

1 **Implementation and prospective real-time evaluation of a**
2 **generalized system for in-clinic deployment and validation**
3 **of machine learning models in radiology**
4

5 James R Hawkins^{1*}, Marram P. Olson¹, Ahmed Harouni², Ming Melvin Qin²,
6 Christopher P Hess¹, Sharmila Majumdar¹, Jason C Crane¹

7

8 ¹Center for Intelligent Imaging, Department of Radiology and Biomedical Imaging,
9 University of California, San Francisco, CA, USA

10 ²NVIDIA, Santa Clara, CA, USA

11

12 *Corresponding Author

13 Email: James.Hawkins@ucsf.edu

14 **Abstract**

15 The medical imaging community has embraced Machine Learning (ML) as
16 evidenced by the rapid increase in the number of ML models being developed, but
17 validating and deploying these models in the clinic remains a challenge. The
18 engineering involved in integrating and assessing the efficacy of ML models within the
19 clinical workflow is complex. This paper presents a general-purpose, end-to-end,
20 clinically integrated ML model deployment and validation system implemented at
21 UCSF. Engineering and usability challenges and results from 3 use cases are
22 presented.

23 A generalized validation system based on free, open-source software was
24 implemented, connecting clinical imaging modalities, the Picture Archiving and
25 Communication System (PACS), and an ML inference server. ML pipelines were
26 implemented in NVIDIA's Clara Deploy framework with results and clinician feedback
27 stored in a customized XNAT instance, linked within PACS. Prospective clinical
28 validation studies of 3 ML models were conducted, with data routed from multiple
29 clinical imaging modalities and PACS. Completed validation studies provided expert
30 clinical feedback on model performance and usability, plus system reliability and
31 performance metrics.

32 Clinical validation of ML models entails assessing model performance, impact on
33 clinical infrastructure, robustness, and usability. Study results must be easily
34 accessible to participating clinicians but remain outside the clinical record. Building a
35 system that generalizes and scales across multiple ML models takes the concerted

36 effort of software engineers, clinicians, data scientists, and system administrators, and
37 benefits from the use of modular open-source software. The present work provides a
38 template for institutions looking to translate and clinically validate ML models in the
39 clinic, together with required resources and expected challenges.

40 **Author summary**

41 Academic medical centers gather and store vast quantities of digital data, and with
42 the increase in accessibility of Machine Learning (ML) techniques, there has been an
43 explosion of ML model development in the medical imaging community. Most of this
44 work remains in research, though, and connecting ML models to the clinic for testing
45 on live patient data and integration into the clinical workflow remains a challenge and
46 impedes clinical impact. We present a general-purpose system, implemented and
47 deployed at UCSF, for in-clinic validation of ML models and their incorporation into
48 patient care. This work, based on free and open-source software packages, can serve
49 as a template for other institutions looking to solve ML’s “last mile” problem and move
50 their models out of research and into the clinic.

51 **Introduction**

52 The medical imaging community is embracing Machine Learning (ML) and Artificial
53 Intelligence (AI) to develop novel predictive models. These models show promise, and
54 have the potential to transform radiology practice and patient care, in areas ranging
55 from data acquisition, reconstruction, and quantification, to diagnosis, treatment

56 response, and clinical workflow efficiency.[1] While the foundation of this work is model
57 development using retrospectively acquired datasets[2], translating AI models from
58 research to the clinic for event-driven, prospective validation is a critical step towards
59 model deployment for routine use in clinical care. Prospective model validation within
60 the clinical workflow not only provides an opportunity to capture expert clinical
61 feedback about a model’s performance, but is also critical for assessing usability,
62 interpretability, and effectiveness of results, and technical issues related to integration
63 with clinical information systems. Moving ML models from “proof-of-concept” to
64 “production” is the critical next-step for medical imaging.[3]

65 The infrastructure and software systems required to clinically integrate models for
66 validation are extensive and can pose major hurdles to ultimately realizing the clinical
67 impact of AI in medicine.[4] Solutions range from commercial products to custom in-
68 house applications[5–8], and offer pros and cons for flexibility, engineering effort, cost,
69 interoperability with clinical systems, support, and usability. The contribution of this
70 paper is a description of a general-purpose end-to-end ML model validation and
71 deployment framework, based on NVIDIA’s Clara Medical Imaging[9] software package
72 and the XNAT[10] imaging study management application, that has been developed
73 and deployed at UCSF within The Center for Intelligent Imaging[11] (ci²). The present
74 implementation is built on free, open-source software packages and represents a
75 template for other sites wishing to translate and deploy clinically integrated ML models.
76 The overall system architecture is presented together with the benefits and challenges
77 associated with this design, based on experience implementing 3 separate model

78 validation studies, each representing different but commonly occurring clinical use
79 cases in radiology AI.

80 **Materials and methods**

81 **Data flow**

82 Fig 1 details the end-to-end AI inference system and networks presented in this
83 work. Briefly, DICOM[12] images are sent from scanning modalities at time of
84 acquisition to a DICOM router. The router directs images to the Picture Archiving and
85 Communication System (PACS)[13] and to specific inference services hosted on a
86 server running NVIDIA Clara Deploy[14]. Results are exported to an XNAT instance
87 running on the same host. Clinicians access the results from a PACS workstation or
88 other UCSF computer by logging into the XNAT web application. Custom buttons in
89 the Visage[15] client running on PACS workstations link directly to relevant results in
90 XNAT, where reviewer feedback is captured for use in assessing model performance or
91 for retraining (Fig 2).

92

93 **Fig 1: High-level system architecture and data-flow diagram of clinically integrated**
94 **inference validation service.**

95 **Fig 2: Workflow for accessing validation studies from the Visage clinical PACS viewer.**

96 From top left, clockwise: 1) Custom buttons in Visage link current study to ML results
97 and/or validation study assessment in XNAT. 2) AI Hip Fracture Detection results and

98 assessment, accessed via Visage link. 3) AI segmentation of the liver (left and right
99 lobes and vessels) for the Liver Transplant Segmentation validation study, displayed in
100 XNAT's OHIF viewer, via Visage link.

101

102 Image Routing and Ingestion

103 All DICOM images are sent from clinical scanning modalities to a DICOM router
104 (Compass[16] from Laurel Bridge) that is configured with rules for directing data to
105 various endpoints, including the clinical PACS and the Clara Deploy inference service.
106 Compass' routing rules are a set of user-defined mappings based on DICOM tags in
107 the data. Three rules, corresponding to three proof-of-concept applications, route
108 images to AI inference pipelines (Table 1). Additionally, the inference pipelines are set
109 up as export destinations in the clinical PACS, allowing clinicians to manually transmit
110 images to specific pipelines on-demand. All images are transmitted via DICOM
111 communication protocols.

112 **Table 1: Compass routing rules for the 3 AI inference pipelines described in this work.**

Clara Pipeline Destination	Called AE Title	DICOM Tag Name	DICOM Tag Group/Element	DICOM Tag Value
Brain Tumor Segmentation	CI2_CD_BTS	SOP Class UID	(0008,0016)	MR Image Storage
		Study Description	(0008,1030)	"BRAIN*NAV"

		Station Name	(0008,1010)	A set of pilot scanners
Liver Transplant Segmentation	CI2_CD_LDN	SOP Class UID	(0008,0016)	CT Image Storage
		Study Description	(0008,1030)	“ABDOMEN FOR LIVER DONOR WITH CONTRAST”
		Station Name	(0008,1010)	A set of pilot scanners
Hip Fracture Detection	CI2_CD_HIP_FRAC	SOP Class UID	(0008,0016)	CR Image Storage
		Study Description	(0008,1030)	“PELVIS”
		Station Name	(0008,1010)	A set of pilot scanners

113

114 AI inference servers

115 The system is comprised of both production and development inference servers
 116 (Fig 3). These are virtual machines (VMs), running on top of VMware’s vSphere[17]
 117 server virtualization software. The servers run Ubuntu 18.04[18], and each is assigned a
 118 dedicated NVIDIA T4–16c GPU, using NVIDIA Virtual GPU Software’s GPU Pass-
 119 Through mode[19]. Table 2 summarizes the server infrastructure.

120 Fig 3: Inference VM internal architecture diagram.

121 Table 2: Inference infrastructure configuration

Physical Server	Cisco UCS C240-M5s
-----------------	--------------------

	2 x NVIDIA T4 GPUs
	NVIDIA GRID software version 11.1 (installed on ESXi host)
Virtual Machines	8 vCPUs (Intel Xeon Platinum 8168 CPU @ 2.70GHz)
	32G RAM (reserved)
	PCI Device: NVIDIA GRID vGPU
	GPU Profile: grid_t4-16c

122

123 **AI inference framework**

124 The AI inference pipelines running on these VM's are controlled by and developed
125 in NVIDIA's Clara Medical Imaging application framework. Each VM runs Clara Deploy,
126 a container-based framework for deploying AI workflows. The framework allows
127 developers to build machine learning pipelines that run inference on NVIDIA GPUs, and
128 it supports end-to-end services that include: DICOM import/export, user-extensible
129 pipeline and GPU management, running multiple AI models on GPUs, and interactive
130 image rendering.

131 Pipelines and services are run in Docker[20] containers and deployed using
132 Kubernetes[21]. AI model inference is run on GPUs using NVIDIA's Triton Inference
133 Server[22]. Pipelines are registered to specific DICOM AE Titles, and when the Clara
134 DICOM Adapter receives a set of images, it looks at the called AE Title and starts
135 processing the images with the associated pipeline.

