



YAMI WEB MEDICAL VIEWER

CREATIS



INSA INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
LYON

CENTRE
DE LUTTE
CONTRE LE CANCER
**LEON
BERARD**



Numido

- ▶ Medical images
- ▶ Web application
- ▶ Run simulations

David Sarrut and Thomas Baudier (Creatis) are developing Numido, a web application that gives a remote access to medical data, image processing workflow, makes links between them and runs simulations.

Dashboard

Patients

DicomSerie

DicomStruct

Results

Tasks

8 new Dicom images available. [Click here to review.](#)5 new Dicom contours available. [Click here to review.](#)Show entriesSearch:

Patients 	Dicom 	Contours 	Calibration 	TAC 	Integration 	Dose (MC) 
P1 Patient1_radioembolisation Tc-99m 226 MBq						
P2 Patient2_radioembolisation Tc-99m 370 MBq						
P3 Patient3_radioembolisation Tc-99m 252 MBq						
P4 Patient4_radioembolisation Tc-99m 216 MBq						
P5 Patient5_radioembolisation Tc-99m 126 MBq						

Showing 1 to 5 of 5 entries

Previous **1** Next

Numido

Example of workflow



Web visualization - Specifications

Imperative

- ▶ JavaScript
- ▶ OpenSource
- ▶ Open: **Dicom series, MHD/raw data**

Major specifications

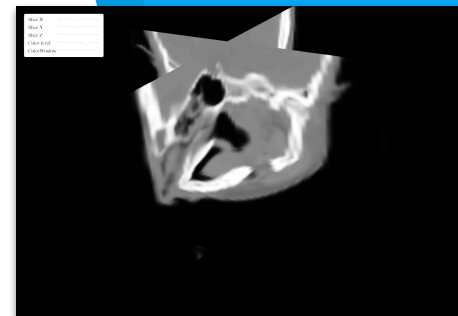
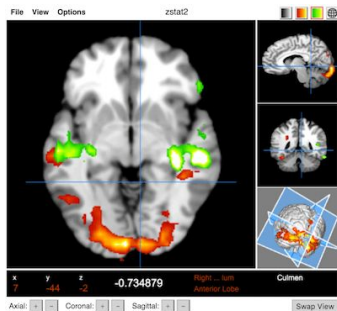
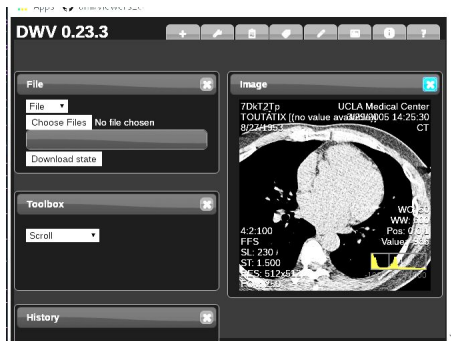
- ▶ Display: fusion, Overlay, ROI (filled or not)
- ▶ Window/Level/LUT interactivity
- ▶ Display pointed value
- ▶ Adapt transparency for fusion
- ▶ Adapt overlay window

Optional specifications

- ▶ Open: RTStruct or mesh, nii (and other medical images)
- ▶ Use time (4D)
- ▶ Can display Sagittal, Coronal, Axial slices (one at a time)



Web visualization



DWV

Papaya

OHIF

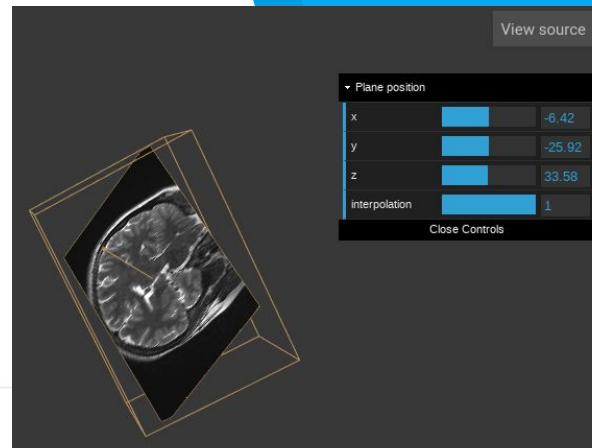
VTK JS

Name	License	DICOM	NIFTI	MHD	Documentation	Maintained	Comment
DWV	GPLv3	Yes	No	No	-	Yes	
Papaya	Copyright	Yes	Yes	No	+	Yes	
OHIF	MIT	Yes	No	No	+	Yes	"Too" complete
AMI	MIT	Yes	Yes	Yes	+	Yes	Lot of examples
VTK JS	Copyright	Yes+/-	Yes	Yes	+	Yes	



