



RTAI

LSM – July 9, 2004

Philippe Gerum, rpm@openwide.fr

The RTAI project

- Founded 1999 – Prof. Paolo Mantegazza
 - Long-time contributor to RTLinux
- Politecnico di Milano (DIAPM)
 - <http://www.aero.polimi.it/~rtai/>
- International contributors
 - Originally based in the USA
 - Now mainly based in Europe
- Large user base



Original technical path

- Real-time hardware abstraction layer
- SMP support
- Comprehensive real-time API
- Hard real-time support in user-space
- First Adeos supporter

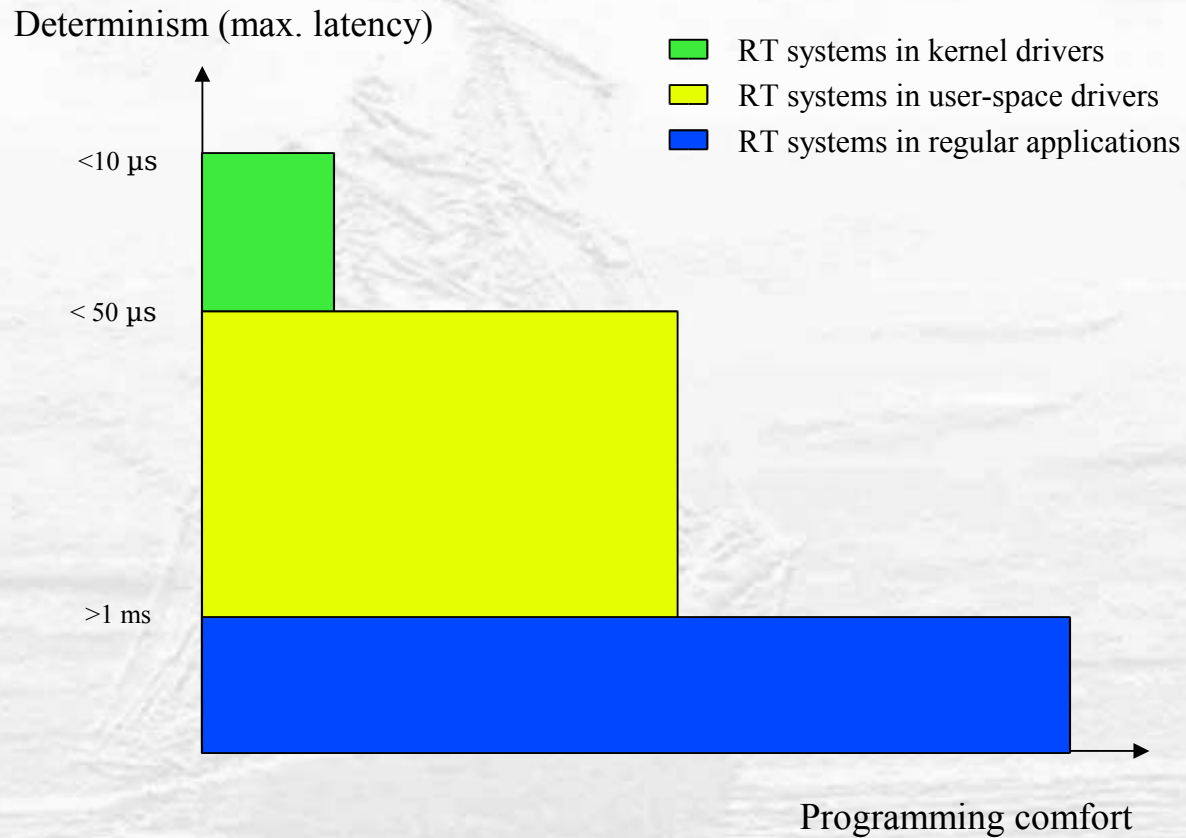
Major project milestones

- 1.x series for Linux 2.2 (1999)
 - LXRT x86, hard RT support in user-space
- 24.1.x series for Linux 2.4 (2000)
 - Adeos/x86 conversion (March 2003)
- 3.0 series (Dec. 2003)
 - Major refactoring
 - Merger with the Xenomai project
- 3.1 series for Linux 2.6 (April 2004)

Future direction

- RTAI/fusion
 - Linux 2.6 and above
 - Portable to any Adeos-enabled architecture
 - Pervasive hard RT support in user-space
 - Nanokernel technology
 - Traditional RTOS emulation
- v0.2 released, July 2004
 - v0.3 SMP due by August

