Defining & Employing a System of Systems COTS Strategy for Information Assurance

Presentation to SSTC: 21 June 2007

Robust Engineering Track

James W. Freeman, Ph.D., CISSP
Computer Technology Associates, Inc.
Colorado Springs, Colorado
jim.freeman@cta.com

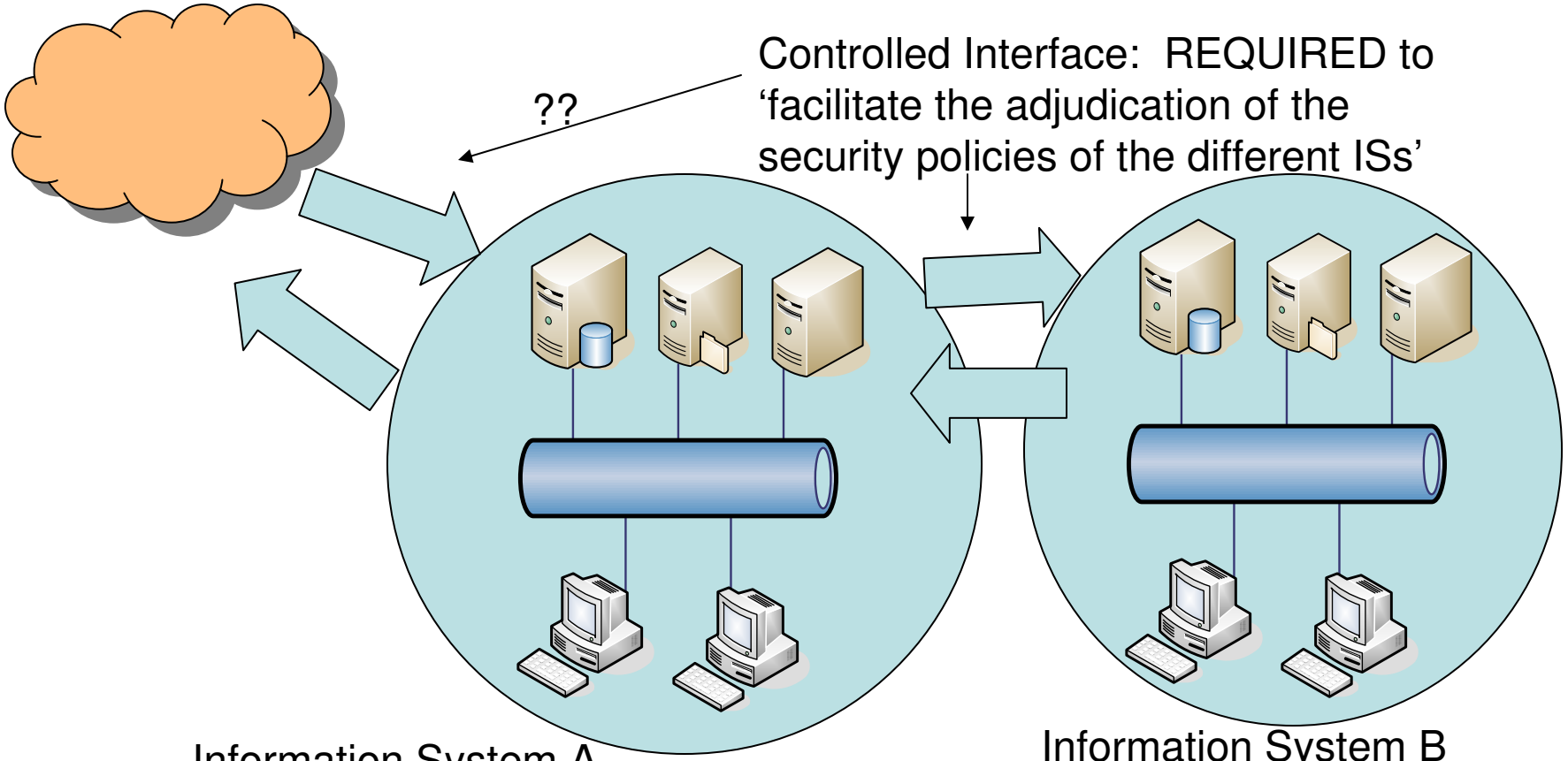
Message:
A Request of Systems Engineering Community
(regarding developing interoperable systems)

