Interface Control Document Systems
Competition Phase 3

Revision 2
June 2026

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

Contents

1 Introduction

This document describes the interface to the DARPA Command Post where teams competing in the DARPA Triage Challenge will submit their Casualty Reports during the competition. The intent is to convey the overall concept of operations for interaction with the Command Post during competition and to describe the hardware and software interfaces necessary to successfully interact with the server. This document covers the Command Post interface for the Systems Competition only. For Data competition, please refer to its respective ICD document. Significant revisions from past versions of this document for the same phase are indicated by blue text.

2 Overview and Concept of Operations

- The DARPA Command Post provides the sole interface by which teams will transmit competition data to DARPA for purposes of scoring, and, as such, is a critical element for competition participation. To facilitate ease of integration between each competitor and DARPA, the Command Post interface is designed to use widely accepted Internet standards, helping to ensure that teams will have plenty of prior experience building a compliant system and a wide variety of off-the-shelf libraries to choose from to help them do so.
- The interface is utilized by transmitting data formatted in JavaScript Object Notation (JSON) using the Hypertext Transfer Protocol (HTTP) over a local Ethernet network between the competitor client and the DARPA Command Post server. The various functions of the interface are exposed as separate Uniform Resource Identifiers (URIs) that HTTP requests are executed against. Teams will authenticate their messages using an access token that will be provided during registration and setup.
- Additionally, DARPA intends to observe and share motion video of Base Station displays acquired via HDMI capture devices and cameras located in the Staging Area.
- To facilitate deriving operational insights from these interactions, DARPA requests that individual UxSs (unmanned system) are identified as appropriate in report submissions. This is to be accomplished through the use of a “system” field on relevant endpoints as described below. The "system" field will be specified for each UxS during system registration in advance of each competition. It must be unique within the team and no more than 10 characters (alphanumeric and underscores). Each UxS's "name" must remain consistent across all relevant API endpoints for the entire competition event.

2.1 Scoring

During the competition, a team’s score will be based solely on the Casualty Reports that the team submits to the DARPA Command Post via the scoring interface.

When a team is confident that they have identified and localized a new casualty, they will submit a POST request (a standard HTTP protocol operation) with a structure that describes the type and location of the casualty to the appropriate URI for score reports. After receiving and processing the report, the server will return an HTTP response that contains a confirmation of the submitted report. If there is a problem with the submitted request, for example an authentication token is not provided or the structure is malformed, a verbose error will be returned instead of an updated score. All properly formatted and authenticated casualty

reports are recorded. Fields within the HTTP response body are used to indicate when a valid report has no effect on the team's score (see POST /api/location). For example, casualty reports made after the official run time has ended have no effect on the score; this will be indicated by dedicated fields within the HTTP response body.

A separate URI will provide an interface for teams to request status information about the current run.

2.2 Base Station Display Capture

In addition to cameras in the Staging Area, DARPA intends to observe and capture the Base Station display(s) via HDMI screen capture devices. The captured footage is anticipated to be used as part of the live-stream broadcast. DARPA intends to provide HDMI cabling in the Base Station Area to capture up to two displays supporting the following options:

- 720p (1280 x 720) max 60 hz
- 1080p/i (1920 x 1080) max 60 hz
- UHD 4k (3840 x 2160) max 60 hz

This will require the Base Station to be capable of mirroring and/or extending its display to available HDMI ports. The capture devices will be capable of inline operation allowing teams to make use of secondary displays.

Verification of this functionality will be incorporated into the team/system on-site checkout procedures.

3 Physical and Network Interface

The connections to the DARPA Command Post physical and network interfaces are built on the foundation of Ethernet (IEEE 802.3) communications, so the general concepts should be familiar to all teams. We discuss here the specifics that teams will need to consider to successfully interact with the DARPA Command Post.

- ***Physical Interface:*** All provided network connections will be 1000BASE-T Ethernet (IEEE 802.3ab) over twisted-pair CAT 5e cable or better terminated with a male 8P8C (RJ-45) connector. The connections will provide sufficient bandwidth for all queries and reports permitted in this document between the DARPA Command Post and the respective teams' Base Stations and will use the standard 1,500 octet maximum transmission unit (MTU).

  DARPA will provide one (1) clearly marked Ethernet cable in the area designated for Base Stations that will be the sole connection to the Command Post. Teams must connect this cable to their equipment in such a manner that it establishes a connection with their Base Station. This cable also provides the link to the DARPA provided Wi-Fi described in section 3.1 DARPA Provided Wi-Fi.

