

Platform and language independent service life cycle management for device centric SOAs

Christian Fabian, Elmar Zeeb, Frank Golasowski,
Dirk Timmermann



Outline

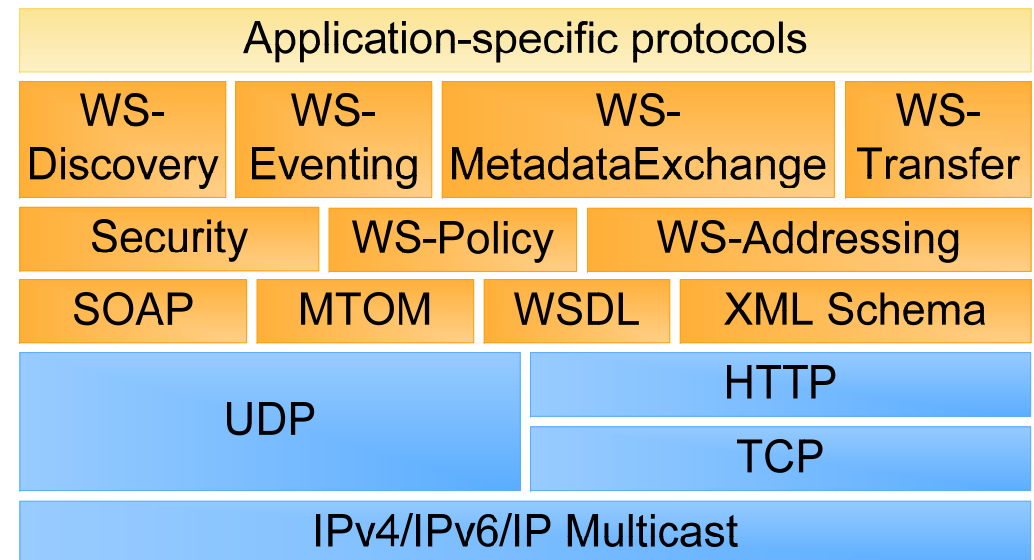
- 1. Motivation and Basics**
- 2. Lightweight life cycle management**
- 3. Installing process**
- 4. Example of Use**
- 5. Conclusion / Outlook**





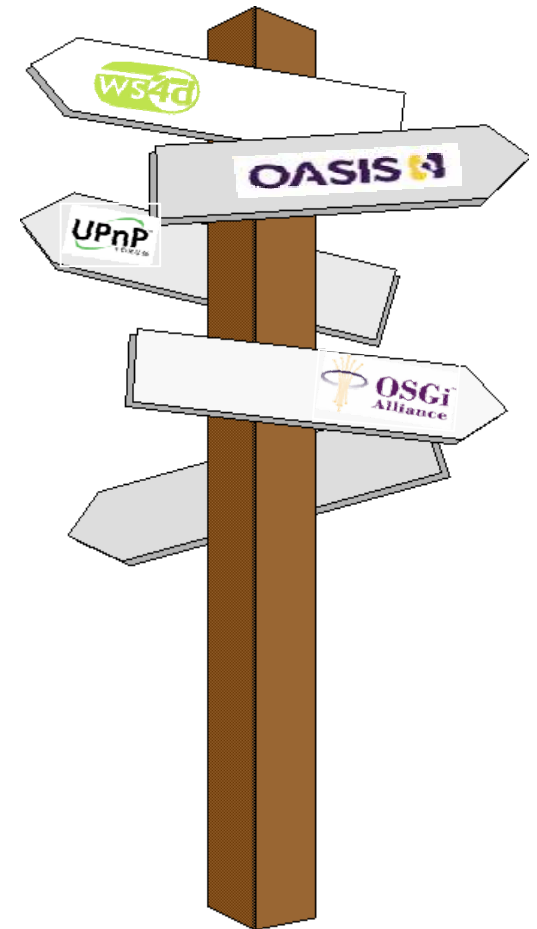
Devices Profile for Web Services – in detail

- message exchange (SOAP)
- Metadata device and service description
(WS-MetadataExchange)
- handling large data as attachment (MTOM)
- Device description (WSDL)
- Data format and data type
(XML Schema)
- Publish/Subscribe mechanism
(WS-Eventing)

