

RadioLogic

a case-based learning and self-assessment tool for the Orthanc
Ecosystem for Medical Imaging

Marco Barnig

1. Introduction

- A few years ago, my son in law, Guillaume Bierry, who is professor of radiology at the university hospital in Strasbourg, asked me if I could create a teaching tool for medical students, based on real clinical cases
- Reference : application about prostate cancer imaging developed by professor Philippe Puech from Lille
- Case-based learning is an efficient method for radiologist education
- Wanted : Tool including a real DICOM viewer to visualize whole medical imaging studies
- Challenge : OFUR (outil de formation universitaire radiologique)



2. My background

- 1974: research assistant at the Federal Institute of Technology in Zurich > first microprocessor Intel 4004
- 1978: POST Luxembourg > new telecommunication services : Alarmis, Euronet, Luxpac, Videotex, ISDN, LUXGSM, Internet
- 1993: responsible for sales and marketing of all telecommunications services
- 1998: first personal website
- 2013: retirement > time, experience, hardware, independence



3. OFUR

- Orthanc as PACS
- Cornerstone or DWV as HTML viewer
- Focus on iPad and Mac
- 3 targets : students, radiologists, MD
- 3 products : OFUR-chu, -lite, -pro
- Interface language : English
- License: open-source
- No medical certification
- Comply with data protection rules
- Project development : 5 steps
- Development duration : 12 months
- Project hosting : Amazon Web Services
- Personal assessment facilities
- Usage without Internet access
- Low costs

Mid 2015 : first prototype of OFUR-lite, based on the Cornerstone framework

4. Clinical cases

August 2015: tests with first clinical case
Issues:

- The download time was high
- Large CR files could not be handled on iPad's
- The cornerstone framework supported only ASCII characters
- CORS was required to make requests across domain boundaries

The DICOM files for the clinical cases and the integration of the pictures in DICOM files were assembled with Osirix

- MSK session with 5 clinical cases
- SPINE session with 3 clinical cases

Histoire	Patient de 40 ans Découverte fortuite sur un scanner abdo-pelvien Absence d'antécédents traumatiques
Clinique	Absence de fièvre Mobilisation indolore de la hanche gauche
Biologie	PNN : 8000 / mm ³ CRP : 2 mg / l PSA : < 4

Diagnostic ?	<ul style="list-style-type: none">• Métastase vertébrale• Lymphome• Dysplasie fibreuse• Chondrome• Fracture de fatigue
--------------	--



5. DICOM is easy

To solve the mentioned issues, it was necessary to dive deeper into the DICOM standard:

- Open-source toolkit DCMTK provided by OFFIS e.V. from Germany
- GDCM, an implementation of the DICOM standard developed by Mathieu Malaterre
- Dicom3tools package created by David A. Clunie
- DVTK editor and other DICOM tools

Roni Zaharia, CEO of H.R.Z. Software Services LTD, maintains a blog "DICOM is easy", including an outstanding tutorial about DICOM. For me DICOM was not easy, but a nightmare.

Despite the steep learning curve, I gradually got the necessary knowledge to manually process the DICOM files to reduce their size by compressing and scaling the images. I learned how to manage private tags to save information about the possible and correct diagnoses. I forked Chris Hafey's DicomParser to support UTF8 characters. To enable CORS I added a reversed proxy (nginx) to the Orthanc server.

6. Name & Logo

All the blocking issues were solved at the end of 2015:

- Decision to progress with the project
- Choice of a definitive name
- Design of a logo

Early 2016:

