This report describes the context of the Maphub demonstration experiment and the lessons learned. The goal of the experiment was to build a Web-application prototype for the annotation of digitized historic maps and to publish user-contributed annotations and associated metadata as dereferencable Web resources, following the Open Annotation Model guidelines. All project resources, including demo and API documentation, were released under an open-source license and made available at http://maphub.github.com. The project did not face any major technical difficulties in implementing the Open Annotation Model on the server side. However, the conceptual simplicity of the supported annotation types and the relative technical complexity and verbosity of the resulting annotation serializations raise model usability concerns. Thus, we propose a set of adjustments to streamline the Open Annotation model.

Maphub is a Web-Application that allows end-users to annotate digitized high-resolution historic maps. It has been bootstrapped with approximately 6000 historic maps from the Library of Congress Map Division (http://www.loc.gov/rr/geogmap/). It implements the following use cases:

- **UC1: Annotating regions on high-resolution map images**: the high-resolution zoomable maps presented to Maphub users are compound Web resources comprising a set of image tiles and a metadata descriptor file. Users can annotate complete maps or map regions.
- **UC2: Georeferencing maps**: users can mark places on maps and link those places to geographical Web resources (e.g., Geonames). Using this information, it is possible to establish a correspondence between a map’s image coordinates and real world geographic coordinates.
- **UC3: Semantic Tagging**: while the user is creating textual annotations on a map or map region, Maphub automatically proposes resources from the Linked Data Web (e.g., DBpedia), which may be semantically related to the annotation and therefore also to the annotated map. Users can accept or reject link proposals.
- **UC4: Map Overlays**: After adding at least three control points to a map, real world locations can be computed for any point on the map. This allows the creation of map views through Google Maps and Google Earth.
- **UC5: Map Annotations on Mobile Clients**: annotations can be viewed and created on Web-enabled mobile clients (e.g., iPad).
Description of Annotation Classes

The implementation of Use Cases 1-3 is reflected in two annotation classes currently supported by Maphub:

**Georeference Annotation:** associates a place URI (a **Semantic Tag**) provided by Geonames or other online gazeteers with a place on the map (the annotation **Target**), as shown in Figure 1.

![Figure 1. Example Georeference Annotation](image1)

**Commentarial Annotation:** Commenting on maps or map segments, as shown in Figure 2 and Figure 3, is the other major use case in Maphub. Users can select and then comment upon map regions. While users are writing their comments, the system proposes possibly relevant tags. These tags are links to DBPedia, and denoted as **Semantic Tags**. Accepted tags become part of the annotation.

![Figure 2. Commentarial Annotation](image2)
All annotations in Maphub are dereferenceable Web resources. When a client issues an HTTP GET request against an annotation’s HTTP URI, Maphub determines the response format based on the value of the HTTP Accept header.

**Example/Illustrative Annotations**

**Georeference Annotation**

Figure 4 shows an example Georeference Annotation represented in the Open Annotation model. Executing the following curl request can retrieve it:

```
curl -H "Accept: text/turtle" http://maphub.herokuapp.com/control_points/4
```

Appendix A shows the Turtle serialization of that annotation.
Commentarial Annotation

Figure 5 shows an example commentarial map annotation created in Maphub. Executing the following curl request can retrieve the annotation:


Appendix A shows the Turtle serialization for this example annotation.

Summary of Obstacles

We did not face any major technical obstacles when implementing the Open Annotation Model in Maphub on the server side. The Web application is based on Ruby on Rails, which is a widely used Web application framework. Libraries for producing any kind of RDF serializations are available and the RESTful design of Ruby on Rails facilitates the publication of annotation serializations on the Web. Implementing the Open Annotation Model was a matter of producing an RDF serialization based on the underlying annotation data. Creating UUID nodes for RDF serializations required another third-party library.
Technical Lessons Learned

We did not face any major technical difficulties in implementing the Open Annotation Model on the server side. The model is expressive enough for the use cases supported in Maphub. However, the conceptual simplicity of the Maphub annotation use cases and the relative technical complexity and verbosity of the resulting annotation serializations raised concerns regarding client usability. We believe that some adjustments in the OA specifications could streamline the model, reduce its verbosity, and thereby facilitate its adoption.

Direct Relationship between Annotation and the Source

"Give me all annotations for resource X", is probably one of the most important queries that needs to be answered by a client. X could be an image URI, the URI of a video, whatever. Since the Target of an annotation may be a resource with its own dereferencable URI or a Specific Target with a UUID node, both options must be considered when formulating a query. The result will be a SPARQL UNION query or some conditional node traversal code when using an RDF API.

