TOWARDS A DECENTRALIZED AND SECURE ELECTRONIC MARKETPLACE

Yingying Chen
Department of Computer Science, Rutgers University
Piscataway, NJ 08854, USA
yingche@cs.rutgers.edu

Constantin Serban
Department of Computer Science, Rutgers University
Piscataway, NJ 08854, USA
serban@cs.rutgers.edu

Wenxuan Zhang
Department of Computer Science, Rutgers University
Piscataway, NJ 08854, USA
wzhang@cs.rutgers.edu

Naftaly Minsky
Department of Computer Science, Rutgers University
Piscataway, NJ 08854, USA
minsky @cs.rutgers.edu

ABSTRACT

For commerce (electronic or traditional) to be effective, there must be a degree of trust between buyers and sellers. In traditional commerce, this kind of trust is based on such things as societal laws and customs, and on the intuition people tend to develop about each other during interpersonal interactions. The trustworthiness of these factors is based, to a large extent, on the geographical proximity between buyers and sellers. But this proximity is lost in e-commerce.

In conventional electronic marketplaces the trust among participants is supported by a central server that imposes certain rules of engagement on all transactions. But such centralized marketplaces have serious drawbacks, among them lack of scalability, and high cost.

In this paper we propose a concept of *decentralized electronic marketplace* (or DEM, for short) which would allow buyers and sellers to engage in commercial transactions, subject to an explicitly stated set of rules, called the *law* of this marketplace—which they can trust to be observed by their trading partners. This trust is due to a scalable decentralized mechanism that enforces this stated law, and to the reputation mechanism that is also supported by the law of DEM.

KEYWORDS

Electronic Commerce, Secure Electronic Marketplace, Distributed Systems, Decentralization, Decentralized Enforcement, Law Governed Interaction

1. INTRODUCTION

For commerce (electronic or traditional) to be effective, there must be a degree of trust between buyers and sellers. When buying an airline ticket, for example, the buyer needs an assurance that what he (or she) is getting is an authentic ticket, issued by the airline in question, and that it is not forged and cannot be duplicated. Also, if the payment for the ticket is done via a credit card, the buyer needs to trust the seller not to use the credit card for anything but the transaction at hand, and not to disclose it to anybody else.

In traditional commerce, the trust between buyers and sellers, as it is, is based on such things as societal laws and customs, and on the intuition people tend to develop about each other during interpersonal interactions. The trustworthiness of these factors is based, to a large extent, on the geographical proximity between buyers and sellers. It is the physical venue of the trade that is subject to specific trading laws, which may be enforced by the local police; and it is the eye contact between the trading partners that may induce a degree of trust between them.

But no such physical venue exists for the electronic marketplace. Moreover the participants in electronic commerce might reside in different countries, and may be subject to different laws and different customs. The trading partners are also invisible to each other, and are often immune from traditional kind of law enforcement. We need therefore some other, non-traditional, means for inducing trust in such a marketplace. The conventional approach to electronic marketplace is to use a *centralized server* to mediate all transactions between buyers and server, subject to some rules of engagement built into this server—which can, thus be trusted by the trading partners. Examples of such marketplaces include EBay (http://www.ebay.com), the Ford marketplace for automotive parts (http://www.pricingcentral.com/ford/), Open Market [9], and AuctionBot [14].

But although these particular marketplaces operate effectively, the general concept of centralized electronic marketplace has several serious drawbacks. First, a centralized mediator of electronic transactions is a single point of failure, and could become a bottleneck, with sufficiently large number of participants. These weaknesses of centralization can be alleviated, in particular, by massive replication—but only at a great cost, as is evident from a system like EBay. (And even EBay, with its enormous resources, crumbled under a denial of service attack.) This means that it is hard and expensive to start a new electronic marketplace of this kind. Another problem with this approach is that the rules that govern a given centralized marketplace are usually implicit in the code of its server, and may not be fully available to the buyers and sellers that use it.

In this paper we propose a concept of *decentralized electronic marketplace* (or, DEM, for short) which would allow buyers and sellers to engage in commercial transactions, subject to an explicitly stated set of rules of engagement, called the *law* of this marketplace—which they can trust each other to observe. This trust is due to a decentralized, and thus scalable, mechanism that enforces the stated law of the DEM.

