A Secured Service-Oriented Architecture for E-government in Tunisia

Mohamed Sellami, Mohamed Jmaiel
ReDCAD research unit
National School of Engineers of Sfax
Soukra road Km 3.5, B.P.W 3038
mohamed.sellami@gmx.net, Mohamed.Jmaiel@enis.rnu.tn

Abstract

With the increase of the use of information and communication technologies, e-government becomes an orientation to follow. Until now, e-government applications in Tunisia are limited to an informative goal, i.e. they essentially offer information and not services. In order to reach successful e-government applications, we have to provide services to citizens. Moreover, in e-government applications, security represents an important feature for both the citizens and the governmental administrations which require a particular interest. In this paper, we propose a secured and service-oriented architecture for the accomplishing of Tunisian e-government applications.

1. Introduction

During the last ten years, the Tunisian government engaged itself in a plan aiming to setting up an electronic administration, where the counter of the citizen is his computer. In fact, the government encourages the use of the new information technologies and fosters the Internet users. Besides, it launched the family computer program aiming to join all the social categories to the computer techniques. Now, the Tunisian technological infrastructure is favoured for e-government actions, some projects were carried out. They have primarily an informative goal. Actually, the realised e-government applications offer information about government services, but they do not offer the required services. Another initiative showing the interest granted by Tunisia for e-government is the creation of a unit for the electronic administration within the first ministry. Tunisia plans to undertake offering e-government applications proposing on-line services on a second stage.

Usually, e-government applications require both a high level of interoperability and security. The interoperability requirement is essentially due to the huge number of heterogeneous participants in such applications. In view of the private and confidential information exchanged between governmental agencies and citizens, it is also necessary to take into account the security exigency. A service-oriented approach presents a solution for those requirements. It allows the inter-connection between diverse heterogeneous administrations through well-defined standards. Web services [4] are commonly used as an implementation for service-oriented approaches. They offer interoperable entities and propose standards to secure them. This fact encourages various countries to engage themselves in Web services based e-governments applications.

Some developed countries has made a good use of the service-oriented approach and implemented e-government applications using them (EU-PUBLI, E-mayor, WebDG) [1-3]. To adapt this approach to the Tunisian context, we must take into account the mentality of the Tunisian citizens and administrations employees. Particularly, the Tunisian administrations are extremely wary of giving external access to their data bases. For this reason, we propose a solution allowing the administrations employees to control, even manually, external access to the data bases.

In this context, and to contribute to the development of the electronic administration in Tunisia, we propose in this paper a secured and service oriented architecture for the Tunisian e-government. This architecture is made up of five layers: the client layer, the presentation layer, the application layer, the data layer and the administration layer. The rest of this paper is structured as follows: section 2 presents the service-oriented approach, Web services technologies and their contributions in the realization of e-government applications. Section 3 presents some Tunisian projects in the field of e-government and the principal requirements/solutions of the e-government applications relative to the Tunisian case. In section 4, we detail, "Web services with switches", a proposed solution for the Tunisian administration "Data

restriction" requirement. Section 5 offers a complete view on the different layers of the proposed architecture. In section 6, we present an applicative scenario of an e-government application. Finally section 7 contains conclusive remarks.

2. Service-oriented architecture and Web services

Since the e-government services require a high degree of interoperability and distribution due to the numerous entities they imply, a service-oriented architecture (using the technologies of Web services) is a potential candidate to solve a significant number of e-government problems. Indeed, in a service-oriented architecture (SOA), all the software components are modelled by services. In this model, the design of an application is concentrated on the combination of slightly connected and distributed services on a network to form larger applications. That's why; the exchanged information between the services must be coded in an interoperable manner. Moreover, the service provider must offer a description explaining the necessary steps to invoke the offered services. In fact, SOA involves three actors (see Figure1):

- The service provider: the entity which creates the service and makes it available to other entities.
- The service consumer: any entity or person using a service created by a service provider.
- The service registry: a centralized directory where the service providers can publish their services, and where any service consumer can search for them. It allows the service consumers to find the services they need.

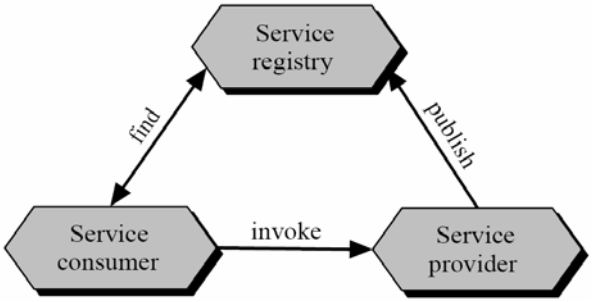

Figure 1. Design model of a service oriented architecture

The objective of such architecture is to offer effective and flexible services in a network. Otherwise, SOA are benefit interoperability since the necessary steps for the invocation of a service must be described in a standardized way. And their invocation is independent from the implementation language and the platform lodging the service.

