A method for designing secure solutions

by J. J. Whitmore

The task of developing information technology (IT) solutions that consistently and effectively apply security principles has many challenges, including: the complexity of integrating the specified security functions within the several underlying component architectures found in computing systems, the difficulty in developing a comprehensive set of baseline requirements for security, and a lack of widely accepted security design methods. With the formalization of security evaluation criteria into an international standard known as Common Criteria, one of the barriers to a common approach for developing extensible IT security architectures has been lowered; however, more work remains. This paper describes a systematic approach for defining, modeling, and documenting security functions within a structured design process in order to facilitate greater trust in the operation of resulting IT solutions.

Trust is the measure of confidence that can be placed on the predictable occurrence of an anticipated event, or an expected outcome of a process or activity.

For business activities that rely on information technology (IT), trust is dependent on both the nature of the agreement between the participants and the correct and reliable operation of the IT solution. The reliance on computerized processes for personal, business, governmental, and legal functions is evolving into a dependency and a presumption (not to be confused with trust) that the processes, and the IT systems within which they execute, will function without flaw. It is reasonable to expect that legal findings relative to the correct and reliable operation of IT solutions will be the basis for whether one party is liable for the damages suffered by another party as a result of a computerized operation.

Trustworthiness of IT solutions can be affected by many factors found throughout the life cycle of solution definition, design, deployment, and operation. The trustworthiness of design of IT solutions can be affected by the clarity and completeness with which the requirements are expressed by stakeholders and interpreted by solution designers. The trustworthiness of operation of IT solutions can be affected by the trustworthiness of the components and processes upon which they are built, the accuracy with which the design is implemented, and the way in which the resulting computing systems are operated and maintained. The trustworthiness of operational IT solutions can also be affected by the environments in which the solutions are positioned, by individuals who access them, and by events that occur during their operational lifetime.

Given that IT components will most likely continue to have flaws, that unexpected events will most likely occur, and that individuals will most likely continue to seek to interfere with the operation of computing solutions and the environmental infrastructure upon which the solutions rely, what can be done to instill a sufficient measure of confidence—that is, trust—in the correct and reliable operation of a given information technology solution?

One realistic expectation is that designers and integrators of IT solutions will enlist all reasonable measures to effect the correct and reliable operation of IT solutions throughout the design, development, and deployment phases of the solution life cycle.

While the responsibility for considering all reasonable measures is shared among all individuals involved in the design, development, and deployment of every IT solution, the role of anticipating the perils that the IT solution may face, and ensuring that the business risks of IT solution operation are mitigated, is generally the focus of IT security professionals.

Information technology security is a discipline that until recently was centered within the military, national security organizations, and the banking industry. With the growth of the Internet as a core networking and cooperative computing infrastructure, the need for, and the value of, IT security expertise has increased dramatically. The position of today's security architect closely parallels the role of the network manager or operator of the early 1980s. The similarities include the need to meet high expectations and service levels, a limited set of tools and techniques, low visibility of the electronic activities within the operational environment, plus the challenge of timely recognition and response to events and peril. In the mid-1980s the development of a systems management discipline provided a focus, a method, and a tool set for standardized approaches to system-wide design, operation, and management.

To date, the application of IT security countermeasures is generally limited to addressing specific vulnerabilities, such as applying network and systems management processes, hardening operating systems for publicly available servers, applying and monitoring intrusion detection systems, configuring and operating digital certificate servers, and installing and configuring firewalls.[1]

Based upon the evolution of destructive computer codes and viruses, the repeated breaches of sensitive computer systems, and recurring incidents of compromise of private information stored on networked computing systems, it is reasonable to conclude that the effectiveness of security measures in computing solutions needs to be improved. Recently security experts from government and industry expressed the need for a more comprehensive approach to describing security requirements and designing secure solutions.[2]

This paper documents the findings and recommendations of a project for which the initial objective was to develop training materials for a recently defined technical discipline, within IBM Global Services, for security architects. During the project, early attempts to organize and present the "prior art" dealing with information technology security produced incomplete and unsatisfactory results, leading to the conclusion that a more fundamental analysis was needed. The refocused analysis produced a thought-provoking proposal for articulating, documenting, and synthesizing security within information technology solutions.

Although the project objectives were met, the by-products are different from those first envisioned. The observations and conclusions from the project are summarized within this paper, including: an examination of the basic motivations for implementing security, a review and recategorization of commonly invoked security standards, an analysis of the fundamental elements of security architecture and its design, and some first attempts to render architectural representations.

Problem statement

A systematic approach for applying security throughout information technology solutions is necessary in order to ensure that all reasonable measures are considered by designers, and that the resulting computing systems will function and can be operated in a correct and reliable manner.

In IBM Global Services, the requirement for a method for designing secure solutions is driven from several perspectives: (1) there is a need to "grow" the community of IT architects with a shared security focus, (2) there is a need to create synergy among the several technical disciplines within the IT architect profession relative to security issues, and (3) there is a need to develop consistent designs because many businesses and organizations have similar security and privacy compliance requirements based upon statute, regulation, and industry affiliation, and many enterprises are multinational, with geographically diverse installations operating under similar security policies and practices.

To be effective, the resulting method should use existing security paradigms, integrate with other information technology architectures, and work with today's technologies.

A logical and systematic technique for designing secure solutions has potential value beyond IBM Global Services: to individuals, by fostering trust within computing environments that would otherwise be suspect; to information technology professionals, by promoting rigor within an emerging discipline of computing science; and to enterprises, by providing a technical standard with which the effectiveness of information technology designs, and designers, can be evaluated.

Analysis

Information technology architects rely on a wide range of techniques, tools, and reference materials in the solution design process. The results of a design activity may include an operational computing system or a set of documents that describe the system to be constructed from one or more viewpoints and at different levels of granularity. The documents provide a visualization of the system architecture.

To arrive at a system architecture, architects may use personal experience, or they may rely upon documented systematic procedures or methods. In addition to methods, architects refer to prior work and employ data collection techniques to define the problem space and the solution space. Reference materials can include a taxonomy of the problem space, a catalog of solution requirements, and documented models, patterns, or integrated solution frameworks. In general, as the definition of a given problem space matures, the taxonomy of the solution requirements stabilizes. This leads to well-defined reference models, proven solution frameworks, and mature solution design methods.[3]

IT security architecture fits this model for limited problem spaces such as securing a network perimeter, where a set of solution requirements can be defined. A solution framework can be constructed for an enterprise firewall, and a solution architecture can be documented using known reference models for "demilitarized zones." IT security does not, in general, fit this model, because: (1) the security problem space has not stabilized in that the number and type of threats continue to grow and change, (2) existing security solution frameworks take a limited view of the problem space, as with firewalls[4] and network-level security,[5] and (3) methods for creating security solution architectures are generally confined to the defined solution frameworks. For ill-defined problem spaces like IT security, the path to maturity of models and methods requires a different approach.[3]

Security-specific taxonomies, models, and methods. ISO (International Organization for Standardization) 7498-2[6] is a widely referenced document associated with IT security solution design. Its purpose is to extend the applicability of the seven-layer OSI (Open Systems Interconnection) system model to cover secure communication between systems. Section 5 of this document describes a set of security services and mechanisms that could be invoked at the appropriate layer within the OSI system model, in appropriate combinations to satisfy security policy requirements. Section 8 documents the need for ongoing management of OSI security services and mechanisms, to include management of cryptographic functions, network traffic padding, and event handling.

Many security practitioners use the OSI security services—authentication, access control, data confidentiality, data integrity, and nonrepudiation—as the complete taxonomy for the security requirements for IT solutions. However, the preamble of ISO 7498-2 specifically states that " . . . OSI security is not concerned with security measures needed in end systems, installations, and organizations, except where these have implications on the choice and position of security services visible in OSI. These latter aspects of security may be standardized but not within the scope of OSI Recommendations."

Security evaluation criteria. Agencies and standards bodies within governments of several nations have developed evaluation criteria for security within computing technology. In the United States the document has the designation "Trusted Computer System Security Evaluation Criteria," or TCSEC. The European Commission has published the Information Technology Security Evaluation Criteria, also known as ITSEC, and the Canadian government has published the Canadian Trusted Computer Product Evaluation Criteria, or CTCPEC. In 1996, these initiatives were officially combined into a document known as the Common Criteria, or CC.[7] In 1999 this document was approved as a standard[7–9] by the International Organization for Standardization. This initiative opens the way to worldwide mutual recognition of product evaluation results.

Common Criteria. Common Criteria provide a taxonomy for evaluating security functionality through a set of functional and assurance requirements. The Common Criteria include 11 functional classes of requirements: security audit, communication, crypto-

graphic support, user data protection, identification and authentication, management of security functions, privacy, protection of security functions, resource utilization, component access, and trusted path or channel. These 11 functional classes are further divided into 66 families, each containing a number of component criteria. There are approximately 130 component criteria currently documented, with the recognition that designers may add additional component criteria to a specific design. There is a formal process for adopting component criteria through the Common Criteria administrative body (www.commoncriteria.org).

Governments and industry groups are developing functional descriptions for security hardware and software using the Common Criteria. These documents, known as protection profiles,[10] describe groupings of security functions that are appropriate for a given security component or technology. The underlying motivations for developing protection profiles include incentives to vendors to deliver standard functionality within security products and reduction of risk in information technology procurement. In concert with the work to define protection profiles, manufacturers of security-related computer software and hardware components are creating documentation that explains the security functionality of their products in relation to accepted protection profiles. These documents are called "security targets." Manufacturers can submit their products and security targets to independently licensed testing facilities for evaluation in order to receive compliance certificates.

Common Criteria as a taxonomy for requirements and solutions. The security requirements defined within the Common Criteria have international support as "best practices." Common Criteria are intended as a standard for evaluation of security functionality in products. They have limitations in describing end-to-end security—because the functional requirements apply to individual products, their use in a complex IT solution is not intuitive.[11] Protection profiles aid in the description of solution frameworks, although each protection profile is limited in scope to the specification of functions to be found in a single hardware or software product.

Common Criteria as a reference model. The Common Criteria introduce few architectural constructs:[8] the target of evaluation, or TOE, represents the component under design; and the TOE security functions document, or TSF, represents that portion of the TOE responsible for security. Under Common Criteria, the system or component under consideration is a "black box"; it exhibits some security functionality and some protection mechanisms for the embedded security functions.

Summary of analysis. For well-understood problem spaces, methods document the prior work and provide best practices for future analysis. For changing problem spaces such as IT security, methods can only postulate a consistent frame of reference for practitioners in order to encourage the development of future best practices. With time and experience the methods and models associated with IT security will mature.

