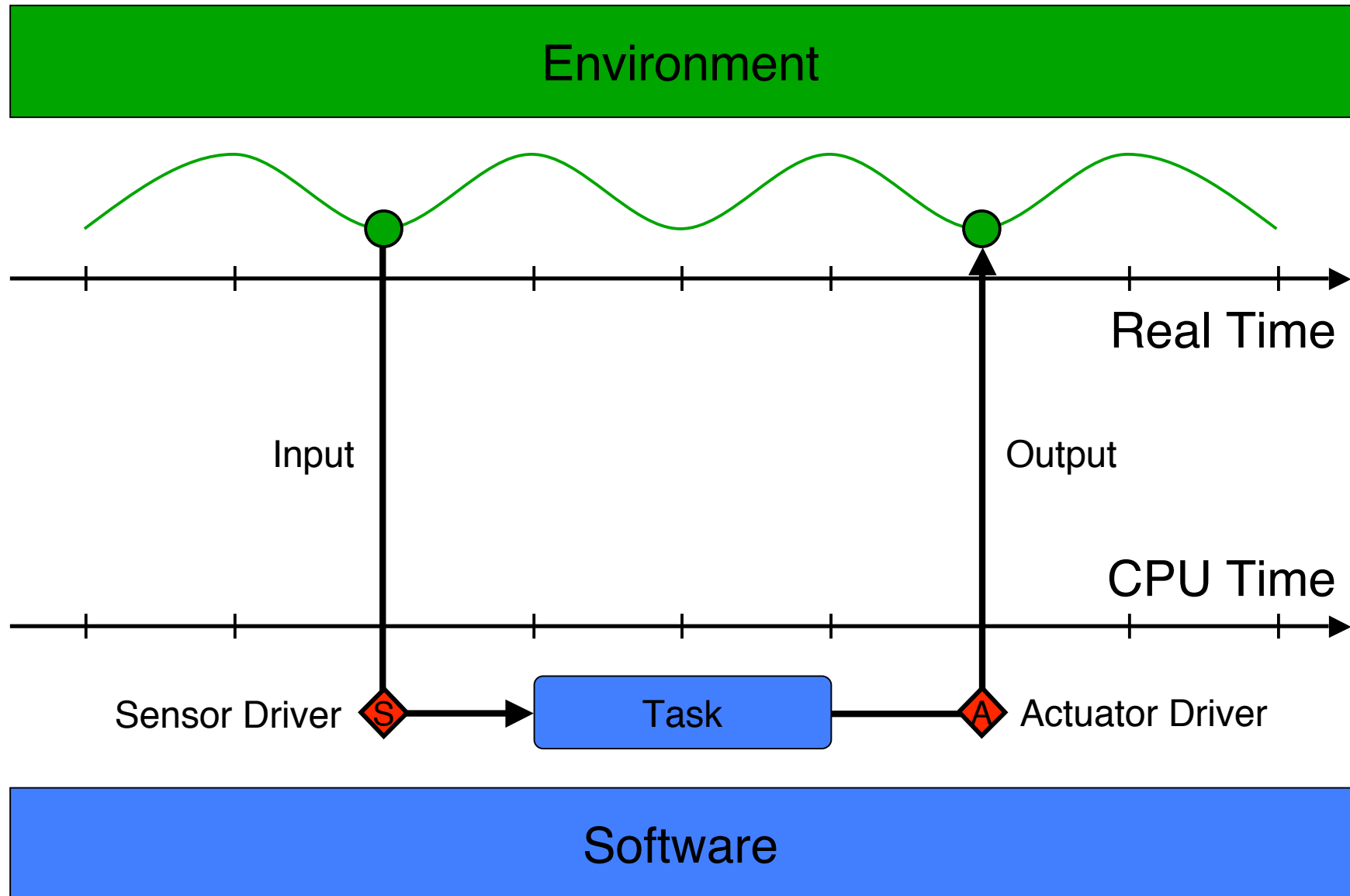# The Embedded Machine: Status and Future Directions

IBM Research Center, Hawthorne, March 26, 2005
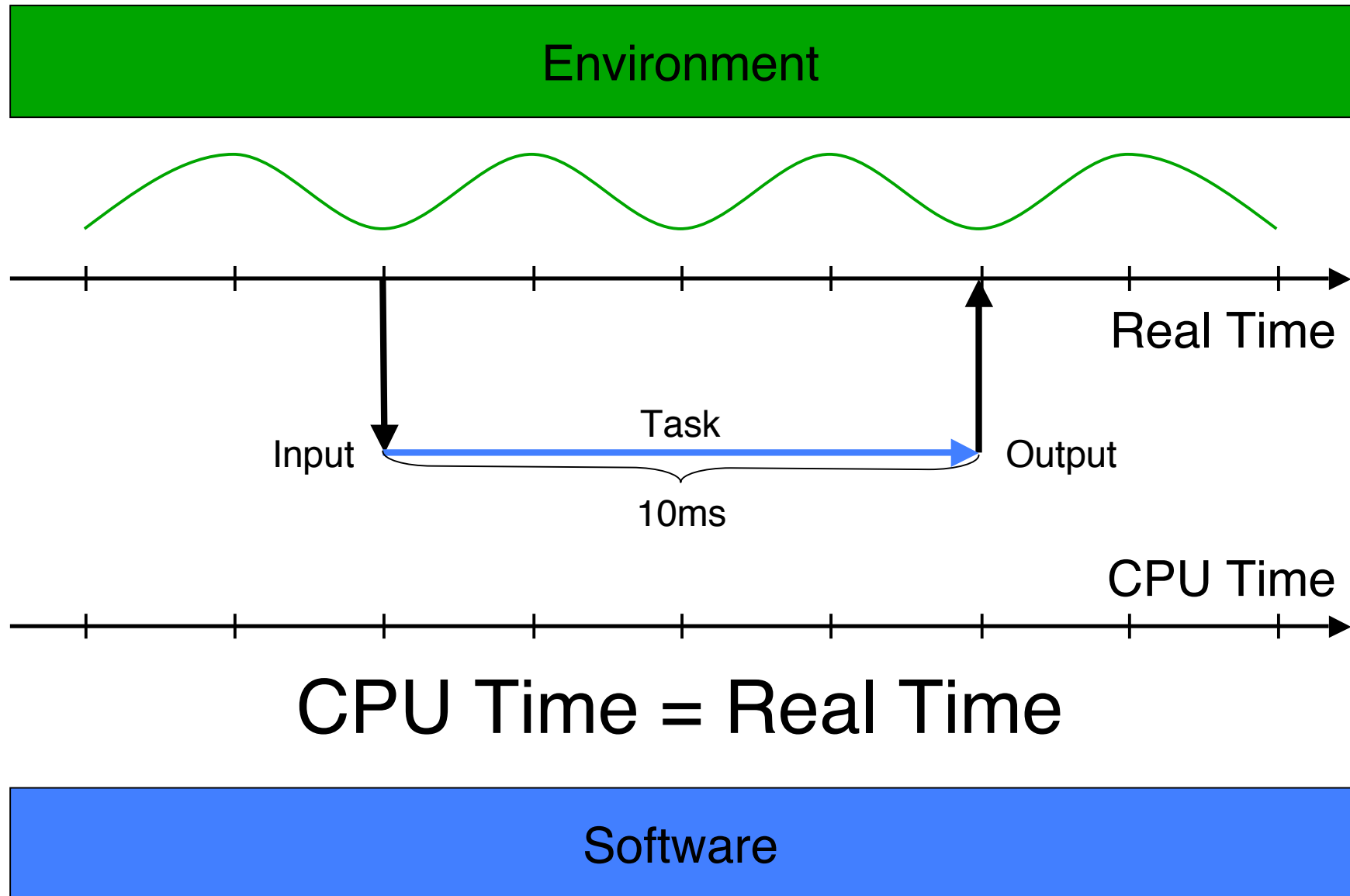
Christoph M. Kirsch

