

# **KLone**

An SDK for  
Embedded Web Applications

*Steven Dorigotti*  
KoanLogic (<http://koanlogic.com>)



August 11, 2006

## KLone

What is it?

Use Cases

Features

Libraries

Workings

## Development

Tutorial

Fine-Tuning

KLone-Devkit

## Applications

Industry

Open Source

## Wrap-Up



# In Brief

- ▶ lightweight web server

# In Brief

- ▶ lightweight web server
- ▶ C/C++ SDK

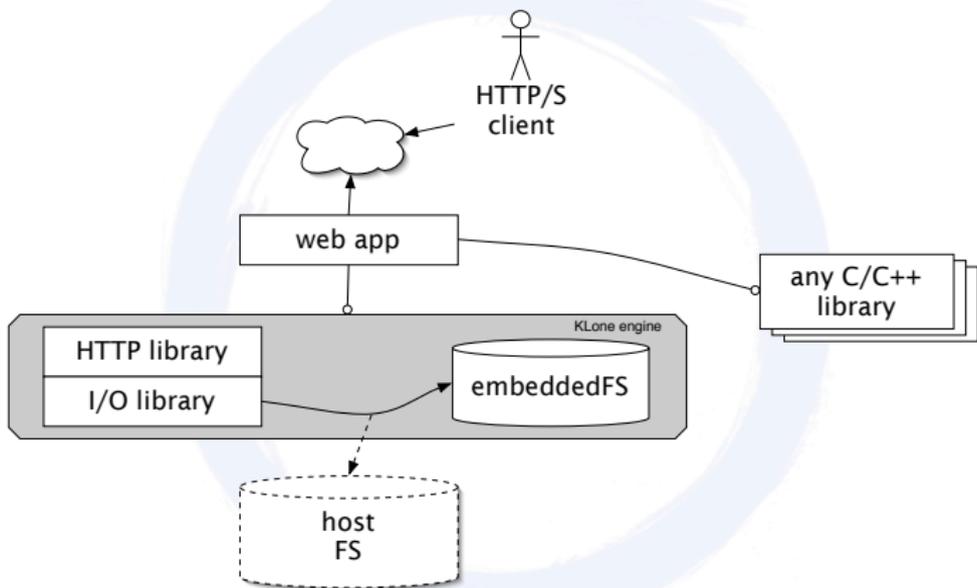
## In Brief

- ▶ lightweight web server
- ▶ C/C++ SDK
- ▶ API (GET/POST, sessions, cookies, I/O, logging, upload, ...)

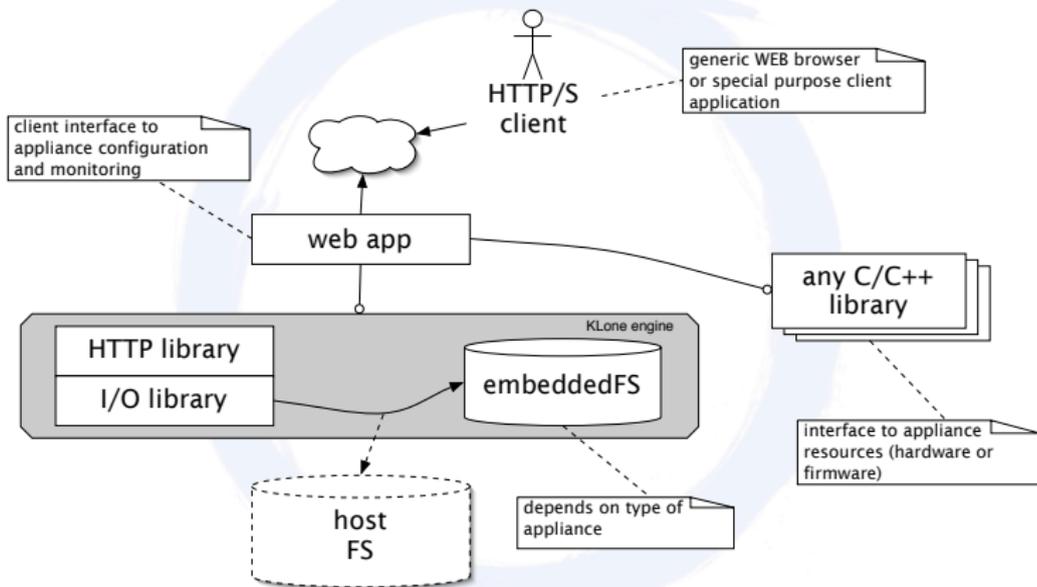
## In Brief

- ▶ lightweight web server
- ▶ C/C++ SDK
- ▶ API (GET/POST, sessions, cookies, I/O, logging, upload, ...)
- ▶ framework (compilation & building).

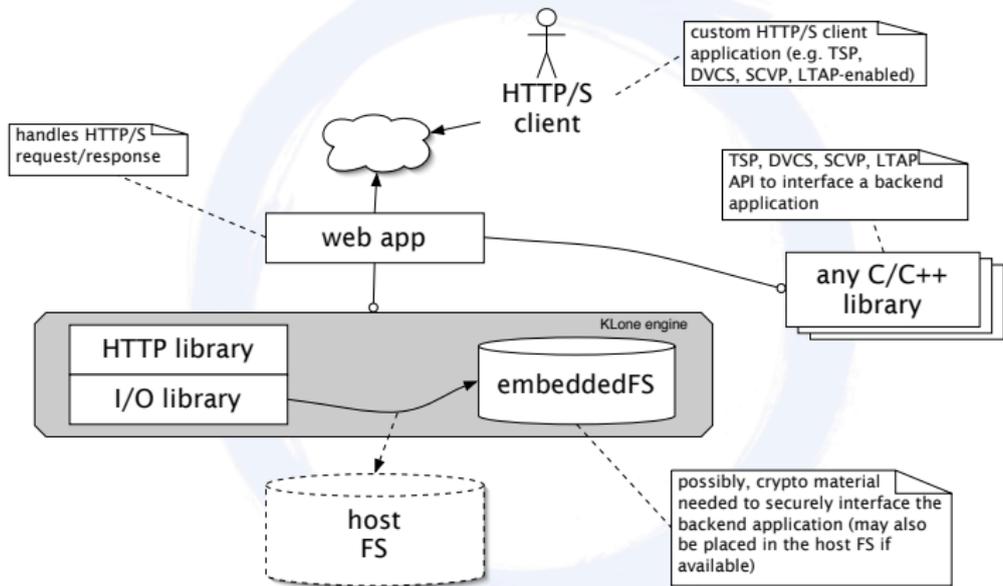
# Generic



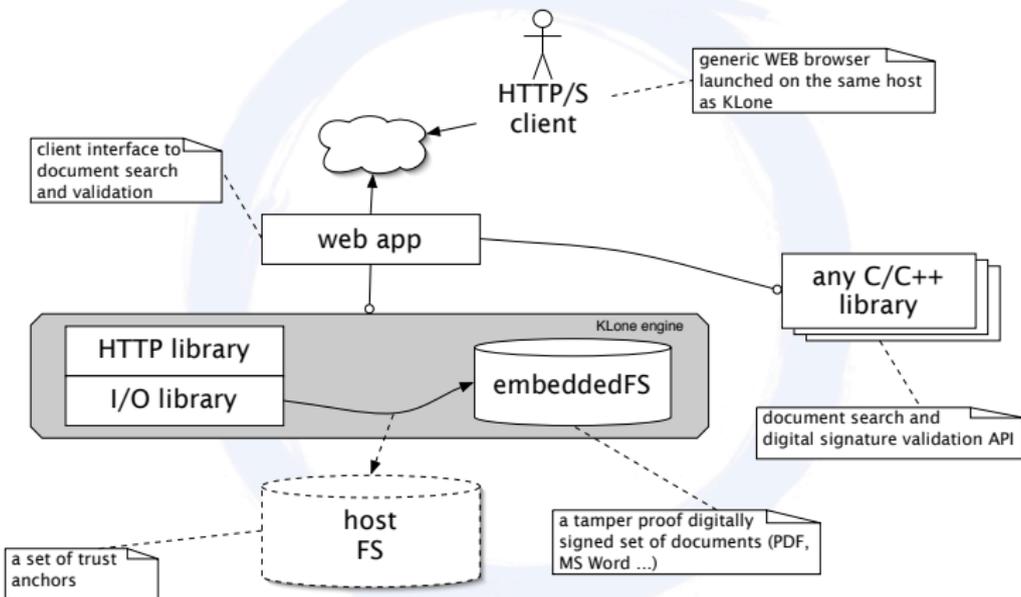
# Embedded System



# Remote Procedure Calls



# Content Management



# General Features

Some standard features:

