The Embedded Machine

Predictable, Portable Real-Time Code

PLDI 2002

Thomas A. Henzinger, Christoph M. Kirsch

UC Berkeley

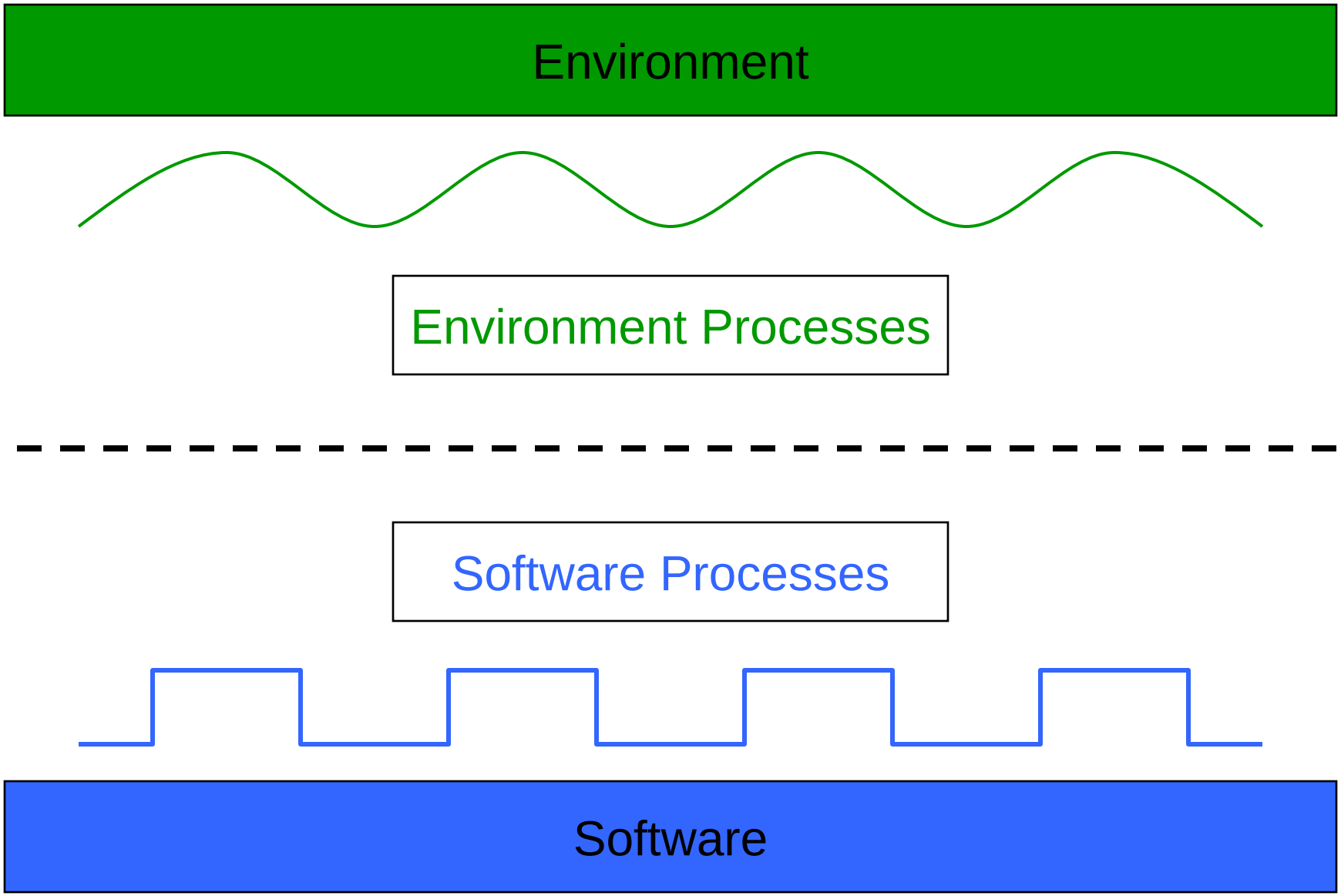
www.eecs.berkeley.edu/~cm

Does It Fly?

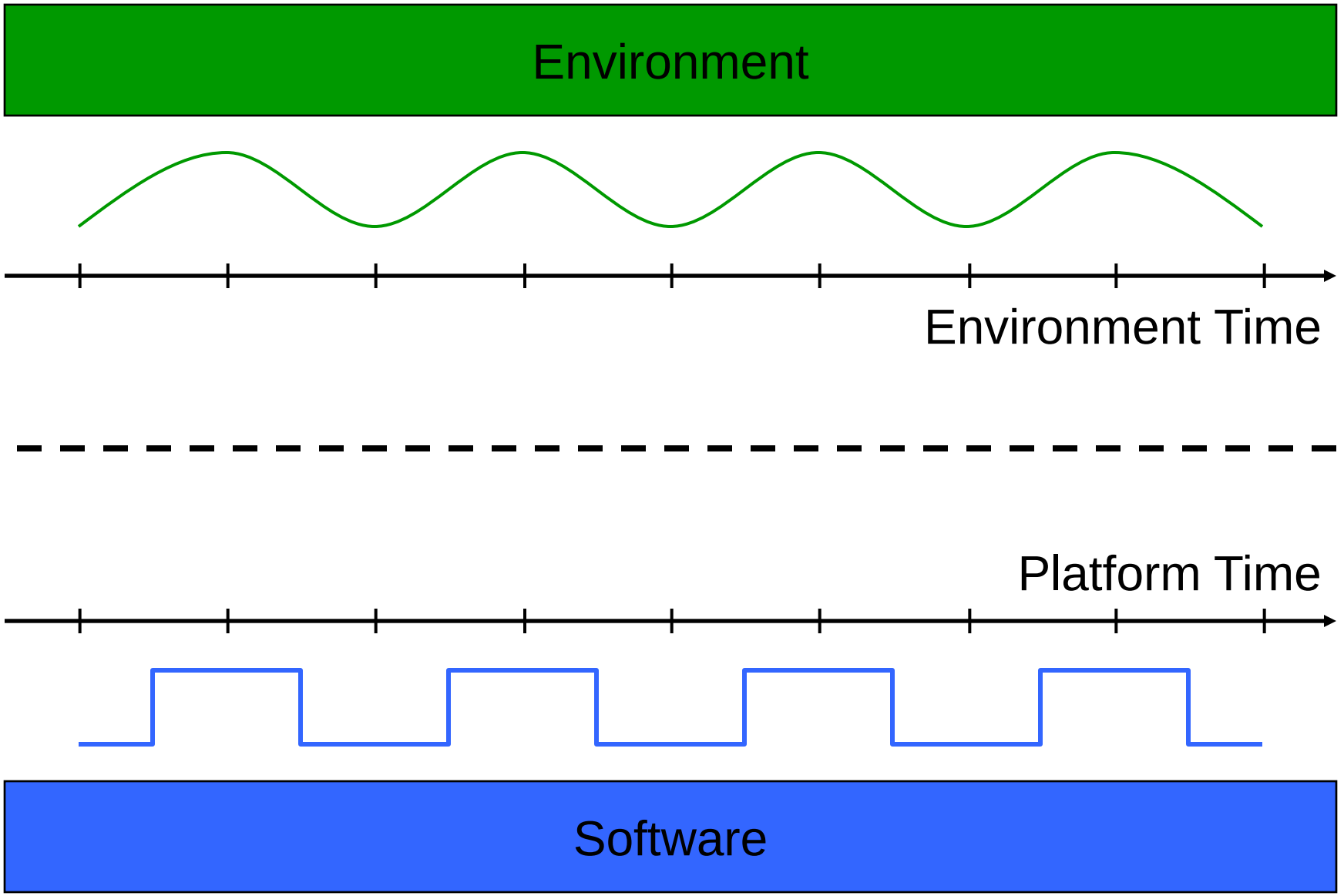
6 degrees of freedom, 3 processors

Does It Drive? (By-Wire)

