




# *Trust Enhanced Security*

Prof Vijay Varadharajan

Professor and Microsoft Chair in Computing, Macquarie University  
Director : Information and Networked System Security Research (INSS)  
([vijay@ics.mq.edu.au](mailto:vijay@ics.mq.edu.au))



# *Talk Overview*

- ❖ Context and Drivers
  - ❖ Technology and Socio-Geo Political Context
- ❖ Security and Trustworthy Computing
- ❖ Trusted Computing
  - ❖ Attestation Properties and Policies
  - ❖ Trusted Platform Enhanced Authorization
  - ❖ Web Services and Trust Negotiation
- ❖ Hybrid Trust Model
- ❖ Trust Enhanced Security
  - ❖ Mobile Agents
- ❖ Concluding Remarks

# *Technological Context*

## ❖ Technology Changes

### ❖ Rapid pace → ICT

- ❖ Speed and power of computing machines and software
- ❖ Interconnected Networks and Communications
- ❖ Information Explosion

### ❖ Enabling design of Large Scale Systems

- ❖ Interdisciplinary
- ❖ Interactive and Personalized/Customized
- ❖ Systems of Systems

→ To address complex problems and develop solutions

# *Socio and Geo-Political Context*

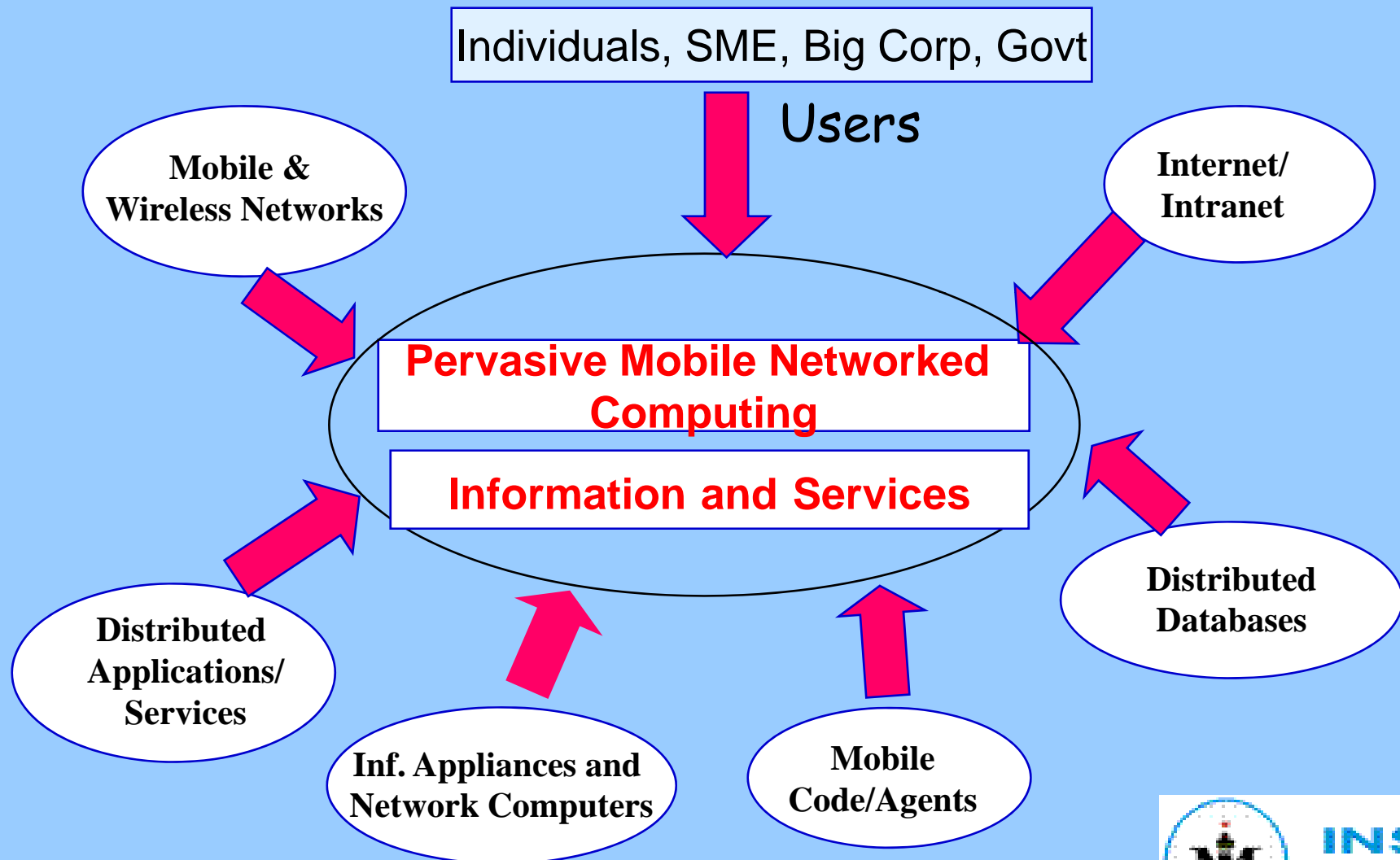
## ❖ Demographical Changes

- ❖ Growth increasingly concentrated in developing countries
  - ❖ Growth in younger generation
- ❖ Developed countries
  - ❖ Older generation
- ❖ Population who will be involved and affected by technology will be increasingly diverse and multidisciplinary

## → Implications

- ❖ on the nature of engineering and computing and the various opportunities for the design and application of solutions
- ❖ the need for greater collaboration and international engagement
- ❖ education in terms of management and communication skills

# Technology: Context and Drivers



# Several Security Challenges

- ❖ Scalable dynamic distributed systems and applications
  - ❖ How can a billion users access the same item at once?
- ❖ Dependability of Systems and Services
  - ❖ Availability, Security, Reliability of Information
- ❖ Management of Interactions over the Network
  - ❖ Managing Trust between Autonomous Unfamiliar Entities in the Provision of Services over the Internet
- ❖ Policies : Security, Trust, Privacy
  - ❖ Propagation, Administration and Enforcement of Policies
- ❖ Information Content Management
  - ❖ How to search, manage and extract useful information?
- ❖ Protection of Mobile Software over the Internet
  - ❖ Privilege Management for Mobile Software Code over an Untrusted Network
- ❖ Detection and Prevention of Denial of Service Attacks
- ❖ Seamless Secure Integration of Wired, Wireless and Mobile Infrastructures

# *Pervasiveness*

- ❖ Technology Pervasive
- ❖ Security Pervasive
  - ❖ Different Technologies and Multiple Platforms
    - ❖ Operating Systems, Networks, Middleware
    - ❖ Databases, Applications, Hardware Devices, Users
  - ❖ Different Business Segments
    - ❖ Healthcare, Finance, Telecommunications, Defence, Transport
  - ❖ Different Security Policies
    - ❖ Different Organizations
    - ❖ Different Industry Segments
  - ❖ Numerous Standards and Interoperability
- ❖ Some Consequences
  - ❖ Research : Different parts of the puzzle
  - ❖ Interconnections → Overall System
  - ❖ Organizational Challenges
- ❖ Facets of Security
  - ❖ People, Technology, Business/Org, Legal

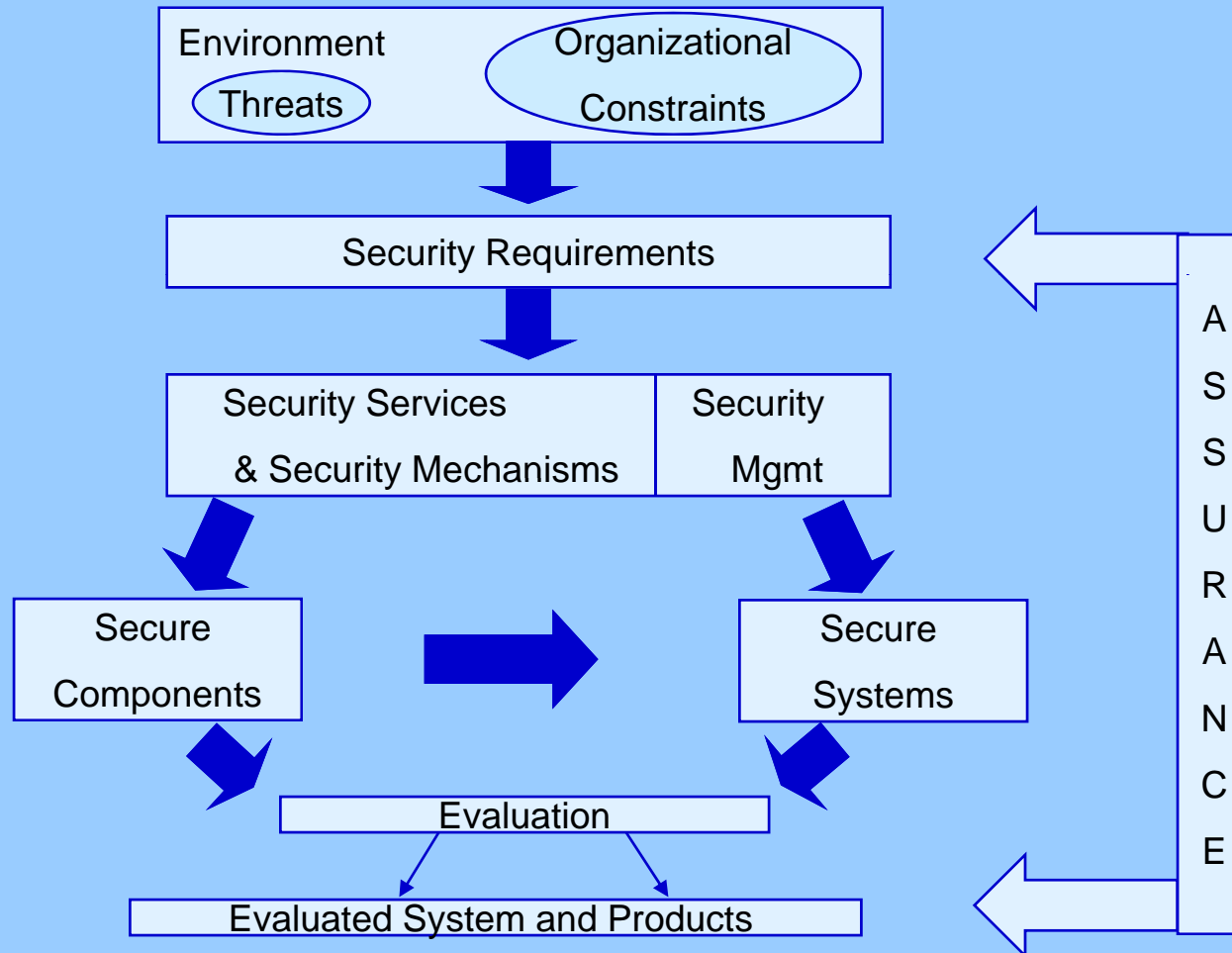
# Security

- ❖ Security
  - ❖ Relative to Threats
  - ❖ Cost, Time, Customer Expectations
  
- ❖ Security : Peace of Mind
- ❖ Security : A Business Necessity
- ❖ Security : Trust
  