Real-time in user-space: reality check



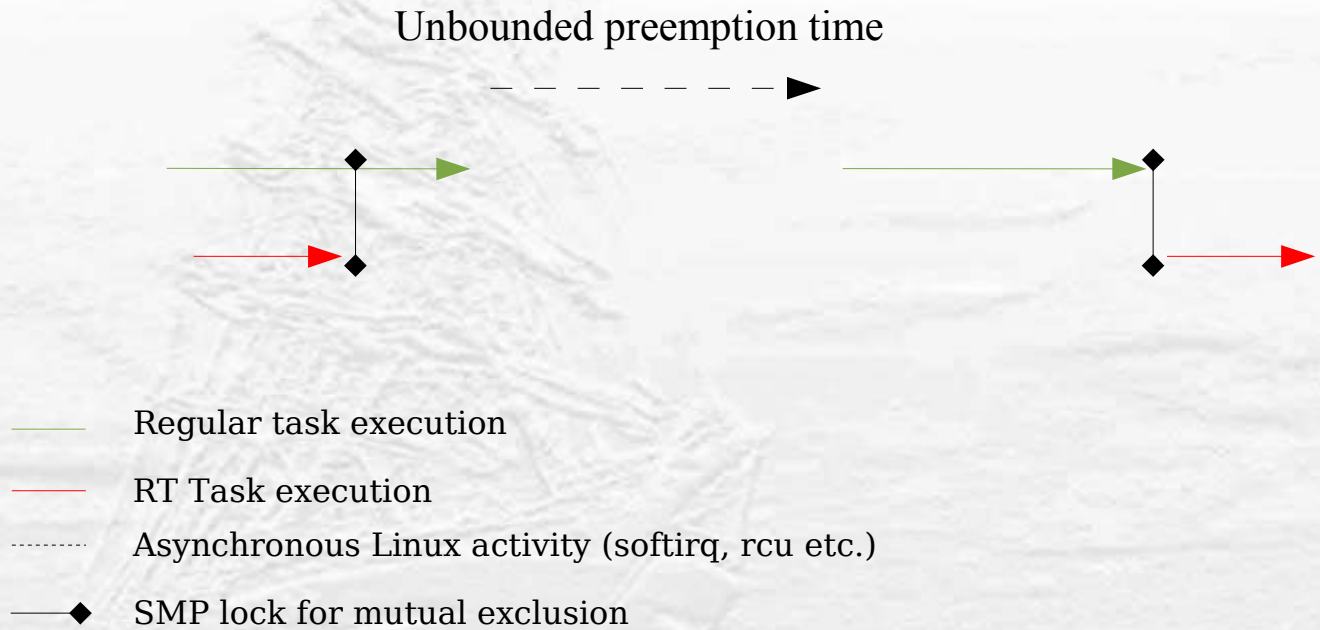


Vanilla Linux RT features: reality check

- High-resolution timers
 - Do not address the interrupt latency issue
- Preemptable kernel
 - Not fine-grained enough
 - Co-exists with a throughput-oriented design



