Java is a trademark of Sun Microsystems, Inc.

JavaOne℠

Augmented Reality with Java™ Platform, Micro Edition (Java ME Platform) Devices

Kenneth Andersson and Erik Hellman

Sony Ericsson Mobile Communications

We like Twitter!

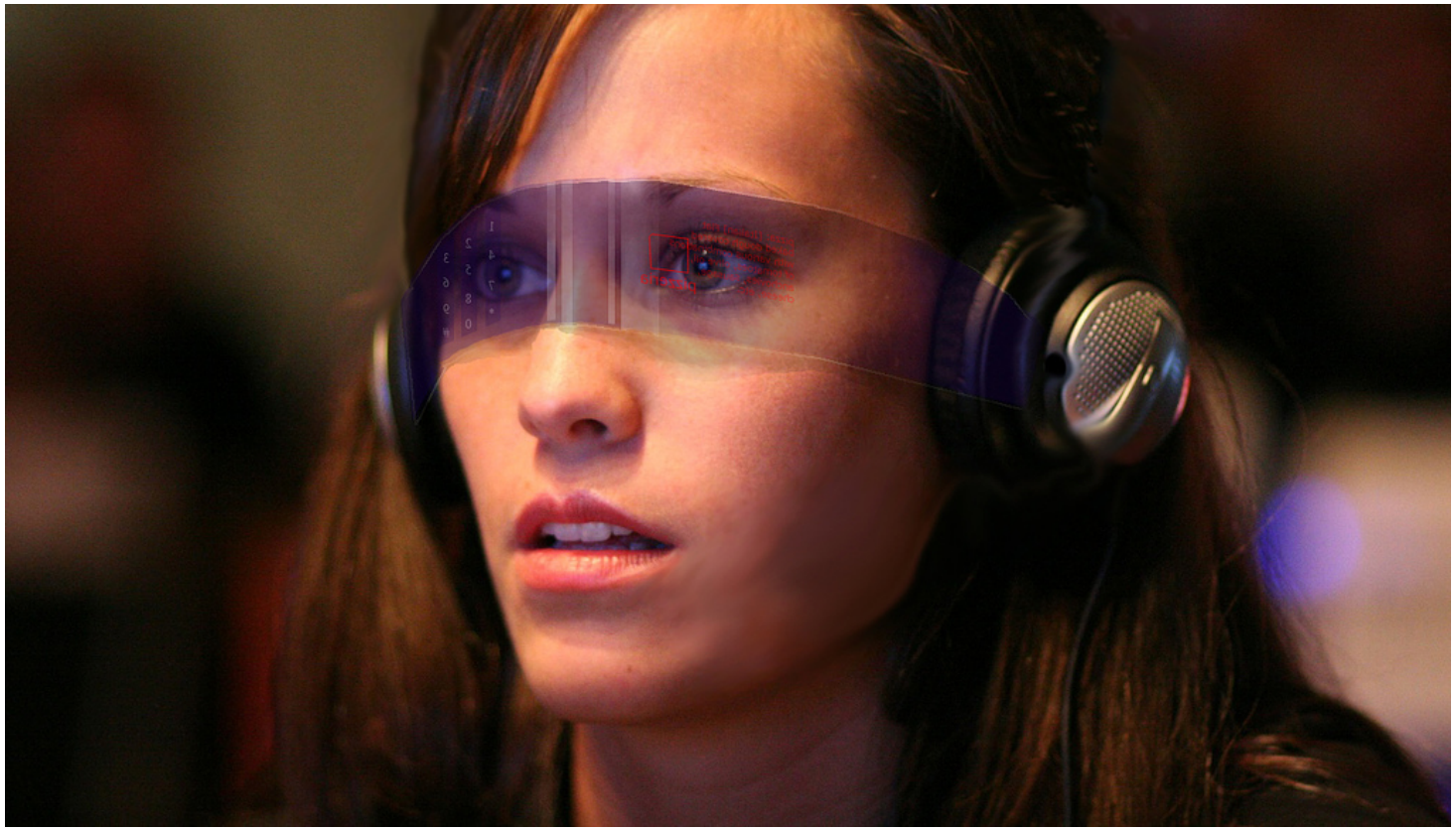
- Follow us on ***@SonyEricssonDev*** on Twitter (http://twitter.com/SonyEricssonDev)
- Use ***#SEMC*** Twitter tag to ask questions or comment on the session.
- We will follow the public feed and try to collect your questions and feedback in real-time. ☺

http://wordle.net/

Frustrated about your Java ME development?