University of Salzburg

www.cs.uni-salzburg.at/~ck

Sensing, Computing, Actuating

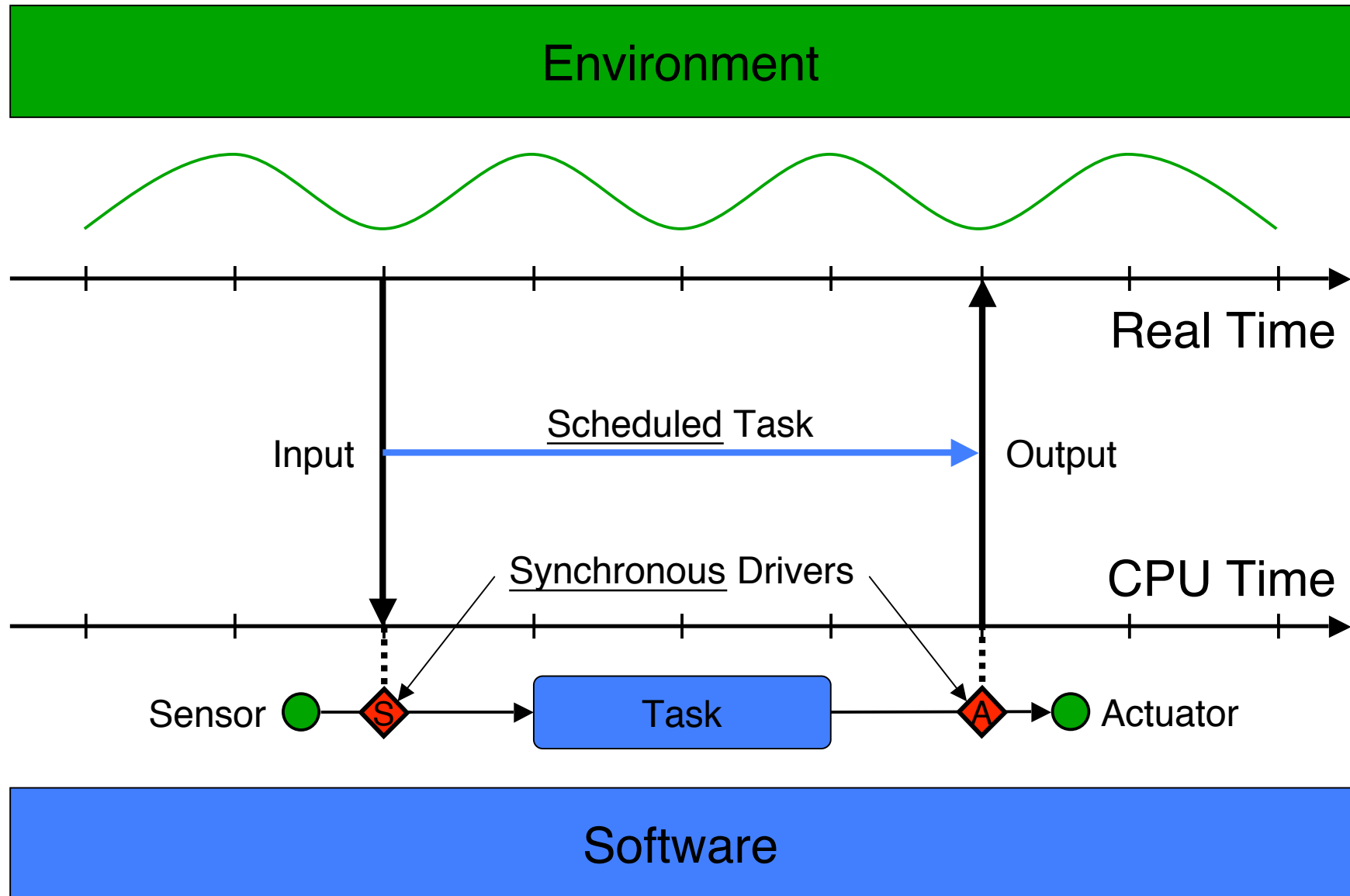**Environment**

Input

Output

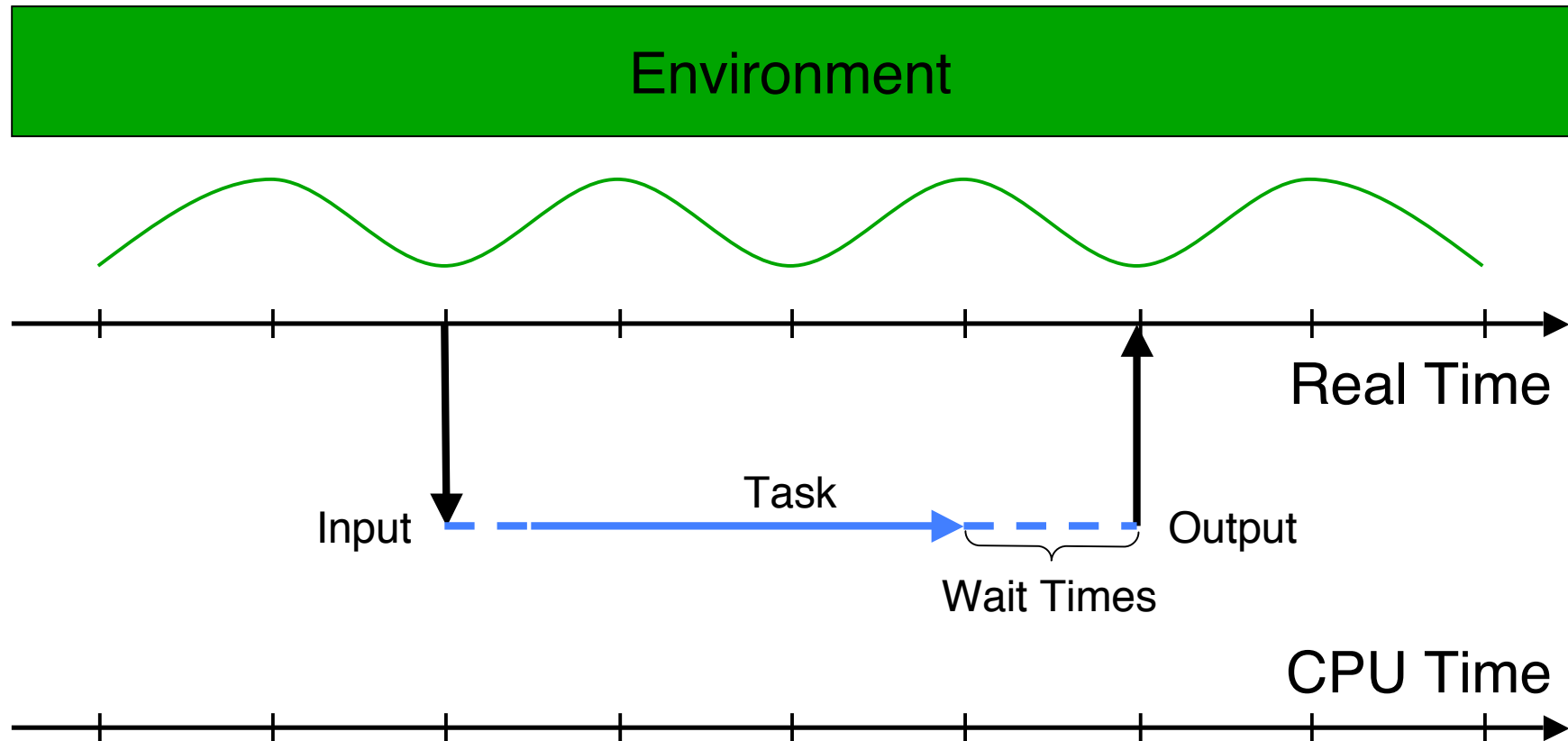Real Time

CPU Time

Sensor Driver **S** → **Task** **A** Actuator Driver

Software

Logical Execution Time (LET)

