An Adaptive System Responsive to Trust Assessment based on Peer-to-Peer Evidence Replication and Storage

Marco Casassa Mont, Lorenzo Tomasi[1], Rebecca Montanari[1]
Trusted E-Services Laboratory
HP Laboratories Bristol
HPL-2001-133
June 6th , 2001*

E-mail: marco_casassa-mont@hp.com, lortom@hplb.hpl.hp.com, rmontanari@deis.unibo.it

evidence management, storage, trust assessment, recommendation system, behavioural monitoring, hybrid peer-to-peer architecture

In the physical world people and enterprises are accountable for their actions. As reputation is more and more a valuable asset, people and organizations retain documents and information for a long period of time as evidence of their behavior and actions.

In today's world, documents are mainly available in a paper-based format and there are mechanisms and infrastructures to manage them as evidence. In a near future this could not be the case anymore because of the growing popularity of the Internet and the shift towards digital documents. Along with many advantages, this will introduce a set of problems, last and not least the management of digital evidence over a long period of time as it involves long-term management of data integrity, long-term confidentiality, long-term identity tracking and long-term storage management.

In this paper we address the problem of storage, integrity and survivability of digital evidence within an enterprise, in the context of an Evidence Management Service. We introduce a peer-to-peer aspect to take advantage (in terms of storage and processing) of cheap and abundant resources (like PCs) available within medium and large enterprises. We describe a hybrid peer-to-peer architecture mitigated by the addition of a centralized trusted control component. The system is adaptive to the behavior of the peers since it is responsive to the assessment of their trustworthiness and reliability. We illustrate a few relevant use cases.

* Internal Accession Date Only
Approved for External Publication
[1] DEIS Department, University of Bologna, Bologna, Italy
© Copyright Hewlett-Packard Company 2001

1. Introduction

In the ordinary world, people and organizations are accountable for their actions and behaviors.

It is a basic aspect of the human nature to rely on any kind of evidence to support a particular point of view or thesis. For example, in the business environment, enterprises use paper-based documents to give explanation for their actions and decisions. These documents traditionally contain information about business-related events and multi-party interactions, like transactions and contracts.

In some cases, both people and organizations are requested to preserve as evidence particular kind of documents (records of activity that generates intellectual property, trading accounts, tax forms, receipts, deeds, wills, etc.), for a very long period of time. Usually these documents have a legal course: they are signed by the involved parties and notarized. Should any kind of problem arise, these documents can be used to settle a dispute either directly between the involved parties or in a court of law.

The advent of the Internet has provided both enterprises and people with a completely new range of infrastructures and tools to interacts and do businesses. Interactions among people and organizations are progressively moving towards the digital world: more and more transactions and interactions will happen by exchanging digital information rather than paper based documents. Recent laws on digital signature and electronic commence [USS.761] also gave to digital documents the same dignity and legal validity that is traditionally attributed to paper-based document. It is likely that in the next few years, we will assist an increase of the usage of digital documents in many sectors of the economy. As a consequence we will assist an increase of cases where digital documents and digital information will be used as evidence during disputes.

Problems such as the integrity of information, the validity of digital signatures, the privacy of digital documents and their storage need to be properly addressed in order to make such a world happen. These problems are even harder if we consider the requirement of preserving digital information over a long period of time (some relevant requirements described by [PRO01]): signatures and encryptions need to be renewed over the long period of time because of the expiration of keys and the availability of new technology, the format of documents need to be renewed (when new rendering tools are available) without compromising their contents, access control needs to be preserved over a long period of time and a reasonable set of copies of documents needs to be available at any time to guarantee their survivability.

It must be possible at any time to demonstrate that a digital document is valid and it has not been tampered.

2. Enterprise Evidence Management

Most of the current solutions support the management of large amount of digital documents and provide storage and search facilities. Little has been done to address the management of enterprise documents over a *long period of time* and in particular to preserve them as "evidence".

The management of digital evidence over a long period of time is a very complex task as it involves both technical and legal aspects. For integrity and validity purposes, digital documents needs to be properly signed and time-stamped by trusted third parties. As signatures grow weaker over a medium period of time, they need to be periodically renewed without compromising their authenticity. The format of digital documents also needs to be renewed over a long period of time to allow the document to be rendered with new technologies, without compromising their content. Privacy and confidentiality needs to be ensured over a long period of time too. This requirement involves the long-term management of identities, encryption, authentication and access control.

Traditional Public Key Infrastructures (PKIs) [Housl99] and Privilege Management Infrastructures (PMIs) [Chadw00] do not address these long-term problems: they provide an infrastructure to issue certificates, revoke them and manage digital keys during their short term lifetime (1-5 years). Renewal processes and long-term management problems are not directly addressed.

Digital documents need to be properly stored. As it is easy to destroy digital information it is necessary to ensure that digital documents are able to survive attacks or disasters.

All the above requirements demand the definition of appropriated trusted processes within enterprises and the involvement of trusted third parties. We believe that in the near future there will be a gradual proliferation of new *Evidence Management Services* to address these needs. An *Evidence Management Service* deployed within an enterprise will explicitly deal with the *long-term evidence management issues* described before: long-term heterogeneous storage, renewal of documents' signatures and their formats, identity and access control tracking and overall integrity management.

The *Trusted E-Services Laboratory* (TESL) at HP Laboratories in Bristol is currently researching in this area.

3. Addressed Problems and Our Approach

In this paper we address the specific problem of the storage of digital documents within an enterprise. This digital information can be used as evidence by the enterprise when dealing with accountability issues.

We investigate how to provide the enterprise with a *best-effort* system to support storage, integrity and survivability of digital documents by using cheap and widely available resources, like personal computers. An Evidence Management Service (EMS) will use this system for evidence storage purposes: it is one of the available storage services to the EMS.

Many solutions to store survivable documents within an enterprise are currently available, including distributed file systems, RAID [Chen94], replicated databases, Storage Area Networks (SAN) and Network Attached Storage (NAS). These resources are traditionally quite expensive.

In spite of this fact, medium and large enterprises already own a vast amount of cheap storage and processing capabilities. In fact, enterprises widely use personal computers and servers to deal with their day-by-day businesses: usually these resources are associated to one or more people that do not use them full time [Douce99]. Both their storage and processing capabilities can be used to store and process enterprise digital documents.

