FlockLab: A Feature-Rich Testbed for Wireless Sensor Networks

Roman Lim, Christoph Walser, Jan Beutel

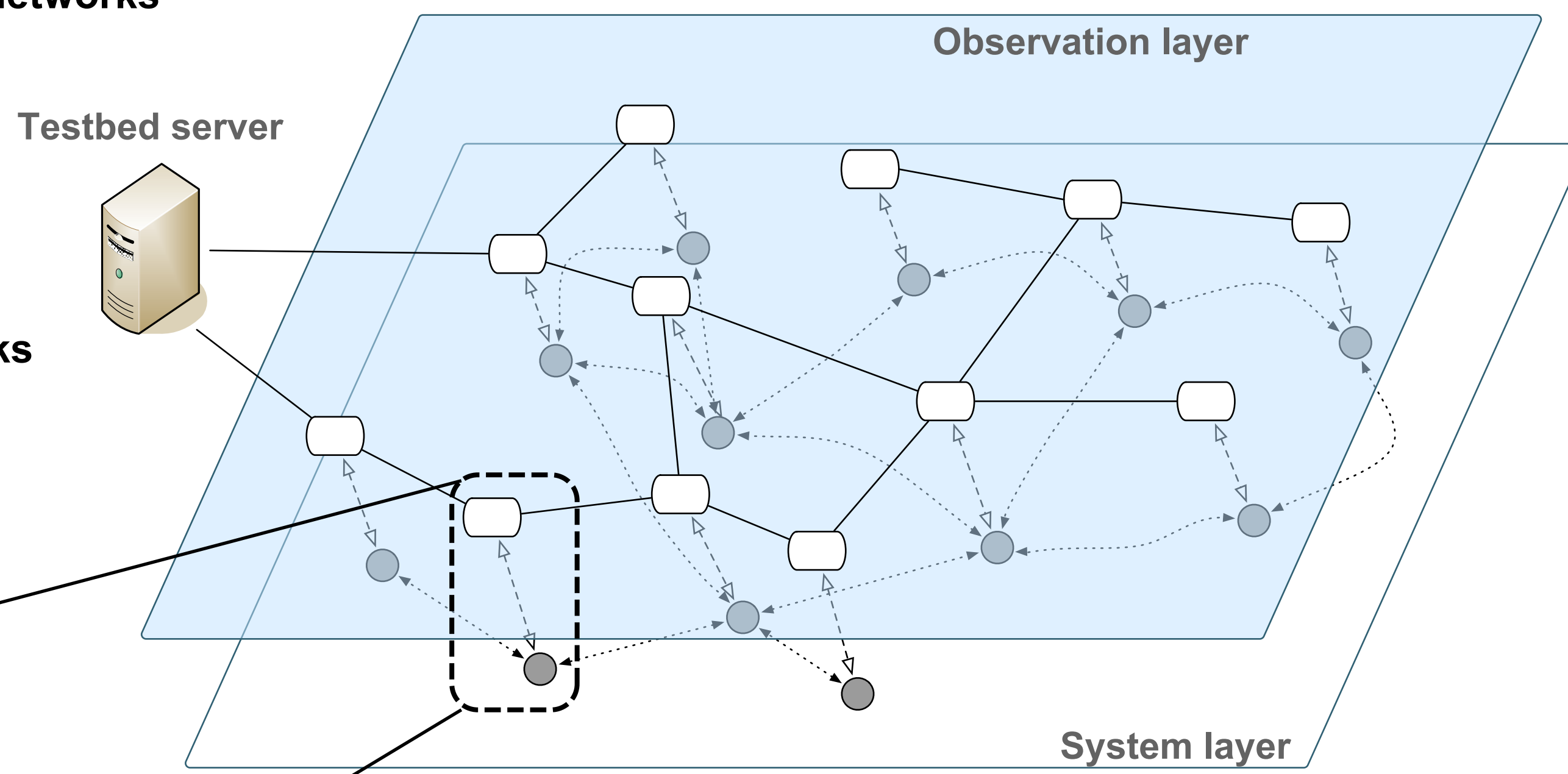Computer Engineering and Networks Lab, Swiss Federal Institute of Technology (ETH) Zurich

Testbed Architecture

Writing software for wireless sensor networks is difficult

- Non-deterministic environment
- working close to resource limit makes systems extremely fragile

Testbeds for Wireless Sensor Networks

- Increase observability
- Enable testing on real hardware
- Help validating simulation results

Testbed server

Observation layer

System layer

Observer · Target · Application

Database · GPIO Monitor · GPIO Setting · Logging · Powerprofiling · Software instrumentation

The observation layer provides means to reprogram target nodes, monitor running programs, and interact with sensor nodes

The system layer represents the actual sensor network, consisting of individual sensor nodes, wireless links and environment.

