

Configurable Memory Protection by Aspects

Daniel Lohmann
Jochen Streicher
Wanja Hofer
Olaf Spinczyk
Wolfgang Schröder-Preikschat



Department of Computer Science IV
Distributed Systems and Operating Systems
Friedrich-Alexander University Erlangen-Nuremberg

<http://www4.cs.fau.de/>

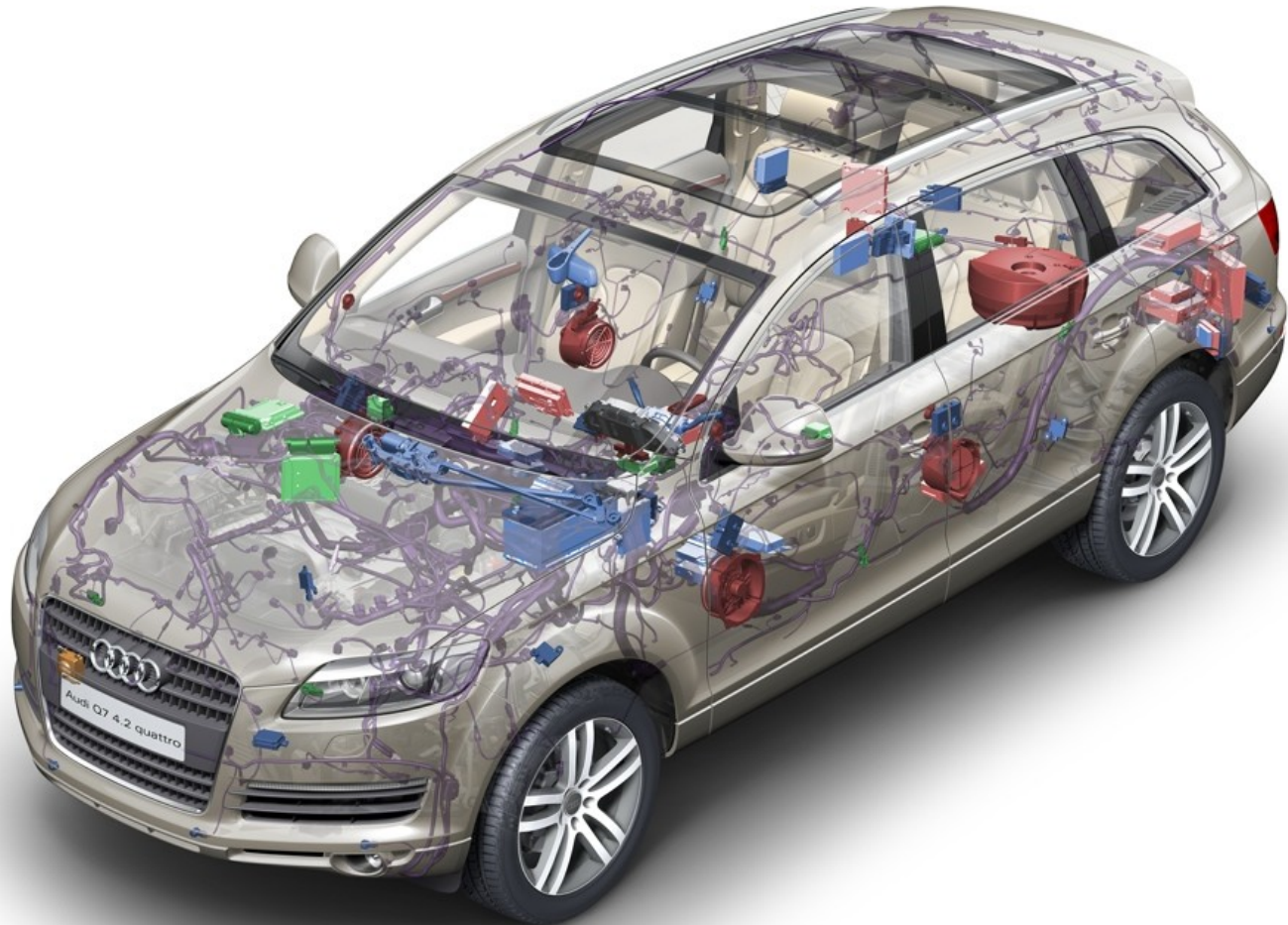


Talk Outline

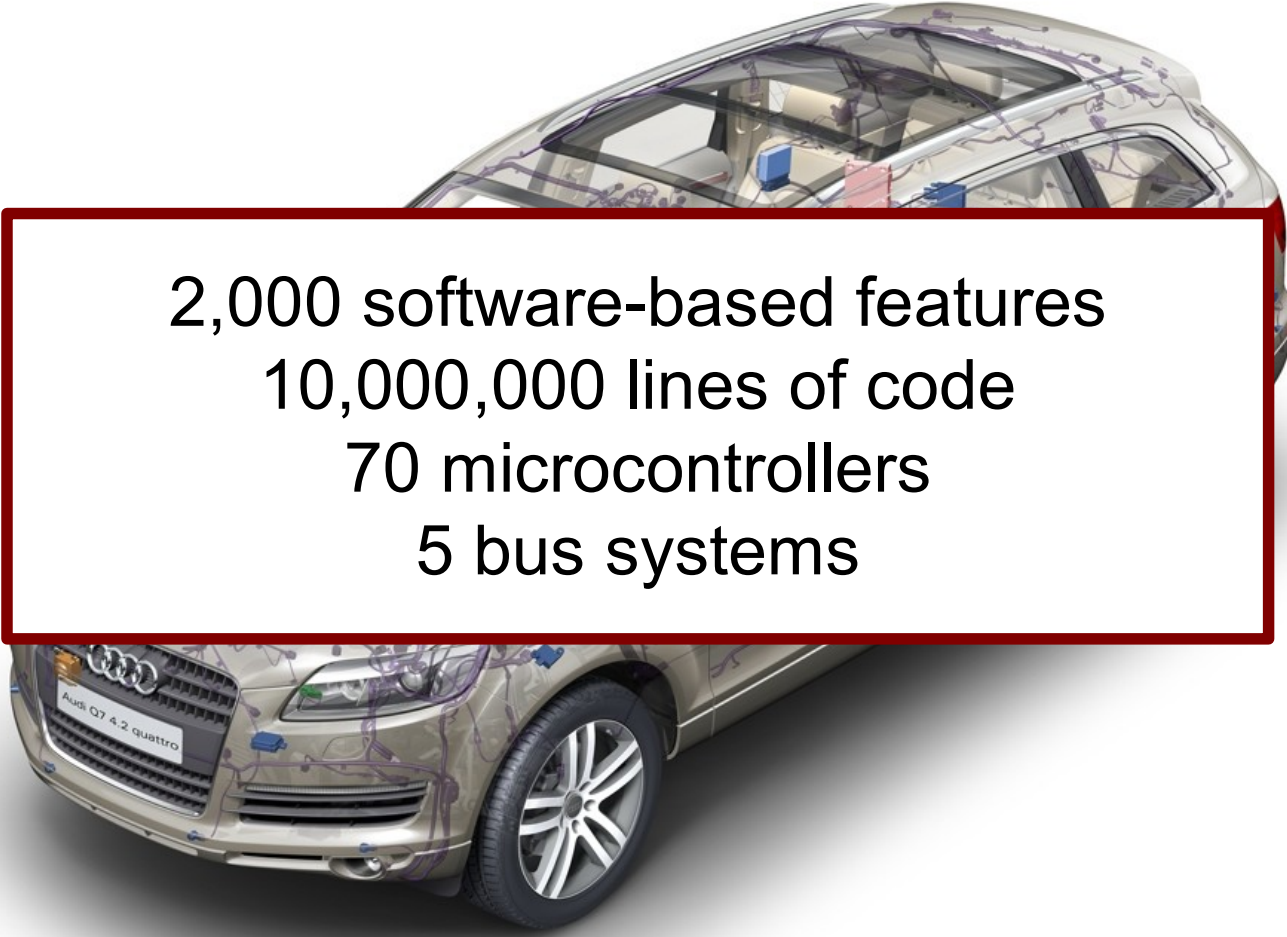
- Motivation
- Aspect-Oriented Programming
- Configurable Memory Protection
- The CiAO Project



Motivation



Motivation



2,000 software-based features
10,000,000 lines of code
70 microcontrollers
5 bus systems



Motivation – AUTOSAR

