Interface Control Document
Data Competition
Revision 4
July 2024

Defense Advanced Research Projects Agency
Biological Technologies Office
675 North Randolph Street
Arlington, VA 22203-2114
TriageChallenge@darpa.mil

Distribution Statement 'A' (Approved for Public Release, Distribution Unlimited)

1. Contents

2. Introduction

The goal of this document is to convey the high-level concept and infrastructure used to evaluate team submissions in the Data Competition. This document also describes the requirements and interface for successful integration with JHU/APL's evaluation system. This document is specific to the Data Competition for Phase 1. For information on the Systems and Virtual Competitions please refer to their respective *Interface Control Documents*.

The remainder of this document is organized as follows: Section 3 describes the evaluation system, messaging protocol, and the evaluation process; Section 4 describes the evaluation environment for Phase 1; Section 5 describes requirements and resources for submission preparation; and Section 6 describes the procedure for submission. Appendices provide supporting information.

3. Evaluation System

This section contains information about the evaluation system planned for workshop and challenge events in the Data Competition. All formal evaluation procedures will be performed on JHU/APL networks.

Model submissions will be evaluated using a *held-out* test dataset in a simulated online prediction environment, in accordance with the scoring procedure described in the DTC Rules Document. Models will be evaluated on a single patient case at a time. For each patient case, models will be provided with sequential segments of data over the course of the case. As the case unfolds, models are given the opportunity at each segment to predict future LSIs relative to the current segment timestamp. Segments typically contain 5 minutes of data, with smaller or larger windows at the edges of a case. Segments are time-ordered and non-overlapping with only new, unseen data provided in each segment. Models will be responsible for accumulating or storing past data within a case, if necessary. Segments may not be contiguous; there may be time gaps between the end of one segment and the beginning of the next in which no new data was available. As part of the input data for each segment, models will be provided with start and stop timestamps and an indicator for the end of the case (see Section 3.2 for details).

Segmented training datasets will be provided in the same form as data will appear during the evaluation. See the DTC Forum for updates to the training datasets available on AWS. For the Phase 1 challenge event, version 1.2 of the data will be used for evaluation.

Figure 1 provides a high-level description of the three interacting modules in the JHU/APL evaluation system:

1. **Evaluator**, which hosts all logic required to distribute test data to the teams' client containers (via Rabbit MQ) and evaluate their responses;
2. **Rabbit MQ Server**, which hosts the server that governs communication between the Client Container and the Evaluator;
3. **Client Container**, a Docker container which processes input data, runs model inference, and responds with LSI predictions (via Rabbit MQ). Also referred to simply as the Client.

For each evaluation event, teams are expected to provide code for building the Client Container (see Section 5 for development resources). After the submission deadline, AWS administrators will pull submitted code and build the Client Container (see Section 6 for details on submission procedure). This Client Container will then run alongside the Evaluator within the JHU/APL network. During evaluation, the Evaluator sends input data to the Client Container and receives prediction responses via Rabbit MQ's messaging protocol (see Section 3 for more information about the evaluation process and message formats). At the end of the evaluation, the system produces reports with Client responses and performance metrics for each team.

The Rabbit MQ Server and Evaluator will be developed and maintained by JHU/APL. Software submissions must comply with the JHU/APL evaluation system in order to be evaluated. JHU/APL will also provide a Client Shell as a starting point for submission preparation (see Section 5.2). To confirm compliance with the evaluation system, the JHU/APL team will provide a Continuous Integration/Continuous Development (CI/CD) system integrated with the DTC AWS network and teams' code repositories that will build provided code and perform an abbreviated evaluation run to test submission compliance (see Section 5.4). Successful test outcome from the CI/CD indicates that the submission is compliant with the evaluation system. Teams will then need to follow the submission procedure to provide code for evaluation (see Section 6).

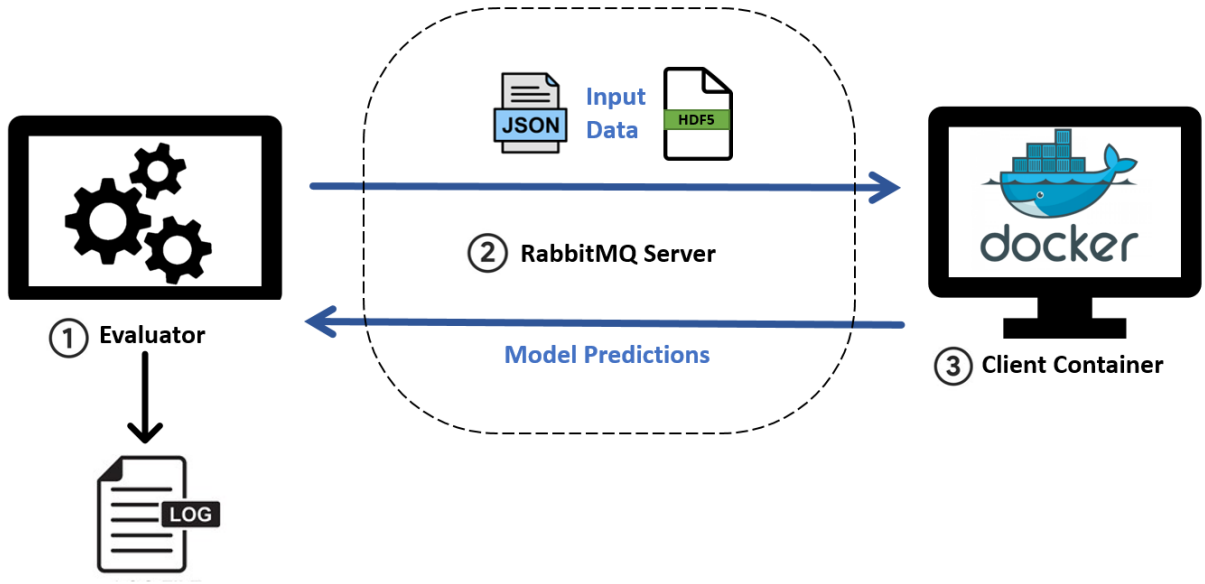

Figure 1: High-level architecture of the JHU/APL evaluation system. *This system is composed of 1) the Evaluator, 2) Rabbit MQ server, and 3) the Client Container, containing the team's model. The Evaluator sends test data to the Client Container using RabbitMQ messages, and the Client Container responds with the LSI predictions.*

In Section 3.1, we describe the main responsibilities of the Evaluator and Client Container and the progression of messages passed between these two components during an evaluation run, as well as the results expected after evaluation. In Section 3.2, we describe the message formats expected between Evaluator and Client Container.

3.1. Evaluation Process

The Evaluator is responsible for accessing the held-out test dataset, packaging and serving data segments to the client container via RabbitMQ, and logging responses from the Client Container.

The Client Container is responsible for housing the submitted model, ingesting data provided by the Evaluator via RabbitMQ, aggregating or storing information across data segments (as needed), preprocessing data and running model inference, and sending model responses with LSI predictions back to the Evaluator.

After the Evaluator establishes connection with the Client Container, the Evaluator will begin sending data segments to the Client Container to perform model inferences and return LSI predictions. Evaluation occurs one case at a time with the Evaluator sending data segments within each case in temporal order from beginning to end.