FlockLab Highlights

Distributed high resolution power profiling on every target

Easily extensible to other target platforms

Monitoring of program state through GPIO pins

Adjustable voltage

Tightly coupled observer-target architecture enables accurate timestamping and long term logging

Several node platforms and operating systems are supported:

Tmote Sky
Opal (collaboration with CSIRO)
IRIS (collaboration with IBM)
TinyNode184

TinyOS, Contiki, Moterunner

Observer hardware:

Embedded Linux Computer (Gumstix)

Tmote Sky · Opal · IRIS · TinyNode
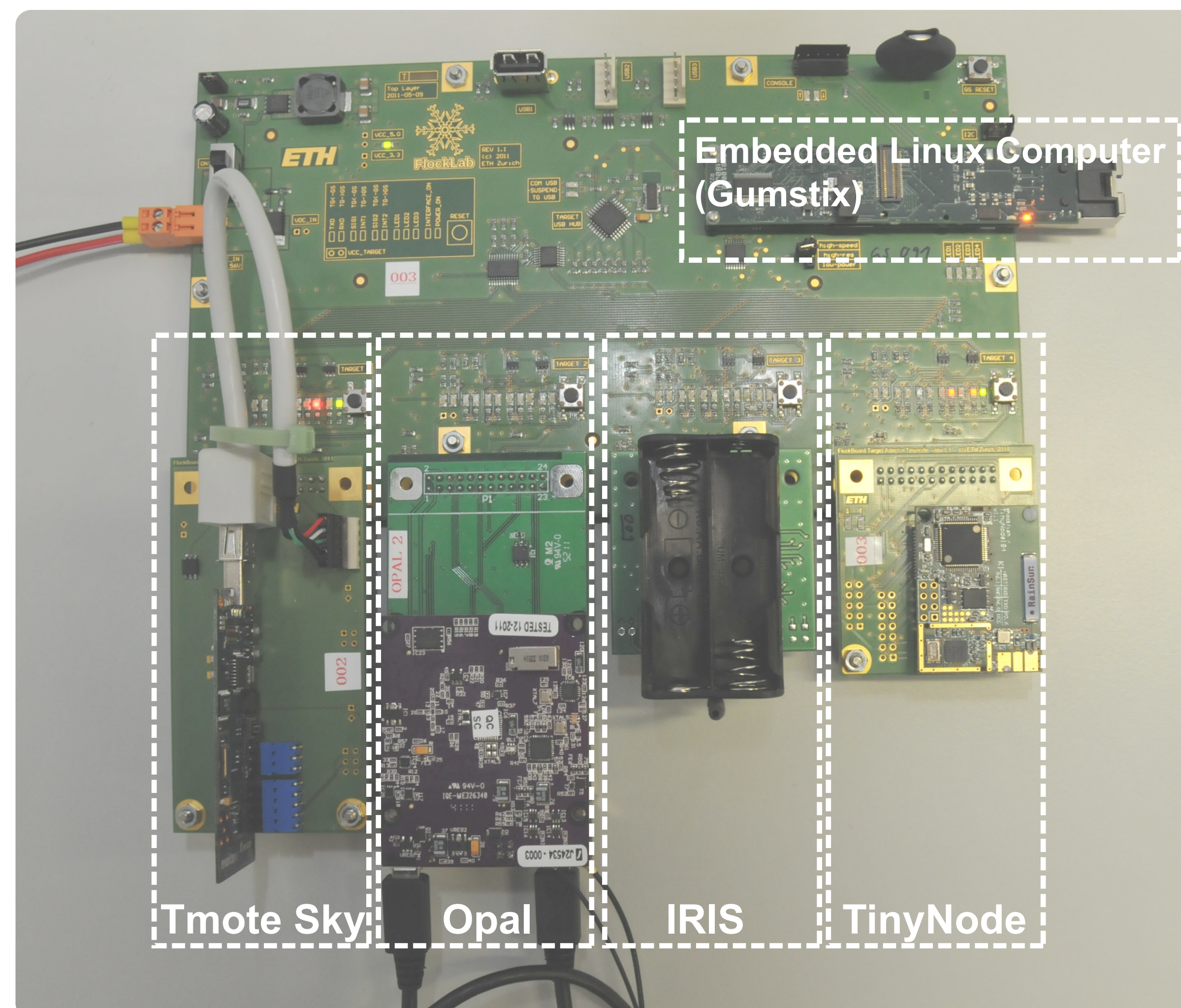
Testbed Installation

- 30 testbed nodes
- Ethernet/WLAN
- In- & Outdoor
- Public access

Indoor and..