- ❖ System Penetrator versus System Designer
- ❖ Code Breaker versus Code Maker

# Security and Privacy

- ❖ Security
  - ❖ *Owner* of Information has control
  - ❖ Security is Not Privacy
- ❖ Privacy
  - ❖ *Subject* of Information has control
  - ❖ Privacy requires Security
- ❖ Anonymity
  - ❖ Has no subject
  - ❖ Requires Security and guarantees Privacy, but is neither

# Security System Development



# *Security and Trust*

- ❖ Trust has been around for many decades (if not for centuries) in different disciplines in different disguises
  - ❖ Psychology, Philosophy, Sociology as well as in Technology
- ❖ Foundation of Security
- ❖ Some Notions
  - ❖ Luhman: “we as humans would not be able to face the complexity of the world without resorting to trust”
  - ❖ Gambetta: “trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends”
  - ❖ Trust : “It will not harm me”, “No Surprises”
  - ❖ Trust : From a malicious point of view

# *Security and Trust*

- Trust
  - Entities with free will (type A) and without free will (type B)
  - Trust in type A is the belief that it will behave without malicious intent
  - Trust in type B is the belief that it will resist malicious manipulation by a type A entity.
- Computing
  - Software agents and computing machines can be thought of as representatives of the human owner(s)
  - If so, special case → focus on type B trust
  - Develop models and mechanisms for reasoning about resisting manipulation by type A -- detecting and preventing such manipulations.

# *Security and Trust*

- Trust Relationship
  - Trustor : an entity that trusts another entity (target)
  - Trustee : an entity that is trusted
  - Action
  - Context
- Trust Relationship is a belief by a trustor on the trustee's actions
  - Competency : Ability
  - Honesty : Intentions
  - Reliability : Correctness and commitments
  - Availability : Resources
- within a context

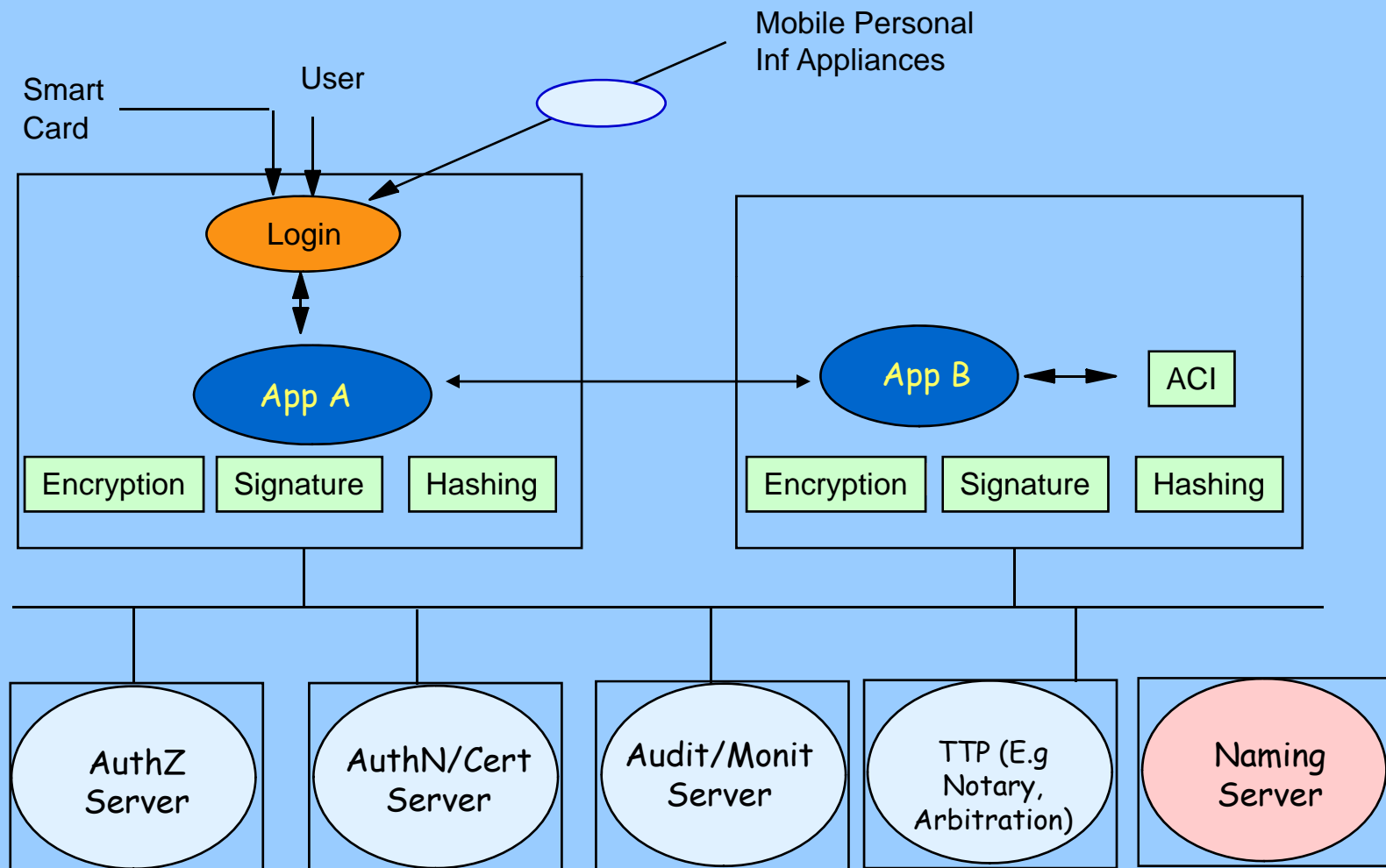
# *Security and Trust*

- ❖ Several Characteristics
  - ◆ Transitivity
    - General
    - Within a Context
  - ◆ Action-Dependent
  - ◆ Time-Dependent
    - Non Monotonic
  - ◆ Trust Building, Trust Destroying
  - ◆ Trusted Authorities
    - Multiple

# Trusted Systems

- ❖ Trusted Computer System Evaluation Criteria (TCSEC) (Orange Book) in the late 1970s and early 1980s
  - ❖ Trust → Process of convincing the observers that a system (model, design or implementation) is correct and secure
    - ❖ Set of ratings is defined for classification of systems
      - ❖ Higher the level, greater the assurance that one has that the system will behave according to its specifications → higher level of “trust”
  - ❖ Trusted Computing Base (TCB)
    - ❖ “totality of protection mechanisms needed to enforce the security policy”
      - ❖ Hardware and Software
  - ❖ “Trusted” Processes
    - ❖ These processes are trusted in that they will not do any harm even though they may violate the security policies of the system

