Raytheon

Customer Success Is Our Mission

AWIPS Continuous Technology Refresh (CTR)

AWIPS Software CTR

ADE System Architecture

February 12, 2009

This document includes data that shall not be duplicated, used, or disclosed – in whole or in part – outside the Government for any purpose other than to the extent provided in contract DG133W-05-CQ-1067. However, the Government shall have the right to duplicate, use, or disclose the data to the extent provided in the contract. This restriction does not limit the Government's right to use information contained in this data if it is obtained from another source without restriction. The data subject to this restriction are contained in all sheets.

AWIPS-II: Architecture Topics

- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

System Concept: AWIPS-II Architecture
Data Fusion and Visualization SOA Framework

- Requirements Vision Drives Architecture
  - Focus on “ilities” drives new architecture
  - Features and capabilities get generalized into reusable patterns
  - Customer TIMs give priority to capabilities
- Architecture Framework Vision:
  - Create a new low cost framework for hosting a full range of environmental services including thick client visualization
  - The framework will scale down to a small laptop and up to clusters of enterprise servers without software change
  - The framework will be based on highly reusable design patterns that maximize reuse, has datatype independence, and fast adaptability
  - Open source is leveraged to maximize reuse

AWIPS-II: Architecture Definitions

- **ADE:** AWIPS Development Environment, source code to execution framework enterprise development kit including tools
- **SOA**, End points, I/O Routing, Transforms: Service Orientated Architecture where system capability is available at ***stateless*** endpoints
- **Canonical XML:** Well formed XML that follows high level rules
- **Patterns:** Implements a design solution that solves a problem that occurs many times
- **Technical Reference Architecture:** A physical software execution framework
- **JMS, JMX:** Java Messaging System (API), Java Management Extensions
- **CAVE:** Common AWIPS Visualization Environment using Eclipse RCP
- **SEDA:** Serial Event Driven Architecture

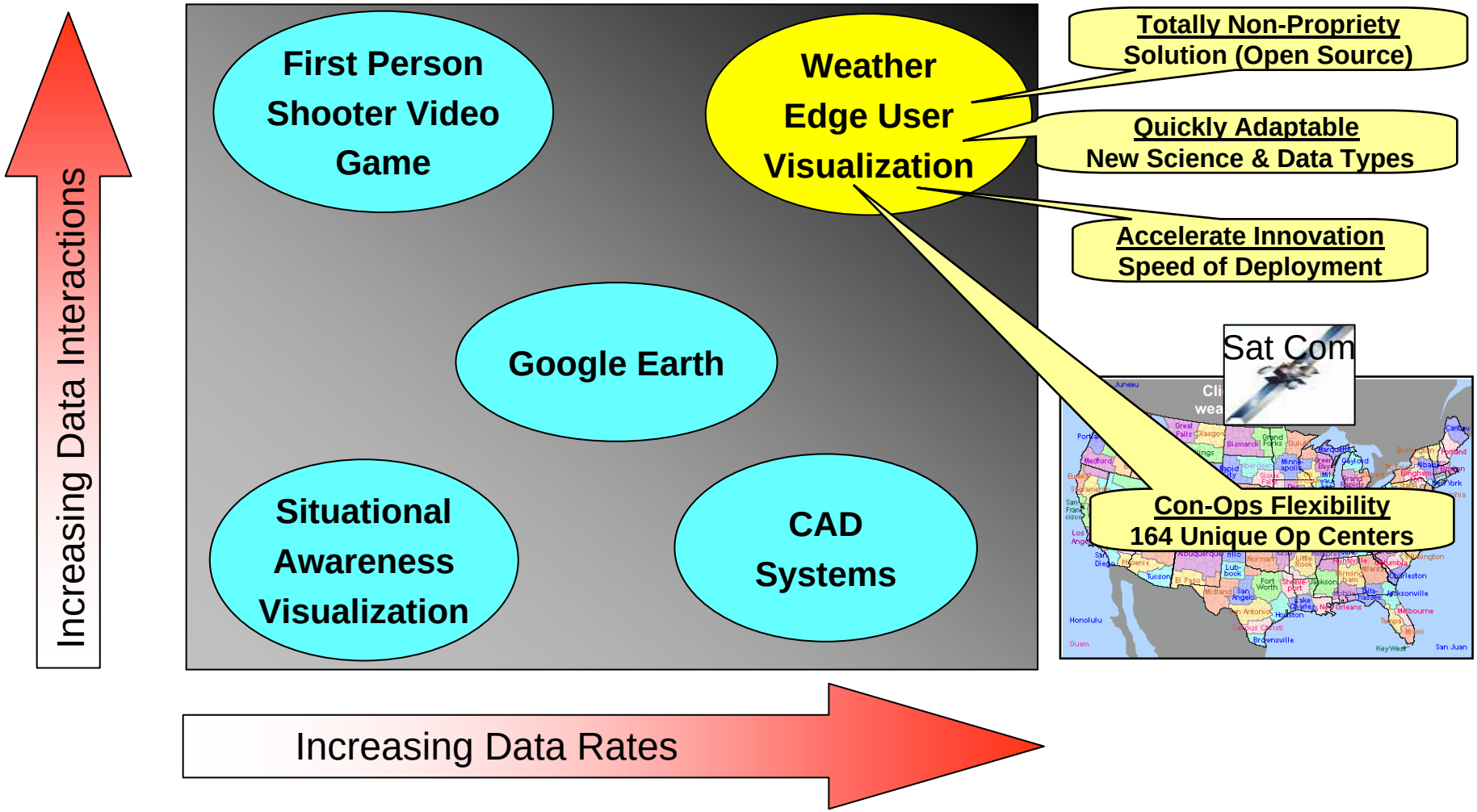
AWIPS-II Architecture Concept –
Architecture Framework Implementation

- **Framework Implementation:** Integrated several best of breed open source projects with a set of advanced enterprise **patterns** to create a highly extendable framework.
  - **Patterns** Implemented in Pure **Java** code (Reuse Example: ProductSrv and AutoBldSrv use uEngine)
  - **Open Source** is primary for re-use:
    - ~15 major open source projects integrated
    - Version controlled with CM baseline, libraries part of run env.
    - Leverage Internet community for core infrastructure
    - Standards compliant, rapid evolution
    - Free, large body of public expertise
    - Open source libraries controlled by putting them in the CM compile library and deploying them to the runtime environment
  - **Packaged** together into the ADE which contains everything from the Source Code repository to the execution environment including operator Clients

AWIPS-II: Problem Definition & User Needs Drives Architecture Towards Advanced Solutions

Raytheon

Architecture Introduction



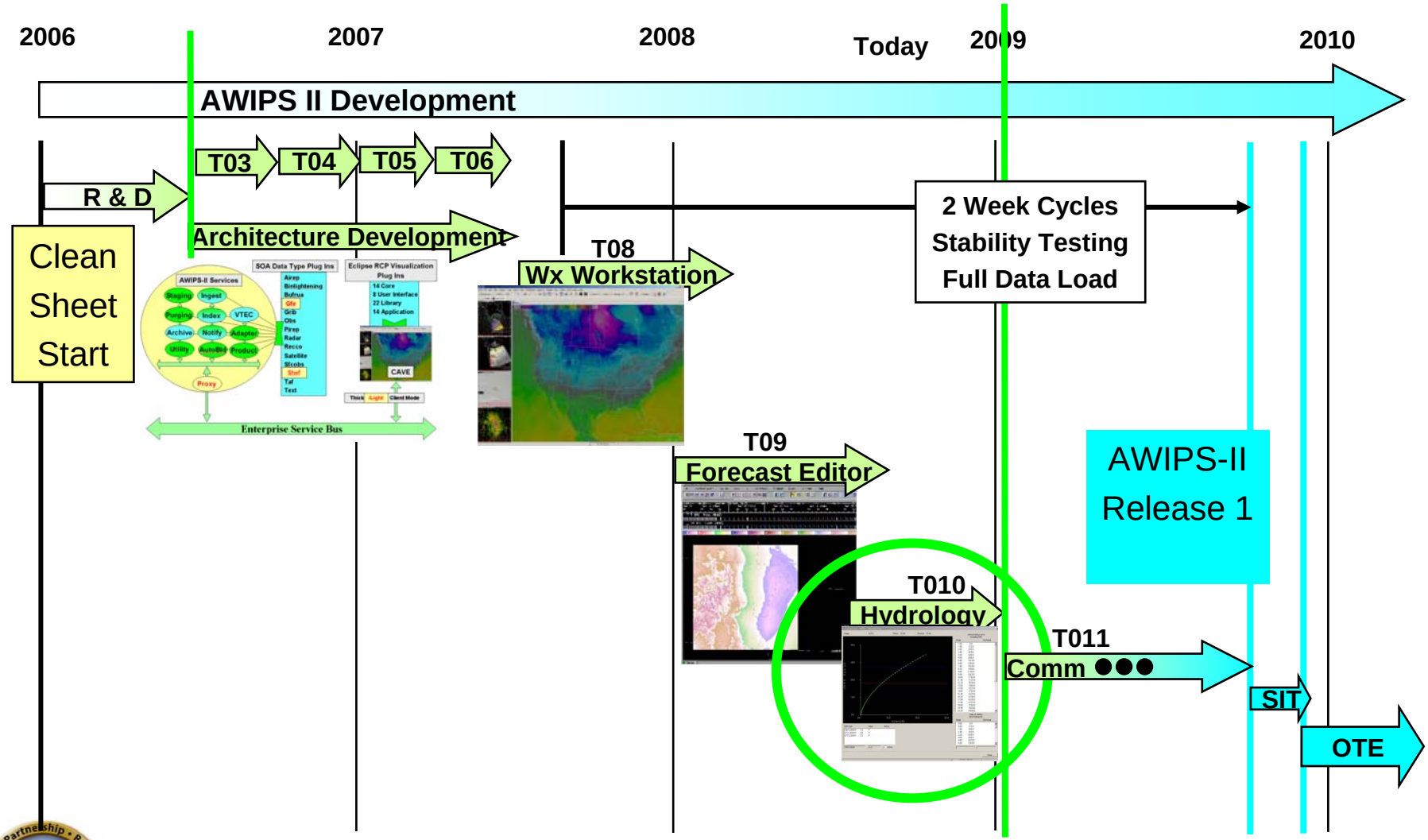
High Data Rates With Gaming Style Interaction

AWIPS-II: Development Sequence, T09 is GFE+

3 to 6 month fixed price task orders

Raytheon

Architecture Introduction



Spiral Development with Interim Deliveries

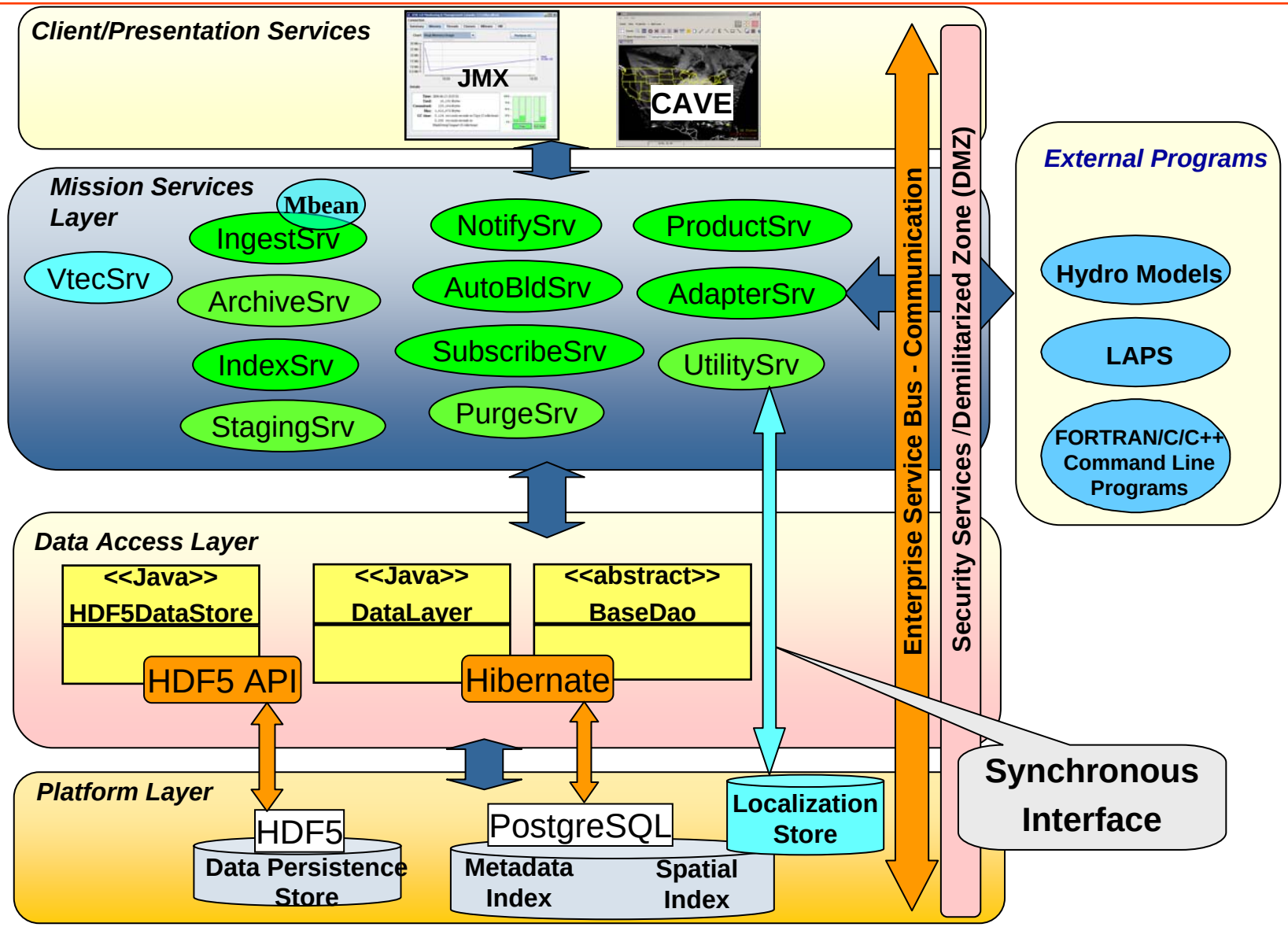
AWIPS-II: Architecture Topics

- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

SOA Architecture – Logical Layered View
Layers Separated By Simple APIs

Raytheon

Architecture Overview



AWIPS-II: ESB SOA Framework

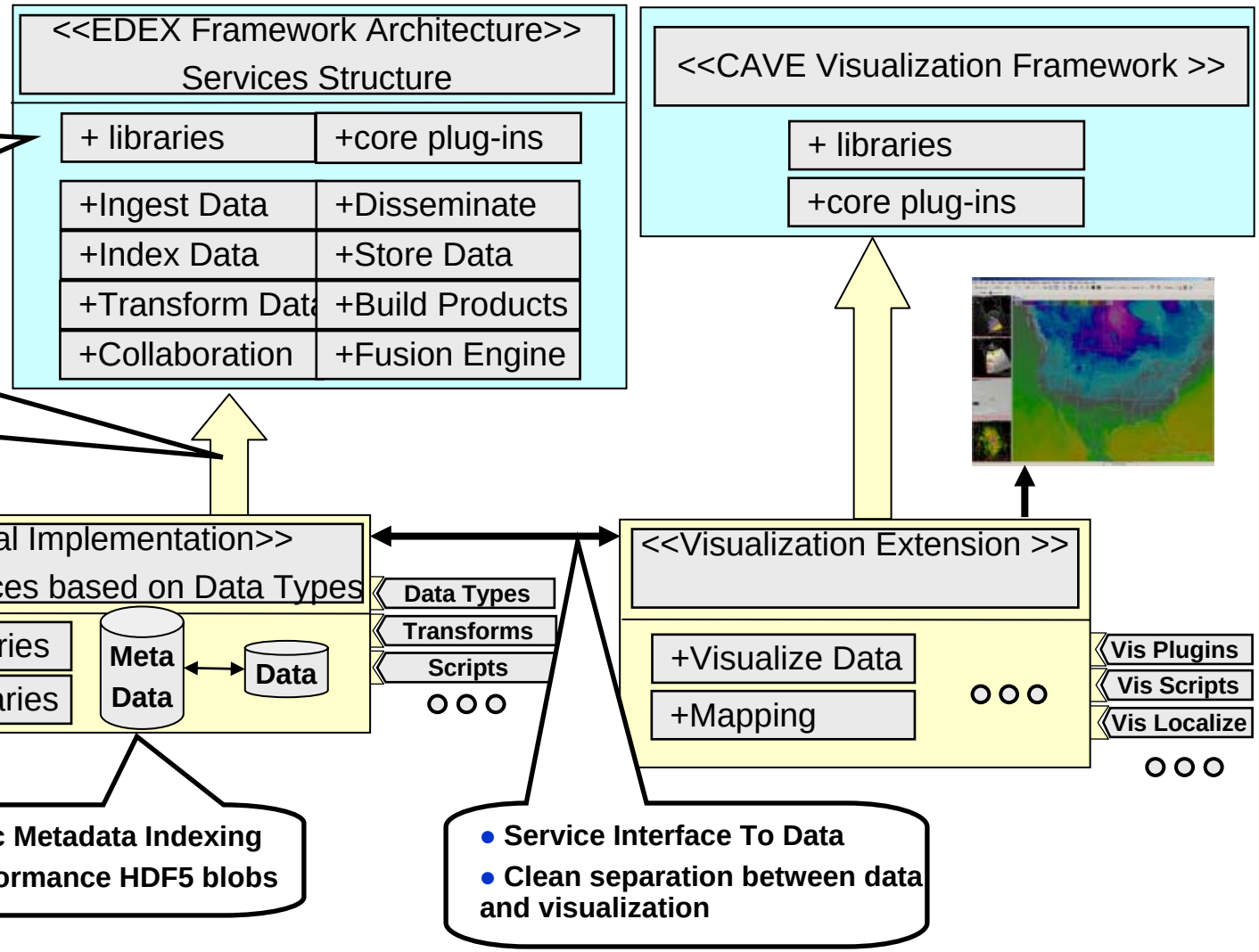
Generic Core Services Adapt Through Plug Ins

Raytheon

Architecture Overview

- **Core Base of Services**
- **Micro Engine**
- **Plug-in Framework**
- **Core libraries**

- **Extend to a specific set of services**
- **Plug in specific libraries**
- **Plug in data types, transforms & scripts**



- **Automatic Metadata Indexing**
- **High Performance HDF5 blobs**

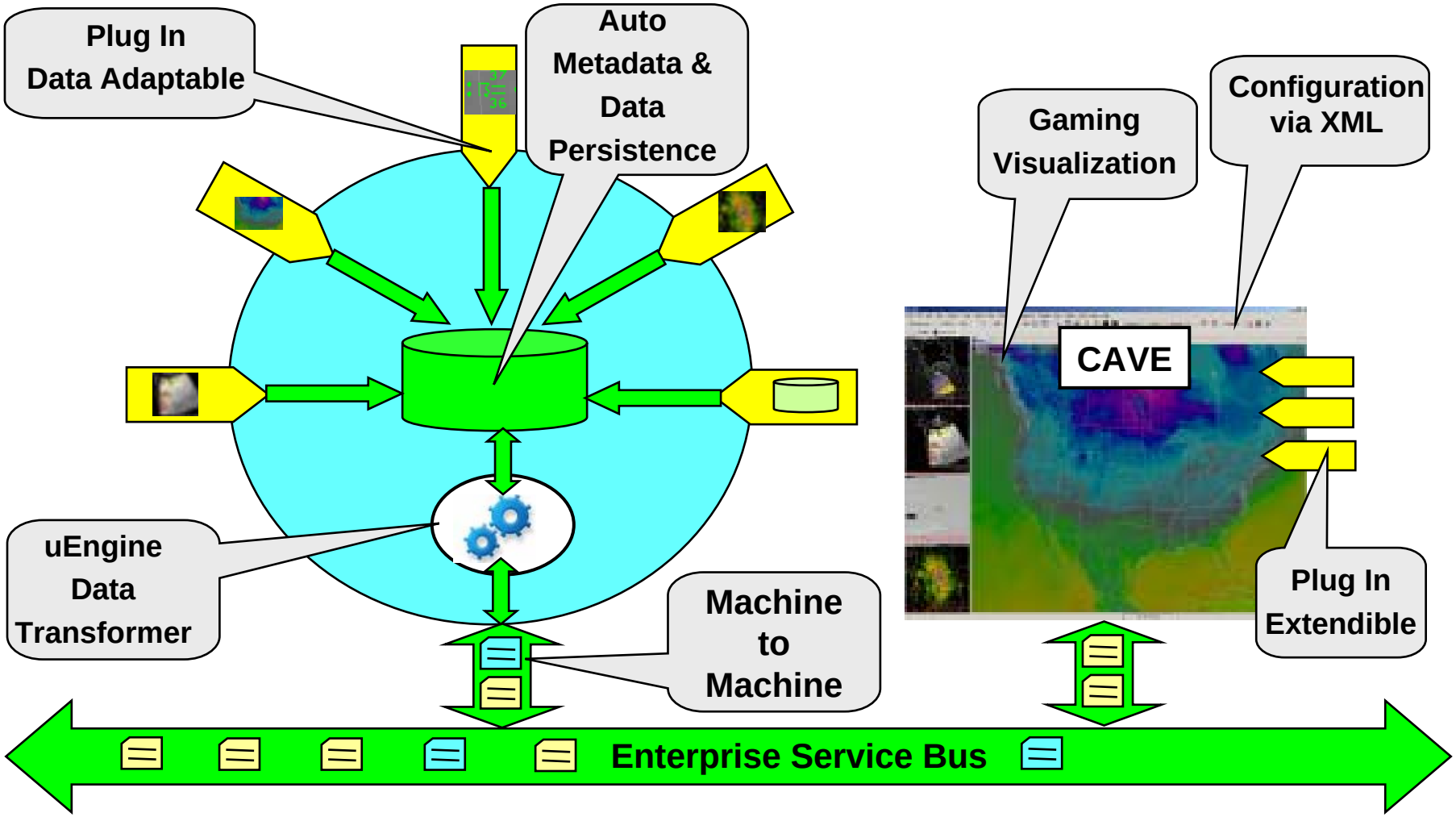
- **Service Interface To Data**
- **Clean separation between data and visualization**

AWIPS-II Designed for Extension to New Domains

AWIPS-II: Plug In adaptable data fusion and data transformation framework with Visualization

Raytheon

Architecture Overview

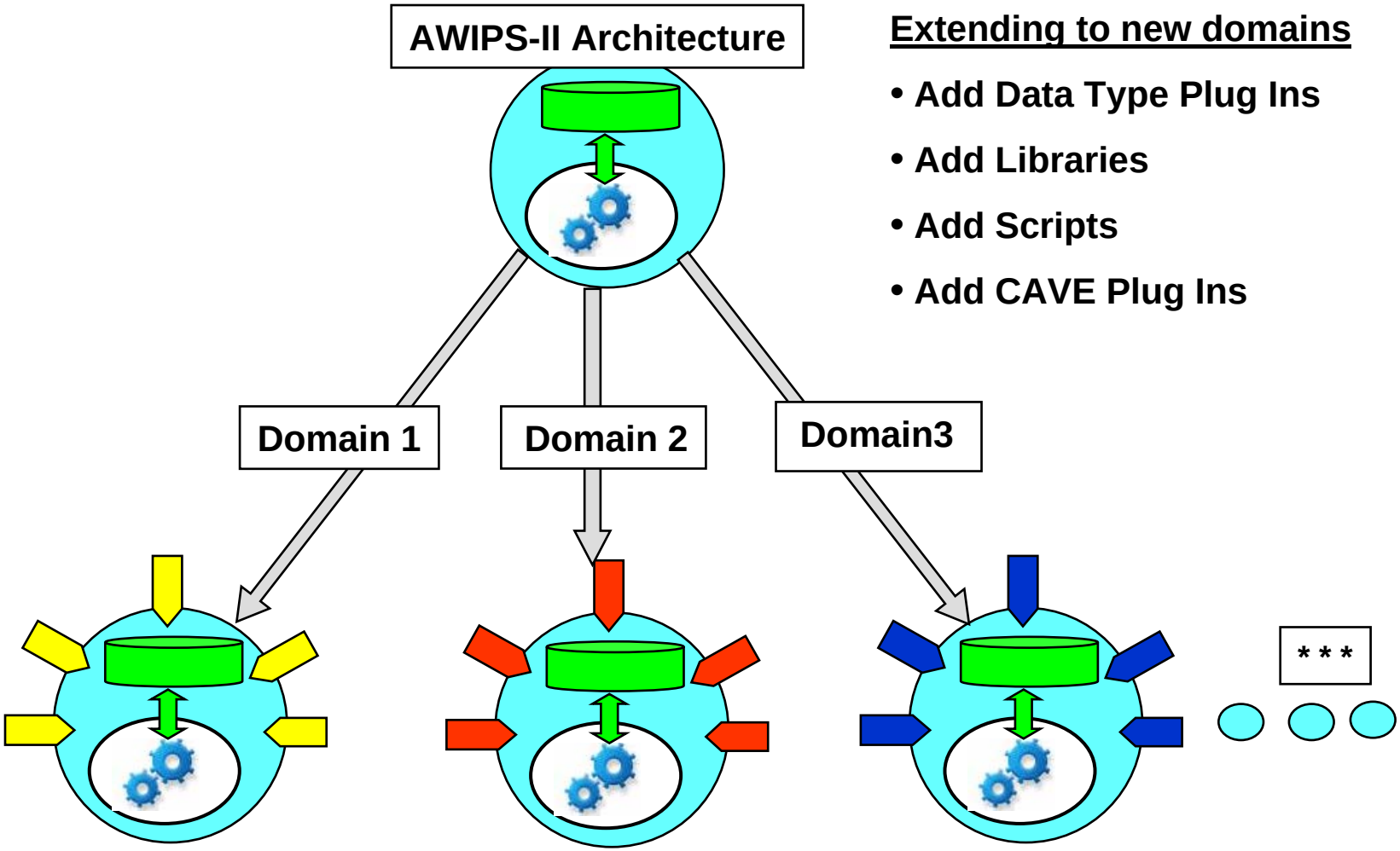


Based on Advanced Design Patterns

AWIPS-II: Designed for reuse across multiple domains

Raytheon

Architecture Overview



Extending to new domains

- Add Data Type Plug Ins
- Add Libraries
- Add Scripts
- Add CAVE Plug Ins

Low Cost Extension to New Domains

AWIPS-II Dependencies: Open Source Based

Free of all fee based COTS dependencies

Raytheon

Architecture Overview

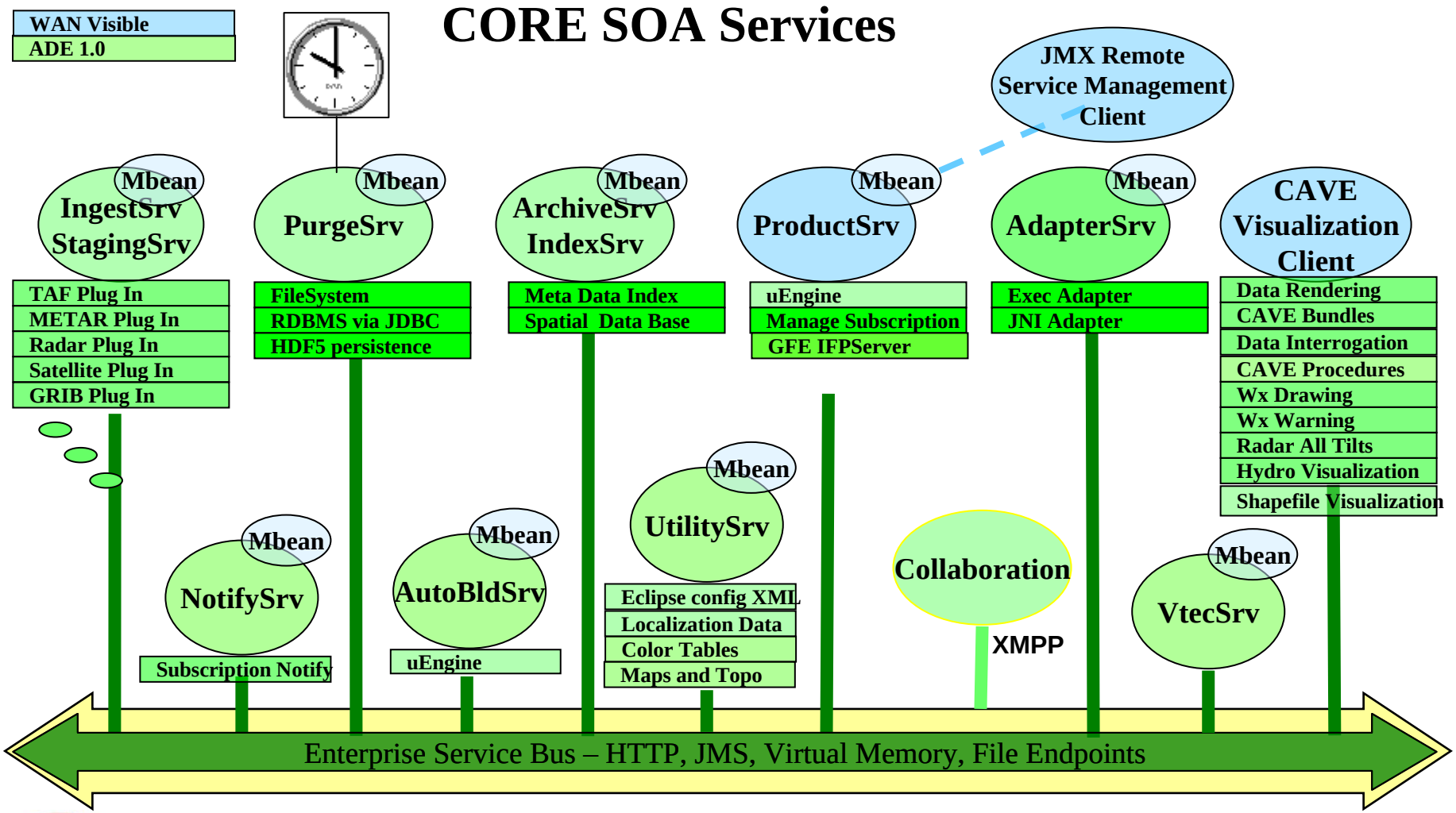
COTS Flexibility

CAMEL + Spring	Enterprise service bus and dependency injection container for SOA services
Eclipse RCP	Plug In driven visualization framework
GeoTools	Enables GIS capabilities and map projection framework
JOGL	Java API to OpenGL enables Gaming level visualization performance
HDF5	High performance file persistence of large data sets such as satellite and radar
JavaScript + Python + numPY	Data transform scripting languages with high performance math library
JAVA + ANT	Primary programming language and software build framework
activeMQ	Java messaging provider with clustering
PostgreSQL	Relational database for storing Metadata from Data plug ins and spatially enables ingested data

