FEUERLABS

Connecting Devices

using exodevice

EUC 2013, Stockholm

Tony Rogvall
tony@feuerlabs.com
http://www.feuerlabs.com

Outline

- Introduction
- Exodevice
- Hardware API
- System API
- Services
- Application
- DEMO

FEUERLABS

Introduction

- Who we are
- What we do
- Why we do it

Exodevice - software

- collection of OTP applications on github
- collected with rebar
  - Feuerlabs/exoport
  - Feuerlabs/exodemo
- run most components on any machine
- build with yocto for various hardware
- use tetrapak from travelping, ipk/opk/deb

Exodevice - hardware

- Any decent hardware with linux support would probably work

- Add some i/o with piface

FEUERLABS

Hardware API

- gpio - Feuerlabs/gpio
- uart - tonyrog/uart
- spi - tonyrog/spi
- i2c - tonyrog/i2c
- eth - tonyrog/eth
- can - tonyrog/can
- piface - tonyrog/piface

GPIO interrupt

```
-module(gpio_example).
-export([test/0]).
test() ->
    gpio:set_interrupt(25, falling),
    receive
        {gpio_interrupt, 0, 25, Value} ->
          io:format("pin 25, high to low\n")
end.
```

System API

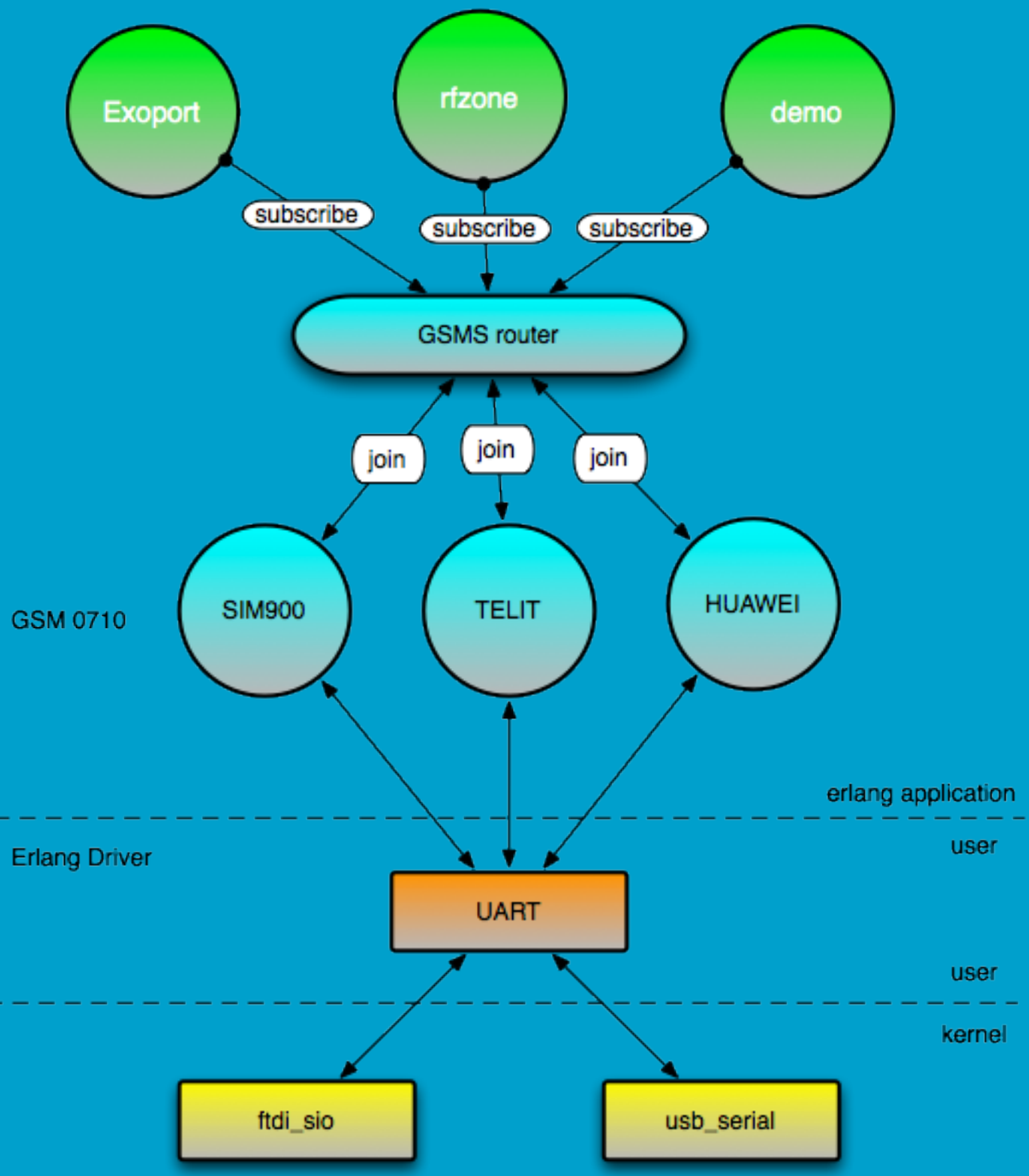
- afunix - tonyrog/afunix
- fnotify - Feuerlabs/fnotify
- kvdb - Feuerlabs/kvdb
- epx - tonyrog/epx (soon)