The Common Criteria document has important value to the security community, given its history and acceptance as a standard for security requirements definition, and its linkage to available security technologies through documented protection profiles and security targets. Common Criteria do not provide all of the guidance and reference materials needed for security design.

To develop an extensible method for designing secure solutions, additional work is required to develop:

1. A system model that is representative of the functional aspects of security within complex solutions
2. A systematic approach for creating security architectures based on the Common Criteria requirements taxonomy and the corresponding security system model

System model for security

Eberhardt Rechtin[12] suggests an approach for developing an architecture, differentiating between the "system" (what is built), the "model" (a description of the system to be built), the "system architecture" (the structure of the system), and the "overall architecture" (an inclusive set consisting of the system architecture, its function, the environment within which it will live, and the process used to build and operate it).

For the purposes of this project, the type of IT solutions addressed is consistent with a networked information system (NIS).[13] Furthermore, the overall architecture is represented by the security architecture found within an NIS, and the security architec-

ture is represented by the structure of a security system model. With a generalized system model for security in an NIS environment, architects could create instances of the system model, based upon detailed functional and risk management requirements.

Rechtin outlines the steps for creating a model as follows: (1) aggregating closely related functions, (2) partitioning or reducing the model into its parts, and (3) fitting or integrating components and subsystems together into a functioning system. The security system model will be represented by the aggregation of security functions, expressed in terms of subsystems and how the subsystems interact. The security-related functions within an NIS can be described as a coordinated set of processes that are distributed throughout the computing environment. The notion of distributed security systems, coordinated by design and deployment, meets the intuitive expectation that security within an NIS should be considered pervasive. In an NIS environment, security subsystems must be considered as abstract constructs in order to follow Rechtin's definition.

For this project, Common Criteria were considered to be the description of the complete function of the security system model. The classes and families within the Common Criteria represent an aggregation of requirements; however, after careful review, it was determined that the class and family structures defined within Common Criteria do not lend themselves to be used as part of a taxonomy for pervasive security. The aggregation is more reflective of abstract security themes, such as cryptographic operations and data protection, rather than security in the context of IT operational function. To suit the objective of this project, the Common Criteria functional criteria were re-examined and reaggregated, removing the class and family structures. An analysis of the 130 component-level requirements in relation to their function within an NIS solution suggests a partitioning into five operational categories: audit, access control, flow control, identity and credentials, and solution integrity. A summary mapping of CC classes to functional categories is provided in Table 1.

While redundancy is apparent at the class level, there is only a small overlap at the family level of the hierarchy defined within Common Criteria and below. Much of the overlap represents the intersection of function and interdependency among the categories.

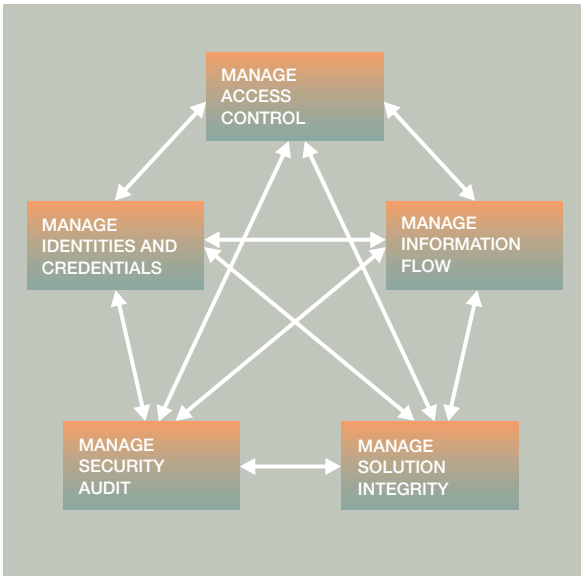
Table 1 Placing Common Criteria classes in functional categories

Functional Category	Common Criteria Functional Class
Audit	Audit, component protection, resource utilization
Access control	Data protection, component protection, security management, component access, cryptographic support, identification and authentication, communication, trusted path/channel
Flow control	Communication, cryptographic support, data protection, component protection, trusted path/channel, privacy
Identity/credentials	Cryptographic support, data protection, component protection, identification and authentication, component access, security management, trusted path/channel
Solution integrity	Cryptographic support, data protection, component protection, resource utilization, security management

Security subsystems. The component-level guidance of Common Criteria documents rules, decision criteria, functions, actions, and mechanisms. This structure supports the assertion that the five categories described in Table 1 represent a set of interrelated processes, or subsystems, for security. The notion of a security subsystem has been proposed previously; the authors of *Trust in Cyberspace*[13] described functions within operating system access control components as belonging to a decision subsystem or an enforcement subsystem.[14] The five interrelated security subsystems proposed here expand the operating system-based concept and suggest that function and interdependency of security-related functions, beyond centralized access control, can be modeled as well. (See Figure 1.)

A brief description of each of the five security subsystems, along with further detail of the aggregation of CC component-level criteria within each subsystem, is now provided. The subsystem diagrams are represented as parts of a closed-loop control system showing the internal processes that each performs, along with its external interfaces. In this representation, each subsystem consists of a managing process with a default idle state and several execution paths that can be invoked either by an asynchronous request signaled by another security subsystem or by a synchronized request from a nonsecurity pro-

Figure 1 IT security processes and subsystems



cess. Complementary representations composed of component views and interaction diagrams for the subsystems are being developed.

Security audit subsystem. The purpose of the security audit system in an IT solution is to address the data collection, analysis, and archival requirements of a computing solution in support of meeting the standards of proof[14] required by the IT environment. A security audit subsystem is responsible for capturing, analyzing, reporting, archiving, and retrieving records of events and conditions within a computing solution. This subsystem can be a discrete set of components acting alone, or a coordinated set of mechanisms among the several components in the solution. Security audit analysis and reporting can include real-time review, as implemented in intrusion detection components, or after-the-fact review, as associated with forensic analysis in defense of repudiation claims. A security audit subsystem may rely upon other security subsystems in order to manage access to audit-related systems, processes, and data, control the integrity and flow of audit information, and manage the privacy of audit data. From Common Criteria, security requirements for an audit subsystem would include:

- Collection of security audit data, including capture of the appropriate data, trusted transfer of audit data, and synchronization of chronologies
- Protection of security audit data, including use of time stamps, signing events, and storage integrity to prevent loss of data
- Analysis of security audit data, including review, anomaly detection, violation analysis, and attack analysis using simple heuristics or complex heuristics
- Alarms for loss thresholds, warning conditions, and critical events

The closed loop process for a security audit subsystem is represented in Figure 2.

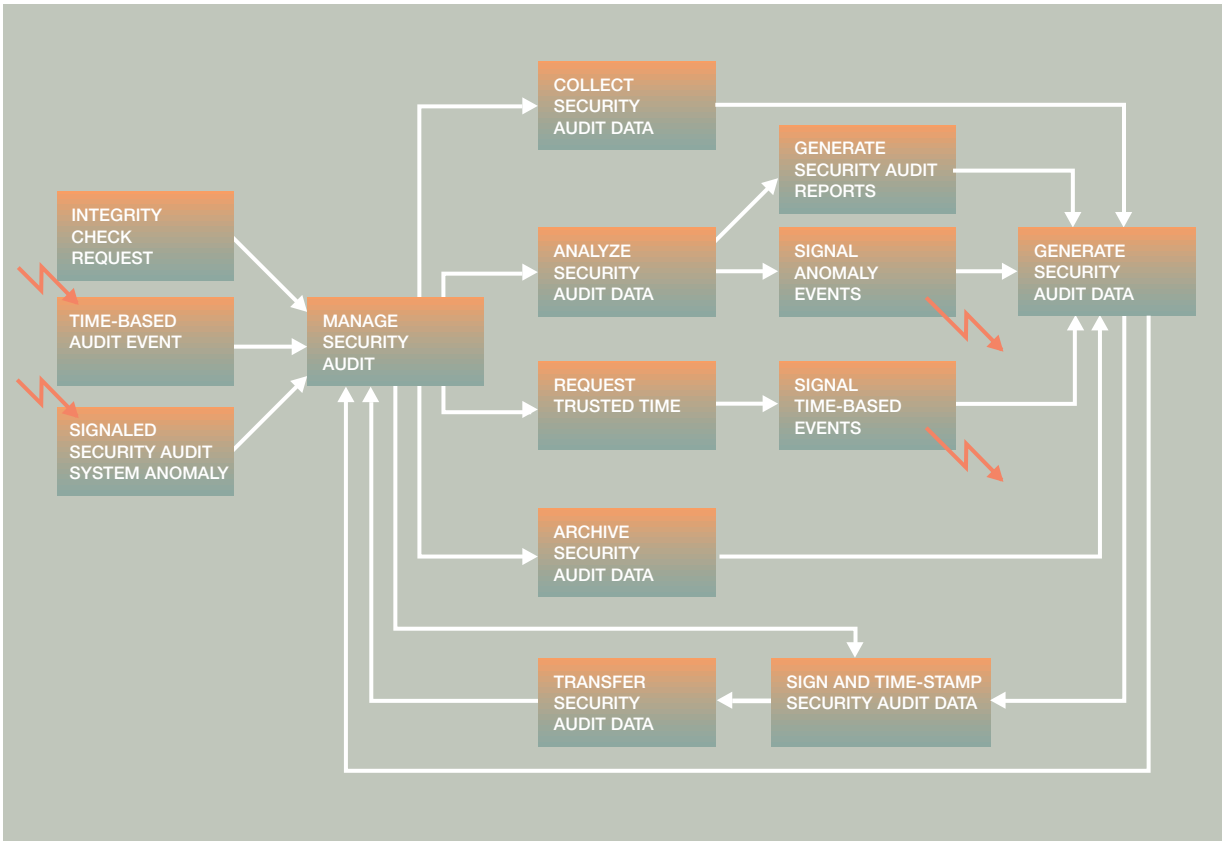
Solution integrity subsystem. The purpose of the solution integrity subsystem in an IT solution is to address the requirement for reliable and correct operation of a computing solution in support of meeting the legal[15] and technical standard for its processes. A solution integrity subsystem can be a discrete set of components or a coordinated set of mechanisms among the several components in the solution. The solution integrity subsystem may rely upon the audit subsystem to provide real-time review and alert of attacks, outages, or degraded operations, or after-the-fact reporting in support of capacity and performance analysis. The solution integrity subsystem may also rely upon the other subsystems to control access and flow. From Common Criteria, the focus of a solution integrity subsystem could include:

- Integrity and reliability of resources
- Physical protections for data objects, such as cryptographic keys, and physical components, such as cabling, hardware, etc.
- Continued operations including fault tolerance, failure recovery, and self-testing
- Storage mechanisms; cryptography and hardware security modules
- Accurate time source for time measurement and time stamps
- Prioritization of service via resource allocation or quotas
- Functional isolation using domain separation or a reference monitor
- Alarms and actions when physical or passive attack is detected

The closed loop process for a solution integrity subsystem is represented in Figure 3.

