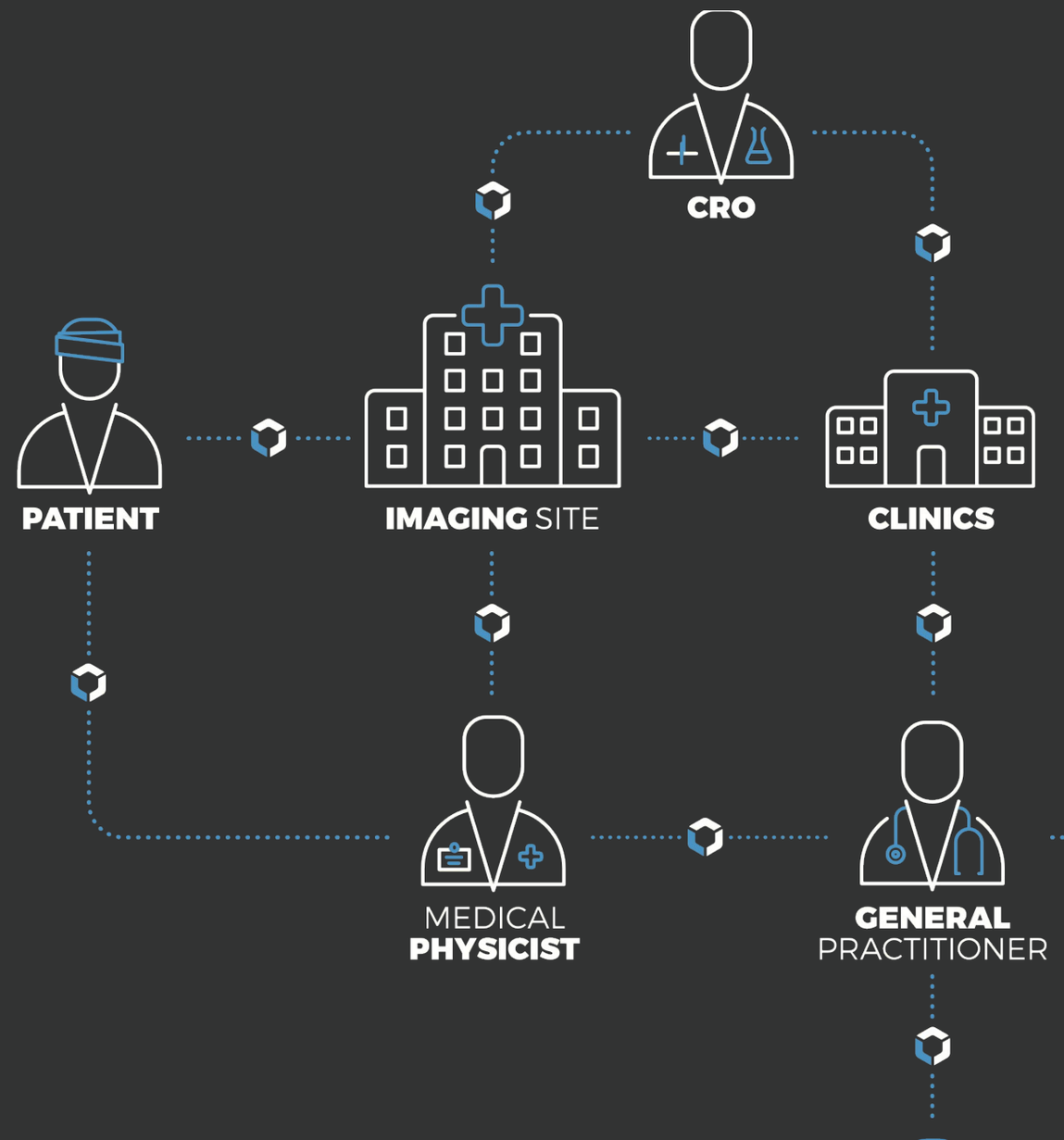
ORTHANC

The future of Orthanc

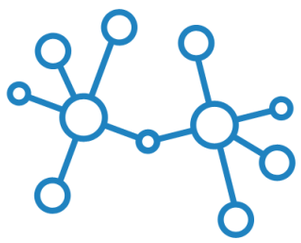


October 2013: Orthanc becomes a VNA

Picture archiving and communication system (PACS)

Vendor neutral archive (VNA)

Storage



Network

DICOM query/retrieve
to remote servers
is now supported!