- Encourage
  - System *Security* Engineering to move from one of the 'ilities' to a 'more robust main stream system engineering'
- Kindly Demand
  - System *Security* Engineering Community to develop appropriate modeling frameworks and skills
    - to address integration of products
    - At least as good as 'standard system engineering' procedures and tool sets
- Examine Approach Outlined Here
  - Determine whether 'worthy of inclusion'

Topics

- Set-Up
  - Architecting an Interoperable System of Systems requiring a Controlled Interface (CI)
    - using a High Assurance Product
  - Review Threat (High-Grade)
  - The Problem
- Restating the Issue / Requirement
- Review Some Current Approaches
- Revisit the Message
  - What if we Ignore the Message?
- Elements of a Strategy
- Applying the Strategy
- Wrap Up & Some References

Set Up: Problem Domain



Processing Up to SECRET

@ PL 1 or PL 2

With Security Policy A

Processing Up to TOP SECRET / SAR

@ PL 3

With Security Policy B

Set-Up (cont): Some Controlled Interface Requirements (via JAFAN 6/3)

- A: Routing Information Shall be Supplied or Alterable *only by* the Security Support Structure of the Controlled Interface (CI)
- B: Each CI shall be configured to ensure that all (incoming and outgoing) communications protocols, services, and communications *Not Explicitly Permitted* are *Prohibited*.
- C: Each CI shall be tested to ensure that it satisfies all of the *appropriate* CI criteria

Reviewing the Threat: High Grade[1]

(as distinguished from 'hacker' or 'criminal')

