

The Embedded Machine

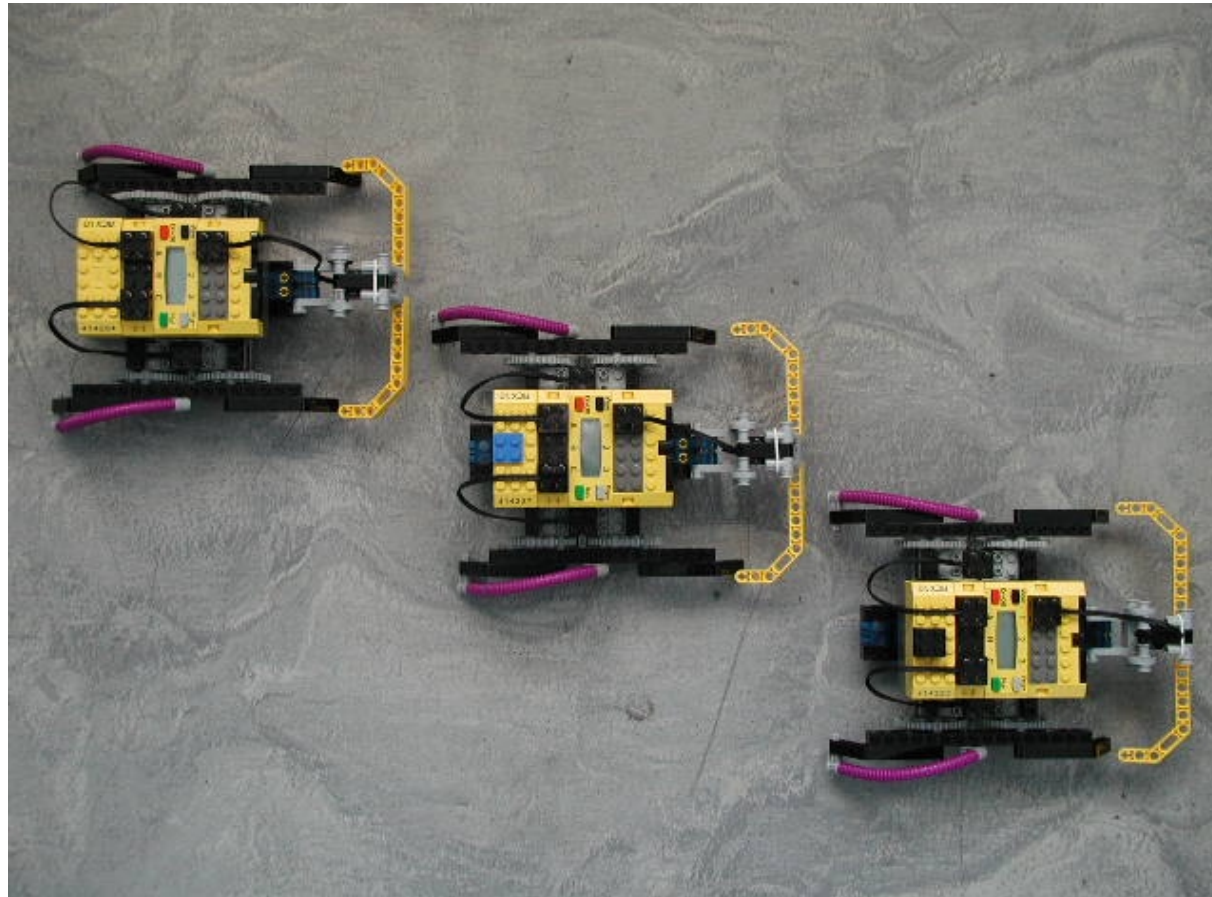
Predictable, Portable Real-Time Code

Thomas A. Henzinger, Christoph M. Kirsch

UC Berkeley

www.eecs.berkeley.edu/~cm

The Mindstorm Machine



Embedded Software



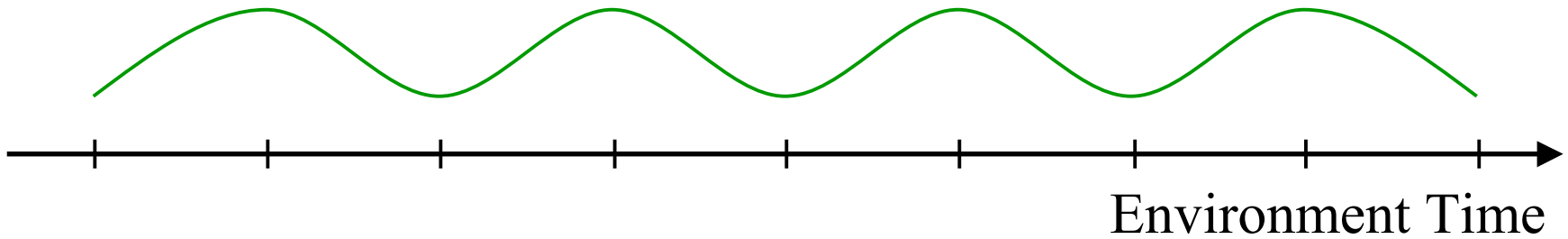
Environment



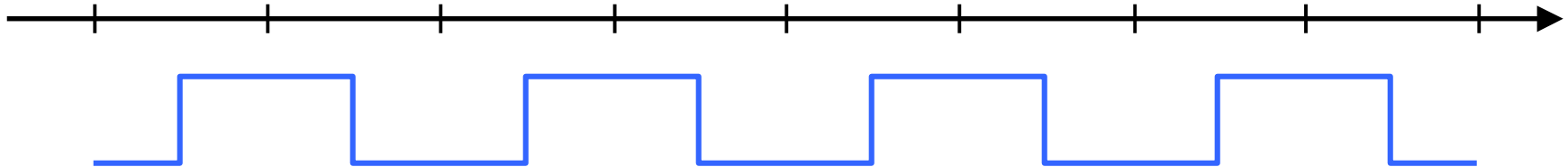
Software

Environment vs. Platform Time

Environment



Platform Time

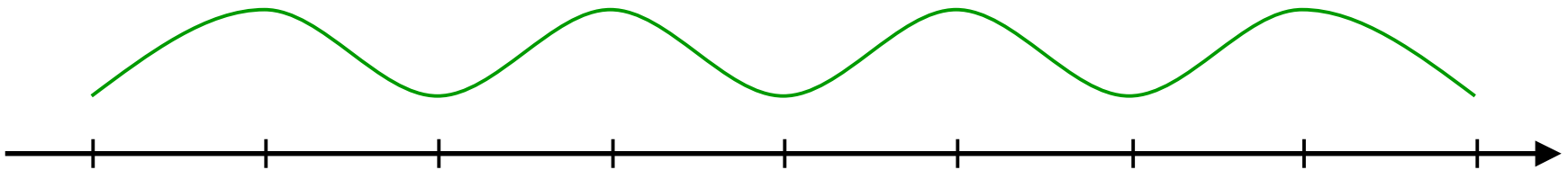


Software



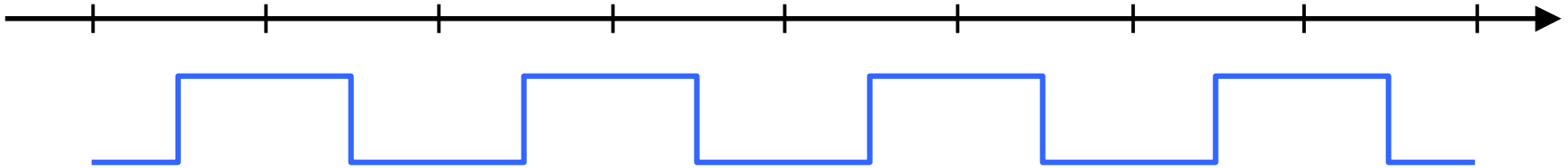
Reactivity vs. Schedulability

Environment



Reactivity!

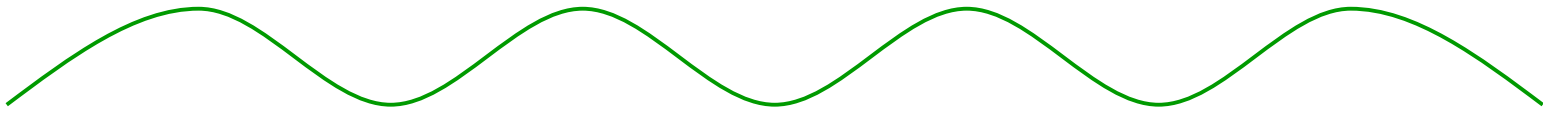
Schedulability!



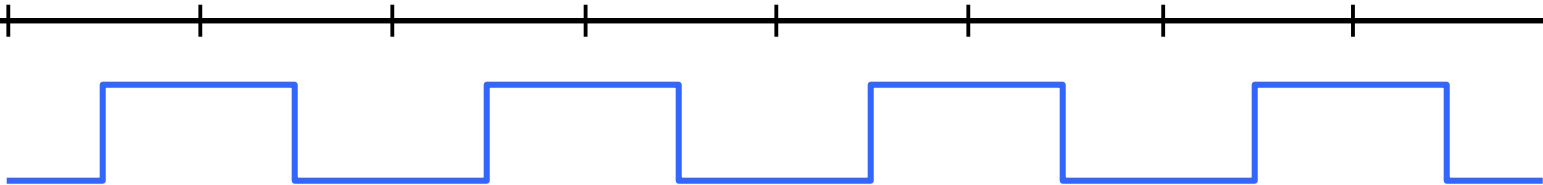
Software

The Art of Embedded Programming

Environment



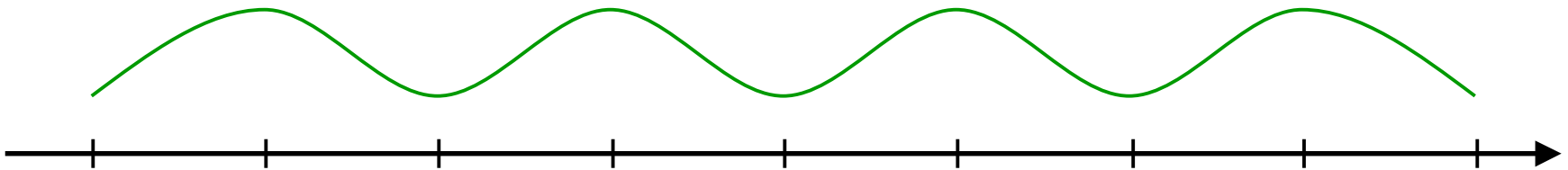
Embedded Programming



Software

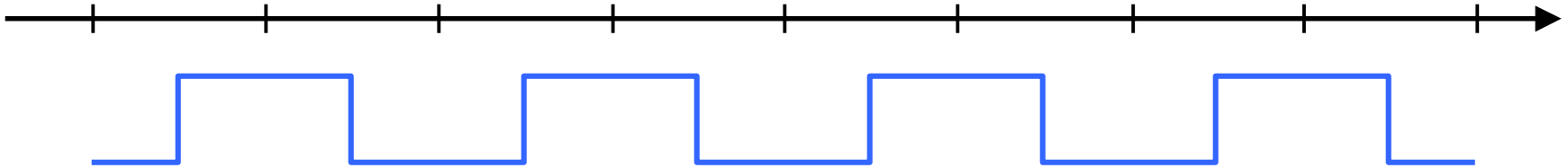
Proposal

Environment



Human: Programming in terms of environment time

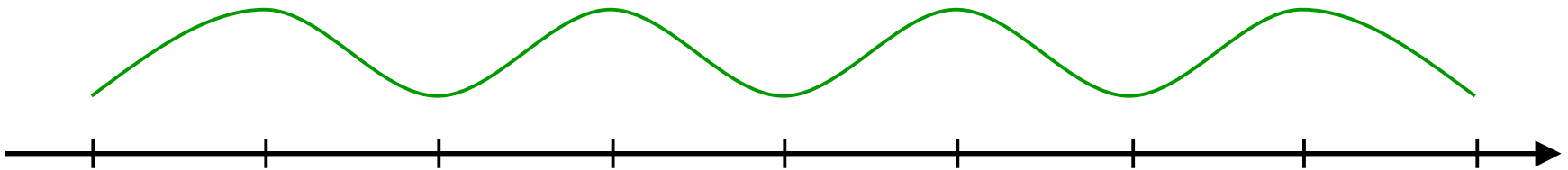
Compiler: Implementation in terms of platform time



Software

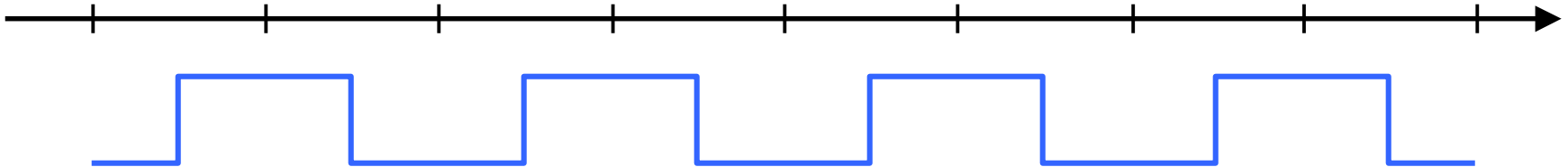
Platform Time is Platform Memory

Environment



- Programming as if there is enough platform time

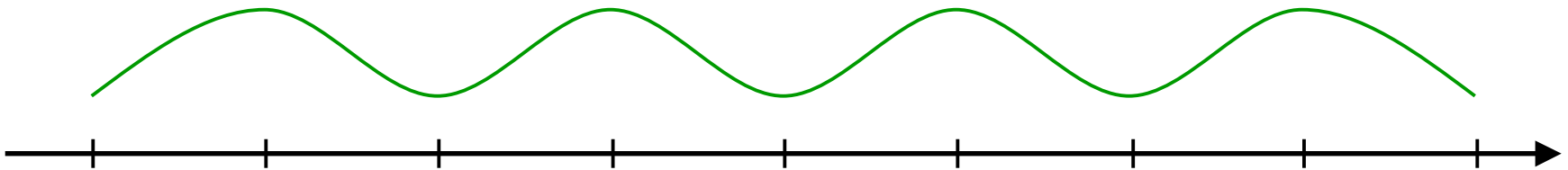
-
- Implementation checks whether there is enough of it



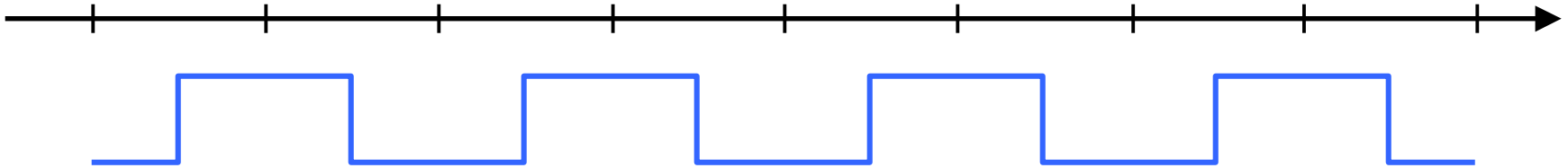
Software

Portability

Environment



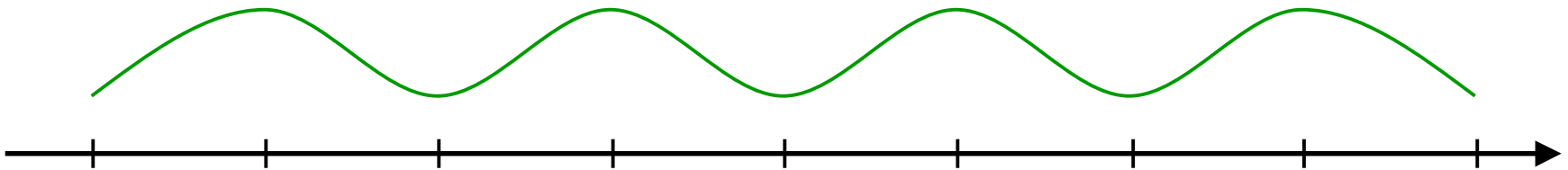
- Programming in terms of environment time yields platform-independent code



