Realtime Java for Industrial and Critical Applications

Andy Walter
Director of Sales, aicas GmbH
June 2006

The aicas group

- aicas GmbH was founded in March 2001 out of the FZI and the University of Karlsruhe in Karlsruhe, Germany
- Bringing object oriented technology to realtime and embedded systems
- aicas incorporated was founded in February 2005 in New Haven, CT to better serve the US market.

aicas' partners

Research

FZI

Universität Karlsruhe (TH)

Fraunhofer Institut Integrierte Schaltungen

Operating Systems

concurrent

RadiSys

Green Hills Software, Inc.

QNX

WIND RIVER

Linux Verband

Industrial

BECKHOFF

B&R

MicroSys

SUTRON Electronic GmbH

TQ components

Trainings

MicroConsult

RP-Cube

Selected aicas Customers

KEBA
Automation by innovation.

BOEING

MULTIVAC
BETTER PACKAGING

EADS

Saurer

JAXA
Japan Aerospace Exploration Agency

SICK
Sensor Intelligence.

esa

SIEMENS

THALES

USTER®
Think quality

aicas Resellers and Sales Offices

Value Added Reseller: net Module

Canada JORAL.ca

France: NeoMore

Germany: GmbH aicas realtime

Japan: Japan aicas realtime

Space Domain: EADS ASTRIUM

USA: Incorporated aicas realtime

Embedded Realtime Applications





Examples:

industrial automation, avionic/satellite, automotive, telecom, medical, ...

Competitive Advantages

- Time to Market
- Reliability
- Flexibility
  - Look and feel
  - Extensibility
- Reuse
- Platform independence

Java in Embedded Systems?

- Traditional Java systems have
  - High memory requirements
  - Poor runtime performance
  - Pauses during execution due to GC
    - Poor user interface feedback
    - Unacceptable for realtime, especially mission critical and safety critical systems
- JamaicaVM was designed for small, embedded realtime systems

JamaicaVM in Embedded Systems!

- Static Compiler Technology with Profiling
  - Faster Code
  - Better time vs. space trade off
- Smart Linking
  - Only include what is necessary
- Deterministic Garbage Collection
  - GC does not interrupt other (realtime) tasks
  - No pauses in the application

Aeronautics Example

- JamaicaVM takes off:
- EADS' Barracuda had it's first flight in May 2006

Aerospace Example

Java solution for Satellites

Result of a project with ESA and EADS Astrium

Partnership with EADS Astrium in Toulouse

Aeronautics Example

BOEING®

- JamaicaVM selected for use in the Boeing 787 Dreamliner
- First milestone reached on time

Industrial Automation Example

SIEMENS A&D

SIEMENS

Siemens licensed the JamaicaVM for Industrial automation.

- Several Simotion Drive products use JamaicaVM

Java Libraries

- Java provides many Standard Libraries:
  - Networking
  - Common data structures (Hash tables, trees, etc.)
  - Graphics
  - File I/O (incl. ZIPs)
- Many additional libraries available:
  - E.g., Bouncy Castle (free Crypto Library)

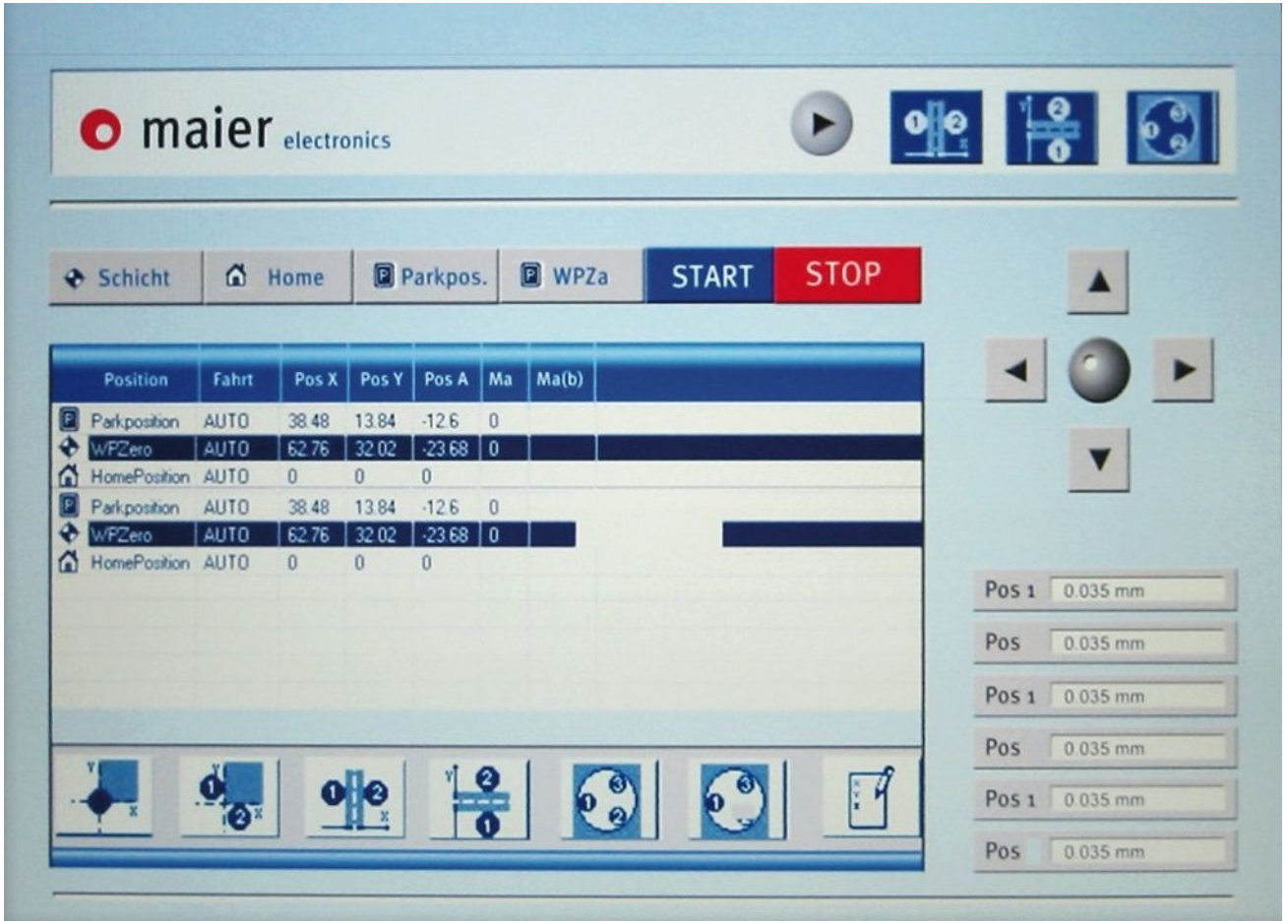
Visualisation with Java

- Very Flexible
- Your Corporate Identity can be used
- Looks identical after OS or CPU change