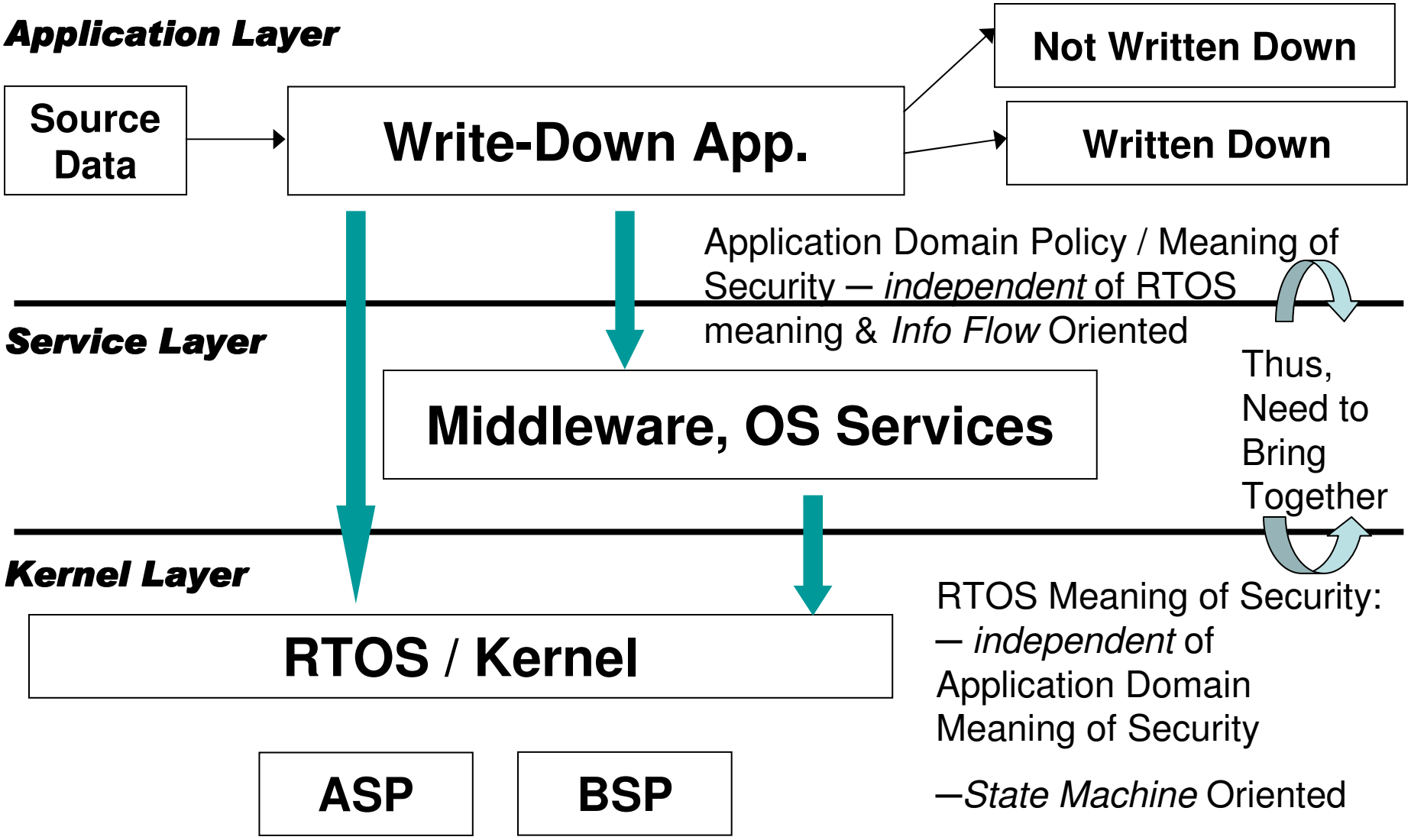
- Has extensive resources
  - money, personnel, technology
- Is patient and motivated
  - Full-time, organized with multidisciplinary staff, each of whom is eager to 'break' the system
- Capable of exploiting a successful attack for maximum long-term gain
  - Attacking team able to keep existence of a successful attack secret from the target
  - Adept in circumventing physical and procedural safeguards with access to clandestine technology
  - Deliberately seek the most obscure vulnerability hidden in darkest corner of the system — to permit maximum long-term exploitation

[1]*Computers at Risk, Safe Computing in the Information Age*, Appendix E, National Research Council, National Academy Press, 1991

The Problem

- How does one provide assurance (s) that
  - 'configuration of the CI will allow *only* the 'permitted protocols, services and communications' (Rqmt B)?
    - In light of Rqmt C, which could be viewed as an 'escape clause'?
- In Particular: Suppose system architecture, in particular, system security architecture based on
  - Use of a 'high assurance' product, e.g., a real-time operating system (RTOS)
  - Need to provide necessary assurances, per DITSCAP, that
    - Integration of COTS / GOTS or NDI software, hardware, and firmware
      - *Complies* with the system security architecture
      - *Integrity* of each product is maintained
    - Use of 'low assurance' products do NOT impact CI rqmts

Problem Refined: Write-Downs (Permitted or not)



Restating the Issue / Requirement

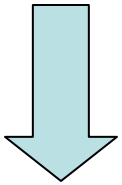
- Need to Resolve
  - ‘Meaning of Security’
    - Among various application domains and
    - The underlying ‘building blocks’
      - In particular when a ‘building block’ is a high assurance component
- Answer to Resolution
  - Rather Robust Policy Modeling with Support Tools
  - More Careful Integration of Policy Modeling Results into Development Process