The Evaluator and Client communicate using the following message formats (described in detail in Section 3.2):

- *Predict* message, containing a data segment provided to the Client for predicting LSIs.
- *Response* message, containing LSI predictions from the Client based off of data received
- *Acknowledge* message, containing status information about evaluation progress
- *Timed Out* message, indicating to the Client that the prediction process timed out
- *Error* message, indicating to the Client that the prediction response was mal-formed
- *Cleanup* message, indicating to the Client that the case is complete and evaluation is continuing with the next case

The following steps describe the sequence of messages sent between the Evaluator and the Client Container during an evaluation run:

1) **Evaluator establishes connection to the Client Container.** An initial message is sent from the Evaluator to the Client Container to establish the RabbitMQ channel for the evaluation run.
2) **Evaluator sends Predict message.** The Evaluator sends a new data segment in a Predict message and waits for a Response message from the Client.
3) **Client performs inference.** The Client receives the Predict message, performs any preprocessing activities and model inference.
4) **Client sends Response message.** The Client sends a response containing the list of predicted LSIs (if any).
5) **Evaluator waits for Response message and saves output.** The Evaluator saves the Response message from the Client for calculating metrics. If the Response message is not received from the Client before the segment duration has elapsed, go to Step 6; if the Response is timely but not well-formed, go to Step 7; otherwise, go to Step 8.
6) **(If applicable) Evaluator sends a Timed Out message and resumes with next case.** If the Client does not produce Response message within time limit (segment duration), then the Evaluator sends a Timed Out message to the Client. The evaluation is then interrupted and resumes at Step 1, starting with the next case in the test dataset.
7) **(If applicable) Evaluator sends an Error message and resumes with next case.** If the Client does not produce a well-formed Response message (e.g., LSI label is misspelled),

then the Evaluator sends an Error message to the Client. The evaluation is then interrupted and resumes at Step 1, starting with the next case in the test dataset.

8) **Evaluator sends Acknowledge message.** The Evaluator acknowledges receipt of a well-formed and timely Response message from the Client with status information about evaluation progress.
9) **Repeat Steps 2-8 for the remaining segments in the case.** The Evaluator continues sending incremental segments of the data for the current case. The Client Container is responsible for storing historical data from previous segments within the same case. For the last segment of the case, the Predict message will indicate the end-of-case using a boolean flag, indicating that the next Predict message will come from a new case.
10) **Evaluator confirms end of case with a Cleanup message.** After the prediction from the last segment of a case is received, the Evaluator sends a "clean-up" message to cue the Client to execute any process before beginning a new case.
11) **Repeat Steps 2-10 for remaining cases.** The Evaluator continues until all cases have been evaluated.

3.1.1. Response log

With each well-formed and timely response from the Client Container (Step 5 above), the following information will be stored to facilitate post-hoc metrics calculation and scoring:

- Patient case identifier
- Segment identifier
- Time window of evaluation segment
- Time elapsed from message sent to response received
- LSI predictions (if any)
- Optional response fields (confidence scores and embeddings, see Section 3.1.7)

This log will be provided to teams after each workshop and challenge event.

3.1.2. Metrics log

After an evaluation run, metrics will be computed from the Client response logs and the test dataset ground truth. This log will include the following metrics as described in the Rules Document for each case:

- Jaccard Index (i.e., time-sensitive intersection-over-union metric)
- Prediction Lead Time

This log will be provided to teams after each workshop and challenge event.

3.2. Communication Protocol

In this section, we describe the contents and format of the following messages sent between the Evaluator and Client: *Predict*, *Acknowledge*, *Timed Out*, *Cleanup*, and *Error*. Further details on implementation are described in Section 5.

3.2.1. Predict Message Format

The Predict message is sent by the Evaluator to the Client for each data segment. It contains EHR data, VS data, and segment metadata. Here is an example of the Predict message format for a single segment:

```
{
  “segment_id”: “OUmJvhCVC5”,
  “case_id”: “wv9ufeR87I”,
  “end_of_case”: False,
  “start_time”: 900,
  “stop_time”: 1800,
  “ehr”: <dict>,
  “vs”: <dict>
}
```

Message 1: Predict message format

Definitions for Predict message fields are as follows:

- *segment_id* is a unique string identifier for the current segment
- *case_id* is a unique string identifier for the current case, composed of many segments
- *end_of_case* is a boolean indicator that this is the last segment for the current case
- *start_time* is the beginning of the segment as time elapsed from start of case in seconds
- *stop_time* is the end of the segment as time elapsed from start of case in seconds
- *ehr* is a dictionary containing Electronic Health Record (EHR) data, described below
- *vs* is a dictionary containing Vital Signs (VS) data, described below

Depending on data availability within each case and segment, the *ehr* or *vs* field may be empty dictionaries or *None*. If present, all fields within these dictionaries are optional. For any available data, field names within both the *ehr* and *vs* dictionaries are consistent with the data dictionary and documentation provided with the training dataset. However, the structure of these dictionaries was simplified from the training dataset as described below.

In response to the Predict message, the Client is expected to respond with a Response message (see Section 3.2.2).

3.2.1.1. *ehr* Data Field

The *ehr* dictionary contains EHR data made available according to the time bounds of the segment, which in total is only a subset of the full training dataset provided. Data are grouped into the following categories:

- **Start-of-Case**. Data available at the beginning of the case, such as GCS taken at the scene, injury type, and general demographic information.
- **At-Admission.** Data available at hospital admission, for example vitals taken at admission.
- **Event Time.** Timestamped data provided according to the segment time window in which they occur, along with timestamps relative to the beginning of the case (in seconds), for example procedures, labs, and medications.

Note that a portion of the EHR training dataset falls in none of these categories and will be excluded completely from the evaluation (for example, fields related to outcome or information not available in the acute period of treatment). Appendix A includes the list of possible fields provided during evaluation within each of the categories above. Appendix B includes example EHR data formatted as JSON files provided at different timepoints during a case.

3.2.1.2. *vs* Data Field

The *vs* dictionary contains Vital Signs (VS) data made available according to the time bounds of the segment. Timeseries data will have timestamps relative to the start of the case (in seconds). The *vs* dictionary will include any available trends and signal data from pre-hospital and in-hospital VS data with an internal structure that mirrors the file structure provided in the training dataset.

Here is an example of the *vs* dictionary containing all VS data sources:

```
{
  "in_hospital":
  {
    "signal": <dict>,
    "trends": <dict>
  },
  "pre_hospital:
  {
    "signal": <dict>,
    "trends": <dict>
  }
}
```

All signal and trends fields contain dictionaries that match the structure and field names of their respective HDF5 files in the training dataset and the accompanying dataset documentation. Only available data will be included, so all fields are optional.

3.2.2. Response Message Format

In response to the Predict message, Clients are expected to respond with any LSI predictions, or an empty list if no predictions are made. Here is an example Response message with two predicted LSIs:

```
{
  “segment_id”: “OUmJvhCVC5”,
  “lsi_predictions”: [“chest_decompression”,“crystalloid_products”]
}
```

Message 2: Response message format

Definitions of the Response message fields are as follows:

- *segment_id* is the unique string identifier for the segment provided in the input message
- *lsi_predictions* is a list of strings, where each string is a predicted LSI label. An empty list will be interpreted as the absence of any LSI predictions.

Both fields above are required in the Response message. For specification of optional fields containing prediction confidence scores and embeddings, see Section 3.2.7.

Table 1 contains the string labels for each LSI group expected in the *lsi_predictions* list, where LSI groups correspond to those used in *LSI_table.csv* in the training dataset.

Table 1: LSI group response labels

LSI GROUP	LABEL
Airway & Respiration	airway_and_respiration
Bleeding Control	bleeding_control
Blood Products	blood_products
Cardiovascular Procedures	cardiovascular_procedures
Chest Decompression	chest_decompression
Crystalloid Products	crystalloid_products
Neurologic Products & Procedures	neurologic_products_and_procedures
RSI Sedation Medications	rsi_sedation_medications
Vascular Access & Monitoring	vascular_access_and_monitoring
Vaso/Cardioactive Medications	vaso_cardioactive_medications
Limb Salvage	limb_salvage
Damage Control Procedures	damage_control_procedures

After the Response message is received by the Evaluator, the Evaluator will send an Acknowledge message (see Section 3.2.3), and the evaluation will continue with the next segment. A Response message must be received within the segment duration from when the corresponding Predict message was sent, otherwise a Timed Out message is sent by the Evaluator (see Section 3.2.6).

