Implementation Guidance:

Information Sharing Standards for Crisis Management and Mutual Aid Technology

DRAFT

Version 1.0
May 2019

National Alliance for Public Safety GIS (NAPSG) Foundation

5335 Wisconsin Ave., NW | Suite 440 | Washington, DC 20015

Table of Contents

Table of Figures

Table of Tables

1 Overview

Mutual Aid and Crisis Management Systems (MACM) in the Emergency Management Enterprise (EME) suffer from a lack of use of interoperability and information exchange standards for system to system interoperability. Information that can be shared directly between systems improves reliability, accountability, speed, and accuracy. Phone calls, radio calls, email, etc. that involve human intervention to transmit and receive information can slow the flow of information and introduce human errors, including mis-transcription, misinterpretation, and delay. Information shared between systems using identical data standards are less prone to these errors and can speed up the flow of information, since it can happen in near real-time. Accurate, reliable, and timely information is critical in emergency management.

This document provides a simple guide on choosing the appropriate information standard for a given need, as well as, identify some key aspects for communicating that information between systems. The goal is to help facilitate the selection of a future system that meets the MACM information sharing requirements and improve overall interoperability in the EME.

1.1 Recommended Standards

The MACM Standards Survey (a comprehensive activity to research best standards to meet MACM requirements) identified a variety of standards from Organization for the Advancement of Structured Information Standards (OASIS) Emergency Data Exchange Language (EDXL) suite of standards that are directly applicable to MACM. These include the following:

- Common Alerting Protocol (CAP)
- Hospital Availability Exchange (HAVE)
- Distribution Element (DE)
- Resource Management (RM)
- Situation Report (SitRep).

The standards listed above are focused on information exchange for emergency management systems. It is recommended that depending on need, multiple standards would need to be implemented to cover the breadth of information to be shared.

An additional standard, the Emergency Management Loose Coupler, from the National Information Exchange Model (NIEM) is recommended for more light-weight information sharing. This standard focuses on the high-level who, what, when, where of information in the EME. It has optional sections for more detailed information about specific types of data.

1.2 Implementation

In addition to providing a single list of applicable standards to MACM, NAPSG seeks to provide first responder decision makers with guidance on how these standards could/should be implemented to improve interoperability, resource management, situational awareness, and crisis management. This guidance discusses the need for Application Programming Interfaces (API), the current modes of network communication, and architectures to cover a full implementation for communication needs. Finally, an appendix is provided with details and samples of what those APIs and architectures might look like to provide additional technical insight.

When looking to implement a new software for emergency response use, it is important for the interested party to focus on the high level requirement areas for a particular system so that the software purchased is of long term use to the purchaser. While user interface and functionality requirements are important, the underlying data, how it is stored and shared, provide the true longevity of any system.

Areas of true interest in the MACM community focus on data sharing aspects of the system: standards and communication. Standards provide an agreed upon data format that knowledgeable groups have vetted through a series of community involved reviews and can allow systems to speak easily to one another. This reduces the need for “customization” and “development” that will often come as an offer with the particular software or system and in turn reduces cost. The purchaser should ask questions of their needs or systems like: What are we trying to share? Whom are we trying to share data with? What are the major elements of that data that we need to share (incident info, resource info, patient info, etc.)? Once these answers are identified, the purchaser can then run through the easy to follow workflow provided in this guide to identify the best standards to cover their needs. As noted, there is no one-stop-shop standard, so multiple standards likely will apply.

Once standards are identified, the purchaser can then turn their attention to the aspect of communication for their system. As noted above, communication when identified as the way systems “speak” to one another, can also limit the amount of customization or cost associated with procuring a particular system. If the purchaser identifies a particular type of communication that surrounding jurisdictions utilize, it would be in their best interest to shop for that same style of communication to ensure interoperability. The purchaser can also use the easy to follow communications workflow included in this guide to determine best communication choice for their software.

After running through these activities, the purchaser should be able to identify the types of interoperable elements most important to their system and can request that of software providers. If the proper system/tool doesn’t currently exist, the informed demand by said purchasers should eventually create a market for this to exist. Gone will be the days where systems are stove-piped by proprietary development and expensive support; an ecosystem of strong software choices that boast interoperability will be created.

The following is meant as a carefully procured guide to allow and empower the first responder/mutual aid community to start building a strong support system for interoperability of important emergency data.

2 Standards Workflow

The following section describes the decision workflow for choosing the appropriate standard based on need. The intent is to aid the selection of which standards should be used based on what information needs to be shared. The workflow in the figure below is split into two branches based on the type of information needed to be shared. The Situational Awareness branch of the workflow aligns itself to the Situational Awareness Requirements from the NAPSG Standards Survey, while the Resource Management branch aligns itself to the Resource Management Requirements.

