

# GENERAL DYNAMICS

Canada



## Rapid Technology Integration Framework

Mike Gingell, P.Eng.  
16 May 2012



# Introduction

---

- Who we are, What we do
- The problem to solve
  - Changing times!
  - Changing customer expectations and needs, vendor lock-in concerns, cost...
  - Changing business factors – need for speed, agility and efficiency
- Our use of Apache open source integration and messaging products

# Canada's Largest Defence Employer Rated #2 Defence Company in Canada in 2012!



- **GD Canada**
- Ottawa
- Calgary
- Halifax



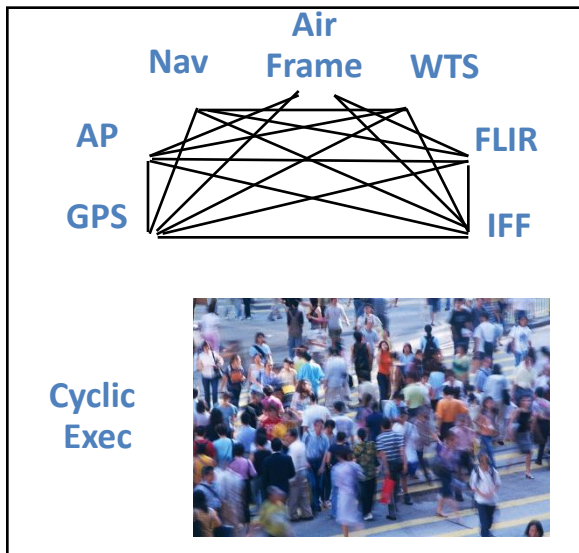
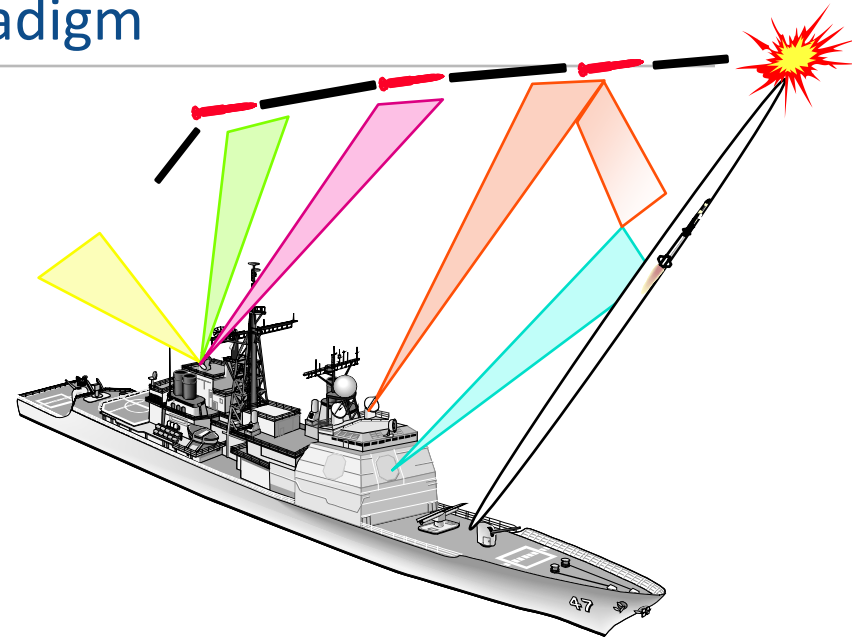
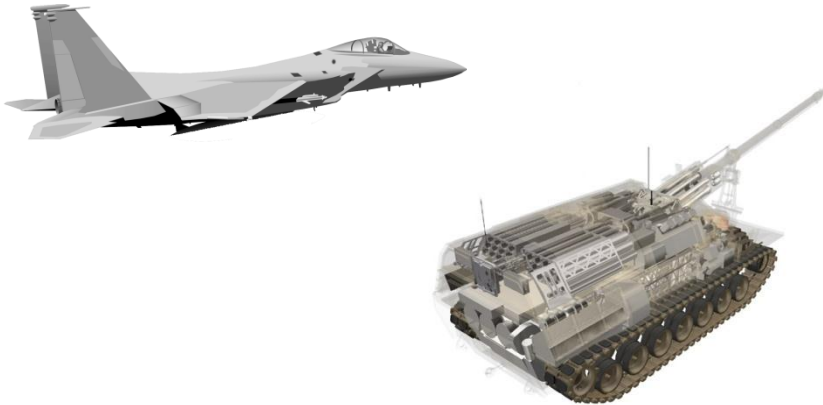
- **GD Land Systems – Canada**
- London
- Edmonton



- **GD OTS – Canada**
- Montreal
- Quebec



# The Problem – Current Design Paradigm



- Legacy systems are designed to be:
  - Stove-piped
  - Proprietary
  - Brittle and non-adaptive
  - Expensive to develop & evolve
  - Vulnerable



# Options

## ■ Traditional commercial solutions

- Expensive
- Hard to customize
- Not optimized for military applications
- Not always interoperable
- No access to source if qualification necessary



## ■ Roll your own

- Often brittle
- Not usually designed for re-use
- Fits application space
- Expensive to build and maintain
- Don't always fully embrace standards that facilitate *re-use* and *interoperability* (at many levels – messaging, marshaling, data)
- Often miss opportunities to build services that could/should be part of the architecture (discovery, service versioning) – only build what you need (or can...)



# Options

## ■ Open source?

- Had a stigma attached to it, especially in the defence community
- Pressures from customer and need to become more efficient and lean have driven people to take a closer look at OSS
- Licencing can still be an issue – GPL, etc.
- Apache License is a commercial friendly license
- Apache community has many mature and active projects that can be used together!
- Apache community is vast! Many hands make light work...
- FuseSource helped make the “right” choice for components in the stack!



## So – back to the problem...

---

- Our problem domain: Command and Control and Situational Awareness
  - Collect and *integrate* data, *mediate* and *route* information from/to many different applications in a timely manner so intelligent decisions can be made...

**A perfect match for Apache Camel!**

- Integration to other applications using **W3C** standard interfaces; i.e. RESTful and SOAP

**A perfect match for Apache CXF!**

- Reliable *transfer* of information through the network
- Message oriented middleware such as JMS is required

**A perfect match for Apache ActiveMQ!**



## So – back to the problem...

---

### ■ Service Management



- In previous installations many service components were required
  - Message broker, HTTP and Servlet container, Discovery services
  - Had to be deployed separately, security and access control needed to be configured and wired-in
  - Overall – an administrative nightmare. And don't even mention application service lifecycle and version management!
- Many options were considered, including various ESBs and commercial solutions.
- OSGi was a partial solution – fixed the discovery, class loading, configuration admin and versioning problem, but needed more!!!

## So – back to the problem...

- The move to OSGi for Servicemix 4 was perfectly timed, providing OSGi benefits with pre-packaged bundle configurations, admin tools on top of Karaf! Cool!

### A Perfect Match for ServiceMix!

- ServiceMix allows us to deploy, manage and update features
  - We use OSGi Declarative Services, Blueprint, etc. to deploy and manage user interface applications based on Eclipse Rich Ajax Platform (RAP)
  - We also deploy data provider service bundles that abstract/decouple database implementations from the app bundles - can use SQL, JPA or whatever....
- For defence applications, we often need more deterministic behaviour for information exchange. Apache Camel architecture allowed us to implement endpoints/components to allow us to use the OMG DDS and QoS profiles!



## So What?

---

- Many projects have a requirement to integrate, mediate, route and disseminate information
- There are many different sources of data and they are continuously changing – need to be agile and able to adapt quickly
  - Sensors
  - Mobile devices
  - Applications (database, web)
- There are many different consumers of information
  - Many different presentations
  - Analysis tools (AGI STK)
  - Visualization (Web pages, Maps, Google Earth)
  - Databases
  - Other “Systems” (SoS)
- This commonality has driven us to develop a core set of services that support these activities and applications
- Hence the Rapid Technology Integration Framework – RTIF!

