Design and Implementation of a Lightweight and Efficient Embedded Web Server for Web-based Network Element Management

Mi-Joung Choi

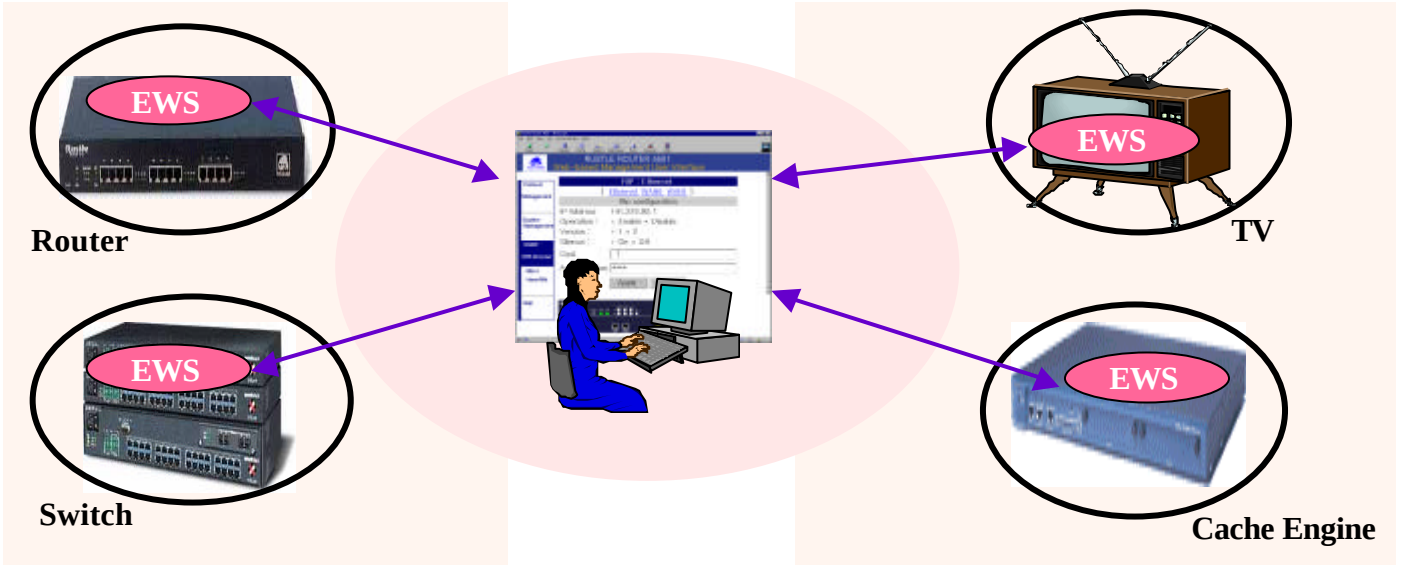
DP&NM Lab.
Dept. of Computer Science and Engineering
POSTECH, Pohang Korea
Tel: +82-562-279-5659
Email: mjchoi@postech.ac.kr

Contents

- Introduction
- Embedded Web Server
  - General Web Server & Embedded Web Server
  - Web-MUI & EWS Web-MUI
  - Advantages of EWS
- EWS Requirements
- Design
- Implementation
- POS-EWS Performance Evaluation
- POS-EWS Optimization
- Related Work
- Conclusions & Future work

Introduction

- Increase of the Network device
- Evolve World-Wide Web Web Technology
- Apply Web technology to network element management for monitoring and control
- The most direct way is to embed a Web server into network devices
- This can provide Web-based Management User Interface to manager



Introduction - cont'd

- Web-based management user interfaces through EWS have many advantages : Ubiquity, User-friendliness, Low-development cost, High maintenance
- Embedded Web Servers have consideration : Low resource usage (CPU usage, memory footprints)

? A lightweight and efficient EWS is essential for Web-based network element management.