136 **Delivery of results**

137 Inference pipelines export imaging results (e.g., spatial segmentations) as well as
138 scalar classifications and derived metrics to an instance of XNAT, an imaging
139 informatics and study management platform. XNAT was chosen as the mechanism to
140 store and display inference results because of its a) ability to store both imaging and
141 derived scalar data together in one application, b) extensibility, which allows
142 developers to define custom schemas and functionality via its plugin architecture, c)
143 built-in DICOM support, d) security and user permissions model, e) REST API[23], f)
144 support for the OHIF image viewer[24], and g) customizable web-based user interface,
145 which can be tailored to meet the data visualization, feedback capture, and workflow
146 requirements of each inference pipeline.

147 The XNAT application and its Postgres[25] database are each run inside Docker
148 containers, and HTTPS communication is proxied through an NGINX[26] container.
149 XNAT user accounts and authentication are integrated with UCSF's Active Directory
150 Service[27], via the LDAP[28] protocol.

151 **Inference pipelines**

152 AI inference pipelines need to perform a consistent set of tasks. The first step often
153 involves parsing an imaging exam to find the relevant input series. Next, images
154 typically require preprocessing, such as intensity normalization, cropping, resampling,
155 and/or registration. Following inference, additional post-processing operations may be
156 required, for example computing derived metrics such as segmentation volumes.
157 Finally, results must be exported to a data management system such as XNAT or a
158 PACS. In Clara Deploy, each of these tasks are implemented as independent software

159 units, called “operators”. Pipelines are composed of series of chained operators, each
160 running as a Docker container. Each operator receives data from its preceding
161 operator, via shared data mounts, and performs one processing task before passing
162 output to the next operator. This architecture allows for the reuse of general-purpose
163 operators and extensibility of other algorithm modules for new pipelines. Fig 4
164 illustrates a typical image segmentation pipeline archetype.

165

166 **Fig 4: Anatomy of an ML inference pipeline.** Each box represents a pipeline operator
167 executing a Docker container, managed by Clara Deploy.

168

169 **NVIDIA Clara Deploy operators**

170 NVIDIA provides a library[29] of Clara Deploy operators as Docker images that can
171 be used to compose pipelines, including operators for DICOM reading/writing, exam
172 parsing, series selection, and deployment of Clara Train developed AI models. NVIDIA
173 also provides a base Docker image which can be used to develop custom operators
174 for additional functionality and integration of models developed outside of the Clara
175 Train framework. The pipelines in this work use a mixture of NVIDIA’s standard Clara
176 Deploy operators, modified operators that extend standard Clara Deploy operators,
177 and fully in-house developed operators (Table 3).

178 **Table 3: List of Clara Deploy pipeline operators used, with estimate of software**
179 **engineering time necessary to develop similar functionality.**

Operator Name	Purpose	Type	Effort
Clara DICOM Reader	Ingests a DICOM series and converts to MHD[30]/NIfTI[31]	Included	
Clara Deploy DICOM Parser	Ingests a DICOM exam and parses DICOM metadata for use with the Series Selector	Included	
Clara Deploy Base Inference V2	Performs AI model inference with Clara Train developed models	Included	
Clara Register Results	Used to export results to DICOM receivers	Included	
Clara DICOM Writer	Writes AI model results into DICOM format; modified to allow for custom DICOM tag values	Included	
Clara Deploy DICOM Parser	Modified to add in additional DICOM tags to the parsed metadata, to parse/convert x-ray exams with multiple instances per series, and to continue converting series in an exam if there is a failure converting a series with unexpected DICOM attributes	Modified	~1 day
Clara Deploy Series Selector	Modified to add in regular expression parsing, to move selected series to output directories, and provide the option to select individual x-rays on an instance (non-series) level	Modified	~1 day
Clara Deploy DICOM Segmentation Writer	Developed in parallel with NVIDIA, and modified with the ability to customize DICOM tag values	Modified	~1 week
DICOM RTSTRUCT Writer	Modified to handle image ordering and orientation overlay issues, and add the ability to customize DICOM tag values	Modified	~1 week

Hip Fracture Detection Inference Operator	Used to deploy a non-Clara Train image classification model (TensorFlow[32] Object Detection[33] based)	In-House	~1 week
Clara Deploy XNAT ROI Collection Exporter	Used to export segmentation results to XNAT, and create segmentation feedback entries in XNAT	In-House	~1 week
Clara Deploy Volume Calculator	Used to calculate segmentation volumes and export them to XNAT.	In-House	~1 week

180 Efforts listed are based on estimates for an experienced software developer familiar
181 with python, medical imaging APIs and containerization technologies, and will vary
182 based on skill level and experience with requirements, underlying technologies, and
183 interfaces.

184

185 Modified Clara Deploy operators are derived from reference Clara Deploy
186 operators, where the application source code has been modified to fit specific pipeline
187 needs. Modifications were necessary, for example to handle data in unexpected
188 formats, to support pipeline specific selection of a subset of exam data relevant for
189 inference, or to read/write DICOM metadata needed for clinical integration.

190 Entirely new operators, written in Python[34], are developed to provide functionality
191 not present in Clara or integrate non-Clara Train based models. The application code is
192 built on top of a base Docker image containing the Clara libraries necessary to
193 integrate with the Clara Deploy framework and other pipeline operators.

194 Management of results

195 Inference results during validation are stored separately from the clinical record,
 196 using XNAT, which provides a Java[35] plugin architecture[36] to support custom data
 197 schemas and extend functionality. Custom plugins were built for each pipeline to store
 198 non-DICOM output (such as volumetrics and classification results) with the
 199 corresponding images, customize how results are displayed to clinicians, and define
 200 feedback forms to assess model performance and clinical efficacy. The AI model name
 201 and version used to generate the results are stored with all results. Table 4 lists the
 202 plugins used in the present system and indicates which were developed in-house.

203 **Table 4: List of XNAT plugins, with estimate of software engineering time necessary to**
 204 **develop similar functionality.**

Plugin Name	Purpose	Development	Effort
XNAT-OHIF Viewer Plugin	DICOM image viewer, with segmentation and ROI contour support[37]	XNAT Team	
XNAT LDAP Authentication Provider Plugin	Integrates XNAT user accounts with UCSF's Active Directory system for authentication[38]	XNAT Team	
XNAT Container Service	Controls processing jobs using Docker containers on data stored in XNAT[39]	XNAT Team	
Hip Fracture Datatype	Stores inference results and feedback	In-House	~1 week
Brain Tumor Segmentation Datatype	Stores inference results and feedback	In-House	~1 week

Liver Donor Segmentation Datatype	Stores inference results and feedback	In-House	~1 week
ROI Volume Datatype	Stores segmentation volume measurements, linked to the segmentation, model name, and model version	In-House	~ 1 day
DICOM Import Identifier	By default, XNAT pulls subject and session information from the DICOM tags PatientName and PatientID when storing images; To integrate with clinical data, this plugin sets up the DICOM SCP to use the DICOM PatientID and AccessionNumber tags to define subject/session	In-House	~1 day

205 Effort will depend on skill level and experience with underlying technologies and
 206 interfaces.

207

208 A sample XNAT results page from the Brain Tumor Segmentation pipeline
 209 (described below) is shown in Fig 5. When a clinician views this page, they can see: the
 210 calculated tumor volumes for the patient’s current and prior exams (in table and graph
 211 format); the percent change of each tumor volume, relative to a baseline, and whether
 212 that percent change is above a threshold for tumor progression; the segmentation
 213 overlaid on top of the source DICOM image; and a feedback form. Clinicians can view
 214 the segmentation results, assess tumor progression, and leave feedback on model
 215 performance. They can also edit the model's segmentation and save a corrected copy
 216 back into XNAT. The volume of an edited segmentation is automatically calculated, via
 217 XNAT’s container service[39] and an in-house developed volume calculation container,

218 and added back into the displayed tumor volume table and plot (without overwriting
219 the original inference results).

220

221 **Fig 5: Brain Tumor ML Segmentation and progression display with feedback capture**
222 **webpage in XNAT.** Contains form for assessing model results, interactive plot and table
223 showing the patient's segmentation volumetrics at each exam time point (top), and
224 embedded OHIF viewer with editable lesion segmentation (bottom).

225

226 **Pipeline development and deployment**

227 Operators, pipelines, and XNAT plugins are developed and initially deployed on a
228 clinical VM dedicated to testing. This test VM runs its own instances of Clara Deploy
229 and XNAT. Pipeline definitions and XNAT plugins/configurations are pulled from an on-
230 premises Gitlab[40] instance and deployed. Kubernetes and the Docker daemon are
231 configured to pull Docker images from the Gitlab container registry. Test data cohorts
232 are manually sent through pipelines via DICOM transfers from PACS, and clinicians
233 review results in XNAT for feedback on usability, design, and the metrics that should
234 be captured about AI model performance. Once it's verified that a pipeline can
235 successfully ingest a clinical exam, select the correct image/s for processing, and
236 output the expected inference results, the inference pipeline is deployed on the
237 production VM in the same manner.

238 As these inference services are integrated with clinical resources, they reside on
239 infrastructure maintained by Radiology Clinical IT. This provides a high level of
240 monitoring and service support, which is necessary for the service up-time required by
241 clinicians participating in validation studies, including after-hours. This also provides an
242 added layer of security, as the inference VM's are isolated behind clinical firewalls. As a
243 result, only authorized personnel have access to these systems for deployment and
244 operations.

