Raytheon

Customer Success Is Our Mission

AWIPS Continuous Technology Refresh (CTR)

AWIPS Software CTR

TO5: ADE System Walkthrough

March 28, 2006

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.

T05 Out Brief Topics
Theme of T05: Server Side Deployment and Scalability

- AWIPS-II Project Overview
- Architectural Status and Changes
- Data Model and Data Access Layer
- Server Side Execution Structure and Clustering
- Legacy Software Process Adapters
- CAVE Visualization Enhancements
- Security

Architecture Migration Roadmap: ADE Development

T05 Finishes the Core of the ADE, T06 Hardens the ADE

Raytheon

Project Overview

1/1/06 — 6/15/06 — 1/4/07 — 3/28/07 — 6/15/07

Raytheon IRAD	TO 3	TO 4	TO 5	TO 6

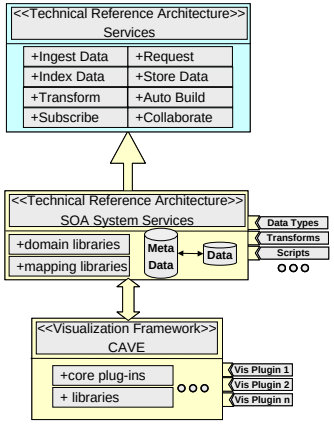
Tech. Reference Arch.
- SOA service framework
- Plug-in framework
- Micro Engine
- Core Base of Services
- Core Patterns
- Initial Open Source Project Integration

Release 0.0
1rst Delivered Code
- CM / Deploy Pattern
- SOA Templates
- Data Plug-ins
- Sub / notification
- mEngine Scripting
- CAVE

Release 0.1
- Regression Test Environ.
- CAVE +
- mEngine +
- Remote Data Access (XML)
- Dissemination

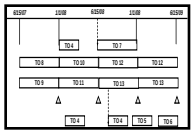
Release 0.2
- Server Execution Structure
- Remote Admin
- Legacy Adapters
- Alerting Pattern
- SOA Services +
- CAVE++

Release 1.0
- Localization Pattern
  - Server
  - CAVE
- Stress Testing
- Code hardening
- Installation Simplification



All execution on workstation

Add Server Execution



App Migration

TO 7

Detailed Application Migration Planning

T05 Architecture Status and Changes New Data Access to Enable Scalibity

Raytheon

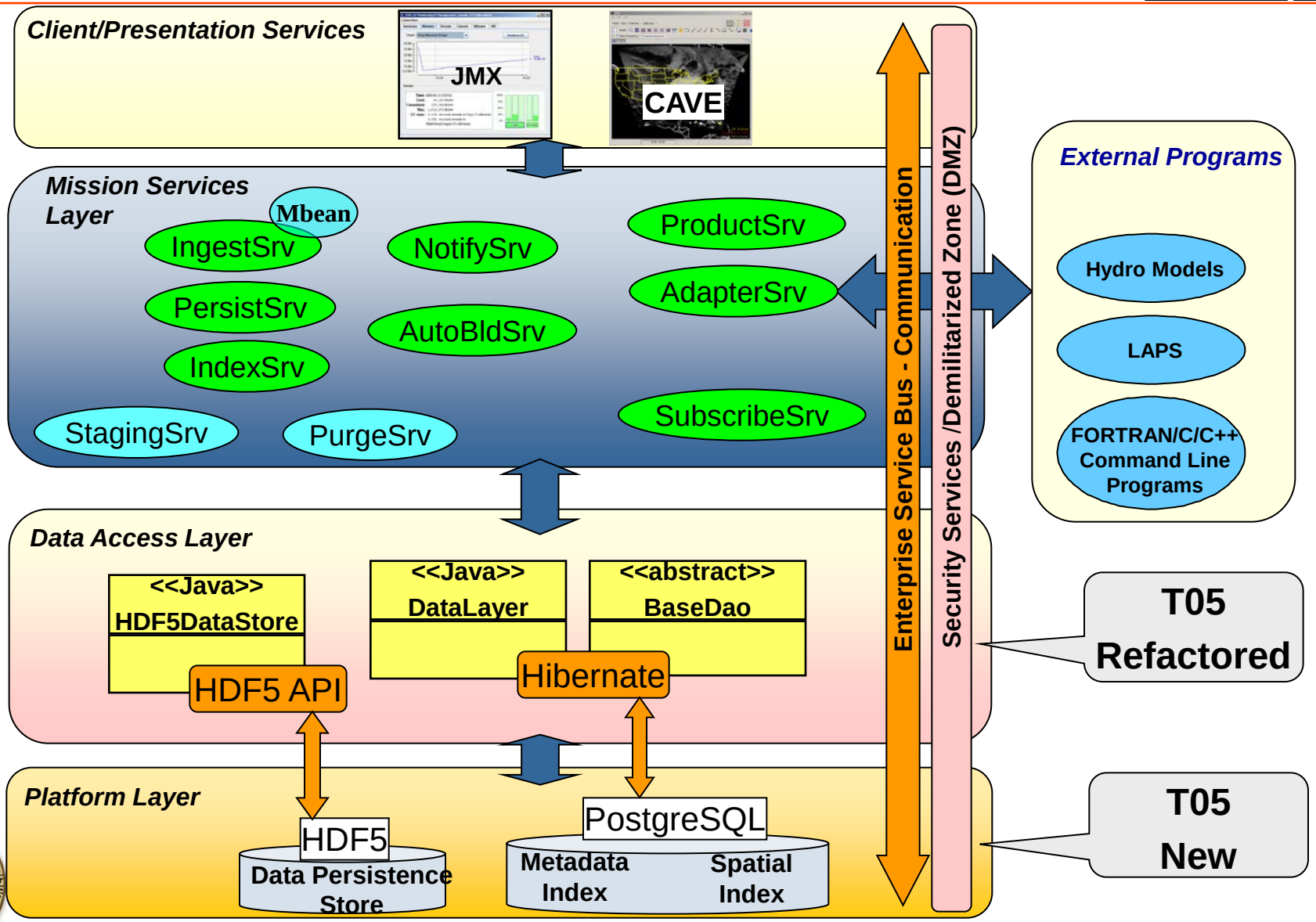
Architecture Summary

- AWIPS-II Architecture High Level Review
- Changes To Layered View
- Technical Reference Architecture Views
- Primary Pattern Summary
- Primary Data Flows
- Leveraged Open Source Projects

SOA Architecture – Logical Layered View
Layers Separated By Simple APIs

Raytheon

Architecture Summary



SOA – Technical Reference View
Technical Reference Architecture Extends

Raytheon

Architecture Summary

Core Base of Services
- Micro Engine
- Plug-in Framework
- Extensible XML model
- Core SOA Services

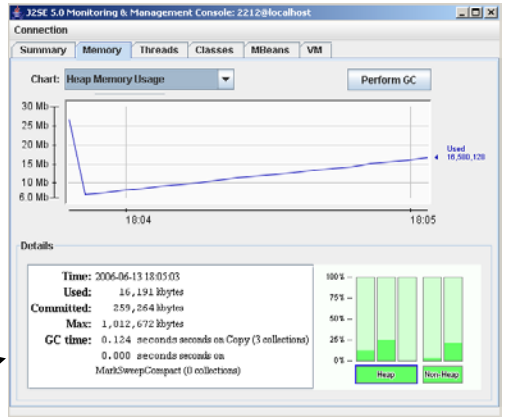
Extend to a specific domain
- Plug in specific libraries
- Plug in data types, transforms

Service Interface To Data
- Clean separation between data and visualization
- Canonical XML data model
- Scriptable Interface

<Technical Reference Architecture> Services

+Ingest Data	+Request
+Index Data	+Store Data
+Adapt	+Auto Build
+Subscribe	+Collaborate

Remote Management (JMX)



Satellite Feed 4.5Mps

Local Data

<Technical Reference Architecture> SOA System Services (EDEX)

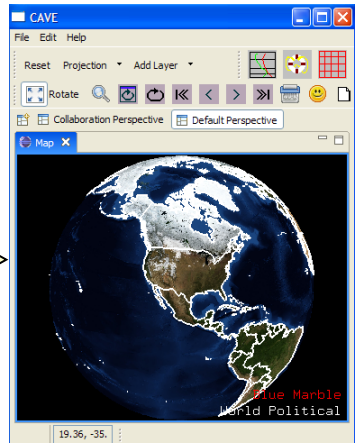
+domain libraries	Meta Data ↔ Data
+mapping libraries	

Data Types
Transforms
Scripts
○○○

<Visualization Framework> CAVE

2D/3D GIS	Drawing
Raster Render	Collaboration
Vector Render	XY Render
Auto Animate	