AWIPS-II High Level System Services
SOA Services Running in an ESB Container

Raytheon

Architecture Overview



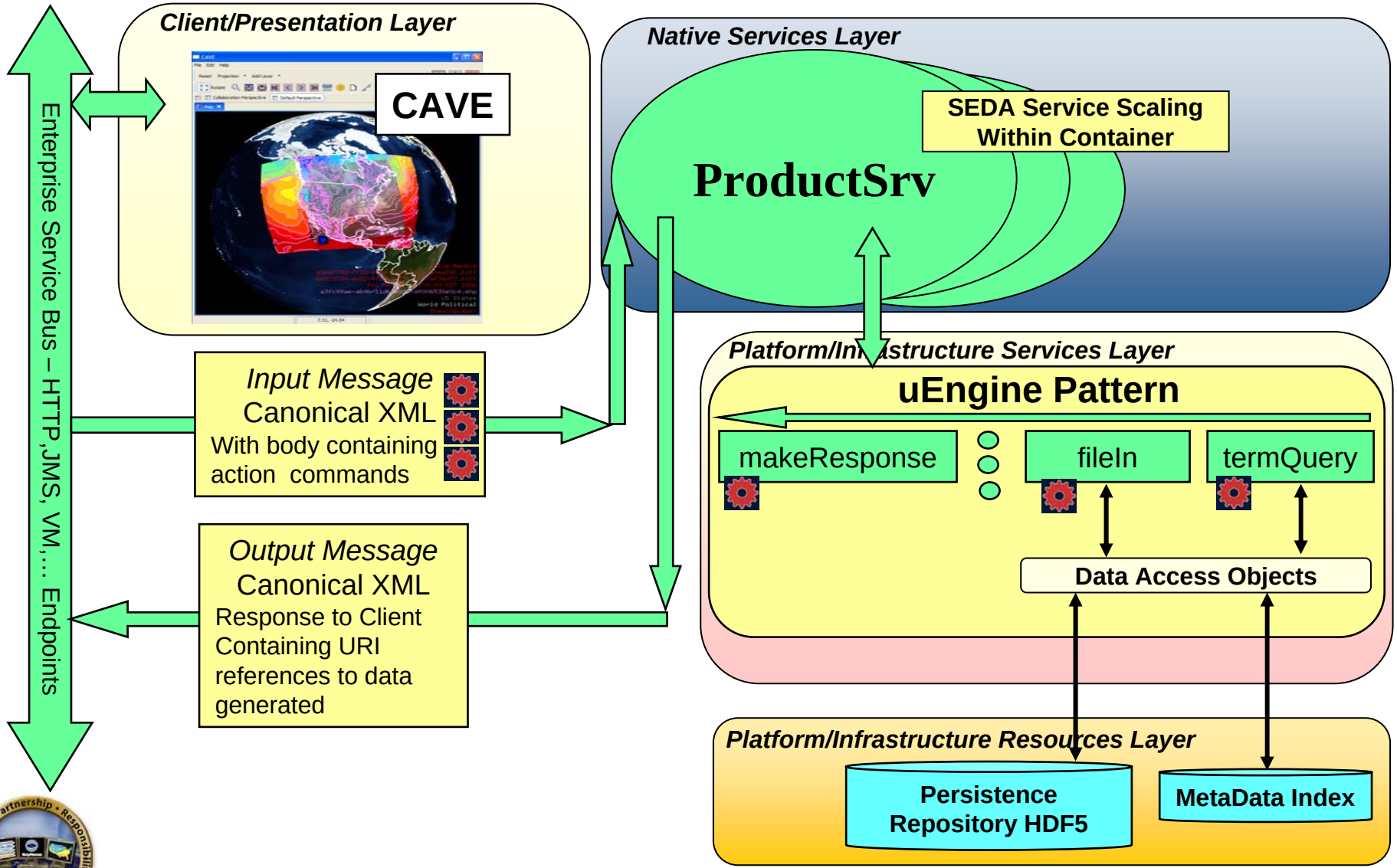
Services Independent of End Points

Requests Data for Display: as a GIS Layer

Layered View of SOA Service with Data Flow

Raytheon

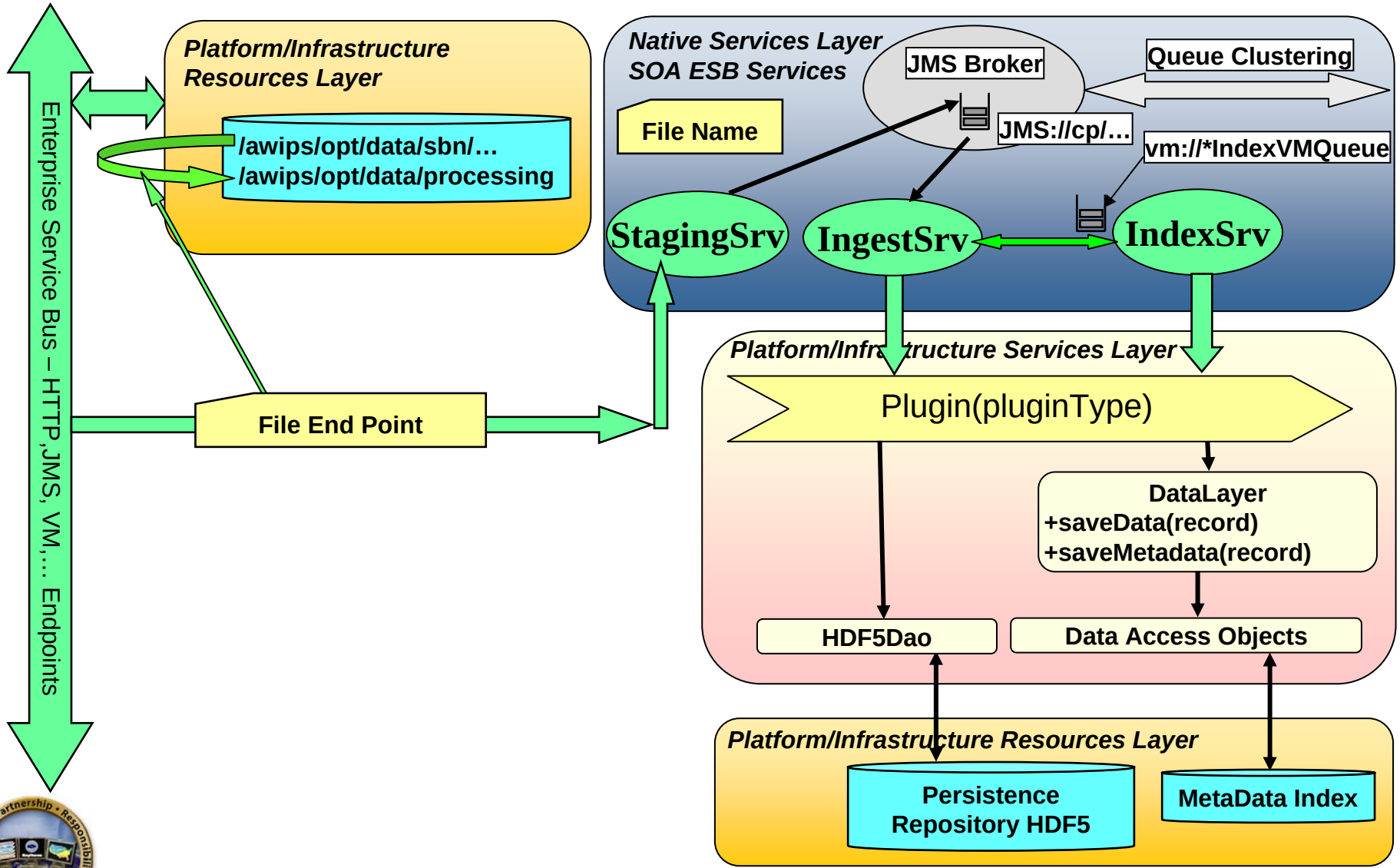
Architecture Overview



Ingest Data Flow: Layered View of SOA Service
Ingest at a Clustered End Point

Raytheon

Architecture Overview

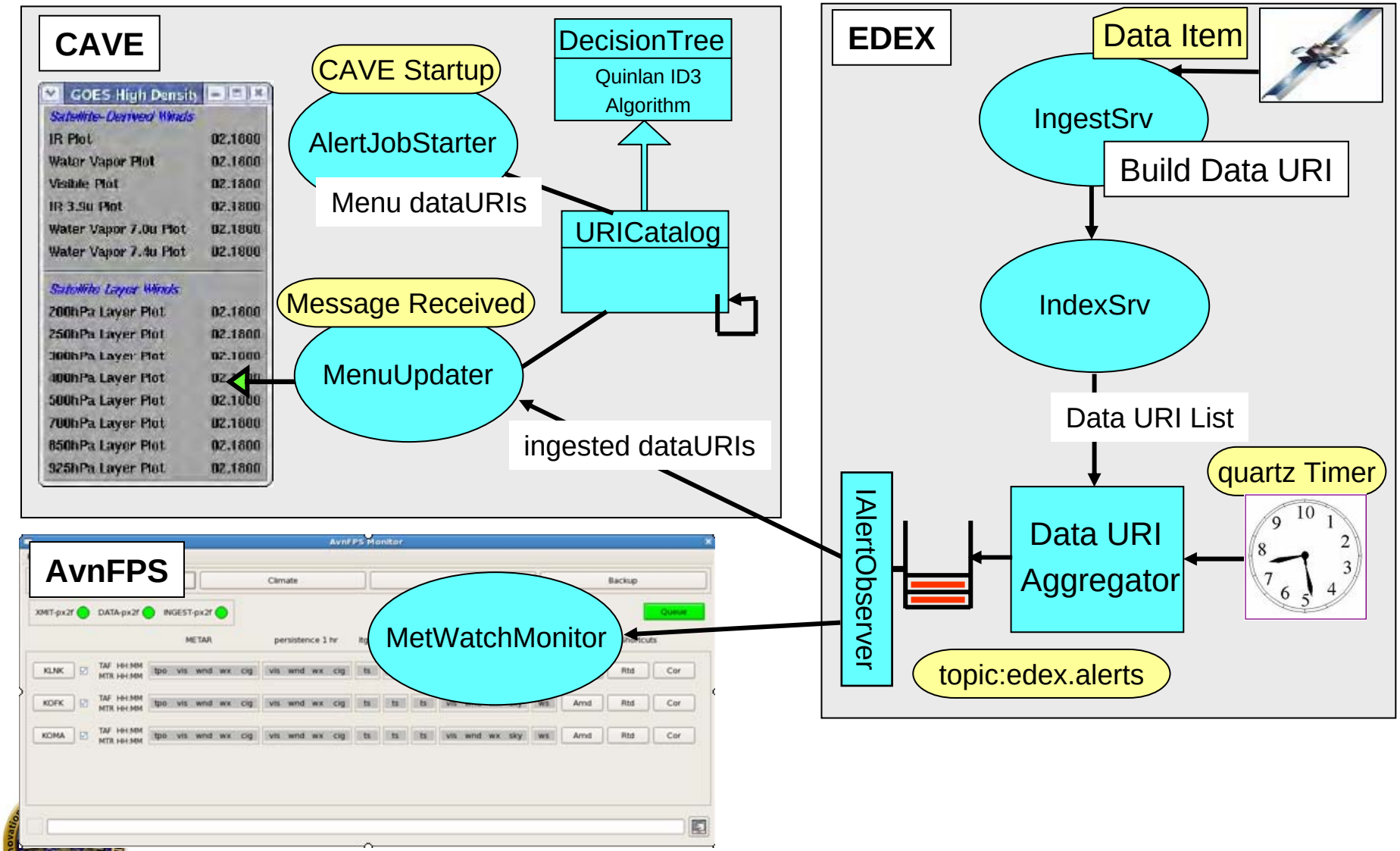


AWIPS-II Alerting Pattern Update

Added Observer Interface to enable multiple users

Raytheon

Architecture Overview

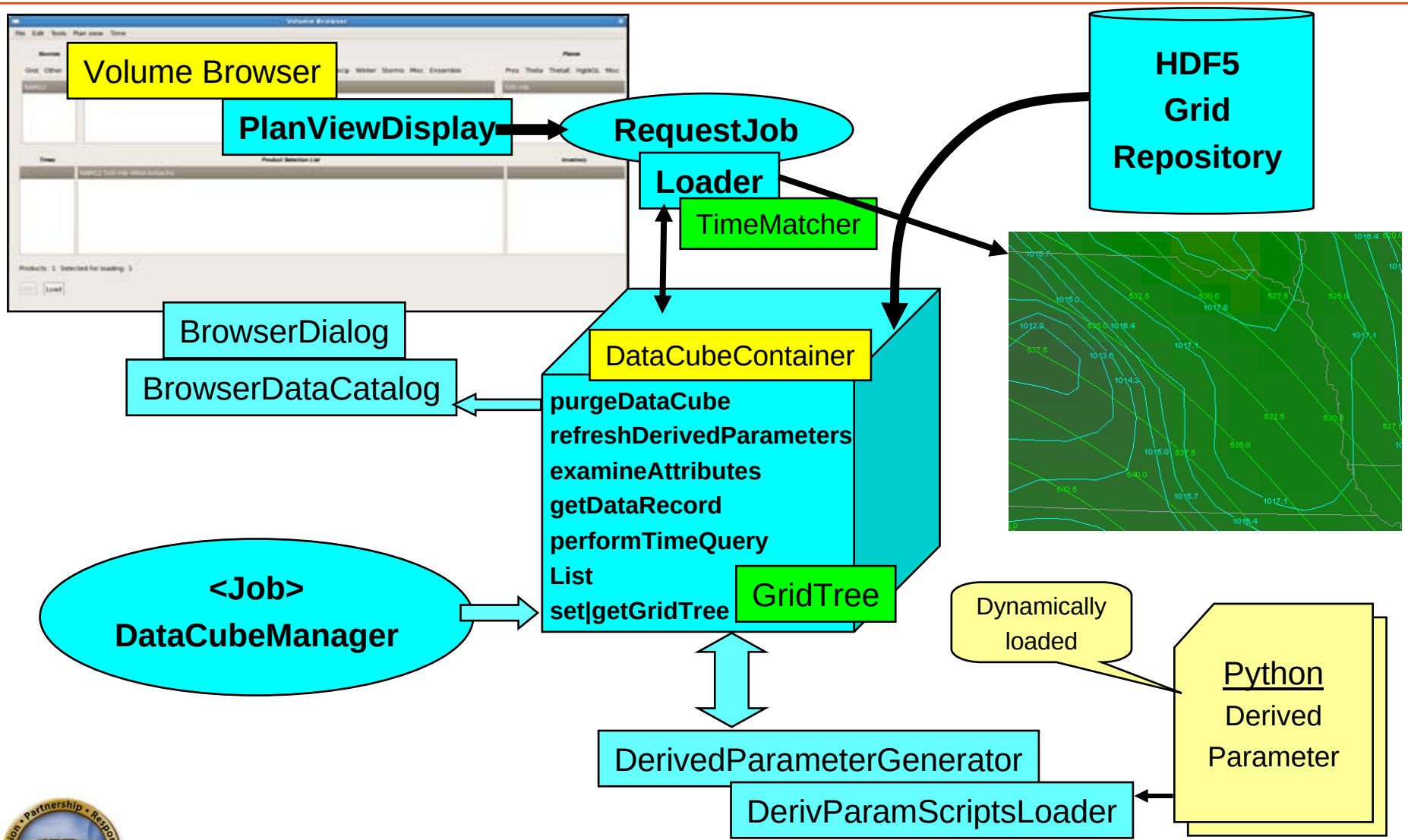


AWIPS-II Data Cube Enhancements

Redesigned data cube to enhance performance & flexibility

Raytheon

Architecture Overview



AWIPS-II: Architecture Topics

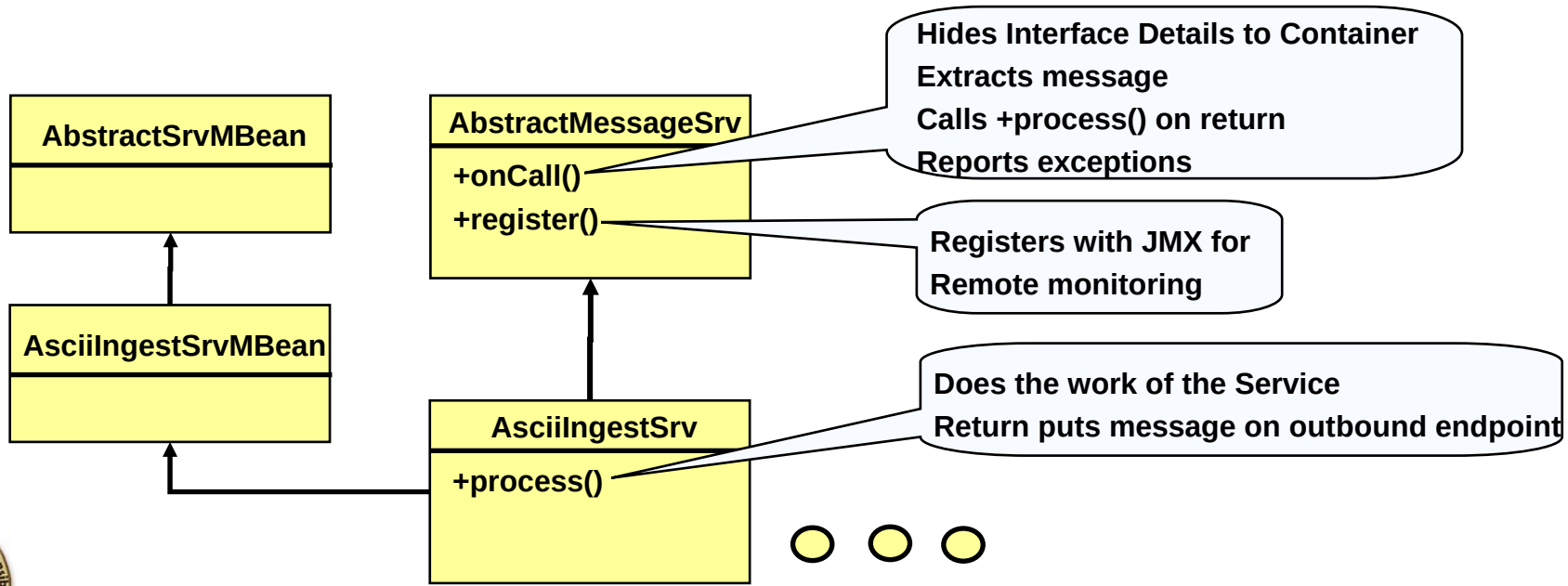
- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

SOA Service Pattern
Design Pattern

Raytheon

SOA Service

- **SOA Service Vision:** All server side capabilities are available as a standard service which communicate using standard network protocols and formats.
- **Implementation:** All services inherit from a single base class that encapsulates life cycle and communicate details. Services are Dependency Injection components that are configured to communicate with any supported endpoint.



SOA Service Pattern
Dependency Injection - Minimizes Coupling

Raytheon

SOA Service

- **Dependency Injection:** All AWIPS SOA services get there interfaces defined by XML configuration which maximizes service reuse and minimizes hardcoded coupling.

```
D:\AWIPS\EDEX\OPT\ESB\CONF
    adapter.xml
    ehcache.xml
    global.xml
    index.xml
    ingest.xml
    log4j.properties
    notify.xml
    persist.xml
    product.xml
```

Service interface and configuration Files
Generally One file for each service

Service output endpoint
Message sent when +process() executes return

Service class file reference

```
<mule-descriptor name="SBN-Sat-Ingest" singleton=[illegible]ue"
        implementation="com.raytheon.edex.se[illegible]ces.noaa.SBNsatIngestSrv"
        outboundEndpoint="vm://persistVMQueue">
        <inbound-router>
                <endpoint
                        address="file://../../data/sbn/sat/?transformers=FileToString">
                        <properties>
                                <property name[illegible]llingFrequency" value="1000" />
                        </properties>
                </endpoint>
        </inbound-r[illegible]
</mule-descriptor>
```

File endpoint wakes up Services' +process() method when data is picked up

SOA Service Pattern
Service Container and ESB startup

Raytheon

SOA Service

- **Startup:** ADE 1.0 has all SOA services running in one container. The start script automatically builds the classpath from jars in lib directory.

SOA Service Container/ESB start scripts
start – linux
start.bat -- Windows

```
D:\AWIPS\EDEX\OPT\ESB\BIN
    start
    start.bat
```

Sets clustering configuration

↓

Builds list of ESB End-point and Wiring Files

Service Configuration Files, Adding a service
Includes adding a new service config file to the startup.

↓

Starts Mule ESB

Dependency Injection Container

Canonical XML Interface Pattern
SOA service interfaces

Raytheon

SOA Service

- **Vision:** A simplified XML based message format that is independent of the endpoints. The message schema is flexible and is orientated towards the client.

Example Response Message From ProductSrv

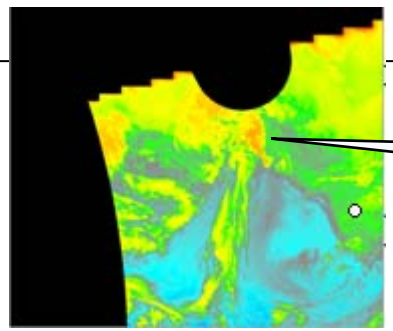
```
<message>
 <header>
  <property name="id" value="2008042422530[illegible]"/>
  <property name="uri" value=""/>
  <property name="time" value="2008042422530[illegible]"/>
  <property name="function" value="response"/>
 </header>
 <body>
  <responseURI>
   <productURI>file:/awips/edex/opt/data/uEngine/e548a5c9-ca03-4b2b-9e56-901e21895286.png</productURI>
   <fileType>png</fileType>
   <dataURI>/satellite/NESDIS/GOES-11(L)/Alaska_National/Imager_11_micron_IR/2008-01-22_14:45:15.0</dataURI>
   <validTime>2008-04-24T22:53:00.185Z</validTime>
  </responseURI>
 </body>
</message>
```

Top Message Tag

Message Header Tag
With properties

Message Body Tag

Response by URI Reference



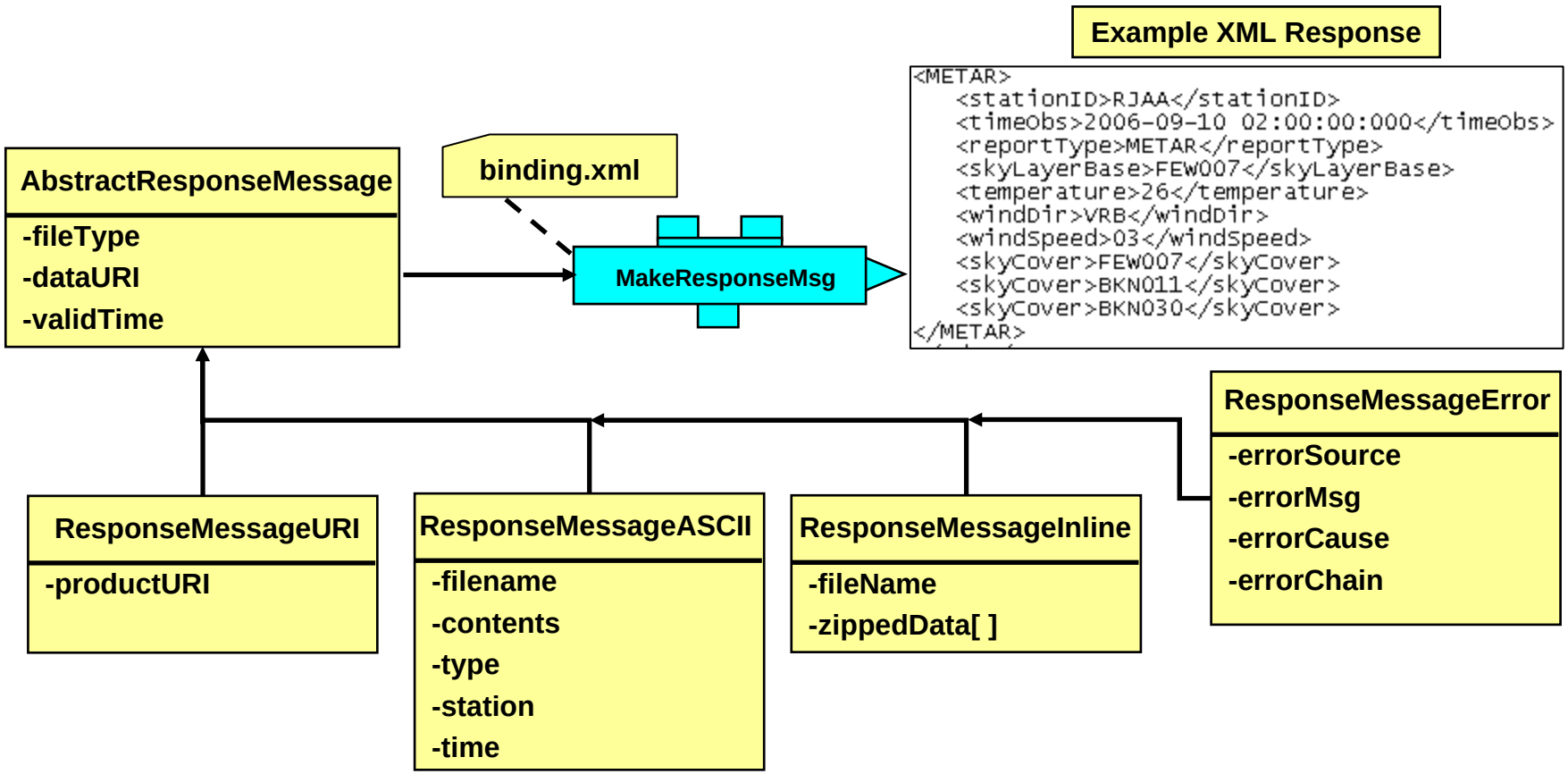
Returned product in “png” image format

Canonical XML Interface Pattern
SOA service interfaces

Raytheon

SOA Service

- **Reponse:** The body of the response is a island of XML, a base64 encoded binary image, or a URI reference for large data sets.



AWIPS-II: Architecture Topics

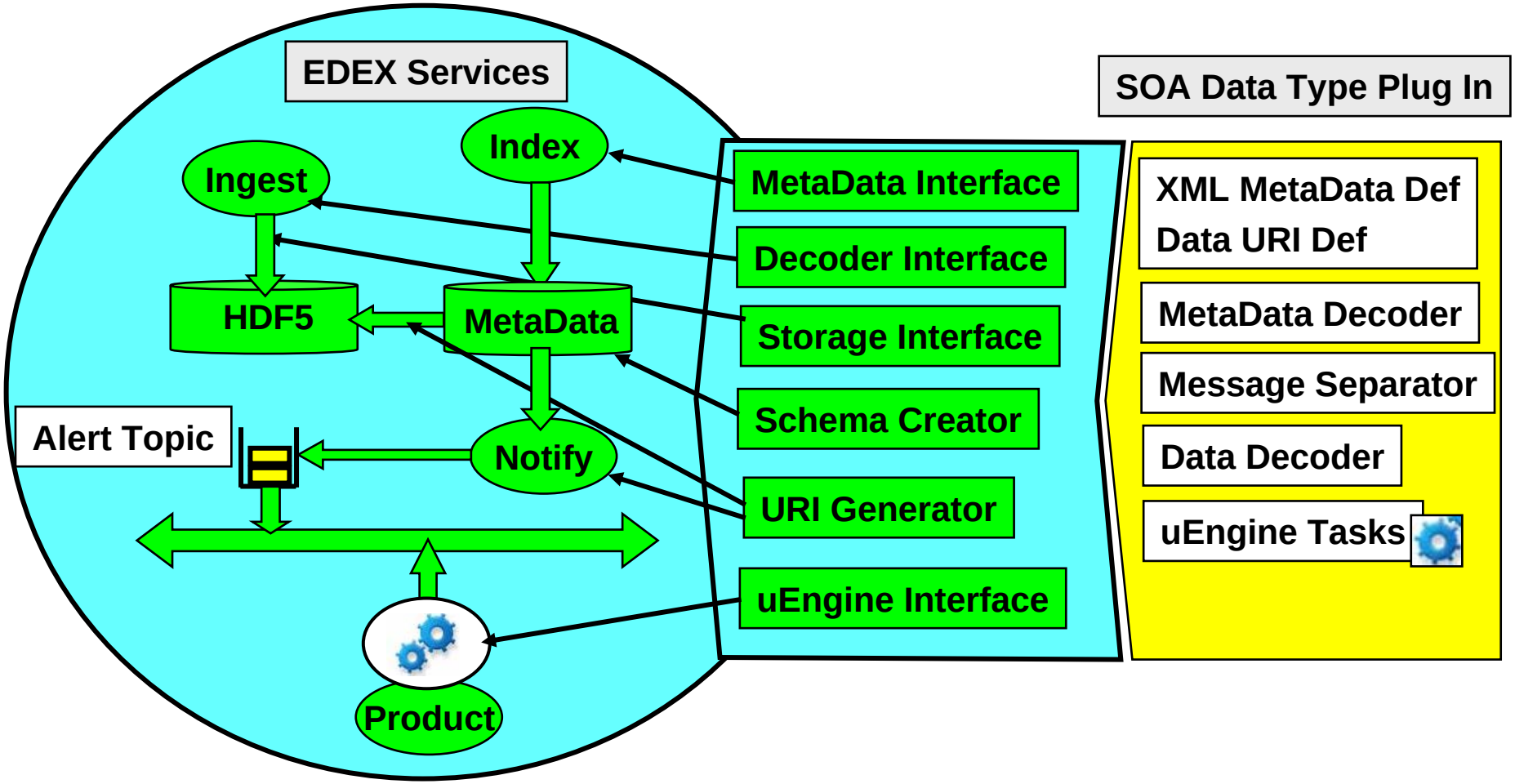
- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

AWIPS-II: SOA Data Type Plug In Pattern

Enables New Data Types and Transforms

Raytheon

Plug In Pattern

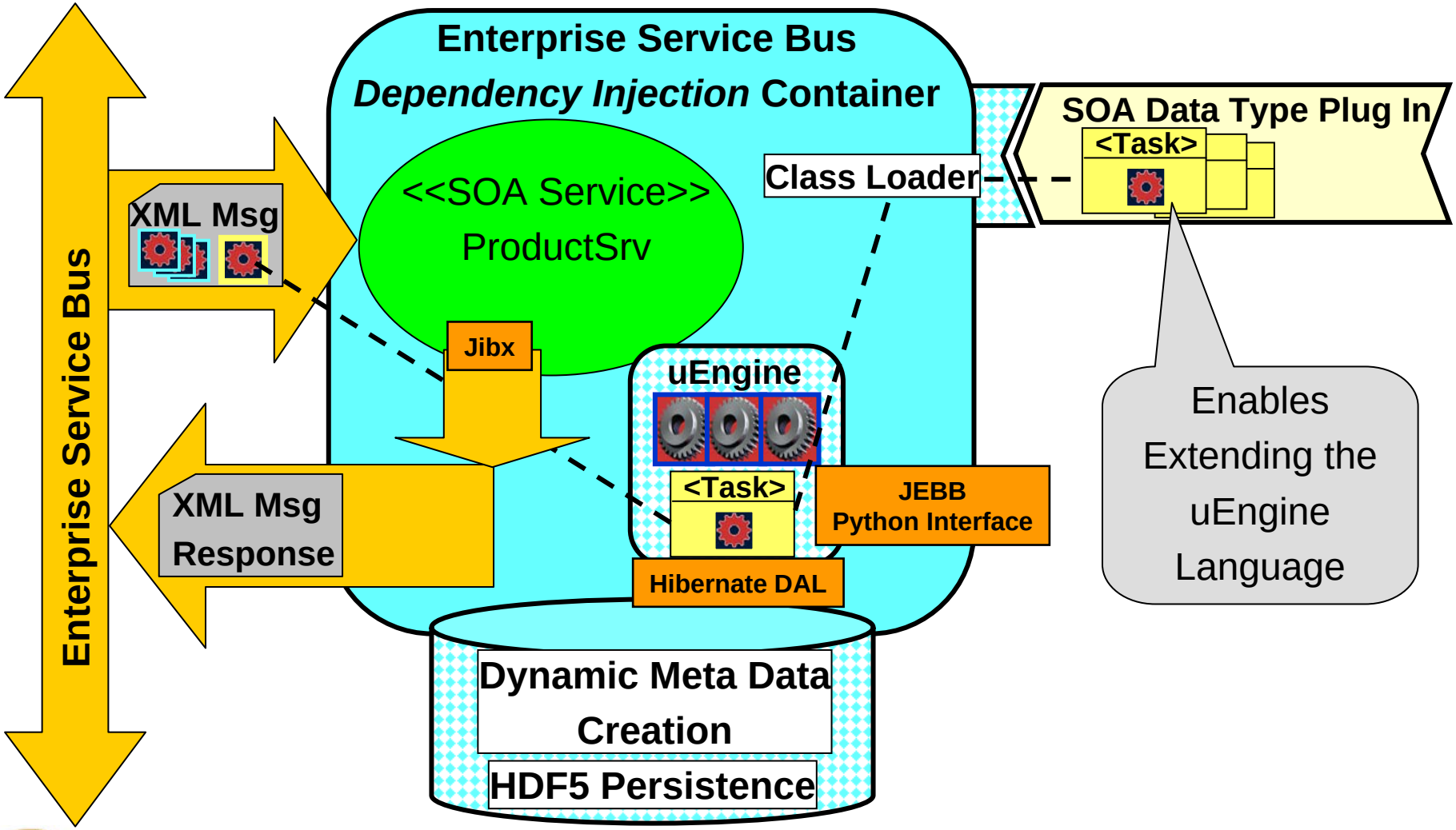


Extreme Adaptability Through Plug In Pattern

SOA Data Plug In: Enables Extensibility
Meta Data, Decode, Storage, Transforms

Raytheon

Plug In Pattern



Dynamically Extends System to New Problems

Plug Ins Enable Adapting to New Data
Automatically Creates Meta Data Persistance

Raytheon

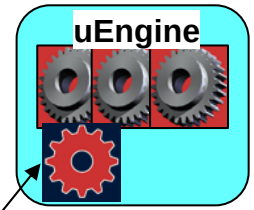
Plug In Pattern

Reference Architecture

- IProduct{Extract Metadata}
- I{*}Record
- IDecode
- IStore
- Task
- Discovery

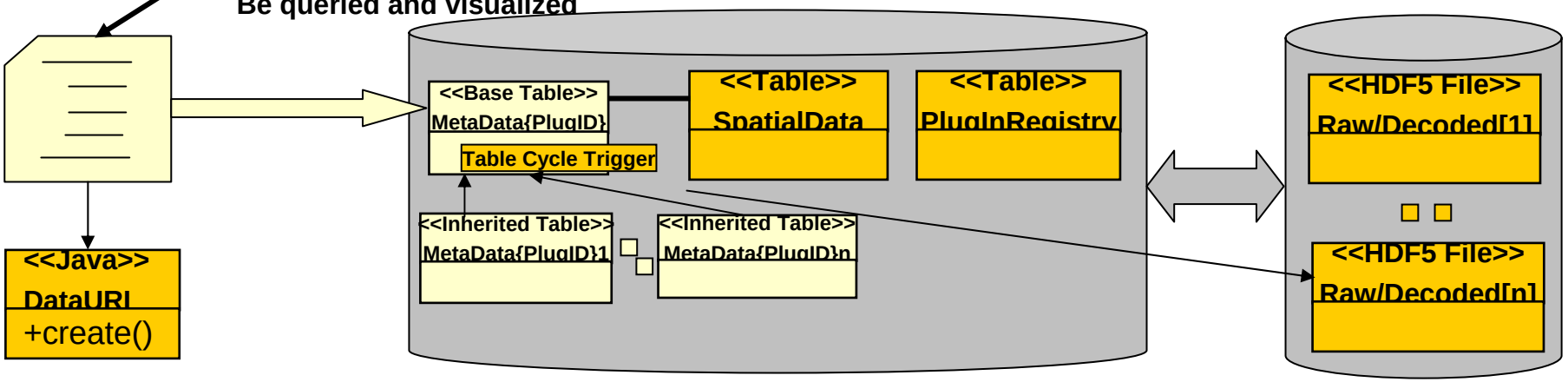
Plug In Package (JAR)

plugin.xml
metadata.xml
attributes.xml
Implementation Classes
Binding Classes
uEngine Task Implementations



Enables new science through new Tasks

Enables new data to Be queried and visualized



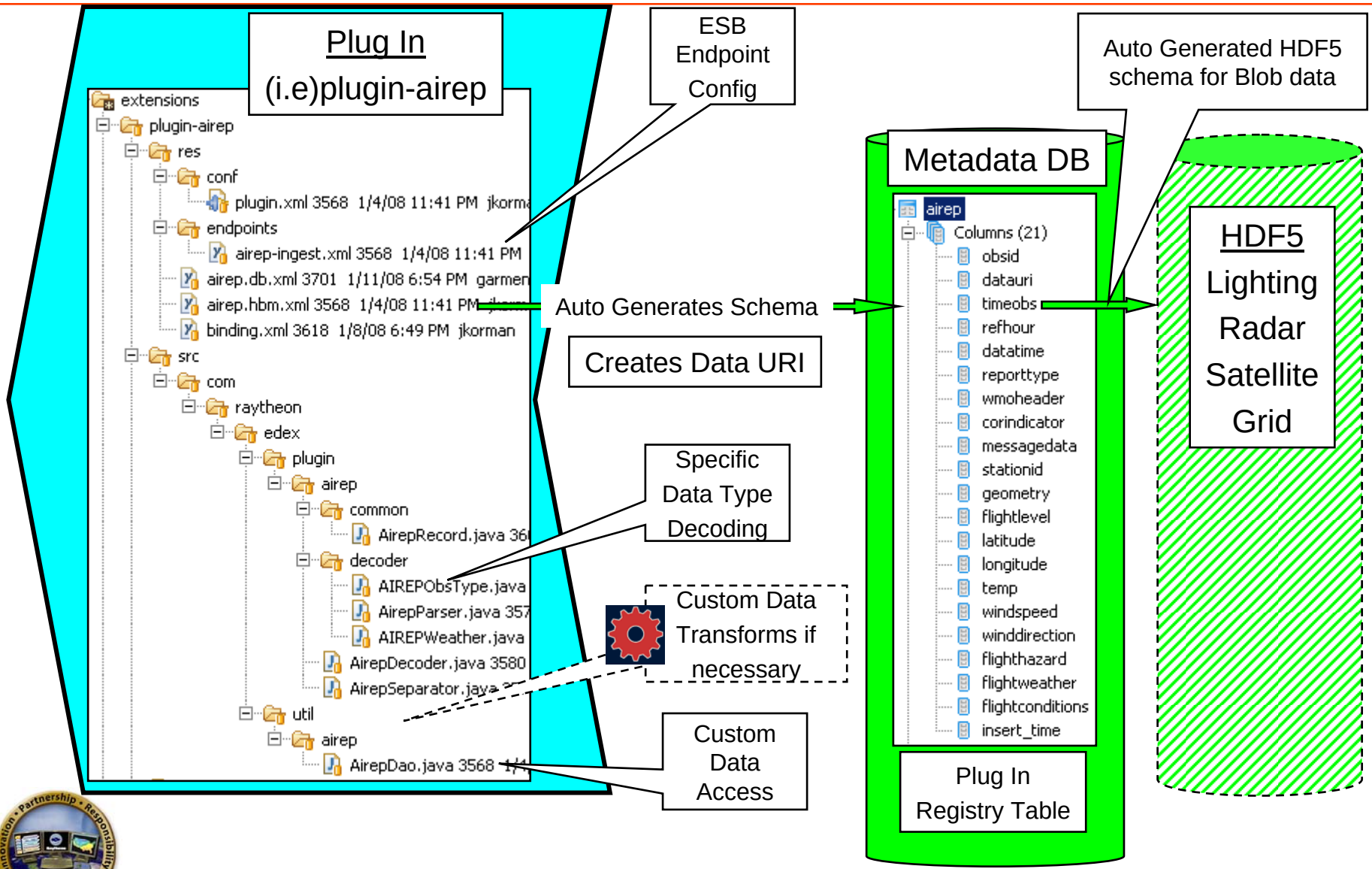
Dynamically Extendible to New Data & Transforms