245 **Proof of concept validation projects**

246 Three AI models, described below, were chosen to pilot clinical pipeline integration.
247 Two of the models were developed in Clara Train, using built-in model architectures,
248 and trained on imaging data acquired at UCSF. The third model was trained and
249 developed outside of the Clara Train framework, utilizing TensorFlow's Object
250 Detection API[33]. The 3 clinical validation studies involved clinicians from different
251 departments within UCSF (Radiology and Biomedical Imaging, Surgery, and
252 Emergency Departments), and received institutional review board approvals with
253 consent waivers. The purpose of these proof-of-concept (POC) projects in the present
254 work is to assess model deployment and integration. Model development and training
255 is beyond the present scope and details are provided in references below.

256 **Brain Tumor Segmentation**

257 A Clara Train 3D U-Net[41] was trained to segment non-enhancing lesions from 3D
258 post-surgical MRIs of patients with low grade gliomas (LGG). Segmented volumes are
259 used to compute tumor volume for the current exam and priors. This was incorporated

260 into a deployed pipeline that aimed at detecting volumetric changes from baseline to
261 monitor for disease progression. A clinical validation study was run to assess whether
262 AI-based segmentation could be incorporated into patient care to detect non-
263 enhancing glioma progression (Fig 5).[42]

264 **Liver Transplant Segmentation**

265 The same Clara Train 3D U-Net model architecture was used to develop a liver
266 segmentation model for use in surgical planning for transplants.[43] A pipeline was
267 developed to automatically segment both the left and right liver lobes as well as
268 vessels from CT images and then calculate volumetrics (Fig 2). Surgeons use the
269 segmentations and the calculated volumes to determine transplant viability and plan
270 the surgical approach. This project captures timing metrics, of both segmentation
271 pipeline execution and review of the results, to compare against current manual and
272 semi-automated segmentation workflows.

273 **Hip Fracture Detection**

274 A third pipeline utilizes an object detection and classification model to localize the
275 left and right hip in x-ray images, and classify each as normal, containing a fracture, or
276 as having surgically implanted hardware.[44] This model was developed by UCSF's
277 Musculoskeletal Quantitative Imaging Research group[45], outside of Clara Train, using
278 the TensorFlow Object Detection model framework. Compass is configured to route
279 pelvis exams from 2 x-ray scanners in the UCSF Emergency Department to the Clara
280 Deploy inference server (Table 1). The deployment is being assessed for its ability to

281 improve emergency room outcomes by improving hip fracture diagnosis and reducing
282 a patient's time to treatment (Fig 2).

283 **Results**

284 The system detailed above was used to deploy the 3 POC projects to support
285 validation studies aimed at characterizing all aspects of pipeline development and
286 integration from data flow to system performance, extensibility, engineering robustness
287 and usability. The present section focuses on results related to characterizing the
288 system's viability as a general-purpose platform for supporting clinical validation of AI
289 models for a variety of representative workflows, workloads and use cases. Specific
290 details pertinent to the clinical use, model performance, and clinical impact of each
291 model is beyond the scope of this paper and will be presented separately.

292 The Brain Tumor Segmentation (BTS) pipeline initially received 30-40 exams per
293 week via automatic Compass routing, from 2 clinical MRI scanners (Table 1), for
294 inference. The imaging protocols that incorporate the sequence used to train the model
295 last around 40 minutes, and images were routed to our Clara VM over that entire
296 timeframe, with a typical exam containing about 1GB of data. The BTS pipeline, which
297 segments LGG tumor and calculates the segmentation volume, takes on average 2.9
298 minutes to execute per exam, including the time to transfer the results to XNAT, with
299 90% of cases processing in under 4.5 minutes. This execution time is not only
300 sufficient for processing automatically routed cases, but also met the requirements of
301 radiologists participating in the model validation study, who requested a <10-minute

302 turnaround time per exam when manually transmitting images from PACS. A
303 longitudinal analysis to assess tumor progression for a patient over 6 timepoints takes
304 <18 minutes to process. A clinical neuroradiologist reviewer, logged-in to a PACS
305 client, was able to search for, transmit, and receive results for 65 current and
306 retrospective MRI exams, across 10 LGG patients, in a 3.1-hour window. The
307 segmentation pipeline completed successfully for all cases. Results and findings from
308 this validation study were reviewed (see Discussion: Governance and Validation
309 Criteria), and supported adoption of the pipeline for routine clinical use at UCSF.

310 The Liver Transplant Segmentation (LTS) pipelines received 3-5 exams per week via
311 automatic Compass routing. This study involved blind reading of 3 different
312 segmentation results from the same exam: expert human reviewer, novice human
313 reviewer and machine segmentation. Human reviewers require 1-2 hours to produce
314 the segmentation, whereas AI results can be delivered in less than 10 minutes. Each
315 segmentation is identified with a unique salted hash that is inserted into the series
316 description of the DICOM Segmentation Object when it is written. This identifier is then
317 stored as a text file which is passed to the ROI Collection Exporter and the Volume
318 Calculator (Table 3) to ensure that the source of the segmentation is retained but
319 appropriately obscured from the reader. Three different clinicians then reviewed the
320 segmentation in XNAT via OHIF and provided feedback in forms linked with each
321 case's unique hash. Clinicians reported review times of 1-5 minutes per case.

322 Over 13 weeks, the Hip Fracture Detection (HFD) pipeline processed 200 exams
323 from 1 emergency department x-ray scanner, sent automatically via Compass routing.

324 Exams typically contained five 2D images of 5MB each. The observed transfer time for
325 a single exam was <1 minute, which defined the patient level time-out for triggering the
326 pipeline, and the inference pipeline's average run time was 20 seconds, including
327 uploading of results to XNAT. The workflow for this pilot study incorporated a "human
328 in the loop" (HITL) step, where a member of the ci²'s 3D Lab[46] assessed the quality
329 and relevance of each input x-ray image that was processed, before placing the
330 inference results on the XNAT worklists of the 2 participating clinical readers, who
331 assessed whether the AI model correctly identified the hip joints in the image and
332 made their own read on whether each joint contained a fracture, no-fracture, or
333 hardware. The HITL quality control workflow was implemented in the HFD XNAT plugin
334 and took the reviewer 1 minute per exam.

335 Discussion

336 Deploying and supporting an ML pipeline in the present framework requires
337 software development and system engineering on multiple fronts. The model must be
338 trained, AI inference operator built, and pipeline execution steps designed; pipeline
339 operators performing additional calculations or data tasks must be built; XNAT plugins
340 need to be developed to store and display result and capture user feedback; finally,
341 data ingestion, pipeline execution, and results display must be tested with clinical data,
342 which will differ from research data in unforeseen ways. Operator and plugin
343 development efforts are estimated in Tables 3-4, but will vary based on skill level and
344 experience. Collaboration with the clinical users is critical to define data display and

345 data flow requirements. At UCSF, ci²'s Computational Core[47] (i²c²) supports this
346 effort bridging the gap between scientific research, software engineering and enabling
347 translation of AI research into the clinic.

348 **Pipeline development efforts**

349 Deploying the Clara Deploy framework and linking it with imaging sources requires
350 knowledge of DICOM protocols, tooling, experience with Python application
351 development and containerization. Development requires familiarity with multiple
352 imaging data formats, AI development frameworks and communication with web
353 services via REST API's. The use of XNAT to create interactive data views entails web
354 development skills and since the framework integrates with clinical systems,
355 knowledge of security best-practices is critical.

356 **Pipeline deployment considerations**

357 Clinical Integration of AI pipelines involves collaboration across multiple
358 organizational units. i²c² engineers coordinate with: Clinical IT, to configure image
359 routing and PACS integration; Clinical Infrastructure, which hosts and maintains the
360 VM's and networking; data scientists and researchers who develop AI models; and
361 clinicians, to define image routing rules, develop effective visualizations in XNAT,
362 gather model feedback, and determine how AI results can integrate with already
363 complex clinical workflows. Models are increasingly incorporating imaging and non-
364 imaging data, e.g., from Electronic Medical Records (EMRs), further increasing the
365 complexity of the landscape. UCSF's APeX Enabled Research (AER) group[48]

366 provides a support path for SMART-on-FHIR[49] enabled EMR access for translational
367 work.

368 **Deployment framework flexibility**

369 Academic medical centers have diverse sets of research groups, spread across
370 departments, doing ML model development. Groups will have independently
371 developed unique model training toolsets, using custom software based on a variety of
372 ML frameworks. A deployment system needs to support integrating models and their
373 supporting code from outside of its ecosystem. While Clara Deploy supports running
374 Clara Train developed models by building a configuration file into the base inference
375 operator, it was also possible to integrate the HFD TensorFlow Object Detection model
376 into a pipeline by building a custom inference operator with refactored research code
377 and the Clara APIs. Engineering teams supporting clinical ML deployment need to
378 encourage scientific research groups to follow software best-practices. Integrating
379 research ML models into reliable clinical pipelines requires software to be packaged
380 into documented, reusable libraries.

381 **Deployment framework modularity**

382 Deploying our first pipeline (BTS) required modifying 3 Clara Deploy operators and
383 the development of 2 custom operators (see Fig 4 and Table 3). The LDS pipeline was
384 able to re-use all of those operators, significantly reducing the engineering effort to
385 deploy. Many pipelines have similar pre- and post-inference needs, and scaling
386 functionality across use-cases is integral to supporting the deployment of multiple ML
387 models.

