Ericsson Update

Trace Compass, TSP, Theia-Based front-end and more

Matthew Khouzam
BNEW SAN Tools Engineering
2021-06-18

This talk

- Recap of Ericsson's involvement with the project
- The team
- The tracing ecosystem
  - Trace Compass
  - Theia Trace Extension
- Future work
  - Roadmap
  - Fun announcement
  - Where can you help?
- Thank you

Expected Duration:
18m

Recap

The goal of this talk is to explain where we are coming from to explain our decision making process

Ericsson has an industrial partner in the research project for over 12 years

We develop and use tools such as LTTng and Trace Compass in the company to solve timing issues and hard to debug problems

Ericsson contributes to and maintains Trace Compass

The Montreal Team has been closely collaborating with the academic and industrial partners. Austin team is ramped up!

We foresee contributing even more soon.

About The Team

Bernd Hufmann – Technical Lead

Patrick Tasse – Sr Developer

Matthew Khouzam – Developer / Product Manager

Ibrahim Fradj – Intern

TBA – Developer

Walter Cigana – Manager

Austin team – Developing internal use cases in open-source way

The Ericsson Tracing Ecosystem

- At Ericsson we have very intricate products offered to the public
- At a high level we trace many individual components
- One of the goals of Trace Compass is to provide a unified troubleshooting experience

Traces

- LTTng
- Other Linux Tracers
- CTF Hardware
- Chromium style
- Open Tracing

Logs

- HTTPD
- SSH
- Java (GC)

Trace Compass

- 7.0 Released
  - API Breakage
  - Fix error message for Java 8
- Core level Annotations (Works in Theia based frontend)
- Fix JSON Parsing issues
- Improve UST annotations in Resources view and Control Flow View
- Add vertical alignment to symbols
- Performance improvements
- Many transparent back-end improvements to enhance the trace server

Theia Trace Extension

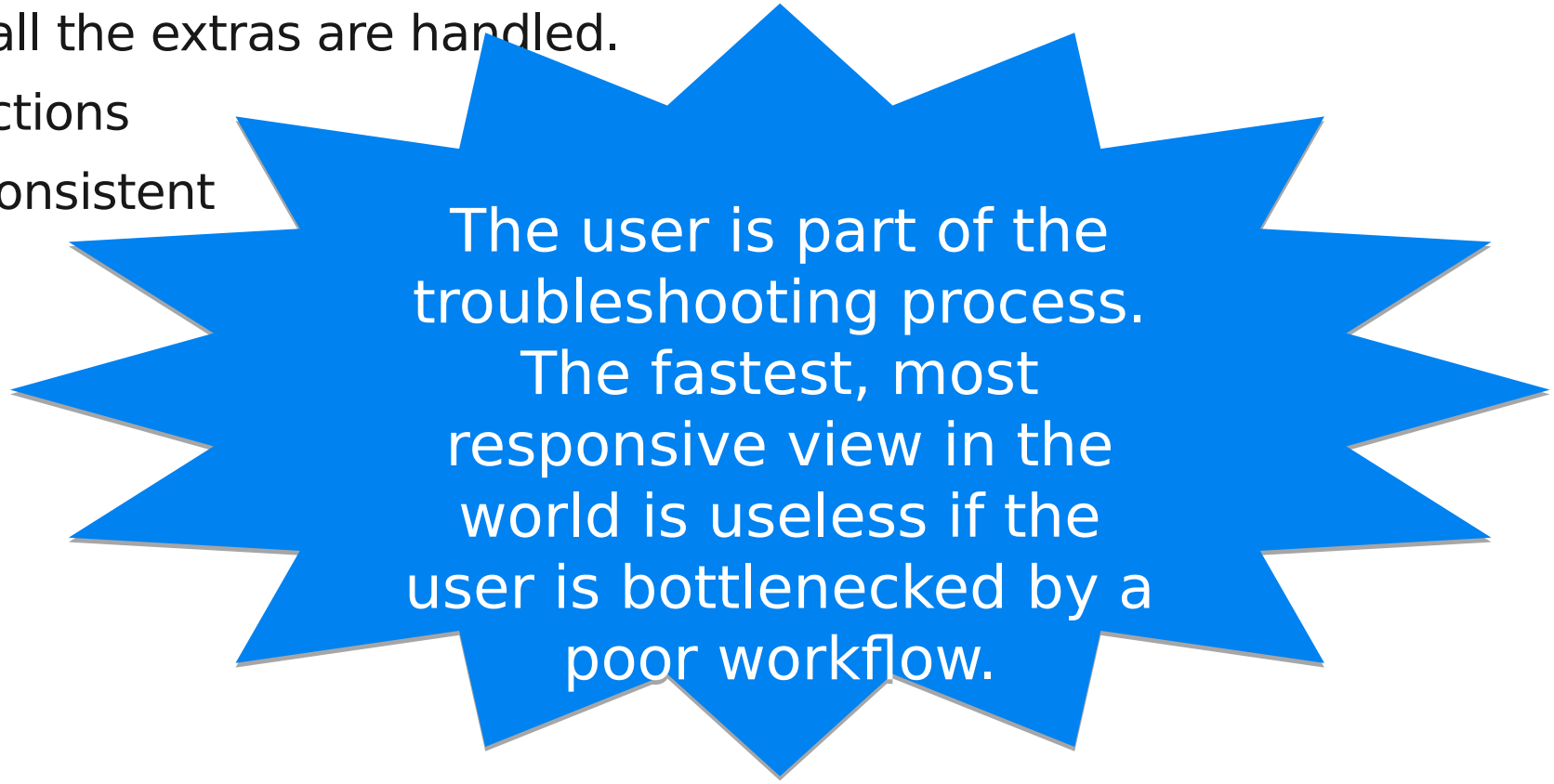
- UX fixes. *Note: Erica opened UX bugs and they get fixed. (HINT HINT)*
- Most improvements are UX related as Theia is a UI shim over Trace Compass's core logic.
- Highlights:
  - Naming improvements like "Open Trace" -> "View Trace", "Analyses"-> "Views"
  - Provide more feedback to the user
  - Performance of the time graph view
  - Annotations
  - Improved "Trace Selection"
  - Tool tips
  - Error messages
  - Hide Trees Based Views as they are currently unsupported (Thanks interns!)

