FEUERLABS

Erlang End-To-End

Building and Managing Connected Devices

Erlang Factory, San Francisco 2012

Ulf Wiger
ulf@feuerlabs.com
http://www.feuerlabs.com
@uwiger

Magnus Feuer
magnus@feuerlabs.com
http://www.feuerlabs.com

Outline

- Connected Devices
- Feuerlabs Exosense
- Exosense and Open Source
- Case study (Magnus)

FEUERLABS

Hans Vestberg, CEO at Ericsson:

“50 billion connected devices by 2020”

“Any device that can benefit from being connected, will be connected.”

FEUERLABS

Hans Vestberg, CEO at Ericsson:

“50 billion connected devices by 2020”

(IBM:“300 billion devices”
Jim Zemlin: “1 trillion devices”)

“Any device that can benefit from being connected, will be connected.”

A Brave New World

- Perfect opportunities for the avid hacker
- Small devices w/ Linux are a commodity
- Go forth and invent new applications!
- But...

New challenges

- A different kind of complexity
- New failure modes (e.g. spotty connectivity)
- Remote configuration, updates, debugging
- High barrier of entry

New challenges

?

- A different kind of complexity
- New failure modes (e.g. spotty connectivity)
- Remote configuration, updates, debugging
- High barrier of entry

Device Mgmt challenges

- Heterogeneous devices
  (different versions, board types, capabilities...)
- "Northbound interfaces" (NOCs, Business logic)
- Upgrade schedules
- Bandwidth metering
- Persistence, scalability, redundancy, fairness