Typical priority inversion



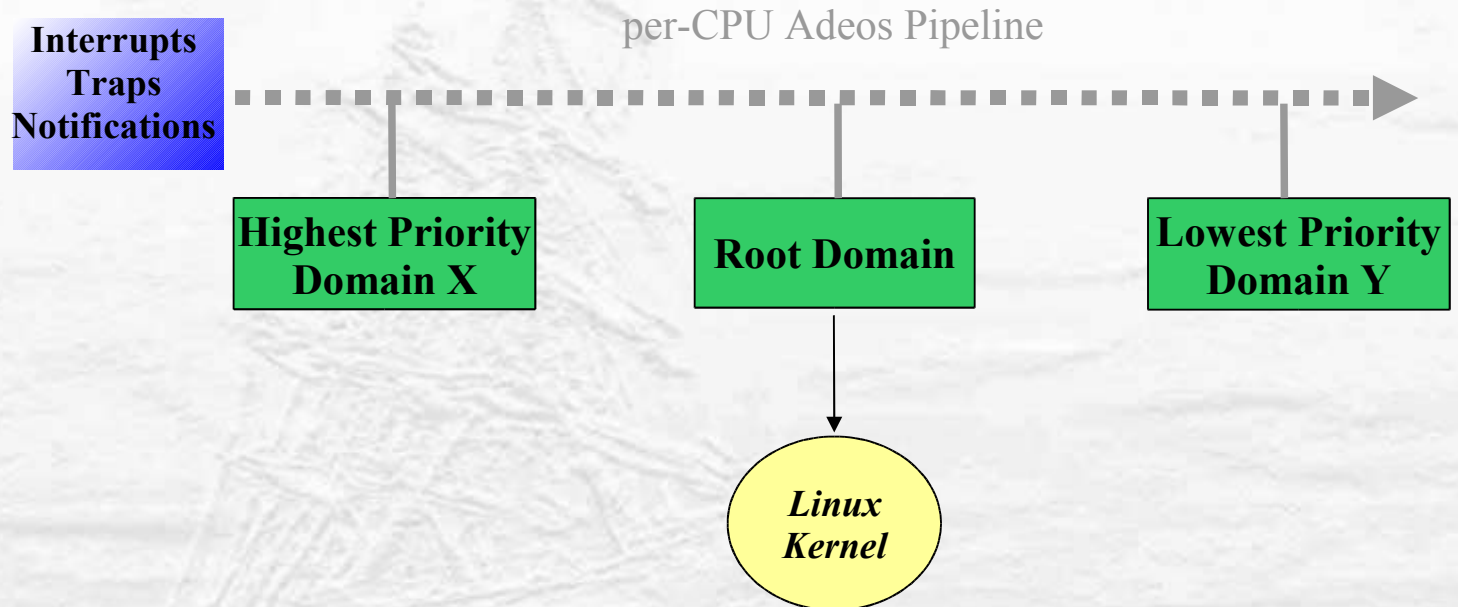
Real-time in user-space: steps

- Obtain full interrupt preemptability
 - Virtualize Linux interrupt handling
- Ensure bounded dispatch latency
 - Prevent priority inversions
- Cooperate with Linux
 - Extend the native RT support
- Ease migration
 - Provide compatibility with traditional RTOS

