3

A Taxonomy for Trusted Services

Jon Ølnes
Norwegian Computing Centre (NR), P.O. Box 114 Blindern, N-0314 Oslo, Norway[2]

Abstract: Electronic commerce must be trustworthy. This includes (technical) trust in the computerised systems and networks, and (organisational) trust in the honest intent of the counterparts. To establish trust, in general one has to rely upon infrastructures consisting of trusted services. In particular for organisational trust, a plethora of different services may be needed. This paper suggests a taxonomy to characterise trusted services in general. The virtues of a taxonomy are easier comparison and a common terminology, increased understanding, and facilitation of tasks like requirements specifications.

1. INTRODUCTION

In order to gain acceptance, electronic commerce must be trustworthy. Trust can be defined as "perceived lack of vulnerability". A trust decision implies a judgement about the vulnerability implied by a certain action, and thus a decision to carry out the action or not. The entity that decides upon this acceptance is ultimately always a human or a humanly controlled entity. As an example, the EU work on "qualified electronic signatures" and related certificates [2] is really about setting the requirements that digital signatures should fulfil in order to be trusted by the (humanly controlled) entities "EU" and "member state".

Trust decisions are not necessarily rational. Trust is a subjective decision, based on perceived, not real, vulnerability. Enough examples may be found where something (press coverage, lobbying organisations) blows minor vulnerabilities out of proportions, or alternatively attempts to turn severe vulnerabilities into trifles. Compiling available information, weighted by common sense and a sound scepticism towards the information, into rational trust decisions is a difficult task.

[2] Present affiliation: PKI Consulting Services AS, P.O.Box 1569 Vika, N-0118 Oslo, Norway. (Email: Jon.Olnes@pki.no) Also part-time associate professor, University of Tromsø.

Trust may be established one-way (I trust my bank but my bank does not trust me) or mutually (we trust one another).

An electronic commerce arena may consist of large-scale systems (or rather large-scale connectivity between systems) and in principle arbitrary communication patterns. The actors may have no prior knowledge of one another, and thus no way to determine the trust to take in a counterpart. This calls for an infrastructure consisting of trusted services, commonly termed TTP-services (trusted third party). A plethora of different trusted services may exist for different purposes.

In this paper, the term TTP is used as a shorthand for any trusted service. The term is sometimes slightly misleading, as a trusted service need not always be provided by a neutral, *third* party. As one example, banks are usually trusted to take on several TTP roles for electronic commerce, even if the banks are highly involved in the financial transactions that result.

This paper focuses on a taxonomy to characterise trusted services and their roles, not that much on the topic of trust per se. There is ample literature on more or less formal trust metrics and reasoning about trust. This is discussed briefly in section 5 but in general the topic of formal trust reasoning is out of scope of this paper.

The virtues of a characterisation are an increased understanding of the roles of TTPs, better means to analyse the properties of the services, easier comparison between services, and facilitation of tasks like requirements specifications. The approach is more engineering-style than formal. The characteristics proposed are:

- service offered and type of trust mediated – technical or organisational (see 2);
- quality of service – as specified by the TTP's policy statements;
- proof handling – production, validation or storage of proofs;
- community of users;
- trust model – with respect to other TTP services;
- legal aspects, jurisdiction, responsibility and liability taken, need for agreements;
- communication pattern – on-line, off-line or in-line service, human user interface and programming (API) interfaces.

The characteristics are discussed in sections 2-7. Section 8 has a brief discussion on the role of licensing and certification. Section 9sums up the taxonomy in the form of a table. An example of an application of the taxonomy would have been beneficial, but a paper format unfortunately leaves no room for this.

2. TECHNICAL AND ORGANISATIONAL TRUST

Fundamentally, there are two different types of trust for electronic communication:

– *Technical trust*[3] in a computerised system and its components, i.e. that the system works as anticipated (reliability), is protected against attacks (security), and protects the interests of the user (safety).
– *Organisational trust* in the honest intent and willingness to co-operate of other actors / users of the system.

This is shown in *Figure* . Technical trust is in what Jøsang [5] calls "rational entities", computers and the like that behave according to programmed instructions. The most important property of rational entities is their security, i.e. that they have not been compromised. For the purposes of this paper the security term also includes reliability and safety. Organisational trust is in "passionate entities" in Jøsang's terms, i.e. entities that may behave according to will. This is related to questions like "will this person pay for the services" or "is this a serious dealer".