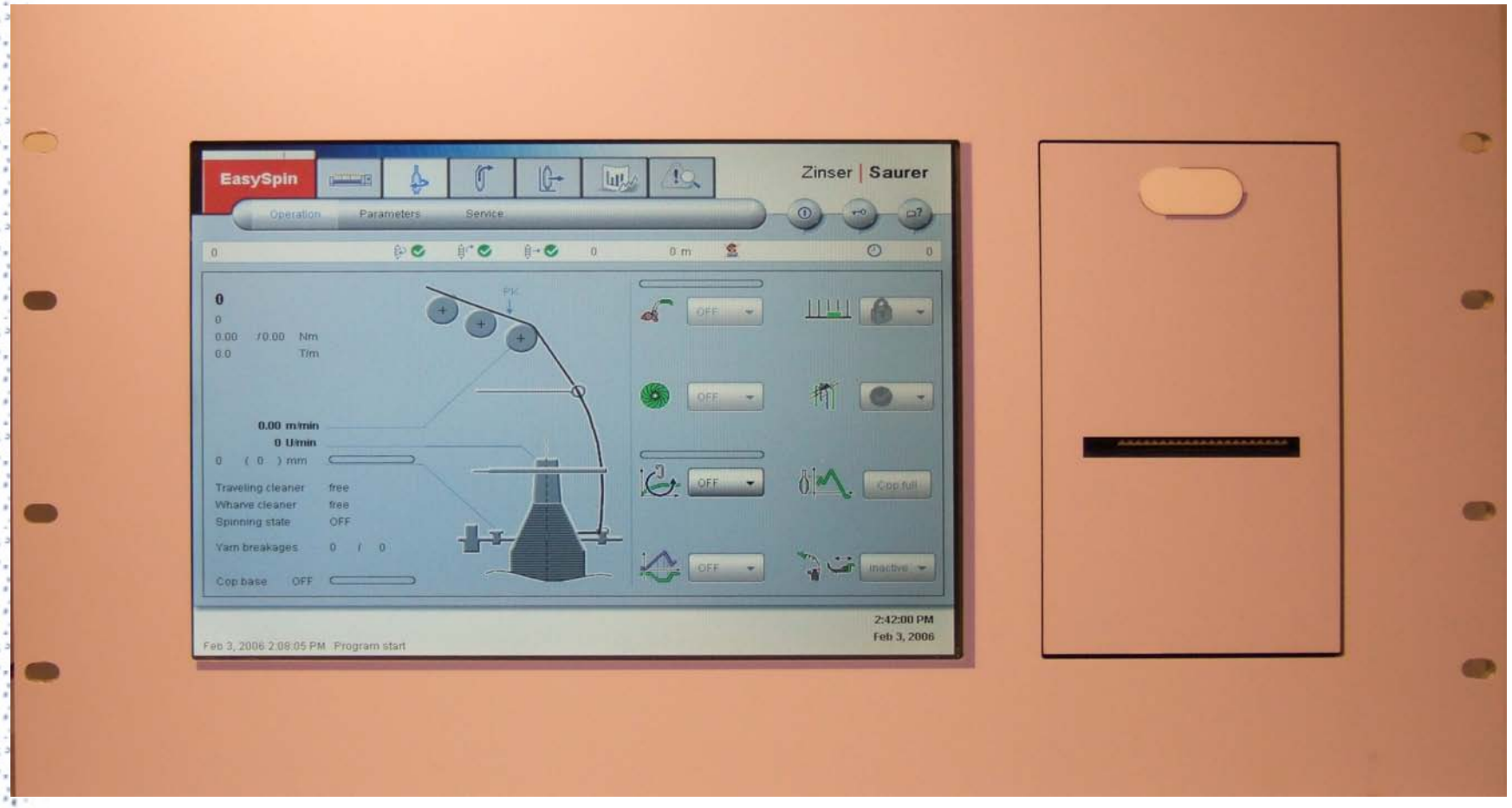
- With JamaicaVM:
  - Fast response time
  - No stopping

Visualisation with Java

Visualisation with Java



Visualisation Java

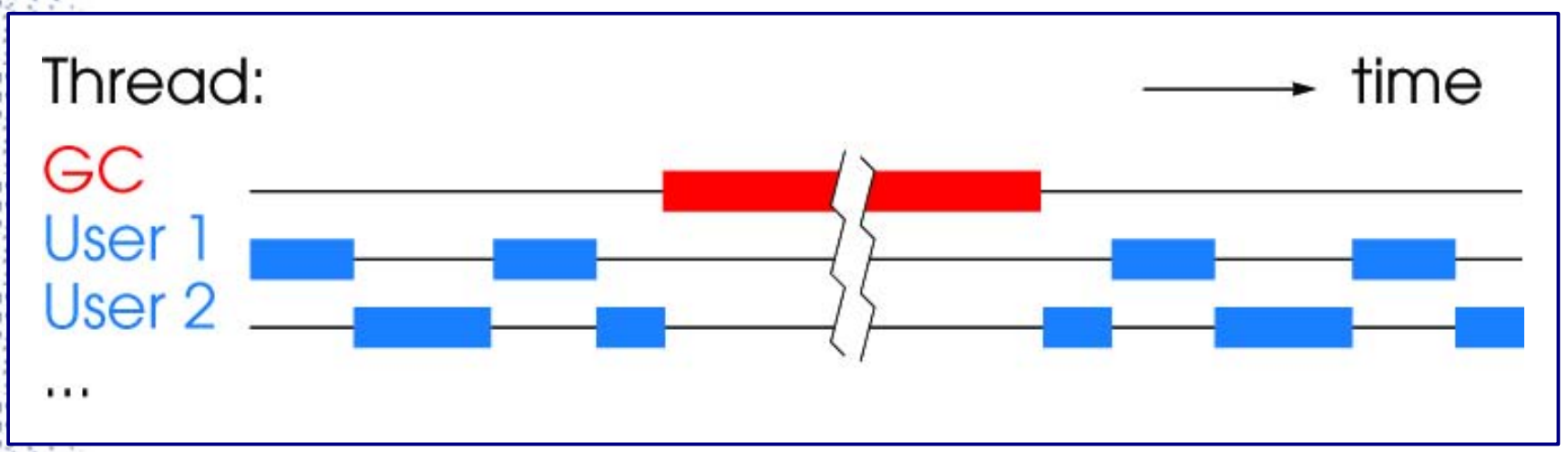


Memory Garbage Collection

- Saves Development Time:
  - No need to release objects explicitly
- Is safe:
  - Lowers the danger of memory leaks Removal of living objects doesn't happen

Classic Garbage Collector

GC can stop execution for long periods of time:

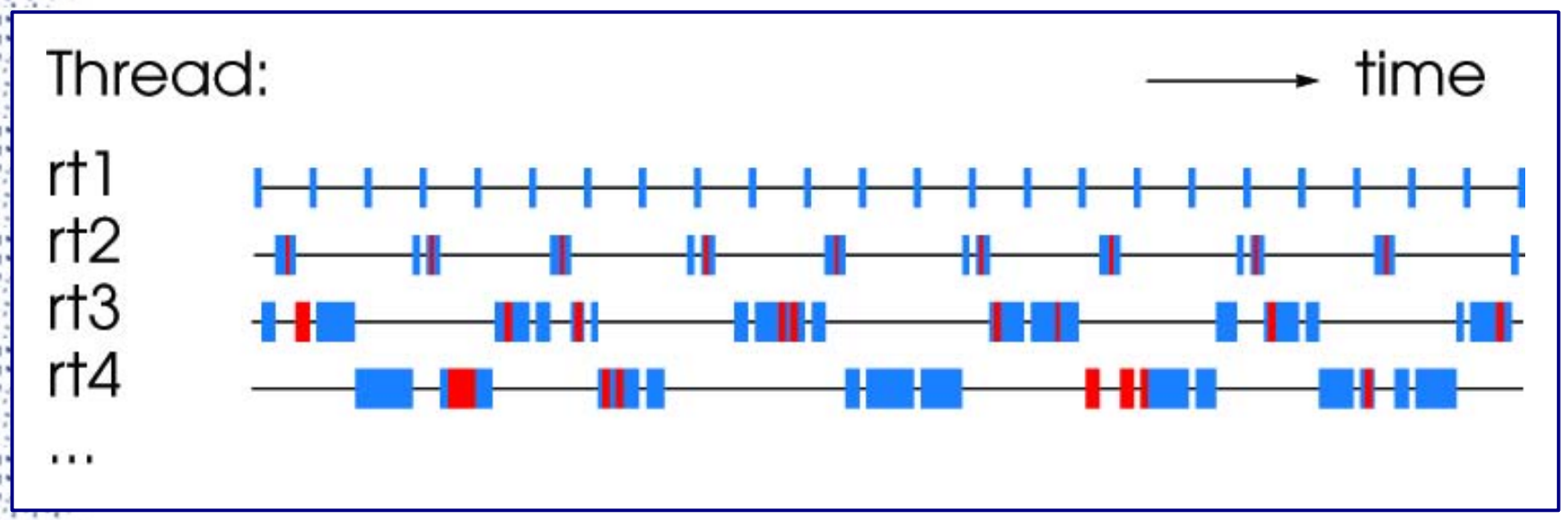


Problem:

long, unpredictable pauses

Realtime Garbage Collection

All Java-Threads must be realtime threads:



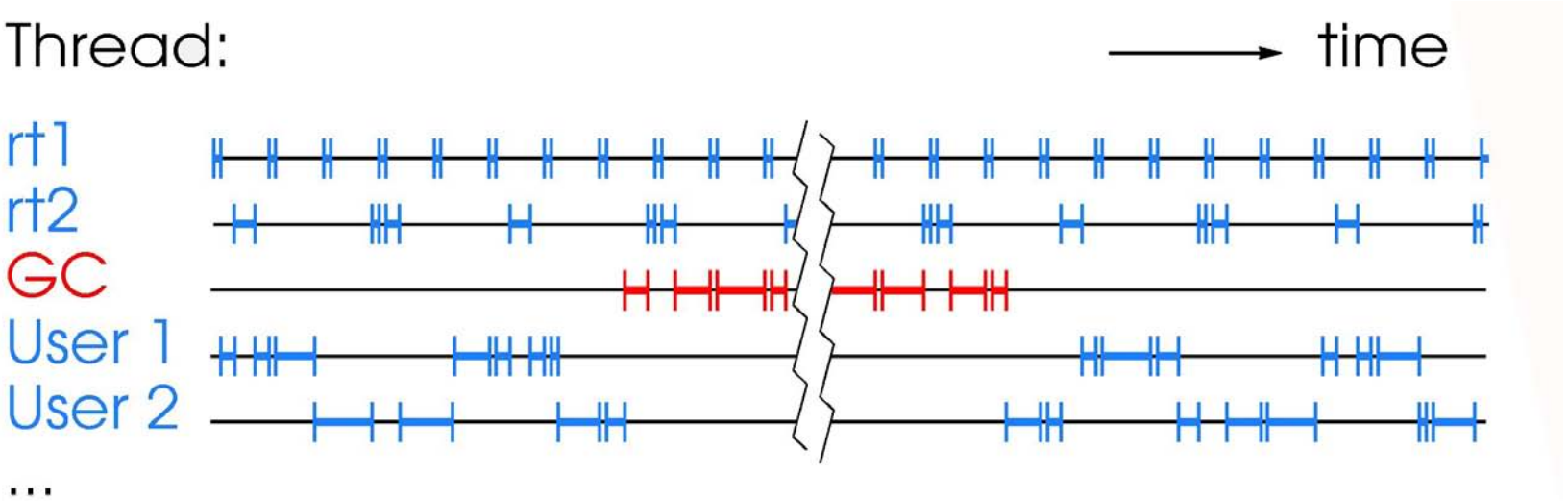
- GC-work is performed at allocation time
- GC-work must be sufficient to recycle enough memory before free memory is exhausted
- WCET for memory allocation is required

Realtime Specification for Java (RTSJ)

- Official Java Community Specification (JCP)
- Access of specific memory areas possible
- Accurate clocks and timers (ns)
- Asynchronous Control Flow
- Asynchronous Event Handlers ("Interrupt Handler")
- Priority Ceiling
- Priority Inheritance
- Can be implemented based on Standard JVMs

Garbage Collection in RTSJ

- Realtime application with RTSJ



- Separation of realtime- and non-realtime part
- No direct synchronisation possible
- No Garbage Collection in realtime threads