Vis Plugin 1
Vis Plugin 2
Vis Plugin n



T05 Packaging and Deployment
Base Deployment + SOA and Visualization Plug Ins

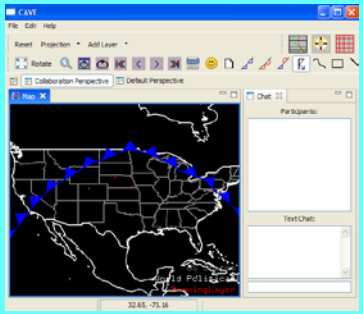
Raytheon

Architecture Summary

JAVA JVM
SOA Service
Container

- TAF
- SHEF
- RADAR
- SATELLITE
- GRIB
- METAR
- MESOWEST

JAVA JVM
Eclipse
RCP



- LIBS
- CORE
- UI
- DRAWING
- WARNGEN
- HYDRO

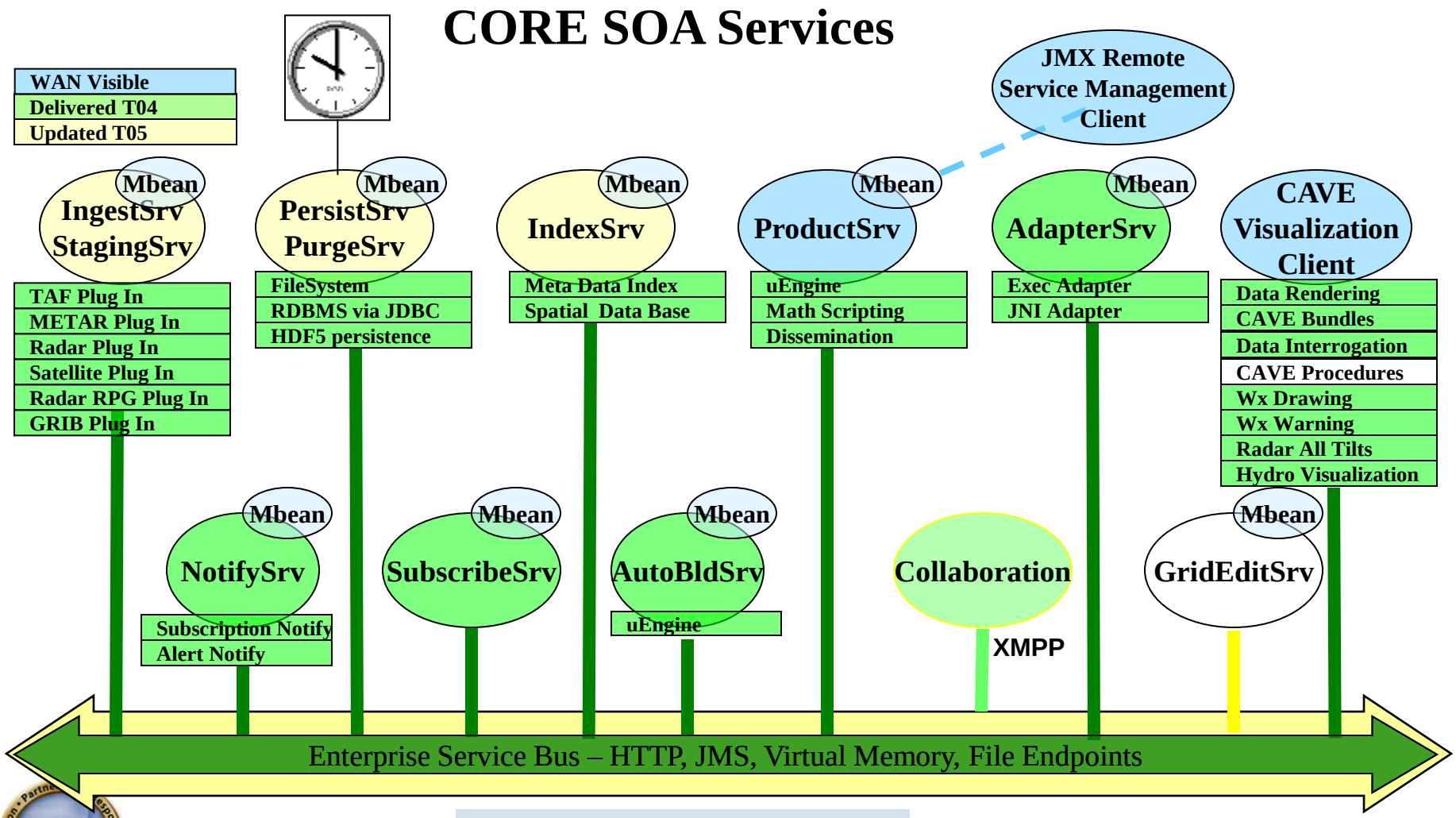
T05 Major
Refactor of Plug In Pattern

Incremental Upgrades Enabled Through Plug Ins

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

Raytheon

Architecture Summary

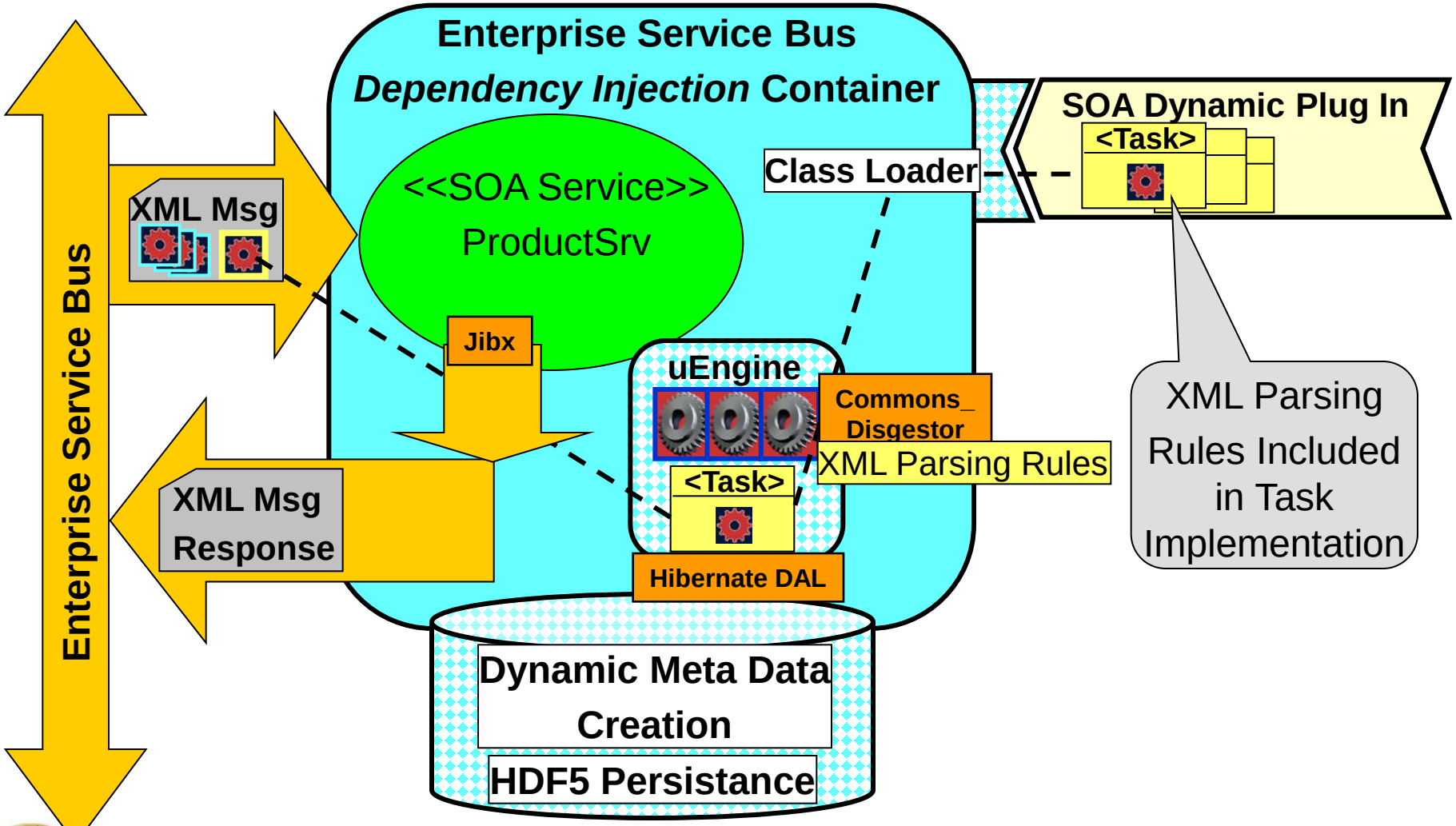


Services Independent of End Points

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

Raytheon

Architecture Summary

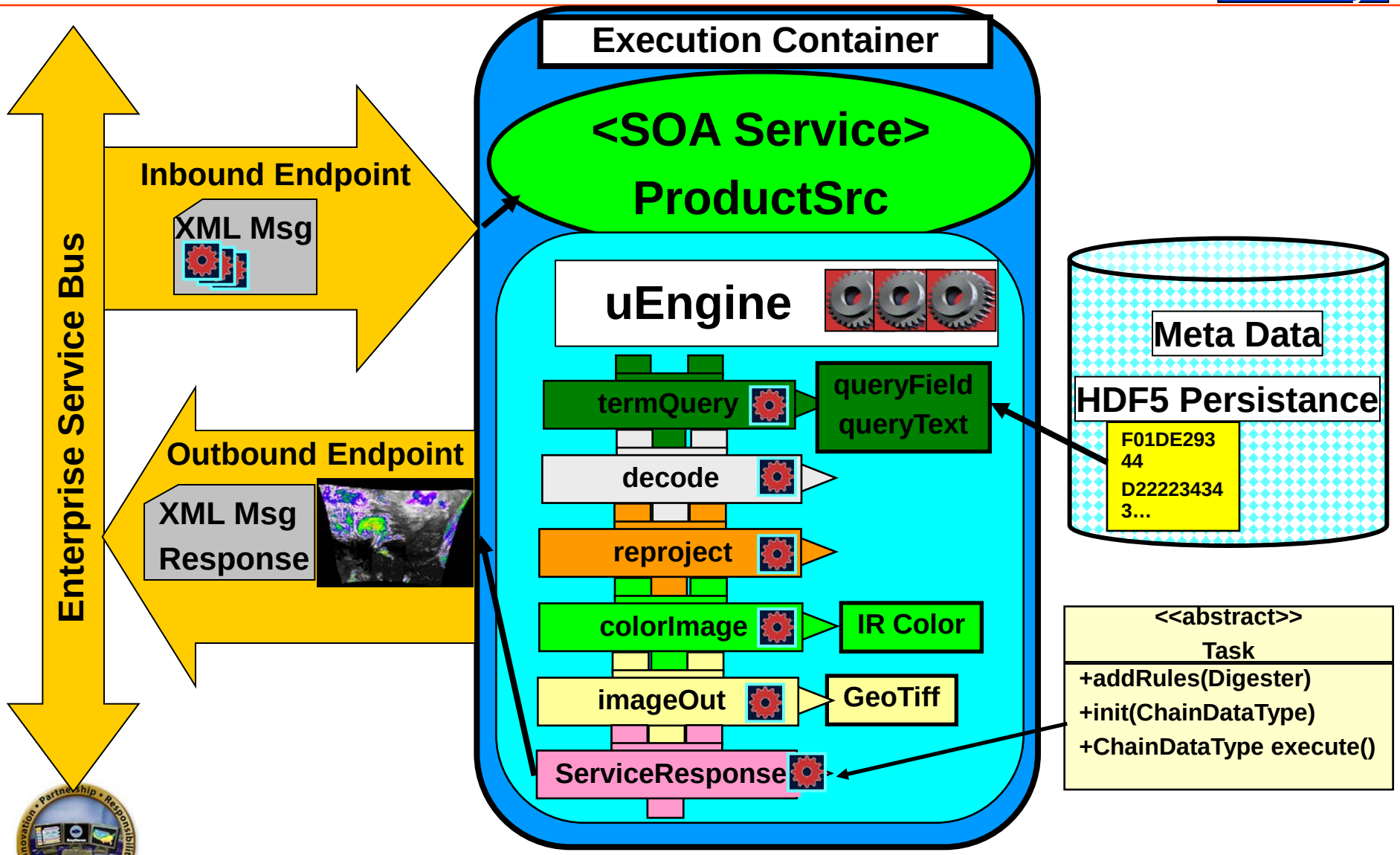


Dynamically Extends System to New Problems

uEngine Pattern Extends Services
Executing Commands to Transform/Fuse Data

Raytheon

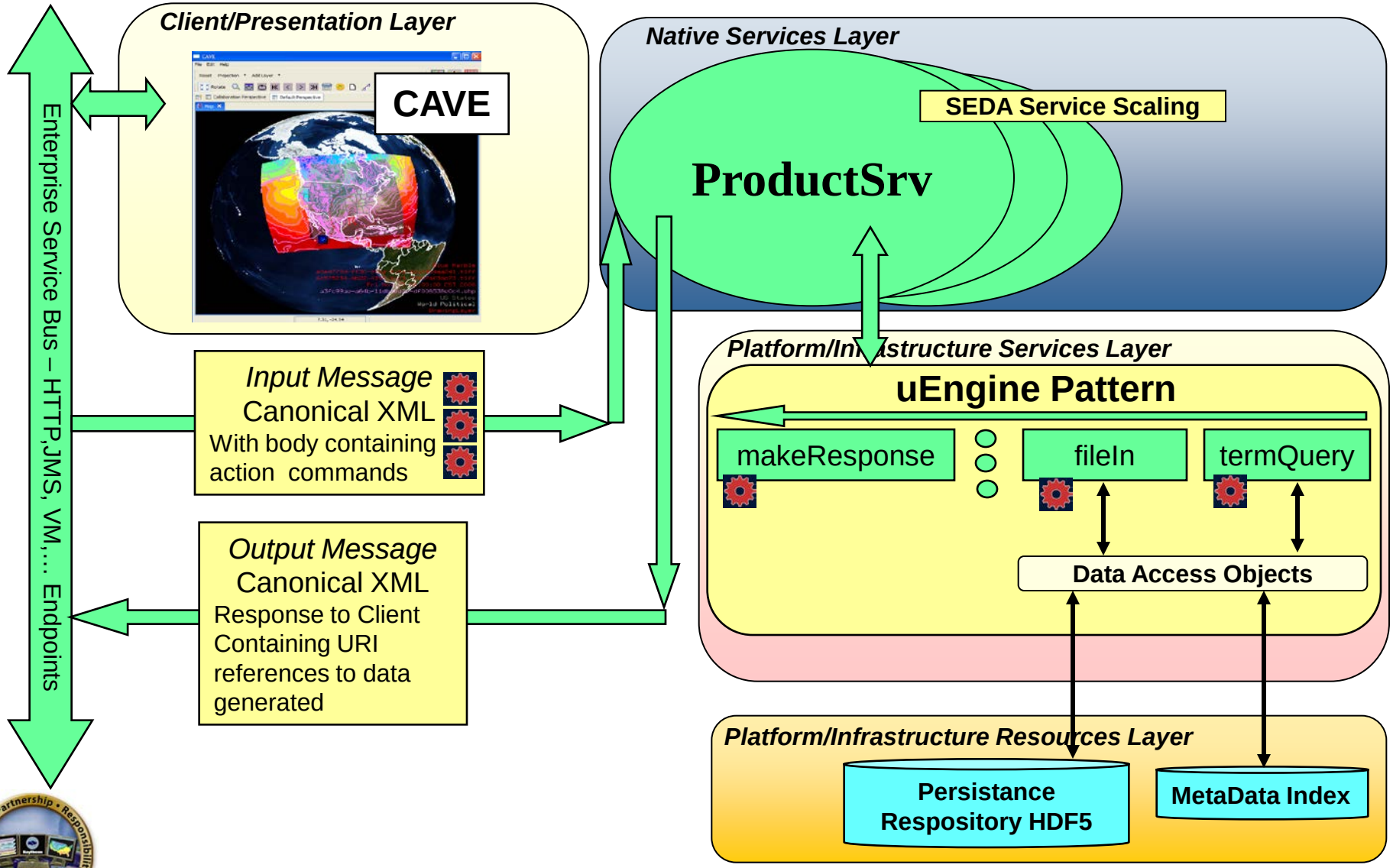
Architecture Summary



Layered View of SOA Service with Data Flow
Cave Requests Data for Display as a GIS Layer

Raytheon

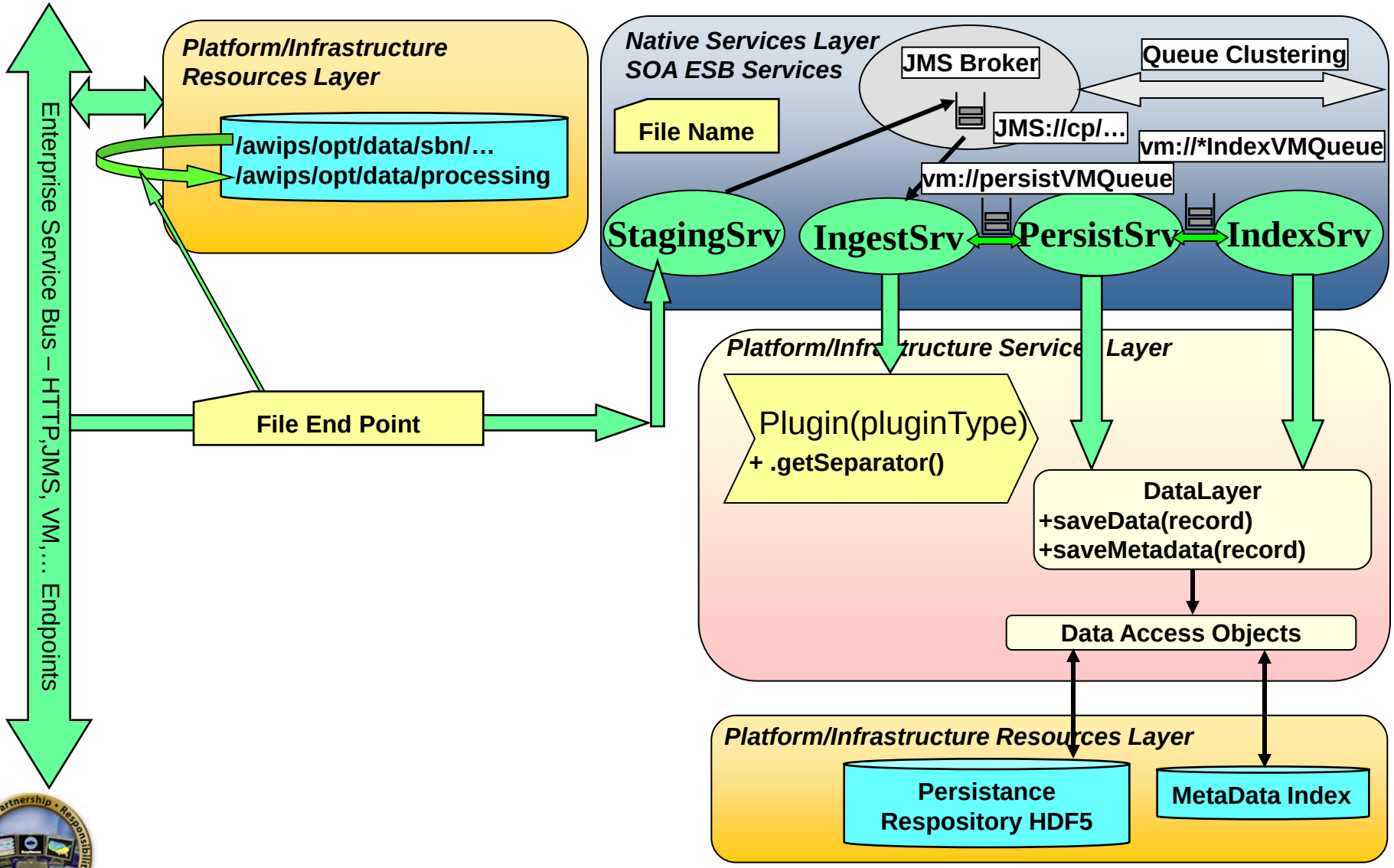
Architecture Summary



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

Raytheon

Architecture Summary



T05 Open Source Projects Leveraged
Lucene Dropped, PostgreSQL, ANTLR, Hibernate

Raytheon

Architecture Summary

ANT 1.6.5	ADE Build scripting
JAVA 1.5.0_09	ADE Java Virtual Machine
postgreSQL 8.2	RDBMS for Metadata, Hydro IHFS
Jibx 1.0	Java Object to XML mapping
Jython	Python scripting in Java
jhdf5	Java API to HDF5 from NCSA
Hibernate 3	Relational to Object Mapping (ORM)
Geotools 2.2.0	GIS libraries and tools
Antlr 2.7.6	Language grammar parser based on EBNF
Xfire 1.2.2	Collaboration Server
Eclipse RCP 3.2	ADE CAVE visualization framework
Jogl	OpenGL Java API for CAVE
Batik	SVG tool library
activemq 4.1.0	ADE Java Messaging System (JMS) broker
Mule ESB 1.3	ADE Enterprise Service Bus + Container

Changed

T05 Data Model Introduction
Canonical XML SOA Interfaces Excluded

Raytheon

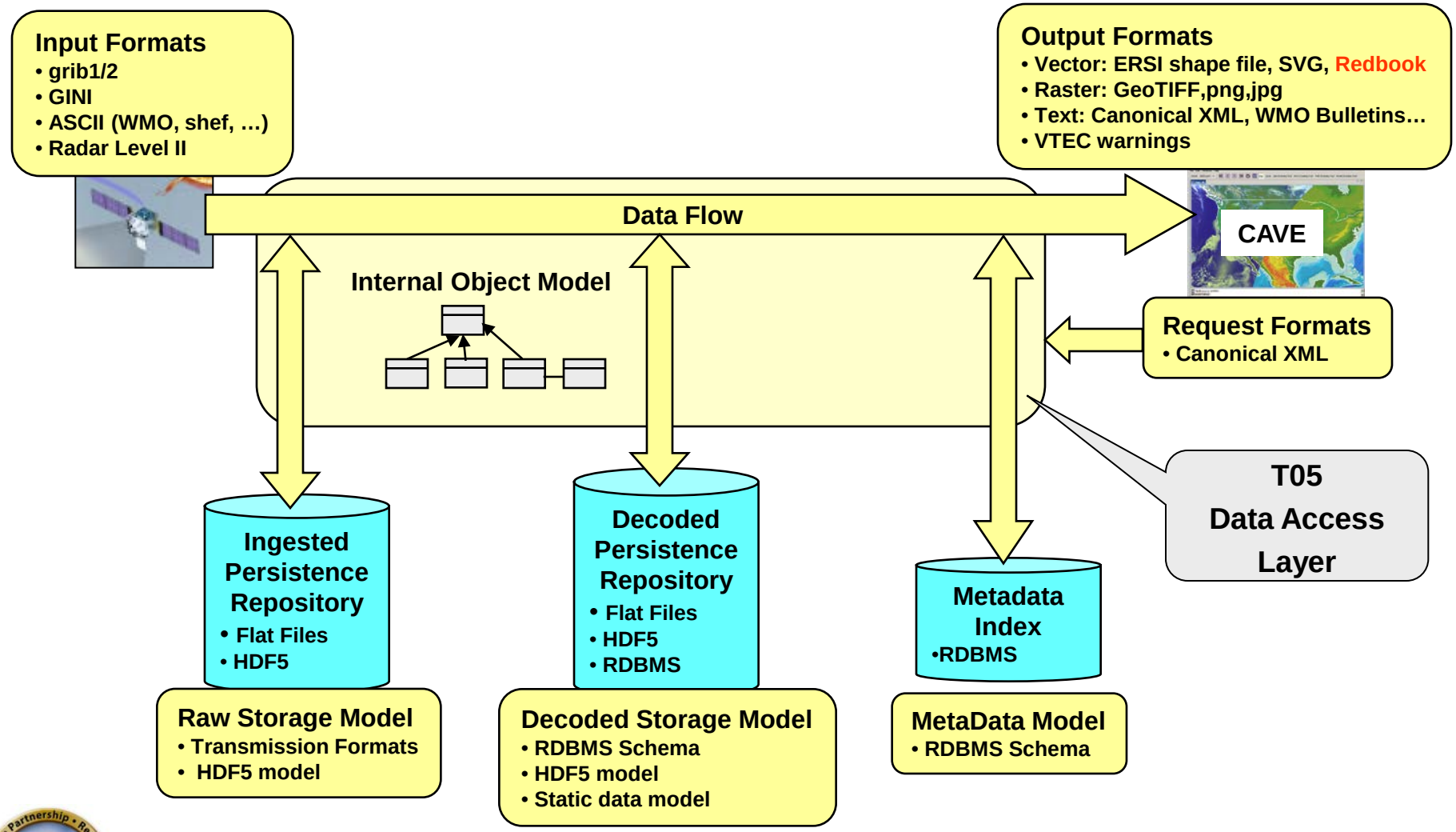
Data Model

- 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 PostgiSQL 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

