A Partitioning Communication System for the MILS Architecture

OMG Real-time Workshop
July 2004

Jeff Chilton
Objective Interface Systems
Herndon, Virginia

Project Funding Sources

- US Air Force
- Lockheed Martin
- Objective Interface

Agenda

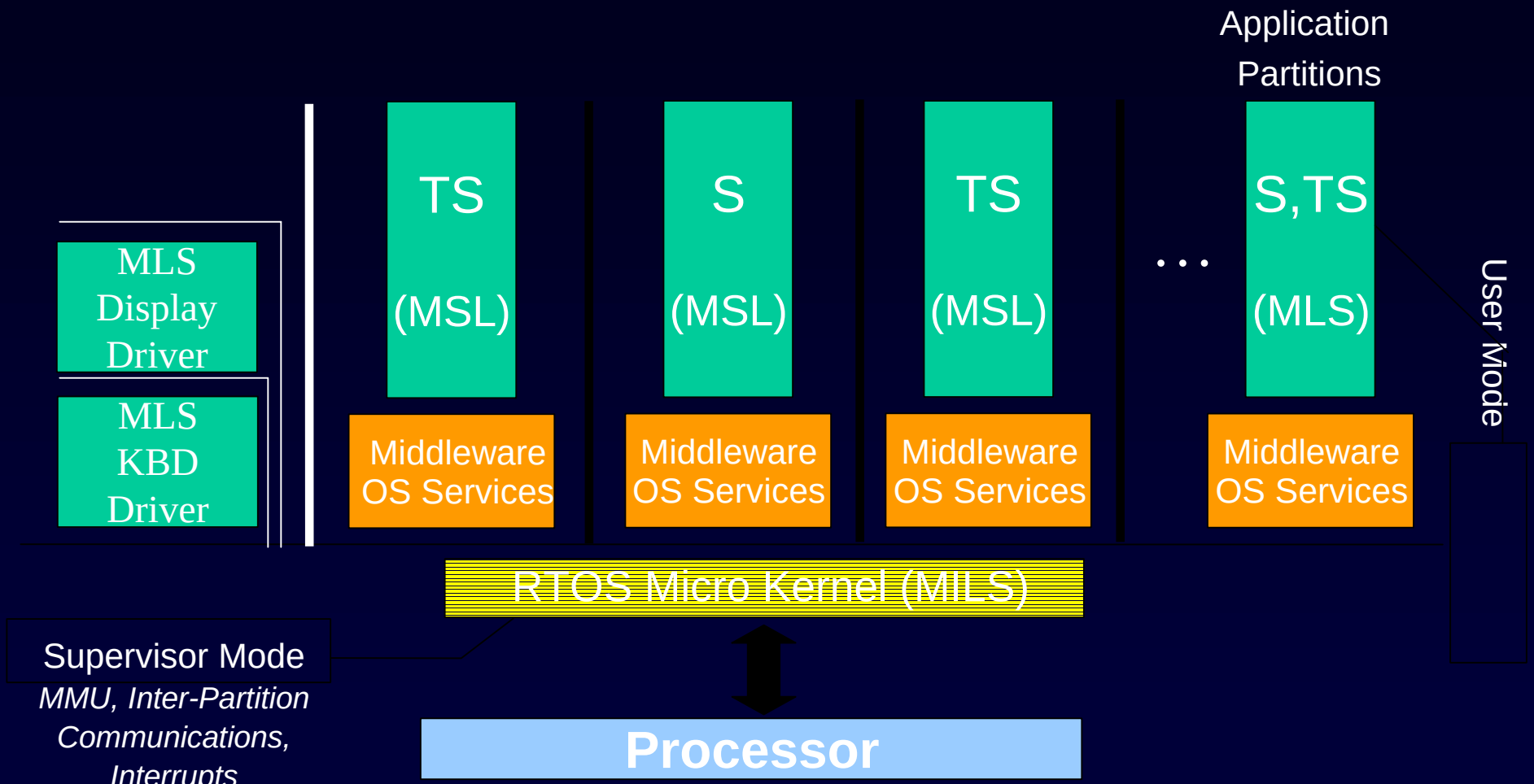
- Security & Embedded Systems
- Where a PCS fits in MILS
- How Did We Get Here?
- What the PCS is not
- What the PCS is Similar to
- The PCS & Real-Time
- The PCS's Environment
- PCS Security Functions
- "Challenges"

