Towards Secure and Trusted Collaboration Environment for European Public Sector

Adomas Svirskas, Jelena Isachenkova, Refik Molva
Institut Eurécom
Sophia-Antipolis, France
{adomas.svirskas, jelena.isachenkova, refik.molva}@eurecom.fr

***Abstract*—e-Business and e-Government implementations are becoming more and more widespread with growing number users depending on availability, accuracy and security of such e-Services. The users must be able to trust these services, otherwise they will be reluctant to embrace the new opportunities and will not be able to reap the potential benefits. In addition, the end users wish to use the e-services in the simplest way possible and to have them "on tap" 24x7 as other conventional utilities. For this to become possible, a robust interoperability fabric among the involved institutions needs to be established. This means having a lot of collaborative interactions invisible to the end-user (a business or an individual citizen) in order to fulfill the promise of e-Services. Such interactions become more complex when the organizations belong to different countries, act according different laws in different languages. This paper presents the work being done to create an efficient, secure and trusted interoperability framework for public sector agencies of European Union member countries.**

Keywords – e-Services; e-Government; SOA; Web services; interoperability; security; trust

I. INTRODUCTION

Nowadays the interactions among the partners of knowledge-intensive collaborations are often based on the Service Oriented Architecture (SOA) paradigm – the partners use each others' services. Such collaborative on-demand interactions take various forms depending on various factors such as complexity, scope, duration of the interactions, level of formalization and the application domain. Within the scope of this paper term *collaborative* denotes the type of interactions where the partners are peers, i.e. they do not have direct control over each other and communicate between themselves by exchanging among themselves mutually understandable messages.

The notion of peers is well suited to collaboration of the governments and other public service agencies of 27 European Union member countries – these organizations are independent, act according to the law of their respective countries yet have strong needs (and obligations) for interoperable interaction. For example, in 2004, in the Hague Programme [1], the European Council stated: "*The mere fact that information crosses borders should no longer be relevant. With effect from 1 January 2008 the exchange of such information should be governed by conditions (...) with regard to the principle of availability, which means that, throughout the union, a law enforcement officer in one Member State who needs information in order to perform his duties can obtain from this from another Member State (...)*". Currently, in European law enforcement domain work is underway to implement the principle of availability with respect to six categories of data: DNA, fingerprints, ballistics, vehicle registrations, telephone numbers and other communications data, and civil registers.

To achieve such interoperability and availability goals, in some cases partners collaborate according to a set of agreed business protocols (expressed as service choreographies) [2] while sometimes these collaborations take more ad-hoc shape. Quite often the overall collaboration setup is a mix of the two mentioned options, where the routine, repeatable tasks are being performed using protocol-based collaborations white more subtle and specific knowledge-dependent tasks are accomplished with the help of ad-hoc interactions. The latter quite often support the former – in order to complete a step of agreed business protocol a collaboration partner may need to carry out some unforeseen work and then use the results for the next step of the protocol. The interactions between the partners need to be fast and efficient, especially in the areas such as law enforcement, mutual legal assistance, terrorism prevention, disaster information exchange and similar.

Invariably, these interactions are subject to a number of general security requirements, which, from a stakeholder viewpoint, can informally and briefly be summarized as follows:

- Authentication:
  - How can the service provider be confident that the requestor is who they claim to be, and vice versa?
  - How can a service provider easily support the multiple types of authentication methods (for example, digital certificates, user IDs and passwords, and more)?
  - How can service consumer avoid authenticating themselves in several places, thus lessening privacy breach concerns and taking advantage of single sign-on benefits?
- Authorization:

- Is service consumer allowed to perform this transaction/access data?
- Is consumer required to reveal their identity/ Personally Identifiable Information (PII) to be authorised?

- Integrity: Is data sent by the sender remained the same when it arrived to the receiver?
- Signatures: Is it possible to create and verify an electronic signature analogous to a handwritten signature?
- Confidentiality: Can collaborating parties be sure that the data has not been seen by anyone else?
- Auditing: Can participants record the transactions at the data and control flow layer for subsequent verification on collaborations?
- Non-repudiation: How a sender or a receiver can legally prove to a third party (e.g., a judge) that the same data was sent and received in a transaction?

These security requirements may seem rather basic, however addressing them in the Web services-based technical context still requires reasonable effort from solution architects and developers side. Our work described in this paper aims to make a contribution towards establishing simple and effective security architecture for collaborative interaction solutions based on SOA principles and Web services implementations. This work is being done as a part of the R4eGov [3] research project, which aims to provide an innovative platform for interoperable interactions among European governmental agencies.

The rest of this paper is structured as follows. Section II introduces the overall solution architecture, Section III explains the security mechanisms, Section IV discussed access control mechanisms and Section V concludes the paper.

