

# Debugging ARM based Heterogeneous Multicore SoCs-

Kathy Tufto

Product Manager

Mentor Embedded Linux, Development Tools

**mentor**  
**embedded**

[mentor.com/embedded](http://mentor.com/embedded)

---

Android is a trademark of Google Inc. Use of this trademark is subject to Google Permissions.  
Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Qt is a registered trade mark of Digia Plc and/or its subsidiaries. All other trademarks mentioned in this document are trademarks of their respective owners.

# Trends Create Opportunities

## Competitive Market



Global Competition  
Pace of Innovation

## Consolidation (Reuse, Space, Weight, Power)



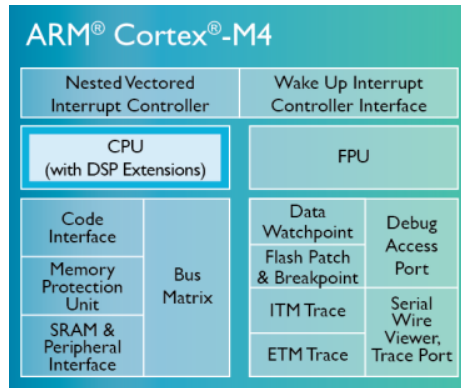
Multiple OS's  
Complex  
Configurations

## SoC Technology Advancement

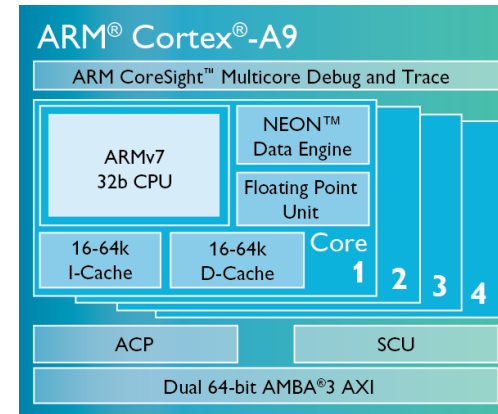


Mixed Core Types  
Increasing  
Complexity

# From Single Core to Multicore



Single Core MCU



Multicore MPU



Term	Description
Multicore	Processor with more than one core
Homogeneous Multicore	Processor with identical cores
Heterogeneous Multicore	Processor with non-identical cores

# System Development and Debug Today

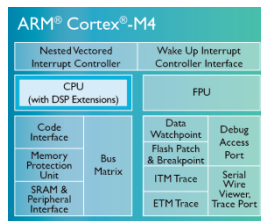
*Patient data acquisition*



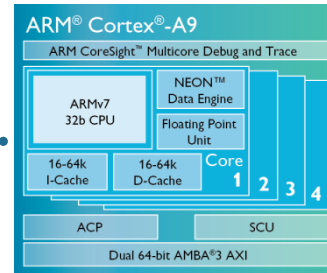
*Patient monitor*



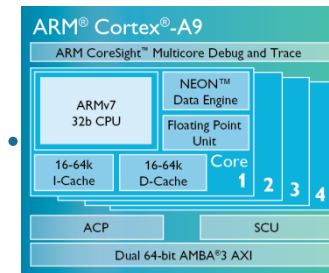
*Enterprise*



RTOS or Bare Metal



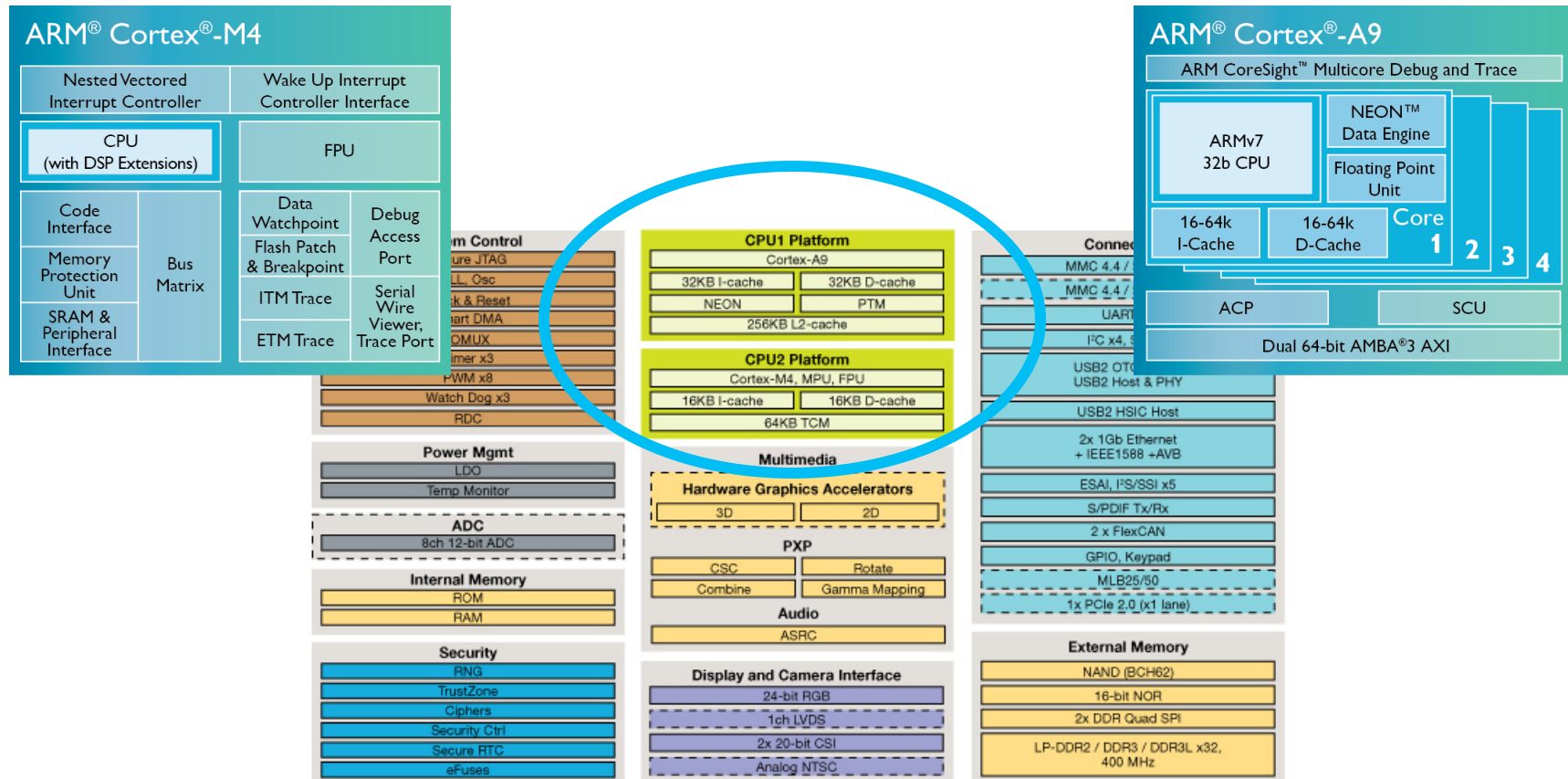
Android™ 



Linux® 

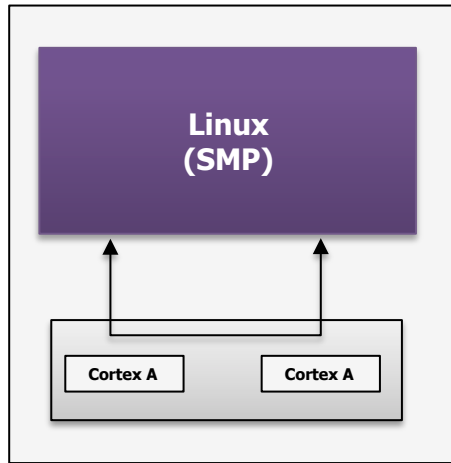
Each Device has Different Tools and Runtime Software  
 Separate and Discrete Processors  
 Applications are Typically Loosely Connected

# From Multicore to Heterogeneous



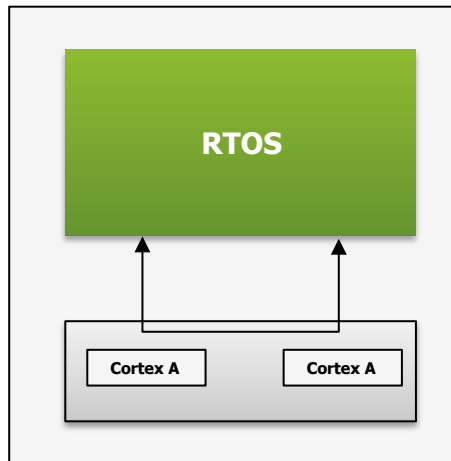
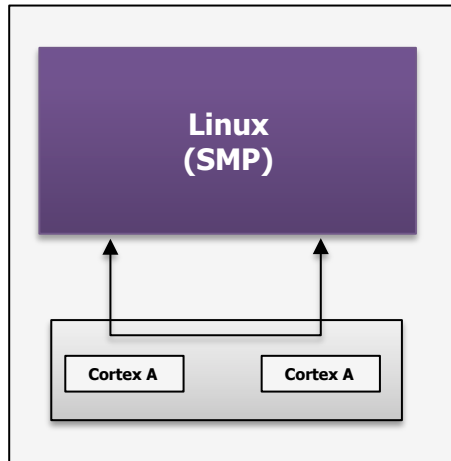
# Multicore Configurations