## RTIF Technology Stack

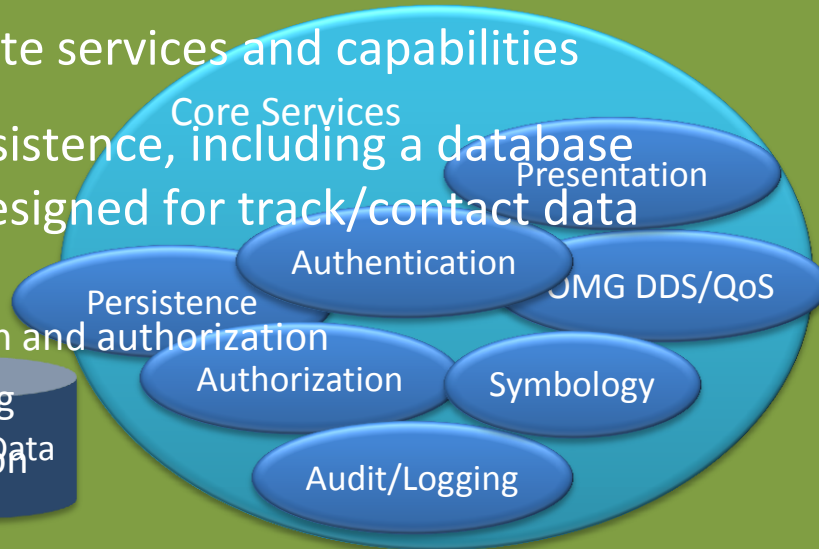
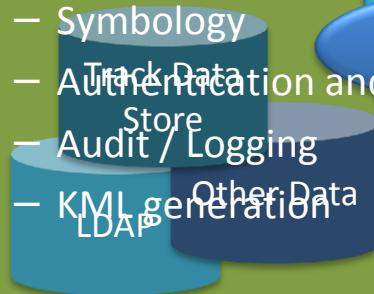
---

- Overall Requirement
  - A requirement existed for a framework to support defence domain application integration in an agile, rapid manner
- Technology Service Stack
  - Chosen from best of breed technologies available at the time
  - Highly decoupled component architecture that promotes the SOA core tenets
  - Lessons learned from SCA, COM, CORBA, J2EE
- Criteria:
  - Actively developed open source projects
  - Non-restrictive license (Apache)
  - Commercial support available if required
  - Pedigree in deployed applications
  - Ability to integrate/work together

# RTIF

- Build aggregate services and capabilities

- Add data persistence, including a database specifically designed for track/contact data



RTIF

ServiceMix

Messaging

- ActiveMQ
- JMS

Camel

- Routing
- Mediation

Services/CXF

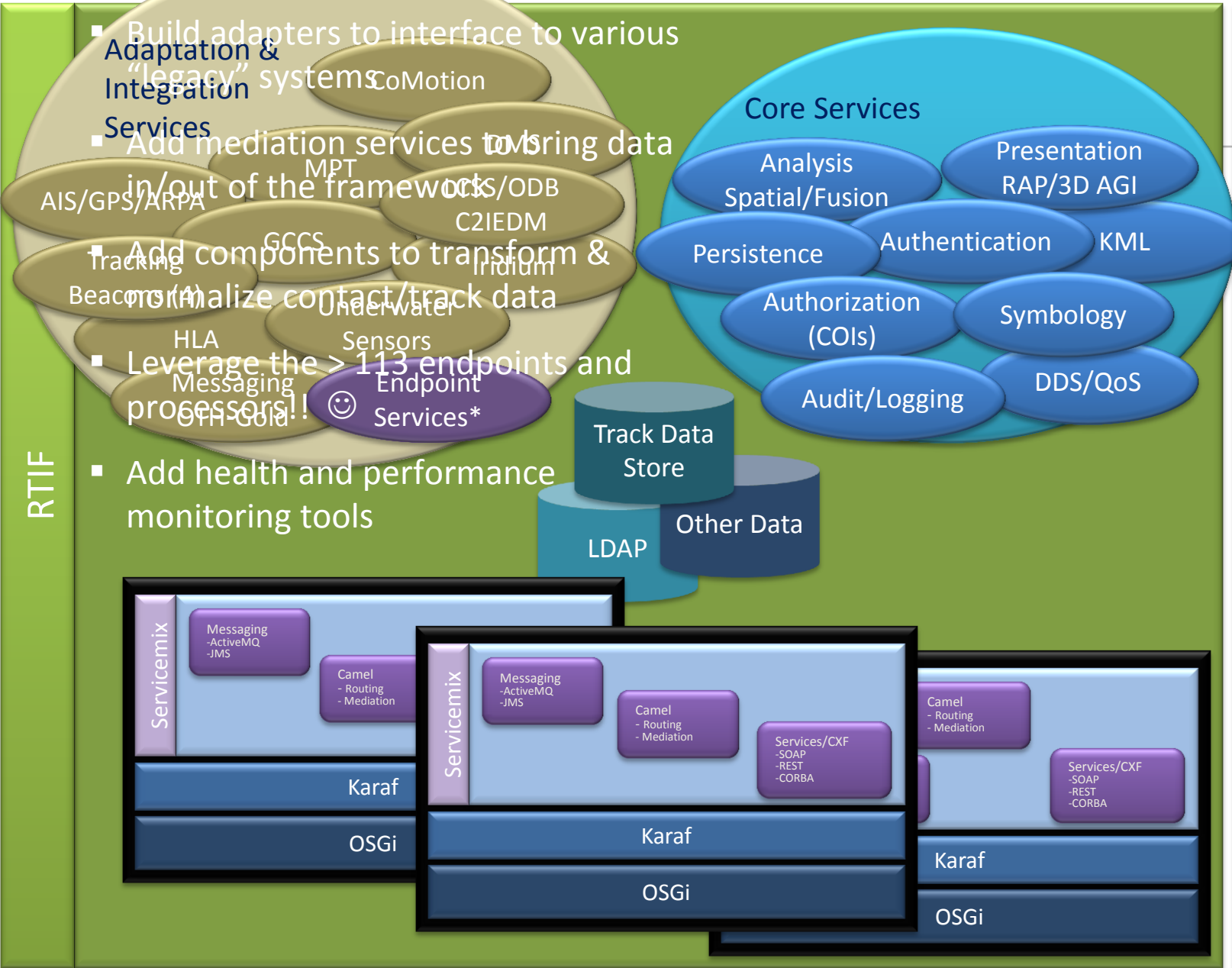
- SOAP
- REST
- CORBA

Karaf

OSGi

Security

Health Monitoring



Security

Health Monitoring

RTIF

- Build adapters to interface to various "legacy" systems
- Add mediation services to bring data in/out of the framework
- Add components to transform & normalize contact/track data
- Leverage the > 113 endpoints and processors!
- Add health and performance monitoring tools

## So what?

---

- Reduce defects
- Increase SLOC
- Agile development
- Rapid prototyping and delivery of capability
- Testability / Metrics
- Monitoring
- Versioning throughout the component lifecycle
  - SVN, Maven, OSGi
  - Compile, test, package, install, deploy

## RTIF from a Software Engineer's perspective...

---

- Don't need to worry about building infrastructure. Can focus on business/customer specific problems.
- Reduced risk through automated unit, integration and system level testing
- Easily collaborate with other software engineers via code from shared artifact repositories (geo-agnostic)
- Tools provide complete software lifecycle management
- Reduced SLOCs using Software Integration Patterns, thus reducing SPRs, Cost and Time to Deliver.
- Open Source!
- Leading edge!