# Security and Trust in Distributed Systems

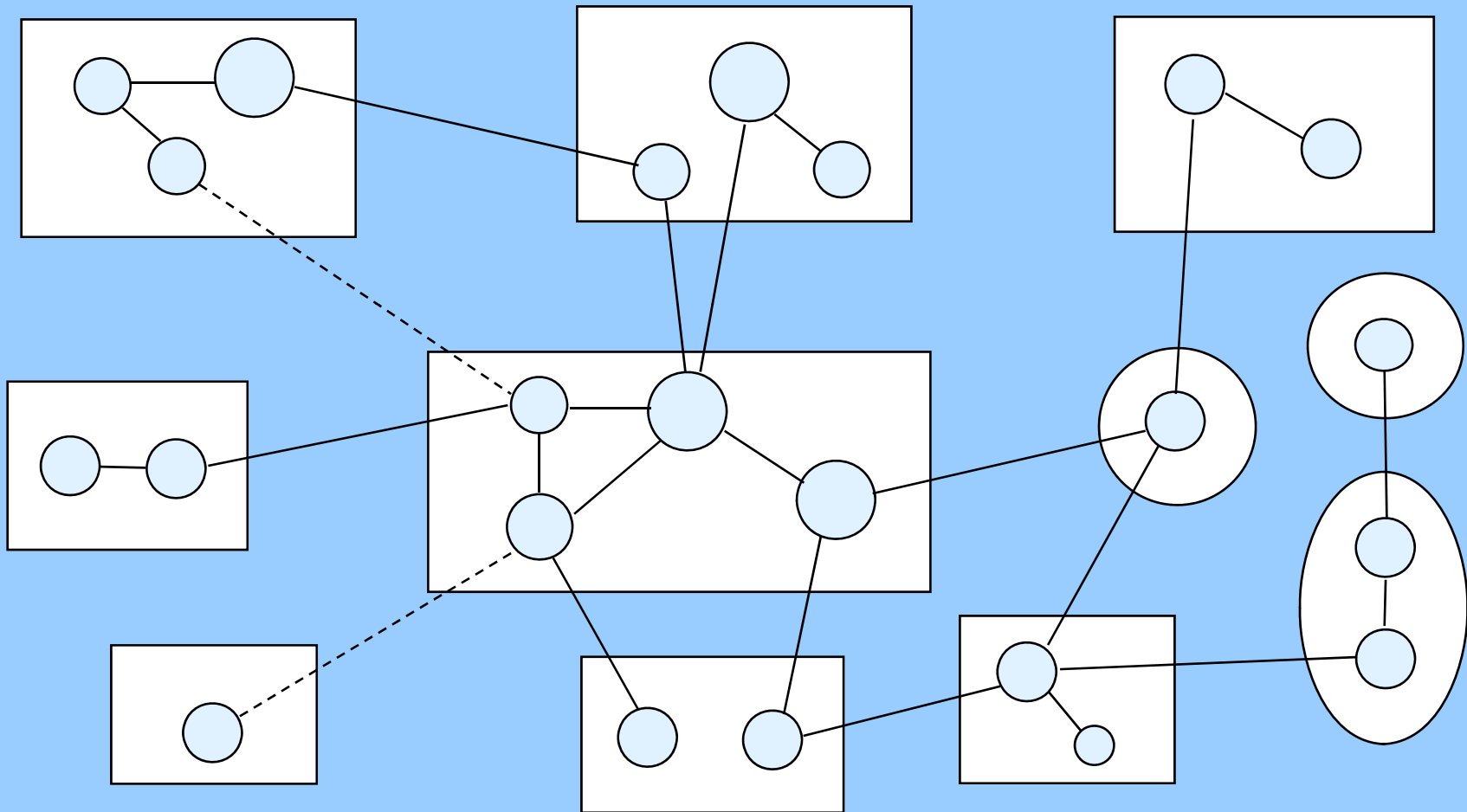


# *Security and Trust in Distributed Systems*

## ❖ Some Examples of Trust

- ❖ Trustor “trusts” a trustee entity to provide a service
- ❖ Trustor “trusts” a trustee entity to access and use the resources s/he owns or controls (e.g. application or service)
- ❖ Trustor (e.g. a user) “trusts” a trustee entity (e.g. CA/AS) to perform authentication and certification of another entity (Authentication Trust)
- ❖ Trustor (e.g. a user) “trusts” a trustee entity (e.g. ACS) to perform authorization actions (Authorization Trust)
- ❖ Trustor “trusts” a trustee entity to make a delegation on its behalf (Delegation Trust)
- ❖ Trustor (e.g. a user) “trusts” a trustee entity (e.g. network) to provide certain services (Infrastructure Trust)

# *Trust in Federated Distributed Environment*



# *Trusted Computing Platforms*

- ❖ A Trusted Computing Platform
  - ❖ has a trusted component (s) in the form of built-in hardware and uses this to create a foundation of trust for software processes
  - ❖ PC, Server, PDA, Printer, Mobile Phone
  - ❖ “Trusted” by local and remote users and software and entities
- ❖ Basis of Trust: Declaration on
  - ❖ the computing platform behaves as expected
  - ❖ the software running on a machine behaves as expected
  - ❖ what entity and to whom the user is talking to
  - ❖ the information is transmitted accurately and its privacy protected

# Trusted Computing Platforms

- ❖ TCPA/TCG view of Trust
  - ❖ Something is trusted “if it always behaves in the expected manner for the intended purpose”
- ❖ TCPA/TCG: Vouches for the State of the Machine
  - ❖ Whether a platform *can* be trusted?
    - ❖ Collect and provide evidence of system behaviour
  - ❖ Whether a platform *should* be trusted?
    - ❖ Provide confidence on the collection and evidence mechanisms
    - ❖ Provide confidence that particular values of evidence represent that the platform is in a “good” state”

# Trusted Computing Platforms

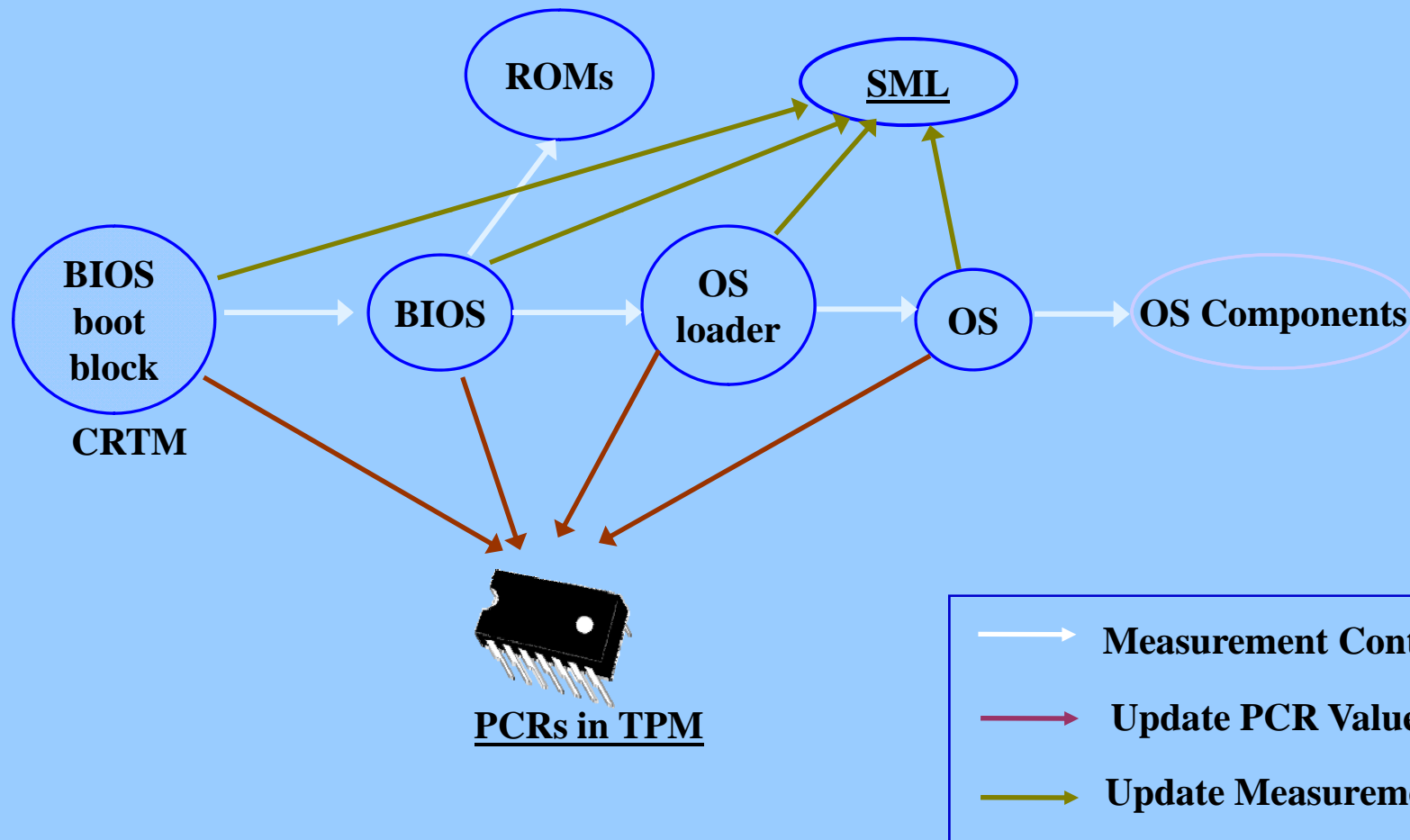
## ❖ Basic Idea

- ❖ A trusted party assesses the platform and declares that if the measurements for the platform are such and such, it can be trusted for such and such purpose.
- ❖ Measurement Process
- ❖ Storage and Reporting of measurements
- ❖ Matching with standard expected values

## ❖ PC

- ❖ BIOS Boot Block starts the measurements and stores the results in Trusted Platform Module (TPM) – tamper resistant chip. This is compared with the expected values
- ❖ This happens for all loading of software and before their execution
  - ❖ BIOS → OS Loader → OS Kernel → Applications

# Chain of Trust



# *Attestation*

## ❖ Integrity Report

### ❖ State of the Platform

- ❖ Requested final PCR values
- ❖ Storage Measurement Log (SML) values
- ❖ Component Validity Certificates (CVCs)
  - ❖ Maps ID of components to their measurements and signed

# Attestation

- ❖ Attestation
  - ❖ Challenger and Attester – a la Challenge-Response
- ❖ Integrity Report Verification
  - ❖ Challenger receives the Integrity Report
  - ❖ Recomputes the PCR values using the SML information
    - ❖ Component by component in the same order
  - ❖ As it recomputes, checks if individual values validate against component validity certificate values.
    - ❖ Check if value in SML = value in certificate
    - ❖ Check if signature of certificate valid
  - ❖ Matches computed PCR values with sent PCR values
  - ❖ If values match, confirms platform is in 'good' state
  - ❖ If not, everything beyond mismatch point 'untrustworthy' as chain of trust is broken

# *Trusted Computing Platform*

- ❖ A PC X booted into a known state with an approved combination of hardware and software (e.g. whose licences have not expired).
- ❖ Now TPM can certify to another party Y about the state of the PC.
  - ❖ E.g. certifying that the PC is currently running an authorised software
- ❖ The other party Y can now make its trust and authorization decisions – using this info.
- ❖ The two parties can now have secure information transfer with the platform -- information protected with a key which is in turn protected by TPM key.
- ❖ TPM releases the appropriate key to the authorised software X.

