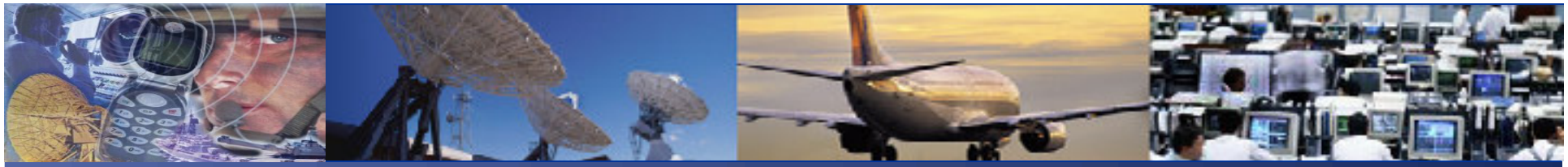
PRISMTECH

Productivity Tools & Middleware

Building High-Assurance Systems out of Software Components of Lesser Assurance Using Middleware Security Gateways



OMG's First Software Assurance Workshop:
Working Together for Confidence
Fairfax, VA, March 6, 2007

Presentation Outline

- Introduction
  - Motivation
  - Background
- Distribution Middleware Level Security
  - Application Level vs. Distribution Middleware Level Security
  - Domain Protection Approach
  - Supported Middleware Types
  - Enforcement Aspects
- Application Level Security Management
  - Security Policy Creation and Definition Support
  - Security Policy Adaptation and Administration Support
- Addressing Information Assurance
  - Information Assurance @ Application/Middleware Level
  - Defense-in-Depth
  - What about MILS ?

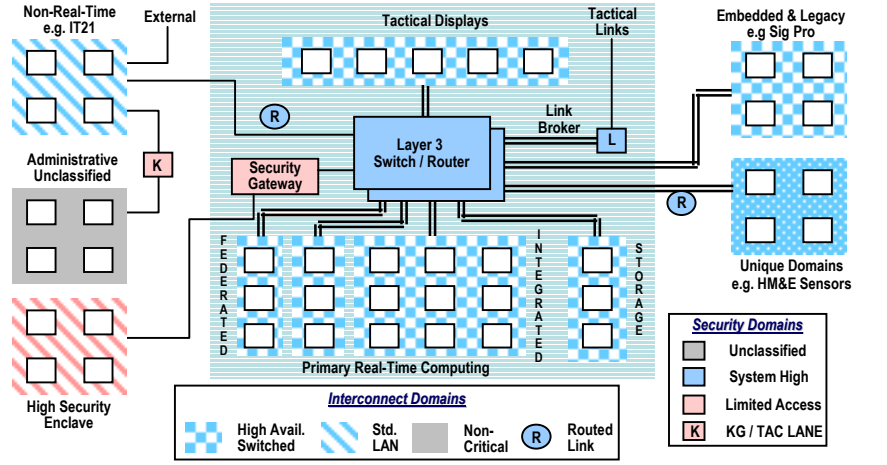
Motivation

- PrismTech involved in a number of projects where customers build systems (and systems of systems) with high information assurance requirements
  - based on distribution middleware
  - as much as possible from COTS hardware and software infrastructure products
  - using several types of standardized middleware
    - CORBA
    - OMG Data Distribution Service for Real-time Systems (DDS)
    - XML/SOAP/WSDL
  - following a paradigm of interacting components
    - Client/Server closely coupled
    - SOA-like (loosely coupled)
    - Publish/Subscribe (LAN, WAN)
- Problem: How to keep control how the components effect each other
- One approach: declarative security enforcement for component interactions by middleware-level security gateways
- What about improving the assurance of the overall system by smart partitioning into components and enforcing declarative security controls on the interactions?

