

# A case study in the use of Open Source technologies for the Australian Competition and Consumer Commission's new Internet and Intranet Web portals

Peter Bailey, Richard Bennett-Forrest, Matthew Sheppard, Nathan Wallace  
Synop Pty Ltd  
PO Box 1024  
Artarmon NSW 1570 Australia  
{peter,richard,matthew,nathan}@synop.com

Justin Freeman  
Agileware Pty Ltd  
PO Box 25  
Charnwood ACT 2615 Australia  
justin@agileware.net

Nigel Martin, Fiona Urquhart  
Australian Competition and Consumer Commission  
PO Box 1199  
Dickson ACT 2602 Australia  
{Nigel.Martin,Fiona.Urquhart}@accc.gov.au

Mark Nearhos  
SecureNet Ltd - a Betrustrusted company  
Ground Level,  
243 Northbourne Ave  
Lyneham ACT 2602 Australia  
mnearhos@securenet.com.au

December 12, 2003

## **Abstract**

In 2001, the ACCC recognised a business need to replace their existing Intranet and Internet websites. The winning consortium (SecureNet Ltd and Synop Pty Ltd) provided a solution using many Open Source technologies.

The Sytadel 4.0 content management framework is written completely in PHP 4.3, and stores all content natively in XML, into a relational database. At the ACCC, the database technology used for Sytadel is MySQL 4. All servers run Linux. The web server used is Apache 1.3.

Throughout the course of the project, the Synop development team experimented with a range of different XML library utilities that interoperate with PHP. The final choices were libxml2 and libxslt, although Sablotron and Xalan/Xerces had also been tried. Other Open Source software projects that were used include: curl, expat, modssl, openssl, xpdf, and zlib.

Sytadel is a proprietary product, but full source code is provided to customers to ensure that they have access to the code to solve any issues that arise themselves.

The ACCC as a regulator of competition is a strong supporter of different forms of software technology being made available to Government, without endorsing one particular technology over another. The project discussed in this paper serves as an excellent demonstration that Open Source technologies can play an effective role in Government IT projects.

*Keywords:* Open source; Government; Critical information infrastructure

## 1 Introduction

This paper details a practical demonstration of a major project built on Open Source technologies for a Government client. We discuss:

- the background to the project;
- the motivation for using Open Source software components;
- a list of the many underlying Open Source software components used and what they were used for
- Sytadel's page delivery architecture;
- statistics on the project;
- issues of securely deploying systems using Open Source software components;
- a commentary on the challenges and benefits of using Open Source software components when developing a major new system;
- the role of Open Source technologies for critical information infrastructure; and
- the approach required to deliver a successful project.

## 2 Background to the project

In 2001, the ACCC recognised there was a business need to replace their existing Intranet and Internet websites. They set about determining their requirements for the replacement of systems, and released an open tender to the market in March 2002. A consortium of SecureNet Ltd as prime and Synop Pty Ltd won the tender, with a solution using significant amounts of Open Source technologies. Activities commenced in mid 2002; the system was in active use within the ACCC in March 2003, and operational changeover occurred in December 2003.

The central Sytadel 4.0 content management framework is written completely in PHP 4.3 [5], and stores all content natively in XML [14], into a relational database. This latest version of Sytadel represents a substantial evolution in its sophistication. Previous versions of Sytadel did not use XML anywhere. Sytadel is now XML-reliant – it could not operate without XML for data storage, and associated XML technologies are used for other parts of the system.

Sytadel provides full version control of content and compliance with relevant Australian Government Online Information System Obligations. The ACCC has substantial requirements with respect to meeting these obligations, as an Australian Government agency.

The ACCC as a regulator of competition is a supporter of different forms of software technology being made available to Government; without endorsing one particular technology over another.

### 3 Motivation

There were a number of motivations for using Open Source components and technologies in this project.

First, Synop has long experience in working with many Open Source technologies. Synop's Sytadel CMS product line has been implemented using PHP since 1999. It is a classic example of using the LAMP (Linux/Apache/MySQL/PHP) web application server platform as a basis for complex web-based application software development.