Services

- CANopen - tonyrog/canopen
- netlink - Feuerlabs/netlink (soon)
- dbus - tonyrog/dbus
- gsms (07.05) - tonyrog/gsms
- gsmux (07.10) - tonyrog/gsmux
- pppd_mgr - Feuerlabs/pppd_mgr
- nmea_0183 - tonyrog/nmea_0183
- bert - Feuerlabs/bert

FEUERLABS

gsms - SMS router



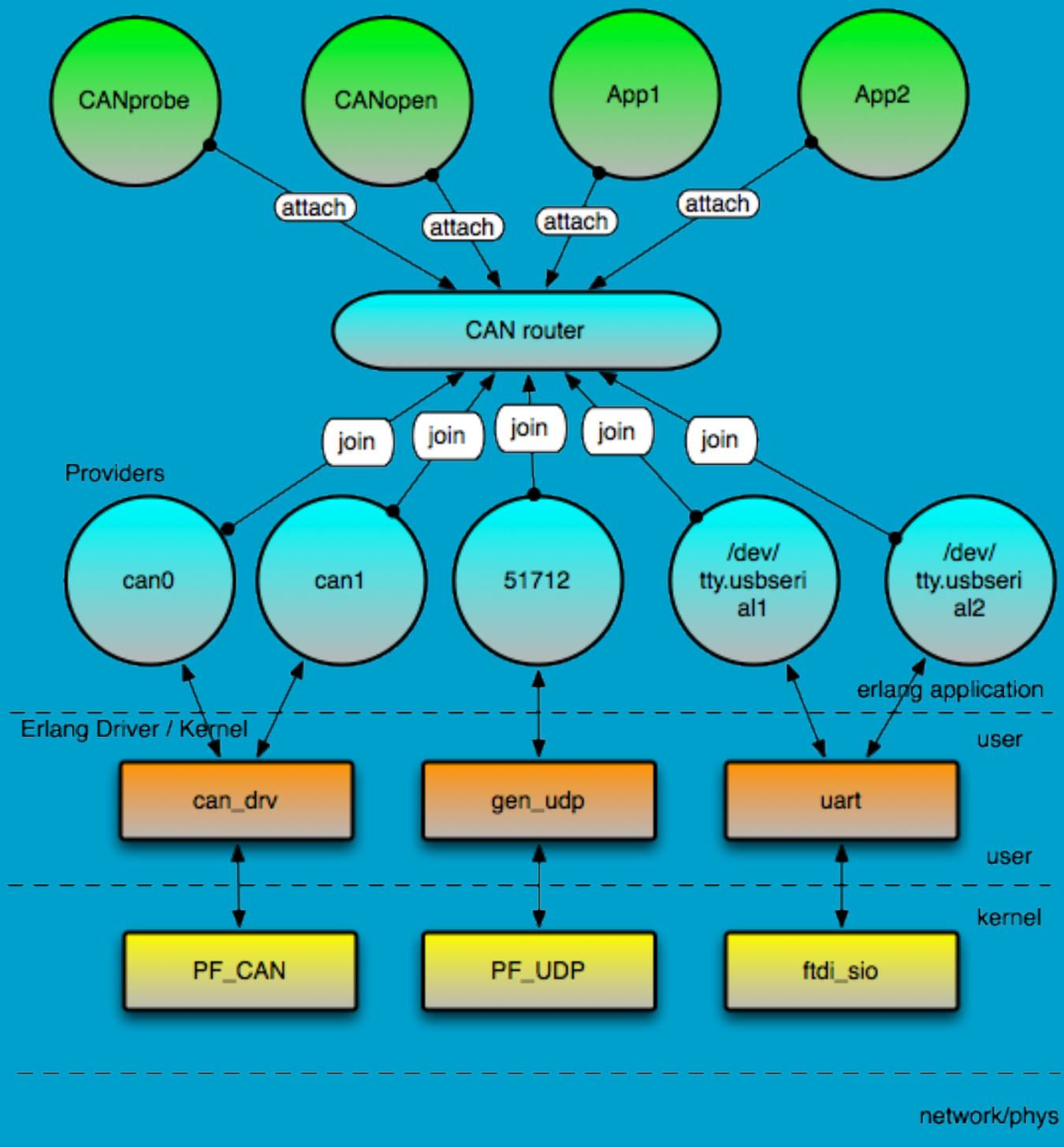
SMS service

```
-module(gpio_sms).
-include_lib("gsms/include/gsms.hrl").
-export([start/0, loop/1]).

-define(FILTER, "^[\\s]*[0-9]+[\\s]*[??+-][\\s]*$").
-define(is_pin(P), (((P) band (bnot 16#1f)) =:= 0)).

start() ->
    spawn(
      fun() ->
              {ok,Ref} = gsms:subscribe([{reg_exp,?FILTER}]),
              loop(Ref)
end).

loop(Ref) ->
    receive
        {gsms,Ref,Pdu} ->
            Sender = (Pdu#gsms_deliver_pdu.addr)#gsms_addr.addr,
            process_command(Sender, Pdu#gsms_deliver_pdu.ud),
            ?MODULE:loop(Ref)
    end.
```

