



Building to the DM-OPEN Non-Weather Emergency Message Interface

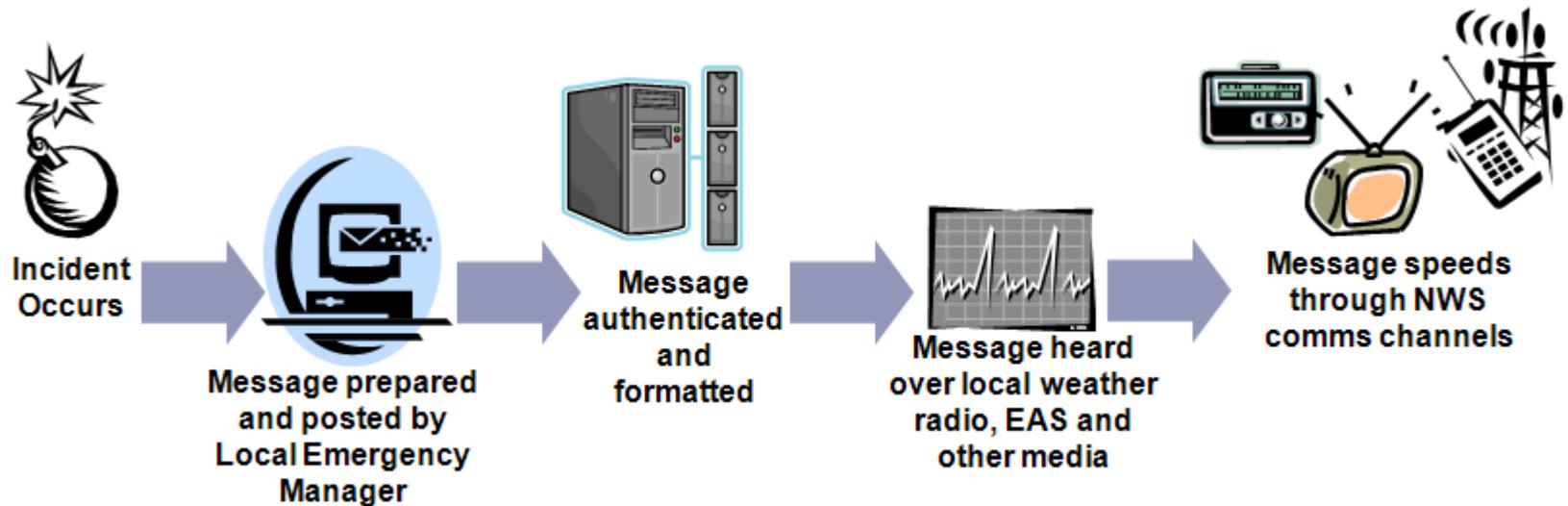
DM-Open SIG May 20, 2009

All-Hazards Emergency Message Collection System

HAZCollect



- Is a nationwide capability developed by the National Weather Service (NWS) in coordination and cooperation with the U.S. Department of Homeland Security (DHS)
- Offers emergency services officials full distribution of critical emergency messages through NWS channels
- The same system that speeds life-saving weather data to people can be used to inform the public about critical local events when lives are at stake
- In an incident an emergency management agency (EMA) issues a message on the impact and recommended action
- The message goes to DM-OPEN servers where it is authenticated and protocol checks are performed
- **HAZCollect** servers send the message through all NWS distribution channels
- The message is simultaneously sent over EMWIN, NOAA Weather Wire Service, Family of Services and ultimately through private vendor communication channels.
- **HAZCollect** achieves the widest possible distribution of critical non-weather emergency information via NWS systems