Second, the savings represented by using Open Source technologies are substantial. A single copy of Microsoft's SQL Server Enterprise Edition for a 1 processor licence is currently listed on Microsoft's website for approximately \$37 000. At least two databases are required for the ACCC system: one for the Intranet and one for the Internet due to security reasons with the deployment. Thus the minimum cost would be \$74 000 (assuming no discounts). By contrast, MySQL is free.

Third, various factors are driving the world towards open standards such as XML for data storage and/or interchange. The adoption of these kinds of standards is an important future-proofing decision with respect to system upgrades or replacements that may be required in years to come.

Fourth, the Open Source nature of the software used in various security packages on Linux means that security flaws can be fixed rapidly and independently of any single Independent Software Vendor (ISV). This reduces reliance on the ISV provider.

### 4 Open Source technologies in use

At the ACCC, the database technology used for Sytadel is MySQL 4 [11]. All servers run Linux. The web server used is Apache 1.3 [3]. (Throughout this document, we simply provide the major version numbers; not the minor revision numbers that have been involved.)

The solution uses three servers: an internet server, which runs Apache, Sytadel and MySQL; an intranet web server, which runs Apache and Sytadel, and an intranet database server, which runs MySQL only. The three servers are located in SecureNets accredited secure hosting facility.

Throughout the course of the project, the Synop development team experimented with a range of different XML library utilities that interoperate with PHP. We were using XML for data storage formats and manipulation, and XSLT for data transformations as part of the presentation layer. PHP's support for XML was under constant evolution; the project started with version 4.2 and is now using the 4.3 series.

Throughout this time the XML support changed substantially. Initially the Sablotron XSLT library was used; however we found it to be insufficiently compliant with the XSLT standards [13] to be relied upon. Our next step was to use the reference library standards Xalan/Xerces, developed by the Apache Software Foundation. We had to develop our own PHP interface to these libraries, where again the Open Source nature of the technologies was essential to our ability to fashion the interface in a very short timeframe. However, the performance of these libraries was not optimal, and having to maintain our interface to them was problematic. The final choices were libxml2 [7] and libxslt [8]. These libraries are extremely fast, reliable, and sufficiently compliant with the published XML and XSLT standards for the needs of the project.

Other Open Source software projects that were used include:

- curl [6] - used by PHP for opening https connections
- expat [10] - also used by PHP for XML parsing
- modssl [4] - used for secure https connections to the web server

