

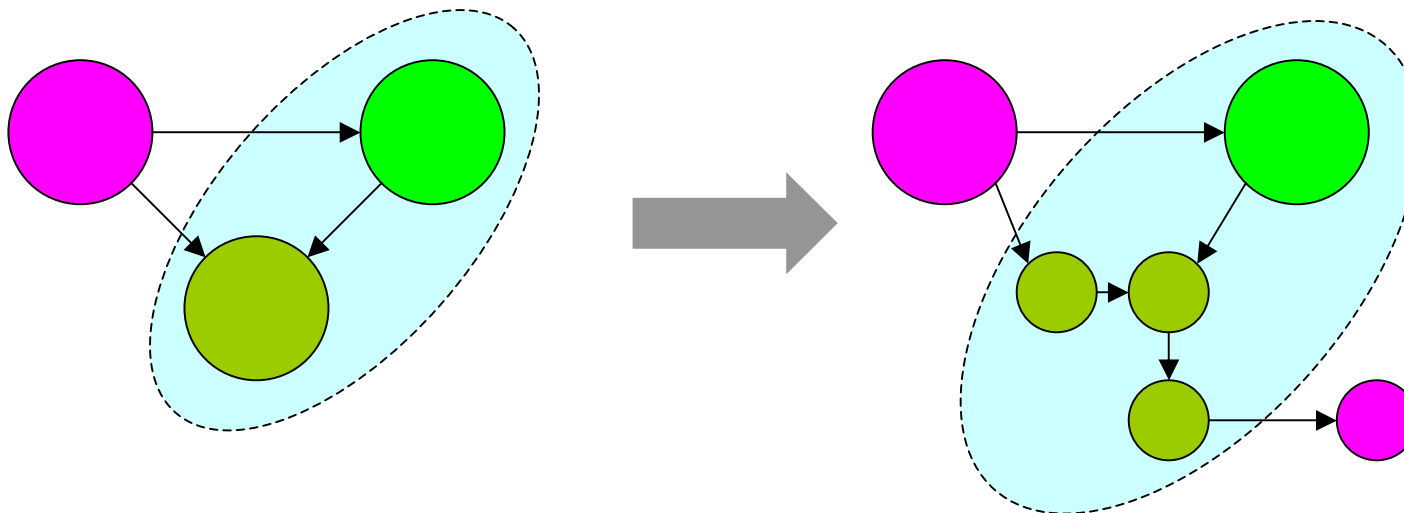
# An Architectural View of Security

Peter F. Linington

University of Kent at Canterbury

# What's so special?

- Defining a precise Security Architecture is particularly difficult.
  - Applying most architectures involves a process of Refinement - making choices that add detail
  - security is concerned with proving things don't happen
  - intruders are, by definition, not cooperative with the spirit of the architecture.



# Security Properties

- What properties is the security trying to achieve?
  - not what functions does it employ
  - apply to particular categories of activity or information
- Common properties needed are:
  - confidentiality
  - integrity
  - non-repudiation
  - reliability and availability

# Security Analysis

- No system is 100% secure in all ways, without qualification as to scope and objectives
- Analysis of system security involves
  - identification of threats
  - consequences of attack
  - cost of successful attack
  - identification of appropriate countermeasures
  - cost of countermeasures
  - cost/benefit analysis and choice of strategy
- Be prepared to review and iterate.

# Types of Threat

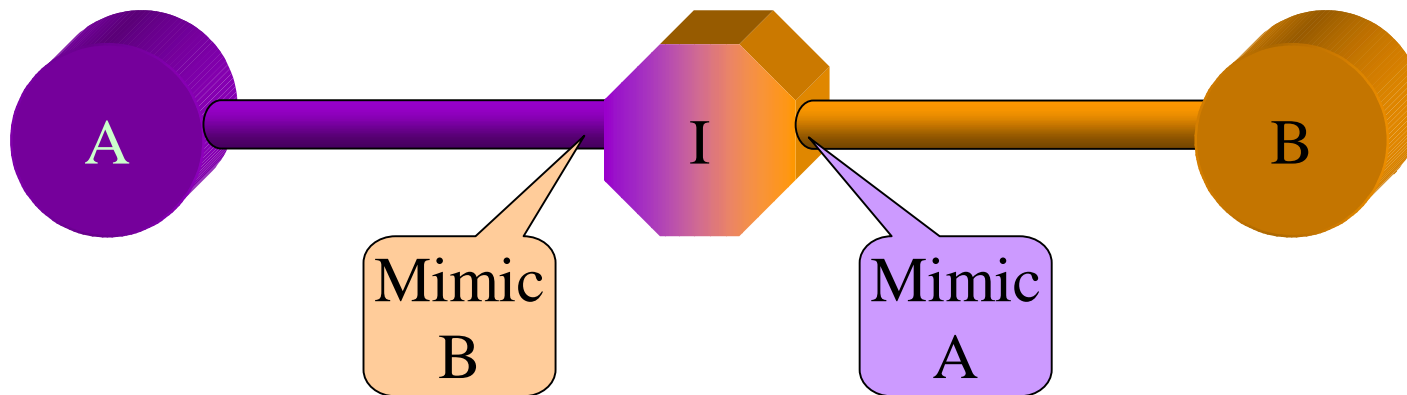
- Common threats are:
  - breach of confidentiality
    - direct external attack
    - leaking
    - inferencing
  - breach of integrity
  - repudiation
  - masquerade
  - denial of service

# Denial of Service

- Denial of service attacks can be based on
  - completely disrupting operations (bombs, crashes)
  - balance of costs (SYN attack, hotpages?)
  - disregarding obligation (locks/transactions)
- statistical degradation of service
  - complete correct process, but delay critical responses
- delegated attacks (to careless but non-malicious proxies)
  - harder to block by detecting patterns
- All based on one of the partners to an assumed contract not behaving as a willing partner, in a reasonable way.