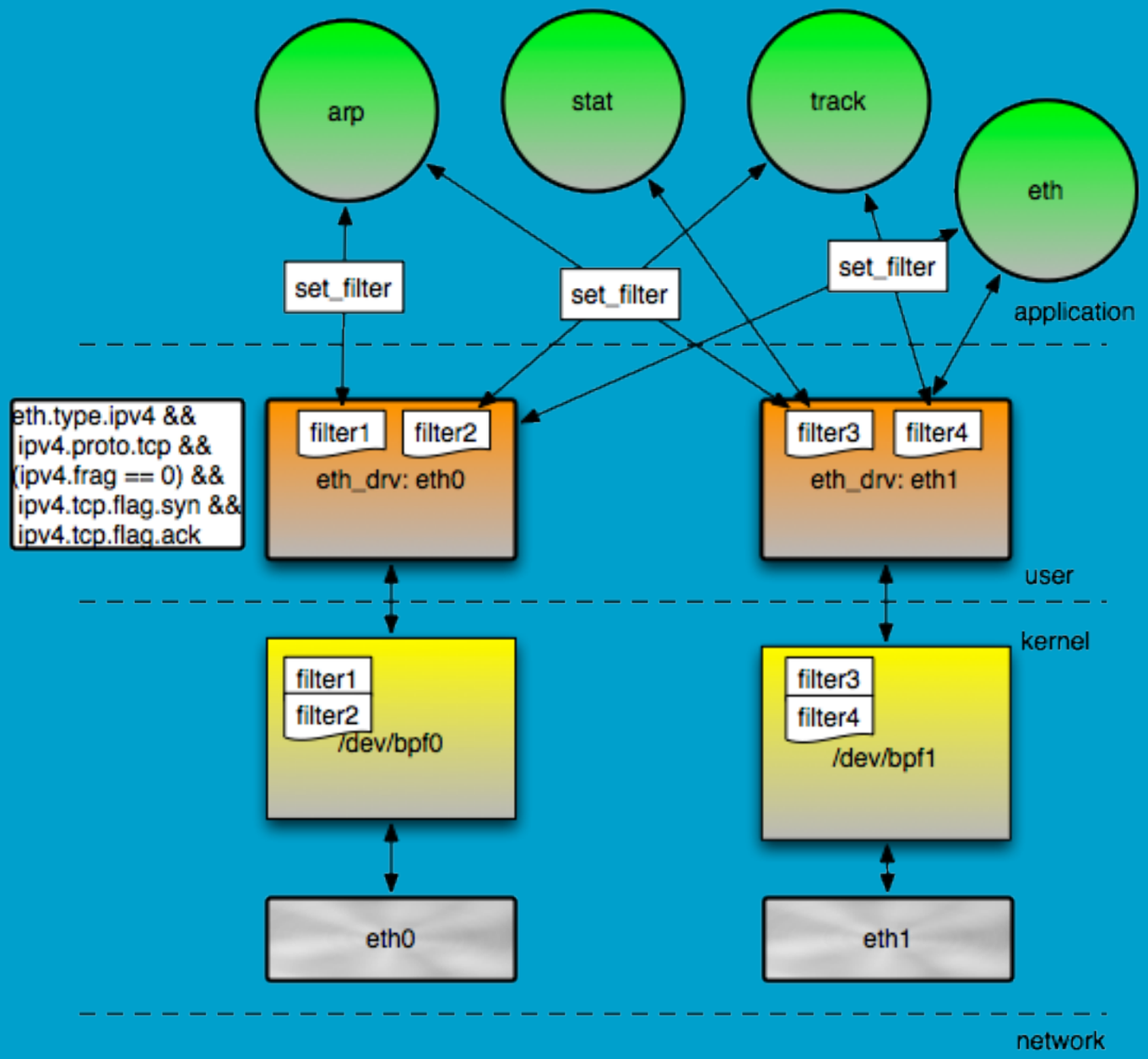
SMS service - 2

```
process_command(Sender, Message) ->
    case parse_message(Message) of
        {Pin,"+"} when ?is_pin(Pin) -> gpio:set(Pin);
        {Pin,"-"} when ?is_pin(Pin) -> gpio:clr(Pin);
        {Pin,"?"} when ?is_pin(Pin) ->
            case gpio:get(Pin) of
                {ok,Level} ->
                    gsms:send([{addr,Sender}],[Level+$0]);
                _ -> ignore
            end;
        _ -> ignore
    end.

parse_message(Message) ->
    case string:tokens(Message, " \t\r\n") of
        [T]     -> string:to_integer(T);
        [T1,T2] -> {(catch list_to_integer(T1)),T2}
    end.
```