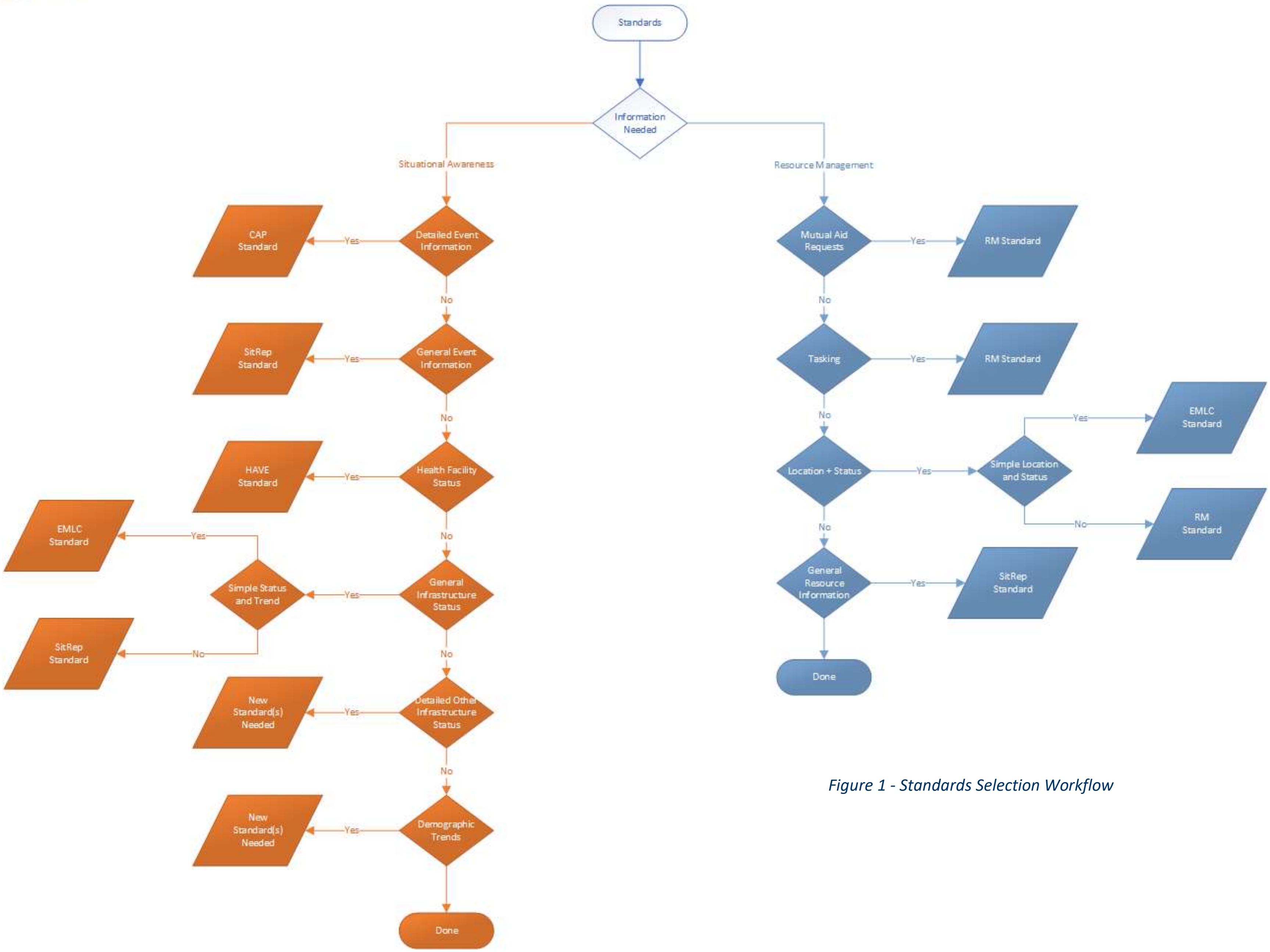

Figure 1 - Standards Selection Workflow

2.1 Situational Awareness Standards

The Situational Awareness workflow branch asks a series of questions about the kind of information needing to be shared. These are loosely based on the NAPSG Situational Awareness requirements. They are broken down in the following table.

Table 1 - Situational Awareness Requirement Association

General Category	SA Requirements	Workflow Questions
Event Information	Event Scale, Event forecast, Event Magnitude	Detailed Event Information, General Event Information
Infrastructure Status	Critical Infrastructure Impact	Health Facility Status, General Infrastructure Status, Detailed Other Infrastructure Status
Demographics	Demographic Trends	Demographic Trends

A Yes to the question points in the direction of a recommended standard, while a No simply moves you to the next question. The following table explains the intent of each workflow question.

Table 2 - Situational Awareness Workflow Questions

Workflow Question	Explanation
Detailed Event Information	Is detailed information needed about a specific event, including current status and location, future predictions and impact area, and how to respond to a particular event
General Event Information	Is more summary level, current information needed about an event
Health Facility Status	Is detailed information needed about a particular health care facility, including bed status, ER capacity, EMS response availability, etc.

General Infrastructure Status	Is a general status needed for a particular infrastructure facility, such as Power Plant X, or for a general infrastructure category, such as communication.
Detailed Other Infrastructure Status	Is detailed information, similar to the type of information in health facility status, needed for a particular, non-health infrastructure facility
Demographic Trends	Is information needed about demographic trends, or demographic information of an area in general

The intent of the workflow is to match high-level information sharing needs to a recommended standard as simply as possible. In a few cases, there is no existing standard that meets an information need from the NAPSG Situational Awareness requirements.

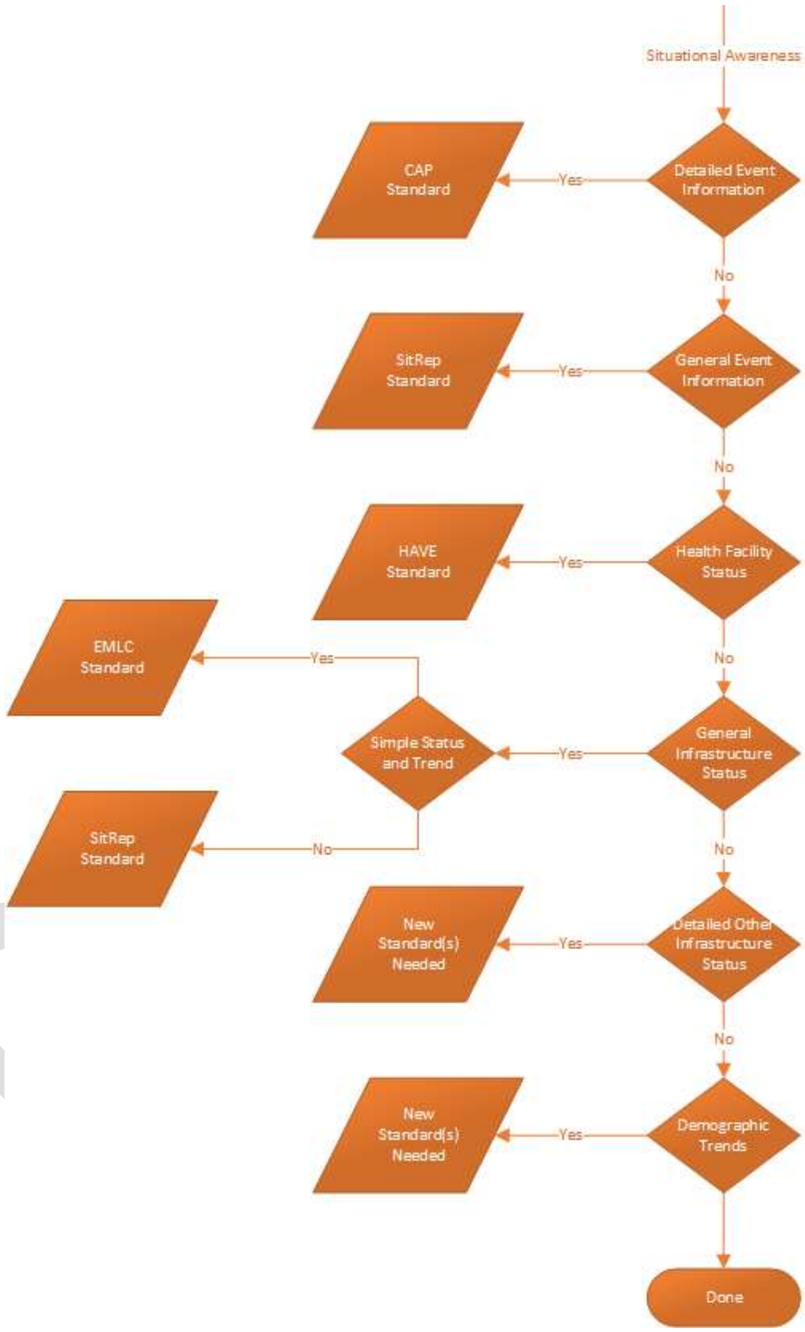

Figure 2 - Situational Awareness Standard Selection

2.1.1 Detail Event Information – EDXL CAP

If detailed event information, such as severity, magnitude, consequences, directed action, location, etc. is required, then the OASIS EDXL Common Alerting Protocol (CAP) standard is recommended. CAP is the standard currently used by the Federal Emergency Management Agency (FEMA) Integrated Public Alert and Warning System (IPAWS) for issuing alerts and warnings around the country. CAP is also used by several other countries, such as Canada and Australia, for issuing alerts and warning.

