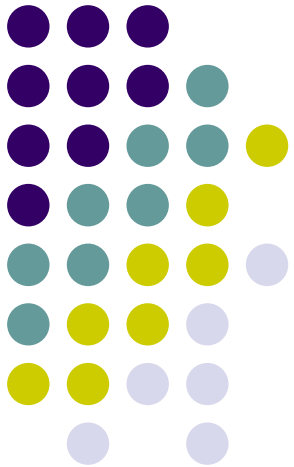
Composition Trust Bindings in Pervasive Computing Service Composition

John Buford, PhD
Panasonic Digital Networking Laboratory
Princeton, NJ, USA

Rakesh Kumar
Polytechnic University
Brooklyn, NY, USA



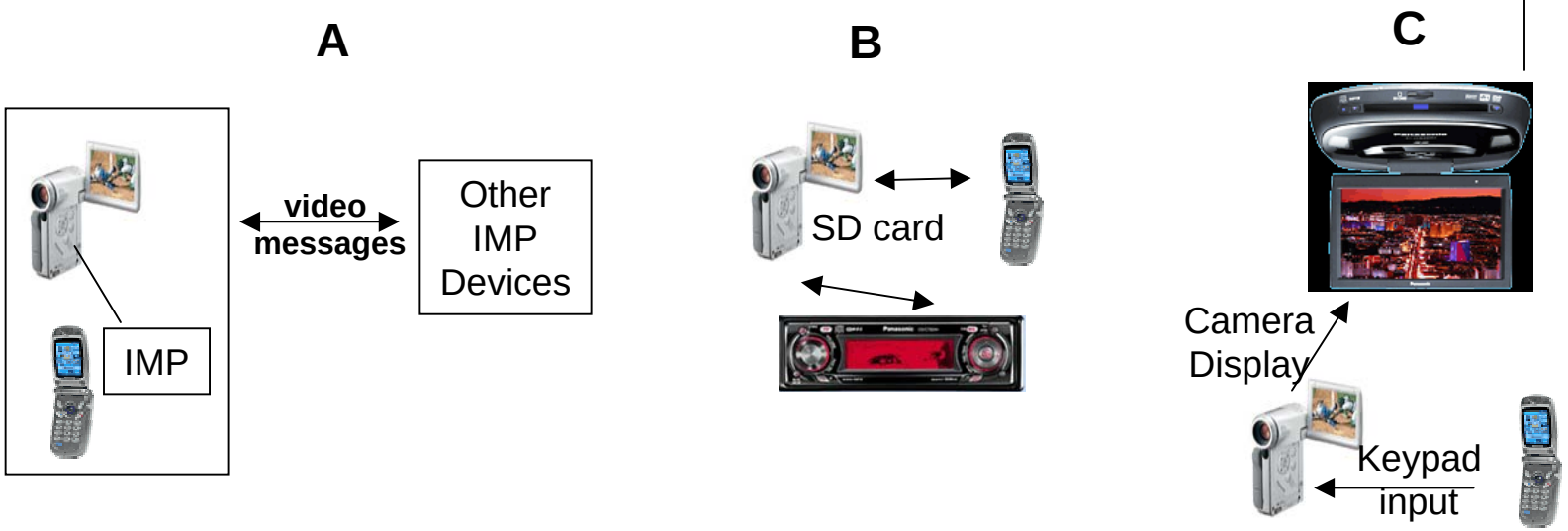
Trust in Peer to Peer Service Composition

- Concept
  - Each peer offers services to other peers
  - New composite services can be created by combining services from different peers
- What problem is solved
  - How can other peers determine whether all components of a composite service are trustworthy or meet other service criteria

Service Composition in CE

- Many consumer electronics (CE) devices are specialized for specific uses
  - Cameras, media players, game machines, internet browsing, car navigation, home security systems, GPS receivers, and personal communicators
- Due to form factor and cost considerations, devices vary in storage, display, user input, power, and processing capacity.
- Categories of composition:
  - Virtual devices
  - Multimodal interfaces
  - Computational concurrency or load distribution
  - Complex service construction

