

Interface for Coordinating Emergencies and  
Disasters  
Requirements Document

October 1, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Definitions, acronyms, and abbreviations . . . . .	3
1.3	Overview . . . . .	3
<b>2</b>	<b>Requirements</b>	<b>4</b>
2.1	Product 0 - Wrapper Interface . . . . .	4
2.1.1	Product shall allow user registration with an email and password combination . . . . .	4
2.1.2	Product shall require users to validate their email addresses before logging in . . . . .	4
2.1.3	Product shall allow users who have received unsolicited emails to remove and/or blacklist their email from the system via an email link. . . . .	5
2.1.4	Product shall allow session sharing between child products . . . . .	5
2.1.5	Product shall allow users to delete their accounts . . . . .	5
2.2	Product 1 - Crowd Sourcing Interface . . . . .	6
2.2.1	Product shall allow users to search for disaster by location . . . . .	6
2.2.2	Product shall allow user to select disaster for future actions . . . . .	6
2.2.3	Product shall allow user to look up user submitted reports . . . . .	6
2.2.4	Product shall allow users to filter based on report reputation / number of reports . . . . .	7
2.2.5	Product shall allow users to submit a report . . . . .	7
2.2.6	Product shall prevent report submissions for events outside of the user's reported location . . . . .	7
2.2.7	Product shall auto-fill report location based on user's current location . . . . .	8
2.2.8	Product shall plot reports on a map . . . . .	8
2.2.9	Product shall provide a tagging interface to narrow down report category . . . . .	8
2.2.10	Product shall allow submission of video / photo evidence in report . . . . .	9
2.2.11	Product shall allow submission of event time ranging from the current time to 24 hours prior . . . . .	9
2.2.12	Product shall disallow a report occurring before the disaster's start date . . . . .	9
2.2.13	Product shall allow organizations to approve / disapprove user submitted reports . . . . .	10
2.2.14	Product shall require organizations to record their related tags . . . . .	10
2.2.15	Product shall enforce that organizations only adjust user reports that share a related tag. . . . .	10
2.2.16	Product shall adjust a user's reputation based on approved and disapproved reports . . . . .	10

2.2.17	Product shall increment a user's reputation by 25 points when approved by an organization . . . . .	11
2.2.18	Product shall decrement a user's reputation by 10 points when disapproved by an organization . . . . .	11
2.2.19	Product shall enforce a limit on the number of reports by a single user in a 100 meter radius by floor(reputation / 100) . . . . .	11
2.2.20	Product shall display an area average for similar reports in an area . . . . .	11
2.3	Product 2 - Communication Protocol . . . . .	12
2.3.1	Product shall provide a public/private key decentralized information distribution protocol . . . . .	12
2.3.2	Product shall build a chain of trust from a built-in starting piece of data . . . . .	12
2.3.3	Product shall allow the creation of specialized (miniaturized) nodes . . . . .	12
2.3.4	Product shall implement authorization / revoke transactions for organization keys . . . . .	12
2.3.5	Product shall require organizations to sign participation keys in each disaster epoch . . . . .	13
2.3.6	Product shall require signing machines be used to register organization's authorization keys . . . . .	13
2.3.7	Product shall implement a no-fee transaction system . . . . .	13
2.3.8	Product shall allow the recording of text data . . . . .	13
2.3.9	Product shall include interface/API support for non-conventional transmission protocols . . . . .	14
2.3.10	Product shall provide structure for administrative transactions . . . . .	14
2.3.11	Product shall implement a method to close or seal expired disasters . . . . .	14
2.3.12	Product shall implement a post-op report interface for closed disasters . . . . .	14
2.3.13	Product shall provide an encryption interface for private event-related data . . . . .	15
2.3.14	Product shall provide an interface to look up other organization's public keys . . . . .	15
2.3.15	Product shall provide programmable interfaces to allow program interaction with custom events . . . . .	15
2.4	Product 3 - Logistics Interface . . . . .	15
2.4.1	Product shall provide an interface for filtering operations pertaining to an organization . . . . .	15
2.4.2	Product shall provide an interface for creating tasks for organization members . . . . .	15
2.4.3	Product shall provide a method for setting the priority and categories of tasks . . . . .	16

