Reliable Shared Record Storage By Using JavaME

R. Sivaraman [1] Dr. S. Sujatha[2] and R.Prabakaran[3]

[1] *Deputy Director, Centre for Convergence of Technologies, Anna University Tiruchirappalli, Tiruchirappalli-24, Tamil Nadu, India.*

[2,3]*Lecturer, Centre for Convergence of Technologies, Anna University Tiruchirappalli, Tiruchirappalli-24,Tamil Nadu, India.*

Abstract

Now-a-day's computational schema is running autonomous computing, which is involved in diverging number of database applications to mobility and facilities the users to achieve the usefulness of anywhere-anytime computing. JavaME is one of the leading technologies in handheld device. Sharing the data in JavaME opens the data to be accessible for all MIDLETs present on the device, hence creating a number of vulnerabilities to the confidential information that is intended to be shared with specific MIDLETs. This paper mainly focuses on providing security measures to overcome this problem and also to improve the performance by using multi-agent systems (MAS). The implementation uses the SAGE-Lite framework as a solution to this proposed work.

Keywords:

MIDLET, Multi-Agent, Record Store, SAGE-Lite.

1. Introduction

With the advancement in information and communication technologies, more and more features are being provided in all walks of life especially in mobile computing. Current research trend towards mobile computing emphasizes the need for distribution of data among various clients in wireless environment. This facilitates the users to achieve the usefulness of anywhere-anytime computing.

Persistence and distribution of data is crucial for small handheld devices as these devices are not as powerful as that of their counterpart – desktop machines. Keeping in view the constrained environment and limited resources on these devices a number of mechanisms are provided for data persistence on these devices. Java 2 Micro Edition (JavaME) is one of the leading technologies in handheld device applications and has a Record Management System (RMS), an Application Programming Interface (API) that provides persistent storage on local device.

RMS is a system for managing records in JavaME. A record is an individual data item. The records in RMS consist of variable length binary field. This is contrary to the typical Database Management System (DBMS). The validation and data consistency checks cannot be applied to the records in RMS. Sharing the data in RMS opens the data to be accessible for all MIDLETs present on the device, hence creating a number of vulnerabilities to the confidential information that is intended to be shared with specific MIDLETs.

To overcome the above mentioned problem layer of multi-agents system is introduced, which handles the whole data manipulation from inside or outside the device. An agent can perform desired action only on its MIDLET data. Exchange of information between to two MIDLETs takes place only through their agents. In order to make changes on different MIDLET data the agent of first MIDLET sends a command to the agent of second MIDLET, which then responds to the agent of first MIDLET. This helps in minimizing the vulnerabilities to the confidential information when the record store is said to be in shared mode.

2. Related Work

The growing trend to introduce the personal agents on mobile devices gives rise to the need of framework that should facilitate developers to implement agents on mobile devices. Multi-Agent Systems are used as core technologies in various applications from information retrieval to business process automation.

LEAP is a Lightweight Extensible Agent Platform; it enables the components of MAS to run on handheld device using JavaME or personal java. It does not have support for object persistence to ensure fault tolerance [5]

Micro FIPA-OS is an agent development toolkit. The system targets PDA devices that have sufficient resources to execute a personal Java compatible virtual machine. The dependence on personal Java limits its development to relatively powerful PDA's. Micro FIPA-OS allows the use of FIPA-OS components such as AMS and DF. FIPA-OS and Micro FIPA-OS use tasks and conversation as the basic metaphor for programming agents, and agent functionality. Since the task and conversation

Manuscript received July 5, 2009
Manuscript revised July 20, 2009

management introduce overhead and latencies entirely on the device, but it is recommended that only one agent be executed on each small device [4].

Grasshopper is based on the older MASIF (Mobile Agent System Interoperability Facility) specifications of OMG (Object Management Group). It was intended for J2SE but a personal Java version is also available. It concentrates on mobile agents and a stable extendible platform. Main drawbacks are lower scalability and a weak internal agent communication organization [5].

