

Open-Source Solution to Secure E-Government Services

Claudio Agostino Ardagna

Università degli Studi di Milano, Italy

Ernesto Damiani

Università degli Studi di Milano, Italy

Fulvio Frati

Università degli Studi di Milano, Italy

Mauro Madravio

Università degli Studi di Milano, Italy

INTRODUCTION

Nowadays, a global information infrastructure connects remote parties through the use of large scale networks, and many companies focus on developing e-services based on remote resources and on interactions between remote parties. In such a context, e-government (e-gov) systems became of paramount importance for the public administration, and many ongoing development projects are targeted on their implementation, security, and release (Bettini, Jajodia, Sean Wang, & Wijesekera, 2002). For open-source software to play an important role in this scenario, three main technological requirements must be fulfilled: (1) the identification and optimization of de facto standards for building e-gov open-source software components, (2) the adoption of open-source techniques to secure e-gov services and (3) the standard integration of these components into an open-source middleware layer, capable of conveying a completely open-source e-gov solution. This article highlights that e-gov systems should be constructed on an open-source middleware layer, providing full public responsibility in its development. The role of open-source middleware for secure e-gov services deployment is discussed, focusing on implementing a security environment without custom programming. An alternative solution is given and consists of the adoption of a stand-alone architecture that fulfils all security requirements.

BACKGROUND

Accessing information on the global Net has become a fundamental requirement of the modern economy. Recently, focus has shifted from access to traditional information

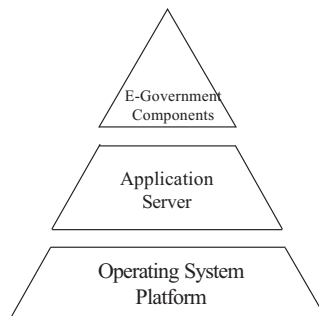
stored in WWW sites to e-services such as e-gov services, remote banking, or airline reservation systems (Corallo et al., 2005; Feldman, 2000).

In particular, the perceived importance of e-gov services is growing (Ardagna, Cremonini, Damiani, De Capitani di Vimercati, & Samarati, 2005). Today, e-gov services provide various kinds of functionalities to individual citizens and companies alike (e.g., paying fines, answering registry office requests, releasing authorizations, and so forth). It is widely acknowledged that e-gov services should follow a layered software structure that is outlined in Figure 1 (Ardagna, Damiani, Frati, and Montel, 2005; Damiani, Grosky, & Khosla, 2003). The structure depicted as follows is composed of three layers, where the first comprises a community of software components, managed by an application server (second layer) in turn running on an operating system platform (third layer). Specifically, the three levels are:

- **E-Government Components Layer:** All software components that implement e-gov services.
- **Application Server Layer:** Middleware over which the application will be deployed. It provides some additional functionalities such as management of security, clustering and persistence. In this manner, the developer can focus on the implementation of application components.
- **Operating System Platform Layer:** The operating system platform over which the application will be distributed.

Many proprietary e-gov projects are affected by budget, transparency, vendor lock-in, integration, and interoperability limitations that represent main crucial problems affecting e-gov applications; also, the European public

Figure 1. E-Government's three-layered structure



administration sector is increasingly reluctant to depend on proprietary solutions for e-gov services and highly support the introduction of open source in public administration.

The introduction of open-source approach can supply above limitations. First, although open-source software is not necessarily free, in most cases is considerably cheaper than proprietary software, giving economic advantages for the public sector.

Second, open-source software, often, adheres to open standards, which means that software development could influence standard and it is conducted in public forums, in contrast to proprietary approach.

Third, in e-gov scenarios, data are exchanged between involved entities and then it is necessary to rely on secure systems to protect the parties against external checking, and/or undesirable release of information to unauthorized users. Security and quality of open-source code, hence, become the major requirements in e-gov environment. Open-source code enables users and other interested parties to check whether the program is written in a secure way and to identify dangerous elements for stability and security of the applications.

