Security Architecture Development and Results for a Distributed Modeling and Simulation System

Dr. Richard B. Neely, CISSP
Science Applications International Corporation

Abstract

This paper reports on an ongoing effort to define the security architecture for the Joint Simulation System (JSIMS), a joint military modeling and simulation system. It also describes the use of the security architecture to support the accreditation of the system.

The JSIMS security architecture must coordinate not only enclaves at different classifications, but also the independent configurations of the multiple stakeholders. These include the various military branches and their separate designated approving authorities.

It has therefore been necessary to develop the security architecture with sufficient breadth and flexibility to describe a variety of JSIMS instantiations, allowing an integrated accreditation by the multiple authorities without necessitating entirely independent accreditations. We have addressed the objective of flexibility by establishing a base, logical architecture along with customized versions of the architecture to meet the joint security objectives.

1. Introduction and background

This paper describes the security architecture of the Joint Simulation System (JSIMS), including the development approach of the security architecture as well as use within the system. JSIMS is a joint military modeling and simulation system being developed to provide a distributed training environment that integrates a number of government and military simulation models. Major security objectives are automated communication between enclaves processing data at different levels of sensitivity; supporting need-to-know based access controls throughout the distributed system; and providing accreditability in an environment of multiple independent Designated Approving Authorities (DAAs). Two major releases will be provided: Initial Operating Capability (IOC) and Full Operating Capability (FOC).

The purpose of the JSIMS security architecture is to provide a common understanding of security and a security engineering direction for JSIMS among all stakeholders: JSIMS Alliance military and government development agents (DAs), the user community, security engineers, infrastructure and model software developers, and accreditors.

The JSIMS security architecture is based on a combination of source information and constraints, including the system architecture, a variety of security requirements and needs, and development limitations. It was developed and described by a collaboration of the development agents, the Security Team, and the user community, with important input from infrastructure developers, system engineers, and accreditors. This paper reports the results of that collaboration, and the method by which the collaboration was used to produce the results.

Section 2 of this paper characterizes the problem to be solved by the security architecture, describing JSIMS security needs, limitations of available security assurance mechanisms, and other JSIMS-related challenges. Section 3 describes the objectives of the security architecture development, description, and application in solving those problems. Section 4 explains the approach used to develop the security architecture as driven by the objectives. Section 5 summarizes the result of that effort: the security architecture description. Finally, conclusions deriving from the effort are presented in Section 6.

2. JSIMS security characteristics and needs

This section presents an explanation of the need within JSIMS for a security architecture. The explanation includes the ambitious security goals of JSIMS, and in contrast the intrinsic limitations of the available security assurance mechanisms. Furthermore, constraints on the JSIMS security solution include JSIMS characteristics that are not directly related to security.

2.1. JSIMS security objectives

The security needs of JSIMS have been described in several forms, and those descriptions have required some restatement and integration to achieve a simple, coherent description of security objectives. Described briefly, the security objectives of JSIMS are to provide security support for:

- enclaves processing data at different levels of sensitivity, with a requirement for simulation-oriented communication between the enclaves;
- need-to-know separation requirements among multiple developers and data owners; and
- multiple user/developer groups with independent accreditors: all their concerns must be accounted for.

The security architecture analysis has been valuable in integrating the security needs, in formulating the security objectives, and in assuring that the JSIMS design complies with the security needs.

The JSIMS security needs derived from several sources: formal security requirements, informal security concerns of the developers and data owners, and accreditor guidance and mandates. The formal security requirements were provided by multiple sources in the combined customer/developer community and are thus based on multiple security standards. Various data owner participants have expressed additional concerns regarding protection of data, particularly in the intelligence community. Multiple DAAs associated with the various development agents must grant approval to operate JSIMS. Further, separate DAAs are associated with the TS/SCI enclave and the Secret enclave.

2.2. Assurance mechanism limitations

Design decisions driven by simulation engine performance have created the need for a monolithic process architecture on each simulation model platform, which results in a serious security assurance limitation. This tends to limit the value of incorporating a high-assurance ("trusted") operating system. Nevertheless, the use of an operating system with sensitivity label functionality and high assurance at Common Criteria (CC) [1] EAL4, with sensitivity label functionality, has other value, as explained below. The CC's EAL4 is essentially equivalent to the DAA-directed protection level.

