Embedded Control System Development with

Giotto

www.eecs.berkeley.edu/~fresco/giotto

Thomas A. Henzinger, Ben Horowitz, Christoph Meyer Kirsch

UC Berkeley

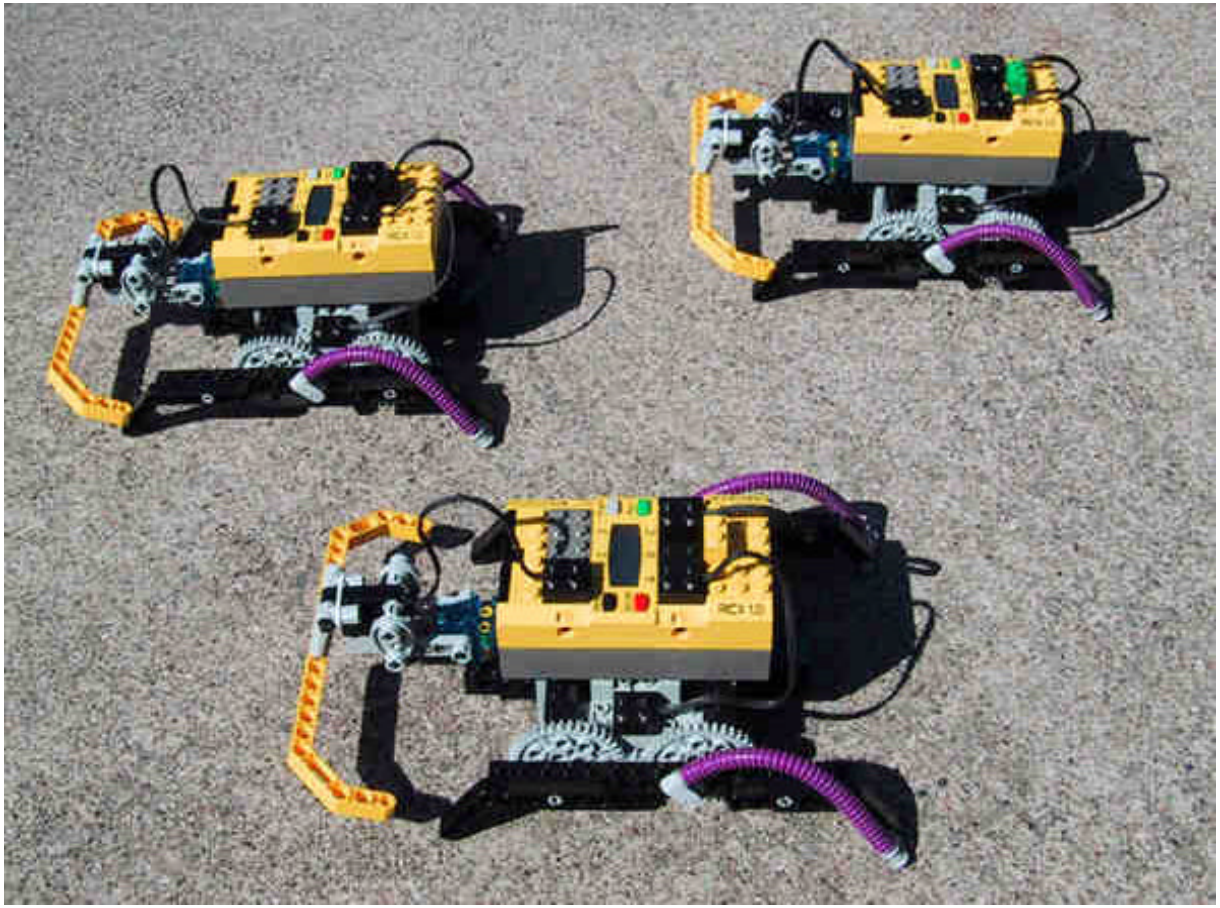
Giotto

- Giotto is a time-triggered and platform independent programming language + compiler + runtime library
- Giotto aims at hard real-time control applications on distributed platforms (safety-critical applications)

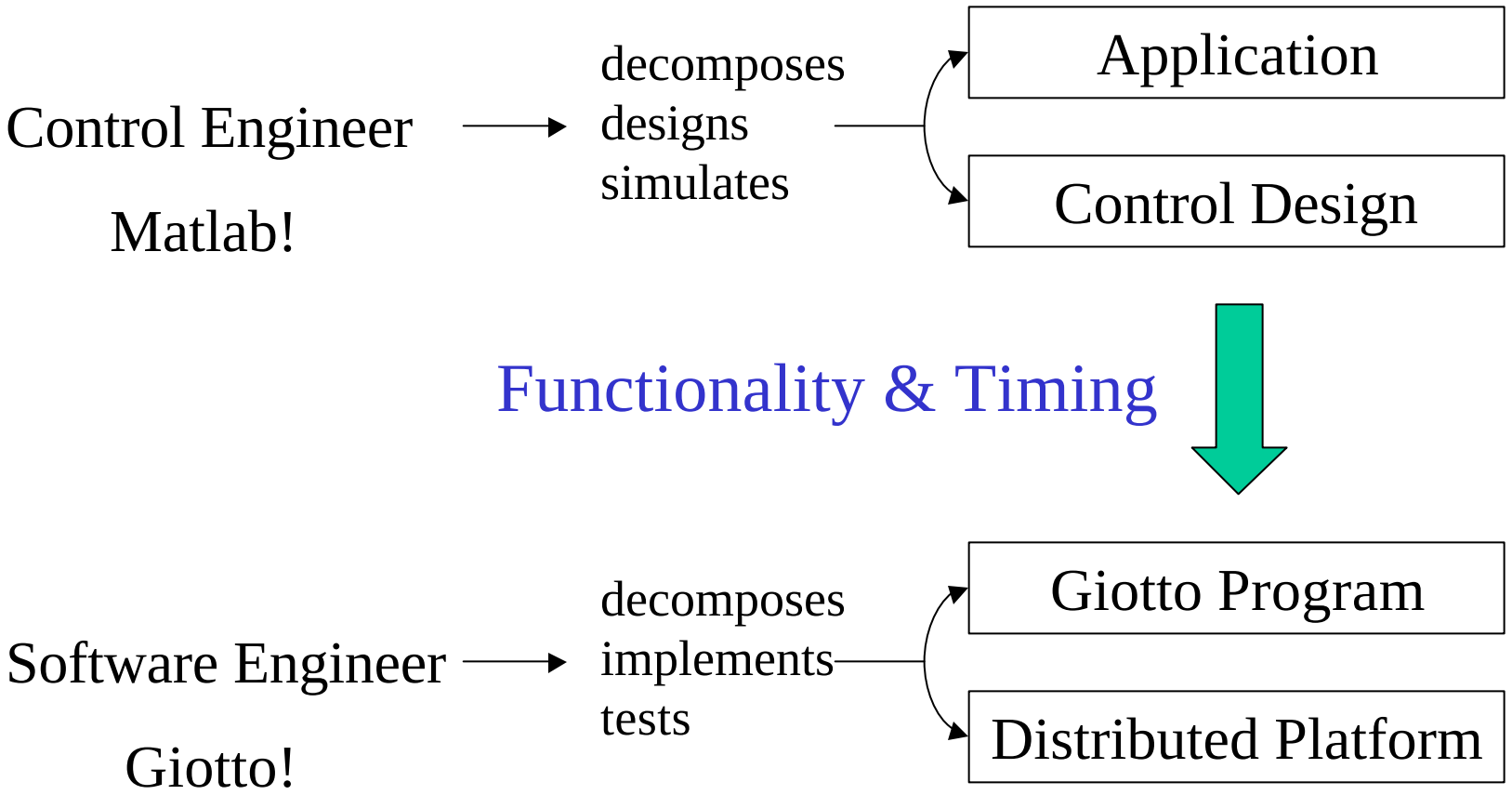
- Giotto provides an abstract programming model for embedded control system development

- Giotto runtime library for VxWorks, model of computation in Ptolemy II (Edward Lee, UC Berkeley)

