We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

A Lightweight SOA-based Collaboration Framework for European Public Sector

Adomas Svirskas[1,2], Jelena Isačenkova[1] and Refik Molva[1]
[1]*EURECOM*
Sophia-Antipolis, France
[2] *Vilnius University*
Vilnius, Lithuania

1. Introduction

E-Business and e-Government solutions are becoming more and more widespread with constantly growing number of 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 fulfil 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 chapter presents the work among to create an efficient, secure and trusted interoperability framework for public sector agencies of European Union member countries.

In a simplified view, the administrations in Europe as a whole can be seen as a forest. Each tree is one specific administration of one member state (Fig. 1.). Each leaf is one particular service provided by this administration. E-Administration today mostly consists of providing services via IT tools and reducing the paper support via a user-friendly interface to the service. This is what we call "e-Administration in the small".

The real challenge is to enable smooth collaboration between the trees, i.e. between administrations of the same or of different member states. This is "e-Administration in the large" and reflects our understanding of collaborative e-Government systems. The R4eGov project (R4eGov, 2005), aims at providing the basic conceptual and technical framework for the first e-Administration in the large for Europe. R4eGov emphasises the basic control and security principles of:

- Local data ownership: each "leaf" uses data stores, in most cases local, to store its data. For example, civil status data (état civil), which records the birth, marriage, death of French citizens born in France are managed by each city.

- Local information access policies: data access is generally restricted to some personnel of the administration which operates the service. Access to some of the data, certainly not to the entire data stores, must be granted to personnel of other administrations, under controlled conditions. Data stores/bases in their entirety should never be displaced nor copied nor merged to permit e-Administration in the large.
- Local enforcement of organisational control and security policies, according to the local legislative acts

