



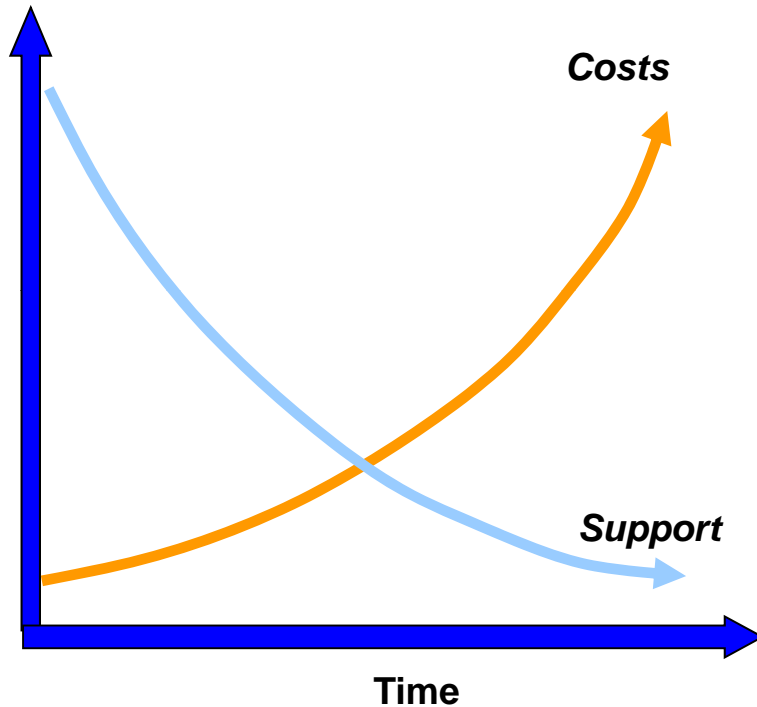
GSAA 2010

Evolution of a Service Oriented Architecture (SOA) C2 System – Advantages Sought, Lessons Learned, and Product Philosophies

Ryan Telkamp



In 2006, Boeing MCC operations were challenged by obsolete systems and dwindling support



- **Command and Control Systems became obsolete**
- **Computer hardware was unsupported**
- **Maintenance costs were increasing**



Every satellite operator shares this dilemma



- Inability to acquire maintenance support
- Ebay® is a necessary repair option
- Spare parts consuming valuable real estate



Obsolescence, along with closed architectures, jeopardize ground investments



- Vendor consolidation has eliminated some chip architectures
- Proprietary systems leave little or no room for evolution





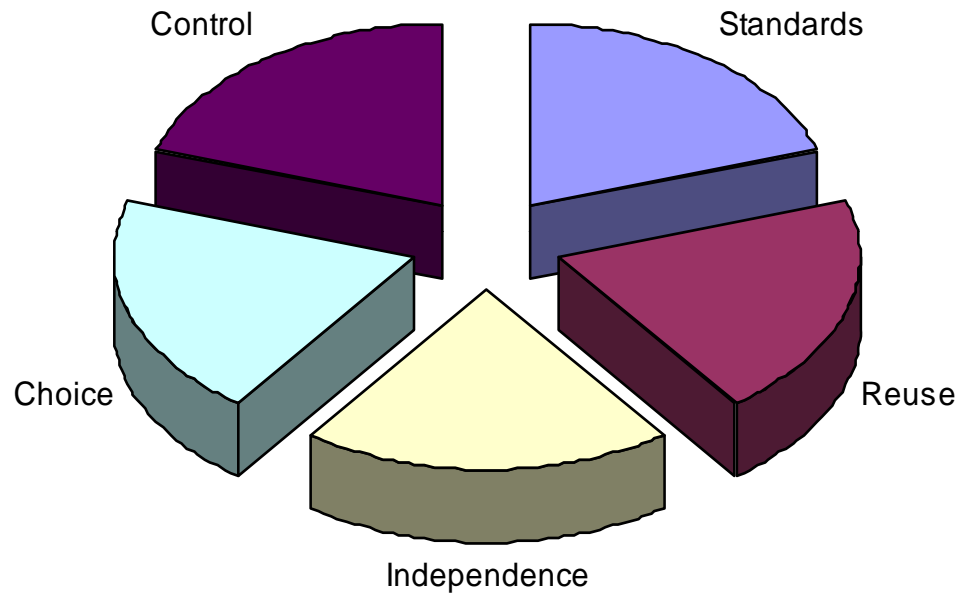
An affordable, open, vendor neutral system is achievable with the right plan