http://www.flickr.com/photos/thefangmonster/490423135/

Try Augmented Reality!



http://www.flickr.com/photos/leonardlow/310039863/

Camera: Augmentation of vision

GPS and Accelerometer: Augmentation of orientation, direction and location

Screen, speaker and vibration: Augmented reality feedback

Bluetooth, WiFi and GPRS/3G: Augmentation of reality (networking)

Microphone: Augmentation of hearing

But really?

http://www.flickr.com/photos/rsdio/490904866/

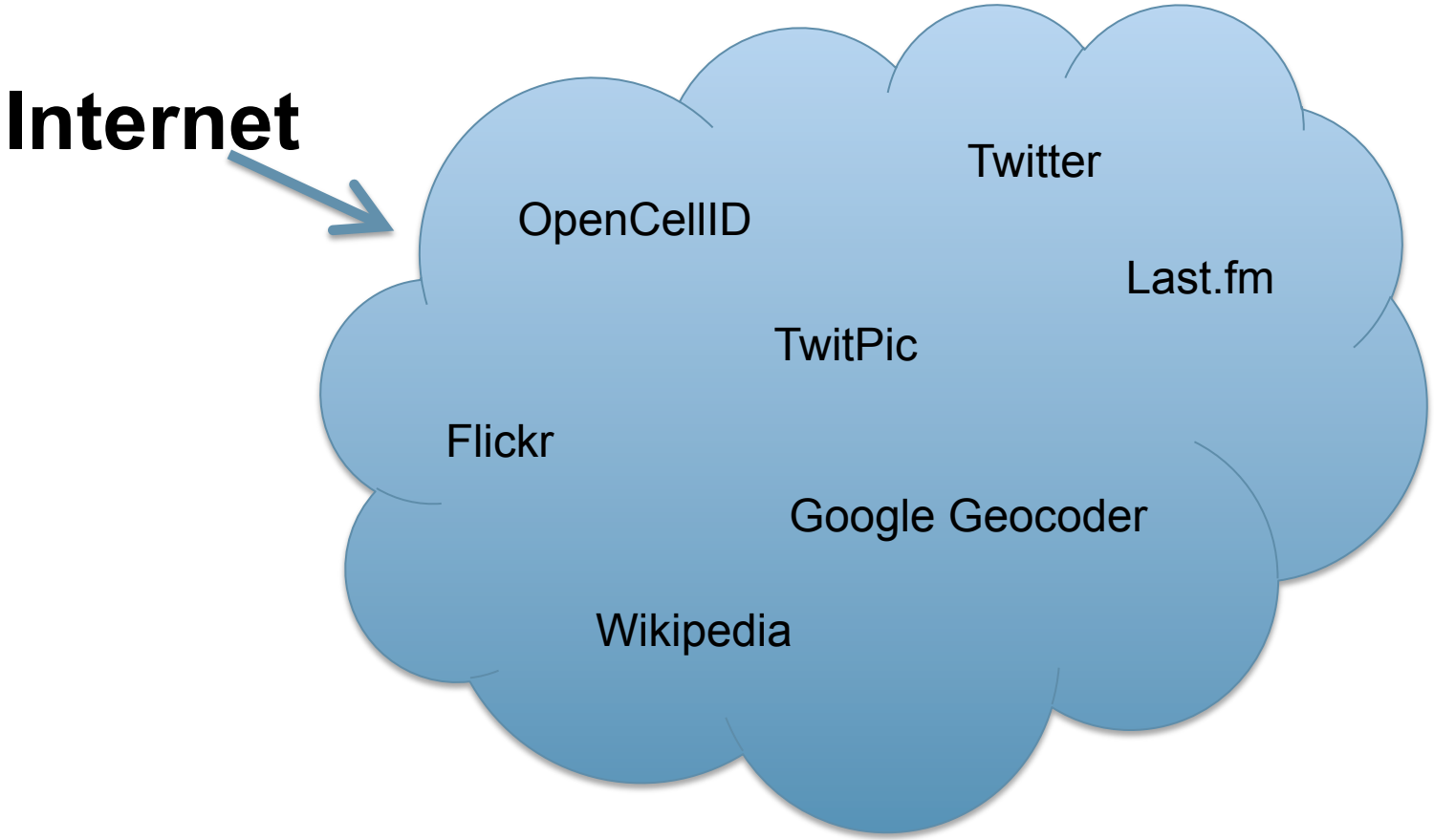
But wait. There's more!