Security & Embedded Systems

- **Australian Water Utility**
  - Vitek Boden, 48, April 23rd, 2000, Queensland, Australia
    - disgruntled ex-employee of equipment supplier
    - Vehicle became command center for sewage treatment
    - Controlled 300 SCADA water and sewage nodes
    - "*was* the central control system" during intrusions
    - Released millions of liters of sewage
    - Killed marine life, blackened creek water, bad stench
  - Caught on 46th attempt
    - Was angling for a consulting job to "fix" the problems he caused

- **Result of embedded systems without security**

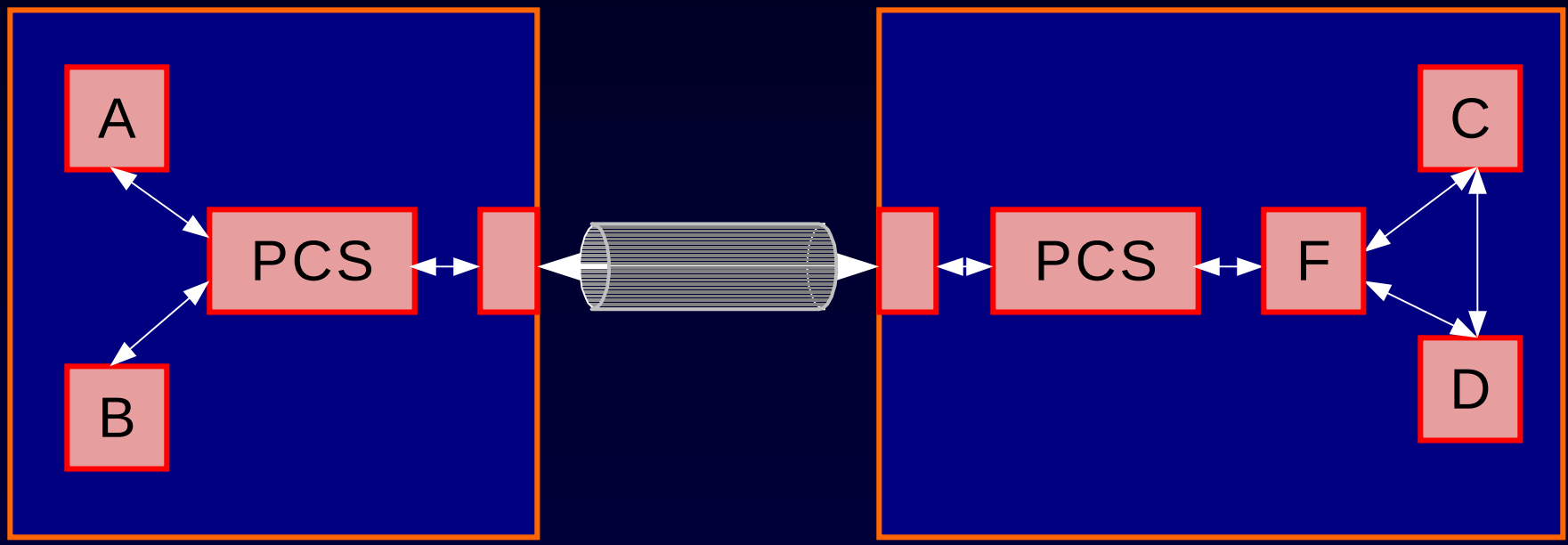
MILS Architecture



Where a PCS fits in MILS

- The PCS is communications middleware for MILS
- Always interposed in inter-node communications
- Might be in some inter-partition comms. Too
- Parallels Separation Kernel's policies

