

---

# Modelling and Analysis Suite for Real-time systems

## Future extensions: MAST-2

**By:**

**Michael González Harbour**  
**Universidad de Cantabria, Spain**  
**<http://mast.unican.es/>**

# 1. Introduction: Objectives of MAST-2

## Align names with MARTE

## Partitioned scheduling

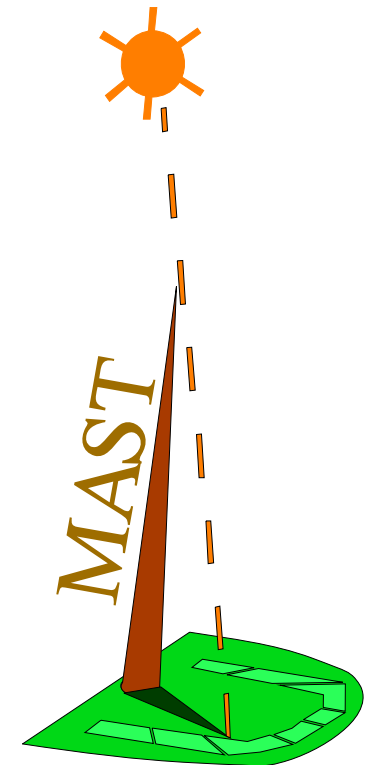
- support for ARINC 653 systems with hierarchical scheduling
  - fixed priorities on top of timed partitions
- support for TTP networks

## Network switches

- support for AFDX deterministic ethernet

## Resource reservations

- virtual resources as a new primary scheduler



# Objectives of MAST-2 (cont'd)

---

## Enhance modelling capabilities

- support for thread locking from a transaction
  - enable modelling synchronous RPC
- enhanced modelling of timers
  - allow multiple timers
- new application requirements
  - queue size requirements