SMP



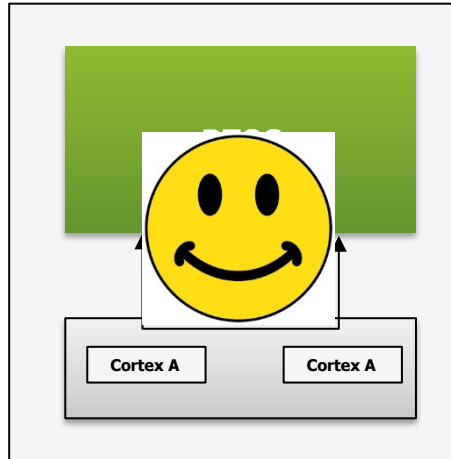
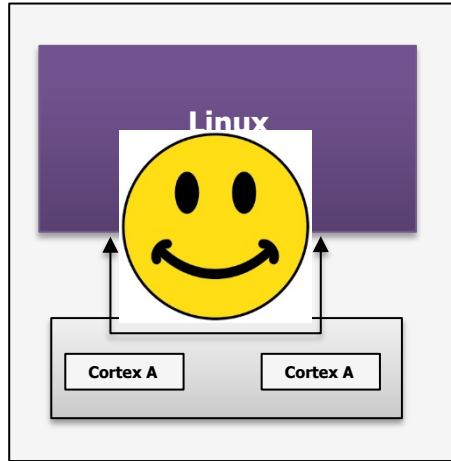
# Multicore Configurations

SMP



# Multicore Configurations

SMP

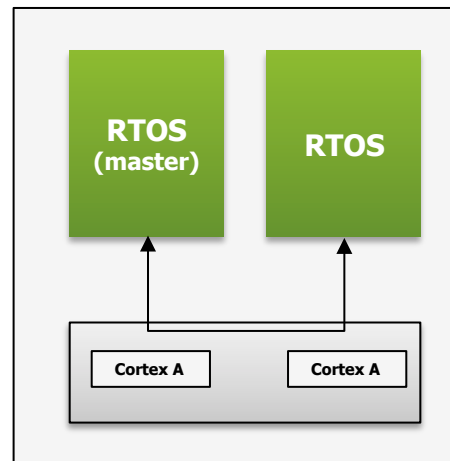
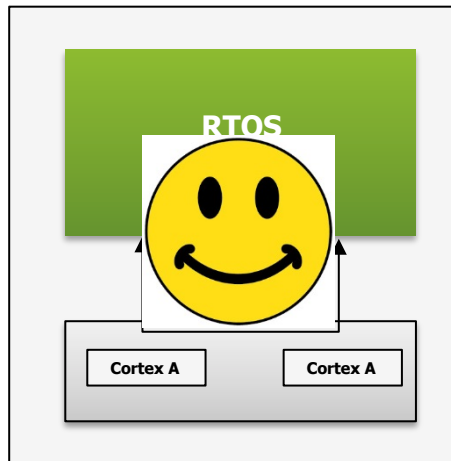
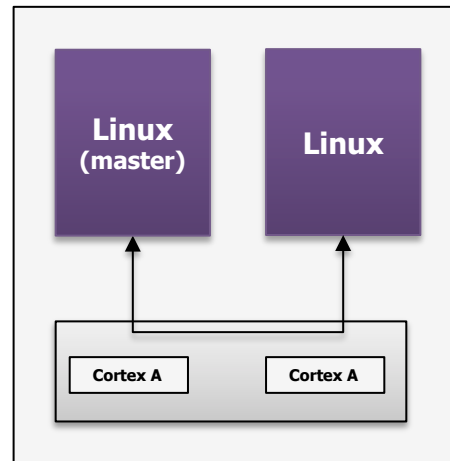
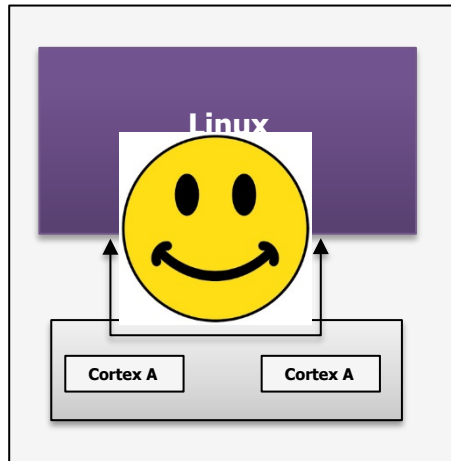




# Multicore Configurations

SMP

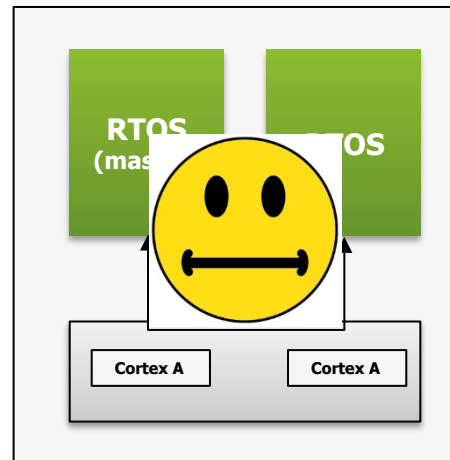
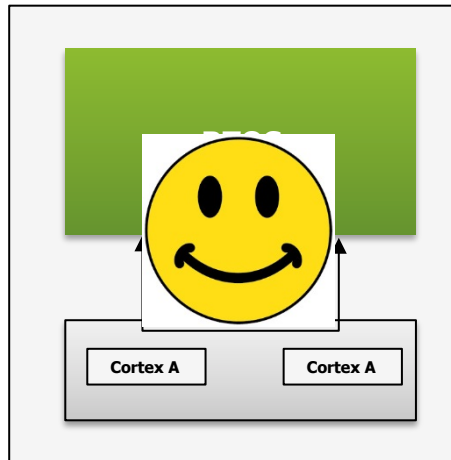
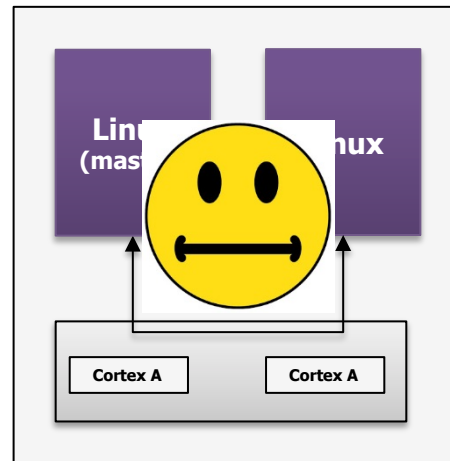
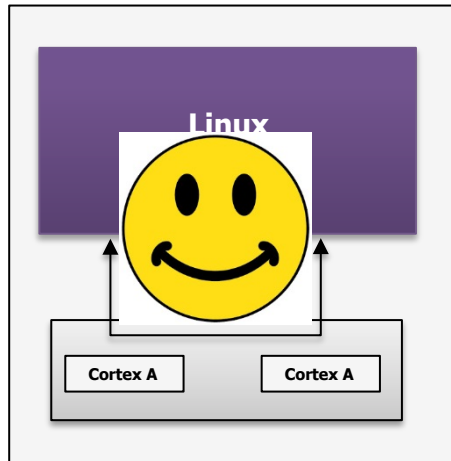
Homogeneous uAMP