SAGE-Lite is a light-weight multi-agent platform having smallest foot-print and capable enough to run on mobile devices efficiently [2].Existing multi agent system frameworks (JADELEAP [5], Grasshopper [4], Micro FIPA-OS [4] etc.) do not provide robustness, context awareness and persistence [2]. SAGE-Lite platform is chosen for this work because it is stand alone framework for the lightweight devices and provides the features of robustness, fault tolerance, object persistence and context awareness [2].

3. Architecture Of JavaME Record Store Management System (RMS)

The architecture of JAVAME record management system (RMS) contains small footprints for storing and manipulating persistent data on local device.

3.1. Record Stores

A record store is chronological collection of records associated to a record store. RMS maintains unique numeric sequence number for each record in a record store known as Record ID. Here, the record store is recognized by the name of MIDLET suite, its vendor name and the record store name itself.

Time-stamp and version information is also maintained by record stores to enable applications to discover when it was last modified [1].

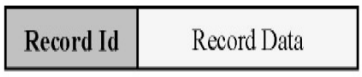

Fig 1: Internal Architecture of Record Store

3.2. Record Stores Sharing

Record stores are made accessible to all other MIDLETs using their shared property. The default property AUTHMODE_PRIVATE allows record store to be accessible only to the MIDLET suite that created it. Record store can be shared by changing the property to AUTHMODE_ANY. When the record stores are shared they can be writable or read-only. If the shared data is confidential, then it is not secure and can be read by all MIDLETs on the device. Also if the data is shared in read/write mode then it is more vulnerable to malicious MIDLETs on the device [3]. In the shared mode the MIDLETs from remote devices cannot access the shared data directly.

3.3. Shared Storage Vulnerabilities

Confidential information can be susceptible to an attack outside the RMS, e.g. it can be accessed and manipulated from device utilities (without using a MIDLET), which is a serious problem, e.g. the whole Record Store can be deleted by calling the deleteFile () method of the class RecordStoreFile [3]. This is shown below in the figure:

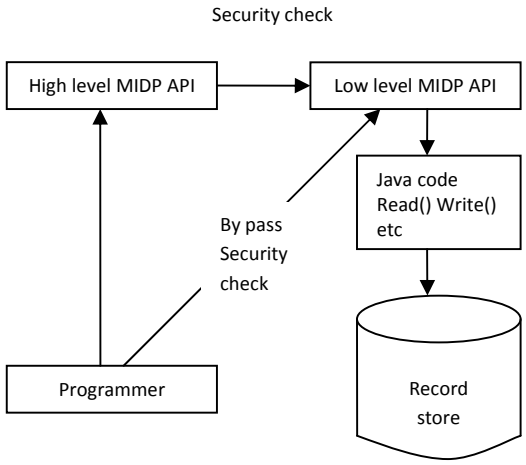

Fig 2 Shared record store vulnerabilities [3]

Non shared record stores can be accessed and modified by the MIDLETs creating them, but the shared record stores can be accessed from any MIDLET on the device. So MIDLETs cannot share their record stores with only a specific subset of MIDLETs. Shared record stores are vulnerable to any attack from outside the Record Management System (RMS) of JAVAME.

4. Multi-Agent System.

An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives. A set of agents that interacts in a common environment to achieve common or individual goals is called a multi-agent system (MAS). A multi-agent platform is a software

infrastructure used as an environment for deployment and execution of agents. SAGE-lite is one such multi-agent platform which is based on FIPA specification.

4.1. Architecture of SAGE-lite

Scalable fault tolerant Agent Grooming Environment (SAGE) is designed to provide distributed and decentralized architecture to achieve tolerance and scalability. The SAGE core architecture is shown in Figure 3. The main components are, AMS (Agent Management System) that manages the platform, DF (Directory Facilitator) which provides yellow pages service and MTS (Message Transport Service) for messages delivery in agent platforms. Also, Agents communicate by using Agent Communication Language (ACL) messages. VMA is an agent that offers a graphical interface to platform administration and monitoring.