II. PROPOSED ARCHITECTURE

A. Business and Technology Context

Before presenting the proposed solution, it is worth to discuss the architectural context in brief. Having studied the business requirements of public sector collaborative interactions [4], [5] and current state of the art of possible implementation technology, we distinguish a number of factors, which determine the context of our solution architecture.

Firstly, there is a need for the public administration agencies (or their units) of the EU member countries to have standard ways for interconnecting and integrating their heterogeneous IS for achieving interoperability at shared information/knowledge level. The service oriented (SOA) approach to integration of the information resources, i.e. each information source being exposed via well-defined interface, following the DaaS (data as a service) principle. The concept of data as a service (DaaS) suggests using service-oriented architecture (SOA) for accessing data "where it lives" - the actual platform on which the data resides doesn't make crucial difference for overall collaborative interaction among the partners.

The architectural approach taken by R4eGov, based on service virtualization, is a sound practical foundation for implementing the DaaS [6] concepts in practice. Virtualization also allows uniform access to the software services exposed by the partners of collaborations. Virtualization and uniformity of services, provided by public administrations, are very important in order to have on-demand data aggregation, also referred to as enterprise mash-ups [7]. This relatively new concept of Web 2.0 has already found its place in e-Government: "The U.S. Department of Defense's lead intelligence agency is using wikis, blogs, RSS feeds and enterprise "mashups" to help its analysts collaborate better when sifting through data used to support military operations."

In a more traditional way, mash-ups can be perceived as composite views of data. They introduce an abstraction that separates applications and data, increasing the value of the data by making it accessible as a service for a larger base of business users. Naturally, appropriate access control measures are crucial for such data compositions to be applicable in corporate and government scenarios.

In our R4eGov solution architecture, service virtualization is implemented using the concept of application-level gateway - each participant of collaboration communicates with the peers via Web services based Interoperability (IOP) Gateway [8]. That is, the real services within an agency participating in the interactions, are accessible by sending a request to well known address of the gateway and specifying what kind of resource is needed. Gateway redirects such request to the internal provider, access control rules permitting. This pattern is not a new concept [9], Schmidt [10] defines a gateway as *a mediator that decouples cooperating peers throughout a network and allows them to interact without having direct dependencies on each other*. We have chosen this pattern and the newest SOA-based technologies to implement a lightweight, flexible and efficient interoperability platform, which will be explained below.

Each participant, with rare exceptions, at different points of the interaction can find itself at either the sending or receiving end of the information (SOAP/XML messages). In other words, in this asynchronous mode of interaction each participant is capable to receive requests from outside (other participants) to access its internal resources (data, services) as well as to initiate the requests towards other participants (or respond to their requests).

The latter case means that the internal resources (legacy/back-end systems) of a participant issue requests to the outside of the participant domain. Thus we have requests coming in and the requests/data coming out for each given participant multiple times during an instance (a collaboration scenario or business protocol, choreography) of collaborative interaction.

B. The Proposed Solution

Technically speaking, R4eGov SOA-based solution architecture primarily relies on Web services technology, including both the basic protocols/specifications such as

SOAP, WSDL, HTTP/S and the more advanced ones – WS-Addressing, WS-ReliableMessaging, WS-Security, WS-Trust etc. – collectively known as Web services advanced architecture [18,19,20]. Web services are used for both inter-domain communication between the gateways and the internal integration of back-end services provided by the partners. Internal services may represent some newly developed functionality or serve as wrappers for legacy systems.

It is quite clear, that implementing the IOP Gateway features, configuring gateway instances for operating in different environments and, in particular, ensuring appropriate security level, is not a trivial task – the gateway needs to map incoming messages to internal operations, support business rules, enact business protocols (choreographies), which govern collaborative interactions, enforce security policies, provide logging, and monitoring, at least – as , as depicted in Figure 1.