Viewer

RIS

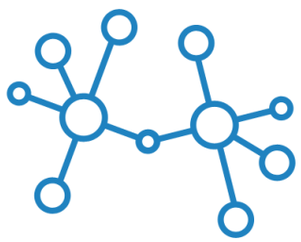
Transcoding

2020

Picture archiving and communication system (PACS)

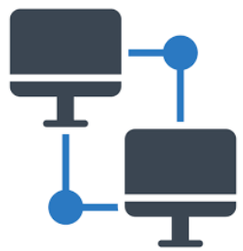
Vendor neutral archive (VNA)

Storage



Network

February?



Storage commitment

April?

Viewer

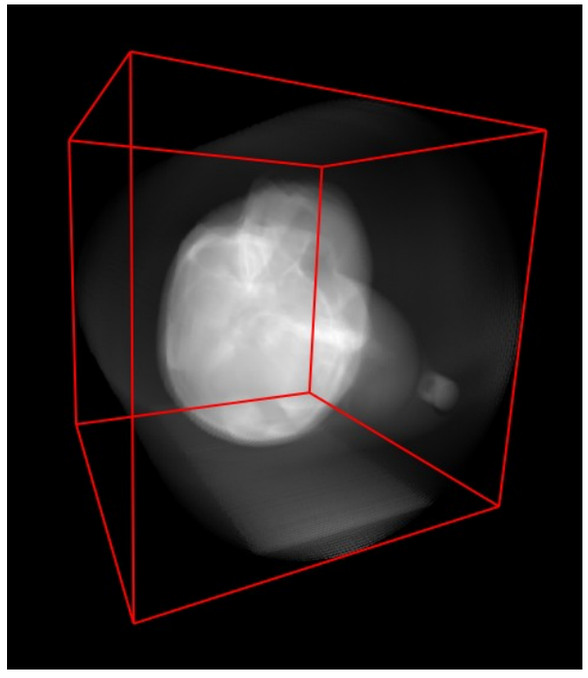
RIS

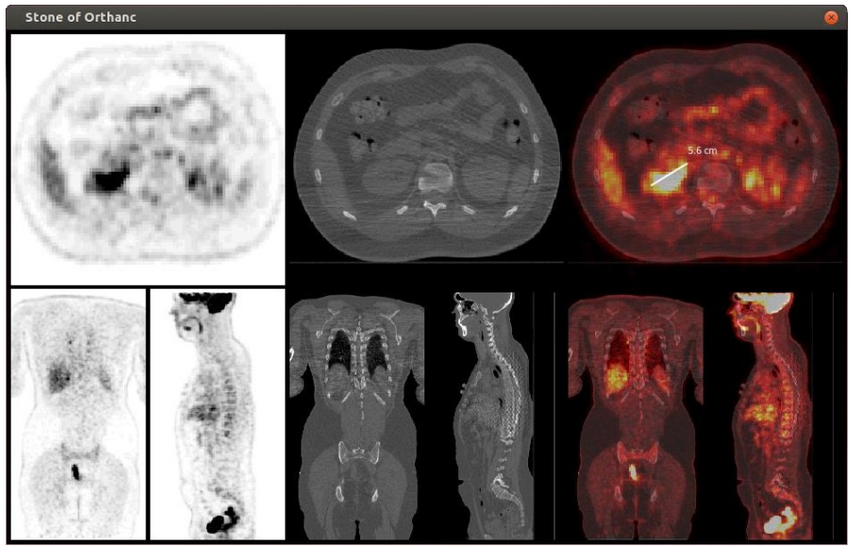
Transcoding

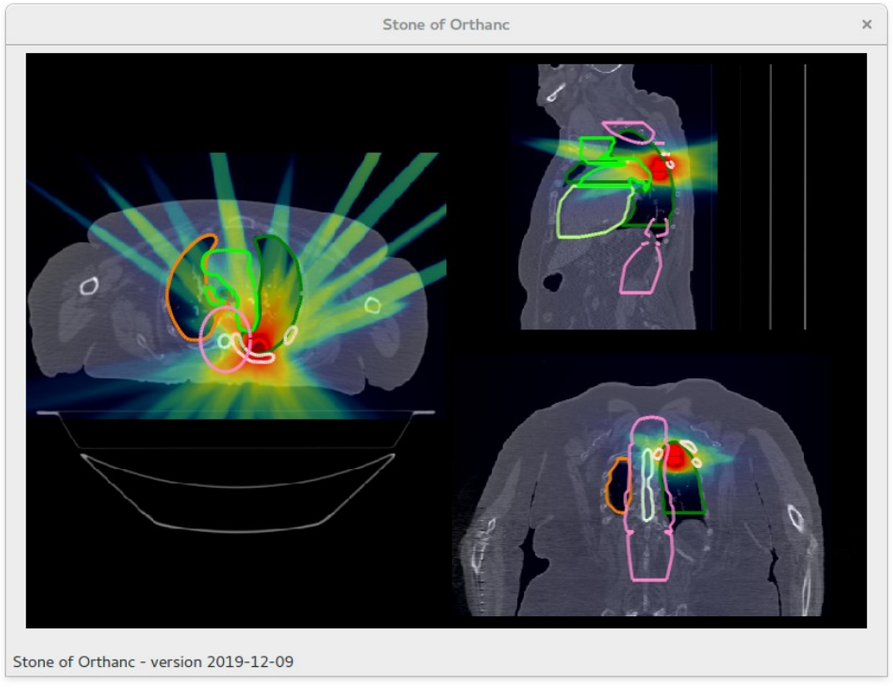
C-GET?
DICOM TLS?

2016 to 2019: Stone of Orthanc

- Standalone, companion project to Orthanc for **viewers**
- **Lightweight, cross-platform C++ library to render medical images** (cf. VTK)
- **Can be run by Web applications** (WebAssembly)
- 2D hardware acceleration (OpenGL/WebGL)
- Built-in support of 3D coordinates (MPR, volume reslicing)
- Support of oncology: PET-CT fusion, doses, contours…

