Natur 4.0
Sensing Biodiversity

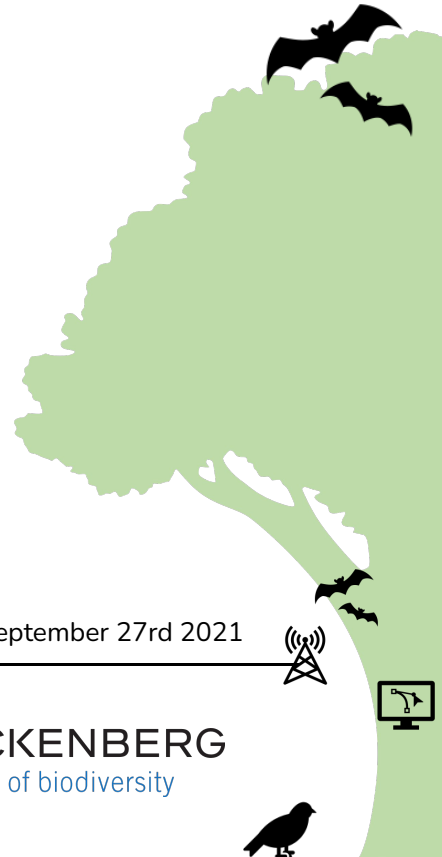
LOEWE
Exzellente Forschung für Hessens Zukunft

tRackIT OS: Open-source Software for Reliable VHF Wildlife Tracking

Jonas Höchst
Jannis Gottwald
Patrick Lampe
Julian Zobel
Thomas Nauss
Ralf Steinmetz
Bernd Freisleben

GI INFORMATIK 2021
CS4BioDiversity

September 27rd 2021

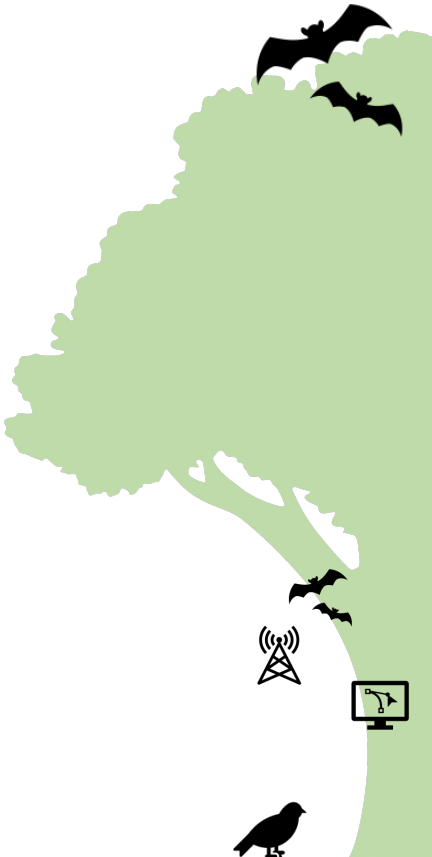


Philipps Universität Marburg
TECHNISCHE UNIVERSITÄT DARMSTADT
JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN
SENCKENBERG world of biodiversity

tRackIT OS: Introduction

State of the art & development goals

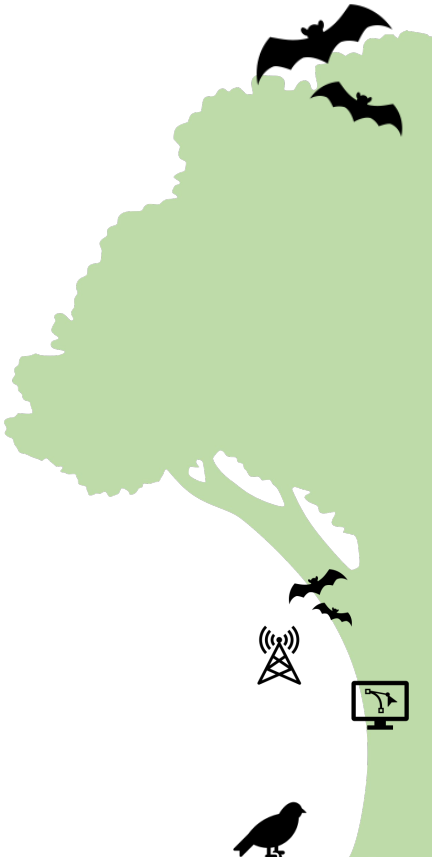
- a novel approach for **automated signal detection of VHF radio tracking tags**,
- means to provide **reliable operation** of tRackIT stations **under harsh conditions**,
- efficient **live data transmission** for monitoring data and detected signals,
- a novel **web-based user interface** for intuitive configuration of tRackIT stations,
- a **comparative evaluation** of tRackIT OS compared to the state-of-the-art.



tRackIT OS: Requirements

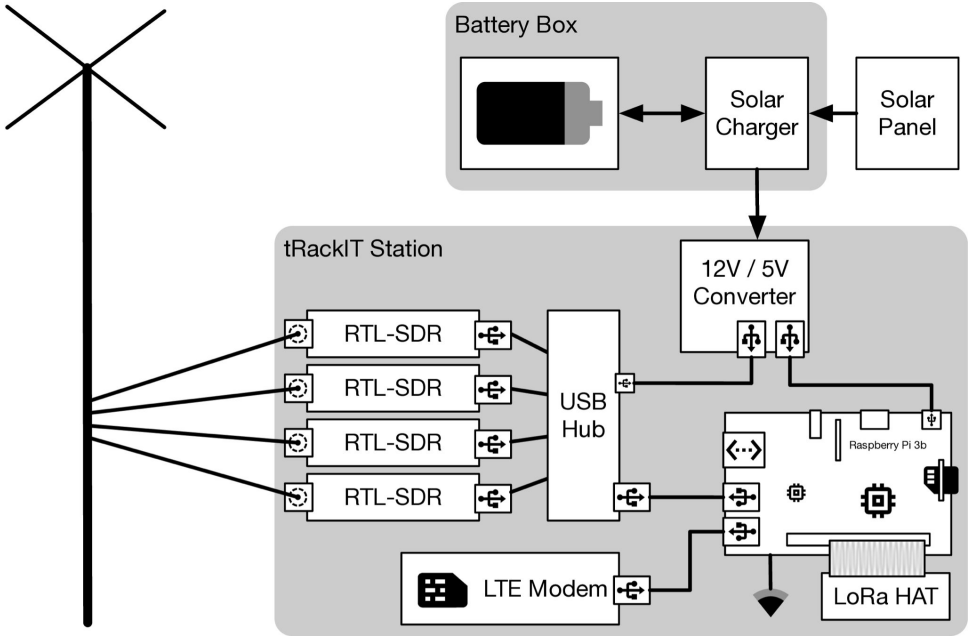
Substituting manual radiotelemetry

1. **Low entry barrier**
   - hardware & software availability,
   - convenient data processing and analysis access, easy to use, and inexpensive.
2. **Reliability**
   - reliable signal detection, minimized amount of interference,
   - automatic failure detection and handling.
3. **Data availability**
   - short delays between data recording and transmission,
   - allow live system data monitoring.



