

SAS: A Secure Aglet Server

Presented by:

Dr. Yu (Cathy) Jiao

Computational Sciences and Engineering Division

Oak Ridge National Laboratory

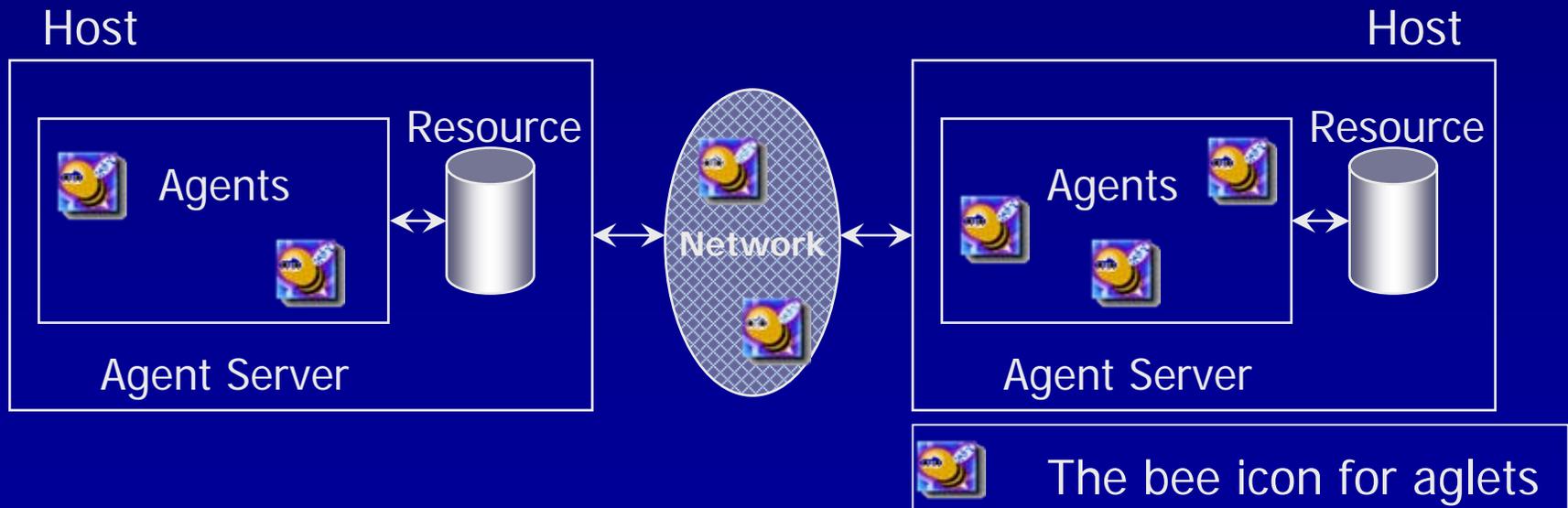
Oak Ridge, TN

Outline

- Introduction
- Background
- Security Vulnerabilities of the Aglet Server
- Secure Aglet Server (SAS)
- Privacy-Preserving Information Retrieval (PIR)
- Conclusion

Introduction

- Mobile Agents
 - What are they?
 - What's the buzz about?



Introduction

- Mobile Agents and Possible Commercial Applications
 - E-Commerce
 - Distributed IR -- Mobile Agent Mobile Data Access System (MAMDAS) [Jiao04]
- Limitations of Mobile Agents -- Security threats [Greenberg98]
 - Protect Hosts
 - Protect Agents

