



Tracing and Monitoring Framework

Linux Symposium / Tracing Summit
Montreal, July 14, 2009

François Chouinard
Ericsson Canada



Summary

- Background
- Project Scope
- Framework Overview
- Framework Structure
- Exemplary Tool Integration (LTTng)
- Screenshots
- Demo



Background

- **Open-Source IDE initiative**
 - Full-fledged, C/C++ development environment
 - State-of-the-art tool suite
 - Open-source
 - Eclipse integration

- **Tracing and Monitoring component**
 - Facilitate the integration of tracing tools
 - Provide out-of-the-box “common” functionalities
 - Hosted by Eclipse Technology / Linux Tools



Project Scope

- **Extendable support for:**
 - Tools discovery
 - Tools control
 - Trace/data retrieval and storage
 - Trace/data visualization
 - Analysis/correlation/comparison/... modules integration

- **Additional features:**
 - Local and remote tools
 - Live and concurrent trace streams
 - Asynchronous events
 - Traces/logs that exceed available memory
 - External, host-based, libraries and analysis tools
 - Custom trace/log parsers



Tool Discovery

- Purpose
 - Identify the available trace providers and their capabilities
 - This information is used to generically control the tools

- Features
 - Discovery of available log providers
 - Discovery of log provider capabilities
 - Integration scheme for existing monitoring tools
 - Support for local and remote tools



Tool Control

- Purpose
 - Control the tool operation
 - Manage the resources allocated to tracing

- Features
 - Basic tool control (conf/start/stop/pause/resume/...)
 - Generic trace triggering, filtering
 - Tracing rate regulation (throttling)
 - ❖ To avoid congestion on the target, host, transport link, ...
 - Budget policy (per trace, trace type, ...)
 - ❖ To constrain target resource usage (CPU, memory, bandwidth)
 - Control settings persistence



Data Retrieval and Storage

- Purpose
 - Collect and store tracing/monitoring data
 - Generic trace/log data interface (for the analysis tools)

- Features
 - Collect monitoring data from the tool
 - ❖ File transfer
 - ❖ Continuous stream
 - ❖ Multiple, heterogeneous streams
 - Provide a generic log file interface
 - ❖ Support for log-specific parsers
 - ❖ Support for sequential, random access, checkpoints, DB, ...
 - ❖ Support for large files (bigger than available memory)



Data Visualization

- Purpose
 - Provide a set of standard data visualization tools
 - Toolbox of widgets (trace agnostic)

- Features
 - Provide generic monitoring views
 - ❖ Event logs (raw, tabular)
 - ❖ Time Line, Sequence Diagram, Logic Analyser, Gantt Chart
 - ❖ CPU/Memory/Heap/Network usage
 - ❖ Search filters, pattern matching, saved search queries, ...
 - Provide generic graphical widgets
 - ❖ Charts, Histograms, ...
 - Extendable for application-specific contents