P2P Service Composition Examples



- (A) A video camera networked to a cell phone can use the cell phone's IMP software to send instant messages
- (B) A video camera networked to a cell phone or car audio receiver can augment the memory of such devices by storing information from either device on its SD card
- (C) A video camera networked to a car flipdown video display and a cell phone can use the former to display its user interface and video playback, and the latter as an input device for keypad input

Multimodal Interface Composition Examples

- Created by combining and coordinating user input/output from multiple devices.
  - Combine geographic maps and location awareness from the car navigation system with streaming video about nearby landmarks to a camera display and speech input from a cellphone
  - Remote speech-input control of home appliances using microphone on camera and sensor feedback display on wristwatch display
  - Stylus input on PDA with synchronized playback of video on camera and photos on a cell phone display

Making composition explicit in the service definition

- Service [
- interface1 […]
- interface2 […] **uses**
- interface3 […] or
- { interface4[…] and interface5[…]}
-]

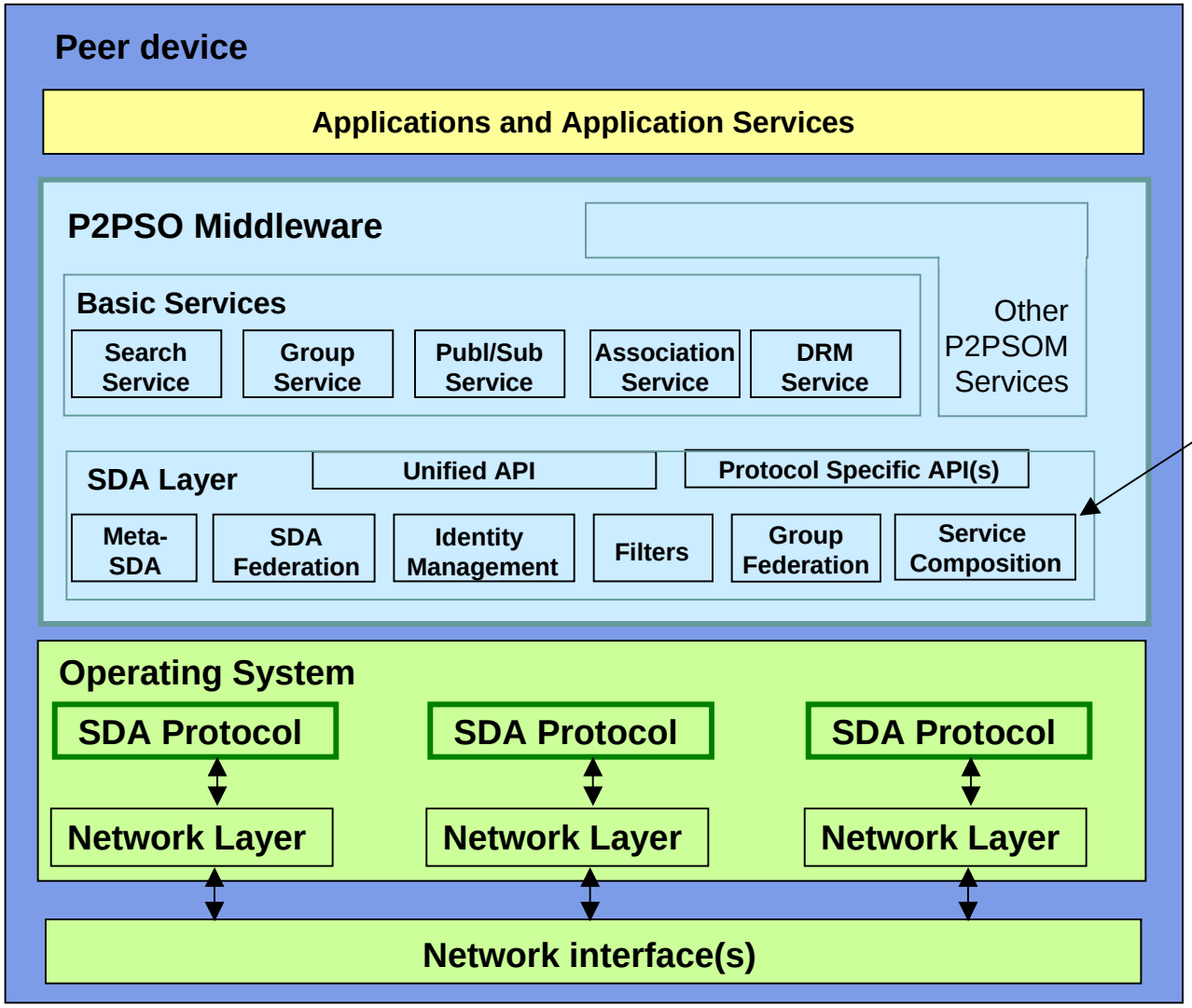
Threats due to service composition

- Control path:
  - Using computational resources for other purposes not explicitly indicated by the service interface
  - Denial of service, by effecting the rate of computation
  - Monitoring computation, to infer data or application use
- Data path:
  - Capturing private or confidential information
  - Modifying data to produce corrupted results
  - Intercepting and distributing session keys

Composition Trust Binding

- A set of rules which define the collection of allowable components for a particular service
- Components are permitted to be used in the combinations for
  - Implementing a service interface (control path)
  - Processing specific content (data path)
- CTB contains the following elements:
  - Id by which the CTB can be identified
  - Identity of the owner of the CTB
  - Service description the CTB applies to
  - Content object(s) which the CTB applies to
  - One or more component rules, each specifying the permitted components, component suppliers, component validators and expiration time of this prescription.
    - component rule can list components in various boolean combinations