Some Current Activities / Approaches

- Incorporation of *derived* functional rqmts and their integration done reasonably well
  - Based on industry standards: architecture specification, design, implementation with solid testing and configuration management
- Incorporation of assurance rqmts less well integrated
  - Mapped to 'current' industry standards and asserted that current practices are 'good enough'
  - Not as visible and traced (e.g., via DOORS)
  - Security Modeling Effort not necessarily clearly related to System and System Security Architecture
- Reactive versus Proactive
  - Penetrate and Patch as reactive
    - 'Patch Management' Appears to have become increasingly 'mainstream' (CrossTalk of Oct '05)
    - 'Six Dumbest Ideas in Computer Security' (Marcus Ranum, Sep '05)
  - Entire SSTC and 'CrossTalk as Proactive

Re-visiting the Message: What If we Ignore the Message

- Message: a request regarding the development of increasingly interoperable systems?
  - Encourage transitioning SSE into mainstream SE
  - Kindly demand SSE to develop better procedures, tool support
    - Examine / Assess Approach to be Presented Next
- 'Multics-30 Years Later' paper
  - Malicious Software
  - Auditing, Intrusion detection



- Resulting Interoperable Systems Having Information Security Risk Inadequately Mitigated

Elements of a Strategy

- Sub goal: *Maximize* use of COTS / GOTS / NDI and *Minimize* Program Developed capabilities
  - Not necessarily new
  - Develop better criteria for 'minimization' and 'maximization' relationships, with focus on associated information security risk
    - 'appears to be newer'
- Integrate System architecture and System security architecture views more clearly
- Incorporate Principles of a security modeling approach,
  - termed Boundary Flow Modeling, within that System / System Security Engineering Integration
- Develop an Ontology (Underlying / Overarching) of System and System Security
  - enable a more rigorous integration of 'distinct and disparate' models and their relationships
- Enhance Existing SE Tool Sets / Develop Supporting Tool Set (s)
- Integrate Resulting SSE Tool Sets Within SE Environment

What is a System Architecture?

- System (of Systems)
  - The composition of a collection of systems into a single integrated System of Systems that satisfies the contractual requirements laid down by a program office
- Architecture
  - An arrangement of design elements and collaborations between those elements that satisfies the customer's requirements
    - Functionality
    - Performance
    - 'Ilities'
- A System Architecture exists in the context of a larger enterprise
  - It's not an IT system
  - It's not a manufacturing system

Architecture Views and Their Uses

Operational View

- *Describes User Interactions*
- *Captures Operational Task Definitions*

Dynamics View

- *Operational Timing*
- *Functional Timing*

Functional View

- *Functional Partitioning*
- *Functional Sequencing*
- *Decision Logic*

Hardware View

- *Hardware Partitioning*
- *Hardware interfaces*

Behavioral View

- *Component Lifecycles*
- *Triggering Events*

Software & Information View

- *Software Partitioning*
- *Software Interfaces*
- *Functional Allocation to Software*
- *Information Content*
- *Information Partitioning*

Applying The Strategy

Security Architecture Description: One Way

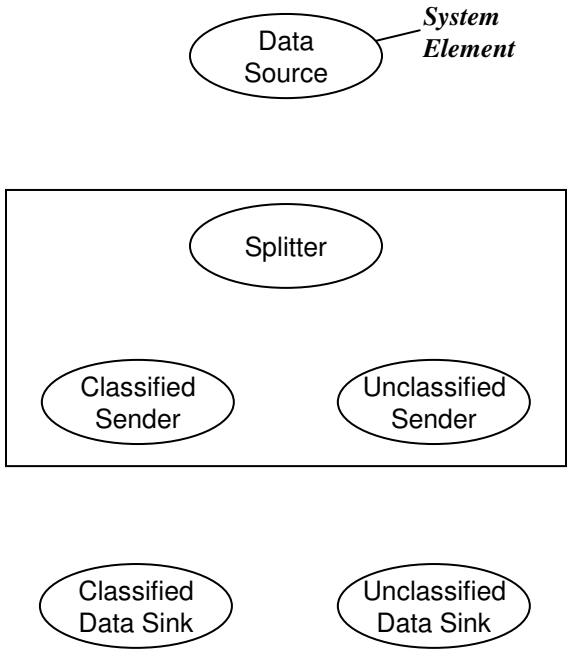
- A complete description of the system with a set of prescribed architecture diagrams
- Security Architecture Views:
  - Scenario View : a hierarchical set of functional views (e.g. UML activity diagrams) that describes security tasks & their sequences
  - Executable View : a hierarchical set of software views (e.g. UML class diagrams) that shows the software entities involved in the system security architecture.
    - This includes all supporting infrastructure and interfaces.
  - User View : a set of operational views (e.g., UML Use Case diagrams) that identifies security administrator roles. Connected to the Scenario View via the usual UML Use Case - Activity Diagram linkage.