# Multicore Configurations

SMP

Homogeneous uAMP

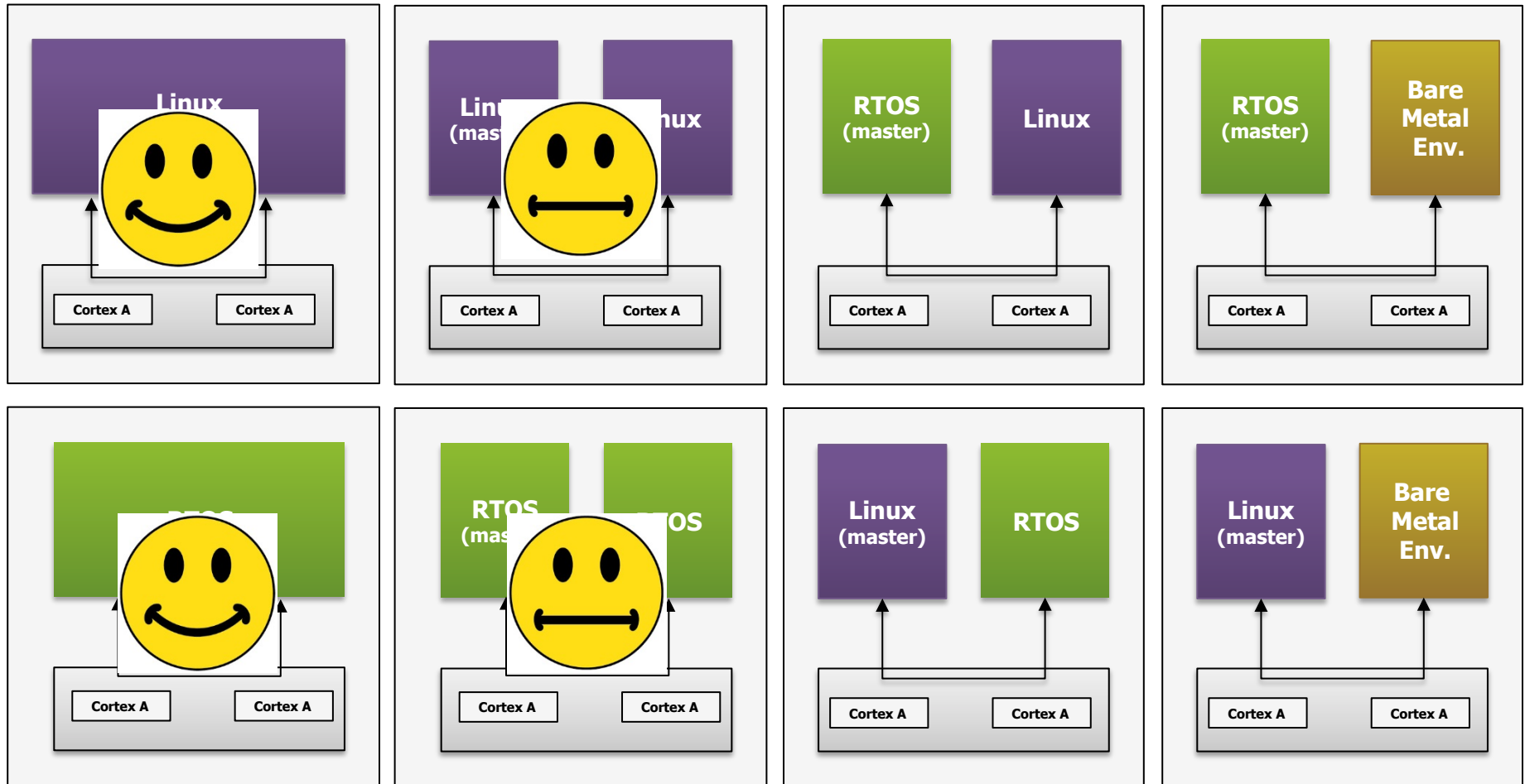


# Multicore Configurations

SMP

Homogeneous uAMP

Heterogeneous uAMP

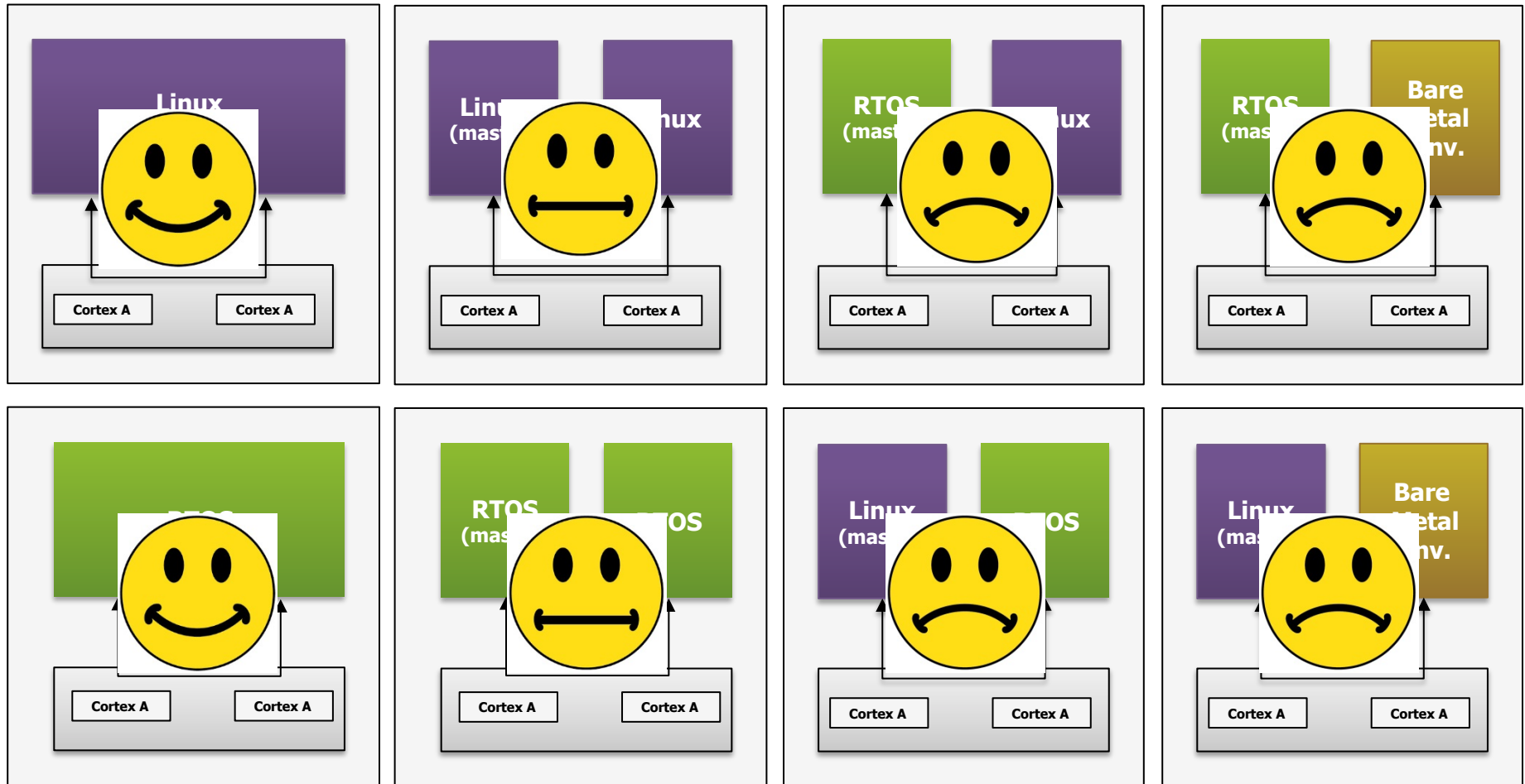


# Multicore Configurations

SMP

Homogeneous uAMP

Heterogeneous uAMP

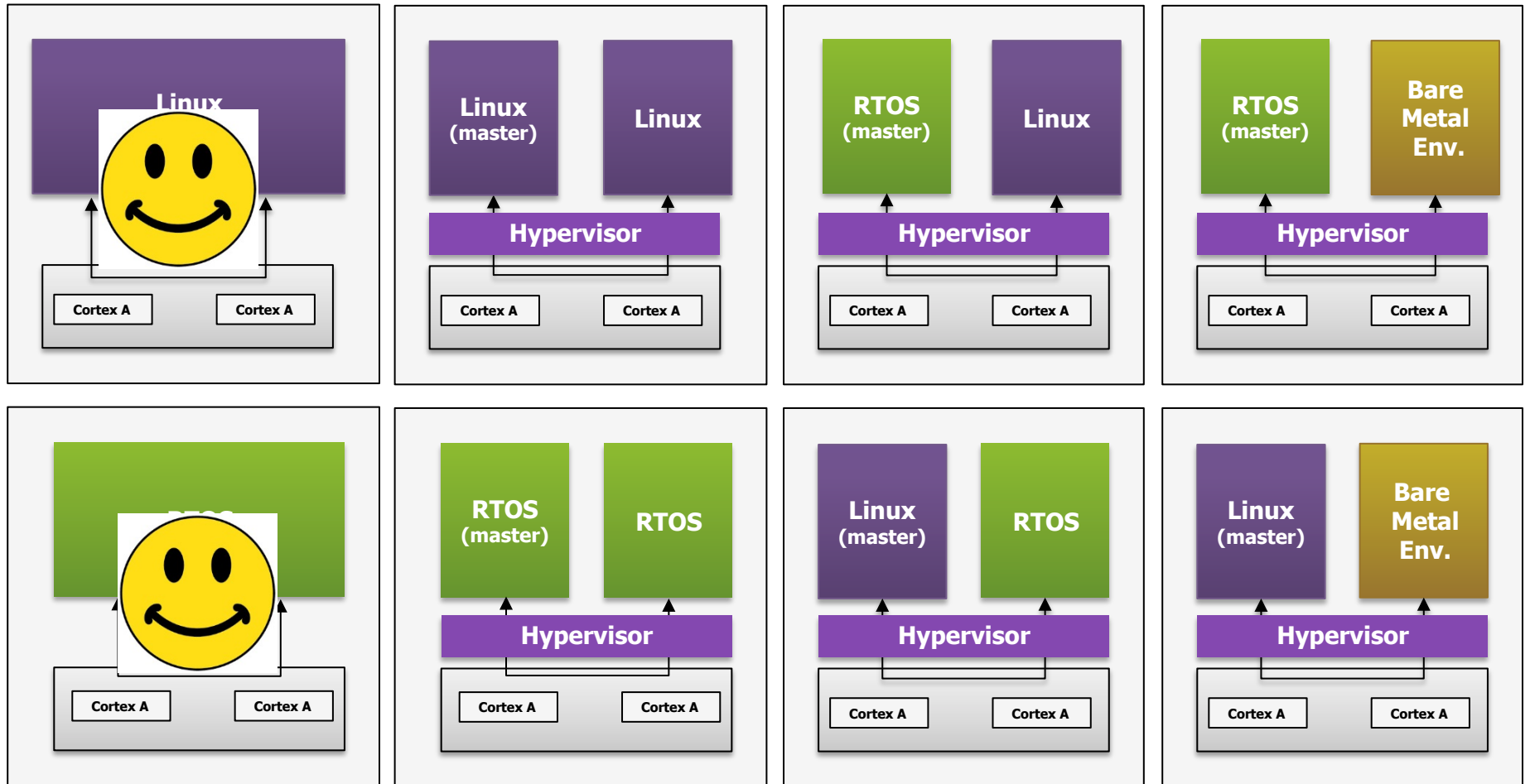


# Multicore Configurations

SMP

Homogeneous sAMP

Heterogeneous sAMP