Technically, it is of course possible to do that, but given the importance of that query, we argue that the solution is not very intuitive and maybe also not very efficient. We believe that this can easily be fixed by introducing a direct relationship property (e.g., oa:annotates, oa:hasTargetSource) between the Annotation and the Source resource.

Fragment URIs as Targets

The Maphub API (the GeoReference part) follows the OA recommendation and uses a Specific Resource and a Fragment Selector to express that a URI annotates an XY point on a raster image. We could express the same information by using W3C Media Fragments and thereby reduce the verbosity and complexity of the resulting serialization. API consumers then don't even need to know about OA-specific "Specific Resources", "Fragment Selectors", etc.

The Open Annotation model currently does NOT RECOMMEND the use of fragment URIs for identifying segments of Targets or Bodies because of three issues (see Specification, Section 5.2.1):

1. "...cannot query the source directly"
2. "...they are not compatible with State and Style Specifiers; many annotations may have the same segment of interest, but have different States and Styles"
3. "Fragment URIs conflate the identity and the description of the segment of interest by including the description inline within the identity"

We believe that having an explicit relationship between an annotation and the source, as discussed in the previous section, can solve Issue 1. From previous emails and discussions we understood that Styles are now directly attached to the Annotation, which also means that that they are contextualized. We believe
that the "State" issue can be solved in a similar way and would also result in a more consistent annotation model that supports fragment URIs.

We cannot follow the rationale of the third issue; however, we believe that for very practical reasons the OA model should reuse what other specifications (Web Architecture, Media Fragment RFCs) already define; these specifications evolve orthogonally to the Open Annotation model specification, bring modularity and flexibility, and avoid the risk of re-designing existing work.

We strongly believe that the benefits of reusing (Media) Fragment URIs in OA prevail the arguments of not using them and therefore propose to RECOMMEND the use of Fragment URIs and only fall back on OA-specific Selectors if Fragment URIs not available or not expressive enough.

**Simple Literal Body Shortcut**

The prevailing view is that annotations are relationships between resources (the body and the target) and that inline bodies are represented as resources using the Content in RDF specification (see 6.1). However, the Maphub demonstrator, the majority of other Open Annotation use cases demonstrated in the OAC Phase 2 workshop, as well as most cookbook examples in the Open Annotation specification show that many annotation bodies are simple strings, which could be represented as literals.

Given these facts, we propose to reconsider the view of annotations being relationships and propose to introduce a "shortcut" property between the Annotation and the "content" Literal (e.g., hasLiteralBody). This allows the expression of simple annotations in a more straightforward way and doesn’t contradict the current oa:hasBody approach.

**Style Attached directly to the Annotation**

We don’t express style information in Maphub serializations because we believe that styling information and data representation should be separated. However, we understand that this feature might be required in other use cases and prefer the approach of optionally attaching style directly to the annotation over attaching it to the Specific Target.

**JSON (-LD) Serialization Recommendation**

At the moment the spec recommends RDF/XML to be used as default serialization language. We propose to also consider JSON(-LD) at least as alternative serialization format for JavaScript clients.
Generalizable Results and Conclusions

The results of the Maphub demonstrator experiment with regard to the Open Annotation model are that (i) the model is expressive enough for the use cases implemented in Maphub, (ii) implementing the required RDF serializations did not lead to any major technical obstacles, and (iii) the current model leads to complex and verbose serializations also if the underlying annotation use case is conceptually rather simple.

We believe that our results also apply for other use cases involving semantic tagging or textual comments on multimedia resources. The adjustments we propose in the previous section, could simplify the model and facilitate its adoption while retaining compatibility for other, possibly conceptually more complex annotation use cases.

