Tcl Agent :

A flexible and secure mobile-agent system

Paper by Robert S. Gray
Dartmouth College
Presented by Vipul Sawhney
University of Pennsylvania

Presentation Overview

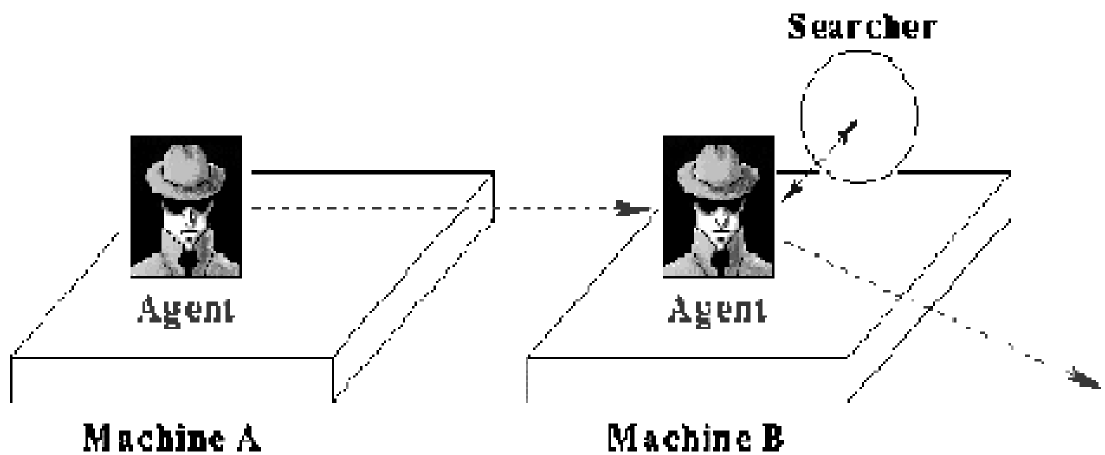
- Introduction
- Agent Tcl Architecture
- Tcl and Agent Tcl
- Security in Agent Tcl
- Applications
- Future Directions and Conclusion

Introduction

- An *information agent* is charged with the task of managing all or a portion of a user’s information space.
- The resources in this space are distributed across a network and can contain tremendous quantities of data.

What is a Mobile Agent ?

A mobile agent is an executing program that can migrate from machine to machine in a heterogeneous network under its own control.



Here an agent has migrated to interact with a search engine an will migrate again to bring the results back to its owner.

A Mobile Agent Can…

- Suspend execution at any point, transport its code and state to another machine, and resume execution on the new machine.
- An agent can access the resource locally and eliminate the network transfer of all intermediate data.
- Move the programmer away from the rigid client-server model to the more flexible peer-peer model.
    - Programs communicate as peers
    - Act as either clients or servers depending on their current needs

Introducing Agent Tcl

- Mobile agent system developed at Dartmouth College.
- Uses a flexible scripting language Tcl as its main language but provides a framework for incorporating addiotional languages.
- Provides migration and communication primitives that don’t require the programmer to explicitly capture state information and hides actual transport mechanisms