tRackIT OS: Hardware

Commodity-off-the-shelf hardware

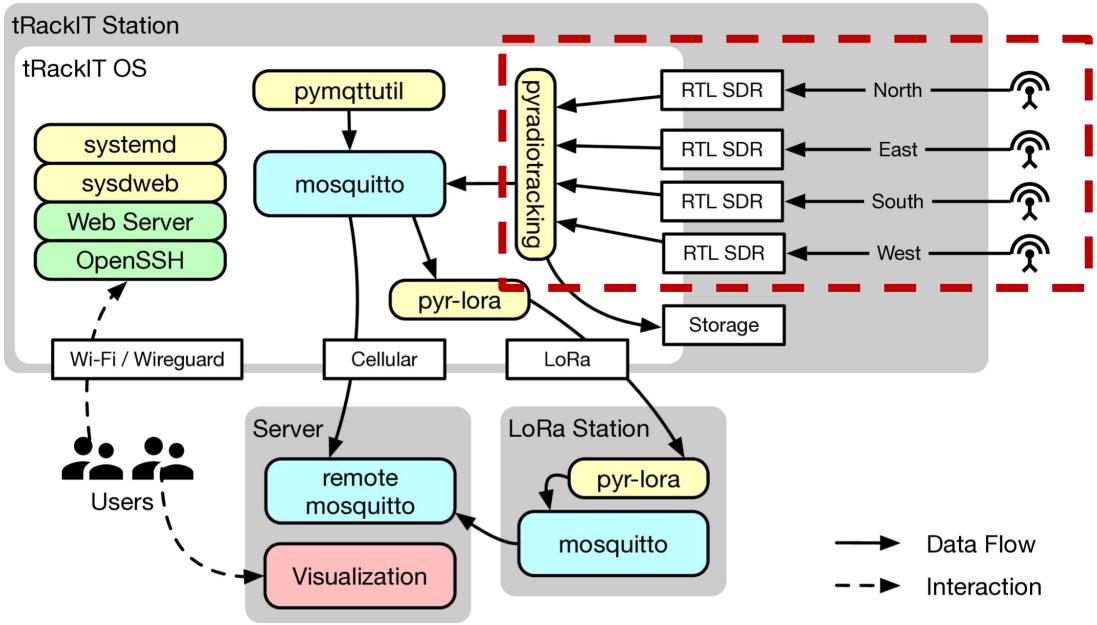


Component	Price
RPi 3 Model B	35€
4x Nooelec SDR	4 * 35€
active USB Hub	10€
Power Supply	10€
Huawei LTE*	50€
LoRa HAT*	35€
LoRa Concentrator*	110€
	200€+

Fig. 1: The hardware components of a tRackIT station.

tRackIT OS: Software Components

GNU/Linux meets IoT components



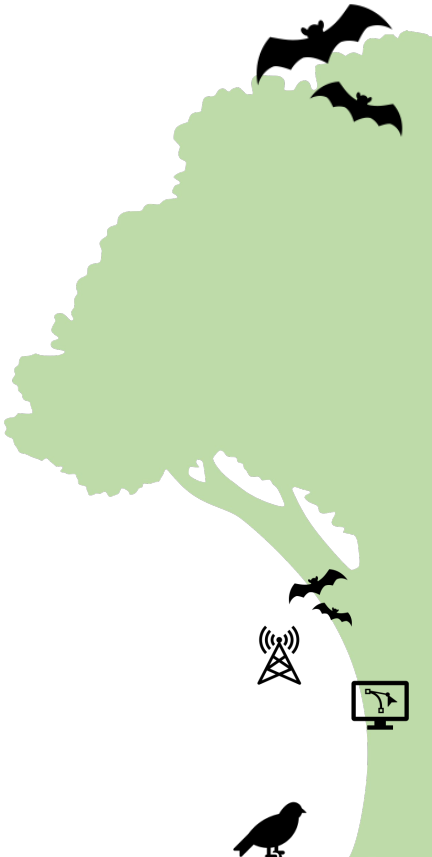


Fig. 2: Overview of the main software components of a tRackIT OS distribution.

tRackIT OS: pyradiotracking

Read antennas and find matching signals

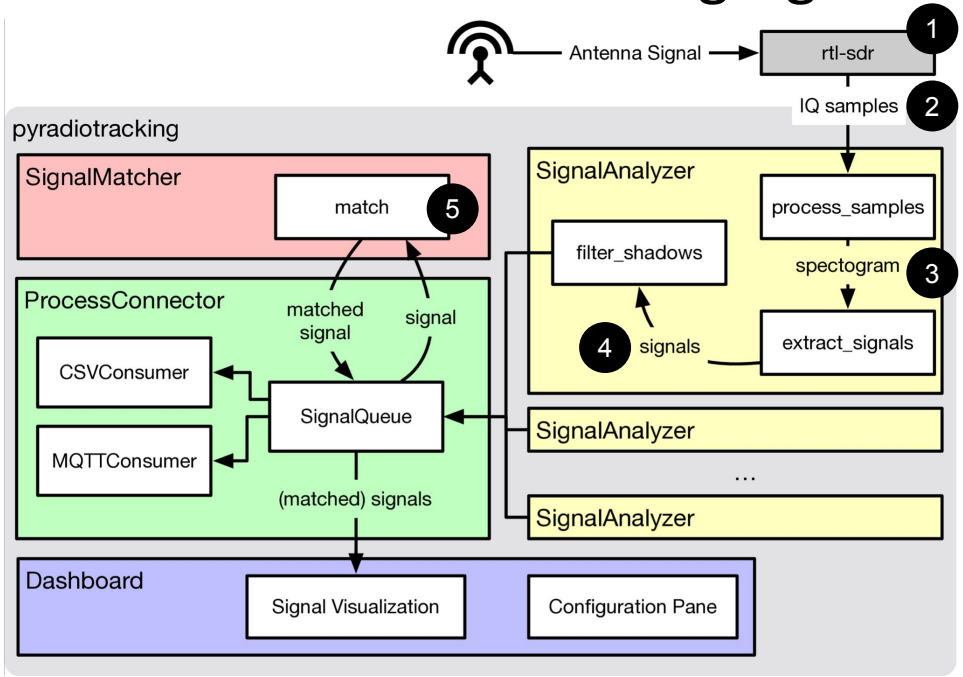
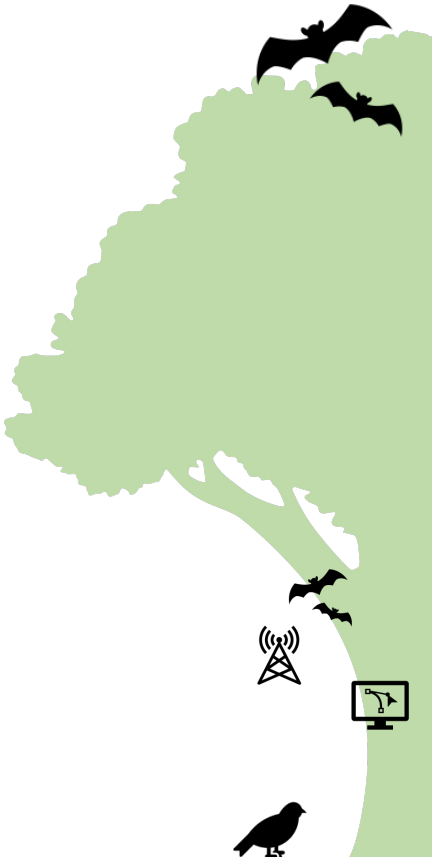


Fig. 3: Signal analysis stages implemented in pyradiotracking.



