Integrating the Healthcare Enterprise

IHE Radiology White Paper

AI Interoperability in Imaging

Revision 1.0 – Public Comment

Date: March 12, 2021

Authors: Brad Genereaux, Brian Bialecki, Karl Diedrich, Kevin O'Donnell,
Chris Roth, Antje Schroeder, Neil Tenenholtz, Harald Zachmann,
and the IHE Radiology Community.

Email: radiology@ihe.net

Please confirm you have the most recent version of this document. See here for Published Versions and here for Public Comment versions.

Foreword

This is a white paper of the IHE Radiology domain.

This white paper is published on March 12, 2021 for Public Comment. Comments are invited and can be submitted at https://www.ihe.net/Radiology_Public_Comments. To be considered in development of the subsequent version of the white paper, comments must be received by April 11, 2021,

General information about IHE can be found at: www.ihe.net.

Information about the IHE Radiology domain can be found at: https://www.ihe.net/ihe_domains.

Information about the organization of IHE Technical Frameworks and Supplements and the process used to create them can be found at: http://ihe.net/IHE_Process and http://ihe.net/Profiles.

The current version of the IHE Radiology Technical Framework can be found at: http://ihe.net/Technical_Frameworks.

Navigating this Document

Please see Section 1.3 for suggestions on how to efficiently review the white paper.

Contents

1 Introduction

This document, the IHE Radiology AI Interoperability in Imaging White Paper, describes an organizing framework and roadmap for creating profiles to support the creation, lifecycle, and use of AI datasets and AI models.

1.1 Purpose of the AI Interoperability in Imaging White Paper

This white paper is intended to document what “AI in Imaging” encompasses, and provide a comprehensive map of the interoperability needs, problems, and challenges that must be addressed to achieve an ecosystem of interoperable products that support all the processes and tasks that make up AI in Imaging. The paper’s use of “imaging” as a shorthand is intended to cover all forms of medical imaging.

This document will be used by the IHE Planning and Technical Committees to identify logical groups of needs that would constitute functional Profiles. The enumerated needs, problems, and challenges will help ensure the Profiles are properly scoped and do not overlook issues that might only be apparent in the “big picture” or after more careful consideration.

1.2 Intended Audience

The intended audience of the IHE Radiology AI Interoperability in Imaging White Paper is:

- Members of the IHE Planning and Technical Committees
- Members of other standards bodies who create and support the standards necessary for AI (e.g., DICOM®[1], HL7®[2], LOINC®[3], etc.)

Many user communities that are also attempting to understand the scope of AI in Imaging and the various needs, problems, and challenges, should find this White Paper useful (and are strongly encouraged to contribute insights and feedback), including:

- Consumers of AI products (radiologists, technologists, informaticists, referring/attending physicians)
- Administrators of AI products (PACS admins, radiology leadership, health records and quality team)
- Developers, analysts, and data scientists
- Support staff for AI products (PACS admins, IT operations)

[1] DICOM® is the registered trademark of the National Electrical Manufacturers Association for its standards publications relating to digital communications of medical information.

[2] HL7®, is the registered trademark of Health Level Seven International and the use of this trademark does not constitute an endorsement by HL7

[3] LOINC® is registered United States trademarks of Regenstrief Institute, Inc.

- Vendors of imaging solutions (Image Manager products, Image Display products, RIS products, Modality Products)
- Vendors of AI products and services (AI Orchestrator and AI Performer products)
- E.g., AI product vendors may be informed of what integration partners and customers in the AI application ecosystem may expect.
- Developers of AI-related standards
- Strategic planning members, product managers, R&D engineering leads
- Health policy agencies and experts, national societies

1.3 Public Comment and Navigating this Document

The AI Interoperability in Imaging White Paper is a long document. Reviewers are encouraged to provide feedback, even if only on portions of the document. To optimize their efforts, reviewers may consider:

- Reviewing the introductory video presentation available on the IHE Wiki.
- Paying particular attention to the Questions in Section 1.4.
- Focusing on specific areas that align with their interests and expertise:
    - **2.1 Applications of AI in Imaging** focuses on activities that AI can be applied to. The use cases in chapter 3 will be driven by these applications. This section may appeal to **clinical reviewers**.
        - Note: this white paper specifically strives to broaden the map of applications beyond radiology image analysis. Reviewers are encouraged to consider:
            - Imaging specialties beyond radiology, such as cardiology, pathology, dentistry, ophthalmology, dermatology, etc.
            - Tasks beyond image analysis, such as ordering, acquisition, reporting, and departmental operations analysis.
- **2.2 Personas and Systems** provides brief definitions that may clarify the subsequent use case sections. This section may appeal to both **clinical and technical reviewers**.
- **3 Use Cases** focuses on the steps that create, update, and use AI Applications. At the beginning of each Use Case Group and Use Case includes a statement of the task covered in the block. This section may appeal to **data scientists** and **technical reviewers**.
- **4 Entities** focuses on common data structures and metadata in AI workflows. This section may appeal to **technical reviewers**.

1.4 Questions

In addition to feedback on any of the content and organization of the whitepaper, the Committee is particularly interested in feedback on the following questions:

1. What applications would you add in Section 2.1?

2. What details of the applications in 2.1 might need special consideration?
   E.g., related to who the AI task is performed for, how they will apply the result, or how they might interact with the AI Application.

3. What use cases would you add in Section 3?

   E.g., are there additional use cases needed to address specific applications in 2.1?

4. Is the organization of Use Cases into Use Case Groups helpful and are they properly divided?

   E.g., the discussion of server query capabilities to support Discovery of Repository Content has been put under Repository Use Cases since it relates to the concerns of the Repository Administrator, while the corresponding discussion of a client leveraging those query capabilities to Obtain Data Records is put under Dataset Assembly Use Cases since it fits in the flow of finding and processing from the data consumer viewpoint.

5. What specific iterations/loopbacks between use cases in Section 3 need elaboration?

   Very few iterations and loopbacks have been made explicit. It is assumed that many situations may result in jumping from almost any use case back to some prior use case. E.g., a failed regulatory validation of a model might lead back to additional training, or to a redesign in Orchestrate Training, or back to Obtaining additional Data Records.

6. What security related topics need to be added to the Use Cases or Common Mechanics to properly drive Profile development?

7. What metadata or details would you add to the Entities in Section 4?
   Specifically, what would you add to the Service and a Model Orchestration Framework?

8. What profiles (Section 5) should IHE write to address the applications and use cases identified in sections 2 & 3, and how should they be prioritized?

9. Where is more depth/concrete detail needed to initiate Profile work? Where does this white paper go too far?

10. What parts of the application/use case landscape is currently too unstable/dynamic to productively characterize now?

11. What proposed profiles should be developed in, or in coordination with, Pathology, Radiation Oncology, IT Infrastructure, etc.?

12. What standards would you add in Appendix C.3?

13. What gaps do you see in the identified standards that should be addressed to support development of IHE profiles?

14. Readers will find some empty bullets and are encouraged to suggest content for them.

1.5 Closed Issues

1. Will actors and transaction be defined?

No.

That is the job of subsequent Profiles. The point of the White Paper is to provide a functional overview of what "AI in Imaging" encompasses, and more importantly provides a good enumeration of the problems and challenges that will need to be addressed to achieve an interoperable ecosystem. That information will be the foundation for future decisions on scoping and prioritizing profiles.

The current focus is on determining what all the key questions are. The profiles will have the job of providing interoperable consensus answers.

2. AI in Imaging or AI in Radiology?

We decided on AI in Imaging, as it applies to enterprise imaging workflows outside of radiology as well. This also mirrors AIW-I (AI Workflow for Imaging).

3. What terms should the paper define or include in the glossary?

Many AI glossaries exist and IHE RAD resources are limited, so definition work has not been done unless the usage in this paper is more specific or differs from common usage of the term.

Specific terms that have been refined include:

- AI components (AI Model, AI Application)
- Datasets (Training Dataset, Testing Dataset, Validation Dataset)
- Data composition (Data Element, Data Record, and Dataset)
- Inference results (AI Result)

2 Landscape

This chapter describes the problems that AI is poised to solve and lays out the applications and personas in greater detail.

Artificial intelligence in medical imaging encompasses an exceedingly broad spectrum of topics. AI can be used to perform or support a wide variety of tasks. Medical imaging includes concepts in radiology, including broader radiology workflows beyond the images themselves (like ordering and reporting), enterprise imaging workflows (like dental and ophthalmology), and other imaging domains (like pathology).

Creating, training, validating, deploying, monitoring the performance of, and using artificial intelligence components in radiology is a large problem space with emerging technologies that need to be integrated into the large space of existing imaging departments and practices.

Having a roadmap will make it much easier to select and properly scope profile proposals and may also help to prioritize/sequence the work. Hashing out some basic points of agreement will likely lead to faster progress when the profiles are done. Vendors and users will benefit from any degree of alignment/harmonization.

2.1 Applications of AI in Imaging

This section identifies and categorizes applications to which AI and Machine Learning are applied in medical imaging. It is not intended to be exhaustive, but is intended to provide a robust list that establishes the broad scope of AI so that:

- The scope of subsequent IHE Profiles can be deliberately considered and decided.
- Use cases, mechanics, and entities in later sections of this White Paper can be vetted to confirm their applicability to this range of applications.

Many AI Applications involve performing a task that would otherwise be performed by a human. AI Applications may fully automate the task, automate the task but the result is subject to oversight by a human, or semi-automate the task where the AI Application and the human collaborate on the task.

Benefits of using AI can include one or more of making the task faster, easier, less costly, more accurate, or more consistent/repeatable. These benefits can also be across individual users or large clinical groups; AI brings uniformity to derived physiologic and anatomic measurements across a potentially large number of human performers for a given use case. Without it, individuals are left all doing their own thing. AI makes it easier for a group of people to collectively do the right thing easily.

2.1.1 Ordering and Scheduling

Ordering and Scheduling refers to the process of selecting and ordering imaging procedures and scheduling resources (scanners, rooms, staff, the patient, other supplies, or devices, etc.) for the execution of those procedures.

Tasks to which AI could be applied include:

- Ranking appropriate procedures to order
    - May be based on the patient history, patient location, historical patterns of the ordering physician, similar patient population outcomes, national guidelines, user free text.
    - See IHE Clinical Decision Support Order Appropriateness Tracking [STD-CDS-OAT]
- Scheduling and prioritizing scanners, rooms, staff, etc. for ordered procedures.
- Participating in chatbot patient or staff customer service interactions.
- Suggesting the right imaging clinic/physical location.
- Predicting and mitigating "no shows"
- Identifying that a patient is due (or overdue) to have a particular imaging study (or other procedure) ordered.
    - May be based on guidelines and patient specific details. E.g.,
        - Due for a screening mammogram based on age and date of last exam.
        - Due for a screening lung CT based on patient risk factors.
- Prompting staff to request follow-up exams recommended in reports. [STD-SDC]
- Proposing appropriate values for procedure codes, body part codes, billing codes, etc. and cross-matching with appropriate ontologies for fully populating metadata

2.1.2 Procedure Guidance

Procedure Guidance refers to interventional imaging procedures and to the use of imaging for the purpose of guiding a non-imaging procedure. Imaging for diagnostic purposes is covered in later sections.

Tasks to which AI could be applied include:

- Detecting that a device such as a breathing tube has been positioned (or remains) in the proper location and informing clinicians.
- Providing guidance to the direction of a clinical approach during a minimally invasive interventional or open surgical procedure.
- Directing surgeons or interventionalists toward the optimal percutaneous location to begin a procedure, or to identify tissue remaining to be treated.
- Positioning a biopsy needle properly within the target tissue

2.1.3 Image Acquisition

Image Acquisition refers to the acquisition of images and associated data by an imaging modality. It includes interactions with the patient and associated devices by the staff acquiring images and the imaging modality as part of the acquisition process.

Tasks to which AI could be applied include:

- Setting acquisition parameters appropriate to the patient and diagnostic task
- Positioning the patient appropriately for the imaging procedure
    - E.g., ensuring the primary anatomic structure for the requested procedure is properly in view in the acquired images.
- Guiding the acquisition process to get adequate or optimal images.
    - E.g., a suitable ultrasound projection for assessing cardiac function.
- Detecting image metadata details that are unlabeled or mislabeled.
    - E.g., image laterality, body part imaged, etc.
    - This task is intended for labeling that would be done at acquisition time. Other tasks related to segmentation and labelling can be found under Image Analysis.
- Detecting significant differences in patient demographics (ensuring that the correct patient is being imaged)
- Detecting and/or compensating for patient motion or warning the degree of motion is beyond a quality threshold to trigger re-imaging.
- Detecting target cardiac waveforms to trigger relevant image acquisition.
- Optimizing radiation exposure and dose (in balance with image quality)

2.1.4 Image Generation

Image generation refers to the reconstruction of images from acquisition data (e.g., generating CT images from sinograms, or generating digital subtraction angiographic images from planar XA images) and related processing to improve the images, such as removing noise or increasing contrast.

Reconstruction and processing are commonly embedded in image acquisition modalities although it can be performed on separate workstations.

Tasks to which AI could be applied include:

- Generating images from “less complete” data than would otherwise be used, e.g., partial CT rotation, or shorter MR acquisition.
- Generating images from an incomplete acquisition, e.g., discontinued acquisition due to medical event.

- Reducing noise in images
- Increasing contrast or conspicuity of important features
- Decreasing or eliminating artifacts from patient motion, metal objects, air pockets, etc.
- Increasing the resolution of images or clarity of small details
- Assessing/scoring the quality of the images (perhaps relative to the reason for imaging)
- Prioritizing or triaging reconstruction and post-processing worklists

2.1.5 Image Analysis

Image Analysis refers to processing the content of images and associated data to generate new information, such as measurements, observations, and content for the image interpreter. Analysis is commonly performed separate from the capture modality.

The "**target**" of the image analysis might be a device (like a pacemaker, a stent, or central line), a condition (like pneumothorax, or osteopenia), a pathological structure (like a tumor or aneurysm), or an anatomical structure (like a kidney, colon, or blood vessel)

Tasks to which AI could be applied include:

- Detecting the presence of a target
    - The purpose of detection tasks may be to support the reason for the study, to detect less obvious findings or those unrelated to the imaging request, or to detect findings that may influence events prior to interpretation of the study (see 2.1.6 Image Interpretation on prioritizing reading worklists).
- Ruling out the presence of a target
    - Similar to the detection task, but with an emphasis on triaging and excluding medical emergencies at the point of care.
        - The focus on things like the rate of false positives and true negatives may differ between detection tasks and rule-out tasks.
- Locating a target
    - Locations may be in the form of a point, an arrow, a rough outline, etc.
- Segmenting a target
    - Segmentations may be in the form of a bounding contour, probability map, etc.
- Measuring a quantitative characteristic of targets
    - Anatomic measurements could include length, area, volume, gap, angle, RECIST measurement, etc.

  - Physiologic measurements could include ejection fraction, tissue blood perfusion, lesion enhancement characteristics, etc.
- Determining the appropriate "score/grade/code" from a grading or classification system
  - Examples include characterizing a tumor, grading stenosis, grading hip joint osteoarthritis, etc.
- Characterizing a qualitative characteristic of a target
  - Characterizations may include shape, spiculation, heterogeneity, or density.
  - E.g., breast density reporting.
- Assessing changes between a current and prior study, across same or different modalities.
- Spatially registering studies with current / prior study across same or different modalities.
- Autogenerating teaching files.
  - E.g., "the typical COVID x-ray looks like this."

2.1.6 Image Interpretation

Image interpretation refers to supporting the radiologist in the review of the content of an imaging study, and associated data, to isolate relevant findings, impressions, and follow-up management recommendations, and capture those in a diagnostic report.

Tasks to which AI could be applied include:

- Prioritizing or triaging images to be viewed and interpreted, such as on reading worklists.
- Composing a summary of the medical history of the patient that includes details most relevant to the requesting physician and their current task.
- Selecting prior images from the patient’s record that are most relevant to the reading of a current study.
- Selecting and optimizing the display protocols for display of the imaging dataset under review
- Extracting critical results from reports to drive notification & additional findings for follow-up workflows.
- Flagging lesions for review during image interpretation, or for quality assurance after human reading to identify any relevant additional findings.
- Detecting discrepancies between the dictated report body and the AI-detected observations from images (e.g., potential auto-detected findings), or known information about the patient (e.g., left-right discrepancies between report and procedure performed or a report that mentions ovaries for a male patient).
- Suggesting the diagnosis or relevant differential diagnosis

- Predicting the progression of a condition or the likely outcome of an intervention
- Distinguishing “meaningful” findings (whether part of the intended reason for exam or not) from less meaningful/significant
    - E.g., a benign mass unrelated to the reason for exam, otherwise known as an “incidentaloma.”
- Recommending appropriate clinical follow-up steps
- Recommending appropriate patient staging for disease progression
- Identifying or preparing an effective presentation ("best" slice, useful cut planes, optimized windowing) to understand a given finding.
- Creating draft sentences or sections of reports based on imaging inputs.
    - May provide material for findings, impressions, diagnosis, and follow-up recommendations.
- Incorporating Image Analysis results (see 2.1.5 Image Analysis) into the report and/or medical record in coded and/or narrative form.
- Transcribing and redistributing natural language dictation into structured terms and formatted reports using natural language processing (NLP)
- Determining appropriate people to alert of specific results.
- Identifying appropriate billing codes for performed procedures.
- Recognizing components of the report that may be missing to bill for a given code.

2.1.7 Patient Management and Treatment Planning

Patient Management refers to activities leading to care decisions for individual patients.

See also “Recommending appropriate clinical follow-up steps” in 2.1.6 Image Interpretation. These use cases might also relate to 2.1.1 Ordering and Scheduling.

Treatment plans might include surgery, interventional radiology procedures, radiation therapy, chemotherapy, referrals to other specialists, and other patient management.

Tasks to which AI could be applied include:

- Proposing care or treatment plans to a human, concordant with relevant guidelines.
    - A care plan may consist of appropriate assessments and diagnostic tests being conducted at specific intervals, appropriate interventions driven from those assessments, and appropriate follow-up steps.
- Assessing ongoing conformance of a human generated treatment plan with relevant guidelines, and whether completed treatment was in conformance with relevant guidelines.

- Performing patient risk stratification
- Alerting personnel to (potentially) critical patient conditions such as pneumothorax, brain bleed or stroke.

2.1.8 Population Health

Population Health refers to activities related to the assessment of an aggregate patient population.

Tasks to which AI could be applied include:

- Selecting relevant patients and data from their records (filtering for a patient cohort)
    - E.g., for clinical trials, or for training AI Models
- Identifying trends or patterns for certain pathologies and co-morbidities
    - E.g., “Tuberculosis is up sharply in the last two weeks", or "The prevalence of thyroid cancer in smokers in the north end of the city has been increasingly trending above the state levels over the last 5 years."
- Identifying trends or patterns for certain treatments and outcomes
    - E.g., “Patients with COVID respond better when they get drug X.”
- Determining efficacy and adverse events for drugs or other treatments
- Monitoring ongoing data to identify deviations from trends of data (e.g., outbreaks)
- De-identifying imaging content for inclusion in aggregate datasets
    - E.g., using NLP to identify patient identifying details in report body prose or private DICOM tags, comment fields, image overlays, etc.

2.1.9 Departmental Analysis

Departmental Analysis refers to activities related to the assessment of the operational performance of a department such as radiology.

Tasks to which AI has been, or could be, applied include:

- Computing and tracking clinical performance indicators (e.g., re-admission rates)
- Identifying trends or patterns in throughput or resource utilization
    - E.g., scanner duty cycle, report turnaround time (TAT), exam length, etc.
- Assessing the utilization and effectiveness of clinical tools such as AI Applications
- Assessing the performance of individual imaging equipment to proactively monitor against machine breakdown.
- Supporting departmental analysis tasks conducted in a real-time, daily, quarterly, or annual basis.

- Analyzing information access patterns to protect systems from cyber-attack or unapproved data sharing.

2.2 Personas, Systems, and Entities

This section introduces the people (“personas”), technical components (“systems”), and digital entities (“entities”) that will be involved in the Use Cases described in the following section.

2.2.1 Personas

Personas describe the role, responsibility, or primary interest of a person in the context of a use case and provide readers with insights into motivations and challenges that may be relevant to addressing the use case. Identifying personas up-front enables use case developers a critical look at all layers of the interoperability boundary.

The personas listed below are not intended to imply a specific job title. In practice, a wide variety of people might be involved in the use cases in chapter 3, and those people might take on various roles in different use cases. These different personas might also work together in groups to collectively complete tasks identified in use cases.

Clinical

Radiologist: Provides differential diagnoses on imaging studies and perform image-guided therapeutic procedures. Increasingly asked to integrate AI into image interpretation tasks. Note: Use cases that reference “Radiologist” are often also applicable to other clinicians who use imaging.

Radiology Leadership: Governs the overall operation of a radiology department, including assessing tools to aid the radiologist team. Responsible for authorizing capital and operational funds for AI software and hardware.

Clinician: Provides care to a patient and may interact directly or indirectly with imaging AI applications or the result they produce.

Health Information Management Team: Manages medical imaging record quality by ensuring that records are complete and free of discrepancies, using software tools potentially augmented by AI to aid them in this work.

AI Lifecycle

Principal Investigator: Responsible for leading research projects. Determines research project goals and plans to attain them.

Data Scientist*:* Responsible for conducting AI research and supporting the design the scope and parameters of imaging AI initiatives to craft models from transformed imaging data. Responsible for assessing models for inclusion into clinical practice. May be clinically or technically trained.

Biostatistician: Supports the data scientist by evaluating that the breadth and depth of datasets statistically support their intended use, e.g., to establish a model's claim of efficacy and appropriateness.

Annotator: Provides data record annotations such as ground truth (I.e., correct results for tasks) to be used for training, testing, and validating AI Models.

AI Oversight Committee: A group of clinical users, AI lifecycle users, and hospital operations users that oversees AI safety, vets new products, and prioritizes projects.

Hospital Operations

ML Ops Analyst: Monitors AI Application operations and clinical and technical performance over time. Monitors for updates that may impact AI Applications and participates in go-live activities (from development environments to pre-production environments to production environments). Monitors for operational QA and tests models running in production.

Repository Administrator: Maintains the collection of all imaging data and related healthcare metadata and supports data scientists in supplying study cohorts with data.

Informaticist: Administers the clinical imaging systems, their data, and workflows, and assessing tools that get integrated into workflows. This could be, for example, a PACS Administrator or a Vice-Chair of Informatics.

IT Operations: Maintains the systems that run the radiology department, including test, development, pre-production, and production environments, and works with informaticists and data scientists on deploying software for evaluation.

Information Security and Privacy Officer: Responsible for data security operations and architecture, preventing data loss and fraud, and identity and access management.

Product Vendors

AI Application Vendor: Provides software that takes imaging studies and provides insights (in the form of classification, measurement, segmentation, severity, etc.).

Vendor Product Manager: Determines the features necessary for medical software applications.

Vendor Validator: Takes software developed by engineers and requirements from product managers and determines whether it is performing as expected.

Product Support Team: Supports deployments of systems that run the radiology department, including test, development, pre-production, and production environments, and works with their customers (e.g., their customers' informaticists and data scientists) on deploying software for evaluation.

Governance

Health Policy Agency: Oversees how healthcare is delivered in a region and may provide guidance on how to incorporate imaging and AI Applications and provide oversight in how it is used.

Payor: Responsible for renumerating the hospital for the costs incurred for a radiology exam and collecting premiums from patients that they serve.

System Regulator: Provides approval for market claims when selling a product in a particular jurisdiction, by assessing claims and supporting data.

2.2.2 Systems

Systems describe the role, responsibility, or primary interest of a piece of hardware or software a person in the context of a use case. This is important as it helps to identify the interoperability boundaries at the technical level.

The systems listed below are not all-inclusive and may not be reflective of all possible configurations or functionality. In practice, for example, the role of a PACS might be fulfilled by a VNA, a viewer and workflow software integrated together to collectively complete tasks identified in by the system.

PACS: Picture Archive and Communication System. Responsible for viewing, storing, and workflow of medical imaging. System may be converged or deconstructed.

EMR: Electronic Medical Record. Responsible for health discrete data, such as patient demographics, allergies, diseases, and reports.

AI Application: A collection of components that include an AI Model but also includes the necessary interoperability components, pre-inference transforms, and post-inference result creation.

VNA (Vendor Neutral Archive): An archive of imaging studies and related data.

AI Application Repository: An archive of AI Models or AI Applications that facilitates discovery, retrieval, and deployment of AI Applications.

Image Display: A viewer for visualizing medical images along with results from AI processing.

AI Application Host: A dedicated system that runs specific AI Models or AI Applications.

Imaging Modality: An image acquisition system generating patient images that may be analyzed by AI. (used in Section 3.4.4)

2.2.3 Entities

Common entities that appear in use cases are described in Section 4 Entities, which also captures initial ideas about what information would go into a digital encoding of those entities.

The definitions from Section 4 are replicated here to clarify the intended use of the entities in the use cases.

Data Element: An individual piece of data. The granularity may vary.
E.g., an individual image (DICOM or PNG), the patient age, a reason for admission, an annotation (RTSTRUCT or Condition Present) or a medication, a lab value.

