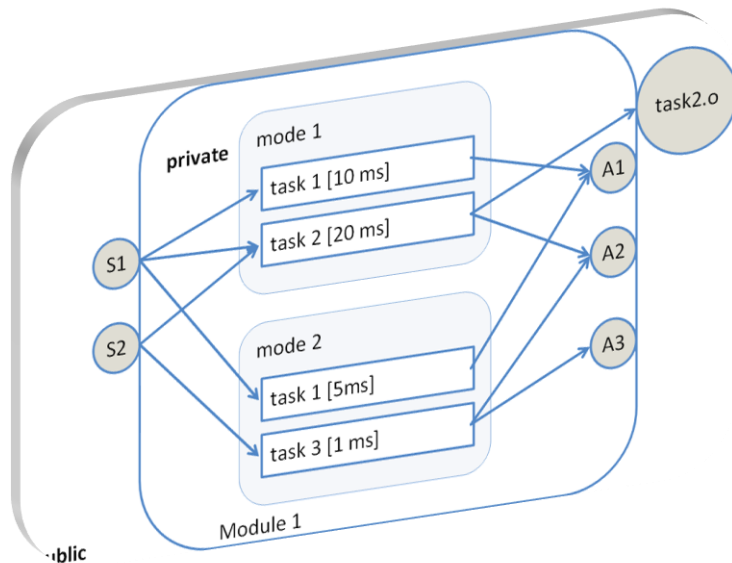# TDL4Gumstix

Gerd Dauenhauer
Patricia Derler

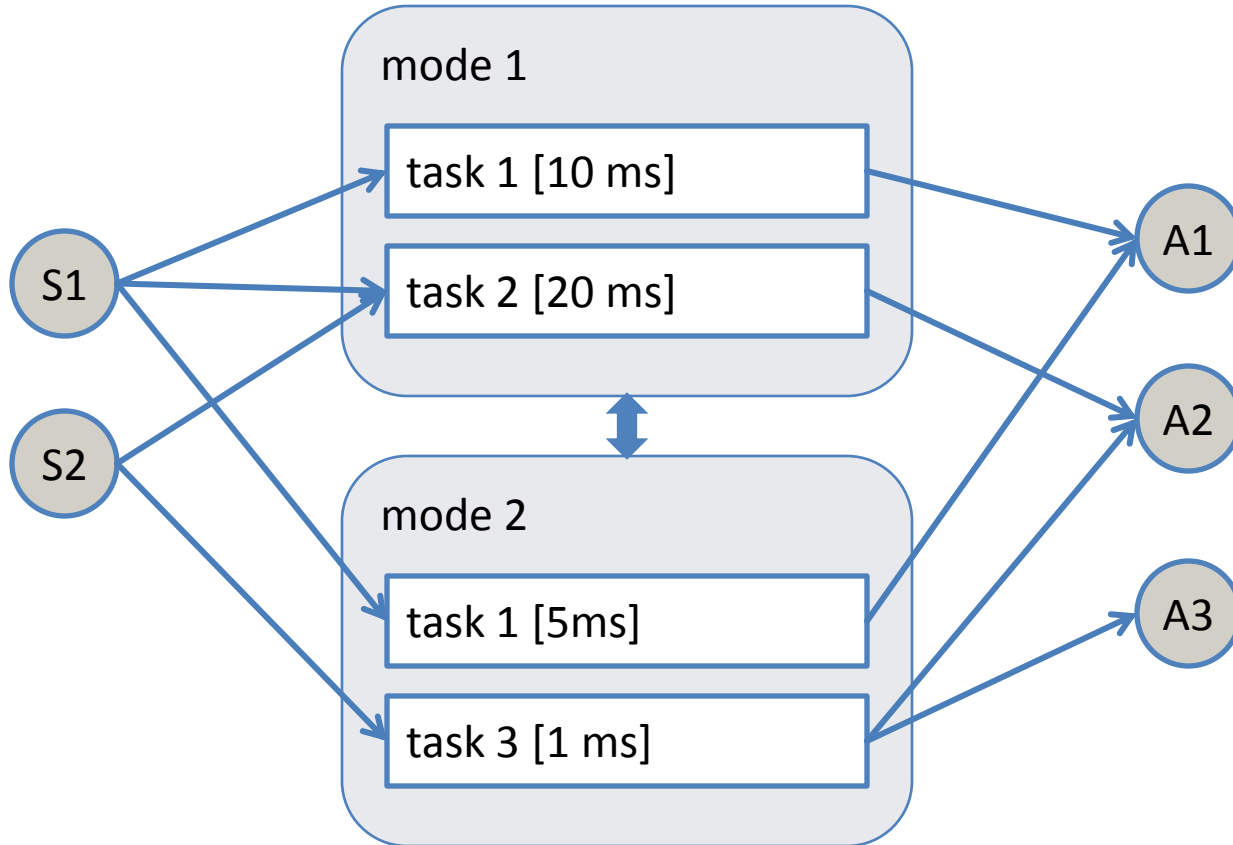# TDL – Timing Definition Language