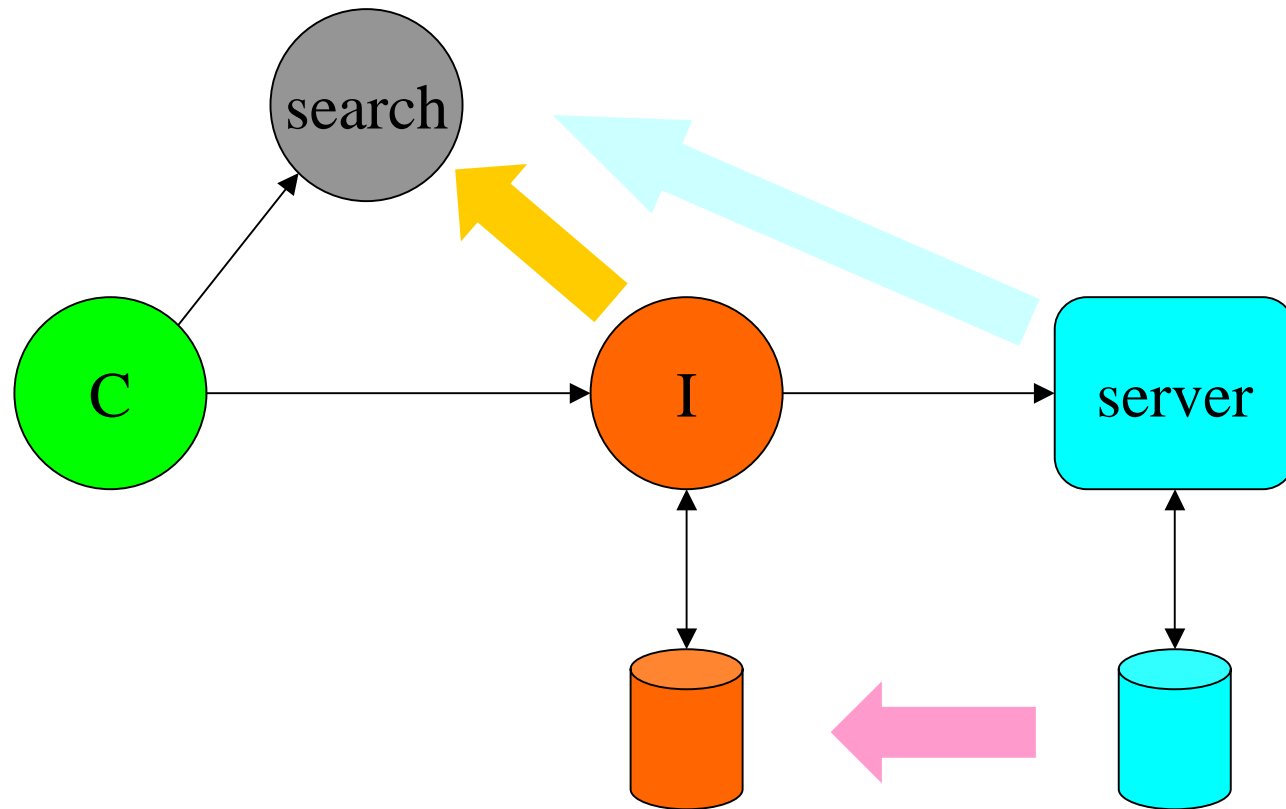
# Man in the Middle

- Intruder positions itself on the path from A to B
  - provides each side with impression it is talking directly to the other
  - intruder monitors or changes information
  - difficulty depends on degree of information A and B share initially



# A Web Example

- Intruder lifts a web page from an e-commerce site and modifies selected pointers.



# Trusted Base

- Any party need some trusted computing base from which to operate:
  - at least a trusted processor
  - some trusted third parties
  - some (but not all) aspects of secondary storage
- an architecture needs to state what the trusted base is
- identify any dependencies - e.g. rdist or similar management tools? archive and recovery mechanisms?
- if the trusted base of a sub-system is compromised, the base must be considered hostile.

