



ERICSSON

ECLIPSE TRACE COMPASS

PROGRESS REPORT

ERICSSON & POLYTECHNIQUE MONTREAL PARTNERSHIP

Matthew Khouzam | Ericsson Canada | July 2026

SINCE FEBRUARY

Five months of momentum

-  Trace Compass 12.0.0 released
-  CTF2 parsing completed
-  TMLL is now a full MCP app
-  ARM has joined the conversation

AGENDA

- Milestones & partnership value
- Recent deliverables
- TMLL as an MCP application
- Collaborating with ARM
- Team, and how you can help



ERICSSON

PROJECT MILESTONES

Research excellence through collaboration



ERICSSON

2026

- CTF2 format support delivered end-to-end
- AI-powered trace analysis via TMLL + MCP
- Linux perf.data support in the incubator

Note: MCP integration brings agentic analysis to Ericsson's tracing workflows



ERICSSON

2024

- Low-overhead GPU trace collection
- Virtualized GPU performance analysis
- Microservice tracing (Vnode)

Note: GPU tracing techniques apply to Ericsson's internal hardware accelerators



ERICSSON

2023

- Distributed critical path computation
- DPDK performance analysis
- ROS 2 message flow analysis

2022

- CPU-GPU visualization
- Anomaly detection with NLP
- Execution trace-based verification



ERICSSON

2021

- DDoS detection
- Integrated modeling tools
- Distributed + kernel tracing combination

2020 AND EARLIER

- Multi-level IoT intrusion detection
- VM execution flow analysis
- Lock contention analysis

Full publication list:
dorsal.polymtl.ca/fr/publications/

PARTNERSHIP VALUE

Ericsson is grateful for all the great contributions from Polytechnique Montreal and the DORSAL lab.

- **ROI:** For every \$1 invested, Ericsson receives over **10x return** in research value
- World-class research publications
- Cutting-edge tracing technology development
- Academic-industry collaboration model

THE DORSAL LAB



THE MOOSE LAB



CURRENT WORK ITEMS

-  CTF2 - Parsing complete!
-  TMLL MCP server - Shipped, iterating
-  Perf Support - Shipped, iterating
-  Power-aware tracing - In progress
-  Critical path in VS Code frontend - In progress
-  SQLite Trace Support - In progress

RECENT DELIVERABLES

February → July 2026

Highlights from the core, incubator, TMML and VS
Code repositories

CTF2 - COMPLETED

- Boolean and dynamic blob field classes added
- Support for CTF2 empty structs
- Removed sequence bit-length calculation (cannot be known)
- Tests for boolean and dynamic blob field classes
- LTTng: detect & validate CTF2 kernel and UST traces

KERNEL & CALLSTACK

- Kernel: thread status made to work again
- Callstack: multiple callstacks sharing the same TID
- Callstack: handle nestable instant events
- CTF2 kernel traces detected as kernel traces

INCUBATOR

- VM flow analysis added
- tsp.client settings & annotation fixes
- Released 0.18.0 on the 2026-06 (e4.40) target

Plus new Linux perf support (next slide)

- The screenshot shows the Flame Graph tool interface. The top section displays a flame graph with various colored bars representing different functions and their execution time. The bottom section shows a detailed view of the selected function, displaying its source code and stack frames.