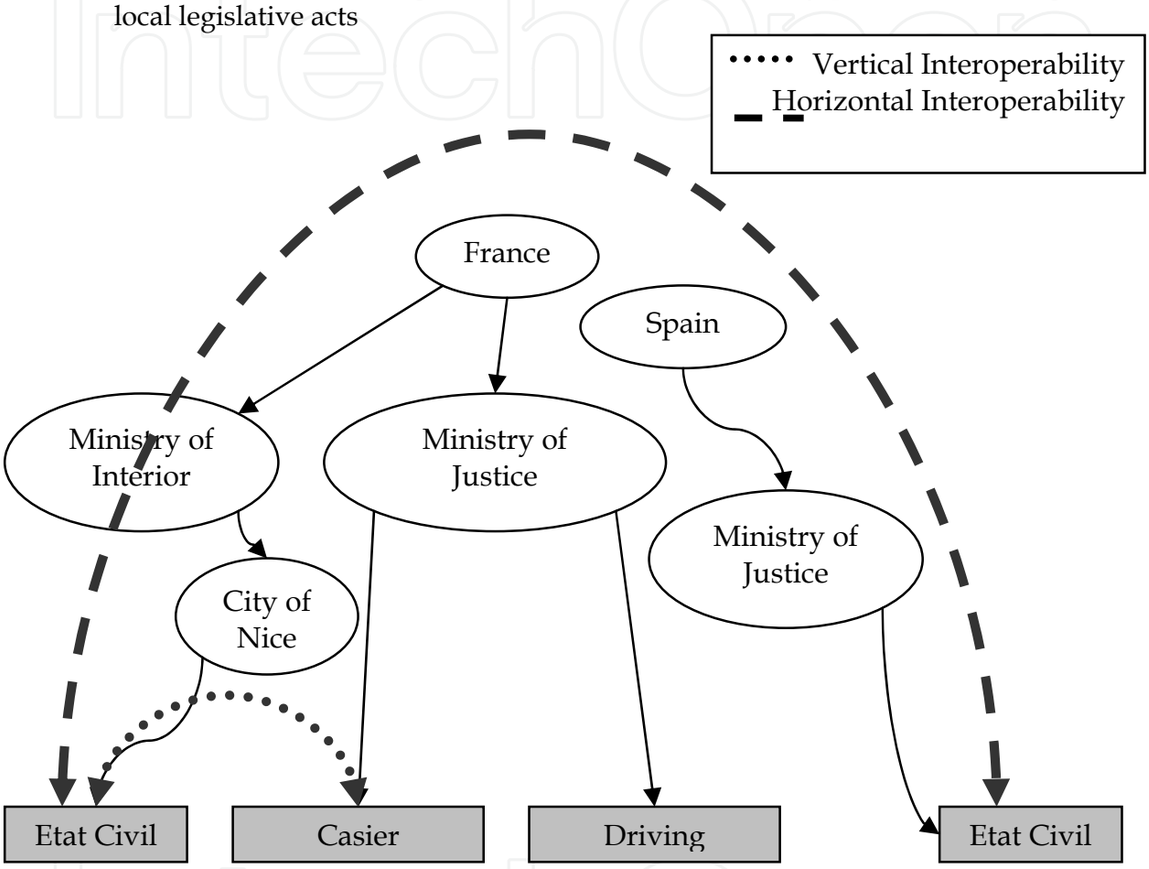

Fig. 1. E-Government Forest and types of interoperability

A stepwise refinement approach focusing on progressive development and seamless deployment of collaborating, cross-organisational collaborative workflows is required as no single process model covering all European administrations will ever be created from scratch. To permit the progressive take-up of government process modelling, the environment must allow for incomplete and not fully formal descriptions, in addition to descriptions being developed separately with little concern for interoperability.

The R4eGov project takes into account the answers provided by existing bodies (eEurope eGovernment Subgroup, IDABC Management Committee , ADAE , etc), which debate on how to best grant access to private data owned by other Member States, while protecting privacy as requested in the constitution of the concerned Member State. To be practical, we address the problems raised by the case studies of the R4eGov project (such as Europol/Eurojust, Austrian government etc). We develop a collaborative interaction model

of the European e-Administration, where the processes, the levels of control, security and timing requirements are described. This will constitute the top level picture of the European e-Government. Boxes at the bottom of Fig. 1 are examples of domains (Etat civil, Casier judiciaire, Driving offences) that have adopted IT and are examples of successful e-Administration applications. Today, exchanges between administrations of one country (such as, for example, between French Etat civil and Casier judiciaire) are being put in place with often little consideration for the need of mutual trust. Exchanges between states are possible but of little practical use. It is urgent to address these issues.

Thus, the three main objectives of the R4eGov project are:

- To gather and elicit the requirements for e-Administration in the large, on basis of which a concrete interoperation of web service enabled legacy public sector applications will be achieved using collaborative interactions.
- To provide the tools and methods for an e-Administration in the large from a technical and sociological perspective.
- To provide the required security and privacy for an e-Administration in the large, defining the appropriate methods and tools for control, security and privacy at the collaborative workflow and application layer.

One of the main results of this project is a reference framework and its proof-of-concept implementations based on several case-studies derived from the business requirements of the public administration organizations participating as partners in R4eGov project, such as Europol, Eurojust etc. The framework, thus, is quite generic and customizable to fit varying business needs.

Interactions among the participants 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 our work, the 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 by exchanging mutually understandable messages among themselves.

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 the Hague Programme (Hague, 2004), 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.

2. A motivating case study from the European e-Government field

We have motivated our work, to a large extent, with a use case from the Europol (EP)/Eurojust (EJ) collaboration domain. These two agencies have been set up to help the EU member states co-operate in the fight against cross-border organised crime. Co-operation in criminal matters is a subject dealt with in the "third pillar" of the EU (Title VI of the Treaty of the European Union). Eurojust stands for the European Judicial Cooperation Unit whereas Europol refers to the European Police Office. Europol and Eurojust carry out very specific tasks in the context of the dialogue, mutual assistance, joint efforts and co-operation between the police, customs, immigration services and justice departments of the EU member states.

Europol became fully operational in 1999; Eurojust was set up by a Council Decision in 2002. Both agencies are based in The Hague, The Netherlands. Establishing a secure IT connection between Eurojust and Europol has been an objective shared by the two organisations for several years. Europol and Eurojust are two key elements of the European system of international collaboration within the areas of law enforcement and justice. Even though they differ considerably in the way they are organized, on how they operate and on their mandated areas, there is an overarching need to ensure smooth collaboration and effective information exchange between the two organizations.