Access control subsystem. The purpose of an access control subsystem in an IT solution is to enforce security policies[16] by gating access to, and execution of, processes and services within a computing solu-

Figure 2 Security audit subsystem processes



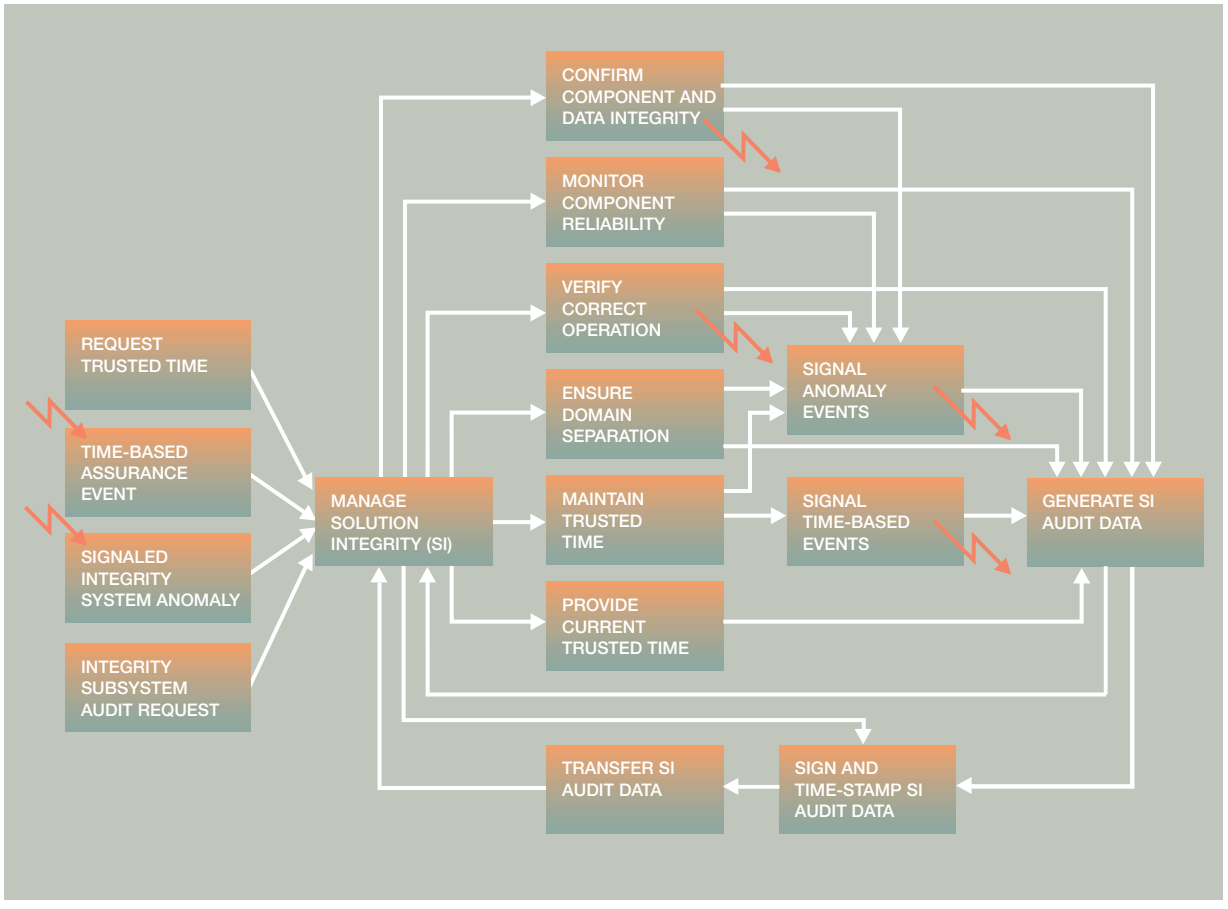
tion via identification, authentication, and authorization processes, along with security mechanisms that use credentials and attributes. The credentials and attributes used by the access control subsystem along with the identification and authentication mechanisms are defined by a corresponding credential subsystem. The access control subsystem may feed event information to the audit subsystem, which may provide real-time or forensic analysis of events. The access control subsystem may take corrective action based upon alert notification from the security audit subsystem. From Common Criteria, the functional requirements for an access control subsystem should include:

- Access control enablement
- Access control monitoring and enforcement
- Identification and authentication mechanisms, including verification of secrets, cryptography (encryption and signing), and single- vs multiple-use authentication mechanisms
- Authorization mechanisms, to include attributes, privileges, and permissions
- Access control mechanisms, to include attribute-based access control on subjects and objects and user-subject binding
- Enforcement mechanisms, including failure handling, bypass prevention, banners, timing and timeout, event capture, and decision and logging components

The closed loop process for an access control subsystem is represented in Figure 4.

Information flow control subsystem. The purpose of an information flow control subsystem in an IT solution is to enforce security policies[16] by gating the flow of information within a computing solution, affecting the visibility of information within a computing solution, and ensuring the integrity of information flowing within a computing solution. The information flow control subsystem may depend upon

Figure 3 Integrity subsystem processes



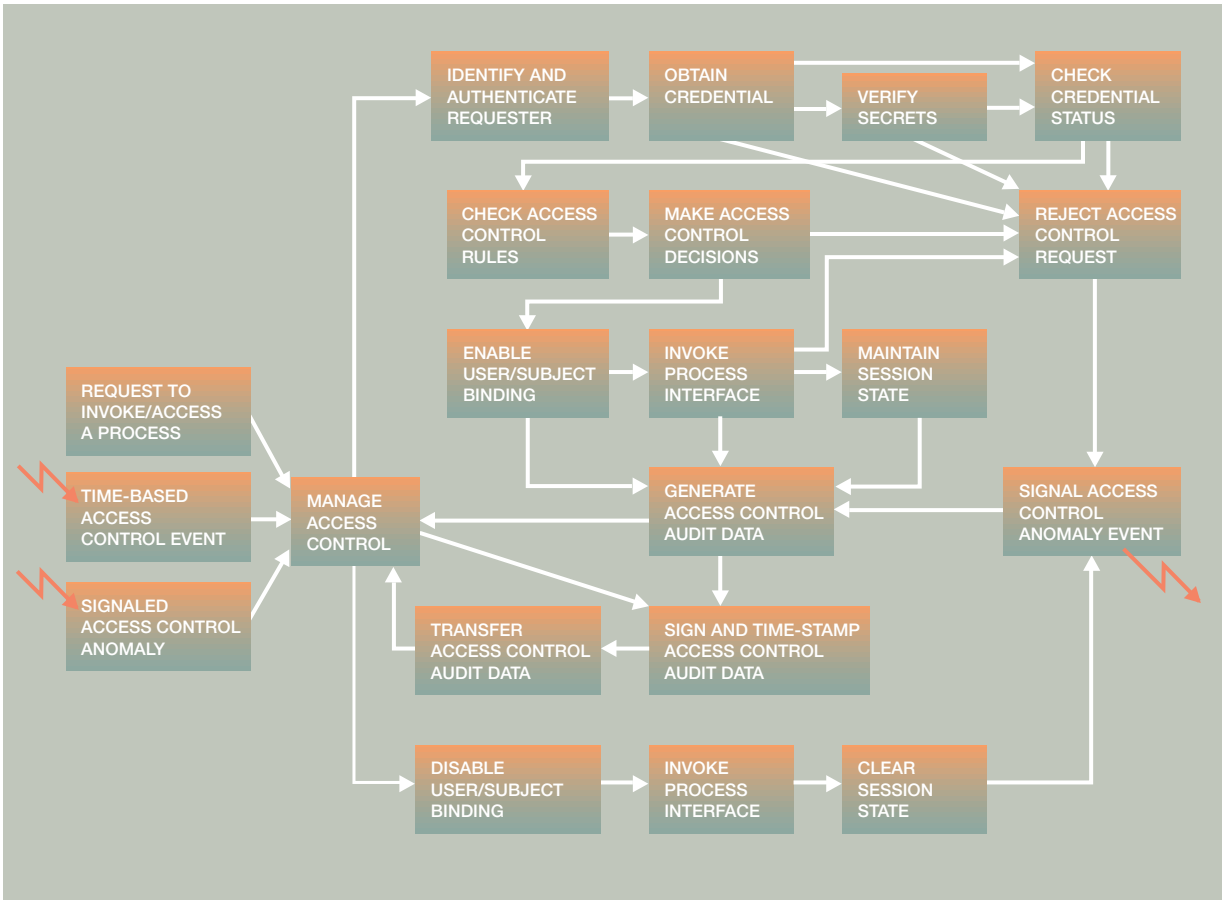
trusted credentials and access control mechanisms. This subsystem may feed event information to the security audit subsystem, which may provide real-time or forensic analysis of events. The information flow control subsystem may take corrective action based upon alert notification from the security audit subsystem. From Common Criteria, an information flow control subsystem may include the following functional requirements:

- Flow permission or prevention
- Flow monitoring and enforcement
- Transfer services and environments: open or trusted channel, open or trusted path, media conversions, manual transfer, import to or export between domains
- Mechanisms observability: to block cryptography (encryption)
- Storage mechanisms: cryptography and hardware security modules
- Enforcement mechanisms: asset and attribute binding, event capture, decision and logging components, stored data monitoring, rollback, residual information protection and destruction

The closed loop process for an information flow control subsystem is represented in Figure 5.

Identity or credential subsystem. The purpose of a credential subsystem in an IT solution is to generate, distribute, and manage the data objects that convey identity[15] and permissions across networks and among the platforms, the processes, and the security subsystems within a computing solution. In some applications, credential systems may be required to adhere to legal criteria[15] for creation and mainte-

Figure 4 Access control subsystem processes



nance of trusted identity used within legally binding transactions.

