Trust Requirements in TAPAS Application Server

N. Mezzetti

7th February 2003

Acknowledgements:

V. Ghini, G. Lodi, F. Panzieri

Outline

- Security and Trust;
- Trust requirements for application hosting:
  - Responsibilities of a Component Execution Environment (CEE);
- Trust-aware Containers:
  - Which guarantees a Trust-aware Container should provide;
  - A Container extension to meet Trust requirements;
- A Scalable Approach to Trust Aware Provision of Application Services:
  - A proposal for a trust architecture in TAPAS;
  - Implememtation issues.

Security:

- Security is the ability of a system to prevent unauthorized access or handling of information;
  - Security is addressed by handling access control using the finest granularity: *each principal is known and is granted a set of permissions*;
  - Security enforcement characterizes a security domain.

Trust:

- Trust is the firm belief about the competence or the honesty of a principal in a particular context;
  - Trust is addressed by making assumptions about the possible behaviour of principals and dividing them into classes of priviledges: *each class is granted a set of permissions*;
  - Trust enforcement characterizes a trust zone: it is usually employed to interconnect several security domains.

Security and Trust (1/2)

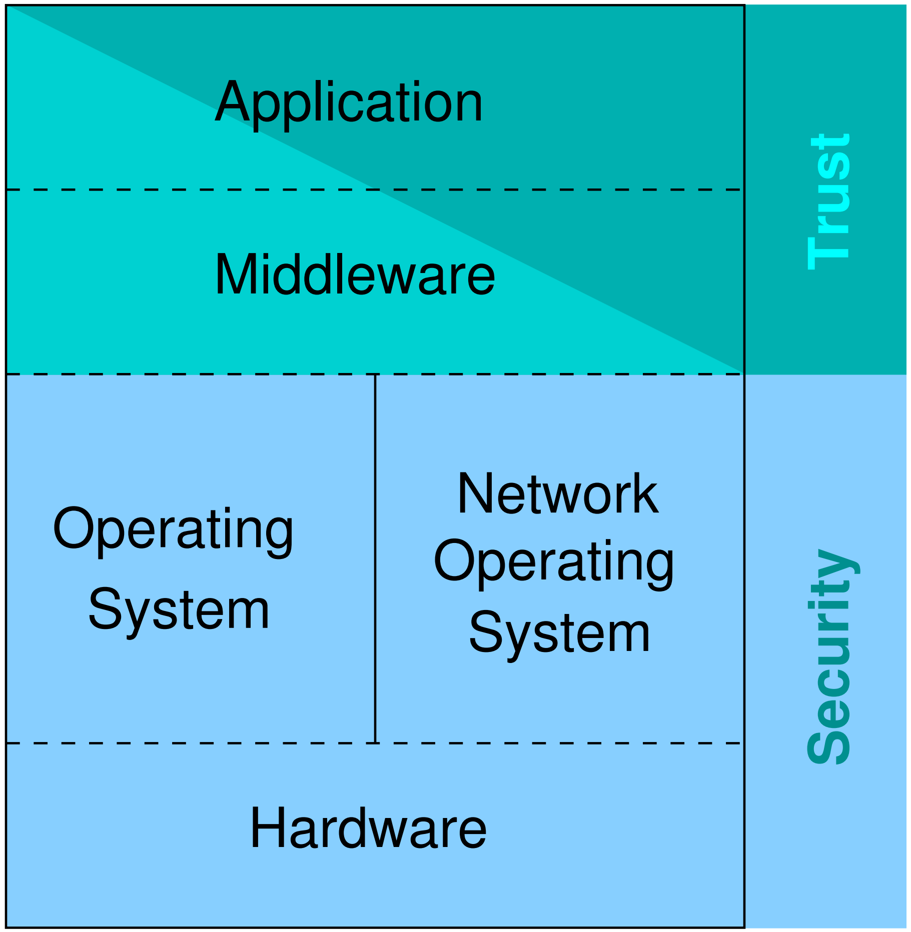
Security: access control to resources in a single security domain.

- Doesn’t scale with respect to the increasing number of principals.

Trust: access control to resources shared in a virtual enterprises.

- Scalability property allows trust management to address access control in arbitrarily large environments.