Web Server & Embedded Web Server

- Web Server (World Wide Web Server)
  - Also known as HTTP Server
  - Designed to communicate with Web clients using HTTP
  - The repository of Web documents whose types are HTML and any application data with MIME type
  - Web documents are either static or generated dynamically
  - Typically runs on general purpose computers

- Embedded Web Server (EWS)
  - A Web server which runs on embedded systems with limited computing resources
  - Provides a Web browser interface between Web client and embedded system applications

WebMUI & EWS-WebMUI

- WebMUI (Web-based Management User Interface)
  - Provides a Web browser user interface for management
  - Provides static, dynamic and interactive content of management information of systems and networks
  - Can be used to configure, monitor and control managed systems via Web browser
- EWS-WebMUI (WebMUI through EWS)
  - EWS makes it possible for a Web browser to connect directly to the embedded system
  - Can be achieved by embedding three components into an embedded system: a Web server, Web documents and management applications

Advantages of EWS

- Provides enhanced user interface
  - Ubiquitous management
  - User-friendly interface via standard Web browsers
- Low development cost
  - No porting & distribution efforts for user applications
  - Platform independent graphical user interface
  - Short development time (short time-to-market)
- High maintenance
  - Web documents and associated programs can be easily modified

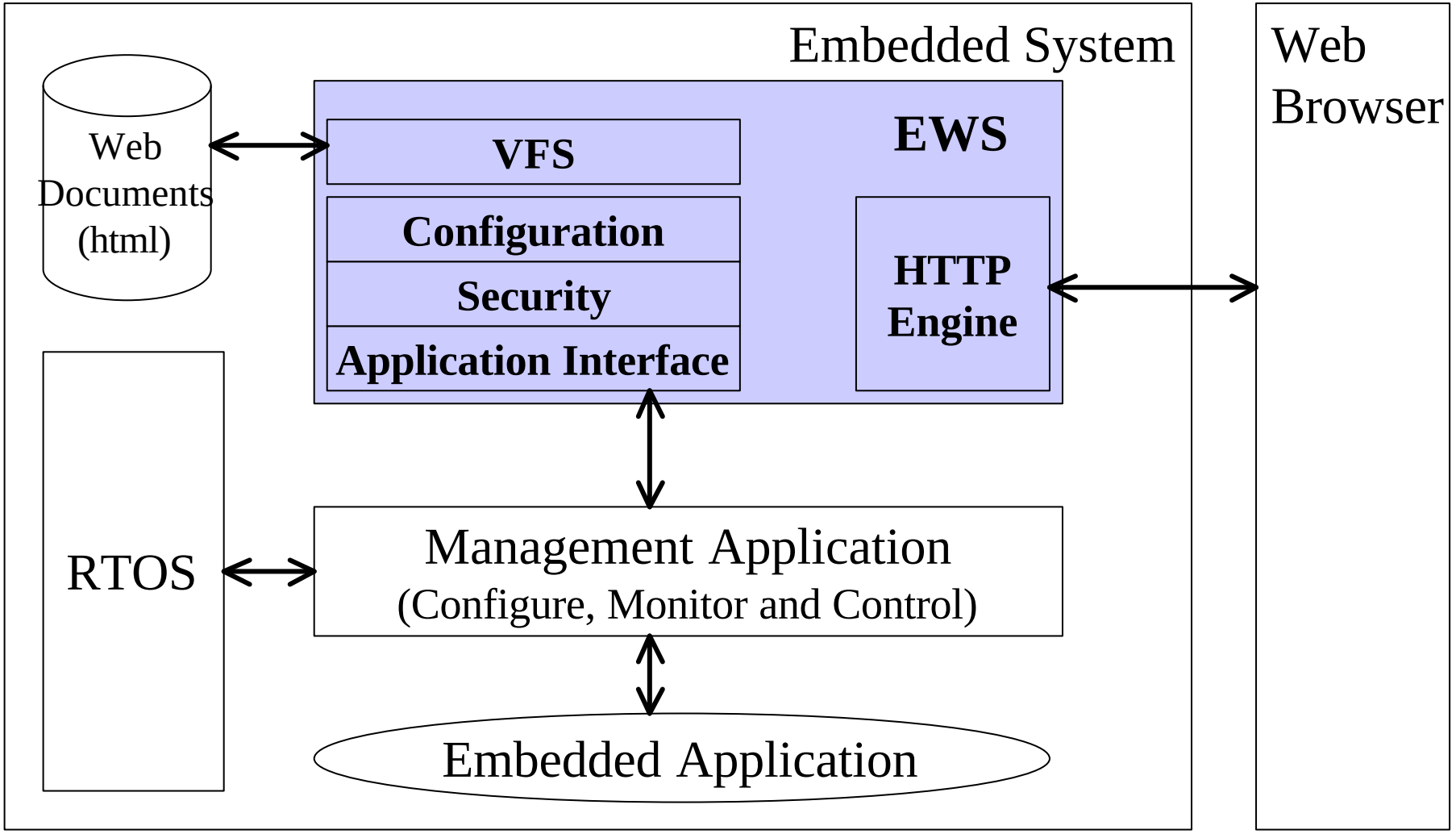
EWS Functional Requirements

- HTTP Engine
  - must deal with HTTP packets
- File System
  - Store Web documents and provide Web clients with them
- Security
  - limit access to sensitive information or configure & control
- Powerful application interface
  - mechanisms for the Web server to interact with embedded applications