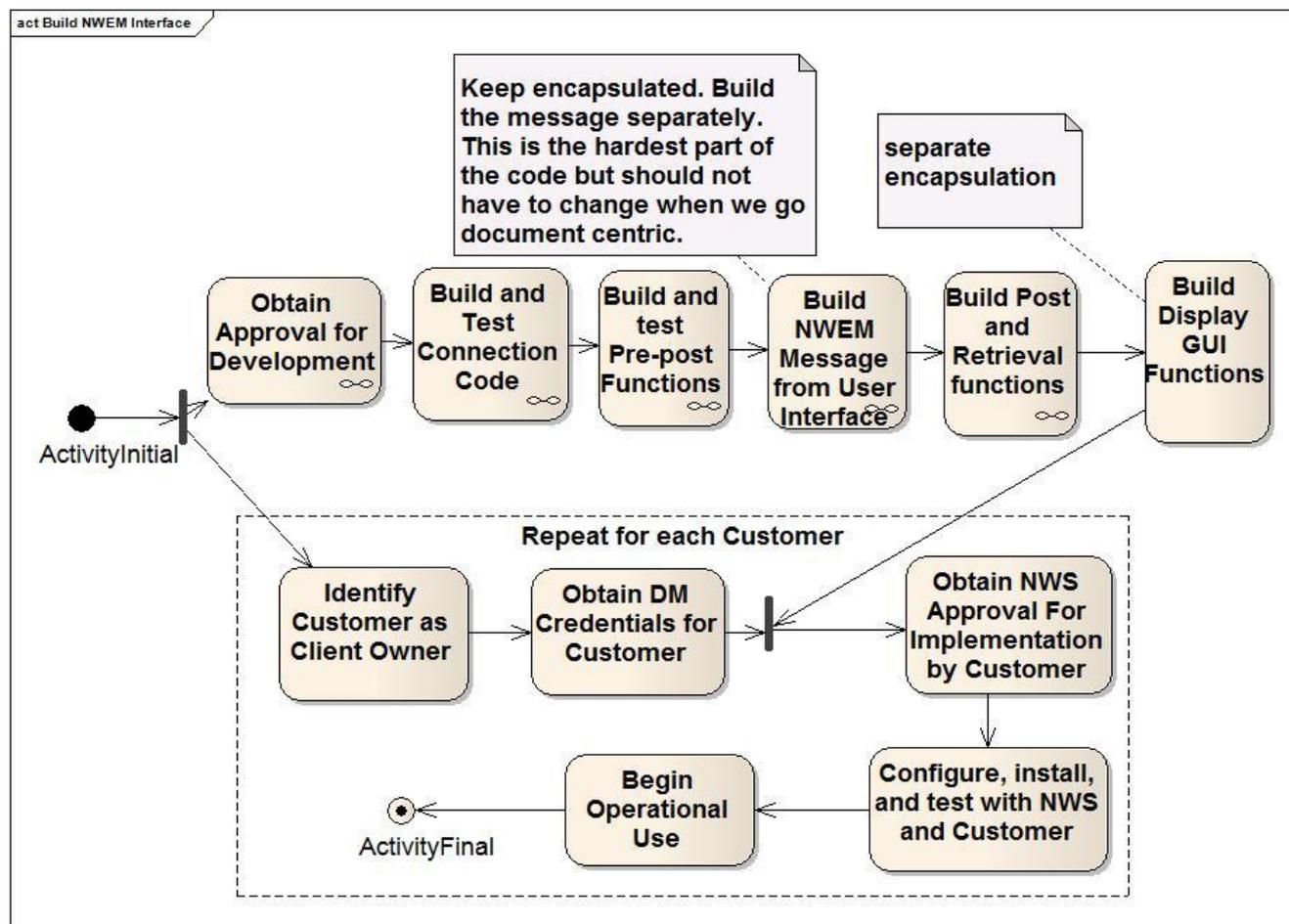
Typical User Post Activity



- Log in
- Determine that login was successful (DM-OPEN is ready to receive messages)
- Confirm that the user is allowed to send HazCollect messages
- Determine the state of the HazCollect Server
- Confirm or generate the allowed FIPS Codes that are authorized for the user
- Build the message
- Post the message to HazCollect and additional COGs
- Confirm that the message was posted successfully