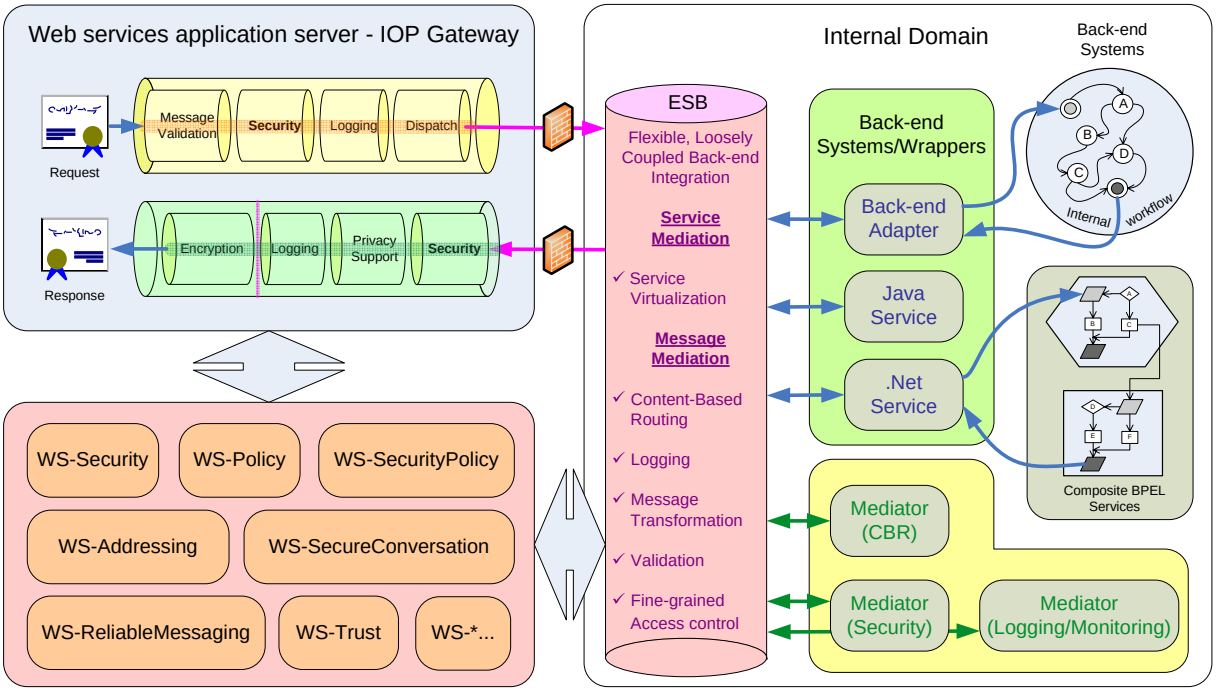

Figure 1 – Architecture of collaborative interaction platform based on interoperability gateways

To ensure manageability, adaptability and flexibility, functionality of the gateway needs to be decomposed and developed/deployed accordingly. Firstly, there is a need for *modularity* – each separate gateway function should be well-defined and of manageable scope. The R4eGov architecture [11], introduces the notion of *extension module*. Extension module is a software component that can be plugged into the execution environment and which fulfils a certain non-functional task in the context of collaborative workflows.

In addition, this architecture need to support *pluggability* – the extension modules should be ready for deployment by changing gateway configuration; no extra coding should be involved for registering, un-registering, changing order of invocation and similar administrative tasks. This helps achieving *flexibility*, as it should be possible to compose modules into flows (pipelines) for sequential message processing, separately for outgoing and incoming messages, specifying necessary order of message processing.

Furthermore, a principle of *isolation* needs to be observed, it is two-fold:

- The modules should be independent from each other, i.e. presence or absence of a module should not directly affect other modules, and no cross-invocation among the modules is allowed either. Modules can only indirectly affect each other in situations when message processing sequence is broken by absence/incorrectness of some data in a message (its header, more precisely) as a result of absence or malfunctioning of a specific module.
- The modules should only be invoked by their containers via contract interfaces, as foreseen in architecture of each container, no external invocations of the modules are allowed. Modules, of course can invoke the services and APIs they need.

Let's look how these principles are applied in R4eGov IOP Gateway architecture. The main functionality of the gateway – connecting inter-domain collaborative interactions with the internal services is achieved combining modern web services engines (e.g. Apache Axis2 [12]), which support pluggable message handler architecture, allowing to implement message processing chains in an elegant and efficient way. Furthermore, modern Enterprise Service Bus (ESB)-based mediation frameworks (such as Apache Synapse [13]) support extensions called mediators, which facilitate such message processing functions as:

- Content-based routing
- Message transformation

- Logging
- Security support
- Message schema validation
- Load balancing and fail-over
- Quality of Service support
- Protocol (e. g. SOAP, REST, JMS) and presentation format (e.g. POX, JSON, XML) conversion

ESB-based mediation framework can act like an intelligent yet lightweight and efficient application level router connecting the internal services providing access to the actual data with the external collaborative interactions. Such framework provides powerful means to define the rules according to which the messages are directed and handled.

As we can see, security-related extension modules can be implemented in two ways, depending on the context - they can be either Web services engine handlers or user-defined mediators in ESB mediation infrastructure. Such combination of Web service message processing handlers and ESB message mediators provides possibilities for composing very flexible chain of actions to be performed on a given message, which is crucial for IOP flexibility and usefulness. The implementations we have chosen, Apache Axis2 and Apache Synapse (packaged as WSO2 protocol stack [14]) are designed to work together, have solid developer and user base, which increases the chances of successful practical use.