Plug Ins Enable Adapting to New Data

Dynamically Extends to New Data & Transforms

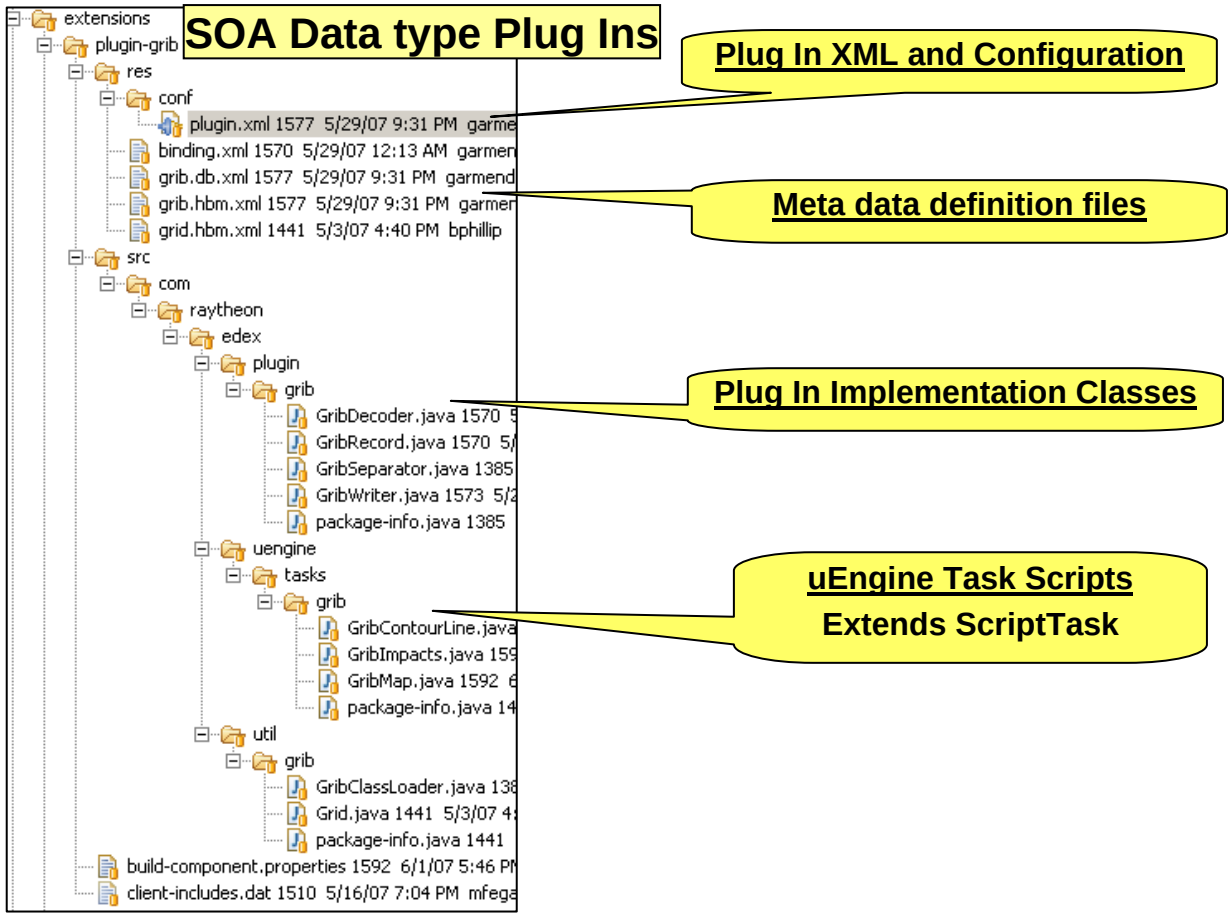
Raytheon

Plug In Pattern



SOA Plug In CM File structure

Plug In Pattern

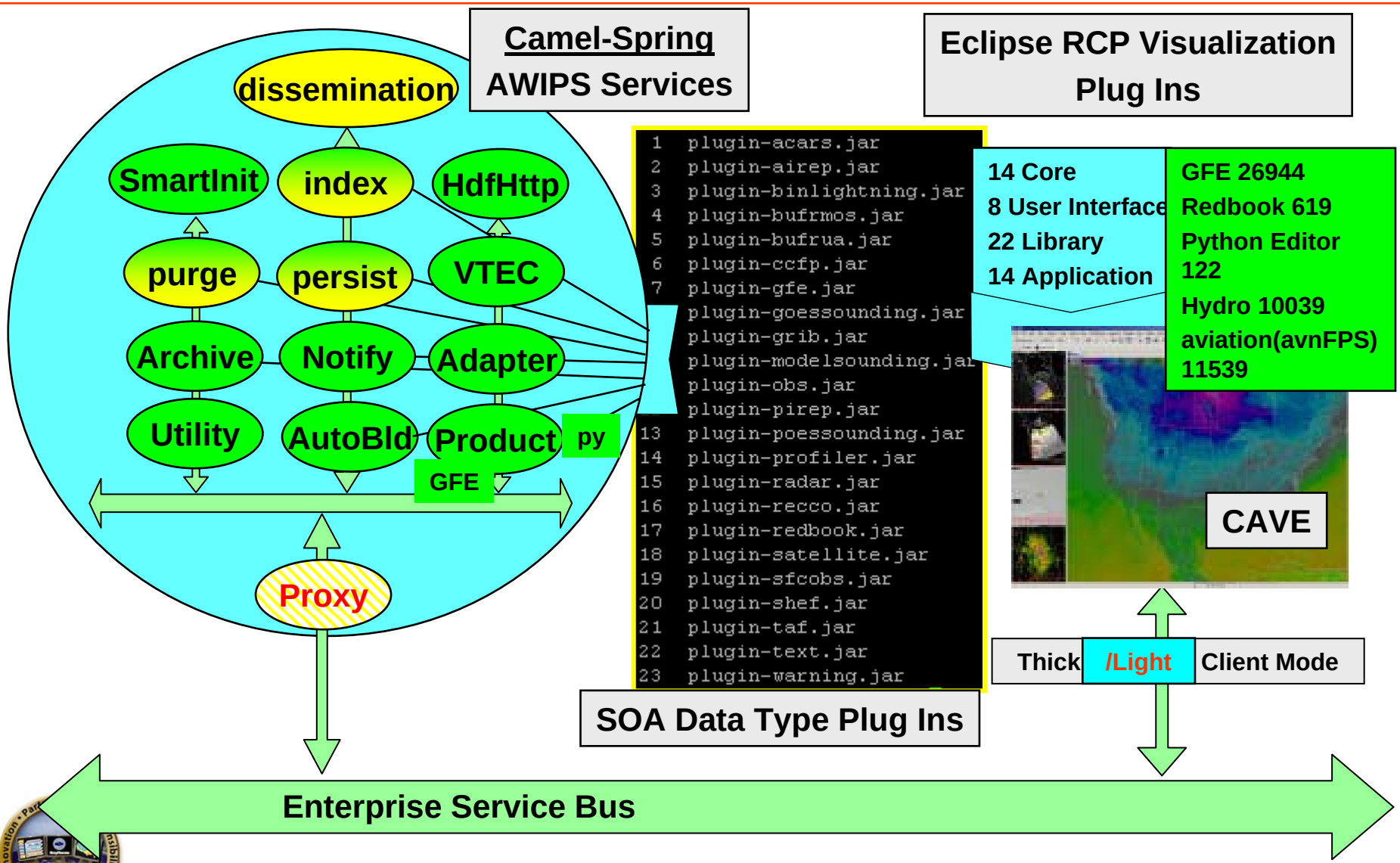


AWIPS Services and Plug Ins

ESB SOA Plug In Adaptable

Raytheon

Future Architecture

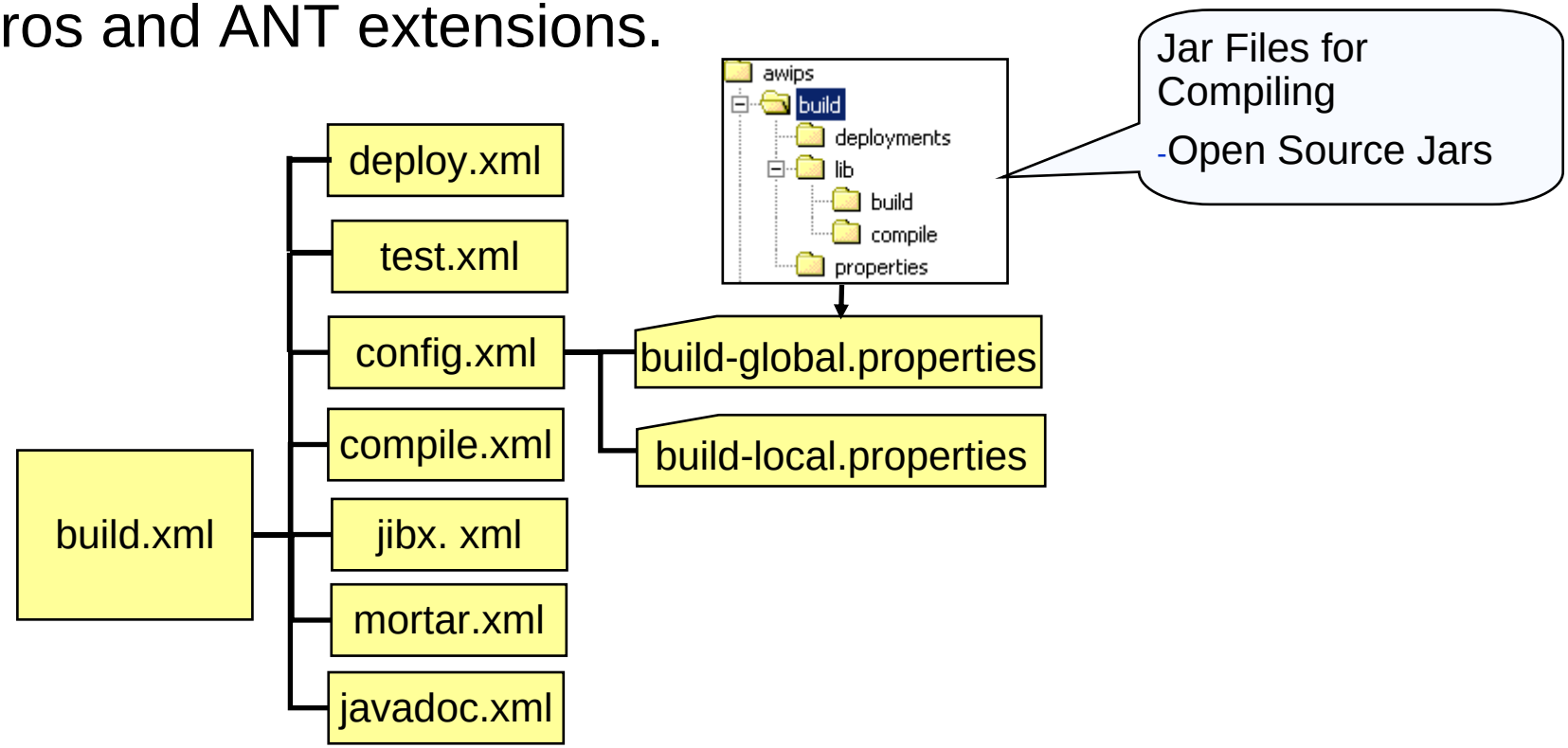


Software CM/Build/Deploy Pattern Design pattern

Raytheon

Build Patterns

- **Build Vision:** Create a simple layered build system that manages component coupling and supports partial deployment.
- **Build Implementation:** Implemented in ANT as a series of macros and ANT extensions.



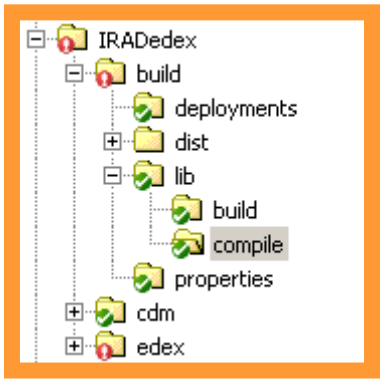
Software CM/Build/Deploy Pattern

Design pattern (Server Side)

Raytheon

Build Patterns

CM Controlled Baseline (Subversion)



1) Build services & components

ant build -Dcompfile=deployments/deployment.properties

```
D:\PROJECTS\AWIPS\BUILD\DIST\ASSEMBLE
    AdapterSrv.jar
    AutoBldSrv.jar
    common.jar
    ingestSrv.jar
    [illegible]Srv.jar
    plugins.jar
    productSrv.jar
    uEngine.jar
```

2) Deploy to run time environment

ant deploy

```
D:\AWIPS\EDEX\OPT\ESB\LIB
    AdapterSrv.jar
    beam-1.0.jar
    common.jar
    commons-configuration-1.2.jar
    geoapi-2.0.jar
    grib.jar
    gt2-api-2.2.0.jar
    gt2-coverage-2.2.0.jar
    gt2-epsg-wkt-2.2.0.jar
    gt2-geotiff-2.2.0.jar
    gt2-indexed-shapefile-2.2.0.jar
    gt2-main-2.2.0.jar
    gt2-referencing-2.2.0.jar
    gt2-shapefile-2.2.0.jar
    incubator-activemq-4.0.jar
    ingestSrv.jar
    jibx-run.jar
    jpeg2000.jar
    JTS-1.7.jar
    junit.jar
    lucene-core-2.0.0.jar
    notifySrv.jar
    productSrv.jar
    sgt.jar
    uEngine.jar
    units-0.01.jar
    vecmath-1.3.1.jar

    grib
        grib.jar
        jpeg2000.jar

    plugins
        plugins.jar
```

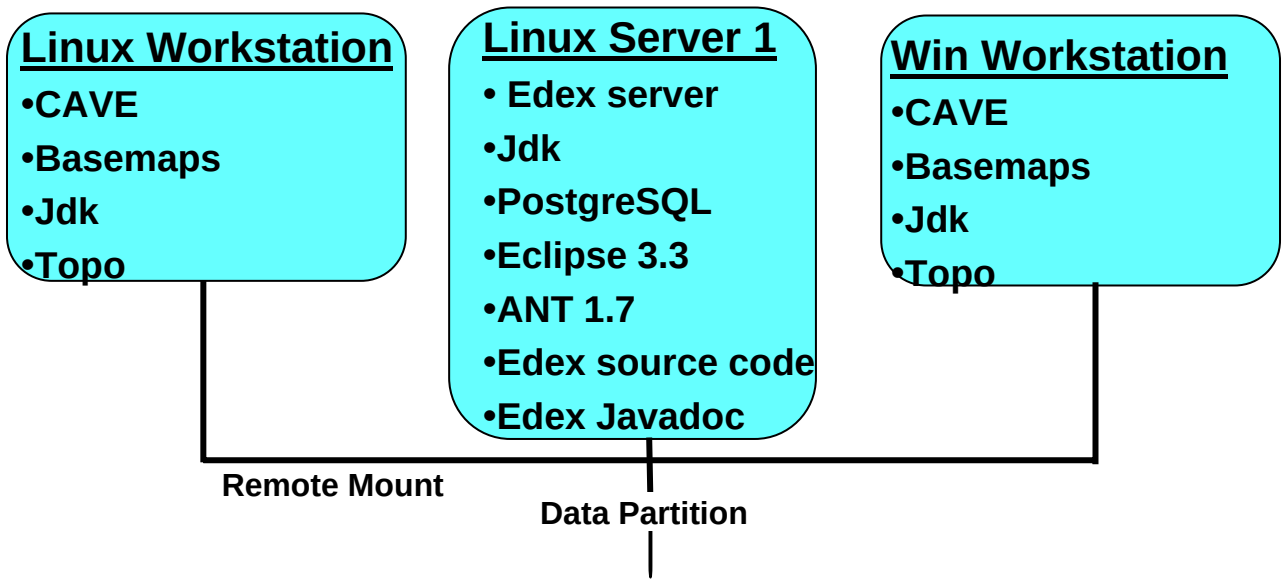
Services and Components
+
Open Source Dependencies

Special Jars

Plugin Jar Files

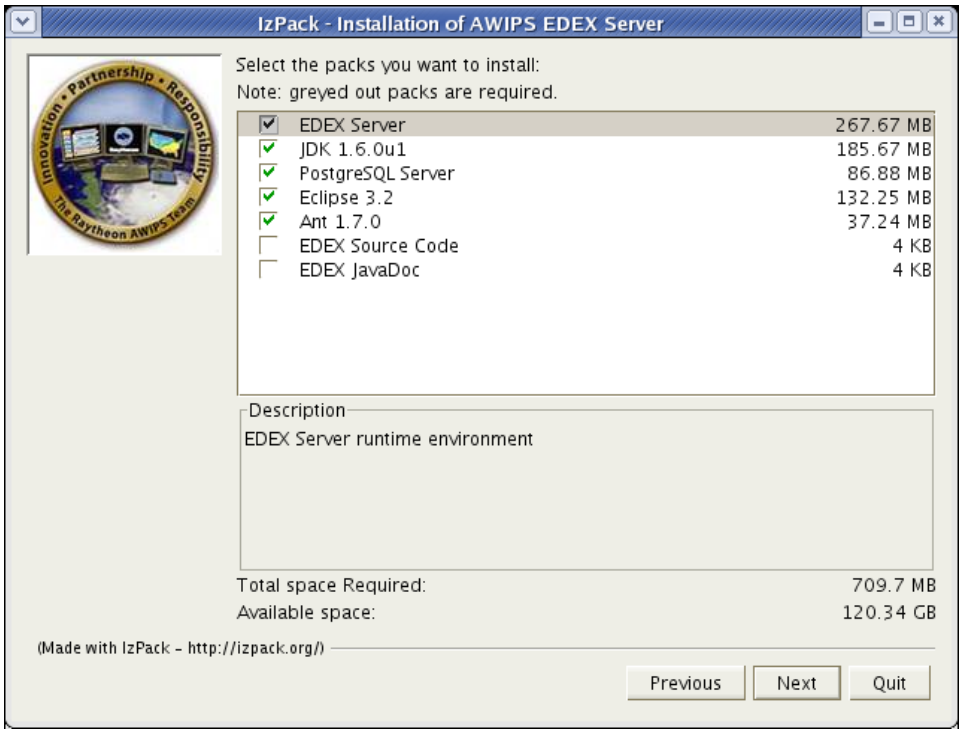
Deployment Simplification

- Created a new GUI based installer
- Standard Linux Server Install for Server Side pieces
- Standard Linux Workstation Install for CAVE
- Windows Workstation for CAVE
- Procedure for clustering (Clustering Con-ops?)



AWIPS-II Deployment Simplification

- CAVE application and EDEX services installed using two separate installers
  - Each installer can install all files to the user's home directory.



AWIPS-II: Localization Pattern
Enables SOA customization of configuration

Raytheon

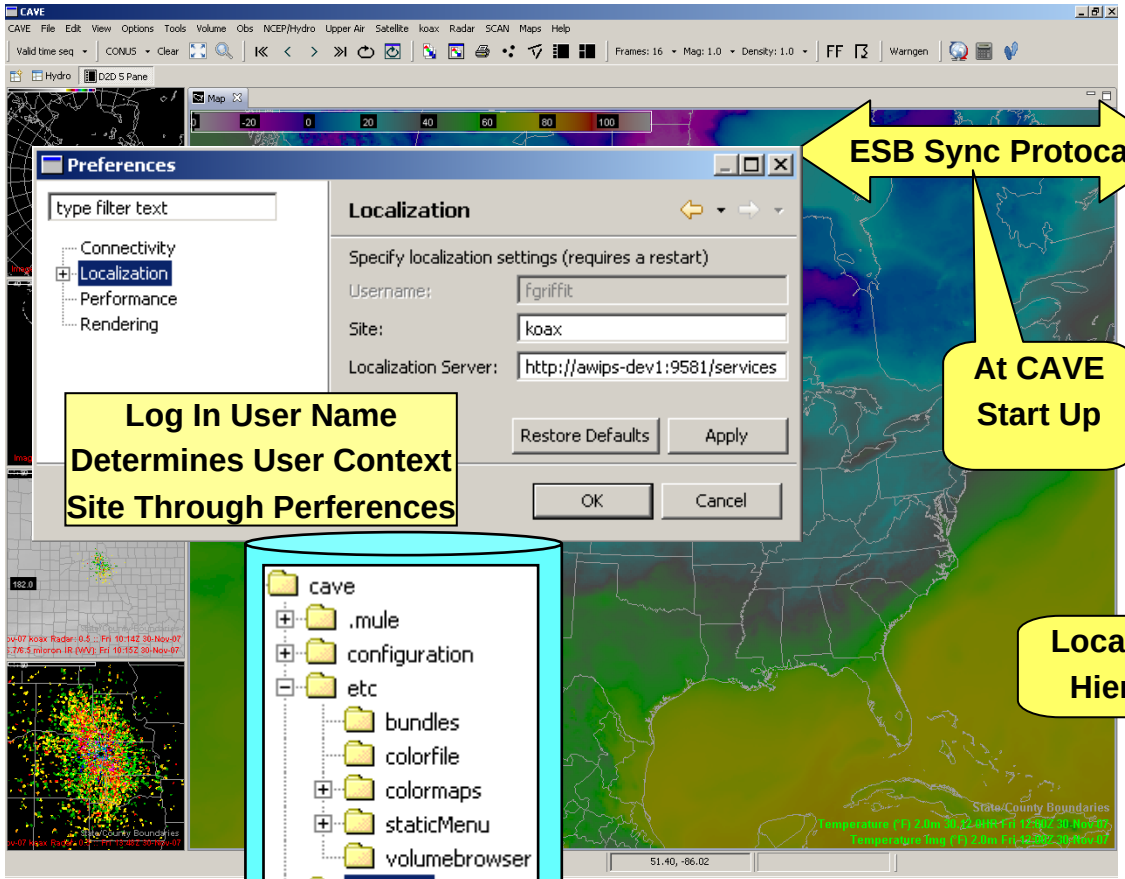
Localization Pattern

- Localization procedure occurs at startup
  - CAVE localization is controlled through Eclipse Preferences
  - Extended Eclipse Preferences to use a XML data store
- Different approach to data since CAVE can work with large data sets less subsetting is required
- Localization is provided through two simple, unified interfaces
  - Configuration for the server
  - Localization for the client, with server synchronization capability
- Localization provides a multi-tiered configuration
  - **Base** (standard national configuration values), **Site** for Server (any values the site chooses to override)
  - **Base**, **Site** and **User** for Client (values that the user overrides)

Workstation (CAVE) Localization Pattern
Uses UtilitySrv with a HTTP interface

Raytheon

Localization Pattern

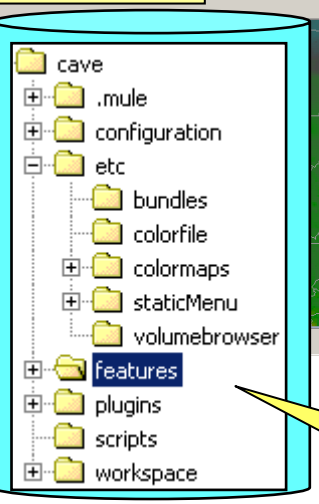


Log In User Name Determines User Context Site Through Perferences

ESB Sync Protocal

At CAVE Start Up

<<SOA Service>> UtilitySrv



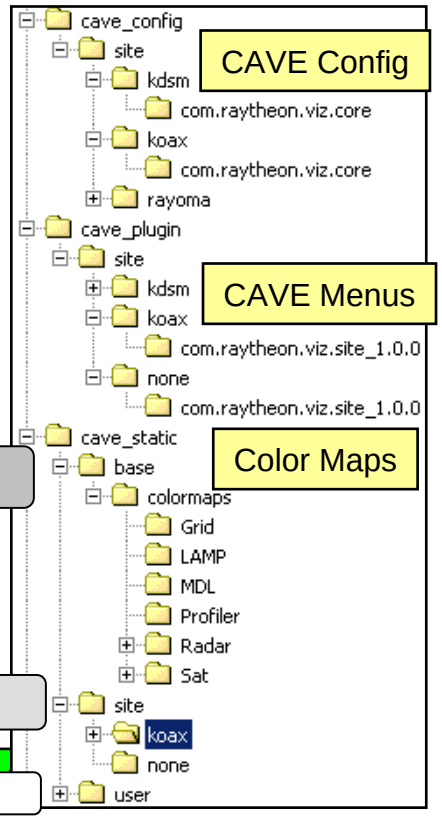
Local Persistence Of Current Active Localization

Localization Hierarchy

Base

Site

User



CAVE Config

CAVE Menus

Color Maps

Localization Data –
Hierarchical Preferences, Menus, Data …

Raytheon

Localization Pattern

Base:

```
<configuration>
   <textureCardPreference>128</textureCardPreference>
   <textureMemoryPreference>384</textureMemoryPreference>
   <framesPerSecondPreference>25</framesPerSecondPreference>
   <tileBoundaries>false</tileBoundaries>
   <connectionMethod>jms</connectionMethod>
   <jmsServerAddress>tcp://localhost:61616</jmsServerAddress>
   <dataDirectory>/awips/opt/data/hdf5</dataDirectory>
   <fontMagnification>1.0</fontMagnification>
</configuration>
```

Site:

```
<configuration>
   <siteName>KOAX</siteName>
   <siteFullName>Omaha</siteFullName>
   <siteType>WFO</siteType>
   <dataDirectory>/oax-awips/opt/data/hdf5</dataDirectory>
</configuration>
```

User:

```
<configuration>
   <fontMagnification>1.25</fontMagnification>
</configuration>
```

uEngine Task Execution Pattern
Breaks up execution into small reusable tasks

Raytheon

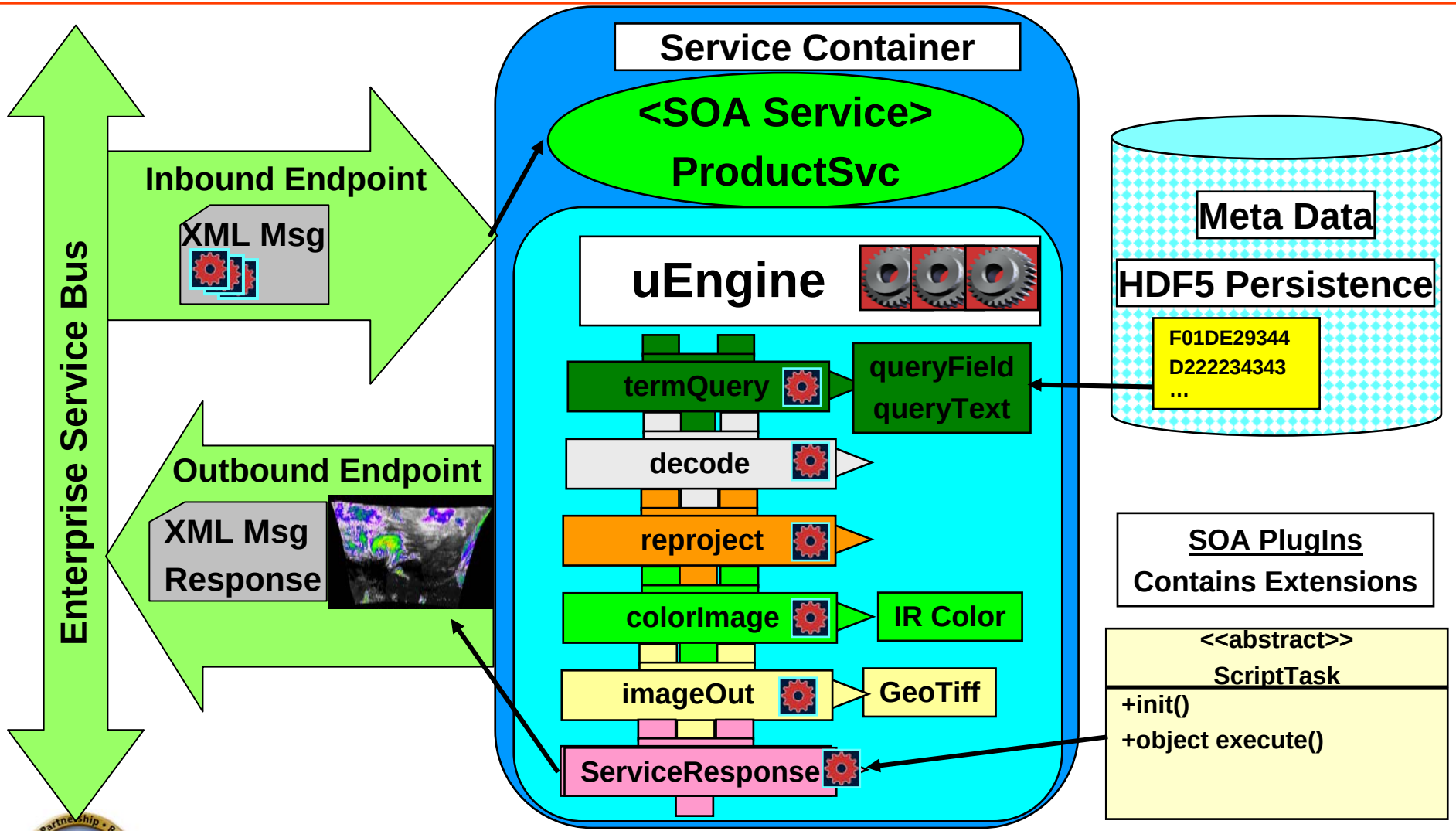
uEngine Pattern

- **uEngine Vision:** Create an execution framework for generating custom SOA requests on-demand. Customer systems can request products by script requests over a network. The script performs small general units of work that get chained together to produce a customer product.
- **uEngine Implementation:** Runs out of a SOA service "ProductSrv" that is attached for I/O to ESB endpoints. The uEngine executes scripts in a standard scripting language that has been extended. Current scripting language is **JavaScript** and transitioning to **Python with NumPy** for improved algorithm performance.

AWIPS-II: uEngine Design Pattern
Executes Commands to Transform/Fuse Data

Raytheon

uEngine Pattern



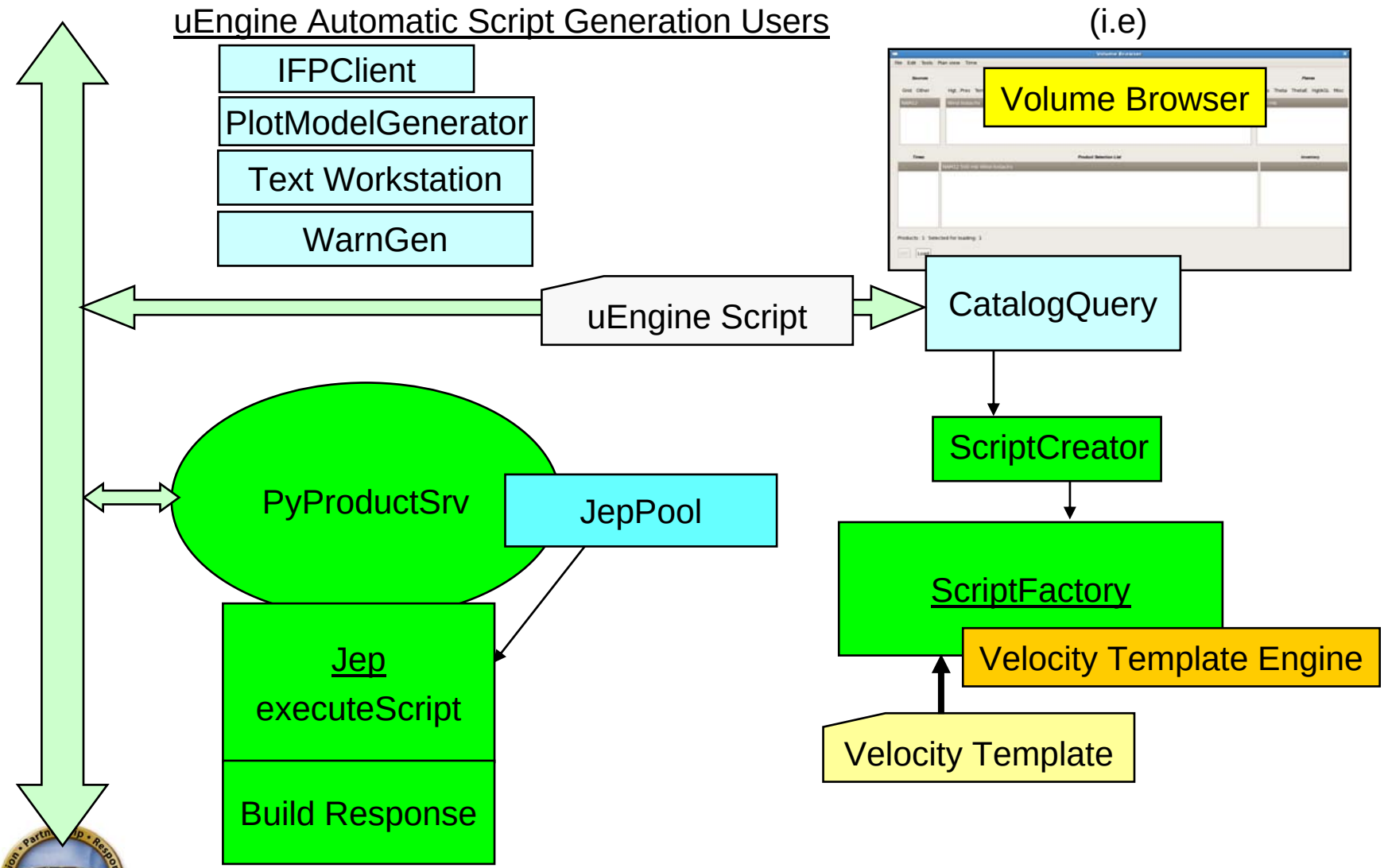
uEngine Enables Flexible User Requests/Response

uEngine Scripting Changed to Python

Enables common scripting language, and numpy benefits

Raytheon

uEngine Pattern



SOA Plug In: uEngine Tasks Extensions
Eclipse IDE Views of Plug In Implementations

Raytheon

uEngine Pattern

Plug Ins

Each gets built & Packaged in a separately Deployable Jar

- extensions/plugin-taf/src
- extensions/plugin-shef/src
- extensions/plugin-radar/src
- extensions/plugin-satellite/src
- extensions/plugin-grib/src
- extensions/plugin-metar/src
- extensions/plugin-mesowest/src

Satellite Plug In

- extensic
- com.
  - package-info.java 997 1/30/0
  - SatelliteFilenameDecoder.java 9
  - SatelliteHeaderDecoder.java 10
  - SatelliteWriter.java 1168 3/7/0
- com.raytheon.edex.satellite
  - SatelliteDecoder.java 1083 2/2
  - SatelliteRecord.java 1083 2/26
- com.raytheon.edex.tasks.satellite
  - DecodeIndexedImage.java 10

Note: Each Plug In Has a uEngine Task Package

Radar Plug In

- extensions/plugin-radar/src
- com.raytheon.edex.plugin.noaa.imagery.radar
  - GateContainer.java 1133 3/3/07 3:05 PM brock
  - ShefRecord.java 1076 2/24/07 11:47 AM garme
  - package-info.java 1000 1/30/07 3:18 PM mfegar
  - RadarTiler.java 1177 3/8/07 12:12 AM brockwoo
  - RadarWriter.java
  - RadialContainer.j
  - RadialLayer.java 1177 3/8/07 12:12 AM brockwo
  - RadialLayer8bit.java 1177 3/8/07 12:12 AM broc
  - SymbologyBlock.java 1177 3/8/07 12:12 AM br
  - SymbologyLayer.java 1177 3/8/07 12:12 AM br
- com.raytheon.edex.plugin.noaa.imagery.radar.level2
  - Level2BaseRadar.java 1177 3/8/07 12:12 AM br
- com.raytheon.edex.plugin.noaa.imagery.radar.util
  - RadarUtil.java 1177 3/8/07 12:12 AM brockwoo
- com.raytheon.edex.radar
  - RadarDecoder.java 1133 3/3/07 3:05 PM brock
  - RadarRecord.java 1133 3/3/07 3:05 PM brockw
  - RadarSeparator.java 1177 3/8/07 12:12 AM bro
- com.raytheon.edex.tasks.radar
  - DecodeRadarImage.java 1177 3/8/07 12:12 AM

Grib Plug In

- extensions
- com.ra
  - GribClassLoader.java 997 1/30/0
  - GribDecoder.java 1135 3/4/07 4
  - GribRecord.java 1170 3/7/07 10
  - GribSeparator.java 1083 2/26/07
  - GribSpatialInfo.java 1058 2/22/07
  - GribWriter.java 1168 3/7/07 7:34
  - package-info.java 997 1/30/07 2
  - UcarGribDecoder.java 1170 3/7/0
- com.raytheon.edex.plugin.noaa.grib
  - GribMsgDecoder.java 1170 3/7/0
  - package-info.java 997 1/30/07 2
- com.raytheon.edex.tasks.grib
  - Degrib.java 1141 3/5/07 10:37 A
  - GribContourLine.java 1117 3/2/0
  - GribImpacts.java 1117 3/2/07 8:
  - GribMap.java 1098 2/27/07 8:58
  - package-info.java 997 1/30/07 2

uEngine Task Execution Pattern
Enables highly flexible SOA request/response

Raytheon

uEngine Pattern

Metadata Fields Defined by SOA Plug In

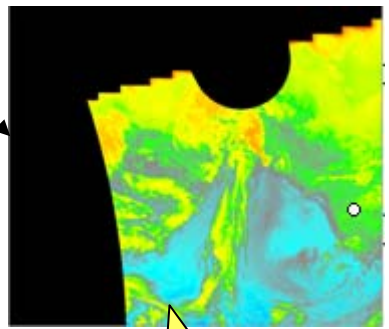
```
include("SatelliteRequest.js");
var dataRequest = new SatelliteRequest();
dataRequest.requestImage(true);
dataRequest.setCount(1);
dataRequest.addParameter("physicalElement","Imager 11 micron IR");
dataRequest.addParameter("sectorID","Alaska National");
dataRequest.addParameter("dataTime","2008-01-22 14:45:15.0");
dataRequest.setColormap("Grid/gridded data");
dataRequest.reprojectImage(false);
dataRequest.setFormat("png");
dataRequest.execute();
```