AMI

A* Medical Imaging toolkit.
FNNDS Boston Children's Hospital
Author : Nicolas Rannou



Features

✓ READY ◆ IN PROGRESS OR LIMITED SUPPORT ✗ ON ROADMAP

Capabilities	Volumes	Meshes	Widgets
✓ 2D Visualization	✓ Dicom	✓ VTK (THREEJS)	◆ Handle (2D/3D)
✓ 3D Visualization	✓ NRRD	✓ STL (THREEJS)	◆ Probe (2D/3D)
✓ Volume Rendering	✓ Nifti	✓ TRK	◆ Ruler (2D/3D)
✓ Lookup Tables	✓ MHD/(Z)RAW	✓ FSM	◆ Orientation (2D/3D)
◆ Label Maps	✓ MGH/MGZ	✗ CURV	◆ Angle (2D/3D)
	✗ JPEG		



30 FPS (0-31)

Data

patient : Patient1_Radioembolisation
date : 2018-01-04 15:09
injection : Tc-99m 2018-01-04 15:09 226 MBq
modality : NM
study : radioembolisation

Basic controls

Prop Click

Move Click and Drag
Move : click on Wheel

Zoom Click and Drag
Zoom In : + , i , Ctr + Wheel down
Zoom Out : - , o , Ctr + Wheel up

Slices Click and Drag
Slice Up : ↑ , Wheel Up
Slice Down : ↓ , Wheel Down

Window Click and Drag
Windowing : Right Click

YAMI - Medical Viewer

[radioembolisation] Patient1_Radioembolisation

image : 111 HU
fusion : 20170 counts

Main image

windowWidth

windowCenter

invert

window

Lut Color

index

Fusion

lut

Threshold

Opacity min

Opacity max

Camera

Close Controls



30 FPS (0-30)

YAMI - Medical Viewer

[radioembolisation] Patient1_Radioembolisation

Data

patient : Patient1_Radioembolisation
date : 2018-01-04 15:09
injection : Tc-99m 2018-01-04 15:09 226 MBq
modality : NM
study : radioembolisation

Basic controls

Pres Click

Move Click and Drag
Move : click on Wheel

Zoom Click and Drag
Zoom In : + , i , Ctrl + Wheel down
Zoom Out : - , o , Ctrl + Wheel up

Slices Click and Drag
Slice Up : ↑ , Wheel Up
Slice Down : ↓ , Wheel Down

Window Click and Drag
Windowing : Right Click

-3111.176460451491 HU
747.6948130277444

0 counts 26274

Main image

windowWidth 3859
windowCenter -1182
invert
window Custom
Lut Color default
index 217

Fusion

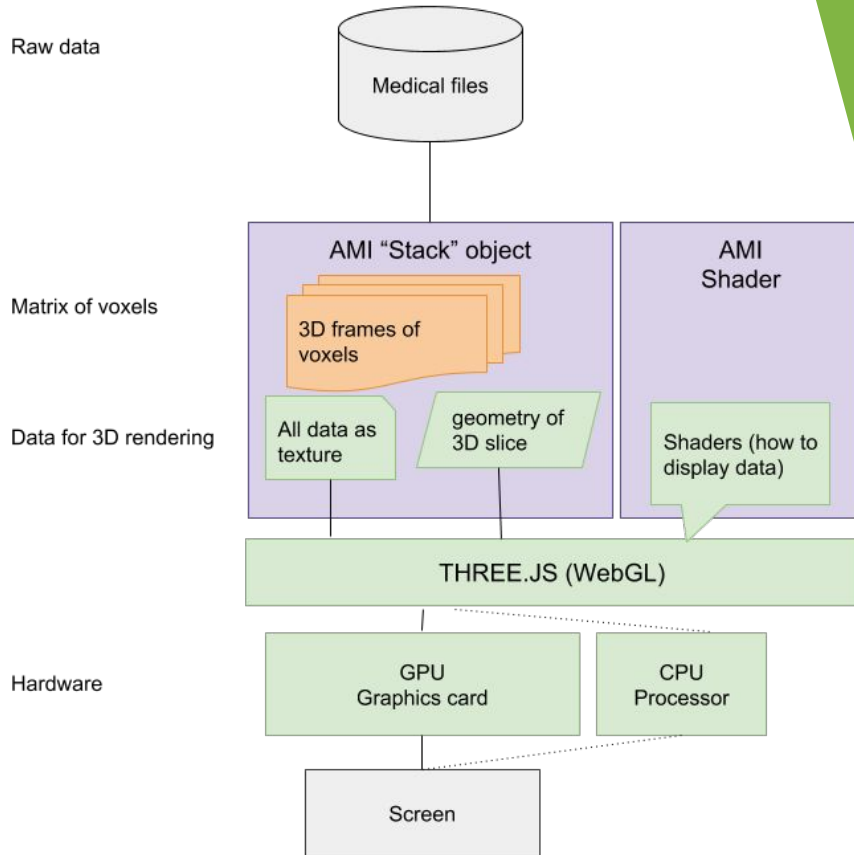
lut spectrum
Threshold 0.01
Opacity min 0.34
Opacity max 0.99