To implement service oriented applications in a normalized way, we should use Web services. In fact, this technology proposes standards for the message exchanges and the service description. A Web service [4] is defined by the W3C[1] as:

> "*A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP[2] with an XML serialization in conjunction with other Web-related standards.* "

Web services provide the necessary standards to implement service oriented architectures: SOAP, WSDL and UDDI. SOAP [5] defines the structure of the messages and the protocol used for the standardized exchange of information in heterogeneous networks. It can be employed as the messages exchange protocol in a service-oriented architecture. Moreover, the service consumers do not know the necessary message format for the invocation of a service. Therefore, the service provider must provide the interface of the service. WSDL [6] is a language employed by the service providers to describe their services in an interoperable manner. By employing SOAP and WSDL, the "invoke" operation, presented in the design model of a service-oriented architecture (see Figure 1), can be applied. To implement the "publish" and "invoke" operations, another standard: UDDI [7] is defined. In our work, we choose the use of a service oriented approach for the implementation of the Tunisian e-government. This approach, which uses the Web services technologies, represents a good solution vis-à-vis a lot of the problems of e-government applications. It makes it possible to exceed the problems related to interoperability between the various government administrations. Indeed, Web services, and service-oriented approaches in general, offer a flexible environment allowing the co-operation between various governmental administrations. The result of these co-operations will be in the form of e-services intended to achieve citizens' needs. Moreover, Web services support the composition of the services by offering languages to this end. This is of a great importance for e-government applications considering the flexibility and the facility of the design of new composed services. In the next section, we present a short outline of some Tunisian e-government works while bringing out the requirements of the Tunisian e-government applications.

[1] World Wide Web Consortium

[2] Hypertext Transfer Protocol.

3. The Tunisian e-government requirements

During our research, we were interested in four Tunisian projects achieved in the field of electronic administration: the SICAD project [8], BAWABA [9], E-CNSS [10] and E-CNRPS [11]. From these studies, we noticed that these applications are primarily informative. In fact, they provide, except the last two, the citizens with information related to public services, but do not offer the services in question. With regard to the E-CNRPS and E-CNSS projects, we can consider that they represent a good example of Tunisian e-government applications. They place at the disposal of the citizen both information, and electronic services (e-services). However, these e-services remain simple services (following-up a file …) which do not request interactions with other administrations. So, when a citizen wishes to receive, for example, a service implying several administrations, he cannot profit from such a service on-line. It is up to him to take the trouble to move in order to regulate his files. This absence is not at all due to technological reasons, which pushes the citizens to request electronic services in replacement of the complicated administrative services. As a result for these facts, appears the need for an architecture guiding the development of e-government applications that offers services to the citizens. Such architecture will have initially to answer the requirements of the citizens and in the second place those of the governmental administrations.

Several requirements are generic to the majority of the e-government applications of various countries (e.g. on-line services, multiple access channels…). In addition to these generic requirements, there are the specific ones to each country. For example, the American project WebDG [3] was interested in the automatic composition of Web services thus, stressing the back-office. This aspect is not a priority for Tunisia which is relatively late on the field and which encourages citizen-centred applications.

In our work, we had discussions with several people working in governmental administrations. We noticed that the Tunisian case represents some specificity that we do not find in other countries. For example, the majority of the citizens do not grant much confidence to the computerized systems. Thus, it proves the necessity to develop e-government applications taking into account their requirements so that we are sure that they will be used.

Another specificity was noticed at the administrations employees level. Indeed, these latter do not wish to offer an automatic access to their data bases and they are very persistent on this point. Therefore, a proposal of architecture for the electronic government applications will have to take into account this constraint.

The architecture that we propose for the Tunisian e-government was carried out after a meticulous study of the principal requirements of the e-governments applications relative to the Tunisian case. We should notice that some of these requirements are common for most e-government applications. These requirements are primarily related to security and are divided into two groups: those relating to the citizens and those to the administrations.

3.1. Citizens requirements

The citizens requirements, defines what a citizen wishes to obtain through an e-government application. The most important recognized requirements are:

Multiple access channels. The citizen wishes to have many ways of obtaining a service. So he can use that which is more appropriate to his needs.

This requirement was solved by the separation of the presentation layer from the application layer in our architecture. Indeed, the presentation layer is responsible for providing the access interface to the services and the application layer presents their business process. Such a separation enables the presentation layer to have several forms (Web server, WAP server), without having to carry out any modification at the applicative level.

On-line information and services. Vis-à-vis the development of information and communication technologies, the citizen wishes to benefit from on-line e-services, without having to move. Nevertheless, some citizens wish to get them manually. For that, the e-government applications must provide on-line information concerning the way of acquiring those services.

To provide citizens and the various administrations with e-services, we use Web services. The presentation layer of our architecture offers them information they need. Moreover, in order to offer the citizen good quality e-services in replacements of the complicated administrative services, an orchestrator is necessary. Indeed, by adding an orchestration engine to the application layer of our architecture, we can offer e-services implying more than one administration. This collaboration is possible since we use a service-oriented architecture.

Authenticity of the e-government application. The citizen would like to be sure that he is connected to the e-government application belonging to the administration in question.