P2P Service Oriented Middleware

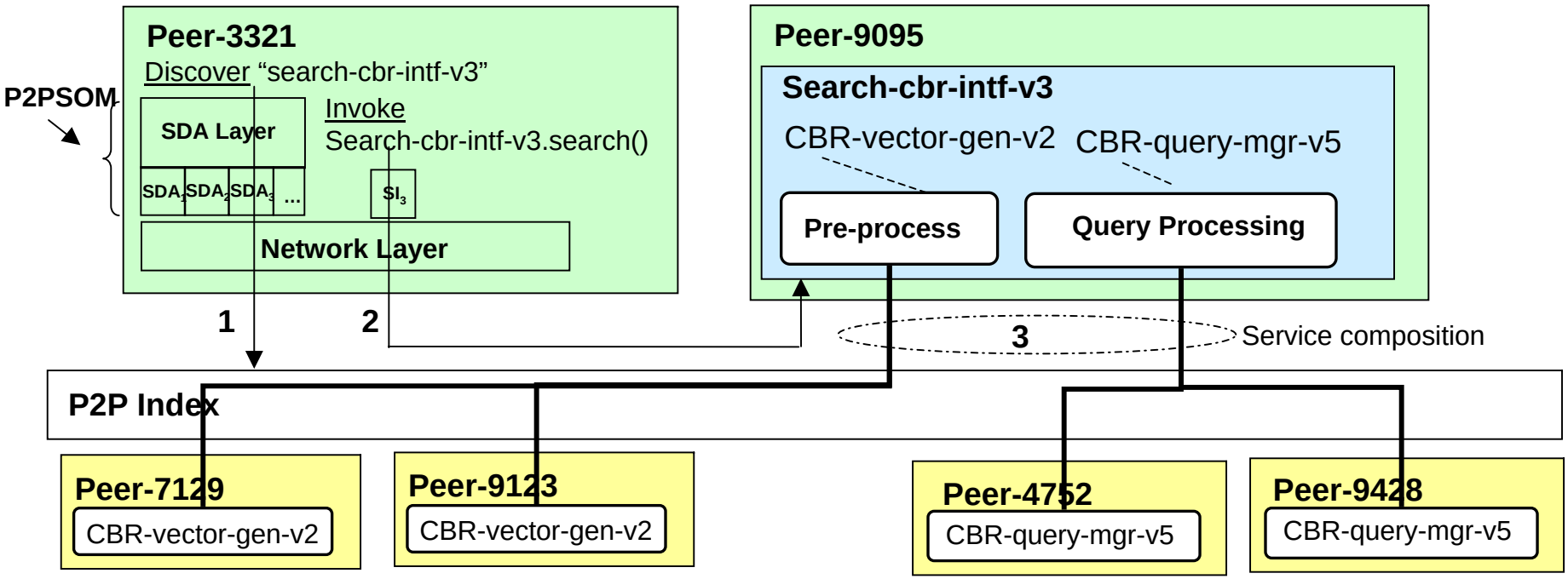


Service composition across multiple service discovery protocols

P2PSOM Service Discovery Layer

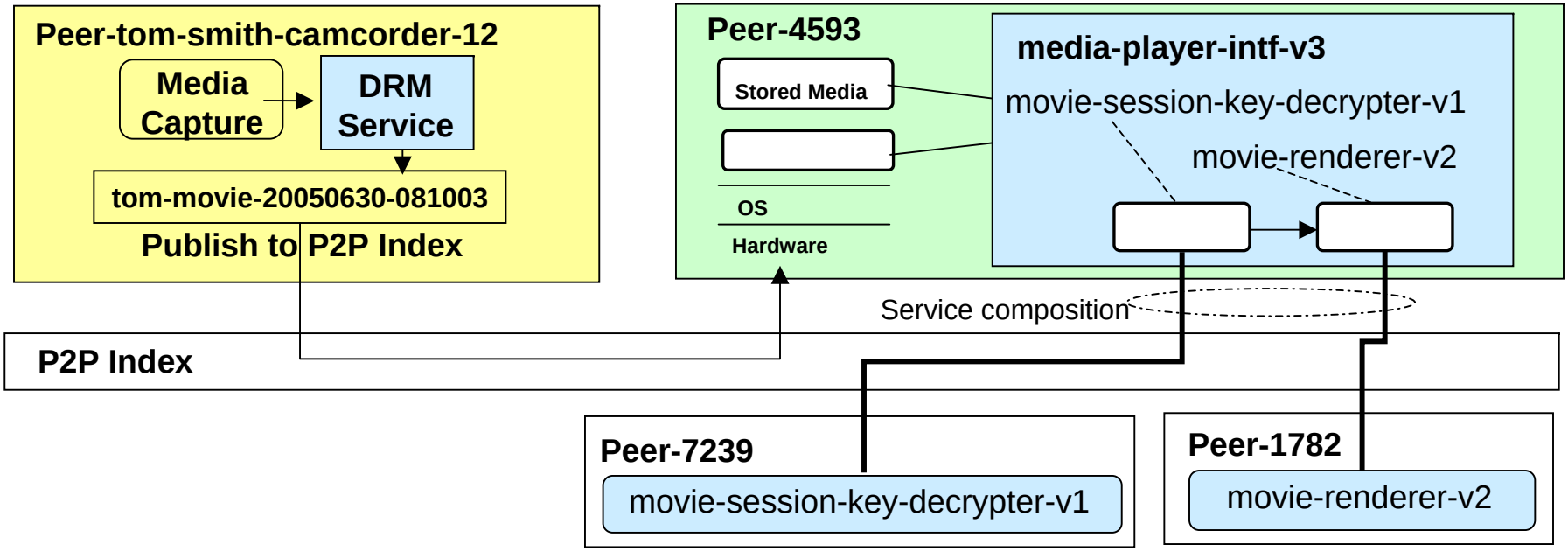
- SDA Layer provides a unified API for applications to use the various SDA protocols in a protocol neutral way. Includes:
  - meta-discovery of service discovery mechanisms organized by domain, location, or other attributes
  - federation of multiple SDA protocols into a unified protocol-independent model supporting the unified API
  - identity management of service and resource identities used in each SDA protocol to provide unified and consistent identities to applications
  - filters which allow applications to control the flow of actions, events, and state between SDA protocols and the unified SDA layer
  - group federation to manage group membership and identities across the SDA protocols and networks in a protocol-independent manner
  - service composition of services inter- and intra-SDA protocols