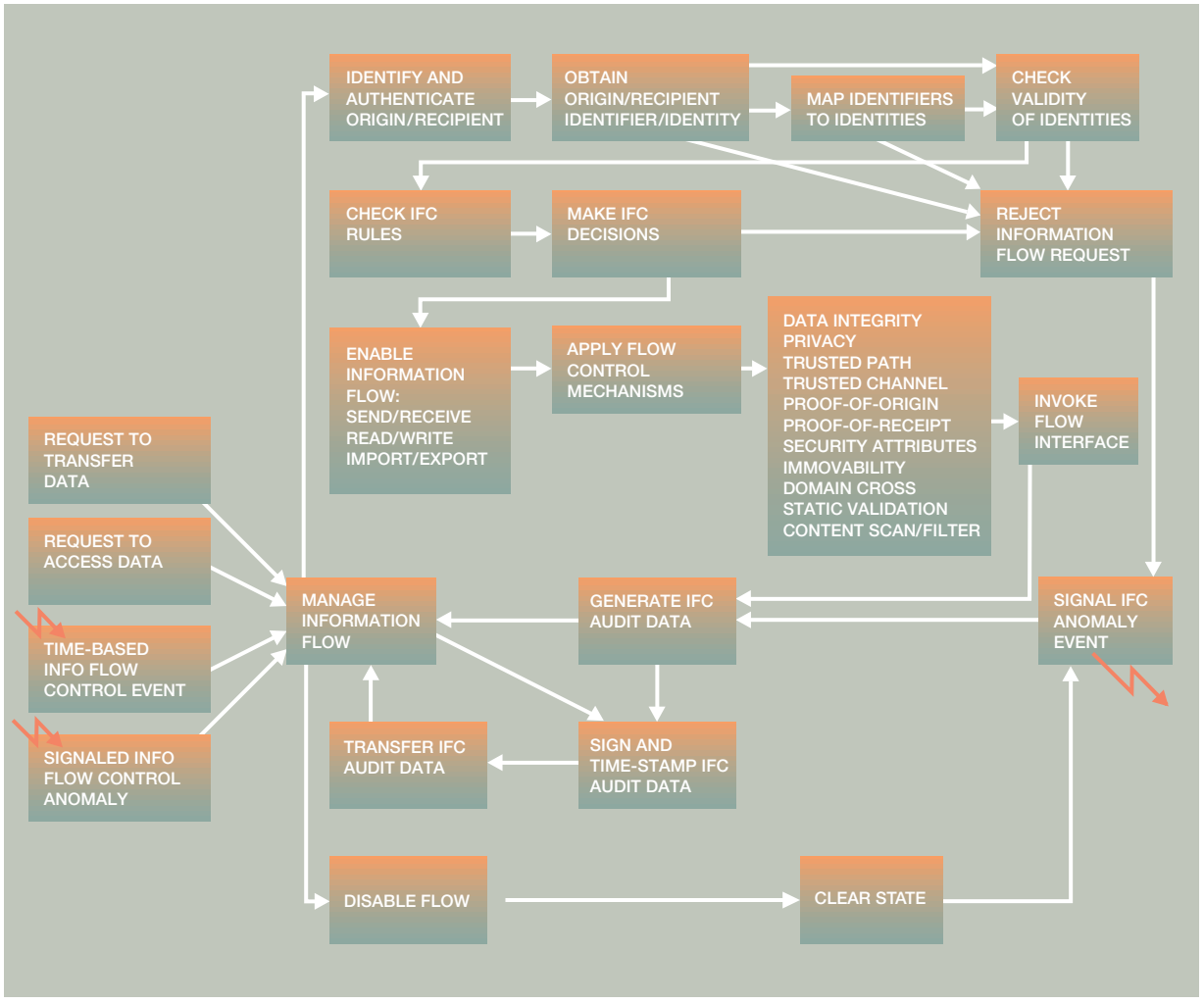
A credential subsystem may rely on other subsystems in order to manage the distribution, integrity, and accuracy of credentials. A credential subsystem has, potentially, a more direct link to operational business activities than the other security subsystems, owing to the fact that enrollment and user support are integral parts of the control processes it contains. From Common Criteria, a credential subsystem may include the following functional requirements:

- Single-use vs multiple-use mechanisms, either cryptographic or noncryptographic
- Generation and verification of secrets
- Identities and credentials to be used to protect security flows or business process flows
- Identities and credentials to be used in protection of assets: integrity or nonobservability
- Identities and credentials to be used in access control: identification, authentication, and access control for the purpose of user-subject binding
- Credentials to be used for purposes of identity in legally binding transactions
- Timing and duration of identification and authentication
- Life cycle of credentials
- Anonymity and pseudonymity mechanisms

The closed loop process for a credential subsystem is represented in Figure 6.

Summary of the security system model. This study postulates that the five security subsystems described here exist within every IT solution at the conceptual

Figure 5 Information flow control subsystem processes



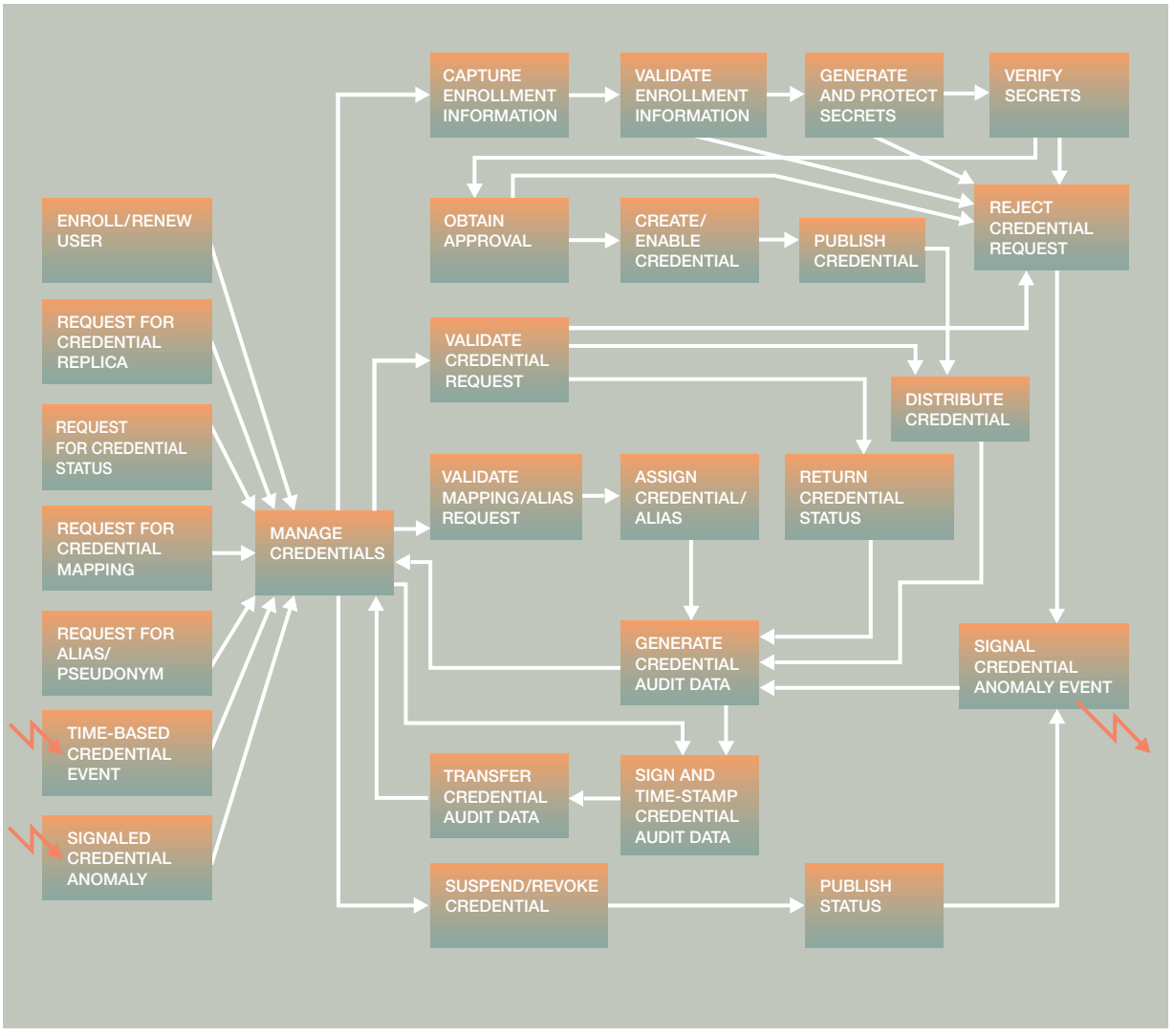
level, and that the design, integration, and interworking of the services and mechanisms associated with these subsystems represent the security functionality of the solution. This "security system model" needs to be combined with a method for developing the detailed security architecture for a given IT solution.

A systematic approach to developing security architectures

A system architecture has been defined as "the structure of the system to be built." In this study, the "system to be built" consists of the security control system found within a networked information system. Figure 7 represents the solution environment. Here an e-business computing solution serves information or supports electronic commerce transactions via the Internet. The e-business computing solution is operated by an enterprise and provides services to one or more user communities.

The e-business computing solution can be described as a set of automated business processes supporting the business context that requires security assurances and protections. The design goal is to infuse security into the computing solution and the related IT environment.