## *Some Issues with Measurements*

- ❖ PCR values
  - ❖ Many components
  - ❖ Many revisions for each component
  - ❖ Fixed set of PCRs
  - ❖ Different ways of concatenation
- ❖ Binary measurements (hash values) of components change for updates/patches
  - ❖ Static States → Runtime Issues
- ❖ Verifier interested in Policies → Policy expression difficult with binary values
  - ❖ May not be possible for the challenger to guess all possible states/values
  - ❖ Meaningful policy expressions : authorization, trust?
- ❖ Privacy
  - ❖ Challenger could learn exact implementation details/vulnerabilities

# *Properties for Attestation*

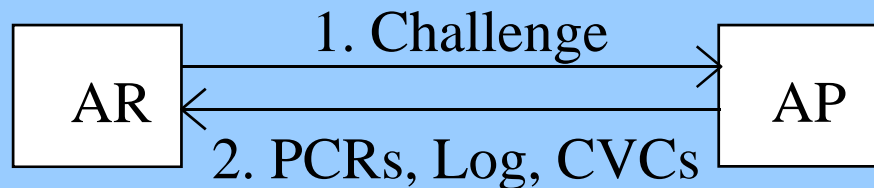
- ❖ Mapping Binary Values to Properties
- ❖ Conceptually
  - ❖ Properties : Higher Level Abstraction
  - ❖ Many States to one Property mapping
  - ❖ States may change but relevant properties should not
- ❖ Types of Properties
  - ❖ Presence of certain security characteristics
  - ❖ Absence of vulnerabilities
  - ❖ Certification Standards such as ISO
- ❖ Properties
  - ❖ Platforms
  - ❖ Components

# *Web Services and Attestation*

- ❖ Web Services Key Players
  - ❖ Attestation Requester AR
    - ❖ Service Provider or Service Requestor
  - ❖ Attesting Platform AP
    - ❖ Service Provider or Service Requestor
  - ❖ Validation Service VS
    - ❖ Performs integrity verification on behalf of AR
    - ❖ Less computation for AR
    - ❖ More privacy for AP
- ❖ Web Service Attestation
  - ❖ Different Architectural Models
    - ❖ Direct, Push, Pull and Delegation
  - ❖ Messages for Integrity Request, Report and Verification

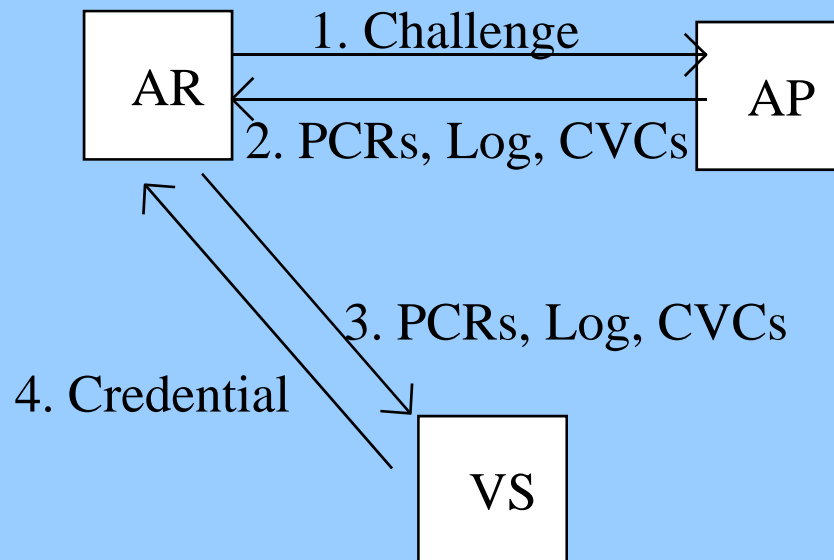
## *Direct Attestation*

- ❖ Msg1: AR challenges AP for its state
- ❖ Msg2: AP sends its state information to AR
- ❖ AR performs verification of AP's state



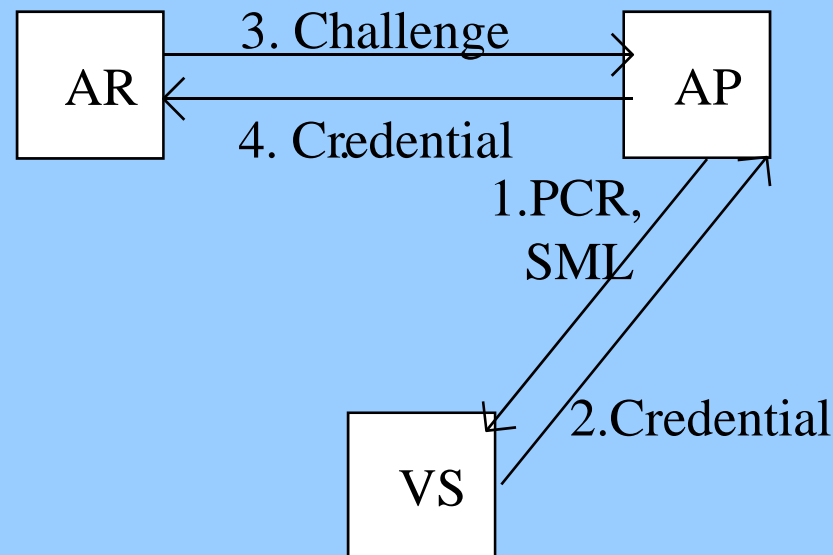
## *Pull Model Attestation*

- ❖ Msg1: AR challenges AP for its state
- ❖ Msg2: AP sends state info to AR
- ❖ Msg3: AR requests VS to validate state info on its behalf
- ❖ Msg4: VS validates state info and creates a credential
  - ❖ AR should wait until credential is generated



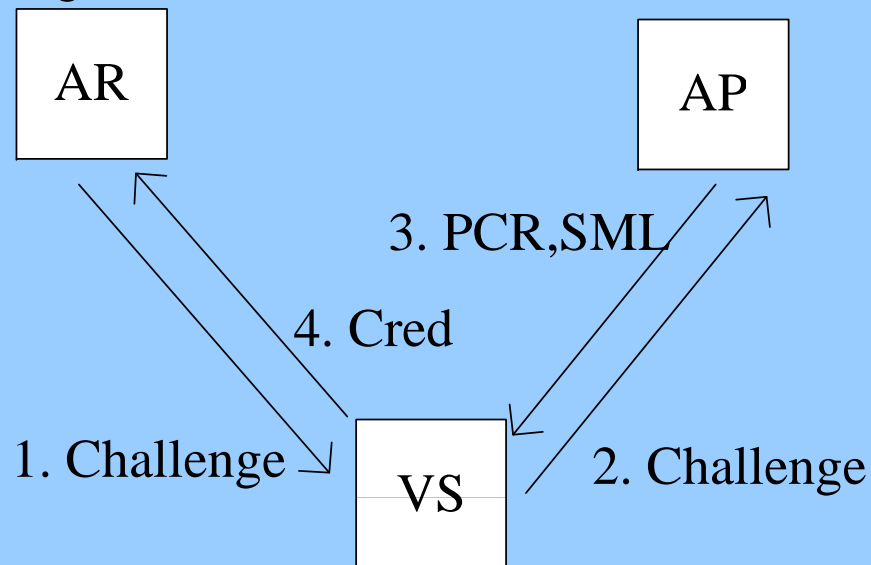
## *Push Model Attestation*

- ❖ Msg 1,2: AP obtains a credential in advance
- ❖ Msg 3: AR presents a challenge to AP
- ❖ Msg 4: AP sends the credential previously obtained to AR
  - ❖ Credential can be re-used for other challenges from different ARs
  - ❖ AR must trust VS that AP chooses
  - ❖ Freshness of credential not guaranteed



## *Delegation Model Attestation*

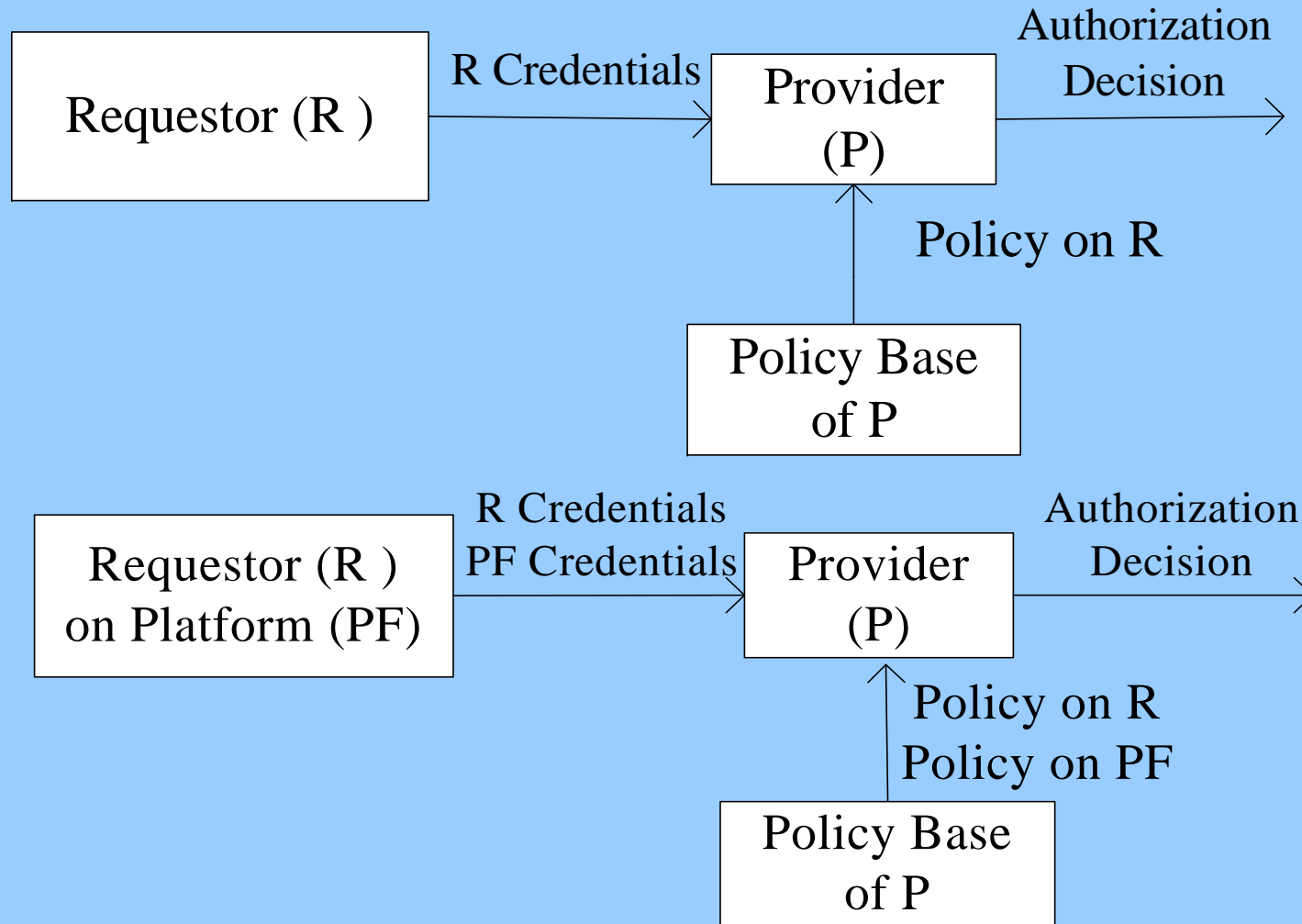
- ❖ Msg1: AR delegates VS on its behalf to perform integrity request and verification
- ❖ Msg2,3: VS requests the state from AR and generates the credential
- ❖ Msg4: Sends credential to AR
  - ❖ AP must trust VS that AR chooses
  - ❖ AR waits longer to receive the credential