- ▶ HTTP 1.0 compliant
- ▶ TLS/SSL
- ▶ virtual hosting

# General Features

Some standard features:

- ▶ HTTP 1.0 compliant
- ▶ TLS/SSL
- ▶ virtual hosting

...but also:

- ▶ optimised ( $\sim 140KB$  ROM,  $\sim 70KB$  RAM)

# General Features

Some standard features:

- ▶ HTTP 1.0 compliant
- ▶ TLS/SSL
- ▶ virtual hosting

...but also:

- ▶ optimised ( $\sim 140KB$  ROM,  $\sim 70KB$  RAM)
- ▶ native compiled code

# General Features

Some standard features:

- ▶ HTTP 1.0 compliant
- ▶ TLS/SSL
- ▶ virtual hosting

...but also:

- ▶ optimised ( $\sim 140KB$  ROM,  $\sim 70KB$  RAM)
- ▶ native compiled code
- ▶ ROMable

# General Features

Some standard features:

- ▶ HTTP 1.0 compliant
- ▶ TLS/SSL
- ▶ virtual hosting

...but also:

- ▶ optimised ( $\sim 140KB$  ROM,  $\sim 70KB$  RAM)
- ▶ native compiled code
- ▶ ROMable
- ▶ Open Source (GPL/commercial).

# Embedded File System

- ▶ static/dynamic pages in **kloned** object

# Embedded File System

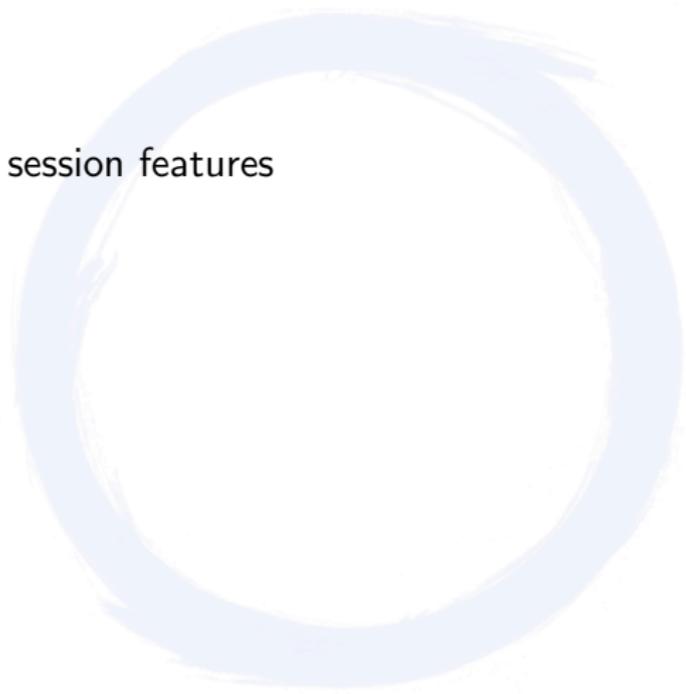
- ▶ static/dynamic pages in **kloned** object
- ▶ compression (zlib)

# Embedded File System

- ▶ static/dynamic pages in **kloned** object
- ▶ compression (zlib)
- ▶ encryption: 256-bit AES (OpenSSL).

# Client-Side Sessions

- ▶ standard session features



# Client-Side Sessions

- ▶ standard session features
- ▶ no storage required

# Client-Side Sessions

- ▶ standard session features
- ▶ no storage required
- ▶ cryptographically protected cookies.

# Client-Side Sessions

- ▶ standard session features
- ▶ no storage required
- ▶ cryptographically protected cookies.

The Internet Draft can be found at the following URL:

▶ <http://www.ietf.org/internet-drafts/draft-barbato-scs-00.txt>

# Portability

- ▶ GNU/Linux (2.2, 2.4, 2.6 - GCC 2.95.x, 3.x, both glibc and uClibc supported)
  - ▶ x86, MIPS, ARM, Cris (Axis), PowerPC, SH4, m68k
- ▶ QNX 6
  - ▶ x86, ARM, MIPS, PowerPC, SH
- ▶ VxWorks 6.x (DCC and GCC compilers)
  - ▶ All supported CPUs
- ▶ FreeBSD 4.X, 5.X and 6.X
- ▶ NetBSD 2.0.X, 2.1 and 3.X
- ▶ OpenBSD 3.8
- ▶ Darwin 7 / MacOSX 10.3
- ▶ Microsoft Windows XP (MinGW framework with MSYS).

# Libraries

3<sup>rd</sup> party dependencies are minimised: zlib (compression), OpenSSL (encryption).

# Libraries

3<sup>rd</sup> party dependencies are minimised: zlib (compression), OpenSSL (encryption).

Internally developed, BSD-licensed libraries:

# Libraries

3<sup>rd</sup> party dependencies are minimised: zlib (compression), OpenSSL (encryption).

Internally developed, BSD-licensed libraries:

- ▶ LibU: utility library
  - ▶ multi-platform
  - ▶ net, string, memory, config, hmap, ...

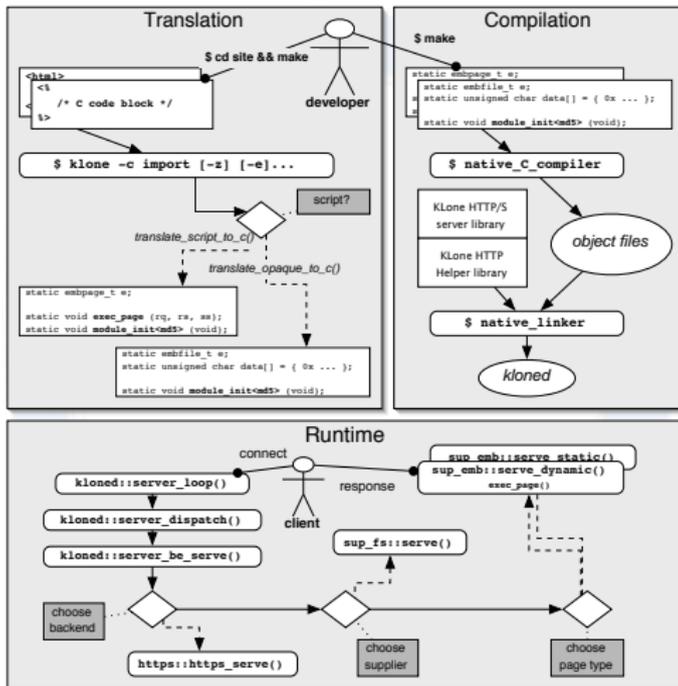
# Libraries

3<sup>rd</sup> party dependencies are minimised: zlib (compression), OpenSSL (encryption).

