MIRA Team:

Eric Wolfer	Tom Diegelman	Robert Pitts
Rodney Grubbs	Steve Lucord	Kelvin Nichols
Laura Hood	Pete Gonzalez	Gary Grobe
Lindolfo Martinez	James R. Mireles	Walt Wilson
Tom Rich	Adam Schlesinger	

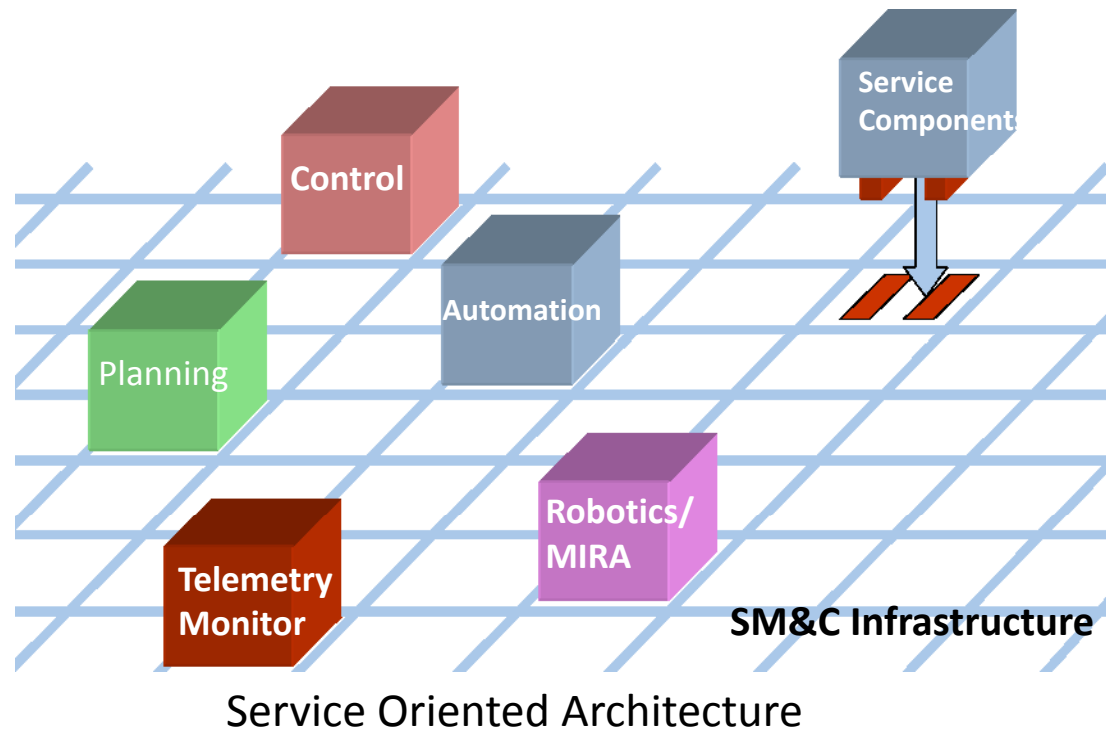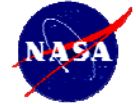# Motion Imagery and Robotics Application (MIRA)

MIRA Project Background

- NASA Digital TV organization will potentially deploy 3 cameras on the Exposed Facility (the "porch") of the ISS JEM Module
- At request of CCSDS MIA WG (R. Grubbs) and SM&C WG
    - JSC Standards started investigating the use of CCSDS Mission Operations Spacecraft Monitor & Control (SM&C) (November 2009)
    - Decision to start prototype design (May 2010)
    - Started prototype implementation (October 2010)
- SM&C Conjecture – The SM&C SOA can provide a standards based methodology to control and status Tele-robotic objects
    - SM&C is an application infrastructure that that can leverage lower layer communication services to provide a consistent set of interfaces to mission development and Ops teams
    - Cameras are a sub-class of Robotic Class of objects
    - Standardizing robotic operations leads to easier integration of various robot types and thus potentially lowers life cycle costs
    - SM&C Parameter and Action services will be used to control camera systems
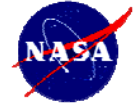
CCSDS Mission Operations Spacecraft Monitor and Control

Service Oriented Architecture

Control

Automation

Service Components

Planning

Robotics/MIRA

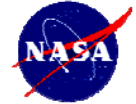Telemetry Monitor

SM&C Infrastructure

MIRA Project Background Continued

- The SM&C SOA approach has the potential for:
  - plug and play
  - software reuse
  - self identification
  - self configuration and discovery
  - cross-agency policy based robot capability sharing
  - lower development as well as lifecycle/maintenance costs