Appendix A – Annotation Serializations

Georeference Annotation – Turtle Serialization

```turtle
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix oa: <http://www.w3.org/ns/openannotation/core/> .
@prefix foaf: <http://xmlns.com/foaf/spec/> .
@prefix oai: <http://www.w3.org/ns/openannotation/extensions/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix maphub: <http://maphub.info/ns/vocab#Georeference/> .

<http://maphub.herokuapp.com/control_points/4>
  a maphub:Georeference, oa:Annotation, oa:Tagging ;
  oa:annotated "2012-08-17T17:13:41Z" <http://www.w3.org/2001/XMLSchema#dateTime> ;
  oa:annotator <http://maphub.herokuapp.com/control_points/881c90ca-e892-11e1-a637-1231381d2be6> ;
  oa:generator <http://maphub.herokuapp.com> ;
  oa:hasTarget <http://maphub.herokuapp.com/control_points/8824e514-e892-11e1-a637-1231381d2be6> ;

<http://maphub.herokuapp.com/control_points/881c90ca-e892-11e1-a637-1231381d2be6>
  fnaf:box 'bernhard.haslhofer@cornell.edu' ;
  fnaf:name 'behas' .

<http://maphub.herokuapp.com/control_points/8824e514-e892-11e1-a637-1231381d2be6>
  a oai:SpecificResource ;
  oai:hasSelector <http://maphub.herokuapp.com/control_points/8825gb62-e892-11e1-a637-1231381d2be6> ;
  oai:hasSource <http://samos.mninf.univie.ac.at/maps/raw/g3200.ct000725C.jp2> .

<http://maphub.herokuapp.com/control_points/8825gb62-e892-11e1-a637-1231381d2be6>
  a oai:FragmentSelector ;
  rdf:value 'g3200.16551,1231,1,1' .

<http://samos.mninf.univie.ac.at/maps/raw/g3200.ct000725C.jp2>
  dc:format 'image/jp2' ;
  a dct:StillImage .
```
Commentarial Annotation – Turtle Serialization

@prefix ctt: <http://www.w3.org/2011/content#>  .
@prefix dotwca: <http://purl.org/ao/twca/>  .
@prefix foaf: <http://xmlns.com/foaf/0.1/>  .
@prefix oai: <http://www.w3.org/ns/openannotation/core/>  .
@prefix oai: <http://www.w3.org/ns/openannotation/extensions/>  .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  .

<http://maphub.herokuapp.com/annotations/4> a oaiAnnotation;
  o:annotated "2012-07-18T02:03:58Z" <http://www.w3.org/2001/XMLSchema#dateTime>;
  o:annotator <http://maphub.herokuapp.com/hasAnnotator>;
  o:annotationTarget <http://maphub.herokuapp.com/annotationTarget>;
  o:hasBody <http://maphub.herokuapp.com/hasBody>;
  o:contains <http://maphub.herokuapp.com/contains>;
  o:hasSemanticTag <http://dbpedia.org/resource/Mediterranean_Sea>,
  <http://dbpedia.org/resource/Strait_of_Gibraltar>,
  <http://dbpedia.org/resource/Pillars_of_Hercules>,
  <http://dbpedia.org/resource/Gibraltar>,
  <http://dbpedia.org/resource/Atlantic_Ocean>,
  <http://dbpedia.org/resource/Classical_antiquity>,
  <http://dbpedia.org/resource/Ancient_history>,
  <http://dbpedia.org/resource/Ancient_Egypt>,
  <http://dbpedia.org/resource/Mediterranean_Basin>.

<http://maphub.herokuapp.com/hasAnnotator> "bernhard.hesloper@cornell.edu";
foaf:firstName "beben".

<http://maphub.herokuapp.com/hasBody> "In antiquity, the Strait of Gibraltar (which connects the Atlantic Ocean with the Mediterranean Sea) was also known by the name "The Pillars of Hercules". This is the reason for this description."


<http://maphub.herokuapp.com/hasSemanticTag> "http://dbpedia.org/resource/Ancient_Egypt".


<http://maphub.herokuapp.com/hasSemanticTag> "http://dbpedia.org/resource/Mediterranean_Sea".

<http://maphub.herokuapp.com/hasSemanticTag> "http://dbpedia.org/resource/Ancient_history".


<http://maphub.herokuapp.com/hasSemanticTag> "bernhard.hesloper@cornell.edu";
foaf:firstName "beben".

<http://maphub.herokuapp.com/contains> "<ctt:ContentAsText ctt:format "text/plain" ctt:chars "In antiquity, the Strait of Gibraltar (which connects the Atlantic Ocean with the Mediterranean Sea) was also known by the name "The Pillars of Hercules". This is the reason for this description."


<http://dbpedia.org/resource/Ancient_Egypt> .


<http://dbpedia.org/resource/Ancient_history> .

<http://dbpedia.org/resource/Mediterranean_Sea> .


<http://dbpedia.org/resource/Mediterranean_Basin> .

<http://dbpedia.org/resource/Atlantic_Ocean> .


<http://dbpedia.org/resource/Ancient_history> .

<http://dbpedia.org/resource/Ancient_Egypt> .

<http://dbpedia.org/resource/Mediterranean_Sea> .