Control Path example:
Service Composition example:
CBR = Pre-process + query processing



(b)

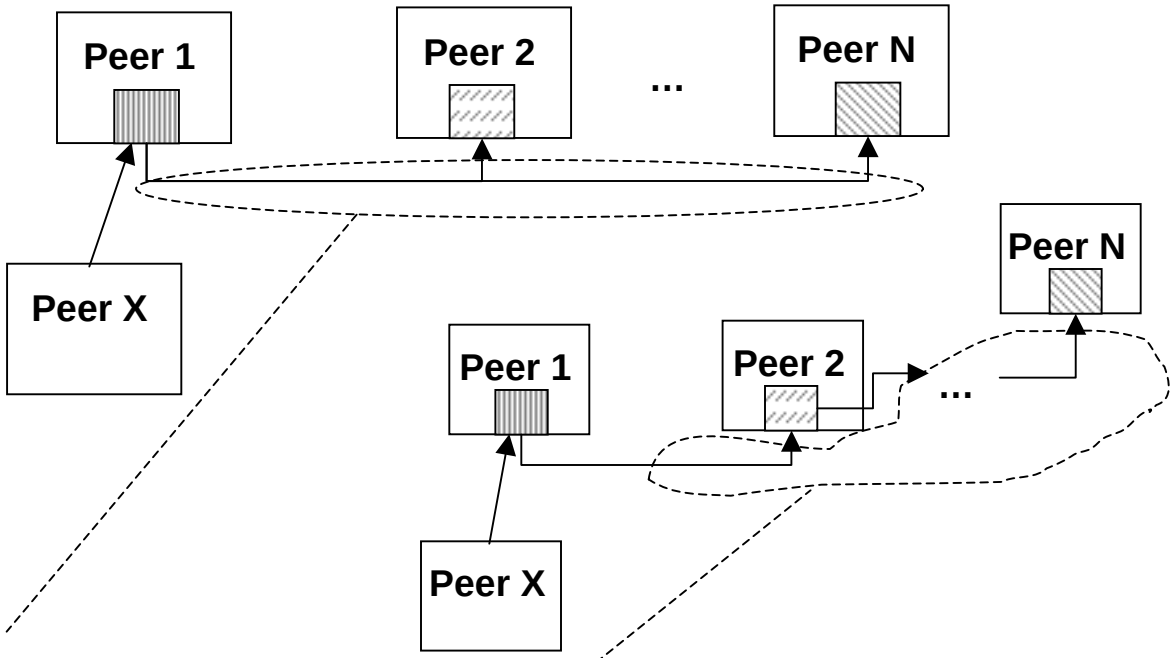
Data Path example: Content player = decrypt + render