For this, while connecting to an e-government application, the citizen will have to receive a digital certificate proving the identity of the owner of the application. To implement this, we use the SSL[3] protocol which allows the authentication of a server thanks to the use of digital certificates.

Privacy Constraint. The citizen want to be conscious in which aim the information he will provide will be used for (identity card number, salary, social security number…) and who will have access to these information.

As a solution for this constraint, we have to show in details to the citizen the itinerary the information sent will follow and who will be able to access them.

Data confidentiality. The information sent by the citizen has to remain confidential.

The data sent to the presentation layer will have to be encrypted with the public key contained in the digital certificate of the concerned administration. For this, we employ the SSL protocol which makes it possible to implement an encrypted session for the data exchange.

An acquittal. After he benefits from a service, the citizen must be able to prove it if necessary.

At the end of the execution of a service, the application will have to refer to the citizen a digitally signed document attesting the fulfillment of the service in question. Henceforth in Tunisia, this document acquired the same legal aspect as a written paper.

3.2. Administrations requirements

In this part, we describe the constraints relating to the administrations which will have to be respected by our application as well as the adopted solutions.

Interoperability. In Tunisia, each governmental administration is responsible for the development of its own information system. Therefore, we notice heterogeneity in the technologies and the platforms used.

This heterogeneity can be overcome by using the service oriented approach which makes it possible to exceed this kind of obstacle. Indeed, this approach allows the co-operation between heterogeneous systems since it is independent of the platform and the implementation language.

Data Restriction. Generally the Tunisian administrations refuse to give access to their data bases. The "Web services with switches" provide a solution for this requirement, while maintaining the autonomy of the e-government application. More details on this solution will be given in section 4.

Citizen authentication. The government administration must be able to authenticate a citizen asking for a service.

This is realized by using the pair «username/password» exchanged between the administration and the citizen.

Administration authentication. A government administration will have to be able to authenticate another administration which requires a service and to authenticate itself near the other administrations.

Since we are using a SOA based on Web services, the different administrations are communicating using SOAP. So, their authentication can be assured through a digital signature contained in the exchanged SOAP messages. The signature will be carried out in accordance to the WS-Security[4] [12] standard in the aim of not interfering with the interoperability constraint of our architecture. Indeed, while following this standard, the blocks of SOAP messages are signed in a standardized way and they remain comprehensible by the manifold implied partners.

Requests integrity. The administration will have to be sure of the received requests integrity, i.e. it checks that they have not been deteriorated.

A digital signature applied to a SOAP message makes it possible to the governmental administration to be sure that the data contained in the latter was not modified. This signature will also be inserted in the SOAP message in accordance to the WS-Security specifications.

Filtered services access. Some administrations may have the right to invoke a service while others not.

As a solution, we allot to each service different pairs of «username/password». To invoke a service, the consumer must provide its pair of authentication as an input.

Data confidentiality. The data contained in the application's data layer and in the exchanged between the different administrations must remain confidential.

Concerning the data layer, it has to be protected by a firewall. This will filter the connections requests by their IP address.

Moreover, the data contained in the exchanged SOAP messages will have to be encrypted in order to ensure their confidentiality. Just as for the signature, encrypting will be carried out following the WS-security specifications.

[3] Secure Socket Layer : it's a protocol securing the exchanges over Internet

[4] A model defining standards for securing SOAP messages.

The presented list of requirements is non exhaustive, but it contains those we consider the most important for the Tunisian e-government. Based on these requirements, we developed an architecture for the Tunisian e-government applications (see Section 6).

4. Web services with switches

Generally, the Tunisian administrations refuse giving other administrations access to their data bases especially when this access is automatic. This comes primarily as a result of the lack of confidence with respect to the computed systems and the fear from a possible deterioration of the data. Accordingly, the Tunisian administrations prefer not to give software entities belonging to other agencies an automatic access to their bases.

To cure these limits, while keeping to the maximum the autonomy of the e-government application, we should propose an additional layer allowing the administrative services. This level makes it possible to the chief employee of the administration to have a follow-up of the system.

In our architecture (see Section 6), an administration can only use data of another administration using their web services. Hence, we defined the «Web services with switches». These services can be switched on or off at will by the chief employee of the administration owning the service. While switching off a service, it will be automatically replaced by a notification service whose role is to inform the administration that someone invoked the switched off service (see Figure 2).

According to this figure, we notice that two cases are possible:

- The service is activated (switched on); in this case the access to the data base by the service is automatic. The event will be recorded in a log file. The administration employee will be able to consult the execution result of this service through a follow-up console.
- The service is deactivated (switched off); when the service needs to access the data layer a notification will be sent to the chief employee of the administration. In the same way, this event will be recorded in a journal. Then, the employee will have to answer manually the received request. He plays the role of an intermediary between the other administrations and the data layer.

The services with switches constitute a solution vis-à-vis the requirement expressed by the Tunisian administrations and which is not to give an automatic access to their data bases. With this solution, the services offered by an administration can be activated or deactivated according to their needs. Thus, an administration which wishes to offer an autonomous e-government application will be able to do it and others will be able to create applications where a human intervention is necessary.

