

A Web Service Platform for Building Interoperable Augmented Reality Solutions

Petros Belimpasakis, Petri Selonen, Yu You
Mixed Reality Solutions, Nokia Research Center, Finland
{firstname.lastname}@nokia.com

Abstract-This paper presents a Web service platform for building augmented and mixed reality solutions. The platform, MRS-WS, allows clients to create, retrieve and modify augmented reality enabled content using a unified interface based on standard Web technologies like HTTP, REST and Linked Data. The platform serves both user generated and commercial geo-content. It aspires to relief developers from the burden of creating a backend infrastructure for their augmented reality applications.

I. INTRODUCTION

Smart phones are becoming excellent platforms for running Augmented Reality (AR) applications with their always-on Internet connectivity, cameras, advanced sensors and fast processors [2]. While platforms and frameworks exist for building stand-alone mobile AR applications (e.g. [5][6]), similar efforts on the server side have been limited. Until recently, most mobile AR systems have served locally stored content with minimal or no linkage at all to centralized systems [1].

Recent mobile AR applications such as Layar [3] and Wikitude [4] link their clients to a backend infrastructure and provide Web service interfaces for developers to publish their AR enabled content. However, this content is only exposed and can only be consumed using the AR browsers of the respective parties. This limits the freedom of developers to create new AR applications, services and new interaction paradigms.

This paper describes the Mixed Reality Web service platform (MRS-WS) that supports building AR solutions for mobile, browser or desktop clients. When building the platform, we recognized the need for having all AR enabled content accessible on the Web by using standard Internet technologies. Following the Linked Data and Representational State Transfer architecture (REST) [8] principles, we use URIs to identify AR content, HTTP as an application protocol to access (retrieve, create, modify and remove) the content, standard representation formats (XML/JSON) and links between the resources. The platform effectively decouples clients and content providers from the backend infrastructure by using standard Web technologies. It further allows different AR solutions to share AR content.

While work towards commonly agreed and standard AR APIs and data formats are still at their infancy, it seems obvious that the Web based architecture (“AR in the Cloud”) is a key enabler for interoperability of AR services as suggested by e.g. the recent W3C AR workshop. The MRS-WS platform with several end to end AR solutions built on top of it is one effort towards this direction.

II. FUNCTIONALITY

When gathering functional requirements for MRS-WS, we identified [7] a set of features that should be offered to clients:

- Store content and AR metadata for it. The metadata comprises spatial relations like geo-location, orientation and accelerometer data. This enables content to be retrieved and visualized in an Augmented Reality view on a 3D space. Store both AR enabled multimedia content like photos, videos, audio and 3D objects, as well as non-multimedia like point clouds and micro-blogging entries.
- Allow adding application specific metadata to content items. That would allow clients with specific needs to store pieces of information they might need, without modifications at the platform side. The platform can transparently host this metadata, linked to a content item, and provide it back to the client when it is retrieved, without the need to further understand it at the server side.
- Aggregate and link content from 3rd party repositories (such as Flickr) to provide a uniform interface to all content available to the clients. Content that is hosted by the MRS-WS platform and externally hosted content should appear via the same API, making the clients agnostic about the interfaces and existence of external repositories. Such server side content aggregation is used for leveraging content from non-AR services; for example, while Flickr is not AR service, the photos it hosts can be machine tagged with AR metadata.
- Support the typical social media extensions for every content item, such as comments, tags, ratings and voting. Also, when those performed on an item that is hosted by an external content repository, they should be relayed there. Thus, all the “standard” social media extensions could be easily added to any application.
- Host commercial real world geo-data that is useful for clients dealing with AR and Mixed Reality (MR) applications, such as street-view panoramas, Points of Interest (POIs), terrain, building information.
- Support user management, i.e. handling user identities, social connections and access control lists, as well as aggregating social connections from 3rd party social networks and providing them in a common interface to the clients.

The main contribution is to provide an easy one-stop API for developers, both internal and external, to build AR applications and web mash-ups, without the need to develop yet again another backend system.

III. TECHNICAL ARCHITECTURE AND APIS

MRS-WS is based on a client-server model, where the clients (mobile, desktop or other servers) communicate with our backend, as shown in Figure 1. The platform can further aggregate and synchronize content with 3rd party external content repositories using their Web APIs.

