SECURE

Secure Environments for Collaboration
among Ubiquitous Roaming Entities

SECURE Applications Scenarios

Giovanna Di Marzo Serugendo
University of Geneva

Outline

- Objectives
- Applications Scenarios
- Taxonomy
- Lana Programming Model

Objectives

- Validation of the SECURE trust-based approach to security
  - Identification of application scenarios
    - Wide-ranging (domains, technologies)
    - Trust/Risk issues
  - Identification of a mobile agent platform
    - Agent paradigm fits into global computing infrastructure
    - Implementation of scenarios
    - Simulation of mobility of users and devices
  - Instantiation of the security framework in the platform
    - APIs of the SECURE models on top of the agent platform
    - Implementation of scenarios using the APIs

Selection Criteria

- Applications are:
  - Targeted towards global computing
  - Possess intricate trust relations involved
    - I.e., not simply a user trusting a super-user à la Unix
  - Wide-ranging:
    - Application Domains (Business, Education, Health, ...)
    - Technologies (PDAs, Agents, Web, Pervasive, ...)
    - Trust-Risk Issues (Recommendations, Risk, Trust, ...)

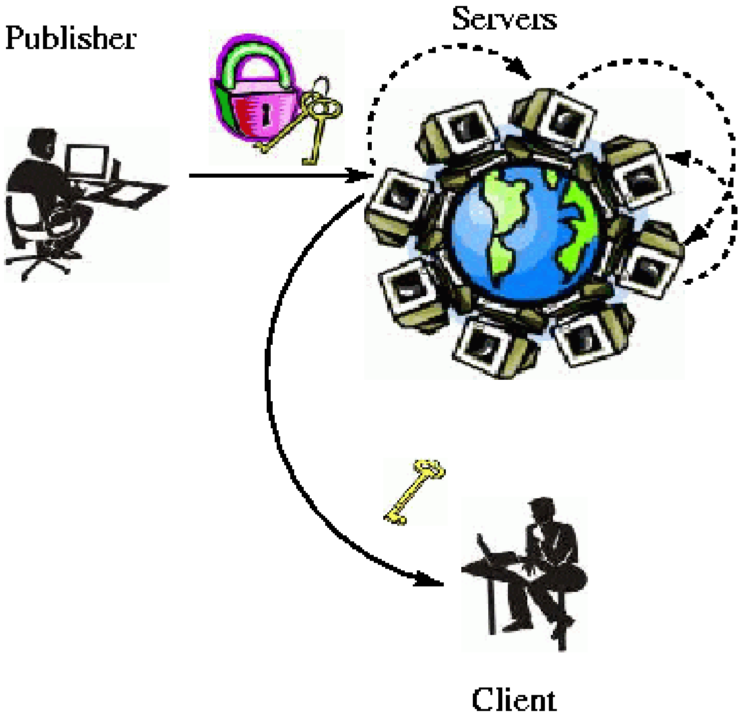
- Applications will be implemented on top of a mobile agent platform: Lana
  - Java extension
  - Designed for Global Computing environments
    - Supports devices autonomy
    - Secure information access

Examples

- PDA networking
  - Peer-to-peer information exchange
  - Micro-payments
- Other *ad hoc* networking
  - Cars on the road exchanging traffic information
- Deploying work on fixed network infrastructure
  - Sending a search agent to search the Web