<<SOA Service>>
ProductSrv
+receive()

<<Java Class>>
UEngineScript
+run()
+setupSubscription(script)

<<Java Class>>
UEngineUtil
+convertFromJsToJava()

Provides Interface To Scripting Language



Output Product in Response by Reference

uEngine Scripting Overview
Enables SOA scripting of scientific algorithms

Raytheon

uEngine Pattern

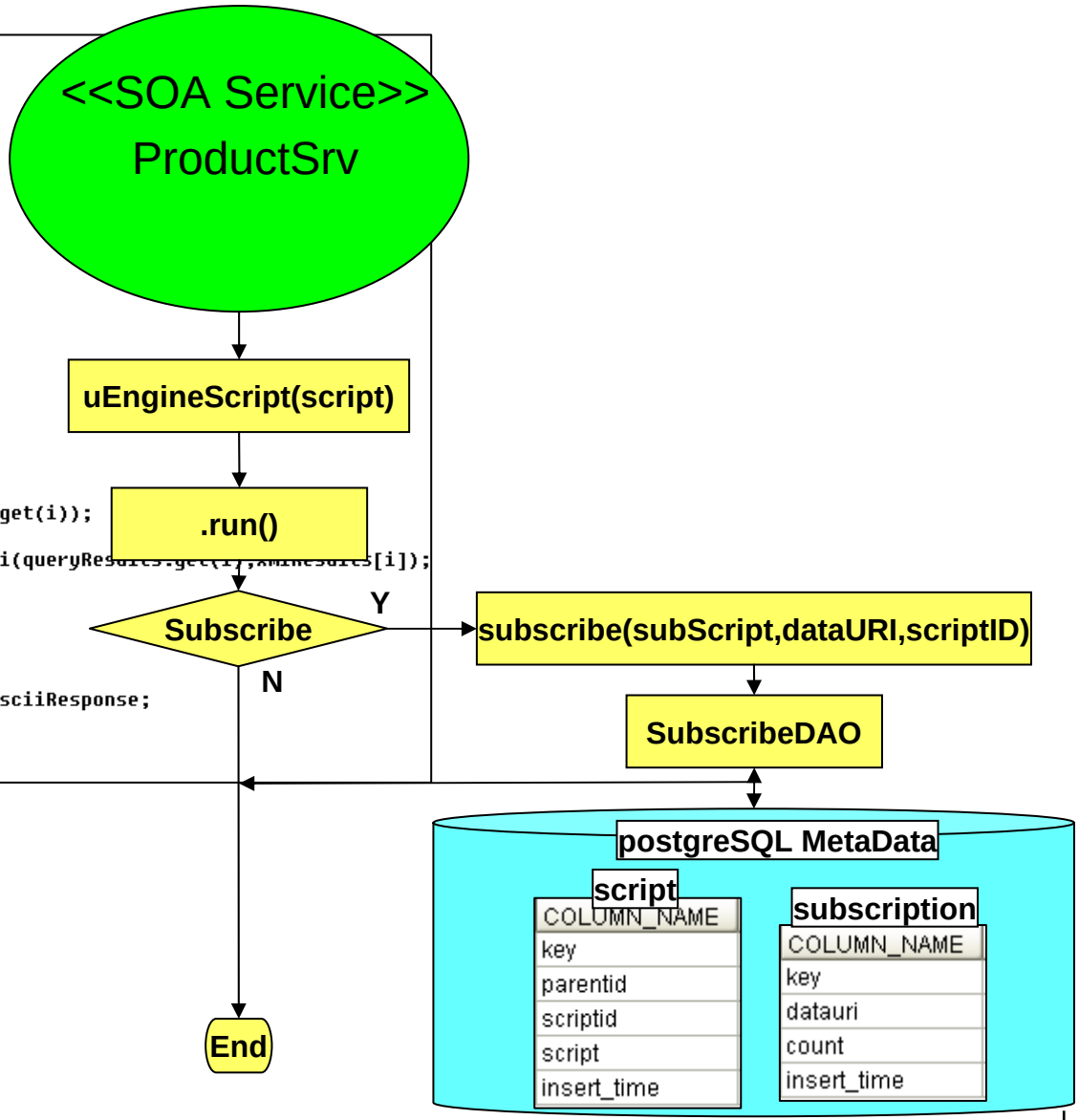
- **Goal:** Provide uEngine scripting in a language suited for scientific algorithms while enabling high performance
- Uses standard Python with JEBB for a bridge to Java
  - Uses **"numPy"** with it's numerical data types and functions
  - Includes interface to Fortran library **"Linpack"** for linear algebra
  - Includes interface to **"BLAS"** for taking advantage of multiple processing chains in modern CPUs for improved performance
- UengineScript class, simply call setScriptText(String javaScriptCode) and run()
- Extend ScriptTask, Basic POJOs with an execute() method
- Tasks only operate on one object (loops at the script level)

ProductSrv Changes
JavaScript uEngine Language & DB for Subscribe

Raytheon

uEngine Pattern

```
function MetarRequest(){
   this.subscribe = true;
   this.subscription = new Subscription();
}
function metarQuery()
{
   var query = new TermQuery("metar");
   query.addParameter("reporttype", "METAR");
   query.addParameter("stationid","KDBN");
   query.setCount(3);
   this.subscription.setup(query);
   var queryResults = query.execute();
   return makeAsciiResponse(queryResults);
}
function makeAsciiResponse(queryResults)
{
   var xmlResults = new Array();
   var response = new Array();
   for(i=0; i < queryResults.size(); i++)
   {
         var toXml = new DataToXml(queryResults.get(i));
         xmlResults[i] = toXml.execute();
         var makeResponse = new MakeResponseAscii(queryResults.get(i),xmlResults[i]);
         response[i] = makeResponse.execute();
   }
   return response;
}
MetarRequest.prototype.metarQuery = metarQuery;
MetarRequest.prototype.makeAsciiResponse = makeAsciiResponse;

var dataRequest = new MetarRequest();
dataRequest.metarQuery();
```



AWIPS-II GIS Capabilities – GeoTools
GIS Data Types, Transforms, and Display

Raytheon

GIS Capability

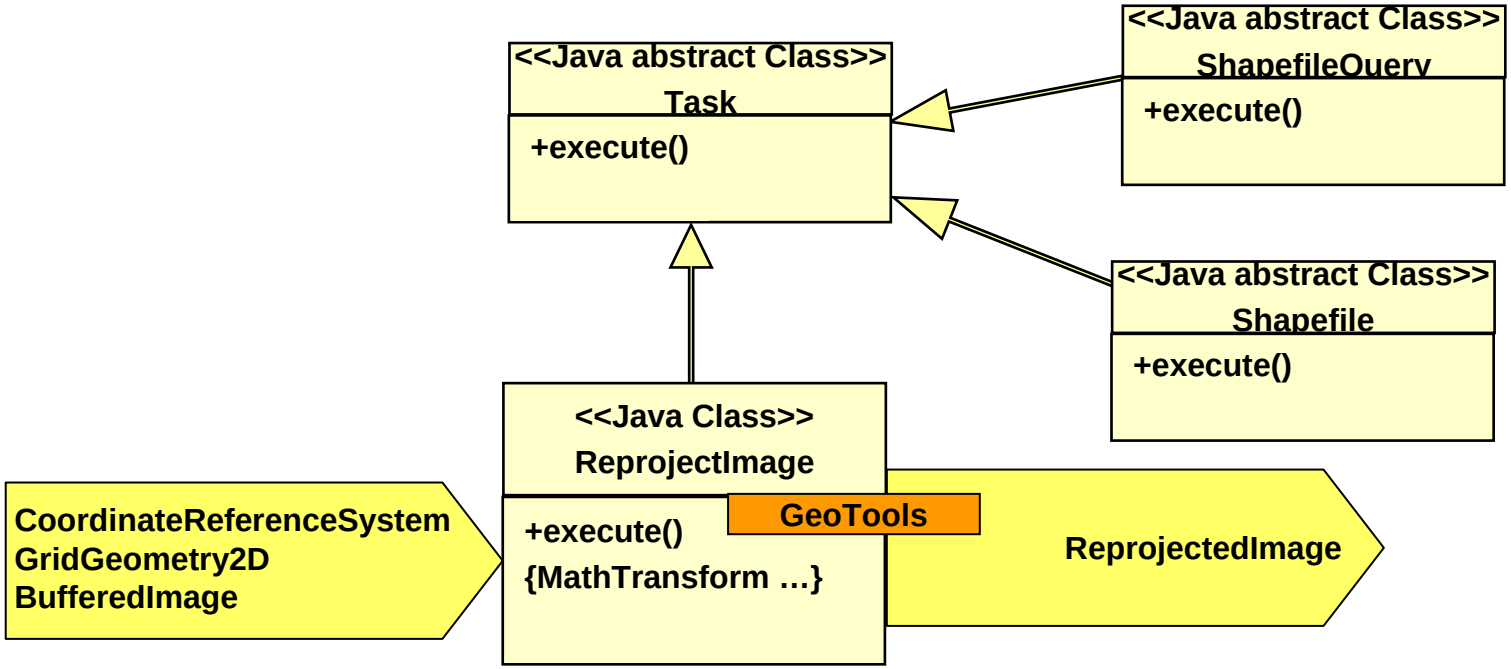
- DataStore and Data Formats
  - ESRI Shapefile (Read/ Write)
  - WFS Web Feature Server (OGC) (Read / Write)
  - PostGIS geometric objects for PostgreSQL
  - GML Geography Markup Language
  - GeoTIFF geo-referenced TIFF image
- Coordinate Transformation (OpenGIS CTS)
  - Map Projections (Mercator, Transverse Mercator, Lambert Conformal Conic, Albers Equal Area Conic, Stereographic, Orthographic, …)
  - Math Transform

Map Library Capability – Map reprojection
Transform Ingested Data to Display Projection

Raytheon

GIS Capability

- GeoTools builds a transformation Matrix that gets reused in transform operations.
- The Transformation matrix can be build up through a series of transforms.
- Used In Barnes Analysis Pattern to calculate distances.
- Degrib Task uses GeoTools to dynamically define projection data (I.e. NCEP hur.)



Map Libraries Through GeoTools + JTS
JTS Topology Suite Spatial Operations Java API

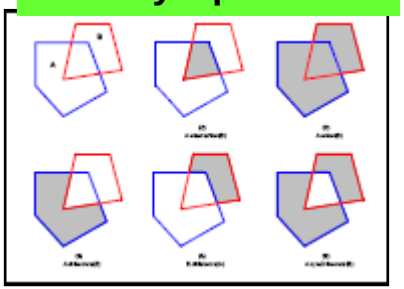
Raytheon

GIS Capability

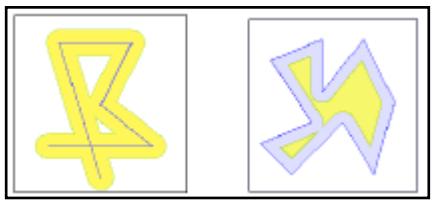
Computing Spatial Relationships
- Shape File Intersect Demonstration
- Distance Tool Demonstration

Method	*Meaning*
Equals	The Geometries are topologically equal
Disjoint	The Geometries have no point in common
Intersects	The Geometries have at least one point in common (the inverse of Disjoint)
Touches	The Geometries have at least one boundary point in common, but no interior points
Crosses	The Geometries share some but not all interior points, and the dimension of the intersection is less than that of at least one of the Geometries.
Within	Geometry A lies in the interior of Geometry B
Contains	Geometry B lies in the interior of Geometry A (the inverse of Within)
Overlaps	The Geometries share some but not all points in common, and the intersection has the same dimension as the Geometries themselves

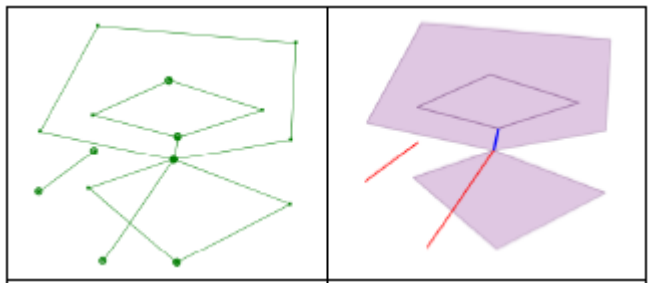
Overlay Operations



Buffer Operations



Polygonization & Quantization



Spatial Table Refactor – Geo Spatial
Postgis extension to PostgreSQL

Raytheon

GIS Capability

- Geo Spatial Enabling Data
  - Chosen Approach: Create Static Spatial Tables in PostgreSQL
    - PostGIS extension: Free, simple, high performance
  - Input / Output:
    - All ingested data is spatially indexed and can be spatially queried
    - Can create Shape and GeoTiff output

PostgreSQL Data Base

radar_spatial

gid [PK] serial	rda_id character varying(4)	rpg_id_des ch	immutablex	wfo_id character varying(4)	eqp_elv numeric	name character varying(36)	lon double	lat double	elevmeter double pre	the_geom geometry
1	TADW	300		LWX	346.1500	Andrews AFB, MD	-76.845	38.695	105.50652	0101000000A

spatial_grids

grid [PK] integer	description character varyin	nx in	ny	dx	dy double pre	crs character varying(2047)	coverage geometry
87	U.S. Area; used in	81			68.153	PROJCS["Polar Stereographi	0103000000010

spatial_obs_stations

gid [PK] serial	icao character var	wmoindex intege	name	country	state ra	wmoregion integer	elevation integer	upperaireleva integer	rbsnindicator character var	stationgeom geometry	upperairgeon geometry
1	AGGH	91520				5	8	9	P	01010000009A9	

spatial_satellite

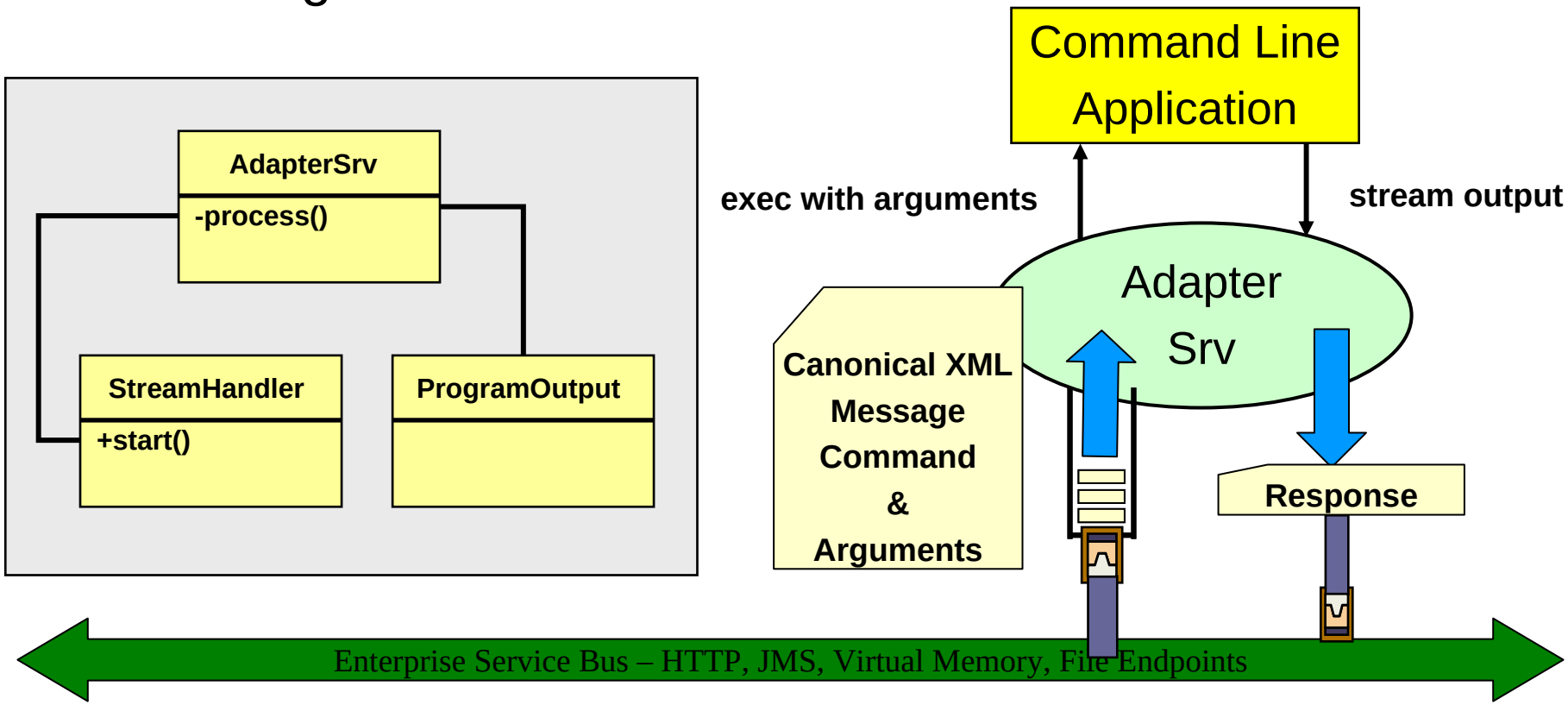
id [PK] ch	description character varying(255)	type	px	py	crs character varying(20	coverage geometry
TICF01	Multi-Satellite Composite	visible	1024	1024	PROJCS["Polar Stereogra	0103000000010

Legacy Adapter Pattern
SOA enable legacy application functions

Raytheon

Adapter Pattern

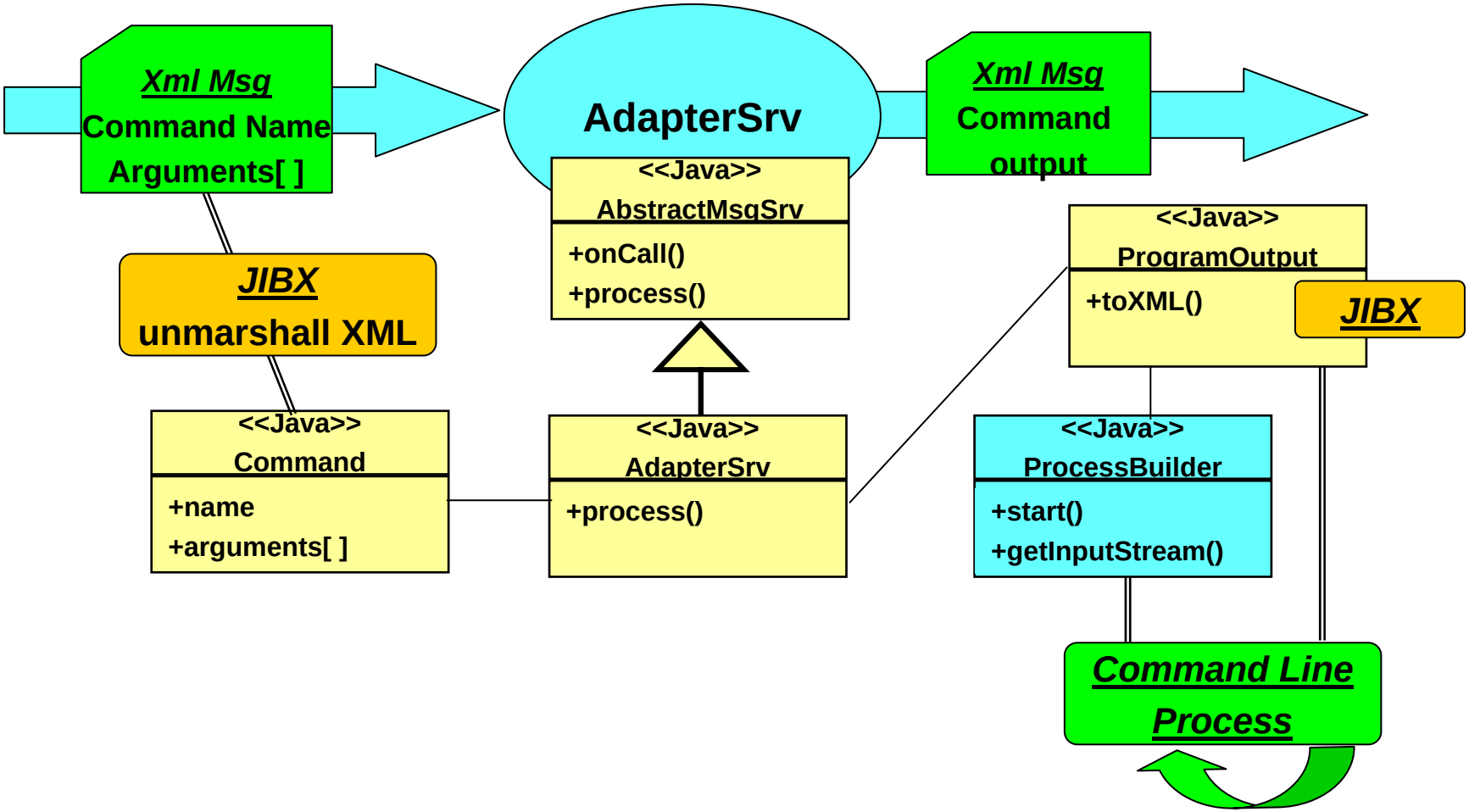
- **Legacy Adapter Vision:** Enable command line functionality to be brought in the the SOA service flow



Legacy Adapters Pattern
AdapterSrv Receives XML messages of commands

Raytheon

Adapter Pattern

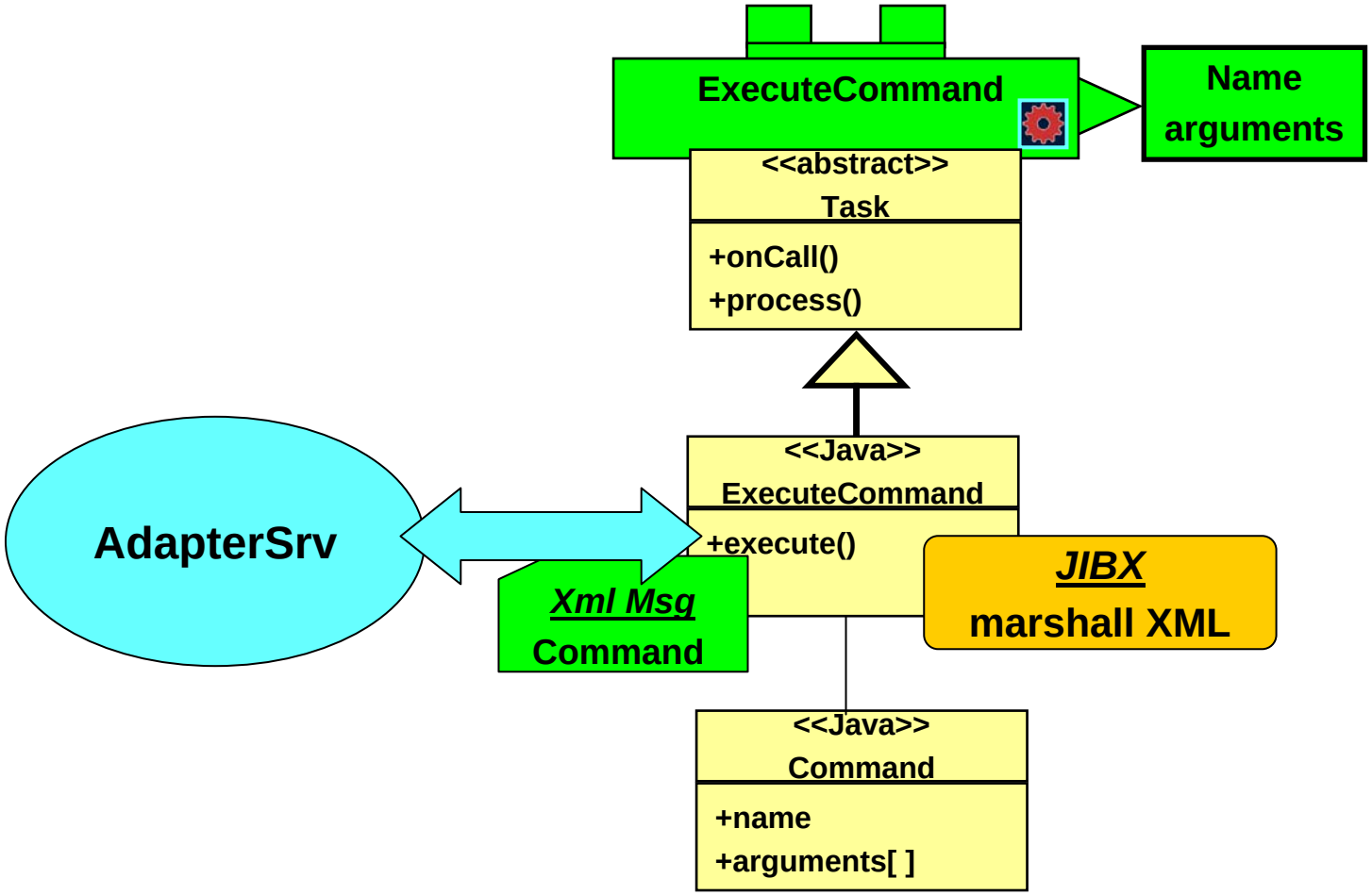


Enables Command Line Process to run as SOA Service

Legacy Adapters Pattern
uEngine Runs Command Within Action Script

Raytheon

Adapter Pattern



uEngine Can Run Command As If It Was Part of the Script

Data Notification / subscription
Enable Automation of Products and Clients

Raytheon

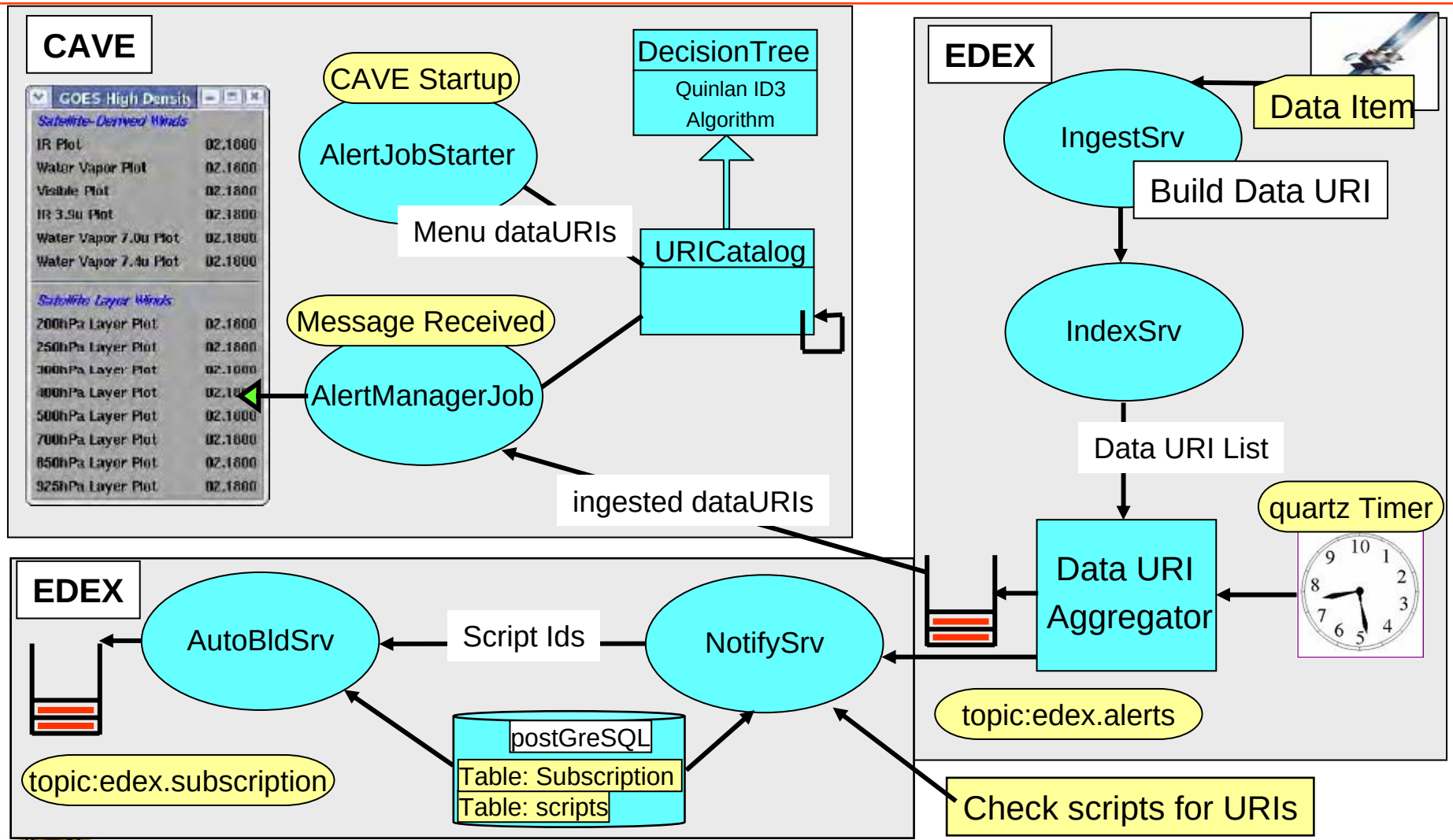
Subscription Pattern

- **Notification / Subscription Vision:** Automate product generation and client product display based on new data ingested.
- **Implementation:** Any uEngine script can be made into a subscription based on ingest of new data or a time schedule.
  - New data triggers the execution of the cached uEngine script
  - The resulting product response is placed on a JMS Topic

CAVE Data Notification + Subscription
Data Matched to Menus by Optimized Decision Tree

Raytheon

Subscription Pattern



Alerting Pattern Based on Ingest
Warning GIS located on Map and Displayed Tree

Raytheon

Subscription Pattern

ascii

File://../data/sbn/ascii

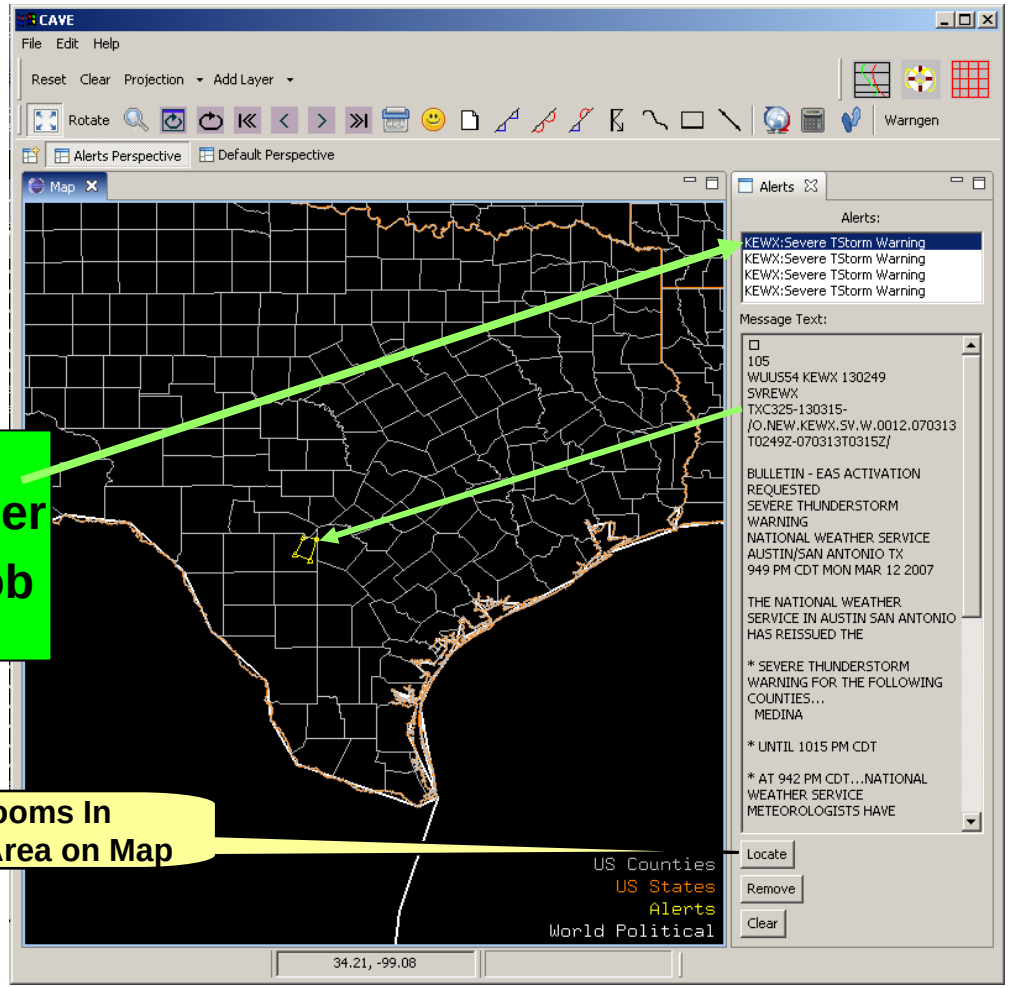
ESB RegExFilter
Fan Out Routing

Threaded Listener
AlertManagerJob

warning

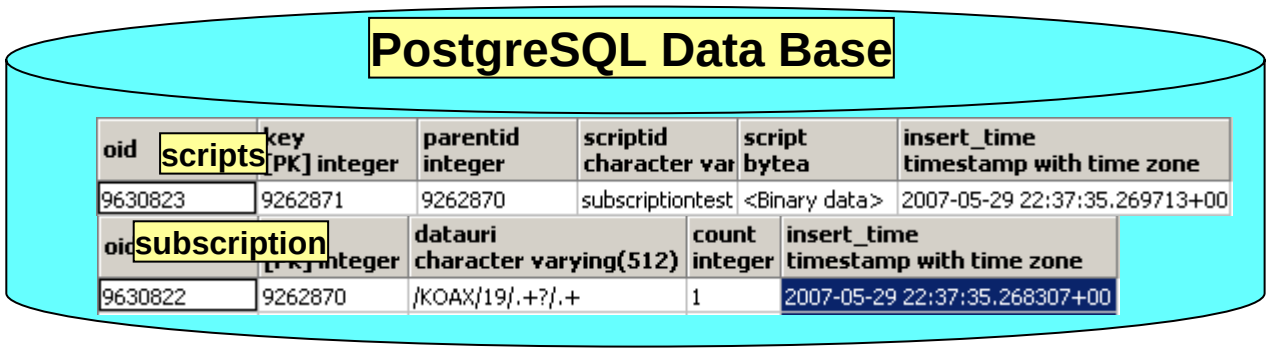
jms://cp/tstormwarning

Locate Zooms In
To Warning Area on Map



Repository Refinement

- Subscription added to PostgreSQL data base
  - Server state data simplifies clustering
- Configurable Purge Retention Period
- Performance enhancements with indexes on metadata



