SICS Technical Report T2006:16

ISRN:SICS-T--2006/16-SE
ISSN 1100-3154

Trust in Micro Service Environments

by

Magnus Boman, Jarmo Laaksolahti, Fredrik Espinoza, Rickard Cöster

2006-11-03

espinoza@sics.se
Swedish Institute of Computer Science
Box 1263, S-164 29 KISTA, SWEDEN

Abstract:
Report produced in the project Enabling and Promoting Trust in Micro Service Environments (EPTMSE) with a web site at www.trust-eze.org. The report gives an overview of the concept of trust in domains such as psychology, sociology, philosophy, and computer science, and then describes the current domain of Micro Service Environments – open and unregulated electronic service environments – where users can create, use, and share electronic services, and where the need for decentralized trust mechanisms is high. Some design and implementation choices and solutions for trust mechanisms are suggested.

Keywords:
Trust, electronic services, service sharing, service environment, trust framework.

Contents

1 Introduction

The goal of the EPTMSE project is to investigate the nature of trust in – and create trust mechanisms for – micro service environments. Such environments allow users to create and distribute their own services in addition to using services created by others. As a result the total number of available services in a system is potentially increased. In addition new services, that better meet user requirements, can be released at a faster pace as there are more developers. However, micro-service environments also raise several trust issues. Normally when using services authored by someone else we e.g. trust well known brand names (Microsoft, Sun, Oracle, etc) to deliver high-quality and safe services. In a micro-service environment where all users are potentially also service providers such brand names do not necessarily exist. So, how can users know that the service they are downloading is not authored by a virus maker? How can they be sure that a service is not so poorly implemented that it crashes their system? Finding ways of helping users to decide whether to trust a service or not in micro-service environments is a major challenge. Ultimately the goal of EPTMSE is to take steps towards meeting that challenge, making it possible for users of micro-service environments to answer the questions:

1. What or who should I trust?

Imagine that a user has found a number of services s/he would like to use, e.g. book trading services. The sheer number of services and service creators in a micro-service environment makes it unlikely that the user will know anything about the services s/he finds, or their creators, in advance. Hence it will be difficult to decide which services to trust or not.

2. In what way should I trust them?

In real life we trust car mechanics to fix our cars but not to cure our ailments. In the same way users must decide in what way to trust electronic services.

3. Why should I trust them?

Finally users must decide why they should trust a service in the first place? What are the *trust cues* that make it reasonable for users' to assume a trusting stance towards a service or its creator? In real life users have access to information such as, brand name, reviews or friends opinions that may be missing in micro service environments.

However, in order to accomplish that we need to clarify what it is we wish to help users accomplish. What exactly do we refer to when talking about trust? Is it a cognitive state of mind or an emotion? Is trust equal to security (in IT systems)? Does trust entail anonymity? What are the dimensions of trust? Can dimensions contradict each other? Trust is a complex phenomenon with many possible meanings and interpretations. We need to clarify which dimensions and meanings of trust are relevant for micro-service environments.

As a first step in creating trust mechanisms for micro-service environments this report will start by surveying literature regarding trust in a number of fields including sociology, psychology, computer science and electronic commerce. The report will also include examples of systems implementing trust mechanisms. The purpose of the report is to clarify what characterizes trust in various fields, what the types and dimensions of trust are, and relate those to micro service environments. The report will serve as a basis for designing a set of trust mechanisms for micro service environments later in the project.

2 Trust Definitions and Theories

Trust has traditionally[1] been the object of study in a number of disciplines including psychology, sociology and philosophy. Consequently there is a substantial amount of literature available on the topic. Furthermore the amount of literature and use-cases is growing as interest in trust-theories and their application in computer systems keep growing.

Moral philosopher Annette Baier used the following as a starting point when defining trust: “One leaves others an opportunity to harm one when one trusts, and also shows one’s confidence that they will not take it. Reasonable trust will require good will, or at least the absence of good grounds for expecting their ill will or indifference. Trust then, on this first approximation, is accepted vulnerability to another’s possible but not expected ill will (or lack of good will toward one”. In fact this ability and propensity to trust is an important social lubricant for cooperative behavior (Luhmann, 1979). Without the existence of trust it would be very difficult to engage in any of the cooperative and/or social behavior common to our society.

Baier also suggests that trust is something that we usually do not consciously think about but instead sees it as inhabiting “a climate of trust as we inhabit an atmosphere and notice it as we notice air, only when it becomes scarce or polluted”. In fact, according to Baier, criminal elements have been the experts at discerning, and utilizing different forms of trust. Hence trust is something we become fully aware of only after something happens that injures it. Trust in the “inhabiting an atmosphere” sense is seldom of a contractual form where an agreement of the limits of trust and consequences for violating it are clearly stipulated. In everyday situations people generally rely on their powers of discretion to determine what is entrusted to them and under what conditions. For instance, when A tells B something very personal and/or intimate s/he is clearly relying on B to keep the information to herself.

Baier defines trust as a three place-predicate, *A trusts B with valued thing C,* where *C* can be any number of things, such as ones health, possessions, privacy, friendship or even children whereas *A* and *B* are people, or represented by people e.g. firms, or nations. Although never explicitly stated by Baier it is clear that according to her, trust relationships, i.e. relationships of the form above, only exists between people and never between artifacts and people. This is supported by her treatment of the discretionary powers that she envisions *A* and *B* to have that allows them to make *intelligent* choices about what to entrust (on A’s part) and what to do with entrusted goods when received (see e.g. Friedman et al 2000).

The question begging an answer then is who should I trust, in what way and why? If trust is something we don’t notice until it is gone, how can we build an atmosphere of trust?

Within psychology trust has been studied from different perspectives. *Personality theory* conceptualizes trust as a trait that can be more or less developed in individuals depending on their prior experiences. *Behavioral psychology* often equates trust with cooperation. Trust is operationalized as a trusting, or cooperative, choice of behavior. Studies performed within the

[1] By traditionally we mean outside the context of computing or computer usage.

field, aiming to reveal the nature of trust, are often based on scenarios like the "prisoners dilemma". In experiments various situational variables are tweaked to determine their effect on the level or trust (cooperation) or distrust (competition) between players (Lewis & Weigert, 1985). One of the most prominent members of this group is Morton Deutsh.

Deutsch defines trust based on the following scenario:

a) An individual is confronted with an ambiguous path that can lead to an event perceived to be beneficial (Va+) or an event perceived to be harmful (Va-)
b) She perceives that the occurrence of Va+ and Va- is dependent on the behavior of another person; and
c) She perceives the strength of Va- to be greater than the strength of Va+

> If the person takes a path with such ambiguous properties the person is said to make a trusting choice; if the person does not take the path s/he makes a distrustful choice.

Based on this scenario Deutsch summarizes trust as "confidence that one will find what is desired from another, rather than what is feared" (Deutsch, 1973). Marsh (1994) notes that trust in Deutsch view is subjective, based on subjective views of the world. Therefore people's assessments of trustworthiness do not necessarily coincide. Deutsch definition also seems to involve some form of cost/benefit analysis, a feature found in many trust definitions. Again making these time consuming cost/benefit analyses for each possible future is in most cases unviable. Trust allows people to assume that certain things are given which equates to assigning a zero or one probability to them.

Most trust definitions seem to embrace the idea that trust is about confidence in some aspect of relationships. However Deutsch also recognizes other types of trust that do not stem from such confidence for instance *trust as despair*. Such trust occurs when the negative consequences of not trusting or staying in the present state of affairs, are so great that the trusting choice is made out of despair. All in all Deutsch recognizes nine types of trust: trust as despair, trust as social conformity, trust as innocence, trust as impulsiveness, trust as virtue, trust as masochism, trust as faith, trust as gambling and trust as confidence. *Trust as confidence* is the area Deutsch has focused on and is what we normally mean by trust.

In contrast to the behavioral view of trust advocated by Deutsch sociology considers trust to be a property of collectives (groups of people) that is applicable to relations among members of the collectives, rather than their individual psychological states (Lewis & Weigert, 1985). Gambetta (2000) citing Luhmann (1979) and Dunn (1984) states that trust has been defined as a device for coping with the *freedom of others* (freedom to violate our trust). Hence trust can be viewed as inherently social.