- ***Network interface:*** Interactions with the Command Post will be handled by a network endpoint on a DARPA-controlled server. The endpoint will have a static IP version 4 address, and teams will

be responsible for ensuring that their specific network configuration can successfully communicate with the network endpoint. Depending on a team's network configuration, this may require a team to, for example, bind a network address to their designated network interface and/or add additional routes to the IP version 4 routing table on their system in order to allow communications with the DARPA Command Post network endpoint.

Each team will be assigned the following configurations that will be used for all interactions with the DARPA Command Post:

- the static, IP version 4 address and TCP/UDP port information of the DARPA Command Post network endpoint where they will send queries and submissions and
- the IP version 4 configuration that the team must configure on their equipment to communicate with the DARPA Command Post network endpoint. This will include an address, gateway, and subnet within the reserved ranges listed below.

Teams' internal networks must not use IP version 4 subnets and addresses that overlap with the following subnets reserved for use by the DARPA network, except for those assigned to each team by DARPA:

- 10.0.0.0/15 (10.0.0.0 – 10.1.255.255)
- 10.192.0.0/10 (10.192.0.0 – 10.255.255.255)

The assigned network configuration and DARPA Command Post endpoint will remain the same throughout each event. This is a change from the previous phases where network configuration and endpoints changed at each course. Teams will no longer need to change their network configuration at each course. They can instead apply a single configuration and leave it unchanged for the duration of each event.

Network data rate for team interactions may be limited to approximately 100 Mbps, full-duplex between the Base Station area and the DARPA Command Post network endpoint. Submitting data sizes near or over this limit may result in delays or loss.

Network communication between the team Base Station area and the DARPA Command Post will be isolated from all other network traffic to ensure security.

3.1 DARPA Provided Wi-Fi

DARPA will provide a 5 GHz Wi-Fi network to each team that will provide connectivity between wireless clients on the network to wired device over the same Ethernet cable described in section 3. Physical and Network Interface. The Wi-Fi network will be Wi-Fi 6 (802.11ax) compliant, and it is strongly recommended that teams use only Wi-Fi 6 (802.11ax) or higher wireless network adapters to ensure maximum performance.

Each team will have a separate wireless network reserved exclusively for that team's use. Each wireless network will be provided in the respective team garage when that team is not operating on a course. When operating on a course, a team's wireless network will be made available on the course assigned to the team.

Consequently, teams should not need to change their wireless network configuration when moving between team garages and courses.

There are some possible caveats when moving a team's wireless network between the team's garage and a course. Stale network state can result, so teams should be prepared to test for and resolve such issues upon each transition among team garages and courses. This is generally a simple matter of clearing the ARP tables on network devices after moving to the new location. ARP tables can be cleared by issuing an appropriate OS command, disabling and reenabling affected wireless NICs, or rebooting affected network stations.

The proper functioning of network communication can be tested using ICMP ECHO (ping) or similar network communications.

4 Protocol Interface

The DARPA Command Post protocols will be built on the common HTTP/1.1[3], and so, the general concepts should be familiar to all teams. HTTP/1.1 utilizes the stream-oriented Transmission Control Protocol[4] (TCP) that handles the underlying mechanisms of forming and using socket connections. HTTP/1.1 allows requests to be targeted at different Uniform Resource Identifiers[5] (URIs) that loosely describe the functionality that is being requested, and the Command Post uses different URIs to allow multiple functions to be provided at the same network endpoint. These URIs are described in detailed protocol descriptions of Sections 5.

Teams must include an authentication token with each HTTP packet that they send in order to act as an additional mechanism for ensuring that submissions were intended (particularly scoring submissions). The "Authorization" header field must include a "bearer" token[6] that will be provided by during system checkout procedures. For example, if the team's token was "flux230{showroom", then they must include an "Authorization" header field with value "Bearer flux230{showroom". Please note that the DARPA Command Post will deviate from standards and will **not** provide a "WWW-Authenticate" response header with error information in the case of an authentication failure.

In an HTTP response, the HTTP status code will be used to indicate the success or failure of the request, and all HTTP response content will be in JavaScript Object Notation[7] (JSON), a self-describing format for representing structured data. Accordingly, the "Content-Type" header field will always be "application/json". Depending on the interface being used, this content could be either:

(1) a JSON-encoded string (i.e., in quotation marks) when there is a human-readable message accompanying this response (such as an error message)
(2) a JSON-encoded object (i.e., key-value pairs enclosed in braces) that contains detailed return information from an interaction and whose content will depend on the interaction

The applicable HTTP status codes are described below in Section 4.1.

