High quality digital tool 1: WSI viewer that can be integrated in assessment software

Technical Overview

Authors: Friedrich Pawelka, Christian Mol

Münster, April 2023

Co-funded by the European Union

The creation of these resources has been (partially) funded by the ERASMUS+ grant program of the European Union under grant no. 2020-1-DE01-KA226-HE-005813. Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

About this document

The target audience of this technical overview are people with a technical background that want to use our adapted WSI viewer into their own assessment software.

In this document, we will first give an overview about the digital tool that we selected and how it was adapted to meet the project goal of supporting collaborative learning of viewing and decisionmaking skills (Chapter "Concept"). After that, we show concrete examples how we integrated the adapted tool into existing assessment software (Chapter "Examples of integration"). Finally we provide a short quickstart guide to integrate the adapted tool into your own assessment software and links to resources on our GitHub repository (Chapter "Integration in own assessment software").

Concept

The resulting digital tool needs to be easy to use and integrable into existing assessment software. Furthermore, it has to be extendable by technical staff to meet custom requirements. It also should provide all relevant features that teachers in this field need.

We decided to use a combination of JavaScript and iframe-HTML-Elements to make our tool easy integrable in a wide variety of existing web based software.

To find a good basis for our digital tool, we first did a needs-analysis with the stakeholders of the project (students, pathologists, teachers etc.). Second, we did a technical evaluation of existing WSI software considering three criteria: 1) free to use and modify 2) active developer community and 3) usage of common high-resolution image library. The full evaluation process is documented here: https://medicampus.uni-muenster.de/ccel/uploads/cLovidKickoff_210610_Part2.pdf.

As result of this evaluation process, we chose the WSI viewer "OMERO.iviewer"[1] from the Open Microscopy Environment (OME)[2] as the basis for our digital tool.

OMERO.iviewer as basis

This WSI viewer is part of OMERO, a client-server software package for managing, visualizing and analyzing microscopy images and associated metadata, which is widely used in the microscopy field.

All software packages from OME are hosted on GitHub[3] and we forked the repository for the OMERO.iviewer (https://github.com/ome/omero-iviewer) to a new repository in our project group on GitHub (https://github.com/clovid/omero-iviewer). This repository contains the adaptions to the viewer that are necessary to integrate it into existing assessment software.

The result is an adapted version of the OMERO.iviewer that can be integrated via an iframe-HTML-Element and a bit of JavaScript into other software.

Integration via iframe-HTML-Element

The main adaption that we made to the OMERO.iviewer was to provide the ability to make it integrable via an iframe-HTML-Element and to control relevant features (zooming, panning and manipulation of annotations) from the external application.