Data Record: A set of related data elements. The relationship is often that they are all associated with the same patient encounter, but this can vary.
E.g., an x-ray, a lab result, and a reason for admission, all associated with a given patient encounter.

Dataset: A set of related data records. Typically, all the data records in a dataset contain roughly the same data elements. Datasets containing data records with patient data will typically span many patients.

Data Repository: An infrastructure that hosts one or more datasets for discovery and retrieval.

Transform: A process that produces one or more output data elements from one or more input data elements, typically by transcoding or resampling.

AI Model: A package of model weights that has been trained to identify a particular target in a supplied input.

AI Application: A package of components to make an entire application, which could include model weights, algorithms, transforms, and input/output operators.

Model orchestration framework: An environment with deep learning infrastructure and appropriate compute power, to run AI Models cloud-hosted or on-premises.

Service: A running instantiation of an AI Application.

Feedback: An assessment by a human or system of the output produced by an AI Model performed on a given data record.

3 Use Cases

This section describes activities (use cases) involved in the development and use of AI in Imaging for purposes such as those described in Section 2. The activities described here span from research and development, through deployment to after-market monitoring and maintenance.

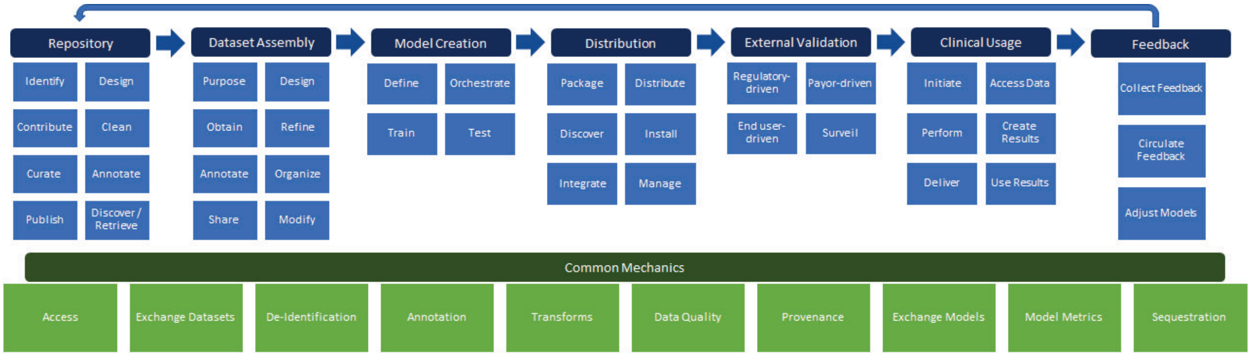

Figure 3-1: Map of AI Use Cases

Each Use Case describes a particular activity (such as Contributing Data to a Repository, Annotating Data, or creating cohorts of data from existing repository elements) and notes important questions, topics, and considerations that might need to be addressed in a Profile which includes that Use Case.

The Use Cases are grouped into significant stages, such as repository use cases that cover all these use cases relating to preparation and management of data, and training and validation use cases that cover the process of training and validating AI Models.

The Use Case Groups and the Use Cases within them are presented as a sequence, however, as with the Waterfall Model [BIB3-1]; there is significant feedback and iteration both within Use Case Groups and between Use Case Groups.

Many use cases involve taking data elements, data records, and/or datasets (see Personas and Systems: Entities), using them, and often creating new or revised data elements, data records, or datasets. To distinguish between these new entities and their source entity, the descriptions will typically prefix a distinguishing adjective that describes the purpose or origin of the entity, e.g., a *contributed* data record, a *repository* dataset, a *validation* dataset, or a *training* dataset.

Some examples of how these entities are manipulated in the use cases include:

- Populating a repository may involve taking a contributed data record, then normalizing and validating several the data elements, removing or de-identifying several other data elements, and adding the resulting data record to the repository dataset.

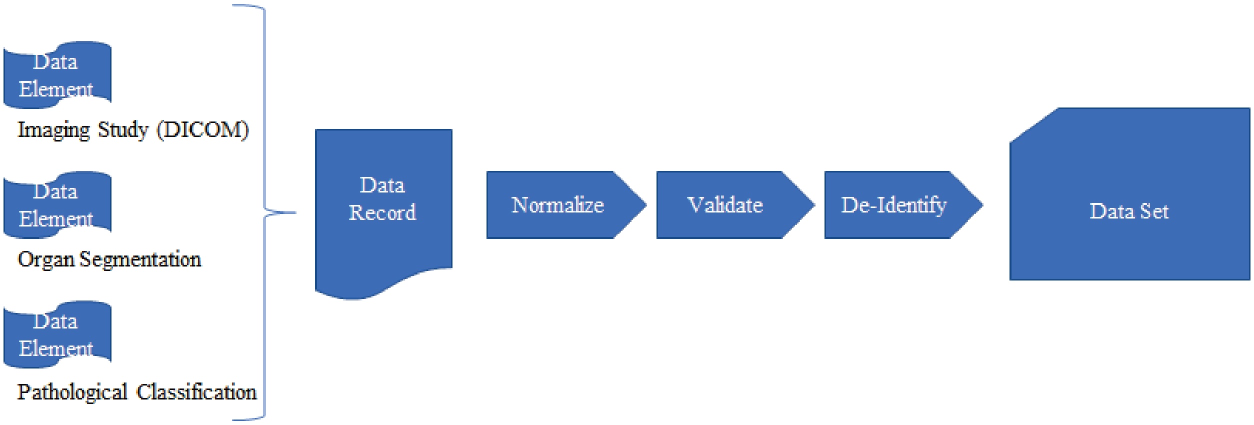

Figure 3-2: Aggregating Data Elements into Data Records and Datasets

- Creating a training **dataset** might involve taking two repository **datasets**, selecting certain **data records** based on inclusion/exclusion criteria, dropping several **data elements** as irrelevant to the planned model purpose, and storing the resulting training **data records** as a single training **dataset**.
- Performing validation might involve taking a **data record** from a validation **dataset**, setting aside the **data elements** that contain the result, presenting the rest of the **data elements** to an AI Model, and then comparing the output of the AI Model to the result **data elements**.
- An annotation task might involve taking a **data record** and adding an annotation **data element** to that **data record**. This is often done for all the **data records** in a **dataset**. This is commonly done when preparing training **data records** and validation **data records**.
- An inference task involves presenting one prepared inference **data record** as an input and producing one (or more) **data elements** as an output/result.

The following scenarios are intended to provide a sense of the overall process. They do not delve into all relevant details.

Scenario 1:

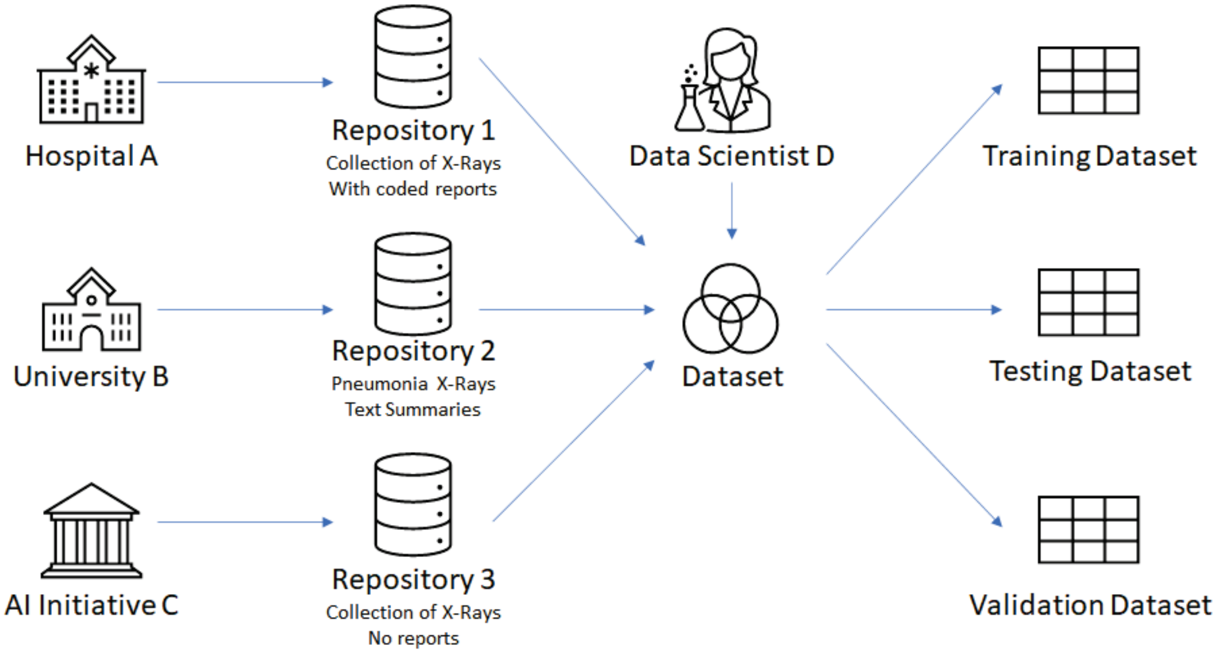

Figure 3-3: Example Scenario: Assembling Datasets from Repositories

Hospital A has created Repository Dataset 1 based on their collection of chest x-rays. Their clinical practice is to create fully coded structured reports. The Repository Administrator has arranged for each data record to contain the chest radiograph image, data elements for each of 6 key demographic details (sex, approximate age, etc.) and each individual finding, impression and recommendation in the report have been broken out from the coded report as separate data elements in the data record. Each record has been de-identified to obscure the patient identity and the dataset has been published in the public repository operated by the research arm of the hospital.

University Research Group B has created a similar Repository Dataset 2 which was intentionally rich in positive pneumonia cases, however the report is included as paragraph formatted text in a single data element.

AI Initiative C has created a Repository Dataset 3, with chest radiographs but no reports.

Data Scientist D plans to train an AI model to detect the presence of pneumonia in chest radiographs images. In designing the primary dataset, Dr. D requires each data record to contain input data elements for the image, the patient sex, the approximate patient age, and an output data element for the presence or absence of pneumonia.

Dr. D discovers the Dataset 1, 2, & 3 and obtains all the data records. For all the data records in Dataset 1, the required data elements are retained, and the rest are discarded. For the data records in Dataset 2, the required input data elements are retained, and the text report is passed through a

Natural Language Processor to extract an appropriate pneumonia value for the output data element. For the data records in Dataset 3, the image headers are automatically parsed to populate the sex and approximate age input data elements, and a radiologist is recruited to perform an annotation task to populate the output data element for each record.

Realizing that the model of x-ray device and several key imaging parameters might be important, Dr. D runs all the records through an automatic parser to extract the model and several imaging parameters from the image header into additional input data elements.

Dr. D organizes the primary data records into secondary datasets for training, testing and validation, maintaining a balance of records derived from each of the 3 repository datasets and balancing demographics, rates of presence of pneumonia, and imaging device/protocol variants.

Scenario 2:

Data Scientist E plans to train an AI model to propose report recommendations for chest radiographs at the end of interpretation based on the findings and impressions. Ms. E designs the primary dataset data records to have input data elements containing findings and impressions and a broad set of patient demographics and output data elements.

The data records from Repository Dataset 1 can be used with little refinement, discarding the images, assigning the finding and impression data elements to be input data elements, and the recommendations as output data elements. Like Scenario 1, Ms. E uses a Natural Language Processor to extract input and output data elements from the reports in Repository Dataset 2. As a refinement step, Ms. E has a graduate student compare the extracted data elements to the Dataset 2 narrative text and fix any observed errors.

Since aside from a few demographics in the image headers, Dataset 3 lacks most of the key inputs and outputs, and resources are not available to do full reporting on all those images, Dataset 3 is not incorporated at this time.

Ms. E is, however, able to discover a repository with a dataset of radiology reports (but no images). The data records from this Repository Dataset 4 can be processed and incorporated into Ms. E's master dataset in a similar fashion to the data records from Repository Dataset 2.

Organizing the primary data records into secondary training, testing and validation datasets is like Scenario 1.

3.1 Repository Use Cases

Repository Use Cases encompass the creation and management of repository datasets.

This use case group is focused on repositories that hold datasets. Typically, repository datasets are established to serve as the basis for datasets used to train, test, and validate AI Models. To avoid confusion, later use case groups that might involve repositories of models or applications will avoid using the term "repository", and instead refer to model zoos, or application libraries.

3.1.1 Identify the Repository Purpose

A Repository Administrator establishes the purpose(s) a repository is intended to serve and the corresponding scope of data it is intended to contain.

While some repositories may be created with the development of a specific model in mind (e.g., a repository of chest x-rays for training algorithms to detect the presence of lung cancer), most repository use cases are model-independent (e.g., a repository of imaging, labs, reports, and clinical notes for patients admitted with chest pain) or at least applicable to a broad variety of questions.

The purpose details will influence decisions in many of the subsequent use cases in this section.

Considerations that may have technical interoperability aspects include:

- Encoding the purpose
    - Encoding the range of intended purposes of the repository in a way that will facilitate the Discover Dataset use case (noting that some data assembly will go beyond the original intended purpose of the repository and that determining suitability is the job of the model Data Scientist not the Repository Administrator).
- Model-independent repositories can be made with or without the guidance of clinical expertise when determining scope (e.g., inclusion and exclusion criteria) as well as the data to be included.

Profile writers may consider some of these questions for further discussion:

- How do we account for the repository purpose shifting over time?
- If you have two different repository purposes (being met by one infrastructure), is that one repository or two?
- How to locate and engage data sources for whom the intended purpose is compelling.

3.1.2 Design Repository

A Repository Administrator designs the data structures, rules, policies, and necessary infrastructure to host the datasets.

Creating a repository involves setting up a technical infrastructure and designing the databases and schemas that will store the dataset(s). Many of these choices will be driven by the intended purpose(s) and scope of the repository and by practical constraints of the infrastructure.

In some cases, an existing infrastructure (e.g., the partial content of a VNA) is being exposed as a repository. In that case, this step is partly about documenting what exists, since there may be limits on how the existing data records can be modified.

The other use cases in this group include considerations that will affect the repository design. Rather than reiterate those here, the reader is encouraged to review the rest of the Repository Use Cases with this in mind.

Considerations that may have technical interoperability aspects include:

- Defining the specific data elements expected or required in each data record.
- Inclusion and exclusion criteria for data records
  - E.g., specific populations, conditions, modalities
  - Being pre-de-identified may be criteria for inclusion or exclusion.
  - These criteria are sometimes referred to as "eligibility criteria."
  - The Design Dataset use case also addresses inclusion/exclusion criteria; however, Repository criteria often differ due to differing purposes which may be broader or narrower.
- Encoding format(s) of each data element
- Standardized code sets or common terminologies to be used.
- Whether data element storage is centralized or distributed
  - Centralized storage makes it easier to normalize access protocols and mechanisms.
  - Distributed storage allows data to be left in their source systems.
- Whether data is separate from production clinical systems
  - This prevents having an impact on clinical system safety and performance, but also requires more storage, and may necessitate extra annotations (clinical and research annotations).
  - Needs to avoid impacting patient care.
- The degree to which data is de-identified.
  - A privacy officer may ascertain that the repository needs to be de-identified, which may influence how the dataset will be structured (e.g., how can related records be captured).
  - Certain data purposes may require certain potentially identifying details to be retained (e.g., approximate dates of datasets, anonymous patient identification to link data from the same patient)
- Data provenance mechanisms (see Common Mechanics: Provenance)
- Providing information to, and performing, an Institutional Review Board (IRB) review
- Access control
- Patient consent
  - Whether consent prevents use of de-identified data varies between jurisdictions.

Profile writers may consider some of these questions for further discussion:

- Sharing a common design (data elements, inclusion criteria, etc.) across different repositories facilitates federating their datasets.
    - Broad/federated datasets are an important method to diversify the dataset and avoid potential sources of bias (scanner model, protocols, patient demographics, annotation bias, etc.)
- Should this be generalized to be another example of data assembly?

3.1.3 Contribute Data

A Repository Contributor contributes data records and/or data elements to a repository.

The first "contributor" may be the Repository Administrator harvesting an initial set of data records to which they have obtained access as part of planning for the repository in the first place.

Incorporating the first data records into the repository dataset often serves as an initial test of the repository design and design assumptions so some iteration with the Design Repository use case may be expected.

Repository Contributors may be internal or external to the organization hosting the repository. The same considerations typically need to be addressed in either case, but the onus for addressing them may shift. Considerations that may have technical interoperability aspects include:

- Obtaining and tracking Consent of patients who are the subject of a contributed data record.
- Confirming compliance with any IRB approval conditions
- Reviewing any licensing or usage constraints on the contributed data records
    - E.g., identified data records may be restricted from being re-released publicly.
- Tracking the provenance of each data element (who contributed it, how it was created, when it was created, if was clinically reviewed and vetted, whether it was derived, etc.)
- The Repository Administrator may wish to constrain who is permitted to contribute.
    - The Repository Contributor may be outside the Repository Administrator organization, and may be an unknown entity, subject to security and inclusion/exclusion concerns.
    - Contributed data records may be unsolicited and/or of unknown provenance.
- See Common Mechanics: Data Access for mechanisms to obtain data element values.
    - (For distributed repositories) Mechanisms to convey references to data, such as URIs, and associated metadata to support indexing.

- Data is being pushed rather than pulled.

Profile writers may consider some of these questions for further discussion:

-

3.1.4 Clean Data Record

A Repository Administrator processes contributed data records and data elements before they are formally incorporated into repository dataset(s).

Considerations that may have technical interoperability aspects include:

- Cleansing and normalizing the data record
    - Fixing up problematic DICOM tags
    - Scaling the resolution into a standard size used for all studies (see Entities: Transform)
    - Normalizing the color sets and pixel values.
    - Coercing values into standard code sets and common terminologies.
        - E.g., RadLex, RadElement, LOINC, SNOMED CT®[4]
    - Normalizing DICOM Study Descriptions, Anatomy, Procedure Codes, etc.
    - Confirming required/minimum elements are present (and handle exceptions by fixing, flagging, or dropping)
    - Confirming inclusion and exclusion criteria have been met (and handle exceptions)
    - Breaking down clinical data into per encounter representations. (clinical data may need to be presented per data element and not as a complete dataset depending on use case)
        - E.g., presented as SAS extracts, R, BIDS, FHIR®[5] bundles, or bulk spreadsheets.
    - De-identifying data as needed depending on the design and policies of the repository (see Common Mechanics: De-identification)
- Augmenting data records by deriving new data elements

[4] Some IHE Profiles incorporate SNOMED® CT, which is used by permission of the International Health Terminology Standards Development Organisation. SNOMED CT© was originally created by the College of American Pathologists. SNOMED CT is a registered trademark of the International Health Terminology Standards Development Organisation, all rights reserved

[5] FHIR is the registered trademark of Health Level Seven International and the use of this trademark does not constitute an endorsement by HL7.

  - Using NLP to analyze reports and expose findings or observations as coded data elements or FHIR Observation [STD-FHIR-OBS] resources.
    - Augmented data may be perfectly accurate or mostly accurate (e.g., weakly label imaging studies)
  - Using data transform tooling to prepare normalized versions of the data to expedite usage of the data.
    - Creating NumPy arrays for pixel data

Profile writers may consider some of these questions for further discussion:

-

3.1.5 Curate Repository

A Repository Administrator reviews the dataset(s) within a repository to ensure that its contents are appropriate for the repository's stated purpose.

Considerations that may have technical interoperability aspects include:

- Quarantining new datasets/data records to prevent their use until necessary curation has been completed.
- Sequestering specific data records to constrain access to them in accordance with a sequestration strategy (see Common Mechanics: Sequestration Strategies)
- Detecting and managing duplication of data
  - Anonymizing data makes this more challenging.
  - Data identification and provenance may help with this. See Common Mechanics: Provenance.
  - Supporting users of the repository who will be trying to avoid duplication of data records across multiple repositories.
- Confirming appropriate consent, depending on the policies of the repository, the conditions of use of the source data, and the ability to identify the subject of the source record.
- Assessing the balance/bias of the dataset
  - Identifying the criteria for when obtaining additional data records is appropriate.
  - See Common Mechanics: Data Quality.
- Confirming the contents of the repository is appropriate for the stated purpose.
  - Assessing data for representation of edge cases and balancing of difficult cases (or cases with confounding factors).

  - Assessing data records for inclusion of relevant clinical details, as determined by clinical subject matter experts.

Profile writers may consider some of these questions for further discussion:

-

3.1.6 Annotate Repository Data

A Repository Contributor augments data records with new annotation data elements.

Typically, the annotation data elements are observations that an AI Model would be expected to produce given the corresponding data record as an input. Since the repository may serve a variety of AI Models, the annotations might not be of use to all users of the repository, but often the Repository Administrator will have a good sense of the kinds of annotations that may be in demand. An advantage of adding annotation data elements in the repository dataset is that it may avoid many users of the dataset having to duplicate the annotation work. It is also possible that the repository may be able to bring more resources to bear to do a better job than some users. In a sense, this is a refinement of the Curate Repository use case.

Considerations that may have technical interoperability aspects include:

- See Common Mechanics: Annotation.
- Transforming annotation data elements to match the Repository Design (see Entities: Transform)

Profile writers may consider some of these questions for further discussion:

- Should repositories establish quality criteria and/or credential checking for annotation data elements contributed to the repository?

3.1.7 Publish Repository

A Repository Administrator exposes the existence of the repository along with descriptive metadata and prepares the dataset(s) for access.

While some repositories may be published with a limited intended use, others will be broader, and applicable to a great number of AI Model applications. Repositories may serve AI goals beyond those envisioned by the administrator. The emphasis in this use case should be on documenting the nature of the repository content and leave it to clients to assess the applicability to their model/goal.

See Data Assembly: Discover Dataset (which is the primary user of the metadata described here) and Repository: Obtain Repository Content (which is the primary user of the prepared dataset).

Considerations that may have technical interoperability aspects include:

- Encoding metadata used in the repository, including:

  - The intended purpose of the repository (both what type of application; see 2.1: Applications of AI in Imaging) and perhaps whether it is appropriate for training, validation, etc.
  - Descriptors of the size and scope of the repository
  - Inclusion and exclusion criteria used to select data records for inclusion in the dataset.
    - Informs analyses of potential bias in the dataset.
    - Including clinical considerations behind the criteria might help related decisions by model developers using the dataset.
  - The core set of common data elements in each data record.
  - Endpoints for the discovery and retrieval use cases.
  - Terms of use, licensing and/or limitations on who may access the data for what purposes.
    - This may get into sequestration strategy topics like whether the data is permitted to be shared or to be used for training.
    - Any IRB restrictions on the use of any data elements must be made clear.
- Communicating metadata about the repository
  - This covers the publication mechanisms and top-level metadata. Data record level metadata is covered in Discover Repository Content.

Profile writers may consider some of these questions for further discussion:

- Where might the “existence” notification be sent/published to? How?
  - Where is the metadata published to or is it retrieved from the repository itself as part of initial interactions by a client?
- Determine the details to be included in the “publication” (this repository exists) versus which details are handled as part of the “discovery and retrieval” during data assembly.
  - I.e., sort of like the difference between what you get in a CFIND vs the full header returned when you retrieve an object.
- Some private or federated repositories may choose to only publish specific data elements and inclusion/exclusion, or what is published may be dependent on the authorized user or group.
  - Only the results are provided back to the developer to preserve data elements or eliminate the need for de-identification.

3.1.8 Discover Repository Content

A Repository Administrator prepares mechanisms to support discovery of repository data records by Data Scientists.

This use case is paired with the Data Assembly: Obtain Dataset use case. It exists here in the Repository Use Cases group because to a large degree it is the repository administrator that controls the metadata that will be made available for discovery and the transfer protocols supported for retrieval.

Considerations that may have technical interoperability aspects include:

- Querying may include de-identification (see Common Mechanics: De-Identification).
- Query based on presence or absence of specific data elements.
    - The Data Scientist may wish to confirm the presence of input data elements that are essential to their model or confirm that their model's output data elements (I.e., ground truth) are present so they will not have to do annotation to generate them locally.
- Query based on the values of specific data elements.
    - The Data Scientist may need data records that support certain diversity targets or that fit certain criteria of their model.
    - Some data elements that might commonly be the basis for such criteria include patient age, gender, imaging modality, device model/version, body part imaged, performed procedure, imaging protocol, contrast usage, and diagnosis (possibly derived from a codeset like RadLex or Gamuts)
- Access controls and security of the data need to be addressed to ensure that sequestered or quarantined data elements are not made available to users by the repository's interfaces.

Profile writers may consider some of these questions for further discussion:

- What existing query standards may be appropriate?

3.1.9 Retrieve Repository Content

A Repository Administrator prepares mechanisms to support retrieval of repository data records by Data Scientists.

This use case is paired with the Data Assembly: Obtain Dataset use case. It exists here in the Repository Use Cases group because to a large degree it is the repository administrator that controls the transfer protocols supported for retrieval.