Figure 6 Credential subsystem processes

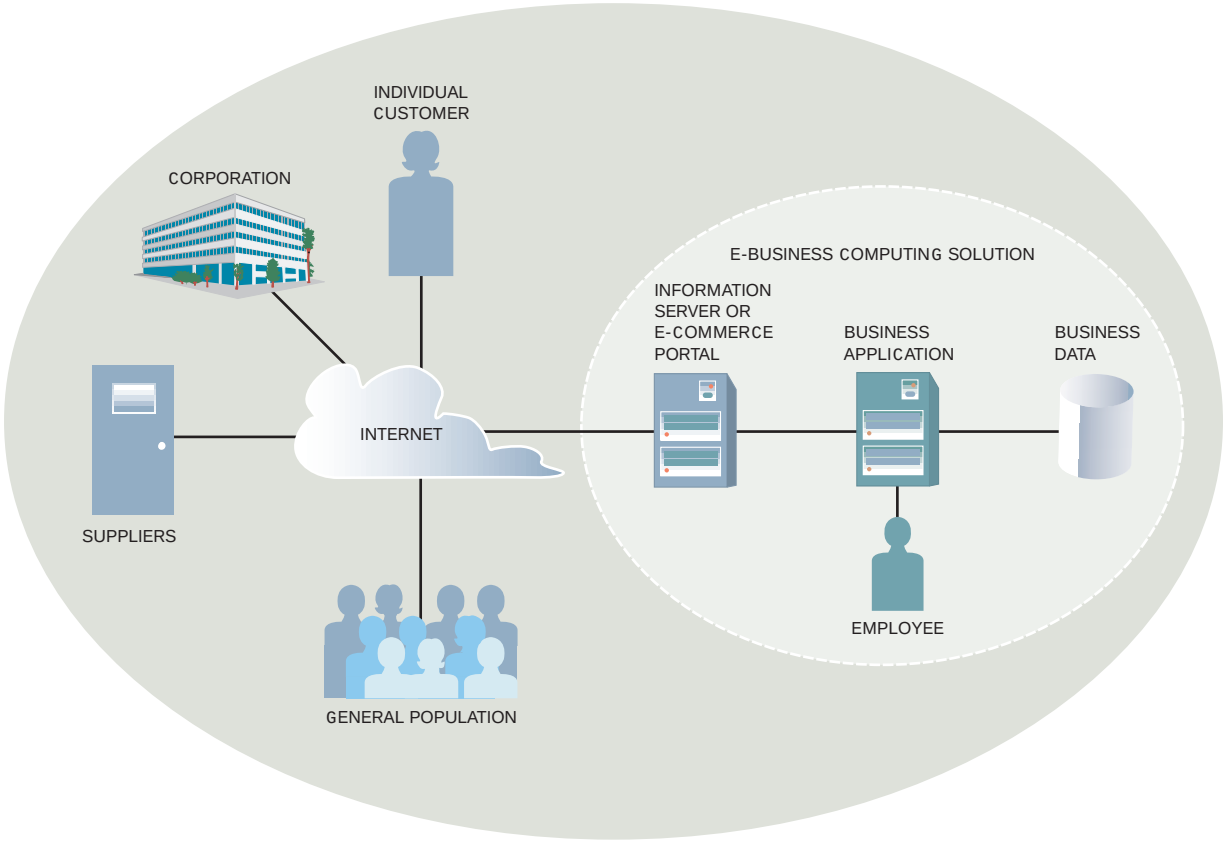


From a business perspective, there are two objectives: (1) to ensure that the desired IT business process flow yields correct and reliable results, and (2) to ensure that the potential vulnerabilities and exception conditions (i.e., perils) within IT business process flows are addressed in ways that are consistent with the risk management objectives. These objectives show the duality of security design: to support and assure normal flows, as well as identify and account for all illicit flows and anomalous events.

Business process model. Figure 8 represents IT process flows for a generalized business system. The process flows reflect the events and conditions in which information assets are acted upon by processes that are invoked by users, or by processes acting on behalf of users. The left arrow represents the model business flow within a trusted environment, and the right arrow represents a more realistic view of the business flow, where perils exist in the operating environment.

Security design objectives. Traditionally, security requirements have been expressed by referencing the security services within the OSI model:[6] authentication, access control, data confidentiality, data integ-

Figure 7 Networked information system environment



rity, and nonrepudiation. This practice introduces ambiguity when applied in the context of business processes. This ambiguity can contribute to a miscommunication of security requirements and a mismatch of functionality within the computing solution. As with other architecture disciplines, the technical objectives of the security design activity need to be articulated in quantifiable terms. Specific design objectives need to be developed and validated for each solution. For reference in this project, the following set of security design objectives were derived as a result of an analysis of the security-incident handling and reporting system for one corporation:

1. There is a need to control access to computer systems and their processes, consistent with defined roles and responsibilities.
2. There is a need to control access to information, consistent with information classification and privacy policies.
3. There is a need to control the flow of information, consistent with information classification and privacy policies.
4. There is a need to manage the reliability and integrity of components.
5. There is a need for protections from malicious attack.
6. There is a need for trusted identity to address the requirement of accountability of access to systems, processes, and information.
7. There is a need to prevent fraud within business processes and transactions, or to detect and respond to attempted fraud.

Selection and enumeration of subsystems. The security design objectives and the solution environment have a central role in the selection and enumeration of subsystems. Table 2 shows a possible mapping of the example design objectives to security subsystems. It indicates where a subsystem may be required (R)

Figure 8 The normal and peril IT business process flow

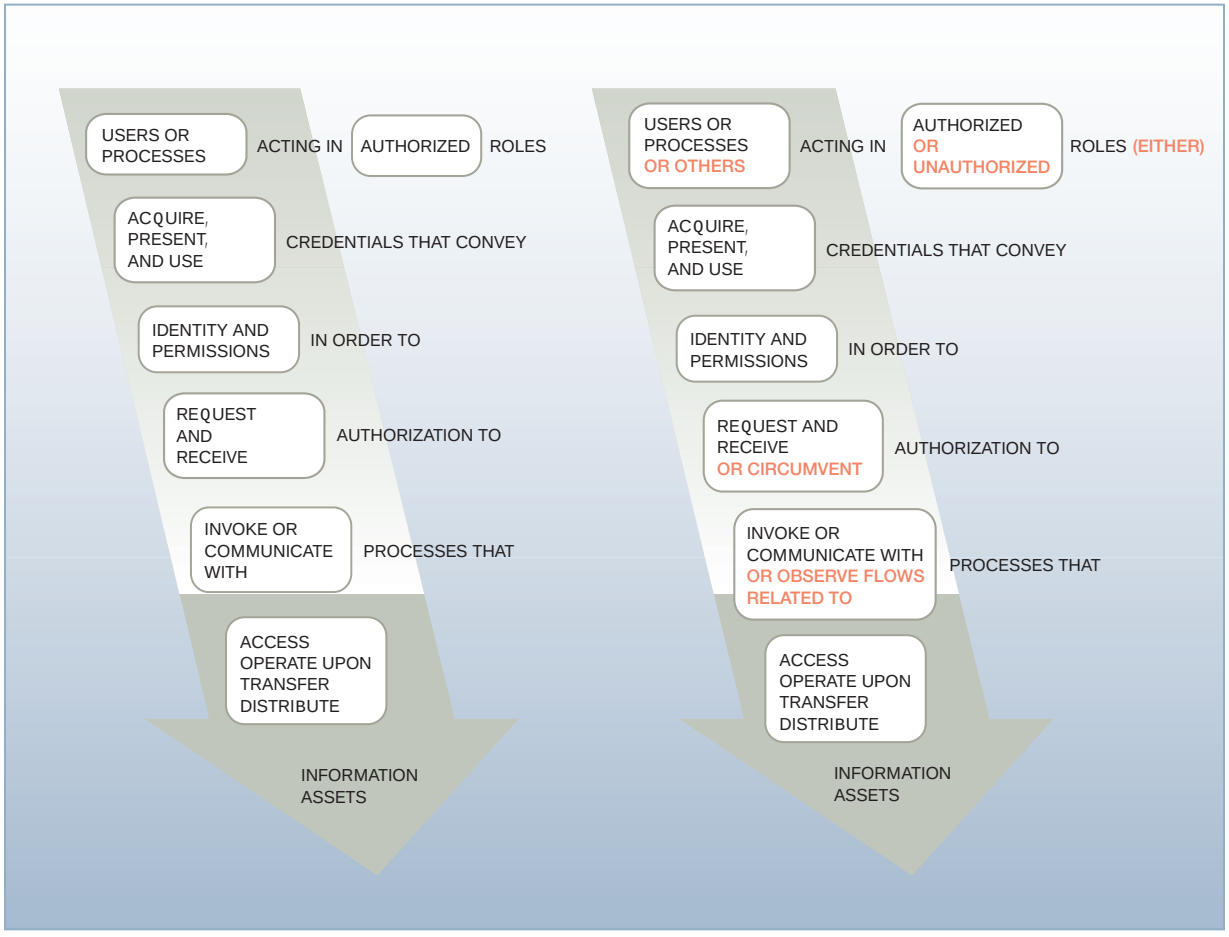


Table 2 Mapping design objectives to security subsystems

Security Design Objectives	Audit	Integrity	Access Control	Flow Control	Credentials/ Identity
Control access to systems/processes	S	S	R	S	S
Control access to information	S	S	S	R	R
Control the flow of information	S	S	S	R	S
Correct and reliable component operation	S	R	S	S	S
Prevent/mitigate attacks	R	R	R	R	S
Accountability through trusted identity	R	R	S	S	R
Prevent/mitigate fraud	R	R	R	R	R

or supplementary (S) in satisfying an individual security requirement. Actual subsystem selection requires documented rationale.

There are many interrelated factors that determine how many instances of a given subsystem appear in the solution. Table 3 suggests motivations for instan-

Table 3 Determining the security subsystems in a design

Subsystem	Number in a Design	Characteristics of the Computing Environment
Security audit subsystem	Few	One subsystem for archive of related critical data One subsystem for analysis of related anomalies One subsystem for fraud detection in the solution
Integrity	Few	One subsystem per group of related critical components
Access control	1 to *n*	One subsystem per unique user-subject binding mechanism or policy rule set
Flow control	1 to *m*	One subsystem per unique flow control policy rule set One or more flow control functions per OSI layer service: physical, datalink, network, end-to-end transport, application One or more flow control functions per domain boundary
Credentials/identity	1 to *k*	Some number of credential systems per domain Some number of credential classes per domain Some number of disparate credentials or uses for credentials per domain Some number of aliases/pseudonyms at domain boundaries

tiating security subsystems within a design. Actual subsystem enumeration requires documented rationale.

Documenting conceptual security architecture. Given the agreed-upon design objectives, a conceptual model for security within the IT solution can be created. Figures 9A and 9B represent a conceptual security architecture. For clarity, security functions have been grouped by design objective.

The diagrams represent the solution environment segmented by risk profile or operational affinity, along with icons for security functions. The legend for the diagrams maps the security subsystems to icons. The information flow control subsystem has a wide range of functions. For this reason, a rectangle is used to indicate a policy evaluation and enforcement function, whereas an oval indicates a data flow function, such as a communication protocol with security capabilities.

From the perspective of the enterprise deploying the solution, the security design objectives will dictate where security functionality is desired; however, the compliance to some or all of the security requirements may be limited by the enforceability of policies beyond the boundaries of the enterprise. Whether and how these credential subsystems and access control subsystems can be integrated into the security architecture can have a major impact on the trustworthiness of the solution as a whole. These issues and dependencies should be considered and documented within architectural decisions.

This type of conceptual model forms the baseline for developing and evaluating a "proof-of-concept" and further refinement of the functional aspects of security within the target environment.