2.1.2 General Event Information – EDXL SitRep

If more high-level, summary event information is required, then the OASIS EDXL Situational Report (SitRep) standard is recommended.

2.1.3 Health Facility Infrastructure Status – EDXL HAVE

If the status and state of health facilities are required, as part of infrastructure status or not, then the OASIS EDXL Hospital Availability Exchange (HAVE) standard is recommended.

2.1.4 General Infrastructure Status – NIEM EMLC / EDXL SitRep

Currently, the only other standards that support infrastructure status are the SitRep and NIEM Emergency Management Loose Coupler (EMLC) standards. These standards only support general infrastructure status.

2.1.4.1 NIEM EMLC

The EMLC standard supports infrastructure status and trending for individual infrastructure entities, such as a power plant, highway, etc. Each entity is uniquely identified and located in addition to their status.

2.1.4.2 EDXL SitRep

The SitRep standard supports infrastructure status in a general category or capability, such as waterways, telecommunication, sewage, etc. Individual entities are not identified, and trending is not included.

2.1.5 Other Infrastructure Details – New Standards Needed

There are no other standards similar to HAVE that are currently available for the other infrastructure types. New standards will need to be developed to capture detailed information about specific infrastructure capabilities, such as power plants, sewage plants, etc.

2.1.6 Demographic Trends – New Standards Needed

New standards would need to be developed to capture the demographic trending information called in the Situational Awareness Requirements in the standards survey. This information is not currently available in the existing MACM standards.

2.2 Resource Management Standards

The Resource Management workflow branch asks a series of questions about the kind of information needing to be shared. These are loosely based on the NAPSG Resource Management requirements. They are broken down in the following table.

Table 3 - Resource Management Requirement Association

General Category	SA Requirements	Workflow Questions
Mutual Aid	Resource Kind, Resource Response Availability, Resource Readiness, Deployment Time, Resource Cost	Mutual Aid Request
Tasking	--	Tasking
Current Status	Resource Kind, Resource Response Availability, Resource Readiness	Location + Status
General Status	Resource Kind, Resource Response Availability, Resource Readiness	General Resource Information

A “Yes” to the question points in the direction of a recommended standard, while a “No” simply moves you to the next question. The following table explains the intent of each workflow question.

Table 4 - Resource Management Workflow Questions

Workflow Question	Explanation
Mutual Aid Requests	Is a request and response for aid needed, including costing
Tasking	Is the ability to task a responding resource needed
Location + Status	Is the current status and location of a particular resource needed
General Resource Information	Is a general status about a particular responding resource and/or all responding resources

The intent of the workflow is to match high-level information sharing needs to a recommended standard as simply as possible. In a few cases, there is no existing standard that meets an information need from the NAPSG Situational Awareness requirements.

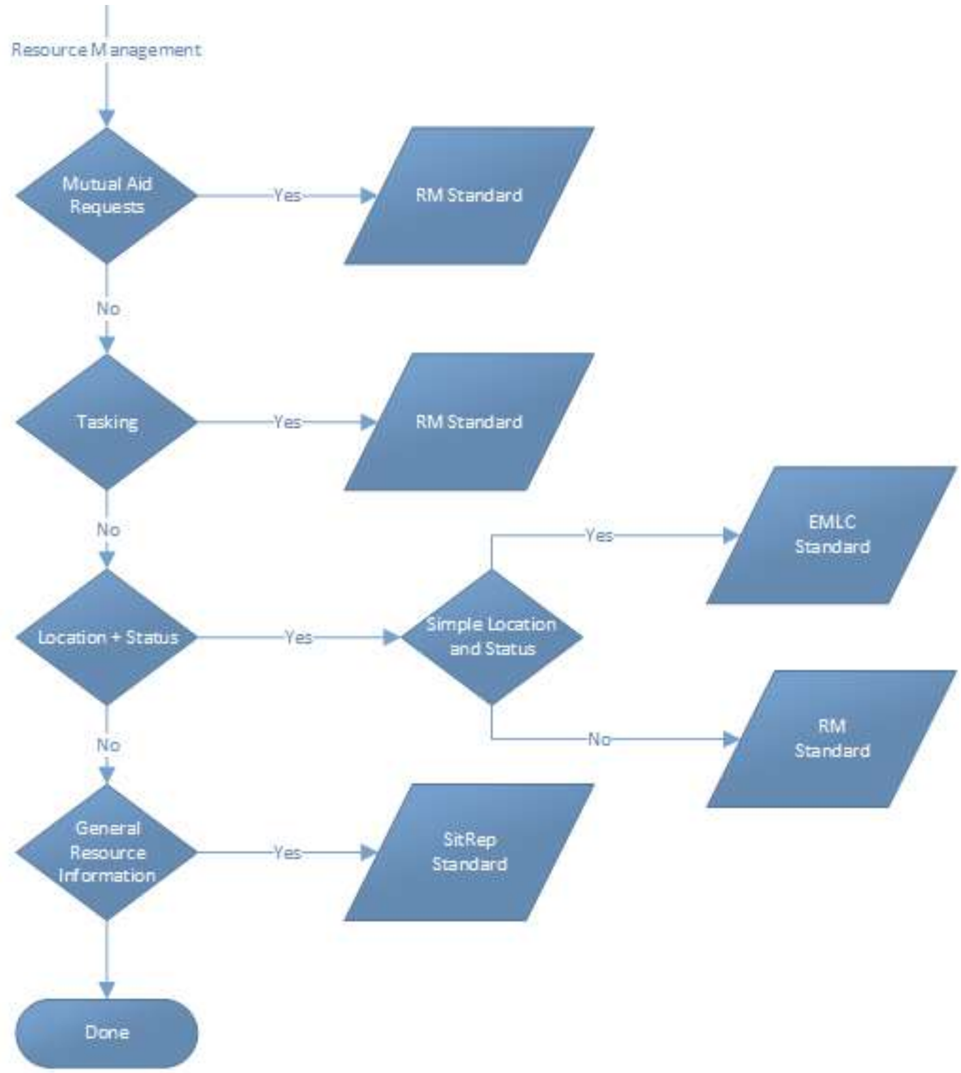

Figure 3 - Resource Management Standard Selection

2.2.1 Mutual Aid Requests – EDXL RM / NIEM EMLC

If making mutual aid requests and responses are required, then depending on the duration of the required aid and formality of the aid request either the OASIS EDXL Resource Management (RM) standard or NIEM EMLC standard is recommended.

2.2.1.1 NIEM EMLC

The NIEM EMLC supports simple mutual aid requests and responses without costing information. This standard is intended for short-term aid requests for an escalating incident most likely from neighboring jurisdictions.

2.2.1.2 EDXL RM

The EDXL RM supports both long-term and short-term aid requests, including costing information.

2.2.2 Resource Tasking – EDXL RM

If simple resource tasking is required, then the OASIS EDXL RM standard is recommended. RM supports simple text strings to indicate mode of transportation, navigation instructions, and reporting instructions as part of the assignment instructions.

2.2.3 Location + Status – EDXL RM / NIEM EMLC

If resource location and status are required, then depending on the need for real-time accuracy or not, either the EDXL RM standard or NIEM EMLC standard is recommended.