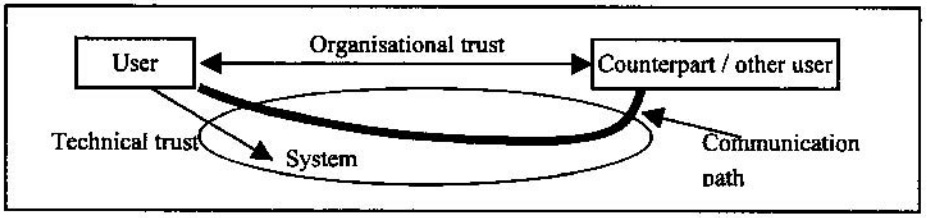

Figure 1. Technical and organisational trust.

Going back to the definition of trust as "perceived lack of vulnerability", it is clear that two different properties are important:
– The system / counterpart must be perceived as trustworthy; and
– The system / counterpart must really be sufficiently trustworthy.

These properties are to a certain degree independent. A rogue counterpart may have a very convincing appearance, while an honest one need not appear that way. It is perfectly possible to give the impression that an insecure system is secure, and a secure system does not necessarily give that impression. There may also be a conflict with respect to making security properties transparent and at the same time give an impression of security. The user interface and friendliness of a system clearly plays an important role in this context.

A limited number of actors can exchange information prior to communication, and can establish *direct* trust relationships by such ad hoc means. It is impossible to pre-establish such trust relationships for communication between a large number of (in principle arbitrary and unknown) actors. The general solution is to define some services as trusted, and derive trust between other actors from the trust in these services. In this, trust is regarded as a transitive property – we do not trust one another, but since both of us trust the TTP that vouches for the other party, we can still establish *indirect* trust. If we do not use the same TTP, we either have to obtain direct trust in more than one TTP, or the TTPs must apply a trust model that allows

3 The terminology technical / organisational trust is suggested by the author.

us to obtain indirect trust in the other party's TTP, resulting in a *chain of trust* between us. This is discussed further in 6.

The TTPs constitute an infrastructure. The two types of trust give rise to different kinds of TTP-services, as shown in *Figure* . TTPs for technical trust enable secure communication between possibly arbitrary actors. TTPs for organisational trust enable co-operation on presumably important matters between possibly arbitrary actors.

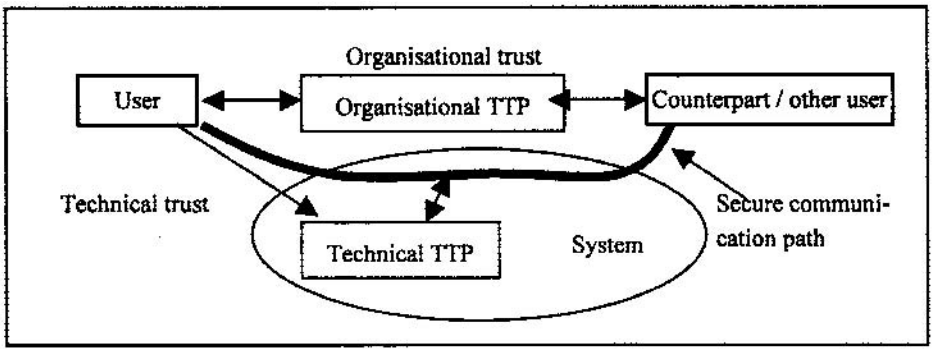

Figure 2. Technical and organisational TTPs.

2.1 Technical Trust and Security

No system is 100% secure. Determination of a "sufficient" level of security will vary from case to case dependent on the requirements. Too low *technical* trust means that a system cannot be used or that communication cannot take place. Technical TTP-services provide a certain guarantee that a sufficiently secure communication channel may be established. The most common example is a certificate authority that issues authentication certificates, which in turn may be used both for authentication and establishment of a secure communication channel.

Technical TTP-services, a certificate authority is a good example, may contribute to organisational trust by providing knowledge for the user's trust decision. However, it is a frequent mistake to assume that authentication implies trust. Knowing your counterpart does not necessarily make you trust him.

2.2 Organisational Trust and Accountability

Reflections about what one wants to communicate *about* (or co-operate on) with a counterpart that one communicates *with,* are commonplace in daily life. The answers are given by the level of trust taken in the counterpart. A low level of *organisational* trust implies that co-operation cannot take place, or alternatively that strong accountability measures must be in place in order to prove and prosecute malicious behaviour. For accountability event logging may often be sufficient.