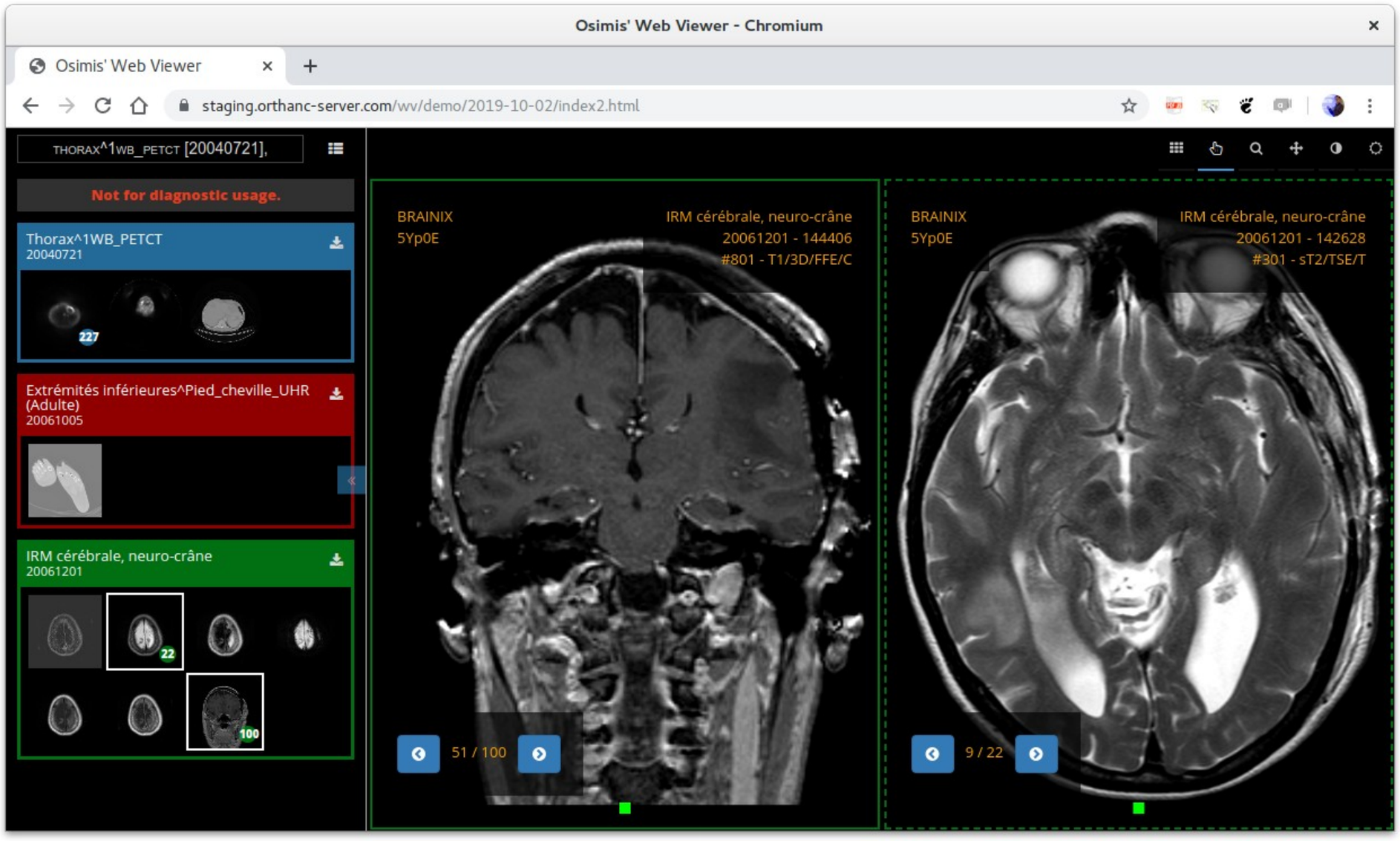






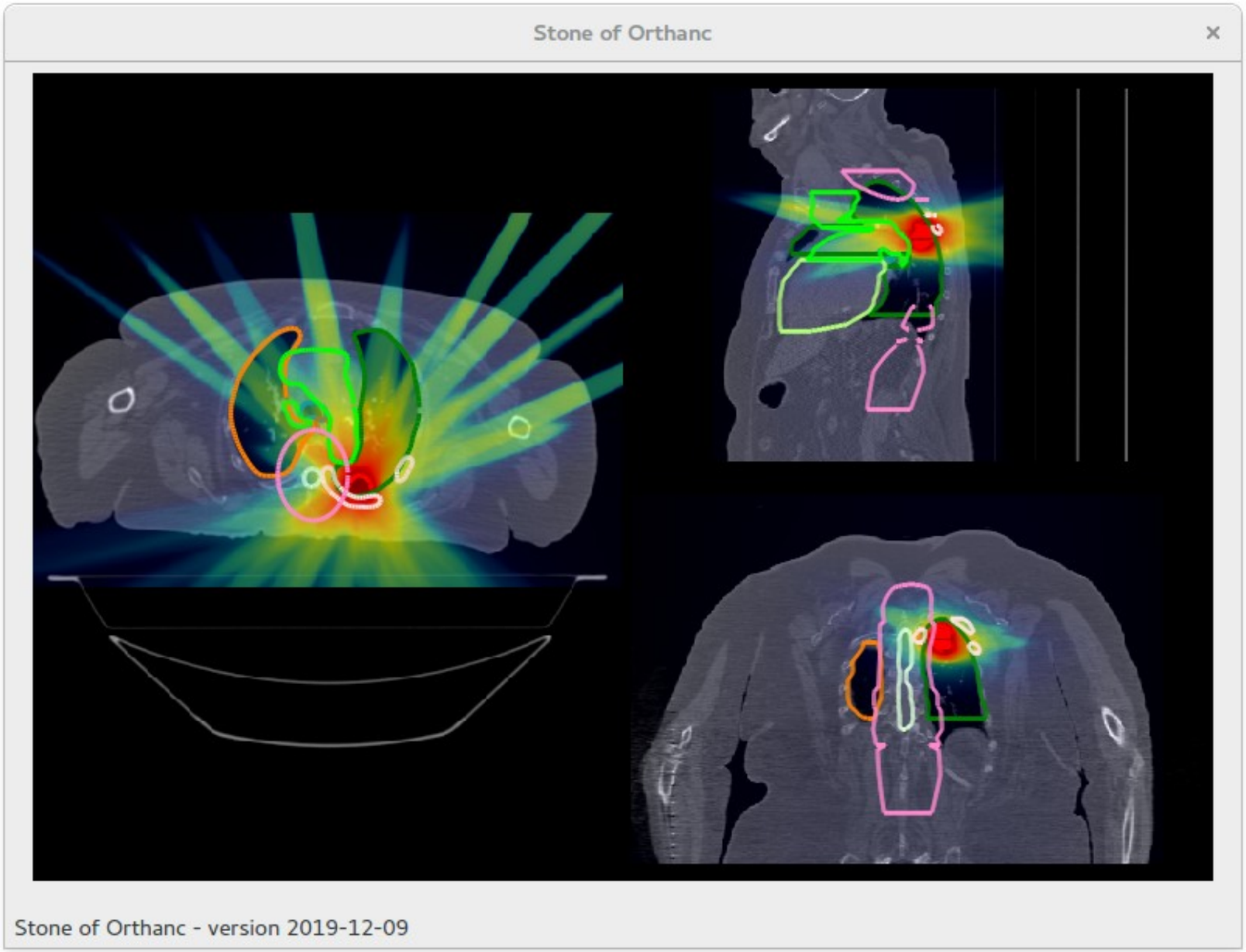
Ultimate goal: Fast development = build a new viewer in a handful of days