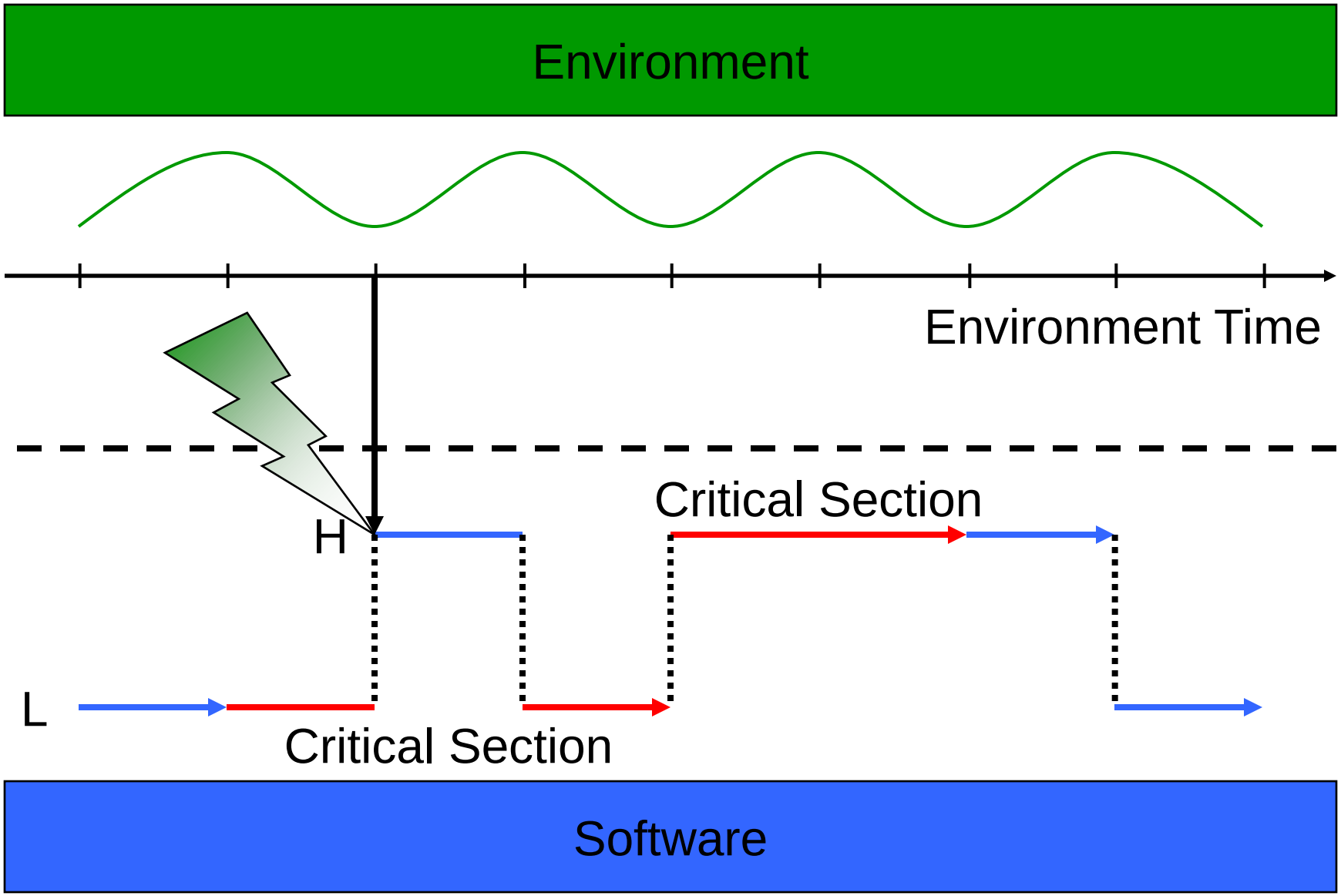
Embedded Software



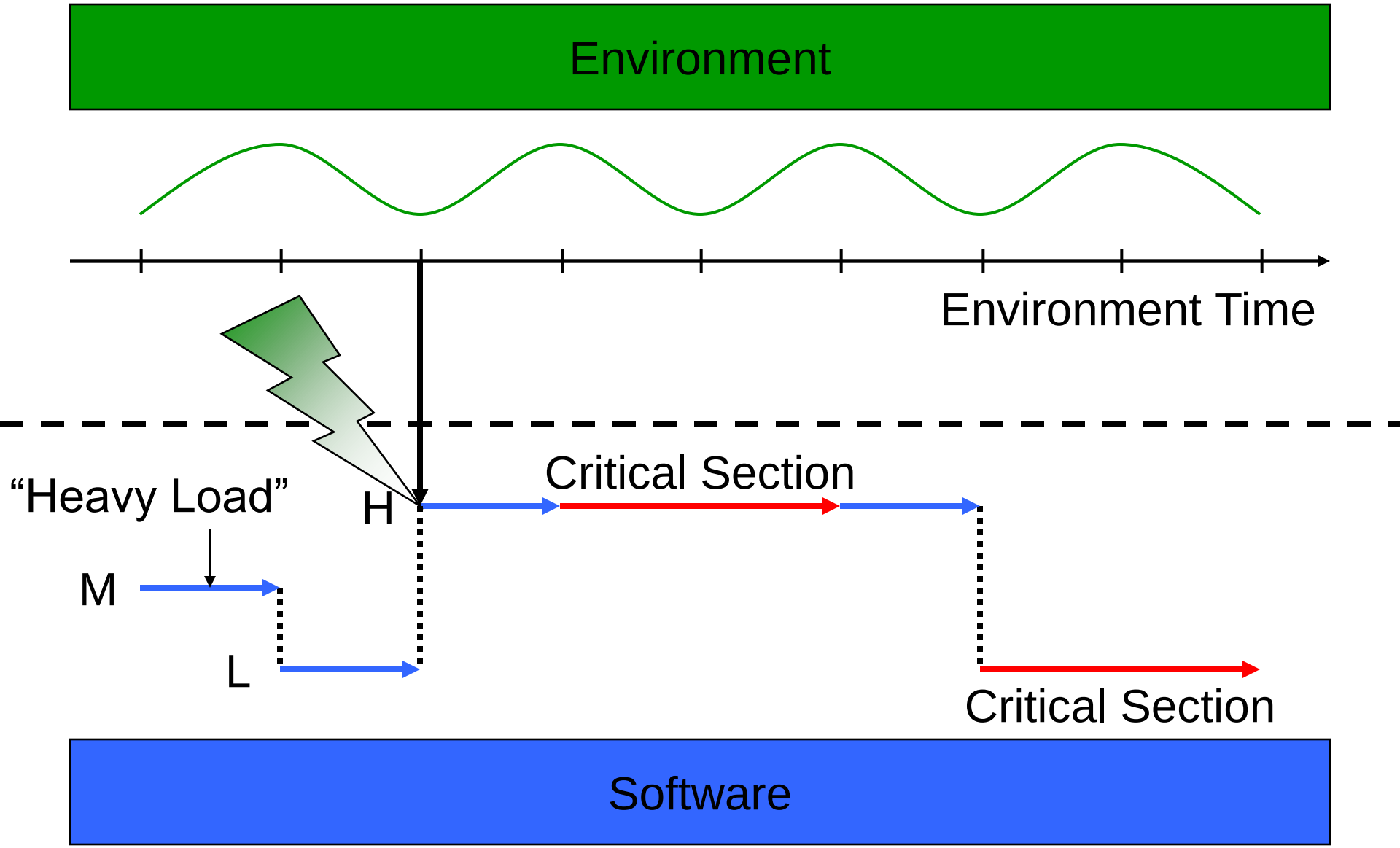
Environment vs. Platform Time



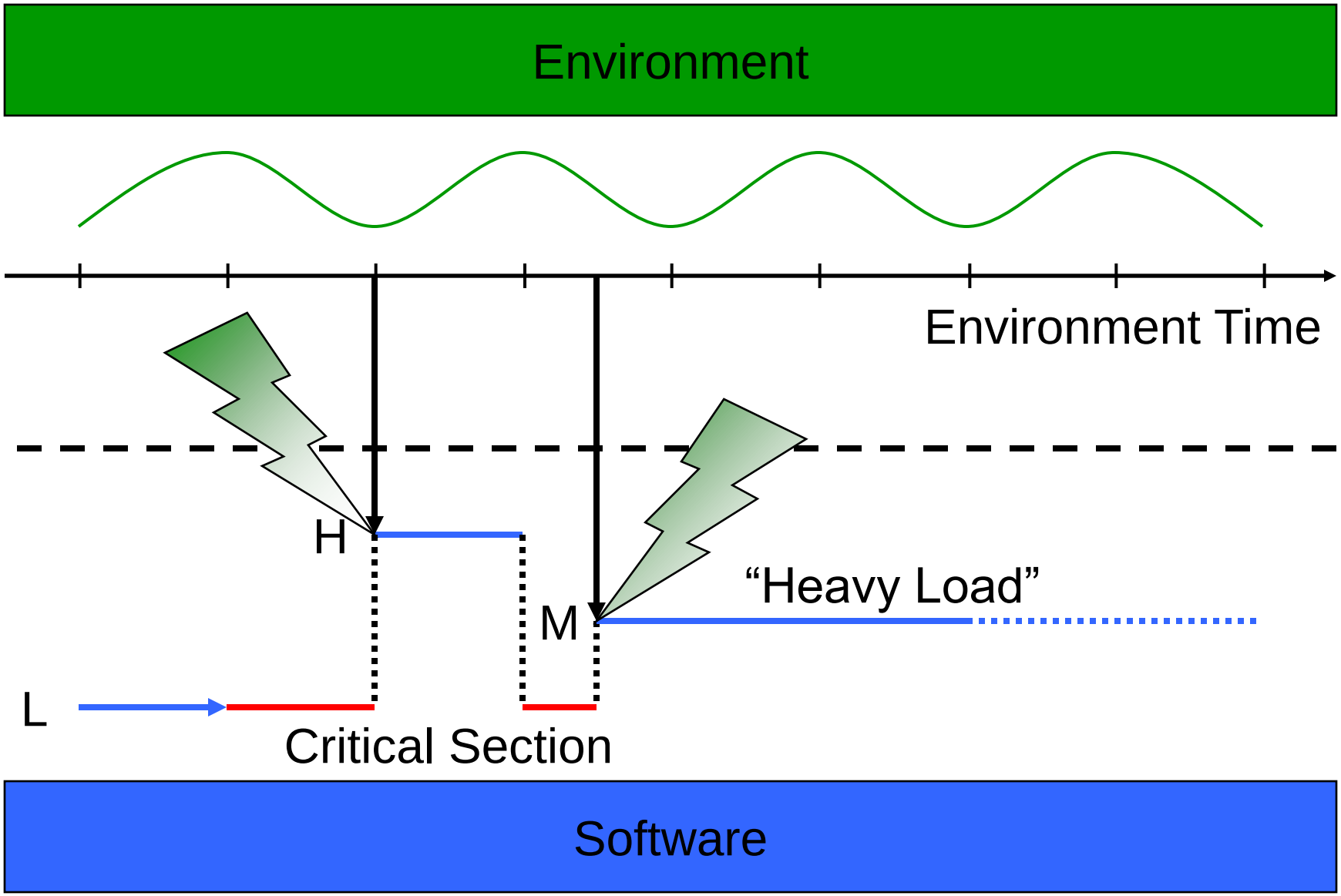
Problem: Non-Deterministic Behavior…



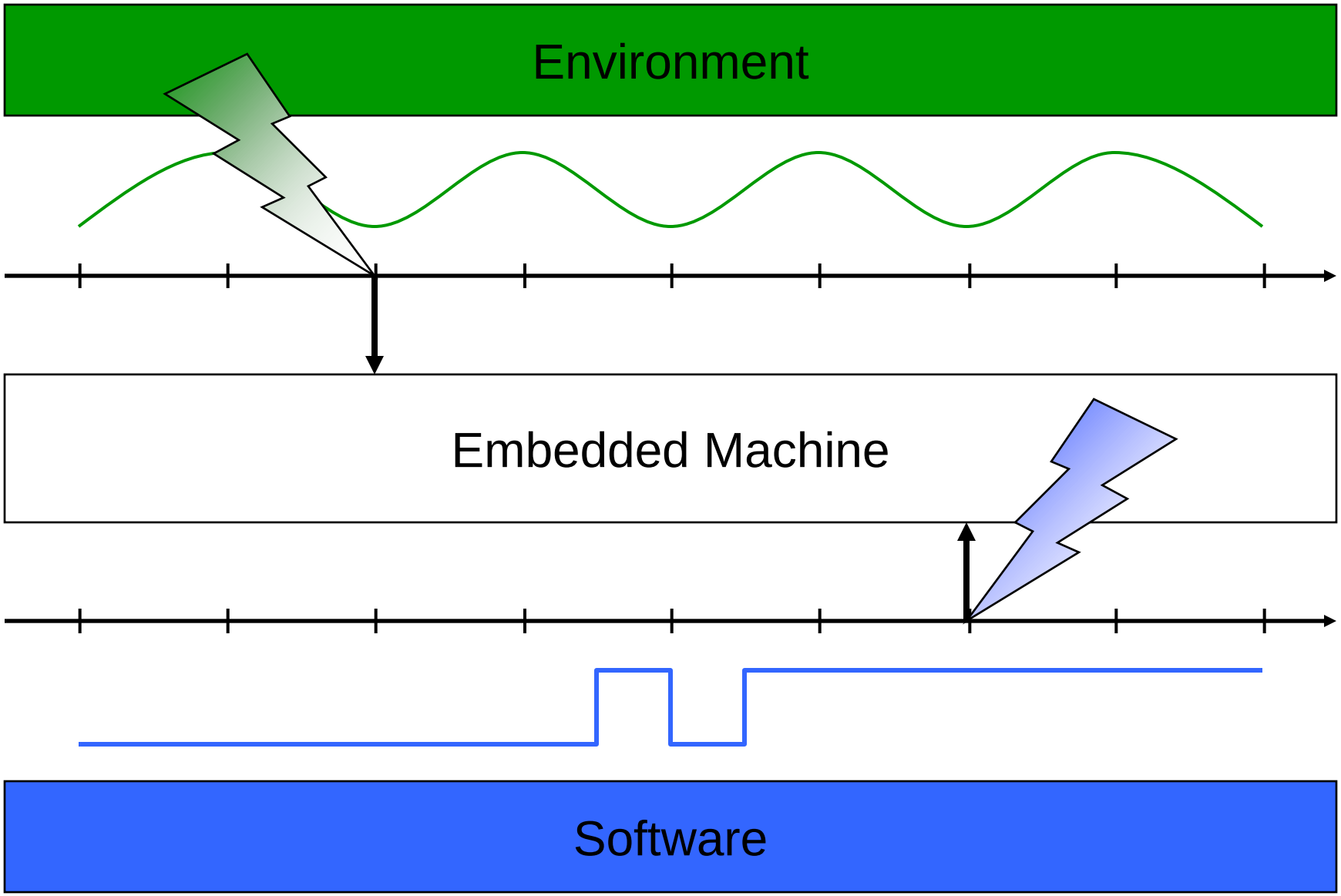
…because of Race Conditions



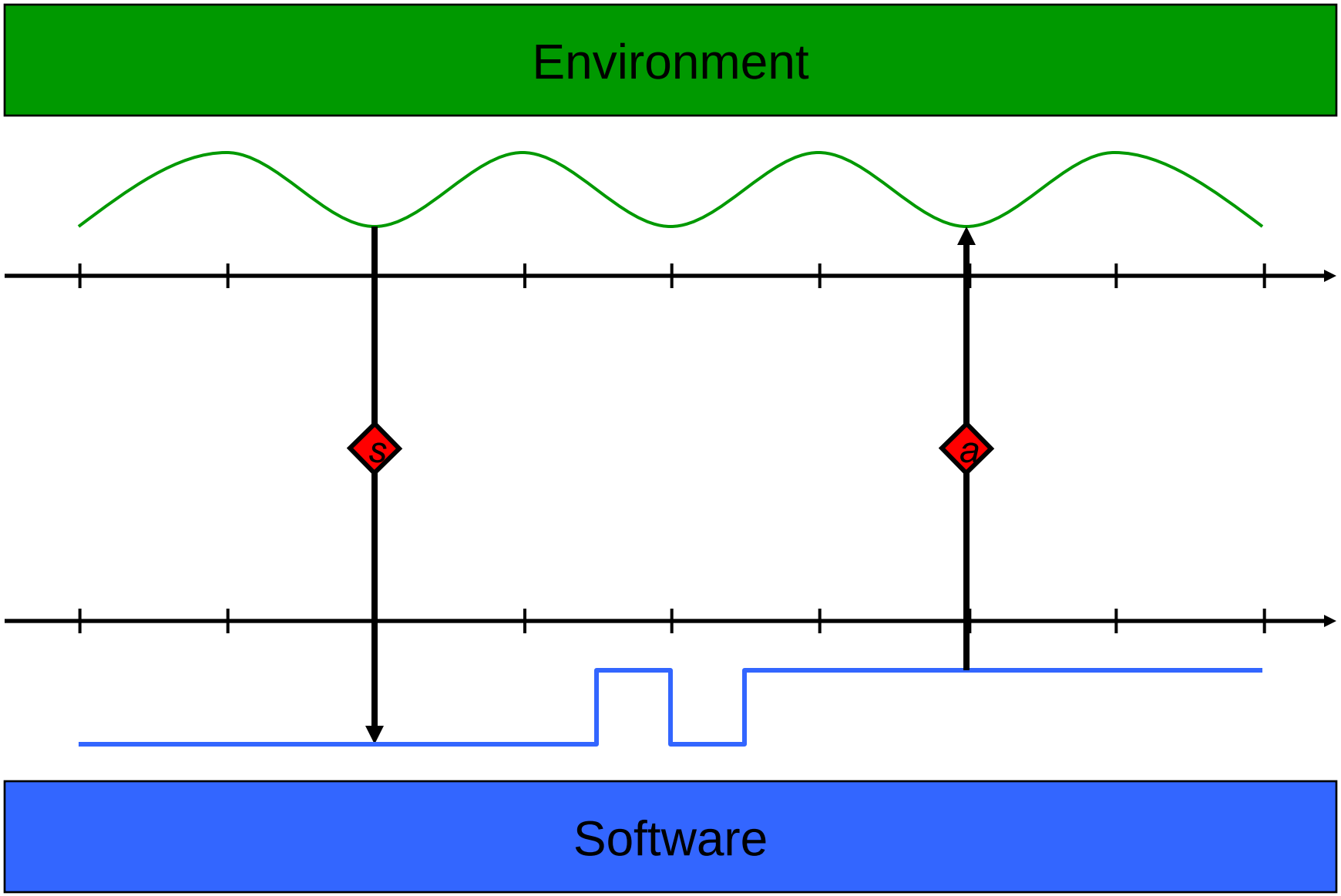
Priority Inversion



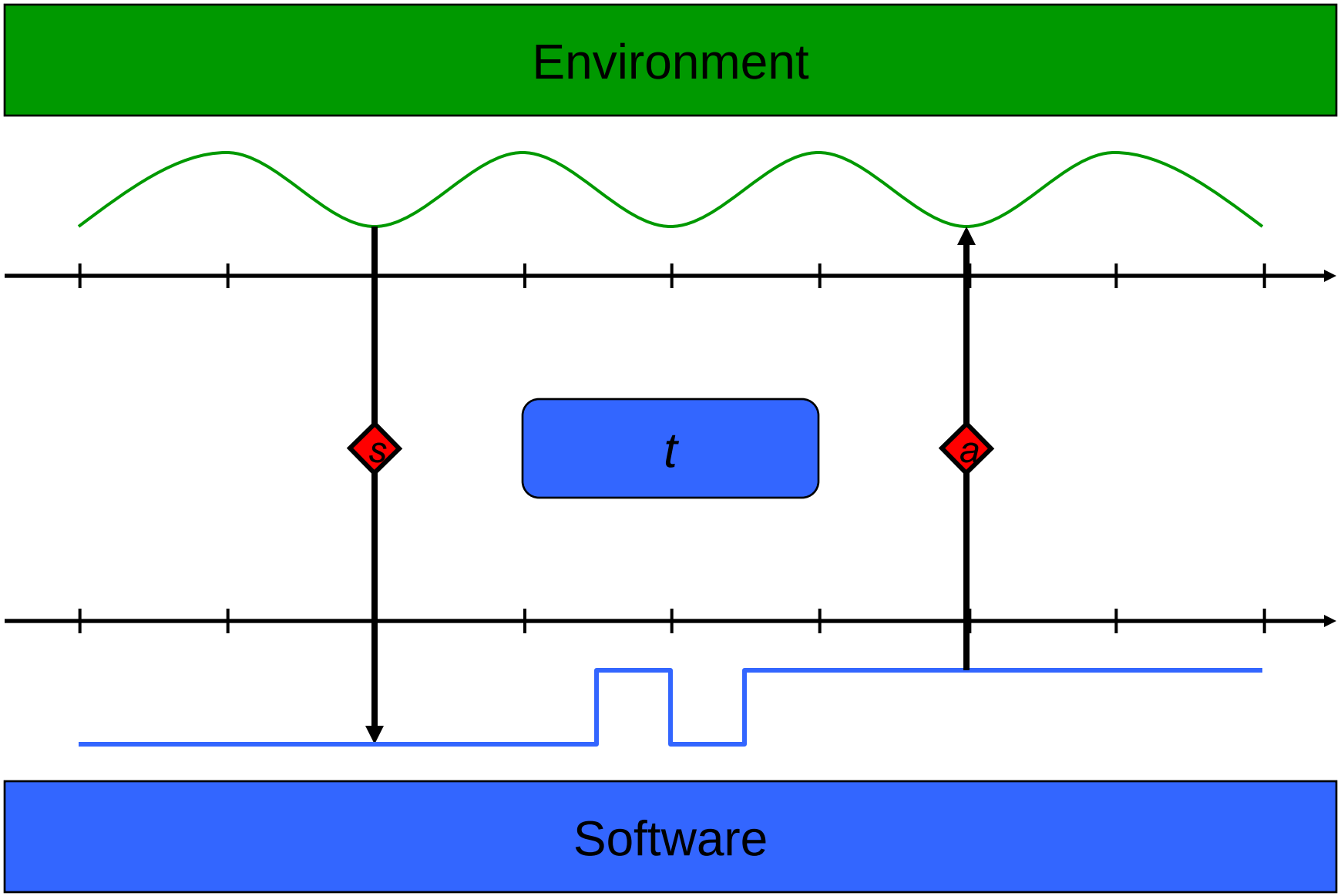
The Embedded Machine



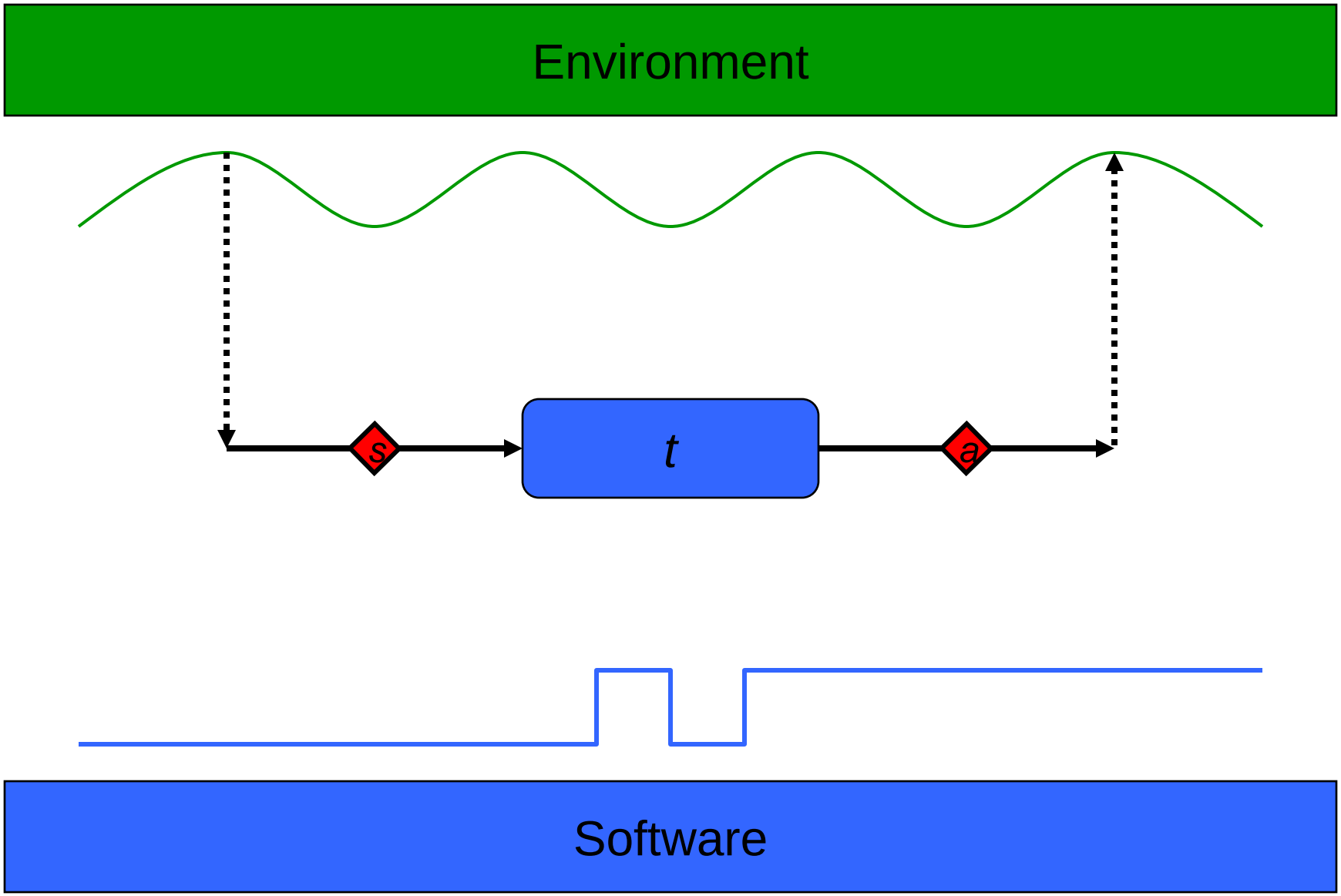
I/O: Drivers



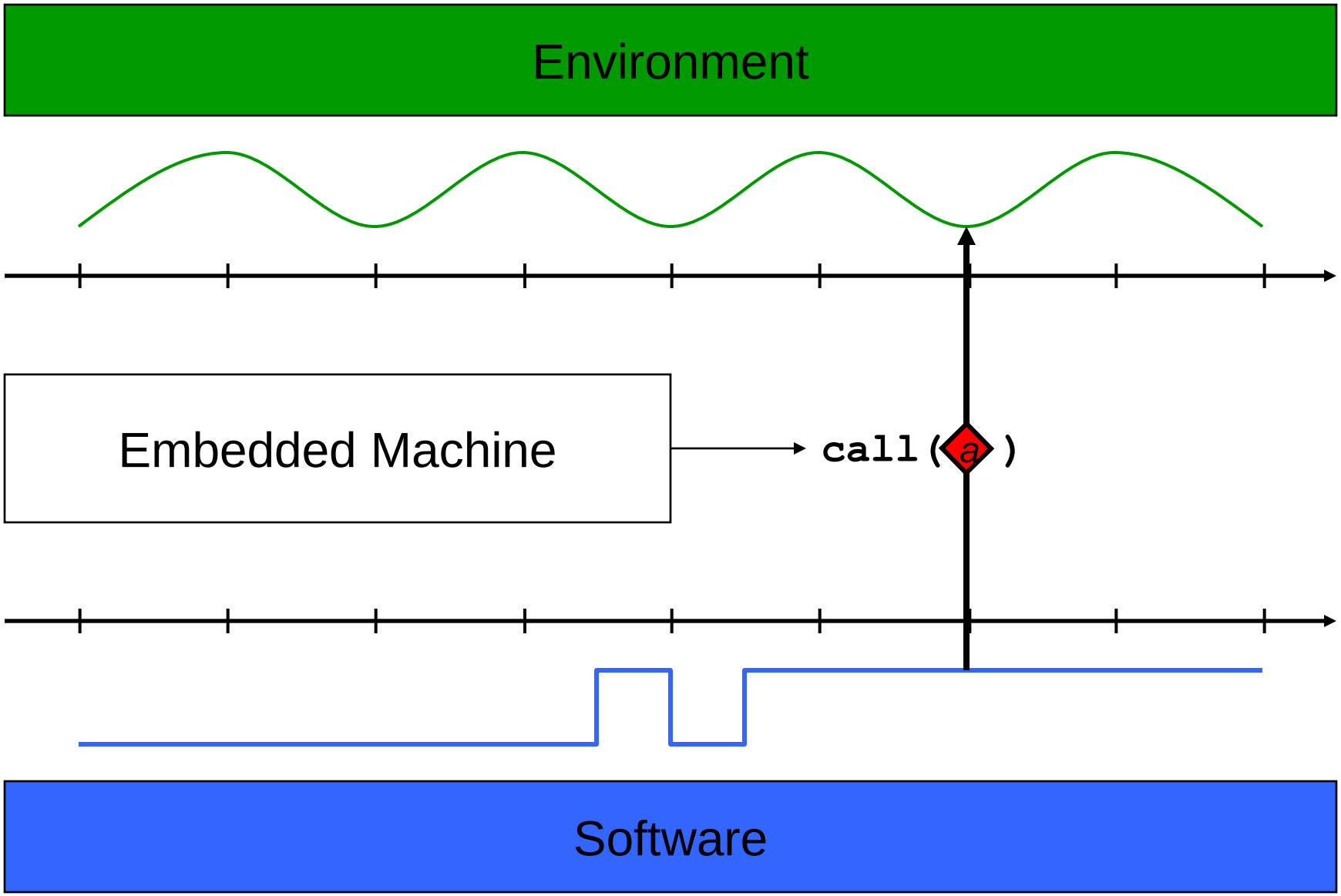
