Chapter 5 – CWID 2006 DEMONSTRATOR AND REALIZATION

The previous chapter described the general scenario of the demonstrator created by the group. Based on this description, the current chapter introduces its realization as it has been deployed at CWID 2006. The abstract view of the architecture consists of a set of software components for each nation. This set of software components supports secured Web Services exchanges in both publish-subscribe and request-response modes.

The high level architecture proposed by the working group was implemented in CWID as it is described in Figure 5.1.

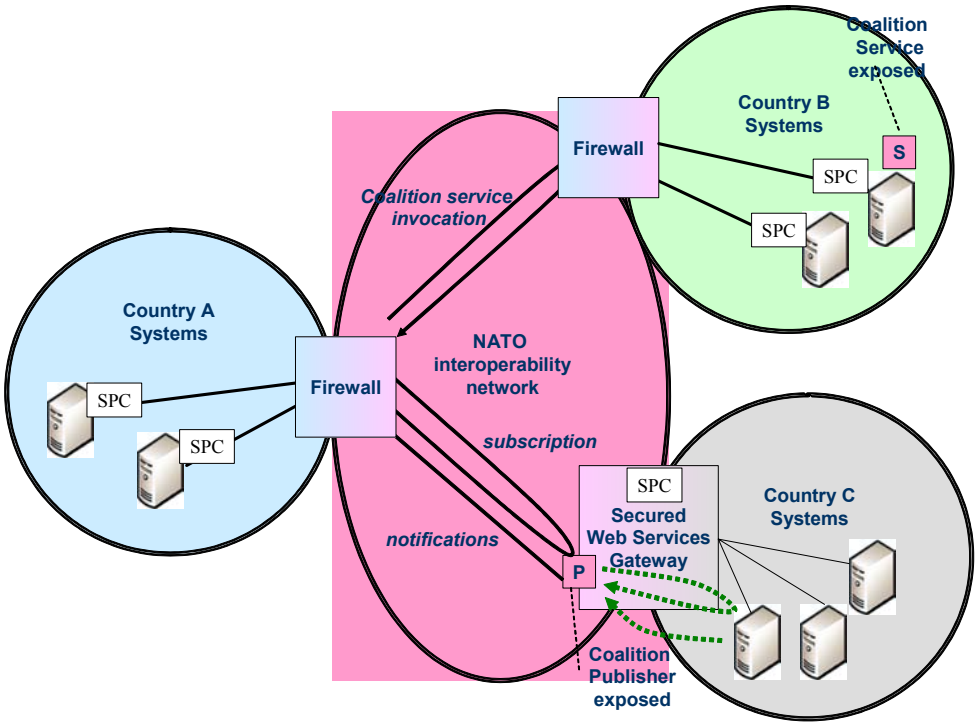

Figure 5.1: Demonstrator General Overview.

Each nation was connected to the CWID purple WAN. Norway and France were service providers and consumers. Poland, Spain and NC3A were service consumers as it was mentioned in the previous chapter.

5.1 GLOBAL ARCHITECTURE

The global architecture consists of the following "building blocks", each of which will be described in the following section. They are:

- Service registry;
- LDAP;
- SPC; and
- Publish / Subscribe mechanism.

5.1.1 Service Registry

The service registry is based on UDDI (Universal Description, Discovery and Integration).

During the trials, each country had the ability to discover new services thanks to the service registry. The service registry was itself a service provided by Norway during CWID.

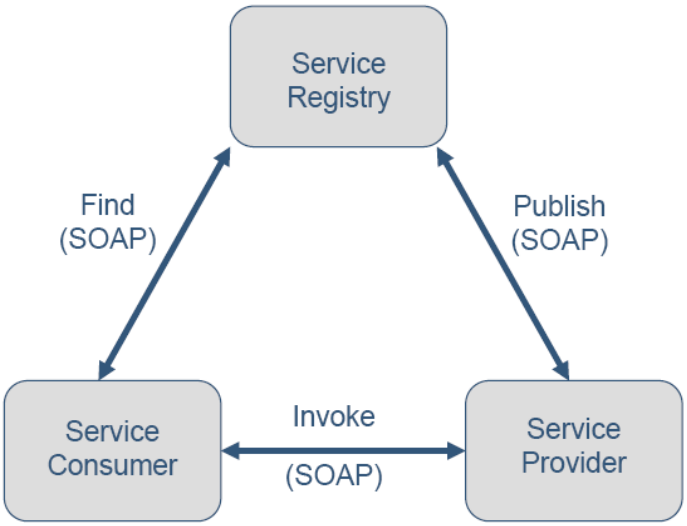

Figure 5.2: Service Registry.

UDDI V3 was chosen for the trials. During CWID, Systinet UDDI V3 COTS was used.

5.1.2 LDAP

The LDAP (Lightweight Directory Access Protocol) directory allows PKI exchange between Nations. The directory relies on the OpenLDAP software.

LDAP is synchronized directly between different nations. The directory synchronization tool was provided by Thales France.

The LDAP in each national demonstrator is directly used by the Security Protection Component (SPC) for authentication as shown in Figure 5.3.

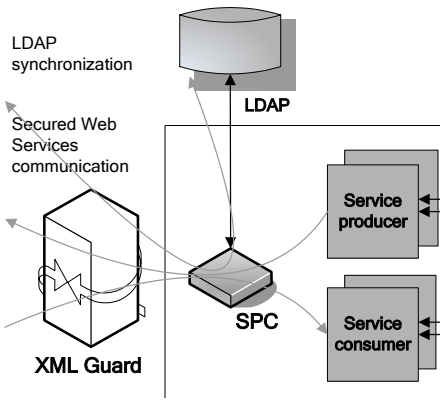

Figure 5.3: LDAP Interaction with SPC.

5.1.3 Security Functionality

The security functionality is placed in the end systems or in a Secured Web Service Gateway.

In Figure 5.4 below, you have a representation of different types of possible architectures. For example, Nation 1 has an SPC for each Service visible by other nations while Nation 2 has a unique Secured Web Services gateway integrating security. The implementation of security is specific for each nation and will be described later in this chapter.