Available automated security guards for information flow have notable limitations in functionality and assurance. Nevertheless, it is anticipated that reductions in those limitations will occur in the time frame when JSIMS needs the additional functionality and assurance, i.e., for the FOC release.

Because of the JSIMS cost profile and schedule (as well as limited availability of appropriate products), it was not possible to use a multilevel secure (MLS)-like solution, which would have eliminated the need for an enclave-based architecture and the resulting inter-enclave communication issues.

The consequence of these limitations is that multiple sources of assurance have been necessary to provide acceptable security safeguards to drive risk down to an acceptable level for accreditation. The combination of security risk management and the structure provided by the security architecture for assurance mechanisms has provided the best use of the sources of assurance.

2.3. Additional constraints

JSIMS presents several challenges in meeting the security objectives beyond those directly related to security. They include:

- the need for a "building-block" approach in order to meet early, IOC requirements while addressing FOC cost effectively;
- the lack of early accreditor availability; and
- a mandate for simulation High-Level Architecture (HLA) compliance for interfacing with external simulation exercises. (HLA is an emerging standard for modeling and simulation systems.)

The technical structuring effect of the security architecture has been beneficial in each of these areas. In particular, the security architecture description has proved to be flexible, admitting of revision and of "multi-faceted" descriptions (e.g., descriptions of earlier and of later system releases).

By a "building-block approach," we mean that while measures are taken to comply with the security requirements levied on JSIMS at IOC, such measures must be directly on the path for satisfying the greater extent of FOC security requirements. In that way, little of the pre-IOC effort must be discarded after that release, and no post-IOC cost spike will occur in preparing for FOC.

Multiple DAAs associated with multiple participating organizations and with different parts of the system architecture present an extraordinary challenge to supporting accreditation. This challenge has been increased by the limited and late involvement of DAAs, typically because of scheduling conflicts. The result has been a lack of direct guidance by the accreditors at times when security decisions have had to be made. Our approach to specifying and validating a security assurance approach has mitigated the potential risk associated with that lack of direction. In particular, the security architecture approach has organized the specifics of the security requirements and the security controls in a way that is somewhat independent of specific direction, by incorporating general, broadly accepted standards, such as the Common Criteria [1] class and family structure.

3. Objectives of the security architecture

The purpose of the JSIMS security architecture is to provide the technical structure needed to coordinate a wide variety of JSIMS development artifacts that affect infor-

mation security protection. The artifacts include the system architecture, the security objectives, the security risk management process, and security requirements to be allocated to various components. We have found, in the process of developing and communicating the security architecture, that a derived (but no less important) purpose of the security architecture is to provide feedback (sometimes not of a security nature) to many aspects of JSIMS development.

To accomplish the purpose described, the security architecture must:

- be flexible so that anticipated and even unanticipated changes to JSIMS can be addressed without debilitating impact;
- be produced, including occasional updates, in a timely way; and
- assure its content is communicated where it is needed within the system development process.

Flexibility of the security architecture has been in part a result of the security architecture development approach. Timely production of the security architecture is partly a result of flexibility, because development flexibility leads to reuse of previous work, and so a rapid turnaround.

A security architecture (and any other development artifact) would be worthless were it not communicated to appropriate audiences within the Alliance. Such communication must address both understandability and ease of interpretation. That means the security architecture must be a working document, analogous to the blueprints on a construction site that (if they are any good) end up covered in sawdust and smudged fingerprints. The JSIMS security architecture document itself forms part of the accreditation package, but it also must be a workable reference document for developers. Further, certain aspects of its content have ended up as "training" or orientation briefings for both members of the Extended Security Team and for other JSIMS developers.

It is useful to note that—though arrived at independently—these purposes and objectives for the JSIMS security architecture closely resemble the security architecture objectives and principles developed by DARPA's Information Assurance (IA) program for the Advanced Information Technology Services Reference Architecture [3]. This is particularly notable in the areas of (1) providing a structural basis for security risk management and application of security requirements (particularly to system enclaves); (2) allowing flexibility of security documentation; and (3) enabling effective integration of multiple sources of assurance.

