

LUCA LIANAS

HEALTHCARE FLOWS



---

# OME-SEADRAGON

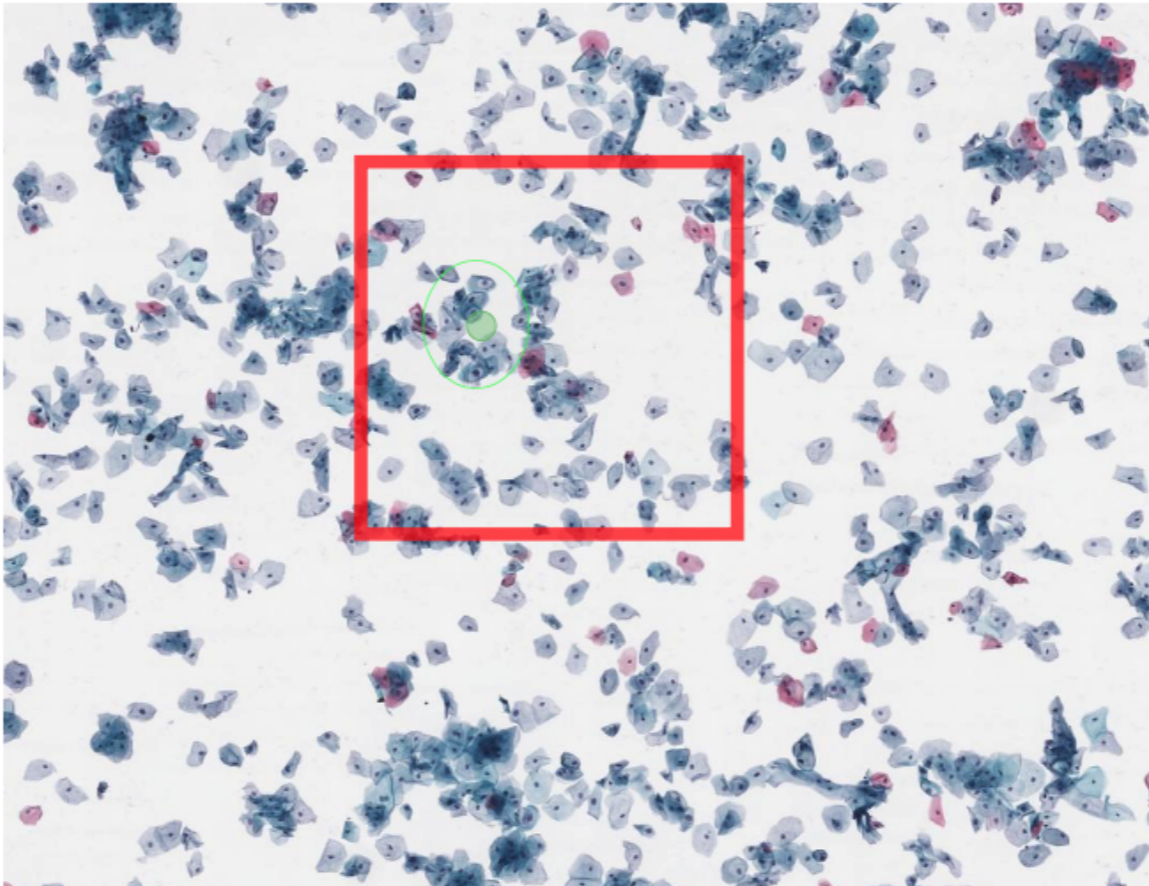
## THE OME-SEADRAGON PLUGIN

- ▶ Developed to be the viewer for the CyTest platform: e-learning system for digital pathology based on Moodle
- ▶ Back-end built as an OMERO.web plugin, based on Deep Zoom Image format specs powered by OME rendering engine and openslide
- ▶ Front-end based on the OpenSeadragon viewer and the paper.js 2D graphics library

**Question 1**  
Correct  
Mark 1.00 out of 1.00

Please click on the cells which you consider as suspect.