Data Path CTB

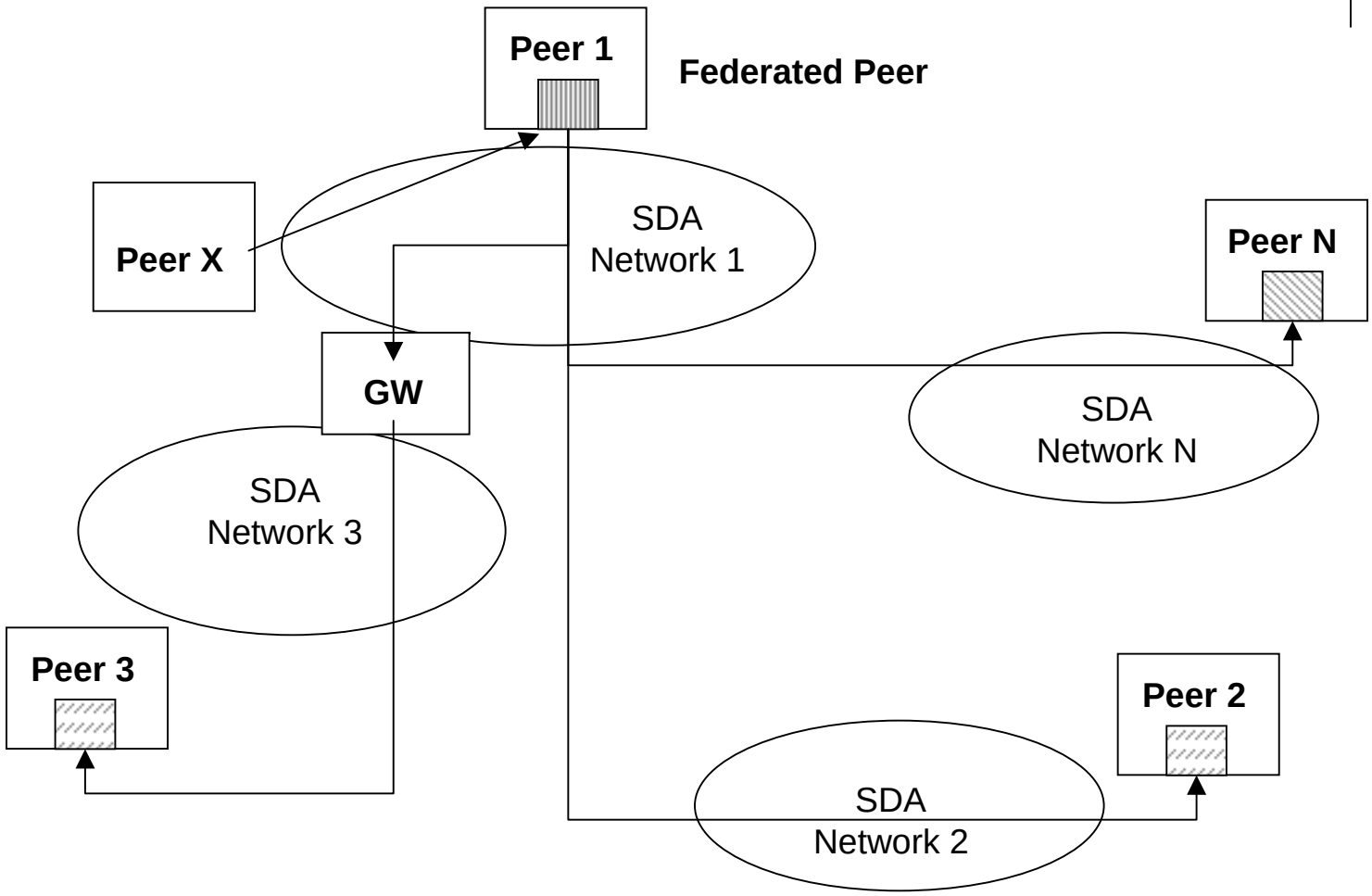
```
<CTB id=tom-smith-ctb-312>
  <specifier peer="tom-smith-camcorder-12" name="Tom's camcorder"/>
  <service-desc intf=media-player-intf-v3 format=WDSL  url=http://192.167.0.3/>
  <content id=tom-movie-20050630-081003  content-type=mpg2 />
  <component-rule-list>
    <component-rule type=and>
      <component intf=movie-session-key-decrypter-v1>
        <component-id>softcorp-session-decrypter-lib-20040930-1423</component-id>
        <version>v3.01.2</version>
        <supplier>softcorp.com</supplier>
        <validator>emx.com</validator>
        <expiration>20081231</expiration>
      </component>
      <component intf=movie-mpeg2-renderer-v3>
        <component-id>xographcorp-mpeg-render-lib-20050114-213</component-id>
        <version>v1.05</version>
        <supplier>xograph.com</supplier>
        <validator>emx.com</validator>
        <expiration>20061231</expiration>
      </component>
    </component-rule>
  </component-rule-list>
</CTB>
```