388 **Deployment framework updates and transitions**

389 Leveraging established, well-tested and supported third-party software frameworks
390 for development offers significant advantages for development cost and product
391 stability; it does, however, pose risks that may include managing changes to APIs or
392 dependencies losing maintenance support and which would present substantial
393 implications for project effort and direction. Choosing to work with open-source
394 software that has strong, communicative leadership is key to mitigating such risks.
395 Building on open standards and industry protocols[50] ensures code portability, and
396 communication within a framework's community will lead to smoother upgrade cycles.
397 NVIDIA's Clara project is transitioning[51] into the Medical Open Network for Artificial
398 Intelligence (MONAI)[52], and though work will be necessary to move from Clara
399 Deploy to MONAI Deploy, the 2 projects' open natures and strong communication
400 within the MONAI Working Group (which includes NVIDIA)[53] promise a minimally
401 impactful transition.

402 **Access**

403 As the primary purpose of the clinical PACS or EMR system is to directly support
404 patient care by providing physicians access to data, clinical IT teams must prioritize
405 the stability and performance of clinical use cases. Any new system that plugs into
406 infrastructure critical to patient care is a potential risk, must be robust and not
407 negatively impact network infrastructure or IT support teams. Clinical integration of a
408 translational framework may thus require flexibility to adapt to authorized access
409 methodologies.

410 Another design consideration during a validation study is balancing the need to
411 segregate AI validation data from the clinical record while simultaneously providing
412 seamless access to AI results for clinical readers within their existing workflow. The
413 present XNAT-based approach to storing AI-derived results is thus designed to
414 facilitate prospective clinical validation of AI models, by providing clinicians seamless
415 access to results from a button in the clinical PACS, while validation results and
416 feedback are stored separately from clinical information systems (Fig 2). A web-based
417 approach allows clinicians flexibility in how and when they review cases, but does not
418 necessarily represent a final solution that fits into a clinical workflow.

419 **Clinical workflows**

420 Radiologists are faced with demanding workloads, and validation workflows must
421 be designed with efficiency in mind, as every additional mouse-click represents an
422 obstacle to adoption.[54] Any new information that an AI model provides must yield
423 clear, concrete improvements to patient care or a clinician's workflow. Ideally, new
424 information would be integrated into an existing tool; however, radiology workflows are
425 largely built around commercial applications which may or may not offer endpoints or
426 APIs for an AI pipeline to interact with. Moreover, even when integration points to
427 hospital wide applications (PACS, EMR) exist, obtaining access approval for
428 translational work may entail a lengthy and uncertain approval process. AI results
429 should be stored in standardized, open formats to allow for flexibility in presentation
430 method within the clinical workflow.[55]

431 **Governance and validation criteria**

432 The decision to use AI results in routine patient care and include them in the clinical
433 record requires careful consideration and a defined governance plan.[56] Ci²'s Clinical
434 Deployment committee[57] provides governance over such decisions and reviews all
435 potential AI applications through a structured cost-benefit analysis process.
436 Application evaluation considers model accuracy, connectivity, and robustness, and
437 the potential impacts to operations and workflow. Pipelines must seamlessly deliver
438 consistently interpretable results within a clinical context. Operational cost, mode of
439 integration, and the benefit and risk to patients and clinicians for reasons ranging from
440 potential model bias to implications from erroneous results and physician “automation
441 complacency”[58] are considered. Ultimately, a pipeline needs to provide clear
442 improvements over the existing standard-of-care.

443 The current platform provides a streamlined mechanism for gathering the real-world
444 feasibility and performance metrics necessary for a governance body to assess
445 whether a given model and implementation is a candidate for routine clinical use.

446 **Conclusion**

447 Implementing a generalized, extensible, and scalable platform for validating and
448 deploying AI-based pipelines in the clinic takes time and effort from a dedicated
449 engineering team, in collaboration with clinical end users capable of providing
450 guidance on usability and requirements. There is a considerable amount of work in
451 system design, infrastructure setup, and software engineering to ensure high reliability

452 and support for a diverse set of workloads and workflows, but the upfront investment
453 does return significant value. The server and network architecture put in place will
454 ultimately support any standards-compliant ML model deployment framework used
455 down the road, should a software transition become necessary, and once connectivity
456 with clinical systems is operational the same architecture will support additional
457 servers and pipelines.

458 Similarly, choosing to use a deployment framework designed to be modular and
459 built on open-source tooling will have benefits for the future. Clara Deploy's modularity
460 has meant that after functionality has been developed for one pipeline, it can be re-
461 used in future workflows, dramatically decreasing the time to deploy new AI models
462 that share similar pre- and post-processing needs. The ability to access and extend
463 Clara Deploy operator source code was essential to developing pipelines and
464 operators that can interact with clinical data and resources that always have edge
465 cases that differ from what a framework's developers expect. Building on standard,
466 open-source software tools also ensures a level of portability should deployment
467 requirements or frameworks evolve.

468 The inclusion of XNAT to store results external to the clinical record was also key in
469 developing an AI model validation framework. XNAT is valuable not only as a multi-
470 modal data manager, but also for its extensibility, which allows it to act as a
471 customizable validation study platform. Having a web frontend to AI results also
472 enables rapid iteration on interactive UIs for presenting model output and generating
473 final reports on findings.

474 While commercial ML model deployment options exist, the choice to build an in-
475 house solution preserves flexibility in data routing, infrastructure, ML model framework
476 choice, and project-specific workflow, visualization, and validation requirements. This
477 is particularly important for supporting translational work for a broad range of use-
478 cases being developed in a large research institution. Leveraging robust open-source
479 components significantly reduces development efforts while providing adaptability and
480 improving resilience. The process and systems outlined above have been
481 demonstrated to provide a flexible and dependable ML model deployment platform,
482 that will scale across pipelines and use-case specific requirements and handle the
483 deployment process from validation study to integration into the clinical workflow.

484 **Acknowledgements**

485 For their hard work in supporting the infrastructure and networking in the above
486 system, we would like to thank: Jeff Block, Matt Denton, and Reese Webb from
487 Radiology Clinical Infrastructure; Peter Storey and Jed Chan from Radiology Scientific
488 Computing Services; Neil Singh and Muse Hsieh from Radiology Clinical IT Operations;
489 and Dr. Wyatt Tellis from Radiology Innovation and Analytics. We would also like to
490 thank Dr. Mona Flores, Dr. Sidney Bryson, Rahul Choudhury, Victor Chang, and David
491 Bericat from NVIDIA for help deploying and developing Clara at UCSF.

492 **Funding**

493 This work had no direct funding sources.

494 **Conflict of interest statement**

495 NVIDIA provided 4 T4 cards as a grant to UCSF.

497 **References**

- 498 1. Radiological Society of North America. Imaging AI in Practice demonstration.
499 [cited 7 Dec 2022]. Available: [https://www.rsna.org/education/ai-resources-and-](https://www.rsna.org/education/ai-resources-and-training/AI-Imaging-in-Practice)
500 [training/AI-Imaging-in-Practice](https://www.rsna.org/education/ai-resources-and-training/AI-Imaging-in-Practice)
- 501 2. Brink L, Coombs LP, Kattil Veettil D, Kuchipudi K, Marella S, Schmidt K, et al.
502 ACR's Connect and AI-LAB technical framework. JAMIA Open. 2022;5.
503 doi:10.1093/jamiaopen/ooac094
- 504 3. Ng A. Bridging AI's Proof-of-Concept to Production Gap. [cited 7 Dec 2022].
505 Available: <https://www.youtube.com/watch?v=tsPuVAMaADY>
- 506 4. Gupta V, Erdal BS, Ramirez C, Floca R, Jackson L, Genereaux B, et al. Current
507 State of Community-Driven Radiological AI Deployment in Medical Imaging.
508 2022.
- 509 5. Visage Imaging Inc. Accelerated AI - Visage Imaging. [cited 7 Dec 2022].
510 Available: <https://visageimaging.com/platform/acceleratedai/>
- 511 6. CARPL.AI Inc. CARPL - CARING Analytics platform. [cited 7 Dec 2022].
512 Available: <https://carpl.ai>
- 513 7. Arterys Inc. Medical Imaging Cloud AI - Arterys. [cited 7 Dec 2022]. Available:
514 <https://www.arterys.com>

- 515 8. General Electric Company. Edison Artificial Intelligence & Analytics | GE
516 Healthcare (United States). [cited 7 Dec 2022]. Available:
517 <https://www.gehealthcare.com/products/edison>
- 518 9. NVIDIA. NVIDIA Clara Imaging. [cited 7 Dec 2022]. Available:
519 <https://developer.nvidia.com/clara-medical-imaging>
- 520 10. Marcus DS, Olsen TR, Ramaratnam M, Buckner RL. The extensible neuroimaging
521 archive toolkit: An informatics platform for managing, exploring, and sharing
522 neuroimaging data. *Neuroinformatics*. 2007;5: 11–33. doi:10.1385/NI:5:1:11
- 523 11. University of California San Francisco. The Center for Intelligent Imaging. [cited 7
524 Dec 2022]. Available: <https://intelligentimaging.ucsf.edu>
- 525 12. National Electrical Manufacturers Association. NEMA PS3 / ISO 12052, Digital
526 Imaging and Communications in Medicine (DICOM) Standard. Rosslyn, VA, USA:
527 (available free at <http://medical.nema.org/>);
- 528 13. Choplin RH, Boehme JM, Maynard CD. Picture archiving and communication
529 systems: an overview. *Radiographics: a review publication of the Radiological*
530 *Society of North America, Inc.* 1992. doi:10.1148/radiographics.12.1.1734458
- 531 14. NVIDIA. NVIDIA Clara Deploy SDK User Guide. [cited 7 Dec 2022]. Available:
532 <https://docs.nvidia.com/clara/deploy/index.html>
- 533 15. Visage Imaging Inc. Visage Imaging. [cited 7 Dec 2022]. Available:
534 <https://visageimaging.com>