http://www.flickr.com/photos/riverman72/529637320/

Obstacles & Impediments

- Diversified hardware specifications
  - GPS, Accelerometer, Camera, Compass etc.
- Java ME fragmentation
  - JSR availability, bugs,
- Java ME security model
  - Limited access for third-party developers
- Limited connectivity
  - Network coverage, GPS coverage etc.
- Data-traffic costs and roaming

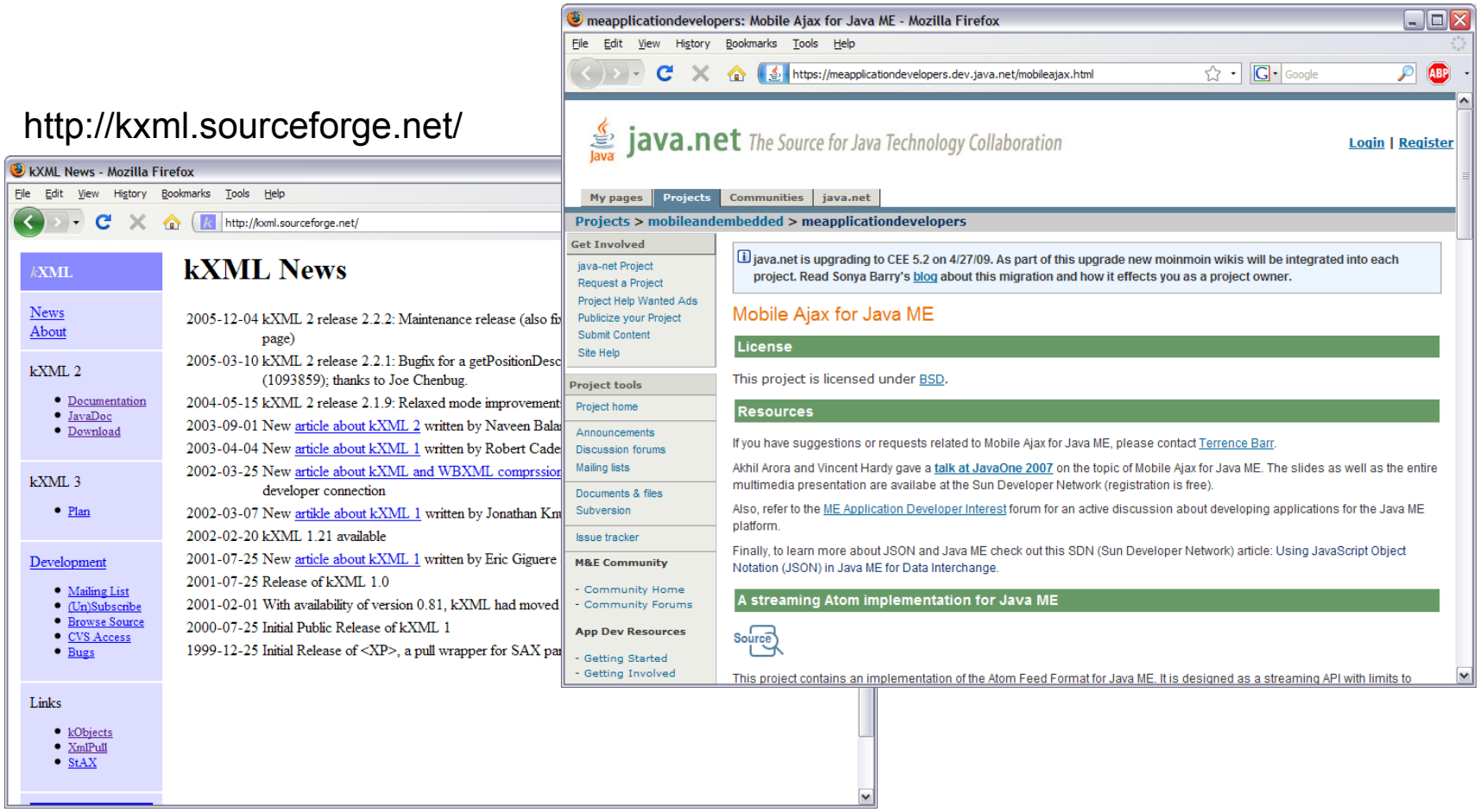
Online Services



3rd party Java ME APIs

https://meapplicationdevelopers.dev.java.net/mobileajax.html

http://kxml.sourceforge.net/



Location search

- Provide a simple interface for searching based on your current location
  - Generic - Google
  - Specific - Wikipedia
  - Category – Last.fm

Code examples (Location API, JSR-179)

```
// init the LocationProvider with default criteria
locationProvider = LocationProvider.getInstance(new Criteria());
// Get the currently last known location
currentLocation = LocationProvider.getLastKnownLocation();
// Listen for updates with default parameters
locationProvider.setLocationListener(this, -1, -1, -1);

// Callback function for location updates
public void locationUpdated(LocationProvider locationProvider,
   Location location) {
    if (location != null &&
        location.getQualifiedCoordinates() != null &&
        location.isValid()) {
        currentLocation = location;
    }
}
```