## Projects Using RTIF

---

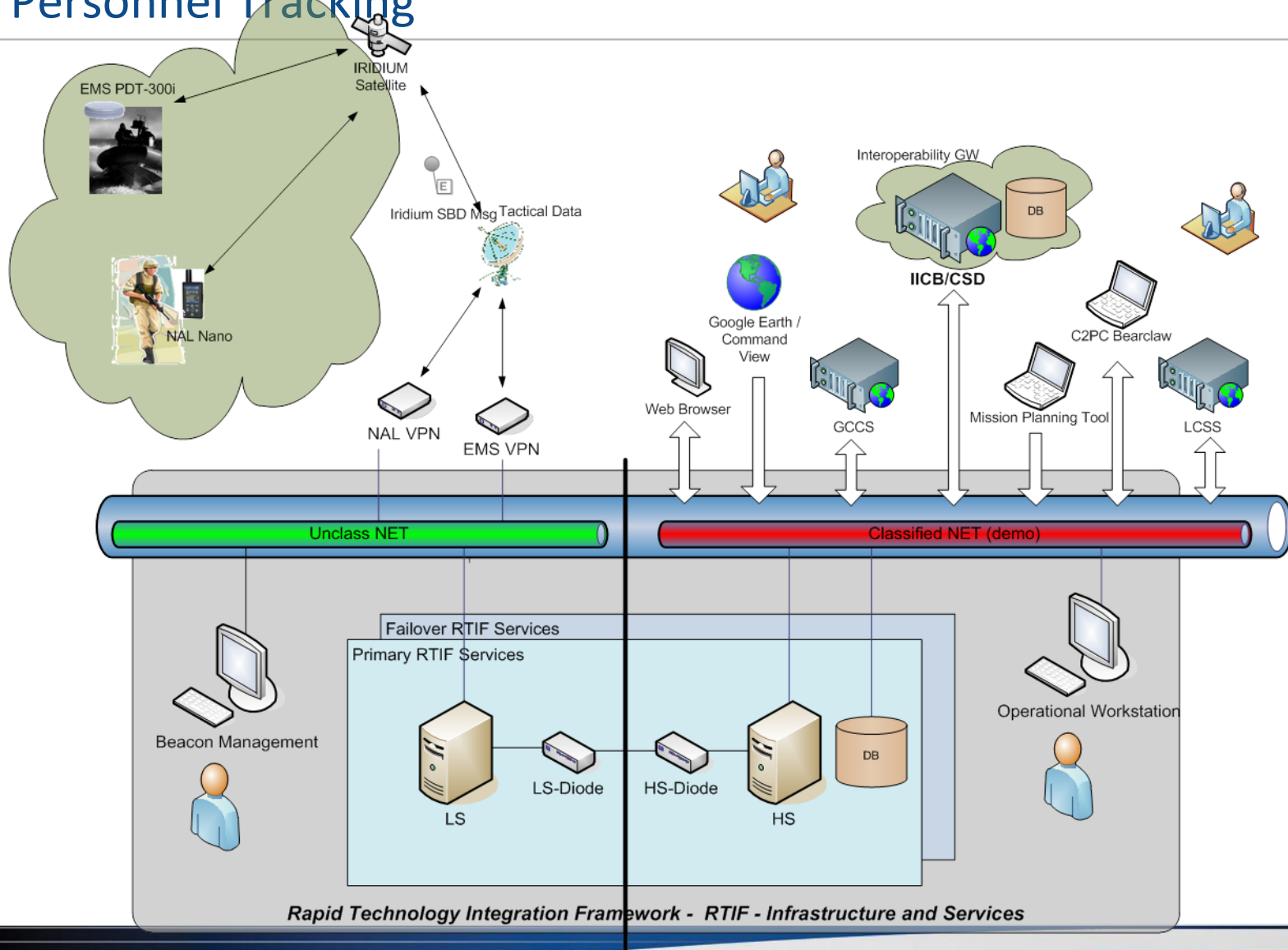
- Personnel tracking (BFT)
- Arctic Surveillance Simulation Testbed (NSA)
- Underwater Sensor fusion and display (ORCA)
- Torpedo Defence (MATADOR)
- Mission Planning Tool for Maritime Operations (MPT)
- Advanced SA and Planning tools (CoCommand)
- Data collection, Intelligence Collection for GDUK/Wales (CCDA)
- And wait! There's more...

## Personnel Tracking

---

- Provides Personnel Tracking using satellite-based tracking beacons
- Built initial prototype/PoC for \$25K using GD Canada RTIF technology.
- Prototype has been operational since November 2010.
- Next task was to “productize”. This is in current development and initial releases have been delivered to the customer.
  - Integrates and manages the provisioning and management of 5 different devices
  - Controls access to devices, keeps position history and text messaging
  - Manages alerts and warnings (911 or emergency calls)
  - Provides a set of user interface components that materialize depending on the user’s role (RAP, dynamic service discovery, etc.)
  - Generates routes, tracks and other information for consumption by Google Earth and keeps history in a geospatially enabled datasource
  - Provides the same data to other legacy systems by routing and mediating information through Camel (Velocity, Bindy, Smooks)
  - Supports authorization and authentication using JAAS
  - Outward facing endpoints secured with TLS

# Personnel Tracking



# Personnel Tracking

**BFT Gateway - Low Side Administration**

File View

System Logs Manage Devices Bandwidth

Service Providers Refresh

Monthly Bandwidth

December, 2011 to December, 2012 Apply

JUL - 2011	AUG - 2011	SEP - 2011	OCT - 2011	NOV - 2011	DEC - 2011
0.00 KB	0.00 KB	0.00 KB	0.00 KB	0.00 KB	2.00 KB

Service Providers

- No Provider
  - No Contract
    - 300034013421660 / Francois-Shout
    - 300234010965580 / Ottawa
    - 300234010272140 / Whisper
    - 300234010553080 / Nano
    - 300034013407000 / Shout

**BFT Gateway - High Side Administration - 1.4.0-SNAPSHOT on 192.168.98.115**

File View

System Logs Manage Devices Manage Communities Manage Users

Communities New Community

- 4444
- cole harbour
- operation 1
- Test
- test2
- test999

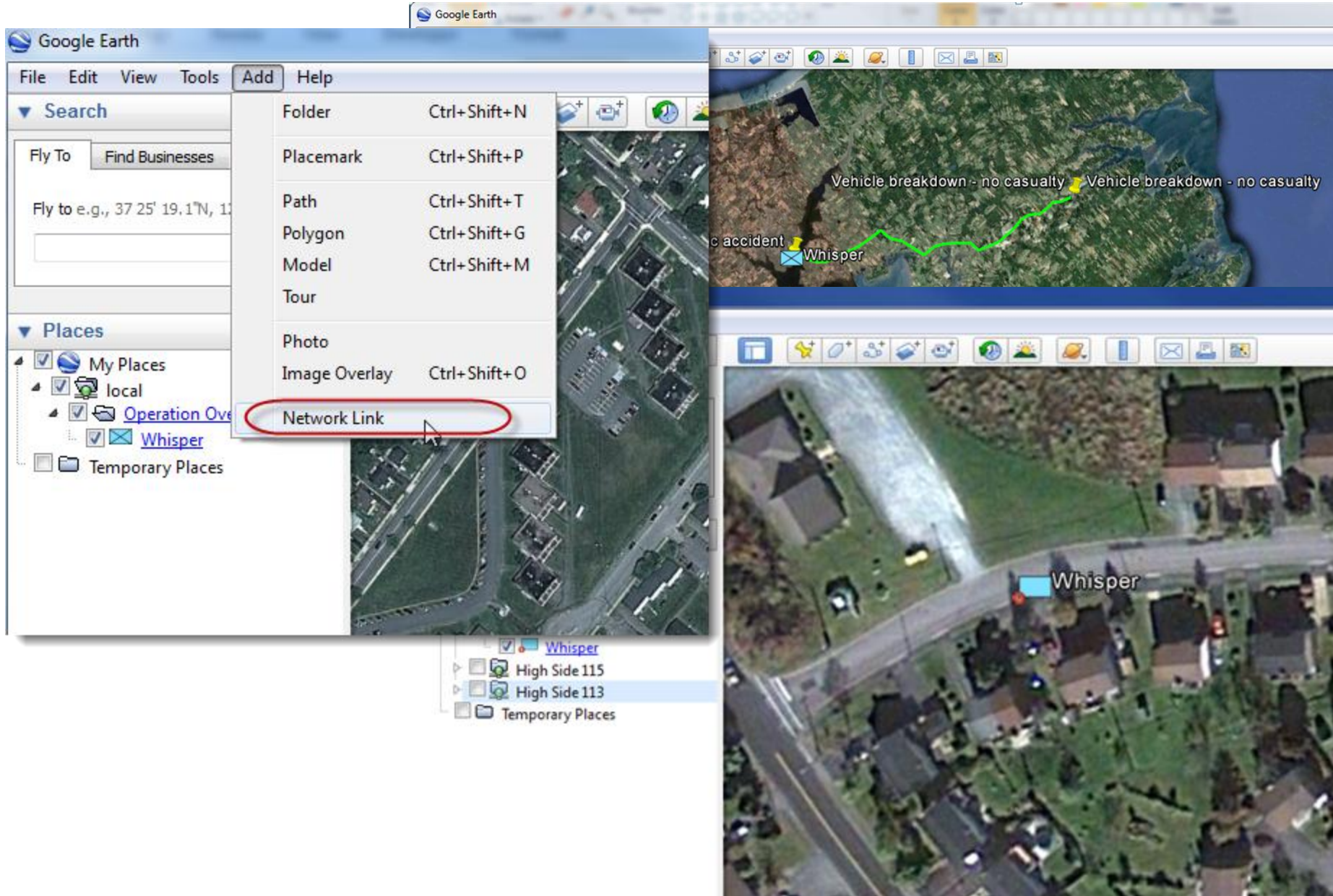
Device Properties

Device Type: Shout / M  
IMEI: 3002340  
Service Provider: No Provid  
Airtime Contract: No Contr  
Last Update: Never