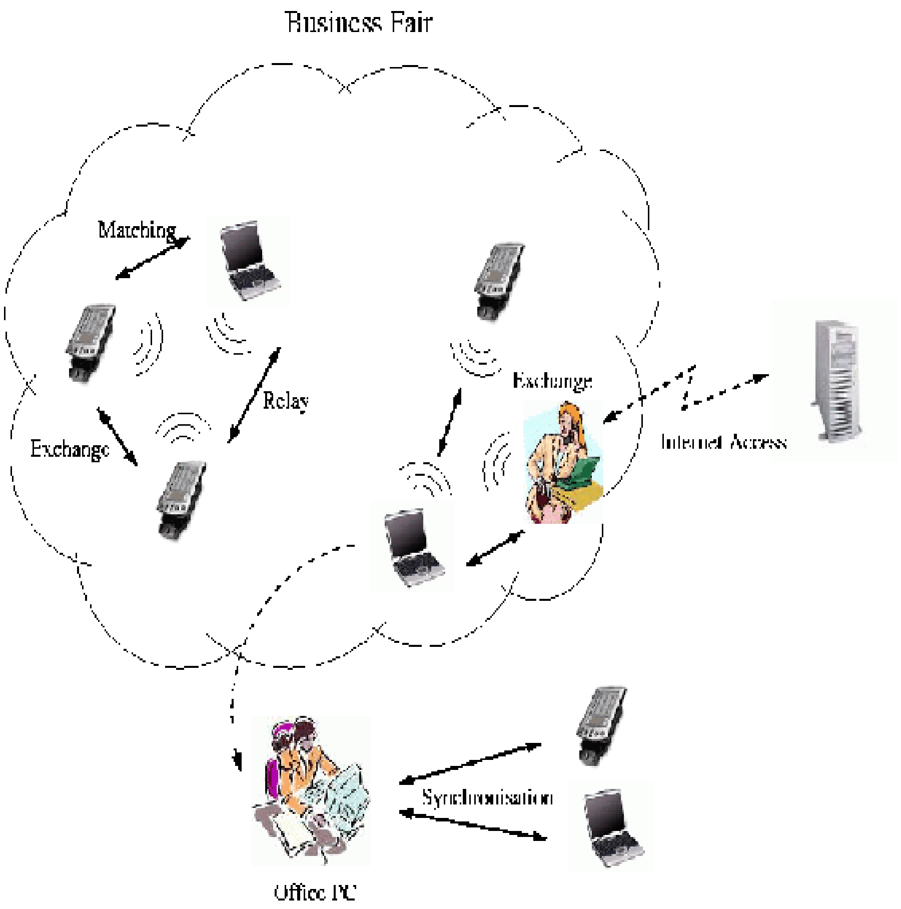
P2P Based Distributed File System

- Overview
  - Using peer-to-peer PC network to store backups
  - e.g., Chord file system
- Principals
  - Publishers, clients, servers, programs
- Trust/Risk
  - Loss of file backups
  - Corruption of files