Stronger accountability services, i.e. non-repudiation, can only be achieved by use of digital signatures and / or TTPs for storing of proofs.

For electronic commerce, visual impression should not imply trust. A backstreet store in a large city gives an impression of the risk of buying something there. For its electronic commerce service, the same store may have a totally different appearance.

Organisational TTPs are commonplace in everyday (physical) life, although one does not normally reflect on the necessity of such actors. The textbook example of a commerce TTP actor is a broker that mediates trade between parties. Electronic broking is an active research area.

Organisational TTPs are not needed in an infrastructure for secure *communication.* But the TTP services are needed – or at least desired – to obtain trustworthy electronic *commerce* (given a broad interpretation of this term). In this aspect, the organisational TTPs are parts of value chains for electronic commerce. Organisational trust and organisational TTP roles are generally not well defined nor understood in today's electronic commerce. As of today, a reasonable assumption seems to be that where a TTP role exists in "traditional" commerce, its electronic counterpart may also be needed, with approximately the same purpose. Only experience will show if this picture is correct, or if roles will appear, disappear or get new content if eventually electronic commerce will develop to something entirely different from traditional commerce.

3. PROOF HANDLING – THE PURPOSE OF A TTP

A TTP plays a role with respect to the knowledge upon which an actor decides a level of trust. The TTP *produces, validates,* or *stores* proofs of statements. Examples of statements are: "I am NN", "I have the right to charge bank account xxxxx", "I have sent message M at time T", "I intend to pay for the goods that I just ordered", or "I run a trustworthy business". It follows that "proof" may be a bit strong word in some cases – a better term might be "plausibility". A TTP may be delegated the responsibility for producing certain information, e.g. cryptographic keys. An important characteristic of a TTP is its role in proof handling, e.g.:

– A certificate authority *produces* electronic IDs, i.e. proofs of identity[4];
– A notary service *stores* proofs related to certain documents or actions;
– An OCSP (On-line Certificate Status Protocol) [10] service *validates* certificates and returns their status (valid, suspended, revoked, etc.).

[4] The electronic ID is only part of the proof. It is only valid when accompanying some piece of information that proves that the originator is in possession of the correct private key.

Proofs can be suitable for human evaluation, or they may be meant for automatic processing by programs and services. This may pose entirely different requirements with respect to representation of the proofs.

4. COMMUNICATING WITH A TTP

A TTP will one way or another be involved in the communication between the actors. Depending on its role in the communication protocol a TTP is denoted as:

- Off-line – does not participate in the communication, but the actors rely on the TTP having produced the necessary proofs in advance;
- On-line – the actors communicate directly, but at least one of them must communicate with the TTP during the communication session, at the time of session establishment or later;
- In-line – all communication between the actors passes through the TTP.

This is shown in Figure 3. The most common example of an off-line TTP is a certificate authority for authentication certificates. An example of an on-line TTP is an OCSP [10] service for validation of certificates. An example of an in-line TTP is a service that provides anonymity. A broker may also operate an in-line service. Even an off-line TTP may offer on-line services, but these services are not necessarily trusted. One example is an on-line certificate directory, which may or may not be a trusted service.

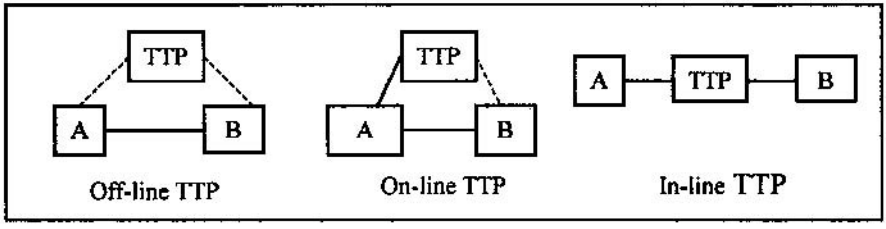

Figure 3. Communicating with a TTP.

A TTP must offer interfaces towards its users. This may be interfaces intended for human users, protocol interfaces that can act as the server side for defined communication protocols, or application program interfaces (API) related to use of component based systems and middleware. An interface may be message based, i.e. well-defined messages are exchanged between the user and the TTP, or call-based, usually by means of remote procedure calls. For on-line and in-line TTPs, the need for interfaces is obvious. An off-line TTP needs to offer interfaces in order to handle requests for production of proofs, like certificates, and for adjunct on-line services, like a directory, if offered.

