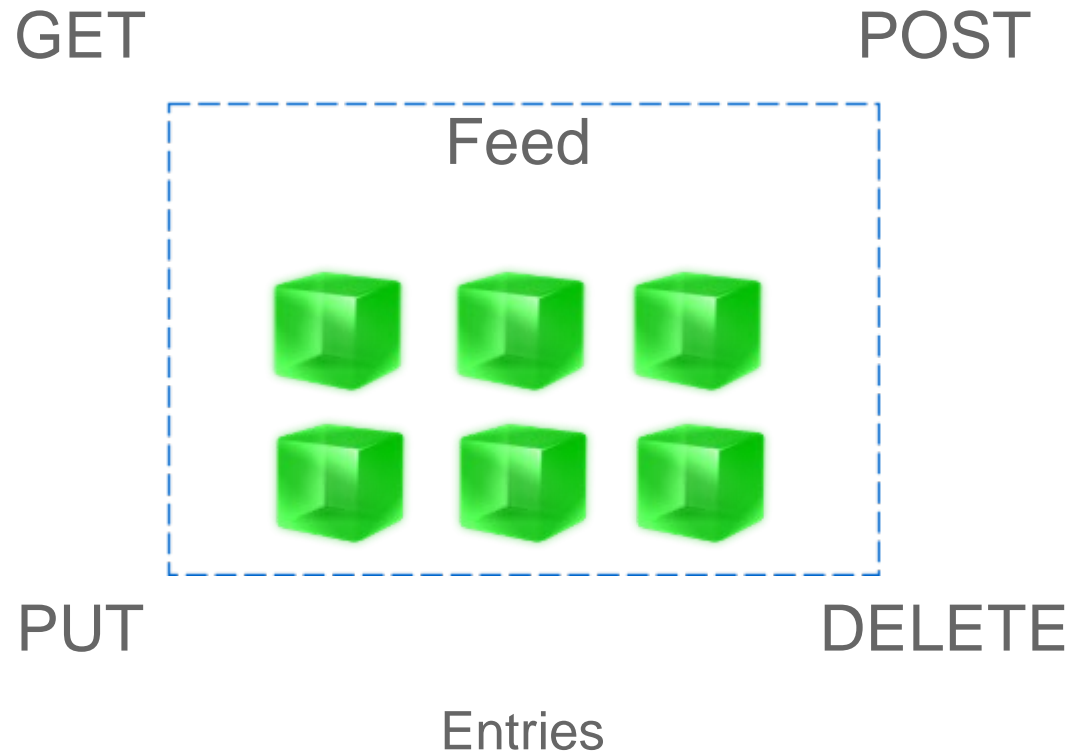
Resources share a uniform interface for transferring state.

HTTP://

GET POST

PUT DELETE

API Types: HTTP :: REST



API Types: HTTP :: REST

**http://api.netflix.com/catalog/titles/series/70023522/
seasons/70023522**

<?xml version="1.0" encoding="utf-8" ?>

<catalog_title>

<id>

http://api.netflix.com/catalog/titles/series/70023522/seasons/70023522

</id>

<title short="The Office: Season 1" regular="The Office: Season 1"/>

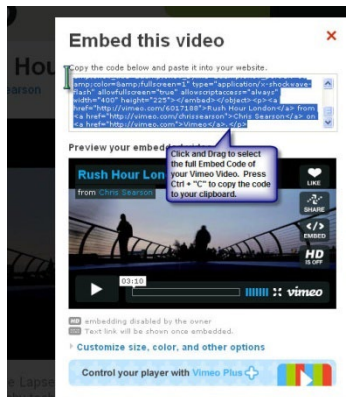
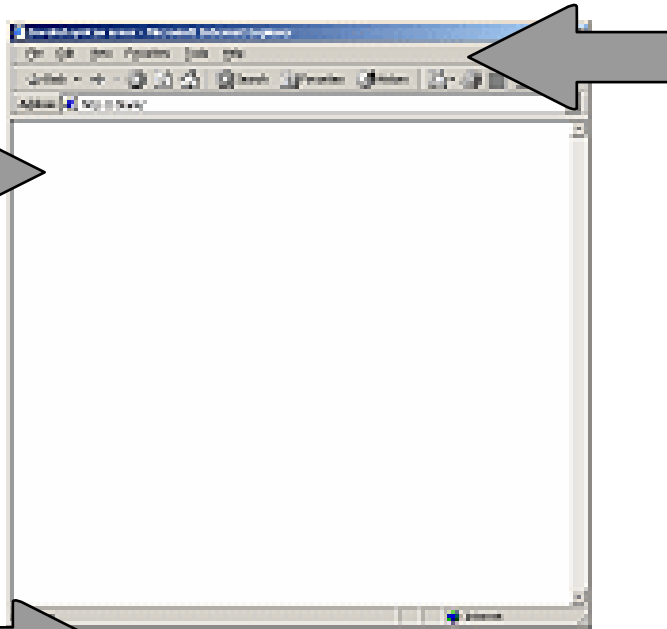
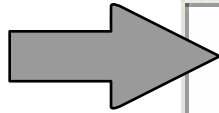
<release_year>2005</release_year>

<runtime>8700</runtime>

...