3.2.3. Acknowledge Message Format

The Evaluator will send an Acknowledge message to the Client to indicate successful receipt of a prediction. Here is an example of an Acknowledge message:

```
{
  "case_id": "s1kojt25",
  "segment_id": "OUmJvhCVC5",
  "delta_runtime_sec": 0.2,
  "runtime_remaining_sec": 144000.8,
  "cases_remaining": 240
}
```

Message 3: Acknowledge message format

Definitions of the Acknowledge message fields are as follows:

- *case_id* is the unique string identifier for the current case
- *segment_id* is the unique string identifier for the current segment

- *delta_runtime_sec* is the time elapsed in seconds from Predict message sent to Response message received by the Evaluator, which contributes to the total evaluation runtime
- *runtime_remaining_sec* is the total evaluation runtime limit minus the cumulative runtime in seconds
- *cases_remaining* is the number of cases remaining in the evaluation

There is no specific content required from the Client in response to an Acknowledge message.

3.2.4. Cleanup Message Format

Following the last segment in a case, the Evaluator will send a Cleanup message to the Client indicating the case has ended and the next Predict message will start a new case. Here is an example of a Cleanup message:

```
{
  "case_id": "s1kojt25",
  "runtime_remaining_sec": 144000.8,
  "cases_remaining": 240
}
```

Message 4: Cleanup message format

Definitions of the Cleanup message fields are as follows:

- *case_id* is the unique string identifier for the current case
- *runtime_remaining_sec* is the total evaluation runtime limit minus the cumulative runtime in seconds
- *cases_remaining* is the number of cases remaining in the evaluation

There is no specific content required from the Client in response to a Cleanup message.

3.2.5. Error Message Format

In the event of an error in the evaluation (e.g., malformed Response message from the Client), the Evaluator will send an Error message to the Client. Here is an example of an Error message:

```
{
  "case_id": "s1kojt25",
  "segment_id": "OUmJvhCVC5",
  "error_message": <string>
}
```

Message 5: Error message format

Definitions of the Error message fields are as follows:

- *case_id* is the unique string identifier for the current case
- *segment_id* is the unique string identifier for the current segment
- *error_message* is a free-text message describing the error encountered

There is no specific content required from the Client in response to an Error message.

3.2.6. Timed Out Message Format

After a Predict message is sent, the corresponding Response message must be received from the Client before the time elapsed exceeds the segment duration. If no Response message is received before this time limit, a Timed Out message is sent by the Evaluator to the Client.

Here is an example of a Timed Out message sent by the Evaluator to the Client:

```
{
  "case_id": "s1kojt25",
  "segment_id": "OUmJvhCVC5",
  "delta_runtime_sec": 0.2,
  "runtime_remaining_sec": 144000.8,
  "cases_remaining": 240
}
```

Message 6: Timed Out message format

Definitions of the Timed Out message fields are as follows:

- *case_id* is the unique string identifier for the current case
- *segment_id* is the unique string identifier for the current segment
- *delta_runtime_sec* is the time elapsed in seconds from Predict message sent to Response message received by the Evaluator
- *runtime_remaining_sec* is the total evaluation runtime limit minus the cumulative runtime in seconds
- *cases_remaining* is the number of cases remaining in the evaluation

There is no specific content required from the Client in response to a Timed Out message.

3.2.7. Optional Response Message Fields

To support greater interpretability of and ability to analyze performance of competitors' models, there are additional fields that can be returned in the Response message. All of these fields are optional and do not factor into a competitor's event score.

Many machine learning models operate by constructing vector representations (also called embeddings) of input data before performing additional processing to produce an output prediction. Providing access to this internal data representation can help improve interpretability of model behavior and provide additional surface area to diagnose and resolve issues with models. The schema for returning these embedding values is meant to help support this type of analysis. Given that there are multiple levels at which teams' models may be operating, we suggest fields for embeddings representing both the segment level and the cumulative level, as well as a catch-all "other" category. None of this is prescriptive of how competitors approach the challenge, but rather a best hypothesis for the kinds of model-internal information that may be available.

Here is an example Response message that includes the optional fields:

```
{
  “segment_id”: “OUmJvhCVC5”,
  “lsi_predictions”: [“chest_decompression”,“crystalloid_products”],
  “embeddings”: <dict>,
  “lsi_confidence_scores”: <dict>
}
```

Message 7: Response message format with optional fields

Optional fields are defined as follows and described in detail in subsections below:

- *embeddings* is a dictionary containing embeddings at different levels of internal data analysis
- *lsi_confidence_scores* is a dictionary containing confidence scores for each LSI group

3.2.7.1. **embeddings** Response Field

The *embeddings* response field contains a dictionary with the following format:

```
{
  "vs_segment": vs_segment_embedding,
  "vs_cumulative": vs_cumulative_embedding,
  "ehr_segment": ehr_segment_embedding,
  "ehr_cumulative": ehr_cumulative_embedding,
  "case_segment": case_segment_embedding,
  "case_cumulative": case_cumulative_embedding,
  "other": [other_embeddings_1,  other_embeddings_2, …]
}
```

Message 8: *embeddings* dictionary format

Note that each individual field in the dictionary above is optional. The value for each embedding vector is expected to be a list of numbers, except for the “other” field, which is expected to be a list of such embedding vectors (i.e., a list of lists).

Embedding vector dimensionality may differ between fields; however, for a given field, the dimensionality should be consistent across all segments. For example, “vs_segment” and “vs_cumulative” embeddings may have different dimensionality, but their respective dimensionality should be the same for each data segment.

Embedding fields description:

vs_segment
: An embedding representing the vital signs data from the current data segment.

vs_cumulative
: An embedding representing the cumulative vital signs signals across prior segments up to and including this segment.

ehr_segment
: An embedding representing the EHR data from the current data segment.

ehr_cumulative

An embedding representing the cumulative EHR data across prior segments up to and including this segment.

case_segment

A higher-level embedding representing the overall case status given the current segment.

case_cumulative

A higher-level embedding representing the overall case status given the cumulative data across prior segments up to and including this segment.

other

Any other embedding(s) that do not fit into the schema above may be included here.

3.2.7.2. **lsi_confidence_scores** Response Field

The value associated with this field is also a dictionary. It should contain a key for **each** of the 12 possible LSI groups (see Table 1) mapped to a confidence score between 0 and 1.

For example:

```
{
  "rsi_sedation_medications": 0.892857821,
  "airway_and_respiration": 0.910848272,
  "limb_salvage": 0.001482986,
    ...
  "vaso_cardioactive_medications": 0.7823
}
```

Message 9: lsi_confidence_scores dictionary format

We reiterate that even if an LSI is not included in the list of "lsi_predictions", it will be useful for its confidence score to be returned in the "lsi_confidence_scores" dictionary.