can - CAN router



eth - packet filter



FEUERLABS

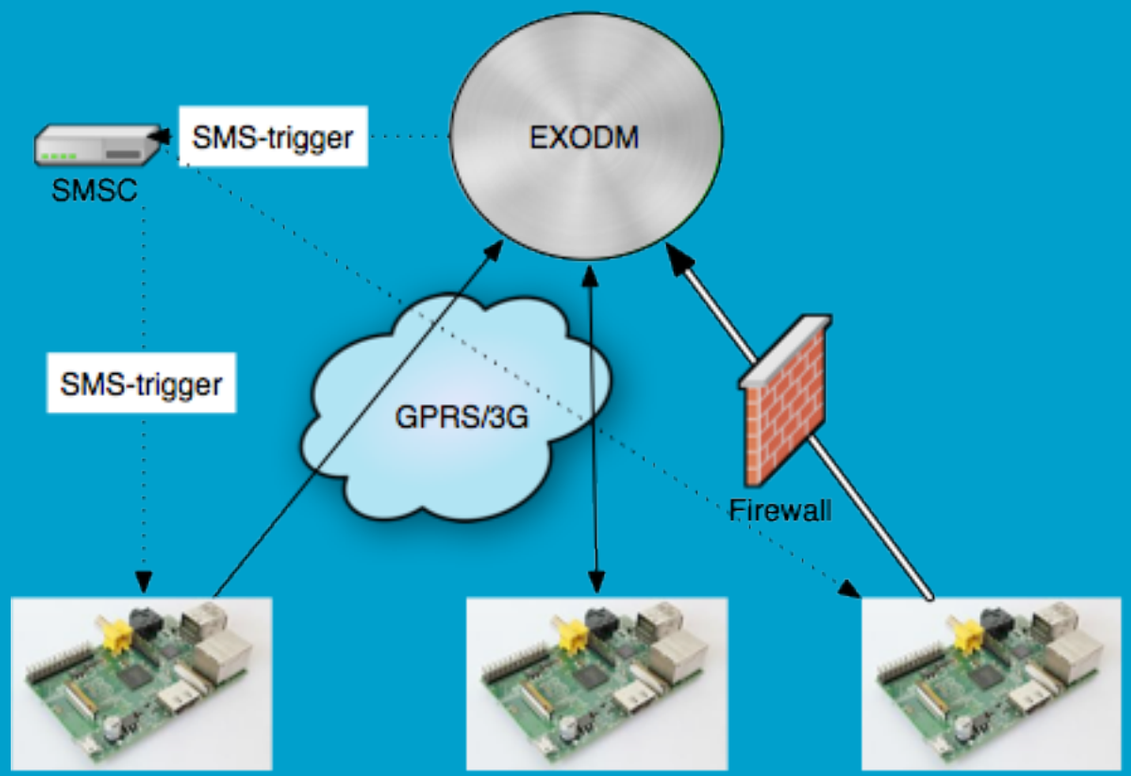
Exosense

- Store and forward RPC engine
- Like Netconf set-config/get-config
- Uses BERT-RPC instead of SOAP/XML
- Uses JSON-RPC on the customer application
- YANG data model

ExoDM/Trigger

- Simple SMS trigger to force device to connect to internet.
- Use default alphabet in regular SMS.
- “EXODM-RPC:(gprs|sms|none): <base64>