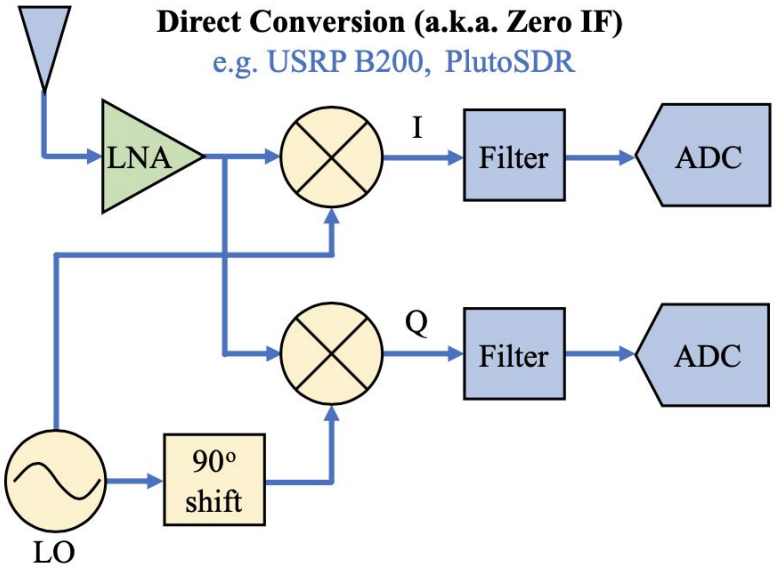
```
center_freq
sample_rate
gain
```

tRackIT OS: Signal Detection (1)

Receiver architecture of software-defined radios

A sine wave of a fixed frequency with variable phase and amplitude can be synthesized by adding a cos() and sin() wave scaled by the two components I (in-phase) and Q (out-of phase).

Commodity SDRs, just like FM Radios, shift the signal down to baseband, by adding a center frequency to a signal (shifted by 90° for Q component).



Source: https://pysdr.org/content/sampling.html#receiver-architectures

tRackIT OS: Signal Detection (2)

IQ-samples visualized

gain
sdr_callback_length

The visualization of the raw IQ samples already shows a sample with I and Q components with high values, which appear as a rectangle.

In the experimen,t the sender was placed right next to the receiver, thus the signals where clipping in the chosen gain setting.

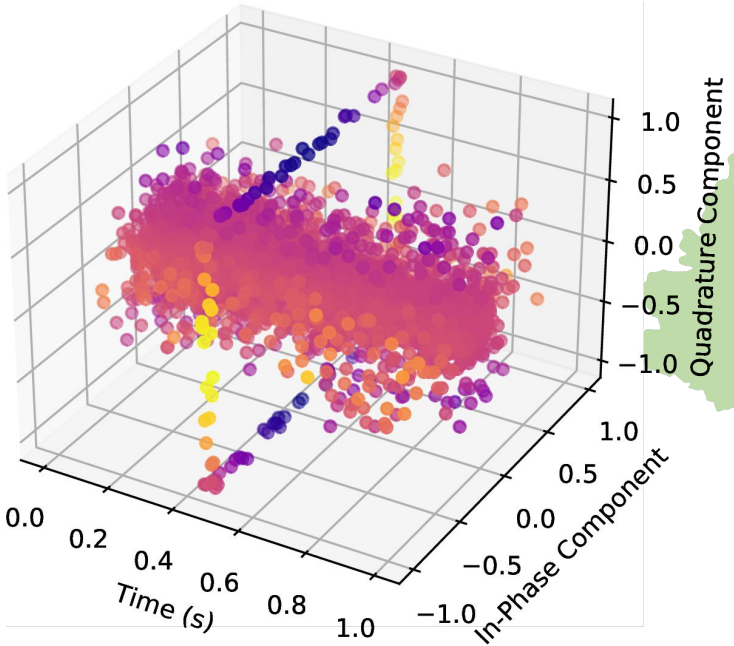


Fig. 4: IQ samples of one second, as received by RTL-SDR.

`fft_nperseg`
`fft_window`

tRackIT OS: Signal Detection (3)

Short-time Fourier Transform (STFT)

The discrete Fourier transform (DFT) converts a sequence of samples of a function into a same-length sequence of equally-spaced samples of the frequency.

A Short-time Fourier Transform (STFT) divides a longer time signal into shorter segments of equal length and computes the Fourier transform separately on each shorter segment.

For signal detection, the power spectral density is computed, i.e., relative power of a frequency at a time.

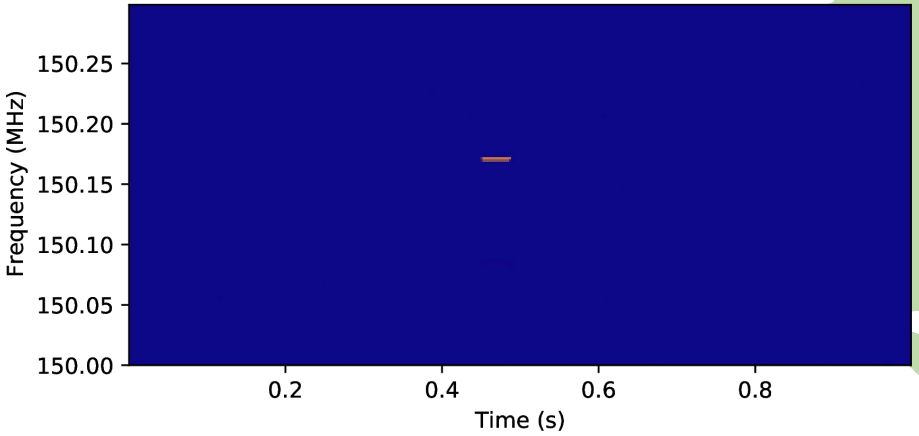


Fig. 5: Power spectral density (PSD) of samples computed via Short-time Fourier Transform (STFT).

tRackIT OS: Signal Detection (4)

Signal power sampling and signal detection

```
signal_threshold_dbw
snr_threshold_db
signal_min_duration_ms
```

All frequencies are scanned in a sampling interval equal to the configured minimum signal duration.

If the sample exceeds the power and SNR thresholds, values to the before and after are scanned until thresholds are undershot.

All detected signals are checked for shadow signals (equal start and stop time but lower power) and filtered accordingly.
The detected signal and multiples of its descriptive features are published.

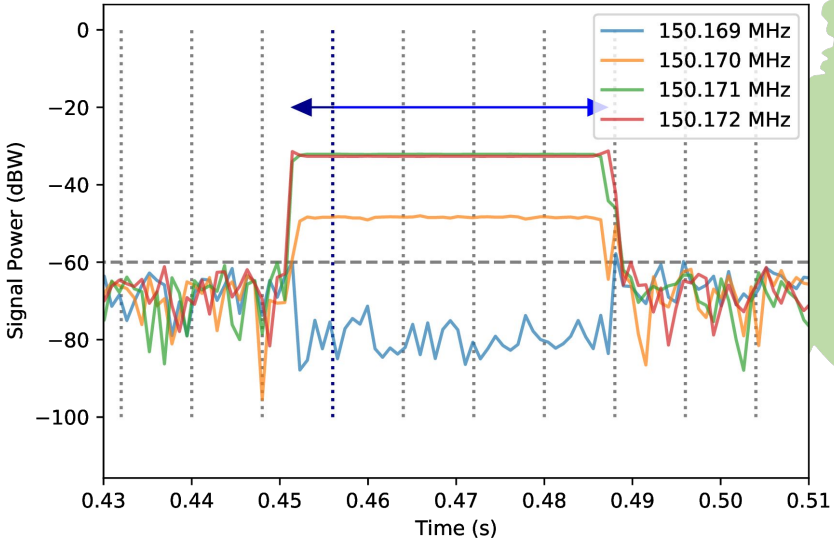
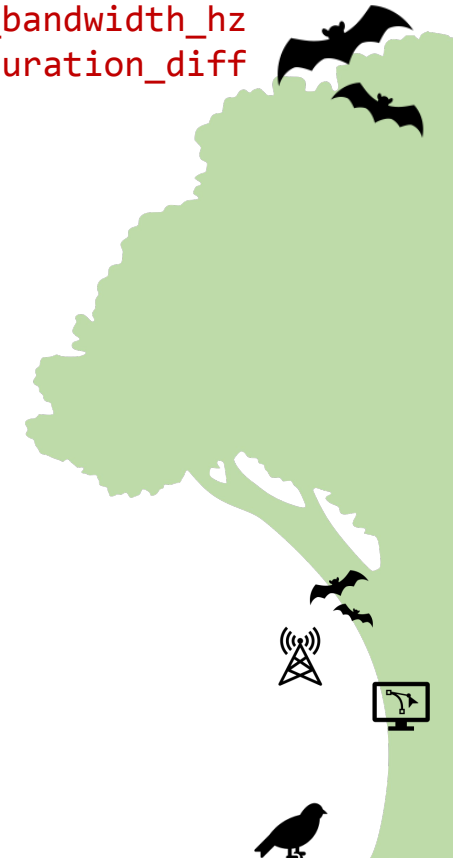