Computation: Tasks



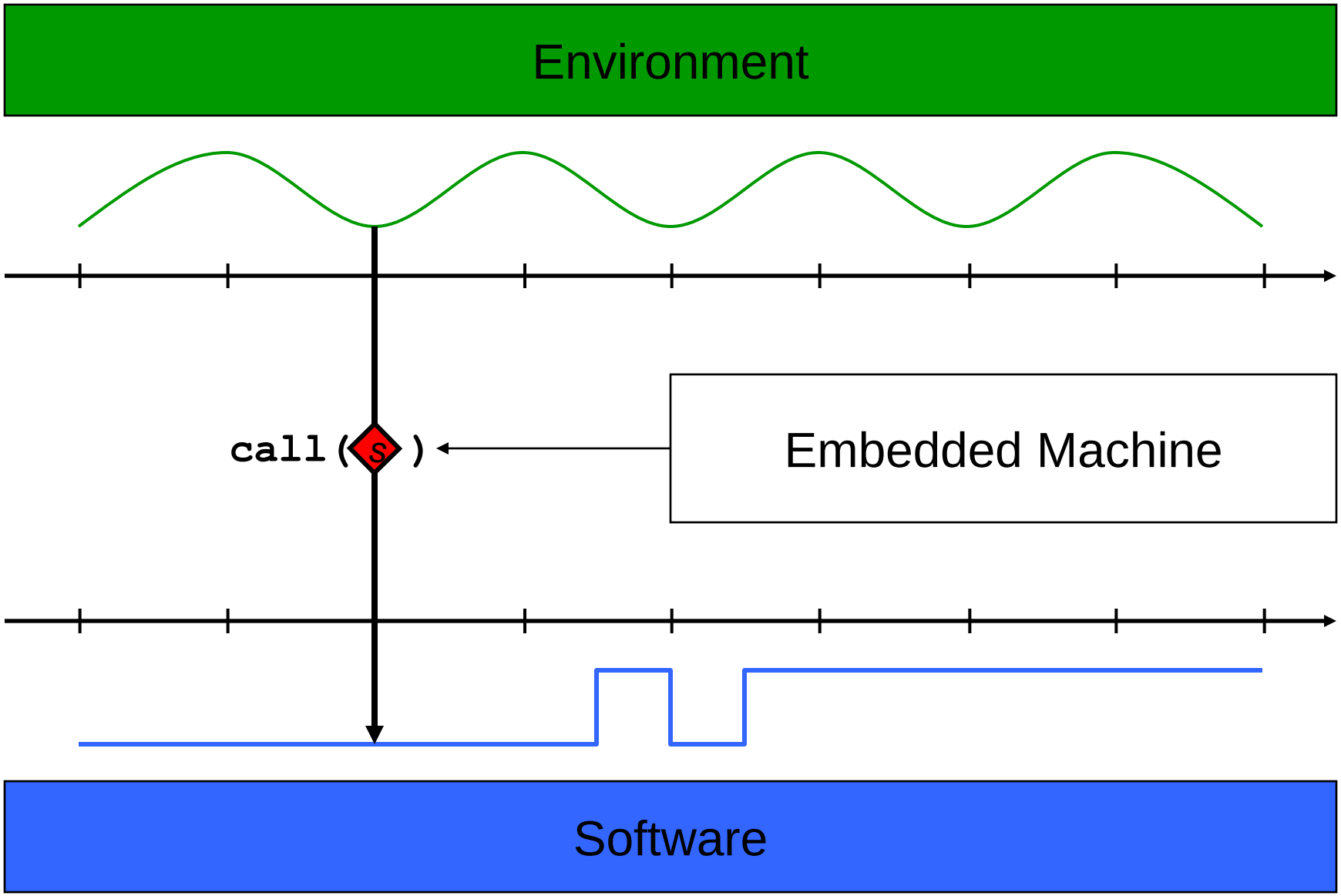
Flow of Data



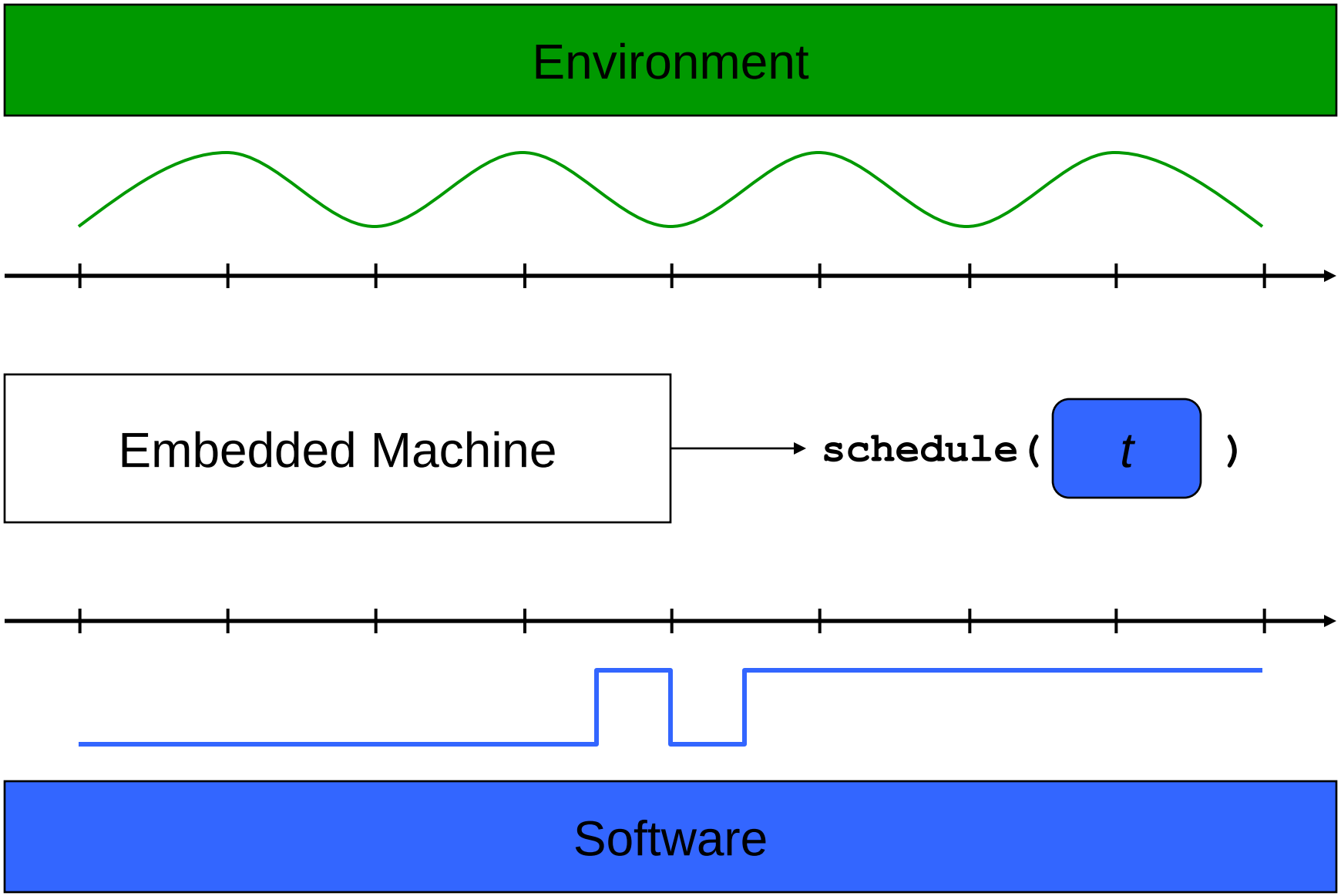
Instruction: `call(`*a*`)`



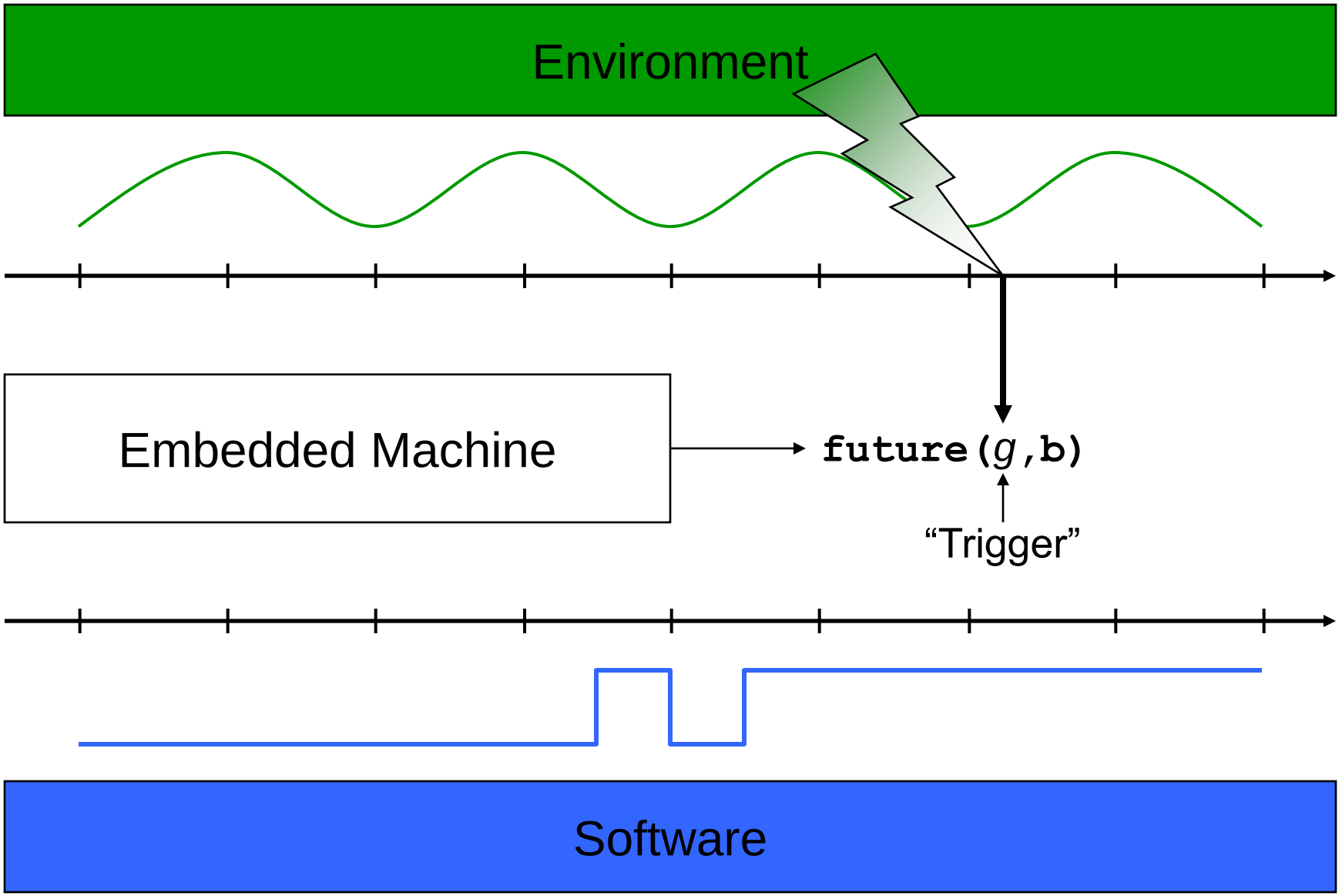
Instruction: `call(`*s*`)`



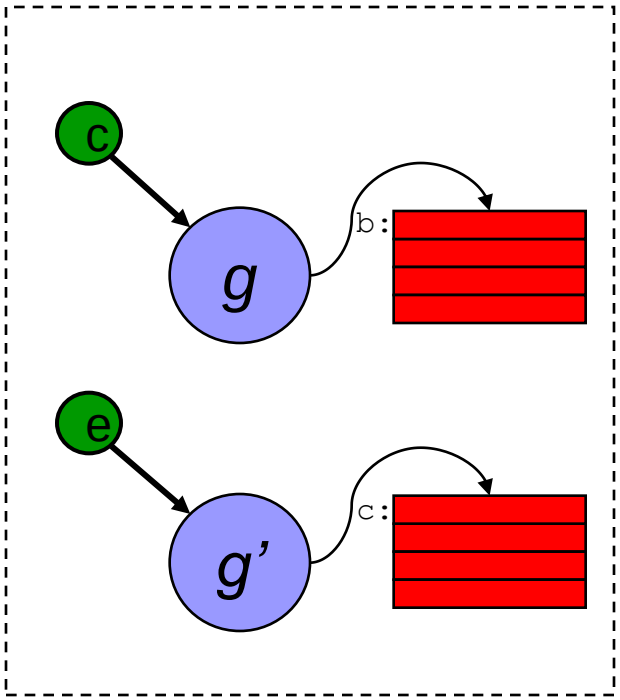
Instruction: `schedule(`*t*`)`



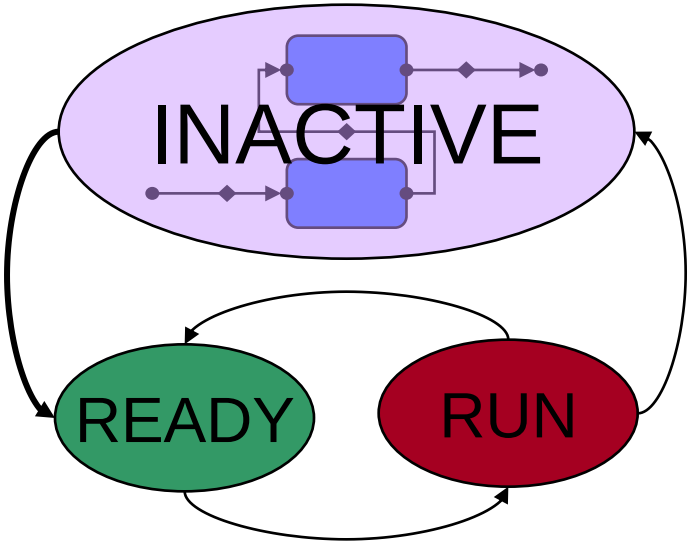
Instruction: `future(g,b)`



Synchronous vs. Scheduled Computation



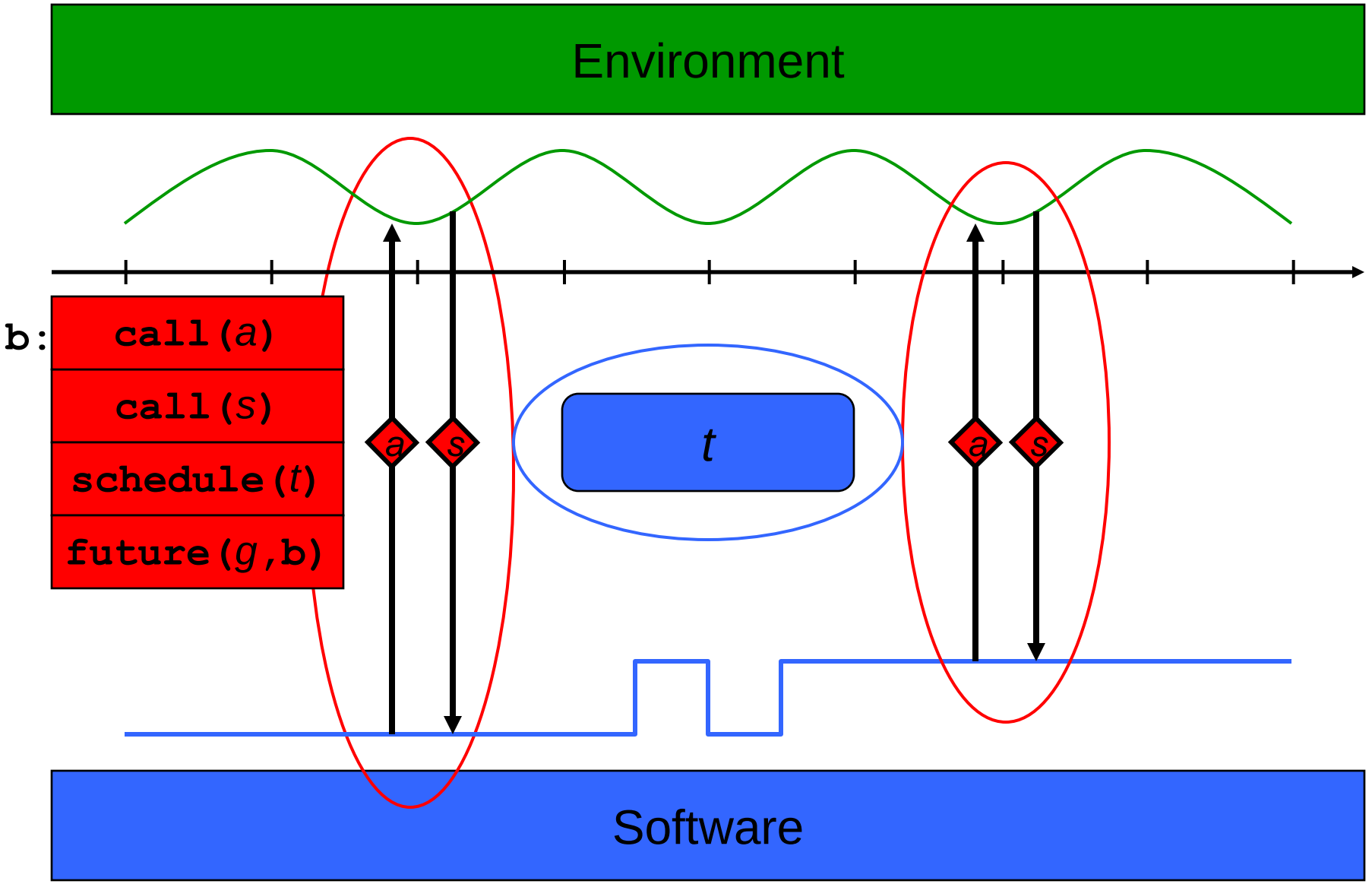
activates



- Synchronous computation
- Kernel context