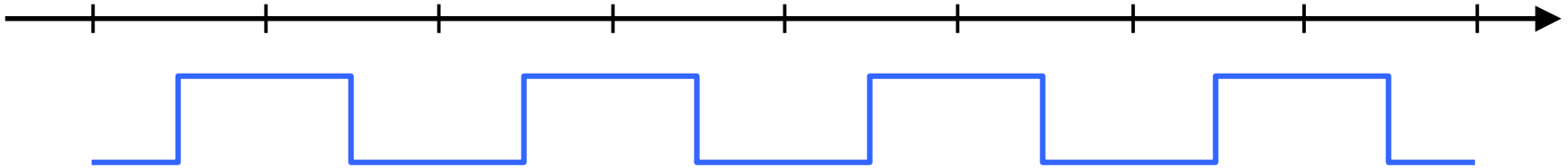
Software

Predictability

Environment

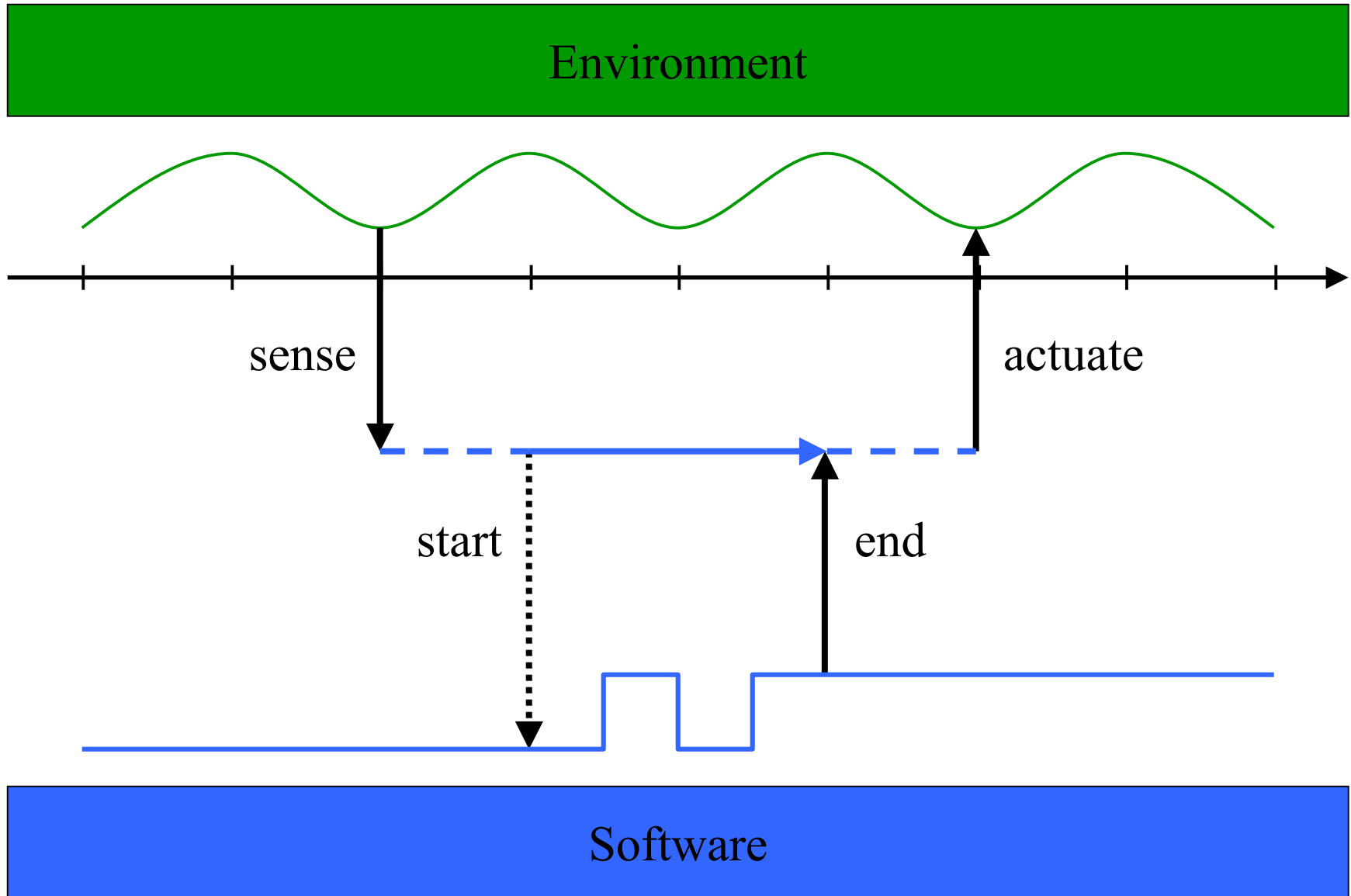


- Programming in terms of environment time yields deterministic code

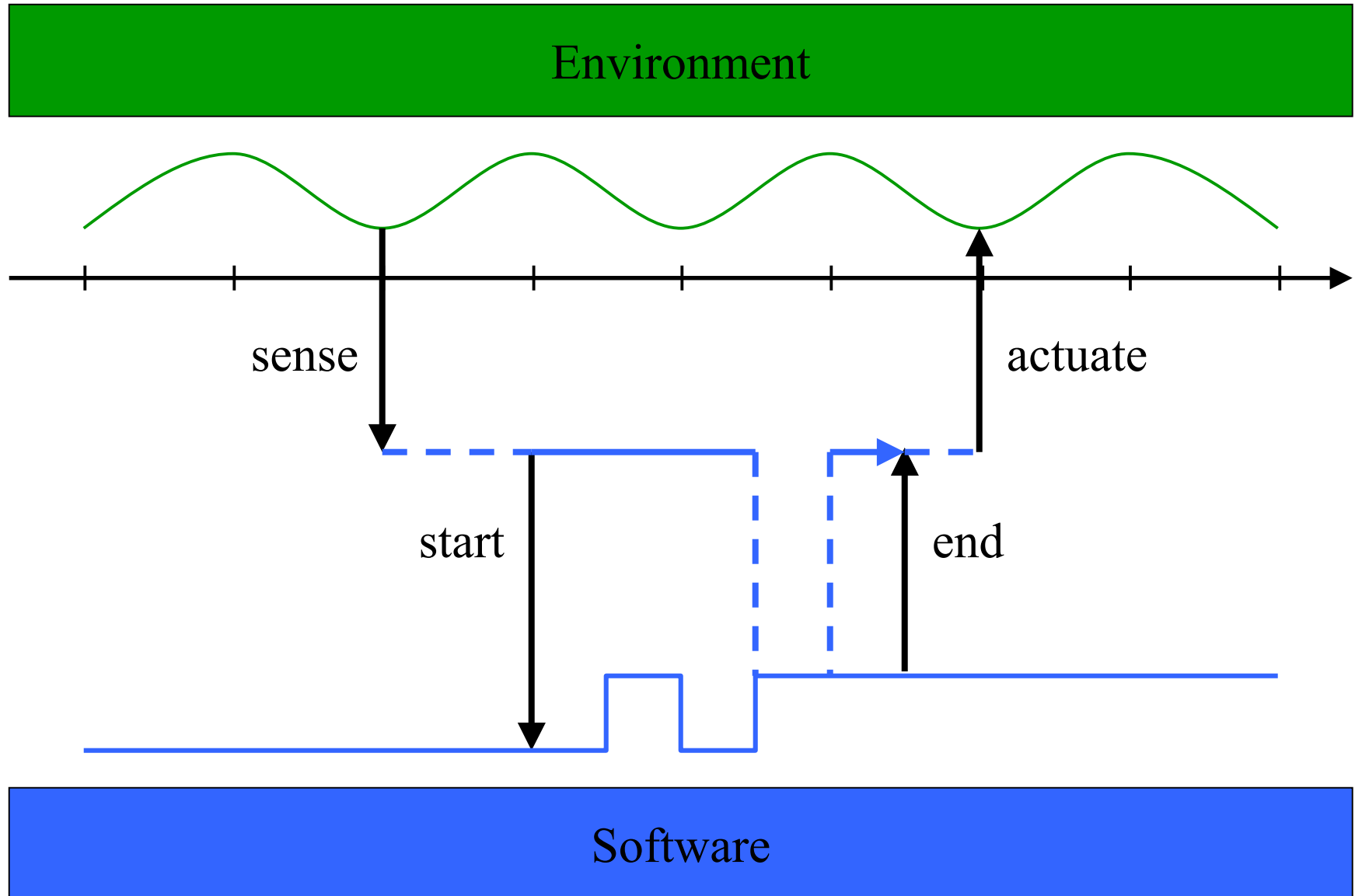


Software

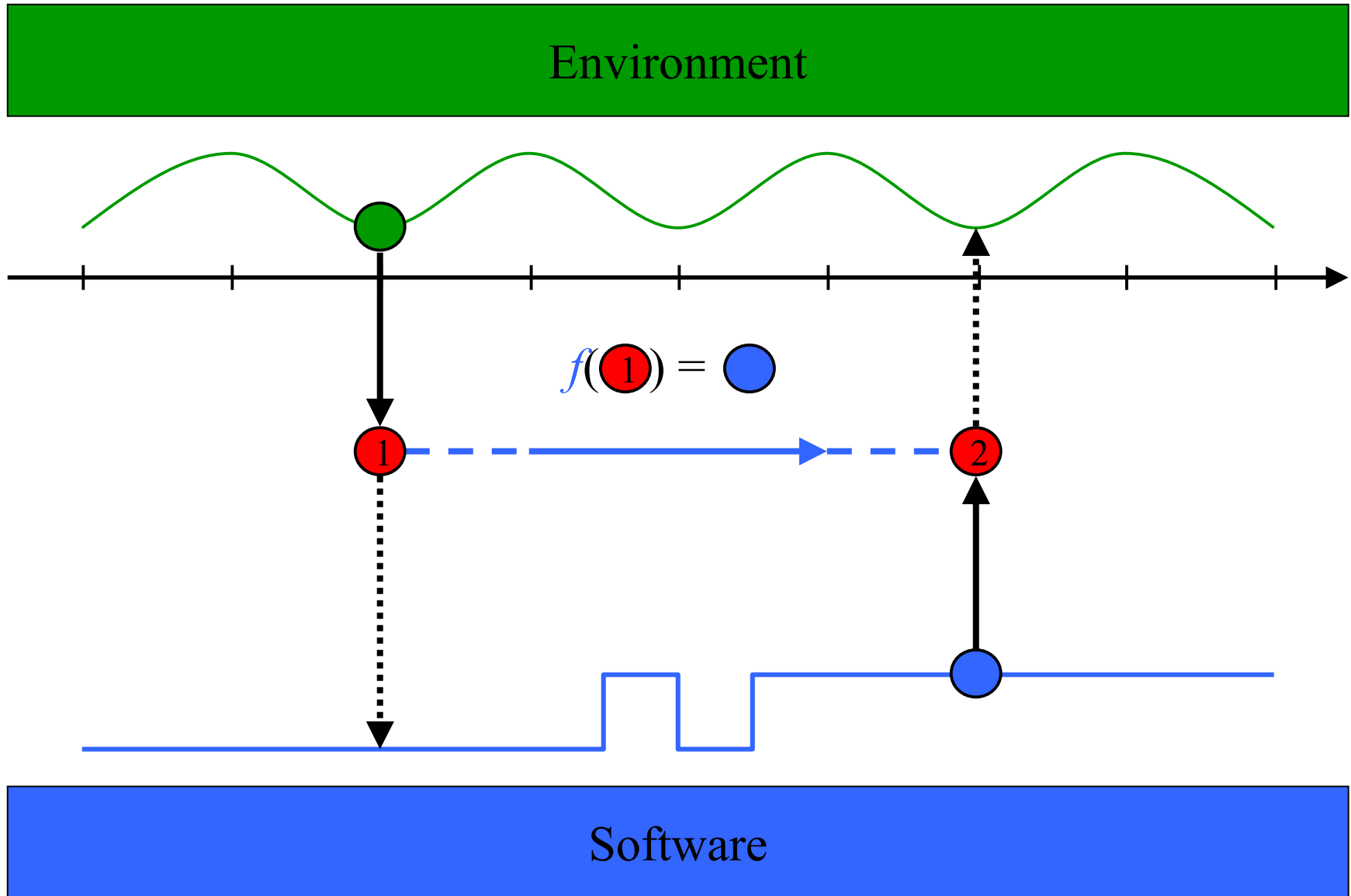
The Task Model



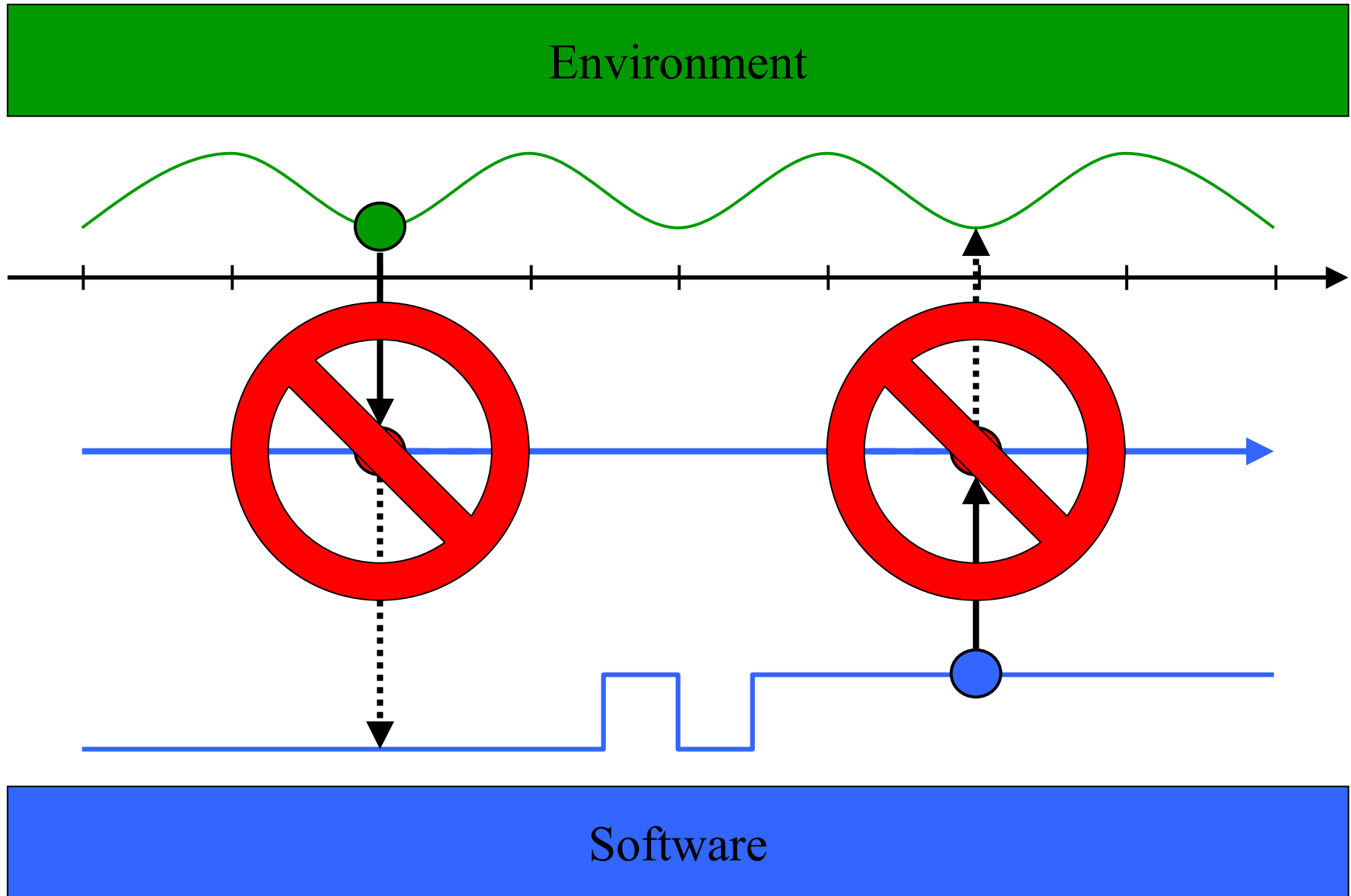
Preemptable...