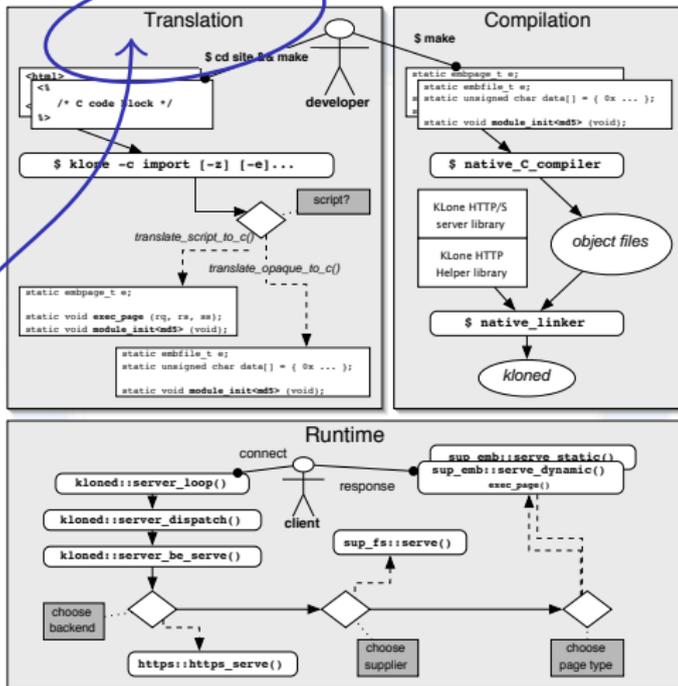
Internally developed, BSD-licensed libraries:

- ▶ LibU: utility library
  - ▶ multi-platform
  - ▶ net, string, memory, config, hmap, ...
- ▶ MaKL: build system
  - ▶ Autotools replacement
  - ▶ Bourne Shell & GNU Make
  - ▶ toolchains