Fig. 6: Power spectral densities (PSDs) of selected frequencies, minimal signal power threshold, and signal power sampling points.

tRackIT OS: Signal Matching (5)

Read antennas and find matching signals

`matching_timeout_s`
`matching_time_diff_s`
`matching_bandwidth_hz`
`matching_duration_diff`



For each detected signal:

- For each matching group:
  - If matching.start > now + matching_timeout:
    - publish and remove matching group
  - If signal.frequency == matching.frequency +/- bandwidth
    and signal.start == matching.start +/- time_diff
    and signal.duration == matching +/- duration_diff:
    - add to matching group
  - Else:
    - create new matching group

Note: A matching group only holds one signal per SDR, signals of higher power are preferred.

sdr_timeout_s

tRackIT OS: Robustness in pyradiotracking

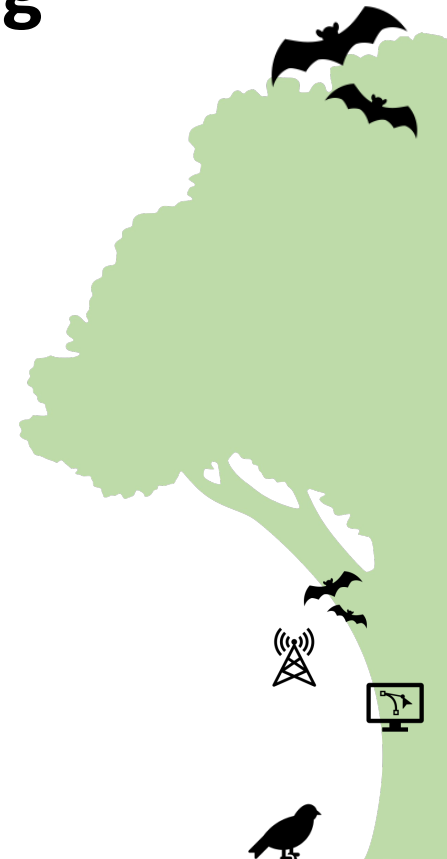
SDR signal analyzer self-monitoring

- **Every** data stream of an **SDR** is analyzed in a **separate process**.
- Processes **monitor themselves** such that an **internal timestamp** is computed based on the samples received from the SDR and compared to the system clock.
- If the samples are **lacking behind** more than the configured sdr_timeout_s, some samples have been skipped and the time of a detected signal is not accurate anymore. The error is reported as and the analyzer is terminated.

```
SDR 0 total clock drift (2.612 s) is larger than two blocks, signal
detection is degraded. Terminating...
```

- In addition to this, every analyzer process (re-)sets a **timer when receiving samples**, which triggers, if no block is received within the configured sdr_timeout_s:

```
SDR 0 received SIGALRM, last data received 2.114 ago.
```



sdr_timeout_s
sdr_max_restart

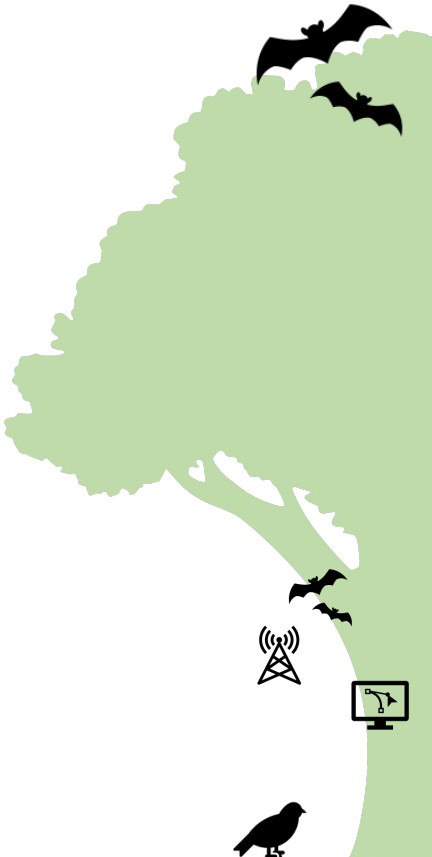
tRackIT OS: Robustness in pyradiotr.

Central monitoring loop

- Check the **timestamp** of the **last received sample block**.
- If the timestamp of the last block is **beyond** the configured sdr_timeout_s, the analyzer is terminated:

```
SDR 0 received last data 2021-06-21 20:27:55.215609; timed out.
```

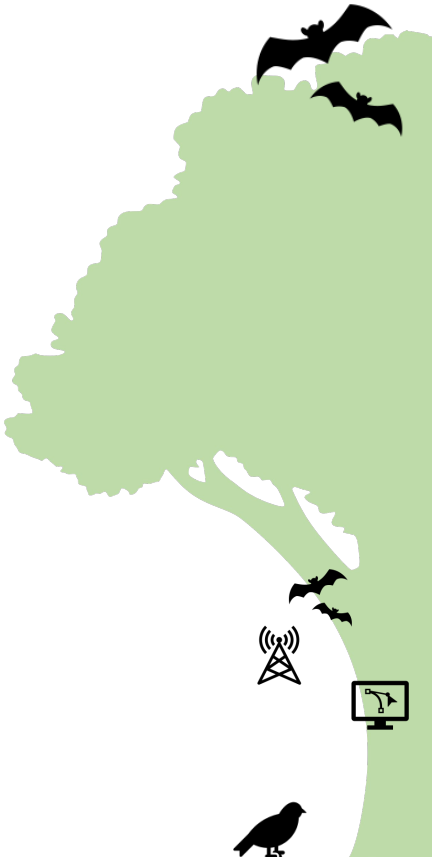
- When an analyzer is terminated, the central monitoring loop **restarts** the analyzer up to sdr_max_restart times.
- All other analyzers are allowed to continue operation normally.
- If the restart count is reached, pyradiotracking will terminate itself.