# Security Policies

- Mechanism is not enough; must have basis for managing allocation of rights e.g.
  - Bell and LaPadula
    - multilevel confidentiality - read down and write up
  - Biba
    - multilevel integrity - write down and read up
  - Clark and Wilson
    - well-formed transactions - authorize user for procedure and data context
  - Chinese Walls
    - based on identifying conflicts of interest
- Role-based versions add flexibility by indirection

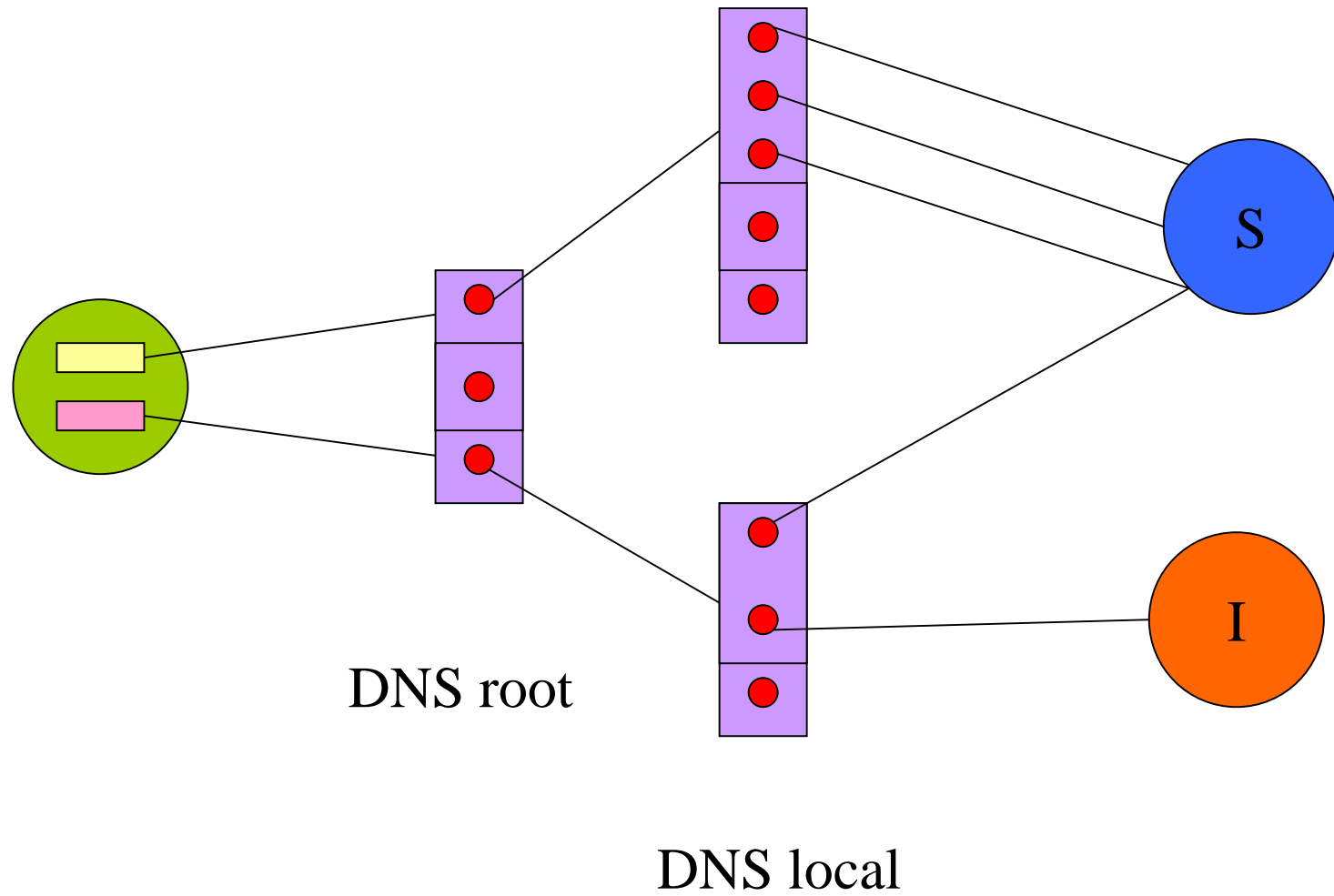
# Domains

- Groups with common policy/management responsibility
  - assume uniform level of trust within a domain
  - concentrate countermeasures at domain boundary
  - domains can be overlapped or nested
  - relations between domains not necessarily transitive
- Domains give a descriptive tool for expressing security policy and relation to organizational structure.

# Responsibility for Services

- It is often not clear how secure services are if they depend on multiple organizations. Problems result if reliance is placed on them.
- JAVA defines restrictions on communication from applets to system from which they were loaded
  - lookup name of system from address of server
  - look up all addresses of name (for multi-home)
  - lookup addresses for communication target
  - allow connection if any address matches
  - network picks best path to name
- Compromise between generality, efficiency and security

