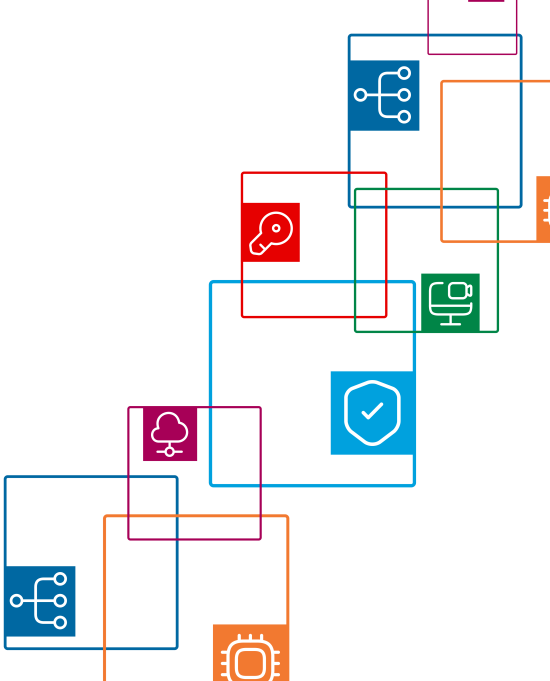
cesnet

Polbox – Longterm SoP Acquisition

Martin Šlapák, Jan Radil, et al.

CESNET

12. 2. 2026



CESNET Polbox

Figure 1: The Polbox prototype developed for SUBMERSE project

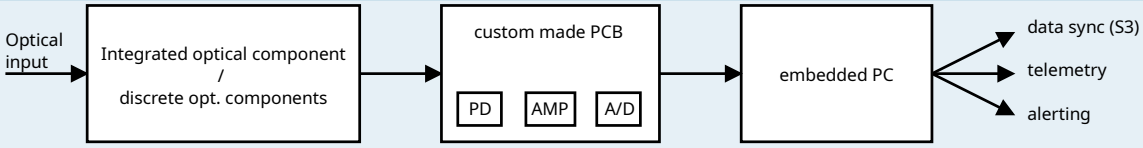
Polbox – features

- Long-term continuous SoP acquisition.
- Fast sampling rate at 20 kHz, 4 channels with 16 bit resolution per channel.
- Variable optical input source: a tap on **existing signal**, or custom laser source.
  - Prototypes are capable to operate from 1270 to 1610 nm (ideal 1520–1610 nm).
  - Optical input power from -15 to 15 dBm (special version from -35 to 5 dBm).
- Durable 1U 19" chassis with dual power sources.
- Monitoring compatible with Influx DB and Grafana.
- Local embedded SSD storage, optionally synchronized to the remote filesystem or object storage (S3 compatible).

How it works

cesnet

Block schema of Polbox



- State of Polarization – orientation and trajectory of the vector of the electric field.
- Two principal states – horizontal and vertical (and anything in between).
  - Linear, circular, elliptical polarizations.
- No reason to go into the theory – but all of us know Polarization Sunglasses, right?
- Only one polarization can pass through.

Use-cases I

- Detecting acoustic-mechanical vibrations and disturbances in fibre and its neighbourhood – e.g. for seismic activity.
- Physical perimeter intrusion detection.
- Critical infrastructure manipulation alerting system.
- Long-term SoP acquisition for later analysis or on-the-flow processing.