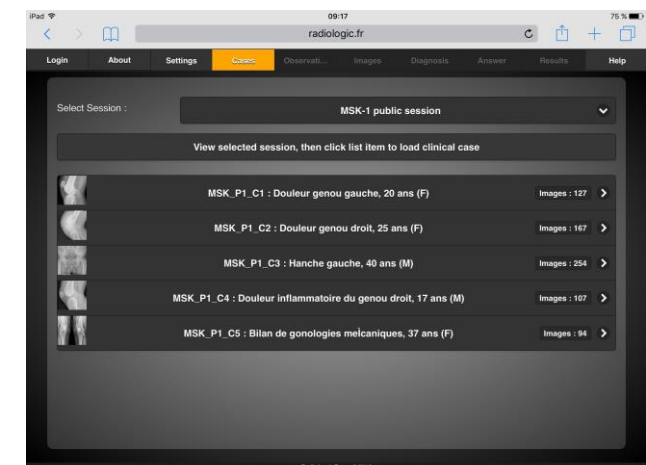
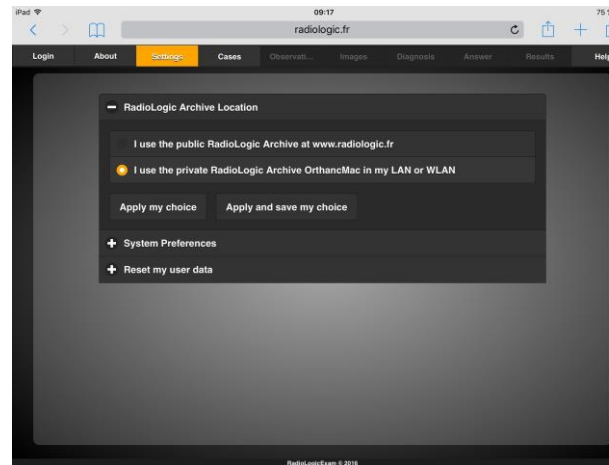
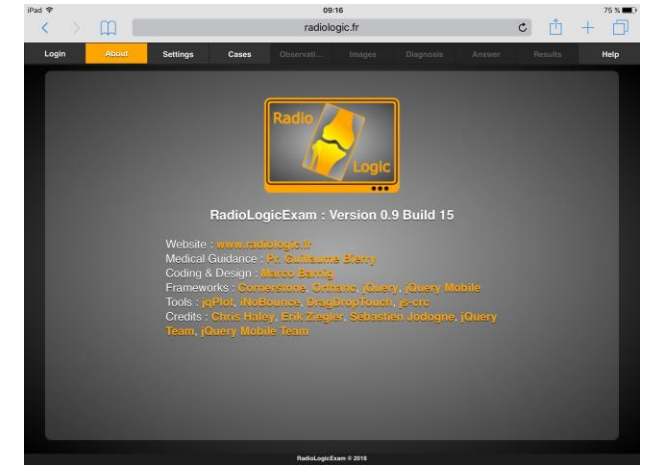
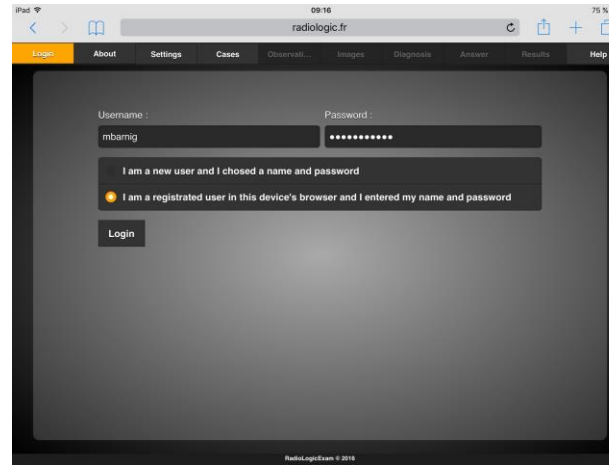
- Final project name > RadioLogic
- Logo > old-fashioned x-ray film viewer
- Domain name > radiologic.fr
- Website > hosting at AWS



7. RadioLogicTutor I

A stable version of the RadioLogic viewer, called RadioLogicTutor, was ready in mid-2016.

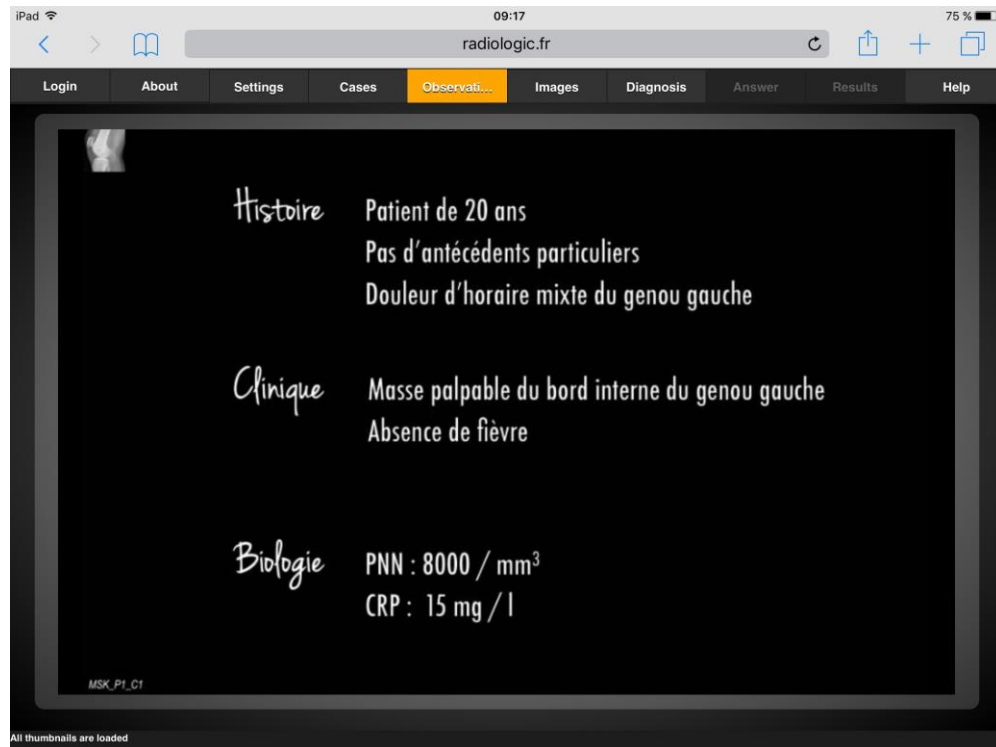
1. Login/Logout > ID and pw saved inside browser
2. About > credits
3. Settings > Orthanc location, system preferences
4. Cases > choice of session and clinical case



