
Access Control for Mobile Objects & Hosts

Jehan Wickramasuriya

jwickram@ics.uci.edu

Distributed Systems Middleware Group
Department of Information & Computer Science
University of California, Irvine

UNIVERSITY OF CALIFORNIA, IRVINE



Distributed Systems Middleware Group

Outline

- Overview of Research
 - Motivation
 - Customizable & Composable Security
 - View Management & Maintenance
 - Privacy & Access Control
-

Overview of Research

- Access Control for Middleware
 - Directory-enabled Access Control (*CompOSE/Q*)
 - Domain-based Access Control (DBAC)
 - Delegation & Revocation in the Presence of Mobility
 - Multilevel Access Control / View Management
-

Motivation

- Access control is an important problem in distributed environments.
 - Prevent service & content providers from obtaining unauthorized access to client data and resources.
 - Reduce complexity at firewalls and networks, supplement them with more robust, host-based security mechanisms.
 - Introduction of mobility further complicates traditional mechanisms used in distributed systems and forces us to re-evaluate how we view systems.
-

Customizable Security

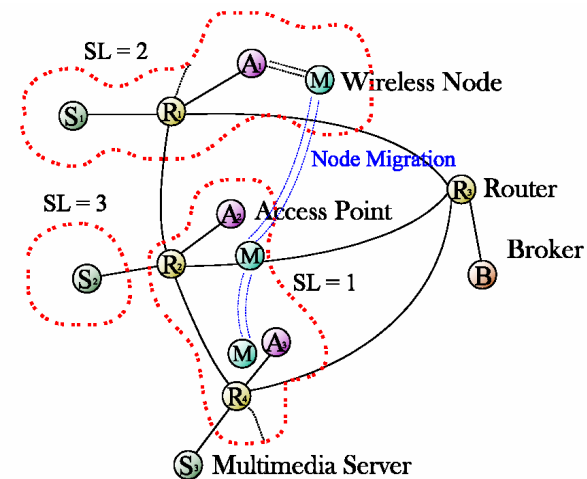
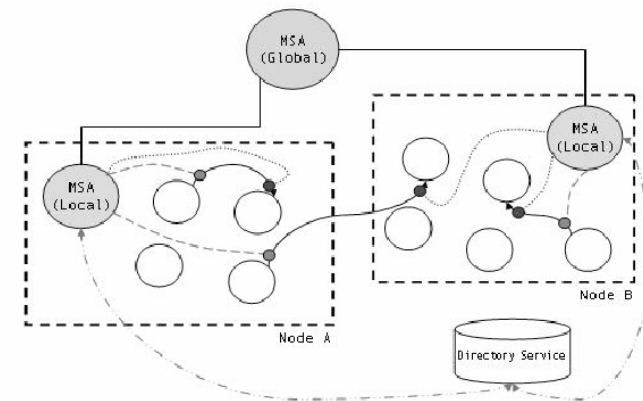
- Capability-based access control architecture
 - Customizable Security Mechanisms
 - Adapt security policies to fit user needs under changing system and network conditions
 - Resource constraints, network bandwidth
 - Dynamic switching of security protocols (key exchange, encryption), offline processing / parallelism of resource intensive security mechanisms
 - Tradeoffs for Secure QoS
 - Security as a QoS parameter; classify user requirements based on 'levels of security' and map these into a set of available security services.
 - Domain-based Security
 - Inter-domain security based on various security levels.
 - Protocols to define object mobility
-

Composable Security Services

- In a middleware framework where a number of services are operating concurrently, must ensure correctness & non-interference of the security architecture.
 - **Object migration & access control**
 - Delegation & Revocation in a mobile environment
 - Restrictive delegation, changing delegated access rights 'on the fly'
 - 'Real-time' revocation of access rights in a highly dynamic, mobile environment
 - Revoking cascaded delegations in a mobile environment
-

Domain-based Access Control (DBAC)