In spite of the presence of legal basis for collaboration between the two organizations since 2002 (the Eurojust Decision and the Agreement between Eurojust and Europol of 9 June 2004), the complexity of the nature of the exchange (content, channels, sources, recipients, means, etc.), the compliance framework (legal basis, compliance and data protection rules) and the different implementation of policies and procedures in the 27 member states, have all limited the collaboration on cases to a small number.

From the technical point of view, it should be noted that both organisations already manage their information through computerised systems (Europol's Overall Analysis System for Intelligence and Support (OASIS) and Eurojust Case Management System) but these systems are currently separate.

Therefore, a structured approach on how to deal with the barriers to information exchange must be undertaken. While lines of communications at various levels already exist, these channels should be supported by computerised technical devices to allow the secure and swift exchange of information in the framework of the above legal instruments and the applicable data protection rules. In practical terms, it is necessary to follow an appropriate methodology to establish sound solution architecture and obtain/create ready-to-use tools (framework and software) in order to:

- Facilitate collaboration between Europol and Eurojust
- Improve the collaboration of the sources of Europol and Eurojust
- Get the tools and implement a practical viable solution

The creation of a framework (conceptual and practical, including tools) that allows dealing with the multiplicity of instances of technological and business rules, would allow for an identification of the barriers that impede international collaboration and for taking appropriate countermeasures on a technological or business level (including the change in legal framework) where necessary.

In order to illustrate the aforementioned issues, let’s take a simple information exchange scenario depicted in the Fig. 1. The scenario involves four parties, each having their own independent administrative and security domains. All the parties should use some kind of

Collaborative Workflow Management System (CWfMS) to better integrate and automate their collaboration processes.
Eurojust uses a Case Management System (CMS) to support the work of its prosecutors, but the CMS is not the only source of information. EJ depends on information provided by EP, which does the main investigation work. Furthermore, as EP is also a centre of collaboration of all member states of the European Union, it can request information on a suspect from a national investigation bureau. This is what happens in the chosen scenario, which, roughly, consists of the following individual collaboration steps:

1) An EJ case is created in the CMS
2) During a meeting at EJ, it is decided that the National Member of state a (NMa) will request data from the Case Analysis bureau of her/his member state (MSCAa)
3) NMa sends the request to the Europol Liaison Officer of her/his state (ELOa)
4) ELOa contacts the Europol National Unit of her/his state (ENUa). ENUa contacts MSCAa and MSCAa sends back data to ENUa, which passes it on to ELOa
5) ELOa retrieves additional data from the InfoEx system
6) ELOa sends back both data sets to NMa.
7) NMa updates the CMS with the data