4. Approach

This section describes the approach used in developing and communicating the JSIMS security architecture to assure that it meets the security architecture objectives given above, in turn supporting the JSIMS security objectives.

This approach consists of several facets:

- the architecture development method;
- multiple views of the security architecture as a framework;
- security risk management as a driver;
- informal effectiveness metrics; and
- documentation of the security architecture and its use.

4.1. The architecture development method

Our method for developing the security architecture has included several important characteristics:

- It has been substantially driven by the primary stakeholders, particularly the model builders ("users" of JSIMS).
- It has involved groups with a variety of needs and backgrounds.
- Multiple interaction mechanisms have been used to refine the architecture.

The multiple-view structure of the JSIMS security architecture came from several sources. Regarding the Logical view (cf. Section 5), the government DAs, who were responsible for the development of most of the simulation models, had the most direct interest in this view. They produced the original version of the Logical view, and continued to be involved in its continuing development. The Logical view has been especially important in understanding the relationships among the system architecture, the implementation, and the security objectives, and in communicating that information to a variety of development groups within JSIMS.

At the same time, the development of the entire security architecture was the result of a combination of contributors, including both the Security Team and the DAs, and to a degree the System Engineering Team and other sources. The breadth of background and expertise thus represented was key to assuring that the developed security architecture addresses the full set of security objectives, as well as maintaining consistency with all JSIMS development artifacts.

Another important aspect of the security architecture development that helped assure it met its objectives was the variety of interaction mechanisms. These included small group interaction, discussion in larger groups (e.g., the Extended Security Team, including DAs), and individuals going off to "do homework" and bring the results back for further group discussion. The larger groups were of course necessary to assure that sufficient breadth of knowledge was brought to bear on each issue. Yet smaller groups usually meant more rapid turnaround, more creativity, and more flexibility. Paraphrasing a quote

attributed to Albert Einstein: Work groups should be as small as possible, but no smaller. (The original quote: "Everything should be as simple as possible, but no simpler.")

4.2. Structuring the development

The security architecture is a complex development artifact, and must relate to the system and its development in many ways. Complexity has been controlled by defining multiple views and describing their relationships, both to one another and to other development artifacts. Specifics of the views and their relationships are given in Section 5.

A security architecture is a method of allocating security controls within a system to best counter security threats. Security controls must be allocated in a way that limits risk to an acceptable level while minimizing the cost of the controls. This means that the security risk management process defined for the target system must be used to drive the security architecture itself.

4.3. Informal effectiveness metrics

As the security architecture development proceeds, it is important to make sure that progress is being made toward the goals of the security architecture: flexibility, timeliness, and communication (cf. Section 3). To make that determination, we defined and used two informal effectiveness metrics:

- frequent comparison for consistency of the developing security architecture product with system artifacts that relate to the security architecture goals; and
- frequent integration of solicited feedback from the stakeholder security representatives, including the development, customer, and user communities.

Such related development artifacts are: the system architecture; the security needs (in the form of formal security requirements and accreditor guidance); and derived security requirements that are, in part, an outcome of the security architecture. Feedback was obtained by review of interim security architecture results by personnel within the DA organizations and discussions with them. On a less frequent but regular basis, review and working meetings were held that combined the Security Team and DA representatives.

4.4. Security architecture documentation

The goal of the security architecture documentation has been to be as useful as possible by:

- allowing customers and users to understand the security architecture as well as possible, in order to provide an effective basis for determining whether their needs are being met;
- providing security guidance to developers;
- recording decisions in a development environment that requires both strong direction and flexibility; and
- supporting accreditation by documenting security decisions and providing an overview of system security.

Based on feedback from the security architecture target community, it is apparent that this goal has been met in large part.

5. Results

This section presents the JSIMS security architecture as of September of this year. Because JSIMS is a joint program, and because a series of JSIMS releases must each be accredited for multiple security modes, the security architecture remains a living development artifact.