TDL =

Giotto abstractions and concepts

- Logical timing
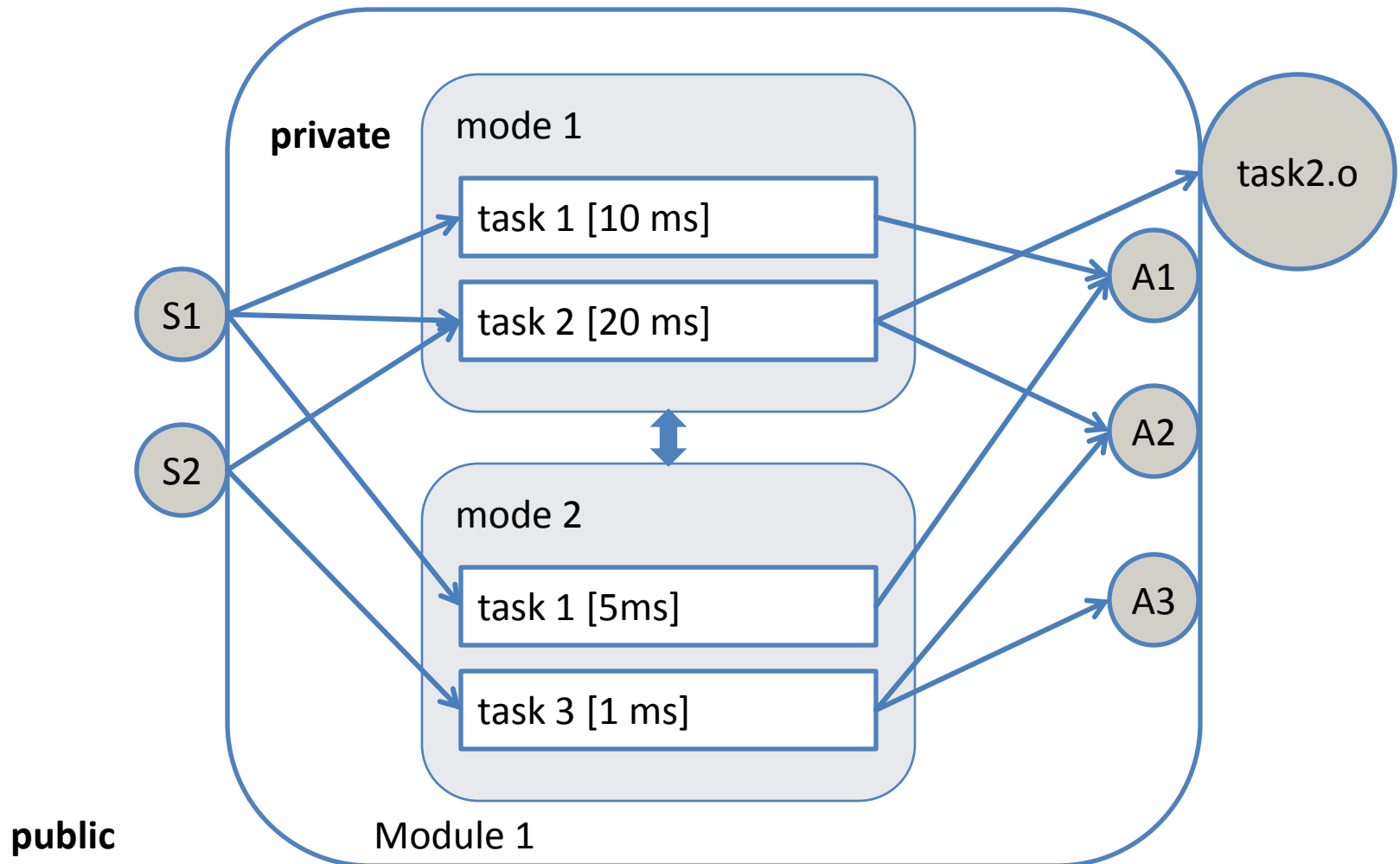- Separation of timing and functionality
- Modes and mode switches