3.2.7.3. Response Message Format, Including Optional Fields

Given the above specification for the optional fields "embeddings" and "lsi_confidence_scores", here is an example client response message that includes the optional fields:

```
{
  "segment_id": "OUmJvhCVC5",
  "lsi_predictions": ["chest_decompression"],
  "lsi_confidence_scores": {
    "airway_and_respiration": 0.0015,
    "bleeding_control": 0.01163,
    ...
    "vaso_cardioactive_medications": 0.7823
  },
  "embeddings": {
    "case_segment": [0.152, 0.023, …, 0.134],
    "case_cumulative": [0.693, 0.193, …, 0.081]
    "other": [
      [0.232, 0.024, …, 0.817],
      [0.251, 0.004, …, 0.948]
    ]
  }
}
```

Message 10: Example response including optional diagnostic fields

4. Evaluation Environment

We are expecting the following hardware environment and constraints for the Phase 1 Challenge Event:

CPU	1 CPU w/ 8 cores @ 3.50 GHz
RAM	100 GB
GPU	Titan V (12 GB VRAM, single GPU)
GPU DRIVER	NVIDIA-DRIVER Version 510.47.03
NETWORK	No access
TEST CASES	~850 cases (~29,000 data segments)
TIME LIMITATION	48 hours

5. Submission Development

Each team must provide code that is compliant with the evaluation process. The following section describes submission requirements, resources, and tools for testing with the JHU/APL evaluation system.

5.1. Requirements

The following subsections describe the minimum requirements for successful submissions.

5.1.1. DTC Base Image

Submissions must be built from an approved DTC Base Image containing the required package *dtc_messaging* for interacting with the Evaluator. The following base images are provided in AWS Elastic Container Registry (ECR):

- *dtc-base-image:latest* for GPU support in Client development
- *dtc-base-image-cpu:latest* for CPU-only support in Client development

Code used to build these images is provided on AWS CodeCommit: https://git-codecommit.us-east-1.amazonaws.com/v1/repos/dtc-base-image

5.1.2. DTC Base Model

Submissions must use a model class that inherits from *DTC_BaseModel* class (included in the *dtc_messaging* package within the DTC Base Image). Callback functions are used to respond to message types described in Section 3. The model class must implement the following callback methods:

- `predict()`: receives a Predict message and returns a Response message
- `acknowledge()`: receives an Acknowledge message, no return message required

- `error()`: receives an Error message, no return message required
- `timed_out()`: receives a Timed Out message, no return message required
- `cleanup()`: receives a Cleanup message, no return message required

An example implementation of the *DTC_BaseModel* class is provided in `template_model.py` in the Client Shell (see Section 5.2).

5.1.3. Docker Entrypoint

The following entrypoint and command will be used to run Client Container submissions within the evaluation system:

```
command: "./run_client.py --host <host> --queue <queue>"
entrypoint: ["python3"]
```

where `<host>` and `<queue>` will be modified at evaluation time to set the host and RabbitMQ queue name. Submissions are required to provide `run_client.py` in the working directory of the Docker container with command line arguments `--host` and `--queue`. Any additional command line arguments will be ignored during evaluation. Additional parameters required by the submission may be included in a configuration file within the Docker container.

An example implementation of the `run_client.py` script is provided in the Client Shell (see Section 5.2).

5.1.4. Client Log

A volume will be mounted to the Client Container where log files may be saved for delivery back to teams in AWS after an evaluation event. This is an optional feature, and any log files will not impact team scores, nor will they be shared between teams. Any log files should be given unique filenames (e.g., with timestamp) so as not to be overwritten by subsequent evaluation runs. Importantly, EHR or VS data (including derived data) should not be saved in log files.

The following directory will be mounted as a volume to the Client Container for saving log files within the Docker container: `/usr/src/app/logs`

5.2. Client Shell

To assist in developing compliant submissions, JHU/APL has provided a Client Shell that includes the minimal code needed to create a functioning Client Container. Teams may incorporate their own packages and model-specific code to the Client Shell to build their submission.

The Client Shell contains the following resources:

- `run_client.py`: script used to run the Client
- `template_model.py`: model implementation of *DTC_BaseModel*
- `Dockerfile`: example to build Client Container from *dtc-base-image* and install additional dependencies

- `ReadMe.md`: additional information on Client Shell usage, also available on the AWS wiki at: https://github.com/JHUAPL-DTC-TA2/wiki/blob/main/Running%20Client%20Shell%20in%20SageMaker.md

The Client Shell is provided on AWS CodeCommit: https://git-codecommit.us-east-1.amazonaws.com/v1/repos/client-shell

5.3. Evaluator Container Testing

Docker images containing the DTC Evaluator (*dtc-evaluator:latest*) and the RabbitMQ Server (*dtc-rabbitmq:latest*) will be released to participants to assist with testing Clients within AWS prior to submitting for evaluation. See future posts to the DTC forum and documentation on the AWS wiki for more information.

5.4. CodeBuild (CI/CD) Compliance Testing

JHU/APL will provide a Continuous Integration/Continuous Development (CI/CD) system within AWS to give automatic feedback on submission compliance with the evaluation system. This system will containerize selected code using a standard *buildspec* and perform an abbreviated evaluation run to assess the code's compliance with the evaluation system.

To submit code to the CI/CD system, teams should push code to the *compliance-test* branch of their team repository. This action will automatically trigger the CI/CD system to pull the repository and build code with the most recent commit on the *compliance-test* branch. Once the build is complete, the CI/CD system will perform an evaluation run using a small validation dataset to assess compliance with the evaluation system. Any artifacts (e.g., logs) produced by the compliance test will be provided to teams for review within their scratch bucket under the *build_logs* directory.

Usage of the CI/CD system is voluntary, and there is no limit to the number of times a team can test code through the CI/CD system. However, costs related to running the CI/CD system will be subtracted from the team's budget. These costs are expected to be minimal, but code should be pushed to the *compliance-test* branch sparingly to minimize budget usage.

Additional information and release updates to the CI/CD system will be posted to the DTC Forum.

6. Submission Procedure

Code will be pulled down from CodeCommit upon the submission deadline using a specific git tag pushed to the team repository on CodeCommit. Tag nomenclature for event submissions should use the following convention:

submission-phase<PHASE_NUMBER>-<EVENT_TYPE>

where PHASE_NUMBER∈{1, 2, 3} and EVENT_TYPE∈{workshop, challenge}. For example, for the Phase 1 Challenge event, submissions should be tagged as:

submission-phase1-challenge

This procedure ensures that a specific, unambiguous commit is evaluated as the official submission. The git tag may be moved to a different commit ahead of the submission deadline, however changing the git tag after submission deadline will not be possible. The submission git tag will also trigger a CI/CD compliance test to ensure submitted code complies with the evaluation system.

7. Appendix A – Fields Provided During Evaluation

Table A.1: Start-of-Case UMB Fields

Table	Field
demo_scores	SEXID
demo_scores	RACEID
demo_scores	race_descrip
demo_scores	AGE
demo_scores	INJTYPEID
demo_scores	INJURYTYPEDESCRIP
demo_scores	ICD10ECODE
demo_scores	RTS_S
demo_scores	HEIGHT
demo_scores	WEIGHT
ems	LANDVSAIR

Table A.2: At-Admission UMB Fields

Table	Field
demo_scores	ADM_TEMP
demo_scores	ADM_SYSBP
demo_scores	ADM_DYSBP
demo_scores	ADM_HR
demo_scores	ADM_RR
demo_scores	ADM_O2SAT
demo_scores	ADM_GCS_EYE
demo_scores	ADM_GCS_VERBAL
demo_scores	ADM_GCS_MOTOR
demo_scores	GCSTOTAL
demo_scores	BRAINSEV
demo_scores	FACESEV
demo_scores	NECKSEV
demo_scores	THORAXSEV

demo_scores	ABDSEV
demo_scores	SPINESEV
demo_scores	UPPEREXTSEV
demo_scores	LOWEREXTSEV
demo_scores	ISS
demo_scores	TRISS
demo_scores	RTS_A
injury	AISSEVERITY
injury	ISSBODYREGION
injury	ICD10CODE
LSI_table*	lsi_group
LSI_table*	lsi_description
LSI_table*	in_hospital

*Note: Pre-hospital LSI records are provided at-admission (in_hospital = 0).