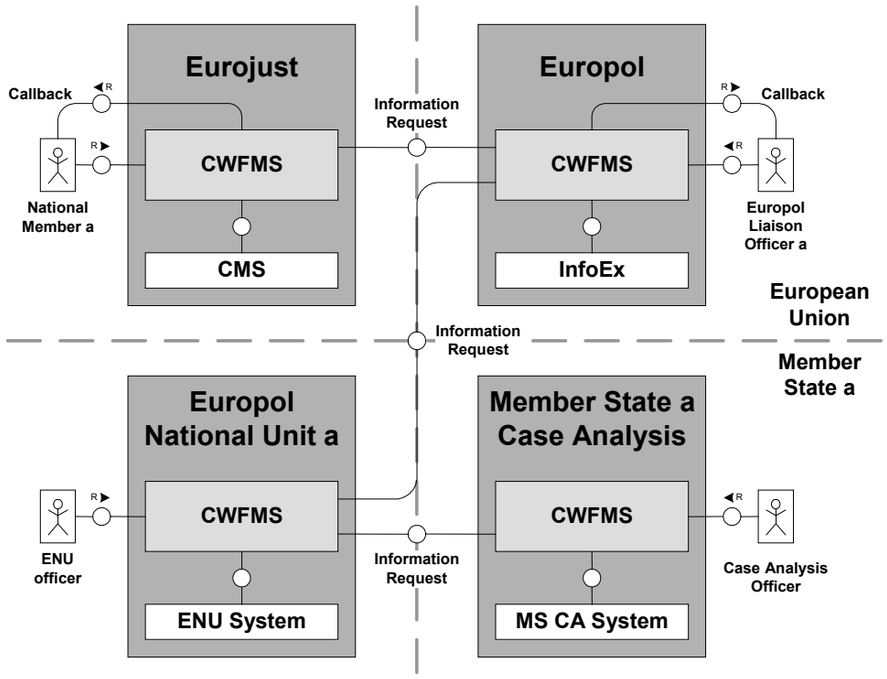

Fig. 2. A sample information exchange scenario from EP/EJ case study

These steps, with some variation, are further depicted in more detailed and complex way in the Fig. 3, which is a real-world example of the issues to be dealt with. Furthermore, this simple scenario is only a very small part of the EP/EJ collaborative interaction needs and, in turn, the EP/EJ case study as whole is just a tiny (however important) bit in the overall European e-Government landscape.

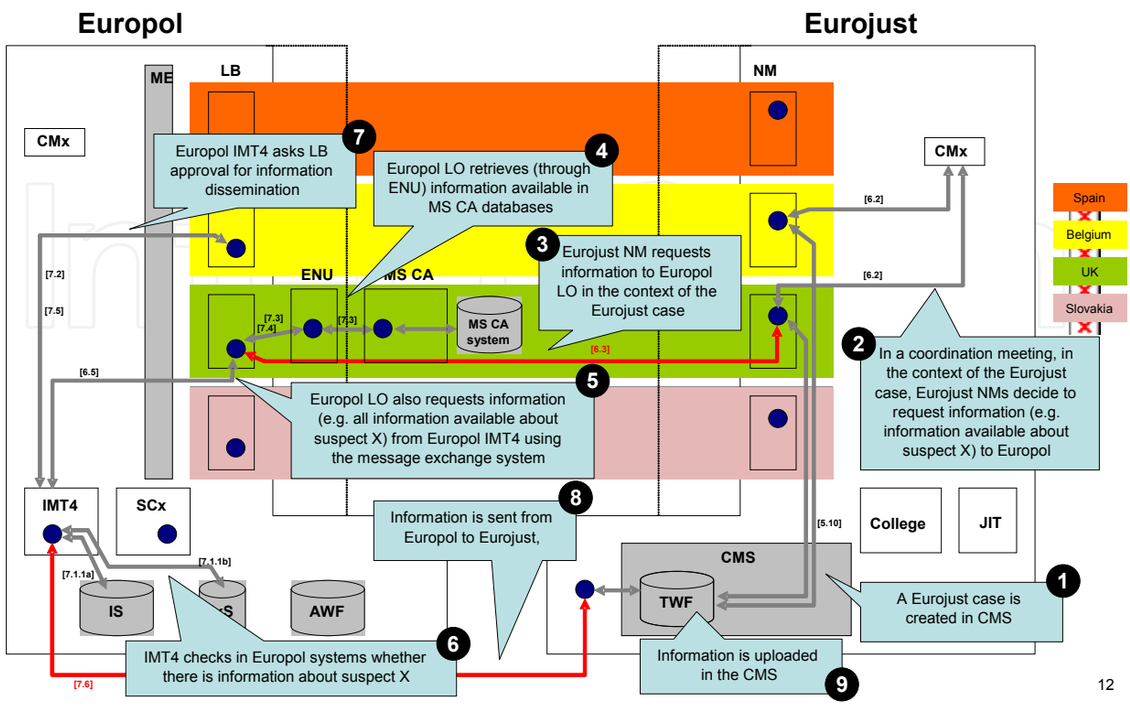

Fig. 3. Interaction Flow - Exchange of information between Eurojust and Europol

It is obvious that the problem of pan-European interoperability among the public administration bodies is a very complex one and, of course, the R4eGov project is not the first effort trying to deal with it. In the following section we will outline the principles, which constitute the basis of our solution architecture and framework.