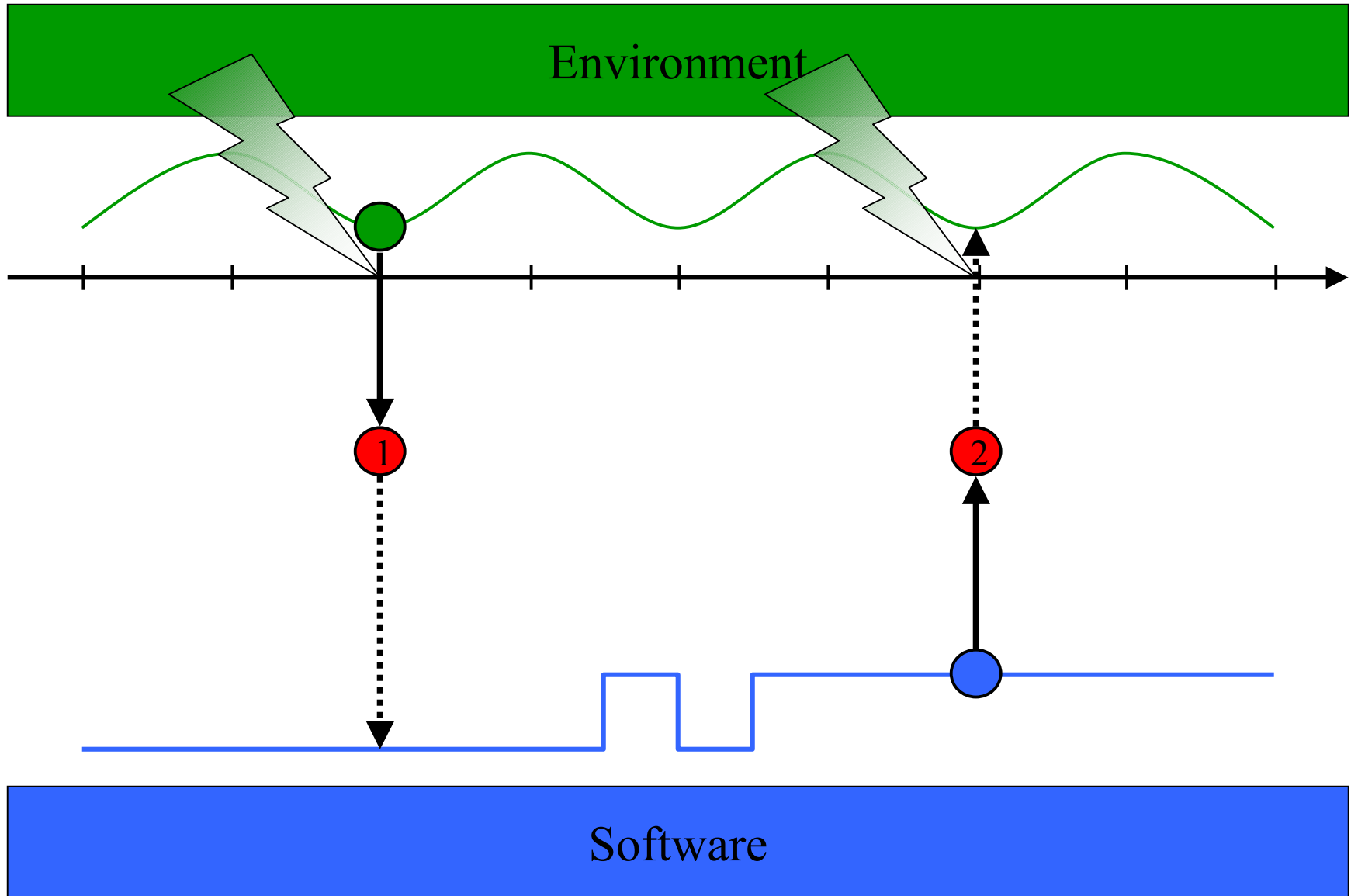
...but Atomic



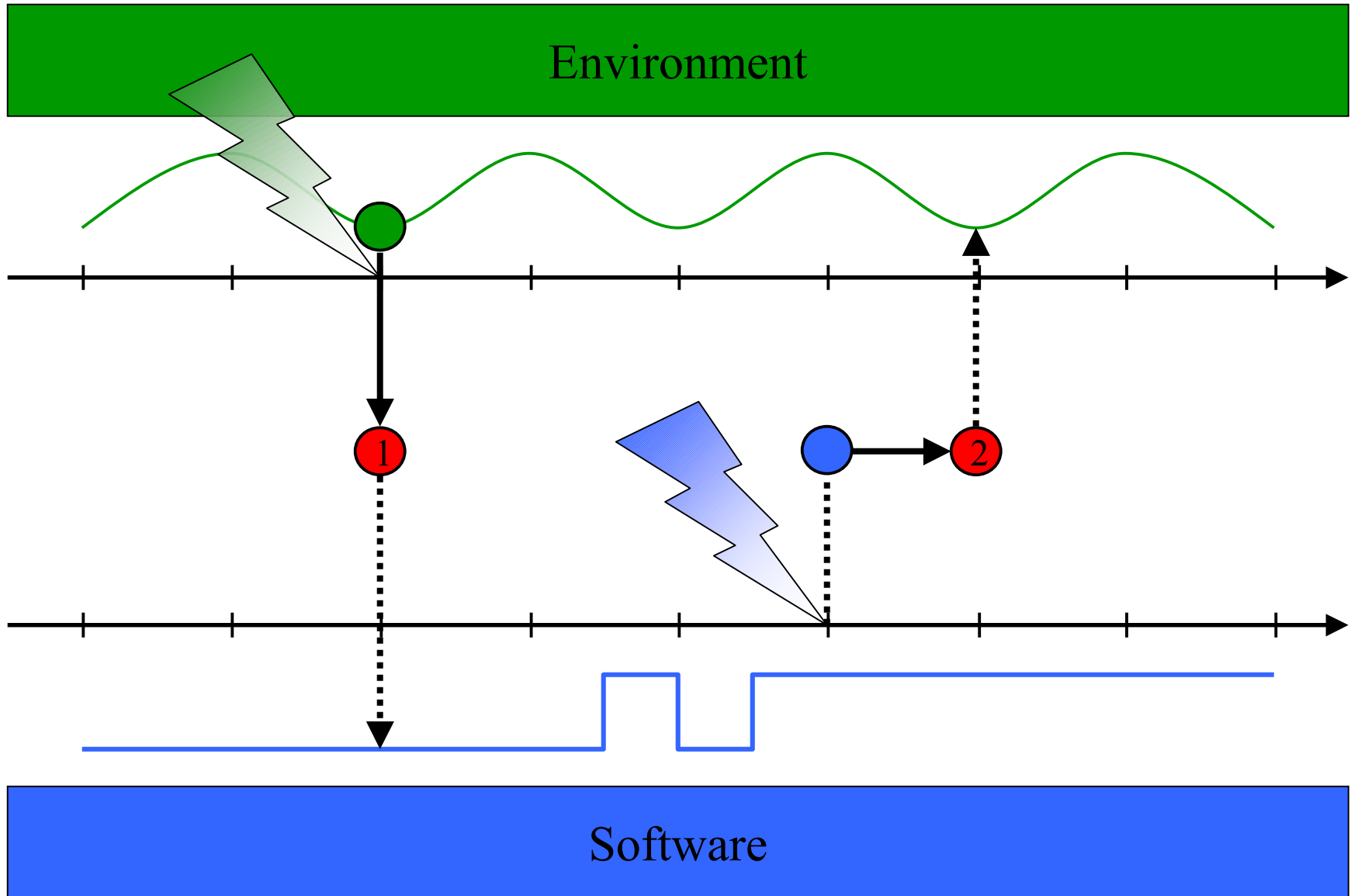
Time Safety



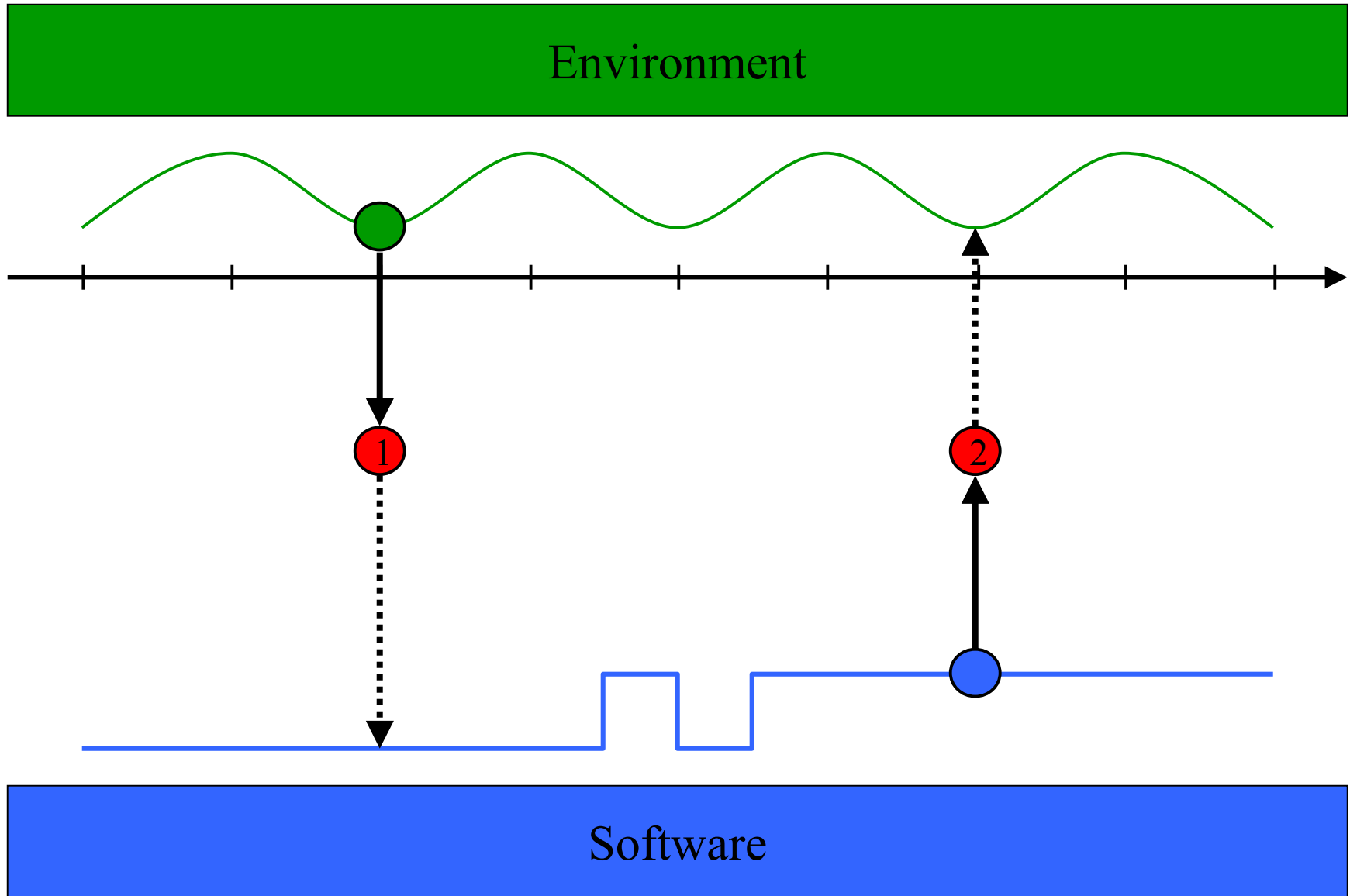
Environment-triggered Code



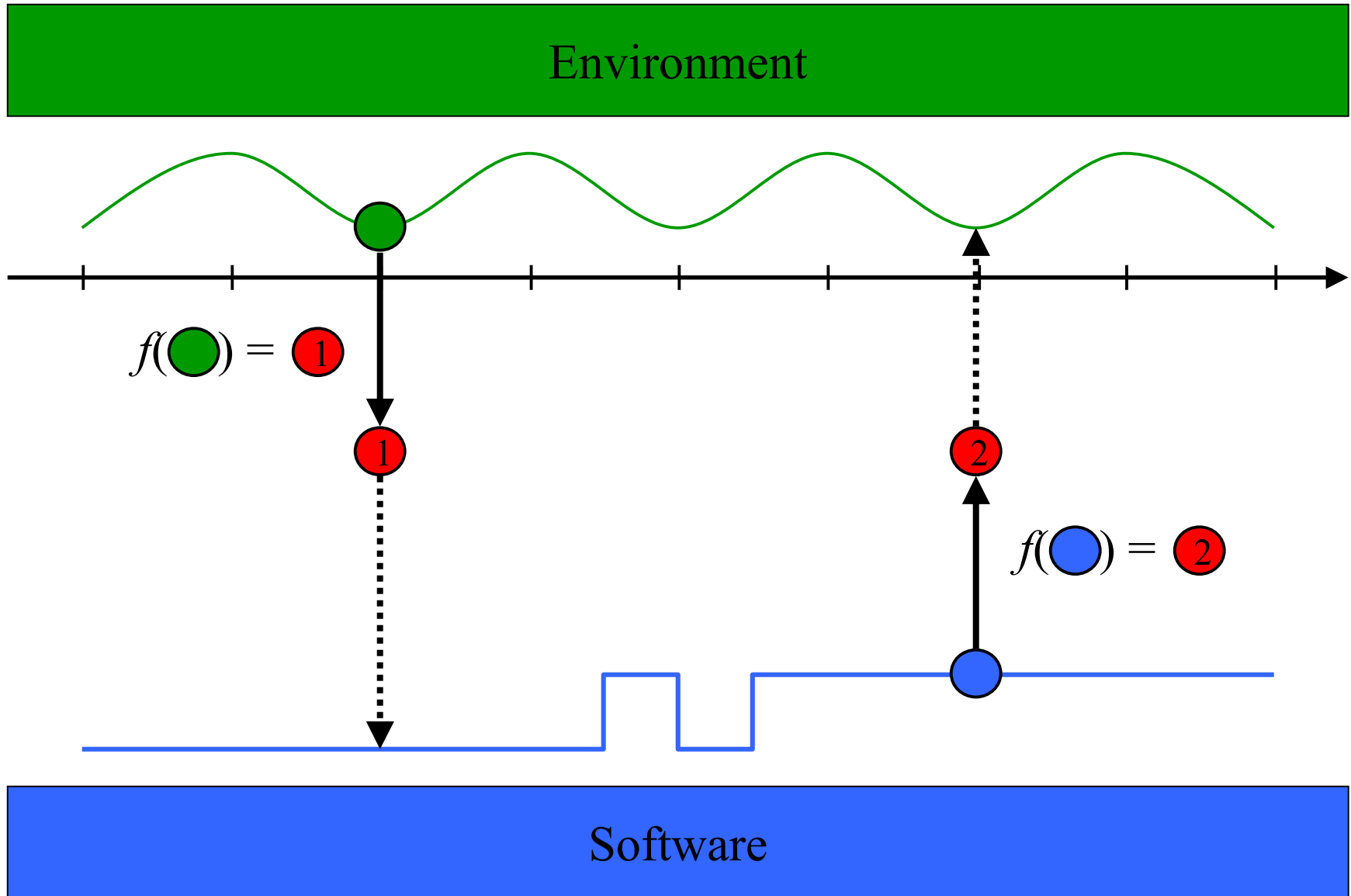
Task-triggered Code



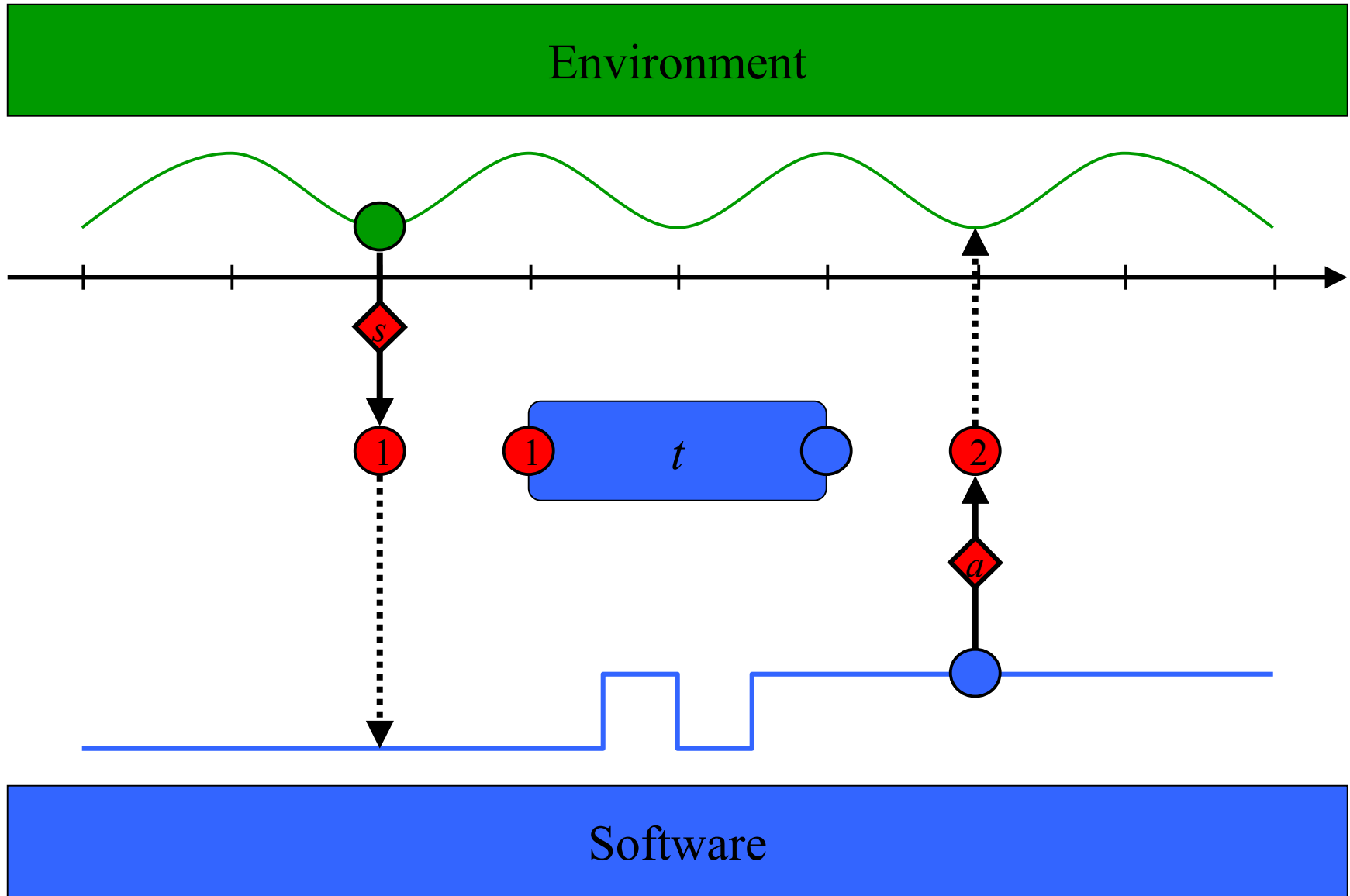
The Driver Model



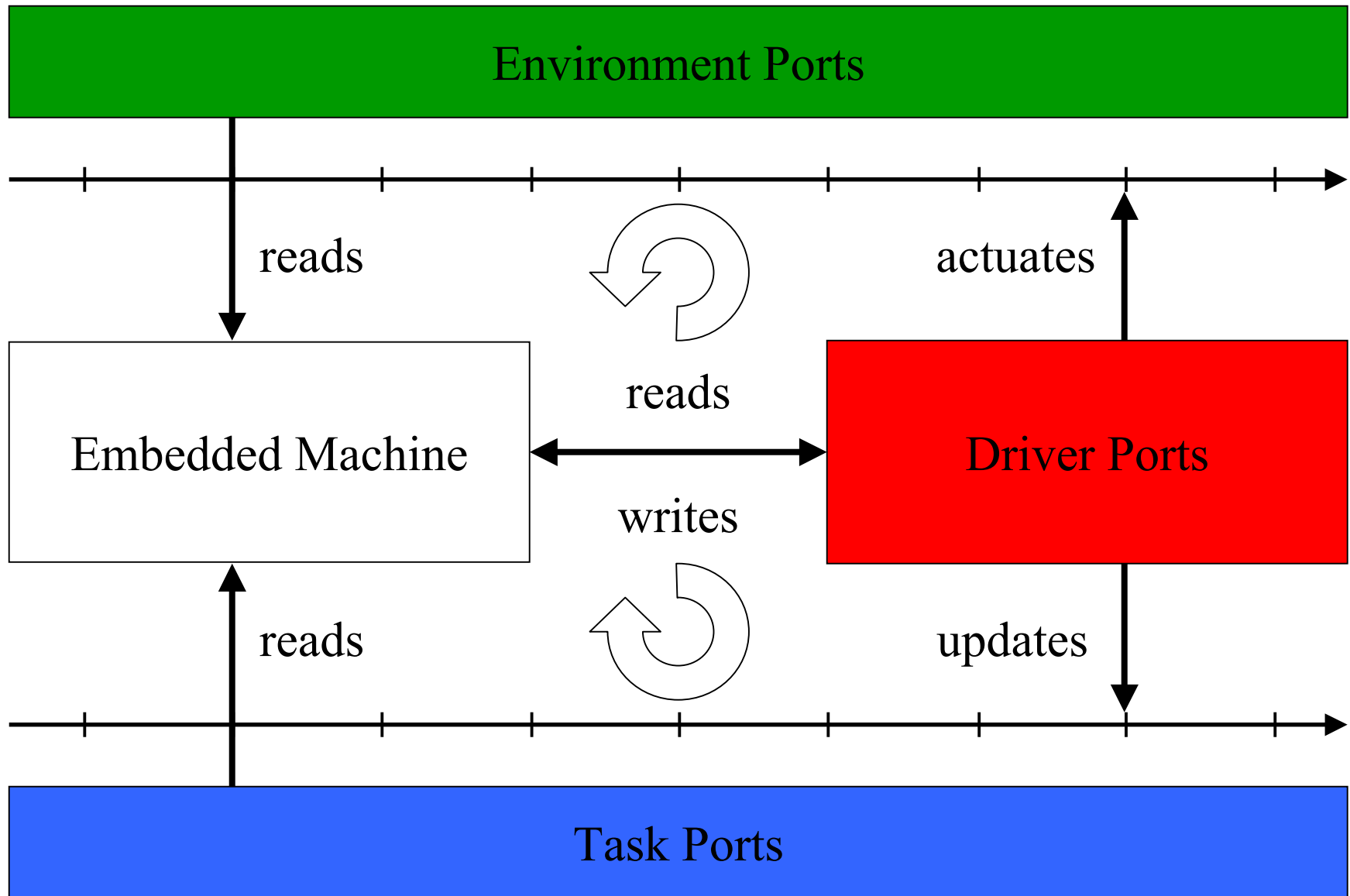
Non-preemptable, Synchronous



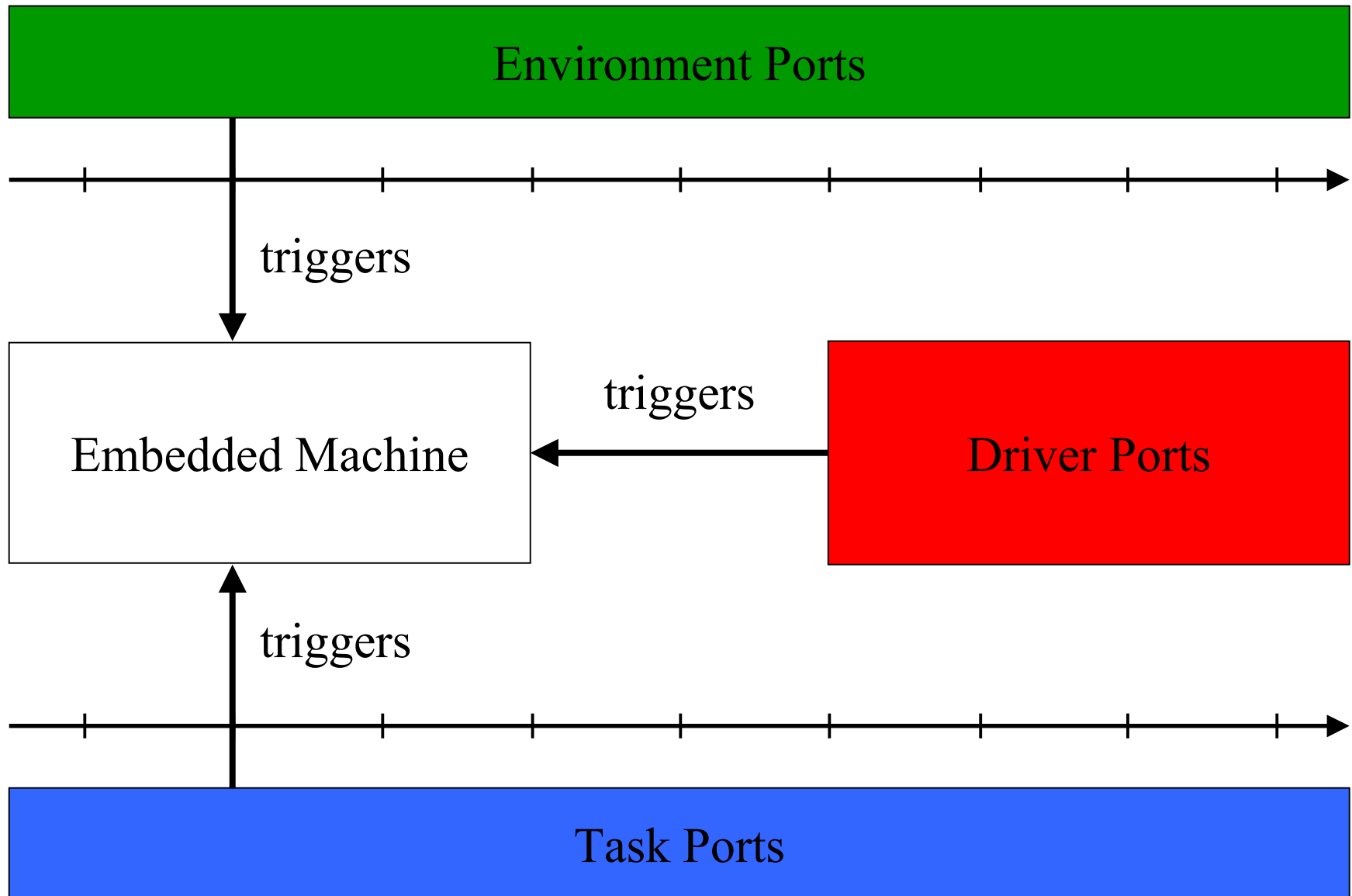
Syntax



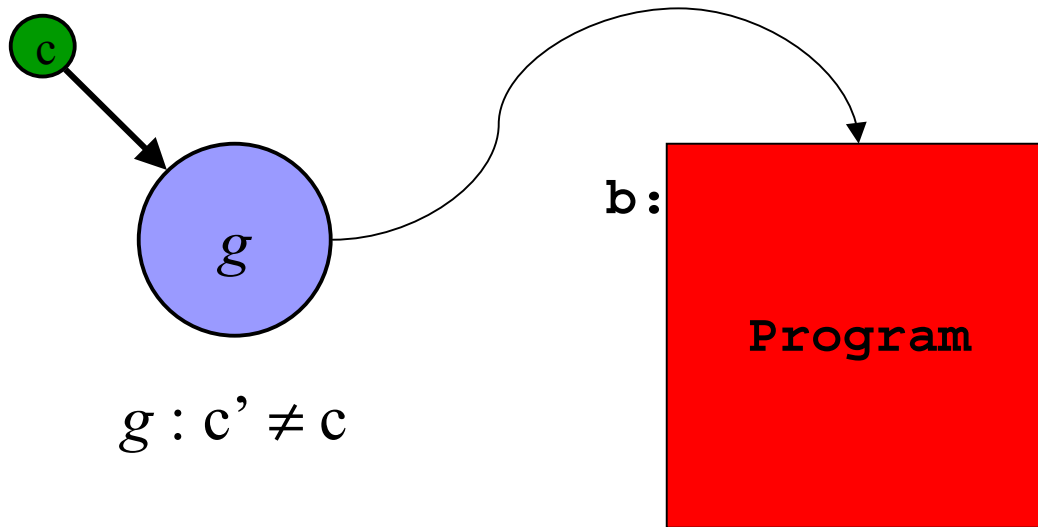
The Embedded Machine



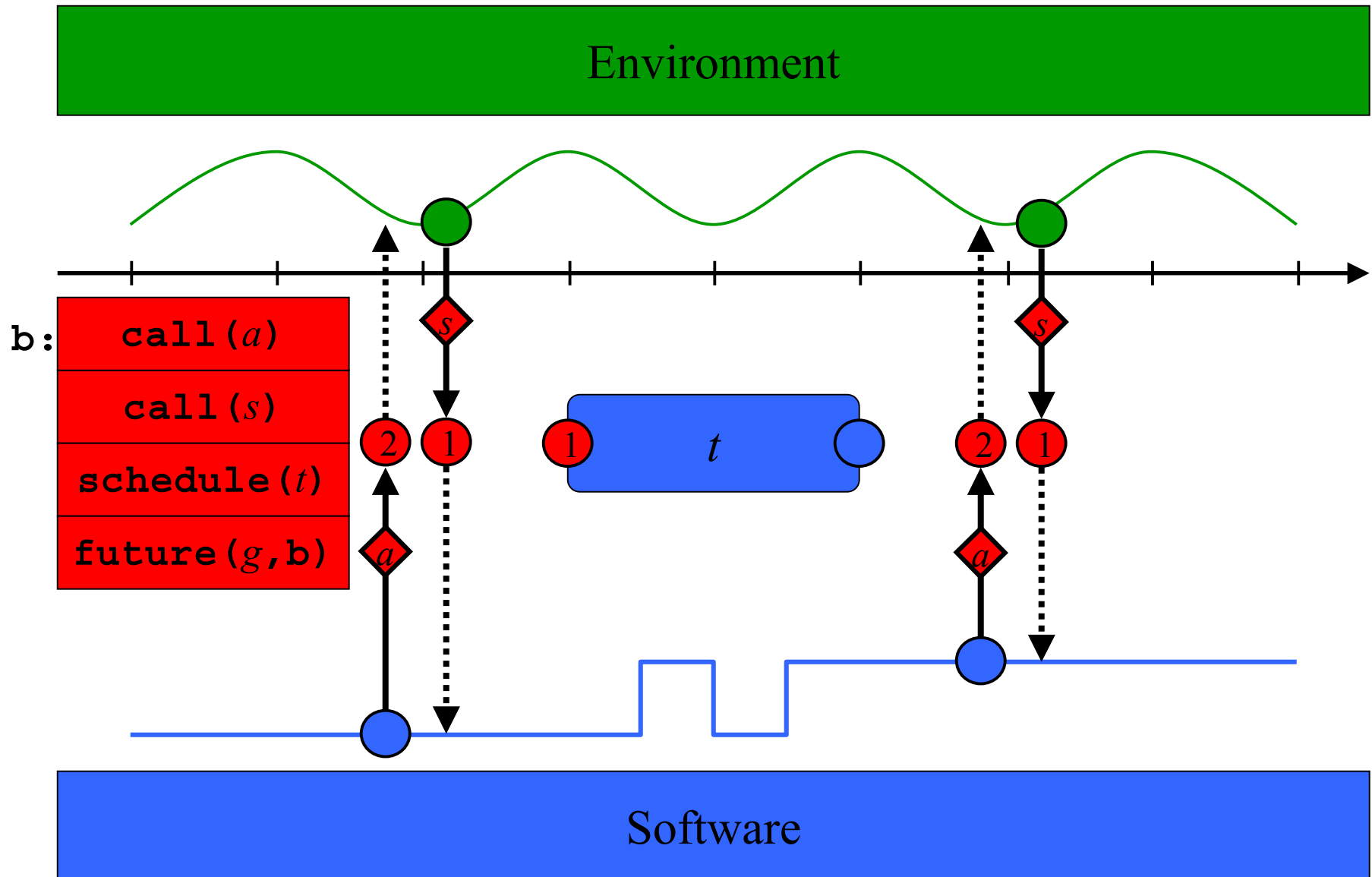
The Trigger Model



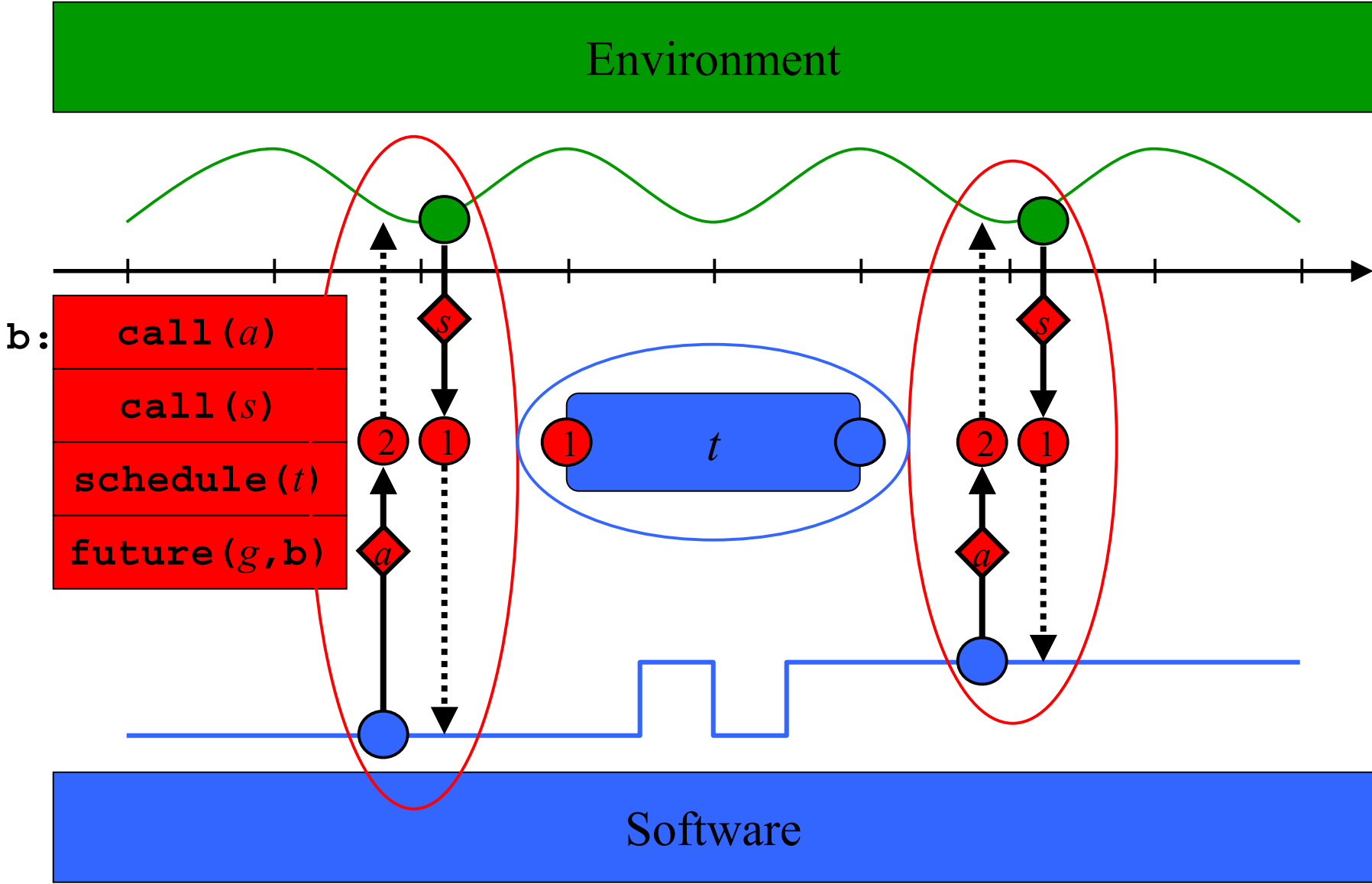
A Trigger g