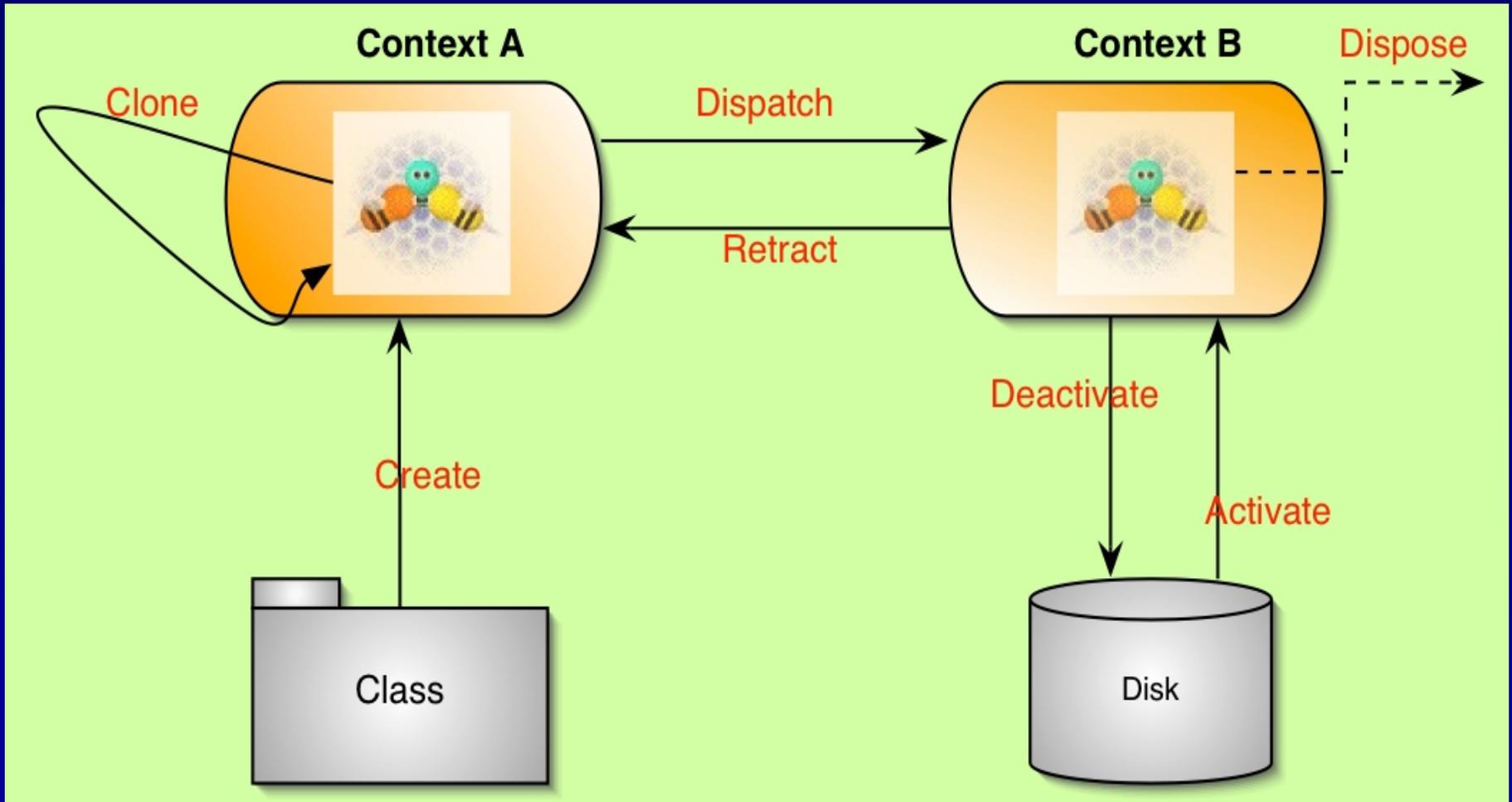
Motivation

- Agents can efficiently address numerous practical issues in mobile business applications, and thus there is a need to help foster their adoptions in such environments.
- The deployment of such applications is hindered by the security threats facing the paradigm.

