DirectDemocracyP2P

— Decentralized Deliberative Petition Drives —

Marius C. Silaghi[1], Khalid Alhamed[1], Osamah Dhannoon[1], Song Qin[1], Rahul Vishen[1], Ryan Knowles[1]
Ihsan Hussien[1], Yi Yang[1], Toshihiro Matsui[2], Makoto Yokoo[3], Katsutoshi Hirayama[4]

DirectDemocracyP2P[5] is an open source platform developed in JAVA and offering peer-to-peer and mobile ad hoc wireless communication capabilities. The platform offers an API supporting plugins, beside its main application: deliberative petition drives (aka citizens' initiatives with integrated argumentation) [1]. An authentication-by-reputation technique based on digital signatures and peer review [2], [3] is integrated into the platform via this main application. Each peer manages independently its database of items of interest. The items of interest are encapsulated as self-contained pieces of information and uniquely identifiable using a system of global identifiers (GIDs). Each GID consists of a combination of public keys with creation dates, or digest values. Communication is based on a combination of push and pull mechanisms. [1]

In the main application, deliberations develop around *motions* (concept from Robert's rules of order, similar to petition and initiative), where justifications represent the arguments for gathered signatures (aka votes). Each motion is relevant to an organization (organization = rules defining a constituency and its jurisdiction). Organizations are independent of each other.

The main predefined types of exchanged items are: *peer, organization, constituent, motion, neighborhood, witness, justification, vote, news, translation*, and *tester*. These items are synchronized between the peers specifying an interest for them. Users have various possibilities to control what kind of data their agent disseminates and what data it stores. For example, they can block all data related to a given organization, motion or constituent. Alternatively peers may accept only data related to specified organizations, motions or constituents. *News* items are related to contexts defined by organizations, motions, and justifications. They can be controlled with quotas per constituent and function of their creation date. *Translation* items refer to the labels of the platform for the graphical interface related to specific organizations. They constitute the basis of a P2P recommendation system for the translations of these labels into the preferred language of the user.

The system also handles *plugin-data* items that do not have GIDs and are exchanged only between peers specified by the corresponding plugins. Currently there are two distributed applications supported by plugins and available with the system: a spacecraft racing game, and a chat application.

The *DirectDemocracyP2P* system consists of several modules, such as: *Data Handling, GUI, Internet and NAT, Supernode, Ad hoc Wireless, Authentication Census, Automatic Updates, Tester Recommendation, Plugin Interface* (Fig. 1).

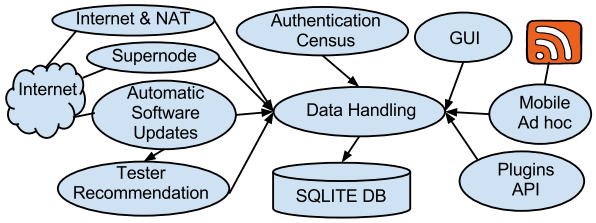

Fig. 1. DirectDemocracyP2P peer architecture.

- The *Data Handling* module is an interface with the personal database of the peer, saving and reading items with verification of integrity (digital signatures).
- The *GUI* module allows the user to view and modify the data in her database and the current state of the system.
- The *Internet and NAT* modules pack and unpack messages to be exchanged between peers via the Internet and contain client and server parts that can communicate both over TCP and UDP (from behind NATs).
- The *Supernode* module enables users with sufficient resources to support other peers in communicating from behind NATs. It can negotiate incentives.
- The *Ad hoc Wireless* module is in charge of configuring ad hoc networks and in packing and unpacking messages broadcast over them. It contains a server and a client.
- The *Authentication Census* module is a reputation system for detecting false identities and Sybil attacks on the collaborative filtering employed by the argumentation.
- The *Automatic Updates* module monitors a set of user configurable mirrors to download and install software updates that pass a user configurable set of criteria, based on reviews from her preferred testers.
- The *Tester Recommendation* module is a P2P recommendation system for suggesting testers to users.
- The *Plugin Interface* module loads available plugin modules dynamically and coordinates the transmission of messages between plugins installed on different peers via the communication system employed by the main application. It also provides a mechanism to exchange certain data (e.g., identity, GIDs) between the main application and plugins, as well as access to local storage and GUI space for menus and applets under the control of plugins.

[1]Florida Tech, [2]Nagoya Inst. of Tech., [3]Kyushu Univ., [4]Kobe Marine Univ.
[5]http://DirectDemocracyP2P.net

978-1-4799-0521-8/13/$31.00

I. Joining Other Peers via Address Containers

Independent networks of peers can coexist. When user *Alice*, wants to accept communication from *Bob*, she exports her address to a file (address container). If the selected file name provided by *Alice* stands for an existing image, then the address is embedded inside this image. This file/image can be emailed to *Bob*, or can be posted an the web.

Bob only has to drag and drop the address container to his software agent. If *Alice* is found behind a NAT, or may move to different IP addresses, she can register herself with a supernode (a node with a stable Internet IP). Each agent has an integrated supernode that an user can enable if she has access to a server with the right resources. The addresses of adopted supernodes are embedded in the address container. An address container encapsulates the GID of the peer (public key), the IP addresses of its machine, and the IP addresses of the supernodes that serve this peer.