Sneak Preview



Embedded System Development



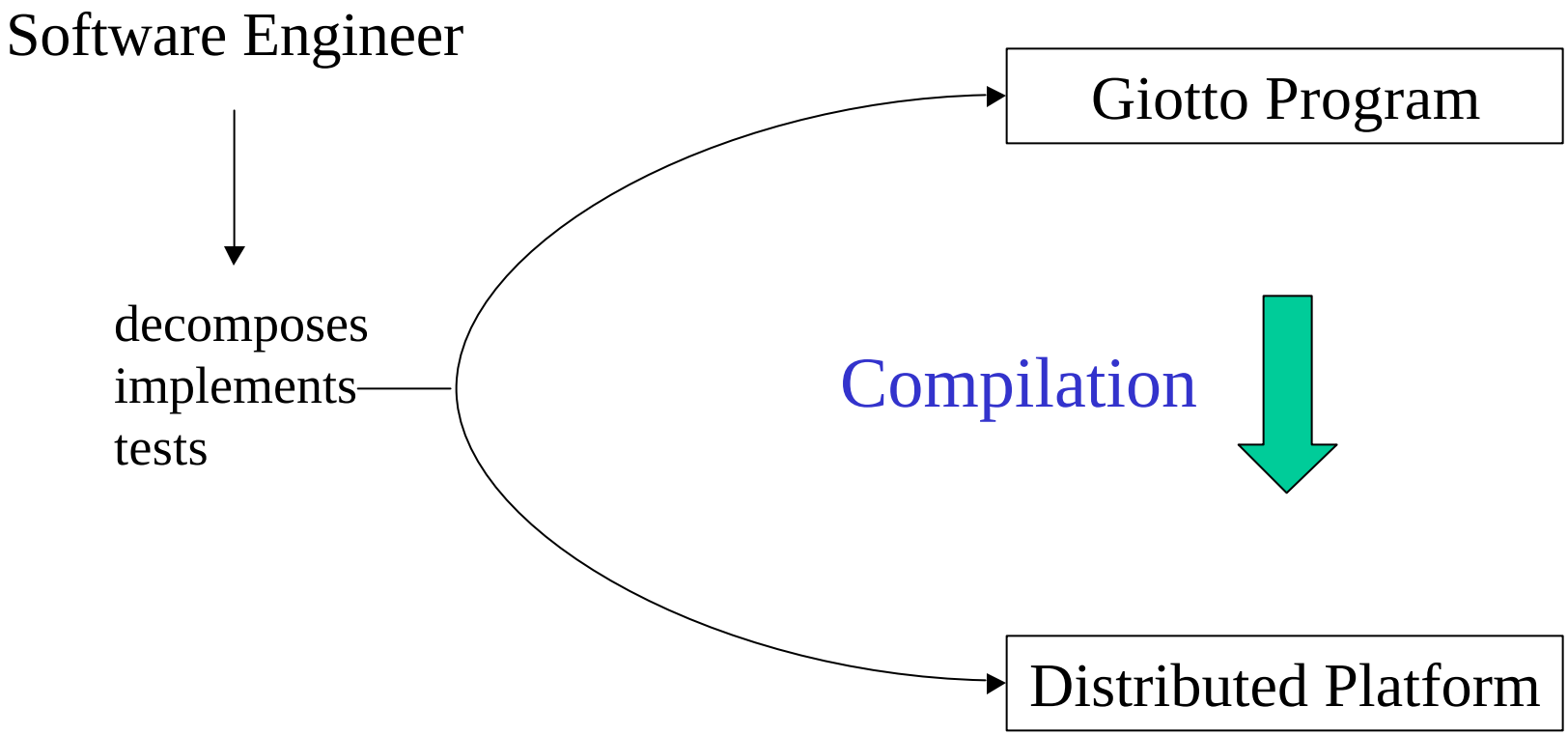
Decomposition: Giotto Modes

Control Engineer → decomposes → Control Design

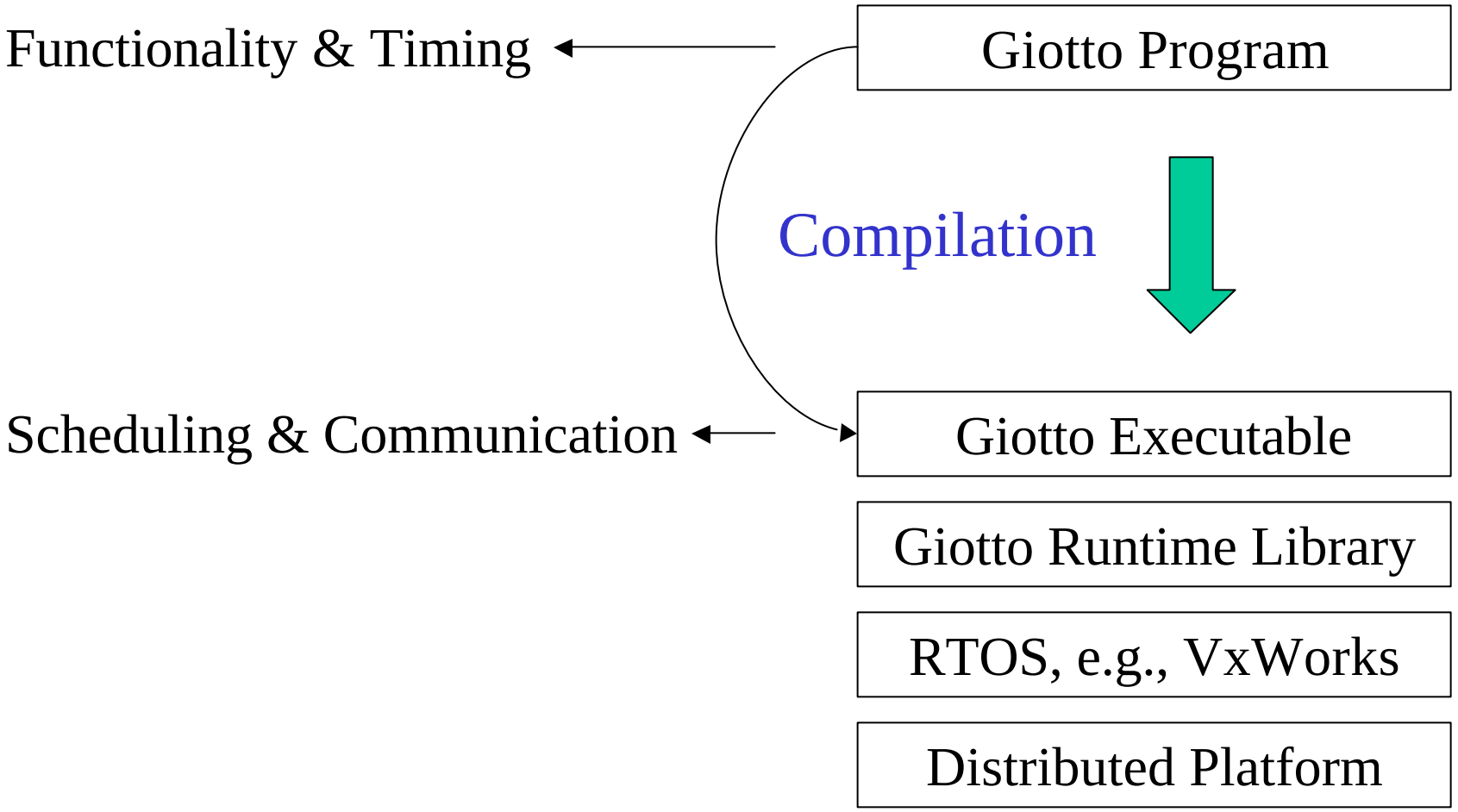
Preserving Structure ↓

Software Engineer → decomposes → Giotto Program

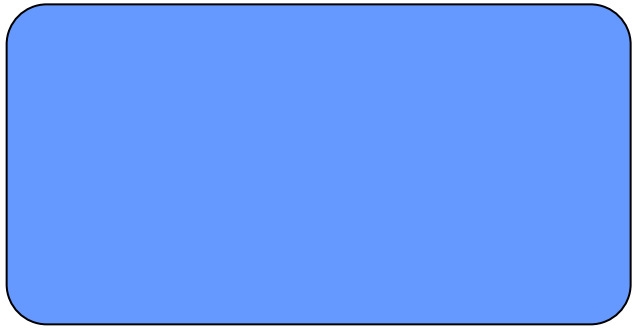
The Giotto Compiler



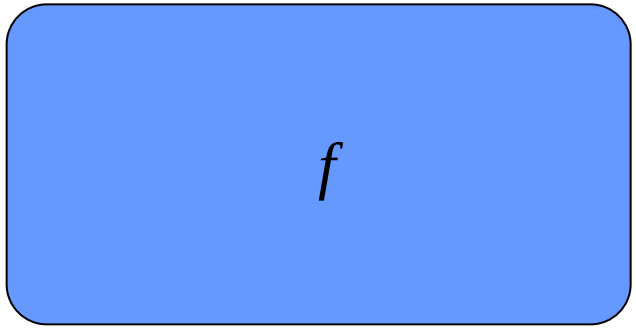
The Giotto Compiler



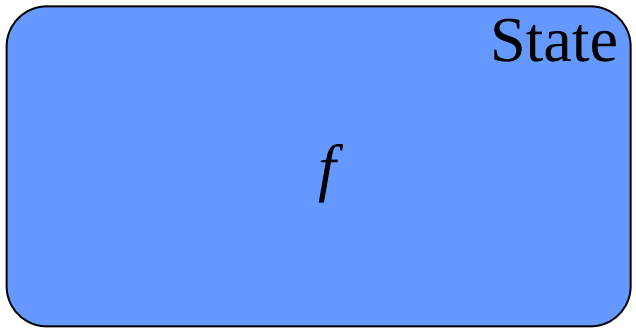
A Task



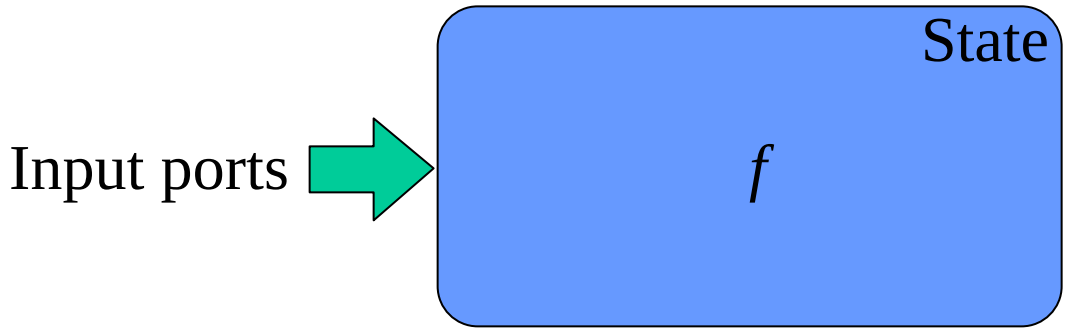
Abstract Syntax of a Task



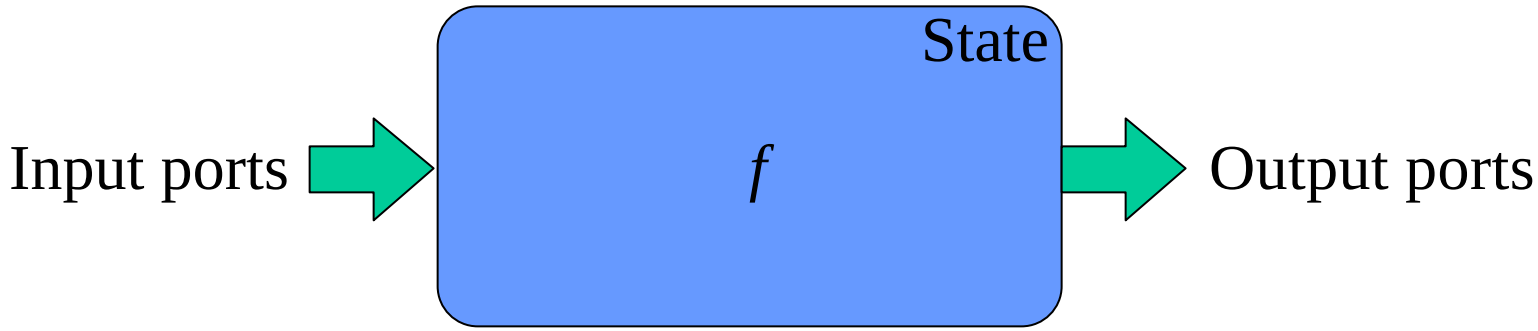
Abstract Syntax of a Task



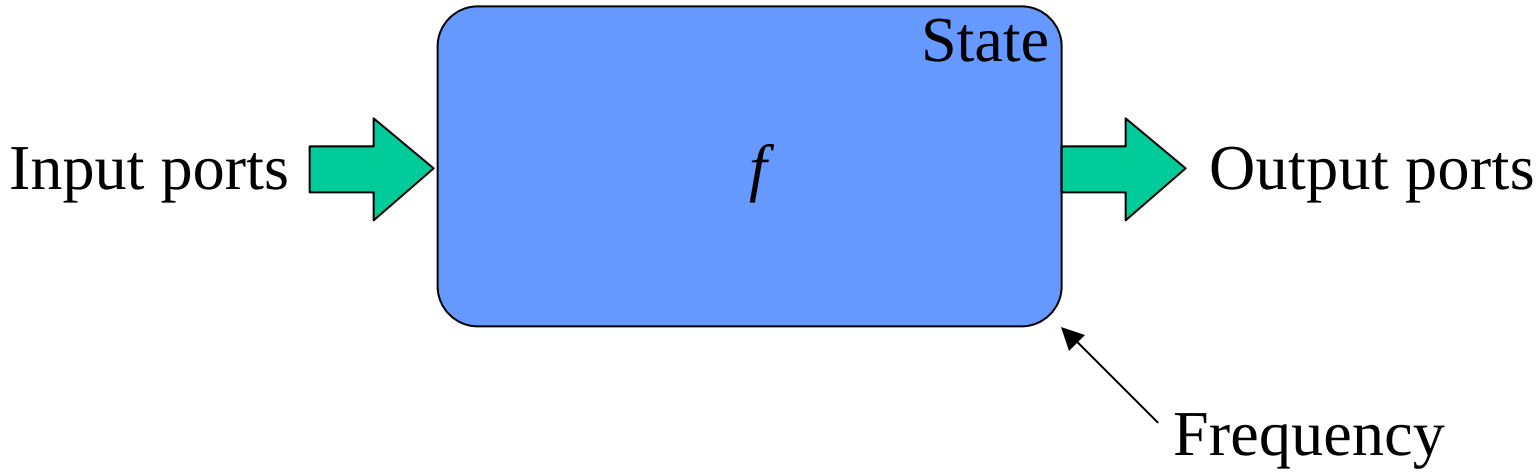
Abstract Syntax of a Task



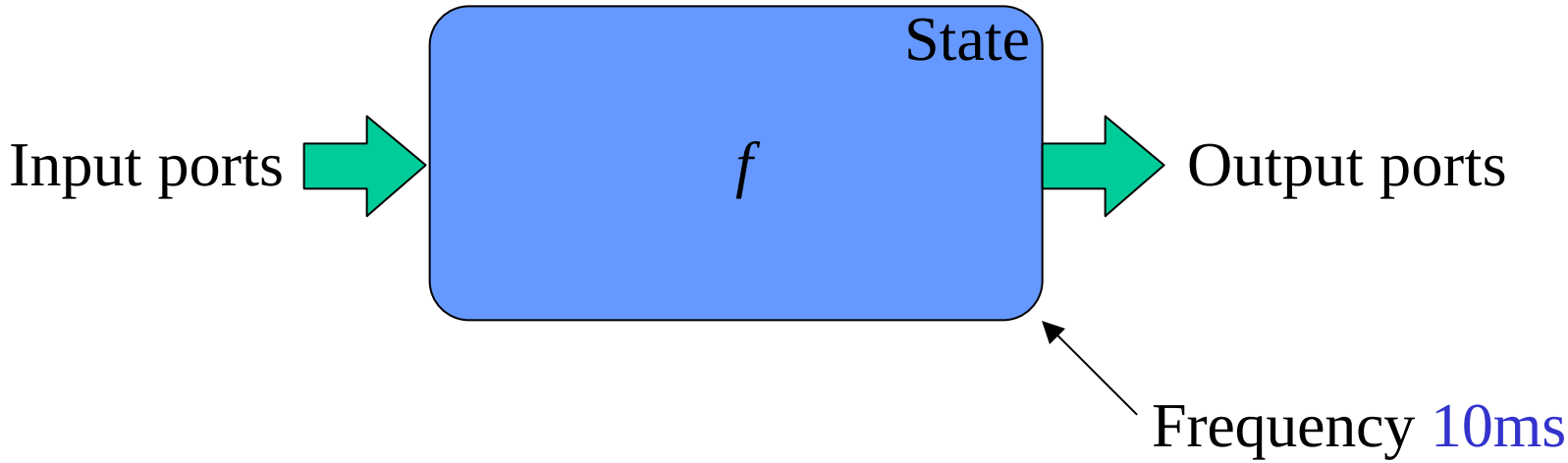
Abstract Syntax of a Task



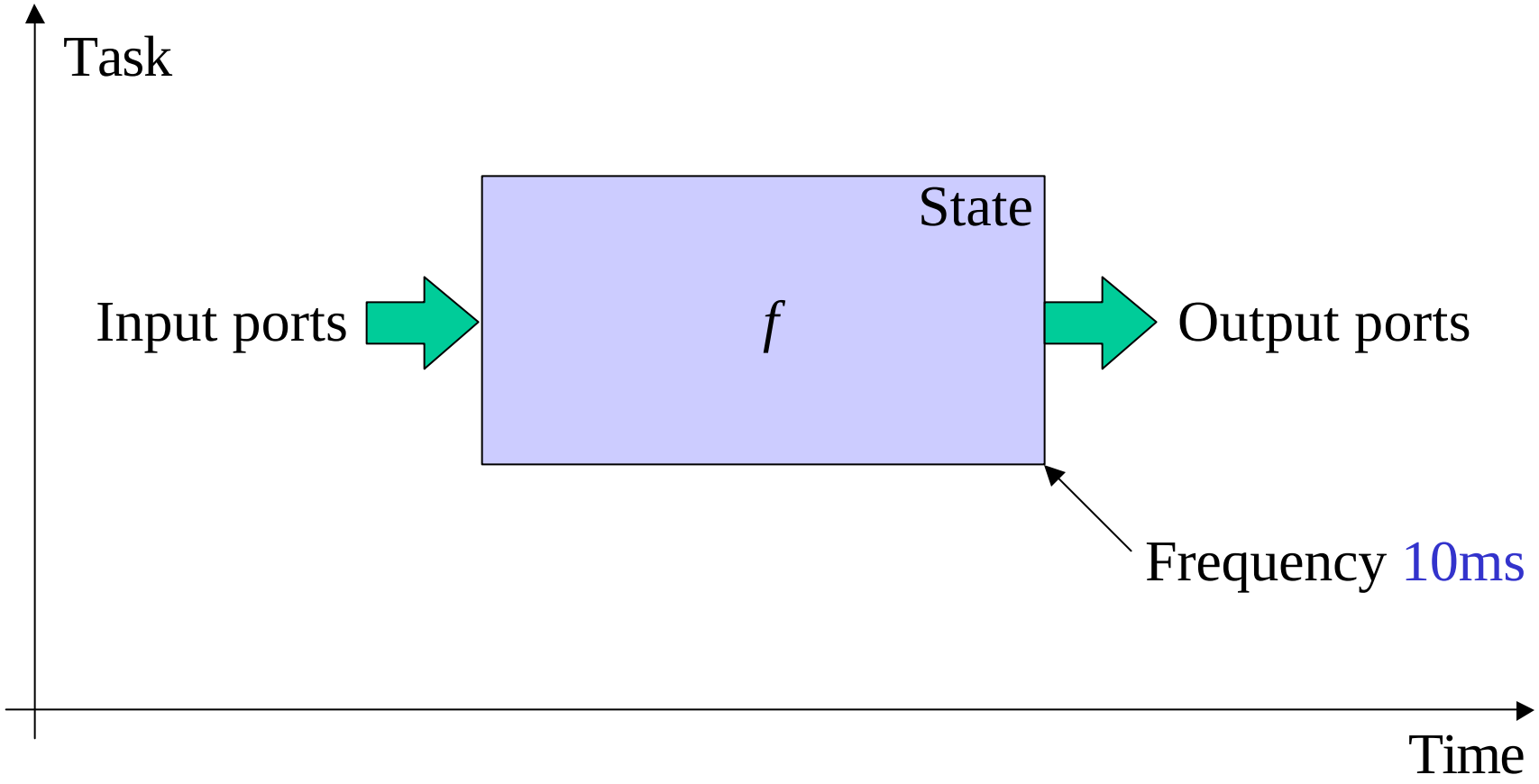
Abstract Syntax of a Task



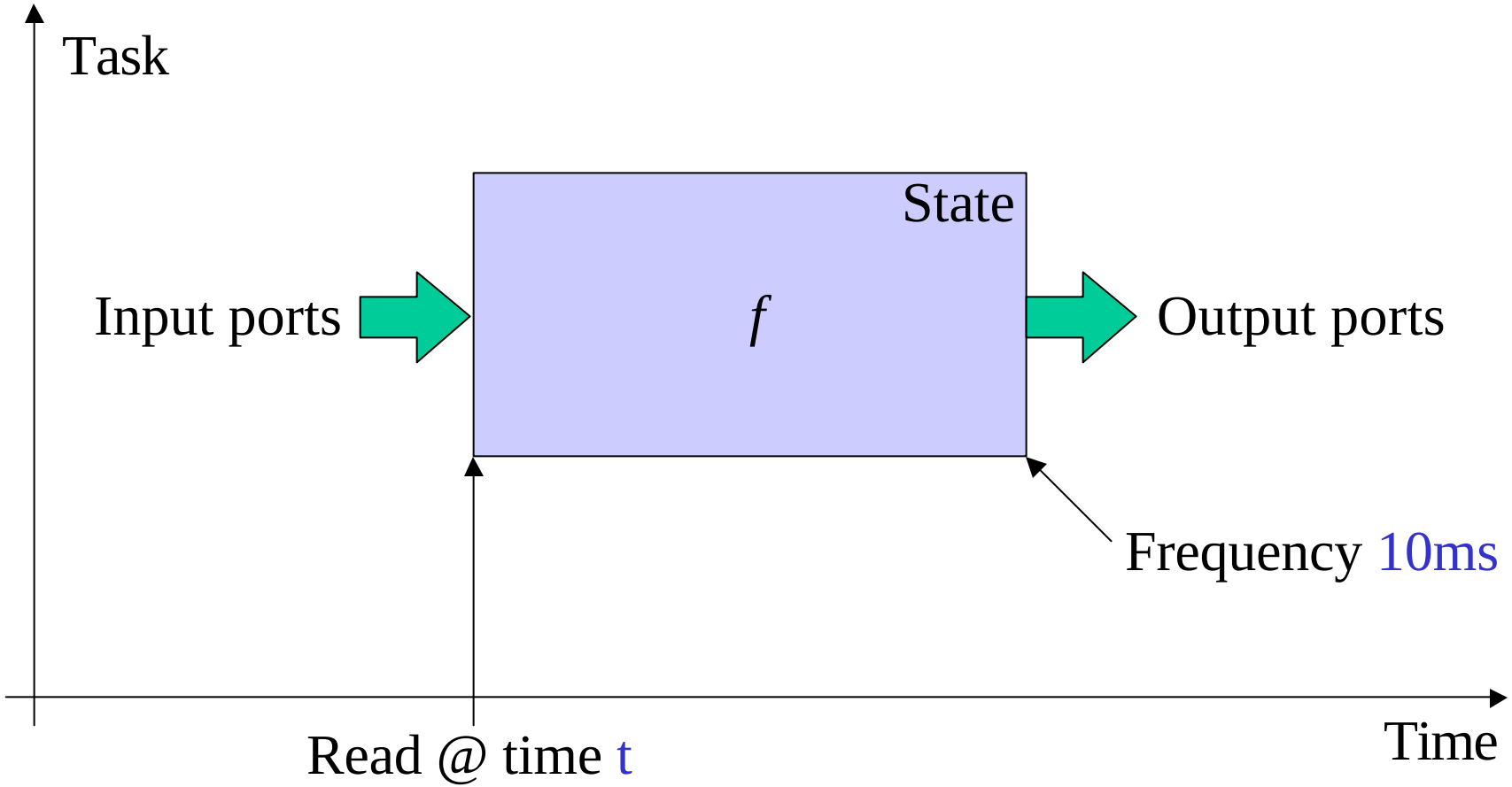
Semantics of a Task



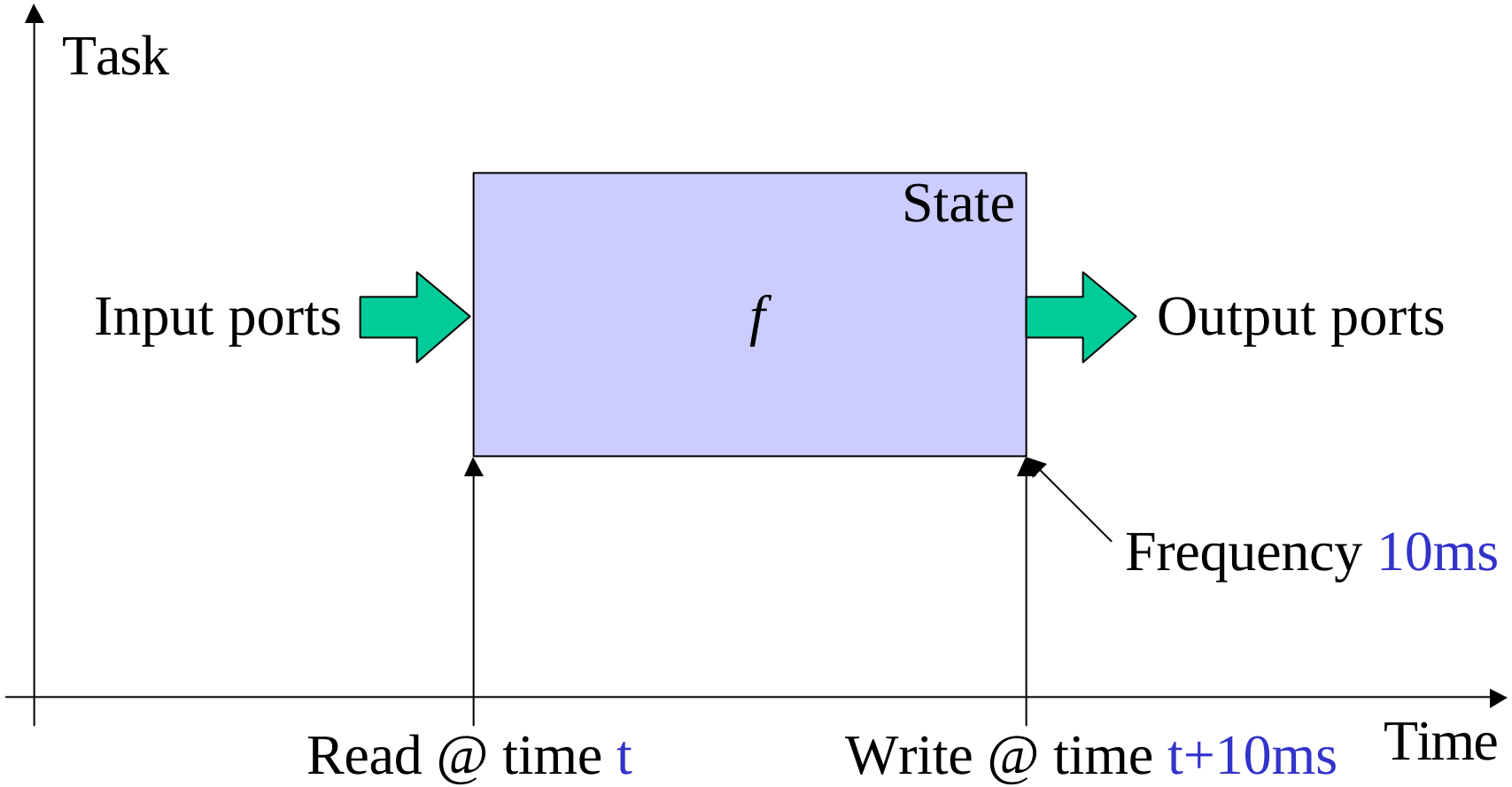
Semantics of a Task



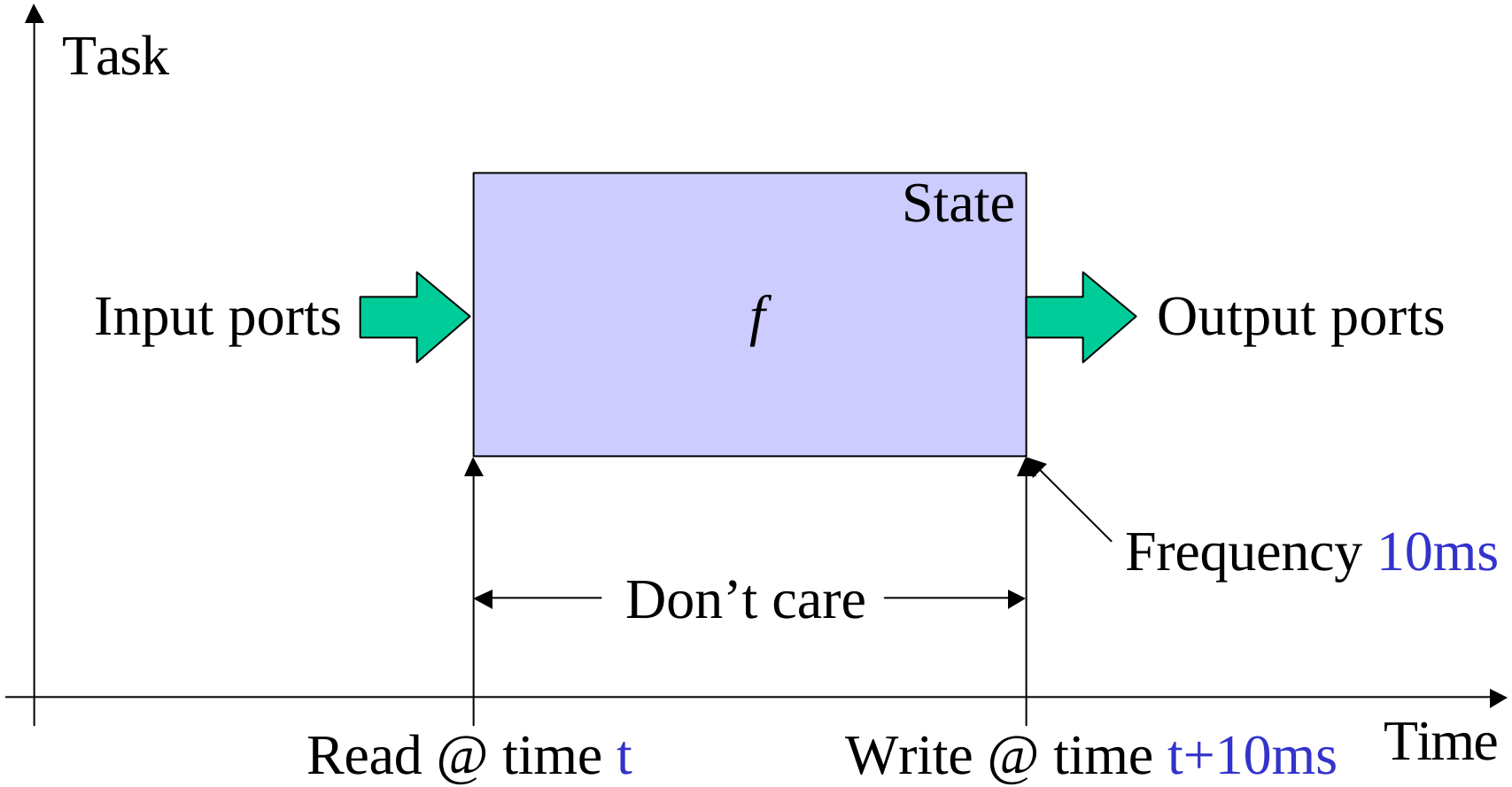
Semantics of a Task



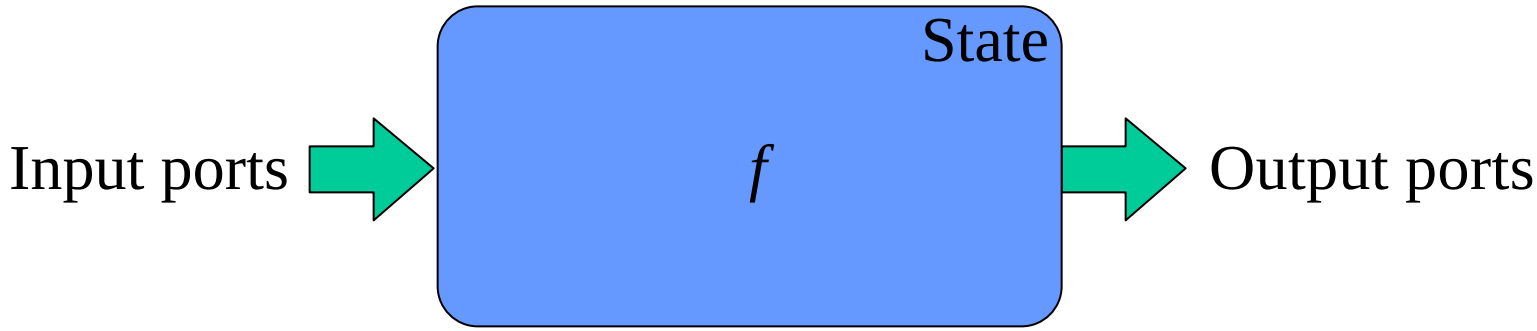
Semantics of a Task



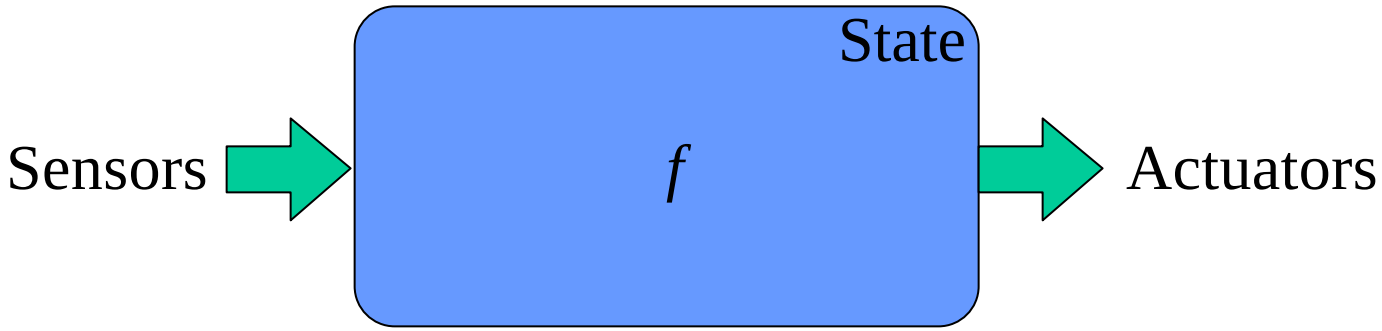
Semantics of a Task



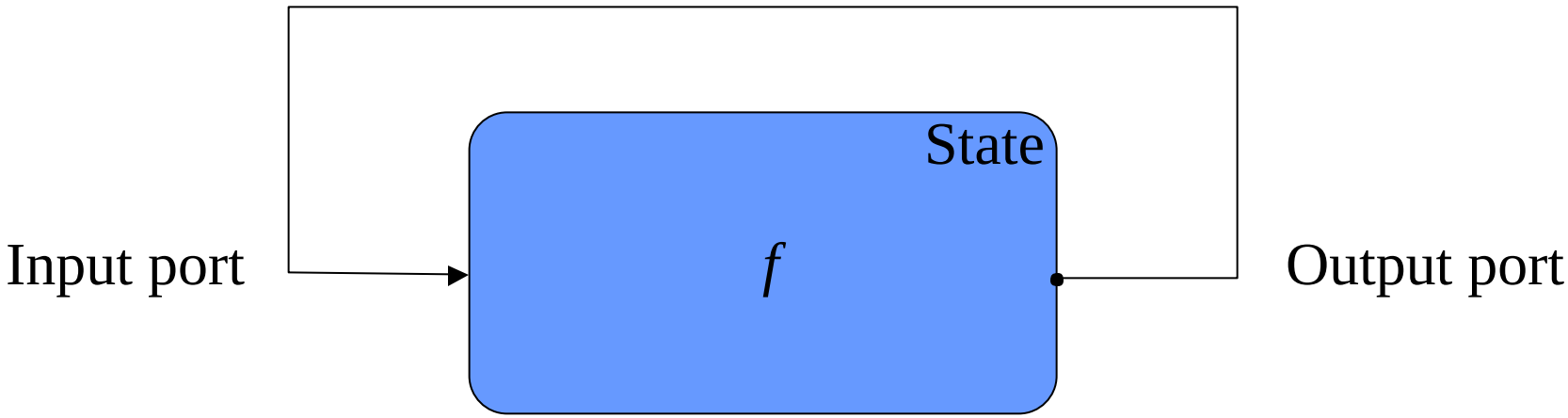
Data Flow



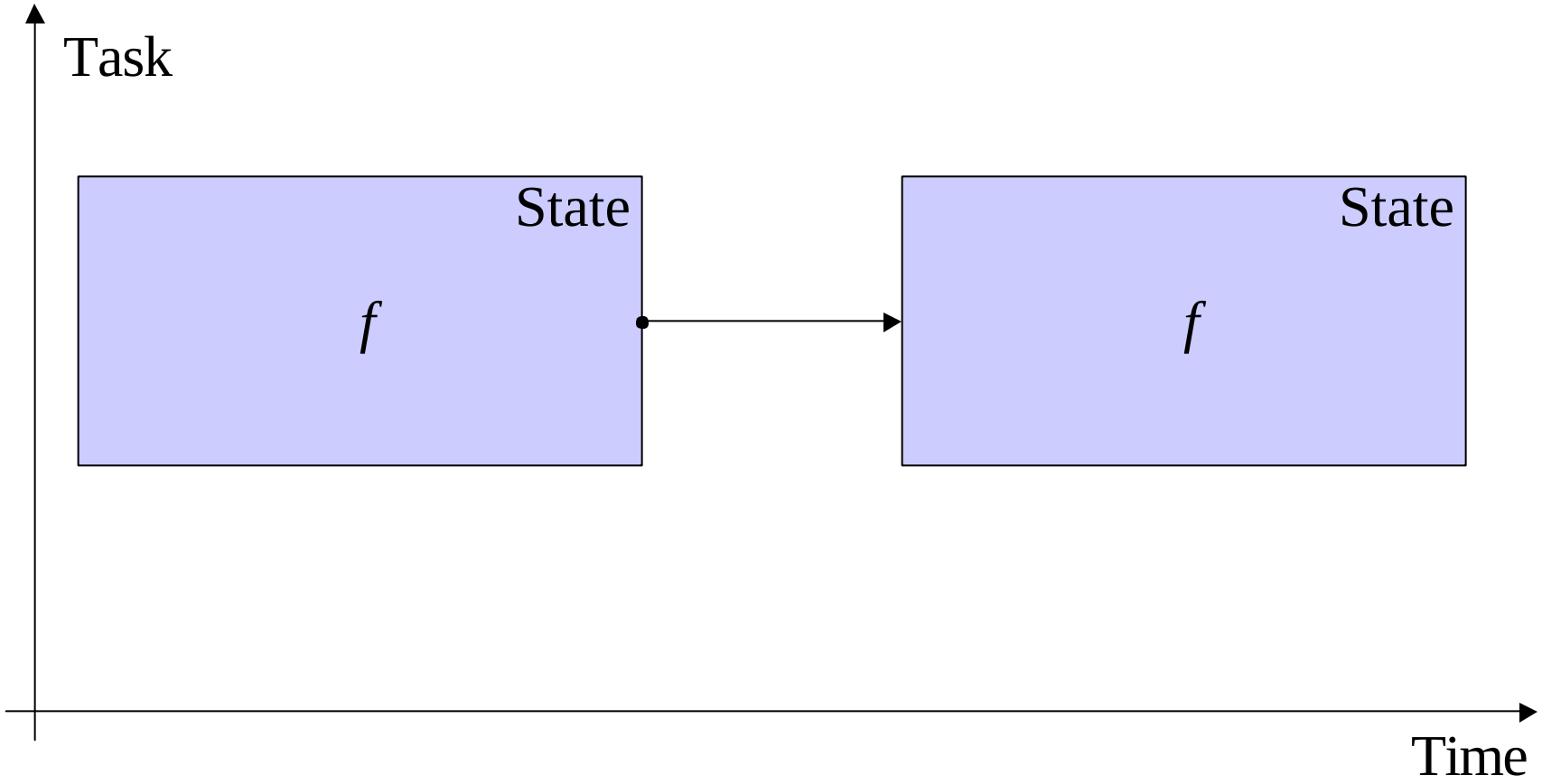
Sensor - Control Law - Actuator



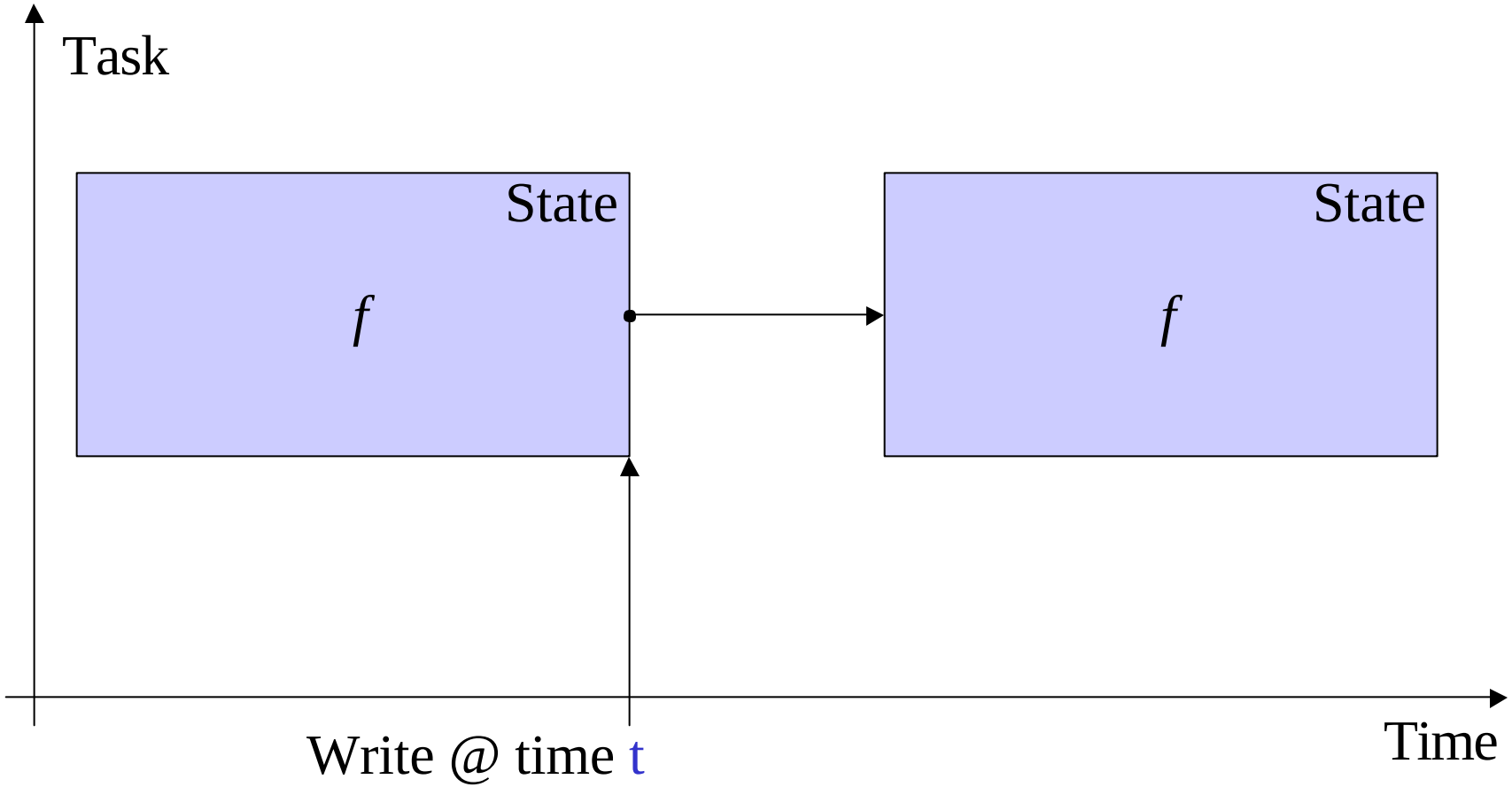
Intertask Communication



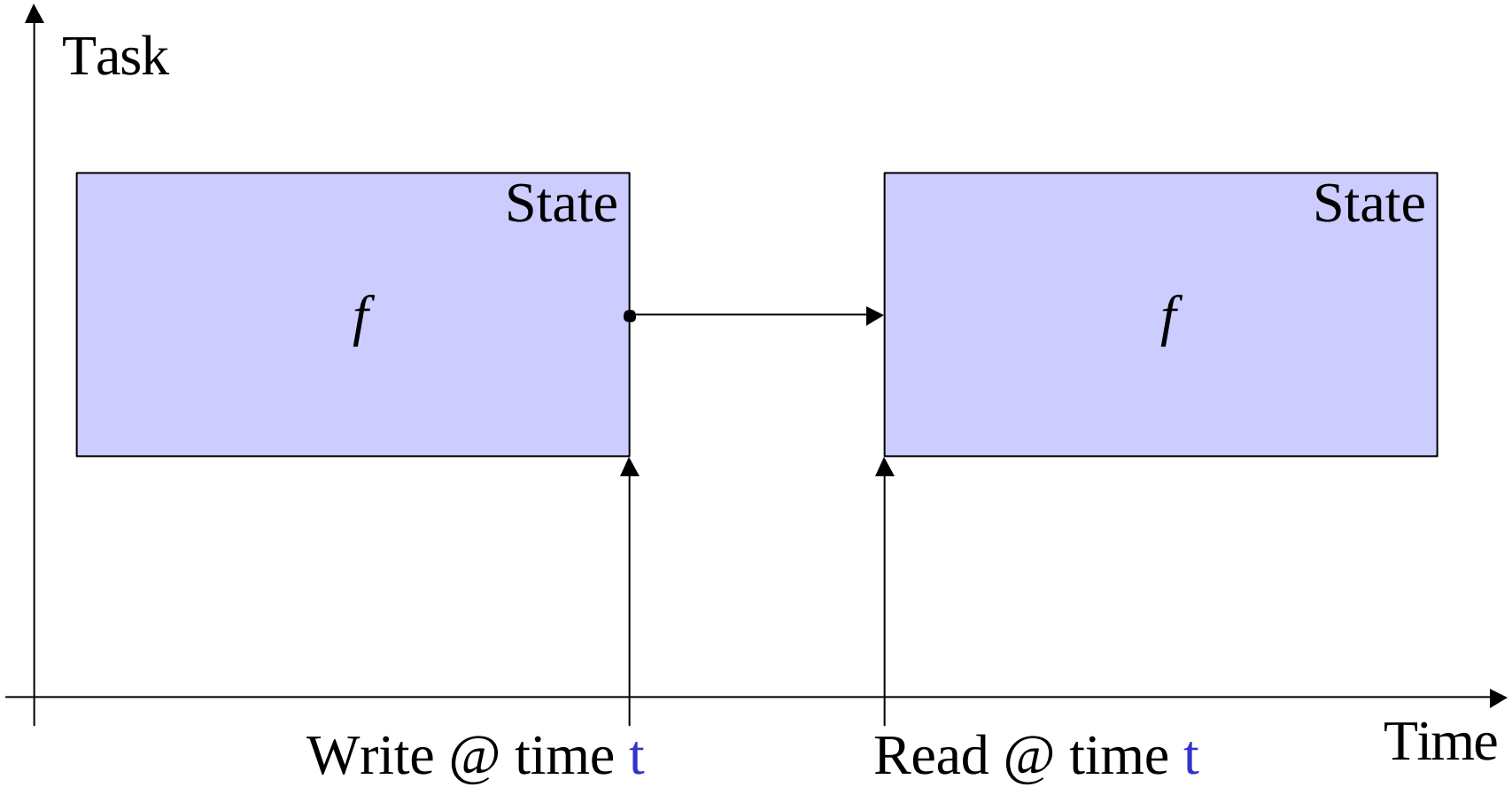
Data Flow Semantics



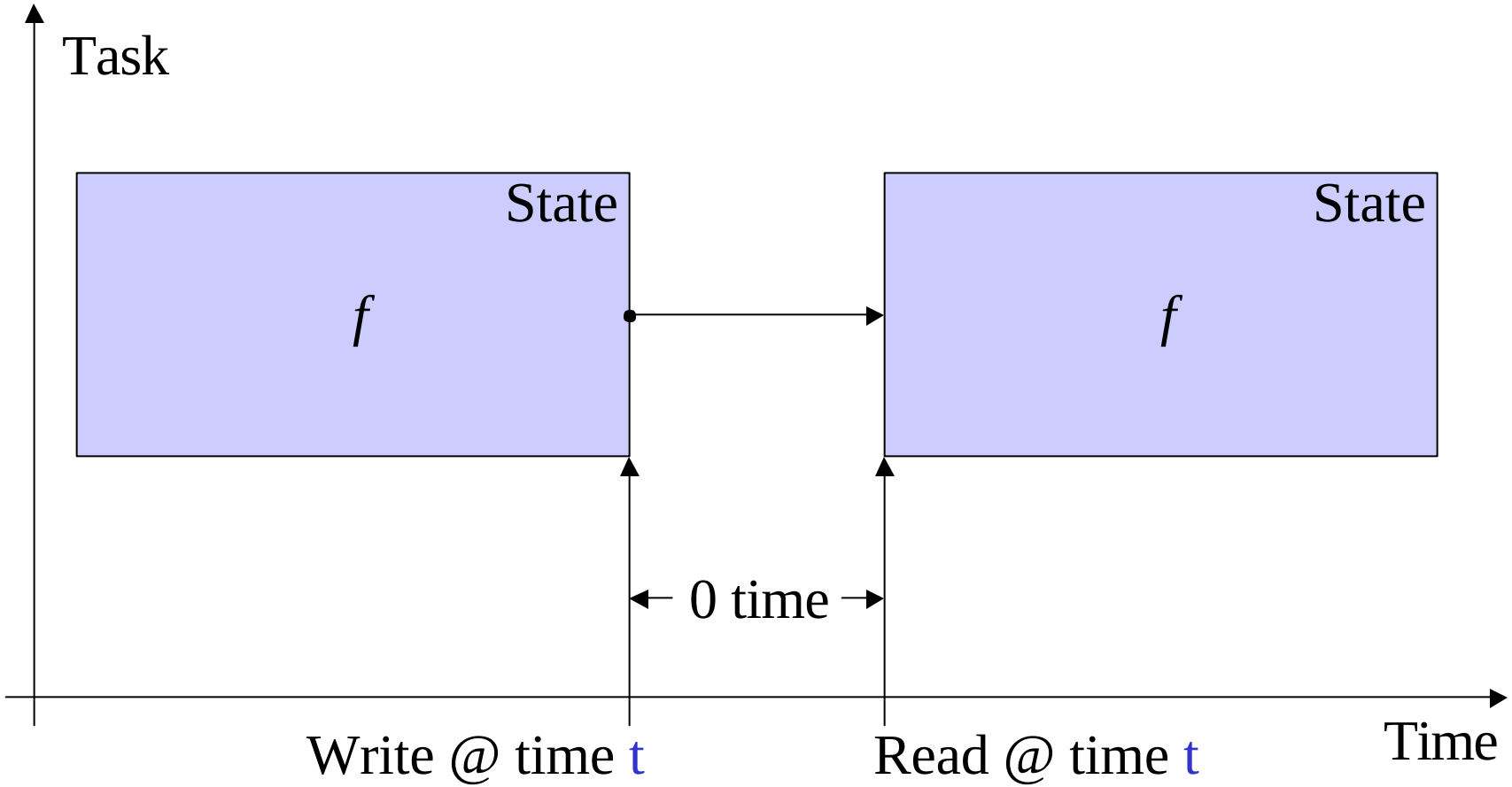
Data Flow Semantics