- Additional goals are to mature SM&C SOA standards specs for Robotic Control
  - Specs are in various stages of CCSDS Blue and Red Book
  - Second use of the SM&C standard in NASA (prior was 2010 DLR Prototype)
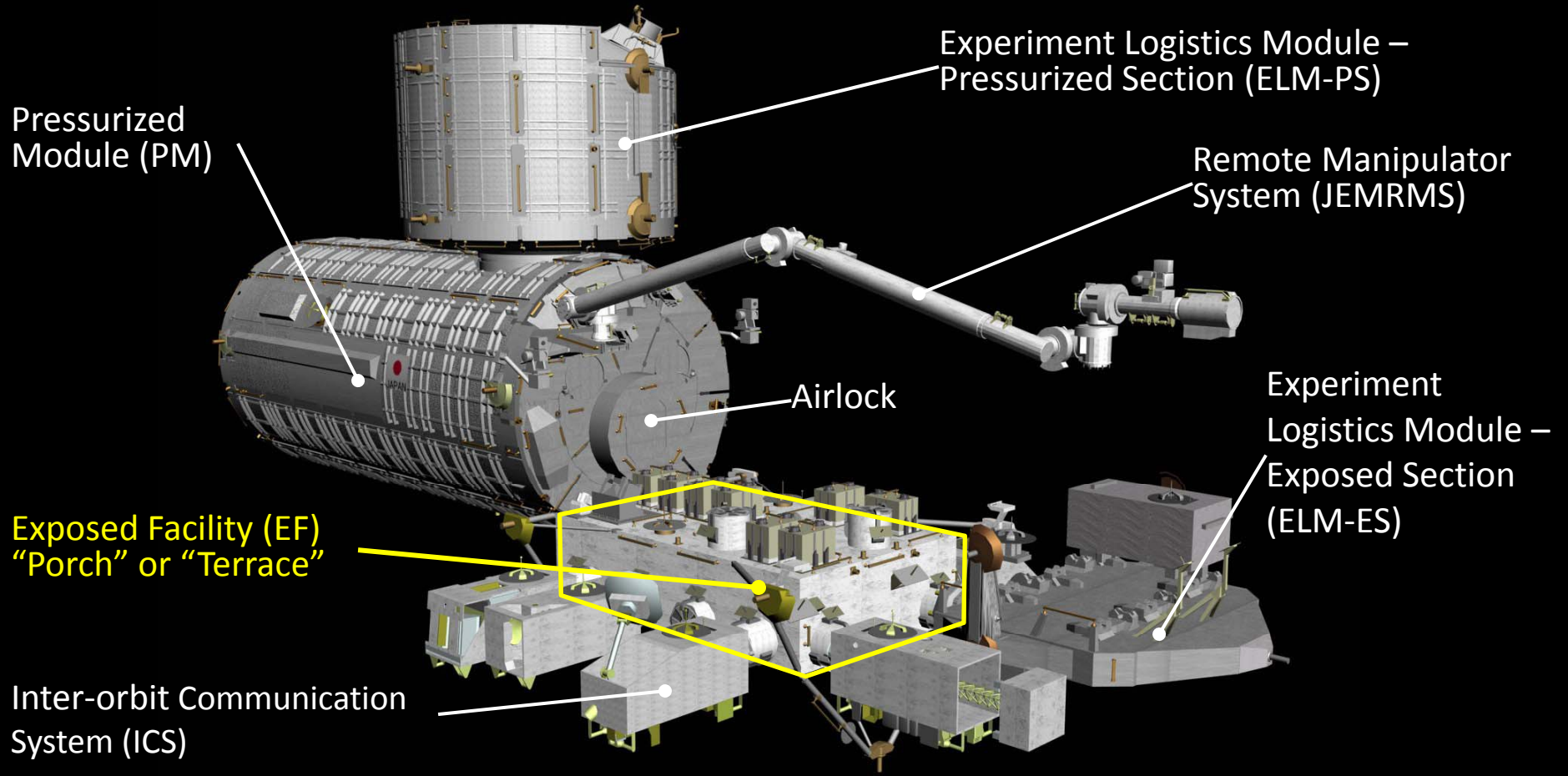 - Evolve Ops concepts for using SM&C and other CCSDS standards
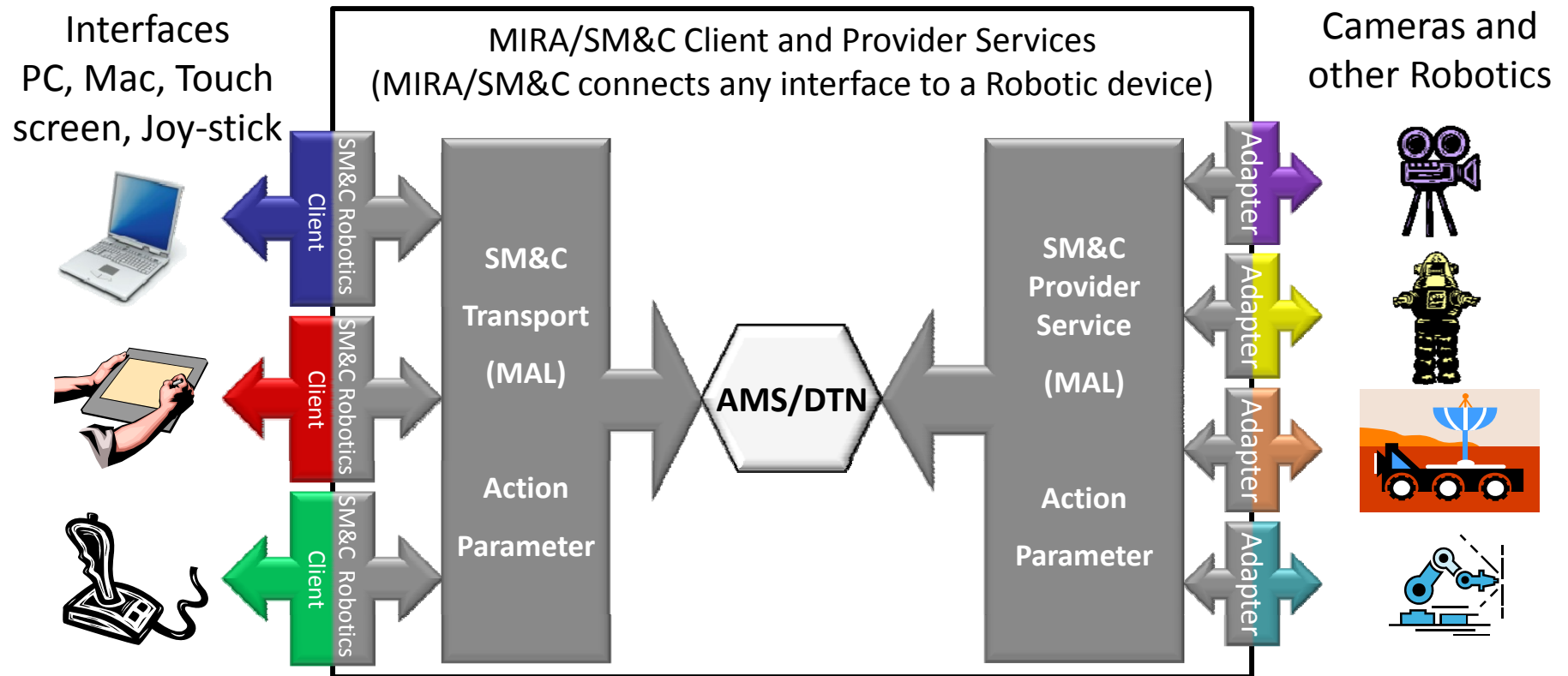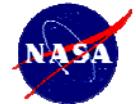
MIRA Project Objectives