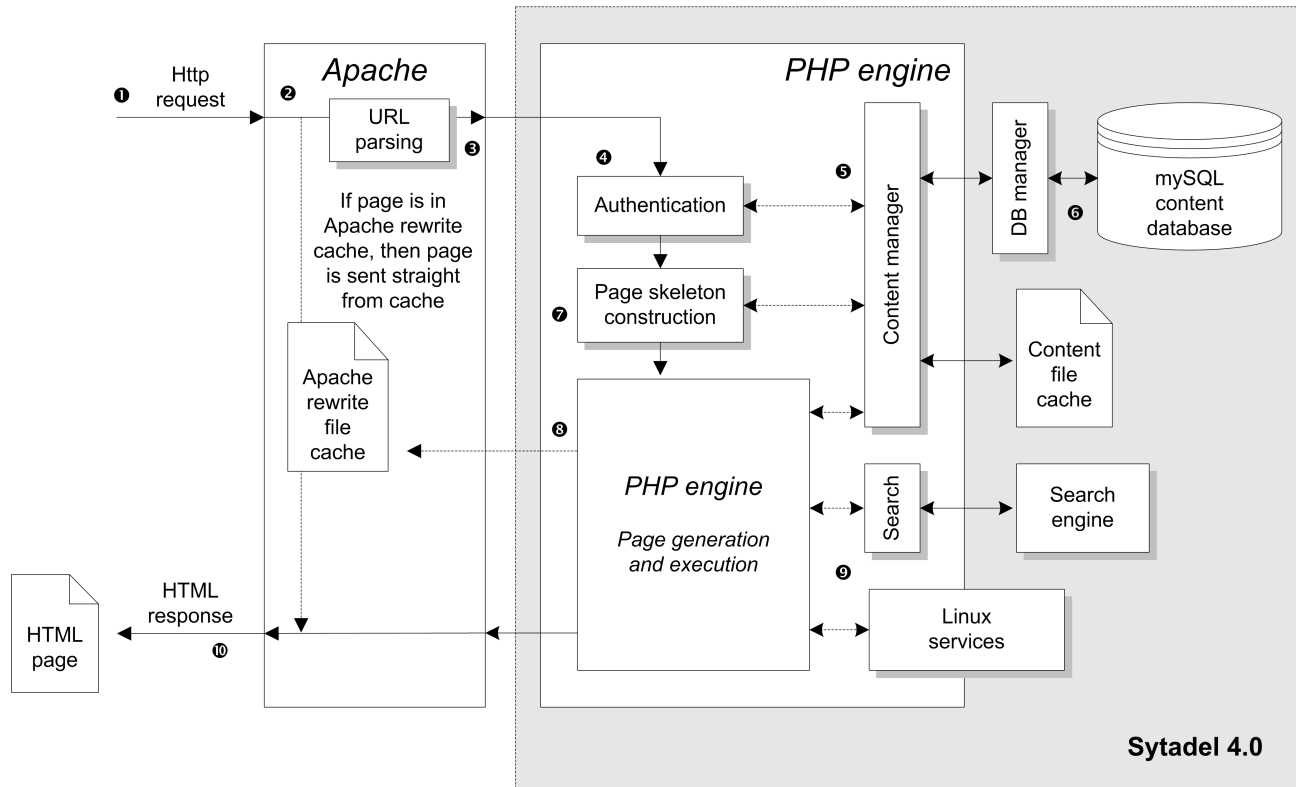


Figure 1: Sytadel page delivery architecture.

- openssl [1] - used by modssl
- xpdf [9] - used to convert PDF files to text for indexing by a search engine
- zlib [2] - used for output compression by PHP

## 5 Sytadel page delivery architecture

Sytadel is a proprietary product, but full source code is provided to customers to ensure that they have access to the code to solve any issues that may arise themselves, if they choose to be responsible for the system without any assistance from Synop.

The Sytadel system and the underlying architecture are highly scalable. Sytadel is a true micro-content based CMS – every single piece of information is stored within the repository as its own XML document.

Individual page views may consist of the amalgamation of on average 60 separate pieces of content. Each content item must be checked for security (the permission to view it and/or to carry out other actions on it), and rendered by an XSL transformation for display. Finally, all the separate rendered content displays are glued together by a templating facility.

An overview of how pages are delivered in response to a request is shown in figure 1.

1. Incoming http request.
2. If the request/user combination is currently in the cache, then Apache rewrite rules are used to redirect the request to a static copy of the page on disk, thus bypassing Sytadel altogether.

3. The URL is parsed into session variables by Apache. We tend to use /parm/1/parm/2 URLs instead of CGI parms to promote search engine indexing.
4. The request is passed to Sytadel, where it is first authenticated.
5. Everything (code, users, security, page definitions etc.) is a content object, so users are authenticated against member objects in the content database. The content manager decides whether to instantiate the content from the database, or to pull it in from the local cache, which it also maintains. The content manager also flushes indirectly related content as appropriate.
6. We have database wrappers for MySQL and Oracle; other database wrappers can be developed.
7. Based on the user and various system settings, we dynamically and recursively build a page layout model which constructs a page of “boxes” within “boxes” which contain content. The resulting skeleton is legal PHP source code, ready to be executed.
8. The PHP source code for the page is compiled and executed (using standard PHP functions). This code then pulls in the content required for each box on the page, which may in turn execute PHP code pulled in from code which is content. It is possible to recursively run the PHP compilation and execution steps several times, depending on the pages content. All HTML is generated using XSLT, including page layout, individual content rendering and most code execution. The final HTML page is also stored in the Apache rewrite cache.
9. Page generation may optionally call system services such as mail and file system functions, or call the search engine for search results. We have a wrapper for plugging in different search engines. The indexing is done through standard http requests and so is not included in the diagram.
10. The final HTML page is returned to the browser.