3. The solution architecture

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 (R4eGov case studies, 2007; eGovInterop Case Studies, 2006) and the current state of the art of possible implementation technology, we distinguish a few important factors, which determine the context of our solution architecture:

- 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 to ensure interoperability at shared information level.
- Independence of each agency and its operation according to the local legislative acts, regulations etc.
- Agencies being very protective of their data with regard to who can have access to it, what kind of "channels" the data can travel, who has seen the data etc.
- Enormous variety of the technical solutions being used by different agencies and countries

- Different countries and agencies are at very different levels of readiness to embrace e-Government collaborative interactions. The differences are found at legal, procedural, conceptual, cultural, technical, human skills levels.

These factors make a good case for using the service oriented (SOA) approach to integration of the information resources, i.e. each data 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. Thus, the key issues such as independence of the agencies and their differences become manageable – the agencies expose as much data as they possibly can and are willing to while the access to this data is granted only to the authorised parties.

3.2 The conceptual SOA approach

The architectural approach based on service and data virtualization is a sound practical foundation for implementing the *data as a service* concept 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. This relatively new concept of Web 2.0 paradigm has already found its place in e-Government: for example, 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, (Havenstein, 2007).

In addition to this, the integration framework needs to be lightweight and reasonably simple, taking into account that the level of integration readiness varies greatly from agency to agency in different countries. Simplicity allows the agencies to start small, experience benefits of integration and collaborative interactions, then iteratively add new services, as needed. Such approach will lower the entrance barrier for the less-prepared partners and will increase the chance of successful adoption of the solution.

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. 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 (Schmidt, 2000; Svirskas, 2007) 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.

3.2 Implementation technology

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 (Web Services, WS Security). Web services are used for both inter-domain communication between the gateways and internal services, which may represent some newly developed functionality or serve as wrappers for legacy systems.

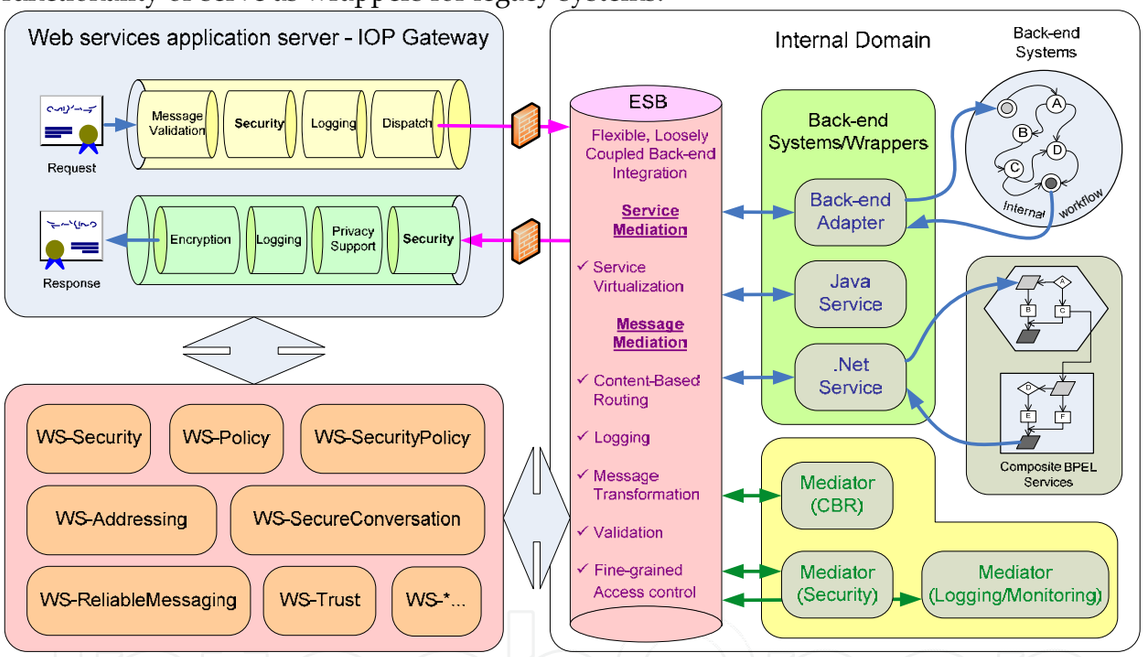

Fig. 4. Architecture of collaborative interaction platform based on IOP gateways

It is quite clear, that implementing the IOP Gateway features, as depicted in Fig. 4., 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 .