AWIPS-II: Architecture Topics

- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

ADE Data Model Introduction
Canonical XML SOA Interfaces Excluded

Raytheon

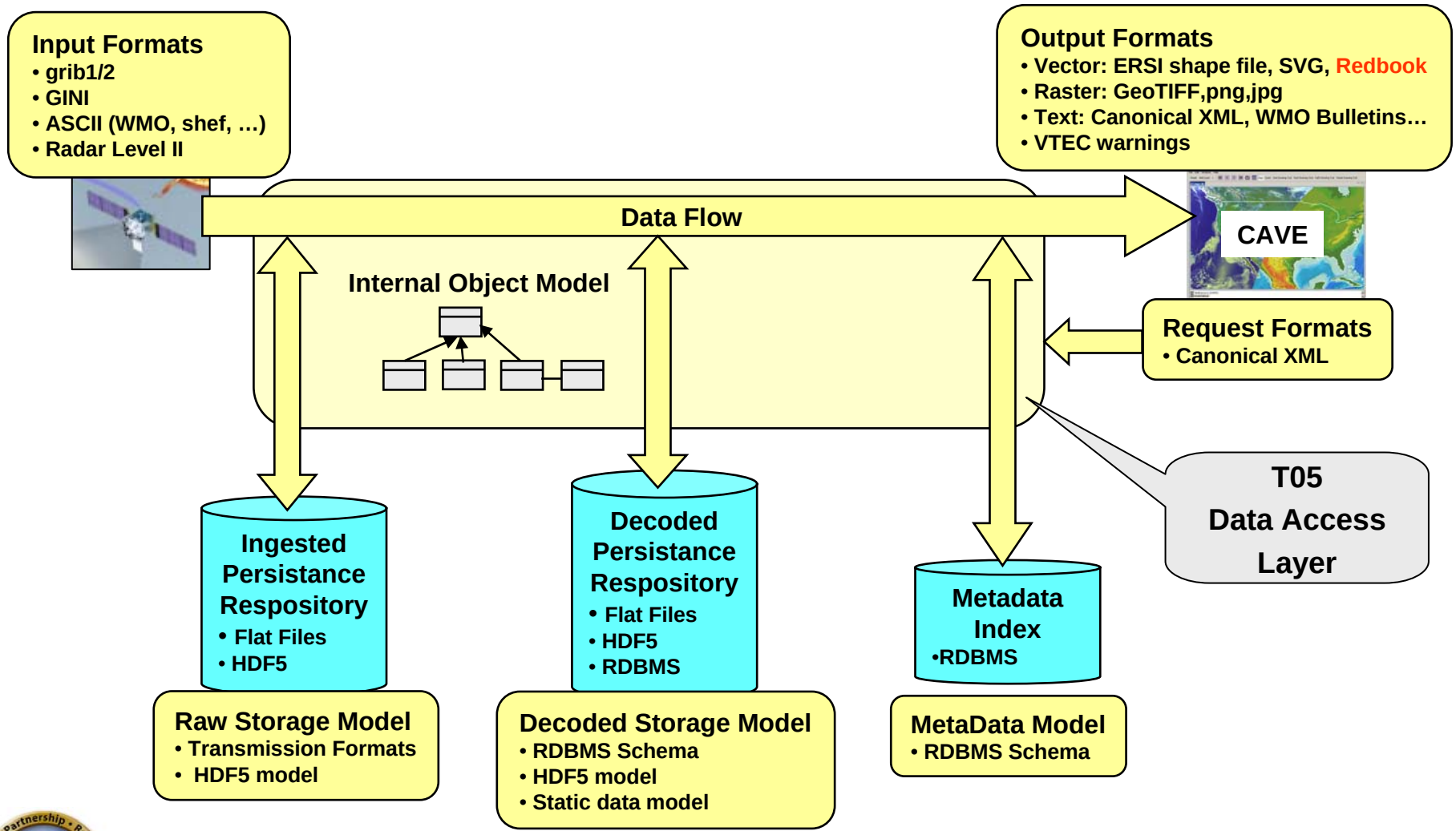
Data Model

- Data Access Layer Implementation Using Hibernate
- Data Access Object (DAO) Concept Leverages Hibernate
- Data Persistence Through HDF5 Why?
  - High performance gaming level interactions supported
  - Chucking of data records supports visualization tiling
  - Flexible retrieval supports 4D rendering
  - Streaming compression
- Meta Data Implemented in PostgisSQL through Inheritance
  - Defined only in plug ins, drives Data URI
- Base Object Model is Extended in Plug Ins
- Data URI Concept Ties Everything Together
- Purging Concept of Circularly Repository Structures

ADE Conceptual Data Model Design
I/O Formats Follows Existing Standards

Raytheon

Data Model



Data Access Layer API
Hibernate Leading Object To Relational Approach

Raytheon

Data Model

- Solves Fundamental Problem of Impedance Mismatch
- Maps between Object Model and Relational Data Model
- Provides Object based Query Facilities
- Improves Performance over JDBC, Designed for Clustering
- Reduces Code Count Improves Productivity
- Built in Support in SPRING

Hibernate Enables Meta Data Performance and Adaptability

Hibernate XML Object/ Relational Mapping
Defined in SOA Plug In: Enables Adaptability

Raytheon

Data Model

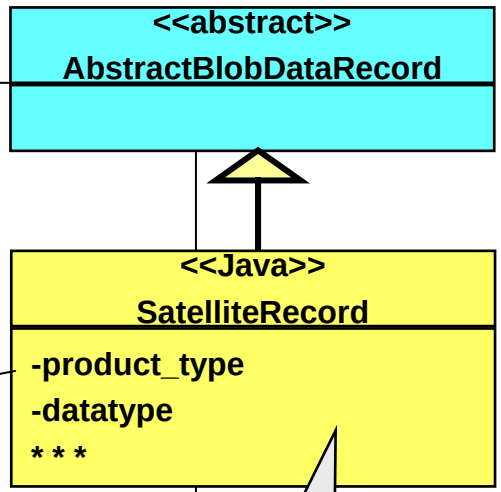
Satellite Plug In

satellite.hbm.xml

SatelliteRecord.java

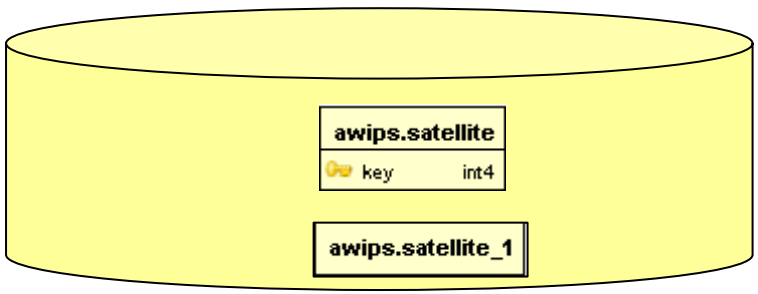
```
<hibernate-mapping default-lazy="false" schema="awips">

        <class name="com.raytheon.edex.satellite.SatelliteRecord"
                table="satellite">

                <id name="id" column="key" type="java.lang.Integer">
                        <generator class="native" />
                </id>

                <property name="dataURI" column="datauri"
                        type="java.lang.String"/>

                <property name="product_type" column="product_type"
                        type="java.lang.String" />

                <property name="datatype" column="datatype" type="java.lang.String" />
```

Data Object Model



mapping

Plug In Defined Object to Relational Mapping by Hibernate XML



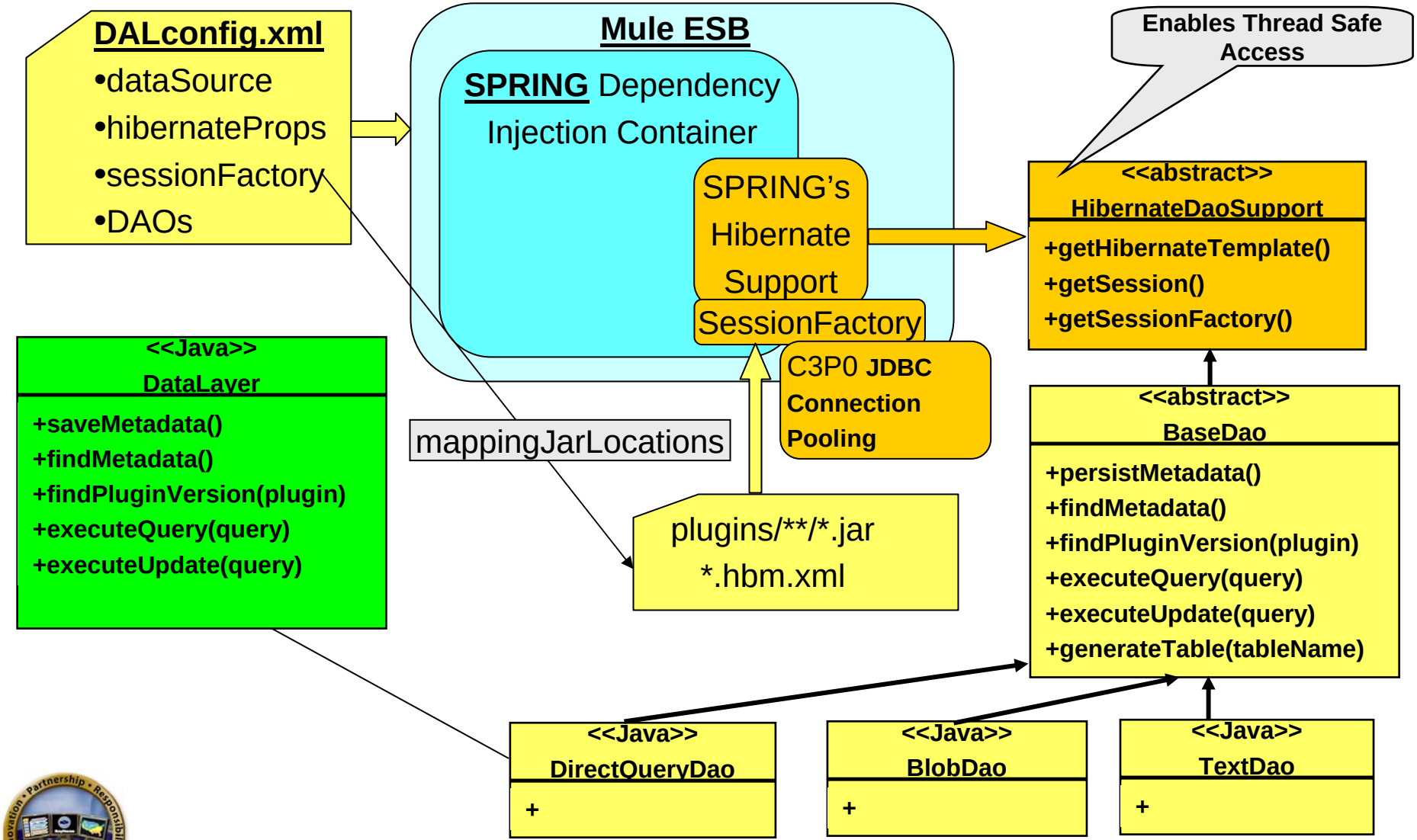
Data Object Model Extended By Plug In Follows Base Model

Plug In Enables Adapting To New Data Types

AWIPS-II: Data Access Pattern
Designed to Support Clustering

Raytheon

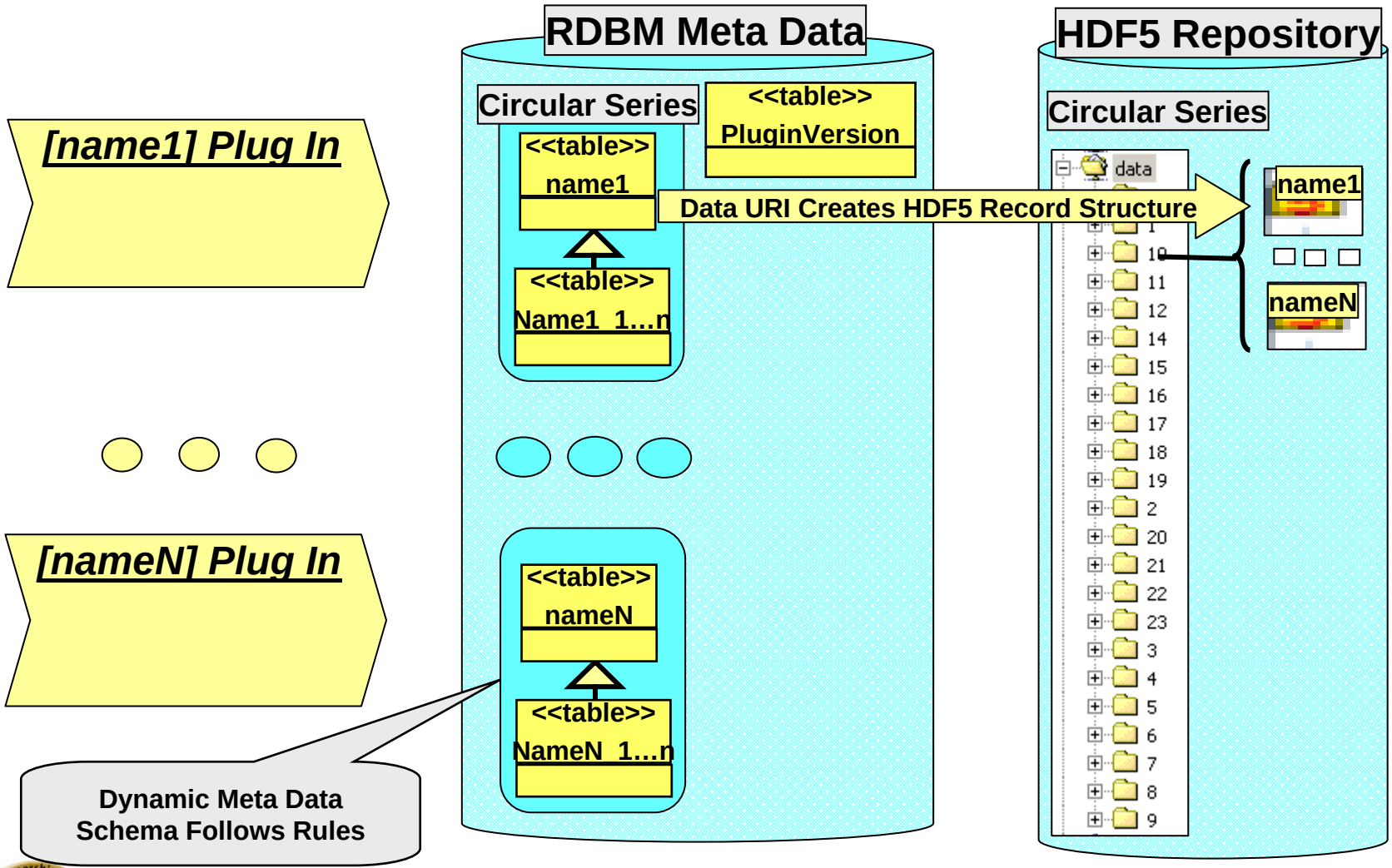
Data Model



SOA Plug In Defines a Meta Data Table Set
Each Plug In Also Defines a HDF5 set

Raytheon

Data Model

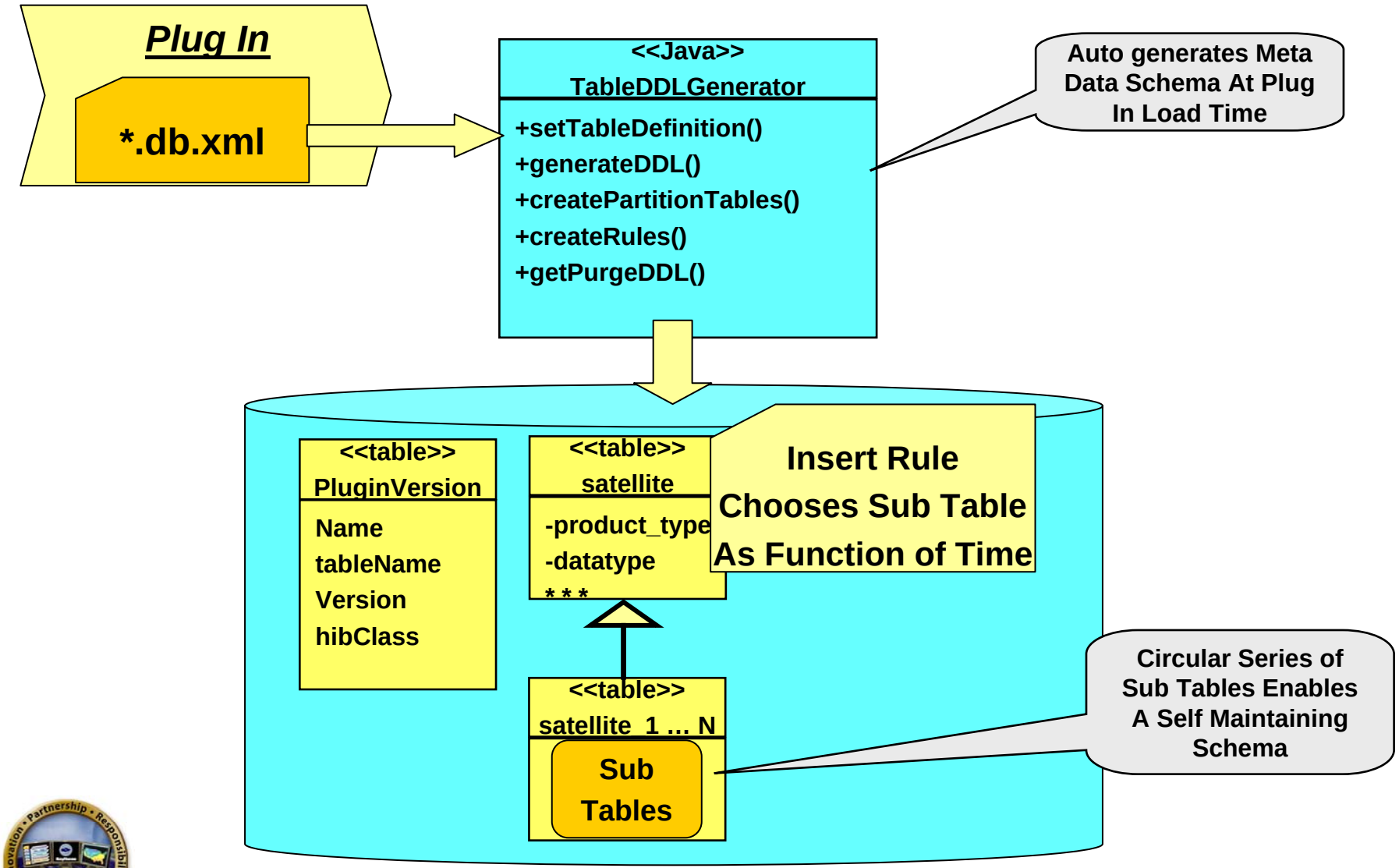


Flexible Data Model is Plug In Extendable To New Data Types

Plug In Creates New MetaData in RDBMS
Uses PostgreSQL Table Inheritance and Rules

Raytheon

Data Model

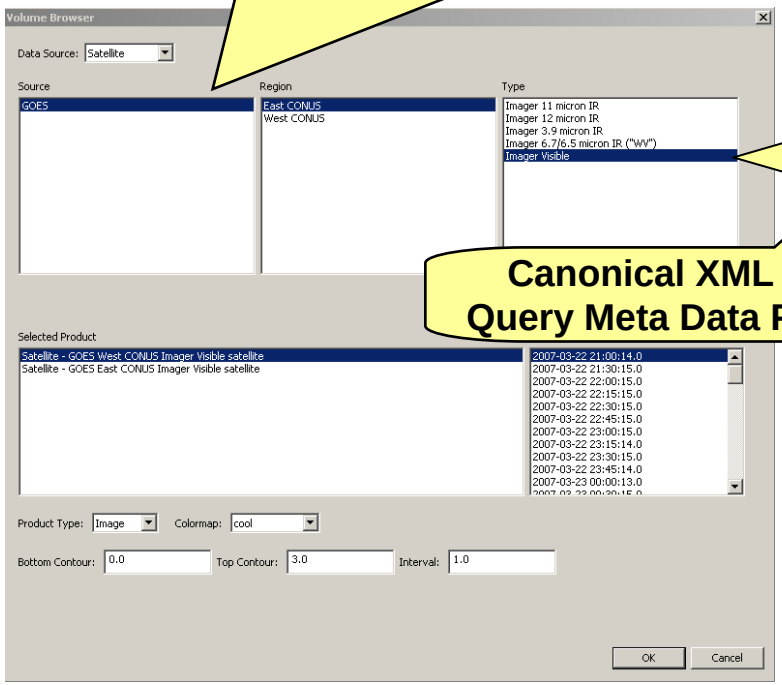


Meta Data Demo
Using CAVE's Volume Browser

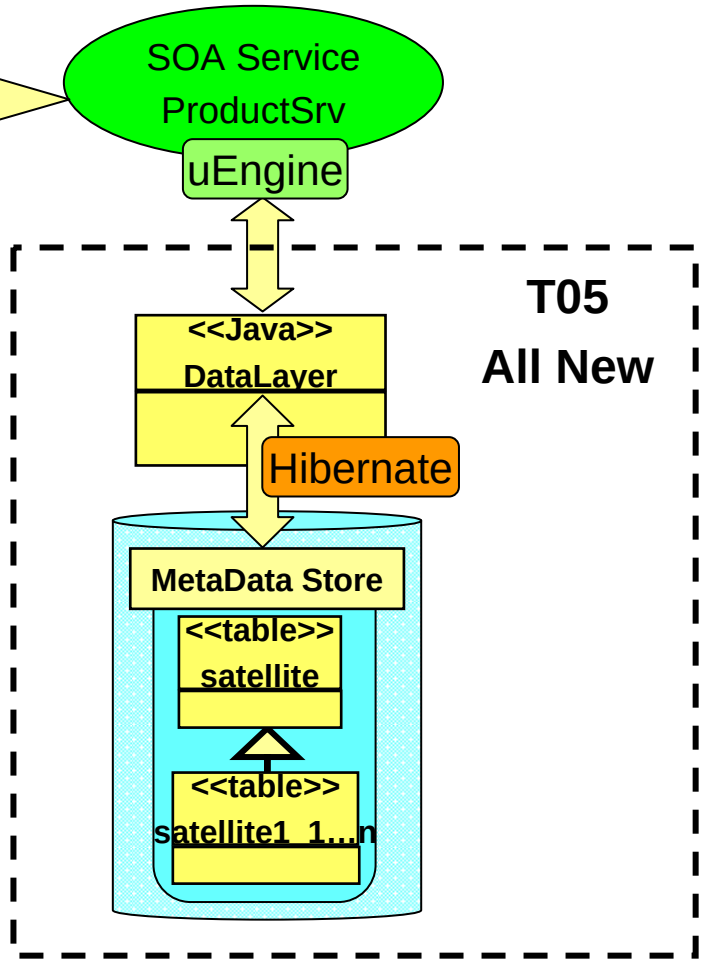
Raytheon

Data Model

Dynamically Populates Select Boxes Through Catalog Queries



Canonical XML Message
Query Meta Data For Catalog



Refines DataURI Concept
Key for System Adaptability to New Data Types

Raytheon

Data Model

- **DataURI** is a reference to data in the data store (i.e. D2D Data KEY)
- Enables **Automatic Subscriptions** For all Ingested Data
- Automatically **ties data persistence** to meta data
- Enables Plug In **Extendibility to new data types** with changing any base code
- T05 implemented a design for automatic generation of DataURIs

Meta Data Model Drives DataURI

Auto generated DataURI couples HDF5 to MetaData

Raytheon

Data Model

Satellite Plug In

satellite.db.xml

SatelliteWriter.java

```
<columnDefinition>
  <name>area_subtype</name>
  <columnType>varchar</columnType>
  <constraintType>none</constraintType>
  <precision>32</precision>
  <scale>0</scale>
  <dataURI>true</dataURI>
</columnDefinition>
```

○○○

XML Meta Data Definition File Tag Specifies If Element is Part of Data URI

Auto Generates Data URI references from XML Definition And Meta Decode of Ingested Record

IngestSrv

<<Java>>
Plugin

+getDataURI()
+getMatchURI()

<<Java>>
SatelliteWriter

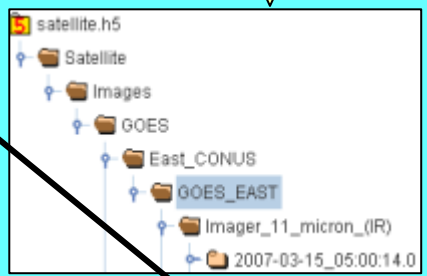
Example Data URI From a LOG File

```
dataURI = [/7/0/86/211/2007-03-15_12:00:00.0/Total_precipitation-3_hour/surface/surface/9]
```

RDBM Meta Data

<<table>>
satellite

HDF5 Repository



Plug In Enables Adapting To New Data Types

Data Persistence Using HDF5

HDF5 Files In Time Ordered Bins Like MetaData

Raytheon

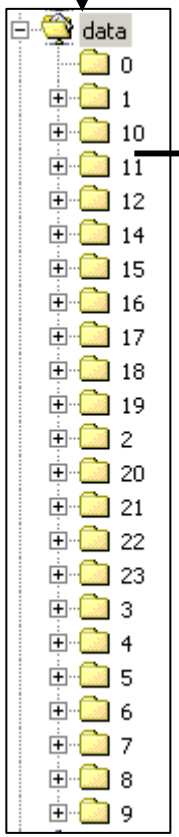
Data Model

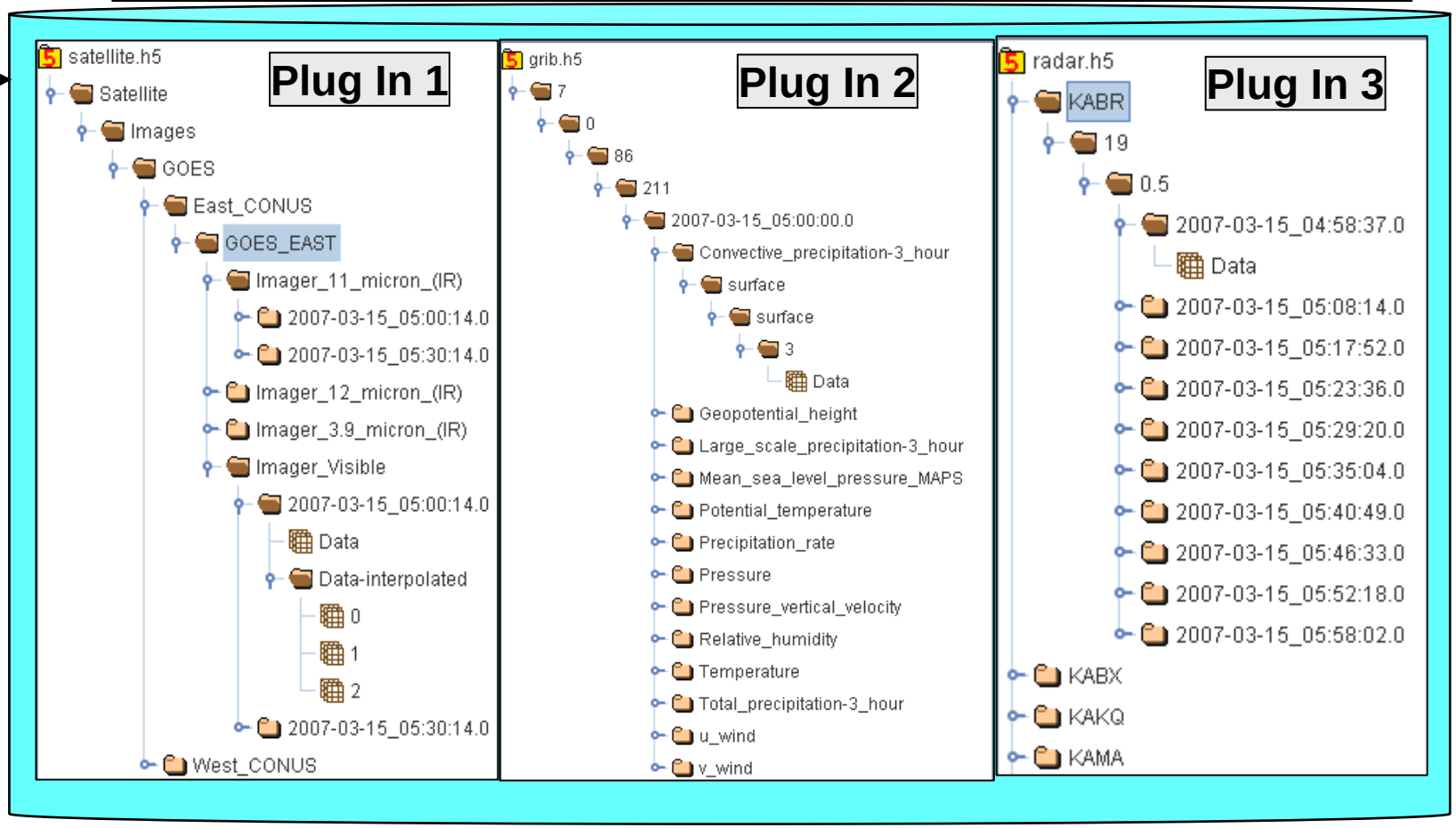
Circular Time Bins

/awips/opt/data/hdf5

i.e. Autogenerated dataURI ties Metadata to HDF5 Record

```
dataURI = [/7/0/86/211/2007-03-15_12:00:00.0/Total_precipitation-3_hour/surface/surface/9]
```

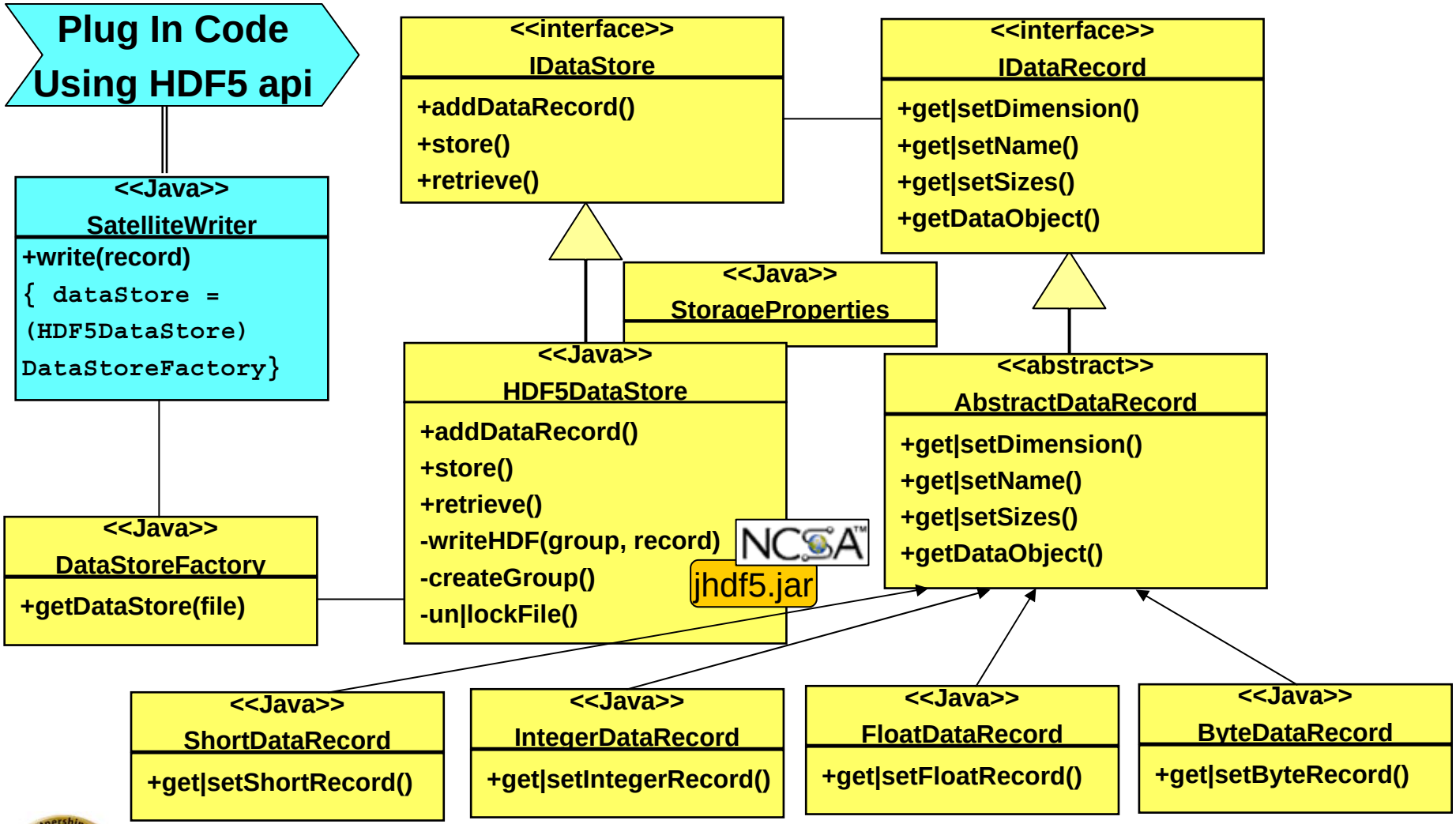




Data Persistence Using HDF5
Application Code Interfaces Through API

Raytheon

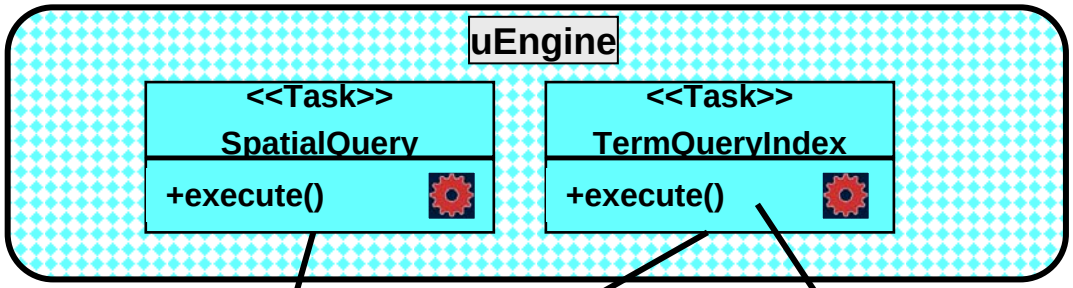
Data Model

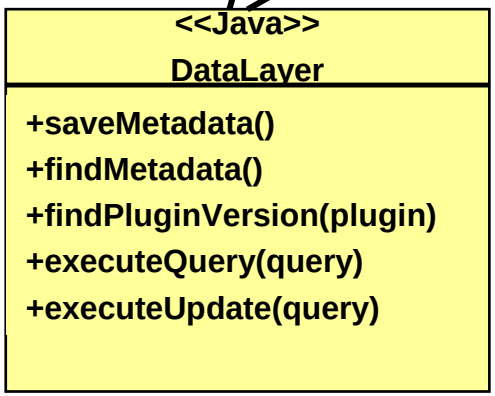


uEngine Using the Data Access Layer
Single API enables uEngine to access all data

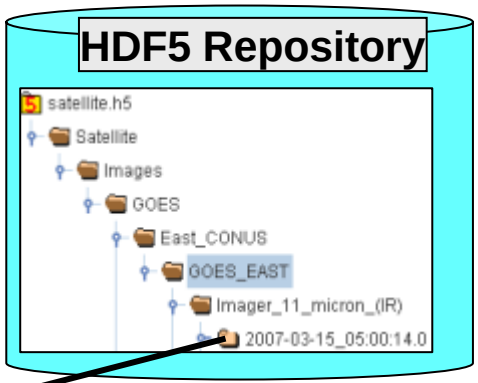
Raytheon

Data Model






```
Results =
dataLayer.findMetaData
(plugin,
 fields[],
 operands[],
 values[],
 sort, count)
createMetaDataMap()
```



Dynamically
Creates URI Reference
to
Data Record

AWIPS T010 Improved Purging Flexibility

Raytheon

More flexible metadata purging

Architecture

Added new interface to plug-in pattern to enable specific plug-ins to have custom purging rules

Default plug-in purging based on retention time stored in the plug in registry table

