Create and Deliver Superior Products Through Innovative Minds

Architectural Approaches to Autonomy Integration in UxV Intelligent Control and Autonomous Replanning of Unmanned System (ICARUS)

Advanced Development Programs ®

John G. Clark
817-777-9463
john.g.clark@lmco.com

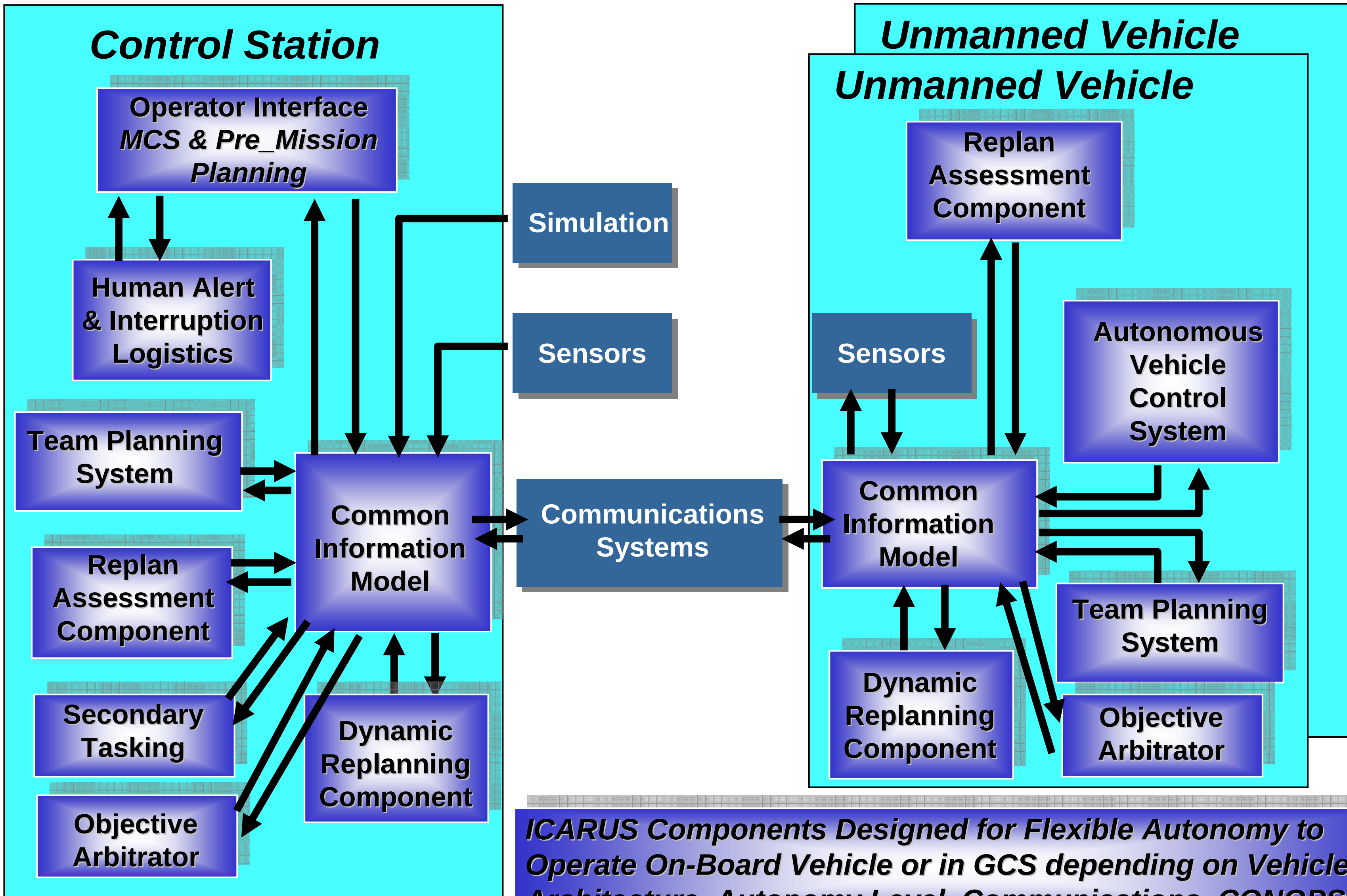
Architectural Integration Breadth

- **As part of the ONR Intelligent Autonomy program, we developed ICARUS and integrated other IA team member capabilities into the ICARUS framework**
  - ***SOA based approach used for organic ICARUS development***
  - ***External architectural compatibility varied component by component***