No GPS - Open Cell ID

```
// System properties for cell id etc.
// Sony Ericsson
mCellId = System.getProperty("com.sonyericsson.net.cellid");
mMCC  = System.getProperty("com.sonyericsson.net.cmcc");
mMNC  = System.getProperty("com.sonyericsson.net.cmnc");
mLAC  = System.getProperty("com.sonyericsson.net.lac");

// Similar properties exist for other vendors!
```

Code examples (Open Cell ID)

```
// Call the online search service
String requestUri = "http://www.opencellid.org/cell/get" +
   "?key=" + API_KEY + "&mcc=" + mMCC + "&mnc=" + mMNC +
   "&cellid=" + mCellId + "&lac=" + mLAC;


// XML result
<rsp stat="ok">
<cell mnc="99" lat="57.8240013122559" lac="0"
   lon="28.00119972229" nbSamples="38" range="6000" mcc="250"
   cellId="29513"/>
</rsp>
```

Code examples (Request API, Mobile AJAX)

```
// The URL we're calling
String URL = "http://...";
// The parameters
Arg[] inputArgs = new Arg[] {new Arg("key","value")};
// HTTP Basic authentication string
String authString = "Basic " + BasicAuth.encode(username,password);
// HTTP header parameters
Arg[] httpArgs = new Arg[] {new Arg(Arg.AUTHORIZATION, authString)};
// Call the URL
Request.get(URL, inputArgs, httpArgs, requestListener, context);

// Callback implemented by the requestListener object
public void done(java.lang.Object context, Response response) {
    handleResponse(response);
}
```