Attributes Keys Rep

Apply

Name: \_\_\_\_\_





## Personnel Tracking

---

- What we are doing different?
  - Customer-centric. Developers talk to their operators daily.
  - Approximately 2-3 week release cycles which are delivered to customer (in some cases via email) using Agile/SCRUM development
  - Upgrading and management of the system is done by their IT support. We offer support.
  - GD Canada has a VPN which allows us to see “some” of their live test beacons. Helps us test/debug.
  
- How have we impacted the customer?
  - Very low cost, don't need to spend money on things that OSS provides
  - They get their features faster and in the hands of the operators
  - Operators have direct access to software engineers as support (phone/email)

# Arctic Surveillance Testbed – Northern Situational Awareness

---

- What is it?

- High fidelity modelling and simulation tool for Canadian Arctic
- Uses Analytical Graphics Inc. (AGI) Satellite Tool Kit (STK)
- GD Canada built custom UI plug-in to automate the users workflow.
- Data feeds for ingest and export all built on RTIF



GDC\_Models - STK 9 - [3D Graphics 1 - Earth]

File Edit View Insert Analysis Satellite Utilities Window Help



8 Mar 2011 19:00:17.134

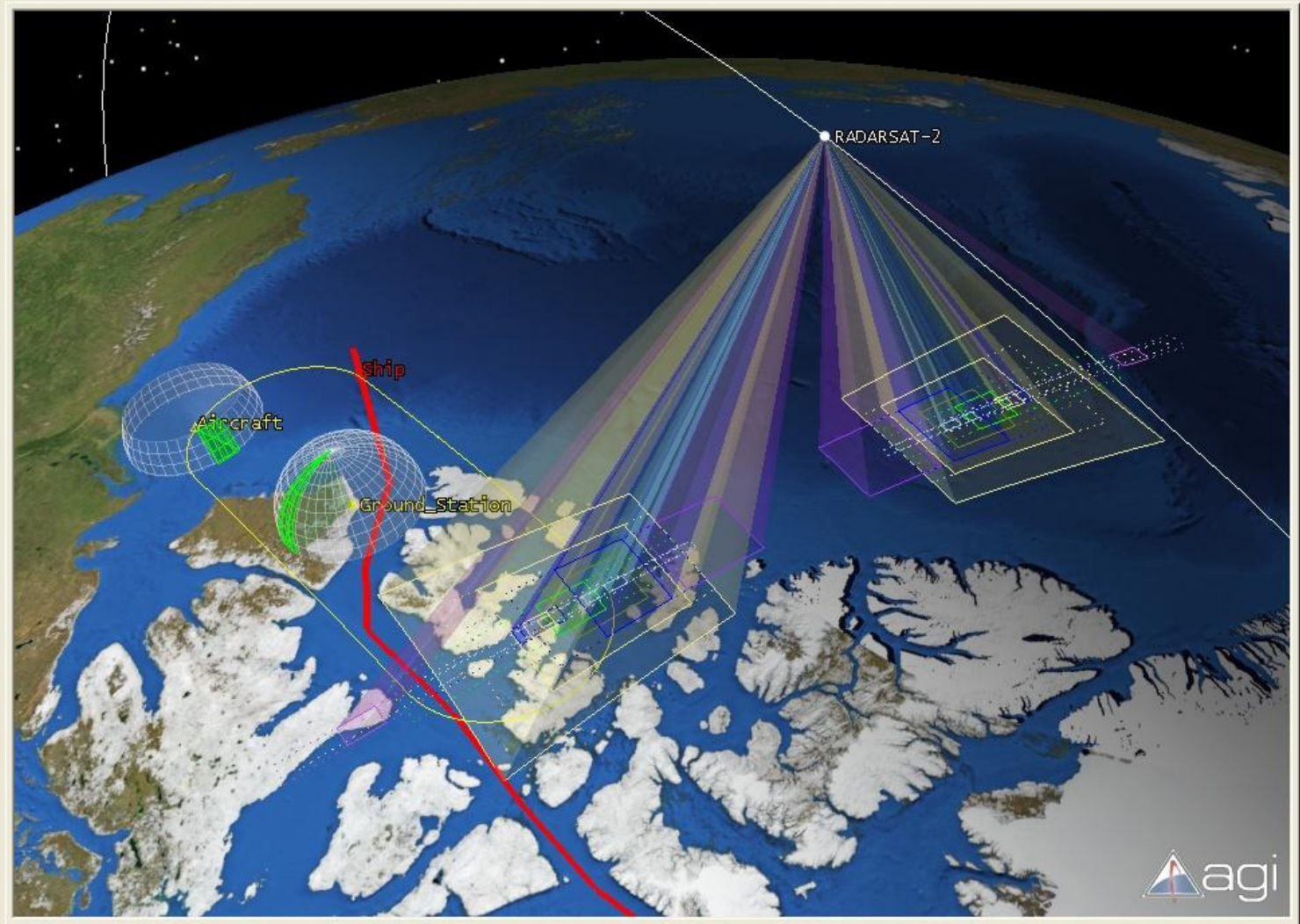
Object Browser

- GDC\_Models
  - Aircraft
    - AIS\_Air
    - AIS\_Air\_Rx
    - AirRadar\_FOR
    - AirScan
    - Radar\_Air
  - Ground\_Station
    - AIS\_Ground
    - AIS\_Ground\_Rx
    - GroundRadar\_FOR
    - GroundScan
    - Radar\_Ground
  - INMARSAT\_3F4
    - GIMROC\_LRIT
    - LRIT\_Space\_Rx
  - DRBCOMM\_FM04
    - AIS\_Space
    - AIS\_Space\_Rx
  - RADARSAT-2**
    - SAR\_GMTI
    - RS2\_ExtendedM...

Globe Manager: 3D Graphics 1 ...

Hierarchy Render Order KML

- Earth
  - BlueMarbleNG\_Bath
  - Clouds.cld
- Sun
  - Sun.jp2



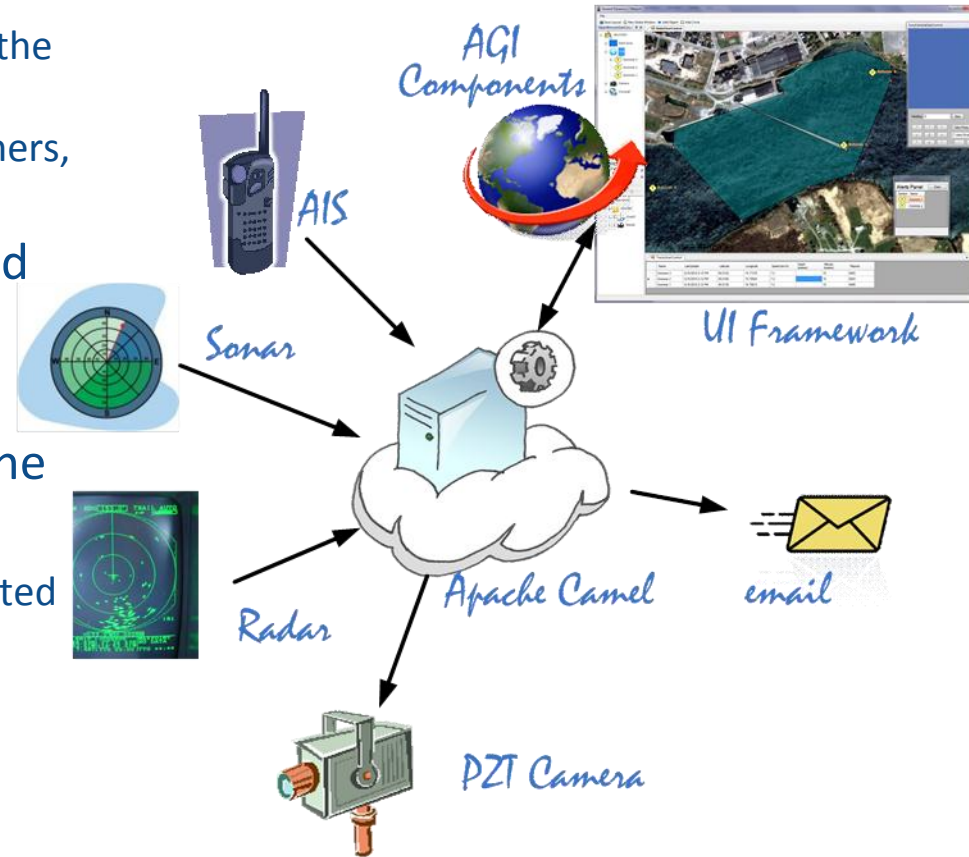
3D Graphics ...