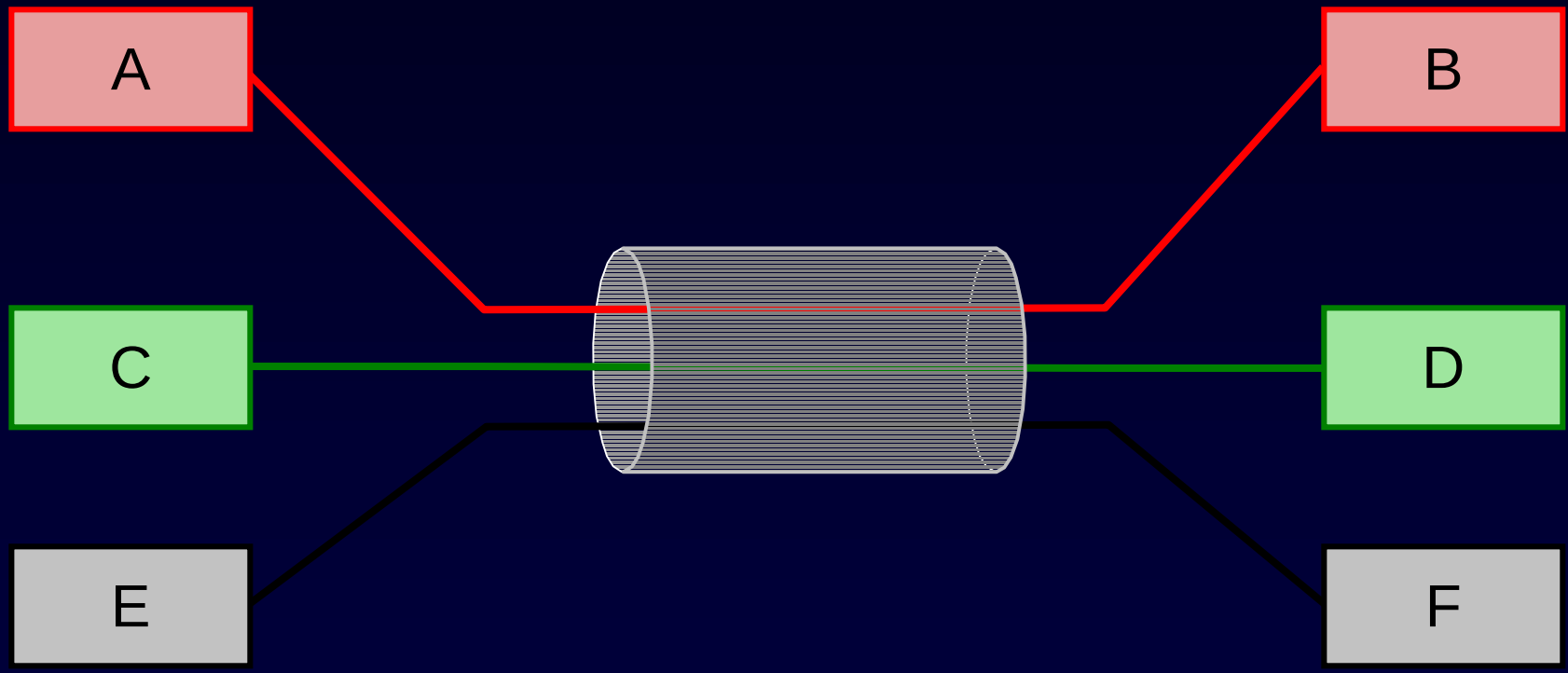
Inter-node Communication



Basic Security Policies

- **The PCS projects the Separation Kernel's security policies throughout a distributed system**
  - Information Flow
  - Data Isolation
  - Periods Processing
  - Damage Limitation