III. Security Architecture

A. The Requirements and Scope

As it was stated above, any collaborative interaction framework attempting to address multi-domain interoperability issues in e-business or e-government area, must ensure appropriate level of security in order to be trusted by the users. Apart from the general security concerns listed in the Section I, one of the frequent questions asked by the collaboration participants is whether their partners will apply adequate security measures to the data handed over to them. In other words, will my partner protect my data as rigorously as I would do?

In a peer-to-peer collaborative environment, where data is transmitted among multiple partners it may happen, for example (Figure 2, left), that the transmitting authority applies higher security standards than the receiving authority, which consequently has to apply additional measures (which it does not usually apply to this type of information) in order to guarantee the same protection as the transmitting authority. Alternatively, if we reverse the roles (Figure 2, right), both parties will simply need to apply their own security measures (and the security measures applied by the recipient of the information will actually be stricter than the ones applied by its owner).

It is the *transmitting party* (information owner), which decides upon the confidentiality level of the information and it can change or remove this level. The receiving party can inform the transmitting party that this level should be adjusted.

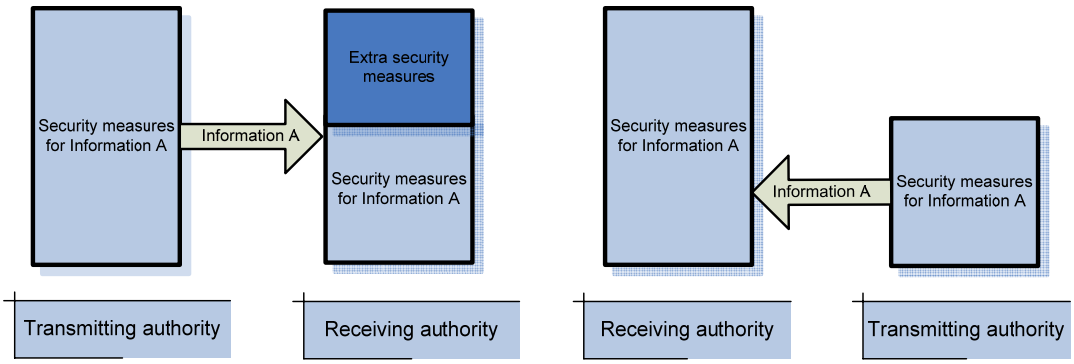

Figure 2 - Different levels of security measures during information exchange

These requirements essentially mean that R4eGov solution will need to provide information *protection policy harmonisation mechanisms*. In our security architecture we address this issue by proposing the collaboration partners to share a common set of distributed roles. Inherently, the access control rights will be transferable across the domain boundaries and harmonisation of the security measures will be quite straightforward. The solution of distributed roles support based on (and extending) XACML is proposed by Lee & Luedeman [15].

In order for this to happen there must be a level of trust established between the transmitter and the receiver. The former needs to be sure that the latter actually enforces the specified policies. There are several ways to establish such trust, for example implementing access control enforcement using trusted code – components and services. Djordjevic et al. [22] describe a method for combining software resource level security features offered by Web services technologies, with the hardware-based security mechanisms offered by Trusted Computing Platform and system virtualisation approaches. They propose a trust-based architecture for protecting the enforcement middleware deployed at the policy enforcement endpoints of web and grid services. Such approach can be used in conjunction with our distributed roles-based access control.

In addition to this, usage of sticky policies [] can further help to ensure that the information owner's preferences are

respected. Bandhakavi et al. [23] categorize information flow types between parties and propose a method in which super-sticky and declassifiable release policies for newly aggregated information can be derived from the original information's release policies and the local release constraints imposed by the creator who has aggregated the new information.

In a complex and heterogeneous ICT environment like the one R4eGov project faces, security requirements and expectations are often interpreted differently by different organizations and individuals, or simply specified in too-vague terms. For any security architecture related activity it is important to scope the area of security measures precisely, otherwise it is not possible to design and implement security solution in a timely and manageable manner.

In short, our security/privacy solution addresses the following security concerns:

- Confidentiality of information items transferred between the IOP Gateways of collaborating organizations – implemented using Web services and XML security specifications (WS-Security, XML Encryption/Digital Signature).
- Authenticity of the information items being transferred between the organizations. This is implemented the same way as confidentiality.
- Role based access control to the resources offered by one organization to another. This is done by the *target* organization, based on the set of *distributed roles*.
- Role and identity based access control to the targets of partner organizations. This is done by the source organization, evaluating for each outgoing request whether the subject is *entitled to be assigned a distributed (external) role in a particular collaboration* and thus have an access right to a *particular resource offered by target organization*.
- Protecting privacy of the subjects in above authorization mechanisms by substituting their identity with pseudonyms. This way, the Personally Identifiable Information (PII) of the subjects remains within the boundaries of their "home" organization.