Different models are available for the storage and the management of document integrity and survivability. At the extremes there are two opposite approaches, one based on a heavily centralized control and the other based on a fully distributed one.

In the former case the control of the storage processes is centralized [Sandb85]. A central component is in charge of managing and coordinating these processes. The advantage of this approach is that there is a well-defined point of control and responsibility. The disadvantage is that the central component is a bottleneck and a point of failure.

In the latter case, there is a completely distributed approach both for processing and control tasks [Ander95]. A pure *peer-to-peer* model fits in this category. The advantage is that there is not a unique point of failure and that this model takes full advantage of the distributed resources. The disadvantage is that in such a model anarchism is likely to prevail along with a possible degradation of performances.

A hybrid approach, implementing a distributed file system is described in [Thekk97] and it assumes a fully trusted environment.

Our approach is based on a *hybrid model* that takes advantage of the best features of the approaches described before. We do not make the assumption of a fully trusted environment: we relax this constraint by including potentially untrusted (but not hostile) components. The model is based on a peer-to-peer architecture [Oram2001] where each peer (local PC, server, etc.) has the responsibility to deal with particular storage and elaboration tasks. This component it is not necessarily trusted but it could become trusted if it behaves appropriately over a reasonable period of time. Part of the control is centrally retained by a trusted component.

Control is *devolved* by the central component to the peers if particular constraints on reliability and trust are satisfied. The emerging behavior of the system is *adaptive* to trust assessments in the sense that the storage management strategy varies according to the behavior of the peers, their accessibility and integrity of the locally stored documents. The more the peers are reliable and trustworthy the more the control is delegated to them. Should this trust be abused, the model contemplates the reduction or the revocation of part of the distributed tasks.

4. Assumptions

In this paper we make the following assumptions:

- a medium/large enterprise is involved and cheap computing resources (personal computers, servers, etc.) are deployed and available within the enterprise for evidence storage and processing purposes.

- a reasonably large set of people is willing to participate to the collaborative effort by sharing part of their resources. This set of people and their resources is evolving dynamically, as players can join and leave at any time.

- our system is used by an EMS system and provides a particular implementation of a digital document storage by using cheap computing resources available within the enterprise. Our system preserves the integrity and confidentiality of the stored information.

- the addressed environment is not hostile even if no assumptions are made about the trustworthiness and reliability of the players in fulfilling the agreed collaborative tasks (i.e. keeping a copy of a document for a predefined period of time).

5. Scenario and Use Cases

In our scenario an Evidence Management Service (EMS) is deployed within an enterprise. It interacts with our system to store and retrieve digital documents over a long period of time.

Our system is composed by two basic components:

- A central trusted component where basic decisions are made in term of the storage of information. This component also monitors and rates the behavior of the peers involved in the collaborative effort.

- A set of agents (peers) distributed across the enterprise and installed within enterprise resources (PCs, servers, etc.). Agents are able to store digital documents, return them to the central component and make autonomous decisions about locally stored information. Agents are not necessarily trusted or reliable but they are not hostile.

Our scenario is described in terms of a few use cases, which stress the importance of having an adaptive system able to change its behavior depending on the behavior of the peers.

5.1 People across the enterprise join the collaborative effort

People willing to participate to the collaborative effort, register their resources within the central component of the system, by specifying the amount of resources (storage, processing, etc.) they want to share. They download an ad-hoc *signed* copy of the agent by using a *secure connection* with the central component and install it on their local computing resources (figure 1):

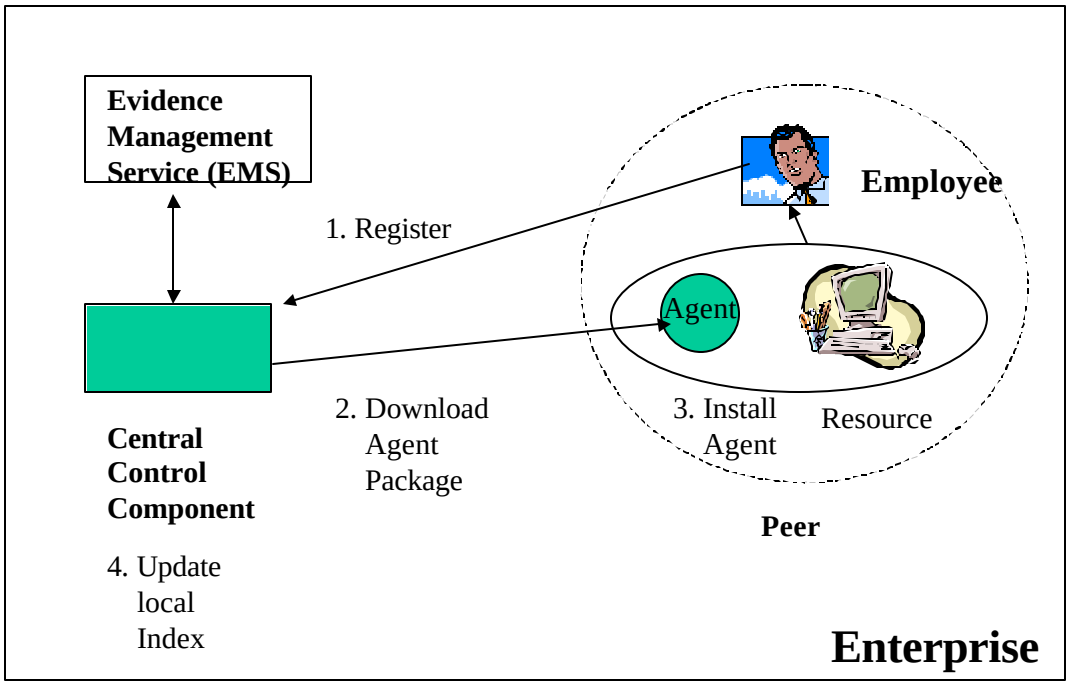

Figure 1

The package downloaded from the central components includes a unique digital certificate [Housl99] and the private key associated to the agent. It also contains the digital certificate associated to the central component. This information is used for authentication with the central component and other peers.

The reason for storing the private key and the public certificate directly in the package is to simplify the overall interaction, by avoiding a fully deployed PKI infrastructure. The risk that a third party intercepts the private key is minimal as the connection is secure, the package is signed and we make the assumption of a non-hostile environment. Even if the private key were intercepted and misused by a third party, the damage would be contained at the agent boundaries. It is not really important if two or more agents share the same identity as far as their locations are unique. An agent is never able to access the content of the documents it stores, as they are encrypted by the central component.