Environment

Real Time

Input   Task   Output

10ms

CPU Time

CPU Time = Real Time

Software

Synchronous vs. Scheduled Computation

A <u>Time-Safe</u> Implementation

Environment

Real Time

Input

Task

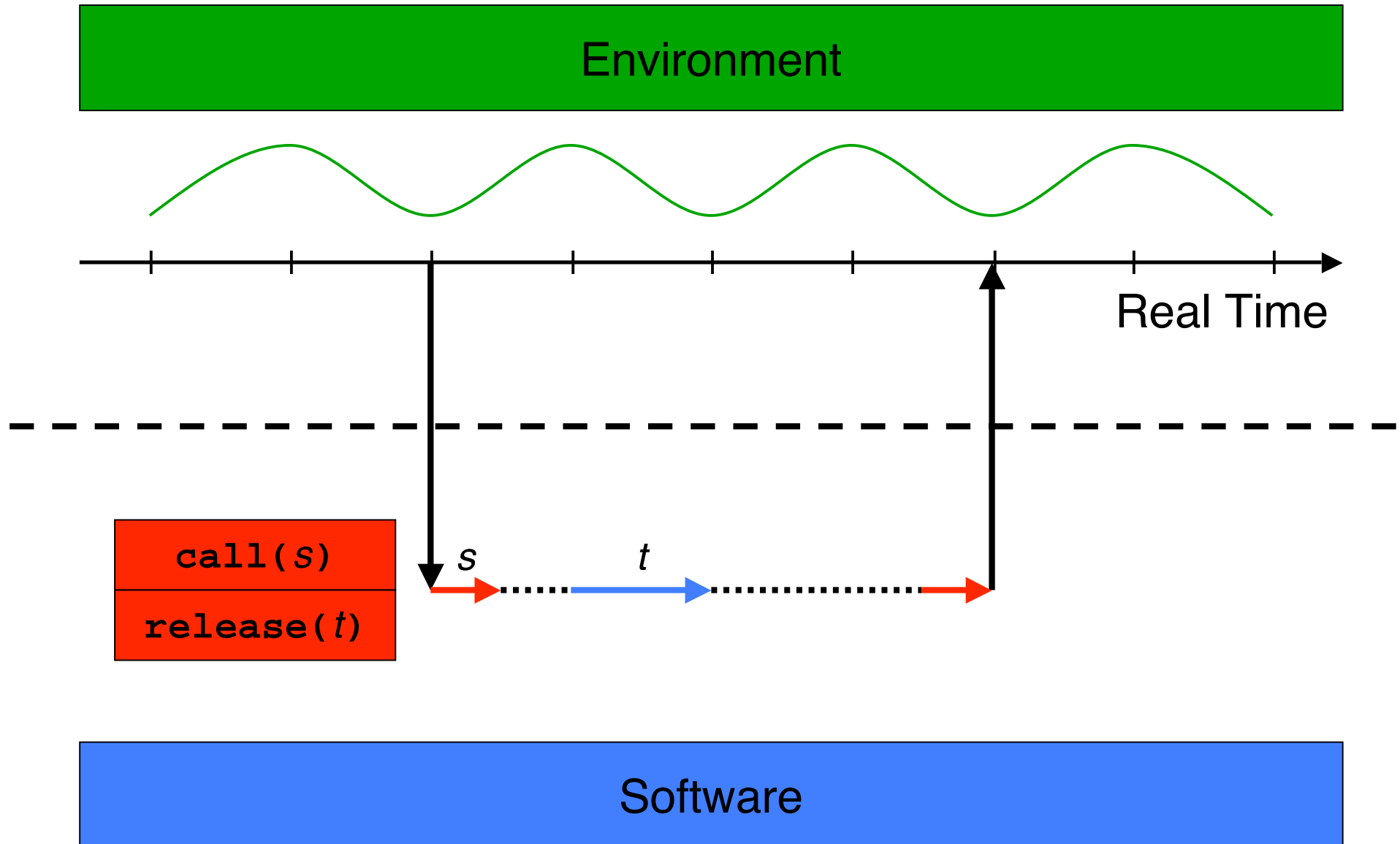Wait Times

Output

CPU Time

Time Safety Implies Time Determinism
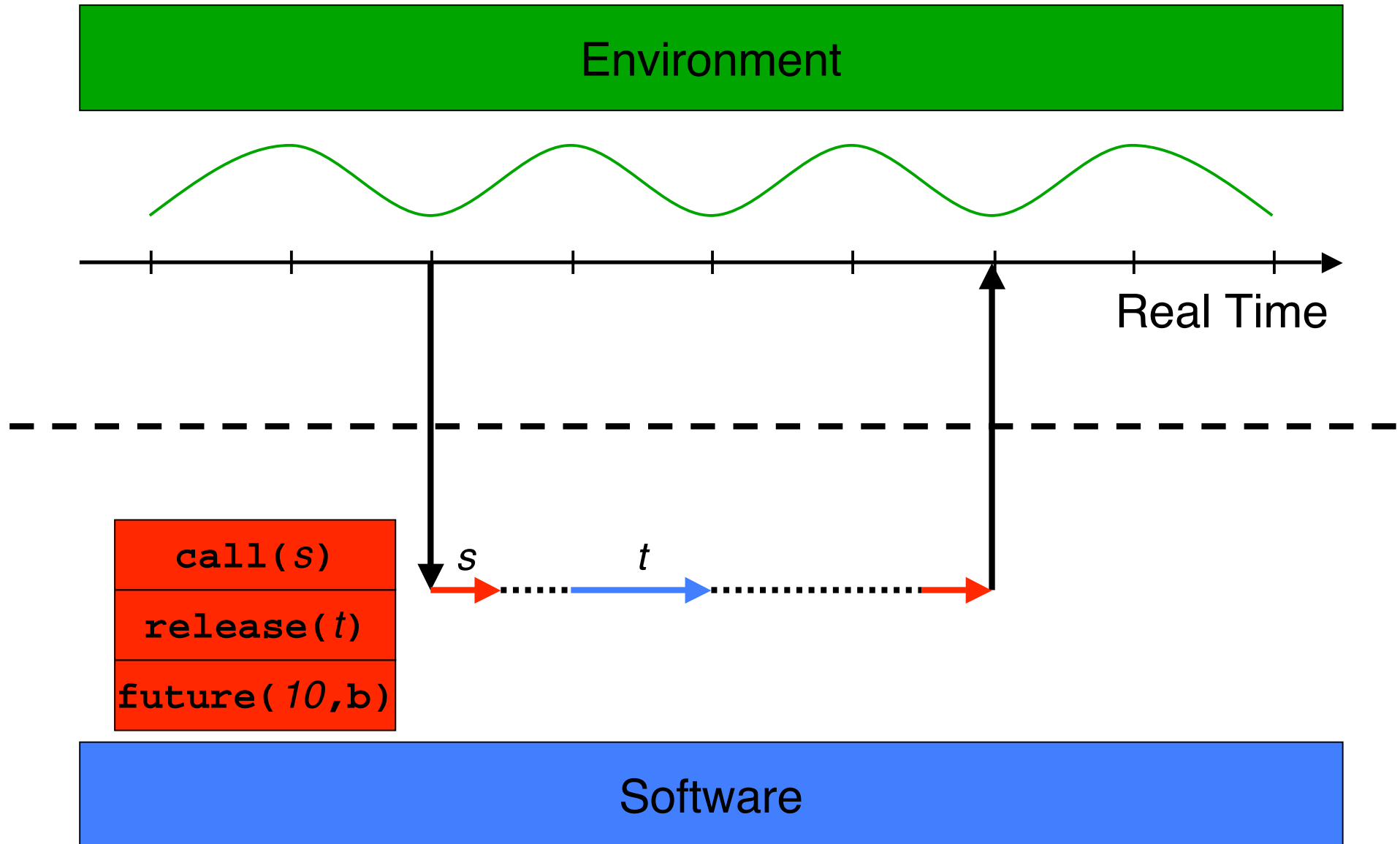
Software