## *Properties in Web Service Attestation*

- ❖ Combine Properties and WS-Attestation
- ❖ VS issues Attestation Credentials
  - ❖ Property based
    - ❖ At present, simple properties
      - ❖ “Integrity verification = true”
    - ❖ Property based Certificates
      - ❖ Policies

# Trusted Platform enhanced Authorization



# *Trusted Platform enhanced Authorization*

## ❖ Requirements

### ❖ User Credentials

- ❖ User Authentication Credentials
- ❖ User Authorization Credentials

### ❖ Platform Credentials

- ❖ Authentication– Presently only Attestation Credential possible
- ❖ Authorization– Property Based Certificates

### ❖ Policy Language

- ❖ Extensions to existing languages required

### ❖ Evaluation Algorithm

### ❖ Provision of Negotiation and Protocols

# Properties

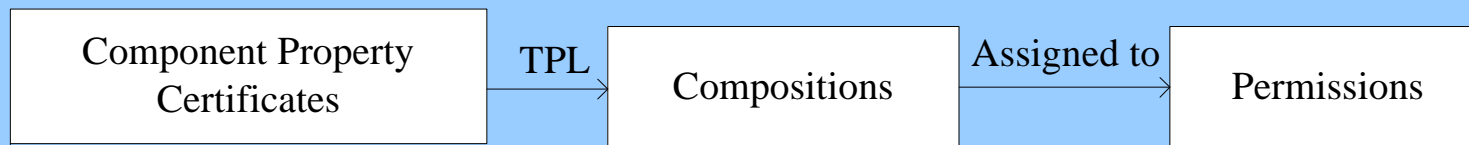
- ❖ Different Granularities
  - ❖ High (S1): Platform level properties
    - ❖ Abstract properties that are at platform level
      - ❖ Platform secure, state verified etc.
    - ❖ More Privacy for Attestation Platform
    - ❖ Less Flexibility for Attestation Requestor
  - ❖ Low (S3): Component specific properties
    - ❖ Properties of individual components
    - ❖ Very fine granular
    - ❖ Less Privacy for Attestation Platform
    - ❖ More flexibility for Attestation Requestor
  - ❖ Mid (S2): A combination of S1 and S3
    - ❖ S2 true if one or more S3 properties are satisfied
    - ❖ Can be associated with labels e.g high, low
    - ❖ Both Privacy for Attestation Platform and Flexibility for Attestation Requestor

## *Property Based Certificates*

- ❖ Includes properties of different granularities
- ❖ <Component ID, Expiry date, Property Set, Signature>
  - ❖ Property set includes S1, S2 and S3
  - ❖ Signed by property validation authority, not necessarily the manufacturer
  - ❖ Can also include state information

# Property Based Policies

- ❖ Policy Base
  - ❖ Policy statements capturing properties of many components
  - ❖ S1 and/or S2 and/or S3 for each component
- ❖ Compose different properties of different components to form '*Compositions*'
  - ❖ e.g. using composition language like TPL
- ❖ Assign compositions to authorization permissions



## *Trust Negotiation*

- ❖ AP requests AR for service & provides its property certificates
- ❖ AR verifies property certificates against its policy base for the requested service
  - ❖ Should have the ability to request for more properties if required
  - ❖ AP and AR can agree on iterative disclosure of properties in increasing order of granularity

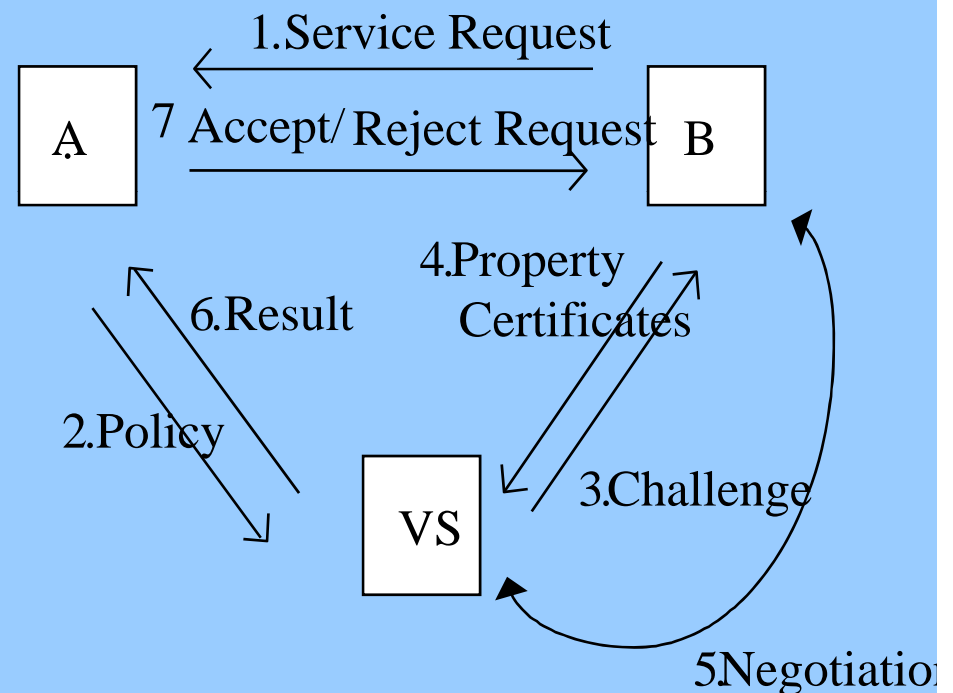


# *Trust Negotiation*

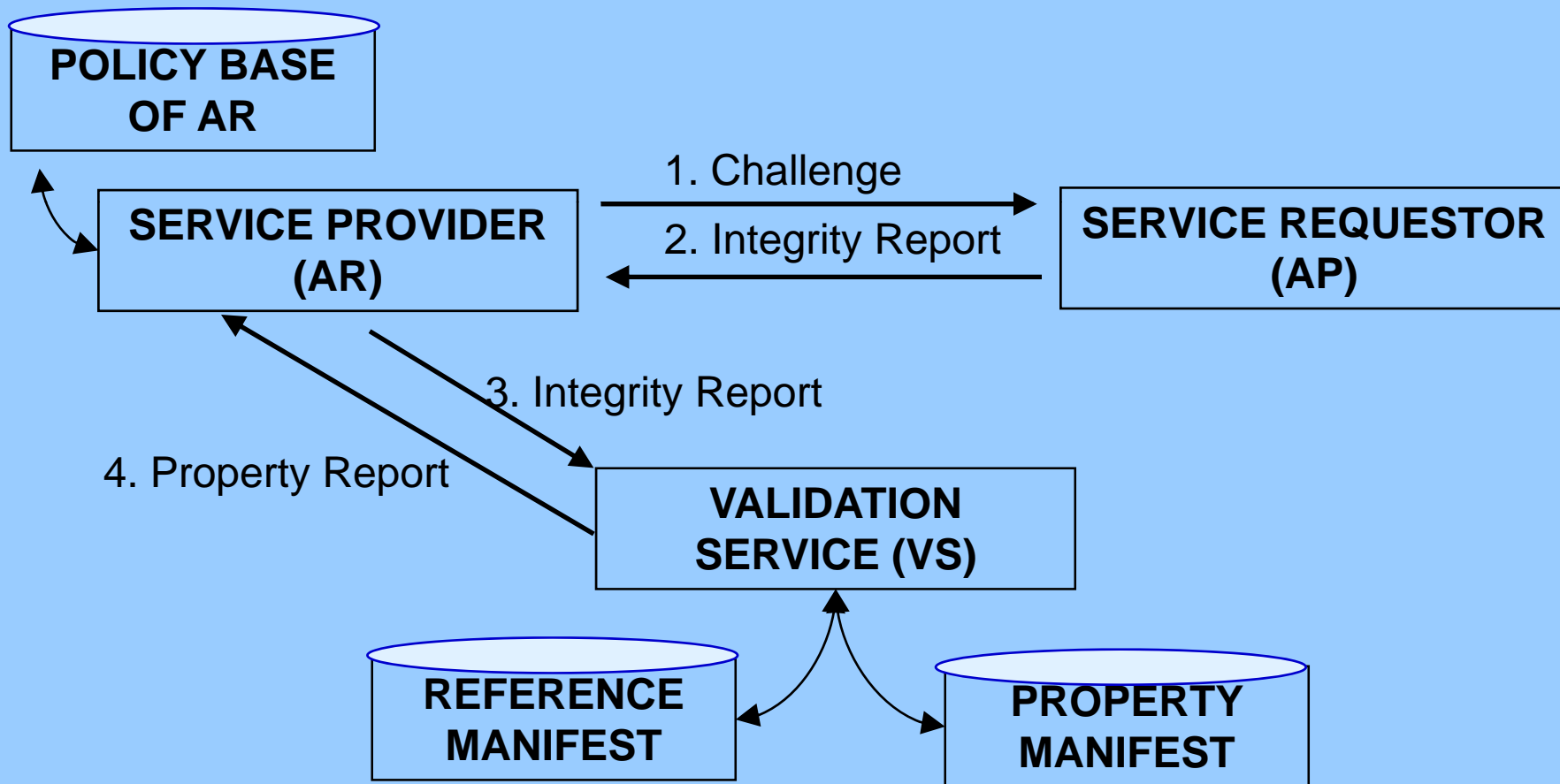
- ❖ Negotiation Design Choices
  - ❖ Type of policies negotiated – S1, S2 or S3
  - ❖ Privacy Policies of AP (c.f AP's State)
  - ❖ Authorization Policies of AR
  - ❖ Trust on VS by AP and AR
  - ❖ Trust between AP and AR
  - ❖ Choice of Push, Pull or Delegation Models
  - ❖ Computational Capabilities of AR