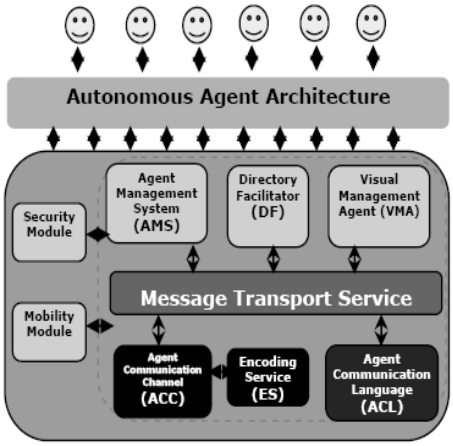

Fig 3 Architecture of SAGE-lite

The decentralized architecture of SAGE also embeds the capability of self-monitoring at the system level by allowing the agents to internally monitor themselves as well as externally monitor other agents.

5. Multi-Agent Based Record Sharing Architecture

In order to solve the problem mentioned in section 3, the concept of Multi-Agents System is used. In multi-agents system single task is divided among multiple agents on the same or different platforms.

Architecture on the basis of an existing FIPA compliant multi-agents system framework, namely SAGE-Lite [2] is provided. SAGE-Lite is a lightweight context aware multi-agents system, which senses the capabilities of the lightweight devices and reacts accordingly. Existing work on secure agents' communication in SAGE-Lite framework [2] gives an advantage of agent security, by restricting unauthorized agents to misuse the information. Keeping the record store property as AUTHMODE_PRIVATE makes it inaccessible to all other MIDLETs.

In the proposed technique actions (save, update, delete, search etc) on non-shared or private data are published as services of the agents and any other agent can access data through these services. The requesting agent can belong to same or different MIDLET on local or remote device. This is due to the fact that SAGE-Lite enables agents' communication on local or remote devices [2].

Following figure proposes the architecture for record store sharing without making it public to all other MIDLETs.

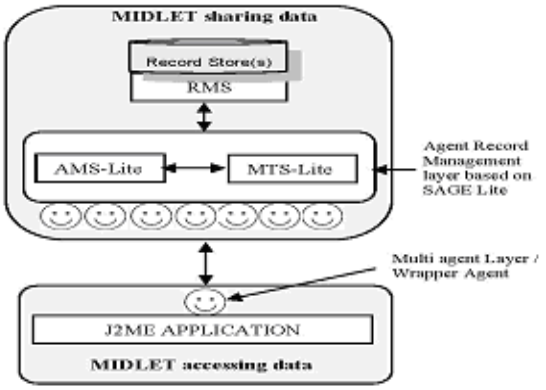

Fig 4 Record Store Sharing based on Multi-Agents

In the above figure Agent Record Management layer acts as a mediator between two MIDLETs sharing data with each other. The use of multi-agents system on this layer enhances the performance of system, as all the actions on data (read, write, search etc) are divided among agents. Two types of applications can access the data through agents: Legacy systems applications can access data through the wrapper agent and multi-agent based applications can access data through their agents by communicating with the Agent Record Management Layer.

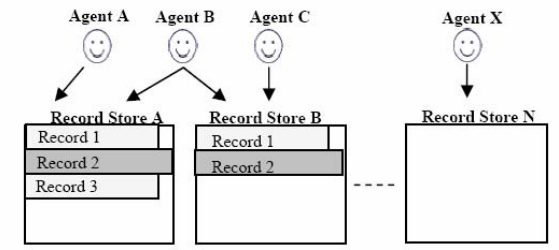

Fig 5 Agents accessing record stores

The above figure shows, how the agents on the Agent Management Layer work.
"Agent A" can perform desired actions on record store A.
"Agent B" can perform desired actions on Record stores A & B.
"Agent C" can perform actions on Record store B & so on.