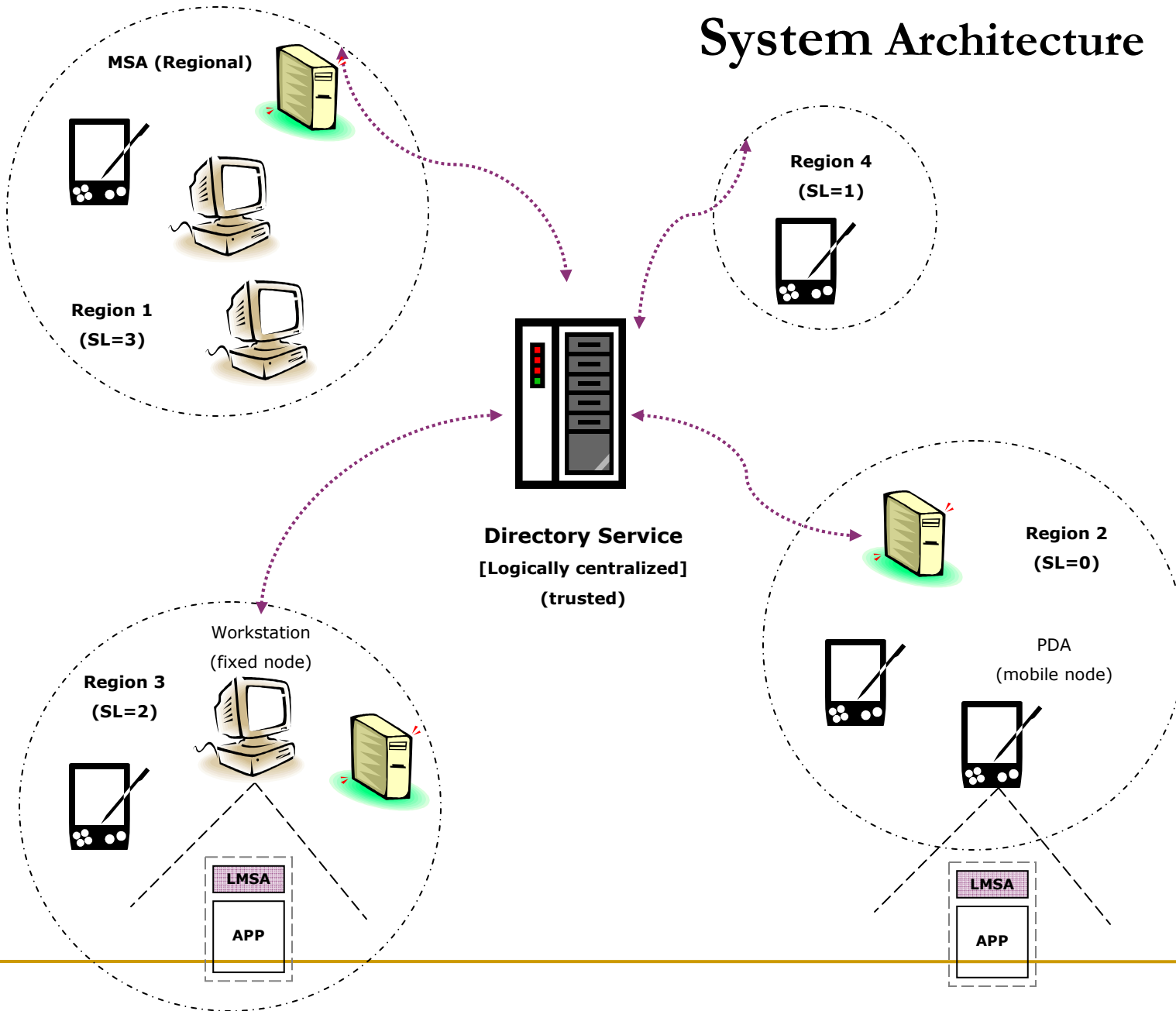
- We model domains as partitions of the network space, encapsulating nodes of varying security levels. These partitions effectively represent certain equivalence classes that allow us to control the granularity of the domain.
- The security level (SL) of a domain is represented by a tag, which is associated with both the objects and the domain itself. Application objects are created in initial domains and move to new ones which may require different credentials or more specifically – different security levels.
- By studying the interactions between various services and policies, objects moving between different domains should preserve a set of properties for availability, secrecy, integrity and semantic consistency.



View Management & Maintenance

- All entities represented as active objects (e.g. files). Hence can prohibit access to objects via manipulation of an objects' acquaintance list.
 - View of an object : representation of state of an object at some point in time. Total view of an object at the highest security level, which is represented in the DS (possible acquaintances, state).
 - Development of a view management algorithm, that deals with instances in which the view of an object can change during system operation (migration, dis(re)connection etc.).
-

System Architecture



Security Meta-actor Functionality

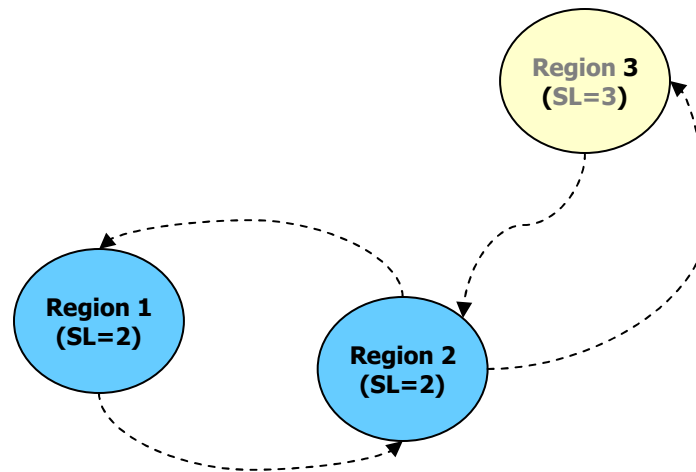
- Functionally, two types of security meta-actors:
 - Regional (RMSA)
 - Local (LMSA) – mobile or fixed nodes
 - Responsible for;
 - Encapsulation & interpretation of security policies
 - Mutual authentication of meta-level communication and authentication with directory services
 - Message processing
-