Considerations that may have technical interoperability aspects include:

- Bulk retrieval versus fine-grained retrieval

  - A repository may support providing data either as singular requests for specific data records (e.g., through providing an index of records with URIs to retrieve further data), or through bulk retrieval (as a ZIP file or data volume). These mechanisms likely rely on a query/discovery mechanism.
  - See Common Mechanics: Data Access.
  - Bulk retrieval of clinical data might put undue load onto clinical systems.
- Location and latency of retrieval of data
- Retrieving may include de-identification on-the-fly (see Common Mechanics: De-Identification).
- Authentication, authorization, and access control.

Profile writers may consider some of these questions for further discussion:

-

3.2 Dataset Assembly Use Cases

Dataset Assembly Use Cases encompass the collection of data records to assemble a specific dataset to be used for the training, testing, or validation of a particular AI Model.

3.2.1 Identify Dataset Purpose

A data scientist defines the specific task to be performed, the specific question to be answered, or the specific piece of information to be produced by an AI Model.

The purpose will drive how Datasets are selected and assembled into cohorts, and a “cohort of training datasets” are derived (e.g., a subset of a repository or federated repositories that are appropriate for the question being asked). Radiology leadership teams and AI oversight committees may also provide insight to defining specific tasks to be performed.

Considerations that may have technical interoperability aspects include:

- Providing an identity for a model makes it more easily transferable, so that all actors understand how the model is to be used, for what data elements it can provide inference, and under what circumstances. Model identification includes global uniqueness, versioning, provenance, training history, and the model owner.
- The type of datasets needed depend on the type of AI Model that is to be created; whether it is to identify a condition (classification), a particular region of the body, or identify a region of interest (detection or segmentation). These dataset elements have specific characteristics that are aided through interoperability.
- Mapping expected data elements in the dataset to AI Model inference inputs and the expected result.

Profile writers may consider some of these questions for further discussion:

- Given the model purpose, what sorts of factors might constitute bias in the training and validation datasets?
    - E.g., patient age, race, insurance status, socio-economic status, scanner type, scanner protocol, image quality, etc.
- How is the intent of the model being captured?
- What sort of metadata is captured when a model is instantiated?
- What are the fundamental outputs?
- Do we need to define the type of data that we want to get feedback on?
- What are the fundamental inputs?

3.2.2 Design Dataset

A data scientist defines the dataset(s) they wish to create.

As part of this process, the data scientist provisions the dataset, and assigns the notable metadata to the dataset. Creating robust generalizable models requires enough data for the problem at hand.

Considerations that may have technical interoperability aspects include:

- Defining the specific data elements expected or required in each data record.
- Inclusion and exclusion criteria for data records
    - Modality type (e.g., includes CT but not MR)
    - Anatomic coverage of imaging (e.g., must include lung apex and base)
    - Image quality scores
    - Presence of adjudicated truth (data to be used for training or validation)
    - Use of image compression and whether it is lossy or lossless.
    - May requires raw pixel data, not for presentation or machine LUT applied.
    - Being pre-de-identified may be criteria for inclusion or exclusion.
- Encoding format(s) of each data element
- Standardized code sets or common terminologies to be used.
- Whether data element storage is centralized or distributed
    - Centralized storage makes it easier to normalize access protocols and mechanisms.
    - Centralized storage may make data element changes such as anonymization easier to manage.

  - Distributed storage allows data to be left in their source systems.
  - Data elements in distributed storage may have their own governance policies and changes will need to be communicated to the repository.
- Whether data is separate from production clinical systems
  - Separation avoids impacting patient care (safety and performance).
  - Accessing production clinical systems directly may reduce storage and avoid extra annotations (clinical and research annotations).
- The degree to which data is de-identified.
  - A privacy officer may ascertain that the dataset needs to be de-identified, which may influence how the dataset will be structured (e.g., how can related records be captured).
  - Certain data purposes may require certain potentially identifying details to be retained (e.g., approximate dates of datasets, anonymous patient identification to link data from the same patient)
- Data provenance mechanisms (see Common Mechanics: Provenance)
- Providing information to, and performing, an Institutional Review Board (IRB) review
- Patient consent
- Accounting for bias and statistical relevance
  - Datasets need enough data records to ensure bias is accounted for and statistical relevance is present within the dataset for its purpose. Data records that represent situations that would be rarely encountered by the model should be carefully distributed.
  - Identifying potential sources of bias in the dataset that could bias performance of the intended task.
  - Consider procedures that will be executed during Obtain Data Records, Refine Data Records and Organize Datasets to ensure that datasets are free of sources of bias and that enable generalizability.

As an example, the data scientist designs a dataset to train a model to classify pneumothorax. They target collecting 100 data records containing a chest radiography study. Their acceptance criteria include that the modality is digital radiography, the body part is chest. The dataset will be collected and used internally so the data records will not be de-identified. The data records will contain an output data element for pneumothorax present/pneumothorax absent.

The data scientist considers potential sources of bias in their inclusion and exclusion criteria and plans to assess areas the dataset is not covering appropriately (for example, skewing toward a subset of their patient population.

Profile writers may consider some of these questions for further discussion:

- What kinds of data elements do we expect in data records?
    - Images, Reports, Annotations, Spatial Registrations, Demographics, Lab Results, other Clinical info, Family History, Admitting Diagnosis, Past Procedures, Current Drugs, …

3.2.3 Obtain Data Record(s)

A data scientist discovers repositories and explores the metadata of the datasets they contain to identify and retrieve data records that would be appropriate for the dataset they are assembling.

This use case interacts directly with several Repository Use Cases. See Repository Use Cases: Publish Repository, Discover Repository Content, and Retrieve Repository Content.

Comparing the published purpose of a repository to the intended purpose of the dataset is often a useful first step, but note that while the dataset being assembled may have a narrow purpose, like training a model to detect the presence of a certain class of liver tumor, repositories often have broad purposes such as a large collection of chest and abdominal CTs with a wide variety of lesions.

In addition to obtaining records from repositories, a data scientist might search local image archives and EMRs directly for relevant patient data from which to construct data records.

Considerations that may have technical interoperability aspects include:

- Tracking the provenance of obtained data records
    - See Common Mechanic: Provenance
- Selecting specific data records from a repository dataset
    - Selection criteria will be driven by the dataset purpose and design.
    - Some repositories may have search capabilities allowing retrieval of a matching subset of records.
    - Even if a repository does support record metadata queries, retrieving an entire dataset and filtering locally allows greater detail and flexibility of filtering.
    - Presence of existing annotation data elements in a data record may be highly desirable since it avoids the cost of creating those annotations.
        - Conversely, externally produced annotation data elements may or may not meet the accuracy or consistency criteria determined by the data scientist for this dataset.
        - Annotation data elements may sometimes be available from a different repository than the one hosting the rest of the data record.
- Seeking out a diversity of data records to avoid sources of bias.

  - May require obtaining data records from a diversity of healthcare institutions, a diversity of acquisition systems (different makes, models, versions), a diversity of acquisition techniques (protocols), and a diversity of patient populations.

Profile writers may consider some of these questions for further discussion:

- How do they find and access the descriptions and metadata made available in the Publish Repository use case?
- How is consent handled?
- How is data licensing handled?
- What details should the query model include? What would a researcher want to know about data(sets)?

3.2.4 Refine Data Record(s)

A data scientist processes obtained data records and data elements before they are formally incorporated into their own dataset(s).

Considerations that may have technical interoperability aspects include:

- Data records obtained from repositories may include collect many data elements that are not relevant to the dataset purpose and may be dropped to reduce the bulk of the dataset.
  - Removing unneeded Series e.g., remove “with contrast” series and keep “without contrast” series.
- Manually inspecting data records and disqualifying based on quality criteria.
- Transforming data elements to match the Dataset Design (see Entities: Transform)
- See Common Mechanics: De-Identification

Profile writers may consider some of these questions for further discussion:

- Given some of the expected uses of the repository, what sources of bias may be of interest to the model Data Scientist, and thus the repository could help by considering during data harvesting and contribution?
- How does de-identification differ between object types and how are patient record sets correlated? E.g.:
  - Multiple timepoints for same patient
  - Different object types CDA®[6] and DICOM in the same study/timepoint)
- What kind of data gaps or errors might be expected in the data records?

[6] CDA is the registered trademark of Health Level Seven International and the use of this trademark does not constitute an endorsement by HL7.

- Are there means to sub-type the dataset that is meaningful further down in the dataset annotation process? (e.g., contrast or no contrast)

3.2.5 Annotate Data Record(s)

An Annotator augments data records with new annotation data elements.

Typically, annotation data elements are results that an AI Model would be expected to produce given the corresponding data record as an input, also known as “**ground truth**”. Annotation data elements may also be added to conform to data record requirements established during dataset design (see Dataset Assembly Use Cases: Design Dataset).

The ground truth is provided to the AI Model during training to drive the weight adjustments of the training process. The ground truth data elements are held back from the AI Model during testing and validation and are compared against the results produced by the AI Model.

Considerations that may have technical interoperability aspects include:

- See Common Mechanics: Annotation.
- Transforming annotation data elements to match the Dataset Design (see Entities: Transform)

Profile writers may consider some of these questions for further discussion:

- Do annotation data elements received in data records from the repository need to be tweaked or replaced based on the details of this dataset purpose?

3.2.6 Organize Datasets

A data scientist organizes data records in a primary dataset into smaller sets for training, testing, and validation.

Datasets are sub-divided into smaller groups for training models, testing the models as part of the training process, and validating models that have been completed. It is important that these groups contain a balanced representative set of imaging studies, so that at each phase of the model ideation process, similar types of data are being used. See [BIB3.2.6-1]

Considerations that may have technical interoperability aspects include:

- Dividing available data
  - o Examples of sub-divided datasets include training sets, testing sets, validation sets, and assessment sets. See Appendix B: Glossary.
  - o A dataset organization may have a mix of these sets, e.g.:
    - ▪ Training set: 80% of all data
    - ▪ Testing set: 20% of all data
- Sequestering datasets to avoid mixing training, testing and validation datasets.

  - Analyze for records not being duplicated between those datasets.
  - See Common Mechanics: Sequestration
- Balancing sources of data within each dataset grouping.
  - E.g., validation datasets should be obtained from an independent third-party repository with data elements that meet the AI Model's stated criteria.
- Where truth is stored needs to be considered as it is use depends on the use case
- Assess the balance of the data records in the dataset that may represent sources of bias.
  - May require obtaining additional data records (see Dataset Assembly Use Cases: Obtain Data Record(s))
  - Bias and statistical considerations need to be accounted for based on populations and expected occurrence of conditions.
  - E.g., balance data elements based on type and finding evenly between training, testing and validation.

Profile writers may consider some of these questions for further discussion:

- To what extent is data at this stage further sub-divided to sub-type it? For example, balancing datasets that contain imaging data with contrast versus no contrast might be done at this point.
- How much does training, testing and validation datasets get its own "autonomy"?

3.2.7 Share Dataset

A data scientist shares datasets with authorized research partners to enrich their data sources for training purposes.

Sharing a dataset may be done as point-to-point, or submitted to a remote repository, or merged to form a "data lake". Sharing datasets is different than publishing repositories, as it is done much more in a targeted and focused manner.

Considerations that may have technical interoperability aspects include:

- Accessing underlying data
  - A dataset could be federated using XDS [STD-XDS] or similar architectural style of registry/repository.
- Uniquely identifying patient and data records
  - Patient and record identity management should be well understood. Lab results or external exams may have different patient identifiers that may need to be managed. Techniques like PIX [STD-PIX] can be employed.

  - De-identification may involve creating new identifiers that may need to be maintained in other data elements within a data record.
- Delivering data to various types of destinations, e.g.,
  - A destination for this data could be a system in the same hospital, perhaps in a different department, or in a central repository.
  - It could be a different hospital (perhaps in the same organization, such as a satellite hospital sending to an academic medical center).
  - It could be a centralized location operated by a cloud service.
- Updating data records that have changed after data has been delivered.
- Supporting updates to the dataset from remote locations
- Auditing and data security
  - As is with the case whenever data is shared, care must be taken to address network and security components.
  - Audit logs should denote that data was shared with other actors.
  - The foreign system must be reachable on the network.
- Data licensing and distribution rights
  - The dataset author should have the rights to share the data.
  - Appropriate permissions for patient consent need to be addressed.
  - If the dataset contains identifying information, distributing it to other organizations may require de-identification (see Common Mechanics: De-identification)

Profile writers may consider some of these questions for further discussion:

-

3.2.8 Modify Dataset(s)

A research lead manages versions as they are used and amended with updated data.

Repositories will expand / contract / revise over time. A strategy is needed to deal with that appropriately with respect to a model. Reasons for modifications to a dataset may include:

Considerations that may have technical interoperability aspects include:

- Reasons for modifications to a dataset may include:
  - **Data removal:** Patient withdraws consent; Data is no longer on the relevant to use case (e.g., technology is end of life)
  - **Data additions:** Additional data records and/or annotation data elements for existing data records become available; Need for additional data due to underfitting.

  - **Data Corrections (Maintenance):** Mislabeled data; Inconsistent presentation formats

- Dataset Provenance
  - A training dataset is “frozen in time” when a model is instantiated – if a training set is changed, it really needs a new identity.
  - Versioning of a dataset is notable, as it is impacted when data is added, modified, or removed.
  - It may also be important to be able to step back in time to a previous copy of a dataset for the purposes of troubleshooting or correcting when bad data has been added to a dataset (e.g., poor image quality, or incorrect labels).
  - A linear versioning strategy (version 1, version 2, …) is not a concept that could work, as multiple derivatives could be branched at any given time. For example, if Dataset 1 Version 1 is shared between two departments, and both contribute their own data records to that dataset, they have not created two Version 2.
  - It should be that once a version has been “saved” or “published”, that it should remain frozen, and derivatives must have a new identifier. A ledger (like that of what Blockchain uses) might be applicable here.
  - See Common Mechanics: Provenance.
- Relationship to models created from this dataset.
  - If multiple models are created from a single dataset, but the dataset itself has been changing and evolving, it would affect reproducibility.
- Scope changes over time
  - If the scope of a dataset changes significantly, it should be considered how relevant the provenance would be.
  - E.g., if a dataset is “liver cancer” and then later expanded to be “liver cancer and other conditions”, the provenance to the original dataset might no longer be relevant. Similarly, if a dataset contains “liver cancer CT exams”, and later expanded to include “liver cancer MRI exams”, the hierarchical relationship may no longer be relevant.
  - If a dataset changes, perhaps not only the version value of the dataset gets updates, but perhaps also the reason for change may need to be captured.
- Retiring a dataset
  - Datasets that are no longer used should be retained for audit and recordkeeping purposes, but marked as inactive.

Profile writers may consider some of these questions for further discussion:

- How do we define and communicate the identity and provenance of an "instance" of a training dataset?
- How and when do you communicate modifications to datasets that are in use?

3.3 Model Creation Use Cases

Model Training Use Cases encompass planning and executing the training of an AI Model using training and testing datasets.

3.3.1 Define Model Task

A Data Scientist identifies the task the AI Model will be trained to perform.

Considerations that may have technical interoperability aspects include:

- What the model is expected to do:
  - A model may perform classification ("what is it"), object detection ("which regions contain it"), or segmentation ("which pixels contain it").
  - A model may be required to identify unique instances of the area of interest (e.g., "three metastases are present") or simply its presence (e.g., "metastases are present").
  - It may be necessary to also convey what the model is not expected to do.
  - See Section 2.1 for example applications an AI Model might address.
- Definitiveness of what the model will identify.
  - This may include computer-aided detection (CADe), computer-aided diagnosis (CADx), computer-aided triage (CADt), or computer-aided optimization (CADo). See [BIB3.3.1-1] for details.
  - AI Models should consider capabilities supporting explainability of results.
- Target performance and necessary clinical performance (see Common Mechanics: Model Performance Metrics for potential metrics of relevance)

Profile writers may consider some of these questions for further discussion:

-

3.3.2 Orchestrate Training

A Data Scientist designs the structure of the AI Model and plans the process that will be used to train the Model.

Considerations that may have technical interoperability aspects include:

- Selecting a model framework and/or toolkit

    - Developing training software from scratch can be very time consuming. Building training software on top of existing toolkits reduces time and takes advantage of training techniques and optimizations built into toolkits.
    - See Appendix C.4 for commonly used toolkits.
- Designating training, testing, and validation datasets
    - See Dataset Assembly Use Cases: Organize Datasets
    - Sequestration should be applied as needed. See Common Mechanic: Sequestration.
- Performing necessary transforms upon data ingestion
    - See Common Mechanics: Transform
- Augmentation of datasets to enrich model training.
    - This involves "virtual records" that are generated on the fly during training to encourage the model to be invariant to these transforms (e.g., rotate study 1 degree to create virtual datasets)
- Selecting a model type and architecture for the neural network.
    - See Appendix A: AI Background
- Determining model hyperparameters, e.g.:
    - Number of layers
    - Number of channels
- Selecting an initial model seed (or base model weights)
    - Models can be trained from scratch (randomly initialized weights) or fine-tuned (pretrained model weights) via transfer learning.
    - Models that are being fine-tuned do not have to originate from the same research organization.
- Determining training hyperparameters, e.g.:
    - Number of training epochs
    - Learning rate
- Organization of epochs and testing/validation patterns
    - Model testing methodology as the model is being trained.
- Distributing training across systems using the same dataset
    - Spreading computation of model training and training data storage across multiple machine nodes or GPUs to enable parallel computation to process larger models and larger training data sets [BIB3.3.2-1] [BIB3.3.2-2] [BIB3.3.2-3].

- Distributing training across institutions using distinct datasets
    - Federated learning occurs when updates to models weights from each institution are combined [BIB3.3.2-4] [BIB3.3.2-5].
    - Distributed learning occurs when a model is trained at one site, then transferred to another site and further tuned, and then transferred to yet another site and further tuned.
    - The Data Scientist may decide that site-specific fine-tuning will take place at clinical sites where the model is installed. This decision may depend on whether the model is considered a regulated medical device.
- Hyperparameter optimization
    - Once a model has been trained, the model and training hyperparameters may be changed triggering another training of the model. This process allows for the discovery of improved model configurations and weights.

Profile writers may consider some of these questions for further discussion:

- When conveying model hyperparameters, how much of that is considered needed to be interoperable?
- When different institutions are communicating model weights and deltas, should interoperable technologies be considered? E.g., XDS [STD-XDS]?
- What further differences need to be described if “the training is taken to the data” or “the data is taken to the training location”?

3.3.3 Train Model

A data scientist trains an AI Model using data records from a training dataset.

Model training adjusts the weights and biases of the model nodes to fit the transformed outputs of the models to match expected ground truth outputs as closely as possible.

Considerations that may have technical interoperability aspects include:

- Pre-fetching data
    - Expedient fetching, parsing, and transforming of data elements to not inhibit model training.
    - Efficient transfer of the data elements to the accelerator (e.g., GPU) in a format it “understands.”
    - Training can be run on local datasets or on remote datasets in a federated fashion.
- Executing training

    - Processing power and storage for training can be local or distributed across multiple servers.
    - See Appendix A: AI Background.
- Completing training
    - Capturing model performance during training and once it is complete.
    - Exporting model weights once training is complete.
    - See Common Mechanics: Exchanging Models

Profile writers may consider some of these questions for further discussion:

- How is the dataset represented in the model?
- How is training responsibilities shared amongst different enterprises?
- How do you know when you are done (e.g., also avoid overfitting)?
- Are there any provisions needed to link “previous dataset used” with the current dataset?
    - E.g., if the original model original dataset was CT studies on Vendor X modality, and fine-tune model uses MRI – is this erroneous? How is this flagged?
- What is the impact on the identity of a model when fine-tuning it?
- What is the impact of “changed annotation styles” around fine-tuning?

3.3.4 Test Model

A data scientist evaluates the trained model using a test dataset.

Often a separate team of data scientists and biostatisticians not involved in training conduct testing. Pre-clinical validation may be conducted by the same organization or a separate pre-clinical testing organization.

Considerations that may have technical interoperability aspects include:

- Using appropriate validation datasets
    - Ensure data from training is kept separate from validation data.
    - See Common Mechanic: Sequestration
- Assessing test performance via relevant metrics (see Common Mechanics: Model Performance Metrics)
- Identifying that a model is ready for next activities (e.g., external validation)
    - Software engineering verification and validation processes can be applied to ensure that not only the correct AI Model was created, but that the AI Model was created correctly.

- Recording and parsing results
- Tracking provenance of the testing dataset and the model being tested
    - See Common Mechanic: Provenance.
- Communicating results and provenance to subsequent testing and validation activities
- Exporting test results and datasets for
    - Peer-reviewed publications.
    - AI Challenges
    - Federated or collaborative model development

Profile writers may consider some of these questions for further discussion:

- Does testing differ (e.g., use different testing datasets or performance criteria) depending on whether this test follows a "from scratch" training run, a "transfer learning" training run, or a "fine-tuning/localization" training run?

3.4 AI Application Distribution Use Cases

AI Application Distribution Use Cases encompass making an AI Model and its associated components into an AI Application that can be instantiated elsewhere for further validation or production uses.

3.4.1 Package Application

A data scientist creates a distributable "application package" containing AI Model(s).

An AI Application could encompass just the AI Model (data-only packages), or include connectors, transforms, and services in self-contained executable containers (executable packaged). See Entities: AI Application.

Shared models must be interoperable in the ecosystem to which they are delivered.

Considerations that may have technical interoperability aspects include:

- Identifying minimum hardware and software requirements
    - The AI Model and surrounding AI Application will need to be validated that they operate according to specifications.
    - Validation standards will include hardware minimums for CPU processors, RAM memory, hard drive space, Graphic Processor (GPU) for running the model in a clinical production environment.
- Identifying packaging with other software or hardware

  - Once a model is validated and has obtained required regulatory clearances it may be packaged with and sold as part of another medical software product, e.g., an Image Display, or packaged and deployed independently.
  - In this case a Docker file format may be well suited to contain a self-contained executable model package.
- Preparing model and application details useful for enumeration (this information is called "AI Model manifest" going forward)
  - What the model does (e.g., classification or segmentation, for what body part, for what modality)
  - Model identity, source, and format
    - Model provenance; see Common Mechanics: Provenance
  - Data input and output characteristics
  - Data transformation characteristics; see Common Mechanics: Transform
  - AI Application Host characteristics (e.g., operating system and platform)
  - Appropriate use of the AI Application and contraindications
  - Status and performance of the model

Profile writers may consider some of these questions for further discussion:

- What format should be used for encoding the AI Model manifest?
- Which container formats should be used for distributing self-contained AI Applications (Docker is a common, Windows self-installable, RAR files)?
- What specific regulatory rules should be conveyed?
  - E.g., regarding labeling intended use and warnings.

3.4.2 Distribute Application

A data scientist distributes the packaged AI model.

Considerations that may have technical interoperability aspects include:

- Identifying targets for sharing the AI Application
  - It could be shared between institutions, for research purposes, or for commercial purposes.
  - See Common Mechanics: Exchanging Models
- Choosing a distribution method, e.g.:

    - **Model Zoos:** The data scientist could upload the model package to a publicly accessible ML cloud platform or to a model repository to allow other data scientists to download the model.
    - **File Sharing:** The data scientist could share the model directly using a common file sharing service or a dedicated server.
        - **Website Download:** Publishing on a vendor website, or from a jurisdictional oversight agency. The model package might be stored in an Enterprise Resource Planning (ERP) system
- Sharing the distribution reason
    - For validation
    - For research use
    - For clinical deployment
    - For trial use

Profile writers may consider some of these questions for further discussion:

- Should the distributed models be access-controlled, and by what means?
- What transport protocols and security mechanisms should be used for model transfer (e.g., HTTPS or SFTP)?

3.4.3 Discover Application

A data scientist and/or IT operations finds and obtains an AI Model for local use.

Considerations that may have technical interoperability aspects include:

- Querying for AI Applications
    - An AI Application source should support model queries based on attributes that would be stored in the model manifest, such as:
        - Intended purpose.
        - Supported modalities.
        - Status of the model
        - Target O/S
        - Application area
        - Body part
        - Official regulatory and release status of the model
    - The definition of a standard format for a model manifest file would be helpful to store properties of each model in a structured form to allow for structured queries.

- Retrieving an AI Application
    - A retrieval location should be made available to retrieve the AI Application, such as an HTTP link.
    - There should be means to assert the authenticity of the AI Application, so the user has the confidence that they downloaded the expected component.
    - See Common Mechanics: Provenance.
- Delivering the AI Application onto an AI Application Host
    - Once the desired model is identified based on the query results, the user copies the model into their own work area (if working on a ML cloud platform) or downloads it to their local system.

Profile writers may consider some of these questions for further discussion:

-

3.4.4 Install Application

IT Operations unpackages, installs, and configures a packaged AI Application.

This may be done in collaboration with a local data scientist in their local environment, for example within a Model orchestration framework.

Considerations that may have technical interoperability aspects include:

- Conveying the general installation process; this typically includes:
    1. Running the installation program,
    2. Registering the AI Application within the host environment.
    3. Licensing of the AI Application.
    4. Configuration of the AI Application.
    5. Testing the proper installation of the AI Application, e.g., by using it on test data that was packaged with the application and comparing the inference output with the expected result.
- Target environments and scenarios (both data-only and executable models):
    - **Local Workstation:** Installation on a local computer within a research environment for evaluation of the model and/or potential improvements to it. Many different commercial or proprietary software environments could be used to run the model.
    - **Commercial Model Orchestration Framework:** Installation in a commercial model orchestration framework for running AI Model inferences, e.g., for "official" model validation by a Clinical Research Organization (CRO), a hospital, or a payor (see

    External Validation Use Cases). This framework could be running in a local private network or in a cloud environment.

