Introducing **LTTng Scope**
Content

- Project goals
- User experience design goals
- User interface design
- Architecture
- Current status
- Demo
- Roadmap
Project Goals

- Trace viewer for CTF / LTTng traces
- Streamline user experience
  - Easy to install
  - Discoverability of features
  - Scenario-driven user interface
- Layered architecture
UX Design Goals

• Driven by investigation scenarios, rather than available data.
• Allow user to focus on the most important information at each step of the investigation.
• User should discover features intuitively without having to read documentation.
UI Design

- Views and analyses are exposed as available widgets.
- User can add/remove/reorder widgets in the view.
  - Can instantiate multiple widgets of the same type, then apply different settings.
UI Design
UI Design

- Widgets should work *together*, not as separate silos.
- Project-wide settings like filters, bookmarks should apply to all widgets similarly.
UI Design

• Filtering is key:
  – Tracing generates a lot of data,
  – The tool should reduce noise,
  – Allow the user to focus on interesting data.
Layered Architecture

- LTTng Scope
  - UI
  - CLI / library
  - Jabberwocky (trace analysis library)
    - CTF reader library
    - State system library
Current Status

- Standalone application based on JavaFX.
- ~6 MB package!
  - Including the client and library
- Implemented features
  - Event table (multiple instances)
  - Event count histograms (XY-charts)
  - Time graph views for kernel traces
    - Threads, CPUs, IRQs
  - Filters based on event name
Demo

Filter results in different views
Demo
Demo
Demo
Demo
Demo

Filter is now defined at the project level.

XY charts add series corresponding to the matching events (Series style to be based on symbol color?)

Time graphs indicate matching events on the timeline.

Event table tags matching events with the symbol.
Demo

Filter can be disabled and restored with one click.
UI Design Roadmap

• *New project* dialog starts with high-level concepts:
  – CPU analysis
  – I/O analysis
  – Network analysis
  – ...

• This sets up a pre-defined set of widgets with pre-defined settings.

• User can then customize, add/remove, etc.
Architecture Roadmap

● Split client/server architecture
  – Choose the best frontend for the job:
    • CLI, Desktop, Web
  – Front-end should be as simple as possible.
  – Opens the door to completely new use cases:
    • Continuous Integration plugin embedding analysis results.
    • Distribute analyses over a cluster/cloud.
Architecture Roadmap
Architecture Roadmap

- Collaboration with Polytechnique students and Ericsson Trace Compass developers on defining a presentation protocol.
Links

- https://github.com/lttng/lttng-scope
  - Installation instructions
  - Report issues