The security architecture is described here in terms of the *Base Logical Architecture* along with a set of *views* and a set of *extensions*. The Base Logical Architecture serves as the starting point for the architecture description. By presenting first this single, logical (i.e., abstracted) description of the security architecture, its important aspects can be simply explained and can lay the foundation for a kind of type accreditation. In that way, more concrete but varying aspects of the architecture, such as system releases and hardware layouts, can be compared with the characteristics of the logical architecture and so avoid a series of start-from-scratch accreditations.

Such comparison is necessary, since it is the implemented, deployed system that must ultimately be accredited. The comparison is given by a mapping between the logical architecture and other architecture representations, including any physical system to be deployed. Such mapping analysis is far from trivial, but is much simpler than would be a full architectural analysis for each JSIMS variation. Complexities of the mapping include its many-to-many nature and the need to demonstrate generic mappings, so that a class of deployments can at once be shown to conform to the logical view.

These aspects of the security architecture are explained in the following subsections, along with the explanation of their use in the joint, multiple accreditation environment. This section also documents feedback provided by the JSIMS stakeholders regarding the security architecture as developed.

5.1. The Base Logical Architecture

Figure 1 depicts the Base Logical Architecture, which includes top-level components, flows, and data

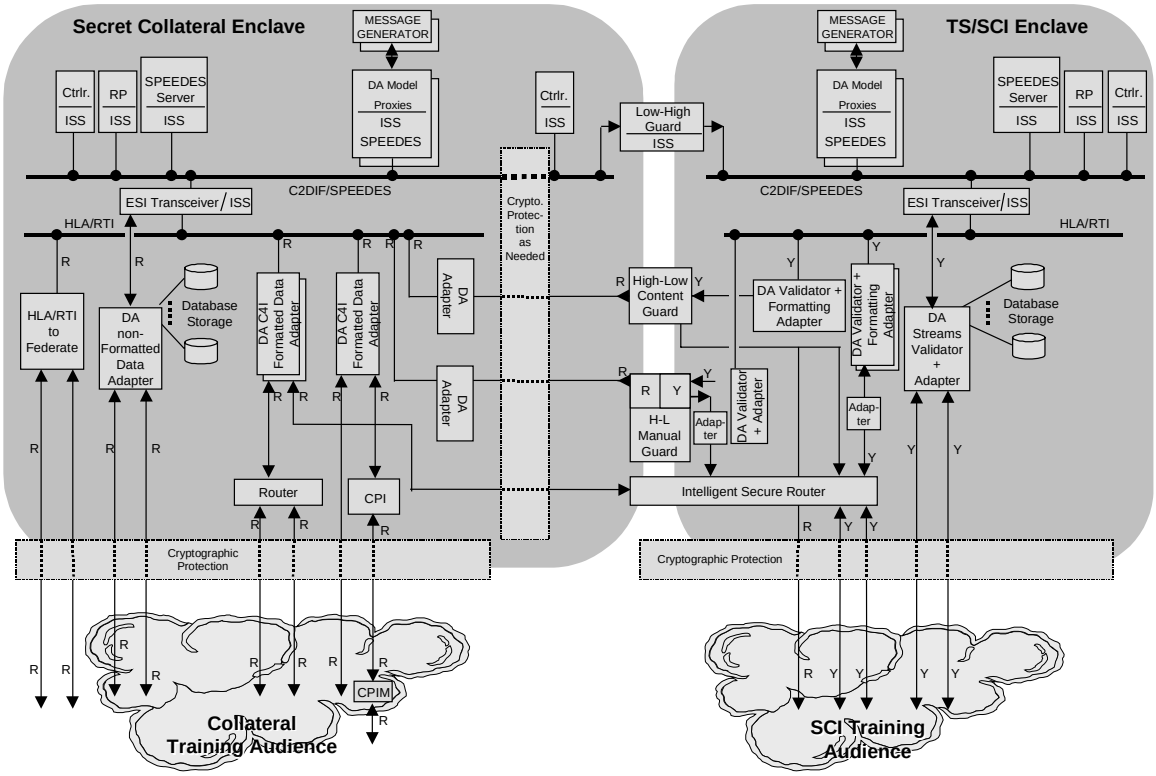

Figure 1. Base Logical Architecture

descriptions (content and format). As allocated security characteristics are applied to the architecture description, it becomes the logical *security* architecture. By "logical" it is meant that *classes* of components and data flows are shown. In that way, security requirements and relationships can be allocated and analyzed for accuracy and risk management purposes. At the same time, the volatility of particular deployments, compositions, and implementation decisions can be addressed by relating them to the logical architecture. In that way, the basic architectural analysis can be stable and accredited on a generic basis.