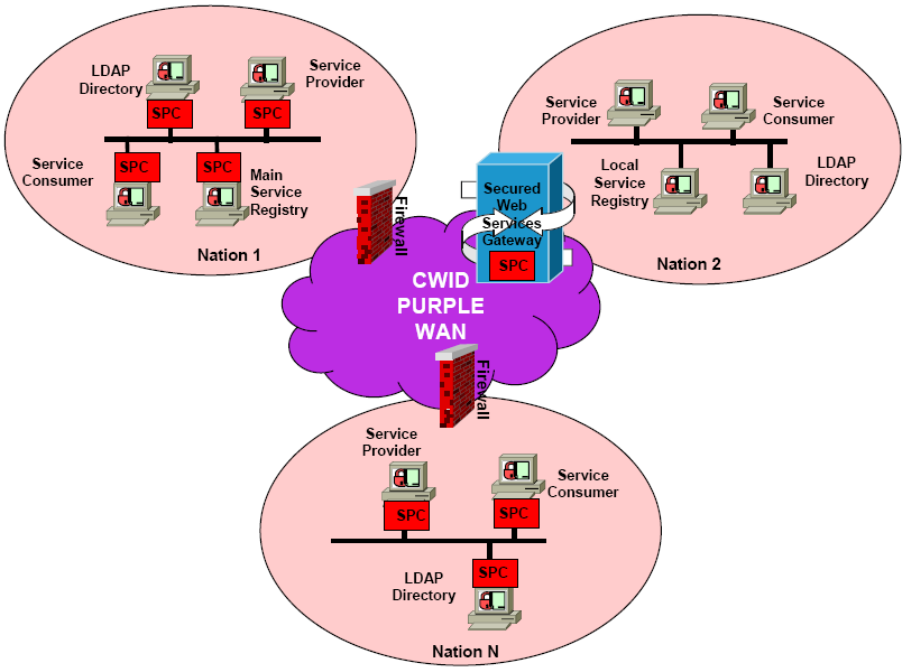

Figure 5.4: Security in the Global Architecture.

5.1.4 Publish / Subscribe Mechanism

The publish / subscribe mechanism used in the demonstrators is based on the Ws-Notification standard.

During the trials, each country had the responsibility to implement Ws-Notification. It will be described more precisely later in the chapter for each country (particularly for the service providers). Figure 5.5 below describes the general architecture for the use of publish/subscribe mechanism linked to the use of a service registry.

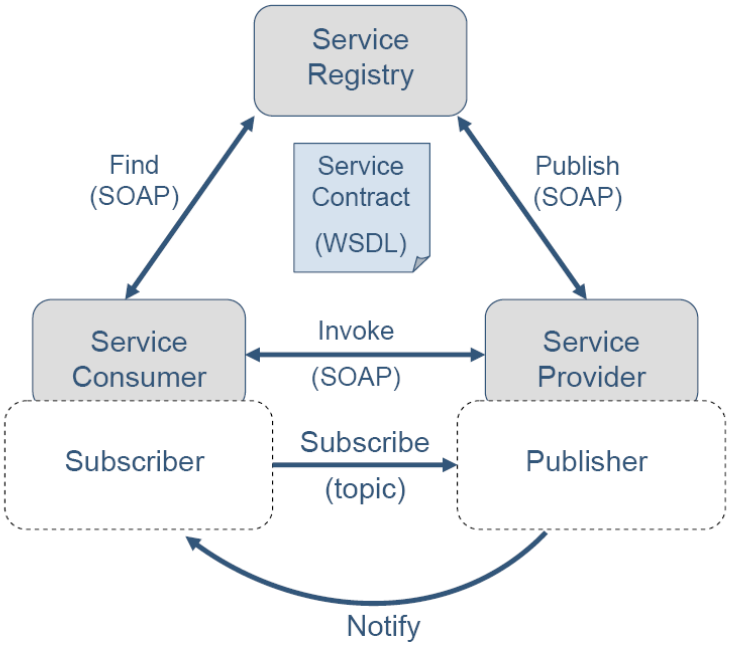

Figure 5.5: Publish/Subscribe Mechanism.

5.2 PARTICIPANTS' DEMONSTRATORS

This section describes the individual demonstrators from the participating nations.

5.2.1 EADS (France)

The EADS part of the French demonstrator represents a platform as described in Figure 5.6.

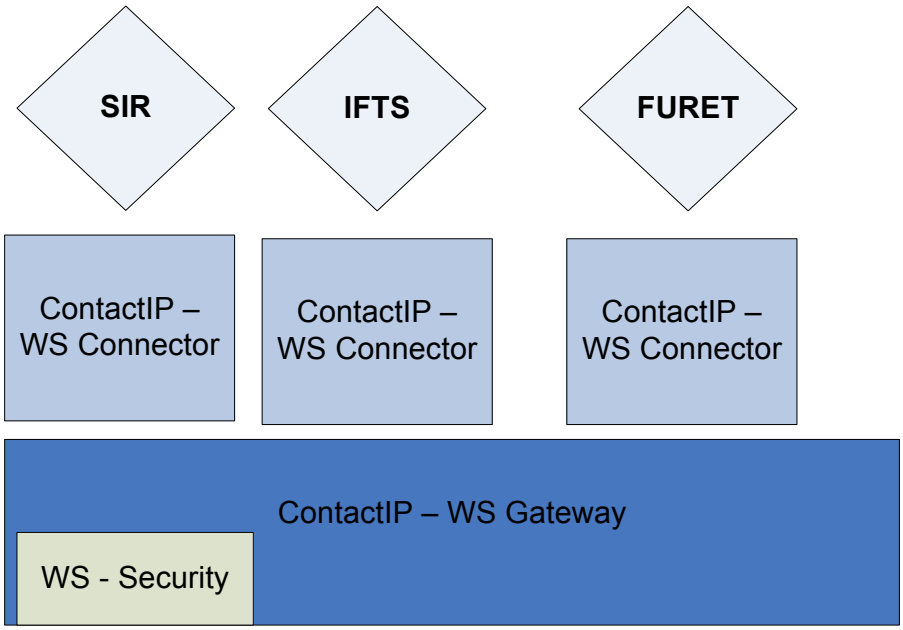

Figure 5.6: Demonstrator Architecture.

This architecture permits to include legacy systems such as:

- IFTS: Interim Force Tracking System deployed for NATO in Kosovo,
- SIR: "Système d'Information Régimentaire", the French regiment level system, and
- FURET: French production chain of tactical intelligence,

within the SOA platform.

This platform is composed of:

- "Contact IP – WS Connector" that allows implementing specific features to wrap existing systems to these new architectures needs.
- "Contact IP – WS Gateway" which is the main block of this architecture. It allows request/ response, publish/subscribe (WS-Notification) mechanisms.
- "WS Security". This block wasn't developed by EADS. As all the Web Services developed rely on WS standards such as WS-Notification and as EADS had decided to put more effort on web service implementation than on security, the demonstrator used the SPC developed by Norway.