- Trigger related interrupts disabled

- Scheduled computation
- User context

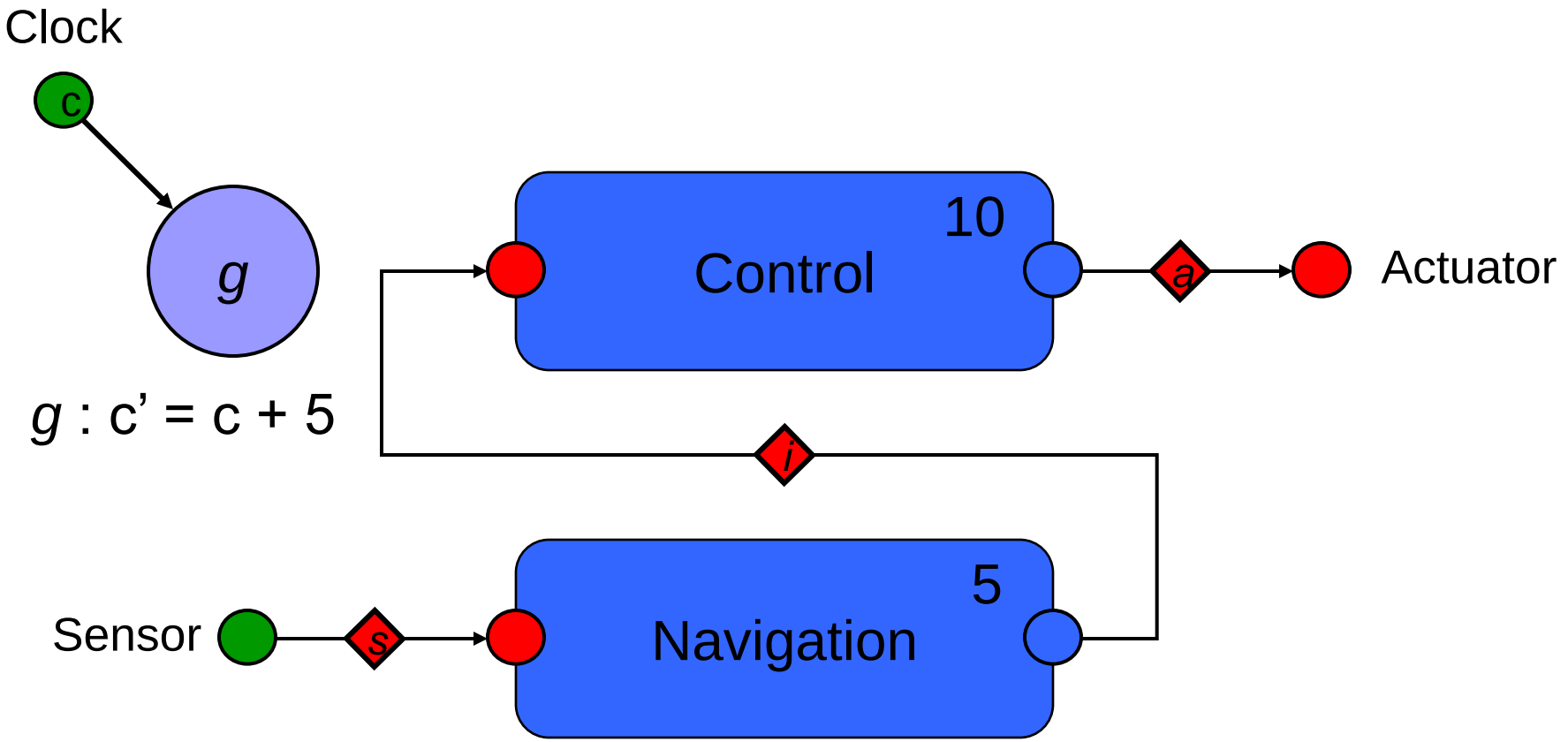
Synchronous vs. Scheduled Computation



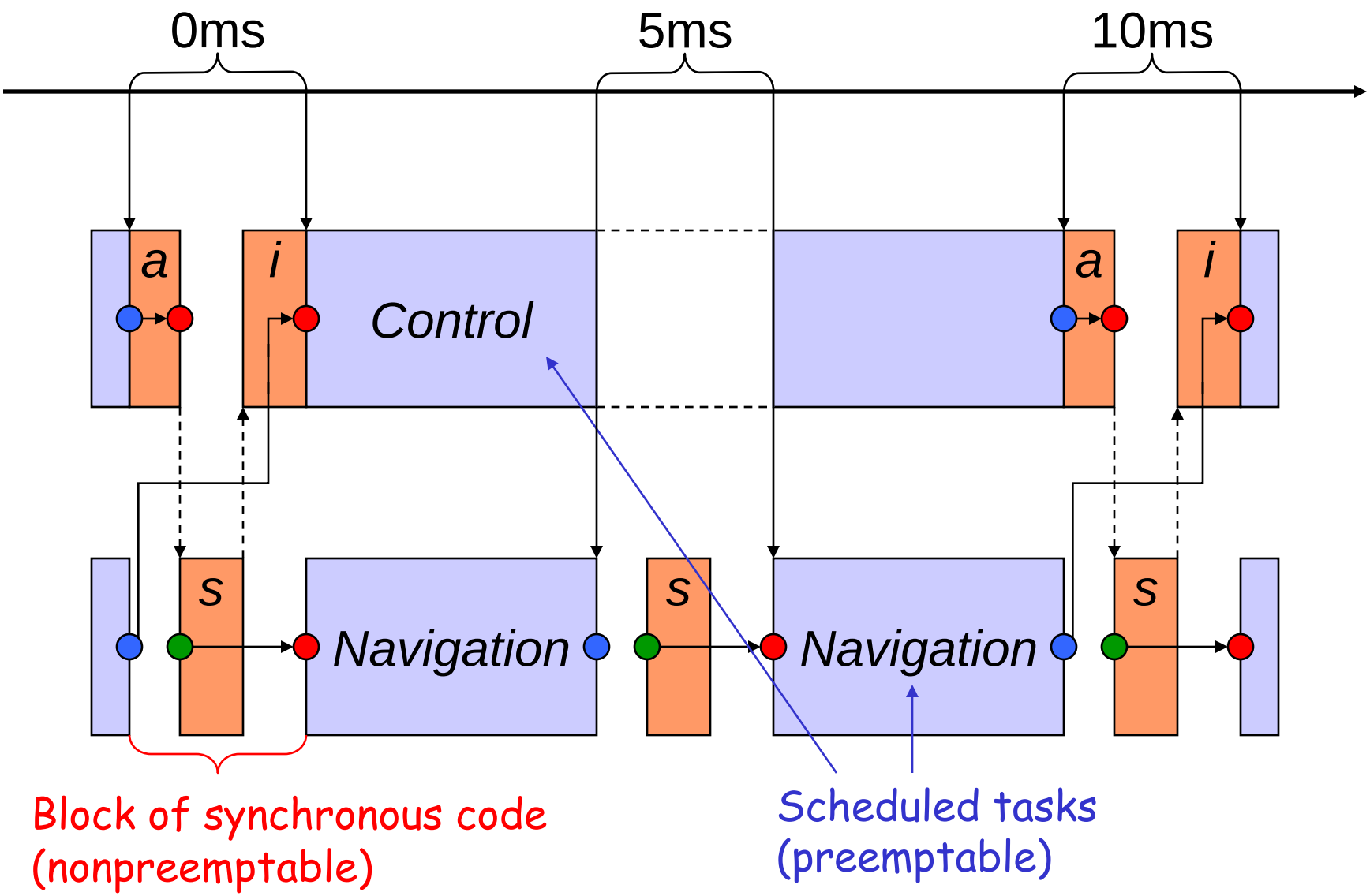
The ETH Zürich Helicopter

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

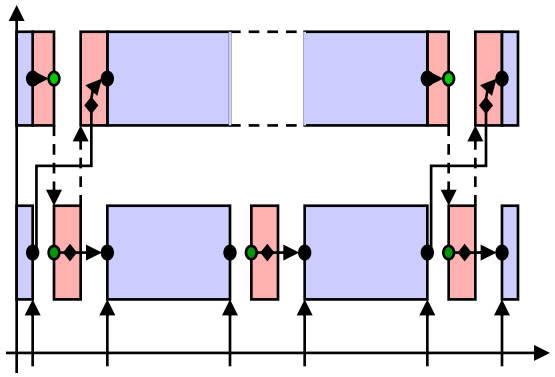
Helicopter Control Software



Environment Timeline



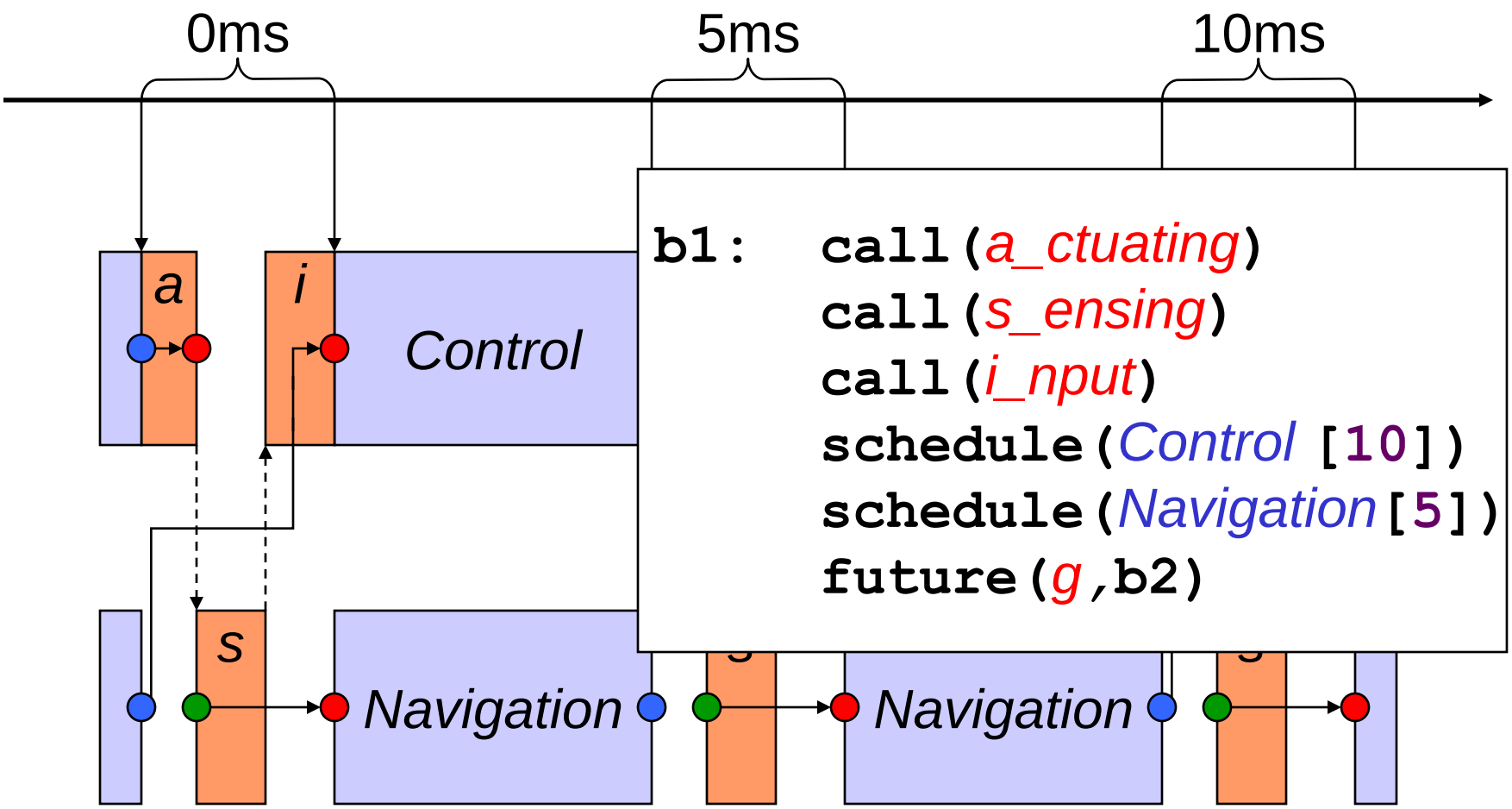
The Giotto Program



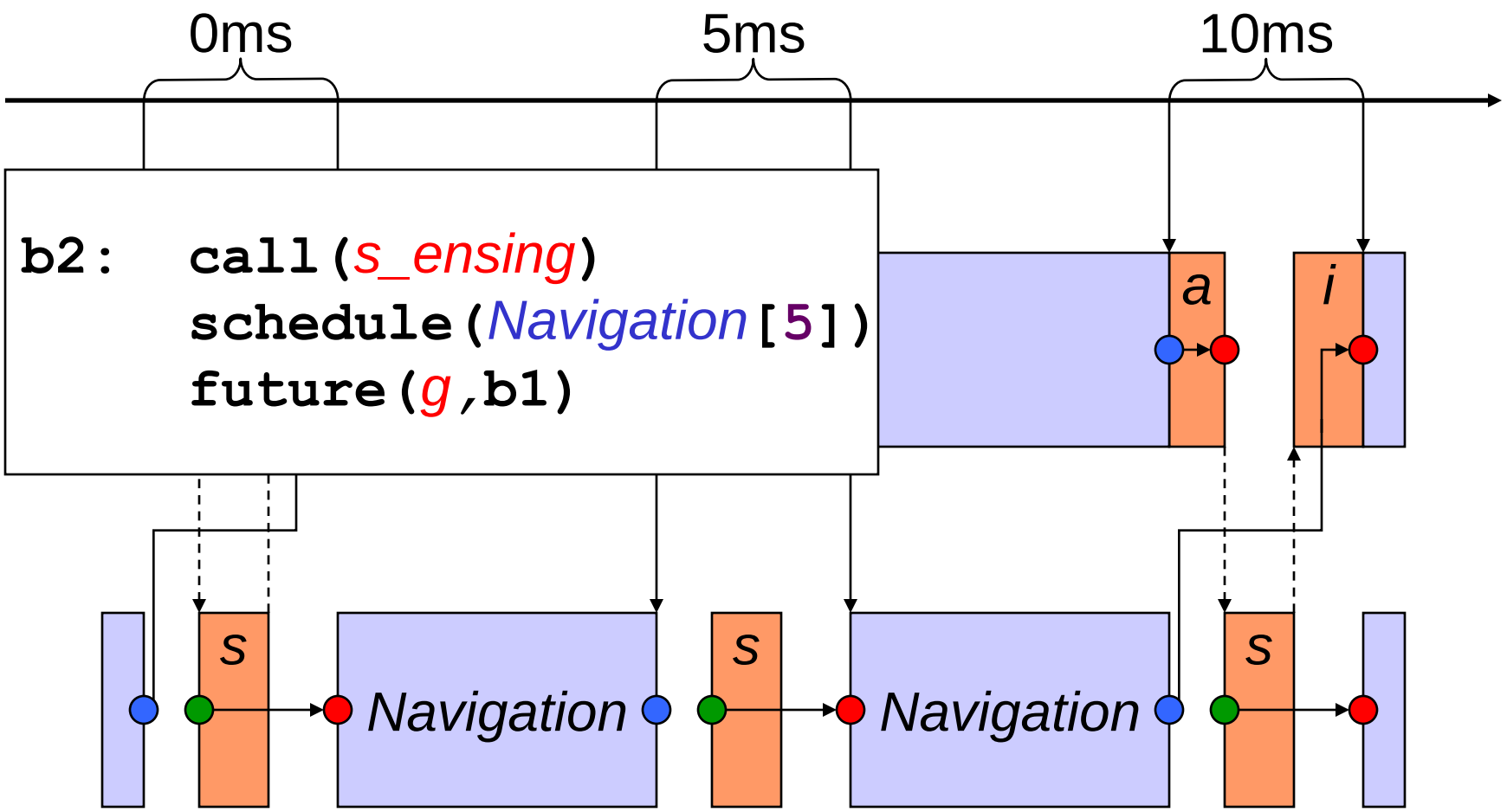
```
…
mode Flight ( ) period 10ms