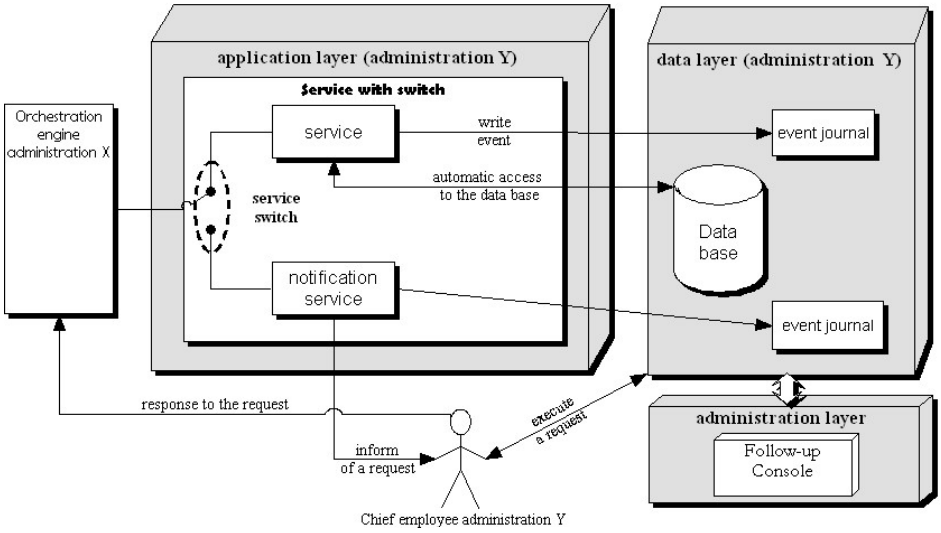

Figure 2. Web services with switches

5. Presentation of the proposed architecture

Considering the advantages service-oriented approaches can offer to the e-government applications, we choose to use them for the Tunisian e-government. Moreover, in view of our requirements, we need to separate the presentation from the application in a e-government application, to use a data layer and to add an administrative one. So, we chose to use a multi-tier architecture. Such architecture subdivides a system or an application in innumerable tiers thus, making it

possible to the various existing e-government applications to be integrated with the new ones. Moreover, these applications can be extensible [13].

The architecture that we propose for the e-government applications is represented in Figure 3. It is divided into five layers, namely: the client layer, the presentation layer, the application layer, the data layer and the administration layer. In our work, we are primarily interested in e-government Web applications considering that they agree more with the Tunisian technological infrastructure. In what follows, we detail the diverse levels of our architecture by taking Web applications as an example.