    - **Server Deployment:** Installation in a hospital IT environment for clinical use on patients (see Clinical Usage Use Cases). The AI Application could be deployed on a dedicated on-prem AI server or PACS workstation, or it could be deployed to an on-prem or cloud-hosted model orchestration framework provided by a medical device manufacturer. In all these cases the AI Application would take on the role of the AI Performer or be embedded into an AI Performer as a proxied AI Model following the IHE AIW-I Profile (see [STD-AIW-I]).
    - **Hardware Deployment:** Installation on a hardware device, e.g., integrated into an imaging modality or another type of medical hardware device. (This scenario may be done by the vendor as part of product development, not by clinical or operations staff at a site). This type of installation is most likely to be performed by the vendor of the hardware device and therefore does not affect interoperability.
- Automation of the AI Application installation
    - E.g., whether the installation fully automated (e.g., in case the model is deployed as a Docker file) or does it involve manual steps.
    - If a previous version of the same model is installed, an un-installer may have to be run first or multiple versions may be used in parallel.
- Incorporation of AI Application details into local management databases.
    - E.g., model identity, purpose, provenance, etc.
    - Mechanisms for confirmation of model ID and version after the download to ensure they got the correct model.
- Registering an AI Application in a model orchestration framework
    - This includes describe its inputs, outputs, and resource needs for running the models (e.g., CPU/GPU, memory, and temporary file storage requirements)
    - This helps the framework manage the resources when starting up running an inference job with the AI Application, e.g., when running the same application in parallel on a distributed set of compute nodes.
- Standardizing configuration settings for AI Applications
    - How to turn optional features on/off
    - Setting input and output paths
    - Identifying IP address, hostname, and port
    - Identifying SSL, authentication, and authorization
    - Selecting a data access protocol like DICOM (and accompanying data like AE title)

- Licensing mechanisms for AI Applications
    - E.g., seat-based licenses, per-use subscription models.

Profile writers may consider some of these questions for further discussion:

-

3.4.5 Integrate Application

An ML Ops Analyst prepares the environment and its users to perform inference with an installed AI Application.

Considerations that may have technical interoperability aspects include:

- Coordinating the necessary transforms of incoming data to be suitable for the AI Application.
    - See Common Mechanics: Transform
    - Defining interfaces for data input and output to AI Models, in cases where the model comes in “data-only” package form and/or requires input data in a particular format rather than accepting a wide range of DICOM objects.
    - Providing and configuring appropriate software modules to perform transforms on input data to make them conform to whatever data format the model requires, e.g., decompression or resampling of pixel data.
- Coordinating the necessary transforms of outgoing data suitable for consuming applications.
    - See Common Mechanics: Transform
    - Providing and configuring post-processing functions that format output data of a model, e.g., wrapping raw pixel data or segmented volumes into DICOM compliant objects. Jupyter Notebooks are a common mechanism used for this purpose.
    - Mapping model outputs in form of simple string identifiers, e.g., “pneumo-thorax suspected”, or numeric measurements, to coded concepts compatible with the consuming systems, e.g., using DICOM Supplement 219.
    - Mapping an AI Application result from the DICOM format it provided into another DICOM format that is compatible with the consumer of the result, e.g., transforming a DICOM Surface segmentation object into a DICOM Segmentation object.
    - Mapping supportive elements like explainability documentation that goes with the outgoing data.
- Coordinating chaining AI Applications together to form processing pipelines.
    - E.g., feeding the output of a lung nodule *detection* application from vendor A into a lung nodule *characterization* application from vendor B.

  - In this case, transport issues may need specific attention, e.g., it may require bridging between a cloud-based AI Model and another clinical app running on a local workstation.
  - Combining AI Applications implemented using traditional image processing algorithms with AI Applications based on deep-learning networks, where each application could have different needs in terms of the underlying infrastructure, e.g., one requiring GPU's vs. the other requiring CPU's or one is parallelizable and can be distributed to multiple compute nodes and the other one is not.
- Registering AI Applications within a model execution framework to enable their automatic invocation based on the nature of incoming data, including information such as:
  - The types of data the AI Application can do inference on
    - E.g., individual imaging studies for AI Applications performing tasks to support a radiologist's interpretation of images.
    - E.g., sets of imaging studies and other EHR data for AI Applications performing assessment of patient treatment regimens and population health.
  - The types of analysis the AI Application performs
    - Mapping of AI Applications to procedure codes used in scheduling exams or to other types of codes used in the operating environment.
- Verify correct operation of the AI Application.
  - For initial verification before go-live, or for surveillance after go-live
  - E.g., by running regular checks with controlled input data and expected results.
  - Maintaining a "status" of AI Applications within the operational environment to indicate whether it is verified, ready for production, etc.
  - This may require connectivity into IT monitoring systems and may use IHE ATNA [STD-ATNA] or SOLE [STD-SOLE].
- Connecting into the right environment
  - E.g., development, pre-production, staging and production environments; or where AI Applications would only be used for processing patient exams for one physician only.
- Promoting an AI Application from one environment to another
  - This switch-over may require reconfiguration of network settings, permissions, and licenses, and maybe facilitated by a hosting AI Orchestration framework.

- Some environments may utilize dual live pathways, and eventually switching over, as follows:

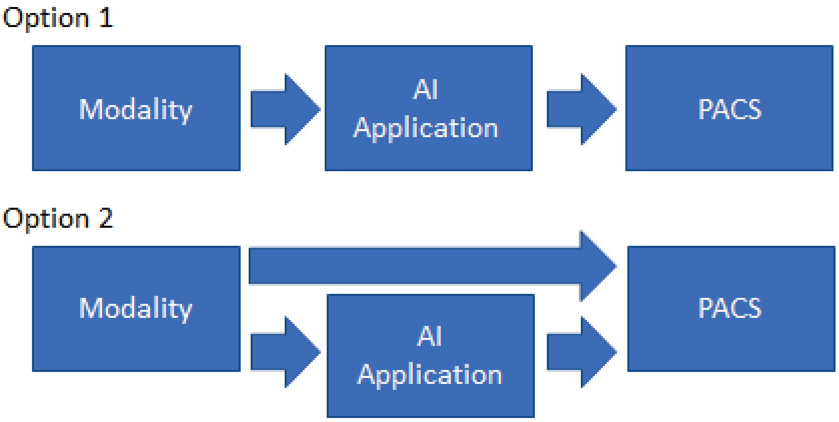

Figure 3.4.5-1: DICOM and other content can flow directly from one system to another, or it can utilize dual pathways

  Care must be taken that duplicate objects are not created.

  - Keep systems in a "pre-go-live" state, e.g., perhaps triggering an AI service is live, but delivery of results goes to an adjacent system.

- Supporting billing and reporting systems
  - These systems must be configured to allow for tabulating the exams on which a model is run to monitor proper operation, and to provide input to billing.
- Training clinicians and support personnel on operating the AI Application.
- Training clinicians on how to appropriately use the AI Application outputs
  - Product specialists need to train and educate potential users of the model to be aware of how the model output is provided to them.
  - E.g., the behavior of the AI Application that has been integrated in PACS Viewing stations or on dedicated workstations, and how they are expected to interact with them.
  - This training covers both the model itself and the other systems the user interacts with in the context of the model (e.g., result display).

Profile writers may consider some of these questions for further discussion:

Are there mechanisms for twin pathways for the same processing task, e.g., a non-AI original path for a breast CAD application, and a path with an AI-based breast cancer detection application, that would allow to compare the performance of both applications across a larger collection of exams? How does this relate to validation steps?

3.4.6 Manage Application Service

An ML Ops Analyst makes the AI Application available as a Service and supports ongoing operation including control, monitoring, and scaling.

Considerations that may have technical interoperability aspects include:

- Starting AI Application services
    - For models running in an AI Orchestration environment, it would be the job of that framework to ensure the model is ready to perform assigned tasks. It could do this by starting it up on a dedicated server or VM to allow for a warm start, or at least ensure there are enough CPU / GPU, and memory resources available to perform efficiently when it is started up to process data. To allow the proper allocation of resources for each model, it is vital that the resource requirements are discoverable in the model registry.
    - For models running on a dedicated workstation or server, the proper allocation of hardware resources would be done by the IT department before directing studies for processing by the model.
    - For models running embedded in a modality, the proper sizing of the hardware is expected as part of the design phase of the device, and not the responsibility of operator action.
- Monitoring AI Applications
    - Processes should be in place to monitor the proper operation of the model to identify:
        - resource bottlenecks
        - problems with input data
        - unusual environment conditions like low disk space and high temperature
        - any other unexpected behavior that would require corrective actions by an IT administrator.
    - Consider built-in monitoring capabilities of the infrastructure running the AI Application, e.g., Kubernetes.
    - Monitor turnaround times, i.e., the time it takes to process a study with a given model and make results available to downstream systems.
    - Monitoring may be done internally by ML Ops Analysts or remotely by Product Support Teams
    - System performance issues may be an indicator of possible clinical performance issues.
        - E.g., if a model is uncharacteristically taking hundreds of gigabytes of disk space, there may be clinical impacts.

- Scaling AI Applications
    - In case the model is run in a virtualized environment with dynamic resource allocation, such as a Kubernetes cluster, the framework could dynamically increase or decrease the available hardware resources to control optimal hardware utilization based on recent and current workload.
    - There may be conditions that warrant "burstable" capability for AI Applications. For example, adding additional capacity during daytime clinic hours and scaling back during the night.

Profile writers may consider some of these questions for further discussion:

- How to automatically startup or shutdown AI Applications safely and without interruption to the clinical operations, e.g., in case of environmental issues.

3.5 External Validation Use Cases

External Validation Use Cases encompass validation of an AI Model by parties "external" to those developing the model.

Although the use cases in this group all involve validation of the model, the different external parties may have different goals and methods.

These use cases assume the model has been installed at the point of validation.

3.5.1 Validate Model (by Regulator)

A Regulatory Body validates the safety and efficacy of an AI application using a validation dataset.

The regulatory approval process typically involves collecting additional data to meet a statistical model for accuracy. Validation focuses on both "was the right product built" as well as "was the product built right".

Considerations that may have technical interoperability aspects include:

- Intended use of the AI Application.
    - Specifying the types of data that the AI Application is appropriate for (body part, modality, patient population).
    - Specifying the type of AI Application – computer-aided detection (CADe), computer-aided diagnosis (CADx), or computer-aided triage (CADt).
    - Can be partially derived partially from a model purpose and the dataset that was used.
    - Whether the application is novel functionality or substantially equivalent to another application
        - This is indicated, for example, by the FDA and 510(k) and Premarket Approval designations

- Using a validation dataset
    - Datasets at this step should not have been used to build or internally validate the AI Application.
    - It should simulate, as best as possible, clinical conditions (such as generated data from a modality, rather than a simulated pre-transformed rendered file).
    - See Dataset Assembly Use Cases: Organize Datasets.
- Understanding validation results
    - Validation results should be available as both human-readable and interoperable with other systems that comprise the solution.

Profile writers may consider some of these questions for further discussion:

-

3.5.2 Validate Model (by Payor)

A Payor Administrator validates that a model addresses their statistical needs.

AI Applications are used beyond clinical purposes to drive hospital workflow and measure efficiencies and effectiveness. This may include hospital finance and purchase teams, ministries of health and governmental agencies that reimburse for performed procedures, as well as insurers for private health. As a basis for reimbursements, payor validation needs to show there is a cost benefit to using the AI Model. Although an AI model may pass regulatory validation and have clinical significance, payors may require validation to determine reimbursement levels.

Considerations that may have technical interoperability aspects include:

- Collecting data to demonstrate return on investment:
    - Return on investment is an important metric – e.g., how much money is saved through early detection or rapid triage of clinical conditions.
    - Collecting timestamps, using a structure akin to SOLE [STD-SOLE], to convey the time particular tasks take in a clinical environment, with measurements taken before the AI Application is deployed, and after

Profile writers may consider some of these questions for further discussion:

-

3.5.3 Validate Model (by User)

A Clinical Validator validates that an AI Application is producing accurate, appropriate results.

AI Applications are evaluated clinically using data not seen by the application before, by using an external validation dataset. Data is acquired from the hosting sites from clinical systems, annotated as needed for model input, and structured in a dataset separate from any data that was used in training or validating the model inside the AI Application. The Clinical Validator submits data to the AI Application and reviews the results inside a clinical application (typically cordoned off from production systems) and compares the results with ground truth to determine if the AI Model is performing correctly. Results may be referred to an AI Oversight Committee.

This activity may also be referred to as acceptance testing. Vendor validators may be responsible for performing testing as part of solution development processes.

Considerations that may have technical interoperability aspects include:

- Type of AI Application being validated.
    - An AI Application created by the local institution.
    - A commercial AI Application being developed or validated, for example, as a research trial
    - As part of a trial program prior to purchasing a solution.
- AI Application deployment environment
    - The AI Application may be deployed in a development environment, a pre-production environment, a testing environment, or in a production environment with appropriate controls.
    - Support teams, such as ML Operators, IT Analysts, and local data scientists likely validate the technical operation of the AI Application with the environment prior to handing it off to Clinical Validators to review.
- Communicating success
    - Scoring systems and AI Application metrics
    - Identifying when an AI Application is ready for production use.
    - Finding comparable models to score against
    - How to communicate scores – coordinate comparisons and benchmarking
    - See Common Mechanics: Data Quality
- Addressing validation failure
    - When clinical validation fails, an AI Application and its associated model(s) may be flagged for local tuning. See Model Creation Use Cases: Train Model.

- Multiple reviewers
    - Ground truth provided by a trained clinician may differ for a variety of reasons (an imaging study with multiple findings, or where an observation looks like a particular finding, but the actual result when combined with other factors makes it a different truth).

  - In addition, AI Applications should be externally validated than those clinicians that may have had a hand in training the model.
  - Results from AI Applications that differ from ground truth should be reviewed to determine if there are clinical factors or artifacts that may have influenced the difference.
  - False positives and negatives need to be evaluated for clinical significance when comparing to ground truth.
- Iterative design
  - In active or continuous learning frameworks, a Clinical Validator may review an AI Application and signify that that further fine-tuning of the AI Model is necessary. Locally annotated datasets may be used to establish further local ground truth to enable re-training, as necessary. This may have impacts on other validation steps (by regulatory and payor validators).

Profile writers may consider some of these questions for further discussion:

- What metrics are used to compare and rank different models with the same or remarkably similar goals? See [BIB3.5.3-1] and [BIB3.5.3-2] for examples.
- Are there assumptions made when performing clinical validation if an AI Application passed regulatory validation and/or payor validation?
- Aside from AI Applications themselves, is there clinical validation applied to various components within a clinical validation?
  - For example, a model produces an inference result, and that result is converted into a DICOM-SR TID1500 object. What value is there if the “transform” is validated outside validating the entire AI Application?
  - Can standardizing validation tooling be used to help assemble testing / training datasets and doing transforms from clinical repositories?
- What are the differences to clinical validation if an AI Application is created and validated within the same institution, from a different institution, or from a commercial entity?

3.5.4 Surveil Model

An ML Ops Analyst or Product Support Team monitors the operational performance of an AI Application.

Monitoring processes run to monitor that AI Models perform as expected. events could be real-time, regularly scheduled, triggered when an event has occurred in the environment, or triggered when specific metadata on incoming or processed data has been found.

Considerations that may have technical interoperability aspects include:

- "When" an AI Application is monitored
  - In real-time
  - Regularly scheduled.
  - Triggered when an event has occurred, for example:
    - by installation of a new scanner, replacement of hardware or firmware components, or new reconstructions algorithms
    - a new protocol or new mode of operation (e.g., multi-energy)
    - changes in "contrast or radiotracer wait times.
    - download/installation of a new model / transform / platform / operating system / hardware.
    - changes in target patient demographics (e.g., adding pediatric patients).
  - Triggered based on examining meta-data, for example:
    - Scanner name
    - Manufacturer
    - Acquisition software version
    - Acquisition firmware tags
    - Patient age
- "How" an AI Application is monitored
  - Control studies with known outcomes can be used on a continual basis to ensure models do not skew too far from known truths. If control studies start to fail, that may signal that something has changed in the hospital ecosystem.
  - Automated feedback may be captured; for example, based on daily QA checks to confirm that studies with known ground truth are still predicting correct results.
    - A sample QA check would be to use studies with a known ground truth, which helps to detect changes to an AI Application.
    - Using studies with a known ground truth will not detect whether there is an issue with image acquisition (for example, a replacement detection plate might produce slightly different imaging objects that can throw off an AI Application).
- "What" aspects of an AI Application are monitored
  - Whether the AI Application is producing "correct" results
  - The number of resources that is being used by the AI Application.
  - The amount of time it takes for work items to be processed.

    - The number of studies queued for processing, per priority.
- “Who” has access to AI Application monitoring information
    - Whether logs are transmitted to the AI Application Vendor and the content they are provided. See also Feedback Use Cases: Circulate Feedback.
    - Whether Payors have access for billing and reimbursement purposes and the content they are provided.
    - Whether Health Information Management Team use the results to supplement the patient heath record
    - Whether Health Policy Agencies use these results for population heath reporting requirements
    - Whether System Regulators use the data to track performance across institutions for recall and warning issuance

Profile writers may consider some of these questions for further discussion:

- How are results from surveillance communicated back to the data scientist?
    - What are the differences in interoperability if surveillance is performed by:
        - the AI Application hosting institution?
        - the AI Application manufacturer?
        - the regulatory body that approved the AI Application?
        - the payor that reimburses based on observations from the AI Application?
- What are the differences when surveillance is performed from a central system, or when it is performed in a de-centralized way?
    - If surveillance is decentralized, how does it access surveillance datasets?
- Does automated surveillance tasks follow the same AIW-I [STD-AIW-I] workflow or does it use more specialized means to manage the associated tasks?
- Is there a type of dataset called a Surveillance Dataset? Is it managed using the same process as described in 3.2?
- Does surveillance happen in production environments, or does it happen in some form of shadow environment that may be used?
    - What would trigger suspension of the algorithm in routine use?
- When performance is degrading, can an AI Application be disabled, ignored, adjusted, marked, or re-routed??
- Is feedback generated by surveillance activities treated differently from feedback generated by users (in 3.7)?

1840 3.6 Clinical Usage Use Cases

Clinical Usage Use Cases encompass the daily use of AI Applications in a clinical setting.

While many of today's AI solutions aim to assist radiologists in reading and reporting on patient exams, there are many other applications where AI can improve clinical quality and outcomes.

The existing IHE profiles AI Workflow for Imaging (AIW-I) [STD-AIW-I] and AI Results 1845 (AIR) [STD-AIR] focus on the former objective. As the following use cases aim to explore gaps and areas beyond the scope of these profiles, it is assumed the reader is familiar with them. The following diagram is taken from the AIW-I Profile for quick reference.

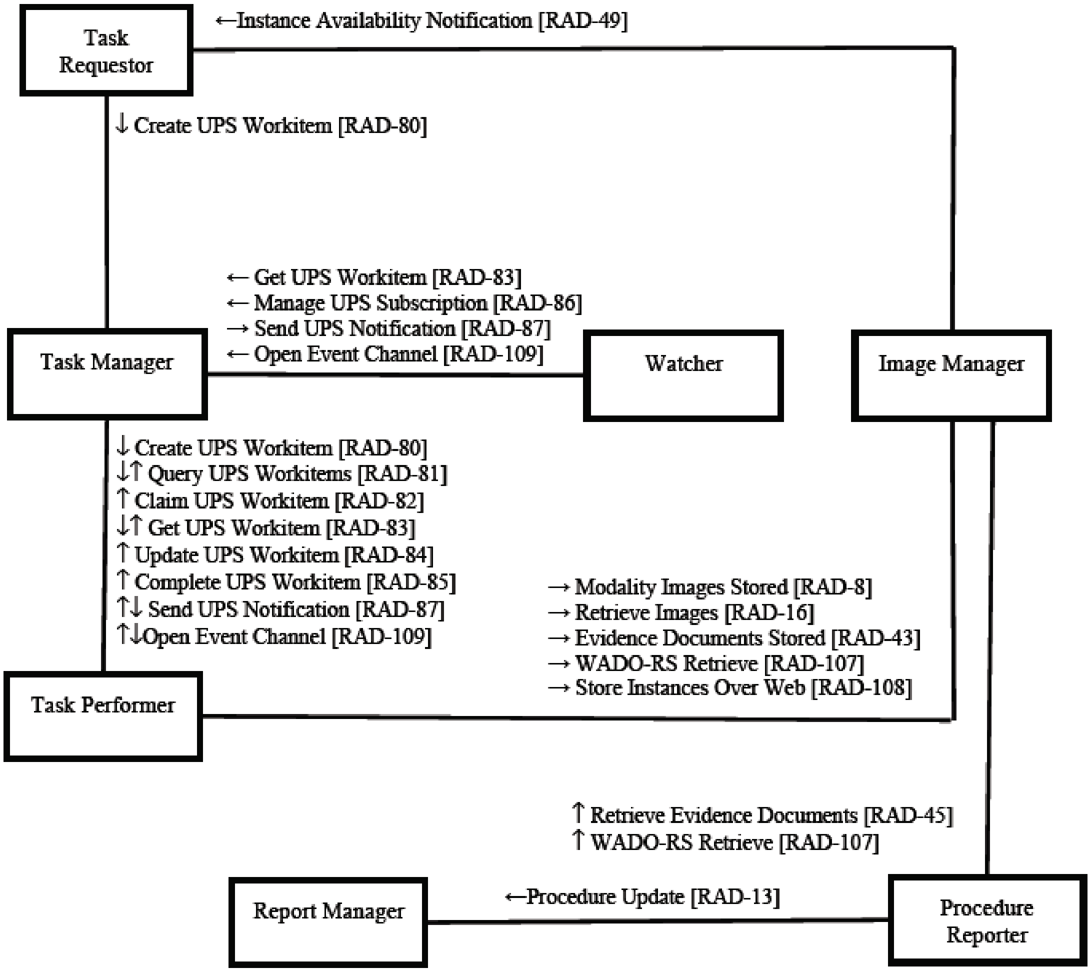
Task Requestor
←Instance Availability Notification [RAD-49]
↓ Create UPS Workitem [RAD-80]
← Get UPS Workitem [RAD-83]
← Manage UPS Subscription [RAD-86]
→ Send UPS Notification [RAD-87]
← Open Event Channel [RAD-109]
Task Manager
Watcher
Image Manager
↓ Create UPS Workitem [RAD-80]
↓↑ Query UPS Workitems [RAD-81]
↑ Claim UPS Workitem [RAD-82]
↓↑ Get UPS Workitem [RAD-83]
↑ Update UPS Workitem [RAD-84]
↑ Complete UPS Workitem [RAD-85]
↑↓ Send UPS Notification [RAD-87]
↑↓Open Event Channel [RAD-109]
→ Modality Images Stored [RAD-8]
→ Retrieve Images [RAD-16]
→ Evidence Documents Stored [RAD-43]
→ WADO-RS Retrieve [RAD-107]
→ Store Instances Over Web [RAD-108]
Task Performer
↑ Retrieve Evidence Documents [RAD-45]
↑ WADO-RS Retrieve [RAD-107]
Report Manager
←Procedure Update [RAD-13]
Procedure Reporter

Figure 50.1-1: AIW-I Actor Diagram

3.6.1 Initiate Inference

A persona or system initiates inference on an inference data record.

In AIW-I, inference is initiated by the “Task Requestor”. Systems that might incorporate a Task Requestor (i.e., “be grouped with”) include imaging modalities, EMRs, Report Managers, Order Fillers, or any other system needing a task performed.

Processing tasks are communicated as DICOM UPS-RS Workitems, which encapsulate references to input data, the type of task to perform (encoded as a processing procedure code), and optionally references to data element access locations, and the priority of the request (see [STD-AIW-I], transaction [RAD-80]).

In AIW-I, the Task Manager brokers tasks from the Task Requestor to the Task Performer. Typically to select an inference performer, it is necessary to know that the performer exists, know what tasks it can do, and know that the data record is adequate for, and accessible by, the performer.

Considerations that may have technical interoperability aspects include:

- Determining when inference is to be performed.
    - For AI applications in 2.1.3 (Image Acquisition), 2.1.4 (Image Generation), 2.1.5 (Image Analysis) and 2.1.6 (Image Interpretation), this might be watching for a complete acquisition of an imaging study (a sufficient time has elapsed before no further DICOM objects being received), or via MPPS, or Instance Availability Notification (IAN - see Section 50.1.1.1 of [STD-AIW-I]).
    - For AI Applications in 2.1.1 (Ordering and Scheduling), 2.1.2 (Procedure Guidance), 2.1.7 (Patient Management and Treatment Planning), 2.1.8 (Population Health) and 2.1.9 (Departmental Analysis), this might include HL7 message triggers or proprietary mechanisms.
    - Other mechanisms are also valid, like polling, or triggered by a human and triggered by interacting with a UI.
- Determining what AI Applications should perform inference on a given input data set.
    - This may be driven by requested procedure codes, modality and body part or anatomical region associated with the exam, see AI Application Distribution Use Cases: Integrate Application.
        - Procedure codes may be extracted from IAN notifications, although it is possible, they are not available in the absence of MPPS (Modality Performed Procedure Step).
    - Additional information about the exam may be retrieved from other systems,
        - e.g., query the EMR to get the admission reason associated with a study.