6. Results

To demonstrate record store sharing using the multi-agents system, an application of booking a product is implemented based on the proposed architecture. This application keeps the record store private provides access to specific MIDLETs on same or remote devices through its agents. This application has three parts Advertisement, Search Item and Item Status.

Fig 6 Advertise an Item

Fig 7 View Advertised Items

Fig 8 View Status of Item

Fig 9 Search Item

Fig 10 Book Items

Fig 11 Item status New

6.1 Advertisement

List of items, price of the items, conditions, specifications such as make, model, color and other details will be displayed as advertisement. Figure 6 shows a form for uploading the item which is to be advertised. Figure 7 shows the list of items such as name of the mobile phone and the price details of each phone respectively.

6.2 Search Item

Status of the item is viewed by both private and public users. Here the specification of the item to be searched, price details, condition of the product, status of the product and so on will be displayed and Figure 9 shows the search item details.

6.3 Book Item

Here, any user can book an item from the list of advertised item and once an item is booked that item will be removed from the advertisement list i.e. from the record store and is shown in figure 10 and figure 11 respectively.

The above example shows the modification and sharing of data among specific MIDLETs.

7. Conclusion & Summary

The growing trend of mobile computing opens door to different issues regarding the data manipulation on small handheld devices. The problem of record store sharing in JAVAME is handled through a layer of multi-agents system. The proposed approach provides an environment that is independent of specific applications and can be used for different types of applications. JAVAME data with this approach can be shared among specific JAVAME applications through their agents. The use of multi-agents improves the performance of data manipulation on the local or remote device, enabling the applications to give restricted access to their confidential information.

Future work can be related to the concurrency control where multiple users are accessing the data simultaneously. The other area that needs to be explored is regarding the authorization and authentication of agents.

References

[1] Databases and MIDP, Part 1: Understanding the Record Management System http://developers.sun.com/mobility/midp/articles/databaserms/

[2] Hafiz Farooq et al: "Persistent Architecture for Context Aware Lightweight Multi Agent System". The Fifth International Joint Conference on Autonomous Agents & Multi-Agent Systems, Japan, 2006.

[3] M. Debbabi, M. Saleh, C. Talhi and S. Zhioua: Security Evaluation of JAVAME CLDC Embedded Java Platform, in Journal of Object Technology, vol. 5, no. 2, March–April 2006, pages 125–15.

[4] Sebastian Fischmeister, Giovanni Vigna, Richard A. Kemmerer, “Evaluating the Security of Three Java-Based Mobile Agent Systems”, Volume 2240/2001, 2001
[5] Anders Liljedahl, “Evaluation of Multi-Agent Platforms for Ubiquitous Computing”, JUNE 2004. SWEDEN

R.Sivaraman received B.E (EEE), M.E (CSE) from Annamalai University. Presently he is in the designation as Director for Center for Convergence ofTechnologies (CCT), Anna University Tiruchirappalli, Tiruchirappalli, Tamil Nadu, India. He is life member of ACEEE, IAENG.

Dr.S.Sujatha is Masters in Computer Applications from Bharathidasan University. She Obtained Ph.D degree from Periyar University, Tamil Nadu, India in the year 2008.
She is presently Lecturer and Research Member in Center for Convergence of Technologies, Anna University Tiruchirappalli. She has good number of National and International Publications in Computer Science. She is a life member of CSI, ACEEE, IAENG and IACSIT.

R.Prabakaran received B.E (E&I), M.Tech (Embedded Systems) from Sastra University, Thanjavur, Tamil Nadu, India in 2000 and 2005 respectively. Presently working as lecturer in Dept of EEE and also pursuing Ph.D and Research Team Member in Center for Convergence of Technologies (CCT), Anna University Tiruchirappalli, Tamil Nadu, India since 2007 to till date. He is also a life member of ACEEE and IAENG.