Quart Timer Once / hour

PurgeSrv

<java> DefaultPurgerImpl

name [PK] characte	database character var	initialized boolean	retentiontime integer	tablename character var	version real
airep	metadata	TRUE	24	airep	1
binlightning	metadata	TRUE	24	binlightning	1
bufrmos	metadata	TRUE	24	bufrmos	1
bufrua	metadata	TRUE	24	bufrua	1
ccfp	metadata	TRUE	24	ccfp	1
gfe	metadata	TRUE	24	gfe	1
goessounding	metadata	TRUE	24	goessounding	1
grib	metadata	TRUE	24	grib	1
modelsounding	metadata	TRUE	24	modelsounding	1
obs	metadata	TRUE	24	obs	1
pirep	metadata	TRUE	24	pirep	1
poessounding	metadata	TRUE	24	poessounding	1
profiler	metadata	TRUE	24	profiler	1
radar	metadata	TRUE	24	radar	1
recco	metadata	TRUE	24	recco	1
redbook	metadata	TRUE	24	redbook	1
satellite	metadata	TRUE	24	satellite	1
sfcobs	metadata	TRUE	24	sfcobs	1
shef	ihfs	TRUE	24	<unspecified>	1
taf	metadata	TRUE	24	taf	1
text	fxa	TRUE	24	<unspecified>	1
warning	metadata	TRUE	24	warning	1

<plugin registry> Plug_Info Table

<java> GfePurger

HDF5 directory tree

30 day & hour

1
satellite.h5
radar.h5
grib.h5
binlightning.h5

2
satellite.h5
radar.h5
grib.h5
binlightning.h5

Delete Referenced HDF5 files

GFE Plug In

AWIPS-II: Architecture Topics

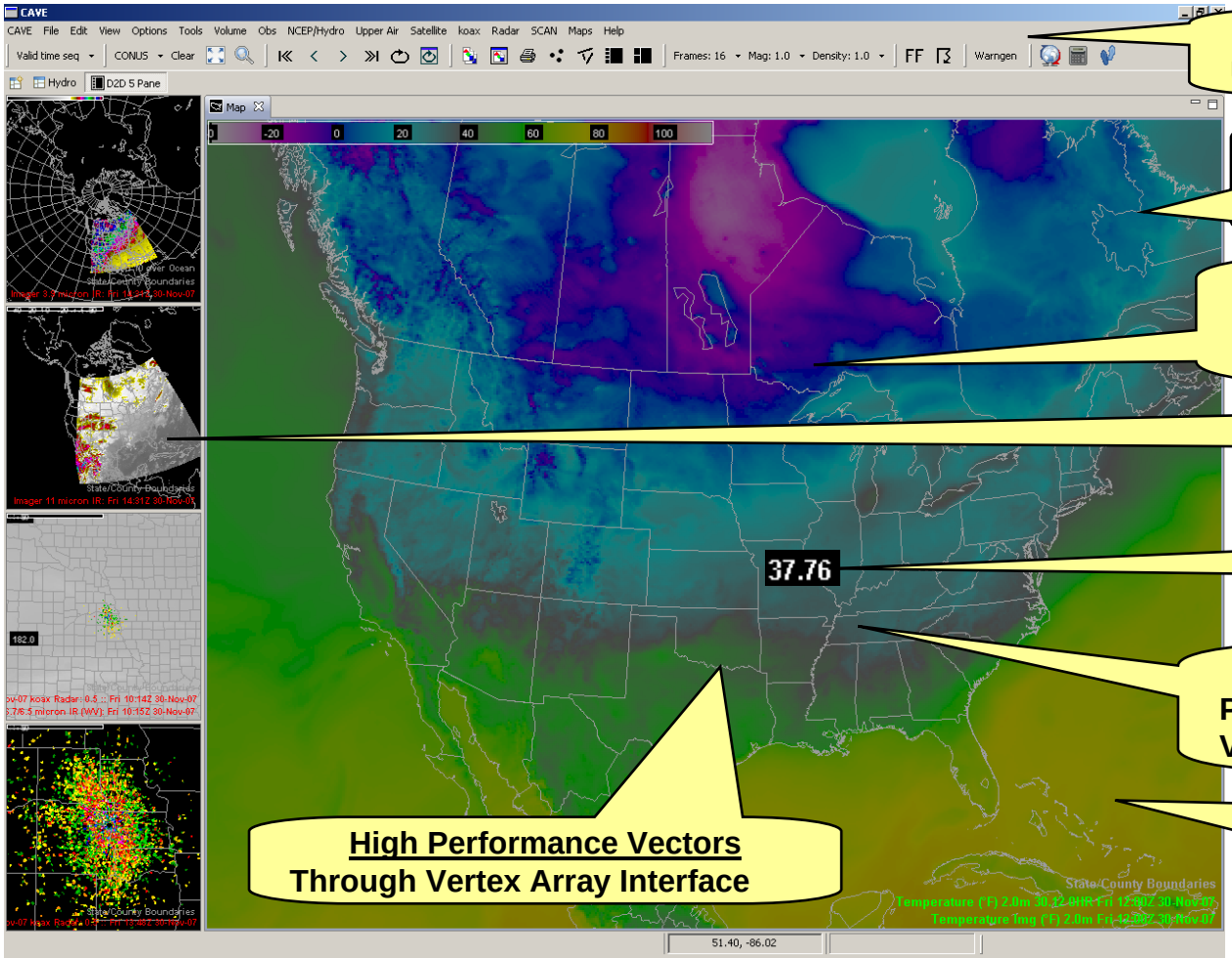
- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

AWIPS-II: Visualization Overview (CAVE)

Built from Eclipse RCP Plugins + is Extensible

Raytheon

Visualization



Eclipse RCP 3.4 XML
User Localized Menus & Tool bar

GIS Display
All data layers map projected, scriptable, and subscribable

High Performance Image Rendering
GPU Shader Language Controls of (Projection,Color, …)

QUAD Tiling of Raster Data
World Wide Google style zoom/pan

Data Sampling/Analysis
Through raw data load

GIS Whiteboard Collaboration
Projection independent whiteboard, VoIP, Chat

Animation of All Data
All displayed data automatically under subscription

High Performance Vectors
Through Vertex Array Interface

Thick/Thin Client
Through ESB Endpoint

Enterprise Service Bus Data Connections

Advanced GIS Visualization Enables Gaming Interactions

CAVE Enhancements
Most Significant Update Since Project Began

Raytheon

Visualization

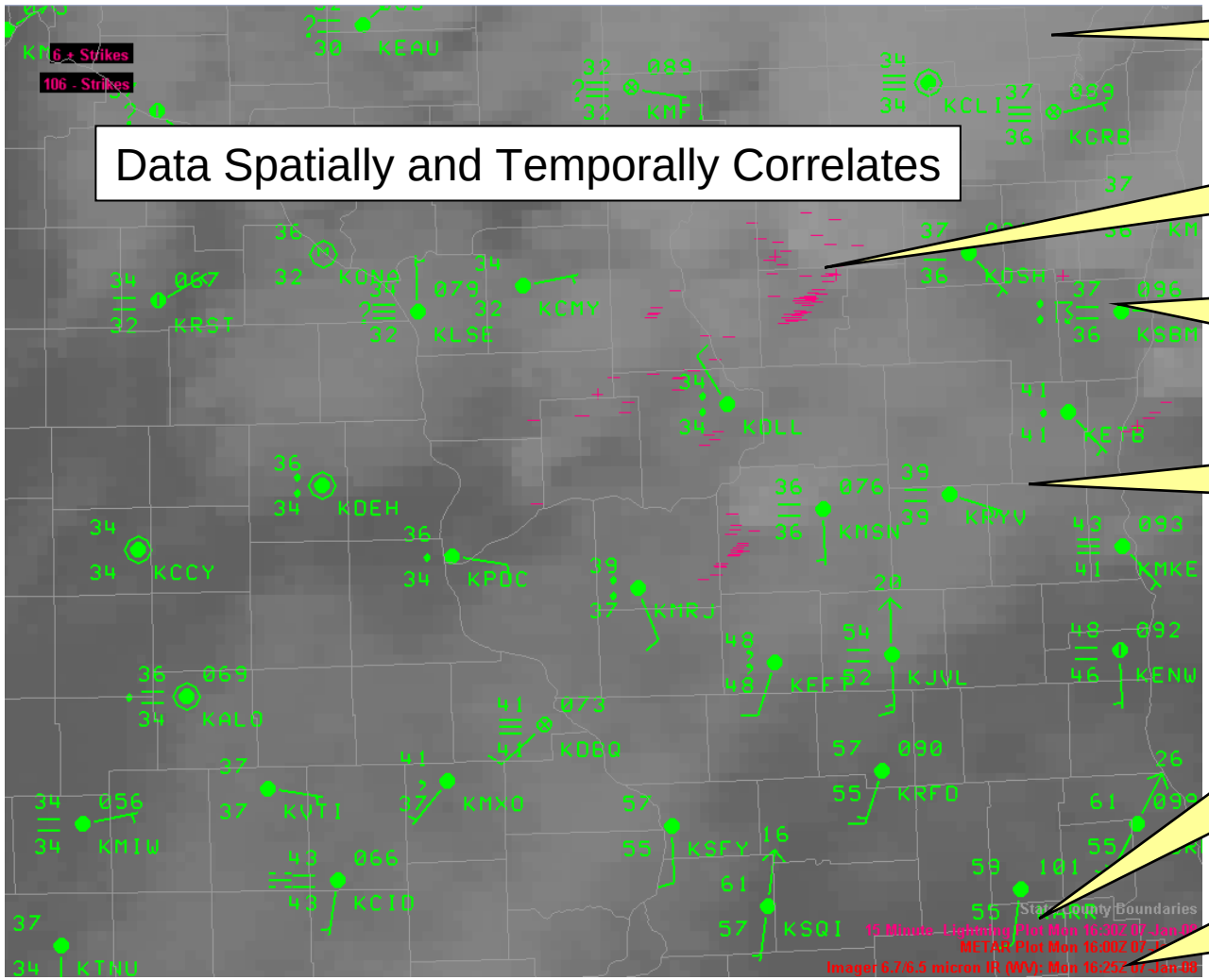
- Eclipse RCP 3.4 based, GeoTools 2.4, JOGL 1.1, Velocity, and batik
- Entirely new raster rendering implementation based on raw float data
- New raster tiling concept using HDF5
- New color map rendering concept
- New GPU mechanism for map reprojection
- Implemented Display Bundles
  - JIBX extension for Resource to Bundle mapping enables quick adaption to new visualization resource
- High Performance radar rendering similar to D2D all-tilts
  - On demand rendering
- Active raster pixel data integration
- Warning generation with “Velocity” parsing of templates

AWIPS-II: Example Data Fusion in CAVE

GIS display of raster, point, and vector data

Raytheon

Visualization



Weather Satellite
QUAD tiled display raster data

Lighting Strikes
High frequency point data converted to vector display

Point Weather Observations
Text base observation converted to vector display

Geo Political Maps
Shape File rendering for ERSI Map data

Complete Layer Controls
Order, Color, transparency, visibility, Active Layer

Automatic Subscriptions
All displayed data automatically under subscription and animates

Automatic Data Fusion for All Ingested Data

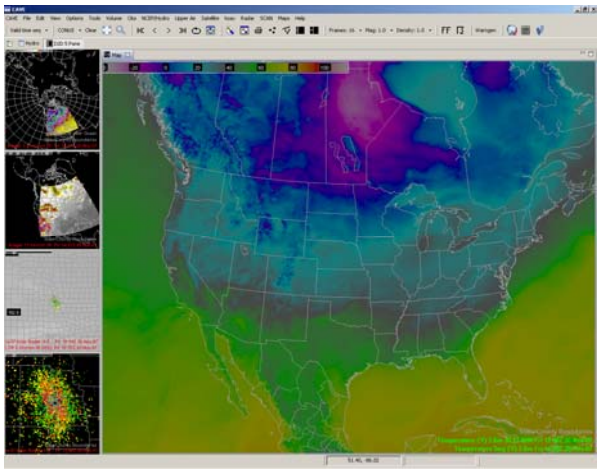
AWIPS-II: Leverages Eclipse Perspectives

Enables Multiple User Interfaces

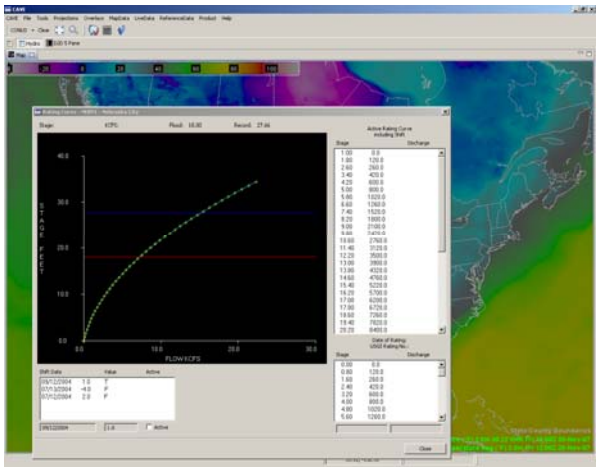
Raytheon

Visualization

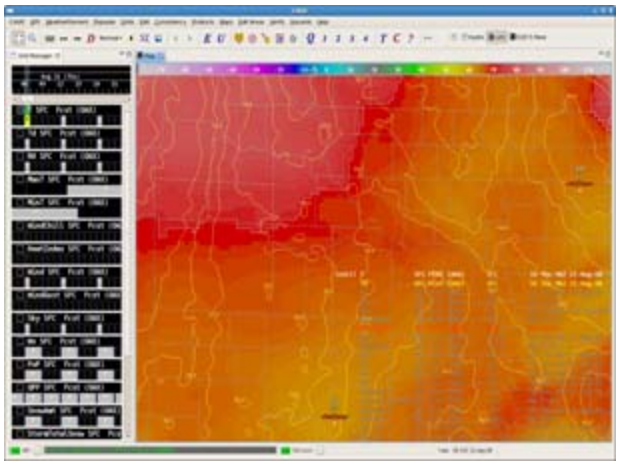
Weather Workstation



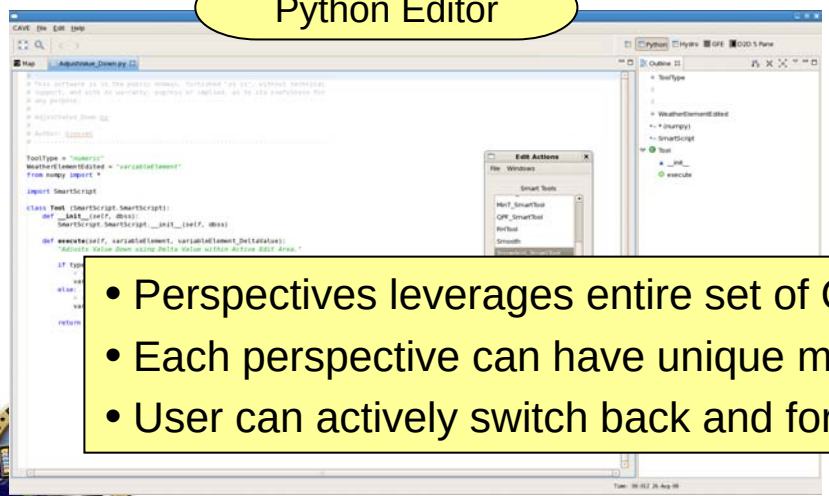
Hydrology Workstation



Graphical Forecast Editing



Python Editor



- Perspectives leverages entire set of CAVE eclipse plug ins
- Each perspective can have unique menus, toolbars, and dialogs
- User can actively switch back and forth between perspectives

ESB

Localization

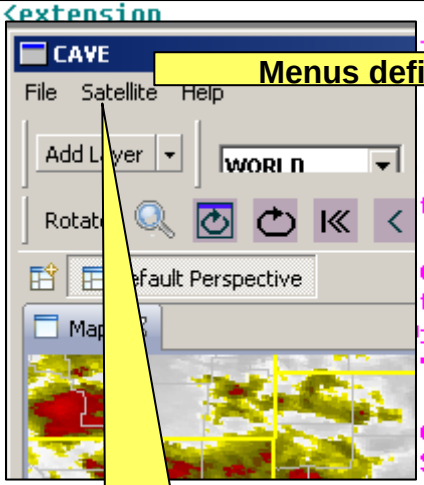
UtilitySRV

XML Driven Menus
Color Tables
Scripts + Config
Maps…

CAVE Menu Extensibility – Eclipse 3.3
Configuration Preferences, Menus, Data …

Raytheon

Visualization



Menus defined in plug in xml

plugin.xml

```
<extension
                                  i_menus">
                                  :org.eclipse.ui.main.me
                                  te">
                                  com.raytheon.viz.ui.loadBundle"
                                  theon.viz.sat.irWindow"
                                  indow"
                                  ">
                                  oductURI"
                                  Satellite/Images/GOES/East_CO
    </parameter>
    <parameter
          name="productName"
          value="IR Window">
    </parameter>
    <parameter
          name="bundleLocation"
          value="etc/bundles/IRWindow.xml">
    /parameter>
    mand>
```

CAVE satellite plug In

bundle.xml

```
<mapDescriptor>
 <resource class="com.raytheon.viz.sat
  <properties>
   <value field="colorMap">COLORMAP { ir_default }</value>
   <value field="colorMapMin">0.0</value>
   <value field="colorMapMax">255.0</value>
   <value field="contrast">1.0</value>
   <value field="brightness">1.0</value>
   <value field="interpolationState">false</value>
   <value field="requestScript">include("SatelliteRequest.js");
var dataRequest = new SatelliteRequest();
dataRequest.setCount(8);
dataRequest.addParameter("satellite","GOES");
dataRequest.addParameter("parameter","Imager 11 micron IR");
dataRequest.setSortValue("valid_time");
dataRequest.addParameter("area_subtype","East CONUS");
dataRequest.execute();</value>
  </properties>
  <resourceProperties>
   <isVisible>true</isVisible>
   <opacity>1.0</opacity>
   <isLabeled>true</isLabeled>
   <isHoverOn>false</isHoverOn>
  </resourceProperties>
```

references bundle

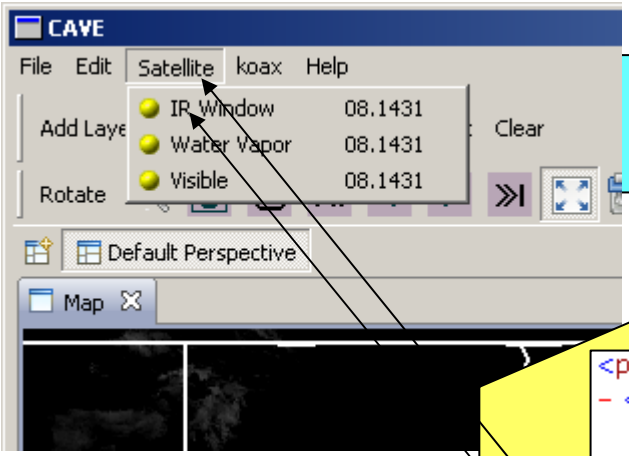
Menu fields update
Automatically through the AlertManagerJob as data is ingested

Embedded uEngine Script
Retreives and transforms data for display

CAVE Menu Extensibility Pattern
Leverages Capability in Eclipse 3.3

Raytheon

Visualization



CAVE satellite Plug In

plugin.xml

```
<plugin>
- <extension point="com.raytheon.viz.core.resource">
    <resource class="com.raytheon.viz.satellite.rsc.SatResource"
      factoryClass="com.raytheon.viz.satellite.rsc.SatResourceFactoryAdapter" nam
      recordClass="com.raytheon.edex.plugin.satellite.SatelliteRecord" />
  </extension>
- <extension point="org.eclipse.ui.menus">
  - <menuContribution locationURI="menu:org.eclipse.ui.main.menu?after=edit">
    - <menu id="satellite" label="Satellite">
      - <command commandId="com.raytheon.viz.ui.loadBundle" id="com.raytheon.v
          <parameter name="productURI" value="/Satellite/Images/GOES/East_CONU
          <parameter name="productName" value="IR Window" />
          <parameter name="bundleLocation" value="etc/bundles/IRWindow.xml" />
        </command>
```

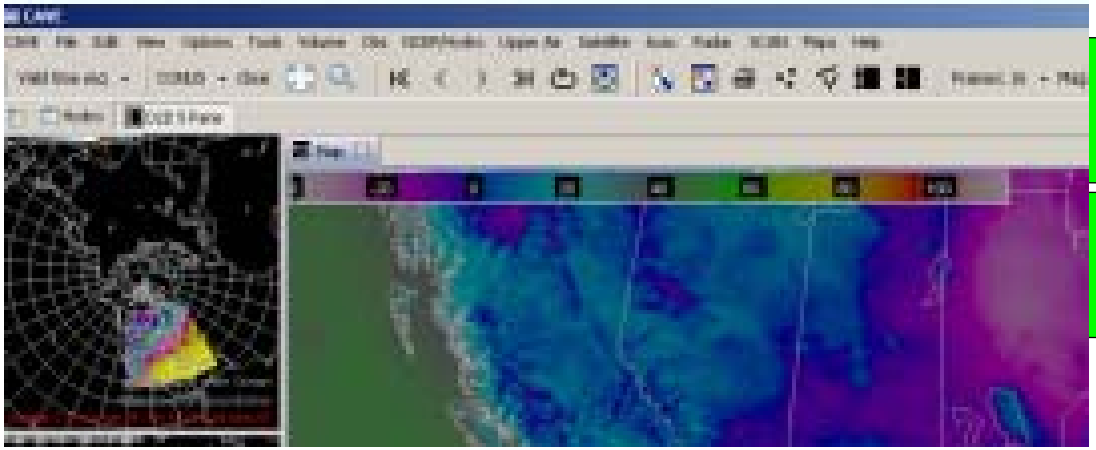
Bundle IRWindow.xml

uEngine Script to Retrieve loop of data

Workstation Menus / Toolbars / Rendering
XML Configurable Menus + XML Bundles

Raytheon

Visualization



.ui.personalitles.awips plugin.xml

.site. (from localization) plugin.xml

XML Display Resource Bundles

Commands

plugin.xml
Commands
Parameters
Menu labels

setScale

RadarDisplayControls

menuRetrieval

productURI
productName
variableList

~19 Map

~11 Plots

1 Skew-T

8 Radar

11 Models

~30 Display Resources

CAVE Display Bundles
Enables Adding Procedures and Display History

Raytheon

Visualization

XML Document

Bundle Definition

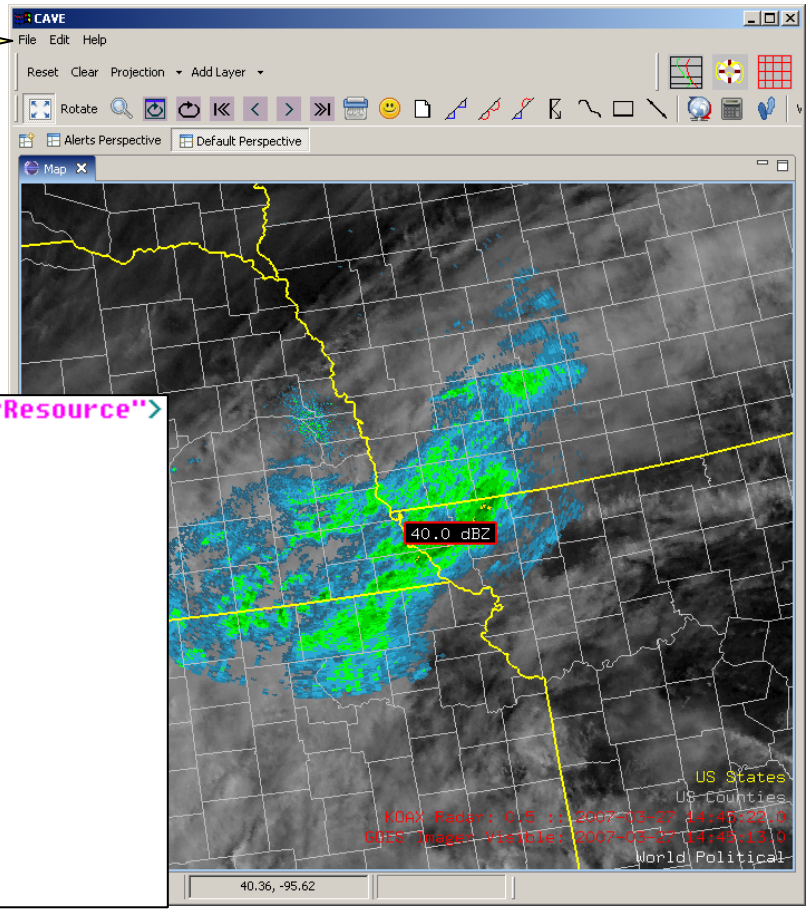
+layers

+color maps

+line widths

****

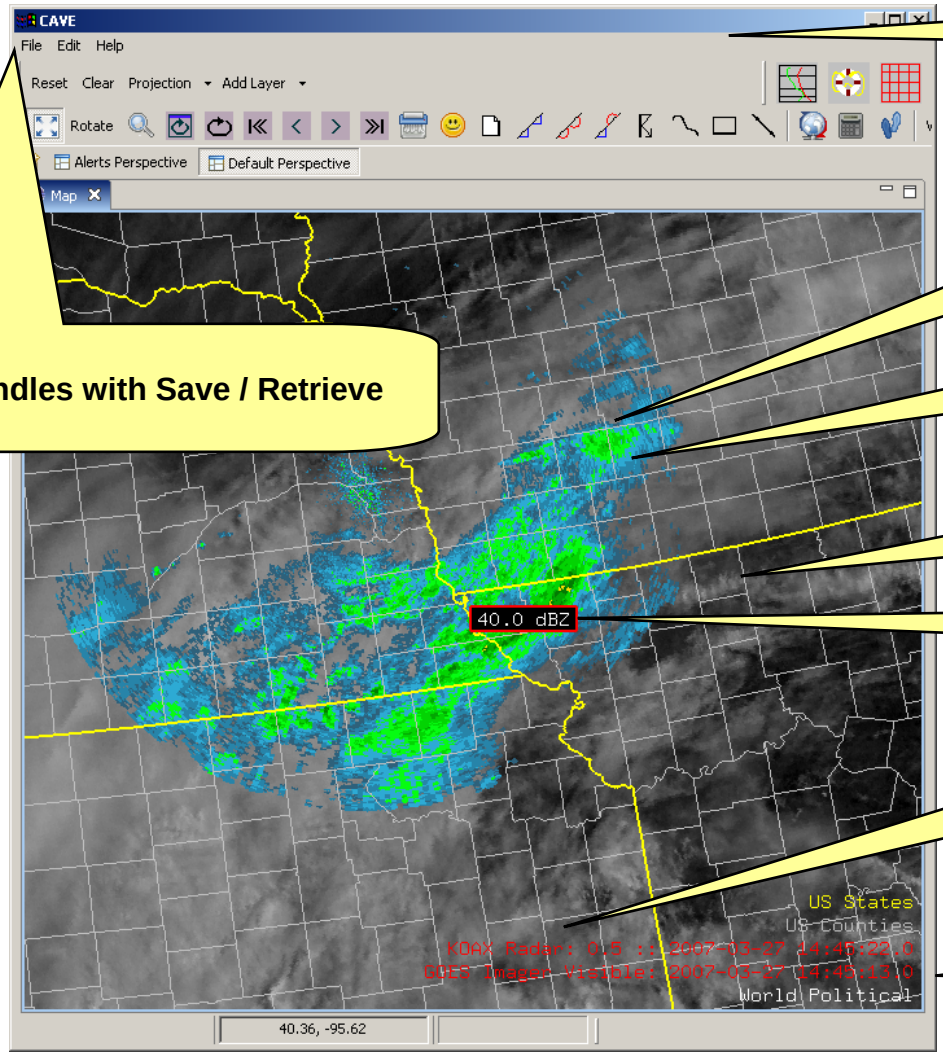
```
<resource class="com.raytheon.viz.core.rsc.radar.RadarResource">
 <properties>
  <value field="colorMap">COLORMAP { radar }</value>
  <value field="colorMapMin">0.0</value>
  <value field="colorMapMax">255.0</value>
  <value field="contrast">1.0</value>
  <value field="brightness">1.0</value>
  <value field="interpolationState">false</value>
 </properties>
 <resourceProperties>
  <isVisible>true</isVisible>
  <opacity>0.925</opacity>
  <isLabeled>true</isLabeled>
  <isHoverOn>true</isHoverOn>
 </resourceProperties>
</resource>
```



CAVE Visualization – Service Endpoint
Enables Gaming Style Data Interactions

Raytheon

Visualization



Eclipse RCP 3.2 – Plug In Extendable
Plug In for Warn Generation Added

Radar Rendering Uses
Dynamic raster tiling

Added Bundles with Save / Retrieve

GPU Shader Language Rendering
Controls (Color, …), Animation

New Quad tiling of large raster
Sets leverages HDF5 chunking

Active Raster Data
Interrogation

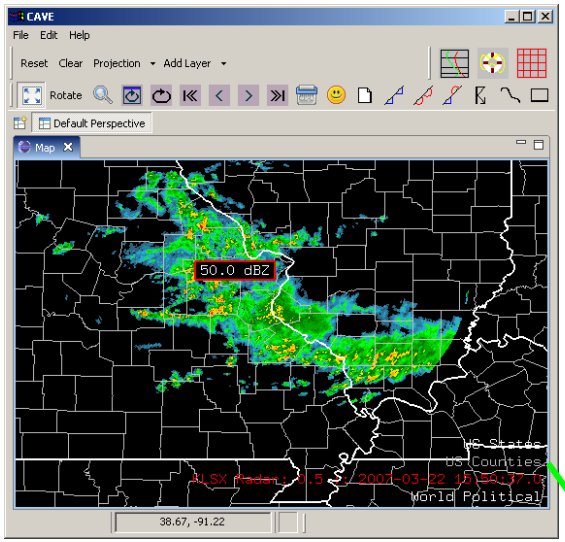
Dynamic map reprojection using
GeoTools Transforms& GPU Warping

All Tilts Keyboard Controls

CAVE Radar Rendering
Radial Data Gets Tiled Dynamically

Raytheon

Visualization



<<interfaces>>

- ICAVEResource
- IColormappableResource
- IImagingResource
- ITimeSeqResource
- IVertSeqResource
- IToggleableResource
- IInspectableResource
- IBlendableResource

<<java>> RadarTileSet

- +createTile()
- +dispose()

<<java>> RadarResource

- +paint()
- +init()
- +dispose()
- +***

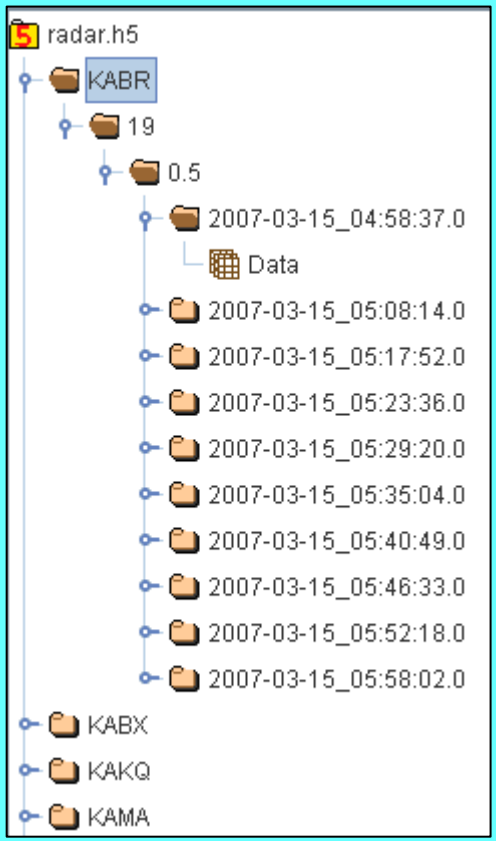
<<Job>> RadarLoaderJob

- +requestLoad()
- +run()

<<java plug in>> RadarTiler

- +createTile()
- +createFullImage()
- +calculatePixels()
- +setPixel()

HDF5 Radar Data

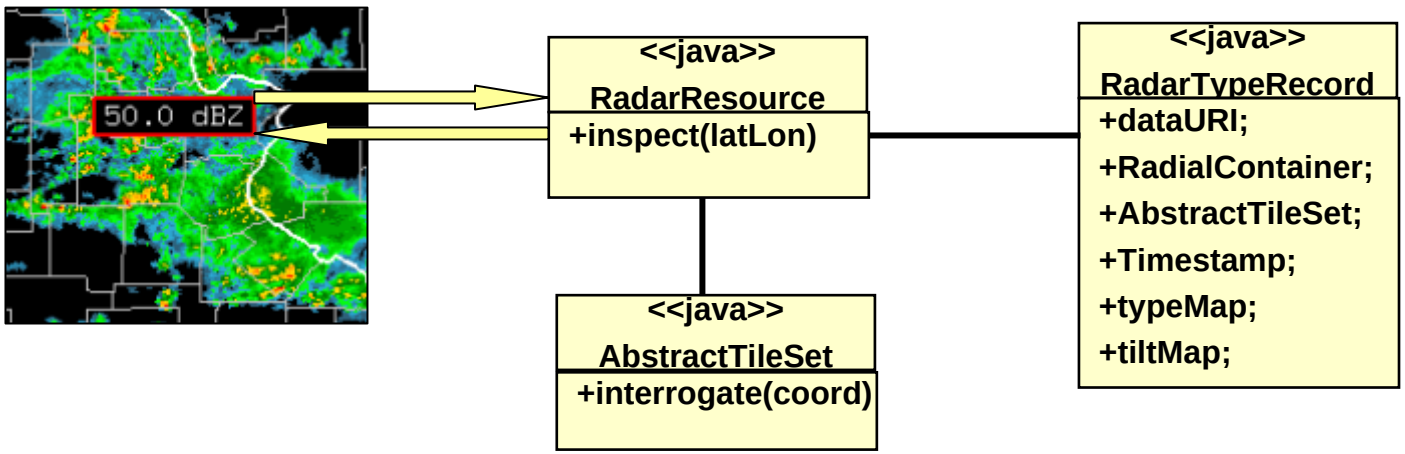


CAVE Raster Data Interrogation
All raster and shape files are inspectable

Raytheon

Visualization

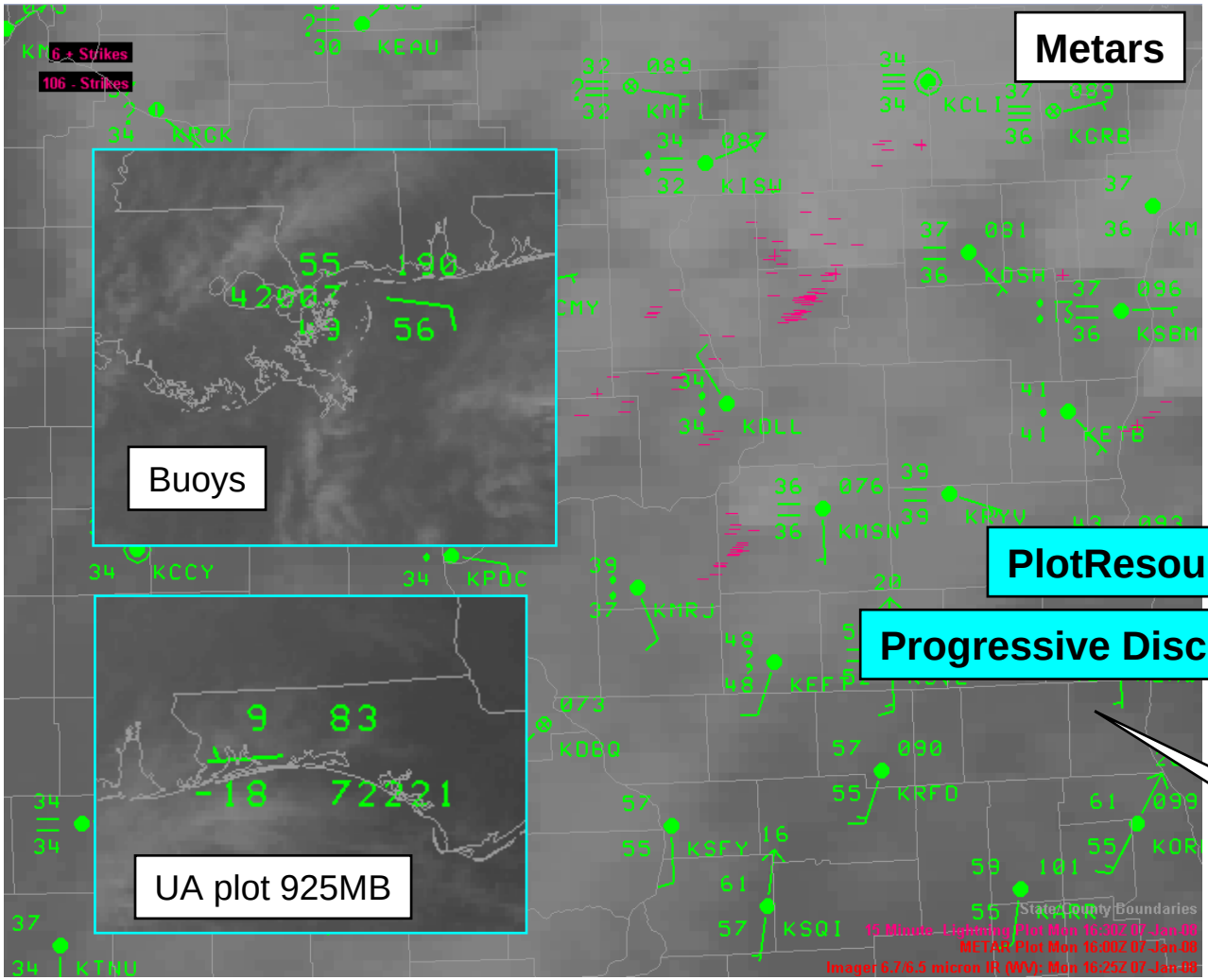
- Mouse X,Y Events are Coordinate Converted to Lat/Lon
- The Lat/Lon location is transformed to retrieve the data out of the correct displayed tile at the x, y location
- The data is transformed into displayable units