Timestamp	Event type	PID	TID	CPU	IP	Address	Comm	Filename	Contents
1969-12-31 19:00:00.000 000 000	PERF_RECORD_79	-1	0			0x0000000000000000			raw@b01a309fe
1969-12-31 19:00:00.000 000 000	PERF_RECORD_09	-1	0			0x0000000000000000			raw@b0ee0e41a
1969-12-31 19:00:00.000 000 000	MMAP	-1	0			0xfffff8b1a00000	[kernel.kallsyms],text		pid=1, tid=0, addr=2120220672, len=42290112, pgoff=2120220672, filename=/kernel.kallsym
1969-12-31 19:00:00.000 000 000	MMAP	-1	0			0xfffffff0495000	/lib/modules/6.8.0-110-generic/kernel/crypto/cryptd.ko		pid=1, tid=0, addr=1068937216, len=24576, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAP	-1	0			0xfffffff0490000	/lib/modules/6.8.0-110-generic/kernel/drivers/pinctrl/intel/pinctrl-tigerlake.ko		pid=1, tid=0, addr=1068900352, len=28672, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAP	-1	0			0xfffffff04a0000	/lib/modules/6.8.0-110-generic/kernel/crypto/cryptd.ko		pid=1, tid=0, addr=106887584, len=16384, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAP	-1	0			0xfffffff04a0000	/lib/modules/6.8.0-110-generic/kernel/drivers/jzcd/zjcd-smbus.ko		pid=1, tid=0, addr=1068838912, len=16384, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAP	-1	0			0xfffffff04b0000	/lib/modules/6.8.0-110-generic/kernel/arch/x86/crpto/aesni-intel.ko		pid=1, tid=0, addr=106882624, len=356352, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAP	-1	0			0xfffffff0511000	/lib/modules/6.8.0-110-generic/kernel/drivers/platform/x86/wmi.ko		pid=1, tid=0, addr=1068429312, len=28672, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAAP	-1	0			0xfffffff0520000	/lib/modules/6.8.0-110-generic/kernel/drivers/tty/typec.ko		pid=1, tid=0, addr=106887584, len=16384, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAAP	-1	0			0xfffffff0520000	/lib/modules/6.8.0-110-generic/kernel/drivers/dma/fdmat4.ko		pid=1, tid=0, addr=1068322816, len=20480, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAAP	-1	0			0xfffffff0531000	/lib/modules/6.8.0-110-generic/kernel/drivers/nvme/common/nvme-auth.ko		pid=1, tid=0, addr=1068298240, len=28672, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAAP	-1	0			0xfffffff054b000	/lib/modules/6.8.0-110-generic/kernel/drivers/mfd/intel-psp.ko		pid=1, tid=0, addr=1068191744, len=12288, pgoff=0, filename=/lib/modules/6.8.0-110-gener
1969-12-31 19:00:00.000 000 000	MMAAP	-1	0			0xfffffff0553000	/lib/modules/6.8.0-110-generic/kernel/drivers/acpi/video.ko		pid=1, tid=0, addr=1068158976, len=7728, pgoff=0, filename=/lib/modules/6.8.0-110-gener

Histogram Properties Bookmarks

 - Event Density
 - Progress
 - Descriptive Statistics
 - Latency vs Time
 - Latency vs Count
 - Latency Table
 - Latency Statistics
 - Valkey Execution Timeline
 - Weighted Tree View (new Callstack)

Number of calls

 - 165526
 - 165526
 - /usr/bin/ping
 - 0xfffffff1efec00
 - 0xfffffff1efec00130
 - 0xfffffff1efec270c1
 - 0xfffffff1ea072bf
 - 0xfffffff1efecdc7
 - 0xfffffff1efead
 - 0xfffffff1efcab8
 - 0xfffffff1efcf79e
 - 0xfffffff1efcfcb



VS CODE TRACE EXTENSION

- Integrated a **rich cursor** across views
- Fixed tooltip pointer-event handling
- Dependency modernization (webpack, qs)

The rich cursor is the first step toward ARM's
dynamic-cursor vision

Arnaud is a **new committer** on the VS Code frontend
too. Congratulations!

RELEASE ENGINEERING

- **Version 12.0.0** released (11.3.0 as interim baseline)
- Updated to Eclipse **2026-06 platform (e4.40)**
- Build with e4.40 target by default
- Updated OOMPH setup, baselines and Maven plugins

