

Open Annotation Data Model

A Practical Overview for IIF & Mirador

Michael Appleby

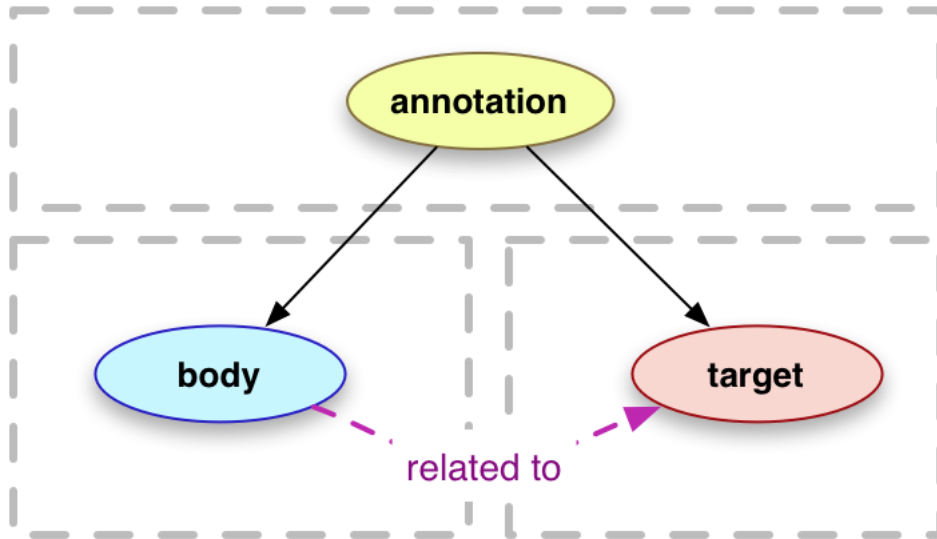
Yale Center for British Art

SWIB17

Goals

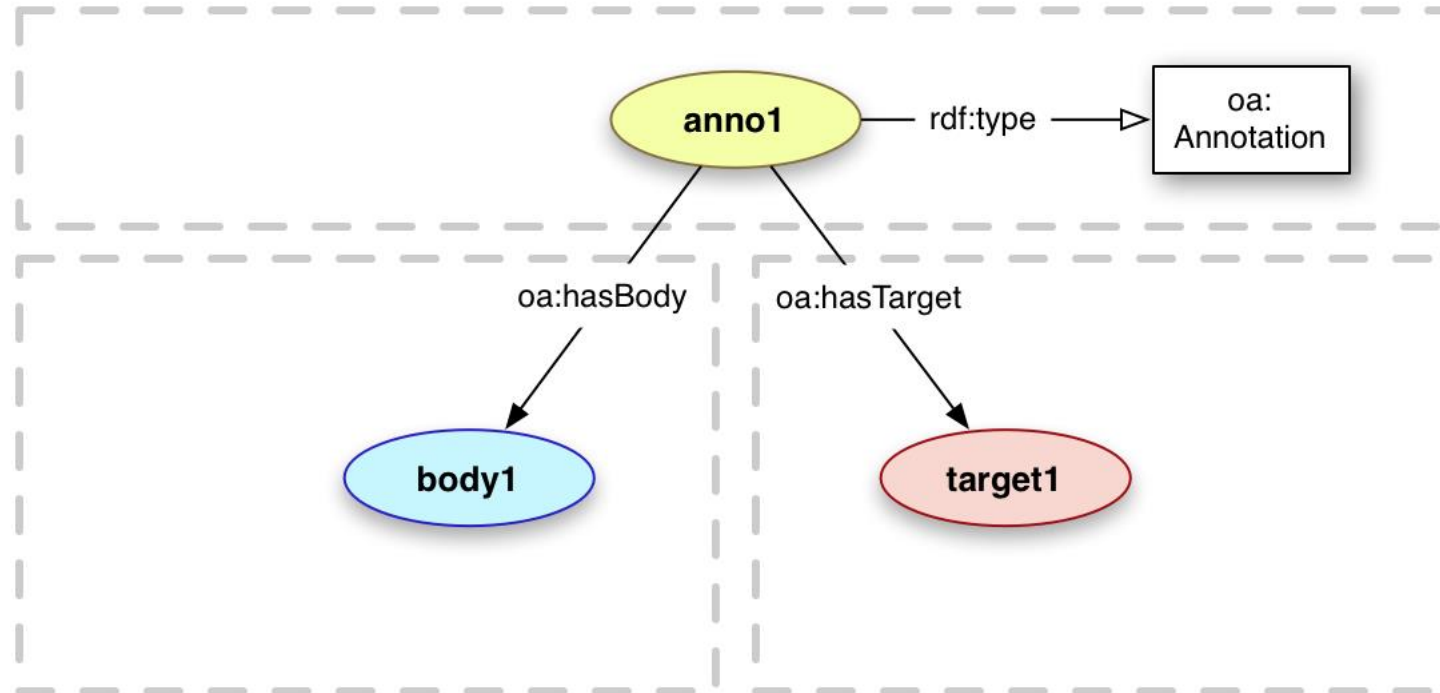
- Review the Open Annotation Data Model
<http://www.openannotation.org/spec/core/>
- Focus on the core elements that are used in IIF and Mirador
- Reference elements not supported in current IIF clients
- Show the IIF JSON-LD serialization that we'll be using in the hands-on examples

What's an Annotation?