Integrating security into the overall solution architecture

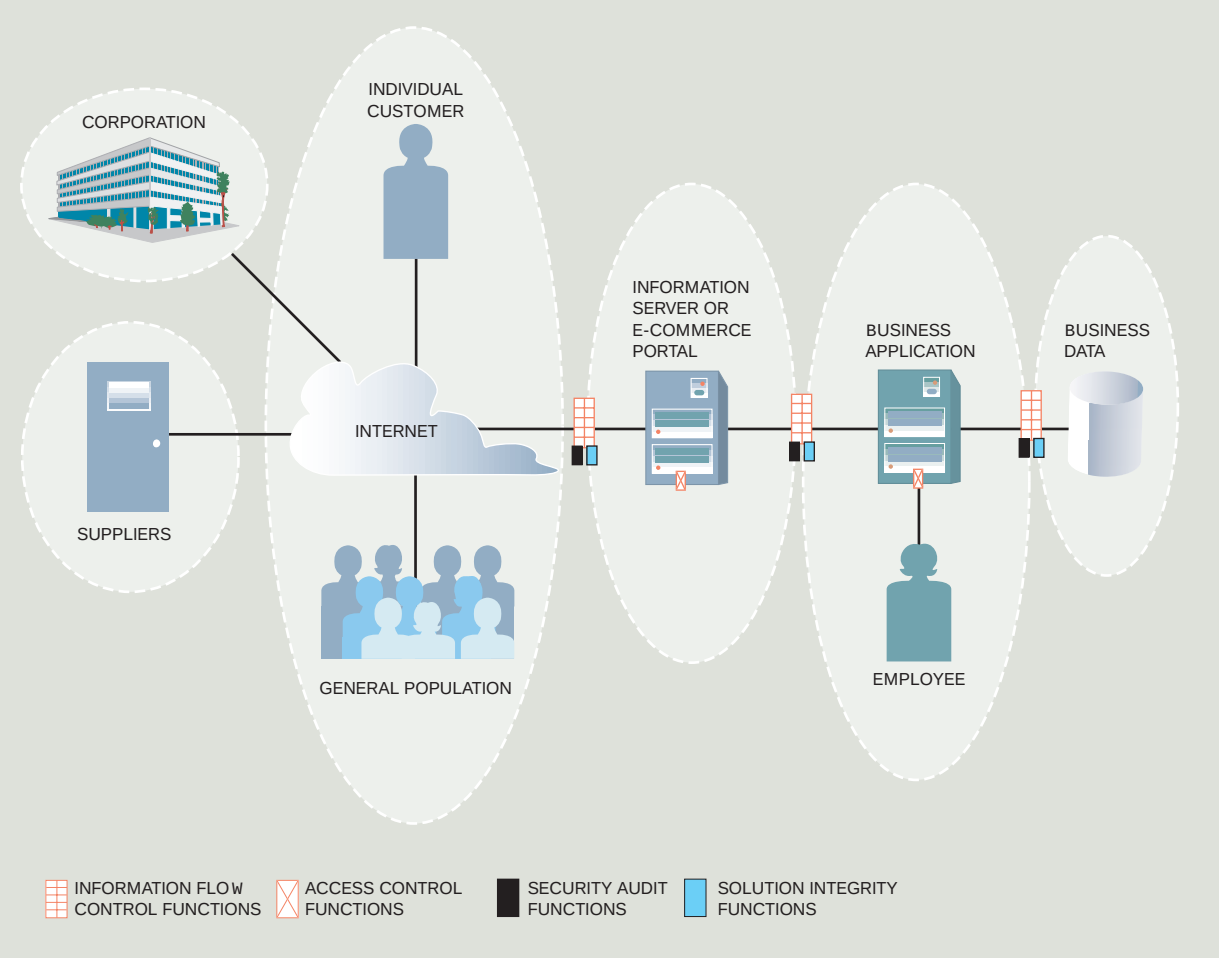
There are several steps involved in translating the conceptual security subsystem functions into component-level specifications and integration guidance. These include: creating models of the solution environment, documenting architectural decisions, developing use cases, refining the functional design, and integrating security requirements into component architectures.

Solution models. Creating an initial solution model is a critical step in the design process. With skill and experience, one-of-a-kind solution models can be developed to fit a given set of requirements. For complex solutions, the practice of using templates derived from prior solutions is becoming commonplace.

The Enterprise Solutions Structure (ESS) provides a range of reference architectures[17] for e-business solutions.

Documenting architectural decisions. Previously, the notion of the duality of security design was described, that is, ensuring correct and reliable operation and protecting against error and maliciousness. Both motivations are based upon managing the business risks of the solution and of the environment. Risks rep-

Figure 9A Defending against attacks: Flow control interfaces, access control, audit and integrity functions within networked information system environments



resent the likelihood that an undesirable outcome will be realized from a malicious attack, unexpected event, operational error, etc. Risks are either accepted as a cost of operation, transferred to some other party, covered by liability insurance, or mitigated by the security architecture.

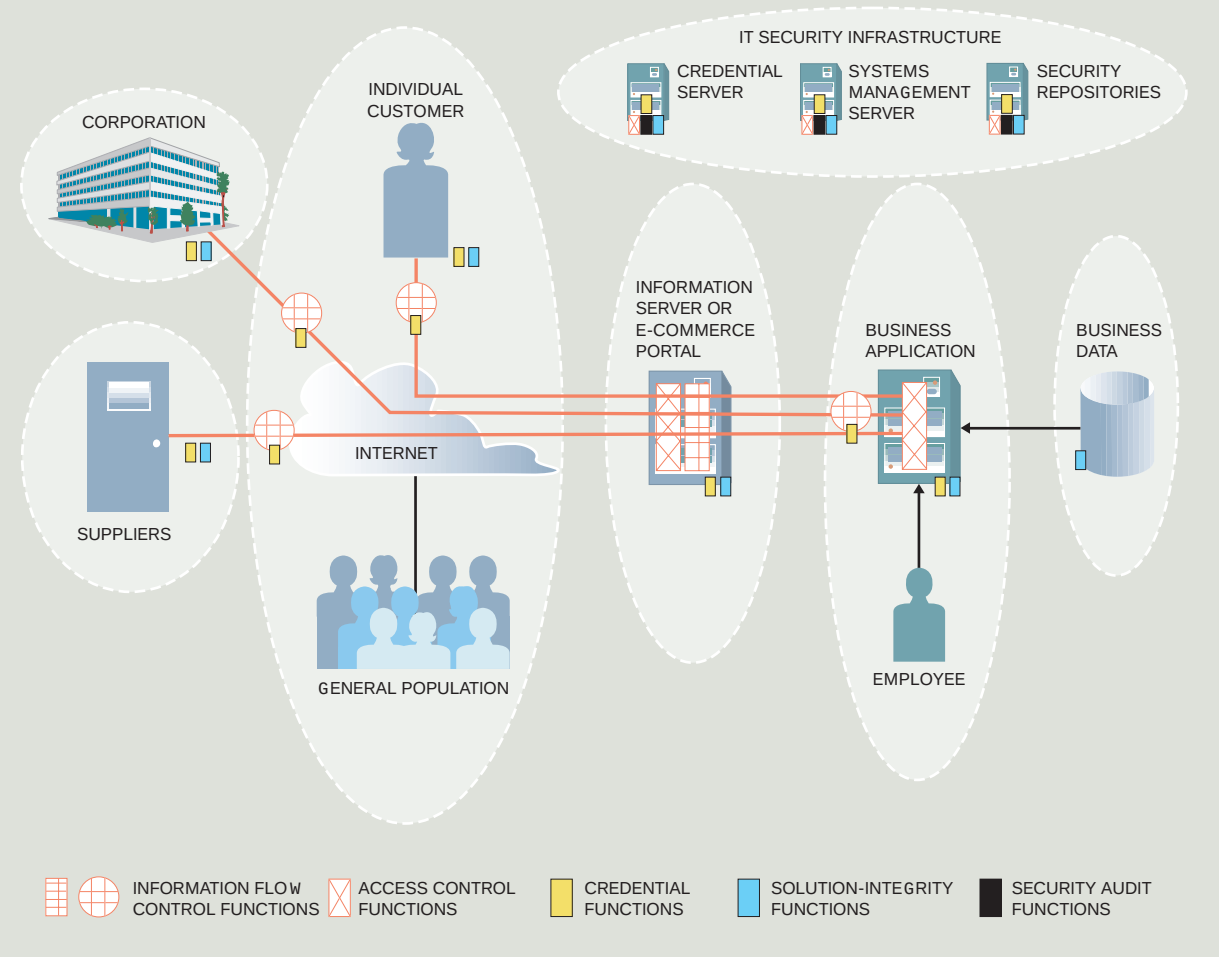
Architectural decisions will dictate how robust the security system architecture should be, which security subsystems to incorporate into the system architecture, which functions and mechanisms within each subsystem should be deployed, where the mechanisms will be deployed, and how the deployment will be managed.

Examples of architectural decisions include:

- Viability of the countermeasures, including the threats addressed, the limitations and caveats of the solution, and the resulting window of risk
- Extensibility of the design, including whether or not the design will serve the total population and if there will be separate designs for defined population segments
- Usability of the design, including whether or not the mechanisms integrate with the technology base and the extent of the burden of compliance for users
- Manageability of the design, including the extent of the burden of life-cycle management

Use cases. Architectural decisions will also drive the evaluation of prototypes and models of functions

Figure 9B Ensuring correct and reliable operation: Access control interfaces, flow control protection mechanisms, credentials, security audit, and solution integrity functions within networked information system environments