Theia Based Front-End Working parts:

- Automatic server management
- Views
  - CPU usage
  - Histogram
  - Resources status
  - Thread status
  - Events table
  - Flame Chart
  - Memory usage
- Partial navigation

//TODO

- We need to make sure all the extras are handled.
- Better inter-view interactions
- Make view navigation consistent
- More user profiling (UX)
- Better file handling
- Flatten learning curve



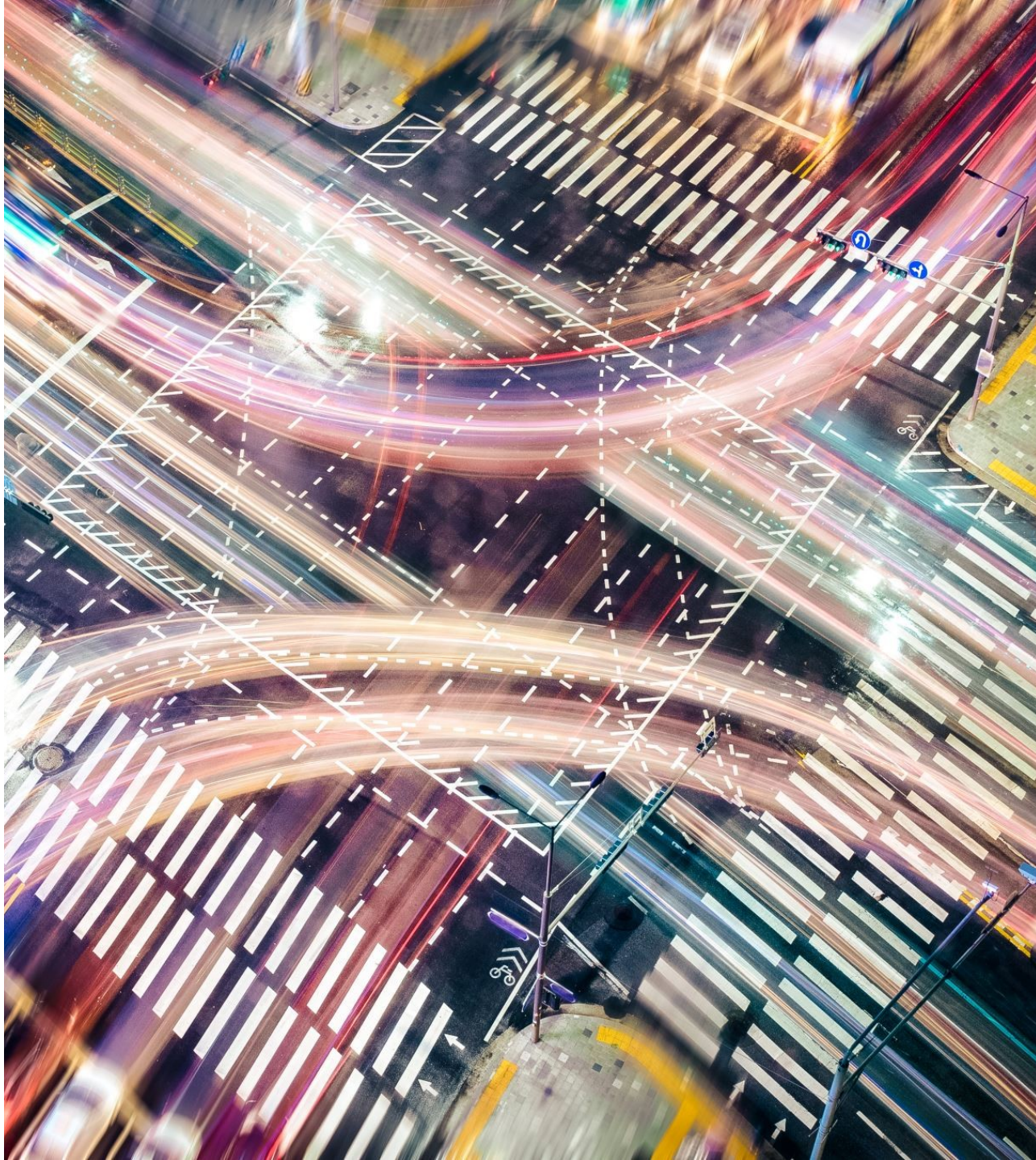
Future Work

Roadmap coming out.