Memory Reclamation without GC

ScopedMemory

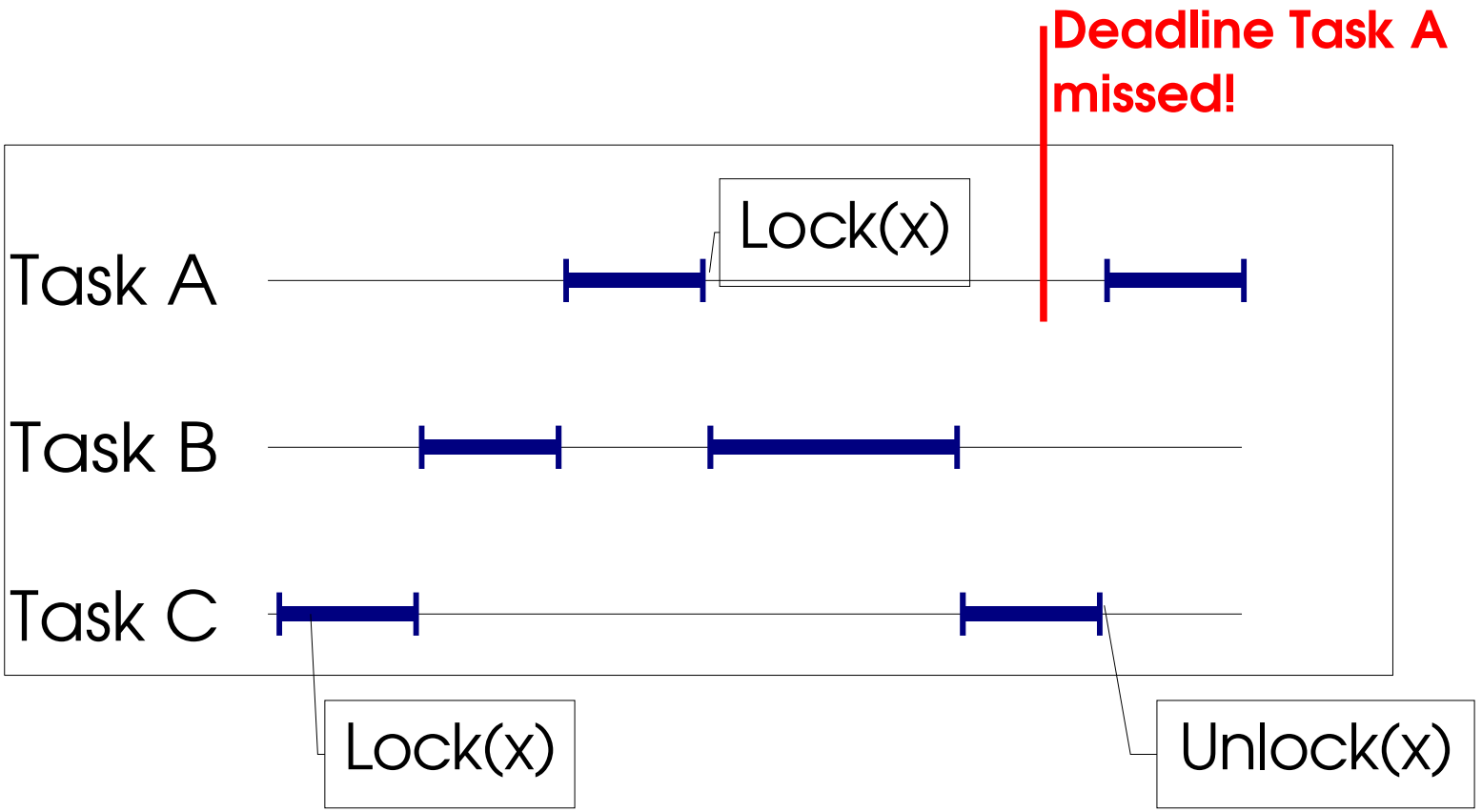
- region based memory management
- only active if entered explicitly
- memory is reclaimed automatically when exited
- Avoiding dangling and false references
- assignment rules prohibit creation of potential dangling references
- IllegalAssignmentError if assignment rule violated

Access to Physical Memory

- RawMemoryAccess for getting and setting bytes of physical memory
  - Reading memory mapped sensors
  - Device control
- LTPhysicalMemory and VTPhysicalMemory for mapping Java object to special memory areas
  - Make special memory available for Java objects, e.g. fast memory.