Background

- Mobile Agent Security:
 - Host Protection [Greenberg98, Esparza03, Ono02, Tschudin99]
 - Code Signing
 - Access Control
 - Proof Carrying Code
 - Path Histories
 - Agent Protection [Greenberg98, Esparza03, Bierman02, Tschudin99]
 - Tracing
 - Obfuscation
 - Trusted Hardware
 - Limitation of current proposals
 - Mostly theoretical
 - No acclaimed solution to protect agents from hosts

Life Cycle of Aglet



Security Vulnerabilities of Aglet Server

- Communication vulnerability
 - Agents need secure communication to perform their most inherent function: mobility.
 - Aglet sever supports authentication of servers through a Challenge-Response scheme based on the Diffie-Hellman algorithm.
 - Can this prevent interception of agents during transmission between servers?

Security Vulnerabilities of Aglet Server

- Using *Dsniff*, we were able to prove that the communication channels of aglet server are not secure
 - Agents, in their serialized state, were easily intercepted during transmission.
 - The available Challenge-Response scheme only verify the identity of servers to prevent threats such as reflection attack, but does not serve as a prelude to encrypted communication.
- The Aglet framework cannot currently satisfy the requirement of agents to migrate exclusively to intended hosts; thus is unsuitable to support agent-based commercial applications.

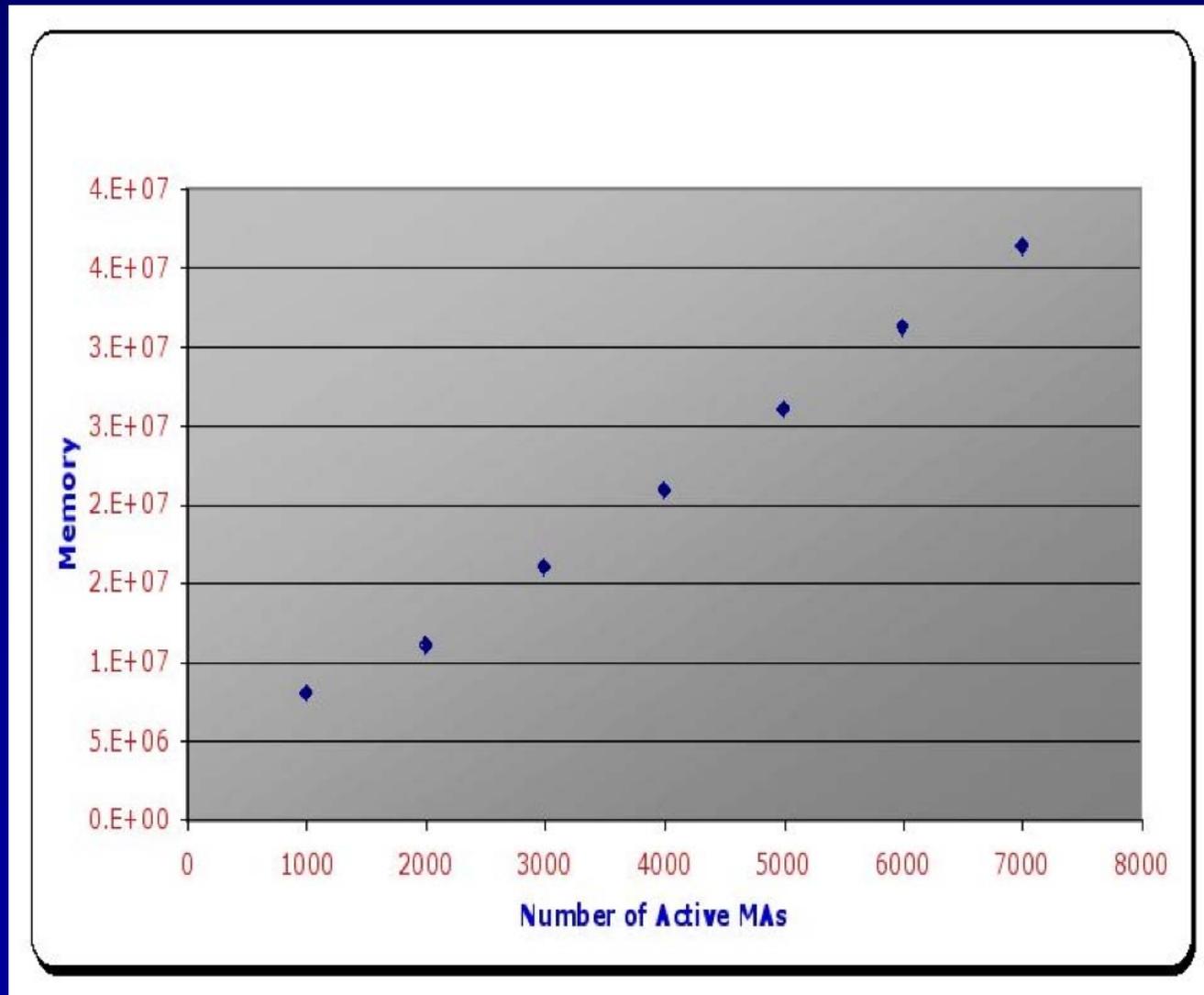
Security Vulnerabilities of Aglet Server