# DNS Attack



# Trusting Information

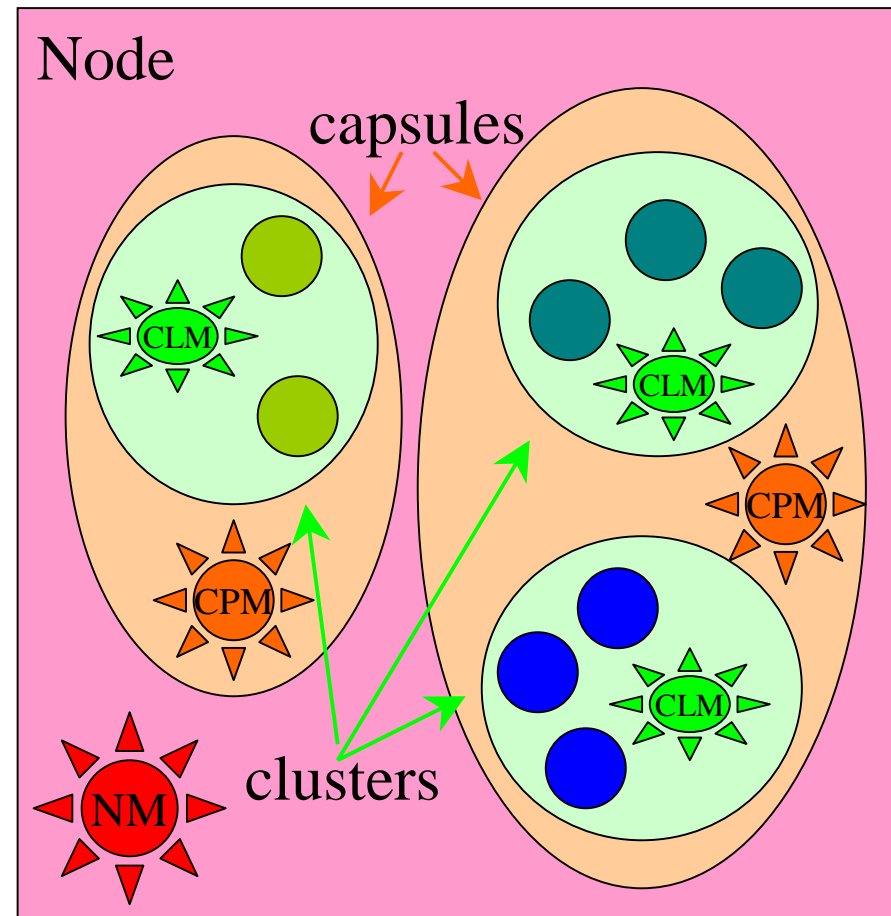
- Need to consider all steps in the paths involved in publication of information
  - particularly paths through a sequence of public or loosely managed services
  - need robust mechanisms
    - cryptographic capabilities
    - layered audit trails
      - problems of domain identity and limited visibility
    - Decoupling via independent trusted authorities
- Academic work on theory, but few open solutions

# ODP - Groups of Objects

- The engineering viewpoint identifies a number of managed groups of objects:

- nodes
  - own resources
- capsules
  - units of protection
- clusters
  - units of persistence
  - units of migration

