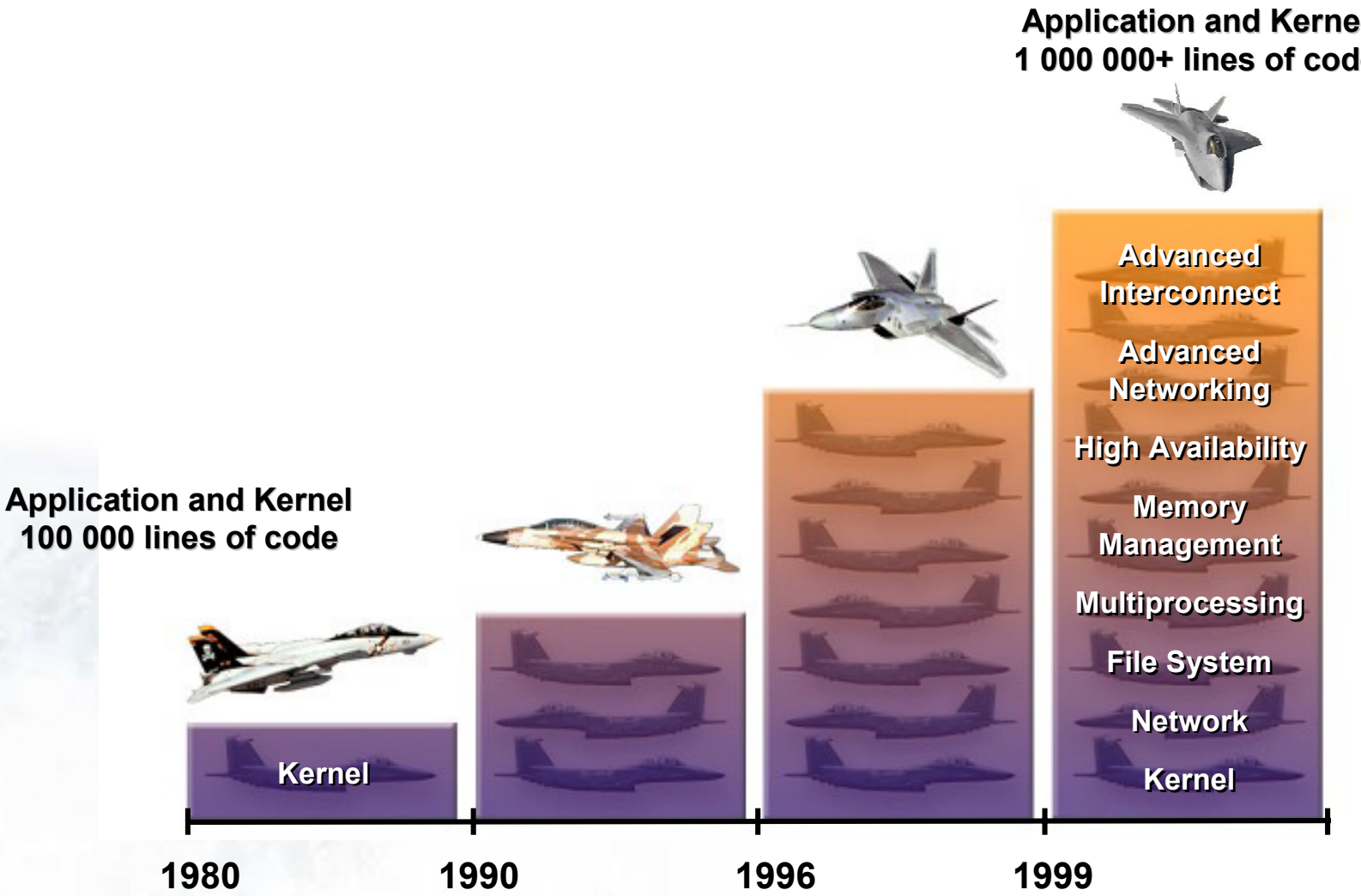
Developing Safety-Critical Systems with GNAT Pro & VxWorks

Trend 1: Increased Software Complexity



Trend 2: Need to Certify Military Avionics

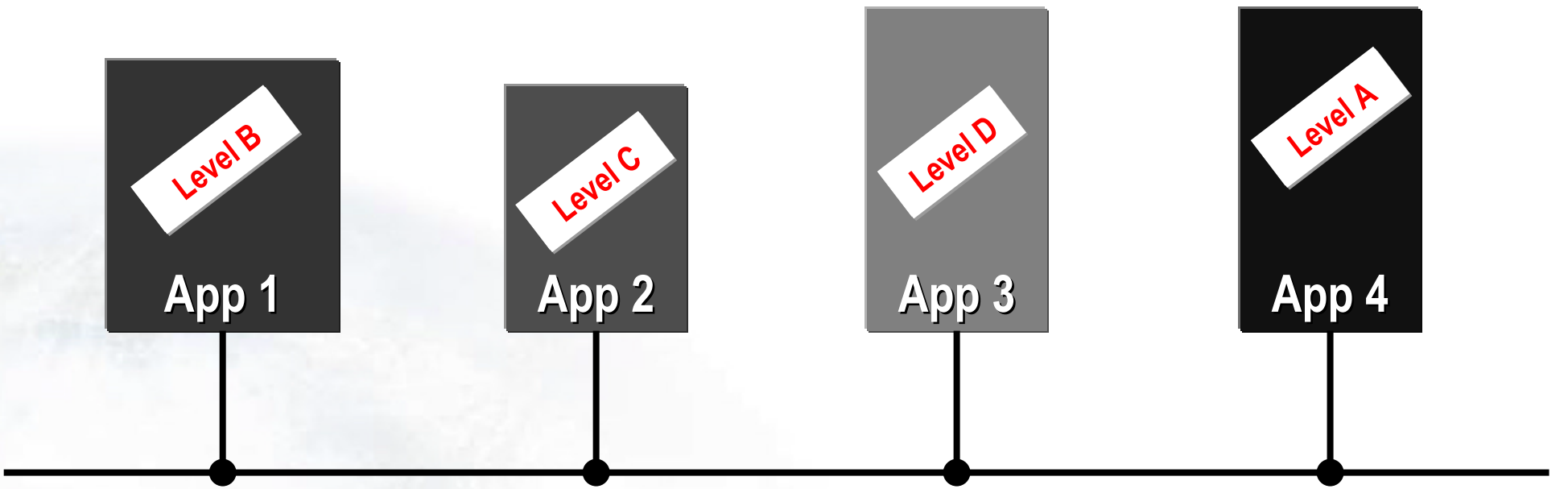
Example: RTCA DO-178B & EUROCAE/ED-12B (civil avionics)

Failure Condition	Software Level
Catastrophic	Level A
Hazardous/Severe - Major	Level B
Major	Level C
Minor	Level D
No Effect	Level E