5. QUALITY OF SERVICE AND QUALITY OF TRUST

An actor's subjective trust evaluation decides whether or not a certain action may be performed, perhaps under certain additional requirements like accountability. When a TTP is involved, the first decision is to what degree the TTP is trusted, and secondly to what degree actors that the TTP vouches for can be trusted. A decision to trust a TTP implies access to all actors / services for which trust is mediated by this TTP. It does not necessarily imply trust in all such actors, but distrust in the TTP implies distrust in the actors.

There is ample literature on approaches to metrics, based on various formal logics, for analysis of trust properties, see for example [1], [6], [7], [8]. Reiter and Stubblebine [8] provide a comparison of some approaches and suggest design principles for metrics. Most papers are particularly targeted at authentication and certificate chains, but the approaches can in most cases be generalised to other trusted services and trust models. Formal reasoning is valuable for a thorough analysis, probably mainly from a system engineering viewpoint. The benefit with respect to trust decisions taken by ordinary users is marginal. Since this paper is about characteristics of trusted services, and not trust per se, a further discussion on formal approaches to trust modelling and metrics is out of scope.

In the context of this paper, an important question is how, in a practical sense, an actor can establish a level of trust in a TTP. The trust is based on knowledge and assumptions (which will always carry an element of uncertainty). In practice, the most important parts of the knowledge are probably the actor that is responsible for the TTP, the name ("brand") of the service, and references to (recommendations from) other actors that trust the service.

Following these characteristics, the quality of the service comes next in importance. To enable determination of the quality of service, a TTP needs to have the following basic properties:

- Available policy statements that give a clear indication of the quality level and other aspects of the service like liabilities – a certificate policy is one example;
- An implementation that fulfils the level given by e.g. the certificate policy – conformance may be backed by third-party evaluation (see 8);
- Frequently objectivity with respect to the actors it serves.

The last property is not always necessary. As one example, a bank may take trusted roles for electronic commerce even if the bank is highly involved in the financial transactions that take place. In such cases, the term Trusted *Third* Party is slightly misleading, but we will nevertheless keep the term TTP for any trusted service, even provided by one of the commuicating actors.

Figuring out the quality of a particular TTP service by simply reading its policy will usually not be practical. A policy (most certificate policies may serve as excellent examples) may be a complex and rather impenetrable document, even to

people with reasonable competence in the problem area. Policies may even be written in a language unknown to the reader, and may refer to laws and regulations that belong to an unknown jurisdiction.

For practical purposes, quality of service should refer to a limited number of discrete values. An example may be different classes of authentication certificates, like Verisign[5] certificates of classes 1-3 (increasing quality). Here, the level is simply claimed by the service provider. Instead, more objective criteria may be established, where a provider may either give a self-assessment about compliance with a certain level, or some kind of certification or licensing may be given (see 7). "Qualified certificate" [2] is one such objective level indicator.

Getting hold of the quality indicator is the next problem. The EU Directive mandates a "qualified" indicator in the certificates themselves, but in most cases certificates will only have a reference to the policy, that is still impenetrable.

This may be solved by trusted validation services, which can be queried for information about other TTP services. One example may be a service that receives certificates issues by "any" certificate authority, and returns an authoritative answer about the validity of the certificate in question along with information about its quality level. This is discussed a bit further in 6.3.

Another approach is taken by the US Federal PKI specifications[6] , which include a bridge certificate authority that will (voluntary agreement) cross-certify with the certificate authorities that serve the Federal administration. The bridge defines a number of discrete quality levels, and indicates, in a cross-certificate, the correct policy mapping (see 6.2) towards the policy of the service in question.

The quality of a service does not automatically indicate the trust level, but the quality level is an important characteristic for the taxonomy. The quality level is one more piece of information upon which quality of trust can be based. An actor may trust services of documented equal quality to different degrees, e.g. decide to trust only certain issuers of qualified certificates.