The next section explains how these concerns are addressed in the proposed security architecture.

B. *The Proposed Solution*

Security mechanisms of such solution are not simple and cannot be implemented in one piece/concentrated in one place. It is commonly (with some variations) acknowledged, that these mechanisms need to be distributed and grouped according to their purpose.

We can distinguish the following main security tiers (Figure 3):

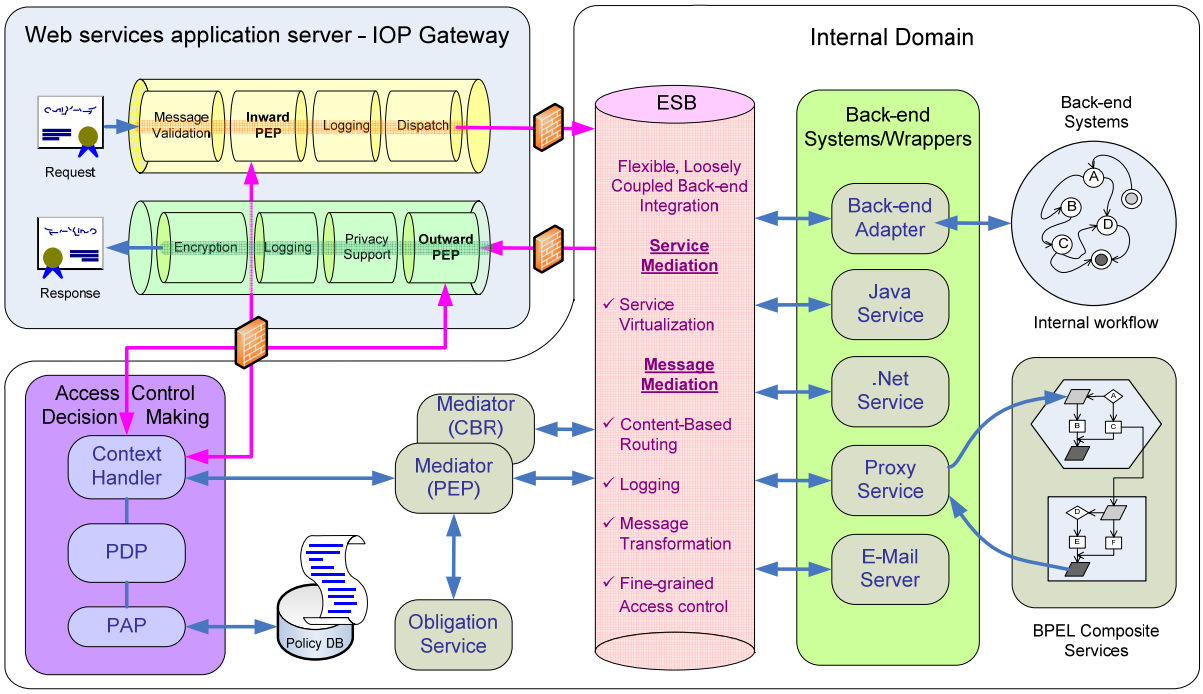

Figure 3 - Distribution of Security Mechanisms in R4eGov framework

- Protection and threat prevention
- Access enablement: Identity and Access Management – IAM

The security mechanisms are distributed accordingly. In our security architecture the "protection and thread prevention" part spans not only network/transport layer security but also message (e.g. SOAP) layer security, delivery of messages

between the gateways according to the confidentiality, authenticity and integrity requirements.

Similar security functionality distribution is also advocated by Mozes [16], he distinguishes between the SOAP intermediaries and security intermediaries in WS-based collaborative security architecture. The authentication (between the IOP gateways) and coarse-grained authorization can be performed at the system boundary (we put these functions into Web services engine message processing handlers), using any one of a variety of authentication mechanisms, such as conventional Web-access management techniques or one of the available federated identity solutions. This ensures that messages must pass a rigorous test before being allowed into the internal network. If service interfaces must be exposed to unauthenticated clients, messages must be subjected to a different test. In this case, schema-validation is a suitable test to prevent XML attacks. In both cases certain attacks remain a problem, e.g. Denial of Service (DoS) attacks. Gruschka & Luttenberger propose a mechanism to address the threats of DoS attacks [17].

Protection and threat prevention part of our solution focuses on data authenticity, integrity and confidentiality, which actually means encryption and digital signatures. In the Web services domain, WS-Security, an OASIS standard, is an open format for signing and encrypting message parts (leveraging XML Digital Signature and XML Encryption protocols), for supplying credentials in the form of security tokens, and for securely passing those tokens in a message. The core standards in this group comprise WS-Security Core (SOAP Message Security) and several token profiles including UserName Token Profile, X.509 Token Profile, Kerberos Token Profile, and SAML Token Profile. The token profiles enable serializing credentials in a consistent manner across platforms, certainly one of the driving forces behind the adoption of WS-Security in the first place.