The central component updates a local Index with the properties associated to the new agent and its location.

The set of collaborative peers is dynamic (figure 2): at any time newcomers join the system while older ones withdraw their involvement.

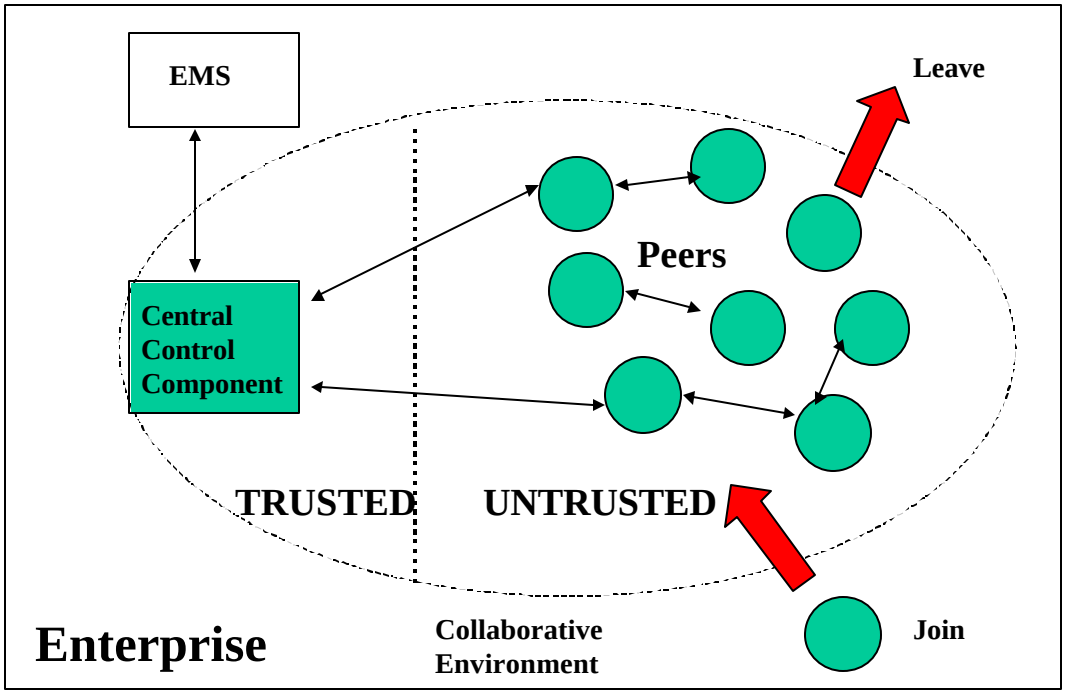

Figure 2

5.2 EMS requires the system to store a digital document

The EMS asks our system to store a digital document (figure 3). A secure connection [FrierKK] is established between EMS and the system, with mutual authentication.

Metadata is passed to the system to specify management information like the importance of the document and the period of time to be stored.

Depending on how critical the digital document is, the central component calculates the required number of replicas to be done and chooses a set of remote available peers (agents) where copies of the digital document can be stored. The choice is made by accessing a local Index containing the list of registered agents and their properties.

The central component encrypts the document by using its private key (or a key within a key pool) and digitally signs it for integrity purposes. It assigns a unique name to the document and securely contacts the selected remote peers for its storage. The unique name is returned to the EMS. The central component stores the locations of the replicas within the local Index.

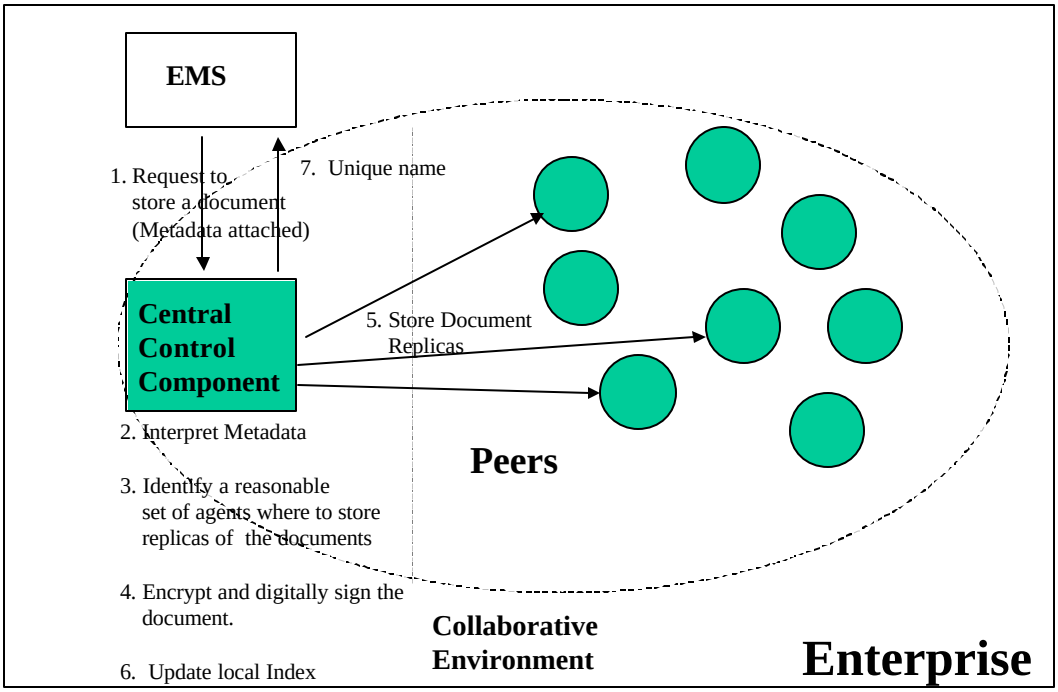

Figure 3

As a variant of the above use case, the central component only contacts a subset of agents where the document has to be stored. These agents are requested to contact other selected peers and ask them to store the document. The central component selects the initial set of agent by using the accumulated rating information (see use case 5.5).

5.3 EMS requires the system to return a valid copy of a digital document

At any time the EMS can ask the system to return a valid copy of a digital document (figure 4). A secure connection is established between the EMS and the system, with mutual authentication. The EMS passes the unique name of the document to the system.

The system consults the local Index and retrieves a few copies of the document by interacting with remote agents.