RADARSAT-2 | 8 Mar 2011 19:00:17.134 | Time Step: 0.50 sec



# Underwater Sensor Fusion and Display

- Swimmer Detection
  - tasked to develop a low cost, state of the art prototype app for detecting and tracking potential threats from swimmers, divers and small boats.
- A prototype was built, field tested and demonstrated in less than 6 weeks!
- Used Apache Camel to support the integration
  - This work was “Pre-RTIF” but contributed to the RTIF service collection!



General Dynamics / Marport

File

Save Layout New Globe Window Add Object Add Circle

ObjectBrowserUserCon... X

GlobeUserControl

SAUCONY

- Alert Zone
- COP
  - Swimmer 3
  - Swimmer 2
  - Swimmer 1
- Camera
- Cornwall

GlobeManagerForm

Extents Street View

5 CHART

Transparency

Terrain Detail

Insight3D

- RASTER
- CHART
- IMAGE
- Cornwall

TracksUserControl

Name	Last Update	Latitude	Longitude	Speed (km/h)	Depth (meters)	Altitude (meters)	Reports
Swimmer 3	12/9/2010 2:12 PM	45.0122	-74.71378	7.2		10	6893
Swimmer 2	12/9/2010 2:12 PM	45.0156	-74.70504	7.2		10	6893
Swimmer 1	12/9/2010 2:12 PM	45.0135	-74.70613	7.2		10	6895

SonyCameraUserControl

Heading: 0 View

Take Photo

C:\data\EmailExport

Alerts Panel

Clear

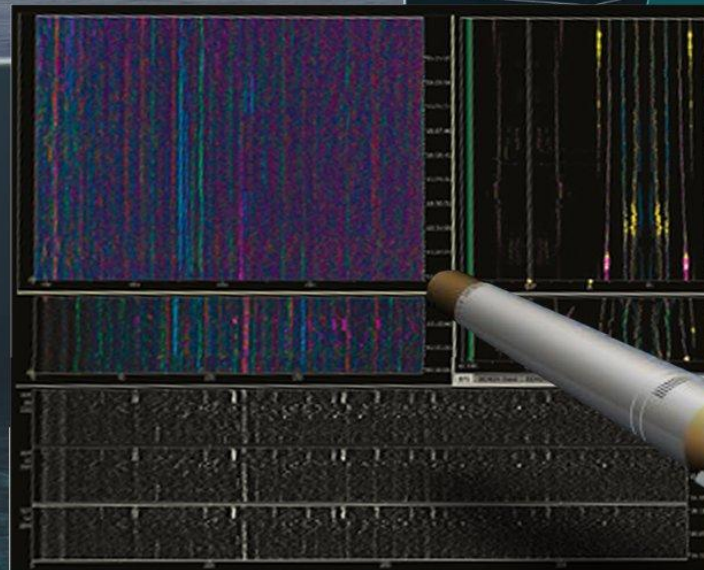
Symbol	Name
?	Swimmer 1
?	Swimmer 2

Latitude: 45.0156, Longitude: -74.6532

## Improved Towed Array Processing Matador Torpedo Detection System (TDS)

- ▀ MATADOR is a lightweight, low-cost torpedo detection system designed to be integrated with naval acoustic sensors and torpedo countermeasures systems.

GD Canada's Matador product has evolved from our collaboration with Defense Research and Development Canada (DRDC) Atlantic and many months of experience in realistic anti-torpedo exercises on the Canadian HALIFAX-class ships.



# Mission Planning

The screenshot displays a mission planning software interface with several key components:

- Execution Checklist:** A timeline view showing mission events such as 'Mission Start', 'VOI at H Hour', and 'VOI Execution Window Start/Stop' for a vessel named 'jamie\_cop'. The timeline spans from 26 2000Z Apr 10 to 28 0800Z Apr 10.
- VOI Properties:** A sidebar menu with options like 'Basic Properties', 'Route Editor', and 'Sensor Coverage'. The 'Route Editor' option is currently selected.
- AIS Viewer:** A map window showing a ship's track in the Mediterranean Sea. A pop-up window titled 'Imagery Satellites - MAERSK QATAR' displays a table of satellite passes.
- Satellite Data Table:**

Access Start	Access Stop	Satellite Name	Catalog Number
30 Jun 2011 10:49	30 Jun 2011 10:49	Noaa19	33591
30 Jun 2011 11:04	30 Jun 2011 11:04	Noaa19	33591
30 Jun 2011 11:41	30 Jun 2011 11:41	OrbView2	24883
30 Jun 2011 11:18	30 Jun 2011 11:18	Noaa15	25338
30 Jun 2011 11:33	30 Jun 2011 11:34	Noaa17	27453
30 Jun 2011 11:46	30 Jun 2011 11:46	Noaa17	27453
30 Jun 2011 10:46	30 Jun 2011 10:46	Noaa18	28654
30 Jun 2011 11:02	30 Jun 2011 11:02	Noaa18	28654
- Map Interface:** Shows a ship's track with a distance of 379.47 km. Search filters include MMSI 220418000 and a search box. A timeline at the bottom shows expanded data from Jun 29 to Aug 05.

# Advanced SA and Planning tools - CoCommand

The screenshot displays the CoCommand software interface, version 1.00, running on a user named SUPT2 connected to a server at 10.12.4.230. The interface is divided into several key sections:

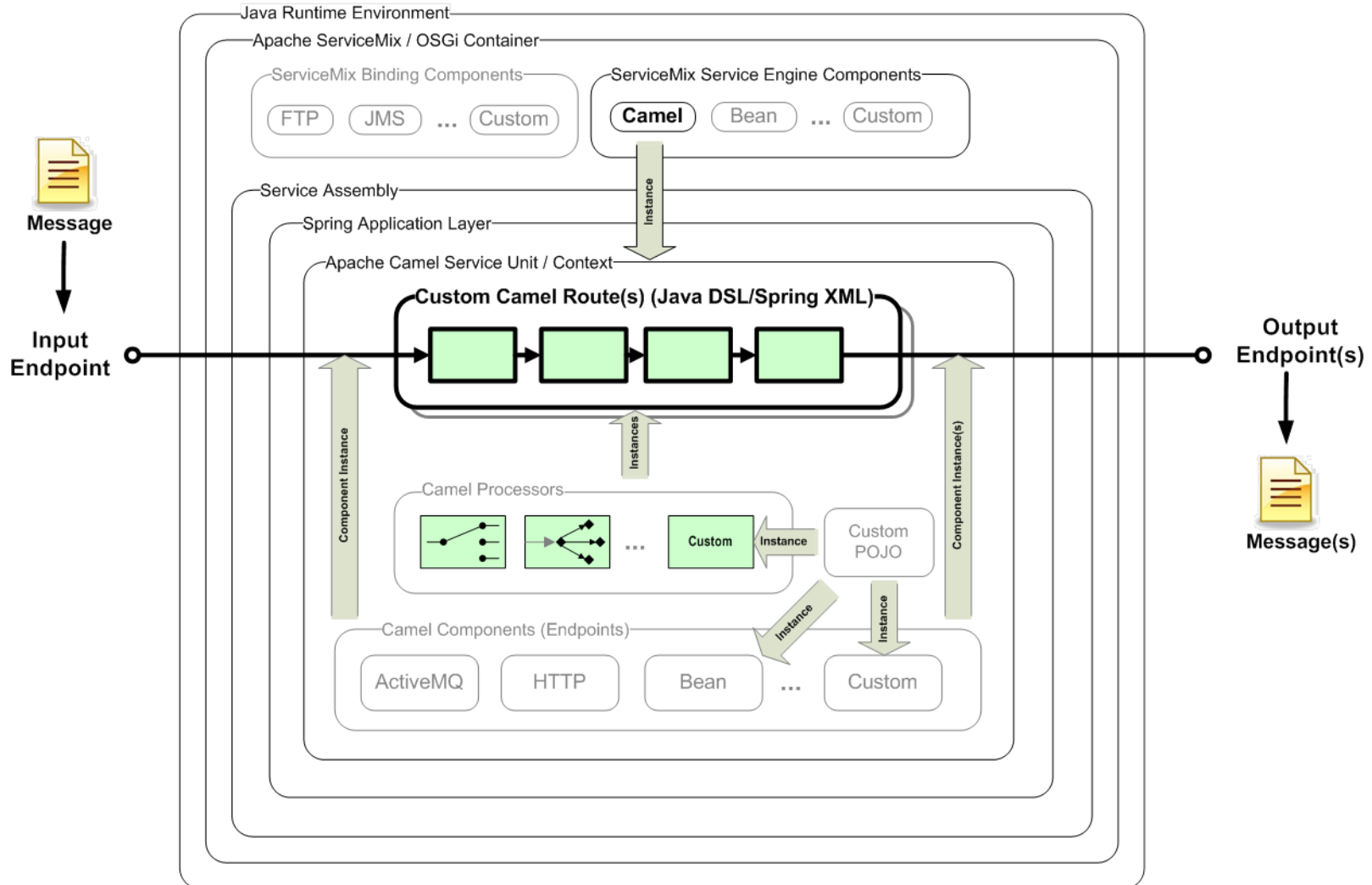
- Left Panel (Layer list):** A hierarchical tree view showing various data layers such as '5M Afghan', '2M Afghan', '250k Afghan', '50k Afghan', '10k Afghan', '5m CIB Afghan', '1m CIB Afghan', 'Map', 'Incidents by Type', 'Task Schedule', 'Data Palette', and 'Unit Palette'. It also includes 'General Tools' like 'Pasteboard', 'Table', 'Stickie', 'Tree Viewer', and 'Excel Importer'.
- Top Panel (Agenda):** A row of tabs for different views: 'Agenda', 'Troop Disposition', 'Current Ops', 'IntSum 22 Non 2011', 'SIGACT SUMMARY', 'Route Clearance Status', 'ISR Tasking Status', 'Visits', and 'Detainee Report'. The 'ISR Tasking Status' tab is currently active.
- Center Panel (Map):** A satellite-style map of Afghanistan with a grid overlay. Several ISR assets are plotted on the map, including 'ISR 2 - UAV', 'ISR 3 - UAV', 'ISR 4 - UAV', 'ISR 5 - UAV', 'ISR 6 - UAV', 'ISR 7 - UAV', 'ISR 8 - UAV', and a 'RECCE Patrol'. A '60' marker is visible on the map.
- Right Panel (ISR Tasking Tree):** A detailed tree view of the ISR tasking structure, including 'ISR Schedule Master', 'ISR Taskings' (with sub-items like 'USA', 'Europe', 'Middle East', 'JTRUAE AO', 'Asia', 'Africa', 'South America', 'Australia', 'Kandahar', 'Current View'), 'JTRUAE SOP', 'Afghan Boundaries', 'aims\_distric\_boundarie', 'Afghan National Border', 'Provincial Boundaries', 'Roads - Class 1', and 'ISR Tasks Scedule'.
- Bottom Panel (Assigned To Schedule):** A Gantt-style chart showing the assignment of tasks to resources over time. The x-axis represents time from 11/22/2011 00:28 to 11/23/2011 10:51. The y-axis lists categories like 'ISR 2 - Helo', 'ISR 3 - UAV', 'ISR 4 - UAV', 'ISR 5 - UAV', 'ISR 6 - UAV', 'ISR 7 - UAV', 'ISR 8 - UAV', and 'RECCE Patrol'. Resources listed include 'Cowboy - 4', 'Eagle - 2', 'Eagle - 4', 'HAWK UAV', 'SKY2', 'Eagle 5', and 'A/1/Plt 3'.

Annotations on the right side of the screenshot provide additional context:

- '...in the truck, they had 3 support Dispatch the QRT 150+ to the IED FOB2 (the closest one) to be'
- '...box should go'
- '...scous deployment.'
- '...will do when they get there):'
- '-show disposition of sub sub units (~3 platoons)'
- 'PUT BALLOONS AT KANDAHAR. AND AT THE 2 FOBs THAT HAVE UAE FORCES, but not the AFGAN FORCES'
- 'ADD 2 RECCE PATROLS WHERE WE HAVE THE AOs'
- 'WEATHER PICTURE FOR THE AREA SATELLITE'

At the bottom right, the text 'SUPT3 15Nov-2011 7:02:29 PM' and 'comotion Powered' are visible.

# CoCommand





## Use Cases

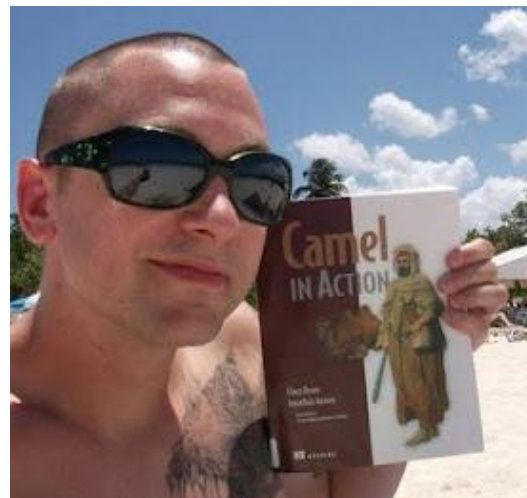
---

- There are huge benefits using the Apache open source integration and messaging products in a technology stack like RTIF
- I want to share some stories from the field where the architecture supported agility!



## Wire Tapping with Apache Camel

- We will start with a story about an engineer who was assigned by his manager to investigate how the company could replace one of their legacy databases that was written many decades ago (from scratch) with a new high performance off-the-shelf database. The new database would allow the company to meet the new demands from prospective customers and gain new business.