6. TRUST MODELS, RELATIONSHIPS BETWEEN TTPS

6.1 Trust Models and Scaling

Users will select certain TTPs that they decide to trust. Of course, other users may select other TTPs offering the same services. Requirements for communication between users of different TTP services are evident. This calls for trust models involving several TTPs. Three trust models can be seen:

[5] http://www.verisign.com

[6] See http://csrc.nist.gov/pki

- Monolithic, or trust list, i.e. only completely separate services – a user must establish trust separately in each of the TTPs;
- Hierarchies, where a TTP is approved (e.g. has a certificate issued by) a TTP at a higher level, and so on through possibly several levels back to a trusted root;
- Web of trust, where pairs of TTPs mutually (one-way is possible, but not common) recognise one another – e.g. through cross-certification,

A monolithic structure does not scale. Similarly, a web of trust structure involving a large number of TTPs will be unmanageable. Hierarchies have well-known scaling properties, but in this context even a hierarchical structuring has its problems. It is fairly clear that one will not end up with a situation where all TTP services of a particular kind are members of one common hierarchy. One is always left with an element of a monolithic trust model, in the form of direct trust in several trust structures.

Trust structures are nevertheless useful. As one example, Norwegian banks develop specifications for a common electronic ID service called BankID. This is a considerable simplification with respect to the situation where each bank has its own solution.

A hybrid structure is formed by combining a hierarchical structure with a web of trust structure. Whether this is allowed or not, and in case at which level in the hierarchy, is decided by the policies in force. Mutual recognition at the root level will effectively chain complete hierarchies, while mutual recognition at lower levels will chain sub-trees or single services. Finding a trust path between actors in a hybrid model (or even in a large-scale web of trust) is very difficult in practice.

6.2 Trust Models and Quality Level

Neither a web of trust model nor hierarchies necessarily mean a consistent quality level of the TTPs involved. The approval represented by e.g. cross-certification may be related only to an assurance that the other TTP runs according to its specified policy, whatever its quality. Such models enable recognition of other TTP services, and processing (e.g. certificate processing) related to the service, but users still have to determine separately the quality of all TTPs.

Rather, a user wants an indication that a given TTP has at a well-known quality, even if the TTP is only indirectly trusted. For web of trust this is achieved by "policy mapping", which implies a mutual recognition that the services are compatible. Within a hierarchy, a consistent policy level is obtained by posing requirements on the policies. A TTP that is not a leaf node of the hierarchy will postulate policy requirements that a TTP at lower levels must adhere to in order to become a member of the hierarchy.

As stated in 5, an actor may not necessarily take an equal trust in all services of a given quality level. Furthermore, the length of the trust chain becomes a new parameter in the trust determination. But more often than not, a user will accept a

service at a well-known quality level if it is a member of a trust structure that the user recognises.

Deep hierarchies, as originally suggested by the PEM specifications [9], are discouraged because of the length of the trust chains and the time-consuming processing. The trust model is one of the main reasons for PEM's failure.

The present direction is towards "shallow hierarchies". Below a root, that typically determines (the level of) the policies in the hierarchy, there is one level of TTPs that in turn service the users. Thus long chains are avoided, and finding a path is easy. The Norwegian BankID project is one example of such a structure, with one CA per bank under a common root. Identrus[7] , an initiative taken by some of the world's largest banks, potentially adds one more layer. The Identrus root-CA issues certificates only for the large banks (level 1), while smaller banks must obtain certificates from level 1 CAs, which additionally serve customers directly.

6.3 Meta–TTPs

Even given trust structures, a user is faced with a large number of TTPs that the user must decide to trust or not for a given purpose. An approach to solving this is meta-TTPs that answer requests about the quality and other aspects of other TTPs.

An example is an on-line certificate validation service. When receiving a certificate, a user will, without any processing of his own, ship the certificate off to the validation service. This service returns the status of the certificate (valid, revoked, suspended, expired) and possibly extra information derived from the certificate, like the values of certain fields or attributes. It may also return a quality level indicator. (As discussed in 5, the quality of TTP services should preferably be categorised into a limited number of discrete levels.) Based on the response, the user will decide to trust or not in order: the issuer, the certificate, and the actor for which the certificate has been issued. A particular validation service may not know everything, but as with all other TTP services, it may be part of a larger trust structure.

Conceptually, this may be regarded a two-level trust hierarchy, with the meta-TTP as the root, and the services it answers for at the second level. A meta-TTP must rely on a registry of services with given characteristics, compiled by the meta-TTP itself, or by other sources. According to the EU Directive [2] issuers of qualified certificates must be registered. The same goes for various TTP-roles that require a license in order to operate.