T05 Conceptual Data Model Design
I/O Formats Follows Existing Standards

Raytheon

Data Model



T05 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

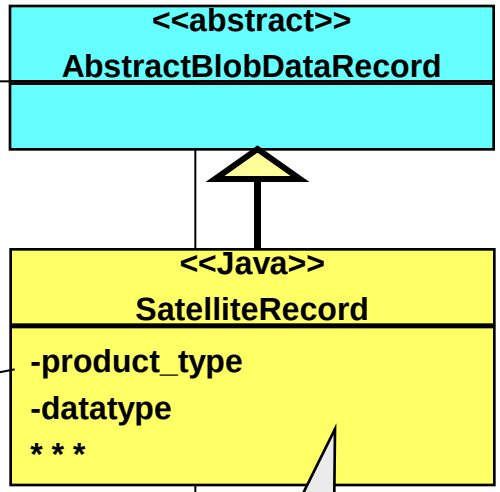
Satellite Plug In

satellite.hbm.xml

SatelliteRecord.java

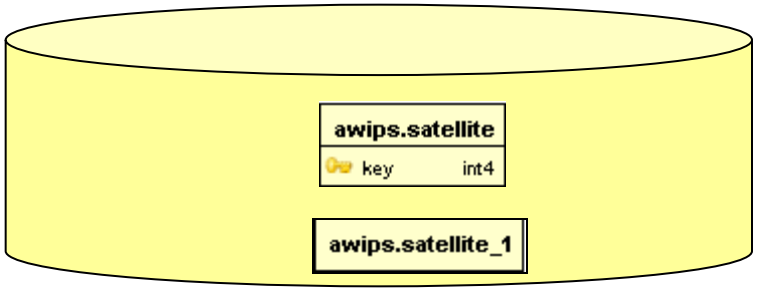
```
<hibernate-mapping default-lazy="false" schema="awips">

        <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



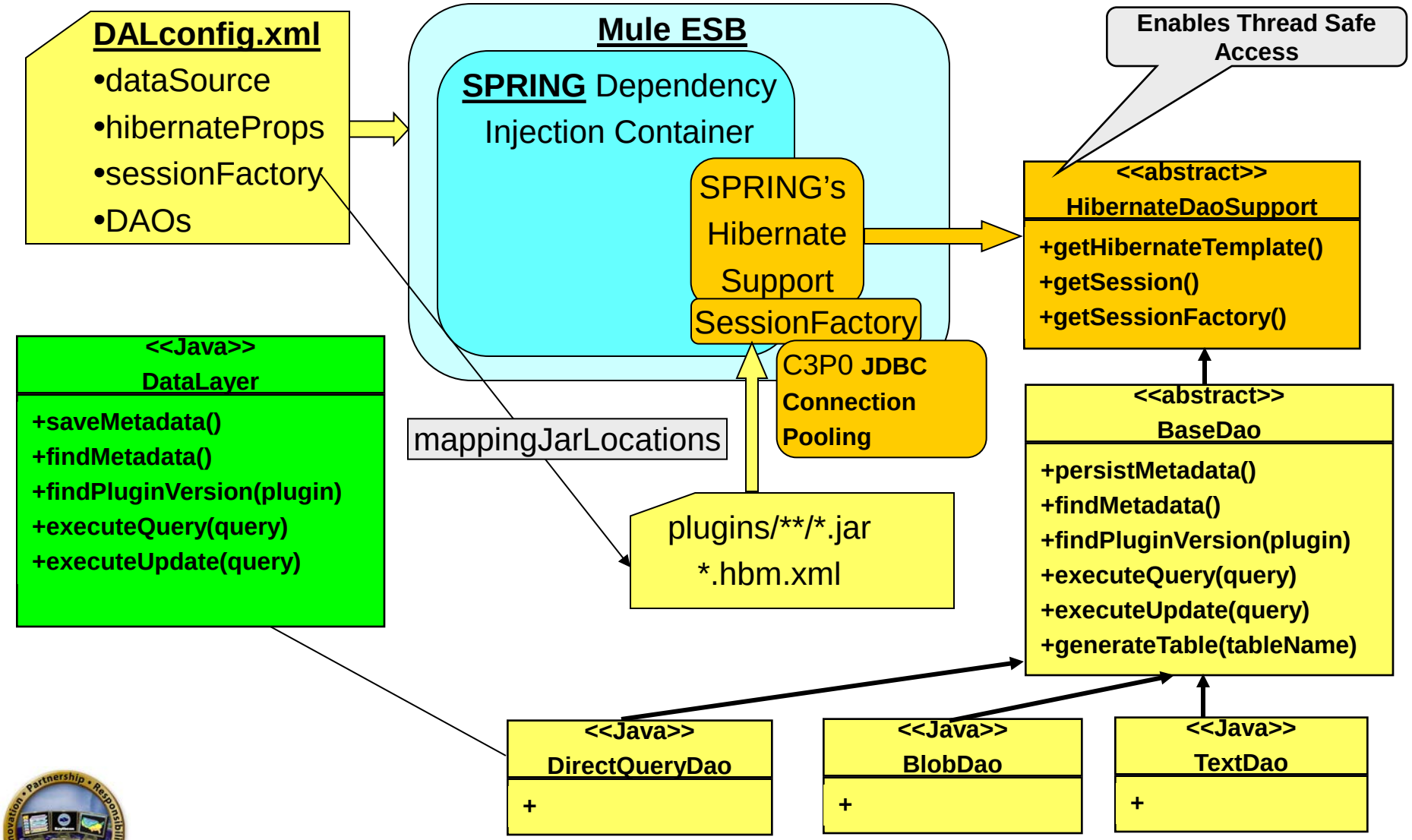
Data Object Model Extended By Plug In Follows Base Model

Plug In Enables Adapting To New Data Types

T05 ADE Data Access Pattern
Completely New in T05 to Support Clustering

Raytheon

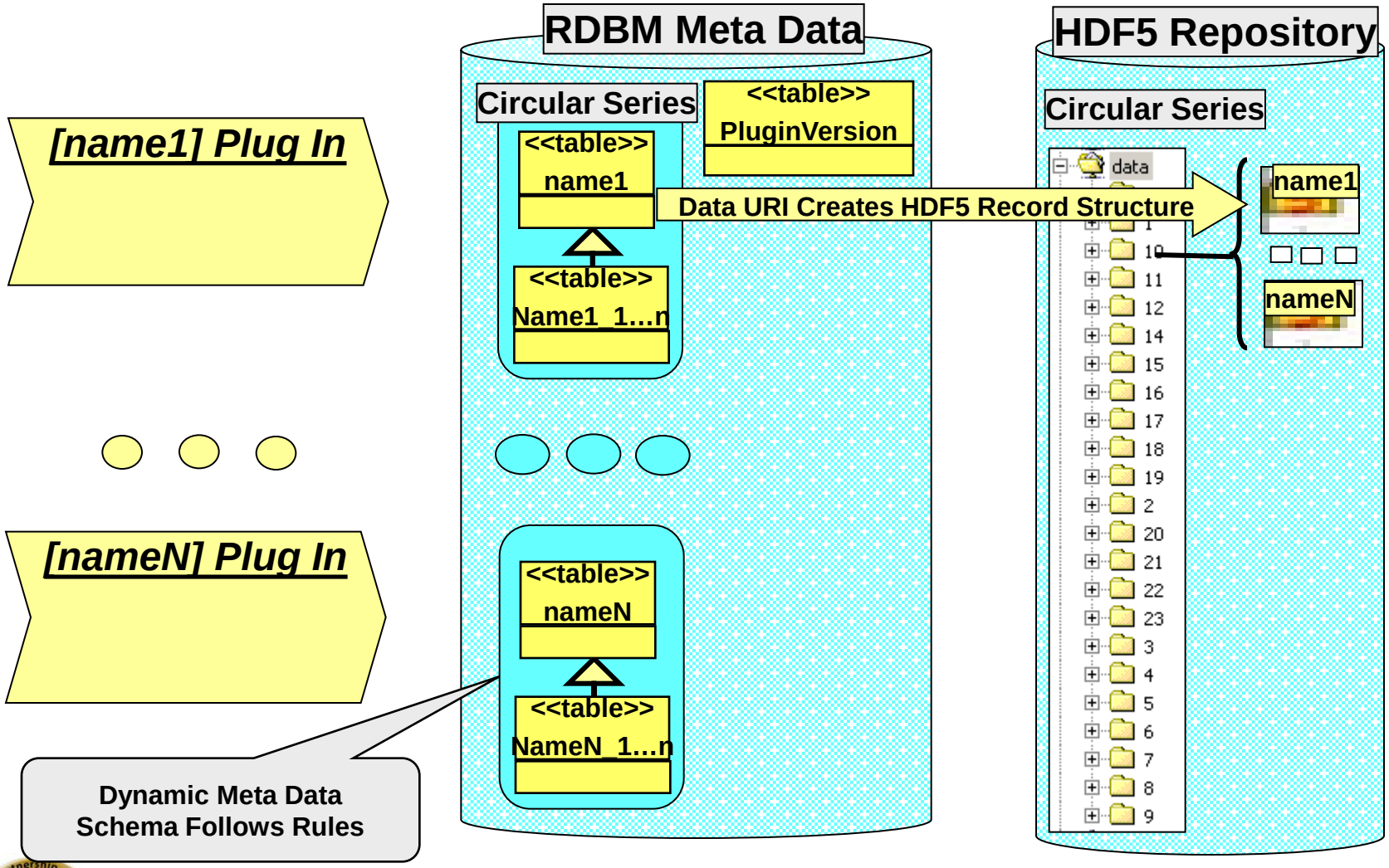
Data Model



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

Raytheon

Data Model

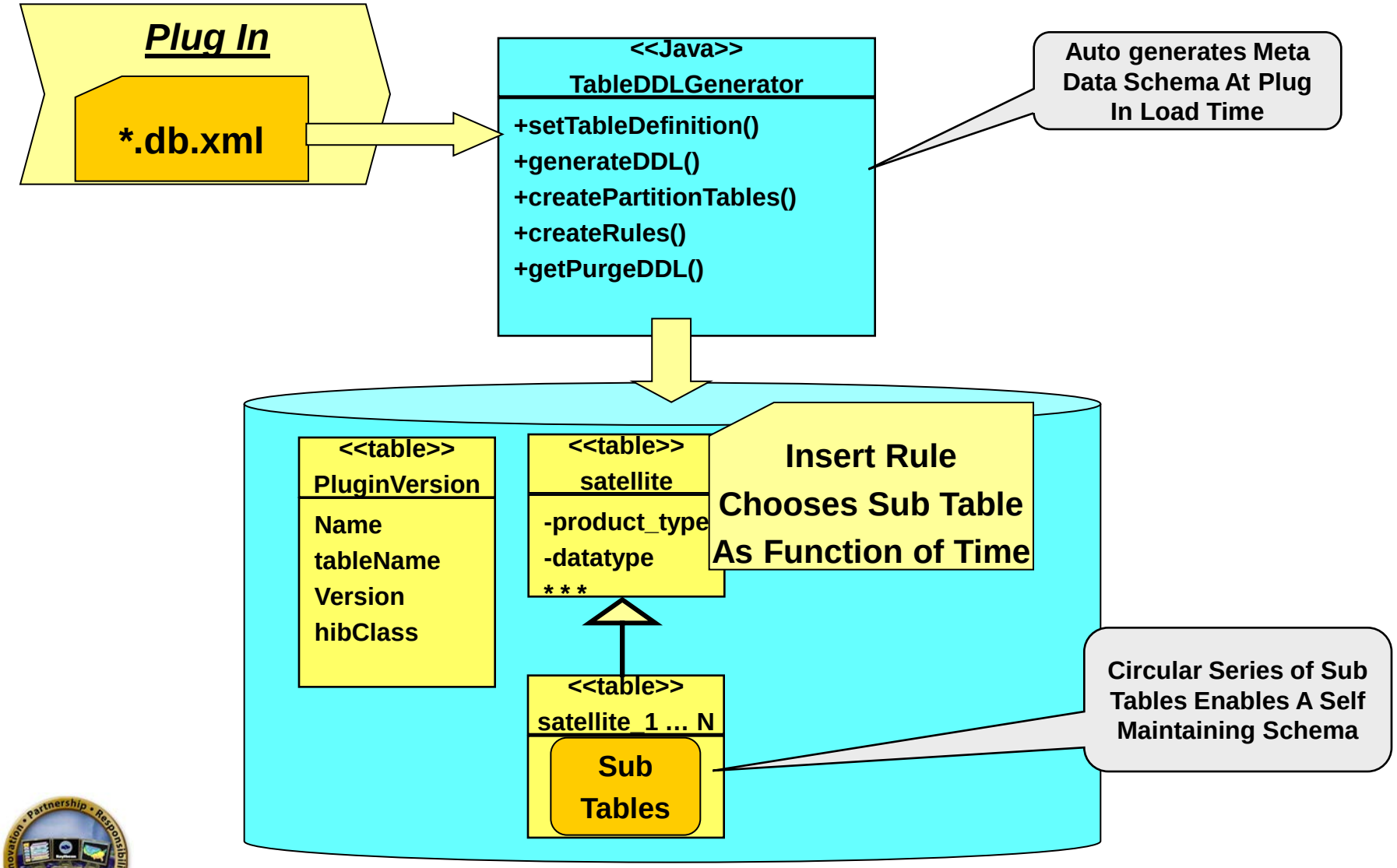


Flexible Data Model is Plug In Extendable To New Data Types

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

Raytheon

Data Model

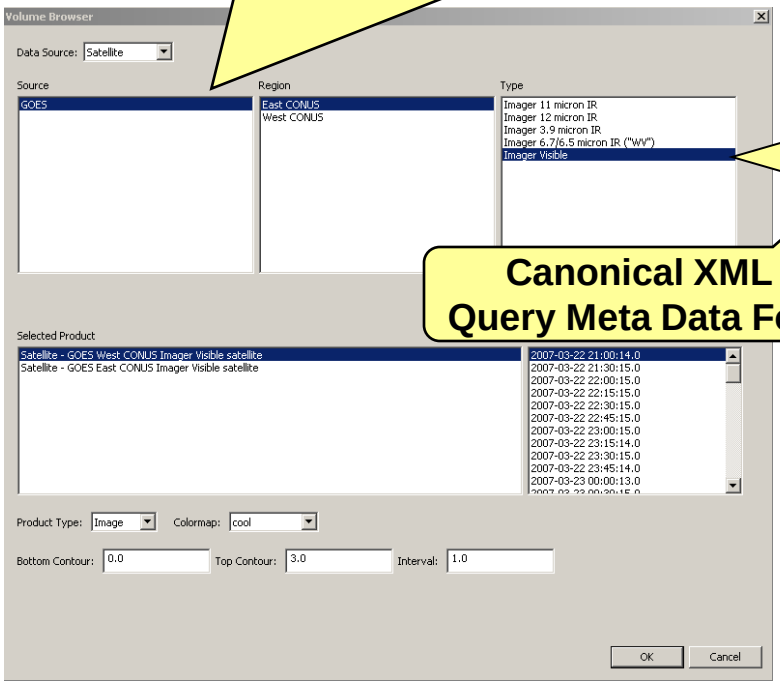


T05 Meta Data Demo
Using CAVE's Volume Browser

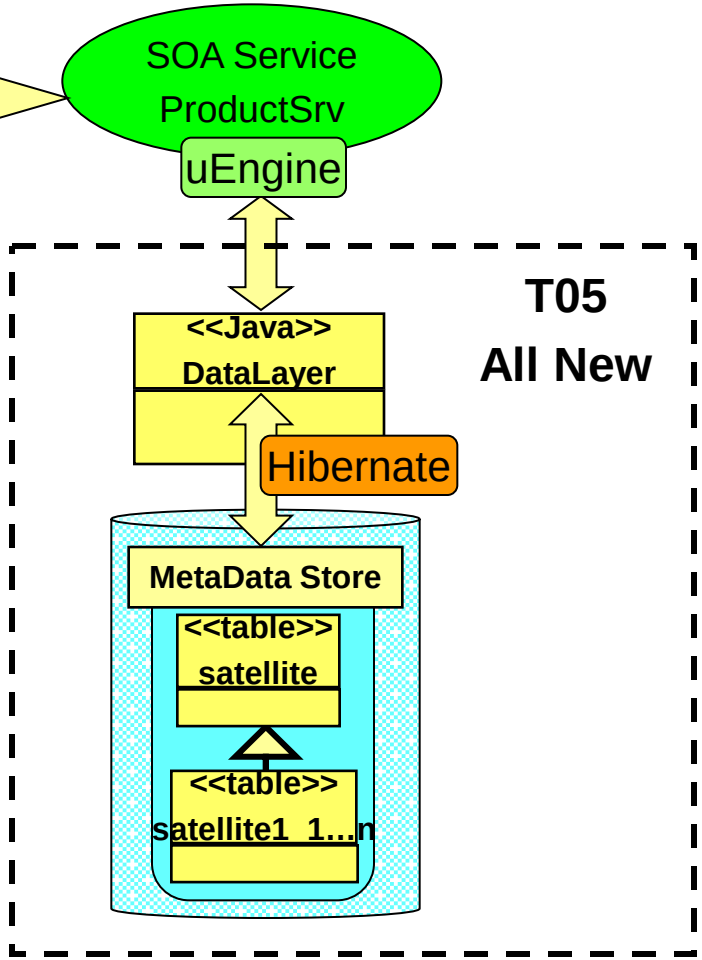
Raytheon

Data Model

Dynamically Populates Select Boxes Through Catalog Queries



Canonical XML Message
Query Meta Data For Catalog



T05 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()

PersistSrv

<<Java>>
SatelliteWriter

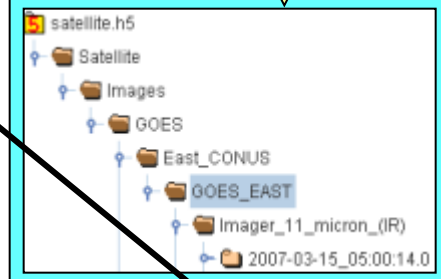
Example Data URI From a LOG File