Other examples: Def-Stan 055

Typical Avionic System Architecture

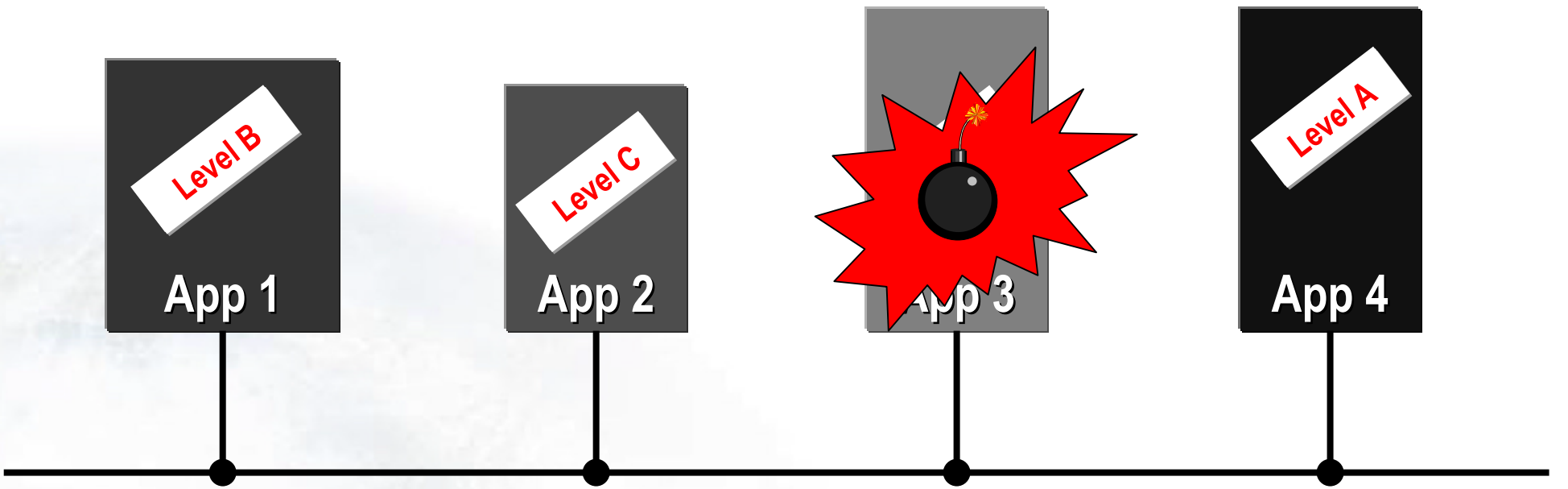
Designed as a federated architecture of dedicated black-boxes



This Architecture is Inherently Robust

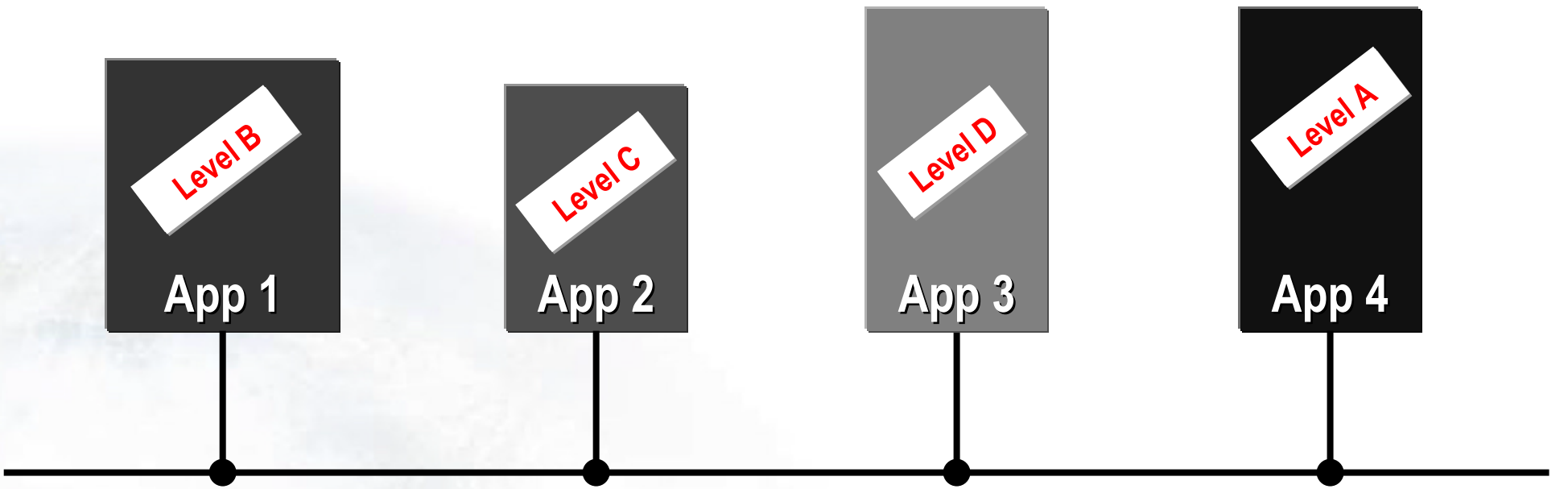
The Apps are physically protected from one another.

If an App fails, it does not affect the others.