  - For tasks driven by clinical messages (such as order or report notification), this might be done by looking at the type of HL7 trigger message, or some of the specific fields (like inpatient versus outpatient, or reason for admission).
  - Some tasks are selected by a human interacting with a UI.
- Selecting the AI Application to perform the inference.
  - For imaging workflows, this could be done in pull workflows by encoding in the Scheduled Station attribute, or arbitrarily done in push workflows. See [STD-AIW-I] for details.
  - For non-imaging workflows, this could be done through HL7 trigger messages, API calls, FHIR requests, or other means.
- Pre-caching inference data elements
  - See next use case "Access Inference Data Record."
  - A "non-PHI-safe" AI Application, e.g., an app running on a wearable device evaluating the wearer's biometric data by comparing it with other patient records in the hospital system, may need to have input data de-identified and pre-cached in a location accessible by the device.
- Communicating the task to the performing AI Application
  - In AIW-I, the details are all in the UPS Workitem. Consider adding any other information that may be useful to convey to know for the AI Application to run most effectively and efficiently.
  - In "Implicit Workflow", the task is not explicitly communicated to the AI Application. Instead, data is pushed to the inference performer which presumes what task to perform, possibly upon inspection of the data. This is often done where the inference performer is only capable of a single task.

Profile writers may consider some of these questions for further discussion:

- How do other AI Applications that are not analyzing images do assist a radiologist in creating a report initiate inference, for example, applications assisting image-guided interventions and surgery such as Cath lab procedures, image-guided biopsies, orthopedic interventions, or tumor ablations.
- How can inference tasks be prioritized? For example:
  - If ten tasks were submitted, 7 routine tasks, 2 urgent tasks, and 1 critical task, how are tasks performed in the correct order?
  - If, while performing a long-running routine task, an urgent or critical task comes in. How should it be handled?
- What is the role of IAN in other, non-diagnostic workflows?

- What are alternate workflows / triggers that have not been considered in above use cases?
    - UPS Workitems can be created by a variety of actors as outlined in AIW-I.
    - Modality, EMR or Report Managers as initiators of AI inferences. How does scheduling and cancellation of inferences work in those cases?

3.6.2 Access Inference Data Record

An AI Application accesses the inputs that comprise an inference data record.

The inference data record which will be the inputs to the inference may include both imaging and non-imaging data elements. The data record may be a single bundle, or the data elements may be distributed across multiple sources (e.g., the record references an image which may be retrieved from the PACS, and a lab result object which may be retrieved from a lab information system). It is possible that the inference data record does not contain everything needed for the AI Application to perform an inference.

There are many other scenarios were discovery and retrieval of additional information is needed for inference, including longitudinal imaging and patient record information, such as frequently stores in an EMR, as well as data stored outside of described healthcare data standard processes.

Considerations that may have technical interoperability aspects include:

- Accessing the data elements of the inference data record
    - Pull – the AI Application retrieves the data elements.
        - AIW-I assumes data elements are pulled using protocols like DICOMweb [STD-DICOMWEB], DICOM DIMSE [STD-DICOM], and XDS [STD-XDS].
    - Push – the data elements are pushed to the AI Application.
        - This is sometimes used in Implicit Workflow where, for example, data is stored into the “input folder” of the AI Application.
- Accessing data elements beyond the inference data record
    - Some AI Applications may be able to make use of information which might not have been included in the inference data record and may have mechanisms for exploring available data servers for relevant data.
        - E.g., obtain prior studies when tracking lesions across multiple time points. If the Task Requestor did not query the Image Manager for prior studies and include them in the inference data record, the AI Application may search for such data itself.
        - E.g., in abdominal CT studies, information about contrast administration and imaged body part(s) may be ambiguous or missing in the DICOM header.
        - Note: It can be adaptive/robust for the AI Application to search, but it requires query interfaces, more installation integration, and having the Task Requester do

        this means relevant priors can be pre-fetched (e.g., offline storage or another network). If the task were triggered by the order, they could even be pre-fetched before the current study is acquired.

        - For improved turnaround times it would be preferrable to have the Task Manager search for the data based on the AI Application's needs and make it available to the AI Application before it is invoked, in which case a standard mechanism to describe and communicate these needs would be helpful.

- Access to data in remote repositories
    - Based on the location of the input data and any priors, non-DICOM protocol-based retrieval may be required to access DICOM studies not stored in DICOM archives, e.g., using cloud service protocols for secure transfer between cloud and on-prem storage.
    - This is an example where a local data cache would be useful to collect all the required input data from disparate data sources and make them available to the model for inference.
- Transforming clinical data before being passed to the AI Application.
    - See Common Mechanics: Transform.
- Retrieving information from clinical documents
    - E.g., an AI Model trying to detect breast cancer might incorporate knowledge about certain genetic dispositions, as well as relevant patient history such as previous occurrences of cancer and whether they were treated, or an AI Model for detecting different types of chest diseases may want to know what implants a patient has.
    - Clinical documents may be accessible using XDS [STD-XDS] (an established IHE profile to facilitate document query and retrieval) or FHIR [STD-FHIR], becoming widely adopted to represent and retrieve non-image-based information.

Profile writers may consider some of these questions for further discussion:

- How can a "proxy archive" remain in sync?
- What is the impact of "chaining algorithms" where pixel data is converted from one format to another?
- Is it an issue if there is a gap on "notification of changes"?
- How can an AI Application's needs for additional data be effectively delegated and communicated to the model execution framework?
- How can profiles be as simple as possible for AI performers (and shield complexity)?

3.6.3 Perform Inference

An AI Model performs inference on an inference data record and generates outputs.

Considerations that may have technical interoperability aspects include:

- Managing exceptions
    - See AIW-I [STD-AIW-I] which describes use cases for managing exceptions, e.g., rejecting, canceling, or re-assigning work items.
- Proxying performing of tasks
    - Instead of AI Models being wrapped by a Task Performer they could be proxied by a separate Task Performer (see AIW-I Profile, use case 4 in Section 50.4.2).
        - This allows for integration of data-only models packages (see AI Application Distribution Use Cases: Package Application) into the AIW-I workflow without having to wrap it into a fully functional Task Performer.
        - In this case, each Task Performer could be dedicated and configured for a particular AI Model or a small set of AI Models, or it could be designed to act as a general-purpose Task Performer that can be configured for all kinds of models.
- Evaluating and reporting inference metrics
    - This may be valuable for troubleshooting and evaluating ongoing needs of updating AI Applications.
    - Data to capture includes model identity (manufacturer, appropriate use of model), model processing metadata (confidence, explainable AI), and model application state (failure behavior)

Profile writers may consider some of these questions for further discussion:

- What could be a good interface standard for how Task Performer proxies delegate work to AI models and how can they be chained together by a proxy?

3.6.4 Create Result

An AI Application encodes the output from an AI Model for consumption by the ecosystem.

Choices about how the results are encoded are typically influenced by the ways and systems in which they are used (see Clinical Usage Use Cases: Use Result).

Considerations that may have technical interoperability aspects include:

- Formats and encoding of results
    - For image analysis results, AIR identifies types of result data elements (“Primitives”) and specifies corresponding DICOM object encodings for each.
        - See “Results” under Entities: Data Element

  - Other output formats than DICOM, e.g., FHIR, and HL7.
- Standardized (or at least converged) code sets
  - E.g., SNOMED-CT, RadLex, LOINC, Common Data Elements for Radiology (RDE), (see also AIR [STD-AIR], Section 49.4.1.4 “Code sets”)
- Explainability of results
  - Where relevant, AI Applications should not only convey the result (“this is pathological condition X”), but also provide evidence and rationale why. This provides confidence to clinicians and data useful for troubleshooting.
- Using results from one AI Model as data elements for training or testing or validating another AI Model.
- Multiple data element results
  - Applications may have multiple output data elements, e.g., a segmentation of the lung, several findings, an impression of pneumonia, and a heat map indicating the parts of the image most influencing the findings.

Profile writers may consider some of these questions for further discussion:

-

3.6.5 Deliver Result

The IT infrastructure delivers the result(s) of the inference to their point of use.

The destination depends on the context and type of analysis performed. It can be a DICOM archive, as described in AIR [STD-AIR] and AIW-I [STD-AIW-I], a FHIR-based repository, or any other information system or database. In AIW-I, this is the Task Performer storing the result to a target location that may be specified in the UPS Workitem.

Considerations that may have technical interoperability aspects include:

- For image analysis results, AIW-I and AIR specify the use of DICOM C-STORE or DICOMweb [STD-DICOMWEB] STOW-RS (transactions [RAD-43], [RAD-18], [RAD-29] and [RAD-108]).
  - If the specified target is an archive, other systems can query for those results. E.g., a DICOM Viewer could display results to a radiologist during image interpretation.
- For non-DICOM output, e.g., a FHIR-encoded report [STD-FHIR-DXR], further standardization is needed (see also Clinical Usage Use Cases: Create Result).
- Inference results may need to be transcoded to be suitable for consumers.
  - This could happen before saving them to the data repository or upon retrieval.

- For embedded AI, this step may not be needed since the inference result is used on the system where it is created.

Profile writers may consider some of these questions for further discussion:

- How are results encoded that are not consumed by DICOM-aware systems? How can they be transformed?
    - E.g., reporting systems are typically consuming data in FHIR format, or EHR's using HL7. Note that DICOM Working Group 20 is working on a transformation of DICOM contents to FHIR for this purpose.
- For AI used in the context of interventional procedures and surgeries, will likely need to deliver its results to one of the systems used in the operating theatre, and only afterwards be sent to a DICOM Archive. More generally, what other forms of distribution of Inference Results are required in those scenarios?
- What humans and AI Models perform the same task, for example both assess an image for pneumonia, how are "human-generated results" and "AI-generated results" combined and routed through the end-to-end workflow?
    - E.g., preliminary ("wet read"), blinded-reads, reads direct from modality, resident reads, etc.
- More generally, what other forms of distribution of Inference Results are required in those scenarios?

3.6.6 Use Result

An AI inference result is applied to the relevant task (see 2.1 Applications of AI in Imaging).

IHE AIR [STD-AIR] and AIW-I [STD-AIW-I] address a subset of the applications listed in Section 2.1. AIR is focused on the case where an Image Display presents image analysis results and imaging studies in an integrated manner to help the radiologist with the interpretation of images. For example, suspicious areas in an image may be outlined or highlighted or an identified lesion may be characterized in detail. AIW-I also covers the scenario where image analysis inference results are used to help prioritized worklists.

Considerations that may have technical interoperability aspects include:

- Levels of AI Autonomy
    - At one end of the spectrum, the AI Result is only provided to inform a human performing the task, at the other end of the spectrum the AI Result is applied automatically, and the task is performed with no human oversight.
- Presentation of AI Results
    - Depending on the task and the level of autonomy, the AI Results and relevant contextual information will need to be presented effectively to humans.

  - Many encodings developed for robust storage, e.g., DICOM SR, leave presentation to display systems which some find challenging. The presence of multiple related AI Results can add to the challenge. Profiling should consider addressing this.
    - IHE AIR suggests some approaches (see Section 49.4.1.3 Result Presentation) but did not have time to flesh them out.
- Human approval and override
  - Depending on the level of autonomy, the AI Results will need to be approved or overridden by a human.
    - Need a mechanism for the human to indicate agreement or disagreement with the AI Result and in the case of disagreement, the "correct" result. Consider both GUI and voice-based mechanisms.
    - Need to record/store such agreement or disagreement (and correct result) - see also Feedback Use Cases
- Tracking activity and performance
  - The results, the outcome of the task the result affected, and the interactions between the human and the results may need to be logged to support retrospective review and performance evaluation.
- Applying the result to the intended task
  - See Section 2.1 Applications of AI in Imaging
- Realtime Interaction between the AI Model and a user
  - Some AI Models generate results in advance for use or presentation to a user. Even with a single "verification" step, is a single sequential pass rather than interaction.
  - Other tasks involve direct interaction with a user.
    - A radiologist might provide a seed point in an image and invoke an AI Model to segment the underlying lesion. Similarly, the radiologist may disagree with a pre-generated segmentation and interactively adjust it.
    - The radiologist may segment a portion of the image and invoke an AI Model to characterize the region as malignant or benign.
- Communicating the result beyond the current task to associated systems performing other tasks.
  - Current Reporting systems offer some of this capability using FHIR [STD-FHIR] objects provided by the AI Application. This is one of the areas where automatic conversion or mapping of information captured in DICOM objects to FHIR objects would be helpful, as some AI Models may output DICOM and not FHIR.

- Operational effectiveness can be measured and improved by gathering information that feeds into metrics like in the following examples:
    - Capture turnaround time for AI Model inference to answer the question if the AI Application is performing fast enough to have maximum impact.
    - Capture user interactions with AI Model results, e.g., to measure how often user agrees or disagrees with AI Results.
    - Capture what percentage of studies are processed by a model, and what percentage is rejected and for what reason.

IHE is considering extending the use of the SOLE [STD-SOLE] Profile to allow logging of events for the above-mentioned purposes.

Profile writers may consider some of these questions for further discussion:

- Many of the applications in Section 2.1 outside of the radiologist's reading workflow use AI inference results in quite different ways. Those need to be explored further considering the considerations listed above.

3.7 Feedback Use Cases

Feedback use cases encompass getting information on the performance of the model during routine clinical use, distributing the feedback and potentially re-training the AI Model.

3.7.1 Collect Model Feedback

A Clinician provides feedback on the performance of an AI Model in routine usage.

Typically, the feedback will identify cases where the AI Model did not perform adequately, however cases with good performance might also be collected, particularly as part of broadening evidence for a wider scope of use or during surveillance (see External Validation Use Cases: Surveil Model).

AI Results can be flagged by clinicians when issues are detected for specific cases. When these events occur, information should be captured and communicated not only to those supporting the AI Application, but also to informatics / quality personnel – and potentially feed into datasets for future re-training of a model. (see Feedback Use Cases: Circulate Feedback)

Considerations that may have technical interoperability aspects include:

- Why feedback on an AI Application is being collected.
    - Clinician reviews AI Results on existing data in repository, e.g., after an algorithm update
    - Clinicians reviews AI Results on clinical cases not in the repository and accepts/rejects/modifies results and wants to add case to repository.

    - Automated performance of an AI algorithm on a dataset with known truth after re-training
    - Automated performance of an AI algorithm after Imaging Hardware has changed.
- Possible reasons why an AI Result was incorrect.
    - Patient may have donated an organ and thus is entirely missing a structure.
    - Patient has features that the model is not trained on (disease specific variations, e.g., running AI on a hypertrophic heart)
    - Patient has had surgery to address condition: E.g., pancreatic cancer where surgery was.
    - Patient has devices, lines or tubes that may interfere with AI inference.
- The encoding in which feedback is communicated.
    - Granularity of the feedback (e.g., the model failed on this case vs manually added segmentations and updated measurements
    - Associated feedback with a concrete finding (e.g., on which specific lung nodule did the model fail)
    - SR encoding to the IHE AIR Profile for the appropriate format to capture this feedback.
    - Encoding the "correct result" the same way as the rest of the annotation data elements in the training datasets would facilitate incorporation in future training or validation.
        - See Common Mechanics: Annotation
- Provenance information to capture.
    - References to specific model (and model version)
    - See Common Mechanics: Annotation
    - See Common Mechanics: Provenance
- Criteria in which to suspend usage of an AI Model.
    - See External Validation Use Cases: Surveil Model

Profile writers may consider some of these questions for further discussion:

- How does the "model used" get captured?
    - What if the "model" can't be distinguished – how to handle where the result comes from a cascade of models and you don't know where the "problem" occurred? The intermediate results should be captured.
- Do we differentiate between "real-time" versus "after the fact" (audit)?

- How does this relate post-market surveillance?
- How does this relate to peer review / over reads / double reads? Can we use similar process patterns?
- Do we differentiate between providing feedback to the local instance of the model, versus back to the original instance author?
- Consider QA / pre-filtering of the feedback – adjudication of feedback, discrepancy resolution between two readers and AI.
    - Consider root-cause analysis (e.g., AI was not expecting to see a chest tube and thus was misinterpreted; or the AI Application reacted poorly to burnt in demographics) and make theories and gather corresponding additional data to "challenge the AI Application".
- How can the model creator inform the feedback source what details should be collected (Consider the fundamental inputs and outputs determined during Model Creation Use Cases: Define Model Task)?
    - This "data completeness" of a training record overlaps the specification of the repository.
    - Can reports be mined and measured against what the model predicts?
        - This could be "structured" checks – e.g., taking measurements from radiologist-generated DICOM SR and comparing against the generated AI Results in DICOM SR.
        - This could be "unstructured" checks with NLP – e.g., understanding the report and comparing against AI classification (e.g., AI reports presence of pneumonia, but the final report makes no mention of pneumonia)
- Also consider tracking and reporting model performance data that is not going to be used for re-training.
    - Link this to payor assessment / regulatory assessment / site assessment / vendor assessment
        - What kind of performance details would we collect for each of these? (sensitivity / specificity / DICE / statistics on measurements, etc.)
    - See External Validation Use Cases

3.7.2 Circulate Feedback

A Data Scientist communicates feedback to the model developers.

Considerations that may have technical interoperability aspects include:

- Feedback payload

  - **Different levels of feedback**: Notification of failure to suspend model usage, corrected findings to re-train algorithm.
  - **Included data records alongside feedback**: ranging from feedback only, all the way to the complete study with findings as created by the AI Application and corrected data elements.
  - Data normalization is needed prior to circulating feedback.
  - When feedback is being provided outside the organization, some elements may need to be de-identified (see Common Mechanics: De-identification).
- Feedback bundling
  - Whether feedback submission is occurring on a case-by-case basis or as batch submission
- Feedback alerting thresholds.
  - If too many studies have failed going through an AI Model, it may trigger alerts or alarms to stop using them until staff can review the model, or the model is re-trained.
  - This might not be needed if doing feedback inside the system that has the model and doing continuous re-training.
- Storage of feedback
  - If the feedback includes corresponding data records, it may sometimes be useful to contribute those data records to an appropriate repository (see Repository Use Cases: Contribute Data) in addition to providing them to the model developers.
  - This links back to “Contributing Data to a Repository” above, including ingestion, QA, record cleanup.
- Impact of patient consent for feedback payloads
- An AI Performer is notified that their model is failing and thus should stop performing inference for an interim period. It could:
  - Unsubscribe from job notifications from the AI Orchestrator.
  - Mark “suspect” or “confirm manually” on jobs it continues to produce.
  - Send additional objects to Image Managers it already processed in the last x hours of possible problems.
  - It could notify IT operations via pager of problems.
    - It could flag that it needs to be retrained at the next training interval.
- Push vs pull feedback mechanisms.
- Billing and licensing implications

  - If AI Results are disputed, this constitutes feedback but may impact billing.
  - Efforts for providing feedback (positive or negative) may be a creditable action.

Profile writers may consider some of these questions for further discussion:

- What is the mechanism to inform the model developers about negative performance?
- Which repository is the recipient of the feedback?
- How does the feedback get conveyed from the point where it is created to where it will be used for re-training?
  - Does the imaging study get captured and stored back, or just the "yes/no" feedback? Is there an in-between (e.g., maybe the model only cares about the "with contrast series" and that is the only thing that we captured?)
- In addition to circulating feedback to the model creator, should it also be circulated relevant repositories? Whose repositories (e.g., a local repository? A remote repository? A temporary repository until remote data scientists has a chance to review it?)
  - When feedback is going external, it likely needs to be de-identified (see Common Mechanics: De-identification).
  - How to encode and communicate details that are important to maintain?
  - If feedback is not being shared externally, should there be a feedback availability notification message?
- Does collection and circulation of feedback overlap with details collected in IHE Teaching Files and Clinical Trials Export (TCE) Profile [STD-TCE]?

3.7.3 Adjust Model

A Data Scientist incorporates the feedback into data records in a training dataset and retrains the model.

Data records that are flagged as diverging from the AI Model can be added to a Dataset (see Dataset Assembly Use Cases: Obtain Data Record), with ground truth data elements identified, that can be sent back to the Obtain Data Records step and a decision might be made at the Organize Datasets step whether to use the feedback data record into training, testing or validation. Models can then be fine-tuned or re-trained as described in Model Creation Use Cases: Train Model.

Considerations that may have technical interoperability aspects include:

- Real-time updates to the model
  - This is generally referred to as active (or continuous) learning and occurs when data records are immediately fed into a re-training or fine-tuning step.

  - A data model is re-trained (to a larger or smaller degree) based on the result of new feedback.
  - Care needs to be taken in clinical systems, as changing models may have impacts on performed external validation, such as by regulators or users.
- Clinical re-use post-model adjustment
  - Leveraging implicit and explicit feedback
  - Impacts from a regulatory, payor, and clinician external validation.
- Feedback as additional ground truth:
  - In Dataset Assembly Use Cases: Refine Data Record and Annotate Data Record, feedback payloads may look different than originally provided ground truth. It is up to the data scientists and model training process to reconcile how to interpret ground truth.
- Model provenance as feedback is created.
  - See Common Mechanics: Provenance

Profile writers may consider some of these questions for further discussion:

- Would requests to update models from Feedback Workflow be triggered via UPS-RS?
- Is there a difference “when we adjust the model” if it was human captured?
  - Who is controlling in which step, that the feedback is valid?
  - Is the provided feedback/case suited for re-training of the used model?

3.8 Common Mechanics

This section describes capabilities that are needed in multiple use case sections above. Because the specifics depend on the “parent” use case, the initial descriptive sentence will use the passive voice, rather than the active voice used above.

When addressing these “sub-use cases” in profiles, it will be important to consider the benefits of using common mechanics for these capabilities. Doing so effectively and designing a robust mechanism will depend on recognizing the multiple use cases to which the sub-use case applies and the different ways a selected mechanism would be used. The various applications can be found by searching the Use Cases sections for the name of the Common Mechanics section since they have been referenced as appropriate.

3.8.1 Data Access

Data is located and retrieved to populate data elements.

Often, this data exists in operational healthcare systems such as EMRs, PACS, VNAs, and laboratory information systems. Due to the breadth of sources and types of data, a variety of data access protocols will likely be needed. At the same time, encouraging some convergence where practical will reduce the cost of implementation and potential interoperability issues.

Considerations that may have technical interoperability aspects include:

- Examples of data elements (and some standards the source might support for access) include:
    - Imaging
        - DICOM DIMSE [STD-DICOM]
        - DICOMweb [STD-DICOMWEB]
    - Patient medical records and demographics
        - HL7 v2 [STD-HL7] ADT and ORU
        - FHIR Patient [STD-FHIR-PT] and Observation [STD-FHIR-OBS]
    - Schedules (timebound) of entities/resources, such as scanners, rooms, staff, and patients
        - FHIR Appointment [STD-FHIR-APPT] and Schedule [STD-FHIR-SCHED]
    - Worklists (order and priority bound) such as reading and processing worklists.
        - DICOM MWL [STD-DICOM-MWL]
        - DICOMweb UPS-RS [STD-DICOMWEB-UPSRS]
    - Orders
        - HL7 v2 [STD-HL7] ORM
        - FHIR ServiceRequest [STD-FHIR-SVCREQ]
        - DICOM MWL [STD-DICOM-MWL]
    - Performed procedures.
        - FHIR Procedure [STD-FHIR-PROC]
        - HL7v2 [STD-HL7] OMG
        - DICOM MPPS [STD-DICOM-MPPS]
    - Patient Diagnoses and Conditions
        - HL7 v2 [STD-HL7] ADT and ORU
        - FHIR Condition [STD-FHIR-COND]
        - ICD [STD-ICD]
        - SNOMED [STD-SNOMED]
        - LOINC [STD-LOINC]

    - Appropriateness criteria: e.g., updated regularly from authoritative source to inference products.
        - Lung-RADS [STD-ACR-LUNG], BI-RADS [STD-ACR-BIRAD], etc.
    - Appropriateness recommendations
        - IHE CDS-OAT [STD-CDS-OAT]
    - Practice guidelines / clinical pathways
    - Traffic and weather conditions / forecasts
        - e.g., impact to no show prediction, icy conditions for increases to ER admissions.
    - Taxi ordering and availability for patient transport.
    - Anatomy atlas
        - FMA [STD-FMA]
    - Care team members, with credential / role / job and contact information.
        - IHE DCTM [STD-DCTM]
    - Billing transactions
        - CPT [STD-CPT]
            - Procedure costs (including costs for running inferences)
- Self-description of data
    - E.g., when extensions are needed, or transforms are performed.

Profile writers may consider some of these questions for further discussion:

- Should access/retrieval also address query/discovery, or is that better handled as a separate service?
- In each use case, will systems need to support push or pull or both.
- When using data, does the user keep a persistent copy (raising synchronization questions) or does it discard and re-retrieve if/when needed (raising versioning questions)
- Does a data record incorporate data element values or include data elements by reference?