[1] https://www.openmicroscopy.org/omero/iviewer/

[2] https://www.openmicroscopy.org/

[3] https://github.com/ome

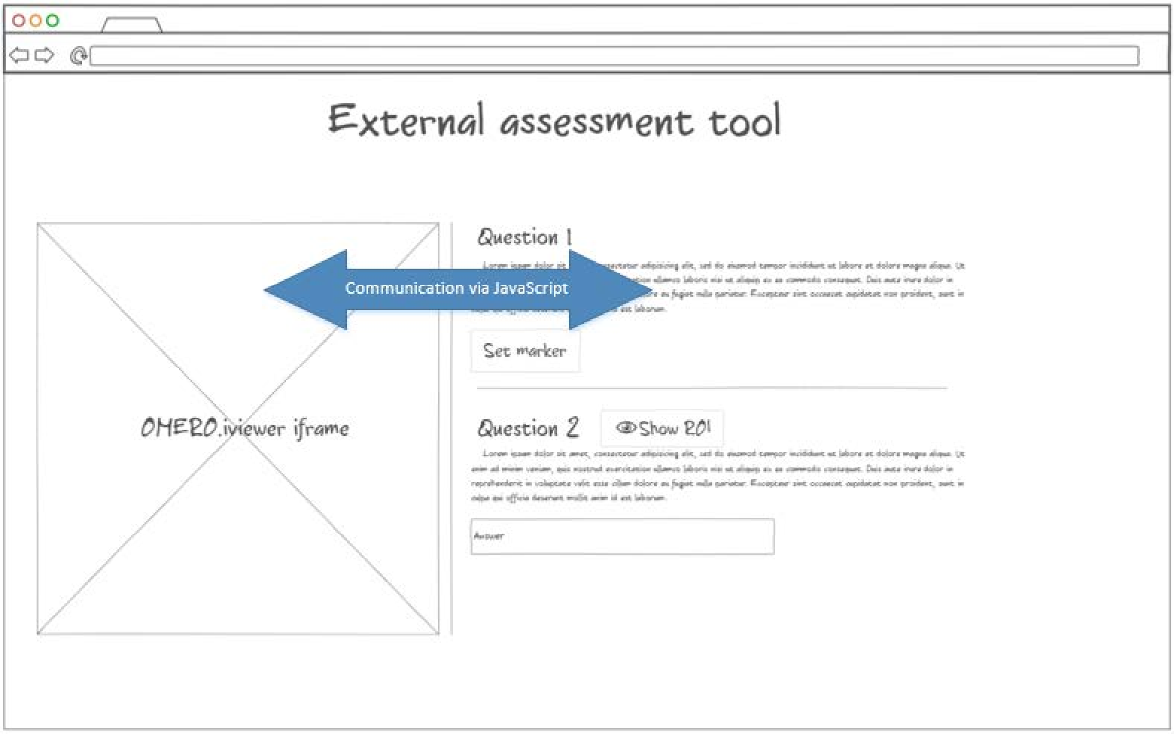
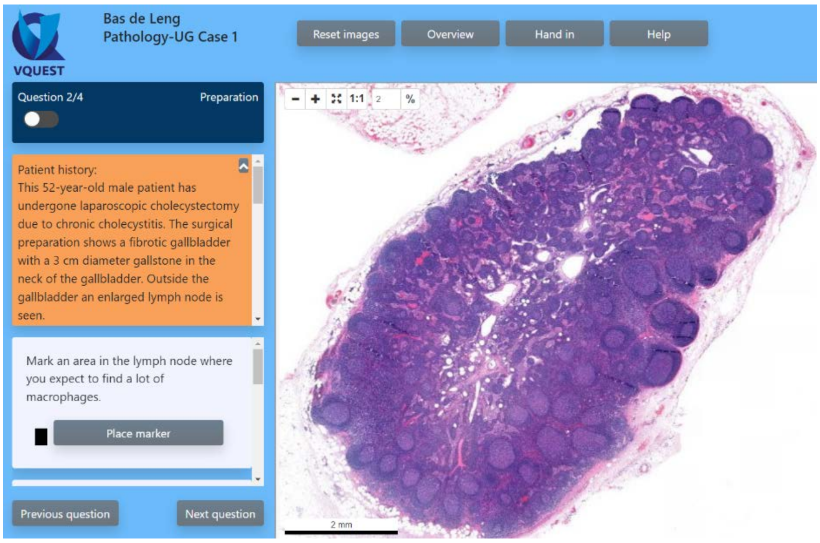


Figure 1 Concept of integration and communication via iframe and JavaScript

For a detailed description of the adaptions and what features can be controlled, see the API documentation: https://github.com/clovid/omero-iviewer/blob/main/README.rst#api. Some examples and instructions how to use these adaptions can be found in the next chapters.

Examples of integration

Our digital tool (the adapted WSI viewer) can be used for various use cases. Within the cLovid project we integrated it in existing software with focus on 1) individual work (Assessment Software VQuest[4]) and 2) group review (Learning Dashboard PRISMA[5]).



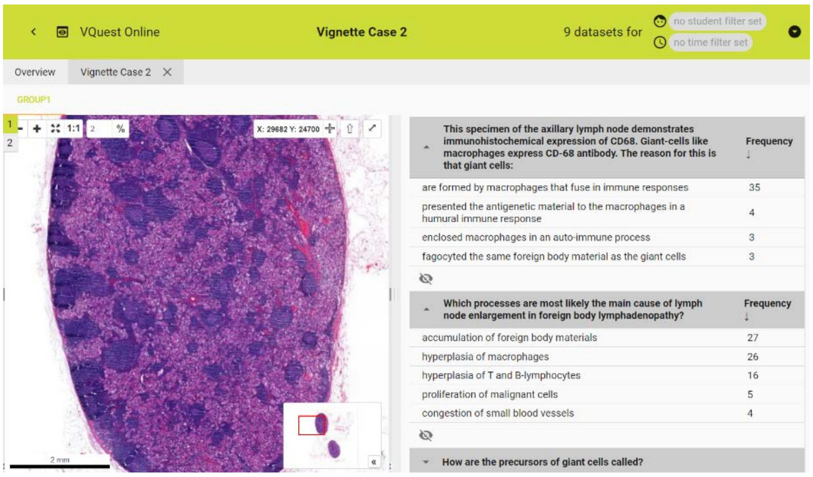
Patient history:
This 52-year-old male patient has undergone laparoscopic cholecystectomy due to chronic cholecystitis. The surgical preparation shows a fibrotic gallbladder with a 3 cm diameter gallstone in the neck of the gallbladder. Outside the gallbladder an enlarged lymph node is seen.