# Multicore Configurations

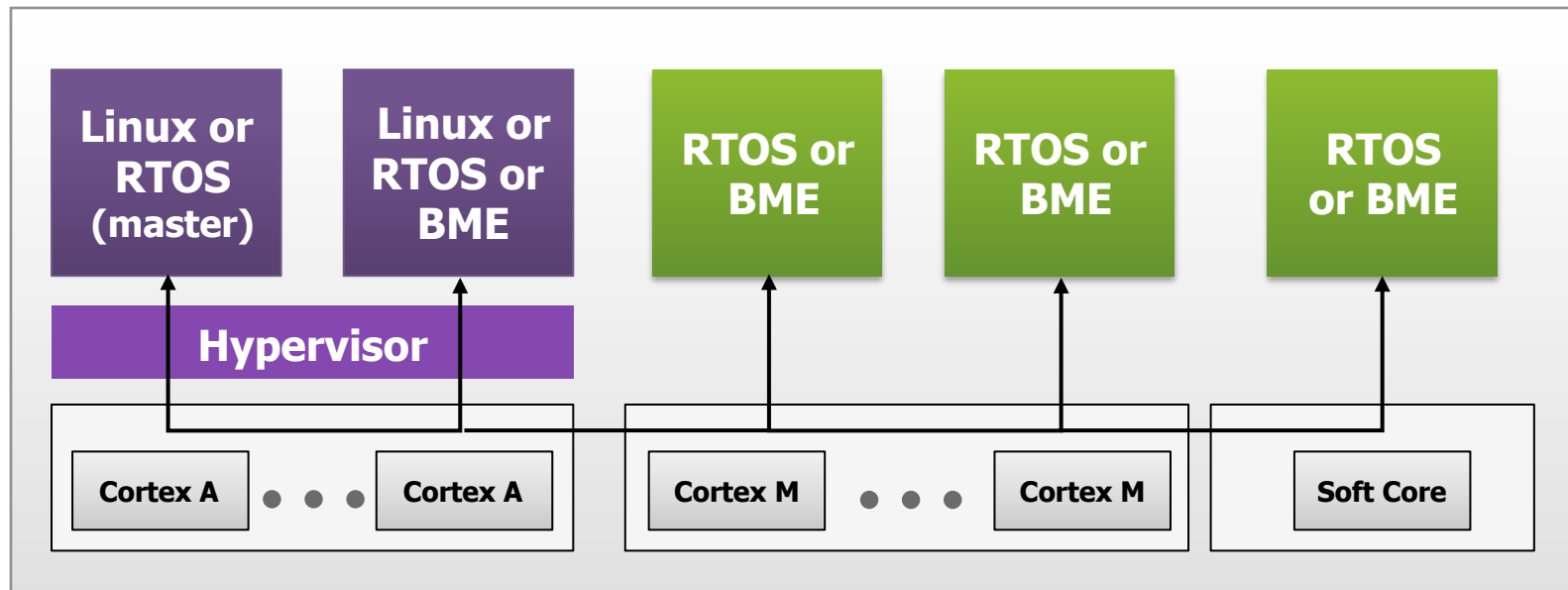
SMP

Homogeneous sAMP

Heterogeneous sAMP



# Adding Heterogeneous Cores



**14 use cases**

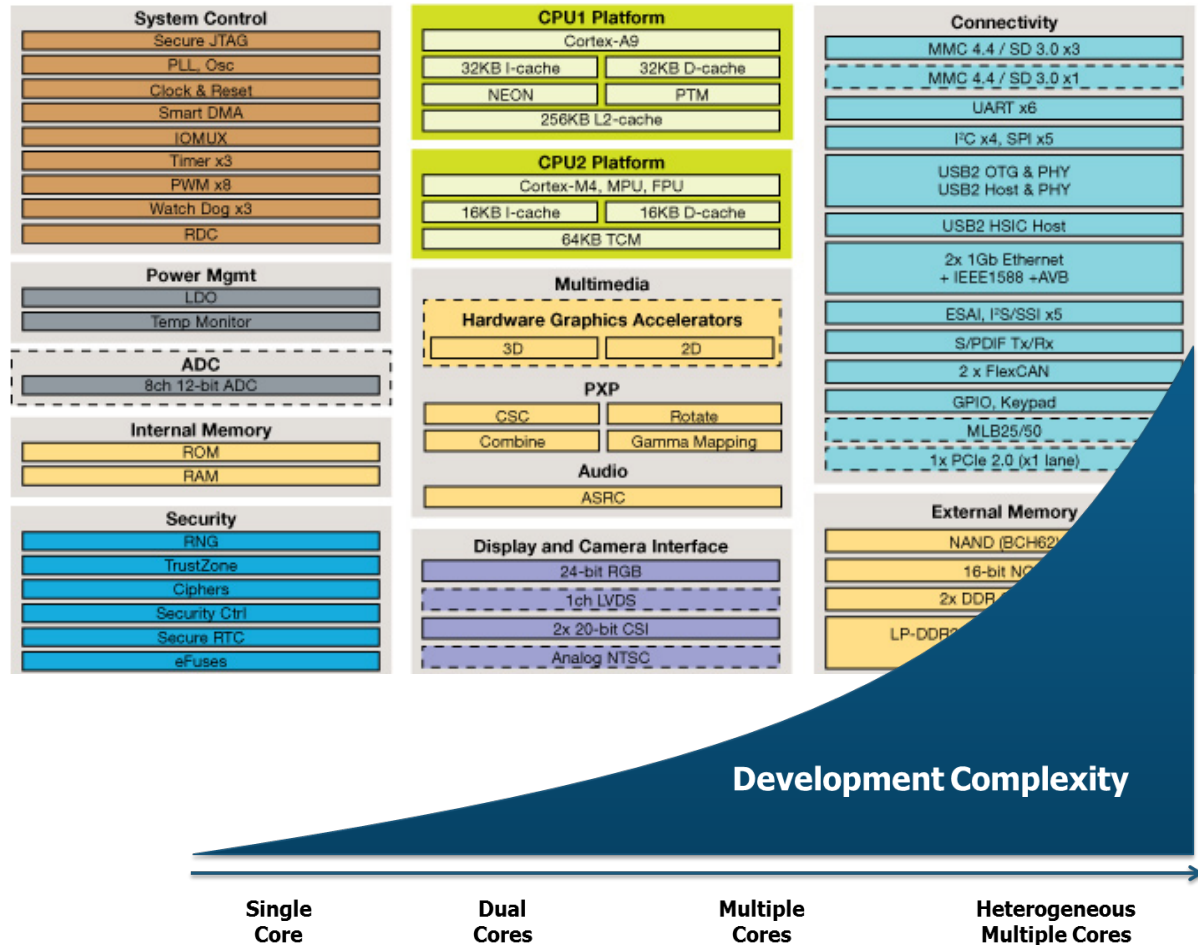
**\* N use cases**

**\* M use cases**



# Complexity Skyrockets

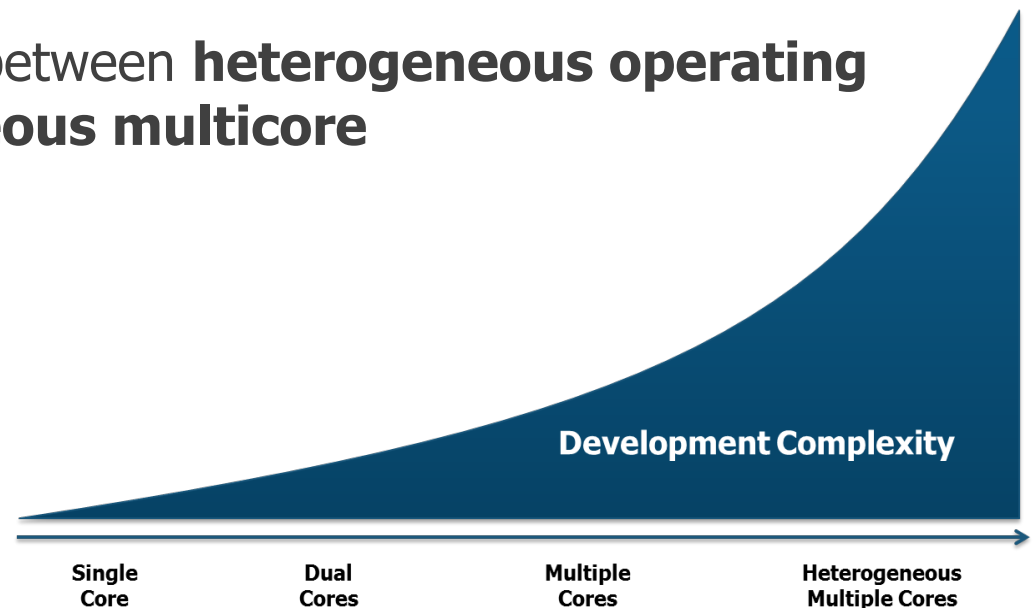
Extreme complexity is introduced with general purpose development