Security Architecture Description: Continued

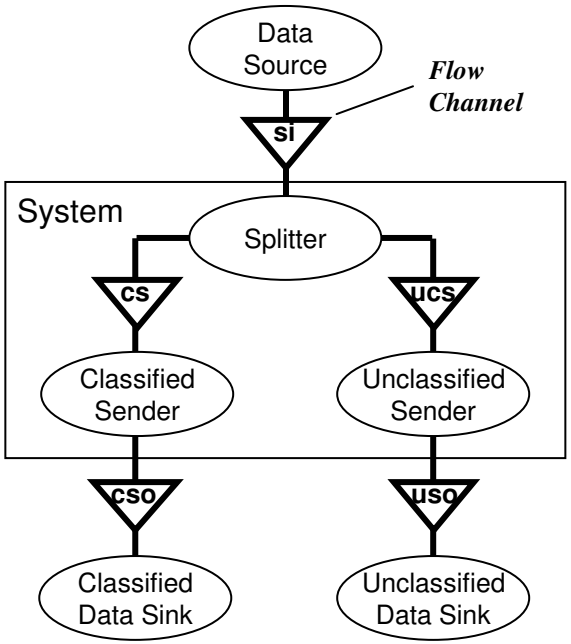
- Security Architecture Views (cont'd)
  - Data View - an information view (e.g., UML class diagram) that shows the location, partitioning and persistence of security related information relative to tiers (hierarchical)
  - Hardware View : a hardware view (e.g., UML class or deployment diagram) that defines the hardware entities involved with security, their interfaces and allocation of software elements to them.
- Most of such information is generated as part of standard design process
- Goal: Incorporate, as explicitly as possible, Security Relevant information within the existing and augmented views

Incorporating Structured Modeling Capabilities

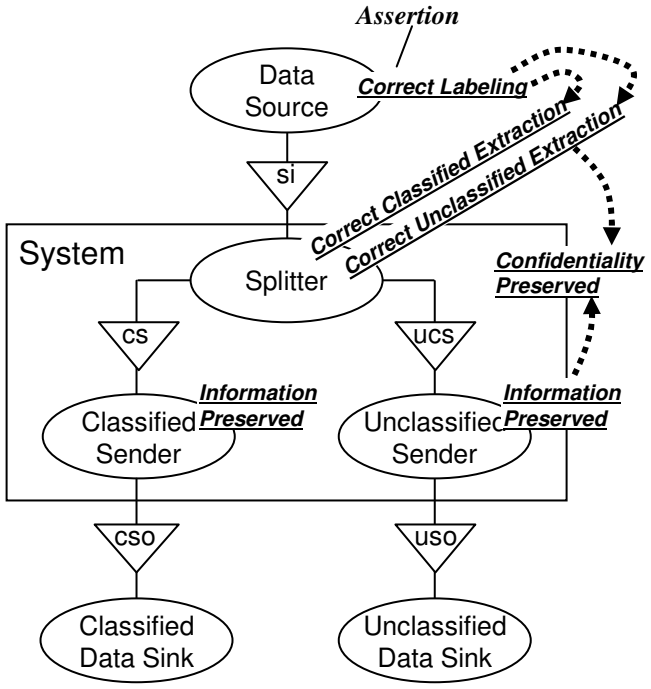
Security Architecture Structure Modeling...