- 535 16. Laurel Bridge Software. Compass - DICOM and HL7 enterprise router. [cited 7
536 Dec 2022]. Available: <https://www.laurelbridge.com/products/compass/>
- 537 17. VMWare. What is vSphere 7 | Server Virtualization Software. [cited 7 Dec 2022].
538 Available: <https://www.vmware.com/products/vsphere.html>
- 539 18. Ubuntu. Enterprise Open Source and Linux | Ubuntu. [cited 7 Dec 2022].
540 Available: <https://ubuntu.com>
- 541 19. NVIDIA. Virtual GPU Software User Guide :: NVIDIA Virtual GPU Software
542 Documentation. [cited 7 Dec 2022]. Available:
543 <https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html>
- 544 20. Docker. Empowering App Development for Developers | Docker. [cited 7 Dec
545 2022]. Available: <https://www.docker.com>
- 546 21. The Kubertes Authors. Kubernetes. [cited 7 Dec 2022]. Available:
547 <https://kubernetes.io>
- 548 22. NVIDIA. NVIDIA Triton Inference Server. [cited 7 Dec 2022]. Available:
549 <https://developer.nvidia.com/nvidia-triton-inference-server>
- 550 23. Wikipedia. Representational state transfer - Wikipedia. [cited 7 Dec 2022].
551 Available: https://en.wikipedia.org/wiki/Representational_state_transfer
- 552 24. Open Health Imaging Foundation. Open Health Imaging Foundation. [cited 7 Dec
553 2022]. Available: <https://ohif.org>

- 554 25. The PostgreSQL Global Development Group. PostgreSQL: The world's most
555 advanced open source database. [cited 7 Dec 2022]. Available:
556 <https://www.postgresql.org>
- 557 26. F5 Networks. NGINX | High Performance Load Balancer, Web Server, & Reverse
558 Proxy. [cited 7 Dec 2022]. Available: <https://www.nginx.com>
- 559 27. Microsoft. Active Directory Domain Services Overview | Microsoft Docs. [cited 7
560 Dec 2022]. Available: [https://docs.microsoft.com/en-us/windows-
561 server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-
562 overview](https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview)
- 563 28. Wikipedia. Lightweight Directory Access Protocol - Wikipedia. [cited 7 Dec 2022].
564 Available: https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
- 565 29. NVIDIA. Bundled Operators. [cited 7 Dec 2022]. Available:
566 <https://docs.nvidia.com/clara/deploy/ClaraContainers/index.html>
- 567 30. Kitware. ITK/MetalO/Documentation - MetalImage. [cited 7 Dec 2022]. Available:
568 <https://itk.org/Wiki/ITK/MetalO/Documentation#MetalImage>
- 569 31. Data Format Working Group. NIfTI-1 Data Format. [cited 7 Dec 2022]. Available:
570 <https://nifti.nimh.nih.gov/nifti-1/>
- 571 32. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow:
572 Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available:
573 <https://www.tensorflow.org/>

- 574 33. Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, et al. Speed/accuracy
575 trade-offs for modern convolutional object detectors. 2016. Available:
576 https://github.com/tensorflow/models/tree/master/research/object_detection
- 577 34. The Python Software Foundation. Welcome to python.org. [cited 7 Dec 2022].
578 Available: <https://www.python.org>
- 579 35. Oracle. Java Programming Language. [cited 7 Dec 2022]. Available:
580 <https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html>
- 581 36. NRG at the Washington University School of Medicine. Developing XNAT
582 Plugins. [cited 7 Dec 2022]. Available: [https://wiki.xnat.org/documentation/xnat-](https://wiki.xnat.org/documentation/xnat-developer-documentation/working-with-xnat-plugins/developing-xnat-plugins)
583 [developer-documentation/working-with-xnat-plugins/developing-xnat-plugins](https://wiki.xnat.org/documentation/xnat-developer-documentation/working-with-xnat-plugins/developing-xnat-plugins)
- 584 37. NRG at the Washington University School of Medicine. XNAT-OHIF Viewer
585 Plugin. [cited 7 Dec 2022]. Available: [https://wiki.xnat.org/documentation/xnat-](https://wiki.xnat.org/documentation/xnat-ohif-viewer)
586 [ohif-viewer](https://wiki.xnat.org/documentation/xnat-ohif-viewer)
- 587 38. NRG at the Washington University School of Medicine. XNAT LDAP
588 Authentication Provider Plugin. [cited 7 Dec 2022]. Available:
589 <https://bitbucket.org/xnatx/ldap-auth-plugin/src/master/>
- 590 39. NRG at the Washington University School of Medicine. XNAT Container Service.
591 [cited 7 Dec 2022]. Available: [https://bitbucket.org/xnatdev/container-](https://bitbucket.org/xnatdev/container-service/src/master/)
592 [service/src/master/](https://bitbucket.org/xnatdev/container-service/src/master/)

- 593 40. Gitlab B.V. Iterate faster, innovate together | Gitlab. [cited 7 Dec 2022]. Available:
594 <https://about.gitlab.com>
- 595 41. Myronenko A. 3D MRI brain tumor segmentation using autoencoder
596 regularization. 2018.
- 597 42. Damasceno PF, Gleason T, Hawkins J, Luks T, Majumdar S, Lupo JM, et al.
598 Clinical Validation of Segmentation-Based Detection of Glioma Progression.
599 medRxiv. 2022 [cited 15 Nov 2022]. Available:
600 <https://www.medrxiv.org/content/10.1101/2022.08.17.22278562v1>
- 601 43. Olson M. Clinical Validation Study of Machine Learning Algorithms for Liver
602 Transplant Planning. UCSF Department of Radiology and Biomedical Imaging
603 Research Symposium. San Francisco, CA, United States; 2020. Available:
604 [https://radiology.ucsf.edu/research/meetings/imaging-research-](https://radiology.ucsf.edu/research/meetings/imaging-research-symposium#accordion-2020-imaging-research-symposium)
605 [symposium#accordion-2020-imaging-research-symposium](https://radiology.ucsf.edu/research/meetings/imaging-research-symposium#accordion-2020-imaging-research-symposium)
- 606 44. Krogue JD, Cheng K v, Hwang KM, Toogood P, Meinberg EG, Geiger EJ, et al.
607 Automatic Hip Fracture Identification and Functional Subclassification with Deep
608 Learning. Radiol Artif Intell. 2020;2. doi:10.1148/ryai.2020190023
- 609 45. University of California San Francisco. Musculoskeletal and Quantitative Imaging
610 Research. [cited 7 Dec 2022]. Available:
611 <https://radiology.ucsf.edu/research/labs/musculoskeletal-quantitative-imaging>

- 612 46. University of California San Francisco. 3D Lab | intelligent imaging. [cited 7 Dec
613 2022]. Available: <https://intelligentimaging.ucsf.edu/resources/3dlab>
- 614 47. University of California San Francisco. Computational Core | intelligent imaging.
615 [cited 7 Dec 2022]. Available:
616 <https://intelligentimaging.ucsf.edu/resources/computational-core>
- 617 48. UCSF Clinical and Translational Science Institute. Consultation for APeX Enabled
618 Research | CTSI Research Consultation. [cited 7 Dec 2022]. Available:
619 <https://consult.ucsf.edu/apex-enabled-research>
- 620 49. HL7 International - FHIR Infrastructure Working Group. HL7.FHIR.UV.SMART-
621 APP-LAUNCH\Overview - FHIR v4.0.1. [cited 7 Dec 2022]. Available:
622 <https://www.hl7.org/fhir/smart-app-launch/index.html>
- 623 50. IHE Radiology Technical Committee. IHE Radiology Technical Framework
624 Supplement – AI Workflow for Imaging. Aug 6, 2020. Available:
625 https://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_RAD_Suppl_AIW-
626 [I.pdf](https://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_RAD_Suppl_AIW-I.pdf)
- 627 51. Zephyr M. Clara Train 4.0 Upgrades to MONAI and supports FL with
628 Homomorphic Encryption | NVIDIA Technical Blog. [cited 7 Dec 2022]. Available:
629 [https://developer.nvidia.com/blog/clara-train-4-0-upgrades-to-monai-and-](https://developer.nvidia.com/blog/clara-train-4-0-upgrades-to-monai-and-supports-fl-with-homomorphic-encryption/)
630 [supports-fl-with-homomorphic-encryption/](https://developer.nvidia.com/blog/clara-train-4-0-upgrades-to-monai-and-supports-fl-with-homomorphic-encryption/)