This Architecture is Inherently Robust

Furthermore you can restart the App after it failed

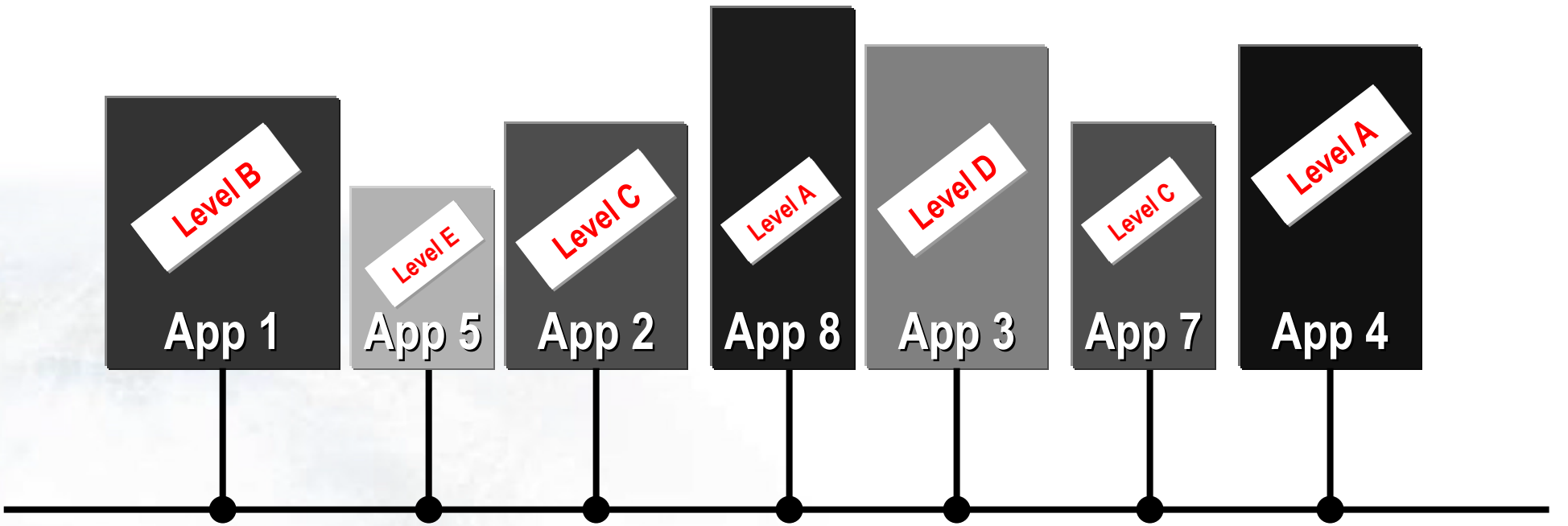


This Architecture is EXPENSIVE to Build

In terms of

- COST, POWER, WEIGHT

Adding Apps means Adding boxes

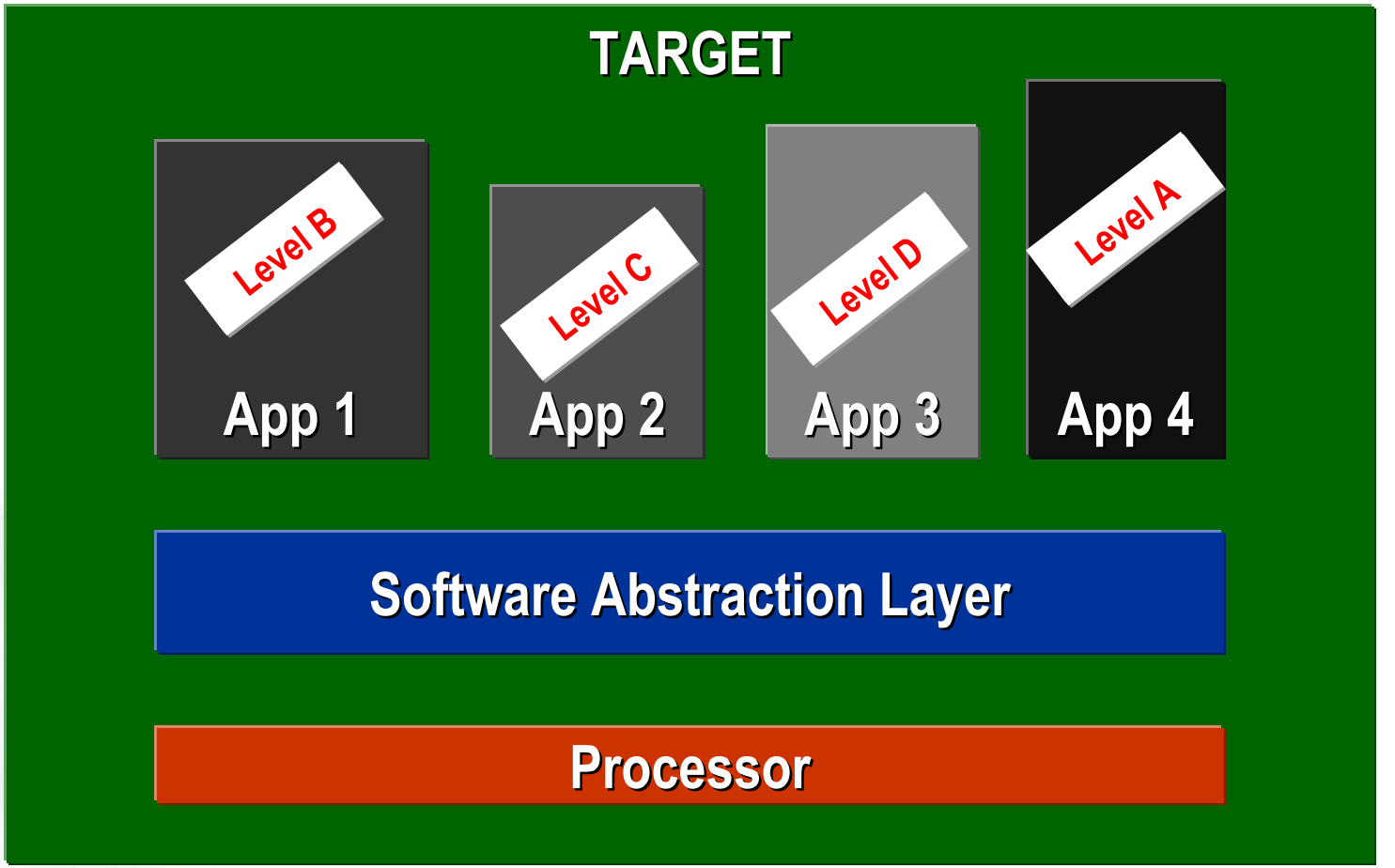


… And is EXPENSIVE to Maintain

Integrated Modular Avionics (IMA)

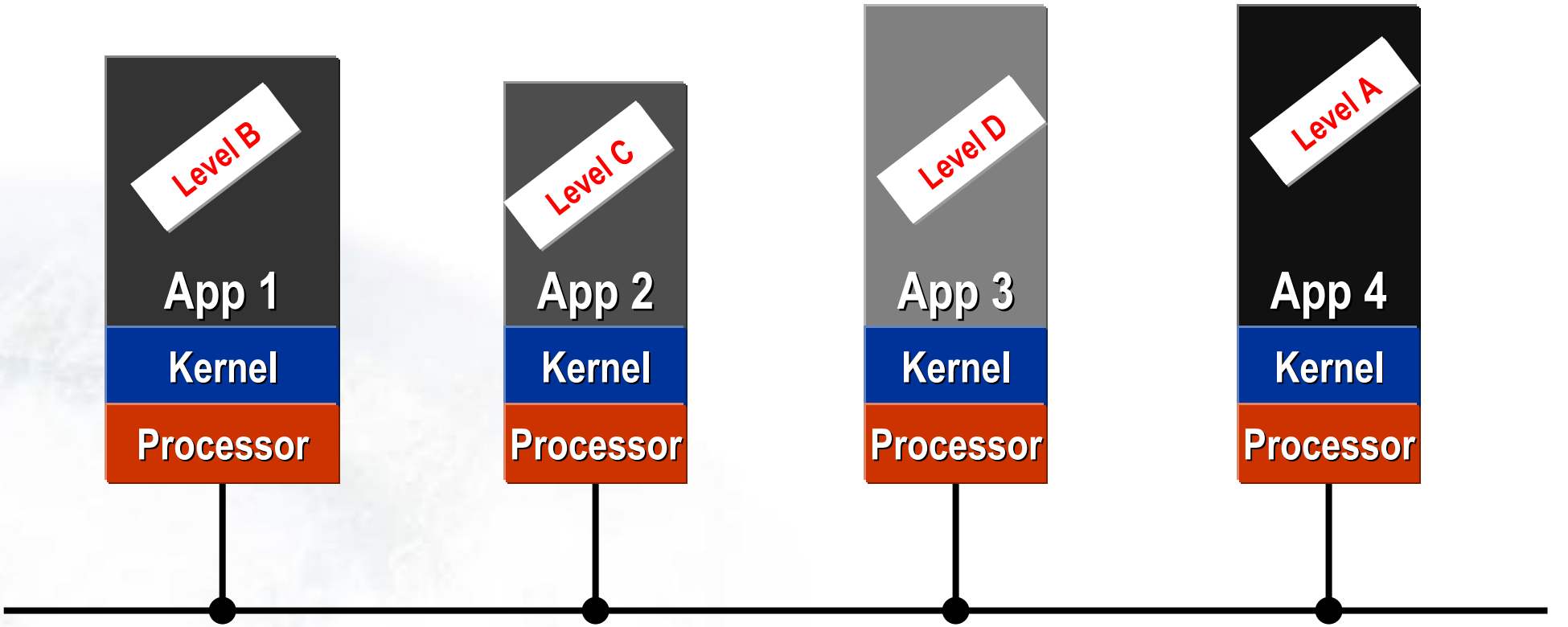
- **A new architecture model has been developed**
- **Common processing subsystems**
  - Allows multiple apps to share computing resources
  - Reduces the number of boards in a plane
- **Software abstraction**
  - Isolate the application from the underlying HW architecture
  - Reduce the impact of HW obsolescence

IMA: Conceptual View



Before IMA

- Kernel & App in single address space
- Protection achieved by means of physical partitioning