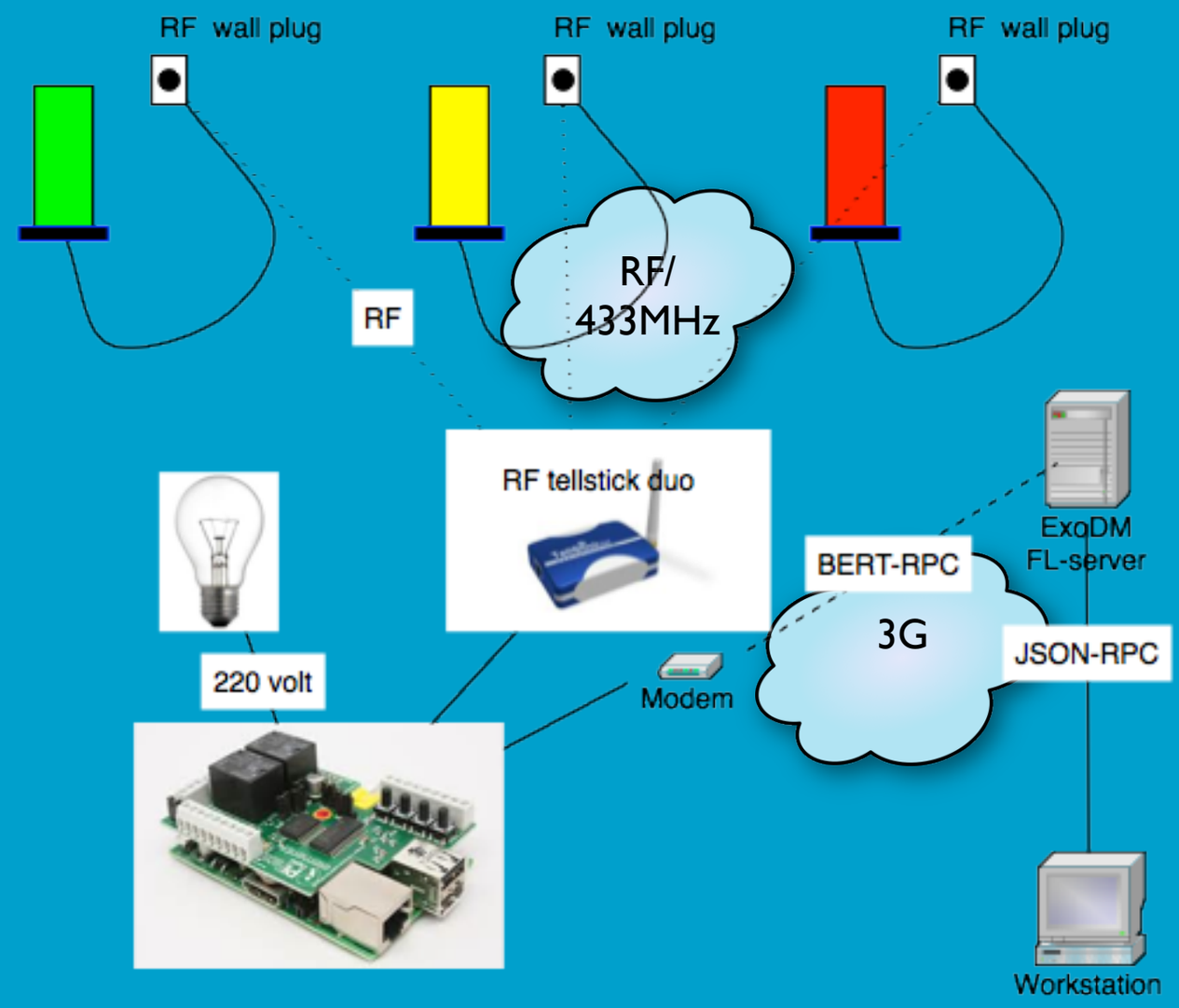
ExoDM/Trigger - 2



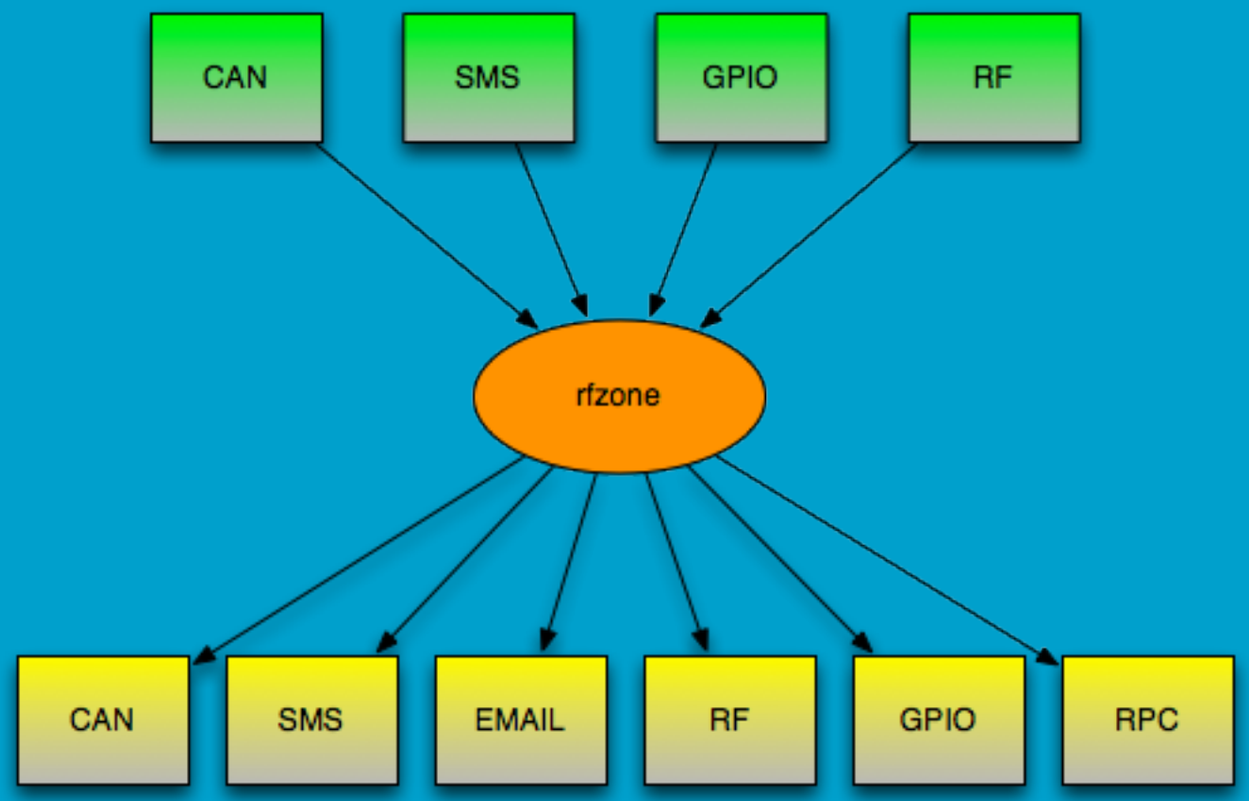
EXODM-RPC:(sms,gprs,none)*:<base64>
EXODM-RET:<base64>

FEUERLABS

Application - Demo



rfZone



FEUERLABS

rfzone INPUT processing

protocol	board	pin	interrupt	polarity	
gpio	cpu	25	falling	FALSE	
gpio	piface	0	both	TRUE	
gpio	piface	1	both	TRUE	
gpio	piface	2	both	TRUE	

COBID	CHANNEL	TYPE	VALUE
20001	100	digital	$value
20001	101	digital	$value
20001	102	digital	$value

protocol	model	data
archtec	codesw	16#E0D
archtec	codesw	16#60D
everflourish		3735439
everflourish		3735455

COBID	CHANNEL	TYPE	VALUE
20003	1	digital	0
20003	1	digital	1
20004	1	digital	0
20004	1	digital	1

protocol	reg_exp	anumber	alphabet
sms	"^[\\s]*off[\\s]*$"	070012345	default
sms	"^[\\s]*on[\\s]*$"	070012345	default

COBID	CHANNEL	TYPE	VALUE
20004	1	digital	0
20004	1	digital	1

rfzone OUTPUT

COBID	CHANNEL	protocol
20001	101	risingsun
20001	102	risingsun
20001	103	risingsun

type	unit	chan	style	polarity	
digital	1	1	springback	FALSE	
digital	1	2	springback	TRUE	
digital	1	3	springback	TRUE	
				TRUE	