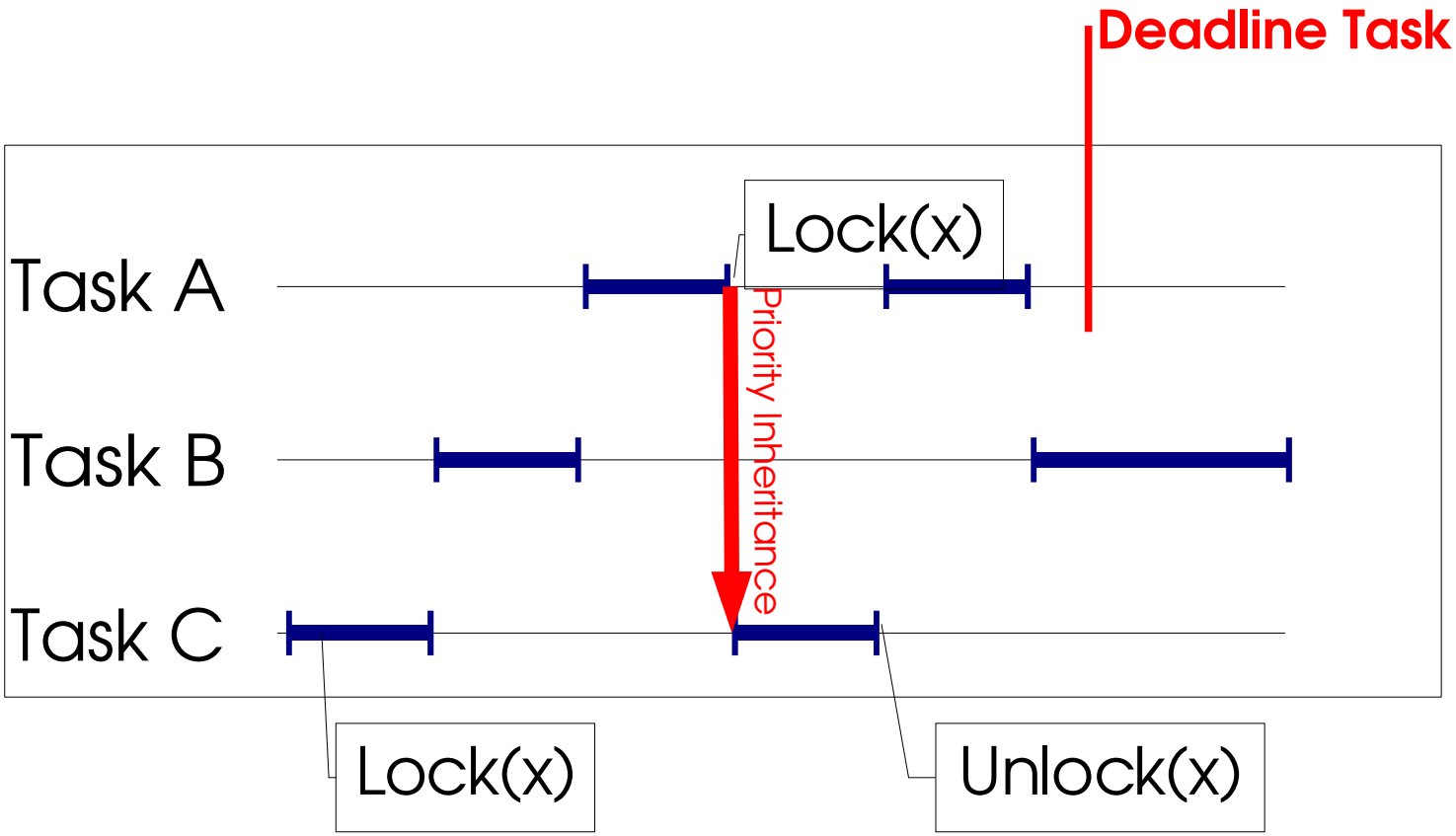
Synchronisation

Priority Inversion can be problematic



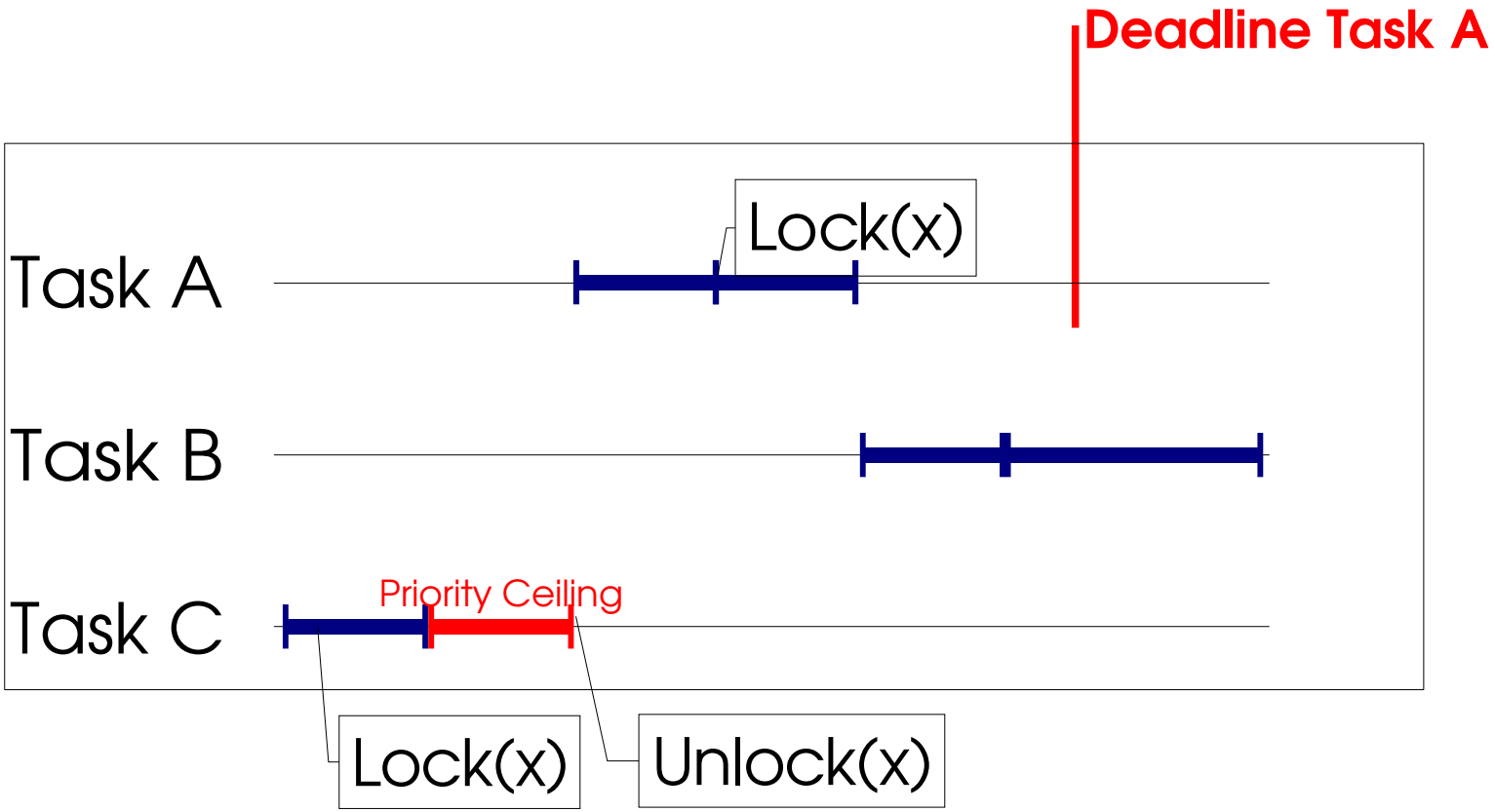
Synchronization

Priority Inheritance



Synchronisation

Priority Ceiling



Priority Inheritance vs Priority Ceiling

Priority Inheritance

+ Does not need any user configuration
- Deadlock can occur

Priority Ceiling

+ Is inherently deadlock free
- Requires manual selection of ceiling priority
- may block threads more often than needed

Asynchronous Event Handlers

- Bind Schedulable object to an event for immediate processing upon reception
- Provides an additional processing paradigm besides RealtimeThreads
- Light weight (>10,000 possible)
- Executed in a thread context but need not be bound to one

RTSJ & Realtime Garbage Collection

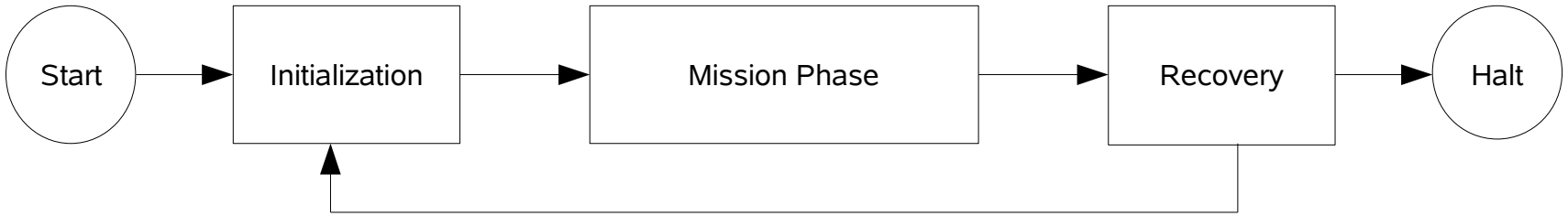
- The RTSJ provides necessary features for realtime programming
- Memory area restrictions can be relaxed
  - Can use RealtimeThread instead of NoHeapRealtimeThread
  - Heap allocation possible in realtime code
  - Synchronization possible with non realtime tasks without GC interference
  - GC does not interrupt thread execution

New: Safety Critical Java (SCJava)

- Aim: Java Subset, DO-178B, Lv. A certifiable
- OpenGroup Consortium with members aicas, Aonix, Boeing, Lockheed-Martin, Raytheon, SUN, Thales and many others
- Votes for aicas' specification proposal: 83%
- Will be base for JSR (Java Specification Request) within the Java Community
- SCJava will be closely related to RTSJ

SCJava Proposal

- Only RealtimeThreads are allowed
- No heap objects/ no GC
- Object allocation in initialisation phase only



- Thread priorities may not be changed
- Always priority ceiling emulation
- OutOfMemoryError may not occur
- Shared Objects must be annotated