tRackIT OS: Systemd Service Supervision

Restart of pyradiotracking

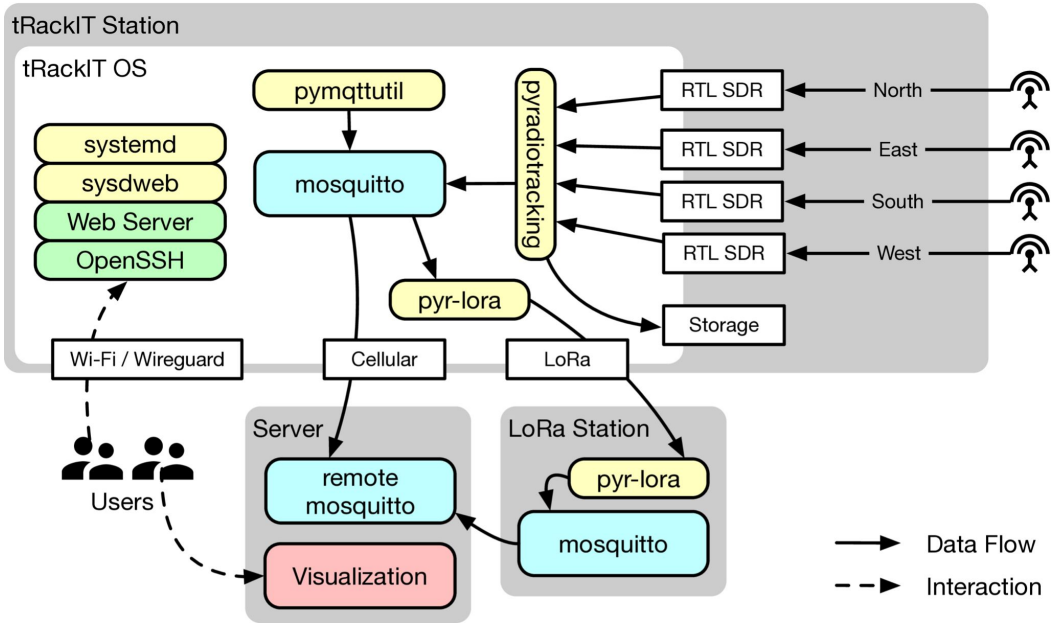
- Systemd keeps track of the radiotracking service.
- Should pyradiotracking terminate itself or crash in some way, it restarts the service after a 10 seconds pause.
- If more than five restarts occur over the duration of 10 minutes, the system will reboot itself.
- Rebooting the system also restarts the USB stack which can sometime help to mitigate errors in signal retrieval.



Note: When debugging errors or copying data from failed stations, the automatic rebooting can sometimes be annoying. You can always disable the `radiotracking` service using `sudo systemctl stop radiotracking`.

tRackIT OS: Software Components

GNU/Linux meets IoT components



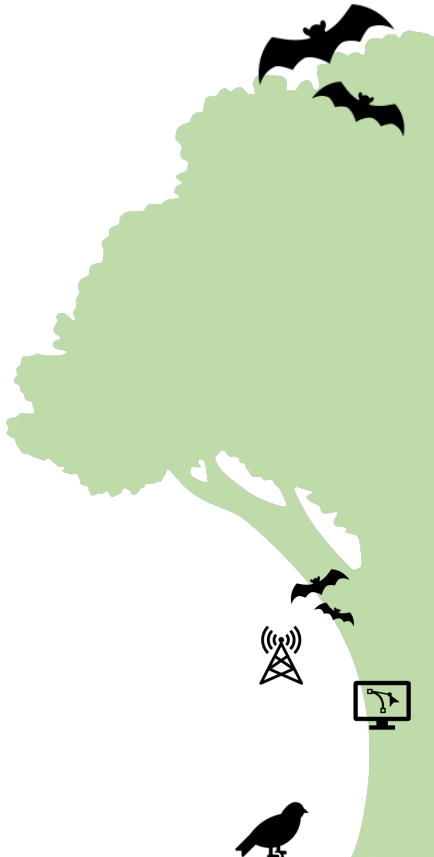
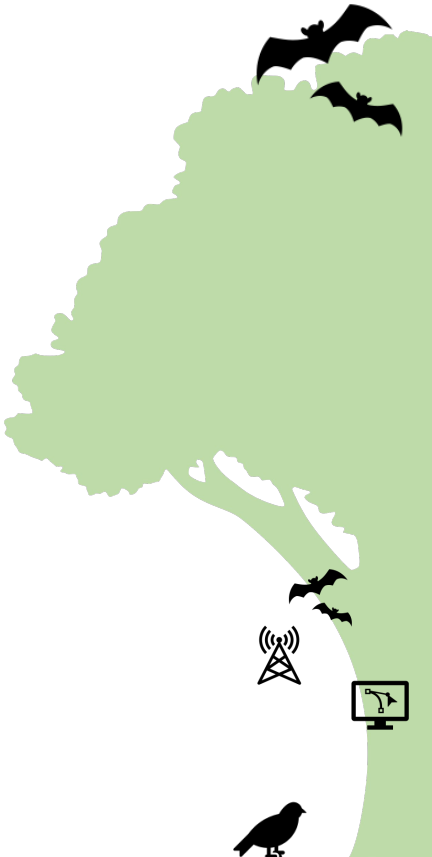


Fig. 2: Overview of the main software components of a tRackIT OS distribution.

tRackIT OS: Software Components

Glossary

- **pyradiotracking**: Signal detection software including dashboard
- **pymqttutil**: System monitoring
- **systemd**: Service control and supervision
- **sysdweb**: Web interface for sytemd
- **wireguard**: VPN / remote access
- **ssh**: Secure shell for WiFi / wireguard login
- **MQTT**: IoT communications protocol used for signal and monitoring data transmission
- **LoRa**: *Long Range* protocol used for signal transmission in areas without cellular coverage
- **InfluxDB**: Streaming database consuming detected signals and monitoring data
- **Grafana**: Data visualization at a central server

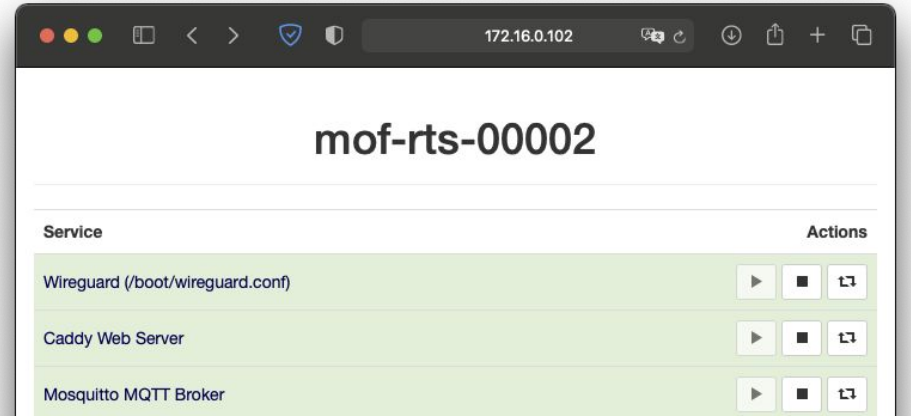


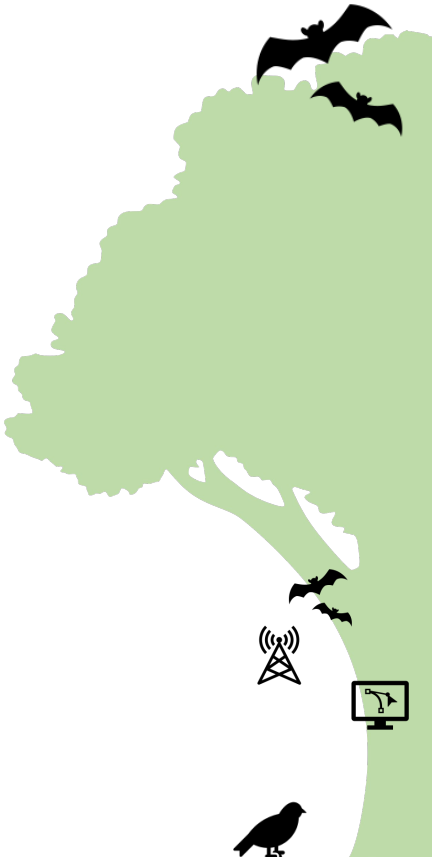
tRackIT OS: Monitoring

Aspects of a tRackIT stations health state

Sysdweb Color Indicator:

- Does not represent the internal state of, e.g., pyradiotracking.
- It shows wether the radiotracking service is running or has failed.
- If the respective service is marked green, it can still be possible that pyradiotracking is starting a single failed logger over and over again.



