Scheme in Industrial Automation

Marco Benelli
<mbenelli@yahoo.com>

SCADA
Supervisor Control And Data Acquisition

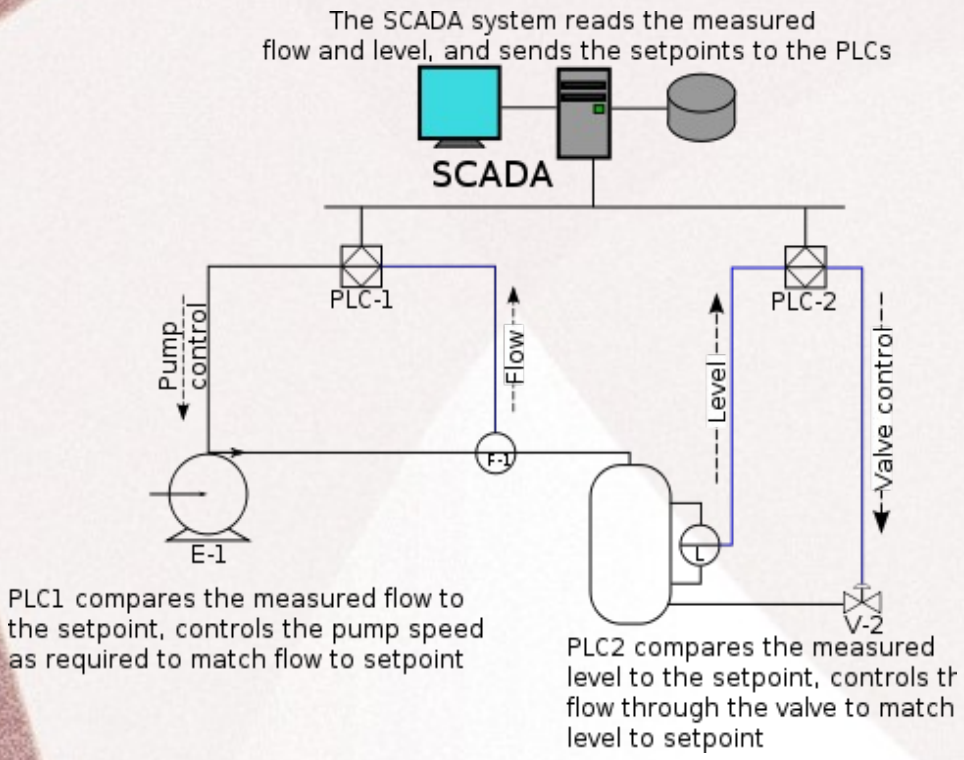


Image from wikipedia

- Sensors / Actuators
- Microcontrollers (PLCs)
- Supervisor
- Human Machine Interface

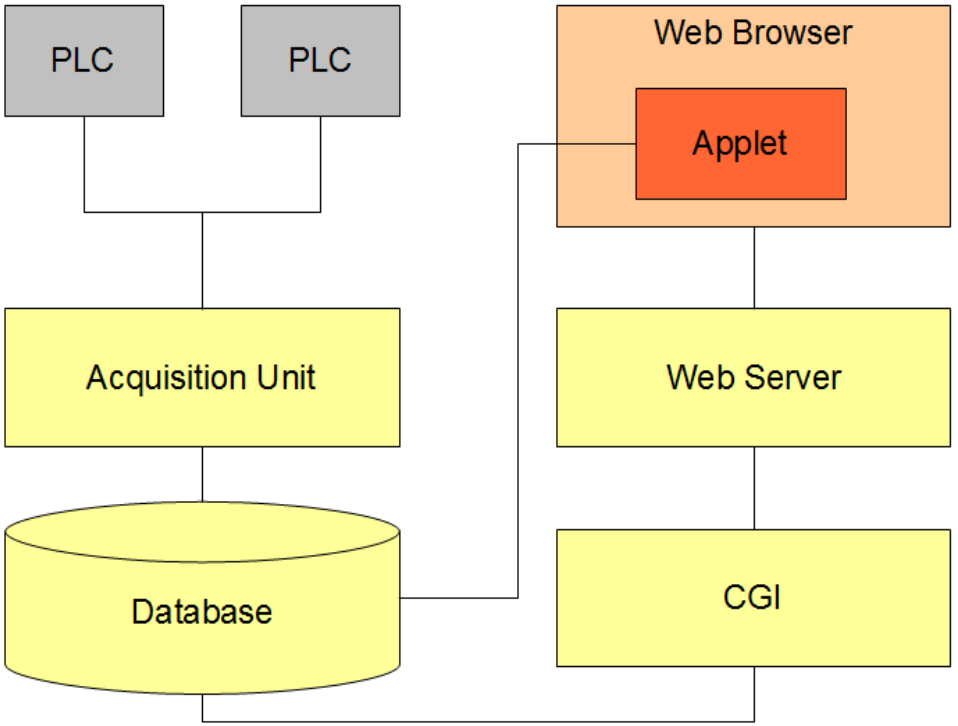
Platforms

- ARM, GNU/Linux
- X86, GNU/Linux
- SH-2, uClinux (no MMU)

CPU: from 200Mhz to >1Ghz

RAM: from 128Mb to 1Gb

Legacy System



This diagram show the typical system: it exists in several variations.

Problems

- Java applet: slow loading time, need certification, poor concurrency handling
- Web pages: domain experts are typically not skilled in writing web pages
- CGIs: written in C for performance, they were just an interface to database, poor client-server decoupling
- Supervisor – HMI interaction relied heavily on database, slow.

What was needed

- Source to source transformations
- Domain Specific Languages
- **Flexible** but **efficient** and **portable** server-side framework
- Good integration with exisisting tools, for a smooth transition

Gambit-C Scheme

- Excellent tools for source to source transformation and DSL (macros, functional programming, SXML)
- Excellent performance and portability (compilation to C)
- Extensions (green-threads, namespaces, define-macro, FFI, decarations, extended ports, built-in SRFIs)

Substituting Java Applets

Applets was used for:

- Data charts: interactive view of data history. Easily replaced with Javascript libraries.
- Synopsis: summary view of the plant, with interactive graphical elements indicating status of device and current values.

Synopsis

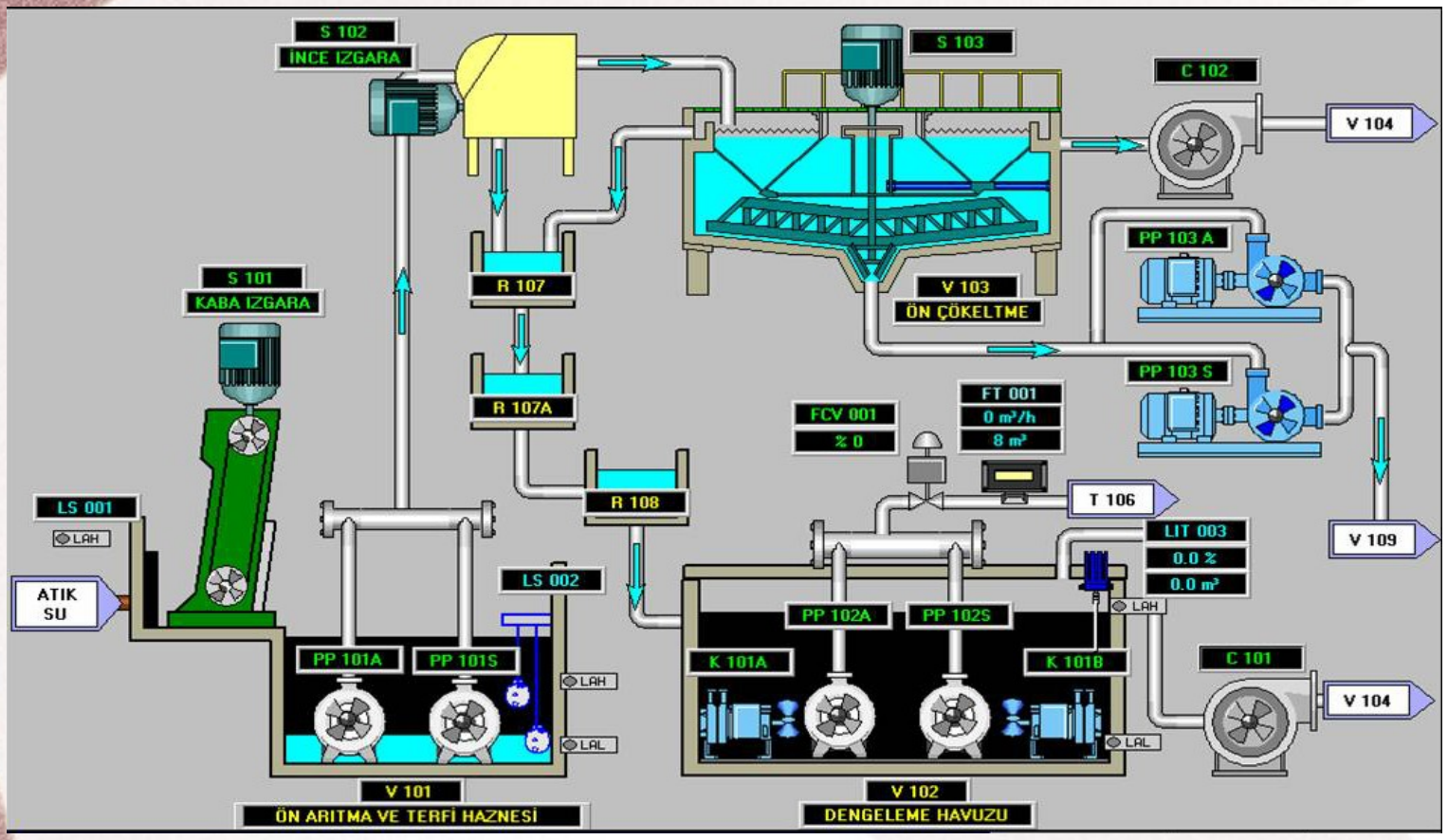
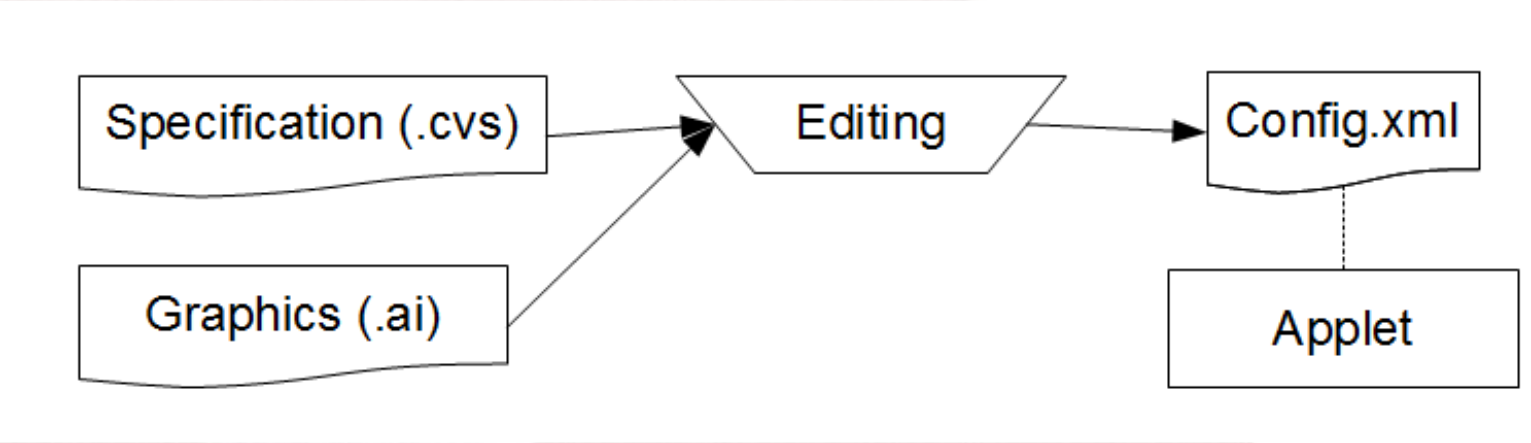
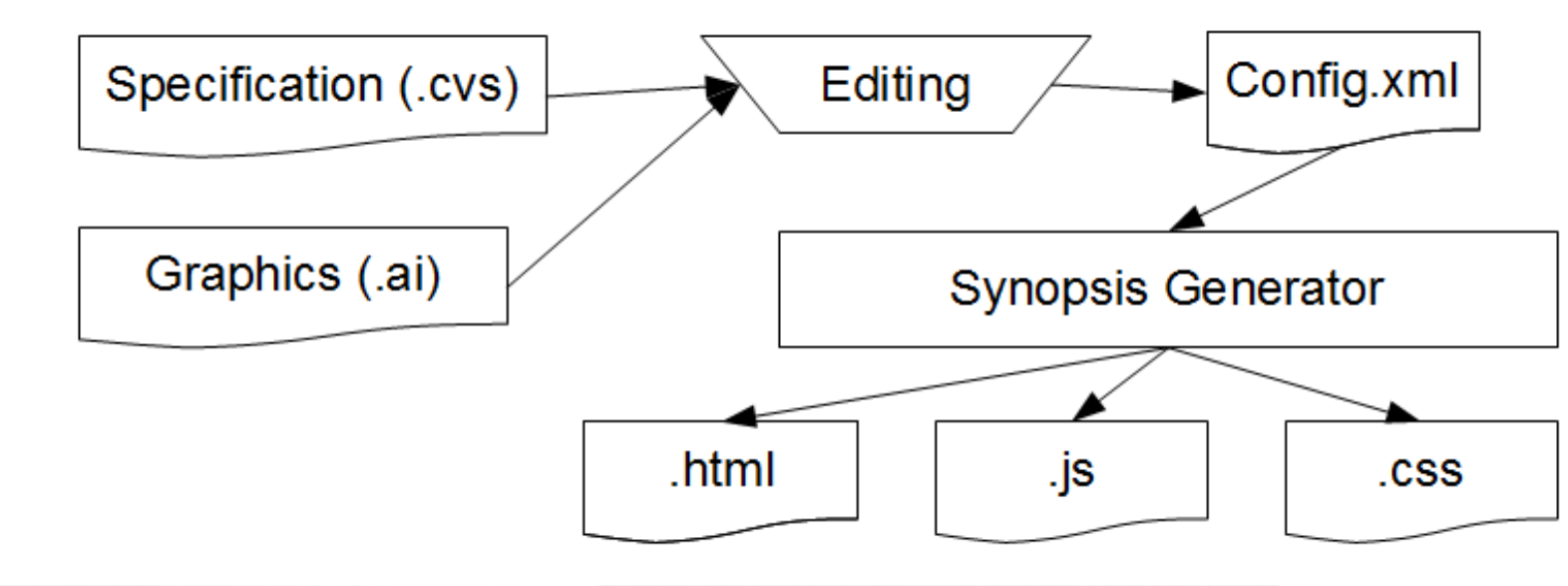


Image for illustrative purpose only, it does not represent the real system

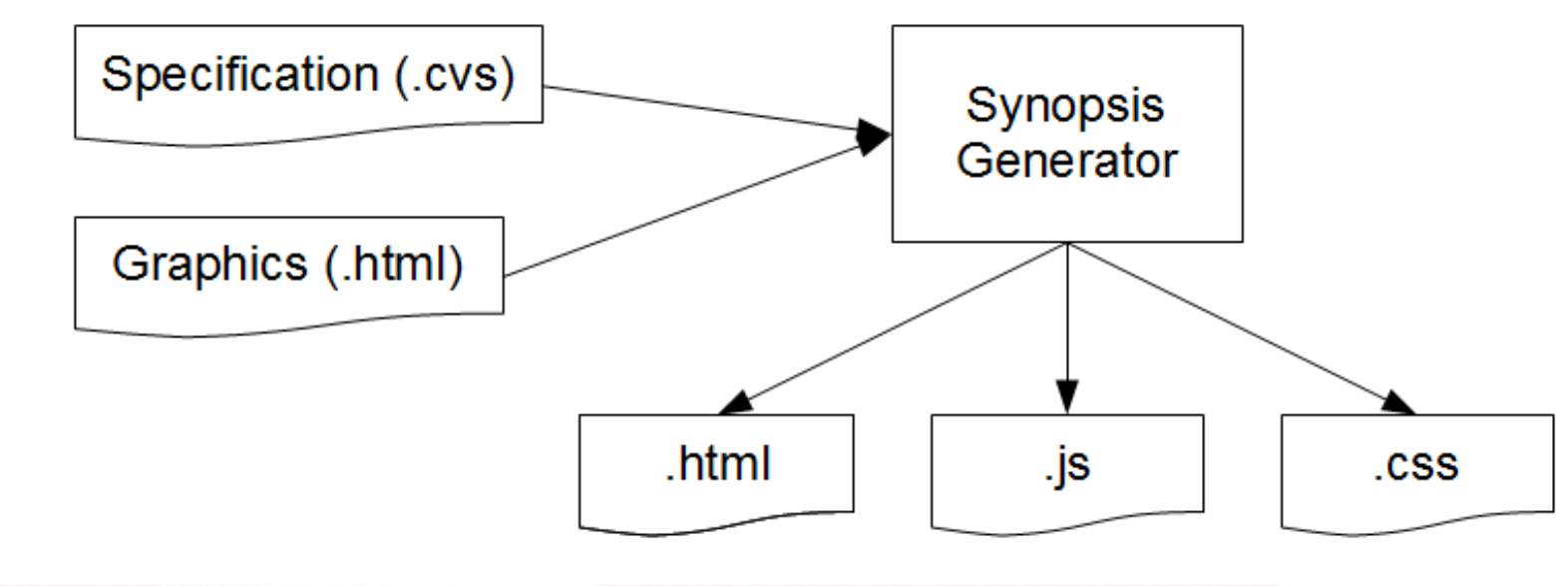
Original Workflow



Dropped the applet



Automated Configuration



Writing web pages

Web Pages

Domain Specific Language for generating static web pages that contains AJAX calls to retrieve dynamic content from the server.

In this way, the existing CGI layer was still used.

Pages Generation (db)

```
(page "Allarms" db
  (dbtable alarmTable
    (sql
      (select description tag status block mail sms)
      (from alarmlist)
      (orderby status))
    ("Description"
     "Tag"
     ("Status"   ("Y" "led_red.png") ("N" "led_grey.png"))
     ("Blocking" ("Y" "tick_16.png") ("N" "void_16.png"))
     ("Mail"     ("Y" "led_red.png") ("N" "led_grey.png"))
     ("Sms"      ("Y" "led_red.png") ("N" "led_grey.png")))))
```

Code for illustrative purpose only

Pages Generation (db)

```
(page "Allarms" db
  (dbtable alarmTable
    (sql
      (select description tag status block mail
      (from alarmlist)
      (orderby status))
    ("Description"
     "Tag"
     ("Status"   ("Y" "l                    png"))
     ("Blocking" ("Y" "tick_16.png") ("N" "void_16.png"))
     ("Mail"     ("Y" "led_red.png") ("N" "led_grey.png"))
     ("Sms"      ("Y" "led_red.png") ("N" "led_grey.png")))))
```

Page type: this page Contains ajax call that fill table(s) quering the database

Define an html page.

Code for illustrative purpose only

Pages Generation (db)

```
(page "Allarms" db

  (dbtable alarmTable

    (sql

      (select description tag status block mail sms)

      (from alarmlist)

      (orderby status))

    ("Description"

     "Tag"

     ("Status"   ("Y" "led_red.png") ("N" "led_grey.png"))

     ("Blocking" ("Y" "tick_16.png") ("N" "void_16.png"))

     ("Mail"     ("Y" "led_red.png") ("N" "led_grey.png"))

     ("Sms"      ("Y" "led_red.png") ("N" "led_grey.png")))))
```

Define a table.

Embedded SQL

Code for illustrative purpose only

Pages Generation (db)

```
(page "Allarms" db
  (dbtable alarmTable
    (sql
      (select description tag status block mail sms)
      (from alarmlist)
      (orderby status))
    ("Description"
     "Tag"
     ("Status"   ("Y" "led_red.png") ("N" "led_grey.png"))
     ("Blocking" ("Y" "tick_16.png") ("N" "void_16.png"))
     ("Mail"     ("Y" "led_red.png") ("N" "led_grey.png"))
     ("Sms"      ("Y" "led_red.png") ("N" "led_grey.png")))))
```

Column label

Value – icon association

Code for illustrative purpose only

Page Generation (poll)

```
(page "Monitoring" poll
  (box monitor base
    (vbox
      (channel "Temperature" T001 T002 "\u00b0C")
      (channel "Humidity" T005 T006 "%")
    (hbox
      (label "Running") (led T010_0)
      (label "Auto") (led T011_2 yellow)
      (label "Alarm") (led T012_7 red)))
    (toolbar
      (btn6 "reset.png" (js (blink "T100_1")))
      (btn5 "leds.png" "leds-read.html"))))
```

Code for illustrative purpose only

Page Generation (poll)

```
(page "Monitoring" poll
  (box monitor base
    (vbox
      (channel "Temperature" T001 T002 "\u00b0C")
      (channel "Humidity" T005 T006 "%")
    (hbox
      (label "Running") (led T010_0)
      (label "Auto") (led T011_2 yellow)
      (label "Alarm") (led T012_7 red)))
    (toolbar
      (btn6 "reset.png" (js (blink "T100_1")))
      (btn5 "leds.png" "leds-read.html"))))
```

Periodically ask values to the server

Tags

Code for illustrative purpose only

Page Generation (poll)

```
(page "Monitoring" poll

  (box monitor base
    (vbox
      (channel "Temperature" T001 T002 "\u00b0C")
      (channel "Humidity" T005 T006 "%")
    (hbox
      (label "Running") (led T010_(
      (label "Auto") (led T011_2 y(
      (label "Alarm") (led T012_7 red)))
    (toolbar
      (bt                                  100_1")))
      (bt                                  1"))))
```

Class attribute

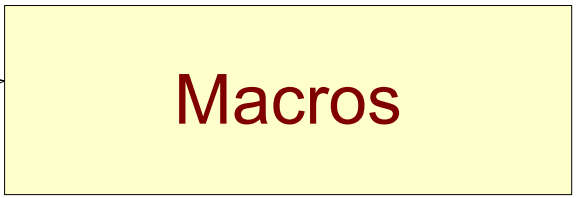
Id attribute

Create an html **div**

Code for illustrative purpose only

Page Generation (poll)

```
(page "Monitoring" poll
  (box monitor base
    (vbox
      (channel "Temperature" T001 T002 "\u00b0C")
      (channel "Humidity" T005 T006 "%")
    (hbox
      (label "Running") (led T010_0)
      (label "Auto") (led T011_2 yellow)
      (label "Alarm") (led T012_7 red)))
    (toolbar
      (btn6 "reset.png" (js (blink "T100_1")))
      (btn5 "leds.png" "leds-read.html"))))
```



Code for illustrative purpose only

Page Generation (poll)

```
(page "Monitoring" poll
  (box monitor base
    (vbox
      (channel "Temperature" T001 T002 "\u00b0C")
      (channel "Humidity" T005 T006 "%")
    (hbox
      (label "Running") (led T010_0)
      (label "Auto") (led T011_2 yellow)
      (label "Alarm") (led T012_7 red)))
    (toolbar
      (btn6 "reset.png" (js (blink "T100_1")))
      (btn5 "leds.png" "leds-read.html"))))
```

Embedded Javascript

Code for illustrative purpose only

Substituting CGIs

Klio Tools

- Libraries from Scheme community (SRFIs, irregex, SSAX, …)
- Libraries from Gambit (base64, http, ...)
- Custom libraries (sqlite bindings, csv...)
- Web Server
- Protocols useful in SCADA (fetchwrite, modbus)

Klio Web Server

- Based on example in Gambit-C sources
- Extended to be HTTP 1.1 compliant
- Born as test/simulation environment
- Focus in creating high-interactive GUIs for a small number of users
- Performant and scalable

Gambit-C example web server

- Multithreaded server
- URI encoding/decoding
- Dispatch table
- Html generation
- Support for continuation-based application

Klio Web Server HTTP/1.1

- Persistent connections
- Caching
- Conformant headers (ie: date)
- Conformant response codes
- Chuncked data

Klio Web Server Adds-on

- Mime type support
- Session management
- HTTP basic authentication
- Cookie based authentication
- Cgi support
- HTTPS support (work in progress)
- Web Socket support (work in progress)

Database: sqlite bindings

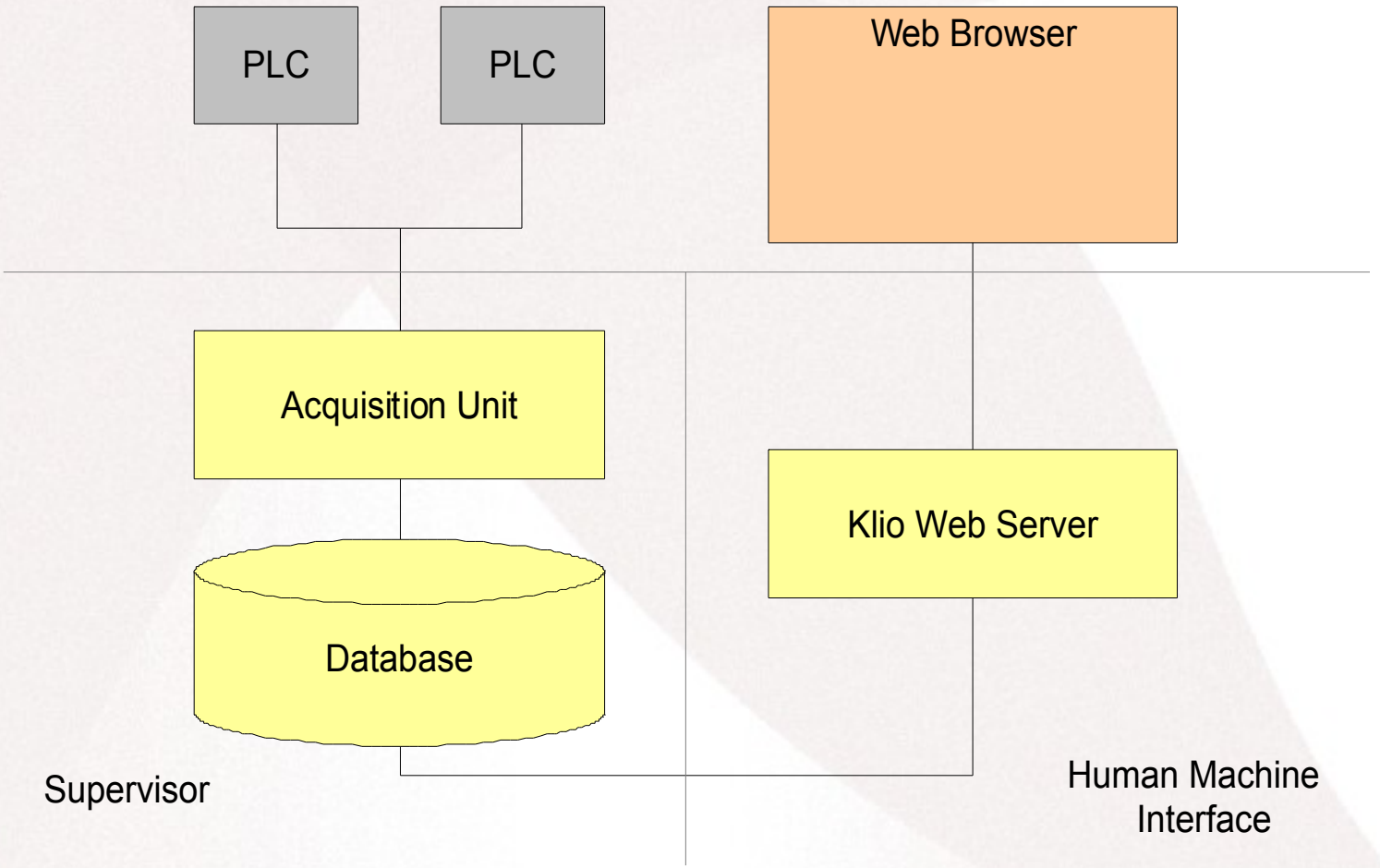
```
(define sqlite3-open
  (c-lambda (char* sqlite3**) int “sqlite3_open”))

(define %sqlite3-open
  (c-lambda (char*) sqlite3*
#<<C-END
  sqlite3* db;
  int res = sqlite3_open(___arg1, &db);
  ___result_voidstar = db;
C-END
))
```

Database: sqlite access via FFI

```
(define (open name)
  (let ((db (%sqlite-open name)))
    (if (zero? (sqlite3-errcode db))
        db
        (raise (sqlite3-errmsg db)))))
```

New System


PLC
PLC
Web Browser
Acquisition Unit
Klio Web Server
Database
Supervisor
Human Machine
Interface

Improving Supervisor – HMI interaction

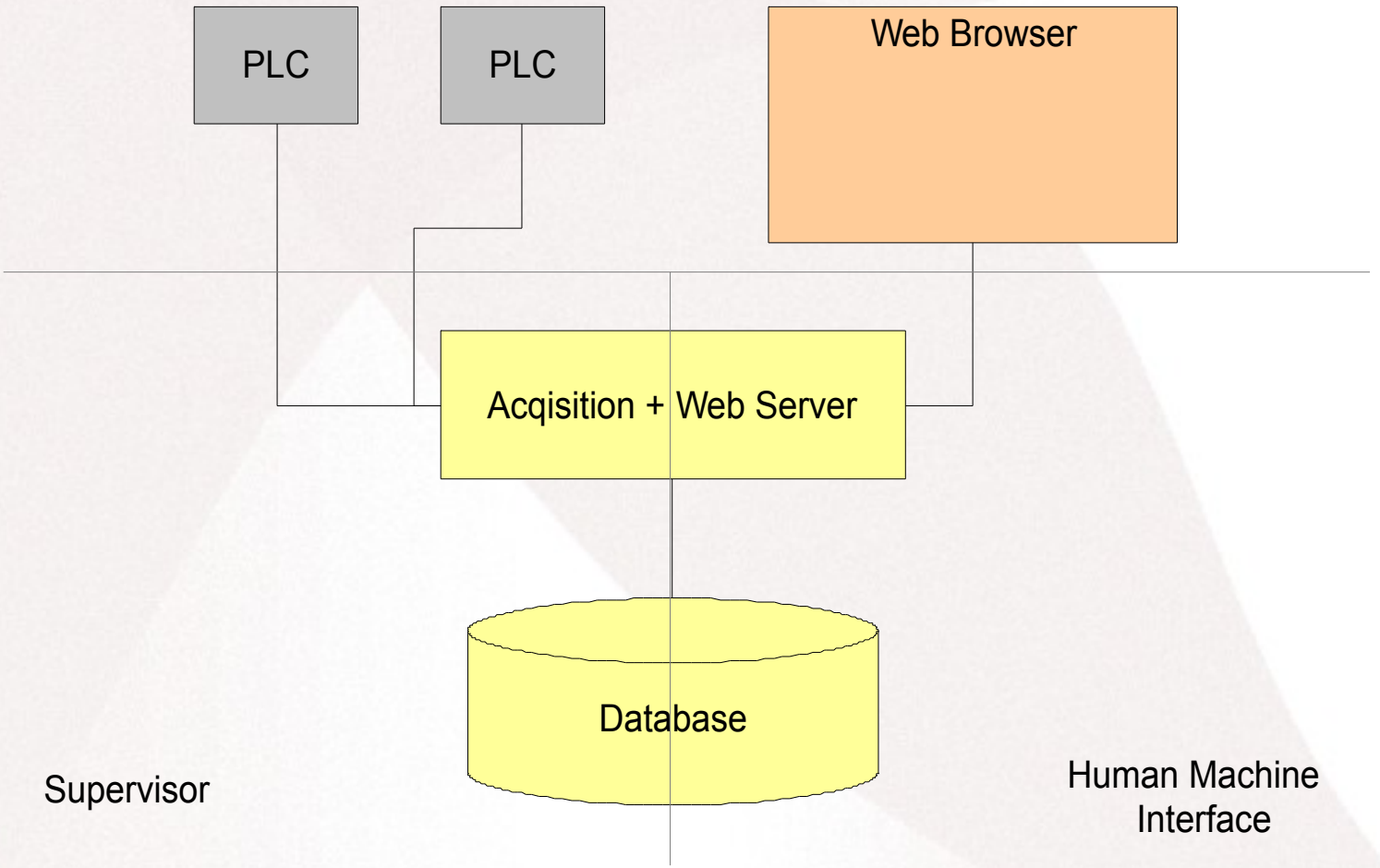
Database access

- With CGIs a lot of “database busy” exceptions were raised on smallest machines
- Retry queries was not an option: the exceptions was raised too frequently
- With Klio a lot less exceptions were raised: queries retrying worked
- But it could be better...

Klio Scada

- Common field protocol (modbus, fetchwrite) implementation in Scheme
- A single application can serve HTTP request and acquire data from PLCs
- Single database access point: concurrency handled by application
- Current data available to HMI without need to be written in database
- Not (yet) in production

Klio Scada


PLC
PLC
Web Browser
Acqisition + Web Server
Database
Supervisor
Human Machine Interface

Conclusions

- Scheme is an excellent tool even in the so called “real world”.
- Gambit-C is a very reliable implementation.
- Batteries not included, but you can find more than you think out of there.

References

- Klio tools:

  http://mbenelli.github.com/klio

- Gambit-C:

  http://dynamo.iro.umontreal.ca/~gambit