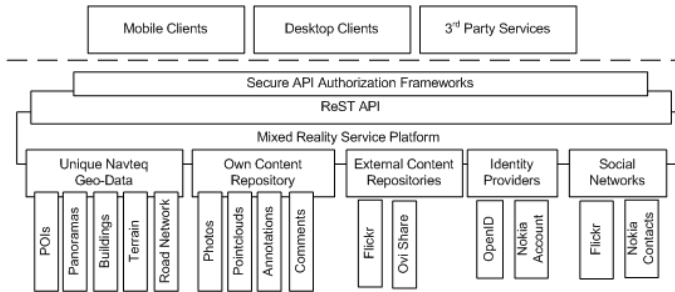


Figure 1. Web Service Platform High-Level Architecture

The platform also hosts commercial geo-content acquired from Navteq, a world leader in premium-quality digital map data and content. The data includes street-view panoramas, building outlines and their 3D models, terrain morphology, road network, Points of Interest (POIs). Moreover, having done some advanced preprocessing, useful associations can be provided to the clients, such mappings to the exact buildings visible in a street-view panorama with appropriate aligning, linking POIs/businesses in a building and associating the neighboring panoramas.

All content, both internal and external, are exposed to the client applications via web interfaces following the REST architectural style. Two types of generic data objects are supported: items and containers. An item represents an individual data object (like a photo or a user). Containers represent a collection of homogeneous items (like all my photos or all the registered users). The supported user generated content include:

- Photos, videos and audio: Typical multimedia content, which via our platform can be geo-referenced and spatially registered in the 3D space.
- Annotations: MR annotations can be attached to a particular location or a building. The annotation can comprise a title and a textual description, as well as links to other content elements. Essentially an Annotation is a way of the users to annotate physical entities with digital information and share those.
- Pointclouds: A pointcloud is a 3D mesh, typically constructed by user-generated photos, by combining common features, after applying computer vision techniques on multiple photos.

Every resource is represented by a Uniform Resource Identifier (URI), on which the standard Hypertext Transfer Protocol (HTTP) operations (GET, PUT, POST, DELETE) can

be applied. Some concrete examples are given for the Annotation API:

| Resource URI | Description | Operations | Status codes |
|--|---|------------------|--------------------|
| /content/annotations | All annotations in the system | GET | 200, 400, 401 |
| /users/{username}/content/annotations | All annotations owned by user <i>username</i> | GET, POST | 200, 201, 400, 401 |
| /users/{username}/content/annotations/{id} | Unique annotation | GET, PUT, DELETE | 200, 400, 401, 403 |

| Subresource URI | Description | Operations | Status codes |
|------------------|--------------------------------------|------------|--------------------|
| /location | Location subresource | GET, PUT | 200, 201, 400, 401 |
| /contenturis | References to other content elements | GET, POST | 200, 201, 400, 401 |
| /comments | Comments | GET, POST | 200, 201, 400, 401 |
| /attrs | Attributes | GET, POST | 200, 201, 400, 401 |
| /tags | Tags | GET | N/A |
| /tags/{tag.name} | Individual Tag | GET, PUT | 200, 201, 400, 401 |

An sample representation of an unique annotation resource is given here:

```
<annotation href="...">
  <id>4195042682</id>
  <owner>/users/demouser</owner>
  <updated>2009-12-18 04:01:13.0</updated>
  <published>2009-12-18 04:01:02.0</published>
  <title>I have an apartment for rent here!</title>
  <alert>true</alert>
  ...
  <locations>
    <location href="...">
      <lat>61.44921</lat>
      <lon>23.86039</lon>
      ...
      <roll></roll>
      <pitch></pitch>
    </location>
  </locations>
  <building href="...">...</building>
  <attrs>
    <attr>
      <key>Agency</key>
      <value>ABC Real Estate</value>
    </attr>
    ...
  </attrs>
</annotation>
```

A similar example for a photo resource:

```
<photo>
  <id>2312332</id>
  <owner>/users/myname</owner>
  <updated>2009-12-18 04:01:13.0</updated>
  <taken>2009-12-18 04:01:02.0</taken>
  <title>Great View</title>
  <mediauri>http://server/users/myname/content/photos/2312332/</mediauri>
```

```

<tnsmalluri>
http://server/users/myname/content/photos
/2312332/tnsmall</tnsmalluri>
<tnmediumuri>
http://server/users/myname/content/photos
/2312332/tnmedium</tnmediumuri>
<tnlargeuri>
http://server/users/myname/content/photos
/2312332/tnlarge</tnlargeuri>
<locations>
<location>
<lat>61.497289</lat>
<lon>23.761303</lon>
<yaw>180</yaw>
<roll>0</roll>
<pitch>0</pitch>
</location>
</locations>
</photo>

```

Most of the fields in the photo item description are self-explanatory like title, time and owner. The binary photo items of different resolutions are URLs capable of linking to content both hosted on our platform or external content providers. The location metadata provides yaw, roll and pitch angles for orientation, referring to rotations around the respective axes starting from a reference defined equilibrium state. Yaw, is defined as the angle between the device's heading and the earth's magnetic north (bearing), while roll and pitch represent the angles from the horizon plane. With the combination of GPS, magnetometer and accelerometer sensors, a mobile device can display this data in the 3D space in an augmented reality view.