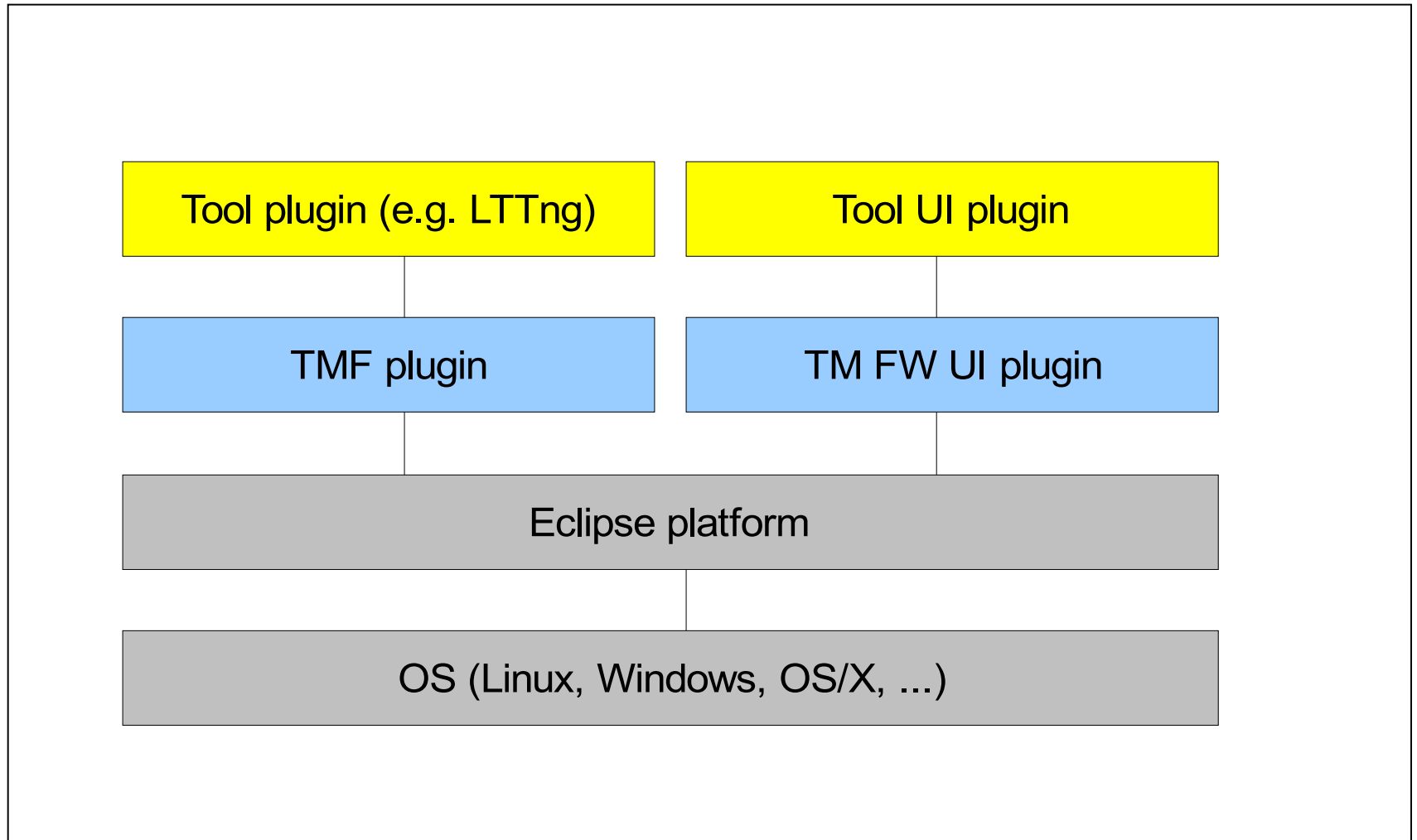


Analysis Tools Integration

- Purpose
 - Provide basic analysis functions
 - Support host-based, external analysis tools and libraries

- Features
 - Log comparison (regression testing, health monitoring, performance analysis,...)
 - Causal dependency analysis
 - ❖ Event Dependency Tree
 - ❖ Critical Path
 - ❖ Correlation of event data
 - ❖ Reconstruction of event sequences from related traces
 - ❖ Execution replay
 - External tools integration
 - ❖ Scheme to access the tracing data generically
 - ❖ Scheme to send the analysis results to UI views/widgets