4.1 HTTP Status Codes

The HTTP response status codes that will be returned by the Command Post, along with their meaning, are given below:

Code	*Description*	*Meaning*
200	OK	Request was accepted and/or response will be valid.
201	Created	Request was properly formatted and resource (i.e., casualty report) was created.
400	Bad request	JSON parsing failed.
401	Unauthorized	Authentication token does not match expected token for this run (or was not provided).
404	Not Found	URI is not recognized
413	Request Entity Too Large	Upload file size was too large
422	Unprocessable Entity	JSON request had incorrect information for this interaction type.
429	Too many requests	Too many requests have been submitted in a given amount of time.

```
HTTP/1.1 422 Unprocessable Entity
Content-Length: 23
Content-Type: application/json

“Missing field ‘type’”
```

5 Authentication

In order to provide additional competition integrity and protections against unintended operations, DARPA will require an authorized user token assigned to the appropriate team during interactions with the Command Post. This token will be utilized throughout the competition event. These user tokens will be generated during system checkout at each competition event. A default username and password will be provided to teams upon arrival at each competition event during team orientation. This username and password will be updated during system checkout with the DARPA Command Post server. After updating the default password, an authentication token will be provided and expected to be used for all API interactions during the competition event.

6 Scoring Interaction Protocol

All scoring interactions will take place using the network specified in Section 3 and the HTTP protocol described in Section 4. The specific URIs and JSON content related to scoring interactions are defined here. Each function is described in its subsections below with the HTTP function and corresponding URI as section name.

The majority of interactions should use one of the available “POST” functions (Sections 6.2-8) at infrequent times to submit casualty report information for identified casualties. Teams should avoid overusing API requests in order to avoid delays in response times. Note: since the response to the POST functions includes updated status information, there is no need to immediately follow “POST” submissions with GET /api/status calls.

The following API functions are available are described in detail in following subsections:

- GET **/api/status** – Provides current run information at given time of request
- GET /api/hmt/expired - Request list of locations for casualties who have expired since the beginning of the run.
- POST **/api/location** – Accepts location information for identified casualties on Gate 1.
- POST **/api/triage** – Accepts first pass triage category for identified casualties on Gate 2.
- POST **/api/assessment** – Accepts initial assessment and second pass triage category for identified casualties on Gate 3.
- POST **/api/vitals** – Accepts initial vital assessment for identified casualties on Gate 4.
- POST **/api/hmt/casualty** – Accepts location and first pass triage category for newly identified casualties on the HMT course.
- POST **/api/hmt/assessment** – Accepts additional assessment and second pass triage category for identified casualties on the HMT course.
- POST **/api/hmt/evacuation** - Accepts a list of casualty IDs and evacuation priority
- POST **/api/system_location** – Accepts location updates of individual UxS.
- POST **/api/casualty_image** - Submit an image of a previously identified casualty

6.1 GET /api/status

Request status information about the current run.

The response content will be a JSON object with the following format:

```
{
    "clock": <float>,
    "team": <string>,
    "user": <string>
}
```

Definitions for the return values of the status response are as follow:

- *clock* represents the amount of time that has passed during the given run at the time of the report submission in seconds
- *team* represents the unique identifier of the team requesting the status
- *user* represents the authenticated user based on the authentication token provided

Example request:

```
GET /api/status HTTP/1.1
Authorization: Bearer flux230{showroom
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 77
Content-Type: application/json
```

```
{ “clock”: 721.4, "team": "dtc1", user: “John B” }
```

6.2 POST /api/location

Submit a location report for a newly identified casualty on Gate 1.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "latitude": <float>,
    "longitude": <float>,
    "level": <int>
}
```

Each field has the following definitions:

- *casualty_id* is a team generated unique integer identifier for the casualty whose condition is being reported
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *latitude* is the latitude coordinate of the reported casualty location
- *longitude* is the longitude coordinate of the reported casualty location
- *level* is an optional integer indicating the floor for casualties located within a building

The following fields are required:

- *casualty_id*
- *team*
- *system*
- *latitude*
- *longitude*

Location fields take specific acceptable values, with semantics as follows:

- *casualty_id:* non-negative integer value
- *latitude*: float-precision number with range from -90 and 90 in degrees
- *longitude*: float-precision number with range from -180 and 180 in degrees
- *level:* non-negative integer where 1 refers to the ground floor of a building, 2 refers to the second floor, and so on. For ground level exterior locations, 1 should be used. If this field is not provided, the default value is 1.

Complete definitions of report fields and minimum requirements for scoring can be found in the DARPA Triage Challenge Competition Rules.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*duplicate id*" indicates this report was not scored because it contained a duplicate *casualty_id* from a previously received report during the same run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
- *casualty_id* confirms the casualty identifier in the submitted report

6.3 POST /api/triage

Submit a triage report for a newly identified casualty on Gate 2. A list of casualty locations with corresponding casualty IDs will be provided prior to the run, where the Casualty ID will be used to associate submissions to the ground truth casualty for scoring.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    “category":  <int>
}
```

Each field has the following definitions:

- *casualty_id* is the unique integer identifier for the casualty whose condition is being reported
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *category* indicates the first-pass triage category for the casualty.