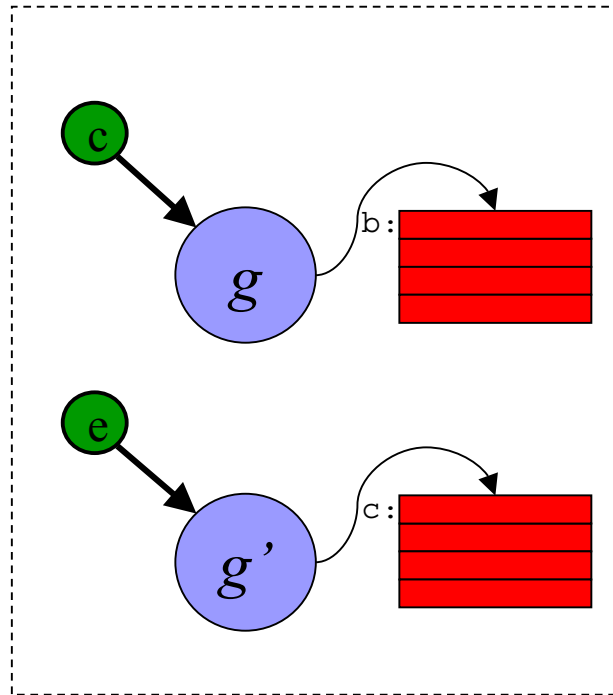
An Embedded Machine Program



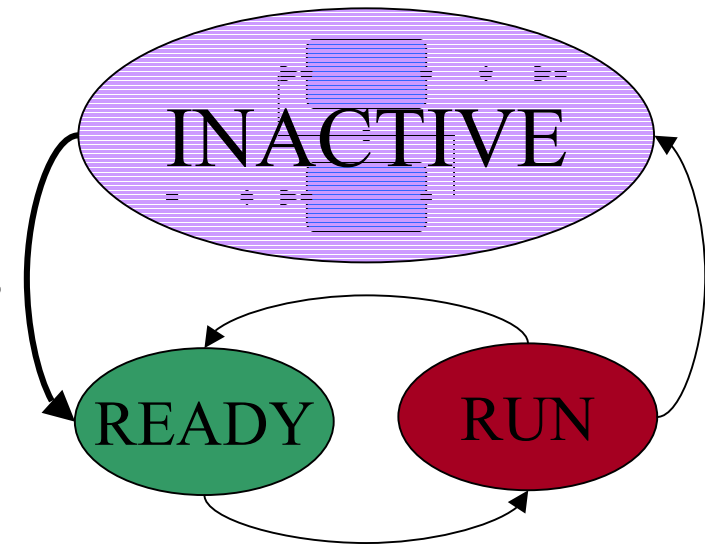
Synchronous vs. Scheduled Computation



Synchronous vs. Scheduled Computation



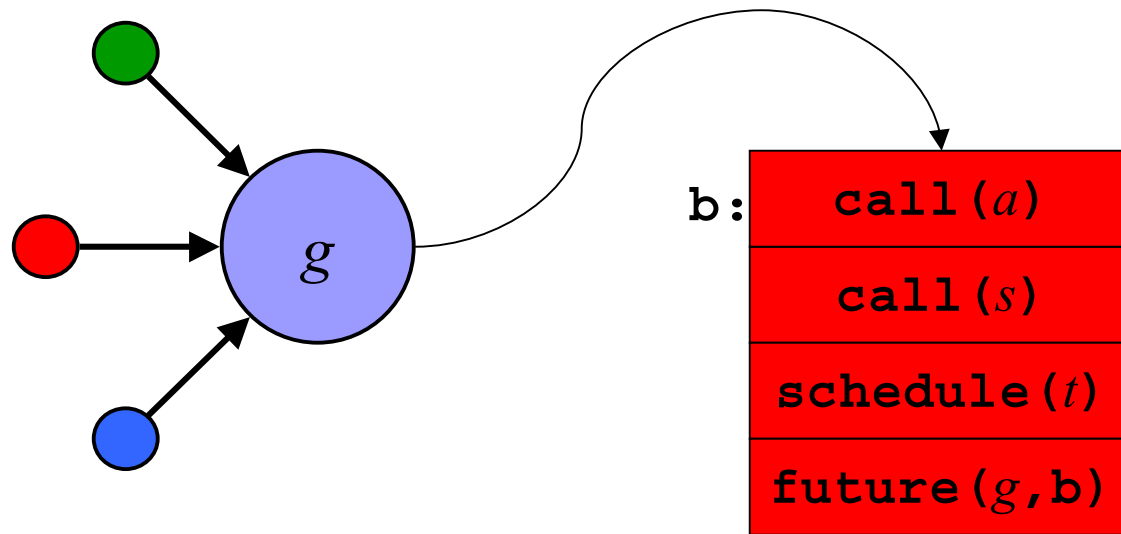
schedules



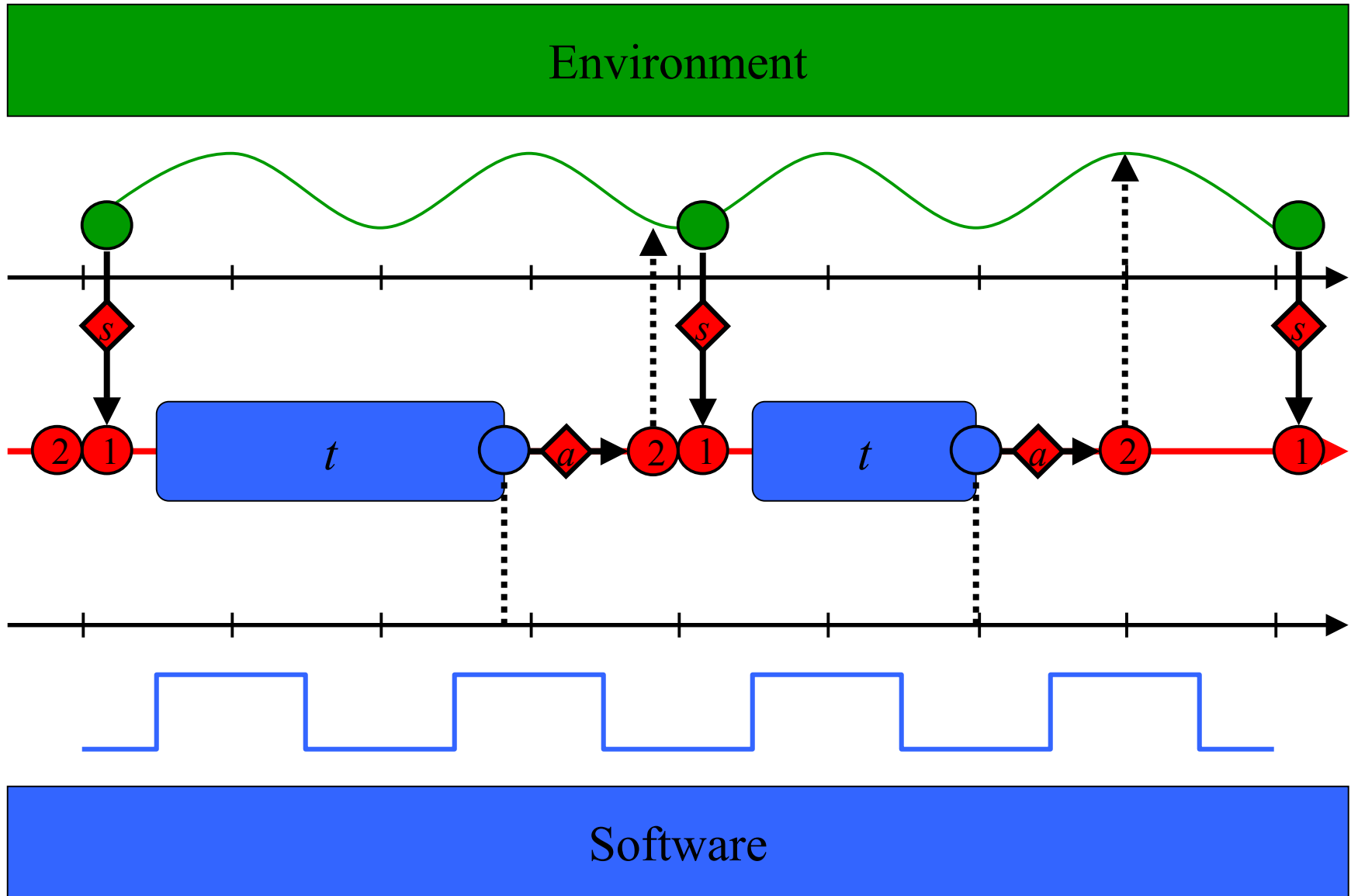
- Synchronous computation
- Kernel context
- Trigger related interrupts disabled

- Scheduled computation
- User context

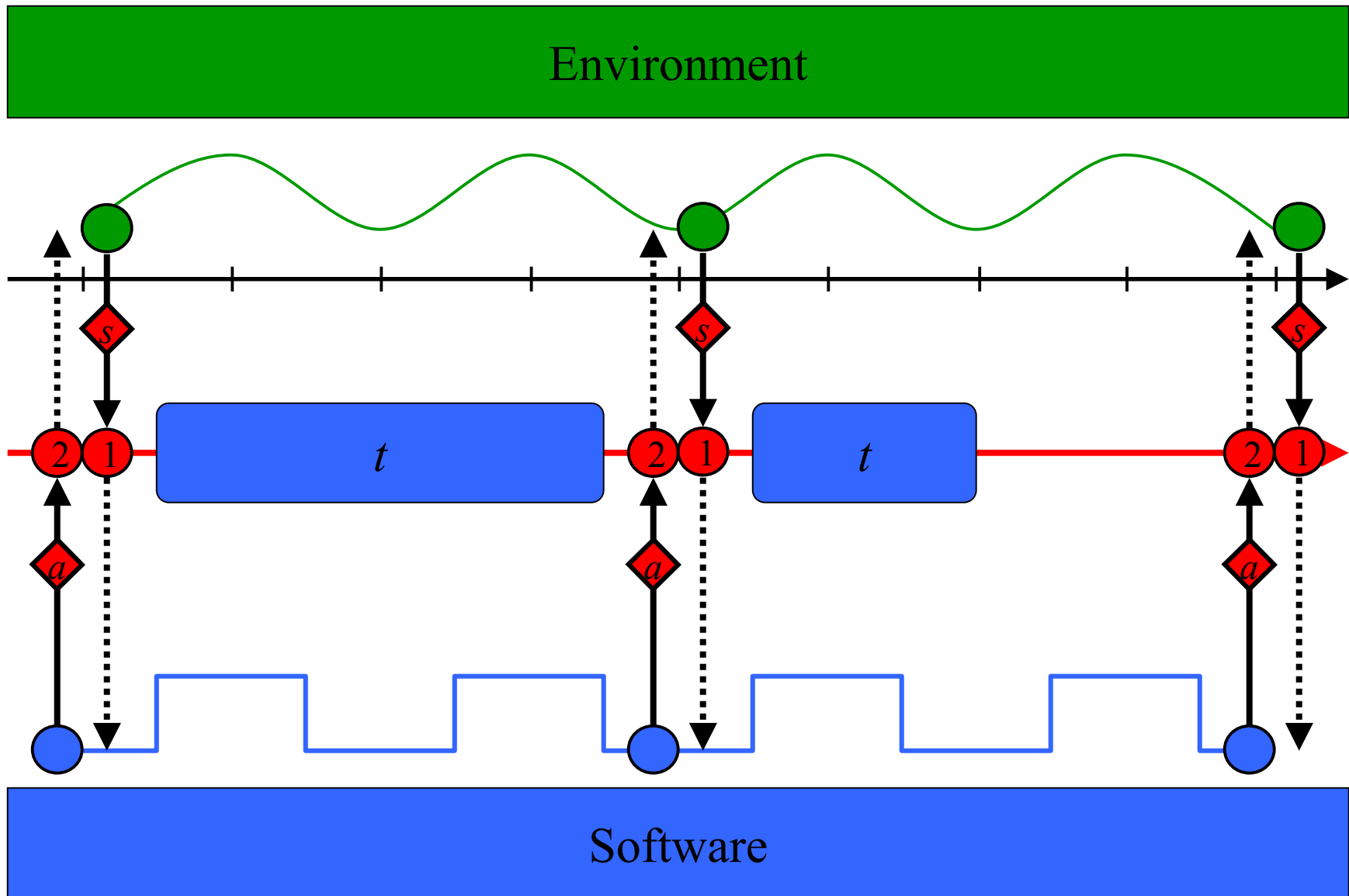
Trigger g : Input-, Environment-Triggered



Input-deterministic If Time Safe



Environment-deterministic If Environment-triggered

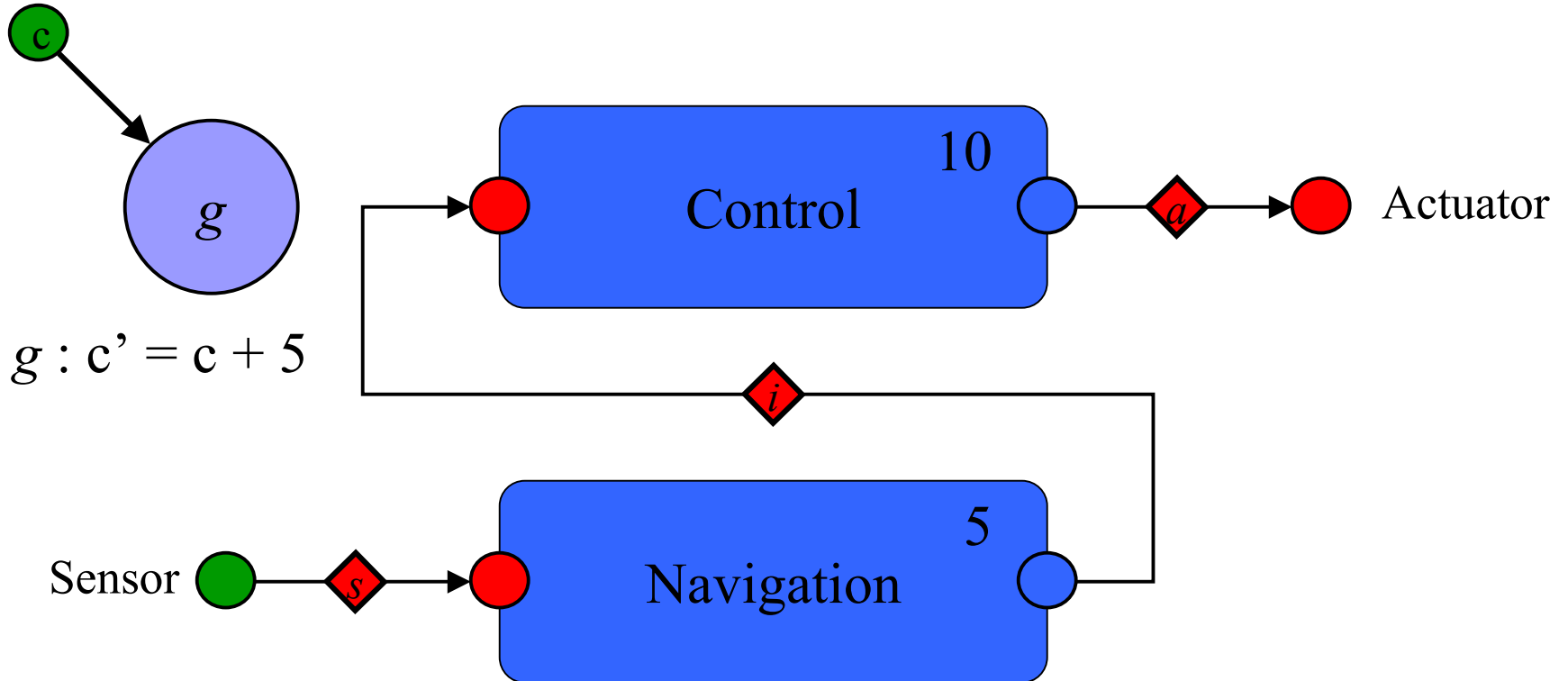