- Prototype a camera service leveraging the CCSDS Integrated protocol stack (MIRA/SM&C/AMS/DTN)
  - CCSDS MIRA Service (New)
  - Spacecraft Monitor and Control (SM&C)
  - Asynchronous Messaging Service (AMS)
  - Delay/Disruption Tolerant Networking (DTN)
- Additional MIRA Objectives
  - Demo of Camera Control through ISS using CCSDS protocol stack (Berlin, May 2011)
  - Verify that the CCSDS standards stack can provide end-to-end space camera services across ground and space environments
  - Test interoperability of various CCSDS protocol standards
  - Identify overlaps in the design and implementations of the CCSDS protocol standards
  - Identify software incompatibilities in the CCSDS stack interfaces
  - Provide redlines to the SM&C, AMS, and DTN working groups
  - Enable the CCSDS MIRA service for potential use in ISS Kibo camera commanding
  - Assist in long-term evolution of this entire group of CCSDS standards to TRL 6 or greater

Japanese Experiments Module (JEM) "Kibo" Module
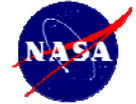
High Level Architecture

Interfaces
PC, Mac, Touch
screen, Joy-stick

MIRA/SM&C Client and Provider Services
(MIRA/SM&C connects any interface to a Robotic device)

Cameras and
other Robotics

SM&C Robotics Client

SM&C
Transport
(MAL)

Action
Parameter

AMS/DTN

SM&C
Provider
Service
(MAL)

Action
Parameter

Adapter

MIRA CCSDS Demo Services Stack

MIRA GUI		Panasonic Camera Object
MIRA App Service	➢ **Control and Status**	**MIRA Bridge/Adapter**
SM&C Consumer		**SM&C Provider**
AMS		**AMS**
DTN		**DTN**