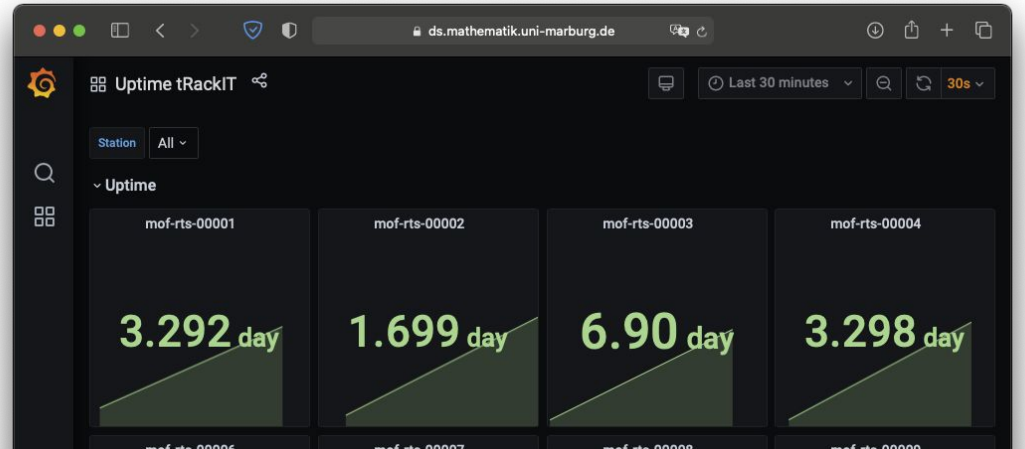


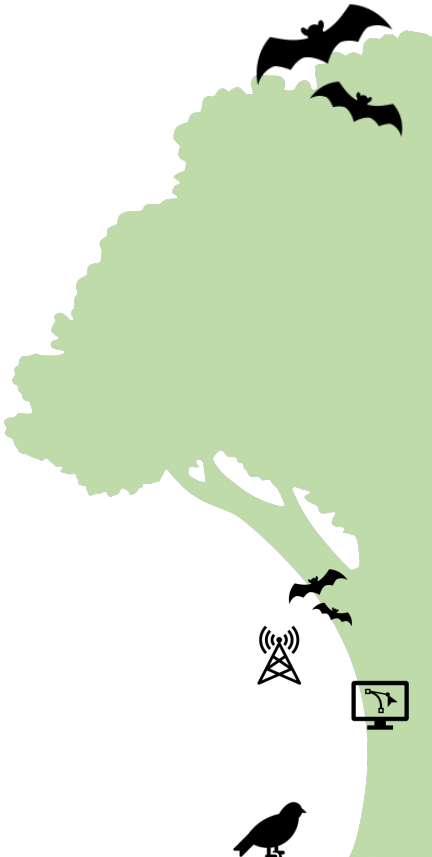
tRackIT OS: Monitoring

Aspects of a tRackIT station's health state

Uptime:

- The uptime is a useful indicator to assess the general health state of a station.
- If the uptime is low, and the station is restarting over and over again, it can be a simple indicator for a configuration of high demand or failing hardware.





tRackIT OS: Monitoring

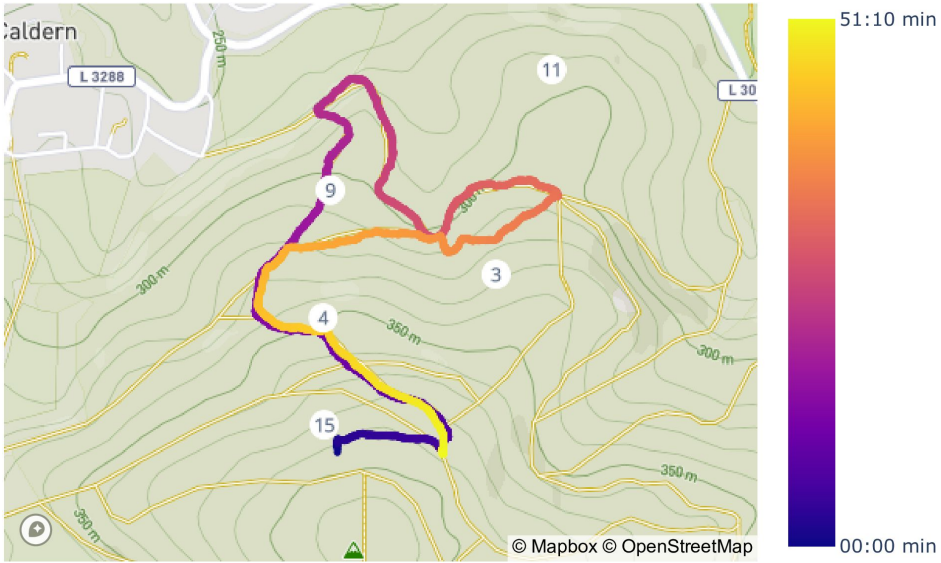
Aspects of a tRackIT station's health state

Detected Signals:

- Observation of the detected signals is the **most direct evidence** of a functioning station.
- However, it is not always reliable, since there may **not be a VHF tag around all the time** to be detected by the station.
- Watch the signals detected on all SDRs.

tRackIT OS: Experimental Evaluation

Experimental setup



Marburg Open Forest installation, using 5 tRackIT stations.