TMLL

Now a full-fledged MCP application

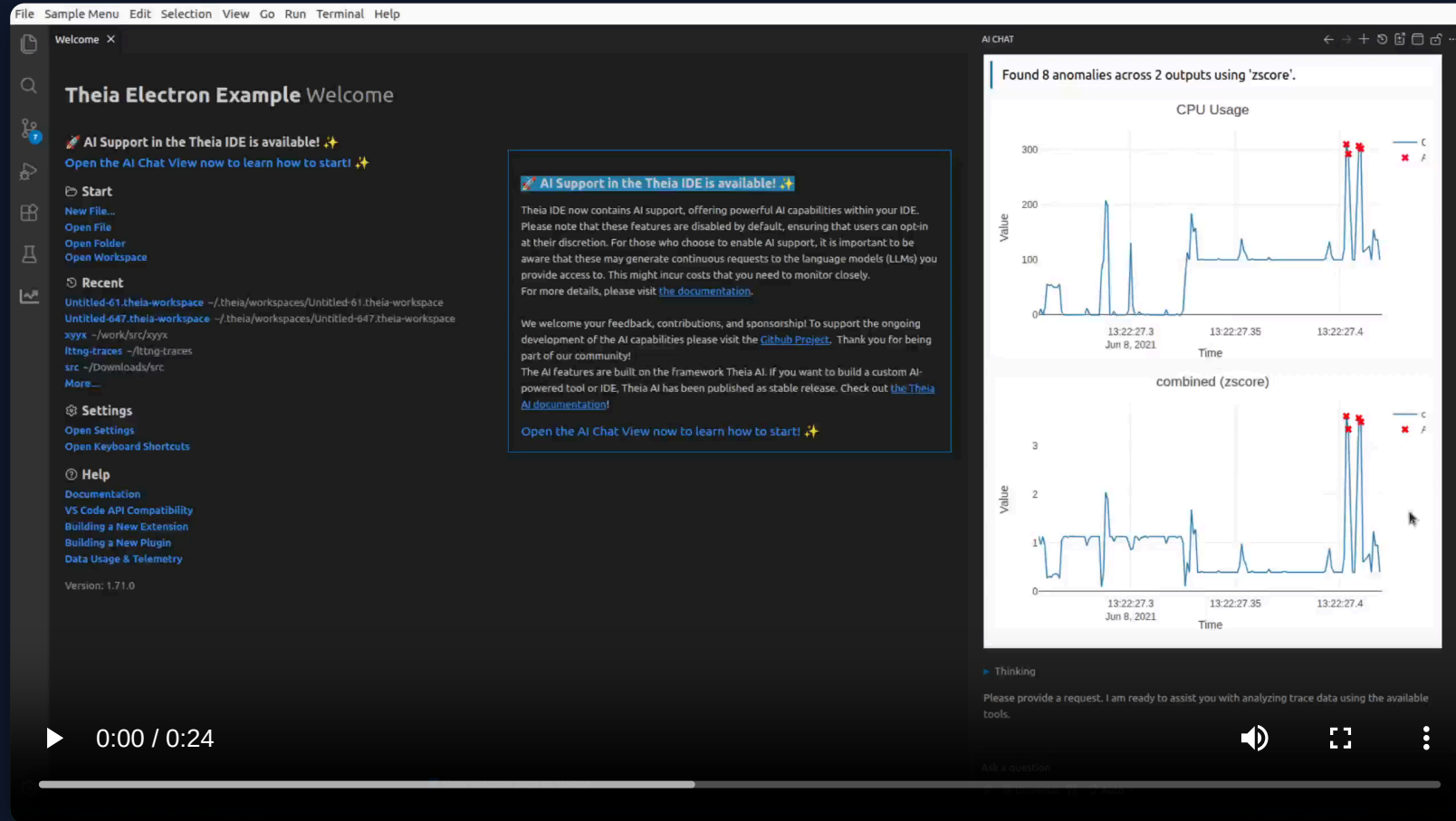
Trace Machine Learning Library, from notebooks to
agents



TRACES AS MCP RESOURCES

- Open experiments & traces are exposed as **MCP resources**
- Agents can browse and reason over live trace data
- Read-only tools marked **non-destructive** via ToolAnnotations
- Progressive discovery keeps the tool surface manageable

MCP IN ACTION



The screenshot displays the Theia IDE interface with a dark theme. On the left is a sidebar with a 'Welcome' tab and a list of recent workspaces. The main editor area shows a 'Welcome' message and a notification about AI support. On the right, an 'AI CHAT' panel is open, displaying two line graphs: 'CPU Usage' and 'combined (zscore)'. Both graphs show a sharp spike in value around 13:22:27.4 on June 8, 2021. Below the graphs, the AI chat interface shows a 'Thinking' status and a prompt: 'Please provide a request. I am ready to assist you with analyzing trace data using the available tools.'

File Sample Menu Edit Selection View Go Run Terminal Help

Welcome X

Theia Electron Example Welcome

AI Support in the Theia IDE is available! ✨
Open the AI Chat View now to learn how to start! ✨

Start

- New File...
- Open File
- Open Folder
- Open Workspace

Recent

- Untitled-61.theia-workspace ~/theia/workspaces/Untitled 61.theia-workspace
- Untitled-647.theia-workspace ~/theia/workspaces/Untitled-647.theia-workspace
- xyyx ~/work/src/xyyx
- ltnng-traces ~/ltnng-traces
- src ~/Downloads/src
- More...