One of the most influential trust theorists overall is Niklas Luhmann a German sociologist who in fact only published two works dealing directly with trust: Trust and power (English translation 1979) and *Familiarity, Confidence, Trust. Problems and Alternatives* (1988). Luhmann seems to have inspired most modern trust theorists. According to Luhmann trust is necessary for the functioning of modern society. In fact without trust we could not function as human beings.

> "Trust, in broadest sense of confidence in one's expectations, is a basic fact of social life. In many situations, of course, man can choose in certain respects whether or not to

> bestow trust. But a complete absence of trust would prevent him from getting up in the morning." (Luhmann, 1979).

Luhmann argues that the main purpose of trust is to reduce the complexity of the world and the many possible choices an individual has to make in it. Given the complexity of the world we inhabit it is not possible to plan for all possible contingent futures, hence we need some mechanism for making this complexity manageable. Trust, in contrast to other possible mechanisms such as rational prediction, does not require us to collect and process information about known causal relationships in order to determine which futures are probable and which are unlikely. Even at the best of times rational prediction is an unviable way of reducing complexity as we simply do not have the available time and resources to keep track of possible future outcomes. Trust succeeds where rational prediction fails "because to trust is to live as if certain rationally possible futures will not occur" (Lewis & Weigert, 1985). Behaviorally this means that to trust is to "act as if the uncertain future actions of others were indeed certain in circumstances wherein the violation of these expectations results in negative consequences for those involved" (p. 971). Hence, from a sociological point of view the practical significance of trust lies in its social action potential.

A somewhat more formal description of trust is given by Gambetta (2000), who summarizes notions of trust from several authors as: "trust (or symmetrically, distrust) is a particular level of subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both *before* he can monitor such action (or independently of his capacity ever to be able to monitor it) *and* in a context in which it affects *his own* action. When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him". Trust in this sense is a threshold point located on a probability distribution ranging from 0, complete distrust, to 1 complete trust, and normally located somewhere around the center. People, or agents, with a predisposition to assign the extreme values and stick to them regardless of evidence, are said to be displaying blind trust or distrust. Uncertainty about other peoples' behavior is central to this notion of trust. We can never hope to know everything about other peoples' motives or responses to changes and if we did there would be no trust involved since we would know how the person would behave beforehand. Hence, without uncertainty there is no trust to be assigned.

Like Baier and Luhmann, Gambetta speaks about agents having a degree of freedom to disappoint our expectations as a necessary condition for trust. Baier talks about this freedom in terms of discretionary powers.

Gambetta also gives an illustrative example of how, in a community that would benefit from mutual trust and cooperation, sub-optimal behavior still may emerge, i.e. everyone looses to some extent. For instance although using public transportation and bicycles instead of using cars would cause less pollution and traffic jams, people still seem to prefer traveling by car. According to Gambetta: "What is lacking is the belief that everybody else is going to cooperate, which generates the fear of being the only 'sucker' around to sweat on the pedals, and the corresponding unwillingness to cooperate".

Given the lack of information at hand and the complexity of the choices that we need to make when deciding to make a trusting choice or a distrusting one, an interesting question is why we should decide to trust instead of adopting a generally distrustful stance. While at a first glance a distrustful stance would seem to be better Gambetta argues that in the long run fostering an atmosphere of mutual distrust is sub-optimal. Instead he argues that adopting a trusting stance (however not blind trust) can do no worse than a distrusting one and in most cases does better. Hence given a situation where the outcome is unknown and where there is no established trust between the actors we can lean towards assigning the outcome a sufficiently low probability of being let down for us to trust. The reason being that:

> "The first is that if we do not, we shall never find out: trust begins with keeping oneself open to evidence, acting as *if* one trusted, at least until more stable beliefs can be established on the basis of further information" Gambetta (2000).

2.1 Trust and risk

A majority of researchers in the field recognizes trust as a means of handling risk (see e.g. Luhmann, 1979; Gambetta, 2000; Lewis & Weigert, 1985). Taking a risk in practice means being faced with a decision situation. There are three basic strategies for handling risky situations. First, one could accept the risk. Since one knows that this might result in bad consequences, it is a good idea to prepare for such consequences. In the case of running someone else's software on your own device, it is a good idea to put virus scanners and other protective software on full alert initially, for example. Secondly, one could mitigate the risk by taking steps to limit the possible damage. A full backup of your vital files before running someone else's software would be an example of mitigation. The possibility to roll back would still require some time and effort, should the software turn out to be malicious, but basically one could return to normal before long. Thirdly, one could transfer the risk. This strategy means that someone else takes full responsibility for any bad consequences. For example, that "someone else" who authored the software might be a colleague sitting next to you, assuring you that should there be any bugs in the code, she will take care of them, then and there. The person sitting next to you might also promise you to pay for your lunch that day, in appraisal of you (unknowingly) beta testing her code.

The third case, risk transfer, is particularly interesting from the perspective of trust. It dates back to the 18th century, where the concept of insurance was born in London tea houses, populated by wealthy merchants who offered shipping insurance. Using their collective knowledge about ships, weather, crews, pirates and other relevant matters, they offered buyers their money back for a fee, should the cargo arrive damaged or not at all. Before long, this grew to cover life insurance and later property insurance in general. As a result, the tea house societies grew into respected institutions, such as *Lloyds*. People trusted the societies at first because of their knowledge about the statistics and logistics involved in the relevant situations, e.g., shipping tea from India. Later, people came to trust the institutions based more on their reputation and record.

If we compare this history to the ongoing case of software provision, the issuing of certificates from established computer software companies springs to mind. The established company here plays the role of the institution, and the person downloading the certified software seeks insurance, i.e. risk transfer. True, malicious software may in rare cases have been certified by established companies, but insurance companies and banks occasionally go bankrupt too. For that latter case, there is reinsurance: a reinsurance company insures insurance companies. In the case of banks, the state often reinsures. In the case of insurance companies, multi-national reinsurance companies usually take this role. For software provision, there is not yet anything analogous to reinsurance. In fact, the issuing of code certificates is itself not 100 per cent safe, as selling certificates is a business, and so attracts non-serious issuers. This fact notwithstanding, there is much to be said for the above analogy, since certificates and any form of software verification (and even some forms of software validation) constitute a form of risk transfer, which always requires trust.

To economists and decision analysts, trust is often coded as risk willingness or even more crudely as willingness-to-pay. A large part of their modeling focuses on the person facing the risk. If she is risk prone or risk averse (or risk neutral) is determined by her utility function. The psychological aspects of trust involved in software provision and adoption are therefore important to the theoretical models under development. Economists are also interested in the macro-economical aspects of the coding of risk transfer (and hence trust) in the area of software provision, for many reasons. First, it is a huge market already. Secondly, as a market it shows rapid growth. Thirdly, the pricing models employed may prove general enough to be generalized to other qualitative procedures for which there are none or only weak theoretical models today. In summary, trust is being studied in connection with risk from the very smallest (micro) perspective of each individual's willingness to pay for risk transfer, to the very largest (macro) perspective of the properties of the market for such payments.

March (1994) states that " … a knowledge of risk and its implications allows us to make plans for the future which take the risks into account, and thus make extensions to those plans which should succeed in the face of problems or perhaps unforeseen events". As such plans are expensive to make for all contingencies trust is an effective way of overcoming the problem.

2.2 Dimensions of Trust

Research shows that trust can vary along a multitude of dimensions. Based on a survey of existing literature Corritore et al. (2003) list five dimensions, *generality, kind, degree, stage* and *level*, along which trust can vary.

Generality refers to how *broadly* a person trusts another. For instance a person can have overall trust in another person i.e. trusting a person in everything from business relationships to personal relationships. Trust can also be very specific, limited to certain domains or circumstances e.g. when trusting a doctor to treat you correctly.

Kind refers to several attributes of the trust relationship including how fast it is and whether it is cognitive or emotional in nature. Slow trust generally builds up over time and is typically found in long term relationships such as working relationships. In contrast swift trust is found in short-term relationships for instance temporary workgroups. It builds quickly but also ceases to exist quickly once the relationship is terminated. Corritore argues that it is likely that swift trust applies to specific trust while general trust requires slow trust. Lewis and Weigert (1985) suggest that there are two kinds of trust: cognitive trust and emotional trust. Cognitive trust involves having good rational reasons for trusting the object of trust whereas emotional trust is motivated by strong positive feelings towards the object of trust. Both forms are usually intertwined but cognitive trust may be more common in large settings or loosely tied groups such as work relationships, whereas emotional trust is more common in smaller closely knit groups, such as families.