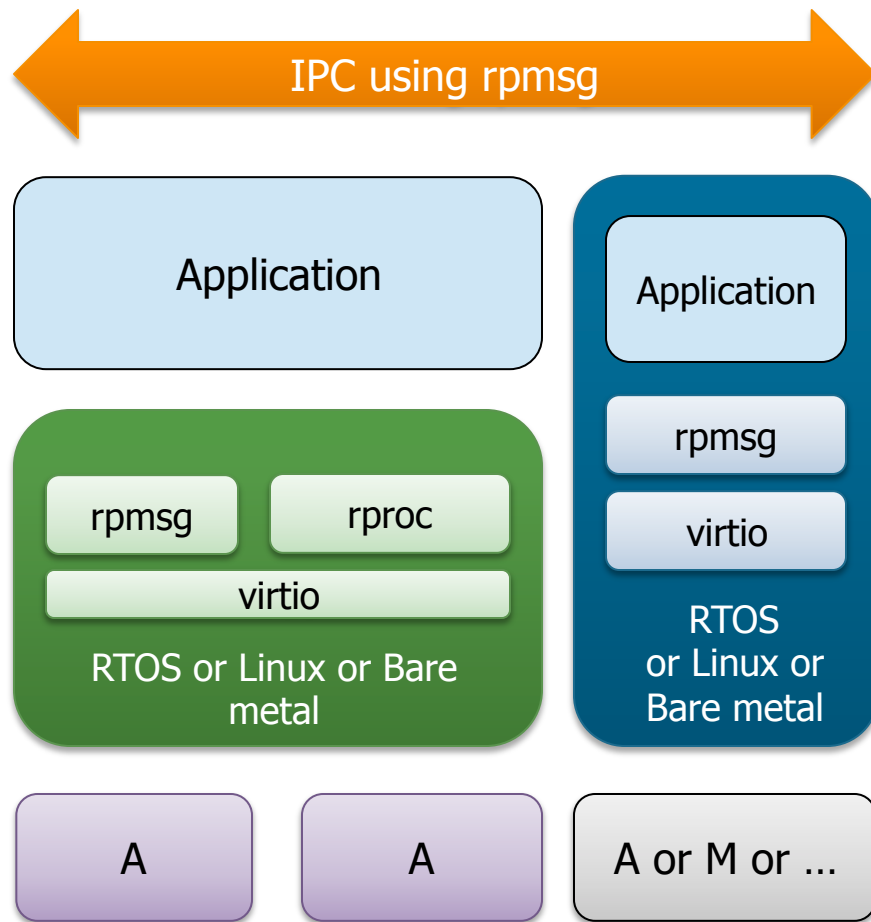


# Heterogeneous Development Challenges

- **Configuring and deploying multiple operating systems** and applications across heterogeneous processors
- **Communicating** between isolated sub-systems on a multicore processor or between **heterogeneous processors**
- **Debugging** heterogeneous operating systems and user-space
- **Visualizing** interactions between **heterogeneous operating systems on heterogeneous multicore**

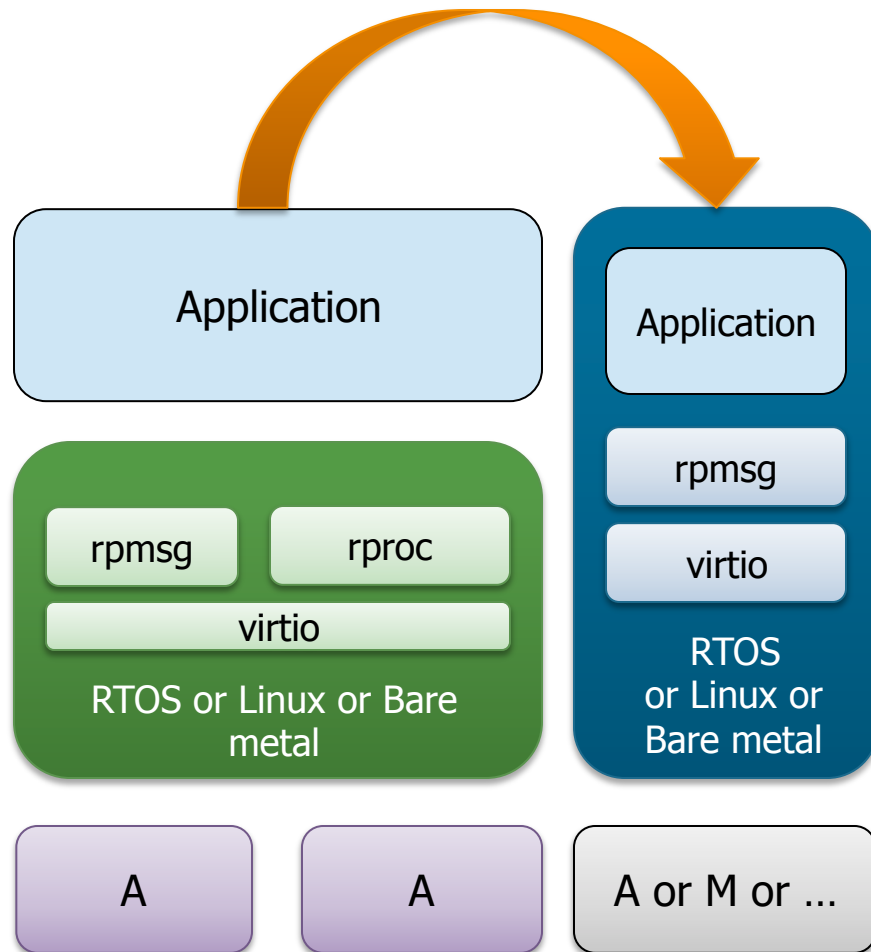


# AMP Framework: IPC



- For inter-processor communications between OS/software contexts
- rpmsg user API for Inter Processor Communication
- Conformance to upstream Linux rpmsg implementation
- Stand alone OS agnostic clean-room implementation of virtio and rpmsg
- Usable from RTOS and BME contexts

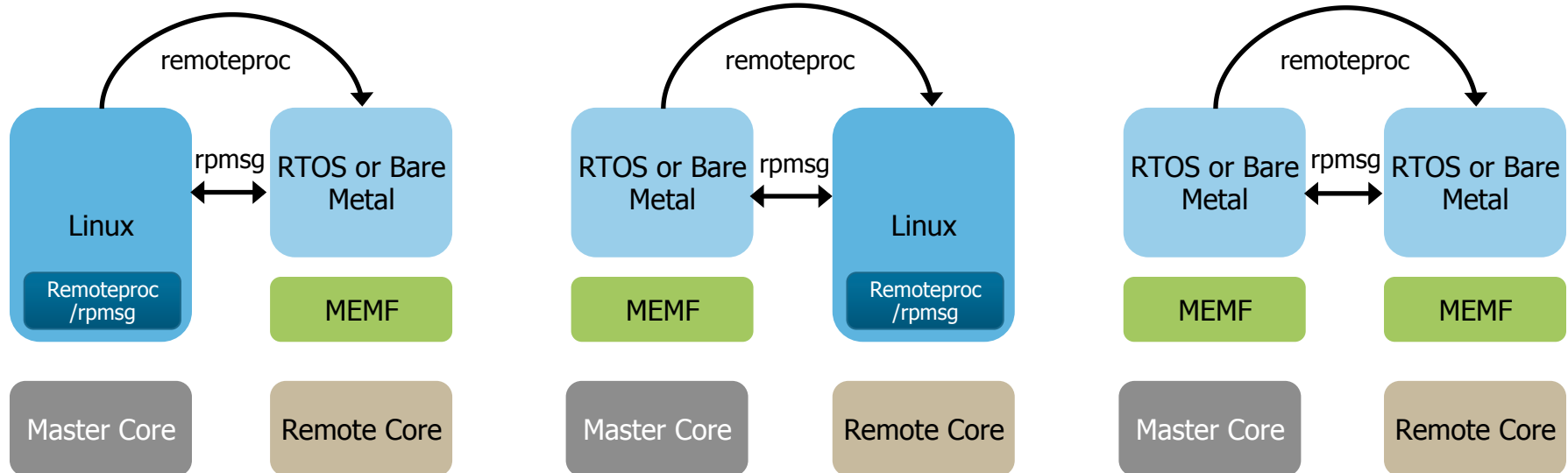
# AMP Framework: Core Management



- Used by master OS to boot remote OSs on remote CPUs
- remoteproc user API for processor lifecycle management
- Conformance to upstream Linux remoteproc implementation
- Stand alone OS agnostic clean-room implementation of remoteproc API