.. outdoor locations

Example Test Data

Synchronized Power and State Traces from Distributed Nodes

Power trace

I Sender

I Receiver

State traces (GPIO)

Sender Radio TX

Sender CPU
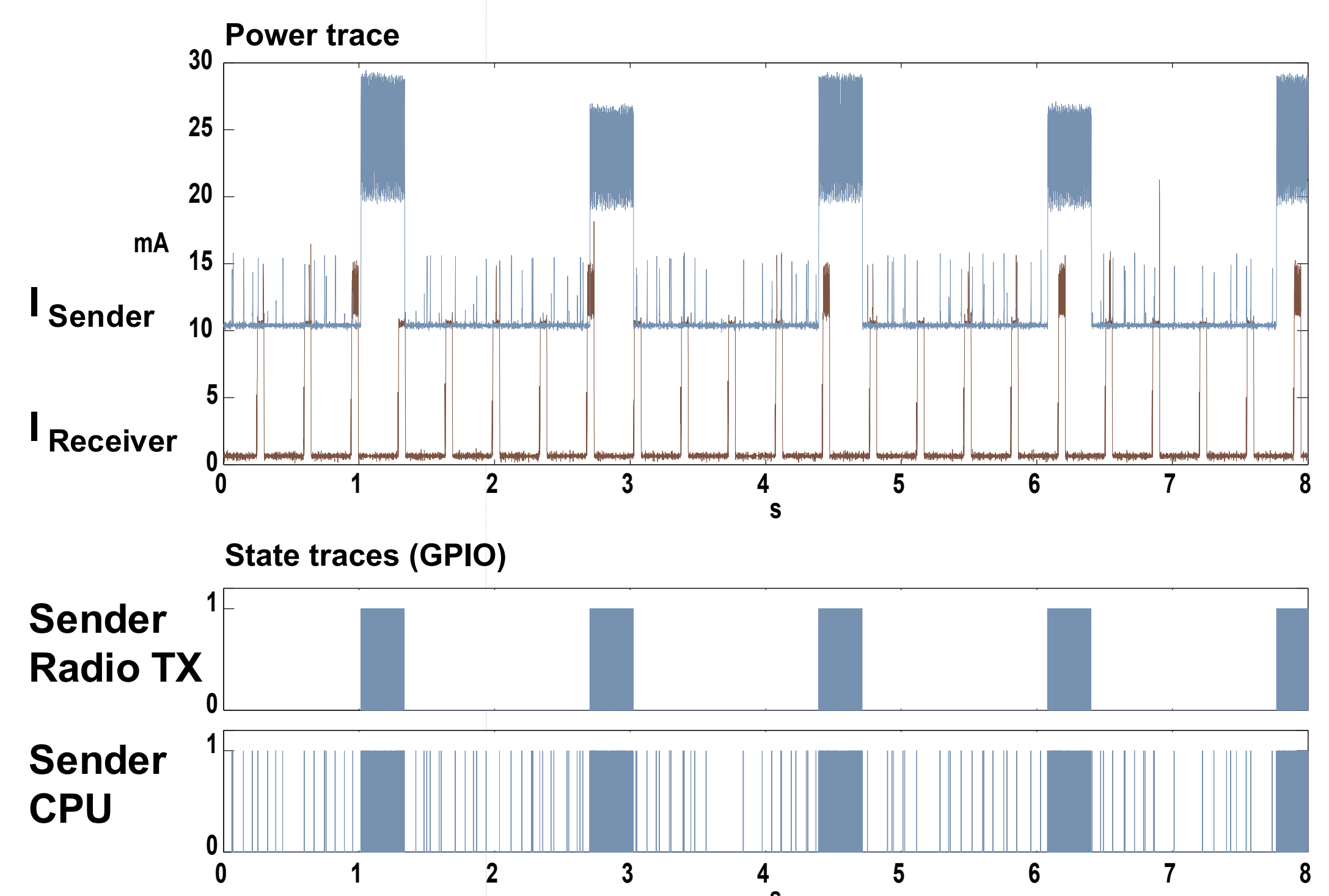
The receiver node periodically polls the radio channel (LPL) while the sender node sends packet burst to communicate with the receiver node. The CPU and transmission activities are exported using a GPIO pin. The power trace of the sender nicely corresponds to the exported states.