Degree of trust is synonymous to depth of trust. According to (Brenkert, 1998) it ranges from *basic* to *guarded* to *extended*. Basic trust is an underlying trust that the world will stay more or less familiar. For instance we trust our neighborhood to be as safe today as it was yesterday. Guarded trust is protected by formal contracts and agreements that stipulate the object of trust and what happens should a breach of trust occur. For instance we contract painters to paint our homes and trust that they are competent to perform their work. Guarded trust is of course limited to the extension of the contract and thereafter expires. Finally extended trust is based on openness. It requires deep relationships that have evolved over time that need not be guarded by formal contracts. Typically friends have such relationships and give extended trust. Another example could be when leaving your credit card number to an online site such as Amazon (although the site's use of your credit card number is undoubtedly also circumscribed by contracts of some sort).

Trust develops over time. Hence it is also characterized by its current stage of development. The first time I use the services of some company I have initial trust for them. Later on if I repeatedly use their services and they perform satisfactorily my trust in the company matures and deepens. Lewicki & Bunker (1996) propose a developmental model of trust that moves from deterrence-based, to knowledge-based to a shared identification-based trust. Deterrence-based trust is an initial trust that is guarded by contracts. Knowledge based trust is an intermediate form of trust that involves sufficient knowledge about the object of trust so that (some) predictions about its behavior can be done. Shared identification-based trust is a mature trust based on deep knowledge about the object of trust and a shared understanding about the trust domain (or valuable good in Baiers terms) and is not guarded by contracts.

3 Online Trust

Essentially online trust is no different than offline trust. The same mechanisms and theories are still applicable. However one aspect of the online world that complicates trust is the large amount of people and services that users can immediately come into contact with, and the fact that most of those people and sites are rather anonymous.

There are at least two main thrusts within the community researching online trust. One is focusing on trust based on encryption and digital signatures while the other is focusing on social mechanisms for regulating trust.

Trust schemes based on encryption and digital signatures attempt to instill trust by keeping untrustworthy elements outside the perimeters of the system if possible, and at the same time guaranteeing the identity of elements inside the system perimeters. Within the system trustworthy behavior is promoted by the fact that breaches of trust can be tracked to the source and acted on.

Trust schemes based on encryption, digital signatures etc. can instill some degree of trust by among other things ensuring accountability. They do not, however, solve the whole problem of online trust. For instance digital signatures do not tell us much about how *trustworthy* a person, website or service is. It only tells us that if our trust is violated we will at best be able to track the erring party and hold them accountable. As such, techniques based on encryption may be good at instilling a sense of trust in the early stages of a trust relationship (see. 2.2). When presented with a decision situation where the choice is between several unknowns' digital signatures, and the associated accountability, may well tip the scale in favor of a party using it. Despite this parties largely remain 'strangers' to each other. To be able to say more about the trustworthiness of a party we also need to know more about their previous dealings.

Reputation systems attempt to imitate some aspects of how trustworthiness is created offline through word of mouth. When for instance looking for a new dentist, we may consult our friends to get some recommendations. If one (or more) of them have been seeing a dentist that they are happy with they may provide us with a recommendation or a warning in the opposite case. Reputations build over time and reflect people's opinions about the performance, abilities and dispositions of parties. Reputation systems basically automate this process by keeping transaction histories for all participants in a community and aggregating this information so it can be used to assess the trustworthiness of a party. Furthermore reputations are believed to have a positive effect on fostering trustworthy behavior as "An expectation that people will consider one another's pasts in future interactions constrains behavior in the present" (Resnick et al., 2000). However, reputation systems also suffer from their share of problems, specifically problems related to scarcity of data. For instance, when new parties enter the community they have no history (no previous transactions are recorded) and consequently there is nothing we can say about their trustworthiness. Another problem is that parties are (usually) required to explicitly provide feedback about how they perceived a transaction (good, bad, ok, …) for the system to work. Often enough there is little incentive for parties to do this (Resnick et al., 2000).

Naturally there are mixtures of the two approaches such as systems working with both digital signatures and reputation systems. While encryption has a longer history than reputation systems the latter recently seems to have received the bulk of interest.

According to the definition outlined by Friedman (2000) trust depends on the ability to experience mental states such as good will towards others, feeling vulnerable or betrayed. As these mental states all presume the existence of consciousness, something which humans have but machines do not, Friedman argues that people do not trust machines but other people. As a result it is pointless to speak about trusting technical systems including services in a micro service environment. However, Friedman grants that in some cases we can speak about *relying* on technical systems but not trusting them.

To date on-line trust has mostly been investigated with respect to websites in e-commerce settings. Although limiting their definition of on-line trust to a specific type of object – transactional or informational websites – the definition proposed by Corritore et al (2003) resembles the one proposed by Baier. The definition simply states that an individual's trust towards a website of the above mentioned kind is: "an attitude of confident expectation in an online situation of risk that one's vulnerabilities will not be exploited".

However this definition does not seem to account for situations where an interaction actually does have harmful consequences for the user, not due to any malicious intent but poorly designed or implemented software. Harmful situations of this kind seem to be (to us) more likely and common in many online situations especially micro-service environments than other types of harmful situations.

As pointed out earlier one question is of course if interactions with technology can ever be said to be trusting or just something we rely on. Nevertheless an individual's ability to correctly assess the risks (fraud, technical breakdown, etc) that are inherent in any online transaction is an important ingredient of trust. However, the technology behind the online world is so complex and rapidly evolving that making (well informed) assessments becomes an insurmountable task for all but the most technically knowledgeable. Hence as stated by Friedman et al.; "…our trust in the designers of technology (or technological artifacts) is bounded by our understanding of the conditions under which the technology functions reliably and safely.” (Friedman et al, 2000).

3.1 Trust in Peer-to-peer systems

The web is based on a centralized client-server architecture. Clients send requests to servers that store all information, all files, services etc. The structure and content of the web is very much dictated by those controlling the servers. Recently another type of system has emerged that works in a more distributed manner. Peer-to-peer systems are systems where all participants take part as equals (more or less anyway). Each peer functions both as a server (for other peers) and a client. The information contained in a peer to peer system is distributed over all the currently connected peers. Hence should some fraction of the connected peers suddenly become unavailable the network will still be able to maintain its functionality. If a user was in the process of accessing information on a peer that becomes unavailable chances are high that the same information is duplicated on some other peer. Hence peer-to-peer systems are much more resilient than traditional client-server systems. In addition many believe that “P2P reflects society better than other types of computer architectures” (Clark, 2001).

Freenet is an example of a peer-to-peer system that attempts to instill a sense of trust in its users by reducing the number of people that they must trust. By guaranteeing anonymity for both producers and consumers of information people do not have to trust anyone. The system backs up this lofty promise by virtue of its routing mechanism that hides both the origin and destination of file requests. Requests to add or retrieve files are treated in the same way and appear to the receiver as if the sender were simply passing along the request. The actual files are transmitted in the same way. In this manner, the producer and consumer of files remain anonymous. One of the motivating factors behind this is to increase user participation. If a user is guaranteed anonymity she is more likely to contribute willingly to the system. While the anonymity guaranteed by Freenet ensures the privacy of its users, there is still the issue of the actual files being transferred between users. Anyone can provide files to the system and rest easy knowing that he can never be accredited as a file's provider. Because of this, unscrupulous users can introduce low quality, legally questionable, or even harmful content to the network. It is the responsibility of the peer that makes requests to protect herself from such content. Freenet is interesting not because of its offering of anonymity to its users, but rather because of its method of reducing the number of people one must trust.