  {
      actfreq 1 do Actuator ( a_ctuating ) ;

      taskfreq 1 do Control ( i_nput ) ;
      taskfreq 2 do Navigation ( s_ensing ) ;
  }
…
```

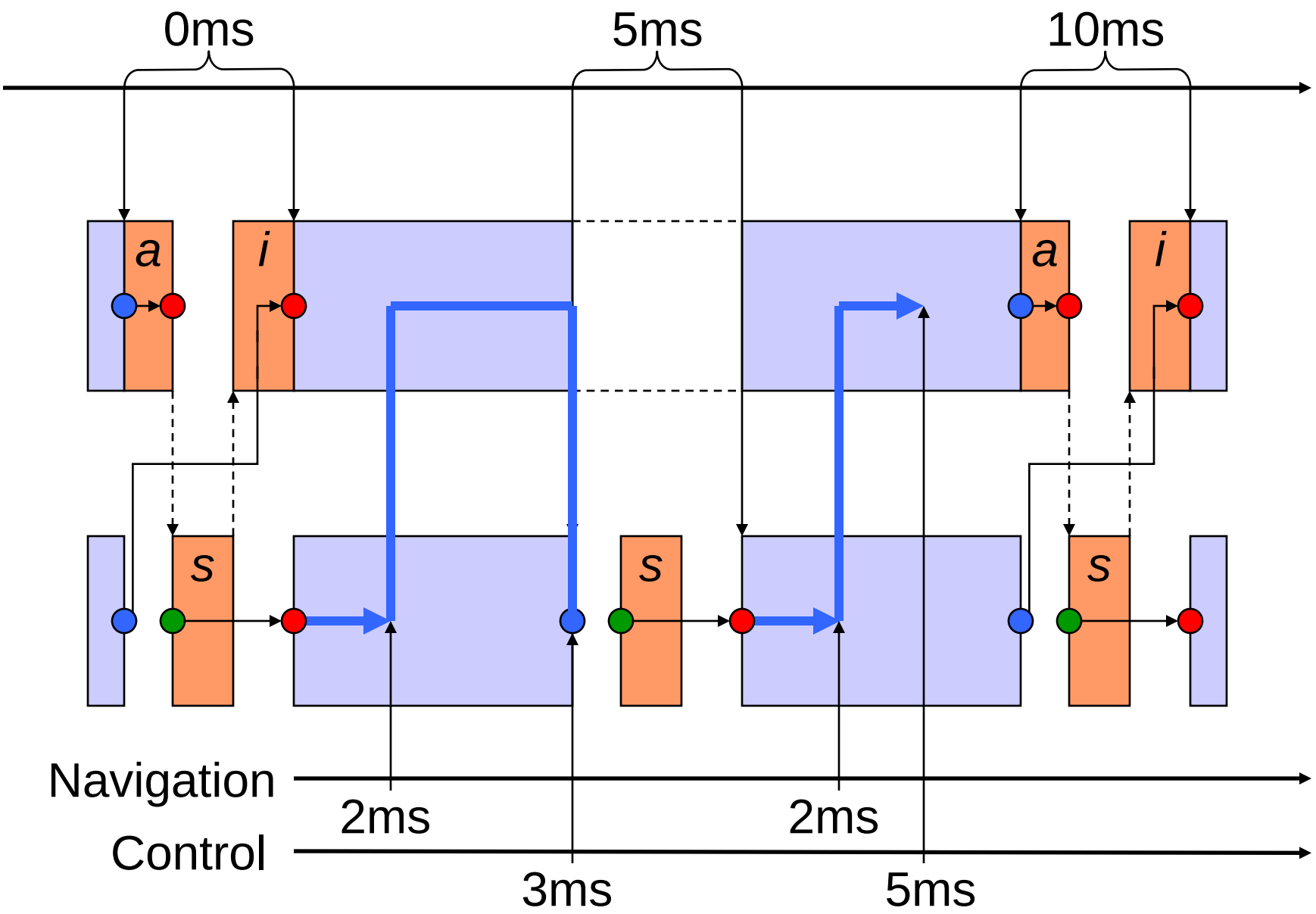
E Code


0ms
5ms
10ms
a
i
Control
s
Navigation
Navigation
b1: call(a_ctuating)
call(s_ensing)
call(i_nput)
schedule(Control [10])
schedule(Navigation[5])
future(g,b2)

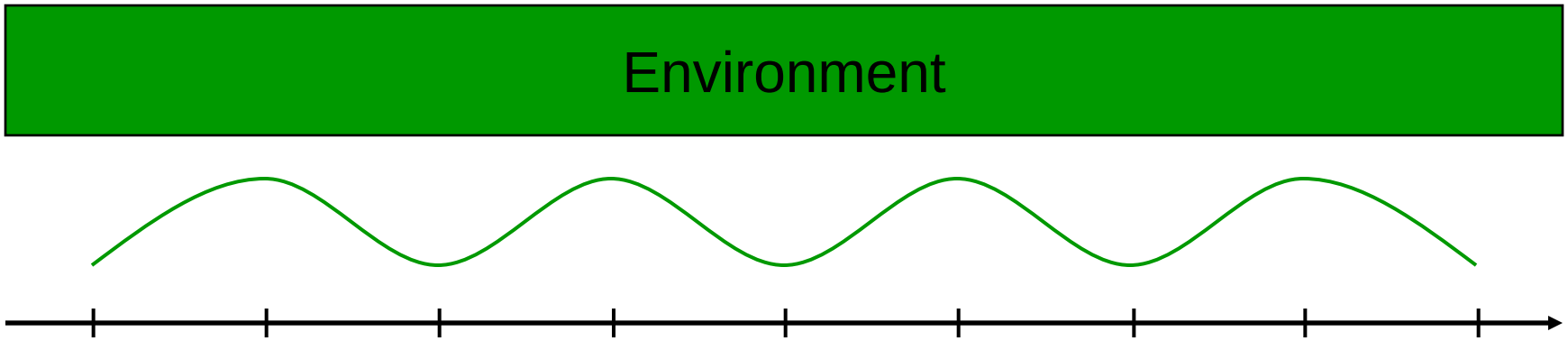
E Code


0ms
5ms
10ms
b2: call(s_ensing)
schedule(Navigation[5])
future(g,b1)
a
i
s
Navigation
s
Navigation
s