The interesting feature is that users may request information related to TTP services that they know nothing of, without referring to any kind of trust structure. A user may send any certificate to a validation service, regardless of who the issuer of the certificate is, and get back the information the user needs for his trust decision.

[7] http://www.identrus.com

The ultimate situation is that this may make all other trust structures void, since trust in TTPs may either be direct, or indirect through a meta-TTP.

7. LEGAL ASPECTS, LIABILITY, AGREEMENTS, USERS, PAYMENT

Part of the quality of a TTP is the degree of certainty of the information supplied or handled by the service. If a user suffers damage because of a mistake or failure by a TTP, severe legal implications may result. Any TTP needs to take precautions, and the legal conditions for use of the service should be clearly stated in the policy. The first step here is identification of jurisdiction and applicable law. Additionally, the TTP will impose limitations on the liability taken in case of failures, usually in the form of statements like not accepting liability if the service is used for a transaction above a certain value, or if the user's actions imply carelessness or violate the TTP's policy.

Relevant laws and regulations may to some extent dictate the liabilities that the TTP must take, and other aspects of its operation. Other issues here are the need for a license in order to be allowed to operate the service, and compliance with laws in areas such as privacy.

While statements like "use of the service implies that one has accepted the conditions stated in the policy" are commonplace, explicit formal agreements between a TTP and other actors are usually also needed. As one example, a certificate authority may require a signature on a written agreement before issuing a certificate to a person, even if the contents of the agreement are covered by the certificate policy. Agreements are also needed towards other TTPs in common trust structures [8], and with actors that somehow assist in the provision of the service. Examples here are registration authorities assisting certificate authorities and outsourcing of parts of services.

A common problem for TTPs is that no agreement need to exist with a party that relies on the TTP's proofs. As one example, an actor that receives a certificate issued by a given certificate authority will in general not have an agreement with the certificate authority. Liabilities towards such relying parties with respect to mistakes by the TTP should be covered by the TTP's policy, but this is nevertheless a difficult legal area.

The legal environment is one important parameter for identification of the customers of the service, and may be especially important if an international market is targeted. However, commercial issues will usually be more important – which market segments will the TTP aim at in order to make a profit out of the operation?

[8] Experience shows that the complexity of the legal aspects of cross-certification may make the process almost prohibitive.

Is the service open to anyone, or is it accessible only to a restricted community? The latter points directly at another important question: how is payment settled for use of the service? Several models are possible here, from subscription fee via per use fee (with a variety of payment methods) to free use. A discussion of payment models is outside the scope here.

8. LICENSING, EVALUATION, CERTIFICATION

Certain organisational roles require a license to operate. Examples are lawyer, medical practitioner, bank, real estate broker, and numerous others. Frequently such roles will be of a TTP type. What we see is really a trust hierarchy. TTPs at higher level certify the rights and credibility of TTPs at the leaf level by issuing a license for a certain role. Electronic license certificates, e.g. to licensed lawyers, should be issued for easy accessibility to other actors.

A license may require an evaluation and certification procedure. The roles, and the trust structure, of such a system are shown in *Figure* . A license granting body[9] is in charge of licensing of the actors that may perform evaluations according to certain criteria, and actors that are entitled to issue certificates of compliance with the criteria. Evaluator and certificate issuer will often be the same actor, but the roles are conceptually different. This is a confidence-building, and thus trust-building, system, making properties like quality and security visible.

There are several standards and systems for certification, with the ISO9000 series for quality as the most well known. The ISO14000 series provides certification with respect to environmental requirements. In security, ISO17799 [4] will be used. Certification may be requested by an actor at its own discretion, e.g. because this will lead to a market advantage, or it may be required by a license granting body in order to obtain a license for a certain role.

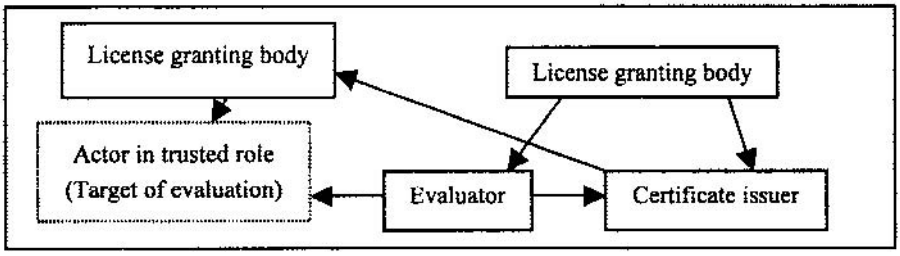

Figure 4. Evaluation, certification, and licensing.