## 6 Project statistics

The current ACCC Intranet web portal has:

- 133 499 separate content versions;
- approximately 4-5 version records for each content version – which record changes to the version throughout its content lifecycle;
- 2 156 775 log messages – which record information about the actions taken for audit purposes.

One of the reasons for why there are so many content records in the repository is that every content item in the system has a full version history. Each content item may have one or more content versions, created over time. The log records ensure that an audit trail of every action that occurs to each content version can be traced. This facility is required in part to meet the OISO requirements on auditing.

## 7 Secure deployment of systems with Open Source components

The benefits of planning and integrating security from the start of the portal development were recognised as being more efficient than treating security as an add on. To this end, a security audit including a Threat and Risk Assessment was performed after the proposed architecture was confirmed at the start of the project.

This process involved the identification, analysis and evaluation of the security risks associated with the Internet and Intranet portal infrastructure (including the software components), and associated business and supporting processes. This enabled the development of risk mitigation strategies to protect the integrity, confidentiality and availability of the portal system.

A second Threat and Risk Assessment was performed just prior to the implementation to assess any changes arising through the development process and to reassess the earlier strategies.

The security audits were used to identify potential weaknesses in the operating system components that were deployed. The Linux operating system distributions were hardened, by removing various facilities that might lead to unnecessary vulnerabilities. Combined with the use of readily validated Open Source software components, application of the latest patched versions of critical software, and an appropriate design for the system architecture, the servers are well protected against malicious attack.

## 8 Challenges and benefits of using Open Source components

There are both benefits and challenges from using Open Source components in a major project such as the ACCC's replacement Internet and Intranet web portals.

### 8.1 Challenges

- A proliferation of different components that must be connected, from different Open Source projects, with relatively frequent changes and inter-dependencies.
- Lack of standardised configurations of the different components, which are internally self-consistent (note, this in part is what the Debian project has sought to address).
- The lack of integration across multiple components makes integrated development environments, capable of traversing the scripting/application layer down to the core web server and associated libraries, difficult, if not impossible, to obtain; this slows development progress.
- Rapid review and feedback cycles produced a large number of change requests. Bugzilla was used to manage the administration of these requests and to ensure that all were tracked and concluded.
- Acceptance of the use of Open Source software components by an existing IT outsourcer and IT staff was initially a challenge. An internal education process resolved these issues which were mainly based on lack of information.
- Unknown problems can arise from using relatively “new” and/or “untested” software in a production environment. Risk mitigation of these issues is extremely difficult.

## 8.2 Benefits

- Ready access to an international community of users and developers who respond quickly and efficiently with bug fixes.
- On some situations, a variety of alternative packages may exist which serve the same purpose, allowing selection or replacement of packages.
- Ability to track bugs down to individual packages and lines of source code, and to modify source code if required.
- Innovative features: rapid adoption of new innovative features and implementation into production.
- Education: availability of open source development tools and education material has enabled ACCC IT staff to self-educate and support the new Open Source platforms and languages. Continuing in-house development is now possible.
- Vendor and platform independence: Open Source software has enabled the ACCC to implement software independent of any specific vendor or platform.

## 9 Open Source technologies and critical information infrastructure

The new Intranet and Internet Web portals for the ACCC form some of the most critical information infrastructure at the organisation. The Internet portal is the public online face of the organisation, which is used by many different organisations and people every day to stay up to date with competition and consumer information. The Intranet is the hub of information dissemination for the organisation, and integrates a number of the other existing enterprise systems.

Using Sytadel and the underlying Linux operating system environment enabled the ACCC to connect these widely disparate systems, which include a human resources (proprietary) system (AURION), the TRIM records management (proprietary) system, and a number of custom ACCC applications developed on top of an Oracle database platform.

PHP is an ideal language for connecting systems together, due to: its simple and rapid coding facilities; its many existing modules for inter-system connections; and the large and helpful developer community that exists for PHP, which increases the probability that other people have solved similar problems already.