User Interface

Web-based user interface
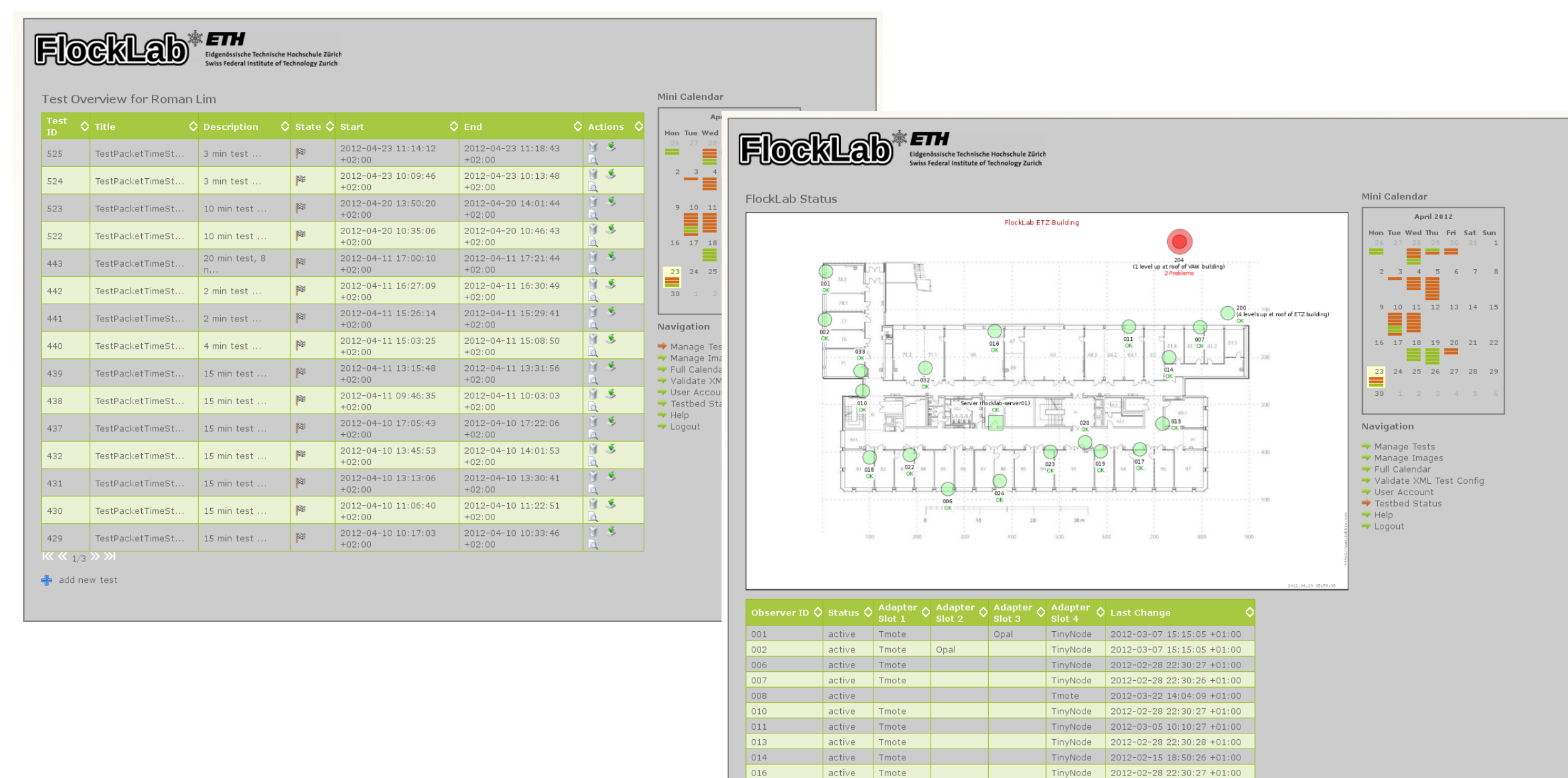
Test configuration is done using XML

```
...
<!-- Target configuration -->
<targetConf>
        <obsIds>033 010 018 022 006 008 029 031</obsIds>
        <voltage>3.3</voltage>
    <dbImageId>232</dbImageId>
</targetConf>

<!-- Power Profiling Service configuration -->
<powerprofConf>
    <obsIds>033 018 022 006 029 031 008</obsIds>
    <profConf>
            <durationMillisecs>180000</durationMillisecs>
            <relativeTime>
                <offsetSecs>0</offsetSecs>
                <offsetMicrosecs>0</offsetMicrosecs>
            </relativeTime>
        </profConf>
</powerprofConf>
...
```

Scripting support:
Test generation can easily be included into a Makefile build environment.

Test results, images and configurations are also available as mountable web directory (WebDAV)