Mark an area in the lymph node where you expect to find a lot of macrophages.

Figure 2 WSI viewer integrated in assessment software

[4] https://vquest.nl/

[5] https://github.com/clovid/prisma



This specimen of the axillary lymph node demonstrates immunohistochemical expression of CD68. Giant-cells like macrophages express CD-68 antibody. The reason for this is that giant cells:	Frequency ↓
are formed by macrophages that fuse in immune responses	35
presented the antigenetic material to the macrophages in a humural immune response	4
enclosed macrophages in an auto-immune process	3
fagocyted the same foreign body material as the giant cells	3

Which processes are most likely the main cause of lymph node enlargement in foreign body lymphadenopathy?	Frequency ↓
accumulation of foreign body materials	27
hyperplasia of macrophages	26
hyperplasia of T and B-lymphocytes	16
proliferation of malignant cells	5
congestion of small blood vessels	4

How are the precursors of giant cells called?

Figure 3 WSI viewer integrated in Learning Dashboard

Integration in your own assessment software

In this chapter we show briefly what needs to be done to integrate our adapted WSI viewer into your own assessment software.

Requirements

- Running OMERO.web v5+ instance with access to configuration
  - See https://omero.readthedocs.io/en/v5.6.7-3/developers/Web/Deployment.html for installation instructions from OMERO
- Installed OMERO.iviewer-cLovid
  - See https://github.com/clovid/omero-iviewer/blob/main/README.rst#installation for installation instructions
- Both OMERO and external assessment software accessible via **https**

Integration scenarios

Some further configuration is needed depending on

- the *access* that is necessary to show WSI (public available images via public user[6] or images only available for logged in OMERO users)
- *actions* that should be done (only showing images and annotations or also creating new annotations, e.g. in marker questions) and
- *domain relationship* between external software and OMERO (same-origin, same-site or cross-origin[7]).

Access: Public images

Images on OMERO that are available via the public user and can be viewed by everyone. See https://omero.readthedocs.io/en/stable/sysadmins/public.html#configuring-public-user for more information.

Access: Protected images

Images on OMERO that are only visible for authenticated users.

[6] https://omero.readthedocs.io/en/v5.6.7/sysadmins/public.html#configuring-public-user
[7] https://web.dev/same-site-same-origin/

Action: Show annotations

Possibility to show and hide one or more existing annotations on images

Action: Create annotations

Possibility to create new annotations (e.g. arrow) on images

Domain: same-origin

External assessment software and OMERO share the same (sub-)domain including port and scheme:

- External assessment software: https://omero.example.com/assessment
- OMERO: https://omero.example.com

Domain: same-site

External assessment software and OMERO share the same domain including Port and Scheme:

- External assessment software: https://assessment.example.com
- OMERO: https://omero.example.com

Domain: cross-origin

External assessment software and OMERO do not share any domain parts:

- External assessment software: https://assessment.example.com
- OMERO: https://omero.clovid.org

Configuration steps

Depending on the integration scenario (see above), different configuration steps are necessary. The table below shows the relevant configuration steps for different combinations of integration scenarios.

Minimal configuration is needed for public images, where annotations should be only shown and that are hosted on the same-origin (see https://omero.clovid.org/examples/02.html). Most complex configuration is needed for protected images, where annotations also should be created and that are hosted on different domains (cross-origin).

		same-origin	same-site	cross-origin
Public images	Show annotations	-	*Cross-origin iframe*	*Cross-origin iframe*
	Create annotations	*CSRF-Middlware* or *Pixel*	*Cross-origin iframe, CSRF-Middleware* or *Pixel*	*Cross-origin iframe, Proxy for Cookies* or *Cross-origin cookies*
Protected images	Show annotations	*OMERO user*	*Cross-origin iframe, OMERO user*	*Cross-origin iframe, OMERO user, Proxy for Cookies* or *Cross-origin cookies*
	Create annotations	*CSRF-Middleware* or *Pixel, OMERO user*	*Cross-origin iframe, OMERO user, CSRF-Middleware* or *Pixel*	*Cross-origin iframe, OMERO user, Proxy for Cookies* or *Cross-origin cookies, CSRF-Middleware* or *Pixel*

