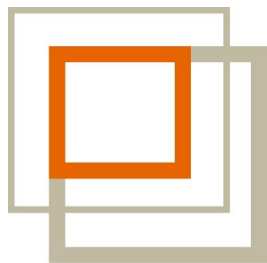




# Creating secure web based user interfaces for Embedded Devices

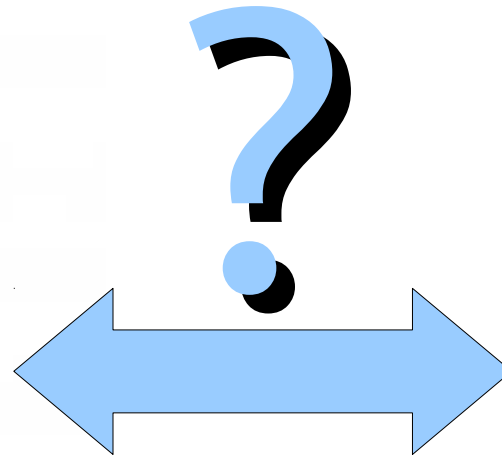


**ESSENSIUM**

© 2011 Essensium N.V.  
This work is licensed under a  
Creative Commons Attribution-ShareAlike 3.0  
Unported License



# How do you talk to an embedded system?



# Web interface is the easiest!

- Custom display
- RS232 – Shell
- Ssh
- Custom protocol
- **HTTP**





**mind**

embedded development

an ESSESIUM division