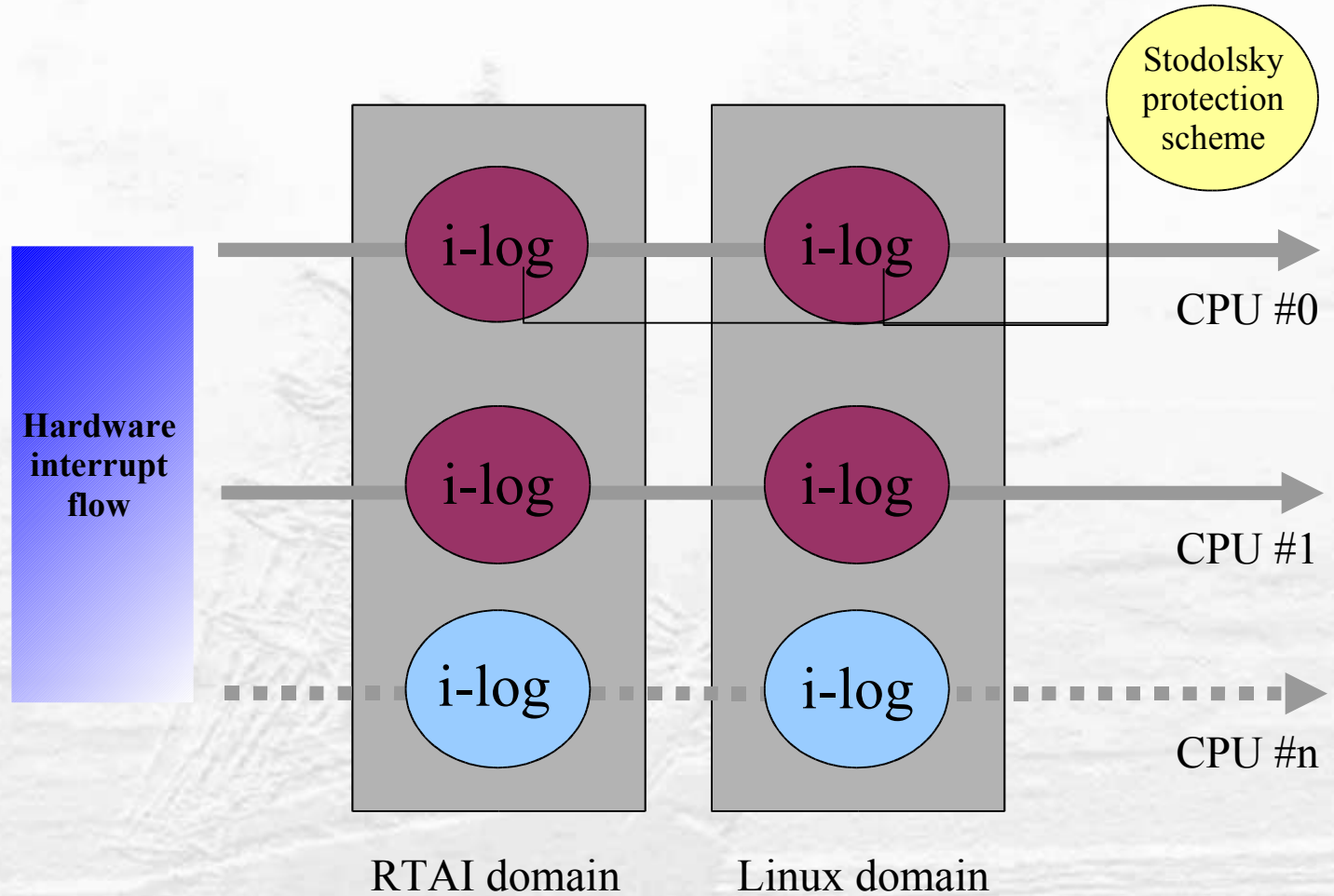
Converging technologies

- Obtain full interrupt preemptability
 - Adeos
- Ensure bounded dispatch latency
- Cooperate with Linux
 - Extended LXRT-type support over Xenomai
- Ease migration
 - Xenomai-based emulators