Code examples (Google Geocoder)

```
String URL = "http://maps.google.com/maps/geo?"
String PARAMS = "oe=utf8&output=json&sensor=true&ll=";

// Call Google Geocoder to get address for our coordinates
public AddressInfo getAddressInfo(QualifiedCoordinates coords) {
    String latlong = coords.getLatitude()
        + "," + coords.getLongitude();

    Response response = Request.get(URL + PARAMS + latlong,
                                    null, null, null);
    if(response.getCode() == 200) {
        return parseJsonResult(response.getResult());
    } else {
        return null;
    }
}
```

Code examples (JSON API, Mobile AJAX)

```
country = result.getAsString("Placemark[0].AddressDetails.” +
   ”Country.CountryName");
addressInfo.setField(AddressInfo.COUNTRY, country);

countryCode = result.getAsString("Placemark[0].AddressDetails.” +
   ”Country.CountryNameCode");
addressInfo.setField(AddressInfo.COUNTRY_CODE, countryCode);

city = result.getAsString("Placemark[0].AddressDetails.” +
   ”Country.AdministrativeArea.Locality.LocalityName");
addressInfo.setField(AddressInfo.CITY, city);

street = result.getAsString("Placemark[0].AddressDetails.” +
   ”Country.AdministrativeArea.Locality.” +
   ”Thoroughfare.ThoroughfareName");
addressInfo.setField(AddressInfo.STREET, street);
```

Code examples (Google Search)

```
String GOOGLE_URL = "http://www.google.com/m/search?q";

public void searchGoogle(String query, String location) {
    midlet.platformRequest(GOOGLE_URL + query + "+" +
   location);
}

// Example usage
// Retrieve our location using Google Geocoder API +
   Location API
String location = geoCoderClient.getLocationString();
// show a dialog for user asking for a search query
String query = getUserQuery();
// Execute the query using platformRequest()
searchGoogle(query,location);
```

Code examples (Wikipedia Search)

```
// The online MediaWiki API for Wikipedia
String WIKIPEDIA_URL = "http://en.wikipedia.org/w/api.php";
String params = "?action=opensearch&search="; // Add search query

// XML Result from a request:
<SearchSuggestion version="2.0">
<Query>Sweden</Query>
<Section>
<Item>
<Text>Sweden</Text>
<Description>
Sweden (pronounced ), officially the Kingdom of Sweden (Swedish: ), is a
   Nordic country on the Scandinavian Peninsula in Northern Europe.
</Description>
<Url>http://en.wikipedia.org/wiki/Sweden</Url>
</Item>
…
</SearchSuggestion>
```

Code examples (Last.fm Search)

```
// Setup Last.fm request…
String LASTFM_URL = "http://ws.audioscrobbler.com/2.0/?" +
   "method=geo.getevents&" + "api_key=<apikey>";

Location location = Location.lastKnownLocation();
QualifiedCoordinates coords = location.getQualifiedCoordinates():
String longitude = "&long=" + coords.getLongitude();
String latitude = "&lat=" + coords.getLatitude();

String requestUri = LASTFM_URL + longitude + latitude;

// Perform Last.fm API request
…
```

Code examples (Last.fm result)

```
<id>997523</id>
<title>The Allman Brothers Band</title>
−<artists>
    <artist>The Allman Brothers Band</artist>
    <headliner>The Allman Brothers Band</headliner>
</artists>
−<venue>
    <name>The Fox Theater</name>
  −<location>
      <city>Oakland, CA</city>
      <country>United States</country>
      <street>1807 Telegraph Ave.</street>
      <postalcode>94612</postalcode>
    −<geo:point>
        <geo:lat>37.808268</geo:lat>
        <geo:long>-122.269901</geo:long>
      </geo:point>
```

Demo 1: Location aware Twitter client

- A Twitter client for Java ME.
- Update the user's location in the Twitter profile.
- List Twitter statuses from nearby users.