Like Freenet, file requests in Gnutella are anonymous by virtue of the system's search mechanism. However, file downloads are done directly between peers and not via intermediaries. Because every Gnutella peer is identified by an IP address, it is possible to catalog information about which peers host which files. And while the IP address itself might not be directly usable to identify an individual person, there are indirect ways in which this can be done. The Gnutella Wall of Shame attests to the system's weakness in ensuring privacy for its users. A Gnutella peer that made available files advertised as containing child pornography (they didn't) logged all of the users that allegedly tried to download these files. The IP addresses of the offenders were later published on a web site available to public scrutiny. This event makes evident the fact that even making pseudonymous information available can have negative consequences for a system's user. Like Freenet, Gnutella makes no concerted effort to ensure the quality, reliability, or authenticity of the information that is exchanged. Unlike Freenet, Gnutella peers are the actual hosts of the files that they make available. This gives peers the means, albeit indirectly, to generate reputations about other hosts. If a user provides low quality information, then other peers will avoid downloading information from him in the future. This is difficult, however, as peers may obtain their IP addresses dynamically, so a host that behaves poorly on the network may return to the network at a later time with a different IP address. The interesting point here is that trust may be built by rating nodes in the network. OceanStore is an extremely wide-area information archival system that is intended to span the globe, giving its users access to persistent information regardless of geographic location. Un-trusted servers owned by third parties form the system's infrastructure making OceanStore's task of manifesting trust challenging. OceanStore aims to instill a sense of trust in its users through the pervasive use of encryption and data redundancy. The system is designed to function much like a traditional file system, and incorporates user accounts in conjunction with strict access control lists (ACL) to ensure that unauthorized parties cannot access information. In addition, all information is encrypted throughout the system, whether on the wire or on disk. This is an example of a more traditional approach to instilling trust: by ensuring that all data is protected and encrypted with trusted mechanisms the system as a whole earns the users' trust.

Reputation based trust management has lately received much attention from the research community. Using this method in P2P systems is however somewhat problematic as: “information about transactions between peers is dispersed throughout the network so that every peer can only build an approximation of the global situation in the network” (Aberer & Despotovic, 2001). In addition one has to take into account that some peers may be tampering with the stored transaction data for their own benefit. Hence the idea of distributing the reputation web in the same way as all other data seems like a good idea and has been proposed by several authors. Aberer & Despotovic presents a system that is based on a distributed data model that can efficiently be accessed and updated. The system does not require a central database of transactions but instead computes trustworthiness from information it receives from peers. More specifically each peer forwards information about dishonest transactions through the network, while honest transactions are considered to be the norm and not reported. While it conserves bandwidth and is efficient from a computational view it is also a weakness of the system. It leads to a binary notion of trust in the system; an agent is either trustworthy or not. Hence there is no (easy) way of discerning between degrees of trustworthiness or quality of service.

Another P2P based reputation management system is presented by Selcuk et al. (2004). Their approach is based on storing information about every other peer it has dealt with in the past in trust vectors. In contrast to the previous system information about honest transactions are also stored with each agent. If a peer does not have any information about a potential transaction partner it can query its’ peers about information they might have available. Once received the information is summarized and stored. The number of past transactions to base trustworthiness calculations on ranges from 8-32. While this model provides a more nuanced view of trust it requires more resources in terms of data stored in each peer. In typical P2P scenarios, where transactions are being made with a large number of peers, this amounts to a significant amount of data. The model also requires users to actively rate downloaded files as benign or malicious and implicitly assumes that files are not downloaded in “chunks” from several peers simultaneously.

In the context of reputation systems assessments regarding trustworthiness may be seen as recommendations to trust or not. However it is possible for peers to leave untruthful recommendations and thereby render the system weaker. Obtreiter (2004) suggests using nonreputable tokens such as digital signatures to overcome this problem. Each statement, or transaction, in the system would generate a receipt that either of the parties involved in the transaction can use to prove their claims. For example when one peer has transferred a file to another peer a receipt is generated that confirms the transfer. If the receiver later on claims that the file was not transferred the receipt functions as evidence that it in fact was. The effect of this is that recommendations become verifiable (in a sense) and lying about trustworthiness becomes difficult (given that peers can not issue receipts indicating successful transactions to themselves).

The Intimate Trusted Advisor (ITA) is an interesting system design that directly builds on Luhmanns definition of trust as a means of reducing complexity (Cofta, 2004). The user can ask ITA for advice regarding whether to trust technological entities or chains of entities in their surroundings. ITA calculates a trust rating for entities based on how much using a particular entity would reduce the complexity of a user's situation. As a basis a simple model is used that takes three factors into account:

1. the extent of proof of trustworthiness of an entity
2. net reduction in complexity that using an entity would lead to
3. user's inherent propensity to trust

No information is however provided on how any of the above values would or could be calculated. Nevertheless the approach is an interesting one as it directly attempts to help users cope with the complexity of their technological context.

3.2 Designing for Trust

There is an abundance of literature that explores trust in considerable depth. However there seems to be a lack of literature dealing with how to design for trust, at least from a general perspective. Friedman et al. (2000) suggests ten trust related characteristics/issues of on-line interactions that need to be addressed when designing systems intended to cultivate conditions for trust online.

Technological reliability and security. Even the most knowledgeable individual cannot determine if a service is technically secure by inspection alone. Hence certain vulnerabilities are always unknown.

Online risks. People have very limited information about how big the risks in an online environment are. For instance what are the chances of being infected by a virus or being duped in an online commercial transaction?

Misleading language and images. Designers should not use misleading language and images to convey to users greater reliability and security in the technology than is warranted.

Definition of Harm. It is unclear how harm should be defined. For instance is accessing a person's private file without reading the contents harmful or not?

Informed Consent involves telling users about the potential harm or benefit of an online transaction and giving users a real choice between participating or declining to participate in the transaction.

Anonymity, or the absence of identifying information associated with an interaction, is a complex issue. On one hand, if we focus on protecting ourselves from potential harm inflicted by online transactions, anonymity is a good thing. On the other hand anonymity can obstruct a climate of trust by making assessments of potential harm more difficult. Accountability and anonymity are closely related.

Accountability. Knowing who you are dealing with and being able to track them if need be is a requirement for many types of trust especially the guarded type of trust that is protected by contracts. While accountability can significantly bolster trust it is also somewhat problematic as it conflicts with anonymity. A high degree of anonymity makes accountability harder and vice versa. When designing for trust there is an important balance to be struck here.

Saliency of trust cues. Sometimes the presence or absence of certain cues is important. For instance information about area of expertise may be vital when deciding to trust medical advice.

Insurance. If something should go wrong in an online transaction having some kind of arrangement to compensate persons for harm done to them is an important trust factor. This could e.g. be financial compensation covering costs. Insurance may also be considered a mechanism for rebuilding trust if/when breaches of trust occur.

Reputation. Previous performance experiences, both direct and reported by others, can be a valuable tool for assessing the trustworthiness of a party.

Corritore (2003) gives an overview of literature dealing with trust cues, cues that convey trustworthiness. Here we will present a few of those trust cues. While her review focuses on trust cues for websites there is no reason to believe that the same cues would be invalid in other circumstances. Not surprisingly important trust cues are the visual design of websites and ease of navigation within the sites. In general the overall look and feel of websites is important. Other factors that have been found to have an impact on trustworthiness is ease of carrying out transactions, easy access to live customer representatives via the website, and good grammar.

Providing the right content, content that is appropriate and useful to the target audience, is also a strong trust cue. On the negative side, mixing advertisements and content has been found to be a negative cue as has poor site maintenance (broken links, outdated information, missing images, long download times etc).

All in all these are all important aspects to consider when designing for trust. Many of the findings are of a common sense nature and should be directly transferable to domains other than websites. Some of course are specific not just to websites but to certain types of websites e.g. ease of carrying out transactions which is only applicable to commercial websites.

4 Discussion

EPTMSE aims to investigate mechanisms for enabling *user perceived trust*. Due to a number of factors including the complexity of an open service sharing framework, the potentially endless variations of possible services and the lack of a central controlling authority there can be no absolute guarantees that users' trust is not sometimes dishonored. Regardless of any trust enabling mechanisms, services downloaded by a user may prove to be unsafe or unstable, they may match their descriptions poorly and they may perform badly or deliver substandard content just to mention a few possibilities. Hence what we are dealing with is not absolute unbreakable trust, if such a thing exists, but trust as experienced by users of a micro-service environment.

4.1 Stakeholder perspectives on trust

Creating trust in online, mobile, environments is important for several groups of stakeholders including operators, service providers and users. However, depending on which role is assumed trust can mean very different things and have different ramifications for a number of areas such as how services are created, distributed and used, how business is conducted, questions of responsibility and how risks are managed.

One aspect of the environment that is important for trust is the level and ownership of control over content in the environment. As illustrated by Figure 1 this can range from full control by some trusted party, e.g. an operator or service provider, to a completely open situation where all parties equally share control over the environment.