- Consolidation on few but more powerful microcontrollers

- AUTOSAR:



- “Automotive open system architecture”
- Core members: Mercedes, BMW, VW, Toyota, Ford, etc.
- Standard for automotive software

- AUTOSAR OS scalability classes:

- Timing protection
- **Memory protection**



Motivation – AUTOSAR

- Consolidation on few but more powerful microcontrollers

- AUTOSAR:



- “Automotive open system architecture”
- Core members: Mercedes, BMW, VW, Toyota, Ford, etc.
- Standard for automotive software

- AUTOSAR OS scalability

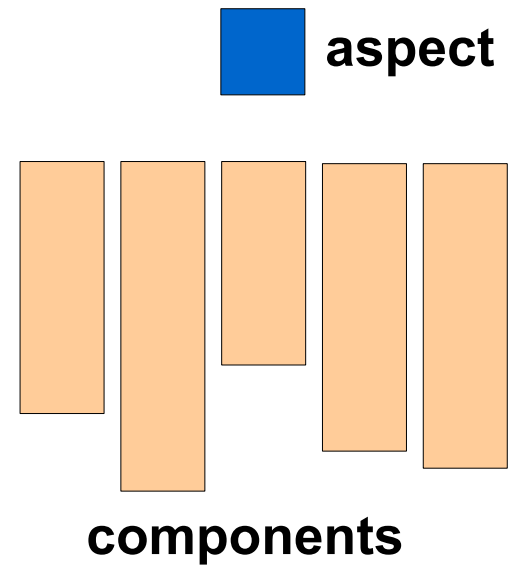
- Timing protection
- **Memory protection**