2.4.4	Product shall provide a method for adjusting the visibility of certain tasks . . . . .	16
2.4.5	Product shall provide a permissions system to adjust who can view/edit/create tasks . . . . .	16
2.4.6	Product shall provide an interface to create custom data objects . . . . .	16

# 1 Introduction

## 1.1 Purpose

As the occurrence of natural disasters continues to cause catastrophic damage to metropolitan and rural areas alike, the need for new innovative disaster relief solutions is readily apparent. After researching currently available solutions for disaster relief management, we have found that there is not a unified platform for the public to use during a disaster situation to centralize related information. Likewise, the public does not have access to a platform or service that allows users to check on the status of local resources, like open shelters or available gas. Possessing such a platform would not only enable more effective relief efforts from aid organizations like FEMA, but also would allow for the implementation of other improvements that have been critically identified. The gap between the systems that currently exist and what new needs have been identified can be bridged by building an interface that would coordinate civilians, first responders, and relief organizations throughout the entire disaster relief process.

The Interface for Coordinating Emergencies and Disasters (I.C.E.D.) aims to be an innovative solution that addresses three major areas of concern that have not been previously answered by another platform or service.

## 1.2 Definitions, acronyms, and abbreviations

- Civilian: Any individual that is not associated with disaster relief. Generally evacuees or inhabitants of an affected area.
- Affiliate: Any organization or individual who does not necessarily belong to a disaster relief organization, but has volunteered or has been contracted to assist in the relief process. Examples include hospitals, hotels, etc.
- Disaster Relief Organization: Any organization that specializes in providing aid to disaster stricken areas. Examples include FEMA, local Emergency Operation Centers, and organizations like the Red Cross.

## 1.3 Overview

Due to the nature of the requirements, three primary objectives were outlined by the clients, which compose three sub projects: A crowd source interface, a decentralized, multi-form communication protocol, and a logistics interface. A set of global requirements were derived that affect all platforms, such as login, logout, and account specific operations which will be identified as Product 0. All other requirements will pertain to a single sub-product. Note that this document will not cover system integration in detail. Even though these products are separated on the requirement level, they will still be integrated together to compose a single project.

## 2 Requirements

### 2.1 Product 0 - Wrapper Interface

#### 2.1.1 Product shall allow user registration with an email and password combination

**Rationale:** By allowing a user to register with an email and password combination, we allow users to track information pertaining to them, in a format that would be most familiar to users.

**Fit Criterion:** User is able to create a personal account by supplying, at minimum, an email and password.

**Sample I/O:**

- Correct email/password: Server responds with a success message and an email verification message.
- Incorrect email / password: Server responds with an input validation message and an array of invalid elements.

#### 2.1.2 Product shall require users to validate their email addresses before logging in

**Rationale:** By requiring a user to validate their email address, we can ensure that the amount of accounts generated will be limited to a user's available number of email addresses, of which is on average around 1. It also ensures that their account is set up correctly in the event that a user forgets their password.

**Fit Criterion:** User is sent an email validation link that when clicked will allow the user to log in.

**Sample I/O:**

- Following Registration: User is emailed a validation link that is unique to their account. A message is displayed to the user asking them to check their email before logging in.
- Login without validation: Server responds with login failed, with an array containing the key emailvalidation. A message is displayed on the front-end stating that they must validate their account before signing in. A link offering to re-send the email is provided, with a delay of 30 minutes before re-sending.
- Successful validation: Server will respond with a success message, allowing the user to login in.
- Expired link: Server will respond with a link expired message and an array with the key linkexpired. The user is presented with a link that allows them to re-send the validation email.