5.2.1.1 Demonstrator Architecture

For the demonstration purpose, we chose to use IFTS' Imp@ct information system to test the whole ContactIP-WS architecture at CWID.

The demonstrator was composed of

- IMP@CT:
  - To visualize MIP situation provided by any nation, and
  - To visualize MTI tracks provided by any nation.
- Simulator:
  - To simulate movement of units that will be provided as changing MIP situation.
- "ContactIP – WS Gateway" based for this demonstration on Globus Toolkit:
  - That allows WS-Notification, and
  - With an additional development provided by Norway for WS-Security.
- "Contact IP – WS Connector:
  - That provides MIP tactical situation,
  - That is client to MIP situation, and
  - That is client to MTI tracks.

Figure 5.7 provides a very simple example of how the different elements of our demonstrator interact with other national systems:

- To provide a MIP situation to all authorized nations; and
- To get MIP situation from Norway and Poland.

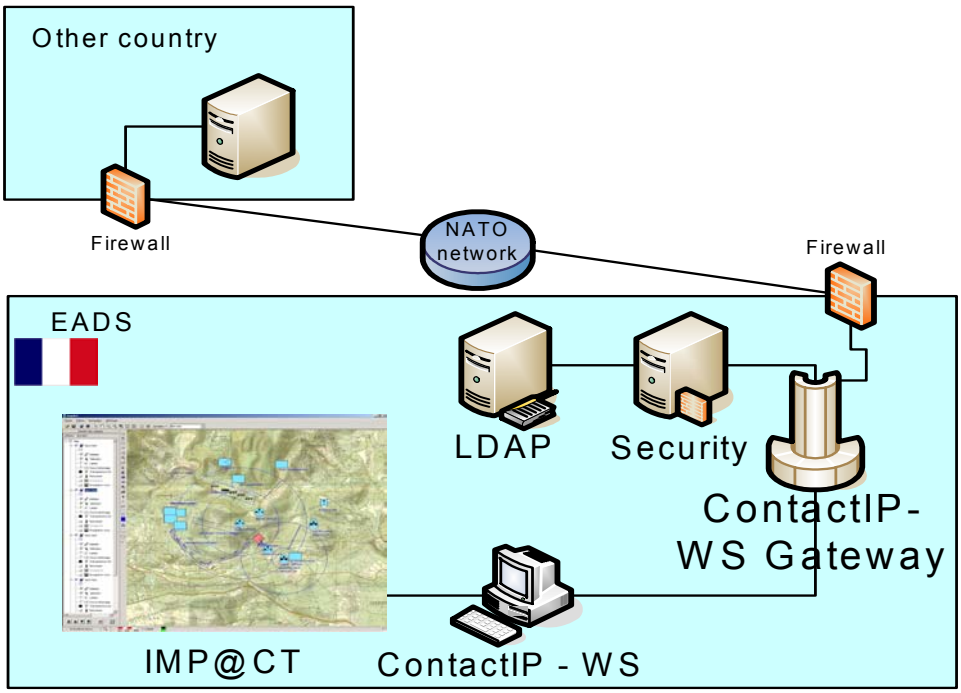

Figure 5.7: Demonstrator Interoperability Example.

5.2.1.2 Demonstrator Implementation

The demonstrator was created in an incremental development:

- Adaptation of the ContactIP framework: creation of the WS gateway;
- Implementation of a connector for legacy systems in conformance with standards to allow use of these web services in any infrastructure;
- Development of the security component; and
- Implementation of subscribing/unsubscribing for the UDDI registry.

As the time for development became shorter, we finally only made the 2 first steps of the demonstrator before CWID.

5.2.1.3 Interoperability before and during CWID

Before CWID, EADS provided a C2 Web Service to all other nations on Internet. That was an easy way for EADS to test all the ContactIP-WS Connectors created, and for other countries such as Poland or NC3A to test their implementations before moving to Lillehammer.

During CWID, EADS provided a C2 Web Service to all participants. Complete secured interoperability between IMP@CT-C2 Web Service and Polish and Norway systems were tested with success. The situation provided by IFTS-Imp@ct was visible on NORCCIS and on the Polish C2 Demonstrator.

CWID was also the opportunity for EADS to test its IMP@CT-C2 client by connecting to NORCCIS and to test its IMP@CT-MTI Client by connecting to the Norway Compilation Demonstrator which are described later in this chapter. The complete secured interoperability between Norway and the EADS clients were tested with success. The situation was viewable on the 2 EADS systems.

5.2.2 Thales (France)

The Thales part of the French demonstrator represents a Platform composed of an IEG Gateway and an Application Infrastructure. The Gateway is the public part of the architecture whereas the Application Infrastructure is hidden. (See Figure 5.8)