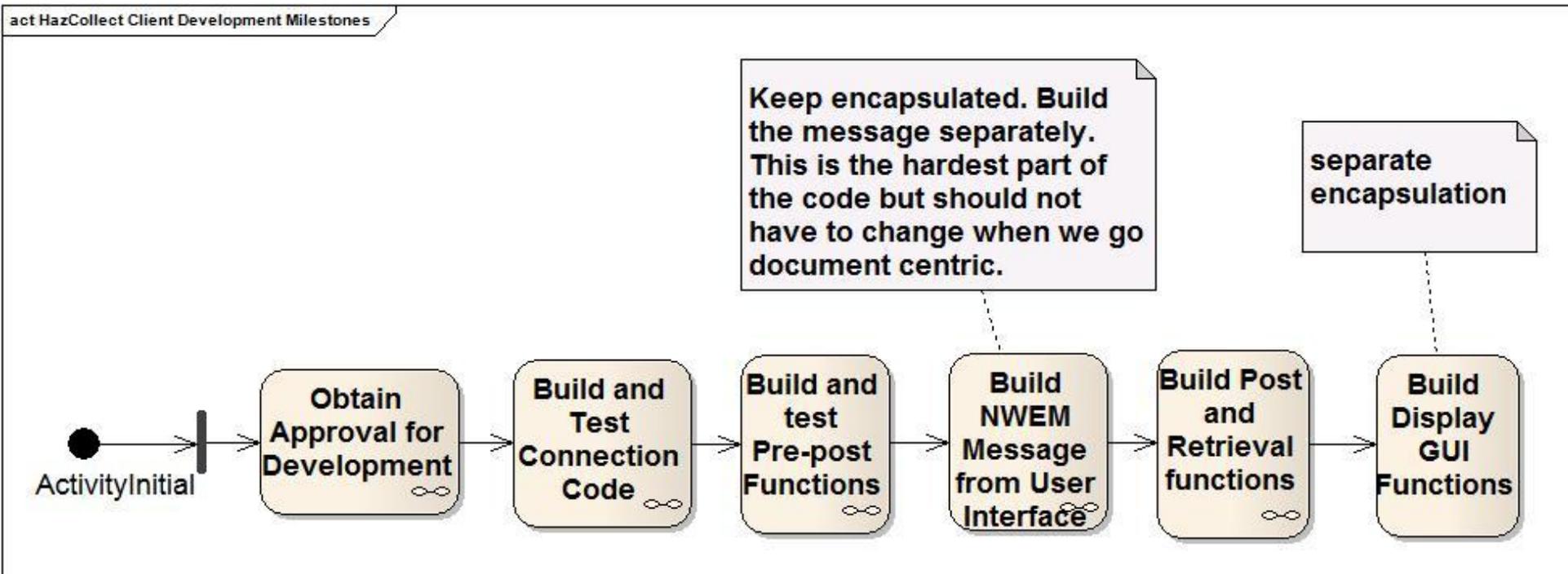
Steps to NWEM Implementation

- **Six project development milestones**
- **Five steps to repeat for each Customer**



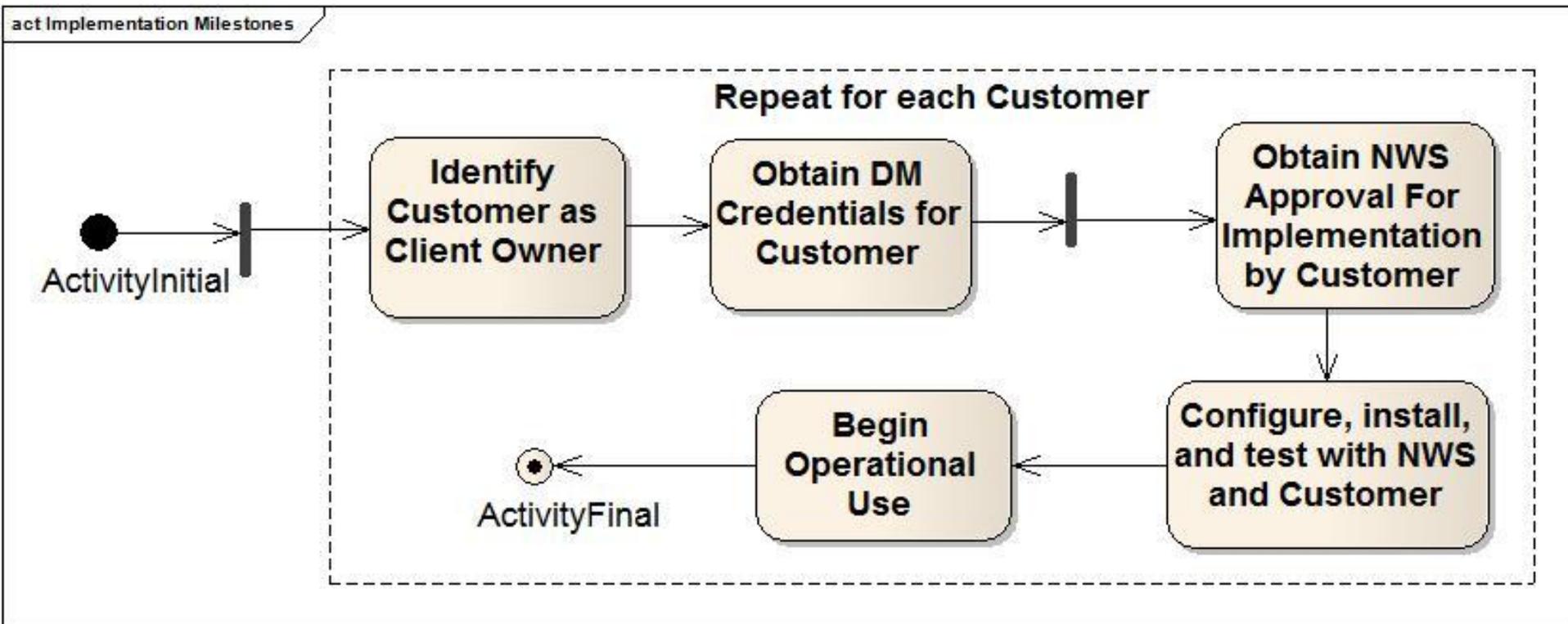
Development Milestones

- An encapsulated set of code blocks.
- Facilitates conversion down the road.
- Each step builds on the other.