Cross-Cutting Concerns



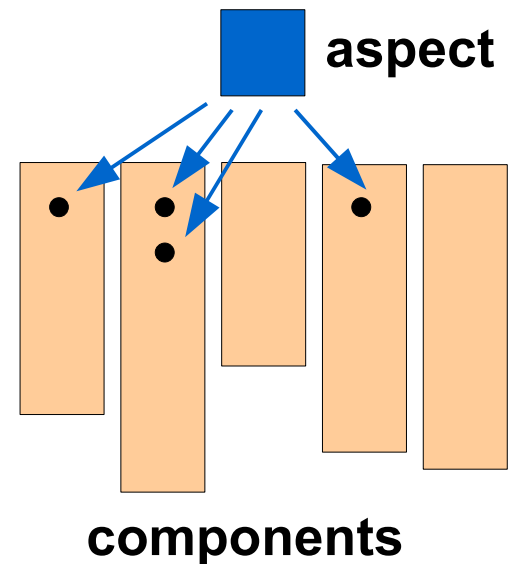
AOP – Short Introduction

- Encapsulation of (cross-cutting) concerns in **aspects**



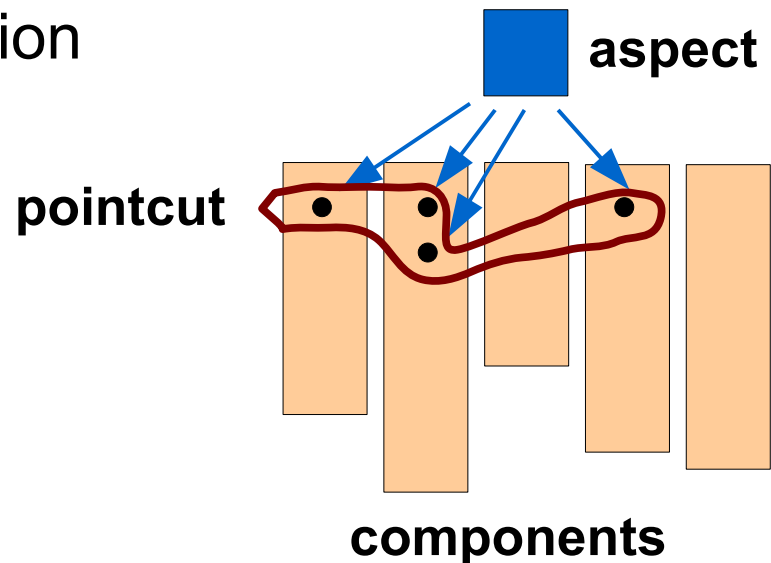
AOP – Short Introduction

- Encapsulation of (cross-cutting) concerns in **aspects**
- Aspects give **advice** to **join points** in the target system



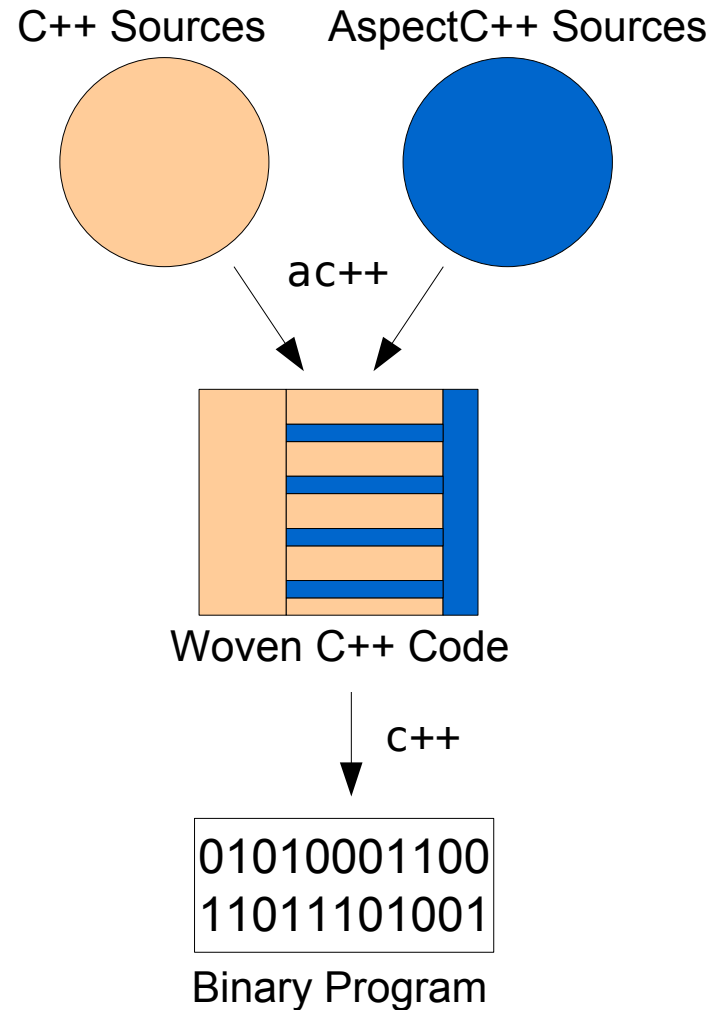
AOP – Short Introduction

- Encapsulation of (cross-cutting) concerns in **aspects**
- Aspects give **advice** to **join points** in the target system
- Set of join points described by a **pointcut** expression



AOP – AspectC++

- Extension to C++
- Source-to-source weaver



AOP Real-World Example: AUTOSAR

AUTOSAR OS specification:

“If interrupts are disabled and any OS services, excluding the interrupt services, are called outside of hook routines, then the operating system shall return E_OS_DISABLEDINT.”



AOP Real-World Example: AUTOSAR

AUTOSAR OS specification:

“If interrupts are disabled and any OS services, excluding the interrupt services, are called outside of hook routines, then the operating system shall return E_OS_DISABLEDINT.”

```
aspect DisabledIntCheck {
  advice call (pcOSServices () && ! pcInterruptServices ())
    && ! within (pcHookRoutines ()) : around () {
    if (interruptsDisabled ()) {
      * tjp->result () = E_OS_DISABLEDINT;
    } else {
      tjp->proceed ();
    }
  }
};
```

