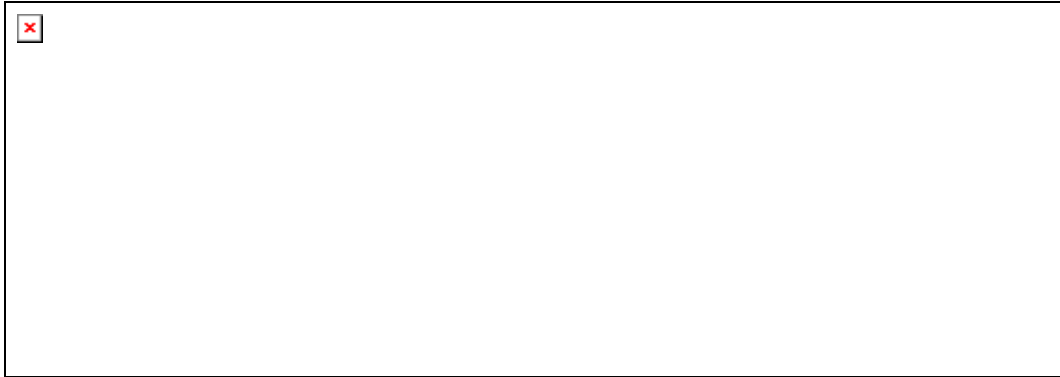


NEXOF-RA

NESSI Open Framework – Reference Architecture

IST- FP7-216446



**Open Architecture Specification Process
Open Construction Cycle #2**

Security Area

Topic: Multilevel Security for SOA

**Thales Research and Technology (UK) Ltd
Position Paper**

Contact: Adrian Waller

Date of publication: 13th October 2009

This work is licensed under the Creative Commons Attribution 3.0 License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.





1 DETAILS ABOUT THE CONTRIBUTORS

Main Contact / Lead contributor

- Name: Adrian Waller
- Affiliation: Thales Research and Technology (UK) Limited
- Email: adrian.waller@thalesgroup.com
- Phone: +44 118 923 8304
- Mobile Phone:

Additional Contributors (if any)

- Name: Arnold Yau
- Affiliation: Thales Research and Technology (UK) Limited
- Email: N/A
- Phone: N/A
- Mobile Phone:

Projects Represented

- Project: N/A
- Programme: N/A



2 MULTILEVEL SECURITY FOR SOA

2.1 Problem statement/Challenge addressed

Multilevel Security (MLS) for Service Oriented Architectures (SOA). Service-oriented architectures are dynamic, flexible and compositional in nature. Security is a significant challenge in a multi-domain environment.

2.2 Background

Thales Research & Technology (UK) Ltd. (TRT(UK)) has been working in MLS research for many years, for UK government and MoD agencies and for the EC. Some notable examples include:

SUPHICE (Secure Unplanned Provisioning of High Integrity Communications for Europe) was an EU-funded research programme (Preparatory Action for Security Research (PASR) 2004) aimed at developing technologies to allow rapid and secure deployment of sensitive networks spanning multiple international jurisdictions. (See also later).

MOSA (Modular Open Systems Architecture) for Naval Combat Systems for UK MoD, proposing a preferred architecture and supporting processes which deliver benefits to MOD and Industry.

From our work, it is apparent that traditional approaches to MLS, based on network layer or physical separation, are inadequate for the needs of more dynamic and service-oriented development. These traditional approaches are static and inflexible, and make it difficult or impossible, to share data effectively between services. More dynamic and flexible protection policies and associated mechanisms are needed. Ideally, protection needs to move up towards the application layer, where services and their associated data reside, to allow the benefits of data sharing and dynamic composition that Service Oriented Architectures (SOAs) provide.

2.3 Approach proposed (for solution)/Details

2.3.1 SUPHICE (EC PASR 2004)

SUPHICE, built on web services standards, is a distributed architecture for policy-based network level authorisation. SUPHICE networks can discover other domains available for connection through a services registry. In order to support deployment across multiple administrative controls and legal jurisdictions, each connection requires authorisation both at the initiator's local domain as well as the target domain. Once authorisation is granted, the two domains may then connect, creating a network secured by cryptographic network gateways using specified parameters (algorithms and keys).

At the heart of SUPHICE is a policy-based authorisation engine running on an authorisation server. Typically, one authorisation server is present in each administrative domain with access to its own policy repository. The SUPHICE policy engine uses a flexible, expressive, natural language-like syntax to define rules for connections based on a wide variety of attributes such as network



identifiers, sensitivity levels, proposed cryptographic parameters etc. The outcome of a policy evaluation is one of accept, reject, or refer to a human operator. A connection may only be initiated with two accept decisions, one at each end, either automatically generated or manually granted.