## Example Scenario

- ❖ Delegated Model
- ❖ A delegates VS for state verification and negotiation
  - ❖ A's policy available to VS
- ❖ Other design choices
  - ❖ A only delegates verification and it performs negotiation
  - ❖ A provides part of the policy and performs part of the negotiation



# Implementation Overview – Sample Pull Model



# *Property based Specifications*

- ❖ Measure and attest the relevant security state needed for security policies → e.g. objects and processes that need to be guaranteed for the security policies
- ❖ Check whether the target system integrity matches with the required security policies
  - ❖ Examples
    - ❖ Information Flow Policies
    - ❖ Role based Access Model
    - ❖ Clark Wilson Model (Sailer et al)
- ❖ Timing Issues
  - ❖ Loading time versus Runtime Issues
- ❖ Update Issues
  - ❖ Updates and Patches Issues

# *Property based Specifications*

- ❖ **Finer Code based Properties**
  - ❖ Attesting specific parts of programs
  - ❖ Introduce Begin-End Block at inserted in binary code by Developers
  - ❖ When block executed, State Information used in the authentication
  - ❖ Secure Kernel needed to protect the attesting program BIND (Shi, Perrig et al)
- ❖ Aiming to minimize the gap between the time of use and time of attestation (attesting just before execution)

# *Aspects for Property based Specifications*

- ❖ What to measure
- ❖ How to measure
- ❖ Time of Measurement versus Time of Verification
- ❖ Protocol for Transfer
- ❖ Policies for Decision and Enforcement
- ❖ Mechanisms for Enforcement

# Challenges with Property Specification

- ❖ What properties are useful and relevant and how to define them?
  - ❖ Security Properties
    - ❖ Will do something (e.g. will provide privacy)
    - ❖ Will not do something (e.g. will not leak information)
  - ❖ Non-security properties
    - ❖ Correctness, Reliability
  - ❖ Use state attestation to reflect configuration information
- ❖ Semantics of Properties
  - ❖ Requirements → Properties
  - ❖ Composition of Properties
  - ❖ Translation between Properties
    - ❖ Different Domains: Organizations, Applications
  - ❖ Dynamic Nature of Properties
    - ❖ Temporal (Measurement time vs Query/Verification time)

# *“Hard” and “Soft” Trust*

## ❖ Hard Trust

- ❖ Trust beliefs derived from concrete security mechanisms
- ❖ E.g. keys and certificates signed by certificate authorities binding the keys to an entity
- ❖ Characterized by “certainty”

## ❖ Soft Trust

- ❖ Trust derived from social control mechanisms and intangible information such as reputation, experiences and cooperation
- ❖ Beliefs not based on concrete security credentials such as keys
- ❖ Characterized by “uncertainty”
- ❖ Dependent on past behaviours
- ❖ Often involves recommendations from multiple entities (“web of trust”)
- ❖ Progressively tune the beliefs over time

# *Hybrid Trust Enhanced Security*

## ❖ Hybrid Trust

- ❖ Combining “Hard” and “Soft” Trust
- ❖ Model more effectively the dynamic changes in trust that arise due to changes in behaviour of users and applications

## ❖ Hybrid Trust

- Improved Secure Decision Making and Optimizing Risk
- Maximizing Utility and Business Benefits

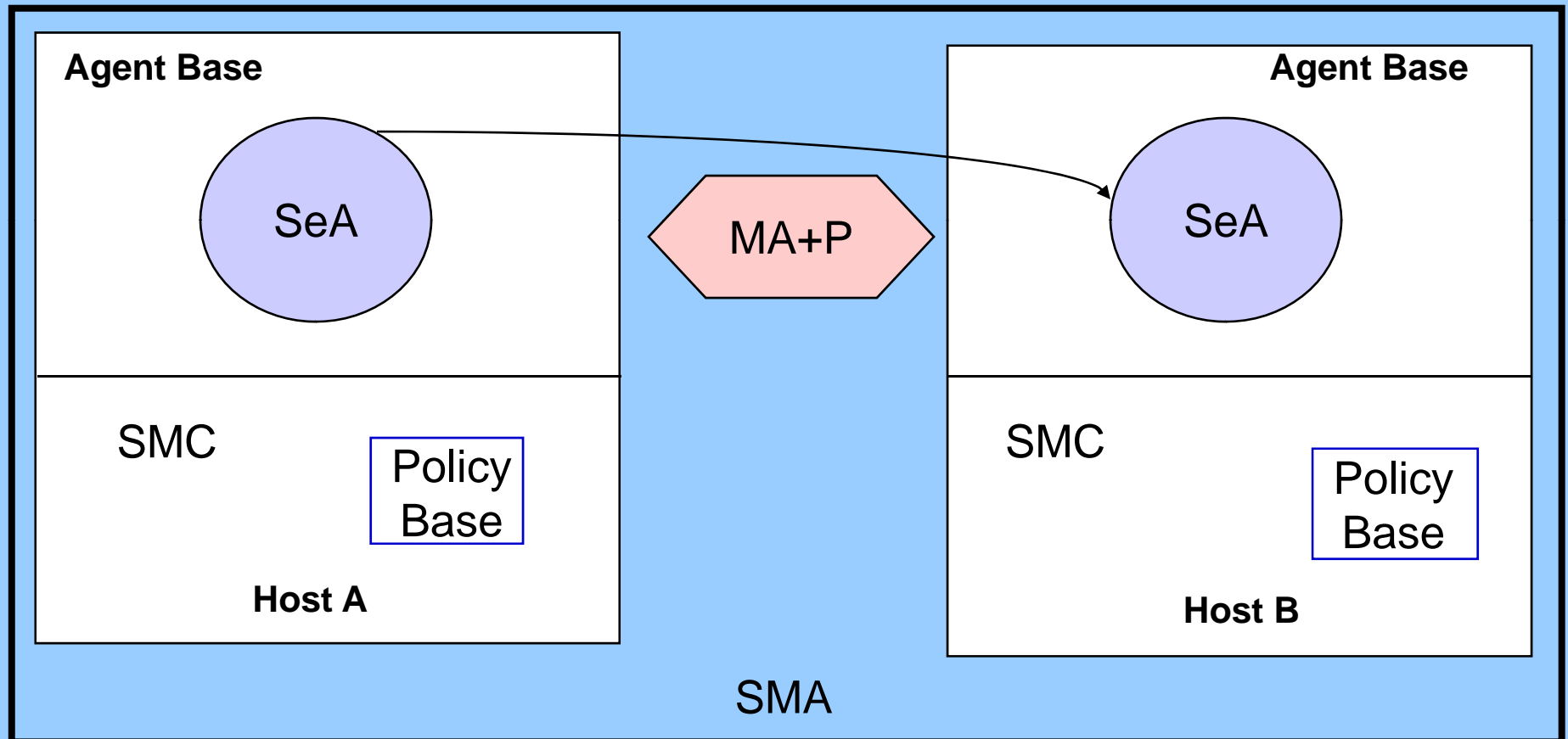
# *Mobile Agents and Security*

- ❖ Mobile Agents are autonomous software entities that move from place to place and interact with each other and the environment to achieve their own goals on behalf of their owners
  - ❖ Executable Code
  - ❖ State
- ❖ Security
  - ❖ Agent attacking the Agent Base Environment
  - ❖ Agent Base Environment attacking the Agents
  - ❖ Agents attacking each other
  - ❖ Attacks against Agents during Network Transfer

# *Mobile Agents Security and Trust*

- ❖ Mobile Agent Issues
  - ❖ Prevention of mobile agent tampering by host
  - ❖ Prevention of unauthorized tampering of the host by the agent
- ❖ Types of Trust
  - ❖ Authentication Trust
    - ❖ Authorization at the host
  - ❖ Execution Trust
    - ❖ Itinerary Composition
  - ❖ Code Trust
    - ❖ Authorization at the host

# Security Enhanced Mobile Agents



# *Security Enhanced Mobile Agents*

- ❖ Security Enhanced Mobile Agent : Agent + Passport
  - ❖ Identifier
    - ❖ (SeA Identifier, Creator-Principal Certificate, Creator-SMC Certificate, Timestamp, Lifetime)
  - ❖ Privilege-Token
    - ❖ {<IdentifierNo,Privilege,Timestamp, Lifetime>}
  - ❖ Agent\_Code
    - ❖ (Security Code, Application Code)
  - ❖ Data\_Store
    - ❖ (Data, Propagation Path)
  - ❖ Security\_Tags
    - ❖ (Security-Tag-C, Security-Tag-S)

# *Security Enhanced Mobile Agents*

- ❖ Authentication
  - ❖ Principal which sent the Agent
  - ❖ Principal which created the Agent
  - ❖ Authentication of the Agent Base
- ❖ Authorisation
  - ❖ Privileges of the Creator of the Agent
  - ❖ Privileges of the Sender of the Agent
  - ❖ Policy Base at the Agent Base
- ❖ Delegation of Privileges
  - ❖ Agent acts on behalf of Sender and/or Creator
- ❖ Non-Repudiation
  - ❖ Agent Base : Agent did such and such action at this time
  - ❖ Agent: Such and such an action was done at the Agent Base
- ❖ Secure Communication