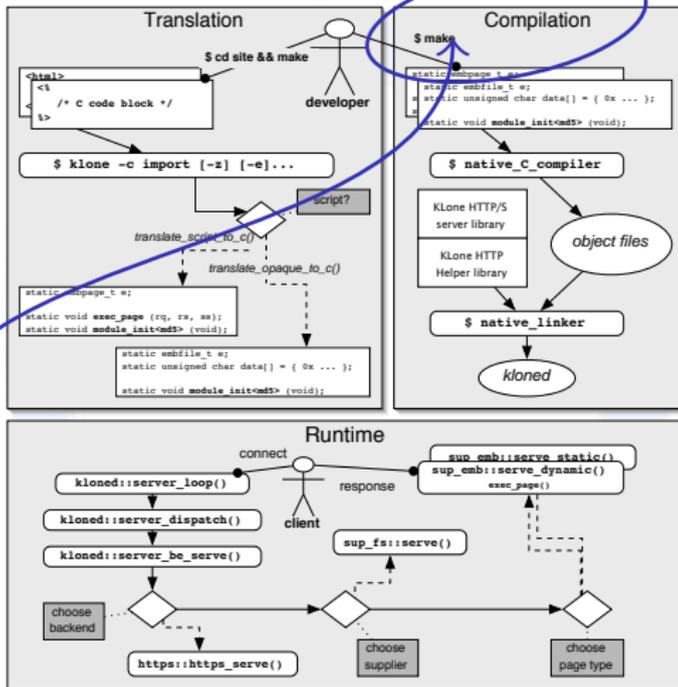
# Workings



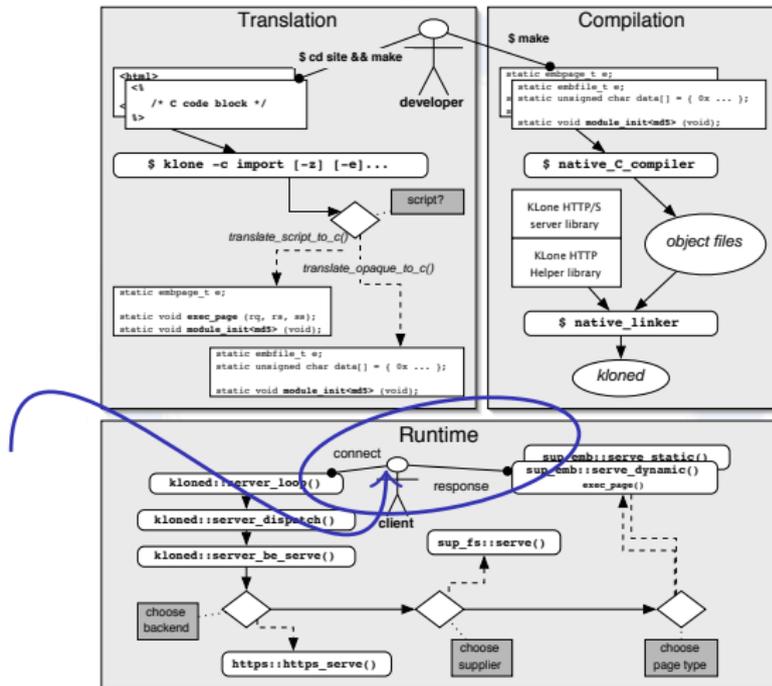
# Workings



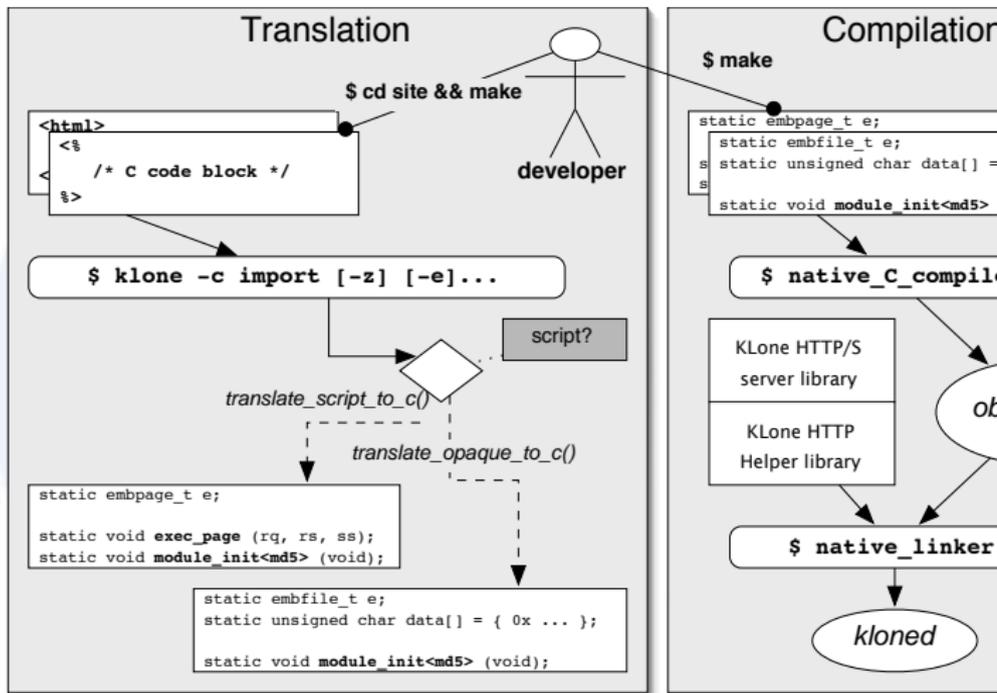
# Workings



# Workings

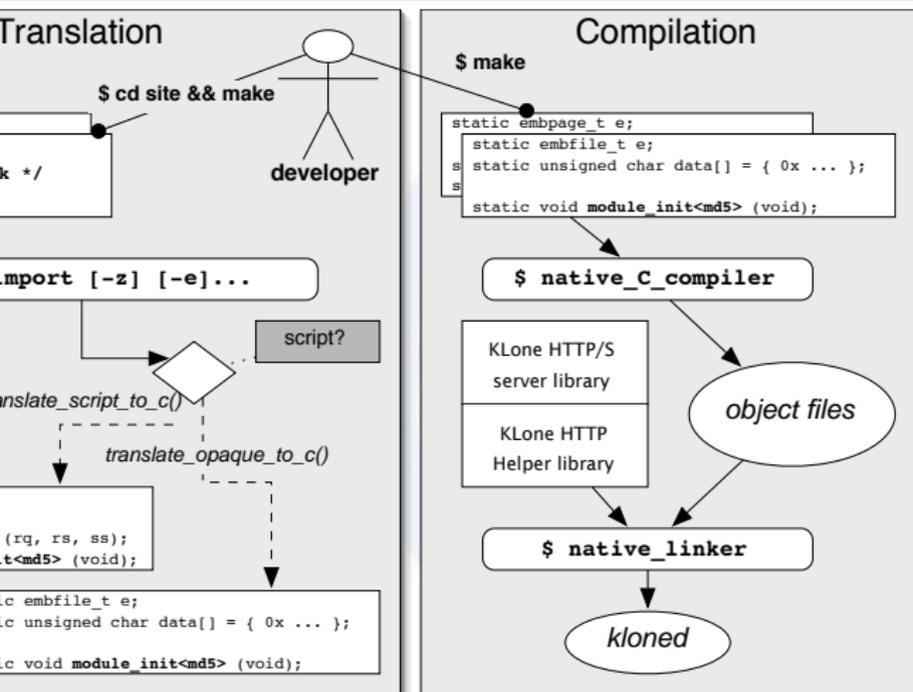


# Workings



## Runtime

# Workings



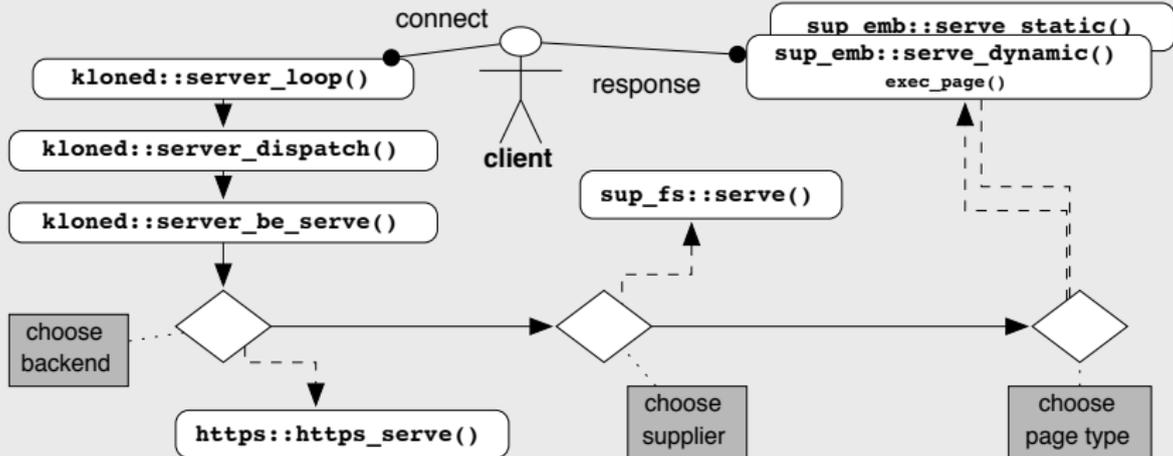
Runtime

# Workings

```
static void module_init<md5> (void);
```

*KLONE*

## Runtime



# Syntax

The syntax for dynamic **.kl1** pages is very intuitive:



# Syntax

The syntax for dynamic **.kl1** pages is very intuitive:

- ▶ standard HTML



# Syntax

The syntax for dynamic **.kl1** pages is very intuitive:

- ▶ standard HTML
- ▶ code blocks:

```
<% /* C/C++ code */ %>
```

# Syntax

The syntax for dynamic **.kl1** pages is very intuitive:

- ▶ standard HTML
- ▶ code blocks:

```
<% /* C/C++ code */ %>
```

- ▶ expression printing:

```
<%= /* char *expression */ %>
```

# Syntax

The syntax for dynamic **.kl1** pages is very intuitive:

- ▶ standard HTML
- ▶ code blocks:

```
<% /* C/C++ code */ %>
```

- ▶ expression printing:

```
<%= /* char *expression */ %>
```

- ▶ declarations:

```
<%! /* C/C++ declarations */ %>
```

# Syntax

The syntax for dynamic **.kl1** pages is very intuitive:

- ▶ standard HTML
- ▶ code blocks:

```
<% /* C/C++ code */ %>
```

- ▶ expression printing:

```
<%= /* char *expression */ %>
```

- ▶ declarations:

```
<%! /* C/C++ declarations */ %>
```

- ▶ preprocessor directives:

```
<%@ directive [param]... %>
```

## Example: Hello World

```
<!-- hello.kl1 -->
<%@ include "header.kl1" %>
<%! #include <time.h> %>
<%
    time_t now;
    int i;

    now = time(NULL);
    io_printf(out, "today is %s <p>", ctime(&now));

    for(i = 0; i < 10; ++i)
    {
        io_printf(out, "hello world! <p>");
    }
%>
<%@ include "footer.kl1" %>
```

## Example: Hello World

```
<!-- hello.kl1 -->
<%@ include "header.kl1" %>
<%! #include <time.h> %>
<%
    time_t now;
    int i;

    now = time(NULL);
    io_printf(out, "today is %s <p>", ctime(&now));

    for(i = 0; i < 10; ++i)
    {
        io_printf(out, "hello world! <p>");
    }
%>
<%@ include "footer.kl1" %>
```

## Example: Hello World

```
<!-- hello.kl1 -->
<%@ include "header.kl1" %>
<%! #include <time.h> %>
<%
    time_t now;
    int i;

    now = time(NULL);
    io_printf(out, "today is %s <p>", ctime(&now));

    for(i = 0; i < 10; ++i)
    {
        io_printf(out, "hello world! <p>");
    }

%>
<%@ include "footer.kl1" %>
```

## Example: Hello World

```
<!-- hello.kl1 -->
<%@ include "header.kl1" %>
<%! #include <time.h> %>
<%
    time_t now;
    int i;

    now = time(NULL);
    io_printf(out, "today is %s <p>", ctime(&now));

    for(i = 0; i < 10; ++i)
    {
        io_printf(out, "hello world! <p>");
    }
%>
<%@ include "footer.kl1" %>
```

## Example: Cookies

```
<!-- colour.kl1 -->
<%! #include <string.h> %>
<%
    vars_t *args = request_get_args(request);
    const char *colour = vars_get_value(args, "colour");

    if (colour && strlen(colour))
        response_set_cookie(response, "colour", colour, \
            time(NULL), NULL, NULL, 0);
%>
<p>What is your favourite colour?
<form action="colour.kl1" method="post">
colour: <input type="text" name="colour"><br>
<input type="submit" value="submit"><br>
</form>
<p>You preference: <%= request_get_cookie(request, "colour") %>
```

## Example: Cookies

```
<!-- colour.kl1 -->
<%! #include <string.h> %>
<%
    vars_t *args = request_get_args(request);
    const char *colour = vars_get_value(args, "colour");

    if (colour && strlen(colour))
        response_set_cookie(response, "colour", colour, \
            time(NULL), NULL, NULL, 0);
%>
<p>What is your favourite colour?
<form action="colour.kl1" method="post">
colour: <input type="text" name="colour"><br>
<input type="submit" value="submit"><br>
</form>
<p>You preference: <%= request_get_cookie(request, "colour") %>
```

## Example: Cookies

```
<!-- colour.kl1 -->
<%! #include <string.h> %>
<%
    vars_t *args = request_get_args(request);
    const char *colour = vars_get_value(args, "colour");

    if (colour && strlen(colour))
        response_set_cookie(response, "colour", colour, \
            time(NULL), NULL, NULL, 0);
%>
<p>What is your favourite colour?
<form action="colour.kl1" method="post">
colour: <input type="text" name="colour"><br>
<input type="submit" value="submit"><br>
</form>
<p>You preference: <%= request_get_cookie(request, "colour") %>
```

## Example: Cookies

```
<!-- colour.kl1 -->
<%! #include <string.h> %>
<%
    vars_t *args = request_get_args(request);
    const char *colour = vars_get_value(args, "colour");

    if (colour && strlen(colour))
        response_set_cookie(response, "colour", colour, \
            time(NULL), NULL, NULL, 0);
%>
<p>What is your favourite colour?
<form action="colour.kl1" method="post">
colour: <input type="text" name="colour"><br>
<input type="submit" value="submit"><br>
</form>
<p>You preference: <%= request_get_cookie(request, "colour") %>
```

