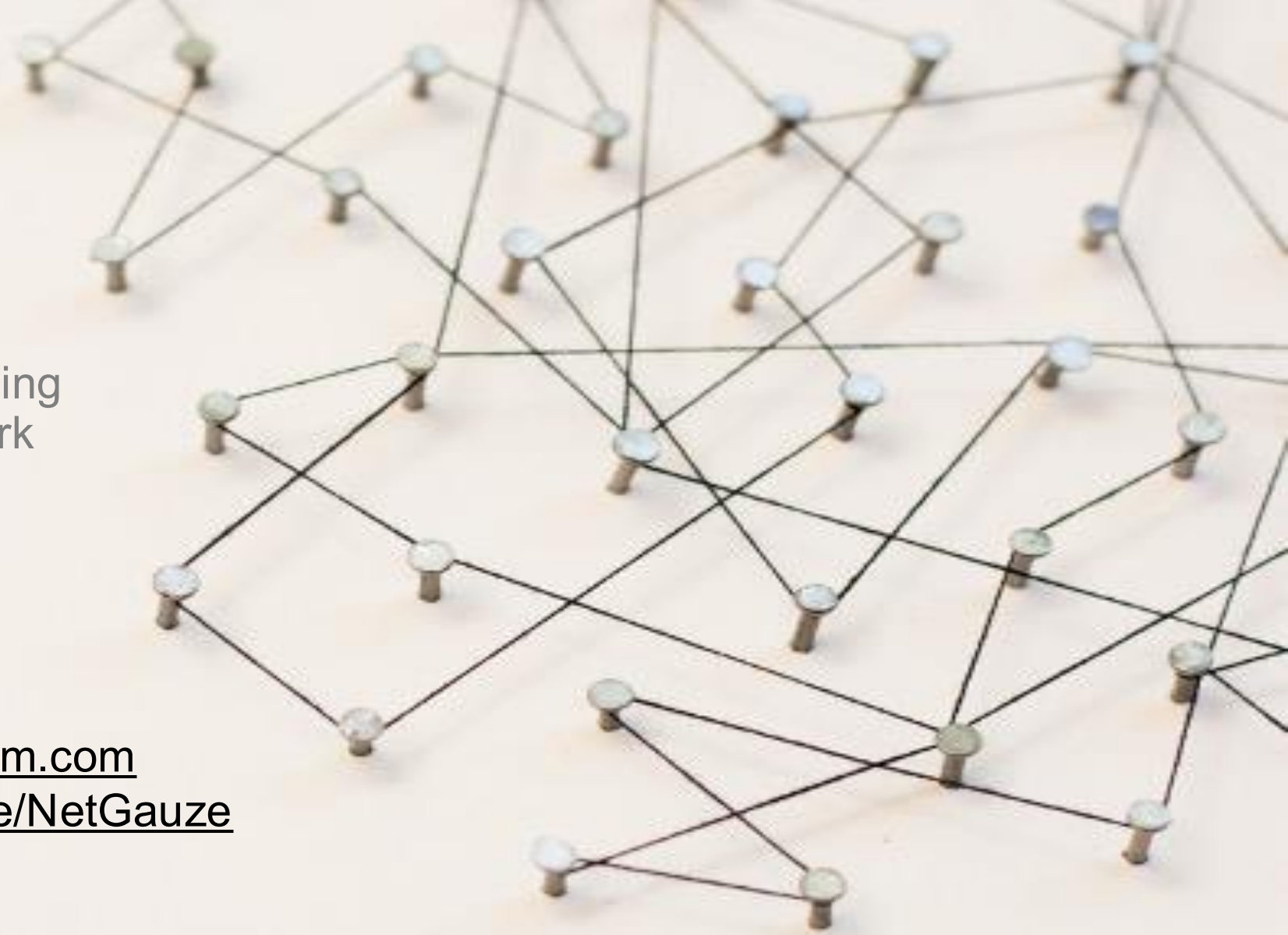
NetGauze

The building blocks for building resilient and scalable network telemetry platforms



Ahmed Elhassany
ahmed.elhassany@swisscom.com
https://github.com/NetGauze/NetGauze

PacketFest 08.05.2025

More network visibility!

What does it take?

More network visibility!
What does it take?

Supporting plethora of protocols

Data plane: IPFIX, sFLow, ..

Control plane: BMP, BGP, IS-IS,..

Management: YANG-Push, NETCONF,..

More network visibility!

What does it take?

Scaling to thousands of devices

Efficient resource handling

High concurrency

Resiliency

Fault tolerance

More network visibility!
What does it take?

