Economics-Inspired Self-Adaptive Security for Assets in Cloud

Simulator Description

We have developed a Java prototype simulation system that implements our framework based on an application scenario involving Cloud-based storage and VOIP services in university environments. The engineered solution manages the runtime security goals and sub-goals of the assets of multiple university employees via the selection and allocation of services and underlying computational resources. For illustration purposes we restrict the available services in the market to VoIP and Cloud-based storage services. Different services of the same type facilitate different security features and come at different prices. VoIP services vary in terms of encryption at transit, media encryption and anonymity mechanisms, where storage services vary in terms of mechanisms for encryption at rest and transit, encryption keys per file, authentication schemes and data segmentation techniques. To support these services our system is configured to support the provision of CPU, memory and disk space resources. Despite the small number of different types of services and resources supported by our simulation, it can be extended to support a wider variety of services/resources based on the needs of the users and the market itself.

Guide to run simulator

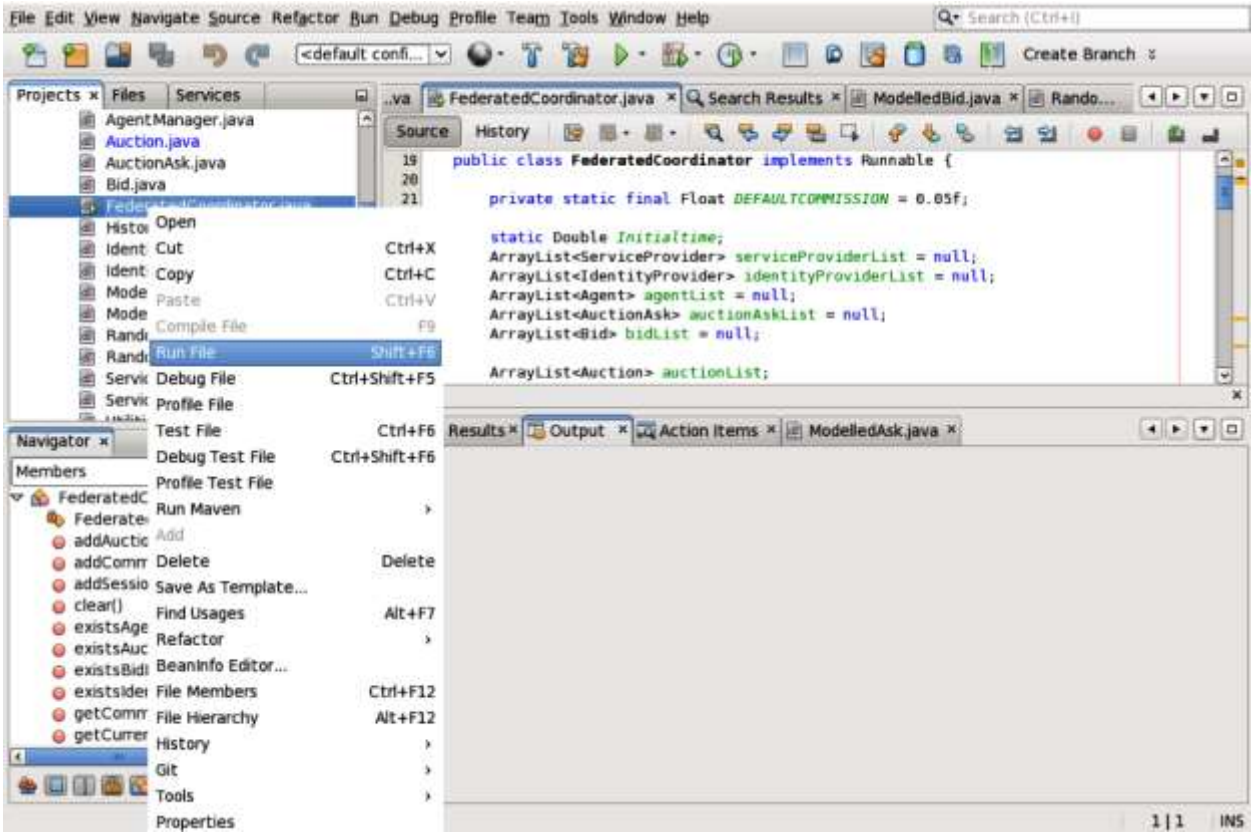
Download the source code from the repository https://github.com/GiannisT/UniMarketSimulation and save its contents in a single folder/directory. The application was developed as a Java Standalone, using the Maven artifact for java applications.

From Console

For a console execution change directory to C:\...\UniMarketSimulation\src\main\java\uk\ac\bham\simulator and execute the FederatedCoordinator.java file. To do so the user needs to use the following commands: javac *.java and java FederatedCoordinator

From NetBeans

Using NetBeans 7.4 as IDE to compile, build and execute the simulation is recommended. The current source code includes the project files for NetBeans. Choose the menu File, option Open Project, and then select the NetBeans project file named simulator. Open the file FederatedCoordinator, there will be a Java Class with the same name, just do right click in the file list, a popup menu will be shown, choose the Run File option or press the Alt+F6 to run the java code and enjoy. The full output can be found in file “output.txt”, which will be generated in the working directory, if you use the console it will be in that directory, if you use the NetBeans it will be in the project folder, the same folder of the pom file.


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default confi...
Create Branch
Projects
Files
Services
AgentManager.java
Auction.java
AuctionAsk.java
Bid.java
FederatedCoordinator.java
Search Results
ModelledBid.java
Source
History
public class FederatedCoordinator implements Runnable {
private static final Float DEFAULTCOMMISSION = 0.05f;
static Double Initialtime;
ArrayList<ServiceProvider> serviceProviderList = null;
ArrayList<IdentityProvider> identityProviderList = null;
ArrayList<Agent> agentList = null;
ArrayList<AuctionAsk> auctionAskList = null;
ArrayList<Bid> bidList = null;
ArrayList<Auction> auctionList;
Open
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Compile File F9
Run File Shift+F6
Debug File Ctrl+Shift+F5
Profile File
Test File Ctrl+F6
Debug Test File Ctrl+Shift+F6
Profile Test File
Run Maven
Add
Delete Delete
Save As Template...
Find Usages Alt+F7
Refactor
BeanInfo Editor...
File Members Ctrl+F12
File Hierarchy Alt+F12
History
Git
Tools
Properties
Navigator
Members
clear()
Results
Output
Action Items
ModelledAsk.java
1 | 1
INS