The Embedded Machine
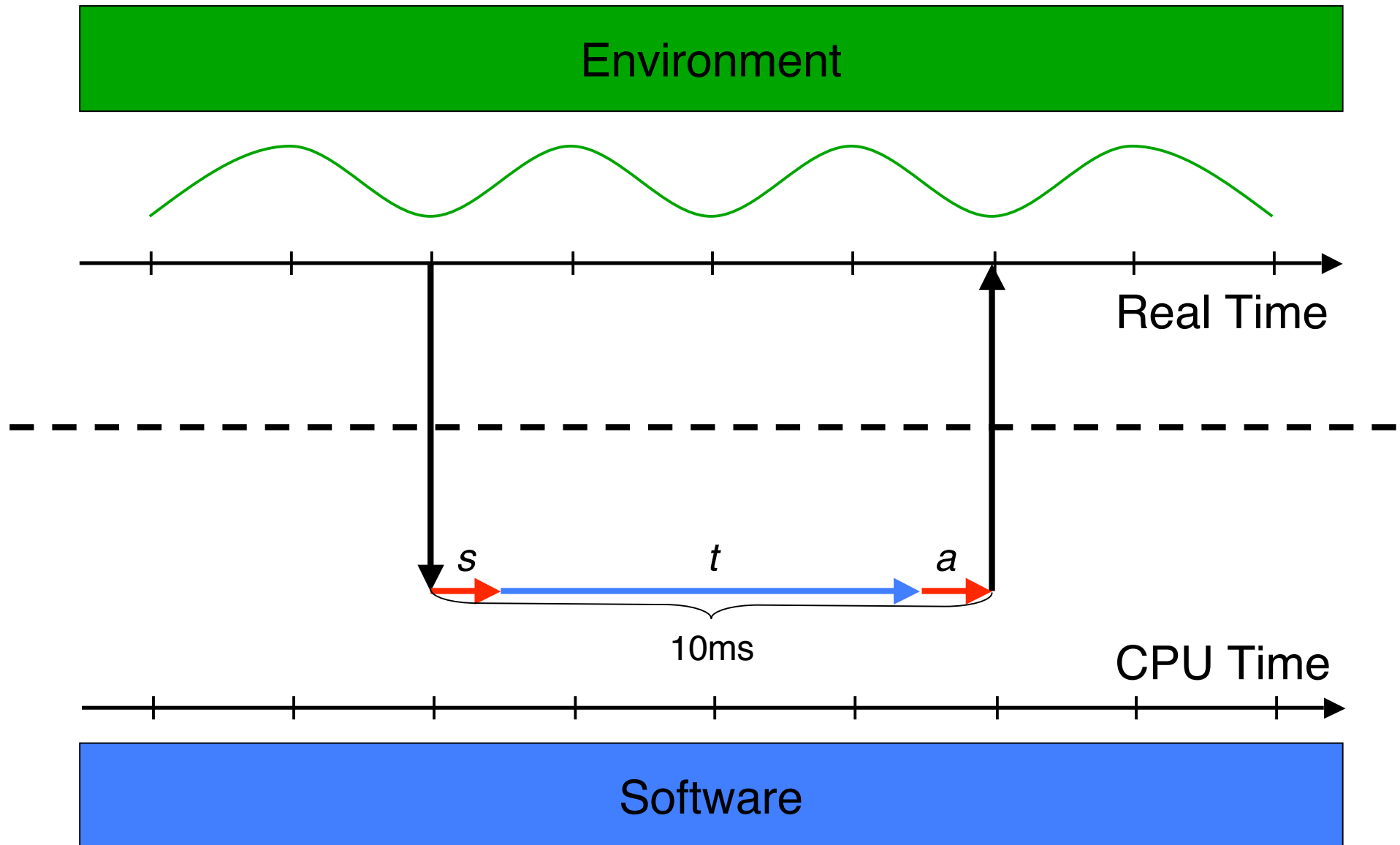
E Code
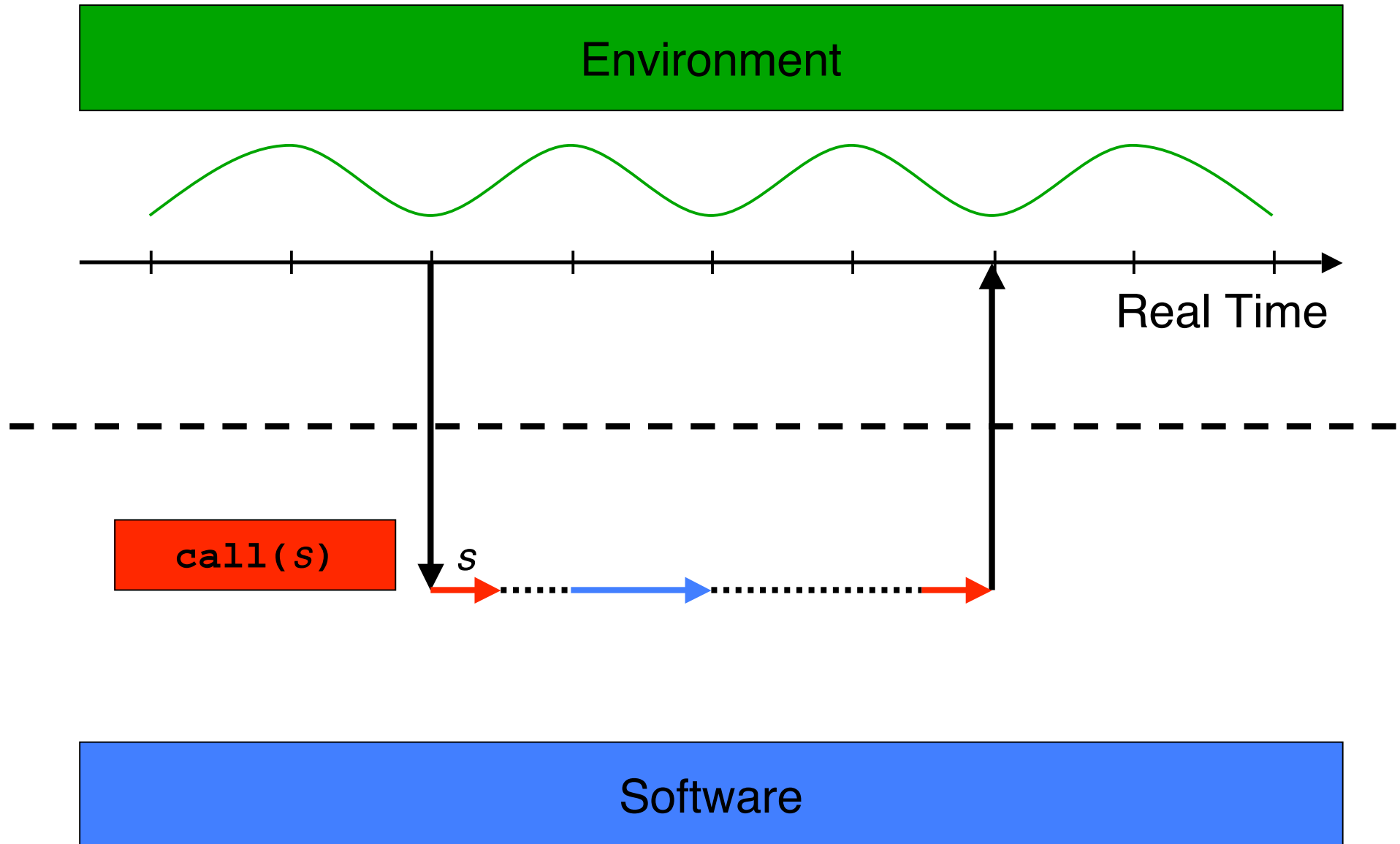
E Code