The following fields are required:

- *casualty_id*
- *team*
- *system*
- *category*

Some fields take specific acceptable values, with semantics as follows:

- *casualty_id:* non-negative integer value with a range between 0 and 99
- *category:*
    - *0 (Urgent / Hemorrhage)* Casualty with suspected life-threatening bleeding needing intervention in the next 5 minutes
    - *1 (Urgent / Airway)* Casualty with suspected life-threatening airway complications needing intervention in the next 5 minutes
    - *2 (Non-Urgent / Ambulatory)* Casualty expected to survive at least 5 minutes without intervention and able to walk or stand
    - *3 (Non-Urgent / Stationary)* Casualty expected to survive at least 5 minutes without intervention but unable to walk or stand
    - *4 (Deceased)* Casualty with no signs of life

Complete definitions of report fields and minimum requirements for scoring can be found in the DARPA Triage Challenge Competition Rules.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
```

```
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*duplicate id*" indicates this report was not scored because it contained a duplicate *casualty_id* from a previously received report during the same run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
- *casualty_id* confirms the casualty identifier in the submitted report

6.4 POST /api/assessment

Submit a casualty report for a newly identified casualty or provide additional information on a previously reported casualty on Gate 3. Each assessment field will be reported with individual submissions, linked by Casualty ID. A list of casualty locations with corresponding casualty IDs will be provided prior to the run, where the Casualty ID will be used to associate submissions to the ground truth casualty for scoring.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "type": <string>,
```

```
    "value": <int>,
    "time_ago": <int>
}
```

Each field has the following definitions:

- *casualty_id* is unique integer identifier for the casualty whose condition is being reported
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *type* is a string indicating the casualty report field being submitted:
  - *sevacuation_priority* indicates the evacuation prioritization of the casualty.
  - *trauma_head* indicates presence of injury on the head for the reported casualty.
  - *trauma_torso_back* indicates presence of injury on the back of the torso for the reported casualty.
  - *trauma_torso_front* indicates presence of injury on the front of the torso for the reported casualty.
  - *trauma_leg_right* indicates the presence and type of injury on the right leg for the reported casualty.
  - *trauma_leg_left* indicates the presence and type of injury on the left leg for the reported casualty.
  - *trauma_arm_right* indicates the presence and type of injury on the right arm for the reported casualty.
  - *trauma_arm_left* indicates the presence and type of injury on the left arm for the reported casualty.
  - *alertness_ocular* indicates the presence and type of alertness based on eye movement at time *time_ago* relative to the time the report was submitted.
  - *alertness_verbal* indicates the presence and type of alertness based on speech and vocalizations at time *time_ago* relative to the time the report was submitted.
  - *alertness_motor* indicates the presence and type of alertness based on gross motor movement at time *time_ago* relative to the time the report was submitted.
- *value* is an integer indicating the assessed information for the provided *type*
- *time_ago* is an integer indicating elapsed time in seconds relative to the time the report was submitted

The following fields are required:

- *casualty_id*
- *team*
- *system*
- *type*
- *value*

Some fields take specific acceptable values, with semantics as follows:

- *casualty_id:* non-negative integer value with a range between 0 and 99
- *type:* string that matches one of the following:

- *evacuation_priority*
- *trauma_head*
- *trauma_torso_back*
- *trauma_torso_front*
- *trauma_leg_right*
- *trauma_leg_left*
- *trauma_arm_right*
- *trauma_arm_left*
- *alertness_ocular*
- *alertness_verbal*
- *alertness_motor*

- *value*: integer with interpretation that depends on the *type* provided:
  - *evacuation_priority*
    - *1 (Urgent)* Casualty with suspected life-threatening injury needing rapid evacuation: Unresponsive AND NOT Deceased AND Respiratory Rate >= 35 BrM
    - *2 (Urgent)* Casualty with suspected life-threatening injury needing surgery within 24 hours: Abnormal alertness AND Respiratory Rate >= 35 BrM
    - *3 (Priority)* Medium priority casualty with serious injuries or illness who will require surgery or advanced medical treatment at a delayed time (may include limb- or eyesight-threatening injuries), or previously urgent casualty who received treatment and remains stable with normal vitals and mental status: Normal alertness AND NOT Ambulatory AND Respiratory Rate < 35 BrM
    - *4 (Routine)* Low priority casualty with minimal injuries or illness, or no signs of life: Ambulatory OR Deceased
    -
  - All trauma fields: *trauma_head, trauma_head, trauma_torso_back, trauma_torso_front, trauma_leg_right, trauma_leg_left, trauma_arm_right, trauma_arm_left*
    - *0 (amputation)* indicates traumatic removal of body part with visible blood at or around wound site
    - *1 (open)* indicates break in the skin that is visibly open to the environment
    - *2 (closed)* indicates evidence of blunt or indirect injury but no visible opening in the skin
    - *3 (burn)* indicates signs of thermal or chemical injury
  - *alertness_ocular*
    - *0 (open)* indicates eyelids are open and/or blinking spontaneously
    - *1 (closed)* indicates eyelids are closed
    - *2 (not testable)* indicates eyelid assessment was not possible
  - *alertness_verbal*
    - *0 (normal)* indicates responsiveness to speech prompts with coherent and relevant speech
    - *1 (abnormal)* indicates confused or pain-/distress-related speech and/or vocalizations
    - *2 (absent)* indicates absence of vocalization

        - *3 (not testable)* indicates speech and vocalization assessment was not possible
    - *alertness_movement*
        - *0 (normal)* indicates walking/standing/sitting unsupported with coordinated movement
        - *1 (abnormal)* indicates lying or sitting supported with minimal limb movement
        - *2 (absent)* indicates absence of limb movement
        - *3 (not testable)* indicates movement assessment was not possible
- *time_ago:* non-negative integer indicating seconds elapsed since the assessment was made, used to determine ground truth for scoring. If not provided, the default value is 0.

Complete definitions of report fields and minimum requirements for scoring can be found in the DARPA Triage Challenge Competition Rules.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop

    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*duplicate id*" indicates this report was not scored because it contained a duplicate *casualty_id* from a previously received report during the same run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
- *casualty_id* confirms the casualty identifier in the submitted report

6.5 POST /api/vitals

Submit a casualty report for a newly identified casualty or provide additional information on a previously reported casualty on Gate 4. Each vitals field will be reported with individual submissions, linked by Casualty ID. A list of casualty locations with corresponding casualty IDs will be provided prior to the run, where the Casualty ID will be used to associate submissions to the ground truth casualty for scoring.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    “type":  <string>,
    "value": <int>,
    "time_ago": <int>
}
```