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 (R4eGov), 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.
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 (Apache Axis2), 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 (Apache Synapse) 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 (WSO2) are designed to work together, have solid developer and user base, which increases the chances of successful practical use.

4. Security features of the R4eGov framework

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.

Apart from these general security concerns 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 a peer-to-peer collaborative environment, where data is transmitted among multiple partners it may happen, for example , 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, 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).

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 (Lee & Luedeman, 2007).

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. 2007) 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.

4.1 Organization of security mechanisms

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:

- 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, 2004), 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 (Gruschka & Luttenberger, 2006). 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.
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.

4.2 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.
An important access control implementation principle of our security architecture is 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) 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 and use the same PDP, essentially enabling authorization as a service.
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 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.
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 is not the case, 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, as depicted in Fig . 5.

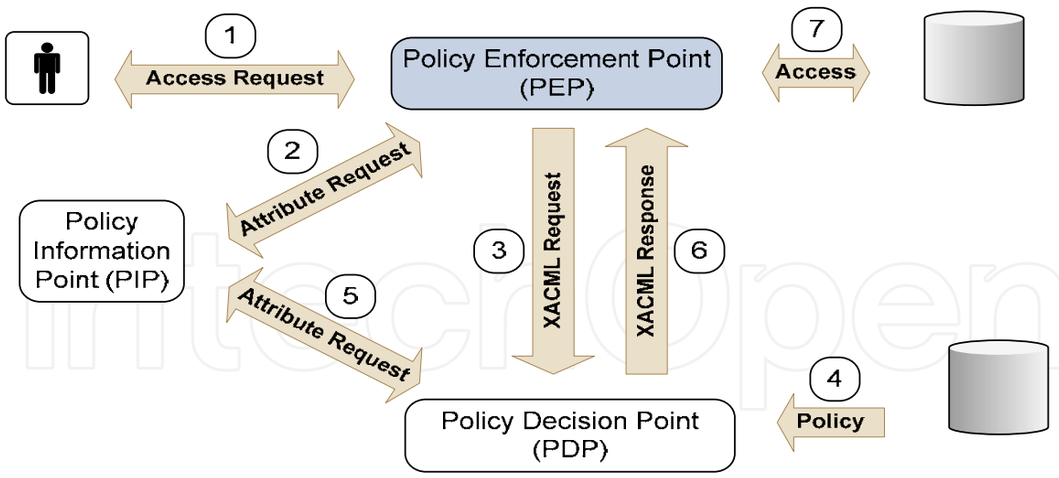

Fig. 5. Typical XACML-based authorization process scheme

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.

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.