COBID	CHANNEL	protocol
20001	100	sms
20003	14	email

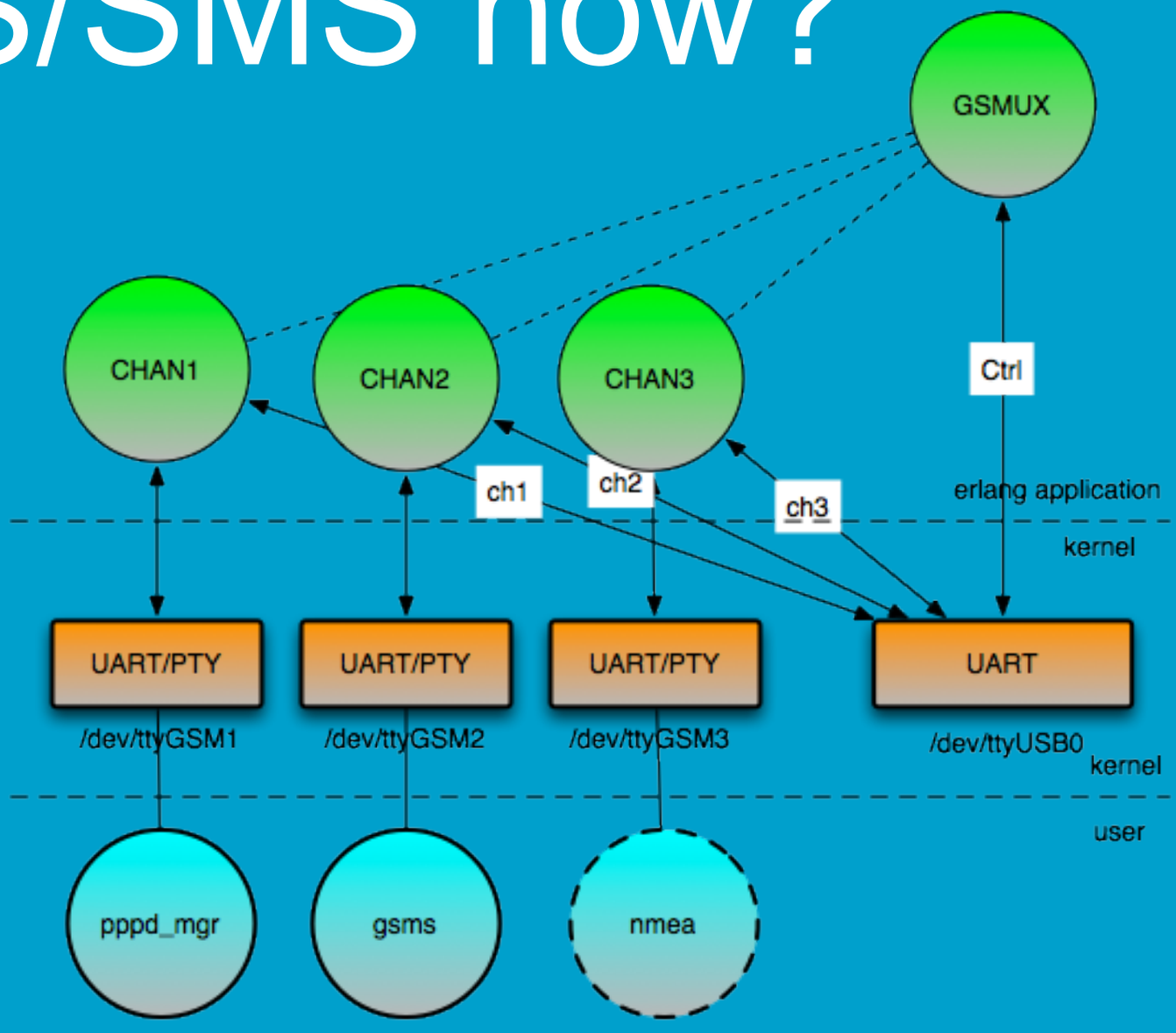
type	inhibit	addr	body		
digital	6000	7023456	“Hello”		
digital	60000	tony@rogvall.se	“alert”		

FEUERLABS

GSM/GPRS modem

- Tricky to get GSM modems up
- Raspberry pi modem based on SIM900
- Restart on several levels.
- Power save mode.
- GPIO interrupt on RI. CALL/SMS
- Power up halted raspberry PI on RI / Time

GPRS/SMS how?


GSMUX
CHAN1
CHAN2
CHAN3
Ctrl
ch1
ch2
ch3
erlang application
kernel
UART/PTY
UART/PTY
UART/PTY
UART
/dev/ttyGSM1
/dev/ttyGSM2
/dev/ttyGSM3
/dev/ttyUSB0
kernel
user
pppd_mgr
gsms
nmea

FEUERLABS

Thanks

Questions?