+ modules as

- units of parallel composition and
- units of distribution

TDL – An Example I

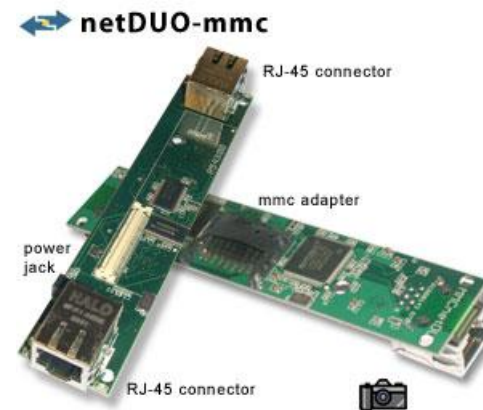# TDL – An Example I

TDL – An Example II

```
module Module {

    sensor double s1 uses getS1;
    sensor double s2 uses getS2;
    actuator double a1 := 1 uses setA1;
    actuator double a2 := 2uses setA2;
    actuator double a3 := 3 uses setA3;

    task t1 {
      input      double port1;
      output     double port2;
      uses t1Impl(port1,port2);
    }

    task t2 {
      input      double port1;
                 double port2;
      output     double port3;
      uses t2Impl(port1,port2,port3);
    }

    task t3 {
      input      double port1;
