PRISMTECH
Productivity Tools & Middleware

Security Model and Enforcement for Data-Centric Pub/Sub with High Information Assurance Requirements

Sebastian Staamann, Director Security Products, PrismTech

OMG's Eighth Workshop on Distributed Object Computing for
Real-time and Embedded Systems
July 9-12, 2007 - Arlington, VA, USA

The Need for Information Assurance (IA) within DDS

- Data-Centric Publish/Subscribe middleware, most notably the Data Distribution Service for Real-time Systems (DDS), heavily used for systems of highest criticality
- DDS used for distribution (and storage) of *classified* information

- Information to be distributed within a system or a system-of-systems potentially of different classification
- Distribution over untrusted networks (network segments)

- Distribution middleware (DDS, CORBA, etc.) is the architectural layer in a distributed systems architecture at which application-level security policies can be defined and enforced, yet with generic, non-application-specific mechanisms/technology

- DDS-based information backbone is (more trustworthy) infrastructure, in contrast to (less trusted) application code

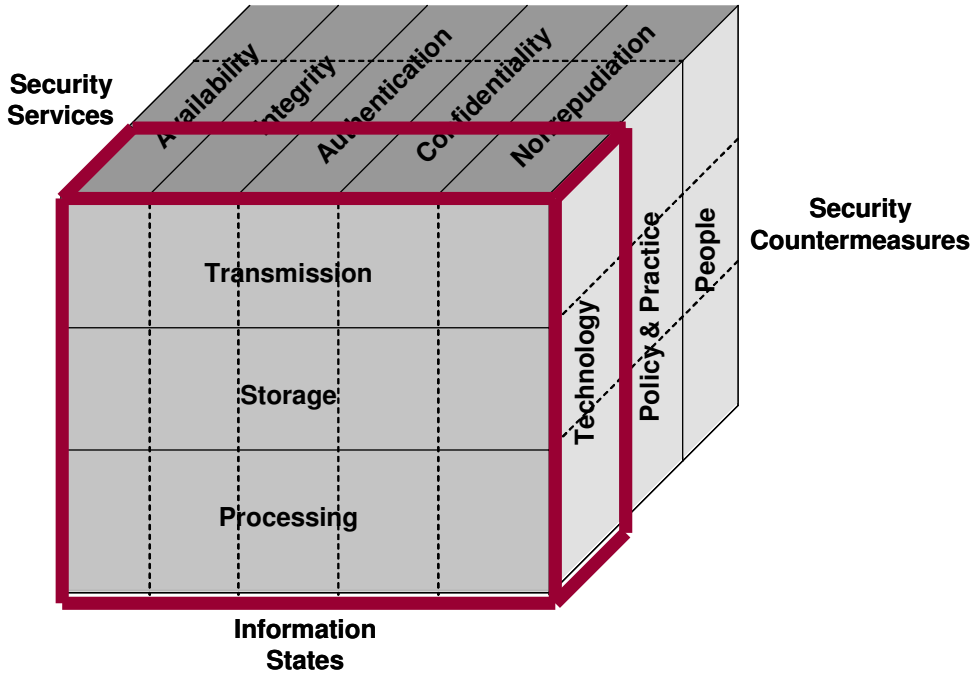
Constraints and Requirements on IA in Typical Application Environments of Data-Centric Pub/Sub

- DDS is the Data Distribution Service for **Real-time** Systems, i.e. required are:
  - Minimum latency
    - Minimum latency implementation of access control
    - Minimum use of cryptography
    - Efficient use of cryptography (minimization of encryption/decryption operation)
    - Using cryptography in modes with minimum latency
    - No unnecessary use of public-key cryptography
    - Non-blocking generation of security audit information
  - Minimum jitter
    - No blocking hand-shake protocols
    - No centralized access decision function servers
- DDS is pub/sub for multi-node systems
  - Full support for multicast needed, no peer-to-peer (hub-and-spoke) approaches as usually used for internet security (like SSL for transport-level communication security)
- Architecture support for RED/BLACK separation (nodes potentially connected via untrusted networks)
- Plug-in / replaceability of crypto algorithms/software/hardware
- **High IA**: Support for **formally proven** security policies and models