2.2.3.1 NIEM EMLC

The NIEM EMLC standard is designed to provide light-weight, real-time updates for responder location and status.

2.2.3.2 EDXL RM

The EDXL RM standard is designed for more periodic or transition updates for responder location and status, such as changes in availability or changes to estimated departure and arrival times.

2.2.4 General Resource Information – EDXL SitRep

If an overview of the resources currently assigned to an incident is required, then the EDXL SitRep standard is recommended. SitRep supports a “Response Resources Totals” report which provides an overview of the current resources, what agencies they work for, what they are assigned to, their status, etc. The SitRep does not provide resource location, unlike the EMLC and RM.

2.3 Areas of Concern/Improvement

2.3.1 CAP

CAP is missing a few of the key fields called out in the Situational Awareness Requirements[1]. These include:

- extent of the event
- indirect event consequences

CAP has an extension mechanism (though the Parameter fields) that would allow this information to be added. If this mechanism is used, the system API should describe its usage to facilitate other systems' interoperability. Alternatively, CAP could be updated through OASIS to include this type of information.

3 Communication

The following section describes the decision workflow for choosing the appropriate communication methodology between systems based on need. The intent is to aid in the vendor selection process when comparing different systems. The aim is to avoid the stove-piped nature of many of the existing systems in the EME. The workflow in the figure below is split into two branches based on the information needs to be shared. This workflow focuses on the two most popular methods of sharing information between enterprise-level systems over the Transmission Control Protocol and the Internet Protocol (TCP/IP): Hypertext Transfer Protocol (HTTP) or Message Queuing Telemetry Transport (MQTT).

[1] See NAPSG Standards Survey

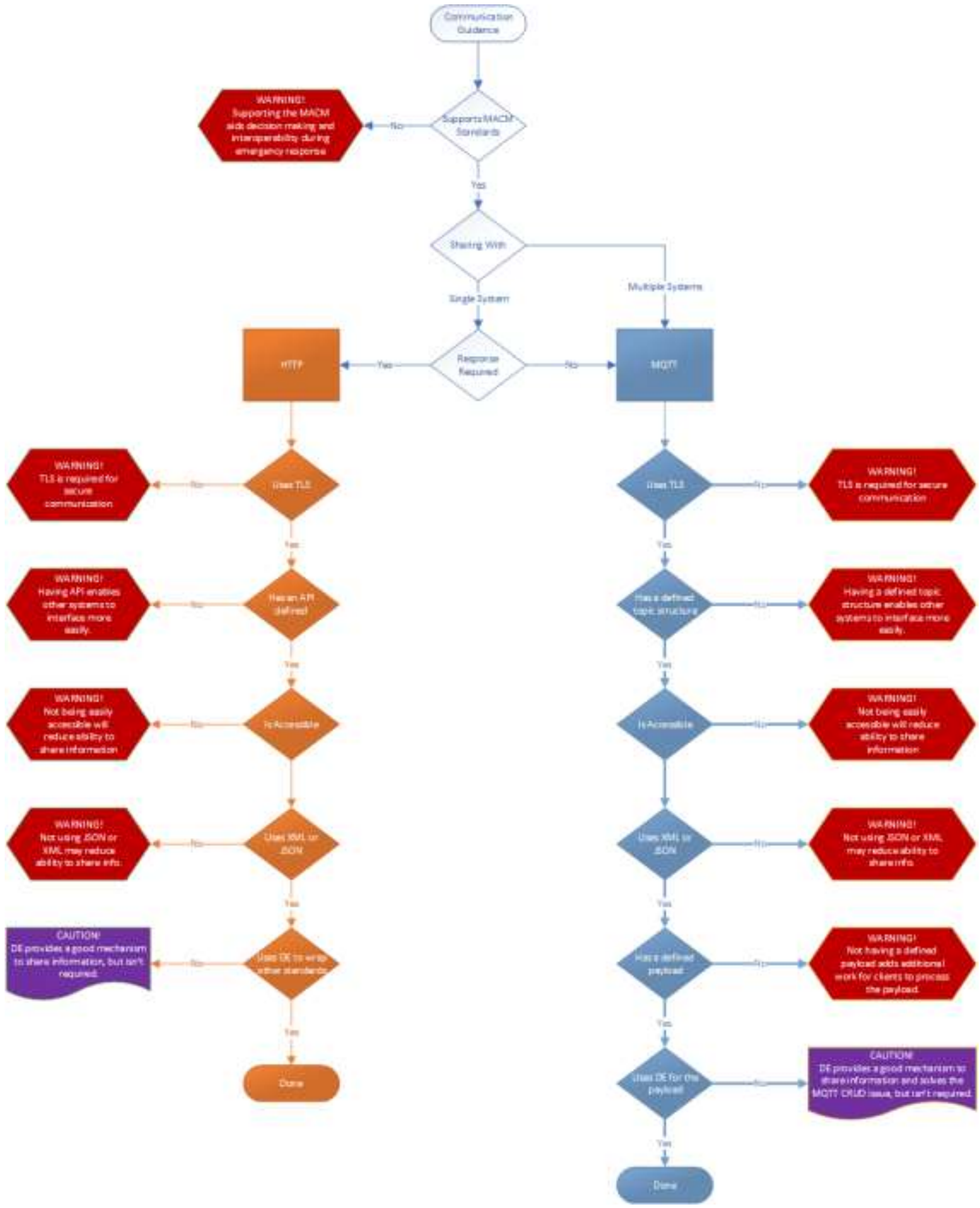

Figure 4 - Communication Methodology Selection Workflow

3.1 General Questions

The communication guidance workflow starts with a couple of questions to help determine which of the two most popular communication methods is recommended. However, the main question to ask is whether or not the system in question supports the recommended MACM standard(s) as determined in the standards workflow above. If the system in question does not, that defeats the intent of these documents and guidance, and is strongly advised to avoid. Which method to use is largely dependent on how the information needs to be shared between two systems or multiple systems and whether not a response to a message is required. In some instances, because of business, operational, policy, or technological requirements, a positive response is required for some piece of information being shared, but this is not always the case.