Comm Link
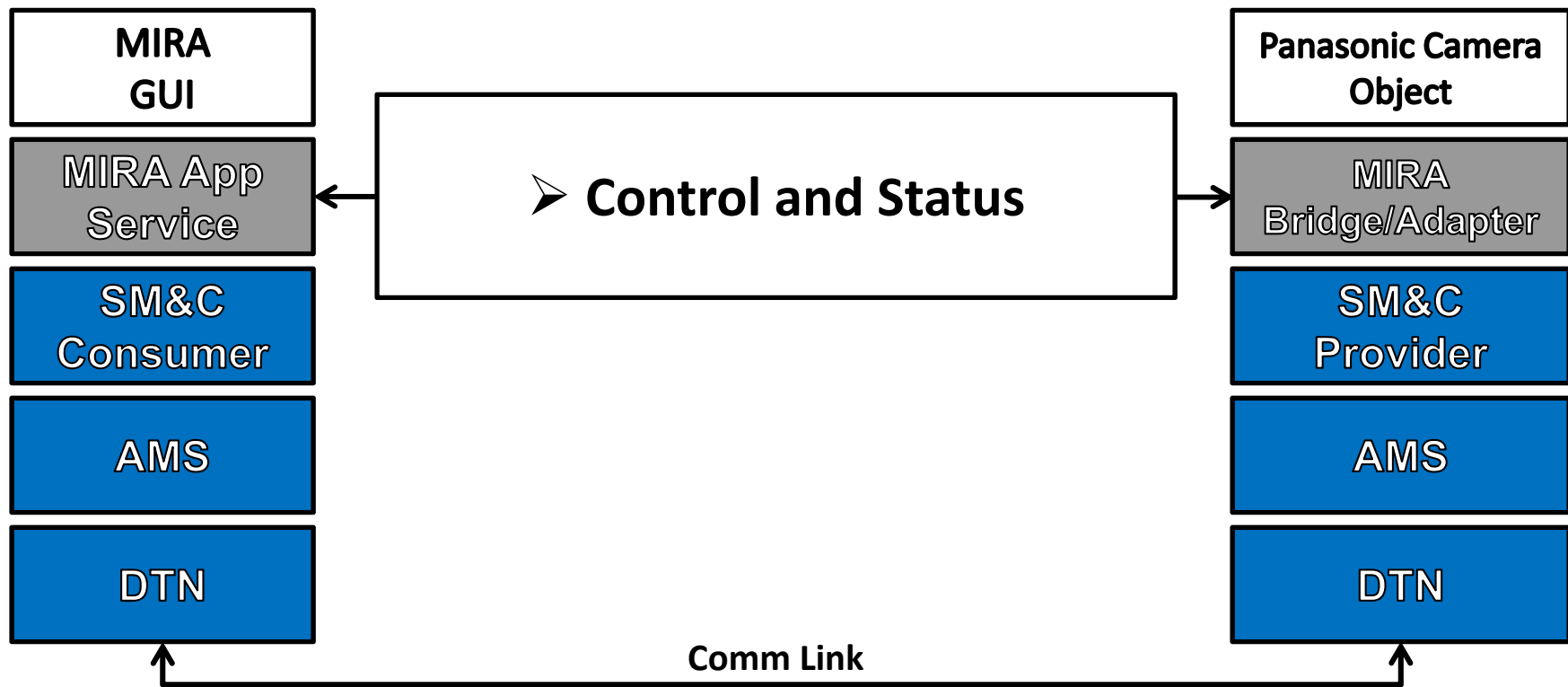
MIRA Demo Control/Status Configuration

IP Network (JSC)

Ground DTN (MSFC)

IP Network (JSC)

MIRA\SM&C\ AMS\DTN\VPN

MIRA\SM&C\ AMS\DTN\wi-fi

SM&C Consumer Application/Bridge

GUI

SM&C Provider Application/Bridge

RS232\ RS422\ CAT5

JSC/Berlin

JSC
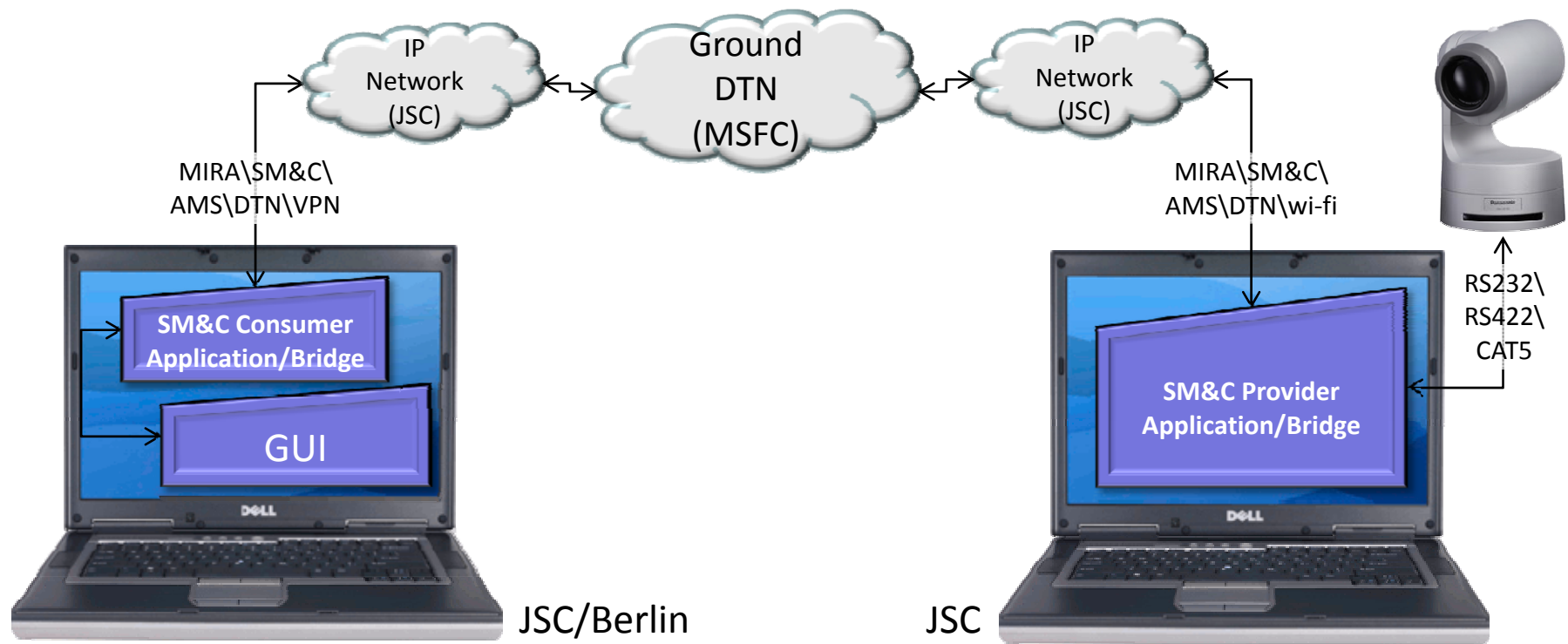
MIRA Demo with ISS Pass-Through

DTN bundles are not unpacked and repacked

ISS DTN Network (via MSFC)