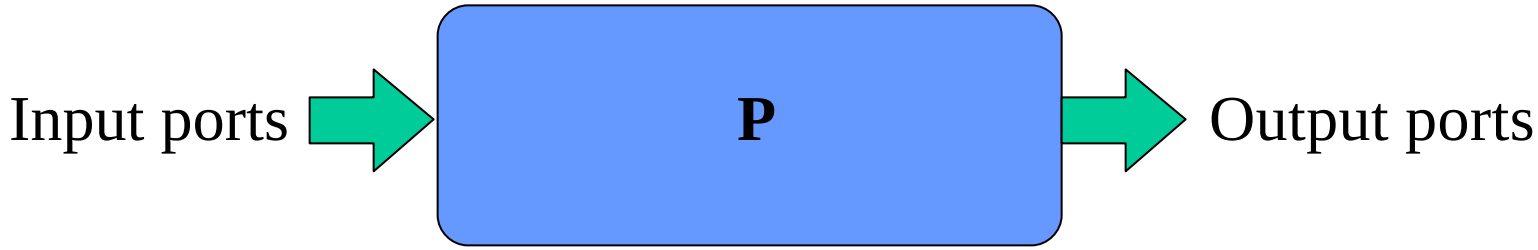
Data Flow Semantics



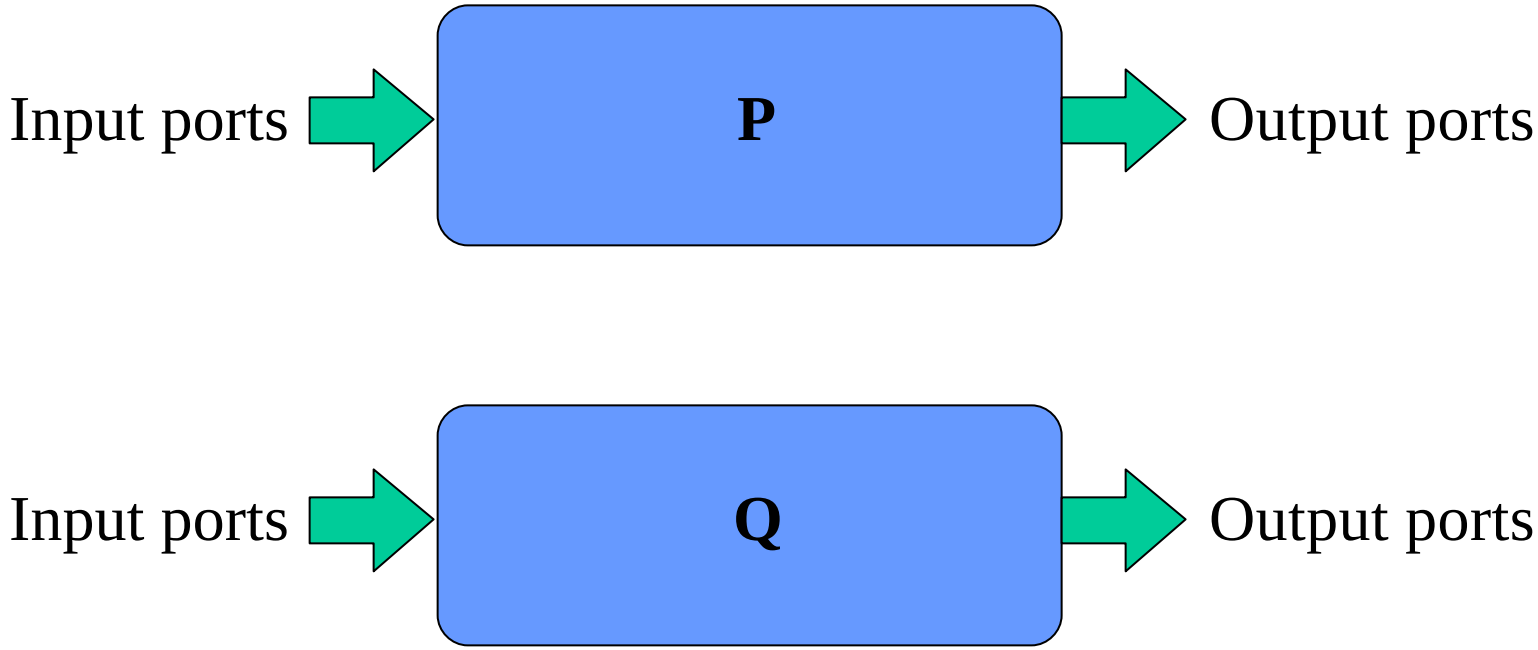
Data Flow Semantics



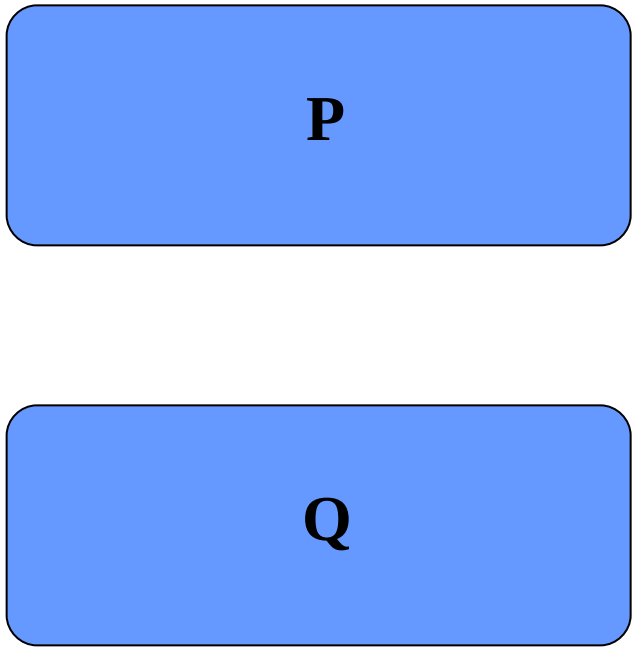
Two Tasks



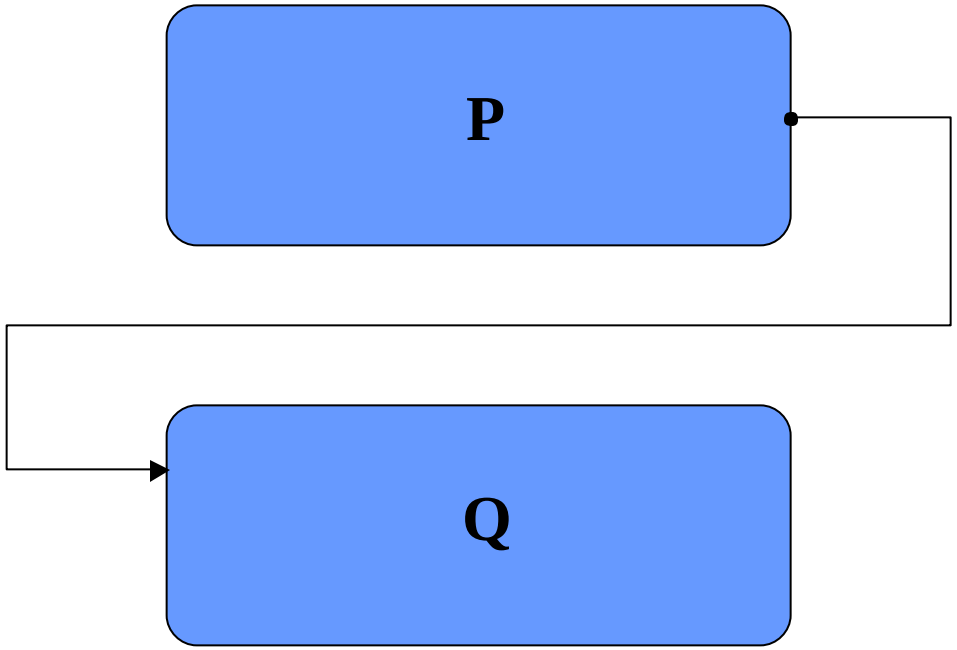
Two Tasks



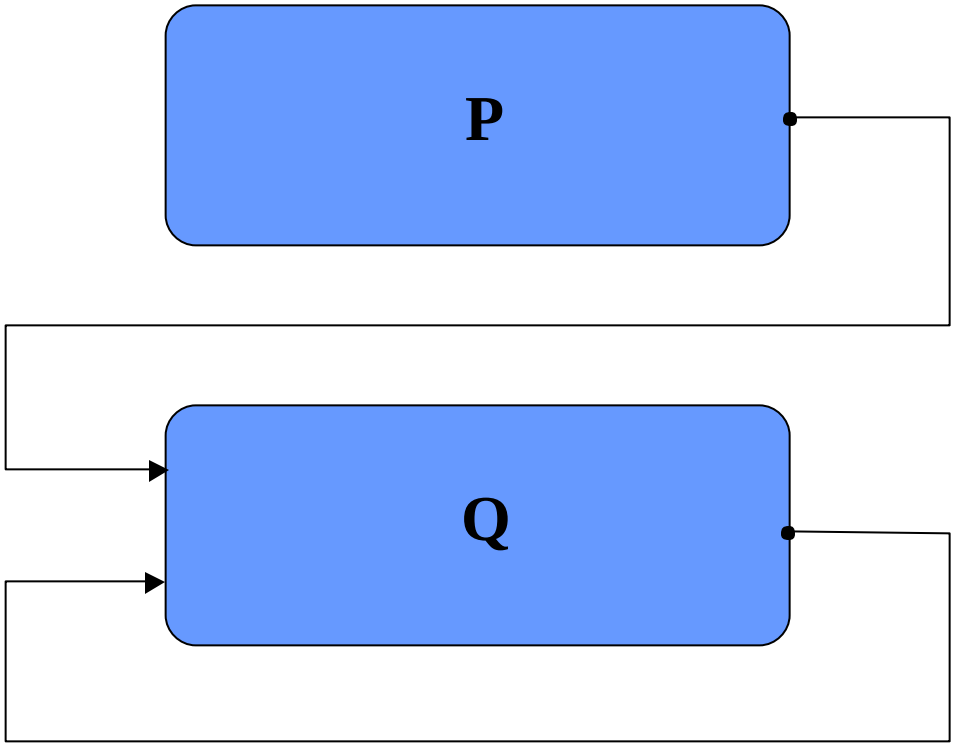
Connections



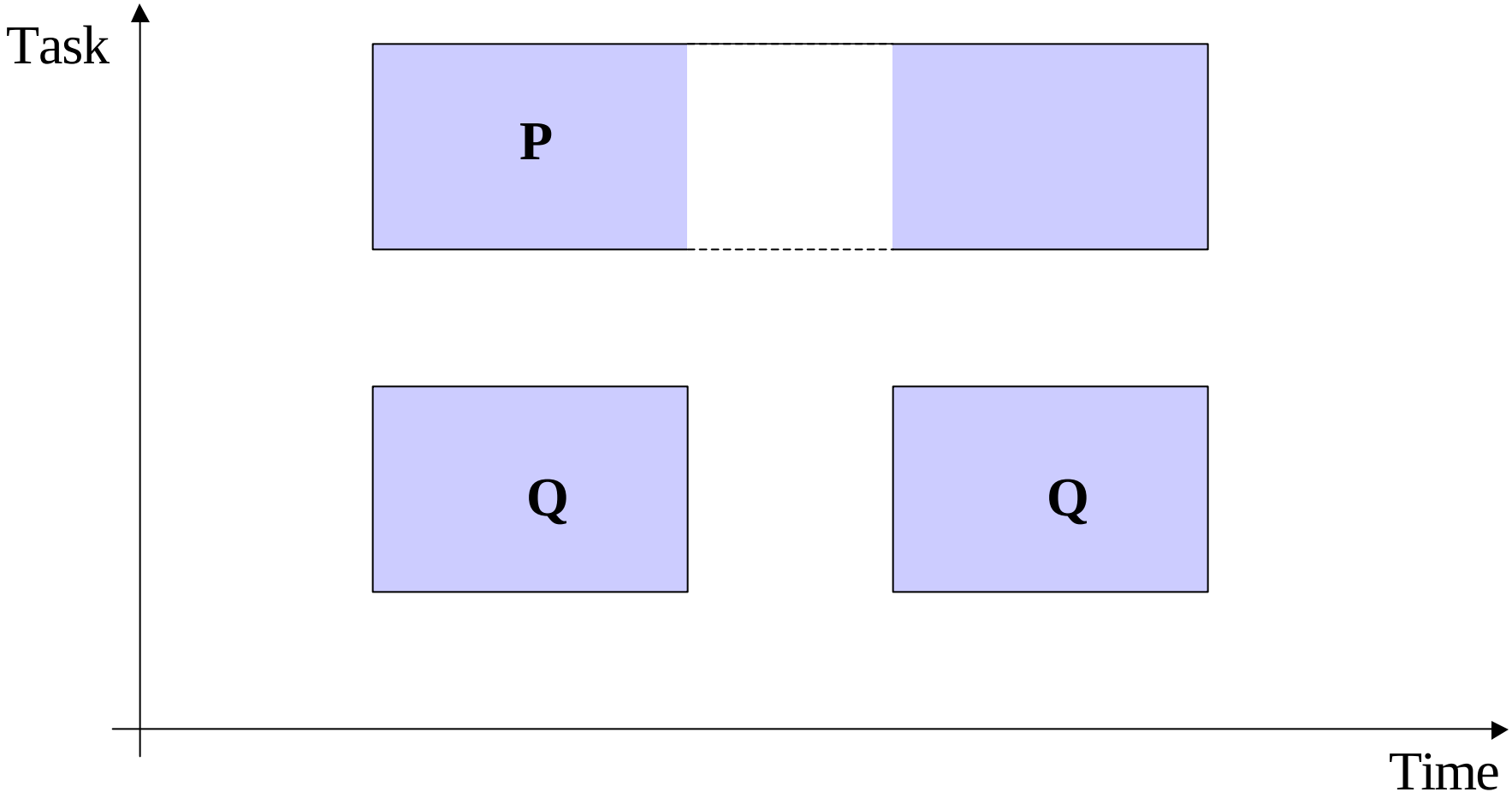
Connections



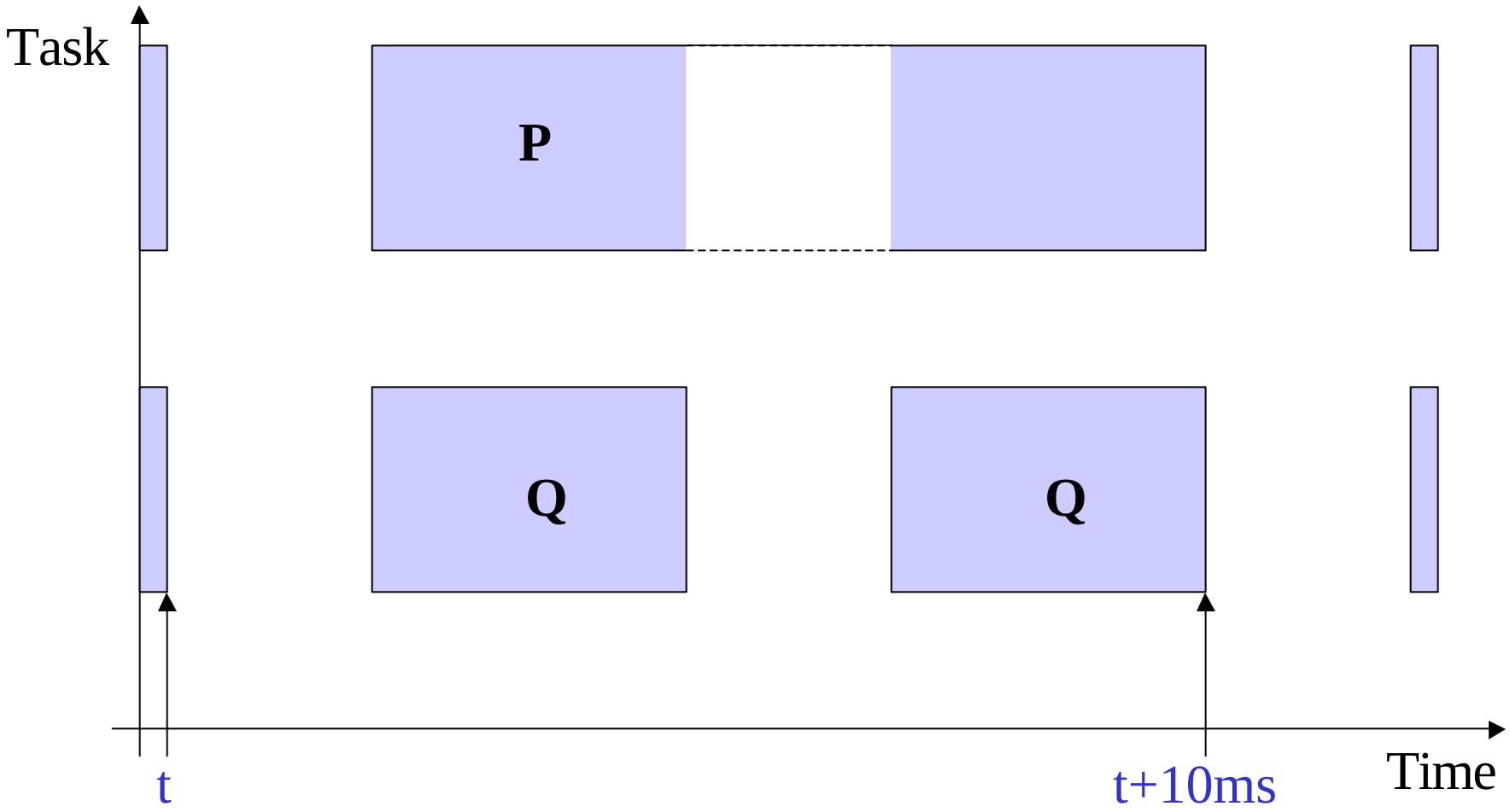
Connections



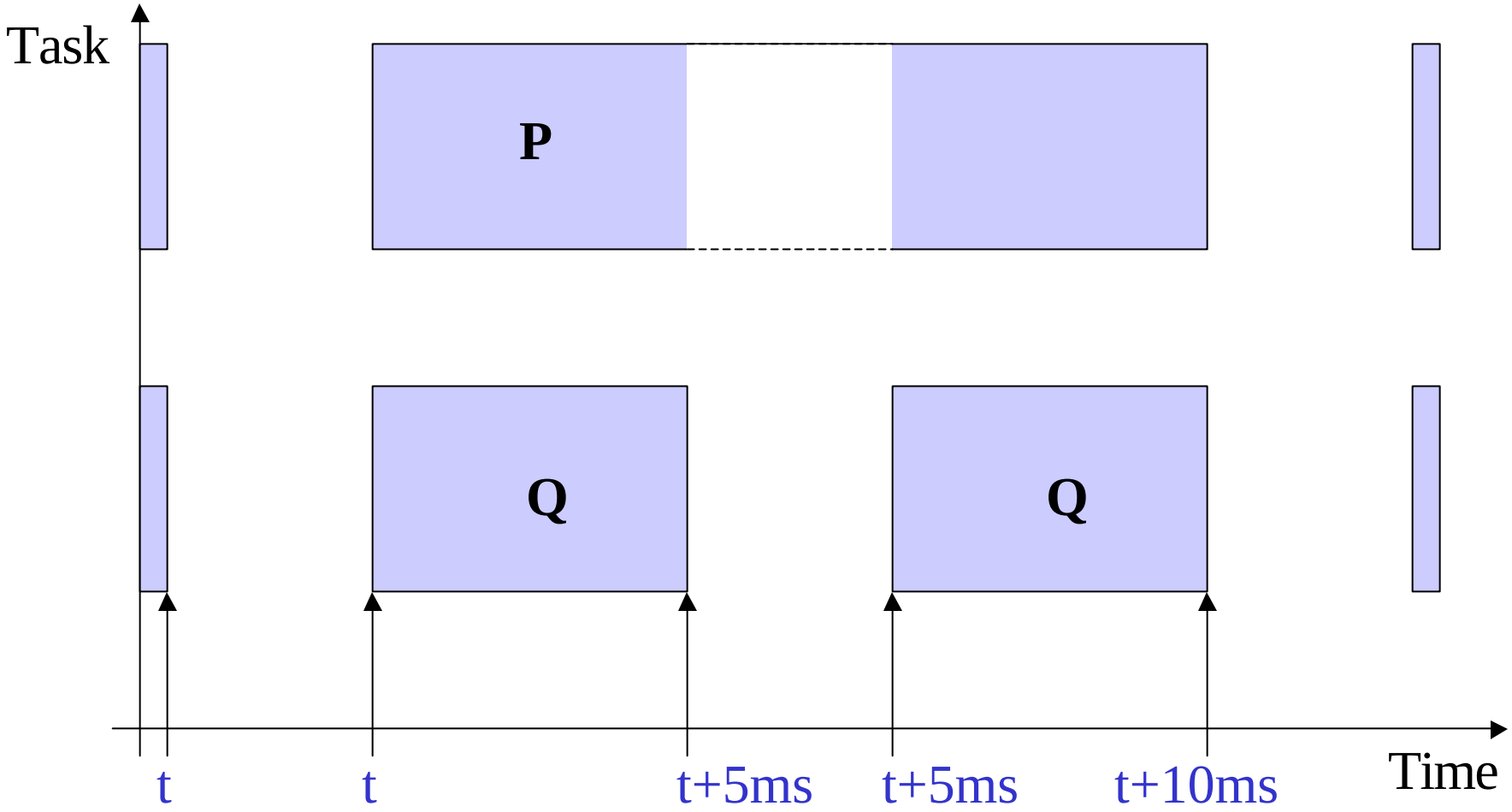
Different Periodicity



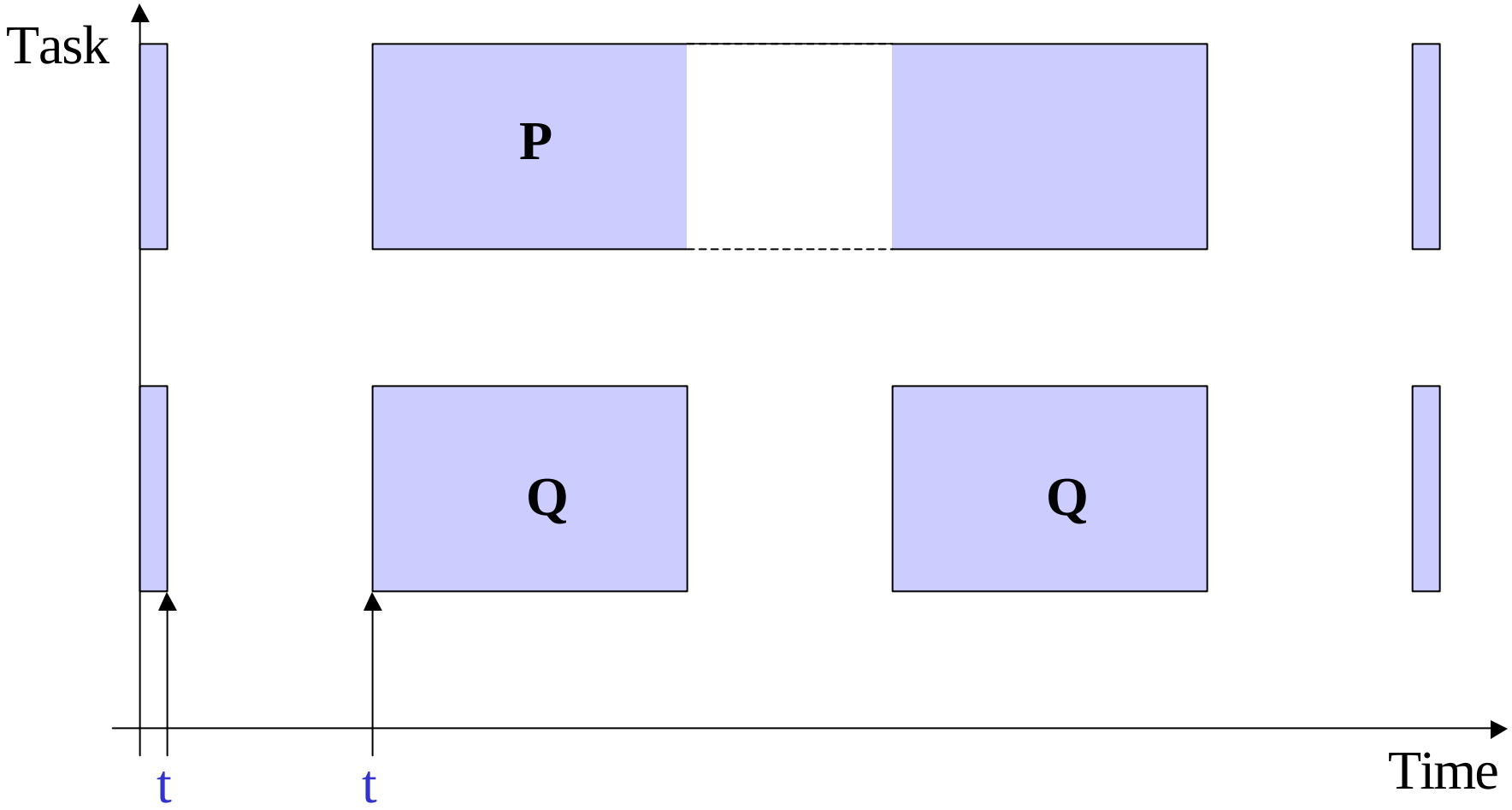
Semantics



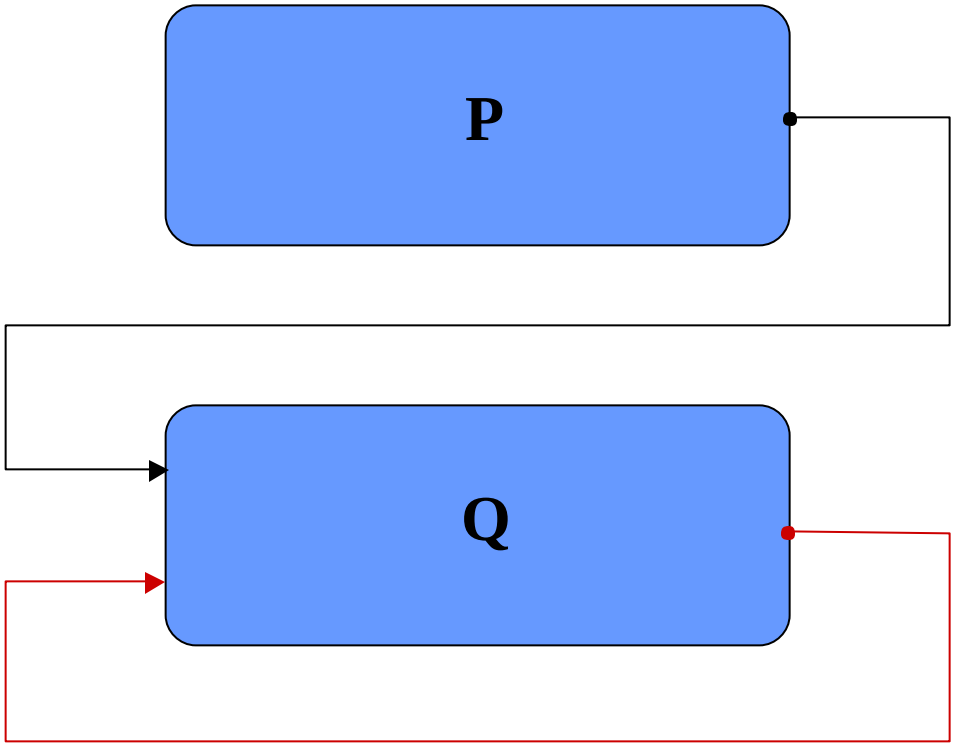
Semantics



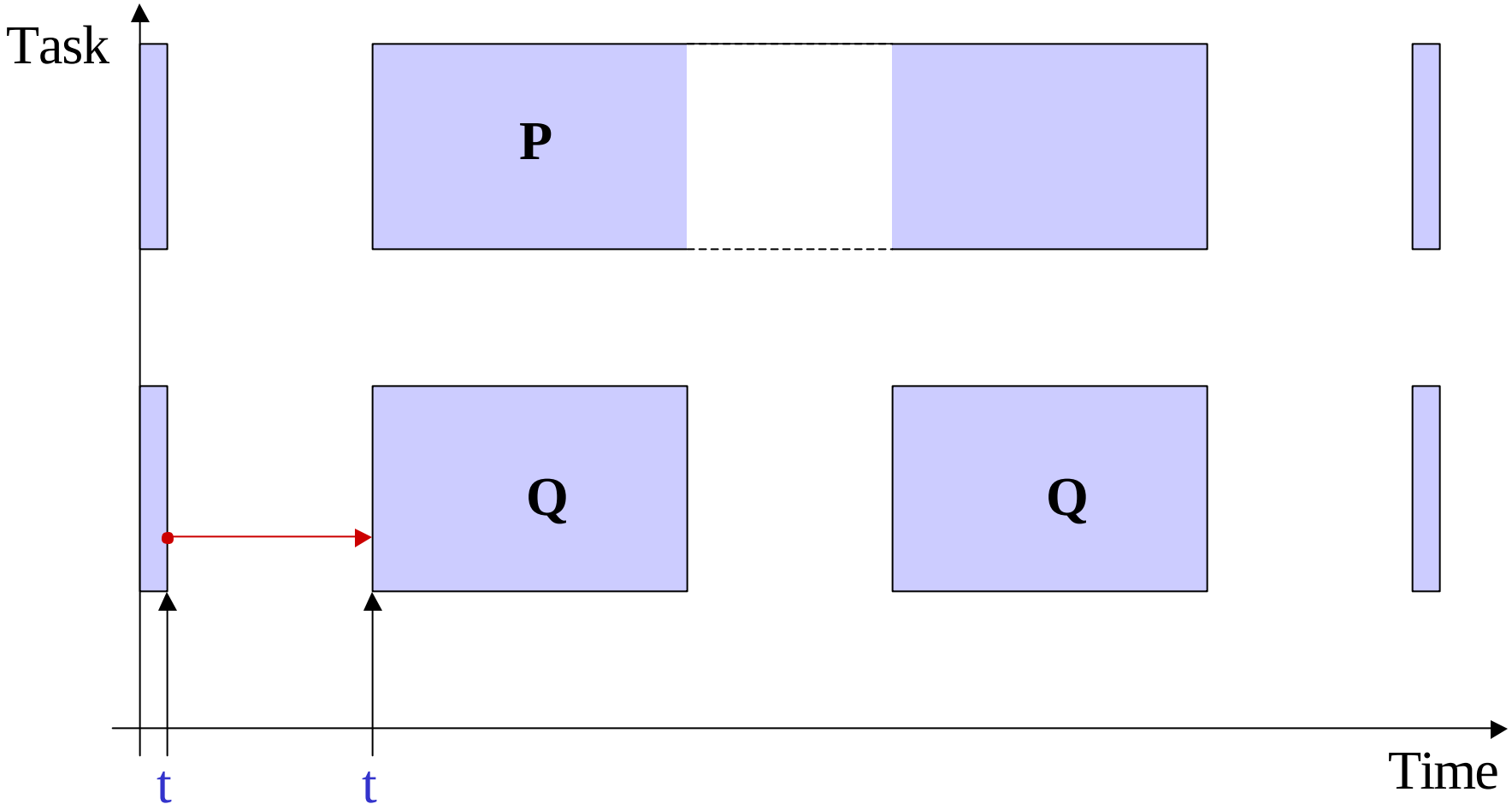
Semantics



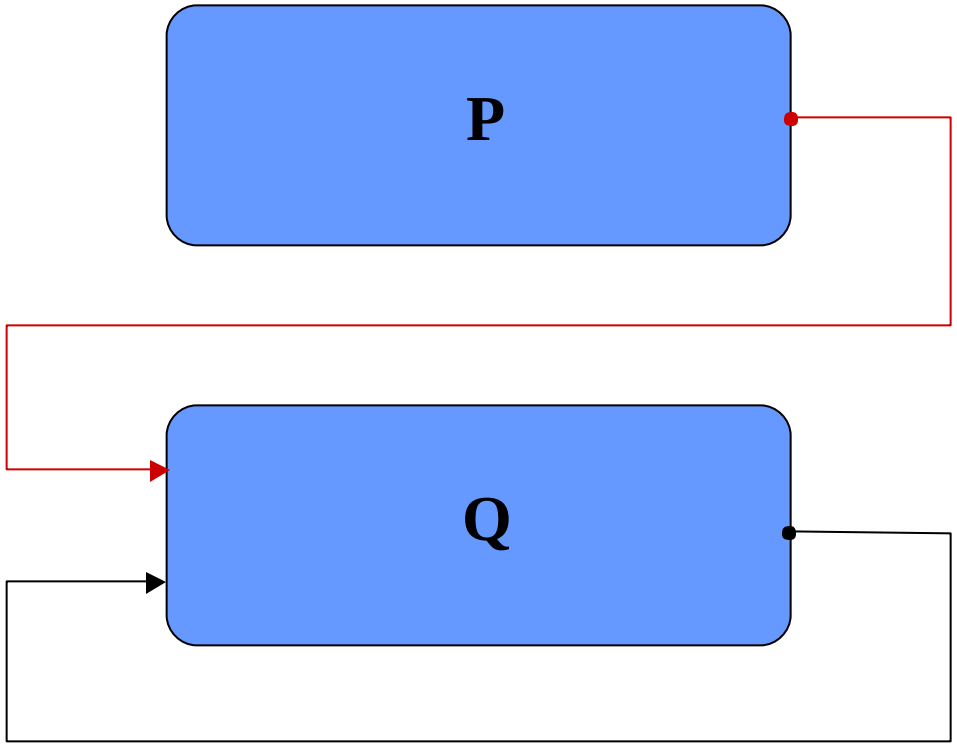
Connections



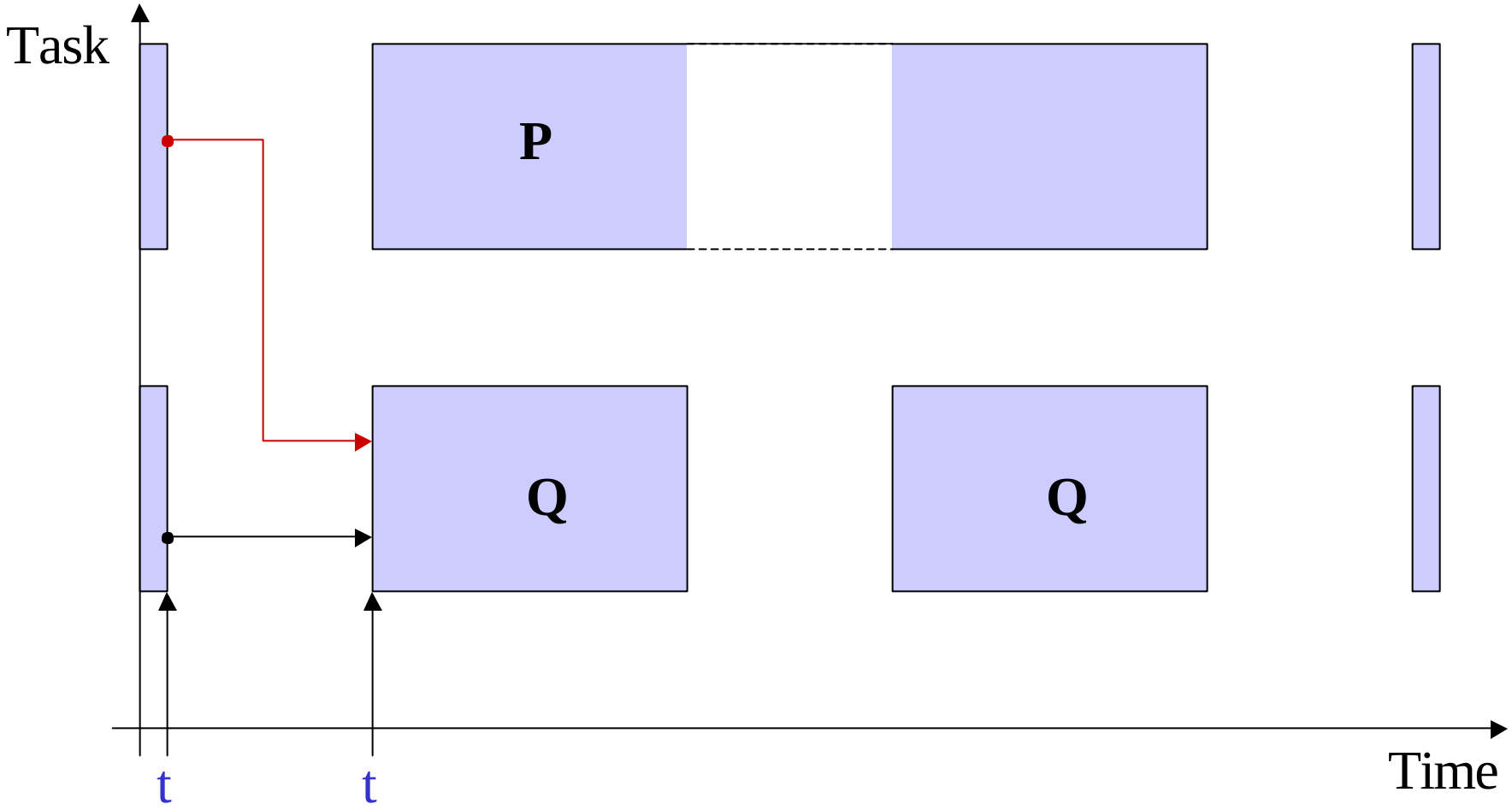
Semantics



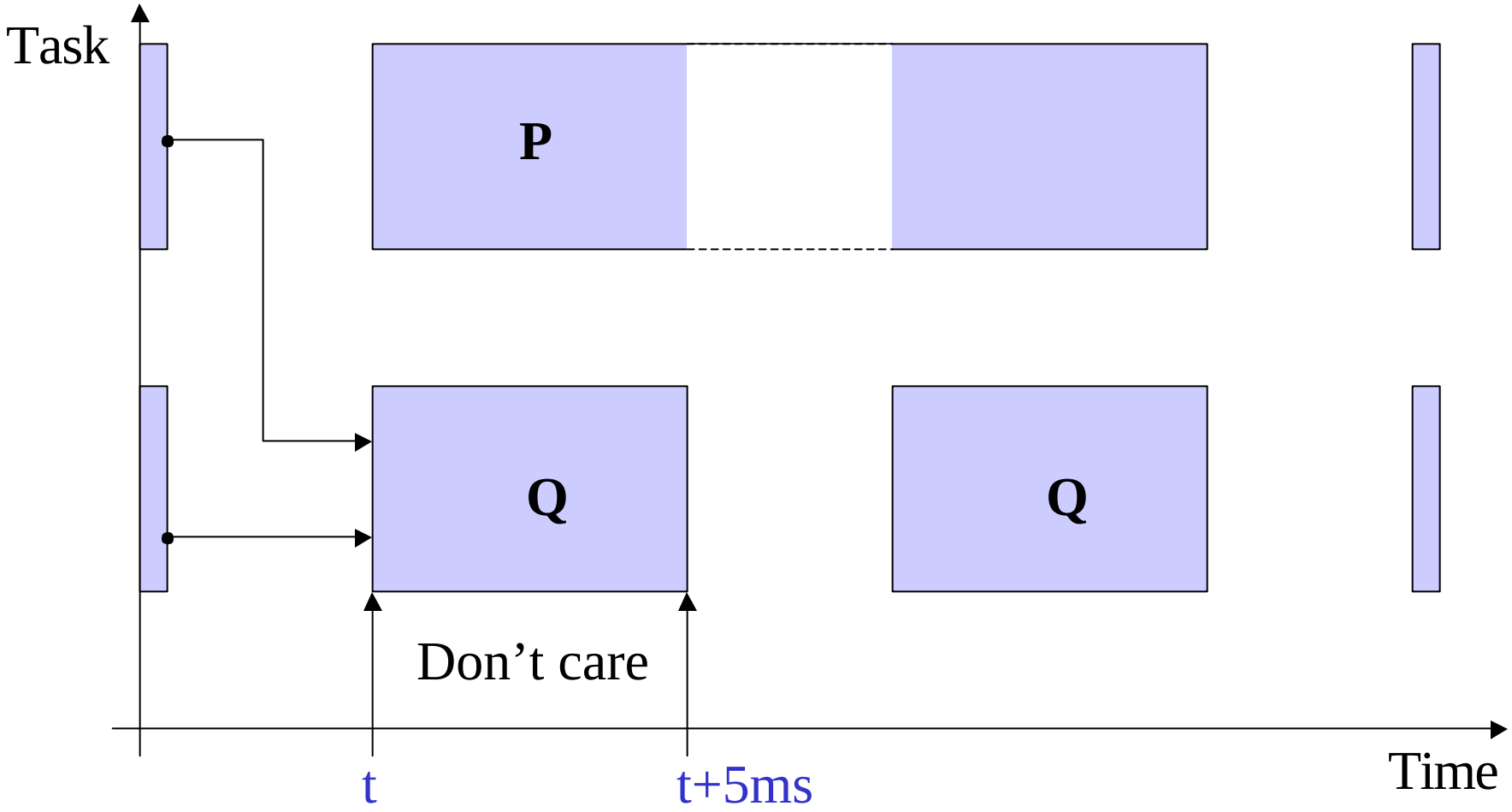
Connections



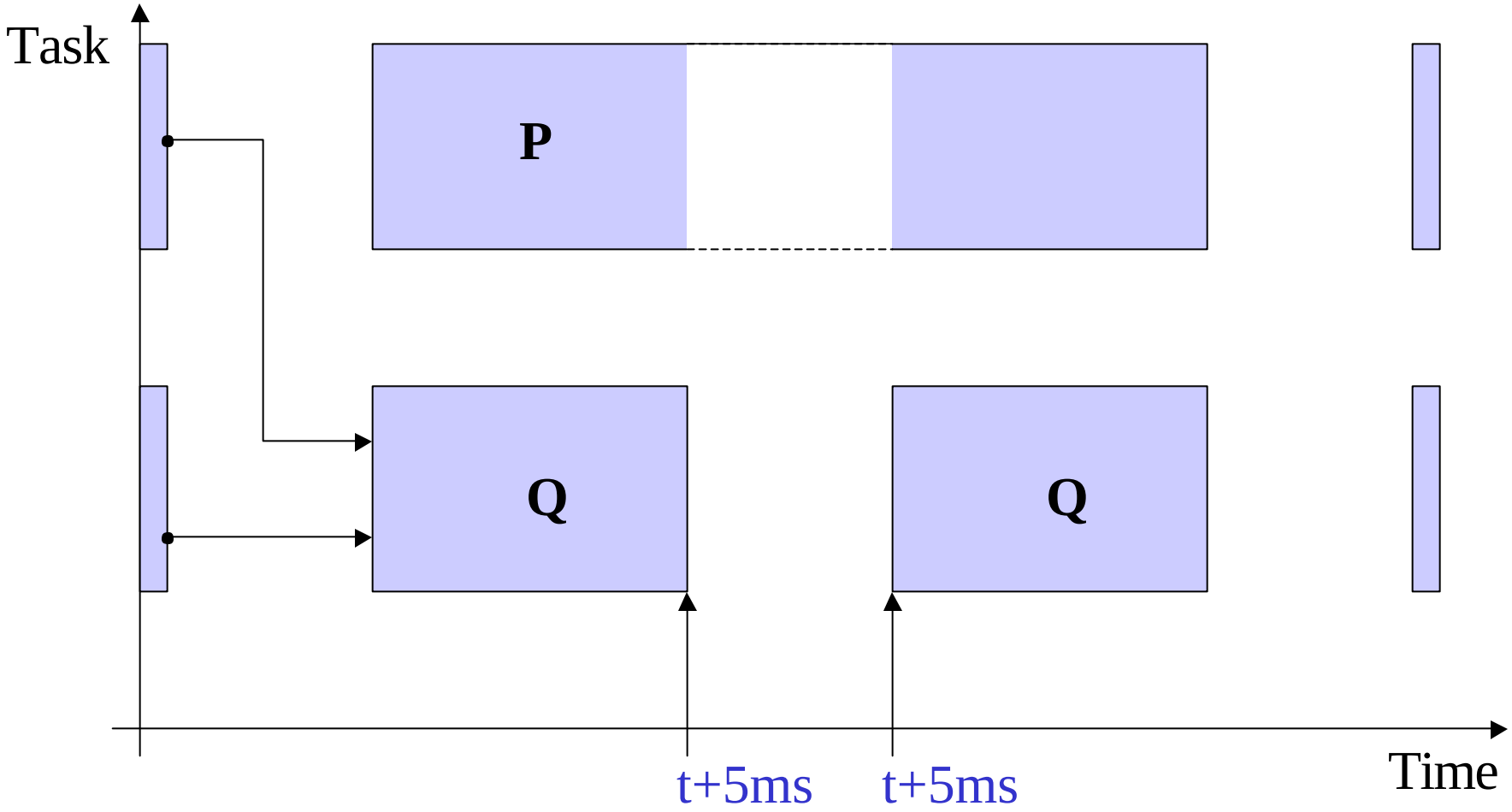
Semantics



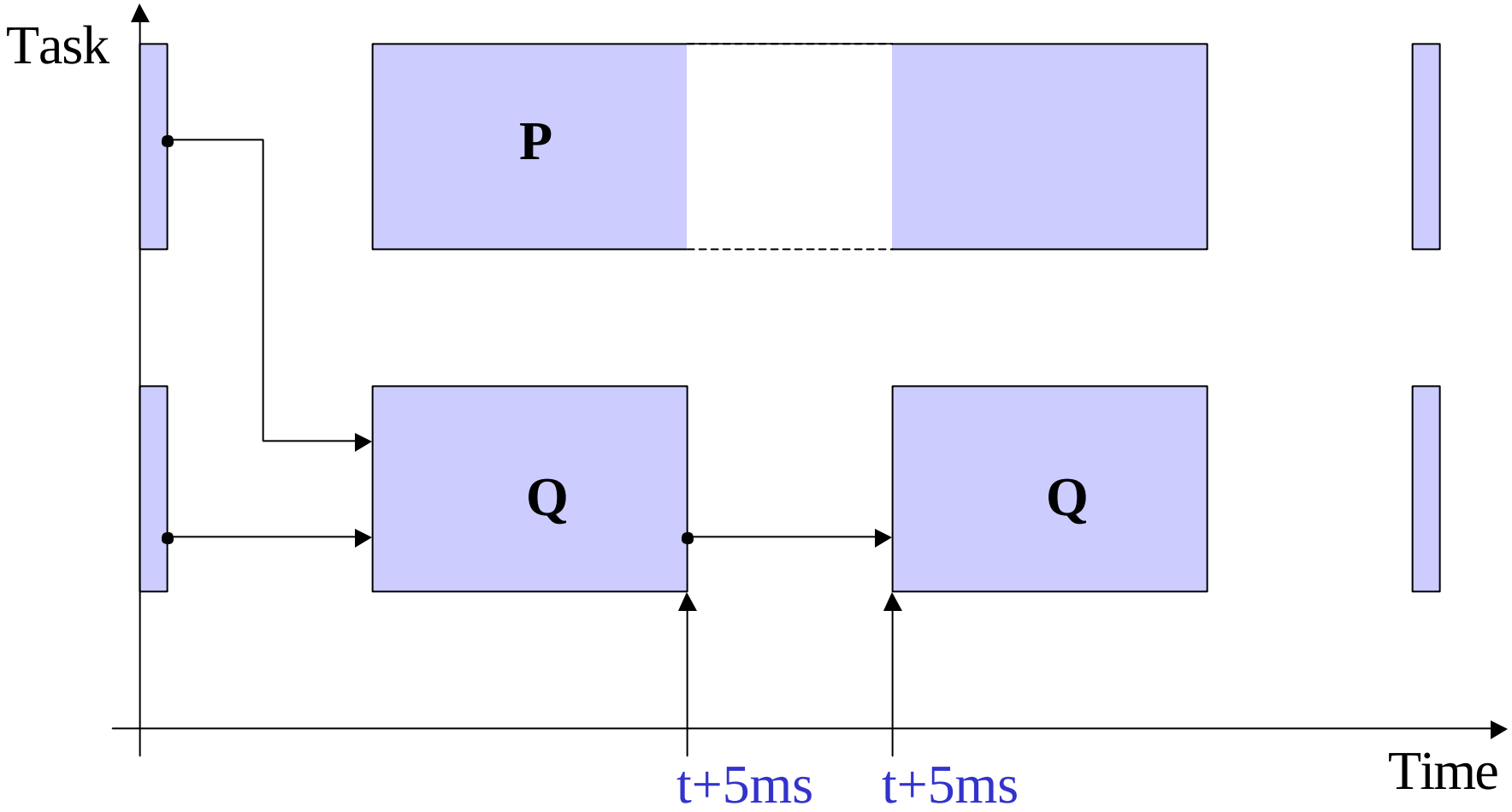
Semantics



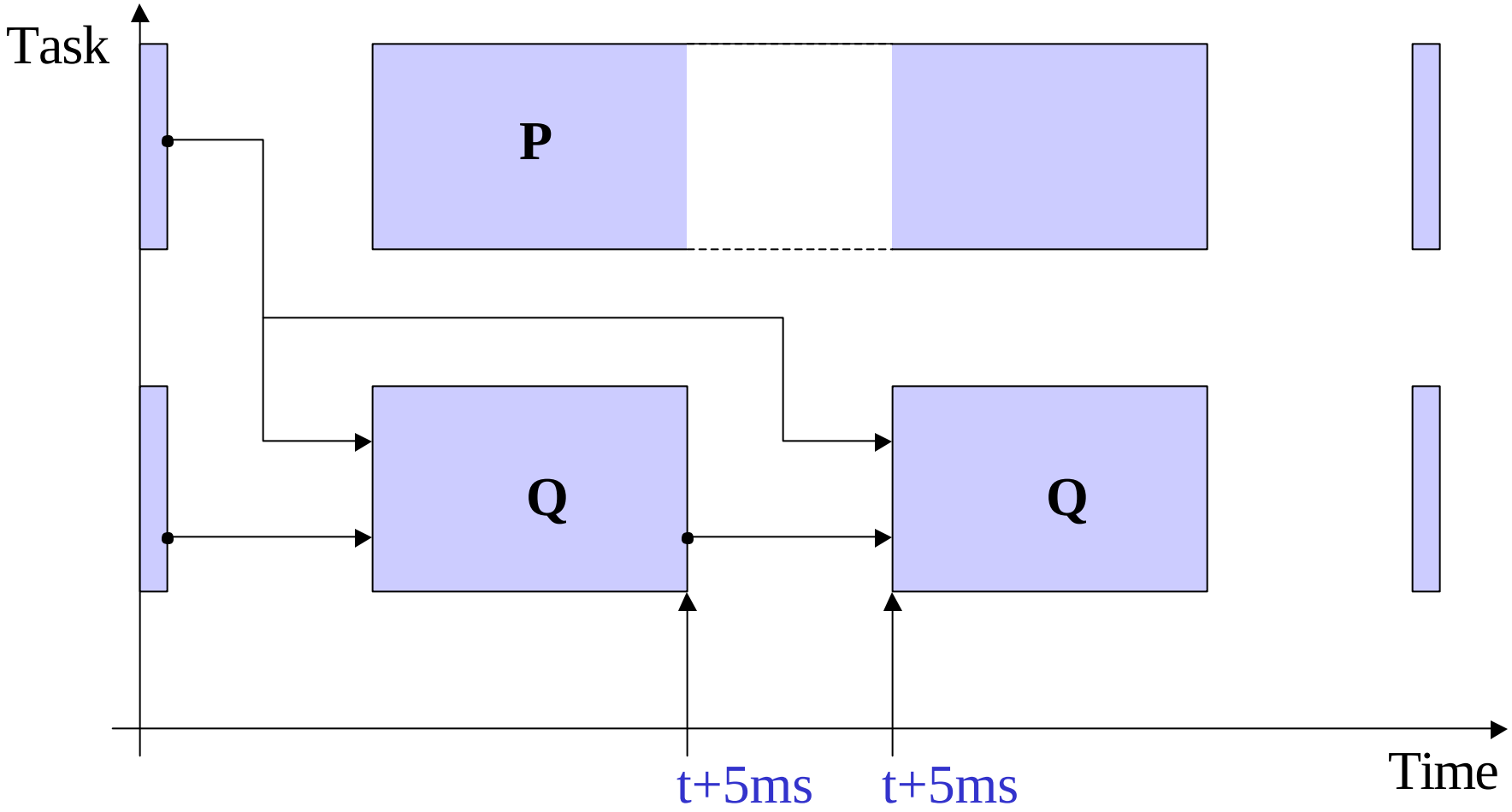
Semantics



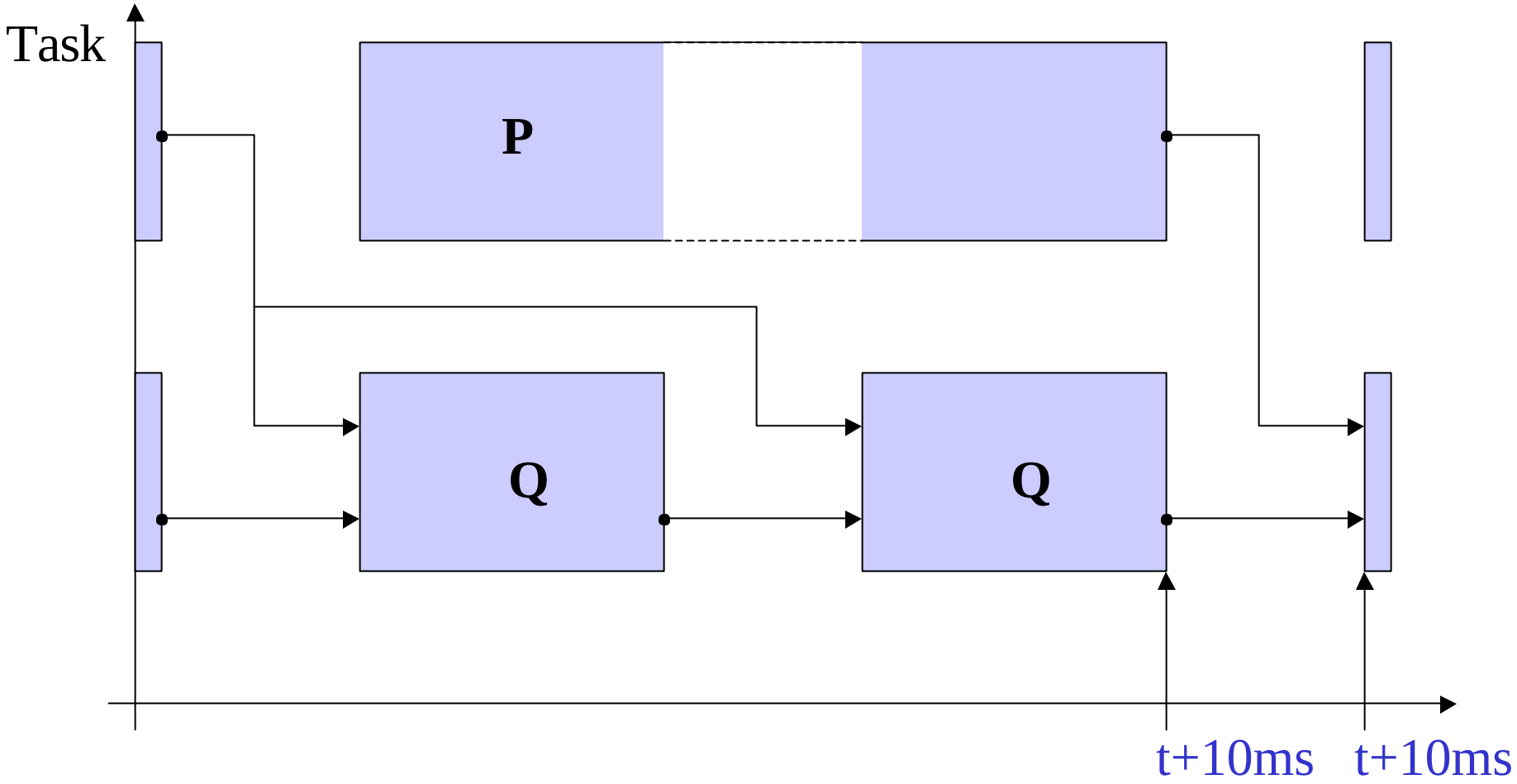
Semantics



Semantics



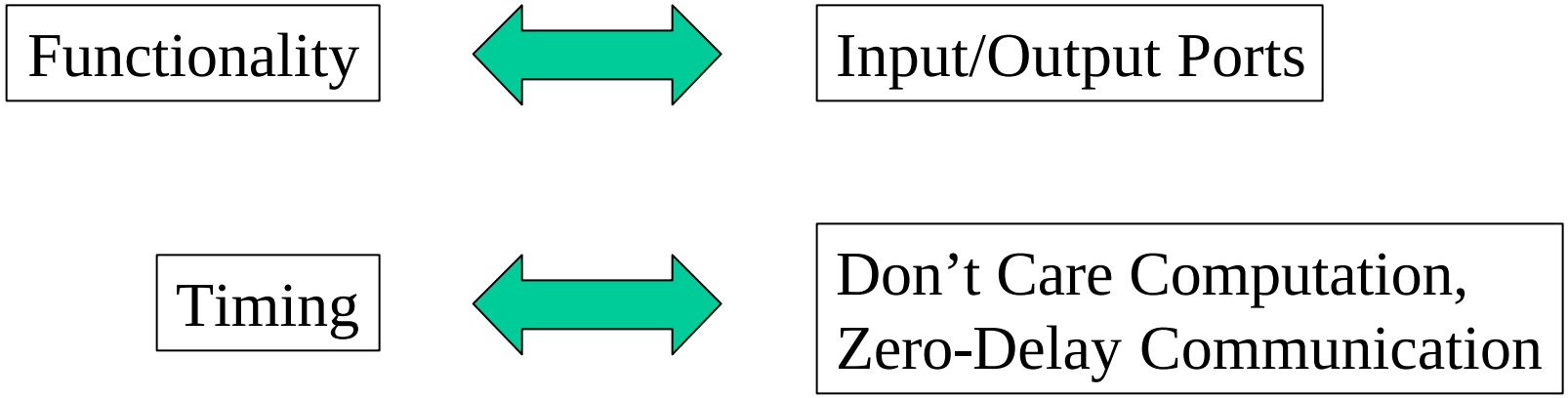
Semantics



Embedded System Development

High-Level Programming

Our Approach



Decomposition: Giotto Modes

Some Motivations:

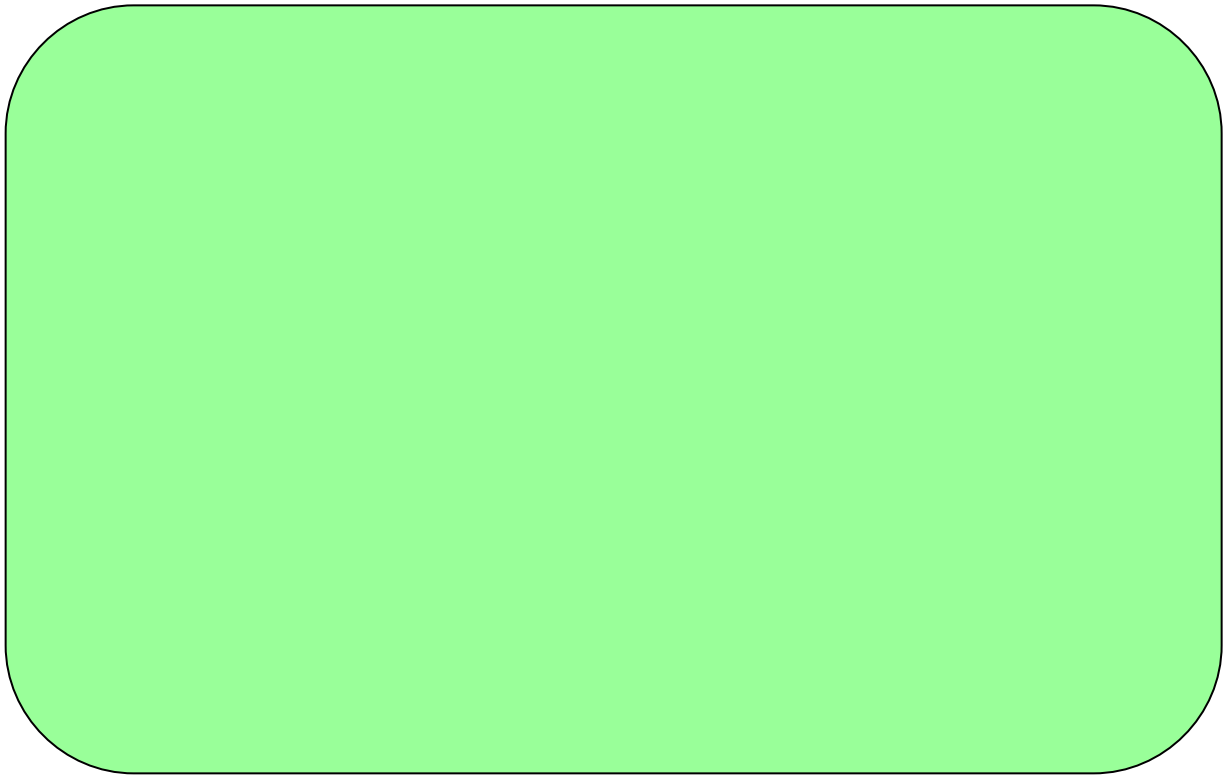
- Multi-modal control
- Fault tolerance
- Events
- Resource sharing
- Uncertain environments

Modes

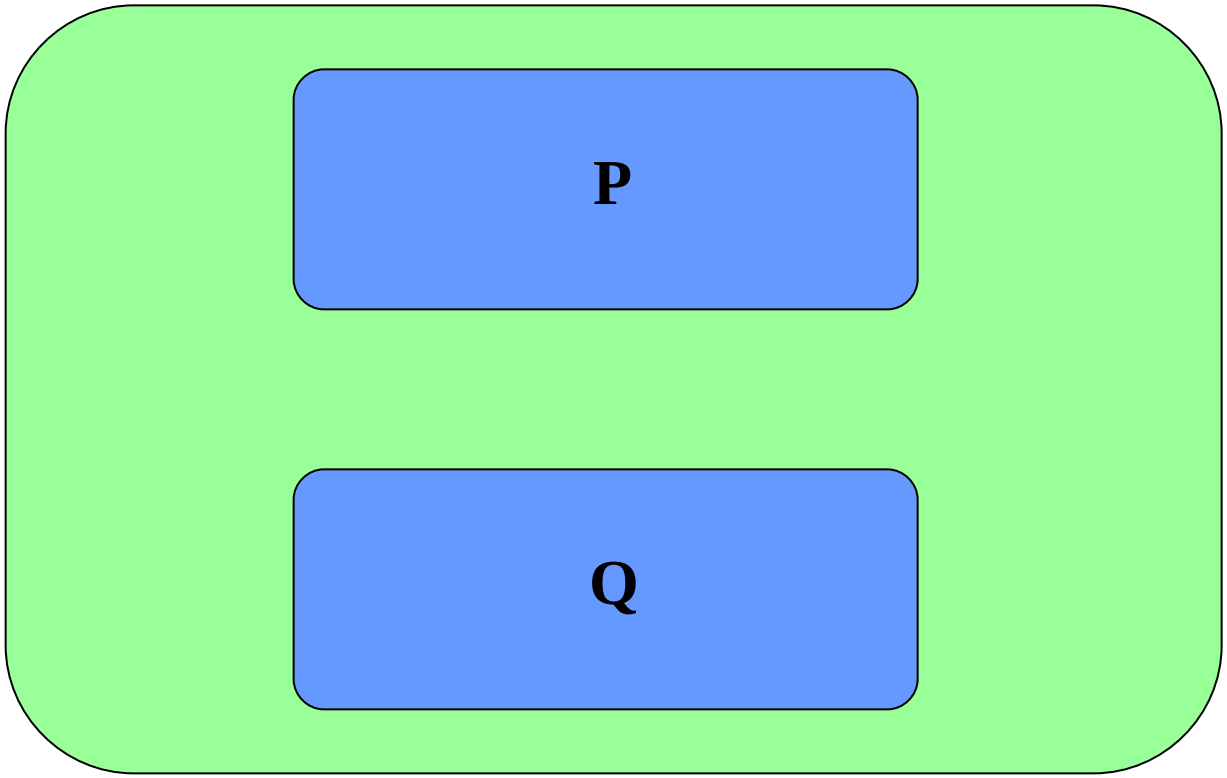
- A mode is a parameterized set of tasks
- A Giotto program consists of a set of modes and mode switches