Table A.3: Event Time UMB Fields

Table	Field	Timestamp Field
death	BRAINDEATH_elapsed_from_start†	BRAINDEATH_elapsed_from_adm
death	WITHDRAWAL_CARE_elapsed_from_start†	WITH-DRAWAL_CARE_elapsed_from_adm
labs	OBSDATETIME_elapsed_from_start†	OBSDATETIME_elapsed_from_adm, RSLTDATETIME_elapsed_from_adm
labs	COMPTEXT	OBSDATETIME_elapsed_from_adm, RSLTDATETIME_elapsed_from_adm
labs	TESTTEXT	OBSDATETIME_elapsed_from_adm, RSLTDATETIME_elapsed_from_adm
labs	RSLTDATETIME_elapsed_from_start†	RSLTDATETIME_elapsed_from_adm
labs	RSLT	RSLTDATETIME_elapsed_from_adm
labs	UNITS	RSLTDATETIME_elapsed_from_adm
non_op_procs	DESCRIP	elapsed_from_adm
non_op_procs	elapsed_from_start†	elapsed_from_adm
operations	CPT	OR_START_TIME_elapsed_from_adm, OR_STOP_TIME_elapsed_from_adm

operations	SERVICEID	OR_START_TIME_elapsed_from_adm, OR_STOP_TIME_elapsed_from_adm
operations	SERVICE_DESCRIP	OR_START_TIME_elapsed_from_adm, OR_STOP_TIME_elapsed_from_adm
operations	PROCTEXT	OR_START_TIME_elapsed_from_adm, OR_STOP_TIME_elapsed_from_adm
operations	OR_START_TIME_elapsed_from_start†	OR_START_TIME_elapsed_from_adm, OR_STOP_TIME_elapsed_from_adm
operations	OR_STOP_TIME_elapsed_from_start†	OR_STOP_TIME_elapsed_from_adm
medications	display_name	elapsed_from_adm
medications	mar_action	elapsed_from_adm
medications	sig	elapsed_from_adm
medications	dose_unit	elapsed_from_adm
medications	route	elapsed_from_adm
medications	pat_loc	elapsed_from_adm
medications	pharm_class	elapsed_from_adm
medications	pharm_subclass	elapsed_from_adm
medications	thera_class	elapsed_from_adm
medications	elapsed_from_start†	elapsed_from_adm
pta_vitals*	PTA_SBP	elapsed_from_adm
pta_vitals*	PTA_DBP	elapsed_from_adm
pta_vitals*	PTA_HR	elapsed_from_adm
pta_vitals*	PTA_RR	elapsed_from_adm
pta_vitals*	PTA_TEMP	elapsed_from_adm
pta_vitals*	PTA_GCS_TOTAL	elapsed_from_adm
pta_vitals*	PTA_GCS_V	elapsed_from_adm
pta_vitals*	PTA_GCS_M	elapsed_from_adm
pta_vitals*	PTA_GCS_E	elapsed_from_adm
pta_vitals*	elapsed_from_start†	elapsed_from_adm
LSI_table	lsi_group	elapsed_from_adm
LSI_table	lsi_description	elapsed_from_adm
LSI_table	in_hospital	elapsed_from_adm
LSI_table	elapsed_from_start†	elapsed_from_adm

*Note: if “elapsed_from_adm” in pta_vitals is *null*, this record will be provided at-admission.
†Elapsed times will be provided from start of case.

Table A.4: Start-of-Case UPitt Fields

Table	Field
airway	AirwayOutcome
airway	AirwayPerformedBy
airway	AirwaySecuredVia
airway	AirwayStatus
burns	BurnsMethod
burns	BurnsPerc1stDegree
burns	BurnsPerc2ndDegree
burns	BurnsPerc3rdDegree
burns	BurnsPercArmsAnterior
burns	BurnsPercArmsPoterior
burns	BurnsPercHeadAnterior
burns	BurnsPercHeadPoterior
burns	BurnsPercLegsAnterior
burns	BurnsPercLegsPoterior
burns	BurnsPercTrunkAnterior
burns	BurnsPercTrunkPoterior
burns	CarboninMouth
burns	PoorVentilation
burns	SingedNasalHairs
burns	TBSA
burns	Type
burns	CO
fluids	INBeforeCOLL
fluids	INBeforeCrys
fluids	INBeforeOther
fluids	OUTBeforeEBL
fluids	OUTBeforeOther
fluids	OUTBeforeUO

hosp_wide	etiology
hosp_wide	inj_type
injurydetails	DrugsAlcoholIndicator
injurydetails	DrugsAlcoholIndicator1
injurydetails	DrugsAlcoholIndicator2
injurydetails	DrugsAlcoholIndicator3
injurydetails	DrugsAlcoholIndicator4
injurydetails	DrugsAlcoholIndicator5
injurydetails	FallHeight
injurydetails	FallHeightUnits
injurydetails	FallSurface
injurydetails	InjuryCause
injurydetails	IntentionalInjury
injurydetails	LandedOn
injurydetails	ReasonForEncounter
injurydetails	WorkRelated
injurydetails	InjuryMechanism
injurydetails	InjuryMechanism1
injurydetails	InjuryMechanism2
injurydetails	InjuryMechanism3
injurydetails	TraumaCenterCriterion
injurydetails	TraumaCenterCriterion1
injurydetails	TraumaCenterCriterion2
injurydetails	TraumaCenterCriterion3
injurydetails	TraumaCenterCriterion4
injurydetails	TraumaCenterCriterion5
ipfmisc	BurnArea
ipfmisc	BurnBodySide
ipfmisc	BurnDegree
ipfmisc	BurnPercentage
ipfmisc	Impression
ipfmisc	AbdomenAppearance

ipfmisc	AbdomenPalpation
ipfmisc	BackBoardPTA
ipfmisc	CCollarPTA
ipfmisc	CIDPTA
ipfmisc	ImmobilizationPTAOutcome
ipfmisc	ImmobilizationPTAPerformedBy
ipfmisc	KEDPTA
ipfmisc	Trachea
ipfmisc	InjurySymptom
ipfmisc	PrimaryInjurySystem
ipfspc	MentalAssessment
ipfspc	NeurologicalAssessment
ipfspc	SkinAssessment
neuro	LevelofConsciousness
neuro	ChemParalyzed
neuro	InitialGCSEye
neuro	InitialGCSMotor
neuro	InitialGCSTotal
neuro	InitialGCSVerbal
neuro	LossofConsciousness
neuro	MotorLA
neuro	MotorLL
neuro	MotorRA
neuro	MotorRL
neuro	PupilReactivityL
neuro	PupilReactivityR
neuro	PupilSizeL
neuro	PupilSizeR
neuro	RevisedTraumaScore
neuro	RevisedTraumaScoreBP
neuro	RevisedTraumaScoreResp
neuro	SensoryLA

neuro	SensoryLL
neuro	SensoryRA
neuro	SensoryRL
patient	AgeInYears
patient	AgeType
patient	ChiefComplaint
patient	ChiefComplaintDuration
patient	Gender
patient	Height
patient	HeightType
patient	MedicalNecessity
patient	Race
patient	ReasonforInterfacilityTransfer
patient	Weight
patient	PatientAge
patient	WeightType
patient	PatientActivity
priorivs	IVGauge
priorivs	IVRate
priorivs	IVSeq
priorivs	IVSite
priorivs	IVSolution
priormeds	MedRoute
priormeds	MedConcentration
priormeds	MedDose
priormeds	MedDrip
priormeds	MedName
priormeds	MedNameCoded
priormeds	MedRouteName
priormeds	MedSeq
priormeds	RouteCoded
scene	AnatomicLocation

scene	AnatomicLocation1
scene	AnatomicLocation2
scene	AnatomicLocation3
scene	AnatomicLocation4
scene	AnatomicLocation5
scene	AnatomicLocation6
scene	AnatomicLocation7
scene	Belongings
scene	FinalAcuity
scene	InitialPatientAcuity
scene	NumberPatientsAtScene
scene	SceneDescription