Platform Timeline: EDF

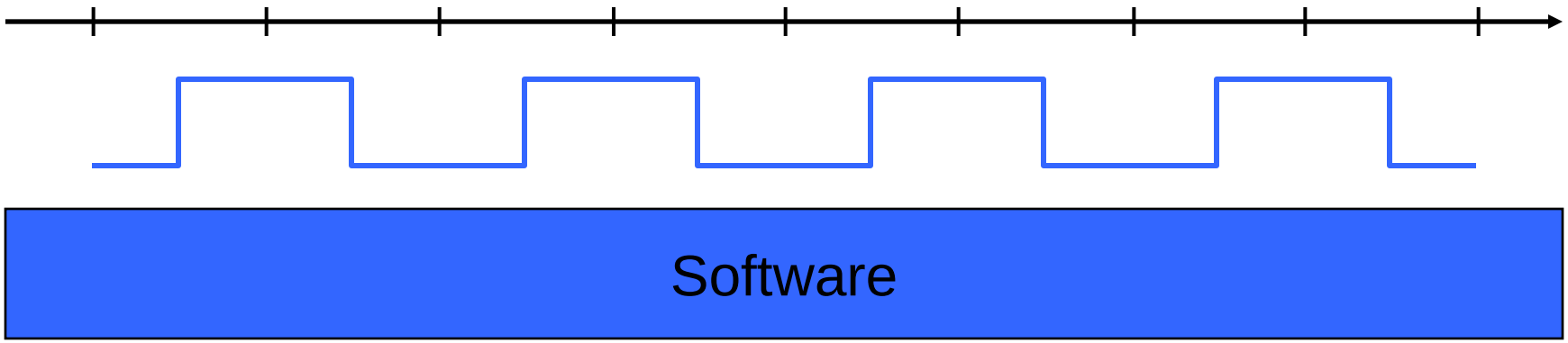


Platform Time is Platform Memory

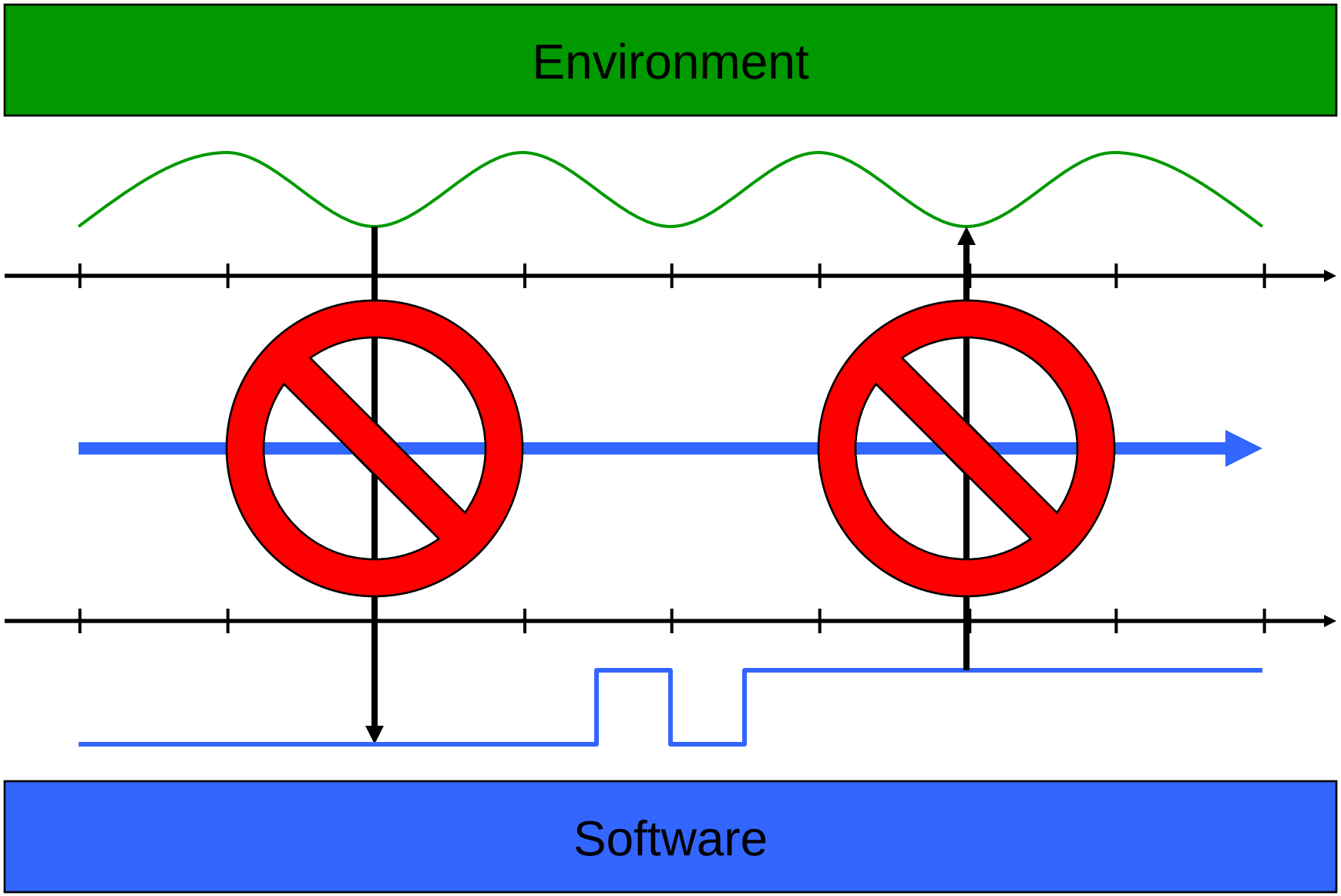


- Programming as if there is enough platform time

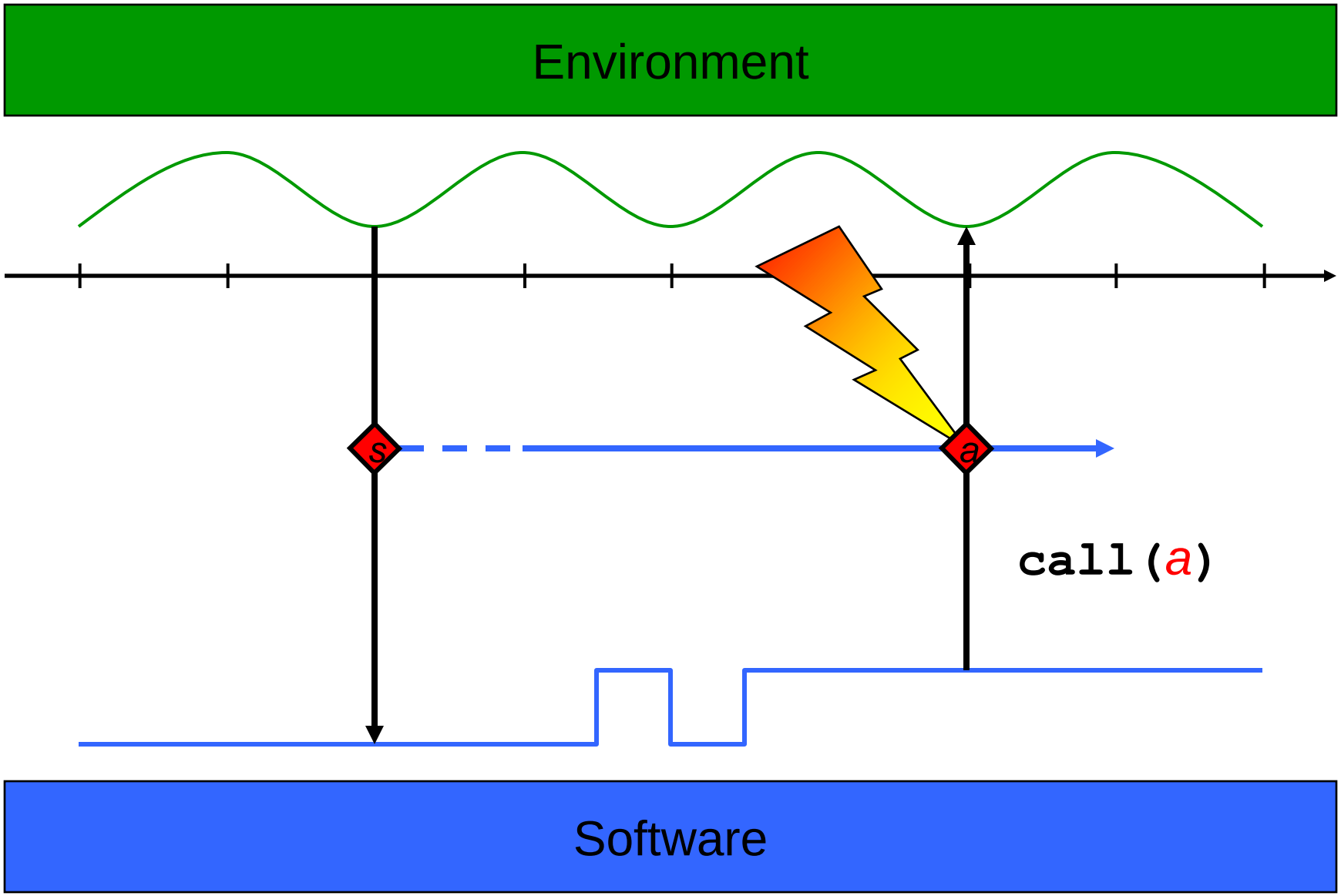
- Implementation checks whether there is enough of it