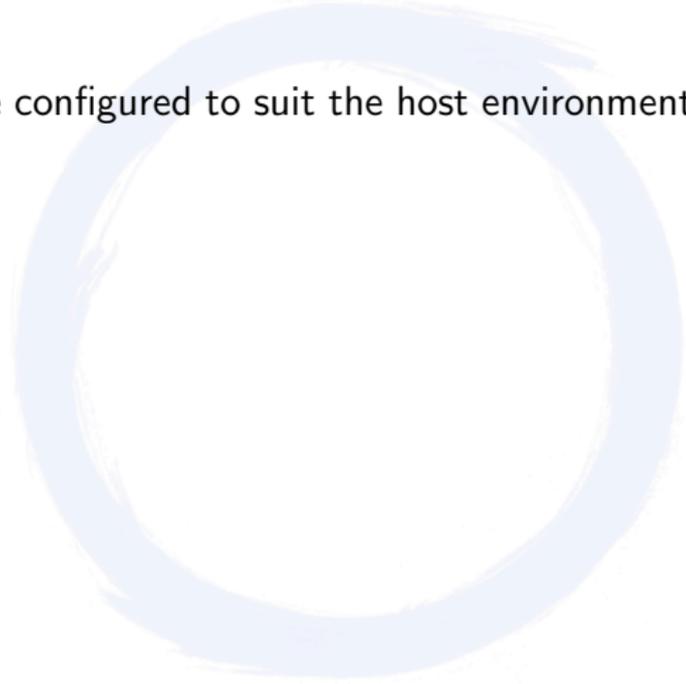
## Example: Cookies

```
<!-- colour.kl1 -->
<%! #include <string.h> %>
<%
    vars_t *args = request_get_args(request);
    const char *colour = vars_get_value(args, "colour");

    if (colour && strlen(colour))
        response_set_cookie(response, "colour", colour, \
            time(NULL), NULL, NULL, 0);
%>
<p>What is your favourite colour?
<form action="colour.kl1" method="post">
colour: <input type="text" name="colour"><br>
<input type="submit" value="submit"><br>
</form>
<p>You preference: <%= request_get_cookie(request, "colour") %>
```

# Fine-Tuning

KClone can be configured to suit the host environment in terms of:



# Fine-Tuning

KClone can be configured to suit the host environment in terms of:

- ▶ CPU, memory and filesystem utilisation:
  - ▶ server model
  - ▶ compression of content and/or sessions
  - ▶ sessions: file, memory or client
  - ▶ logging: file, syslog or memory

# Fine-Tuning

KClone can be configured to suit the host environment in terms of:

- ▶ CPU, memory and filesystem utilisation:
  - ▶ server model
  - ▶ compression of content and/or sessions
  - ▶ sessions: file, memory or client
  - ▶ logging: file, syslog or memory
- ▶ security requirements:
  - ▶ uid/guid
  - ▶ chroot
  - ▶ file and/or session encryption.

# Zero Storage / In-memory Sessions / HTTP

```
server_list my_http

log
{
    type          memory
    memory.limit  200
}

my_http
{
    type          http
    addr.type     IPv4
    addr.port     8080
    dir_root      /www

    session
    {
        type          memory
        compress      yes
        memory.limit  8192
        memory.max_count 2
    }
}
```

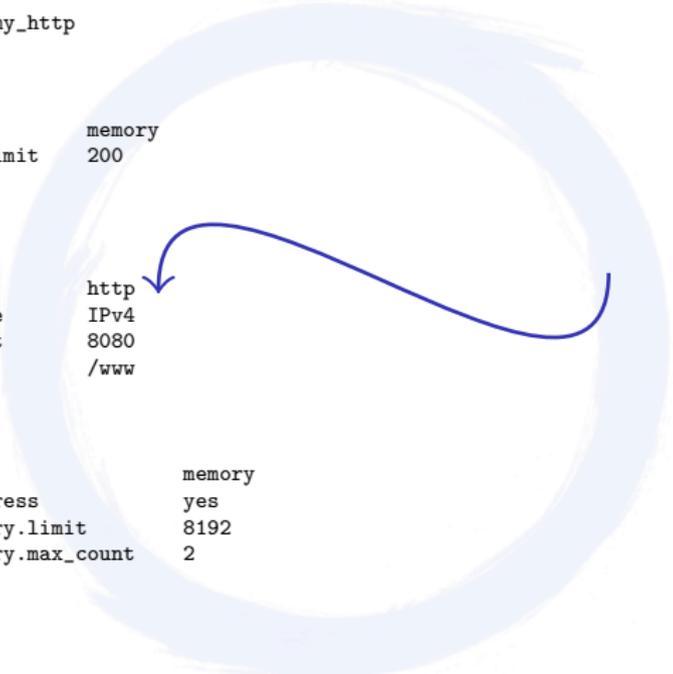
# Zero Storage / In-memory Sessions / HTTP

```
server_list my_http

log
{
    type          memory
    memory.limit  200
}

my_http
{
    type          http
    addr.type     IPv4
    addr.port     8080
    dir_root     /www

    session
    {
        type          memory
        compress      yes
        memory.limit  8192
        memory.max_count 2
    }
}
```



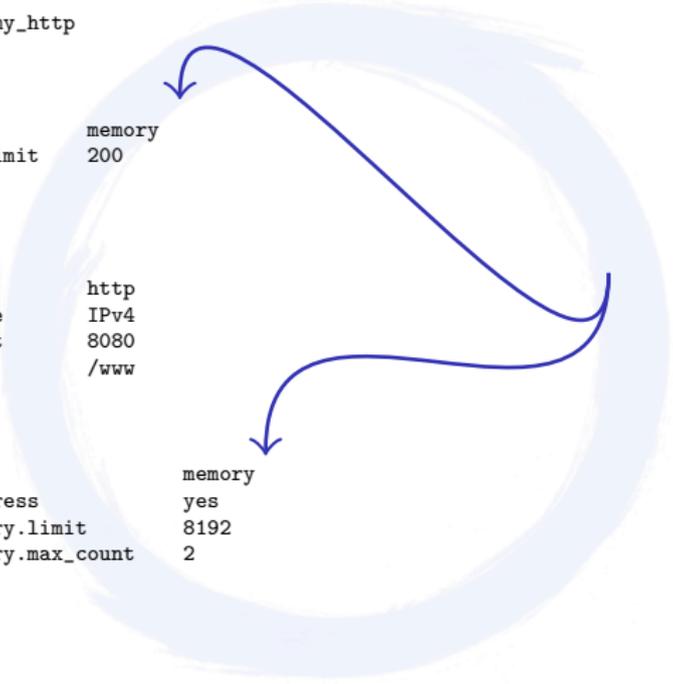
# Zero Storage / In-memory Sessions / HTTP

```
server_list my_http

log
{
    type          memory
    memory.limit  200
}

my_http
{
    type          http
    addr.type     IPv4
    addr.port     8080
    dir_root      /www

    session
    {
        type          memory
        compress      yes
        memory.limit  8192
        memory.max_count 2
    }
}
```



# Zero Storage / Client-side Sessions / HTTPS

```
server_list my_https

log
{
    type          memory
    memory.limit  200
}

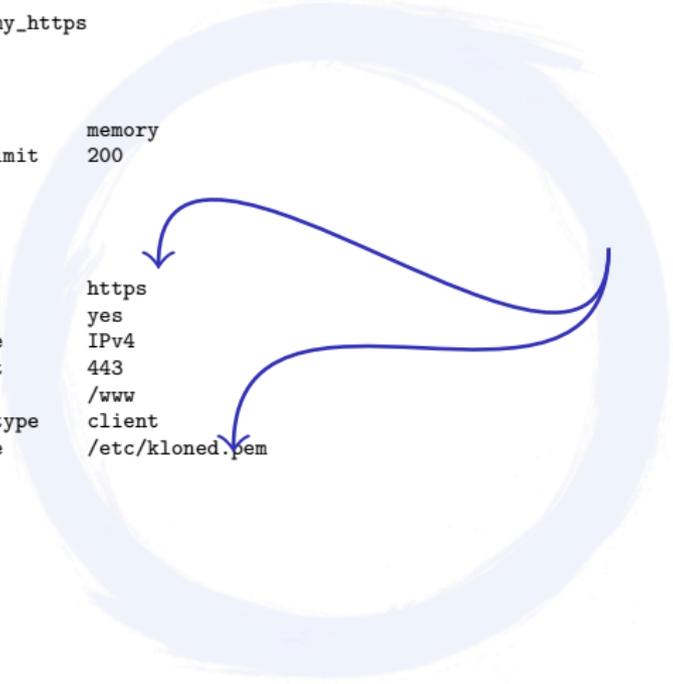
my_https
{
    type          https
    compress      yes
    addr.type     IPv4
    addr.port     443
    dir_root      /www
    session.type  client
    cert_file     /etc/kloned.pem
}
```

