APM

POSEIDON HOUSE • CASTLE PARK • CAMBRIDGE • CB3 0RD UNITED KINGDOM
+44 1223 515010 • Fax: +44 1223 359779 • Email: apm@ansa.co.uk • URL: http://www.ansa.co.uk

E2S

End-to-End Security Over The Internet: Deliverable D1 - Implementation Architecture

Andrew Herbert

(Project Technical Director)

Abstract

This document is the description of the implementation architecture for the E2S Project Pilot Demonstrators. It identifies, positions and outlines the function of the main elements of the common technology framework used by the pilots and the overall system security model for their deployment and use. The first version of the implementation architecture was used to plan the development of the demonstrators and supporting technology. This final version has been updated to reflect the lessons learned.

The document is intended for a technical audience. It assumes the reader has reasonable familiarity with Internet and security technology.

Detailed specifications for the elements of the framework identified in this report have been produced as separate E2S deliverables, referenced as appropriate in the text.

APM.1819.06	**Approved**	21st November 1997
	Architecture Report	

Distribution: E2S Partners
Supersedes:
Superseded by:

End-to-End Security Over The Internet:

Deliverable D1 - Implementation Architecture

ESPRIT Project 20.563: End-to-End Security over the Internet

Deliverable D1 - Implementation Architecture

Andrew Herbert
Project Technical Director

APM.1819.06

21st November 1997

The material in this report has been developed as part of the ESPRIT End-to-end Internet Security Project (E2S), RTD project number 20.563.

The partners in the E2S Project are: APM Ltd, GEMPLUS, GMD German National Centre for Information Technology, Hewlett-Packard European Laboratories, Hewlett-Packard Grenoble Networks Division, Hewlett-Packard Worldwide Customer Support Organisation, Onyx Ltd, Smart Card Forum, Swiss Bank Warburg, Technische Universität Berlin, VISA International.

The author acknowledges the help and assistance of his colleagues, both from APM and from the other E2S partners, in the preparation of this report.

Questions concerning this project should be addressed to:

APM Limited

Poseidon House
Castle Park
CAMBRIDGE
CB3 0RD
United Kingdom

TELEPHONE UK	(01223) 515010
INTERNATIONAL	+44 1223 515010
FAX	+44 1223 359779
E-MAIL	apm@ansa.co.uk

Contents

1 Introduction

1.1 Scope

This document specifies the implementation architecture for the ESPRIT End-to-end Internet Security Project (E2S), RTD project number 20.563. It is the deliverable from Task D1.

The specification has been revised from the initial version in the light of:

- changes in user requirements (particularly the evolving model of an electronic marketplace and commercial purchasing)
- changes in available technology (particular the use of compartmentalised mode workstations for system partitioning)
- feedback from the roll-out of the E2S pilot demonstrator projects.

The major changes to the architecture comprise:

- introduction of the “browse, order, pay, deliver, reconcile” model for commercial purchasing
- definition of end-to-end secure transactionprotocols (secure application sessions for IT integration, secure purchasing sessions for corporate purchasing)
- introduction of “trust centres” to manage the trust relationships established between users and services in E2S systems
- use of trusted operating systems to provide high assurance platforms and to assist with optimising performance and systems partitioning without weakening the separation of controls on security sensitive components
- clearer description of the coupling between the components.

The complete E2S implementation architecture consists of:

- this overall specification
- a set of component specifications
- descriptions of pilot demonstrators built according to the architecture
- guidelines for implementing systems using the E2S architecture.

This document sets out the architecture - i.e., the common technology framework - of the E2S project pilot demonstrators. The architecture identifies, positions and outlines the function of the main technologies used in the project. Importantly it determines the trust and security model for the use of those components in the demonstrators.

Detailed architectural specifications for the components identified in this report have been produced as deliverables of E2S “infrastructure component” tasks. The examples of application of the architecture and guidelines for implementing further systems using the architecture have been produced as deliverables of E2S “pilot demonstrator” tasks. The particular technologies and standards used within the pilots are noted in the architectural

description. However the architecture itself has been deliberately positioned at a higher level of abstraction so that other choices can be made.

1.2 Audience

The document is intended for use

- within the E2S consortium to document the common technology framework across all the E2S pilot demonstrators and to position the technology deliverables.
- outside the E2S consortium to explain the scope and rationale of the E2S common technology framework to a technical audience.

1.3 Structure of document

This document is divided into the following sections:

1. Introduction (this section)
2. System model
   — a high level view of an E2S system in terms of the major roles, responsibilities and requirements in business-to-business electronic commerce
3. Security model
   — a description of the security functions and security information used in the architecture and the trust placed in users, administrators and security technology
4. Summary
   — a brief description of the framework as a whole to provide an overall picture for the following three sections describing the full detail of the framework
5. Client technology
   — Technology for user interaction with end-to-end secure Internet applications
6. Secure connectivity technology
   — Technology for securing an end-to-end Internet path between users and applications
7. Server technology
   — Technology for supporting securely accessed Internet applications
8. Examples
   — A summary of the choices made from the common technology framework in the E2S project pilot demonstrators
9. Viewpoint analysis
   — Concepts and rules from the architecture categorised in ISO/ITU ODP viewpoints [ISO 10746-3] to enable alignment of the E2S implementation architecture with other standards for open distributed processing.

1.4 Background

The E2S architecture has been derived from a pragmatic assessment of E2S partner requirements for technology appropriate to large scale electronic commerce. These requirements, documented in *User Requirements* [C1], determine both the functionality required by the pilot demonstrators and the constraints on architecture and technology choices dictated by regulatory concerns, availability of standards and market pressures (hence the positioning of the architecture as an "implementation architecture" rather than a "reference model").

The criteria for building the architecture were:

- **universality** - the security provisions should be widely applicable to as many Internet and Intranet electronic commerce scenarios as possible (i.e., the provisions should not be specific to the E2S pilot demonstrators)
- **security** - the architecture shall embody the high levels of security required to enable businesses to put trust in electronic processes
- **reliability** - the architecture shall embody the high levels of resilience and recovery necessary to support mission critical functions
- **portability** - the architecture shall accommodate multiple computing platforms
- **pragmatism** - the architecture shall be implemented with minimal changes to existing paradigms and APIs
- **performance** - the architecture should not make applications slower or more difficult to use
- **durability** - the architecture should anticipate expected changes in Internet and platform technology
- **end-to-end** - the architecture should provide security for the complete path from a user on the Internet through to the supporting applications and data on the internal networks ("Intranet") of the organisation delivering an electronic commerce service to the client
- **system-to-system**- in addition to enabling electronic commerce between users and electronic commerce applications, the architecture should allow for business-to-business transactions in which the "user" is a computer application running on behalf of an organisation.

1.5 Application

Within the E2S project, the architecture has been used to develop a common technology framework across a number of pilot demonstrators covering the areas of:

- Secure telecooperation in administration (Technical University of Berlin)
- Customer support for a major computer vendor (HP)
  - information services for value-added resellers
  - warranty and claims processing for printer repairs
  - on-line sale of licensed computer software
- Internet investment banking (Swiss Bank)
- On-line marketplace for business-to-business commerce (Onyx).

These applications can be classified both in terms of the style of commerce supported and the nature of the goods and services involved. Table 1.1 shows the E2S focus is on business-to-business transactions involving soft goods (e.g., software licences for HP WCSO), services (e.g. Investment Banking for Swiss Bank) and hard goods (e.g. Electronic components for Onyx).

A key point about business to business transactions is that often they take place in the context of pre-existing business relations, which may already be supported by IT and therefore integration with both the consumer's and the seller's IT is a major concern for deployment.

Table 1.1:

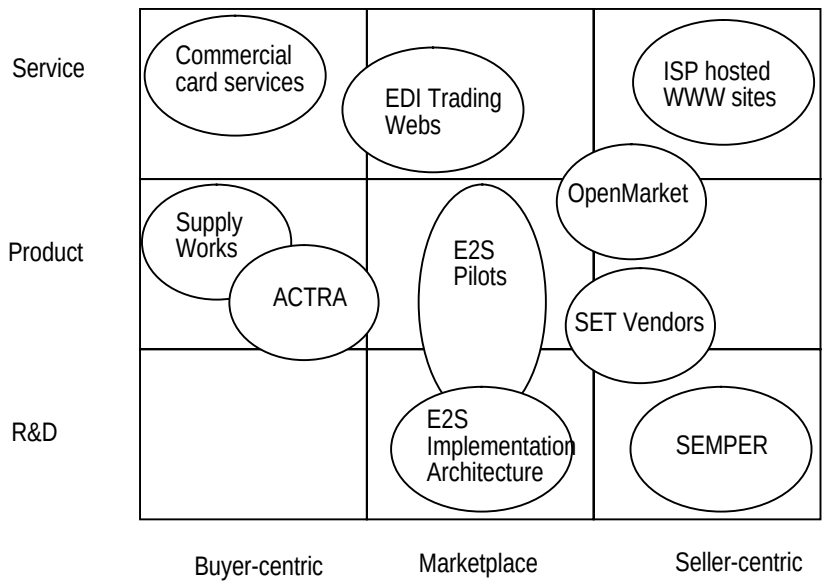
Style	Hard Goods	Soft Goods and Services
Consumer-to-business		Swiss Bank
Business-to-business	Onyx HP	TUB SBCW HP Swiss Bank

Today these relationships are typically based on telephone and fax communications. To be successful Internet-based electronic commerce must provide added value, and this will come from reduced transaction costs and increased efficiency from successful IT integration.

1.6 Positioning

The positioning of E2S relative to other initiatives in the electronic commerce arena as indicated by Figure 1.1: the E2S architecture is oriented towards the support of maintenance, repair and operations, in contrast to seller-centric architectures oriented towards casual consumers and buyer-centric systems oriented to supply chain maintenance.

Figure 1.1: Positioning



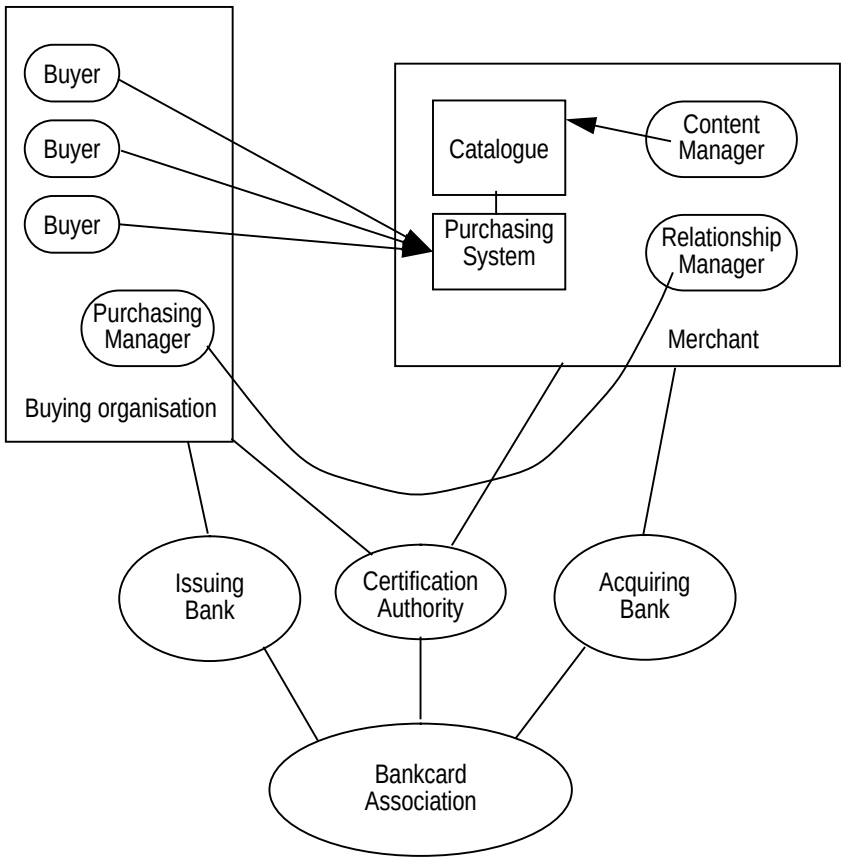
2 System Model

2.1 Generic Model for Business-to-Business Electronic Commerce

The generic model for E2S is shown in Figure 2.1. It is explained in more detail in the [C1] *Consolidated User Requirements* and [E2.10] *Plan for Live Business Payment Trial.*

The four primary roles are the **buying organisation**, the **merchant**, the **bankcard association** and a **certification authority**.

Figure 2.1: System Model



Within the buying organisation there are individual **buyers** authorised by the organization's **purchasing manager** to make purchases on behalf of the organization from the Merchant.

Within the Merchant there is a content manager responsible for how goods and services are displayed in the merchant's catalogue and a relationship manager who has the job of negotiating business relationships with potential buyers and informing the content manager the terms and conditions under which the buyer is to be offered those goods and services. Different buyers

may negotiate different terms and conditions - for example, by committing to bulk or recurrent orders.

The **purchasing and payment infrastructure** is provided by a bankcard association. **Issuing banks** within the association provide buying organisations with **purchasing bankcards** to enable ordering and payment. **Acquiring banks** provide **payment gateways** through which a merchant can request payment for transactions made with a purchasing bankcard.

The **certification authority** provides certified digital identities to the other parties. (For this reason, the certification authority is often described as a "trusted third party".) Digital identities are cryptographically signed names. The certification authority undertakes to ensure that a digital identity is only issued to the entity the name denotes. This necessarily involves making physical checks on identity and ensuring the owner of the identity protects the cryptographic keys he will use to sign transactions. Typically certification authorities associate a level of confidence with digital identities based on the thoroughness of these checks.

The WCSO pilot is a good example of the generic model (see 8.2).

2.1.1 Requirements

Within this overall framework there are strong requirements on all the parties:

- Merchants require:
  - integration of card acceptance with order entry (i.e., the order is tied to use of a purchasing bankcard)
  - capability to send order confirmation to the buying organisation
  - capability to allow buyer to track progress of the order through to fulfilment
  - capability to supply a receipt with full line item details (buyer, order, terms and conditions)
  - convenient secure on-line upload of content.
- Buyers require:
  - ability to use the system from both inside and outside their corporate firewall
  - capability to recover past orders
  - capability to make repeat orders
  - confirmation of terms and conditions on order "signature"
  - contract-specific pricing, terms and conditions
  - personalised views of the catalogue.
- Purchasing managers require:
  - capability to modify purchasing authority for individual buyers
  - capability to act as certification authority for buyers within the buying organisation.
- Bankcard associations require:
  - commercial cards to be identified by merchants

  - buying organisation to be recognised as a member of a marketplace by a merchant
  - identification of the buying organisation for taxation and/or internal reporting purposes
  - capability to provide full transaction information to merchant
  - capability for merchant to exchange tax and other data with buyer.
- Certification authorities require
  - their digital identities to be accepted by buyers, merchants and banks
  - capability to check the identity of users
  - capability to verify the holders of digital identities protect cryptographic keys adequately
  - capability to revoke identities when the status of a person or organization changes to warrant continued use of the identity invalid.

2.2 Marketplace Infrastructure

According to the [B1] *Market Review* report, there is a business opportunity for third party **marketplace operators** take on the content provision and purchasing infrastructure for a set of Merchants.

A marketplace operator provides an on-line marketplace. The marketplace allows buyers to browse merchant's catalogues and select items for purchase as a single marketplace transaction. The marketplace operator deals with splitting the buyers transaction into sub-transactions with each merchant directed towards the individual merchant's order processing IT.

To be successful, Internet-based electronic commerce must provide added value over existing bankcard, telephone and fax methods. The aim of banks and marketplace operators in providing a marketplace infrastructure is to reduce transaction costs and increase efficiency from successful IT integration.

The link from the marketplace operator to the merchant's order processing system and the means by which the merchant uploads content and relationship information to the marketplace must necessarily be secured. This can often be viewed as use of the generic model with the marketplace operator functioning as the "merchant" and the merchant functioning as the "buyer".

In detail(see [E2.5]), a market place operator:

- provides electronic commerce "facilities management services" to merchants
- acts as the certification authority for the buyers (perhaps through a trusted third party)
- provides a catalogue with secure access, search facilities
- supports a range of pricing structures list, discount, repeat buy, cost plus etc.)
- acts as a trusted third party with respect to licensing and delivery
- provides links to merchant's logistics and financial IT.

The marketplace operator requires the support of one or more banks. In particular banks:

— “brand” marketplaces with their bankcard

— act as certification authorities for buying organisations (perhaps through a trusted third party)

— provide full reporting to the bankcard holder.

The Onyx pilot is an example of a marketplace infrastructure (see 8.4)

2.3 Centralised administration

The TUB Secure Telecooperation pilot (see 8.1) is a special case of the generic model specialised towards the needs of a central administration for a departmental organisation.

In this context the central administration acts towards other units in that departments as a merchant providing information services and centralised purchasing (i.e., the purchasing manager is also the buyer). Since the transactions are internal to the organization the payment infrastructure is simplified because the units have budgets rather than real accounts and there is no requirement for electronic payments to be made between departments and the administration.

The central administration may itself act as a buyer, on behalf the other units, using electronic commerce to forward their orders after approval and make payment on their behalf. Alternatively, where appropriate the administration can choose to delegate buying authority to departments, following the generic model.

2.4 Exported services

The SBCW investment banking pilot (see 8.3) is also a special case of the generic model in two respects. First the merchant delivers the catalogue and purchasing system across the network as mobile code to run on a server in the buyer’s organisation. Second the bank chooses to act as the certification authority rather than using a trusted third party.

By exporting a server to the buyer organisation, the system allows the server software to integrate with the buyer’s IT (e.g., for keeping audit records, linking to buyer’s in-house applications). It requires that transactions between the downloaded server and the host systems in the bank are themselves secured. In effect a secure message channel has to be built from the server back to the bank.