- 631 52. MONAI Working Group. MONAI: Medical Open Network for Artificial Intelligence.
632 [cited 7 Dec 2022]. Available: <https://monai.io>
- 633 53. MONAI Working Group. About MONAI. [cited 7 Dec 2022]. Available:
634 <https://monai.io/about.html>
- 635 54. Agarwal M, van der Pol CB, Patlas MN, Udare A, Chung AD, Rubino J.
636 Optimizing the radiologist work environment: Actionable tips to improve
637 workplace satisfaction, efficiency, and minimize burnout. *Radiol Med.* 2021;126:
638 1255–1257. doi:10.1007/s11547-021-01397-x
- 639 55. IHE Radiology Technical Committee. AI Results - IHE Wiki. [cited 7 Dec 2022].
640 Available: https://wiki.ihe.net/index.php/AI_Results
- 641 56. Chang PJ, Kapur J, Kohli MD. AI Governance in Medical Imaging: How to Herd
642 the Cats and Avoid Chaos. RSNA 2022 Radiology Conference & Annual Meeting.
643 Chicago, IL, United States; 2022. Available:
644 <https://epro02.ativ.me/web/planner.php?id=RSNA22&tabid=2>
- 645 57. University of California San Francisco. Clinical Deployment | Intelligent Imaging.
646 [cited 7 Dec 2022]. Available:
647 <https://intelligentimaging.ucsf.edu/resources/clinical-deployment>
- 648 58. Ross P, Spates K. Considering the Safety and Quality of Artificial Intelligence in
649 Health Care. *The Joint Commission Journal on Quality and Patient Safety.*
650 2020;46: 596–599. doi:10.1016/j.jcjq.2020.08.002