Each field has the following definitions:

- *casualty_id* is a team generated unique integer identifier for the casualty whose condition is being reported
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *type* is a string indicating the vitals estimate being submitted:
    - *hr* indicates the heart rate in beats per minute for the reported casualty at time *time_ago* relative to the time the report was submitted.
    - *rr* indicates the respiration rate in breaths per minute for the reported casualty at time *time_ago* relative to the time the report was submitted.
- *value* is an integer for the vitals estimate of provided *type*
- *time_ago* is an integer indicating elapsed time in seconds relative to the time the report was submitted

The following fields are required:

- *casualty_id*
- *team*
- *system*

- *type*
- *value*

Some fields take specific acceptable values, with semantics as follows:

- *casualty_id:* non-negative integer value with a range between 0 and 99
- *type:* string that matches one of the following:
  - *hr*
  - *rr*
- *value*: integer with interpretation that depends on the *type* provided:
  - *rr:* non-negative integer indicating respiratory rate in breaths per minute
  - *hr:* non-negative integer indicating heart rate in beats per minute
- *time_ago:* non-negative integer indicating seconds elapsed since the estimate was made, used to determine ground truth for scoring. If not provided, the default value is 0.

Complete definitions of report fields and minimum requirements for scoring can be found in the DARPA Triage Challenge Competition Rules.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules

- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*duplicate id*" indicates this report was not scored because it contained a duplicate *casualty_id* from a previously received report during the same run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
- *casualty_id* confirms the casualty identifier in the submitted report