E Code

E Code
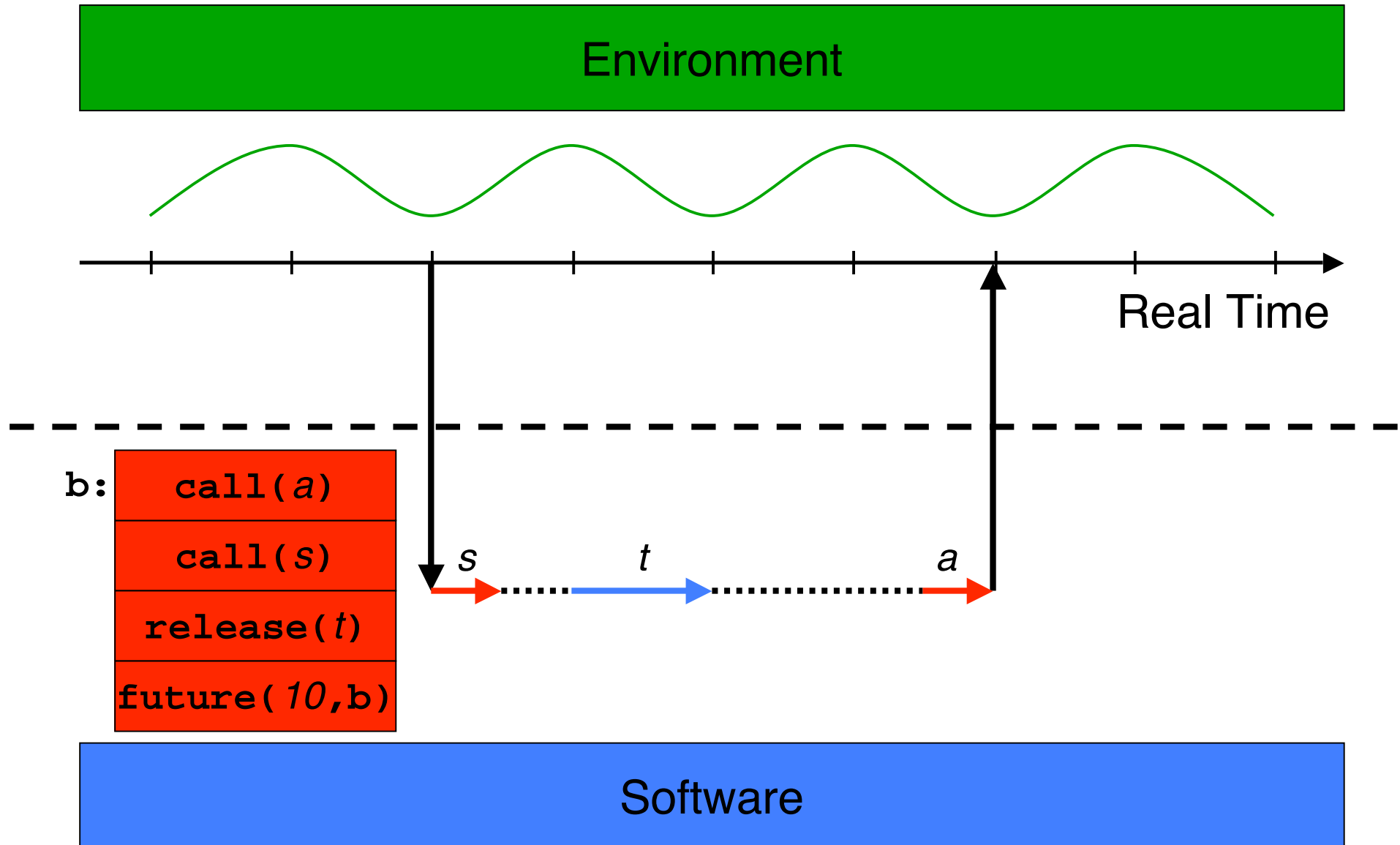
Two Tasks, Different Frequency

Environment

b1: release(*t1*)
release(*t2*)
future(*5*,b2)

b2: release(*t2*)
future(*5*,b1)