- Data Vulnerability
 - Protecting agents from malicious hosts requires more than secured communication.
 - Agents need to ensure the integrity of their data as they travel between hosts.
 - No acclaimed solutions exist to protect agents' data from being manipulated, thus aglet server has no mechanisms to address the issue.

Security Vulnerabilities of Aglet Server

- Resource Vulnerability
 - Aglet is Java-based and thus benefits from the language's sandboxing techniques.
 - Agents are however allowed a set of actions inherent to their lifecycle
 - Our experiments show that Aglets, through their normal lifecycle, can indeed generate a Denial Of Service (DoS) attack on a host through:
 - Repeated cloning
 - Creation and dispatching of Aglets to a target host
 - Activation and retraction of Aglets

Security Vulnerabilities of Aglet Server



Secure Aglet Server (SAS)

- To foster the development of agent-based commercial applications, SAS has been developed to provide:
 - Secured Communication
 - Integrity and reliability of agent's data
 - Controlled resource consumption of agents

Secure Aglet Server (SAS)

- Secured Communication
 - As shown earlier, agents could possibly be intercepted during transmission
 - SAS implements SSL at the communication layer and thus provides:
 - Encrypted communication
 - Client and server authentication
 - Mechanisms to support flexible security specifications from administrators

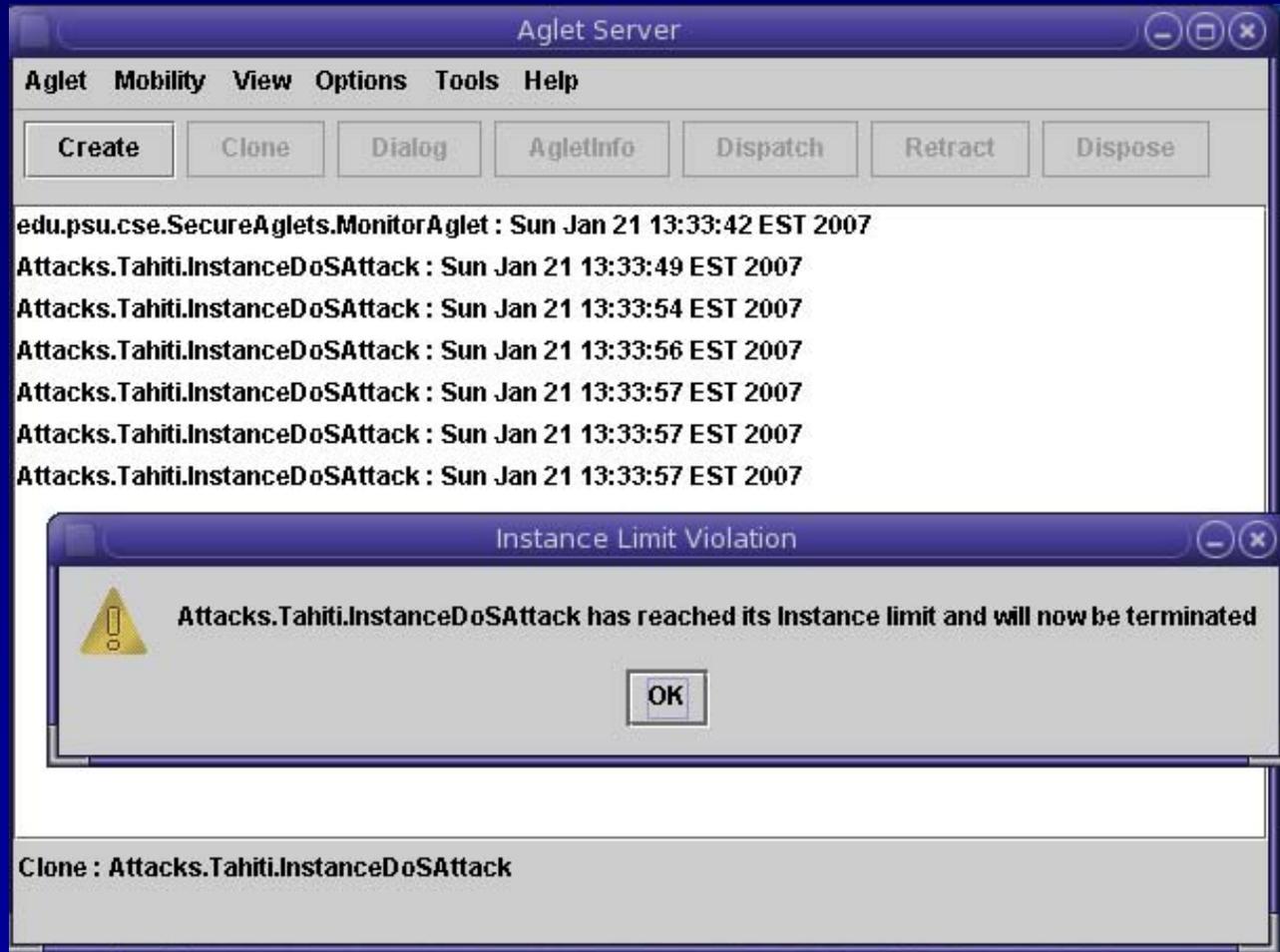
Secure Aglet Server (SAS)

- Secure Data
 - At the least, agents should be made aware of such occurrences whenever possible
 - SAS makes use of the Java Cryptography Extension
 - To implement the concept of Read-Only Data through provision of a Java Class library allowing
 - Detection of data tampering
 - Detection of active malicious hosts' in an agent's itinerary

Secure Aglet Server (SAS)

- Secure Resources
 - Preventing agents from overusing hosts' resources requires detection and reaction to resource violations
 - As such, SAS tracks the resources (number of instances) of an agent through the introduction of a MonitorAglet to
 - Manage the number of active instances of an Aglet
 - Negotiate resource requirement before travel

Secure Aglet Server (SAS)