- A Giotto system is in a single mode at the same time

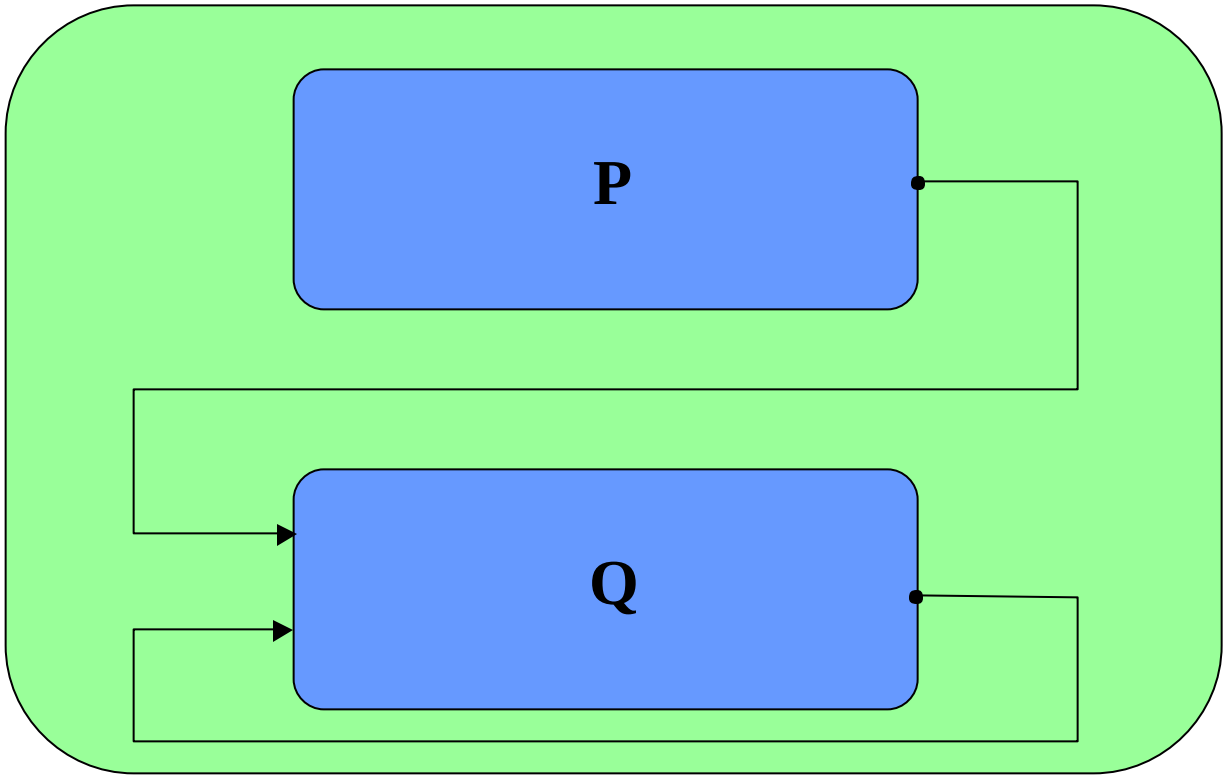
A Mode



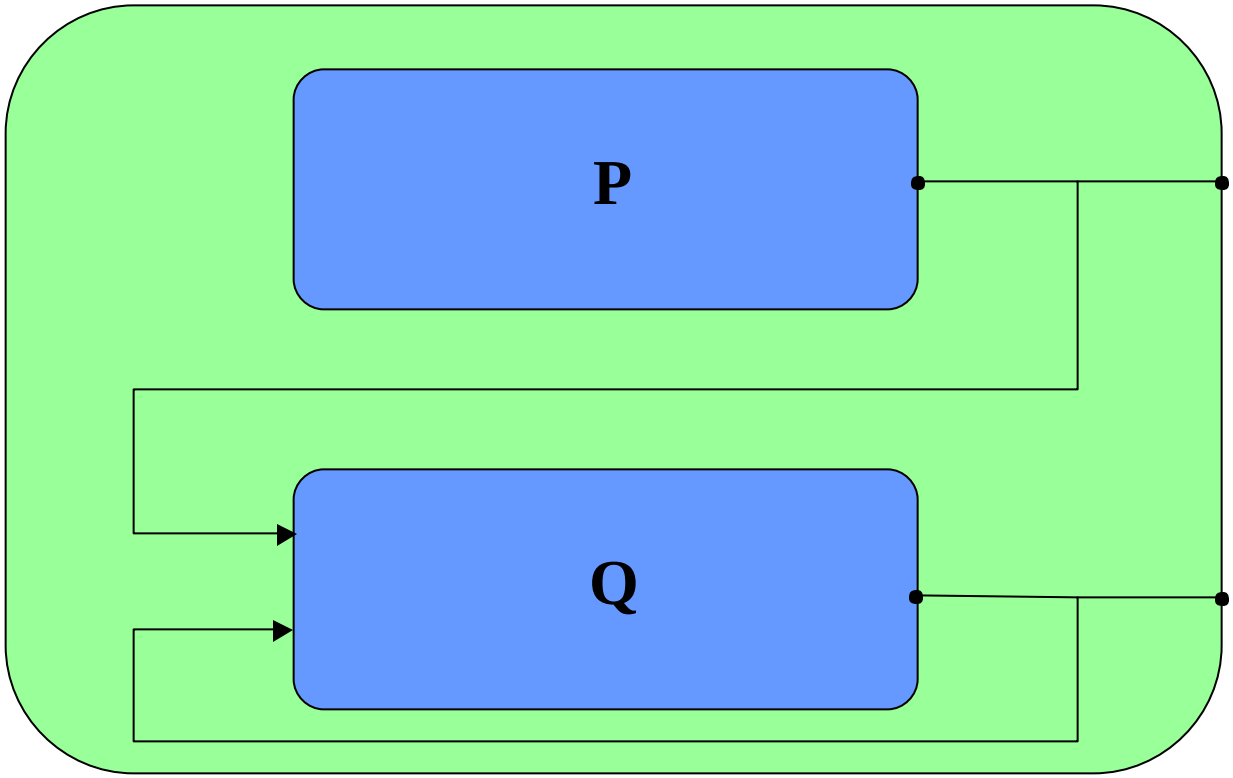
Abstract Syntax of a Mode



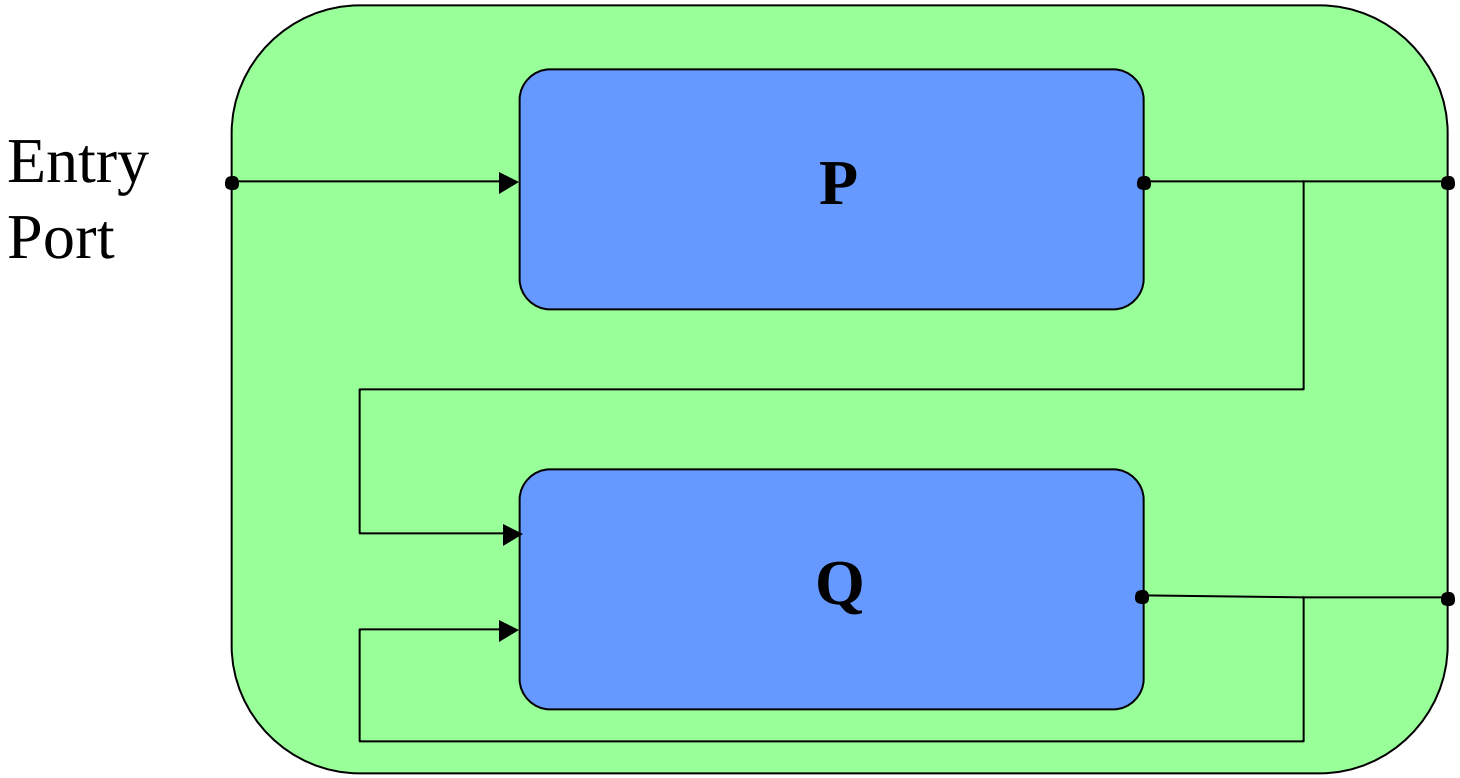
Abstract Syntax of a Mode



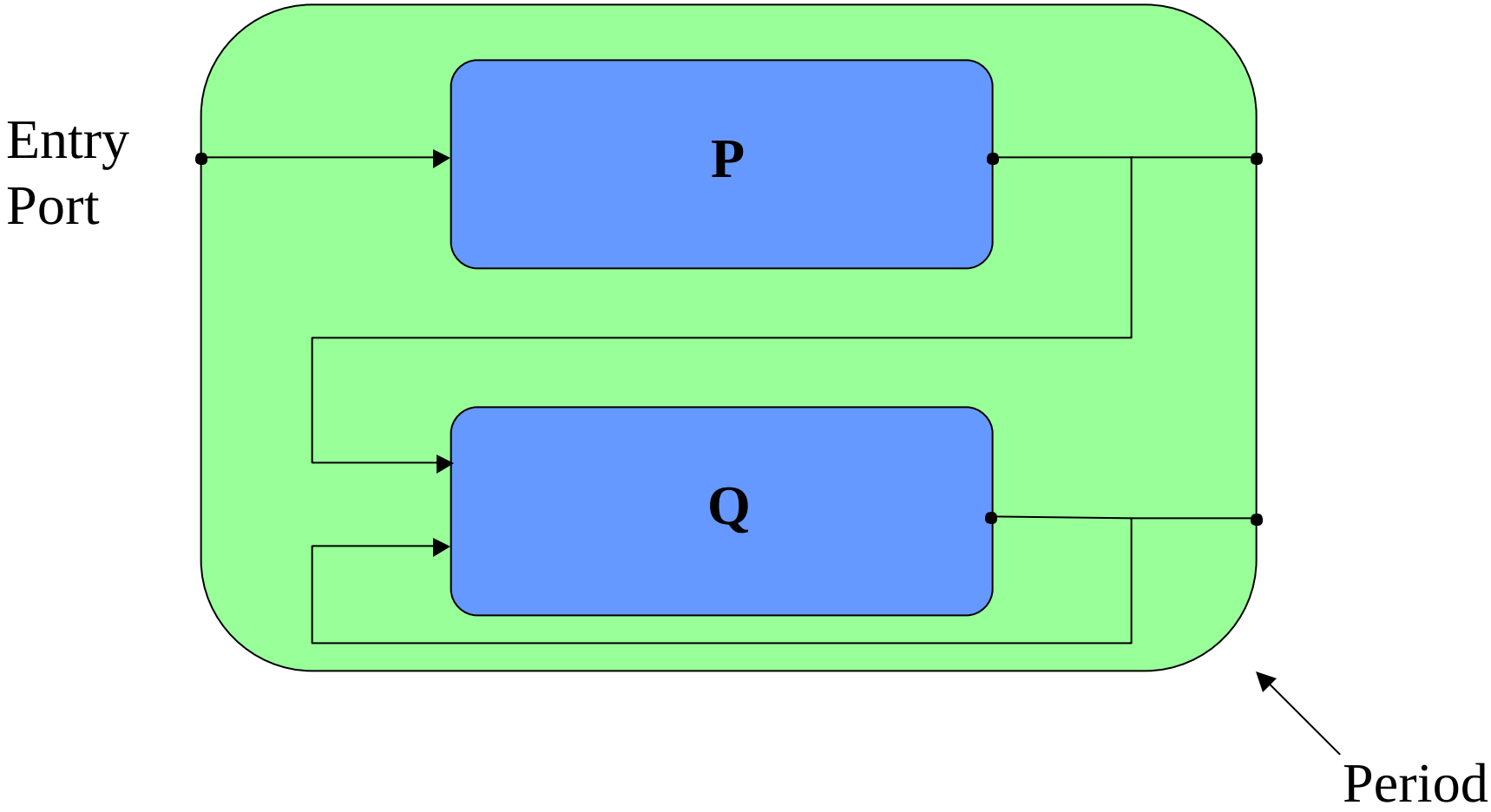
Abstract Syntax of a Mode



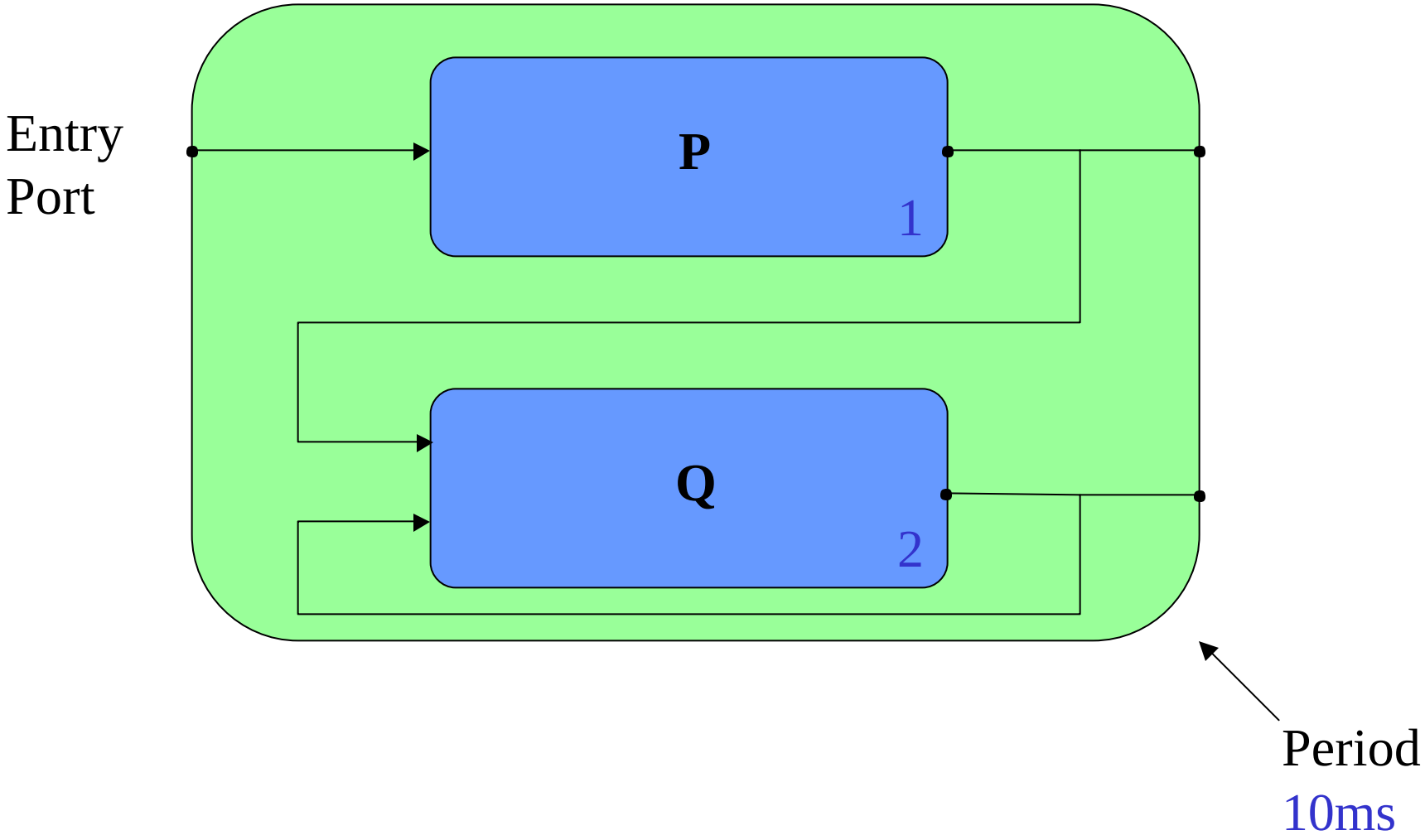
Abstract Syntax of a Mode



Abstract Syntax of a Mode



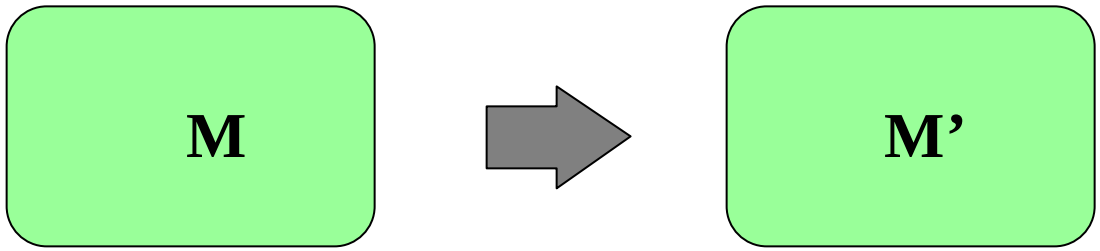
Semantics of a Mode



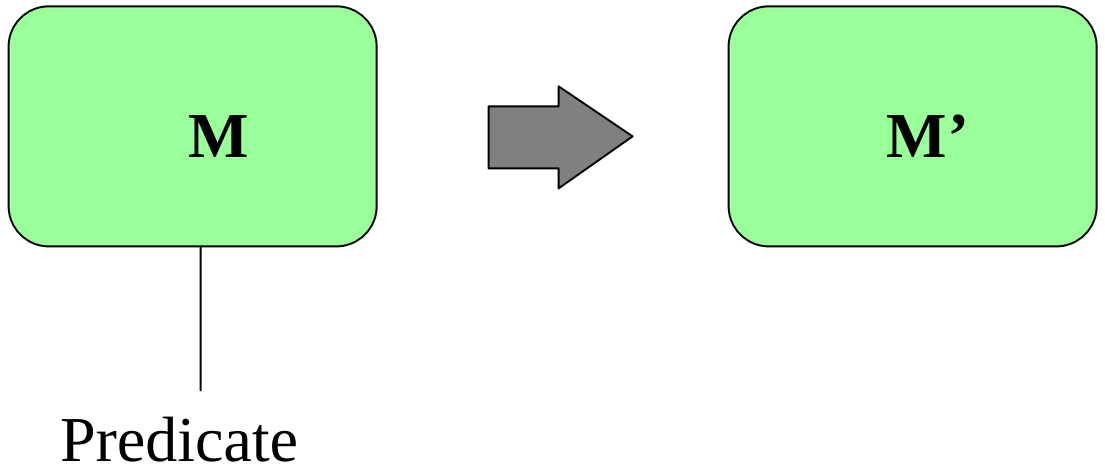
Concrete Syntax

```
mode m ( ) period 10 ms
  {
     taskfreq 1 do int x = P ( ) ;
     taskfreq 2 do int y = Q ( x, y ) ;
  }
```

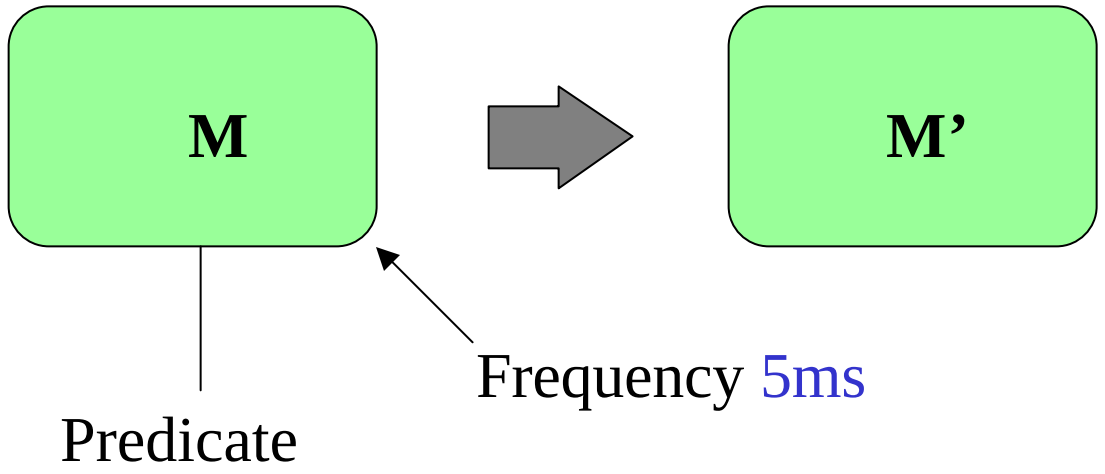
A Mode Switch



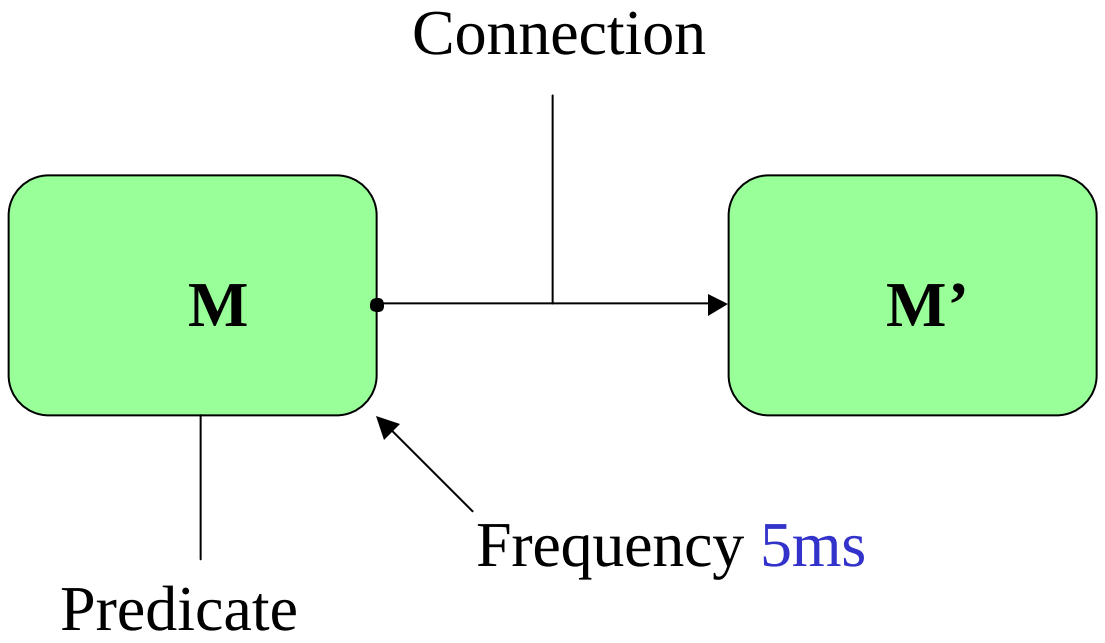
Abstract Syntax of a Mode Switch



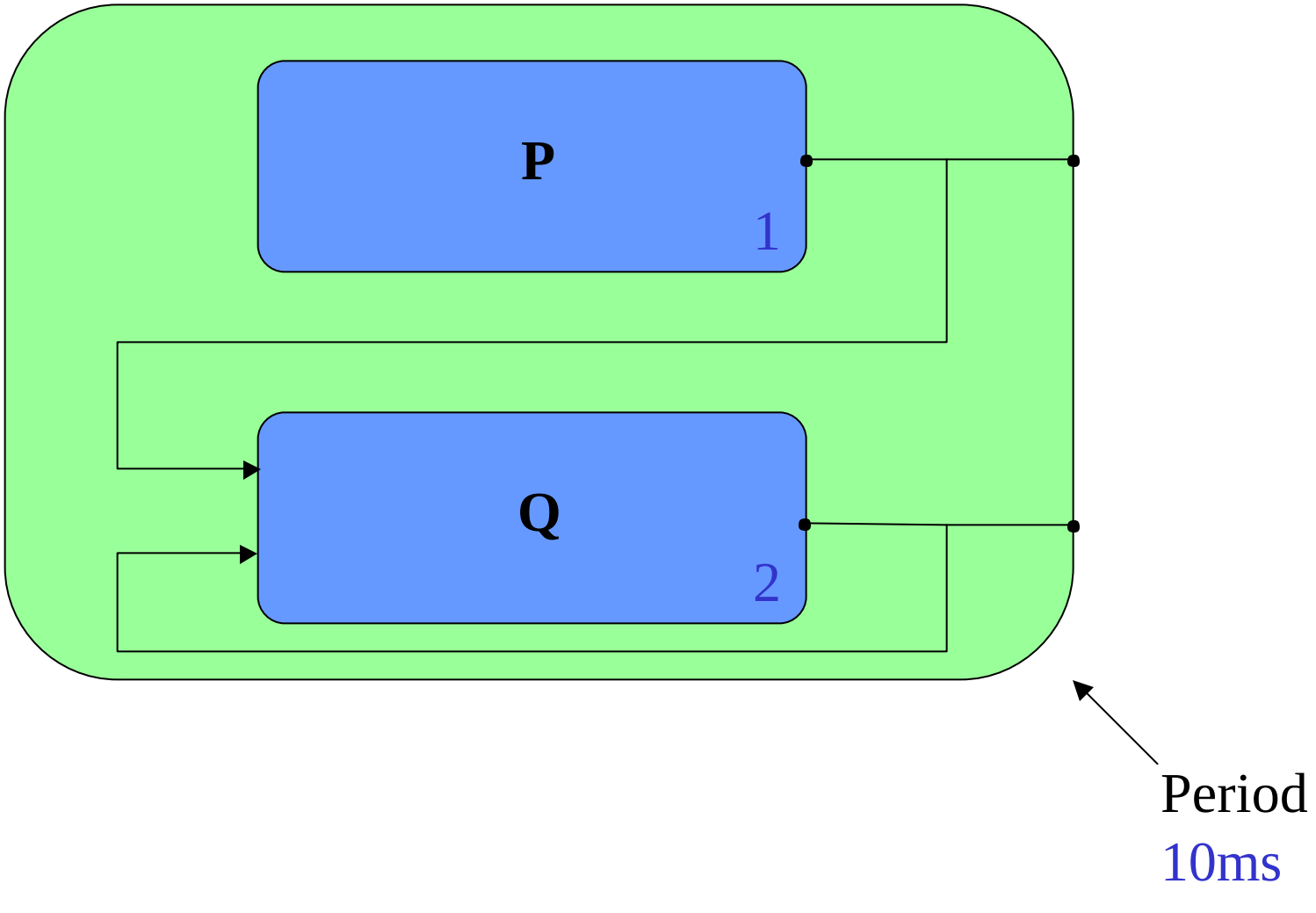
Abstract Syntax of a Mode Switch



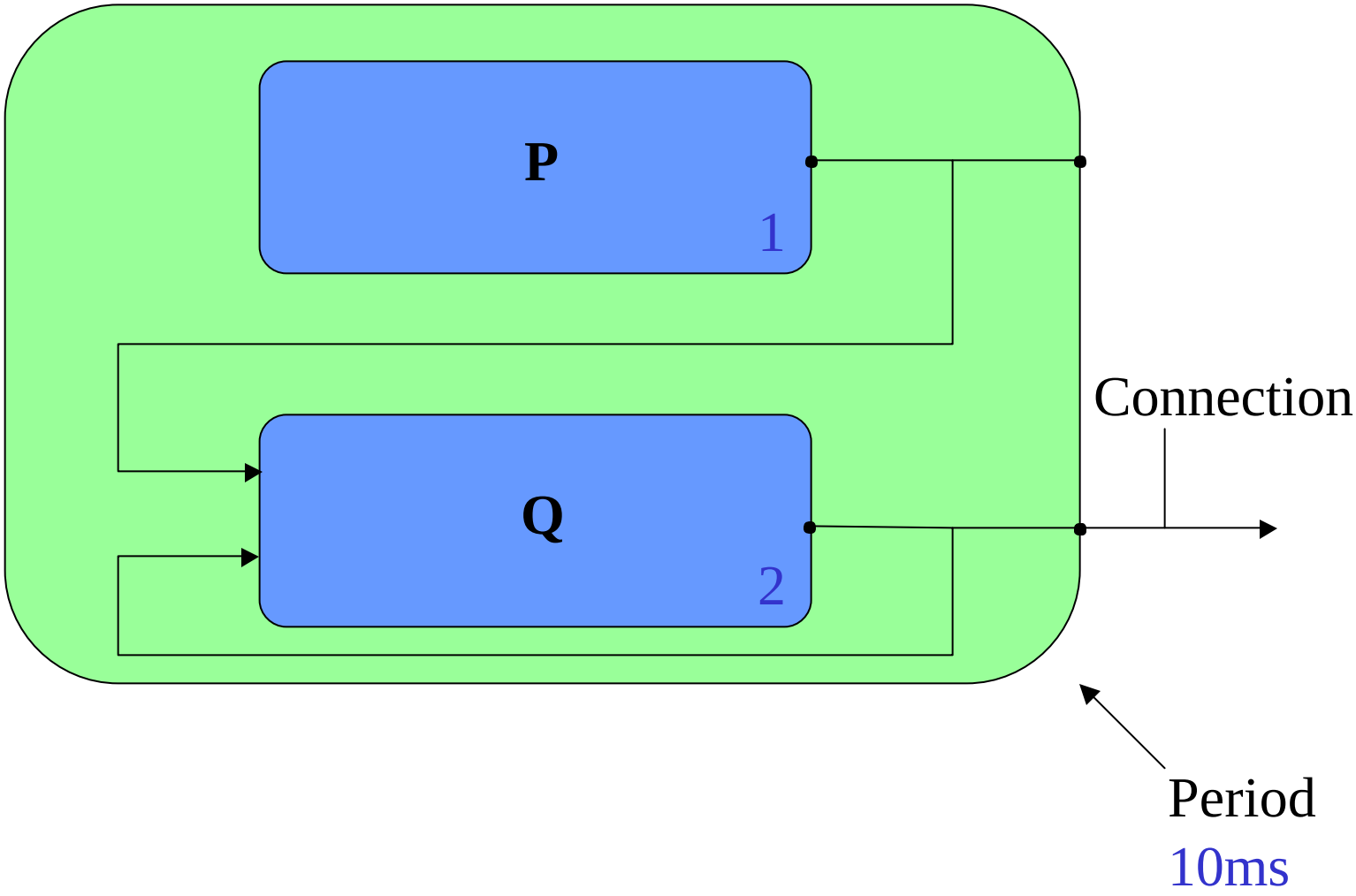
Abstract Syntax of a Mode Switch



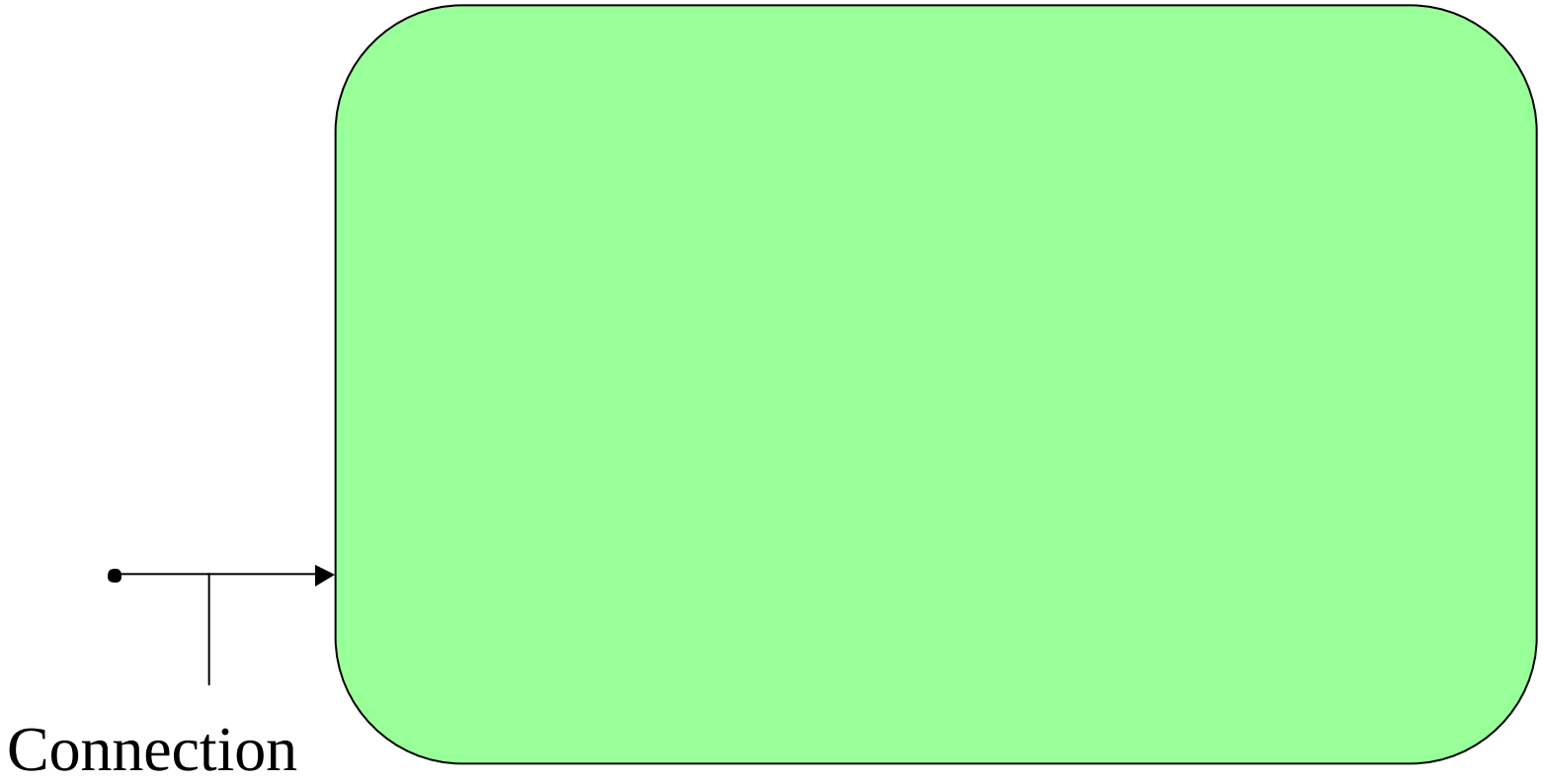
Mode M



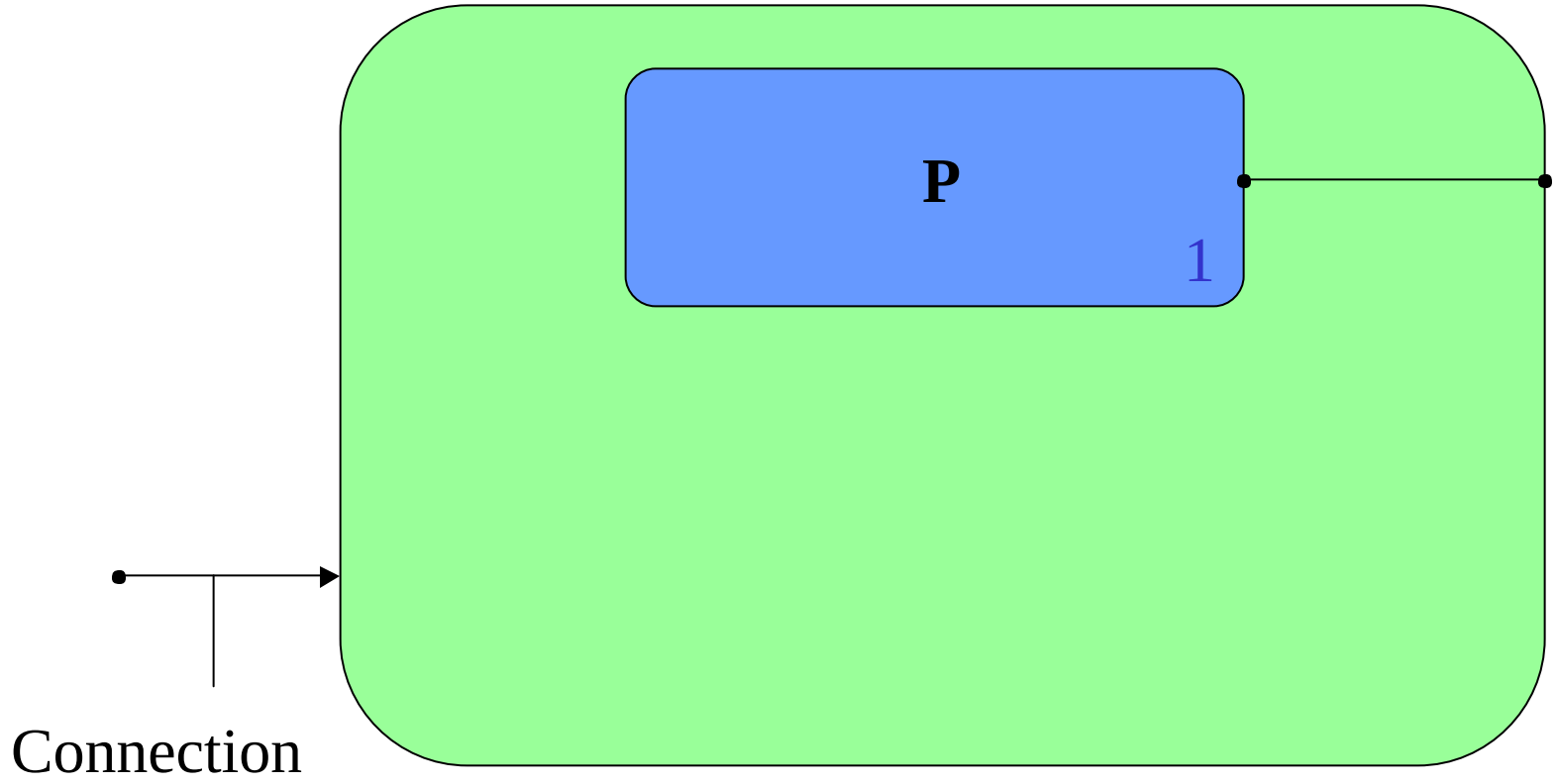
Mode M



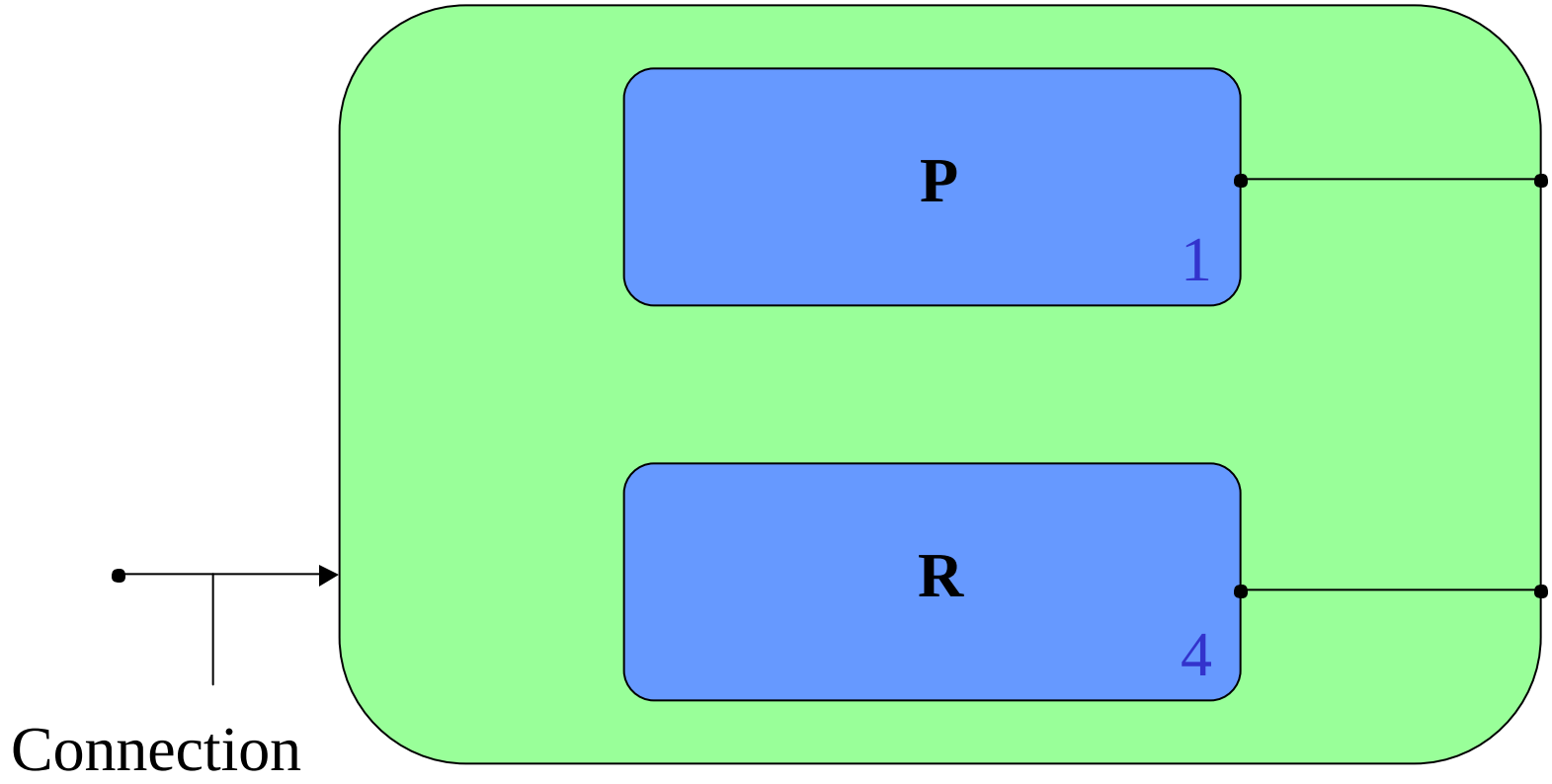
Mode M’