### **2.1.3 Product shall allow users who have received unsolicited emails to remove and/or blacklist their email from the system via an email link.**

**Rationale:** In order to prevent potential email abuse, a user should be allowed to remove their emails from the system without knowing their account details. If a malicious user registers using another individual's email, nothing would be in place to prevent validation email spam. Blacklisting will require a support email to unlock the email address if a user wishes to register at a later date.

**Fit Criterion:** Validation emails, when sent, contain a link at the bottom which allows users to either remove or blacklist their emails from the system.

**Sample I/O:**

- User requests email unlink: Server removes the account registered for the user and displays a success message.
- Link is not valid: Server responds with a failure message and the user is notified to contact support if the message is in error.
- User requests email blacklist: Server removes the account registered for the user and appends the email to a blacklist table. User is notified with a success message and notified that in order to remove the blacklist entry, they must contact support. A support email or website link must be provided.

### **2.1.4 Product shall allow session sharing between child products**

**Rationale:** Since all products operate under a parent interface, there should be no reason to require users to create multiple accounts for every product. Instead, one account should be created to access all products.

**Fit Criterion:** All products link to the same login/logout form, and work with the same account structure.

**Sample I/O:** N/A

### **2.1.5 Product shall allow users to delete their accounts**

**Rationale:** A control process should be set up to facilitate the deletion of user accounts. Personally identifiable information must be cleared, but user ids will be locked and associated posts will be kept in the system to ensure that other records stay intact.

**Fit Criterion:** A user account is deactivated, personally identifiable records are deleted, and the user record is deactivated.

**Sample I/O:**

- User requests deletion: Server deactivates account, clears out email and personally identifiable details and responds with a success message, notifying the user of a success account deletion.

## 2.2 Product 1 - Crowd Sourcing Interface

### 2.2.1 Product shall allow users to search for disaster by location

**Rationale:** Users who are looking up crowd sourcing information or posting will want to search for recorded disasters by location. By facilitating a disaster lookup based on location, we can expedite the amount of time it takes for a user to retrieve or add information and keep relevant information close by.

**Fit Criterion:** A user searches for a location and a list of associated disasters is returned.

**Sample I/O:**

- User enters zip code: Server looks up location using external API or database, converts it to lat/long, and returns disasters within a certain radius of the coordinates.
- User allows application to retrieve coordinates: Server returns disasters within a certain radius of the coordinates.

### 2.2.2 Product shall allow user to select disaster for future actions

**Rationale:** Users who are preparing for a disaster or are currently affected will be interested exclusively in that disaster. By allowing a user to select a disaster, we can simplify the amount of information a user has to supply when doing information lookups.

**Fit Criterion:** A user is able to select a disaster, and disaster related lookups will utilize the saved disaster if none is specified in the search parameters.

**Sample I/O:**

- User selects current disaster: Server records the disaster ID and returns a success message. Related queries should use the recorded disaster ID if one isn't specified in a server lookup.
- User has expired disaster recorded: Server resets the user's saved disaster and returns a message notifying the user that the disaster is no longer active.

### 2.2.3 Product shall allow user to look up user submitted reports

**Rationale:** Users in an affected disaster area may want to look up user submitted reports to assist in planning an evacuation or to assist in returning to an affected area. Crowd sourced reports not only assist organizations in their relief efforts, but also the affected civilians for the aforementioned reasons.

**Fit Criterion:** An interface facilitating user lookups of crowd sourced information for a disaster is supplied.

**Sample I/O:** N/A



#### 2.2.4 Product shall allow users to filter based on report reputation / number of reports

**Rationale:** Most users and organizations may initially be hesitant to trust crowd sourced information as there is the possibility of including false information in critical plans. By allowing users to specify the minimum reputation we ensure that critical plans have a safeguard against less reliable information.

**Fit Criterion:** A method for allowing a user to specify a numerical minimum reputation is provided, and returned results honor the limit.

**Sample I/O:**

- User specifies a reputation of 0 or lower: Reputation limit is ignored.
- User specifies a reputation limit by search: Server supplies the reputation limit as a search parameter and returns results that have a reputation greater than or equal to the limit.
- User specifies a reputation limit in their preferences: Server records the reputation limit in the database and uses the limit when no associated parameter is supplied.