Settings

- Open Settings
- Open Keyboard Shortcuts

Help

- Documentation
- VS Code API Compatibility
- Building a New Extension
- Building a New Plugin
- Data Usage & Telemetry

Version: 1.71.0

0:00 / 0:24

AI CHAT

Found 8 anomalies across 2 outputs using 'zscore'.

CPU Usage

Value

Time

combined (zscore)

Value

Time

Thinking

Please provide a request. I am ready to assist you with analyzing trace data using the available tools.

An agent driving TMLL over MCP, browsing traces, running analyses and plotting results live

ANALYSIS AT THE AGENT'S FINGERTIPS



- `plot_xy_with_anomalies` returns inline images & HTML
- `create-field-plots` generates & posts XML analyses via TSP
- Both a **CLI** and an **MCP** interface
- Robust stdio transport (logs redirected to stderr)

A trace becomes a first-class context an LLM can query, plot and explain



SKILLS VS MCP

Why a trace visualizer wants MCP, not a skill

- **Skills** teach the agent *how to read* a trace, reusable, static know-how baked into the prompt
- **MCP** gives the agent *access to the data itself*, live traces, experiments and analyses

Different jobs: one is knowledge, the other is a connection to the data

WHY MCP FITS HERE



ERICSSON

- Traces are **huge and dynamic**, they must be fetched at runtime, not pasted into a prompt
- The data lives in the **tracing backend (TSP)**; MCP bridges the agent to it
- Analysis is **interactive**, query, plot, drill down, which maps to MCP tools & resources
- A skill could describe *how* to interpret a flame chart, but only MCP delivers the actual flame chart data

MCP = access to the data · Skills = the know-how to read it

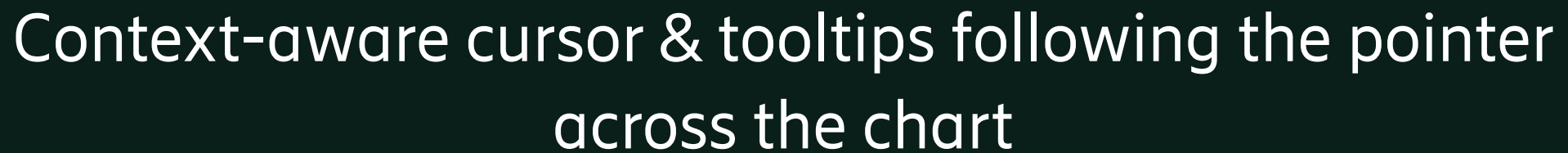
A NEW COLLABORATOR: ARM

There is interest from **ARM**, and we are working with them.

FEATURES ARM SUGGESTED

-  **Dynamic cursor**, smarter, context-aware navigation
-  **Mipmapped zooms**, smooth multi-resolution zooming over huge traces
-  **And more** to come as the collaboration grows

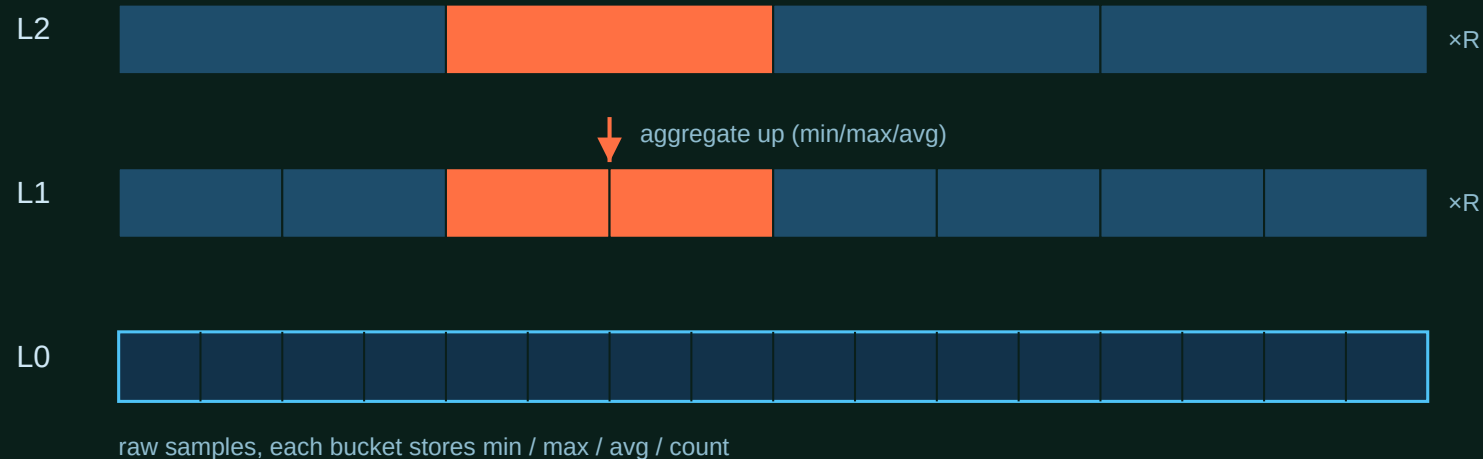
Great features from a great partner, the rich cursor work is already a foot in the door