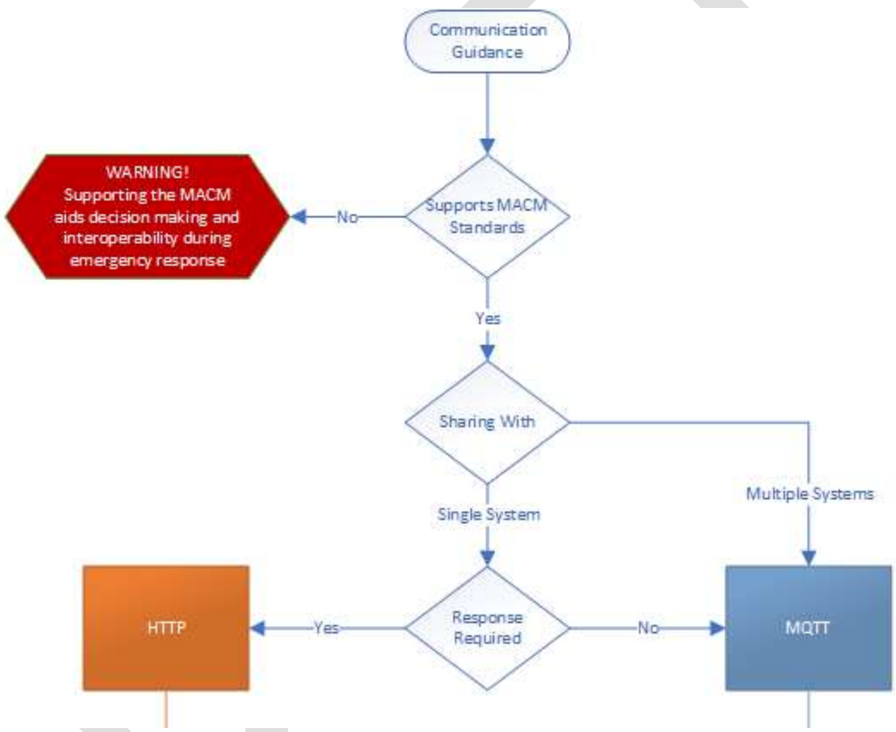

Figure 5 - Communication Main Questions

3.2 HTTP Branch Workflow

The HTTP workflow branch asks a series of questions about how the information will be shared. They are broken down in the following table.

Table 5 – HTTP Workflow Questions

Workflow Questions	Explanation
Uses TLS	TLS stands for Transport Layer Security and is the current cryptographic protocol for securing network traffic while in transit.
Has an API Defined	API stands for Application Programming Interface. It defines the methods and information needed to interface to a system.
Is Accessible	How accessible is the system? Can it be accessed from the internet?
Uses XML or JSON	XML and JSON are two of the most widely used data formats. This is how the information is structured in a message or file.
Uses DE to wrap other standards	Does the system use the OASIS EDXL DE standard to transport information?

The intent of the workflow is to ensure the HTTP system supports security and enables information sharing.

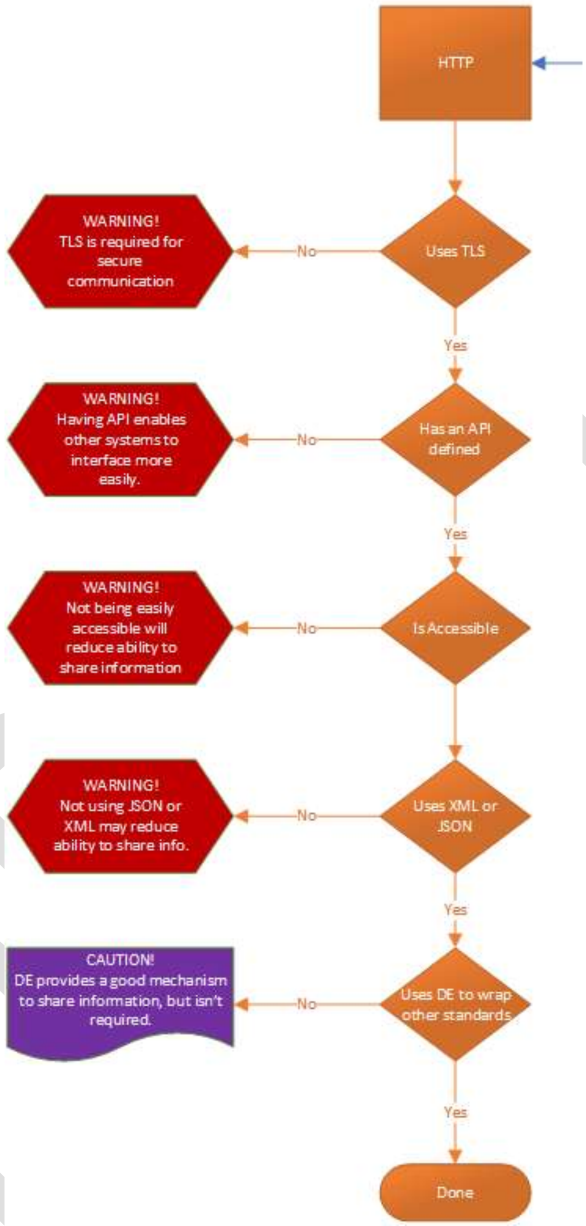

Figure 6 - HTTP Branch Workflow

3.2.1 Uses TLS

Using TLS ensures messaging over the network is secure. This is a requirement for sharing information between systems.

3.2.2 Has an API Defined

Having a defined API allows other system vendors to more easily sharing information to and from this system. Without it, other systems may not be able to share information to this system. Certainly, they will not be share information easily, as they may not understand what methods are available for them to use and what information is expected. API's also need to define the security aspects of the system, so other vendors know what methods to use. For example, does the system use username and password authentication or certification authentication or some other mechanism? Does it support single sign on? This information is important to other vendors as they determine how to share information to and from this system.

3.2.3 Is Accessible

How accessible is the system? Can other systems access it from the internet? If a system is behind paywalls, on a private network, or some other hinderance, this will reduce the ability of other systems to share information with it. While a system doesn't have to be on the open internet, it should be accessible from it. For example, a system could be running in a private cloud environment, but still accessible through the general internet. The problem arises if that system requires others to be in the same private cloud environment to access it.

3.2.4 Uses XML or JSON

Does the system support either XML or JSON or both for messaging? Other formats may reduce the ability to easily share information. Most modern systems use either XML or JSON.

3.2.5 Uses DE to wrap other standards

Does the system use the OASIS EDXL DE to transport other information? While this is not a hard requirement, the EDXL DE provides an excellent mechanism to share a wide-variety of information. It acts as a wrapper for other information standards, much like an envelope wraps a letter. A DE-based system would be able to send and receive all of the recommended MACM standards without the need for different endpoints or topics necessarily. This simplifies the overall system to system architecture and would aid in the ability to share a wide array of information. New standards could be added as EDXL DE payloads, supporting new and unexpected CONOPS, without the need to update the interfaces between systems. It is highly recommended that the EDXL DE be used as the primary transportation mechanism.

3.3 MQTT Branch Workflow

The MQTT workflow branch asks a series of questions about how the information will be shared. They are broken down in the following table.

Table 6 – MQTT Workflow Questions

Workflow Questions	Explanation
Uses TLS	TLS stands for Transport Layer Security and is the current cryptographic protocol for securing network traffic while in transit.
Has a Defined Topic Structure	Is the topic structure defined? Does it have an easy to use template pattern?
Is Accessible	How accessible is the system? Can it be accessed from the internet?
Uses XML or JSON	XML and JSON are two of the most widely used data formats. This is how the information is structured in a message or file.
Has a Defined Payload	Have the payloads been defined, including whether XML or JSON is expected?
Uses DE to wrap other standards	Does the system use the OASIS EDXL DE standard to transport information?

The intent of the workflow is to ensure the MQTT system supports security and enables information sharing.