- **Integration path for each component had unique characteristics challenging the overall architectural approach**

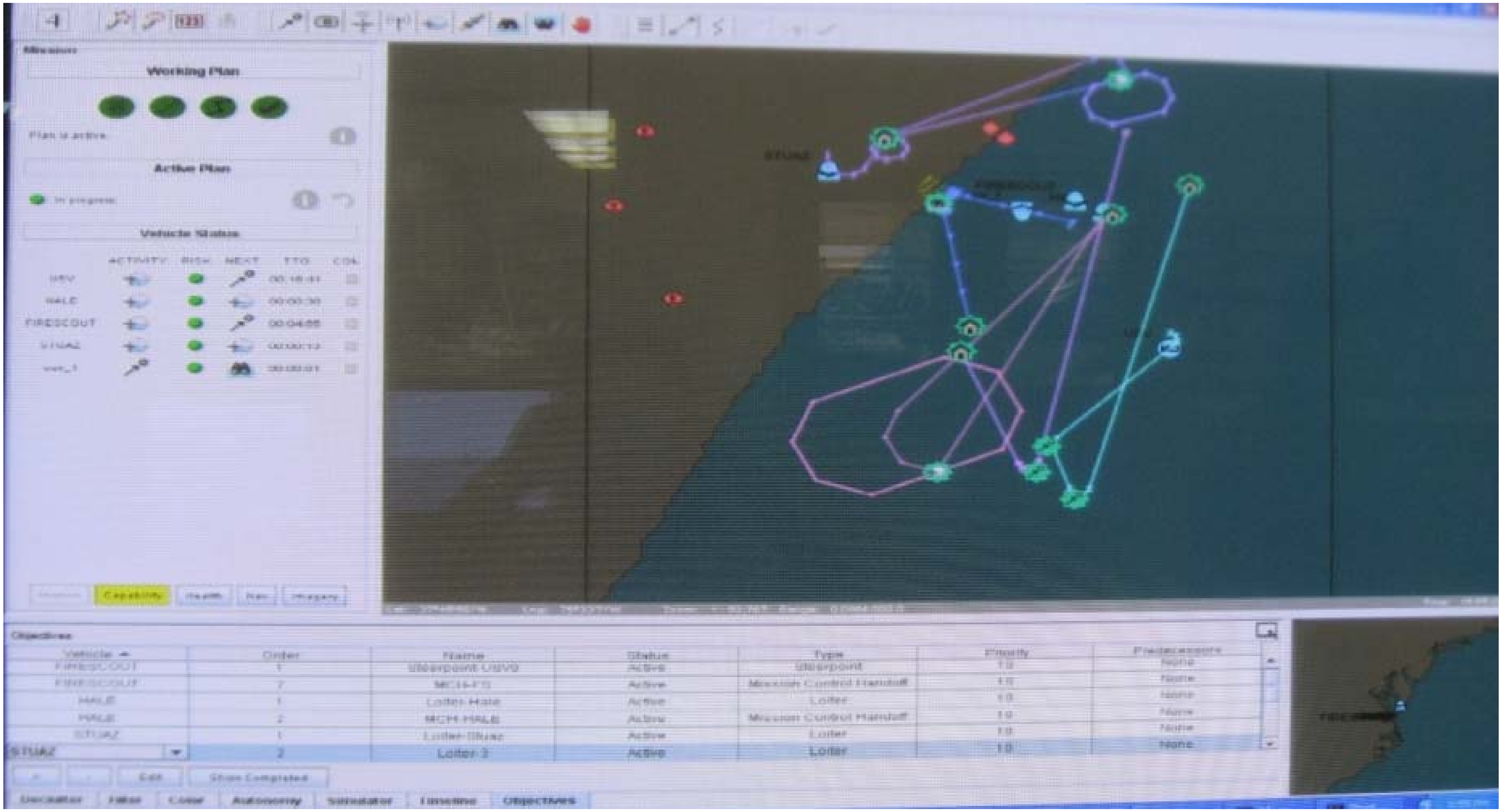
- **The LM team successfully tied several technologies into the ICARUS framework:**
  - ***NAVAIR ACETEF Simulation Tools (JIMM)***
  - ***USRDL Fire Scout Simulation***
  - ***Charles River Gantt Chart Timeline Display***
  - ***Draper RMDR***
  - ***UPenn Planning technologies (STOC)***
  - ***GT Mission Specification (Pre-Mission Planning)***
  - ***MIDAS (Aptima)***