2020: Next generation of Web viewers

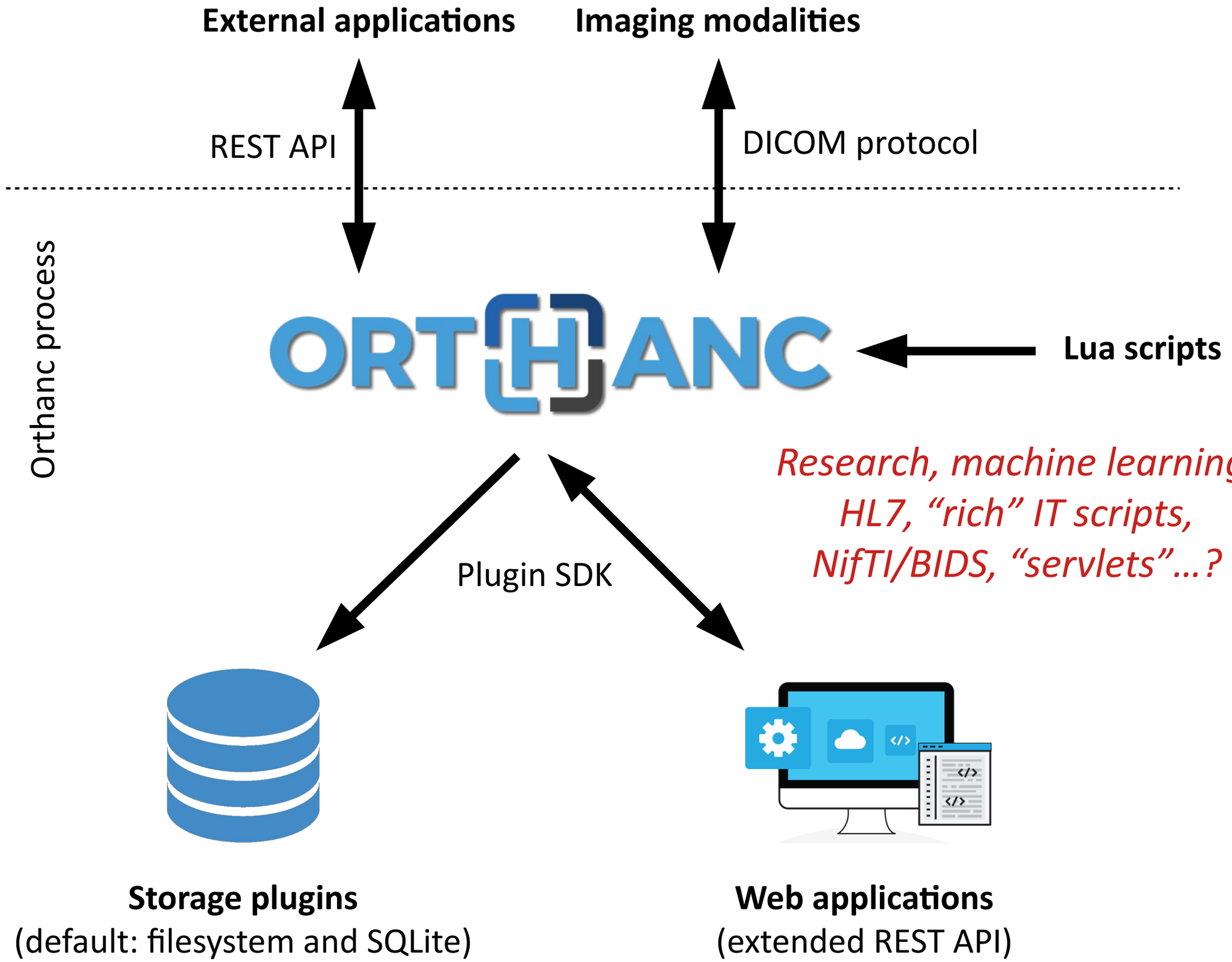


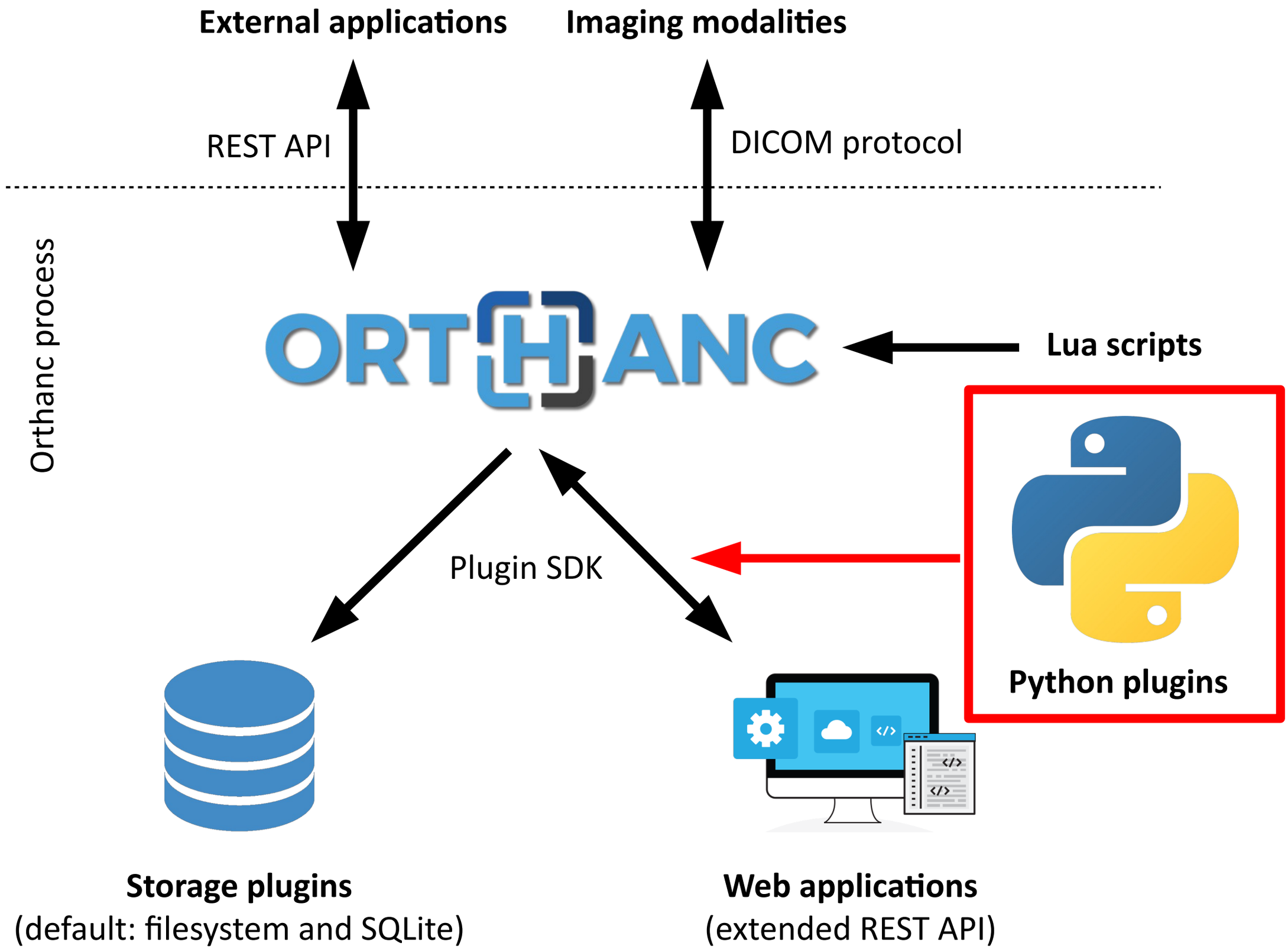
*This is **WebAssembly using Stone**!*
Full access to 3D geometry (MPR, reference lines…)
Independent of Orthanc: DICOMweb or any kind of “DICOM source”
Not decided yet: Heavyweight (Qt) or Web application?

2020: Radiotherapy viewers for end-users



Goal: Patient empowerment
(bring those highly specific images out of hospitals, clinical study)


External applications
Imaging modalities
REST API
DICOM protocol
Orthanc process
ORTHANC
Lua scripts
Research, machine learning,
HL7, “rich” IT scripts,
NifTI/BIDS, “servlets”…?
Plugin SDK
Storage plugins
(default: filesystem and SQLite)
Web applications
(extended REST API)


External applications
Imaging modalities
REST API
DICOM protocol
Orthanc process
ORTHANC
Lua scripts
Plugin SDK
Python plugins
Storage plugins
(default: filesystem and SQLite)
Web applications
(extended REST API)

Thanks for your contributions!