**Track 3 Prompt: Hack for the Ocean – A DevOps Challenge**

**Scenario**

It’s August 30, 2005 and you’re in a Super Stallion, CH-35 helicopter taking off from the USS IWO JIMA (LDH 8), and amphibious assault helicopter carrier deployed as the first responder to support the many people in need on the gulf coast. You touch down in the Navy’s command and control center within a disaster stricken New Orleans. Hurricane Katrina made landfall yesterday and is in the process of becoming the deadliest and costliest hurricane in US history. Major streets are flooded, windows are blown out of almost every building, entire highway overpasses are underwater. Your task force has been given the responsibility to deploy the infrastructure to coordinate the Navy’s disaster relief efforts with civilians in need and independent NGOs.

As it stands, the Navy has stood up a cloud environment, but it is up to you to build an application that by deploying services within the environment and connecting these services as a cohesive application. Important data will be streaming through these services and the first responders are depending on you. See Participant Read Ahead for more information on how to access the environment and to learn more about the environment itself.

Figure 1. USNS MERCY Delivering relief supplies in support of hurricane Katrina victims.

**Requirements and Submission Guidelines**

There are three mandatory requirements for this application. Your team is required to work within the Navy environment that has been stood up for this scenario. You must package and deploy the application using containers (i.e. Docker, Kubernetes, etc.). You must use Git throughout to build and update your application.

The deadline for submissions is at 12:30 (Noon 30) on Sunday. To be in the running to win the competition, teams will be expected to submit a completed application, a short written overview of the application, a diagram of the services and how they connect, and give a 4 minute pitch to the judging panel on the application you have deployed.

**Application Components**

Services you have available to build the application fall within the categories listed below. More technical information on how to build the application and the data sets available for integration are available in the subsequent sections.

* Data Pipeline Framework
* Data Processing and Analytics Framework
* Data Representation & Persistence Framework
* Data Visualization Framework
* Geospatial Data Access Layer & GIS Services
* Ingress and Egress Dataflow Management Framework
* Marketplace
* Mediation Framework
* Platform as a Service (PaaS)
* Security Management

For information on scoring see the Scoring Rubric and Judging Scorecard.

General Guideline:

You should plan to bring your own device to the challenge and ensure it is set up optimally ahead of time. Participants can develop the app in whatever modern languages and frameworks they see fit. Recommendations are GoLang, Python, Javascript, node.js, React, and Ruby, but depending on approach other languages or frameworks are acceptable.

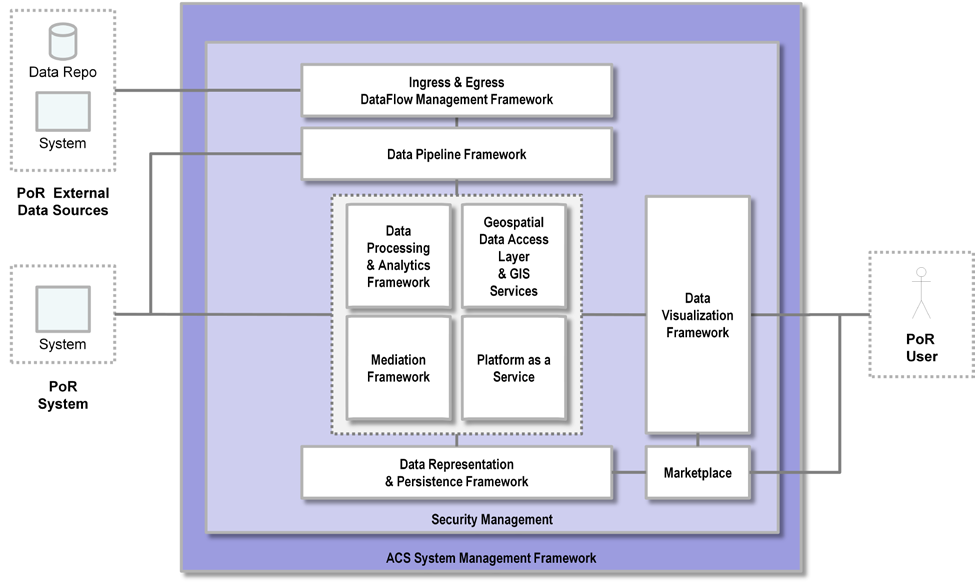
Participants should plan to develop and deploy their application as containers. This means that teams will use Dockerfiles, Docker Compose files, Kublet files, or a similar alternative to deploy their application as containers on either RedHat Openshift and/or Docker EE. Participants must also use a GIT based version control system and container registry as a DevOps pipeline into the competition environment.

How to get Started:

This is the section that needs the most work

Agile Core Services Overview

Appropriate SMEs: Douglas Rapp, Jason Wong, Jeff Millard PhD

ACS provides a collection of Computer Software Configuration Items (CSCIs) to support data analytic and common service functionality as diagrammed below.

Participants can use any of the services within the environment, however, the services listed below are the services that teams should plan to prioritize deploying. More information about these services within the context of the ACS environment is available in the ACS Developer Guide.

* Data Pipeline Framework:

Scalable and reliable data pipeline enabling access and transfer of data records between large numbers of data producers and consumers. The data pipeline enables a high throughput of data organized by topic (i.e. a stream of related records)

* + Apache Kafka
  + Apache Zookeeper
* Data Processing and Analytics Framework:

Enables rapid and efficient big data (in volume, velocity, and variety) processing analytics, and queries across a shared data repository

* + Apache Spark
  + Apache Storm
  + Intellex
* Data Representation & Persistence Framework:

Persist and query large volumes of data in an efficient, reliable and secure means. Provides a clustered data store supporting Big Data needs as well as traditional Relational Database management Systems storage

* + Apache Accumulo (on HDFS)
  + DRIF
  + Enterprise DB PostgreSQL
  + GeoMesa
  + Rya
  + Structured Objects
* Data Visualization Framework:

Mechanism to host, organize, access, and share Widgets

* + Ozone Widget Framework
* Geospatial Data Access Layer & GIS Services:

Provides common geospatial data and base maps as well as the ability to store, process, and analyze geographic data

* + CJMTK (ArcGIS Server, AGIS Portal, GeoEvent Processor)
* Ingress and Egress Dataflow Management Framework:

Data routing, transformation, and mediation logic for automation and management of data flows in and out of ACS

* + Apache NiFi
* Platform as a Service (PaaS):

Hosting and management of server-side applications for use leveraging virtualization to maximize resource utilization, providing necessary isolation of hosted applications to minimize required administration and maintenance

* + Red Hat JBoss Enterprise Application Platform
  + Red Hat OpenShift Container Platform
* Security Management:

Centralized services for use by POR applications to authenticate and authorize users and systems

* + OpenAM

Data Sets Available for Integration

Appropriate SMEs: Cameron Kruse

Four data sets are available to teams. These data sets are as follows:

* Streaming calls for service throughout New Orleans:
* Streaming first responder locations (synthetically generated)
* Air traffic data
* AIS Data around New Orleans
* Structural damage data (going to MongoDB)

Here I will add more information about how to access and what can be done

General overview of services and SMEs available to help