Cross-origin iframe

To allow embedding the OMERO.iViewer via an iframe, OMERO.web need to be configured accordingly:

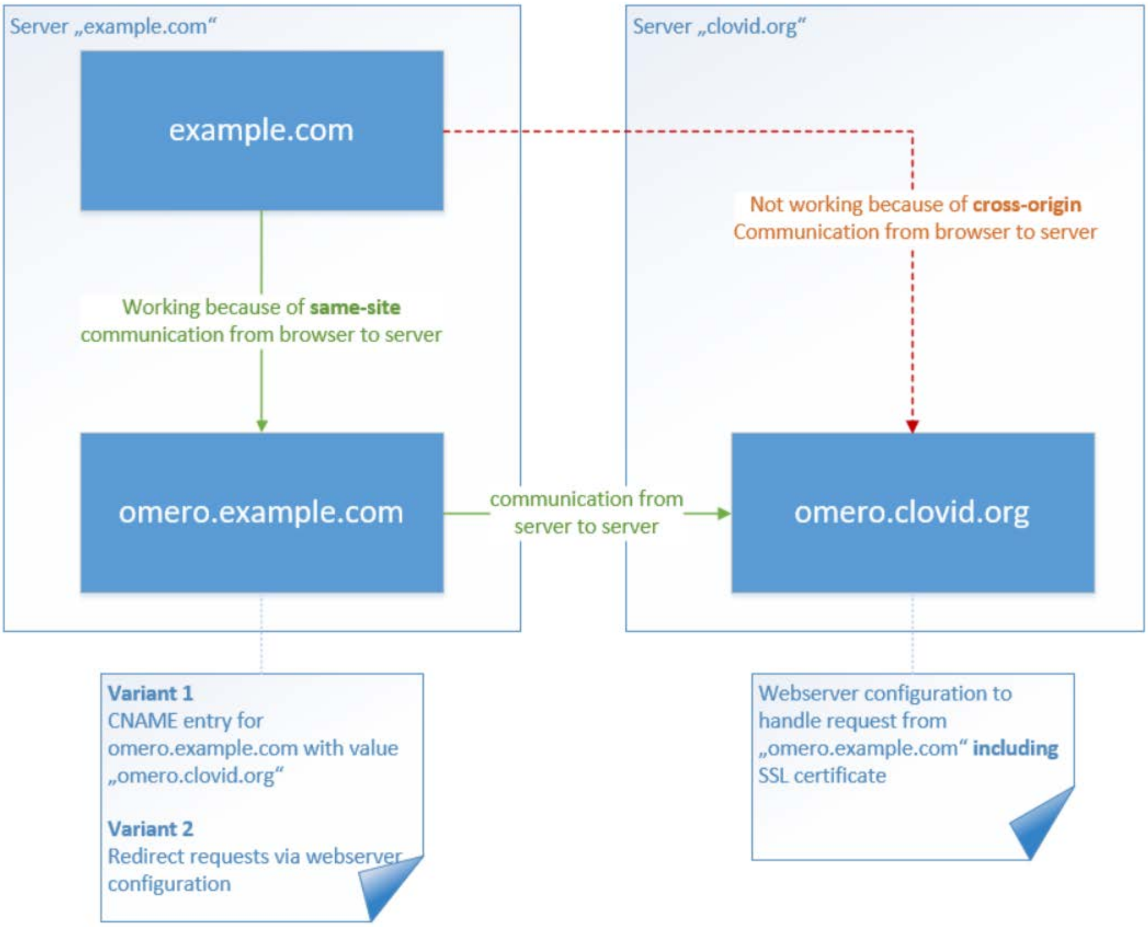
```
config remove -- omero.web.middleware '{"index": 6, "class":
"django.middleware.clickjacking.XFrameOptionsMiddleware"}'
```

Note: for security reasons this should be combined with the `frame-ancestors` *Content-Security-Policy (CSP), see: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy/frame-ancestors.*

Proxy for cookies

To avoid multiple configuration changes in cross-origin settings a proxy domain should be set up. This prevents multiple problems with cookies, especially with regard to the modern browser behavior to disable 3rd party cookies by default.