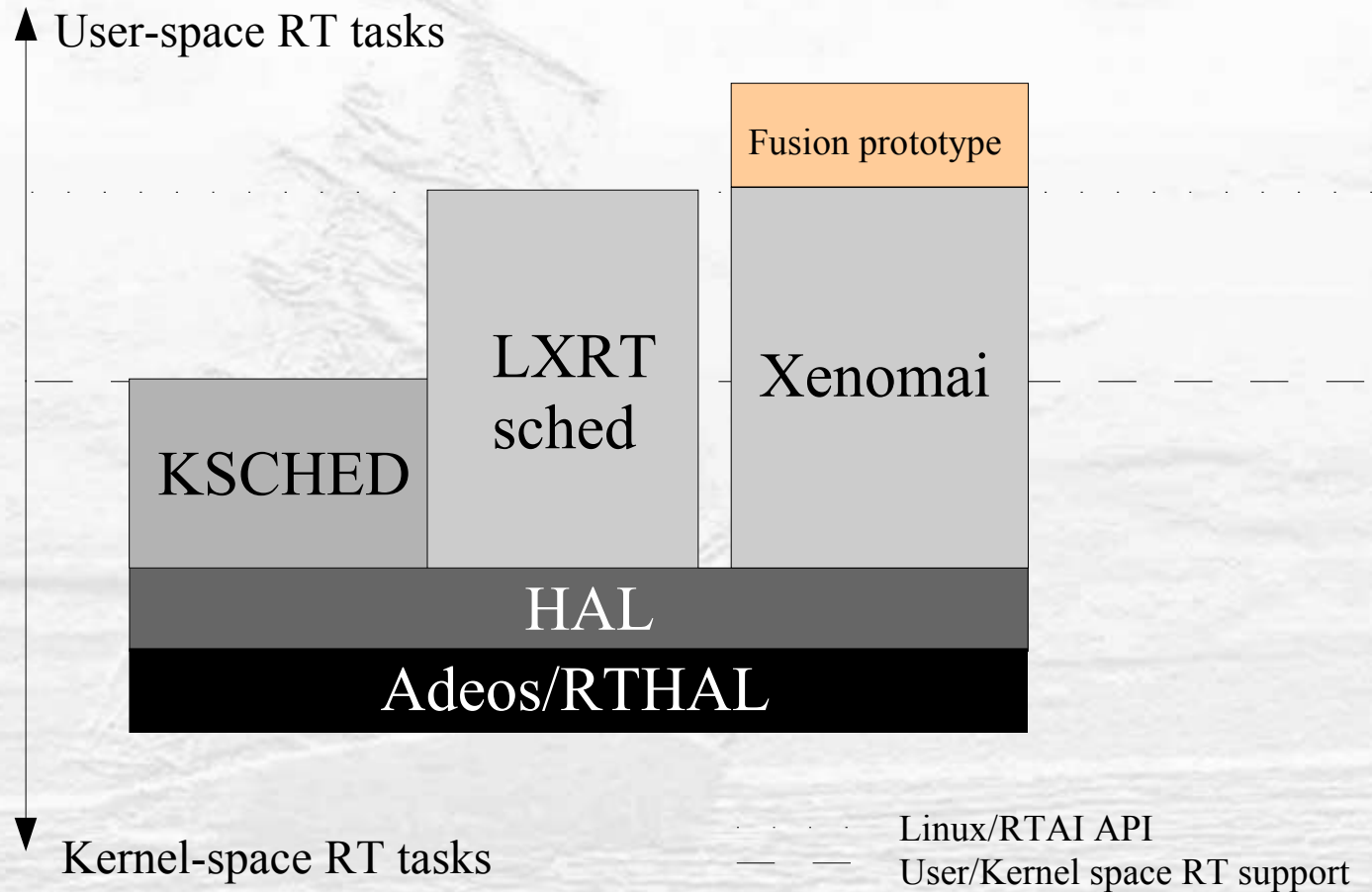
Adeos pipeline



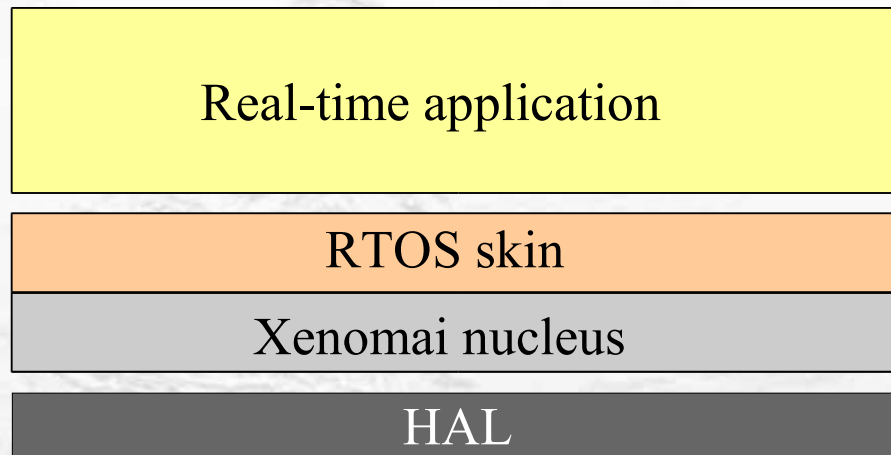
Optimistic interrupt protection





Functional layers in RTAI 3.x



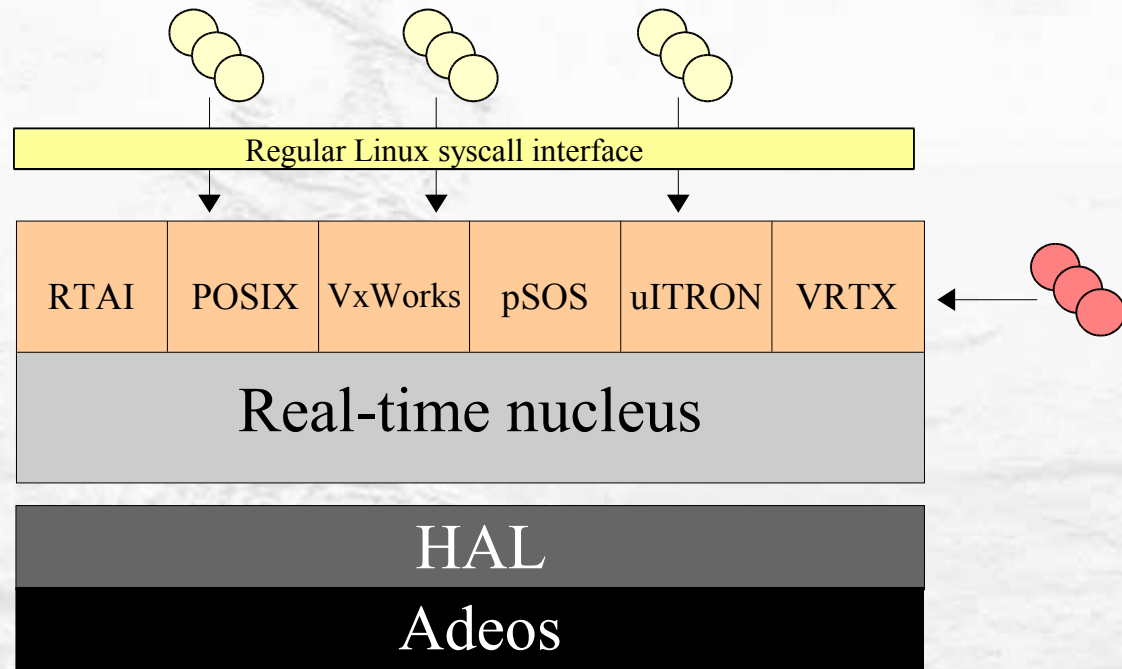
Xenomai-based architecture



Adeos

-  API layer (RTAI, pSOS+, VRTX, VxWorks, uITRON ...)
-  RTOS abstraction layer

Functional layers in RTAI 4.x



User-space applications

Kernel-based applications

Going beyond LXRT

- One nucleus, many interfaces
 - Abstract the common real-time objects
- Two real-time domains
 - Prevent priority inversions in Linux
 - Keep ultra-low latencies in RTAI
- Portable by design
 - Normalized architecture-dependent layer

