ACR's Platform-Model Communication for AI

The American College of Radiology's Data Science Institute (ACR DSI) has drafted a model API in collaboration with industry, organizational and other partners that it believes will lower the barrier for entry for developers looking to enter the healthcare space, as well as help to standardize communications to the models created, allowing them to be platform agnostic. The ACR DSI seeks public comment on this specification as well as interest in evaluating a proof-of-concept implementation on the ACR DSI's AI-LAB platform.

Background

As of writing, there does not exist a standard for AI model inference in a clinical workflow. As a result, models must be packaged for a given orchestration platform, limiting their portability. Should deployment on a different orchestration platform subsequently be desired, the model must be repackaged for the new API, tested for correctness, and potentially recertified for regulatory compliance. Alternatively, should this not be feasible, an additional orchestration platform could be deployed in the clinical environment to support the new model at additional cost, namely payment to the platform vendor, compute nodes for hosting the platform, and personnel to manage it. Neither arrangement is desirable for the parties involved.

- Purchasers wish to select and deploy models regardless of their orchestration platform.
- IT staff wish to minimize the number of systems which must be supported and managed.
- Model developers wish to sell their models to customers without the overhead of repackaging them on a case-by-case basis.
- Orchestration platform vendors wish to offer their customers the best user experience.

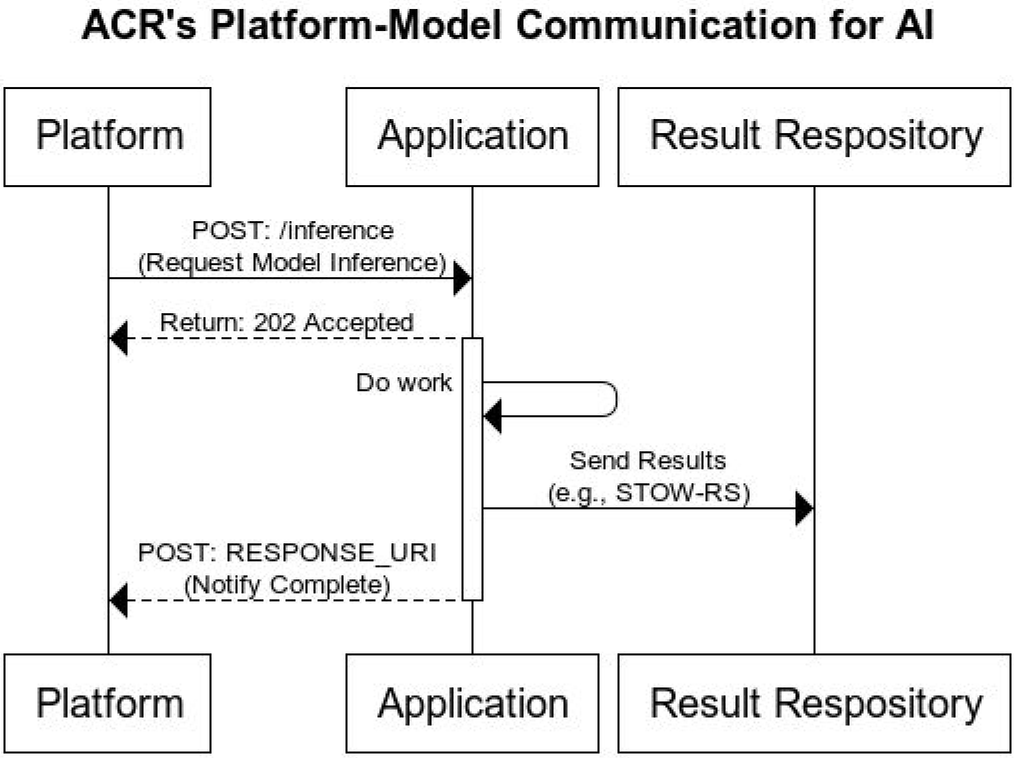
The proposed API aims to solve these challenges by simultaneously being simple enough for those without deep expertise in medical standards to implement as well as flexible enough to support the most complex workflows. To provide the model with maximum flexibility, lessen the requirements of the orchestration platform, and minimize the coupling between the two, the API carries a minimalist philosophy, conveying only the information that is required for execution -- should an algorithm require additional data, it is free to fetch it from the provided clinical datastores.