FEUERLABS

Feuerlabs Exosense
an End-to-end solution

The community pitch

Feuerlabs

Tony Rogvall

Magnus Feuer

Marcus Taylor

Ulf Wiger

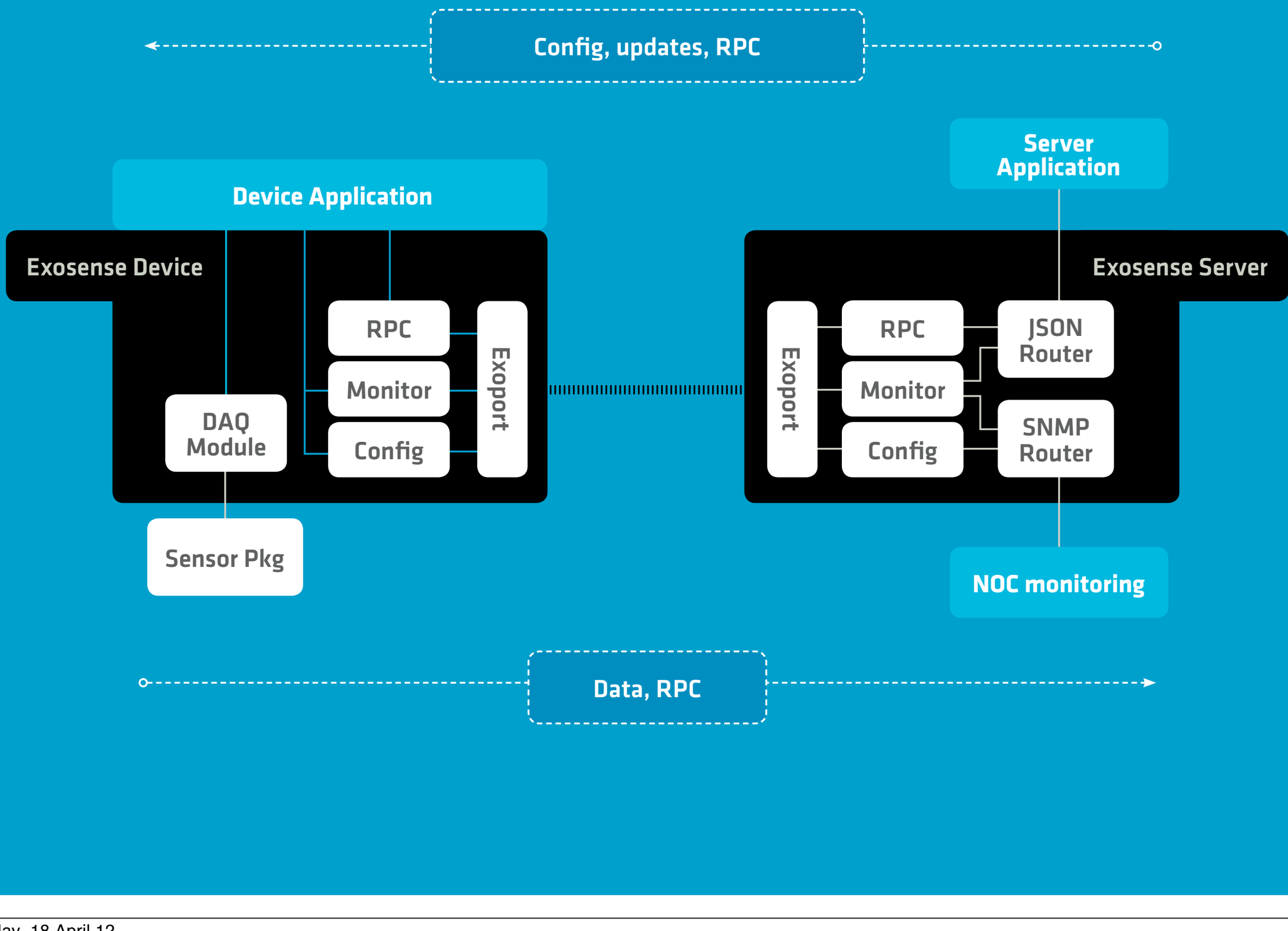
MOTIVO

Erlang

- Small, but decades of accumulated telecoms/telematics experience

Feuerlabs Exosense

FEUERLABS



FEUERLABS

Compared to conventional approaches...

- Lower entry threshold
- Shorter time to market
- Higher quality
- Better scalability