The following slides are a TENTATIVE roadmap.

There is no guarantee.

The following is for illustrative purposes only



Road Map

- Be a good Ericsson Citizen - Ongoing
- Present to LTTng users (Ericsson/LTTng workshop) - Q3 2021
- Open Source Activity and Leadership (General) - Ongoing
- Trace Compass Releases (Eclipse) - Ongoing
- Open-Source Features to support internal implementation of trace viewer (Theia) - Ongoing
- Open Source Trace Compass Cloud MVP (LTTng) (lower priority) - Q4 2021

Trace Compass

- New Trace Compass Cloud Project – Q3 2021
- Linux Tracing use case – Q4 2021
- Internal Support – Ongoing
- Performance improvement – On demand

Theia Based Front End

- Open-Source Features to support of internal trace format (Theia) - Ongoing
- Open Source Trace Compass Cloud MVP (LTTng) (lower priority) - Q4 -2021
- UX Design – Participate in UX workshops - Q2 / Q4 2021
- UX Design – Limited Prototypes of UX – Q2 2021 (See later presentation)
- Common – Migration to VsCode plug-in - (Low Priority) Q3 2021
- Nano-second support for LTTng – Q3 2021
- Performance Improvements - Ongoing

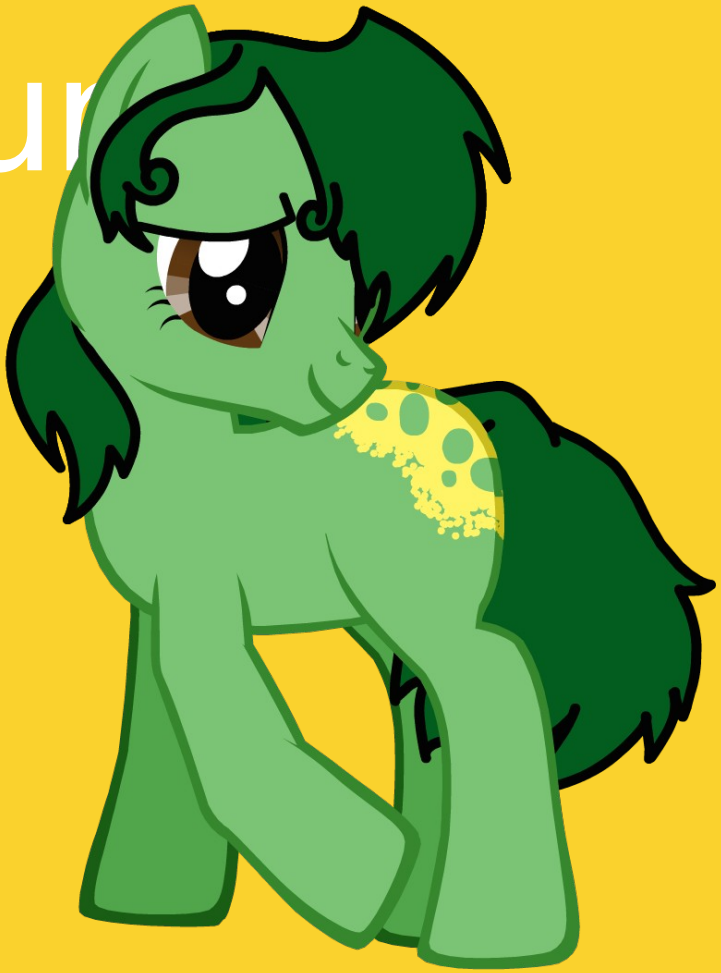
Command Line/Headless

- CLI-Headless: Support of investigations for internal solutions
- CLI-Headless: EASE support internal solution RCP
- CLI-Headless: Python Client to trace server

New Requested Mainline Linux Feature

New Requested Mainline Linux Featur

FTRACE



Artist: Deirdré Straughan

Where can you help?

Drop a review, a bug or a feature request

Fix a bug, especially if it helps you and is not in our interest

We have limited bandwidth, we would appreciate code contributions

Also: We have a couple of jobs we want to sponsor, either apply or send a friend.

Thank You

This Photo by Unknown author is licensed under CC BY-SA.

- Code Contributors
- Issue Reporters
- Designers
- Community Maintainers
- Enthusiasts
- Steak Holders