An annotation is considered to be a set of connected resources, typically including a body and target, and conveys that the body is related to the target. The exact nature of this relationship changes according to the intention of the annotation, but most frequently conveys that the body is somehow "about" the target.

oa:Annotation



```
<anno1> a oa:Annotation ;  
oa:hasBody <body1> ;  
oa:hasTarget <target1> .
```

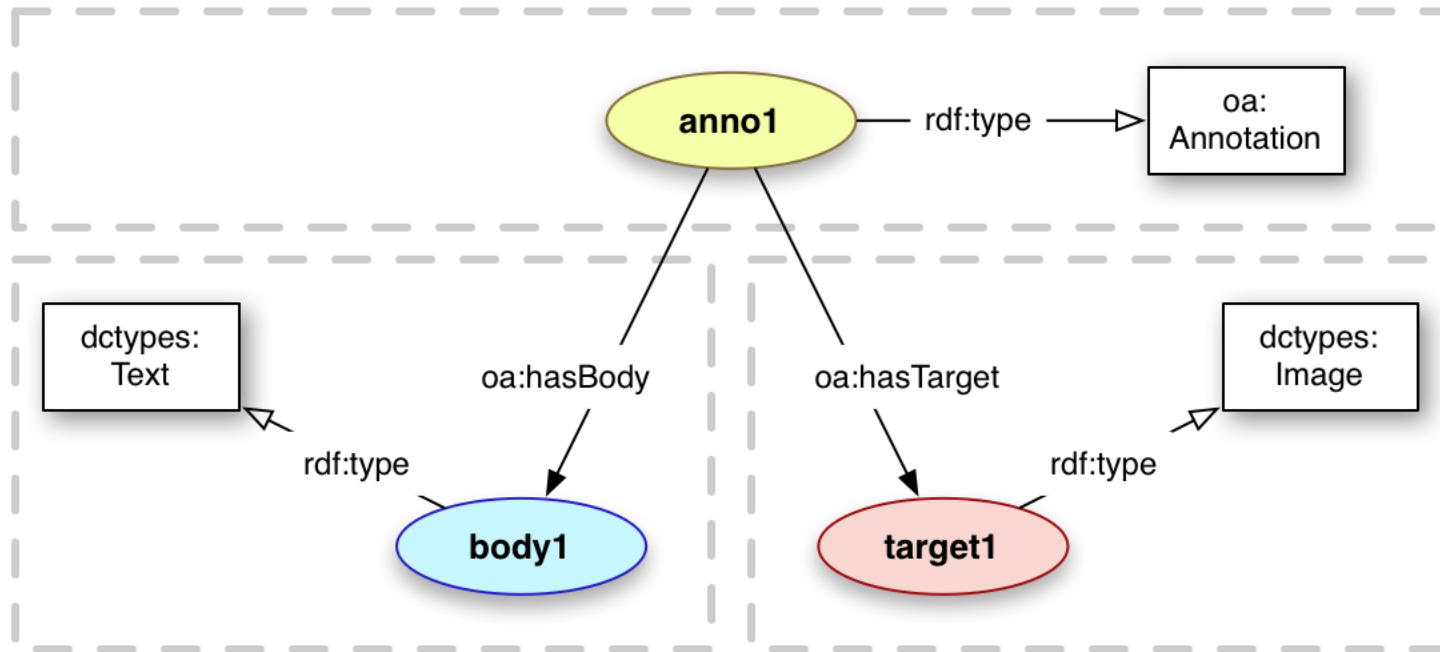
Typing the Body and the Target

In order for the client to have some idea of what to do with the resource, it's helpful to have some type information. So:

- There **SHOULD** be 1 or more content-based classes associated with the Body and the Target resources
- The Dublin Core Types vocabulary is **RECOMMENDED**

Vocabulary Item	Type	Description
dctypes:Dataset	Class	The class for a resource which encodes data in a defined structure
dctypes:Image	Class	The class for image resources, primarily intended to be seen
dctypes:MovingImage	Class	The class for video resources, with or without audio
dctypes:Sound	Class	The class for a resource primarily intended to be heard
dctypes:Text	Class	The class for a resource primarily intended to be read

Text Annotation



```
<anno1> a oa:Annotation ;  
oa:hasBody <body1> ;  
oa:hasTarget <target1> .
```