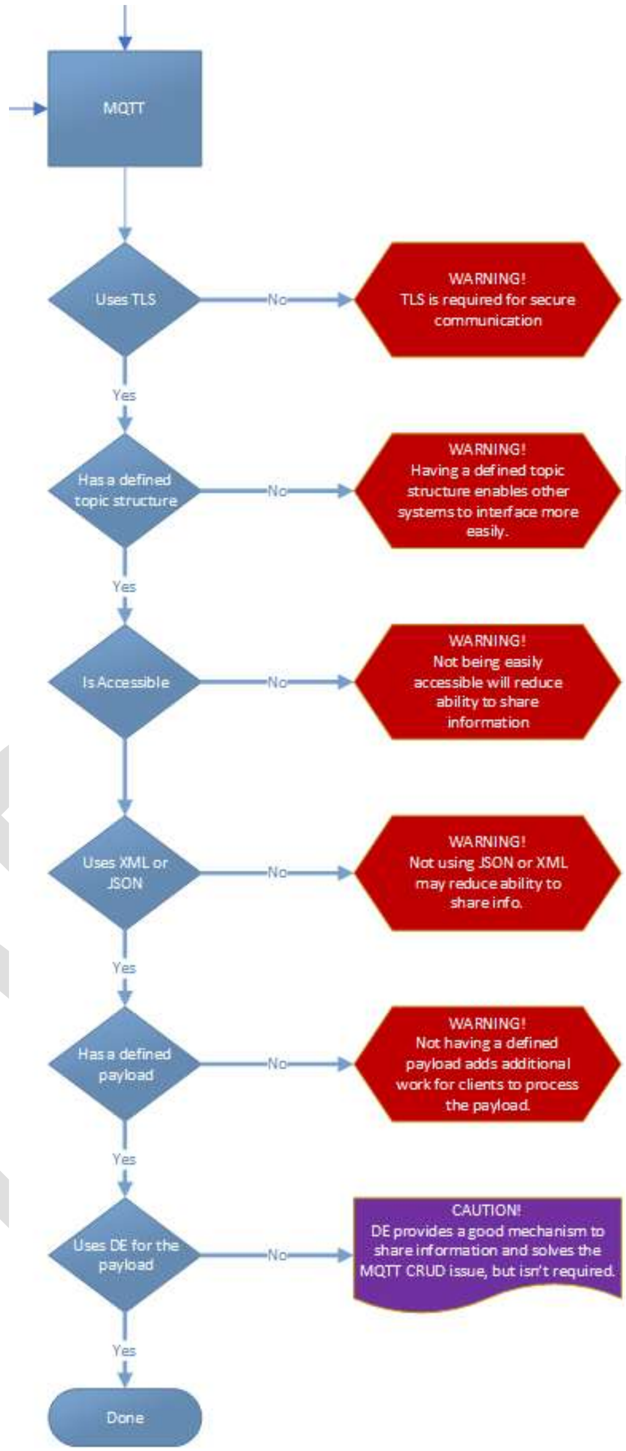

Figure 7 - MQTT Branch Workflow

3.3.1 Uses TLS

Using TLS ensures messaging over the network is secure. This is a requirement for sharing information between systems.

3.3.2 Has a Defined Topic Structure

Topics in MQTT are a way to filter and categorize information. MQTT clients publish information and receive information through these topics. For example, a client might publish resource location on a topic like /<agency>/<unit id>/location, which might look like in real life as /lafd/ra52/location. Other clients would subscribe to the /lafd/ra52/location topic to receive updates for the Los Angeles Fire Department rescue ambulance 52’s location.

The difficulty in using MQTT topics is they can be organized in a variety of ways and by the client. There is no discoverability mechanism that allows other clients to know what topics are available that they can subscribe to. This makes information sharing more difficult. By defining the topic structure ahead of time as part of an API for the system, other vendors will understand what the expectations of the system are, which improves the information sharing situation.

3.3.3 Is Accessible

How accessible is the system? Can other systems access it from the internet? If a system is behind paywalls, on a private network, or some other hinderance, this will reduce the ability of other systems to share information with it. While a system doesn’t have to be on the open internet, it should be accessible from it. For example, a system could be running in a private cloud environment, but still accessible through the general internet. The problem arises if that system requires others to be in the same private cloud environment to access it.

3.3.4 Uses XML or JSON

Does the system support either XML or JSON or both for messaging? Other formats may reduce the ability to easily share information. Most modern systems use either XML or JSON.

3.3.5 Has a Defined Payload

Payloads in MQTT are typically text-based and can be any type of text information. This can make it very difficult for receiving systems (i.e. clients that have subscribed to a topic) for parsing and understanding the information they are receiving. In improve information sharing the system’s payload format (XML or JSON) and content format (i.e. what data standard) should be defined as part of the system’s API. This will aid other vendors in understanding what to expect and how to parse the information coming from the MQTT server on a given topic. A mapping should be made for each topic in terms of format and content, so it is clear what is expected to be sent and received for each given topic.

3.3.6 Uses DE to wrap other standards

Does the system use the OASIS EDXL DE to transport other information? While this is not a hard requirement, the EDXL DE provides an excellent mechanism to share a wide-variety of information. In particular, it is well suited as an MQTT payload as it provides a mechanism internally to indict the usual CRUD (Create, Read, Update, Delete), as well as tasking and requests and responses, mechanisms that a lot of systems employ today. It is highly recommended that the EDXL DE be used as the primary transportation mechanism.

4 Security

Security is an extremely important aspect of information sharing between systems and involves multiple levels. Information should be secured in transit at a minimum and should ideally be secured while at rest. Transport Layer Security (TLS) is a cryptographic protocol to secure messages in transit over a network. It is a staple of secure web-communication. Both HTTP and MQTT support TLS communication in the form of HTTPS and MQTTs and used be used for communication between systems. As an additional layer of security, the message itself can be encrypted on top of TLS. If message encryption is required, this information and methodologies required to encrypt and decrypt the message must be a part of a system's API, so clients can behavior appropriately.

In addition to TLC and message encryption, system authentication and authorization requirements need to be considered. Authentication may be as simple as a username and password, or more complicated using client-server certificates. As note, username and password should never be exchanged in plain text and should be obscured/encrypted to secure them. Authorization is granted after authentication has been established and enables role-based permissions. System to system communication offers some unique challenges to authentication and authorization. Users having to enter multiple usernames and passwords to access information can hamper a mission. However, federated authentication and single sign on are not always possible. When possible, the goal of authentication should be to reduce user impact while maintain secure systems and communication. Client-server certificate maintenance and distribution can be challenging. Revoking and issuing new certificates can be time consuming and difficult.

While these challenges can be difficult to deal, security needs to be a forefront of system to system communication and access. Message integrity is critical to emergency managers, so they don't question the information being provided. Message provenance is also critical to emergency managers, so they understand the information being provided is from a reliable source. Communication and system security help ensure these things.

5 Mutual Aid Mass Casualty Scenario

Since this document has walked through various decision trees to determine the best applicable standards and further implementation guidance, we thought it pertinent to walk through a "real life" emergency scenario to show how these implementations could improve and enhance the way data is shared in the field. The premise of this scenario is a large-scale traffic accident involving multiple injuries, fatalities, vehicle fires, hazardous material leakage, and multi-jurisdiction response. This scenario will attempt to highlight how each of the identified standards could be used to improve situational awareness and response.