ICARUS System Architecture



ICARUS Components Designed for Flexible Autonomy to Operate On-Board Vehicle or in GCS depending on Vehicle Architecture, Autonomy Level, Communications, CONOPS

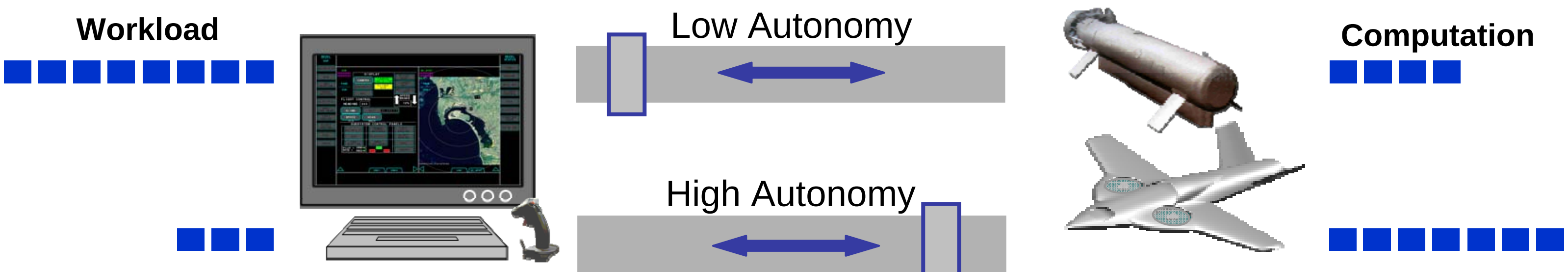
ICARUS Live Asset Demonstration (1 of 2)



ICARUS Live Asset Demonstration (2 of 2)

Flexible Autonomy

Formulate Control System to Match Operator and Mission Requirements



Adjustable Autonomy

- Supports modifying the level of responsibility of system decision making versus operator decision making
  - Example: High-level commands versus waypoint navigation versus direct control
- Operator can delegate authority for specific decisions to SW
- Must balance trust versus workload reduction

Mixed-Initiative Adjustable Autonomy

- Adjustable Autonomy in which operator or system can adjust autonomy level based on workload, emerging mission needs
- System alerts operator for assistance
- System works in concert with operator providing required data and mission parameters at the appropriate time