Proven Security Policies and Models for High IA

- Traditionally (especially in Defense), focus of security policies for high-IA environments on confidentiality of information
  - Implementing multilevel security for IT (security [secrecy] levels, compartments [need-to-know])
  - Controlling the information flow (containment of Trojan horse effects, covert channel suppression, RED/BLACK separation...)
  - Cornerstone: Bell-LaPadula Confidentiality Model
- Nowadays, especially with the processing of real-time control information, the integrity of the information has become equally important.
  - Model with proven foundation: Biba Integrity Model (sibling of Bell-LaPadula model)

Information assurance (IA)
MUST be supported/enabled by the appropriate technology



Extended McCumber Model
(chosen by the IA group in the NCOIC as the reference model for their work on the IA Design Framework)

Security Policy and Models briefly reminded

Military Security Policy:

Each piece of information is ranked at a particular sensitivity level, such as *unclassified*, *restricted*, *confidential*, *secret*, or *top secret*. Each piece of classified information (information object, or object for short) is associated to one or more compartments which represent the subject matter of the information. The combination ‹ *rank*; *compartments* › is the *classification* (C) of a piece of information. An entity (traditionally a person) that wants to access a piece of classified information must be cleared. A *clearance* (C) is an indication that an entity (a subject, traditionally a person) is trusted to access information up to a certain level of sensitivity and that this subject needs to know certain categories of information. Also the clearance of a subject has the form ‹ *rank*; *compartments* ›. A subject dominates an object, meaning the subject can read the information object if and only if

(i) the rank of the subject is at least as high as the rank of the subject ($rank_{object} \leq rank_{subject}$), AND
(ii) the subject has a need to know about all compartments for which the object is classified ($compartments_{object} \subseteq compartments_{subject}$).

Bell-LaPadula Confidentiality Model:

Two properties characterize the secure flow of information (i.e., the enforcement of the military security policy):

Security Property: A subject may have *read* access to an object only if the subject dominates the object, i.e., $C_{subject} \geq C_{object}$

*- Property : A subject *s* that has *read* access to an object *o* may have *write* access to an object *p* only if $C_p \geq C_o$

Biba Integrity Model:

In the Biba model, two properties guarantee the integrity of the information in the system:

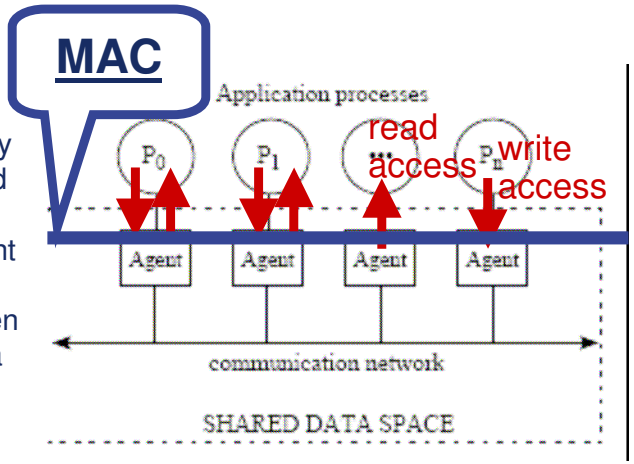
Simple Integrity Property: A subject can modify (have *write* access to) an object only if $I_{subject} \geq I_{object}$

Integrity *-Property: If a subject *s* has *read* access to an object *o* with integrity level I_o, *s* can have *write* access to an object *p* only if $I_o \geq I_p$

Building Distributed Application Systems Based on Data-Centric Pub/Sub Middleware – what does it mean for IA?

- Proven security policies and models for high IA are ***data-centric***
- The underlying interaction model for distributed applications systems based on data-centric publish/subscribe middleware also is ***data-centric*** (in contrast to object-oriented or service oriented distribution middleware like CORBA or SOAP/WSDL)

"The software architecture, named SPLICE, that we developed for distributed embedded systems basically consists of two types of components: applications and a shared data space. Applications are active, concurrently executing processes that each implement part of the system's overall functionality. Besides process creation, there is no direct interaction between applications; all communication takes place through a logically shared data space simply by reading and writing data elements." [1]



"Implementing Data-Centric Applications: Data-centric applications can be implemented using the precepts of data-oriented programming. In general, the tenets of data-oriented programming include the following principles: (i) Expose the data. Ensure that the data is visible throughout the entire system...." [2]