651

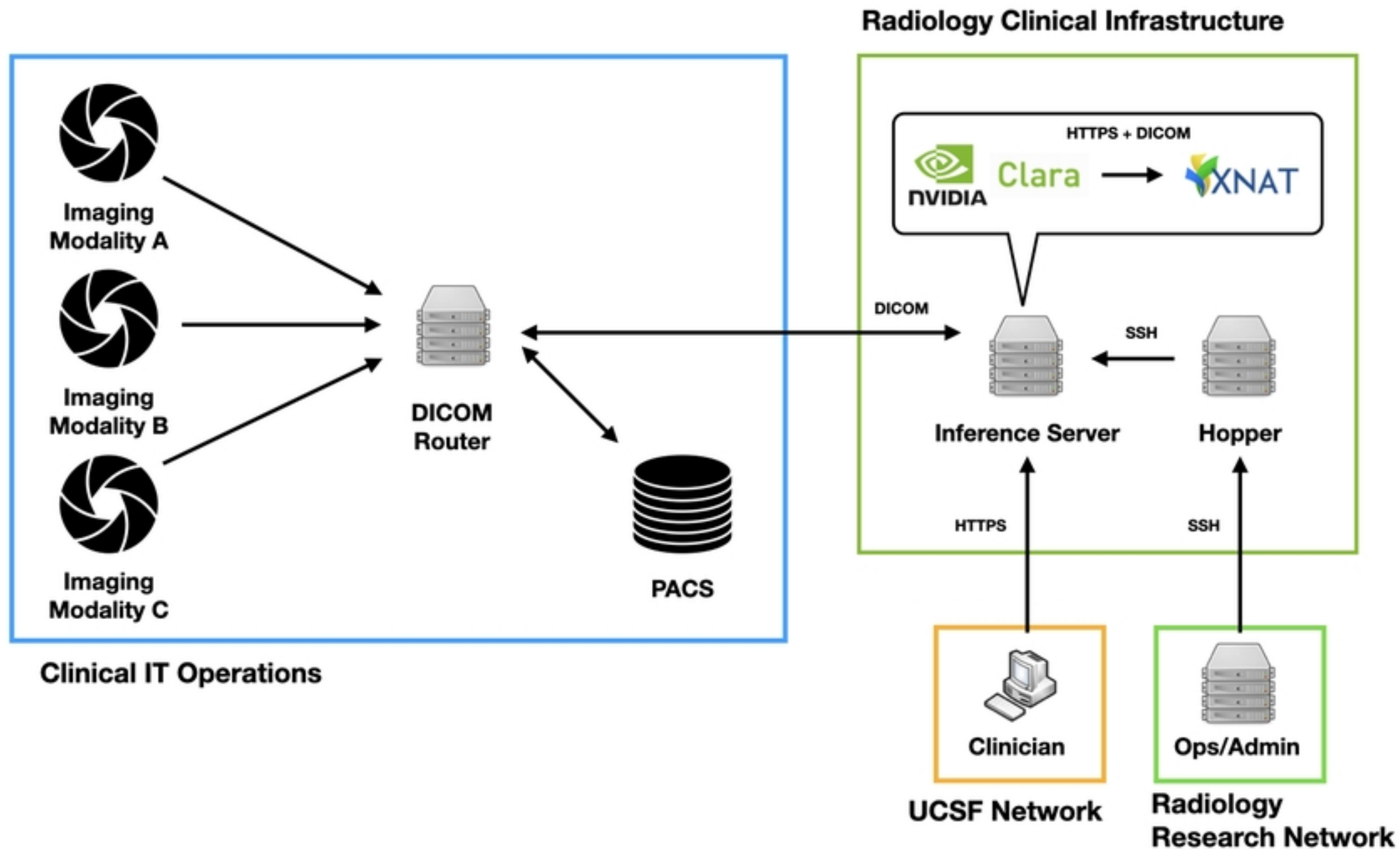


Figure 1

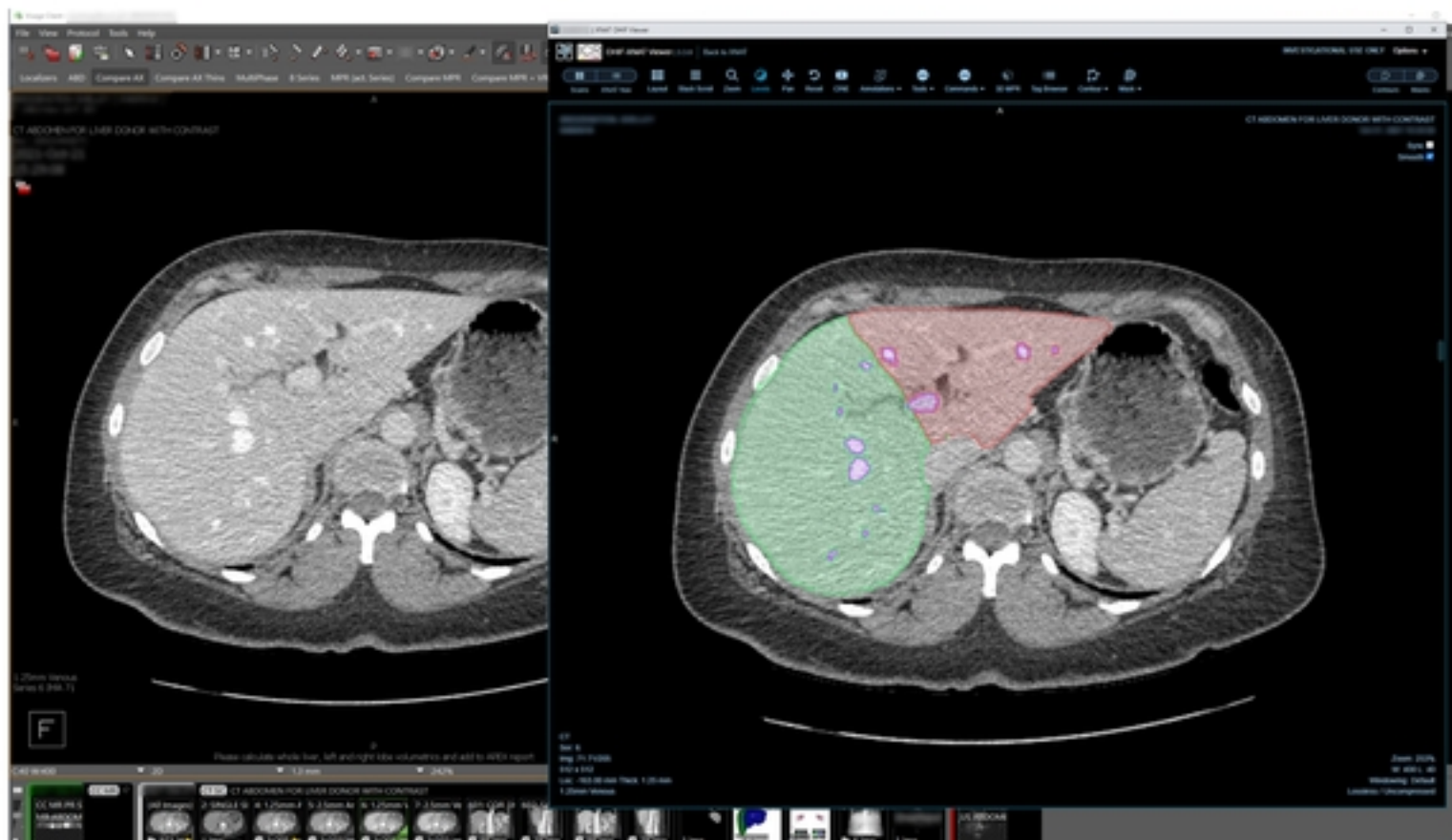
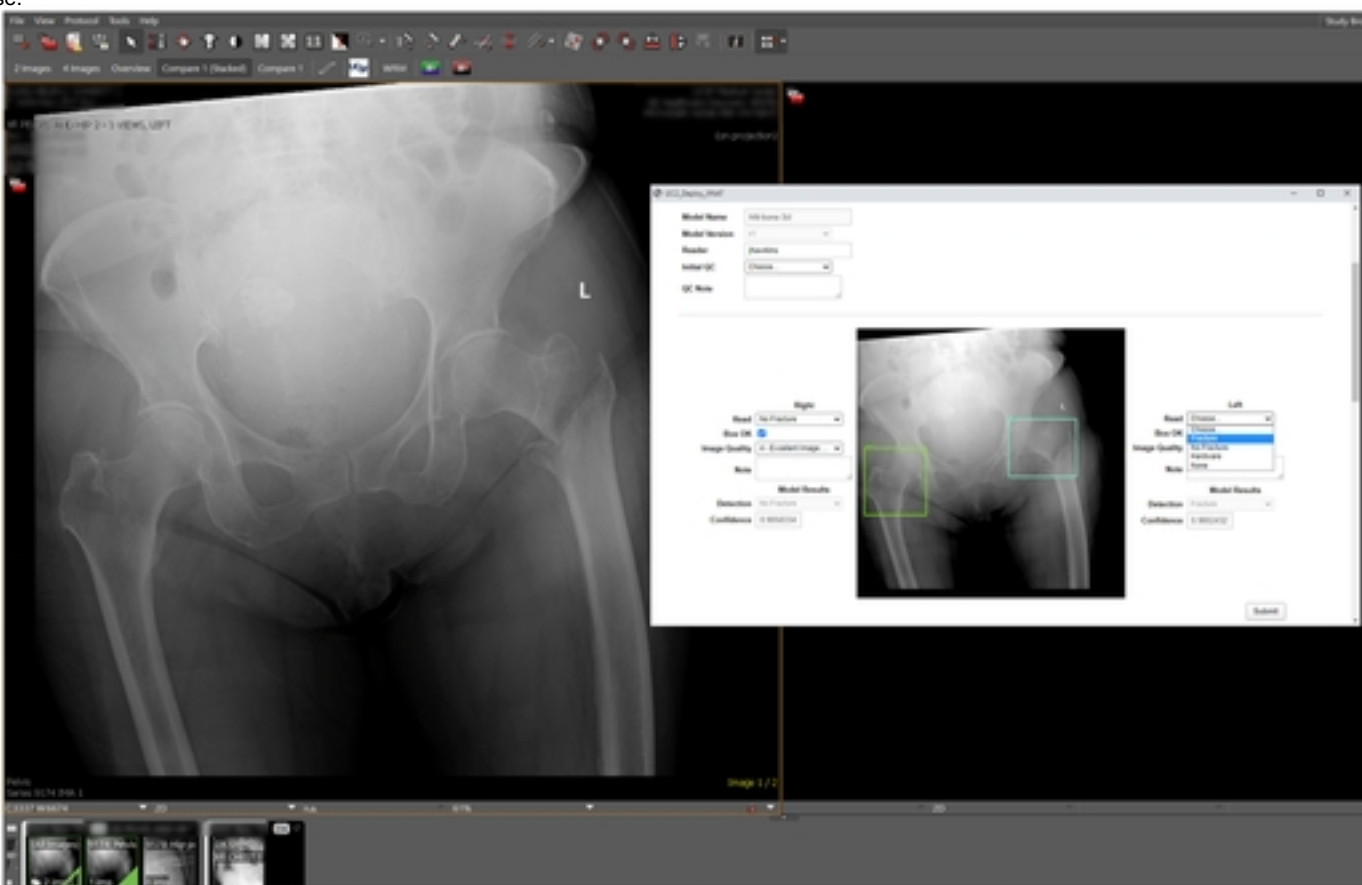
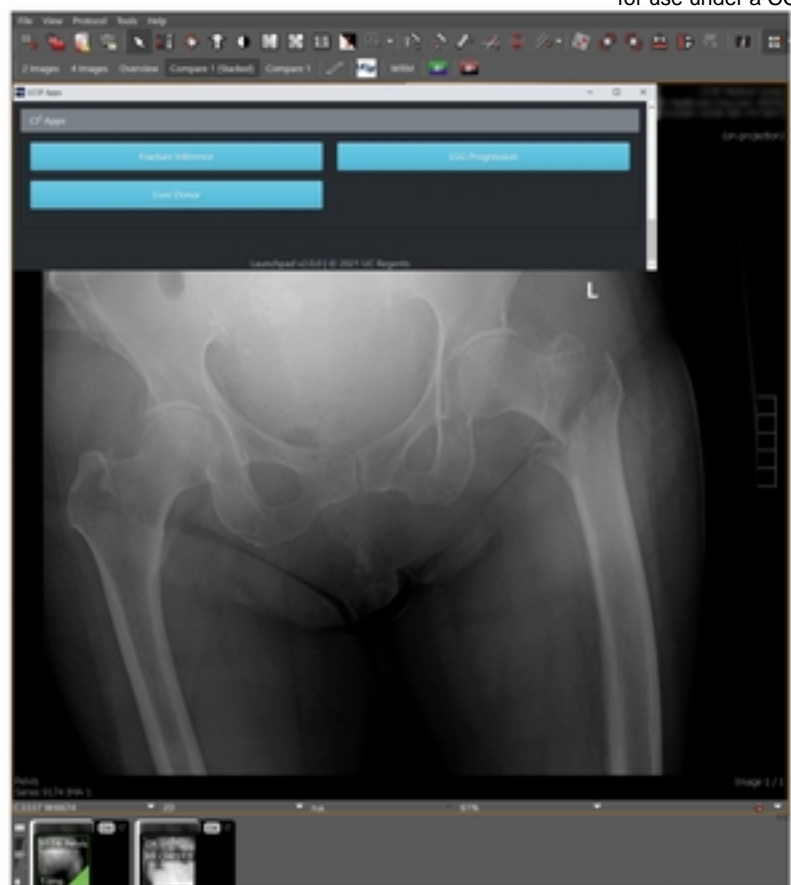