Flexibility to Meet UxV Requirements

Derived System Architecture Principles

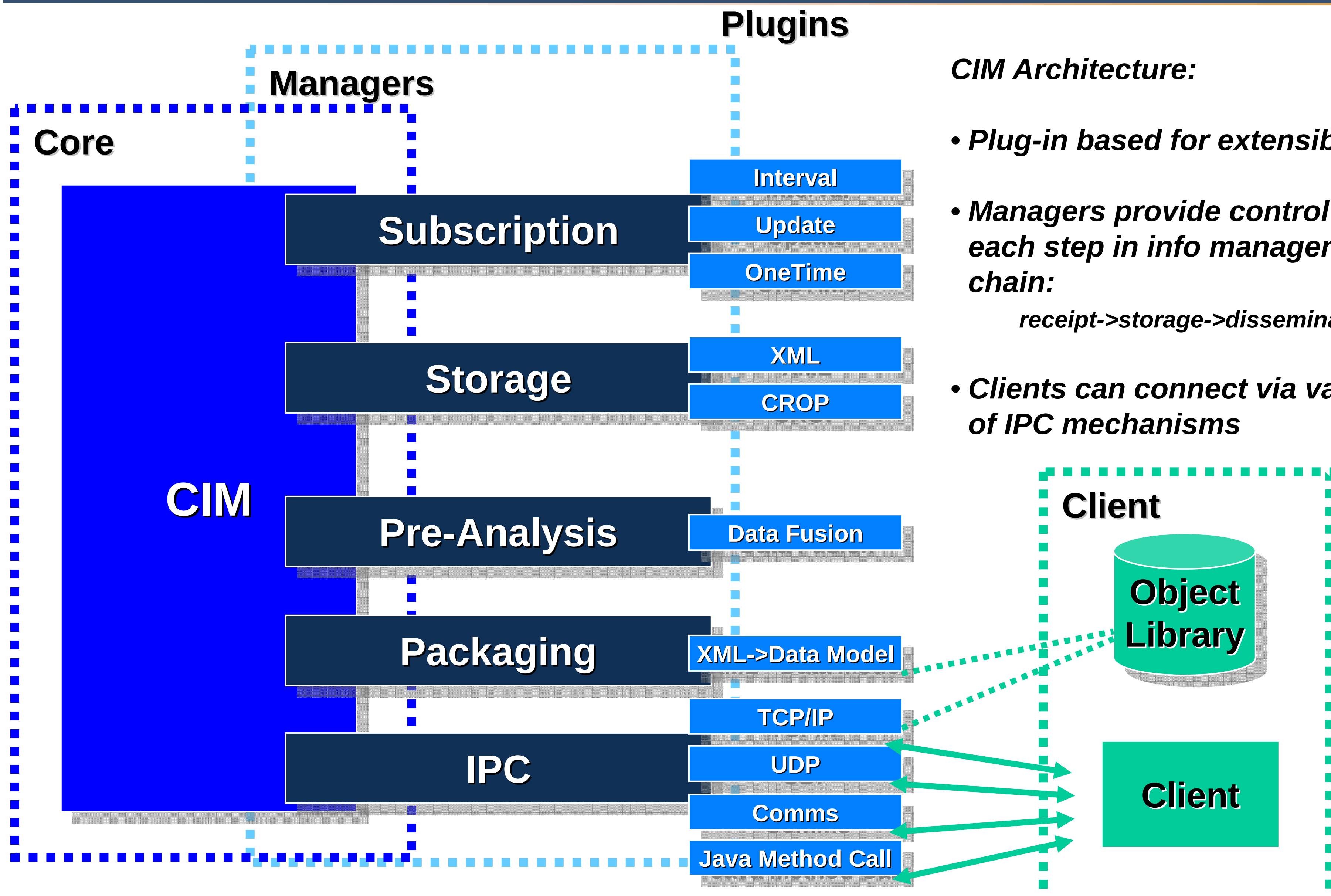
- **ICARUS Architecture Formulated as a "Service Oriented Architecture"**
  - ***Modular Components***
  - ***Connected through Publish/Subscribe Mechanism***
  - ***Components function as service providing an autonomy capability to the system***

- **XML used as Primary Internal Data Format**
  - ***Additional DIS Interface for Simulation***

- **System Architected to allow Communication with Other Formats/Protocols**

Publish/Subscribe enabled by LM Developed Software

Common Information Model



CIM Architecture:

- ***Plug-in based for extensibility***
- ***Managers provide control of each step in info management chain:***

  receipt->storage->dissemination

- ***Clients can connect via variety of IPC mechanisms***

CIM Support for Integration

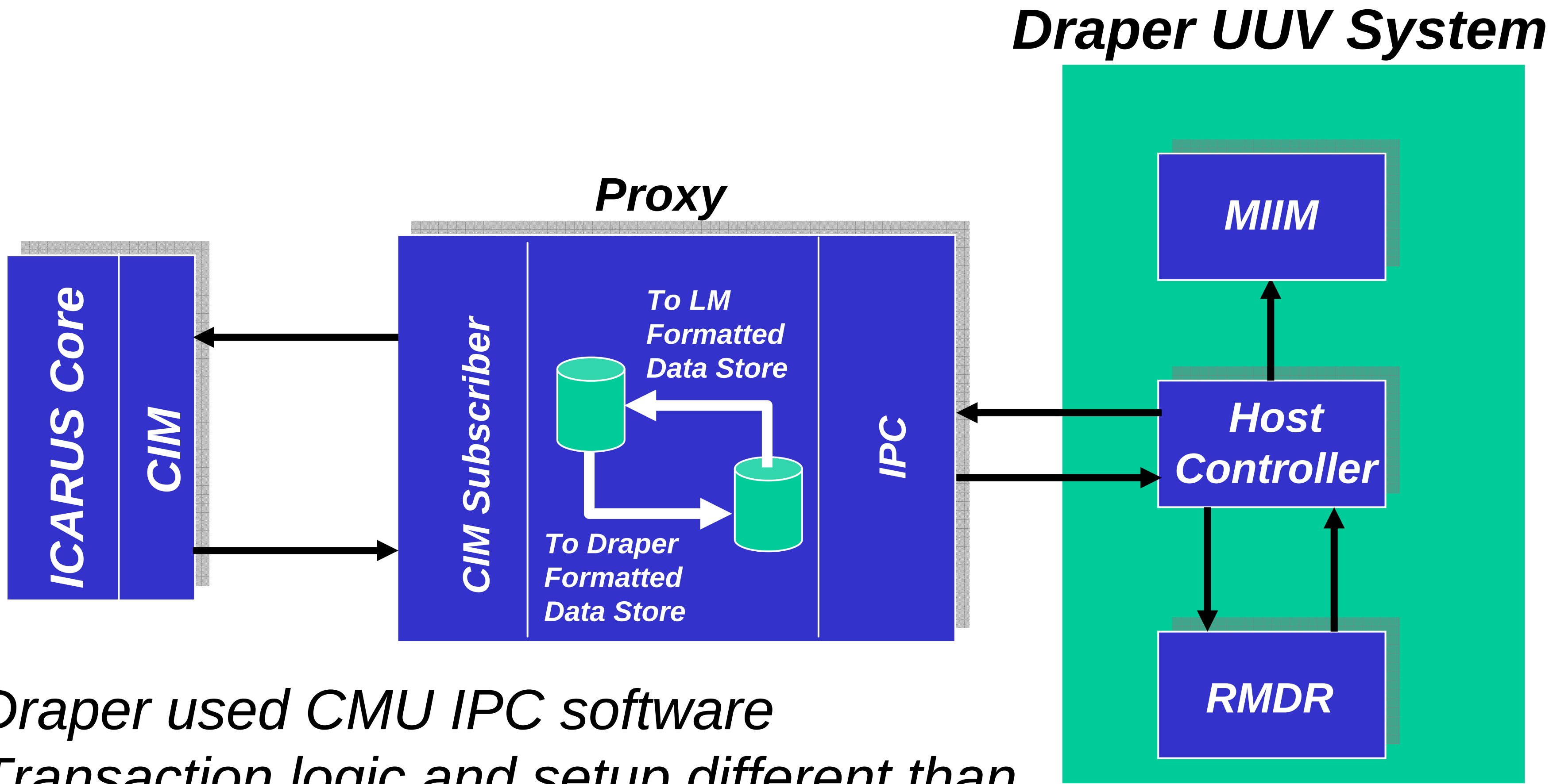
- **Provide a suite of debug/test tools for component testing**
- **TCP, UDP and direct communication IPC plugins**
- **Handling of disconnecting and reconnecting subscribers**
- **Provide user guide and example clients in Java and C++**
- **Schema updated for all Phase III additions and improvements**