Demo 1: Twitter API Code

```
public static final String URL = "http://search.twitter.com/
   search.json";

Arg[] params = new Arg[]{new Arg(”geocode”,locationString)};

// Perform an asynchronous HTTP request
Request.get(URL, params, httpHeaders, reqListener, context);

// Read statuses from your friends Twitter timeline
public void done(java.lang.Object context, Response response) {
   Result result = response.getResult();
   int statusCount = result.getSizeOfArray("");
   for(int i = 0; i < statusCount; i++) {
      parseStatus(result, ”[” + i ”].”);
   }
}
```

Demo 1: Twitter API JSON result

```
[{
  …
  "created_at": "Thu Jun 04 10:00:00 +0000 2009",
  "id": 1919521,
  "text": "I'm sitting at JavaOne and listening to Erik and
   Kenneth talking about cool Java ME apps!",
  "user":{
    "screen_name":"J1hax0r",
    "name":"Java Fan #1",
    "location":"Howard Street, San Francisco",
    …
  }
  …
},…
]
```

Demo 1: Twitter API Code

```
public static final String URL = "http://twitter.com/
   statuses/update.json";

// The new location we want on our Twitter profile
Arg[] params = new Arg[]{new Arg("location",
   locationText)};
String authString = "Basic " +
   BasicAuth.encode(username, password);
Arg[] httpHeaders = new Arg[] {new
   Arg(Arg.AUTHORIZATION, authString)}

// Post the new location to our Twitter profile
Request.post(URL, params, httpHeaders, reqListener,
   context);
```

Demo 2: Picture sharing with TwitPic

- Take a photo with your Java ME phone.
- Post it to TwitPic and add location tags.

Demo 2: Code – JSR-135

```
// Setup up the camera
player = Manager.createPlayer("capture://video");
player.realize();

// Create a new VideoControl to take snapshots
VideoControl  vc = (VideoControl)
  player.getControl("VideoControl");

player.start();

// take snapeshot with default encoding
byte[] imageData = vc.getSnapshot(null);
```

Demo 2: Code – TwitPic

```
// Upload photo to TwiPic
Part[] parts = new Part[4];
Arg[] imageHeaders = new Arg[]{new Arg("Content-
  Disposition", "form-data; name=\"media\"")};
parts[0] = new Part(imageData, imageHeaders);
// Add username, password and message to parts…

// Create the PostData object with a boundrary
PostData postData = new PostData(parts,
  "====================twitpic-upload========");

// Post the image
Request.post(UPLOAD_AND_POST, null, null, this,
  postData, UPLOAD_AND_POST);
```

Demo 3: Phone as a game input device

- Read accelerometer data for all axes.
- Send accelerometer data to PC.
- Use accelerometer data to control a game.

Demo 3: Code – JSR-256

```
// Setup SensorManager and listen for sensor events
SensorInfo[] sensorInfos =
   SensorManager.findSensors("acceleration",
   SensorInfo.CONTEXT_TYPE_USER);

if(sensorInfos.length > 0) {
     String url = sensorInfos[0].getUrl();

     SensorConnection sensor = (SensorConnection)
   Connector.open(url);

     sensor.setDataListener(this, 10);
}
```

Demo 3: Code – Sending to PC

```
// Send the sensor data to a broadcast address
// as a UDP package.
connection = (DatagramConnection)
   Connector.open("datagram://224.0.0.10:6060");

public void dataReceived(SensorConnection sensorConnection,
   Data[] datas, boolean b) {
    // Read the values from datas…
    byte[] msgData = parseData(datas);

    // Send the msg - should be run in a new thread!
    Datagram msg = connection.newDatagram(msgData,
   msgData.length);
    conn.send(msg);
}
```

Summary

- The mobile phone is the only device people carry with them **all the time**.
- Smart mobile applications don't require a smartphone!
- Think Web 2.0 – lots of useful, free, online services available!

- All demos are available at http://developer.sonyericsson.com

Questions?

http://www.flickr.com/photos/orinrobertjohn/3040007953/

Kenneth Andersson
kenneth.andersson@sonyericsson.com
Erik Hellman
erik.hellman@sonyericsson.com
http://twitter.com/ErikHellman

http://developer.sonyericsson.com

Sun microsystems