A DEM is characterized not by any physical server that manages it—as there is none—but by the law that governs all transactions made through it. Such a marketplace can be launched by essentially defining its law. This act that has no real cost because it does not involve the creation of any central mediator. Once launched, a DEM can grow, in a scalable manner, simply by sellers and buyers joining it. (In practice, such growth is likely to require some advertising, which is not discussed in this paper.)

The architecture of such decentralized marketplaces is based on the concept of Law Governed Interaction (*LGI*) [7,8], which provides means for the specification of the law of a given DEM, and for its decentralized enforcement. The LGI concept has been implemented as a middleware, available for public access at http://www.moses.rutgers.edu/ , and it can, in principle, support our concept of DEM. But for a DEM to be usable by buyers and sellers throughout the Internet, this middleware needs to be commercially, and widely, deployed. Such deployment is beyond the scope of our work, and of this paper.

The rest of the paper is organized as follows: Section 2 introduces a motivating example of a DEM used for the trading of airline tickets. Section 3 is the outline of LGI, which provide the computational basis for our concept of marketplace. Section 4 describes the implementation of the example marketplace introduced in Section 2. Section 5 discusses some related work. And we conclude in Section 6.

2. AIRLINE TICKET TRADING – A MOTIVATING EXAMPLE

We introduce here an example of a decentralized marketplace, called AT, for trading airline-tickets. We focus on the law that is to govern this marketplace, we express it here informally, calling it a “policy”, and we later discuss its formalization as an LGI law. But we must first introduce the various participants in this marketplace, and explain broadly the role that they are to play.

The participants in this marketplace include the following: (1) The *airlines*, which provide sellers with their tickets, and have no other involvement in the marketplace. (2) The *banks*, which provide credit card authorization and money transfer services for buyers and sellers (we assume here that all payments are to be

done via credit cards). (3) The *sellers* are required to be authenticated. (4) An *auditor*, which receives copies of messages that were exchanged during the trading. And (5) two *certification authorities:* ca_1, is employed to authorize airlines, banks, and the auditor; ca_2 is employed for authenticating the identity of buyers and sellers.

Note that the only other agent involved in the trade beside the buyer and the seller is the bank. A bank needs to be involved here, as it is in traditional commerce, because we have chosen payments via credit cards. Would we have chosen payment via digital cash [10], which is easy to do under LGI (as we have done in [2]), we would have no direct central involvement in an individual purchase.

This marketplace is to be governed by the following seven-point policy, stated here informally:

1. *Authenticity of the tickets:* Tickets sold by sellers are required to be authentic, i.e. issued by the specified airline. The sellers should not be able to forge or copy tickets: every such ticket can be sold only once.
2. *Security and privacy of credit card payment:* Payments under this marketplace are to be made via credit cards, with the following guarantee to buyers: (a) the credit card would be charged only for the cost of the purchased airline ticket, and only once; and (b) no information about the credit card being used would be leaked to the seller itself.
3. *Ticket reservation:* If a seller agrees to reserve a ticket for a specified period of time, it is not allowed to sell it to anybody else.
4. *Money back guarantee:* A buyer can cancel a transaction, by returning the ticket within a certain time period following the purchase. The buyer is guaranteed to receive its money back (minus a service fee, perhaps).
5. *Monitoring:* A copy of all message exchange between buyers and sellers is sent to an *auditor*.
6. *Authentication of Identity:* The sellers are required to identify themselves via certificates issued by the certifying authority ca_2.
7. *Reputation services:* The tracking and reporting of reputation will be provided for, in a decentralized manner. This will be done along the lines discussed in [14], and explained briefly in Section 4.

3. AN OVERVIEW OF LGI

LGI is a mode of interaction that allows an open group of distributed heterogeneous agents[1] to interact with each other with confidence that the explicitly specified policies, called the *law* of the open group, is complied with by everyone in the group [7][8]. The messages exchanged under a given law *L* are called *L*-messages, and the group of agents interacting via *L*-messages is called a *community* C, or more specifically, an L-community C_L.