Applications

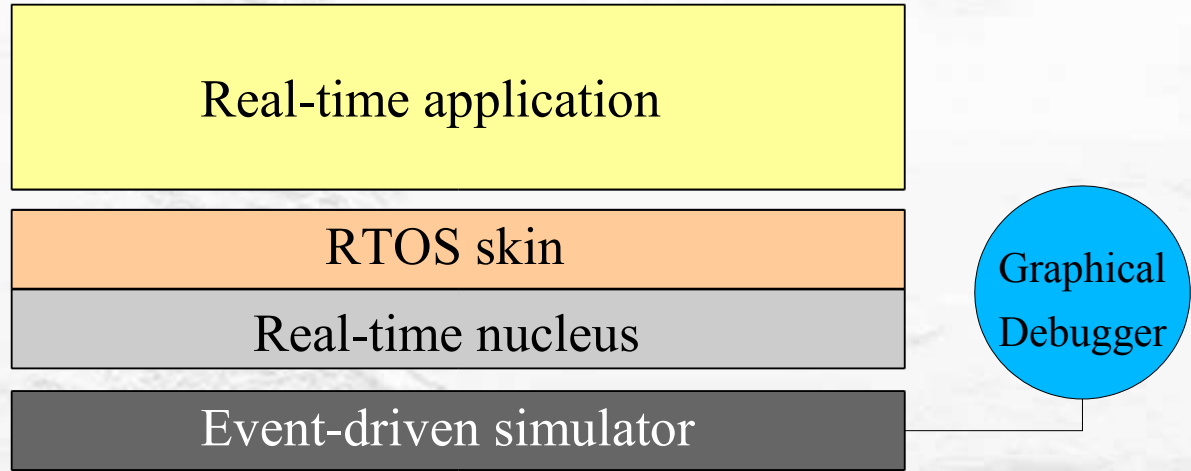
- Complete RT systems under MMU control
 - High determinism
 - Standard programming environment
 - HYADES (<http://www.hyades-itea.org/>)
- Mixed RT systems
 - Kernel-based hw management
 - Main implementation in user-space
 - Deterministic IPC



Tools

- GDB support (planned)
- LTT support (planned)
 - Combo kernel patches (Adeos + LTT)
 - Nucleus instrumentation
- Complete simulation system
 - Event-driven virtual machine
 - Graphical RTOS-aware debugger



Simulation engine



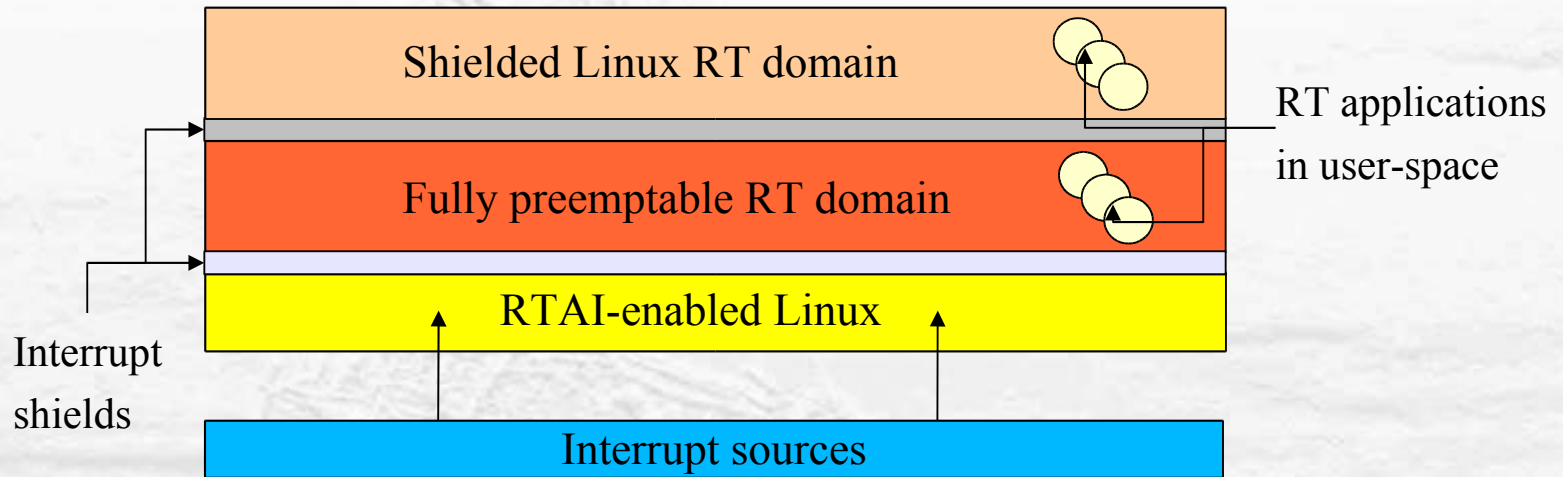
-  API layer (RTAI, pSOS+, VRTX, VxWorks, uITRON ...)
-  RTOS abstraction layer