The Zürich Helicopter

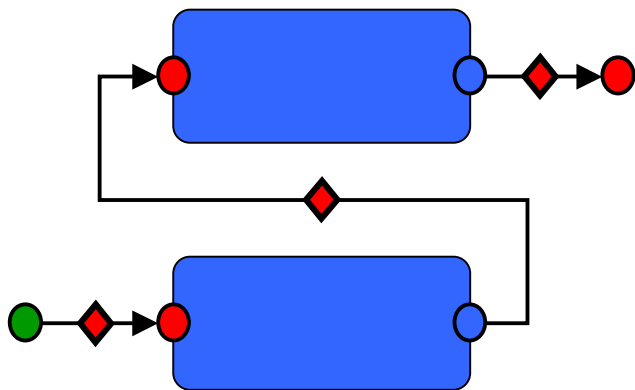


Helicopter Control Software

Clock



Giotto Syntax (Functionality)



```
sensor gps_type GPS uses c_gps_device ;  
actuator servo_type Servo := c_servo_init  
    uses c_servo_device ;
```

output

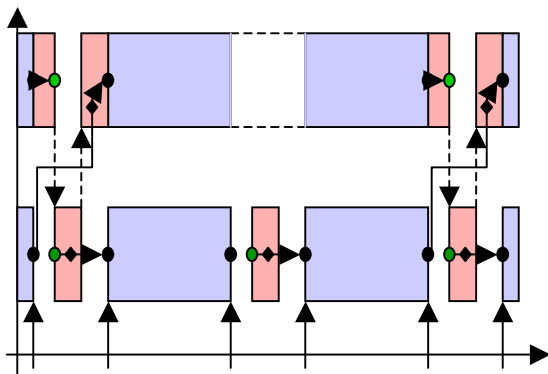
```
ctr_type CtrOutput := c_ctr_init ;  
nav_type NavOutput := c_nav_init ;
```

```
driver sensing (GPS) output (gps_type gps)  
{ c_gps_pre_processing ( GPS, gps ) }
```

```
task Navigation (gps_type gps) output (NavOutput)  
{ c_matlab_navigation_code ( gps, NavOutput ) }
```

...