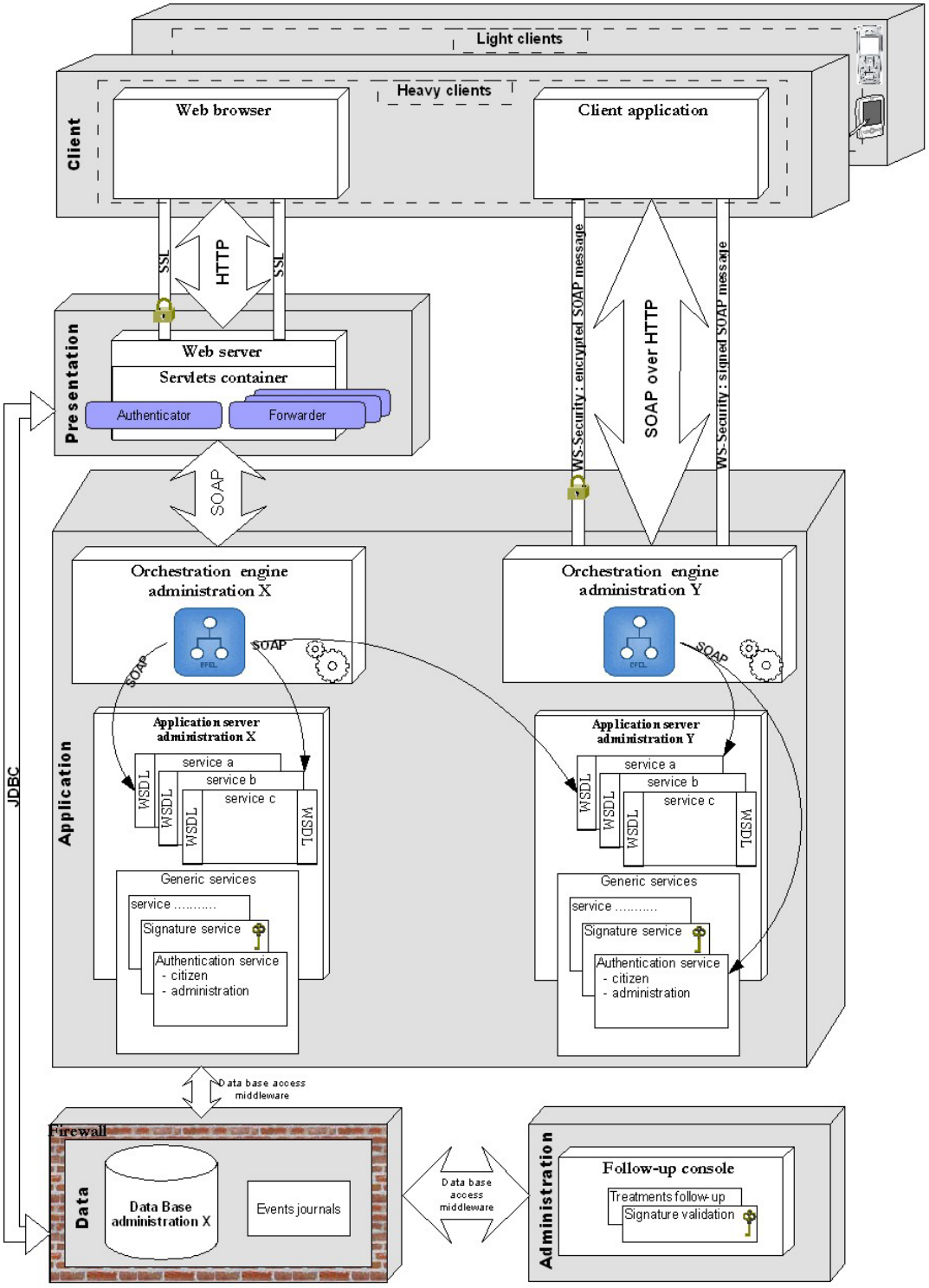

Figure 3. The proposed architecture for the Tunisian e-government

5.1. Client layer

Client layer represents the various e-government application access channels. Our architecture supports different types of clients: heavy since a computer or light via a PDA or a cellular. In our work, we are interested only in the heavy clients considering that Tunisia is not much advanced in the field of the light client applications. For the case of heavy clients, the governmental services can be reached through a Web browser, when the application is in the form of Web site, or through a self developed application invoking them.

If the client (a citizen or a governmental employee) is connected through a browser, it will interact with the application layer via the presentation layer (these two levels will be detailed lower). The communication with the presentation layer will be done using the HTTP protocol made safe by SSL. The choice of the HTTPS protocol (HTTP over SSL) comes from the increasing needs for security on the Internet. In fact, a citizen who sends personal information on the Web will feel more trustful while knowing that they are secured. Moreover, the HTTPS makes it possible to the client to check the identity of the web site to which he is connected thanks to the use of the digital certificates.

When a client invokes governmental services through an application, it will interact directly with the application layer without having to pass by the presentation layer (the presentation is achieved by the application itself).

The communication with the application layer will be done by using SOAP over HTTP. The choice of HTTP comes from the fact that it can pass through software firewalls which constituted an obstacle for the distributed applications. To preserve the confidentiality and the integrity of the exchanged messages with the application layer, we quantify and digitally sign SOAP messages according to the WS-Security specifications.

5.2. Presentation layer

The presentation layer manages the interface proposed for the clients interacting with the e-government application. For example, it contains a Web server for the customers connected via a Web browser. In addition, this level processes the data emitted and received by the clients and manages their interactions with the application layer. Separating this layer from the application layer makes the application accessible via various channels such as Web browsers; self developed applications and even cellular phones, without having to change the application's implementation. In the case of an application accessible through a Web browser, this level contains a Web server which organizes the presentation and the interactions with the application layer. The communications with the application layer will be done using the SOAP protocol over HTTP. In the case of a Web application, the Web server imbricates a Servlets[5] container. The functions achieved by Servlets are generally connecting or processing data to the data layer. In our architecture, we identified two functionalities pertaining to the presentation layer which are generic for all e-government applications: the Authenticator and the Forwarder. These two functionalities are ensured by Servlets. Authenticator authenticates a citizen so that it can access the services offered by the administration. The Forwarder manages the transmission of the information seized by the citizen to the application layer. In other words, it invokes the services (simple or composite) by transmitting, in the good format, the inputs seized by the citizen.

5.3. Application layer

The application layer is the core of the e-government application. It contains two fundamental components: the application server, and the orchestration engine.

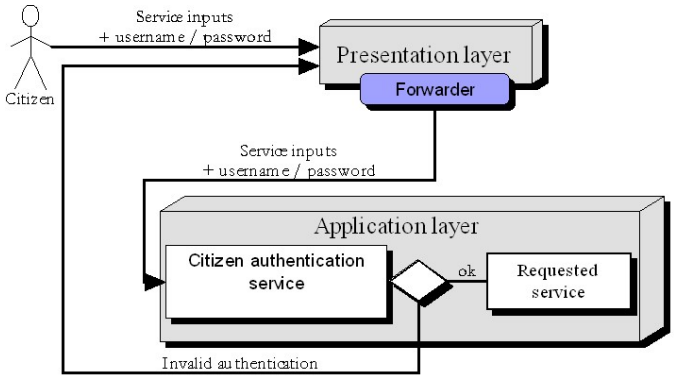

Figure 4. Generic Service: Citizen Authentication

The application server. It lodges the Web services of a governmental administration. Among these services, we identified two which will have to be generic with any application following our architecture: the signature service and the authentication service.