DisabledIntCheck.ah

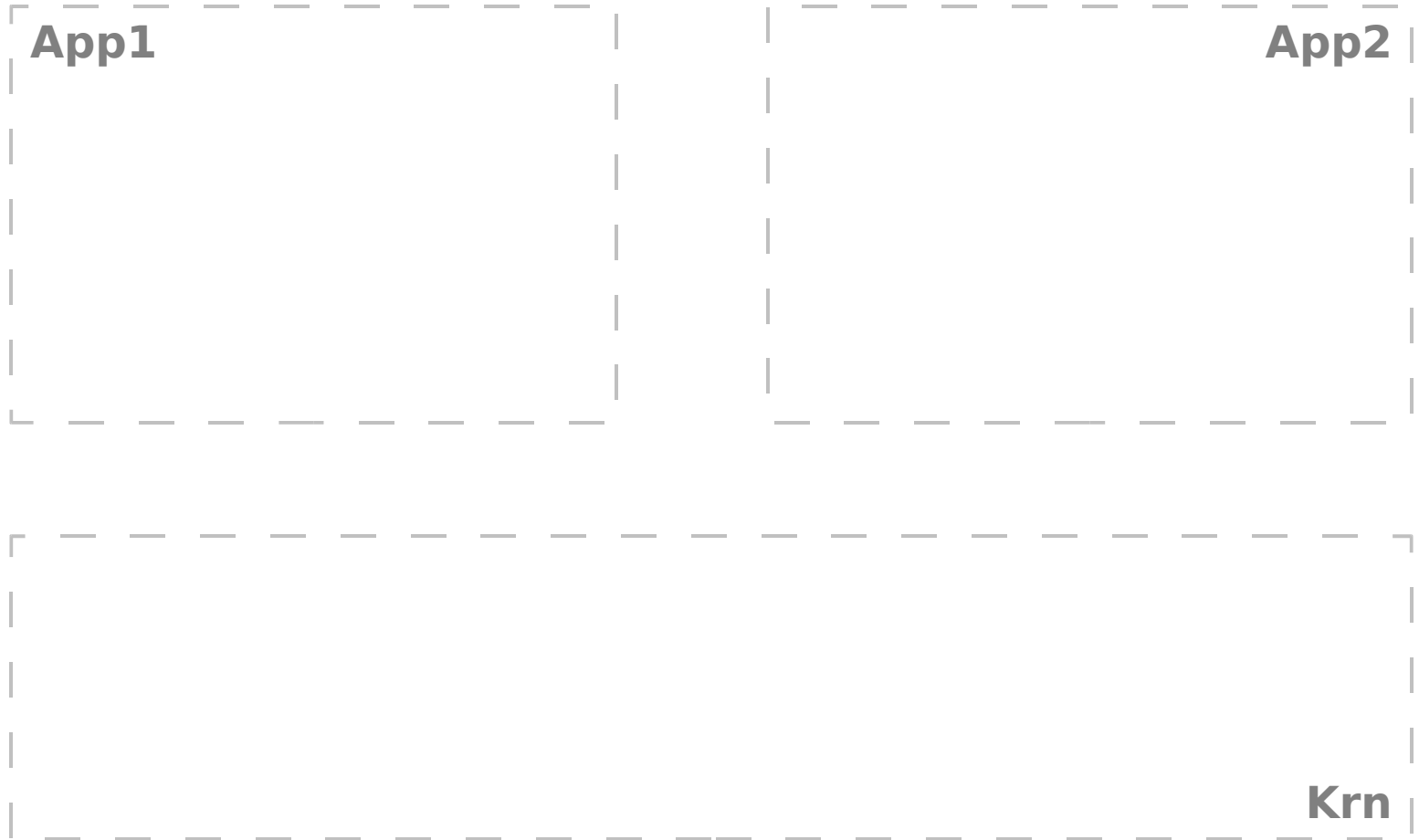


MP in Embedded Systems

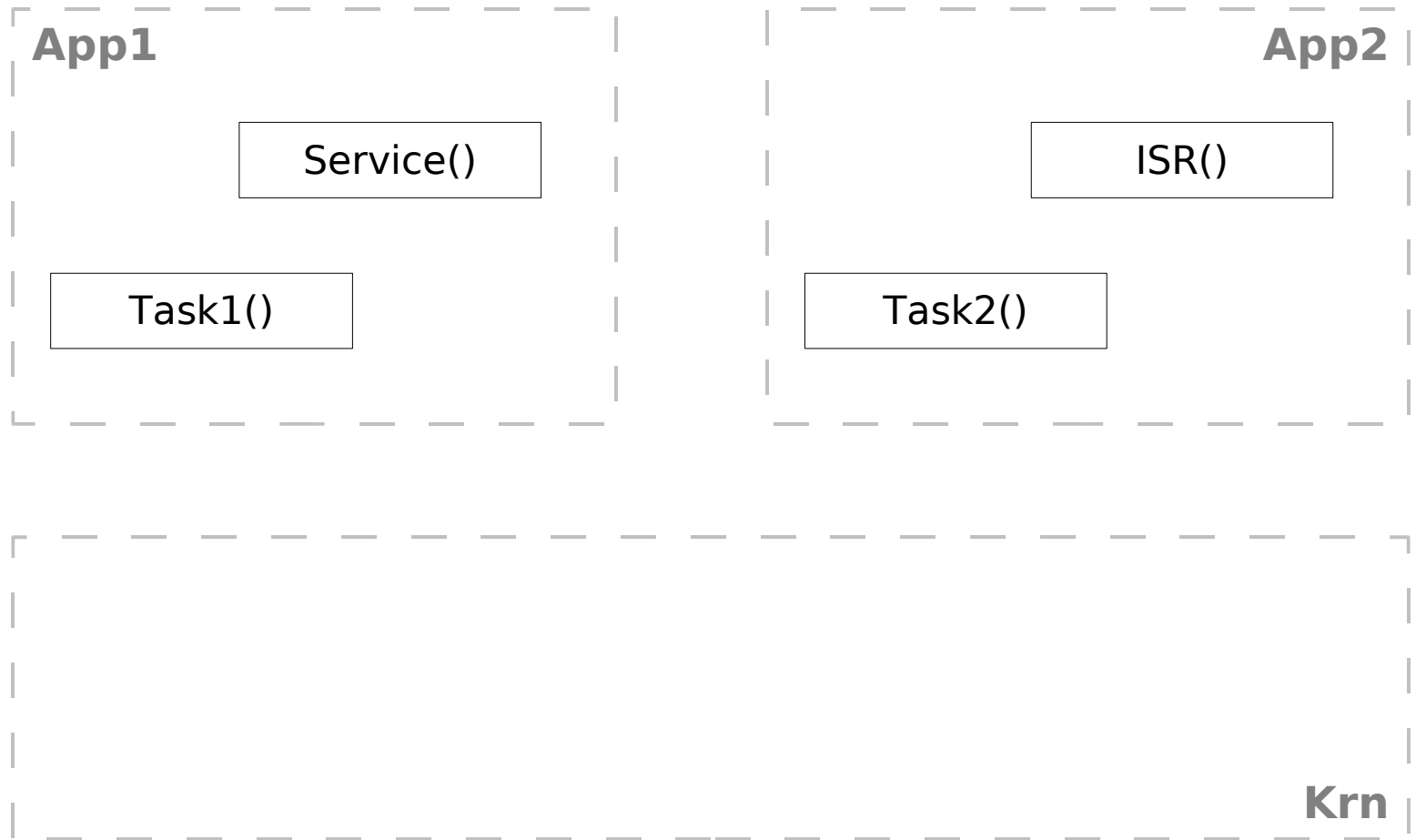
- Segmentation (MPU), not virtualization (MMU)
- Safety, not security
- Data protection, not code protection
- Configurable protection granularity:
 - No protection
 - Kernel protection
 - Application protection
 - Task protection