IP Network (JSC)

IP Network (JSC)

Ground DTN (MSFC)

MIRA\SM&C\ AMS\DTN\VPN

MIRA\SM&C\ AMS\DTN\wi-fi

SM&C Consumer Application/Bridge

GUI

SM&C Provider Application/Bridge

RS232\ RS422\ CAT5

JSC/Berlin

JSC
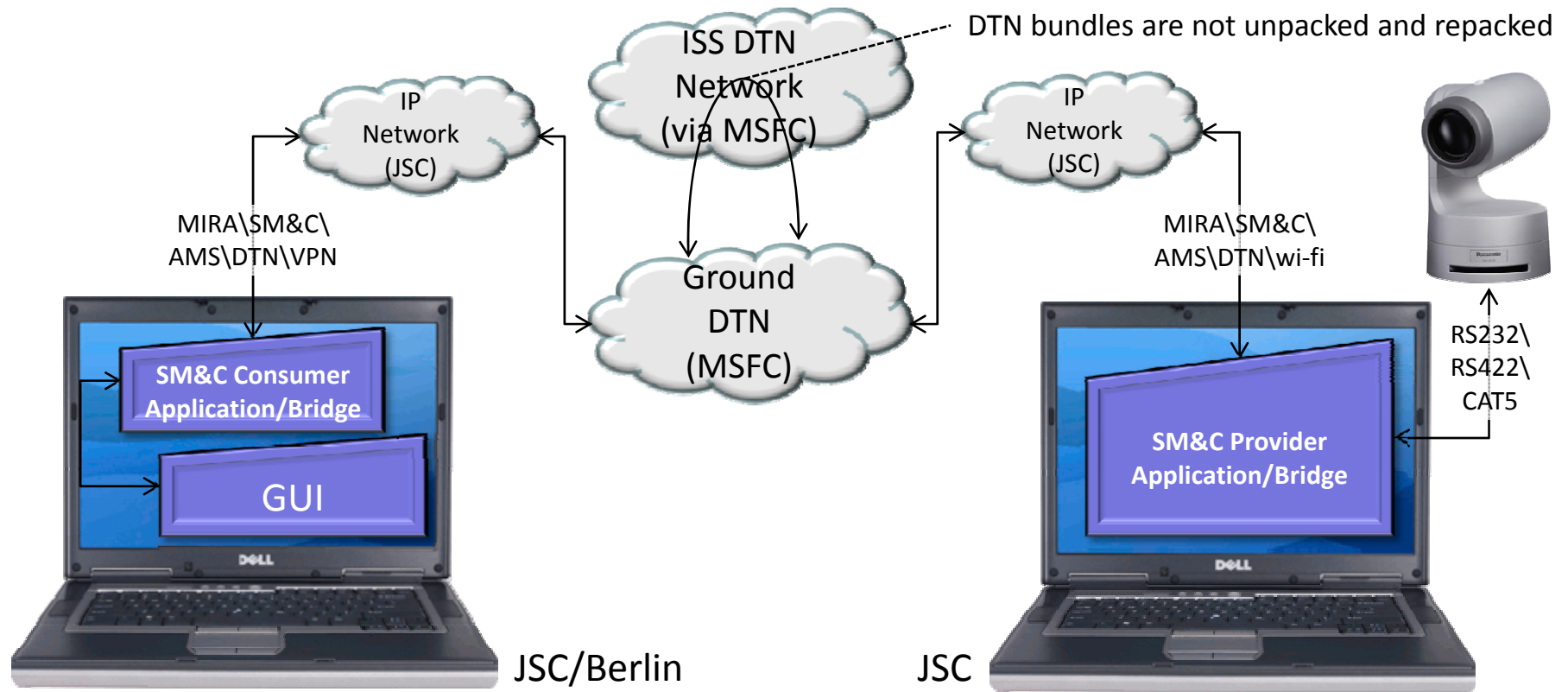
MIRA Video Demo
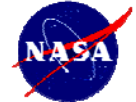
AMS\DTN and Channel Emulation Testing in the Lab

optibase

MGW Micro
Encoder
(Standard Def)

RAMS/AMS
Continuum
(hosted on SL5
virtual machines)

Multicast

H.264
H.264
H.264

H.264
H.264
H.264

Multicast

DTN Bundles

IP DTN Packets

H.264
H.264
H.264

Multicast

VLC Player
(VideoLAN)

Channel Emulator adds delay – both
inbound and outbound. Can also
drop packets and introduce bit
errors.

HP xw6600 with two NICs, one to the
LAN, the other to the SL5 DTN laptop.