# AMP Framework: Use Cases

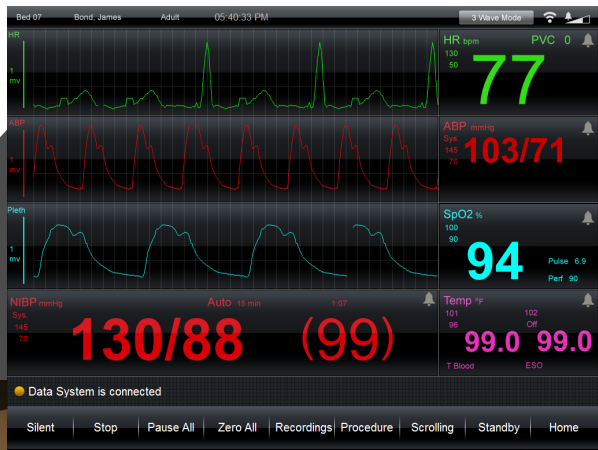
- Separation:
  - Resource constrained, for example power management
  - User interface applications from critical functions
- Offload work for:
  - Computationally intensive operations
  - Real-time and/or deterministic tasks



# Heterogeneous System: Example

Vitals Data Acquisition

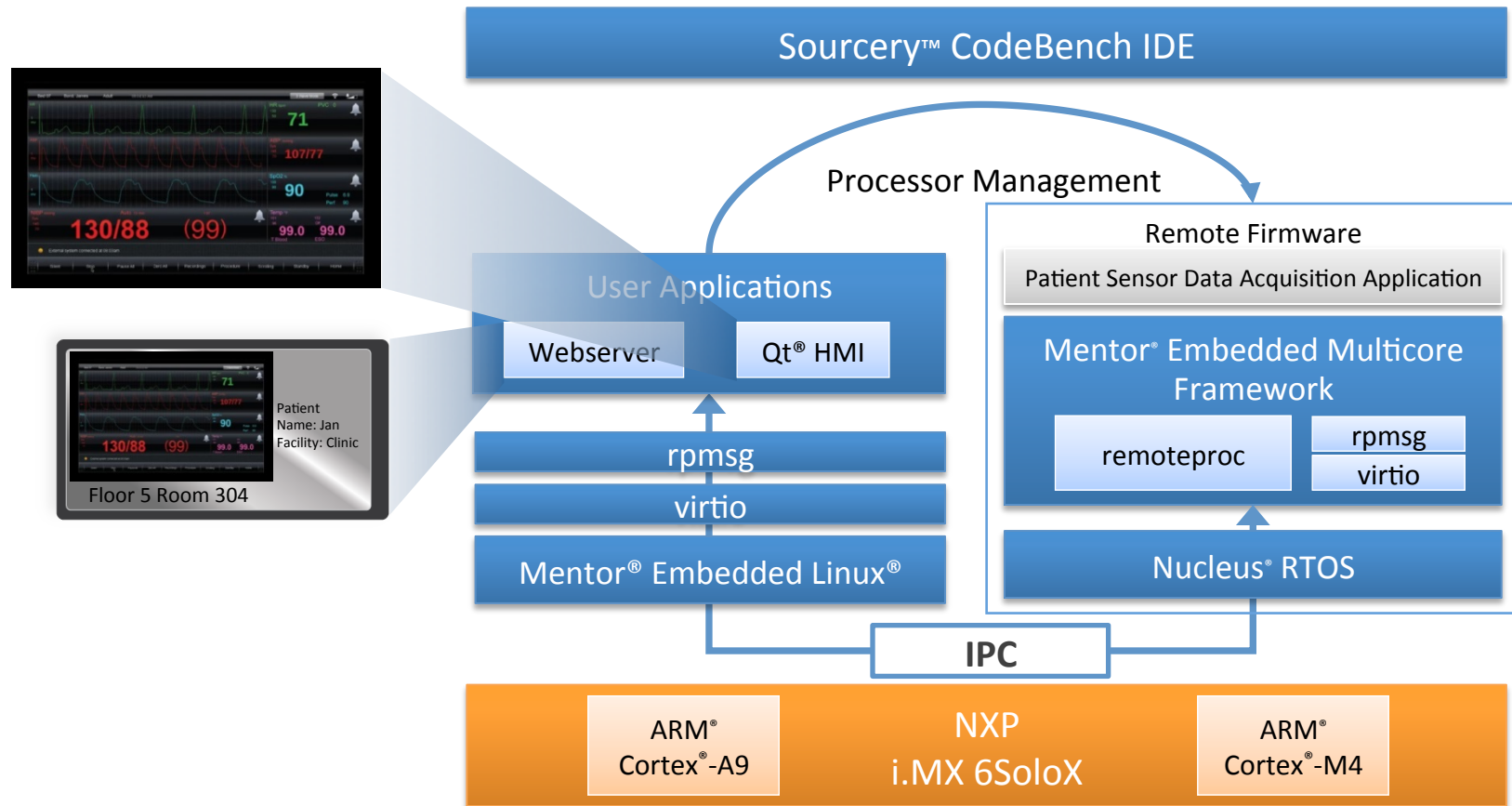
Real-Time Display



Web View



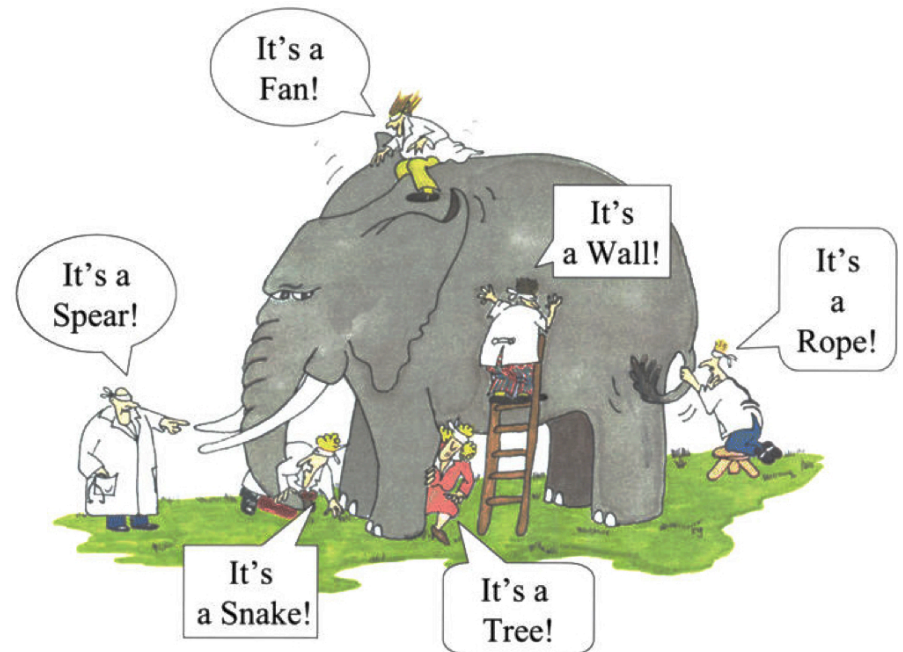
# Heterogeneous System: Architecture



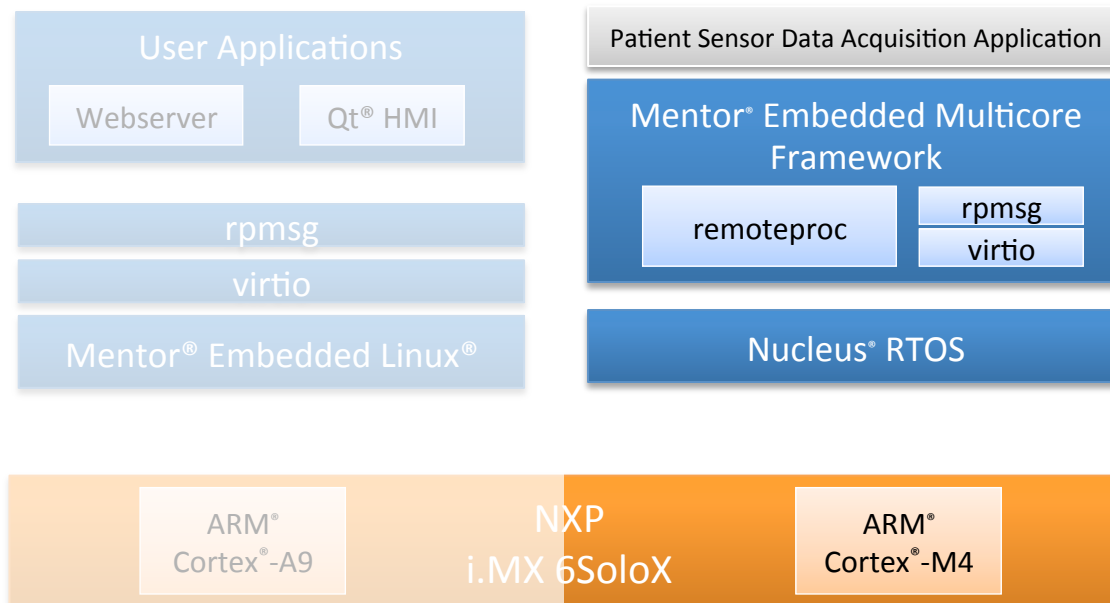
# Heterogeneous Debugging Challenges

Extreme complexity is introduced with heterogeneous systems

- You need to be able to debug all the applications running on all the different operating systems and processor cores.
- You must determine which cores need to halt when you hit an OS level breakpoint on each core.
- You need to identify where there might be shared resource contention, or saturation of processors, busses or devices.
- You need a way to see how behavior on one part of the system affects, or is affected by, a different part of the system.