EWS Non-Functional Requirements

- Low resource requirements
  - must use as little RAM, ROM and CPU as possible
- High reliability
  - highly reliable like one of the embedded system components
- High portability
  - portable on various RTOS and embedded systems

EWS Architecture



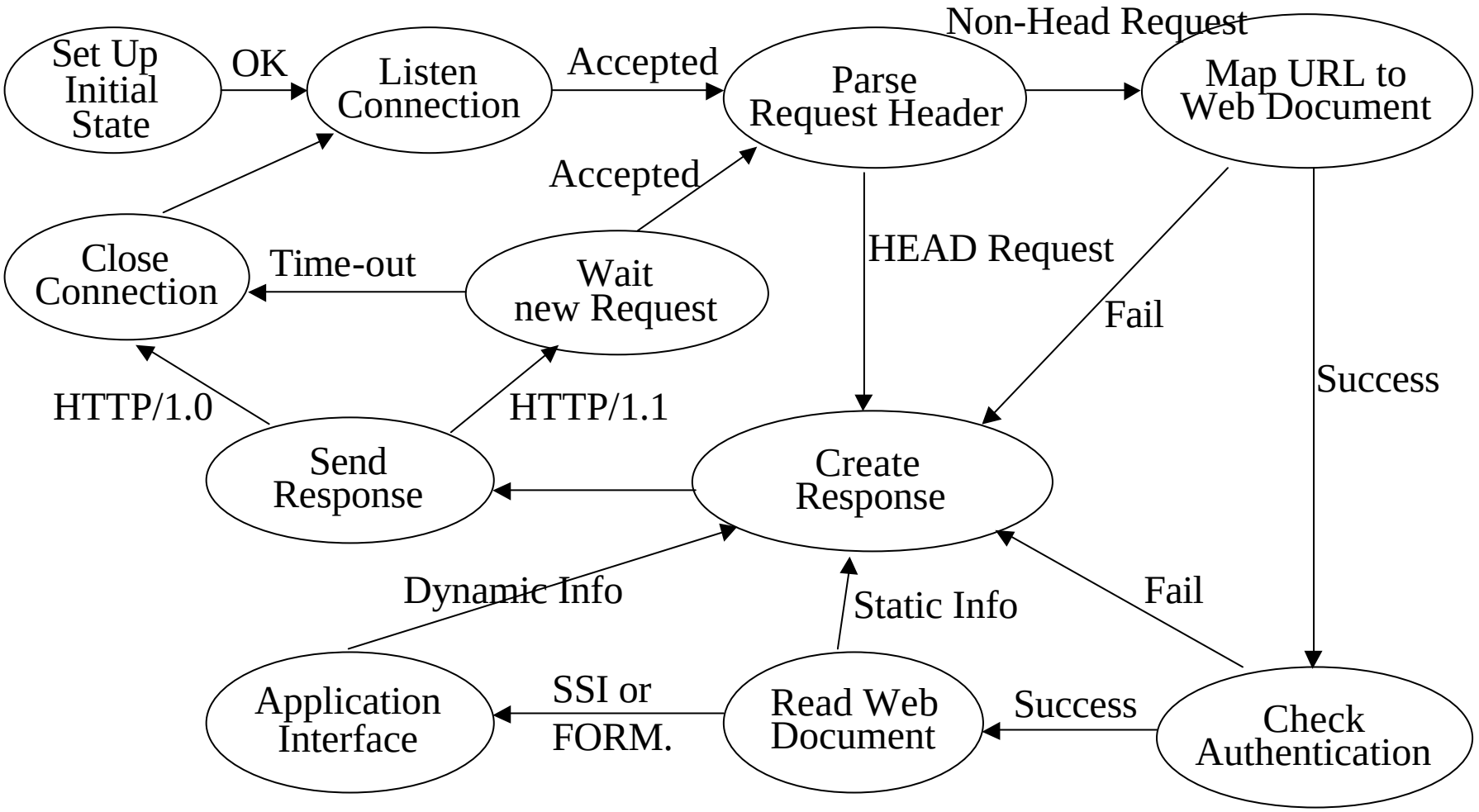
Design Issues

- Protocol Consideration
  - Explicit cache control
    - For static Web page, caching is desirable, eliminating requests for redundant information
    - Dynamically-generated Web documents must not be cached in order to retrieve up-to-date information
    - HTTP/1.1 allows server to control the Web cache
  - Persistent TCP connection
    - TCP implementations maintain connection state information for two minutes after connection closed
    - HTTP/1.1 allows for a single persistent TCP connection between the browser and server
- Application Interface
  - CGI (Common Gateway Interface)
  - SSI (Server Side Include)

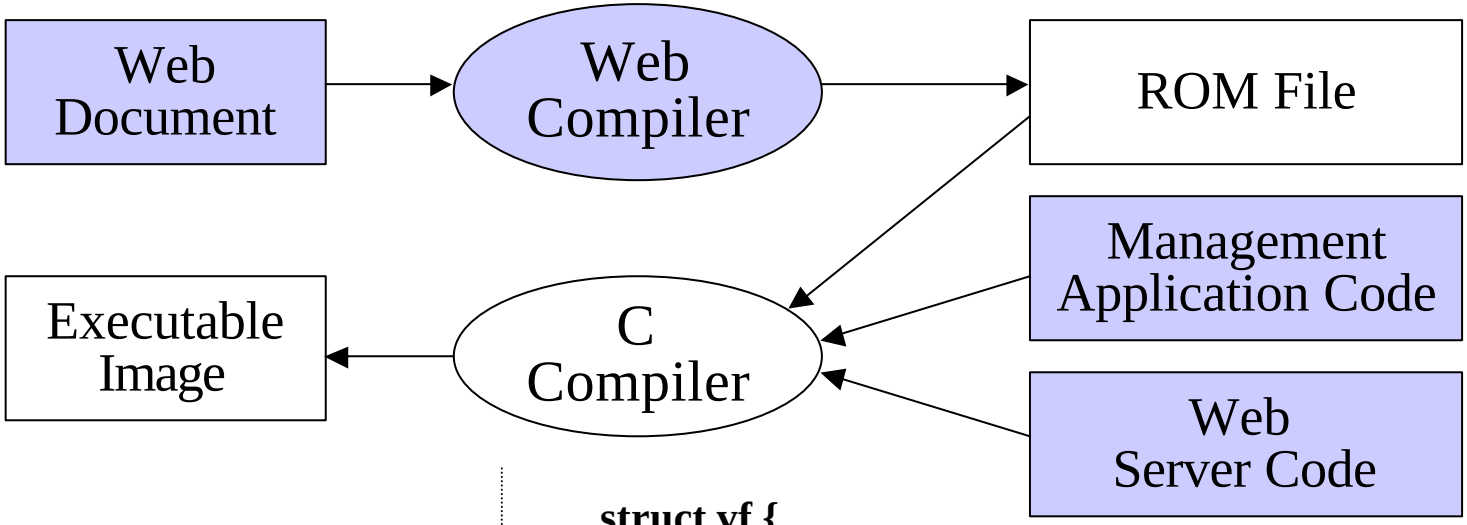
POS-EWS Features

- HTTP/1.1 compliant
  - Cache control
  - Persistent TCP connection
- Single thread based on extended architecture
  - Simple scheduler
  - Multiple finite state machines
- Virtual File System
  - Limited set of read-only file interface
  - Compression at compile time & Decompression at run time
- Web compiler
  - To build up virtual file system
  - Efficient server side include
- Cookie : state management
- OS : Xinu, pSOS, CPU : MPC 860