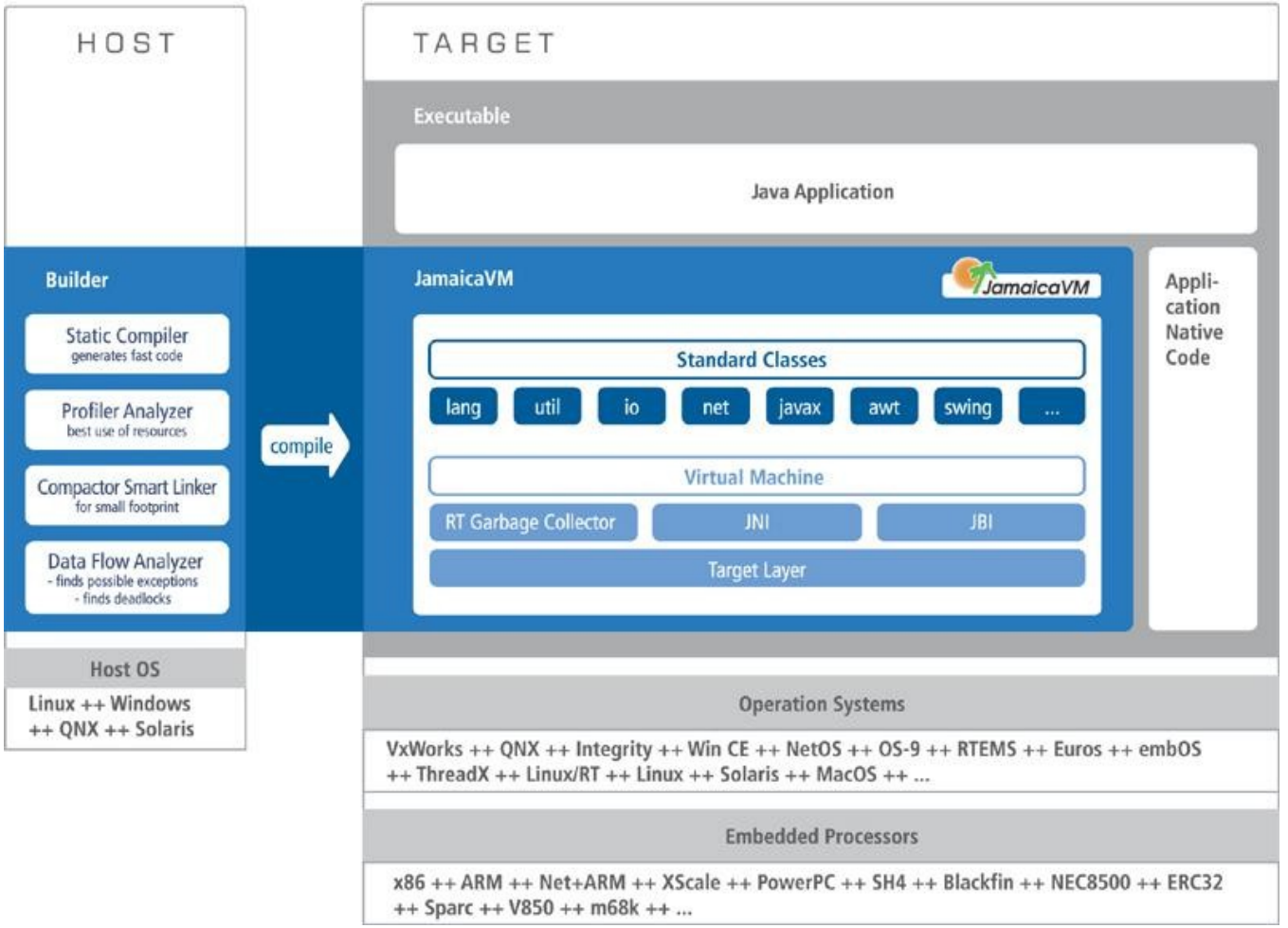
Class Libraries for SCJava

- Base: CLDC (IMP - Information Module Profile) with
  - Floating point support
  - Error Handling
  - JNI
  - RTSJ-Subset (javax.realtime)

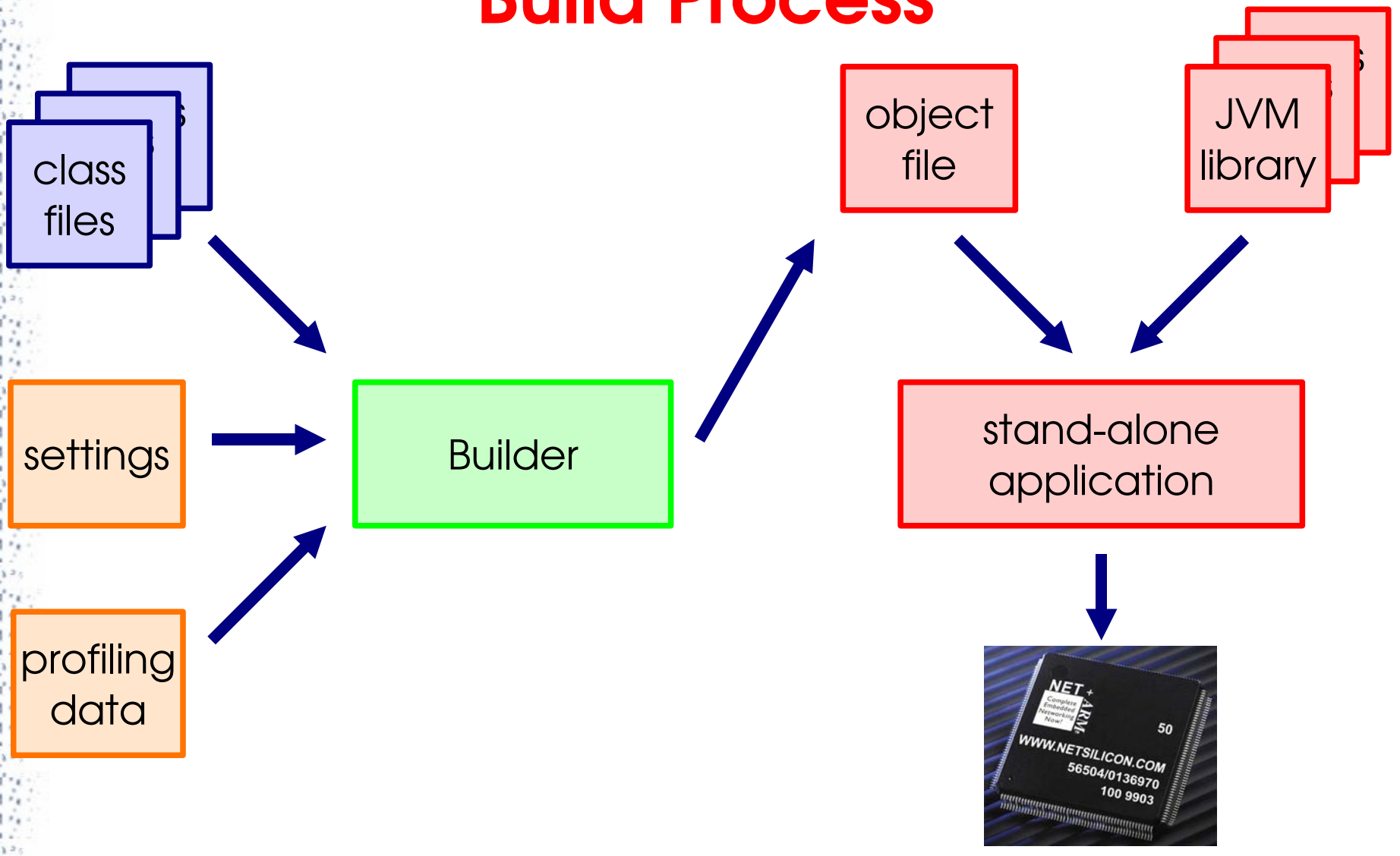
==> circa: lang, reflect, io, net, util, and realtime

- Exact library set has not been fixed
- Super sets for DO-178B, Level B and C will be specified.