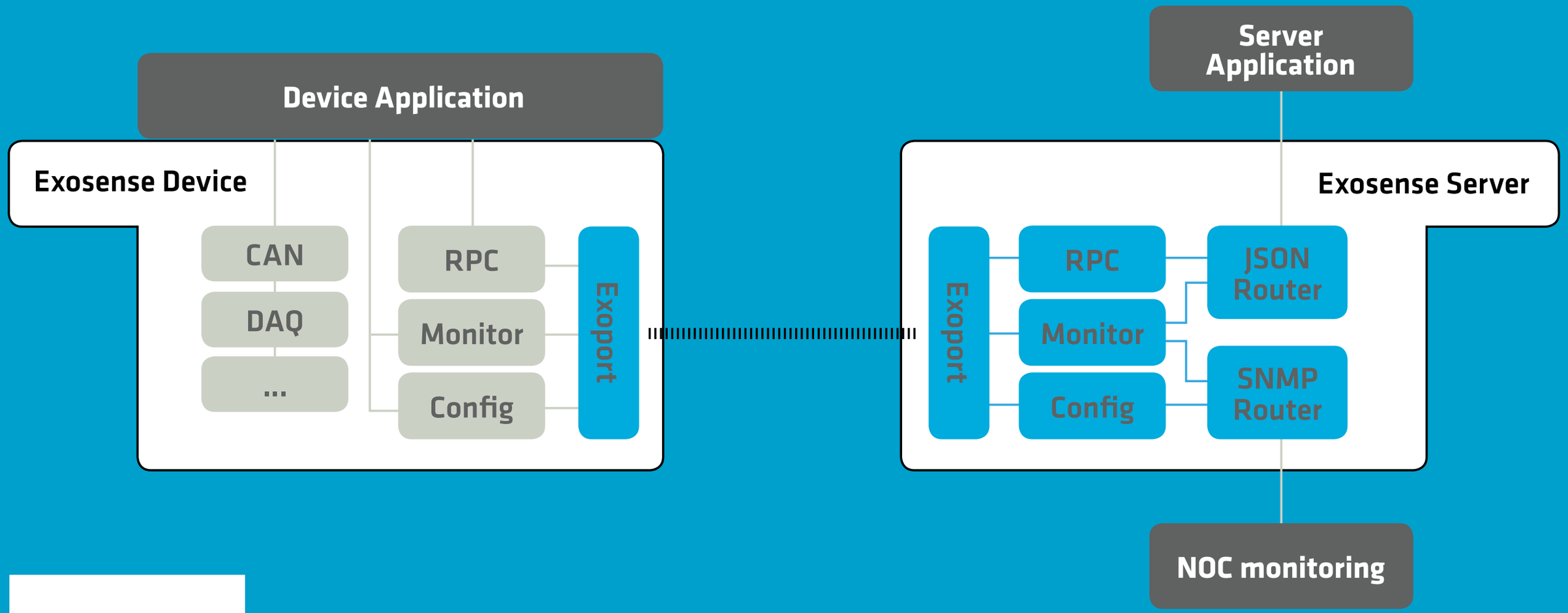
FEUERLABS

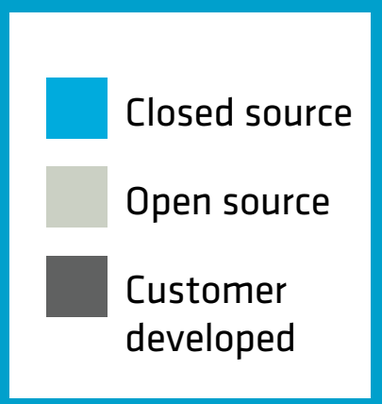
Compared to conventional approaches...

- Lower entry threshold
- Shorter time to market
- Higher quality
- Better scalability
- Better fit to multicore
- Easier to debug
- Finer granularity of updates
- Better abstractions for connectivity

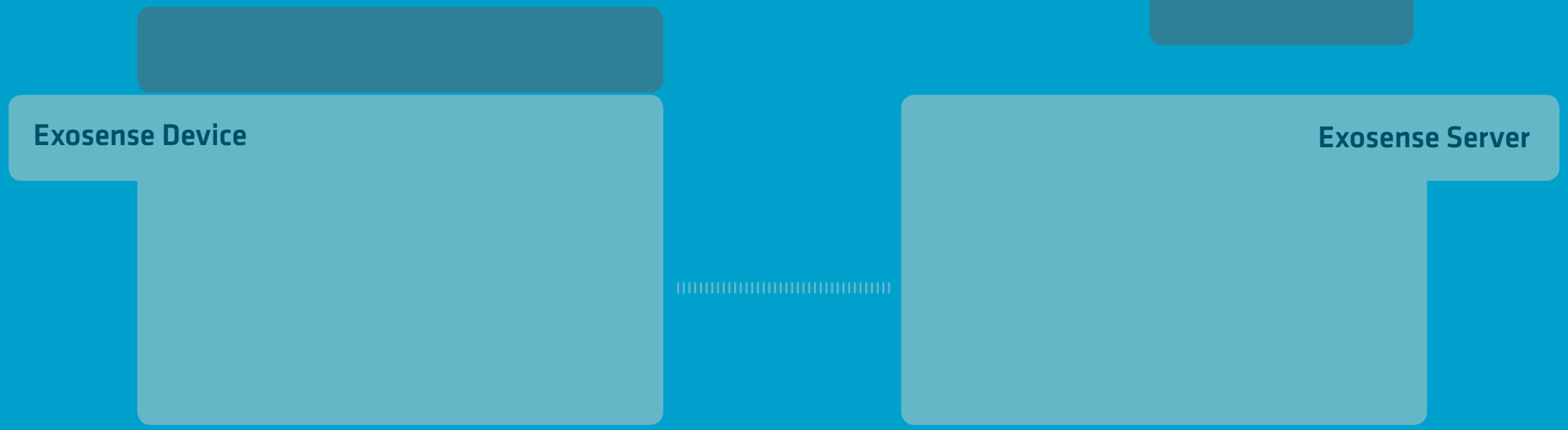
Exosense and Open Source

FEUERLABS





FEUERLABS



- **Exosense Device - Open Source**
- Free download
- Ready to use
- Add application
- Telematics connection optional