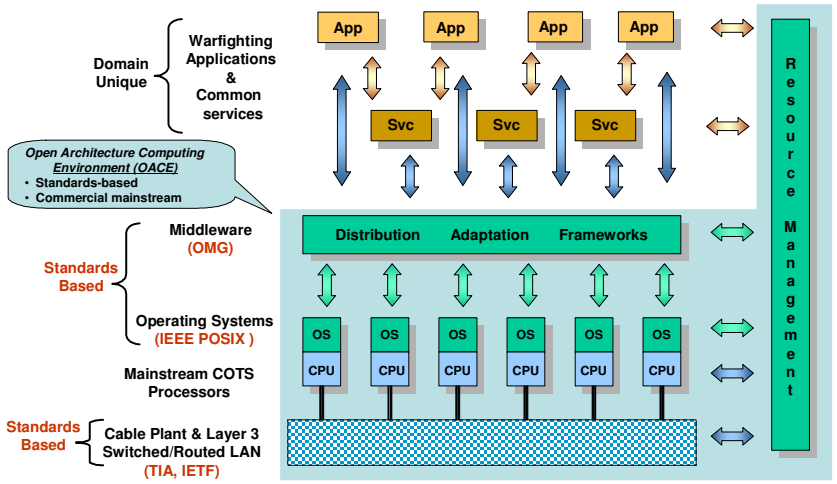
Background: OACE (example multi-middleware CE)

- A Technical Architecture for systems of systems mandated by the US Navy… Mainstream Standards Based... Open Architecture Computing Environment (OACE)
  - Middleware
  - Operating Systems
  - Mainstream COTS Computing Technology
- A Functional Architecture…Technical Reference Model That Identifies Software Domains and Interface Relationships
  - Warfighting Applications
  - Common Service Applications
- Specifications, Standards & Design Guidance
- An OA Approach is an Integrated Technical and Business Strategy Focused on Increasing the Effectiveness of a Capability, While Decreasing Cost, Increasing Operational Life and Providing Improvements Through Planned Upgrades

Network & Computer View



Operating System & Middleware View



- Components are defined and interact with each other using distribution middleware
  - CORBA
  - DDS
  - XML/SOAP expected

Background: PrismTech

- Providing distribution middleware, component technologies, and modeling/development environments (productivity tools) is PrismTech's core competence
- With its enterprise middleware security gateway product line (firewall like, for IIOP [CORBA, Java-RMI] and SOAP), substantial experience gained in the enforcement of security at the middleware layer following a gateway approach (incl. performance and high-availability issues)
- Where we're going: middleware security gateways for DRE environments
- Vision: such gateways together with appropriate modeling (deployment) support can also serve for higher assurance (bringing separation to the higher abstraction levels (inspired by separation kernel OSs)

Application Level vs. Middleware Level Security

- COTS best-practice security technology (network level or transport level encryption, firewalls, 3A products, IDS products etc.) does not support
  - application level authorization/access control
  - application level content inspection/filtering
  - application level audit

  for any of the distribution middleware expected to be used in OACE-like computing environments
- Application level security being built per application would be contrary to many goals of an Open Architecture (e.g., portability, reusability, cost reduction)
- Most appropriate architectural layer for application security enforcement seems to be the distribution middleware layer,

  …but lack of mature standards and/or products wrt. Security (CORBA, DDS)

Domain Protection Approach

- Question: What is the best approach (and implementation technique) for security enforcement to achieve information assurance (somehow similar to the question of best granularity of components)
- A fact of life: Application level security can best be implemented for interactions between software components (coherent function blocks under one security administration)
- Our proposal: using the domain concepts (familiar from dealing different security/classification levels) for the structuring wrt. security in general.

→ Security is achieved by domain protection. A domain can typically be

- A component server (EJB, CCM etc.)
- A sub network with a defined entry point.

- Sub network can be protected more effectively (one component of higher level evaluation) → Security Gateways

Middleware Types Targeted

- RT-CORBA
  - Well defined interoperability protocol
  - Well defined message formats
- DDS
  - Well defined interoperability protocol on the way of standardization
  - Well defined message formats
- Web Services
  - Well defined interoperability protocol
  - Well defined message formats
  - XML as format very widely used
- Further distribution middleware as soon as adopted

Typical Environmental Enforcement Aspects

- Distribution Middleware Security Gateway must be able to act autonomously, e.g., if network connection to infrastructure servers is lost. However, security policy changes (e.g., for reflective resource management) must be enforced without any delay.
- Distribution Middleware Security Gateway must itself be real-time capable (e.g. prioritization of message handling)
- Very high performance, delay, throughput, and jitter requirements