## Wire Tapping with Apache Camel

---

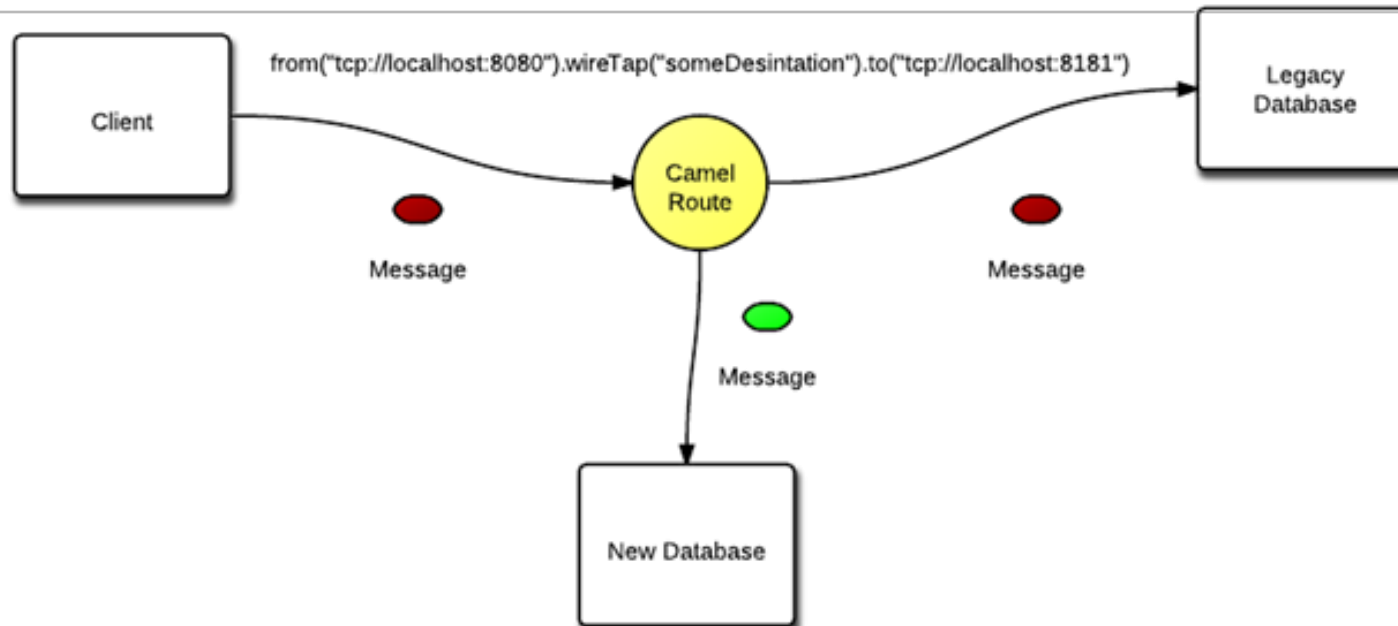
- The engineer spent a few weeks examining the cost and risk of replacing the database and realized it was no easy task.
  - The legacy database has over 60,000 lines of code and replacing it with a new database in a reasonable timeframe is risky and unproven
  - There are roughly 5500 different modules within the software that persist data to the database
  - Some of the companies customers have asked for the new database but also want to keep the existing legacy database running in case something goes wrong in operations.
  - The legacy database is implemented in an old programming language and the company wants to use a modern programming language.
- The engineer concluded that it is going to cost the company a few hundred thousand dollars to replace the database, and approximately a year to develop, integrate and test the new database with the existing modules in the legacy system

## Wire Tapping with Apache Camel

---

- The engineer pitches the idea to his boss that it might be possible for the company to slowly over time phase out the legacy database while developing new features for the new database.
  - If they could somehow run both side-by-side until the new database was mature enough and the engineers could switch the light on old database off with confidence.
  - The engineer realizes that all the modules send data using TCP sockets to the database and that if the data could be cloned and persisted to the new database then they may have a chance!
  - There is just one problem, the old code cannot be modified because it is working in other projects and they don't risk the possibility of breaking it.
  - The engineer gets out his trusted EIP cards and realizes that Apache Camel contains a Wire Tap EIP that he could use that would allow the data to be persisted to both databases without changing any existing code

## Wire Tapping with Apache Camel



```

new RouteBuilder() {
  @Override
  public void configure() throws Exception {
    // listen on http and wire tap to tcp
    from("http://localhost:8080")
    // copy message, asynchronously send to tcp
    .wireTap("tcp://someHost:5555")
    .to("http://localhost:8181");
  }
}
  
```

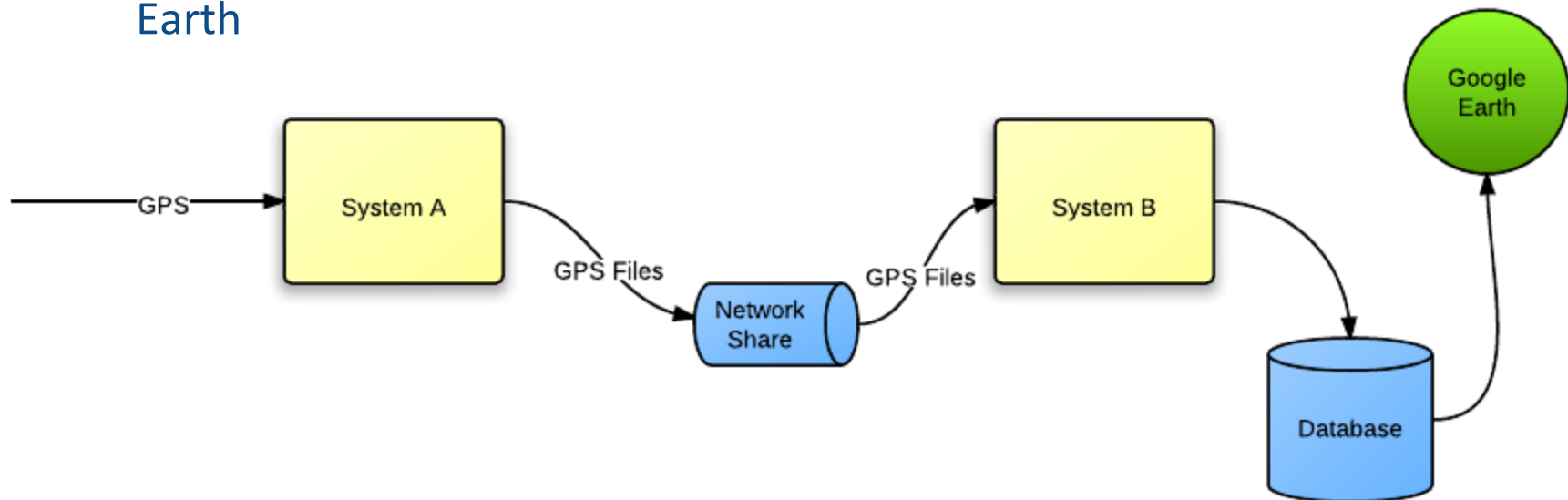
## Wire Tapping with Apache Camel

---

- The engineer demos to his boss Camel's wire tap EIP and they conclude that they will be able to run both databases side-by-side with very little risk!
- The engineer discovers that the new database uses an XML data model while the old legacy database uses proprietary binary messages that were written in the mid 1980s
  - They realize that they will need to find a convenient way to convert the messages from the legacy binary format to the new XML format.
  - As it turns out, Camel provides a way for engineers to seamlessly switch back and forth between various data models using Type Converters!

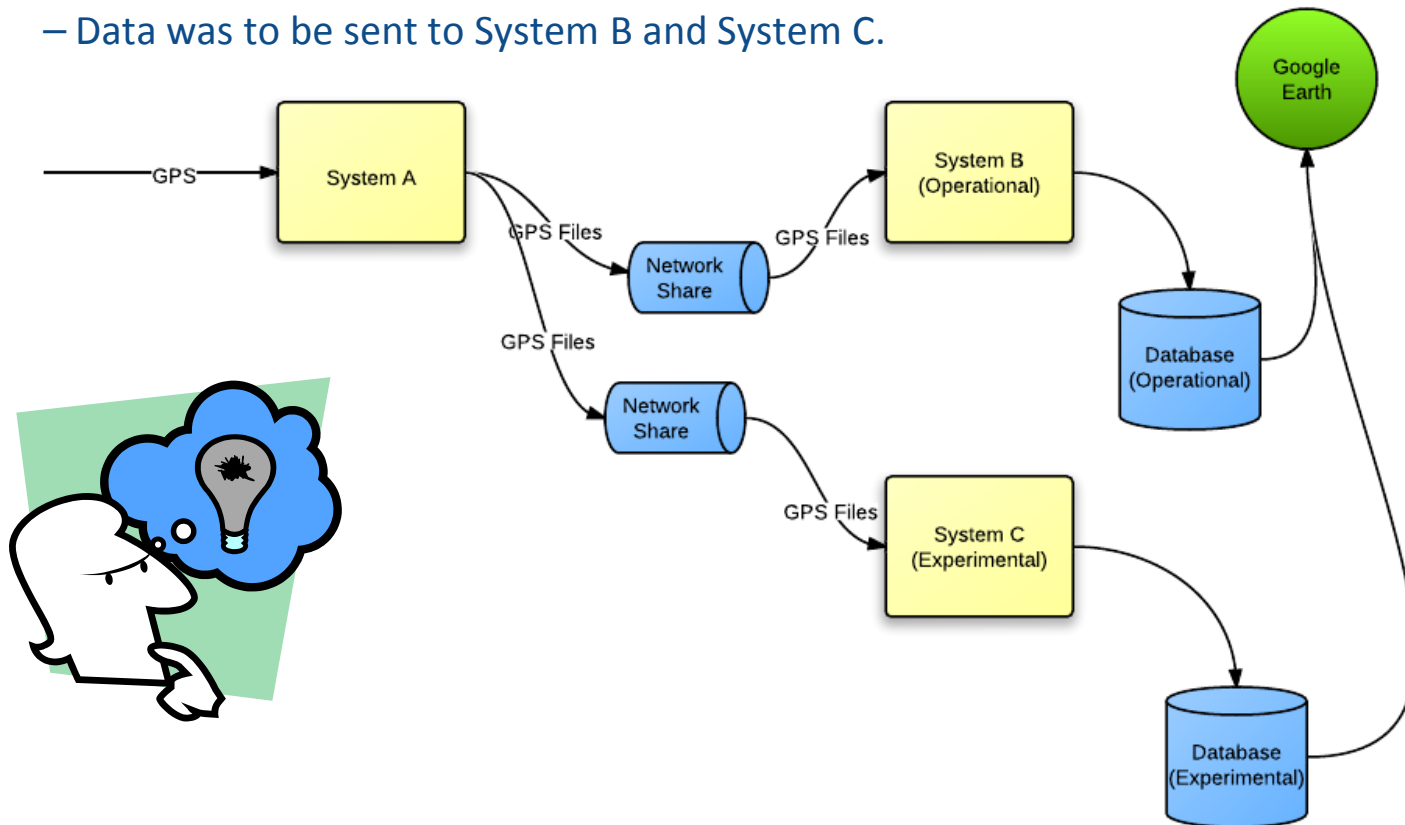
## The Importance of Agility in the Field