```
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

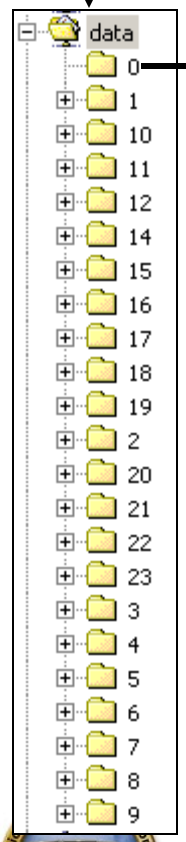
Data Model

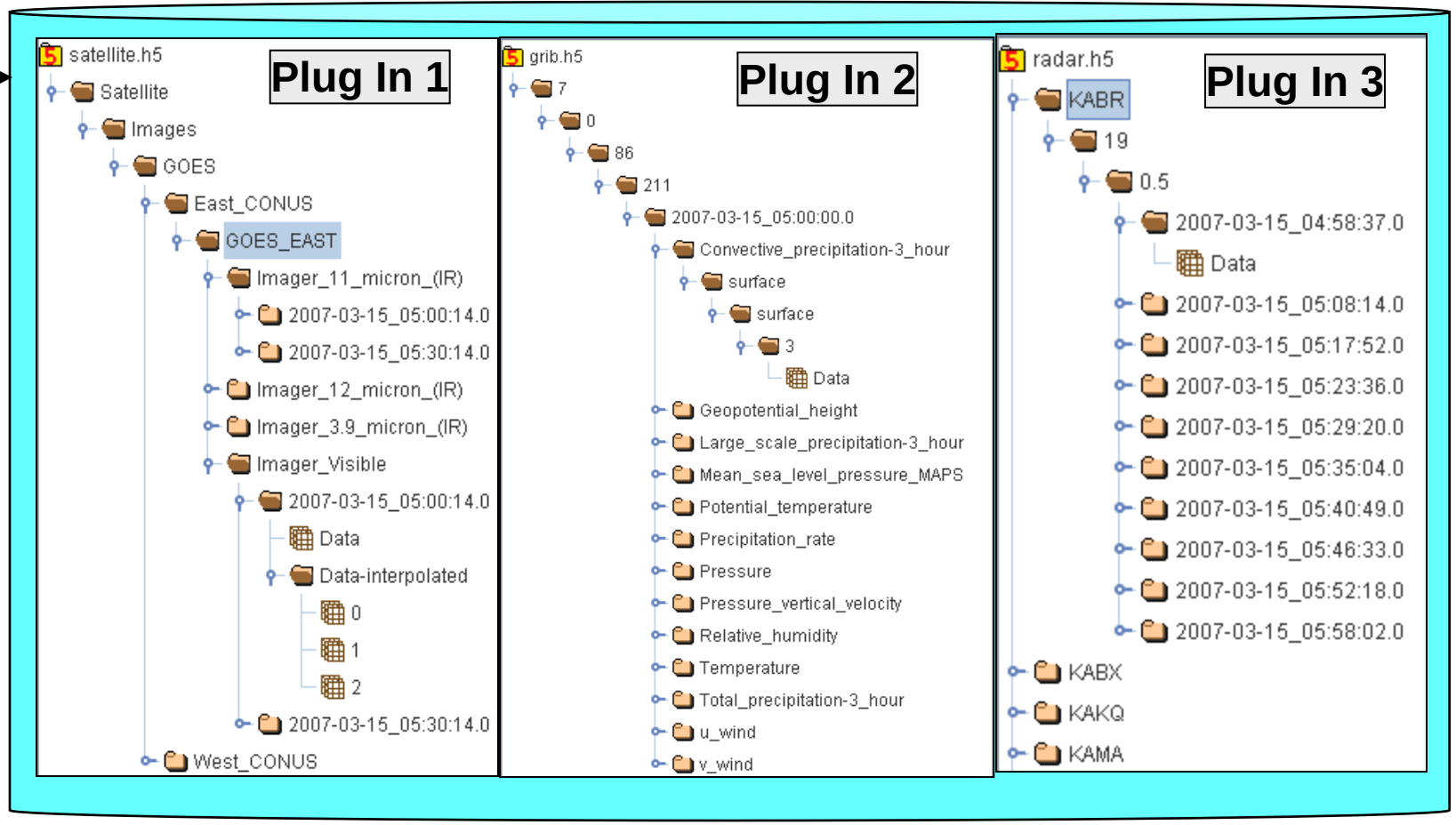
Circular Time Bins

/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]
```

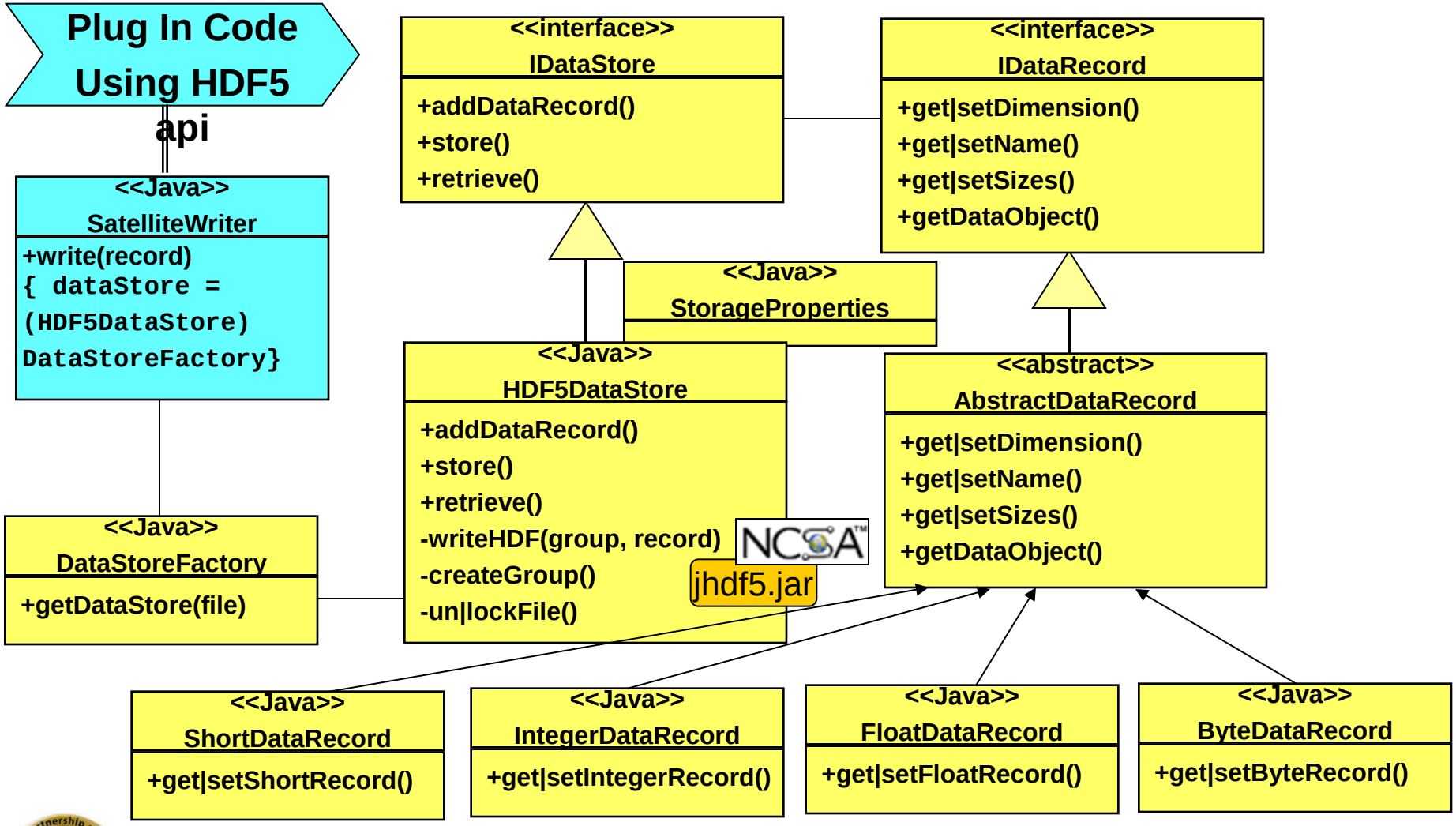




T05 Data Persistence Using HDF5
Application Code Interfaces Through API

Raytheon

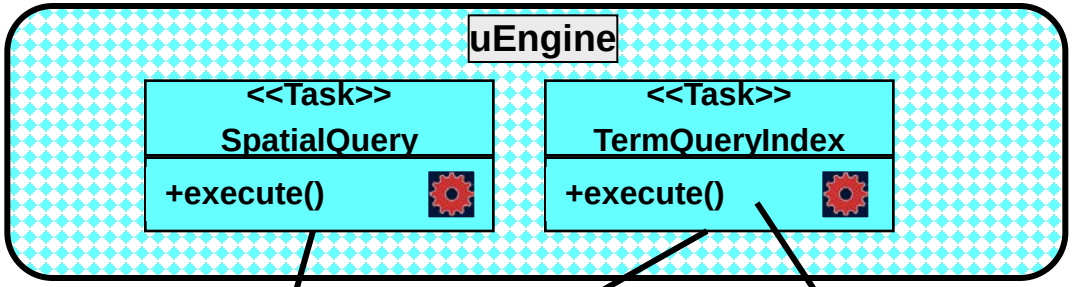
Data Model

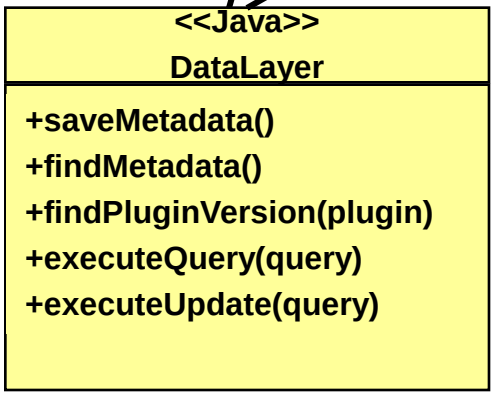


uEngine Using the Data Access Layer
Single API enables uEngine to access all data

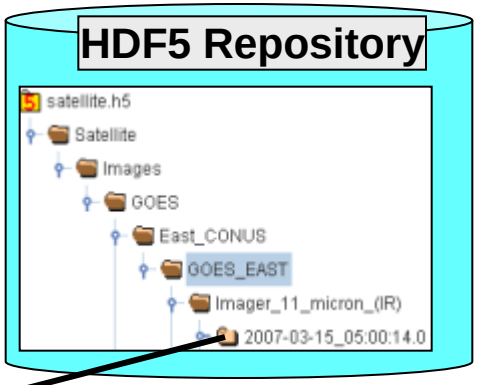
Raytheon

Data Model






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

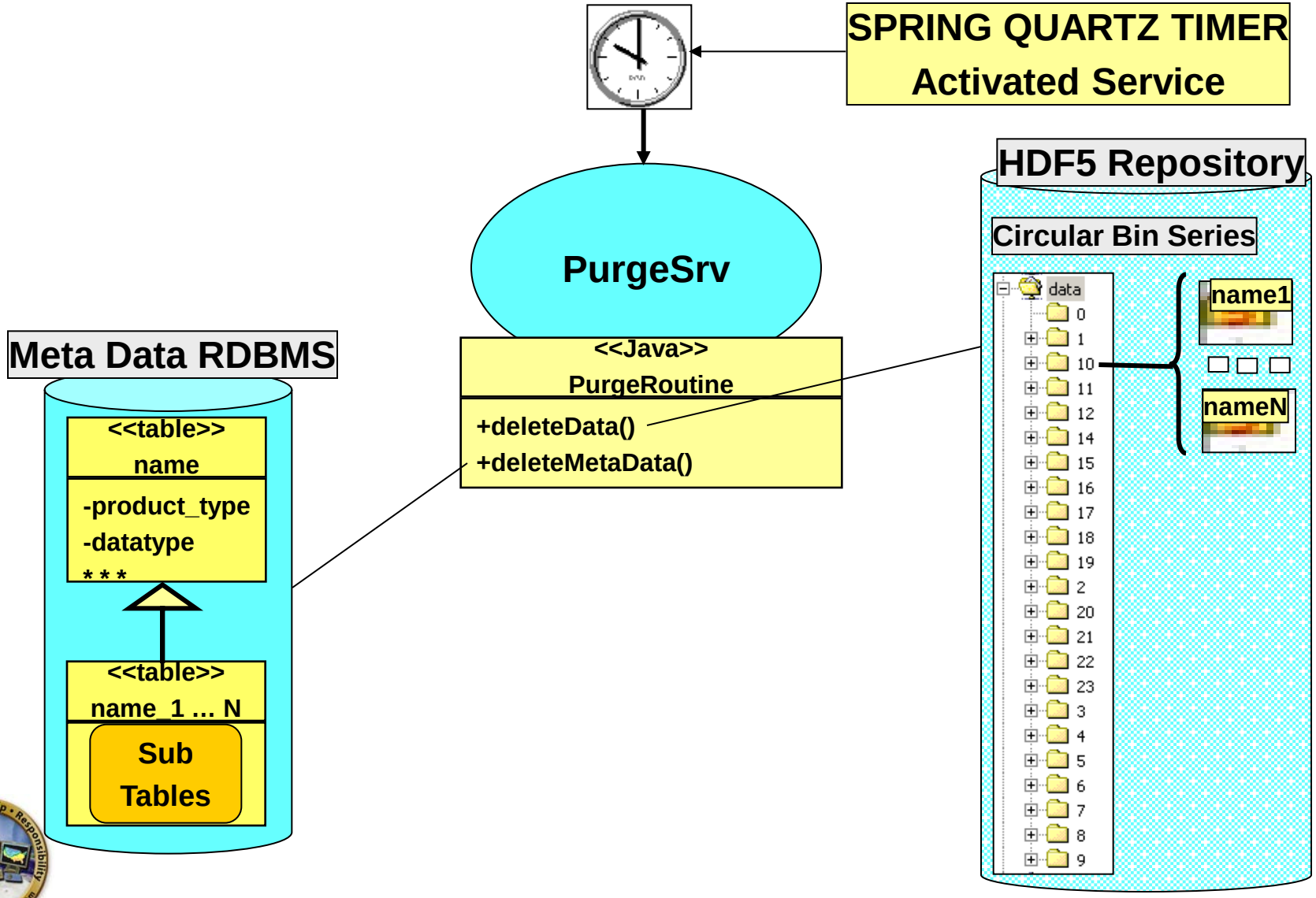


Dynamically
Creates URI Reference
to
Data Record

T05 Purging Data: Self Maintaining
Drops MetaData Tables & HDF5 Bins Periodically

Raytheon

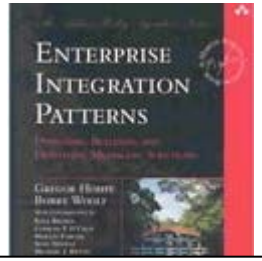
Data Model



T05 ADE 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
- **Ross Mason** identified these problems back around 2001 and developed the **Mule Project** which now has a large and active community
- A parallel development by Progress Software created **Sonic ESB** and they coined the term
- **Mule ESB** is based on "Enterprise Integration Patterns" and is the most elegant extensible solution
- The **high decoupling** of Spring's Dependency Injection Container with Mule'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

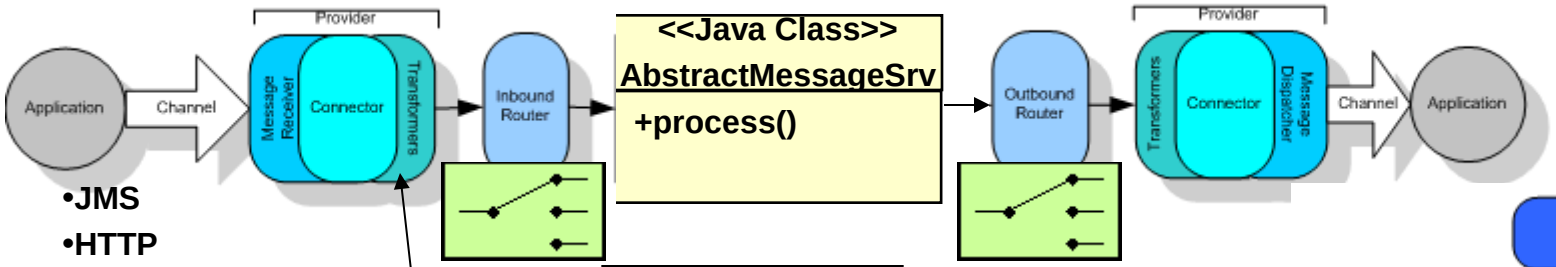
Mule ESB + Spring Becomes a Discriminator

Mule ESB Concepts – Enabling Flexibility
T05 Leverages Several Features

Raytheon
Server Execution

Endpoint Abstraction

ADE SOA Service

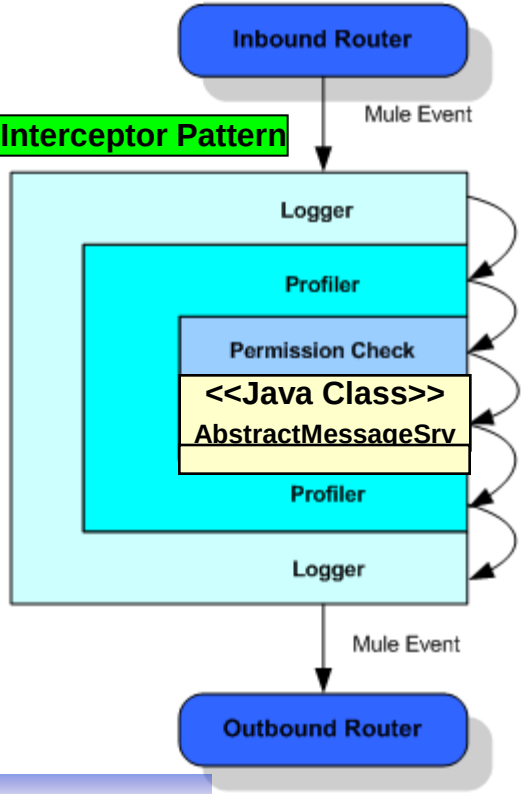


- JMS
- HTTP
- FTP
- E-mail
- TCP
- UDP
- Quartz
- SOAP
- *** 20

Extendable Transformers

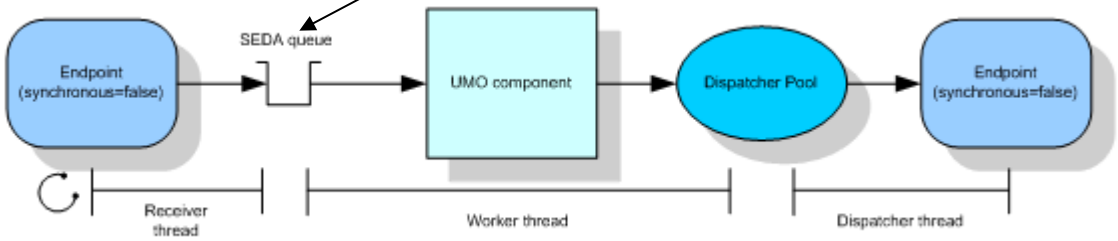
Fan In/ Out Routing

Extendable Interceptor Pattern



Asynchronous Endpoint Processing

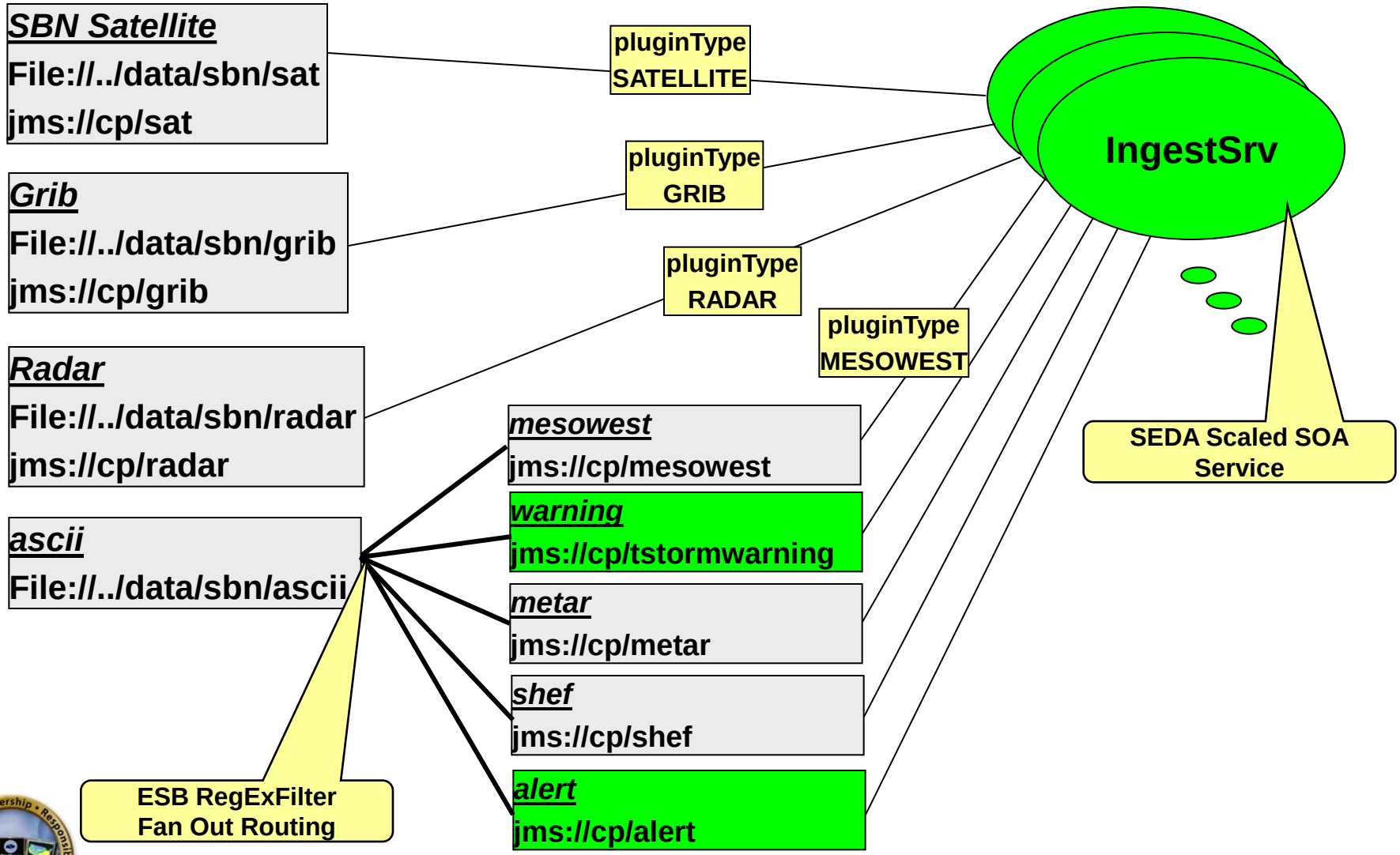