By the phrase "open group" we mean (a) that the membership of this group can be very large, and can change dynamically; and (b) that the members of a given community can be heterogeneous. LGI does not assume any knowledge about the structure and behavior of the members of a given L-community. All such members are treated as black boxes by LGI, which only deals with the interaction between them. Members of a certain LGI community are allowed to participate in other LGI communities or in normal communication not regulated by LGI.

For each agent *x* in a given L-community, LGI maintains the *control-state* CS_x of this agent. These control-states, which can change dynamically, subject to law *L*, enable the law to make distinctions between agents, and to be sensitive to dynamic changes in their states. The semantics of control-states for a given community is defined by its law, and could represent such things as the role of an agent in this community, privileges, and reputations.

We continue this section to further discuss the concept of law, emphasizing its local nature, and with a description of the decentralized LGI mechanism for law enforcement. The concept of *obligation* is elaborated briefly. We do not discuss here several important aspects of LGI, including the *interoperability* between communities, the treatment of *certificates* and *exceptions*, the deployment of L-communities, and the

[1] Given the popular usages of the term "agent", it is important to point out that we do not imply by it either "intelligence" nor mobility, although neither of these is being ruled out by this model.

performance of its current implementation. For a complete understanding of these issues, the reader is referred to [8][12]. An explanation of LGI, with examples, can also be found at www.moses.rutgers.edu.

The concept of law and its enforcement: Generally speaking, the law of a community C is defined over a certain types of events occurring at members of C, mandating the effect that any such event should have – this mandate is called the *ruling* of the law for a given event. The events subject to laws, called *regulated events* include (among others): the *sending* and the *arrival* of an L-message; the *coming due* of an *obligation* previously imposed on a given object; and the submission of a *digital certificate*. The operations that can be included in the ruling of the law for a given regulated event are called *primitive operations*. They include, operations on the control-state of the agent where the event occurred (called, the "home agent"); operations on messages, such as *forward* and *deliver*; and the imposition of an obligation on the home agent.

Thus, a law L can regulate the exchange of messages between members of an L-community, based on the control-state of the participants; and it can mandate various side effects of the message-exchange, such as modification of the control states of the sender and/or receiver of a message, and emission of extra messages.

Figure 4 displays a section of the Java implemented *LAT* law that deals with ticket authenticity issues. A Prolog formulation of the law is also available. The law implements the following event methods: sent and arrived; it accesses the control state through the CS variable; and it calls the following primitive operations: doForward, doDeliver, doRemove, and doAdd.

The arrived event deals with the situation when a message sent by an airline agent arrives at a seller; the sent event deals with the situation when a seller sends a ticket confirmation to a buyer agent. A more detailed description of the ticket authenticity rules in the *LAT* law is offered in Section 4.

Enforced obligation: Informally speaking, an obligation under LGI is a kind of *motive force*. Once an obligation is imposed on an agent – generally, as part of the ruling of the law for some event at it – it ensures that a certain action (called *sanction*) is carried out at this agent, at a specified time in the future, when the obligation is said to come due, and provided that certain conditions on the control-state of the agent are satisfied at that time. Note that a pending obligation incurred by agent *x* can be *repealed* before its due time. The circumstances under which an agent may incur an obligation, the treatment of pending obligations, and the nature of the sanctions, are all governed by the law of the community.

The local nature of laws: Although the law L of a community C is *global* in that it governs the interaction between all members of C, it is enforced locally at each member of C. This is possible due to the following locality properties of LGI laws:

- L only regulates local events at individual agents.
- The ruling of L for an event *e* at agent *x* depends only on *e* and the local control-state CS_x of *x*.
- The ruling of L at *x* can mandate only local operations to be carried out at *x*, such as an update of CS_x, the forwarding of a message from *x* to some other agent *y*, and the imposition of an obligation on *x*.

The fact that the same law is enforced at all agents of a community gives LGI its necessary global scope, establishing a *common* set of ground rules for the members of C and providing them with the ability to trust each other, in spite of the heterogeneity of the community. And the locality of law enforcement enables LGI to scale with community size.