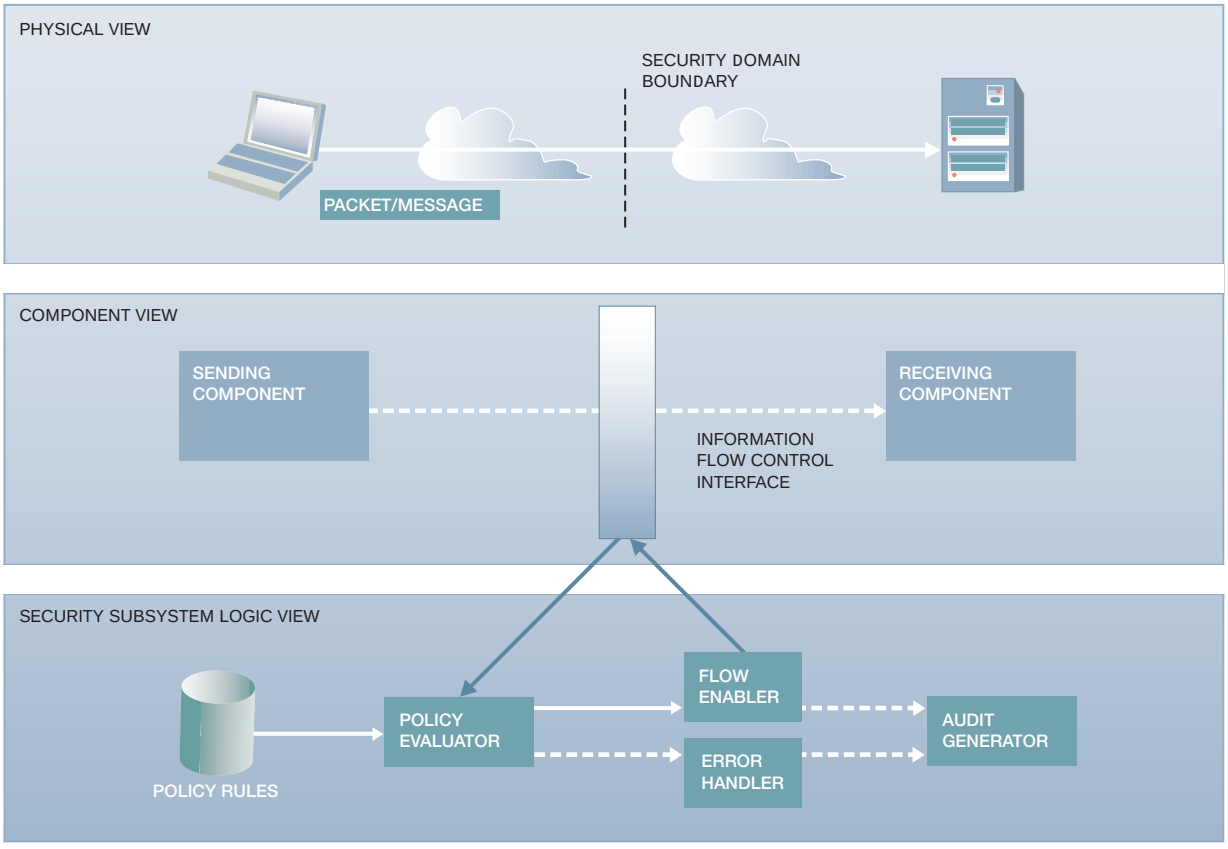
within the solution. One form of prototype is called a *use case*. Both security threats and normal interactions and flows can be validated with use cases.

Example 1: Interception of errant packet or message flow. Figure 10 represents several levels of detail for the operation of an information flow control subsystem that is designed to monitor send and receive operations that cross a boundary between two networks. The computer systems are represented in the physical view. In the component view, an information flow control interface, positioned between source and destination, will examine one or more aspects of packets or messages sent across the boundary. Some components of this information flow control subsystem are shown in the logic view, where the monitored conditions and the programmed actions are carried out, based upon a set of rules.

Valid packets are allowed to flow across the boundary; however, packets or messages of a specified format, or from an invalid source, or to an invalid destination, are disabled by the security subsystem. A record of the event is generated by invoking an interface to a security audit subsystem.

This example is representative of the type of filtering, analysis, and response that is performed within packet filter firewalls, or electronic mail gateways.

Figure 10 Boundary flow control with security subsystems

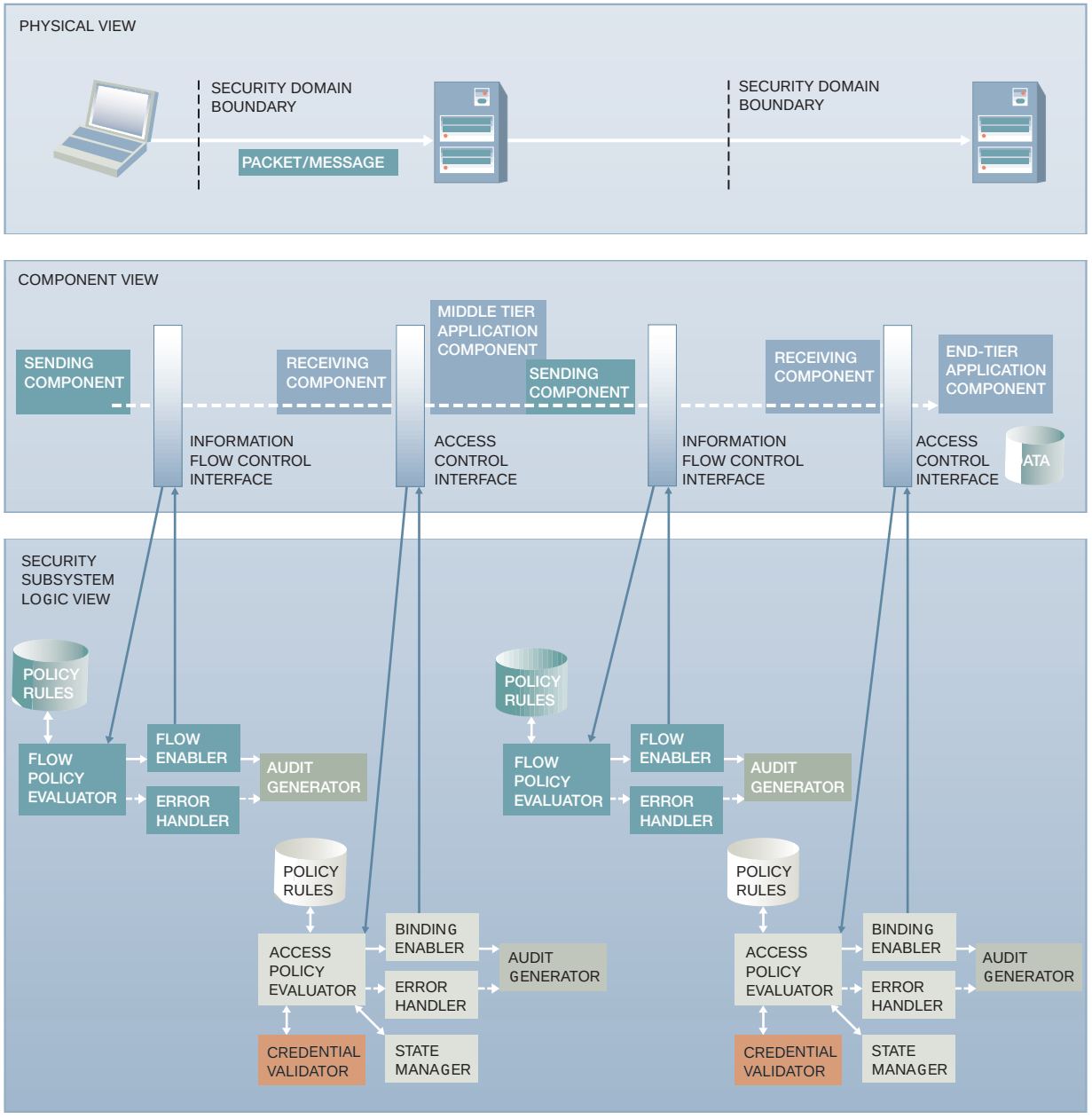


There are many architectural decisions to be evaluated within each iteration of the design. The effect on performance due to processing delays, plus the effect of data collection and analysis on the overall operation of the solution, are significant factors.

Example 2: Three-tier client/server input flow. Figure 11 illustrates an input flow for a three-tier client/server process that is typical of the integration of enterprise computing with the Internet environment. Several instances of security subsystems are depicted, spread among three network security domains. An information flow control subsystem is positioned at the boundary points between networks. An access control subsystem is positioned between a receiving component and its corresponding application component. Interfaces to related credential subsystems and security audit subsystems are shown in the security subsystem logic view. No integrity subsystem functions are referenced in this example. The scenario follows:

1. The business process interface is invoked by a user or a process and the request is transferred via a sending component.
2. The request flows across a security domain in a manner that is acceptable to the sending and receiving components, based upon the defined information flow control rules.
3. Identification, authentication, and access control decisions are made based upon the external identity associated with the request by an access control subsystem associated with the middle-tier application.
4. The middle-tier application is invoked via a user-subject binding. The actual processing is not covered here—it may include business presentation and data mapping logic, or it may be performed by an application-level information flow control subsystem, such as a proxy server.
5. The middle-tier application initiates, or relays, a request to the end-tier application. The request

Figure 11 Three-tier client/server input flow with security subsystems



is scrutinized at another network boundary control point.

6. At the end-tier application, an access control decision may be performed on the request relative to the identity of the user represented by the middle-tier application, depending on the design of the application and the exchange protocols used.
7. The business process is invoked by a user-subject binding if the access control decision is positive.

This demonstrates how security functions from several subsystems are distributed throughout the solution. As with the first example, architectural decisions will guide the design of the security subsystem

functions, which in turn may put constraints on the overall business flow in order to achieve the risk management objectives.

Refining the functional design. Walk-throughs of complete business processes, including exception conditions and handling processes, assist in creating a viable solution outline and refining requirements and interdependencies among the solution building blocks.

Example 3: PKI digital certificate enrollment. This example uses the credential subsystem model to describe the generalized flow for enrolling a user into an identity or credential system based upon PKI digital certificates[18] as the first step in developing a security system architecture. The process involves combining the subsystem model with assumptions about the business environment, the business processes, the risk management requirements, the technical specifications, and possibly the legal and business compliance requirements[16] associated with issuing PKI digital certificates.

In Figure 12, the yellow blocks represent manual processes, the blue blocks map to automated processes, and the peach blocks map to automated audit data capture points. The blue data storage icons represent sensitive repositories, the pink icons map to cryptographic secrets, the white icons represent unique contents of the certificate, and the lavender icon is associated with the certificate.

The enrollment process flow depicted demonstrates the exchange of sensitive user information and secrets, plus the export of the credential outside the control of the issuer. The full enrollment scenario should include processes from a corresponding information flow control subsystem. For public key credentials, the format of certificates, along with details of how the credentials are formatted, transported, and stored are important design considerations. All scenarios must be validated against existing and proposed business processes. Validation of the scenarios substantiates the architectural decisions discussed earlier. Subsequent design steps are needed to develop and map the functions of the security subsystems to Common Criteria specifications and ultimately onto the nodes and physical components.