...Enhanced by Information Flow...



...with Chain-of-Logic Assertion Dependencies



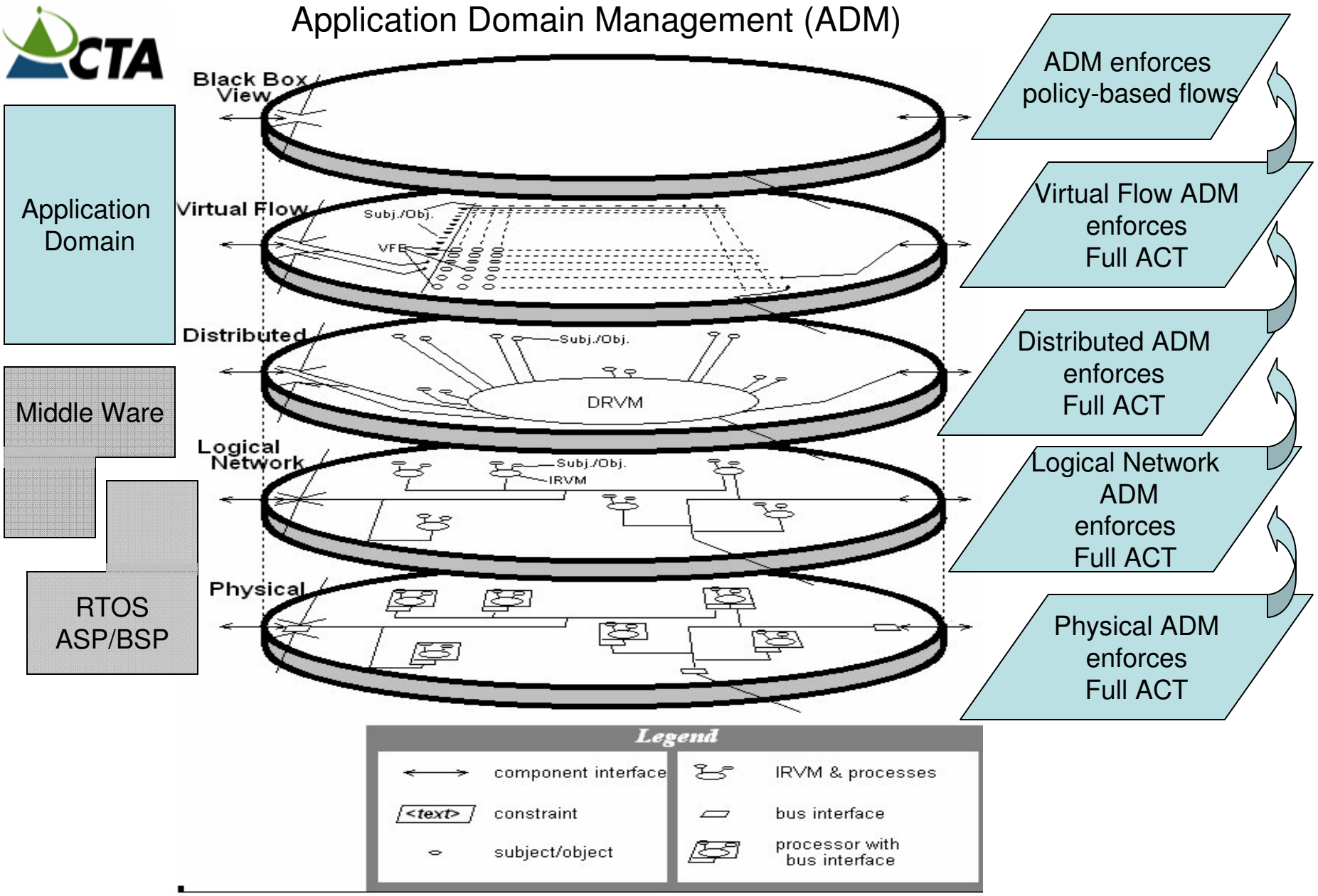
Legend:

si-source input

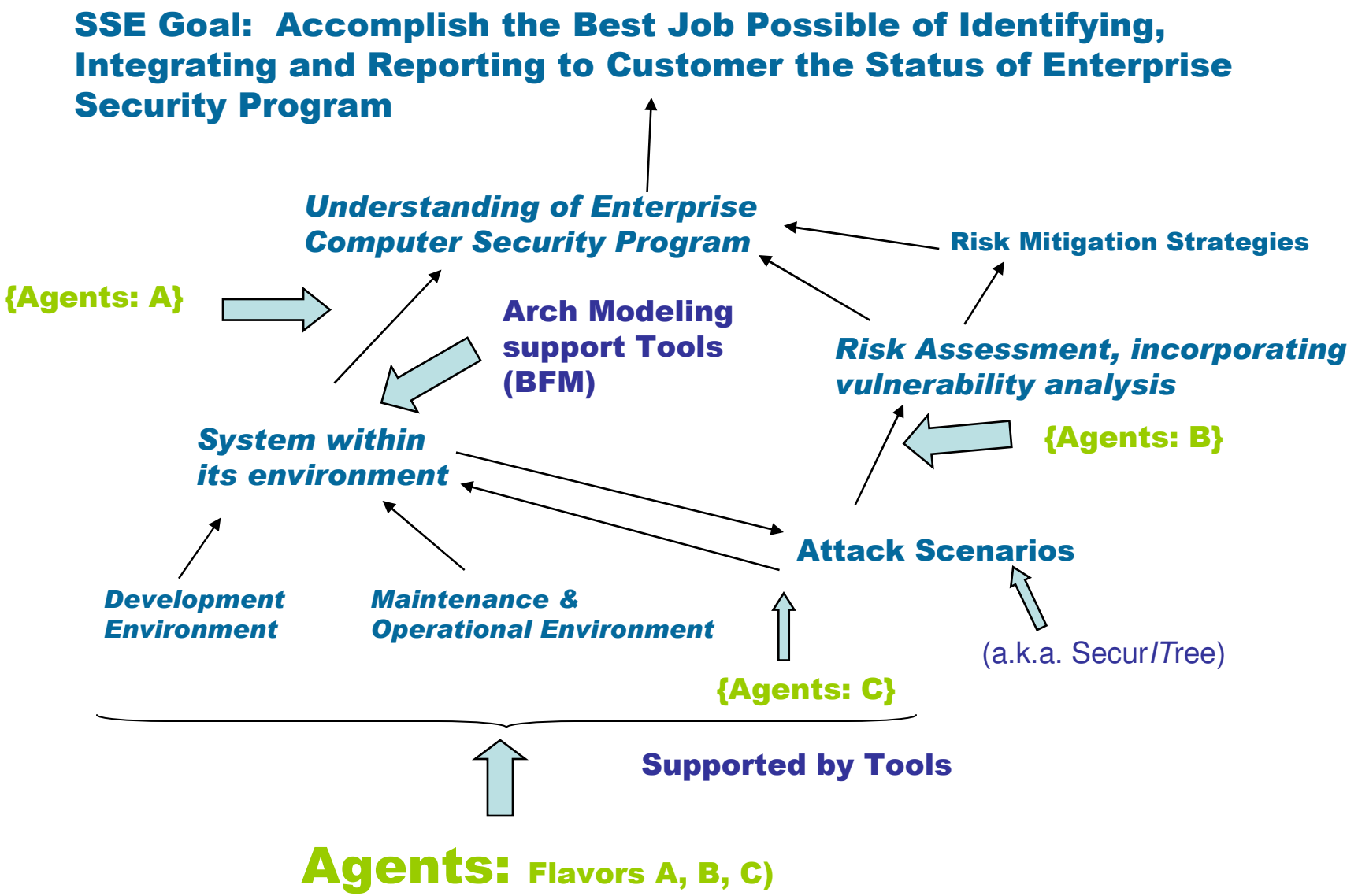
cs (ucs): to class (unclassified) sender

cso (uso): to class (unclassified) sink output

Application Domain Management (ADM)