Implementation Milestones

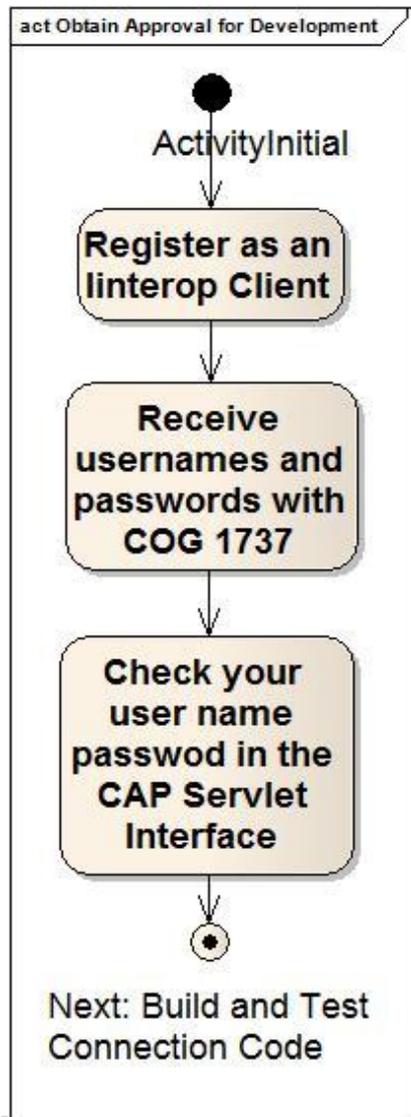
- Each Customer must have a DM COG assigned.
- Each COG must be Weather Service approved for particular FIPS
- The Path goes through DM-OPEN, but approval of users is a Weather Service function.



Development Milestone - Obtain Approval For Development



DISASTER MANAGEMENT
INTEROPERABILITY SERVICES



- Register
 - <http://disasterhelp.gov/disastermanagement/open>
 - Select “Request” OPEN access
 - Fill out the form
- You will get membership in the “interoperability COG.”
- Check your Credentials by using the following string in the servlet interface substituting your user ID and password (all one line):

https://interop.cmiservices.org/axis/servlet/CAP?version=1.1&user=operator1&pass=Operator_1&co g=1737&op=getCAPAlert&fromDate=2009-05-01T12:44:43-04:00

You should get a big long string of xml alerts. It will contain all alerts posted or received by cog 1737 after the time you use in the string. If you get results your credentials are good.