+ Add   Edit   X Clear



(x,y): 34452.235898881, 23806.182263368

\* Focus Areas:  
\* Your markers: Marker 1

The set of ROIs (Regions Of Interest) of this question is:

- [632] Correct ✓

To correctly answer you have to select one of the following ROIs: [632]

## BACK-END FUNCTIONS

- ▶ Retrieve data from OMERO (projects, datasets, images, TAGs, ROIs)
- ▶ Generate thumbnails for images
- ▶ Retrieve tiles for the images as expected by OpenSeadragon viewer (based on Deep Zoom Image format)

## DEEP ZOOM IMAGE FORMAT

In order to build a back-end that is compliant with the DZI format we need to

- ▶ produce the XML file with the description of the image, used by the viewer to configure itself
- ▶ properly handle levels and resolutions (by calculating the correct scale factor for each level) for the pyramid as expected by the viewer when loading tiles
  - ▶ openslide has its own DeepZoom adapter, no extra work needed
- ▶ OMERO pyramid images need to be adapted to DZI format

## DEEP ZOOM IMAGE XML DESCRIPTION

```
<Image
  xmlns="http://schemas.microsoft.com/deepzoom/2008"
  Format="jpeg"
  Overlap="1"
  TileSize="256">
    <Size Height="44123" Width="53775" />
</Image>
```

## DEEP ZOOM LEVELS AND SCALE FACTORS

The max zoom level for a DZI is calculated as

$$\log_2(\max(\text{img\_width}, \text{img\_height}))$$

The scale factor for the level L of an image with a max DZI level ML is calculated as

$$0.5^{ML-L}$$

## OMERO PYRAMID VS DZI PYRAMID

Resolutions for levels of OMEMO and DZI pyramids for an Aperio SVS image with a resolution of 53775 x 44123 (DZI levels are 16)

OMERO	DZI
53775 x 44123	53775 x 44123
	26887 x 22061
13443 x 11030	13443 x 11030
	6721 x 5515
3360 x 2757	3360 x 2757
	1680 x 1378
936 x 768	
	840 x 689
	420 x 344
	....



## ADAPTING OMERO TO DZI SPECS

To retrieve a tile for a given DZI level from an OMERO pyramid we must

- ▶ get the lowest level of the OMERO pyramid with a resolution higher than or equal to the one that the DZI level should have
- ▶ get the scale factor between OMERO level and DZI level
- ▶ apply the scale factor to obtain the size and the origin point of the tile from the OMERO image
- ▶ if needed, apply the scale factor to resize the obtained tile

## OPENSLIDE VS OMERO RENDERING ENGINE

- ▶ Openslide engine pros:
  - ▶ supports 3DHISTECH images (.mrxs format)
  - ▶ native support for Deep Zoom format
- ▶ OMERO engine pros:
  - ▶ image formats coverage higher than openslide
  - ▶ z-index and time sequences support

## PERFORMANCE AND SCALABILITY: OME-SEADRAGON CACHE

A caching mechanism can be enabled in order to boost performances and make the system scale.

Cache is based on REDIS, a well known key-value store mostly used as web cache.

Tiles and thumbnails are stored for fast retrieval, cache response time can be 10x faster than rendering engines.

Objects in cache are managed applying the LRU (Less Recently Used) policy.