- **Boeing has chosen to pursue a non-COTS centric solution**
- **An open architecture, independent from specific vendor hardware or software**
- **An architecture neutral to different operational concepts**

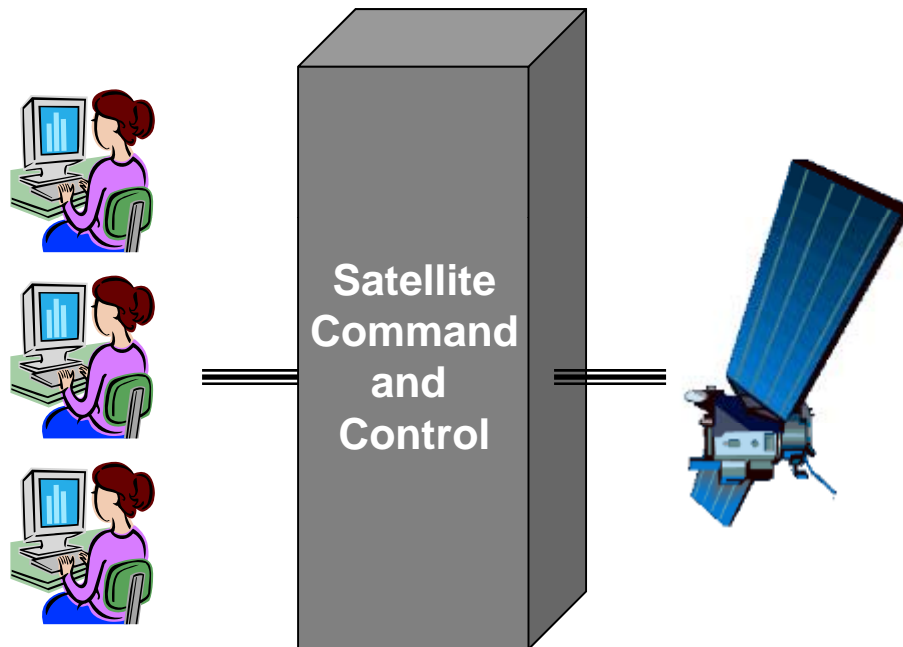


Develop a vendor neutral architecture that safeguards operational investment



- **Leverage reuse**
- **Use industry standards where appropriate**
- **Design the system to be independent of details within embedded COTS components**

Mitigating obsolescence



- Provide standardization through neutral Applications Programming Interfaces (API)
- Isolate COTS components behind the neutral API
- Loosely coupled services, provided through a Service Oriented Architecture (SOA), enhances flexibility
- SOA aids independence from a fixed operational concept

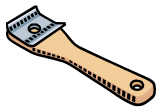
A tool box philosophy enables multiple operational concepts



Common Control Functions



Common Display Functions



Common Data Access Functions



Variable Application Methods

- Design services to provide generic algorithms
- Allow the Mission to define the algorithm sequence, not the system
- Select the graphical display language (icons) for openness and transportability
- The graphical display language must be robust, expandable, and easily support the API
- Develop interfaces to a variety of computer languages
- Support simple, yet powerful, languages for automated procedures



Balance make – buy - open source for decreased cost of ownership



- Strive for a manageable balance of software types for the best Cost of Ownership
- Not all open source licenses are created equal, assess the terms for your operation
 - » Assess “Share Back” requirements
 - » Potential Intellectual Property and ITAR / Export concerns
- Open source software can provide cost mitigation
- Research the open source product team like any vendor
- Leverage reuse where you can