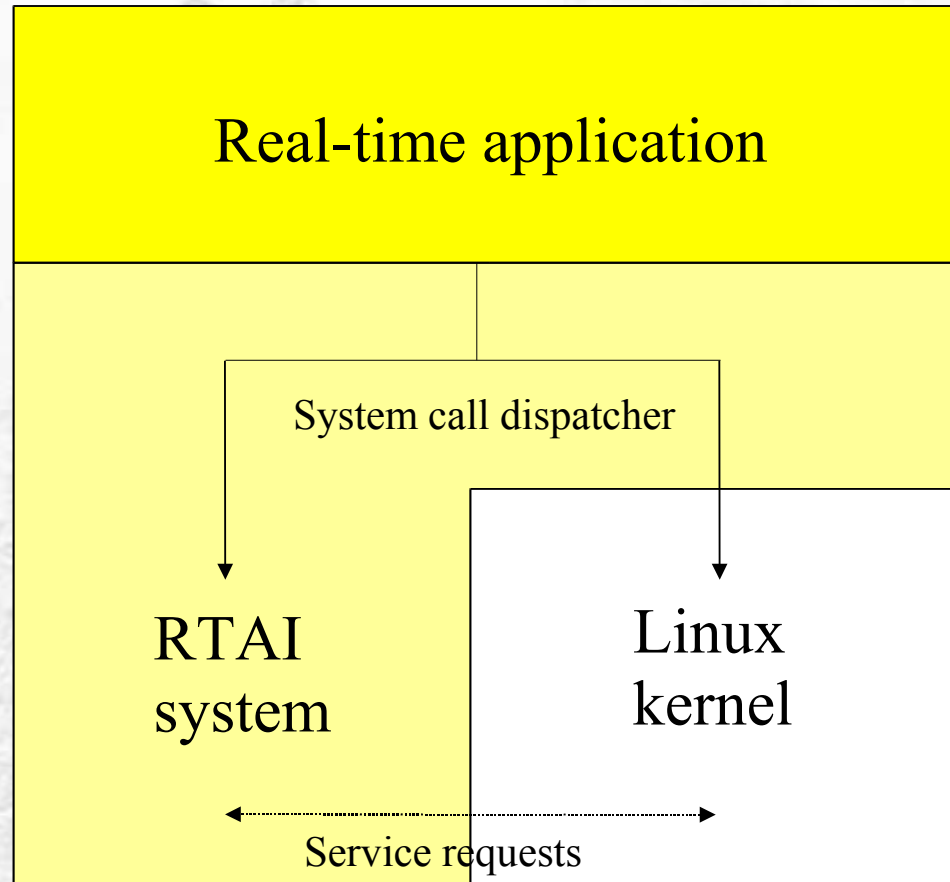
Real-time domains

- Ultra-low latency RT domain
 - Micro-second level timers
 - Ultra-low dispatch latency
 - Controlled by RTAI
- Shielded Linux RT domain
 - Access to the native Linux APIs
 - Protected from asynchronous Linux preemption
 - Controlled by Linux and RTAI