      output     double port2;
                 double port3;
      uses t3Impl(port1,port2,port3);
    }
```

```
start mode mode1 [period=10 ms] {
    task
       [freq=10] task1 {port1 := s1;}
       [freq=20] task2 {port2 := s2;port1 := s1;}
    actuator
       [freq=10] a2 := task2.port3;
       [freq=10] a1 := task1.port2;
    mode
        [freq=10] if s1 = 100 then mode2;
  }

  mode mode2 [period=1 ms] {
    task
       [freq=5] task3 {port1 := s2;}
       [freq=1] task1 {port1 := s1;}
    actuator
       [freq=5] a2 := task3.port2;
       [freq=5] a3 := task3.port3;
       [freq=1] a1 := task1.port2;
  }

}
```
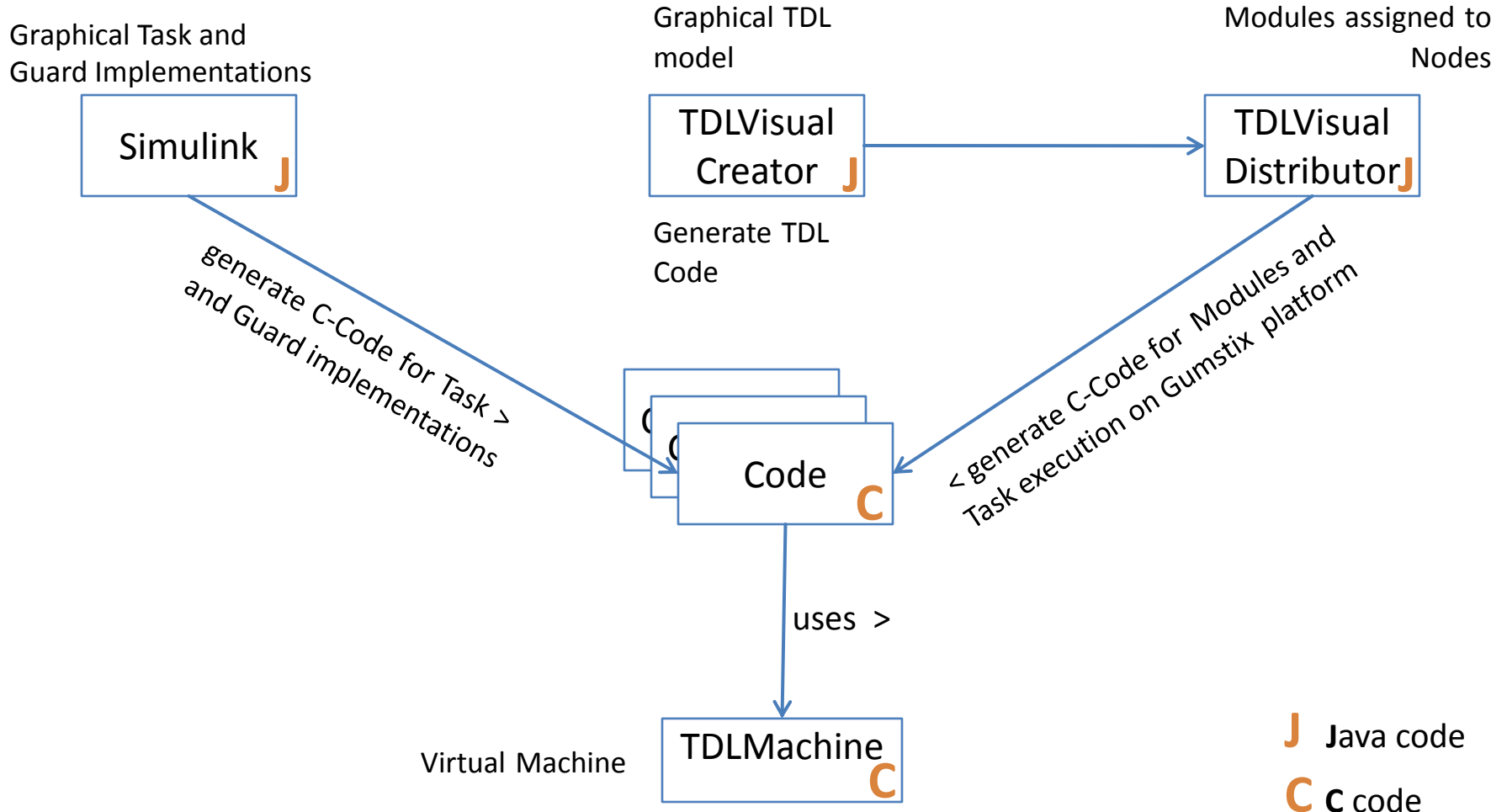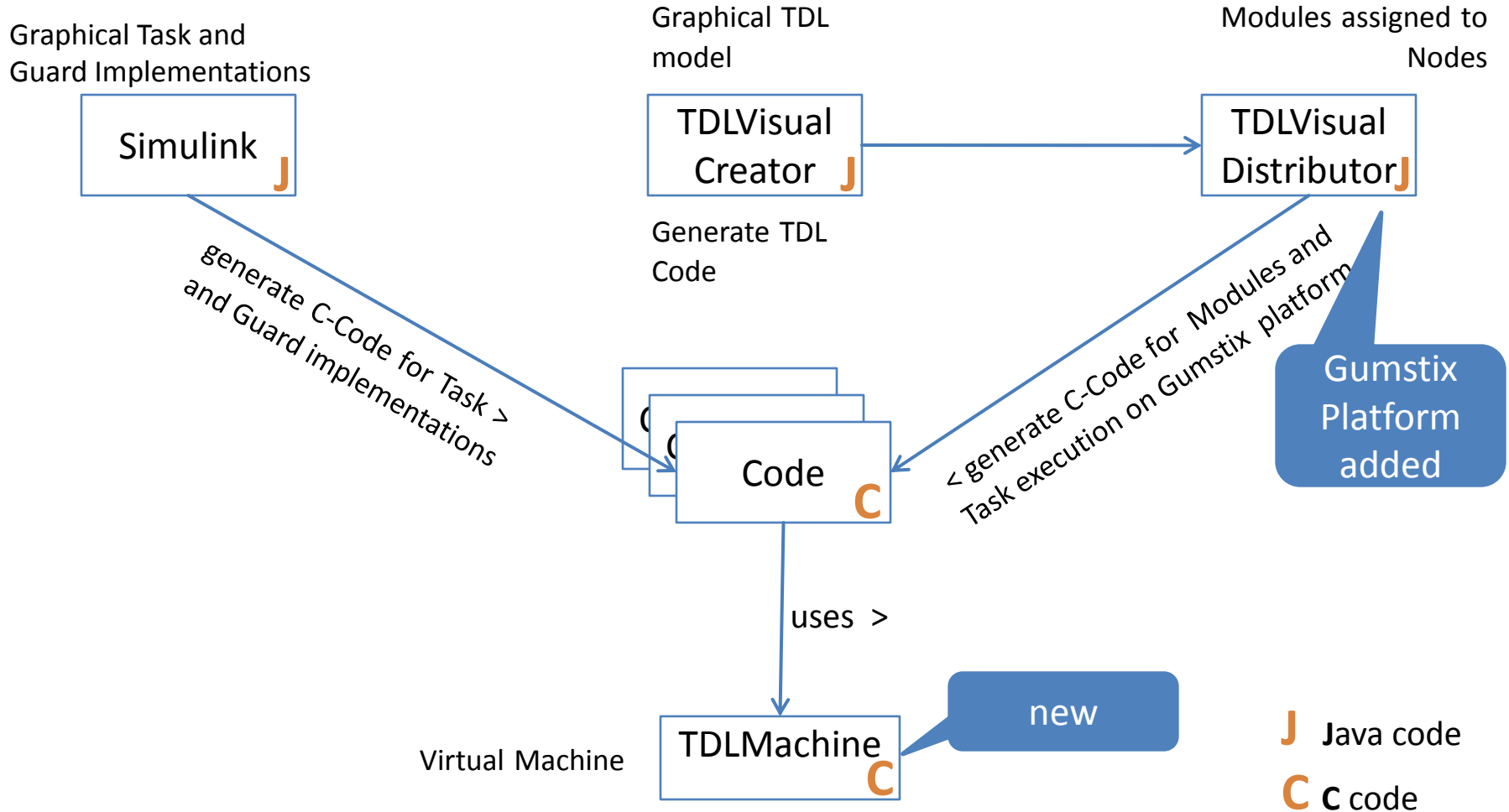
Gumstix