The system verifies the integrity of the document, decrypts it and if it is not compromised it returns it to the EMS. If the replica is compromised, the system retrieves another replica and repeats the checking process.

The system provides a best effort service to the EMS. Thanks to the monitoring of the agents and their stored documents (see use case 5.5), the system is reasonably able to prevent that all the replicas are corrupted or destroyed as it proactively creates new replicas if the current number of copies is below a predefine threshold.

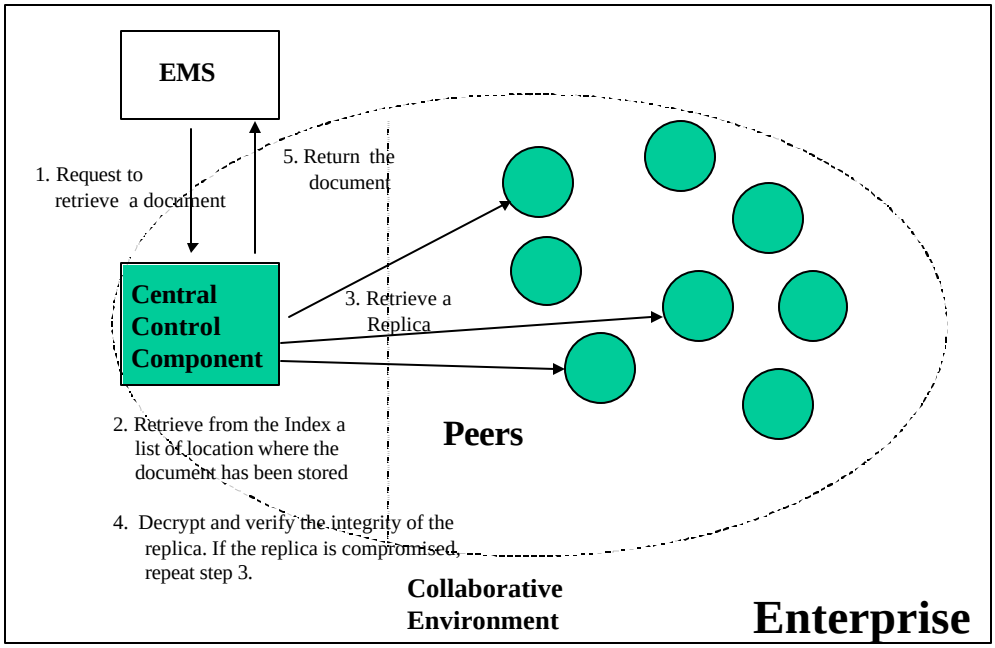

Figure 4

5.4 EMS requires the system to delete a digital document

At any time the EMS can ask the system to delete all the stored copies of a digital document (figure 5). A secure connection is established between EMS and the system, with mutual authentication.

The system consults the local Index and retrieves the set of locations where replicas of the document is stored.

The system interacts with each involved agent and requires it to delete the document. At the end of this task, the related Index entries are deleted. Should any of the involved agents be unavailable, the system will remember this and it schedules for a future deletion task.

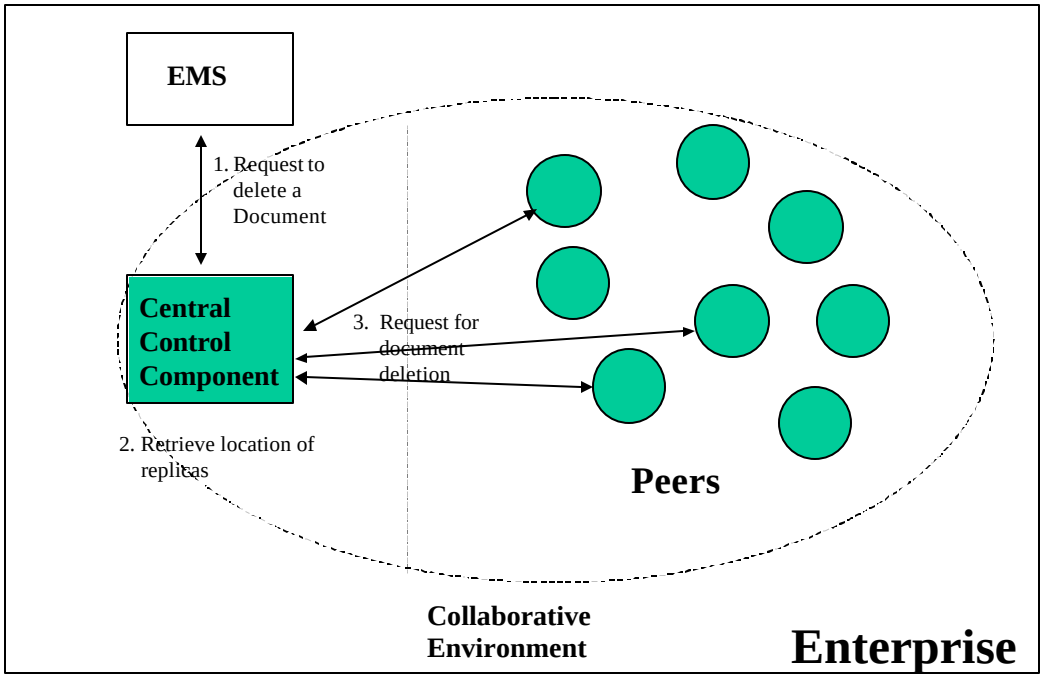

Figure 5

As a variant of the above use case, the central component only contacts a subset of agents where the document has to be deleted. These agents are requested to contact other selected peers and ask them to delete the document. The central component selects the initial set of agent by using the accumulated rating information (see use case 5.5).

5.5 The system monitors the behavior of peers and collects rating information

Because of the dynamic evolution of the peers, the central component needs to monitor them in order to ensure that stored documents are preserved over a long period of time (figure 6).

A monitoring module (within the central component) periodically verifies if the replicas of a document have not been compromised (by checking their signature) and if there are still enough copies.
Should replicas of a document be compromised or destroyed, the system creates alternative copies, as defined by policies. These policies dictate how many copies should be available at any time and the threshold under which actions needs to be taken.

While monitoring peers, the system collects information about their behavior: this information includes the number of time a remote peer was unavailable, the number of time local replicas have been compromised or destroyed.

The collected data is used to provide the system with information about the reliability and "trustworthiness" of remote peers.