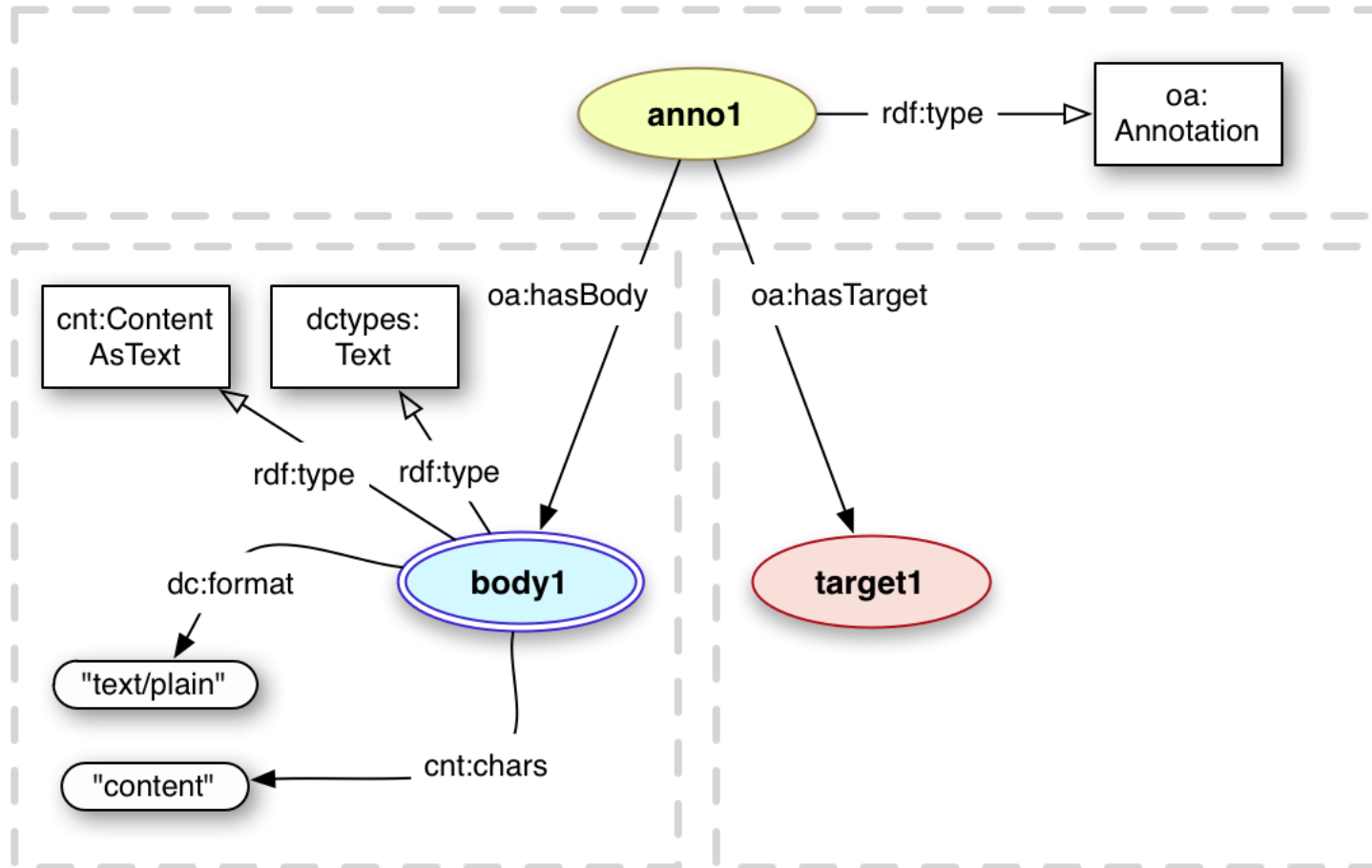
```
<body1> a dctypes:Text .  
<target1> a dctypes:Image .
```

Embedded Textual Bodies

Mirador (and other IIF clients) will expect textual content to be embedded in the annotation. They will not dereference the Body's URI to obtain the text. To embed a textual body:

- Body type is `cnt:ContentAsText`
- Provide an appropriate `dc:format` (e.g., `text/plain` or `text/html`)
- Provide the textual body using the `cnt:chars` property
- Additional properties may be used to describe the Body text (e.g., `dc:language`)

Example: Inline Body



```
<anno1> a oa:Annotation ;  
  oa:hasBody <body1> ;  
  oa:hasTarget <target1> .
```

```
<body1> a cnt:ContentAsText,  
  dctypes:Text ;  
  cnt:chars "content" ;  
  dc:format "text/plain" .
```


A Quick Note on Serialization

- “The RECOMMENDED serialization format is [JSON-LD](#).”
- OA provides a JSON-LD context:
<http://www.w3.org/ns/oa-context-20130208.json>
- The IIF Presentation API context maps IRIs to different terms

```
"hasBody" : {"@type": "@id", "@id" : "oa:hasBody"}
```

in <http://iiif.io/api/presentation/2/context.json> is:

```
"resource": {"@type": "@id", "@id": "oa:hasBody"}
```

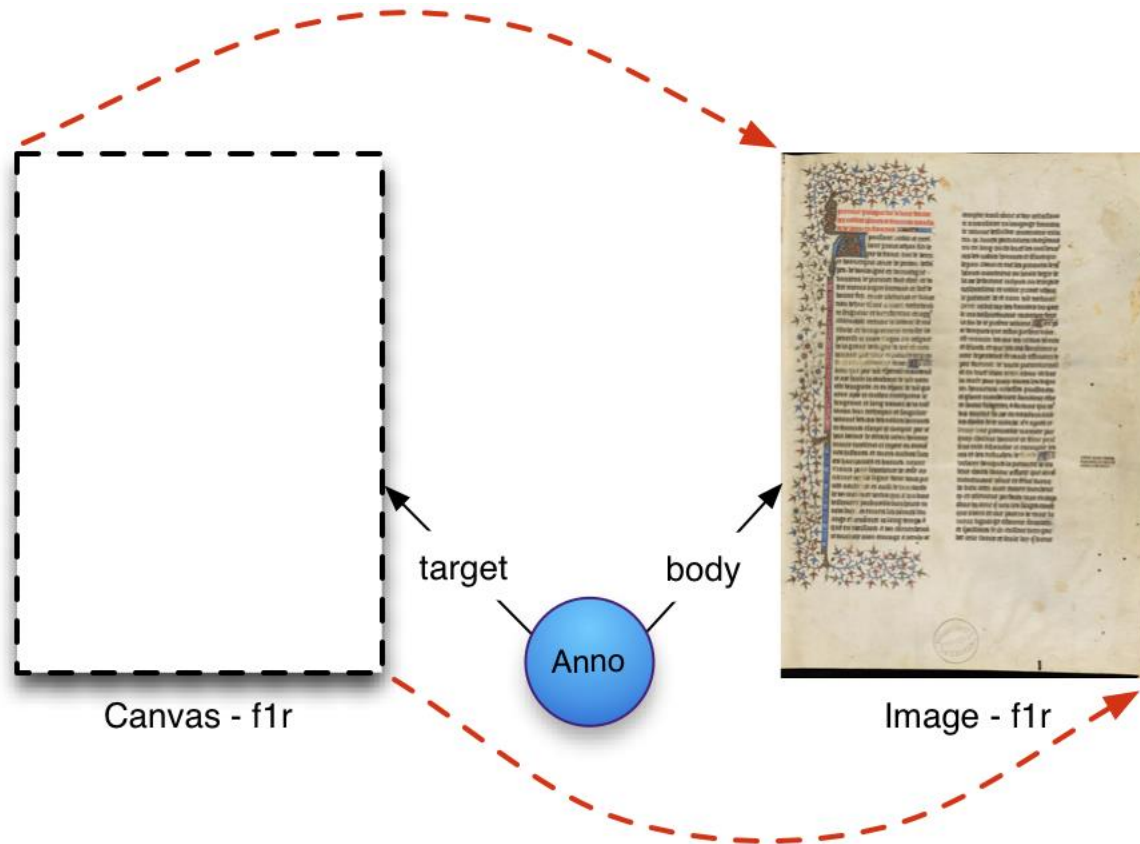
IIIF JSON-LD Serialization of Inline Body

```
<annol> a oa:Annotation ;  
  oa:hasBody <body1> ;  
  oa:hasTarget <target1> .
```

```
<body1> a cnt:ContentAsText ;  
  cnt:chars "Book one..." ;  
  dc:format "text/plain"  
  dc:language "en" .
```

```
"resource": {  
  "@type": "cnt:ContentAsText",  
  "chars": "Book one...",  
  "format": "text/plain",  
  "language": "en"  
}
```