The next section discusses access control mechanisms of R4eGov – this is one of the core security points, as it is much closer to the application security, much less commoditised in comparison to the message-level security instruments and of crucial importance for the owners of the R4eGov case studies.

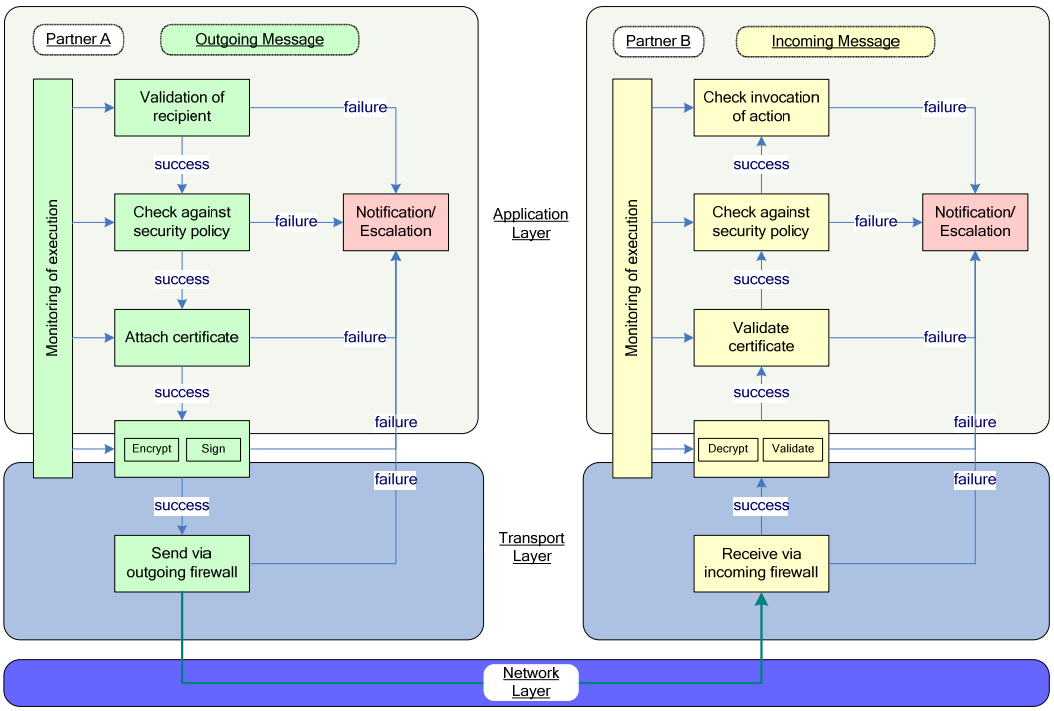

Figure 4 – Protection and threat prevention on the external IOP Gateway border

On the other hand, schema-validation, fine-grained authorization and other aspects of security policy can be enforced close to the application environment. This allocation of security services also supports an appropriate division of responsibilities between network administrators, who are responsible for the integrity of the internal network and who must have the controls necessary to do that, and application administrators, who are responsible for policy enforcement in the applications and who must have the controls necessary to do that. This functionality of our security solution resides in ESB mediators (Figure 3).

IV. Access Control Mechanisms

One of the central questions in security solutions is that of access control. In a nutshell, access control is the process of mediating every request to data and services maintained by a system and determining whether the request should be granted or denied. Access control is meant to protect resources (i.e., data and services) against unauthorized disclosure (secrecy, confidentiality) and unauthorized changes (integrity), at the same time ensuring accessibility of the resources by authorized users whenever needed (availability). These aspects sometimes

are mutually conflicting and balancing them requires a careful approach.

Figure 3 also illustrates an important access control implementation principle of our security architecture, i.e. well coordinated operation usage of Policy Enforcement Points (PEP) and Policy Decision Point (PDP) – access control decision making is concentrated in a single place (dealing with a vulnerability of having a single point of failure is a separate issue) and accessed from several PEPs:

- Loose standards-based (XACML [21]) coupling of PEP and PDP facilitates flexibility of potential deployment
- Policy management service supports the specification, interpretation and instantiation of different types of policies: access control & obligation (event-condition-action, ECA).
- Policy deployment service supports distribution and deployment of policies for usage by PDP.