Workstation Plot Model and Maintenance
Plot Vector Library Converted to SVG (XML)

Raytheon

Visualization



Obs MetaData & Parsed data fields

{…} Plug In

Job

PlotModel Generator

PlotResource

Progressive Disclosure

SVG Plot Models

Entire present Wx symbol set converted to SVG

Workstation Warning Generation

T08 Delivered | T09 Delivered | T010+ Delivered

Raytheon

Visualization

WarnGen GUI

Life Cycle Rules

tornado.vm

severethunderstorm.vm

Templates 3..19

GIS Capabilities

Warning by PolyGon

Dissemination

VTEC Database

TextDB

Store

Set VTEC

Generate Text

Send to TextWS

Text Workstation Editor

UPDATE LIST

Severe Weather Statement

Partial: KOAX Full: KOAX

Track type: One Storm / Line of Storms

Edit: Box / Track / Box and Track

Redraw Box on Screen from: Track | Warned/Hatched Area | Warned Area Visible

Product type: Flash Flood / Severe Thunderstor / Tornado

Time Range: Duration 30 | 14:22 Tue 8-Jan | 14:52 Tue 8-Jan

********** BASIS FOR WARNING (CHOOSE 1) **********
Doppler radar indicated
Trained spotters reported
public reported
local law enforcement reported
********** THREAT (CHOOSE 1 each wind/hail) **********
60 mph wind
70 mph wind
80 mph wind
Penny size hail
Nickel size hail
Quarter size hail
Golf ball size hail
Baseball size hail
********** CALL TO ACTIONS (CHOOSE 1 OR MORE) **********
TOR WATCH IN EFFECT...svr tstms can produce tornadoes
Svr t-storms can produce tornadoes...
This is a dangerous storm...
Go to lowest floor...potential for sig injury
Lightning in addition to hail/winds
Over Lake - Boaters seek shelter
History of producing damaging winds/large hail
History of widespread wind damage
Report of damaging winds- edit win speed reported

Instructions:
Instructions go here

Restart | Cancel | OK

TextDB: nodedistance, statematch, stdtextproducts, textproductinfo, versionstable, watchwarn

VTEC Database: ae_index, afos_product, ap_index, application, class, ep_index, event_tracking, events, flood_record_status, flood_severity, immediate_cause, phenomena, significance, static

Text Warngen: OMARMEOMA (Now Editing)

File Edit Options Version Tools Scripts Products Help

AFOS Browser | Load History | WMO Search | Enter Editor | Accum

AFOS Cmd: | WMO TTAAii CCCC: | AWIPS ID:

TTAA00 CCCC DDHHMM
RMEOMA

Save | Cut | Copy | Paste | Fill | Edit Header | Send | Cancel | Attach | INS

/O.NEW.WMOID123.TO.W.0001.080108T2209Z-080108T2239Z

BULLETIN - EAS ACTIVATION REQUESTED
SEVERE THUNDERSTORM WARNING
NATIONAL WEATHER SERVICE OMAHA/VALLEY NE
1010 PM GMT TUE JAN 8 2008

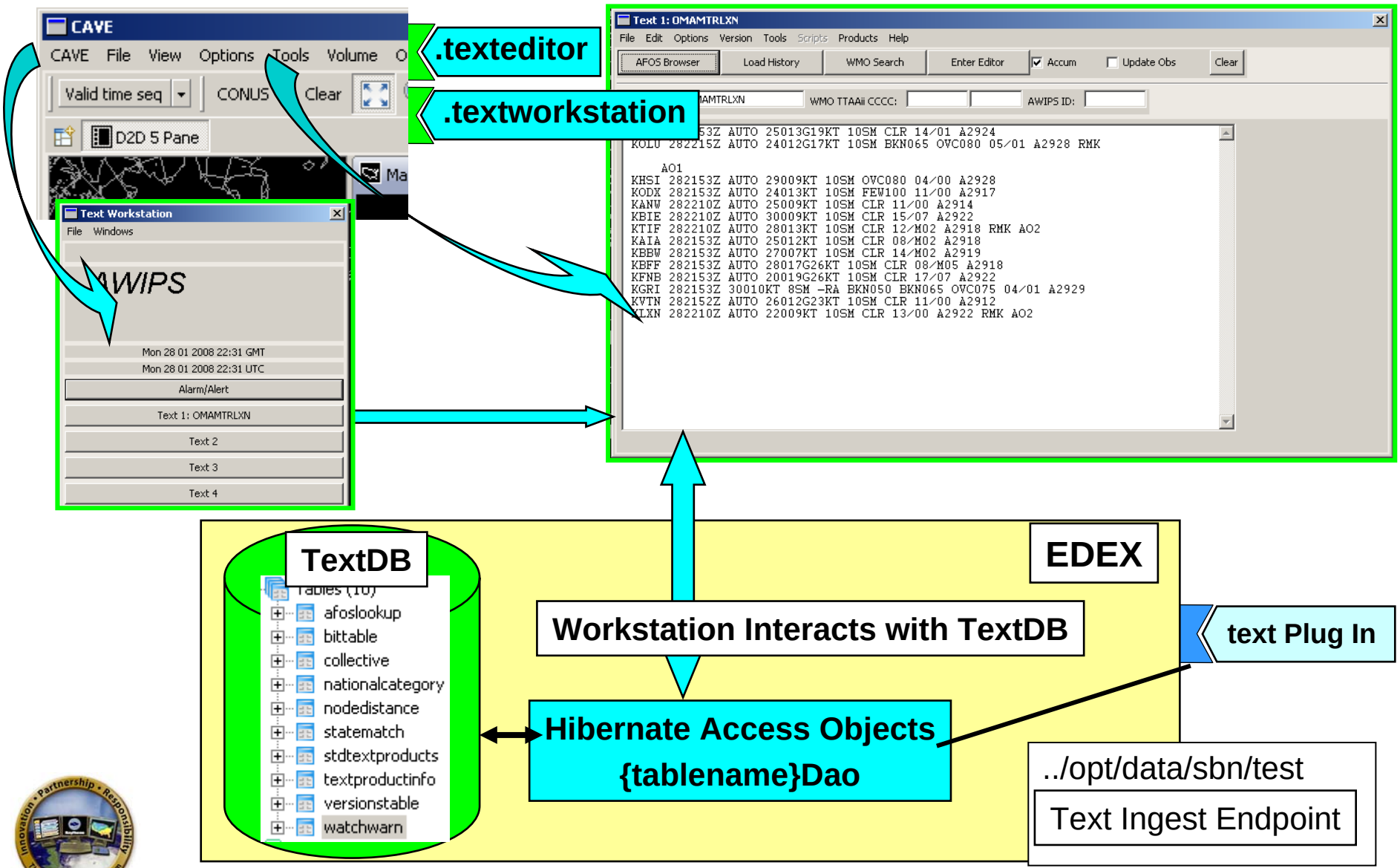
THE NATIONAL WEATHER SERVICE OFFICE IN OMAHA HAS ISSUED A

* SEVERE THUNDERSTORM WAR
EXTREME EAST CENTRAL SA
EXTREME SOUTH CENTRAL H
EXTREME NORTHWESTERN PO
IOWA...
DOUGLAS COUNTY IN EXTRE
THIS INCLUDES THE CITY OF OMAHA
WASHINGTON COUNTY IN EXTREME EAST CENTRAL NEBRASKA...
SARPY COUNTY IN EXTREME EASTERN NEBRASKA...

Workstation Text WS and Wx Editor
Implemented as 2 RCP plug ins in CAVE

Raytheon

Visualization

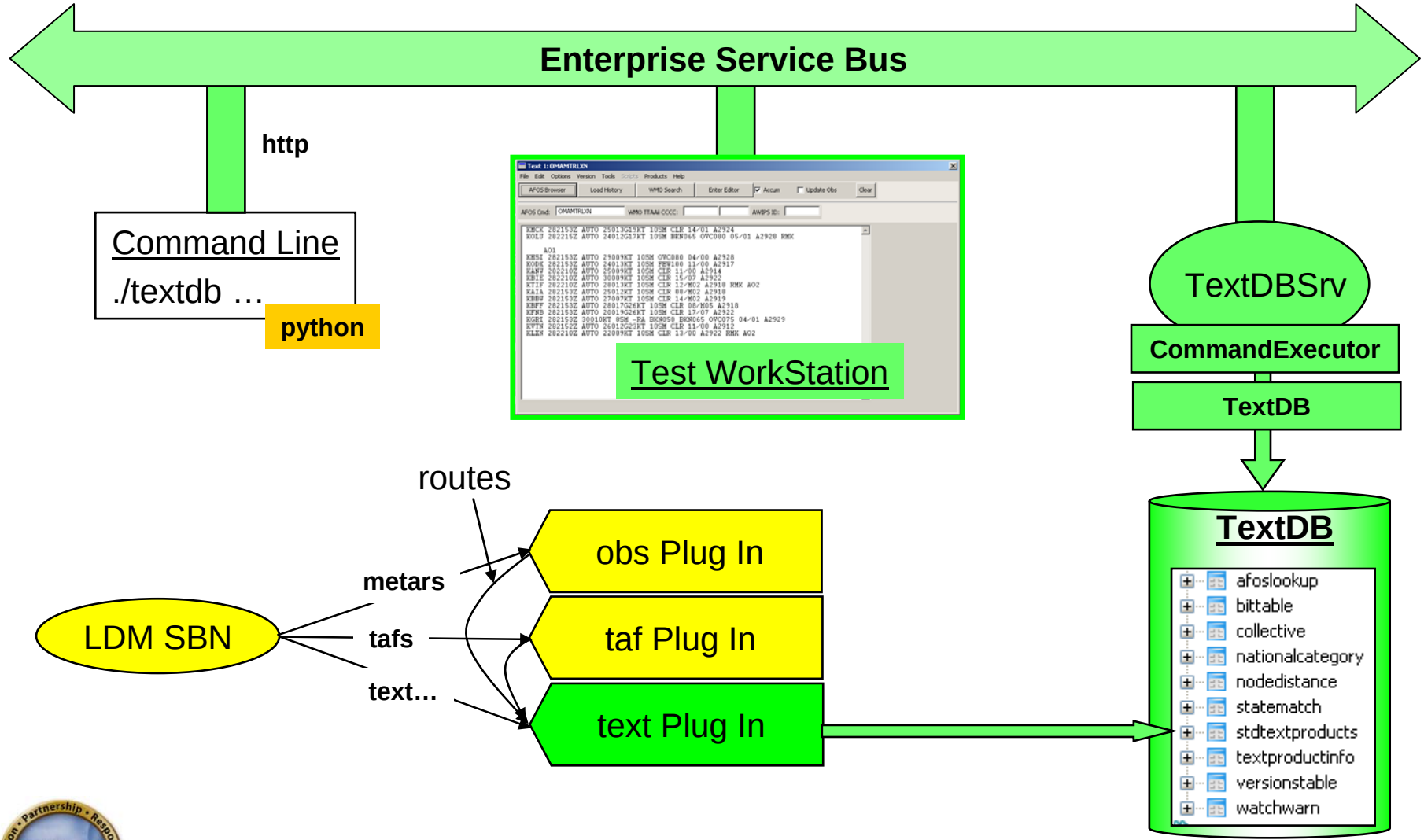


TO10 TextDB

New service, DataType Plugin, TextWS interface

Raytheon

Migration

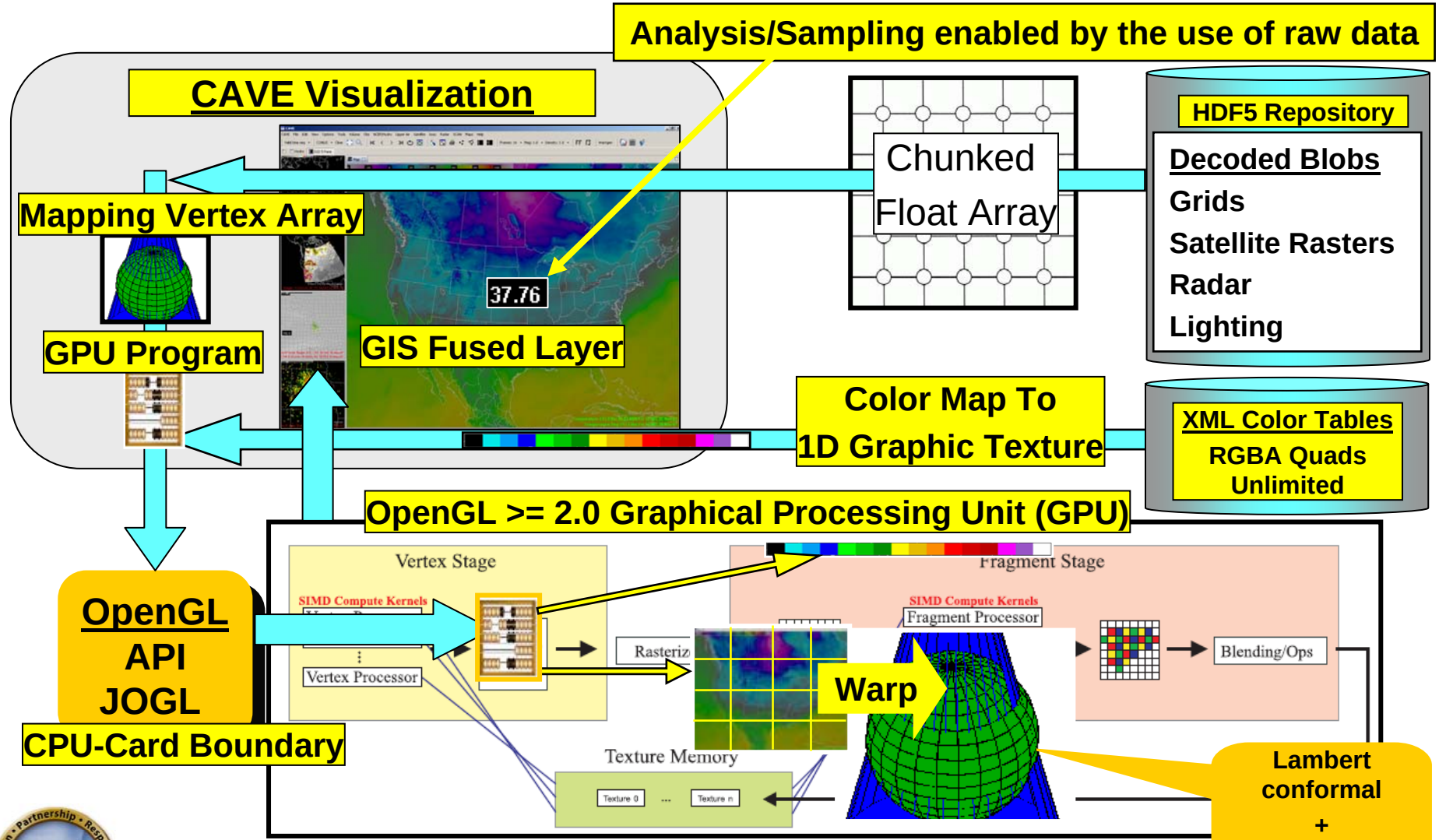


GIS Map Reprojection: By Texture Warping

GPU Program 50 times Faster than CPU

Raytheon

Visualization

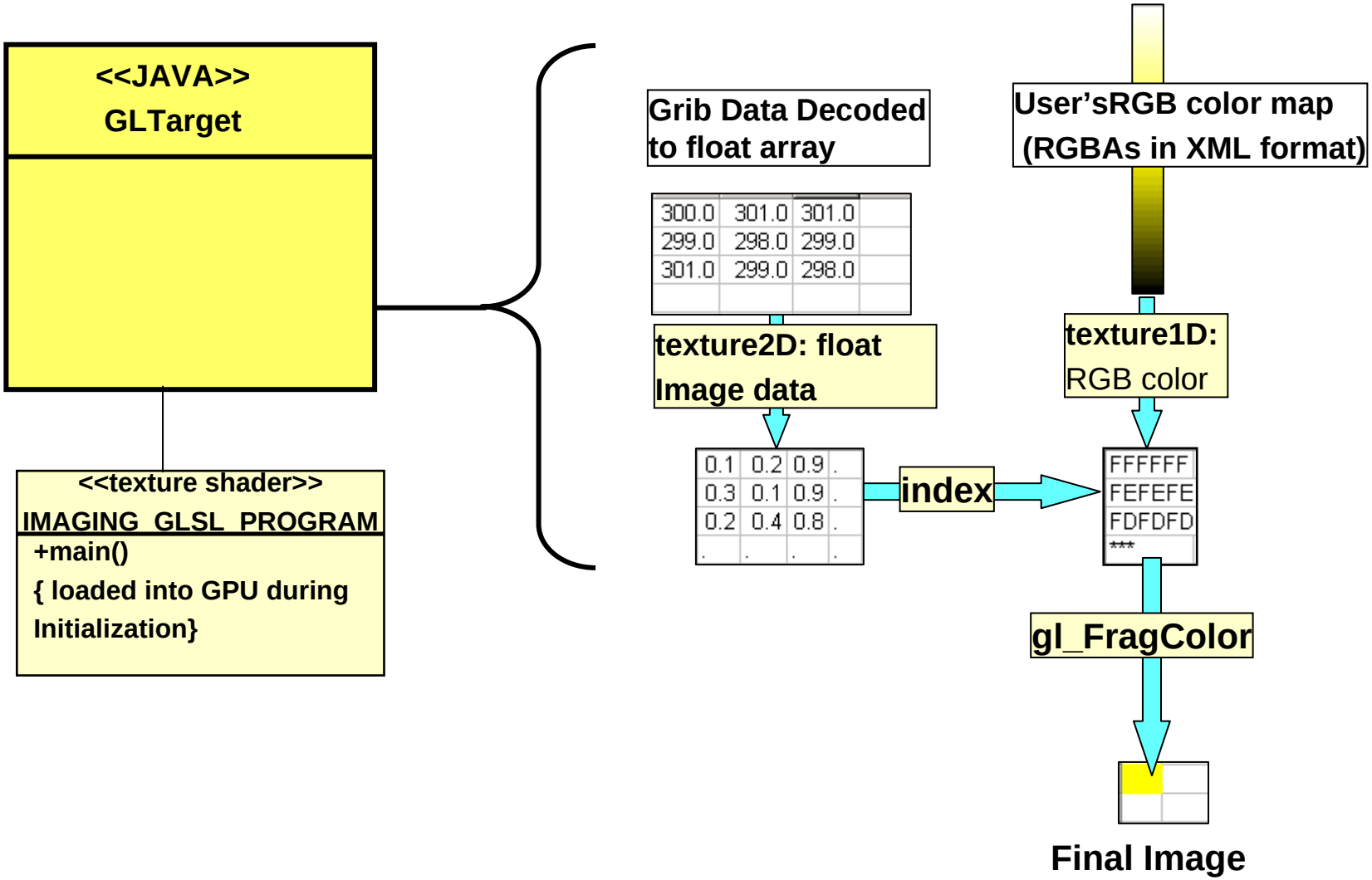


Enables Gaming Style GIS Data Fusion

GPU Fragment Shader Color Mapping
Enables Interactive Color Controls

Raytheon

Visualization

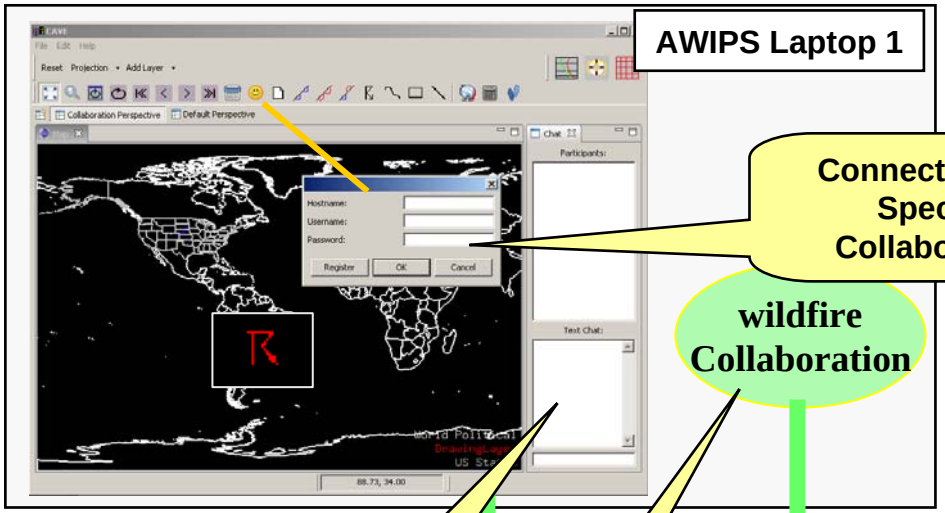


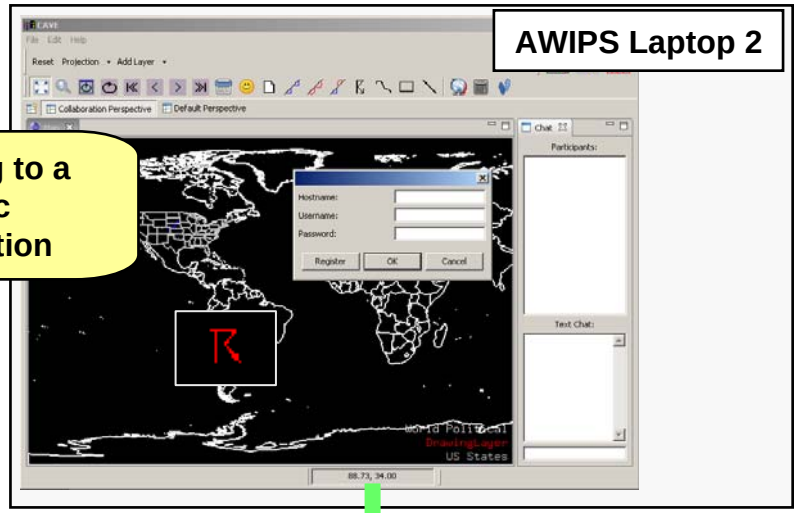
Interactive Color Maps 20 to 50 Times Faster Than CPU

GIS Collaboration Capability: Two Laptops
Based on XMPP protocol standards

Raytheon

Visualization





Connecting to a Specific Collaboration

wildfire Collaboration

Wide Area Network xmpp

Concurrent CAVE Perspective has Chat Area

Supports VoIP through plugin

Jabber Standard Collaboration Protocol

CAVE Menu Interface To Collaboration

AWIPS-II: Architecture Topics

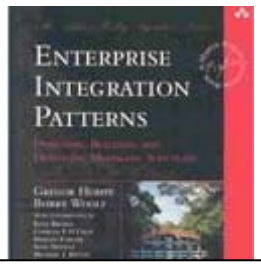
- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

AWIPS-II: Server Execution Structure
Marrying ESB with Dependency Injection

Raytheon

Server Execution

- ESB's evolved out of the expense of getting services to communicate across multiple protocols and languages
- A parallel development by Progress Software created **Sonic ESB** and they coined the term
- **CAMEL ESB** is based on "Enterprise Integration Patterns" and is the most elegant extensible solution
- The **high decoupling** of Spring's Dependency Injection Container with CAMEL's **endpoint abstract** creates a high value approach



Author: **Hohpe, Gregor**
Author: **Woolf, Bobby**

- Solving Integration Problems Using Patterns
- Messaging Systems, Channels
- Message Construction
- Message Routing
- Message Transformation
- Messaging Endpoints
- System Management

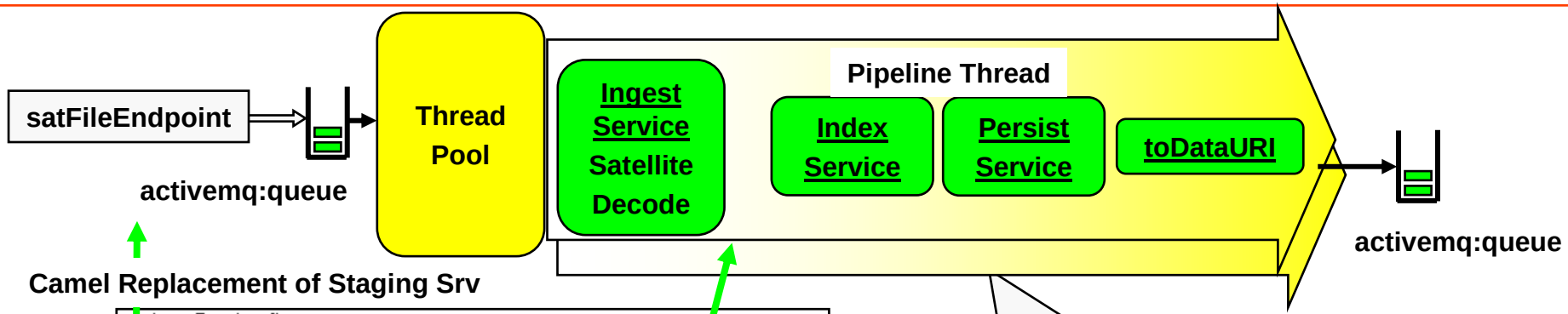
CAMEL ESB + Spring Becomes a Discriminator

Camel ESB Wiring Packaged in Plug In

Simplies plug in deployment

Raytheon

Architecture



Camel Replacement of Staging Srv

Example: Satellite Ingest

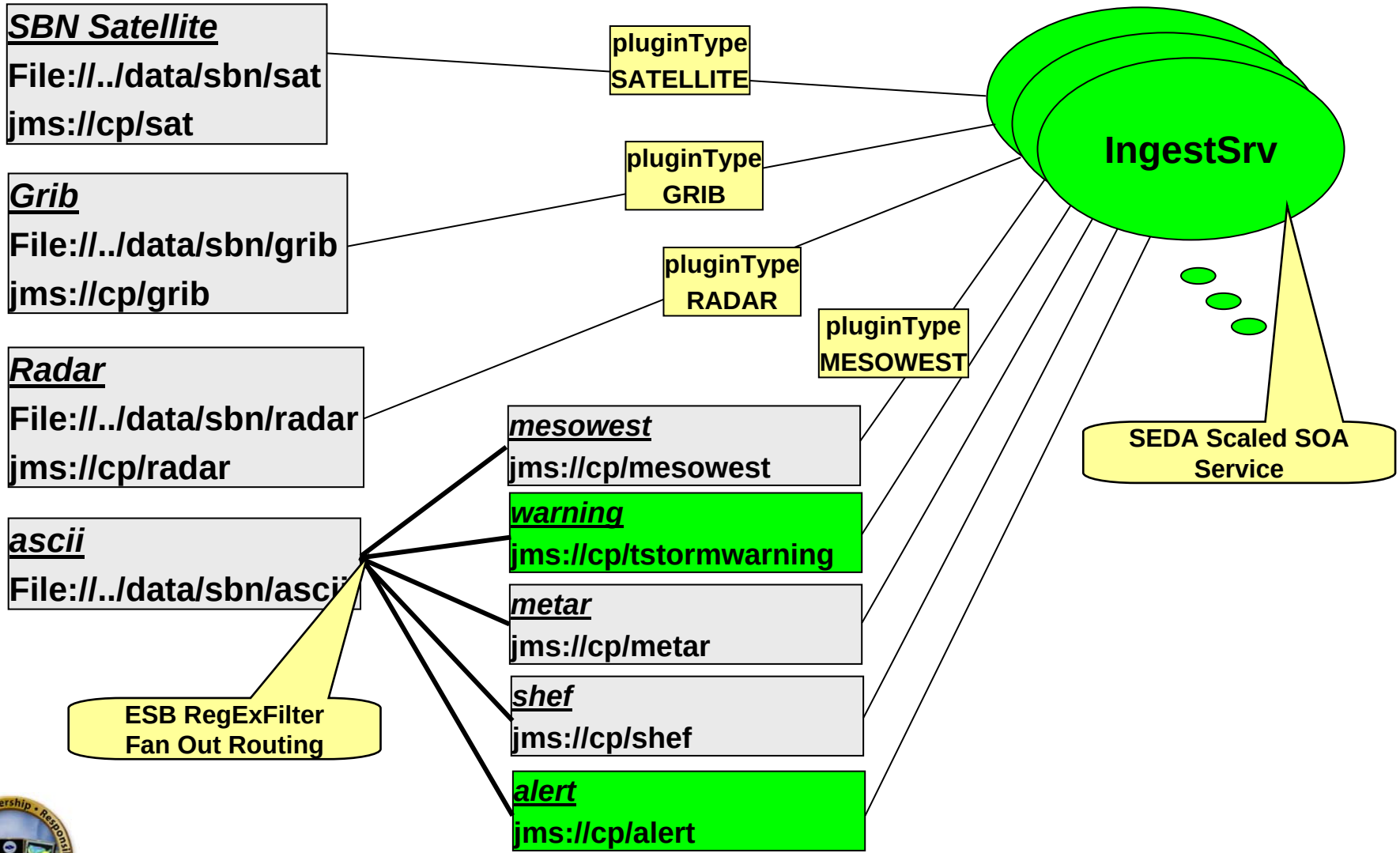
```
<!-- Begin Sat routes -->
<route id="satFileConsumerRoute">
    <from ref="satFileEndpoint"/>
    <bean ref="fileToString" method="absPath"/>
    <setHeader headerName="pluginName">
        <constant>satellite</constant>
    </setHeader>
    <to uri="activemq:queue:Ingest.Generic"/>
</route>
<route id="satIngestRoute">
    <from uri="directvm:satelliteIngest"/>
    <multicast parallelProcessing="false">
      <try>
        <pipeline>
          <bean ref="stringToFile" method="toFile"/>
          <bean ref="satDecoder" method="decode"/>
          <to uri="directvm:persistIndexAlert"/>
        </pipeline>
        <catch>
          <exception>java.lang.Throwable</exception>
          <to uri="log:sat?level=ERROR&amp;showBody=fal
        </catch>
      </try>
    <bean ref="deleteFile" method="delete"/>
    </multicast>
</route>
```

- Pipeline out of thread pool <u>decreases thread switching</u> increasing performance
- Thread Pool <u>enable fine grain tuning</u> and enables high priority data to pass through

AWIPS-II: Data Ingest Endpoints
Example Warnings Routed to Their Own Endpoint

Raytheon

Server Execution

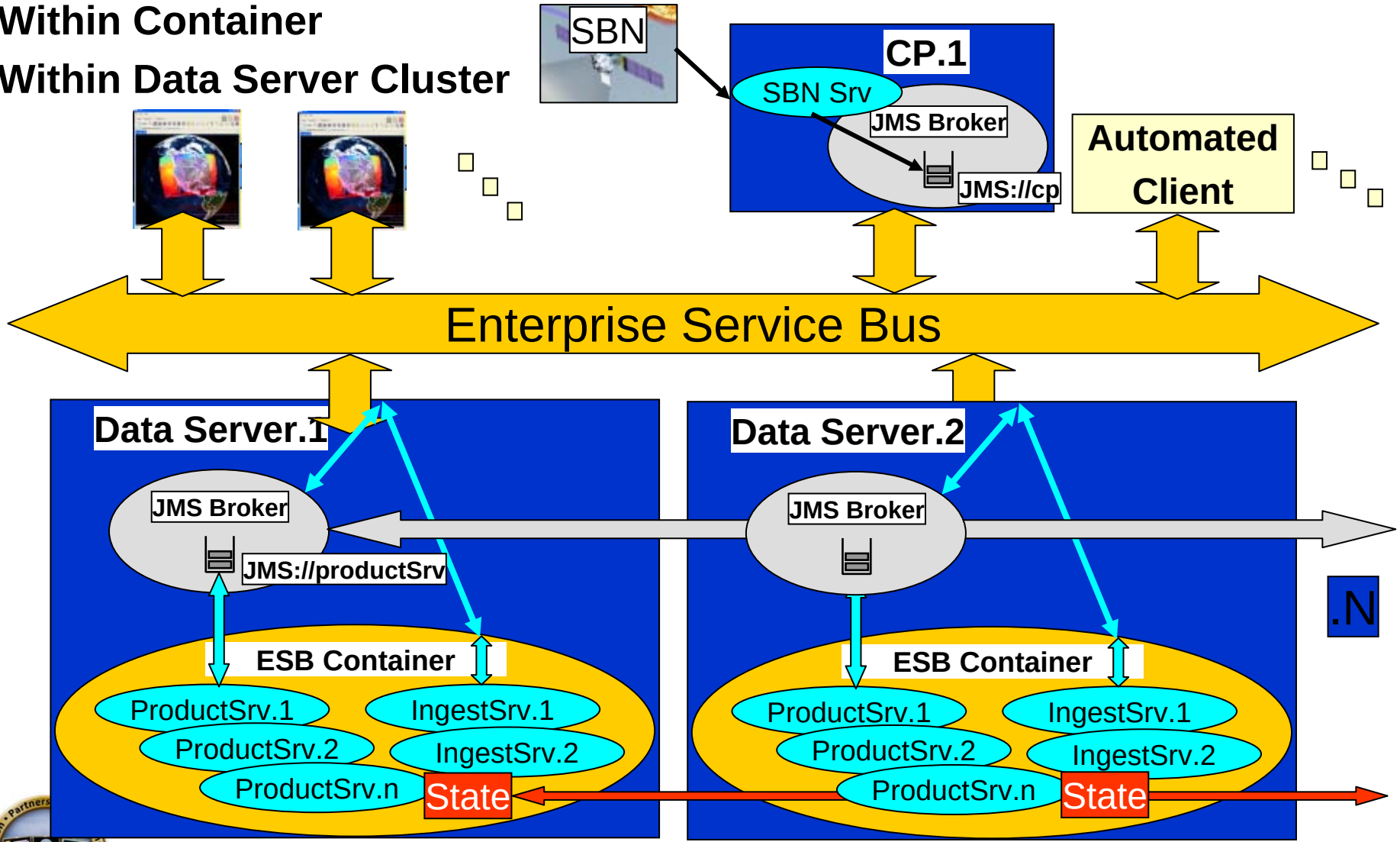


SOA – Capability Available as Services
Scales Up with SEDA Clustering on two Levels