Need for Developing an Ontology: Target



Wrap-Up

- Observation: Result of System *Security* Engineering (process and products) can be no better than the underlying *System* Engineering (process & products)
- Observation: Though we system security engineers construct, use, and analyze various models, such activity is most often implicit; we need to
  - do it more explicitly, and
  - ensure such activity is more readily integrated into system engineering process

Wrap-Up: Continued

- Request: (at least for Interoperable System of Systems)
  - Encourage SSE to become more 'mainstream' within SE,
  - Do so via integrating SSE activities and products within SE activities and products more directly as outlined
  - Encourage SSE to 'earn the right to sit at the SE Table'
- Suggested Metric for 'Success'
  - How Well Is the Job Being Done Relative to
    - Malicious Logic – as discussed in
      - 'Multics 30 Years Later' of P. Karger, R. Schell
      - 'Trusting Trust' of K. Thompson
    - Audit and Intrusion Detection – as discussed in
      - 'Multics 30 Years Later'

Some Definitions

- *Agent:* an entity authorized to act on another's behalf; *s/w agent* is program that acts to accomplish tasks on behalf of its user; *intelligent* agent attributes: goal-oriented, autonomous, collaborative, flexible, persistent, mobile.
- *Controlled Interface:* Mechanism to 'facilitate the adjudication of the security policies of different information systems'. (JAFAN 6/3)
- *Ontology:* basic structure around which a knowledge base can be built; addresses domain conceptualizations, free of technical requirements. (Edgington, Choi 04)
- *Protection Level:* Indication of the implicit level of trust that is placed in a system's technical capabilities. PL relative to clearance (s), formal access approval (s), and need-to-know of all direct and indirect users receiving information without manual intervention and reliable human review. (JAFAN 6/3)
- *SecurITree:* Tool developed by Amenaza Technology, Ltd that implements the modeling of attacks via a capability-based approach
- *Write-Down Program:* Program that produces information at a classification level less than that to which it has access.

References

- *Adopting Ontology to Facilitate Knowledge Sharing, Communications of ACM, Vol 47, #11, Nov 2004*
- *Reflections on Trusting Trust*, K. Thompson, Communications of the ACM, Vol 27, #8, (Turing Award Lecture) August 1984
- *Six Dumbest Ideas in Computer Security*, M. Ranum, certifiedsecuritypro.com, or via B. Schnier on Security, Sep 2005
- *Software Security: Shifting the Paradigm From Patch Management to Software Assurance*, J. Jarzombek, Special Issue: Ensuring Secure Software, Cross Talk, October 2005
- *Thirty Years Later: Lessons from the Multics Security Evaluation*, P. Karger, R. Schell, 18th Annual Computer Security Applications Conference, Las Vegas, NV, Dec 2002
- *A Validated Security Policy Modeling Approach*, J. Freeman, R. Neely, M. Heckard, 10th Annual Computer Security Applications Conference, Orlando, FL, Dec 1994

Acronyms

ACRONYM	DEFINITION
ACT	Access Control Table (generalized)
ADM	Application Domain Manager
ASP	Architecture Support Package
BFM	Boundary Flow Modeling
BSP	Board Support Package
CI	Controlled Interface
COTS / GOTS	Commercial (Government) Off-the Shelf,
CISSP	Certified Information Systems Security Professional
DITSCAP	DoD Information Technology Security Certification and Accreditation Process
DOORS	Dynamic Object Oriented Requirements System
DRVM	Distributed Reference Validation Mechanism (generalized)
IRVM	Individual Reference Validation Mechanism (generalized)
IS	Information System
JAFAN	Joint Air Force-Army-Navy
NDI	Non-Developmental Item
PL	Protection Level
RTOS	Real Time Operating System
SE	System Engineering
SSE	System Security Engineering