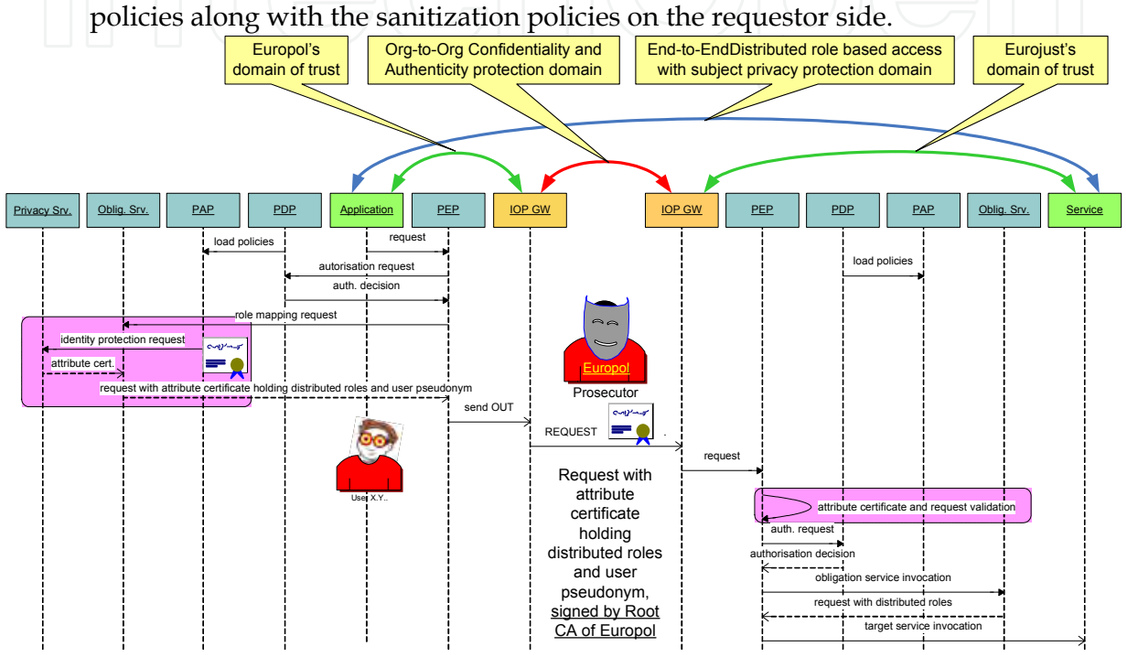

Fig. 6. Typical XACML-based authorization process scheme

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. Fig. 6. illustrates privacy-preserving access control protocol of the R4eGov framework.

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).

5. 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 chapter 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 achieving needed 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 prototype of the described 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 (R4eGov case studies, 2007) 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.

6. Acknowledgements

The work presented here is partially funded by the European Commission under contract IST-2004-026650 through the project R4eGov (R4eGov). 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 Bundes-chancellor office, in particular.

7. References

Apache Axis2, http://ws.apache.org/axis2
Apache Synapse, http://ws.apache.org/synapse
Corporation, (2002). http://www.verisign.com/wss/architectureRoadmap.pdf
eGovInterop Case Studies, 2006, http://www.egovinterop.net

Djordjevic, I.; Nair, K.S. & Dimitrakos, T. (2007). Virtualised Trusted Computing Platform for Adaptive Security Enforcement of Web Services Interactions. ICWS 2007, p.p. 615-622

Gruschka, N. & Luttenberger, N. (2006). Protecting Web Services from DoS Attacks by SOAP Message Validation

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

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

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

Moses, T. (2004). Security in Web services world", Entrust Inc.

R4eGov, R4eGov case studies (2007). EU IST FP6 project, http://www.r4egov.eu

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

Svirskas, A.; Wilson, M.D.; Roberts, B.; Ignatiadis, I. (2007). 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

Web Services, W3C, (2004). http://www.w3.org/TR/ws-arch/#technology

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

WS Security: A Proposed Architecture and Roadmap, IBM Corporation & Microsoft

XACML, eXtensible Access Control Markup Language , OASIS Standard

E-learning Experiences and Future

Edited by Safeeullah Soomro

ISBN 978-953-307-092-6
Hard cover, 452 pages
Publisher InTech
Published online 01, April, 2010
Published in print edition April, 2010

This book is consisting of 24 chapters which are focusing on the basic and applied research regarding e-learning systems. Authors made efforts to provide theoretical as well as practical approaches to solve open problems through their elite research work. This book increases knowledge in the following topics such as e-learning, e-Government, Data mining in e-learning based systems, LMS systems, security in e-learning based systems, surveys regarding teachers to use e-learning systems, analysis of intelligent agents using e-learning, assessment methods for e-learning and barriers to use of effective e-learning systems in education. Basically this book is an open platform for creative discussion for future e-learning based systems which are essential to understand for the students, researchers, academic personals and industry related people to enhance their capabilities to capture new ideas and provides valuable solution to an international community.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Adomas Svirskas, Jelena Isacenkova and Refik Molva (2010). A Lightweight SOA-based Collaboration Framework for European Public Sector, E-learning Experiences and Future, Safeeullah Soomro (Ed.), ISBN: 978-953-307-092-6, InTech, Available from: http://www.intechopen.com/books/e-learning-experiences-and-future/a-lightweight-soa-based-collaboration-framework-for-european-public-sector

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen