



# Diploma Project

## Telecommunication equipment management using web services

### Embedded Server

Version V1 (Emb.S)

**Professeurs:**

Philippe Joye

François Buntschu

**Mandatory:**

Daniel Gachet

**Expert:**

Nicolas Mayencourt

**Students:**

Thierry Kiki

David Schneider



## Table of Contents

1	Introduction .....	3
2	List of few embedded servers.....	4
3	Comparative table of few servers.....	6
3.1	Free servers.....	6
3.2	Shareware Servers .....	6
3.3	Selective criteria of servers.....	7
3.4	Required fields.....	7
3.4.1	Security.....	7
3.4.2	Dynamic Content.....	7
3.4.3	Common functionality.....	8
3.5	Classification of compared servers.....	8
4	Suggestions.....	9
4.1	Web server with incorporated module C/C++.....	9
4.2	Web server using SSI / SSJ .....	9
4.3	Web server with incorporated module CGI.....	9
5	Conclusion.....	9
6	Annexes.....	9
6.1	Revision history.....	9
6.2	References .....	9



# 1 Introduction

An embedded web server works the same way a standard web server does. The difference lies in its performance (usage, resources, memories, etc...) because it is downloaded on devices.

It allows to access, communicate and control remotely the device through a traditional browser or a particular interface of communication.

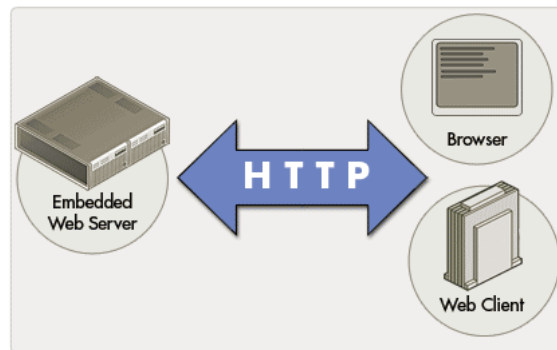


Figure 1 : general illustration

Dans le cas de notre projet, le serveur web sera embarqué sur la core unit card (carte n°11) qui n'offre que 256 MB de RAM et 128 MB de mémoire flash au maximum.

La ROM (flash memory) ne sera pas du tout utilisée car toutes les web pages seront non seulement générées dynamiquement mais ne seront pas sauvegarder. Ceci implique que la page sera systématiquement détruite une fois que l'utilisateur ne l'aura plus en visuel. Seul le code permettant de générer la page restera dans la RAM.

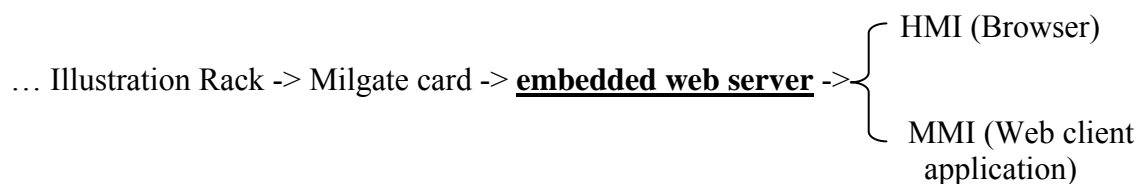


Figure 2 : Milegate case

## 2 List of few embedded servers

On the table below, there is a list of several servers designed for different usage. There are web servers that can be embedded on devices.

Overview									
Server name	O.S	Dev. Language	Generate dynamically web pages	License / Open Source	Shareware / free	Multi-thread / Poll	Code/Memory Footprint (KB)	Widely used	Release Date (Stable Version)
AppWeb	Embedded Linux...	C++	Yes	Dual License* / yes	✓ / ✓	✓ / ✓	??/800	😊😊	2008-03-14 (2.4.2)
GoAhead	Embedded Linux...	C	Yes	Dual License / yes	✓ / ✓	- / ✓	??/60	😊	2003-12-02 (2.1.8)
Klone	Embedded Linux...	C	Yes	Dual License */ yes	✓ / ✓	- / ✓	??/ 110→350	😐	2009-03-06 (2.2.0)
NicheStack HTTPServer	Any 16 or 32bit embedded	C	Yes	Closed source	✓ / -	- / ✓	??/??	😞	???????? (3.1)
Barracuda Web Server	Embedded system (Linux...)	C	Yes	Dual License / No	✓ / -	✓ / ✓	??/?? (Small) **	😊	
RomPager	Any Operating system	C	Yes	Commercial License / No	✓ / -	- / ✓	**	😊😊	
Fusion http Server	Embedded devices	C	?	Commercial License / No	✓ / -	✓ / ✓	??/ 7→11	😞	?

Server name	O.S	Dev. Language	Generate dynamically web pages	License / Open Source	Shareware / free	Multi-thread / Poll	Code/Memory Footprint	Widely used	Release Date (Stable Version)
Boa								☹	
Lighttpd						- / ✓		☹	
Allegro								☹	
VQEmbWeb	Any OS with support for JAVA 5							☹	
Kolibri/+		C++		GPL & close source				☹	
Obelisk-Http	Any OS with Python	Python						☹	
WT	Embedded Linux...	C++						☹	

\* = GPL and commercial licenses

\*\*= The official web site writes “small footprint” but doesn’t precise how small is it.

