

# perfs-SONAR

## Visualising perfSONAR on-demand measurements

1st European perfSONAR Workshop

Antoine Delvaux — [antoine.delvaux@man.poznan.pl](mailto:antoine.delvaux@man.poznan.pl)

Jisc, London, UK



# Once upon a time...

- ... there was a psUI
- Multi-user GUI
  - As a Service provided by GÉANT
  - As a package you could install on your network
- Visualising past measurements (archived)
- GUI to trigger measurements on remote pS nodes
- But...
  - perfSONAR evolving, multiple changes needed
  - hard to maintain and main developers had left
- Now discontinued

# What are the use cases?

- Graphical User Interface to perfSONAR, to pScheduler
  - to use instead of the CLI
- Most missing feature:
  - UI for on demand measurements
- Browsing and comparing past measurements
- Viewing details (intervals) of latency and throughput tests
- Ideally it would be a multi-users tool, federated

# What now?

- We'd like to replace it, trying different options
- Visualisation
  - Standalone GUI based on D3.js
  - As Grafana panel and dashboard
- Trigger a test
  - Dedicated UI
  - Call pScheduler API
- Retrieve measurement data
  - Archive (Esmond API)
  - pScheduler API

# What have we tried so far?

- Grafana visualisation through dedicated and new:
  - Esmond data source
  - pScheduler data source
- Dedicated visualisation using D3.js
- LS and pScheduler proxy
  - To cache and navigate LS information
  - To cache pScheduler results when there are many (ex: 24h repeated measurements)

# What are the limitations?

- Esmond data source
  - Difficult to navigate metadata in Grafana, need separate UI
- pScheduler data source
  - Difficult to navigate LS and construct generic test specs in Grafana, need separate UI
  - Grafana is good at time series, but we mostly have one-time measurements, can we display that too?
  - How long is pScheduler storing data?
- D3.js visualisation: more code maintenance

# Adding a separate UI

- Navigate LS data
  - Communities
  - Source and destination pair
- Schedule and trigger a pScheduler tasks
- Basic PoC for now
  - Only latency and throughput
  - A few parameters only
- Can be tightly integrated with D3.js visualisation

# We want to collect feedback

- Your feedback!
- Set of 5 questions to help us better understand how you would use such a new GUI
- Go to <https://menti.com> with code 64 44 32