In IIIF the Target is (almost always) a Canvas



When using IIIF we target the Canvas rather than a specific image.

Annotations can target the entire Canvas or a region of it

Rectangular Canvas regions can be defined by adding a fragment selector to the Canvas URI. For example: `#xywh=1000,1200,100,100` would target a rectangle with the upper-left corner located at 1000,1200 on the Canvas, and a width and height of 100.

More complex shapes can be targeted using SVG, as we will see later.

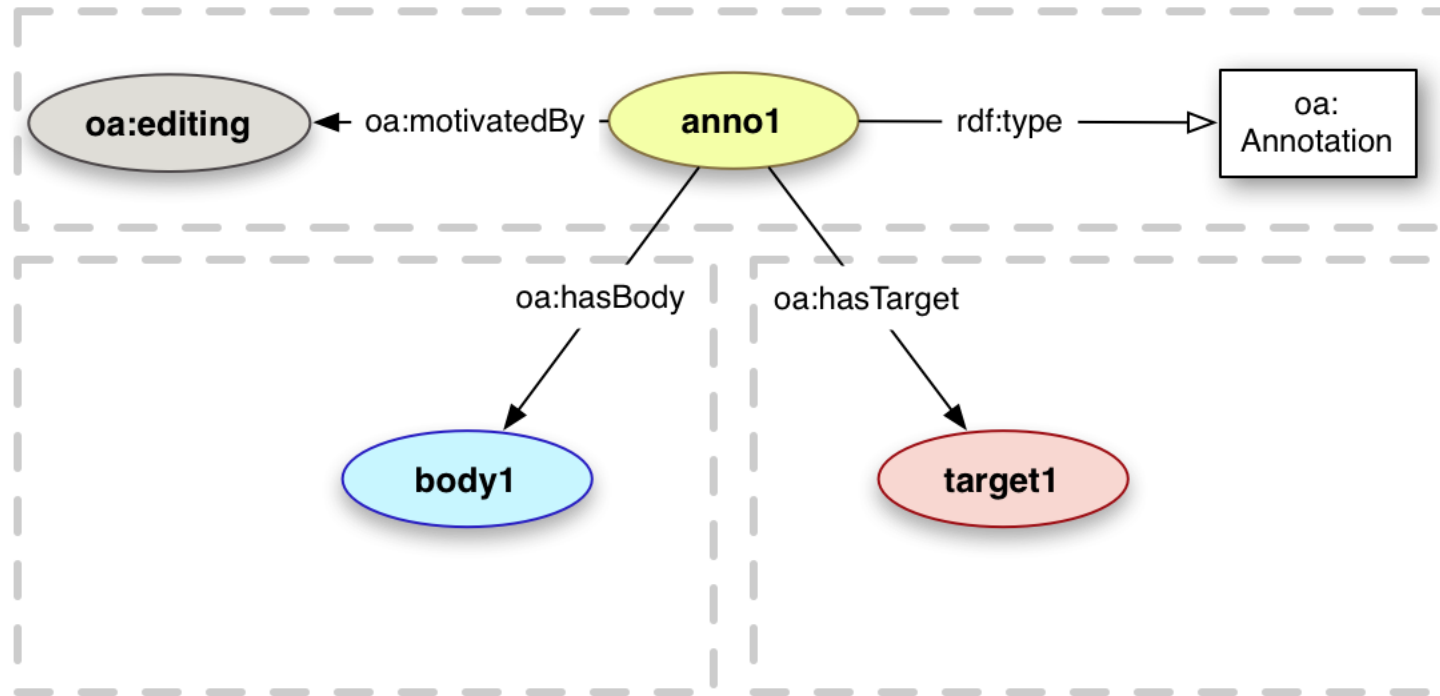
Motivations

- Motivations explain why an annotation was created
- Motivation `sc:painting` is used to “paint” or draw content onto the IIF Canvas
- “Note that all resources which are to be displayed as part of the representation are given the motivation of “sc:painting”, regardless of whether they are images or not. For example, a transcription of the text in a page is considered “painting” as it is a representation of the object, whereas a comment about the page is not.”