## FRONT-END FUNCTIONS

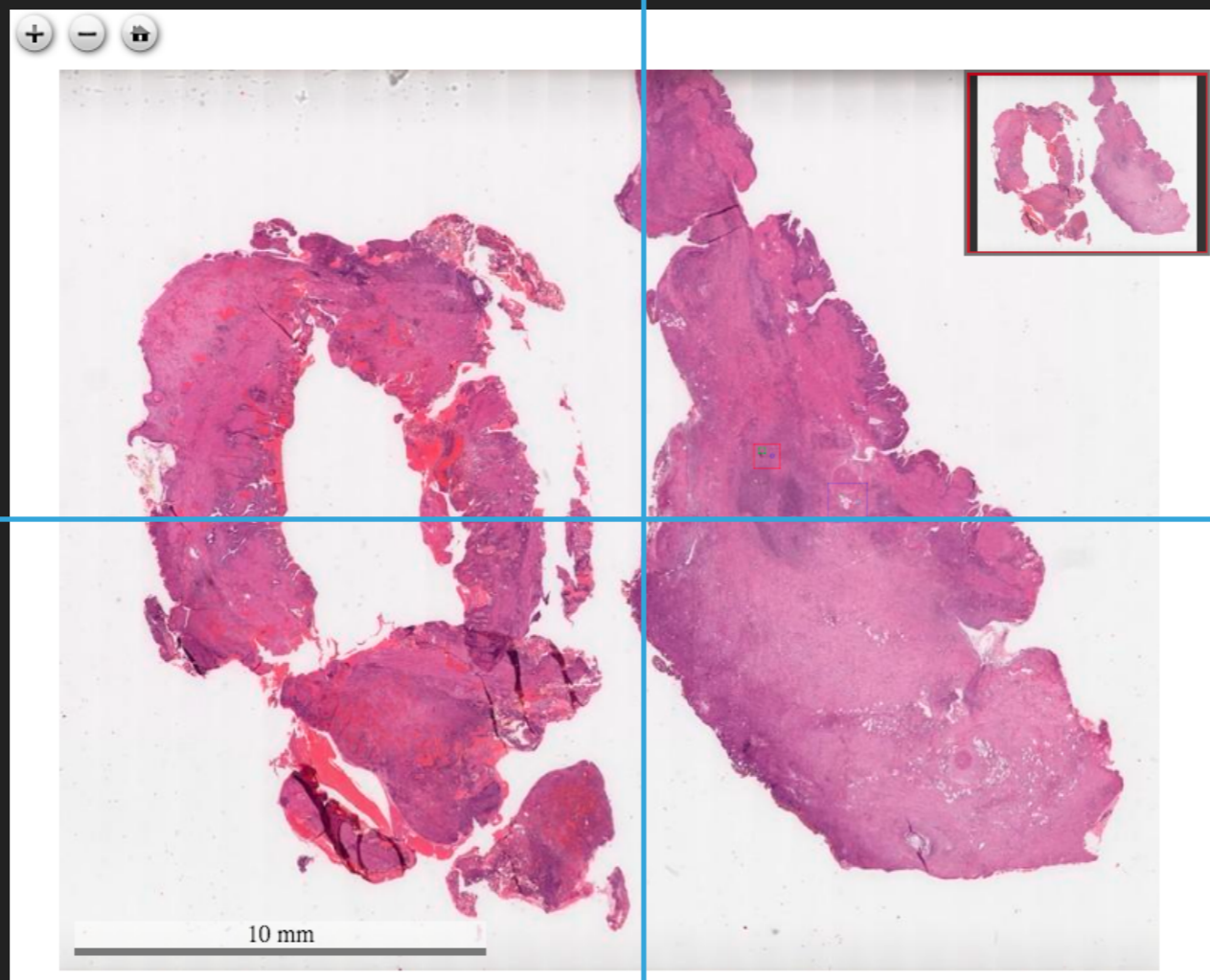
- ▶ Provide simple access to OpenSeadragon and paper.js functions by wrapping them in Javascript classes
- ▶ Automatic propagation of events (zoom, panning) occurred in OpenSeadragon to paper.js
- ▶ Shapes wrapped in classes to simplify management
- ▶ Interactive tools manager that works as an extension of the paper.js canvas wrapper

## OPENSEADRAGON AND PAPER.JS MANAGEMENT

- ▶ Two different HTML elements, a DIV for OpenSeadragon and a CANVAS for paper.js, overlapping (using CSS)
- ▶ OpenSeadragon viewer is managed by a class called **ViewportController** which handles:
  - ▶ events propagation to paper.js
  - ▶ coordinates and zoom conversions
- ▶ paper.js canvas is managed by a class called **AnnotationsController** which handles:
  - ▶ shapes drawing and management
  - ▶ interactive events driven by user interaction involving shapes (drawing of new ones or update existing ones)

## OPENSEADRAGON AND PAPER.JS INTEGRATION

To easily manage coordinates on the paper.js canvas, when the canvas is bound to the viewer its origin is moved to the center of the image and its zoom factor is adapted to the one of OpenSeadragon.



## DRAWING OMERO SHAPES

Shapes retrieved from OMERO's ROIs are wrapped in dedicated objects; the AnnotationsController then:

- ▶ creates the proper paper.js object for each shape
- ▶ applies, if needed, the transformation matrix related to the shape
- ▶ translates the shape by the offset used to calculate the position of the origin of the paper canvas
$$x = \text{img\_width} / 2 \quad y = \text{img\_height} / 2$$
- ▶ draws the shape to the canvas

## INTERACTIVE TOOLS

- ▶ User interaction with the paper.js canvas is managed by the `AnnotationsEventsController` class
- ▶ Interactive tools are bound to the `AnnotationsController` by extending the object
- ▶ Currently we have only an interactive tool used to place circular markers on the image
- ▶ Polygons drawing tools in progress



## TODO LIST

- ▶ Add proper z-index support
- ▶ Add an authentication mechanism (currently using OMERO.web public user feature)
- ▶ Build drawing tools
- ▶ Write documentation
- ▶ Test, test, test, test!!!

Source code (for stable releases) can be found in GitHub

[https://github.com/crs4/ome\\_seadragon](https://github.com/crs4/ome_seadragon)

New features and works in progress can be found in my fork

[https://github.com/lucalianas/ome\\_seadragon](https://github.com/lucalianas/ome_seadragon)