Figure 2

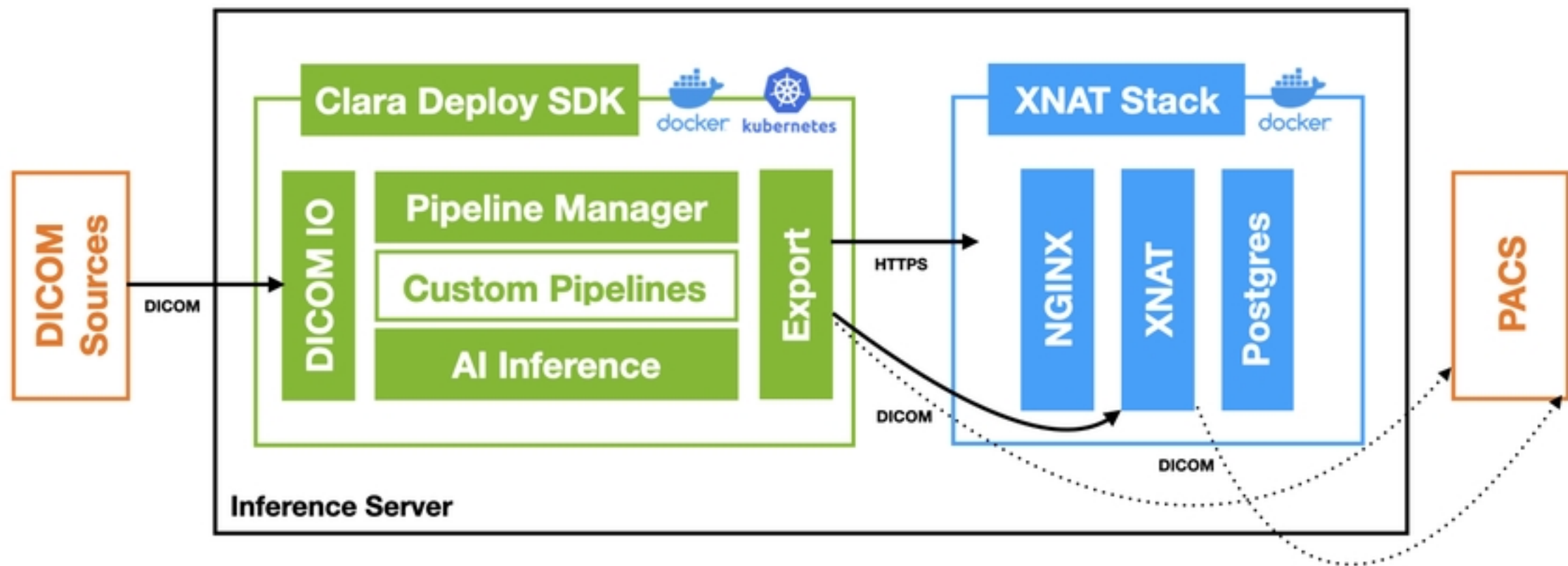


Figure 3

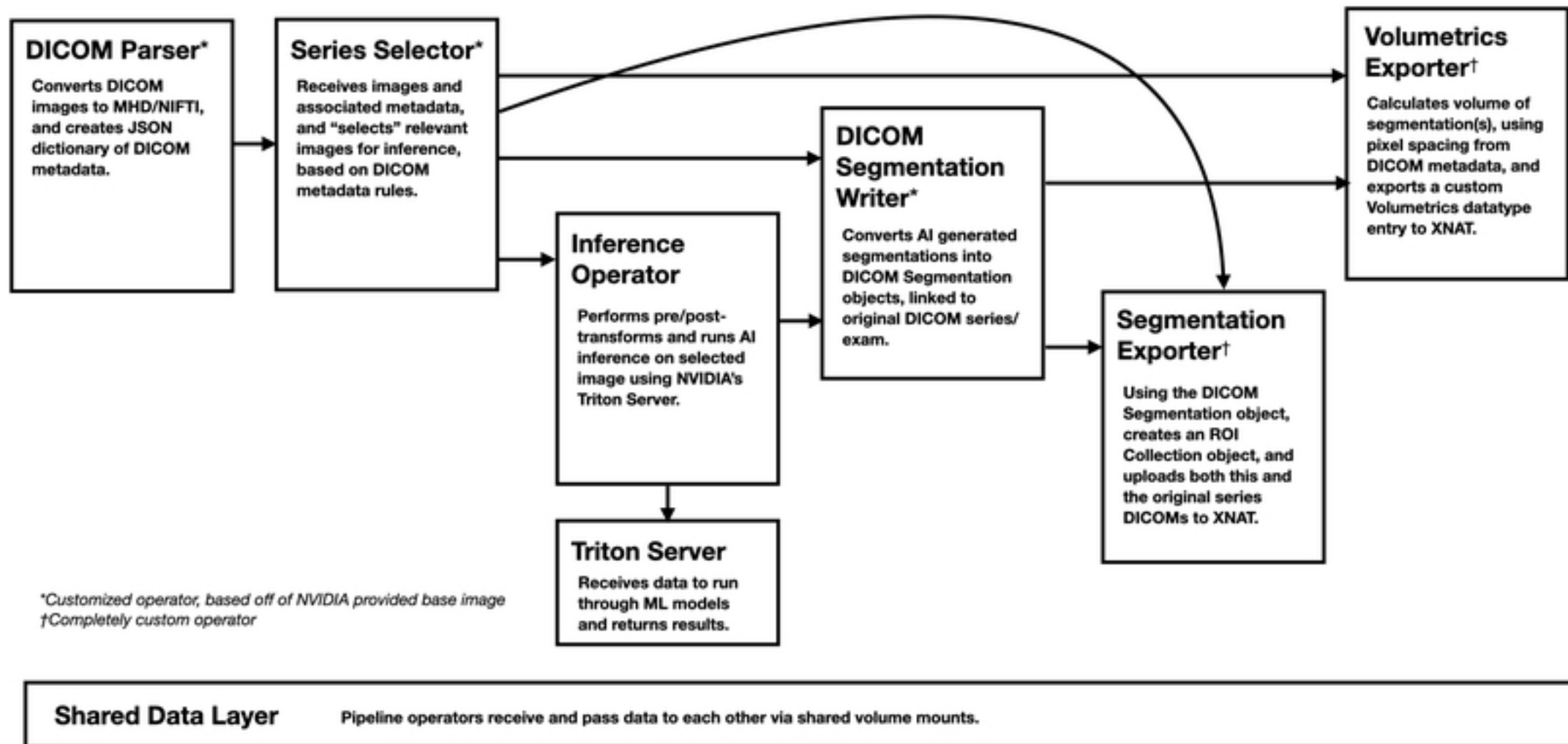


Figure 4

PROJECT: [Project] > SUBJECT: [Subject] > SESSION: [Session] > brain_tumor_FT2A

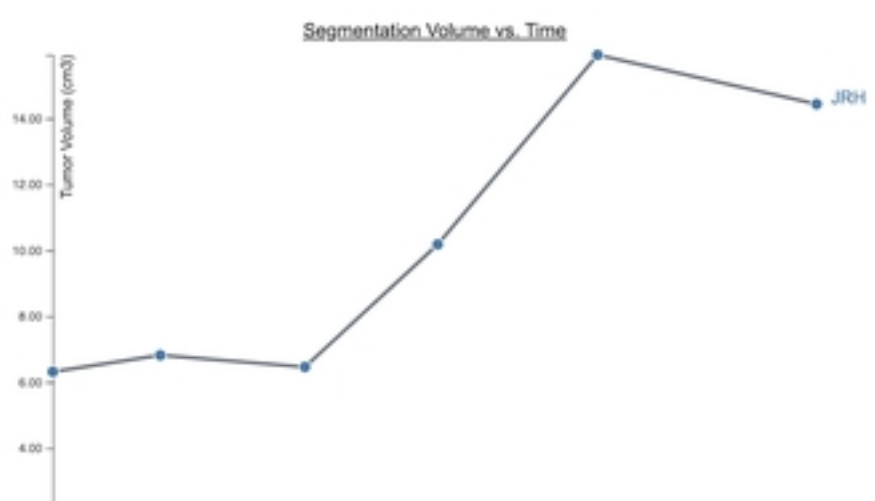
Flair T2 All Assessment

Patient: [Redacted]
 Assessment: [Redacted]
 Exam Date: [Redacted]

Progression Read: Progression
 Model Name: flair-2all
 Model Version: v3

Segmentation Score: 5 - Excellent
 Segmentation Edited: []
 Notes: []

Reader: jhawkins
 [Submit]



medRxiv preprint doi: <https://doi.org/10.1101/2023.03.07.23286951>; this version posted March 9, 2023. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted medRxiv a license to display the preprint in perpetuity. This article is a US Government work. It is not subject to copyright under 17 USC 105 and is also made available for use under a CC0 license.

ROI Version Displayed: v3-QC

ROI Volumes and %Change to Current							
✓	Date	Accession	ROI	Version	Region	Volume (cm³)	%Change
✓	[Redacted]	[Redacted]	JRH	v3-QC	T2 All	6.316855	Baseline
✓	[Redacted]	[Redacted]	JRH	v3-QC	T2 All	6.8190136	7.9495
✓	[Redacted]	[Redacted]	JRH	v3-QC	T2 All	6.461947	2.2969
✓	[Redacted]	[Redacted]	JRH	v3-QC	T2 All	10.179794	61.1529
✓	[Redacted]	[Redacted]	JRH	v3-QC	T2 All	15.927261	152.1391
✓	[Redacted]	[Redacted]	JRH	v3-QC	T2 All	14.435079	128.5169

40 % change in volume for progression [Recalculate]

[Draw Line] [Generate DICOM Report]

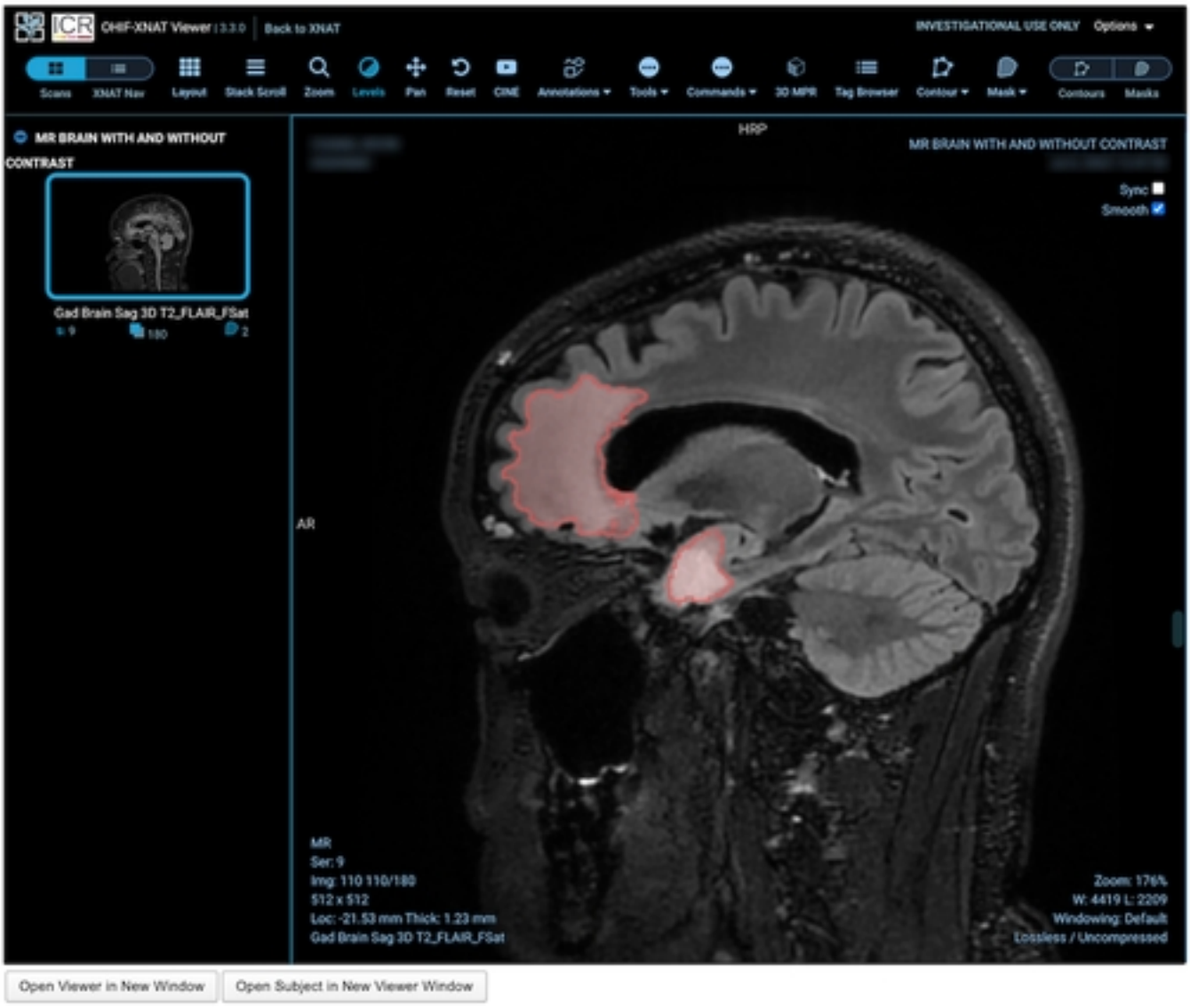


Figure 5