t1

t2   *t2*

5ms   5ms

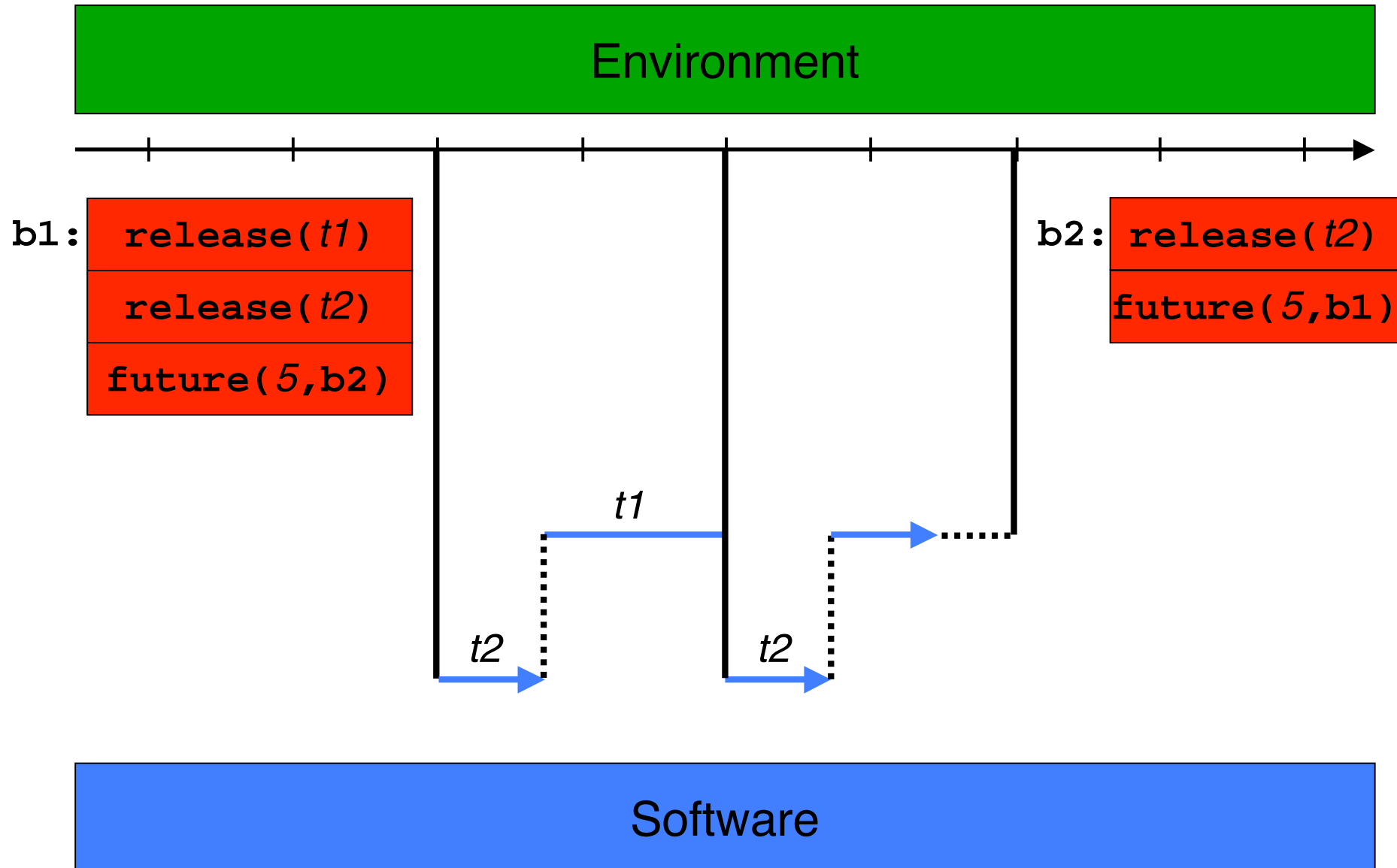Software
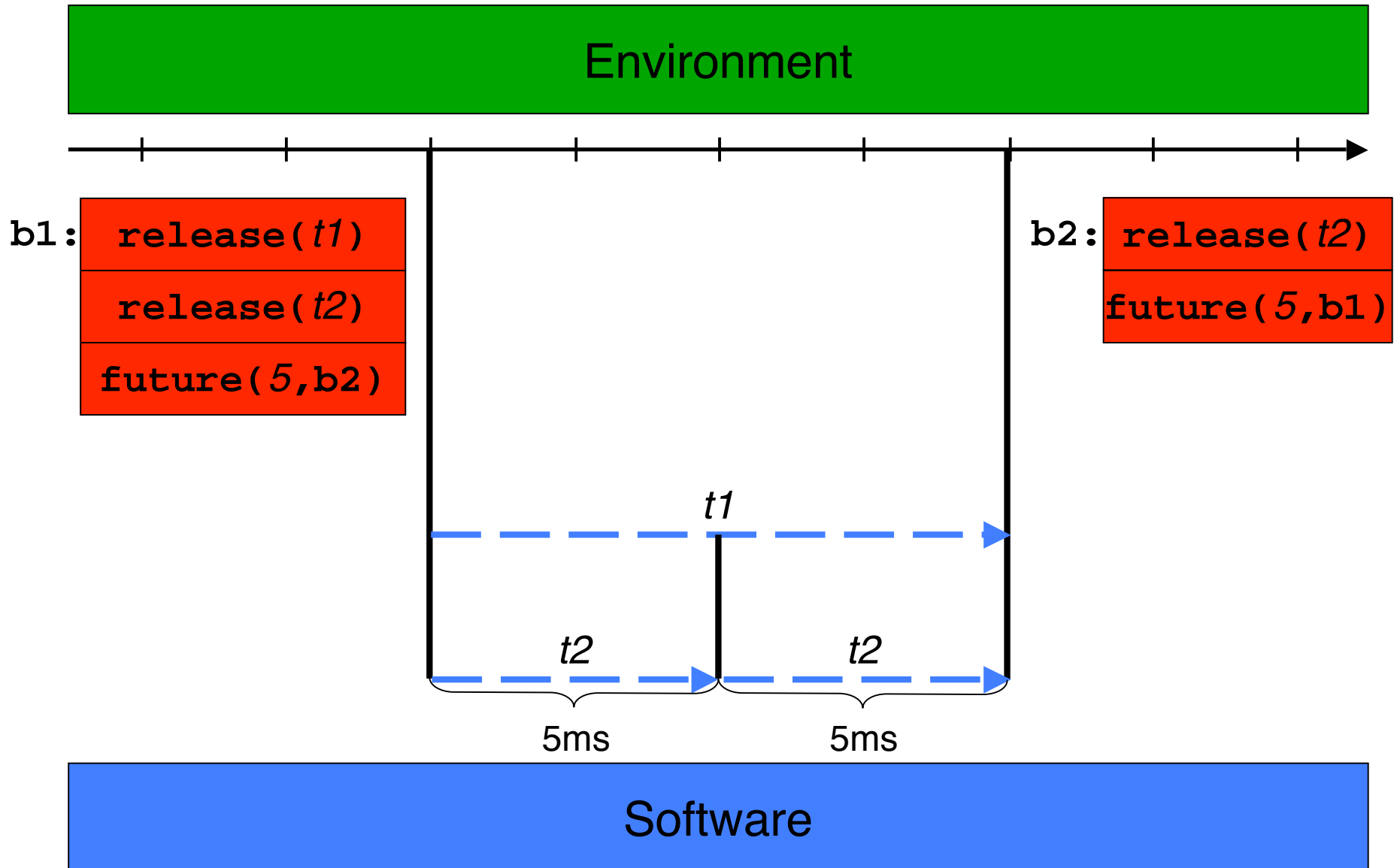
Earliest Deadline First Scheduling

Giotto on the ETH Zürich Helicopter
(Kirsch, Sanvido, Henzinger, Pree in Proc. of EMSOFT 2002)

6 degrees of freedom, 1 processor (StrongARM 200Mhz)