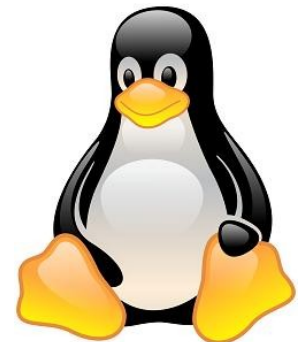


# Creating secure web based user interfaces for Embedded Devices



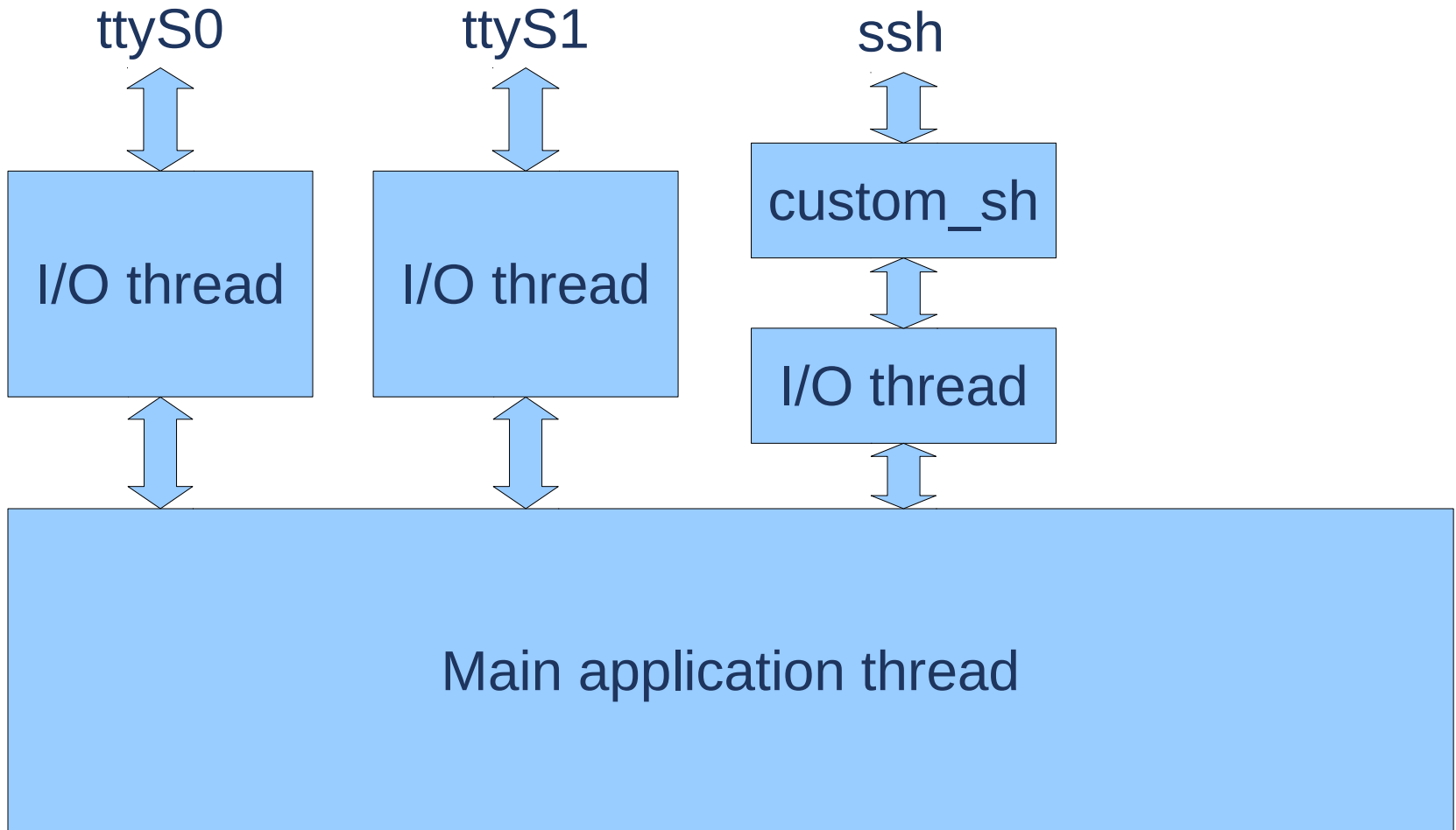
**ESSESIUM**

Arnout Vandecappelle  
arnout@mind.be



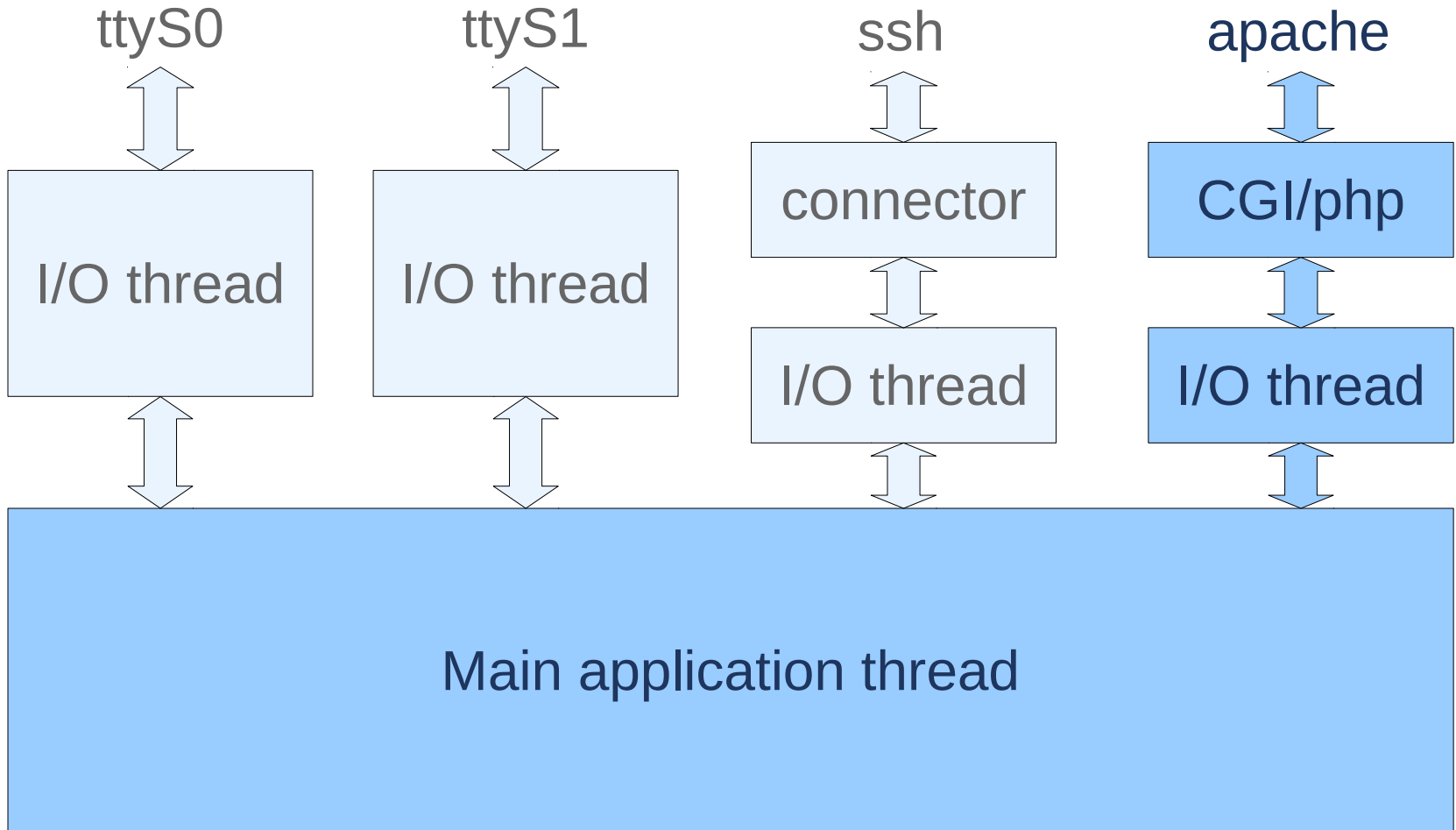
- 1 Why web-based user interfaces?
- 2 Adding an HTTP server to firmware
- 3 Security considerations
- 4 Using a browser as the UI

# Traditional firmware architecture





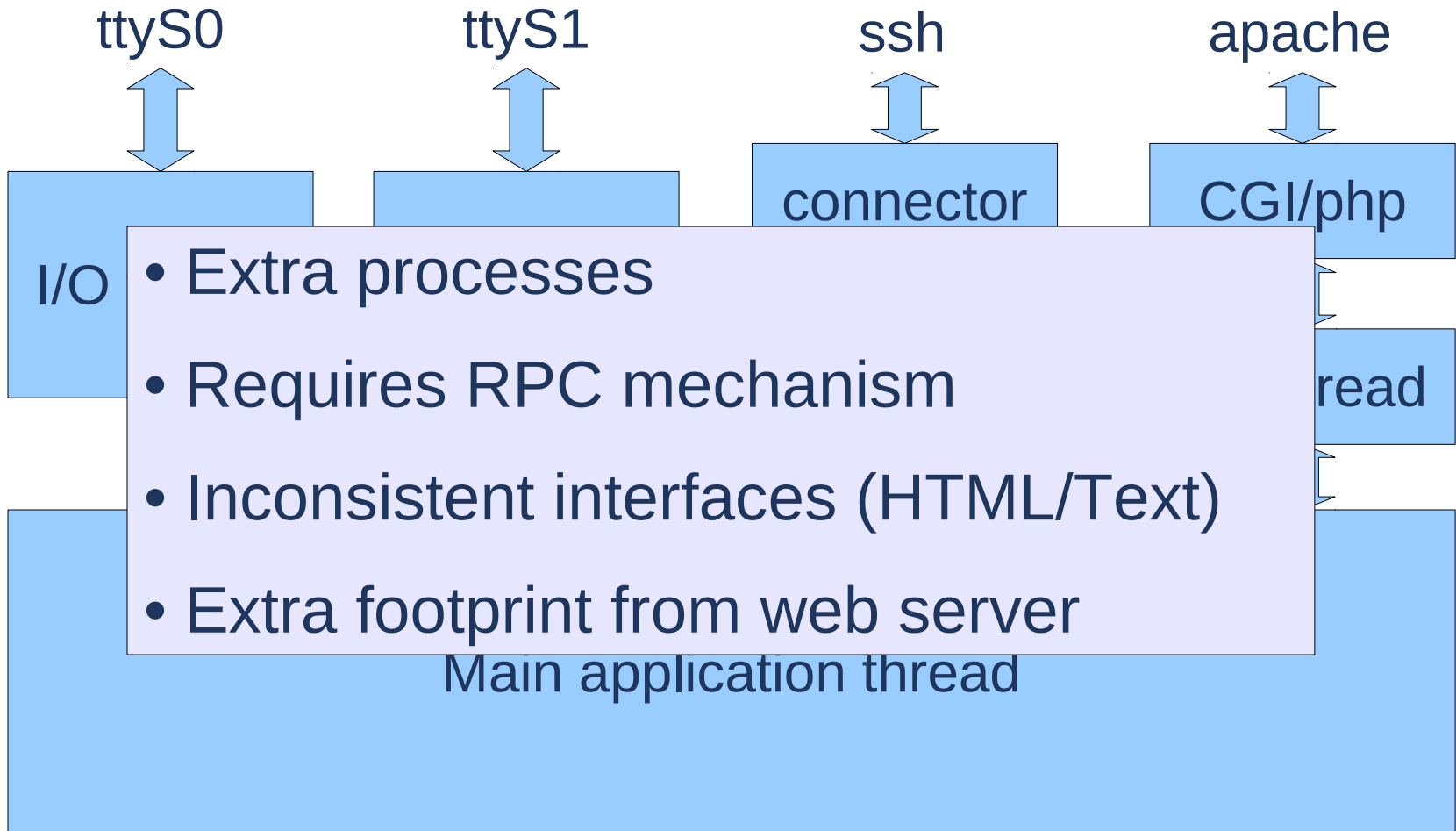
# Adding a web interface to traditional firmware architecture



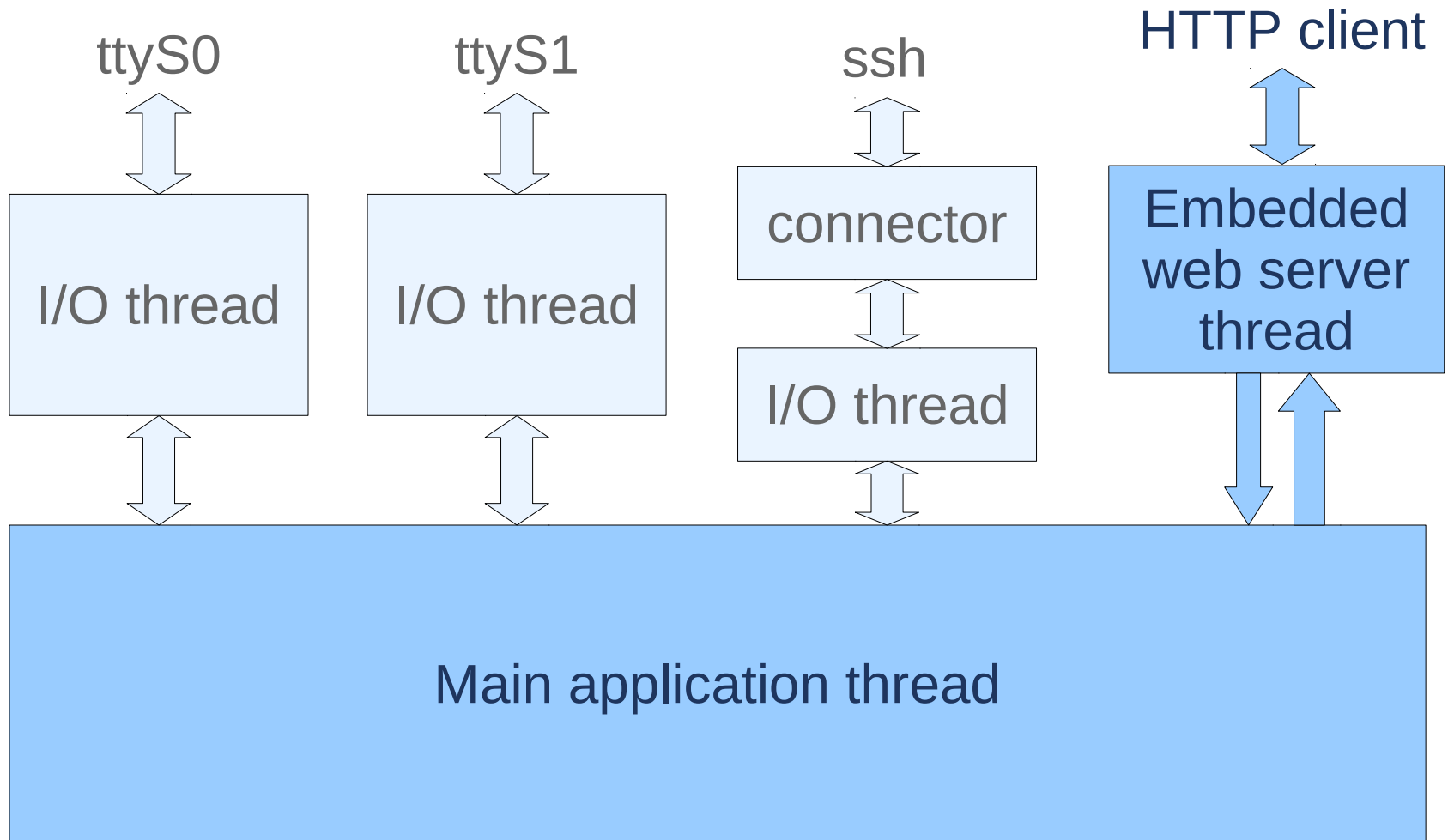


- Browsers are universal
  - availability
  - familiarity
  
- The network is everywhere
  
- Browsers allow a rich interface
  
- A web-interface is the cheapest interface
  - only software (and a network connection)

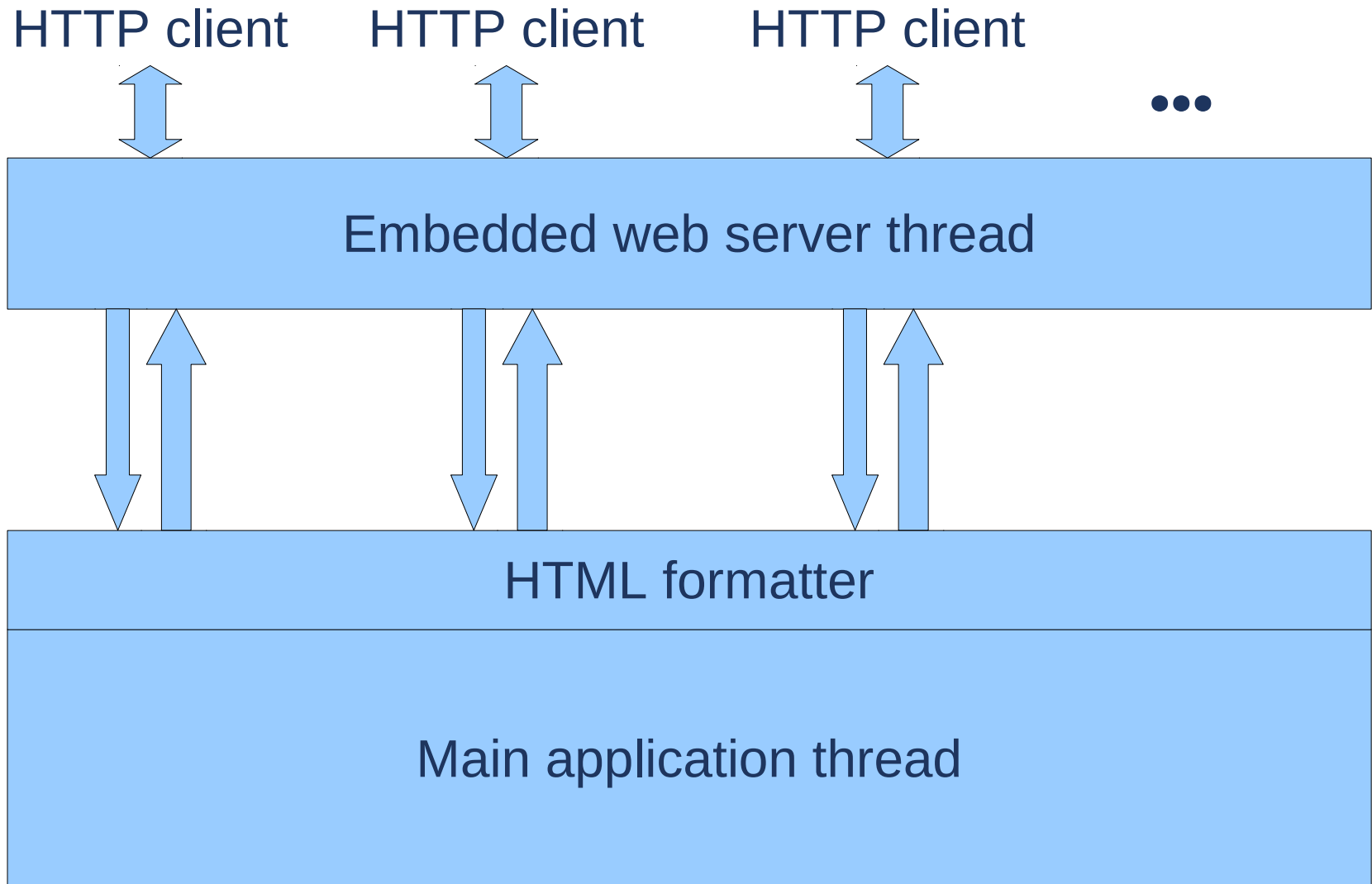
# Adding a web interface to traditional firmware architecture has many problems



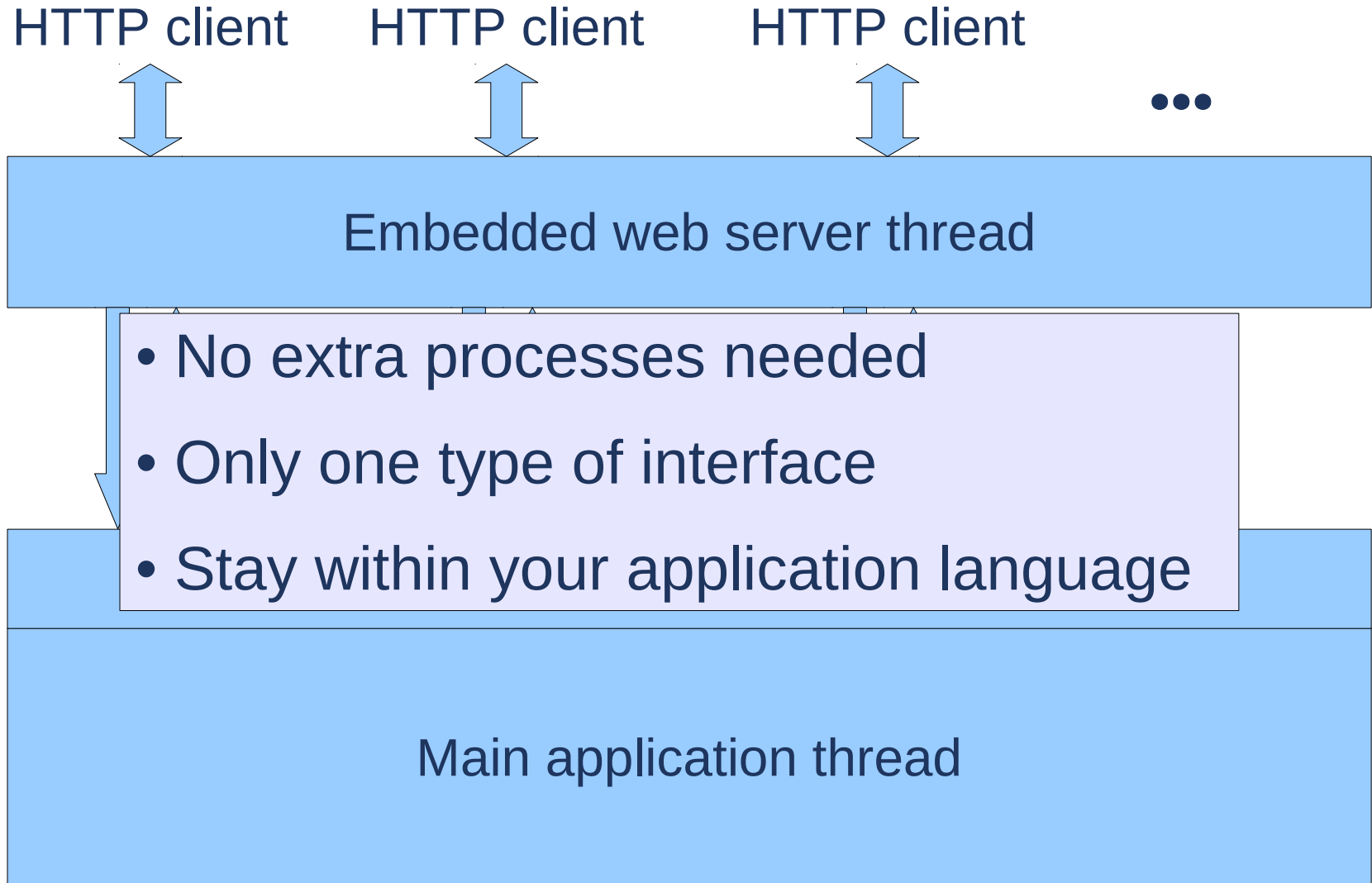
# Web interface architecture: embedded



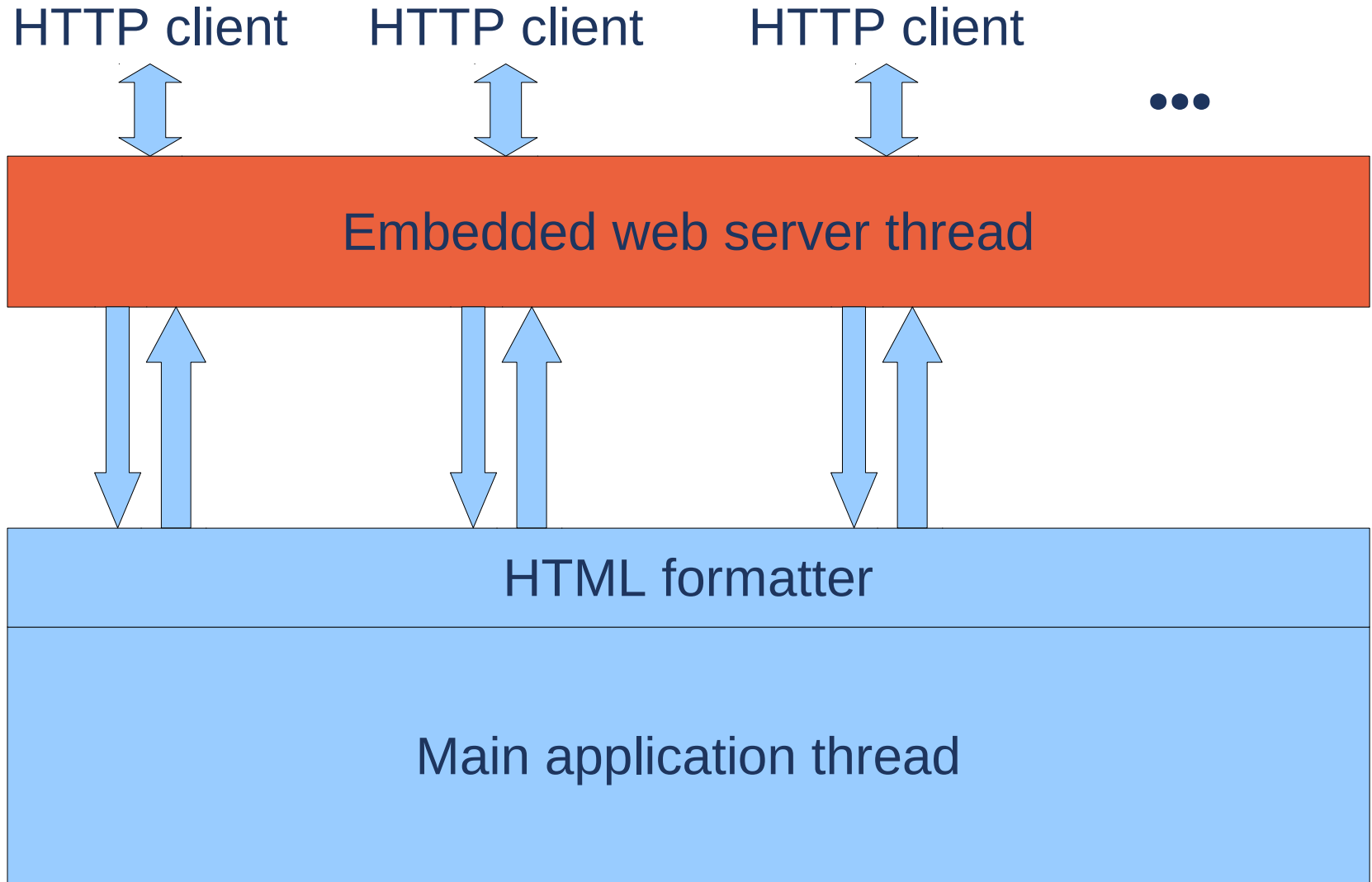
# Web interface architecture: embedded and no other interfaces



# Web interface architecture: embedded and no other interfaces



# The embedded web server



- ❑ Listen & accept connections
- ❑ Parse HTTP request (GET/POST/PUT/...)
- ❑ Parse HTTP headers
- ❑ Parse URI and find a handler for it
  - Serve static pages directly from filesystem
  - Provide MIME-type for static pages
  - Serve dynamic pages **by calling a handler function**

- ❑ **40 Kilobytes**
- ❑ License: MIT/X11 (BSD like)
- ❑ Small, easy to hack
- ❑ Actively developed
- ❑ Multiplatform (Posix, RTEMS, Windows (mingw/MSVC))
- ❑ Unit tests
- ❑ SSI (Server Side Includes), Directory listing, ...
- ❑ CGI calls
- ❑ HTTP Authenticate:, Range:, Etag:, ...
- ❑ SSL (see later)

<http://code.google.com/p/mongoose/>

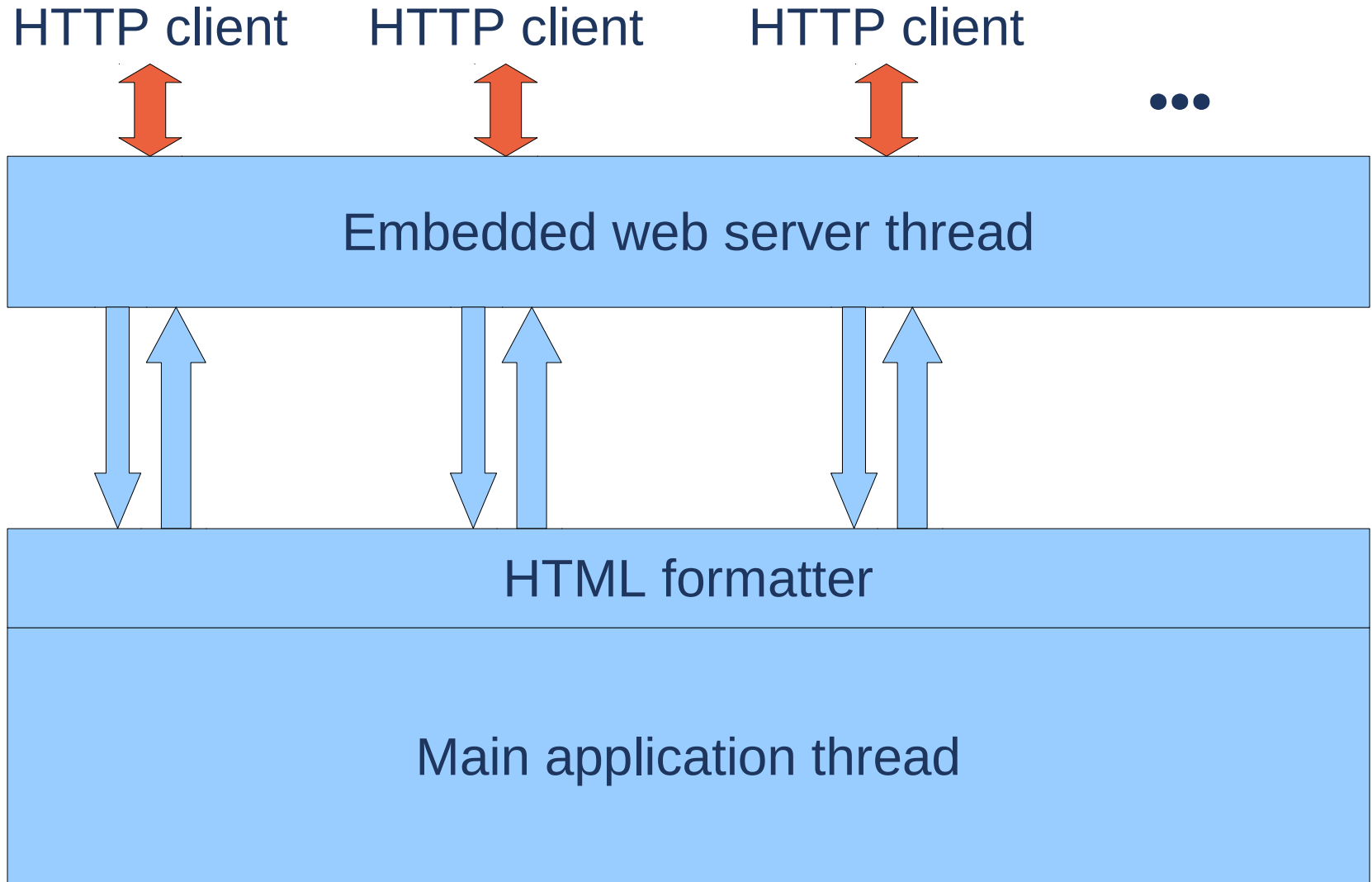


```
ctx = mg_start(&event_handler, NULL, options);  
...  
mg_stop(ctx);
```

```
static void *event_handler(enum mg_event event,  
    struct mg_connection *conn,  
    const struct mg_request_info *request_info)  
{  
    if (event == MG_NEW_REQUEST) {  
        if (!request_info->is_ssl) {  
            return redirect_to_ssl(conn, request_info);  
        } else if (!is_authorized(conn, request_info)) {  
            return redirect_to_login(conn, request_info);  
        } else if (strcmp(request_info->uri, "/get_messages") == 0) {  
            return ajax_get_messages(conn, request_info);  
        }  
    }  
    return NULL;  
}
```

- ❑ Not used very heavily → limited testing
- ❑ Not a lot of API for embedded handlers
  - e.g. generate standard headers
  - e.g. calculate Etag: etc.
- ❑ No IPv6 support (yet)
- ❑ Still very young
  - API subject to change
  - Will it survive?
  
- ❑ Contributions very welcome!

- Klone <http://www.koanlogic.com/klone/>
  - GPL/Commercial
  - PHP-like embedding of C/C++
  - Slightly larger (?); certainly more complex
  - IPv6 support
  
- libwthttpd (see later)
  
- Non-embedded
  - Apache
  - Lighttpd
  - Monkey
  - ...



Web based embedded system is vulnerable

- Port 80 is attacked
- Anybody can try to connect
- Text based communication → buffer overflows
- Authentication → password sniffing
- Request forgery and replay attacks

- ❑ Time out connections  
otherwise you run out of threads
  
- ❑ HTTP Digest Authentication  
otherwise passwords can be sniffed
  
- ❑ URL-encoding of session
  - Always use a different URL
  - If bookmarked → redirect to login page first
  
- ❑ SSL/TLS

- GnuTLS
  - License: LGPL
  - Pretty complete
- OpenSSL
  - License: BSD with advertising clause
  - Most well-known
  - Large and clumsy
- CyaSSL
  - License: GPL/Commercial
  - Specifically targeted at embedded
    - focuses on most used features
  - Optimized for speed (e.g. assembly for embedded uPs)
  - OpenSSL API (simplified)

# Comparison of SSL libraries: Protocol support and library size

|         | SSLv2 | SSLv3 | TLSv1.0 | TLSv1.1 | TLSv1.2 |
|---------|-------|-------|---------|---------|---------|
| GnuTLS  | No    | Yes   | Yes     | Yes     | Yes     |
| OpenSSL | Yes   | Yes   | Yes     | No      | No      |
| CyaSSL  | No    | Yes   | Yes     | Yes     | Yes     |

|         | Debian x86 | OpenWrt MIPS |
|---------|------------|--------------|
| GnuTLS  | 944K       | 323K         |
| OpenSSL | 1649K      | 506K         |
| CyaSSL  | 90K        | 60K          |





## This Connection is Untrusted

You have asked Firefox to connect securely to **localhost**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

Get me out of here!