The proposed version of the API will be explained in this document along with items that were discussed as potential edge cases and/or gaps and how they may be addressed under the current structure. Please refer to the Model API attached in Appendix A which contains comments and details. Appendix B is an example of the Model API without the comments.

API Details

Workflow

The proposed API details an asynchronous workflow designed to support potentially long-running AI model (also known as algorithm) inference requests. Upon notification of the availability of clinical data, the platform (otherwise known as orchestration system) may request zero or more AI models to execute on the new data. The request issued to the model by the platform consists largely of metadata associated with the newly acquired clinical data, allowing the model to fetch data as needed. If it is able to process the request, the model shall return a 202 Accepted and begin processing the job asynchronously. Once the inference is complete and all results have been successfully sent to the appropriate output datastores, the model shall issue a response to the responseURI included in the request body, enumerating the resources it created and signalling its completion.



Inference Request-Reponse

Inference Request Header

The API supports, but does not mandate, the use of the authorization header in the request. Given the wide variety of authentication mechanisms available, if present, the authentication scheme is dictated by the orchestration system and must be offloaded, via a sidecar or other means, prior to when the request is passed to the model. The model is ***not*** mandated to perform any authentication of requests.

Inference Request Body

A transaction ID must be included in all requests for correlation. This is used by the platform to track subsequent requests from the model and shall be included in all requests to clinical datastores (where applicable). Additionally, it shall be included in the response.

A response URI shall be provided to the model. This URI is where the response is to be posted upon completion of inference or to signal failure to successfully complete the request.

Priority may be included in the request body. If priority is not specified in the request, but is used by the model, the model will use a default priority of 128. Priority is available for models that handle multiple requests, or can scale resources. Larger values indicate higher priority with the valid range being 0-255.

Input metadata provides the details surrounding the inference request.

- There is an optional workflow stage. This tells the model what triggered the execution request. The supported options at this time are study acquisition, storage or interpretation. A FHIR encounter or CCG, a user trigger, which can be used for interactive model requests and other, which can be combined with additional information and can be used in special cases.
- The details field allows for one or more DICOM UIDs, FHIR Resources, accession numbers, or a DICOM patient ID[1].
    - DICOM UID(s) may refer to one or more study, series, instances, or frames. Multiple studies, series, and instances may be passed to the model through this nested request structure. Of note, if frames are specified, only one SOP instance UID shall be specified in the SOPInstanceUID field. In this case, if multiple SOPInstanceUIDs are needed to fully specify the request, the same series can be repeated in the array structure[2].
    - FHIR resource(s) may be specified by indicating both the resource type and resource id. Valid resource types are limited to Observation, ImagingStudy, DiagnosticReport and Bundle.

[1] Only a single data type can be provided in the inference request; if for example DICOM UIDs and FHIR Resources were both required, a more flexible mechanism is provided (detailed below in Special Use Case Note.

[2] While this breaks the symmetry of the request structure, it is believed that the API will be used less frequently to specify frames than instances and that a potentially non-trivial number of instances may be specified. Therefore, this decision was made to minimize the verbosity of the API.

- Accession number(s) may be specified to refer to imaging studies, lab tests, or other similarly identified medical data.
- Patient ID is a DICOM-specific field and may correlate to an MRN, but in some cases may not.

Input resources shall be provided in the inference request and list where the model may find the additional data it needs to perform inference. The supported interface types include XDS, DICOMWeb, DIMSE, FHIR, S3 and file folders. The order in which these are listed in the request, should there be more than one endpoint that can provide data, suggests the order in which the model should query. The connection details must be provided and enumerate a list of operations the endpoint offers to the model as well as additional necessary connection information. These resources may or may not differ from output resources.

Output resources shall be provided and list where the model shall store artifacts and resources it creates as a result of its inference. These may or may not be different from the input resources, therefore artifacts should only be sent to locations specified as output resources. The same list of interface types are available as outputs, and the valid operations for the endpoint must be specified along with connection information.