At the ACCC, ultimately all the enterprise systems used the Oracle database to store information. This allowed Sytadel to connect directly to the database through its database abstraction layer, with an Oracle connector we developed. The TRIM system also stores certain files in an FTP repository; Sytadel was able to access these directly based on the information retrieved from the database about the file locations.

A more sophisticated approach will be required when interactions with TRIM are two-way. In this event, a web services interface to the TRIM API running on a COM or DCOM server would be the obvious way to support interactions. The current version of TRIM in use at ACCC only supports a COM interface, and although PHP has a COM interface module, it would require the Sytadel system to be running on the same server (a Microsoft Windows 2000 server) as COM does not support distributed connections.

## 10 Approach for successful projects involving Open Source software components

To be a success, the consultant team needs to be familiar with the software components it needs to integrate. Software components that are actively supported and have a proven track record for the intended purpose should be selected. The consultants must also be ready to dive into finding problems with the components if they arise, and to seek support from and participate with the community which exists around each component. Synop for example created and maintains the FAQTs website [12] which provides a collaborative knowledge forum for various communities, including those for PHP, Javascript, and Python.

The key to successful projects involving Open Source software components is effective teamwork between all parties. Just as Open Source software is often developed in a distributed community, so a large scale project will typically involve a distributed community of the client and the consultant team members. There are many different client needs, and similarly different abilities and skills that the consultant team must bring to the project. Some of these skills are technical; others are about the personal interactions between team members.

Relatively little in these comments mark such a project as distinctively different from a project involving only proprietary software components. Perhaps the major noticeable difference is one of attitude – since most if not all of the software is freely open to the client, there tends to be a corresponding approach to being inclusive in communications between the parties. This approach fosters an open collaborative nature to the project; which in turn leads to a trust relationship, and an ability to resolve problems with mutual understanding.

## 11 Conclusions

In conclusion, the project has been a substantial success and was enabled through the careful adoption of Open Source components and technologies. Open Source technologies helped to keep the total cost of ownership down by many thousands of dollars. They also allowed changes and fixes to be made as the system evolved throughout development and deployment.

The inclusion of source code with the Sytadel CMS has provided a platform for future development of Open Source modules that can be developed in house for or by the ACCC, thereby providing an opportunity to reduce ongoing costs of system build and maintenance. The project discussed in this paper serves as an excellent demonstration that Open Source technologies can play an effective role in Government IT projects.

## 12 Acknowledgements

This project could not have been a success without the many thousands of developers who have contributed to the different Open Source software components used by us to develop and deploy these web portals. We gratefully acknowledge the hard work that has gone into all of these separate projects.

## Bibliography

- [1] OpenSSL: The Open Source toolkit for SSL/TLS. <http://www.openssl.org/>.
- [2] Zlib. <http://www.gzip.org/zlib/>.
- [3] Apache Software Foundation. Apache 1.3 web server. <http://httpd.apache.org/>.



- [4] Apache Software Foundation. mod\_ssl: The apache interface to openssl. <http://www.modssl.org/>.
- [5] Apache Software Foundation. PHP (4.3) hypertext processor. <http://www.php.net/>.
- [6] Daniel Stenberg. cURL and libcurl. <http://curl.haxx.se/>.
- [7] Daniel Veillard. Libxml2: The XML C parser and toolkit of Gnome. <http://xmlsoft.org/>.
- [8] Daniel Veillard. Libxslt: The XSLT C parser and toolkit for Gnome. <http://xmlsoft.org/XSLT/>.
- [9] Glyph and Cog. XPDF. <http://www.foolabs.com/xpdf/home.html>.
- [10] James Clark. The Expat XML Parser. <http://expat.sourceforge.net/>.
- [11] MySQL AB. MySQL 4 relational database. <http://www.mysql.com/>.
- [12] Synop Pty Ltd. FAQs: Get the faqts, and nothing but the faqts. <http://www.faqts.com/>.
- [13] W3C. XSL Transformations (XSLT) 1.0. <http://www.w3.org/TR/1999/REC-xslt-19991116>, November 1999.
- [14] W3C. Extensible Markup Language (XML) 1.0 (Specification). <http://www.w3.org/TR/2000/REC-xml-20001006>, October 2000.