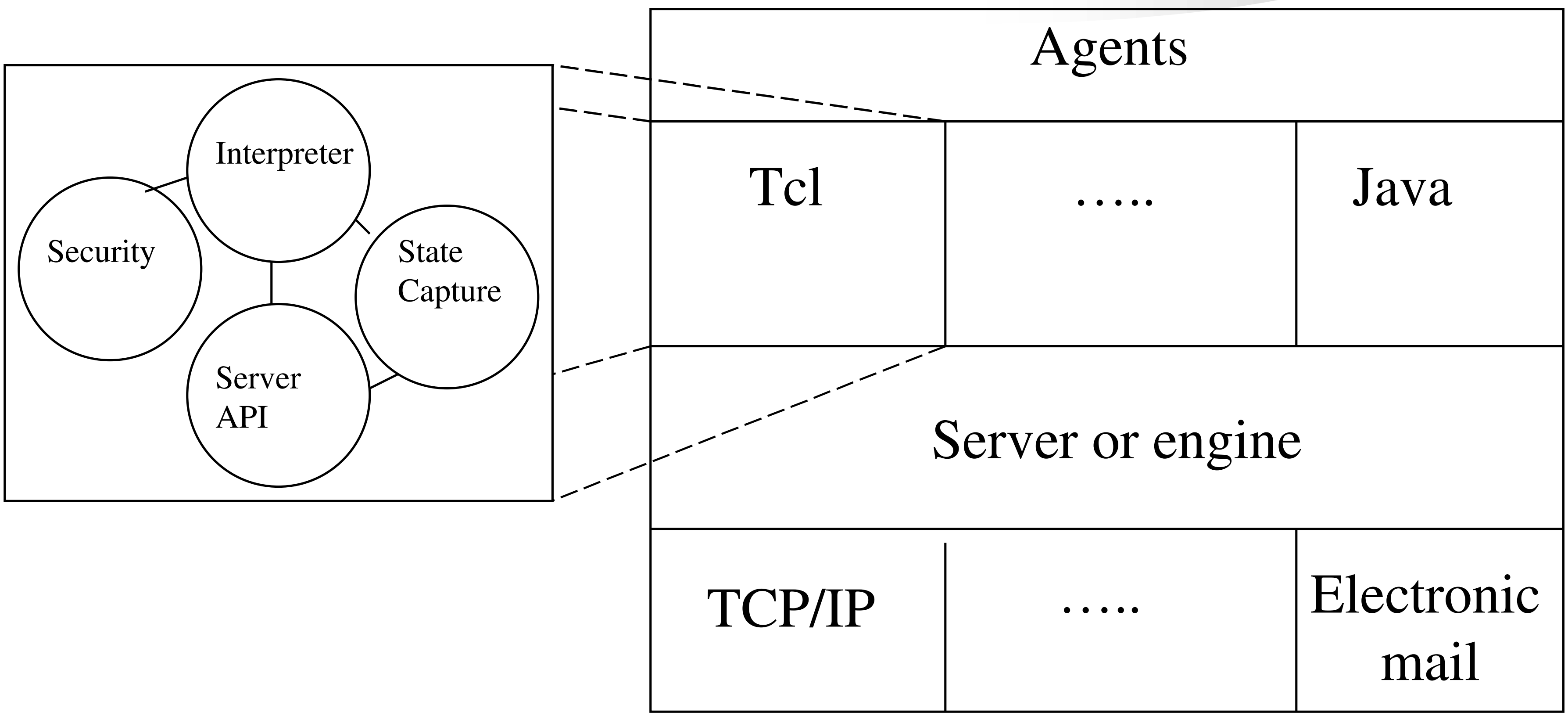
Agent Tcl continued…

- Uses Safe Tcl security model to protect a machine from a malicious agent and agents from each other.
- Allows agents
  - To migrate from machine to machine *or* remain stationary and access resources from across the network.
  - To create child agents to perform subtasks.
  - To communicate with other agents on the local and remote machines.
- Intended as a general environment for distributed applications.

The Architecture

- 4 main goals
  - Reduce migration to a single instruction.
  - Provide transparent communication among agents.
  - Support multiple languages and transport mechanisms.
  - Provide effective security in the uncertain world of the Internet.

Architecture continued…



Levels in Architecture

- 4 Levels to architecture
  - API
  - Server
  - Interpreter
  - Agents

Server Level

- Server performs following tasks:
  - *Status* - keeps track of the agents on machines
  - *Migration* - accepts each incoming agent, authenticates identity of its owner, passes to appropriate interpreter
  - *Communication*- provides a hierarchical namespace for agents and allows agents to send messages to each other within this name space
  - *Nonvolatile Store* - provides access to a nonvolatile store so agents can back up their internal state as desired.

Interpreter Level

- One interpreter for each available language
- Four components to interpreter
  - – interpreter
  - – security module
  - – state module
    - *captureState*
    - *restoreState*
  - – API

Tcl and Agent Tcl

- Current implementation does not provide event messages or the nonvolatile store
- Has a single language (Tcl)
- Single transport mechanism (TCP/IP)
- Incoming agents are authenticated using PGP

Tcl

- High level scripting language
- Advantages as a mobile agent language
  - Easy to learn
  - Interpreted thus highly portable and easy to make secure
  - Can be embedded in other applications
  - Can be extended with user-defined commands
- Disadvantages of Tcl
  - Interpreted - thus much slower than machine language
  - No facilities for capturing the complete internal state of an executing script
    - This was modified by adding to the Tcl core

Agent Tcl

- Two components
  - Server
    - Runs at network site
    - Implemented as two cooperating processes
      - One watching the network
      - Other maintaining a table of running agents
  - Modified version of Tcl 7.5 and a Tcl extension
    - Provides explicit stack and state-capture routines
    - Extension provides commands for an agent to migrate, communicate, and create child agents.

How does it work ?

- Agent simply a Tcl script running on top of a modified version of Tcl 7.5
  - **agent_begin -** register with server and obtain a name in flat namespace
  - **agent_submit** - used to create a child agent on a particular machine
  - **agent_jump** - migrates an agent to a particular machine
  - **agent_send, agent_receive** - to send and receive messages

How does it work, continued …

- **agent_meet -** used by source agent to send a connection request to the destination agent
- **agent_accept** - used by destination agent to receive the connection request and send wither and acceptance or rejection.
  - Acceptance includes sending a TCP/IP port number to which the source agent connects

Agent Tcl and Security

- Main Problems are to
  - *Protect the machine*
  - *Protect other agents*
  - *Protect the agent*
  - *Protect a group of machines*

Authentication

- Based on PGP (Pretty good Privacy)
  - Encrypts a file using the IDEA private key algorithm and random private key
  - Encrypts the private key using RSA and public key
  - Sends encrypted key and file to recipient
- Initiated by the **agent_begin** command

Authorization and Enforcement

- Resources divided into 2 types
  - *Indirect* resources
    - Only accessed through another agent
  - *Builtin* resources
    - Directly accessible through language primitives
    - Security maintained using Safe Tcl and
    - A set of *resource manager* agents
- Safe Tcl creates trusted and untrusted interpreter

Applications

- Who Agent
  - Logs onto a Unix machine and executes Unix command who
  - Sends resulting list back to its parent
- Medical Agent
  - Retrieves distributed medical records
- Text-Retrieval Agent
  - Searches distributed collections of text documents

Conclusion

- Future work includes incorporating Java into the architecture
- Agent Tcl
  - Secure mobile agent system which gains flexibility and simplicity from use of hih-level scripting language Tcl