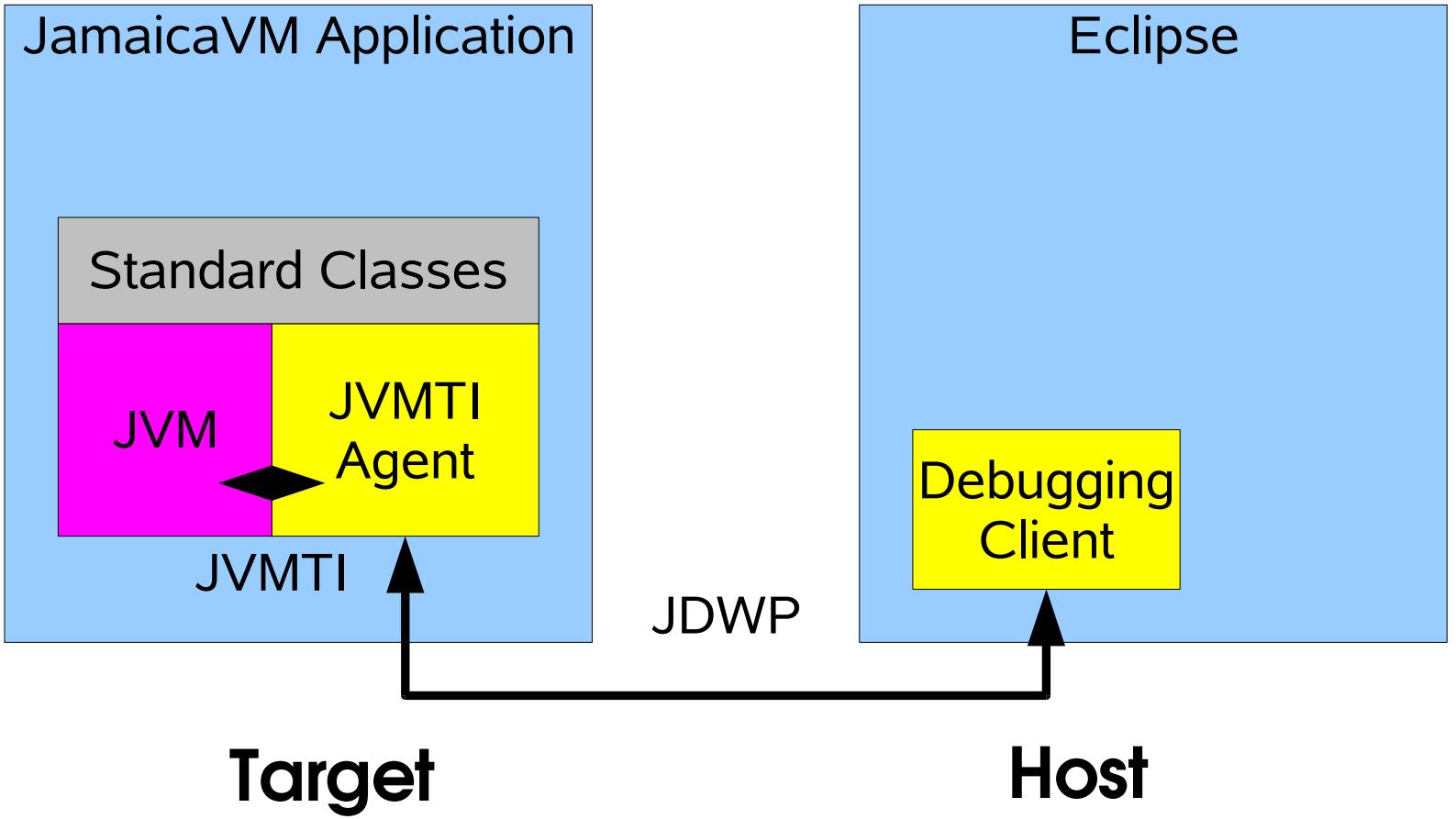
The JamaicaVM Toolset Overview



Build Process



Debugging and Monitoring in Java



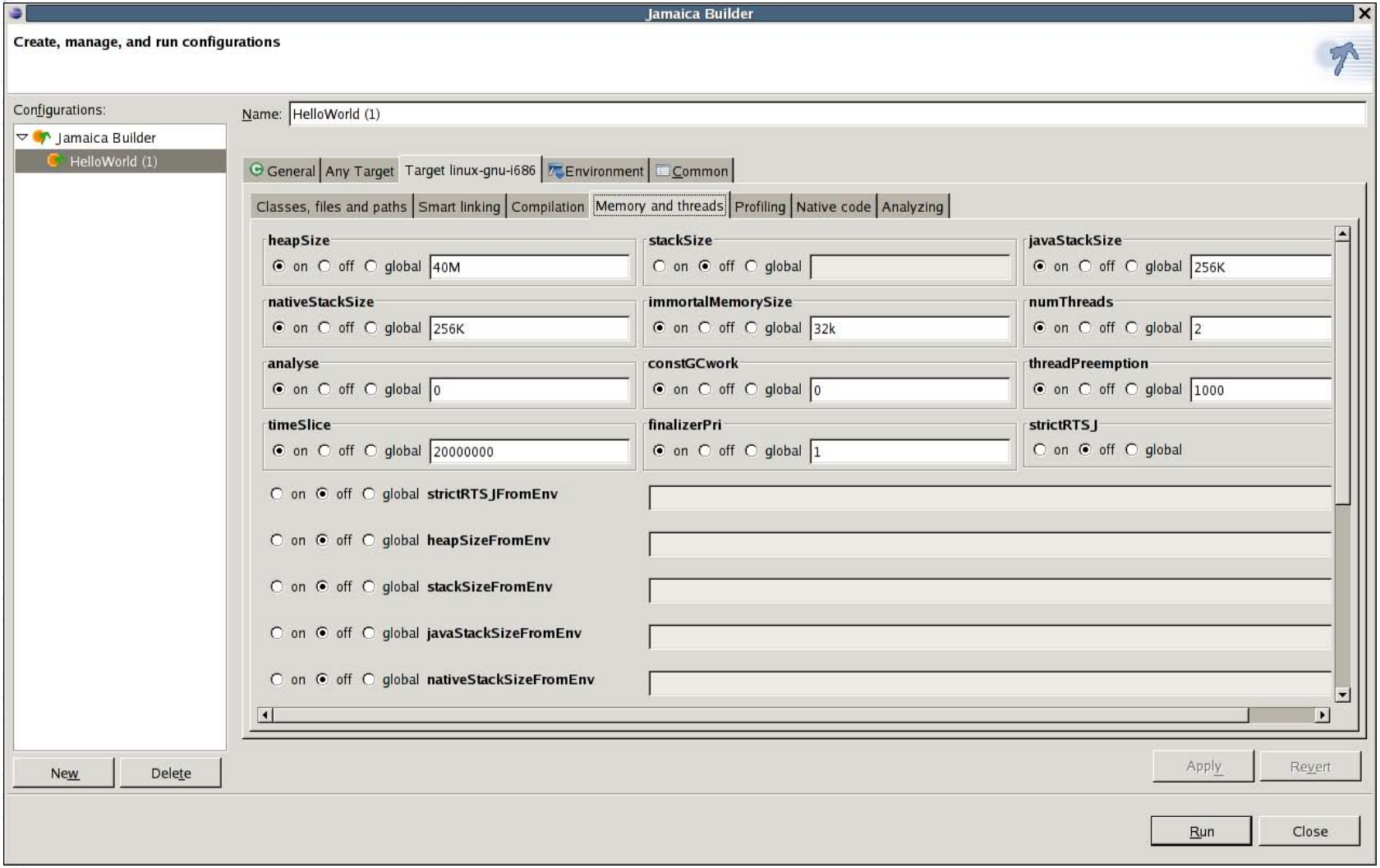
Debugging Capabilities

- Set breakpoints
- Stop and start threads
- Stepping
  - by byte code or line
  - into and out of methods
- Examine variables, objects, their fields, and array elements.

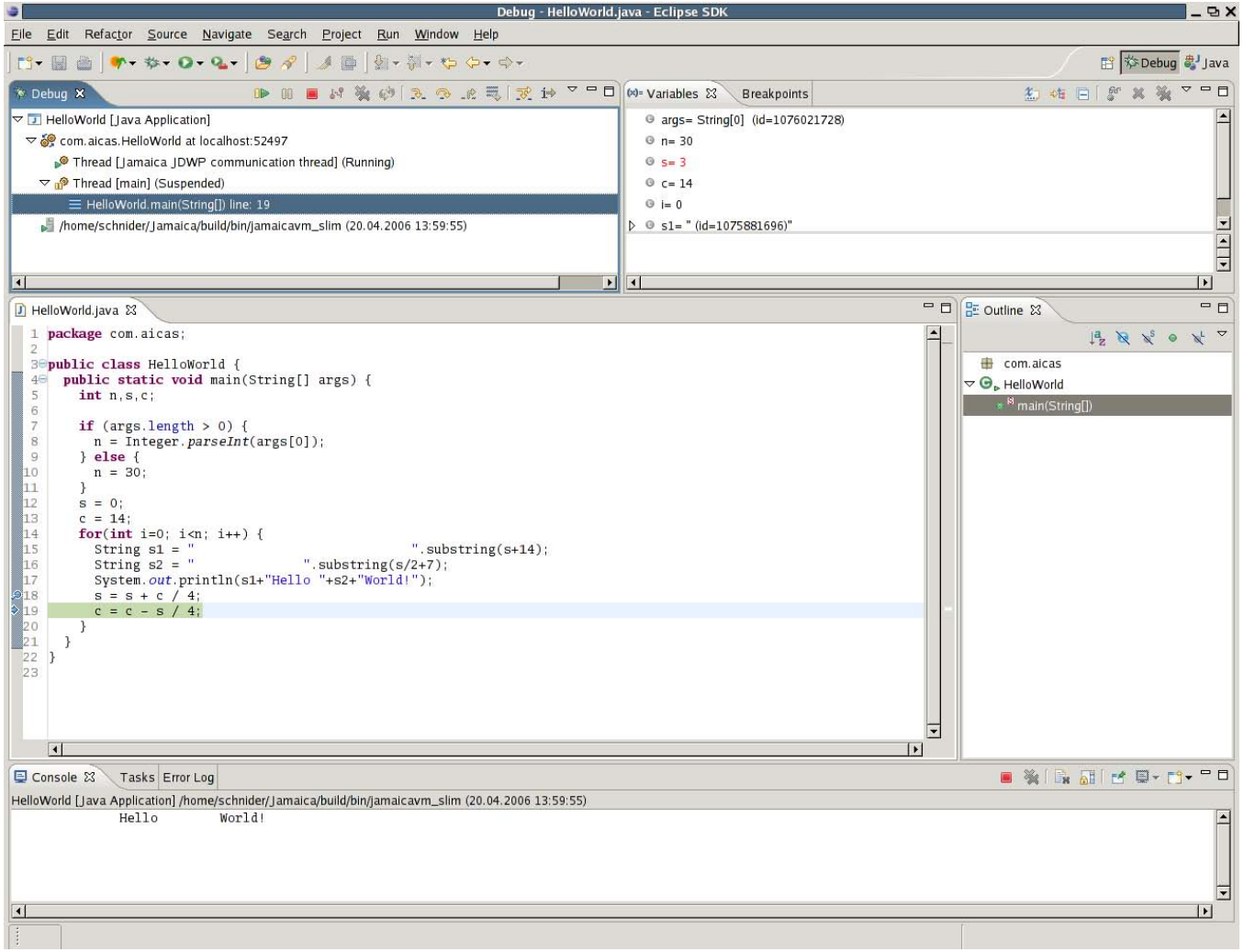
Debug Cycle

- Edit code in Eclipse
- Build executable w/ JamaicaVM Plug-in
- Download onto target
- Start executable
- Control application with debugger
  - Set breakpoints
  - Step and examine objects and variables
  - Continue execution

The JamaicaVM Builder Plugin



Eclipse Debugger & JamaicaVM



Data Flow Analysis

Additional tools can detect

- many runtime errors
  - Possible exceptions
  - Possible Deadlocks
- maximum stack usage

Runtime Errors

```
...
if (device instanceof MyDevice
{
  MySensor s = (MySensor) device.sensor;


  int value = s.reading();

  ...
}
...
```

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;


  int value = s.reading();

  ...
}
...
```

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException

ClassCastException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException

ClassCastException

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

device != null

NullPointerException

ClassCastException

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

device != null

NullPointerException

ClassCastException

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

device != null

NullPointerException ✔

ClassCastException

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

values(MyDevice.sensor)
contains only MySensor

ClassCastException

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException

values(MyDevice.sensor)
contains only MySensor

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException ✔

values(MyDevice.sensor)
contains only MySensor

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException ✔

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException ✔

null ∉ values(MyDevice.sensor)

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException ✔

null ∉ values(MyDevice.sensor)

NullPointerException

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException ✔

null ∉ values(MyDevice.sensor)

NullPointerException ✔

HIJA High-Integrity Java Application

Runtime Errors

```
...
if (device instanceof MyDevice)
{
  MySensor s = (MySensor) device.sensor;

  int value = s.reading();

  ...
}
...
```

NullPointerException ✔

ClassCastException ✔

NullPointerException ✔

HIJA High-Integrity Java Application

Java in Device Systems

- Java brings
  - Higher productivity (Stand Libraries, Code Reuse)
  - Platform independence
  - Robustness (type and pointer safety)
  - Flexibility

We make Java realtime capable!

Thank you for your attention!