Giotto on the UC Berkeley Helicopter
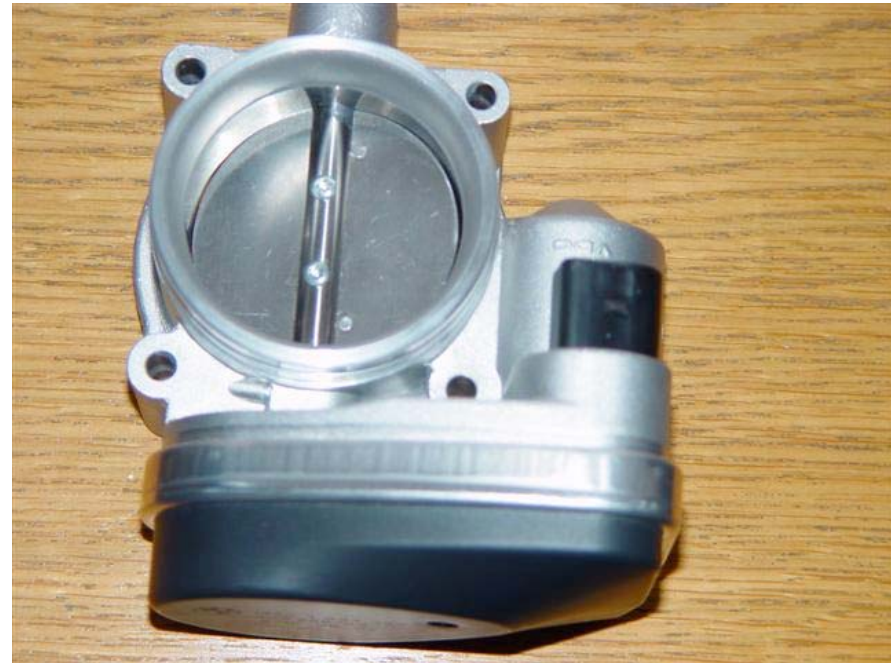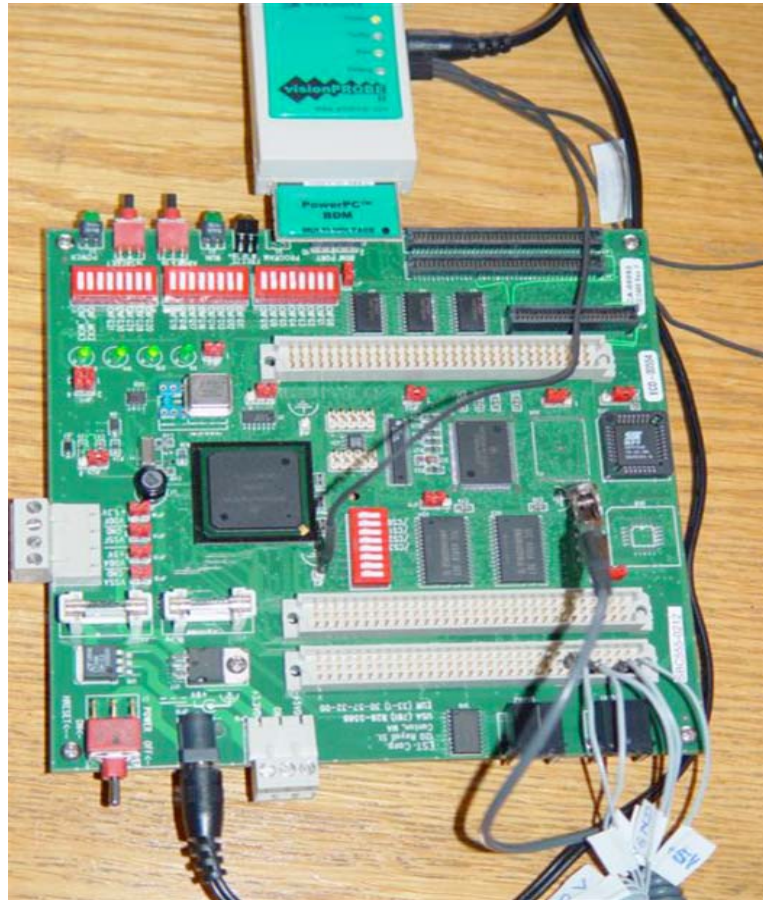(Part of the DARPA SEC Project with Boeing and Honeywell)

6 degrees of freedom, 3 processors (Intel x86)

Giotto for a Drive-By-Wire BMW Throttle
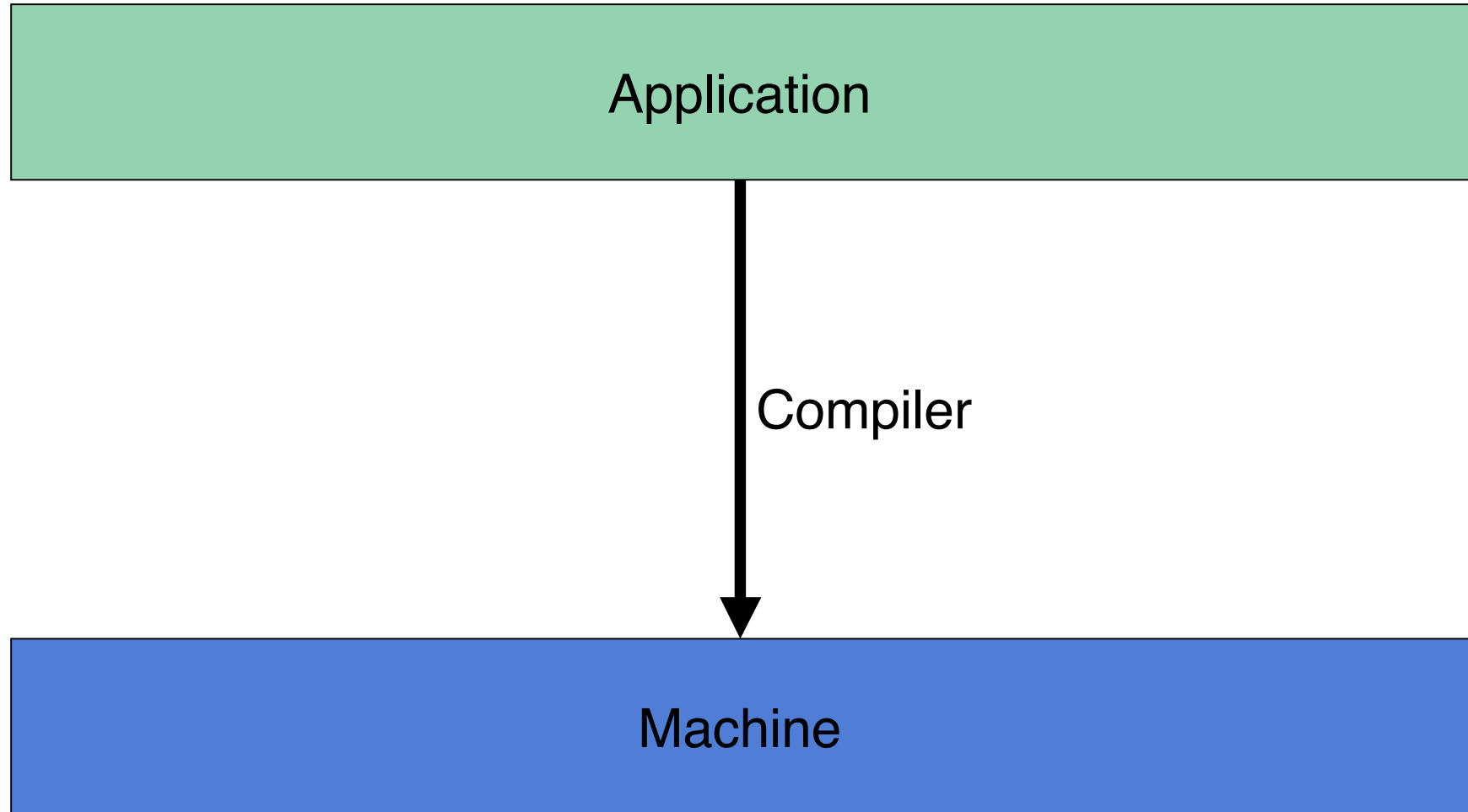(Part of the MoBIES Project continued at Universität Salzburg)

OSEKWorks RTOS, 1 processor (Motorola MPC555 40Mhz)