Framework Structure



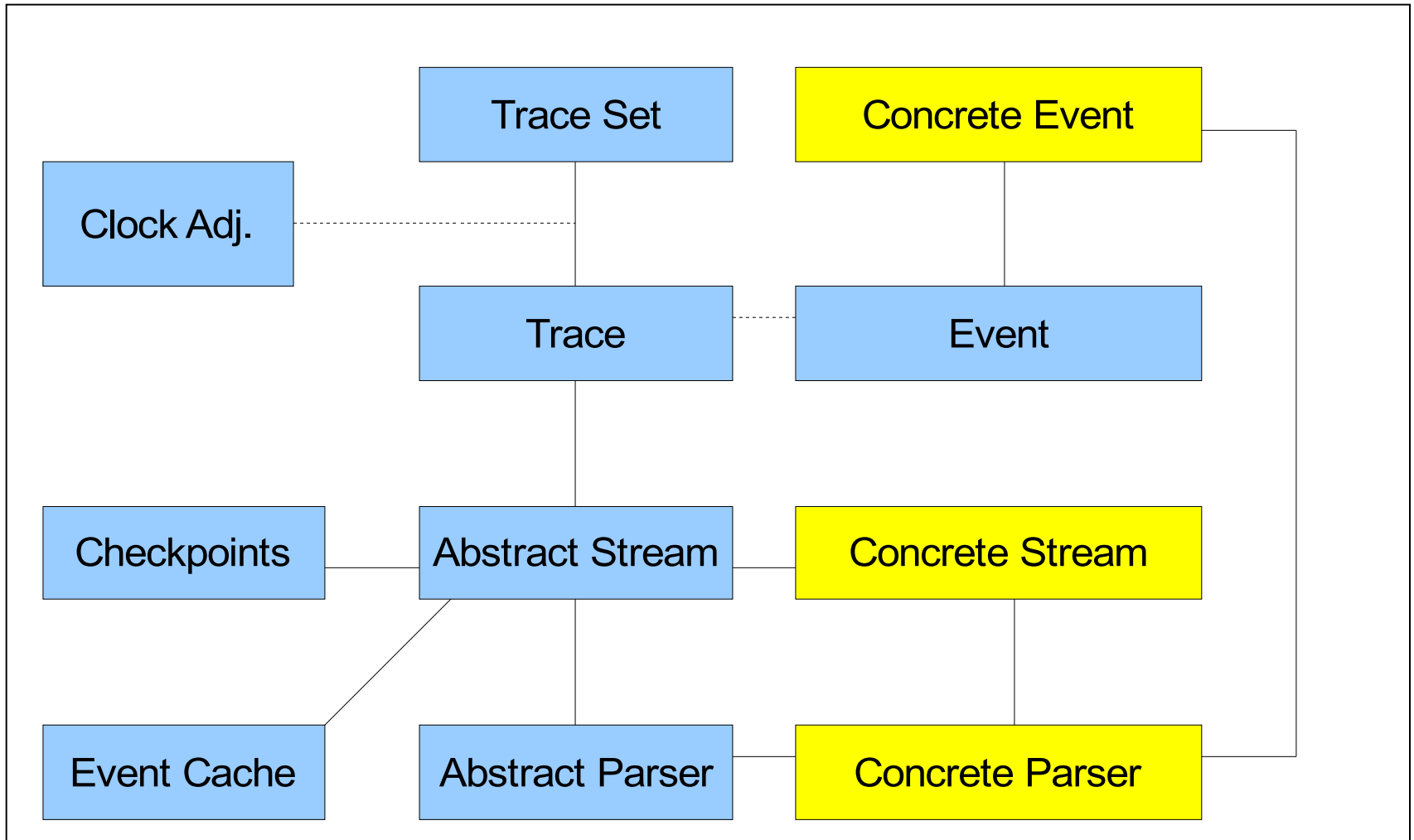


Framework Structure

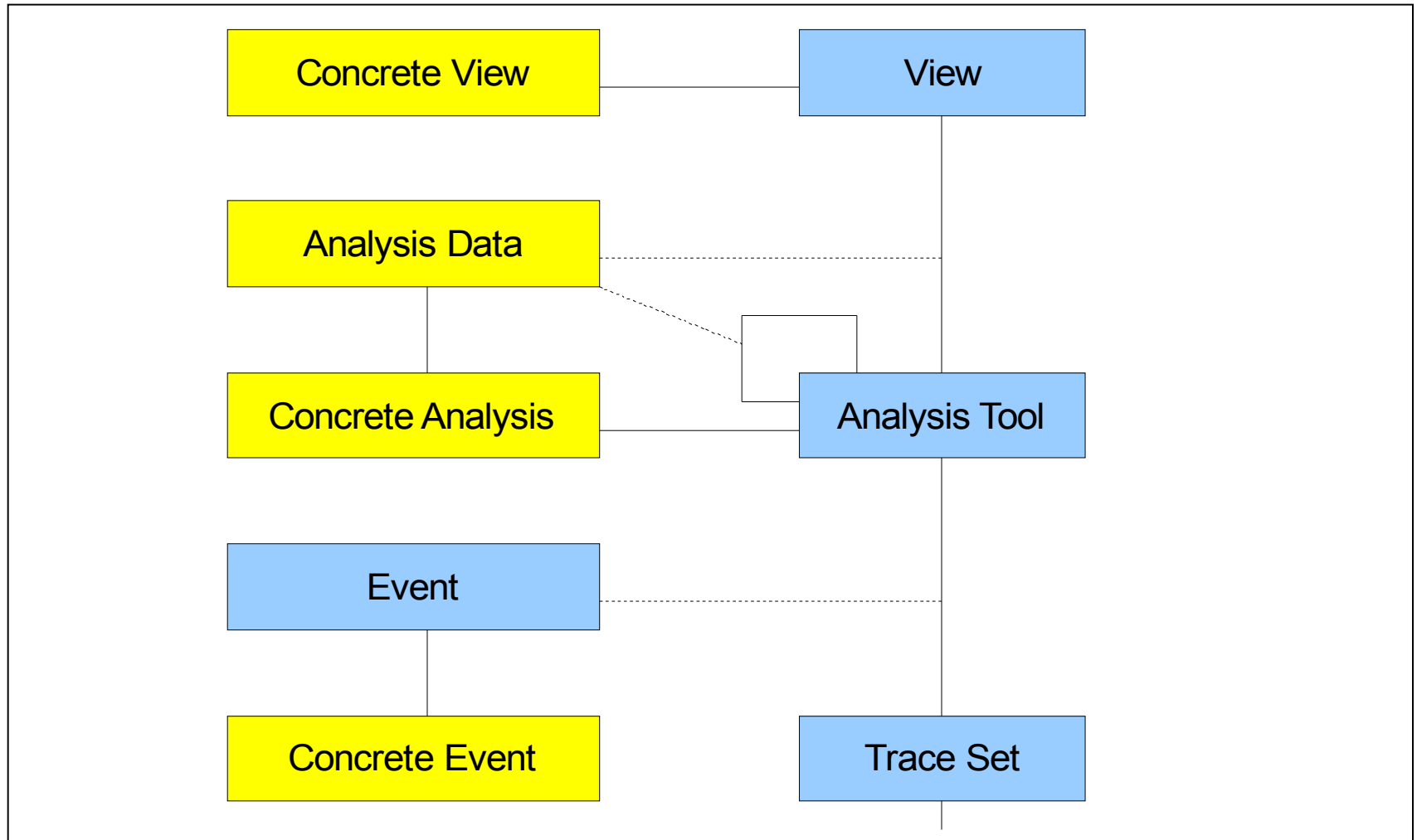
- Key Concept: The Event
 - Basic Event
 - ❖ Timestamp
 - ❖ Source
 - ❖ Type (→ format)
 - ❖ Content (→ fields)
 - Extended for application-specific events
 - Possibility to handle derived/synthetic events

- Benefit:
 - Allows the handling of events using the framework generic components and APIs