POS-EWS Finite State Machine



POS-EWS Web compiler

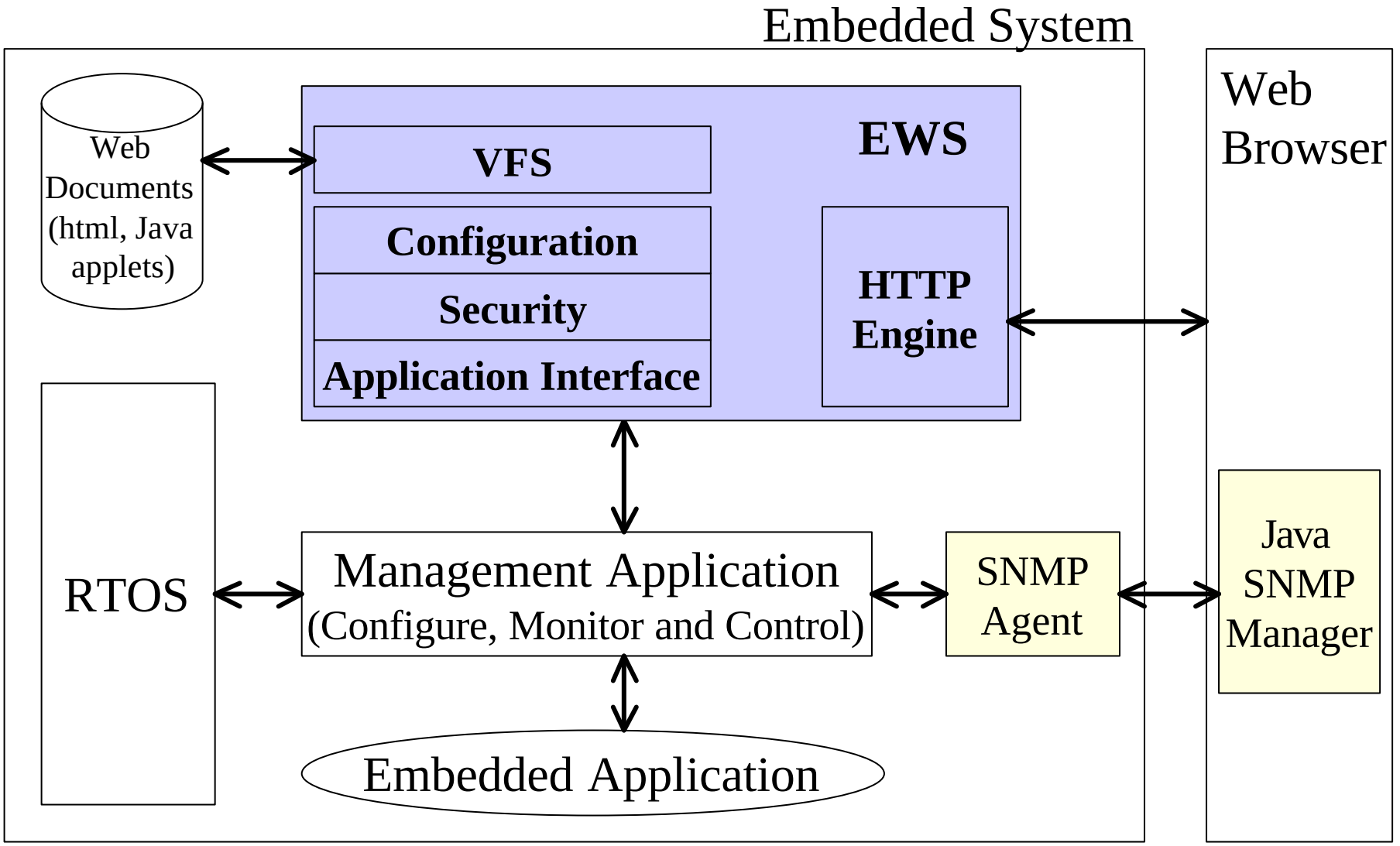


sysname.html

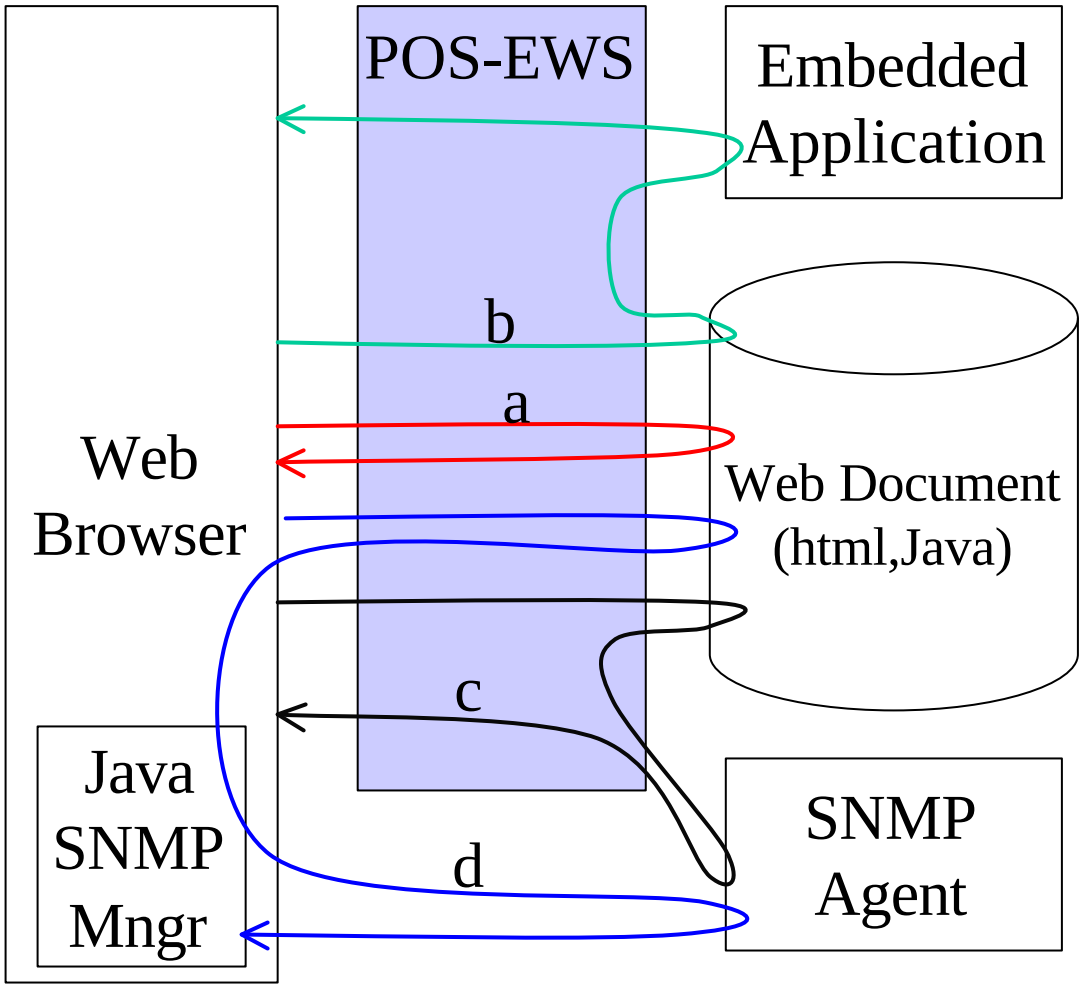
```
<HTML>
<HEAD>
<TITLE>Sample</TITLE>
</HEAD>
<BODY>
System Name: <% sysname %>
</BODY>
</HTML>
```