Camera

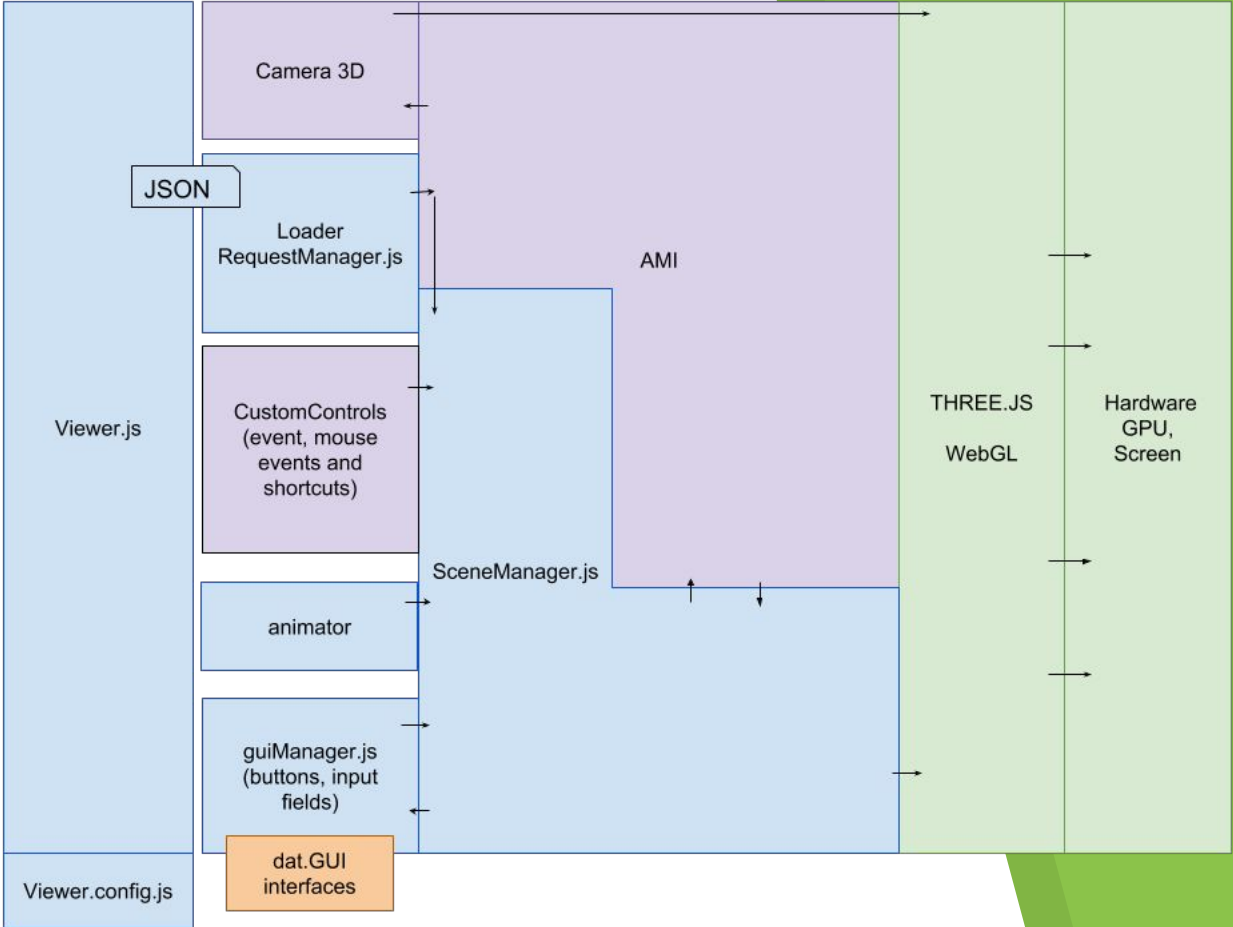
Close Controls

image : 115 HU
fusion : 944 counts

Tech - How AMI works ?



Tech - How modules communicate?



- Input/Output
- AMI lib
- YAMI code
- webGL related
- Other



Conclusion

- ▶ Setting up AMI for basic use is simple and fast
- ▶ Integrating it in an application must be thought carefully
- ▶ It's a client-side render ()
 - ▷ Less work for the server
 - ▷ Longer to load

(server-side calculation could still be done if needed)
- ▶ AMI is in Alpha : it evolves fast
 - ▷ The doc may not be always complete/updates
 - ▷ Some features you don't know exist
 - ▷ Sometime behavior you don't understand is just a bug.