Distributed law-enforcement: Broadly speaking, the law L of community C_L is enforced by a set of trusted agents called *controllers*, that mediate the exchange of L-messages between members of C_L. Every member *x* of C has a controller T_x assigned to it (*T* here stands for trusted agent) which maintains the control-state CS_x of its client *x*. And all these controllers, which are *logically* placed between the members of C and the communication medium as illustrated in Figure 1 carry the *same law L*. Every exchange between a pair of agents *x* and *y* is thus mediated by *their* controllers T_x and T_y, so that this enforcement is inherently decentralized. However, several agents can share a single controller, if such sharing is desired. The efficiency of this mechanism, and its scalability, are discussed in [8]. Secure transmission between controllers is carried out via traditional cryptographic techniques, and we assume that these controllers are correctly implemented. Furthermore a community can choose to only use controllers certified by a certain certification authority(*CA*), which is specified by the law L.

Controllers are generic, and can interpret and enforce any well- formed law. A controller operates as an independent process, and it may be placed on any machine, anywhere in the network. We have implemented

a *controller-service*, which maintains a set of active controllers. To be effective in a widely distributed enterprise, this set of controllers need to be well dispersed geographically, so that it would be possible to find controllers that are reasonably close to their prospective clients. We postulate the existence of a "public utility" of trusted controllers which we call a controller-service. It is an organization that maintains a set of trusted hosts that function as controllers. The controllers, the controller-service and the related utility are currently implemented by a middleware called Moses. Moses is released in September and it is available for public use.

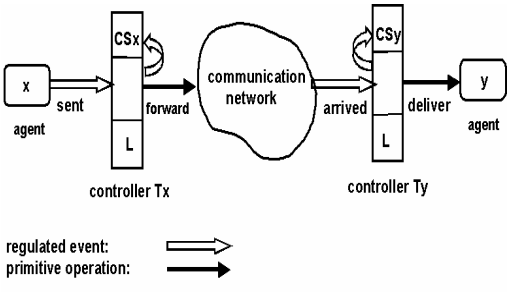

Figure 1. Enforcement of LGI through controllers

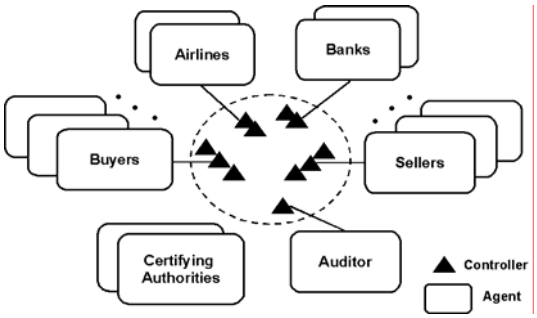

Figure 2. Architecture of LGI-based airline ticket trading

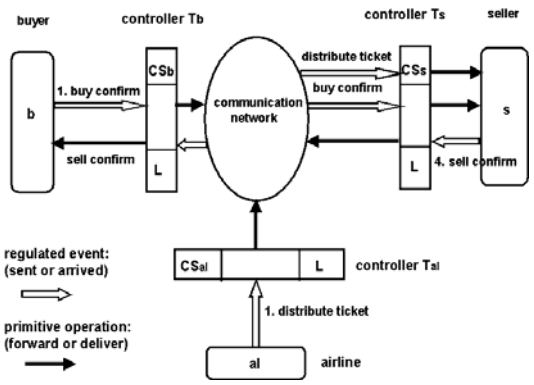

Figure 3. Flow of ticket authentication

Ticket Authenticity – Fragment of the *LAT* Law

```
law(LAT,language(java))

/* The body of the law*/
import java.util.*;
public class LAT extends Law {

public static final String ctrl= "controller";
/* Event methods*/

/* When a ticket is confirmed, its hash is compared against a
hash previously stored in CS. Upon success, send the ticket
and remove the hash from the CS. Otherwise send a denial
message*/

public void sent(String source,String message,
                        String destination, String destlaw) {
     String serial_num = getMsgContentFirst(message);
     String order_num = getMsgContentLast(message);

     /* Ticket authentication is checked*/
     if (CS.has("ticket("+hash(message)+")")){
        String c_name = getCustomerName(order_num);
        if (c_name != null){
            /* Prevents double selling of the ticket */
            doRemove("ticket("+serial_num+")");
            Term t = Cs.find("customer_order("+order_num+
                                    ",%NM,%OI,%PS)");
            doRemove(t);
            doForward(Self, message, c_name);
         } return;
     }
     doDeliver(ctrl,"ticket_not_authentic ",Self);
 }

/* Term ticket(hash(tk)) is added into CS for every ticket tk
sent by an airline al that arrives at the seller s.*/

 public void arrived(String source, String sourcelaw,
                            String message, String dest){
   if (message.startsWith("distribute_tkt")){
        String serial_num = getMsgContentFirst(message);
         doAdd("ticket(" +hash( message)+")");
         doDeliver();
   }
 }

 /* Helper methods*/
 public String getMsgContentFirst(String message){ … }
```