a) tRackIT OS 0.7.0
b) paur 4.2

- VHF sender + GPS receiver
- 0:51 h experiment runtime
- 3,193 sent signals

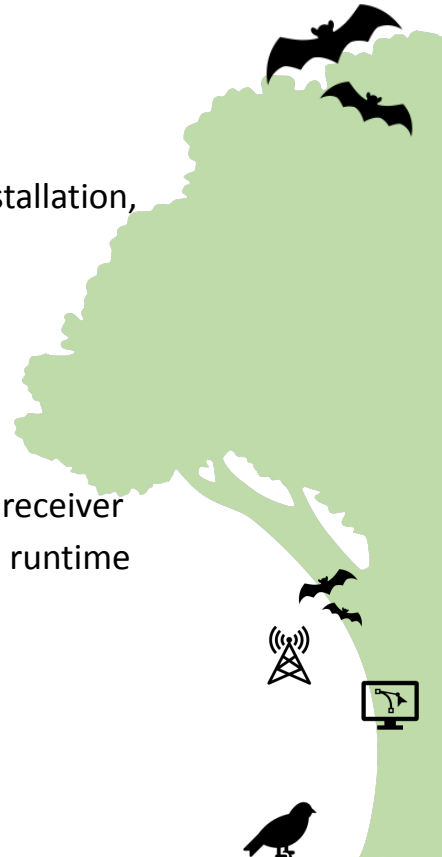
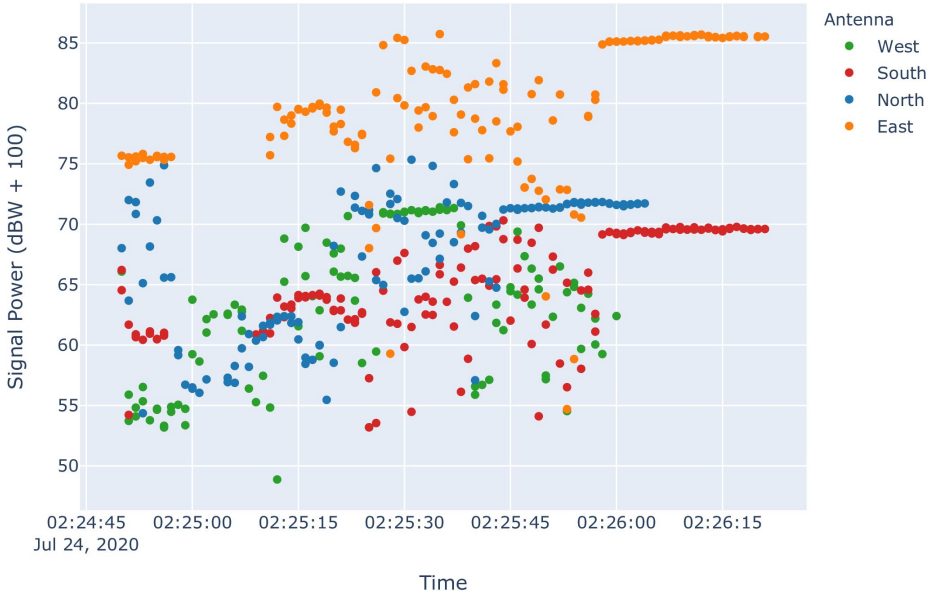


Fig. 7: GPS trace of the experimental evaluation track and the corresponding tRackIT stations.

tRackIT OS: Experimental Evaluation

Signal delay in paur 4.2



Signal delay of up to 8 seconds on different antennas of the same tRackIT station.

Signal delays lead to problems in signal matching, hence false bearing calculations.

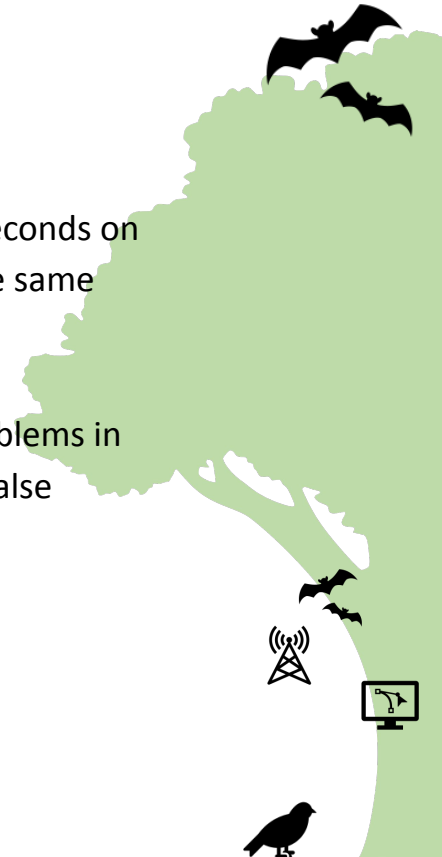
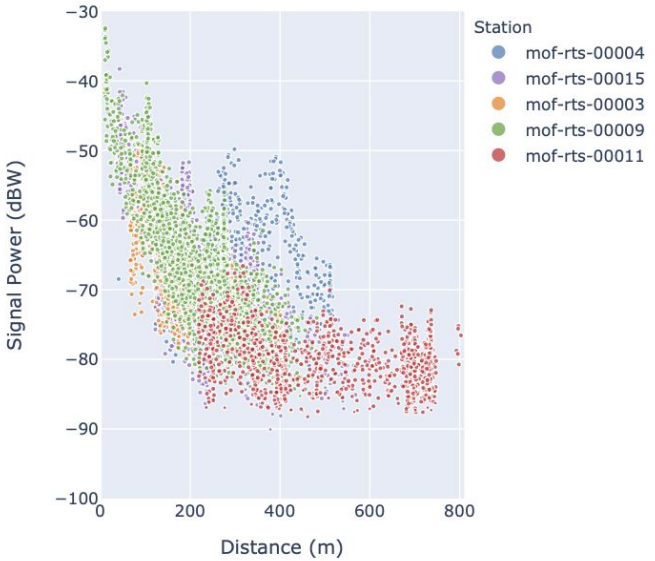


Fig. 8: Example of signal delay among different receivers observed in the 2020 field season using paur.

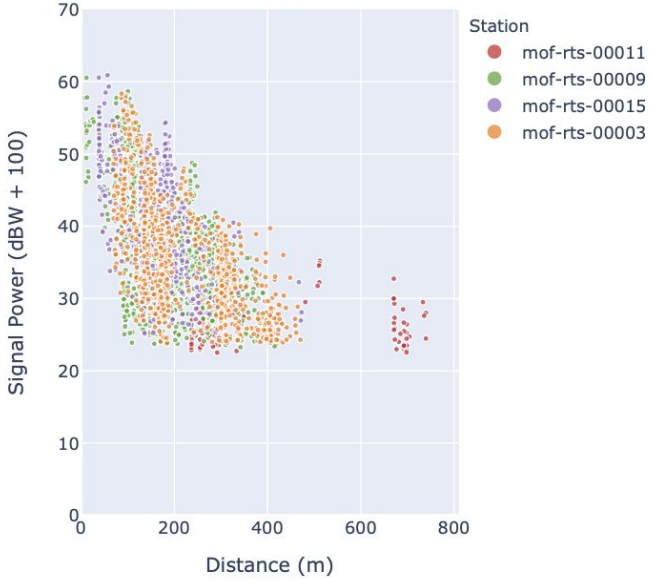
tRackIT OS: Experimental Evaluation

Comparing tRackIT OS and paur signal reception.



a) tRackIT OS

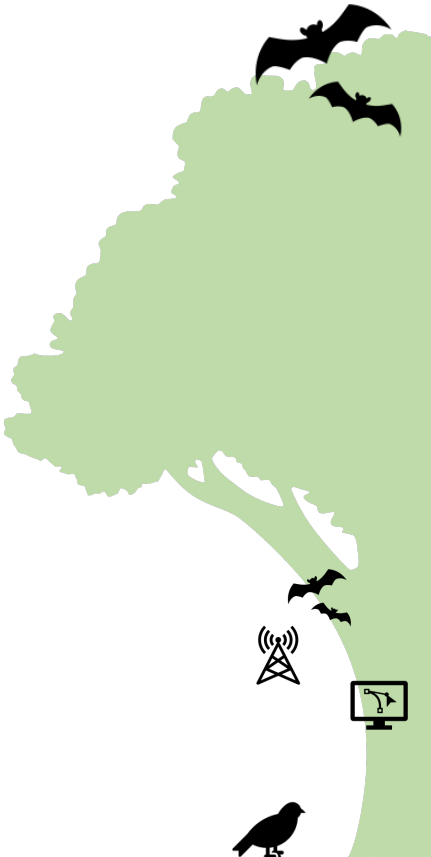
4,438 sigs (+103.3 %) | +62.7% (with -78 dBW threshold)