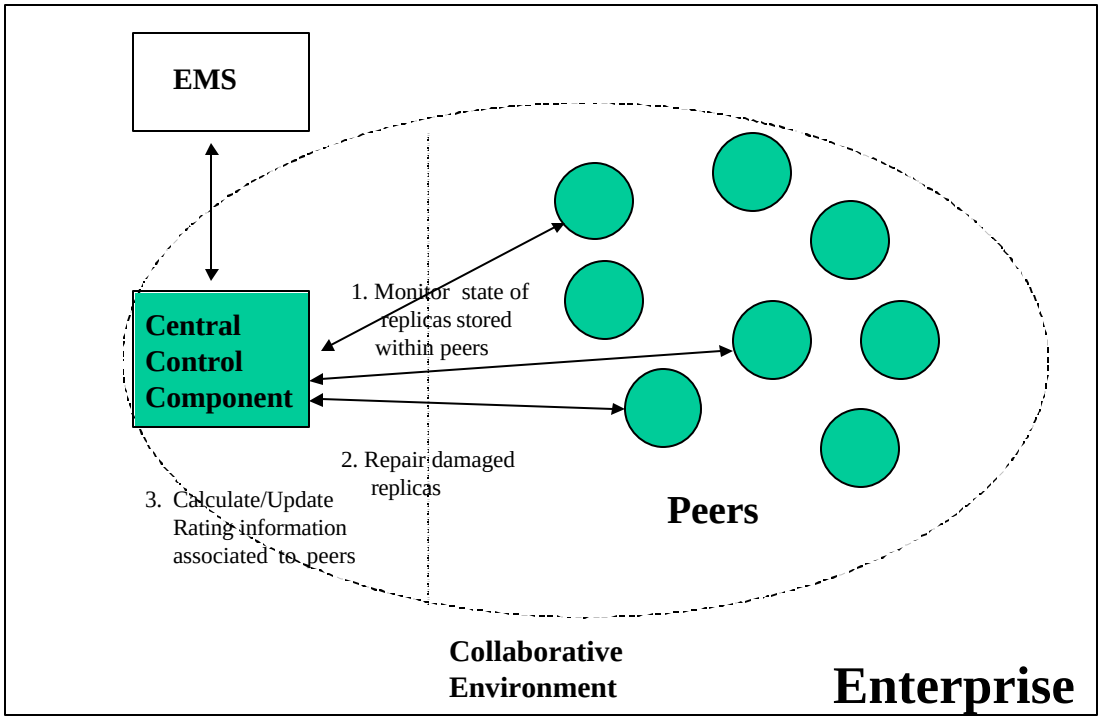

Figure 6

5.6 The system delegates monitoring tasks to peers

The monitoring activity can be quite heavy, as it requires a periodic verification of the state of remotely stored document replicas.

The central system can delegate part of its monitoring tasks to remote peers (figure 7). Rating information is used to identify an appropriate set of remote peers according to local policies.

The system securely contacts the remote agents, enable their monitoring features and delegates a few monitoring activities to them. The central component retains the task of monitoring the behavior of these agents and collecting related rating information.

Each remote peer executes the delegated tasks on behalf of the central system and periodically sends back information to the central component.

Depending on the level of trust and reliability, a remote peer can be delegated the task of repairing damaged replicas of a digital document.

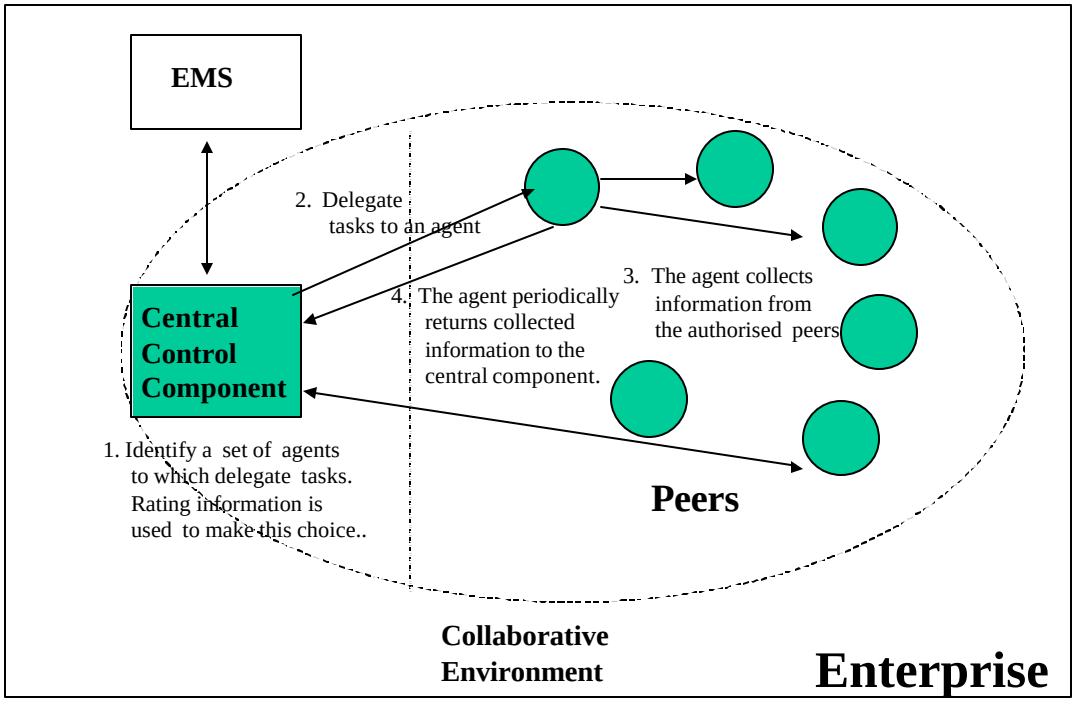

Figure 7

5.7 Peer-to-peer interaction originated by an agent

A person hosting an agent on their PC can change their mind regarding the amount of resources to be allocated for the collaborative effort. For example, this decision can have the consequence of needing to delete some of the locally stored documents.

The local agent takes the initiative to inform the central system about this event and asks the central system for alternative locations where the involved documents might be stored (figure 8). Depending on the level of trustworthiness and reliability, the agent can be authorized to coordinate this activity.

This approach does not prevent the PC's owner from directly deleting documents from the local storage. In this case the "perception" of trust associated to the resource will be negatively affected because the PC's owner is not acting as agreed.