VLC Player
(VideoLAN)
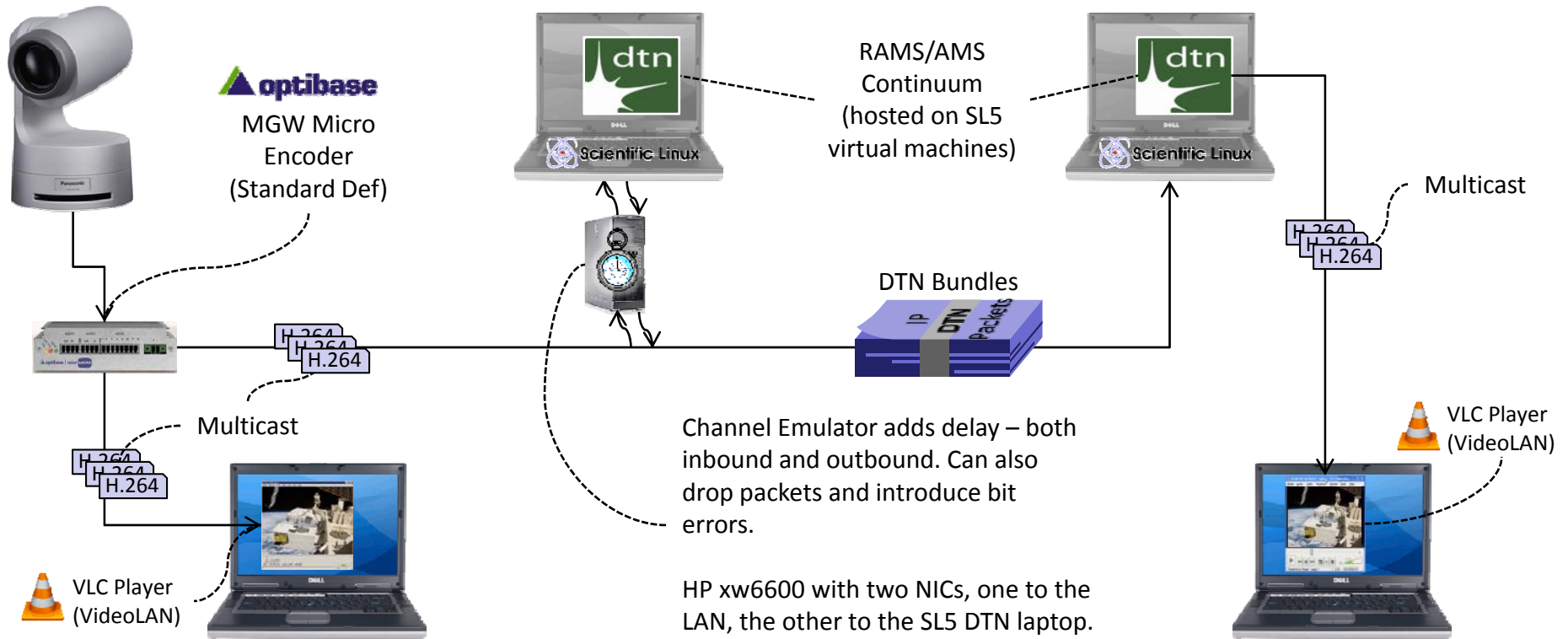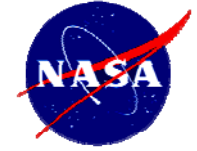
Demonstration

MIRA Innovation Day Architecture

JSC Wireless Peer-to-Peer

B14
(Camera on
Morpheus)

JSC Wireless
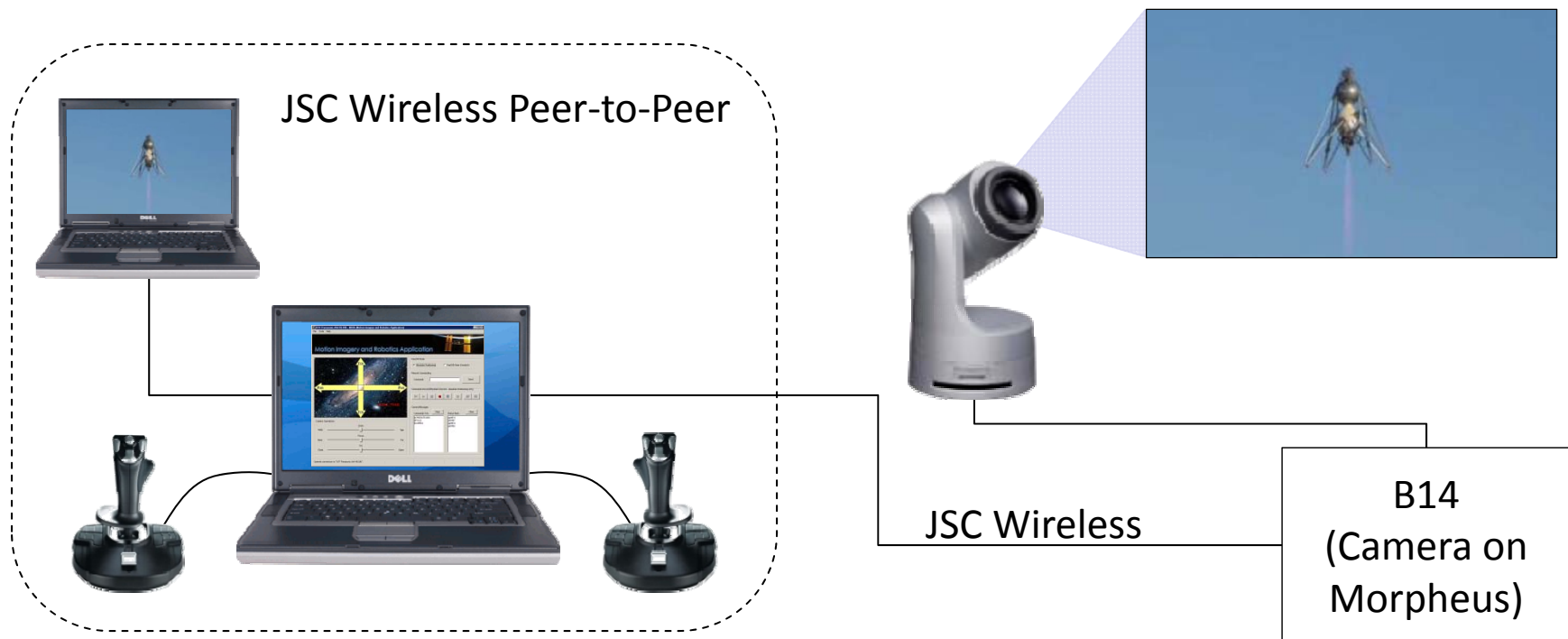
MIRA Prototype Performance
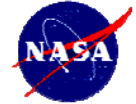
- Performance Components:

SM&C Encode	001 msec
AMS Publish/Receive	003 msec
SM&C Decode	002 msec
AMS/DTN (R/T JSC-to-MSFC)	200 – 500 msec
Camera Encoder	200 – 300 msec
Total	**406 – 806 msec**
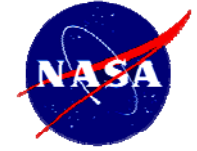
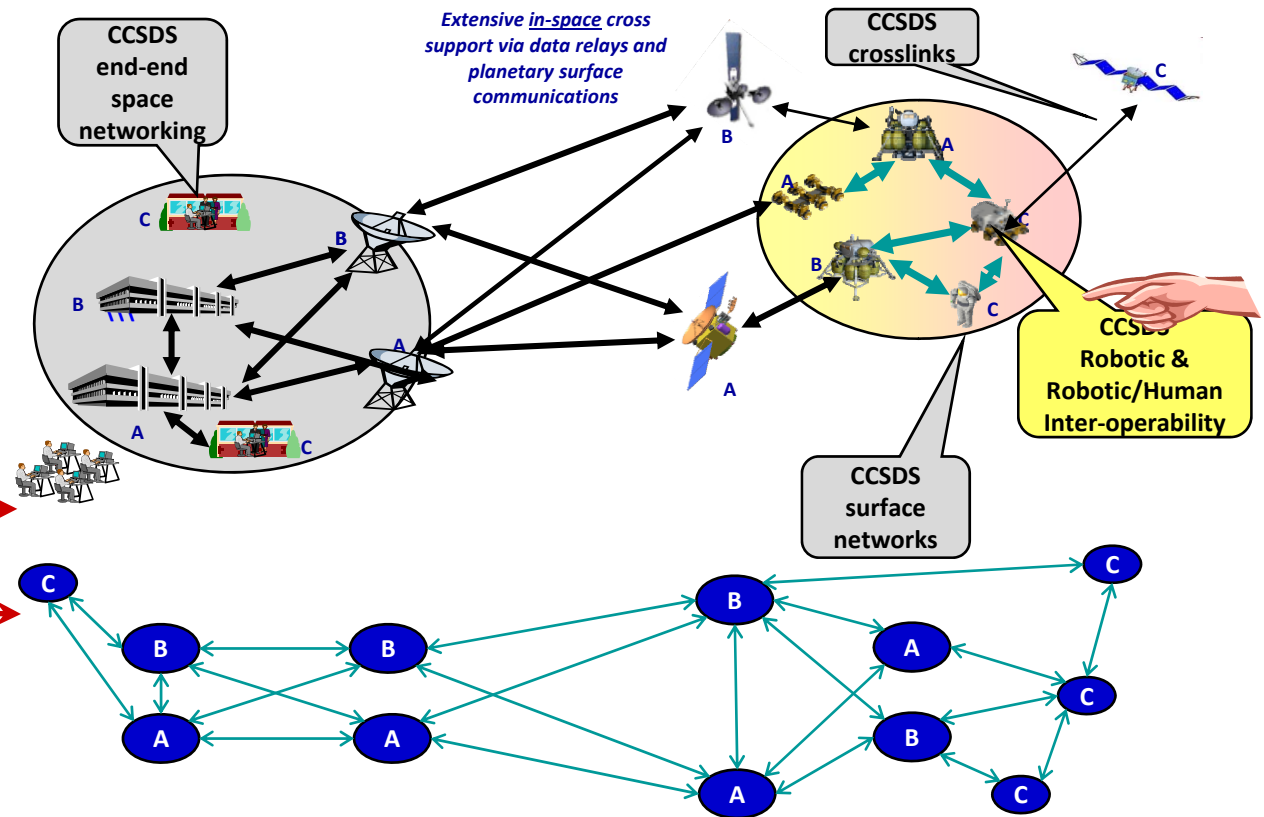# Forward Plan

Future Applications of Robotic Standards

Future scenarios indicate that international cross-support will grow to become:

- **Multilateral**
- **Both space and ground-based**
- **A mix of point-to-point and**
  multipoint-to-multipoint
- **More complex and dynamic**
- **More highly automated**