When getting/searching for a resource, some further parameters can be specified by the client, for filtering the results. For example:

- Geo-searching, either in a bounding box (e.g. /annotations/?lat1=41.95&lon1=-87.7&lat2=41.96&lon2=-87.6), or in proximity (e.g. / annotations/?lat=41.8&lon=-87.6&radius=1). Allows the client to request results in a specific region, which is typically the area where the mobile device is located in.

- Temporal searching: It is possible to limit the scope of the search within a specific time window, for retrieving only latest content, for example. In the search query the time window (since- until) is specified in the form of UNIX timestamps. The matching is done against an item's "Updated time" or "Taken time", or combination of both (e.g. /photos/lat=40.1&lon=-13.2&radius=0.5&updated_since=126223100).

The API also offers access to other "static" real world geo-content originating from Navteq, such as

- Building (/buildings) footprints and 3D models (Figure 2).
- Terrain (/terrain) tiles of earth's morphology (Figure 3)
- Street-view 360 panorama (/panoramas) photos (Figure 4), at multiple resolutions
- Point-of-interest (/pois), with details about businesses and attractions. Those are "yellow pages" POIs, containing the

address/location of businesses, their description, phone numbers, opening hours, etc.

- Road network, with the details and geometry of different road segments

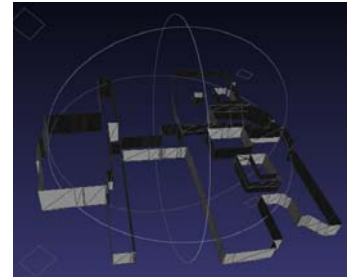


Figure 2. Building elements (2D footprint & 3D model)

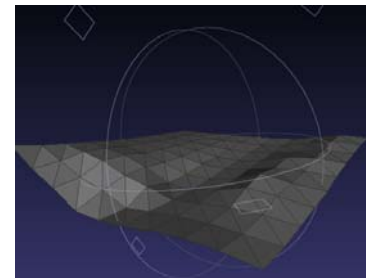
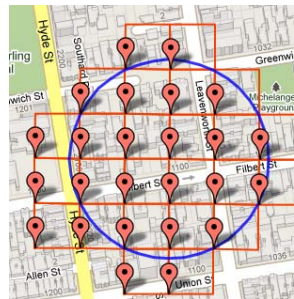


Figure 3. Terrain tiles (2D footprints & 3D model)



Figure 4. Street-view panorama

For all the real world geo-content, the 2D representation is based on a set of points, multilines and multipolygons aligned to the world coordinate system (e.g. a building is described by all its corner coordinates in XML/JSON format). The 3D representations are provided in the PLY file format [9].

To all those resources and containers, user-generated content and static geo-data, a uniform set of operations can be applied. Examples include:

- Geo-searching, as described earlier
- Pagination [x-mrs-api: page(), pagesize()] for controlling the number of objects fetched from a container per request by setting the page size and number
- Verbosity [x-mrs-api: verbosity()] for controlling the representation subset, ie. selected attributes for the retrieved items.
- Inlining [x-mrs-deco: inline()] for selecting which subresources and referenced items are included per resources

in the response to decide when to reduce the number of requests to the server and when to reduce the amount of data transferred.

Inlining is heavily used in MRS-WS for exposing advanced geo-data associations. For example, we have pre-calculated from the raw Navteq data the specific buildings that are visible in a given street view panorama, along with their associated POIs. A client that requests a specific panorama image can also request the building data (e.g. links to 3D building models) to be inlined. This combined information can allow very advanced applications, such as touching and highlighting buildings, in an augmented reality view, for interacting with the real world (e.g. touch a building to find the businesses it hosts).

IV. EXAMPLE APPLICATIONS

MRS-WS was first developed with Nokia Image Space [10] at Nokia Research Center. The service provides a new view to the photos owned by individual users, as well as to those of his friends and other people. All captured photos, via a smart phone, do not only contain the location of the image, but also the orientation and exact camera pose (tilt angles), which were uploaded to our platform as orientation metadata. The user can then consume the content using either of the two Image Space clients, a desktop PC client and a mobile client (Figure 5). The mobile client, running on the Symbian smart phone platform, displays the content in an augmented reality view, in-situ, as the photos are overlaid on top of the live camera feed. Image Space is currently offered as a public beta service.