Other Motivations

Instances of oa:Motivation		
oa:bookmarking	Instance	The motivation that represents the creation of a bookmark to the target resources or recorded point or points within one or more resources. For example, an Annotation that bookmarks the point in a text where the reader finished reading. Bookmark Annotations may or may not have a Body resource.
oa:classifying	Instance	The motivation that represents the assignment of a classification type, typically from a controlled vocabulary, to the target resource(s). For example to classify an Image resource as a Portrait.
oa:commenting	Instance	The motivation that represents a commentary about or review of the target resource(s). For example to provide a commentary about a particular PDF.
oa:describing	Instance	The motivation that represents a description of the target resource(s), as opposed to a comment about them. For example describing the above PDF's contents, rather than commenting on their accuracy.
oa:editing	Instance	The motivation that represents a request for a modification or edit to the target resource. For example, an Annotation that requests a typo to be corrected.
oa:highlighting	Instance	The motivation that represents a highlighted section of the target resource or segment. For example to draw attention to the selected text that the annotator disagrees with. A Highlight may or may not have a Body resource
oa:identifying	Instance	The motivation that represents the assignment of an identity to the target resource(s). For example, annotating the name of a city in a string of text with the URI that identifies it.
oa:linking	Instance	The motivation that represents an untyped link to a resource related to the target.
oa:moderating	Instance	The motivation that represents an assignment of value or quality to the target resource(s). For example annotating an Annotation to moderate it up in a trust network or threaded discussion.
oa:questioning	Instance	The motivation that represents asking a question about the target resource(s). For example to ask for assistance with a particular section of text, or question its veracity.
oa:replying	Instance	The motivation that represents a reply to a previous statement, either an Annotation or another resource. For example providing the assistance requested in the above.
oa:tagging	Instance	The motivation that represents adding a Tag on the target resource(s). Please see the section on Tagging and Semantic Tags for more information.

Annotation: Body, Target, Motivation



Example: Annotating a Canvas



Canvas dimensions 17367 x 10796



xywh=11540,8790,600,850

Creating a comment annotation

```
{  
  "@context": "http://iiif.io/api/presentation/2/context.json",  
  "@id": "https://example.com/annotation/1",  
  "@type": "oa:Annotation",  
  "motivation": "oa:commenting",  
  "resource": {  
    "@type": "cnt:ContentAsText",  
    "chars": "A Doctor of Musick",  
    "format": "text/plain",  
    "language": "en"  
  },  
  "on": "https://manifests.britishart.yale.edu/canvas/ba-obj-25894-0010-pub#xywh=11540,8790,600,850"  
}
```


of a Master 11 Fellowships and 28 Scholarships

 [Edit](#)  [Delete](#)

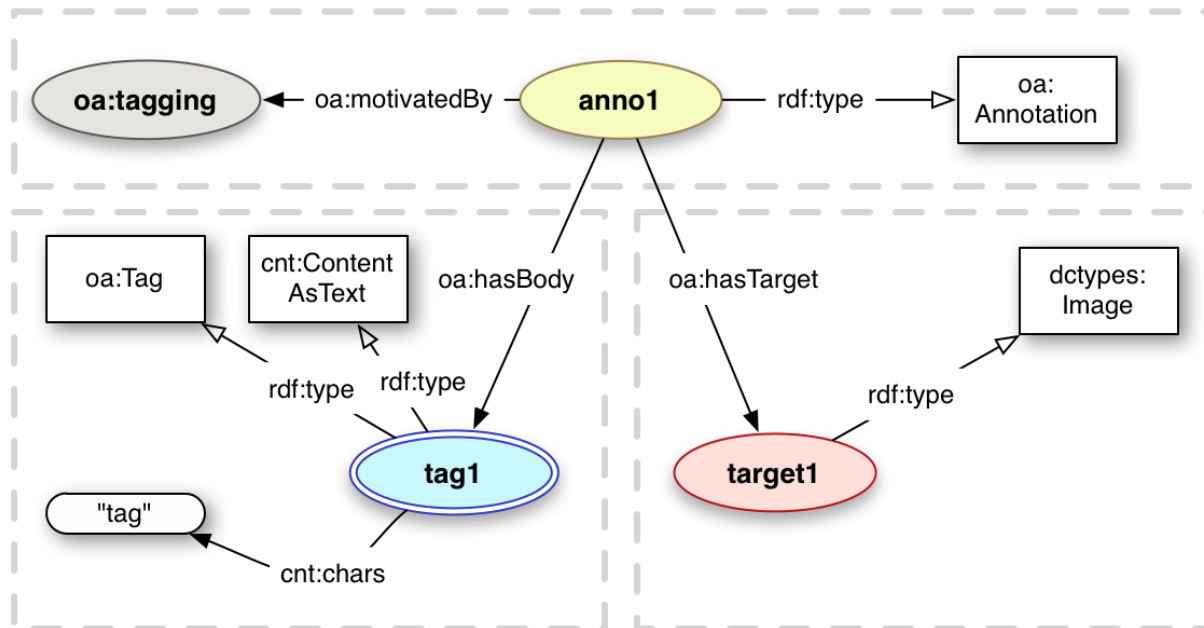
A Doctor of Musick



Tagging: Text Tags and Semantic Tags

Text tags differ from inline textual bodies:

- The annotation's motivation is `oa:tagging`
- The body's type is `oa:Tag`



```
<anno1> a oa:Annotation ;  
oa:motivatedBy oa:tagging ;  
oa:hasBody <tag1> ;  
oa:hasTarget <target1> .
```

```
<tag1> a oa:Tag,  
cnt:ContentAsText ;  
cnt:chars "tag" .
```