### 3 Comparative table of few servers

#### 3.1 Free servers

Features													
	Security			Dynamic Content					Common functionality				
Server name	Authentication (Basic, digest..)	SSL& which SSL stack	Open SSL	CGI / Fastcgi		PHP	SSI/SSJ	C /C++	Session / Cookies	Compression / chunking	IPV6		
AppWeb	Basic & digest authentication			- / ✓		No	- / ✓						
Klone				yes				yes					
GoAhead													
Lighttpd													

#### 3.2 Shareware Servers

Features													
	Security			Dynamic Content					Common functionality				
Server name	Authentification (Basic, digest..)	SSL& which SSL stack	Open SSL	CGI / Fastcgi	WSDL compiler / SOAP API	PHP	SSI/SSJ	C /C++	Xml → C compiler	Compression / chunking	IPV6		
Baraccuda		SharkSSL						Yes					
RomPager													
NicheStack HTTPServer		NicheStack SSL											
Fusion Web <sup>2</sup>				✓ / -	✓ / ✓				Yes				



### **3.3 Selective criteria of servers**

To compare different servers, we narrowed the long list of « light » servers. That list is presented in 2. We considered the following points to come to that conclusion:

- Popularity: A popular application has an important FAQ and many forums that solve an array of problems.
- Last release date : For security reason it is important to know how often the server is updated and the availability of updated and new versions.
- Code & Memory footprint: As the server will be used for embedded devices, it is very important to know the size and space the application needs to run as well as its effect on the processor.
- Development language: According to KEYMILE' requirements, the development language can only be C or C++.
- Operating System: The server will support at least an embedded linux.

However, many of these servers are also available for Vxworks, Windows, Mac, Unix, FeeBSD, Posix and so on...

### **3.4 Required fields**

To properly compare different servers, we define three main domains of assessment. They are:

- Security
- Dynamic content
- Common functionality

#### **3.4.1 Security**

Three sub catégories exist:

- Authentication: we evaluate the type of authentication that the server offers - Basic or Digest. The possibility to connect to an authenticated server such as RADIUS or KERBEROS is not considered because KEYMILLE already fulfilled that requirement.
- SSL/TLS: It is important to know if the server offers a secure connection (SSL/TLS layer over HTTP) or not. In case it supports SSL/TLS, it is necessary to know if it is possible to use Mocana SSL Stack to encrypt exchanged messages because KEYMILE already has bought a license of this software.
- OpenSSL: The possibility to use OpenSSL (compatibility) so it remains free and more importantly OpenSource.

#### **3.4.2 Dynamic Content**

This part informs on the technology to use to generate dynamically information that the server will make available. They are mainly:

- CGI (Common Gateway Interface): with the help of a program written in any language (C/C++, Perl, Python...) it allows to generate information to be published through data provided by the user (client).
- FastCGI: An Advanced form of CGI.



Each http request is processed in a unique way with the standard CGI. Therefore, the server is overworked whenever there are important activities (many requests means many process). It is for that reason that FastCGI was built to process a bigger number of http requests.

- SSI/SSJ (Server Side Includes / Server Side JavaScript).
  - SSI is a programmable technical for web application. It allows helping and building an HTML document from several documents. In fact, it is a code generated out of the thin air and included in an HTML page when browsing other documents. The server later translates the HTML document and displays it to the user.
  - SSJ it is simply a JavaScript for a server. It is increasing in usage as it allows to generate a script by the server for the client. The script is sent and executed on the user's interface (browser), and then displayed information according to the results obtained on the user's device.
- C/C++

....Trouver ou schématiser une illustration .....

Figure 3 : Dynamic content representation

### 3.4.3 Common functionality

We tried to group other valuable data:

- Compression : Possibility to compress the exchanged messages
- Segmentation or chunking: Useful when exchanged messages are bigger than 1500 Octets.
- Session / cookies: Possibility to use sessions and manage cookies.
- Documentation
- Support IPv6

### 3.5 Classification of compared servers

The following tables are used to classify the different servers.

	C/C++	Support Nano SSL	Compress / Chunking	Other SSL Stack	Auth : B /D	Free/Open Source (F/O)	
points	3	3	1/3	1	1/1	1/3	

Memory footprint	< 200 KB	< 400 KB	< 600 KB	< 800 KB	< 1 MB
points	8	6	3	2	1





Servers	C/C++	Nano	Com/chunk.	Other ssl	Auth B/D	Footprint		F/O	Total
AppWeb									
Klone									
GoAhead									
Lighttpd									
RomPager									
Barracuda									

## 4 Suggestions

### 4.1 *Web server with incorporated module C/C++*

Coming soon

### 4.2 *Web server using SSI / SSJ*

Coming soon

### 4.3 *Web server with incorporated module CGI*

Coming soon

## 5 Conclusion

Coming soon

## 6 Annexes

### 6.1 *Revision history*

Doc ID	Revision		Short description of the modification	Prepared by	Checked by	Approved
	Version	Date				
Emb.S	V1	06-04-09	First Draft (Incomplete)	T. KIKI		

### 6.2 *References*

- <http://www.appwebserver.org/>
- <http://www.goahead.com/products/webserver/Default.aspx>
- <http://www.koanlogic.com/klone/features.html>
- <http://www.allegrosoft.com/rpproduct.html>
- [http://barracudaserver.com/Barracuda\\_web\\_server\\_SDK.html](http://barracudaserver.com/Barracuda_web_server_SDK.html)
- <http://www.iniche.com/webport.php>