Table A.5: At-Admission UPitt Fields

Table	Field
cardiac	AEDUse
cardiac	CardiacArrestPresent
cardiac	CPRBy
cardiac	DefibBy
cardiac	DefibrillatorType
cardiac	EndEvent
cardiac	Etiology
cardiac	InitialRhythm
cardiac	ReasonTerminated
cardiac	Resuscitation1
cardiac	Resuscitation2
cardiac	TherapeuticHypothermia
cardiac	WhoWitnessed
cardiac	CPRType
cardiac	CPRType1
cardiac	CPRType2
cardiac	CPRType3

cardiac	CPRType4
cardiac	DestRhythm
cardiovascular	CapillaryRefill
cardiovascular	Edema
cardiovascular	JVD
cardiovascular	PulseBrachialL
cardiovascular	PulseBrachialR
cardiovascular	PulseCarotidL
cardiovascular	PulseCarotidR
cardiovascular	PulseFemoralL
cardiovascular	PulseFemoralR
cardiovascular	PulseRadialL
cardiovascular	PulseRadialR
cardiovascular	TempDegreesType
cardiovascular	Temperature
cardiovascular	TemperatureObtainedMethod
cardiovascular	InvasiveArterialLine
drain	ChestTubeSizeL
drain	ChestTubeSizeR
drain	ChestTubeSuctionL
drain	ChestTubeSuctionR
drain	Colostomy
drain	FoleySize
drain	Hematuria
drain	Ileostomy
drain	NGSuction
drain	NGTubeSize
drain	OGSuction
drain	OGTube
drain	Other
drain	PEGCapped
drain	PEGTube

drain	Ventriculostomy
drain	VentriculostomyColor
patient	BarriersToCareNew
patient	BloodType
receiving	HospitalUnit
receiving	ModetoRec
receiving	ReceivingHospitalUnit
receiving	ReceivingID
receiving	ReceivingType
receiving	RecHospDesignation
receiving	ReceivingModeDescriptor
receiving	ReferringModeDescriptor
receiving	ReceivingName
respiratory	BreathSoundsLeft
respiratory	BreathSoundsRight
respiratory	OxygenDeliveryMethod
respiratory	OxygenFlow
respiratory	RespEffort
scene	FinalAcuity
ventilator	BreathType_vent
ventilator	FIO2_vent
ventilator	IERation_vent
ventilator	InspPressure_vent
ventilator	ITime_vent
ventilator	MeanAirwayPressure_vent
ventilator	MV_vent
ventilator	PEEP_vent
ventilator	PIP_vent
ventilator	PressureSupport_vent
ventilator	Rate_vent
ventilator	RespRateM_vent
ventilator	TV_vent

ventilator	Ino_vent
ventilator	PTA
ventilator	VentMode
hosp_ais	ais_severity_issbodyregion
hosp_wide	hospital
hosp_wide	destination
hosp_wide	age
hosp_wide	sex
hosp_wide	ecode
hosp_wide	sbp
hosp_wide	resp
hosp_wide	gcs
hosp_wide	gcs_m
hosp_wide	iss
hosp_wide	ethnicity
hosp_wide	etiology
hosp_wide	hgt_fall
hosp_wide	extric_s
hosp_wide	sign_life
hosp_wide	base_def
hosp_wide	blood_pre
hosp_wide	pulse_a
hosp_wide	pupil_a
hosp_wide	par_drg_a
hosp_wide	intubat_a
hosp_wide	temp_c
hosp_wide	lsys_bp_s
hosp_wide	lsys_bp_1
hosp_wide	lsys_bp_r
hosp_wide	lsys_bp_2
hosp_wide	maximum_ais_1
hosp_wide	maximum_ais_2

hosp_wide	maximum_ais_3
hosp_wide	maximum_ais_4
hosp_wide	maximum_ais_5
hosp_wide	maximum_ais_6
hosp_wide	max_sev_aisbr_1
hosp_wide	max_sev_aisbr_2
hosp_wide	max_sev_aisbr_3
hosp_wide	max_sev_aisbr_4
hosp_wide	max_sev_aisbr_5
hosp_wide	max_sev_aisbr_6
hosp_wide	max_sev_aisbr_7
hosp_wide	max_sev_aisbr_8
hosp_wide	max_sev_aisbr_9
hosp_wide	race
hosp_wide	transfer
hosp_wide	rts
hosp_wide	pstriss
hosp_icd_dx	icd9dx
hosp_icd_dx	icd10dx
hosp_trqt	trqt_use
hosp_trqt	trqt
hosp_trqt	trqt_place
hosp_trqt	trqt_type

Table A.6: Event Time UPitt Fields

Table	Field	Timestamp Field
events	BloodPressureDiastolic	elapsed_from_adm
events	BloodPressureSystolic	elapsed_from_adm
events	BPMethod	elapsed_from_adm
events	Carboxyhemoglobin	elapsed_from_adm
events	ECGMethod	elapsed_from_adm
events	EndotrachealCO2	elapsed_from_adm

events	EndotrachealCO2Type	elapsed_from_adm
events	GCSEye	elapsed_from_adm
events	GCSMotor	elapsed_from_adm
events	GCSVerbal	elapsed_from_adm
events	Glucose	elapsed_from_adm
events	HeartRate	elapsed_from_adm
events	HeartRateMethod	elapsed_from_adm
events	LevelofConsciousness	elapsed_from_adm
events	MeanArterialPressure	elapsed_from_adm
events	OxygenSaturation	elapsed_from_adm
events	Procedure	elapsed_from_adm
events	PulseRhythm	elapsed_from_adm
events	Respiration	elapsed_from_adm
events	RespiratoryEffort	elapsed_from_adm
events	RhythmCoded	elapsed_from_adm
events	TempDegreesType	elapsed_from_adm
events	Temperature	elapsed_from_adm
events	TemperatureObtainedMethod	elapsed_from_adm
events	VitalRhythms	elapsed_from_adm
events	elapsed_from_start†	elapsed_from_adm
ipfgen	AssessmentLocation	elapsed_from_adm
ipfgen	elapsed_from_start†	elapsed_from_adm
labs	ALB	elapsed_from_adm
labs	AlcoholOnBreath	elapsed_from_adm
labs	ALT	elapsed_from_adm
labs	APAP	elapsed_from_adm
labs	APHI	elapsed_from_adm
labs	AST	elapsed_from_adm
labs	BE	elapsed_from_adm
labs	BGAccess	elapsed_from_adm
labs	BILI	elapsed_from_adm
labs	BloodAlcohol	elapsed_from_adm