# *Security Enhanced Mobile Agents*

- ❖ Access Control
  - ❖ SeA's Privileges and SMC's Policy Base
  - ❖ Language Based Approach
  - ❖ Variety of Access Policy Rules based on
    - ❖ Agent Identity
    - ❖ Agent and Agent Base Identities
    - ❖ Creator Principal and Agent Base
    - ❖ Creator and Sender Principals
    - ❖ Domain
    - ❖ Privileges
    - ❖ Attributes
    - ❖ Combination of All these

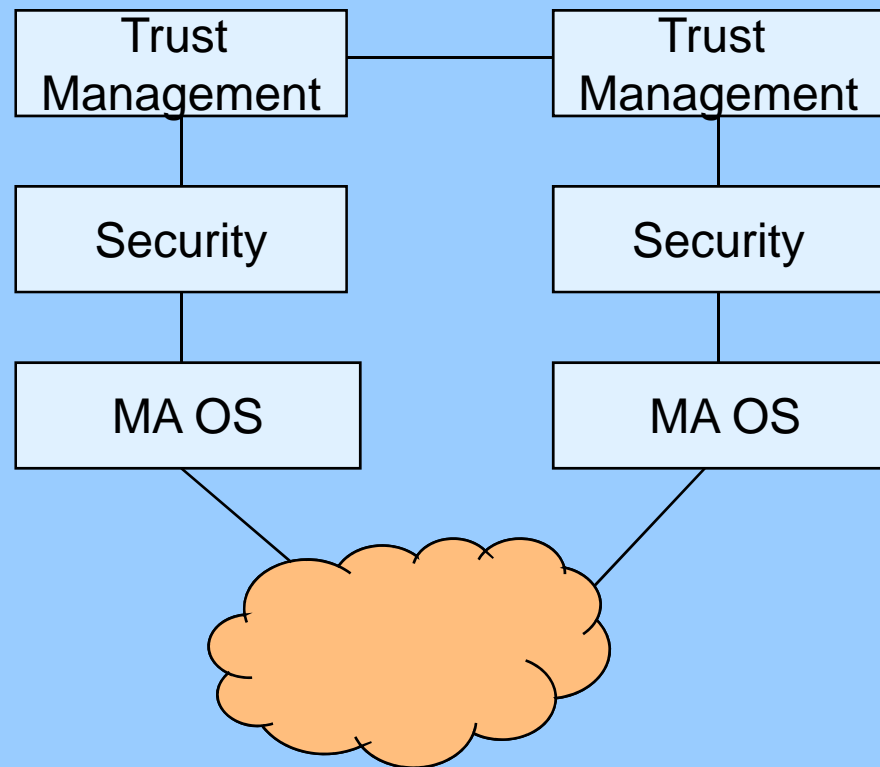
# *Security Enhanced Mobile Agents*

- ❖ If all checks successful, SeA executed
- ❖ Results stored in the Data Store
- ❖ Integrity and Origin Authentication of Results
  - ❖ Hashed Digest of Results and Timestamp Signed using the Private Key of the SMC of the Target Agent Base that is providing the service
- ❖ If Confidentiality of Results needed
  - ❖ Target Agent Base SMC generates Secret Data Key.
  - ❖ Secret Data Key encrypted using Public Key of the Client that requested the operation
  - ❖ Secret Data Key used to encrypt results

# Mobile Agent based Services Scenarios

- ❖ Single Hop Mobile Agent : Moves from one host to another and performs tasks
  - ❖ Agent returns back or returns results to originator in the form of messages
- ❖ Roaming Mobile Agents : Moves from host to host performing tasks.
  - ❖ Agent stores results until it returns to the originator in the end or sends results back time to time
- ❖ Applications
  - ❖ Flight Finder
  - ❖ Electronic Auction

# *Trust Enhanced Secure Mobile Agents*

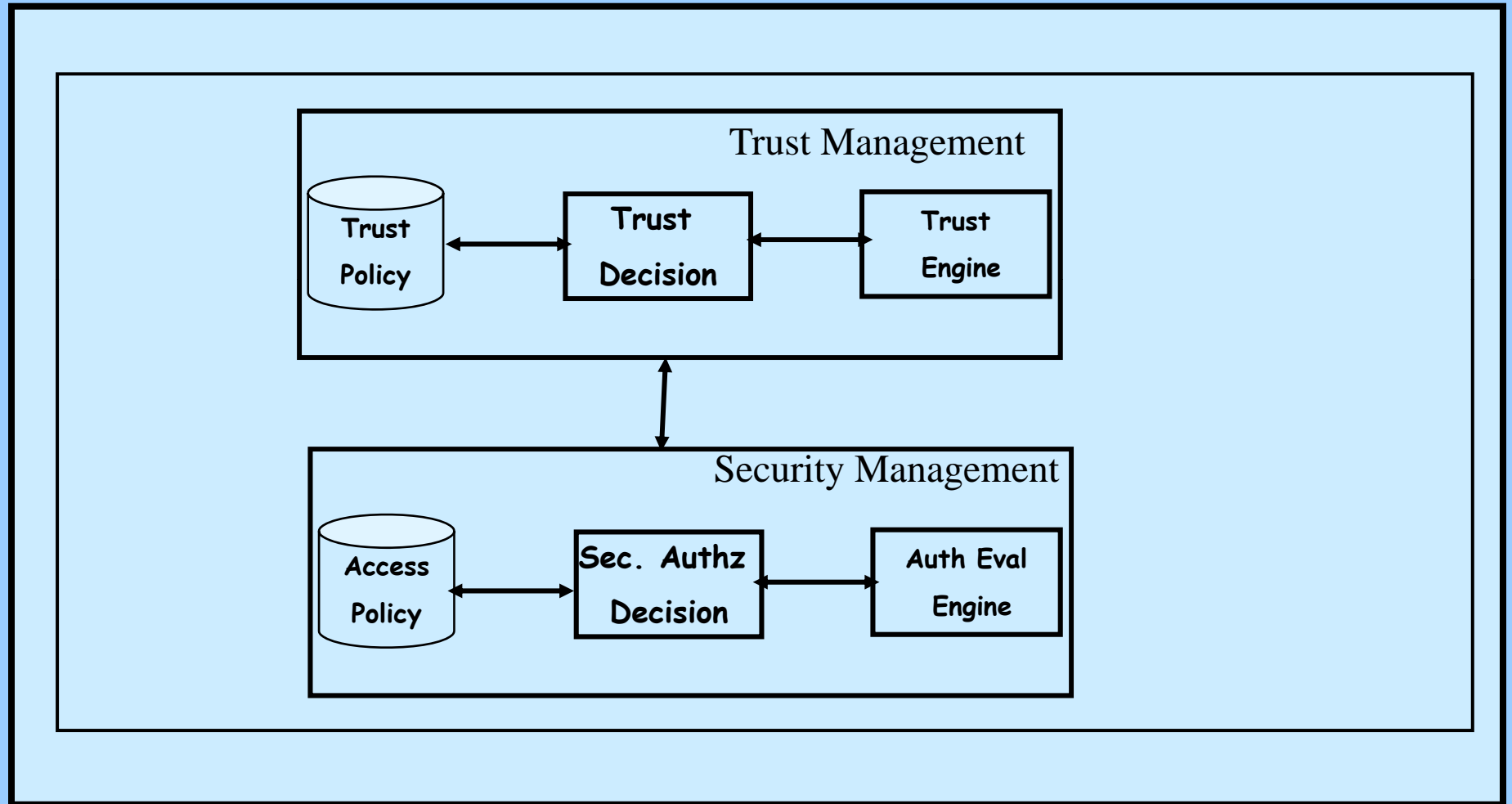


# *Trust Enhanced Secure Mobile Agent System*

## ❖ Trust Enhanced Security Solution

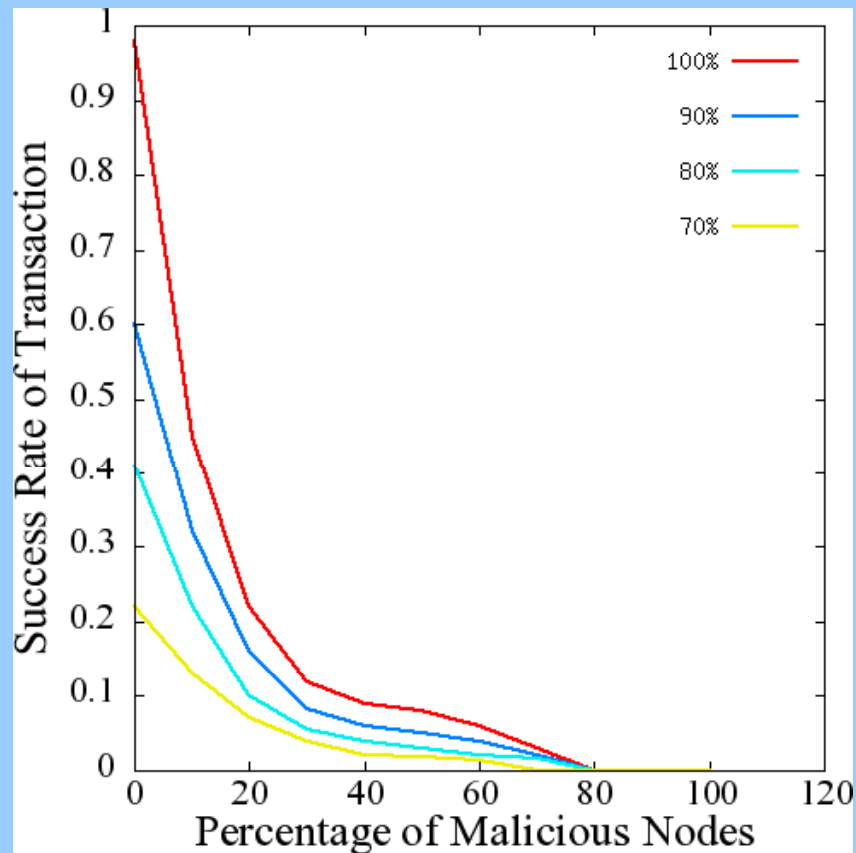
- ❖ Hybrid Trust Model : Combining “Hard” and “Soft” Trust
- ❖ Trust Model that is capable of capturing
  - ❖ Range of Trust Relationships
    - ❖ Direct, Recommended, Derived
  - ❖ Different types of Trust
    - ❖ Authentication, Execution and Code
- ❖ Trust Management Architecture
  - ❖ Representation, Evaluation and Updating of Trust Relationships and Decisions
- ❖ Trust Outcomes Enhance Security Model and Decision Making
  - ❖ Trust based Itinerary → Execution Trust (Mobile Code Security - Malicious Host Problem)
  - ❖ Trust based Authorization → Code Trust (Host Security - Malicious Agent Problem)