Software Support for IMA

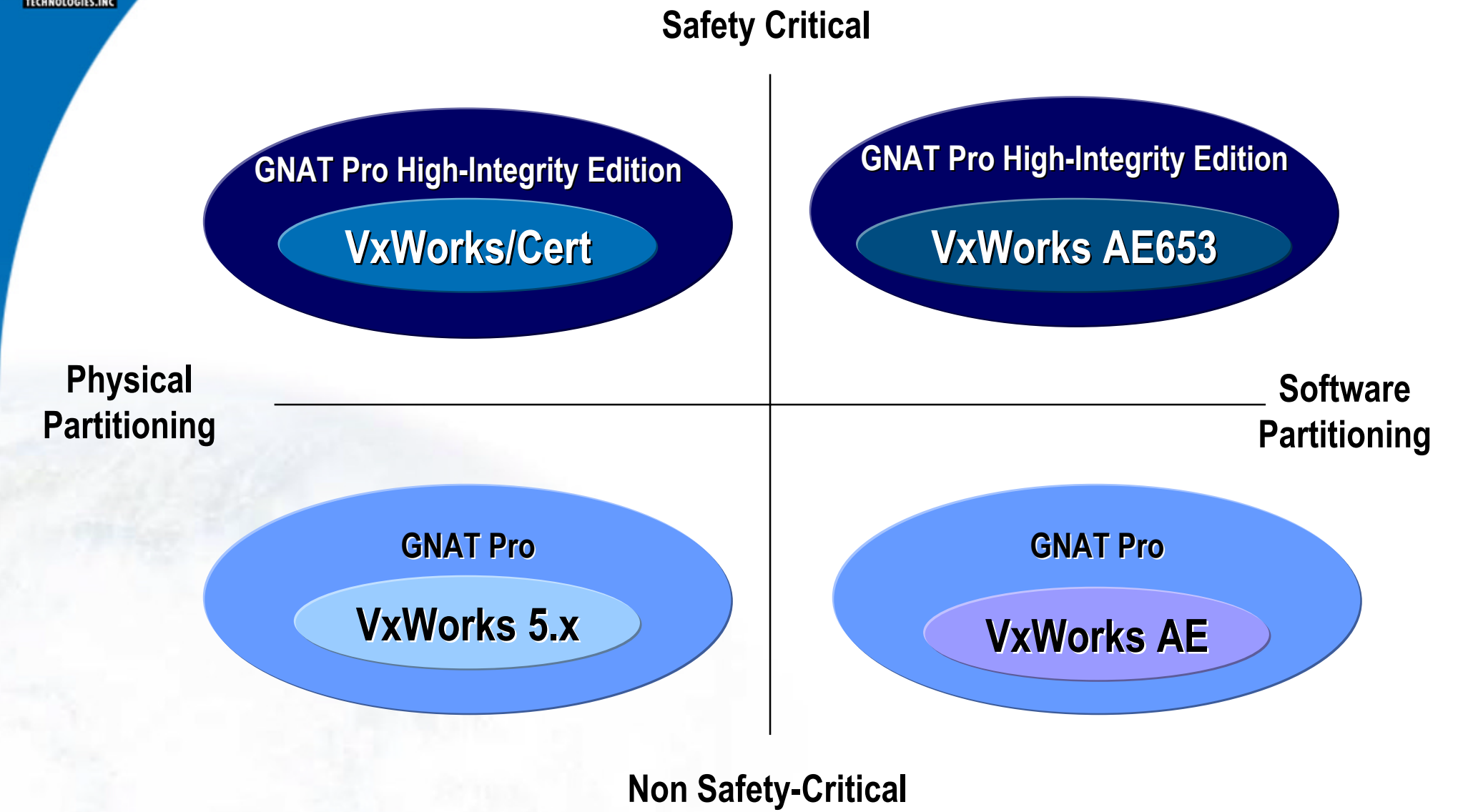
Apps must be partitioned (protected from one another)

- **Spatial Partitioning**
  - Memory protection
  - Resource protection
- **Temporal Partitioning**
  - CPU protection
- **Some standards addressing software support from IMA:**
  - E.g. ARINC 653, RTCA/DO-255, EUROCAE WG-60

Protection

Memory Protection	An illegal memory access by an App cannot bring down the whole system
Resource Protection	An App cannot exhaust all the kernel resources
Temporal Protection	An App cannot starve other Apps by keeping all the CPU to itself