Figure 4. Ticket Authenticity–Fragment of the *LAT*

4. IMPLEMENTATION OF THE AIRLINE TICKET MARKETPLACE

A marketplace is represented by a dynamic set of buyers, sellers and other actors that are involved in a trading activity as introduced in Section 2. The marketplace is defined practically by the set of rules (policy) governing the common interaction between all the actors involved in the trading. Formally, in LGI this set of rules represents the law of airline ticket trading for the marketplace community. This law, called *LAT* represents and defines the AT marketplace. It can be observed that by changing the trading law, LGI could be easily used to apply to other trading applications in the electronic marketplace.

Figure 2 shows the interaction within the marketplace defined by the law *LAT*. A set of controllers represents the decentralized trusted infrastructure that mediates all the interaction between the actors, thus enforcing the trading policy. First buyers, sellers and other actors examine *LAT*. If they agree with it, they can decide to participate in the marketplace by adopting the controllers under law *LAT*.

The following explains the implementation of various rules from the trading policy in Section 2. A short fragment of the corresponding *LAT* law is presented in Figure 4, and the entire law is available at: http://www.moses.rutgers.edu/examples/marketplace/trade.java1 - for the Java implementation of the law and http://www.moses.rutgers.edu/examples/marketplace/trade.law for its Prolog counterpart.

Ticket authenticity and reservation: Every ticket *tk* sold by a seller is required to be authentic, i.e. issued by the specified airline. The sellers should not be able to forge or copy tickets: every such ticket can be sold only once.

Such forgery is prevented in a straightforward manner if the marketplace involves a central entity: for every bought ticket, the buyer can verify with the (central) airline that the ticket is authentic and it has not been previously sold. This solution, however, introduces the airline in the direct trading path, impeding the decentralized nature of the marketplace. Another centralized solution is to employ a trusted third party holding the ticket to verify and enforce its validity.

DEM achieves this property by putting the trust on the controller. Whenever a seller receives a ticket from an airline, the seller's controller maintains the hash of the ticket in its control state. Whenever a ticket is sold, its hash is compared against the hash stored in the control state.

Usually an airline ticket *tk* contains crucial data such as serial number *SN*, date and time *DT*, source *SO* and destination *DE*, class *CL*, and etc. During the ticket distribution from airlines *al*, the one-way hash $h_{tk}(SN, DT, DE, CL)$ can be computed for each ticket and stored in the control-state of the seller *s* . In the case of *s* altering some of the information in *tk* the controller T_s will detect the fraud and prevent the selling message of *s* from reaching the buyer *b*. One possible fraud is to change the class level *CL* of a ticket in order to make more profit, and then to send a fraudulent selling response *tk'* to a buyer *b*. Once *tk* is sold, the hash value of *tk* will be removed from T_s. This method prevents the ticket to be sold more than once by a dishonest seller as follows. When the selling message from *s* arrives at T_s, if T_s could not find the hash value of *tk*, it infers that the ticket has already been sold, and thus it denies the fraudulent selling message originating in *s*. The complete interaction involving the airline, the seller and the buyer is depicted in Figure 3, and a fragment of LAT dealing with the implementation of these rules can be observed in the code in Figure 4.