3.8.2 Exchanging Datasets

The content of an entire dataset is accessed or moved from one location to another.

Considerations that may have technical interoperability aspects include:

- Encoding of the data elements

- Encoding of data records (bundle of elements or vector of references to elements)
- Packaging and bundling of datasets.
- Protocols for exchanging datasets.
    - At the time of writing, DICOM is developing standards for inventory and migration of DICOM content. See [BIB3.8.2-1].
    - IHE XDS [STD-XDS] may also be appropriate.
- In each use case, will systems need to support push or pull or both.
- Protocols used for accessing individual data elements or records should be adequate for accessing large datasets.
- When using data, does the user keep a persistent copy (raising synchronization questions) or does it discard and re-retrieve if/when needed (raising versioning questions)

Profile writers may consider some of these questions for further discussion:

- Does a dataset incorporate data records or include data records by reference?
- At what point does the size of the dataset raise new considerations?
    - Is there a need to exchange/access part of a dataset?
    - Should we consider some but not all the data records? Which end controls which records are included in each part?
    - Should we consider some but not all the data elements? Which end controls which elements are included in each part?
    - Should we consider both some of the elements for some of the records?
- Should access/retrieval also address query/discovery, or is that better handled as a separate service?

3.8.3 De-identification

Identifying information is modified or removed from data elements, data records or datasets.

The data used in AI is often data from real patients. To protect the identity and privacy of the patients, information which could expose their identity to individuals outside their chain of care needs to be processed so that is no longer reasonably possible. The words anonymization and pseudonymization are sometimes also associated with this process.

Considerations that may have technical interoperability aspects include:

- Determine what de-identification is needed based on where the data is being sent and for what purpose.

  - In clinical use inside a hospital, patient data remains identified. If using onsite AI Models, there might be no need for de-identification.
  - Arrangements like HIPAA BAAs (Business Associate Agreements) engage a hospital partner in the responsibility to protect identified patient data and might not require de-identification.
  - If the security of the data sent to an external service cannot be protected, or to increase the level of security, the data might be de-identified before sending to the external service.
  - Training and validation of AI Models is typically outside of the individual care of specific patients and so de-identification is generally the rule.
  - Data being stored and published in repositories is expected to be de-identified.
- Determining whether certain details should be retained.
- Handling the need to coordinate de-identification across multiple data elements or data records.
  - E.g., by using the same artificial patient ID for a given patient so that a pre-treatment record and a post-treatment record can be linked.
- Considering whether there is a need to later re-identify the patient.
  - E.g., by maintaining a protected lookup table so that a patient can be notified if an unexpected health problem is uncovered during use of the data.
- De-identification should be performed in a deliberate fashion, ideally based on relevant specifications.
  - DICOM Basic Application-Level Confidentiality Profile [STD-DICOM-DEIDENT]
    - Using associated options such as retaining patient characteristics or retaining longitudinal temporal information.
  - IHE White Paper analysis [BIB3.8.3-1]
  - Health Insurance Portability and Accountability Act (HIPAA) [BIB3.8.3-2]
  - General Data Protection Regulation (GDPR) [BIB3.8.3-3]
- Open-source and commercial tools exist to support de-identification.
  - Some providers have imaging datasets with artificial identities that can be used to test/evaluate de-identification tools and processes against their recommended results.
  - An AI Model could be trained to obfuscate burnt-in demographics.
  - See Appendix C.4: Reference Toolkits.

Profile writers may consider some of these questions for further discussion:

- Methods to preserve elements that are generally de-identified as they may be required for the operation of AI Applications.
    - E.g., analysis may depend on patient age, or the relative time between two sets of images.
    - Balancing these needs may involve removing the birthdate but leaving the approximate age intact or modifying the study dates to obscure the actual date the patient encounters occurred but maintain the relative times between them.

3.8.4 Annotation

Data records are supplemented with data elements. Typically, these represent the direct result that an AI Model is intended to learn to generate, or additional observations that may help an AI Model distinguish cases. Because of the obvious symmetries between what the AI Model is intended to generate (Annotations of ground truth for learning) and what the AI Model generates during clinical use (Results), much of this section also applies to Results. Similarly, since Feedback (see Feedback Use Cases: Collect Model Feedback and Entities: Feedback) often incorporates an annotation representing a corrected result, much of this section also applies to Feedback.

In some cases, annotations represent the definitive correct answer or "ground truth"; in other cases, annotations represent the best effort of a human to determine the answer. The human might make use of preliminary inference results to assist them in doing this.

Considerations that may have technical interoperability aspects include:

- Types of things that are the subject of annotations.
    - Ground truth annotations represent the correct result of the task to which the AI Model is being applied. See 2.1 Applications of AI in Imaging for a broad set of examples.
        - Popular examples of ground truth data elements for image analysis tasks (see Applications of AI in Imaging: Image Analysis) include determining presence or absence of pathologies, categorizations, organ segmentations, image quality, patient positioning, presence of devices like chest tube, etc.
        - A chest x-ray image may have one or more labels, e.g., presence of pneumonia, presence of pneumothorax, presence of COPD, presence of scarring, etc.
        - A mammography study may have one class of breast composition, according to BI-RADS (https://www.acr.org/-/media/ACR/Files/RADS/BI-RADS/Mammography-Reporting.pdf), and a list of lesions with BI-RADS classifications 1-5 and additional descriptors for morphology and texture of the lesion.

        - A CT of the abdomen may have a segmentation identifying liver tissue, liver lesions, and non-liver tissue.
    - Observations or assessments of the input data elements in the data record
        - E.g., whether the image quality is adequate and how good it is, how the patient is positioned and whether it is adequate for the task, whether devices like chest tubes that might confound image analysis are present, etc.)
    - Additional input data elements
        - E.g., demographics, laboratory values or coded report findings are not ground truth annotations per se, but rather may be considered additional data elements as described in Repository Use Cases: Curate Repository.
- Form of annotations (depends on the type of information)
    - Coded Labels can be used for a wide variety of observations such as identifying pathologies, anatomy, objects, and characteristics.
        - Use standard codes (RadLex, LOINC, SNOMED, ICD-10, FMA)
        - Decide on, and consistently use, a level of granularity that is appropriate to the task result (e.g., encoding anatomy as temporal lobe->cerebral cortex -> cerebrum -> brain -> head)
    - Segmentations indicate the spatial extent of a target.
    - Bounding Boxes indicate the rough spatial location and extent of a target.
    - Spatial Measurements (area, volume, angle)
    - IHE AI Results Profile [STD-AIR] proposes “primitives” that correspond to several forms of annotations for Image Analysis applications.
- Generating annotation data elements
    - Humans who are expert performers of the task the AI Application is intended to perform are a common source.
    - Having multiple experts annotate the same data record can be a way to increase the quality of the ground truth when the intra-observer variability is greater than desired.
        - Duplicate annotations could be aggregated statistically.
        - A discrepancy resolution method such as those used for radiology over-reads could be applied.
        - An algorithm could be used like requiring 3 observers to agree before the corresponding data record is added to the dataset.
- Assessing the quality of ground truth

    - Arguably, the quality of the ground truth data elements has more of an impact on the quality of the resulting model than the quality of some other data elements
    - One practice is to have a second expert review the first experts' ground truth (rather than submitting a second blinded annotation)
- Information that might be relevant to record in an annotation data element:
    - The identity of the observer that generated the annotation.
    - Indications of the expertise/skill of the observer (e.g., formal qualifications or credentials)
    - Tools used by the observer (in case the tool induces a bias or quality issue that skews the data)
    - Whether the annotation is based on the observer applying specific standard criteria (e.g., BIRADs score criteria), local criteria, or the observer's personal judgement.
    - Confidence or quality of ground truth (e.g., absolute versus confidence scores, level of ambiguity or hedging)
    - Considerations for that ground truth (causes that support ground truth or additional observations)
    - Observations about the input data
    - See also Common Mechanics: Provenance
- Encoding annotation data elements
    - Encoding annotation data elements similarly to other data elements reduces complexity and increases compatibility.
    - IHE AI Results Profile [STD-AIR] proposes encodings for several annotation data elements of Image Analysis applications. The proposed encodings (defined in Section 6.5.3 of AIR) include:
        - Comprehensive 3D SR Storage [STD-DICOM-SR] stores:
            - qualitative image findings in form of coded concepts such as presence or absence of conditions,
            - measurements, locations, ROI's, and tracking identifiers of image findings,
        - Segmentation Storage [STD-DICOM-SEG] stores voxel-based spatial segmentations and the associated labels
        - Parametric Map Storage [STD-DICOM-PM] stores floating point pixel images that encode things like saliency heat maps.
        - Key Object Selection Document Storage [STD-DICOM-KOS] is a type of SR that applies a label to a list of images or other DICOM objects.

- Other DICOM encodings of potential interest include:
    - RT Structure Set Storage [STD-DICOM-RTSS] for contour segmentations (of organs and biopsy or radiotherapy targets)
    - Surface Segmentation Storage [STD-DICOM-SSEG] stores segmentations as a polygonal surface mesh
    - Secondary Capture Image Storage (and its multi-frame and color variants) [STD-DICOM-SECCAP] are “screenshots”. Although readily displayed, this lowest common denominator lacks spatial information and is not machine readable. As such, its use is discouraged, except as an adjunct display for information already encoded in a proper format.
    - Grayscale Softcopy Presentation State Storage (“GSPS” and its color variants) [STD-DICOM-GSPS] store a selected presentation of one or more images and can specify graphic and text overlays. As with Secondary Capture, it is widely supported by PACS, but is not designed as a method of storing primary data so it is best limited to use as an adjunct display.
- DICOM Supplement 219 [STD-DICOM-SUP219] specifies a simplified JSON encoding of SR objects that is fully equivalent/transcodable. It is intended to facilitate easy processing by data scientists who are more familiar with JSON.
- HL7 FHIR [STD-FHIR] includes a variety of Resources that will likely be increasingly supported by deployed HIT systems. Those resources can be relevant sources and sinks for inference task data elements and annotations and data for training those AI Models. A few resources of interest include:
    - FHIR Observation [STD-FHIR-OBS] captures clinical observations and will likely become the preferred format for things like lab results, vital signs, clinical assessments, and readings from monitoring devices.
    - FHIR DiagnosticReport [STD-FHIR-DXR] aggregates multiple related observations.
    - FHIR Patient [STD-FHIR-PAT] captures patient identity and demographics like sex and age.
    - FHIR Encounter [STD-FHIR-ENC] captures information about an encounter between the patient and a healthcare provider like the reason for admission and the department being visited.
    - FHIR Procedure captures information about a procedure that has been, or will be, performed on the patient.
- HL7 V2 Message Segments [STD-HL7-V2] are currently the most deployed encoding for a variety of observation and operational data.

    - Machine readable encodings are generally preferable so annotations can be used in training and automated testing/validation. Non-machine-readable encodings (e.g., secondary capture), might be usable for human interpreted testing and validation to compare results visually.
    - For some annotations there may multiple roughly equivalent encodings that are suitable. Failure to converge on standard encodings can create a significant obstacle to data sharing and interoperability. A transcoding that is accurate, robust, and automated can mitigate some of the difficulties.
        - E.g., segmentations may be encoded as a polygon surface, contours on slices, RT structure sets, pixel/voxel masks, etc.
    - Annotations are informally recorded in some research groups using textual labels and NIFTI (an Analyze-style data format)
- Transporting annotation data elements
    - Annotation data elements are often simply transported as part of the data record/dataset to which they belong.
        - See Common Mechanics: Exchanging Datasets
    - Annotation activities may be performed outside the organization hosting/managing the data records in which case they need to be conveyed back.
        - See Common Mechanics: Data Access for some possibilities.

3.8.5 Transforms

A data element is produced by algorithmically processing one or more other data elements.

Transforms are typically performed to make a data element that has been provided by a source acceptable to the needs or preferences of the destination. Transforms are sometimes referred to as “pre-processing”.

Transforms are used during Repository curation to normalize data and create data elements that meet the repository design. Transforms are used during Dataset Assembly to again normalize data and create data elements that meet the dataset design. Transforms are used during Model Creation to present data elements to the AI Model in a form it can process, and potentially to enrich the training dataset by creating records that are variants of other records. Transforms are used during Clinical Usage to put AI Model outputs into a form desired by the HIT systems that will use it.

Considerations that may have technical interoperability aspects include:

- Type of transforms, e.g.:
    - Transcoding (e.g., rendering a DICOM image as a PNG file, or rendering an inference result as a DICOM SR)

    - Extracting (e.g., taking key parts from an HL7 message and creating a CSV file of pertinent values, creating data elements from fields in the DICOM header, extracting an SR coded concept into a FHIR Observation)
    - Cropping, rotating, and/or flipping an image
    - Adjusting the window level/center
    - Resampling (e.g., turning a 1024x1024 mammogram into a 256x256 image consumable by an AI engine, or extracting multiple 256x256 files for processing)
    - Converting a set of frames into a three-dimensional matrix
- Where the transform logic is used:
    - Inside the AI Application package
    - As part of the ingestion pipeline
    - As part of a DICOM router, such as MIRTH
- Lossy vs lossless
    - Some data loss may occur on transformed objects, and thus, care should be taken to protect the original data or be able to source it again when needed.
    - Source data may be needed after inference has occurred to relate the results, for example, of a segmentation (where both the original image and the segmentation mask is present). as well as for inference; but could occur.
- Generating synthetic data
    - Transforms may be utilized to create additional synthesized datasets.
    - E.g., taking an image with a ground truth and rotating it one degree 360 times.
- Provenance of transforms
    - Providing the list of transforms performed and their results as part of processing pipelines may aid in troubleshooting, and in some cases, may need to be part of the clinical record.
    - See Common Mechanics: Provenance.
    - There may be differences in mathematical calculations for certain transform operations that may impact AI operations.
    - E.g., when shrinking images, a transform from Library 1 and a transform from Library 2 might average pixels differently, which may have a negative effect on the outcomes.
    - See [BIB3.8.5-1] for an example of flags.

Profile writers may consider some of these questions for further discussion:

- How does this apply to different classes (imaging, labs, visit data, diagnostic reports, demographics, medical records, etc.)?

3.8.6 Data Quality

The content of an output is assessed for correctness.

Considerations that may have technical interoperability aspects include:

- Type of entity
    - A repository of high quality should contain data records that align with the stated repository purpose, contain higher resolution images and more complete data records, with annotated ground truth made by and verified by trained annotators, and use strongly typed ontologies for representing codable concepts.
    - A dataset of high quality should contain a balanced dataset that supports the defined purpose, with annotations that align with the desired end-goals of the data science team.
    - A model of high quality should support the stated intent and purpose, score highly on internal validation sets as well as when externally tested with independently created data.
    - Feedback of high quality should be specific, generated by a trained clinician, and provide enough detail on how the result was correct or incorrect so that models can be further fine-tuned.
- How quality is conveyed
    - Scoring (Dice scores, Area under the curve, etc.)
    - Aggregating feedback from experts
    - Human or computational assessments
- Leveraging quality data
    - How to indicate repositories / datasets / models / feedback of questionable quality

Profile writers may consider some of these questions for further discussion:

- How do characterize and communicate the quality?
    - Of a dataset, of annotators (and skill level), of a model, …
- Diversity and bias of the dataset
    - Sometimes this depends on the specific task.
    - Factors in the assessment of diversity

3.8.7 Provenance

The origin and history of an identifiable artifact is recorded and communicated.

Artifacts include AI Models, datasets, data records and possibly individual data elements.

Being able to manage and trust artifacts depends on being able to identify and differentiate them, recognize different versions, and understand how they were created.

Considerations that may have technical interoperability aspects include:

- Uniquely identifying an artifact
  - Defining (for each class of artifacts) what changes require assigning a new identity.
- Describing how the artifact was created.
  - For data (elements, records, sets), distinguish between "real data" versus artificial/derived/synthetic data, e.g., generated by enrichment processes.
- Detecting and describing changes to the artifact
- Describing relationships between an artifact and its predecessors
- Recording the change "log"
- Version control
- Tamper proofing
  - A digital entity is protected from undetected modification. The digital entities may include AI Models, datasets, data records, or even individual data elements. Controlling access to all elements both at rest and in-flight is key to developing healthcare systems.
  - The security of other data elements in the repository and available through the interface needs to be factored into the repository creation as well.

Profile writers may consider some of these questions for further discussion:

- How are entities traced to their source?
- Could blockchain be used to convey provenance?
- How is duplication (where two artifacts that are identical have different identifiers) detected and resolved (de-duplication)?

3.8.8 Exchanging Models

The information necessary to replicate the operation of an AI Model is accessed or moved from one location to another.

Considerations that may have technical interoperability aspects include:

- Shareable Model format
  - See Appendix C.4 for examples.
- If a dominant format for AI models is identified and if image archives would be a useful infrastructure to store/distribute AI models, a DICOM wrapper could be defined (as was done for 3D Print Files).
- Binary format for loading into binary applications to make the fastest and smallest applications.
  - See Appendix C.4 for examples.
  - Additional information is captured in "model manifest" to help others understand how to use the model.

Profile writers may consider some of these questions for further discussion:

-

3.8.9 Model Performance Metrics

A quantitative estimate of the ability of the model to perform its intended task is derived, recorded, and communicated.

Accurate assessment of model performance is critical both prior to clinical deployment (for regulatory and evaluative purposes) but also post-deployment to ensure continued functionality. Relevant metrics of assessment are often a function of both problem type (classification, segmentation, etc.) as well as the clinical use case being addressed. Reported metrics should allow a stakeholder to confirm continued model efficacy as well as improvements in the clinical workflow. In many circumstances, it may be difficult to effectively summarize model performance in a single metric. with site-specific measures, thus necessitating the persistence of individual, study-level performance for post-hoc metric computation.

Considerations that may have technical interoperability aspects include:

- Metrics will vary based on the aim of the task:
  - *Classification* – e.g., sensitivity, specificity, accuracy, precision, F1, etc.
  - *Object Detection* – e.g., precision, recall, mean average precision, etc.
  - *Segmentation* – e.g., Dice coefficient, intersection over union, etc.
- Measuring a model's performance, e.g., using:
  - Accuracy
  - Sensitivity or recall and specificity [BIB3.8.9-1]
  - Precision
  - Area under the curve (AUC)

- Metrics apply to a specific “version” of an AI Model.
    - See Common Mechanics: Provenance
- Metric values themselves may sometimes (e.g., when being used for regulatory validation) need to establish provenance in terms of the version and details of the assessment procedure and the dataset used to generate the metric.
- In the FDA AI/ML- based SaMD (Software as a Medical Device) Action Plan there are regulatory oversights identified that once enacted, must be adhered to
- Metrics may affect which Application is invoked for inference, e.g., Application A performs better on male patients and Application B better on females.

Profile writers may consider some of these questions for further discussion:

- Where will logging be done and who will have access?

3.8.10 Sequestration

Access to certain data elements, data records, and/or datasets is restricted for data independence reasons.

Datasets that have been, or are intended to be, used for a certain purpose may need to be isolated to prevent their use for a potentially conflicting purpose. For example, a validation dataset would be biased if it contained data records that had been used for training the model being validated.

Individuals assembling datasets for training, testing, or validation should consider a strategy for how they will avoid “cross-contamination” of their datasets.

Sequestration involves designating data records for a certain purpose and controlling access to them, so those records are not used for other purposes.

Considerations that may have technical interoperability aspects include:

- Each data element that is used as part of a dataset must be accounted for based on that specific use case and point in time. An inventory of data elements must be part of the dataset.
- Any data elements added to a dataset after its initial creation needs to be specified and the version update.
- Provenance is an important tool for sequestration since it can support reliably tracking the identity of data records and their source and chain of custody, which would help detect if the same data record has been used in different datasets for conflicting purposes.
- Private and federated repositories must have mechanisms to ensure data elements only appear once in the model development lifecycle. This can become a challenge when data is obtained from multiple sources and has gone through de-identification.

- Often the sequestered data may be from different physical sites to test if models can be generalized to new sites.
- Repositories must ensure access to sequestered data is only allowed for appropriate use cases. For regulatory purposes, this same data may be used to validate multiple models to ensure consistency.

Profile writers may consider some of these questions for further discussion:

- Should a single repository host datasets for different purposes such as training, testing and Validation?
    - Should validation data ever be completely deidentified since UIDs cannot be confirmed as not having been used for training?

4 Entities

Certain entities appear in multiple use cases. This section provides a starting point for scoping what these entities represent and what associated metadata would need to be captured. Ultimately, such work will be done during profile development, so this section is not intended to be definitive or all-inclusive.

It is intended that the attributes in the entity metadata tables are driven by the implied and explicit requirements of all the use cases that refer to these entities; however, this work has not yet been completed. Input is welcome.

4.1 Data Element

The data element entity represents a data element and its associated metadata.

A **data element** is an individual piece of data.

The primary use of data elements is to represent the **inputs and outputs of an inference task**. Training, testing, and validation data records may also contain supportive data elements that are neither passed as inputs and are not outputs but are needed for administrative tasks. For example, the patient race might not be passed to the AI model as an inference input but might be maintained in the data record to support the assessment and avoidance of racial bias in the development of the AI model.

The **granularity** of data elements may vary. E.g., a data element may be an individual image (DICOM or PNG), the patient age, a reason for admission, a segmentation (contours or RTSTRUCT), a label (Condition Present), a medication, or a lab value (creatinine level).

Some types of data elements are often referred to by specific terms.

- When a human performs the task that the AI Model is intended to learn to perform, the human will generate the “correct” output data element. This process is referred to as **Annotation** (see Common Mechanics: Annotation, and Entities: Annotation). Annotation might also be performed by a different piece of software. Annotation might also be performed to generate input data elements that are missing in a data record.
- When the AI Model performs an inference on a data record in actual use, the output data element is referred to as a **Result**. See Clinical Usage Use Cases: Create Result.

Data elements often pass through several **processing steps** in the course of being created, contributed, cleaned, shared, refined, and used.

- Data elements may be de-identified at various stages (see Common Mechanics: De-identification)
- Data elements may be transformed to change the encoding, or extract smaller data elements from a larger composite entity (see Common Mechanics: Transform)

  - Transform processes may occur during the original clinical usage, during contribution, cleaning, curation, annotation, and sharing of the repository Dataset, during obtaining, refining, annotating, organizing, and sharing of the training, testing or validation dataset, and just before passing the data element to the AI Model during training, testing, validation, or clinical use.

- In principle, the above could be captured in the provenance details for the data element and the data record but doing so is often challenging and impractical.

Table 4.1-1: Data Element Metadata

Attribute	Description
identity	Uniquely identifies the data element.
provenance	A record of: • the origin of the data element (where it came from, who/what created it, the quality/skill of the creator, etc.) • changes to the data element (transforms, normalizations, de-identifications, assessments, etc.). Further modelling to map out a common data structure for provenance details has not yet been done. See Common Mechanics: Provenance for further details.
format	The encoding of this data element. Note: in a uniform dataset, this is the same for all occurrences of the data element so it could be coded at the data record or dataset level.
content	The data that comprises the data element. The granularity may be elemental, or compound as noted above.

The **format/representation** of data elements will vary widely.

- This is due in part to the wide variety of applications (see Applications of AI in Imaging) and the corresponding variety of information that data elements represent: images, demographics, device characteristics, clinical observations, billing details, workflow details, diagnoses, procedures, spatial segmentations, measurements, scan protocols, classifications, etc.
- This is also due to the variety of encoding formats available for the various types of information. Such differences represent an interoperability challenge. Some of these may be addressed with Transforms (see Common Mechanics: Transform). Profiling will also attempt to encourage convergence on an effective subset of the possible formats.

Annotations of images are of significant interest in this whitepaper, both as input data elements and output data elements.

Profile writers may consider some of these questions on Result Data Elements for further discussion:

-

4.2 Data Record

The data record entity represents a collection of data elements and its associated metadata.

A data record is a set of related data elements. The relationship is often that they are all associated with the same patient encounter, but this can vary. E.g., a data record might consist of an x-ray, a lab result, and a reason for admission, all associated with a given patient encounter.

A training data record may identify which of the data elements are input data elements for the inference and which are the “result” output data elements, or that may be configured into the training engine. See the “role” attribute in Table 4.2-1.

A simple data record might contain two data elements, an input image (e.g., a chest x-ray image) and an output text label (e.g., pneumonia). A data record can include data from different time points, such as a current image and a corresponding prior image from 6 months earlier. Some AI Applications may make use of data elements beyond those in the data record provided to them, for example by searching available data sources for additional data elements.

Table 4.2-1: Data Record Metadata

Attribute	Description
identity	Uniquely identifies the data record.
provenance	A record of: • the origin of the data record (where it came from, who/what created it, the quality/skill of the creator, etc.) • changes to the data record (addition or deletion of data elements, etc.). Further modelling to map out a common data structure for provenance details has not yet been done. See Common Mechanics: Provenance for further details.
Data Elements	References to the specific Data Elements that constitute this data record.
>role	The role played by this data element in this data record. The role reflects the purpose of the dataset the data record is in. A

	data element that is an output in one dataset, might be an input in another. Input: This data element is an input to the AI Model. Output: This data element is an output of the AI Model. Note: this presumes a "push" dataflow where the environment learns the needs of the AI Model and pushes appropriate data. The alternative "pull" dataflow would tag the concept encoded in each data element at the AI Model would use that to locate the inputs it needs in each data record. Also, since the role would be the same for all data records in a dataset, it probably makes sense to have this at the dataset level rather than each data record.