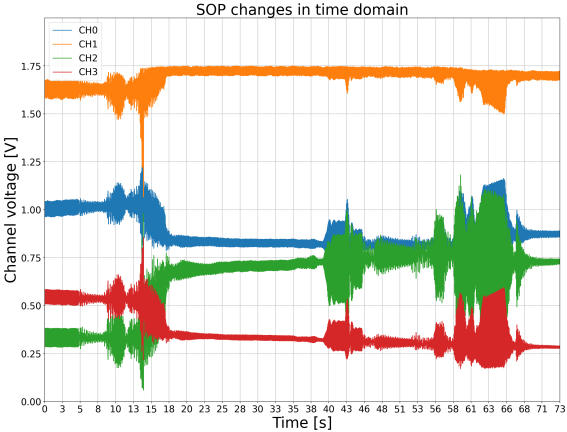
Use-cases II

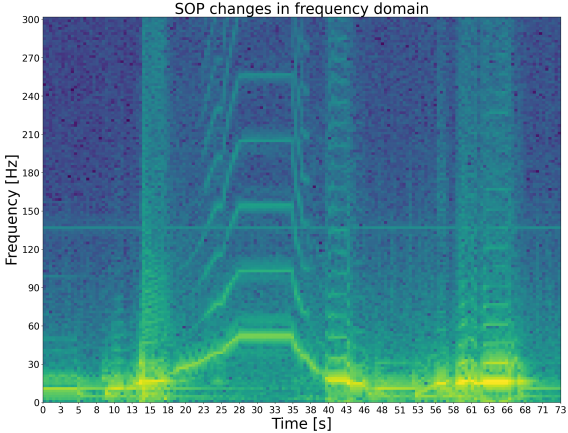
Practical notes and technology limitations

cesnet

- Important to mention one big difference between SoP and DAS equipment.
- **SoP cannot tell us where the disturbance is.**
- This can be pretty important for some use-cases, or users.
  - It is possible, using one well-know Speed-Distance-Time (SDT) equation.
  - distance = speed x time, easy, huh? Like vehicles accidents...
  - But photons are pretty quick vehicles, travelling 200 000 km/s.
  - This equals 0.2 m in 1 nanosecond. 200 m in 1 microsecond. And 200 km in 1 millisecond.
- What does this mean?
- We need very accurate time stamping methods like *White Rabbit*.
  - Sampling every 1 ms (1 kHz) is really not good enough.
  - This sampling rate is used in coherent transcievers. They can report SoP but spatial resolution is as written above.

Example data





Data format – raw data stream

cesnet

Raw data stream

```
#         S0/raw0   S1/raw1   S2/raw2   S3/raw3
array([[ 0.1234,   0.2345,   0.3456,   0.4567],    # row 0
       [ 0.2234,   0.3345,   0.4456,   0.5567],    # row 1
       [ 0.3234,   0.4345,   0.5456,   0.6567],    # row 2
       ...
       [ 0.8765,   0.1111,   0.2222,   0.3333]])   # row N
```

- In case of uncalibrated unit, there are 4 raw values of photodiode's voltage instead of S0–S3.

Data format

- Recorded SoP *raw time-series data* are stored locally in HDF5 format with *metadata* describing the recording context, acquisition parameters and could be synchronized to the remote storage.
- The format is designed to be compact, self-descriptive, and easily readable from common scientific tools such as Python (h5py, pandas) or MATLAB.
- **Actually, there is no standard for SoP data storage format and we have proprosed the initial draft and working hard on the first public version.**
- Depending on the input signal, the recorded data stream approximately needs 6-13 GB daily (DAS data volume starts at 1 TB/day).
- Actual data file is rotated each hour (configurable).

Data format – metadata

cesnet

Metadata format

```
creator: str                    Identifier of the software that created the datase
description: str                Experiment description.
channels: int                   Number of channels | typically 4.
sampling_frequency: int         Sampling frequency in Hz (default: 20000).
timezone: str                   Time zone offset in the format +XX:YY.
timezone_name: str              Time zone name (e.g., CEST).
channel_unit: str               Unit of the channel data, typically Volt or a norm
samples: int                    Number of samples per channel.
duration: float                 Duration of the recording in seconds.
epoch_start: unix_timestamp     Unix timestamp of the recording start.
epoch_written: unix_timestamp   Unix timestamp of when the dataset was written to
timestamp_start: str            Human-readable timestamp of the recording start.
timestamp_written: str          Human-readable timestamp of when the dataset was w
events: str                     JSON object containing event annotations (see the
```

Data format – metadata – events

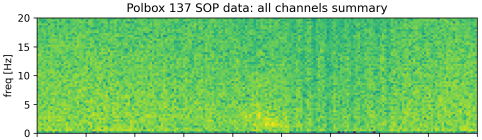
cesnet

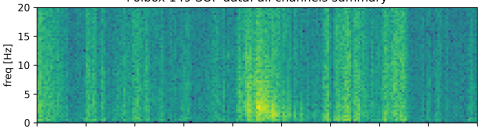
- Two types of an event annotation: *interval* – covers a defined time span, or *event* – represents a single point in time.

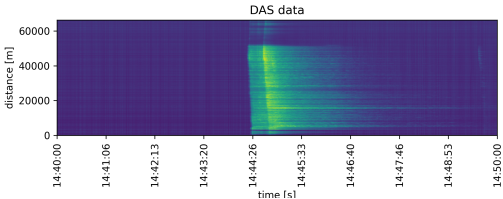
Metadata for event annotation

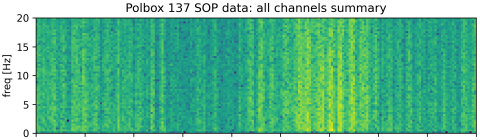
```
start: int                Index (sample number) of the event start.
end: int | null           Index (sample number) of the event end; null for point eve
time_start: str           Human-readable timestamp of the event start.
time_end: str | null      Human-readable timestamp of the event end; null for point
label: str                A human-readable label.
label_class: str          Event class, or an empty string "" if not classified.
annotation_type: str      Type of annotation: "event" | "interval".
detector: str             Description or identifier of the detector or detection met
score: float              Event score (e.g., class probability or confidence value).
properties: dict          Additional properties of the event.
  -> freq_low: float         Lower frequency of the event.
  -> freq_up: float          Upper frequency of the event.
  -> channel: int            Channel number (0-3).
```

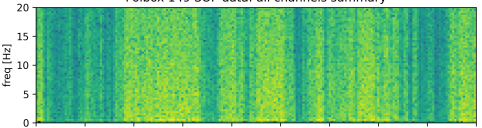
Example data – SUBMERSE SoP and DAS aligned