Textual Tag

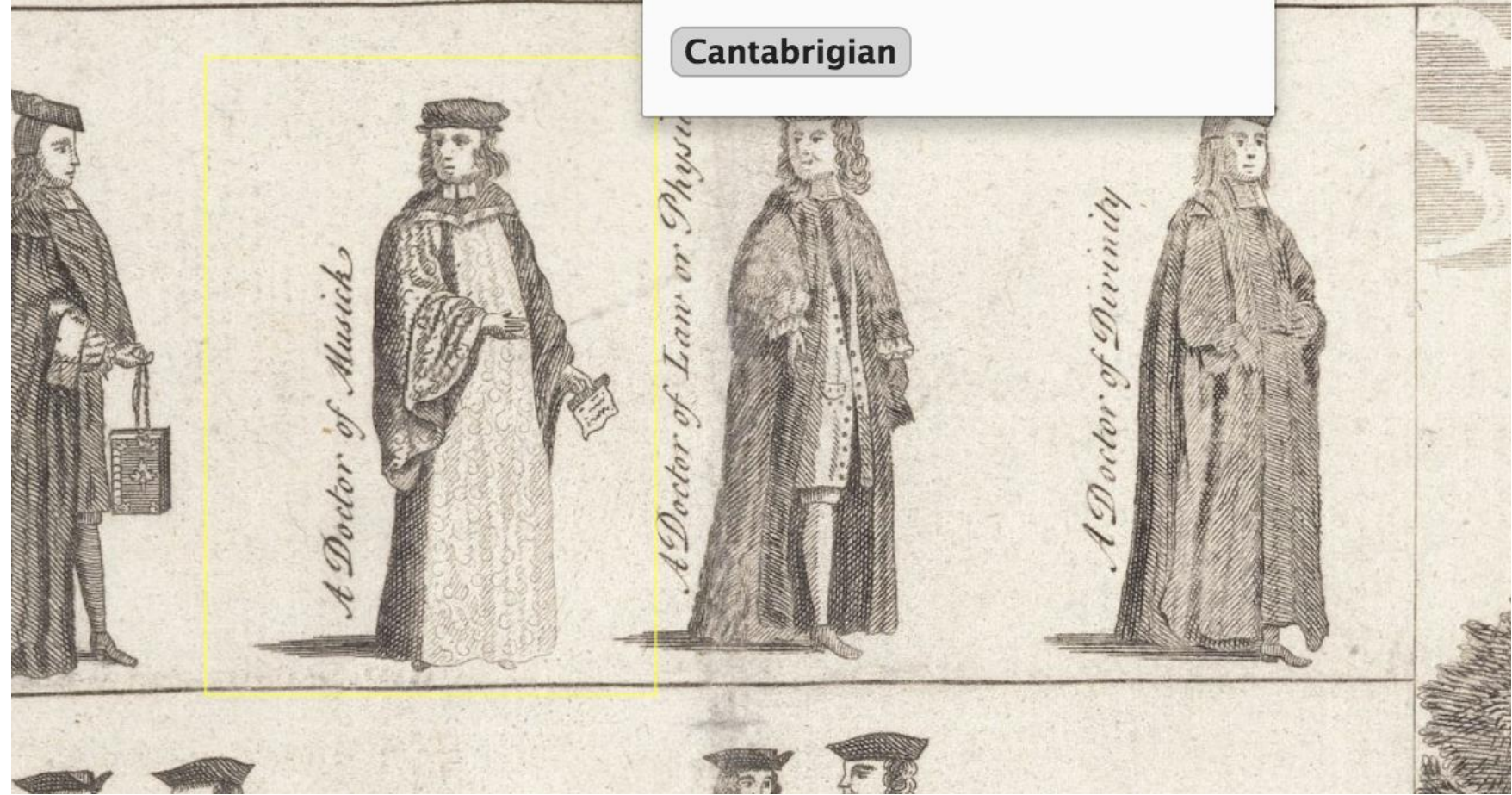
```
{
  "@context": "http://iiif.io/api/presentation/2/context.json",
  "@type": "oa:Annotation",
  "motivation": "oa:tagging",
  "resource": [
    {
      "@type": "cnt:contentAsText",
      "chars": ""
    },
    {
      "@type": "oa:Tag",
      "chars": "Cantabrigian"
    }
  ],
  "on": "https://manifests.britishart.yale.edu/canvas/ba-obj-25894-0010-pub#xywh=11540,8790,600,850"
}
```

88, who by her will gave 5000 pounds for that use, S. Francis
John Breseton have been bountiful Benefactors to it.
consists of a Master 11 Fellowships and 18 Scholarships

made at Can
large as in th
Founder, Its Re

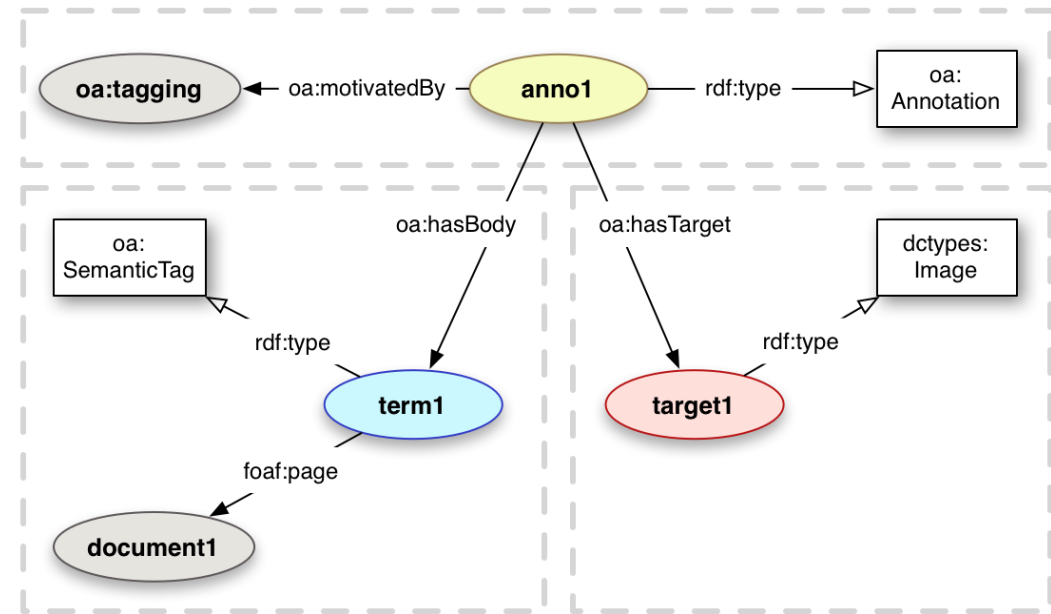
 Edit  Delete

Cantabrigian



Semantic Tags

- Body URI is the URI of the tag
- Body `rdf:type` is `oa:SemanticTag`
- Can use `foaf:page` to reference a web page that describes the tag

