A Web Service Platform for Building Interoperable Augmented Reality Solutions

Petros Belimpasakis, Dr. Tech. Principal Researcher
12.10.2010
Background

There have been many Augmented Reality applications / demos / trials but, most of them

- Are build around very limited set of data / content
- Typically have limited linkage to a backend infrastructure
- Rarely share data with other similar systems

Some demos utilize external web services (such as Flickr), but those services are not built with MR/AR applications in mind

- They have limited support for media types
- Limited geographical searches

Essentially, every new MR application has to “re-invent the wheel”
Creating a Mixed Reality Web Service Platform

• Moving from stand-alone services to a platform
  • A common backend, secure and scalable, for easily creating mixed reality applications (desktop & mobile) and services
  • The platform handles user management social connections, user generated content with MR extensions, street-view panoramas, building models, road network, terrain, POIs...
  • Allow rich mash-ups, utilizing seamless interfaces between user-generated and commercial geo-content

Approach

• We are not just linking our platform to others. They can link their services to us
• APIs with secure authentication framework, so that privacy & security are top priority
• Use standard & defacto technologies as well as recognize early new trends in this area
Mid-term Target: Allow open & quick innovation for Mixed Reality

Long-Term Target: Influence the defacto APIs for MR services & applications
High-level view

Mixed Reality Web Service Platform

- Own Content repository
  - Photos, point clouds, 3D objects...

- External repositories & services
  - Flickr, Picasa

- Identity providers
  - Nokia account, OpenID

- Social Networks
  - NAVTEQ

- Unique Geo data

- 3rd party clients & services

Our MR Clients

RESTful interfaces
ReST APIs

- GET content/annotations/?lat=41.879933&lon=-87.62465&radius=0.5
- GET content/annotations/?lon1=23.857&lat1=61.4460&lon2=23.858&lat2=61.4470
- GET users/abc/content/annotations/
- GET content/annotations/206366
- POST content/annotations/
- PUT content/annotations/206366
- DELETE content/annotations/206366

Nokia Research Center
Now replace the “annotations” with... “photos”

- GET content/photos/?lat=41.879933&lon=-87.62465&radius=0.5
- GET content/photos/?lon1=23.857&lat1=61.4460&lon2=23.858&lat2=61.4470
- GET users/abc/content/photos/

- GET content/photos/1234

- POST content/photos/

- PUT content/photos/1234

- DELETE content/photos/1234

Nokia Research Center
Now replace the "annotations" with... "panos"

- GET /panos/?lat=41.879933&lon=-87.62465&radius=0.5
- GET /panos/?lon1=23.857&lat1=61.4460&lon2=23.858&lat2=61.4470
- GET /panos/232243
Now replace the "annotations" with... "bldgs"

- GET /bldgs/?lat=41.879933&lon=-87.62465&radius=0.5
- GET /bldgs/?lon1=23.857&lat1=61.4460&lon2=23.858&lat2=61.4470
- GET /bldgs/us_chi_23343
Now replace the "annotations" with... "terrains"

- GET /terrains/?lat=41.879933&lon=-87.62465&radius=0.5
- GET /terrains/?lon1=23.857&lat1=61.4460&lon2=23.858&lat2=61.4470
- GET /terrains/74342323
Now replace the “annotations” with... “roads”

- GET /roads/?lat=41.879933&lon=-87.62465&radius=0.5
- GET /roads/?lon1=23.857&lat1=61.4460&lon2=23.858&lat2=61.4470
- GET /roads/usa_chi_32433
Now replace the “annotations” with... “pois”

- GET /pois/?lat=41.879933&lon=-87.62465&radius=0.5
- GET /pois/?lon1=23.857&lat1=61.4460&lon2=23.858&lat2=61.4470
- GET /pois/5739434
Important: Linked data

• GET /panos/232243 = give the this specific panorama with id 232243
• GET /panos/232243
  • HTTP header: x-mrs-inline: blds = give the this specific panorama with id 232243, but also let me know the VISIBLE buildings in it
Important: Linked data

• GET /panos/232243

• HTTP header: x-mrs-inline: bldgs, pois = give the this specific panorama with id 232243, but also let me know the VISIBLE buildings in it AND the POIs in each bldg