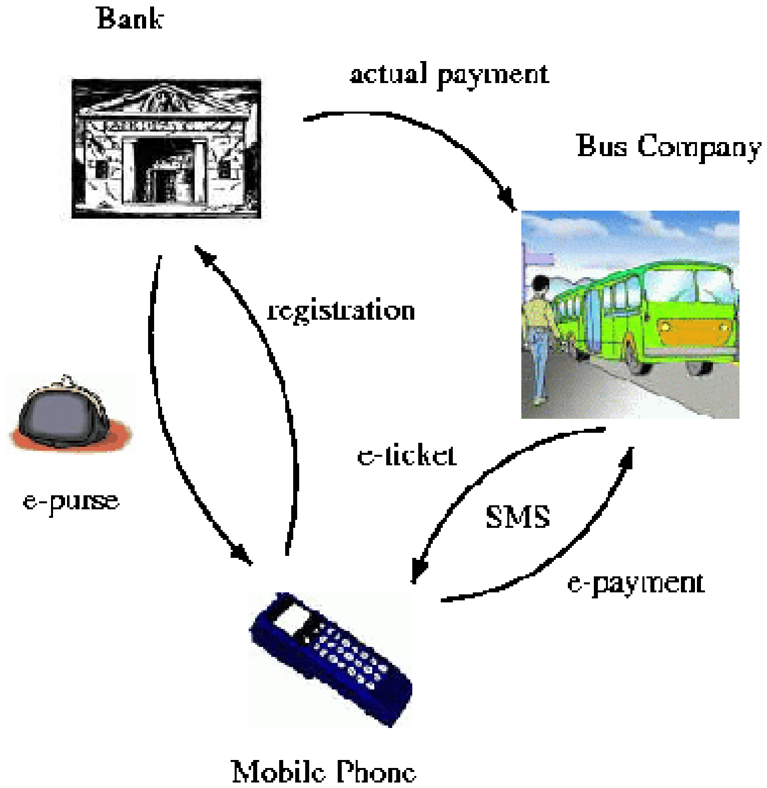
PDA-based spontaneous networks

- Business Fair
  - PDAs based recognition and information exchanges
- Principals
  - Visitors, PDAs
- Trust/Risk
  - Cheating Visitor
  - Corruption of sensible data



E-Purses on Mobile Devices

- E-Purses
  - Micro-payments
  - Mobile telephones
  - Bluetooth devices allow free data exchange
- Principals
  - User, bank, bus company, phone
- Trust/Risk
  - Loss of e-purse
  - False e-money



… still other scenarios

- Personalised Web Portal
  - Customer query solved through an agent backbone
  - Agent is trusted (running on backbone, delivery of result)
- Distributed Spam Filtering
  - E-mail filtering
    - Spam rejected on the basis of e-mail header (IP address, from, subject)
    - Reject Spam mails on the basis of trust
- Smart Spaces
  - Smart University Campus: delivery of urgent messages, availability of rooms, student recognition
  - Trust among students, and staff
- Collaborative Gaming
  - Blackjack game (through PDAs)
  - Trust in players

Use Cases

- Ad Hoc Network Routing
- Restaurant
- Coffee Machines
- Distributed Post-its
- On-line Auctions
- Car Rental
- Car Obstacle Avoidance

- Medical Records
- News Article
- Amazon
- P2P
- Grid Computing

Taxonomy

- Applications domains
  - Business, Education, Health, Science, Information Sharing, Entertainment, Network
- Technologies
  - PDAs, Virtual Spaces, Decentralised, Agents, Web, Pervasive
- Trust-Risk Issues
  - Recommendations and reputation, Observation, Risk, Benefit, Trust

Lana Programming Model (1)

- Extension to Java
  - Object-oriented, (classes, single inheritance, interfaces, packages)
- Designed for Global Computing environments
  - Multi-programmed language
    - Programs are mono-threaded, but several programs run simultaneously
  - Supports device autonomy

Lana Programming Model (2)

- Secure information access
  - programs are hierarchically organised
    - root=platform, children=programs
  - communication:
    - Locally: parent gives authorisation to communicate (among brothers)
    - Remotely: communication is forbidden (default)
      Parent can activate method permit to enable communication with any other program

Lana Programming Model (3)

- Unit of accounting
  - An object belongs to only one program
- Unit of mobility
  - A program moves with all contained objects
    - No shared objects!
    - Copies of objects can be transferred, but no sharing of references
- Unit of protection
  - Each method call on a program is verified by a security policy

Lana Programming Model (4)

- Asynchronous Method Calls
  - Among different programs
  - Avoid dependencies among programs
- Events
  - Returned values of method calls
  - Security Violation
  - Target Moved
  - Events are locked by keys

Lana Programming Model (4)

- Keys
  - Unique
    - Platform automatically generates new unique keys
  - Fixed
    - Several programs can generate the same key
    - Allows transfer of object copies through the use of a common key (cf message board)
- Message Board
  - Exchange of copies of objects
  - Objects are locked with keys

Status and Future Work

- Implementation status
  - Most scenarios are under implementation by SECURE partners
- Short Term
  - Trust and Risk description of some scenarios
  - Implementation in Lana of the E-purse scenario, integrating trust calculation and risk assessment
- Medium Term
  - Instantiation of the security framework in Lana
  - APIs for trust formation, trust calculation, risk assessment, trust-based access control