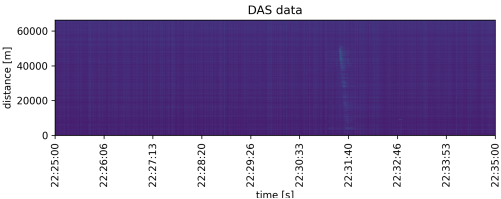












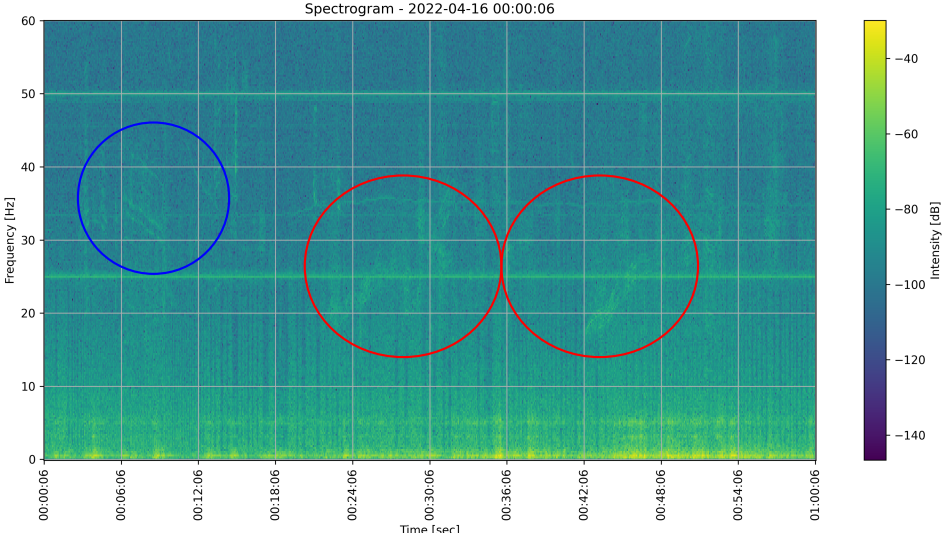
Example data – SUBMERSE SoP

A periodic signal at 5.5 kHz

cesnet

Example data – terestrial SoP

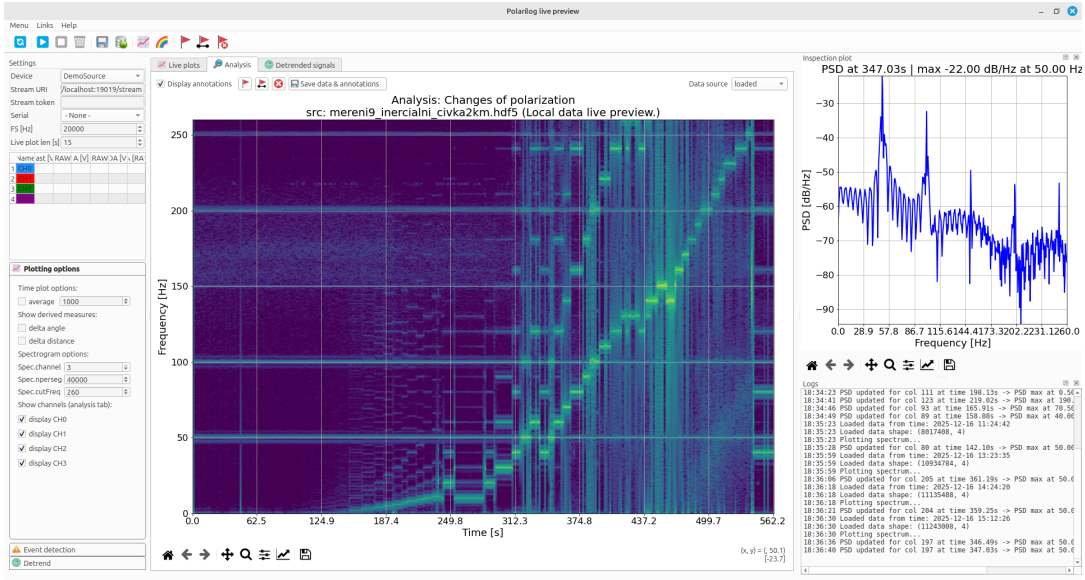
Disturbances along the railway



Analysis tools

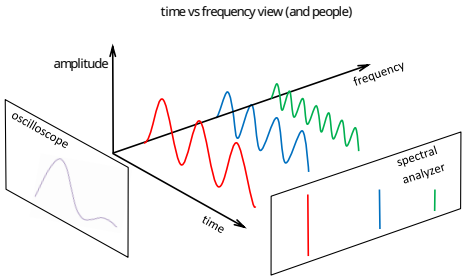
- A specific SoP data analysis is made in Python, utilizing well known tools as *h5py, scipy, matplotlib, pandas, ...* with focus on particular task.
- Common routines as data plotting in time/frequency domain were already prepared as parametrized scripts.
- For live demonstration we have developed a GUI (Polive) capable of recording, displaying the live time and frequency domain graphs, support for manual event annotation and also offline analysis and annotation.
- We have been working on the automated event detection with several approaches both in time and frequency domains.
- **The problem is unlabeled data: We see something but cannot say what it is!**

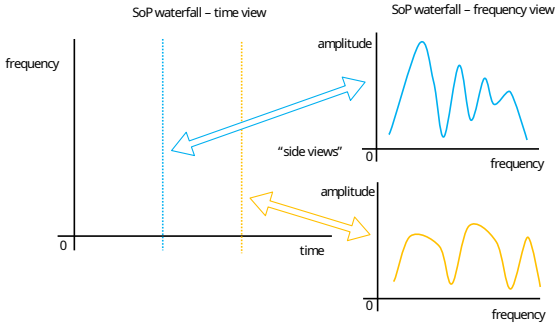
Analysis tools – Polive GUI



Frequency vs. Time Views

- It seems there are two groups of scientific/research people.
- One group see in the time domain and the other group in the frequency domain.
- You?:-)





Conclusion

- SoP is one alternative to DAS (Interferometric methods also available.).
  - Producing significantly less data.
  - Can detect higher frequencies easily.
  - Localization of events not easy but possible (more details in the first reference – OFC2025).
- CESNET Polbox – transfer of know-how to the license partners.
- Thanks to colleagues from CESNET.
- Thank you for your kind attention.

References

- N. Okada et al., *Demonstration of State-of-Polarization Change Localization Based on Digital Coherent Transceivers*, OFC 2025, paper M3Z.1.
- GN5-2-White-Paper_Fibre-Sensing Technologies.pdf
- GN5-2-WP6-White-Paper_Fibre Sensing: Technologies, Users and Use Cases for NRENs
- Tapping into the ground: terrestrial applications of Distributed Fibre Sensing
- GÉANT Infoshare: Backscattering based sensing technologies, equipment and use cases