Composition Patterns



Trust binding enforcement at invocation, service advertisement, or service description time

Service Composition in Federated P2P



Discussion

- Utility of the CTB approach depends on
  - (1) Ability of represent the policies and composition scenarios of interest
  - (2) Ability to securely enforce the CTB in a distributed context
- Representation issues effecting the CTB include:
  - (1.1) Updating the CTB for changing service interfaces, component interfaces, and component suppliers
  - (1.2) Complexity of the CTB

1.1 CTBs and Interface changes

- Interfaces change relatively slowly compared to implementations
- Service offering peer might move to a new version of an interface with a different composition model before the service user or content provider has validated these and produced a new CTB
- Implementations that had already been validated might be obsoleted
  - vendor no longer supports them or the vendor no longer exists
- These problems are not unique to CTBs
  - A solution for CTBs that cannot be updated is a backward compatible deployment of the necessary services
  - For CTBs that can be updated, a mechanism by which content licensees can obtain updated CTBs as needed.

1.2 CTB Complexity

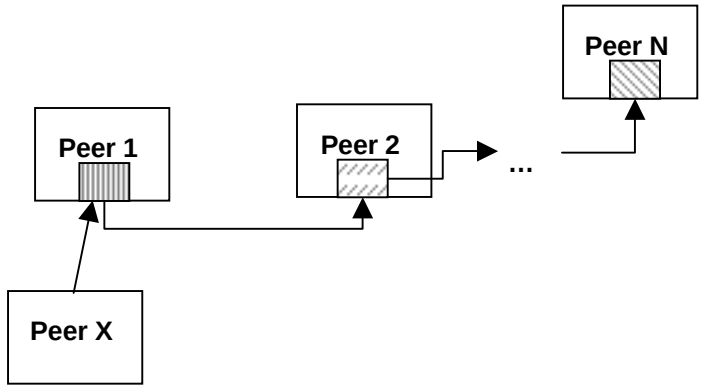
- A service may be composed of other services to an arbitrary nesting level
  - A CTB might prescribe only the first level components, if the validated component services include CTBs that the composite CTB trusts
  - A CTB might prescribe component compositions to several levels, but this increases the complexity of the CTB and makes it more difficult to maintain
- CTB also becomes sizeable if components are implemented by large numbers of software providers.
- CTB assumes that content publishers and service users are aware of services from different implementers.
  - Past experience with component suppliers for distributed middleware such as DCOM, CORBA and Java EJB suggests that this is manageable, and online registries could be created to streamline this communication.

2 Securing and Validating CTB

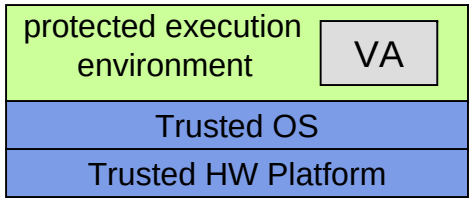
- **CTB validator** is an agent that verifies that a particular execution combination of components, services and/or peers is valid
  - According to the trust binding required by the invoking peer
  - According to the trust binding required by the content owner or licensor which is processed by the combination.
- Placement: **validation agent** (VA) may be in the component participating in the application or service, or in the OS
  - VA may communicate with each component, OS and hardware, and may communicate with VAs on other platforms if components or services are located on those other platforms.
  - VA may be itself secured via a secure OS on a trusted computing base, a smart card, Java Card, or other security technology.
  - VA could be integrated with the system loader to monitor launching of components on a given platform.