8. RadioLogicTutor II

5. Observation > clinical data about the patient with free text in a grey/color image

6. Images > four viewports, thumbnails-bar for DICOM series, drag and drop



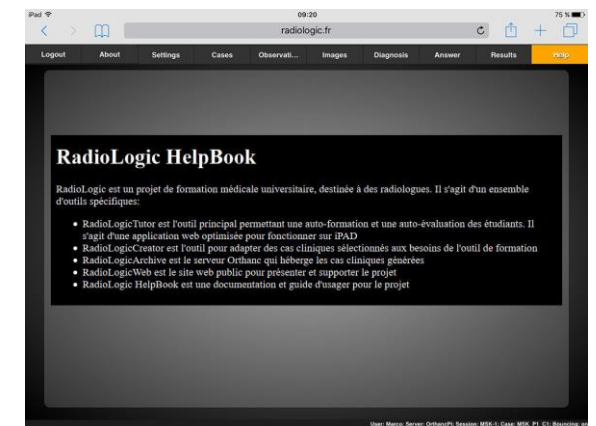
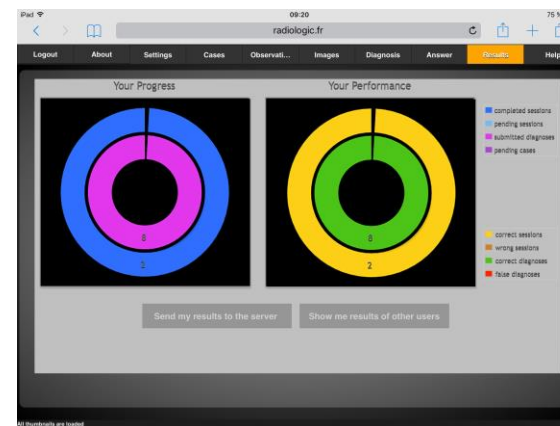
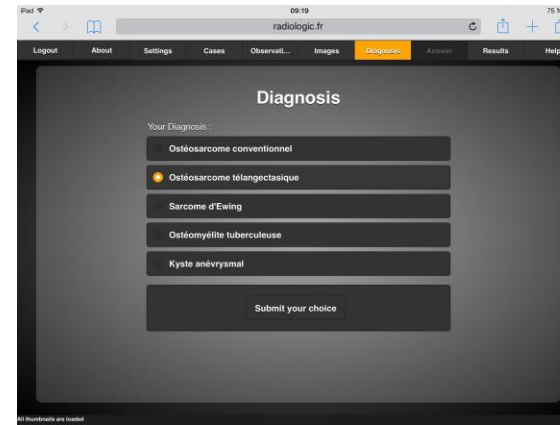
9. RadioLogicTutor III