With regard to the *signature* service, its role is to digitally sign the certificates (under PDF[6] format) delivered by the governmental administrations. This service fulfils the citizen's acquittal requirement (see section 3.i).

[5] A java application allowing the automatic data generation within a HTTP server.

[6] Portable Document Format

In addition, the government services generally require a data base access, hence an authentication mechanism for the services consumers is necessary. These consumers are of two types: citizens or other governmental administration. In the case of a citizen, the pair «username/password» to be used is already given in the client layer. Thus, it will be transmitted to the application layer by the Forwarder. It is on this level that the authentication service checks the identity of the service requester and decides to launch or not the requested service (see Figure 4).

The administration authentication arises if an administration (administration1) wishes to use a service which belongs to another (administration2). In this case, the authentication is also done by the pair «username/password» exchanged between the two administrations before. The authentication data will be transmitted within the inputs of the service. In our architecture, all the offered services are composed of the authentication service and the service in question (see Figure 5). Thus, we are sure that no non authenticated user will be able to use a service. Moreover, the same identifier will be used for data base connection if necessary. Consequently, different access rights are allotted to each identifier.

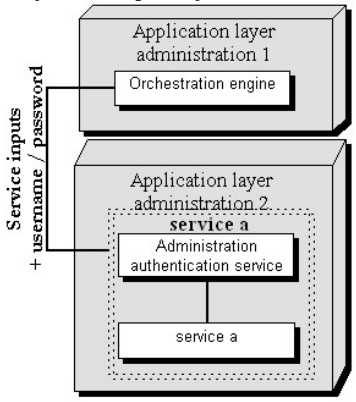

Figure 5. Generic Service: Administration Authentication

The orchestration engine. The application layer integrates also an orchestration engine which is responsible for the orchestration of the various government services implied in a composed service. As Figure 3 shows, an orchestration engine of a government administration Y can orchestrate Web services pertaining to this administration, as well as those of an administration X. In our architecture, each agency has its own orchestrator. By using an orchestration engine, the e-government applications can offer composite services to the citizens in replacement to the most complicated governmental services. Hence, the citizen provides only necessary information and it is the orchestrator who contacts the implied administrations through their Web services. As a composition language for the Web services, we propose to use BPEL [14] which represents the more credited candidate to become a standard in the field of the Web services composition.

The application layer communicates with the data layer to save/load the data by using a middleware giving an access to data bases.

5.4. Data layer

The data layer ensures the governmental administration data storage and persistence. This is carried out by using one or more data bases. For each access requester to this level corresponds a single identifier. In this way, different access rights can be allotted to the many governmental administrations. The management of the access rights to the data is ensured by the DBMS[7]. In addition, this layer lodges the Web services event journals which make it possible to preserve a trace on the invoked services.

This layer must also be protected from possible external intrusions by using a firewall. The latter, will filter the exchanged data and will block all the communications except those with the presentation, application and administration layer pertaining to the governmental administration.

5.5. Administration layer

The administration layer is only accessible to the chief employees of a governmental administration. It ensures a follow-up of the services requested by the citizens or the other administrations. This will be done thanks to a follow-up console inter-connected to the data base containing the historical of the carried out treatments. In addition, this console contains a digital signature checking module which is used to check the validity of the certificates delivered to the

[7] Data Base Management System.

citizens if necessary. The console also integrates the switches of the Web services (see section.5) used to activate or deactivate the services.

This layer is very important for the Tunisian case. Indeed, as mentioned previously, the administration's employees grant little confidence to the computed systems and feel more trustful when having a follow-up on the treatments carried out. Moreover, by using the switches, they will be able to automate or not the access to the data bases.

6. Case Study

In this section we will treat a case study to illustrate how the architecture presented in the section above can be adopted by the Tunisian governmental administrations. During our research, we contact some employees of the National Fund for Social Security (CNSS). We noticed that several social services constitute frustrating and cumbersome tasks for the citizens. These latter must often visit various agencies located in and sometimes away from their towns.