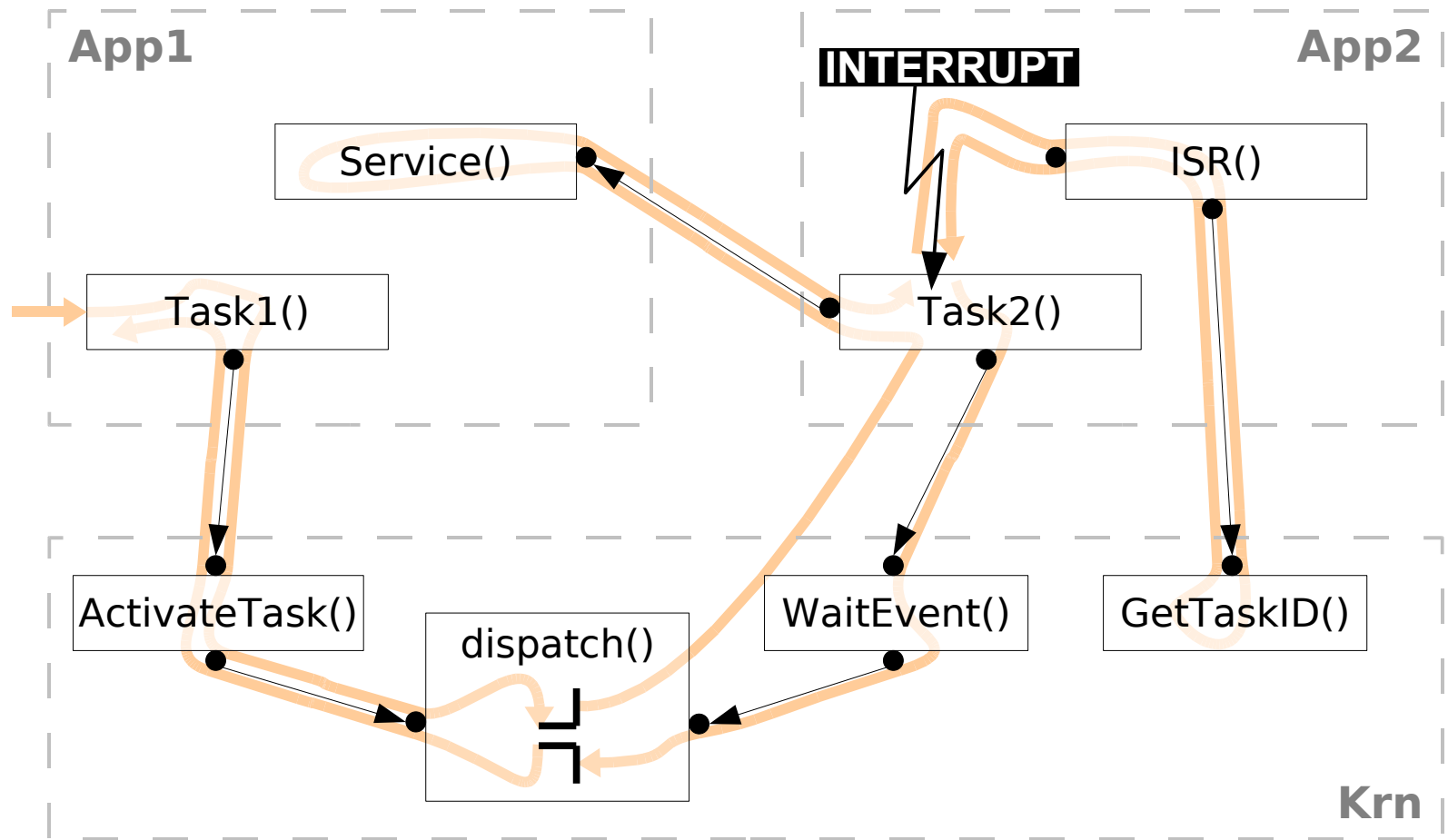
MP in CiAO – Concept



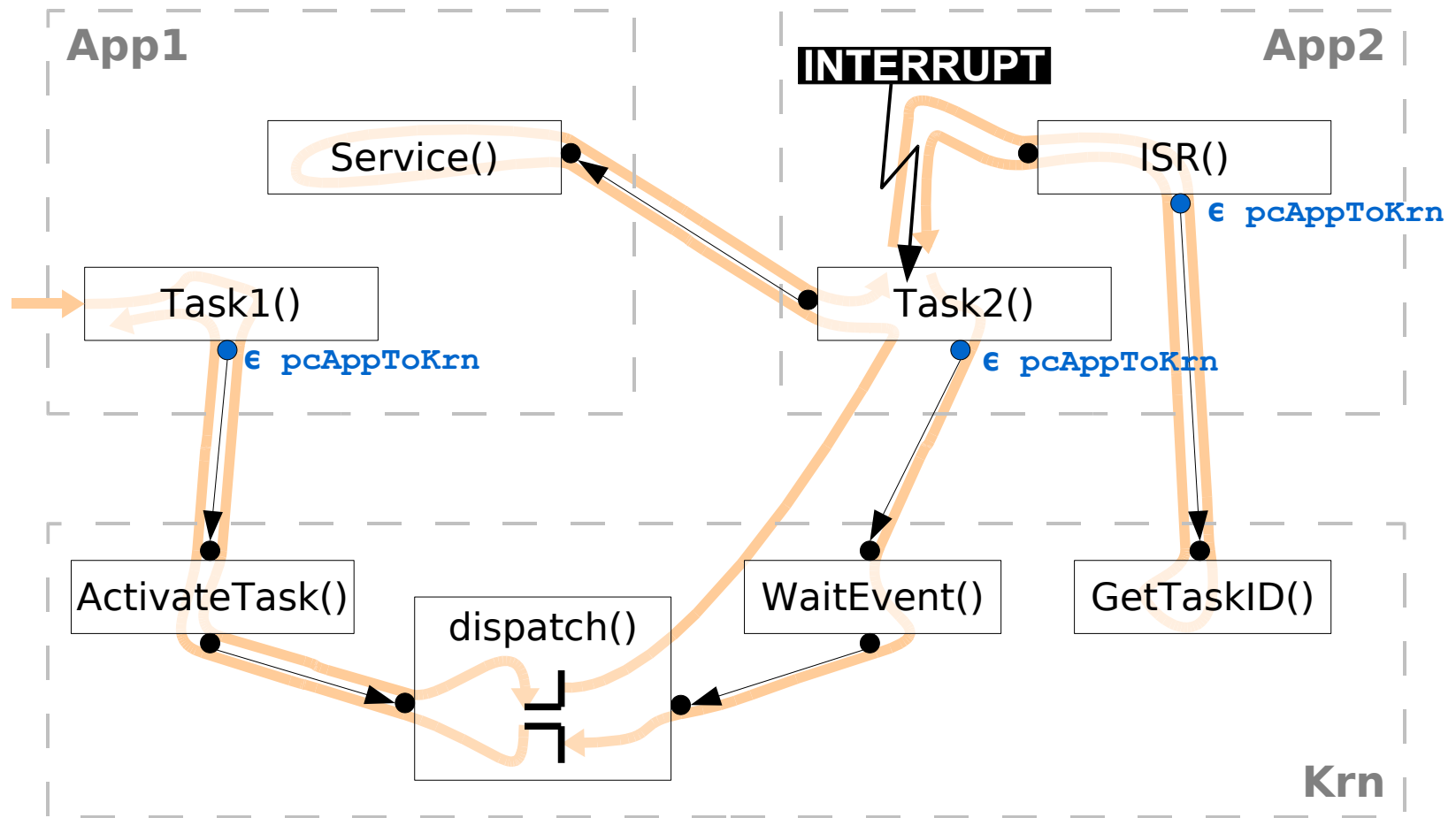
MP in CiAO – Concept