- **Integration of new components with CIM has been shown to be fast due to:**
  - ***Example integrated clients***
  - ***CIM user guide***
  - ***Data flow diagrams***
  - ***Debug and test tools for integration (injector, scheduler, verification, etc.)***

CIM Benefits and Limitations

- **Benefits**
  - ***CIM resulted in very streamlined integration for all LM team components***
    - Well defined XML schema with variety of messages available for consumption (subscriptions)
  - ***Simplified integration with external components***
    - STOC, Mission Lab
  - ***Integration with new components could be accomplished in weeks…not months!***
    - We provided code examples that could be easily reused
- **Limitations**
  - ***Scalability***
    - XML messaging began to burden the network
    - Synchronous health and status updates
  - ***Latency***
    - As network traffic increased, latency of messages limited system effectiveness
    - Hub and Spoke Network simplified integration testing but introduced unnecessary message routing
    - Latency most prevalent in system *business logic*

Integration with Different Pub/Sub Systems



- *Draper used CMU IPC software*
- *Transaction logic and setup different than CIM*
- *IPC software procedurally sensitive*

Pub/Sub Compatibility is not a given!!!

Other Integration Approaches

- **CRA and Aptima integration into the OI based on OpenMap framework**
  - ***CRA module essentially used as a plug in through an exposed interface***
  - ***Aptima was integrated as another layer on map layer of the OI***

- **UPenn and GT work integrated as stand alone autonomy services**
  - ***Each had messaging designed and dedicated for the unique service***
  - ***Each were integrated through the CIM***
    - GT mission specification objectives were imported into the OI via the CIM
    - STOC planning process interacted with several components through the CIM

Results and Lessons Learned

- **As the system became more sophisticated, the complexity of updating individual services increased**
  - ***Adding new capability without affecting current functionality***
- **Installation process for setting up full system is intensive**
  - ***Each of the ICARUS services requires configuration***
  - ***Required an detailed installation CD that removed the user from many of the configuration processes***
- **Proxy communication between ICARUS XML Schemas and RMDR XML Schemas achieved most of the desired integration necessary for ICARUS to control RMDR vehicles and receive desired health and status information**
  - ***Differences in system implementation limited the amount of autonomy between the two systems***
    - For example, the systems handle contingencies differently
  - ***Future integration of disparate C2 systems should ensure that they are working on same CONOPS foundation***
- **Distributed Integration enabled by the SOA architectural approach helped facilitate quicker identification of configuration problems that don't show up in a single lab environment**

SOA Results

- **XML is a good encoding mechanism to be used during the investigative portion of the system development:**
  - ***Great for R&D environment***
  - ***allows for easy and quick modification of the message schema and component interfaces.***
  - ***for Java and C++ based components, integration with XML can be greatly simplified by using an automatic code generation tool for XML marshalling such as Castor for Java***
  - ***once the component interfaces are finalized, message exchange/handling rate can be improved significantly by switching to a custom static encoding mechanism since XML parsing is not efficient.***

- **Configuration management of schema changes is extremely important to maintain system stability throughout development**
  - ***Define full schema early***
  - ***Modify rarely***

SOA Lessons Learned

- **Thorough component-level testing is extremely important as it influences the duration and quality of system integration significantly**

- **Elimination of integration challenges enabled team to spend more time looking at the algorithms and system functions**
  - ***Quick way to test algorithmic processes in a system wise context***

- **It is very helpful for the infrastructure to provide component-level testing tools that allow distribution of canned messages and format verification of received messages**

Summary

- **Significant progress has been made in an architecture for Multi-Vehicle Control and Autonomy in Unmanned Vehicles**
  - ***Additional Growth Remains***
- **A Service Oriented Architecture Illustrating Autonomy has been tested and shown excellent properties for future development**
  - ***Scalability, Extensibility, IP Based, etc…***
- **Fielding of an SOA based system will need alternative approaches to meet rigorous military demands**
  - ***Find best mix of XML and binary messaging***
  - ***Migrate away from Hub and Spoke configurations***