2 Securing and Validating CTB

- For nested compositions, validation agents coordinate with downstream agents to insure compliance
  - Enforcement of constraints on nested components appears to be recursively decomposible.
    - Local agent communicates with each remote agent enforcing the next level service composition
    - Local agent trusts that the remote agent enforces the immediate composition as well as nested constraints. Each remote agent repeats the process for its nested compositions.

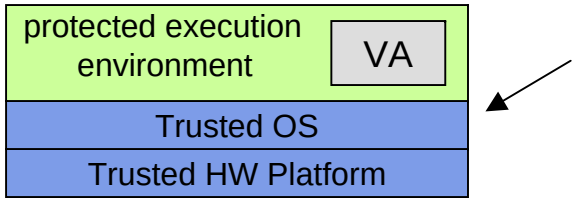


Securing the Validation Agent



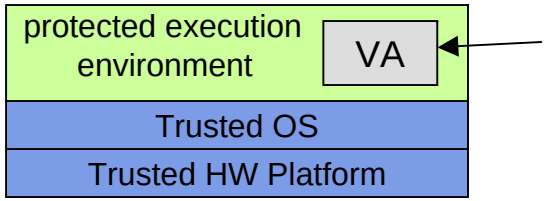
- Trusted System
  - E.g, Trusted Computing Group's (TCG) Trusted Platform Module (TPM)
  - TPM controls the system during boot, validates the boot ROM, loads and executes it, and verifies the system state.
  - TPM then validates an initial piece of the OS along with all boot time load modules.
  - These steps securely instantiate the system into a known state before the OS takes over.
  - The TPM can also provide authentication that it is a trusted platform via the peer entity authentication protocol.

Securing the Validation Agent



- Secure OS
  - A secure OS with a well-designed security policy is needed in addition to the trusted system
  - Most secure OSes feature Mandatory Access Control (MAC)
    - limits process capabilities and provides protected domains based on the intended use of the system
  - Examples: CE Linux Forum, Microsoft Palladium, and research systems such as SELinux, RBAC, and TE.

Securing the Validation Agent



- Required Properties of the VA
  - The secure OS is one or more software programs digitally signed by the provider(s) and/or integrator of the secure OS
  - The secure OS is certified for secure and trusted installation and execution on the TCP
  - The VA is a software program digitally signed by the provider of the VA and may be additionally signed by a third-party verifier
  - The VA is certified for secure and trusted installation and execution on the secure OS

2 Securing and Validating CTB

- Each component of a binding must have a secure, unique, verifiable id.
- The binding must be encrypted, such as by using the peer's public key when the CTB is generated.
- CTB is not intended to describe dynamic variations in component use at different points in a process lifetime.
- VA may not be able to enforce or monitor all possible communication paths between possible components, and the CTB is not intended to replace access control and authorization mechanisms.
- Strength of the CTB is to express a known set of component relationships that have been validated through other means (e.g., software audit and integration testing) for a specified environment or platform
  - Components adhering to the specified service interfaces and signed by trusted parties are expected to adhere to the desired access policy with greater reliability than component combinations that have not be validated.

Summary

- In pervasive computing, peers
  - implement services using services from other peers
  - use components from various sources
- CTB is a prescriptive set of rules
  - Define the combination of allowable components for a particular service or application
- Existing authentication and authorization methods in service composition do not address the trust requirements of distributed composition.

Summary

- Extends the practice of digitally signed software as used to provide software component trust
  - Assurance of digitally signed software is invisible to remote applications which invoke services on the platform which in turn use these components.
  - Similarly, content owners whose content has been transferred to the platform for processing have no way to obtain assurance about the components processing the content by using digitally signed software alone
- Enforcement of a CTB provides additional assurance in networks where nodes in multiple administrative domains
  - Share computational resources
  - May be used to process information which is under an access control policy.