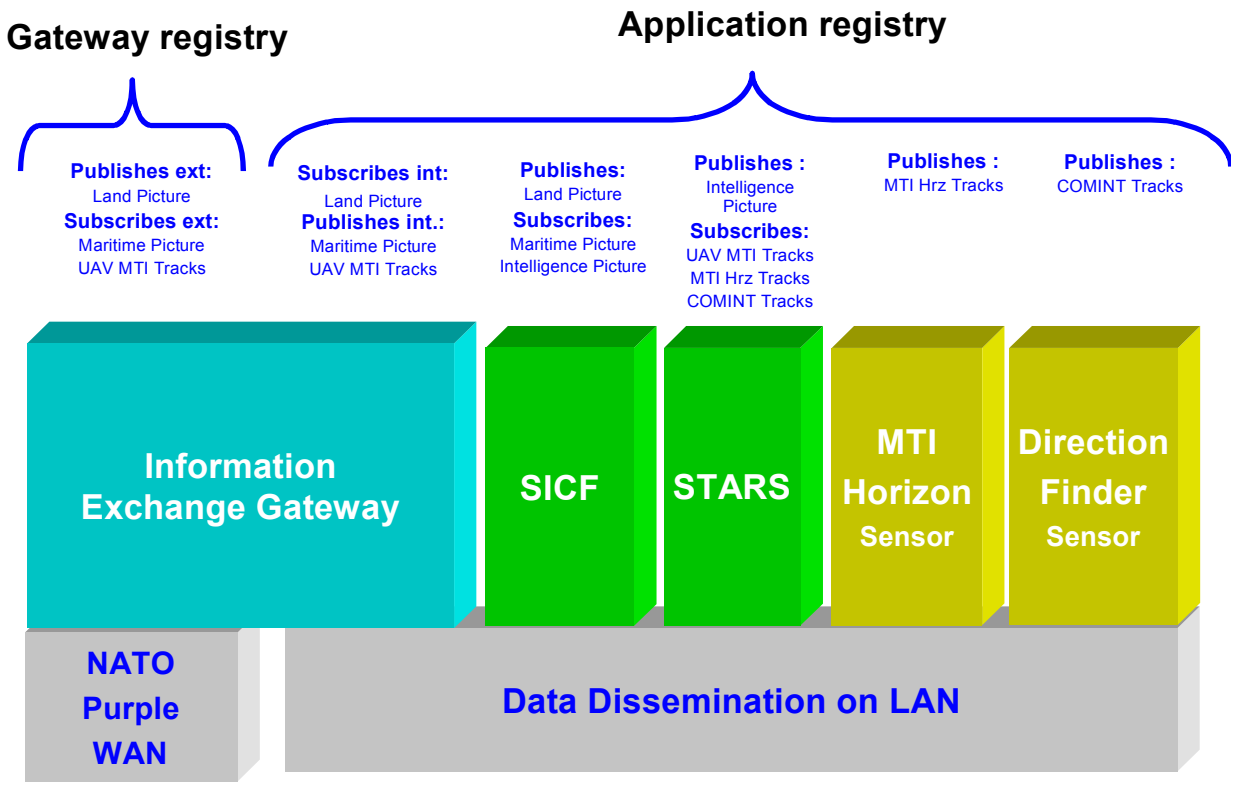

Figure 5.8: Thales Part of French Demonstrator.

The Main Components of the demonstrator are:

- An Information Exchange Gateway: This is a bridging component between the external world and the internal part of the Demonstrator.
- An Application Infrastructure comprising:
  - SICF: "Système d'Information et de Commandement des Forces" Command and Control Information System to display both land and maritime pictures and to generate the land picture;
  - STARS: "STructure D'Accueil pour le Renseignement et la Surveillance" Intelligence and Surveillance Framework for Tracks handling and fusion from different sensors;
  - MTI Horizon Sensor: Simulation of heliborne MTI radar tracks;
  - Direction Finder Sensor: Simulation of COMINT tracks; and
  - Data Dissemination Layer: Communication node distributing data within the LAN.

The demonstrator includes legacy systems such as SICF, STARS and Sensors. These legacy Systems have been integrated in the SOA architecture although they were not designed to be deployed in this way.

5.2.2.1 IEG

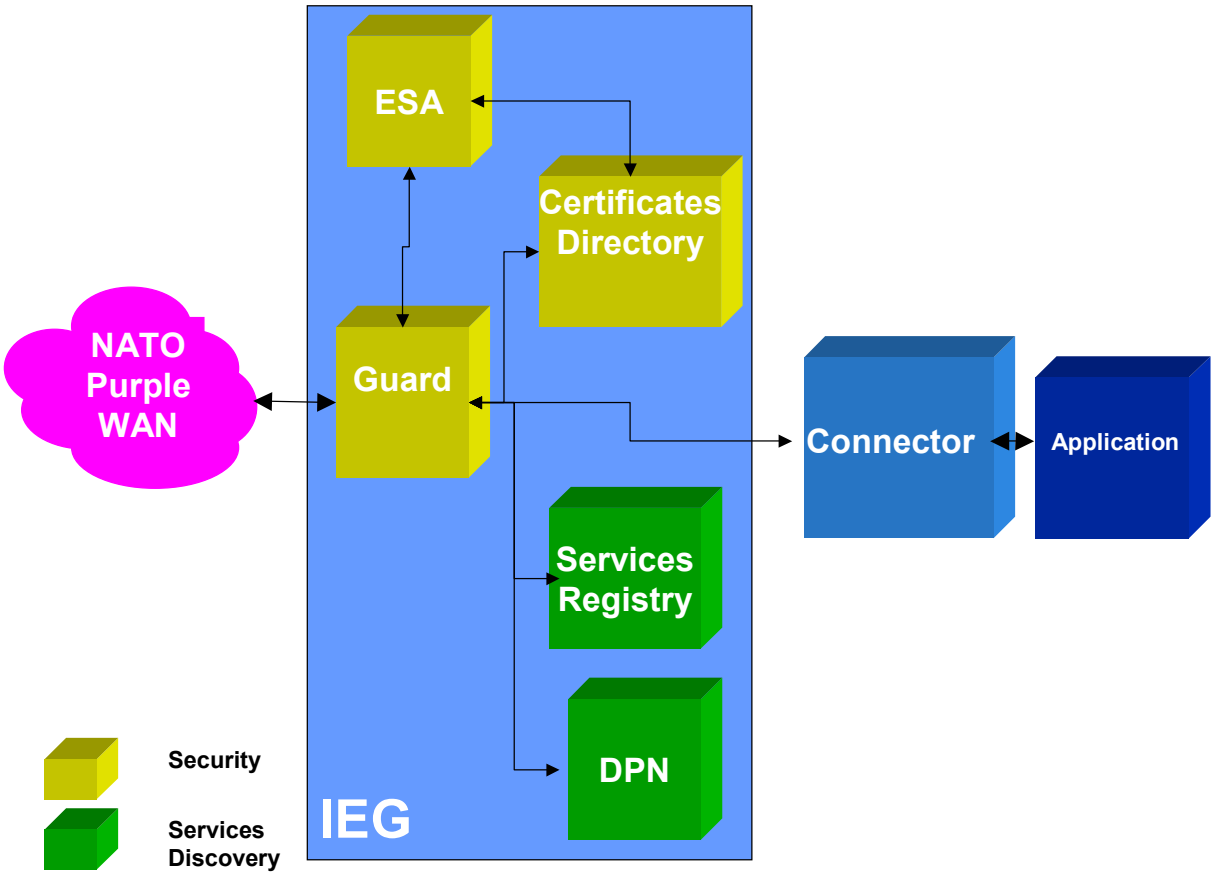

Figure 5.9: French (Thales) IEG.