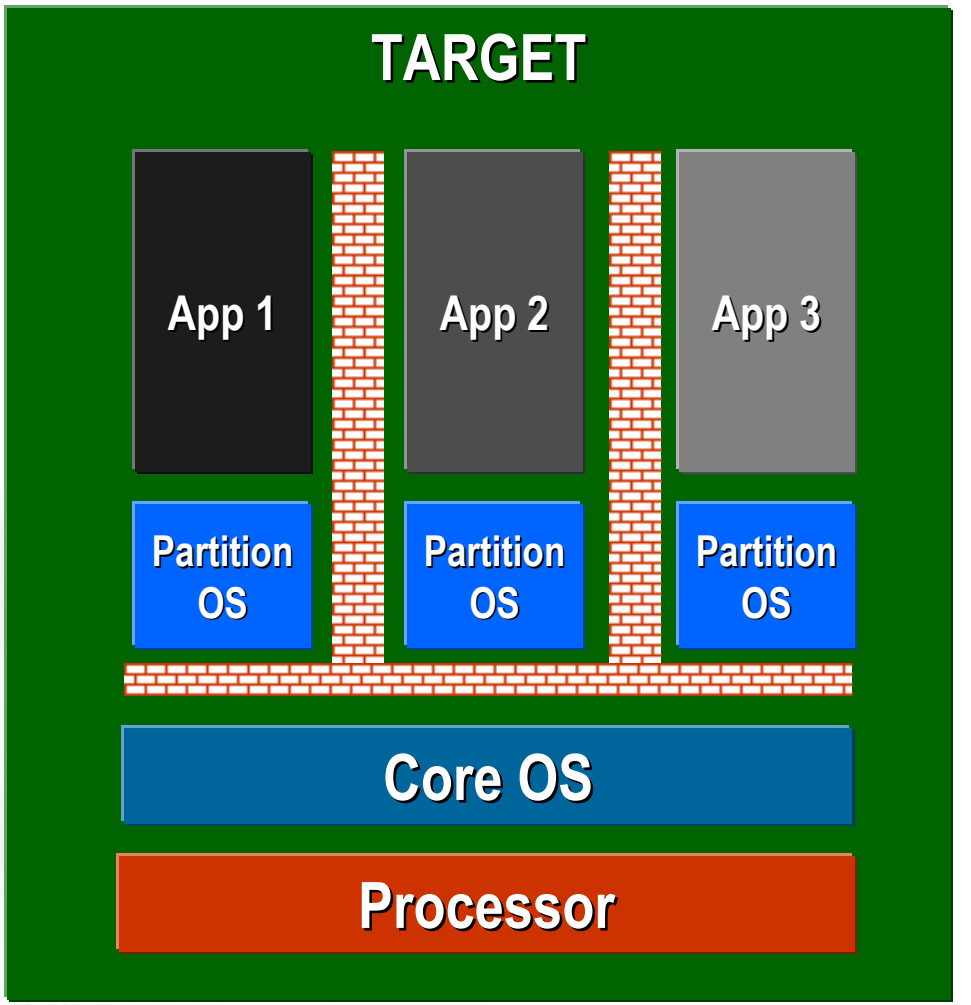
Product Families



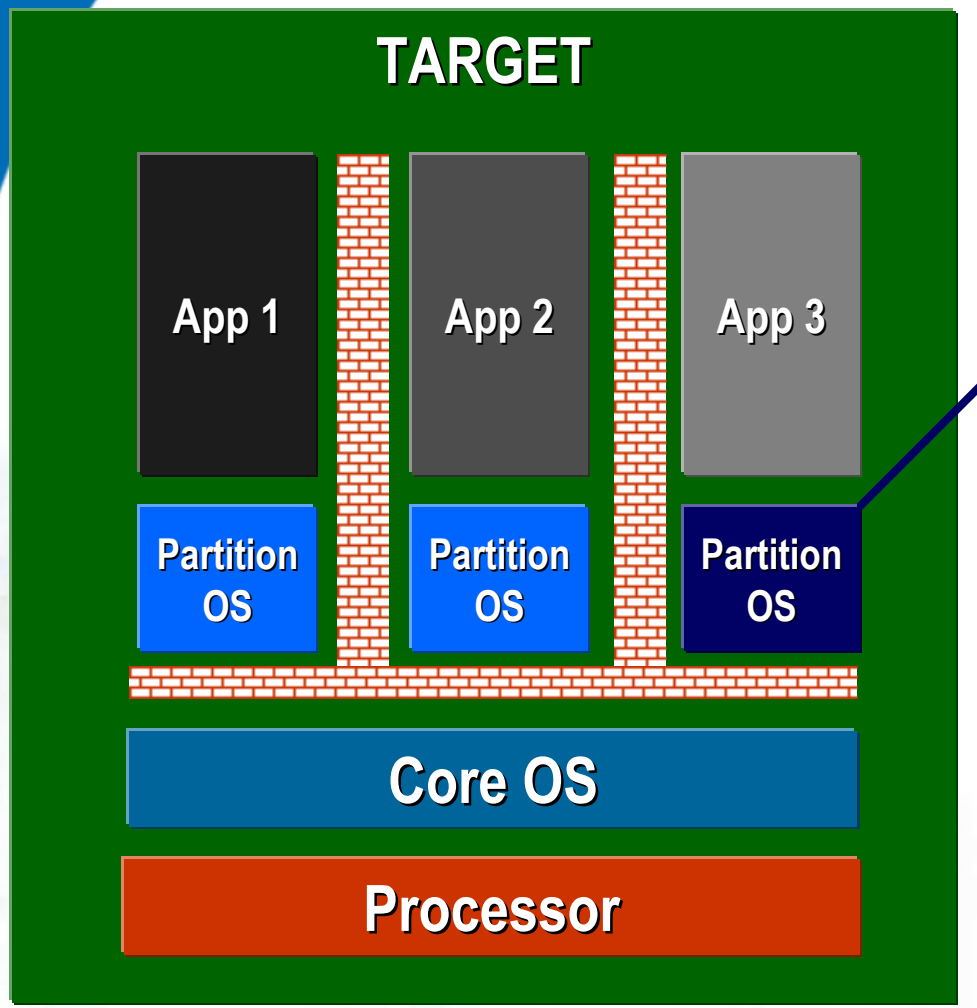
VxWorks Certifiable Kernels

- **VxWorks/Cert**
  - DO-178B Level A certifiable multitasking RTOS
- **VxWorks AE653 (ARINC 653)**
  - DO-178B Level A certifiable multitasking RTOS
  - Spatial Partitioning (memory protection, resource protection)
  - Temporal Partitioning (ARINC 653 scheduler)

VxWorks AE653 Architecture



Partition OS



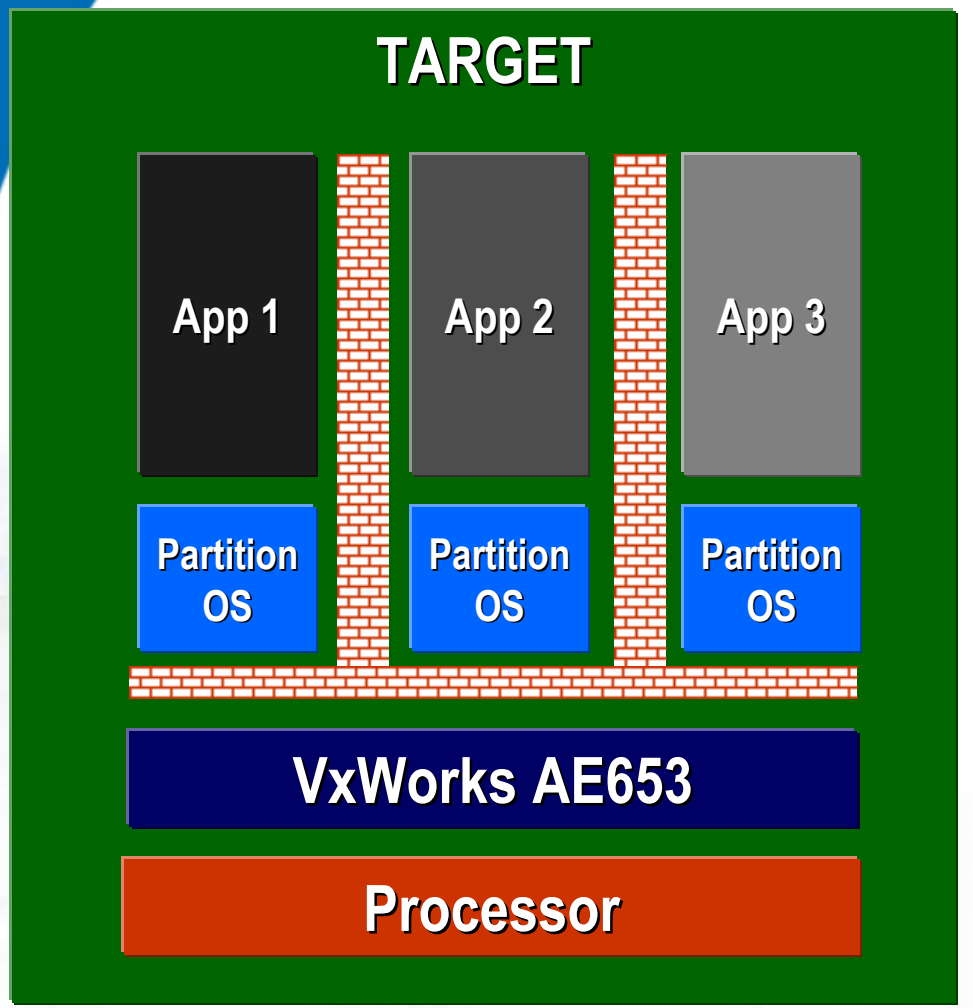
VxWorks 5.x or VxWorks/Cert microkernel

- Partition OS threads are called vThreads
- vThreads run as user-level threads
- Priority-based preemptive scheduler
- Partition OS provides the following APIs to the app within the partition:
  - ARINC 653 API
  - POSIX API
  - VxWorks API