A service sharing framework must support at least a basic set of functions – or usage activities – to operate properly. They can roughly be divided into the following non-sequential functions/activities:

- Create services

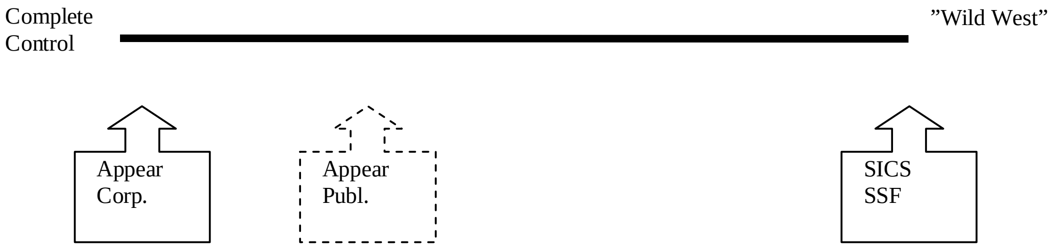


Figure 1 Dimension of control

- Share services
- Find services
- Obtain services
- Use services
- Manage services (update, delete, revoke, …)

The actual implementation of these functions will likely differ depending on requirements of the context in which they are deployed.

For instance the Service Sharing Framework (SSF) perspective is one of complete openness. Each service consumer can also be a service provider. The range of services offered by the framework is wholly determined by its users. In such an environment there is no central authority that can guarantee the trustworthiness of services. It is up to each individual user to decide whether to trust a service or not. The framework offers a multiplicity of services in the same way as the Open Source movement, but also a potentially lower level of trust. In an open environment all of the mentioned usage activities are fully supported.

However, a completely open environment may conflict with the (business) goals of some stakeholders. For instance questions of ownership or IPR may be important issues that need to be controlled. The service distribution policy of Appears Networks' APS system is based on an administrator that is responsible for setting up services for each location, e.g. a sub-way station, a room or a more undefined geographic area. When entering an APS network (or zone) users are automatically presented with a set of services that are available and executable depending on the context (location, time of day, device, etc). Users themselves can not influence the selection of services available at a certain location in other ways than by suggesting changes to the administrator. However, in some cases, if the service is designed for it, users can save a service to their device and use it in other locations as well. The administrator of an APS system also functions as a trust proxy in that s/he can vouch for the trustworthiness of the services in the system. Such a system does not have the same multiplicity of services available but does in all likelihood have a higher level of trust. In a more restrictive scenario like the one described some usage activities – e.g. service sharing – may not be supported at all.

In both models a service administrator, be it an end user or a system administrator, has to decide whether to trust a service or not. Hence both models seem to be quite similar. However, even in a controlled environment end users have to decide whether or not to trust a service whether a system administrator has approved of the service or not. For instance, I may trust that my system administrator has properly installed a word processor on my machine but still not trust the software itself e.g. due to susceptibility to macro viruses.

The Operator (TeliaSonera)

From the user's point of view, the operator has a high level of responsibility for preventing the users from getting harmed. This includes situations where the services do not fully work, where one service disturbs other services, when users are charged unexpectedly, where privacy is compromised, and many other situations.

Even in situations out of the operator's control, e.g. modem hijacking, users expect the operator to take full responsibility, either by preventing it from happening, or if it still happens, by taking responsibility for the economical consequences.

It may seem unfair that users should expect this, but it may actually be good business for the operator, because it also indicates a high level of trust. Consequently, the operator will have a competitive advantage compared to many other actors in the telco eco-system.

The consequence of this is that in situations where the operator has a hard time to protect the users, new methods must be introduced and introducing trust mechanism in micro service environments is one of the methods. Together with this is it also important that the operator gets a better overall understanding of trust.

An alternative would be that the operator clearly informs users where the limit of the responsibility goes, but that threatens to harm the high level of trust that users have for operators today.

A wild-west scenario is a situation where the mobile terminals (phones, PDA etc.) are open, meaning that users can download applications from any software vendor provider and use services from any service provider. This situation exists today for many terminals, but because the application market is still quite small the impact is limited.

This will, however, become a problem when downloaded applications become more common. The users may encounter viruses, applications that interfere with the terminal’s normal operation, applications that do not deliver the services they are expected to, applications that cause enormous amount of costly traffic and much more.

The operator will surely want to prevent this type of problem, partly because the users are expecting this of the operator, and also because it is important for the operator to make the overall environment work if the operator wants to make money on the application and service business.

There are two possible methods to prevent the problems:

- One could close the terminal in such a way that only applications trusted by the operator are allowed, or that a mistrusted application is limited in functionality. This has as an additional advantage that the operator can control the market for applications, and has the ability to make money on applications.
- One could keep it open, but leave it to the user to determine if the application is trustworthy. As mentioned earlier, the application market is small, and one way to make the market grow is to let many actors profit from applications. To keep the terminal open is one possible way.

Today is it not clear which way the industry will go, but if “keep it open” is the future, methods to determine if an application is trustworthy will become more and more important.

Operators are likely to have a long term interest in trust issues, especially issues related to mobile terminals. The market is moving towards greater independence of access medium (wap, gprs, wlan, …) and operator. As many existing trust mechanisms are tied to a specific medium of access, and sometimes also an operator, new solutions are required. Currently there are a number of joint efforts involving a number of stakeholders that are starting to address, among other things, trust issues, for instance *Liberty Alliance*[2] and *Open Mobile*[3] *Alliance*.

The Service Aggregator (Appear Networks)

[2] http://www.projectliberty.org/

[3] http://www.openmobilealliance.org/

Appear Networks is a Franco-Swedish software vendor specialized in innovatory platforms for broadband wireless networks. Appear Networks markets technologies for wireless positioning and context-aware service presentation within IP wireless networks.

Appear Networks is most often a technology brick within an overall solution and the user's trust needs therefore to be considered from the perspective of the overall solution.
We need to differentiate trust related to the channel of communication (reliability of the network, such as its level of security) and trust related to the service itself.
The user usually expects the operator to ensure the quality of service; both regarding the channel of communication and the services. In deed for the user, it is he operator's responsibility to ensure the overall quality of service.
Of course, Appear Networks needs to commit to the same level of quality of service, in order not to jeopardize the quality of service of the overall system.

A secure deployment of the APS requires that the following components and actors are trusted to fulfill their responsibilities. The list below describes the requirements for a secure APS deployment:

Local network link layer security
The APS is based on the assumption that the local network does not need to be secured at the link level. This is necessary since securing wireless networks is non-trivial. SSL is used to provide secure communications.

Wide area network security
The APS does not need a trusted WAN. Since SSL is used between the proxy and the server, the traffic can be sent over public networks.

System access control
It is required that the computers running the publisher, the proxy and the server are secured from public access and can only be accessed by trusted administrators. This access control is outside the scope of the APS system itself.

Software components
The administrator can trust that the different components of the APS will properly implement authentication and access control, as well as properly securing the end-points of the communication.

Authentication plug-ins
By deploying an authentication plug-in the administrator trusts that the user authentication is performed in a secure way.

End-user
The end user is trusted not to give authentication credentials to other people. It is possible for an administrator to track user and device access to the system to trap logins from other devices than the usual.

Administrator

- The administrator is trusted to configure the system correctly.
- The administrator is trusted to configure services with the correct settings regarding clients of services platform support and requirements. The APS does not automatically check the content or requirements of services.
- The administrator is trusted not to deploy applications that would harm the end-user device such as viruses. The APS does not automatically check for viruses in services.
- The administrator is trusted to change user passwords.

• The administrator is trusted to configure role based access control in such a way that sensitive services are only available in categories limited to users with the required roles. The default access level for a category is unlimited access.
Once some kind of trust mechanism has been implemented between the technology actors, the next step is to convey trustworthiness to the user and get him/her to perceive some form of it.

Trust from the user's perspective

Industry research suggests that 95% of the PDA (Personal Digital Assistants) owners have never used any other application than the ones that were preinstalled in their devices. In fact, only 1% has really tried and succeeded in installing themselves a new application. For Appear Networks this shows clearly one of the bottlenecks in the technical value chain: the distribution mechanism needs to be improved.

Appear Networks does not look into the trustworthiness of the service itself but into a trustworthy manner of distributing the service to end users. Appear Networks' focus is on the user's experience in order to ease the access to and running of services. Ensuring the reliability of the service distribution process and making it more intuitive for the user helps enhancing the trust level. The user can not trust the service 100%, but he/she is conveyed trust by the overall environment.
Appear Networks focuses on proposing a user-friendly, trustworthy environment to the service (everything around the service).