To ensure that a buyer *b* obtains the ticket it asked for, we have implemented a ticket reservation scheme. Following a ticket request from a buyer *b*, a seller *s* will put *tk* into a "reserved" state on behalf of *b*. The reservation state is in effect for a certain time period and it is implemented using the *obligation* mechanism at the controller of *s* (mechanism discussed in more detail in Section 3). If the buying confirmation of *b* arrives at T_s before the obligation is *due*, *tk* is sold to *b* and the obligation is *repealed*. Otherwise, *tk* will be released from the reservation state when the obligation comes *due* and it will be free to new queries from other buyers.

Security and privacy of credit card payment: As discussed in the trading policy of Section 2, when shopping for airline tickets, the following guarantee will be made to a buyer *b*: (a) the credit card would be charged only for the cost of the purchased airline ticket, and only once; and (b) no information about the credit card being used would leak to the seller itself.

The local controller T_s of the seller *s* protects the buyer's confidentiality by maintaining the credit card information without disclosing it to the seller itself. T_s will perform the credit card authorization check for a buyer *b* by sending a credit card checking request to bank *bk*. Meanwhile, the controller maintains a repository *R* for active buyers at *s*, by saving a customer_order in the local *CS* of *s*. This information is

transient and it will be removed once the order is completed either successfully or by failure. If the credit card check is successful, T_s will deliver the purchasing order to *s* in order to take an appropriate action. Otherwise, if the credit card authorization fails, T_s will send a denial message back to *b*.

It is worth mentioning that in certain electronic marketplaces a buyer may not want a seller to know its identity due to the sensitivity of the type of products to be purchased; this kind of identity privacy protection could be easily achieved by LGI using the approach discussed here.

Monitoring: *LAT* provides monitoring of the trading activities for the purpose of auditing. If an auditor exists, an agent can request copies of the messages exchanged during the trading and use this information to study the behavior of the agents involved in the marketplace. The trading activities are monitored to ensure that a copy of every transaction is sent to an auditor. More than one auditor can exist and *LAT* is flexible to specify what transaction messages are sent to which of them. Even though *LAT* specifies what messages are sent to what auditors, it imposes no restrictions on the way the auditors handle the messages they receive.

Authentication of Identity: There are situations when a buyer *b* would like to securely identify a seller *s*, and obtain assurance that it is legitimately established and accredited. *LAT* provides authentication of identity in the following manner. When joining the L-community, each airline *al*, bank *bk*, or auditor *a* has to present a *digital certificate* to its controller. This certificate, signed by the certifying authority ca_1, serves for authentication of identity and in order to assess the role as airline, bank, or auditor. Prior to selling airline tickets, each seller *s* has to present a *digital certificate* signed by ca_2 to its local controller for authentication of identity. This prevents *s* from impersonating other sellers. After the certificate has been presented at the controller, and it has been verified to be valid, a *certified* event will be triggered in the *LAT* law. As a result the identity of the seller is added into the *CS* of the respective agent.

Reputation Tracking: Reputation tracking of the sellers is a service necessary to a marketplace. Unlike in traditional e-marketplaces where reputation management requires an online central server, *LAT* implements a decentralized reputation tracking mechanism to store and update the reputation information for each active seller. The reputation information of each seller *s* is maintained and updated in its controller T_s. For each successful trade, a buyer can rate *s* by updating its *rating*, while its associated *seniority* grows automatically.

5. RELATED WORK

We have already pointed out that the conventional approach to electronic marketplaces is based on a central mediator. We have explained the limitations of this approach, despite some very successful system, such as EBay. Others have observed these limitations as well. We will mention here two such cases.

First, Schmees [10] in his 2003 paper "Distributed digital commerce," discussed the benefits of decentralized market for digital goods, and studied the processes involved in digital trading and their implementation using P2P communication. Although Schmees admitted the importance of trust and security in the marketplace, he did not propose any mechanism for achieving them. The DEM model proposed in this paper addresses exactly these issues.

Second, the European SEMPER [6][13] project attempted to examine systematically the security requirements of electronic marketplaces, and proposed a framework for addressing them. The resulting open security architecture of SEMPER offers users the ability to select components of choice from the SEMPER libraries, and to associate a certain level of trust with these components. Before trading, SEMPER proposes a series of agreements that establish a set of rules for each role: buyer, seller, bank, certification authority, and etc. Users playing these roles can commit to abide by these rules. The agreement is signed on paper with a third party. It establishes in advance the liability of the parties regarding the future transactions to conduct. The basic trust assumption of SEMPER has been that each user trusts his or her own machine, but not the machine of the partner. The SEMPER project proposed no practical implementation, and had no continuation after the project has been completed in 2000.