Future Directions
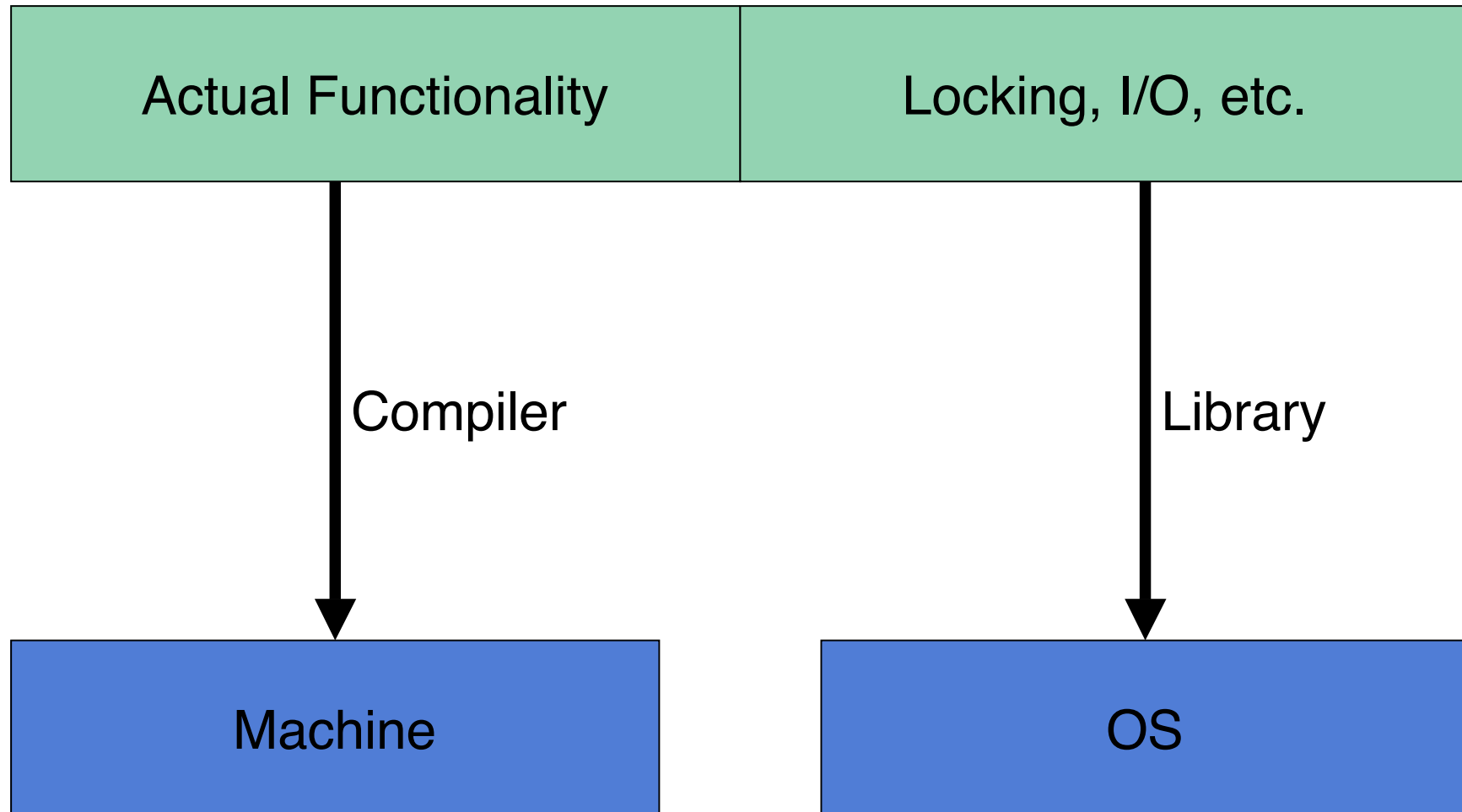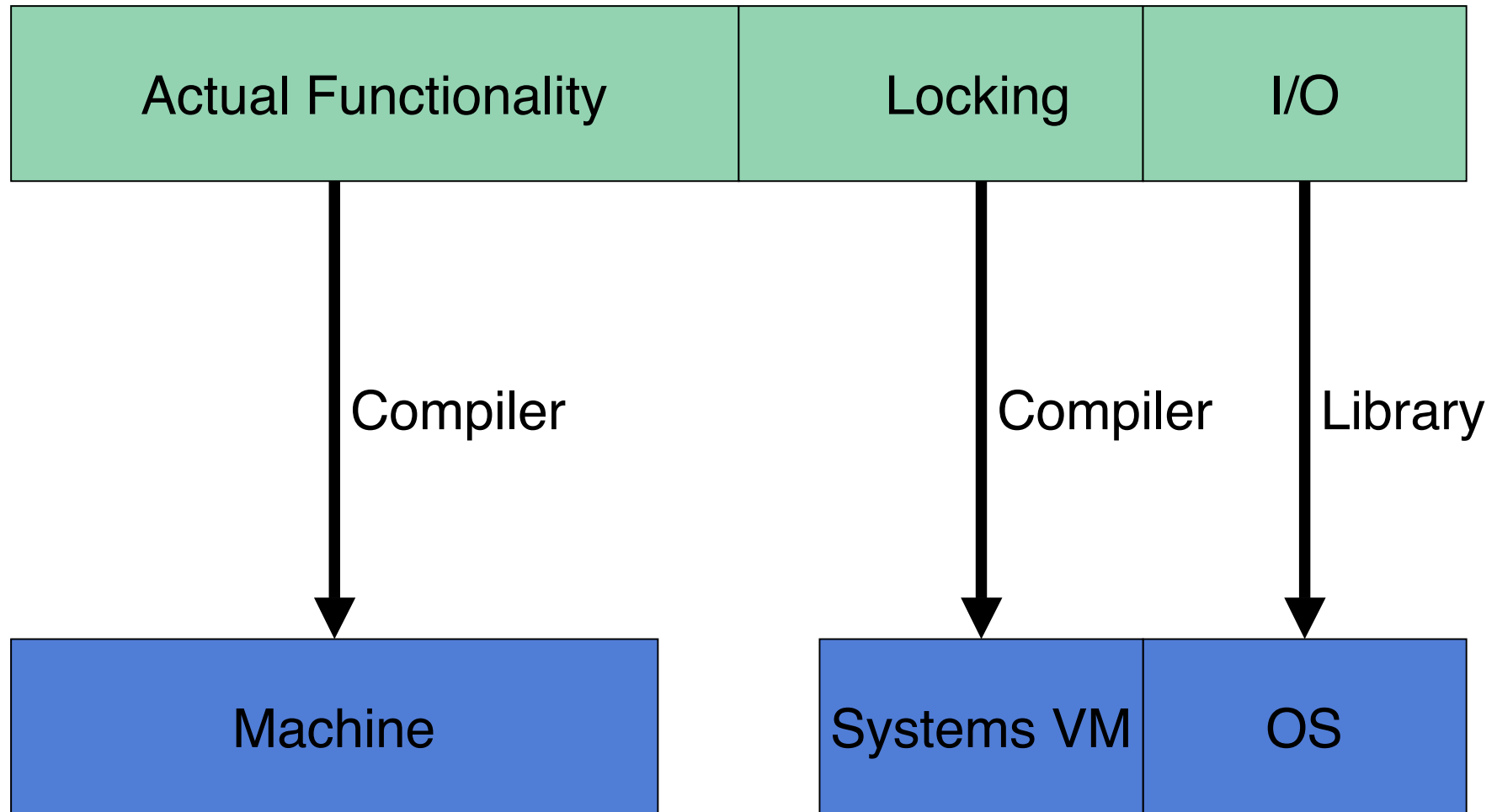
Machine vs. OS

Actual Functionality	Locking, I/O, etc.

Compiler

Library

Machine	OS

"Systems VM"

Actual Functionality	Locking	I/O

Compiler ↓   Compiler ↓   Library ↓

Machine	Systems VM	OS

Non-Preemptive Scheduling

© 2005 C. Kirsch -19-

Compile-time vs. Run-time

Actual Functionality	Locking, I/O, etc.

Compiler

Compiler

Machine	Systems VM