II. Automatic Update

Our solution to security of automatic updates from *stacking the deck attacks* [5] (i.e., take-over of the development process by powerful players unhappy with a democratic process), inserts independent intermediary $testers$ (software reviewers) between the developers and the end-users. The independent testers can build and test an existing source code revision from an open source repository, and then distribute a signed binary release of it together with reviews [4]. To encourage independence of the testers, essential for the desired security, a P2P recommendation mechanism is employed. It suggests testers for end-users using various metrics (such as: reviews, connection distance, frequency of usage) [5].

III. Incentives for Supernodes

In our approach, each human owning a peer can control the traffic supported by her system. Peers found behind NATs require significant support from computers with Internet IP addresses, called supernodes. Users with access to servers that have the needed resources are encouraged to offer supernode services with incentives such as: (a) helping the endpoints of the communication, e.g., based on friendship, (b) helping a cause, e.g., based on the content/topic of the communication, (c) reputation, or (d) the utility brought to the supernode user by the handled data (which can be seen as subsuming the case b) [6]. Figure 2 illustrates the current interface where users can configure the negotiation of their agents with the supernodes.

IV. Ad hoc Wireless

The wireless ad hoc communication module allows peers to create a unique wireless SSID, `DirectDemocracy`, at frequency 2.462 GHz, resulting in the cell 46:32:D1:F2:88:67. This cell is common for all peers using the application. The module interface detects wireless interfaces, and can be configured to use any subset of them.

Each of the broadcast queues are preloaded with messages. Configurable policies are used in order to load items of certain

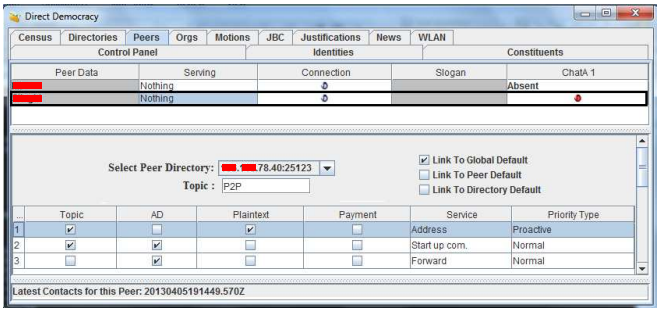

Fig. 2. Managing incentives at the client-side: adding new terms, updating existing terms and setting terms priorities.

types (personal, similar to personal, recent, random, round-robin, requested). The agent will only broadcast items related to specified interests. These queues have specific mechanisms for loading and reloading.

The broadcast client picks items from the various existing queues based on a probability distribution that can be specified by the user. A utility-based scheme is used to automatically optimize this probability distribution in order to maximize the satisfaction of interests of the sender. The utility-based scheme uses information such as: the number of other vehicles traveling in the same direction or in opposite direction, their relative speed, their declared interests, and the worth associated by the user to the dissemination of different types of information [7]. A GPS device can be used to extract the needed information.

V. Demo Items

The demo focuses on: communication modules (NATs, ad hoc wireless networks, supernodes), GUI control, and plugins. It shows how to manage one's peer identity and to connect to other peers, how to create or join organizations, how to review other peers, to submit and sign motions and news, and to specify interests and filters. It illustrates how to develop and install plugins, demonstrates available ones, and how to register testers and mirrors for automatic updates.

References

[1] M. Silaghi, S. Qin, K. Alhamed, T. Matsui, M. Yokoo, and K. Hirayama, "P2P petition drives and deliberation of shareholders," in *Workshop on Decentralized Coordination*, 2013.

[2] S. Qin, M. Silaghi, T. Matsui, M. Yokoo, and K. Hirayama, "P2P decentralized population census," in *Workshop on Decentralized Coordination*, 2013.

[3] S. Qin, M. C. Silaghi, T. Matsui, M. Yokoo, and K. Hirayama, "Reputation system for decentralized population census," in *IJCAI* 2^{nd} *Workshop on Incentives and Trust in E-Commerce (WIT-EC'13)*, 2013, pp. 37–48.

[4] K. Alhamed, M. Silaghi, I. Hussien, and Y. Yang, "Security by decentralized certification of automatic-updates for open source software controlled by volunteers," in *Workshop on Decentralized Coordination*, 2013.

[5] K. Alhamed, M. C. Silaghi, I. Hussien, and Y. Yang, "Distributed recommendation of testers for software updates in agent systems," in *IEEE Intelligent Agent Technology (IAT)*, 2013.

[6] K. Alhamed and M. Silaghi, "Incentive human-to-human chat support protocol," in *Workshop on Decentralized Coordination*, 2013.

[7] O. Dhanoon, R. Vishen, and M. Silaghi, "Protocol and heuristics for synchronizing opinion poll items in vehicular ad-hoc network," in *Workshop on Decentralized Coordination*, 2013.