# Zero Storage / Client-side Sessions / HTTPS

```
server_list my_https

log
{
    type          memory
    memory.limit  200
}

my_https
{
    type          https
    compress     yes
    addr.type     IPv4
    addr.port     443
    dir_root      /www
    session.type  client
    cert_file     /etc/kloned.pem
}
}
```



# Zero Storage / Client-side Sessions / HTTPS

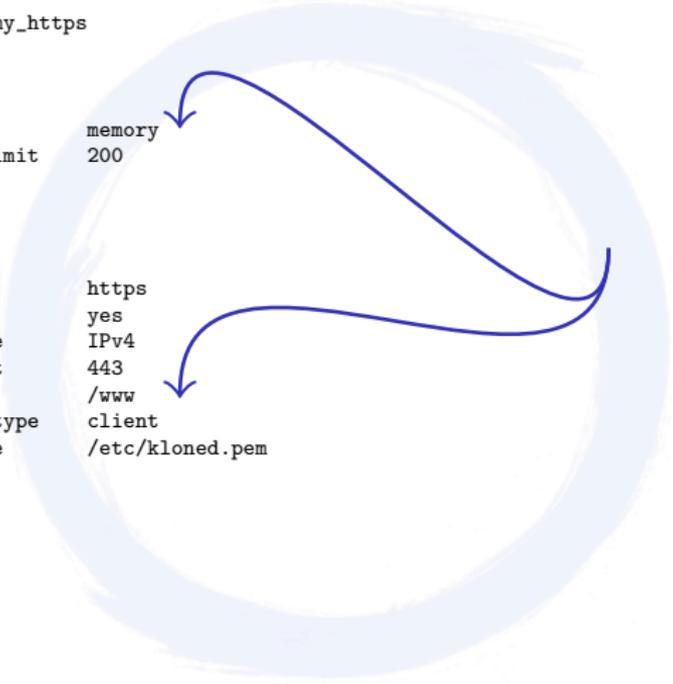
```

server_list my_https

log
{
  type          memory
  memory.limit  200
}

my_https
{
  type          https
  compress      yes
  addr.type     IPv4
  addr.port     443
  dir_root      /www
  session.type  client
  cert_file     /etc/kloned.pem
}

```





# KClone-Devkit

KClone-Devkit was developed with developers in mind:

- ▶ simplified learning curve

# KClone-Devkit

KClone-Devkit was developed with developers in mind:

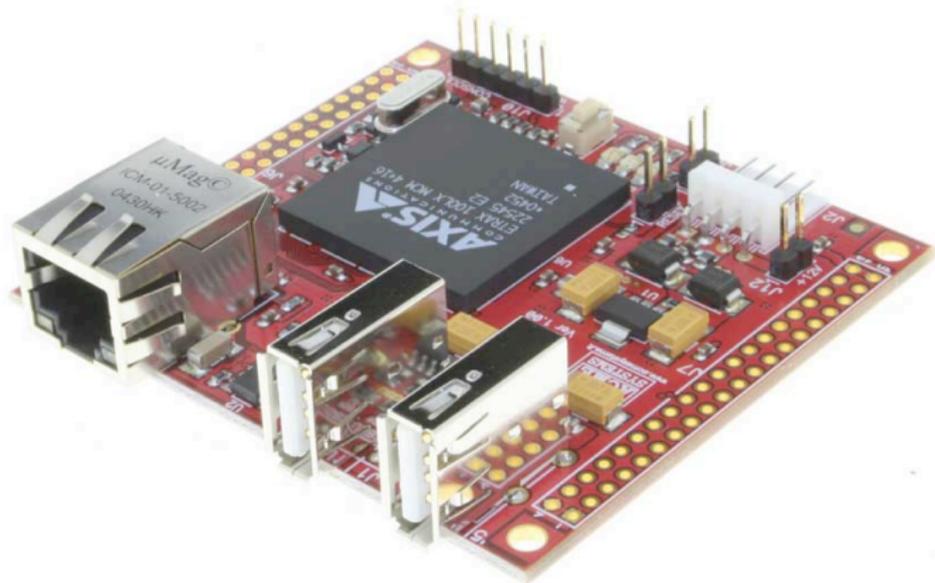
- ▶ simplified learning curve
- ▶ build environment

# KClone-Devkit

KClone-Devkit was developed with developers in mind:

- ▶ simplified learning curve
- ▶ build environment
- ▶ provides working examples:
  - ▶ Fox-Demo (basic and VHDL)
  - ▶ PXAregs-GUI
  - ▶ TSP-Proxy
  - ▶ Doc-Search.

# Kclone-Devkit: FOX-Demo



# KClone-Devkit: FOX-Demo

```
<html>
  <body>
    <%! #include <foxbone_userspace.h> %>
    <%
      unsigned int release, fd;
      fd = open("/dev/foxbone", O_RDONLY);
      release = ioctl(fd, IOCTL_FOXBONE_READ, \
        FOXBONE_VER_REG);
      io_printf(out, "%x", release);
      close(fd);
    %>
  </body>
</html>
```

# KClone-Devkit: FOX-Demo

```
<html>
  <body>
    <%! #include <foxbone_userspace.h> %>
    <%
      unsigned int release, fd;
      fd = open("/dev/foxbone", O_RDONLY);
      release = ioctl(fd, IOCTL_FOXBONE_READ, \
        FOXBONE_VER_REG);
      io_printf(out, "%x", release);
      close(fd);
    %>
  </body>
</html>
```

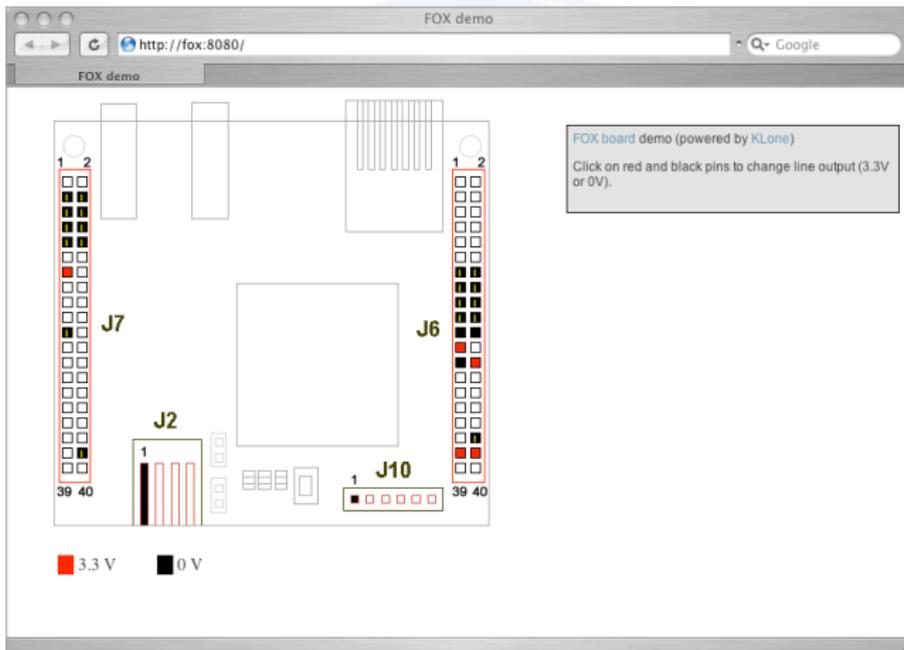
# KClone-Devkit: FOX-Demo

```
<html>
  <body>
    <%! #include <foxbone_userspace.h> %>
    <%
      unsigned int release, fd;
      fd = open("/dev/foxbone", O_RDONLY);
      release = ioctl(fd, IOCTL_FOXBONE_READ, \
        FOXBONE_VER_REG);
      io_printf(out, "%x", release);
      close(fd);
    %>
  </body>
</html>
```