MP in CiAO – Concept

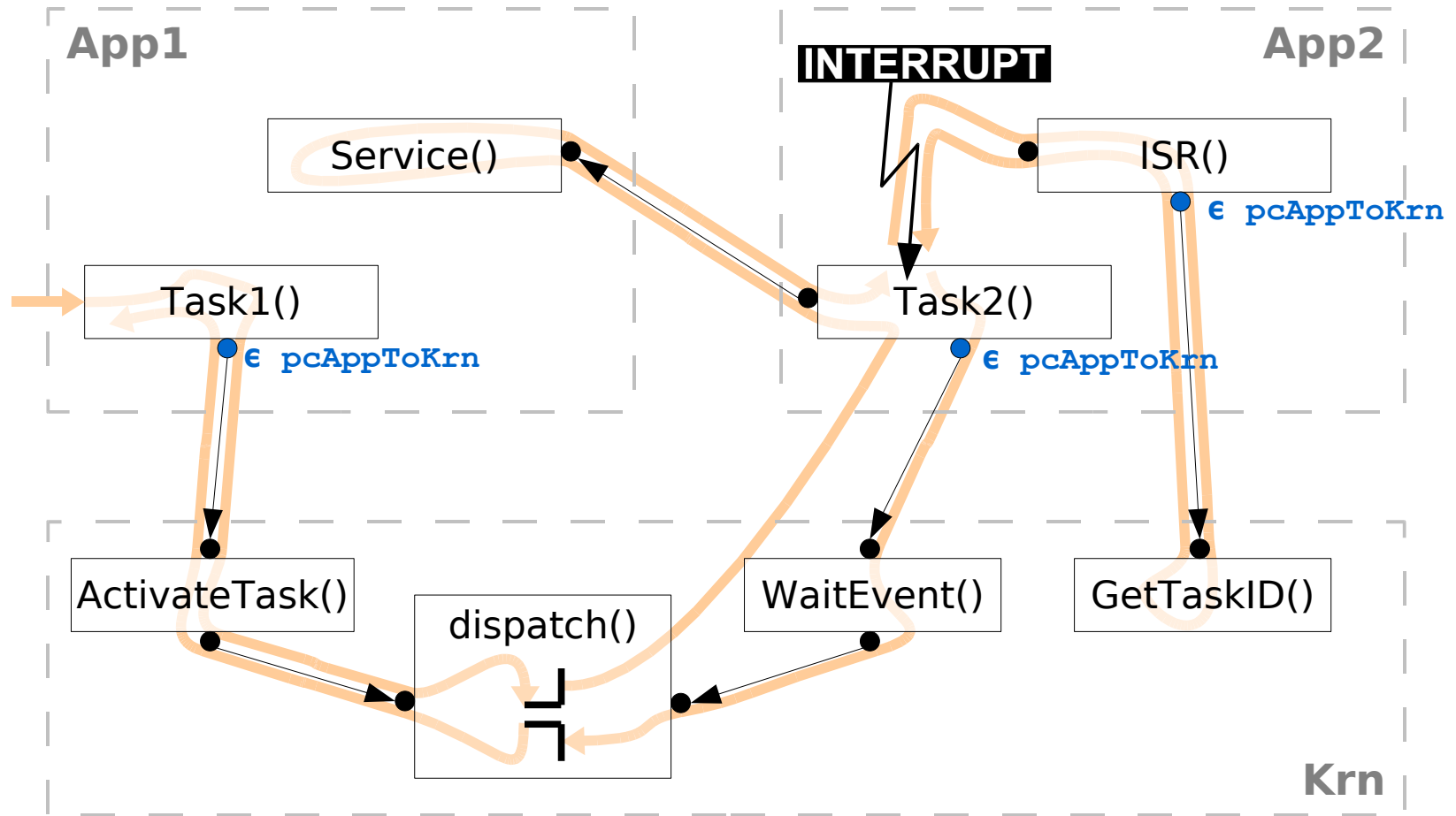


MP in CiAO – Concept



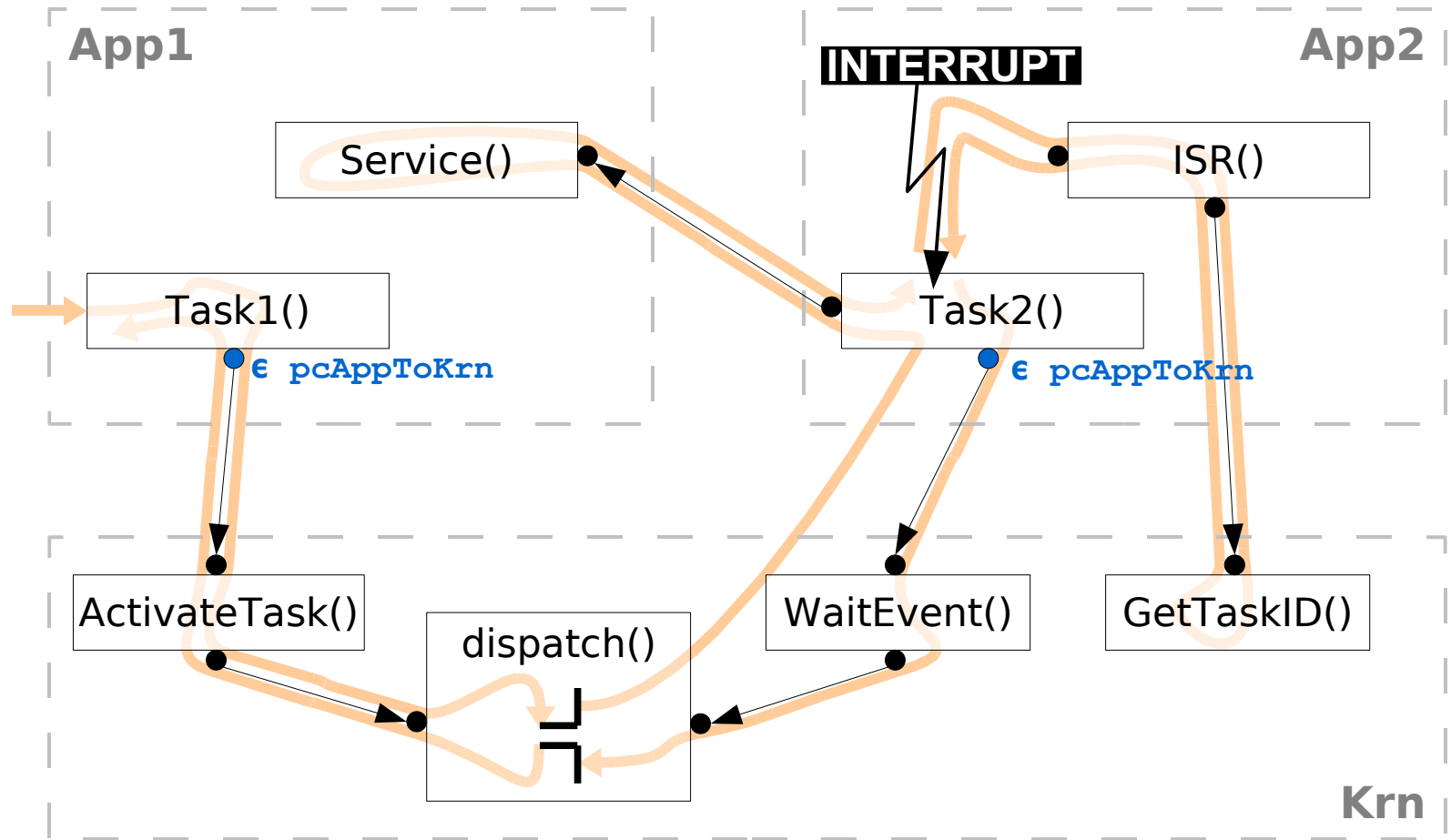
MP in CiAO – Concepts

```
pointcut pcAppToKrn () =  
    call (pcKrn ()) &&  
    ! within (pcKrn ());
```