Giotto Syntax (Timing)



...

```
mode Flight ( ) period 10ms
```

```
{
```

```
actfreq 1 do Servo ( actuating ) ;
```

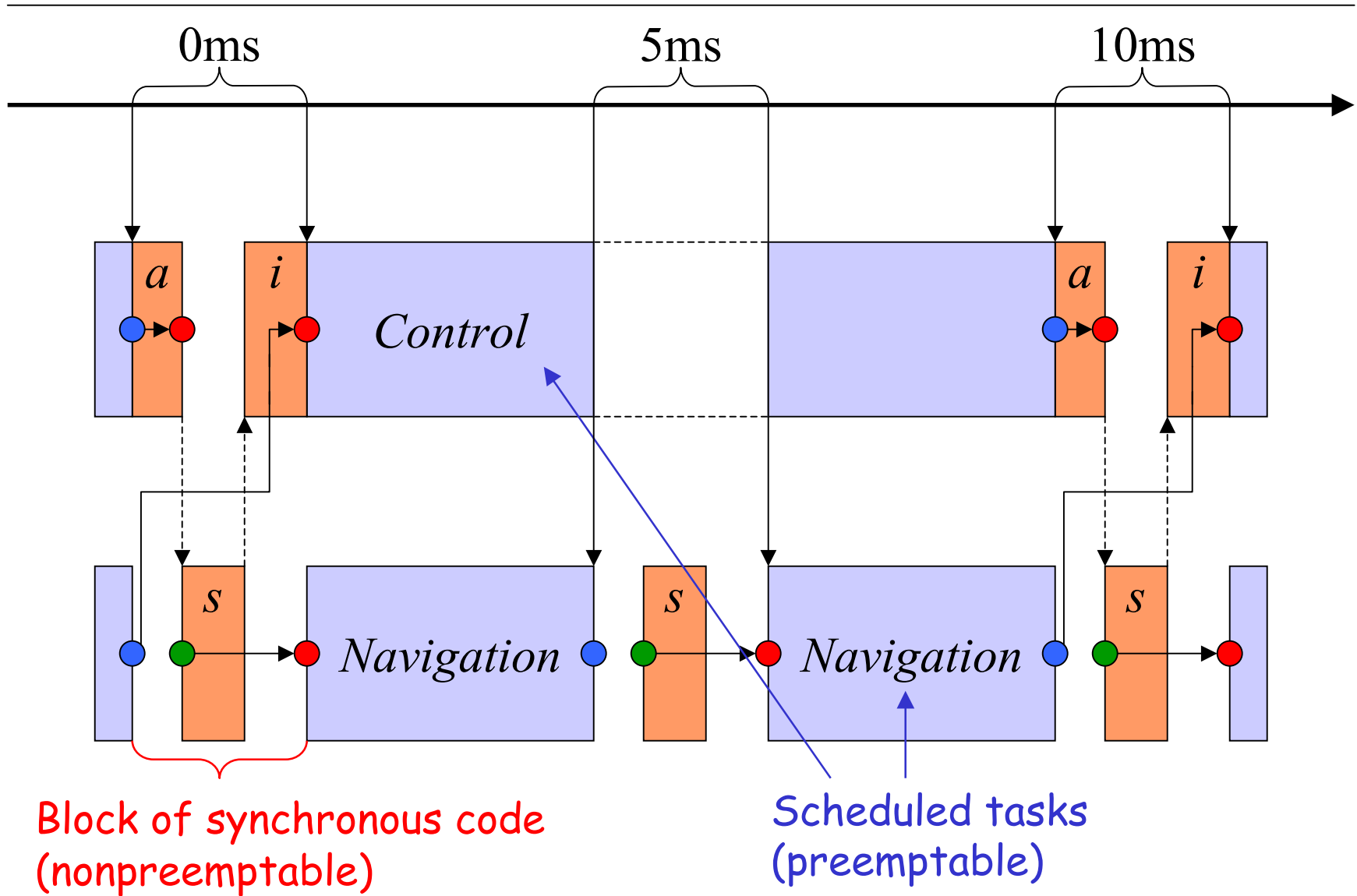
```
taskfreq 1 do Control ( input ) ;
```

```
taskfreq 2 do Navigation ( sensing ) ;
```

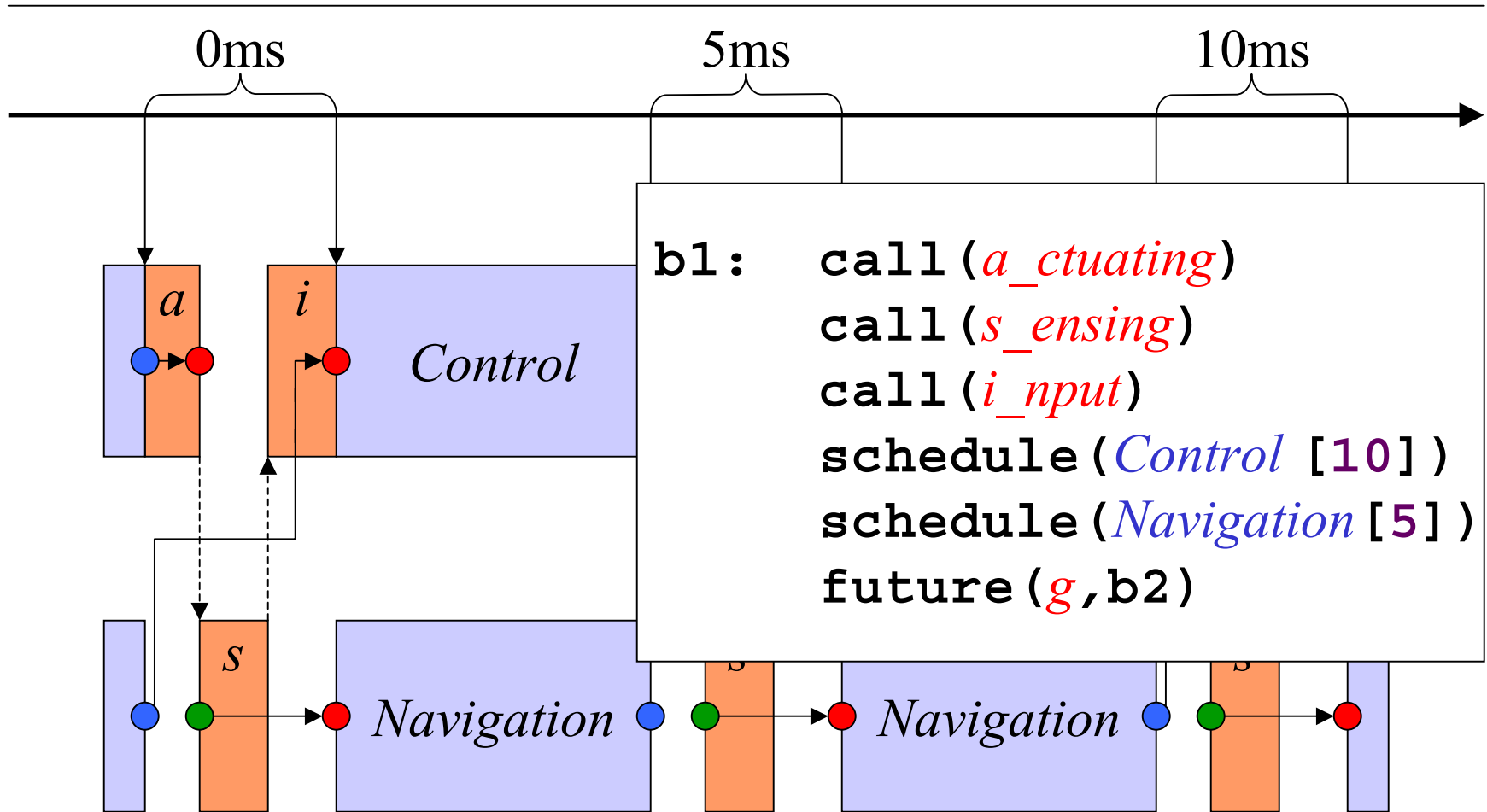
```
}
```

...