Advanced features

Flexible aggregation

Correlating different data streams

Enrichment with dynamic sources

More network visibility!
What does it take?

Advanced features

Flexible aggregation

Correlating different data streams

Enrichment with dynamic sources

NetGauze architecture principles

Our secrete sauce

NetGauze architecture principles

Our secrete sauce

Construct reusable modules

Small modules with clear interfaces

Test, test, and then test again!

Fuzz testing for core modules

NetGauze architecture principles

Our secrete sauce

Scale to the moon

Use actor programming model

Synchronize via message; no locks

Use async I/O

NetGauze fast packet parsers

Speed, resiliency and standard conformance

NetGauze fast packet parsers

Speed, resiliency and standard conformance

Rich type system

```rust
#[derive(Debug, Clone, PartialEq, Serialize, Deserialize)]
#[cfg_attr(feature = "fuzz", derive(arbitrary::Arbitrary))]
pub enum BgpMessage {
    Open(BgpOpenMessage),
    Update(BgpUpdateMessage),
    Notification(BgpNotificationMessage),
    KeepAlive,
    RouteRefresh(BgpRouteRefreshMessage),
}
```

NetGauze fast packet parsers

Speed, resiliency and standard conformance

Supports multiple serialization formats

```
let msg = BgpMessage::KeepAlive;

let json_str = serde_json::to_string_pretty(&msg)
    .expect("Failed to serialize to JSON");
println!("JSON representation of BGP packet:\n{json_str}");

let yaml_str = serde_yaml::to_string(&msg).expect("Failed
to serialize to YAML");
println!("YAML representation of BGP packet:\n{yaml_str}");
```

NetGauze fast packet parsers

Speed, resiliency and standard conformance

Verbose errors

```
LocatedBgpMessageParsingError {
  span: BinarySpan {
        offset: 21,
        fragment: [255, 1]},
  error: BgpRouteRefreshMessageParsingError(
          UndefinedOperation(
              UndefinedRouteRefreshSubcode(255)
          ))
}
```

NetGauze fast packet parsers

Speed, resiliency and standard conformance

Fuzz testing

```rust
#[derive(Debug, Clone, PartialEq, Serialize, Deserialize)]
#[cfg_attr(feature = "fuzz", derive(arbitrary::Arbitrary))]
pub enum BgpMessage {
    Open(BgpOpenMessage),
    Update(BgpUpdateMessage),
    Notification(BgpNotificationMessage),
    KeepAlive,
    RouteRefresh(BgpRouteRefreshMessage),
}
```

NetGauze roadmap

Develop, test, deploy, repeat!

NetGauze roadmap

Develop, test, deploy, repeat!

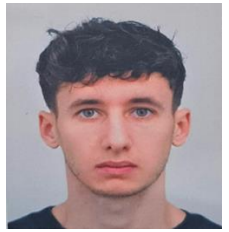
Open-source from

day 1

Ahmed Elhassany
Swisscom

Leonardo Rodoni
Swisscom

Uwe Storbeck
Swisscom



Maxence Younsi
INSA Lyon

NetGauze roadmap

Develop, test, deploy, repeat!

Protocol support

BGP

BGP Monitoring Protocol (BMPv3)

BGP Monitoring Protocol (BMPv4, under review)

NetFlow v9

IPFIX: supporting all IANA code points.

UDP-Notif

YANG-Push (pull request soon)

NETCONF late 2025

sFlow 2026?

NetGauze roadmap

Develop, test, deploy, repeat!

Listener libraries

- BGP
- BMP
- Flow: NetFlow v9 and IPFIX
- UDP-Notif: Supporting YANG-Push
- NETCONF late 2025
- sFlow 2026?

NetGauze roadmap

Develop, test, deploy, repeat!

Advanced analytics

Flow Aggregations

Flow enrichment

YANG-Push enrichment 2025

BGP->Flow correlations 2025

YANG->BGP->Flow correlations 2026

sFlow 2026?

NetGauze roadmap

Develop, test, deploy, repeat!

Output formats	Apache Kafka JSON
	Apache Kafka AVRO
	Apache Kafka YANG 2025/2026

NetGauze roadmap

Develop, test, deploy, repeat!

Initial deployment

IPFIX collection for ~400 nodes

Receive ~11 thousands messages per second

Kafka AVRO output (after flattening and aggregations)

32 thousands messages per second.

NetGauze

So what's it about?

Collections of Rust networking libraries

A data collection of network telemetry

An analytics and aggregation engine for network telemetry

And more!