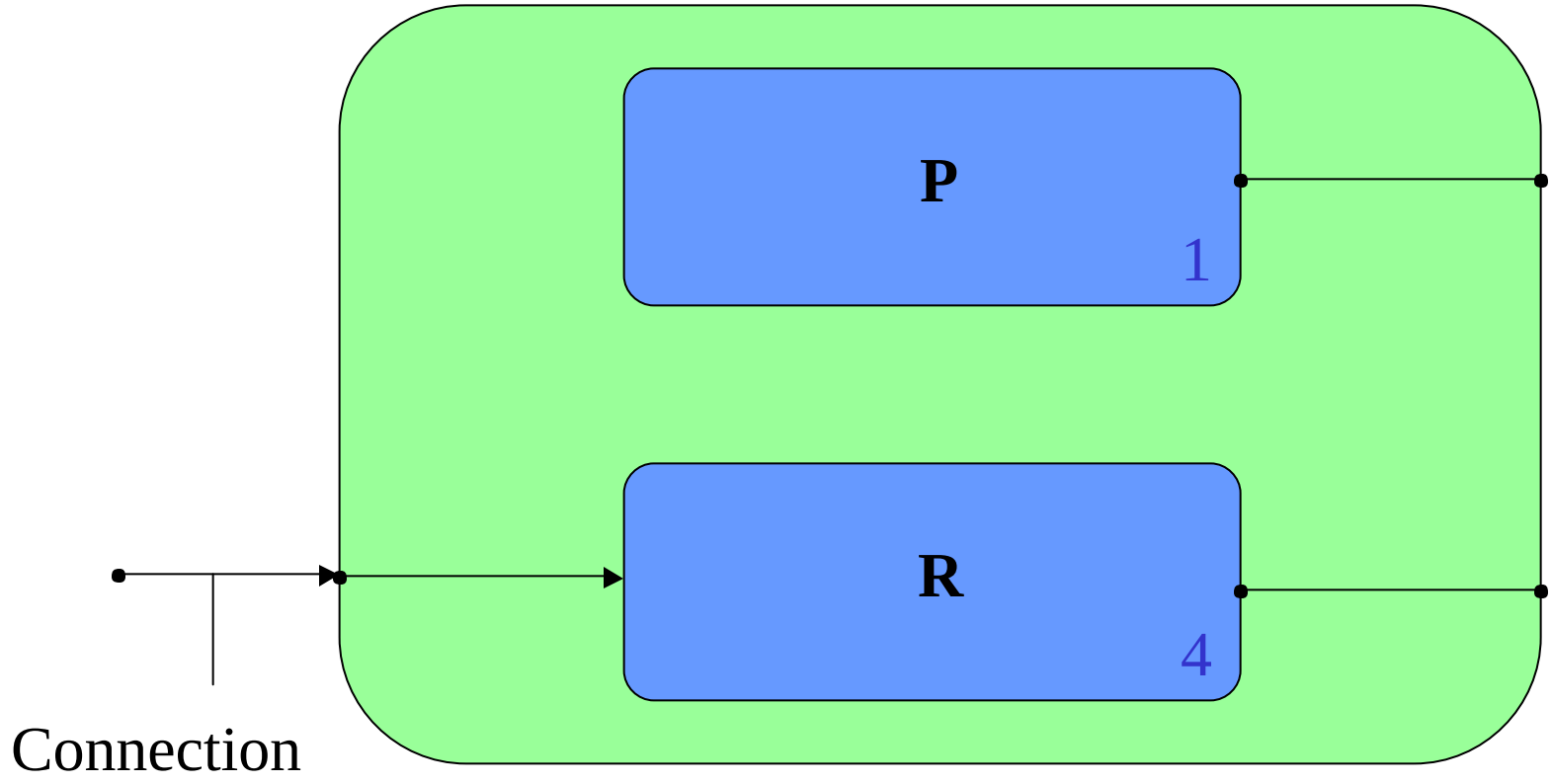
Mode M’



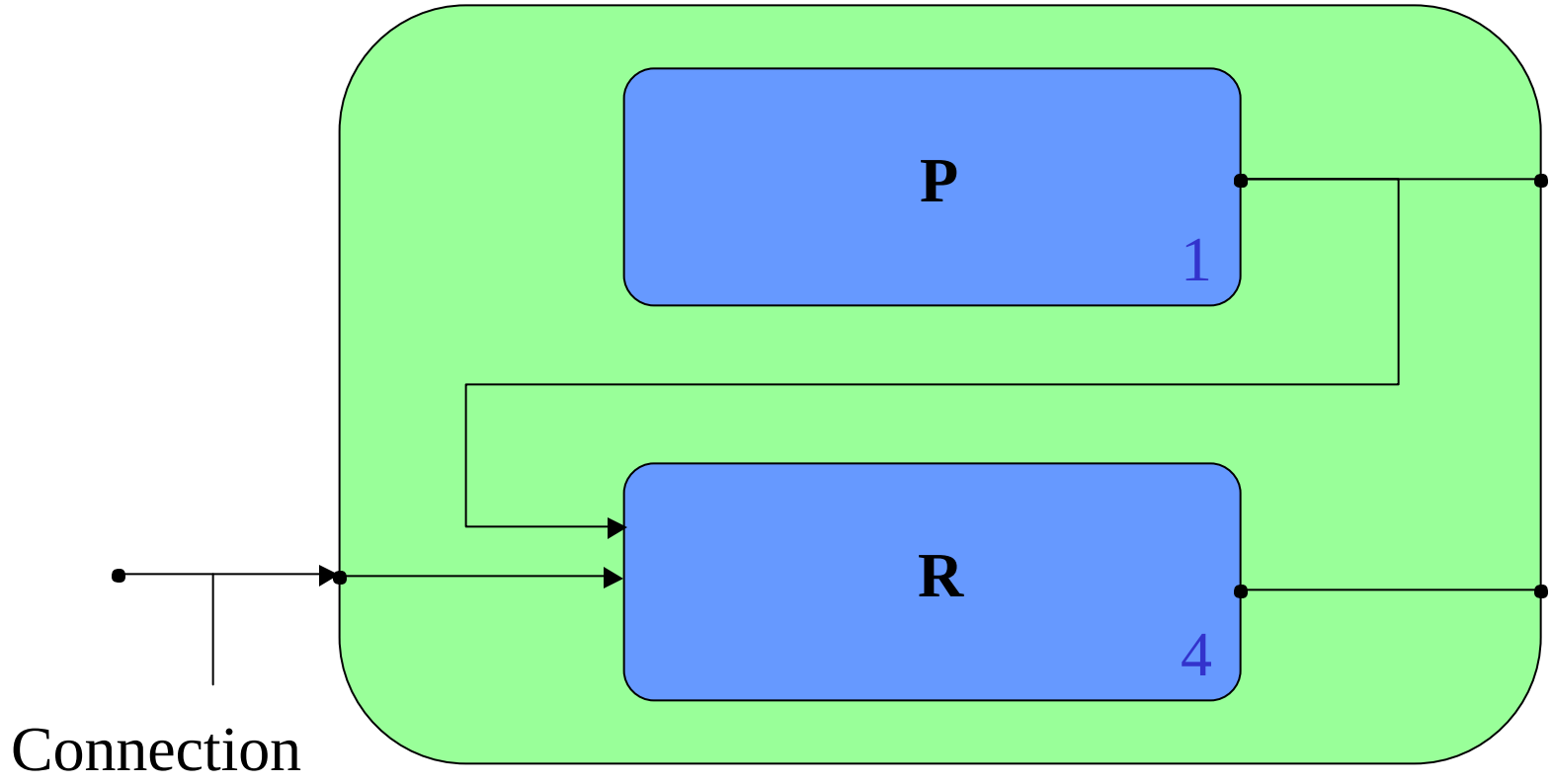
Mode M’



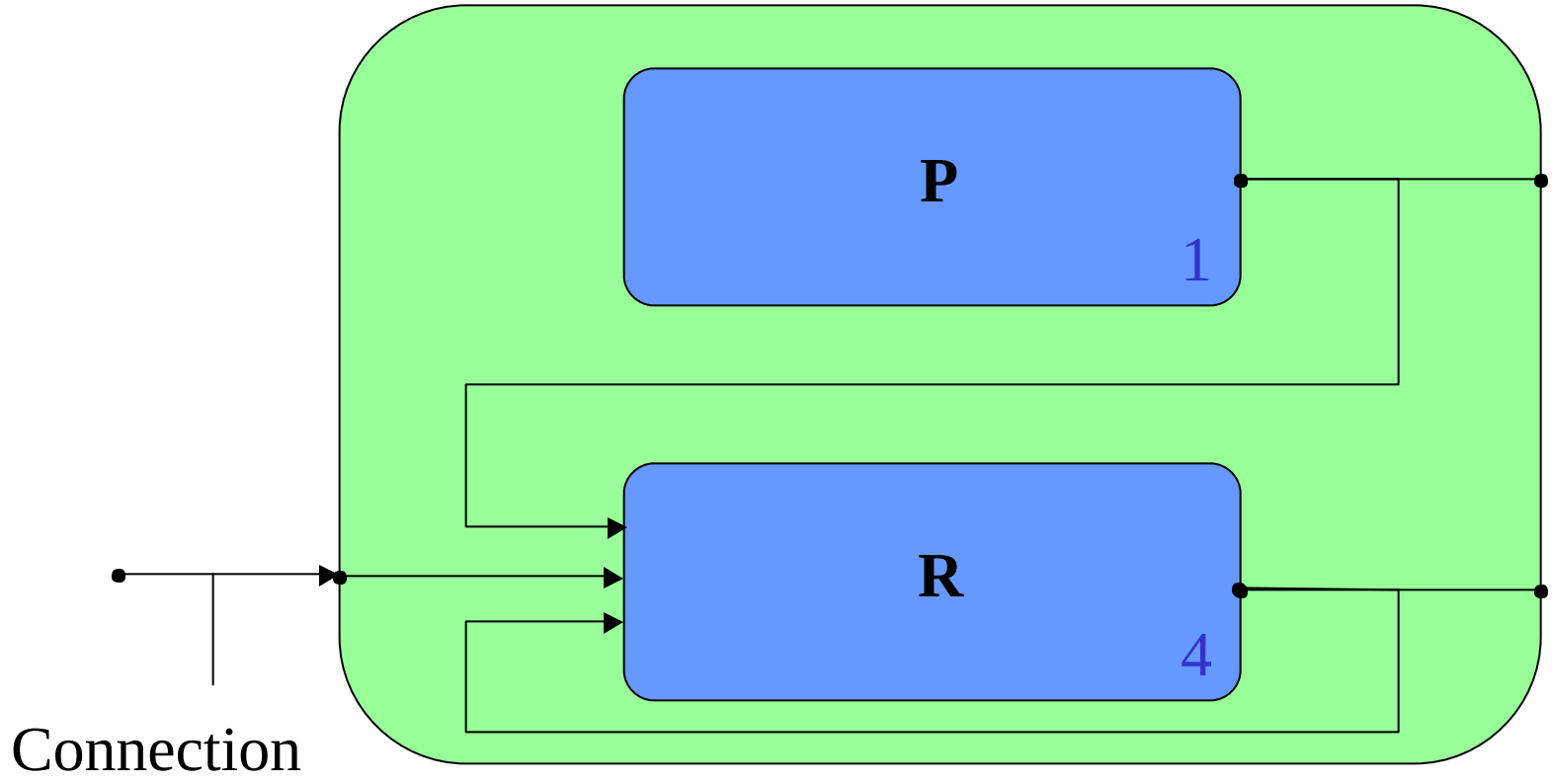
Mode M’



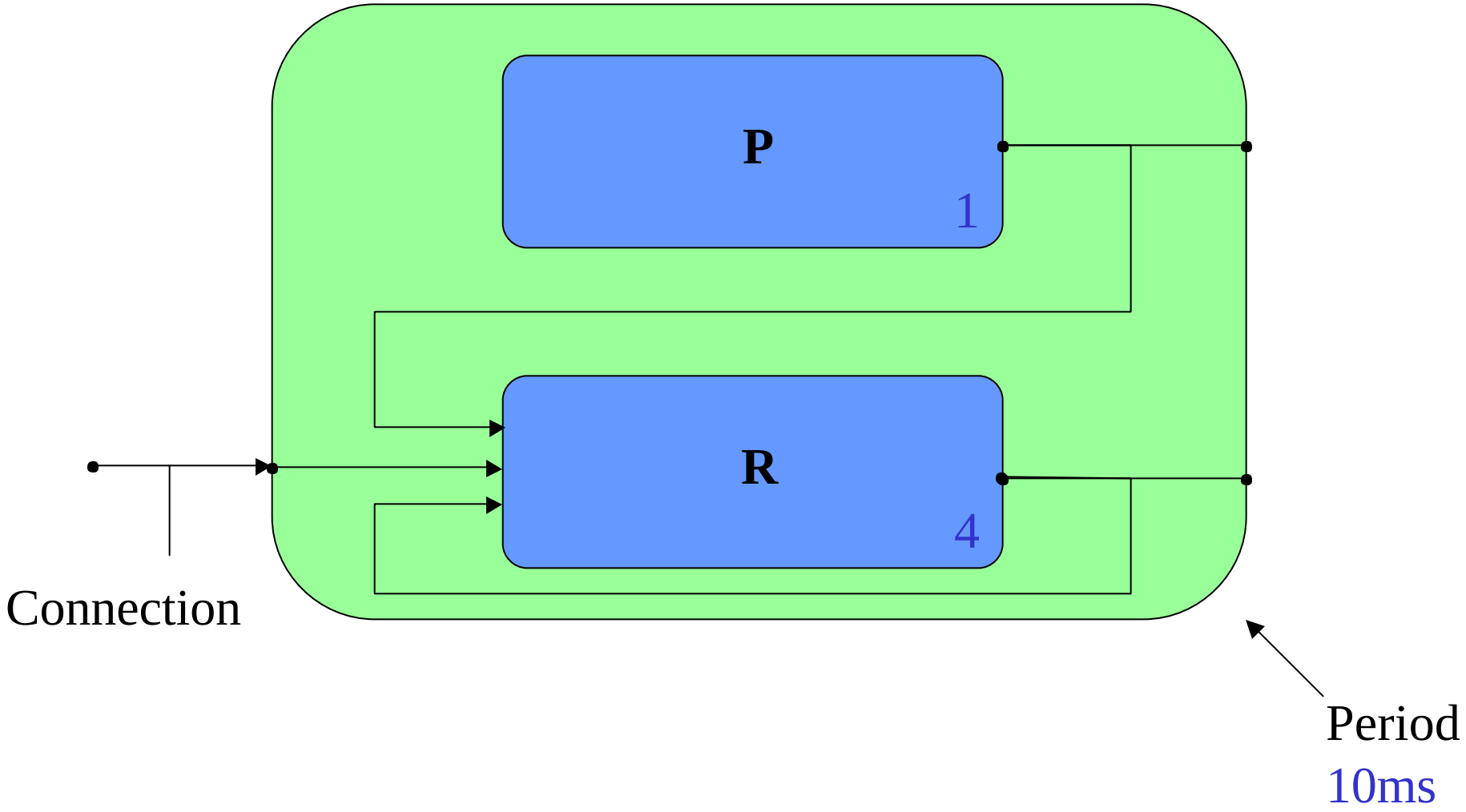
Mode M’



Mode M’



Mode M’

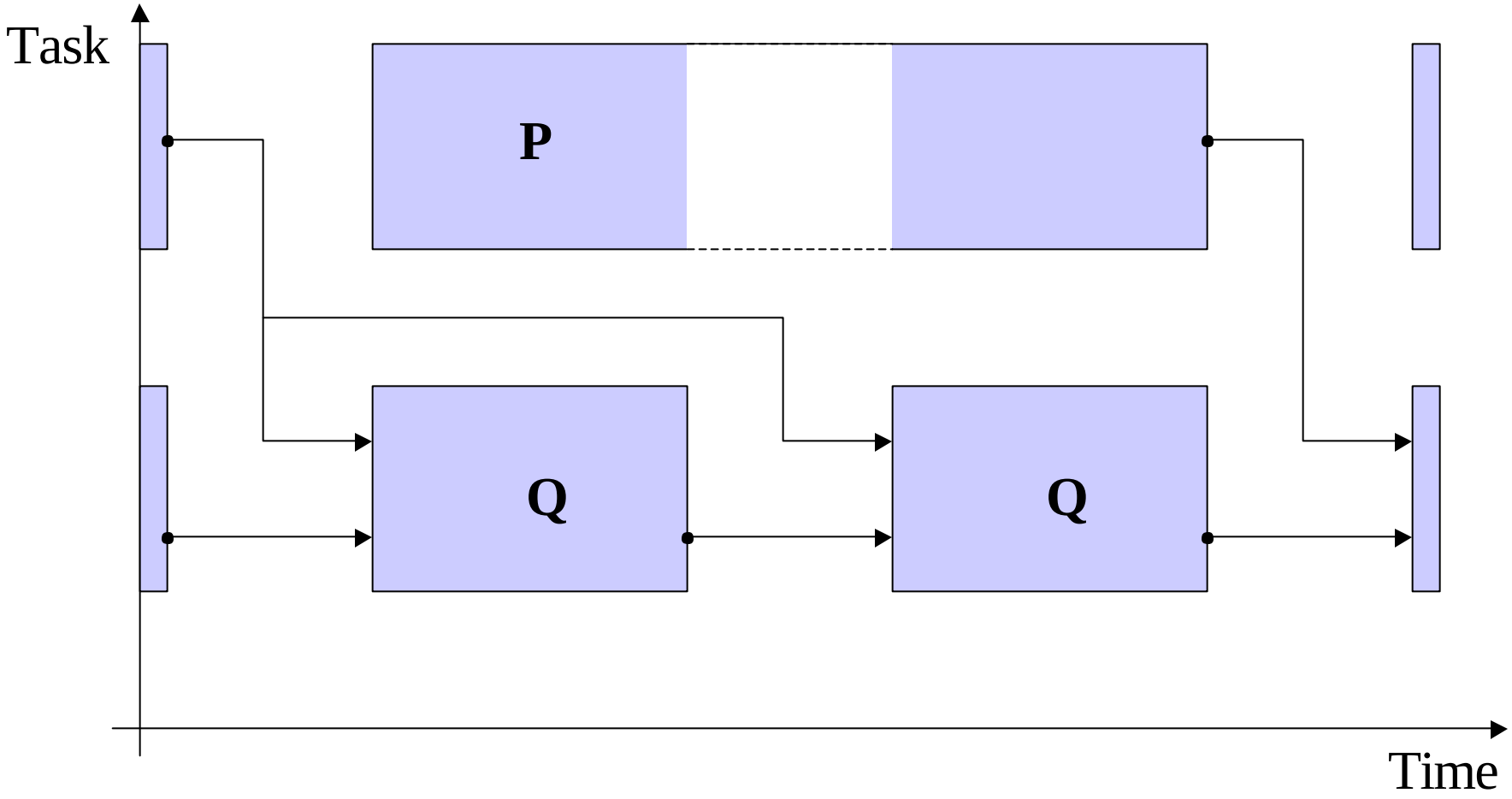


Concrete Syntax

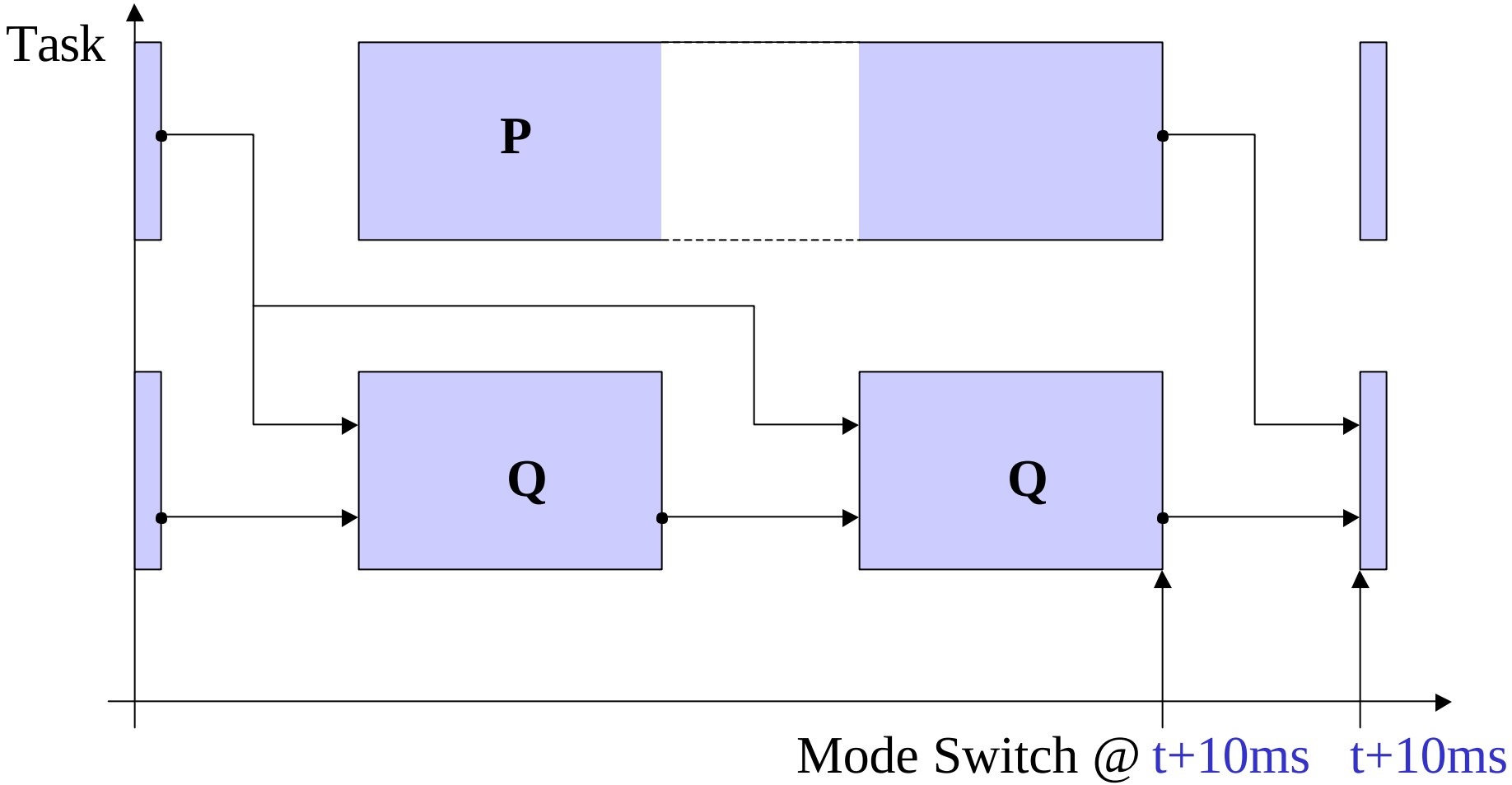
```
start m ( ) {
    mode m ( ) period 10 ms entryfreq 1 {
        taskfreq 1 do int x = P ( ) ;
        taskfreq 2 do int y = Q ( x, y ) ;

        exitfreq 2 if y = 5 then m' ( y ) ;
    }
    mode m' ( int z ) period 10 ms entryfreq 2 {
        taskfreq 1 do int x = P ( ) ;
        taskfreq 4 do int u = R ( x, z ) ;
    }
}
```

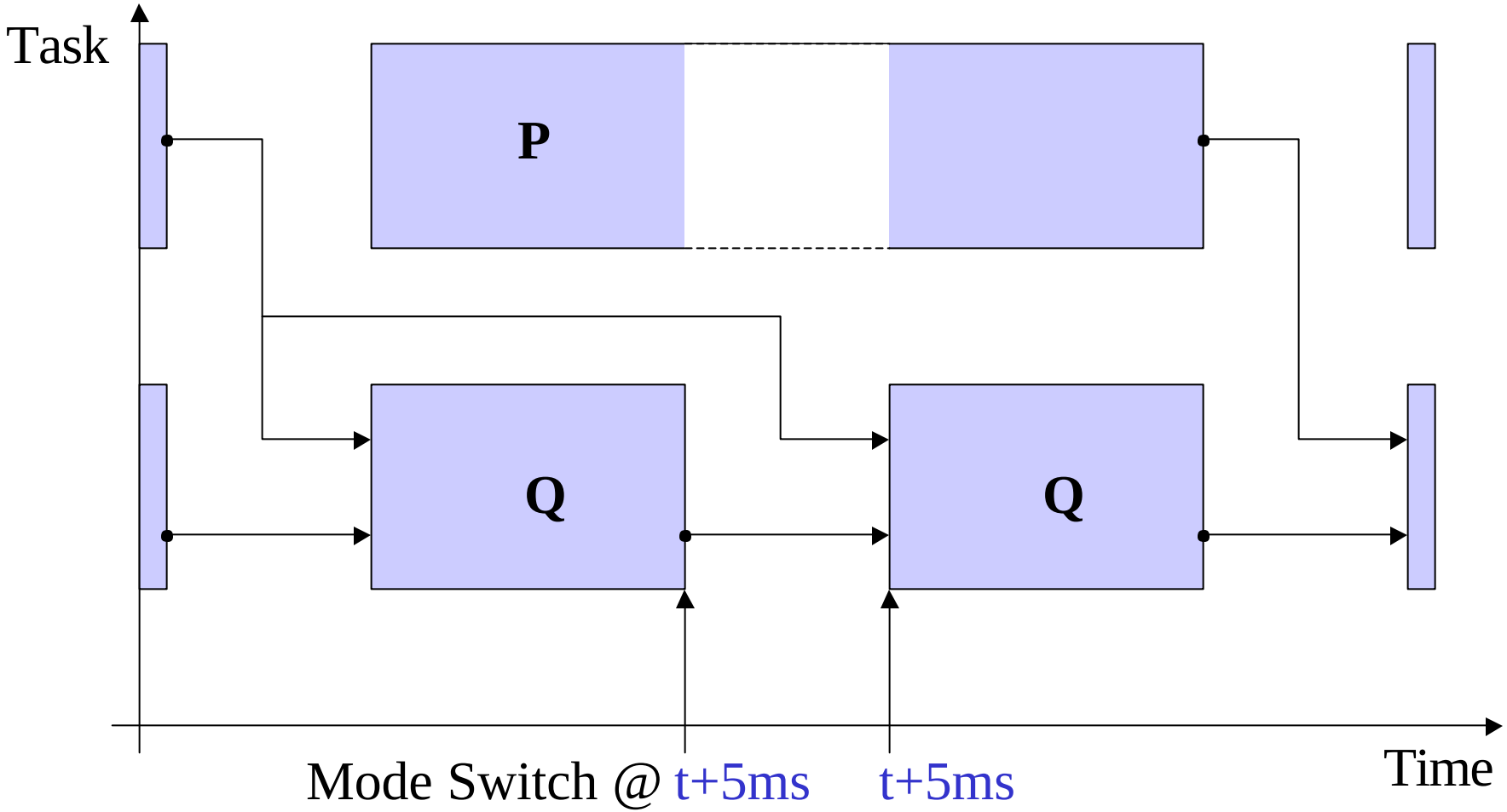
Semantics of a Mode Switch



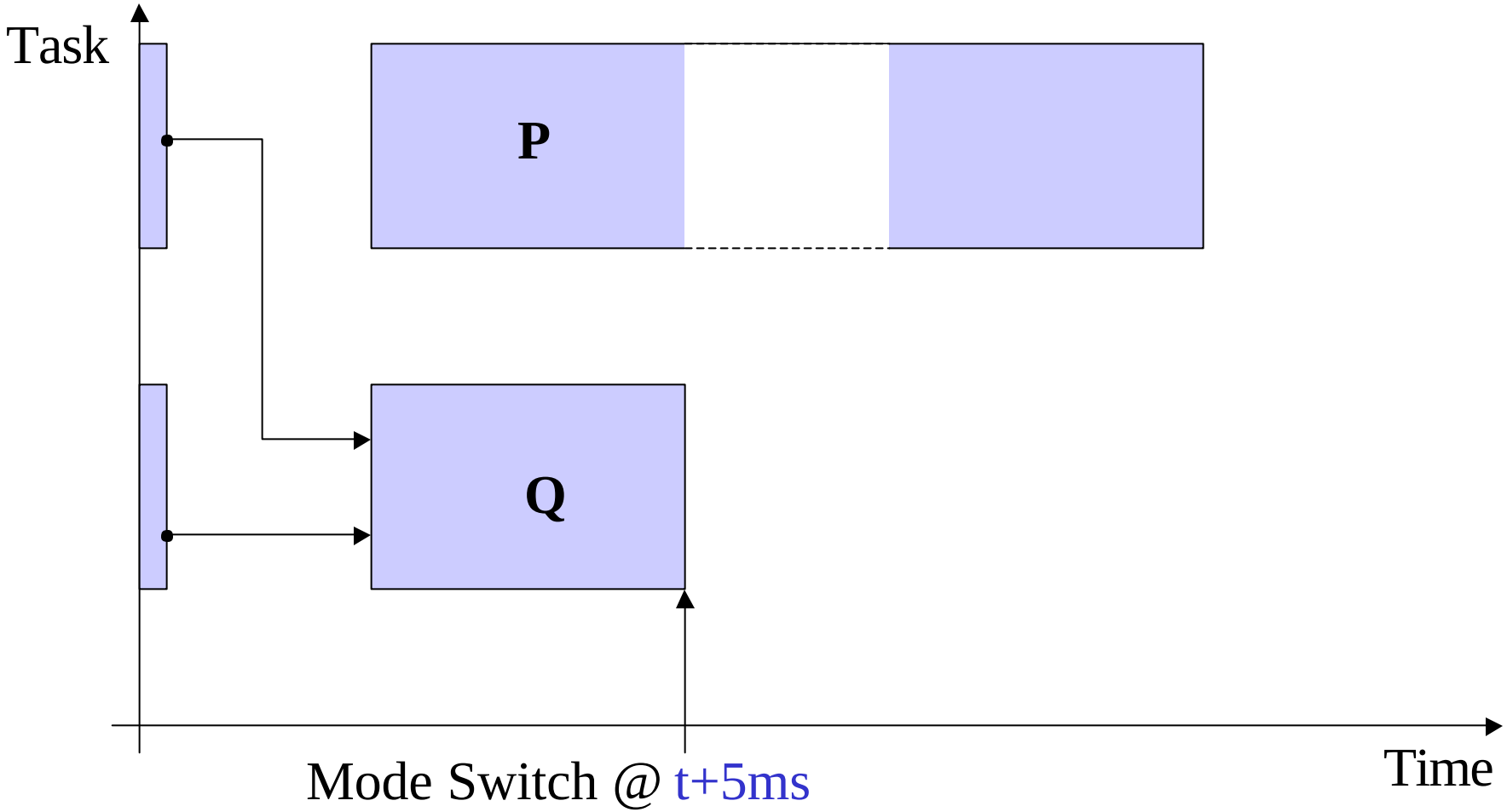
Semantics of a Mode Switch



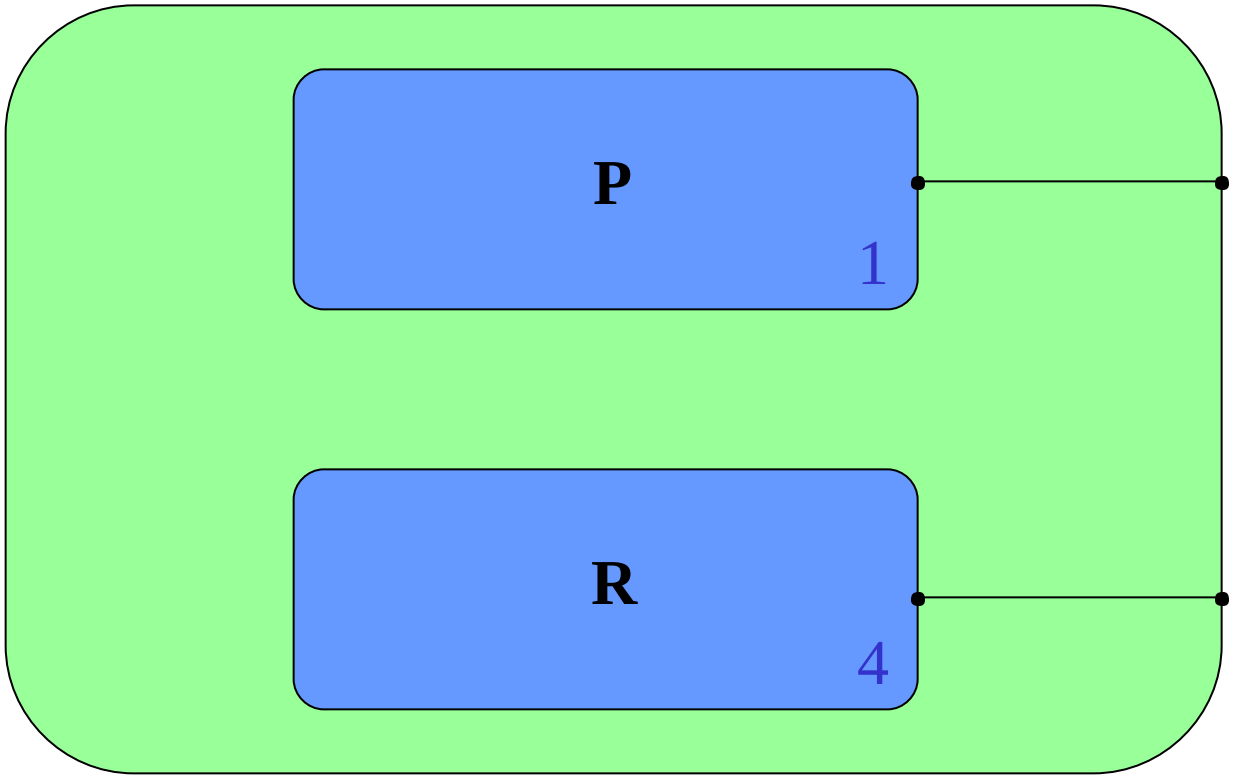
Semantics of a Mode Switch



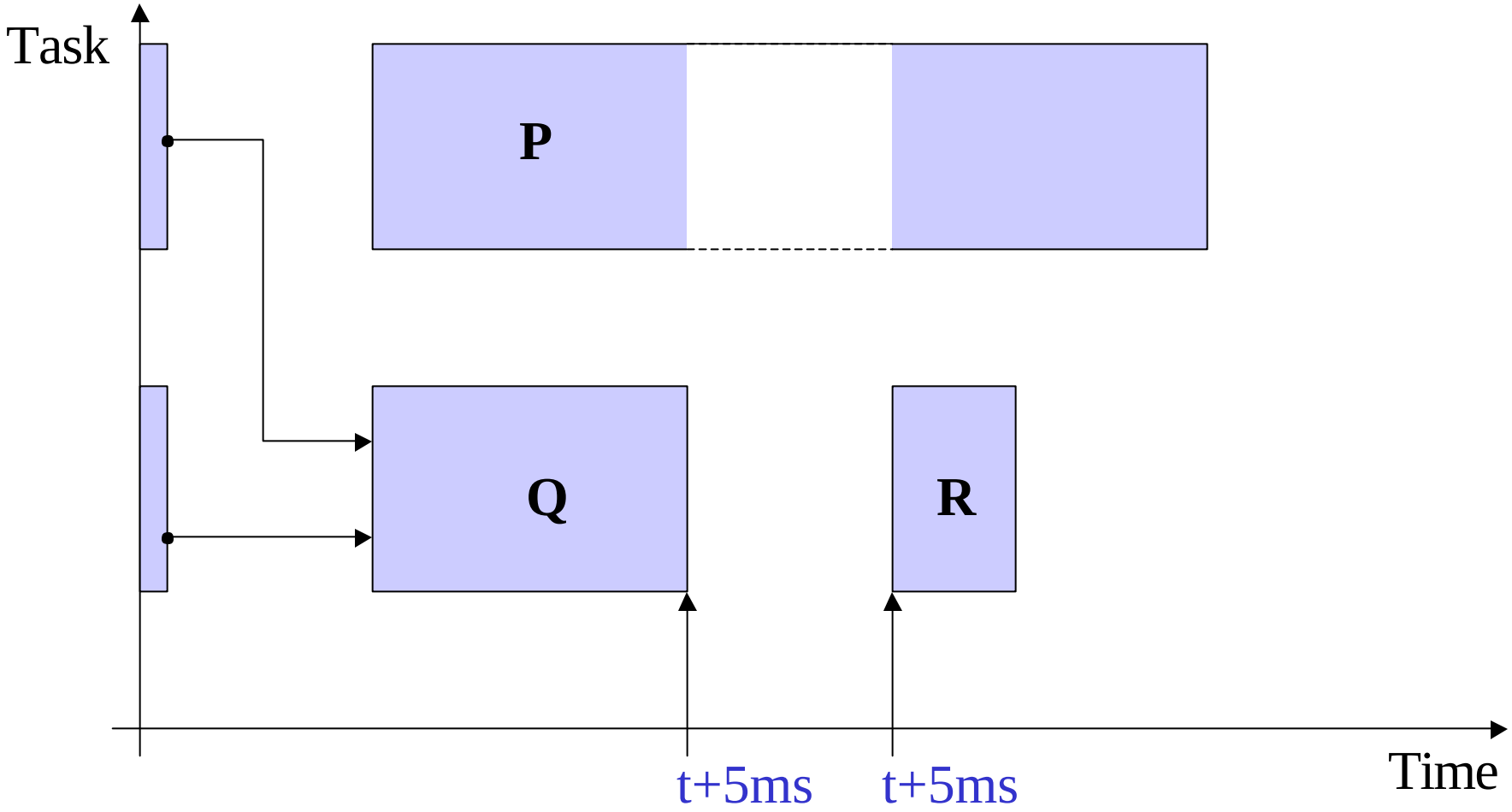
Semantics of a Mode Switch



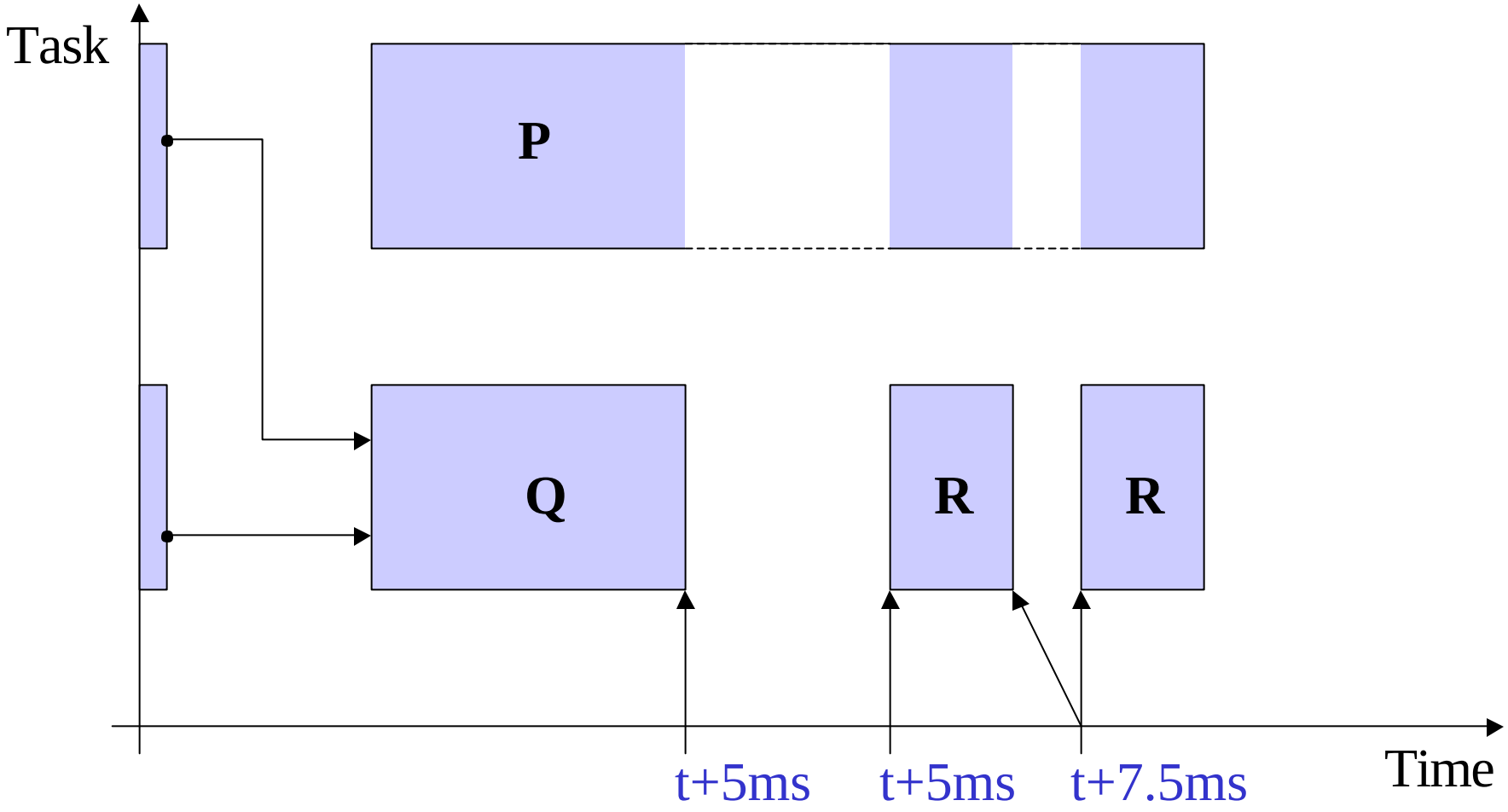
Mode M’