</catalog_title>

API Types: Visual

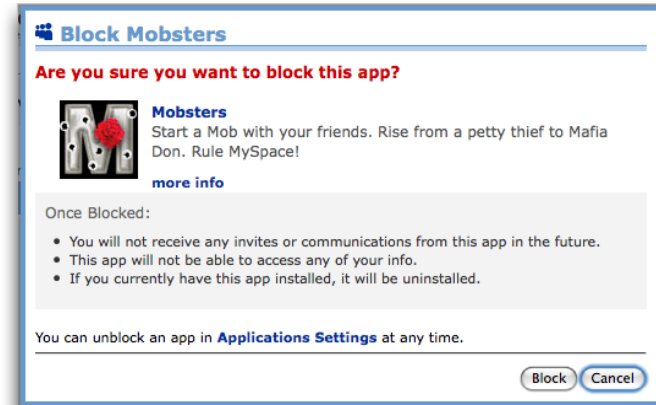


API Types: Visual



```
<script type="text/javascript"
src="http://videocallroom.oovoo.com/oovoorooms.js">
</script>
<script type='text/javascript'>
  roomProps.roomID = '60D56CE75A321CE3E01230144F7E8E22' ;
  roomProps.width = '795' ;
  roomProps.height = '640'
  roomProps.backgroundColor = '#666666' ;
  roomProps.captionText = 'Best room ever' ;
  roomProps.captionColor = '#FFFFFF' ;
  var myRoom = CreateRoom() ;
</script>
```


API Types: Plugins



API Types: Plugins



```
<widget:preferences>
  <preference name="helloworld" type="text" label="Hello who
  ?"
```

```
    defaultValue="World" />
</widget:preferences>
```

```
<title>Title of the Widget</title>
```

```
<script type="text/javascript">
  widget.onLoad = function() {
    var who = widget.getValue('helloworld');
    widget.setBody(' <p>Hello ' + who + '!</p>');
  }
</script>
```

Outline

- Mashup basics
- Web APIs
- Google APIs

Google APIs

HTTP

REST | RPC

Google data APIs

Adwords API

Geocoding API

Visual

Google Maps API

Google Visualization API

Google Charts API

Google Web Elements

Plugin

OpenSocial Gadgets

Spreadsheets Gadgets

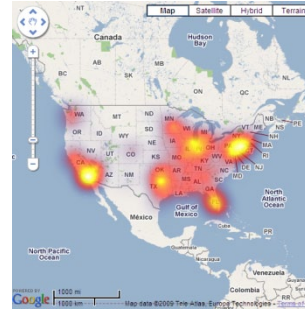
Wave Gadgets/Robots

Google APIs: Google Maps APIs

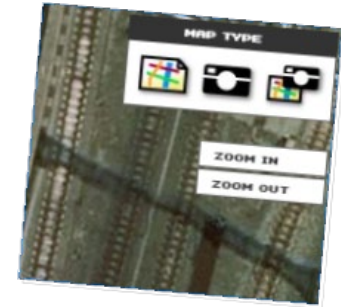
Maps Data API



JS Maps APIs

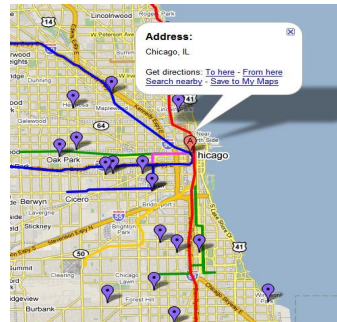


Maps API for Flash



Static Maps APIs

Mapplets



Das Bermuda-Dreieck
Seit Jahrzehnten
beschäftigt das
geheimnisvolle Bermuda ...



EURO 2008 Vom 7. bis 29.
Juni 2008 findet in
Österreich und der S ...



Weltstädte mit Flagge
"Flaggen der Erde" Karte
mit allen Welthaupt ...