Emphasis on fully-standardized end-to-end networked data transfer

CCSDS
end-end
space
networking

Extensive in-space cross support via data relays and planetary surface communications

CCSDS
crosslinks

CCSDS
Robotic &
Robotic/Human
Inter-operability

CCSDS
surface
networks

Collaborative
Points

Long Term Goal (CCSDS Tele-Robotic Service)

Robotic GUI

Robotic App Service

SM&C Consumer

AMS

Other Middleware

DTN

- ➤ Multi-agent, distributed coordination for autonomous robots
- ➤ Hierarchy of Control (Policy-based robotic sharing)
- ➤ Social Behavior patterns, intelligence, modeling, perception, directories, inter-element behaviors and human-robotic interaction behavior
- ➤ Cross-Support policy & management and planning
- ➤ Safety policy and management
- ➤ Security policy and management
- ➤ Domain Specific Language (DSL) for control
- ➤ Standards for docking, cameras, autonomous and articulated robots, autonomous rovers and other types of robotic elements
- ➤ Communications stacks based on recognized international interoperability standards
- ➤ Defining semantics in Telerobotics
- ➤ Areas for standardization in the Telerobotics domain
- ➤ Architectures and experiences in area of Telerobotics
- ➤ Application of communications architectures in Telerobotics domain
- ➤ Benefits of standards for current and future projects or missions

Robotic Object

Robotic Bridge/Adapter

SM&C Provider

AMS

Other Middleware

DTN

Comm Link

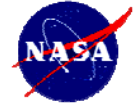# Lessons Learned

SM&C Lessons Learned

- MIRA Service duplicates the Parameter and Action interface

- Only stub and skeleton code is reused
  - MIRA service required developing new business logic
  - Code reuse is limited to what should be auto generated.

- Intra-Process Communication (IPC) works well going from consumer to provider. No assistance going the other direction.

- Encoding Layer
  - Encoding strategies may be developed to reduce overhead in bandwidth constrained environments
  - MAL End Point may also further reduce message size using lookup tables

- Message Abstraction Layer
 - Enabled the swap of encoding layer without modification to applications

SM&C Lessons Learned
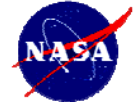
- ## Device Bridge
  - Individual Action and Parameter Service work well in the bridge to the device

- ## UIF Bridge
  - MAL implementation is analogous to JVM: Runs only on platforms with an available implementation
  - No Windows version yet
 - Possible porting options include Apple IOS, Google Android, MS Windows

DTN Lessons Learned
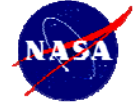
- ION could use additional logging, debugging, tuning, and general data visibility and system health reporting capabilities

- Arrange for DTN nodes in DMZs to have NTP access (clock drift)

- Expect to start RAMS 75 seconds after AMS

- To get RAMS to run you sometimes have to delete the petition.log file

- To get RAMS to run sometimes you have to kill and restart both ramsgate processes
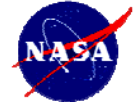
AMS Lessons Learned - Threads
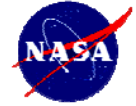
- The thread that registers the subscriber must also receive the messages

  - No other threads are allowed to receive messages for which the subscriber has registered.

  - AMS will post an error message if another thread tries to process the messages

- It appears that certain parts of AMS are not completely thread-safe. For example:

  - Experienced problems trying to register a subscriber immediately after registering a publisher

  - AMS will fail unless a 1 second delay is inserted between the call to register the publisher and the call to register the subscriber

AMS Lessons Learned – Threads (cont)

- In more complex scenarios that involved interaction between more than 2 applications having AMS publishers and subscribers, we found the results to be inconsistent:

    - We found that some subscribers would never receive messages from a publisher unless a 1 second delay was added after the call to register the subscriber

   - This problem only occurs with certain subscriptions and does not happen every time

AMS Lessons Learned – Startup Sequence
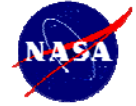
- MIRA consists of 4 AMS applications that need to be started in sequence. If one fails to start, then all of them have to be shutdown and restarted.

    - It appears that TCP sockets opened for communications between the senders and receivers are not released. Subsequent attempts to connect to the receiver will fail. The only solution is complete shutdown and restart of the sequence from the beginning.
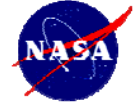
MIRA GUI Lessons Learned
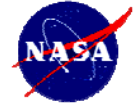
- Users prefer a single view:
  - Camera commanding
  - Command Recording VCR buttons
  - Pan/Tilt Panel
  - Sliders for Zoom/Focus/Iris

- GUIs must support mouse or joystick operation

- Manage User Experience Expectations:
  - Encoder lags can diminish the GUI experience (Solution: Faster Encoders)
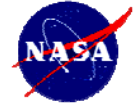
MIRA GUI Lessons Learned

- A heartbeat from the provider through the stack to the GUI adds value

- Java works well as a development environment:
  - Supports game controllers and embedded video.
  - Linux and Windows

Team Lessons Learned

- Develop on your target platform:
  - Minimizes porting issues
  - Java: write once, test everywhere, not write once, run anywhere

- Build an environment spec sheet:
  - Capture environment constraints and the performance specs of the final end-to-end system
  - Develop on the least-capable platform
  - Develop within available bandwidths
    - Identify uplink and downlink constraints
    - Build with flexibility where constraints may vary

- Develop configuration management strategy at project inception