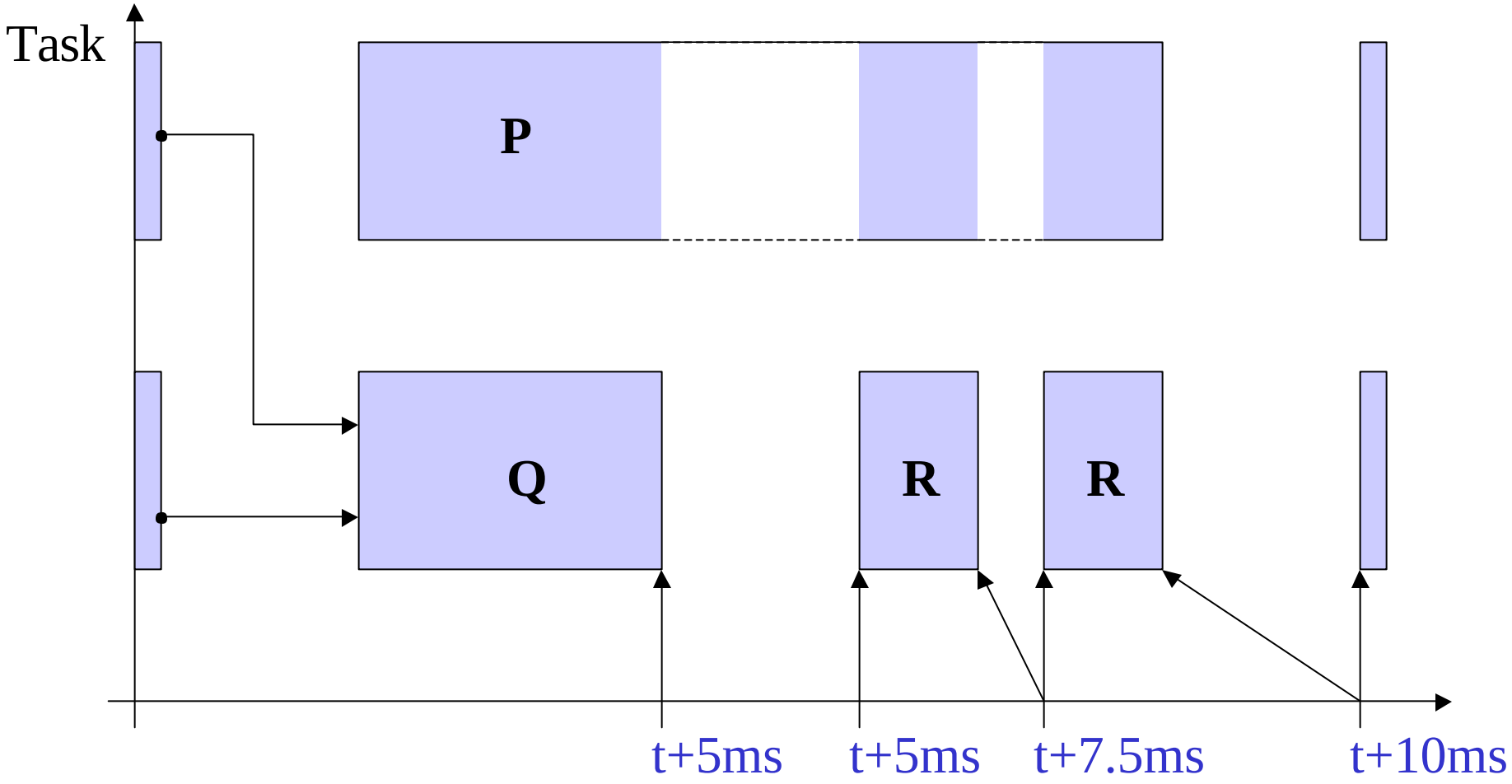
Semantics of a Mode Switch



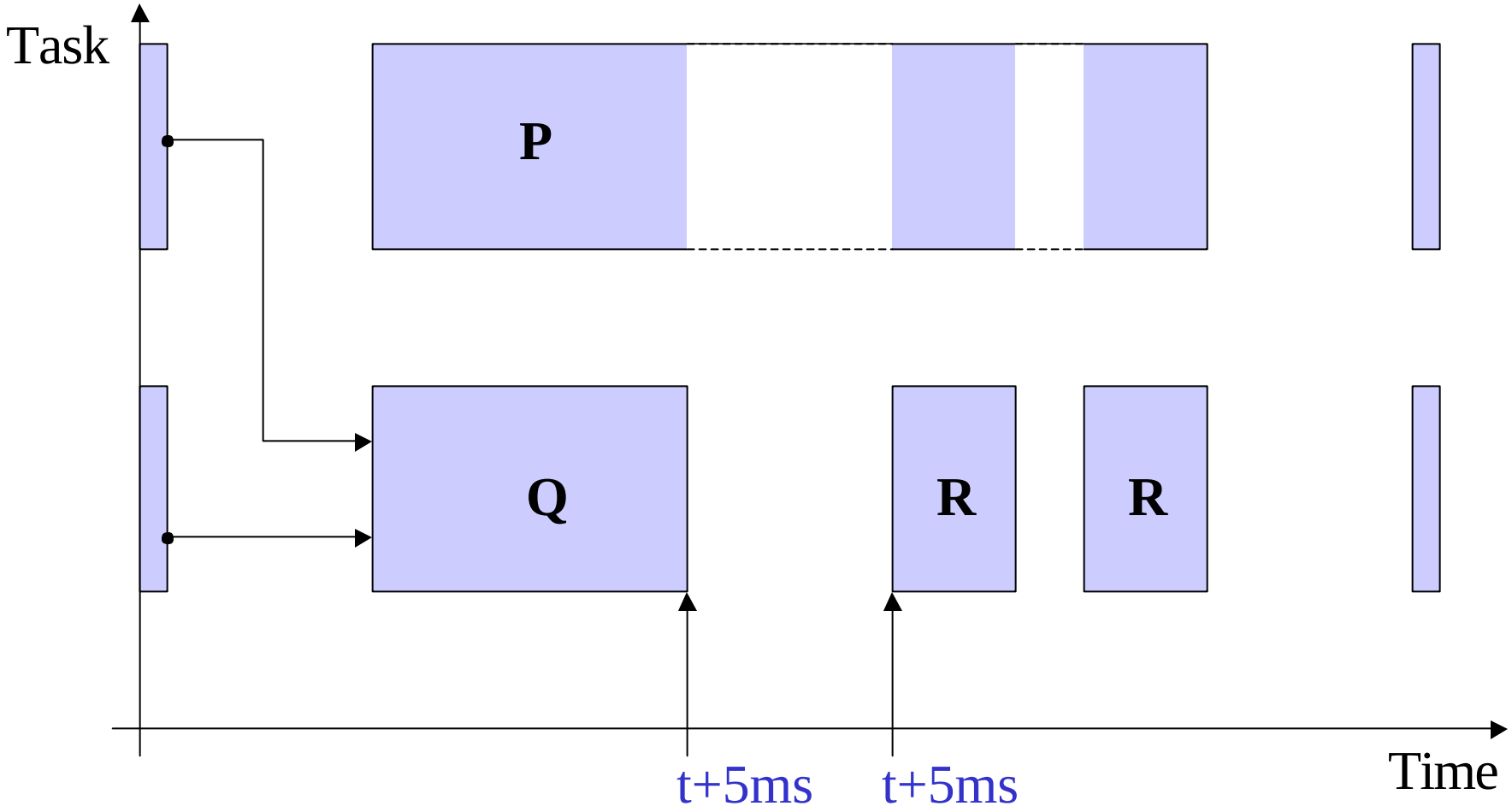
Semantics of a Mode Switch



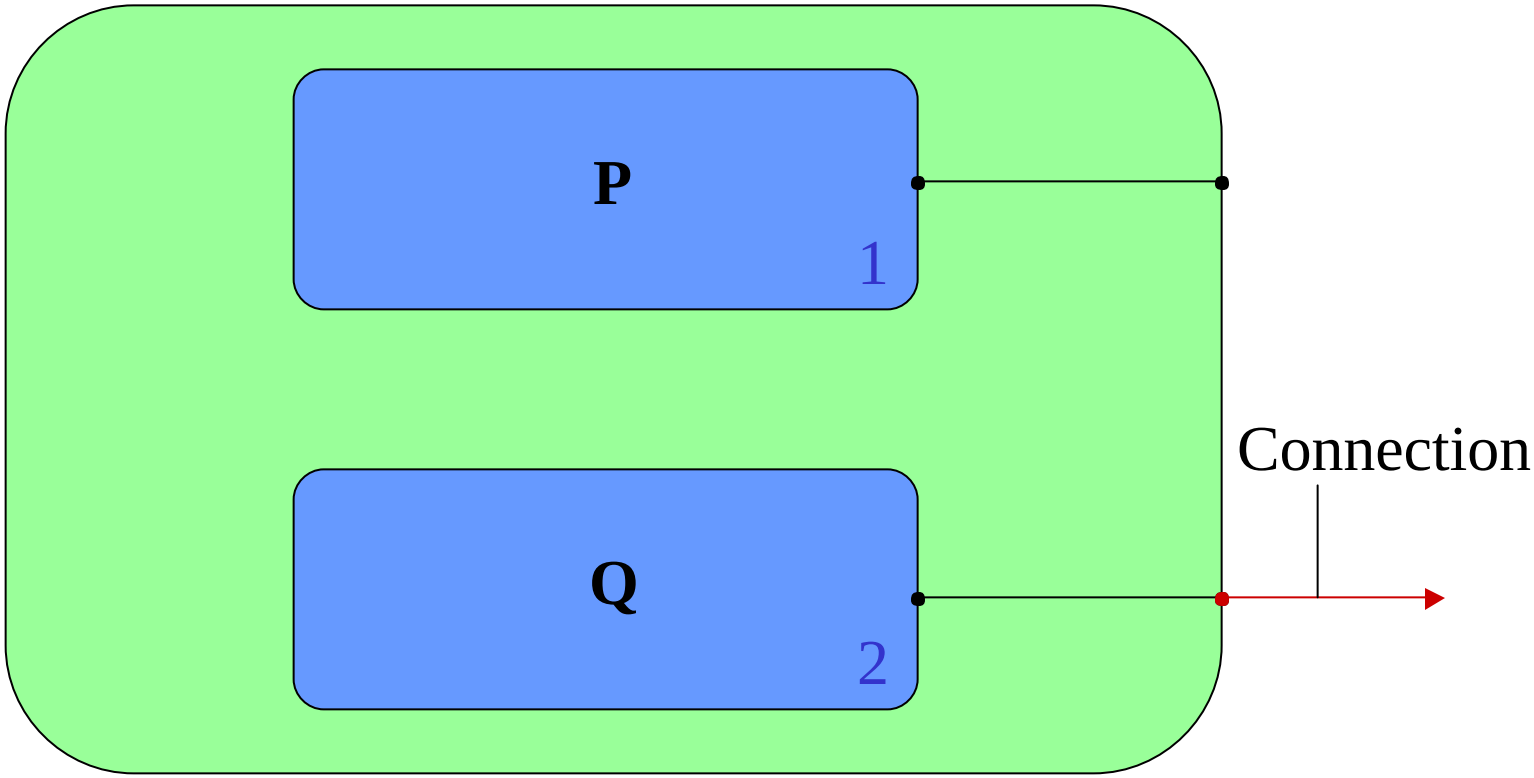
Semantics of a Mode Switch



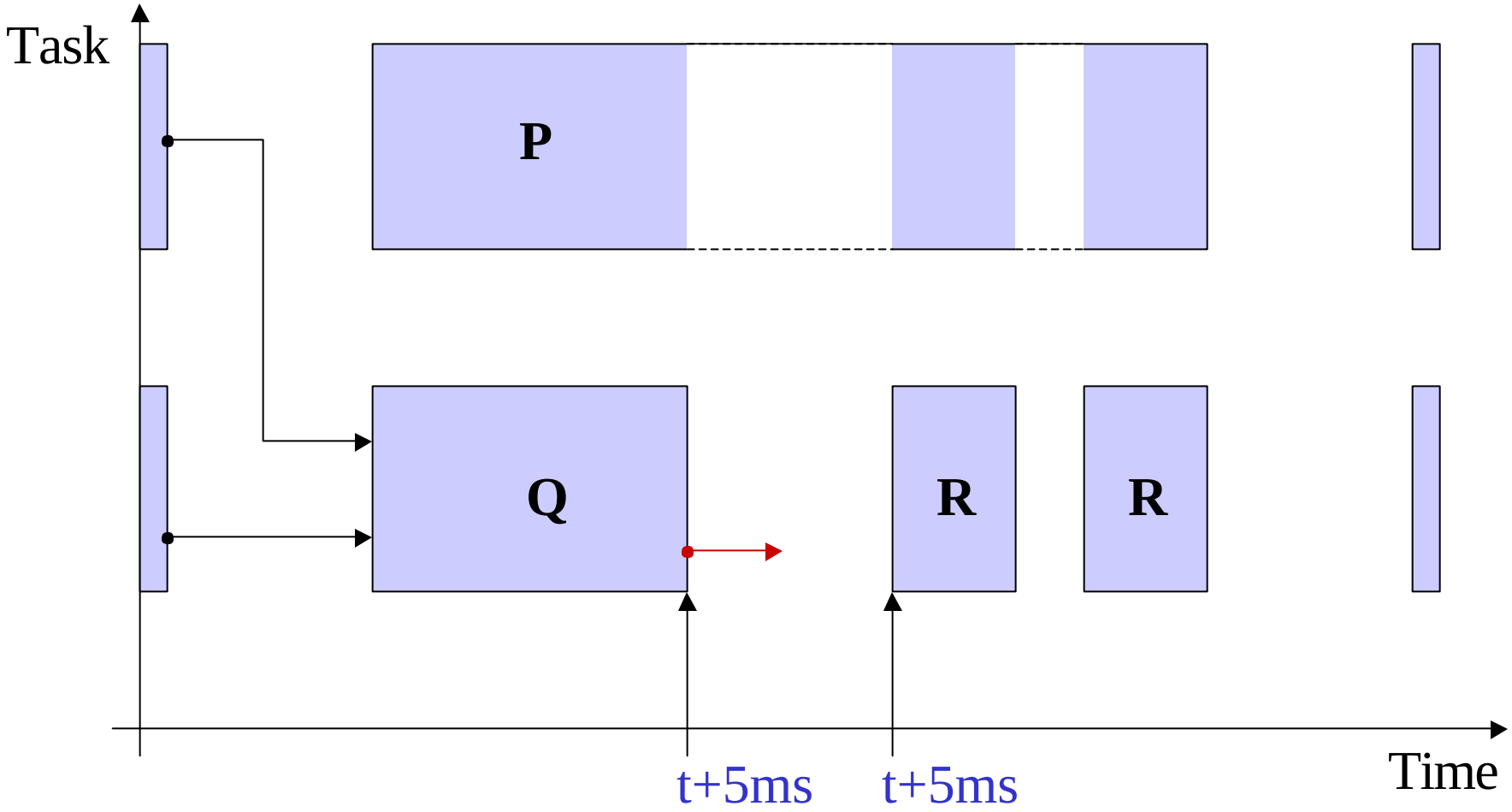
Semantics of a Mode Switch



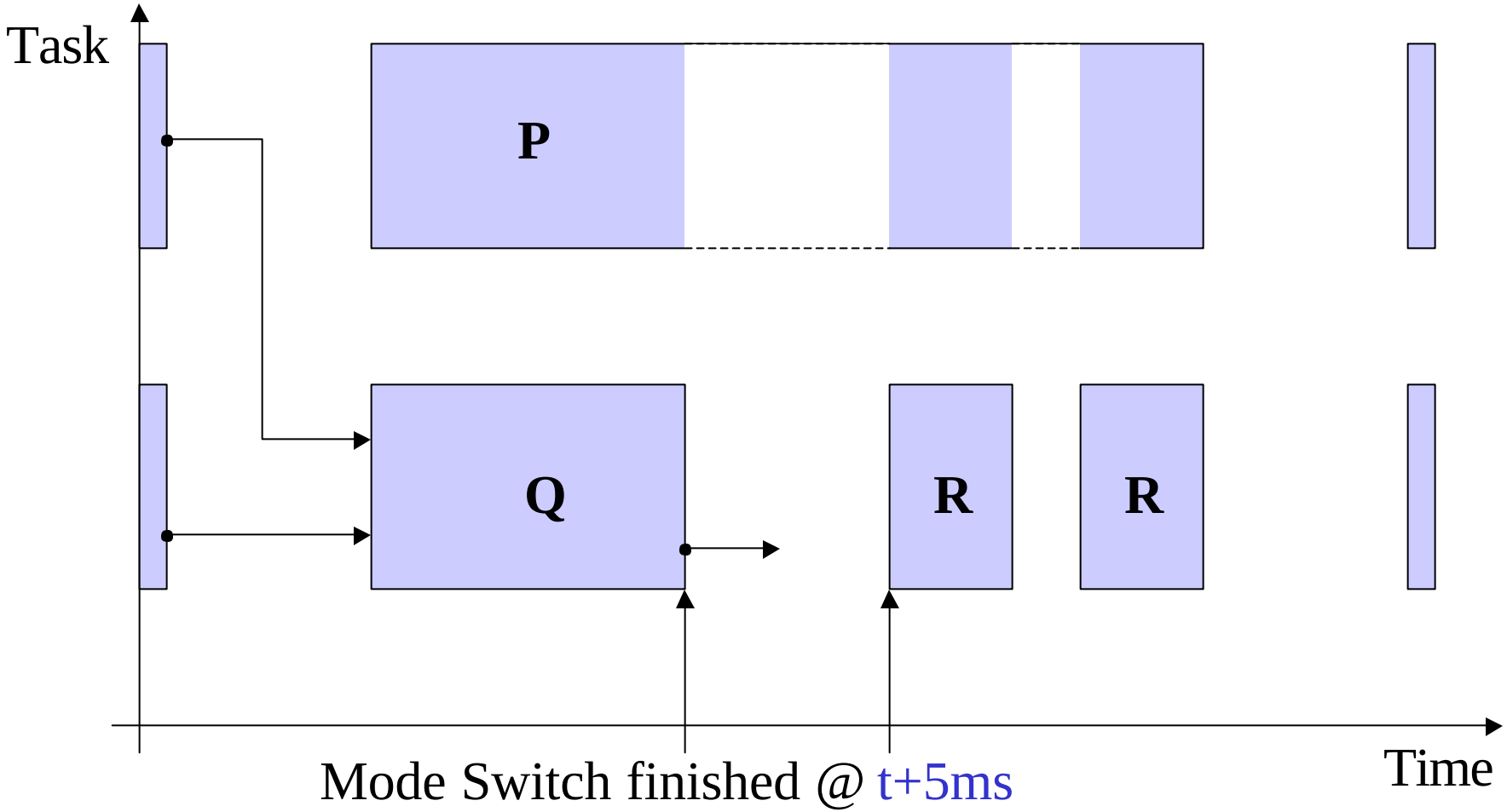
Mode M



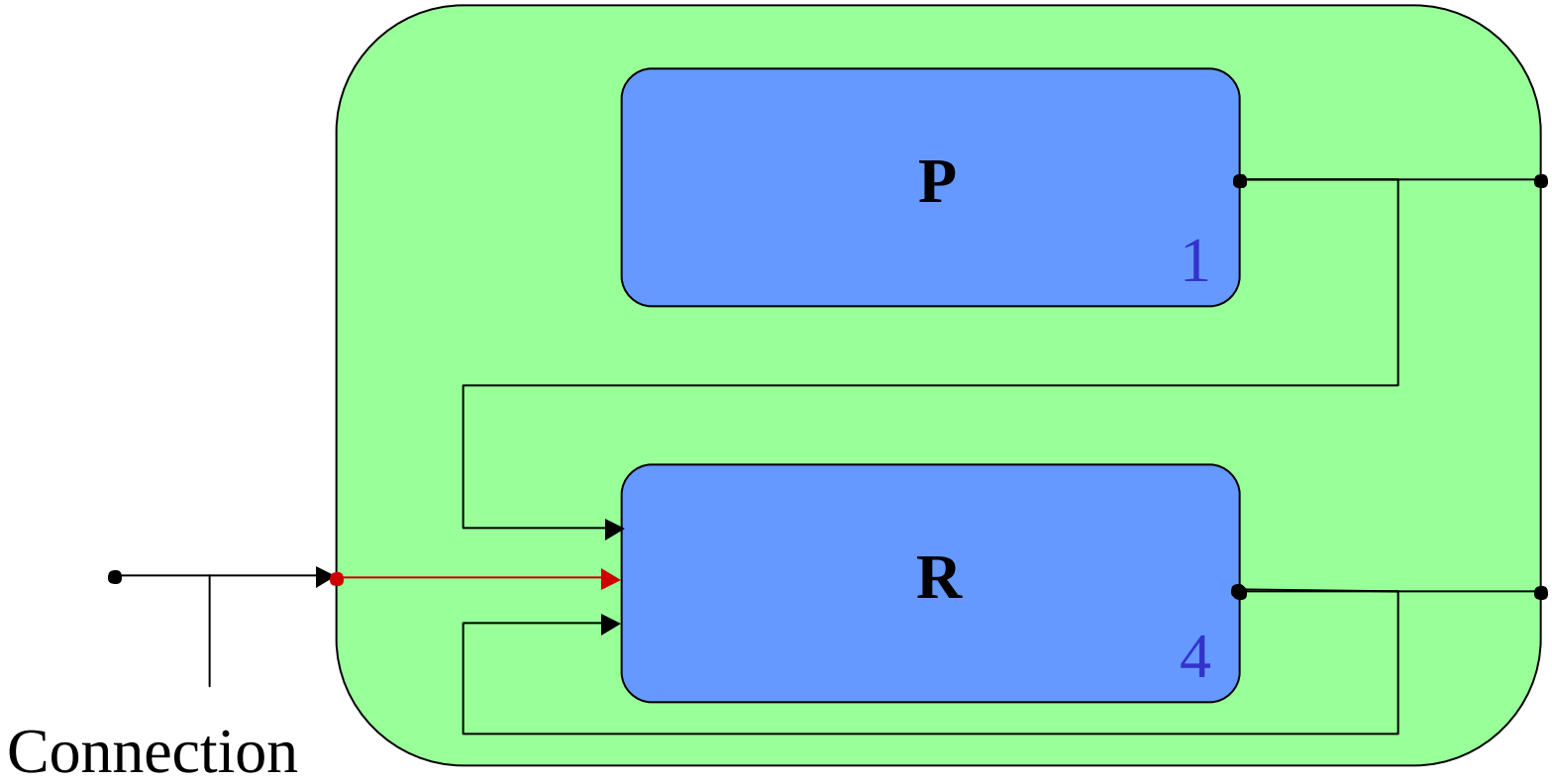
Semantics of a Mode Switch



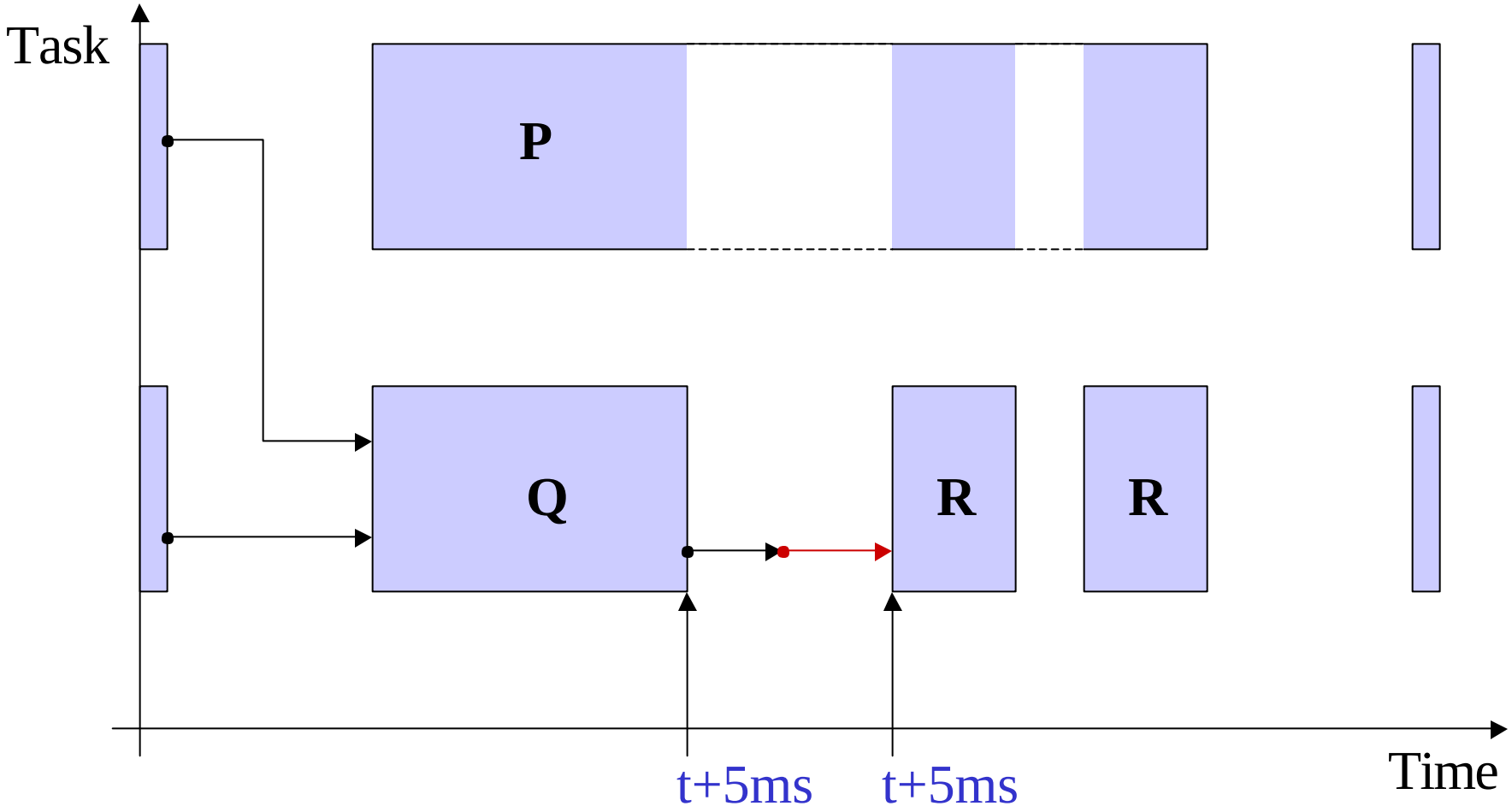
Semantics of a Mode Switch



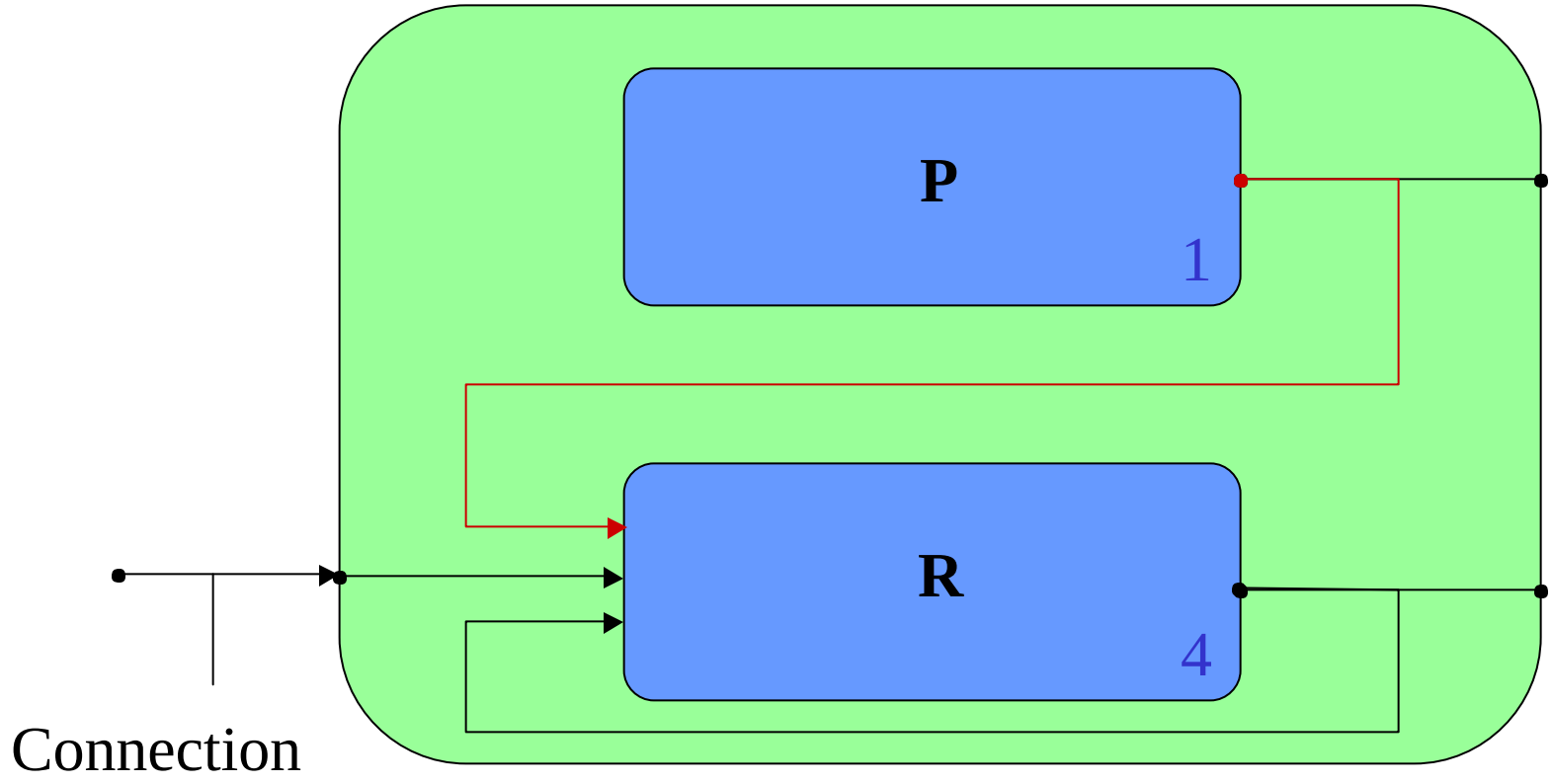
Mode M’



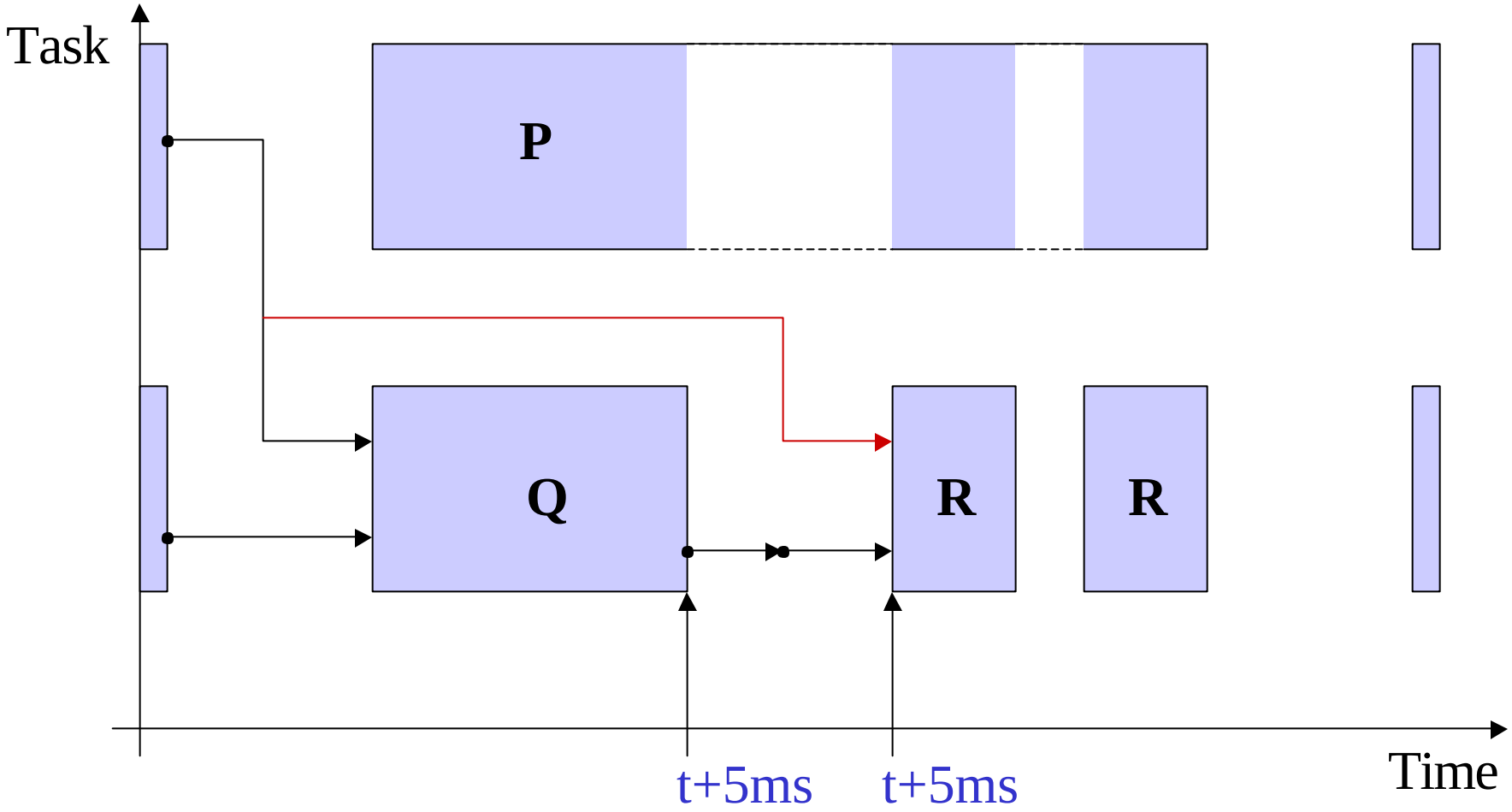
Semantics of a Mode Switch



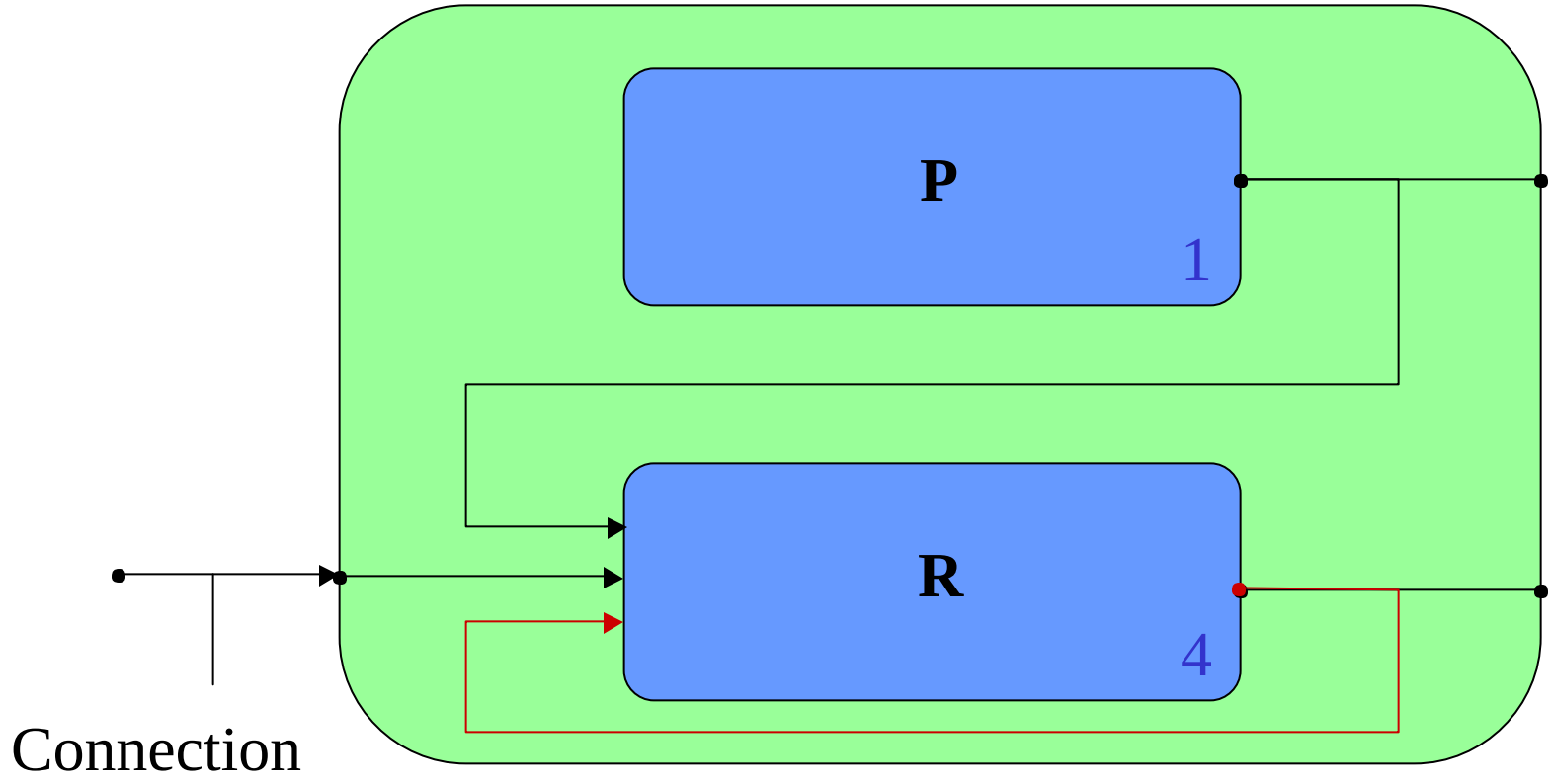
Mode M’



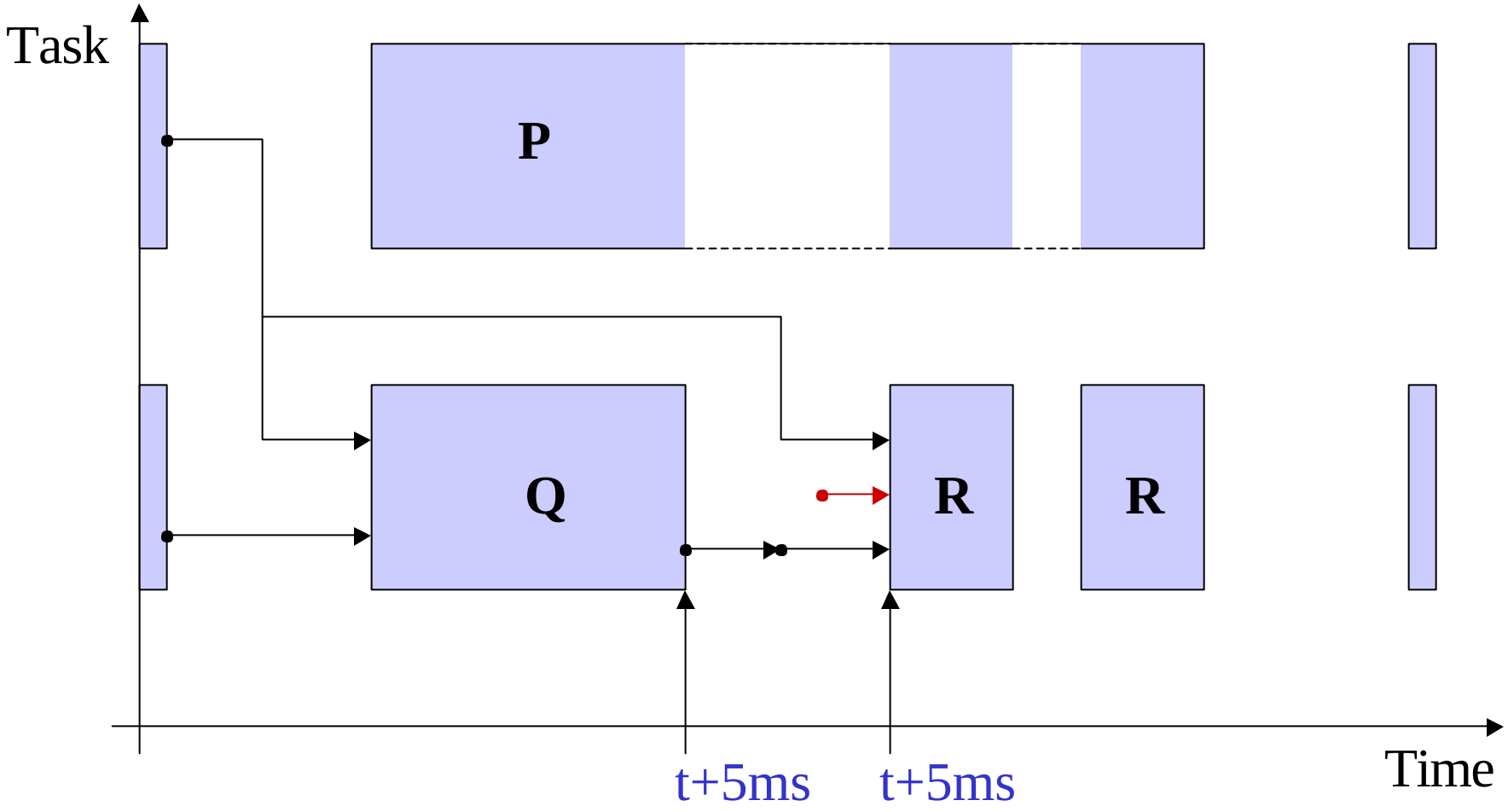
Semantics of a Mode Switch



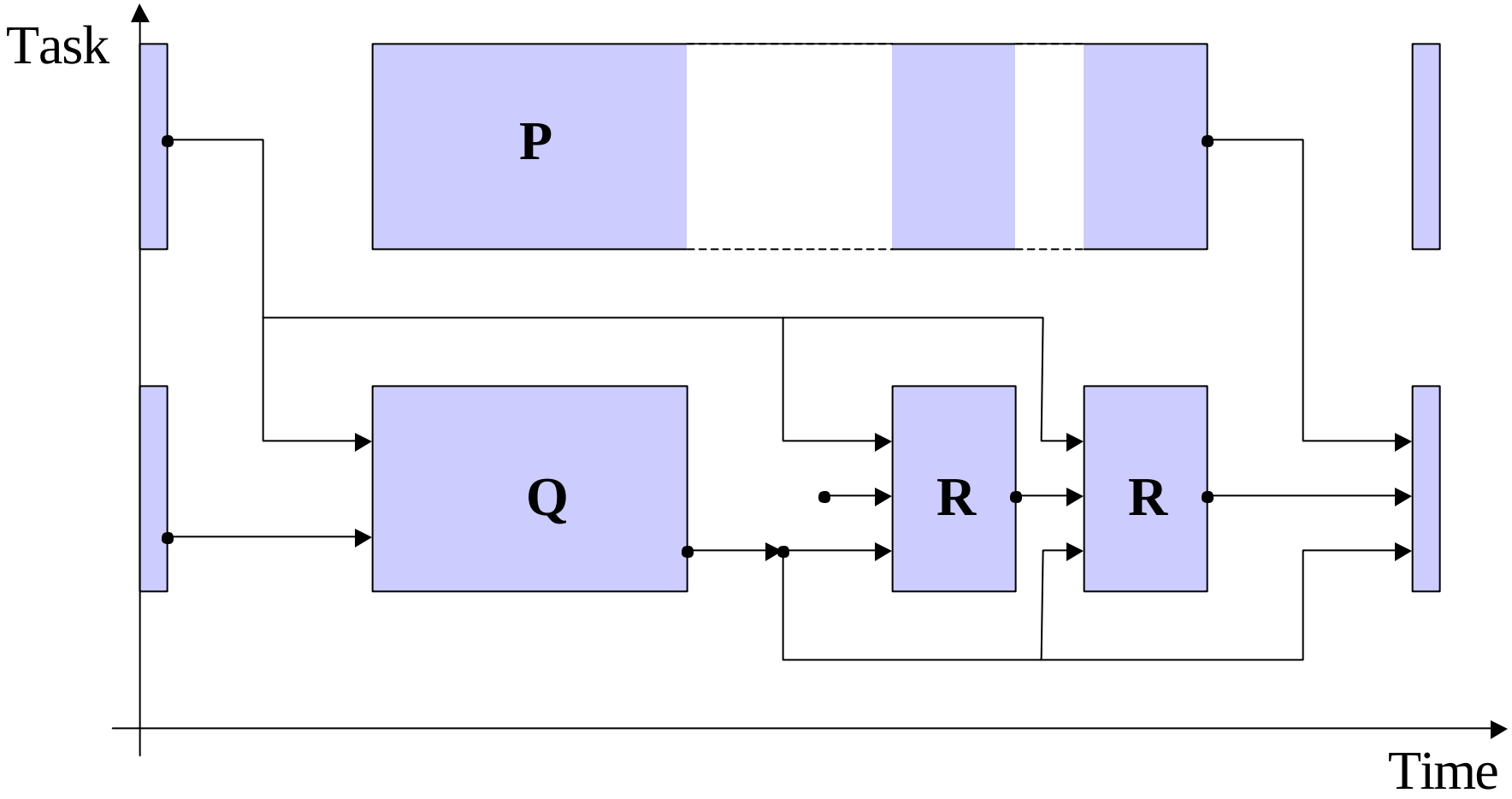
Mode M’