labs	BNP	elapsed_from_adm
labs	BUN	elapsed_from_adm
labs	CA	elapsed_from_adm
labs	CL	elapsed_from_adm
labs	CO2	elapsed_from_adm
labs	COHB	elapsed_from_adm
labs	CPK	elapsed_from_adm
labs	CR	elapsed_from_adm
labs	DDimer	elapsed_from_adm
labs	GLU	elapsed_from_adm
labs	HcgSerum	elapsed_from_adm
labs	HcgUrine	elapsed_from_adm
labs	HCO3	elapsed_from_adm
labs	HCT	elapsed_from_adm
labs	HGB	elapsed_from_adm
labs	INR	elapsed_from_adm
labs	IonizedCalcium	elapsed_from_adm
labs	K	elapsed_from_adm
labs	LactateArterial	elapsed_from_adm
labs	LactateVenous	elapsed_from_adm
labs	Lipase	elapsed_from_adm
labs	MG	elapsed_from_adm
labs	NA	elapsed_from_adm
labs	PCO2	elapsed_from_adm
labs	PH	elapsed_from_adm
labs	PLTS	elapsed_from_adm
labs	PO2	elapsed_from_adm
labs	PT	elapsed_from_adm
labs	PTA	elapsed_from_adm
labs	PTT	elapsed_from_adm
labs	RBC	elapsed_from_adm
labs	SAT	elapsed_from_adm

labs	SVO2	elapsed_from_adm
labs	TBILI	elapsed_from_adm
labs	Troponin	elapsed_from_adm
labs	WBC	elapsed_from_adm
labs	elapsed_from_start†	elapsed_from_adm
proclabs	LabIonizedCalcium	elapsed_from_adm
proclabs	LabLactateArterial	elapsed_from_adm
proclabs	LabLactateVenous	elapsed_from_adm
proclabs	LabPTA	elapsed_from_adm
proclabs	LabsBE	elapsed_from_adm
proclabs	LabsBGAccess	elapsed_from_adm
proclabs	LabsCl	elapsed_from_adm
proclabs	LabsGlu	elapsed_from_adm
proclabs	LabsHCO3	elapsed_from_adm
proclabs	LabsHgb	elapsed_from_adm
proclabs	LabsK	elapsed_from_adm
proclabs	LabsNa	elapsed_from_adm
proclabs	LabsOther	elapsed_from_adm
proclabs	LabsPCO2	elapsed_from_adm
proclabs	LabspH	elapsed_from_adm
proclabs	LabspO2	elapsed_from_adm
proclabs	LabsSAT	elapsed_from_adm
proclabs	LabSVO2	elapsed_from_adm
proclabs	elapsed_from_start†	elapsed_from_adm
procmain	AirwayAction	elapsed_from_adm
procmain	AirwayOxygenFlow	elapsed_from_adm
procmain	CardiacAction	elapsed_from_adm
procmain	CardiacDefibType	elapsed_from_adm
procmain	CardiacPacingEnergy	elapsed_from_adm
procmain	CardiacPacingMode	elapsed_from_adm
procmain	CardiacPacingRate	elapsed_from_adm
procmain	CardiacShockEnergy	elapsed_from_adm

procmain	CardiacShockType	elapsed_from_adm
procmain	DrainAction	elapsed_from_adm
procmain	DrainSide	elapsed_from_adm
procmain	DrainSize	elapsed_from_adm
procmain	DrainSizeUnit	elapsed_from_adm
procmain	DrainSuction	elapsed_from_adm
procmain	HospNotifyAlertType	elapsed_from_adm
procmain	HospNotifyMethod	elapsed_from_adm
procmain	HospNotifyRegistryCandidate	elapsed_from_adm
procmain	ImmobOption	elapsed_from_adm
procmain	InitiateIVGauge	elapsed_from_adm
procmain	InitiateIVSite	elapsed_from_adm
procmain	InitiateIVType	elapsed_from_adm
procmain	IntubationCMatLips	elapsed_from_adm
procmain	IntubationCuffFill	elapsed_from_adm
procmain	IntubationCuffFillQty	elapsed_from_adm
procmain	IntubationLaryngoscopicGrade	elapsed_from_adm
procmain	IntubationLarynogoscopeBlade	elapsed_from_adm
procmain	IntubationMallampati	elapsed_from_adm
procmain	IntubationMethod	elapsed_from_adm
procmain	IntubationRSIProtocol	elapsed_from_adm
procmain	IntubationSellickManeuver	elapsed_from_adm
procmain	IntubationSize	elapsed_from_adm
procmain	IntubationStyletUsed	elapsed_from_adm
procmain	MedicalConsultMethod	elapsed_from_adm
procmain	MedicationDosage	elapsed_from_adm
procmain	MedicationDosageUnit	elapsed_from_adm
procmain	MedicationName	elapsed_from_adm
procmain	MedicationRate	elapsed_from_adm
procmain	MedicationRateUnit	elapsed_from_adm
procmain	MedicationRoute	elapsed_from_adm
procmain	Procedure	elapsed_from_adm

procmain	ProcedureAttempts	elapsed_from_adm
procmain	ProcedureAuthorization	elapsed_from_adm
procmain	elapsed_from_start†	elapsed_from_adm
procmain	ProcedureResponse	elapsed_from_adm
procmain	ProcedureSuccessful	elapsed_from_adm
procmain	TitrateMedication	elapsed_from_adm
procmain	TitrateNewDose	elapsed_from_adm
procmain	TitrateNewDoseRoute	elapsed_from_adm
procmain	TitrateNewDoseUnit	elapsed_from_adm
procmain	ComplicationAirwayIntubation	elapsed_from_adm
procmain	ComplicationAirwayIntubation1	elapsed_from_adm
procmain	ComplicationAirwayIntubation2	elapsed_from_adm
procmain	ComplicationAirwayIntubation3	elapsed_from_adm
procmain	ComplicationAirwayIntubation4	elapsed_from_adm
procmain	ComplicationAirwayIntubation5	elapsed_from_adm
procmain	ProcedureComplication	elapsed_from_adm
procmain	ProcedureComplication1	elapsed_from_adm
procmain	ProcedureComplication2	elapsed_from_adm
procmain	ProcedureComplication3	elapsed_from_adm
procmain	ProcedureComplication4	elapsed_from_adm
procmain	ProcedureComplication5	elapsed_from_adm
procmain	ProcedureFailureReason	elapsed_from_adm
procmain	ProcedureFailureReason1	elapsed_from_adm
procmain	ProcedureFailureReason2	elapsed_from_adm
procmain	ProcedureFailureReason3	elapsed_from_adm
procmain	ProcedureFailureReason4	elapsed_from_adm
procother	ImmobilizationAction	elapsed_from_adm
procother	ImmobilizationAction1	elapsed_from_adm
procother	ImmobilizationAction2	elapsed_from_adm
procother	ImmobilizationAction3	elapsed_from_adm
procother	ImmobilizationAction4	elapsed_from_adm
procother	ImmobAssessment	elapsed_from_adm

procother	ImmobAssessment1	elapsed_from_adm
procother	ImmobAssessment2	elapsed_from_adm
procother	ImmobAssessment3	elapsed_from_adm
procother	ImmobAssessment4	elapsed_from_adm
procother	ImmobAssessment5	elapsed_from_adm
procother	ImmobSite	elapsed_from_adm
procother	ImmobSite1	elapsed_from_adm
procother	ImmobSite2	elapsed_from_adm
procother	ImmobSite3	elapsed_from_adm
procother	ImmobSite4	elapsed_from_adm
procother	elapsed_from_start†	elapsed_from_adm
procother	IntubationIndication	elapsed_from_adm
procother	IntubationIndication1	elapsed_from_adm
procother	IntubationIndication2	elapsed_from_adm
procother	IntubationIndication3	elapsed_from_adm
procother	IntubationIndication4	elapsed_from_adm
procother	IntubationIndication5	elapsed_from_adm
procother	IntubationVerification	elapsed_from_adm
procother	IntubationVerification1	elapsed_from_adm
procother	IntubationVerification2	elapsed_from_adm
procother	IntubationVerification3	elapsed_from_adm
procother	IntubationVerification4	elapsed_from_adm
procother	IntubationVerification5	elapsed_from_adm
procother	IntubationVerification6	elapsed_from_adm
procvent	Procedure	elapsed_from_adm
procvent	ITime	elapsed_from_adm
procvent	MeanAirwayPressure	elapsed_from_adm
procvent	VentilatorAction	elapsed_from_adm
procvent	VentilatorMode	elapsed_from_adm
procvent	VentilatorMode1	elapsed_from_adm
procvent	VentilatorMode2	elapsed_from_adm
procvent	nippv_id	elapsed_from_adm