The two large gray areas in Figure 1 depict the two JSIMS enclaves, which in the Base Logical Architecture process Secret Collateral and TS/SCI information. The top parts of the enclaves, down to the C2DIF/SPEEDES backbone, represent the simulation (models, infrastructure, and simulation engine) part of JSIMS. The lower parts of the enclaves represent support for interfaces to external systems. The guards, which bridge between the enclaves, make up the inter-enclave controlled interface. For geographically remote data communications (enclave-to-enclave and JSIMS-to-external-systems), cryptographic protection is provided.

5.2. The views

The architecture *views* provide multiple ways to perceive the JSIMS security architecture. This base-architecture-plus-views approach is an important part of the support for a generic accreditation. Mappings are provided between the logical architecture and the other views, resulting in constraints on the views. Implementations that satisfy the view constraints maintain the legitimacy of the mappings.

The views of the JSIMS security architecture are depicted in Figure 2. Four of these views are shown as hierarchical layers. The fifth view, Organizational, underlies all the hierarchical views.

Details of the views. In the Abstract view, the distributed, multi-enclave nature of the JSIMS implementation is invisible, and the software infrastructure is seen by simulation models only in terms of the JSIMS API (JAPI). The purpose of the JAPI is to present to the simulation model applications a specification of the simulation mechanisms. The mechanism implementations appear within the Software view. The primary security characteristic of this view is a particularmechanism that supports, in the high

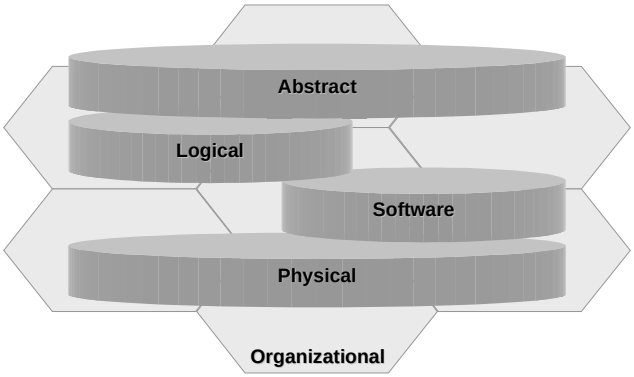

Figure 2. The Views of the JSIMS Security Architecture

enclave, the nomination of simulation objects to be sent to the low enclave via a high-assurance guard.

The Logical view, as indicated previously, describes data flows, including formats and content. One security characteristic of the Logical view consists of the controlled interfaces: guards supporting data flow between the enclaves and between JSIMS and external systems. The other security characteristic is support for integrity of data during transit through lower assurance components.

The Software view contains the operating system, middleware (e.g., database systems), simulation infrastructure mechanisms, and the simulation models. Security in this view deals with assurance needed for off-the-shelf components and developed software; and integrating sources of assurance, for example providing isolation of functions using multiple means.

In the Physical view, a "wiring diagram" of networks and platforms is provided. Security assurance is supplied within this view by hardware mechanisms exploited by software and by constraints on physical connectivity.

The final security architecture view is the Organizational view. This view is not part of the hierarchy, but relates to each of the other views. This view describes the organizational characteristics for both JSIMS development and operation. It provides a place to address management and procedural security controls. In order to manage security risk, such controls must be present and effective in order to complement the product-oriented technical controls. The accreditation process includes evaluation of these controls, so they must be visible in the architecture itself.

View mappings. The Abstract view—the JAPI specification—is implemented by the Software view. Each of the interface objects of the JAPI has been analyzed to determine whether applicable security requirements necessitated a high-assurance implementation. The relationship between the Abstract view and the Logical view has required analysis to determine the effect of security requirements associated with the Abstract view on the distributed nature of the implementation and on its two-enclave nature.