The Gateway comprises services for information exchange with other members of the coalition according to two patterns: "Publish/Subscribe" and "Request/Response". Technical services in the IEG are:

- Information Assurance Services:
  - The Guard checks all outgoing information according to a Label with an appropriate security level and a signature. Allowed outgoing messages are transmitted. The integrity of incoming messages is verified;
  - The Encryption and Signature Authority (ESA) encrypts and signs outbound messages, and decrypts inbound messages coming from the Guard; and
  - The directory of Security Certificates provides Synchronisation Capability between Platforms.
- Discovery Services:
  - The registry of services that are exposed to the NATO Coalition; and
  - The Data Publishing Node (DPN) supporting the publish-subscribe mechanism for the Coalition Network.

The Connector bridges the application infrastructure with the Gateway.

5.2.2.2 Application Infrastructure

Behind the IEG, the Application Infrastructure relies on a Data Dissemination Layer where operational systems can provide and consume Data.

A user of the Data Dissemination layer can affiliate itself as a producer or consumer of a list of flows. Several of these flows are aggregated into dissemination channels. Each channel and each flow within a channel has a unique name. Here are some examples for possible channels with their respective flows from the present demonstration:

- Dissemination channel: C2 Channel:
  - Information flow: Land Picture; and
  - Information flow: Maritime Picture.
- Dissemination channel: ISR Channel:
  - Information flow: MTI Horizon Tracks;
  - Information flow: COMINT Tracks; and
  - Information flow: UAV MTI Tracks.

The connector within the IEG is a user of the Data Dissemination Layer. Hence it can transfer information between the internal part of the demonstrator and the external NATO Network. The connector is a subscriber of the national services (internally visible and exposed in the internal registry but hidden to the external world) through the Data Dissemination Layer. The national services are in turn published on the NATO registry as NATO services (externally visible and exposed by the Gateway Registry and/or NATO registry, but hidden to the internal part). In opposite operation, the connector is a publisher on the NATO registry of services that it has previously subscribed to on the Data Dissemination Layer.

5.2.2.3 Demonstrator Capabilities

The French Thales Demonstrator has mainly interoperated with the Norwegian demonstrator. From an Interoperability point of view, the demonstrated capabilities have been:

- Browsing and Discovery of Services in NATO Registry;
- Providing the Land Picture provision service. Publishing the existence of the Land Picture service provider in the NATO Registry;
- Subscribe to the Maritime Picture provider via address received from the NATO Registry;
- Send a Sensor Request to the Norwegian Demonstrator;
- Subscribe to the UAV MTI Tracks provider using address received from the NATO Registry;
- Exchange XML-based messages using the security mechanisms;
- Conversion of XML Messages between MPIA format (Modèle Pivot Inter-Armées: Standard of French Armies) and C2IEDM XML format; and
- Security Certificates Directories Synchronisation.

5.2.2.4 Demonstrator Implementation

The demonstrator has required the development of several components. To reduce development overhead, several COTS components (commercial and Open Source) have been included. If required, a wrapping to

adapt them to the specifications of the demonstrator was created. Table 5.1 below lists some of these components with their wrappers.

Table 5.1: Components of the Demonstrator

Components	COTS Inside	Standard
Guard	XML-Security	WS-Security
ESA	XML-Security Xerces	WS-Security
Certificates Directory	OpenLDAP	LDAP
Services Registry	Systinet Registry	UDDI
DPN	Axis TOMCAT	WS-Notification SOAP
Connector	Axis TOMCAT	SOAP
Data Dissemination	–	Proprietary

The French Thales Demonstrator has a different implementation from the other demonstrators. For the interoperability, the other demonstrators rely mainly on a Norwegian component, namely the Security Protection Component (SPC), which has been developed by Thales Norway. The SPC uses a COTS component, namely the Globus Toolkit 4 (GT4), for the Interoperability based on Web Services. The SPC component is used by all participants (except by the FR Thales demonstrator). As a consequence of the use of the Globus toolkit, there was a Synchronous behaviour on the NATO Network due to the fact that the GT4 acknowledges each terminating transaction.

The intent of the French Thales demonstrator was to demonstrate a platform operating on a battlefield using a radio Network. This intent requires the use of an asynchronous Network. To reduce the effort required to integrate with the other partners, it was decided to keep the NATO Network synchronous and to wrap the Thales Demonstrator behind a component that bridges the synchronous and asynchronous behaviours. The structure of the resulting network is shown in Figure 5.10.