#### 2.2.5 Product shall allow users to submit a report

**Rationale:** Allowing users to report hazards and obstacles in an affected area is the core requirement behind the crowd sourcing platform. Reported hazards and obstacles are viewable by anyone in an affected area, including civilians who are planning on evacuating or returning, and organizations looking to get an in-depth view of the situation on the ground before first responders arrive.

**Fit Criterion:** Users are able to submit a report with information on a hazard or obstacle in an affected area.

**Sample I/O:** N/A

#### 2.2.6 Product shall prevent report submissions for events outside of the user's reported location

**Rationale:** Allowing users to submit reports outside of their reported location or selected disaster makes it easier for malicious users to generate fake information. It is assumed that if you are going to report information as an individual in the disaster, then you must be present in the affected area.

**Fit Criterion:** A user should be prohibited from generating reports in areas where they're not currently present.

**Sample I/O:**

- User submits report inside affected area: Server records the event and returns a success message.
- User submits report outside affected area: Server returns a failure message, notifying the user that they must be in the affected area to submit a report.

### 2.2.7 Product shall auto-fill report location based on user's current location

**Rationale:** To simplify the amount of information gathered, a report is assumed to be created near where the hazard is located. If a user's device supports location detection, then the product should auto-fill the report location with the user's location provider. This value is not meant to be permanent and is only used to speed up the report generation time. The value can be changed by the user afterwards if desired.

**Fit Criterion:** If a location provider is present, the report location is auto-filled when a user attempts to generate a report.

**Sample I/O:**

- User creates report with location provider: Report's location field is auto-filled with device's reported location
- User creates report without location provider: Report's location is left blank and will require user interaction to fill in.

### 2.2.8 Product shall plot reports on a map

**Rationale:** To assist in the decision making process for other users, reports should be plotted on a map based on their location. This makes it easier for users and organizations who cannot manually plot given the recorded coordinates, and can make viewing a large amount of reports much easier.

**Fit Criterion:** A mapping API is displayed with visible reports plotted.

**Sample I/O:** N/A

### 2.2.9 Product shall provide a tagging interface to narrow down report category

**Rationale:** Most hazards and obstacles can be narrowed down to a small set of possibilities, including things like flooding, downed utilities, or debris. While this list may be extensive depending on the type of disaster, by allowing the user to select pre-defined attributes we can vastly improve search and display capabilities for users who are looking for specific kinds of hazards. This makes it significantly easier for organizations looking for related events. One such example would be for power companies looking to repair power lines. If they can filter on downed utilities and power poles, then it can decrease planning / deployment time even before the area is safe to traverse.

**Fit Criterion:** A tagging interface is provided in the report form. Search fields should also include the tagging interface.

**Sample I/O:** N/A

### 2.2.10 Product shall allow submission of video / photo evidence in report

**Rationale:** To further increase the credibility of user claims, the product should allow users to submit recorded media as evidence of the hazard. In situations where a simple text based report is inadequate, organizations and users can view the recorded footage and further analyze the threat in question.

**Fit Criterion:** A field for media submission is provided in the report form.

**Sample I/O:** N/A

### 2.2.11 Product shall allow submission of event time ranging from the current time to 24 hours prior

**Rationale:** Some individuals may need to generate reports after observing the hazards, due to non-stop traveling or conflicting activities. A user should be able to create reports for hazards they've witnessed in the past 24 hours.

**Fit Criterion:** An autofilled time field should be provided in the report form.

**Sample I/O:**

- Report is submitted with timestamp in allowed range: Server records report and returns a success message.
- Report is submitted with timestamp outside of allowed time range: Server returns an error and notifies the user of the time conflict.

### 2.2.12 Product shall disallow a report occurring before the disaster's start date

**Rationale:** In the edge case where a disaster event is created and a user submits a report up to 24 hours prior to the start date, the server should disallow submission. If a user is cataloging and preparing a list of reports for a disaster that hasn't occurred yet, it's very possible that it's malicious activity and should be prevented in case. Ensuring that timestamps only occur after the disaster start date makes data constraints on the back-end easier.