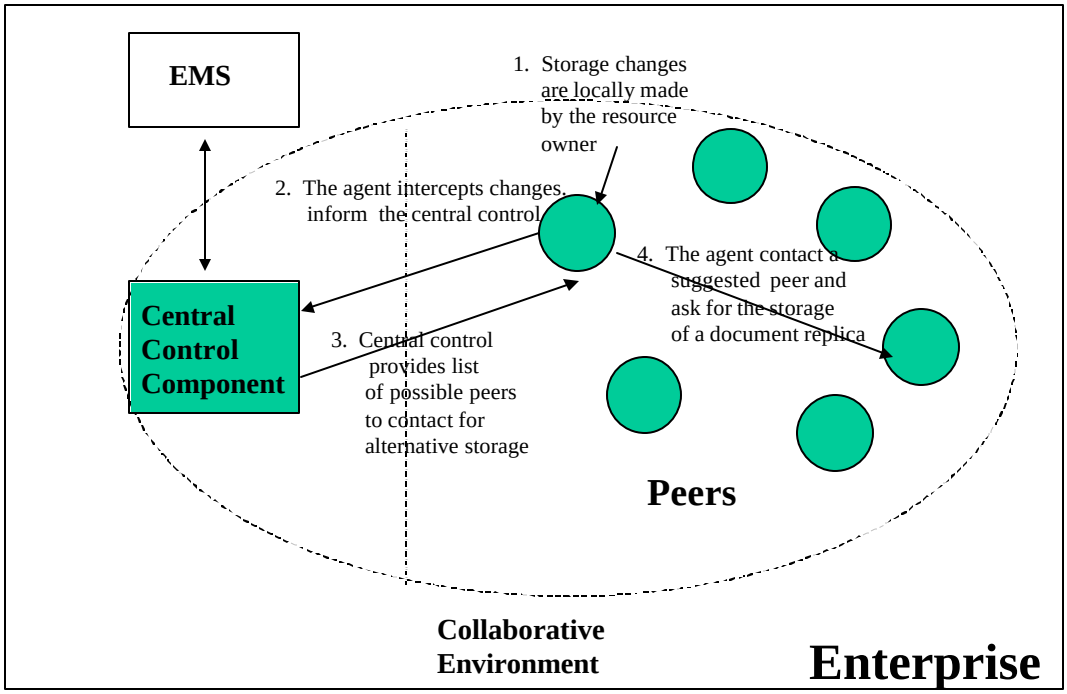

Figure 8

5.8 The system revokes delegated tasks

The central system can at any time revoke tasks delegated to a remote agents depending on rating information it collects (figure 9). This decision could be dictated by remote agents misbehavior or their unreliability. The central system contacts the remote agent and disables the delegation feature of the agent.

For similar reasons, the central system can store replicas of a document elsewhere and reduce the responsibilities attributed to a remote agent.

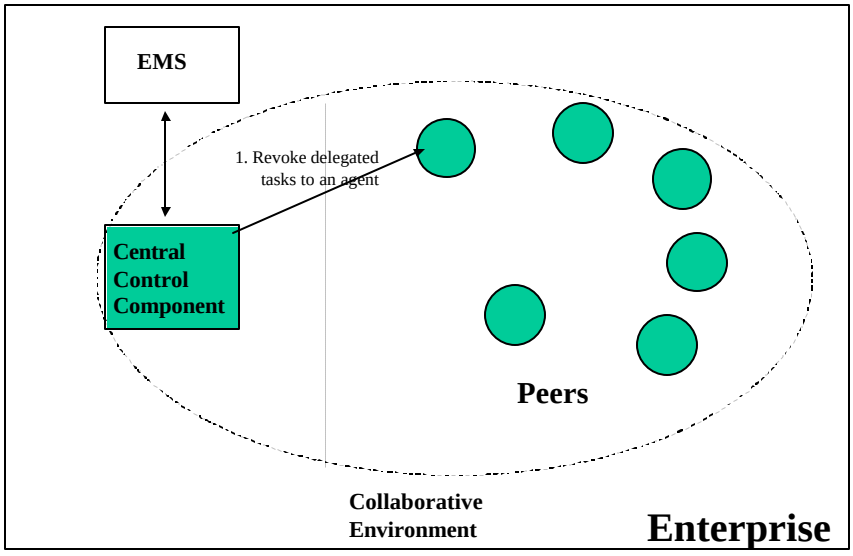

Figure 9

6. High Level Architecture

The following picture contains a high level description of the system architecture (figure 10):

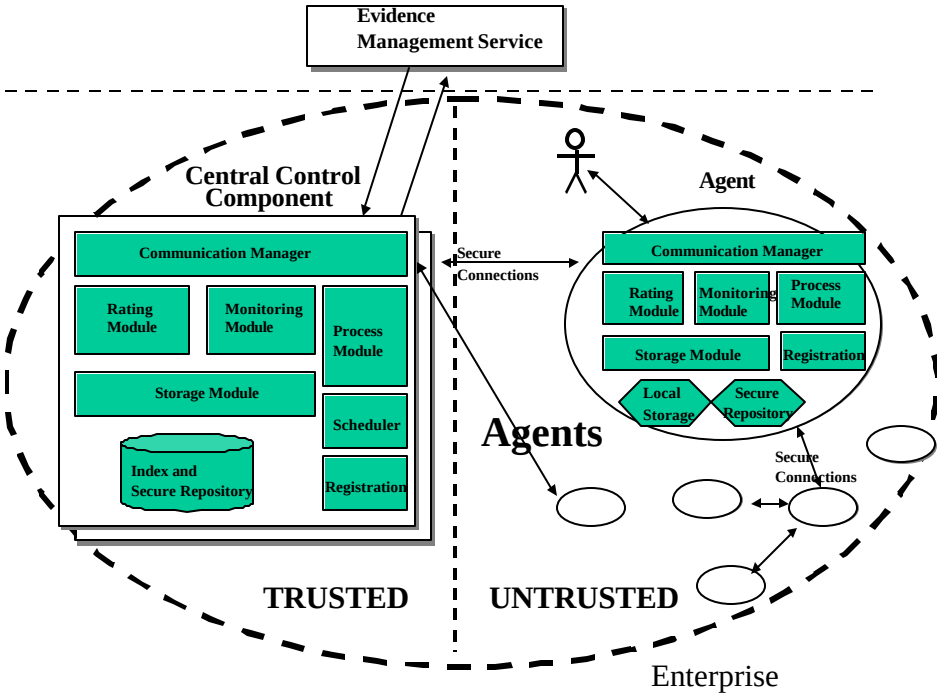

Figure 10

The architecture is based on a hybrid approach mixing a *central trusted control component* with an *untrusted peer-to-peer component* based on collaborative agents.