# *Trust Enhanced Security*

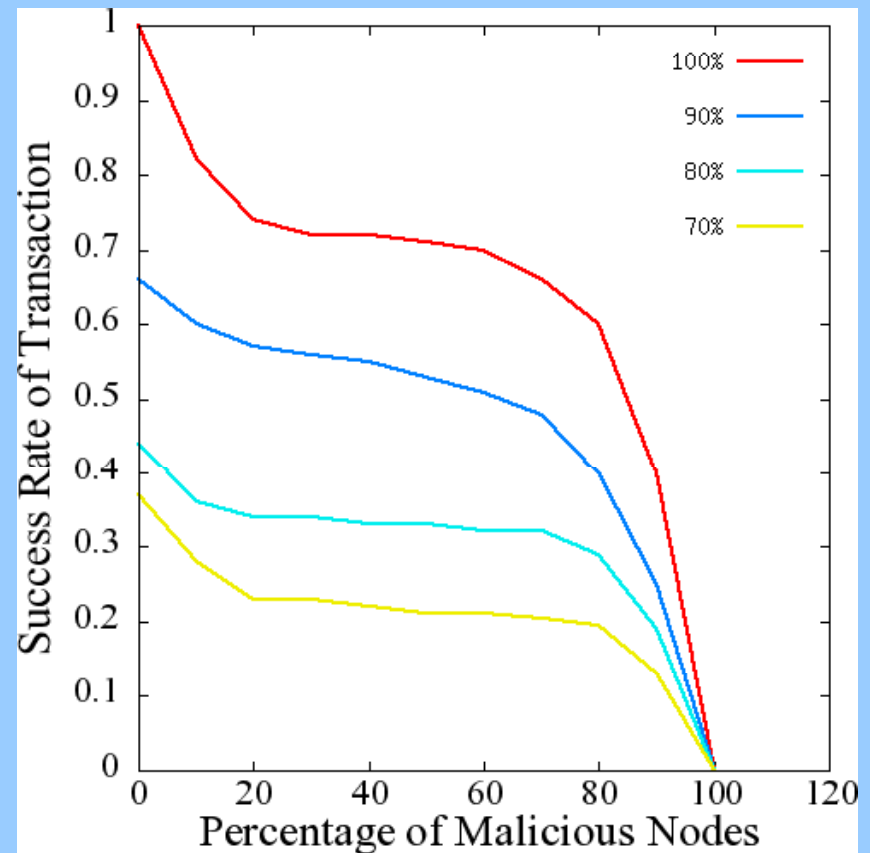


# Agent Protection

## Successful Transaction Rate

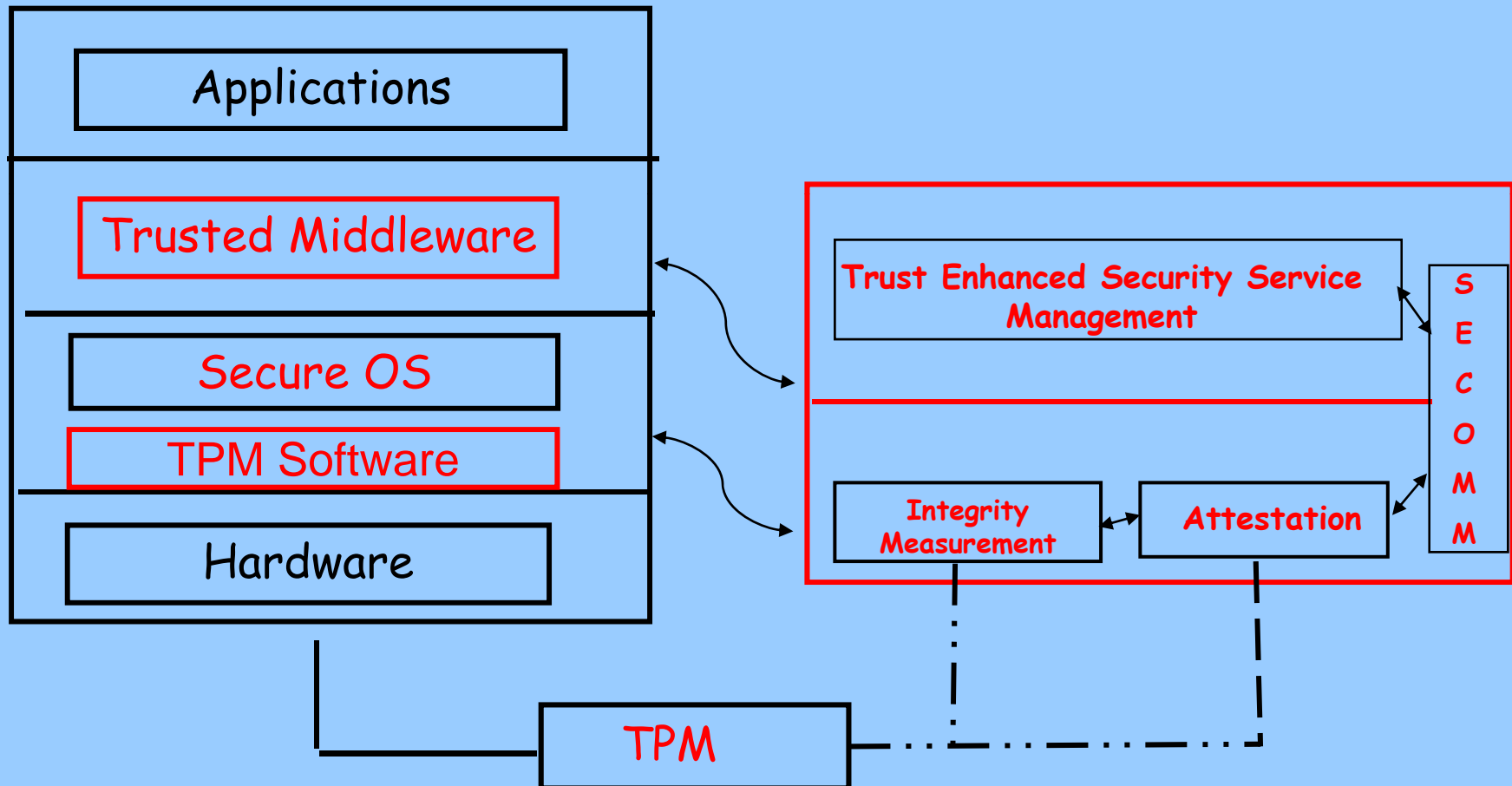


Without Trust

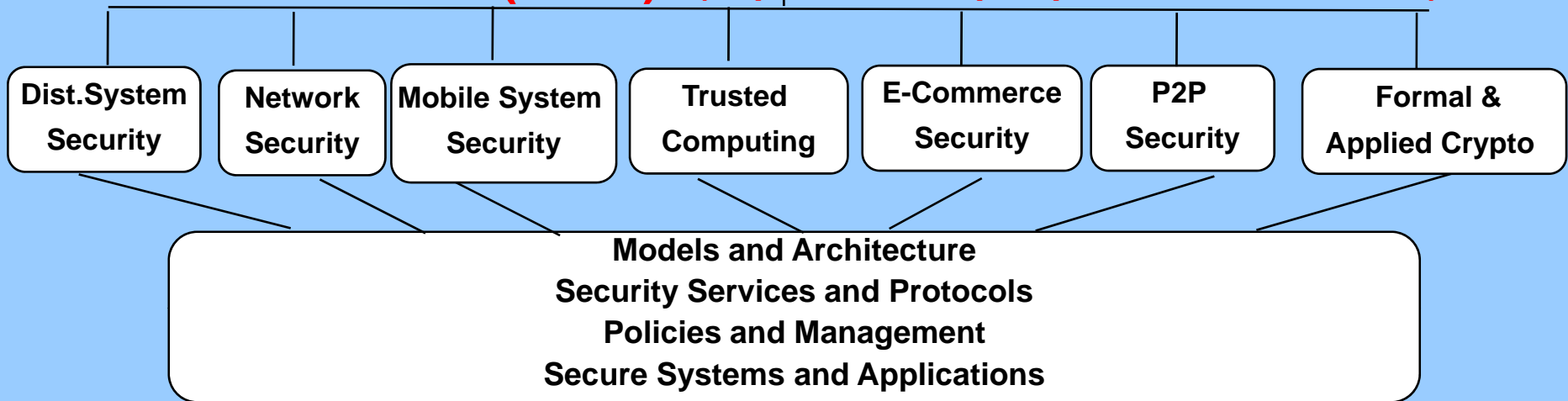


With Trust

# Trust Enhanced Secure Distributed Applications



# Information and Networked System Security Research (INSS) (<http://www.comp.mq.edu.au/research/inss>)



## Research Team

**Professor Vijay Varadharajan**

**8 Academic Staff and Post Doctoral Fellows at  
Macquarie University**

**7 Affiliate Academic Staff and Researchers  
from International Institutions**

**10 PhD Students**

(c) Prof Vijay Varadharajan APTISS August 2008, KL