Special Use Case Note:
When additional details and parameters are required for model inference, these details can be placed in a DICOM SR. This DICOM SR can be hosted by the platform at its own endpoint or stored in some other endpoint that is specified as an input resource to the model. The model shall not be concerned as to which this is. In this case, a DICOM UID shall be used to specify the SR containing such information.

Inference Response

Upon receiving a request, the model shall respond to the API with a valid HTTP status code indicating whether the request has been accepted for processing.

Completion Request-Response

Completion Request

Upon the completion or failure of model inference, the model shall post a message to the URI specified in the inference request. This request allows the platform to close the transaction requested of the model.

The request shall include the original transaction ID as well as a status code indicating whether the model successfully completed its computation. Status codes shall follow the DICOM standard listed in Section 8.5 of Part 18. An optional human readable message may be included in this request.

Additionally, if the model posts any artifacts to the output data sources, the request shall include a list of created resources. These resources will follow the structure from the input metadata section for DICOM. FHIR resources will be listed as a tuple of pairs defined as a resource type and id. Enterprise Imaging are resources stored to XDS endpoints defined by submission set id. Clinical data other specifies items that may be stored as files in a folder or as a GUID. The location of these resources need not be specified as the Transaction ID links the created resource to the available output resources that were defined in the request.

Completion Response

The platform shall respond with a valid HTTP status code to indicate whether it successfully processed the completion request.

Health Probes

Models must provide liveness and readiness endpoints for platforms to be able to use.