Security and Trust (2/2)



Trust Requirement for Application Hosting

To host an application, an ASP must trust:

- the Application Owner (AO);
- that every part of the application it is hosting belongs to the AO
  - Both SLA and application code must be signed by the AO;
  - SLA must uniquely identify the code to be deployed
    - e.g., SLA contains hash value computed on application binaries.

Trust-aware Application Hosting

A Trust-aware CEE has to grant to applications:

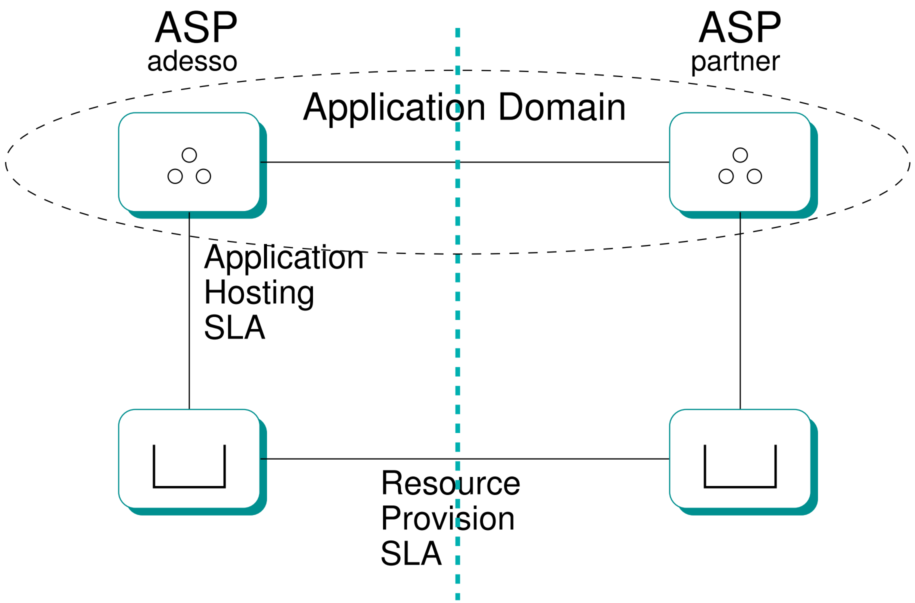
- Security: prevention of unauthorized access or handling of information
  - Principal Authentication and Authorization of service invocations;
- Privacy: prevention of eavesdropping of data in transit or on storage.

A Trust-aware ASP should provide applications with:

- Confidentiality: keep secret about application reserved data, if known
  - Implementable as a contractual statement.

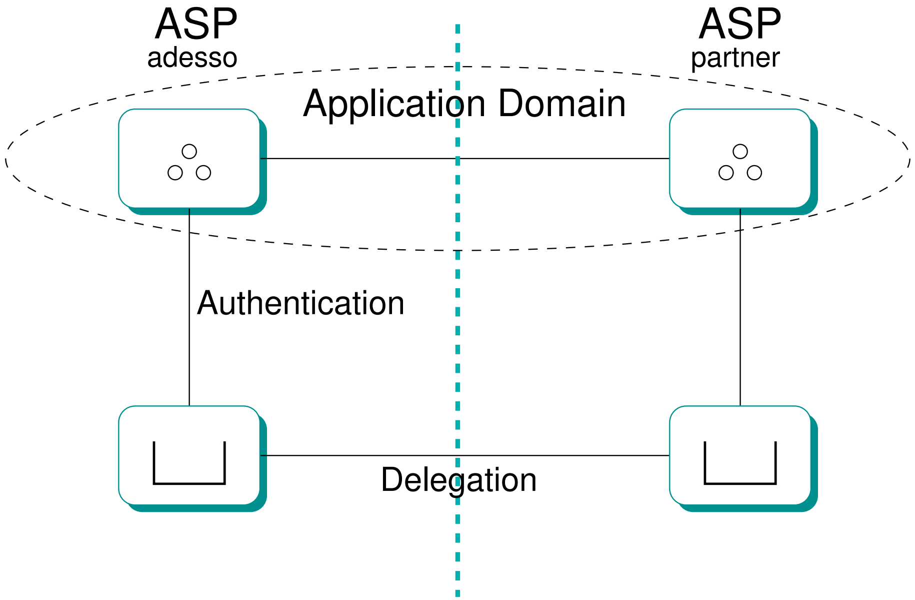
Trust Relationships in Application Hosting (1/2)

- Application Owner has an Application Hosting SLA with the ASP;
- The ASP can have SLAs with partners for supporting Application Hosting requirements.