A proxy domain can be realized either with a CNAME DNS entry or with a webserver entry (e.g. `proxy_pass` for Nginx). See the following illustration for both variants.


Server „example.com"
example.com
Working because of same-site communication from browser to server
omero.example.com
communication from server to server
Server „clovid.org"
Not working because of cross-origin Communication from browser to server
omero.clovid.org
Variant 1
CNAME entry for omero.example.com with value „omero.clovid.org"
Variant 2
Redirect requests via webserver configuration
Webserver configuration to handle request from „omero.example.com" including SSL certificate

See https://github.com/clovid/integration-example#about-3rd-party-cookies for more information about 3rd party cookies.

Cross-origin cookies

Note: This is only needed if "Proxy for cookies" is not realized.

To allow cookies in a cross-origin setting, OMERO.web. needs to be configured accordingly:

```
config set -- omero.web.session_cookie_secure 'true'
config set -- omero.web.csrf_cookie_secure 'true'
config append -- omero.web.django_additional_settings
'["SESSION_COOKIE_SAMESITE", "None"]'
config append -- omero.web.django_additional_settings
'["CSRF_COOKIE_SAMESITE", "None"]'
```

Furthermore, a django extension needs to be installed on OMERO.web:

```
/opt/omero/web/venv3/bin/pip install 'django-cookies-samesite'
```

CSRF-Middleware

When creating an annotation, the iViewer within the iframe sends a POST request to OMERO. By default all POST request need to be sent with a CSRF cookie. However, this cookie is not set when accessing the iViewer.

To accept POST requests without the CSRF cookie the relevant middleware need to be removed from OMERO.web:

```
config remove -- omero.web.middleware '{"index": 4, "class":
"django.middleware.csrf.CsrfViewMiddleware"}'
```

Note: for security reasons the "Pixel" approach should be preferred.

Pixel

To set the CSRF cookie (see *CSRF-Middleware*), we need to call the base OMERO url additionally to the iViewer url. This will set a CSRF cookie that could be used during the POST requests. An easy way to realize this is to add a small iframe on the same page in the external assessment software where the iViewer is integrated:

```
<iframe src=”https://omero.example.org” width=”1” height=”1”
frameborder=”0”></iframe>
```

OMERO user

To show protected images, an OMERO user is needed, that has read-annotate[8] access to the relevant WSI. This user will be used to create a session token via the OMERO API to give direct access to the WSI (see *Integration steps*).

Note: A cleaner approach could be to create for each user of the external assessment software an own OMERO account on the fly via that API and grant access to the relevant WSI.

Integration steps

When the relevant configuration steps are completed, the integration contains two main steps and one additional, if protected images are used.

Step 1: Iframe

```
<iframe allow="fullscreen" width="100%" height="100%" style="border: 0;"
src="https://omero.clovid.org/iviewer_clovid/?[GET params]"></iframe>
```

See https://github.com/clovid/omero-iviewer/blob/main/README.rst#embedding for required GET params.

[8] https://omero.readthedocs.io/en/v5.6.7/sysadmins/server-permissions.html#term-Read-annotate

Step 2: Iframe communication with JavaScript

We use JavaScript messages to initialize the connection between external assessment software and iViewer. After the initialization process we can send actions to trigger actions inside iViewer (e.g. zooming, panning or annotation creation).

See https://github.com/clovid/omero-iviewer/blob/main/README.rst#communcation-via-javascript for a code example.

Step 3: Session token generation

For protected images a routine to generate a session token[9] needs to be generated via the OMERO API and provided in the iframe url.

See https://github.com/clovid/omero-iviewer/blob/main/README.rst#session-token for more information.

Minimal Examples

We prepared several working examples that can be uses as basis for implementation in your own assessment software.

- https://omero.clovid.org/examples/01.html: Basic example with all functionalities
- https://omero.clovid.org/examples/02.html: Focus on individual work: create annotations (without saving)
- https://omero.clovid.org/examples/03.html: Focus on group review: show teacher annotations
- https://omero.clovid.org/examples/04.html: Focus on group review: show students annotations

[9] https://omero.readthedocs.io/en/stable/developers/Web/Webclient.html#reusing-omero-sessions