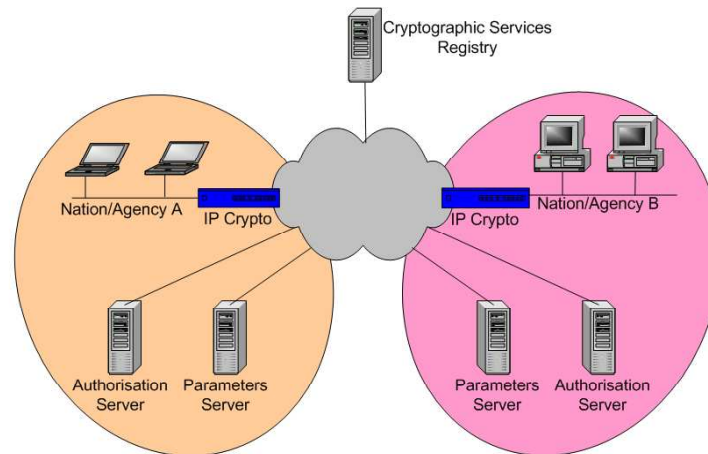


Figure 1 - SUPHICE Architecture

The SUPHICE authorisation server creates an audit trail over the course of its normal operation, providing a record of events, authorisation requests and decisions. The audit data allows investigation of past activity if a breach of policy is suspected.

A demonstrator was implemented by TRT(UK), and has since been refined. It is an example of service-oriented security, and the components and concepts developed are readily adaptable to SOA since SUPHICE runs on the Java EE platform and makes use of web services standards including UDDI, SOAP and XML. It can be easily enhanced by integrating web services security standards.

To incorporate MLS for services, each SOA service provider would attach a security level to all services on offer, the details of which may then be published on the services registry. On the SOA service provider's platform, the provider can define its own access rules and evaluation outcomes. Typically this will include the authenticated client's clearances for security level and need-to-know, as well as the provider's own security policy and local legislation restrictions. The policy decision may specify the level of protection of the service in the context of its classification. Differentiated protection can be achieved by using cryptography of various strengths.

2.3.2 OBSCURE[®]

OBSCURE[®] is a technology developed and patented by TRT(UK) that ensures data is only accessible to intended data consumers as specified by data producers. By providing application-level data security for services, OBSCURE[®] is an ideal complement to SUPHICE, which can mediate service-level access.

The main features of OBSCURE[®] are:

- Cryptographic protection at the application layer to items of data to give:





- Strong protection independent of storage medium or communication channel
- Protection at rest or in transit
- Cryptographic attachment of metadata to protected data items to provide:
 - A means by which content can be searched
 - Protection of data throughout its life-time
 - Permanent association of security attributes (e.g. sensitivity level, policies etc.)
- Controlled access using policy-based authorisation so that:
 - Producers maintain control over which users (e.g. services) have access to content
 - Unauthorised users can only read the metadata
 - Portions of data may be further restricted by nesting containers within containers