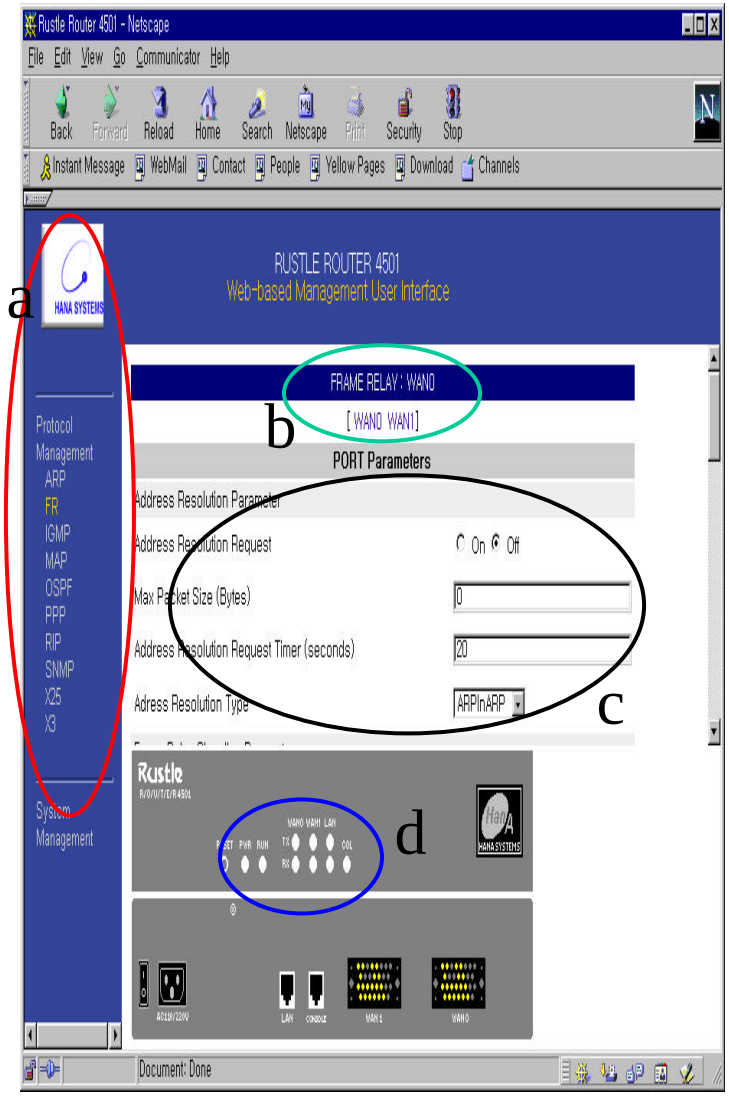
```
struct vf {
    char * data = sysname_html;
    char date[]="1999:06:12";
    int size = 78;
    struct sc_list *head = &sysname_html_sysname;
} sysname_html;
struct sc_list {
    int start = 44;
    int end = 55;
    (char *) (*fptr)() = sysname;
    struct sc_list *next = (struct sc_list) NULL;
}sysname_html_sysname;
char sysname_html = "<HTML> ……"
```

Extended Management Architecture



POS-EWS Application Example





POS-EWS Performance Evaluation

- Performance Metrics of General Web Server
  - Requests per second
  - Throughput in bytes per second
  - Round-trip time
  - Error rate
- Performance Metrics of EWS
  - Code size : 30 Kbytes
  - Run-time memory : 64Kbytes
  - CPU usage : Lowest priority
  - Maximum user connectivity (capacity)

POS-EWS Optimization

- Implement as a Finite State Machine (FSM) supporting single thread
- Keep TCP connection open and reuse by the Keep-Alive option
- Reduce the processing time using Web compiler through preprocessing
- Compress the documents for saving ROM resources
- Implement binary encoding of headers instead of ASCII

Related Work

Company & Product	OS supported	CPU supported	HTTP code size (version)	Features					
				SSI	VFS	Com-piler	Compression	Security (encode)	Coo-kie
Agranat Systems, EmWeb	No OS	Any CPU with a C compiler	25kbytes (1.1)	O	O	O	Proprietary	Basic + Digest	
AllegroSoft, RomPager	Any RTOS, No OS	Any processor with an ANSI-C Compiler	10-40 kbytes (1.1)	O	O	O		Basic	O
Spyglass, MicroServer	LynxOS, QNX,OS-9, pSOS, VxWorks	Any CPU with a C compiler	35-110 kbytes (1.1)	O			None	Baisc + Digest + SSL	
Magma, Lava	Any RTOS	Any CPU with a C compiler	15-40 kbytes (1.0)	O			Proprietary	Basic + SSL	
Quiotix, QEWS	pSOS, LynxOS, VxWorks	Any CPU with a C compiler	45-50 kbytes (1.0)	O	O		GZIP	Basic	
Web Device, Pico Server	LynxOS, Nucleus Plus,pSOS, VxWorks	Any CPU with a C compiler	15-30 kbytes (1.0)	O	O		ZIP-like	Basic + Digest + SSL	
POSTECH, POS-EWS	Real-time Xinu, pSOS	Any CPU with a C compiler	30kbytes (1.1)	O	O	O	GZIP/ CSS-Style	Basic + Digest (Base64)	O

Conclusions & Future work

- EWS-WebMUI provides ubiquitous, simple but powerful, user-friendly management user interface
- POS-EWS: HTTP/1.1 Compliant, Efficient embedded Web Server
- SNMP can be easily integrated with EWS
- Provide easy integration mechanisms to embedded management applications
- This "***webification***" of network devices has generated new philosophy for network element management

- Smaller and more powerful POS-EWS
- Port POS-EWS to other CPUs and embedded OSs
- POS-EWS on a chip
- Network Management of EWS-equipped Network Devices