Appendix A

```
# HEADER
# Auth, if present, handled by auth provider external to algorithm
# Will reject unauthorized requests
{
    "Authorization": "ALGO_TOKEN"
}

# Request
{
    # [required]
    # for correlation -- to be included in all requests
```

```
"transactionID": "MY_TRANSACTION_TOKEN",

# [required]
# URI to which algorithm responds to signal completion
"responseURI": "http://foo.bar",

# [optional]
# Bearer auth token used in response to responseURI if present
# Should be placed in "Authorization" header
"responseAuthToken": "MY_RESPONSE_TOKEN",

# [optional]
# the priority at which the request should be treated
# higher is greater; valid range is 0-255
# if not specified, defaults to priority 128.
"priority": 128,

# [required]
# details the data associated with the inference request
"inputMetadata": {
    # [optional]
    # what triggered execution. options include:
    #   STUDY_ACQUISITION
    #   STUDY_STORAGE
    #   STUDY_INTERPRETATION
    #   FHIR_Encounter
    #   FHIR_CCG
    #   USER_TRIGGERED (interactive request)
    #   OTHER
    "workflowStage": "STUDY_ACQUISITION",

    # [required]
    "details":
      # one of the following types
      # every entry in this list must contain, at a minimum, a field "type"
      # which details how the object should be parsed
      # Valid values of type include:
      #   DICOM_INSTANCE_UID - used for any DICOM object(s) or complex workflows which
      #   can leverage atypical request parameters in an SR
      #   DICOM_PATIENT_ID
      #   FHIR_RESOURCE
      #   ACCESSION_NUMBER
      {
```

```
    # [required]
    "type": "DICOM_INSTANCE_UID",

    # DICOM_UID implementation
    # [required]
    "studies": [
        {
            "StudyInstanceUID": "2.16.840.1.114488.0.4.123489834087.13300714",

            # [optional]
                "series": [
                    {
                        # [required]
                        "SeriesInstanceUID": "",

                        # [optional]
                        "instances": {

                            # [required]
                            "SOPInstanceUID": ["", ""],

                            # [optional]
                            "frames": [
                                # [required]
                                "FrameNumber": ["",""]
                            ]
                        }
                    },
                    # ...
                ]
           },
           # ...
      ]
  },

# {
#     # [required]
#     "type": "DICOM_PATIENT_ID",
      #     # DICOM_PATIENT_ID implementation
      #     "PatientID": "XXX"
      # },
      # {
      #     # [required]
      #     "type": "FHIR_RESOURCE"

            # FHIR_RESOURCE implementation
      #     "resources": [
      #         {
      #             # [required]
      #             # May be one of
      #             #   Observation
      #             #   ImagingStudy
      #             #   DiagnosticReport
      #             "resourceType": "Observation",
      #             # [required]
      #             "id": "24653",
      #         }
      #      ]
      # },
      # {
      #     # [required]
      #     "type": "ACCESSION_NUMBER",

      #     # ACCESSION_NUMBER implementation
      #     "accessionNumber": ["XXX",]
      # }
},

# [required]
# data sources which can be used to fetch information
# Algorithms to operate via a pull model for maximum flexibility
# May leverage resources provided by supplying transactionID
# Resources listed are those requested/approved upon deployment
# Resource types include, but are subject to extension:
#   C-CDA
#   FHIR_RESOURCE
#   ENTERPRISE_IMAGING
#   DICOM_IMAGING
#   CLINICAL_DATA_OTHER
#
# Permitted operations include, but are subject to extension:
```

```
#   QUERY
#   RETRIEVE
#   STORE
#   MOVE
#   GET
#   PUT
#   POST
#   READ
#   WRITE
#   Retrieve Document Set
#   Stored Query
#   WADO Retrieve
#   Retrieve Images
#   Retrieve Presentation States
#   Retrieve Reports
#   Retrieve Imaging Document Set
#   Retrieve Key Image Note
#   Retrieve Evidence Documents
#   Provide and Register Imaging Document Set
#   C-MOVE
#   C-STORE
#   STORAGE COMMIT
#   STOW-RS
#
# Permitted interfaces include, but are subject to extension:
#   XDSb
#     "connectionDetails" :{
#          "operations": ["list of permitted operations"],
#          "uri": "http://my.xds.endpoint",
#          "authID": "TOKEN"
#     }
#   XDS-I.b
#     "connectionDetails" :{
#          "operations": ["list of permitted operations"],
#          "uri": "http://my.xds.endpoint",
#          "authID": "TOKEN"
#     }
#   DIMSE
#     "connectionDetails" :{
#          "operations": ["list of permitted operations"],
#          "aet": "MYPACSAET",
#          "hostName": "mypacs.domain.org",
#          "port": "104"
#     }
#   DICOMweb
#     "connectionDetails" :{
#          "operations": ["list of permitted operations"],
#          "uri": "http://my.xds.endpoint",
#          "authID": "TOKEN"
#     }
#   S3
#     "connectionDetails" :{
#          "operations": ["list of permitted operations"],
#          "hostName": "https://bucketname.s3.Region.amazonaws.com",
#          "bucket": "bucketname",
#          "authorization": "KEY"
#     }
#   FileFolder
#     "connectionDetails" :{
#          "operations": ["list of permitted operations"],
#          "path": "\\mainfolder\subfolder",
#          "user": "UserName"
#     }
#   FHIR
#     "connectionDetails" :{
#          "operations": ["list of permitted operations"],
#          "baseUri": "http://hackathon.siim.org/fhir-overview/fhir/",
#          "apiKey": "MyApiKey"
#     }
#
# The order of the list suggests to the algorithm the order of
# query when multiple interfaces of the same type of present.
"inputResources": [
    {
        "interface": "DICOMweb",
        "connectionDetails": {
            "operations": ["QUERY", "RETRIEVE"],
            "uri": "http://localhost:8000/dicom-web",
            "authID": "TOKEN" # [optional] bearer auth token
        }
    },
],

# [required]
# Resources to which model results are output
# May differ from those specified in inputResources
```

```
    "outputResources": [
        {
            "interface": "DICOMweb",
            "connectionDetails": {
                "operations": ["STORE"],
                "uri": "http://localhost:8001/dicom-web",
                "authID": "TOKEN" # [optional] bearer auth token
            },
        },
    ]
}

# valid response codes include:
#   * 200 - ack’d and processing request, still may not run algorithm
#   * 204 - the algorithm will not run (?)
#   * 400 - malformed request
#   * 401 - unauthorized
#   * 403 - forbidden
#   * 500 - internal error
#   * 501 - trigger not supported
#   * 503 - service unavailable
#   * 505 - http version not supported
# RESPONSE:
#   {
#     “message”: “message providing further information re:status code”
#   }


# Upon Inference Completion
# POST: responseURI
#
# Request:
{
    # [required]
    # for correlation -- to be included in all requests
    "transactionID": "MY_TRANSACTION_TOKEN",

    # [required]
    # see DICOM status codes
    # http://dicom.nema.org/medical/dicom/current/output/chtml/part18/sect_8.5.html
    "status": 500,

    # [optional]
# A human-readable message providing additional information
"message": “my string”,

# [optional]
# A list of created resources
# follows list of inputMetadata:
"resources": [
      {
          "type": "DICOM_INSTANCE_UID",
          "studies": [
              {
                  "StudyInstanceUID": "2.16.840.1.114488.0.4.123489834087.13300714",

                  # [optional]
                      "series": [
                           {
                                # [required]
                                "SeriesInstanceUID": "",

                                # [optional]
                                "instances": [

                                     "SOPInstanceUID": ["", ""],

                                     # [optional]
                                     "frames": [

                                            "FrameNumber": ["",""]
                                     ]
                                ]
                           },
                           # ...
                      ]
                  },
                  # ...
             ]
        },
          #...
          # "FHIR_Resource (optional)":["Resource,ID","Resource,ID","Resource,ID"],
          # "ENTERPRISE_IMAGING (optional)":["SubmissionSetUID","SubmissionSetUID"],
          # "CLINICAL_DATA_OTHER (optional)":["FileName or GUID","FileName or GUID"]
    ]
}

# Response:
# HTTP Status Code
#
# Note: This request is only made once the algorithm has
# submitted all output to the various output resources.


 # Liveness/Readiness Endpoints
 # GET /health/live
 # GET /health/ready
```