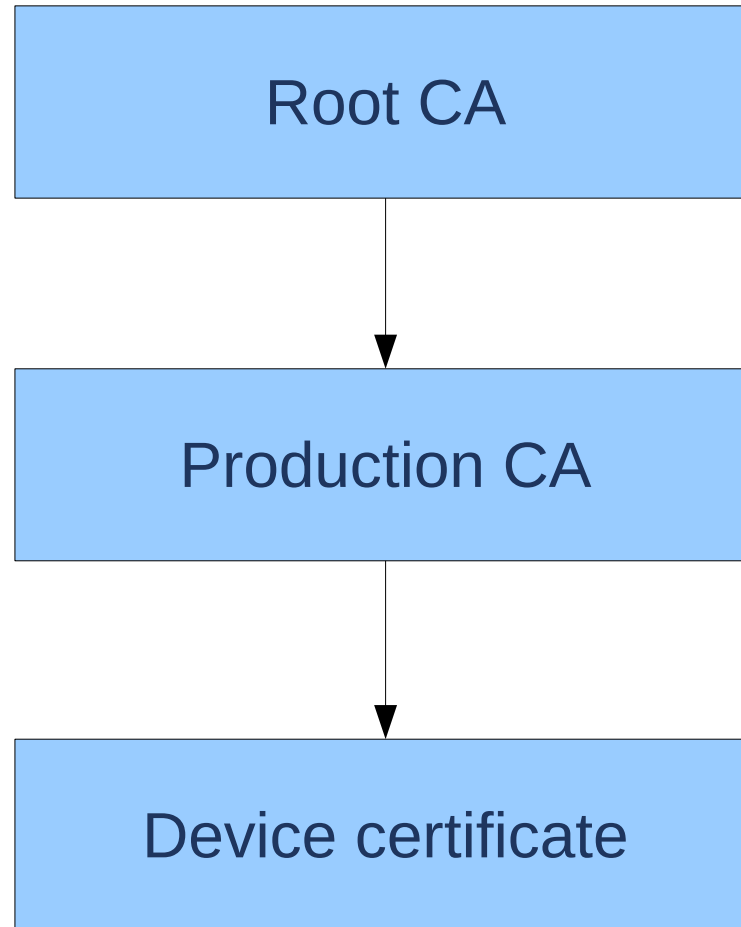
### Technical Details

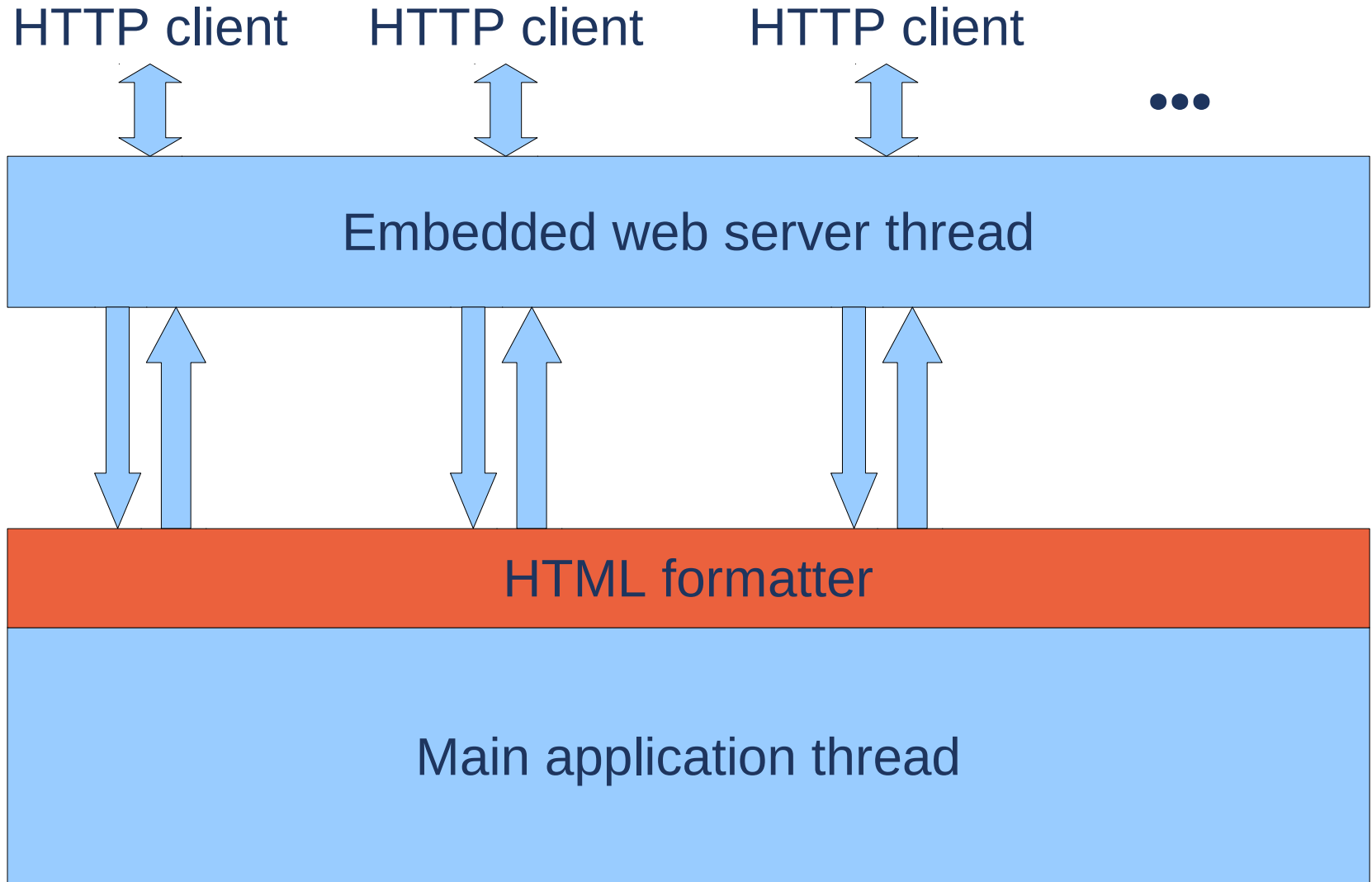
localhost uses an invalid security certificate.

The certificate is not trusted because it is self-signed.  
The certificate is only valid for 10.1.2.3

(Error code: sec\_error\_untrusted\_issuer)