Open Source Trade Space Requires Careful Assessment

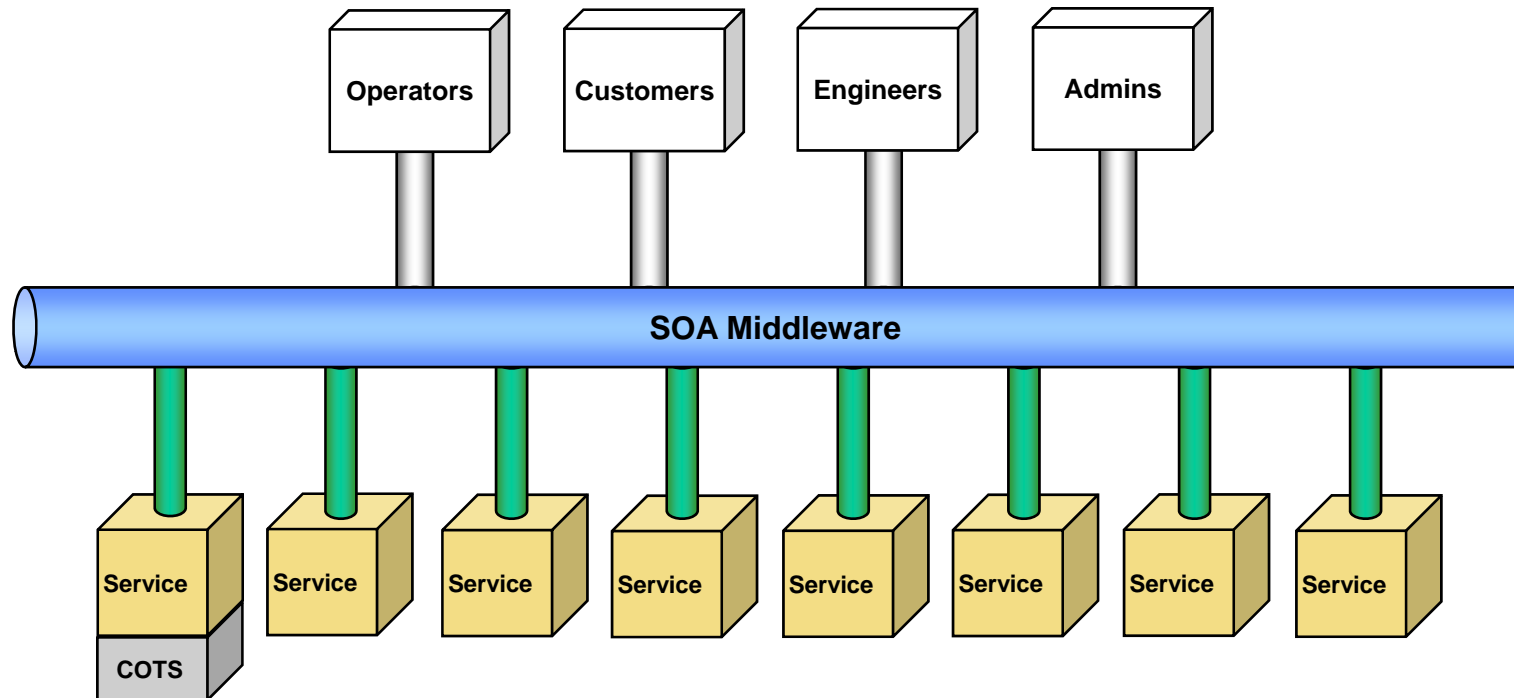


Boeing developed the Advanced Satellite Control System (ASCS)

- **A real-time Satellite command and control system, providing telemetry and command services, procedure execution, and user displays**
- **A vendor agnostic architecture that safeguards operational investment**
- **Works as a replacement of, or in conjunction with, an operator's COTS based satellite command and control solution**
- **Technology Readiness Level (TRL) 7, TRL 8 in Q3 this year**



Boeing Developed a vendor neutral architecture



- SOA Web Services and Publish/Subscribe Framework
- Tcl/Tk language provides open, transportable, presentation layer
- C, C++, C#, and Java language bindings
- Broad library of tools



Availability

- **ASCS is the Boeing standard for all Commercial, Civil, and DoD missions**
- **Boeing is offering to Customers ASCS as a ground framework for the flight operations of single satellites, fleets*, and constellations****

* Fleets - multiple satellites of the same model and vendor

** Constellations - multiple satellites of different models and vendors

ASCS provides future Possibilities