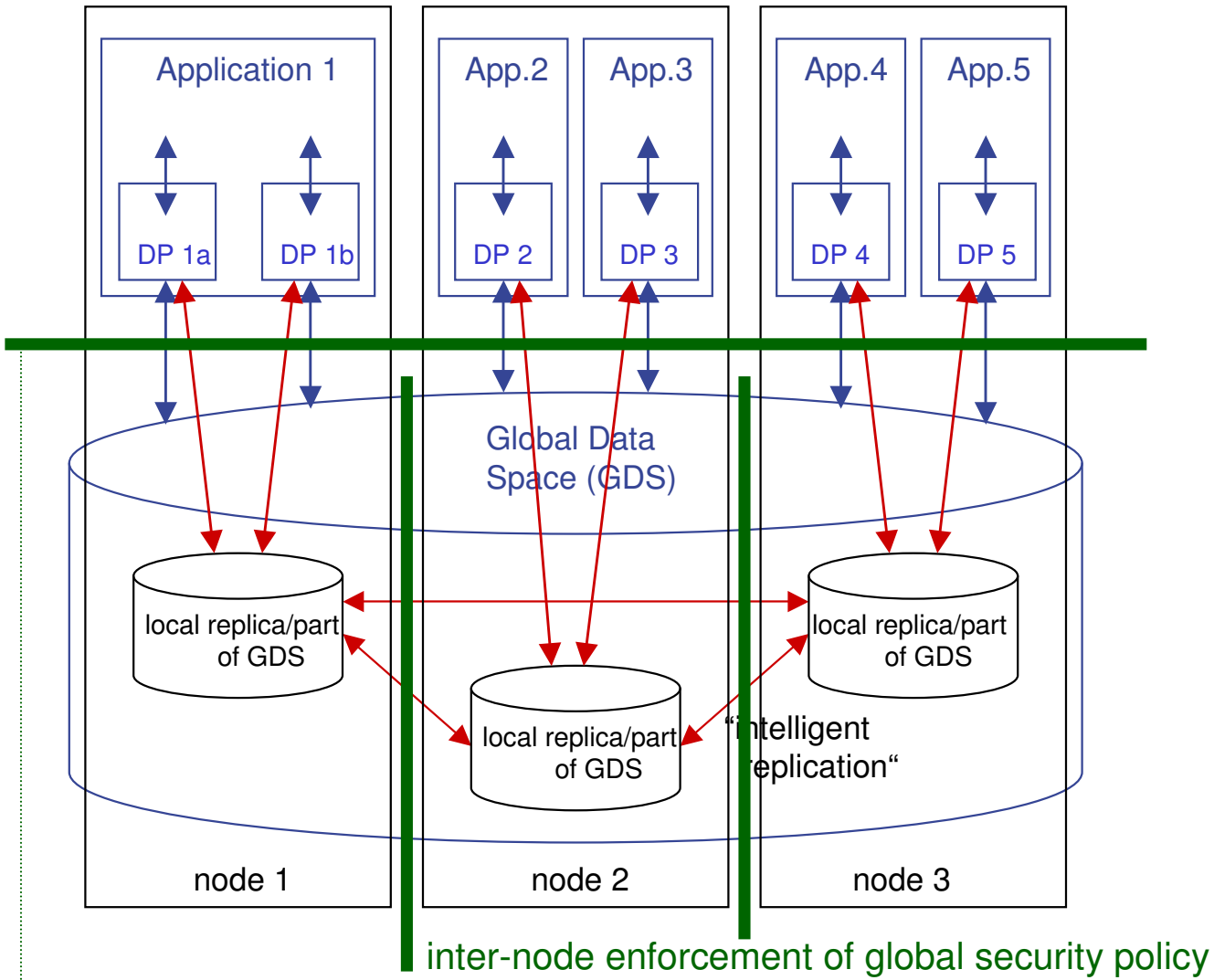
[1] M. Boasson and E. de Jong. "Software Architecture for Large Embedded Systems." *In Proc. of the IEEE. RTSS'97 Workshop on Middleware for Distributed Real-Time Software Systems*, San Francisco, CA, Dec. 1997.