vThreads

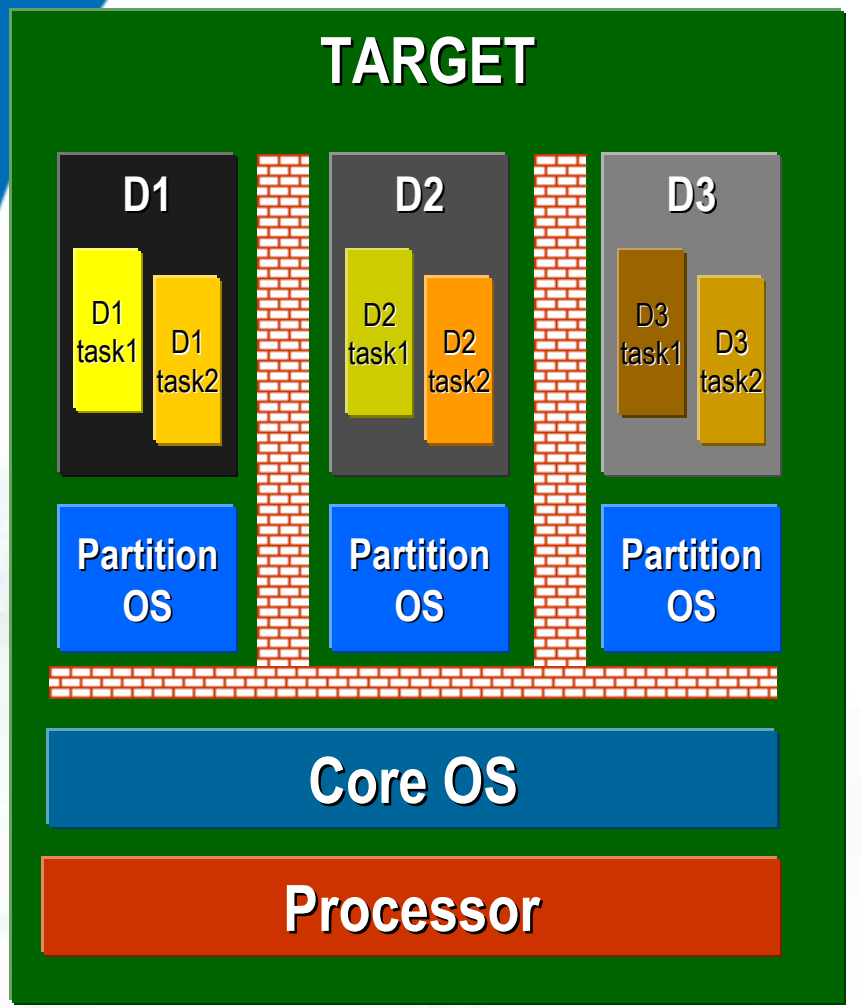
- **vThreads do not address kernel objects in other partitions**
- **vThreads service API calls from the app**
- **vThreads allocate resources owned by the partition to the app**
- **vThreads pass on kernel calls which need to be serviced by the Core OS**
  - But only after the input parameters have been validated
  - The message-passing implementation between the Partition & Core OS is private

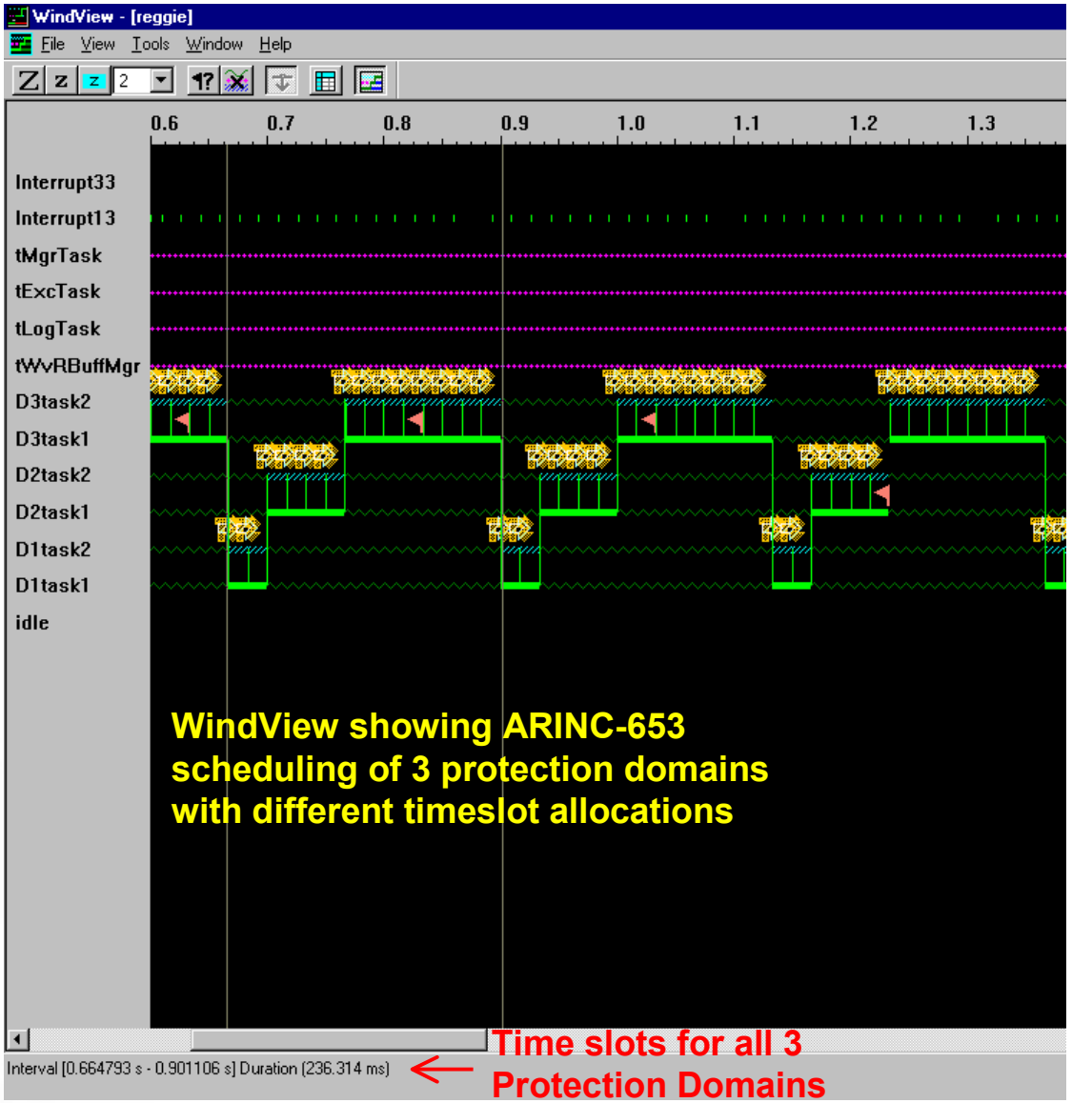
Core OS: VxWorks AE Technology



- Allocation of system resources to partitions
- Detection of attempted violations to partition boundaries
- Kernel protection (user/supervisor protection)
- Overrun protection: stack, heap, CPU lockouts
- Resource reclamation: heap, stack, code & data memory
- ARINC 653 Partition Scheduler
- Support for inter-partition communication via distributed messaging