b) paur

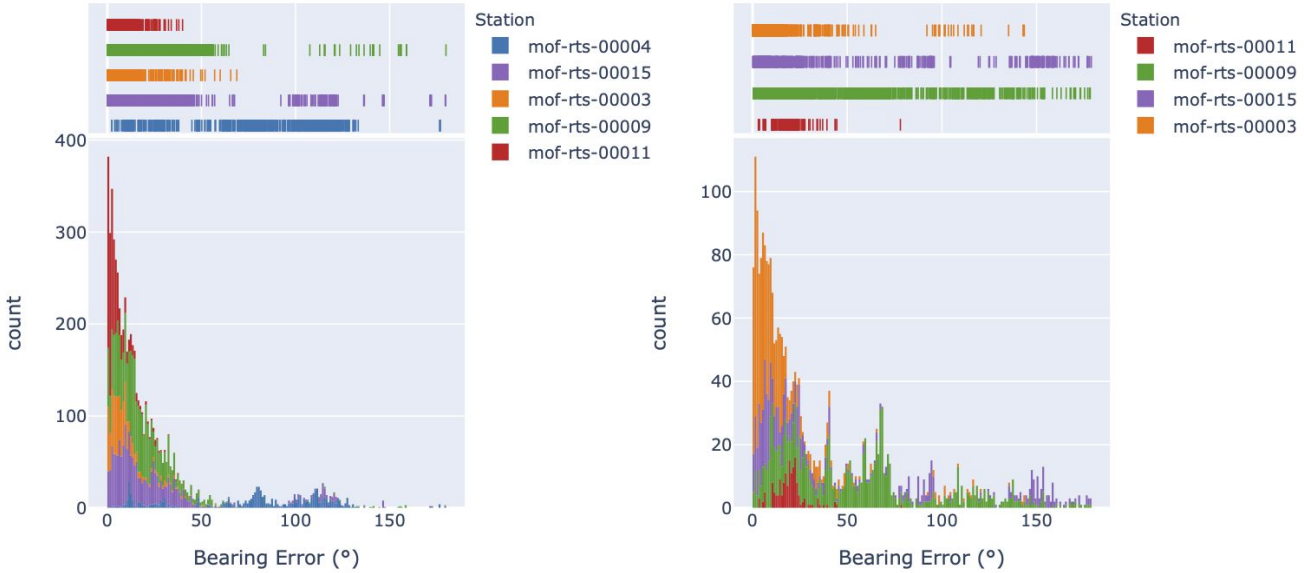
2,828 signals

Fig. 10: Signal power and distance to a receiving station.



tRackIT OS: Experimental Evaluation

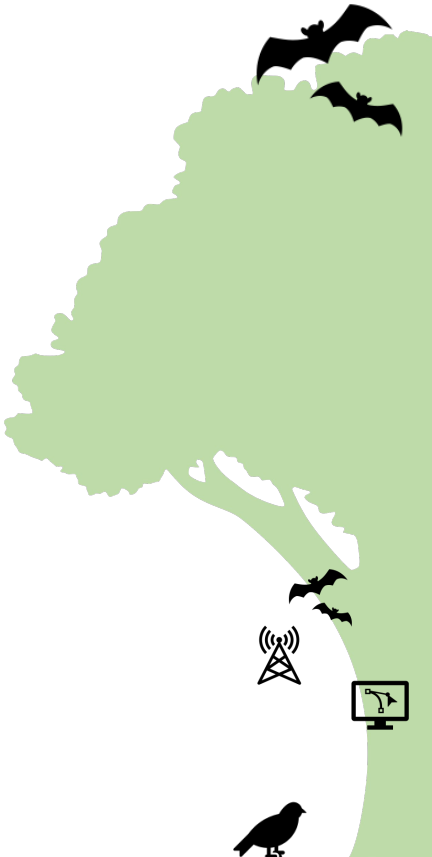
Comparing tRackIT OS and paur signal reception.



a) tRackIT OS
mean: 23.7° | standard dev 30.7°

b) paur
mean: 38.9° | standard dev 42.6°

Fig. 11: Histogram of bearing errors.



Conclusion

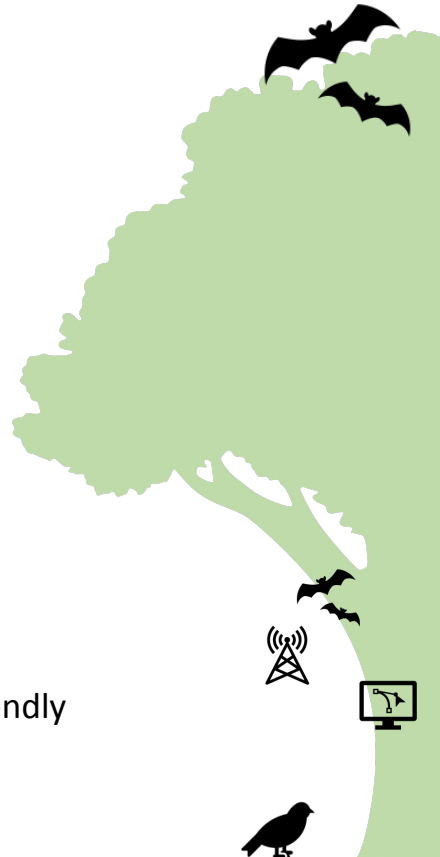
Results and future work

tRackIT OS

a) enables reliable VHF signal detection for bearing calculation,
b) increases the number of usable signals by 103.3%,
c) improves the mean bearing calculation error from 38.9° to 23.7°, and
d) introduces only a slight overhead in power consumption of 2.55% or 0.2 W.

Challenges:

- Exact bearing calculation is challenging, since signals are affected by multiple factors: vegetation, topology, humidity and rainfall. Ideas:
  - Incorporate context information; aggregate data from multiple stations.
- Automated continuous preparation and processing of collected data: building a user-friendly widely applicable animal tracking system.



Natur 4.0
Sensing Biodiversity

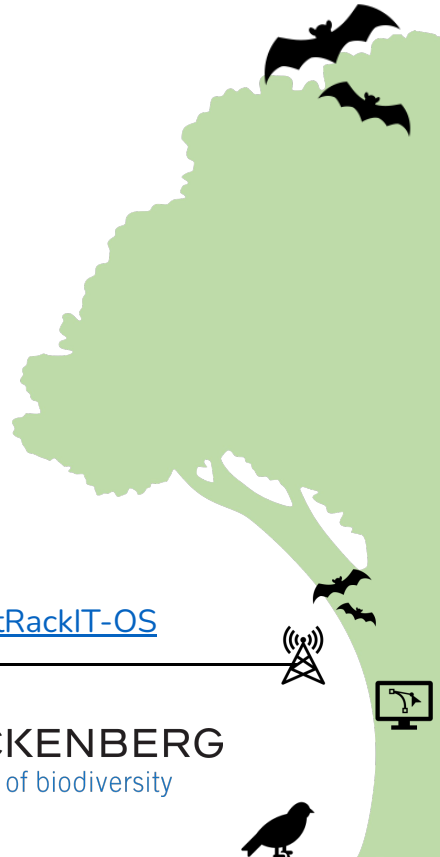
LOEWE
Exzellente Forschung für Hessens Zukunft

tRackIT OS: Open-source Software for Reliable VHF Wildlife Tracking

EOF

Jonas Höchst hoechst@informatik.uni-marburg.de
Jannis Gottwald
Patrick Lampe
Julian Zobel
Thomas Nauss
Ralf Steinmetz
Bernd Freisleben

Software Development & Download: https://github.com/Nature40/tRackIT-OS



Philipps Universität Marburg

TECHNISCHE UNIVERSITÄT DARMSTADT

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

SENCKENBERG world of biodiversity