- **Exosense Server - Closed Source**
- License for self hosting
- Feuerlabs hosting available
- Add application

Device platform: Community effort

- Several different players contribute
  - E.g. ESL, Tail-f, Travelping, ...
- Lower the threshold for entrepreneurs
- A cottage industry for “the Internet of Things”

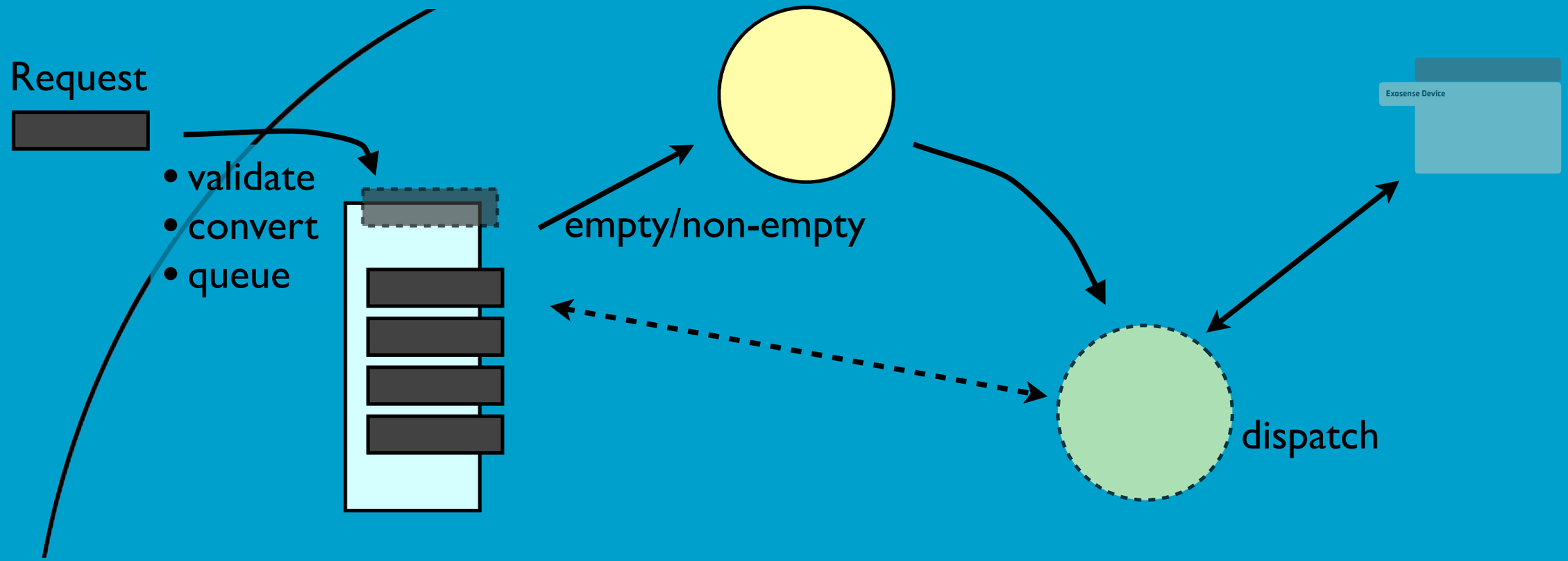
Data models

```
grouping std-callback {
    description "Elements included in...";

    leaf transaction-id {
        type transaction-id;
        description "The transaction ...";
    }
```

- Yang (RFC 6020) data models
- Automatic generation of JSON API + docs
- Automatic verification of JSON requests
- http://github.com/tonyrog/yang

Exosense backend



- Persistent lightweight queueing (using kvdb)
  - both to and from device
- Event-triggered dispatch (using gproc)
- Limited retry