NM - node manager  
CPM - capsule manager  
CLM - cluster manager



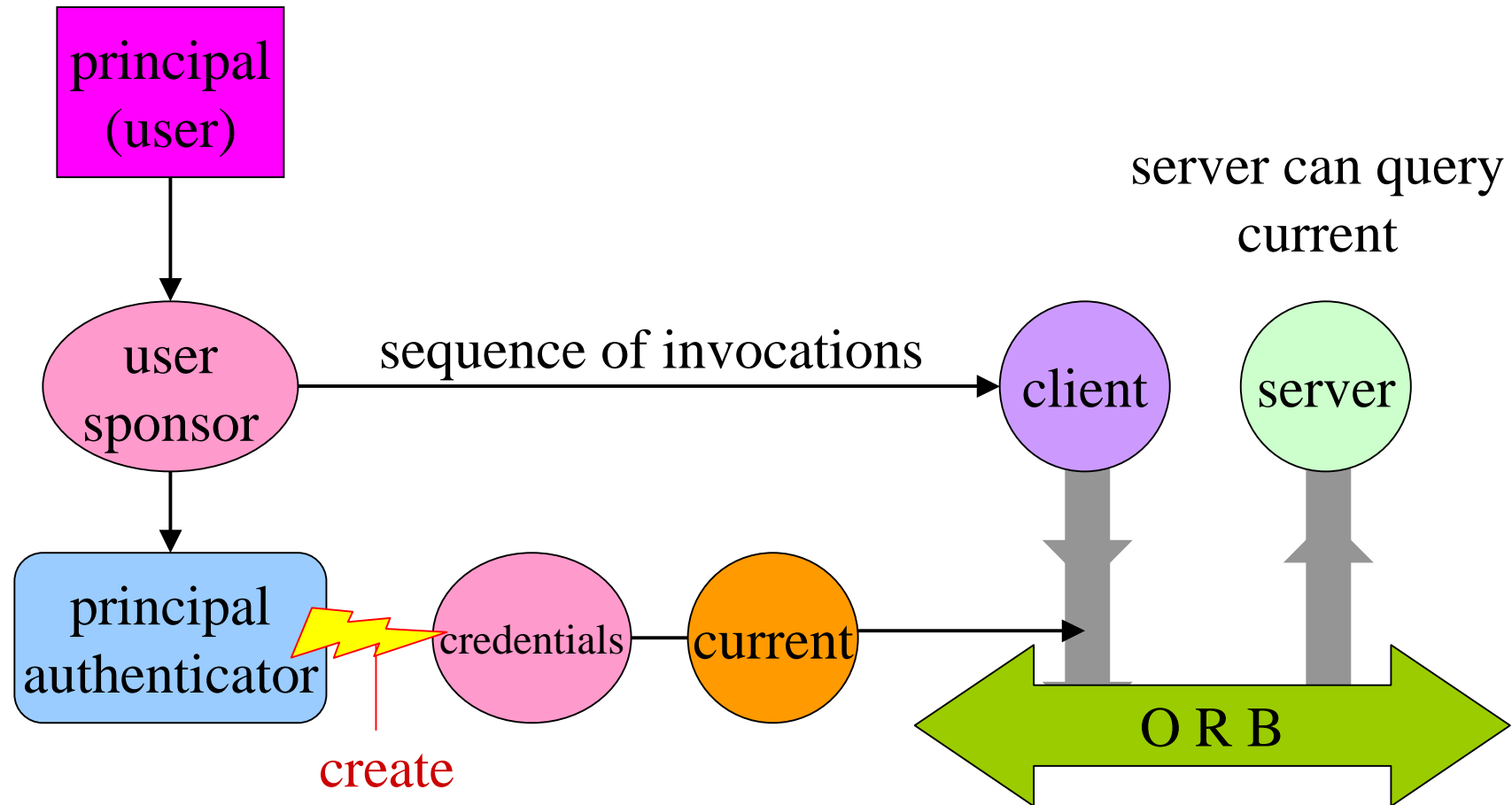
# Aware/Unaware

- There are two kinds of application to consider
  - security unaware
    - application logic not concerned with security
    - runs in an environment secured at initialization or resource allocation in a way independent of the application.
    - Provides a legacy route
  - security aware
    - application logic based on security information
    - uses services for signing, checking credentials, etc directly as part of logic.

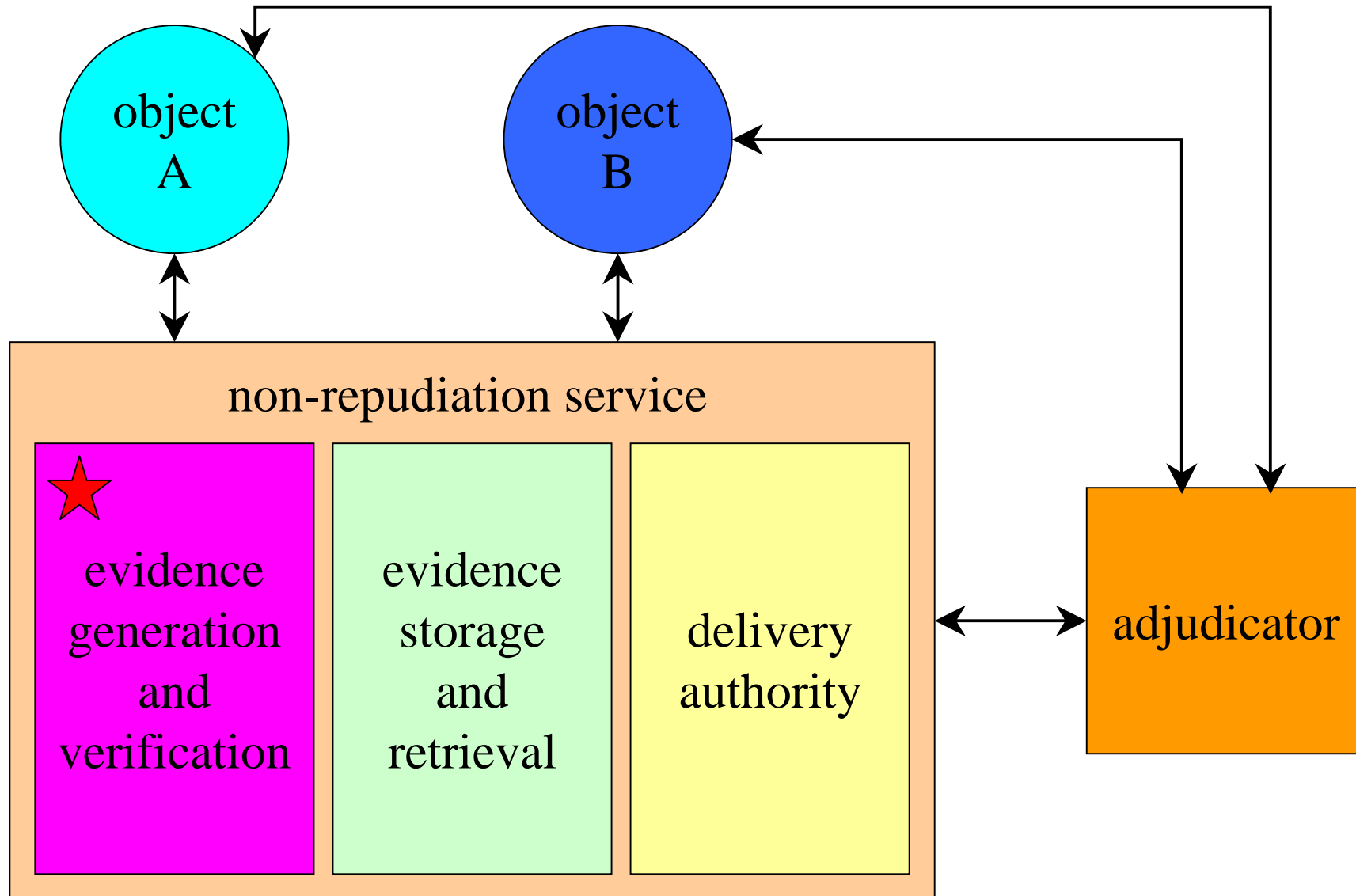
# Secure Object Models

- Simple interaction or configuration architecture concentrates on object or component interaction.
  - There are few larger structures, for example supporting
    - transactions
    - persistence
- Security model concentrates on the context in which operations are to be performed
  - original cause of interactions
  - ownership of resources
  - presentation of credentials
  - chains of trust and publication