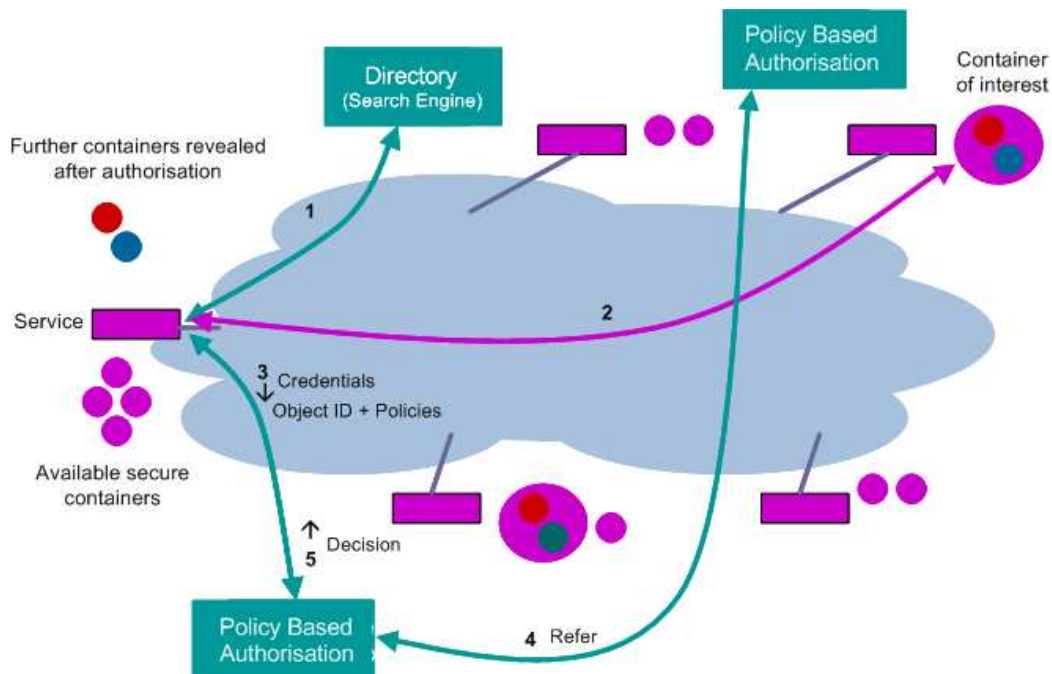


Figure 2 – Using OBSCURE®

The high level architecture of OBSCURE® consists of content producers, data consumers and a set of authorisation servers. In a typical OBSCURE® usage, a content producing service wishes to securely distribute sensitive data to a small group of consumer services. Using OBSCURE® the producer cryptographically protects the data and encapsulates it within a digital container along with metadata containing security attributes and keywords. The producer places the container on an access server and the metadata allows it to be indexed, searched and downloaded by all services in the community. While any service



will be able to identify a container and download it without restriction, only services specified by policy will be able to use the content after authorisation by the OBSCURE[®] authorisation server.

Uses for OBSCURE[®] include:

- MLS data distribution service: multi-level distribution of sensitive data between systems that would normally be separated onto different secure networks
- Secure data storage service: persistence of data containers in any storage medium

TRT(UK) has implemented this technology in several demonstrators and an industry trial, including Secure Situation Awareness: real-time data sharing between multiple emergency services, journalists and other interested parties that are allowed different levels of access. The implementation makes use of web services standards, including SAML, WS-SecurityPolicy and WS-Trust.

2.3.3 Our Approach

We propose an MLS SOA solution based on SUPHICE and OBSCURE[®]. With SUPHICE, in order to support MLS connection, services would be augmented with sensitivity level and compartment labels, and UDDI service registries would reflect this sensitivity information. Mandatory access control would be built into the policy authorisation engine to ensure services are only accessible to those with sufficient clearance. User interfaces would support the MLS policy definition, manual authorisation, security alerts and so forth. The SUPHICE architecture could also support differentiated levels of protection based on sensitivity labels. Data security services based on OBSCURE[®] would complement SUPHICE-based secure service connection. Secure data storage and transfer, would both be provided by encapsulation of data within cryptographically secured containers. MLS capability is inherently supported by OBSCURE[®] by declaration of the content's sensitivity level in its container's metadata field. The OBSCURE[®] authorisation server ensures that the decryption key is only released to services with sufficient clearance.

SUPHICE and OBSCURE[®] could also be integrated with other WS-* standards into a fully functional, user-centric, MLS service architecture. In particular, there is a level of overlap between SUPHICE and OBSCURE[®] authorisation services, allowing the processes to be streamlined or use of Federated Identity Management to enhance their efficiency.

2.4 Dependencies

This contribution does not depend on present work by TRT(UK) or other topics in this call, although it has some relevance to "Dynamic Security in SOA".

3 SUMMARY

We propose a multilevel secure SOA architecture by enhancing existing web services standards with the following:



- SUPHICE providing MLS service labelling and discovery, and fine-grain policy-based service authorisation based on service credentials
- OBSCURE[®] providing application level data security by means of cryptographically protected, security-labelled containers

We believe this is in line with the proposed scope and baseline of the topic.



REFERENCES

- [1] S. Butler, SUPHICE and the Use of Web Services, TRT Whitepaper NET060702, 2006. Available from <http://www.thalesresearch.com/LinkClick.aspx?link=NET060702.pdf&mid=1199>
- [2] R. Craddock, Secure Situation Awareness using Web Based Mashups, TRT Whitepaper VCS070602, 2007. Available from <http://www.thalesresearch.com/LinkClick.aspx?link=VCS070501b.pdf&mid=1329>
- [3] SUPHICE project website, <http://www.suphice.com>