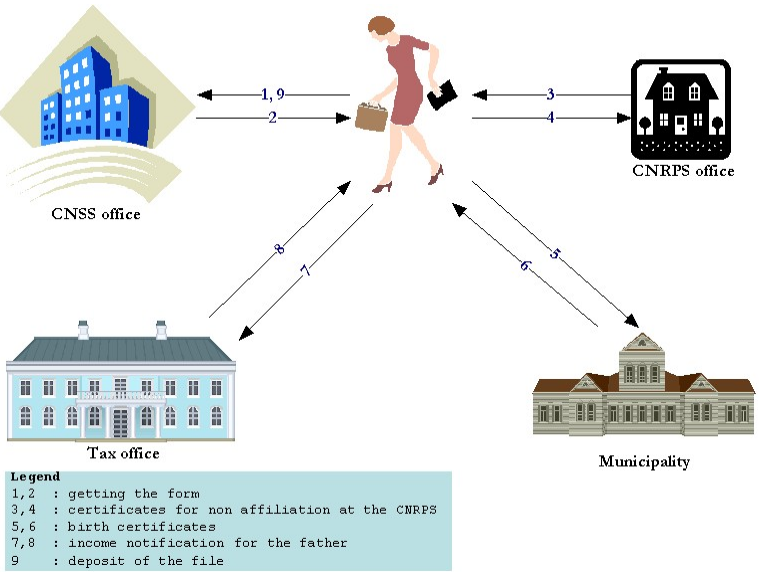

Figure 6. Undertaking a parent under responsibility

For our case study, we chose to implement the service: «undertaking a parent under the responsibility of an affiliated member». We chose that service because it requires the implication of at least four governmental administrations.

Let us take the case of a citizen wishing to affiliate his father, who is living with him, at the CNSS. First of all, he must go to the office of the CNSS, which can be away from his town, in order to take the form relating to his request. He must also go to the National Fund of Retirement and Social Welfare (CNRPS) to extract certificates of non affiliation at the CNRPS for him and his father. Then, he must extract their birth certificates at the municipality which will be used as a proof of paternity by the CNSS employee's. Afterwards, he goes to the tax office to extract an income notification for his father, since the CNSS imposes that the father should not have any income. Finally, the citizen must return to the CNSS office to deposit his file. This process is presented by Figure 6, its takes much time and requires much moving.

To implement an electronic service for this scenario, those four administrations should cooperate. Since each one has its own information system, our application has to be in favor of interoperability. Moreover, the application should also be secured since the most exchanged information are confidential, such as the income of the father.

Our case study takes in account those requirements and proposes an e-government application based on our suggested architecture. This application allows, starting from a single access point, the launch of the whole process related to the selected scenario. The citizen connects to the CNSS Web site which offers the desired service via his browser. The presentation layer of our application is responsible for transmitting to the citizen the form to be filled.

When the citizen validates his keyboarding, his request and the information he provides will be encrypted and transferred to the application layer. It is on this level that the process is launched. The request of the citizen will be treated by the CNSS orchestration engine. Indeed, the orchestration engine contains a specification in BPEL describing the process and orchestrating the various implied government services. This made up service is represented in Figure 7. The orchestration engine will invoke multiple Web services contained in distinct applications servers (application server

of the municipality, CNRPS…). Once the process is finished, a response to the request as well as a certificate of acquittal will be sent to the citizen.