SEDA Component Scaling



Most Advanced Enterprise Service Bus

T05 Data Ingest Endpoints
Warnings Routed to Their Own Endpoint

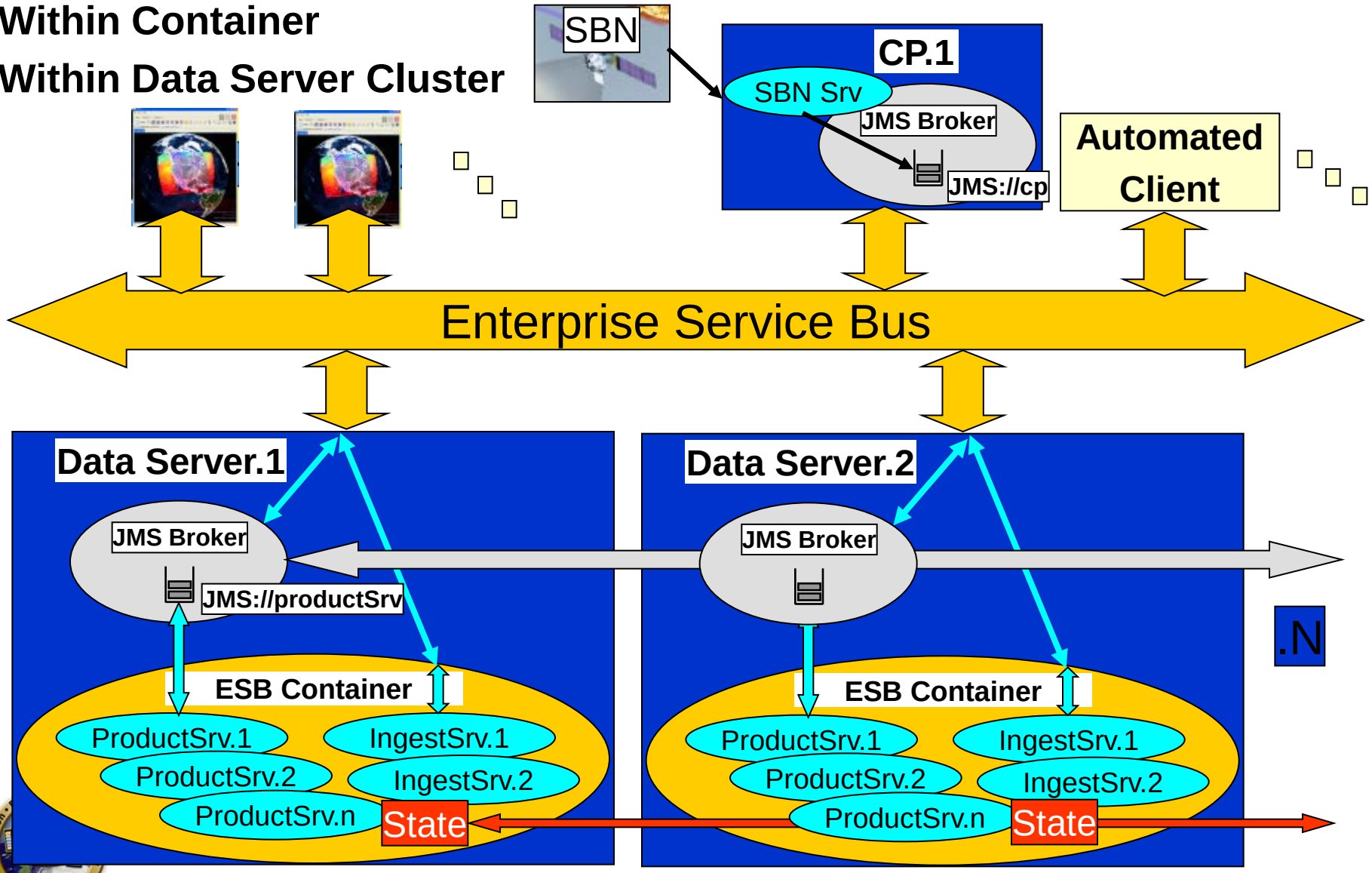
Raytheon
Server Execution



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

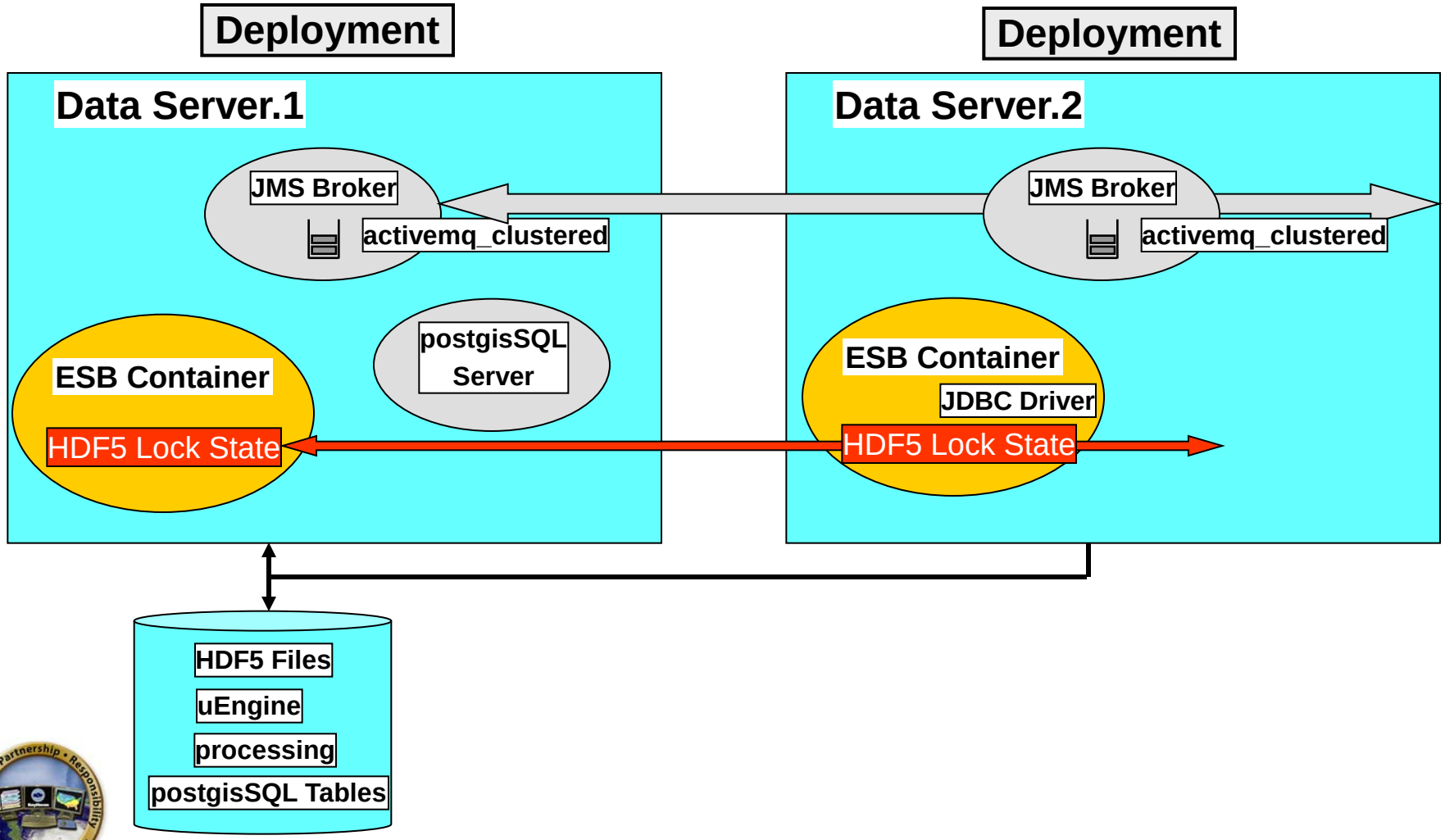
Raytheon
Server Execution

- Within Container
- Within Data Server Cluster



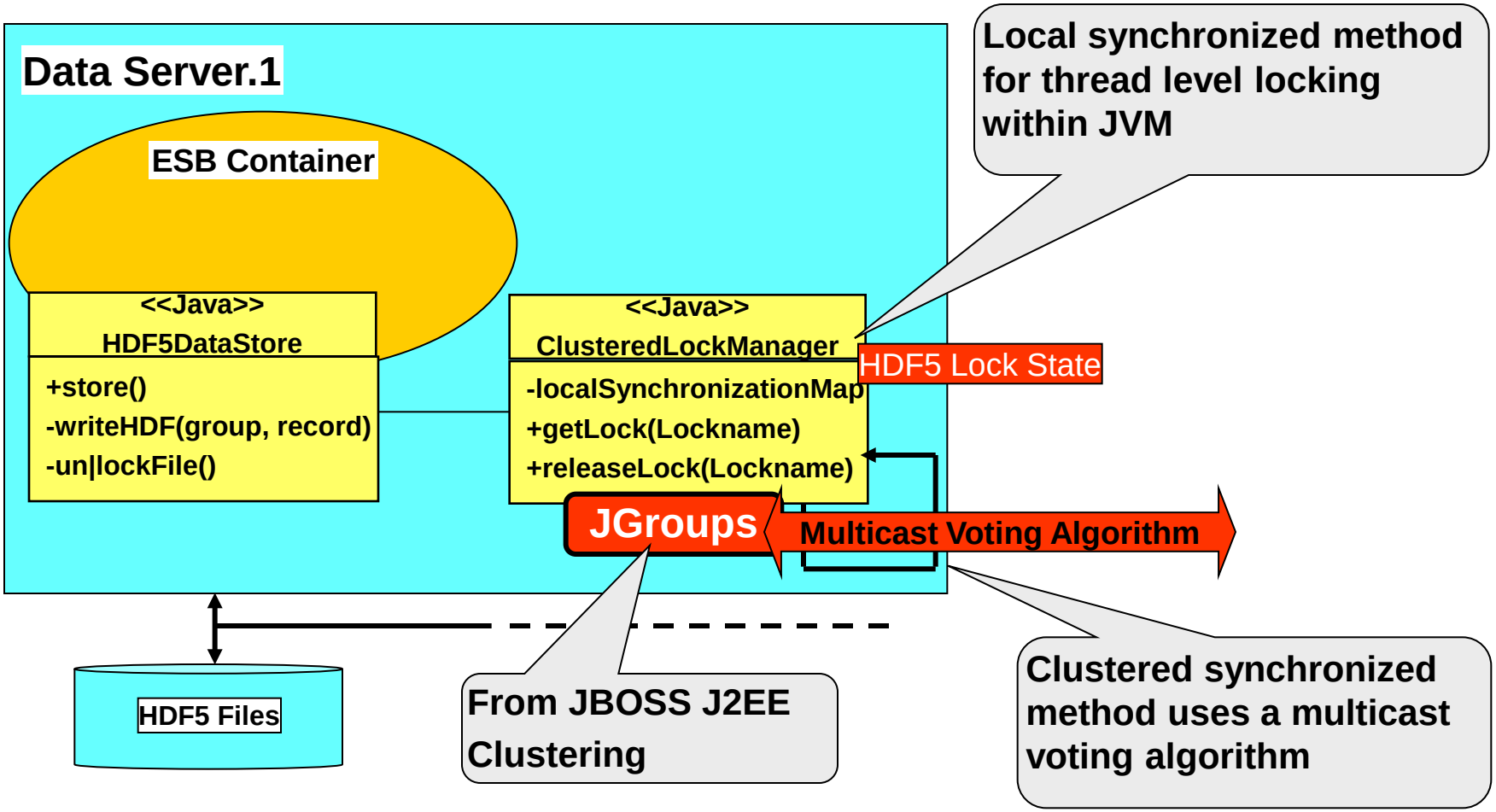
T05 Server Deployments
Clustered JMS Broker Enables SEDA load balancing

Raytheon
Server Execution



Server State Synchronization of Locks
Ensures Data Integrity of HDF5 Writes

Raytheon
Server Execution



File Level HDF5 Locking For Cluster

T05 Plug In: uEngine Tasks Extensions
Eclipse IDE Views of Plug In Implementations

Raytheon
Server Execution

T05 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

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

Note: Each Plug In Has a uEngine Task Package

Radar Plug In

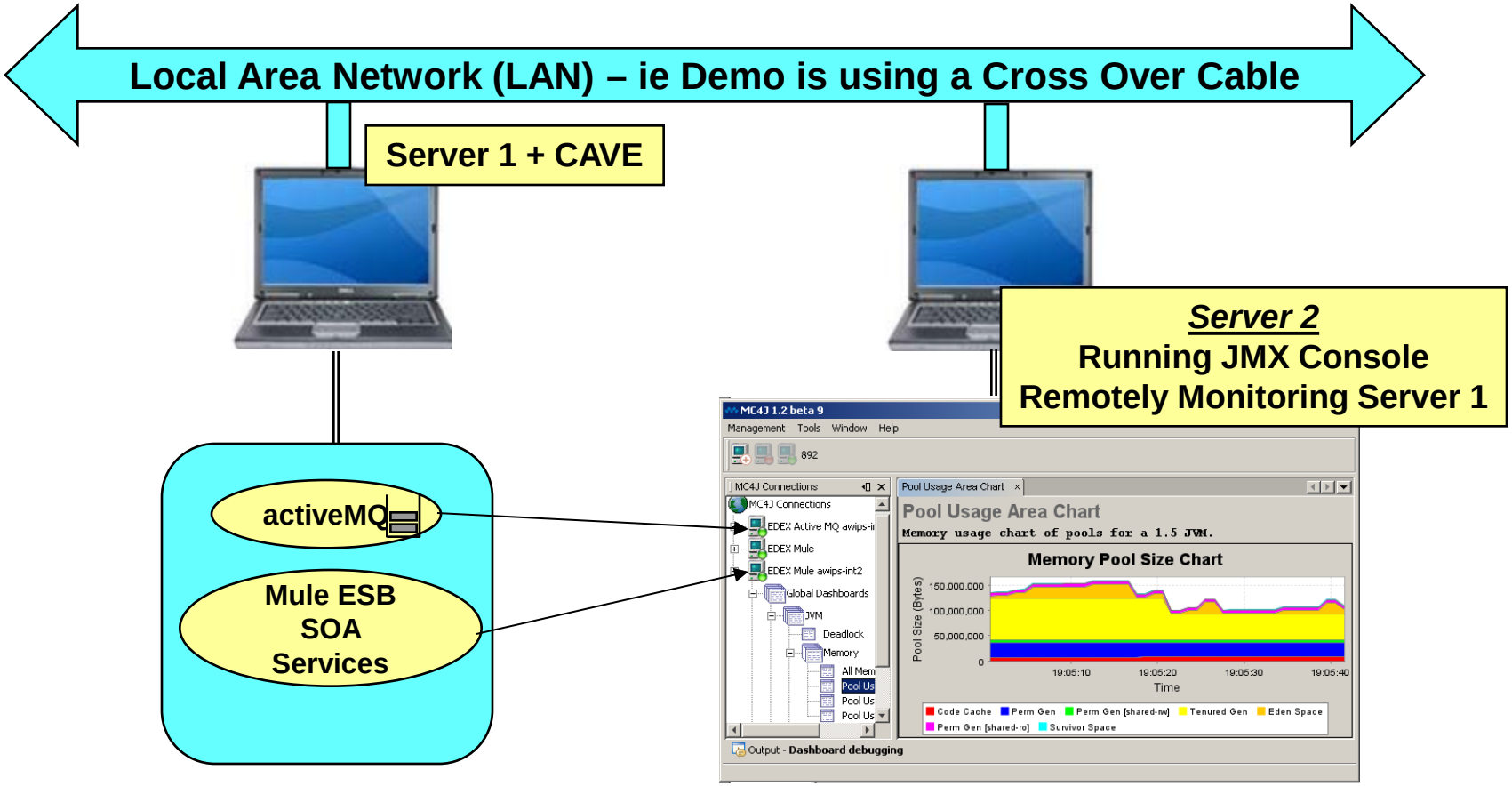
- extensions/plugin-radar/src
- com.raytheon.edex.plugin.noaa.imagery.radar
  - GateContainer.java 1133 3/3/07 3:05 PM brock
  - ShefRecord.java 1076 2/24/07 11:47 AM garme
  - package-info.java 1000 1/30/07 3:18 PM mfegar
  - RadarTiler.java 1177 3/8/07 12:12 AM brockwoo
  - RadarWriter.jav
  - RadialContainer.
  - RadialLayer.java 1177 3/8/07 12:12 AM brockwo
  - RadialLayer8bit.java 1177 3/8/07 12:12 AM broo
  - 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/
- 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

Remote Management Through JMX
Gives Detailed Insight Into Remote Java JVMs

Raytheon
Server Execution

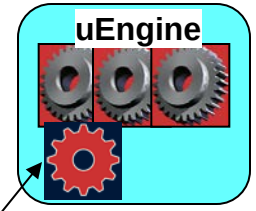


Plug Ins Enable Adapting to New Data
Automatically Creates Meta Data Persistance

Raytheon
Server Execution

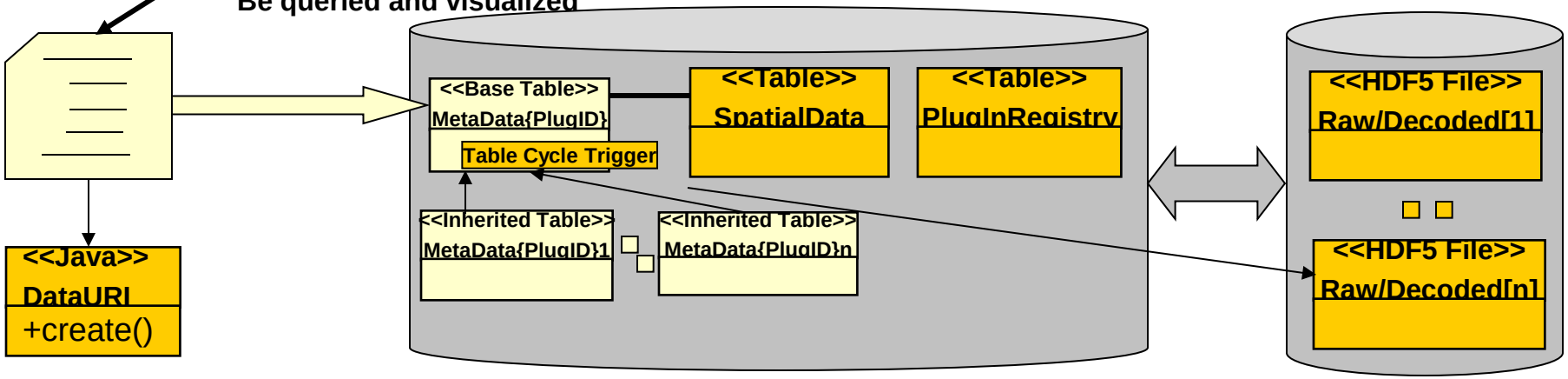
Reference Architecture
- IProduct{Extract Metadata}
- I{*}Record
- IDecode
- IStore
- Task
- Discovery