Development Milestone: Build and Test Connection Code

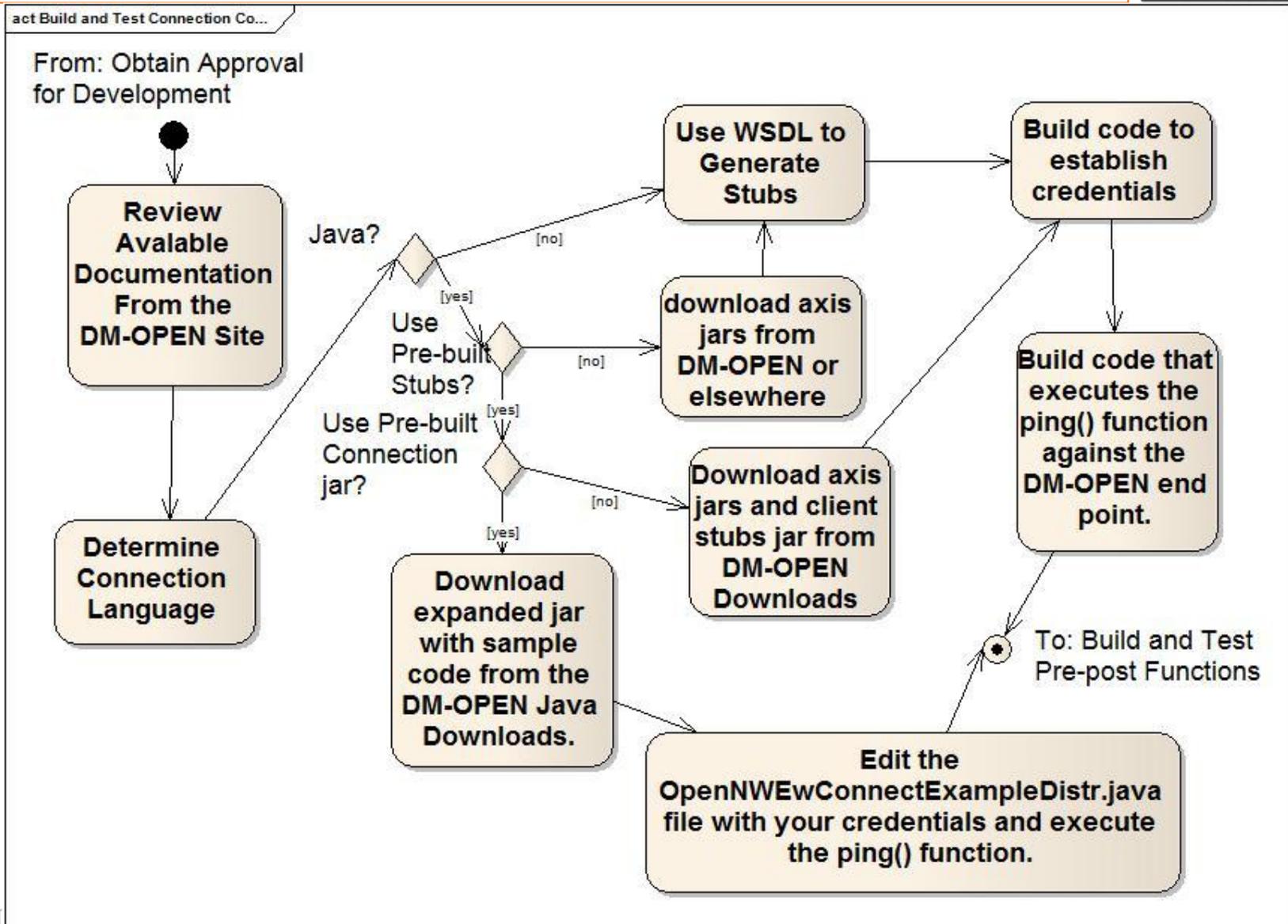


- The WSDL:
 - Development: <http://interopdev.cmiservices.org/axis/services/NWEM?wsdl>
 - Production: <https://interop.cmiservices.org/axis/services/NWEM?wsdl>
 - They are actually the same except for the endpoint.
- The Downloads Location
 - <http://www.disasterhelp.gov/disastermanagement/open/downloads.shtml>
 - NWEM Programmers manual
 - Example in Java but the concepts apply in all languages
 - Section Seven and Appendix A are valid in all languages
 - There is one change to Appendix A – multiple geocodes in an Area are allowed
 - More later
 - Java Downloads - Jar with stubs and axis Jars included
 - Java Downloads – Jar with above plus sample code and OpenNWEMConnection Class that makes encapsulation of Document building from Object building easier