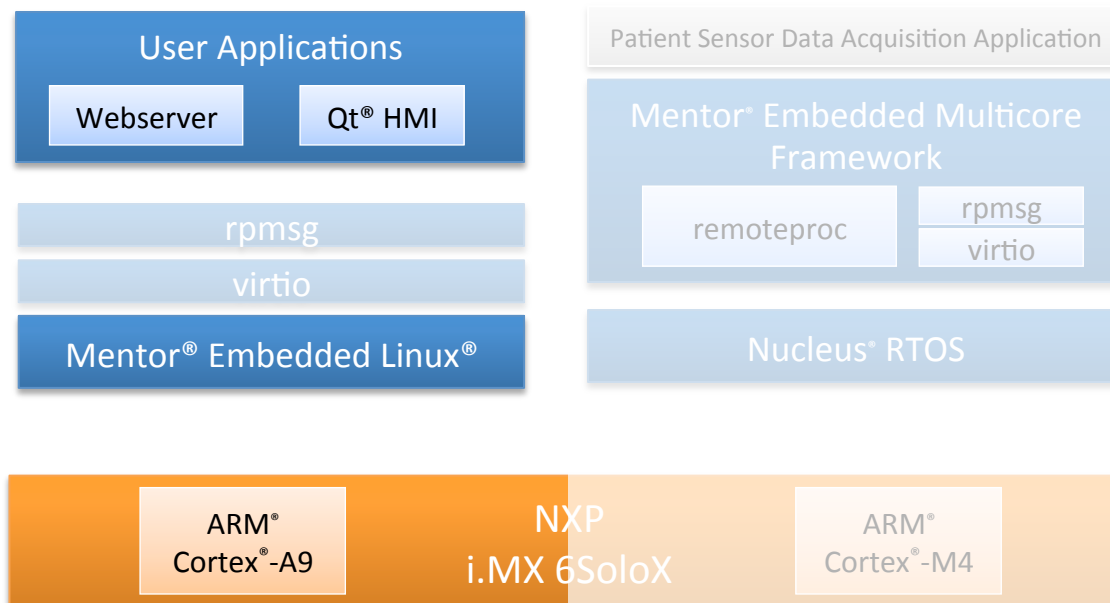


# How would you debug an RTOS?

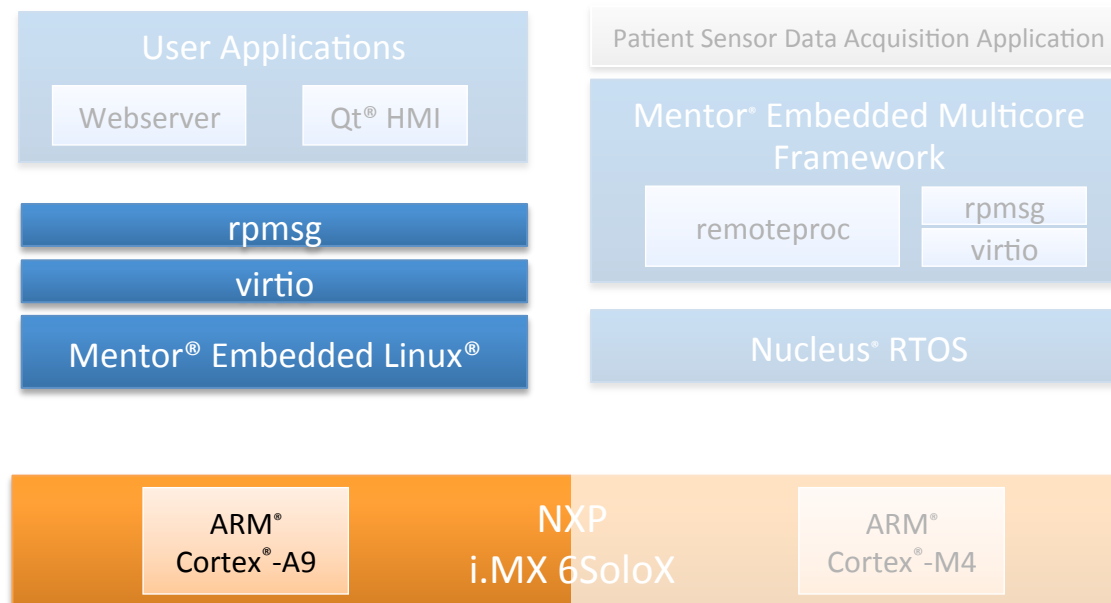




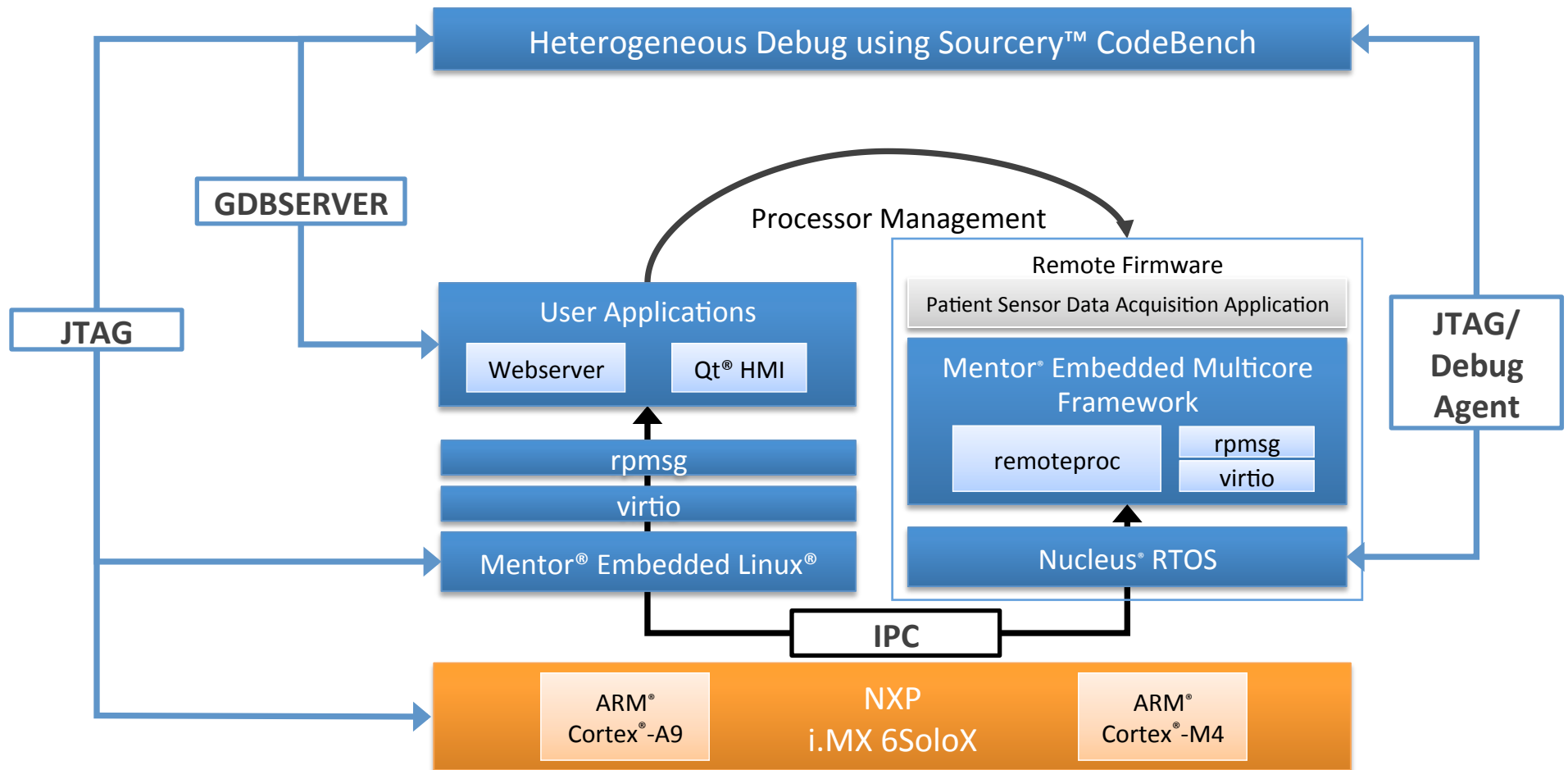
# How would you debug a Linux App?