The FHIR Bundle resource [STD-FHIR-BUNDLE] combines related results into a group and may be a relevant encoding method for data records.

Profile writers may consider some of these questions on Result Data Elements for further discussion:

-

4.3 Dataset

The dataset entity represents a dataset and its associated metadata.

A dataset is a set of related data records. Typically, all the data records in a dataset contain roughly the same data elements. Datasets containing data records with patient data will typically span many patients.

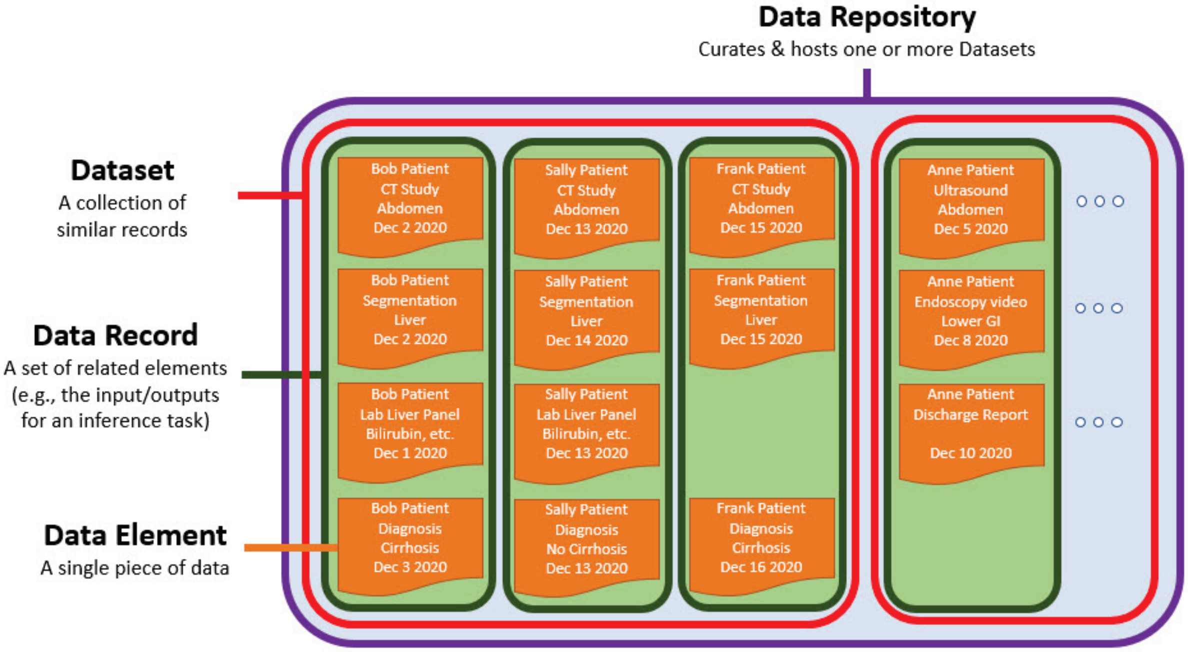
Data Repository
Curates & hosts one or more Datasets
Dataset
A collection of similar records
Data Record
A set of related elements (e.g., the input/outputs for an inference task)
Data Element
A single piece of data
Bob Patient CT Study Abdomen Dec 2 2020
Bob Patient Segmentation Liver Dec 2 2020
Bob Patient Lab Liver Panel Bilirubin, etc. Dec 1 2020
Bob Patient Diagnosis Cirrhosis Dec 3 2020
Sally Patient CT Study Abdomen Dec 13 2020
Sally Patient Segmentation Liver Dec 14 2020
Sally Patient Lab Liver Panel Bilirubin, etc. Dec 13 2020
Sally Patient Diagnosis No Cirrhosis Dec 13 2020
Frank Patient CT Study Abdomen Dec 15 2020
Frank Patient Segmentation Liver Dec 15 2020
Frank Patient Diagnosis Cirrhosis Dec 16 2020
Anne Patient Ultrasound Abdomen Dec 5 2020
Anne Patient Endoscopy video Lower GI Dec 8 2020
Anne Patient Discharge Report Dec 10 2020
○○○
○○○
○○○

Figure 4.3-1: Data Element, Record, and Set Hierarchy

Figure 4.3-1 shows the hierarchical relationship between data elements (orange boxes) and data records (rows of boxes) within a dataset (blue box).

Table 4.3-1 lists some possible metadata attributes for a dataset.

Table 4.3-1: Dataset Metadata

Attribute	Description
identity	A globally unique identifier for the dataset.
provenance	History of the dataset
status	The status of the dataset
title	A human-readable description of the dataset
purpose	Whether the dataset contains training, testing, validation, verification, or simply repository data records
author	Who created the dataset
reviewer	Users who have looked at the dataset to ensure it is of sufficient quality. See Common Mechanics: Data Quality.
organization	Healthcare institution that created the dataset

Attribute	Description
inclusion criteria	Criteria used in include data records in the dataset. Records will all meet these characteristics. See Repository Use Cases: Retrieve Repository Content for further discussion.
exclusion criteria	Criteria used to exclude data records from the dataset. Records with these characteristics will not be present.
data records	References to the data records included in this dataset.
deidentified	Whether all the records in this dataset have been deidentified

Profile writers may consider some of these questions for further discussion:

- Does identity change when the content of the dataset changes or do we have layered versioning?
- How is model "bias" accounted for in the assembly of datasets?
- How can problems be identified and mitigated?
    - E.g., "faulty burnt-in demographics in a dataset which lead to a faulty model."
- How do we organize inclusion and exclusion criteria?
- Are git hashes something we would consider on fixing the dataset in time?
- What is the impact of dataset compilation on model reproducibility?
- Does the size of the dataset need to be conveyed, in number of records or sheer size on disk?
- How are any IRB restrictions conveyed?
- How are data elements and data records of a dataset described in the identity and any other identities they may be part of?

4.4 Data Repository

The repository entity represents a repository and its associated metadata.

A data repository is an infrastructure that hosts one or more datasets for discovery and retrieval.

Table 4.4-1: Data Repository Metadata

Field	Considerations
identity	Globally unique identifier for the repository. It does not change when the datasets change.
title	A human-readable description of the repository
purpose	A description of the intended purpose of the repository
endpoints	Endpoints for the discovery and retrieval use cases
licensing	
terms	Terms of use of the repository
author	The identity and contact information for the Repository Administrator
organization	The organization responsible for the repository
number of datasets	
datasets	UIDs of the datasets contained in the repository

Profile writers may consider some of these questions for further discussion:

-

4.5 Transform

The transform entity represents a transform and its associated metadata.

A transform is a process that produces one or more output data elements from one or more input data elements. See Common Mechanics: Transform.

Table 4.5-1: Transform Metadata

Attribute	Considerations
identity	Globally unique identifier
title	Human readable label
input	What does the transform expect as input
output	What does the transform produce as output
version	Human readable version number
library	Software library performing transform

Profile writers may consider some of these questions for further discussion:

-

4.6 AI Model

The AI Model entity represents an AI Model and its associated metadata.

An AI Model is a package of model weights that has been trained to identify a particular target in a supplied input.

Table 4.6-1: AI Model Metadata

Attribute	Description
identity	Globally unique identifier
title	Human readable label
status	Draft, published, validated, clinical use, bad
body site	Part of body this model applies to (SNOMED? FMA?)
condition	The type of finding this model aims to classify / segment
intent	The type of model (classification, segmentation, measurement)
contraindication	When should this model not be used
target	Other target constraints (e.g., run only for a specific clinician or group)
input	What does the model expect as input
input transforms	The types of transforms expected of data before processing
output	What does the model produce as output
parent model	Parent model this was derived from
created	Author / institution / date
contributor	Sources of training data
data	References to source training data
score	How the model performed with validation data. See Common Mechanics: Data Quality.

Profile writers may consider some of these questions for further discussion:

- How are diversity zones of data conveyed and summarized?

  - E.g., types of patients, types of equipment

- What other aspects of provenance need to be considered?

4.7 AI Application

The AI Application entity represents an AI Application and its associated metadata.

An AI Application is a package of components to make an entire application, which could include model weights, algorithms, and input/output transforms.

Table 4.7-1: AI Application Metadata

Attribute	Description
identity	Globally unique identifier
title	Human readable label
status	Draft, published, validated, clinical use, bad
model	Model(s) packaged in the application (note: the metadata exposed through the model is likely important; body site, condition, etc.)
input	Data elements the model expects as input
input transforms	References to the types of transforms that may need to be applied expected of data before processing
output	Data elements the model produces as output
run conditions	Environment in which model should be run in (minimum requirements)
created	Author / institution / date

4.7.1 Types of Packages

There are two main types of application packages:

- **Data-only packages**: which just include the AI Model with layers, weights, and offset biases, but that cannot be run by itself without further integration work, and
- **Executable packages**: that are self-contained executable "containers".

"Data-only" Model Package

The data scientist saves all model data, i.e., network structure and weights.

Sharing a package that contains just the AI Model in a common format makes most sense in a research environment as it allows the receiver to run, improve or re-define the model in their own development environment while using their own tools. In this case the data scientist would download the model from a repository, extract the deep learning network along with all its parameters, convert and copy it into the machine learning runtime framework of their choice (see [BIB3.4.1]), and then implement the functions to transform their input data into the required model-specific format, to finally run inferences on the model.

Such information may include:

Table 4.7-2: Data-Only Metadata

Attribute	Description
identity	Model ID and author's email – a globally unique ID used to identify this model and trace it back to the author.
version	Model Version number
status	The status of the model, e.g., ready-for-validation, validated, cleared-for-use (by whom), or deprecated
purpose	What is the purpose of the model, i.e., what problem does it solve? What are its indications for use, and what regulatory clearances have been obtained for it?
types of data	Description of the types of data that the model was evaluated on (modality, body part, scan protocols, use of contrast agent) and therefore can be used on
input / output	Required input and resulting output data format, e.g., different types of DICOM objects, along with a description of necessary pre- and post- transforms (e.g., scaling, flips, rotation, removal of metadata inappropriate for training runs like laterality), as well as what coded concepts are provided using which medical dictionary
performance	Model performance with reference to a test data set, or better include the test data with this package
provenance	Information about the model provenance, e.g., the internal architecture of the model, including what standard models it is based on

The resulting data-only model package consists of the model file itself and the model manifest file.

Executable Model Package

The other package option for distributing models is to share a self-contained executable software package in a common format, which combines the model itself along with any required pre-processing functions and possibly even the deep learning runtime framework. This allows the receiver of this package to simply deploy and install it in their target environment and start running inferences on it without the need to write any transform software or figure out how to configure the model in the target system. This option is better suited for clinical use and for model validation, as it provides a more controlled environment and ensures reproducible outcomes, which is key for these types of uses.

This type of package extends the contents of the data-only model package to include all required software to install and run inferences of supplied input data on the model on a target environment such as an operating system or container environment. The additional software may include:

- the ML framework package matching the model definition file, e.g., Tensorflow or PyTorch (although this is most likely available in the target environment and therefore not necessary to be part of the package)
- code to transform input data into a suitable format to be supplied as input to the model, typically written in Python, Java, JavaScript, Go for Kubernetes, and C++.
- additional software libraries needed to run the transform code, e.g., Python libraries.

All these files may be combined into an executable package, such as one of the following:

1. A common “notebook” format such as Jupyter Notebook which can be imported into cloud services ([BIB4.7.2-2] [BIB4.7.2-3] [BIB4.7.2-4], [BIB4.7.2-5]) to be deployed and run on the applicable cloud platform ([BIB4.7.2-1] [BIB4.7.2-6]).
2. A self-contained Docker file that can be deployed in a container environment like Kubernetes and executed on cloud services.
3. An executable, possibly packaged with an installer, for local deployments, that can be downloaded and run natively on a base O/S.

Optionally, the executable package may require a license to run.

4.8 Service

The Service entity represents a service and its associated metadata.

A Service is a running instantiation of an AI Application.

Table 4.8-1: Service Metadata

Attribute	Description
AI Application	The specific AI Application that is hosted by the Service
status	The state of the Service, such as running or offline
endpoint	The interface in which to interact with the Service

Profile writers may consider some of these questions for further discussion:

-

4.9 Model orchestration framework

The model orchestration framework entity represents a model orchestration framework and its associated metadata.

A model orchestration framework is an environment or platform with deep learning infrastructure and appropriate compute power, to run AI Models cloud-hosted or on-premises.

4.10 Feedback

The Feedback entity represents feedback and its associated metadata.

Feedback is an assessment by a human or system of the output produced by an AI model from inference performed on a given data record.

This entity will likely reference Entities: Data Element, Entities: Data Record, and Entities: AI Model.

Table 4.10-1: Feedback Metadata

Concept	Definition
model reference	Uniquely identifies the model that produced the subject of the feedback.
data record reference	Input data record from which the model produced the result/output for which feedback is provided
study reference series, instance, frame, coordinates of issue (of a label, of the pixels, of the markup, etc.)	This references the study where the feedback was generated from.
assessment	How well the result produced by the model represents the correct result for the task
feedback source	Identifies the human or process that provided this feedback
correct result	Optional, provide result data element(s) considered correct by the feedback source

Considerations that may have technical interoperability aspects include:

- The specific study reference may be optional, as the feedback may be going to an external 3rd party system.
    - This might also be a "multi-step" process; might be internally collected (where patient details are included) and then aggregated for anonymized analysis (where patient details are excluded)
- Feedback might also be implicit, e.g., when there is no feedback, they model may not be used at all, or the model is performing great. It is unlikely clinicians will provide feedback on well-working models all the time.
    - This is important to differentiate, as a model should not be reinforced if there is no implicit or explicit feedback.

Profile writers may consider some of these questions for further discussion:

- When feedback is being sent externally, how can it be de-identified but not lose purpose?
    - E.g., should patient population or general demographics be transmitted instead?
    - Does the institution or the specific scanner / version / firmware be included?
- What are the different types of feedback that would warrant different interventions?
    - E.g., a study failing because the imaging study is blurred should be treated differently than when the study has failed because it missed a finding.
    - Both types of data should be useful input to model training.
- When a model fails (with an error message), should this be treated the same as a feedback message?
- How is model feedback weighted?
- Is IHE SOLE [STD-SOLE] or IHE ATNA [STD-ATNA] a better construct for providing feedback?

5 Potential Profiles

<This section will be completed after Public Comment.>

With the areas of AI Interoperability in Imaging have been reviewed in previous chapters, this chapter highlights considerations for next steps.

For example., types of profiles, priority, extensions to existing standards, and considerations like security and data quality.

Appendices

Appendix A – AI Background

A.1 Deep Learning

A neural network is a computational learning system that uses a network of functions to understand and translate a data input of one form into a desired output and was inspired by neurobiology. [BIBA.1.10] Deep learning neural network models are the primary model structure used because of their top scoring performance on computer vision object detection, and segmentation benchmark tasks on data sets such as COCO (Common Objects in Context) [BIBA.1-1] [BIBA-2].

At a high level, a deep learning model uses the concept of the neuron to describe how it reacts in a specific layer of a particular region of a 2D or 3D image. These neurons will take an input, multiply the value with a model weight, and provide the output to the next layer of the model. These weights are critical to producing a deep learning result and are created as part of the model training process.

The recent performance gains of deep neural networks were made possible by increases in computing power and storage capacity to train these deep neural networks. Distributed learning increases computing power and storage for model training by spreading training across multiple server nodes or GPUs training the model in parallel [BIB3.3.2-3].

Neural network (NN) models are structured in multiple layers of nodes that use matrix multiplications to transform input data and pass output to adjacent layers. The layers are designed to learn increasingly higher-level imaging features [BIBA.1-3].

The nodes of the layers have a weight and an offset bias that transform input data to output. The weight determines the contribution of the input data to the model and the offset bias sets an activate threshold for the node to pass the transformed input data to the next layer. The nodes can be fully connected to the nodes of adjacent layers or only to spatially adjacent data by convolutions that match templates to the image data [BIBA.1-4].

Convolutional neural networks (CNN) are commonly used in medical imaging tasks because they only link data spatially close together. Data spatially close would have more relation than data far apart in medical images. CNNs have proven high performing in computer vision benchmarks.

Other, fully connected, architectures such as transformer networks [BIBA.1-5] can also be used in computer vision and therefore medical image tasks.

Transfer learning is reusing a previously trained model as a starting point for a second task [BIBA.1.11] by retraining the last layers [BIBA.1.12]. Offset bias is the learned constant additional inputs to nodes of neural network models that adjusts how easily a node will activate that are learned during model training [BIBA.1.13].

Four errors that could occur as part of model training include overfitting, bias, predictive errors, and variance errors. Overfitting a model is a type of error where the model function fits limited training data too closely and performance on novel data is lower as a result [BIBA.1.14]. Bias, a type of model error, is important to communicate, as clinical users of the model would want to know about to understand what kind of errors the model may make in clinical practice. Predictive error occurs due to systematic prejudice from faulty assumptions. Simpler models tend to have from bias [BIBA.1.15] [BIBA.1.16]. Variance errors occur when there are changes in predictive estimates of models with different training data. More complex models with more parameters tend to have higher variance.

A.2 Training Methods

Training (sometimes also called Learning) involves determining good values for all weights and biases of a model from labeled examples [BIBA.2-1].

Transfer Learning or "fine-tuning" involves small adjustments often with a smaller learning rate to model weights and biases to improve predictive performance of an existing model. [BIBA.2-2]

In Federated Learning, the model collaboratively learns a shared prediction model with distributed training data while keeping the data holders' data private. [BIBA.2-3]

Distributed Learning involves spreading computation of model training and training data storage across multiple machine nodes or GPUs to enable parallel computation to process larger models and larger training data sets.

A.3 Training Process

Training data composed of input and expected ground truth output samples are split into sets for training, and internal validation or cross-validation. Back propagation [BIBA.3-1] is used to iteratively adjust the node weights and biases of the model to fit the transformed outputs to expected training data outputs over epochs.

The error between the transformed output and expected ground truth is measured as loss. The model weights and offset biases are adjusted to minimize loss.

At the end of training epochs, the internal validation set is transformed by the partially trained model and compared to the expected output. Internal validation set loss growing larger than training set loss indicates model overfitting to the training portion of the data and the model does not generalize to data it was not trained on. Overfitting is reduced by randomly dropping out some nodes of layers each epoch, reducing the parameters of models by reducing nodes or layers, and cutting off training epochs when the internal validation set loss exceeds the training set loss.

Model weights are often saved after multiple epochs so the performance characteristics of the trained models after different epochs can be compared for selecting the best trained models for different tasks.

At the end of the training and internal validation runs model hyper-parameters such as number of layers, nodes in layers, initial weights, and offset biases, training epochs, learning rates may be adjusted. The training, internal validation sets are seen repeatedly by the model during training cycles. The hyper-parameters for the model are optimized for performance on the internal validation set. Models can be trained, starting from random weights and offset biases or from pretrained base models.

An alternative to model pre-training is self-training [BIBA.3-2]. In self-training models are trained with unlabeled data to perform data transforms and restorations such as local pixel shuffling and restoring [BIBA.3-3]. The unlabeled data is more common than labeled data increasing the self-training sets size. Then after self-training labeled data specific to the medical tasks are used to fine-tune unlocked layers of the model as in transfer learning.

A.4 Fine-Tuning Models

Models that have been fine-tuned should be able to function independently from the source model; it should have its own identity (with provenance tracked).

An existing pre-trained model, trained either at the same location or at a different location, can have the model weights and offset biases adjusted to address a new dataset, whether it was produced on different types of equipment, a different population, or other factors.

Pretrained models may come from online collections; see Appendix C.4: Reference Toolkits.

Existing models could have been pre-trained with non-medical images, or on more common medical images. The existing trained model layers can be transferred to address new datasets through a process called transfer learning. In transfer learning individual layers of trained models are locked or frozen from additional training or unlocked for fine-tuning of the weights and offset biases. More unlocked layers will require more fine-tuning epochs and more training samples in an iterative training process like training from scratch.

A.5 Training Models in Healthcare

Healthcare AI has a specific problem in that the data used is highly confidential and anonymizing that data may have damaging effects (especially when multiple pieces of data need to be linked together, or uses textual reports where anonymizing is an exceedingly difficult problem).

Medical image data’s different characteristics from the common computer vision datasets used to train model architectures can require training from scratch. Training from scratch starts with randomly initialized weights and offset biases of the nodes. Training from scratch requires sufficient ground truth, annotated positive and negative training, internal validation set and testing data.

To train a model and derive the correct weights, a large dataset is needed to guide the training process. There may not be enough ground truth annotated medical imaging data to train deep neural networks from scratch. Especially for positive cases of rare diseases. Starting with trained model layers and fine-tuning model weights and bias offsets with annotated medical image training sets can reduce the required training data.

Models can optionally fine-tune to individual institutions locally. The weights and offset biases of nodes in model layers can be unlocked and further trained on local image data to tune the model for the institution. This would change the model and earlier validation test results may no longer apply to this model. Local tuning would be more commonly used in research trials though the FDA is proposing regulatory frameworks for continuous tuning of models [BIBA.5-1].

Model architectures for medical imaging tasks can be selected by finding high performing architectures for similar computer vision tasks on benchmark image datasets. The selected base model architecture will need further adaptation for medical imaging tasks. Computer vision model architectures are often structured for two-dimensional color image data. Model architecture must be adapted for grayscale medical images and three dimensions for CT, MRI, and ultrasound. Trained base architecture layers can be selected as a starting point and further trained for medical imaging use cases.

AI Models and Applications often require significant hardware resources to analyze large volumes of images efficiently and produce results in time for clinicians to have them available when they need them. For example, AI Applications analyzing imaging exams taken in an ER will need to have shorter turnaround times than AI Applications looking for less urgent conditions like cancer or dementia. It is a significant challenge to provision and monitor resources to satisfy the many needs different departments in a hospital or similar institutions in an efficient and reliable manner.

The element of “time” plays a key role in developing AI Applications. Some data elements are related to other data elements in that they capture an observation about the same entity at a different point in time. For example, a volume measurement of the same tumor in the same patient, taken from two different studies at different time points.

Appendix B – Glossary / Definitions

The complete IHE Glossary for IHE Technical Framework terms is available here.

3160 For common terms used in AI and Machine Learning, several good references are:

- "Glossary of AI Terms." MITA, https://www.medicalimaging.org/about-standards/glossary-of-ai-terms
- Ranschaert, Erik R., Morozov, Sergey, Algra, Paul R. Artificial Intelligence in Medical Imaging. *https://www.springer.com/gp/book/9783319948775*

3165 Table B-1 consists only of terms that are used in this white paper in a more specific way than their "common" meaning.

Table B-1: Definitions for Terms in this White Paper

Term	Definition
data element	An individual piece of data that represents the inputs and outputs of an inference task
data record	A collection of related relevant data points related to a particular case. For example, this could be an imaging study and a classification label.
annotation	General term that describes a characterization of a particular portion of data. This could be as simple as a label (e.g., "pneumonia" tagged to an imaging study), or as complex as a multi-label segmentation (e.g., a set of pixel overlays that identify brain, brain tumor, and non-brain tissue). These should be represented as a tuple (code, codeset, description).
dataset	A group of data relating to a specific area. E.g., "pneumonia-positive patients" or "brain and tumor segmentations". The inclusion of data into this dataset is generally pre-determined, selected, and recorded, and not generated on-the-fly. Generally, the data in a dataset is used to train a model and knowing what data was used is important for model reproducibility and troubleshooting. It should also be specific, with coded values; for example, ": Should also be specific, have coded values defined etc., such as BRAIN CT AND MR, frontal, parietal, temporal, occipital lobes segmented as well as cerebellum and brainstem. All image slices must be 1mm or less. No contrast should have been administered. Subjects must be at least 5 years of Age."