# CORBA Security Model (1)



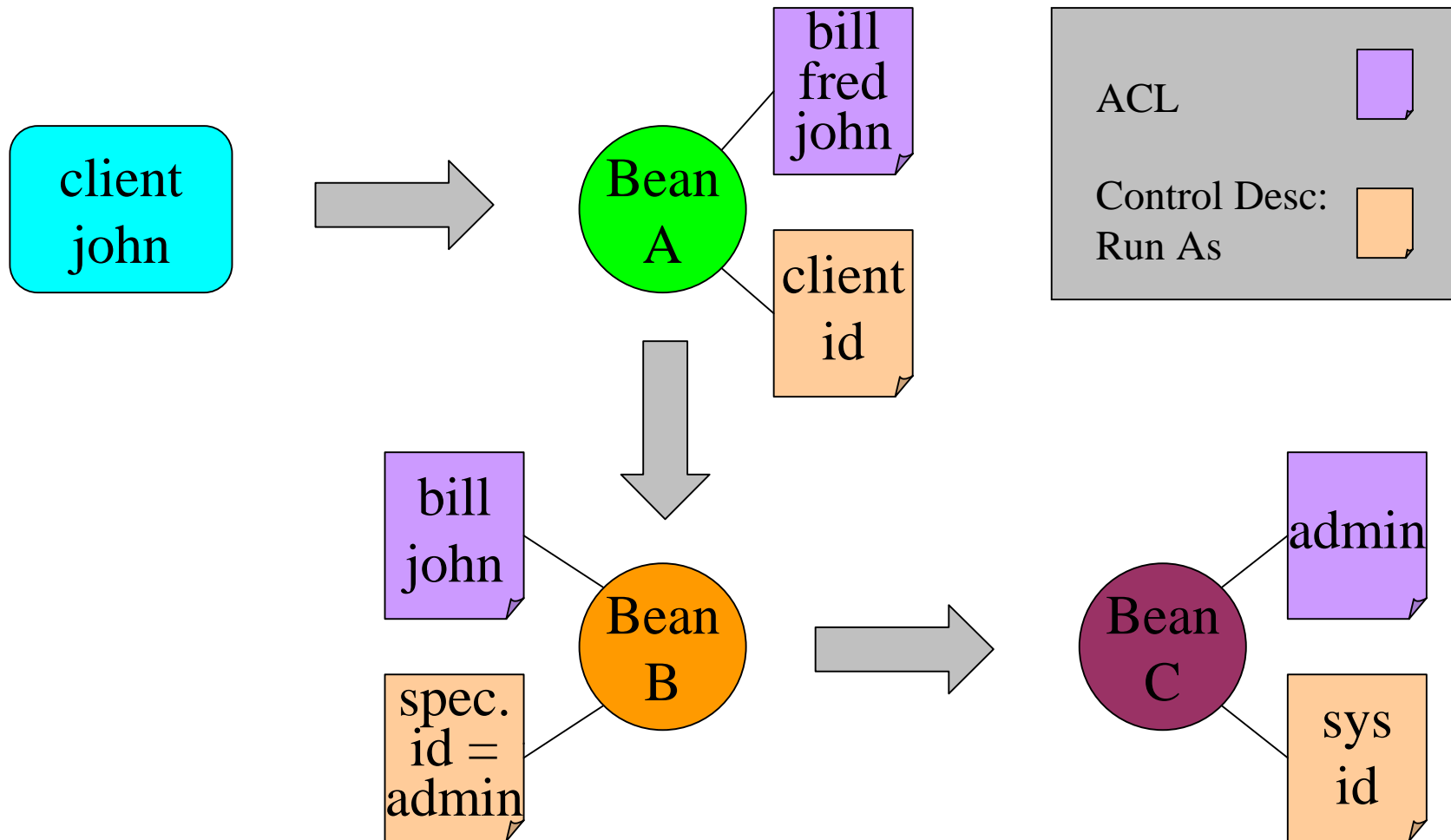
# CORBA Security Model (2)



# EJB Security

- EJB 1.1 Component model provides
  - role-based access control at method level
    - logical roles in deployment descriptor
    - methods allowed access for each role
    - local environment includes mapping from organizational to logical roles
  - chained invocation passes client identity
- EJB1.0 provides “runs as” for identity once executing
  - client\_identity - from caller
  - specified\_identity - locally specified
  - system\_identity - privileged