Development Milestone: Build and Test Connection Code



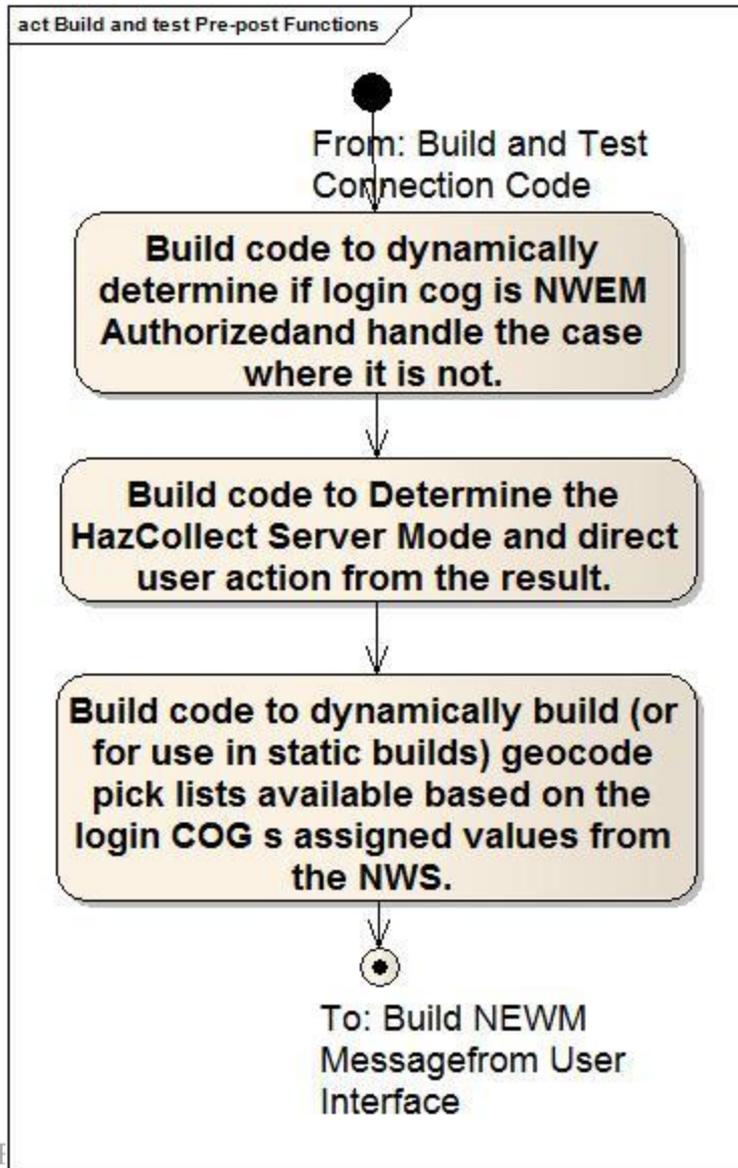
DISASTER MANAGEMENT
INTEROPERABILITY SERVICES



Development Milestone: Build and Test Pre-post Functions



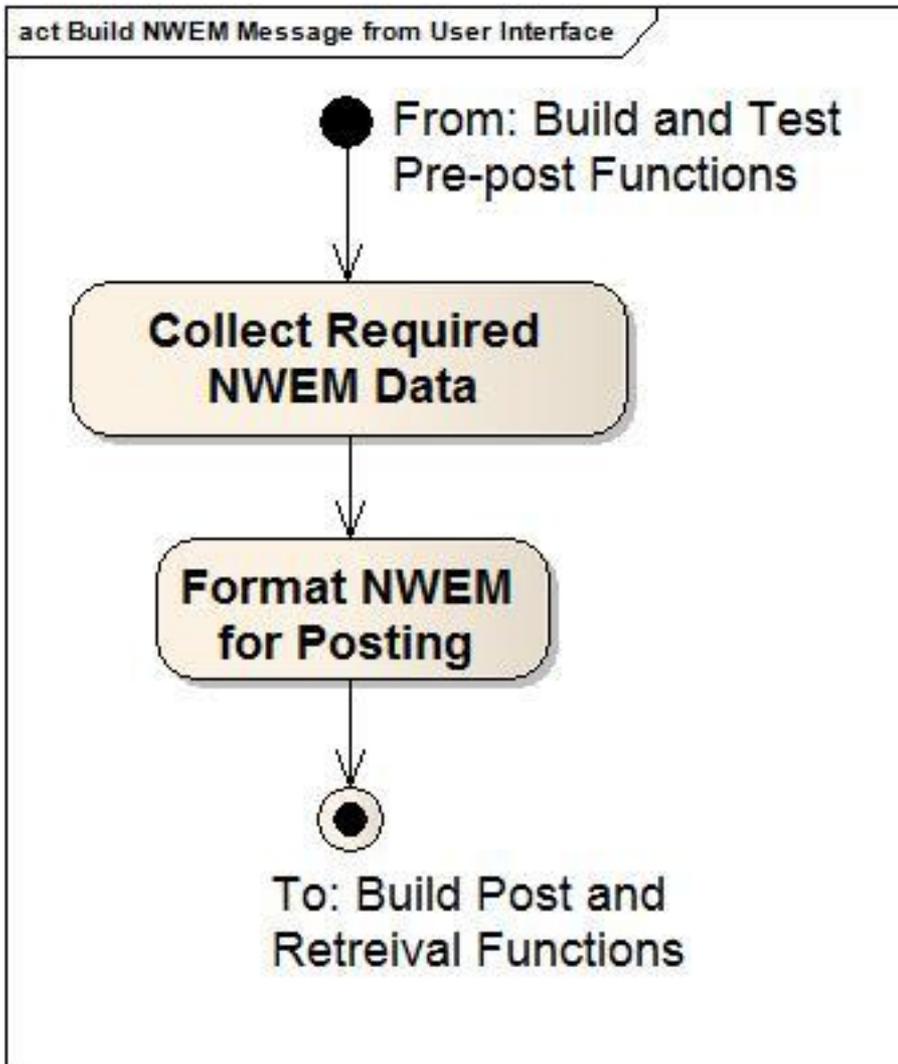
DISASTER MANAGEMENT
INTEROPERABILITY SERVICES



- The HazCollect isCogAuthorized() function will let you know if the COG has been authorized by the NWS for use. They can change it dynamically, so it is a good idea to check.
- The GetHCSMode() function lets you know the HazCollect server Status. It must be “Active” for an actual broadcast to go out. See Appendix B to the Programmer’s manual for details on HCSMode and what it means for your messages.
- The getNWEMAuxData() function brings back the list of FIPS code that have been authorized for you to use based upon your COG. You use this function to dynamically (or statically with periodic refresh) build a pick list of localities that you can use in your NWEM.