ARINC 653 Scheduler





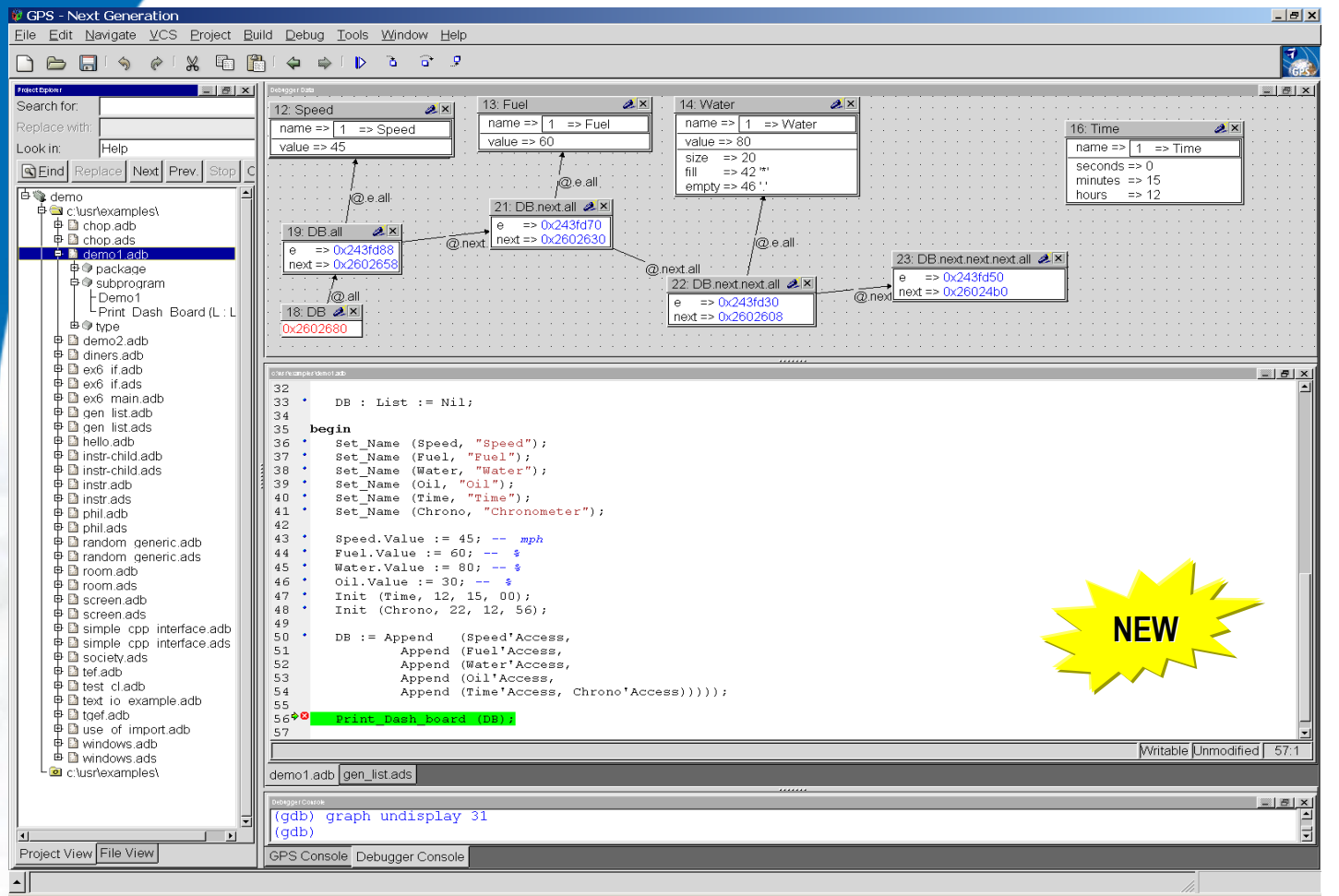
GNAT Pro High-Integrity Edition

The Ada 95 Solution to Develop Safety-Critical Systems

GNAT Pro High-Integrity Edition (HIE)

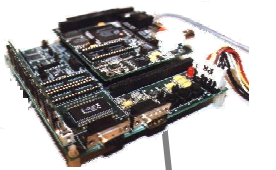
- **Choice of 3 profiles:**
  - ***Profile 1***: No Ada run-time (no tasking)
  - ***Profile 2***: Ravenscar tasking
  - ***Profile 3***: User-defined
- **Upwardly compatible with the SPARK profile**
  - The high-integrity subset of Ada by Praxis

GNAT Pro: A Single Visual Environment



NEW

Target Development



Host Development

Beta: June
FCS: Q4

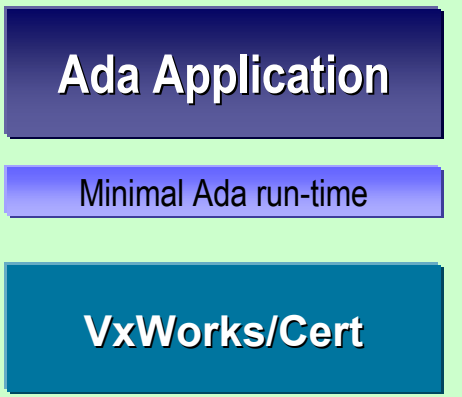
GNAT Pro HIE: 3 Profiles

***Profile 1*: No Ada Run-Time**

- No Ada tasking
- Use VxWorks tasks directly

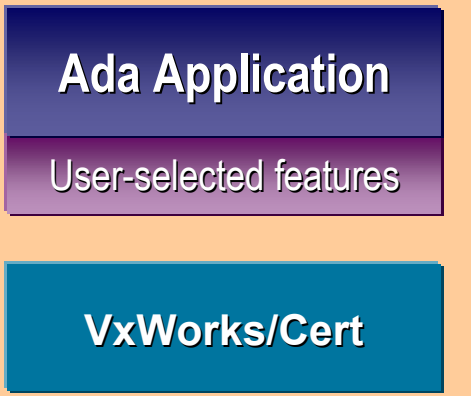
***Profile 2*: Ravenscar**

- Safe Ada 95 tasking
- Minimal Ada run-time



***Profile 3*: User-defined**

- Pick & choose
- Certify what you need



Profile 2: Ravenscar Tasking

- **Ravenscar Ada 95 tasking subset**
  - Defined in Ravenscar, UK in 1997, to be part of the next Ada 0Y standard
  - Satisfies the certification requirements of safety-critical real-time systems
  - Allows schedulability analysis (in particular RMA)
- **These design objectives are part of certifiable VxWorks kernels**
- **Only ~200 Ada SLOCs required to implement Ravenscar on top of VxWorks**
- **Very efficient & compact implementation**

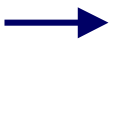
Profile 3: Extend the Ada Subset Allowed

- **Profile 3 allows users to define their your own Ada subset**
- **Why is that useful?**
- **To certify an existing application**
  - It may be cheaper to customize the Ada subset and certify it
  - Rather than recode the application to meet existing Ada profiles
  - Especially if you do not have to certify at the highest integrity level

Example: Allowing Integer Exponentiation

Ada source

```
function  F (X, N :    Integer)
                  return   Integer
is
      Y : Integer := X + 1;
begin
      return  Y ** N;
end  F;
```