6.6 POST /api/hmt/casualty

Submit a casualty report for a newly identified casualty on the Human-Machine Teaming (HMT) course.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "latitude": <float>,
    "longitude": <float>,
    "level": <int>,
    "category":  <int>,
    "time_ago": <int>
}
```

Each field has the following definitions:

- *casualty_id* is a team generated unique integer identifier for the casualty whose condition is being reported
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *latitude* is the latitude coordinate of the reported casualty location
- *longitude* is the longitude coordinate of the reported casualty location
- *level* is an optional integer indicating the floor for casualties located within a building
- *category* indicates the first-pass triage category for the casualty.
- *time_ago* is an integer indicating elapsed time in seconds relative to the time the report was submitted

The following fields are required:

- *casualty_id*

- *team*
- *system*
- *longitude*
- *latitude*
- *category*

Some fields take specific acceptable values, with semantics as follows:

- *casualty_id:* non-negative integer value with a range between 0 and 99
- *latitude*: float-precision number with range from -90 and 90 in degrees
- *longitude*: float-precision number with range from -180 and 180 in degrees
- *level:* non-negative integer where 1 refers to the ground floor of a building, 2 refers to the second floor, and so on. For exterior ground level locations, 1 should be used. If this field is not provided, the default value is 1.
- *category:*
    - *0 (Urgent / Hemorrhage)* Casualty with suspected life-threatening bleeding needing intervention in the next 5 minutes
    - *1 (Urgent / Airway)* Casualty with suspected life-threatening airway complications needing intervention in the next 5 minutes
    - *2 (Non-Urgent / Ambulatory)* Casualty expected to survive at least 5 minutes without intervention and able to walk or stand
    - *3 (Non-Urgent / Stationary)* Casualty expected to survive at least 5 minutes without intervention but unable to walk or stand
    - *4 (Deceased)* Casualty with no signs of life
- *time_ago:* non-negative integer indicating seconds elapsed since the casualty was found and assessed for first-pass triage, used to determine ground truth for scoring. If not provided, the default value is 0.
    -

Complete definitions of report fields and minimum requirements for scoring can be found in the DARPA Triage Challenge Competition Rules.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*duplicate id*" indicates this report was not scored because it contained a duplicate *casualty_id* from a previously received report during the same run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
- *casualty_id* confirms the casualty identifier in the submitted report

6.7 POST /api/hmt/assessment

Submit additional information on a previously reported casualty on the Human-Machine Teaming (HMT) course. Casualty ID is used to link between reports on the same casualty. The last received report submission for a given type and Casualty ID will be scored.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "type": <string>,
    "value": <int>,
    "time_ago": <int>
}
```

Each field has the following definitions:

- *casualty_id* is unique integer identifier for the casualty whose condition is being reported
- *team* is a unique string identifier for the team submitting the report

- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *type* is a string indicating the casualty report field being submitted:
  - *evacuation_priority* indicates the evacuation prioritization of the casualty.
  - *trauma_head* indicates presence of injury on the head for the reported casualty.
  - *trauma_torso_back* indicates presence of injury on the back of the torso for the reported casualty.
  - *trauma_torso_front* indicates presence of injury on the front of the torso for the reported casualty.
  - *trauma_leg_right* indicates the presence and type of injury on the right leg for the reported casualty.
  - *trauma_leg_left* indicates the presence and type of injury on the left leg for the reported casualty.
  - *trauma_arm_right* indicates the presence and type of injury on the right arm for the reported casualty.
  - *trauma_arm_left* indicates the presence and type of injury on the left arm for the reported casualty.
  - *alertness_ocular* indicates the presence and type of alertness based on eye movement at time *time_ago* relative to the time the report was submitted.
  - *alertness_verbal* indicates the presence and type of alertness based on speech and vocalizations at time *time_ago* relative to the time the report was submitted.
  - *alertness_motor* indicates the presence and type of alertness based on gross motor movement at time *time_ago* relative to the time the report was submitted.
  - rr indicates the respiration rate
  - hr indicates the heart rate
- *value* is an integer indicating the assessed information for the provided *type*
- *time_ago* is an integer indicating elapsed time in seconds relative to the time the report was submitted

The following fields are required:

- *casualty_id*
- *team*
- *system*
- *type*
- *value*

Some fields take specific acceptable values, with semantics as follows:

- *casualty_id:* non-negative integer value with a range between 0 and 99
- *type:* string that matches one of the following:
  - *evacuation_priority*
  - *trauma_head*
  - *trauma_torso_back*
  - *trauma_torso_front*
  - *trauma_leg_right*

- *trauma_leg_left*
- *trauma_arm_right*
- *trauma_arm_left*
- *alertness_ocular*
- *alertness_verbal*
- *alertness_motor*
- *rr*
- *hr*

- *value*: integer with interpretation that depends on the *type* provided:
  - *evacuation_priority*
    - *1 (Urgent)* Casualty with suspected life-threatening injury needing rapid evacuation: Unresponsive AND NOT Deceased AND Respiratory Rate >= 35 BrM
    - *2 (Urgent)* Casualty with suspected life-threatening injury needing surgery within 24 hours: Abnormal alertness AND Respiratory Rate >= 35 BrM
    - *3 (Priority)* Medium priority casualty with serious injuries or illness who will require surgery or advanced medical treatment at a delayed time (may include limb- or eyesight-threatening injuries), or previously urgent casualty who received treatment and remains stable with normal vitals and mental status: Normal alertness AND NOT Ambulatory* AND Respiratory Rate < 35 BrM
    - *4 (Routine)* Low priority casualty with minimal injuries or illness, or no signs of life: Ambulatory OR Deceased
  - All trauma fields: *trauma_head, trauma_head, trauma_torso_back, trauma_torso_front, trauma_leg_right, trauma_leg_left, trauma_arm_right, trauma_arm_left*
    - *0 (amputation)* indicates traumatic removal of body part with visible blood at or around wound site
    - *1 (open)* indicates break in the skin that is visibly open to the environment
    - *2 (closed)* indicates evidence of blunt or indirect injury but no visible opening in the skin
    - *3 (burn)* indicates signs of thermal or chemical injury
  - *alertness_ocular*
    - *0 (open)* indicates eyelids are open and/or blinking spontaneously
    - *1 (closed)* indicates eyelids are closed
    - *2 (not testable)* indicates eyelid assessment was not possible
  - *alertness_verbal*
    - *0 (normal)* indicates responsiveness to speech prompts with coherent and relevant speech
    - *1 (abnormal)* indicates confused or pain-/distress-related speech and/or vocalizations
    - *2 (absent)* indicates absence of vocalization
    - *3 (not testable)* indicates speech and vocalization assessment was not possible
  - *alertness_movement*
    - *0 (normal)* indicates walking/standing/sitting unsupported with coordinated movement