Appendix B

```
# API Example with options
# REQUEST
# Header
{
 "Authorization": "ALGO_TOKEN"
}
# Details
{
"transactionID": "MY_TRANSACTION_TOKEN",
"responseURI": "http://foo.bar",
"responseAuthToken": "MY_RESPONSE_TOKEN",
"priority": 128,
"inputMetadata": {
    "workflowStage": "STUDY_ACQUISITION",
    "details": {
        "type": "DICOM_UID",
        "studies": [
            {
            "StudyInstanceUID": "2.16.840.1.114488.0.4.123489834087.13300714",
                "series": [
                    {
                    "SeriesInstanceUID": "2.16.840.1.114488.0.4.123489834087.13300714.2",
                        "instances": [
                          "SOPInstanceUID": ["2.16.840.1.114488.0.4.123489834087.13300714.2.1"],
                            "frames": [
                                "FrameNumber": ["10","11","12"]
                                ]
                            ]
                        }
                    ]
                }
            ]
        }
    },
    "inputResources": [
        {
            "interface": "DICOMWeb",
            "ConnectionDetails": {
                "operations": ["QIDO-RS", "WADO-RS", "WADO-URI"],
                "uri": "http://myvna.net:8000/dicom-web",
                "authID": "TOKEN"
            }
        }
    ],
    "outputResources": [
        {
            "interface": "DICOMWeb",
            "ConnectionDetails": {
                "operations": ["STOW-RS"],
                "uri": "http://myvna.net:8001/dicom-web",
                "authID": "TOKEN"
            }
        }
    ]
}
# Model should respond to request with a valid response code, ie * 200 - ack’d and processing request
# RESPONSE
# Header
{
  "Authorization": "ALGO_TOKEN"
}
# Details
{
    "transactionID": "MY_TRANSACTION_TOKEN",
    "status": 200,
    "message": "Inference completed successfully.",
    "resources": [
        {
            "type": "DICOM_UID",
            "studies": [
                {
                    "StudyInstanceUID": "2.16.840.1.114488.0.4.123489834087.13300714",
                    "series": [
                        {
                        "SeriesInstanceUID": "2.16.840.1.114488.0.4.123489834087.13300714.2.1",
                            "SopInstanceUID": [
                                "2.16.840.1.114488.0.4.123489834087.13300714.2.1.1",
                                "2.16.840.1.114488.0.4.123489834087.13300714.2.2.1"
                            ]
                        }
                    ]
                }
            ]
        }
    ]
}
# The model must also provide Liveness/Readiness Endpoints
# GET /health/live
# GET /health/ready
```