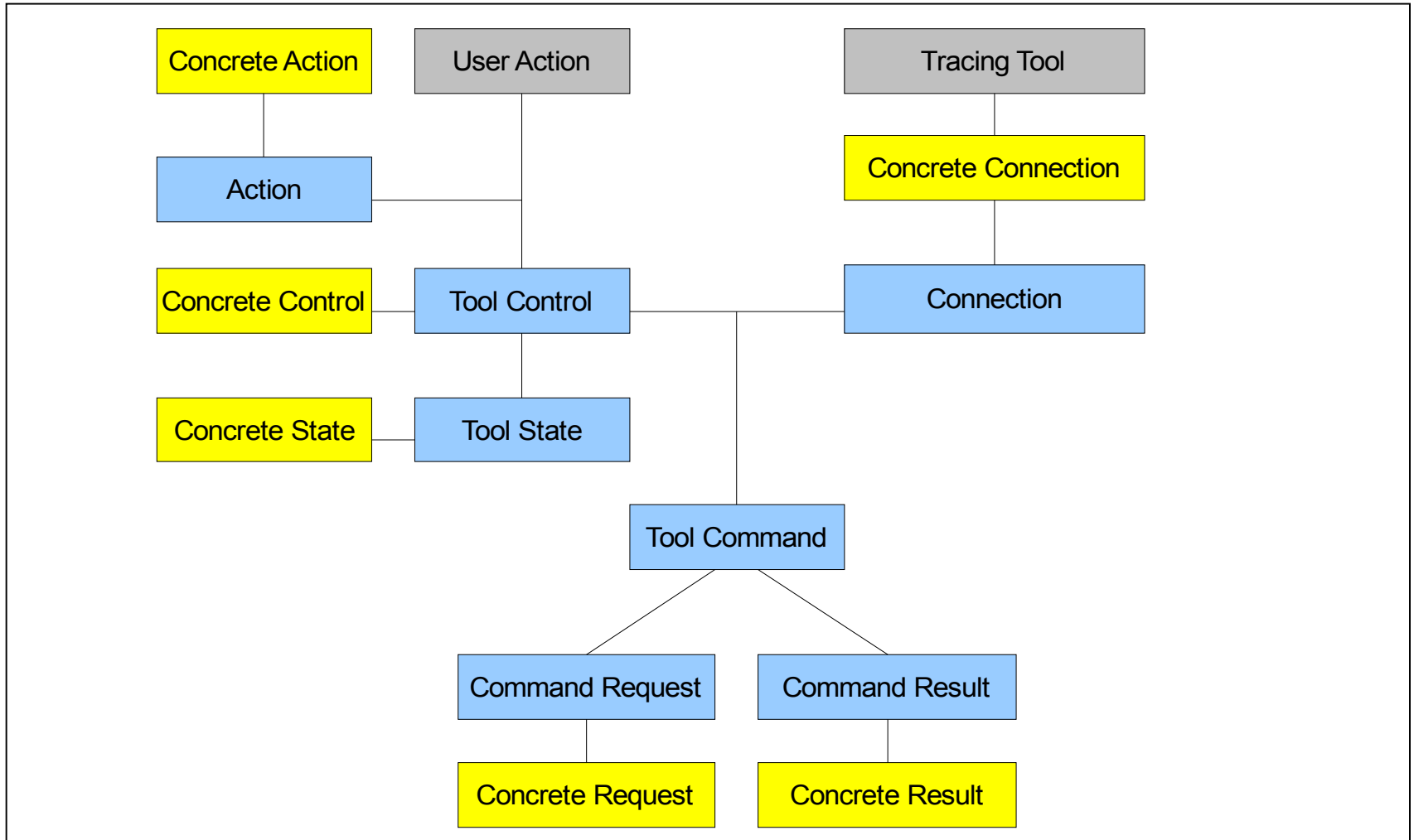
Framework Structure



Framework Structure



Framework Structure

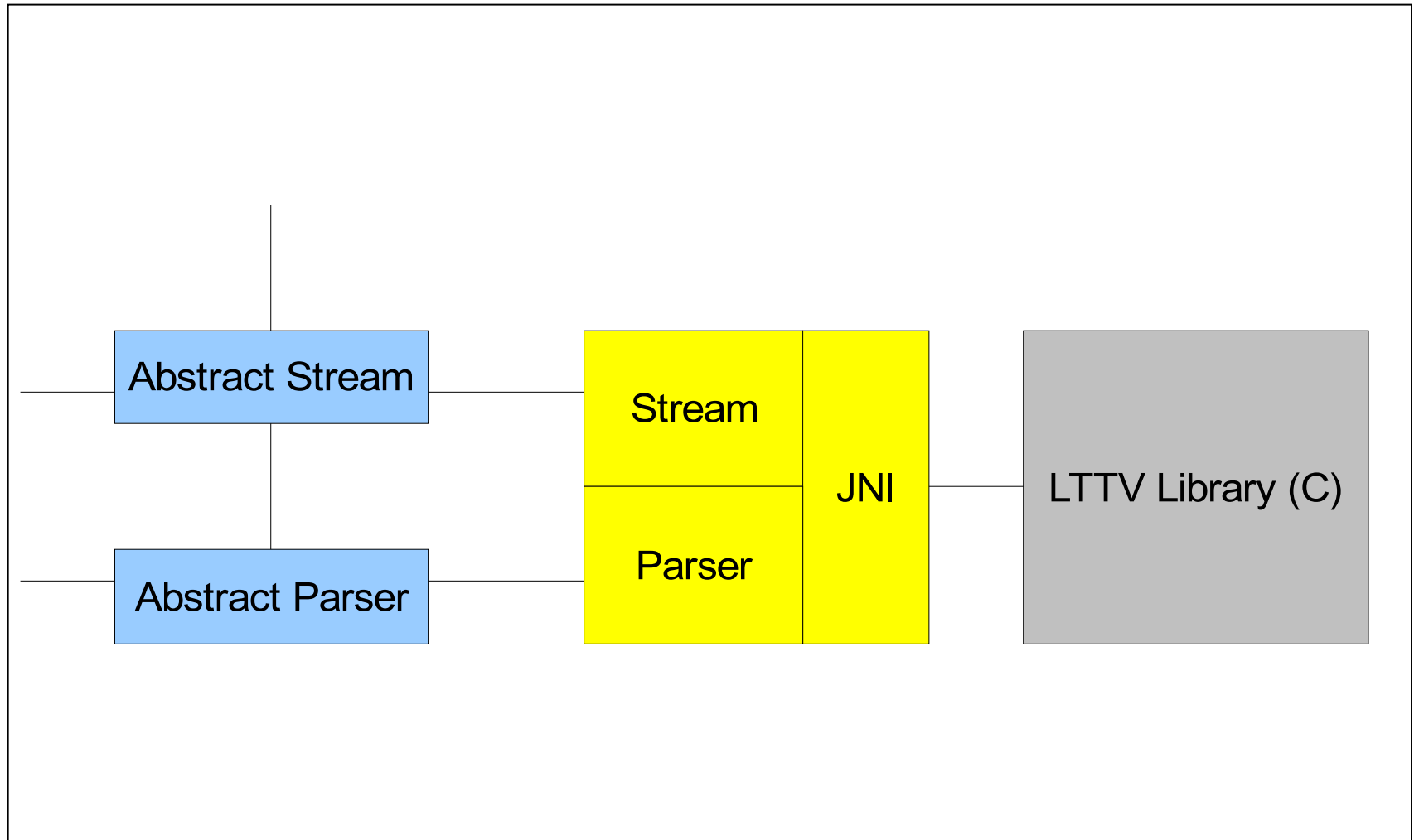




LTTng Integration

- LTTng Perspective
 - Project View
 - Control View
 - Time Frame View
 - Statistics View
 - Events Table View
 - Control Flow View
 - Resources View
 - Histogram View
- LTTng Control (remote and local)
 - Probe configuration
 - Start, stop, pause, resume
 - Trace retrieval
- Framework Models
 - Event Model
 - Event Log (Trace) Model
 - Request Model
 - View Model
 - Control Model
- Framework Components
 - Generic Events Table View
 - Widgets Toolbox
 - Support for very large trace files
 - Support for non-java parsers
 - Support for analysis components