One of the apparent virtues of the XACML framework is its modularity. XACML specification explicitly acknowledges that PEPs can be implemented in a variety of ways. For instance, PEP may be part of a remote-access gateway, a part of a Web server or part of an email user-agent, etc. In our architecture we can foresee two types of PEP. Firstly, the *incoming* requests received by the IOP Gateway are processed by a chain of handlers, one of them serving as PEP (Figure 4) and providing the initial crude screening of the request. This PEP acts as a "bouncer", performing fast "face control" and protecting the inner workings of the gateway from obviously unwelcome requests. The *outgoing* requests and responses are subject to inspection, outward access control and potential transformations, which are achieved by processing these outgoing messages by a chain of handlers controlling the outgoing flow. Once again, one of these handlers acts as a PEP and ensures enforcement of applicable policies.

Assuming that integration of the IOP Gateway with the back-end systems of the participants of the collaborations is done using Enterprise Service Bus (ESB), certain functionality of modern ESB implementations can be leveraged to further secure the interactions. In particular, the feature of *mediators*, which can be set up to intercept the messages sent via ESB, can be used for installing additional PEPs for finer-grained access control to the resources. There can be additional PEPs implemented as required and installed at some points of the system, which can't be foreseen in advance due to the scale of integration. In addition, there can be legacy PEPs, which will need to enforce new and/or updated policies.

Given the variety of PEPs, it is unrealistic to expect that all the PEPs in an enterprise do currently, or will in the future, issue decision requests to a PDP in a common format. Nevertheless, a particular policy may have to be enforced by multiple PEPs. It would be inefficient to force a policy writer to write the same policy several different ways in order to accommodate the format requirements of each sort of PEP.

Therefore, there is a need for a canonical form of the request and response handled by an XACML PDP. This canonical form is called the *XACML context*. Its syntax is defined in XML schema.

The XACML-conformant PEPs may issue requests and receive responses in the form of an XACML context. But, where this situation does not exist, an intermediate step is required to convert between the request/response format understood by the PEP and the *XACML context* format understood by the PDP.

The benefit of this approach is that policies may be written and analyzed independent of the specific environment in which they are to be enforced. The principle of separating the concerns of policy modelling/management from their enforcement environments/decision request formats is very important, as it allows to have consistent policy definition, verification and reasoning for all the requests/resources. XACML specification provides an abstraction-layer that insulates the policy-writer from the details of the application environment. As mentioned before, the canonical representation of a decision request and an authorization decision is called XACML context. Context handler is an entity, which converts decision requests in the native request format to the XACML canonical form and converts authorization decisions from the XACML canonical form to the native response format.

In multi-party interactions quite often is important to preserve privacy of the subjects (requestors), without compromising appropriate access control. Our contribution aims to solve this issue without a need to explicitly involve a third trusted party into the interactions. Privacy preservation is a complex task, affected by different kind of policies, defined by different parties:

- Access control policies govern access/release of data/services managed by the party (as in traditional access control)
- Release policies govern release of properties/credentials/PII of the party and specify under which conditions they can be disclosed
- Sanitization policies provide filtering functionalities on the response to be returned to the counterpart to avoid release of sensitive information related to the policy itself
- Data processing policies define how the PII will be (or should be) used and processed In our solution we will be using access control policies for fine-grained resource protection on the service provider side and properties/credentials release policies along with the sanitization policies on the requestor side.

The traditional identity-based access control models where subjects and objects are usually identified by unique names are not always suitable due to privacy concerns. It is easy to foresee a need to protect subject's privacy in e-Government interactions, for example in judicial and/or law enforcement domain. Therefore, attributes other than identity are needed to determine the party's rights to access a resource. In this case access restrictions to the data/services should be expressed by policies specifying the attributes a subject has to possess to get

access to the data/services. For example, a role or several roles (unless very explicit, such as top management) does not reveal person's PII. This is in line with role-based access control principles.

There are various ways to implement attribute based security tokens, one of them is to use digital certificates. Traditionally, a digital certificate has been mostly used as the identity certificate. An identity certificate is an electronic document used to recognize an individual, a server, or some other entity, and to connect that identity with a public key, thus solving key management issue. Another type of digital certificate is attribute certificate, which can be used in attribute-based access control mechanisms. An attribute certificate has a structure similar to an identity certificate but contains attributes that specify access control information associated with the certificate holder (e.g., group membership, role, security clearance).

V. Conclusions

The number of complex multi-domain/multi-country collaborations is constantly increasing, as the SOA concepts and supporting technologies are maturing. In order to gain acceptance, such solutions must be efficient, easy to use, secure and trusted. The work presented in this paper aims to leverage the best implementations of standard and interoperable Web services specifications to provide a lightweight and modular framework for inter-organizational collaborative interactions. The concept of application-level gateway is implemented using pluggable extensions of Web services engine and further enhanced by using intelligent message processing based on Enterprise Service Bus and mediation techniques. This kind of virtualization allows to achieve desired flexibility and security level providing standards based interoperability, data confidentiality, authenticity, integrity and role/policy based access control. These features, combined with the concept of Data as a Service (*DaaS*) enable the end users to have more power of creating ad-hoc enterprise data mash-ups, leverage benefits of enterprise social computing and gain additional opportunities when creating and reusing value-added knowledge.