Figure 5. Image Space mobile (Augmented Reality) client

Another application built on top of MRS-WS is a city exploration mixed reality application (Figure 6) for mobile smart phones. It allows users to explore surroundings in a street-view style, discover the businesses hosted in the buildings and associate user generated media to the buildings (e.g. leave annotations/notes on buildings). At the time of writing the application is not yet publically available but a research concept.



Figure 6. Mixed reality city exploration application

Our whole system was designed with developers in mind, targeting to open this system to the public at some later point in time. The platform is accompanied by a rich “Developer’s Corner” documentation that allows developers to study the documentation and try any request call/parameters and get their results visualized on a browser, as shown in Figure 7. This allows better understanding of the API, and verifying the results that the client would be receiving.

Building API

Building footprint outlines are based on Navteq data and are “static” content, which can only be accessed via GET, without belonging to any user. Current Buildings are represented by sub-buildings (one building might have many sub-buildings). In turn, a sub-building has a geometry that has one or many multiple pairs (lon,lat), which represent the corners of a building’s footprint projection.

| Resource URI | Description | Operations | Status codes |
|--|---|------------|--------------|
| /Buildings | All buildings available | GET | TBD |
| /Buildings/{id} | A specific building item | GET | TBD |
| /Buildings/{obj} | A specific building Obj file (details on Obj file format) | GET | TBD |
| /Buildings?lon=nn.nnnn&lat=nn.nnnn&radius=nn.nn | Buildings within nn.nn km distance of point (lon,lat) | GET | TBD |
| /Buildings?lon1=nn.nnnn&lat1=nn.nnnn&lon2=nn.nnnn&lat2=nn.nnnn | Buildings within the bounded box of (lon1,lat1)-(lon2,lat2) | GET | TBD |

Try real data and overlay it on Google Maps

Request:
<http://www.example.com/buildings/mrs/rest/buildings?lon1=7.45&lat1=46.95&lon2=7.452&lat2=46.956>

Example URL: [search in bounded box](#), [search in x/y/m/r](#), [get one building](#)

Page number: Page size:

Response:

Figure 7. “Developer’s Corner” screenshot

We are currently evaluating our developer offering via a closed set of university partners. Our plan is to gradually expand the developer API access to more partners, before deciding if it is ready for fully public developer access.

V. 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.

Moreover, we unify the way user-generated and commercial geo-content, such as real life street-view panoramas, 3D building models, terrain, road network and POIs, are accessed. Once developers are familiar with one content type, it is simple and intuitive to access all the rest.

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.

REFERENCES

- [1] X. Luo. From Augmented Reality to Augmented Computing: A Look at Cloud-Mobile Convergence. In *Proceedings of the International Symposium on Ubiquitous Virtual Reality (ISUVR09)*, Gwangju, South Korea, July 2009.
- [2] M. Kähäri, and D. J. Murphy. MARA - Sensor Based Augmented Reality System for Mobile Imaging. In *Proceedings of the Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR06)*, Santa Barbara, USA, October 2006.
- [3] Layar Augmented Reality Browser, Available: <http://layar.com/>
- [4] Wikitude World Browser, Available: <http://www.wikitude.org/>
- [5] N. Elmqvist, M. Fjeld, D. Axblom, J. Claesson, J. Hagberg, D. Segerdahl, Y. Tai So, A. Svensson, M. Thorén, M. Wiklander. 3DVN: A Mixed Reality Platform for Mobile Navigation Assistance, *Proceedings of ACM CHI2007 Workshop on Mobile Spatial Interaction*, San Jose, CA, USA, April 2007.
- [6] S. Uchiyama, K. Takemoto, K. Satoh, H. Yamamoto, and H. Tamura. MR Platform: A Basic Body on Which Mixed Reality Applications Are Built, In *Proceedings of the 1st international Symposium on Mixed and Augmented Reality*, Washington, DC, USA, September 2002.
- [7] P. Selonen, P. Belimpasakis, Y. You, "Experiences In Building a ReSTful Mixed Reality Web Service Platform". *Proceedings of the 10th International Conference on Web Engineering (ICWE 2010)*, Vienna, Austria, July 2010.
- [8] R. T. Fielding. Architectural styles and the design of network-based software architectures, Ph.D. dissertation, University of California, Irvine, 2000.
- [9] The PLY Format, Available: http://www.cc.gatech.edu/projects/large_models/ply.html
- [10] S. Uusitalo, P. Eskolin, P. Belimpasakis. A Solution for Navigating User-Generated Content, *Proceedings of the Eighth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, Orlando, FL, USA, October 2009.