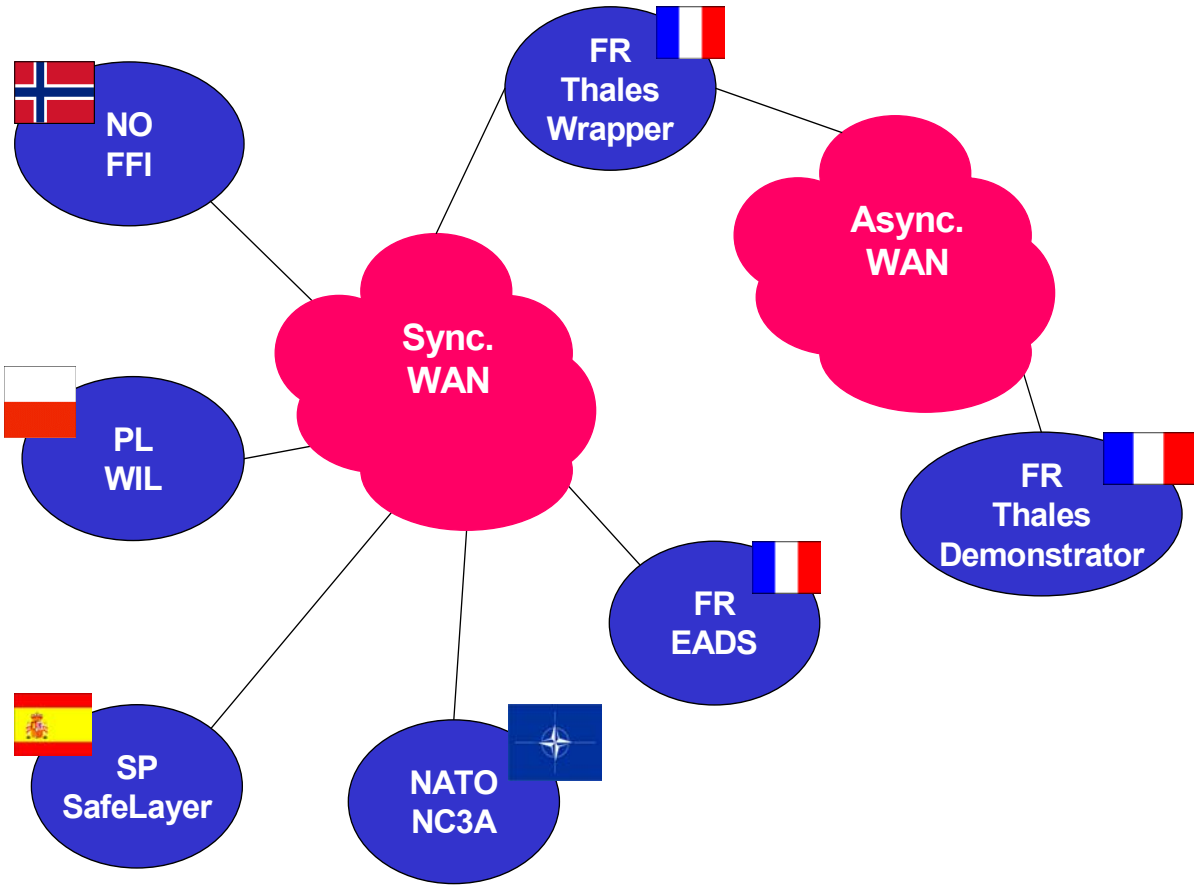

Figure 5.10: Asynchronous vs. Synchronous.

5.2.2.5 Demonstrator Flexibility

A main feature of the Demonstrator is its flexibility. From an external point of view, the Gateway exposes operational services that are dynamically added to or removed from a registry. The connector in the IEG allows bridging to the external world with another architecture, which may be service-oriented or a legacy one.

In the present case, the internal architecture relies on a data dissemination layer. Legacy Systems can plug in this layer through an adaptation connector. Systems can be easily added or removed. Additionally required development consists only of the Adaptation connectors and possibly a Data model.

This flexibility allows building generic components suitable to different platforms. Moreover the insertion of legacy systems, even legacy infrastructures, will be the future challenge for meeting NEC requirements in a progressive migration.

In a summary but not comprehensive, the Demonstrator has the following features:

- Adding/Removing of active services in operation;
- Adding/Removing of Systems providing/consuming services in operations when appropriate adaptation connectors are ready; and
- Insertion of legacy systems or infrastructures.

5.2.3 FFI / Thales (Norway)

This section describes the architecture of the Norwegian SecSOA Demonstrator at NATO CWID 2006. It has been developed by FFI, with substantial contributions from Thales Norway, especially on the security components.

The FFI Demonstrator is a distributed system consisting of several loosely coupled modules deployed across different physical machines. Figure 5.11 illustrates the CWID06 deployment of the Demonstrator. The Demonstrator is made up of the following logical parts:

- Simulation environment;
- A set of DPNs (Data Publishing Nodes);
- Service Registry based on UDDI; and
- Certificate server based on LDAP.

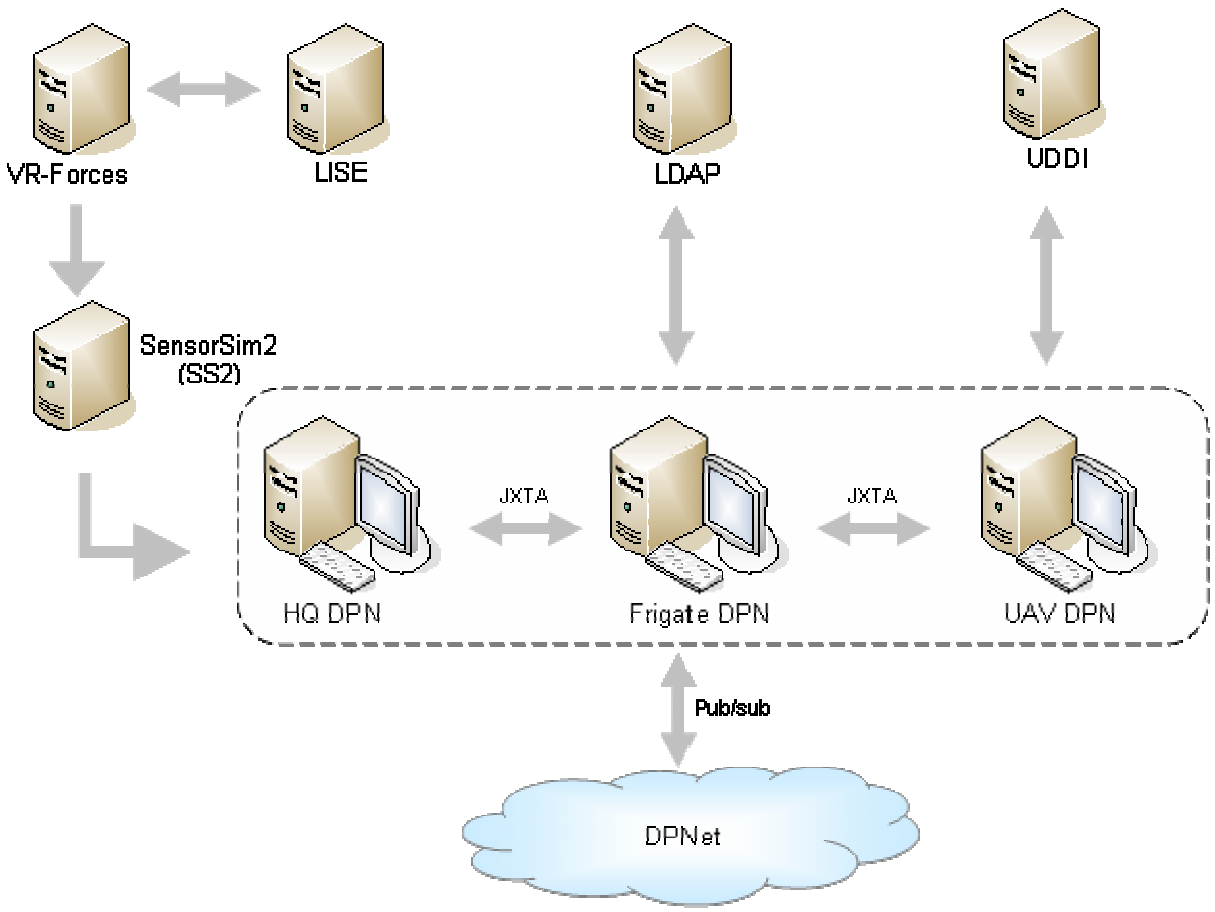

Figure 5.11: Demonstrator Deployment CWID 2006.