Trust Relationships in Application Hosting (2/2)

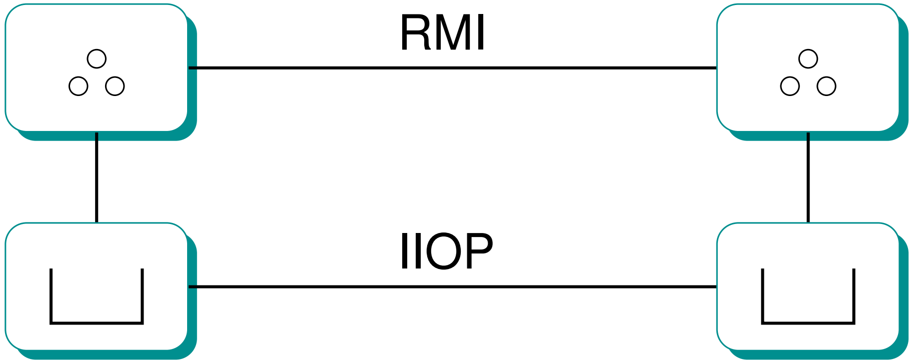
- Vertical Relationships require authentication to be checked
  - Application Hosting SLA provides trust;
- Horyzontal Relationships need delegation
  - Resource Provision SLA provides trust.



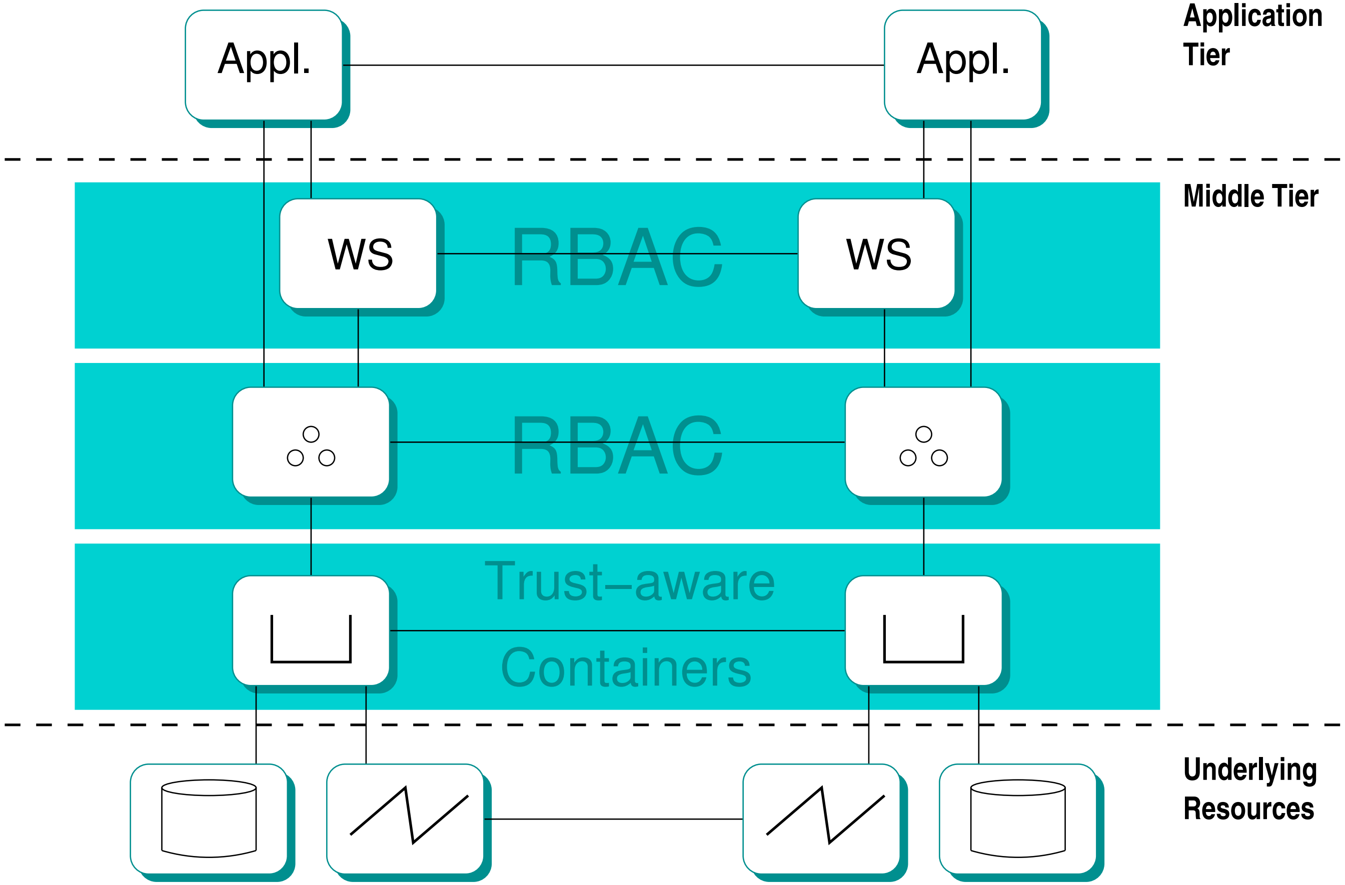
Trust-aware Component Execution Environment