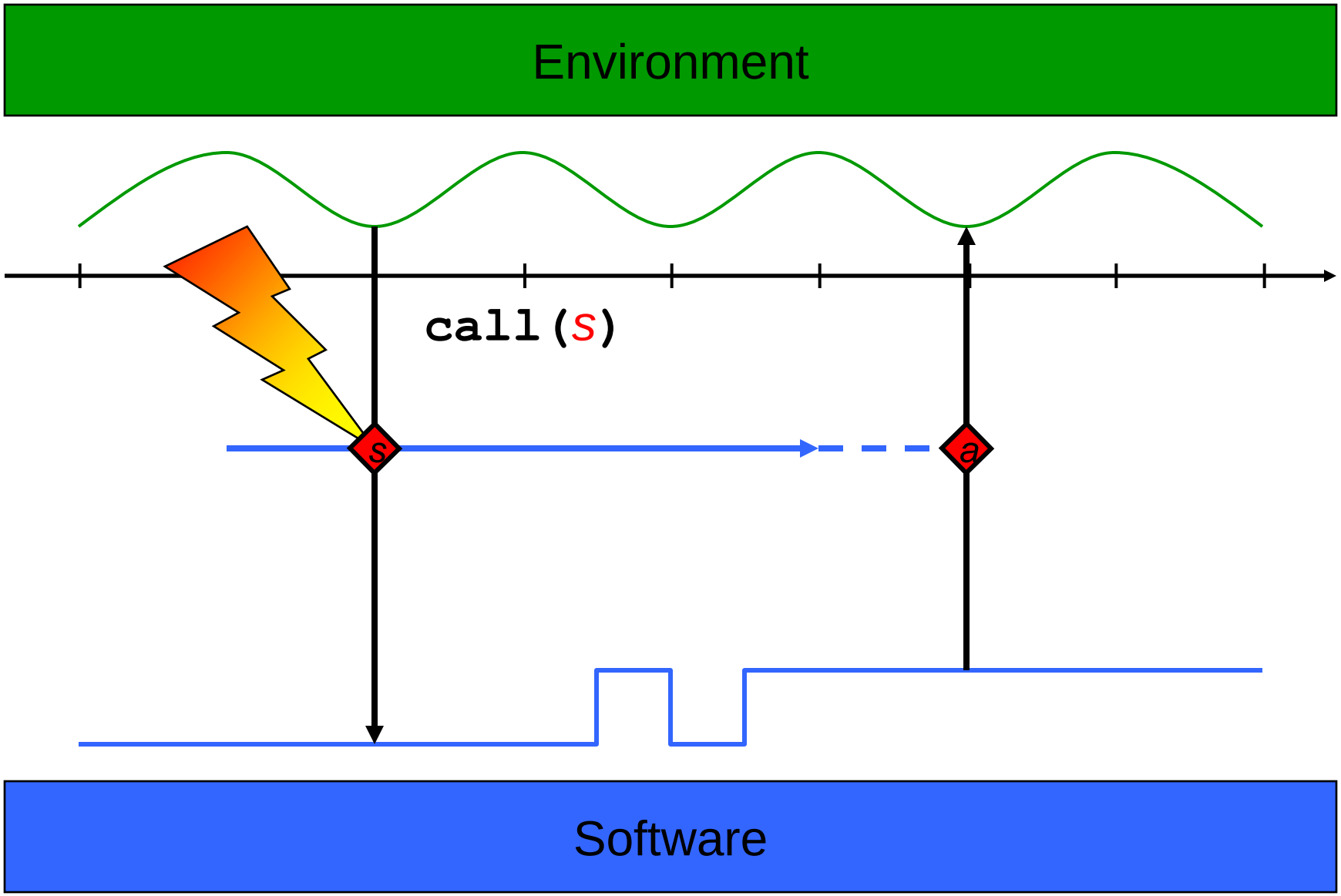
Time Safety



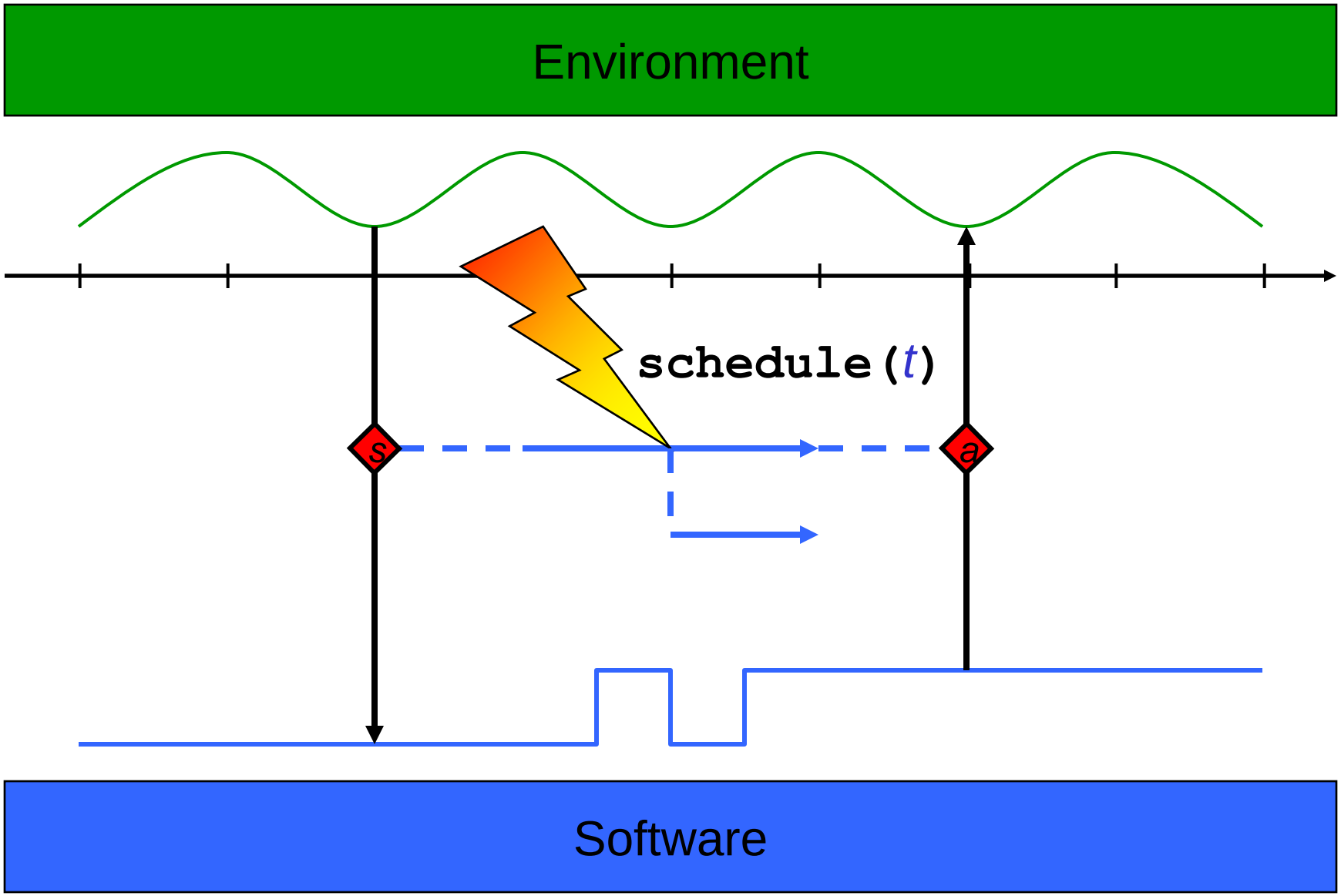
Runtime Exceptions I



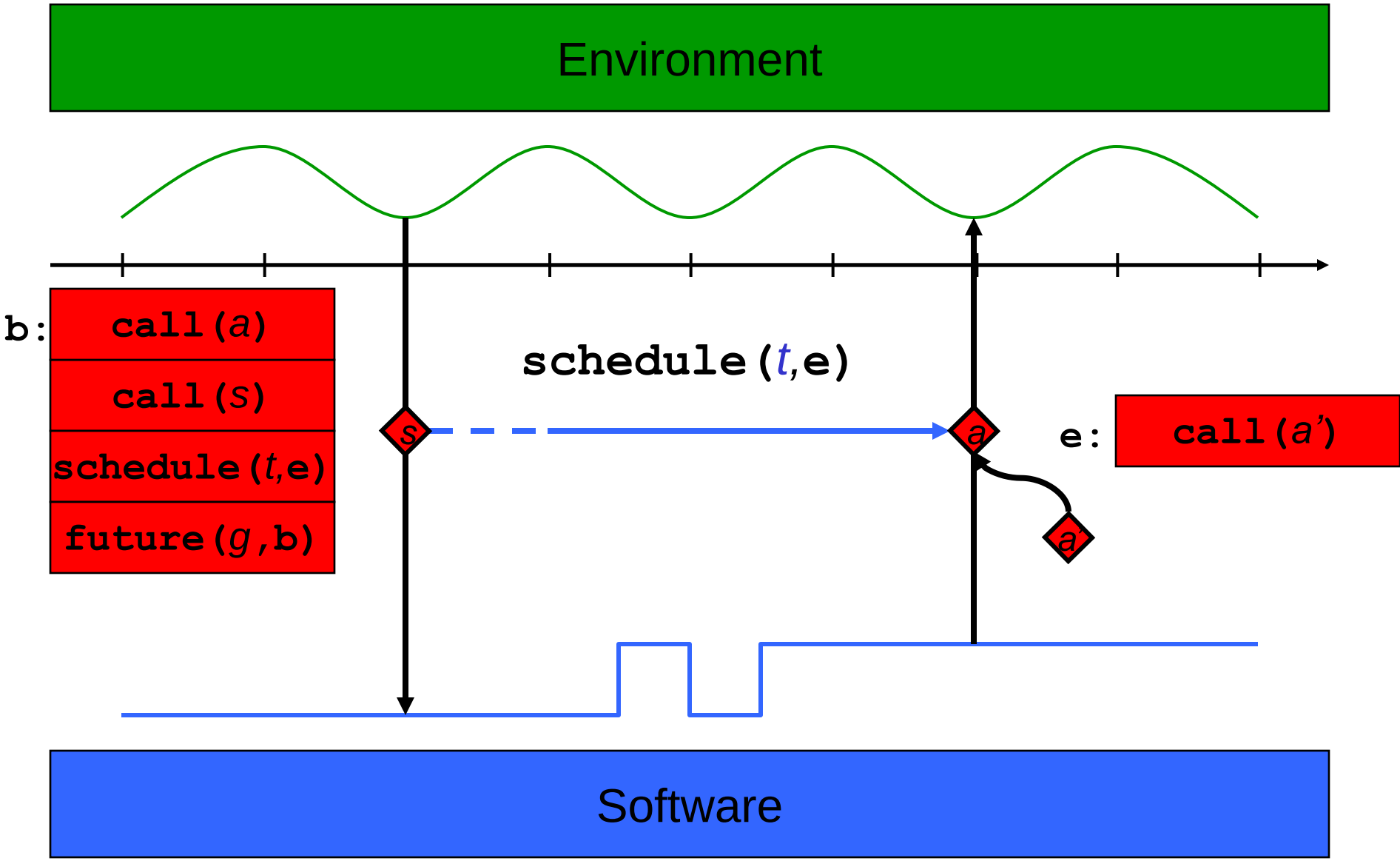
Runtime Exceptions II



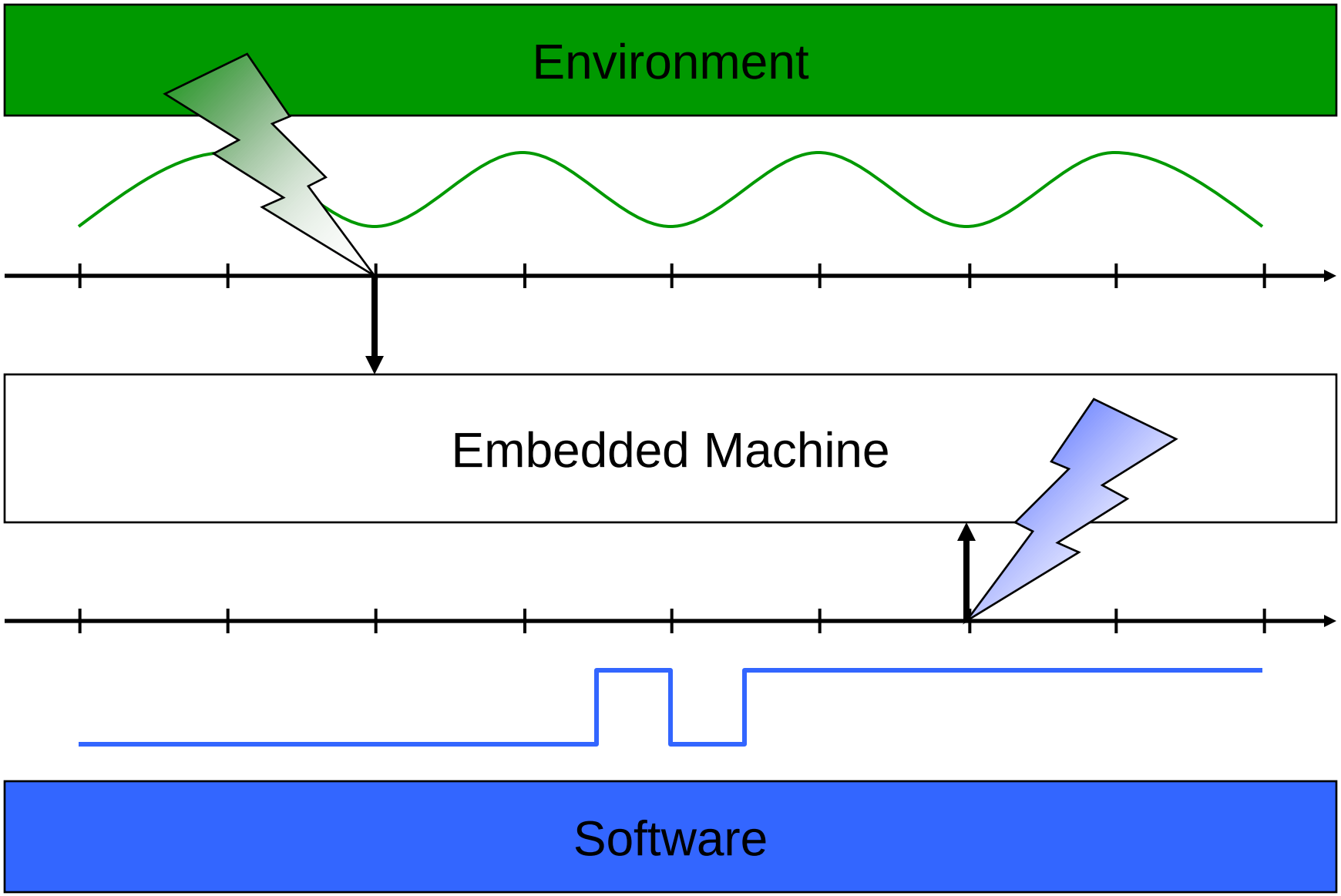
Runtime Exceptions III



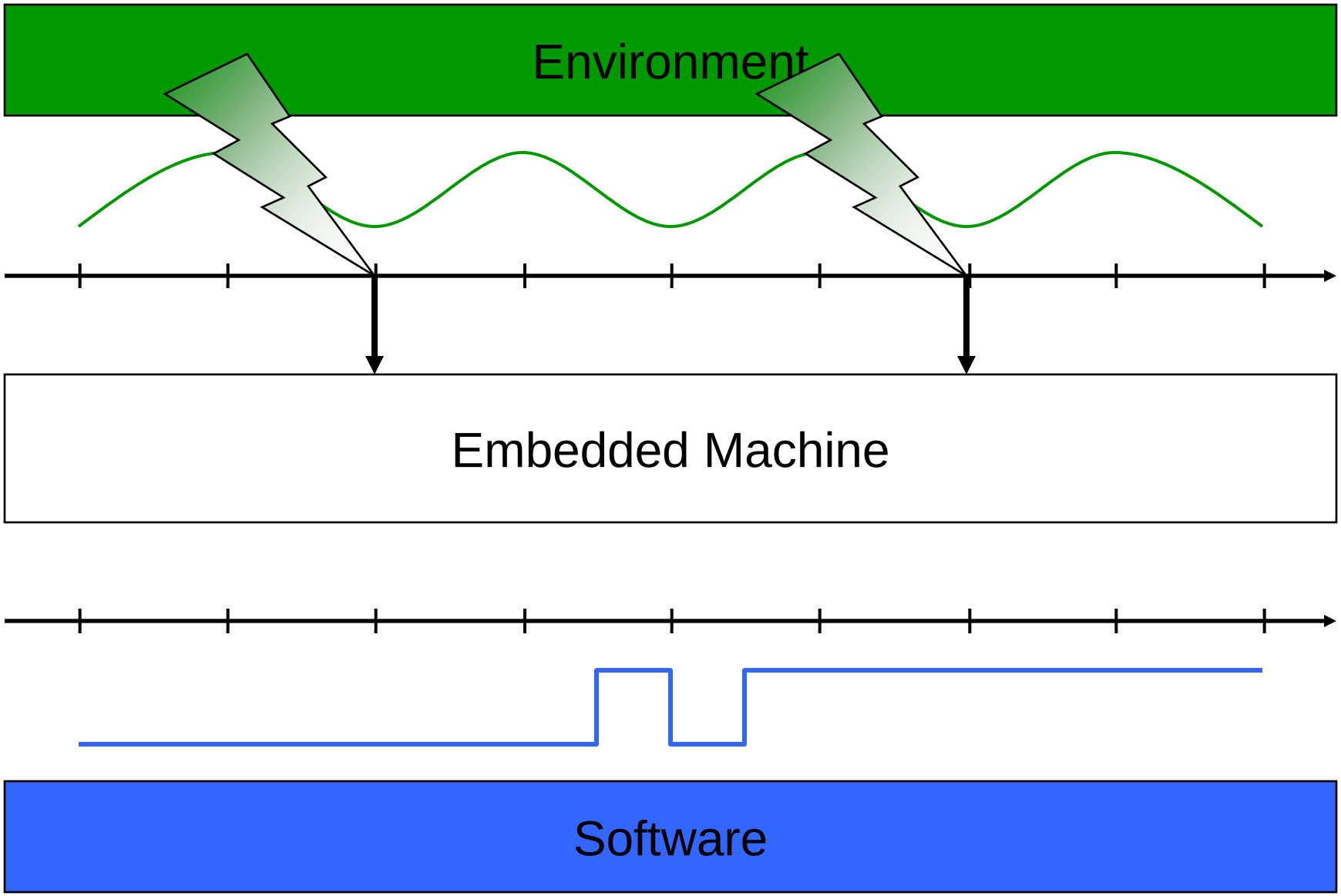
An Exception Handler `e`



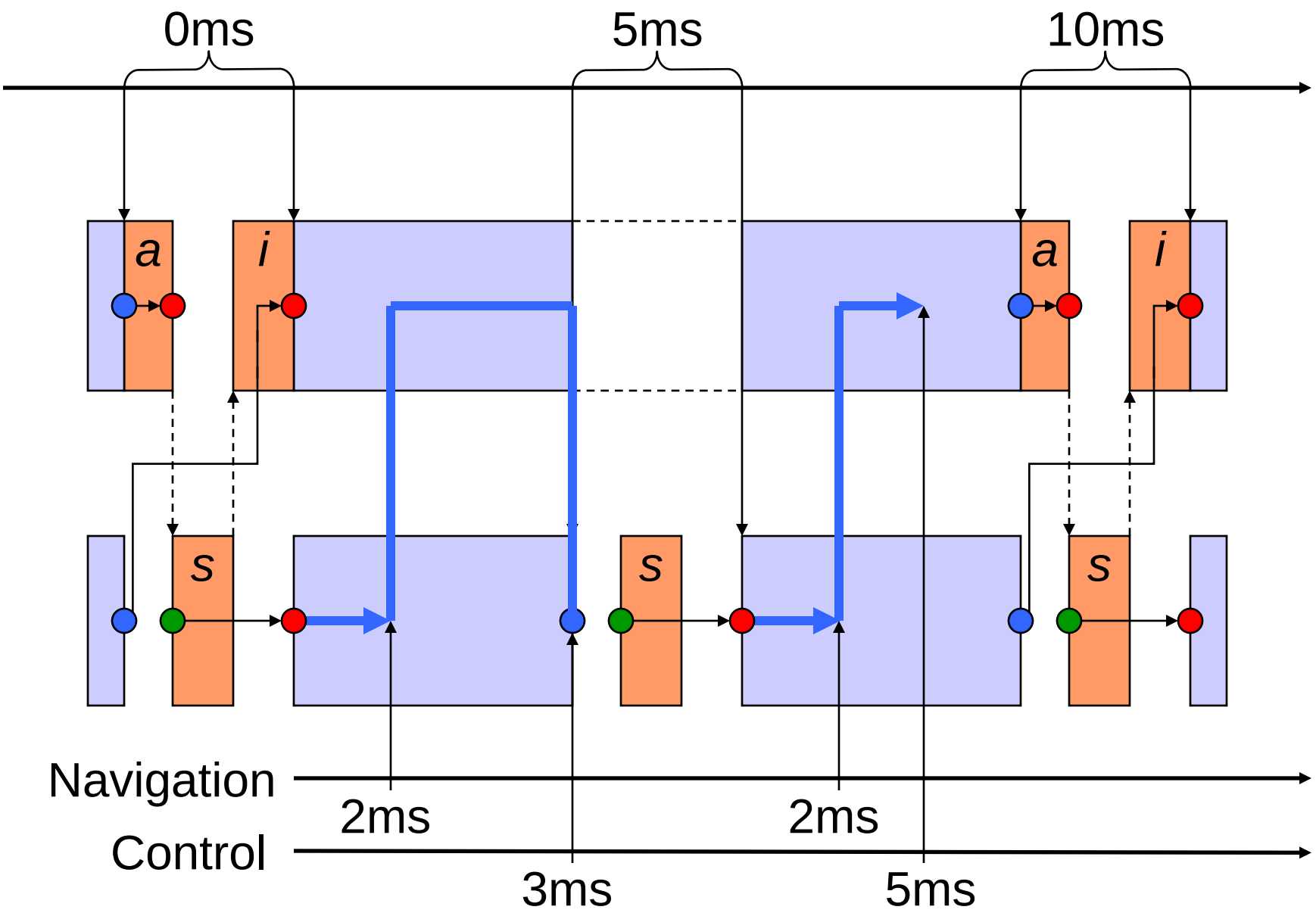
Input-Determined E Code



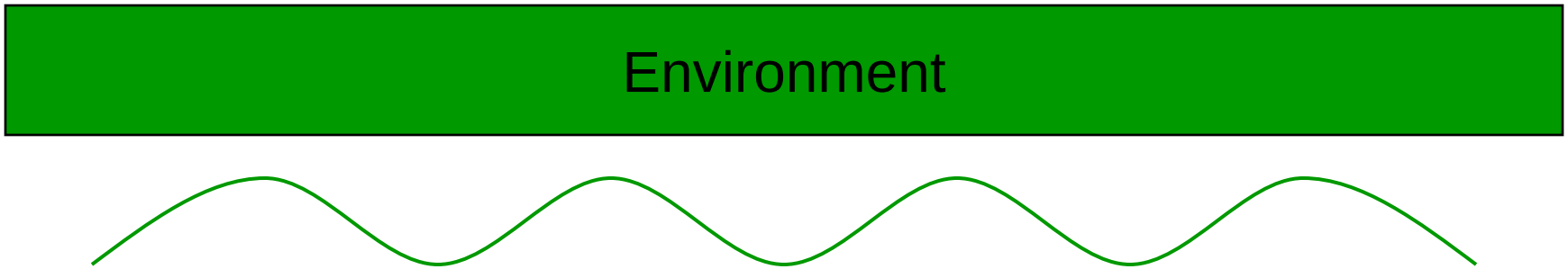
Environment-Determined E Code



Environment-Triggered E Code



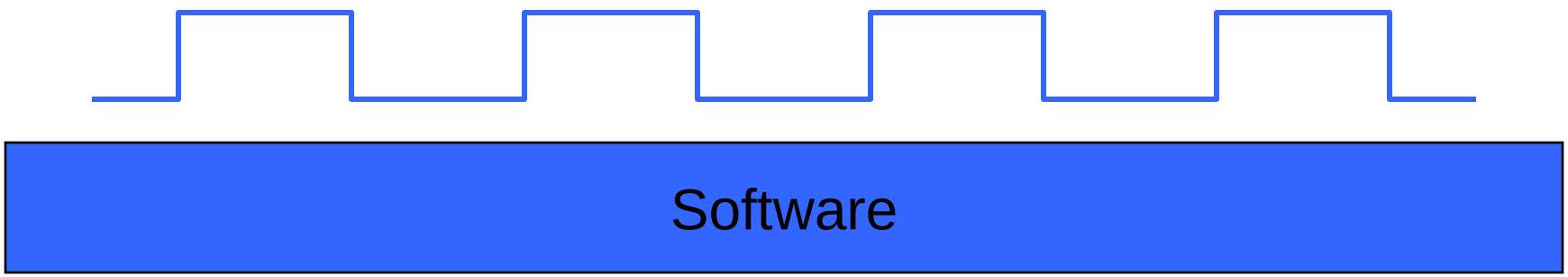
Features



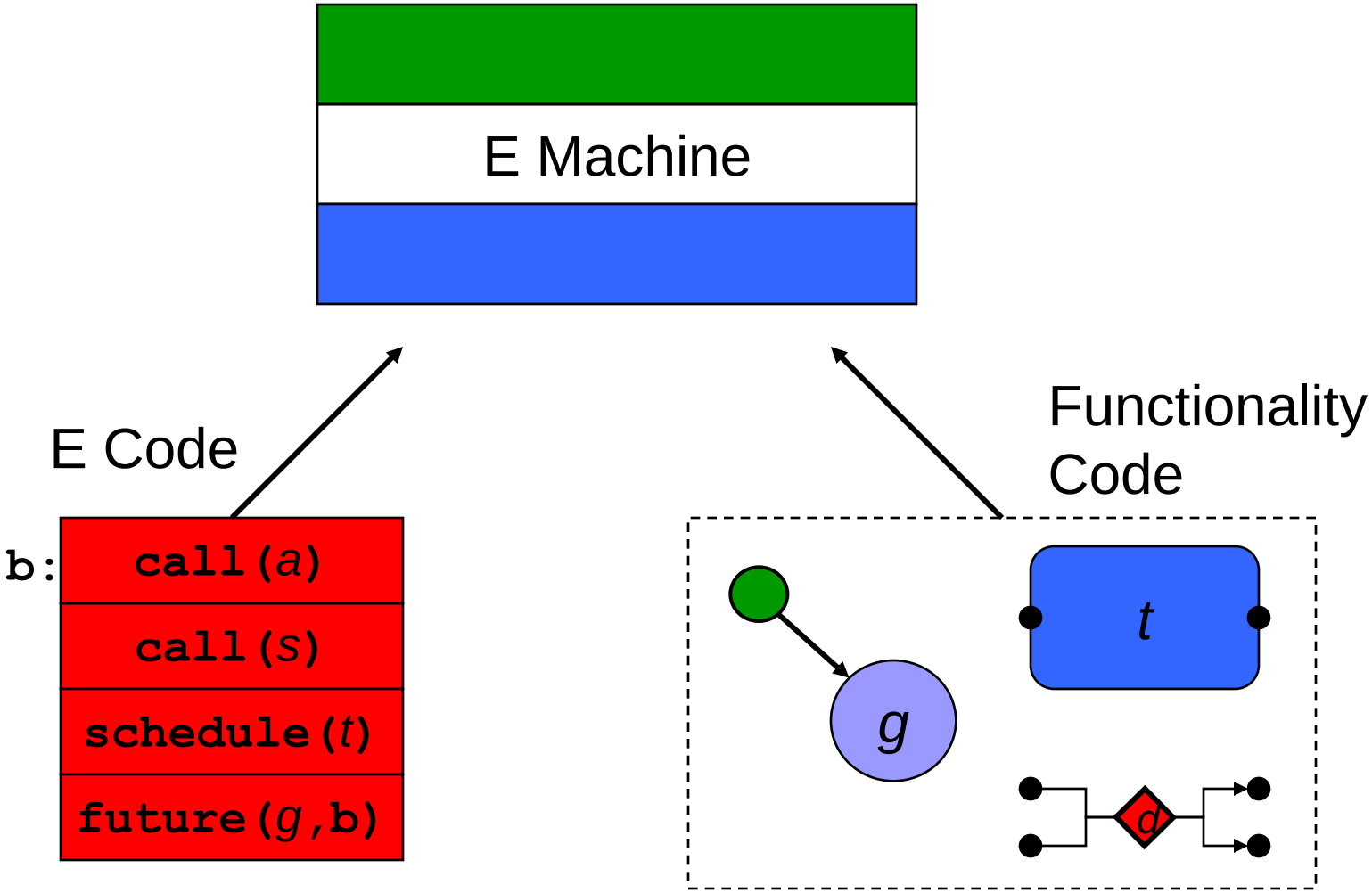
E Code:

- is *portable* real-time code (if environment-triggered)
- is *predictable* real-time code (if time-safe, or else exceptions)
- can be *linked*/*patched* (dynamically)

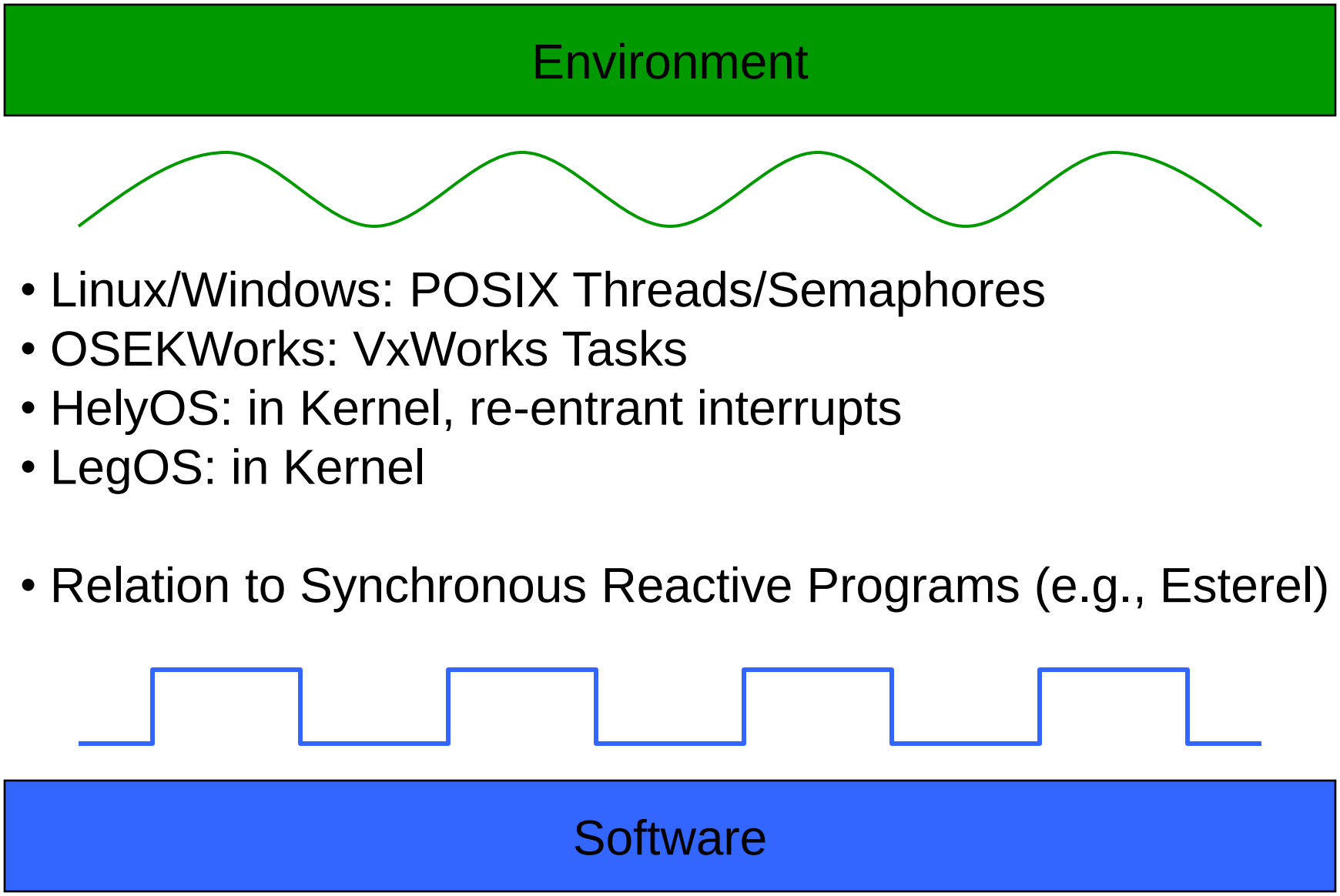
- *changes* perspectives: Schedulability = Program Analysis?



Dynamic Linking



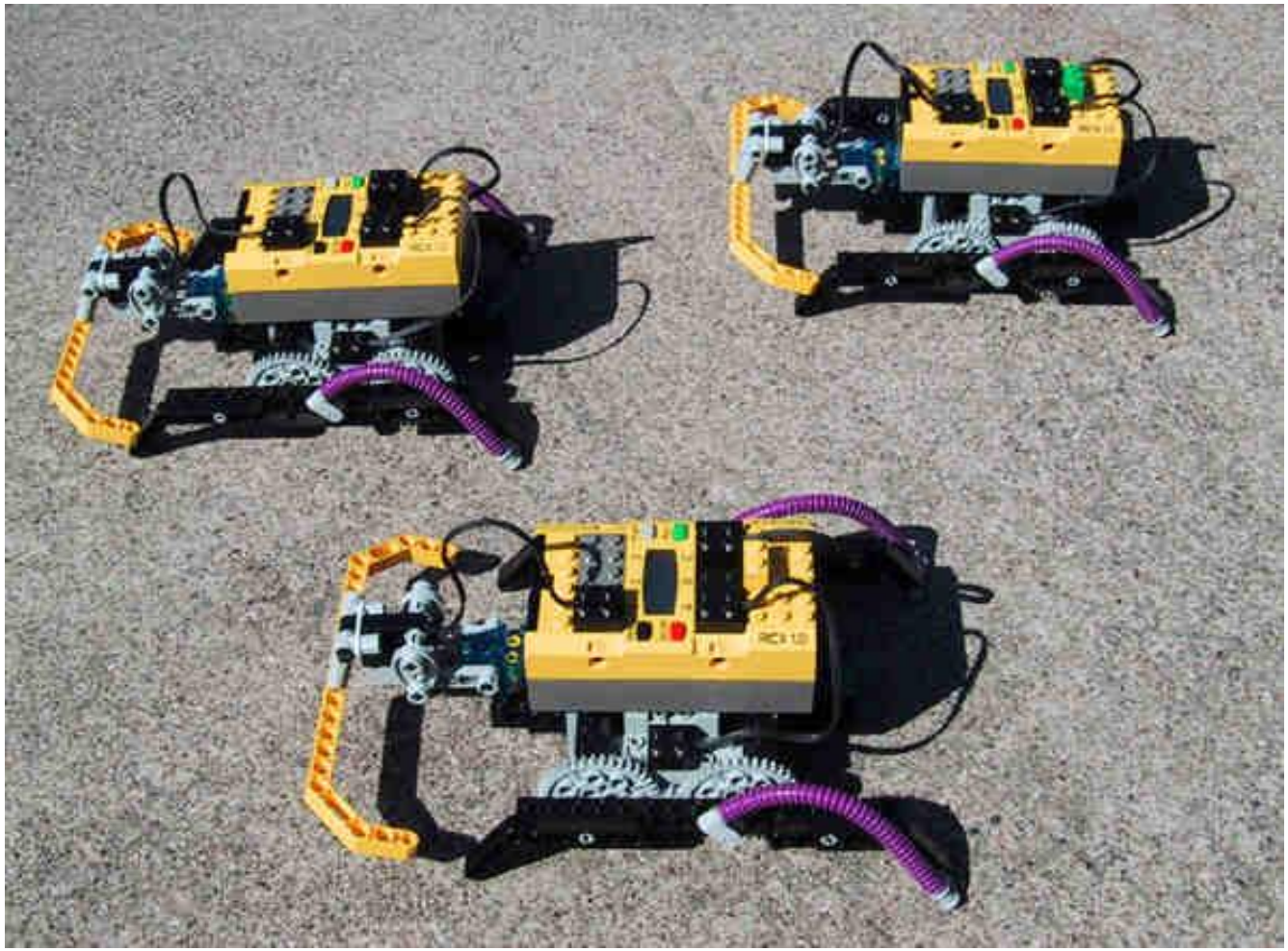
Implementations, Related Work



- Linux/Windows: POSIX Threads/Semaphores
- OSEKWorks: VxWorks Tasks
- HelyOS: in Kernel, re-entrant interrupts
- LegOS: in Kernel

- Relation to Synchronous Reactive Programs (e.g., Esterel)

That’s It



swarms of robots, Hitachi H8 microcontroller 16MHz, IR link