- Gumstix, inc. maker of the world's smallest full function miniature computers(FFMC,) brings gum stick size, 20mm x 80mm x 8mm Linux motherboards to designers providing easy design flexibility and open source at nearly half the price of and a third the size of all other offerings.

- Communication over Ethernet
(Dual ethernet and mmc slot)

TDL – Tool Chain

Graphical Task and
Guard Implementations

Graphical TDL
model

Modules assigned to
Nodes

Simulink **J**

TDLVisual
Creator **J**

TDLVisual
Distributor**J**

Generate TDL
Code

generate C-Code for Task >
and Guard implementations

< generate C-Code for Modules and
Task execution on Gumstix platform

Code **C**

uses >

Virtual Machine

TDLMachine **C**

J **J**ava code

C **C** code

TDL – Tool Chain

Graphical Task and
Guard Implementations

Graphical TDL
model

Modules assigned to
Nodes

Simulink **J**

TDLVisual
Creator **J**

TDLVisual
Distributor**J**

Generate TDL
Code

generate C-Code for Task
and Guard implementations >

< generate C-Code for Modules and
Task execution on Gumstix platform

Code **C**

Gumstix
Platform
added

uses >

Virtual Machine

TDLMachine **C**

new

J **J**ava code

C **C** code

TDL and Gumstix

Generated Code:

Matlab RT Workshop:
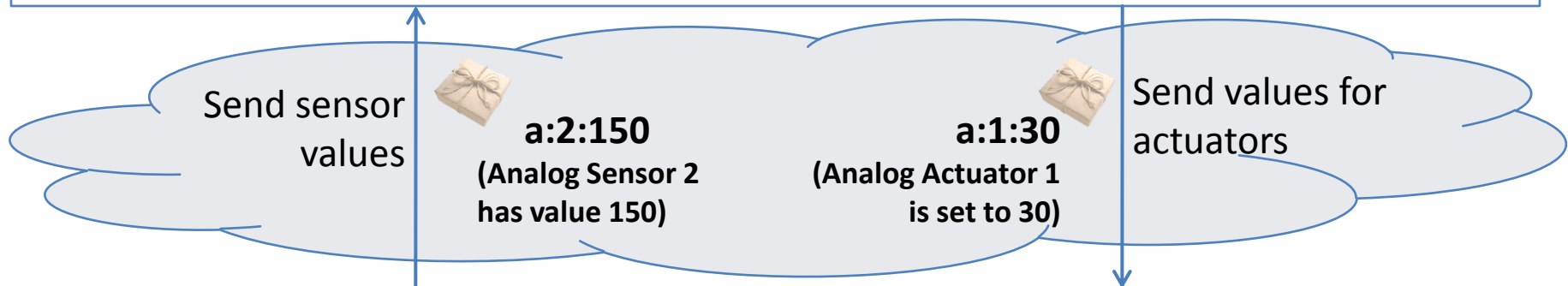
- Task Implementierungen und Guard Implementierungen

TDL Tools:

- Wrapper für die von Simulink generierten Implementierungen
- tdl_main
- Module Code

Other:

- **TDL Machine**

… running on Gumstix

Send sensor values

a:2:150
**(Analog Sensor 2
has value 150)**

a:1:30
**(Analog Actuator 1
is set to 30)**

Send values for actuators

… running on any other machine

Java Client for sending sensor output and receiving actuator input (UDP Packets)