Security Policy Creation and Definition Support

- Security policies to be enforced by the middleware security gateway are formalized policies (XACML, possibly extended) in form of files to be held locally
- Are loaded and activated in a secure way (encryption)
- Can be subject to evaluation and certification
- Initial policies can be generated as a result of the modeling of the system to be protected
- Security policies (in the formal representation) can be analyzed/evaluated with regards to consistency in itself
- Security policies (in the formal representation) can be analyzed/evaluated with regards to consistency with the security policies of other domains

Policy Adaptation and Administration Support

- The middleware security gateways have a secure policy upload facility that interacts with a security policy server (push and pull modes supported).
  Each rule can have an annotation if it still applies in case the gateway cannot be sure the policy version locally available is still the most current one (e.g., if the policy server cannot be reached).
- Policy adaptation before deployment and binding
- Graphical User Interfaces for policy administration
- Programming Interfaces
- Command-line interfaces
- Automated tools for security policy versioning, audit, and roll-back.

Information Assurance @ Application/Middleware Level

- Application security policy enforcement between domains of the same security level (and other classifications) enables the implementation of a divide-and-rule security philosophy
  - High assurance only needed at the control/enforcement point (middleware security gateways, active security policies, incl. gateway software, OS hardware)
  - Not all software run within a domain must have the same high assurance
  - Effort and cost for the verification and certification of the overall system/platform (system of systems, potentially from different vendors) can drastically be reduced.
- Best practice security (encryption and network or transport level firewalling) cannot ensure this
- Application security policy enforcement between domains of different security levels can be made easier (requires MLS certification / future option)

Defense-in-Depth

- Security measures of a domain can be implemented following a defense-in-depth strategy, e.g.,
  - Packet filter, transport level or stateful inspection at the domain entry point
  - Encryption/decryption of the stream
  - The middleware security gateway
  - Application level security logic at the component server
- Security gateways enable the ad-hoc implementation of security checks without access or replacement of the applications
  - Third-party vendors
  - Immediate action required

What about MILS ?

- The middleware security gateway enables the implementation of the MILS philosophy (minimum chunks of code that need verification) for an increased overall system/platform assurance at the higher abstraction level of distribution middleware.
- It can be implemented on top of a MILS system, e.g., in a separate partition
- Separate domains which interactions are security controlled by the middleware security gateway can be implemented in different partitions of one MILS system

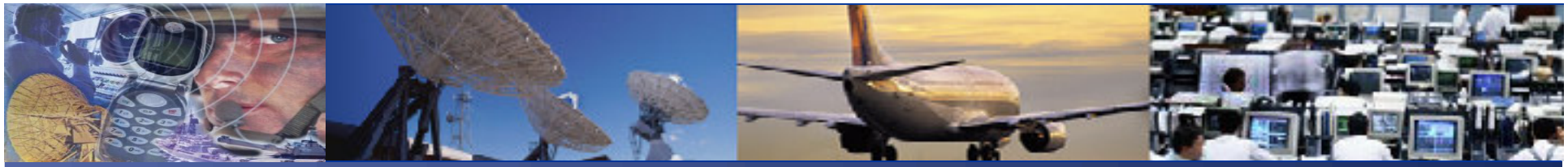
Multi-Purpose Component

- Often, CORBA based application systems and architectures (also in RTE systems) use object references as a means of implicit authorization.
- Implicit authorization (in this context) means handing over an object reference to a client object does not only provide the necessary addressing and context information but also authorizes the client entity to access the respective object.
- An implicit assumption in such schemes is that other entities which have not legally received an object reference for the object can not access the object. (object references are often provided by a CORBA Name Service instance only and access to the Name Services is restricted to legal client entities).
- The main advantage of the implicit authorization principle is that it avoids the need for the implementation of an additional logic consisting of explicit access rights basically just restate the interaction logic between the CORBA objects.

PRISMTECH

Productivity Tools & Middleware

Building High-Assurance Systems out of Software Components of Lesser Assurance Using Middleware Security Gateways



OMG's First Software Assurance Workshop:
Working Together for Confidence
Fairfax, VA, March 6, 2007