**Fit Criterion:** Reports attempting to post before a disaster's start date should be prevented.

**Sample I/O:**

- Report is submitted with a timestamp occurring after event creation date: Server records report and returns a success message.
- Report is submitted with a timestamp occurring before the event creation date: Server returns rejection message notifying the user to correct their report timestamp.

**2.2.13 Product shall allow organizations to approve / disapprove user submitted reports**

**Rationale:** User reports by themselves should never be considered 100% trustworthy until an official organization has the opportunity to approve or disapprove the supplied information.

**Fit Criterion:** Organizations are provided with an approve/disapprove interface that allows them to verify or reject reported information for an area.

**Sample I/O:** N/A

**2.2.14 Product shall require organizations to record their related tags**

**Rationale:** By requiring organizations to record their related tags, we can simplify the search and report responsibilities based on an organization's tags. For example, utility companies should be able to report and change the status of user events pertaining to their tags, which would likely include utilities or power poles.

**Fit Criterion:** Accounts associated with an organization are given an interface to adjust their related tags.

**Sample I/O:** N/A

**2.2.15 Product shall enforce that organizations only adjust user reports that share a related tag.**

**Rationale:** By requiring that organizations only adjust reports pertaining to their affected fields, we can ensure no organization will step over their report boundaries and minimize the damage if an organization account were to be compromised to a sub set of their tags.

**Fit Criterion:** Organization accounts are restricted to approving / disapproving reports pertaining to their related tags.

**Sample I/O:**

- Organization attempts to approve / disapprove related user reports: Server records response and approval / disapproval timestamp and returns a success message.
- Organization attempts to approve / disapprove unrelated user reports: Server returns an error message notifying the user that the report is not under their organization's related tags.

**2.2.16 Product shall adjust a user's reputation based on approved and disapproved reports**

**Rationale:** By assigning a reputation value to users, organizations and users can quickly discern the credibility of a report.

**Fit Criterion:** A reputation is displayed for a user next to their posted reports.

**Sample I/O:** N/A

**2.2.17 Product shall increment a user's reputation by 25 points when approved by an organization**

**Rationale:** Approval from an organization is confirmation that a user report is accurate and reliable. Every time a user's report is approved, their reputation should increase by a marginal amount to show that they have posted accurate information in the past.

**Fit Criterion:** On approval, a user's reputation should be increased by 25 points.

**Sample I/O:** N/A

**2.2.18 Product shall decrement a user's reputation by 10 points when disapproved by an organization**

**Rationale:** Disapproval from an organization is confirmation that a user report is either false or misleading. Every time a user's report is disapproved, their reputation should fall by a marginal amount to show that they have posted false or misleading information in the past.

**Fit Criterion:** On disapproval, a user's reputation should be decreased by 10 points.

**Sample I/O:** N/A

**2.2.19 Product shall enforce a limit on the number of reports by a single user in a 100 meter radius by  $\text{floor}(\text{reputation} / 100)$**

**Rationale:** Experienced users who have submitted several approved reports in the past will most likely submit valid reports in the future. To defend against malicious spam, a cap is placed on the amount of reports a user can make in a 100 meter radius. The higher your reputation is, the more reports a user is permitted to send, as a user with a higher reputation is less likely to maliciously use the system.

**Fit Criterion:** A user's max report amount in a 100 meter radius is limited to the floor of their reputation divided by 100 number of posts.

**Sample I/O:** N/A

**2.2.20 Product shall display an area average for similar reports in an area**

**Rationale:** As multiple users will be posting similar events in an area, an average should be generated to show the reputation of users. If a lot of high reputation users are reporting on the same thing in an area, the confidence of those reports should also go up.