5.1 Setup

A large winter storm is impacting a small mid-west town with blizzard conditions and reported power outages. A regional Emergency Operations Center (EOC) has been activated to monitor and respond to the changing weather conditions. Emergency operations plans are in place to deal with the situation. Reports of a multi-vehicle pileup on the interstate are starting to trickle in. Response to the incident will be coordinated through the EOC.

5.2 Available Data

The EOC situational awareness system is already receiving health facility status (EDXL HAVE reports) from the regions' hospitals. This information includes bed status, emergency services status, and EMS services status. Additionally, the EOC is receiving real-time local agency CAD information about active incidents, unit status, and unit locations via the NIEM EMLC reports. Blizzard warnings and alerts (EDXL CAP) are being received from the FEMA IPAWS system as the National Weather Service updates them.

5.3 Missing/Future Data

In addition to the HAVE reports, a new type of report dealing with the electric grid would be useful in this situation. This information would include facility id, overall capacity, current status, and expected issue correction estimates. A new type of report dealing with DOT road information would also be useful in the situation. This information should include things like road id, road status, and expected future status.

5.4 Incident Field Data

During the incident, field reports (EDXL SitRep) can be generated from boots on the ground to indicate any immediate needs and initial observations from the scene. This information could also be generated by personnel in the EOC/dispatch from radio reports from the field.

Converting this information to SitRep reports allows the information to be shared more easily across systems.

5.5 Mutual Aid Requests

As the incident escalates, both automatic and longer-term mutual aid requests can be made to nearby jurisdictions using EDXL RM. The automatic request would not typically include costing information, but would include unit availability, unit capabilities, unit type, and estimated departure/arrival information. The long-term aid would most likely include costing information as well as the other fields.

5.6 Context Diagram

The following diagram shows what data could be provided by what system and in what format. It is intended to highlight what could be possible when the right standards are used. The communication methods (HTTP or MQTT) will be system dependent and are not displayed.

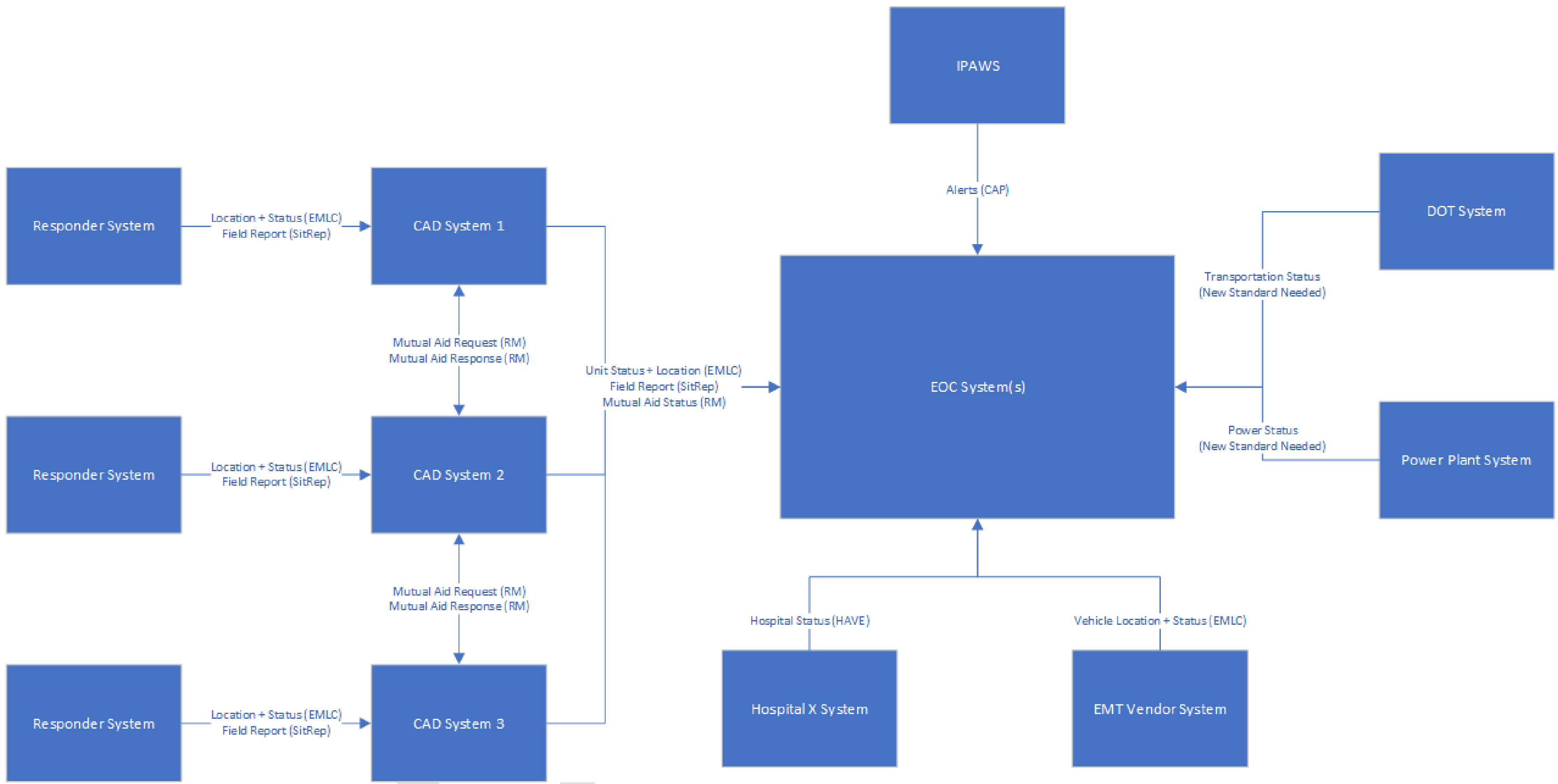

Figure 8 - Scenario Standards Overview

6 Appendix

This section contains more detailed information on HTTP and MQTT as well as some insights to some of the common issues with both.

6.1 MQTT

One of the challenges of using MQTT to send and receive information is its topic structure and lack of verbs, like HTTP has, to describe what to do with the new information. Depending on the type of information being shared, a delete or cancel operation might be necessary. If this type of information is not embedded in the standard, either a new topic will need to be used or the message payload will need to account for this operation. MQTT is a completely different technology and architecture from HTTP is not a 1:1 replacement of it. It is designed for light-weight messaging and rapid distribution to multiple clients.