To optimize access to information, Appear Networks uses two mechanisms:

a- Context-aware services, the idea being to:

- retrieve the maximum amount of information about the user of the wireless network and the context in which he is situated (user profile, geographical position, time/date, type of communicating equipment used, etc.);
- anticipate the requirements of the user from this information;
- dynamically and automatically offer the user the services that are liable to be of interest to them in their current context;
- guarantee access to the information with a single click;
- manage, upon leaving a context, the information that was related to this context, in the best way possible.

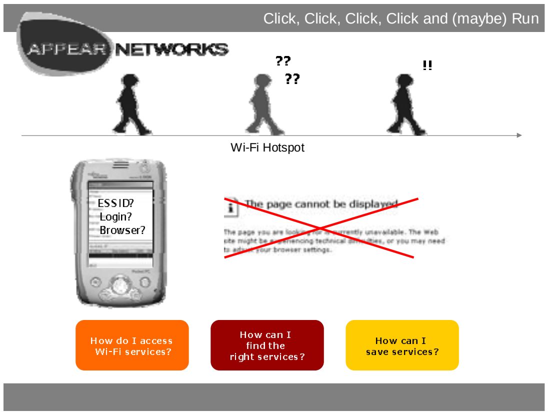
Thanks to context-awareness, all services that appear on the user's device are relevant for him/her which conveys trust since services are customized and perceived as such by the user.

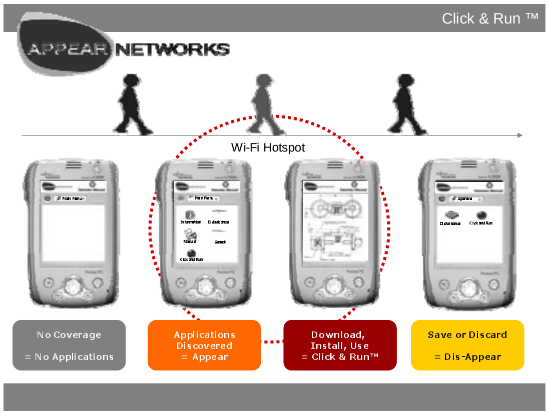
b- The Click&Run^{TM} technology:

- Services appear automatically according to the zone the user is in
- Applications can be downloaded, installed and run with a single click
- Applications can be saved for use in disconnected mode or disappear completely from the terminal's memory immediately upon leaving the hotspot

Click&Run^{TM} conveys trust to the user since he/she is always only one click away from any service. Moreover since services are automatically discarded from the terminal after disconnecting from the network, they are not perceived as intrusive which enhances the user's trust. The service distribution and discard processes are completely automated and the user does not have to worry about any technical aspects (configuration, installation, un-installation, etc.).

We are going from **Click, click, click, click and maybe run** to **Click and run**:


Click, Click, Click, Click and (maybe) Run
APPEAR NETWORKS
??
??
!!
Wi-Fi Hotspot
ESSID?
Login?
Browser?
The page cannot be displayed
How do I access Wi-Fi services?
How can I find the right services?
How can I save services?


Click & Run ™
APPEAR NETWORKS
Wi-Fi Hotspot
No Coverage = No Applications
Applications Discovered = Appear
Download, Install, Use = Click & Run™
Save or Discard = Dis-Appear

What type of trust mechanism could be implemented in order to reach, beyond the environment, services themselves?

4.2 Trust in Micro-Service Environments

Because of its peer-to-peer based distribution system, the characteristics of a micro service environment include: a lack of centralized coordination, a lack of a central database, incomplete knowledge of the whole system, global emergence of behavior from local interactions, autonomous peers, unreliable peers, and connections between peers. From this follows the questions:

1. Which model of trust should be used? Should it be based on statistics of prior market based or socially based experiences of peers or on game theory?
2. What algorithms can be used to establish trust given the data above? One has to consider that the sources of data may not be trustworthy, or available.
3. Can the trust system be made to scale? The data collection and communication between nodes must scale to the number of nodes.

A vulnerability of any framework is the possibility of being *hacked.* This has happened in several file sharing networks like Direct Connect (DC) or KAZAA. Users have for instance hacked the client software to report larger file shares than are actually present. This represents a big challenge for trust in micro service environments. In case the framework itself is hacked it is possible to bypass the systems' trust mechanisms and mark bad or even malicious services as trustworthy. This introduces another level of trust: trust in peers. Or more generally trust in the actual framework and its capacity to guarantee things like identity (although possibly anonymous) of peers and correct functioning of trust mechanisms.

Hence there seems to be (at least) three different aspects of trust in micro service environments:

- Trust in content that is delivered (e.g. quality of MP3 files or correlation between file name and file content)
- Trust in the functionality of services (trust the code)
- Trust in the framework and especially the trust mechanisms themselves

4.2.1 Trusting Functionality and Content

In general, creating trust using digital signatures is somewhat problematic since it requires that 99% of distributed serviced are signed to be effective. Otherwise users will face situations in which they are forced to accept unsigned content since there are no other alternatives.

An alternative to traditional digital signatures as such is to use various forms of *real-time signatures* based on the actual usage of services. By aggregating and analyzing usage/download statistics of a service the framework can provide users with a sense of how trustworthy services are. The *top list* service of the downloadable SSF distribution is a simple example of such a mechanism. The service maintains an ordered list of the most frequently downloaded services in the framework. The supporting idea behind the list is that bad news, i.e. a service is not to be trusted, travels fast. Hence frequently downloaded services are likely to be more trustworthy than those that are seldom downloaded since we in a sense know more about them. More complex forms of signatures where e.g. users are weighted differently, e.g. depending on expertise, similarity according to some profile, or how well you know someone, are also possible. In many cases trust is affected by how well you know persons and/or their competence. Such mechanisms can/should also “learn” from experience improving over time.

A great benefit of using a well defined framework like the SSF is that it provides infrastructure for creating real-time signatures such as top lists. This is harder to accomplish in environments such as the WWW.

In addition to downloads the framework makes it possible to take negative feedback such as un-installations and aborted downloads into account. In contrast many systems that rely on usage statistics to provide some service, e.g. recommender systems, suffer from only receiving positive feedback (a user has liked something). In such systems users must actively provide negative feedback (e.g. give a negative rating after reading a book), if it is possible at all, while the SSF can handle that automatically.

Trust mechanisms relying on aggregated information suffer from the same bootstrapping problem as recommender systems. Before a new service can be trusted a certain amount – or kind of – users have to download and use the service. But most likely users will not use services that do not have some form of signature yet.

Services that are “pushed” to users instead of “pulled” as above, introduce other possibilities but maybe other problems too. However the same signature mechanisms as above should also be applicable to pushed services. Pushed services can potentially be beneficial during service creation. In addition to the user actively searching for service components, services can propose themselves either based on users’ perceived task/profile or opportunistically based on knowledge about services users’ already have (possibly also in combination with users’ profiles). Agents that are sent from one user to another to perform some task are also examples of possible push services.

Trust mechanisms blocking certain types of files as found in many mail programs is one – albeit possibly crude – way of enabling trust for pushed services and content. The question is how the framework determines which services to block.

4.2.2 Trusting the Framework

Several issues related to sharing and management of services need to be resolved in the framework including:

- Control over who you share services with. Users might e.g. wish to create different trust tiers.
- Revoking services. If a service turns out bad or is published by mistake (or maliciously) there should be some way of revoking it, or at least marking it as "nuked".
- Discovering what services are capable of. Presentation of what is possible could be enhanced.
- Finding services. Currently only keyword based search is possible. Other mechanisms such as recommender systems or context based search should also be possible (or pushed services. See above.).
- Service/Peer browsing. In addition to search it should also be possible to browse available service collections (including virtual collections) or even peers. Collections could/should be divided into categories, either manually or automatically, to make browsing easier.
- Licensing mechanisms/Ownership

One solution to the revocation problem is to sign all services and construct the sharing infrastructure in such a way that when a service is started is checks the validity of its certificate. Revoking a service then simply becomes a matter of revoking the certificate. Time limited services would also become possible by issuing a time limited certificate. However it would also involve a rather complicated certificate management architecture and possible involvement of third party certificate issuers such as VeriSign.

4.3 Personalized trust

Most literature on the subject seems to focus on systems of trust where trust, accountability etc. is guaranteed by some global mechanism that stays the same for all users. Very few systems seem to have been created that take into account individual differences regarding risk adversity and trust cue preferences and catered for those.