Example: Object Migration

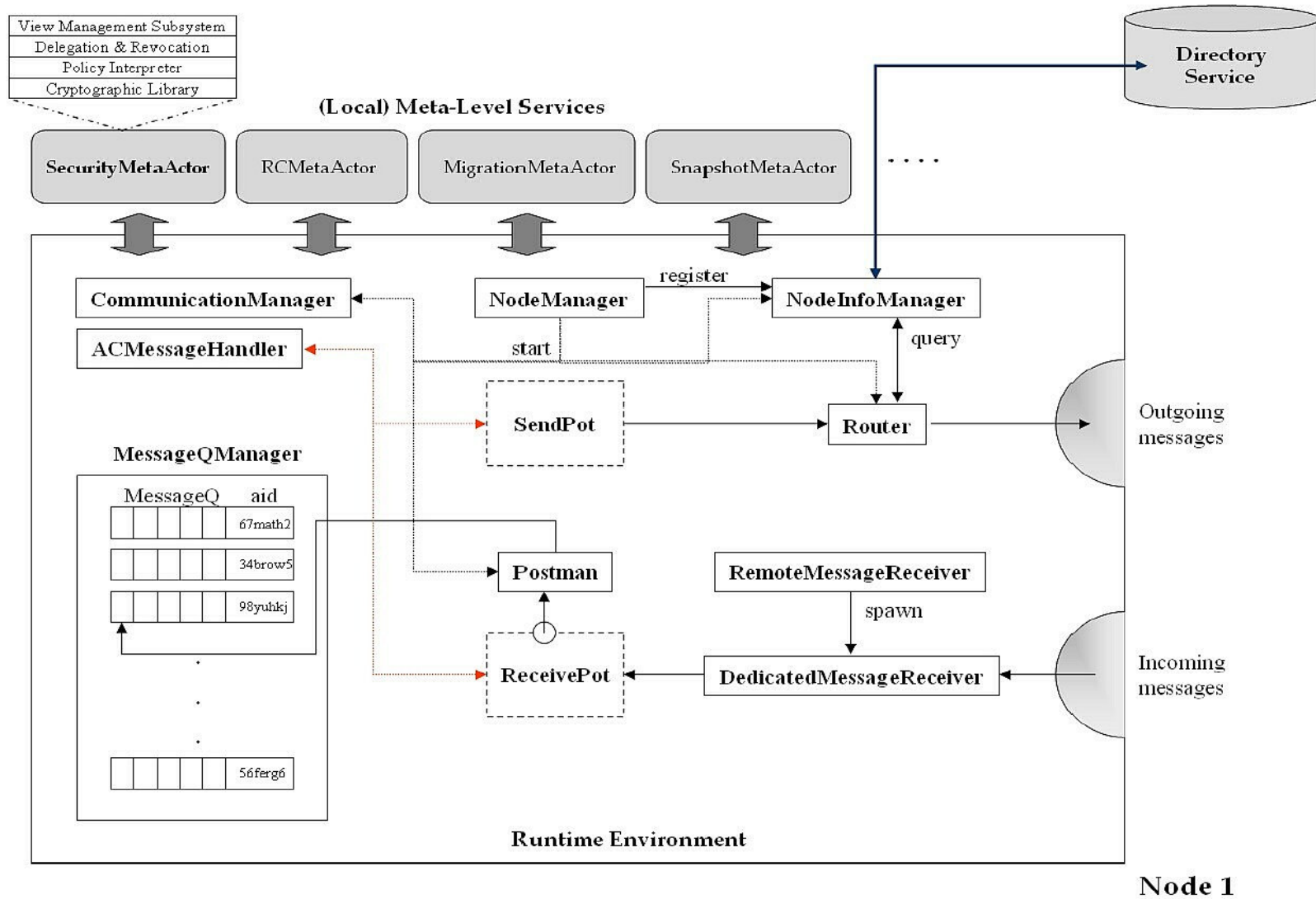
```
OBJECTMIGRATION()
1  for migration_request( $\alpha, v$ )
2  do if  $SL(\alpha) > SL(dom(v))$ 
3      then
4           $SL\_Reduce(\alpha, SL(dom(v)))$ ;
5  else if  $SL(\alpha) < SL(dom(v))$  AND
6           $(SL_{max}(\alpha) < SL(dom(n)))$ 
7      then
8           $SL\_Restore(\alpha, SL(dom(v)))$ ;
```

Example: View Approximation

- A view is associated with the current location (more specifically a domain) of an object. Assuming the underlying location management system uses a history-based profiling mechanism the protocol looks to identify common movement patterns and then approximate the view to prevent excessive computation of constantly switching state



Interaction of meta-level services with the *CompOSE/Q* runtime



Privacy & Access Control

- Access control mechanisms can be exploited in developing global privacy policies.
 - Example: A privacy policy may depend on the relationship between the entity requesting data, and what entities are described by that data. Granting rights to the data subject defines whether can access or even update its stored data.
 - Video-based media spaces are a prime candidate for privacy violations. Privacy preservation techniques are largely unexplored in today's solutions. Topic of ongoing research.
-