7. Diagnosis > multiple choice radio-buttons, submission

8. Answer > grey or color image explaining the correct diagnoses with free text and pictures

9. Results > user progress and performance in graphical form (assessment), comparison with peers

10. Help > hints and info how to use the application



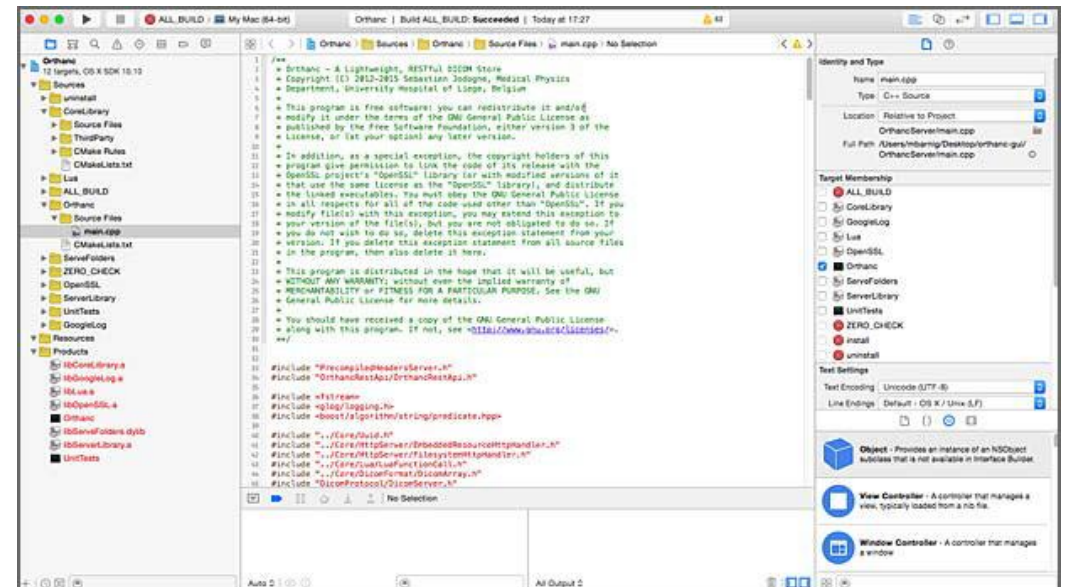
10. OrthancPi & OrthancMac

Early 2016, I ported the Orthanc source code to the Raspberry Pi, a credit-card-sized single-board computer

- Proof of concept
- Tests with models B Pi1, Pi2 and Pi3 and different WiFi adapters