Nokia Research Center
City Scene: A mirror-world application created on top of our platform

- Street-view mirror world
- 3D model (touch & highlight buildings)
- Explore POIs and leave content on buildings

Nokia Research Center
Other mobile specific considerations

• Pagination (give me page x, of pages with size of y elements)

• Verbosity (id, summary, full, raw)

• Client can choose data representation (XML or JSON)

• Compression of all responses possible on client’s request
Developing on our API:
1. Access and API Development Guide
2. Apply for a developer key
3. Get a sample client code
4. Give us your feedback

News & Updates
- Read the latest service platform release notes & feature additions

API Terms & Conditions
- Read the API Terms & Conditions

Developer Helping Tools & Utilities
- Developer Help Center

Useful External Development Resources
- OAuth protocol & OAuth client libraries
- Beginners Guide to OAuth
- Building Web Services the REST Way

Reporting a bug
- We are using code.nokia.com as a bug system. To assign tickets to us, you need to do the following:
- To be done only once, for the first time.

Mixed Reality Service Platform API Documentation
Here is our API for the Mixed Reality Service Platform (MRSP).
It is designed based on open Internet technologies to support 3rd party integration and mash-ups.
The selected service architecture style is REST.

1. RESTful Service Architecture in detail
   1. Resources
   2. Representations
   3. URI structure
   4. Links
   5. Scenarios
   6. Supported return types

2. Data Formats
   1. Object type
   2. Properties
   3. Verticity
   4. Content linking
   5. External service inclusion
   6. Compression

3. Authentication of Requests
   1. Two-step HTTP Authentication
   2. OAuth Authentication

4. Security
   5. Access Control
   6. User API
   7. Group API
   8. Content API

   1. Common subresources
   2. Location of a content item
   3. Uploading a new content piece
   4. Attributes
   5. Searches and projections
   6. Search results
   7. Entity API
Developer's Corner

Mixed Reality Services

Developer's Corner

Building API

Building footprint outlines are based on Navdea data and are "static" content, which can only be accessed via GET, without belonging to any user. Currently only a limited amount of cities is available.

Buildings are represented by sub-building (one building might have many sub-buildings). In turn, a sub-building has a geometry that has one or many multipolygons. A multipolygon is a set of coordinate points (corners), which represent the corners of a building's footprint projection.

<table>
<thead>
<tr>
<th>Resource URI</th>
<th>Description</th>
<th>Operations</th>
<th>Status codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddIgn</td>
<td>A building instance</td>
<td>GET, POST</td>
<td>200, 400, 404</td>
</tr>
<tr>
<td>AddIgn</td>
<td>A specific building item</td>
<td>GET, POST</td>
<td>200, 400, 404</td>
</tr>
<tr>
<td>AddIgn</td>
<td>Buildings within 0m from the point (lon, lat)</td>
<td>GET, POST</td>
<td>200, 400, 404</td>
</tr>
<tr>
<td>AddIgn</td>
<td>Buildings within the bounding box of (lon1, lat1) to (lon2, lat2)</td>
<td>GET, POST</td>
<td>200, 400, 404</td>
</tr>
</tbody>
</table>

Supported features:

- Predictive services
- Location-based services
- Access/Control
- Compression
- Representations

Yes | No | No | Yes

Try real data and overlay it on Google Maps

Request:


Example: Get all buildings in bounded box, screen in overview: get one building

Page number: 1  Page size: 20  Try me
Image Space: Another service built with MRS-WS

Lets users share their experiences, photos and other content about specific places, while helping them understand their spatial relationships.

http://betalabs.nokia.com/apps/nokia-image-space
Image Space: The mobile client
Image Space: The mobile client
Image Space: Social Network

Nokia Research Center
Conclusions

• Using standard Web technologies, MRS-WS provides the core components for building augmented and mixed reality solutions

• Taking care of media items storage, social connections, identity and aggregation of content from external repositories it off-loads the end application/client development

• We unify the way user-generated and commercial geo-content are accessed

• While we are only evaluating the possibility of opening up the platform to 3rd parties, we are looking forward to align our efforts with the on-going standardization activities for AR APIs and data formats as they are advancing

• We believe that the Web based architecture would significantly speed up and ease the development of AR applications.
Thank You