**Fit Criterion:** An average reputation for a viewed area on a map should be visible.

**Sample I/O:** N/A

## 2.3 Product 2 - Communication Protocol

### 2.3.1 Product shall provide a public/private key decentralized information distribution protocol

**Rationale:** Constant communication is very difficult to maintain in a disaster scenario. To ensure that digital information is properly transmitted and maintained, a protocol has to be designed that is both secure and decentralized to facilitate the propagation of information.

**Fit Criterion:** A decentralized information distribution protocol is implemented **Sample I/O:** N/A

### 2.3.2 Product shall build a chain of trust from a built-in starting piece of data

**Rationale:** A chain of trusted data is a well known way of ensuring data arrives in a particular order and is not modified. In order to facilitate the chain's construction, an agreed on starting point needs to be built in to all clients implementing the protocol that declares all needed starting variables to begin chain construction.

**Fit Criterion:** A common starting point is built into the protocol.

**Sample I/O:** N/A

### 2.3.3 Product shall allow the creation of specialized (miniaturized) nodes

**Rationale:** In order for the protocol to be readily accepted and utilized, the amount of data storage required should be relatively small, within at least 25MB. Since most chain based data structures require all data be downloaded, an effort should be made to find a solution in which sub sets of data can be extracted and utilized.

**Fit Criterion:** A specialized node is successfully able to reduce the chain's storage size.

**Sample I/O:** N/A

### 2.3.4 Product shall implement authorization / revoke transactions for organization keys

**Rationale:** While organizations are able to register keys to add identifying information to the chain, they must also be able to revoke the keys in the event of a data breach or key replacement. Organizations should also be able to add new sub-keys for each disaster for further security.

**Fit Criterion:** Organizations can transmit revoke / authorization transactions

### **2.3.5 Product shall require organizations to sign participation keys in each disaster epoch**

**Rationale:** To assist in the construction of specialized nodes, organizations requesting the ability to transmit data must register their identification keys in each disaster they wish to transmit to. By doing so we can reduce the amount of data that needs to be saved by the specialized nodes.

**Fit Criterion:** Organizations are able to register keys in each event epoch.

**Sample I/O:** N/A

### **2.3.6 Product shall require signing machines be used to register organization's authorization keys**

**Rationale:** The use of signing machines is the prerequisite requirement for implementing Proof of Authority. Signing machines also allow the product to filter permitted transactions and control what data is added to the chain. For an organization to register their first key, they must request it and have a signing machine append the information on their behalf.

**Fit Criterion:** A signing machine and only a signing machine can add an organization's first key.

**Sample I/O:** N/A

### **2.3.7 Product shall implement a no-fee transaction system**

**Rationale:** The protocol's transaction system will be initially based on a generic blockchain's transaction system, which requires the use of transaction fees to prevent transaction spam and to incentivize transaction mining by mining machines. In this protocol, the use of signing machines removes the need for mining incentives and transaction spam will be limited as keys must be whitelisted before transactions can be made, making the use of transaction fees obsolete.

**Fit Criterion:** No fee is needed to send a transaction.

**Sample I/O:** N/A

### **2.3.8 Product shall allow the recording of text data**

**Rationale:** Most blockchains allow the addition of text data through no action opcodes. Since this chain will be built specifically for chain data, it should have codes reserved specifically for the addition of any kind of text data, compressed or uncompressed in place of balance transfers.

**Fit Criterion:** Special structures are implemented to facilitate text transfer.

**Sample I/O:** N/A

### **2.3.9 Product shall include interface/API support for non-conventional transmission protocols**

**Rationale:** To make adoption easier and to assist data transmission in difficult or hard hit areas, an API should be constructed to allow other protocols to transmit and receive this protocol's data. For example, if users wanted to facilitate the use of AM radio transmissions to broadcast data, it would be trivial to write a driver to send the data, receive it, and decode it.

**Fit Criterion:** An interface / API is created to simplify the process of transmitting over other protocols.

**Sample I/O:** N/A

### **2.3.10 Product shall provide structure for administrative transactions**

**Rationale:** Administrative transactions should be easy to identify so that specialized nodes can easily detect and record them. By keeping track of administrative transactions, specialized nodes can simplify their pruning logic when dealing with non event-related blocks of data.