In parallel, I compiled the Orthanc source code for OS X

- MacBook as DICOM archive
- OS X El Capitan



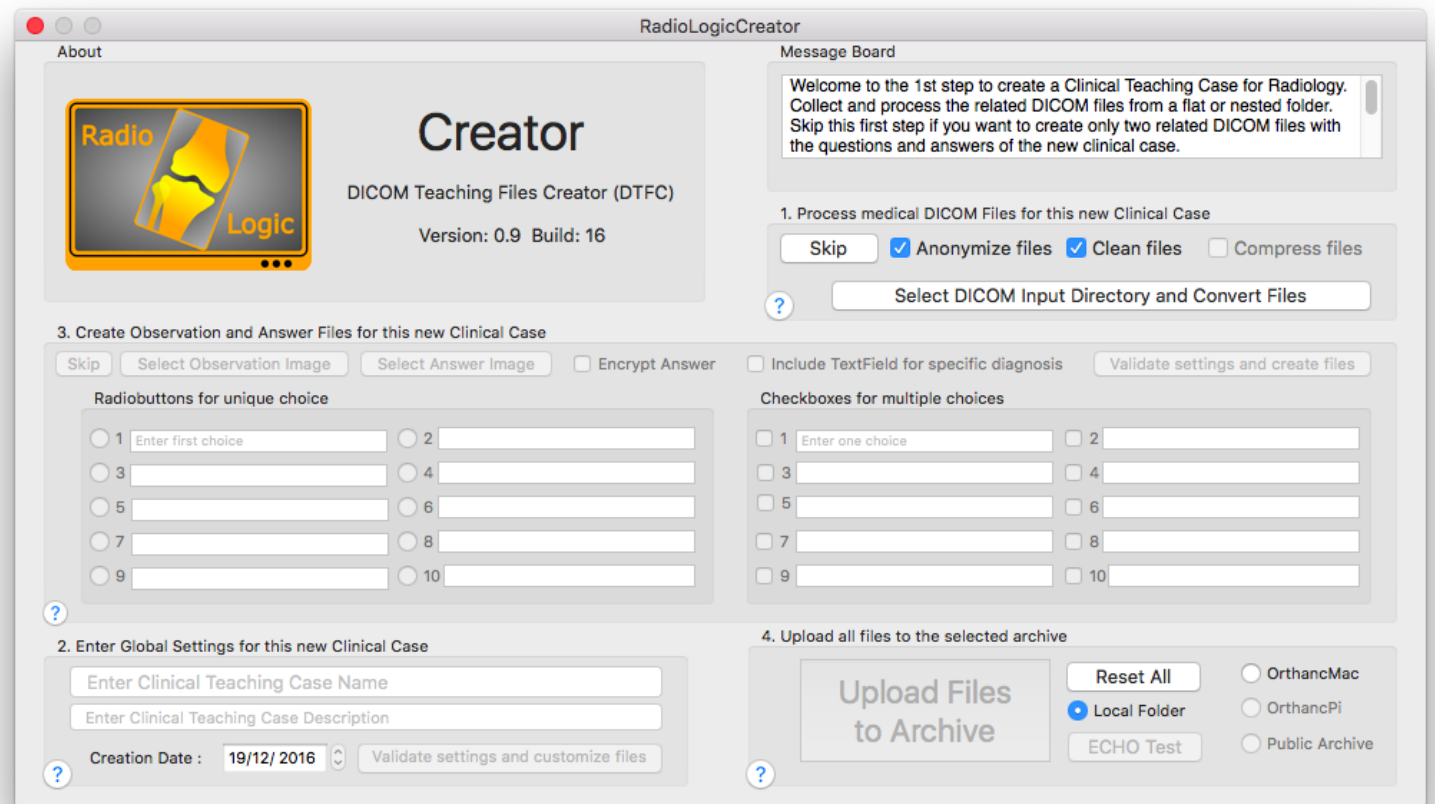
11. RadioLogicCreator on Mac I

Assembling, processing and checking selected DICOM files manually with a half-dozen different tools to create clinical cases was cumbersome and not error-proof

Mid 2016: tool to automate the process on Mac

- C, C++, Objective-C or Swift 3
- Experience with Java, PHP, Python, Javascript
- Four steps: compression & scaling, customization, private tags, uploading

End 2016 : 1st stable version



12. RadioLogicCreator on Mac II

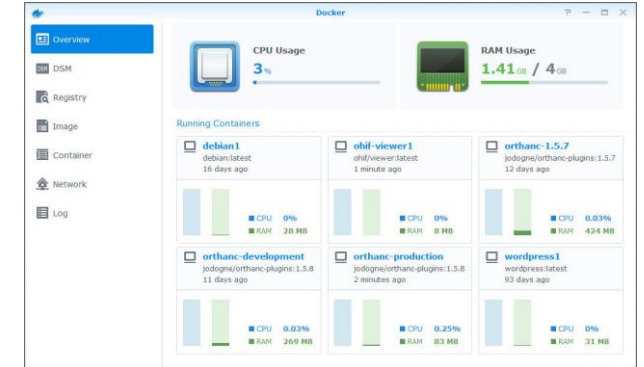
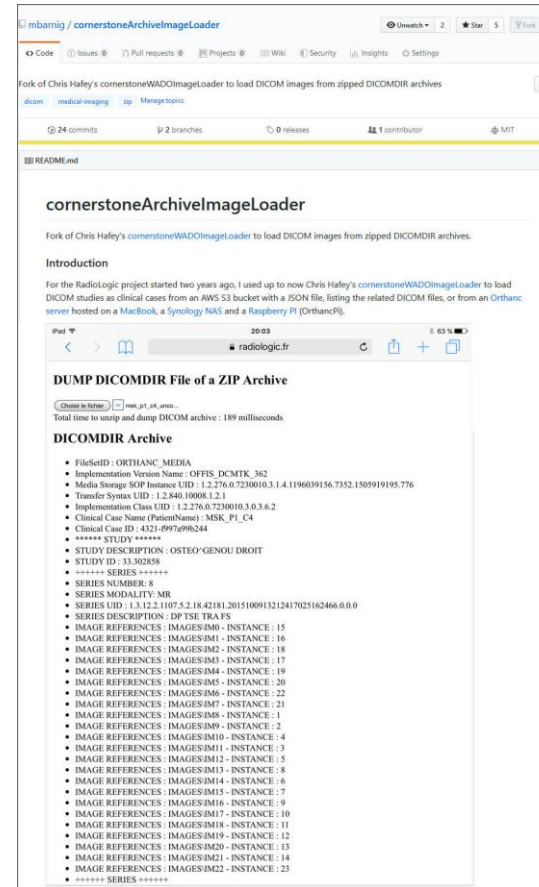
The RadioLogicCreator tool running on Mac was a continuous source of frustration for the users and the developer, for the following reasons:

- Deployment was difficult due to the severe security restrictions of Apple, despite the signature of the app by me as certified developer
- At each change of the OS X, it was necessary to adapt the application to new rules: El Capitan, Sierra, High-Sierra, Mojave, now Catalina
- The processing of DICOM files in four steps was synchronous and it took some waiting time before the creation of a clinical case was finished
- The main problem was the connection between the MAC computers running the RadioLogicCreator tool and the Orthanc server working as a DICOM archive with an integrated reversed nginx proxy in an ad-hoc local network

13. Improvements & Enhancements

In 2017, I looked for solutions to tackle the network issues:

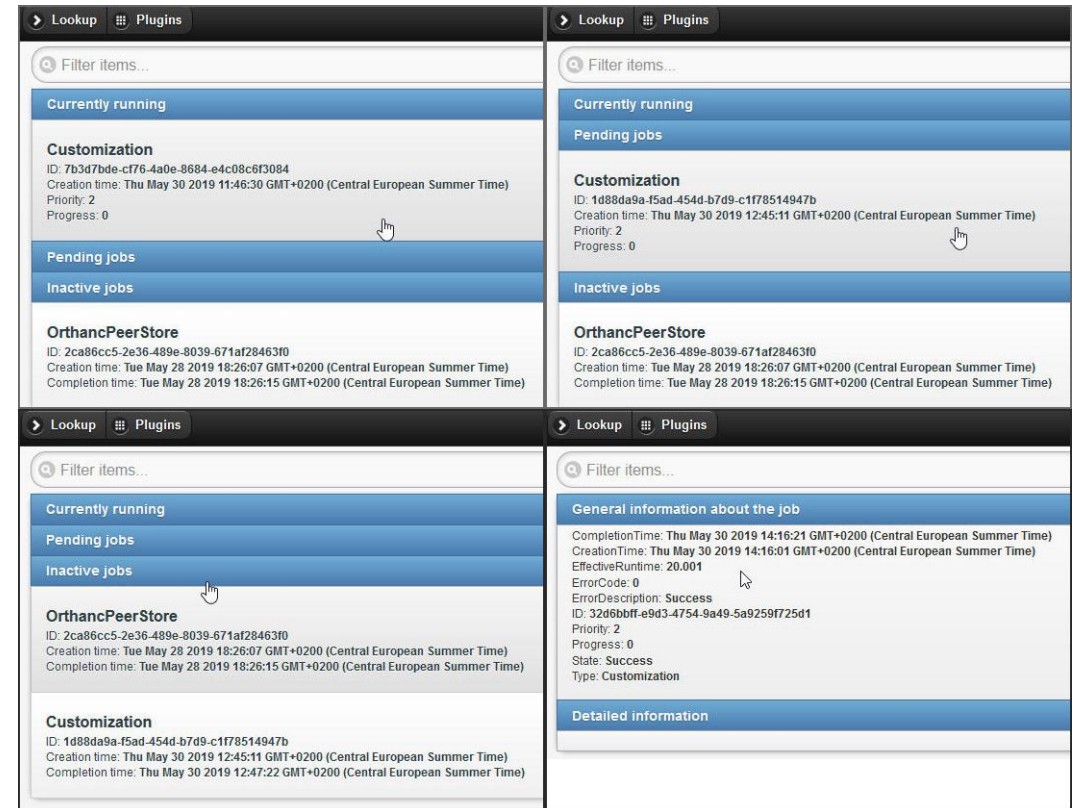
- CornerstoneArchiveImageLoader to store clinical cases in zipped archives inside the browser (new HTML5 technologies: indexedDB)
- Installation of Orthanc inside a Docker container
- Replacement of the MacBook, used from time to time as Orthanc server, by dedicated hardware: Synology diskstation



14. The last mile

After RadioLogicTutor, RadioLogicArchive was the second component of the RadioLogic system which was ready for a real-life deployment. The last component, RadioLogicCreator, was still waiting for improvement.

In July 2018, version 1.4 of Orthanc was released including a new advanced job engine and new metadata options. I was convinced that a plugin, using asynchronous jobs to create clinical cases, would solve all remaining problems, at least most of them.



15. RadioLogicCreatorPlugin

- HTML5 - Javascript webpage in the colors and style of the Orthanc explorer
- Similarity with the RadioLogicCreator OS X tool
- Creation of clinical cases with added Orthanc buttons
- Drop zone to upload external DICOM files recursively

The screenshot displays the RadioLogicCreator web application interface. At the top, a dark header bar contains the Orthanc logo and the application name. Below this, a navigation bar shows the current page. The main content area is divided into several sections:

- Top Right:** A dark box displays study information: StudyDate: Monday, February 25, 2019; StudyID: CHEM10752929; StudyInstanceUID: 1.3.51.0.1.1.10.10.10.150.10752929.3967065. Below this is a yellow button labeled "Create clinical case from this study".
- Interact Section:** A grey box with a close button contains a button labeled "Delete this study".
- Main Form:**
 - Author:** Pr Guillaume Bierry
 - Case Type:** Study
 - Welcome message:** Welcome to RadioLogicCreator. 'Case Type' and 'DICOM UID' are readonly.
 - Clinical Case:** MSK_P1_C1
 - DICOM UID:** 1.3.51.0.1.1.10.10.10.150.10752929.3967065
 - Description:** Douleur genou gauche, 20 ans (F)
 - Buttons:** "Select stored DICOM files" (blue) and "Upload new DICOM files" (grey).
 - Diagnosis Fields:** A grid of text input fields for Diagnosis 0 through 9. Diagnosis 0 contains "Métastase vertébrale", Diagnosis 1 contains "Ostéosarcome", and Diagnosis 2 contains "Sarcome d'Ewing".
 - Image Selection:** Two blue buttons labeled "Choose Observation Image File" and "Choose Answer Image File". Below each is a "Browse..." button and a "No file selected." message.
 - Correct Diagnosis:** A dropdown menu showing "1".
 - Scramble Key:** A text input field containing "1873".
 - Clinical Case Date:** A date picker showing "13/11/2019".
 - Buttons:** "Validate" (blue) and "Submit" (yellow).
 - DICOM Upload:** A dashed box labeled "Drop DICOM Folder here" and a button labeled "Choose DICOM Files".
 - DICOM Directory Tree:** A dashed box labeled "DICOM Directory Tree:".

The footer of the application displays "RadioLogicCreator © 2019".

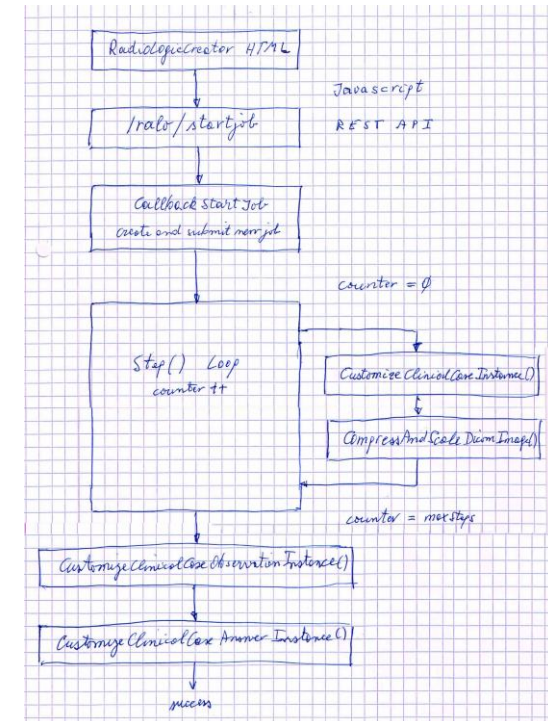
16. Plugin Functions I

The program code of the plugin is a mix of the following elements:

- Orthanc SDK functions (from the modules Toolbox, Orthanc, Callback and REST)
- Members of the Orthanc C++ wrapper class
- Specific C++ functions

Main functions:

- CallbackStartJob()
- RadioLogicCreator::Step()
- CustomizeClinicalCaseInstances()
- CompressAndScaleDicomImage()
- CustomizeClinicalCaseObservationInstance()
- CustomizeClinicalCaseAnswerInstance()



17. Plugin Functions II

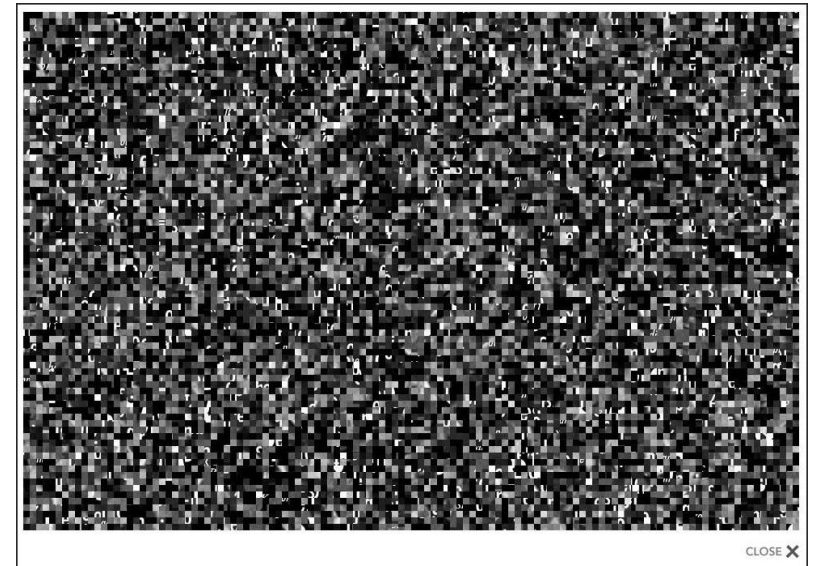
The Step() function works in the following three states by checking and incrementing the value of the member variable counter:

- `counter == 0` : the member variable `maxSteps` is set to the number of instances to process. The anonymization and customization of the instances is prepared and the function `CustomizeClinicalCaseInstances()` is called a first time. The job information content is initialized. The counter is incremented and the `Step()` function returns with "continue".
- `0 < counter < maxSteps` : `CustomizeClinicalCaseInstances()` and `CompressAndScaleDicomImage()` are called at each step by passing the related `instanceld`. The job content is regularly updated. If the `CustomizeClinicalCaseInstances()` returns an error, the `Step()` function returns with "stop" (failure), otherwise with "continue".
- `counter == maxSteps` : `CustomizeClinicalCaseInstances()` and `CompressAndScaleDicomImage()` are called a last time, followed by calling `CustomizeClinicalCaseObservationInstance()` and `CustomizeClinicalCaseAnswerInstance()`. If everything works fine, the `Step()` function returns with "success".

18. Plugin Functions III

`CustomizeClinicalCaseObservationInstance()` and `CustomizeClinicalCaseAnswerInstance()` add the possible diagnoses and other data as private tags into the stored DICOM files. Due to issue 140 with Orthanc core, it is not possible to insert or modify private tags inside the Orthanc REST API's `"/create-dicom"` and `"/modify"`. As a workaround I use a DICOM template with empty private tags, replace the pixels in the template with the data of the observation and answer image-files and modify these templates with a system call to the Offis DCMTK tool `"dcmmodify"`.

The answer picture and the correct diagnosis are scrambled with javascript before the upload to the server to prevent clever users looking at the results before submitting their own diagnosis in RadioLogicTutor. The whole scrambling process is not hacker proof, but I think that encrypting the answers with public-key cryptography would be design-overkill.

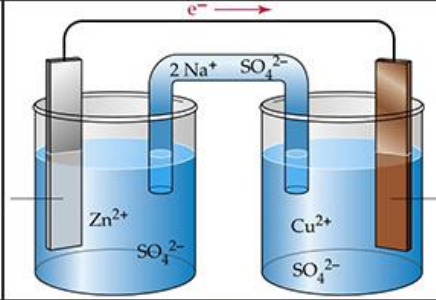


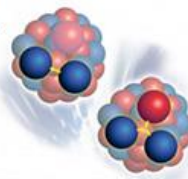


19. Learning with quizzes

The Orthanc ecosystem and the presented RadioLogic tools are not limited to store, to view and to assess knowledge to interpret medical images.

Technically we can use pictures presenting any real-world problem with a related question and use DICOM tags, or Orthanc metadata, to store the correct answer.

A simple example is an e-learning quizz. Such quizzes are widespread on the Internet and related to all sort of topics : IQ tests, STEM (science, technology, engineering, mathematics) education, general knowledge, trivia or even fun. The answer types of these quizzes could be multiple-choice, true-false or open-ended.

$\begin{aligned} \text{Mario} + \text{Mario} + \text{Mario} &= 60 \\ \text{Mario} + \text{Peach} + \text{Peach} &= 30 \\ \text{Peach} + \text{Mickey} + \text{Mickey} &= 9 \\ \text{Mario} + \text{Peach} \times \text{Mickey} &= ? \end{aligned}$					
<p>Can you name this flower?</p> 	$\begin{aligned} \text{Boot} + \text{Boot} + \text{Boot} &= 30 \\ \text{Boot} + \text{Snowflake} + \text{Snowflake} &= 20 \\ \text{Snowflake} + \text{Ice} + \text{Ice} &= 9 \\ \text{Snowflake} + \text{Ice} \times \text{Boot} &= ? \end{aligned}$				
$\begin{aligned} \text{Apple} + \text{Apple} + \text{Apple} &= 30 \\ \text{Apple} + \text{Banana} + \text{Banana} &= 18 \\ \text{Banana} - \text{Coconut} &= 2 \\ \text{Coconut} + \text{Apple} + \text{Banana} &= ?? \end{aligned}$	<p>Which particle is the mediator of the strong force?</p>   <p>1</p> <table border="1"><tbody><tr><td>A. Neutralino</td><td>B. Z boson</td></tr><tr><td>C. Gluon</td><td>D. Quark</td></tr></tbody></table>	A. Neutralino	B. Z boson	C. Gluon	D. Quark
A. Neutralino	B. Z boson				
C. Gluon	D. Quark				

20. Conclusions

I close my presentation with a few personal conclusions:

- A project which evolved from a minimum viable project in 2012 to an outstanding open-source ecosystem and gave birth to a university spin-off, which occupies now more than 12 qualified people, is a great success story. Congratulations to Sébastien Jodogne for creating Orthanc and co-founding Osimis.
- A large project related to advanced technologies takes three times more resources and time than the initial estimation, due to the appearance of non-expected issues.
- The interest in new technologies can turn into addiction.

Thank you for listening.