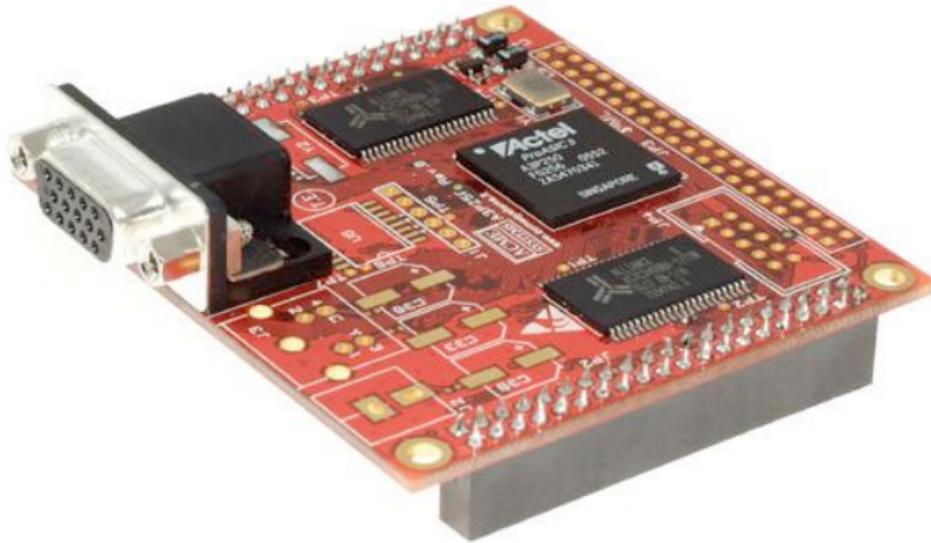
# KClone-Devkit: FOX-Demo

```
<html>
  <body>
    <%! #include <foxbone_userspace.h> %>
    <%
      unsigned int release, fd;
      fd = open("/dev/foxbone", O_RDONLY);
      release = ioctl(fd, IOCTL_FOXBONE_READ, \
        FOXBONE_VER_REG);
      io_printf(out, "%x", release);
      close(fd);
    %>
  </body>
</html>
```

# KLone-Devkit: FOX-Demo



# KClone-Devkit: FOX-VHDL



# KLone-Devkit: PXAregs-GUI



# KClone-Devkit: PXAregs-GUI

**PXAREGS GUI**

Please choose the register you want to read/modify from the list or type the register name (or part of it) in the blank field.

Click on register values to modify them.

Filter:  
 P

- PMCR (Power Manager Control Register (3-23))
- PSSR (Power Manager Sleep Status Register (3-29))
- PSPR (Power Manager Scratch Pad Register (3-30))
- PWER (Power Manager Wake-Up Enable Register (3-25))
- PRER (Power Manager Rising Edge Detect Enable Register (3-26))**
- PFER (Power Manager Falling Detect Enable Register (3-27))
- PEDR (Power Manager Edge Detect Status Register (3-28))
- PCFR (Power Manager General Configuration Register (3-24))
- PCSR0 (Power Manager GPIO Sleep State Register 0 (3-32))
- PCSR1 (Power Manager GPIO Sleep State Register 1 (3-32))
- PCSR2 (Power Manager GPIO Sleep State Register 2 (3-33))
- PMFW (Power Manager Fast Sleep Wakeup Register (3-31))
- GPLR0 (GPIO Pin Level Register 0 (4-7))
- GPLR1 (GPIO Level Register 1 (4-8))

Powered by [KClone](#).

PXA code by H. Schurig.

PRER	0x00000000	Power Manager Rising Edge Detect Enable Register (3-26)
PRER_RE0	0	PM wake up due to GPIO 0 rising edge detect enabled
PRER_RE1	0	PM wake up due to GPIO 1 rising edge detect enabled
PRER_RE2	0	PM wake up due to GPIO 2 rising edge detect enabled
PRER_RE3	0	PM wake up due to GPIO 3 rising edge detect enabled
PRER_RE4	0	PM wake up due to GPIO 4 rising edge detect enabled
PRER_RE5	0	PM wake up due to GPIO 5 rising edge detect enabled
PRER_RE6	0	PM wake up due to GPIO 6 rising edge detect enabled
PRER_RE7	0	PM wake up due to GPIO 7 rising edge detect enabled
PRER_RE8	0	PM wake up due to GPIO 8 rising edge detect enabled
PRER_RE9	0	PM wake up due to GPIO 9 rising edge detect enabled
PRER_RE10	0	PM wake up due to GPIO10 rising edge detect enabled
PRER_RE11	0	PM wake up due to GPIO11 rising edge detect enabled
PRER_RE12	0	PM wake up due to GPIO12 rising edge detect enabled
PRER_RE13	0	PM wake up due to GPIO13 rising edge detect enabled
PRER_RE14	0	PM wake up due to GPIO14 rising edge detect enabled
PRER_RE15	0	PM wake up due to GPIO15 rising edge detect enabled

# KClone-Devkit: TSP-Proxy

TSP (RFC3161) Gateway Test Page

http://dual:8080/

## TSP (RFC3161) GATEWAY TEST PAGE

This is a test page for KClone use case [RPC](#), showing how you can build an HTTP interface to a [different protocol](#) matching the request-response paradigm.

### INSTRUCTIONS

- Upload a file to be timestamped (max accepted size is 0.5MB):  
 dylan-thomas
- choose your preferred backend TSA in list:  

IAIK
PolITo
Actalis
nCIPHER
ANDXOR
- go for it:

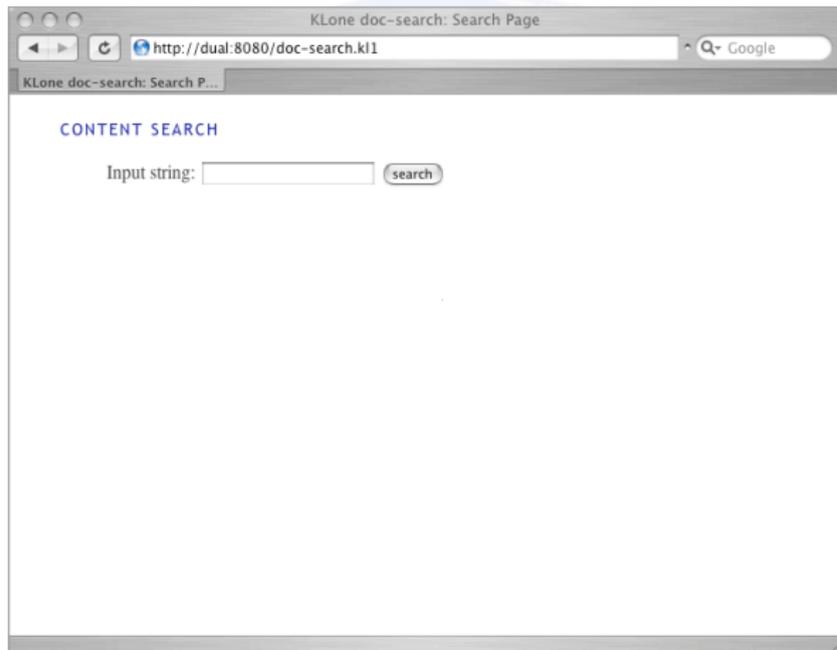
You'll get back an HTTP response with either Content-type application/timestamp-response together with the ASN.1 DER-encoded Time-Stamp Response message, or a suitable HTTP error with the reason why your request has been rejected.