Development Milestone: Build NWEM Message from User Interface

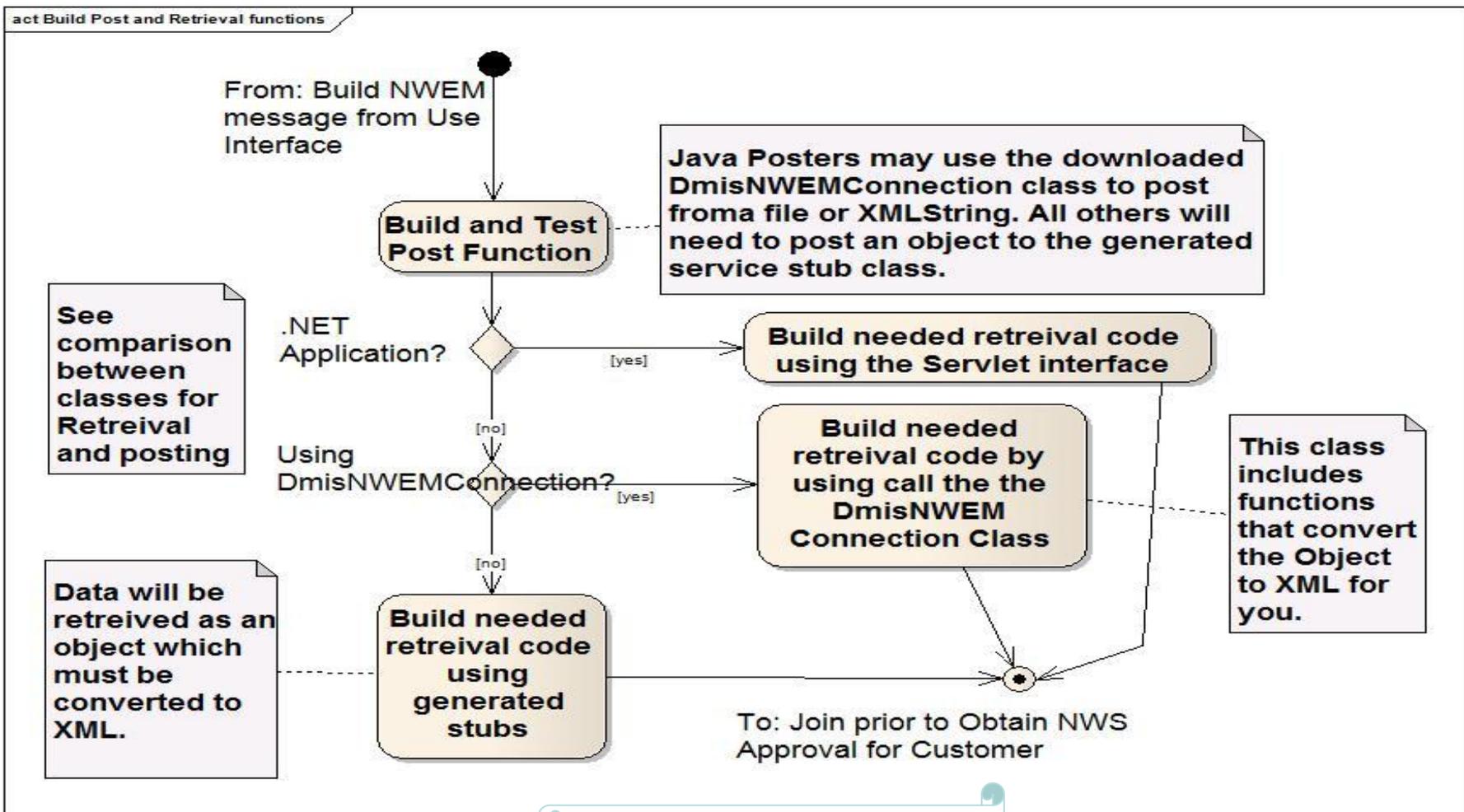


- Specific Data requirements are found in Appendix A to the Draft programmer's Manual.
- Programmers who use the downloaded OpenNWEMConnection Class have the option of posting from a file, an XML String, or a DMIS CAP Object. Others must post the object as defined from the WSDL. The former may be advantageous as we transfer to a document-literal interface later this year.

Development Milestone: Build Post and Retrieval Functions



DISASTER MANAGEMENT
INTEROPERABILITY SERVICES



Two ways to post.
Three ways to retrieve.