The Physical view is also directly related to both the Logical and Software views. The security characteristics of these related views depend on the characteristics of the hardware platforms and the connectivity established by the Physical view. One example is that the process isolation provided by operating systems is dependent on the correct operation of the hardware memory management scheme. Another example is that the capability for the security guards bridging the enclaves to provide partial isolation between the enclaves is dependent on the physical connectivity (particularly the lack of it) among the various platforms.

The Logical view is dependent on the Software view in that the software implementation controls the logical connectivity. Largely, the model applications specify the data flows, and the underlying software (operating system, simulation engine, and developed infrastructure) executes those specified flows. The control of connectivity takes on a specific security flavor at the low-to-high and high-to-low guards between the enclaves, where developed software takes on a support function for some of the guards.

The relationship between the Organizational view and the other views is one of organizational interdependence and communication in the form of Memoranda of Agreement (MOAs) and Memoranda of Understanding (MOUs). Such communication prevents expectations from "falling through the cracks."

5.3. The extensions

The Base Logical Architecture and related views describe one "point" in a multidimensional space of JSIMS variations. We term other points in the space *extensions* to the base architecture. As depicted in Figure 3, the dimensions are termed *Release*, *Composition Type*, and *Operational Phase*.

The architecture extensions provide specific support for the accreditation of multiple JSIMS versions and deployments by "reusing" parts of accreditations.

Table 1 describes several aspects of the JSIMS variation space. Each dimension of the space is given by a row in the table, with related extensions in the cells of that row. The extensions are positioned in the table so that one of the columns (shaded) shows where the Base Logical Architecture fits in the variation space. The points on each variation axis are emphasized with a dark border.

JSIMS variations represented by all combinations of extensions (i.e., all points in the space) will be realized and must be accreditable. Consequently, at least 12 ($2 \times 2 \times 3$) variations must be analyzed relative to security constraints

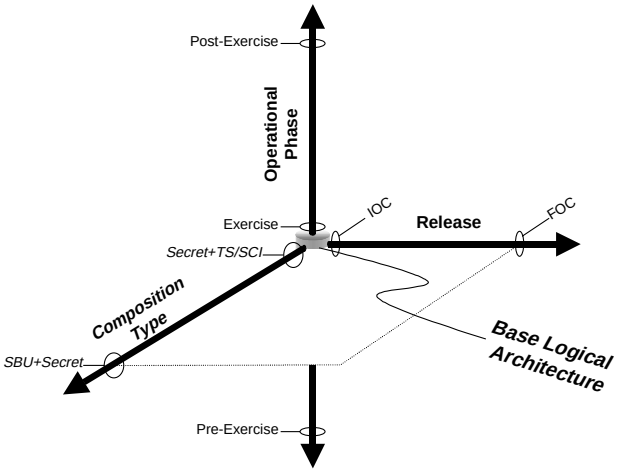

Figure 3. The JSIMS Security Architecture Variation Dimensions

and reuse of accreditation results. The phrase "at least" is appropriate because:

- JSIMS releases requiring accreditation, in addition to IOC and FOC, are possible; and
- a granularity of compositions finer than that of the two types given will be necessary.

Those finer-grained compositions are the result of the needs of the multiple DAs, which also constitute JSIMS user groups.

Table 1. Architecture Extension Descriptions

Dimensions	**Extensions with Axes Emphasized**		
		In Base Logical Architecture	
Release		At IOC	At FOC
Composition Type	Secret Collateral + TS/SCI	SBU + Secret Collateral	
Operational Phase	Pre-Exercise	Exercise	Post-Exercise

5.4. Effectiveness and use of the security architecture

The JSIMS security architecture is documented in part in the JSIMS Systems Security Authorization Agreement (SSAA) [6]. The DoD Information Technology Security Certification and Accreditation Process (DITSCAP) [2] (the accreditation process used within JSIMS) specifies the SSAA as the primary repository for accreditation support information. Releases of the SSAA are published on a limited-access JSIMS intranet site, making the security architecture is available to all JSIMS stakeholders.

Recall that the two informal metrics of the security architecture are (1) comparison and consistency with related system artifacts and (2) incorporation of feedback from the JSIMS stakeholders. In applying these metrics, it is important to recall that the development of the security architecture has not been unilateral: contributions were made by all the DAs and by the Security Team. Consequently, individuals involved in the security architecture development were often also those involved in providing feedback.