All components are interconnected in a national LAN, and reachable from the Data Publishing Network (DPNet), which represents the outside world that we interact with.

The simulation environment consists of three physical machines, VR-Forces, LISE and SS2. VR-Forces is a commercial application, and it requires a separate license server on the LISE machine, in order to run correctly. VR-Forces simulates object movements of both friendly and hostile units, and sends these to

SensorSim2 (SS2). SensorSim2 is an application which adds sensor simulations (e.g. radar simulation) to data coming from VR-Forces. SensorSim2 sends the simulated objects and sensor data to the DPNs.

Each of the DPN nodes represents a physical unit having a set of sensors attached to it, with an ability to communicate to other DPNs. Three national DPNs were used at CWID06:

- HQ DPN, which represents the national headquarters;
- Frigate DPN, representing a frigate; and
- UAV DPN, representing an unmanned aerial vehicle (UAV).

The internal DPN-to-DPN communication within the national domain is realized using the JXTA peer-to-peer technology. Each DPN also implements the Publish/Subscribe (pub/sub) mechanism. Publish/subscribe allows a DPN to communicate with other members of the DPNet, either by acting as a data producer or data consumer:

- A DPN can produce data to other DPNs; and
- A DPN can subscribe to and receive data from other DPNs.

At CWID06, the Demonstrator was integrated with the Norwegian Command and Control System NORCCIS II, enabling information exchange based on Web Services. Security functions in the form of a System Protection Component (SPC), was implemented in each DPN, and in the Service Registry. The SPC was developed by the Norwegian team, relying heavily on the contributions from Thales Norway. For CWD06, the SPC was made available to all participating nations for re-use.

The Service Registry is based on a commercially available UDDI registry from Systinet, with added functionality for security and extended search. It allows the DPNs to register their services so they can be discovered by other DPNs. At CWID06, the Norwegian Service Registry was centrally available for use by all nations participating in SecSOA.

5.2.3.1 End-to-End Security

Our implementation uses a combination of several security mechanisms in order to achieve the goal of end-to-end security at the information object level. The following bullets outline the security functionality developed as part of the demonstrator:

- All SOAP messages were attached with a security label, encrypted and signed;
- All advertisements in the Service Registry were attached with security labels and signed before storage;
- Before any notifications or UDDI records were released to a requestor, their security privileges were checked against the security label of the information objects; and
- A PKI and an LDAP Directory were used for providing the security infrastructure for exchange of certificates and certificate revocation lists.

As shown in Figure 5.4, all components that provide or consume services contained the SPC (System Protection Component). This component handled all parts of the security processing, i.e., performed certificate validation, created and validated signatures, encrypted and decrypted, and did access control based on the security labels. Thus, in our architecture security was handled in an end-to-end fashion.

5.2.3.1.1 SOAP Security

All information exchange was done using SOAP messages. The security of the SOAP messages was based on the use of the OASIS WS-Security standard with extensions in order to include an XML Security

Label. The security label (and other important fields) was bound to the SOAP message by a digital signature. The content of the SOAP messages was compressed, encrypted, labeled and signed before transmission. Upon arrival the security was validated and the originator was identified in order to see if the message came from a reliable source.

5.2.3.1.2 Securing the UDDI Registry

All records stored in the UDDI registry were labeled and signed in order to indicate their sensitivity and to protect them from being changed during storage. UDDI v.3 defines APIs for access to the data within the service registry. Two of these are the Inquiry API, which is used for searching for records, and the Publish API, which is used for insertion and updates of records. In order to secure these interfaces and enforce differentiated access control on the stored records, we used the System Protection Component (SPC) as part of the Security Abstraction Layer in front of the UDDI APIs. This security abstraction layer performed the WSS related security processing of the SOAP messages (authentication, signature handling and encryption), in addition to performing differentiated access control on the UDDI records based on the security labels of the UDDI records and the privileges in the user certificates.

5.2.3.1.3 Securing Subscriptions and Notifications

When the Notification Producer had a notification to distribute, the NotificationProducer matched the InformationSecurityLabel in the SOAP message of the notification against the User Privileges registered for each subscription. For each of the subscriptions for which it identifies a match, it released the Notification to the Subscriber associated with the subscription.

5.2.3.1.4 Security Infrastructure

The security infrastructure consists of a Public Key Infrastructure (PKI) used for handling X.509 certificates and CRLs, and a distributed LDAP system. Each nation had a Certificate Authority (CA) and trust was distributed amongst the systems via the exchange of the National PKI's Root certificates. The Root certificates were placed into a "trust file," which the systems used during digital signature verification. Only the PKI Root CA certificate was needed in the trust file because the CA certificate and the certificate of the system that performed the signature were available from the LDAP Directory together with the Certificate Revocation Lists (CRLs). Smartcards were used to store user certificates. The replication of the LDAP information was done periodically by exchanging LDIF files using Publish/Subscribe functionality. By subscribing to the periodical update of each national LDAP server (using the WS-Notification specification), the LDAP information replicated was protected by the SOAP security functionality. This also shows how a non-XML legacy system like LDAP may be included using Web Services technology.

5.2.4 Safelayer (Spain)

SP-ATI2 stands for Advanced Trusted-Information Interoperability and it is the formal title of the Spanish Demonstrator for CWID 2006.

This document describes the Spanish Secure SOA demonstrator at NATO CWID 2006, including the results and the lessons learned.

5.2.4.1 Demonstrator Description

The Safelayer' SP-*ATI2* demonstrator:

- Illustrated how to secure the data exchange between Nations using the new Web services technology in a trustworthy common environment;

- Allowed data exchange in a trusted environment, providing authentication mechanism, data confidentiality and the data integrity by means of using strong encryption technology; and
- Guarantied the interoperability and the global data exchange, establishing a *global trust environment between different Nations*.

For this purpose SP-ATI2 brought the concept of the PKI-Based Trust Service Provider into practice. The PKI-Based Trust Service Provider:

- Implements PKI-based security functions to Web applications demanding digital Signatures and data Protection, alongside authentication, authorization and audit functions; and
- Implements a full policy-based system in which the execution of a service is always guided by the rules of a concrete applicable service Policy.

Safelayer's TrustedX Web services Platform were used to implement a PKI-Based Trust Service Provider.

5.2.4.2 Demonstrator Architecture

As we have seen previously, the main purpose of the demonstrator is to show interoperability among different Nations using standard Web services in an application to securely interchange data. The chosen paradigm is a publish/subscribe schema in which a Nation will subscribe to different topics that another Nation will publish. All interchange messages will be secured in a standardized way. For more details see others chapters in this document.