[9] The term “accreditation” is usual for this role, but this is just a special case of licensing.

Similarly, security evaluation of (part of) a technical system may be done, e.g. according to the ISO15408 criteria [3]. This provides a guarantee that the system fulfils certain security criteria, given a certain class of environment. Technical system evaluation and certification may be a prerequisite for certain roles, e.g. with respect to systems used to store or process certain kinds of information.

9. THE TAXONOMY – TABLE FORMAT

Based on the discussions in the previous sections, the following table sums up the suggested taxonomy:

Characteristic	**Parameters**	**Description**
Service type	Name of service	E.g. certificate authority, lawyer, broker, notary service
	Type of trust mediated	Technical/organisational – technical may contribute to both
	Trusted functions	E.g. authentication, payment transfer, key management, safe electronic commerce
	Adjunct services	E.g. directory service for a cert. authority
Quality of service	Policy reference	Policy defines most aspects of the service
	Service level	Derived from policy – preferably one of a discrete number of alternatives
	Owner	Who is behind the service?
	Brand name and references	Brand name for the service. Reference to actors that accept the TTP
Proof handling	Produce/validate/store	Part of protocols between actors
User community	Leaf-TTP or not	Wrt. trust models: Serves users directly or role wrt. other TTPs (e.g. root of cert. hierarchy, or license granting authority)
	Identification of customers	Restricted group or not
	Restrictions / requirements	Only certain application areas or open – depends on legal environment
	Payment model	Who pays, and billing method
Trust model	Trust structures and/or named counterparts	Trust structures or named TTPs that the service relates to, including trust model (hierarchy, web-of-trust etc.)
Legal aspects	Jurisdiction and law	Must be named in policy
	Licenses needed	In order to operate service
	Certifications needed	ISO17799, ISO15408 etc.
	Agreements needed	With users, authorities and other parties
	Liabilities taken	Important for most TTPs and for users
Comm. pattern	Off-line/on-line/in-line	Wrt. to protocols between actors
	Human/user interface	If access directly from human users
	API characteristics	Syncronous or asyncronous, message based or call based, standards and other interface specifications (message formats, protocols, APIs, middleware etc.)

10. CONCLUSION

This paper suggests a taxonomy for TTP services. Such a characterisation may help understanding of the area, may provide guidance when comparing services and may help e.g. in writing of requirements specifications. The services are characterised according to type of service offered including type of trust mediated (in the security of the technical systems or in the trustworthiness of the counterparts that one wants to co-operate with), community of users, trust model (hierarchy, monolithic, web of trust) with respect to other TTPs, quality of service, proof handling (production, storage, validation), and communication pattern (off-line, on-line, in-line TTP).

Acknowledgement. The work on this paper has been carried out as a part of the (Harmonization for the Security of Web Technologies) HARP project, partly funded by the EU under the IST research programme.

11. REFERENCES

[1] T.Beth, M.Borcherding and B.Klein, Valuation of Trust in Open Systems, Proceedings of the 1994 European Symposium on Research in Computer Security (ESORICS '94), 1994.
[2] EU, Community Framework for Electronic Signatures, Directive 1999/93/EC of the European Parliament and of the Council, December 1999.
[3] ISO15408, Evaluation Criteria for IT Security, Parts 1-3, 1999.
[4] ISO/IEC 17799, Information Security Management – Code of Practice for Information Security Management, 2000.
[5] A.Jøsang, The Right Type of Trust for Distributed Systems, Proceedings of the 1996 New Security Paradigms Workshop, 1996.
[6] A.Jøsang and S.J.Knapskog, A Metric for Trusted Systems, Proceedings of the 21st National Security Conference, 1998.
[7] U.Maurer, Modelling a Public-Key Infrastructure, Proceedings of the 1996 European Symposium on Research in Computer Security (ESORICS '96), 1996.
[8] M.K.Reiter and S.G.Stubblebine, Authentication Metric Analysis and Design, ACM Transactions on Information and System Security, Vol. 2, No. 2, pp 138-158, May 1999.
[9] RFC1421-1424, Privacy Enhancement for Internet Electronic mail (PEM), 1993.
[10] RFC2560, Internet X.509 Public Key Infrastructure Online Certificate Status Protocol, June 1999.