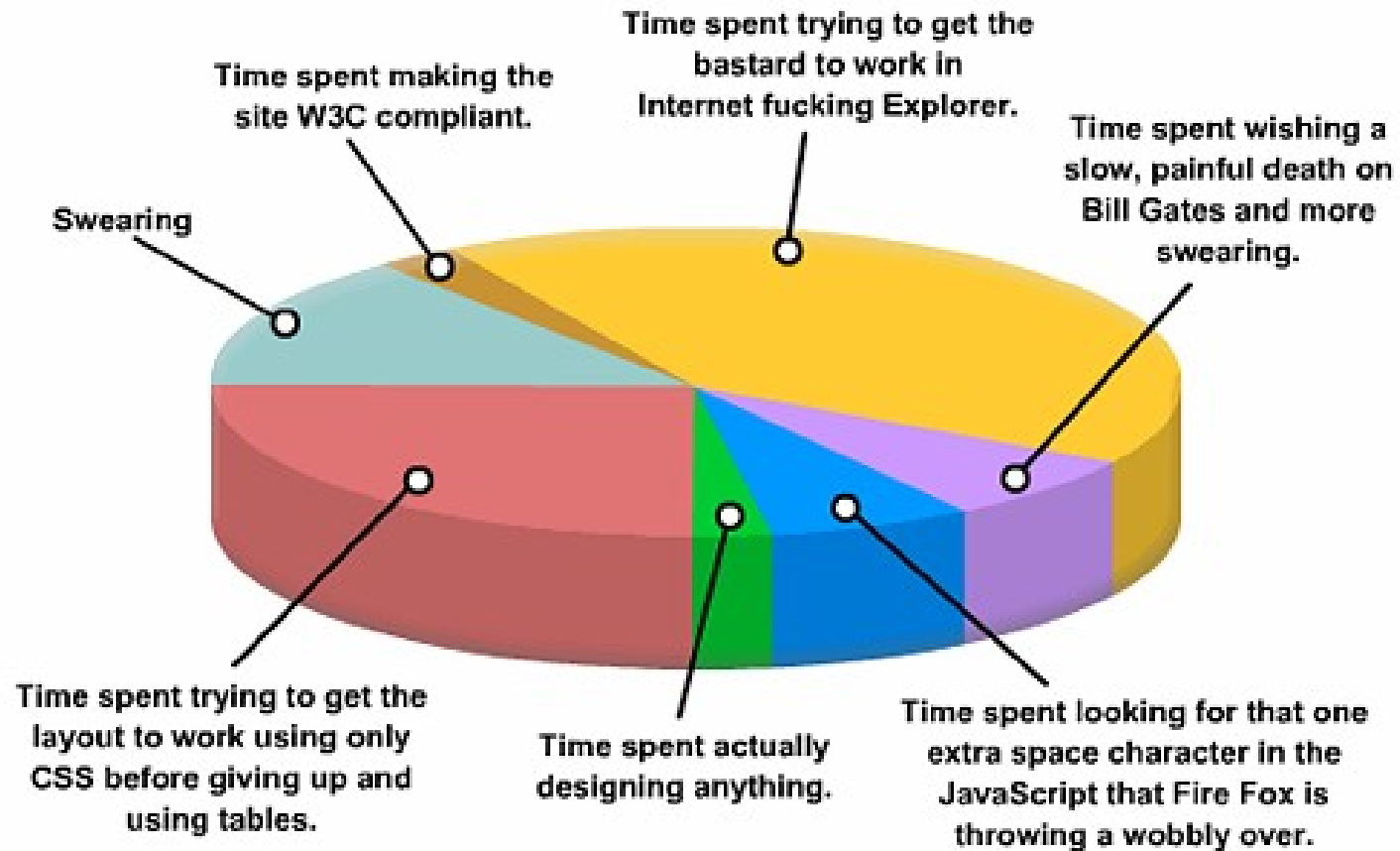
### I Understand the Risks





- ❑ Status pages
  - AJAX or long polling to refresh it
- ❑ Forms to manipulate settings
  - CSS to make it look nice
  - Javascript (jQuery) to pre-verify constraints
- ❑ Graphical output
  - Image maps
  - HTML5 canvas
  - Javascript plotting
- ❑ Internationalization: serve pages in the user's language
  - Accept-Language
  - Translations

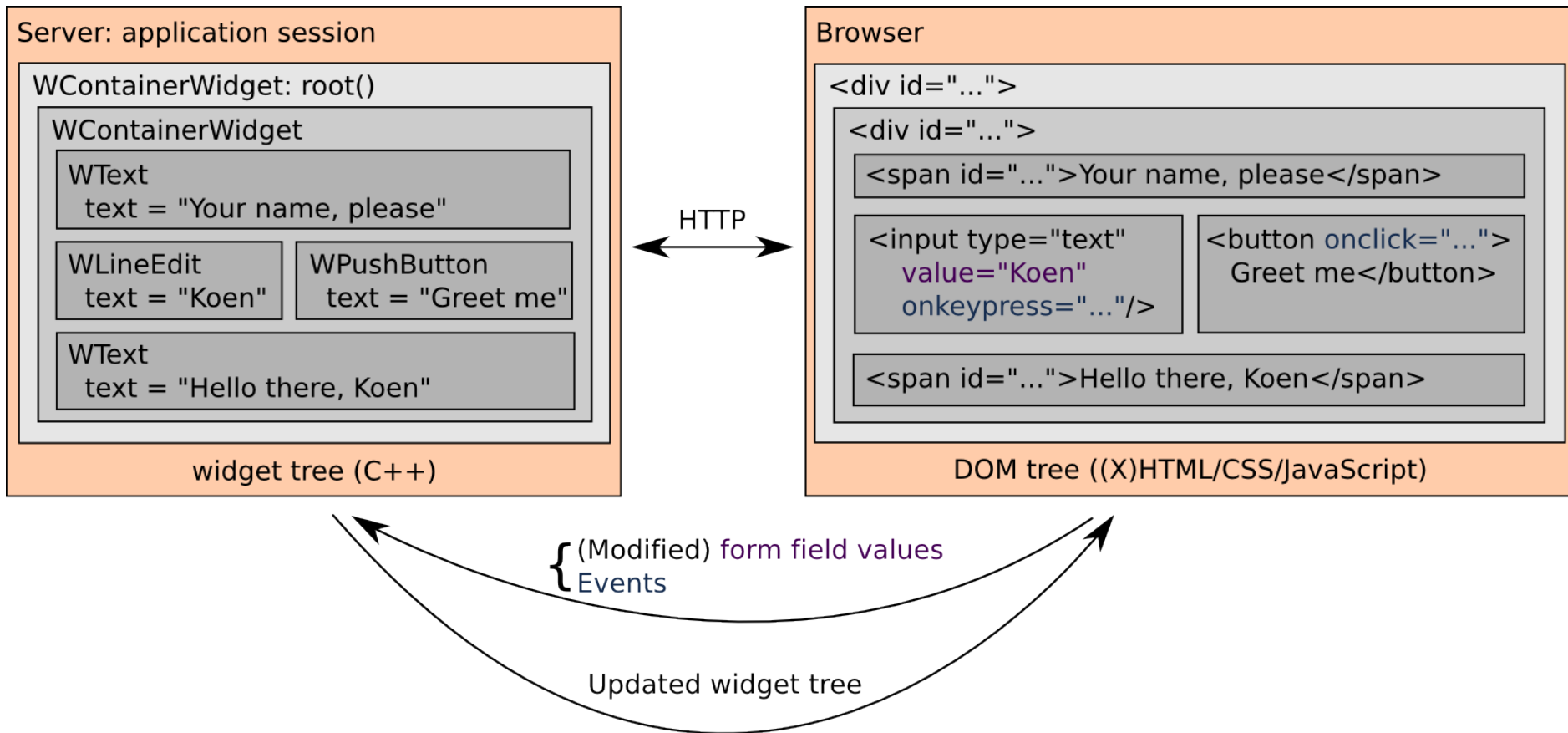
## TIME BREAKDOWN OF MODERN WEB DESIGN



©2006 Alan "IE users must DIE" Foreman [poisonedminds.com](http://poisonedminds.com)

- JQuery <http://jquery.com/>
  - 30K
  - License: MIT
  - *The* standard on the web
  - Still need to write a lot of Javascript→ doesn't solve all cross-browser issues
  
- Wt <http://www.webtoolkit.eu/wt>
  - C++ UI library
  - 2MB (incl. Webserver)
  - License: GPL/Commercial
  - Stay within 1 programming language  
no javascript required

# Wt is a UI library for *web* applications



- ❑ Web interface is the cheapest UI for embedded systems
- ❑ Embed it directly into your application  
mongoose, kclone, libwhttp
- ❑ Don't forget about security
- ❑ Cross-browser support is difficult  
→ use jQuery or Wt





[www.mind.be](http://www.mind.be)

[www.essensium.com](http://www.essensium.com)

**Essensium NV**  
**Mind - Embedded Software Division**  
**Gaston Geenslaan 9, B-3001 Leuven**  
**Tel : +32 16-28 65 00**  
**Fax : +32 16-28 65 01**  
**email : [info@essensium.com](mailto:info@essensium.com)**