ABSTRACT

Within this chapter, we present the requirements and a possible conception of a framework, which provides a platform and starting point for the development of mobile agents in peer-to-peer networks. Peer-to-peer networks like Kazaa, Gnutella, and so forth have reached a widespread use on the Internet. By deploying mobile agents that can travel between network nodes to a large P2P network, one could embrace the peer-to-peer technology and use it for all kinds of services like making network traffic anonymous, distributing storage of documents, replicating contents of heavily accessed Internet sites, trading of information, and so forth. For many of these things there are solutions available, but by using a common framework and moving the logic into the agents, there is the opportunity to access all kinds of information through a common API, which guarantees extensibility and widespread use.

MOBILE AGENTS

A mobile agent is software that acts autonomously and can travel from one computing device to another (Di Stefano & Santoro, 2002). Essentially, an agent should act on behalf of its user and may complete tasks on nodes accessible to him. Agents are mobile, if their code can be transferred to other computers. The transfer includes the agent’s state as well so that the agent resumes execution at the same state. There are several approaches and solutions for developing mobile agents available (W3C, 1996), but most of these are focusing on a very special kind of problem and none has reached widespread use in day-to-day applications.
This is remarkable because there were many who hoped that the use of agents would lead to a shift in programming paradigms. Despite the pending breakthrough, many possible scenarios, useful applications, and benefits are being discussed that hopefully will be realized in the future (e.g., see Chess, Harrison, & Kershenbaum, 1995, and Lange & Oshima, 1999).

To help mobile agents to become a widespread technology, we want to develop a common platform and framework, which makes the development of mobile agents as simple as possible while integrating a standard level of security into the system to allow normal users to maintain their computers’ integrity. The agents shall operate on a peer-to-peer network and contain the logic of the network services. Thereby, the network can be extended by the simple means of deploying a new agent into the network. No software updates at the already deployed nodes are necessary (see Lübke & Marx Gómez, 2003).

**REQUIREMENTS FOR THE FRAMEWORK**

For being successful and widely deployed by all kind of users, the peer-to-peer network has to provide an infrastructure and a set of standards that allows:

- Easy deployment of mobile agents (auto-generated agents/code)
- The agents’ owners to be identified by the system and the users
- The easy yet safe use of individual security policies for limiting the power of locally run agents

The given requirements are mainly motivated by the constraint of being operated by ordinary users. Since they are not developers, programming tasks must be hidden and technical details must be hidden by the application. Furthermore, there are purely technically motivated requirements, allowing normal operation of mobile agents in peer-to-peer networks. The system must therefore also allow:

- Agents to be moved or copied between network nodes
- Agents to access contents of the node they are running on based on the security policies in effect
- The announcement and management of network nodes and their capabilities
- Agents to communicate with each other as long as the local security policies allow this

These requirements lead to the following decisions, which are outlined below.

**Network Infrastructure**

Relying on the Internet, the peer-to-peer network has to be based on the transmission control protocol/internet protocol (TCP/IP) protocol suite (Socolofsky & Kale, 1991). Because the agents’ code and state has to be transferred intact, TCP connections will be used and not UDP datagrams, since TCP guarantees packets will be transmitted correctly by using checksums and acknowledgements. A container service, which manages all agent movements and the execution of mobile code, should bind to a specific port and wait for incoming connections. When this service is started, it has to locate neighbouring peers, announce its presence, and connect to neighbouring nodes.

The retrieval of these neighbour-lists are one of the problems peer-to-peer networks face because this is one point that is not easily done without centralized resources like a node-list.

The peer-to-peer client should first look if any other peer is available within the local area network. For example, this could be done by using the service location protocol (SLPv2) (Guttmann, Perkins, Veizades, & Day, 1999) or local