A first prototype of the solution has been implemented using Apache and WSO2 Web services platform, WS-Security family of specifications. This prototype will be used to assess solution performance and suitability before moving towards enhancing choreographed interactions. A good case study for application of such compliance proof mechanism can be collaboration between public administrations of different EU Member States in legal/law enforcement domain [4] where efficiency, security and trustworthiness of interaction steps is highly important. Further work is planned on privacy-preserving access control protocol, fine grained specifications of access entitlement and distributed authorization mechanisms.

Acknowledgment

The work presented here is partially funded by the European Commission under contract IST-2004-026650 through the project R4eGov [3]. The authors would like to thank members of the organizations involved in R4eGov for their contribution: SAP Research Labs, University of Hamburg, Unisys Belgium, Europol, Eurojust, Austrian and others.

References

[1] The Hague Programme : strengthening freedom, security and justice in the European Union, 2004, http://www.libertysecurity.org/IMG/pdf/hague_programme_en.pdf

[2] S. Ross-Talbot, Interview for THE SOA NETWORK on Services choreography and WS-CDL, 2006 http://soanetworkarchitect.com/2006/07/19/wscdl-complementary-to-wsbpel.aspx

[3] R4eGov, EU IST FP6 project, http://www.r4egov.eu/

[4] R4eGov Case Studies, 2007, http://www.r4egov.eu/resources/details_s.php?Id_resources=22

[5] eGovInterop Case Studies, 2006, http://www.egovinterop.net/SHWebClass.ASP?WCI=ShowDoc&DocID=1213&LangID=1

[6] D. Needle, IBM's QEDWiki Adds 'Data as a Service', 2007, http://www.internetnews.com/ent-news/article.php/3699636

[7] Havenstein, H., US Government Agency Embraces Web 2.0, PCWorld, 2007, http://www.pcworld.com/article/id,129328-c,internetnetworking/article.html

[8] C. Wolter, H. Plate, C. Hebert, "Collaborative Workflow Management for eGovernment," Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference on , vol., no., pp.845-849, 3-7 Sept. 2007

[9] A. Svirskas, M. D. Wilson, B. Roberts, I. Ignatiadis, Adaptive Support of Inter-Domain Collaborative Protocols using Web Services and Software Agents, eds. O. Vasilecas, J. Eder, A. Caplinskas, Frontiers in Artificial Intelligence and Applications (IOS Press, Amsterdam), 2007

[10] D. C. Schmidt, "Applying a Pattern Language to Develop Applicationlevel Gateways," in Design Patterns in Communications (L. Rising, ed.), Cambridge University Press, 2000

[11] R4eGov collaborative workflow architecture, http://www.r4egov.eu/resources/details_s.php?Id_resources=43

[12] Apache Axis2, http://ws.apache.org/axis2/

[13] Apache Synapse, http://ws.apache.org/synapse/

[14] WSO2 Web services Application Server, http://wso2.org/projects/wsas/java

[15] H. Lee, H. Luedeman “A Light-weighted Decentralized Authorization Model for Inter-domain Collaborations”, 2007, ACM Workshop on Secure Web Services (SWS'07)

[16] T. Moses, “Security in Web services world”, Entrust Inc, 2004

[17] N. Gruschka., N. Luttenberger “Protecting Web Services from DoS Attacks by SOAP Message Validation”, 2006

[18] Web Services Technologies, W3C, 2004, http://www.w3.org/TR/ws-arch/#technology

[19] Advanced Web Services, http://msdn2.microsoft.com/fr-fr/webservices/Aa740686.aspx

[20] Security in a Web Services World: A Proposed Architecture and Roadmap, IBM Corporation & Microsoft Corporation, 2002, http://www.verisign.com/wss/architectureRoadmap.pdf

[21] eXtensible Access Control Markup Language (XACML), OASIS Standard

[22] I. Djordjevic, K. S. Nair, T. Dimitrakos “Virtualised Trusted Computing Platform for Adaptive Security Enforcement of Web Services Interactions”. ICWS 2007, p.p. 615-622

[23] S. Bandhakavi, C. C. Zhang, M. Winslett “Super-sticky and declassifiable release policies for flexible information dissemination control” In Proceedings of the 5th ACM Workshop on Privacy in Electronic Society (Alexandria, Virginia, USA, October 30 - 30, 2006). WPES '06. ACM, New York, NY, 51-58.