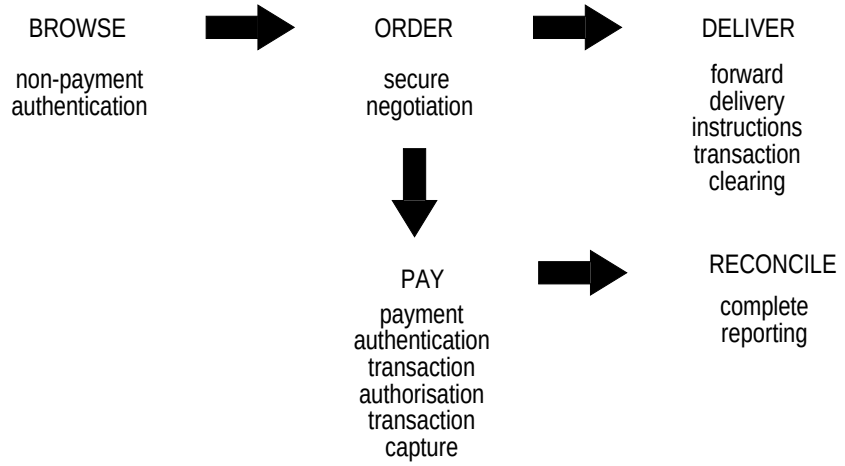
Within the buyer organisation, the generic model can be applied to allow individual buyers to connect to the server and use the system. Depending upon the nature of the buyer’s IT infrastructure it may be necessary to ensure this is secured, following the E2S architecture.

2.5 Browse, order pay, deliver, reconcile paradigm for purchasing

Commercial purchasing differs from general consumer purchasing because the merchant offers different prices and/or terms and conditions to different buyer organisations and in addition different rules with respect to value added tax often apply.

E2S purchasing follows a browse, order, pay, deliver, reconcile sequence as shown in Figure 2.1.

Figure 2.2: Commercial Purchasing



The browse stage can be subdivided into a public browse step to locate an appropriate catalogue, an authentication step (of the buyer) followed by a second stage of browsing in a private catalogue showing buyer specific information.

During the browsing phase, the user selects items into a “shopping basket”. When the buyer has completed her selection, the transaction proceeds to the order phase. The merchant makes an “offer” of a price, terms and conditions for all the good in the basket. This offer is digitally signed by the merchant, the buyer and as a confirmation by the merchant again. So after the exchange of this signed document, both parties have a real contract which can be offered as evidence of a dispute.

The contents of the shopping basket will need to be regrouped by VAT rate for order processing.

The quotation from the catalogue publisher and the order form completed by the buyer provides data which can be captured on-line by the merchant’s sales order processing system to avoid duplication and effort and potential for error in re-keying.

The pay phase is made through the payment system. Information about the contract, the kind of goods to be supplied and the price are included in the payment transaction.

The pay phase has three steps:

- seeking approval from the issuing bank to request payment
- upon gaining authorisation, issuing delivery instructions (i.e., forwarding the order to the merchants delivery system)
- transferring transaction data (including VAT and line item details) from the merchant to the acquiring bank for completing payment of previously authorised transactions.

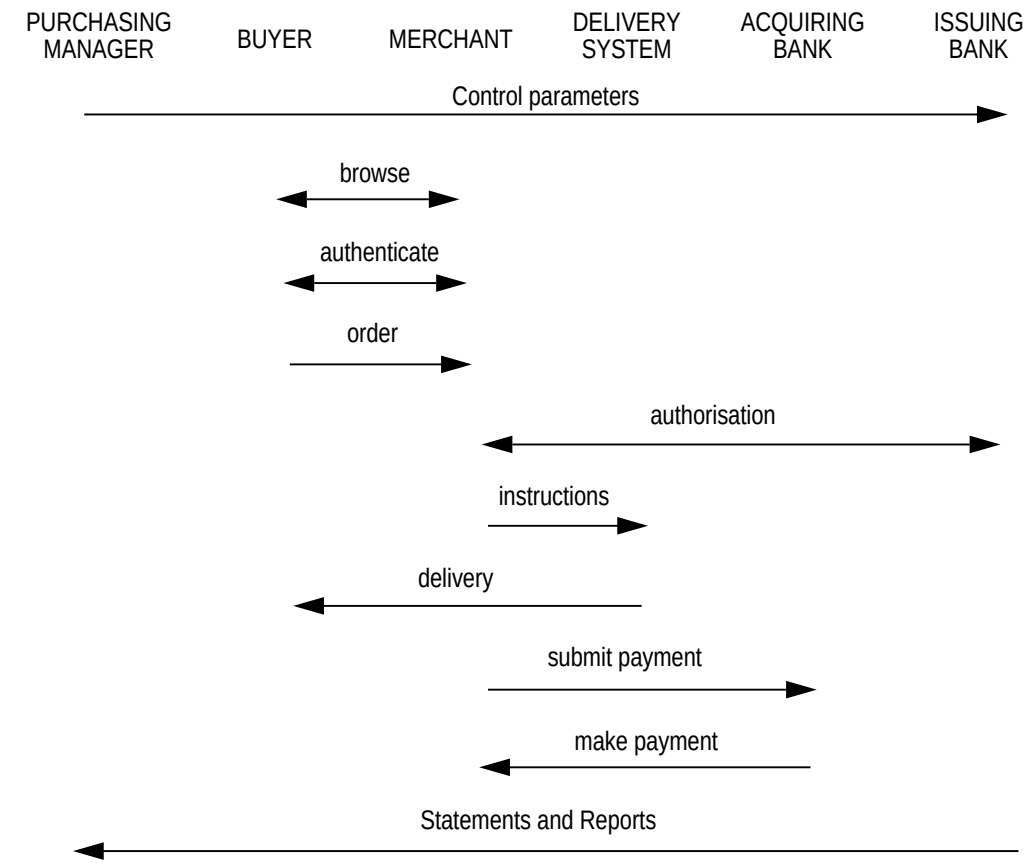
For hard goods delivery falls outside of the scope of E2S, in except as much as elements of the architecture may be used to connect an on-line marketplace to a merchant’s sales order processing system, and for a merchant to signal to a

marketplace when an order has been fulfilled so that the transaction can be submitted for clearing. For soft goods, as in the HP WCSO software licensing pilot, delivery is within the architecture. The software and license are transmitted to the user electronically, protected from theft by cryptography and validated by digital signature.

Finally at some point after the transaction the payment system makes reconciliation information (e.g., consolidated invoices and VAT reports) available to the purchasing managers and the merchants respectively.

A summary of the message flows is shown in figure 2.1 below.

Figure 2.3: Transaction Flow



3 Security model

The purpose of this chapter is to explain the security philosophy and model that underpins the E2S Implementation Architecture.

3.1 Secure electronic commerce

3.1.1 Security policy

Security is a balance of risk against cost; it is not practical to defend against every possible threat particularly when the risk (e.g., financial loss, bad publicity) associated with the threat is small. This in turn means that there is unlikely to be a single security design which meets all the needs of all applications.

For this reason the E2S Implementation Architecture consists of a framework of:

- *system components*
  - — **trusted** components that provide the foundations for security
  - — **untrusted** components that provide the means of delivering services
- *rules* for combining those components to deliver services securely, end-to-end
- *guidelines* for selecting appropriate components to address specific needs.

The use of the implementation architecture is illustrated in Figure 3.3 (taken from [D3] *Security Models and Policies*.).

The figure shows how the architecture for an E2S system (e.g., one of the pilot demonstrators) can be derived from the E2S Implementation Architecture:

- the **system architecture** is developed as a specialisation of the E2S architecture by adding components and functions required to support a **business process**
- constraints on the business process are captured in a **business policy** defining constraints due to
  - — government or industry regulation
  - — “corporate practice”.

Business policy scopes the requirement for security in the system; **security policy** is the outcome of quantifying the acceptable level of **risk** associated with **security threats** against the business policy.

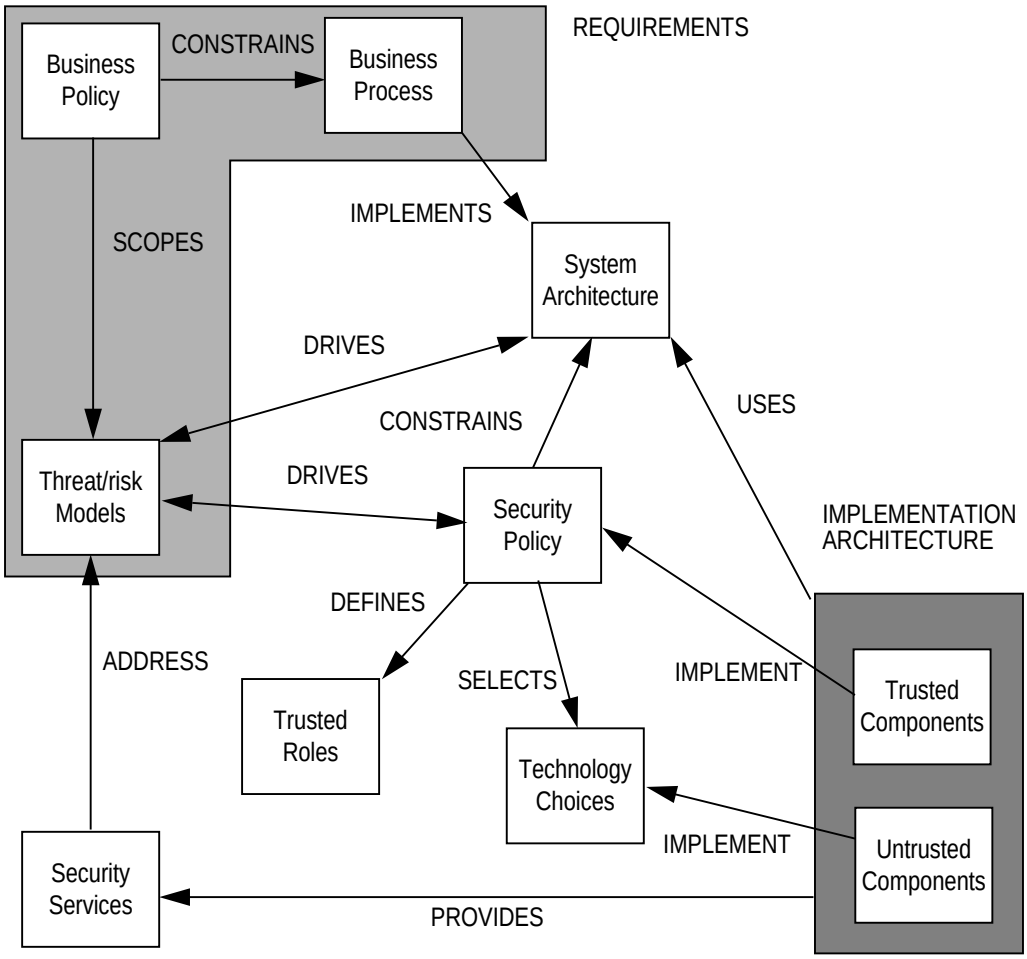
Thus, security policy defines

- the level of trust associated with different **user roles**
- the **trusted components** upon which security is founded

- acceptable **technology choices** from the E2S Implementation Architecture

The E2S Implementation Architecture provides security functions which are used to reduce the risks associated with security threats to an acceptable level.

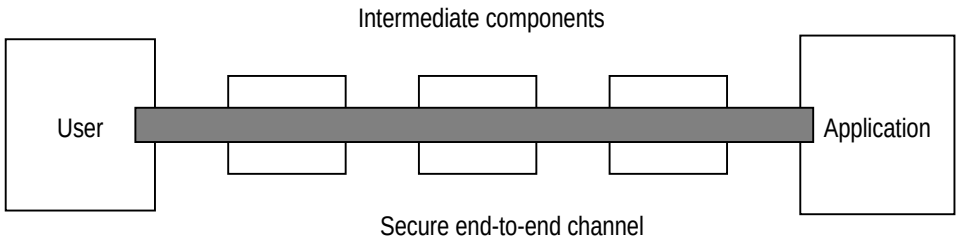
Figure 3.1: Security and architecture



3.1.2 End-to-end security

To argue that a system is secure, the designer must show that business policy, security policy and security services are consistent. For a large system where many components are involved this can be a difficult task. Therefore the E2S Project has focused on **end-to-end security** which is easier to analyse. In simple terms, “end-to-end” describes the approach in which security functions are divided between the user and the application to provide a “secure channel” between them, without requiring any security guarantees from the intervening networks and computers, as illustrated in Figure 3.3 below. This approach is contrasted with conventional “hop-by-hop” security techniques. When correctly applied, the end-to-end approach can rule out “man-in-the-middle” attacks and limits the investment in physical protection to the end systems.

Figure 3.2: End-to-end security



In E2S **smartcards** provide the trusted component at the client side, server-side trust is localised in the use of trusted operating systems:

- smartcards overcome many of the weaknesses of password schemes and the risks of storing cryptographic keys on insecure computers; they are physical tokens of security which brings advantages in usability and manageability
- trusted operating systems provide facilities for partitioning applications and controlling communication between them so that high levels of assurance can be given in terms of system integrity, auditability and ease of management.

From an analysis of user requirements it is evident that the common security requirement of the secure channel is mutual authentication - the parties at either end are sure of each other's identity. Other requirements such as confidentiality and transaction integrity are application specific but can be bootstrapped from a secure session created as a side-effect of authentication.

Therefore, the scope of the E2S trusted components is:

- smartcard infrastructure
- end-to-end authentication
- a tool-kit for building application-specific security protocols
- server-side infrastructure.

There will often be technology constraints which require parts of a system to use hop-by-hop security technology (viz., techniques that are not necessarily end-to-end), for example to created trusted network paths to management interfaces. This is permitted in the E2S Implementation Architecture, but is not included in the security analysis. It is the responsibility of the designer to show that the introduction of conventional security technology has not compromised the integrity of the system.

3.1.3 End-to-End Authentication

From an analysis of user requirements, available standards and technology, the E2S project has chosen a **public key infrastructure** as the means to achieve authentication. Public key infrastructures are a widely accepted technology for user authentication, moreover there is an emerging market of public key infrastructure providers (e.g., Verisign Inc., Ice-tel [Verisign, ICE-TEL]) becoming available.

3.2 Security assumptions

The security of an E2S system depends on:

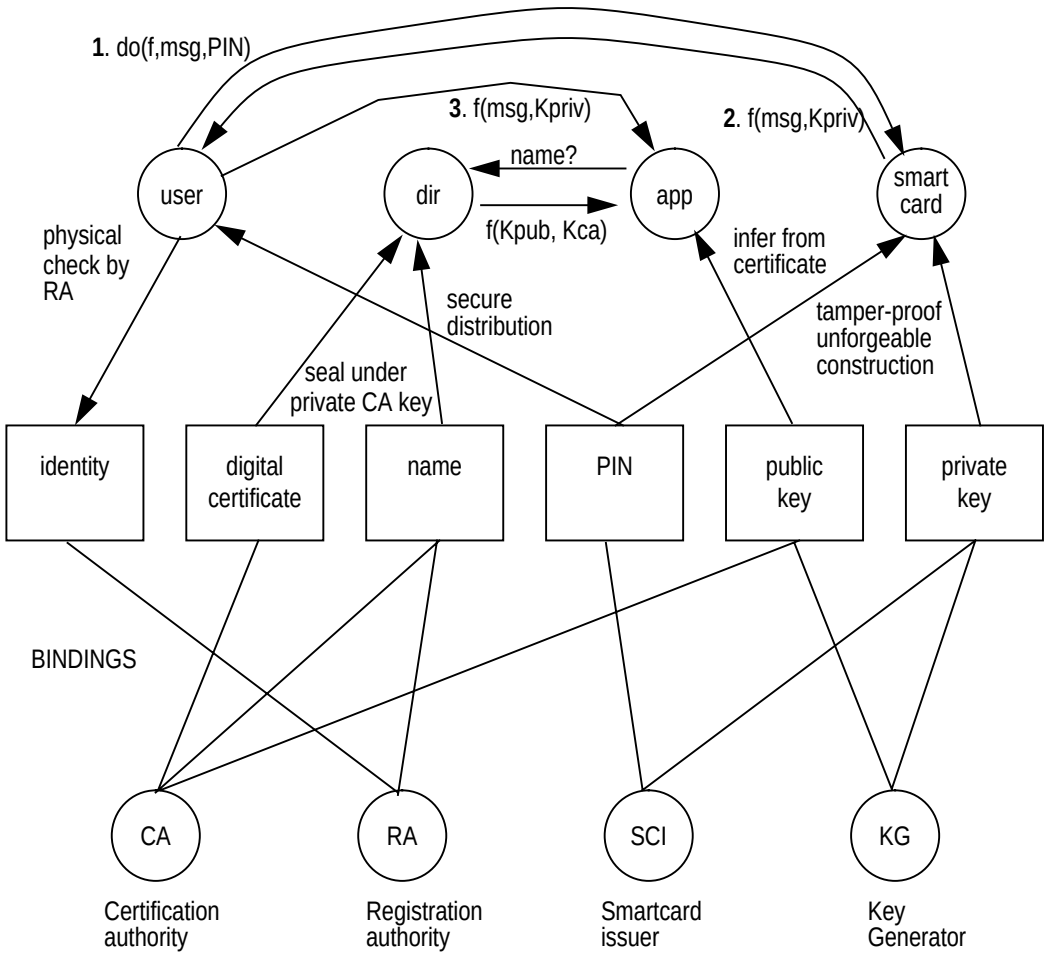
- **public key cryptography**
  - — the private key associated with a public key is a secret
  - — the public key is securely associated with its owner, and this association can be verified globally
  - — the cryptographic functions performed using a private key can only be verified using the corresponding public key (and vice versa)
  - — the private key cannot be predicted from knowledge of the public key (and vice versa)
- **controlled access** to interfaces
  - — based on controlled checking of a user assertion of identity, role or purpose
- **digital signatures** to confirm origin (a digital signature is a bit pattern that can only have been produced by the owner of a private key, and which can be verified only by using the corresponding public key)
- **digital sealing** to confirm content (a digital seal is a cryptographic digest of content that can only have been produced by the owner of a private key, and which can be verified by using the corresponding public key)
- **digital sealing** of time stamps, sequences numbers etc., to confirm timeliness, prevent replay and tie together transaction steps
- **encryption** to provide confidentiality
- the ability to **identify** individual people and roles by name
  - — so that usersand roles can be distinguished within the system in terms of their names
- the ability of **people to keep secrets**
  - — so that transaction steps enabled by knowledge of a secret can be undeniably accredited to a person or organisation
- the use of **smartcards**
  - — as a tamper-proof, unforgeable means to store data (in particular, private keys)
  - — as a means to apply cryptographic functions over both data stored on the smartcard and input data (e.g., encryption of messages under a private key)
- **secure administration**
  - — the minimum requirement is to transfer secrets between system administration components
  - — use of trusted network communications for on-line distribution of secrets
  - — physically secure communication for off-line distribution of secrets (and bootstrap of trusted networks)
- **secure storage**
  - — of access control information

- **trusted operational support**
  - — assured correct operation of security measures for both operational and backup systems
- trustworthy system **administrators**
  - — who control security information.