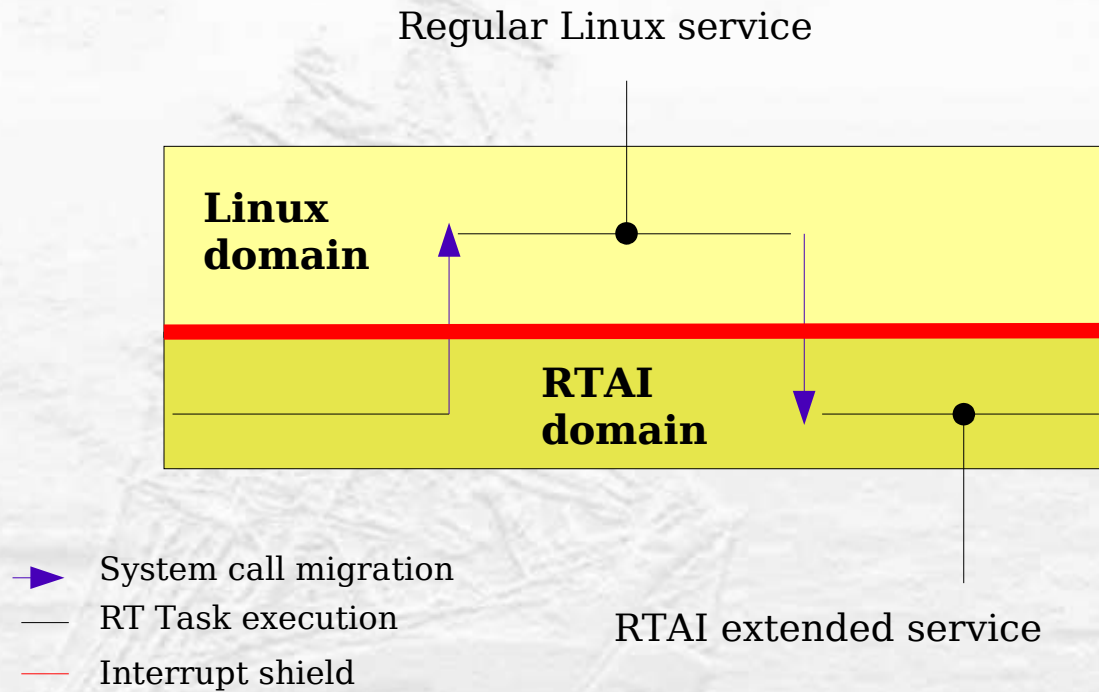
Real-time execution



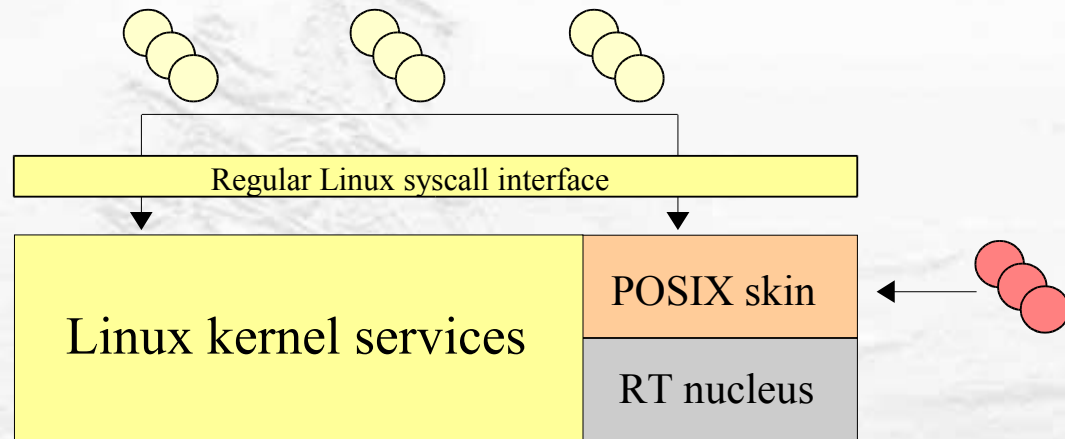
Seamless RT services integration



Real-time task migration



Mixed POSIX support



User-space applications

Kernel-based applications

Preliminary results

Number cruncher x86/UP (execution jitter, user-space)

	<i>Min.</i>	<i>Max.</i>	<i>Avg.</i>
Vanilla Linux 2.6.7	-29 μ s	521 μ s	51 μ s
RTAI/fusion	-3 μ s	35 μ s	-3 μ s

Nanosleep x86/UP (dispatch latency, user-space, 1Khz)

	<i>Min.</i>	<i>Max.</i>	<i>Avg.</i>
Vanilla Linux 2.6.7	-682 μ s	741 μ s	-1 μ s
RTAI/fusion	4.5 μ s	22.4 μ s	8.5 μ s

Thank you for attending.