Containers should provide components with:

- prevention of interferences between components belonging to different applications;
- enforcement of an access control policy between components of the same application;
- privacy mechanisms in components interaction protocol.

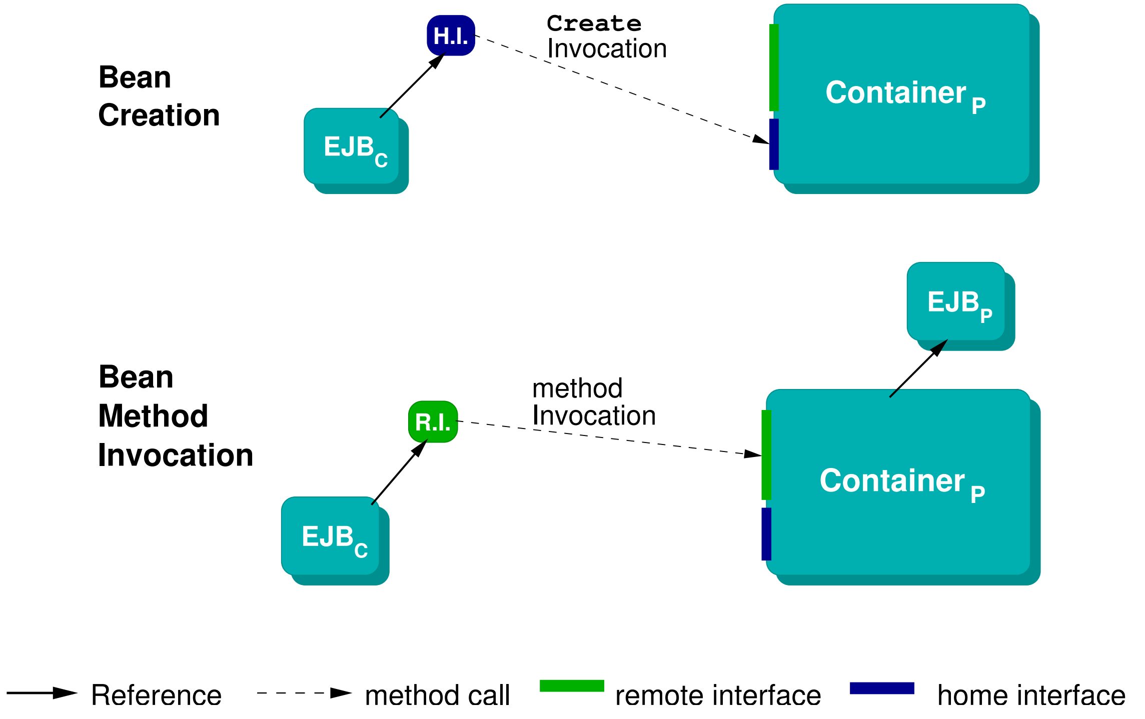


Trust Architecture in Tapas



Bean Interaction

A Container is a a Skeleton that receives `create` calls to instantiate beans and bean specific method calls.