MIPMAPPED XY PLOTS

Smooth zooming over billions of points

Dormant in Trace Compass for 10 years. It is about to be activated.





MIPMAP ALGORITHM: BUILD

- **Aggregate:** as events stream in, fold every R points into one parent bucket storing *min / max / average / count*
- **Stack:** repeat over each level to form a pyramid of $\log_n(N)$ levels, built in a single pass

min/max per bucket preserves spikes; average keeps the trend, no visual aliasing when zoomed out

MIPMAP ALGORITHM: QUERY

- **Pick a level:** for a viewport of W pixels, choose the level where one bucket $\approx$ one pixel
- **Draw:** read only $\sim W$ pre-aggregated buckets instead of scanning all N points
- **Result:** zoom/pan cost is $O(W \cdot \log N)$, independent of trace size

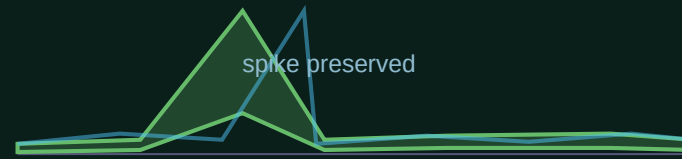
Billions of samples, but only a screenful of buckets
ever touched per frame

SPIKES: KEPT, NOT GUESSED

Naive downsampling



Mipmap min / max envelope



- Downsampling makes spikes **appear & disappear** as you pan/zoom
- Mipmap min/max keeps every spike, **investigate them, and analyze faster**

WHY IT MATTERS

- Broadens the industrial base behind Trace Compass
- Pushes performance for very large traces
- More collaborators → more sustainability for the project

CODERABBIT INTEGRATION

Free AI-powered code reviews for FOSS projects

- Accelerating review process
- Tightening feedback loop with patches
- **For students:** Commit early and get free advice
- Automated code quality insights

TEAM UPDATE

Where the project stands today

ERICSSON TEAM



- Matthew Khouzam
- Project lead & Principal Researcher

RENESAS



- Renesas continues to help out
- We are grateful for their contribution

ARM

arm

A new player has entered the chat

- **ARM** is now engaged and contributing ideas
- Dynamic cursor, mipmapped zooms, and more



ERICSSON

POLYTECHNIQUE MONTREAL



- **16 years** of great work together
- The DORSAL lab: research that keeps Trace Compass ahead
- The foundation this whole collaboration is built on

WELCOME & THANK YOU, HIBA

Hiba has started contributing to Trace Compass,
beginning with TMLL.

Thank you for jumping in, we are glad to have you on
the project!

CRITICAL NEED

We **NEED** contributions from students for the survival and thriving of this project.

- Project continuity depends on student involvement
- Research opportunities available
- Real-world impact on production systems
- Open source contribution experience

MOVING FORWARD

- Seeking active student contributors
- Maintaining research excellence
- Growing the industrial collaboration (Renesas + ARM)
- Building the next generation of tracing & AI tools

Together we can ensure the project thrives.

QUESTIONS?

Discussion and feedback welcome

<https://tracecompass.org>

Our YouTube channel: [@EclipseTraceCompass](#)

Contact: [matthew.khouzam<at>ericsson<dot>com](mailto:matthew.khouzam@ericsson.com)

REPOSITORY LINKS



ERICSSON

TRACE COMPASS CORE + TMLL

github.com/eclipse-tracecompass/org.eclipse.tracecompass



REPOSITORY LINKS



ERICSSON

INCUBATOR

github.com/eclipse-tracecompass-incubator/org.eclipse.tracecompass.incubator/



REPOSITORY LINKS



VIEWER + TSP

github.com/eclipse-cdt-cloud/vscode-trace-extension