Raytheon

Server Execution

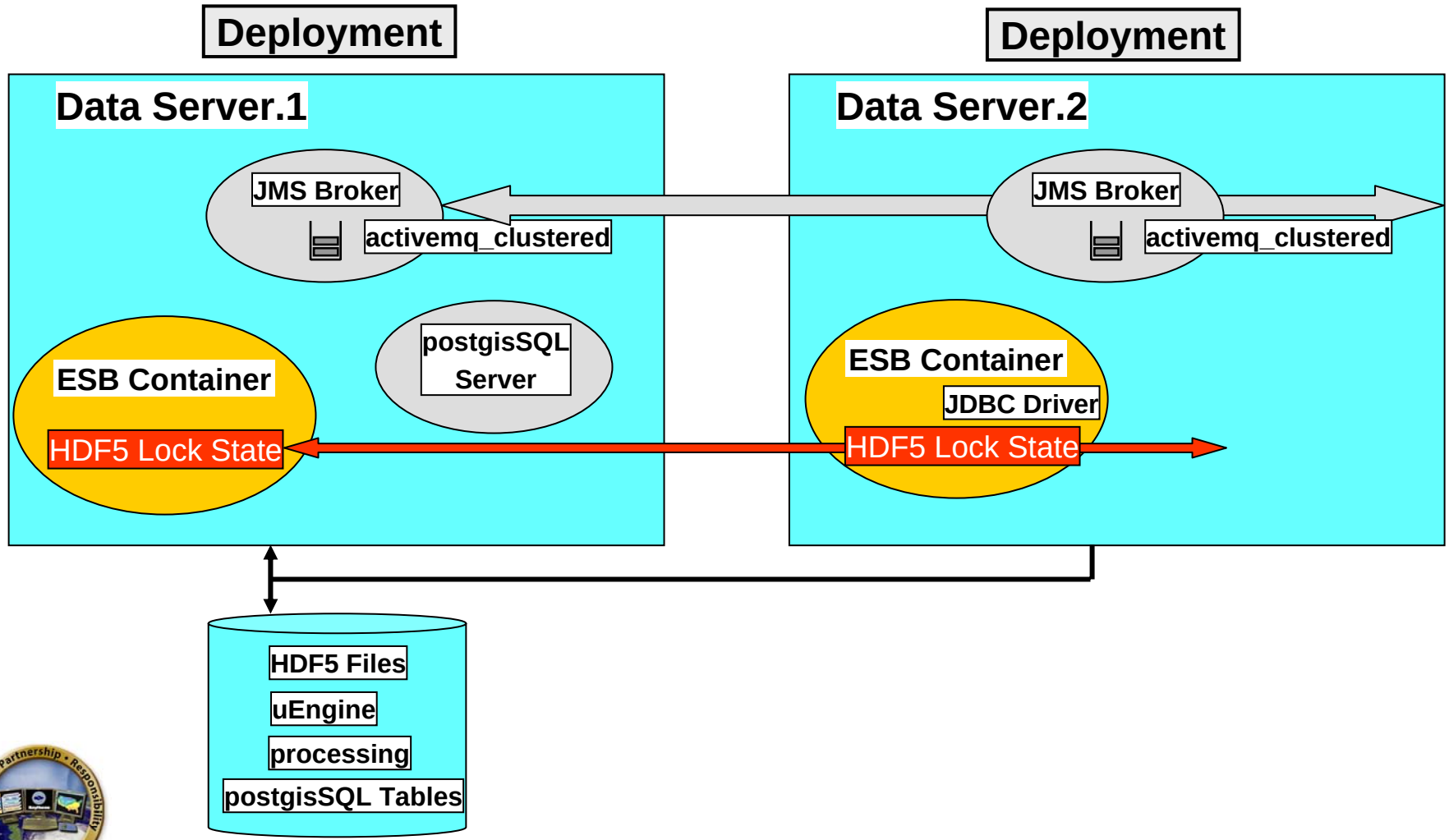
- Within Container
- Within Data Server Cluster



Server Deployments
Clustered JMS Broker Enables SEDA load balancing

Raytheon

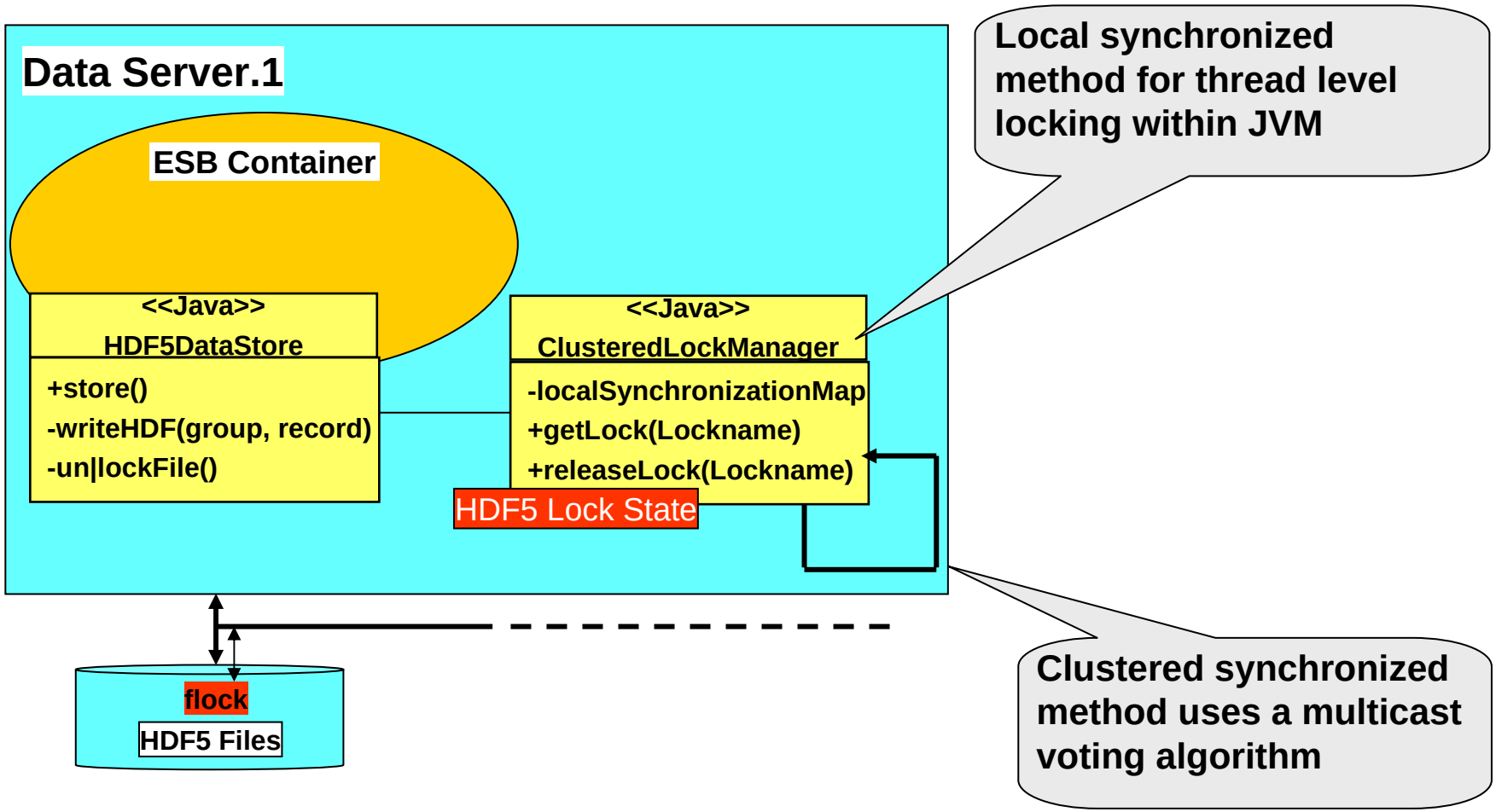
Server Execution



Server State Synchronization of Locks
Ensures Data Integrity of HDF5 Writes

Raytheon

Server Execution

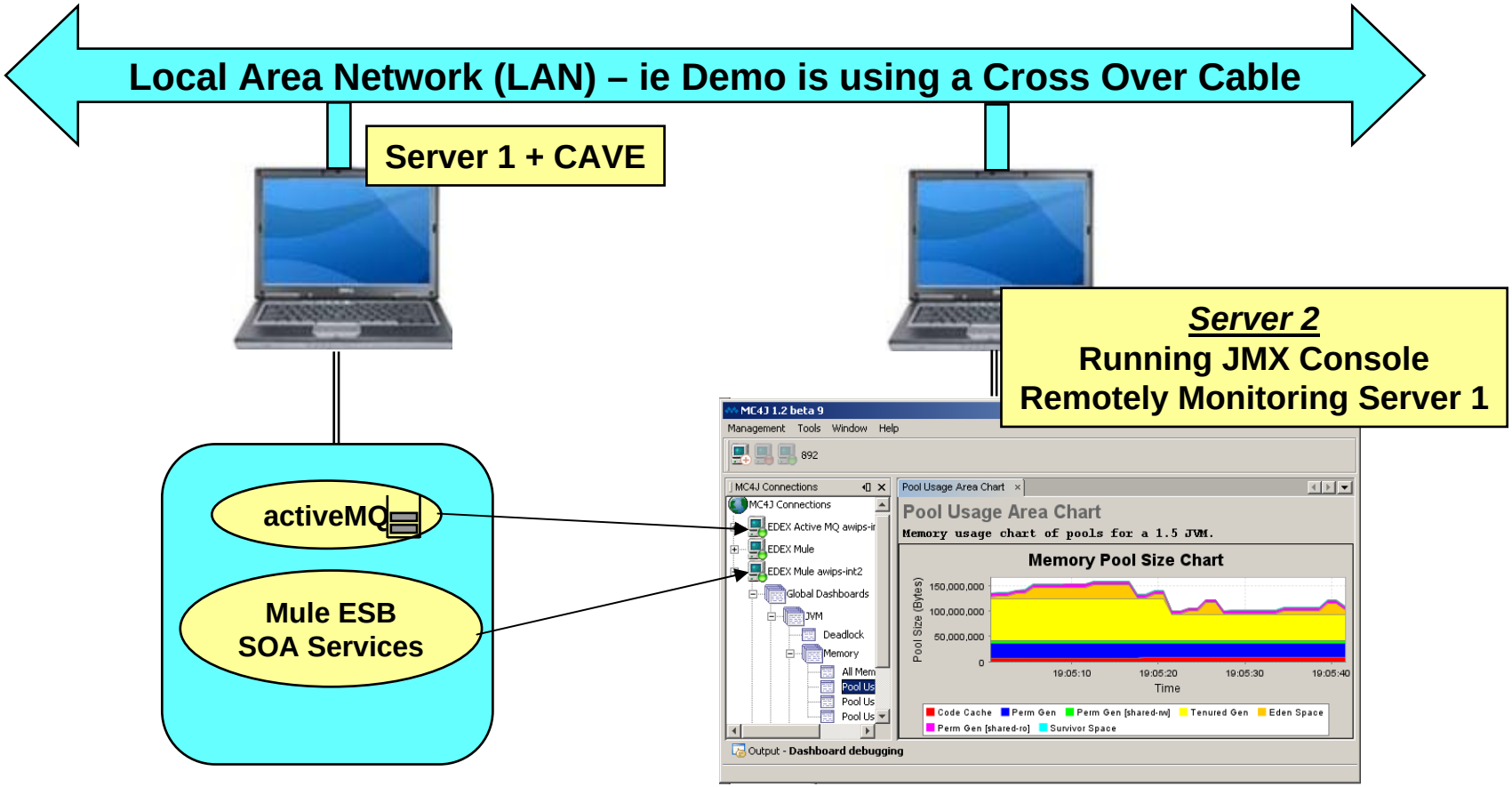


File Level HDF5 Locking For Cluster

Remote Management Through JMX
Gives Detailed Insight Into Remote Java JVMs

Raytheon

Server Execution



AWIPS-II: Architecture Topics

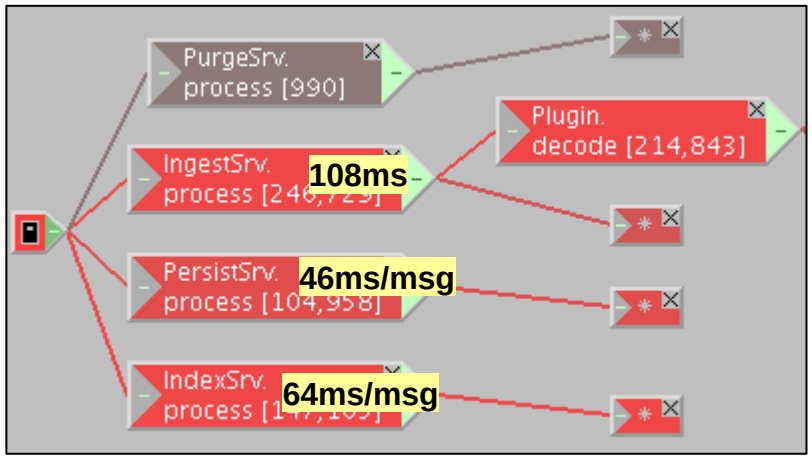
- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

Hardening – JPROBE dynamic code analysis

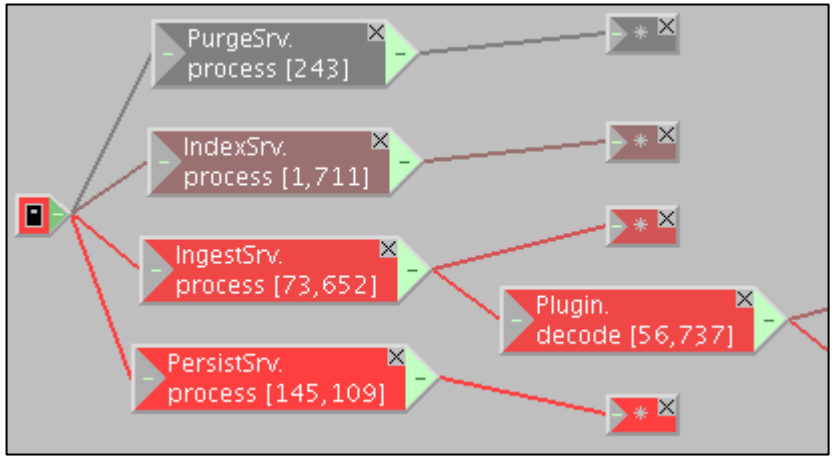
Raytheon

Hardening

Jprobe Execution Time Breakout For Grib Ingest



Satellite Ingest

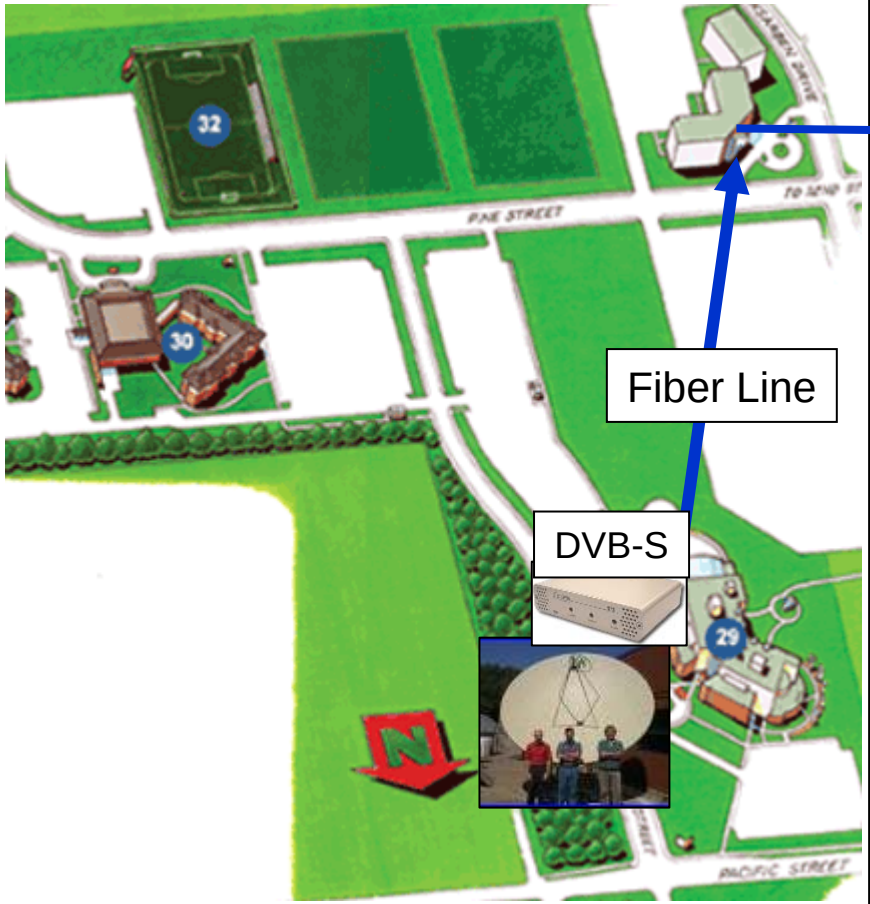


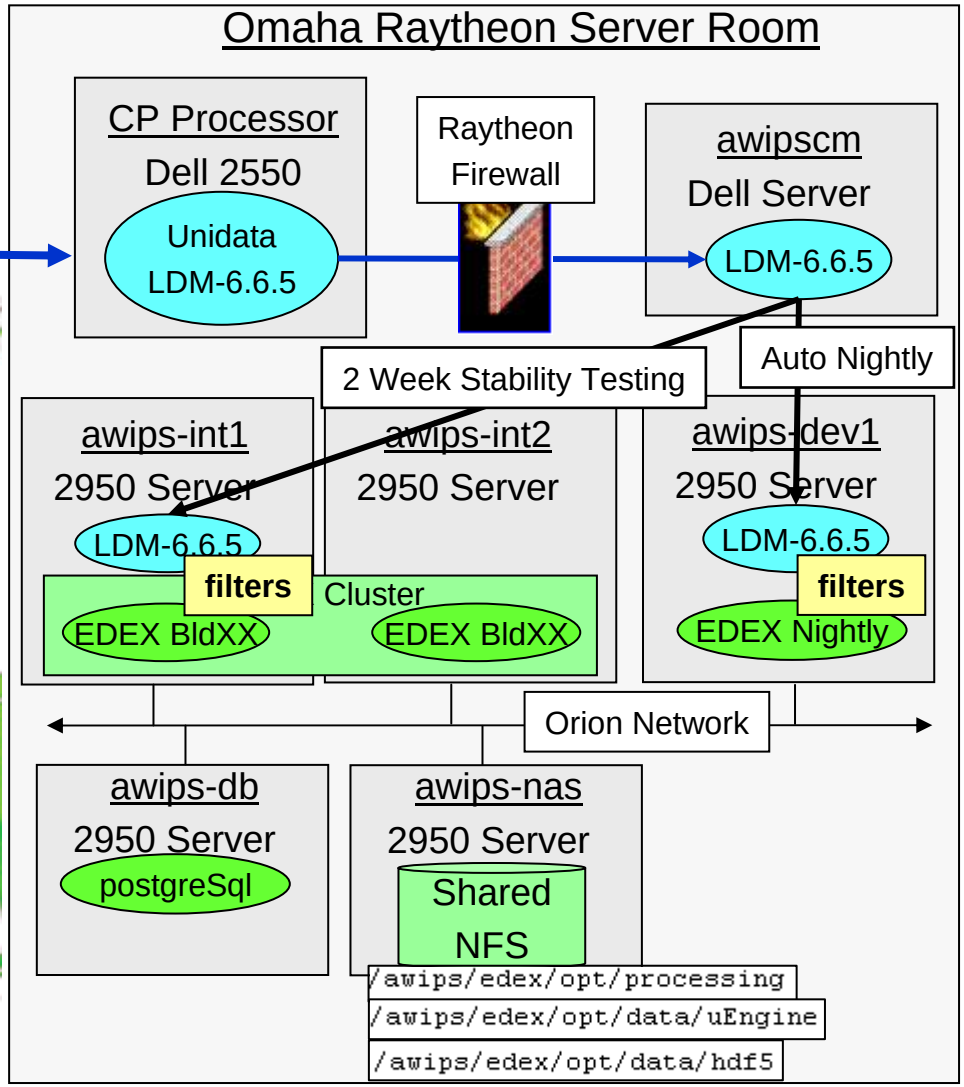
- CAVE memory leak when running loops overnight, fixed
- activeMQ dropping Mule connections (time outs), fixed
- Jgroups configuration for clustering when clustering Con-ops determined
- Message reference memory leak in AWIPS services fixed
- Message file caching for idle messages eviction policy added to prevent memory growth

Omaha Test Bed Configuration
Development and Test Hardware Resources

Raytheon

Hardening



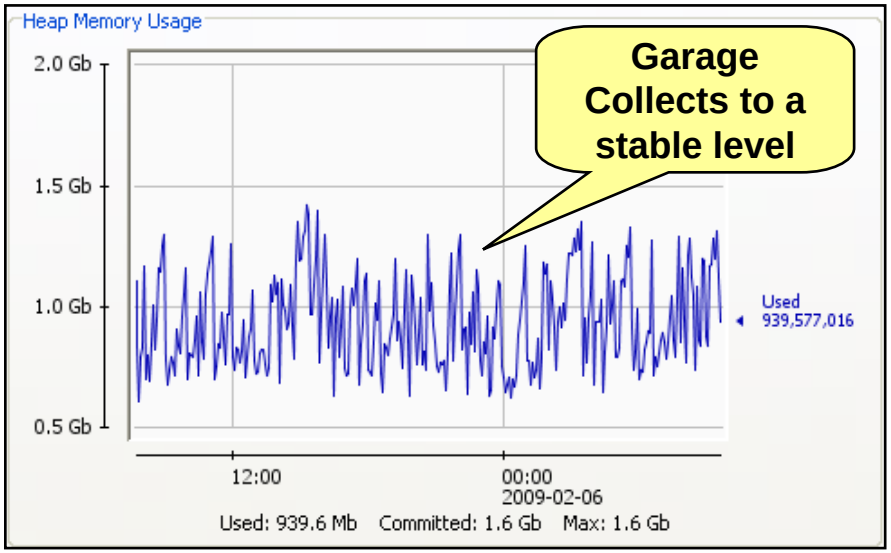


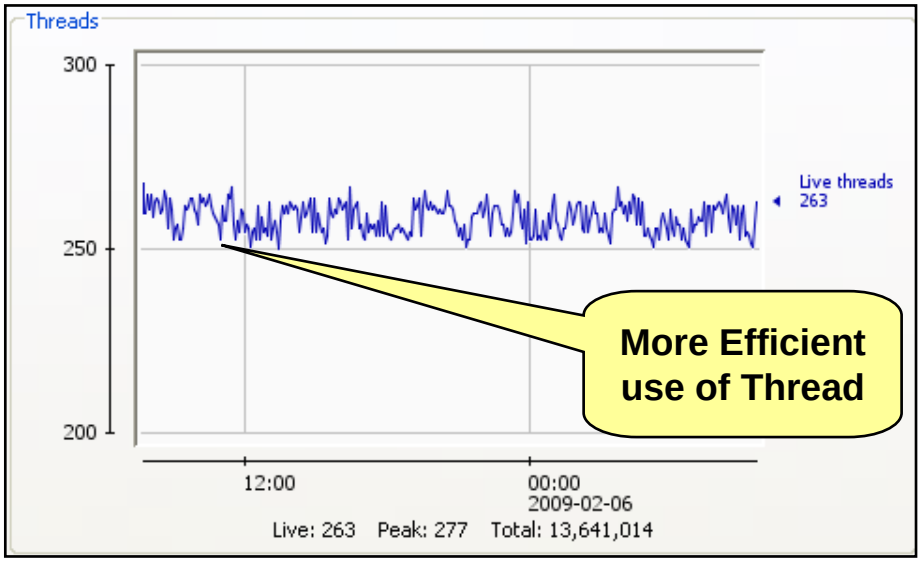
TO10 AWIPS Stability Run Results

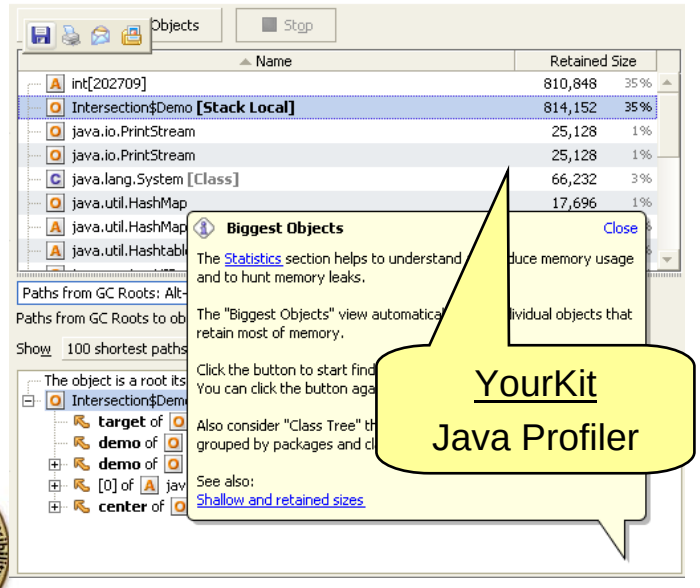
Latest build JVM 1.6.0_05

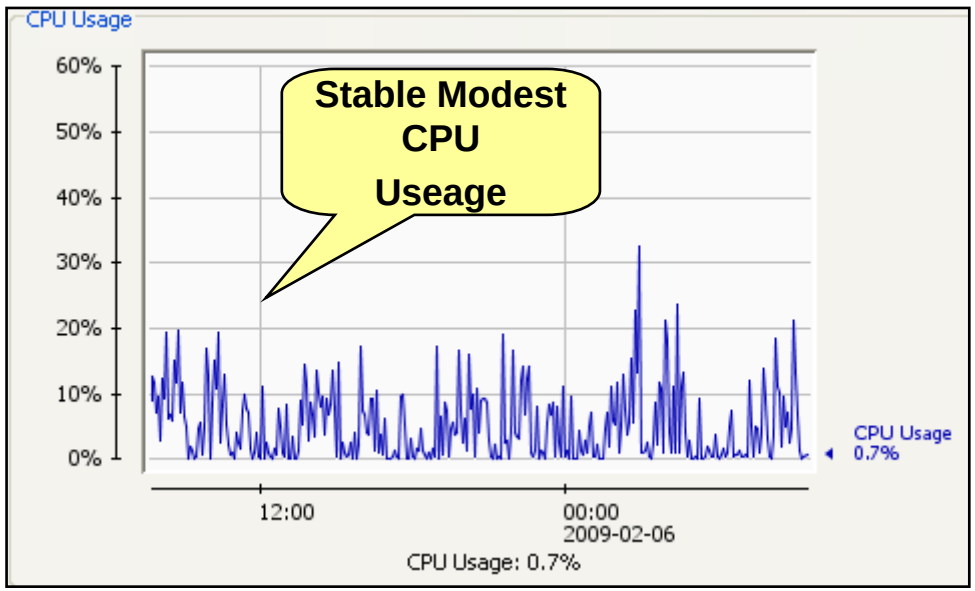
Raytheon

Stability





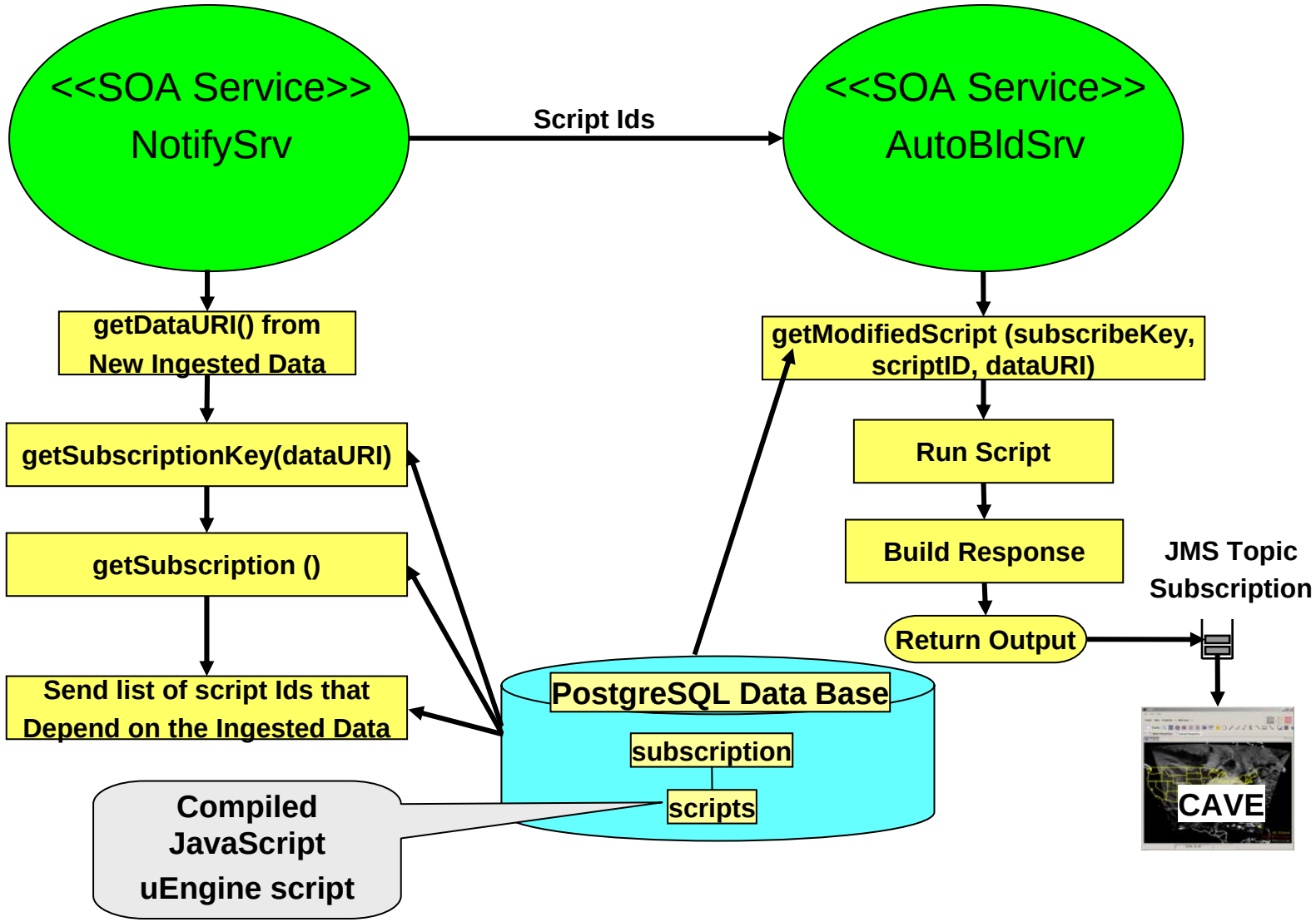




NotifySrv Changes
Scripts and Subscriptions are in RDBMS

Raytheon

Hardening



AWIPS-II: Architecture Topics

- Architecture Introduction
- Architecture Overview
- Architecture of SOA Service
- Architecture Patterns
  - Plug In pattern
  - Build pattern, Localization Pattern
  - uEngine pattern, GIS capability, Adapter pattern
  - Notification/Subscription pattern
- Data Model
- Visualization
- Server Execution
- Hardening
- Security

Security Architecture Concepts
Do not change anything significantly

Raytheon

Security

- Each installation has perimeter security similar to what exists now we are not changing the security model.
- Government is responsible for C&A package
- Goal is to be able to use a remote service as if it is local
- Access Authentication
- Service Authorization
- Data Ingest Through Data Diode Type concept
- SOA Endpoint Security Through ESB & JMS provider

Proxy Gateway Enforces SOA Polices
Encapsulates Customer Connection Polices

Raytheon

Security

Commercial Packages
- JaxView, AmberPoint

SOA PROXY

Service Mediation
- Protocol Conversions
- Throttling
- Load Balancing
- Service Level Agreements

Authentication
WS Security…

En/Decrypt
Message Content

Blocking
Non-compliant Requests

Registry
jUDDI…

Opening Up The Enterprise

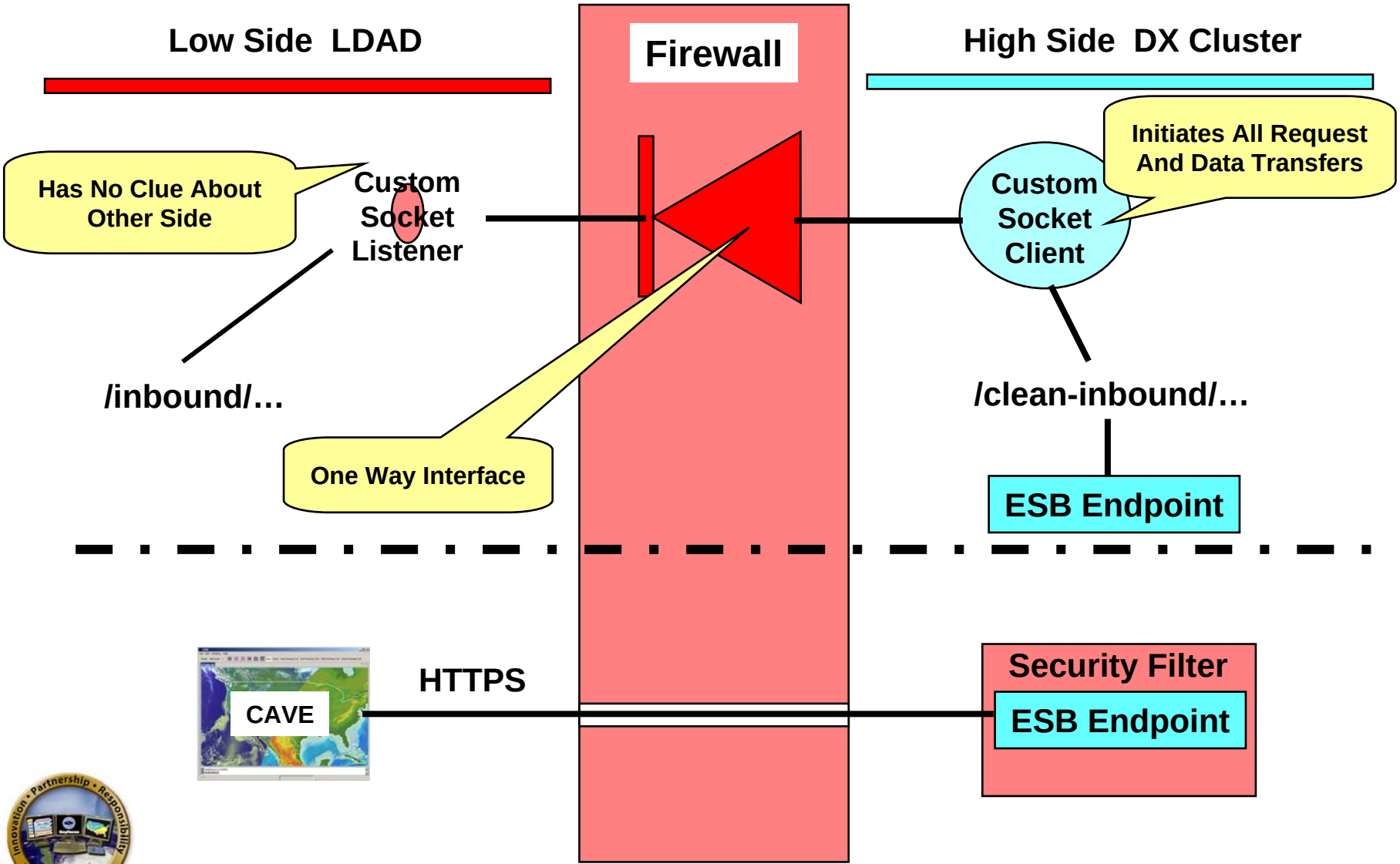
Security Concepts Continued
SOA infrastructure has built in capabilities

Raytheon

Security

- **JMS provider**: activeMQ
  - Support JAAS as a provider for authentication
  - Default points to a “login.config” on classpath that in turn points to:
    - user.properties {has: user=password info}
    - group.properties {has: group=user1,user2… }
  - User Roles on Queues and Topics:
    - Read, Write, Admin (create)
  - Configurable message authorization policy allows each message to be content based authorized by a custom policy
- **Mule Security:** Acegi; JAAS; PGP
  - Built in configurable security manager is responsible for authenticating requests
  - Configurable encryption strategy includes transports such as SSL and HTTPS
  - JAAS security provider (Java Authentication & Authorization Service) – API in since Java 1.4
  - Mule is extendable to PGP signed encryption on endpoints for end-to-end communications

Security Architecture
Data Ingest Concept of Data Diode for Firewall

Raytheon

Security