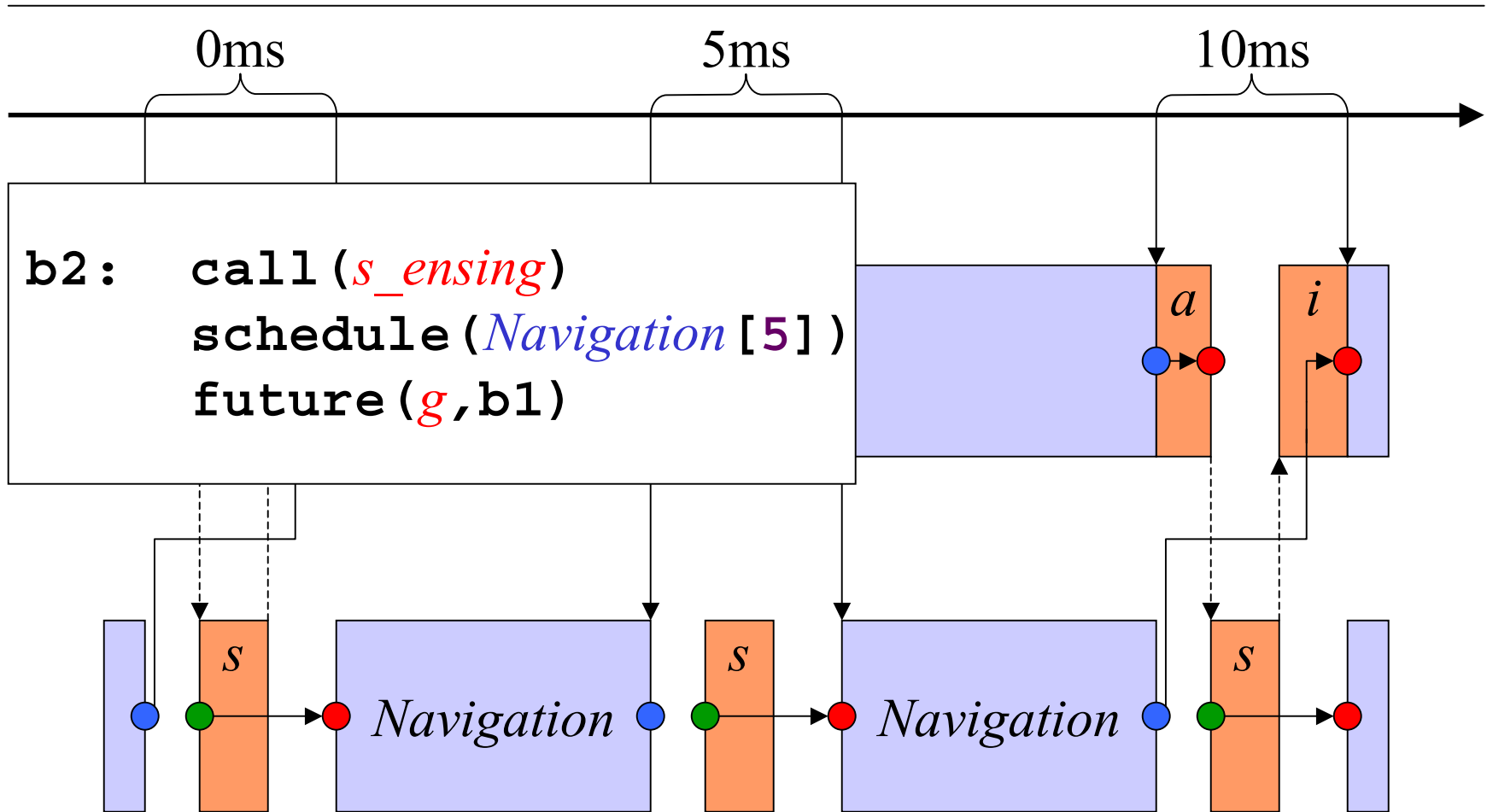
Environment Timeline



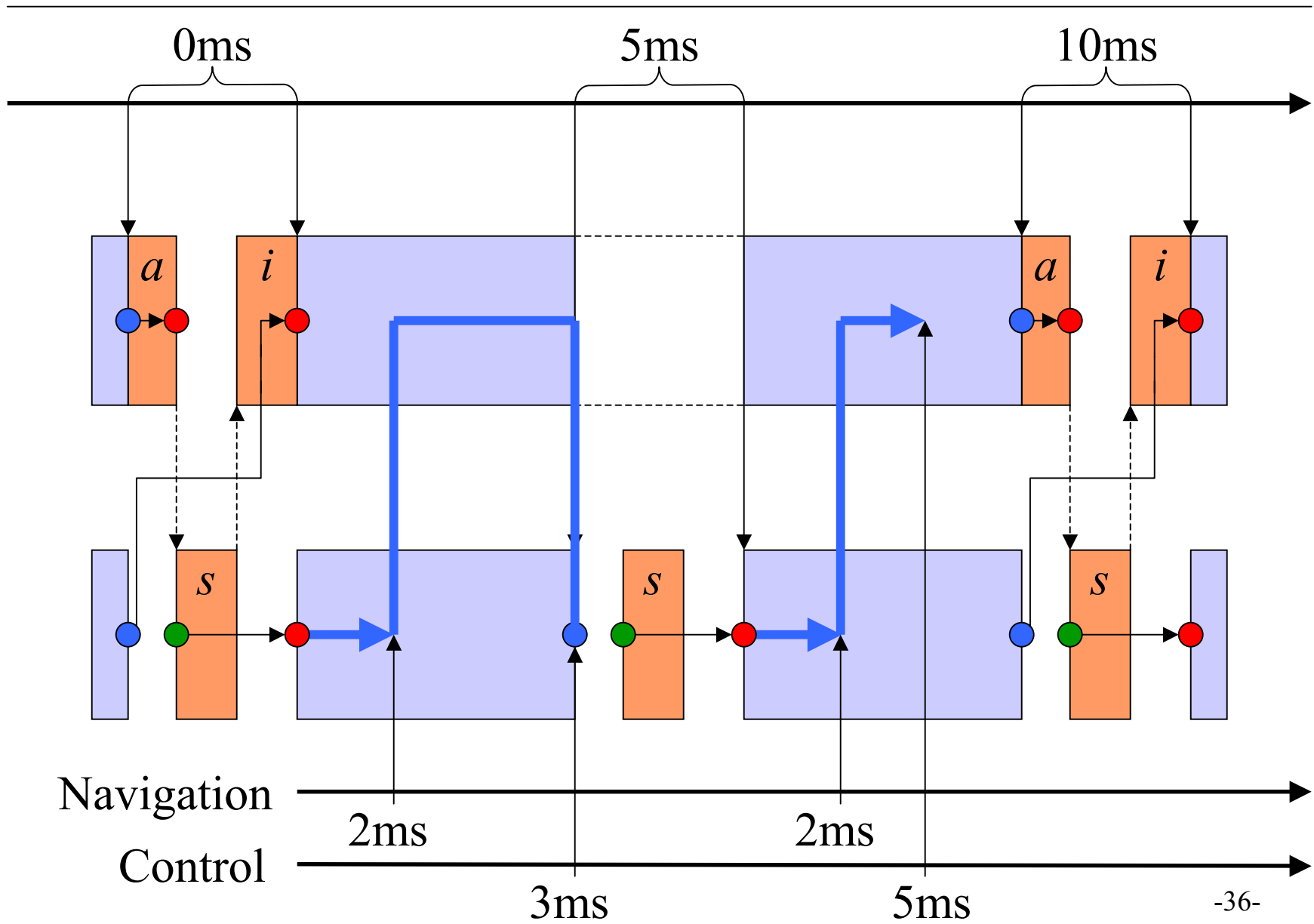
E Code



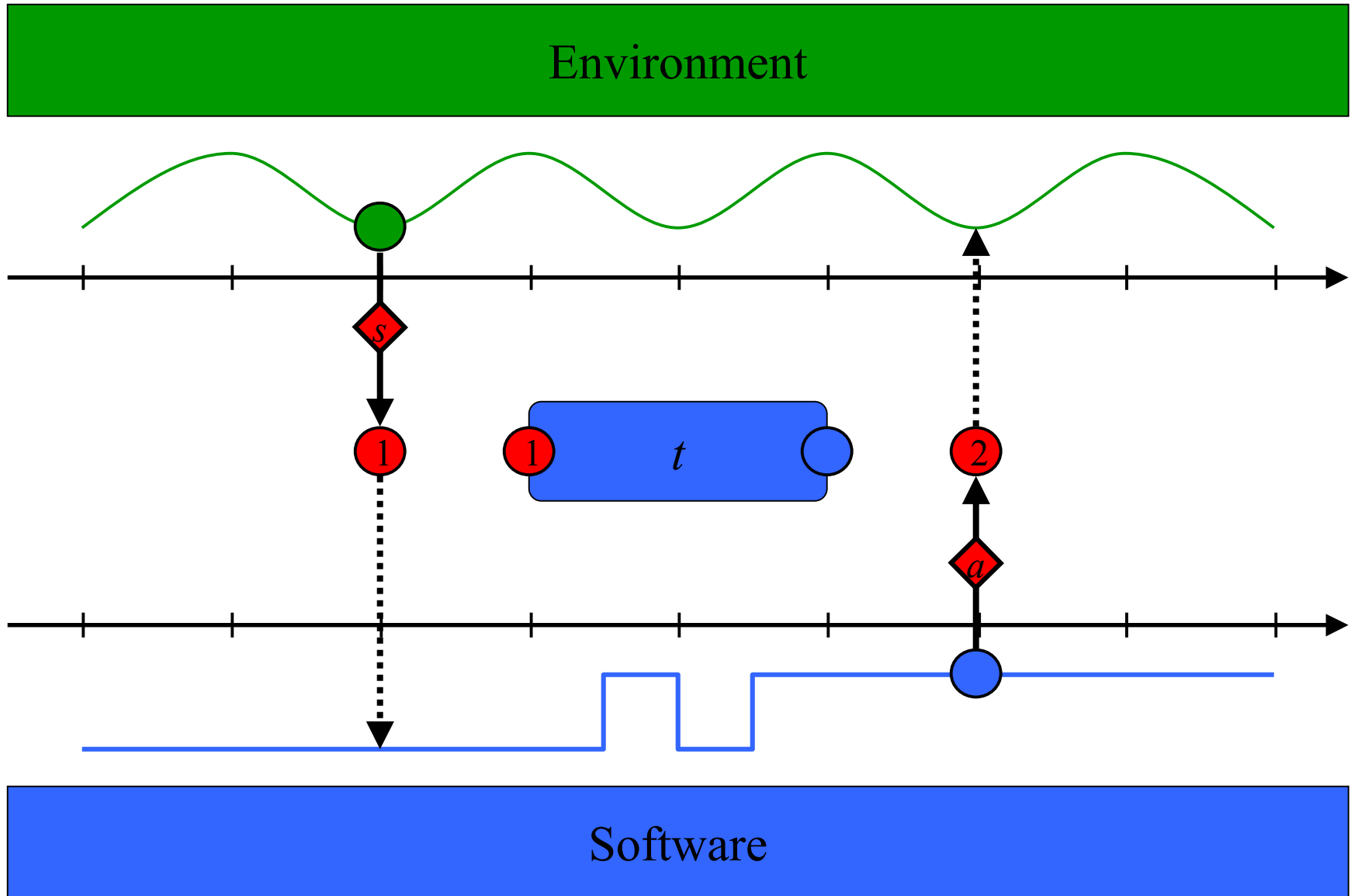
E Code



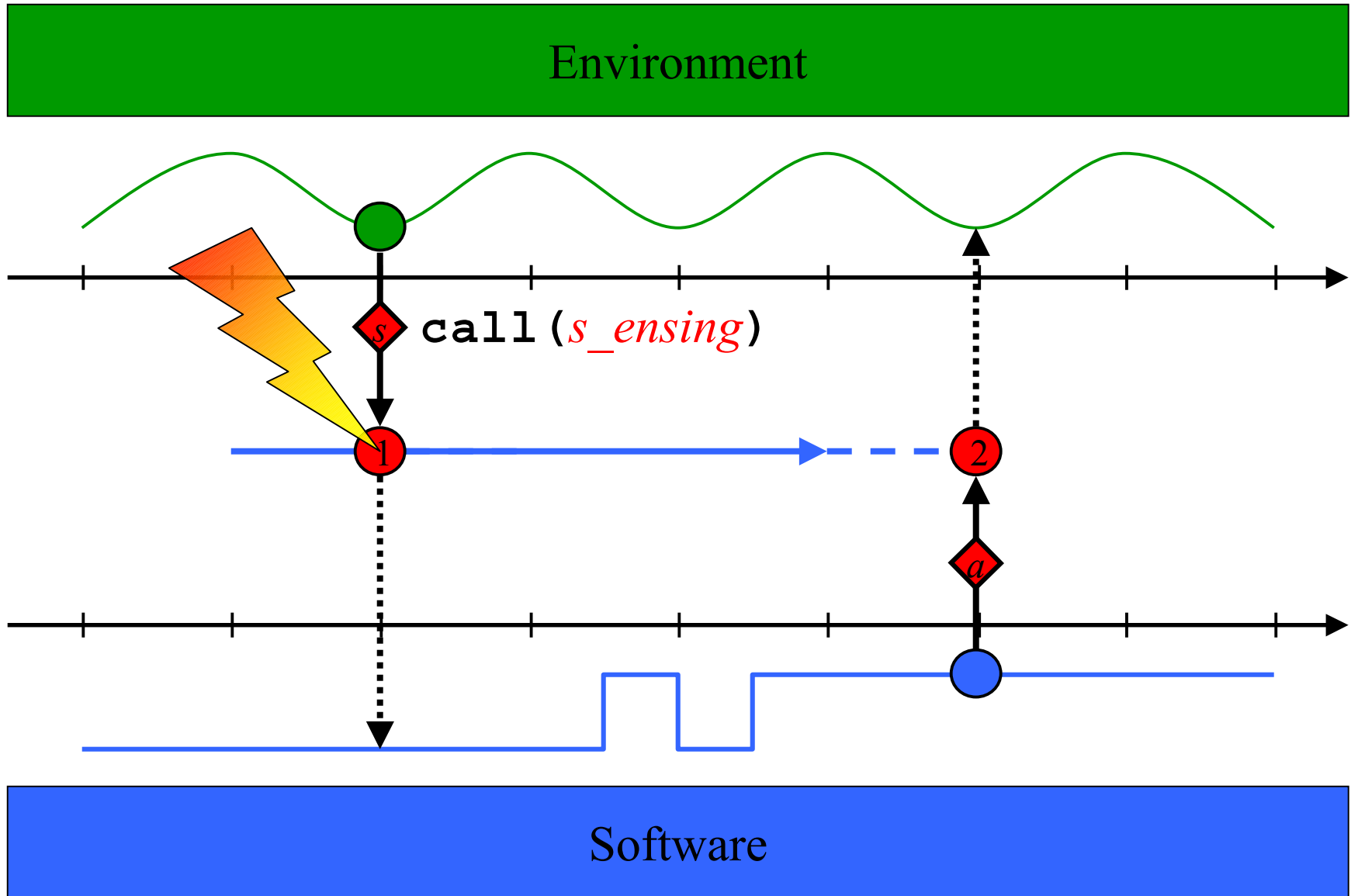
Platform Timeline: EDF



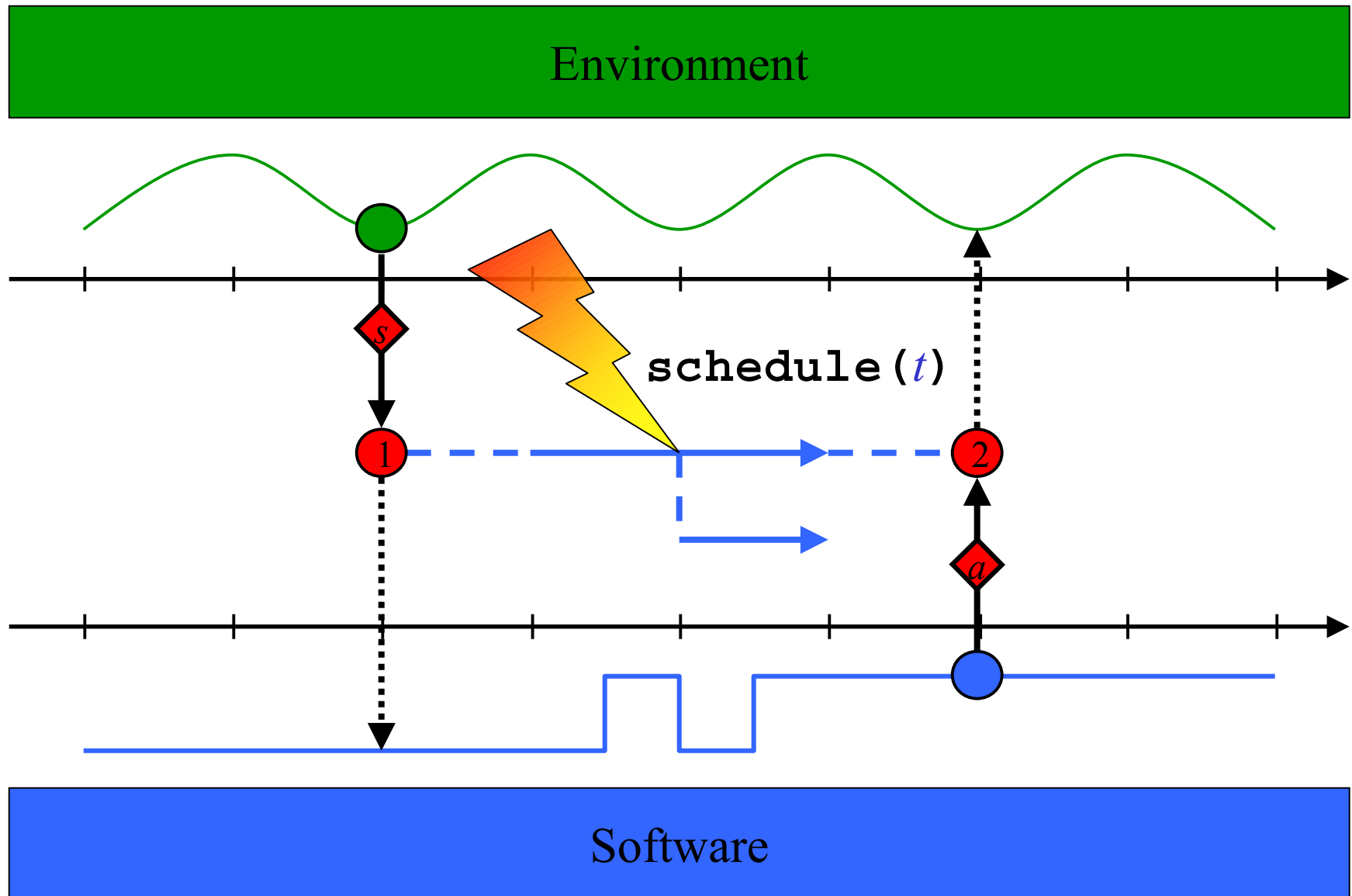
Time Safety



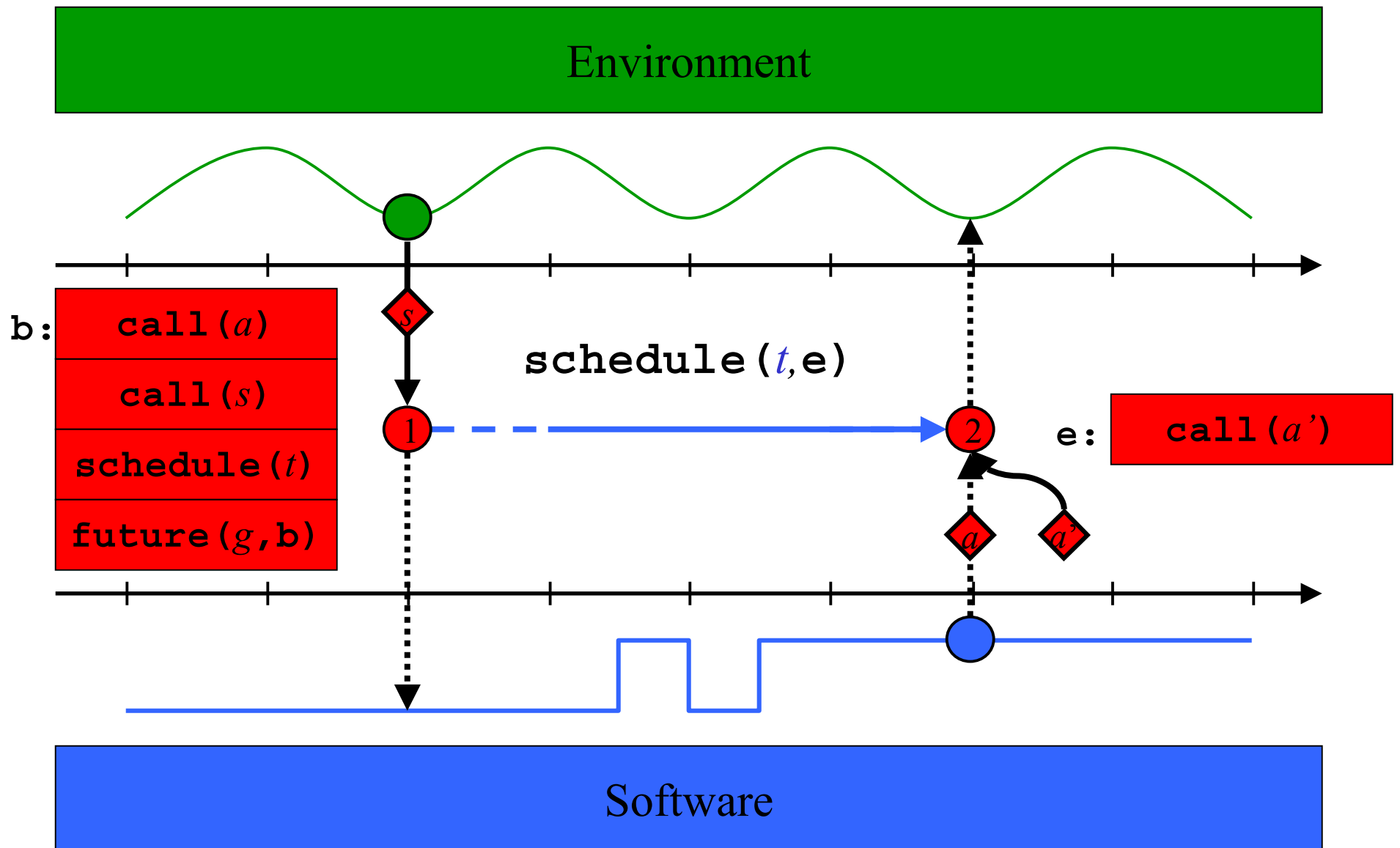
Runtime Exceptions II



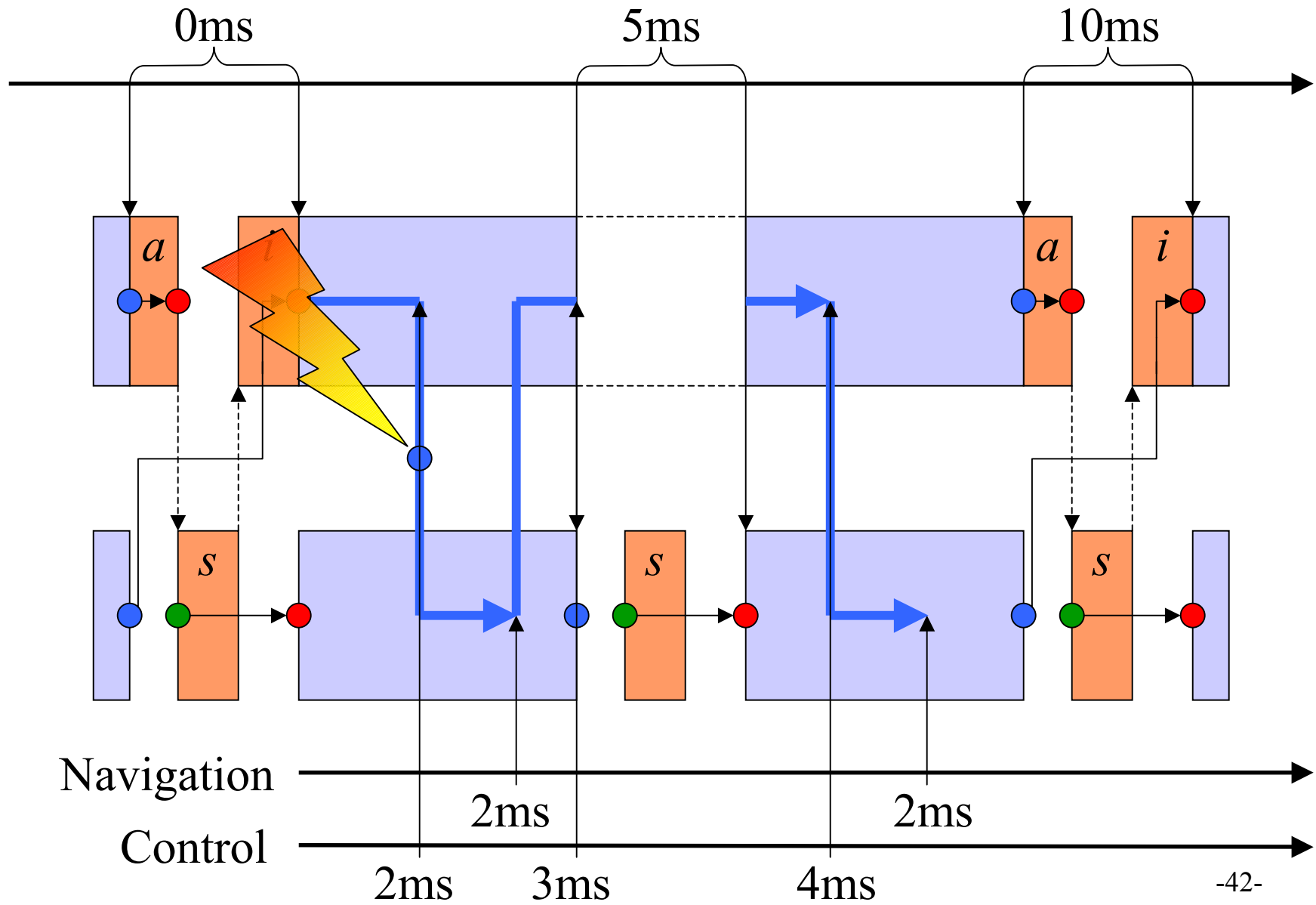
Runtime Exceptions III



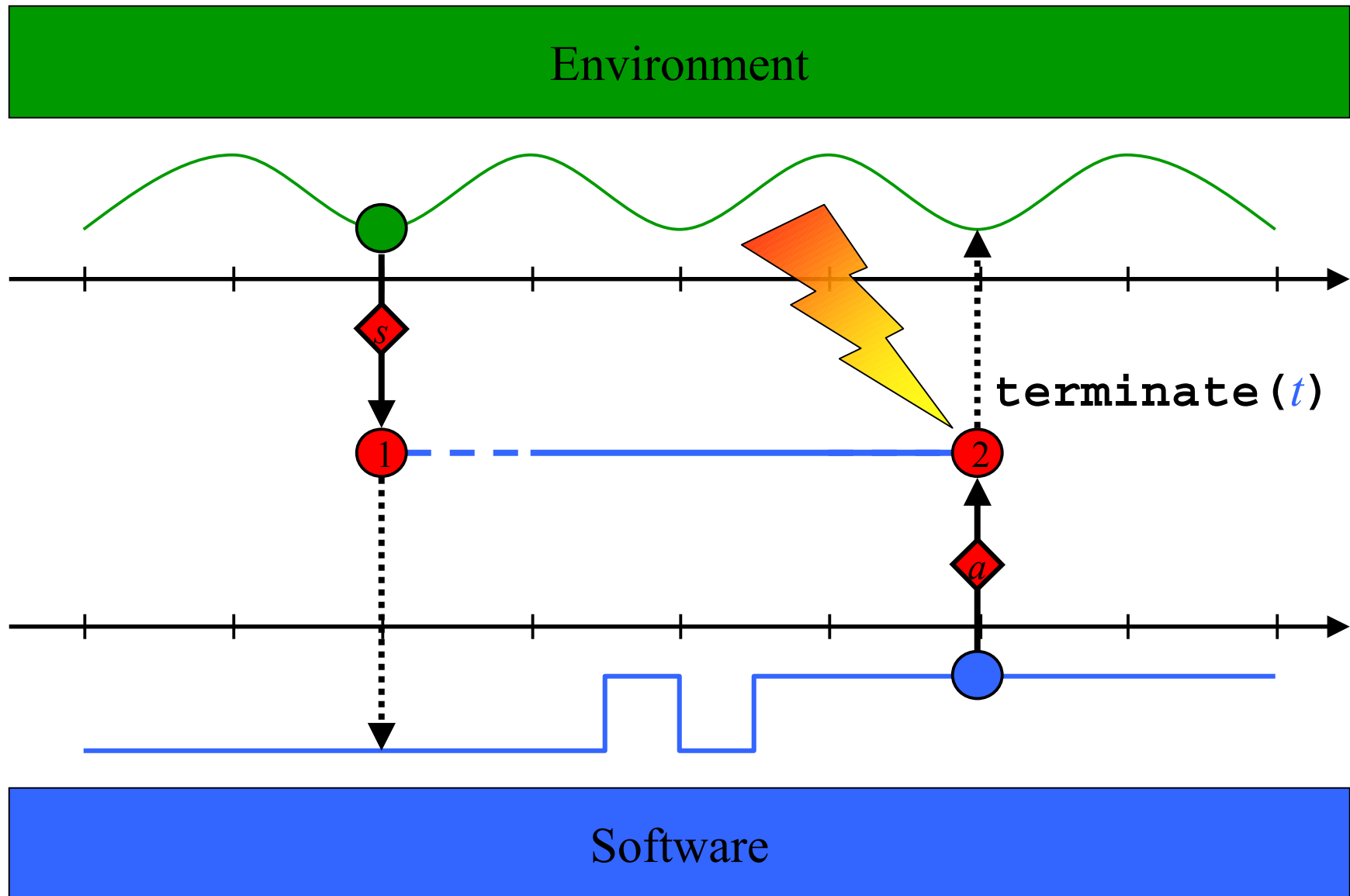
An Exception Handler e



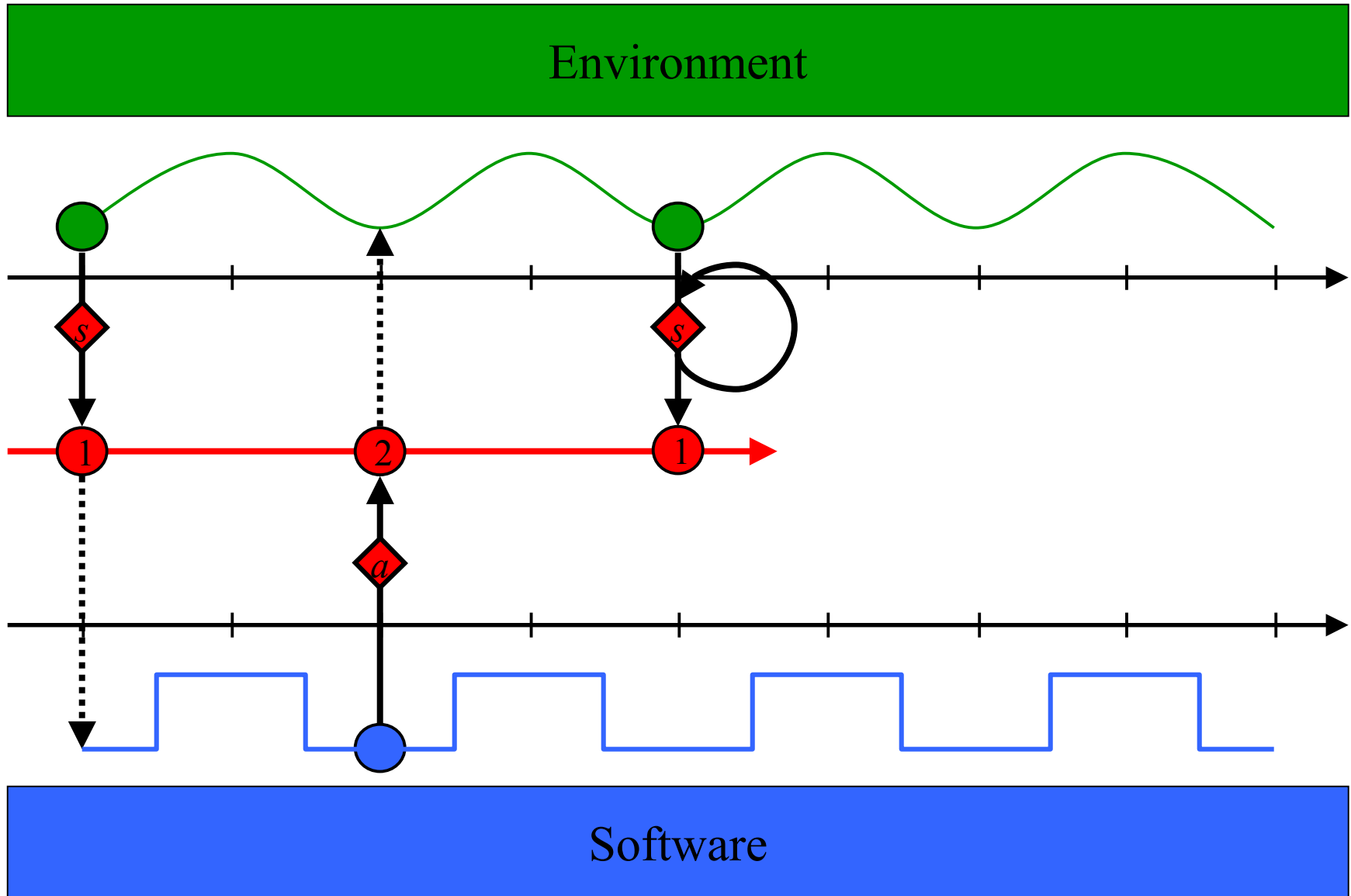
How to Loose Determinism: Task Synchronization



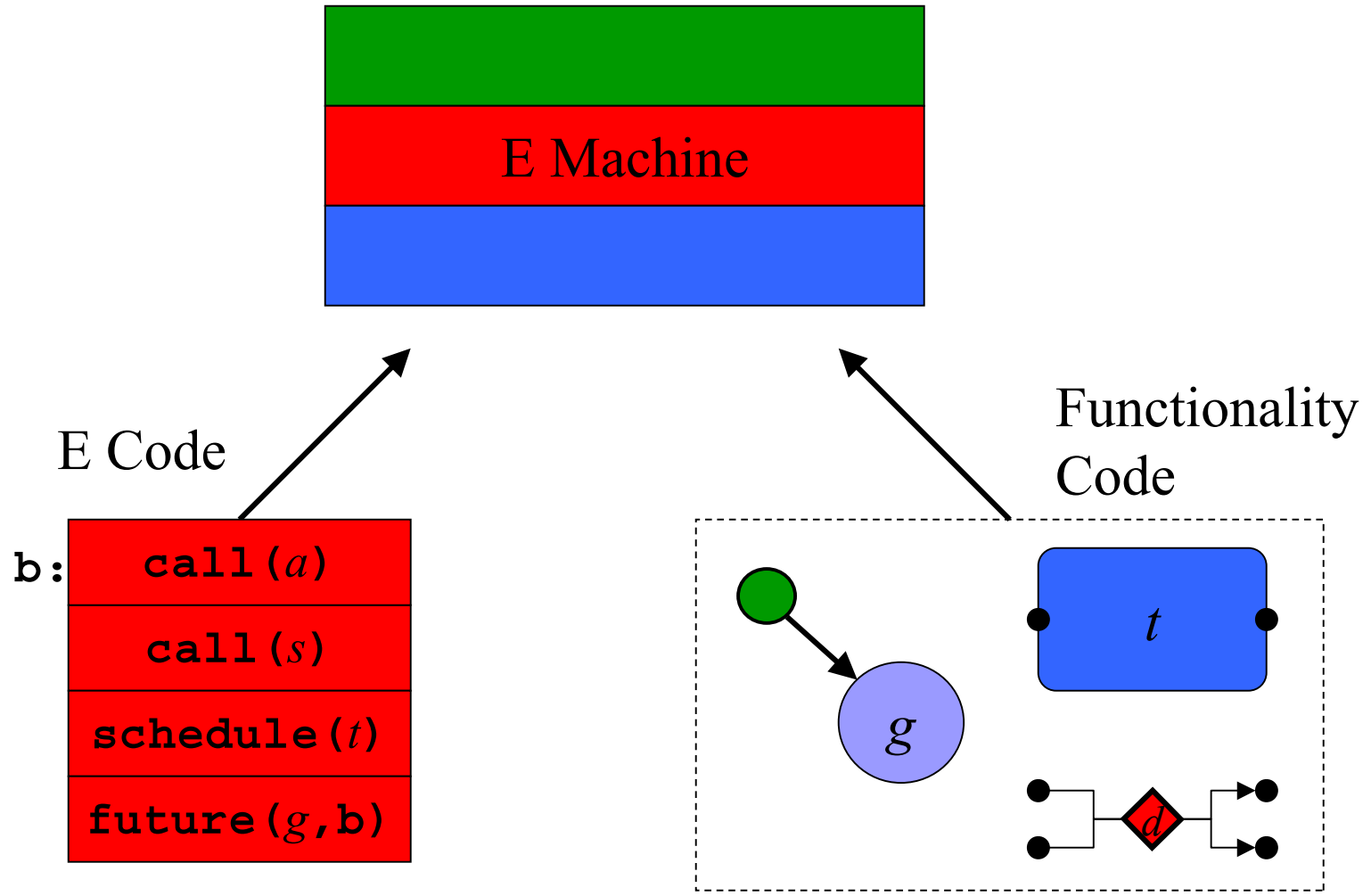