Downloads

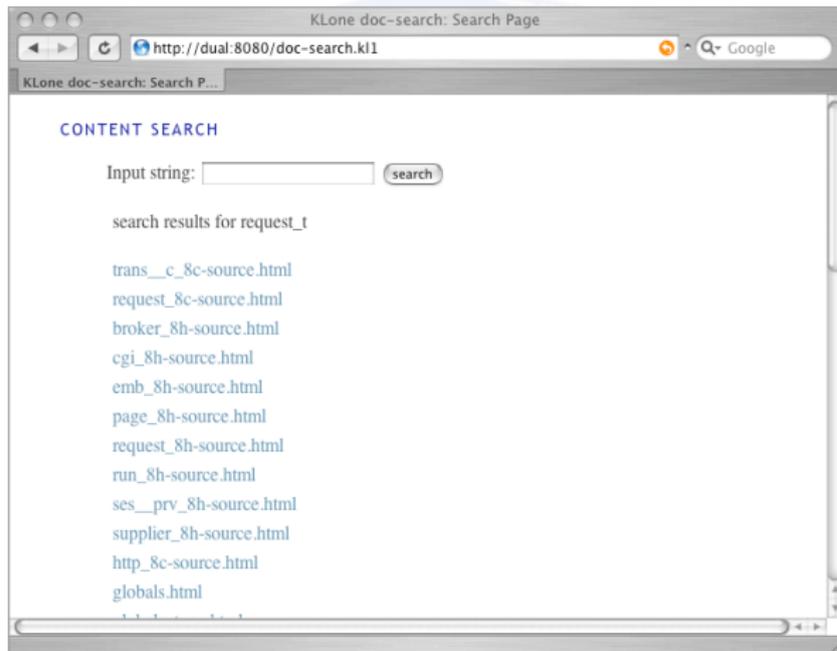
tsp\_proxy-1.kl1  
2.3 KB

Clear 1 Download

# KLone-Devkit: Doc-Search



# KClone-Devkit: Doc-Search



# KLone-Devkit: Doc-Search

KLone: entry.c Source File

http://dual:8080/entry\_8c-source.html

KLone: entry.c Source File

Main Page | Modules | Data Structures | Directories | File List | Data Fields | Globals

src / kloned

## ENTRY.C

Go to the documentation of this file.

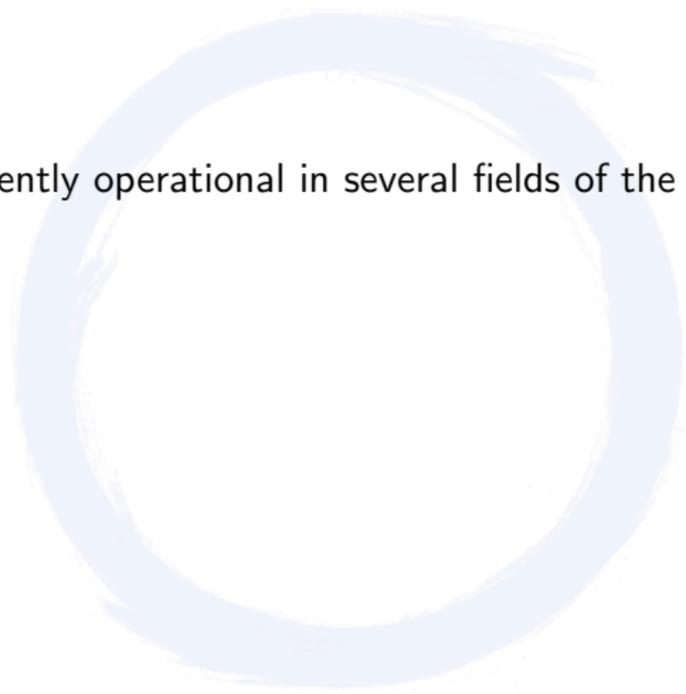
```

00001 /*
00002  * Copyright (c) 2005, 2006 by KoanLogic s.r.l. <http://www.koanlogic.com>
00003  * All rights reserved.
00004  *
00005  * This file is part of KLone, and as such it is subject to the license stated
00006  * in the LICENSE file which you have received as part of this distribution.
00007  *
00008  * $Id: entry.c,v 1.17 2006/01/09 12:38:38 tat Exp $
00009  */
00010
00011 #include "klone_conf.h"
00012 #include <stdlib.h>
00013 #include <unistd.h>
00014 #include <stdio.h>
00015 #include <u/libu.h>
00016 #include <klone/klone.h>
00017 #include <klone/server.h>
00018 #include <klone/os.h>
00019 #include <klone/context.h>
00020 #include <klone/utills.h>
00021 #include <klone/version.h>
00022 #include "main.h"
00023
00024 int facility = LOG_LOCAL0;
00025
00026 static context_t c;
00027 context_t *ctx = &c; /* exported */
00028

```

# Industry

KLone is currently operational in several fields of the IT industry:



# Industry

KLone is currently operational in several fields of the IT industry:

- ▶ FIAT: industrial control system

# Industry

KLone is currently operational in several fields of the IT industry:

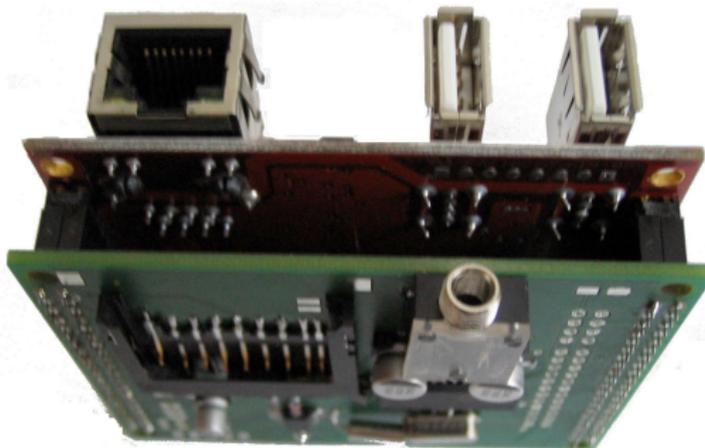
- ▶ FIAT: industrial control system
- ▶ RAI: network data analyser

# Industry

KLone is currently operational in several fields of the IT industry:

- ▶ FIAT: industrial control system
- ▶ RAI: network data analyser
- ▶ T-Services:
  - ▶ authentication system
  - ▶ validation authority.

# Open Source: Alp-MP3



# Open Source: Alp-MP3



## ALP-MP3 Fox Edition

### MP3-Player

Control  
Internet Radio  
Playlists

### Settings

Options  
Clock  
Wake Up  
Restart / Reboot

### Hardware Settings

Network  
Storage Media

Project Homepage

Update

SourceForge

## Player Control

Status : Playing Stream  
#DNBRADIO  
(http://85.12.11.155:8010/)

Volume : 20/30



loadtime : 0.100 sec

# Future Development

A preview:



# Future Development

A preview:

- ▶ HTTP 1.1 compliance



# Future Development

A preview:

- ▶ HTTP 1.1 compliance
- ▶ AJAX helpers

# Future Development

A preview:

- ▶ HTTP 1.1 compliance
- ▶ AJAX helpers
- ▶ performance & file size optimisation

# Future Development

A preview:

- ▶ HTTP 1.1 compliance
- ▶ AJAX helpers
- ▶ performance & file size optimisation
- ▶ more porting.

# Bookmarks

For more information:



# Bookmarks

For more information:

- ▶ Info, documentation and source code: <http://koanlogic.com>

# Bookmarks

For more information:

- ▶ Info, documentation and source code: <http://koanlogic.com>
- ▶ Mailing lists:
  - ▶ [klone-users@koanlogic.com](mailto:klone-users@koanlogic.com)
  - ▶ [klone-dev@koanlogic.com](mailto:klone-dev@koanlogic.com)

# Bookmarks

For more information:

- ▶ Info, documentation and source code: <http://koanlogic.com>
- ▶ Mailing lists:
  - ▶ [klone-users@koanlogic.com](mailto:klone-users@koanlogic.com)
  - ▶ [klone-dev@koanlogic.com](mailto:klone-dev@koanlogic.com)
- ▶ Other information: [info@koanlogic.com](mailto:info@koanlogic.com)

## Special Thanks

- ▶ Stefano Barbato
- ▶ Thomas Fossati

▶ <http://acmesystems.it>



▶ <http://phrozen.biz>

**PHROZEN**.biz  
components and embedded systems

# Questions