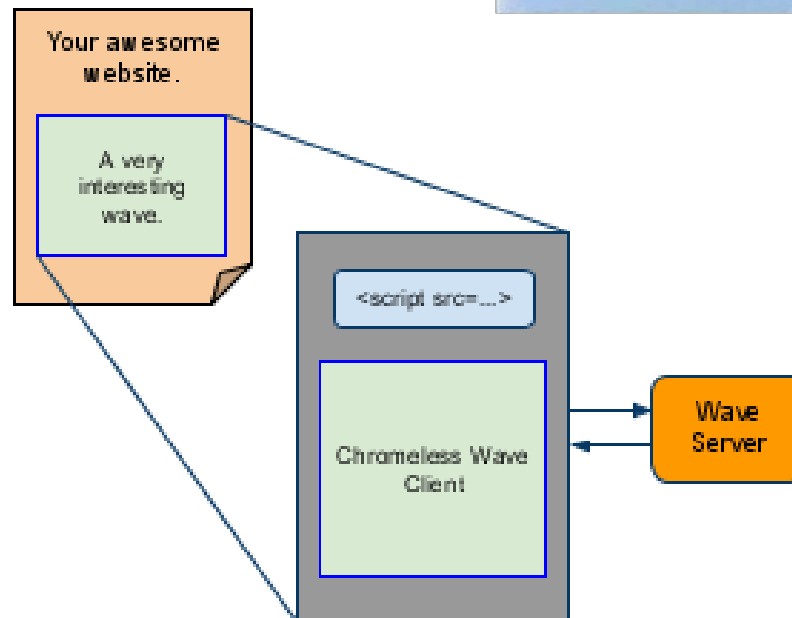
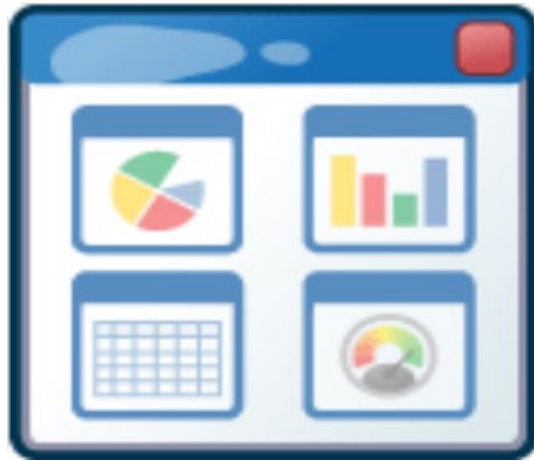


TAPS: Trans Alaska Pl ...
The Trans Alaska Pipeline is
a 800 miles (1,300 km) long

Google APIs: Google Maps APIs

TrendsMap

Google APIs: Google Wave APIs



Google APIs: Google Wave APIs

```
public class MaileyBotServlet extends AbstractRobotServlet {

    public void processEvents(RobotMessageBundle bundle) {
        Wavelet wavelet = bundle.getWavelet();
        sendEmail(wavelet.getTitle());
    }

    public void sendEmail(String title) {
        Message msg = new MimeMessage(session);
        msg.addRecipient(Message.RecipientType.TO,
            new InternetAddress("pamela.fox@gmail.com"));
        msg.setSubject("the wave " + title + " was updated");
        Transport.send(msg);
    }
}
```

Google APIs: Google Wave APIs

Emoticony

Cards

Google APIs: Google Data APIs



Google APIs: Google Docs API

POST /feeds/documents/private/full

```
<entry xmlns="http://www.w3.org/2005/Atom">
  <category scheme="http://schemas.google.com/g/2005#kind"
    term="http://schemas.google.com/sites/2008#folder" label="folder"/>
  <title>New Folder</title>
</entry>
```

Google APIs: Google Docs API

Docs Editor

Summary

- Mashup Basics
 - Definition
 - Type: aggregators, search/search aggregators, visualizers, Maps, Mobile, ...
 - Web APIs
- HTTP::RPC, HTTP::SOAP, HTTP::REST
 - Visual
 - Plugin
- Google APIs
 - Maps, Wave, Data, Docs, ...

Exercises

- Write a simple Google Map application, a Web page shows the map of Sun Yat-sen University
 - get the latitude and longitude our campus via Google Search
 - use Google Maps JavaScript API V3
 - follow steps in [Google Map Javascript API V3 Tutorial](#)

Further Readings

- Mashup (web application hybrid)
http://en.wikipedia.org/wiki/Mashup_%28web_application_hybrid%29
- Programmable Web <http://www.programmableweb.com/>
- Google Ajax APIs <http://code.google.com/apis/ajax/>
- Google Maps API Family
<http://code.google.com/apis/maps/index.html>
- Create your first map
<http://code.google.com/apis/maps/articles/yourfirstmap.html>
- Google Maps API Tutorial <http://econym.org.uk/gmap/>

Thank you!