How to Loose Determinism: Termination



Time Liveness: Infinite Traces



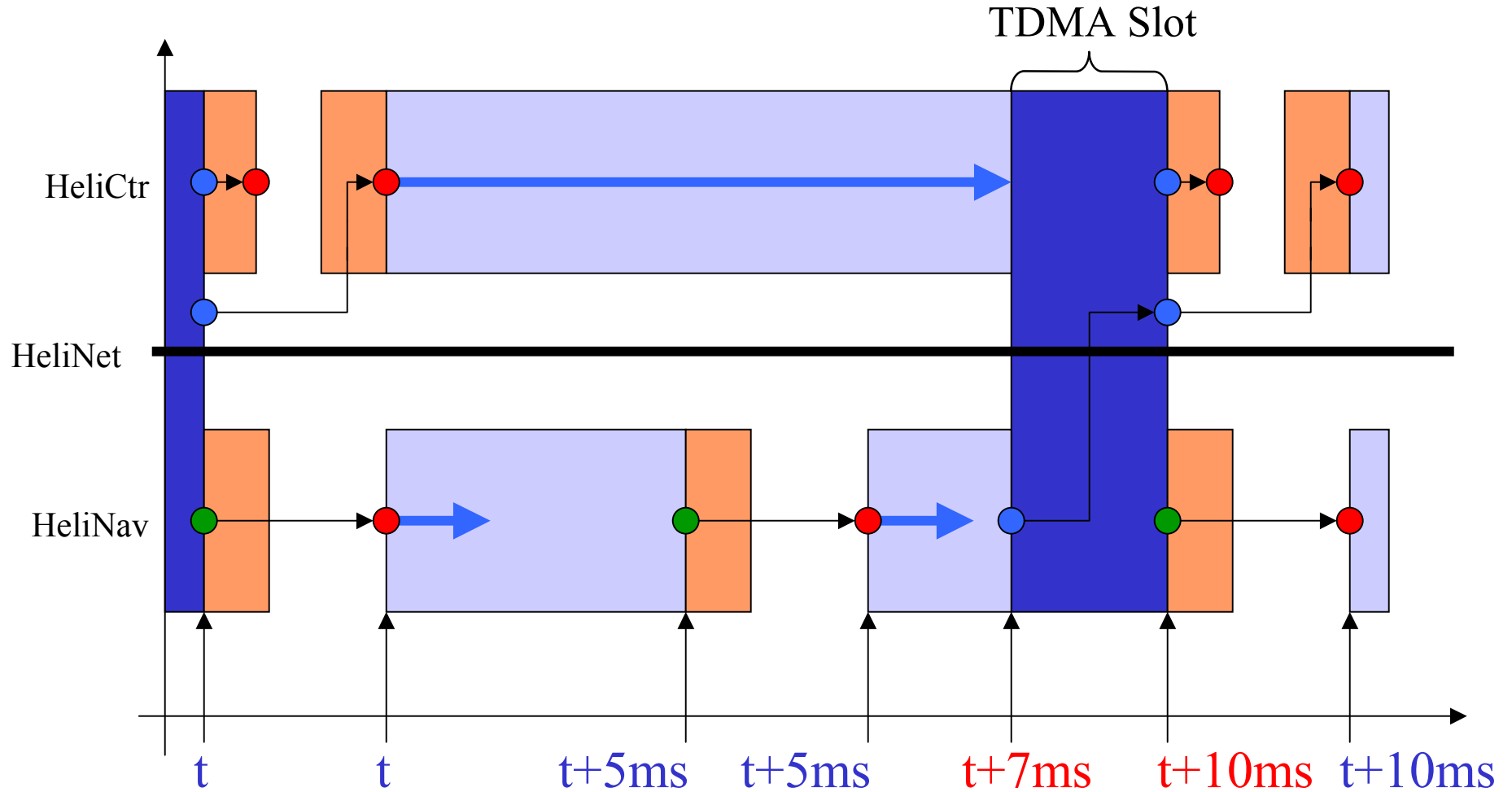
Dynamic Linking



The Berkeley Helicopter

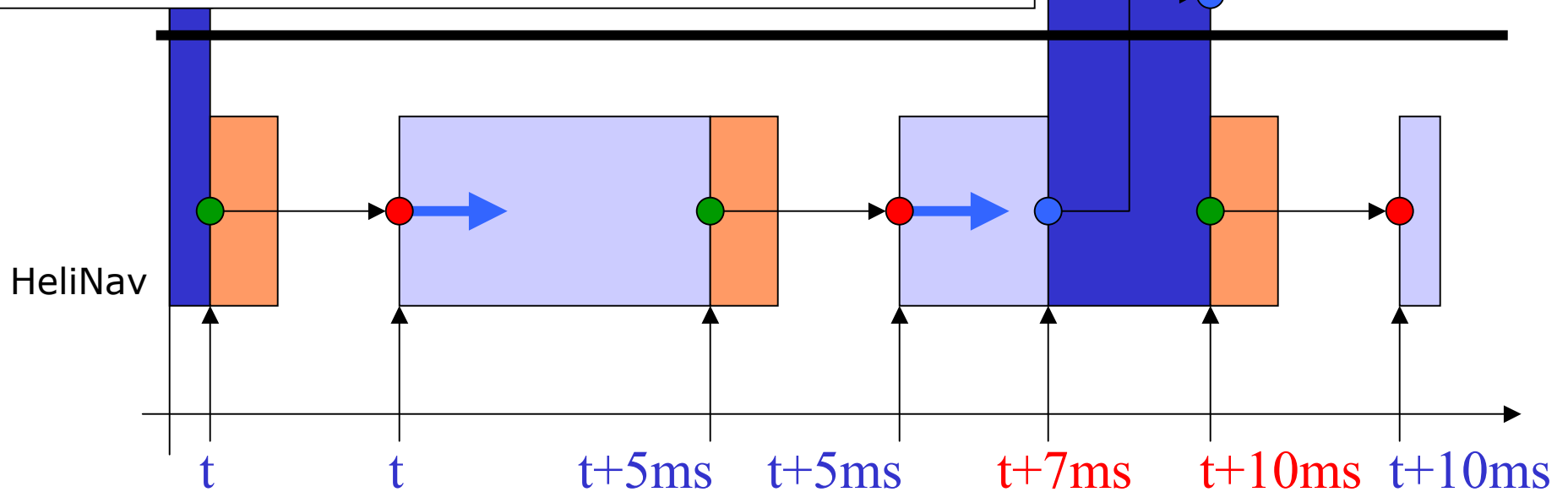


Platform Timeline: Time-triggered Communication



Code Generation for HeliNav

```
b2:  call(s_ensing)  
     schedule(Navigation[2])  
     schedule(Connection[(7,10)])  
     future(g,b1)
```



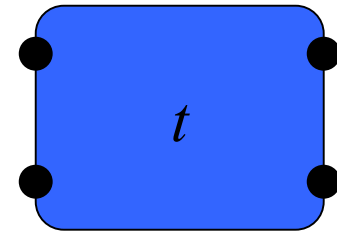
Instructions

Synchronous
Driver:



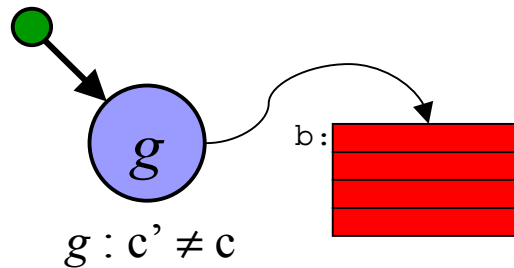
`call (d)`

Scheduled
Task:



`schedule (t)`

Triggering:



`future (g, b)`