Semantics of a Mode Switch



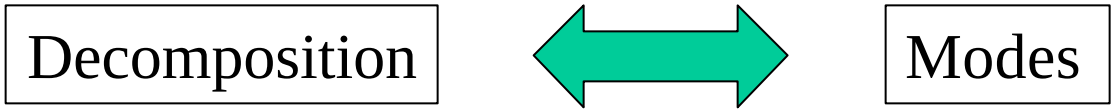
Semantics of a Mode Switch



Decomposition: Giotto Modes

Preserving Structure

Our Approach

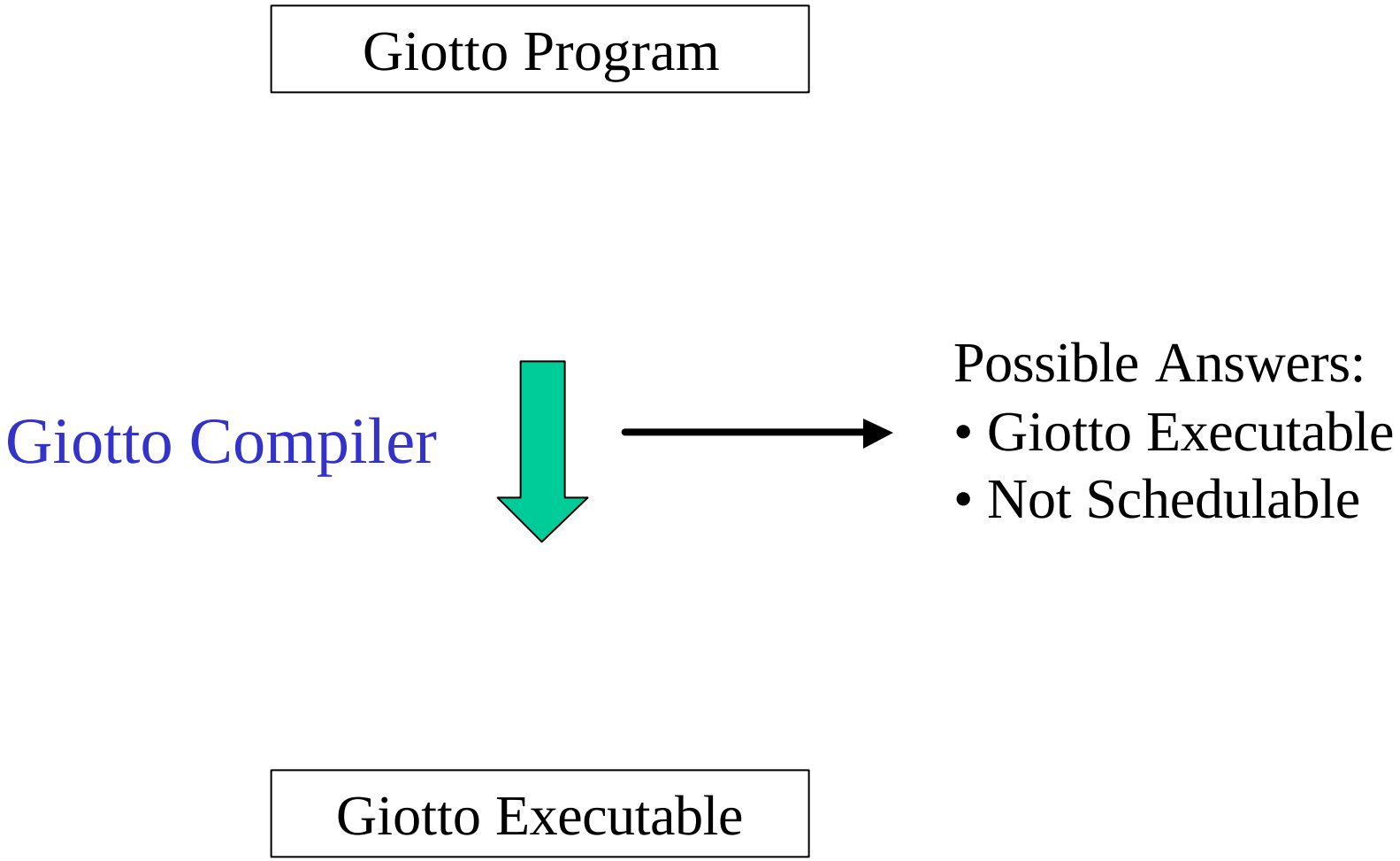


Compilation

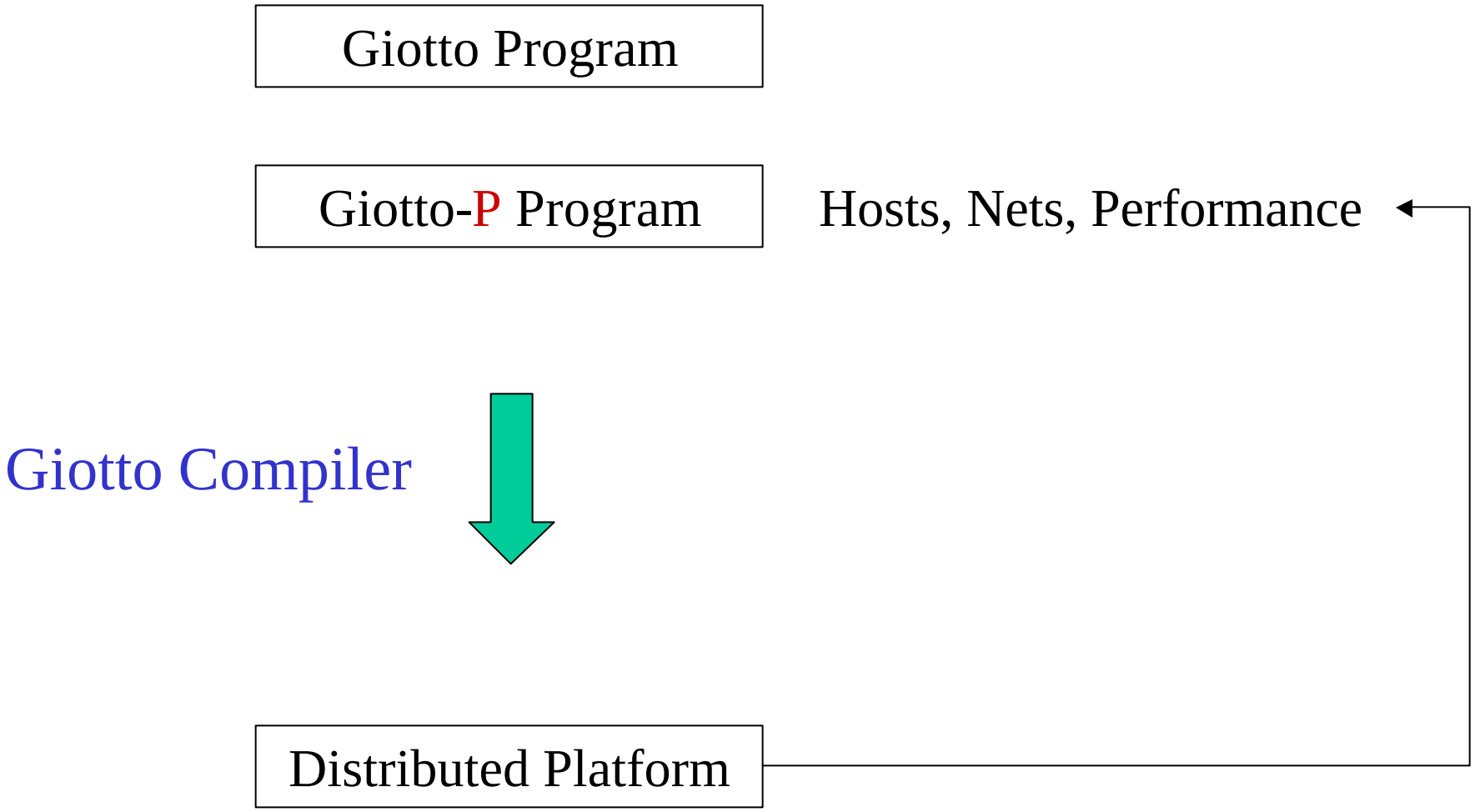
The Giotto Compiler:

- Automatic code generation
- Stepwise program refinement with annotated Giotto for distributed platforms