We envision that these are important factors for creating trust in micro service environments, especially since they often lack a central authority that can guarantee trust. Hence what we envision is rather a form of personalized trust mechanism that is configurable by each user; a micro-trust mechanism.

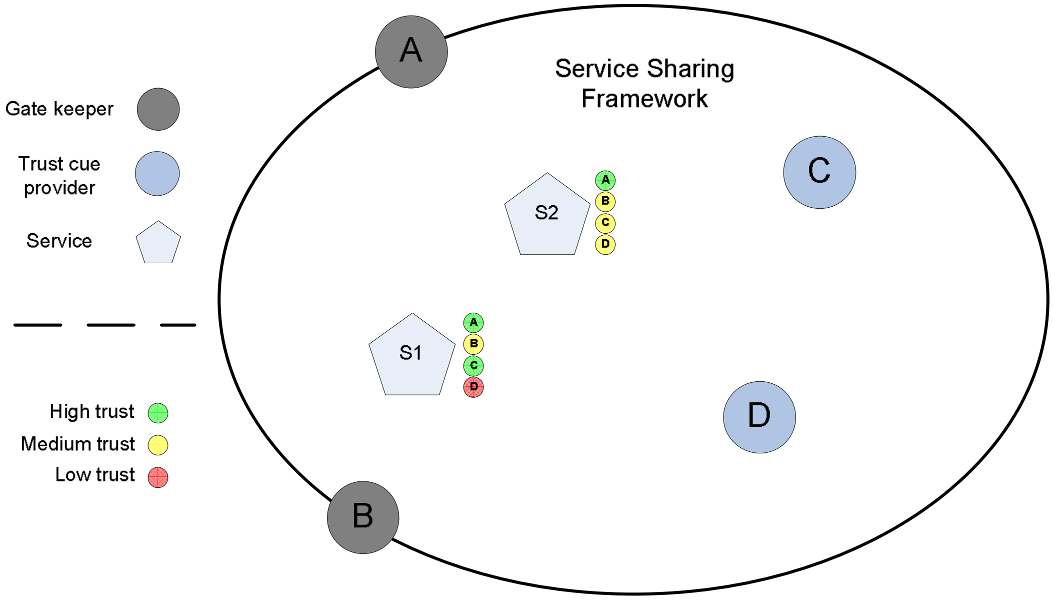
Enabling and enhancing user perceived trust can be a double edged sword. As illustrated by the following quote from Baier (1986) trust can be used to the detriment of users: “Criminals, not moral philosophers, have been the experts at discerning different forms of trust. Most of us notice a given form of trust most easily after its sudden demise or severe injury. We inhabit a climate of trust as we inhabit an atmosphere and notice it as we notice air, only when it becomes scarce or polluted”. Hence an absolute requirement of any trust enabling mechanism designed within the project is that it should not, in any way, willfully give users false impressions about the trustworthiness of services. Friedman (2000) also expresses concern for this dimension of trust as illustrated by the quote: “Trust depends not only on assessing harm and good will but what to reasonably expect of the technology. Yet designers routinely use misleading language and images to convey to users greater reliability and security in the technology than is warranted”. The decision to trust or not to trust always rests in the users hands and should not be unduly *manipulated* by trust mechanisms.

Our propensity to trust is something highly individual, some people are more trusting or risk prone than others. Taking a relevant example some people seldom worry about what they download or which of their actions are logged while being engaged in an online transaction while others have developed an almost paranoid attitude. Hence it is unlikely that all users would feel comfortable with the same set of trust mechanisms. Overly restrictive security policies that attempt to instill trust by forcing everyone to “do the right thing” often fail as people tend to start breaking the rules because they get in their way. A better approach is to leave the choice of trust mechanisms and their configuration to users themselves as far as possible. In addition users would become more familiar with their own trust mechanisms and be able to make more well-informed decisions.

Having reviewed a portion of the available literature on trust one might ask how this can help our work on trust in micro service environments. As illustrated by this survey, and others, there is a plethora of trust and trust framework definitions. It seems unlikely that a consensus on this matter will be reached in the foreseeable future. In addition it seems reasonable to embrace the fact that people are individually different with regards to trust and risk propensity. Hence instead of adding to the collection of theories and all encompassing trust frameworks, EPTMSE should focus on how we can design for trust with a diversity of trust definitions and individual differences in mind. Some ideas for how that could be accomplished are presented below.

One possible way to create such an open ended trust mechanism (or framework), is to focus on personalized trust mechanisms. Such mechanisms allow users to control how they operate at least to some extent. Personalization is an important issue from an end user perspective as people tend to be more or less risk aversive. What is a perfectly acceptable risk for some users may be a totally unthinkable risk for others. People also tend to emphasize different things when deciding to trust or not. For instance, when deciding whether to install a piece of software or not some people rely wholly on their friends’ opinions while others trust experts’ opinions more. Catering for all these different *trust styles* is important if we wish services that we create to have any diffusion. Accomplishing this with a single monolithic trust mechanism seems undoable.

Instead, grounded in the concept of micro-services, an alternative would be to have a highly customizable *meta-trust service*. The basic idea is that users define their own *trust metric* supported by *trust cue providers* that can be plugged into the meta-service. Trust cue providers would be just another type of service in SSF terms or alternatively could lie outside the SSF. Additionally providers could be components supplied both by SSF administrators as well as components developed by users themselves. Users choose which available trust cue providers they would like to use for their personalized trust metric or create new ones to suit their needs (some providers might actually be mandatory). In addition users should have a means of assigning relative importance to the different cues making up their metric thus allowing for personalization.



Providers can work at both a system level and a personal level. We envision that system level providers would run on the server side[4] and e.g. inspect services injected into the system and act as gatekeepers (e.g. a virus scanning service), keeping unwanted services out of the system. Alternatively services could be tagged by the gatekeepers letting potential users know that there is a high probability that the service is untrustworthy in some way and that they are taking a great risk in using it. System level providers could be mandatory. If they are not they would at least provide trust cues to all users in the system. In contrast personal providers would run locally on users' devices and provide cues only to one user[5]. Personal providers would cater for the more "personal" aspects of trust. For instance some users may find interface design to be an important trust factor, or "code quality" (conformance to design patterns etc). How to extract information about these aspects is for the user to decide but could range from wholly manual extraction (i.e. it is done by a human) to fully automatic. Automating as much as possible of the trust cue provisioning seems like a good idea as users often are reluctant to provide feedback manually.

Most users in an SSF will in all likelihood not be service creators let alone trust cue provider creators. However, in an SSF sharing trust providers is no different from sharing any other services meaning all users could still benefit from providers created by someone else. However, some providers may take a long time to run or require large amounts of (system) resources. Hence it would sometimes be desirable to somehow share the cues themselves instead of the providers. How to accomplish this is unresolved but several methods come to mind including tagging services with cues or collecting cues in a distributed database. As a

[4] To the extent that servers exist in P2P networks
[5] Or a group of users depending on how the provider and the network is implemented

result users would have more cues to choose from without having to host all of them. Both methods mentioned above have pros and cons which need to be investigated further.

Possible cues that may prove to be useful as trust indicators include:

- Number of found bugs in a service
- Number of bugs introduced by a service creator (how good a programmer is s/he)
- Visual appearance (aural too perhaps)
- Code quality
- Reputation (service and service creator)
- Fame (Karl Lagerfeld is famous I am not)
- Consequences (see below)
- Match new services against profiles of known ones to understand a bit what they are about (system provider)
- Use virtual characters as a medium for trust cues

Assessing the possible negative consequences of trusting something is essential to the trust equation. Hence it would be beneficial to have some way of automatically estimating possible negative consequences of using a service. This would allow users to reason along the lines “this service does not seem trustworthy as it is badly designed but the possible consequences are negligible so I’ll try it anyway” or vice versa. How to automatically estimate consequences (if it is at all possible) need further investigation. In addition caution is necessary when introducing a feature that can make users overconfident.

An important balance needs to be struck between traceability of users and their privacy. From a system perspective providing services such as recommender systems for services as well as service creators would be beneficial but could infringe on privacy.

Seen from a reputation management perspective the proposed individual trust provisioning framework is somewhat analogous to virtual raters in recommender systems. These raters have some algorithm according to which they rate items. Virtual raters are a way of making it possible to recommend items before anyone (a real person that is) has rated them.

5 References

Cited references

Aberer, K., Despotovic, Z. (2001), Managing Trust in a Peer-to-Peer Information System, Proceedings of the tenth international conference on Information and knowledge management, pp. 310-317.