The two metrics are not entirely distinct. Comparison with artifacts has typically been accomplished by obtaining input from "owners" of the artifacts, who are generally JSIMS stakeholders. In particular, we facilitated accurate comparison with system artifacts (e.g., other system architecture representations, security requirements, and software design) by direct analysis of the artifacts with the assistance of the personnel responsible for them. Such personnel include the system architect, the Security Team member responsible for security requirements analysis, the software infrastructure design team, and DA security design teams. In that way we have been able to achieve a consensus of consistency between those artifacts and the security architecture.

Such inputs are obtained frequently via electronic communication and small group meetings, and less frequently in larger meetings, the latter to assure consistency of the inputs themselves. The outcome, as remarked over time, has been that the resulting security architecture continues to address the full complement of system and security concerns.

6. Conclusions

An accurate security architecture is closely related to the underlying system architecture. Ideally, it would be expected that the development of a security architecture would be based on an existing system architecture. What may happen instead (as in the case of JSIMS) is that well into a development effort important aspects of the system architecture are incomplete. The result is then that the system architecture as documented cannot serve as a basis for a security architecture. As a JSIMS example, until early this year, only the Abstract architecture and, to some degree, the Software architecture were documented.

As a result, the security architecture development, description, and dissemination often provide value to a system development far beyond its immediate target of addressing the structure of security constraints. That has been the case with JSIMS. By May 1999, the Security Team had developed the view-and-extension description of the security architecture and. In collaboration with all the DAs, the Security Team had also filled in much of the previously missing detail for particular views. That detail

included not only security attributes, but system architecture information as well.

The security architecture development method, with its small-team-plus-larger-group approach, proved to be as effective as hoped, judging from the results. The security architecture itself was effective as judged by the given informal metrics. In particular, as changes came up (such as new expectations provided by DAAs), it was possible to make necessary changes to architecture descriptions quickly and without great upheaval to the overall security architecture.

The Security Team has successfully used the varied-team approach for other security tasks within JSIMS, such as security requirements analysis. Based on the author's experience with other large projects in which security was a major component, the security architecture approach itself is applicable to a broad class of targets. The concept of the multiple views of the architecture has proved valuable, though the identity of the specific views could change for different systems.

Improvements to the process and the results could be made in two areas, one that was beyond our control, and one where we believe, with hindsight, that we could have improved the approach. The first area is in the "wish list" department—it is that more could have been done with less effort had it been possible to make a solid start with security in JSIMS at the beginning of the program, rather than many months into the program. Subsequent to significant system and software design and implementation, bringing the program under control with respect to security was difficult. This is a lesson that security engineers have learned many times over, but it is worth adding yet another data point.

Regarding the second area, we could surely have improved the approach by an early change in priorities. Once the security effort within JSIMS had begun, several tasks were quickly started, one of which was *not* security architecture development. As a result, the effort initially was not well coordinated, and the resulting lack of efficiency limited the initial impact of security on the overall program. Once the security architecture definition was begun, we were able to orient much of the security effort around that definition. After that point our progress notably accelerated.

7. Acknowledgements

Acknowledgements are due three JSIMS groups: the Extended Security Team (DA representatives plus core Security Team), the System Engineering Team, and the software developers. Discussions with those groups provided valuable input into, and in some cases were substantial to, the development of the JSIMS security architecture. Thanks are also due Phil Taylor for his careful reading of the paper.

8. References

1. *Common Criteria for Information Technology Security Evaluation*, version 2.0, May 1998.
2. *DoD Information Technology Security Certification and Accreditation Process (DITSCAP)*, November 1997.
3. Information Assurance Security Focus Group, *Security Architecture for the AITS Reference Architecture*, Revision 1.0, December 1997.
4. *JSIMS System Segment Description Document (SSDD)*, Overview and Discussion volume, version 2.5, January 1999.
5. *JSIMS System Segment Description Document (SSDD)*, version 3.0, Appendix A: JSIMS Application Program Interface Definition Document, April 1999.
6. *JSIMS Systems Security Authorization Agreement (SSAA)*, October 1999.