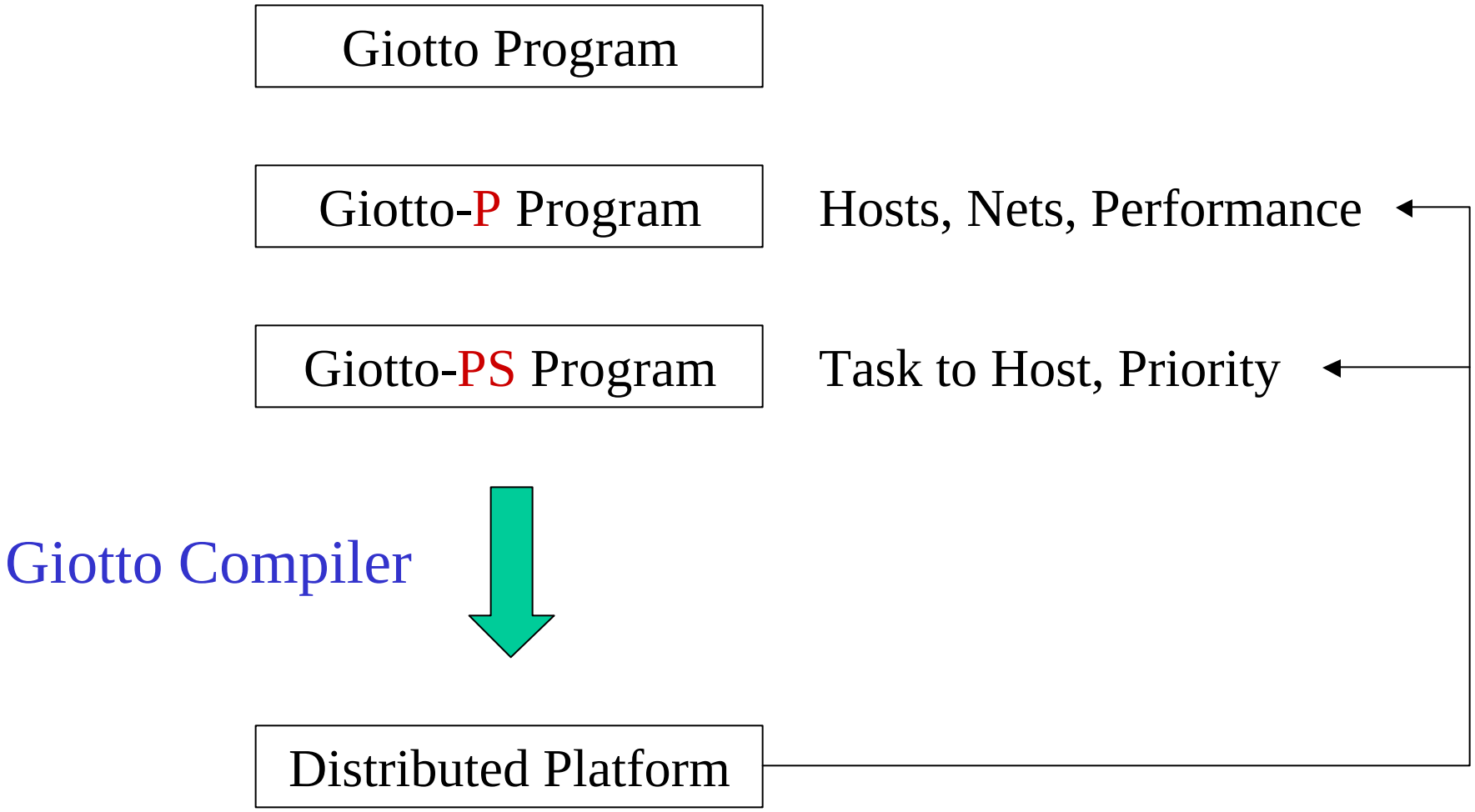
The Giotto Compiler



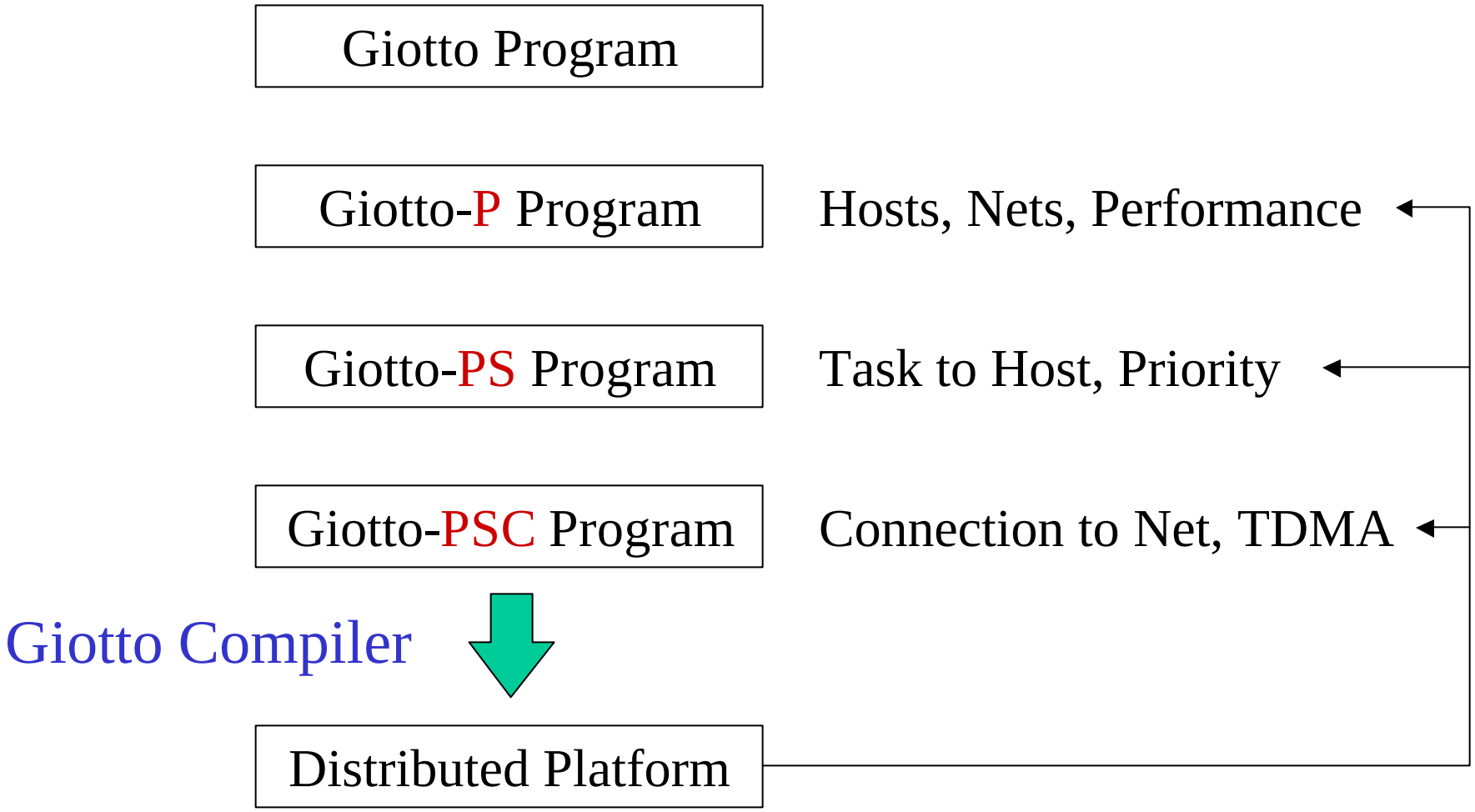
Giotto-Platform



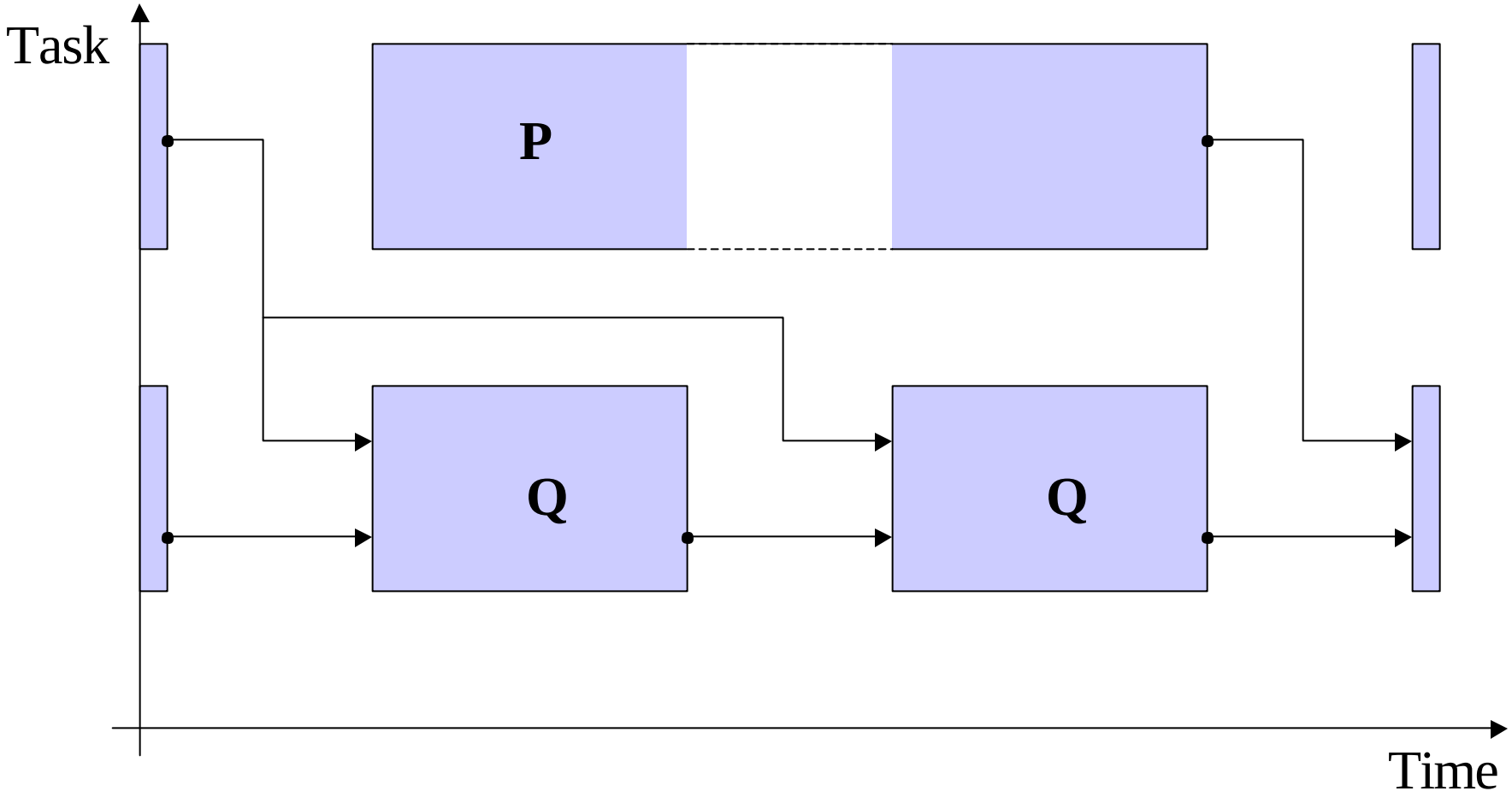
Giotto-Scheduler



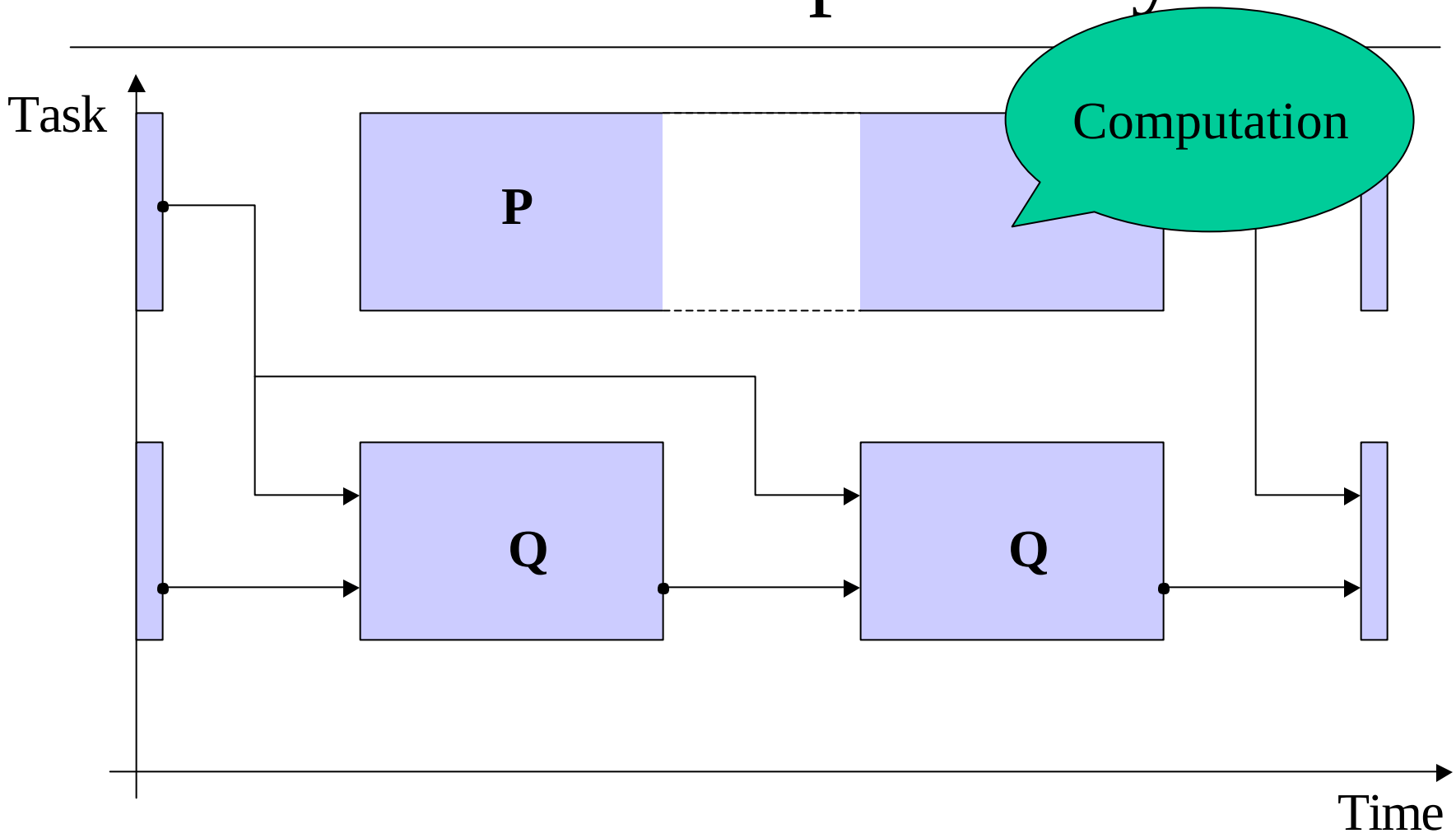
Giotto-Communication



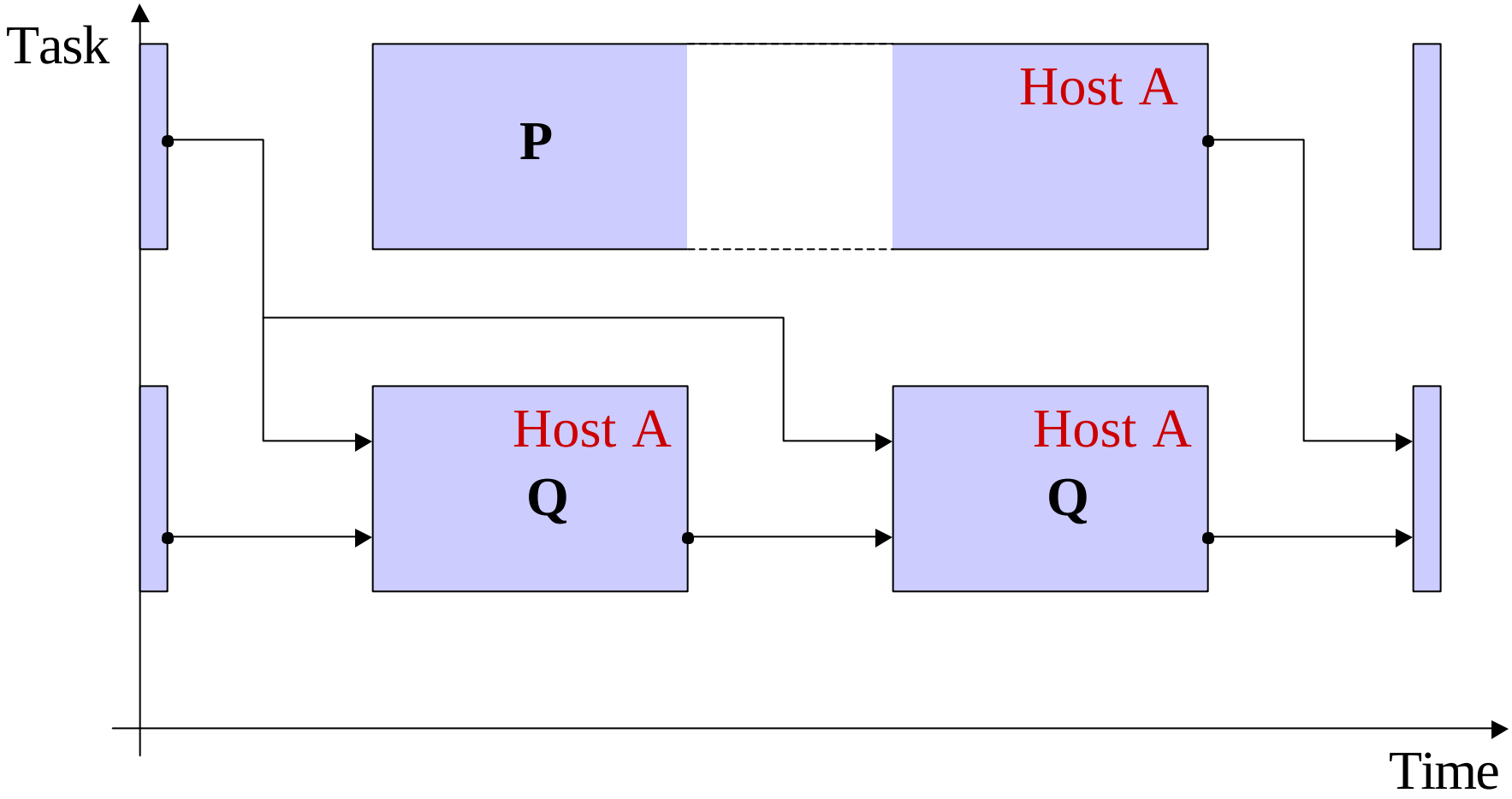
Platform Dependency



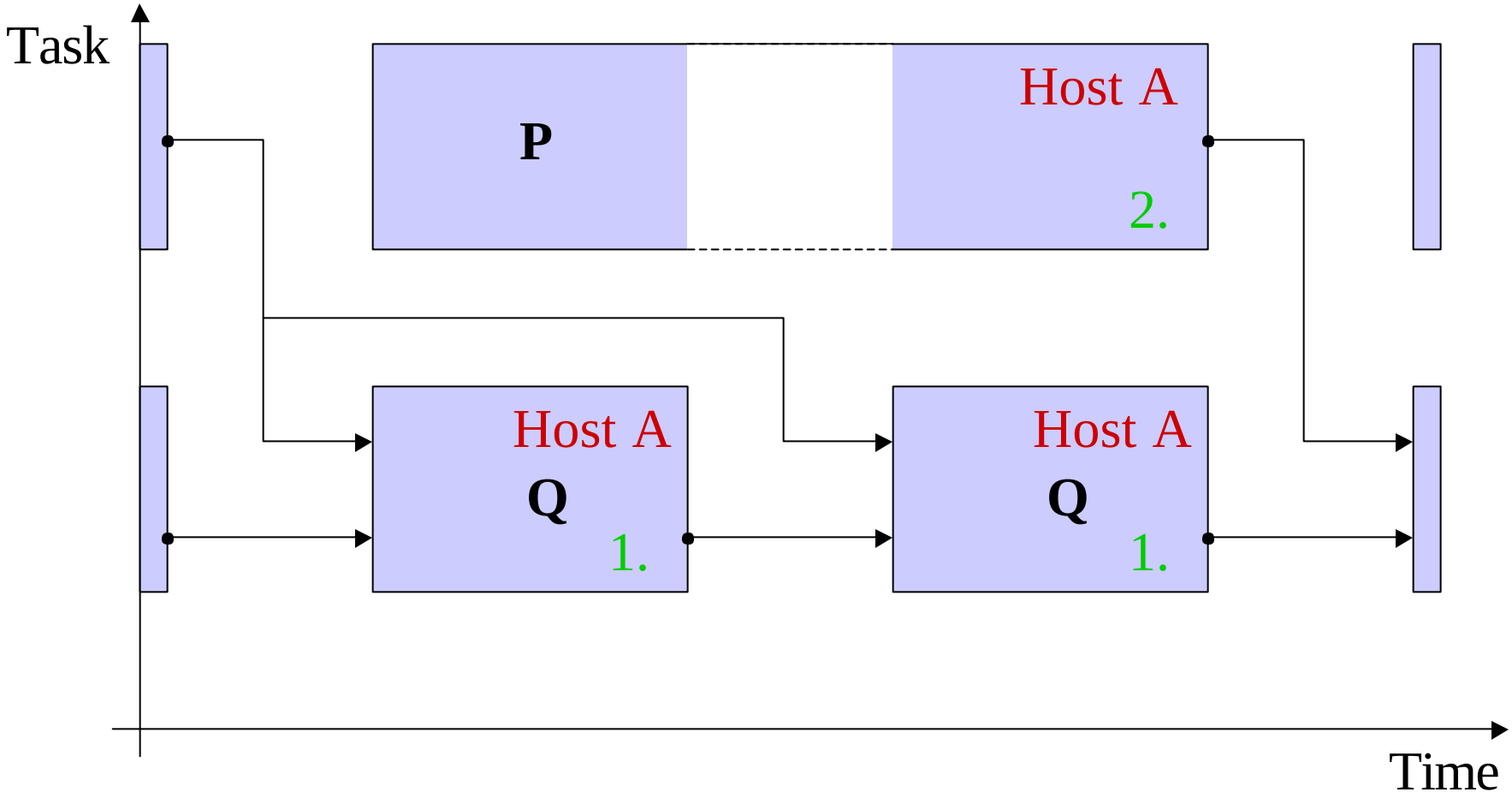
Platform Dependency



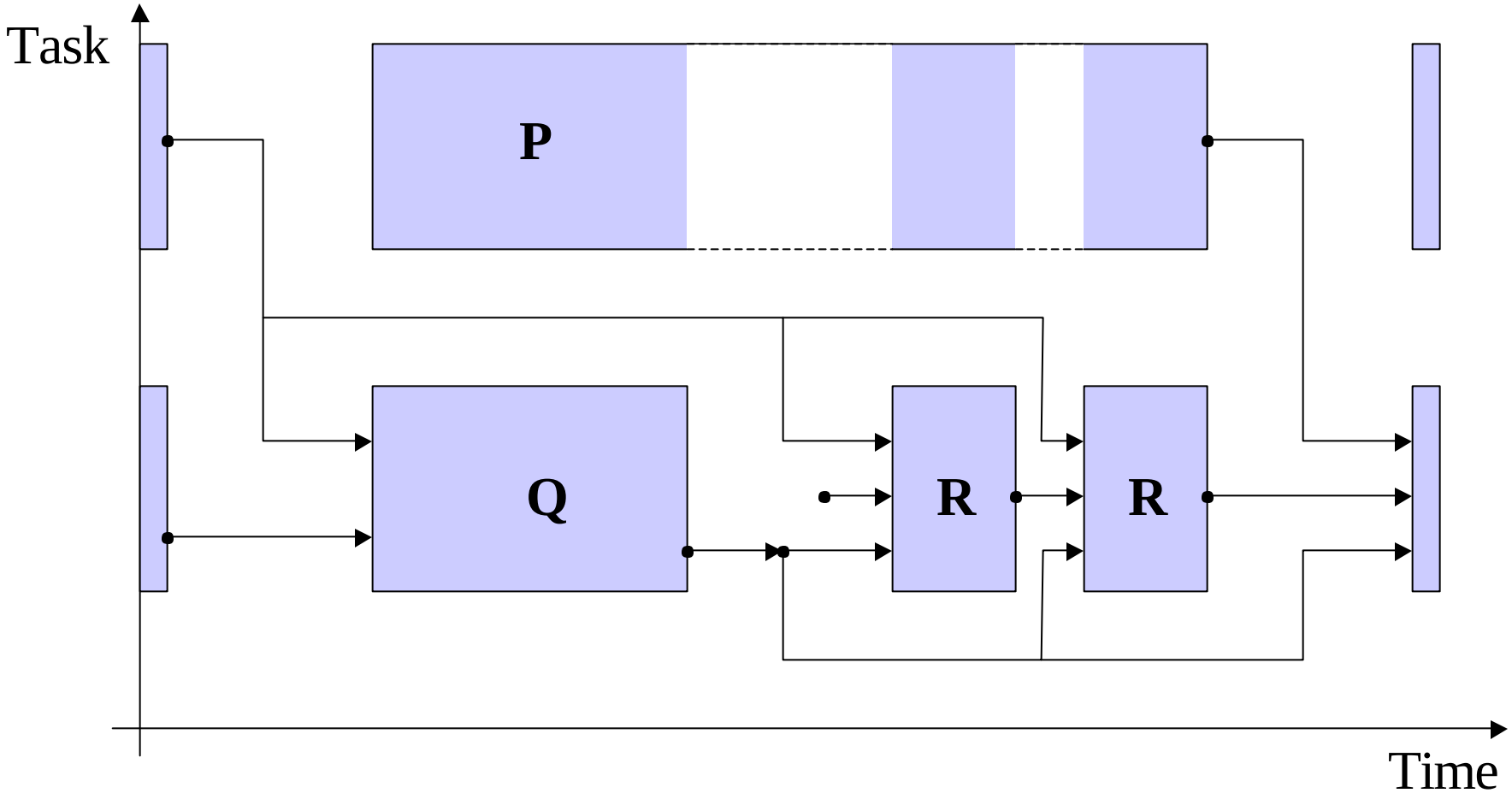
Giotto-PS



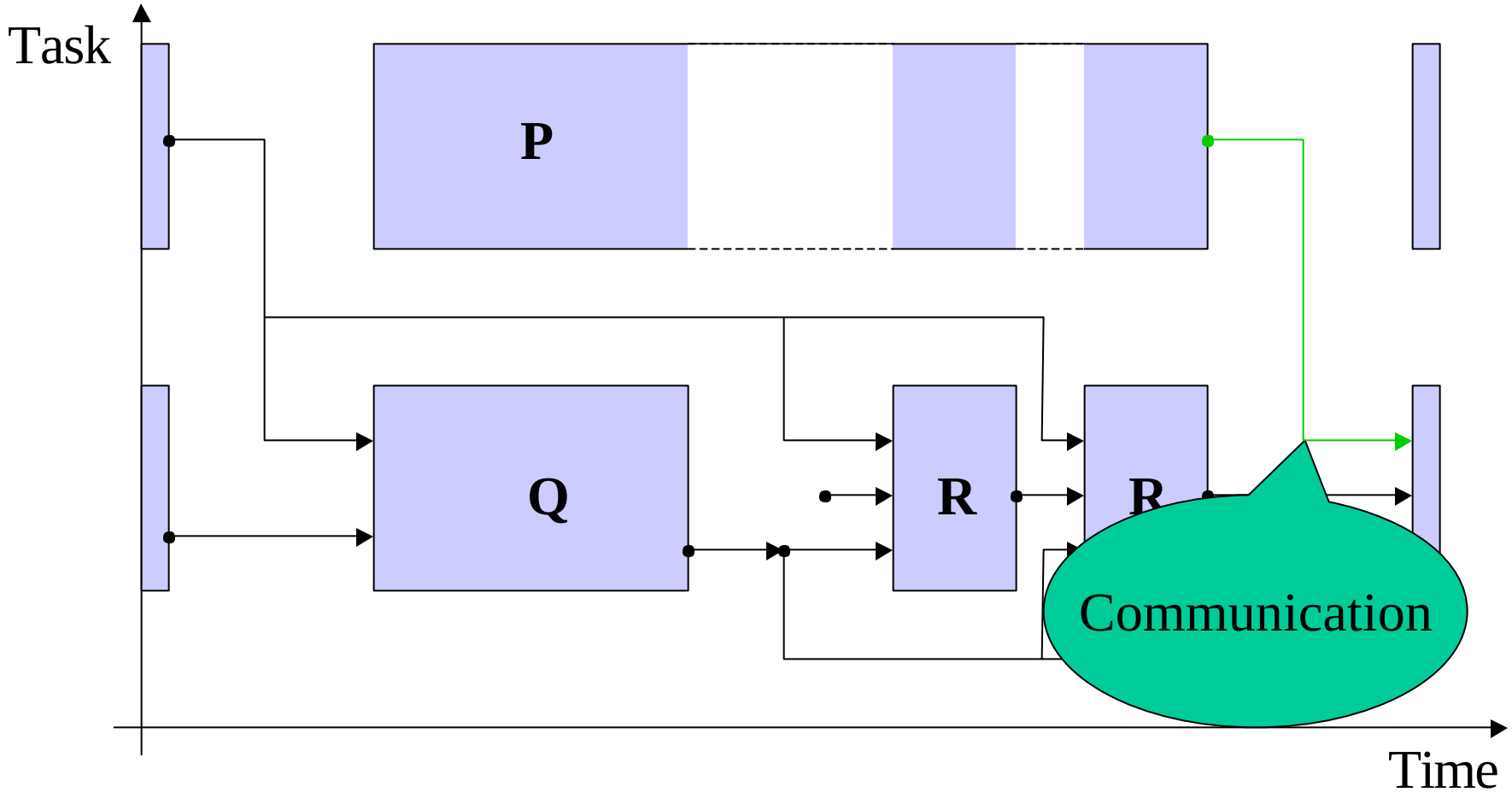
Giotto-PS



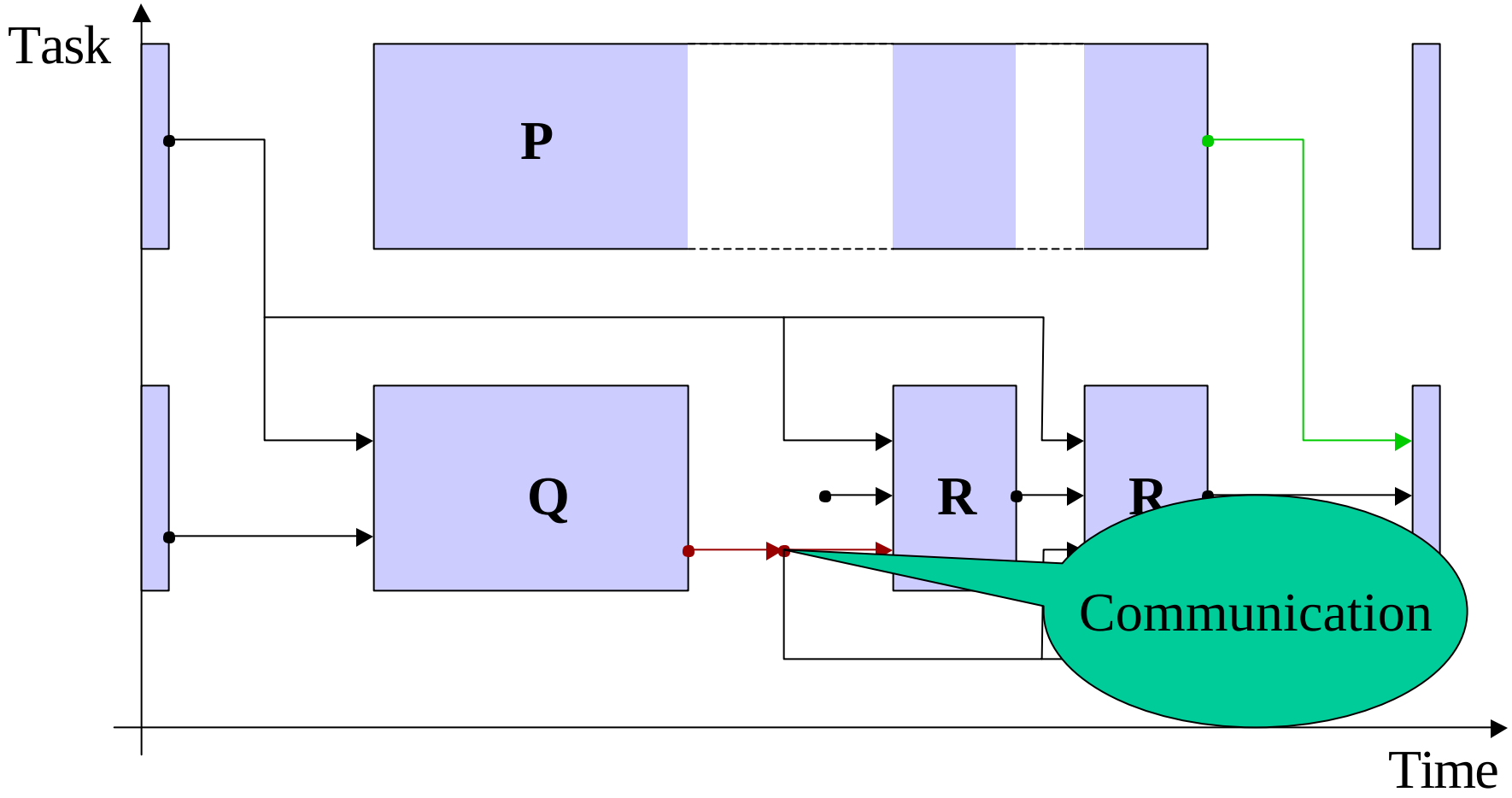
Platform Dependency



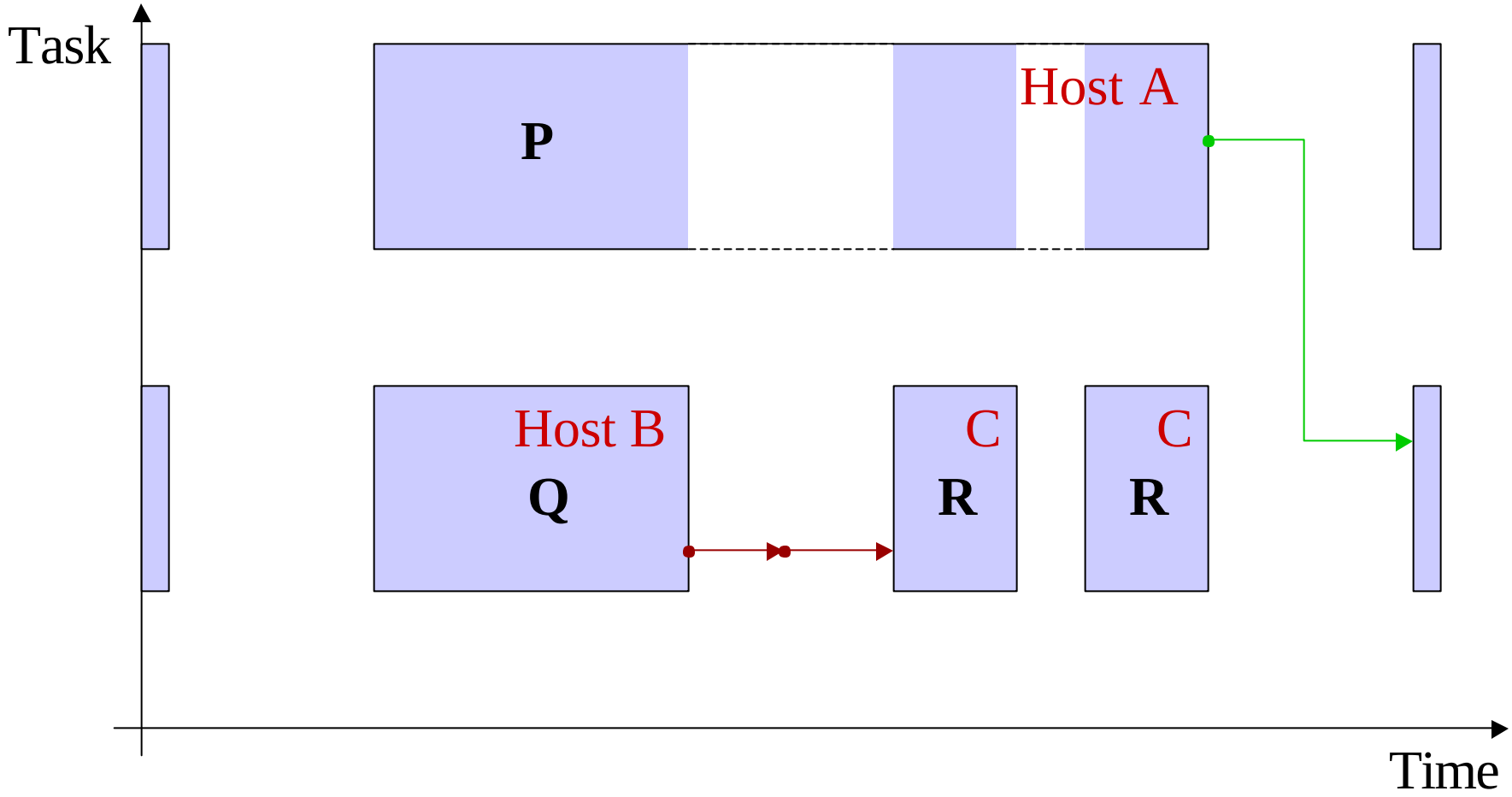
Platform Dependency



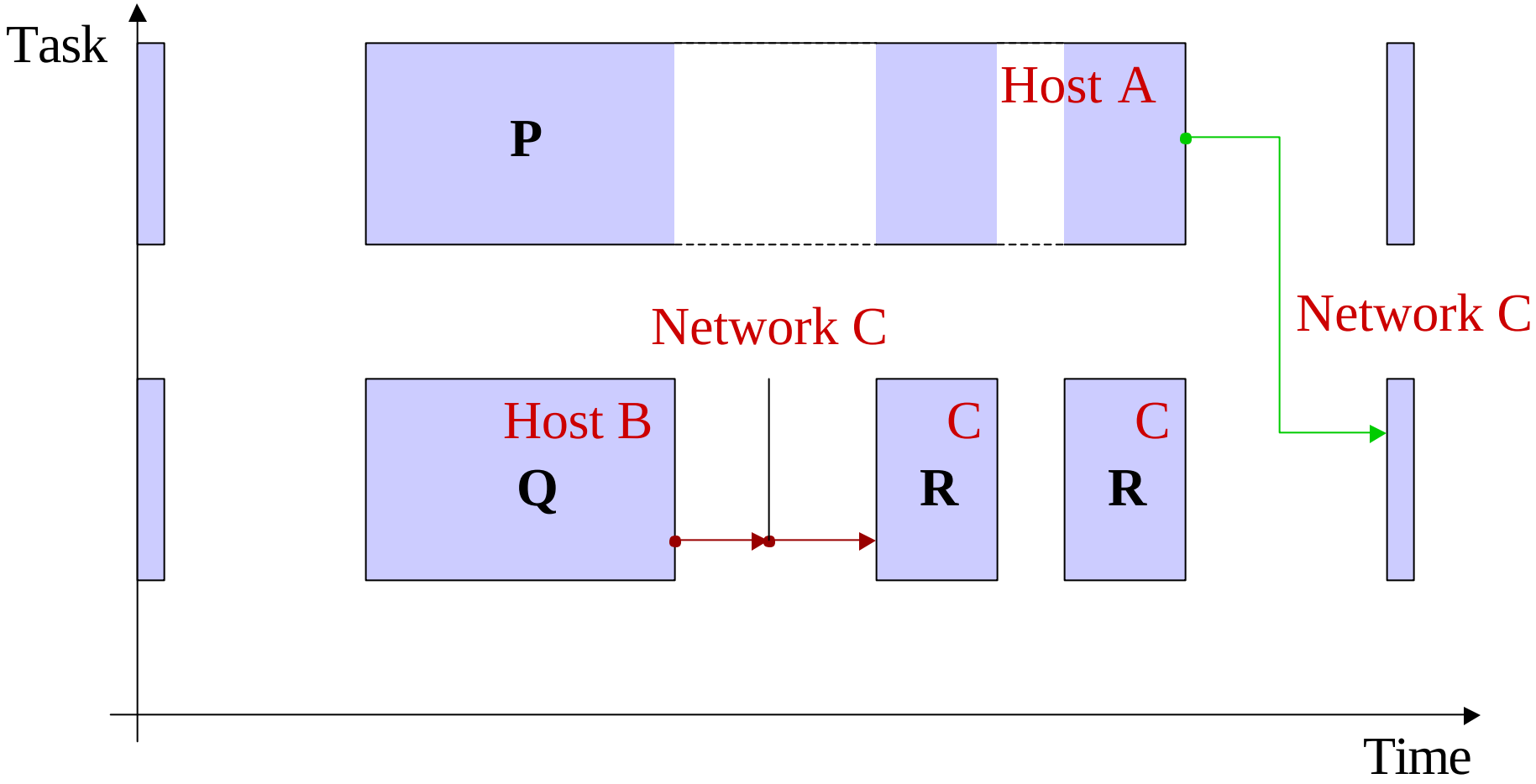
Platform Dependency



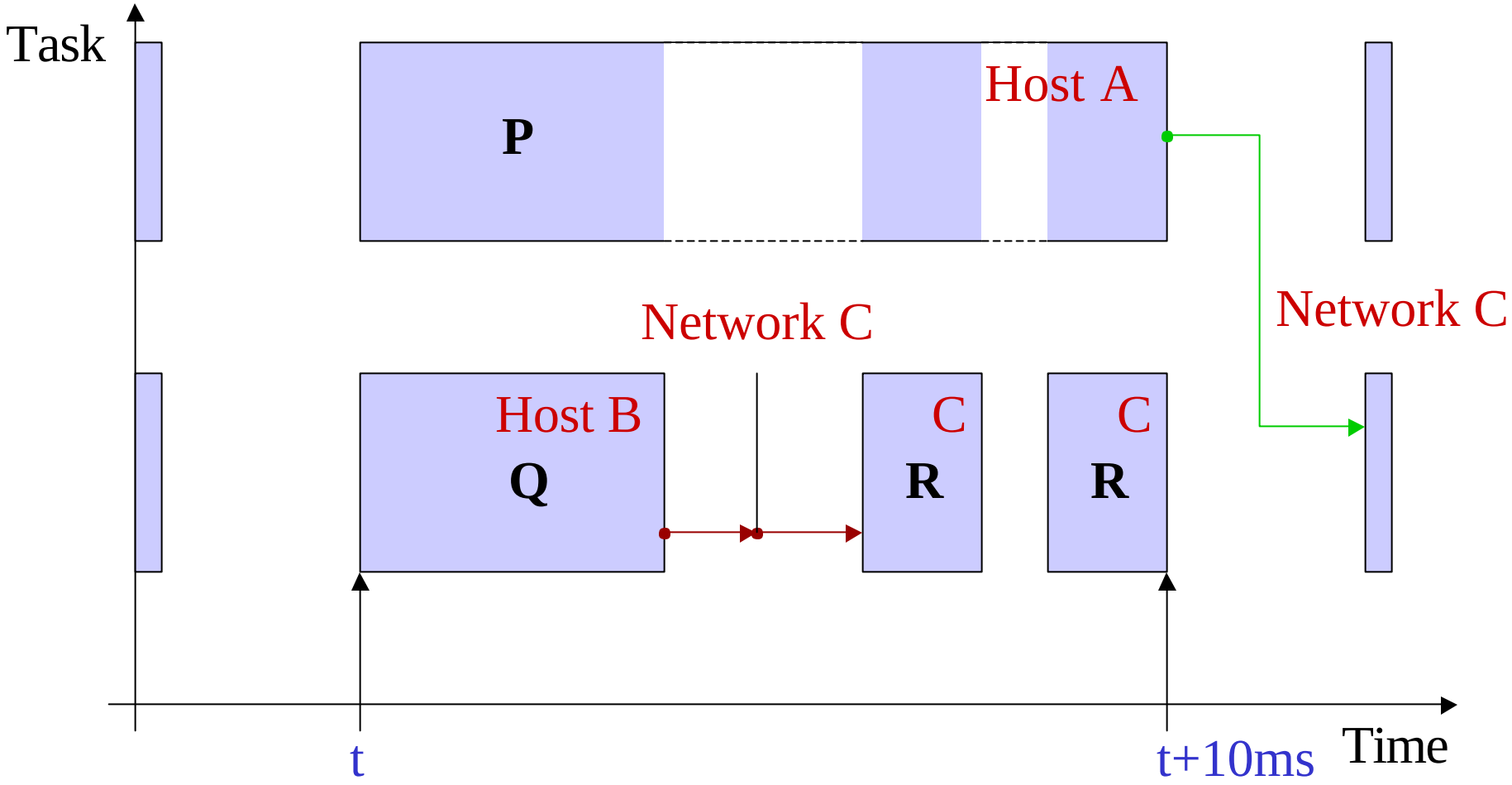
Giotto-PSC



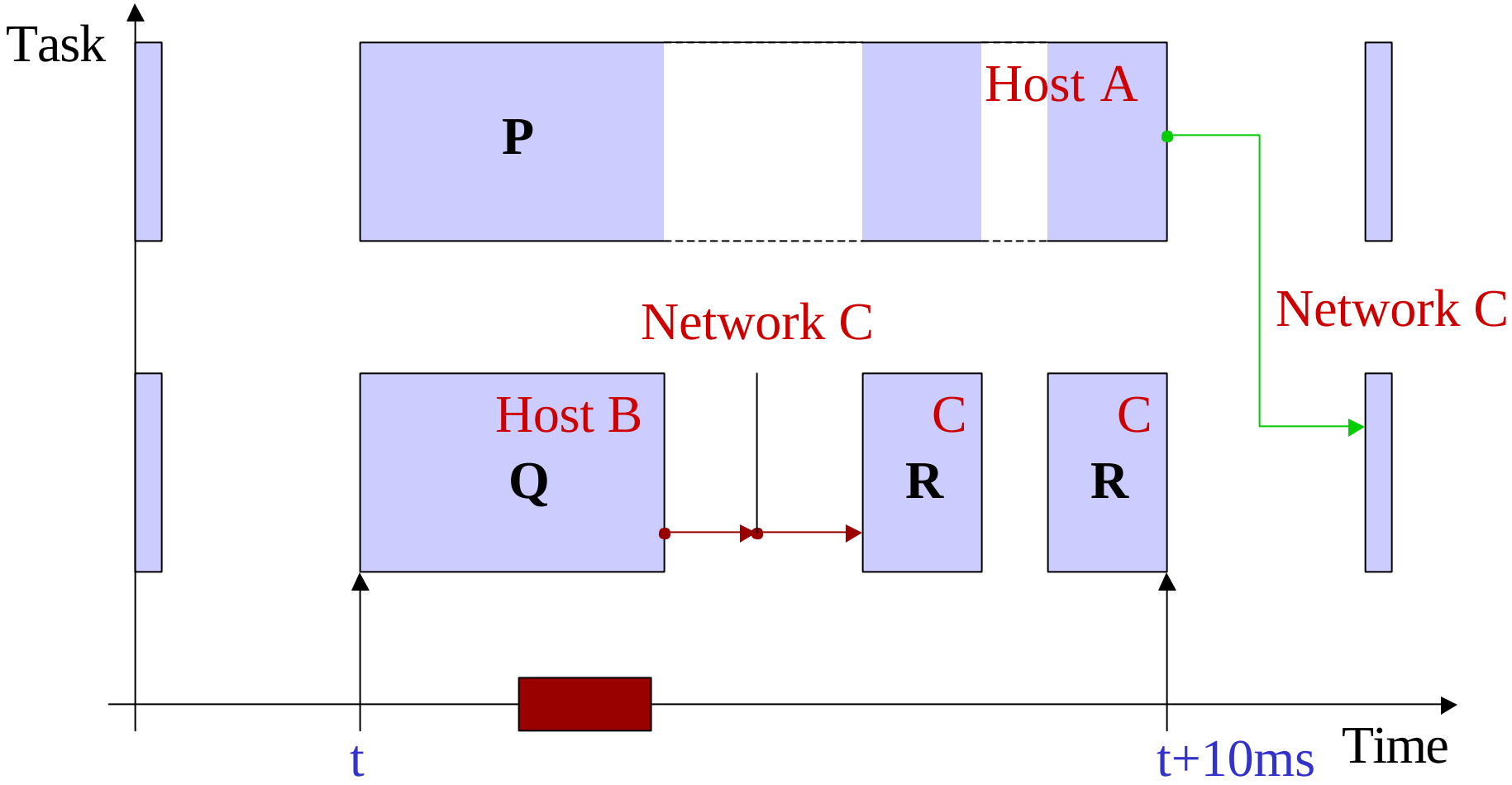
Giotto-PSC



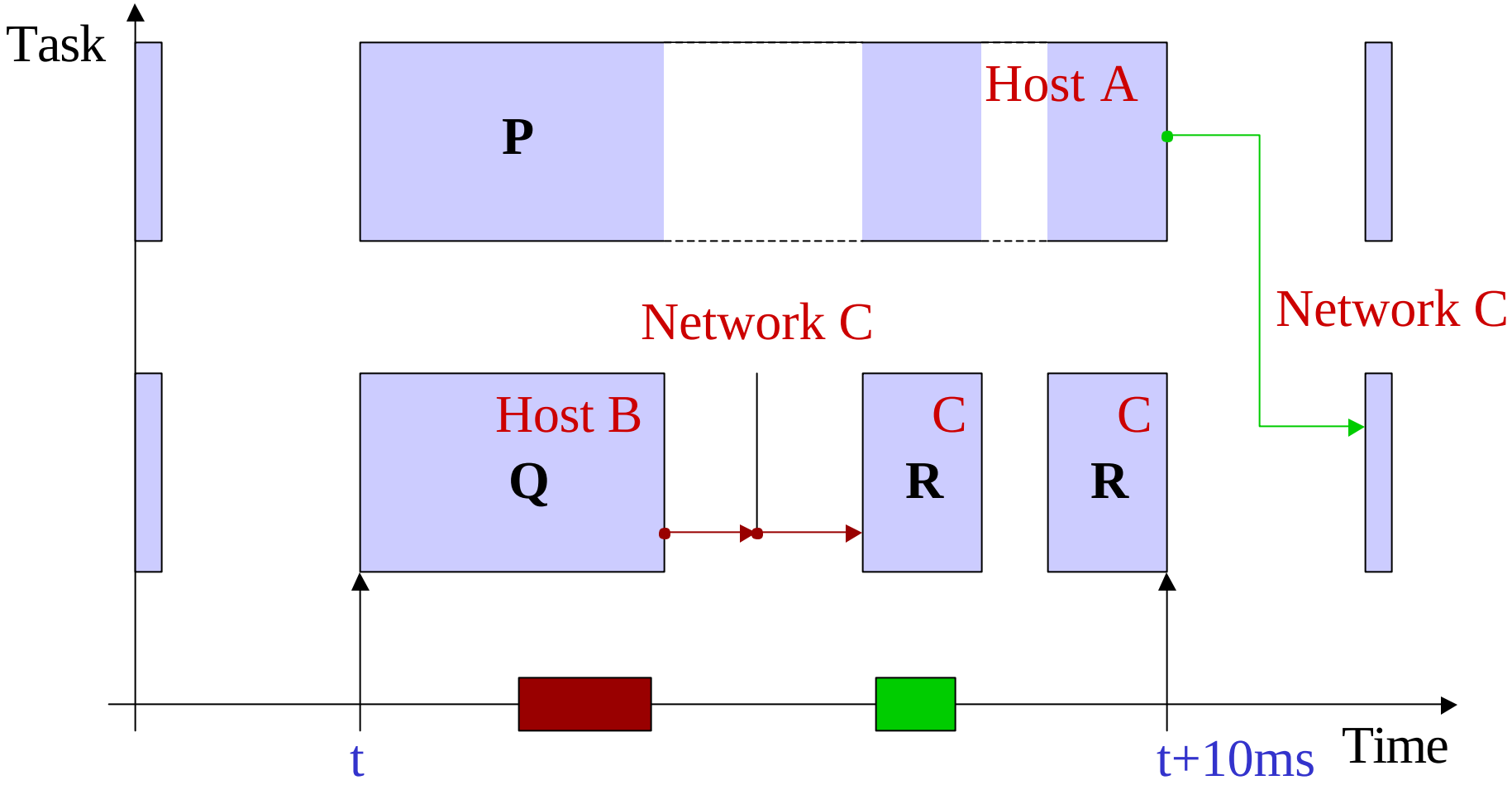
Giotto-PSC



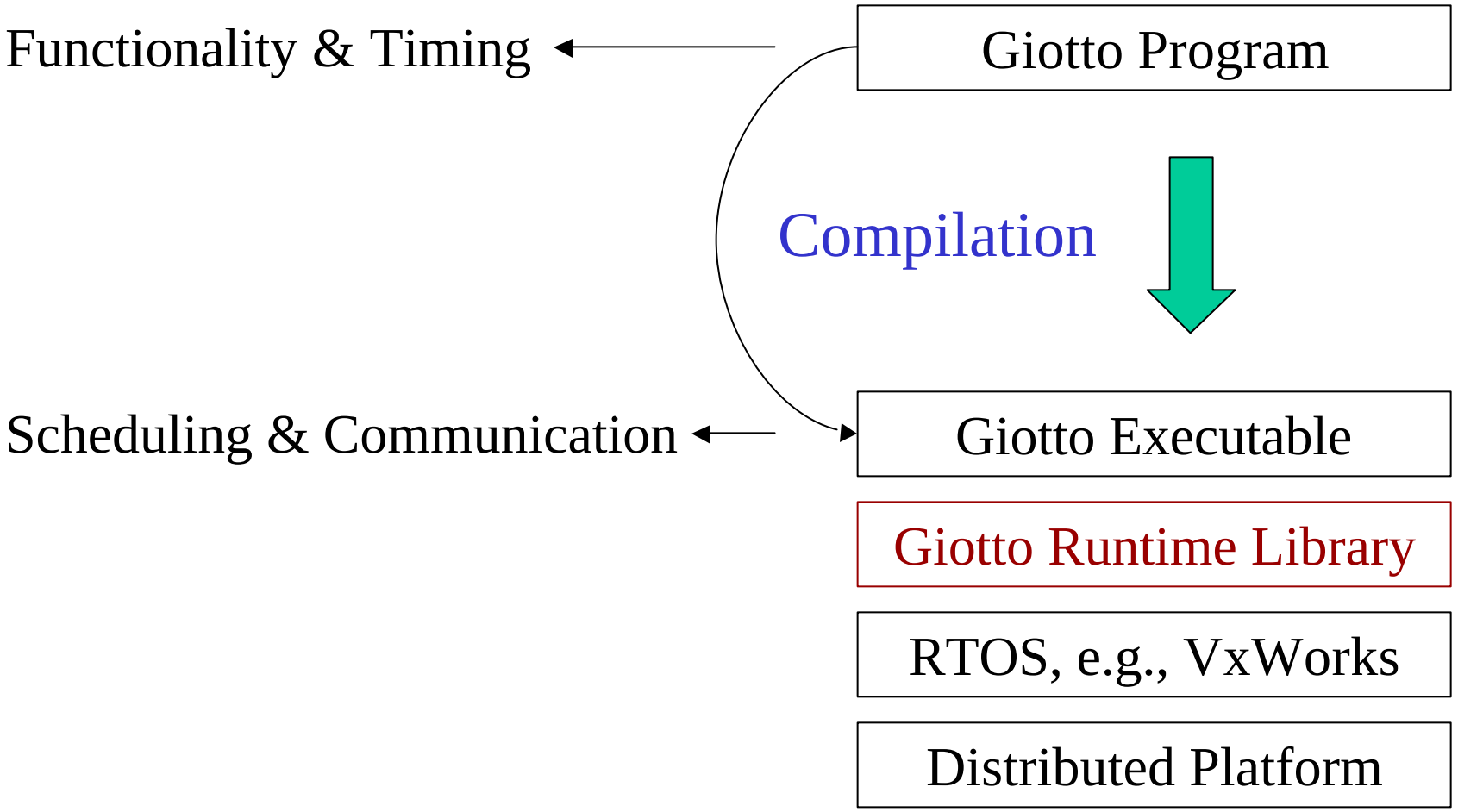
Giotto-PSC



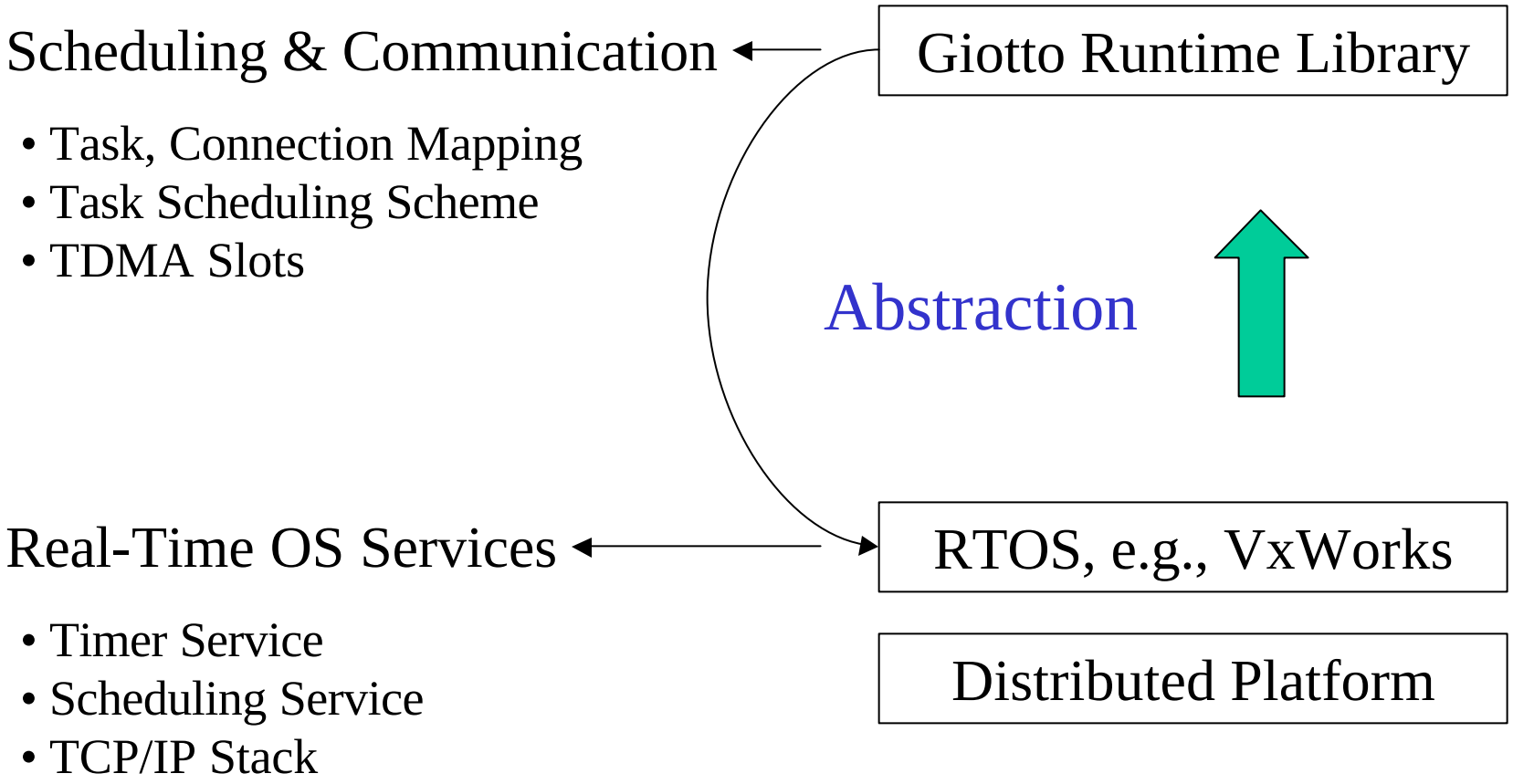
Giotto-PSC



The Giotto Runtime Library



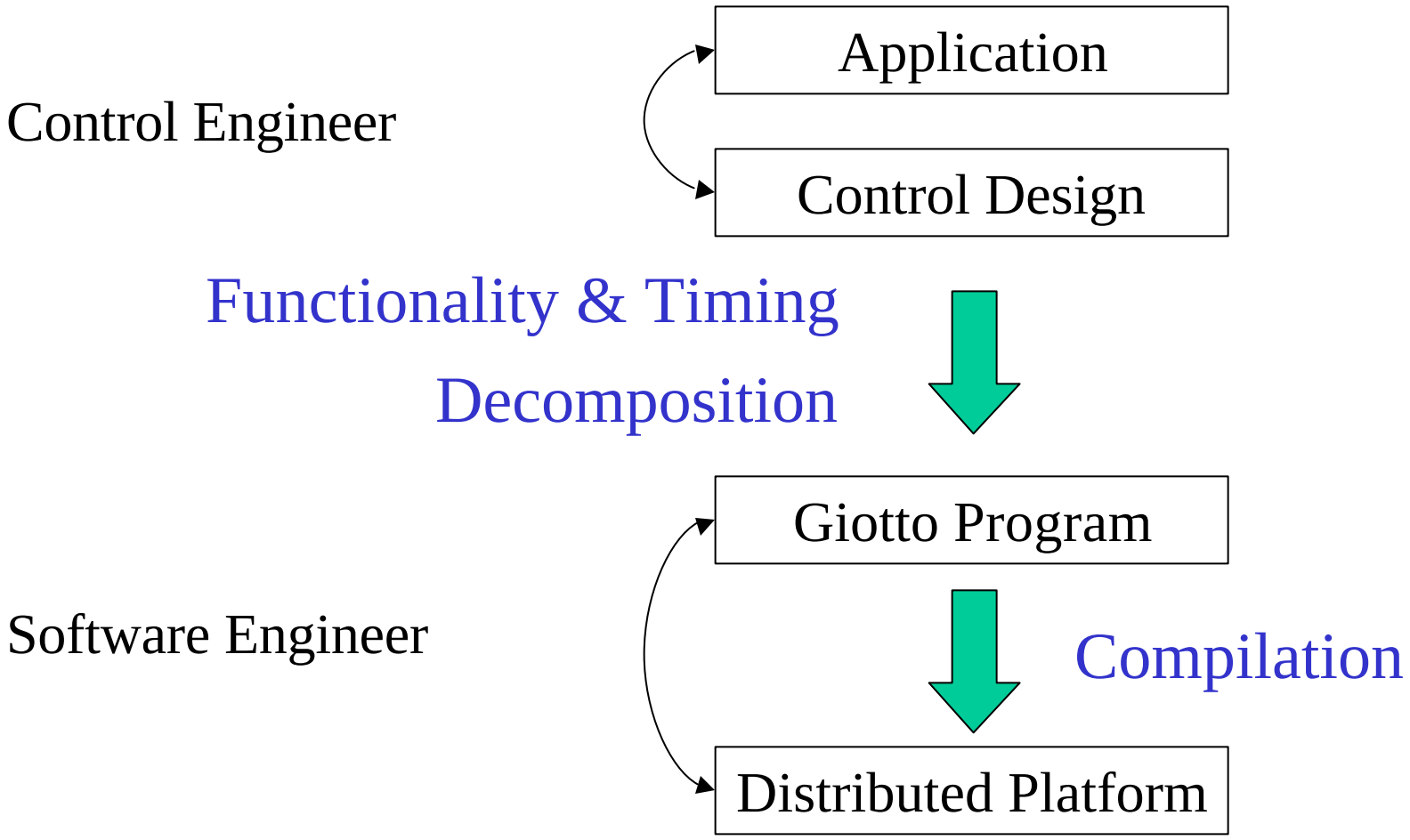
The Giotto Runtime Library



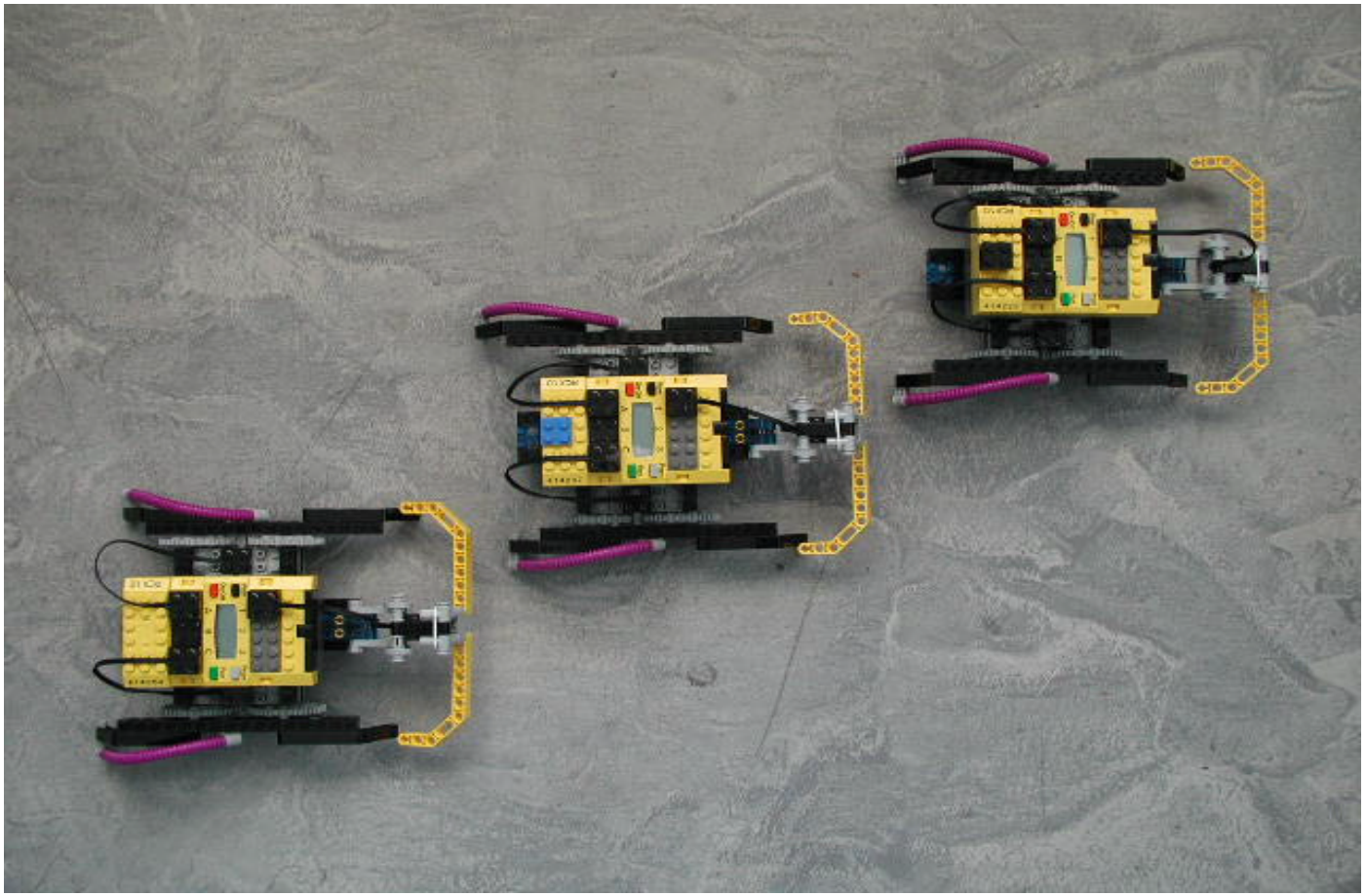
The Giotto Runtime Library

- Available on VxWorks and x86 targets
- Current Experimental Robots:
  - credit card size x86 single board computer running VxWorks
  - wireless Ethernet
  - Lego motors, sensors
- Next Experimental Platform:
  - unmanned Yamaha helicopters
  - PC 104 single board computer running VxWorks
  - wireless Ethernet, GPS, inertial navigation sensor, …

Summary


Control Engineer
Application
Control Design
Functionality & Timing
Decomposition
Software Engineer
Giotto Program
Compilation
Distributed Platform

Demo



Demo

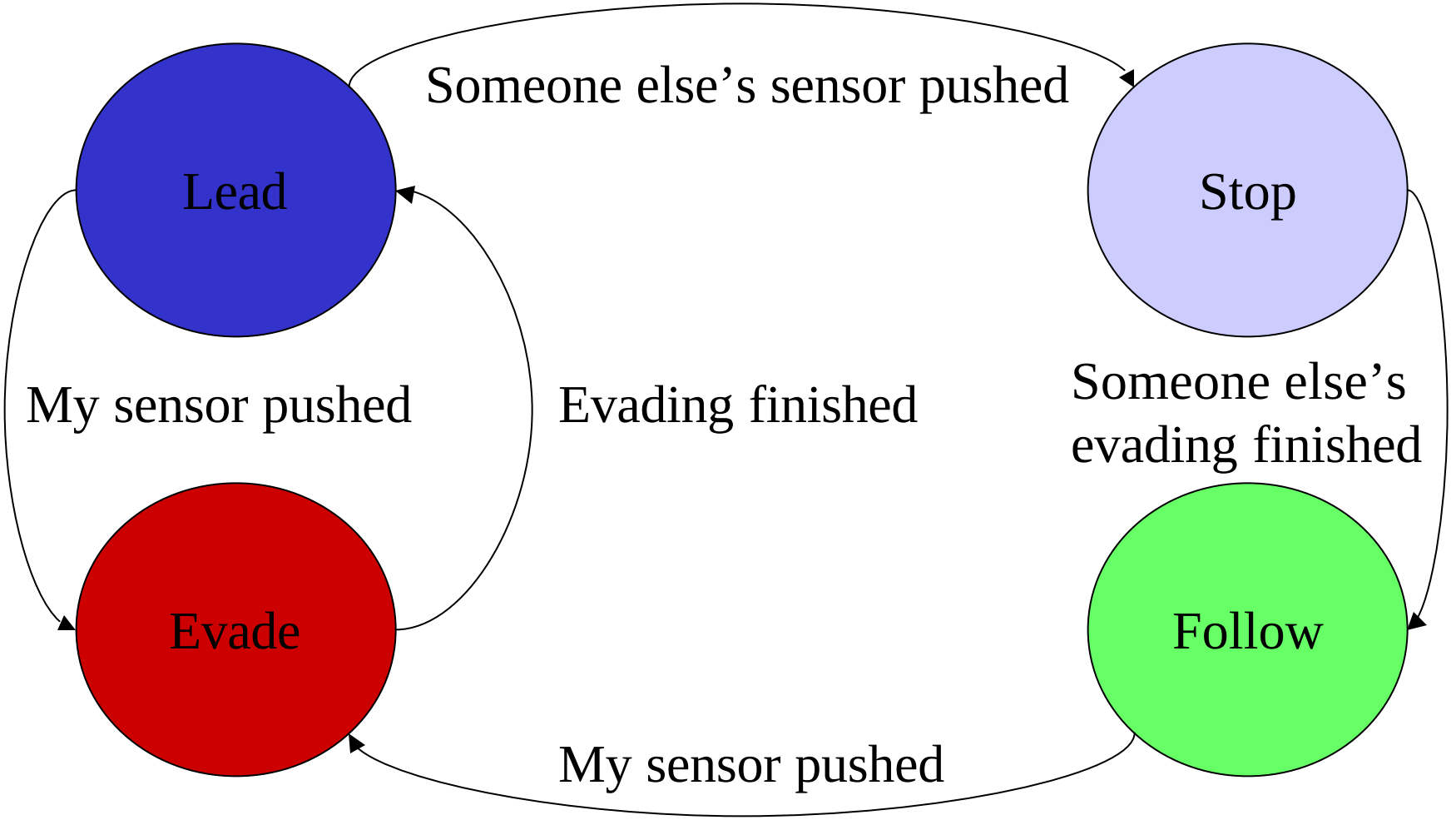
- Giotto on Lego Mindstorms:

  Concurrent distributed execution
- Giotto on Lego Mindstorms and x86 bots running VxWorks:

  Heterogeneous platforms
- Giotto in Ptolemy:

  Design and simulation of Giotto programs

Behavior of a Bot



Demo

- Giotto on Lego Mindstorms:
  Concurrent distributed execution

Heterogeneous Platforms

1. Lego Mindstorms: Lego Firmware, Hitachi Microcontroller, Infrared Link
2. x86 bots: VxWorks, x86 single board computer, Wireless Ethernet

3. Infrared – Wireless Ethernet Bridge: Laptop

Demo

- Giotto on Lego Mindstorms and x86 bots running VxWorks:

  Heterogeneous platforms

Dmitry Derevyanko

Benjamin Horowitz

Christoph Meyer Kirsch

Win Williams

Ptolemy Simulation

Ptolemy is a software package supporting design and simulation of concurrent, real-time, embedded systems:

Giotto defines a model of computation

1. Ptolemy simulates bot example
2. Lego Mindstorms display simulation

Demo

- Giotto in Ptolemy:
  Design and simulation of Giotto programs

Christoph Meyer Kirsch
Edward A. Lee
Xiaojun Liu

End