3.3 Key handling

The use of cryptographic keys in E2S is illustrated in Figure 3.3.

Figure 3.3: Security model



The key to the model are the relationships between the security resources shown in the centre of diagram (identity, digital certificate, etc.). These relationships are maintained by a set of security agents (certification authority etc.) shown at the bottom of the diagram. On the basis of these relationships, the security resources can be distributed between users, smartcards, applications, and directories in such a way that:

- a user can be issued with an individual smartcard
- the user can use that smartcard as a means of initiating mutual authentication across the Internet with an application

- no other user can achieve authentication (even if that user acquires the smartcard)
- this in turn provides the foundation for further interactions to establish access permissions and/or additional security resources (e.g., session keys) required for transaction integrity, non-repudiation and confidentiality.

The basic steps of the authentication process in the figure show the user requesting the smartcard to perform a cryptographic function on a message. The smartcard returns the result of applying the selected function over the message and a private key held within the card. This information can be sent across the network and used by the application to authenticate that it has been sent a message by a the particular user associated with the private key.

3.3.1 Key handling entities

The enterprise objects in the security model are described below (agents are shown in **bold** type, resources in *italic* type):

- a **user**
  - with a distinct *identity*
  - to be authenticated to an **application**
  - trusted to remember, and keep secret a *personal identification number (PIN)*
  - assigned a *name* by a **registration authority**
  - assigned a *private, public key pair*
  - assigned a *digital certificate* by a **certification authority**
- a **smartcard**
  - securely encapsulating a *PIN*, a *private key, a public key, and public key infrastructure* information
  - securely providing on-board cryptographic functions
  - issued to a **user** by a **smartcard issuer**
  - enabled by input of the *PIN*[1]
- a **directory**
  - of *name* to *digital certificate* mappings (a *digital certificate* is a digitally sealed record containing a ***user name*** and an associated *public key*)
- a server **application**
  - protected by access control based on **user** authentication
  - to be authenticated to a **user**
- a **certification authority**
  - for unambiguously assigning *public keys* to *names*
  - for creating *digital certificates* for valid *public key, name pairs*

1. A PIN-protected smartcard requires that the user input the correct PIN when the card is inserted into a reader, to prevent inappropriate use of a lost or stolen card. Thus the PIN is only for access control to the card and could be replaced in the future by biometric recognition for example.

- a **registration authority**
  - — responsible for identifying the **user** and associating an unambiguous *name* with the **user**
- a **smartcard issuer**
  - — responsible for issuing a **smartcard** to the **user**
  - — containing the **user**'s *private key*
  - — enabled by an unpredictable *PIN*
- a **key generator**[1]
  - — responsible for creating an unpredictable *public key, private key pair*
  - — The *private key* is a shared secret between the **smartcard**, the **key generator** and the **smartcard issuer**; communication between these entities must be secured either using trusted networks or safe physical information transfer (e.g., registered post, trusted courier, etc.).

3.3.2 Key handling rules

The smartcard is not a secret. It is only enabled when inserted in a smartcard reader and with the correct PIN input. The connections between smartcard and the PIN input device and between the application software and the smartcard must be secure (e.g. by using a verified copy of a trusted operating system)[2].

The PIN is a secret shared between the user, the smartcard and the smartcard issuer; communication between these entities must be secured either using trusted networks or safe physical information transfer.

The user's public key is not a secret.

The integrity of the binding between a user's public key and the user's name must be trustworthy. This is achieved by having the binding represented as a digital certificate constructed by a certification authority. The certificate must be globally available (i.e., by replicating it widely).

The private key used by the certification authority to sign and seal the digital certificate is a secret and must be kept confidential to the certification authority (e.g., by creating the certificate off-line in a physically secure location).

The public key corresponding to the certification authority's private key is not a secret. It is required to be available to the application. The public key is trusted and should be made globally available by replicating it widely, and by cross-certifying with other certification authorities.

---

1. Key generation is also the point at which key escrow may occur to meet government / business regulatory constraints, and at which keys might be securely archived to enable key recovery after accidental destruction of a key.

2. A particular concern here, especially with personal computer operating systems, is the risk of virus or trojan horse attack either via the network, or via infection of software distribution media (i.e., disks). It is anticipated that during the lifetime of the E2S project, operating systems vendors will improve their defences against such attacks. However, a system designer should take such risks into account when using the architecture, for example, by putting less trust in personal computers that are not under the supervision of a trusted IT administrator.

The registration authority must use physical means to ensure the identity associated with the user and bound to the name is correct (e.g., by reference to legal documents, physical characteristics and so forth).

Since digital certificates are self-describing, directories of certificates need only be protected against denial of service attacks.

The application must include an access control function. Any table of name to privilege rules within this function must be stored securely to prevent tampering, and any communication between an application component and the access control function must be secure if they are not in the same physical location.

The registration authority, certification authority and smartcard issuer must cooperate to ensure that, for a given valid smartcard, digital certificate, user triple:

(i) the name in the digital certificate corresponds to the user

(ii) the private key in the smartcard corresponds to the public key in the digital certificate

(iii) the PIN known to the user is the PIN known to the smartcard.

From these security assumptions it is possible for:

(i) the user to ask the smartcard to perform a cryptographic function on some data

(ii) the transformed data to be sent to the application

(iii) the application to verify that the transformed data could only have originated from the user associated with the name.

This provides sufficient information:

— to enable authentication and hence access control

— to enable the generation of secure tokens and sequence numbers in business protocols for secure transactions.

A user's right to access an application can be withdrawn:

- by changing the application's access control policy
  - — which does not affect the ability of the user to access other applications
- by the certification authority placing the digital certificate in the directory on a certificate revocation list
  - — which effectively "cancels" the user's smartcard, provided applications check the directory as part of validating a user's key
  - — which requires the directory to be highly available and efficient.

3.4 Trusted Operating Systems

Server side security in E2S depends upon the provision of a trusted operating system meeting CMWSEC criteria for evaluating trusted systems [DIA] originating in the needs of government and military system. A computer (client or server) meeting these criteria is called a **compartmentalised mode workstation** (CMW). From the E2S perspective the key features of a CMW are mandatory access control (MAC), privileges, command authorizations and audit.

The combination of these security features makes CMW especially suitable as an application gateway. Some features make it easier to administer and maintain the gateway machine in a secure state and to detect attempts at attack: the detailed auditing, the command authorizations allowing separation of duty and retirement of the root account, and the trusted execution path combating Trojan horses. Other features make it possible to build and run applications securely: MAC and privileges in particular.

Mandatory access controls are enforced consistently by the operating system - users cannot choose which information will be regulated. On CMW all information has associated with it a sensitivity label. The sensitivity label comprises a “classification” and a number of “compartments”. The operating system labels files, processes and network connections. In general, to have read access to some data, a process must have a sensitivity label which “dominates'” the label of the data (i.e., when its classification is higher or equal to the data classification, and when it includes all compartments included in the other label). For write access, a process's label must exactly equal the data's label.

In E2S compartments are used to partition data so that access to separate sets of data is given to different groups of users, e.g., external users, staff, administrators.

CMW supports trusted networking. When communicating with hosts that are not trusted or do not support labelling, the system automatically attaches sensitivity labels to all packets arriving from or sent to the remote host. The label can be applied according to which interface card the packets arrived on or the Internet Protocol address of the remote host. This combines with the MAC features, so the operating system prevents the remote host communicating with processes at other sensitivity levels and accessing inappropriate information.

On CMW, the root account's special powers are replaced by a large set of individual privileges. The relevant privilege is checked by the kernel whenever a process tries to make a system call which could in some way compromise security.

Some of the most dangerous privileges are those which allow a process to override the MAC, and these must be carefully granted to allow selected traffic to cross the firewall. For safety, privileges are only granted to small relay programs which are specially designed and carefully reviewed. These trusted programs allow information to cross compartment boundaries, so that large pre-existing applications can be safely accessed from sensitivity levels other than their own. The trusted programs follow the “least privilege” principle: they raise a privilege only while it is needed for a particular operation and lower it again immediately afterwards.

Command authorizations are the sisters to privileges. They are given to users, whereas privileges are granted to programs. Authorizations allow control over which users are allowed to invoke which trusted programs. By allocating different sets of authorizations to different users, E2S implementations can achieve separation of duties. No single user has absolute control of the system; rather there are a number of administrative roles with complementary powers, for example the separation of access control management from key management in the trust centre (see 6.1.3 on page 35).

The trusted kernel audits system calls, and trusted applications can audit their own actions using a standard auditing subsystem interface. This

auditing cannot be overridden without special privilege. It will normally be configured to log any access denial or insufficient privilege for an attempted operation. Trusted programs can log their actions directly in an easily understood form, so an administrator can track any suspicious behaviour involving overriding MAC without having to decipher long sequences of system calls.

4 Architecture summary

The E2S architecture is summarised diagrammatically in Figure 4.1.

The global areas of technology covered by the E2S implementation architecture are:

- **client technology**, concerned with user interaction
- **secure connectivity technology**, concerned with securing an end-to-end Internet path between users and applications
- **server technology**, concerned with supporting Internet applications

Structurally,

- each technology comprises a set of features from which an E2S implementation can select (such as security management)
- each feature depends upon an underlying set of infrastructures (such as key management)
- each infrastructure is made up of a number of architectural components.

The set of features included in the scope of the architecture has been driven by an analysis of user requirements and a desire to maximise the use of common technology. Thus, the design of the E2S Implementation Architecture is intended to enable the re-use of infrastructure components across a wide set of features and hence application scenarios. The mapping of features to demonstrators is described in [E1] *Implementation Plan*.

A system conforms to the E2S Architecture if it makes a consistent choice of features from each area and satisfies the security model specified in Chapter 3.

The rules for making consistent choices are specified in the separate reports giving the detailed architecture for each feature and associated infrastructures. The top-level description in this document focuses on the major relationships and the overall management of security in E2S.

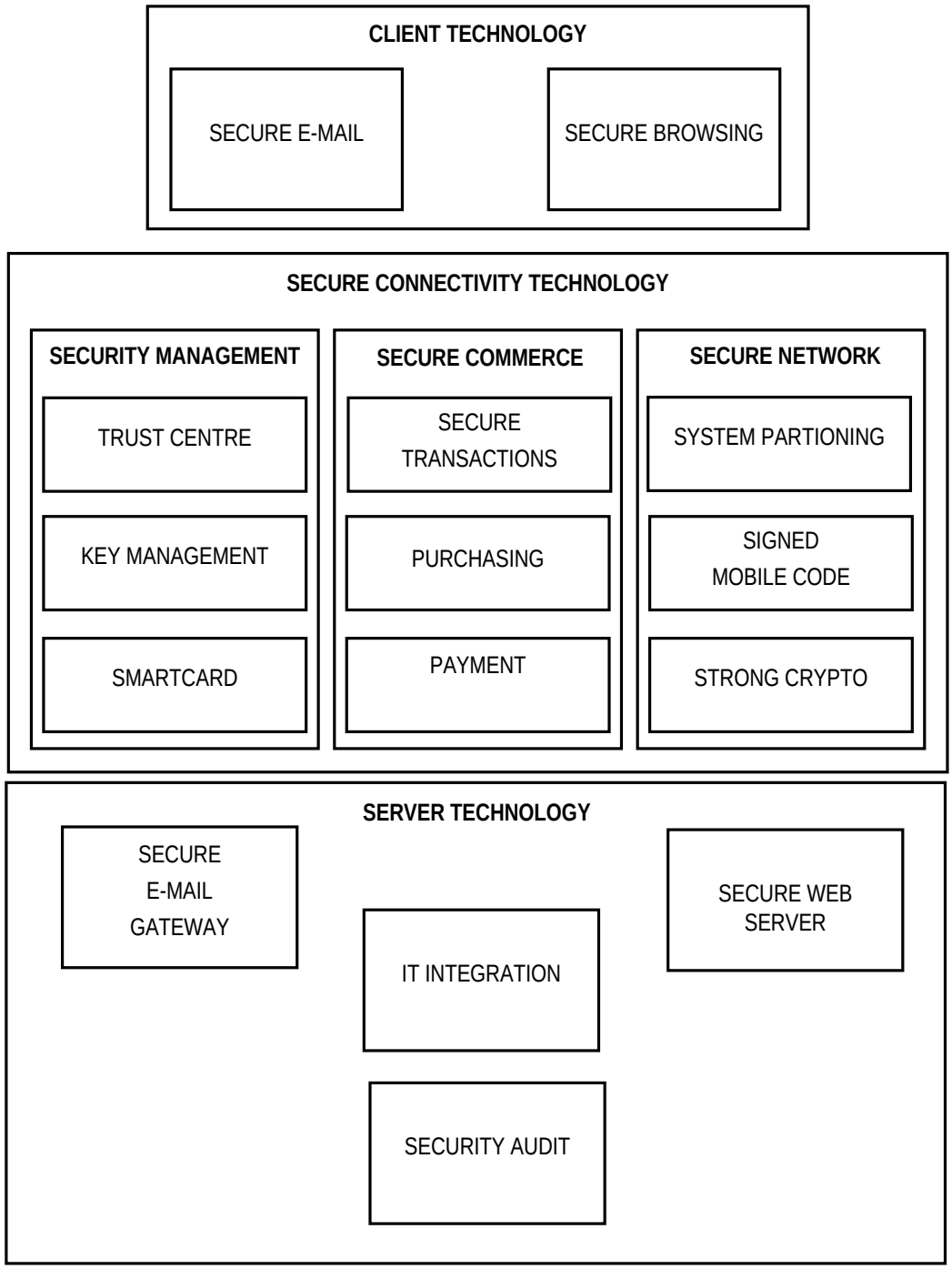
The architecture can be used recursively, in the sense that management functions for E2S components can be implemented themselves as end-to-end secure applications (for example, an HTML forms-based interface for updating a registration authority's directory service). In using the architecture recursively the designer must be sure that cyclic dependencies are not created.

4.1 Client technology

Client technology is used to provide the interface between users (i.e., on-line customers) and the servers which are providing electronic services to those users.

The need for two kinds of client features have been identified in the analysis of user requirements:

Figure 4.1: Architecture Summary



- **secure electronic mail** for **telecooperation** involving the secure exchange of messages, documents and instructions
  - for use where business processes and data protection regulations are formulated in terms of document handling policies. It is also well suited to applications serving users who may not be on-line continuously (e.g., out-of-office sales staff with laptop computers) or where an asynchronous mode of operation is more appropriate.
- secure web browsing for transactional **interactive sessions** to enable activities such as searching, selecting, ordering and reporting.

— for applications where rapid access to a wide range of information and a fast response is required. It requires that the user remain on-line for the duration of a session (e.g., to purchase a selection of goods).

Associated with both kinds of client technology is the need for **user authentication** based on **smartcards**.

4.2 Secure connectivity

A pre-requisite of end-to-end security is connectivity technology to secure interaction between clients and servers.

Analysis of the end-to-end security aspects of E2S user requirements shows the need for three features to secure connectivity:

- **Secure network infrastructure**
  - components used to protect trusted applications and network links
- **Secure commerce infrastructure**
  - components that ensure that electronic business transactions representing the sale and purchase of goods and services, and the associated financial transactions including payment are conducted correctly and securely
- **Security management infrastructure**
  - components for establishing, verifying and revoking user roles and access privileges and their representation as digital identities, and cryptographic keys sealed within smartcards.

4.2.1 Secure network infrastructure

Secure network infrastructure comprises three components:

- **system partitioning** using compartmentalised mode workstations for delineating and controlling entry and exit between security domains. The foundation of security management and auditing
- **signed mobile code** (such as browser plug-ins or Java applets) as a means to distribute application components to users
- **cryptography** as the foundation of authentication, confidentiality. integrity and non-repudiation.

4.2.2 Secure commerce infrastructure

E2S user requirements show the need for three features in secure electronic commerce:

- **secure transaction infrasture to support** end-to-end procedures (e.g., browse, order, pay, deliver) and maintain appropriate levels of privacy, obligation and non-repudiation between interacting parties through the duration of the procedure
- a **bankcard purchasing infrastructure** (including a network of supporting banks) for electronic business-to-business corporate purchasing (e.g., of office supplies), based on corporate bankcards
- an electronic **payment infrastructure** enabling electronic payments.

Bankcard based payment has been selected for E2S since it is international in scope and has a well-understood financial risk model compared to other forms of electronic payment. On-line use of bankcards is facilitated by using the *Secure Electronic Transactions* [SET] standard.

Pilot SET infrastructures are being created in the time-scale of the E2S project and SET components for card-holders and merchants are integrated into the E2S produced components.