[2] G. Pardo-Castellote and S. Oberoi. "A Data-Centric Approach to Distributed Application Architecture". White paper. March 2007. *www.devx.com*

→ **Direct applicability of proven security policies and models**
(interaction model: common access of all applications to a replicated database)

if the information backbone (replicated database) is implemented as a trusted infrastructure that enforces mandatory access control (MAC), i.e. is an IA-enabled information backbone

The DDS-based IA-enabled Information Backbone
for Building Secure Distributed Application Systems



- A Global Data Space is made accessible to applications through an DDS-based information backbone.
- The IA-enabled backbone enforces security controls on the (global) distribution (replication) of the data within the Global Data Space and on the access to the (local) replicas.
- The IA-enabled information backbone covers the technology part of the five pillars of IA:
  - Availability
  - Integrity
  - Authentication
  - Confidentiality
  - Non-repudiation

The Pieces of the Puzzle

- What is the scope of a security policy and model? What is the scope of a single security administration (what is the security domain / trust domain) ?
  The DDS Domain

- What are the (identifiable) principals (that can be authenticated) in the security domain?
  The applications, resp. the registered userIDs (human or system) that the applications run for (like processes in Unix)

- What are the objects to which access control is to be enforced in a DDS-based information backbone (global data space) ?
  ClassifiedDataPartitions and ClassifiedDataTopics

- What are the subjects that access these objects and that act for a principal (with the principals privileges) ?
  The Domain Participants

- What about nodes of varying trustworthiness?
  All nodes are under the same security administration and governed by the same security policy and model. However, the level up to which a node can be trusted to handle classified information appropriately may vary (due to hardware, operating system, physical exposure etc.). Nodes get assigned max. security levels. Inter-node security enforcement prevents distribution (replication) of higher-classified information to lesser classified nodes.

The Pieces of the Puzzle (cont.): Clearances & Classifications

- Security attributes (clearance) of a subject (Domain Participant, inherited from userID):
  - Secrecy level
  - Integrity level
  - Set of Compartments

  (all assigned by the security administration)
- Security attributes (classification) of *ClassifiedDataPartition* (object):
  - Secrecy level
  - Integrity level
  - Set of Compartments

  (all assigned by the security administration)
- Security attributes (classification) of *ClassifiedDataTopic* (object):
  - assignment to an (already existing) *ClassifiedDataPartition*

  (assigned by the security administration)
- Security attributes (classification) of node:
  - max. Secrecy level
  - max. Integrity level
  - Set of Compartments

  (all assigned by the security administration)

Mandatory Security Policy and Models looked at again

Military Security Policy:

Each piece of information is ranked at a particular sensitivity level, such as *unclassified*, *restricted*, *confidential*, *secret*, or *top secret*. Each piece of classified information (information object, or object for short) is associated to one or more compartments which represent the subject matter of the information. The combination ‹ *rank*; *compartments* › is the *classification* (C) of a piece of information. An entity (traditionally a person) that wants to access a piece of classified information must be cleared. A *clearance* (C) is an indication that an entity (a subject, traditionally a person) is trusted to access information up to a certain level of sensitivity and that this subject needs to know certain categories of information. Also the clearance of a subject has the form ‹ *rank*; *compartments* ›. A subject dominates an object, meaning the subject can read the information object if and only if

(i) the rank of the subject is at least as high as the rank of the subject ($rank_{object} \leq rank_{subject}$), AND
(ii) the subject has a need to know about all compartments for which the object is classified ($compartments_{object} \subseteq compartments_{subject}$).

Bell-LaPadula Confidentiality Model:

Two properties characterize the secure flow of information (i.e., the enforcement of the military security policy):

Security Property: A subject may have *read* access to an object only if the subject dominates the object, i.e., $C_{subject} \geq C_{object}$

*- Property : A subject *s* that has *read* access to an object *o* may have *write* access to an object *p* only if $C_p \geq C_o$

Biba Integrity Model:

In the Biba model, two properties guarantee the integrity of the information in the system:

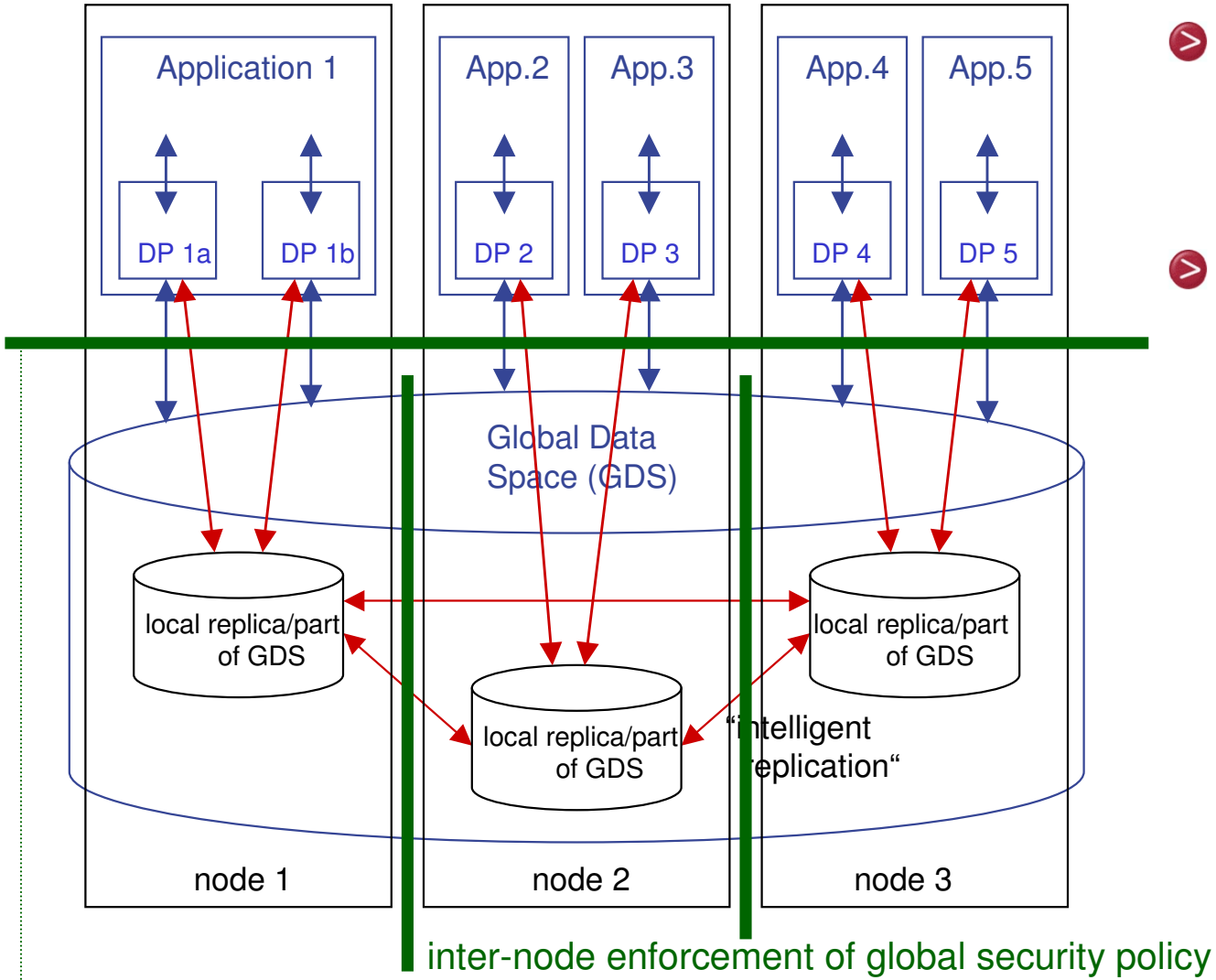
Simple Integrity Property: A subject can modify (have *write* access to) an object only if $I_{subject} \geq I_{object}$

Integrity *-Property: If a subject *s* has *read* access to an object *o* with integrity level I_o, *s* can have *write* access to an object *p* only if $I_o \geq I_p$

Security Model Activation

- The Bell-LaPadula *- Property can be deactivated for single applications (trusted applications, MLS applications)
- The Biba Integrity *- Property can be deactivated for single applications (trusted applications, MLS applications)

End-to-end Security / Inter-node Security



- End-to-end security = intra-node security + inter-node security
- Inter-node security includes
  - preservation of confidentiality and integrity of information communicated over unsecure networks
  - labeling of differently classified information
  - cryptographic of differently classified information
  - RED/BLACK separation at the node/network boundary by means of a dedicated network process

Summary / Q&A

- Most application environments for Data-centric Pub/Sub require the integration and support of proven security models (Bell-LaPadula etc.) by the information distribution backbone
- Data-centric way to build distributed application systems very suitable for the implementation of such models
- End-to-end security = intra-node security + inter-node security
- Proposed mapping of DDS entities to subjects and objects of established security models:
  - ClassifiedDataPartitions as objects (ClassifiedDataTopics as specialization)
  - Domain Participants as subjects
  - Applications as principals