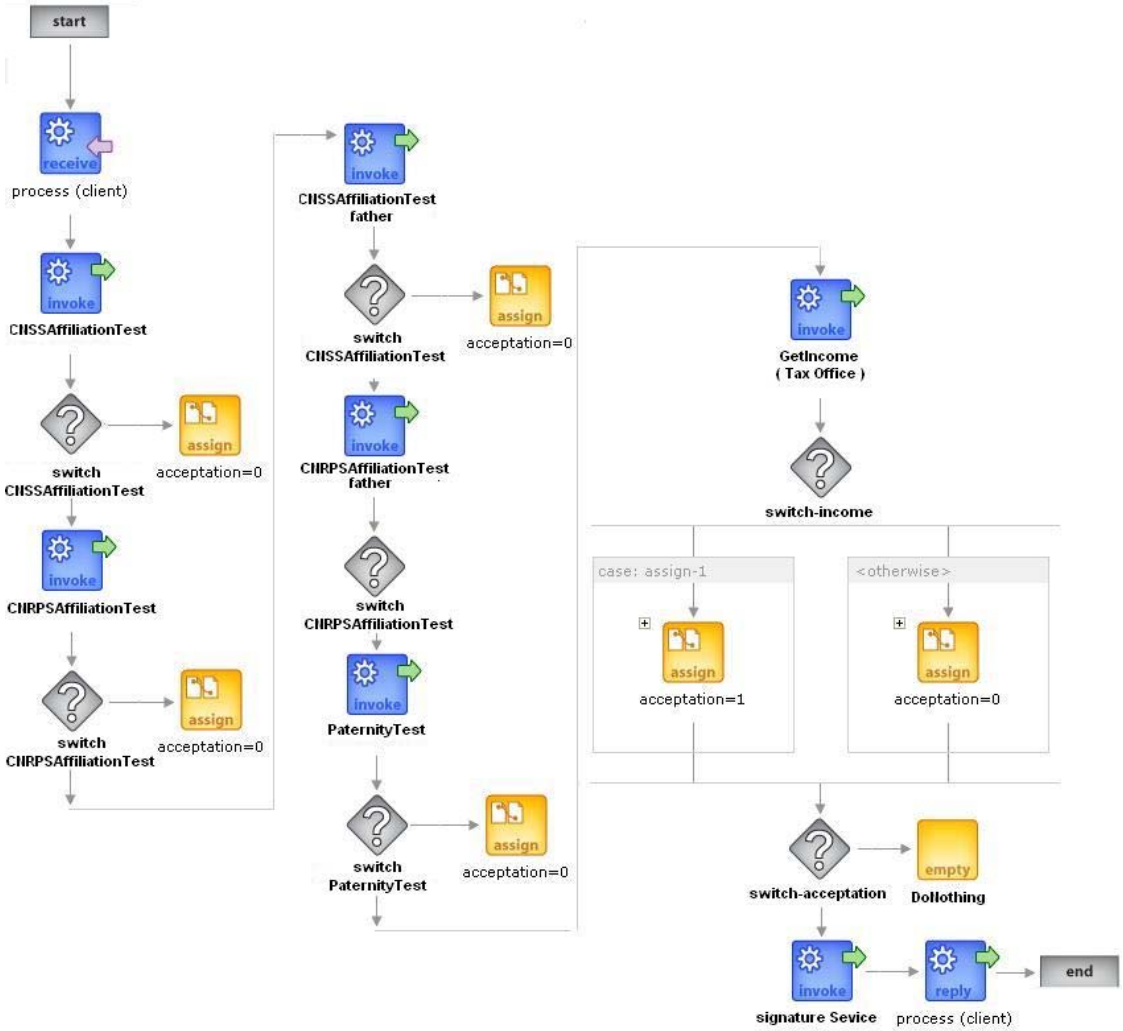

Figure 7. Graphical Representation of the BPEL process

Furthermore, since Tunisian administrations are not confidant with e-services, we should allow them to activate or deactivate them on will. On the other hand, the deactivation of a service can cause the process execution failure. To ensure that this does not happen, we replace the deactivated service with a notification one (Web services with switches).

From this case study, we can conclude that the development of e-government applications in Tunisia using a service-oriented architecture offers several advantages while respecting the requirements which we identified:

- Making it possible to the citizen to profit from various services while saving time. In the scenario presented above, the citizen will be able to affiliate his parent without having to present himself at the four administrations implied in the process of affiliation.
- Allowing a money profit for the citizen (less transport) and for the government (less paper).
- Decreasing the load of the employees in various governmental administrations.
- Allowing interoperability between the different governmental administrations in spite of their heterogeneity, thus facilitating the co-operation between them.

7. Conclusion

In this paper, we proposed a secured and service-oriented architecture for the Tunisian e-government. Then, we applied this architecture for a Tunisian case study. We are convinced that service-oriented architectures provide a solution to cross the borders between the citizens and the governmental administrations, and those between several existing administrations in Tunisia.

Our architecture was carried out while ensuring the security of the applications and with respect to the specific constraints of the Tunisian mentality. Moreover, the suggested architecture is appropriated for the Tunisian technological infrastructure and allows different heterogeneous administrations to interact together. As part of our research team's goal, we plan to pursue this work by implementing other applicative scenarios to more validate our architecture.

8. Acknowledgments

The authors would like to thank the employees working at the National Fund for Social Security of Sfax for providing us with the necessary information.

9. References

[1] EU-PUBLI, Eu-publi project, http://www.eu-publi.com, 2004.
[2] P. Hengeveld and A. Kaliontzoglou, «emayor final report», Tech. report, Deloitte, Expertnet, February 2006.
[3] B. Medjahed, A. Rezgui, A. Bouguettaya, and M. Ouzzani, «Infrastructure for e-government web services», *IEEE Internet Computing 7 (2003)*, no. 1, 58–65.
[4] D. Booth, F. McCabe, H. Haas, M. Champion, D. Orchard, C. Ferris, and E. Newcomer, «Web Services Architecture», *W3C note*, W3C, February 2004.
[5] N. Mitra, «SOAP Version 1.2 Part 0: Primer», W3C recommendation, W3C, June 2003.
[6] C.K. Liu and D. Booth, «Web Services Description Language (WSDL) Version 2.0 Part 0: Primer», Candidate recommendation, W3C, March 2006.
[7] L. Clement, A. Hately, C. Von Riegen, and T. Rogers, «UDDI Version 3.0.2, Technical committee draft», OASIS, October 2004.
[8] SICAD, «Le site de l'information et de la communication administrative à distance», http://www.sicad.gov.tn, 2006.
[9] BAWABA, «Le portail de l'administration Tunisienne», http://www.bawaba.gov.tn, 2006.
[10] CNSS, http://www.e-cnss.nat.tn, 2003.
[11] CNRPS, http://www.cnrps.nat.tn/e-cnrps, 2003.
[12] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, «Web Services Security: SOAP Message Security 1.1», Oasis standard specification, OASIS, February 2006.
[13] M. Juric, R. Nagappan, R. Leander, and S.J. XBasha, «Professional J2EE EAI», *Wrox Press Ltd.*, Birmingham, UK, UK, 2001.
[14] A. Alves, A. Arkin, S. Askary, B. Bloch, F. Curbera, Y. Goland, N. Kartha, C.K. Liu, D. König, M. Marin, V. Mehta, S. Thatte, D. Van Der Rijn, P. Yendluri, and A. Yiu, «Web Services Business Process Execution Language Version 2.0, Committee draft», OASIS, August 2006.