4.2.3 Security management

To support the E2S security model, three features of security management are required:

- a **trust centre** for maintaining **security attributes** that represent trust relationships
- a **key management infrastructure** for making, distributing, checking and revoking cryptographic keys used for authentication and access control
- a **smartcard infrastructure** for issuing and verifying smartcards.

4.3 Server technology

Analysis of E2S user requirements shows the need for two different features for delivering secure services to users, one based on electronic mail, the other based on secure user sessions.

Alongside this user-facing functionality is a need to integrate electronic commerce technology with “back office” applications.

To meet a requirement for continued assurance of system security there is additionally a need to monitor and audit server technology.

E2S server technology comprises:

- a **secure e-mail gateway** infrastructure to act as the focus for applications based on secure email. The server provides secure mail boxes and functions such as re-distribution of mail directed to an organisational unit
- a **secure web server** infrastructure acting as the focus for user sessions initiated by client browsers
- **IT integration** infrastructure enabling back office applications to be exported via mail gateways and web servers. It is concerned with the connectivity between the Internet (via which clients access services) and internal “Intranets” on which services are deployed. IT integration includes the capability to download “applets” from the server to the client to enable customisation of the client interface.

In some situations, for example the SBCW pilot (see 8.3), connectivity is server-to-server, rather than client-to-server. The buyer organization hosts a server which is downloaded from the merchant. This downloaded server then supports access to the merchant. The features of this structure are that the buyer organisation has local autonomy over the management of buyers, and the downloaded server can, if appropriate, be given access to the buyer’s IT infrastructure (e.g., to access additional information, deposit audit records etc.).

5 Client technology

Client technology enables a user (i.e., a **person**) to interact securely across the Internet with an **application**.

It comprises:

- secure electronic mail (email)
- secure browsing.

5.1 Secure electronic mail

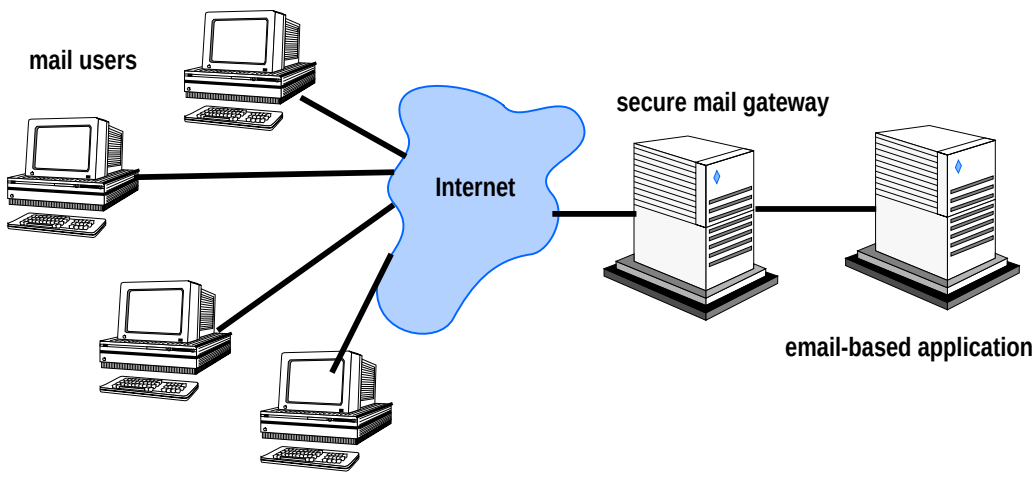
Secure electronic mail is required for applications where business processes and data protection regulations are formulated in terms of document handling policies. It is also well suited to applications serving users who may not be on-line continuously (e.g., out-of-office sales staff with laptop computers) or where an asynchronous style of interaction is appropriate.

Secure electronic mail enables **telecooperation** based on the secure exchange of messages, documents and instructions for:

- publication of authentic information
- confirmed delivery of information
- secure access to sensitive information.
- Secure electronic mail is described in [E.4] *Secure Telecooperation Software.*

Secure electronic mail is illustrated in Figure 5.2. A set of mail users can communicate securely with one another by electronic mail and interact with applications (such as document servers) by sending commands and receiving results as mail messages.

Figure 5.1: Secure email scenario



The **secure mail gateway** (see 7.1 on page 47) acts as a repository for messages in transit and also in the extended form of a **secure mail exploder** provides support for sending mail to **distribution lists** identifying **group**s of users.

Thus secure electronic mail can support user-to-user, user-to-business and (if a program is substituted for a mail user) business-to-business electronic commerce.

Mail users require, in different situations, combinations of

- confirmation of the origin of messages
- confirmation of the content of message
- guarantees that messages will only be delivered to the intended recipients
- confidentiality for messages.

Therefore electronic mail uses **the key management infrastructure**, **secure client mailers** and **secure mail sessions** to sign, seal and encrypt mail messages appropriately for the transactions taking place.

Two kinds of secure client mailers are defined in E2S:

- **security enhanced mailer**
- **protected insecure mailer.**

5.1.1 Security enhanced mailer

A security enhanced mailer is one which supports cryptographic sealing and signing of messages using for example, Privacy Enhanced Mail (PEM) [PEM] or Pretty Good Privacy (PGP) [PGP] technology.

A security enhanced mailer requires cryptographical functions using the user's private key. If the mailer runs on a computer accessible to other users, the functions on the user's key should be accessed via a **smartcard**[1].

A security enhanced mailer exchanges mail with

- **secure mail gateways**
  - — to communicate with users with **protected insecure mailers**
  - — other security enhanced mailers
  - — to communicate with user **group**s.

5.1.2 Protected insecure mailer

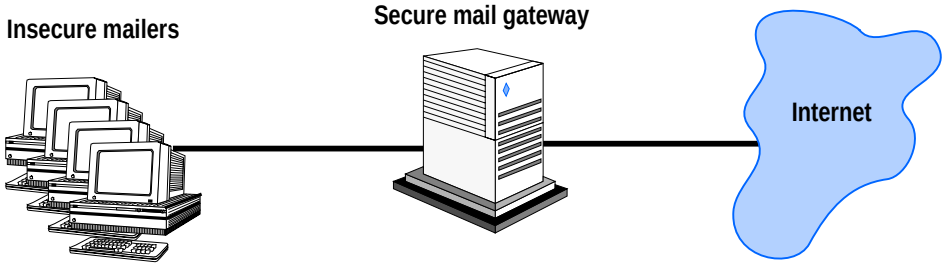
An important requirement for electronic mail-based electronic commerce is to provide access for users with mail packages which have no built in support for security.

Insecure mailers must be protected by a **secure email gateway**, as illustrated in Figure 5.2.

All mail to and from the workstations to other computers is intercepted by the **secure mail gateway**. This gateway cryptographically signs, seals and possibly encrypts outgoing mail according to a security policy. The gateway

1. An alternative, when the associated risks are acceptable, is to store user's private keys on the computer's disc, encrypted by a password / pass phrase. Whenever the mailer needs to access the key, the user is requested to input the password and the mailer decrypts a temporary plain text copy of the key which is destroyed after use.

Figure 5.2: Protected insecure mailers



checks the seals of incoming mail and decrypts it if necessary, appending to the plaintext contents an explanation of the trust that may be put in the message (e.g., indicating the message was signed by a particular individual and sent confidentially).

Insecure mailers must be subject to **security audit** to ensure that:

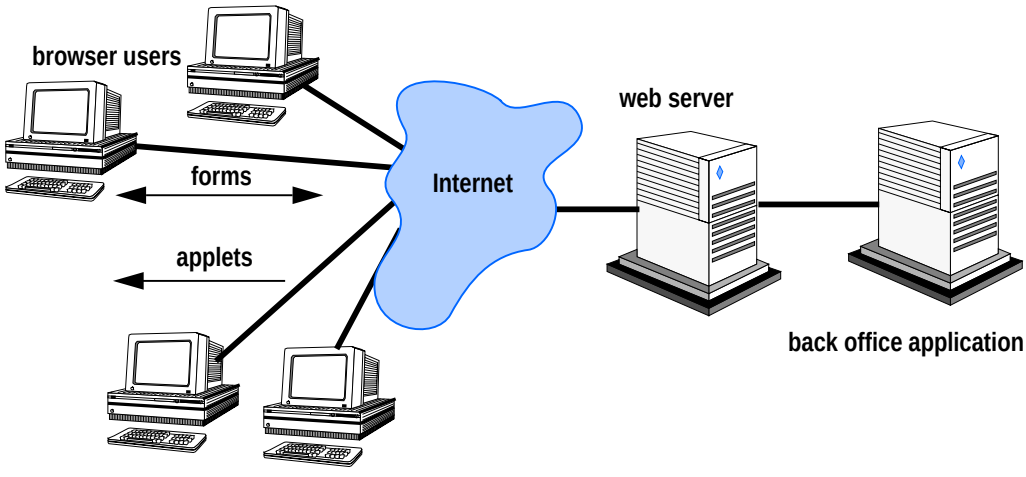
1. insecure mailers are isolated from the Internet e.g., by positioning behind a **firewall** to avoid either "spoof" mail being accepted or sensitive mail being allowed to escape
2. one user does not masquerade as another (e.g., by associating passwords / pass phrases with names, or putting computers in secure locations associated with named users).

5.2 Secure interactive sessions

Secure interactive sessions meet the user requirement for interactive on-line applications of electronic commerce.

The basis of secure interactive session is web browsing, as illustrated in Figure 5.2. A Web browser (e.g. Netscape Navigator [NETSCAPE]) is shown, providing a page-oriented interface to a Web server (see 7.2 on page 48) over which HTML [HTML] and other forms of document can be displayed to the user. HTML includes the provision for the return of filled forms from the user to the server, either for processing within the server, or for hand-off to a back office application through **IT integration** technology.

Figure 5.3: Web browsing scenario



The browser user requires:

- a guarantee that the session is with a server under the control of a trusted business
- the forms downloaded and returned are not tampered with (and in some situations remain confidential).

Web browsing can be made secure in several ways:

- **secure http sessions**: the browser and server can support transport level security protocol such as the secure socket layer protocol [SSL]
  - — if neither the browser user or the server have a digital identity SSL simply gives confidentiality
  - — if the server has a digital identity the server can be authenticated to the browser user
  - — the browser user send enter a user name, password to an authenticated server
  - — if the browser user has a digital identity this can be used to authenticate the user to the server[1]
  - — however in all these cases only the path from the browser to the server is secured and therefore is only suitable for secure access to information held on web server itself
- **secure application sessions**: a web page may include a reference to a **mobile code module** (an **applet** or a **plug-in**) implements a **secure transaction protocol**
  - — Plug-ins can be pre-configured, or installed on demand, depending upon the specific browser in use and the client's firewall policy with respect to mobile code)
  - — a plug in can intercept a form completed by the user and sign and/or encrypt it using keys from the user's smartcard, and the transfer the content in this protected wrapping to the web server (i.e., the form is transferred as S/MIME data rather than "plaintext" HTTP). This approach is used in the **secure commercial purchasing protocol** (see 6.2.1.1 on page 38)
  - — an applet has the choice of either using S/MIME and HTTP, or if permitted to open a direct network connection to the server and use another, potentially custom, secure application session protocol. (See for example [E2.1/E2.6] *Star System Access Control Software*).

If mobile code modules are used the browser user must be protected against malicious or accidental errors in those modules or substitution of false modules. Mobile code modules should either be downloaded via authenticated secure http connections and/or, as described in 6.3.2 on page 44, digitally signed.

The authenticated source/signature authority should be used to enforce security policy on the downloaded code.

1. Current browsers store user digital identities on disc which is potentially insecure. The E2S implementation use a plug-in to take the digital identity from a smartcard and shuts down the secure session if the smartcard is removed. In an early implementation based on user name, password checking the protection against a stolen session was achieved by using timeouts to detect and shutdown idle sessions.

6 Secure connectivity technology

Secure connectivity technology provides end-to-end security across the Internet between clients and applications. It comprises:

- security management
- commercial purchasing
- secure networking.

6.1 Security management

Security management is primarily concerned with the maintenance of trust relationships through the management of the cryptographic keys used in secure communications and secure transactions.

Security management comprises:

- trust centre infrastructure
- key management infrastructure
- smartcard infrastructure.

To reduce forward references in the description, the trust centre infrastructure is described last.

A summary of the major components of the key management and smartcard infrastructures is shown in Figure 6.1. The figure shows how the components divide into three groups:

- those associated with clients (i.e., security enhanced mailers and web browsers)
- those associated with servers (i.e., secure mail gateways and web servers)
- those associated with security administration.

The arrows in the figure denote the principal service requests that occur between the components.

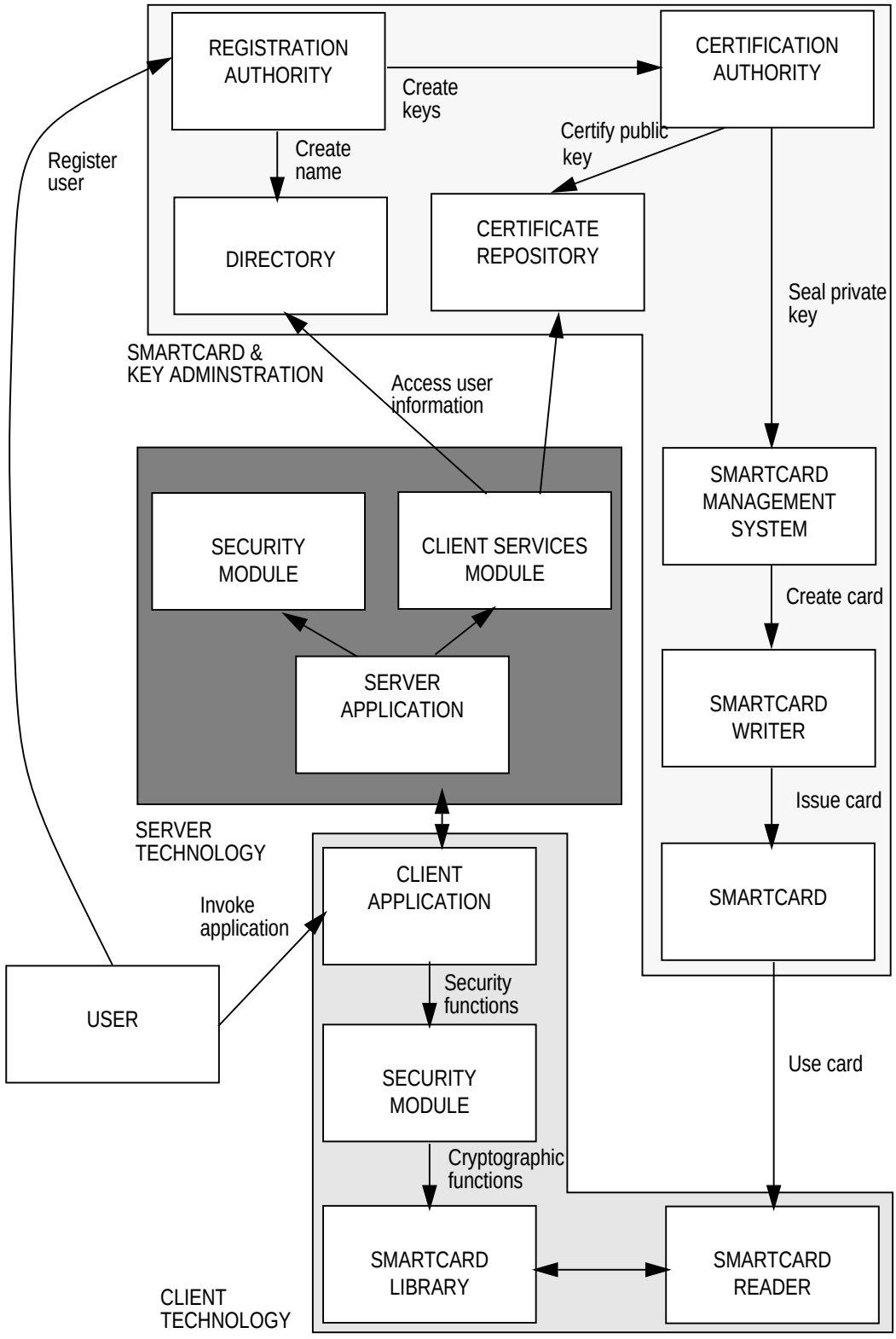
The implementation of this part of the architecture uses the SecuDE [SecuDE] and Osisec [OSISEC] toolkits based on X.500 directory services [X.500], X.509 digital certificates [X.509], PEM [PEM] and PGP [PGP] technology and is described in deliverable [E2.2] *Key Management and Smartcard Infrastructure.*

6.1.1 Key management infrastructure

Secure communication based on public key cryptography relies on securing the association between cryptographic keys and their owners as explained in Chapter 3.

The functions of the key management infrastructure are:

Figure 6.1: Smartcard and key management



- **personal security environment (PSE) management -** i.e managing all of a users personal security attributes (Distinguished Name, Keys, Certificates, etc.)

  - PSE management allows key management infrastructure managers to define and generate a PSE (i.e. a set of personal security attributes) to be distributed to a user
  - PSE management allows applications (e.g., a browser) to access a set of security attributes.
- **key generation -** i.e. generating secret/public key pairs
  - key generation can be devolved to the owner of the key (as in PGP) or it can be under the control of the key management infrastructure manager
  - if keys are created by the infrastructure manager they must be distributed securely to the owner (e.g., embedded in a smartcard)
  - if a secret key is distributed other than on a smartcard, the key owner must store it securely
  - the channel between key generation and key storage must be fully secure and as short as possible.
- **key escrow -** i.e. saving a user security attributes, including private keys in a secure vault where they can be made available on specific conditions defined by the security policy followed by the infrastructure manager.[1]
- **key recovery** - i.e. restoring a user key package including private keys
  - complementary to key escrow
- **key distribution -** supplying a PSE to a user:
  - this feature is limited when key package is generated by the end user.
  - the key package can be distributed by **secure http sessions** with **user authentication**[2]
  - the recommended way to distribute key package is to store it on a **smartcard**.
- **certification** - i.e., associating a key to a name in the form of a **trusted digital certificate**
  - certificates are held in on-line **directory services.**
- **registration** - i.e., associating a name to a person
  - registration information is held in on-line **directory services.**
- **attribute assignment** - i.e., associating an attribute to a person, for example
  - a **role** (e.g., “head of department”)
  - a **capability** (e.g., “access to sales statistics”)
  - **context** information(e.g., “for bank use only”)

1. Key escrow allows decryption of enciphered messages by corporate security managers or by legal agencies, to meet government / business regulatory constraints. Key escrow is not in itself an E2S requirement. Before entering into a trust relationship based on cryptographic protocols each party must be sure that the joint policy with respect to key escrow and key recovery is compatible with their own security needs.

2. However if the secure http session is based on a weak method of user authentication (e.g. user name and password it may be easily compromised the distributed key stolen

verification - that a purported key can be trusted for the purpose to which it is applied (e.g., “Giles S. Murchiston, acting in the role of repairs budget holder, purchasing printer spares”)

The components of a key management infrastructure are:

- **key generator**:
  - — responsible for issuing key pairs
  - — in the E2S implementation this module is implemented as a local module of the SCMS platform and is provided by SecuDE.
- **certification authority**:
  - — responsible for issuing digital certificates and certificate revocation lists
  - — in the E2S implementation this module is a local SecuDE object, directly linked to the SCMS platform.
- **client services module**:
  - — providing access to the directory and certificate repository for **secure mail gateways**, **web servers** and **IT integration** components
  - — in the E2S implementation this module is an LDAP client
- **certificate repository**:
  - — responsible for storage and retrieval of certificates
  - — in the E2S implementation this module is an LDAP Directory
- **user security module**:
  - — responsible for giving access to the PSE and performing cryptographic functions like signature, verification and encryption
  - — in the E2S implementation this module is built using SecuDE
  - — Several PSE implementations are possible:
    - (i) software PSE: the user attributes are stored on disk in DES enciphered files, protected by a pass phrase
    - (ii)smartcard PSE: the user security attributes and cryptographic functions are entirely stored in a smartcard
    - (iii)smartcard PSE extension: a software PSE containing (principally) static non-confidential security attributes (e.g., certificates) and cryptographic functions linked to an enabling smartcard containing dynamic, confidential attributes. The extension can be distributed by CD-ROM or network access for more specific or dynamic information. The smartcard unlocks the extension
  - — Several SecuDE implementations are possible:
    - (i) SecuDE tool kit: the user application access security services running SecuDE commands with SecuDE script files.
    - (ii)SecuDE programmable API: the user application includes a SecuDE DDL library
  - — in the E2S implementation this module the Bako secure commercial purchasing protocol accesses SecuDE using the PEM API. The

Smartcard Management System (SCMS) gives access to SecuDE using the Nortel CMS-API.

- **key administration**:
  - an interface to the infrastructure, for policy management and audit.

The inter-relationship and structuring of instances of these components depends upon the needs of the particular application and community of users. In particular two cases are supported:

- **external registration and certification authorities** such as Ice-tel and Verisign Inc. [VERISIGN, ICE-TEL] in systems where access is to be granted to the general public (i.e., users are not pre-registered)
- **internal certification and registration** authorities where the service provider delivers key management alongside a service, either as part of that service, or to control the "branding" of the service, or to restrict use to a registered set of users.

The **trust centre** provides an integration of key management, certification, registration and access control functions in one functional unit.

The security of an E2S system depends upon the security of both certificate repositories and security modules. Both should be subject to **security audit**.

Certificate repositories and directories should be secure and the interface for adding and/or changing keys and relationships made available only to trusted **key management infrastructure security manager**s.[1]

Registration authorities are required to take prudent steps to be sure the person they associate with a name is the appropriate individual, for example by checking passports or similar legal identifications against the person claiming to associated with the name being registered.

6.1.2 Smartcard infrastructure

E2S has selected smartcards as the preferred means of issuing keys to users and for key verification because a smartcard is:

- a personal, physical token - to use it requires both the presence of the card and knowledge of a secret personal identification number (PIN)
- a strong, tamper-proof and unique location for a user's private key
  - because of the card's physical properties the key cannot be read, duplicated, moved or falsified.

A smartcard infrastructure consists of **smartcard technology** and a **smartcard management system**.

The security of smartcards depends upon the PIN remaining a shared secret between the smartcard issuer and the smartcard user.

6.1.2.1 Smartcard technology

Smartcard technology consists of

- a smartcard architecture

1. The protection of the certificate repository can be reinforced if the root key is used to create a set of "operating keys", and the repository for the root key then disconnected from any network and stored in a safe. If an operating key is compromised, an alternative operating key can be substituted. If the root key is compromised there is no means of recovery, except to re-issue all keys.

- a smartcard reader/writer device
- a software library for access to smartcards via the reader/writer device.

E2S requires a smartcard architecture in which a smartcard can

- store data (keys, certificates) with a high level of security
- verify digital certificates
- sign and verify blocks of data (e.g., protocol messages)
- generate truly random numbers
- encipher/decipher data.

(An architecture with “on-board” cryptographic processing allows the card to be used with relatively insecure operating systems, since they keys in the card cannot be read, compared to keys on a local disk.)

The E2S implementation has selected the GEMPlus GPK2000 card to meet these requirements. It supports:

- RSA, DSA and DES algorithms
- true random number generation
- SHA-0, SHA-1 and MD5 hashing
- storage of application data.

6.1.2.2 Smartcard management system

The smartcard management system provides the link between the **smartcard infrastructure** and the **key management infrastructure**.

The smartcard management system is responsible for issuing smartcards to users:

- receiving a set of private keys from a public key infrastructure manager
- creating a smartcard encapsulating the keys and assigning the PIN
- securely delivering the smartcard and knowledge of the PIN to the smartcard user (e.g., by postal mailing in separate packages).

The communication between the smartcard infrastructure and the key management infrastructure must be secure[1] and subject to security audit to avoid the loss or compromise of keys before they are encapsulated within a smartcard.

6.1.2.3 User authentication with smartcards

Smartcard based authentication is the recommended form of authentication in E2S. The guarantee of mutual authentication between the user and the business providing the server he is using must persist only for the duration of a session. If the user removes his smartcard the client technology should request re-authentication before processing further interactions.

All of the client technology for user authentication (computer, operating system, keyboard, card reader, security software) must be verified to be free of security flaws and immune to virus or trojan horse attack.

1. This security can be achieved by co-location of key generation and smartcard issuing, by using secure transport between the two infrastructures or by recursive use of the E2S architecture (i.e., treating card issuing as an application).

6.1.3 Trust Centre

The **trust centre** provides an integration of key management, certification, registration and access control functions in one functional unit.

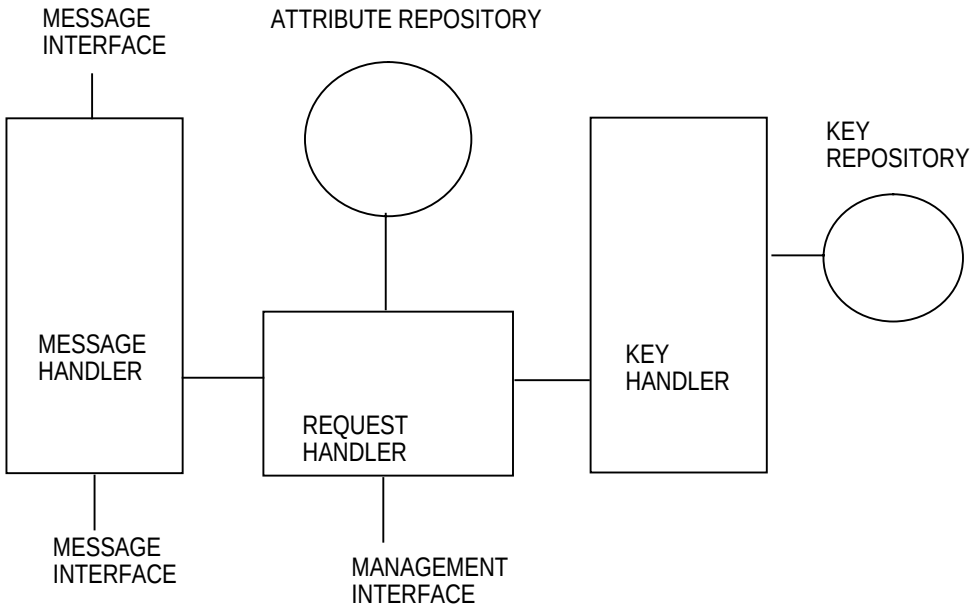
The purpose of a trust centre is to maintain manage trust relationships between individuals, roles and authorities and deliver authorisation decisions.

For example some business processes require both proof of identity and proof of ability to pay. Both of these can be represented as certificates, one issued by a **certification authority**, the other by a **payment infrastructure**. A trust centre takes the certificates, **verifies** them individually and then delivers a decision (perhaps in the form of another certificate - i.e., a **capability**) based on the results of the verification and an access control policy.

Often it is convenient to associate authorities with roles: for example, a content manager has the privilege to upload new content to a marketplace catalogue. However the individual in an organisation that fulfils the role of content manager may change over time. Therefore in addition to authorisation decisions, the trust centre provides individual to role assignment mapping and checking.

The structure of a trust centre is shown in Figure 6.1.

Figure 6.2: Trust Centre



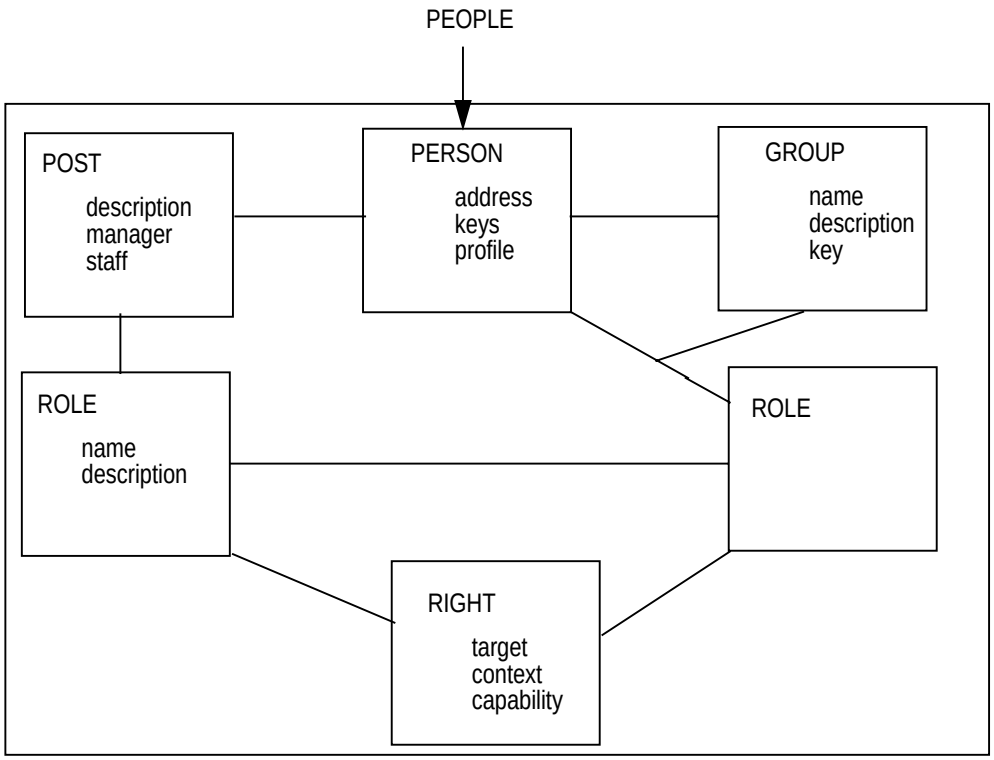
- Transactions are received by the message handler
  - — in the case of secure email the message handler can be integrated with the mail exploder component of a secure email gateway since the transactions will be requests to deliver messages
  - — in the case of secure transactions the message handler will process transaction steps that require authorisation checks
- The message handler converts the message into a request for an authorisation check and invokes the request handler. It does so by

inspecting plaintext routing information in the message (i.e., mail headers for email, URLs for HTTP interactions).

- The request handler checks the validity of the request against a database of trust attributes
- In making the check the request handler may invoke the key handler to perform cryptographic functions on the request
- the key handler performs cryptographic operations on data using keys held in the key repository
- the request handler provides a management interface to manipulate the information model held in the attribute repository
- the attribute repository may be self contained or it may access certificates from **directory services**.

The information model is shown in Figure 6.1.

Figure 6.3: Trust Centre Information Model



- rights are the basic items in the model which permit specific actions on a target. The right may be restricted to a context in which it can be exercised. A capability identifies a right in detail and specifies information to be supplied to an access control component to enable the right to be exercised
- rights are assigned to roles (e.g., “buying manager”). Roles are an abstraction of some function in the organization
- people within an organisation are regarded as instances of posts (e.g., “financial controller”. Roles are ascribed to posts. During their career within an organisation a person may hold different posts at different times, and sometime several posts simultaneously

- in addition, roles are assigned to groups (e.g., “the audit department”)
- all items in the model can have attributes to relate the information to the organisation (e.g., location, key, name).

The trust centre is a critical security component and the use of trusted operating system facilities (least privilege, secure partitioning) alongside security auditing is indicated to ensure:

- only trusted credentials managers can change the attribute repository
- access policy rules are stored securely
- keys are store securely.

In the E2S implementation a key handler using PEM and SecuDE or an alternative based on PGP is available.

6.2 Secure commerce

The secure commerce feature of the E2S architecture consists of the infrastructure needed to select, purchase and pay for good and services securely. The secure commerce feature comprises three infrastructures:

- secure transactions infrastructure
- electronic purchasing infrastructure
- corporate payment infrastructure.

The secure transaction infrastructure is responsible for the overall coordination of the cycle linking each step together. The electronic purchasing infrastructure provides the means to link together the IT of the buyer, the marketplace and the merchant. Within the purchasing infrastructure, the payment layer provides the means for the buyer to pay the merchant for the good or services delivered.

6.2.1 Secure transactions

Secure transactions are the electronic analogues of the procedures entered into every day by businesses and people e.g., when updating records, ordering goods or buying services.

The secure transaction infrastructure must therefore provide:

- confidentiality - so that transactions are only visible to the participants involved
- integrity - the transaction follows a correct procedure
- authentication - the participants in a transaction are convinced of each other’s right to undertake the roles they fulfil in the transaction
- non-repudiation - at the end of a transaction each participant has proof the transaction took place.

There are four kinds of secure transaction infrastructure in the E2S implementation architecture:

1. secure mail sessions for telecooperation
   - — the “transactions” are mail oriented steps such as sign, encipher, send, read, verify, decipher
2. secure http sessions (SSL) for secure browsing

3. secure application sessions for end-to-end secure transactions
4. secure commercial purchasing infrastructure - a specific secure transaction infrastructure for “browse, order, pay, deliver” transactions.

6.2.1.1 Secure commercial purchasing infrastructure

The E2S secure transaction infrastructure is based on the Bako protocol [BAKO, E2.8, E2.5] and implements the browse, order, pay part of the interactions shown in Figure 2.1 on page 5.

The protocol consists of a plug-in to the buyer’s browser and a script in the merchant’s web server. These two components in turn use SecuDE to implement a PSE holding keys etc. The buyer’s PSE is held on a purchasing smartcard.

1. the merchant sends an unsigned login page to the buyer
2. the buyer creates an X.509 conform message and sends it to the merchant
3. the merchant verifies the signature and the content of the log-in page, generates an X.509 conform message and returns it to the buyer together with the URL of the private catalogue and a session identifier (as a cookie, a hidden field or an extended URL)
4. at this point buyer and merchant are mutually authenticated. The received session id is stored by the buyer and sent enciphered with every request for a new catalogue page
5. after the buyer has collected all the goods, the merchant prepares a signed offer and sends it to the buyer
6. the buyer verifies the signed offer and if the offer is acceptable signs it with the buyers signature and returns it to the merchant
7. the merchant verifies the buyer’s acceptance signature, signs it again and return a copy to the buyer as proof the order was accepted
8. the buyer then initiates payment using the purchasing infrastructure (see 6.2.2 below)
9. if the goods are to be supplied electronically (e.g., software or other digital media) the merchant signs the goods and sends them to the buyer.

Each of the message is transferred using the normal HTTP protocol between browser and server. However the browser plug in and server script convert the HTML pages being exchanges to S/MIME blocks so that signatures can be associated with the content and transferred from end-to-end. In addition, if confidentiality is required the pages can be enciphered.

The client software implementing the commercial purchasing infrastructure is sometime described as the E2S **commercial wallet**.

6.2.2 Bankcard purchasing infrastructure

The bankcard purchasing infrastructure handles ordering, record keeping, payment and reconciliation of accounts between buying organizations, merchants and banks. It is operated by a **bankcard association**.

The purchasing infrastructure distributes accounts and reports on purchases made and received, taxes due etc., to users and merchants as appropriate. Additionally it ensures the corresponding payments are made and received using the payment infrastructure.

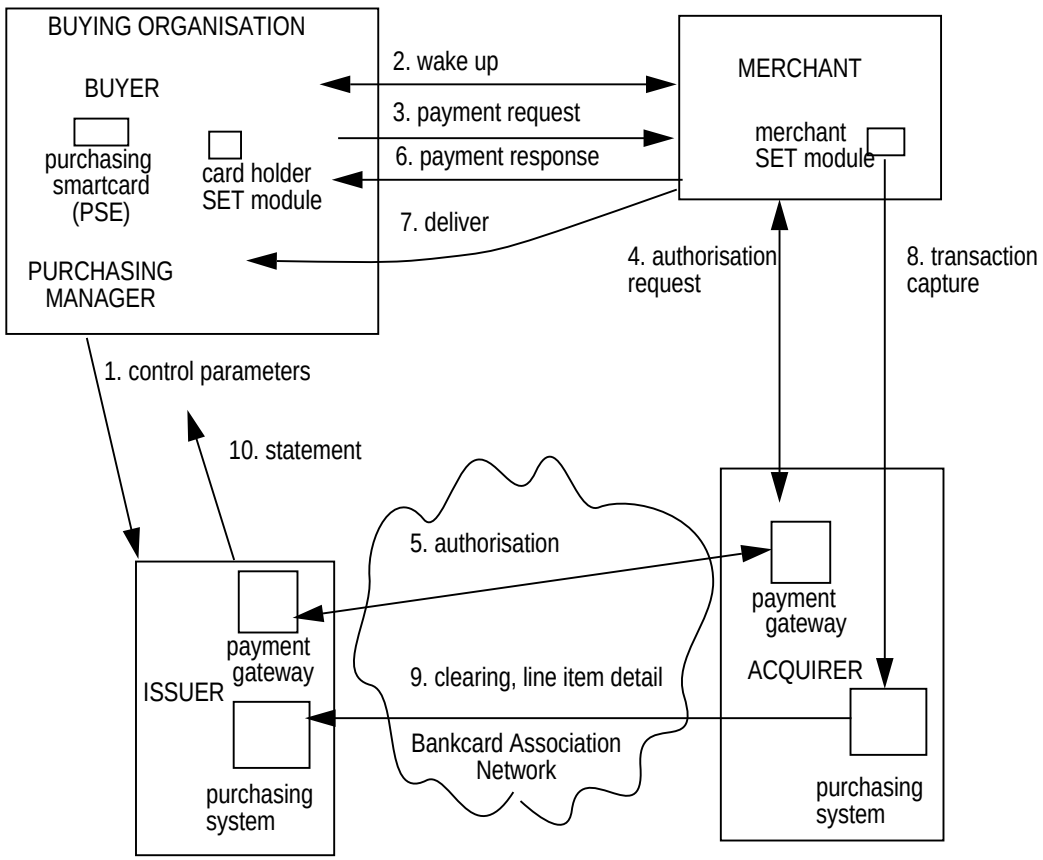
The top-level architecture of the purchasing system and the key roles within were described in Chapter 2, *System Model.*

The purchasing system is an extension of the payment system (see 6.2.3) with additional data to enable buyer restrictions defined by purchasing managers to be enforced by the purchasing system:

- the identity of the buyer as well as the account to be charged
- the kinds of goods or services being purchased
- the kind of terms and conditions (i.e., **contract**) involved.

The purchasing infrastructure is illustrated in Figure 6.1.

Figure 6.4: Purchasing Infrastructure



The buying organization is supplied with commercial purchasing cards by the issuing bank. The use of these cards is supervised within the buying organization by the purchasing manager.

1. **control parameters**: the purchasing manager establishes with the issuing bank control parameters such as spending limits and usage controls for each buyer
2. **wake up**: the buyer requests SET supplier and payment gateway certificates from the merchant and tells the merchant the buyer's payment certificate, payment card type and the promotional card name. This information allows the merchant to identify a purchasing card transaction and the type of business relationship[1]

3. **payment request**: the buyer verifies the merchant and gateway certificates by traversing the trust chain to the root key and forwards the purchasing request as a secure payment request
4. **authorisation request**: the payment instruction is received by the merchant, verified and sent on to the payment gateway
5. **authorisation**: The payment is forwarded via the bankcard association network to the issuing bank for cross checking with the control parameters and, if authorised, the authorisation returned to the merchant
6. **payment response**: the merchant returns the authorisation as a payment response to the buyer to indicate payment has been made
7. **deliver**: the merchant delivers the goods services purchased to the buyer. (If the good are not available, the merchant must cancel the payment)
8. **transaction capture**: the merchant submits transactions to the acquirer bank for payment, including VAT and line item detail.
9. **clearing**: the bankcard association network arranges for the buyers account to be debited and the merchant's account credited appropriately
10. **statement**: periodically the purchasing manager receives reports detailing transactions for approval.

It should be noted that in the purchasing infrastructure the Internet is used for all payment related messages, using the payment infrastructure. Clearing and statement processing is accomplished via the bankcard association's private network infrastructure.

If the merchant is operating through a third party marketplace operator the all but the delivery, payment and transaction capture steps will be executed by the marketplace operator. The market place operator will instruct the merchant to make the delivery. This requires **IT integration technology** (see 7.3 on page 49) between the marketplace operator and the merchant.

The IT Integration may be real-time, directly connected to the merchant's order processing system, in which case the marketplace operator can hold the payment until the merchant confirms his ability to deliver. Alternatively if batched, the market place will have to credit transactions for goods or services which cannot be supplied.

6.2.3 Payment infrastructure

Means for electronic payment is fundamental to electronic commerce. Given the objectives and time-scales of the E2S project it was important to select a payment system that:

- is convenient for users
- spans national boundaries
- has an accepted status in the financial community
- is available for use immediately.

1. this wake-up step occurs at the end of the browse phase of the secure transaction protocol.

This led to the choice of electronic bankcards, in particular the *Secure Electronic Transactions* (SET) standard [SET] and its implementation by a number of vendors (e.g. Verifone [VERIFONE]).

SET is an open, vendor neutral, non-proprietary, license-free specification for securing on-line transactions.

The payment infrastructure for the E2S implementation comprises:

- the existing "**VISANet**" network for clearing Visa transactions
- a **card-holder SET module**
- a **merchant SET module**.

This components are embedded in Figure 6.1 above. The card-holder SET module interacts with the SET merchant module across the Internet to initiate and complete a payment transaction.

The merchant SET module transfers notification of completed SET payments from the merchant to a **payment gateway** at the merchant's acquiring bank.

Within the current SET protocol, details of the order are not cryptographically protected. The protocol supports full transmission of line-item detail, but only a digest of the order data is signed.

SET requires that a buyer keep private keys secret. In E2S this is achieved using a **smartcard**.

Since the merchant SET module stores the merchant's keys it should be a protected system component. In the E2S implementation this is achieved by assigning the key handler a separate compartment on a CMW workstation and authenticating the source requests to use the keys for cryptographic operations (see [E2.3 GA]).

The lessons learned from the work with SET during E2S are described in [E2.9].

6.3 Secure networking

Secure networking is required to ensure that electronic commerce and the support infrastructure is safe from network threats such as snooping, replay and other malicious or erroneous events.

Secure networking by itself does not guarantee security. The requirements for trusting people performing critical roles in the architecture and for physical protection of critical components must also be respected.

Secure networking comprises three infrastructures:

- system partitioning
- signed mobile code
- strong cryptography.

6.3.1 System partitioning

On the assumption that the **software** in the computers does not contain **security flaws**, **viruses** or **trojan horses**, and that the people with physical access will not introduce such threats, it is assumed cryptographic protocols can be trusted.

Thus security of an E2S implementation ultimately depends upon

1. access to system components is only being permitted after successful access control checks (which in turn rely on user and/or privilege authentication)
2. failures in the implementation or management of one component not damaging the integrity of another.

Therefore an E2S system should be partitioned into **secure domains** with physical separation between them and filtering of transactions between domains.

6.3.1.1 Firewalls

Classically security domains in networks are created using **firewalls** (see, for example, [CB 94]). A firewall consists of three sets of components:

- **filters** to block and/or audit transmission of certain kinds of message (specified by type, destination or some combination of both)
- **gateways** which forward acceptable messages from one side of the firewall to the other
- **application proxies** which perform application specific access controls, monitoring and auditing.

For complex systems, or those supporting sensitive data, the classical firewall approach can be expensive in terms of both physical infrastructure and management effort.

6.3.1.2 Compartmentalised Mode Workstations

Within E2S, trusted operating systems are used to achieve high assurance service platforms within single machines. **Compartmentalised mode workstations** (CMW) are used to achieve system partitioning with in single computer [DALTON]. Examples of the use of partitioning in support of network security permitted by CMW include:

- firewall proxy hosting [ZHONG]
- Trusted SOCKS [E2.3-GC].

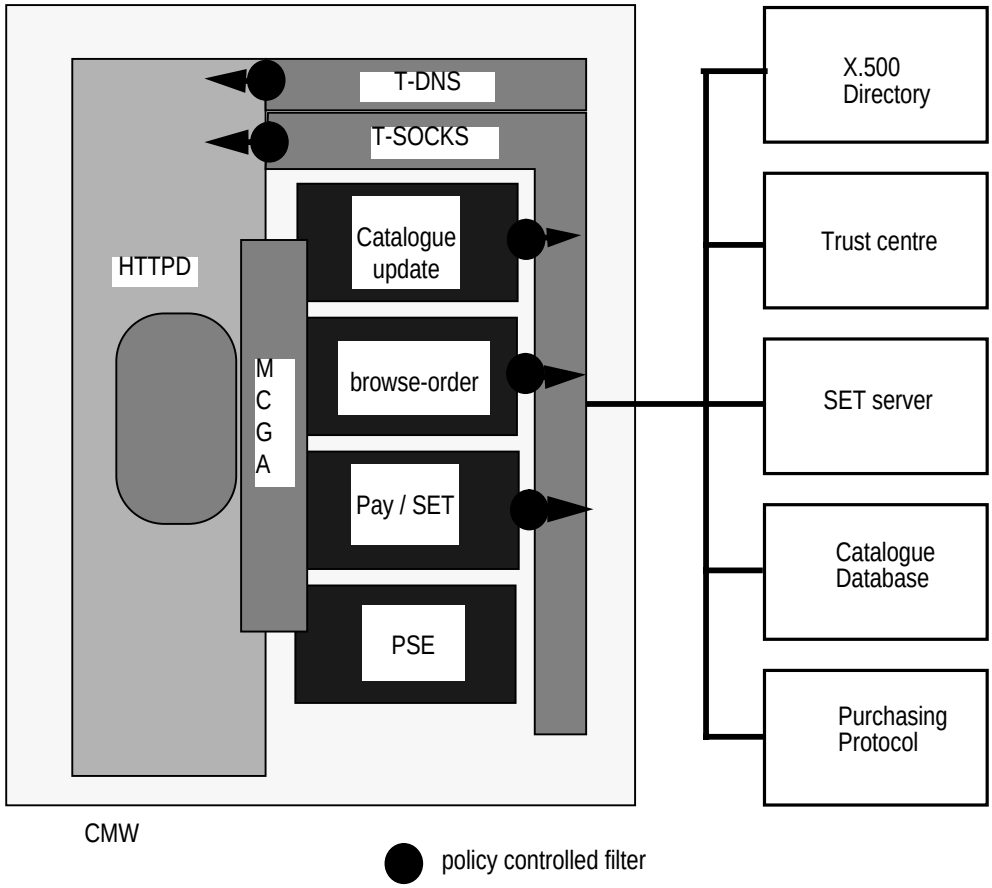
Figure 6.1, taken from [E2.3-GA] shows a CMW set up to host the server side implementation of secure commerce.

The system is divided into several compartments

1. an “outside” compartment connected to the Internet
2. an “inside” compartment connected to the merchant’s internal IT systems
3. a set of compartments, one for each element of the secure commerce system
   (i) catalogue updating function offered to content managers
   (ii) purchasing function
   (iii) payment function
   (iv) the merchant’s personal security environment (keys)

- The outside compartment contains a web server (httpd) to receive HTTP requests from Internet users. This server is bound to a trusted multi-compartment gateway agent which routes received requests to the appropriate function[1].

Figure 6.5: CMW for secure electronic commerce



- The “function” compartments are at the same level so that for example the purchasing function can invoke the payment function
- The inside compartment is permitted to communicate with internal services
- where required, each function compartment is provided with access to a policy controlled filter. This is an object which is privileged to copy data between compartments at different levels provided the data passes checks imposed by the filter’s policy. For example the payment functions filter will restrict SET requests to be directed to the SET server
- the inside compartment also provides policy controlled filters for SOCKS (a TCP proxy) and DNS (the Internet name service) so that internal services can connect to external services, for example to enable the SET server to connect across the Internet to a **payment gateway**.

This architecture provides a high level of assurance that

1. both the functions and the internal services are isolated from the Internet
2. failures in any of the functions are contained
3. all interactions between compartments are subject to policy checks.

1. In the early E2S trials the HP Virtual Vault web server [HP] was used. This has a simple inside-outside configuration. For the final pilots the multi-compartmentalised structure described in [E2.3-GA] was developed.

6.3.2 Signed mobile code

The E2Send-to-end secure channel model requires that a client execute an application-specific protocol and the code for this protocol has to be delivered to the user. Whilst this code does not contain keys or private data (taking them instead from a smartcard) there is the potential for an attacker to substitute a hostile implementation of the security software.

One possibility is physical distribution on CD-ROM or floppy disc along with the smartcard. Alternatively the software can be downloaded across the Internet from a server. In either case there is the risk of the software being substituted, which can be prevented by digital signature. A digital signature is only a “watermark” conveying information about who signed the code[1]. It does not of itself imply what level of trust can be put in the code.

The distributed code can be either a “plug-in” intended for a specific client environment (e.g., a particular browser) or it can be written in a platform independent “mobile code” (e.g., Java).

A plug-in has the advantage of close integration with the browser (including transparent automatic installation), but the disadvantage of being browser and often platform specific.

Mobile code has the advantage of running on most platforms making the marketplace more open. Additionally, being a self-contained environment, a wider set of facilities for user interface design and integration with both client-side and server-side IT are available. This can be particularly important where the transactions to be supported are computer-driven rather than user-driven on the client side (e.g., secure messaging as in the SBCW pilot).

The ability of mobile code to access local resources on a host machine, including its network connection raise serious concerns about security, since the potential exists for the mobile code, whether by accident or malicious design to steal, destroy or damage client data, launch bogus transactions or perform denial of service attacks. Therefore secure mobile code environments must provide a means to link the signatures used to “watermark” mobile code to access control policies.

These issues are discussed in detail in [E2.1] *Mobile Code Study Report.* and are an area where technology is developing rapidly.

6.3.3 Strong cryptography

The security of secure protocols depends upon the strength of the cryptographic algorithms they use and on the length of keys.

Many government impose constraints on the key lengths that can be used for cryptography. Therefore E2S transaction protocols separate out authentication, integrity and non-repudiation functions since these can be built using digital signature techniques (and indeed signatures can be computed without using cryptographic functions - i.e., using one-way hash functions instead). This gives the potential to negotiate the use of long keys to give strong authentication, integrity and non-repudiation with the option of shorter keys for confidentiality.

---

1. A digital signature is a stronger statement than downloading the mobile code over an authenticated secure http session since the later does not guarantee that the module has not been tampered with on the web server - the use of signatures is more in the spirit of end-to-end security.

Since deployment of cryptography is constrained by export regulations, import regulations and government policy the E2S architecture can only make recommendations:

- the greatest strength cryptography permitted should be used
- security protocol implementations should be parameterised by algorithm and key length so that alternatives can be substituted
- specific algorithms and key lengths (except in the form of constraints on minimum size) should not be built into applications
- location information should be associated with system components so that politically correct choices of algorithms and key length can be made.

7 Server technology

Server technology provides the means to deliver secure services to users.

It consists of:

- secure email gateway
- secure web server
- IT integration
- security audit.

7.1 Secure email gateway

Secure email gateways are required to support secure electronic mail-based telecooperation.

A secure email gateway acts as a gateway between a secure Intranet (e.g., a LAN) and the open Internet. It guarantees that any mail exchanged with users outside the Intranet is protected from attack (theft, invasion of privacy, modification or forgery) by third parties. The users on the LAN must be trusted to use unsecured user authentication on their workstations correctly. If this is not appropriate they should be supplied with security enhanced mailers enabled with smartcards.

Directing mail to a distribution list require mail to be decrypted from a public key associated with the group and redistributed as messages encrypted using the public keys of the recipients. To do so at least secure email gateway must be integrated with the **message handler** of a **trust centre** (see 6.1.3 on page 35) which has an information model for the membership of the distribution list. A secure email gateway in this configuration is called a **secure mail exploder** [HEFERT].

Secure email gateways and **secure mailers** at different locations can cooperate to define a **secure email infrastructure** for the users they protect.

A secure mail gateway provides:

- message origin authentication
- message content integrity
- message content confidentiality
- message non-repudiation.

A secure mail exploder provides in addition:

- addressing distribution lists
- addressing recipients by role.

A secure mail gateway provides automatically:

- signing and signature verification for both clear text and confidential messages (driven by the sender's **name**)

- encipherment and decipherment of confidential messages (driven by the recipient's **name**).

A secure email gateway is a trusted system component. A high level of assurance can be achieved by partitioning the trust centre component across separate components using a CMW workstation and subjecting the trust centre to security audit.

In the E2S implementation the SecuDE toolkit is used together with a smartcard based PSE to implement the cryptographic elements of the secure email gateway.

In addition to delivering mail to user mailers, the gateway may also deliver mail to **IT integration** technology which implements a **telecooperation** application (for example, retrieving documents from a database, or driving a **secure transaction protocol** for an office procedure such as ordering supplies).

7.2 Secure web server

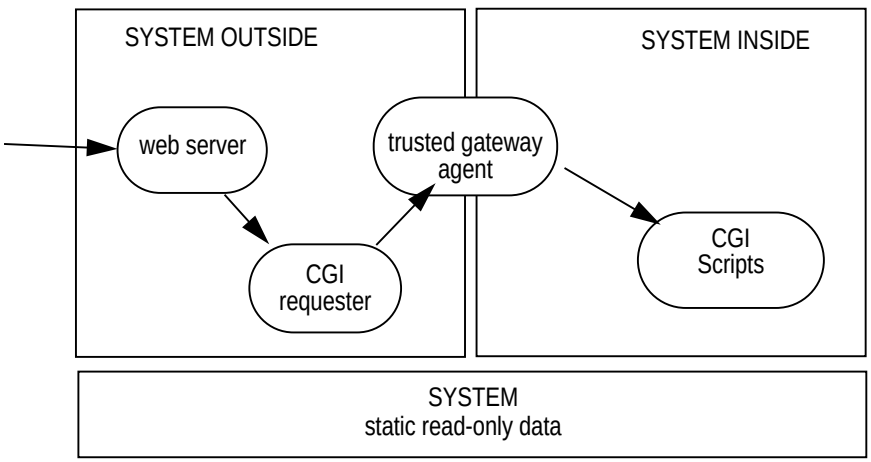
A web server is required to support interactive sessions as described in Section 5.2 on page 27.

A web server can be extended with **scripts** which are the analogues of plug-ins or applets in the browser. Where plug-ins or applets are used to implement secure transaction infrastructures there will be a corresponding script in the web server.

A typical web server contains too many complex functions and is too extensible to be a trusted system component. Although the web server itself does not need many privileges to run, it can be used as a stepping stone to attack the internal applications that interact with it.

This is overcome in E2S by splitting the implementation of a web server across compartments supported by a compartmentalised model workstation as shown in Figure 7.2 taken from [ZHONG].

Figure 7.1: Secure Web Server



There are two compartments, "inside" and "outside", and one classification "system". The read-only web server content is labelled with sensitivity

"system" but not compartmentalised. Therefore it can be read but not modified by processes running in either compartment. The trusted gateway agent provides a restricted secure path between the outside web server and the inside applications.

The web server is split at the CGI interface. (This is the point at which a web server spawns a child process to run a script indicated in a URL and passes all the arguments to it.

The trusted gateway agent ensures that the CGI requester only calls programs CGI scripts that have been registered with it. The agent has the privilege to change to the system inside compartment and run the script.

If the outside web server is compromised it cannot modify the static data nor go into the internal system since it has no privileges.

If data passed in from the outside web server triggers a bug in CGI scripts damage is confined to the "inside" compartment. Since this compartment cannot open connections to the outside network the attacker cannot directly control the broken process or make use of it.

Figure 6.1 on page 30 shows how a secure web server can be integrated with other E2S components to provide a CMW-based corporate purchasing gateway.

7.3 IT Integration technology

IT integration technology enables back office applications to be exported via web servers to users. Examples of the need for IT integration include:

- systems for which the secure commerce feature of E2S is too limited
  - — a more powerful GUI is required than that offered by HTML
  - — the transaction dialogue is too complex to represent as an exchange of forms
  - — integration with client side IT is required
  - — multi-party transactions are required.
- systems in which the computer-to-computer communication rather than user-to-computer communication is required
  - — transactions are automated, for example, purchasing driven by an inventory control system
  - — a buying organization has an in-house purchasing system which acts as a concentrator for external purchases
- between a marketplace operator and a merchant.

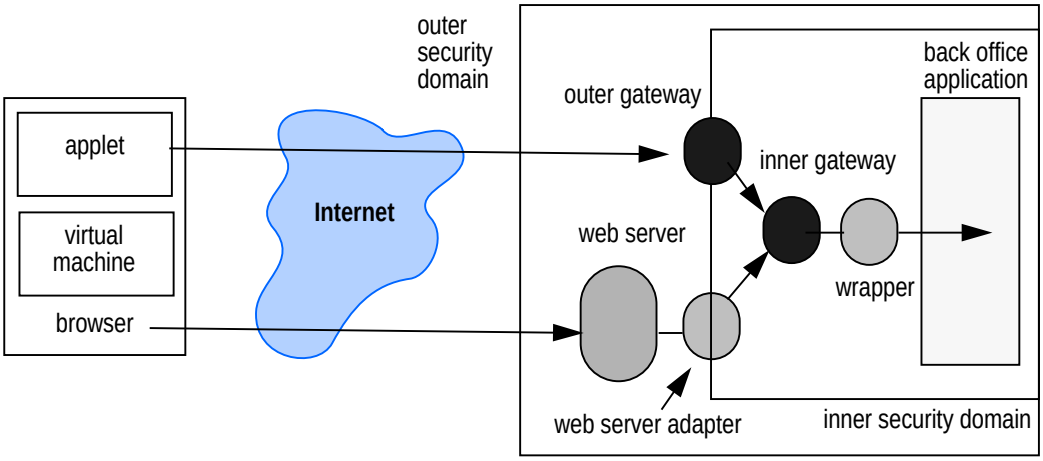
IT integration must satisfy two key requirements:

- *controlled access to data and applications in back office systems*
  - — confidential data must not be allowed to leak into the Internet
  - — mission-critical application must be protected from attack via the Internet
- *custom, branded session delivery*
  - — control of presentation to the user
  - — control over division of processing between browser, IT integration component and back office application

— user confidence in trustworthiness of system by virtue of trust in the "**brand image**".

The IT integration technology shown in Figure 7.2 supports these both together and separately.

Figure 7.2: IT Integration Technology



The back office application and associated data is held with an inner security domain protected either by a **firewall** or by use of a **trusted network**. A gateway at the boundary of the domain exports controlled access to the application and associated data to an outer security domain.

In the case where HTML and forms are being used for user interaction, a **web server adapter**[1] is used to translate web browsing and form filling actions to requests on the inner gateway. The adapter must enforce access control (e.g., by requiring protected user authentication and a **secure http session** between the browser and the server).

In the case where interaction is controlled by a **mobile code module** added to the browser, the web server adaptor can support an end-to-end **secure application session protocol** which tunnels its messages through the web server using HTTP. However this makes the web server a potential bottleneck, so in the case where interaction is controlled by an applet downloaded from the web server into the client browser, the applet can connects via a dedicated outer gateway to the inner gateway.

The inner gateway ensures that the back office application can only be accessed by either the outer gateway or the web server adapter and performs translation from the Internet session protocol to the commands of the back office application.

The outer gateway, web server adapter and inner gateway necessarily contain application specific and access control policy specific functionality. To maximise re-use and consistency across applications, they should be constructed using CORBA distributed object technology.[2] In addition to providing support for distribution of these components, standard CORBA services provide **wrapper** technology for a wide range of application interfaces (OLTP, remote SQL database, etc.).

1. The adapter is linked to the Web Server's "Common Gateway Interface (CGI)" or equivalent (e.g., the Netscape "NSAPI" interface).

The applet to outer gateway path provides a direct means for business-to-business interactions, whereas the Web server route is best suited to supporting user-to-business interaction.

Because of their role in access control both the web server adapter and the gateways require management interfaces for use by security administrators and require access to directories and certificate repositories. Consequently they must be subject to **security audit**.

7.4 Security audit

Security failures are more often attributed to errors in the management and deployment of security technology than in a failure of the technology itself. Additionally in any large organisation there is the risk of an attack by an “insider”. To make a system resilient against such threats security must strengthened by providing logging of security related events (dynamic auditing) and regular checking that physical security and access control policies are correctly implemented (static auditing).

Security audit tools include:

- technology for keeping secure logs of security related events
  - — such logs are critical components and need strong integrity protection
  - — E2S technology can be used to provide the security component of such protection when it is otherwise not available.
- tools for analysis of audit trails kept by critical infrastructure components (e.g., firewalls, key management infrastructure functions)
- intrusion testing tools for ensuring access controls are in place and known security flaws are fixed (e.g., by system probing, review of system configuration files etc.).

2. Vendors of CORBA technology are currently supplying Java ORBs to enable Java applets to invoke CORBA services. There is in addition an increasing convergence of CORBA services and Java enterprise beans (e.g. JDBC for database access). In the E2S implementation both CORBA and enterprise beans interfaces have been demonstrated.

8 Examples

This chapter shows, at a high level, how the components of the common technology framework are deployed in the four E2S project pilot demonstrators.

The detailed descriptions of the pilots described in the documents produced from the corresponding tasks are definitive; the examples here are given simply to illustrate the applicability of the architecture and put it into the overall context of the project.

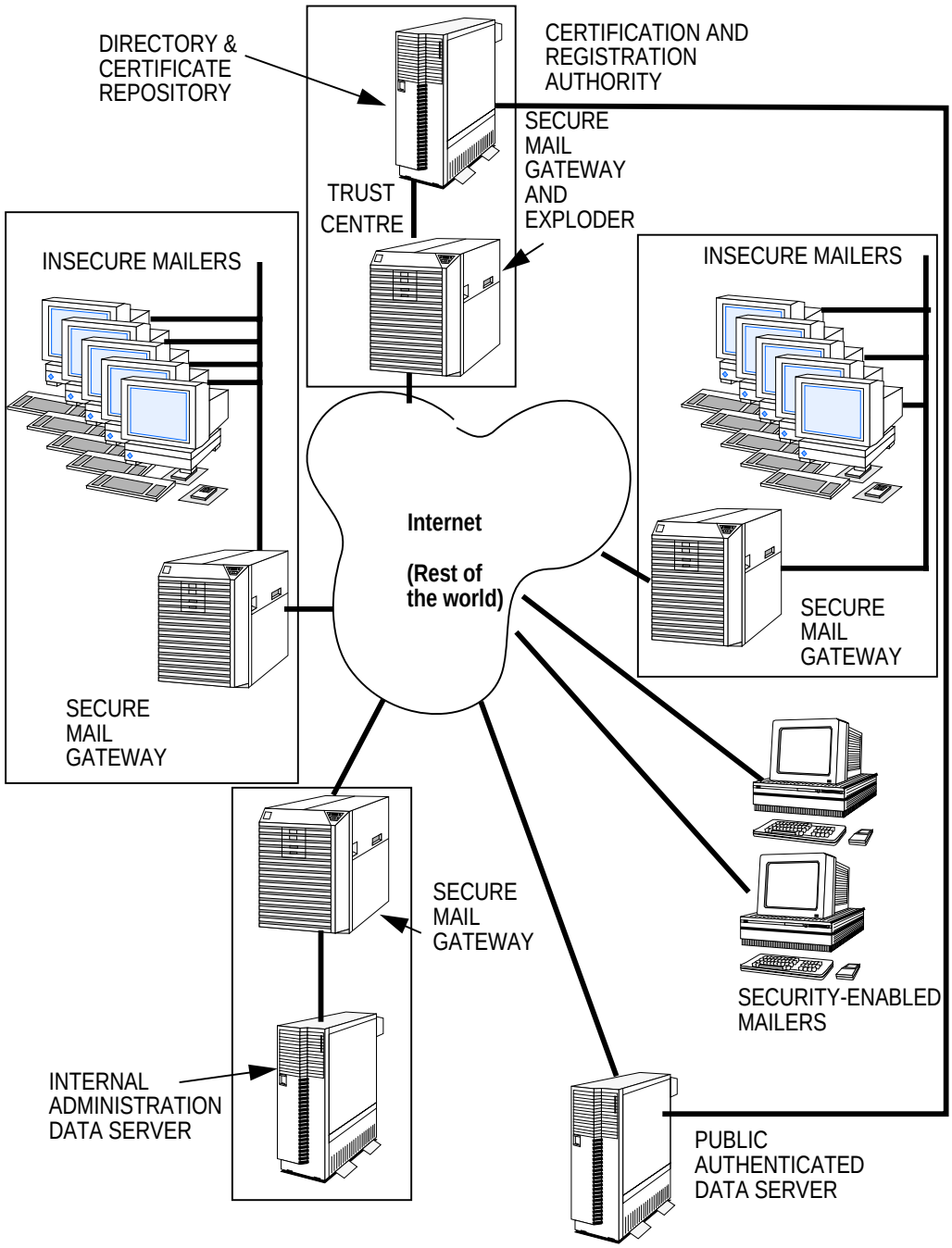
In this version of the architecture document, the descriptions reflect the initial designs of the pilots resulting from user requirements analysis. As development of the pilots proceeds the descriptions are likely to change to some extent as a result of user feedback and changes in technology. These changes will be reflected in updates to this document.

The pilots show four different styles of use of the E2S architecture, representative of different electronic commerce requirements:

- secure telecooperation
  - — a “one-to-one” scenario
- customer access to an on-line service with public certification
  - — a “many-to-one” scenario with exploiting public security infrastructure for wide reach
- customer access to an on-line service with private certification
  - — a “many-to-one” scenario using a private security infrastructure to protect brand integrity
- customer access via a secure market-place to a set of on-line services
  - — a “many-to-many” scenario via a shared, structured catalogue.

8.1 Secure telecooperation (TUB)

Figure 8.1: Secure telecooperation



The secure telecooperation pilot shows the use of E2S technology to enable an organisation (viz., an administration) to create an “Intranet” out of the Internet.

Figure 8.1 shows the pilot in a simplified form:

- two sets of insecure mailers each representing different administration departments within an organisation, connected to the Internet via secure mail gateways

- a number of security enabled mailers belong to administrators connected to the Internet directly.

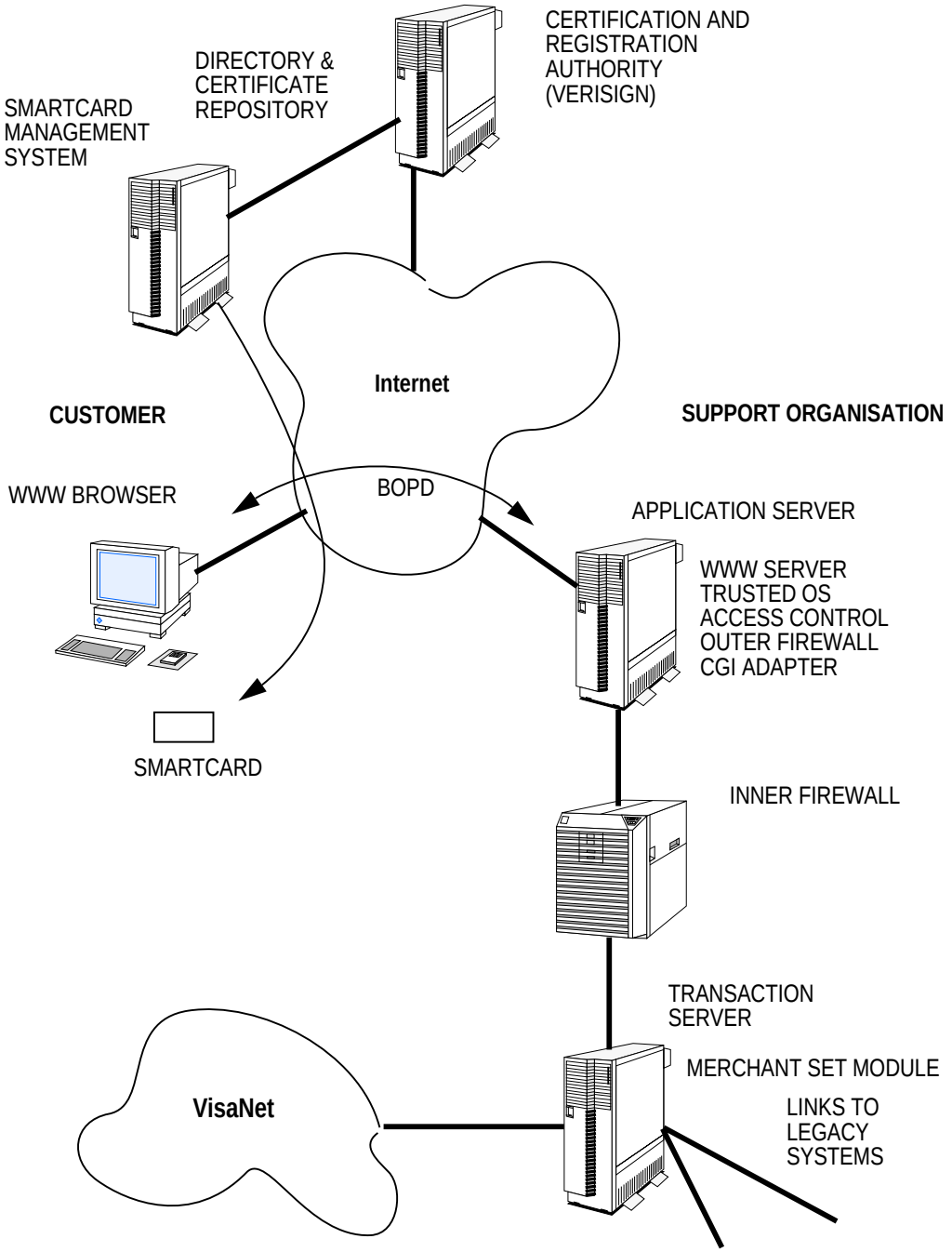
Authentication of users to mailers is part of local logon to the computer supporting the mailer (e.g., a user name, password verification). In the case of an insecure mailer, the logon merely establishes the user's right to use a name. In the case of a secure mailer the logon also enables the mailer to access to user's private key for signing and encrypting messages.

The administration provides a certification and registration authority for all of the users in the administration, together with management of secured distribution lists (through a secure mail exploder).

In addition to supporting mail, administrative data is made available through an electronic mail interface. The administrative data server is protected from the Internet by a secure mail gateway.