Then, open-source paradigm also guarantees supplier independence. A lock-in situation, in fact, arises when software is purchased and depends on when and how the supplier modifies the product. This lock-in problem does not affect open-source software. Data is not stored in a proprietary format, and it is possible for users to change between several different systems and therefore also between several different suppliers.

Finally, customization and re-use are simply addressable because source code is freely available and modifiable; hence, public authorities can adapt open-source software to their particular needs, resulting in lower customization costs.

At this point, we can argue that an important non-functional requirement for e-gov projects could be to implement the entire application following the open-source approach.

However, many ongoing projects do not completely satisfy the above requirement because only the first and

third layer, of the structure depicted in Figure 1, are developed following open-source paradigm, while the middleware is, often, a commercial application server, subject to a license and all the limitations that this license implies. For example, the Italian project PEOPLE (<http://www.progettopeople.it>) is aimed at providing local authorities with a complete set of open-source software components that run on the Linux operating system.

This approach is claimed to provide an open-source e-gov environment; still, experience has shown that in a deployed system a substantial amount of the executable code, including the one implementing important functionalities such as access control, authentication, and so on, usually belongs to a proprietary application server.

In the following is showed how a complete open-source e-gov application can be secured without custom programming. It is firstly presented a solution that relies on open-source middleware functionalities and then a solution to develop a stand alone framework that could be applied to implement authentication services for users that require e-gov services.

OPEN-SOURCE APPLICATION SERVERS IN E-GOV APPLICATIONS

With the rising of popularity of e-services, there is an increasing need for dynamic, maintainable, secure, and scalable programming platform. The J2EE specification has been introduced to satisfy this requirement. Therefore, an application server based on this standard is able to provide all the needed services to build a modern application. Most e-gov applications heavily rely on middleware that provides several "horizontal" functionalities, that can be used in an e-gov environment to simplify and improve developed applications, such as security, persistence, clustering, transaction, and cache management.

JBoss (JBoss, 2005; Scott, 2003), an open-source fully J2EE compatible application server, is one of the most widely accepted implementation available on the net and represents an interesting solution for e-gov services. One of the most important characteristics of JBoss is its highly modular design.

The adoption of JBoss functionalities to implement a security environment provides clear separation between business components and security modules and does not require any kind of code customization.

In particular, using JBoss features, it is possible to implement a complete and secure access control environment by the configuration of JBoss XML descriptor files and modules only.

Access control is a fundamental part of any e-gov application and must support the definition of restrictions

4 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/open-source-solution-secure-government/11671

Related Content

Palming the Future: E-Government Strategy Development for a Tertiary Education Organisation

Judith Symonds (2007). *International Journal of Electronic Government Research* (pp. 62-74).

www.irma-international.org/article/palming-future-government-strategy-development/2031

Electronic Government Systems for e-Procurement Procedure in the EU

Sergio Bravo Martínand Francisco José García Peñalvo (2013). *E-Procurement Management for Successful Electronic Government Systems* (pp. 29-51).

www.irma-international.org/chapter/electronic-government-systems-procurement-procedure/69586

Factors Influencing Citizens' Intention to Use E-Government Services: A Case Study of South Korean Students in China

Isaac Kofi Mensah, Mi Jianingand Dilawar Khan Durrani (2017). *International Journal of Electronic Government Research* (pp. 14-32).

www.irma-international.org/article/factors-influencing-citizens-intention-to-use-e-government-services/181279

I-FGM as a Real Time Information Retrieval Tool for E-Governance

Eugent Santos Jr., Eunice E. Santos, Hien Nguyen, Long Pan, John Korahand Huadong Xia (2008). *International Journal of Electronic Government Research* (pp. 14-35).

www.irma-international.org/article/fgm-real-time-information-retrieval/2043

Anti-Corruption Capabilities of Public E-Procurement Technologies: Principal-Agent Theory

Arjun Neupane, Jeffrey Soarand Kishor Vaidya (2014). *Technology Development and Platform Enhancements for Successful Global E-Government Design* (pp. 185-203).

www.irma-international.org/chapter/anti-corruption-capabilities-of-public-e-procurement-technologies/96696