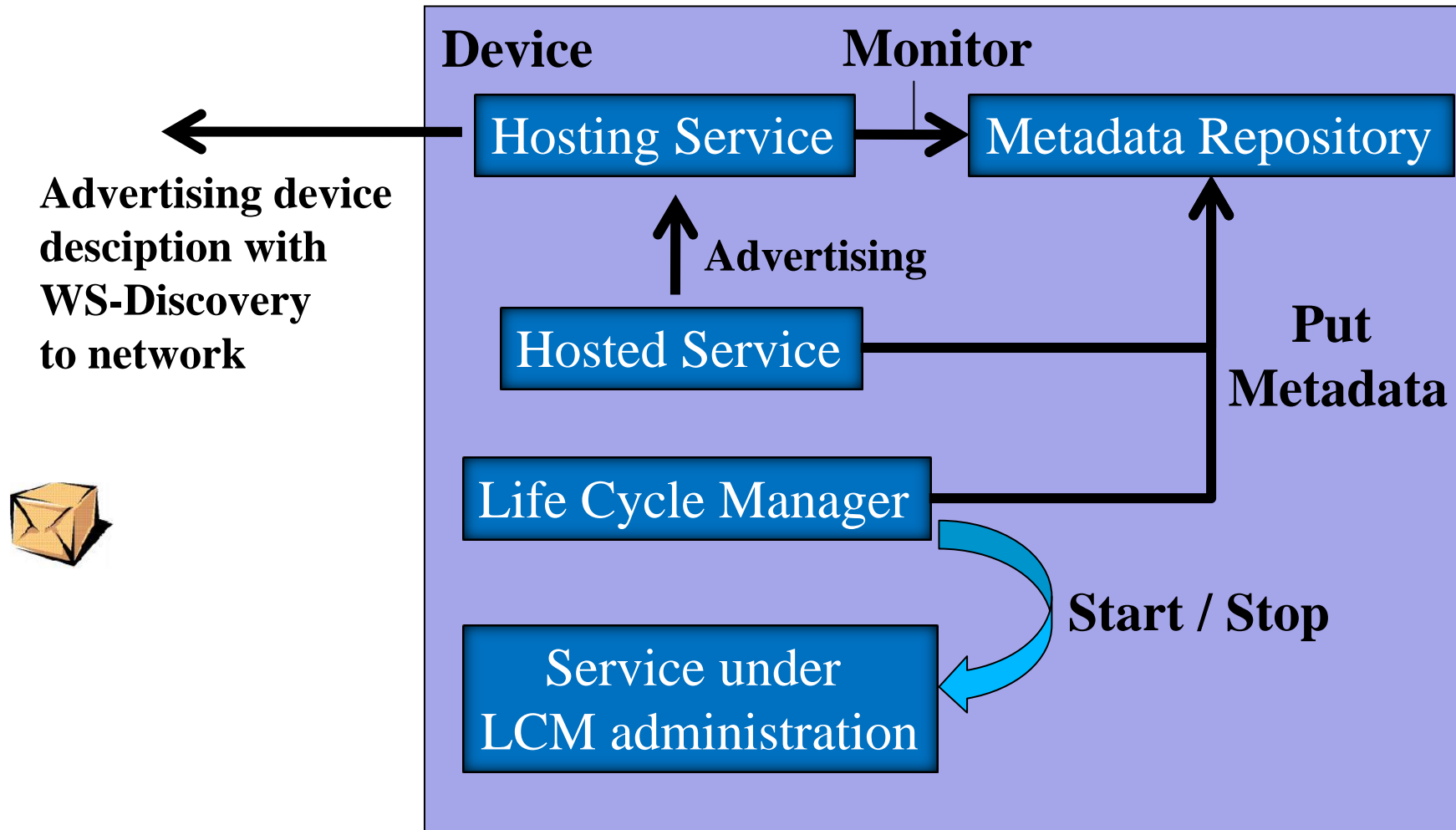


Motivation

- Using the DPWS toolkit from the University of Rostock
 - Developed for embedded systems
 - available under GPL/LGPL license
 - Ad-hoc device discovery
 - WS-Eventing
 - Based on well known protocols and WS-specifications
 - Working with Windows Vista / 7 / .net Micro



DPWS device



Installing a new service

- ServiceID is used to distinguish several installed services
- Bundle content:
 - Service: executable file
 - WSDL: interface description of the service
 - XML: metadata description
 - Scripts (resolve-, start-, stop- and isAlife script)
 - Further files (e.g. Library, pictures, ...)