The main goals of the SP-ATI2 Demonstrator are two fold:

- Technical interoperability among different implementations of standard Web services, namely, WS-Notification, WS-Security, SOAP, WSDL, etc.
- Technical interoperability in security aspects surrounding PKI mechanisms.

On the other hand, along side the aforementioned goals, it is also a primary target to test the technical interoperability of Safelayer product called "TrustedX secure Web services Platform".

Figure 5.12 shows the basic architecture of SP-ATI2 depicting all the participating components.

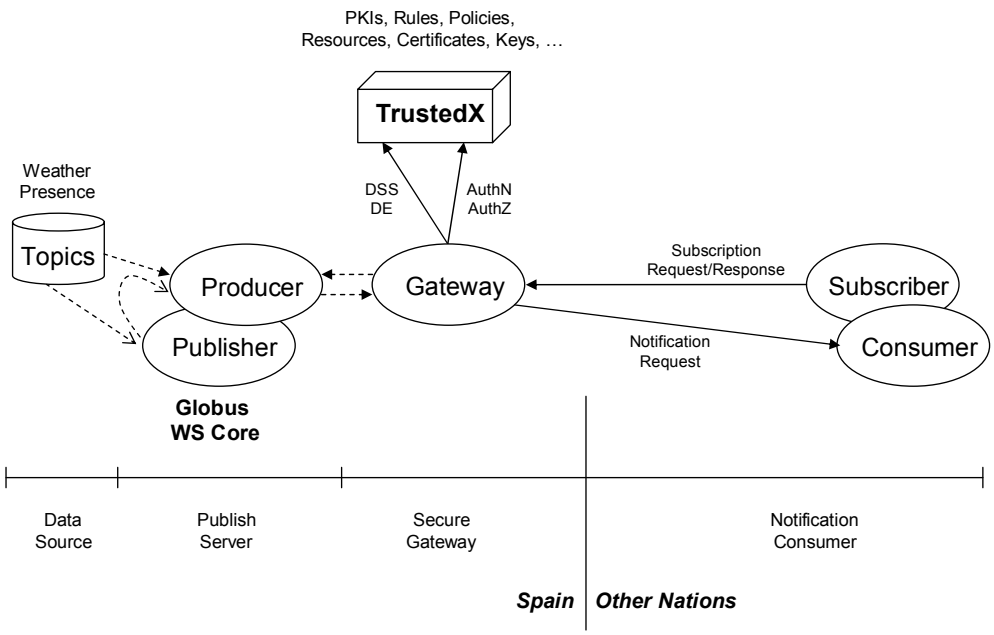

Figure 5.12: Architecture of SP-ATI2.

TrustedX is central in the performance of security services and also, very important, in the management. The idea is that TrustedX services are accessed by a gateway that intercepts the messages of the final Web service provider, in this case, a Notification service. In doing so, TrustedX will perform:

- The Authentication of the originating entity (the subscriber) when a subscription request message comes in, as well as the integrity of the message. For that, the SOAP message header containing a WS-Security compliant Digital Signature will be verified.
- An Authorization process (access control enforcement) so that the subscriber entity has the right to consume the subscription service in general, and if it has the right to subscribe to the requested Topic in particular.
- A Decryption process in order to decrypt the SOAP request message body.
- An Encryption process in order to encrypt the SOAP response message body.
- A Digital Signature generation process in order to protect the SOAP response message.

TrustedX takes care of unwrapping and wrapping the security elements of the request and response messages to/from the final provider service (Producer) and the consumer (Subscriber). However, and most importantly, TrustedX semantically verifies and checks the security elements following the locally defined policies. In this way, TrustedX provides to other components of the Service Oriented network a centralized container and processor of:

- All trusted PKI elements, that is, Certification Authorities, Validation Authorities, and Time Stamping Authorities.
- A policy-based mechanism to define both
  - i) Which are the allowed entities (Authentication Policies); and
  - ii) That have the right to access a set of defined resources (Authorization Policies).
- A policy-based mechanism to establish trust development in digital signatures (authenticity and non-repudiation) and the plethora of trusted third parties defined in the system (Digital Signature

Policies). Trust development is done in a semantic way, that is, a policy is applied to the digital signature in order to get a measurable Trust diagnostic by processing all security elements (including some environment-related elements, like time, IP addresses, authentication mechanism, etc.).

- A policy-based mechanism to protect data (Data Encryption Policies) so that data can be encrypted and decrypted taken into account security and environment-based elements.

TrustedX also provides a trusted point of robust security by centralizing sensible key material in different degrees as well as certificates, revocation information, etc. All accesses to information, whether sensible or not, are controlled by a policy enforcement mechanism that will require the right credentials to access the resource. Moreover, data communication is also secured by the right protocol, for instance, SSL.

It is also worth to shortly describe how the Gateway component works for outgoing Notification messages (from Producer to Consumer):

The Producer application (Globus Toolkit version 4) sees the Gateway as an HTTP Proxy, thus the Gateway will forward the protected message to the suitable end point.

The Gateway is configured to authorize the use of TrustedX services to a Producer application with a fixed IP address. Nevertheless, inner traffic is supposed to be friendlier than outer traffic.

The Gateway uses the SOAP message header wsa:To endpoint reference to select the right encryption certificate/key.

The Gateway signs the SOAP message using the name (X.509SubjectName) of the Producer application.

5.3 LIST OF TEST CASES

The test cases described below are the ones agreed by this working group:

1) Enhanced end-to-end WS-Security

   Show that all SOAP messages exchanged between nations are secured using PKI-based end-to-end object level security mechanisms.

2) Information delivery using Publish/Subscribe

   Show that services are made available to others by publishing, and that efficient delivery of updates is achieved by subscribing to an information delivery service.

3) New services made ready for use

   Show that a new instance of a known service interface, or a new service with a not previously defined data format, can be published and used.

4) COI Cooperation

   Show NetCentric cooperation between the C2 and ISR COIs using the object oriented MIP data model.

5) Access control at the object level

   Show that the information objects (WS-notifications or UDDI records) can be securely marked and that only users with the right security privileges can access/receive them.

6) Distributed Security Management

   Show that Certificates/user privileges can be issued or revoked, and evaluate the time that is needed until the change has been propagated to all nations involved.

7) Dynamic Service Replacement

   Show that a broken service can be automatically replaced.