The *central control component* is in charge of coordinating the interaction with the external Evidence Management Service (EMS) and the Agents. It is also in charge of monitoring the overall system and rate its behavior.

This component is made of the following modules:

- Communication Manager: it is in charge of dealing with EMS interactions. This module is also in charge of communicating securely with agents, dealing with their incoming calls and authorization issues [Casas01].

- Registration Module: it is in charge of dealing with the registration of remote agents. This module request the subscriber for information like the resource location, the max amount of resource to be shared, details about the owner, etc.

Registration requests might be rejected depending on local policies (minimum amount of resources to be shared, location, etc.)

- Index and Secure Repository: it is a local repository containing information about the registered agents, their rating and the locations where documents have been stored. The Secure Repository also contains sensitive material like private keys, trusted certificates, etc. *This is not a single point of failure because these data structures can be replicated by using traditional techniques.*

- Process Module: it is in charge of all the operational tasks. These tasks include the encryption and signature of documents, the selection of proper agents where replicas can be stored and the update of the information contained in the Index and Repository. This module supports the processes for delegating tasks to agents or revoking them, according to local policies, the system workload, agents' reliability and their trustworthiness. This module is also in charge of dealing with system "maintenance" tasks like the re-encryption and signature of documents when private keys are expiring.

- Scheduler: this module is in charge of scheduling system related tasks that need to be done periodically like the re-encryption and signature of digital documents.

- Storage Module: this module is in charge of all the storage and retrieval operations within the local Index and Secure Repository. It also manages the names associated to the documents to be stored;

- Monitoring Module: it is the module in charge of periodically monitoring the states of replicas stored within agents. This module contains a component that selects agents and replicas to be monitored according to local policies. The monitoring module interacts with the process module to verify the integrity of stored replicas. It supplies the Rating Module with the result of all this tests.

- Rating Module: it is in charge of calculating the rating information for each agent that participates to the collaborative effort. The rating calculus is based on *trust and reliability functions* based on information retrieved about the agents, like the availability of the PC agents are running on, the availability of the information stored at their sites, the integrity of this information. This rating information is used to make decisions about the selection of agents for storage and delegation purposes.

For efficiency and survivability reasons, our architecture supports a *pool* of central control components, all of them sharing the same Index and secure Repository. The Index and the security Repository can also be replicated by using traditional databases.

The *Agent Component* is very similar to the Central Control Component with the exception of the global Index (of agent locations and the distribution of replicas across the enterprise). A global Index is maintained only by the central component.

The agent's Monitoring and Rating modules have cut down functionalities, which can be enabled only when particular tasks have been delegated to the agent by the central component.

The agent's Communication Module contains a module to simplify the interaction with the user that owns the shared resources. This module includes a UI to mediate the interactions between the user and the local storage. Changes to the local storages are communicated by the agent to the central component. These changes might include the deletion of local replicas. If authorized, the agent's Process Module can take the initiative of interacting with other agents to create new replicas.
The agent's Process Module is also in charge of orchestrating the activities delegated to the agent by the Central Control Component such as the monitoring of other agents' behaviors.

In term of *security*, the interacting components always authenticate themselves by using certificates that have been issued to them by the central component. The central component acts as a local Certificate Authority. Delegated tasks are also asserted within digital attribute certificates [Casas01] issued and signed by the central component.
This does not prevent private keys from being stolen from local agents and misused. However, the negative effects are limited, as an agent interaction with other agents must always be approved/delegated by the central control.

The system supports a *best-effort survivability* of the stored digital documents. Thanks to the monitoring activities the system can detect in advance corruption or degradation of the stored information and react accordingly. As personal computers are usually geographically distributed across enterprise sites and are available in a large number, they can provide a viable support to cope with the survivability issue in case of disaster or attack.

The system manages the *confidentiality* of the stored documents by encrypting their content and periodically renewing the encryption, over a long period of time.

The system deals with the *integrity* of the stored documents by digitally signing the documents and renewing the signature over a long period of time.

The overall architecture is *adaptive to trust and reliability assessment*. The system is able to monitor the agents and verify the integrity and accessibility of the locally stored documents. The more the agents are reliable and trustworthy, the more monitoring and control tasks are delegated to them. Rating tasks can be partially *delegated* to agents if trusted. Trusted agents (having a proper authorization) can directly interact with other agents (peer-to-peer interaction) in order to fulfill particular tasks. The system is able to react to situations where the overall reliability decreases by hardening the control and re-

centralizing it. Policies defined in each module describe how to deal with such situations and drive the behavior of the system.

7. Status

We are currently refining the architecture described in the previous section with particular attention to the monitoring and rating mechanisms. A prototype is currently under development. We are also investigating how to extend our model to an inter-enterprise context.

8. Related Work

Several efforts have been made to achieve document survivability, confidentiality and integrity in a distributed environment.

The PASIS project [Wylie00] describes an architecture for building information storage systems whose availability, confidentiality and integrity policies can survive component failures and malicious attacks. Client applications interact with a PASIS storage system through a PASIS agent. Storage devices and repair agents monitor the system status. This system is a completely distributed storage system where decentralization is hidden to clients by using client-side agents. The implication is that PASIS agents need to be installed and maintained at each client site.

Architectures to achieve survivability are also described in the Intermemory Project [Goldb98], the Eternity service [Ander96], e-Vault [Iyeng98] and Delta-4 Project [Deswa91].

The Farsite [Bolos00] and OceanStore [Kubia00] projects address this problem by using a “pure” peer-to-peer approach.

The Farsite project describes an architecture for a serverless distributed file system. The system does not assume mutual trust among the client computers. It provides security, availability and reliability by distributing multiple encrypted replicas of each file among the client machines. Machine performance and behavior is measured and reported.

OceanStore is a utility infrastructure that spans the globe and provides continuous access to persistent information. The infrastructure is made of untrusted servers: data is protected through redundancy and cryptography. Data can be cached anywhere and at anytime. The monitoring of usage patterns adapts the system to regional outages and denial of service attacks. A pro-active movement of data enhances the overall performance.