Trust-aware Containers

Non-interference among unrelated components and privacy are achievable by introducing security technologies in component interaction protocol:

Non-interference: two applications execution environments must not share components:

- Containers must be aware of the domains which they belong:
  - application domain;
  - security domain;
- Authentication session before accessing the container.

Privacy: components interaction protocol must hide reserved data:

- Encrypted communications provides components with privacy.

Setting up a Trust-aware CEE

At configuration time the ASP:

1. Authenticates to its partners;
2. Delegates control over negotiated resources to target application domain (i.e., configures component execution environment).

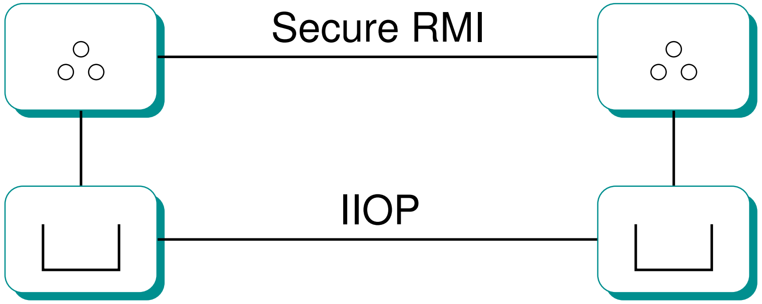
To call a method on a remote bean, a running application:

1. Authenticates to the container, giving a proof of its rights in accessing the container;
2. Performs the call.

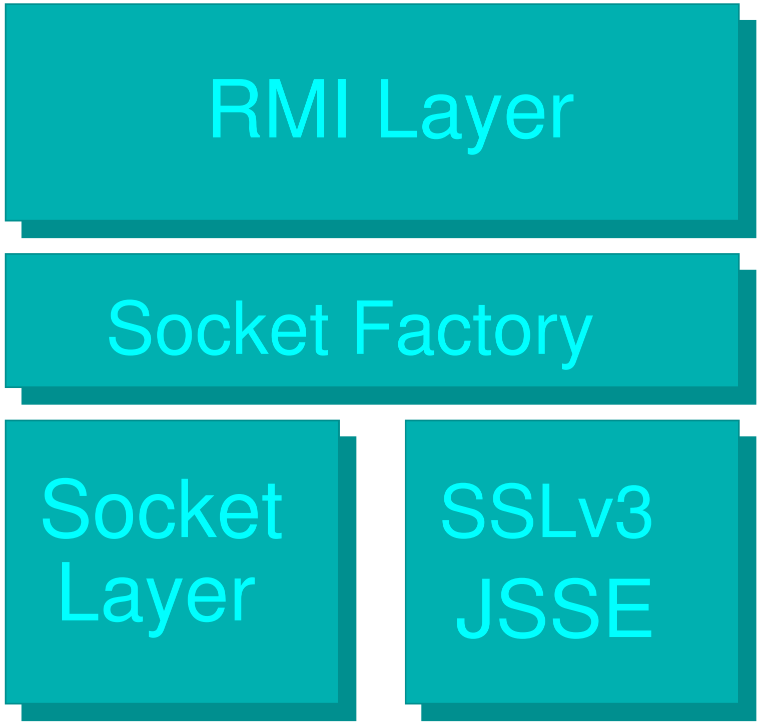
Secure RMI

Non-interference and privacy are achievable by integrating authentication and encription in component interaction protocols

1. Implementing RMI (over IIOP) over SSL (easier);
2. Extending RMI (over IIOP) with GSS (General Security Services).



Secure RMI (SSL)



RMI Layer

Socket Factory

Socket Layer

SSLv3 JSSE

Future work:

- Understand if SSLv3 is enough for meeting TAPAS trust requirements;
- If needed, understand how to extend RMI to meet trust requirements using GSS APIs;
- Does IIOP need to be secured?

References:

1. Sun Microsystems J2EE 1.4 Platform Specification
2. Sun Microsystems J2SE 1.4 Documentation and Specification
3. RFC 2246 Transport Layer Security v1.0
4. RFC 2853 General Security Services - API
5. D. Lamanna, J. Skene, W. Emmerich, SLAng: A Language for Defining Service Level Agreements