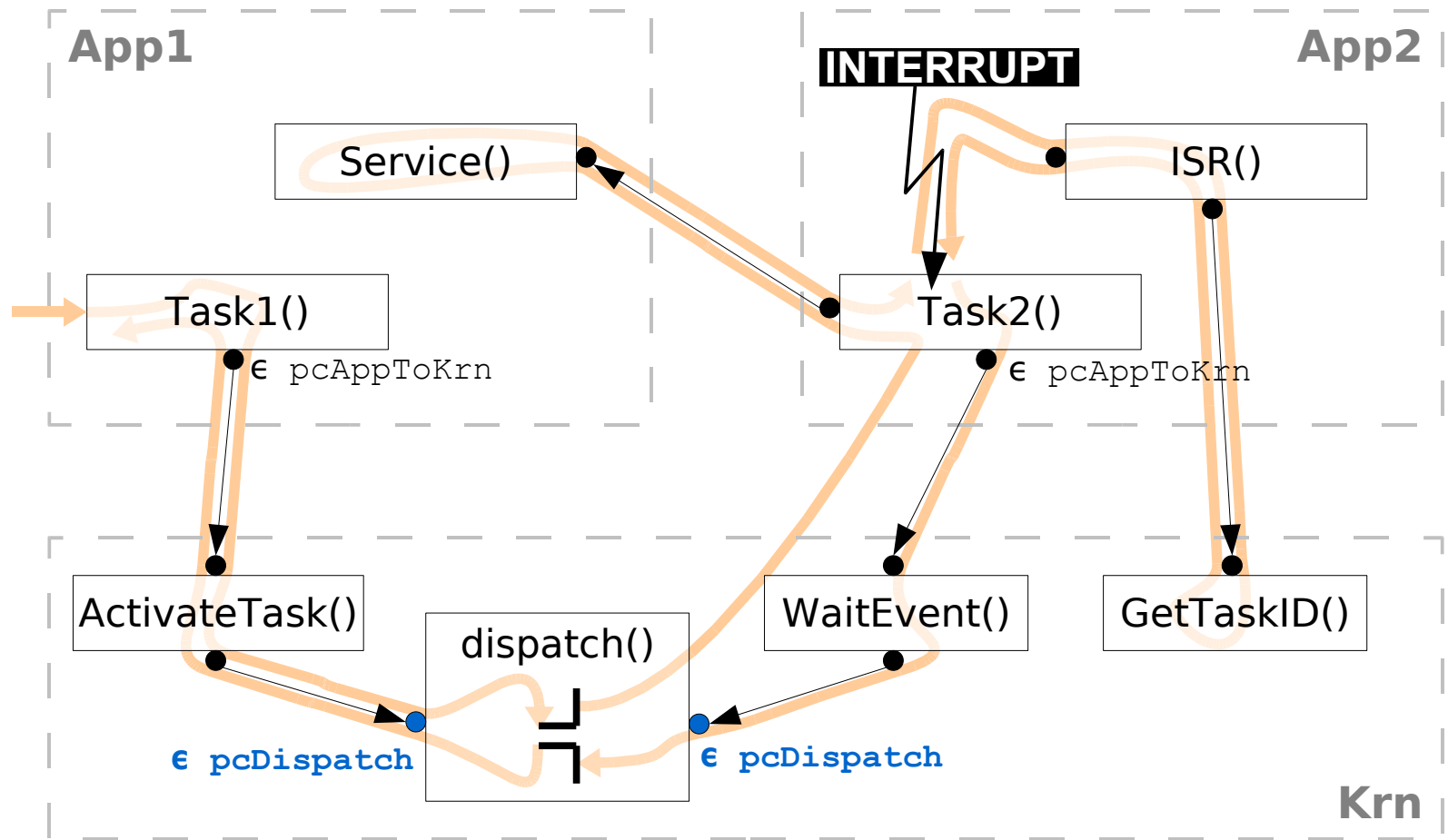


MP in CiAO – Concepts

```
pointcut pcAppToKrn () =  
    call (pcKrn ()) &&  
    ! within (pcKrn ()) &&  
    ! call (pcConst ());
```



MP in CiAO – Concept



MP in CiAO – Aspects

```
aspect KernelProtection {
  advice pcAppToKrn () : before () {
    enterKernel ();
  }
  advice pcAppToKrn () : after () {
    leaveKernel ();
  }
  advice pcDispatch () : after () {
    if (tjp->arg<1> ()->firstRun_) {
      leaveKernel ();
    }
  }
};
```

KernelProtection.ah



MP in CiAO – Aspects

```
aspect KernelProtection {  
  aspect ApplicationProtection {  
    advice pcAppToApp () : before () {  
      enterKernel ();  
      switchApplication (JoinPoint::Target::AppId);  
      leaveKernel ();  
    }  
    advice pcAppToApp () : after () {  
      enterKernel ();  
      switchApplication (JoinPoint::That::AppId);  
      leaveKernel ();  
    }  
    advice pcDispatch () : before () {  
      if (tjp->arg<0> ()->owningApp_ !=  
          tjp->arg<1> ()->owningApp_) {  
        switchApplication (tjp->arg<1> ()->owningApp_);  
      }  
    }  
  }  
};
```

ApplicationProtection.ah



MP in CiAO – Semi-Trusted Mode

- Applications executed in supervisor mode
- Memory protection still enabled!
- No kernel traps needed
 - Reduced costs
 - Function-level linking
 - Inlining



The CiAO Project (CiAO is Aspect-Oriented)

- Product line of embedded operating systems
 - Statically configurable
- Major goal:

Configurability of
architectural properties
by **aspect-oriented** techniques



The CiAO Project (CiAO is Aspect-Oriented)

- Product line of embedded operating systems
 - Statically configurable
- Major goal:

Configurability of
architectural properties
by **aspect-oriented** techniques

- Architectural properties:
 - **Memory protection**
 - Interrupt synchronization
 - Interaction
 - Timing protection