GNAT Pro HIE compiler

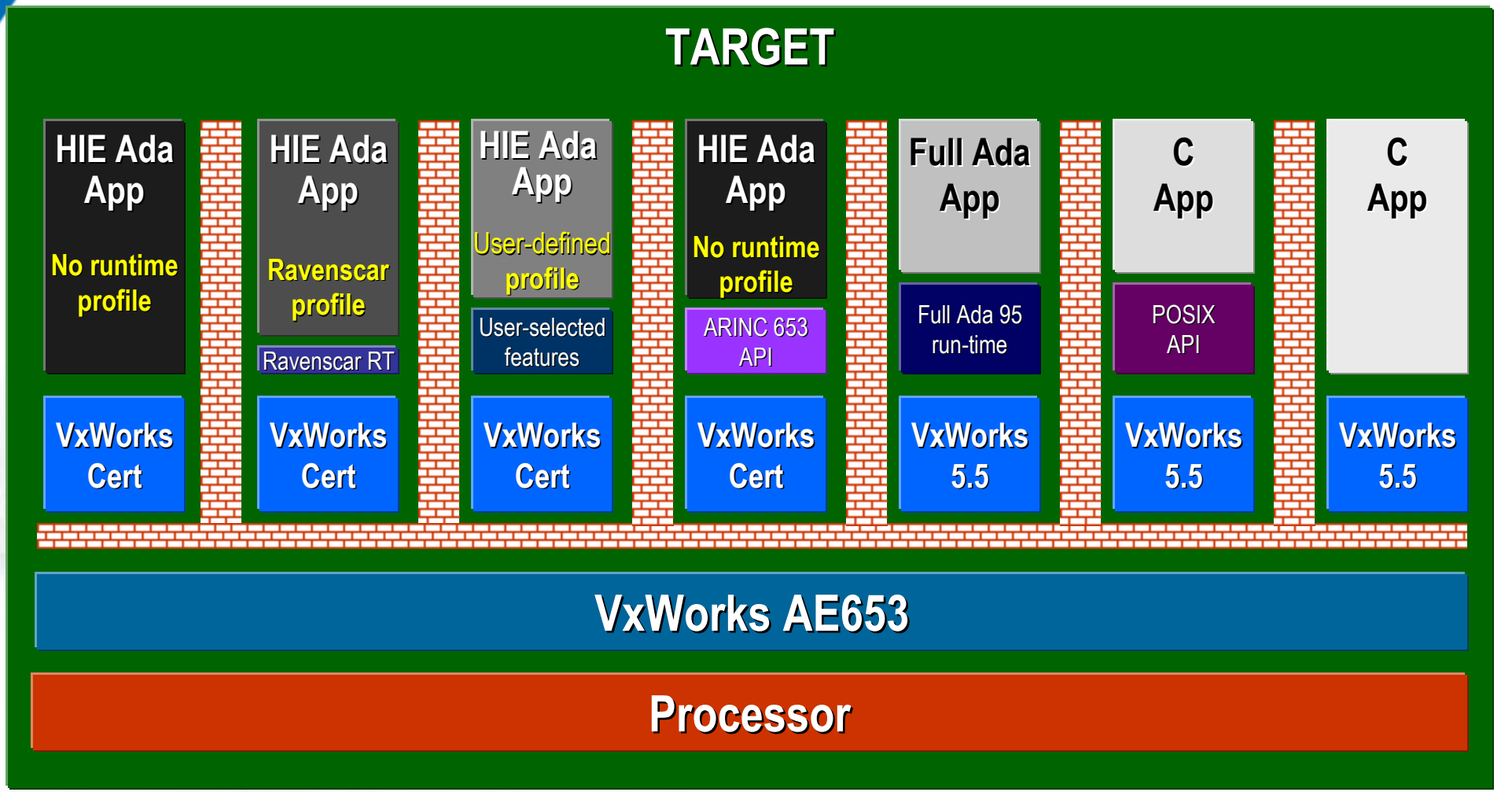
Assembly

```
stwu 1,-8(1)
mflr 0
stw 0,12(1)
addi 3,3,1
bl system__exn_int__exn_integer
lwz 0,12(1)
mtlr 0
addi 1,1,8
blr
```

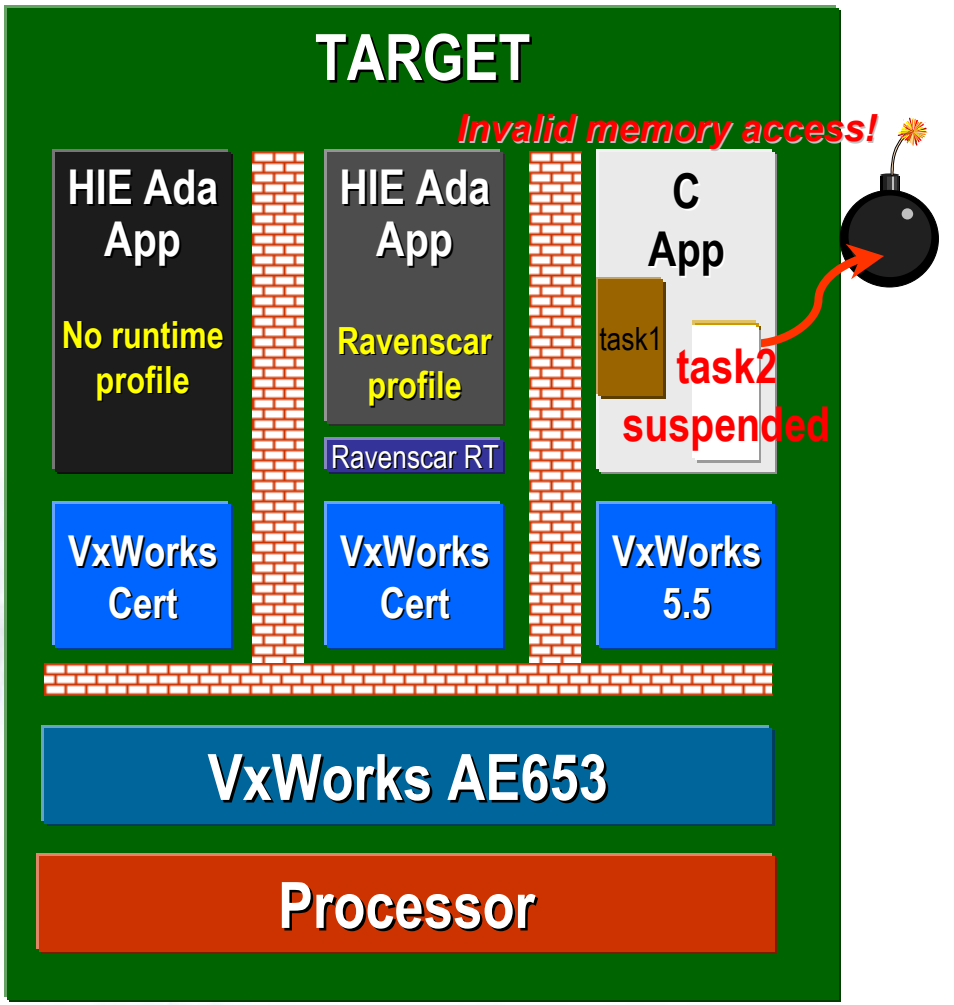


```
-- GNAT Pro run-time unit implementing integer "**"
package  System.Exn_Int  is
      pragma  High_Integrity;
      function Exn_Integer Left : Integer; Right : Natural) return Integer;
end System.Exp_Int;
```

AE653 Supports Heterogeneous Apps



AE653 Memory Protection Model



Other applications unaffected!

AE653 kernel unaffected!

AE653 Memory Protection Model

What happens when a task tries to access protected memory?

- A hardware exception is generated
- The kernel sends a signal to the errant task
- If no signal handler available, the task will be suspended

This prevents an errant pointer access from corrupting another application

GNAT Pro & VxWorks: Key Points

- DO-178B certifiable solution up to LEVEL A
- Supporting the development of IMA applications
- With a tightly integrated, friendly & tool-rich environment

GNAT Pro HIE & VxWorks

The solution of choice for safety-critical systems