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

T05 Alerting Pattern Based on Ingest

Warning GIS located on Map and Displayed

Raytheon

Server Execution

ascii

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

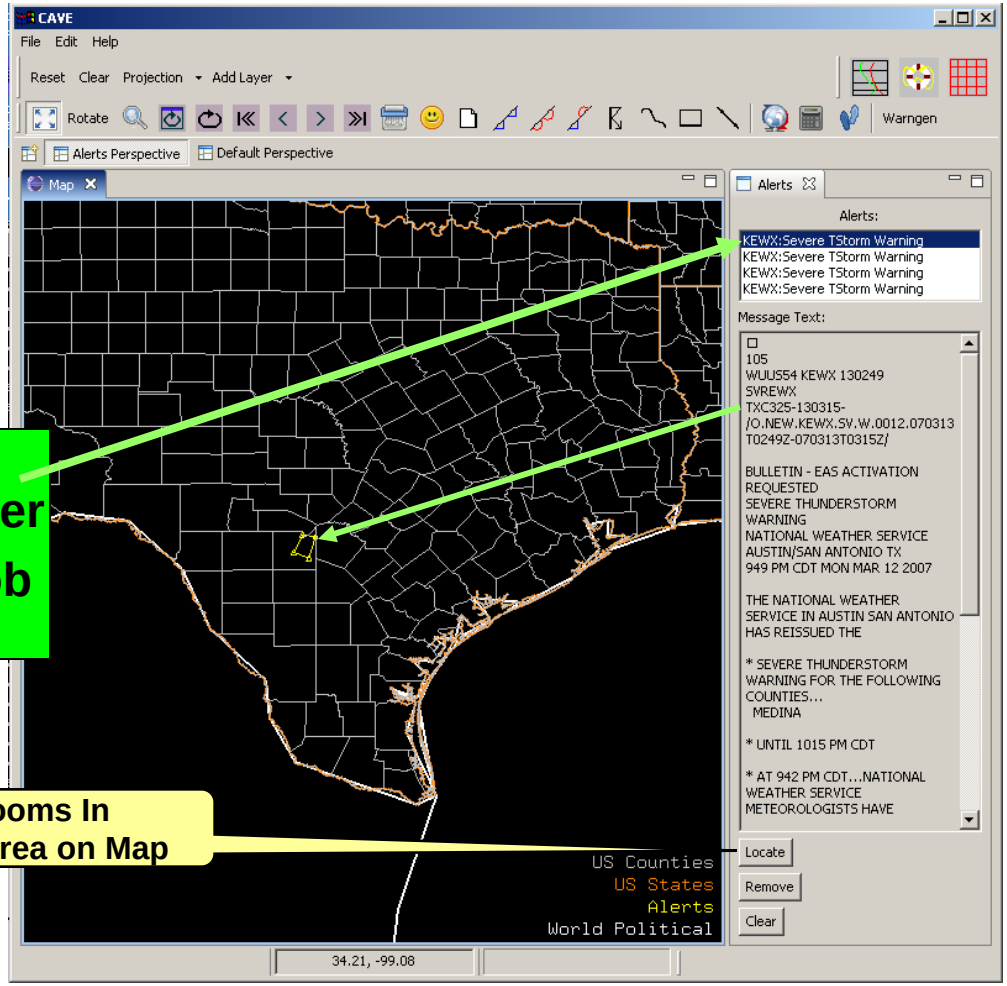
ESB RegExFilter
Fan Out Routing

Threaded Listener
AlertManagerJob

warning

jms://cp/tstormwarning

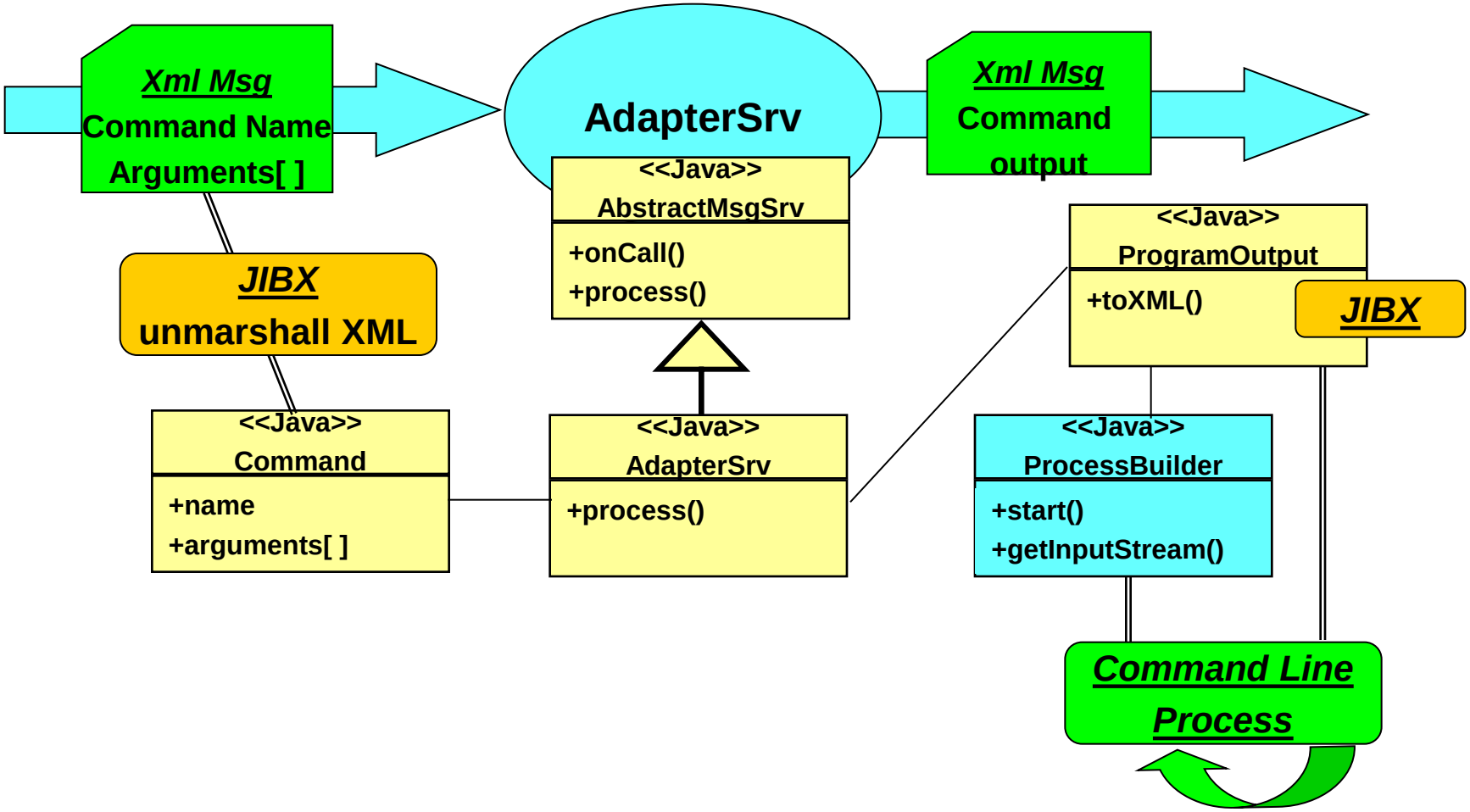
Locate Zooms In
To Warning Area on Map



T05 Legacy Adapters Enhancements
AdapterSrv Receives XML messages of commands

Raytheon

Adapters

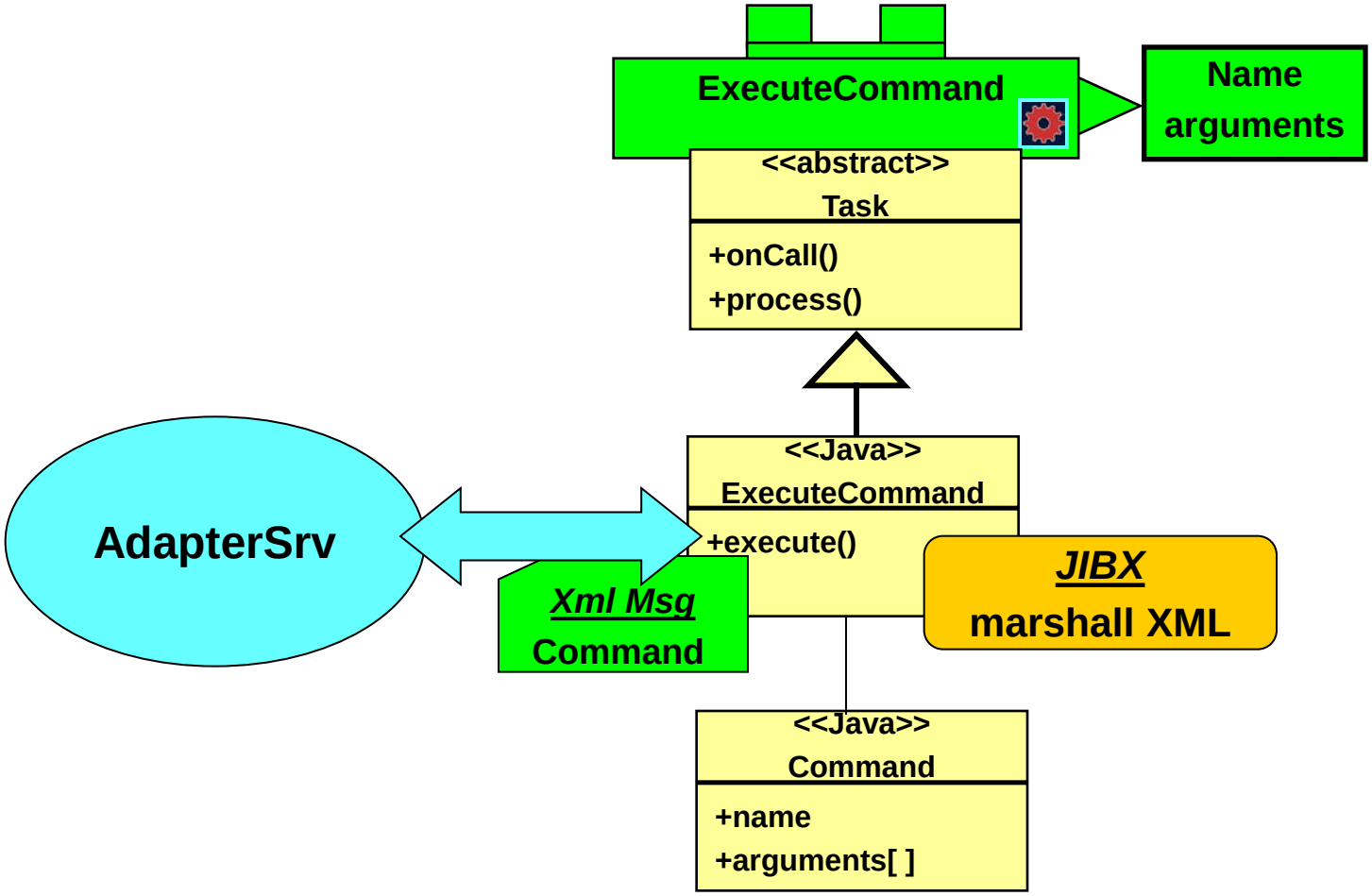


Enables Traditional Command Line Process to run as SOA Service

T05 Legacy Adapters Enhancements
uEngine Runs Command Within Action Script

Raytheon

Adapters



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

T05 CAVE (Visualization) Enhancements
Most Significant Update Since Project Began

Raytheon
CAVE Visualization

- Eclipse RCP 3.2 based, GeoTools 2.2, JOGL 1.0, ANTLR, 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 re-projection
- Implemented Display Bundles
  - JIBX extension for Resource to Bundle mapping enables quick adaptation to new visualization resource
- High Performance radar rendering similar to D2D all-tilts
  - On demand rendering
- Active raster pixel data integration
- Warning generation with ANTLR parsing of templates

T05 CAVE Display Bundles
Enables Adding Procedures and History Later

Raytheon
CAVE Visualization

XML Document

Bundle Definition

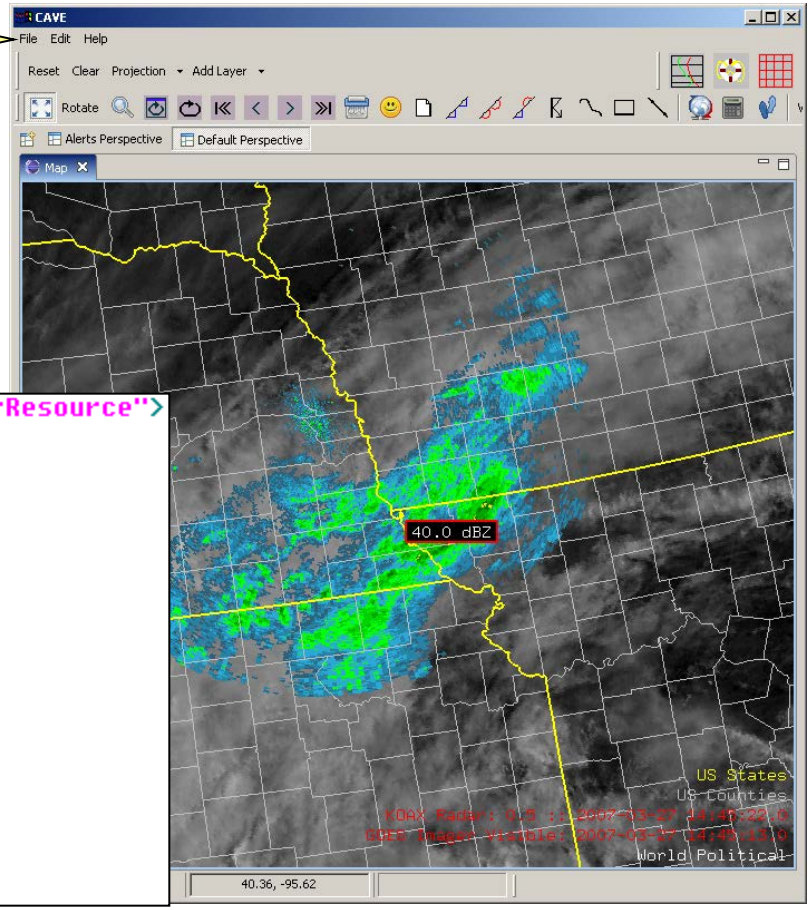
+layers

+color maps

+line widths

****

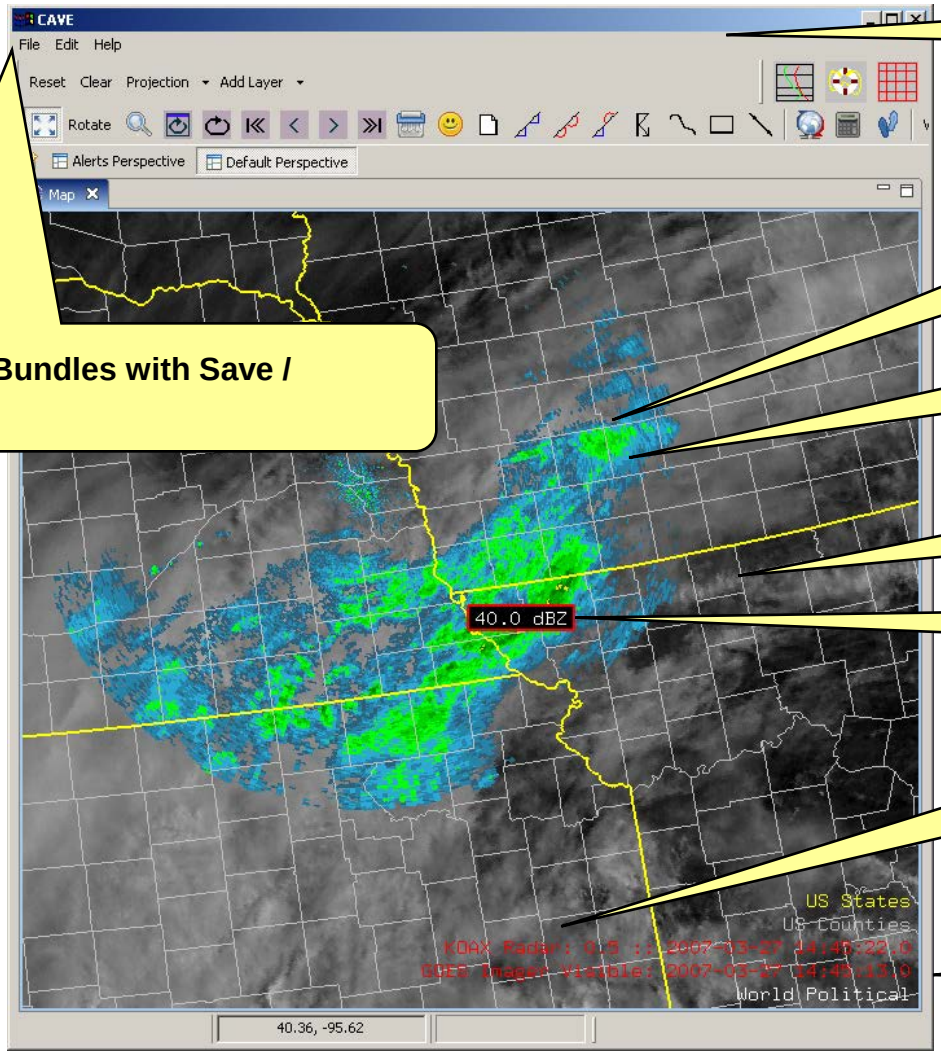
```
<resource class="com.raytheon.viz.core.rsc.radar.RadarResource">
 <properties>
  <value field="colorMap">COLORMAP { radar }</value>
  <value field="colorMapMin">0.0</value>
  <value field="colorMapMax">255.0</value>
  <value field="contrast">1.0</value>
  <value field="brightness">1.0</value>
  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