Class Differences: Generated Stub vs. Connection Class



class System

```

classDiagram
    class OpenNWEMConnection {
        ~ cogId: String
        ~ dmiURL: String
        ~ nwem: NWEMSoapBindingStub
        ~ password: String
        ~ service: NWEMService
        ~ svcURL: URL
        ~ userId: String
        + getAuxDataXML() : String
        + getHCSServerMode() : String
        + getMyCog() : SimpleCOG
        + getMyCogXML() : String
        + getNwem(String, String, boolean) : String
        + getNwemsXdays(int, boolean) : ArrayList
        + isConnectionCogNwemAuthorized() : boolean
        + OpenNWEMConnection(String, String, String, String) : void
        + ping() : boolean
        + postNwemAlertToOpen(Alert, String, ArrayList) : boolean
        + postNwemFileToOpen(String, String, ArrayList) : boolean
        + postNwemXmlStringToOpen(String, String, ArrayList) : boolean
    }
    class NWEMSoapBindingStub {
        - cachedDeserFactories: java.util.Vector = new java.util.V...
        - cachedSerClasses: java.util.Vector = new java.util.V...
        - cachedSerFactories: java.util.Vector = new java.util.V...
        - cachedSerQNames: java.util.Vector = new java.util.V...
        - _initOperationDesc1() : void
        # createCall() : org.apache.axis.client.Call
        + getAlert(java.lang.String) : org.cmis.interopserver.beans.cap1_1.Alert
        + getAlertsByPostedDate(java.util.Calendar) : org.cmis.interopserver.beans.cap1_1.Alert[]
        + getAlertsBySentDate(java.util.Calendar) : org.cmis.interopserver.beans.cap1_1.Alert[]
        + getHCSServerMode() : java.lang.String
        + getMyCog() : org.cmis.interopserver.beans.SimpleCOG
        + getNwemAuxData(java.lang.String) : gov.noaa.nws.hazcollect.support.view.CogGeoFipsCodeImpl[]
        + getServerTime() : java.util.Calendar
        + isCogAuthorized(java.lang.String) : boolean
        + NWEMSoapBindingStub()
        + NWEMSoapBindingStub(java.net.URL, javax.xml.rpc.Service)
        + NWEMSoapBindingStub(javax.xml.rpc.Service)
        + ping() : java.lang.String
        + postNWEM(org.cmis.interopserver.beans.cap1_1.Alert, java.lang.String, org.cmis.interopserver.beans.SimpleCOG[]) : void
    }
    OpenNWEMConnection --> NWEMSoapBindingStub : +nwem
    
```

```

classDiagram
    class NWEMSoapBindingStub {
        org.apache.axis.client.Stub
        org.cmis.interopserver.service.nwem.NWEM
        - cachedDeserFactories: java.util.Vector = new java.util.V...
        - cachedSerClasses: java.util.Vector = new java.util.V...
        - cachedSerFactories: java.util.Vector = new java.util.V...
        - cachedSerQNames: java.util.Vector = new java.util.V...
        - _initOperationDesc1() : void
        # createCall() : org.apache.axis.client.Call
        + getAlert(java.lang.String) : org.cmis.interopserver.beans.cap1_1.Alert
        + getAlertsByPostedDate(java.util.Calendar) : org.cmis.interopserver.beans.cap1_1.Alert[]
        + getAlertsBySentDate(java.util.Calendar) : org.cmis.interopserver.beans.cap1_1.Alert[]
        + getHCSServerMode() : java.lang.String
        + getMyCog() : org.cmis.interopserver.beans.SimpleCOG
        + getNwemAuxData(java.lang.String) : gov.noaa.nws.hazcollect.support.view.CogGeoFipsCodeImpl[]
        + getServerTime() : java.util.Calendar
        + isCogAuthorized(java.lang.String) : boolean
        + NWEMSoapBindingStub()
        + NWEMSoapBindingStub(java.net.URL, javax.xml.rpc.Service)
        + NWEMSoapBindingStub(javax.xml.rpc.Service)
        + ping() : java.lang.String
        + postNWEM(org.cmis.interopserver.beans.cap1_1.Alert, java.lang.String, org.cmis.interopserver.beans.SimpleCOG[]) : void
    }
    
```

Function	OpenNWEMConnection	NWEMSoapBindingStub
Establish connection	instantiate class	instantiate Java connection
ping()	Returns: Boolean	Returns: String
getAuxData()	Returns: String of XML	Returns object array: CogGeoFipsImpl []
getNwems()	Returns: ArrayList of XML Strings	Returns object array: org.cmis.interopserver.beans.cap1_1.Alert []
postNwem	May post: Alert Object, File name, or XML String	Must post object: org.cmis.interopserver.beans.cap1_1.Alert

Alternative Retrieval: The Servlet



- The Servlets:
 - Development:
http://interopdev.cmiservices.org/axis/servlet/CAP?version=1.1&user=operator1&pass=Operator_1&cog=1737&op=getCAPAlert&fromDate=2009-05-13T12:44:43-04:00
 - Production:
https://interop.cmiservices.org/axis/servlet/CAP?version=1.1&user=operator1&pass=Operator_1&cog=1737&op=getCAPAlert&fromDate=2009-05-13T12:44:43-04:00
- The servlet operation is a work around to retrieve CAP messages (including NWEM CAP) until the new document-literal service is in place

Development Milestones: Build Display and GUI Capabilities



- Look and feel is your business.
- Use of retrieved data is also your business.
- You then sell to NWS authorized customers and/or help your customers obtain approval from the NWS to send Non-weather Emergency messages.
- Let us know if you have questions? (gary.ham@eyestreet.com)

Potential Presentation Opportunities



- NEMA - Columbus, OH (October 10-13)
- IAEM – Orlando, FL (October 31 – November 4)
- Others? – If you build to the NWEM we will support you as you prototype.

Questions?