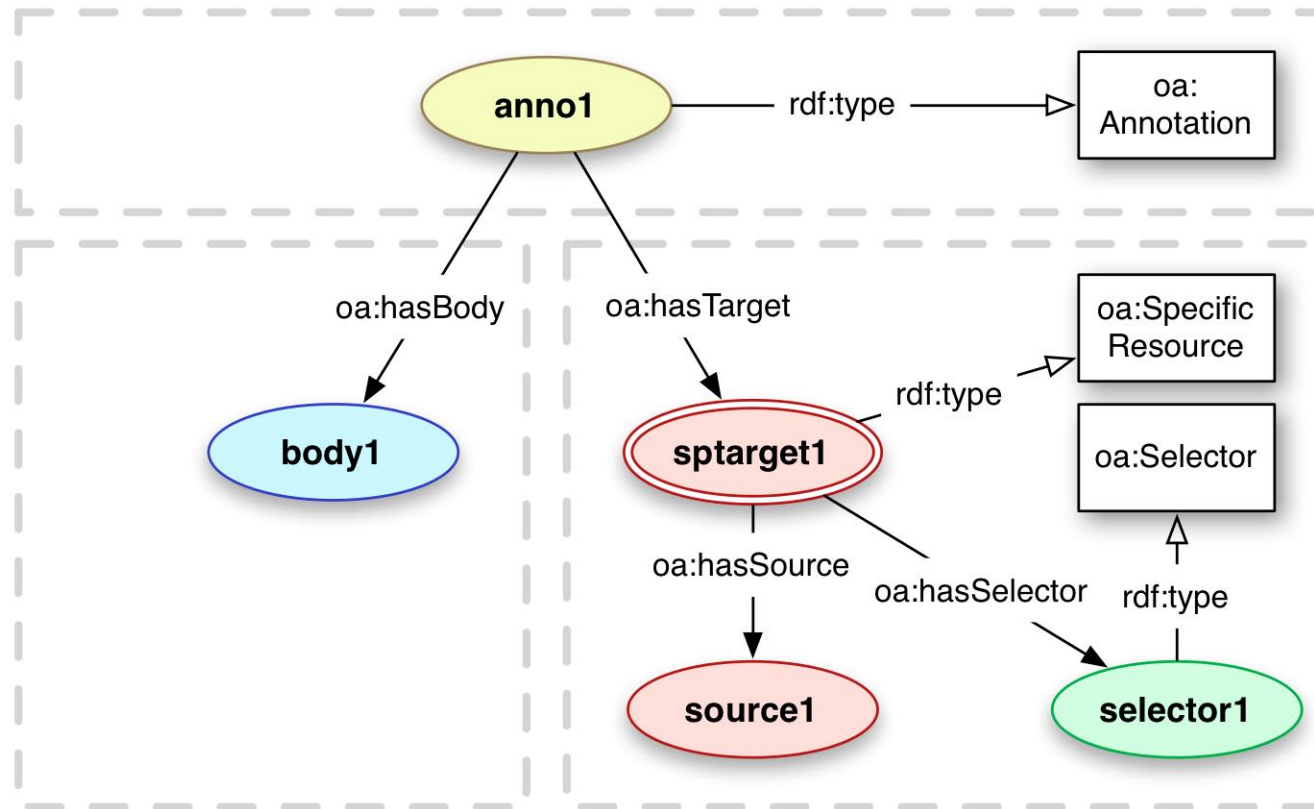


Not supported in Mirador

Specific Resources

- There are many cases where fragment URIs are not sufficient
- An `oa:SpecificResource` can support richer selection criteria
- The `oa:SpecificResource` is associated with both the ultimate target or body resource (“source”) and a subclass of `oa:Selector`
 - `oa:FragmentSelector`
 - `oa:SvgSelector`

Specific Resources



Fragment Selector

The `oa:FragmentSelector` allows for use of a fragment specification without appending a fragment to the target URI. Instead the fragment is included as a property.

```
"@type": "oa:SpecificResource",
"full": "https://manifests.britishart.yale.edu/canvas/ba-obj-1474-0001-pub",
"selector": {
  "@type": "oa:FragmentSelector",
  "value": "xywh=2100,900,1300,2000"
}
```

This selects the same region as:

```
"on": "https://manifests.britishart.yale.edu/canvas/ba-obj-1474-0001-pub#xywh=2100,900,1300,2000",
```


Area Selector: oa:SvgSelector

The `oa:SvgSelector` specifies an SVG path that selects an area of the source resource.



```
"selector": {
  "@type": "oa:SvgSelector",
  "value": "<svg xmlns='http://www.w3.org/2000/svg'><path
xmlns=\"http://www.w3.org/2000/svg\"
d=\"M2664.05147,1142.71965c280.98277,-301.56649 677.3545,-526.13284
1075.64703,-525.86454v0v0c398.29253,0.2683 745.27214,210.3465
1075.64703,525.86454c330.37489,315.51804 458.78742,796.30663
445.54759,1269.5493c-13.23983,473.24267 -168.13202,938.93943 -
445.54759,1269.5493c-277.41556,330.60987 -677.3545,526.13284 -
1075.64703,525.86454c-398.29253,-0.2683 -794.93866,-196.32787 -
1075.64703,-525.86454c-280.70837,-329.53667 -445.47899,-792.55044 -
445.54759,-1269.5493c-0.0686,-476.99886 164.56482,-967.98281
445.54759,-1269.5493z\" /></svg>"
}
```

Multiplicity Constructs

Three methods for grouping resources:

- Choice: client should choose one to display
- Composite: all resources in the set are required for the annotation to be interpreted correctly
- List: a target containing an ordered list of resources, for use in cases where display order is important

Agents

The data model has support for provenance information:

- `oa:annotatedBy`
- `oa:serializedBy`



Mirador does not make use of this data, although it can be customized to do so.