6.1.1 Topics

Topics in MQTT are freeform but follow a hierarchical structure. Meaning a topic can be created about anything but typically follow a logical pattern. For example, if you had a home security system using MQTT where each security sensor reported its information to the MQTT message broker, you might have a topic structure that format like this: /home, home/<room>/, home/<room>/<sensor type>. <Room> and <sensor type> are template placeholders for actual data from a house. A real topic example for a house might look like this:

/home
/home/living room/
/home/living room/motion detector
/home/living room/smoke detector
/home/living room/window alarm
/home/mud room/
/home/mud room/door alarm
/home/kitchen/
/home/kitchen/smoke detector
/home/kitchen/flammable gas detector
Etc.…

An alternative topic structure might be by sensor instead of room: /home/<sensor type>/<room>

/home
/home/motion detector/
/home/motion detector/living room
/home/smoke detector/

/home/smoke detector/living room
/home/smoke detector/kitchen
/home/window alarm/
/home/window alarm/living room
/home/door alarm/
/home/door alarm/mud room
/home/flammable gas detector/
/home/flammable gas detector/kitchen
Etc.…

Either template is fine and works. This flexibility is great for using MQTT internally for a single system with multiple clients. However, this flexibility is very challenging when trying to use MQTT to connect external systems together. Part of an MQTT API must detail the what the topic structure is and how it is expected to be used. Without this, it will be very difficult for an external system to know what it can subscribe to for topics and what it can publish to for topics.

6.1.2 Topic Discoverability

Additional challenge for MQTT is that topics are not necessarily discoverable. It is completely API dependent. There is no native way to discover all of the topics in use on a given MQTT message broker. This is a significant challenge for the MACM domain where new topics could be easily added on the fly, and downstream (listening) clients would not know there is a new topic to listen to. If the system does not allow new topics to be added on the fly, this is not an issue. However, in a large-scale event, there may be a need to add new topics to organization information in new ways. One potential solution around this issue is support a well-known topic, that would be used by every MQTT MACM system, a topic like “/topics”. An MQTT client to an MACM system would know it could subscribe to this topic and receive the information about the current topics. The MQTT broker would need to be setup so any client subscribing to this topic would receive the full history on this topic. When the MQTT broker is started, a “master” client of the system would connect and publish the list of available/default topics to this topic. Once a new client subscribed to this topic, the existing topic information would be pushed to it. This information needs to be more than just a list of topic strings, it should also include the payload format, and payload content for each topic. The payload for the “/topics” topic would be a simple JSON object that contains the remaining topic information. It might look something like this:

```
{
        topics: [
                {
                        topic:”/content/have”,
                        format:”application/json”,
                        content:”urn:oasis:names:tc:emergency:EDXL:DE:1.0”
```

```
                description:"Hospital Status Updates"
        },
        {
                topic:"/content/rm/request",
                format:"text/xml",
                content:"urn:oasis:names:tc:emergency:EDXL:DE:1.0"
                description:"Resource Management Requests"
        },
        {
                topic:"/content/rm/response",
                format:"text/xml",
                content:"urn:oasis:names:tc:emergency:EDXL:DE:1.0"
                description:"Resource Management Responses"
        }

    ]
}
```

A structure like this would enable a client to understand what topics are available, what the format of the payload is, what standard the payload is using, and a human readable description of the topic. In theory, a client could publish information about a new topic, before publishing information to the new topic. This would update all other clients that a new topic is available and allow them to subscribe to it to receive new information.

The “topic” field would be a simple string containing the topic. The format field should be limited to the media type for XML and JSON: text/xml and application/json respectively. The content field should either be a publish code list representing the different MACM standards or could simply be the namespace associated to the top-level element in the standard. The advantage to the second option is an external list would not need to be created.

6.1.3 Payloads

As with topics, MQTT payloads are designed to be flexible in nature. The MQTT payload represents the dynamic part of an MQTT message and can be information that can be encoded into bytes, up to 256MB in size total. This is typically in the form of text, and is often JSON, but could be XML, CVS, ASCII, etc. There is control field to indicate what the format of the payload is. This means there needs to be an agreement between the publishers and subscribers on the format of the payload, so in particular a subscriber can digest and understand the payload. For example, if a publisher publishes a payload in XML but the subscriber was expecting JSON, the subscriber will not be able to digest and understand the payload. Additionally, if the publisher publishes Resource Management information, but the subscriber was expecting Situational Awareness information instead, then the subscriber will not able to digest and understand. It is important that these details are spelled out in the MQTT API.

6.1.3.1 Payload Operations

Additionally, there may be times were some information needs to be deleted or cancelled, such as alert, like a shelter in place warning. Unless this information is embedded in the content of the payload, MQTT doesn't natively provide a mechanism to support this type of operation. Either the topic structure will need to account for this type of operation or the payload itself will need to contain the operation. Fortunately, some standards such as CAP and DE already support this type of operation within their data structures. Depending on the standards transmitted in the payload, it may be necessary to define a new payload structure to account for these operations (if needed) or adopt a topic structure that will support these operations. See the Appendix for more discussion on MQTT payload recommendations.

6.2 HTTP

Most modern web-based systems have adopted REST as their architecture. Consequently, the API guidance here will focus on a REST implementation. HTTP messages have two parts: the header and the body. The HEADER contains the HTTP operation, authentication information, media-type, etc., which helps the server determine what do with the client's request. The BODY (if present) contains the shared information. Unlike MQTT, HTTP has a well-known, well-defined set of operations for handle HTTP requests. These operations are generally defined as follows:

- GET – retrieves information from the system
- POST – adds new information to the system
- PUT – wholesale updates (replaces) existing information in the system
- DELETE – removes information from the system
- PATCH – partially updates (updates portions of) existing information in the system

These operations allow for a variety of actions to be taken. A HTTP API should attempt to practice high cohesion, so these operations perform as expected. It is not uncommon for a POST operation to be overloaded so both additions and updates are performed. This should be avoided, as it can add confusion on the intent of the operations.

6.2.1 REST

REST focuses on resources that are available in the system. An example for a mutual aid domain might be alerts. A REST-based API would describe the URL endpoints of that system that would allow for alerts to be created, updated, deleted, retrieved, and patched. These might look something like this, with the HTTP operation in the HEADER ….

GET - https://some.server.com/alerts - retrieves all available alerts

POST - https://some.server.com/alerts - creates a new alert

DELETE - https://some.server.com/alerts - deletes all available alerts

GET - https://some.server.com/alerts/<some alert id> - retrieves a specific alert

PUT - https://some.server.com/alerts/<some alert id> - updates a specific alert

PATCH - https://some.server.com/alerts/<some alert id> - patches a specific alert

DELETE - https://some.server.com/alerts/<some alert id> - deletes a specific alert

Like MQTT's topic structure, REST endpoints can be very flexible, and it can be difficult to organize an API in a meaningful way. There have been several attempts in the past to describe a RESTful API in a machine-readable way, but there has been no consensus on a single approach. Consequently, this makes discovering RESTful endpoints very difficult if not impossible. Unlike MQTT, HTTP clients can not create new endpoints, so the need to allow for discoverability is reduced. Proper API documentation should suffice.

6.2.1.1 DE Distribution Type and HTTP Verbs

The three DE distribution types provide an opportunity to setup a HTTP server in one of two ways. A simplified endpoint structure can be provided that simply supports two HTTP verbs, GET and POST. In this instance, the POST endpoint takes a DE and relies on the Distribution Type to determine how the DE message and content is handled. The alternative is a more RESTful HTTP server that supports GET, POST, PUT, and DELETE, where the POST, PUT, and DELETE endpoints are expected to receive a DE with the corresponding Distribution Type (Report, Update, Cancel).