T05 CAVE Visualization – Service Endpoint
Enables Gaming Style Data Interactions

Raytheon

CAVE Visualization



Eclipse RCP 3.2 – Plug In Extendable
T05 Plug In for Warn Generation Added

T05 Radar Rendering Uses Dynamic raster tiling

T05 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 re-projection using GeoTools Transforms& GPU Warping

All Tilts Keyboard Controls

T05 Radar Rendering
Radial Data Gets Tiled Dynamically

Raytheon
CAVE Visualization

<<interfaces>>

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

<<java>> RadarTileSet

- +createTile()
- +dispose()

<<java>> RadarResource

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

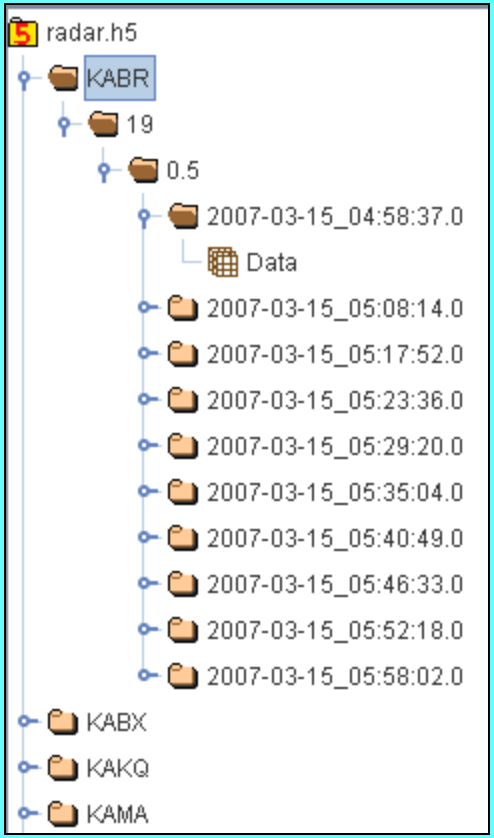
<<Job>> RadarLoaderJob

- +requestLoad()
- +run()