Finally, in a project closely related to the present one, which also employs LGI, a Decentralized Peer-to-Peer Auctions had been proposed [4].

6. CONCLUSION

The concept of *decentralized electronic marketplace* (DEM) proposed in this paper has no physical place in which the market takes place, and not even a virtual place, in a form of central manager, and mediator for all transactions. Yet, it deserves the name "marketplace" in that it provides a single, unifying, law that govern all the transactions made through it—in some analogy to the laws that govern traditional marketplaces.

The law of a given DEM is explicitly defined, and visible to all its participants. Moreover, the law is strictly enforced, via the LGI mechanism, in a completely decentralized manner. This makes such a marketplace easy to launch, essentially by writing its law; easy for a buyer or a seller to engage in, simply by locating an authenticated controller (using some controller-service), and adopting it with the law of the DEM; and easy to scale, due to the freedom from any central mediator.

We provide supporting evidence to the efficacy of this concept by presenting a study of a DEM devoted to trading in airline tickets, governed by law (*LAT*). Although this law has been only partially described here, due to space limitation, it has been fully defined and experimented with, and it is available on our web site.

It should be pointed out, however, that although the LGI mechanism, on which our concept of DEM is based, has been fully implemented (and has been released for public use), for a DEM to be usable by buyers and sellers dispersed throughout the Internet, the LGI middleware needs to be commercially, and widely, deployed. Such deployment is beyond our capacity, and we can do here no more than advocate that this would be done by some financial or governmental institutions.

REFERENCES

[1]Abad-Peiro, J., Asokan, N., Waidner, M., 1998. Designing a generic payment service. *IBM Systems Journal, Internet Computing*, volume 37, number 1, page 72,.

[2]Ao, X., and Minsky, N., 2003. Flexible regulations of distributed coalitions. In *Proceedings of the Eighth European Symposium on Research in Computer Security (ESORICS).* Norway.

[3]Ellison, C., 1999. The nature of a usable pki. *Computer Networks*, (31): 823-830,.

[4]Fontoura, M., Ionescu, M., and Minsky, N., 2005. Decentralized Peer-to-Peer Auctions. In *the Journal of Electronic Commerce Research (JECR*).

[5]Ionescu M., Minsky N., and Nguyen T., 2004. Enforcement of Communal Policies for Peer-to-Peer Systems. In *Proc. of the Sixth International Conference on Coordination Models and Languages*. Pisa Italy.

[6]Lacoste, G., 1997. SEMPER A security framework for the global electronic marketplace. *Comtec – the magazine for telecommunications technology*.

[7]Minsky, N., 1991. The imposition of protocols over open distributed systems. *IEEE Transactions on Software Engineering*.

[8]Minsky, N., and Ungureanu, V., 2000.Law-governed interaction:a coordination and control mechanism for heterogeneous distributed systems. *TOSEM, ACM Transactions on Software Engineering and Methodology*, 9(3): 273-305.

[9]Open Market, 1998. *Internet Commerce: The Open Market Transact Solution.* Technical white paper, Open Market, Inc.Available from www.openmarket.com.

[10]Schmees, M., 2003. Distributed digital commerce. In *Proceedings of the 5th International Conference on Electronic Commerce*. ACM, S.131-137. Pittsburgh, PA, USA,.

[11]Schneier, B., 1996. *Applied Cryptography*. John Wiley and Sons.

[12]Ungureanu, V. and Minsky, N.,2000. Establishing business rules for inter-enterprise electronic commerce. In *Proceedings of the Fourteenth International Symposium on Distributed Computing (DISC 2000)*. LNCS 1914, pages 179-193, Toledo, Spain.

[13]Wainder, M., 1996. Development of a secure electronic marketplace for Europe. In *Fourth European Symposium on Research in Computer Security (ESORICS)*. Rome, Italy.

[14]Wurman, P., Wellman, M., and Walsh, W., 1998. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In Katia P. Sycara and Michael Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 301-308, ACM Press. New York.