# How would you debug a kernel module?



# Heterogeneous Debug Use Cases



# Heterogeneous Debug Using Sourcery CodeBench

The screenshot displays the Sourcery CodeBench IDE interface on an Ubuntu 14 64-bit test1 system. The main window shows a debug session for the project `arm-none-linux-gnueabi/src/glibc/sysdeps/unix/syscall-template.S`. The interface is divided into several panes:

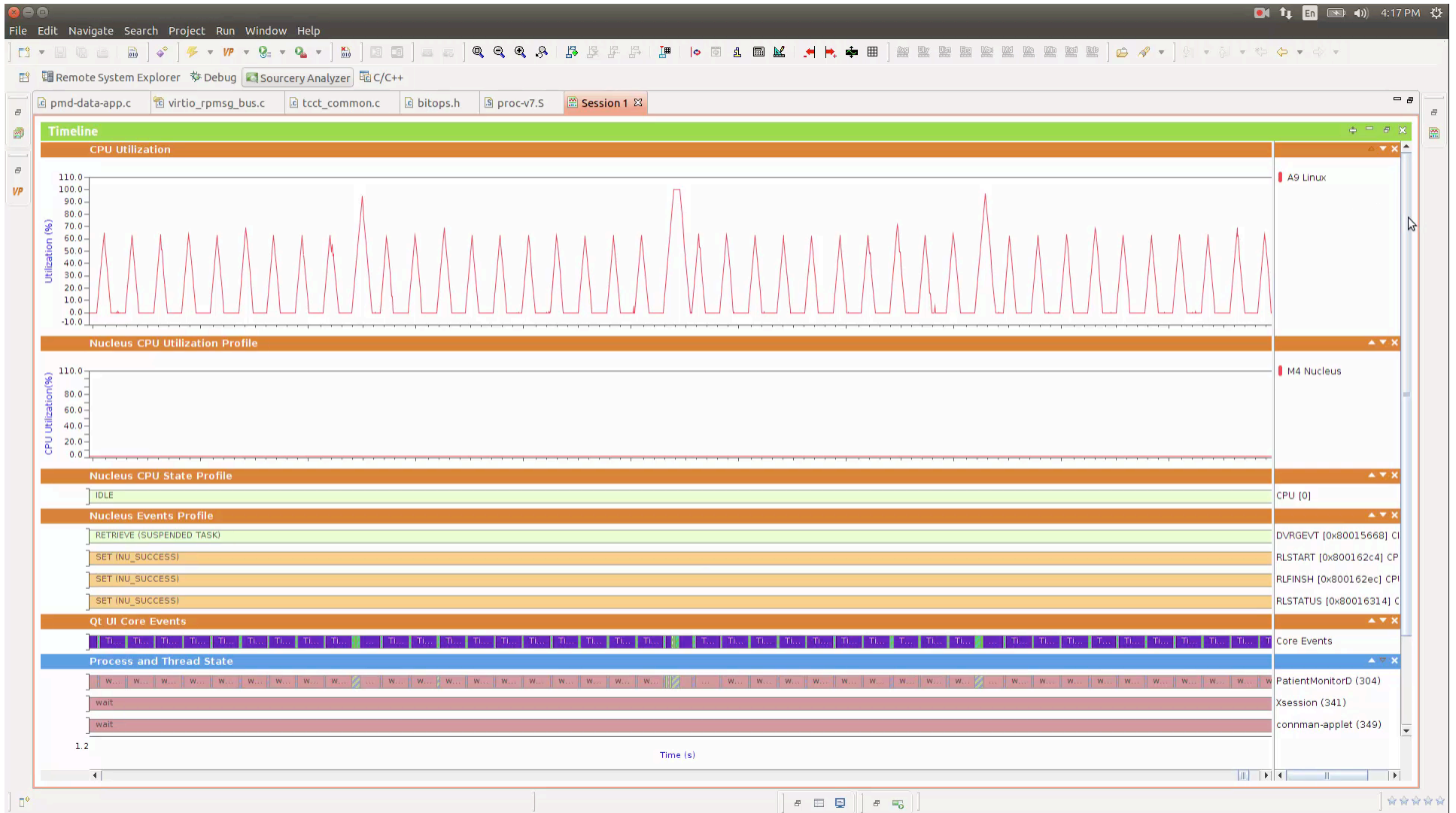
- Debug Console:** Shows the execution of `gdb` on `pmd-master`. The output indicates: `The target endianness is set automatically (currently little endian)`.
- Analysis Sessions:** Lists several debug sessions, including `pmd-data-app` and `kern debug proj`.
- Variables:** Displays the current state of variables, including `sync.c [function: sys_sync]`, `pmd-data-app.c [line: 148]`, and `virtio_rpmmsg_bus.c [line: 794]`.
- Code Editor:** Shows the source code for `syscall-template.S`, including macros like `T_PSEUDO_ERRVAL` and `T_PSEUDO_END_ERRVAL`.
- Outline:** Provides a hierarchical view of the project files, such as `sysdep-cancel.h`, `sysdep.h`, and `T_PSEUDO`.

Overlaid on the right side of the IDE is a medical monitor interface for `SSVNC: mx6-solox0 - Press F8 for Menu`. The monitor displays vital signs for `Bed 07` (Patient: `Bond, James`):

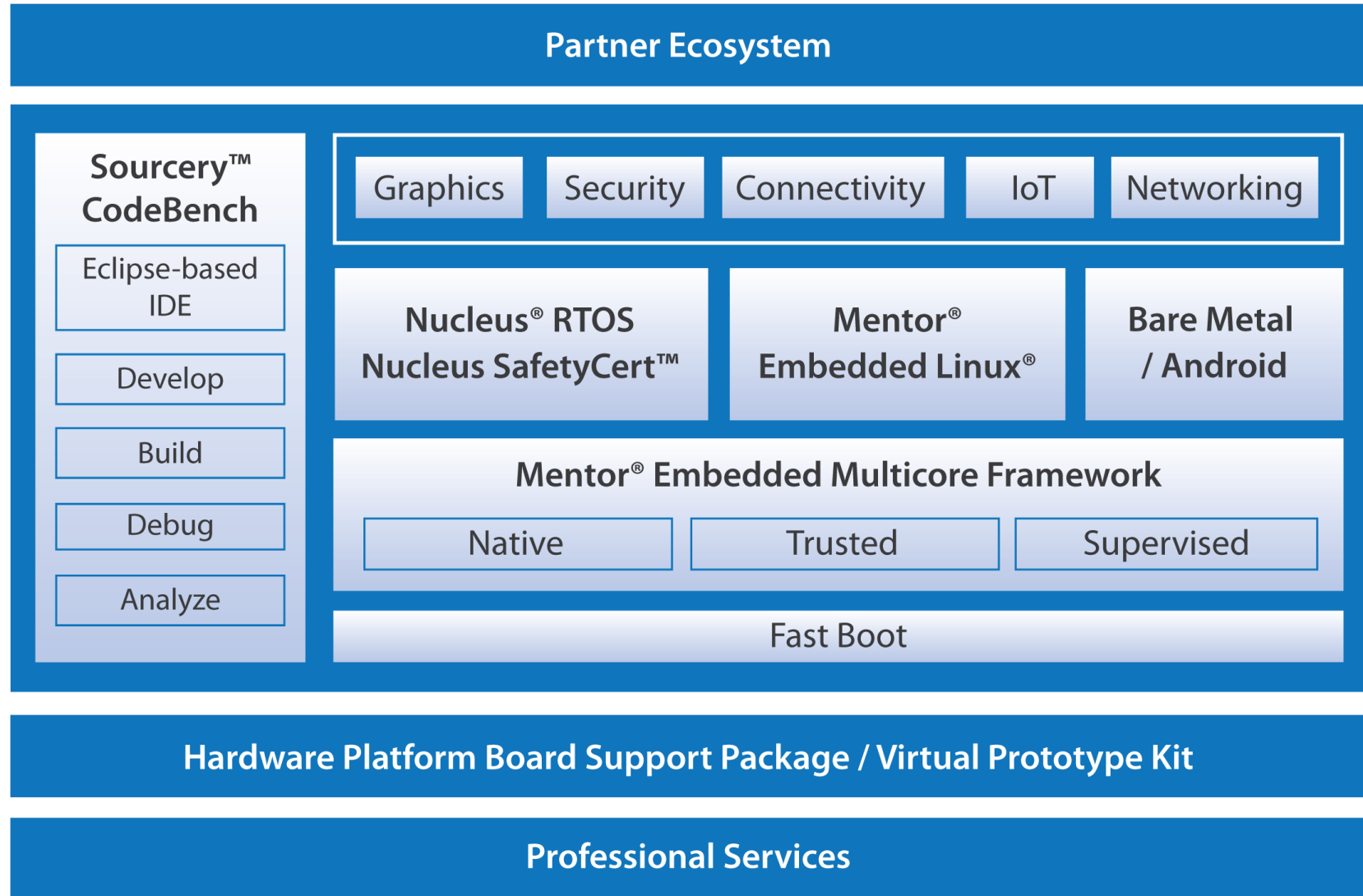
- HR bpm:** 68
- ABP mmHg:** 110/88
- SpO2 %:** 96
- Temp °F:** 99.0
- Pulse:** 6.9
- Perf:** 90

The monitor also shows a `PVC 0` alert and a `(99)` status indicator. At the bottom, it lists `Recordings`, `Procedure`, `Scrolling`, `Standby`, and `Home` options.

# System Analysis using Sourcery Analyzer



# Mentor Graphics Runtime Platforms Portfolio



# THANK YOU