<<java plug in>> RadarTiler

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

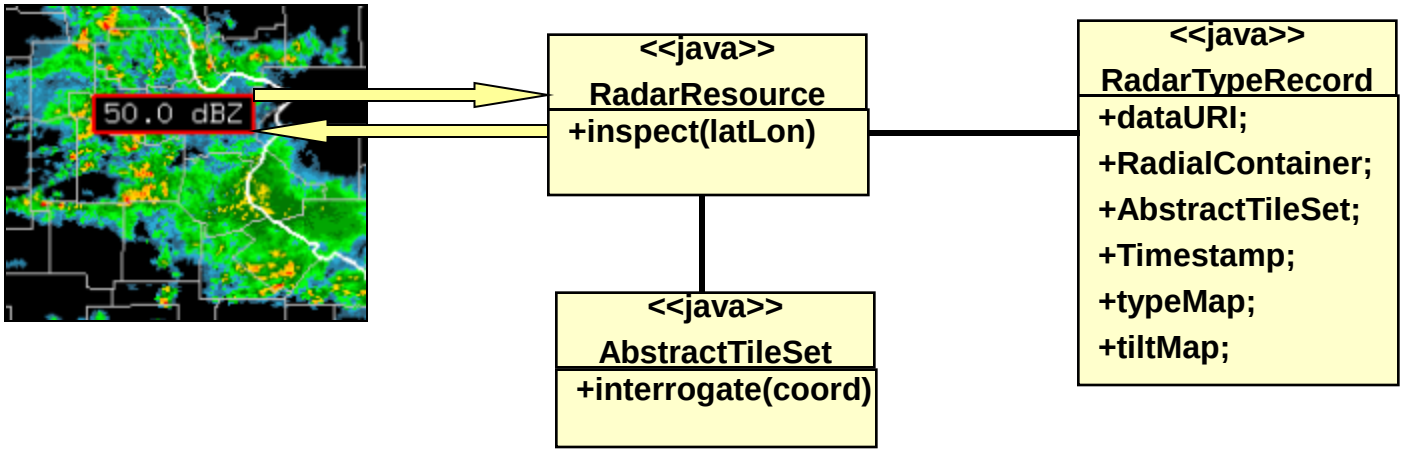
HDF5 Radar Data



T05 Raster Data Interrogation
All raster and shape files are inspectable

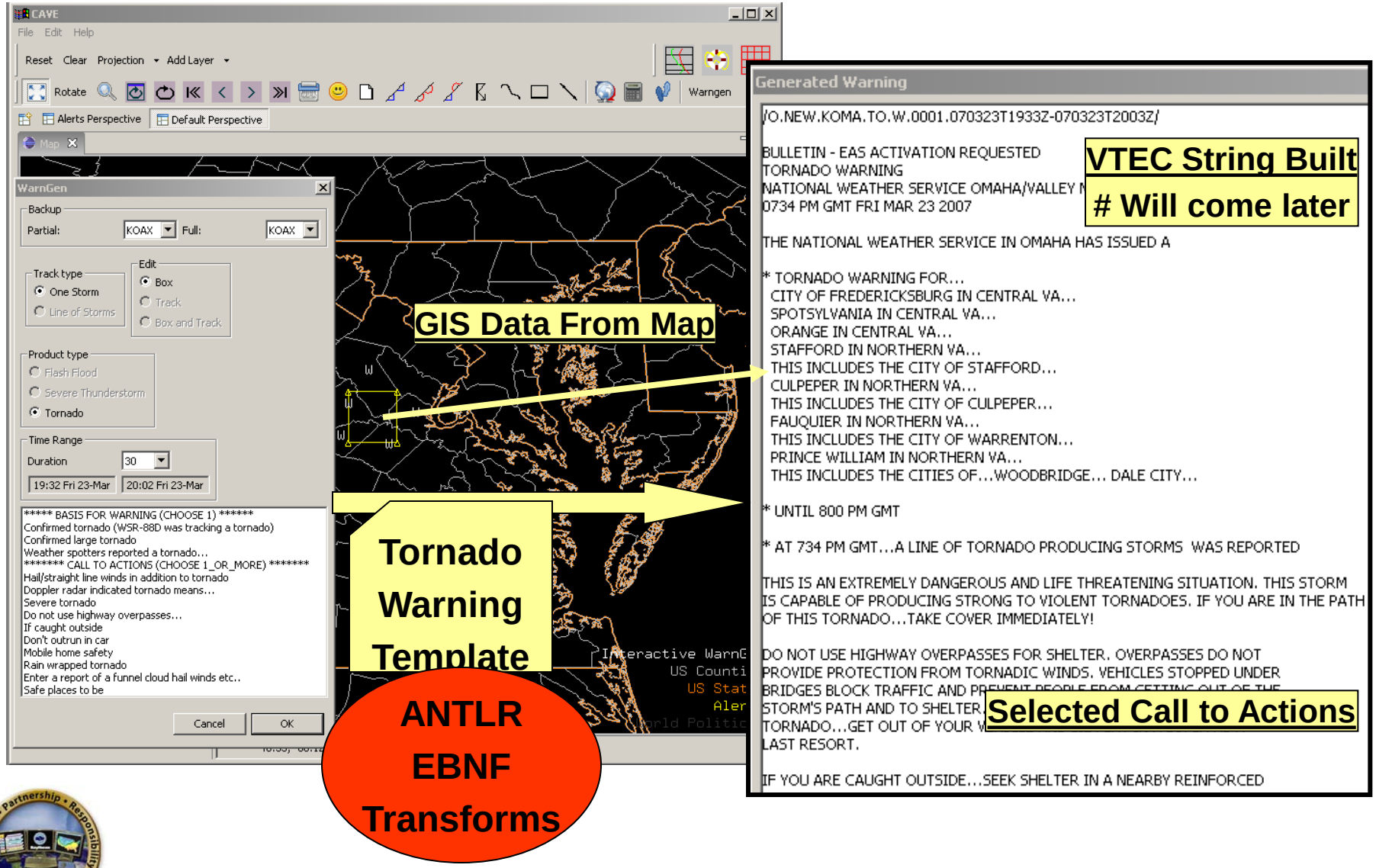
Raytheon
CAVE Visualization

- 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



T05 CAVE Warning Generation Pattern
ANTLR Template Parsing for Warning Text

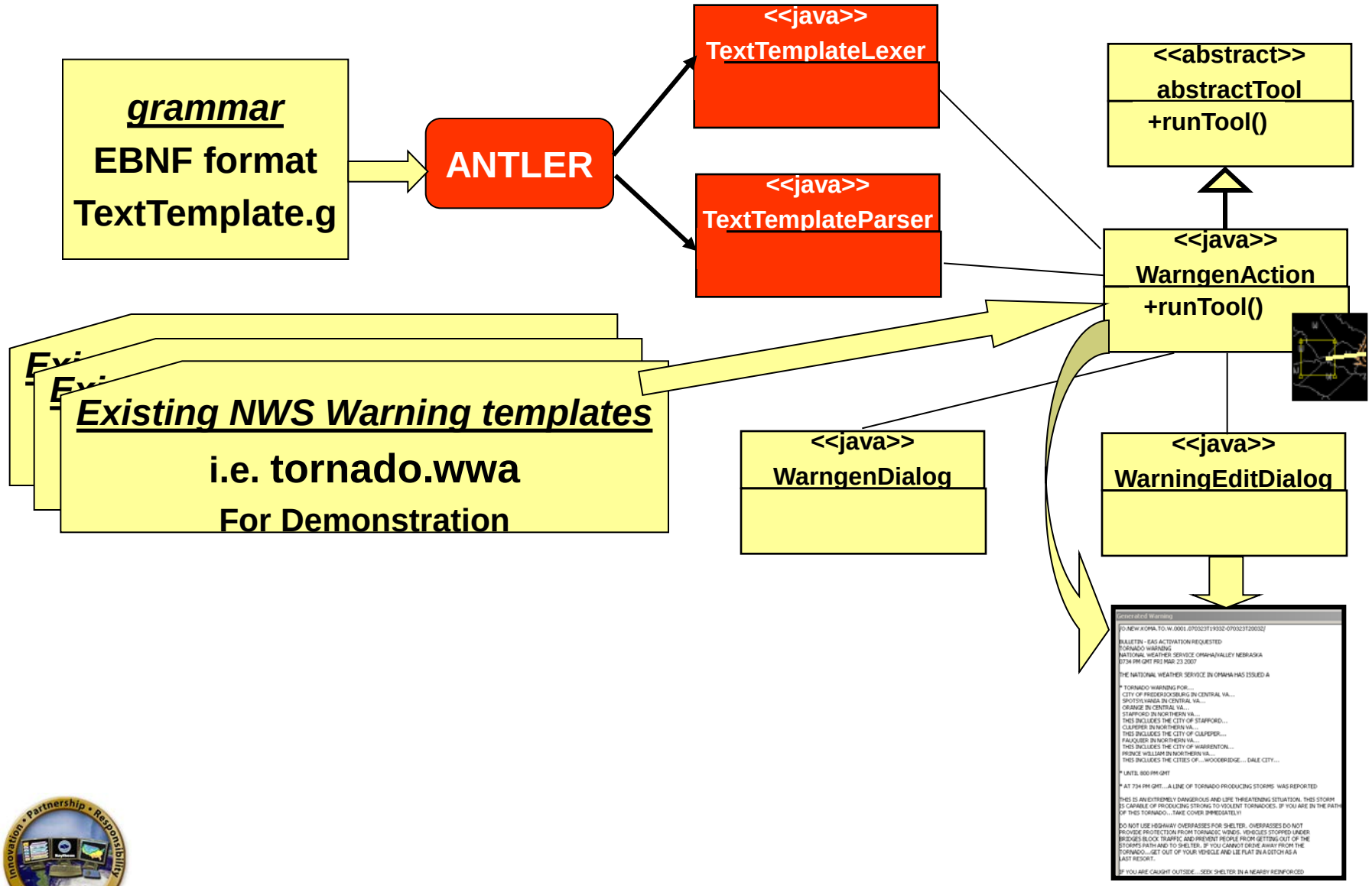
Raytheon
CAVE Visualization



T05 Warning Template Approach
Reuse Existing WWA Templates

Raytheon

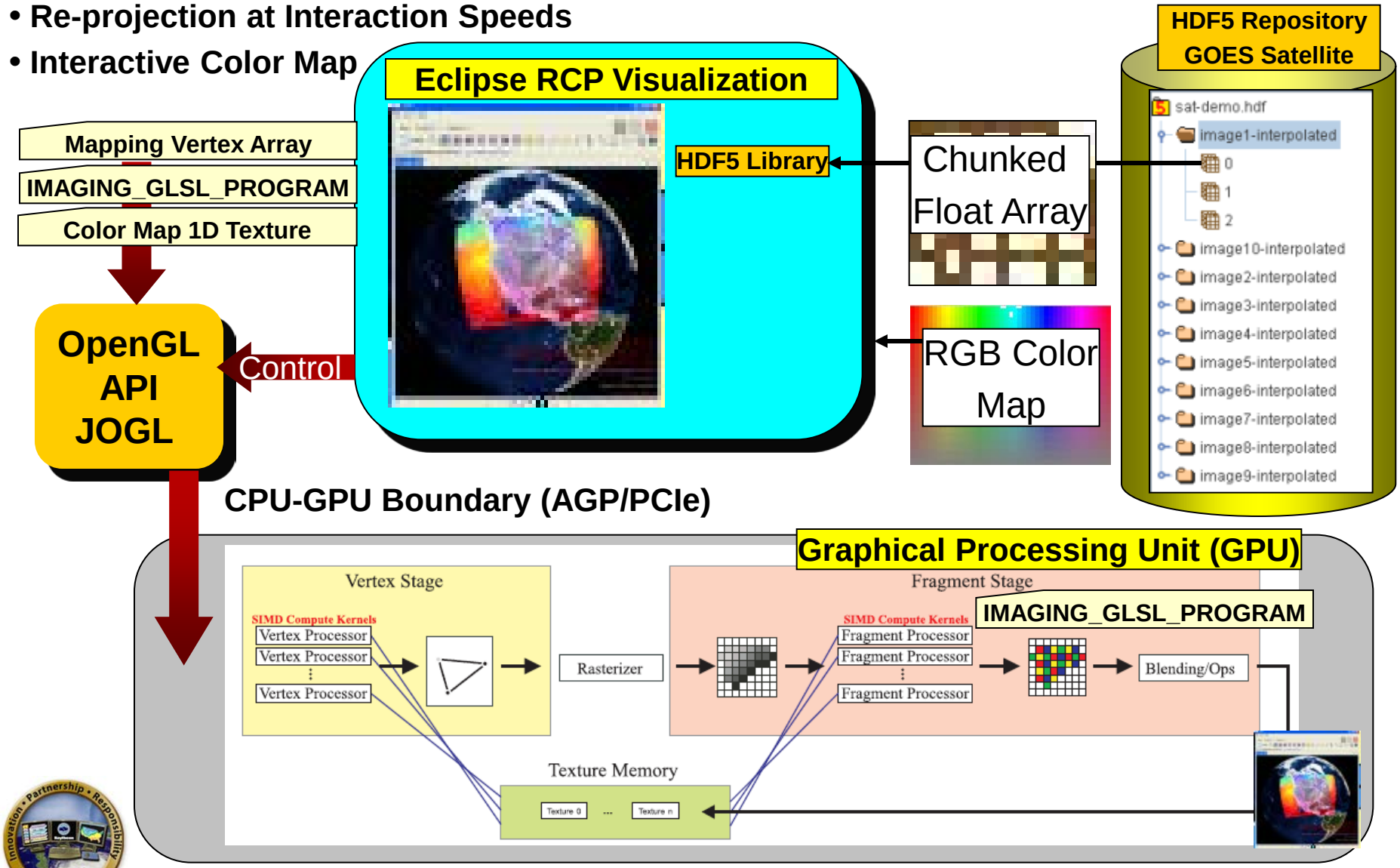
CAVE Visualization



Map Re-projection: Texture Warping
GPU Fragment Shader 50 times Faster than CPU

Raytheon
CAVE Visualization

- Re-projection at Interaction Speeds
- Interactive Color Map



GPU Fragment Shader Color Mapping
Enables Interactive Color Controls & Interrogation

Raytheon

CAVE Visualization

<<JAVA>>

GLTarget

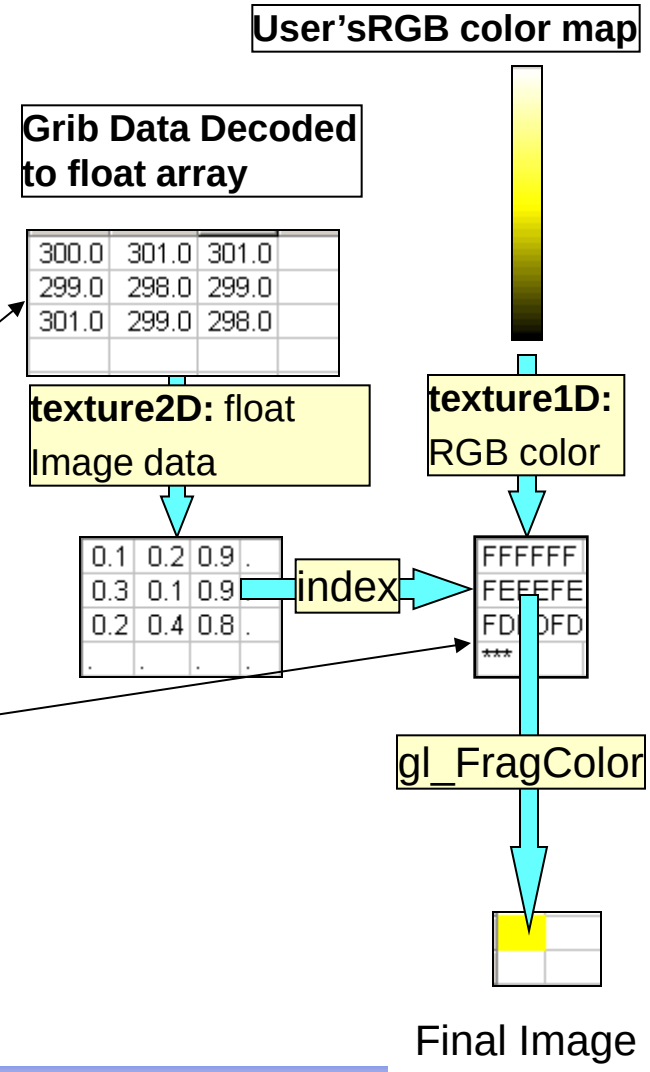
<<texture shader>>

IMAGING_GLSL_PROGRAM

+main()

{ loaded into GPU during Initialization}

```
private static final String IMAGING_GLSL_PROGRAM = "uniform sampler2D tex;"
                + "uniform float brightness;\n"
                + "uniform float contrast;\n"
                                                  "
                                                  ;\n"
                                                  ;\n"
                                                  "
                                                  p;\n"
                                                  ;\n"
                                                  "
                                                  ;\n"
                                                  (0.5, 0.5, 0.5);\n"
                + "    vec4 textureColor = texture2D(rawTex, gl_TexCoord[0].st);\n"
                + "    if(doColorMap == 1) { \n"
                // + " float naturalValue = ((textureColor.r * (20000)) - 10000);\n"
                + "       float naturalValue = ((textureColor.r * (naturalMax - naturalMin)) + naturalMin);\n"
                + "       float index = ((naturalValue - cmapMin) / (cmapMax-cmapMin));\n "
                + "       if(index < 0) {\n"
                + "          index = 0; \n"
                + "       } else if(index > 1) {\n"
                + "          index = 1;\n"
                + "       }\n"
                + "       vec3 color = texture1D(colorMap, index).rgb;\n"
                + "           textureColor = vec4(color, alphaVal);\n"
                + "    }"
                + "    vec3 textureColor3 = vec3(textureColor);\n"
                + "    vec3 adjustedColor   = mix(AvgLuminance, textureColor3, contrast);\n"
                + "    float curAlpha = min(textureColor.a, alphaVal);\n"
                + "    gl_FragColor = vec4(adjustedColor.r * brightness, adjustedColor.g * brightness,\n"
                + "                        adjustedColor.b * brightness, curAlpha);\n"
                + "}";
```



Interactive Color Maps 20 to 50 Times Faster Than CPU

T05 Security Architecture Concepts
No significant change

Raytheon

Security

- Government owns C&A
- AWIPS II will:
  - Continue to employ perimeter security (similar to AWIPS I)
  - Replicate current controls
  - Revise controls as required/directed (NOAA, NWS, AWIPS Directives)
- AWIPS II Goal: 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

T05 Security Concepts Continued
SOA infrastructure has built in capabilities

Raytheon

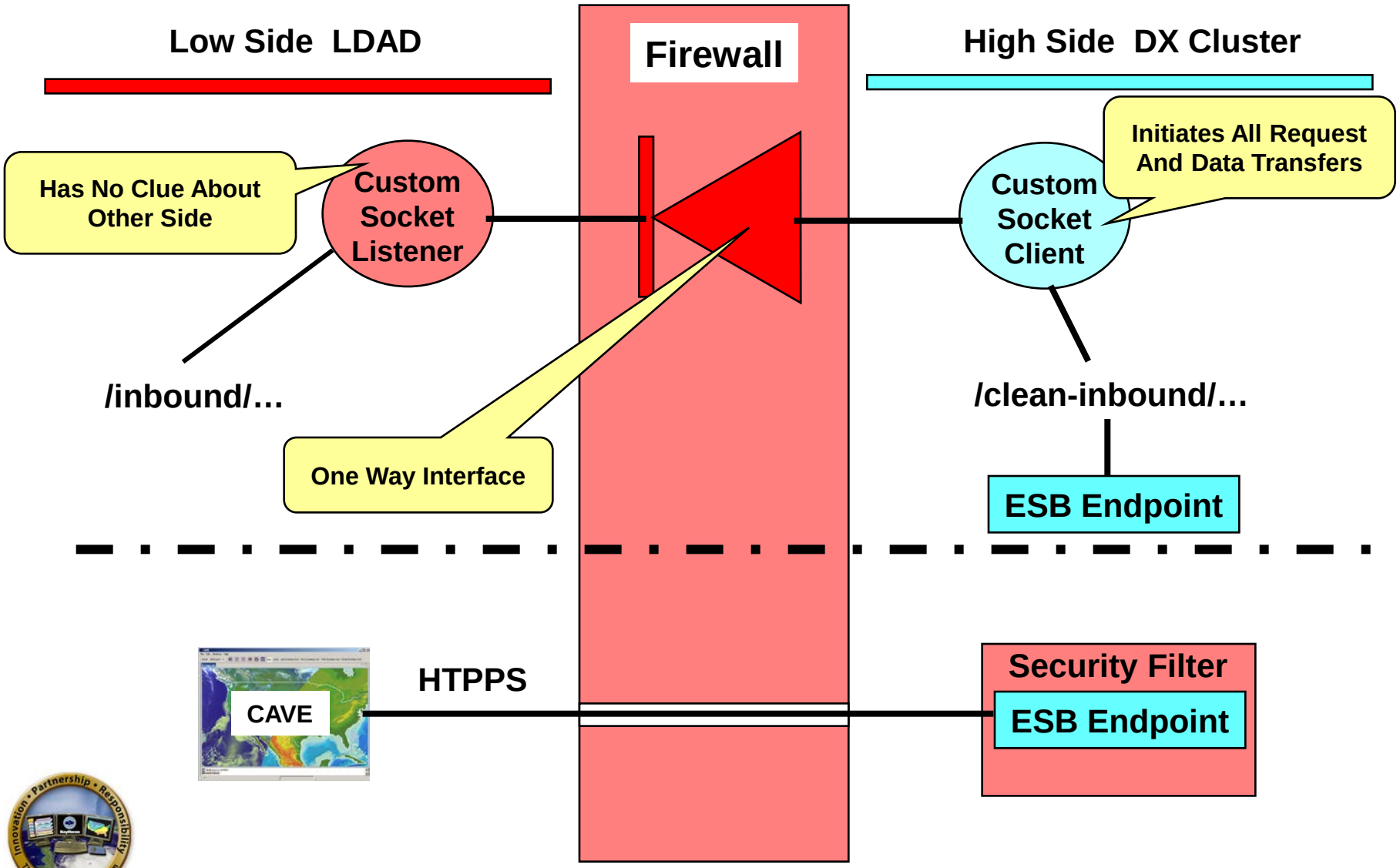
Security

- **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

T05 Security Architecture
Data Ingest Concept of Data Diode for Firewall

Raytheon

Security



Summary –

End

- T05 Achieved the original goals of server side deployment and clustering
- T05 CAVE rendering performance significantly enhanced and used to implement Radar All Tilts
- T05 Clustering Validated on Live SBN ingest data feed
- T05 Gives a solid foundation for T06 hardening
- ??s