# EJB 1.0 Runs-As



# Some Architectures

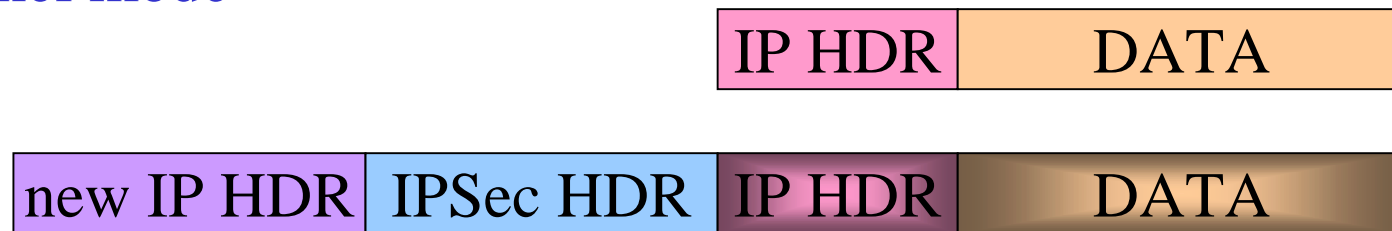
- Many specific security architectures for distribution have been proposed, including:
  - SESAME (Esprit),
  - Kerberos,
  - Yaksha (public key kerberos),
  - DCE,
  - DSSA (Digital).

# Standard Packages of Functions

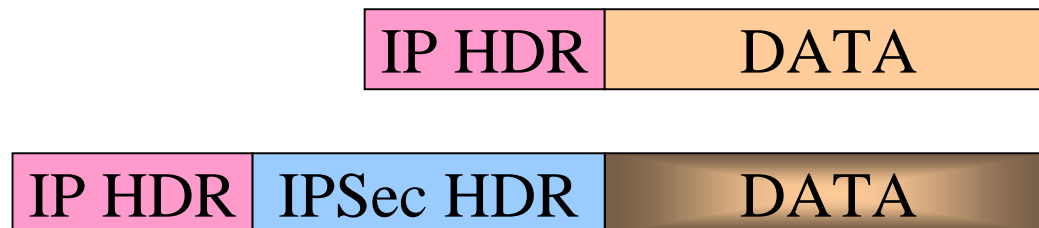
- There are a number of well know security solutions, and attempts have been made to provide independence and portability between them.
  - Generic Security Services API (GSS-API) is one of the best known [RFC 2078]. It provides for
    - creation of security contexts
    - data communication in these contexts
      - interface in terms of exchange of standard credentials and of tokens that are mechanisms specific
  - Various mechanism already support GSS-API (Kerberos, Sesame, etc)
  - Java binding of GSS-API in progress

# IPSec

- Network protocol based solutions for Security Associations
- main components are IPSec and IKE (key exchange)
- framework supporting variety of authentication (AH) and encryption (ESP) options - choice is negotiated
- tunnel mode



- transport mode

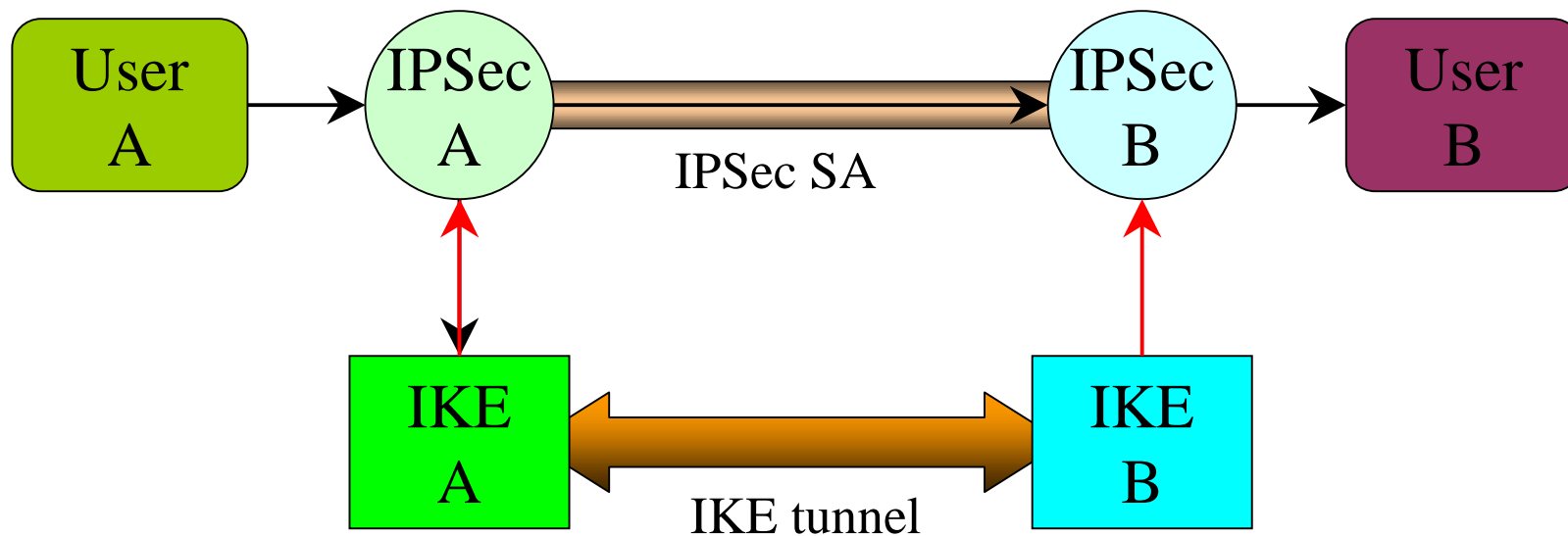


# Internet Key Exchange

- Internet Key Exchange (IKE)
  - creates an authenticated, secure tunnel between entities,
  - then uses it to negotiate a secure association for IPSec
- authentication options
  - pre-shared keys
  - mutual public key based challenge
  - digital signature exchange
- key exchange
  - use Diffie-Hellman protocol to set session key