        - *1 (abnormal)* indicates lying or sitting supported with minimal limb movement
        - *2 (absent)* indicates absence of limb movement
        - *3 (not testable)* indicates movement assessment was not possible
    - hr and rr
        - Integer value representing the heart rate or respiration rate
- *time_ago:* non-negative integer indicating seconds elapsed since the assessment was made, used to determine ground truth for scoring. If not provided, the default value is 0.

Complete definitions of report fields and minimum requirements for scoring can be found in the DARPA Triage Challenge Competition Rules.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start

- "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run

- *casualty_id* confirms the casualty identifier in the submitted report

6.8 POST /api/hmt/evacuation

Submit a list of previously reported Priority 1 casualties on the Human-Machine Teaming (HMT) course who have been selected for immediate evacuation. Casualty ID is used to link between reports on the same casualty. The last received report submission will be scored.

The request content must be a JSON-encoded object with the following format:

```
{
    "casualty_ids": [<int>]
}
```

Each field has the following definitions:

- *casualty_ids* is an array of unique integer identifier for the casualty whose condition is being reported

The following fields are required:

- *casualty_ids*

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response

- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
    - "casualty limit exceeded" indicates that this report sent too many casualty ids in the casualty_ids array
    - "invalid time window" indicates that this report was not scored because it was submitted before the last 5 minutes of run

6.9 GET /api/hmt/expired

Request list of locations for casualties who have expired since the beginning of the run.

The response content will be a JSON object with the following format:

```
{
    "casualty_locations": [
        <location>,
        <location>,
        …
    ]
}
```

where each **location** is a sub-array with the following format:

```
[<latitude: float>, < longitude: float>, <level: int>]
```

Definitions for the values of the status response are as follow:

- *casualty_locations* contains a list of locations for casualties whose simulated condition has degraded since the beginning of the run and are now deceased. Casualties who were deceased at the beginning of the run will not be included in this list.
- *latitude* is a float-precision number with range from -180 and 180 in degrees
- *longitude* is a float-precision number with range from -180 and 180 in degrees
- *level* is a non-negative integer where 1 refers to the ground floor of a building, 2 refers to the second floor, and so on. For exterior ground level locations, 1 will be used.

6.10 POST /api/system_location

Provide platform location information at regular intervals (1-4 Hz) during runs on all courses: Gates 1-4 and HMT.

The request content must be a JSON-encoded object with the following format:

```
{
    "team": <string>,
    "system": <string>,
    "type": <int>,
    "latitude": <float>,
    "longitude": <float>,
    "altitude": <float>
}
```

Each field has the following definitions:

- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the identification of the casualty
- *type* is the type of autonomous platform (UAV or UGV)
- *latitude* is the latitude coordinate of the autonomous platform
- *longitude* is the longitude coordinate of the autonomous platform
- *altitude* is an optional altitude coordinate for airborne systems

The following fields are required

- *team*
- *system*
- *type*
- *latitude*
- *longitude*

Some fields take specific acceptable values, with semantics for each casualty report field as follows:

- *type:*
    - *0 (UGV)* indicates ground-based autonomous platform
    - *1 (UAV)* indicates airborne autonomous platform
- *latitude*: float-precision number with range from -90 and 90 in degrees
- *longitude*: float-precision number with range from -180 and 180 in degrees
- *altitude*: non-negative float-precision number in meters

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "system": <string>,
```

```
    "type": <float>,
}
```

The semantics of the response fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *type* confirms the reported system type

6.11 POST /api/casualty_image

Submit an image of a previously identified casualty on all courses: Gates 1-4 and HMT.

The request content must be multipart/form-data with the data object in following format:

```
{
    "casualty_id": <int>,
    "team": <string>,
    "system": <string>,
    "time_ago": <int>
}
```

And with an attached file object with content type of image/png or image/jpeg.

The semantics of the fields are as follow:

- *casualty_id* is a previously reported integer identifier used to associate this update report with the initial report on the same casualty
- *team* is a unique string identifier for the team submitting the report
- *system* is a unique string identifier for the unmanned system responsible for the image
- *time_ago* is a non-negative integer with seconds elapsed since image capture.

The image should be a PNG or JPEG image with file size not exceeding 5MB. In addition, images will only be accepted once per second (rate of 1Hz).

If the image size exceeds 5MB, a HTTP error code of 413 will be returned.
If the request endpoint has submitted at too high of a rate, a HTTP error code of 429 will be returned.

The response content will be a JSON object with the following format:

```
{
    "run": <string>,
    "team": <string>,
    "user": <string>,
    "system": <string>,
    "clock": <float>,
    "report_id": <string>,
    "report_timestamp": <string>,
    "report_status": <string>,
    "casualty_id": <integer>
```

```
}
```

The semantics of the fields are as follows:

- *run* represents an identifier for the current run for this report
- *team* represents the name of the team that submitted the report
- *user* represents the authenticated user that submitted the report
- *system* represents the unmanned system responsible for providing the report generation
- *clock* represents the amount of time that has passed during the given run at the time of the report submission
- *report_id* is a unique identifier for the recorded report response
- *report_timestamp* represents the absolute time that the report was received, following the ISO 8601 combined date and time format. This timestamp is used to award bonus points according to the official scoring guidelines in the DARPA Triage Challenge Competition Rules
- *report_status* indicates the status of the casualty report.
    - "*accepted*" indicates the report was accepted to be evaluated and one allotted report was used
    - "*admin stop*" indicates this report was not scored because it was submitted during an administrative stop
    - "*run not started*" indicates this report was not scored because it was submitted prior to run start
    - "*no initial report*" indicates this report was not scored because an initial report with the same *casualty_id* has not yet been received during the run
    - "*time limit exceeded*" indicates that this report was not scored because it was submitted after the end of the run
    - "*size limit exceeded*" indicates that the image size exceeds the limit
    - "*rate limit exceeded*" indicates that the rate of image submissions exceeds the limit
- *casualty_id* confirms the casualty identifier in the submitted report

7 Appendix A – Example image submission

Instances of localhost must be changed to actual server IP. Auth_token is valid only for the test submission server. A different value will be provided during test events.

Example Command Line Curl Call

```
curl -X 'POST' \
  'http://localhost/api/casualty_image' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIi-
OiI4M2Q3OGM4ZS04MzhhLTQ0NzctOWM3Yi02N2VmMTZlNWY3MTYiLCJpI-
jowfQ.i4KuwEtc5_6oIYz5TDWcdzl5bMkvCpLZTSZG2Avy84w' \
  -H 'Content-Type: multipart/form-data' \
  -F 'file=@test.jpeg;type=image/jpeg' \
  -F 'casualty_id=1' \
  -F 'team=TEST_TEAM' \
  -F 'system=UGV_1' \
  -F 'time_ago=0'
```

Example Python Script

```
import requests

server = 'localhost'
url = f'http://{server}/api/casualty_image'
auth_token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIi-
OiI4M2Q3OGM4ZS04MzhhLTQ0NzctOWM3Yi02N2VmMTZlNWY3MTYiLCJpI-
jowfQ.i4KuwEtc5_6oIYz5TDWcdzl5bMkvCpLZTSZG2Avy84w'
headers = {
    'Authorization': f'Bearer {auth_token}'
}

file_path = 'test.jpeg'

data = {
  'casualty_id': 1,
  'team': 'TEST_TEAM',
  'system': 'UGV_1',
  'time_ago': 0
}

with open(file_path, 'rb') as f:
  files = {
      'file': f
  }
```

```
  response = requests.post(url, files=files, headers=headers, data=data)

if response.status_code == 200:
  print("Successful")
else:
  print(f"Error Type: {response.status_code} Details: {response.text}")
```

8 Appendix B – Standards/References

The following are the applicable standards and references for this document:

- Ethernet: IEEE 802.3; twisted-pair cable, e.g., IEEE 803.3ab
- HyperText Transfer Protocol 1.1: IETF RFC 7230 (https://tools.ietf.org/html/rfc7230)
- TCP/IP: IETF RFC 1122 (https://tools.ietf.org/html/rfc1122)
- URI: IETF RFC 3986 (https://tools.ietf.org/html/rfc3986), specifically the path information
- Bearer Token Authentication: IETF RFC 6750 (https://tools.ietf.org/html/rfc6750)
- JSON: IETF RFC 8259 (https://tools.ietf.org/html/rfc8259)