procvent	vent_id	elapsed_from_adm
procvent	vent_before	elapsed_from_adm
procvent	AutoPeep	elapsed_from_adm
procvent	BreathType	elapsed_from_adm
procvent	BreathType1	elapsed_from_adm
procvent	BreathType2	elapsed_from_adm
procvent	Deltap	elapsed_from_adm
procvent	FiO2	elapsed_from_adm
procvent	FlowRate	elapsed_from_adm
procvent	IERation	elapsed_from_adm
procvent	InspPressure	elapsed_from_adm
procvent	MV	elapsed_from_adm
procvent	PEEP	elapsed_from_adm
procvent	PIP	elapsed_from_adm
procvent	PressureSupport	elapsed_from_adm
procvent	Rate	elapsed_from_adm
procvent	elapsed_from_start†	elapsed_from_adm
procvent	RespRateM	elapsed_from_adm
procvent	Sensitivity	elapsed_from_adm
procvent	TV	elapsed_from_adm
procvent	ProcedureVentilatorPTA	elapsed_from_adm
procvent	VentilatorVersion	elapsed_from_adm
times	DateArrived_elapsed_from_start†	DateArrived_elapsed_from_adm
times	DateArriveRec_elapsed_from_start†	DateArriveRec_elapsed_from_adm
times	DateAtPt_elapsed_from_start†	DateAtPt_elapsed_from_adm
times	DateAvailable_elapsed_from_start†	DateAvailable_elapsed_from_adm
times	DateDispatched_elapsed_from_start†	DateDispatched_elapsed_from_adm
times	DateEnroute_elapsed_from_start†	DateEnroute_elapsed_from_adm
times	DateInQtrs_elapsed_from_start†	DateInQtrs_elapsed_from_adm
times	DateLeavePt_elapsed_from_start†	DateLeavePt_elapsed_from_adm
times	DateLeaveRef_elapsed_from_start†	DateLeaveRef_elapsed_from_adm
times	DateNotified_elapsed_from_start†	DateNotified_elapsed_from_adm

times	DateofDescent_elapsed_from_start†	DateofDescent_elapsed_from_adm
times	DateReceived_elapsed_from_start†	DateReceived_elapsed_from_adm
times	DateStandby_elapsed_from_start†	DateStandby_elapsed_from_adm
times	DateTxCare_elapsed_from_start†	DateTxCare_elapsed_from_adm
LSI_table	lsi_group	elapsed_from_adm
LSI_table	lsi_description	elapsed_from_adm
LSI_table	in_hospital	elapsed_from_adm
LSI_table	elapsed_from_start†	elapsed_from_adm

†Elapsed times will be provided from start of case.

8. Appendix B – Example EHR Partition JSONs

All data is drawn from the sample dataset, which contains unprotected, scrambled. As a result, the values of fields may not be consistent.

Example EHR data segment with start-of-case data from the UMB dataset:


```
{
  "demo_scores": [
    {
      "SEXID": "M",
      "RACEID": "C",
      "race_descrip": "White",
      "AGE": 61,
      "INJTYPEID": 1,
      "INJURYTYPEDESCRIP": "Blunt",
      "ICD10ECODE": "V22.4XXA",
      "RTS_S": 7.8408
    }
  ],
  "ems": [
    {
      "LANDVSAIR": "A"
    }
  ]
}
```


Example EHR data segment containing hospital admission and timestamped data from the UMB dataset:


```
{
  "LSI_table": [
    {
      "lsi_group": "Bleeding Control",
      "lsi_description": "Pelvic Binder",
      "in_hospital": 0,
      "elapsed_from_start": NaN
    }
  ],
  "demo_scores": [
    {
      "ADM_TEMP": 36.4,
      "ADM_SYSBP": 88,
      "ADM_DYSBP": 50,
      "ADM_HR": 79.0,
      "ADM_RR": 16,
      "ADM_O2SAT": 94,
      "ADM_GCS_EYE": 4.0,
      "ADM_GCS_VERBAL": 5.0,
      "ADM_GCS_MOTOR": 6.0,
      "GCSTOTAL": 15.0,
```

```
      "BRAINSEV": 0,
      "FACESEV": 0,
      "NECKSEV": 0,
      "THORAXSEV": 0,
      "ABDSEV": 3,
      "SPINESEV": 0,
      "UPPEREXTSEV": 0,
      "LOWEREXTSEV": 0,
      "ISS": 9,
      "TRISS": 0.989,
      "RTS_A": 7.1082,
      "elapsed_from_start": 2400.0
    }
  ],
  "injury": [
    {
      "AISSEVERITY": 2.0,
      "ISSBODYREGION": 4,
      "ICD10CODE": "S36.529A"
    },
    {
      "AISSEVERITY": 3.0,
      "ISSBODYREGION": 4,
      "ICD10CODE": "S36.893A"
    }
  ],
  "non_op_procs": [
    {
      "DESCRIP": "Cardiac Monitoring",
      "elapsed_from_start": 2400.0
    },
    {
      "DESCRIP": "CT Scan - Abdomen",
      "elapsed_from_start": 2400.0
    },
    {
      "DESCRIP": "CT Scan - Cervical Spine",
      "elapsed_from_start": 2400.0
    },
    {
      "DESCRIP": "CT Scan - Thoracic Spine",
      "elapsed_from_start": 2400.0
    }
  ],
  "pta_vitals": [
    {
      "PTA_SBP": 113.0,
      "PTA_DBP": 67.0,
      "PTA_HR": 85.0,
      "PTA_RR": 18.0,
      "PTA_TEMP": NaN,
      "PTA_GCS_TOTAL": 15.0,
      "PTA_GCS_V": 5.0,
      "PTA_GCS_M": 6.0,
      "PTA_GCS_E": 4.0,
      "elapsed_from_start": NaN
```

```
    }
  ]
}
```


Example EHR data segment containing timestamped data after hospital admission from the UMB dataset:


```
{
  "LSI_table": [
    {
      "lsi_group": "Crystalloid Products",
      "lsi_description": "Plasmalyte",
      "in_hospital": 1,
      "elapsed_from_start": 2870.0
    }
  ],
  "labs": [
    {
      "COMPTEXT": "Osmolality (serum)",
      "TESTTEXT": "Osmolality",
      "RSLT": "339",
      "UNITS": "MoM/kg",
      "RSLTDATETIME_elapsed_from_start": 2750.0,
      "OBSDATETIME_elapsed_from_start": 350.0
    },
    {
      "COMPTEXT": "Albumin Level",
      "TESTTEXT": "CMP",
      "RSLT": "5.0",
      "UNITS": "g/dL",
      "RSLTDATETIME_elapsed_from_start": 2930.0,
      "OBSDATETIME_elapsed_from_start": 350.0
    }
  ],
  "medications": [
    {
      "display_name": "plasmalyte-A bolus",
      "mar_action": "Given",
      "sig": 500.0,
      "dose_unit": "mL",
      "route": "Intravenous",
      "pat_loc": "PACU",
      "pharm_class": "Minerals & electrolytes",
      "pharm_subclass": "Electrolyte Mixtures",
      "thera_class": "Nutritional Products",
      "elapsed_from_start": 2875.0
    }
  ]
}
```