Partitioning the Channel



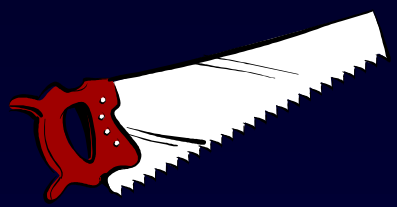
Properties of Channels

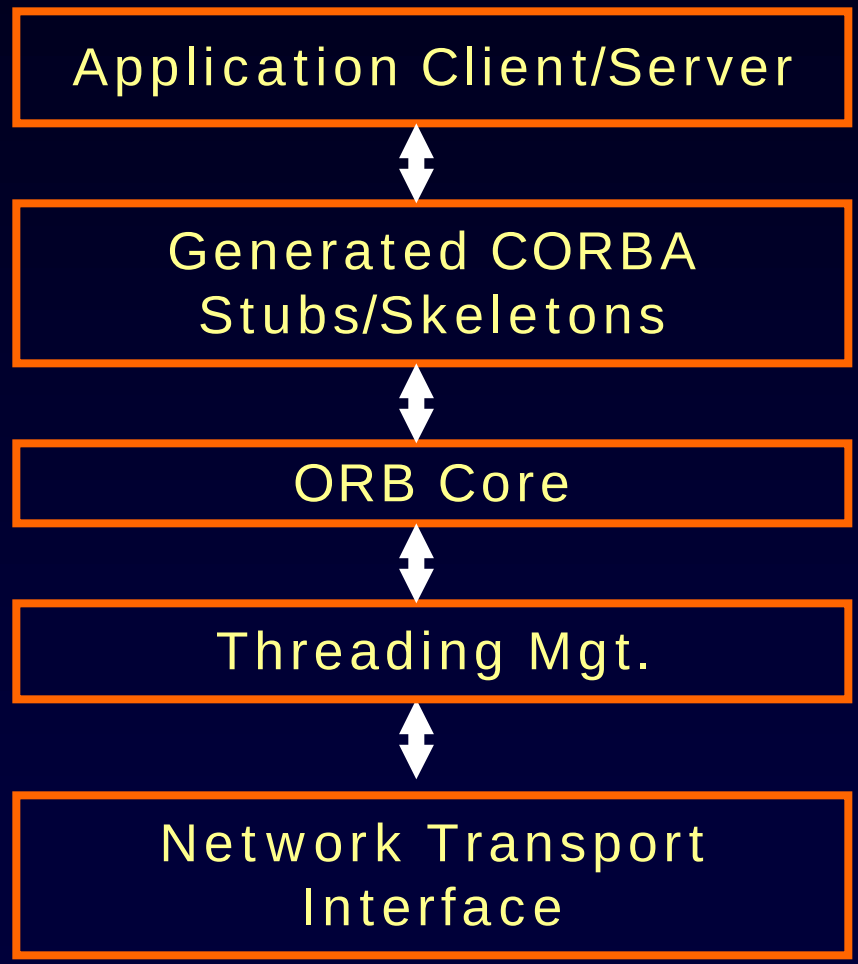
- Known or predictable capacity and latency
- Reliability delivery, message integrity
- Confidentiality protection
- Observability protection

How Did We Get Here?

- Started with CORBA for MILS
- Conducted a threat collecting exercise
- Presented multiple partitioning architectures
- Problems with ORB-in-a-partition
- Some parts must be in same partition as client
- What's left is a PCS

Where to Make the Cut?





What the PCS is not

- **Not a Guard or Application Firewall**
  - Doesn't examine message content
  - Can't enforce security policies delegated to the application layer

What the PCS is Similar to

- **Some functionality in common with Kang and Moskowitz's "Pump" (1993)**
  - "…to provide quantifiable security, acceptable reliability, and minimal performance penalties by interposing a device (called the Pump) to push messages to the high system and provide a controlled stream of acknowledgements to the low system."

The PCS & Real-Time

- **Real-Time and Anything…**
- **One-way information flow policy means:**
  - No acknowledgements allowed;
  - Blind up-writing is unreliable... Unless;
  - We're sure the receiving end will keep up.

The PCS's Environment

- **Policies**
  - P.DATA_ISOLATION
  - P.INFO_FLOW
- **Threats**
  - T.COVERT_COMM
  - T.EAVESDROP
  - T.MASQUERADE
- **Assumptions**
  - A.PARTIONING_KERNEL
  - A.REAL-TIME_KERNEL

PCS Security Functions

- **Information Flow Policy enforcement**
- **Channel capacity (bandwidth) management**
- **End-point authentication**
- **Message encryption (optional)**
  - For confidentiality on the wire
  - For down-grading ahead of a less-trustworthy device driver
- **Traffic padding (optional)**

More on Information Flows...

- **"Crosstalk" – disallowed comm. between subjects**
- **Subject to external entity**
- **External entity to subject**

Finally, CORBA Again

- **MILS**
  - Implementation location as configuration
  - Result of rigorous analysis
- **CORBA**
  - Implementation location transparency
  - Focus is on flexibility

"Challenges"

- Synchronization of distributed configuration info.
- Crypto key management
- Interaction with kernel's time-partitioning
- Composition from assured components

Contact Information

Objective Interface Systems, Inc

www.ois.com

+1 703 295 6500

Jeff.chilton@ois.com