Integrating security requirements into component architectures. The security functions within the design need to be apportioned throughout the solution. However, many of the mechanisms and services within the IT solution that implement security functionality operate within other than security components, for example: database systems, application systems, clients, servers, and operating systems. The task of adopting security function into the network, application, middleware, security, systems management, and infrastructure architectures is shared by the several architects and integration specialists involved in the design project. The process involves a structured approach, considering the purposeful allocation of functions and requirements throughout the component architectures by:

- Mandate, based upon a legal or contractual compliance requirement
- Best practice for security, or for balance of security and business process
- Component capability, knowing the existence of a mechanism that supports the required process or action
- Location in the configuration, based upon interaction with components or flows
- Impact, considering the risk, security objective, or the component capacity to perform
- Necessity, because there may be no better alternative

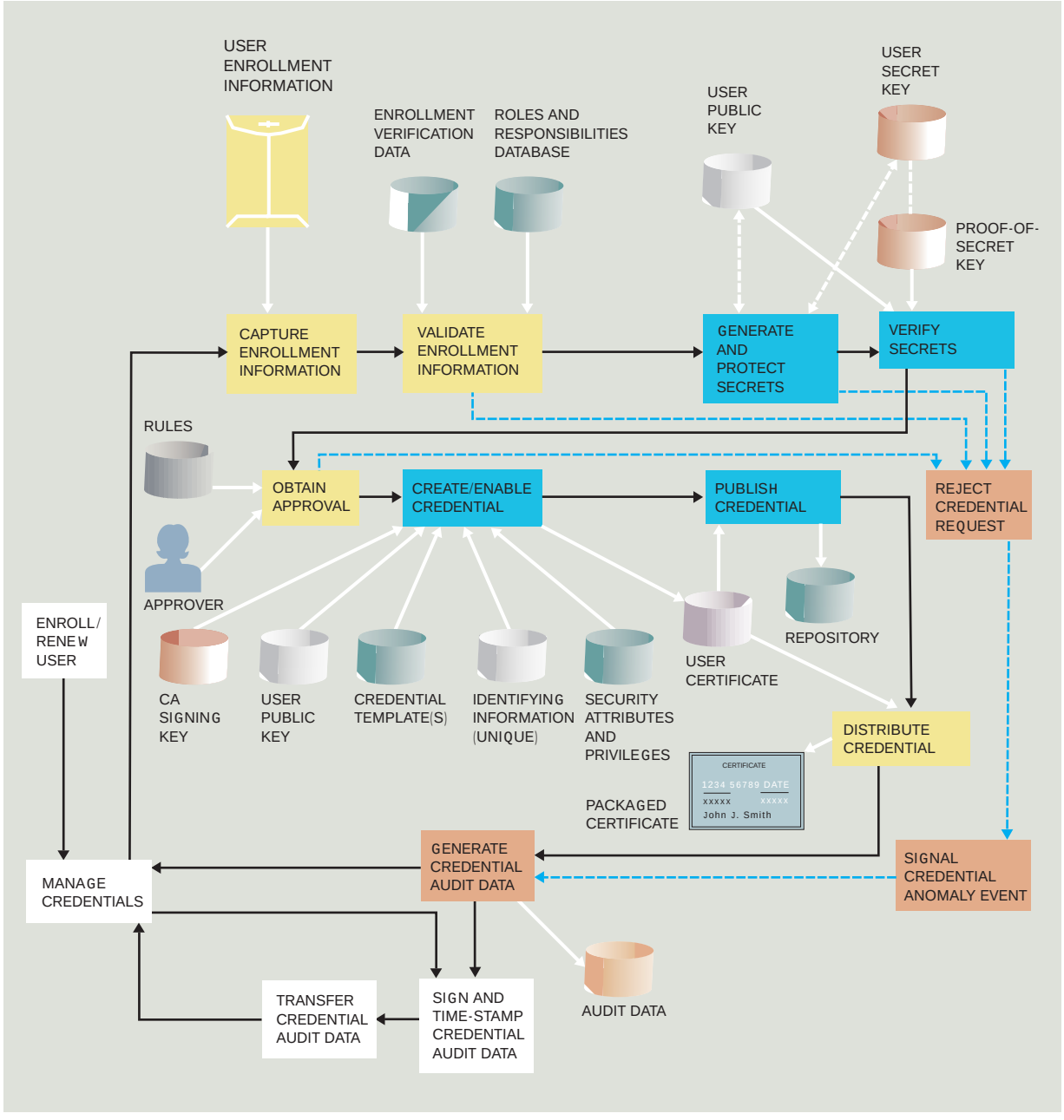
Summary of the design process. This section has described the process for translating the conceptualized security solution into a set of detailed specifications, for an integrated IT security control system, using the security subsystem construct. The design is documented, refined, and validated against the business processes through use cases and scenarios. The detailed security requirements, expressed in terms of Common Criteria component-level detail, are distributed throughout the operational model for the IT solution. At this point, integration-level detail can be finalized, and the implementation plan can proceed.

Conclusions

This paper has examined the issues and circumstances that affect the design of comprehensive security functions for computing solutions. It has outlined a system model and a systematic process for security design with the Common Criteria international standard at its foundation.

Several summary observations can be made relative to this proposed model and process: security is a shared responsibility among all IT design disciplines; security design is linked to business objectives be-

Figure 12 Sample PKI digital certificate enrollment process flow



yond the need for protecting against attack, and conversely, protecting against attack does not in itself meet all the business requirements for security; and many, if not most, security control points within IT solutions are found in portions of solutions that are not typically considered security components.

Reliable and correct operation of solutions using secure data exchange protocols, such as IPSec and secure sockets layer, is predicated on functions within all five of the security subsystems defined in the proposed model and design process. These protocols are based upon trusted identities that utilize crypto-

graphic keys requiring storage integrity, reliable key exchange protocols, strong access control mechanisms, reliable data exchange protocols, and trusted audit trails for enrollment and key life-cycle management. Furthermore, the proposed model provides a new perspective for viewing Common Criteria protection profiles in the context of security subsystems. For example, the protection profile for an application gateway firewall suggests the functionality of all five security subsystems. The fact that a front-line security device, such as a firewall, might fit the definition of a credential subsystem highlights the critical nature of its design, integration, and operation.

Actions and further study

The concepts and the supporting detailed information presented in this paper were incorporated into training for IBM Global Services architects this year. Additional work is underway to develop notations, models, and visualization techniques that enhance its adoption in related methods and architect disciplines. A patent application has been filed for the system and process, designated Method for Architecting Secure Solutions, or MASS.

Acknowledgments

This study and the project that it represents would not have been possible without the leadership of Art Gilbert, who works tirelessly in building and linking the Security and Privacy Services methodologies into the larger IBM Global Services community. It is truly an honor to work with Art. Additionally, the synergy between the concepts presented in this paper and the tools and notations used by IBM Global Services architects is attributable to the insight and hard work of Mark Buckwell, Imran Tyabji, and Bill Blake, who are coinstructors of the class. I would also like to thank several of my colleagues for their assistance, support, and understanding in this endeavor, including: Jean-Francois Ragu, Wiel Bruls, Andre Carrington, and Hamid Bacha, plus the many hearty souls who participated in the review sessions and early classes.

Cited references

1. J. J. Whitmore, "Security and e-business: Is There a Prescription?" *Proceedings, 21st National Information Systems Security Conference*, Arlington, VA (October 6–9, 1998); available at http://csrc.nist.gov/nissc/1998/proceedings/paperD13.pdf.
2. D. Verton, "Common Ground Sought for IT Security Requirements," *Computerworld* **35**, No. 11, 8 (March 12, 2001).
3. P. B. Checkland, *Systems Thinking, Systems Practice*, John Wiley & Sons, Inc., New York (1981).
4. W. R. Cheswick and S. M. Bellovin, *Firewalls and Internet Security: Repelling the Wily Hacker*, Addison-Wesley Publishing Co., Reading, MA (1994).
5. RFC 1825, Security Architecture for the Internet Protocol (August 1995); available at http://www.ietf.org/rfc.html.
6. Security Architecture for Open Systems Interconnection for CCITT Applications, ITU-T Recommendation X.800/ISO 7498-2 (1991). Obtainable from http://www.itu.int/itudoc/itu-t/rec/x/x500up/x800.html.
7. Information Technology—Security Techniques—Evaluation Criteria for IT Security—Part 1: Introduction and General Model, ISO/IEC 15408-1 (1999); available from http://isotc.iso.ch/livelink/livelink/fetch/2000/2489/lttf_Home/PubliclyAvailableStandards.htm.
8. Information Technology—Security Techniques—Evaluation Criteria for IT Security—Part 2: Security Functional Requirements, ISO/IEC 15408-2 (1999).
9. Information Technology—Security Techniques—Evaluation Criteria for IT Security—Part 3: Security Assurance Requirements, ISO/IEC 15408-3 (1999).
10. See http://www.commoncriteria.org/protection_profiles/pp.html.
11. Guide for Development of Protection Profiles and Security Targets, ISO/IEC PDTR 15446, available at http://csrc.nist.gov/cc/t4/wg3/27n2449.pdf, pp. 69–74.
12. E. Rechtin, *Systems Architecting: Creating and Building Complex Systems*, Prentice Hall, New York (1991).
13. Committee on Information Systems Trustworthiness, National Research Council, *Trust in Cyberspace*, National Academy Press, Washington, DC (1999).
14. A. Patel and S. O. Ciardhuain, "The Impact of Forensic Computing on Telecommunications," *IEEE Communications Magazine* **38**, No. 11, 64–67 (November 2000).
15. *Digital Signature Guidelines*, American Bar Association (1996), Section 1.35, available from http://www.abanet.org/scitech/ec/isc/dsgfree.html.
16. F. B. Schneider, "Enforceable Security Policies," *ACM Transactions on Information and System Security* **3**, No. 1, 30–50 (February 2000).
17. P. T. L. Lloyd and G. M. Galambos, "Technical Reference Architectures," *IBM Systems Journal* **38**, No. 1, 51–75 (1999).
18. H. Johner, S. Fujiwara, A. S. Yeung, A. Stephanou, and J. Whitmore, *Deploying a Public Key Infrastructure*, Redbook SG24-5512-00, IBM Corporation, http://www.redbooks.ibm.com.

General references

S. McClure, J. Scambray, and G. Kurtz, *Hacking Exposed: Network Security Secrets & Solutions*, McGraw-Hill Publishing Company, Maidenhead, Berkshire (1999).

RFC 2316, Report of the IAB Security Architecture Workshop (April 1998); available at http://www.ietf.org/rfc.html.

Accepted for publication May 25, 2001.

James J. Whitmore *IBM Corporation, 5267 East Simpson Ferry Road, Mechanicsburg, Pennsylvania 17055 (electronic mail: whitmore@us.ibm.com).* Mr. Whitmore is currently a senior consulting IT architect within the IBM Global Services Security and Privacy competency segment. His professional career includes 17 years in networking and information systems security at IBM, along with prior IT experience in the electric utility industry and

the U.S. government. He holds an M.S. degree in telecommunications management and a B.S. degree in electrical engineering, both from the University of Maryland. Mr. Whitmore is a member of both IEEE and the ACM.