Exosense Config DB

```
Eshell V5.9  (abort with ^G)
1> conf_db:open().
starting kvdb_conf, [{file,"/tmp/kvdb.db"},
                     {backend,sqlite},
                     {tables,[data]},
                     {encoding,{raw,term,raw}}]
{ok,<0.35.0>}
2> conf_db:diff_test(4711).
write: key=<<"devices*x00001267*config*candidate-a*name">>, value=<<"A">>
write: key=<<"devices*x00001267*config*candidate-a*items*item[0]">>, value=<<"0">>
write: key=<<"devices*x00001267*config*candidate-a*items*item[1]">>, value=<<"1">>
write: key=<<"devices*x00001267*config*candidate-a*items*item[2]">>, value=<<"2">>
write: key=<<"devices*x00001267*config*candidate-a*items*item[4]">>, value=<<"4">>
write: key=<<"devices*x00001267*config*candidate-b*name">>, value=<<"B">>
write: key=<<"devices*x00001267*config*candidate-b*items*item[0]">>, value=<<"0">>
write: key=<<"devices*x00001267*config*candidate-b*items*item[1]">>, value=<<"1">>
write: key=<<"devices*x00001267*config*candidate-b*items*item[3]">>, value=<<"3">>
write: key=<<"devices*x00001267*config*candidate-b*items*item[4]">>, value=<<"4">>
[{update,<<"name">>,<<"B">>},
 {add,{<<"items*item[3]">>,[],<<"3">>}},
 {delete,<<"items*item[2]">>}]
```

- Single-image (sqlite3)
  Yang-style config database
- Diff between old and new configurations

Device programming example: UART

```
open(DeviceName) ->
    case file:read_file_info(DeviceName) of
        {ok,Info} ->
            case Info#file_info.type of
                device ->
                    case uart:open(DeviceName,
                                   [{ibaud,4800},{obaud,4800},
                                    {active,once},{buffer,1024},
                                    {packet,line}]) of
                        {ok,Port} ->
                            {ok,device,Port};
```

- Serial-line library with gen_tcp-like options
- Runs on Linux, MacOS, Windows, ...
- http://github.com/tonyrog/uart

Device programming example: UART

```
handle_info({uart,U,Line}, State) when State#state.port =:= U ->
    uart:setopt(U, active, once),
    nmea_line(Line, {undefined,undefined}, State);
```

```
nmea_line(Line,{Date0,Time0},State) when is_list(Line) ->
    Verify = verify_checksum(Line),
    if Verify =:= valid; Verify =:= no_checksum ->
            case re:split(Line, ",", [{return,list}]) of
                ["$GPGGA",UTC,Lat,LatNS,Long,LongEW,Quality,_SatCount |
                 _Various ] ->
                    Valid = Quality =/= "0",
                    set_state(Valid,
                              [{lat, string_lat_dec(Lat,LatNS)},
                               {long, string_long_dec(Long,LongEW)},
                               {time, string_time(UTC,Time0)}], State);
```

- Flow control
- Pattern-matching power

Community contribs: rebar extensions

FEUERLABS

rebar.config.script:

```
%% -*- erlang -*-
{ok, Opts} = file:consult(
               filename:join(
                 filename:dirname(SCRIPT), "rebar.config")).
case os:getenv("REBAR_DEPS") of
    false -> Opts;
    [] -> Opts;
    Dir ->
        lists:keystore(deps_dir, 1, Opts, {deps_dir, Dir})
end.
```

- A simple way to “patch” the rebar environment
- For faster turnaround during development

Community contribs: bypass RelTool

FEUERLABS

```
$ make
$ make dev
$ cd /tmp/exodm
$ $EXODM/devrun -name n1
```

- ‘setup’ can now read a reltool config
  http://github.com/esl/setup
- Builds boot scripts based on code path
  - rather than copy to separate dir, like RelTool
- Faster builds, easier patching

Summary so far

- Exosense Device Platform
  - Lower the barrier of entry for connected device development
  - Community effort, Open Source
- Exosense Device Management Service
  - Telematics, OTA config and upgrade
  - Low instep for rapid prototyping

FEUERLABS

Case study

Magnus Feuer
CTO, Feuerlabs

http://getaround.com

Feuerlabs Exosense

FEUERLABS