**Fit Criterion:** A method to identify administrative transactions is implemented

**Sample I/O:** N/A

### **2.3.11 Product shall implement a method to close or seal expired disasters**

**Rationale:** Organization keys that are delegated to a disaster should be revoked after the disaster is finished. In addition, some method of ensuring that no extra information is appended after the disaster.

**Fit Criterion:** Some method to signal the end or closing of a disaster is implemented.

**Sample I/O:** N/A

### **2.3.12 Product shall implement a post-op report interface for closed disasters**

**Rationale:** Some organizations will release post operation reports years after the event occurred. To allow these reports to be appended without compromising the disaster seal, a special transaction identifier and report key for organizations must be registered.

**Fit Criterion:** An interface to facilitate the addition of post operation reports is implemented.

**Sample I/O:** N/A



**2.3.13 Product shall provide an encryption interface for private event-related data**

**Rationale:** Some organizations may wish to use the protocol, but will need to send sensitive data. To facilitate this requirement, some form of data encryption should be supported.

**Fit Criterion:** Data encryption is supported through some interface.

**Sample I/O:** N/A

**2.3.14 Product shall provide an interface to look up other organization's public keys**

**Rationale:** To facilitate the requirement for encrypting data, an interface for looking up public key - organization pairs should be implemented.

**Fit Criterion:** A search interface for organization public keys is implemented.

**Sample I/O:** N/A

**2.3.15 Product shall provide programmable interfaces to allow program interaction with custom events**

**Rationale:** Some organizations will prefer to integrate the protocol with existing systems. Thus, an API to replicate most transmission / receiving functions should be implemented to make the integration process easier.

**Fit Criterion:** An interface / API is created to provide external access to protocol functions

**Sample I/O:** N/A

**2.4 Product 3 - Logistics Interface**

**2.4.1 Product shall provide an interface for filtering operations pertaining to an organization**

**Rationale:** To simplify logistical operations and the associated data flows, users should be able to filter out events related to the organization or to a subset of related tags. For example, a hospital may only be interested in medical or supply related events, and should be able to filter based on these parameters.

**Fit Criterion:** A method to filter operations and related data entries is implemented.

**Sample I/O:** N/A

**2.4.2 Product shall provide an interface for creating tasks for organization members**

**Rationale:** By implementing a task delegation system, users inputting specific events for entities like operating teams can easily view and report on their assigned tasks.

**Fit Criterion:** A task delegation / creation interface is created.

**Sample I/O:** N/A

#### **2.4.3 Product shall provide a method for setting the priority and categories of tasks**

**Rationale:** To assist in logistical operations, a method to view events / data points by priority and category should be implemented. By allowing organizations to set and view the priority of certain events / tasks, they can better plan out operations and distribute tasks.

**Fit Criterion:** An interface to set and view task categories and priorities is implemented.

**Sample I/O:** N/A

#### **2.4.4 Product shall provide a method for adjusting the visibility of certain tasks**

**Rationale:** Not all events should be public or organization specific. In situations where the visibility of a task is confined to a group or individual, this product should be able to facilitate it.

**Fit Criterion:** An interface to adjust a task's visibility is implemented.

**Sample I/O:** N/A

#### **2.4.5 Product shall provide a permissions system to adjust who can view/edit/create tasks**

**Rationale:** A single user, or every user, shouldn't have the ability to globally modify any event in an organization's logistics platform. Thus, a permissions system should be implemented to allow organization administrators to set access controls.

**Fit Criterion:** A permissions system is implemented.

**Sample I/O:** N/A

#### **2.4.6 Product shall provide an interface to create custom data objects**

**Rationale:** For organizations that have logistical requirements that are not satisfied using generic categories or objects, some support to add custom data objects should be implemented. This can help instances where, for example, an organization specializing in supplies has custom data attributes to attach to trackable items.

**Fit Criterion:** An interface to create custom data objects is implemented.

**Sample I/O:** N/A