Term	Definition
training dataset	A dataset (i.e., a set of data records) intended for use in training. Each record contains data elements that are the inputs to a model and, for supervised learning tasks, data elements that correspond to the model's expected output (link to ground truth / annotation).
testing dataset	A dataset intended for use as part of the training process to evaluate performance as the model weights converge.
validation dataset	A dataset intended for use as part of activities relating to user acceptance, regulatory approval, or payor approval.
AI Model	An AI Model is a package of model weights that has been trained to identify a particular target in a supplied input.
AI Application	An AI Application is a package of components to make an entire application, which could include model weights, algorithms, transforms, and input/output operators.
Validation	Determining whether the content of an entity meets its specifications. I.e., "I made the product correctly." [BIBD-1]
Verification	Determining whether the specification of an entity meets the user needs and intended use(s) I.e., "I made the correct product." [BIBD-1]

Appendix C – Further Reading and Bibliography

C.1 Bibliography

- [BIB3-1] "Waterfall Model.", Wikipedia, accessed Feb 27, 2021. https://en.wikipedia.org/wiki/Waterfall_model.
- [BIB4.2-1] "Introduction to Data in Machine Learning". GeeksforGeeks, May 17, 2020. https://www.geeksforgeeks.org/ml-introduction-data-machine-learning/.
- [BIB4.2-2] "What is annotation in machine learning?" Quora. https://www.quora.com/What-is-annotation-in-machine-learning
- [BIB4.2-3] "What is Data Annotation?" Appen, July 10, 2020 https://appen.com/blog/data-annotation/
- [BIB4.3.1] "Framing: Key ML Terminology." Machine Learning Crash Course, Google. https://developers.google.com/machine-learning/crash-course/framing/ml-terminology.
- [BIBD-1] "Design Controls", Joseph Tartal, FDA. https://www.fda.gov/media/116762/download
- [BIB3.4.1] "Comparing Machine Learning as a Service: Amazon, Microsoft Azure, Google Cloud AI, IBM Watson." Altexsoft, Sept 27, 2019. https://www.altexsoft.com/blog/datascience/comparing-machine-learning-as-a-service-amazon-microsoft-azure-google-cloud-ai-ibm-watson/
- [BIB4.7.2-1] "Learn more about all the notebooks experiences from Microsoft and GitHub." Microsoft. https://notebooks.azure.com/.
- [BIB4.7.2-2] "AI Platform Notebooks." Google. https://cloud.google.com/ai-platform-notebooks
- [BIB4.7.2-3] "Jupyter." Jupyter. https://jupyter.org/
- [BIB4.7.2-4] "Amazon Sagemaker." Amazon. https://aws.amazon.com/sagemaker/
- [BIB4.7.2-5] "IBM Watson Studio". IBM. https://www.ibm.com/ca-en/cloud/watson-studio
- [BIB4.7.2-6] "Use Notebooks with Azure Machine Learning". Use a Jupyter Notebook with Microsoft offerings, Microsoft. https://docs.microsoft.com/en-us/azure/notebooks/quickstart-export-jupyter-notebook-project#use-notebooks-with-azure-machine-learning.
- [BIB3.5.3-1] "Object Detection." Papers with Code, accessed February 21, 2021, https://paperswithcode.com/task/object-detection.
- [BIB3.5.3-2] "Semantic Segmentation." Papers with Code, accessed February 21, 2021, https://paperswithcode.com/task/semantic-segmentation.

- [BIB3.3.2-1] "Distributed training with TensorFlow". TensorFlow, accessed February 27, 2021, https://www.tensorflow.org/guide/distributed_training.
- [BIB3.3.2-2] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenberg, Tim Verbelen, Jan S. Rellermeyera. "A Survey on Distributed Machine Learning". CoRR, 2019. https://arxiv.org/pdf/1912.09789.pdf
- [BIB3.3.2-3] Hao Zhang. "Intro to Distributed Deep Learning Systems". February 6, 2018. https://medium.com/@Petuum/intro-to-distributed-deep-learning-systems-a2e45c6b8e7
- [BIB3.3.1-1] Nicholas Petrick. "Pre- and Post-Market Evaluation of Autonomous AI/ML: Lessons Learned from Prior CAD Devices". https://www.fda.gov/media/135712/download
- [BIB3.3.1-2] Ekin Tiu. "Metrics to Evaluate your Semantic Segmentation Model." August 9, 2019, https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2.
- [BIB3.3.2-4] Brendan McMahen, Daniel Ramage. "Federated Learning: Collaborative Machine Learning without Centralized Training Data". April 6, 2017. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.
- [BIB3.3.2-5] Wenqi Li, Fausto Milletari, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sebastien Ourselin, M. Jorge Cardoso, Andrew Feng. "Privacy-preserving Federated Brain Tumour Segmentation". October 2, 2019. https://arxiv.org/pdf/1910.00962.pdf.
- [BIB3.2.6-1] Jason Brownlee. "What is the Difference Between Test and Validation Datasets?". July 14, 2017. https://machinelearningmastery.com/difference-test-validation-datasets/
- [BIB3.8.2-1] "Minutes: WG-33 Data Archive and Management", DICOM, January 20, 2021, http://dicom.nema.org/dicom/minutes/wg-33/2021/WG33-2021-01-20-tcon-minutes.pdf.
- [BIB3.8.3-1] "Analysis of Optimal De-Identification Algorithms for Family Planning Data Elements." Integrating the Healthcare Enterprise, December 2, 2016, https://www.ihe.net/uploadedFiles/Documents/ITI/IHE_ITI_WP_Analysis-of-DeID-Algorithms-for-FP-Data_Elements.pdf.
- [BIB3.8.3-2] "Health Information Privacy." HHS.gov, accessed February 28, 2021, https://www.hhs.gov/hipaa/index.html.
- [BIB3.8.3-3] "Complete Guide to GDPR Compliance." GDPR.eu, accessed February 28, 2021, https://gdpr.eu/.
- [BIB3.8.5-1] "dcm2nii DICOM to NIfTI conversion." Dcm2nii, accessed February 28, 2021, https://people.cas.sc.edu/rorden/mricron/dcm2nii.html.

- [BIB3.8.9-1] Parikh, R., Mathai, A., Parikh, S., Chandra Sekhar, G., & Thomas, R. (2008). Understanding and using sensitivity, specificity, and predictive values. Indian journal of ophthalmology, 56(1), 45–50. https://doi.org/10.4103/0301-4738.37595. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2636062/
- [BIBA.1-1] "Computer Vision." Papers with Code, accessed February 28, 2021, https://paperswithcode.com/area/computer-vision.
- [BIBA.1-2], "What is COCO?" Common Objects in Context, accessed February 28, 2021, https://cocodataset.org/#home.
- [BIBA.1-3] Hosny, A., Parmar, C., Quackenbush, J. et al. Artificial intelligence in radiology. Nat Rev Cancer 18, 500–510 (2018). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6268174/
- [BIBA.1-4] Mazurowski, Maciej A et al. "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI." Journal of magnetic resonance imaging: JMRI vol. 49,4 (2019): 939-954. doi:10.1002/jmri.26534. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6483404/
- [BIBA.1-5] "End-to-end Object Detection with Transformers." Facebook AI, May 27, 2020, https://ai.facebook.com/research/publications/end-to-end-object-detection-with-transformers.
- [BIBA.1-10] "What is a Neural Network?" DeepAI, accessed February 28, 2021, https://deepai.org/machine-learning-glossary-and-terms/neural-network.
- [BIBA.1-11] "A Gentle Introduction to Transfer Learning for Deep Learning" Machine Learning Mastery, September 16, 2019, https://machinelearningmastery.com/transfer-learning-for-deep-learning/.
- [BIBA.1-12] Mohamed Elgendy. "Deep Learning for Vision Systems", Chapter 6. https://www.manning.com/books/deep-learning-for-vision-systems
- [BIBA.1-13] "Weights and Biases." AI Wiki, Accessed February 28, 2021, https://docs.paperspace.com/machine-learning/wiki/weights-and-biases.
- [BIBA.1-14] Will Kenton. "Overfitting." Investopedia, November 28, 2020, https://www.investopedia.com/terms/o/overfitting.asp.
- [BIBA.1-15] Jaspreet. "Understanding and Reducing Bias in Machine Learning." April 5, 2019, https://towardsdatascience.com/understanding-and-reducing-bias-in-machine-learning-6565e23900ac.
- [BIBA.1-16] Jason Brownlee. "Gentle Introduction to the Bias-Variance Trade-Off in Machine Learning." Machine Learning Mastery, October 25, 2019, https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/.

- [BIBA.2-1] “Descending into ML: Training and Loss.” Machine Learning Crash Course, Google, Accessed February 28, 2021, https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss.
- [BIBA.2-2] “Fine-Tuning.” Dive into Deep Learning, accessed February 28, 2021, https://d2l.ai/chapter_computer-vision/fine-tuning.html.
- [BIBA.2-3] Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning of predictive models from federated Electronic Health Records. Int J Med Inform. 2018 Apr; 112:59-67. doi: 10.1016/j.ijmedinf.2018.01.007. Epub 2018 Jan 12. PMID: 29500022; PMCID: PMC5836813. https://pubmed.ncbi.nlm.nih.gov/29500022/
- [BIBA.3-1] Geoff Currie, K. Elizabeth Hawk, Eric Rohren, Alanna Vial, Ran Klein. “Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging.” Journal of Medical Imaging and Radiation Sciences. October 7, 2019. Vol 50 Issue 4 P477-487. https://www.jmirs.org/article/S1939-8654(19)30504-1/fulltext
- [BIBA.3-2] Barret Zoph and Golnaz Ghiasi and Tsung-Yi Lin and Yin Cui and Hanxiao Liu and Ekin D. Cubuk and Quoc V. Le. “Rethinking Pre-training and Self-training.” 2020. https://arxiv.org/abs/2006.06882
- [BIBA.3-3] Zongwei Zhou and Vatsal Sodha and Md Mahfuzur Rahman Siddiquee and Ruibin Feng and Nima Tajbakhsh and Michael B. Gotway and Jianming Liang. “Models Genesis: Generic Autodidactic Models for 3D Medical Image Analysis”. 2019. https://arxiv.org/abs/1908.06912
- [BIBA.5-1] “Proposed Regulatory Framework for Modifications to Artificial Intelligence/Machine Learning (AI/ML)-Based Software as a Medical Device (SaMD).” FDA, https://www.fda.gov/files/medical%20devices/published/US-FDA-Artificial-Intelligence-and-Machine-Learning-Discussion-Paper.pdf

C.2 Recommended Reading

The following papers were recommended by contributors of this white paper.

C.2.1 Medical Imaging Papers

- Mongan, Moy, Kahn. Checklist for Artificial Intelligence in Medical Imaging (CLAIM): A Guide for Authors and Reviewers, Radiology: Artificial Intelligence, Vol. 2, No. 2, https://pubs.rsna.org/doi/10.1148/ryai.2020200029
- Ranschaert, Erik R., Morozov, Sergey, Algra, Paul R. Artificial Intelligence in Medical Imaging. https://www.springer.com/gp/book/9783319948775
- J. Raymond Geis, Adrian P. Brady, Carol C. Wu, Jack Spencer, Erik Ranschaert, Jacob L. Jaremko, Steve G. Langer, Andrea Borondy Kitts, Judy Birch, William F. Shields, Robert van den Hoven van Genderen, Elmar Kotter, Judy Wawira Gichoya, Tessa S. Cook, Matthew B. Morgan, An Tang, Nabile M. Safdar, Marc Kohli. Ethics of Artificial

Intelligence in Radiology: Summary of the Joint European and North American Multi-society Statement. https://pubs.rsna.org/doi/full/10.1148/radiol.2019191586

- Martin J. Willemink, MD, PhD, Wojciech A. Koszek, MS, Cailin Hardell, MS, Jie Wu, MS, Dominik Fleischmann, MD, Hugh Harvey, MD, Les R. Folio, DO, MPH, Ronald M. Summers, MD, PhD, Daniel L. Rubin, MD MS, and Matthew P. Lungren, MD, MPH. "Preparing Medical Imaging Data for Machine Learning.", *Radiology*, April 2020, pp 4-15, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7104701/.
- Levine AB, Schlosser C, Grewal J, Coope R, Jones SJM, Yip S. Rise of the Machines: Advances in Deep Learning for Cancer Diagnosis. Trends Cancer. 2019 Mar;5(3):157-169. doi: 10.1016/j.trecan.2019.02.002. Epub 2019 Feb 28. PMID: 30898263. https://pubmed.ncbi.nlm.nih.gov/30898263/
- Kohli, M.D., Summers, R.M. & Geis, J. Medical Image Data and Datasets in the Era of Machine Learning—Whitepaper from the 2016 C-MIMI Meeting Dataset Session. J Digit Imaging 30, 392–399 (2017). https://doi.org/10.1007/s10278-017-9976-3
- Marc Kohli, Luciano M. Prevedello, Ross W. Filice and J. Raymond Geis Implementing Machine Learning in Radiology Practice and Research: American Journal of Roentgenology: Vol. 208, No. 4 (AJR) https://www.ajronline.org/doi/full/10.2214/AJR.16.17224
- Curtis P. Langlotz, MD, PhD Bibb Allen, MD Bradley J. Erickson, MD, PhD Jayashree Kalpathy-Cramer, PhD Keith Bigelow, BA Tessa S. Cook, MD, PhD Adam E. Flanders, MD Matthew P. Lungren, MD, MPH, David S. Mendelson, MD Jeffrey D. Rudie, MD, PhD Ge Wang, PhD Krishna Kandarpa, MD, PhD A Roadmap for Foundational Research on Artificial Intelligence in Medical Imaging: From the 2018 NIH/RSNA/ACR/The Academy Workshop Radiology 2019; 291:781–791 https://doi.org/10.1148/radiol.2019190613
- Harvey H., Glocker B. (2019) A Standardized Approach for Preparing Imaging Data for Machine Learning Tasks in Radiology. In: Ranschaert E., Morozov S., Algra P. (eds) Artificial Intelligence in Medical Imaging. Springer, Cham. https://doi.org/10.1007/978-3-319-94878-2_6
- Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J-W, da Silva Santos LB, Bourne PE, Bouwman J, Brookes AJ, Clark T, Crosas M, Dillo I, Dumon O, Edmunds S, Evelo CT, Finkers R, Gonzalez-Beltran A, Gray AJG, Groth P, Goble C, Grethe JS, Heringa J, 't Hoen PAC, Hooft R, Kuhn T, Kok R, Kok J, Lusher SJ, Martone ME, Mons A, Packer AL, Persson B, Rocca-Serra P, Roos M, van Schaik R, Sansone S-A, Schultes E, Sengstag T, Slater T, Strawn G, Swertz MA, Thompson M, van der Lei J, van Mulligen E, Velterop J, Waagmeester A, Wittenburg P, Wolstencroft K, Zhao J, Mons B. The FAIR Guiding Principles for scientific data management and stewardship. Sci Data. 2016; 3:160018. ISSN: 2052-4463. https://doi.org/10.1038/sdata.2016.18.

- Li, Tian and Sahu, Anit Kumar and Talwalkar, Ameet and Smith, Virginia. “Federated Learning: Challenges, Methods, and Future Directions.” IEEE Signal Processing Magazine, Vol 37 No 3 May 2020. https://arxiv.org/abs/1908.07873

C.2.2 Medical Imaging AI Websites

- “Federated Learning: Collaborative Machine Learning without Centralized Training Data.” Google, April 2017, https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
- “Artificial Intelligence (AI) and Machine Learning (ML) in Medical Devices.” FDA, October 20, 2020, https://www.fda.gov/media/142998/download.
- “Request for Information (RFI) in the Federal Register.” Federal Register, https://www.federalregister.gov/documents/2019/05/01/2019-08818/artificial-intelligence-standards
- “New guidance for AI in screening.” Public Health England, March 14, 2019. https://phescreening.blog.gov.uk/2019/03/14/new-guidance-for-ai-in-screening/
- “ACR’s Platform-Model Communication for AI.” American College of Radiology, Accessed February 27, 2021, https://www.acrdsi.org/-/media/DSI/Files/ACR-DSI-Model-API.pdf.

C.2.3 Deep Learning Recommended Reading

- Ian Goodfellow, Yoshua Bengio, Aaron Courville. "Deep Learning". The MIT Press, 2016. https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf
- Pullum, Laura, Steed, Chad, Jha, Sumit Kumar, and Ramanathan, Arvind. Mathematically Rigorous Verification & Validation of Scientific Machine Learning. United States: N. p., 2018. https://www.osti.gov/servlets/purl/1474476
- Mohamed Elgendy. “Deep Learning for Vision Systems”. Chapter 6 Transfer Learning. October 2020. https://www.manning.com/books/deep-learning-for-vision-systems
- R.J. Brooks, A.M. Tobias. “Choosing the best model: Level of detail, complexity, and model performance.” Mathematical and Computer Modelling, Volume 24, Issue 4, 1996, Pages 1-14, ISSN 0895-7177. https://www.sciencedirect.com/science/article/pii/0895717796001033

C.2.4 Deep Learning Websites

- “Papers with Code.” Papers with Code, accessed February 21, 2021, https://paperswithcode.com/.

- “Fine-Tuning.” Dive into Deep Learning. https://d2l.ai/chapter_computer-vision/fine-tuning.html
- “Object Detection.” Papers with Code, https://paperswithcode.com/task/object-detection
- “What is a machine learning model?” Microsoft. https://docs.microsoft.com/en-us/windows/ai/windows-ml/what-is-a-machine-learning-model

C.3. Inventory of Standards

IHE Radiology Profiles

- [STD-AIR] “AI Results (AIR).” Integrating the Healthcare Enterprise, July 16, 2020, https://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_RAD_Suppl_AIR.pdf
- [STD-AIW-I] “AI Workflow for Imaging (AIW-I).” Integrating the Healthcare Enterprise, August 6, 2020, https://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_RAD_Suppl_AIW-I.pdf.
- [STD-FUNC] “Follow-Up of Non-Critical Actionable Findings (FUNC).” Integrating the Healthcare Enterprise, August 11, 2017, https://docs.google.com/document/d/1pEQAIWDuD0HPQisBLlzF_FaovG8aWuIKgl163I3kr8E/edit.
- [STD-CDS-OAT] “Clinical Decision Support Order Appropriateness Tracking (CDS-OAT).” Integrating the Healthcare Enterprise, April 25, 2019, https://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_Rad_Suppl_CDS-OAT.pdf.
- [STD-SOLE] “Standardized Operational Log of Events (SOLE).” Integrating the Healthcare Enterprise, July 27, 2018, https://www.ihe.net/uploadedFiles/Documents/Radiology/IHE_RAD_Suppl_SOLE.pdf.
- [STD-TCE] “Teaching File and Clinical Trial Export (TCE).” Integrating the Healthcare Enterprise, September 14, 2020, https://wiki.ihe.net/index.php/Teaching_File_and_Clinical_Trial_Export.

IHE IT Infrastructure Profiles

- [STD-XDS] “Cross-Enterprise Document Sharing (XDS).” Integrating the Healthcare Enterprise, June 20, 2019, https://wiki.ihe.net/index.php/Cross-Enterprise_Document_Sharing.
- [STD-ATNA] “Audit Trail and Node Authentication (ATNA).” Integrating the Healthcare Enterprise, March 19, 2020, https://wiki.ihe.net/index.php/Audit_Trail_and_Node_Authentication.

- [STD-PIX] "Patient Identifier Cross-Referencing (PIX)." Integrating the Healthcare Enterprise, July 2, 2018, https://wiki.ihe.net/index.php/Patient_Identifier_Cross-Referencing.

IHE Quality, Research and Public Health Profiles

- [STD-SDC] "Structured Data Capture (SDC)." Integrating the Healthcare Enterprise, March 19, 2019, https://ihe.net/uploadedFiles/Documents/QRPH/IHE_QRPH_Suppl_SDC.pdf.

IHE Patient Care Coordination Profiles

- [STD-DCTM] "Dynamic Care Team Management (DCTM)." Integrating the Healthcare Enterprise, December 11, 2020, https://wiki.ihe.net/index.php/Dynamic_Care_Team_Management_(DCTM).

DICOM Standards

- [STD-DICOM] "Digital Imaging and Communications in Medicine." DICOM, https://www.dicomstandard.org/.
- [STD-DICOMWEB] "DICOMweb." DICOM, https://www.dicomstandard.org/dicomweb
- [STD-DICOM-DEIDENT] "Attribute Confidentiality Profiles." DICOM PS3.15 E, http://dicom.nema.org/medical/dicom/current/output/chtml/part15/chapter_E.html
- [STD-DICOM-MPPS] "Modality Performed Procedure Step Information Object Definition." DICOM PS3.3 B.17, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_B.17.html
- [STD-DICOMWEB-UPSRS] "Worklist Services and Resources." DICOM PS3.18 11, http://dicom.nema.org/medical/dicom/current/output/chtml/part18/chapter_11.html
- [STD-DICOM-MWL] "Modality Worklist SOP Class." DICOM PS3.4 K.6.1, http://dicom.nema.org/medical/dicom/current/output/chtml/part04/sect_K.6.html#sect_K.6.1
- [STD-DICOM-SR] "Structured Report Document Information Object Definitions." DICOM PS3.3 A.35, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_A.35.html
- [STD-DICOM-SEG] "Segmentation IOD." DICOM PS3.3 A.51, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_A.51.html
- [STD-DICOM-PM] "Parametric Map IOD." DICOM PS3.3 A.75, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_A.75.html
- [STD-DICOM-KOS] "Key Object Selection Modules." DICOM PS3.3 C.17.6, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.17.6.html

- [STD-DICOM-RTSS] “RT Structure Set IOD.” DICOM PS3.3 A.19, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_A.19.html
- [STD-DICOM-SSEG] “Surface Segmentation IOD.” DICOM PS3.3 A.57, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_A.57.html
- [STD-DICOM-SECCAP] “Secondary Capture Modules.” DICOM PS3.3 C.8.6, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.8.6.html
- [STD-DICOM-GSPS] “Greyscale Softcopy Presentation State IOD”, DICOM PS3.3 A.33.1, http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_A.33.html#sect_A.33.1
- [STD-DICOM-SUP219] “Sup 219 – JSON Representation of DICOM Structured Reports”, DICOM, https://www.dicomstandard.org/News-dir/ftsup/docs/sups/Sup219.pdf

HL7 Standards

- [STD-HL7] “HL7 Version 2 Product Suite.” HL7, https://www.hl7.org/implement/standards/product_brief.cfm?product_id=185
- [STD-FHIR] “Fast Healthcare Interoperability Resources.” HL7, https://www.hl7.org/fhir/overview.html
- [STD-FHIR-OBS] “Observations.” FHIR, HL7, https://www.hl7.org/fhir/observations.html.
- [STD-FHIR-DXR] “DiagnosticReport.” FHIR, HL7, https://www.hl7.org/fhir/diagnosticreport.html.
- [STD-FHIR-PT] “Patient.” FHIR, HL7, https://www.hl7.org/fhir/patient.html.
- [STD-FHIR-APPT] “Appointment.” FHIR, HL7, https://www.hl7.org/fhir/appointment.html.
- [STD-FHIR-SCHED] “Schedule.” FHIR, HL7, https://www.hl7.org/fhir/schedule.html.
- [STD-FHIR-SVCREQ] “ServiceRequest.” FHIR, HL7, https://www.hl7.org/fhir/servicerequest.html.
- [STD-FHIR-PROC] “Procedure.” FHIR, HL7, https://www.hl7.org/fhir/procedure.html.
- [STD-FHIR-COND] “Condition.” FHIR, HL7, https://www.hl7.org/fhir/condition.html

Other References

- [STD-CPT] “CPT Codes Lookup”, Codify by AAPC, Accessed February 28, 2021, https://www.aapc.com/codes/cpt-codes-range/.

- [STD-ICD] “International Statistical Classification of Diseases and Related Health Problems”, World Health Organization, Accessed February 28, 2021, https://www.who.int/standards/classifications/classification-of-diseases.
- [STD-SNOMED] “SNOMED International”, SNOMED, Accessed February 28, 2021, https://www.snomed.org/.
- [STD-LOINC] “LOINC”, LOINC, Accessed February 28, 2021, https://loinc.org/.
- [STD-ACR-LUNG] “Lung CT Screening Reporting & Data System”, American College of Radiology, Accessed March 4, 2021, https://www.acr.org/Clinical-Resources/Reporting-and-Data-Systems/Lung-Rads.
- [STD-ACR-BIRAD] “ACR BI-RADS Atlas 5th Edition”, American College of Radiology, Accessed March 4, 2021, https://www.acr.org/Clinical-Resources/Reporting-and-Data-Systems/Bi-Rads.
- [STD-FMA] “Foundational Model of Anatomy”, University of Washington School of Medicine, Accessed February 28, 2021, http://sig.biostr.washington.edu/projects/fm/AboutFM.html.

C.4. Reference Toolkits

The following toolkits were recommended by contributors of this White Paper. There was no intent to make this list comprehensive. No commercial tools were included. These tools may inform use cases and future profile development. Feel free to suggest others.

Deep Learning Toolkits

- MXNet: https://mxnet.apache.org/
- PyTorch: https://pytorch.org/
  - MONAI: https://monai.io/
- TensorFlow: https://www.tensorflow.org/
  - DLTK: https://dltk.github.io/

Sharing Model Formats

- HDF5: (https://www.hdfgroup.org/solutions/hdf5/)
- ONNX: (https://onnx.ai/)
- Python pickle used by PyTorch: (https://pytorch.org/tutorials/beginner/saving_loading_models.html, https://docs.python.org/3/library/pickle.html).
- Tensorflow “SavedModel”: (https://www.tensorflow.org/guide/saved_model)

Binary Model Formats

- Tensorflow lite and “xxd”: https://www.tensorflow.org/lite/microcontrollers/build_convert
- TorchScript: https://pytorch.org/tutorials/advanced/cpp_export.html

Model Sources

- GitHub: https://github.com/.
- Kaggle: https://www.kaggle.com/.
- Model Zoo: https://modelzoo.co/.
- Papers with Code: https://paperswithcode.com/.

De-Identification

- PixelMed DICOM Cleaner: http://www.dclunie.com/pixelmed/software/webstart/DicomCleanerUsage.html