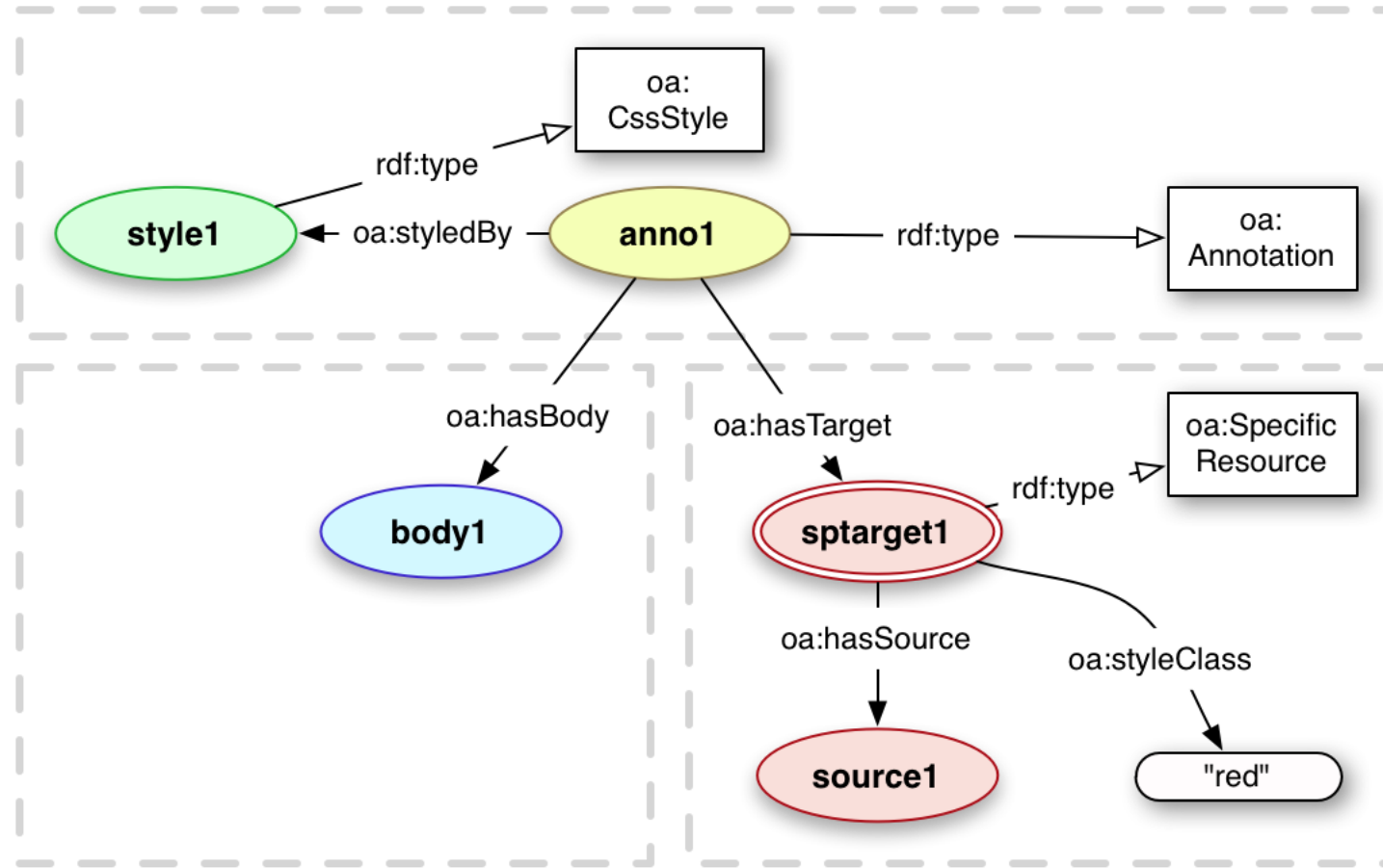
Elucidate annotation server has the ability to store agent data and query (using a custom API) for annotations based on the creator.

Styling and CSS

An annotation may be associated with style information. OA defines a pattern for this:

- The annotation is `oa:styledBy` with an `oa:CssStyle` that defines the CSS styles to be used
- Used only with `oa:SpecificResource` targets/bodies
- The `SpecificResource` references the string name of a CSS class via `oa:styleClass`
- The `oa:styleClass` property can be repeated
- If the referenced style is not defined by the `oa:CssStyle` it must be ignored by clients

Styling and CSS



Styling and CSS

“It must be expected that such clients will not process all, or even any, of the styling hints provided in the CSS Style resource.”



Mirador does not process CSS Styles

Line color/thickness information for the annotation bounding boxes are included in the SVG contained in the `oa:SvgSelector`

State and Scope

- States describe the intended state of the body or target with regard to the annotation. For example, `oa:TimeState` to specify the timestamp of the desired version of the resource, or `oa:HttpRequestState` to retrieve a particular content type when requesting a resource.
- Scopes capture the context of the resource when the annotation was made. For instance, the specific webpage in which an image was being displayed.



The Presentation API defines a property `within` that is used to relate IIIF resources to their containing resources. Mirador will supply `within` to specify the Manifest in which the Canvas was present when the annotation was made.