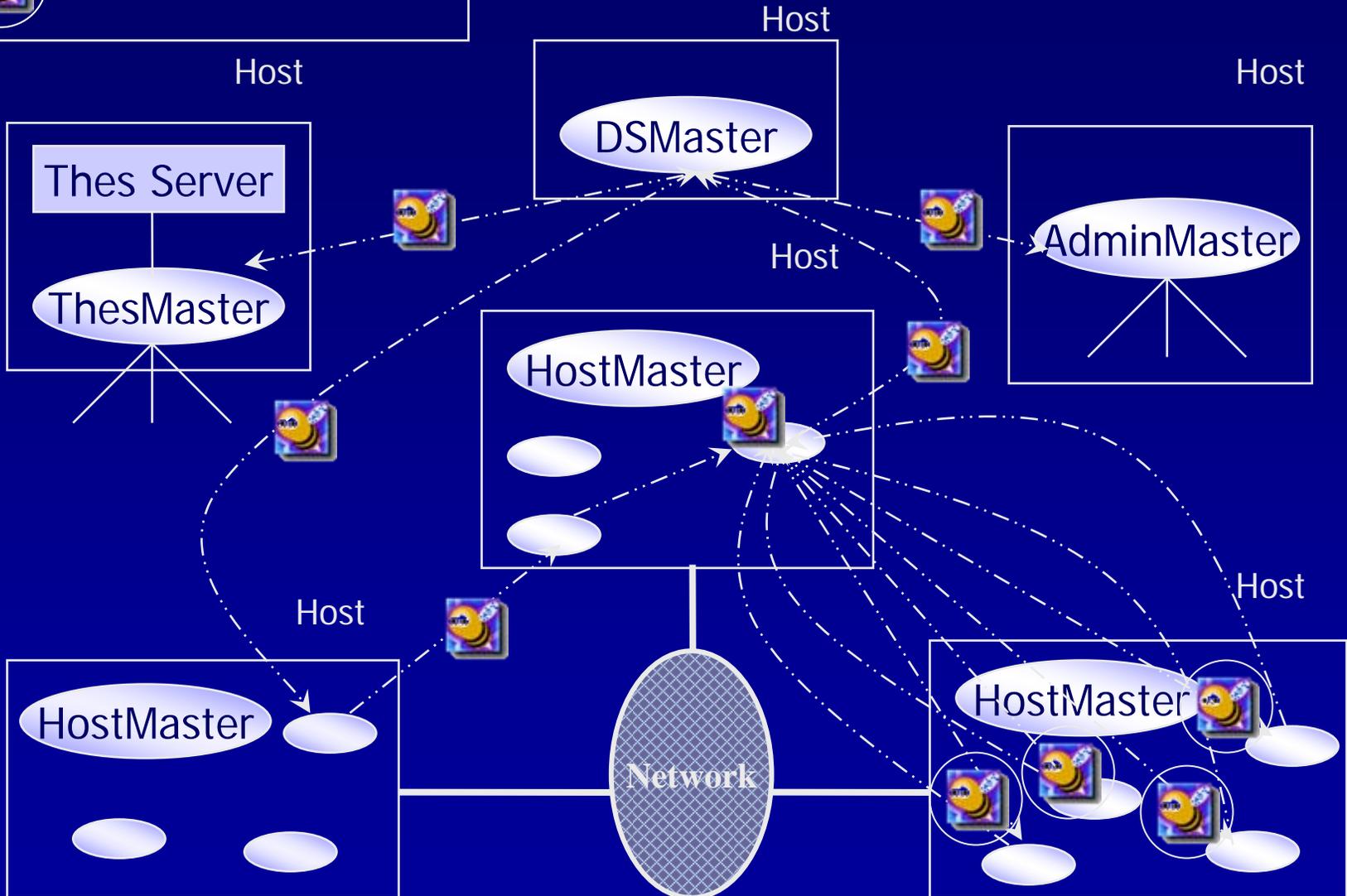


Detection of Instance Violation by MonitorAglet

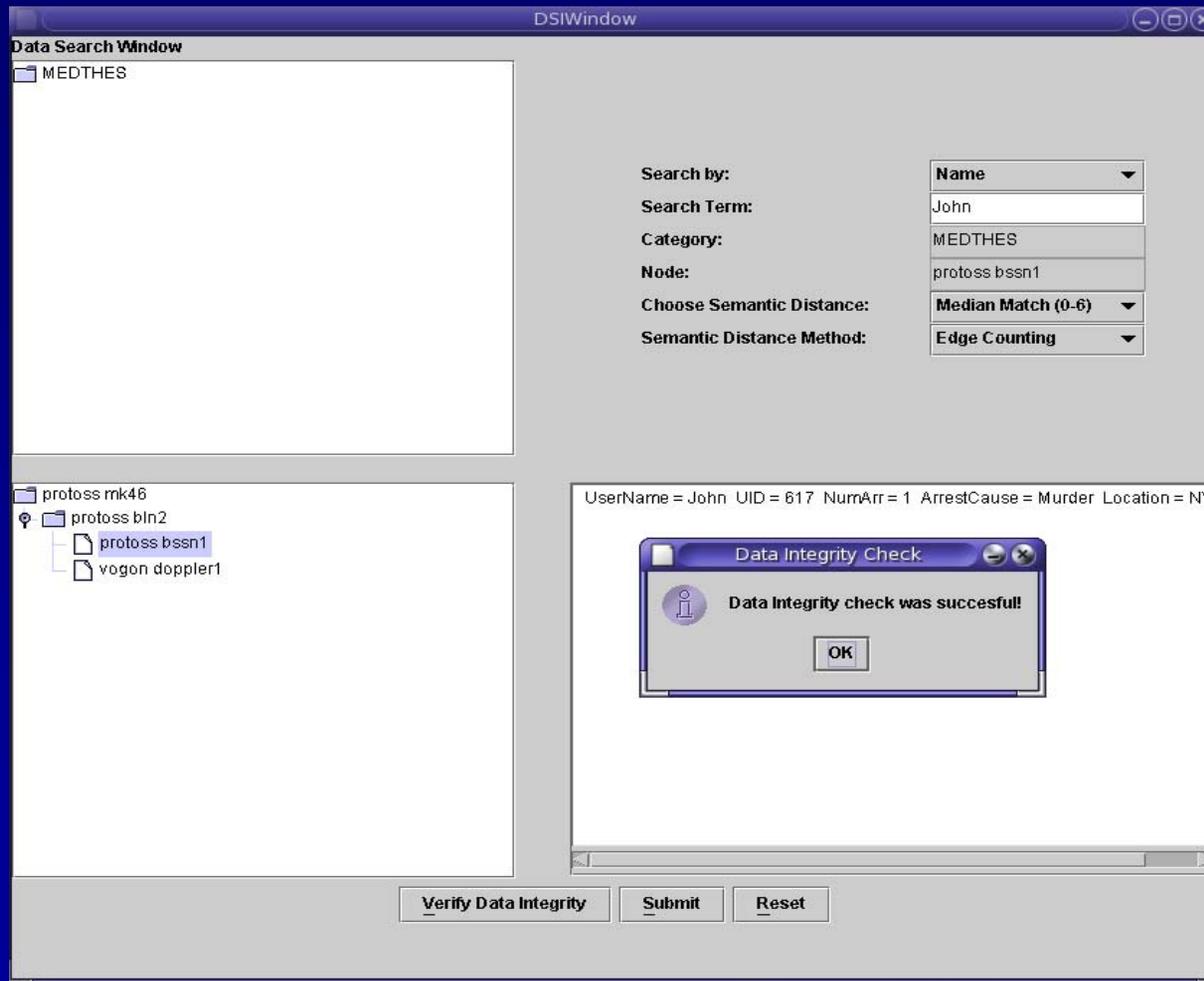
Privacy-Preserving Information Retriever (PIR)

- PIR is a prototyped agent-based application to highlight the contribution of our work.
- PIR is a system based on MAMDAS, designed to help companies reach a decision about a potential employee.
- The system emulates collection of data from various governmental and private sources that may come into play in hiring an individual; such as:
 - Arrest records
 - Previous salaries
 - Credit reports

PIR Overview



Privacy-Preserving Information Retriever (PIR)



Conclusion

- SAS has addressed the security shortcomings of aglet server in an attempt to help foster the emergence of mobile agents in commercial applications.
- SAS addresses agent security from a practical standpoint, and provides:
 - Secured communication
 - Detection of data tampering
 - Controlled resource consumption
- While SAS does provide a secure framework for agent-based application development, some issues remain to be tackled:
 - Agent security beyond local networks