External Library Integration



Screenshots: LTTV



The screenshot displays the Linux Trace Toolkit Viewer (LTTV) interface. The main window is titled "Linux Trace Toolkit Viewer" and contains several panels:

- Traceset:** A tree view on the left showing the hierarchy of traces. The "events" trace is selected, and its statistics are shown in the right pane: irq_entry: 2021, timer_update_time: 1546, softirq_raise: 1925, sched_try_wakeup: 433, irq_exit: 276, send_signal: 278.
- Resource:** A timeline view showing the execution of various resources. The "Trace 0" resource is expanded, showing CPU0, IRQ 14 [irq 14], IRQ 239 [irq 239], SOFTIRQ 1, and SOFTIRQ 4.
- Process:** A table listing the processes being traced. The processes shown are: /usr/lib/policykit/polkit-read-auth-helper (PID 8998), /usr/lib/ConsoleKit/run-session.d/pam-foreground-compat.ck (PIDs 9014, 9015, 9016, 9017, 9018), and /usr/bin/cut (PID 9014).
- Trace:** A table showing the individual trace events. The events listed are: mm.page_alloc (PIDs 9014, 9014, 9014, 9014, 9014) and mm.page_free (PID 9014).

The bottom of the window shows the "Time Frame" controls, including start and end time (701s 631306766ns to 702s 631306766ns), Time Interval (1s 0ns), and Current Time (701s 945032256ns).

Screenshots: LTTng Perspective



The screenshot displays the LTTng Eclipse Platform interface. The main window shows a control flow view with a timeline from 0:000:000:500 to 0:000:002:000. A tooltip is visible over the Source-6 process, showing details such as Process Name (Source-6), Class Name, State (Type-3), Test Tip2 (Test Value tip2), Test Tip1 (Test Value tip1), Start Time (00:00:000.001.261), and Duration (00:00:000.000.091).

Timestamp	Source	Type	Reference	Content
952.090116049	Kernel Core	kernel/0/printk	/home/francois/Work:	ip : 0xc03ee88c
952.092222957	Kernel Core	kernel/0/vprintk	/home/francois/Work:	loglevel : 7 string : LTT state dump begin ip : 0xc04f402c
952.102730748	Kernel Core	kernel/0/irq_entry	/home/francois/Work:	ip : 3222386054 handler : 0xc0104528 irq_id : 0 kernel_mode : 1
952.103653406	Kernel Core	kernel/0/timer_update_time	/home/francois/Work:	jiffies : 4295126951 xtime_sec : 1243865376 xtime_nsec : 710000064 walltomonotonic_sec : -1243864439
952.105639911	Kernel Core	kernel/0/softirq_raise	/home/francois/Work:	softirq_id : 1
952.106503816	Kernel Core	kernel/0/softirq_raise	/home/francois/Work:	softirq_id : 8
952.107331649	Kernel Core	kernel/0/sched_try_wakeup	/home/francois/Work:	pid : 1856 cpu_id : 0 state : 1
952.107362024	Kernel Core	kernel/0/sched_try_wakeup	/home/francois/Work:	pid : 2731 cpu_id : 0 state : 1

Time Frame settings:

Start Time	End Time	Interval	Current Time
952 sec 12140052 ns	960 sec 12140052 ns	8 sec 0 ns	952 sec 12140052 ns

Demo!





References

- **Linux Tools Project**
<http://www.eclipse.org/linuxtools>
- **LTTng Integration**
<http://www.eclipse.org/linuxtools/projectPages/ltnng>
- **Linux Tracing Toolkit (LTTng)**
<http://ltnng.org>



Contacts

- ❖ François Chouinard (TM Framework, LTTng)
francois.chouinard@ericsson.com

- ❖ Dominique Toupin (Open-Source IDE)
dominique.toupin@ericsson.com

Questions?



ERICSSON 
TAKING YOU FORWARD