8.2 On-line software licensing (WCSO)

Figure 8.2: Online Software licensing

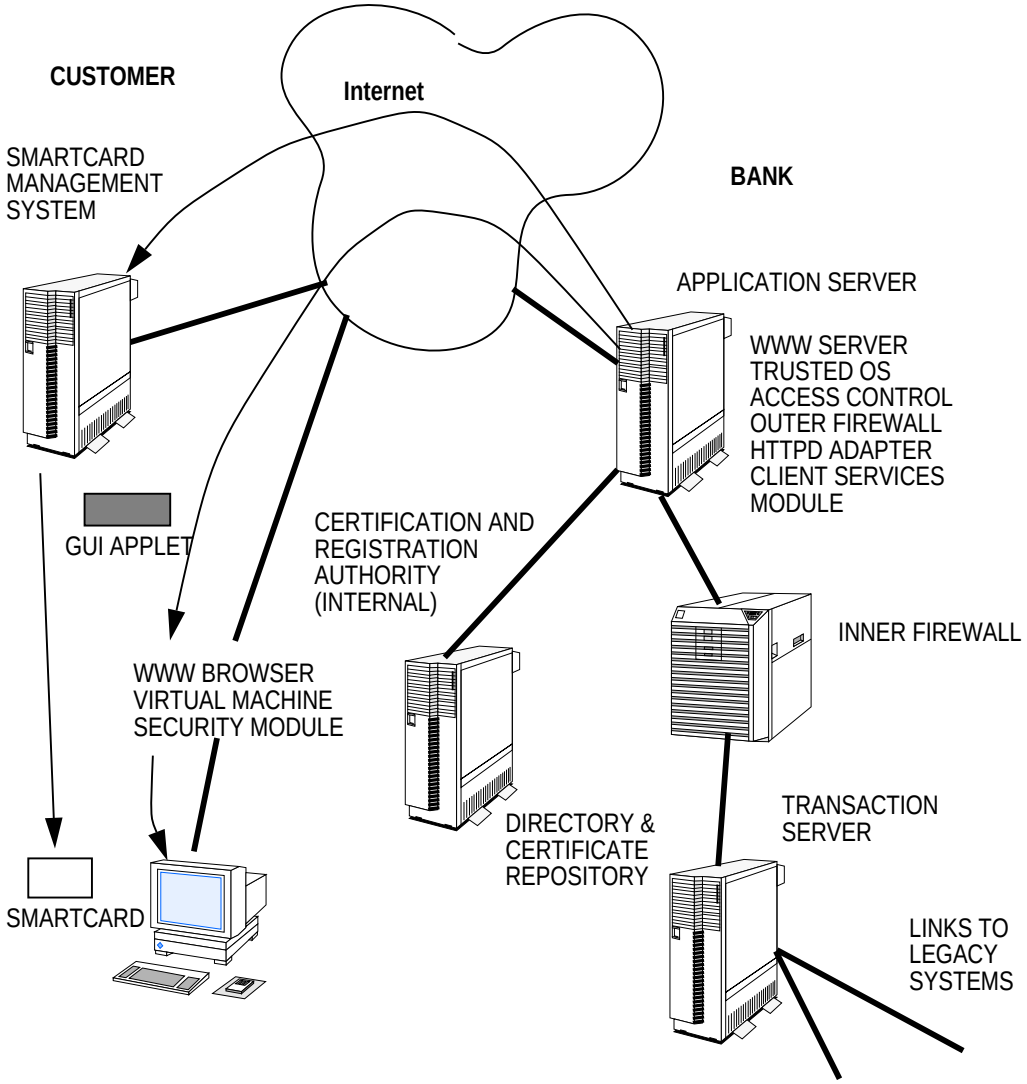


The on-line software licensing pilot enables customers to download software from the world-wide web for evaluation and then buy licenses for continued use of that software if it meets their needs. The users are registered and issued with private and public keys via Verisign Inc., an external registration and certification authority. The keys are delivered to the user via a smartcard. The software vendor uses a CMW to co-locate a web server, an access control function, an outer firewall and an httpd adapter as the customer-facing part of the system. The CGI adaptor communicates, via an internal firewall, with a

transaction server that in turn connects to the existing IT structure for recording license details within the support organisation.

8.3 On-line services for investment banking (SBCW)

Figure 8.3: On-line Services for Investment Banking

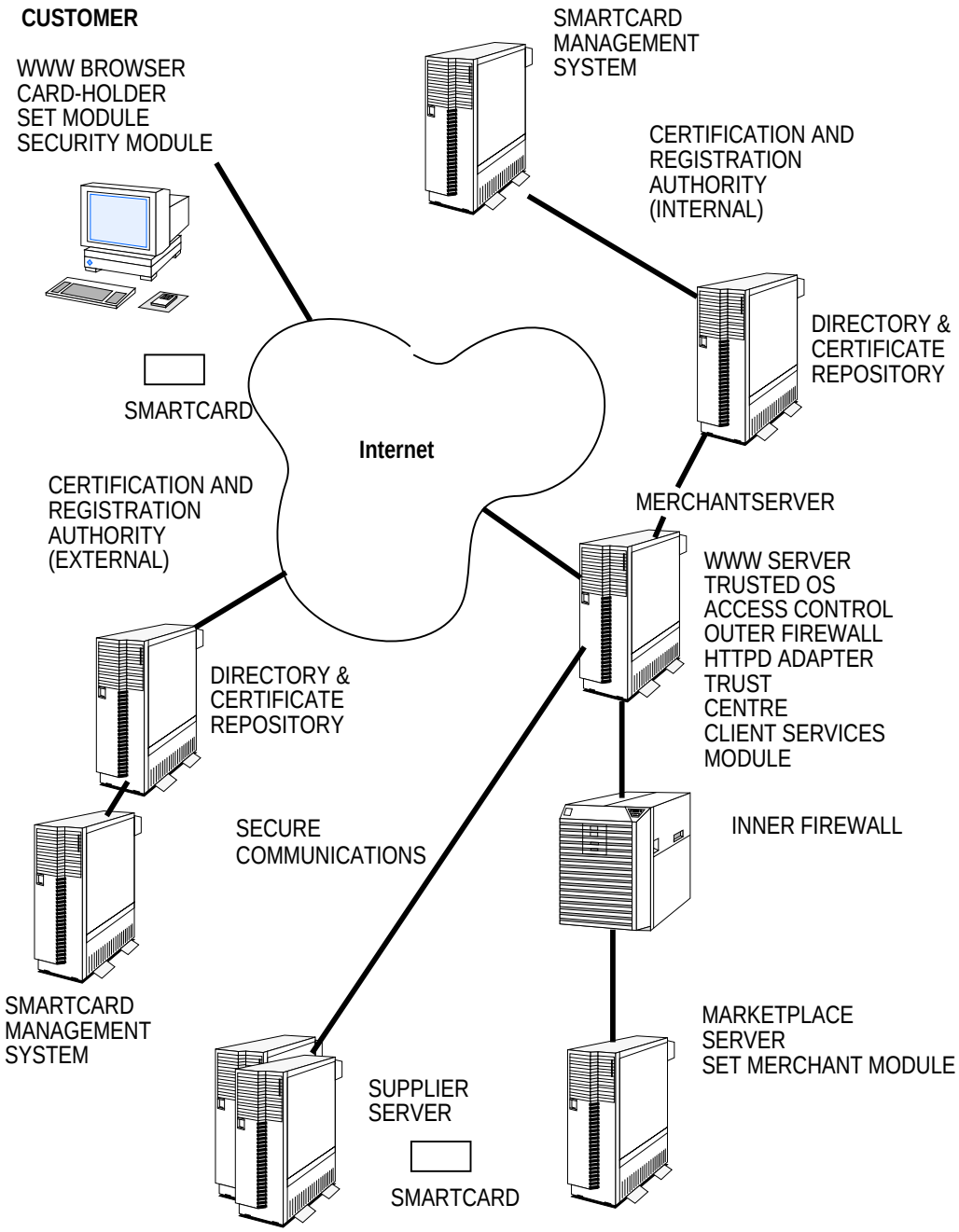


This system uses similar technology to the customer support example but differs in two important respects:

- the key management and smartcard functions are managed by the bank (since “security” is part of its brand image)
  - — however smart issuing and user registration is performed by the customer using a downloaded smartcard management system supplied by the bank
- a custom user interface is down-loaded to the customer because the forms handling capabilities of HTTP are too restrictive for the needs of the application which is highly interactive.

8.4 Third party merchant service (Onyx)

Figure 8.4: Third party merchant service



This example also shares much in common with the customer support and investment banking example.

The merchant server acts as a “market-place” for a number of service suppliers who have the ability, subject to access controls, to up-load data from their own systems into the merchant server and to download management information about service utilisation and payments due etc.

Customers are authenticated via smartcards, these may be:

- cards issued by the third party merchant service provider

- cards issued by external certification and registration authorities.

Since the market-place provider has to manage relationships between certification authorities, customers, suppliers and the electronic payment infrastructure, a trust centre is used to tie together authentication and access control policies.

9 Viewpoint Analysis

9.1 Viewpoints

This chapter summarises the E2S architecture in terms of the viewpoints of the ISO Reference Model for Distributed Processing (ISO/IEC 10746-3, ITU Recs. X.903).

It consists of a structured list of the architectural concepts and rules defined in chapters 2 to 6 inclusive.

This analysis:

- enables alignment of the E2S architecture to other distributed processing architectures
- shows the separation of concerns in the E2S architecture between
  - roles, objectives and policies (the enterprise viewpoint)
  - information resources and processes (the information viewpoint)
  - functional elements and the interfaces between them (the computational viewpoint)
  - distribution infrastructure (the engineering viewpoint)
  - technology choices (the technology viewpoints).

Given the objectives for the E2S architecture set out on Chapter 1, the design of the architecture has deliberately set out to minimise constraints in the enterprise, engineering and technology viewpoints so as to permit the widest range of applications and implementation freedom, consistent with retaining a common technology framework.

9.2 Enterprise viewpoint

- person
  - group (of people)
- business
  - brand image
- application
- telecooperation
- interactive sessions
- purchasing
- payment

9.2.1 Secure electronic mail

- security enhanced mailer

- insecure mailer
- secure email gateway
- secure email exploder

9.2.2 Web browser

- browser
- WWW server
- page-oriented interface
- custom user interface

9.2.3 Key management infrastructure

- key generation
- key escrow
- key recovery
- key certification
- user registration
- key verification
- key revocation
- certification authority
  - internal
  - external
- registration authority
  - internal
  - external
- key management infrastructure security manager

9.2.4 Trust centre

- access control
  - role
  - capability

9.2.5 Smartcard infrastructure

- smartcard
- smartcard issuer
- smartcard user

9.2.6 Secure transactions infrastructure

- contract

9.2.7 Payment infrastructure

- merchant

- card-holder
- bankcard association
- issuing bank
- acquiring bank

9.2.8 Purchasing infrastructure

- merchant
- buyer
- purchasing manager
- content manager
- relationship manager
- banking infrastructure

9.2.9 Firewalls

- computer
- Internet (as a “community”)
- security domain
- software
- security flaw
- virus
- trojan horse
- security policy

9.2.10 Security audit

- secure logs
- secure system configuration files
- intrusion detection

9.3 Information viewpoint

9.3.1 Person

- name
- identity
- post
- role

9.3.2 Secure electronic mail

- email distribution list

9.3.3 World Wide Web (WWW)

- HTML

- S/MIME
- page
- form
- applet

9.3.4 Key management infrastructure

- public key, private key pair
- certified public key
- X.509 digital certificate

9.3.5 Smartcard infrastructure

- PIN
- smartcard branding

9.3.6 Secure transaction protocols

- PEM format message
- digital signature

9.3.7 Purchasing infrastructure

- order
- account

9.4 Computational viewpoint

9.4.1 Secure electronic mail

- insecure mailer
- security enhanced mailer

9.4.2 World Wide Web

- Web browser
  - — virtual machine
- mobile code module
  - — applet
  - — plug-in
- Web server
  - — script

9.4.3 Key management infrastructure

- certification authority
- registration authority
- client services module
- certificate repository

- directory
- security module
- personal security environment (PSE)

9.4.4 Smartcard infrastructure

- smartcard
- smartcard reader/writer
- smartcard software library

9.4.5 Trust centre

- message handler
- request hander
- key handler
- attribute repository
- key repository

9.4.6 Purchasing infrastructure

- purchasing card

9.4.7 Payment infrastructure

- user SET module
- merchant SET module
- payment gateway
- “VISAnet”

9.4.8 Secure transaction infrastructure

- secure mail session
- secure http session
- secure application session
- secure commercial purchasing session
- commercial wallet

9.4.9 System partitioning

- filter
- gateway
- application proxy

9.4.10 IT integration

- “back office” application
- web server adapter
- outer gateway
- inner gateway

- wrapper

9.5 Engineering viewpoint

- Internet (as a network)
- physically secure location

9.5.1 Smartcard architecture

- physical encapsulation of keys and algorithms

9.5.2 System Partitioning

- firewall
- compartmentalised mode workstation

9.6 Technology viewpoint

- Internet (as a set of standards defined by IETF etc)
-
- Privacy Enhanced Mail (PEM)
- Pretty Good Privacy (PGP)
- Netscape 2.0/3.0
- Secure Socket Layer protocol (SSL)
- Verisign
- Ice-tel
- X.500 Directory
- X.509 digital certificate
- SecuDE tool-kit
- Osisec tool-kit
- GEMPlus GPK-2000
- RSA
- DES
- SHA-0, SHA-1
- Secure Electronic Transactions (SET).

References

[ACTRA]

see http://www.actracorp.com/

[B1]

Market Report , E2S Project Deliverable B1, Onyx Ltd., Gatehead UK, 1996.

[BAKO]

Kolletzki, S., "Secure Internet Banking with Privacy Enhanced Mail. A Protocol for Reliable Exchange of Secured Order Forms (BAKO)", *Proc. 7th Joint European Networking Conference*, Budapest, May 1996.

[C1]

Consolidated User Requirements, E2S Project Deliverable C1, Onyx Ltd., Gateshead, UK, 1996.

[CB 94]

Cheswick, W.R., and Bellovin, S.M., **Firewalls and Internet Security: Repelling the Wily Hacker**, Addison Wesley, Reading MA, USA, 1994.

[D3]

Security Models and Policies, E2S Project Deliverable D3, Gemplus, Gemenos, France, 1996.

[DALTON]

Dalton, C.J, and Griffin, F.J., "Applying Military Grade Security to the Internet, *Proc. 8th Joint European Networking Conference*, Edinburgh, 1997.

[DES]

ANSI X3.92, "American National Standard for Data Encryption Algorithm (DEA)," ANSI, 1981.

[DIA]

Defense Intelligence Agency, "Compartmented Mode Workstation Evaluation Criteria", Report DDS-2600-6243-91, 1991.

[E2.1]

Mobile Code Study Report, part of E2S Project Deliverable E2.1, APM Ltd, Cambridge, UK, 1996.

[E2.1/E2.6]

Star System Secure IT Integration, part of E2S Project Deliverable E2.1/E2.6, APM Ltd., Cambridge, UK, 1997.

[E2.2]

Key Management and Smartcard Infrastructure., E2S Project Deliverable E2.2, HP ENDS, Grenoble, France, 1997.

E2.3 GA]

Payment Gateway Architecture, E2S Project Deliverable E2.3-GA,HP Laboratories, Bristol, UK, 1997.

[E2.3-GC]

Gateway Components, E2S Project Deliverable E2.3-GC, HP Laboratories, Bristol, UK, 1997.

[E2.4]

Secure Telecommunication Application, E2S Project Deliverable E2.4, Technical University of Berlin, Berlin, Germany, 1997.

[E2.5 MF]

Marketplace Framework, E2S Project Deliverable E2.5-MF, HP Laboratories, Bristol, UK, 1997

[E2.5 BP]

Business Protocol Definition, E2S Project Deliverable E2.5-BP, HP Laboratories, Bristol, UK, 1997

[E2.8]

Secure User Session Infrastructure, E2S Project Deliverable E2.8, HP Laboratories, Bristol, UK, 1997.

[E2.9]

SET Payment Infrastructure, E2S Project Deliverable E2.9, HP Laboratories, Bristol, UK, 1997.

[E2.10]

Plan for Live Business Payment Trial, E2S Project Deliverable E2.10, Visa International, Paris, 1997.

[HEFERT]

Security Enhanced Mailing Lists, *Proc. 8th Joint European Networking Conference*, Edinburgh, 1997.

[HP]

Hewlett-Packard Co., “Virtual Vault Transaction Server Concepts Guide”, 1996.

[HTML]

see http://www.w3.org/pub/WWW/MarkUp/Activity

[ICE-TEL]

see http://www.darmstadt.gmd.de/ice-tel/

[ISO 10746-3]

ISO/IEC 10746-3:1996(E), ITU Rec. X.903, Information Technology - Open Distributed Processing - Reference Model: Architecture, ISO, Geneva, 1996.

[NETSCAPE]

see http://www.netscape.com/

[OPENMARKET]

see http://www.openmarket.com/

[OSISEC]

see http://www.cs.ucl.ac.uk/research/ice-tel/osisec/

[PEM]

IETF Privacy Enhanced Mail Documents - Part I: Message Encryption and Authentication Procedures (RFC 1421); Part II: Certificate-Based Key Management (RFC 1422); Part III: Algorithms, Modes and Identifiers (RFC 1423); Part IV: Key Certification and Related Services (RFC 1424).

Available as http://ds.internic.net/rfc/rfc1421.txt etc.

[PGP]

The Official PGP User's Guide, MIT Press, Boston, USA, 1995.

[RSA]

R.L. Rivest, A Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, **21**, 2, Feb. 1978, pp 120-126.

[SECUDE]

see http://saturn.darmstadt.gmd.de/secude/secude.html

[SEMPER]

see http://www.semper.org/

[SET]

see http://www.visa.com/

[SHA]

National Institute of Standards and Technology, NIST FIPS PUB 186, "Digital Signature Standard", US Dept. of Commerce, May 1994.

[SSL]

see http://home.netscape.com/eng/ssl3/index.html

[SUPPLYWORKS]

see http://www.supplyworks.com/

[VERISIGN]

see http://www.verisign.com/

[VERIFONE]

see httP;//www.verifone.com

[X.500]

ITU Recs X.501-510, The Directory. ITU, Geneva, 1995.

[X.509]

ITU Rec. X.509, The Directory - Authentication Framework, ITU, Geneva, 1989.

[ZHONG]

Zhong, Q., *Providing Secure Environments for Untrusted Network Applications*, IEEE Conference on Enterprise Security, 1997.