In the last two cases the main objective is to provide a distributed file system and storage within an untrusted environment. Replication and measurement are used to ensure data

survivability. Encryption and digital signatures are used to ensure confidentiality and integrity.

In our approach we relax the assumption that the whole system is untrusted. We use a hybrid approach. Trust resides at least in the central component, which is in charge of controlling and monitoring the overall system behavior. The remote machines are not necessarily trusted but, because of the collaborative enterprise-based environment, we assume they are not hostile.

Monitoring and rating mechanisms are not only used to supply a self-healing functionality but also they are used to change the perception of trust associated by the central system to the remote machines. The higher is the rating associated to a remote machine the more the central system is willing to delegate tasks to it.
Because of the hybrid architecture that mixes a central component with a peer-to-peer based one, the system can adapt its behavior, its workload and perception of trust depending on the circumstances.

Our system is a best effort system and it must be considered in the overall context of an Evidence Management Service (EMS), built for the purposes of long-term management of evidence. Our system provides an adaptive storage of evidence and it is one of the storage components available to the EMS.

9. Summary and Conclusion

Evidence is a key aspect when dealing with accountability and reputation issues. People and organizations retain documents and information for a long period of time as evidence of their behavior and actions. In the physical world, the most common sources of evidence are paper-based documents, used in mostly all kind of human-based interactions.

Because of the shift to the Internet paradigm, it is likely that digital documents are going to be more and more relevant as digital evidence. This introduces a broad set of problems to be solved like digital evidence integrity, privacy, renewal and storage. In particular, the management of digital evidence over a long period of time is undoubtedly going to be one of the major challenges. HP Labs, TESL – Bristol, are currently researching on an *Evidence Management Service* to address the above problems.

In this context, we describe a system for the storage of digital evidence within an enterprise, supporting survivability, integrity and confidentiality.

Related work in this field makes the assumption that either all the involved components are trusted or all of them are untrusted.

Our system is based on a hybrid architecture mixing a trusted centralized control with untrusted peer-to-peer components, made of cheap enterprise resources. Confidentiality and integrity are ensured by using cryptographic techniques.

We suggest the usage of cheap and abundant resources (like PCs) available within the enterprise to achieve these goals and minimize costs. Because of the volatility of these resources, a best-effort replication mechanism and an adaptable monitoring system are used to support the evidence survivability. A rating mechanism is used to evaluate the trustworthiness and reliability of the peers to store documents and adapt the workload of the system accordingly.

The success of such a system is highly dependable on the number of the people within an enterprise willing to participate to the collaborative effort and the amount of resources they are sharing.

10. Acknowledgements

We would like to thank people from the "Trusted E-Services Laboratory – HP Laboratories, Bristol" and "DEIS – University of Bologna" for their feedback and support.

11. Bibliography

[Wylie00] J. J. Wylie, M. W. Bigrigg, J. D. Strunk, G. R. Ganger, H. Kiliccote, P. K. Khosla - Survivable Information Storage Systems, *IEEE Computer* - August 2000

[Goldb98] A. Goldberg and P. Yianilos - Prototype Implementa-tion of Archival Intermemory - *Proc. Advances in Dig-ital Libraries (ADL 98),* IEEE CS Press, Los Alamitos, Cali*f.,* pp. 147-156 - 1998

[Ander96] R. Anderson - The Eternity Service - *Proc. PRAGO-CRYPT 9*6, CTU Publishing House, Prague - 1996

[Iyeng98] A. Iyengar et al. - Design and Implementation of a Secure Distributed Data Repository - *Proc. 14th IFIP Int'l Information Security Conf. (SEC 98),* ACM Press, New York – 1998

[Deswa91] Y. Deswarte, L. Blain, and J. Fabre - Intrusion Tolerance in Distributed Computing Systems - *Proc. IEEE Symp. Security and Privacy,* IEEE CS Press, Los Alamitos, California, pp. 110-121 - 1991

[Bolos00] W. J. Bolosky, J. R. Douceur, D. Ely, and M. Theimer - Microsoft Research - Feasibility of a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs - SIGMETRICS 2000 - 2000

[Kubia00] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, D. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zao – OceanStore: An Architecture for Global Scale Persistent Storage – University of California, Berkley – ASPLOS 2000 - 2000

[Oram01] A. Oram - Peer-to-Peer: Harnessing the Power of Disruptive Technologies - O'Reilly - March 2001

[Ander95] T. Anderson, M. Dahlin, J. Neefe, D. Patterson, D. Roselli, and R. Wang - Serverless Network File System - ACM Operating Systems Review, vol. 29, no. 5 - December 1995

[Thekk97] C. Thekkath, T. Mann, E. Lee - Frangipani: A Scalable Distributed File System - 16th SOSP, pp. 224-237 – December 1997

[Sandb85] R.Sandberg, D. Goldberg, S. Kleiman, D. Walsh, B. Lyon - Design and Implementation of the Sun Network File System – Summer USENIX Proceedings - 1985

[Douce99] J.R. Doucer, W. J. Bolosky - A Large Scale Study of File System Contents - SIGMETRICTS '99 27(1), pp. 59-70 - May 1999

[USS.761] S.761 - The "Electronic Signatures In Global And National Commerce Act"

[Housl99] R Housley, W. Ford, W. Polk, D. Solo - RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL profile, IETF – 1999

[Chadw00] D. W. Chadwick, S. Legg - Internet X.509 Public Key Infrastructure Additional LDAP Schema for PKIs and PMIs, IETF - 8 September 2000

[Casas01] M. Casassa Mont, R. Brown - PASTELS Project: Trust Management, Monitoring and Policy-driven Authorization Framework for E-Services in an Internet-based B2B Environment, HPL Report, HPL-2001-28 - 2001 [HP Internal]

[Chen94] P. Chen et al. - "RAID: High Performance, Reliable Secondary Storage" - ACM Computing Surveys, pp. 145-186 - June 1994

[FrierKK] A. O. Frier, P. Karlton, and P. C. Kocher - The SSL protocol - IETF - web site: ftp://ietf.cnri.reston.va.us/internet-drafts/draft-freier-ssl-version3-01.txt

[PRO01] Public Record Office U - Electronic Record Management - web site: http://www.pro.gov.uk/recordsmanagement/eros/default.htm - 2001