# IPSec/IKE

- Build IPSec Security Association on demand
  - from end system to end system; or
  - from router to router across open segment

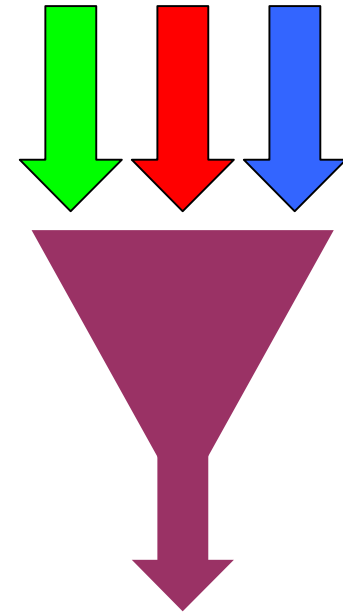


# SSL

- An application-specific solution
  - authentication end to end using PKC
  - no special involvement from network
  - need independent trusted mechanism for associating public keys with identities.
- Negotiation of profile
  - key exchange method
  - digital certificate type
  - session data encryption
  - message digest function

# Aggregation and Multiplexing

- Forming a multiplex results in the members being treated equally, and without distinction
- there is a balance between protection and control
  - traffic masking is a benefit
  - need to meet most stringent requirements
- all members see the same Quality of Service and Routing decisions



# Firewalls

- Need to defend the whole perimeter
  - no private back doors can be allowed
- Decide level of control
  - address-based
  - application protocol
  - application-specific filtering
- Decide how to deal with patterns of communication
  - ORBs using private ports
  - callback requests
  - factories and object instantiation
    - ORBIX Wonderwall as a quite capable example.

# Firewalls (2)

- Need to monitor traffic to identify reference passing
  - dynamic creation of tables
  - policy on how general support should be
  - need full decode to apply a general policy
- Pressure to use HTTP rather than IIOP to simplify firewalls
  - but http filtering requirements becoming more complex
    - riding e.g. IIOP over http
    - SOAP?
- Basic issue is still “what should the access policy be?”

# Intruder Detection

- By network monitoring and pattern-matching
  - need very robust systems, with as little access and visibility as possible
  - need data on current policies and valid patterns
  - need route for exceptions/alarms to supervisory systems
- By data-mining in the audit data
  - patterns or exceptions?
    - repeated failed attempts
    - unexpected hotspots
- By selected specific mantraps

# Audit

- Need strong, robust audit procedures
- Audit trail of all activities
  - applications
  - infrastructure management
  - policy maintenance
- Apply the 10% rule
  - you should expect audit to consume 10% of resources
    - traffic
    - cpu
    - storage
- Write only audit is not useful.

# Virtual Private Networks

- Authentication
  - enhanced digital certificates
  - directory-based coordination of information (via DEN)
- Perimeter security
  - high performance firewalls
- Confidentiality and Integrity
  - high performance encryption solutions
    - IPSec, IKE (also in IPv6 framework)
- Intrusion detection
  - threat database records becoming a commodity
- Policy based management

# Configuration and Management

- Problems with fine-grain security solutions
  - very large sets of configuration data
  - need for frequent updating to reflect organizational changes
  - need powerful tools to optimize configurations
  - large volumes of audit and event data
  - wide range of requirements leads to course checks or frequent false-alarms.
- Need to apply large-scale information systems solutions to the infrastructure.

# Directory Enabled Networking

- Applying policies to operational networks requires us to
  - broad and flexible view of devices
  - link users, applications and services to devices
- Solution is to leverage directory (e.g. map to LDAP)
  - directory contains structures for:
    - people,
    - organization,
    - services,
    - network devices
  - metamodel and DEN information model schema
  - policy constraints in terms of schema

# Claimed Benefits from DEN

- Network-wide info on user - supports single sign-on;
- Rapid deployment of new services, tailored to user requirements;
- Ability to identify, and thus protect mission-critical traffic;
- Enhanced network management - emphasis on exception reporting
- Easy access to new services
- Simple deployment and configuration
  - integration of plug-and-play solutions?

# Mobility Issues

- User and object mobility/nomadic computing raise new issues:
  - need for security guarantees from forwarders and relocators
  - new threats
    - denial of service - object kidnap in nomadic systems
  - new requirements
    - mobile security environments - mobile agents
    - security of active networks
    - confidential clusters to allow partial hiding of imported information - secure workflows?

# Security and Evolution

- Countering threats such as the man in the middle requires the checking of configuration in as much detail as possible
  - need to guarantee no unauthorized interception takes place
- Federation and evolution depend on the ability to introduce interceptors to perform various forms of translation
  - technology domain boundaries
  - enterprise domain boundaries
- Need to distinguish benign and hostile interceptors
- Reduces to establishment of shared knowledge, with controlled transformations.

# Conclusions

- In creating a distributed security architecture:
  - perform full threat and countermeasures analysis
  - take a systems view of the problem
  - maintain a balance of costs
  - balance specificity of checking with requirements for federation and evolution