- Recently, one of my engineers was on site upgrading a custom software system we built for them for tracking GPS positions.
- The system receives data and stores the report data as files on a network drive. The files are consumed by System B which persists them in a database which subsequently can be displayed on Google Earth



## The Importance of Agility in the Field

- On his arrival the customer requested that a second system be deployed that they could use for testing and experimentation
  - The new test system could not interfere with the existing operational system
  - Data was to be sent to System B and System C.





## The Importance of Agility in the Field

---

- If this had been a legacy stove-pipe system, the customer would be required to submit an ECP which would then have to be approved. This would lead to additional requirements, software development, testing and of course time and money.
- From the customer's point of view the request was simple and they felt that duplicating data to both systems should be an easy task that should require little or no effort. Why does such a simple request have to be so costly? Well it doesn't!
- We were able to fulfill their request in under an hour. Oh and by the way, the toolbox did NOT contain a compiler!

## The Importance of Agility in the Field

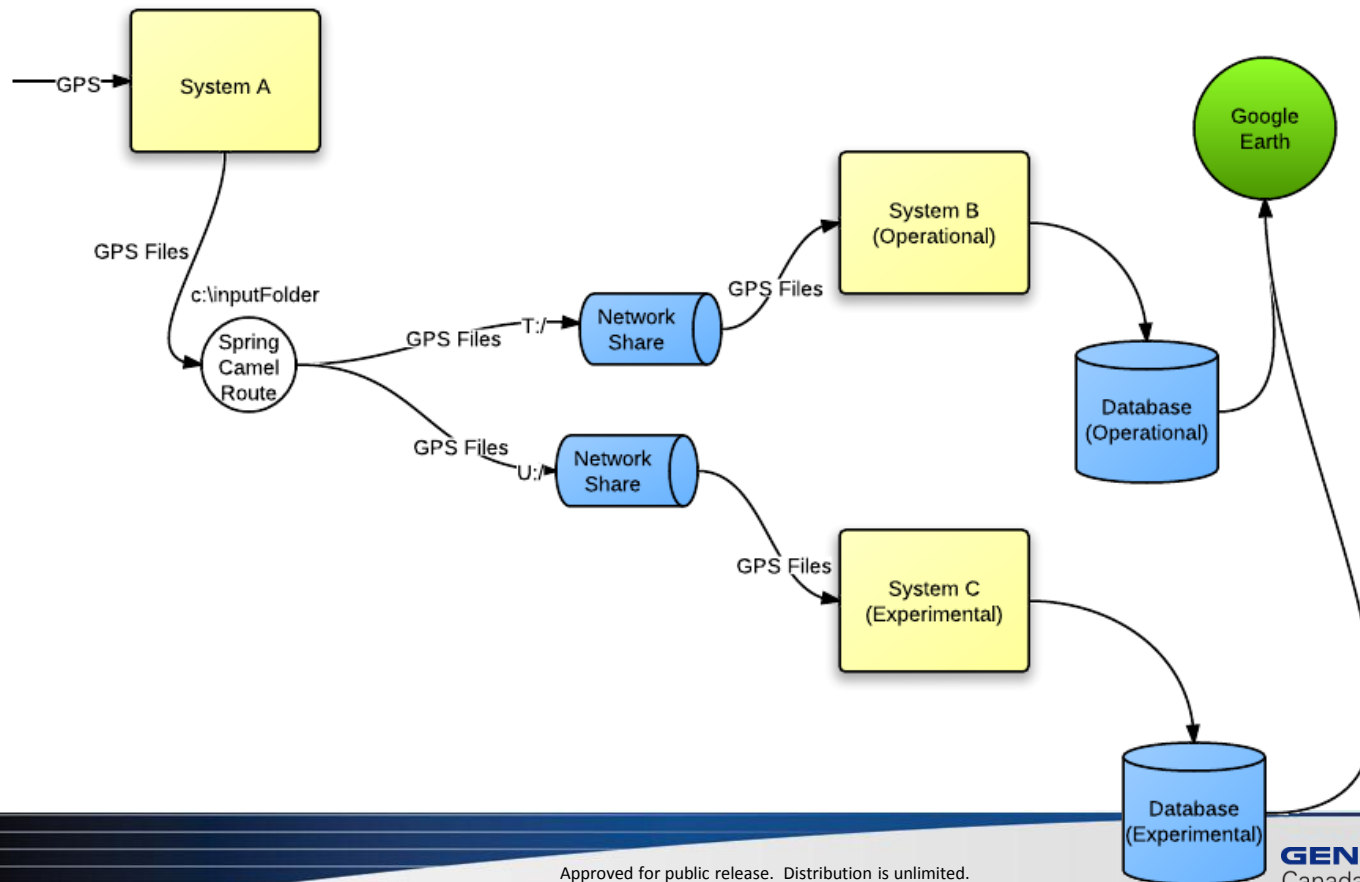
- We cloned System B and using a regular text editor created a camel route that duplicated the GPS Files.

```
<beans>
<camelContext xmlns="http://camel.apache.org/schema/spring">
  <route>
    <from uri="file://C:/inputFolder" />
    <multicast>
      <to uri="file://T/" />
      <to uri="file://U/" />
    </multicast>
  </route>
</camelContext>
</beans>
```

- How can XML possibly copy files from one location to another? The reason is that the XML file is actually a Spring XML file (denoted by the namespace).
- Fortunately, the application was built on Apache ServiceMix...

## The Importance of Agility in the Field

- Apache ServiceMix has an auto deploy feature that allows you to drop Spring XML files into a "deploy" folder and it will launch them as services.



## The Importance of Agility in the Field

---

- Although the Spring Camel Route was really simple, it shows how powerful combining Spring and Camel can be.
- This example illustrates how agility plays an important role in developing and designing systems
- The customer can do things with their software that they never thought possible.
- Our customer was really impressed that we were able to meet the on-demand task of setting up a second system and have their data duplicated to both, while not affecting operations!

## Issues

---

- Very few issues, but...
- When systems get large and complex, deployment of features and containers is problematic
- Fabric (Fuse ESB Enterprise) seems to be one solution!
- Still unsure if the new FAB will cover all of our use cases (already have two JIRA posts...)
- Bugs? Yes there are some – but Fusesource support has been great!

## Summary

---

- Benefits
- Apache software with FuseSource!!

# Questions?

# **GENERAL DYNAMICS**

## Canada

[www.gdcanada.com](http://www.gdcanada.com)

### Contact Information:

[michael.gingell@gdcanada.com](mailto:michael.gingell@gdcanada.com)

[james.archibald@gdcanada.com](mailto:james.archibald@gdcanada.com)