AMI : ----- 174.000 lines of code

Simple example : ----- 100 lines of code

YAMI : ----- 2.500 lines of code



CREDITS

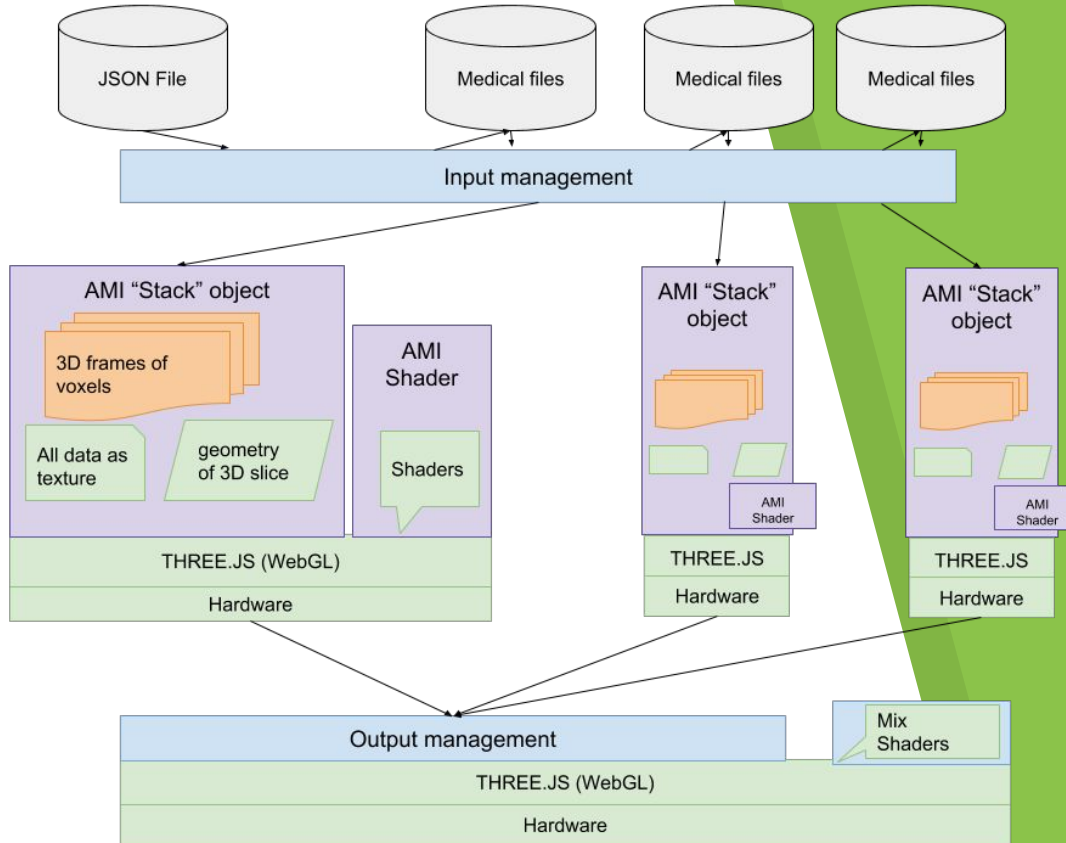
- ▶ WSY. David Sarrut, Thomas Baudier, Creatis
- ▶ DWV - <https://github.com/ivmartel/dwv>
- ▶ Papaya - <https://github.com/rii-mango/Papaya>
- ▶ OHIF - <http://ohif.org/>
- ▶ AMI - <https://github.com/FNNDSC/ami>
- ▶ VTK JS - <https://kitware.github.io/vtk-js/>

THANKS!

Any questions?

Tech - Details

Tech - How do I use AMI ?



Tech - JSON

```
{
  "study": "radioembolisation",
  "image": {
    "data": [
      [
        "data/Patient1_Radioembolisation/CT_2.mhd",
        "data/Patient1_Radioembolisation/CT_2.raw"
      ]
    ],
    "unit": "HU"
  },
  "fusion": {
    "data": [
      [
        "data/Patient1_Radioembolisation/NM_1.mhd",
        "data/Patient1_Radioembolisation/NM_1.raw"
      ]
    ],
    "unit": "counts"
  },
  "information": {
    "patient": "Patient1_Radioembolisation",
    "date": "2018-01-04 15:09",
    "injection": "Tc-99m 2018-01-04 15:09 226 MBq",
    "modality": "NM"
  }
}
```