**Statement of Work (SOW) for Phase One of the**

**Emergency Data Exchange Language (EDXL) Project:**

**Reference Implementation of**

**EDXL - Common Alerting Protocol version 1.2**

**(EDXL-CAP-v1.2) with or without**

**EDXL Distribution Element version 1.0**

**(EDXL-DE-v1.0)**

**OASIS (The Organization for the Advancement of**

**Structured Information Standards)**

**Emergency Management Technical Committee (EMTC)**

**Template:** [**Betterproposals.io**](https://betterproposals.io/template/index.php?ProposalID=bUx5_GprIKml2dXZ6ftCvI-SNMORSmhrpJjj_PfqUlI&cat=sow)

### Introduction

This document specifies activities and deliverables for Phase One (1) of the open source project to develop and deliver a Reference Implementation for the EDXL suite of specifications – the EDXL - Common Alerting Protocol version 1.2 and/or the EDXL - Distribution Element version 1.0.

This project aims to create a modern Emergency Management Framework (EMF) across all jurisdictional entities engaged in the work of emergency management, international and domestic.

For the purpose of this SOW, we take the [Wikipedia definition of Reference Implementation](https://en.wikipedia.org/wiki/Reference_implementation#:~:text=In%20the%20software%20development%20process,and%20corresponding%20customizations%20are%20derived.):

In the [software development process](https://en.wikipedia.org/wiki/Software_development_process), a **reference implementation** (or, less frequently, **sample implementation** or **model implementation**) is the standard from which all other implementations and corresponding customizations are derived. A reference implementation often accompanies a [technical standard](https://en.wikipedia.org/wiki/Technical_standard), and demonstrates what should be considered the "correct" behavior of any other implementation of it.

To be specific about this reference implementation and the EDXL set it represents, we stipulate that it will include every required and optional element from the specification correctly implemented. Additionally, it should include Authentication and Authorization for access control, with a standard API and interface for approval of the EDXL message payload by the administrative entity under which the emergency incident is managed. This should also extend across all EDXL specifications for which this reference implementation stands as a model.

**Brief and Objectives**

The purpose of the SOW is to develop and deliver a Reference Implementation for the [OASIS EDXL Common Alerting Protocol version 1.2](https://docs.oasis-open.org/emergency/cap/v1.2/os/) with and without the [OASIS EDXL Distribution Element version 1.0](https://docs.oasis-open.org/emergency/edxl-de/v1.0/EDXL-DE_Spec_v1.0.doc). It is understood that most/all EDXL messages will be routed according to information specified in EDXL-DE although it is understood CAP does not *require* the DE. We are stipulating EDXL-DE-v1.0 at this time. We may reissue this reference implementation with a different version number in the future.

The scope and tasks required to complete this project will follow the timelines outlined within in order to deliver the components at the associated costs listed.

This project has been proposed to achieve the following objectives:

**OBJECTIVE 1:** Finish the existing project, with adjustments by mutual agreement to be detailed in a separate document signed and dated by OASIS Emergency Management Technical Committee representative and software developer respectively.

**OBJECTIVE 2**: Ensure the software reference implementation is functional yet easy to use so various users may upload content and administrator users can modify, correct and approve individual messages through the appropriate graphical web interface.

**OBJECTIVE 3**: Optimize API for future development as an open source project and ensure that we do not make locked-in technology choices for future software developers, beyond the initial choices made for this project using C#, ASP Net Core Model-View-Controller (MVC) Software Design Pattern-Architecture. Ensure that any secondary software coding languages can be used.

### Project Scope & Tasks

Below is the project scope of EDXL-CAP-v1.2 with or without EDXL-DE-v1.0 as follows:

1. **WEBSITE DESIGN AND DEVELOPMENT The existing** implementation governs the basic design of the user-facing interface for first responder. This design can be changed, but it must follow the structure of the pages in extant project. Additional pages include an Administrator interface that allow for changes, additions, and final approval similar in design to shopping carts and checkout pages currently deployed extensively for e-commerce throughout today’s internet-web.
2. **INFORMATION ARCHITECTURE** This project will use the MVC pattern noted above and specified in greater detail below.
3. **INTELLECTUAL PROPERTY RIGHTS. The OASIS IPR detailed** [here](https://www.oasis-open.org/policies-guidelines/ipr) **and as customized by mutual agreement will apply.**
4. **HOST AND DOMAIN SETUP. OASIS** will specify the hosting and this reference implementation will become a part of the OASIS domain. Any and all software, documentation and other deliverables will be the property of OASIS.

### Deliverables & Components

**DELIVERABLES** (1) The software implementing the Reference Implementation. (2) The documentation describing, the Reference Implementation, including in-line software comments, automated generation of javadoc-like web-based API reference, and user and admin manuals. (3) demonstration of the work to the Reference Implementation Subcommittee.

**COMPONENTS** This Reference Implementation will consist of working software including a web-based graphical interface and associated application programmer interface to enable information in CAP and DE format to be

* entered into a web-based form;
* stored in a standard relational database, and a No SQL document based database;
* visualized on a map, table, or text web format;
* with automatic validation and conformance with CAP and DE specifications;
* with CAP and DE information representation in JSON and XML formats;
* with a RESTful API for creation, retrieval, update, deletion, validation and search of CAP and DE messages;
* with basic security features for authentication and authorization.

An example of a limited Reference Implementation for CAP is the combination of the google cap-library <https://github.com/google/cap-library> with the as graphical interface <http://cap-validator.appspot.com/validate#r> . Note that this example is for reference only, and it is not a complete Reference Implementation, being limited to CAP and is missing some key abilities, such as security, the ability to enter CAP messages in a web form and to support messages in JSON format.

An example of a graphical user interface for CAP data through the International Filtered Alert Hub can be found at <https://alert-hub.s3.amazonaws.com/cap-feeds.html> access to the free, open-source software for this feature can be found at <https://s3-eu-west-1.amazonaws.com/alert-hub/index.html>

### Assumptions

Previous work on a reference implementation for EDXL has been performed using C#, MS MVC, and a relational database, including the development of JSON schemas for CAP and DE, web forms and storage capabilities. The performer shall assess and leverage this work to the extent feasible.

The following list touches on some possible but not currently feasible assumptions and this should be taken into account. Details that come out of this list during consultation may be added or deleted.

* Technological limitations (coding languages and platforms discussed above)
	+ TBD
* Industry standards software dev should stick to
	+ MS ASP NET CORE MVC
	+ MS Template for Authentication and Authorization (Identity Management)
* CI/CD pipeline diagram. May not be included except as a reference to design principles of DevOps
* Information about the testing of the product, involved parties, and necessary hardware or software. TBD
	+ List of accepted devices, screen resolutions, and browsers for the testing process.
* Means and tools for communication between the client and the outsourcing vendor. May include, but not restricted to Meeting Services such as AdobeConnect, Zoom, etc.
* Procedures for minor and major order changes. TBD
* Penalties for late deliveries and bonuses for extra labor.

### Deadlines, Schedule & Budget

**Monthly or Bi-weekly Meetings** by mutual arrangement will include, but not necessarily be restricted to Developer, OASIS EM TC Representative(s)

**Schedule**: The work will be performed starting as soon as possible but not later than 1 September 2020 and completed with all deliverables provided on or by 1 December 2020.

**Cost**: to be paid upon completion of work and receipt of approval of deliverables.