Baier, Annette (1986), Trust and Antitrust, Ethics, Vol. 96, No. 2, pp. 231-260

Brenkert, C. G. (1998), *Trust, morality and interpersonal business*, Business Ethics Quarterly, Vol. 8, No. 2, pp. 293-317.

Clark, D. (2001), Face-to-Face with Peer-to-Peer Networking, IEEE Computer, January 2001

Corritore, C. L., Kracher, B., Wiedenbeck, S. (2003), On-line trust: concepts, evolving themes, a model, In *International Journal of Human-Computer Studies*, No. 58, pp. 737-758

Dunn, J. (1984), The concept of trust in the politics of John Locke, In R. Rorty, J. B. Schneewind, and Q. Skinner (eds.), *Philosophy in History*, Cambridge University Press.

Friedman, B., Kahn, P., Howe, D.C. (2000), *Trust Online*, In Communications of the ACM, Vol. 43, No. 12, pp. 34-40.

Gambetta, Diego (2000) ‘Can We Trust Trust?’, in Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition, Department of Sociology, University of Oxford, chapter 13, pp. 213-237, <http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf>.

Lewis, J. D., Weigert, A. (1985), Trust as a Social Reality, Social Forces, Vol. 63, No. 4, pp. 967-985

Lewicki, R.J., Bunker, B.B. (1996), *Developing and maintaining trust in work relationships*. In Kramer, R., Tyler, T. (eds.), Trust in Organisations: Frontiers of Theory and Research. Sage, pp. 114-139.

Luhmann, N. (1979), *Trust and Power*, Chichester, Wiley.

Marsh, S., (1994) Formalising Trust as a Computational Concept, PhD Thesis, Department of Computing Science and Mathematics, University of Sterling

Resnick, P., Zeckhauser, R., Friedman, E., Kuwabara, K. (2000), Reputation Systems, Communications of the ACM, Vol. 42, No. 12, pp. 45-48.

Selcuk, A. A., Uzun, E., Pariente, M. R. (2004), A Reputation-Based Trust Management System for P2P Networks, International Workshop on Global and Peer-to-peer Computing, IEEE/ACM CCGRID 2004.

General bibliography

Lundgren Kurt (2003), En lärande IT-Politik för tillväxt och välfärd,

Blaze, M., Feigenbaum, J., Lacy, J. (1996), Decentralised Trust Management, Proceedings of IEEE Conference of Security and Privacy

Jalava, J. (2003), From Norms to Trust – The Luhmanian Connections between Trust and Systems, European Journal of Social Theory, Vol. 6, No.2, pp. 173-190

Gray, E., O’Connel, P., Jensern, C., Weber, S., Seigneur, J., Yong, C., (2002) Towards a Framework for Assessing Trust-Based Admission Control in Collaborative Ad Hoc Applications, Technical Report 66, Department of Computer Science, Trinity College, Dublin Ireland.

Schneiderman, B. (2000), Designing Trust into Online Experiences, Communications of the ACM, Vol. 42, No. 12, pp. 57-59

Palen, L., Dourish, P. (2003), Unpacking “Privacy” for a Networked World, In proceedings of CHI 2003, Vol. 5, No. 1, pp. 129-136

Preece, J. (2004), Etiquette, empathy and trust in communities of practice: Stepping-stones to social capital, In *Journal of Universal Computer Science*, Vol. 10, No. 3.

Massa, P., Bhattacharjee, B. (2004), Using Trust in Recommender Systems: an Experimental Analysis, iTrust 2004.

Feng, J., Lazar, J., Preece, J. (2004) Empathic and predictable communication influences online interpersonal trust, Behavior and Information Technology (accepted, in press).

Lee, S. M., Lee, S., Yoo, S. (2004), An integrative model of computer abuse based on social control and general deterrence theories, Information & Management, No. 41, pp. 707–718

Ren, K., Li, T., Wan, Z., Bao, F., Deng, R. H., Kim, K. (2004), Highly reliable trust establishment scheme in ad hoc networks, Computer Networks, In press.

Liu, C., Marchewka, J. T., Lu, J., Yu, C. (2004), Beyond concern—a privacy-trust-behavioral intention model of electronic commerce, Information & Management (In press).

Dunn, P. (2000), THE IMPORTANCE OF CONSISTENCY IN ESTABLISHING COGNITIVE-BASED TRUST: A LABORATORY EXPERIMENT, Teaching Business Ethics, No. 4, pp. 285-306

Oliviero, N., Lunt, P. (2004), Privacy versus willingness to disclose in e-commerce exchanges: The effect of risk awareness on the relative role of trust and control, In *Journal of Economic Psychology*, No. 25, pp. 243-262.

De Vries, P., Midden, C., Bouwhuis, D. (2003), The effects of errors on system trust, self-confidence, and the allocation of control in route planning, In *Int. J. Human-Computer Studies*, No. 58, 719-735.

Grabner-Kräuter, S., Kalusha, E. A. (2003), The effects of errors on system trust, self-confidence, and the allocation of control in route planning, In *Int. J. Human-Computer Studies*, No. 58, 783-812.

Johnston, J., Eloff, J. H. P., Labuschagne, L. (2003), Computers & Security, Vol. 22, No. 8, pp. 675-685.

Zolin, R., Hinds, P. J., Fruchter, R., Levitt, R. E. (2004), Interpersonal trust in cross-functional geographically distributed work: A Longitudinal study, Information and Organisation, No. 14, pp. 1-26.

Kim, D. J., Song, Y. I., Braynov, S. B., Rao, H. R. (2004), A multidimensional trust formation model in B-to-C e-commerce: a conceptual framework and content analyses of academia/practitioner perspectives, Decision Support Systems, XXX

Fam, K.S., Foscht, T., Collins, R. D. (2004), Trust and the online relationship—an exploratory study from New Zealand, Tourism Management, No. 25, pp. 195-207.

Hwang, P., Burgers, W. P. (1997), Properties of Trust: An Analytical View, ORGANIZATIONAL BEHAVIOR AND HUMAN DECISION PROCESSES, Vol. 69, No. 1, pp. 67–73.

Vlachos, V., Androutsellis-Theotois, S., Spinellis, D. (2004), Security applications of peer-to-peer networks, Computer Networks, Vol. 45, pp. 195-205.

Chivers, H., Clark, J. A. (2004), Smart dust, friend or foe?—Replacing identity with Configuration Trust, Computer Networks, XXX

Hertzum, M., Andersen, H. H. K., Andersen, V., Hansen, C. B. (2002), Trust in information sources: seeking information from people, documents, and virtual agents, Interacting with Computers, Vol. 14, pp. 575-599.

Trcek, D. (2004), Towards trust management standardization, Computer Standards & Interfaces, XXX.

Jøsang, A., Presti, S. L. (2004), Analysing the Relationship between Risk and Trust, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 135–145, 2004., Springer-Verlag.

Cofta, P. (2004), Computing Recommendations to Trust, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 340–346, Springer-Verlag.

Djordevic, I., Dimitrakos, T. (2004), Towards Dynamic Security Perimeters for Virtual Collaborative Networks, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 191–205, Springer-Verlag.

English, C., Terzis, S., Waegella, W. (2004), Engineering Trust Based Collaborations in a Global Computing Environment, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 120–134, Springer-Verlag.

Baldwin, A. (2004), Enhanced Accountability for Electronic Processes, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 319-332, Springer-Verlag.

Liu, J., Issarny, V. (2004), Enhanced Reputation Mechanism for Mobile Ad Hoc Networks, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 48-62, Springer-Verlag.

Orbreiter, P. (2004), A Case for Evidence-Aware Distributed Reputation Systems, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 33-47, Springer-Verlag.

Bussard, L., Molva, R., Roudier, Y. (2004), History-Based Signature or How to Trust Anonymous Documents, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 78-92, Springer-Verlag.

Jonker, C. M., Schalken, J. J. P., Theeuwes, J., Treur, J. (2004), Human Experiments in Trust Dynamics, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 206-230, Springer-Verlag.

Signeur, J. M., Jensen, C. D. (2004), Trading Privacy for Trust, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 93-107, Springer-Verlag.

Ishaya, T., Mundy, D. P. (2004), Trust Development and Management in Virtual Communities, C.D. Jensen et al. (Eds.): iTrust 2004, LNCS 2995, pp. 266-276, Springer-Verlag.