LUA for independent control of services

- Broker to translate between the life cycle manager and services
- small sized interpreter (120kB)
- Lua to start/stop services language and platform independent
- Read environment variable via „os.getenv“
- Start service via „os.execute“

```
1 | -- test the operating system
2 | if os.getenv("OS") == "Windows NT" then
3 |     print("Windows") ;
4 |     if Resolve() == 0 then
5 |         -- start the service
6 |         os.execute(". /simpleExample.exe ")
7 |     else
8 |         print("Resolve not successful!")
9 |         return 0
10|     end
11| else
12|     print("OS is not Windows !")
13|     return 0
14| end
15| return 1
```

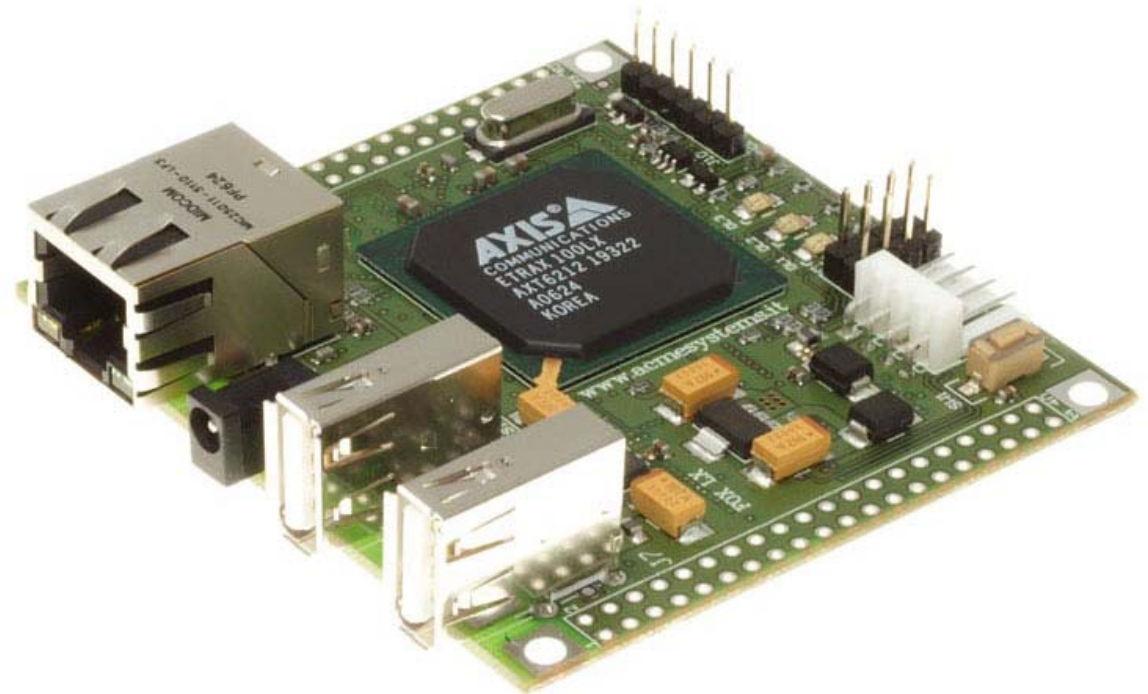

Example of Use on the FOX-Board LX832

SOFTWARE features

Linux Kernel:	Up to version 2.6.19
Server:	HTTP (Web), FTP, SSH, Telnet
Language:	C, C++, PHP, Python, etc.

HARDWARE features

Size:	66 x 72 mm (2.6 x 2.8 inches)
CPU:	32bit RISC, 100 Mhz
Memory:	8MB FLASH, 32 MB RAM
Power:	5V 280 mA (1W)
Ports:	1 Ethernet (10/100 Mb/s) 2 USB 1.1 1 serial console port



Conclusion

- ✓ Service-based life cycle manager for an implementation of DPWS
- ✓ Manage the services of a device during runtime through remote reconfiguration
- ✓ Platform and language independent by using a script-based interaction
- ✓ Transfer and run the concept on an embedded system
- ✓ able to modify a device during runtime

Outlook

- Support more platforms
- Implementing security functionality
- possibility to move a service between several devices
- Reducing memory consumption



Thank you for your attention