## 2. Partitioned scheduling

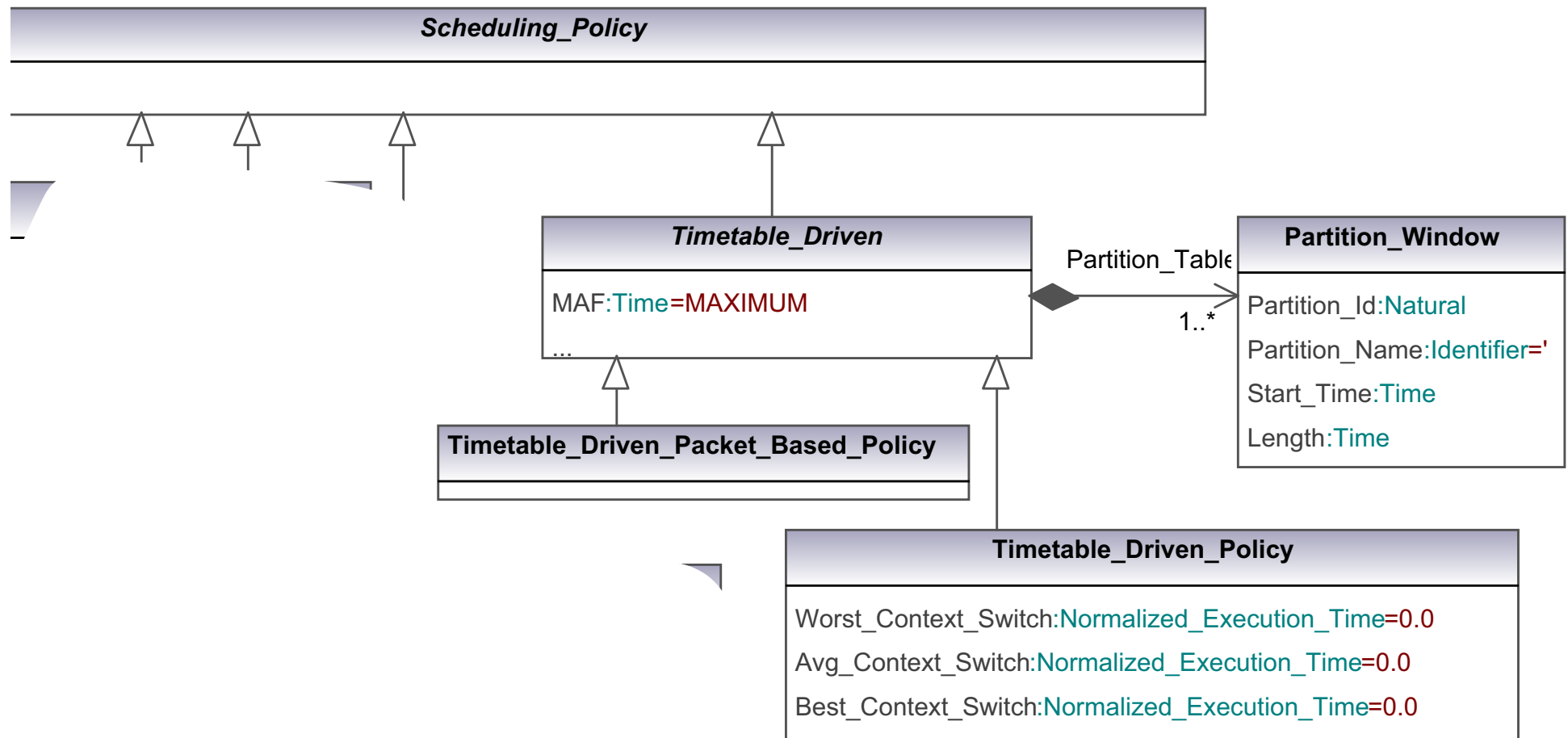
---

### Support for ARINC 653 systems with hierarchical scheduling

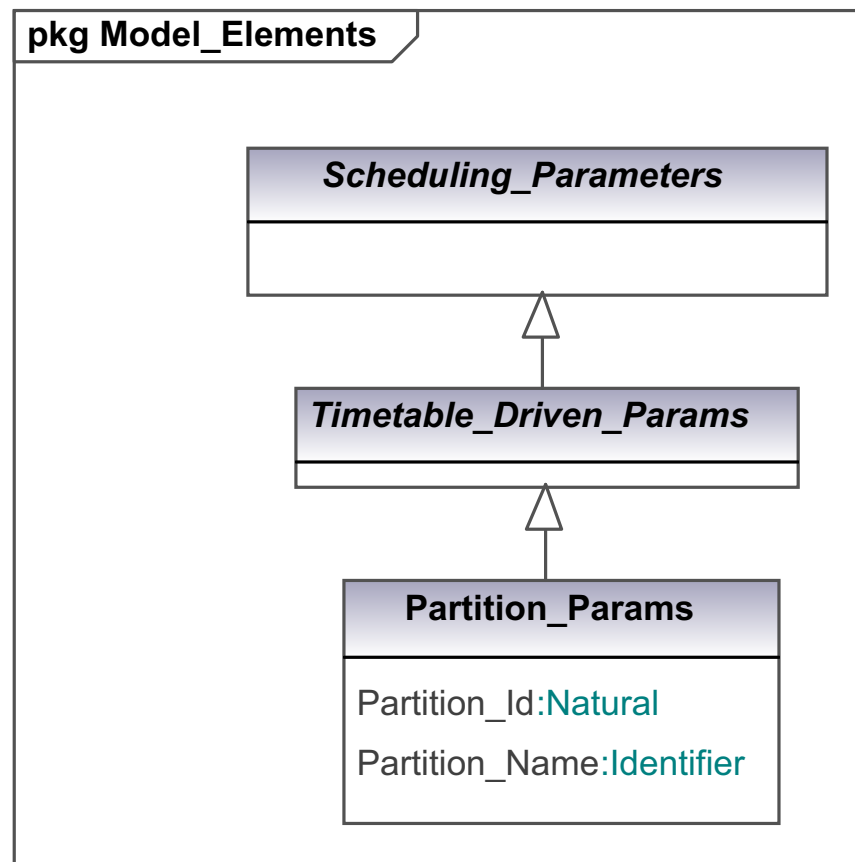
- fixed priorities on top of timed partitions
- EDF?

### Support for TTP networks

# Partitioned scheduling



# Partition-Based Scheduling Parameters



# 3. Network switches and routers

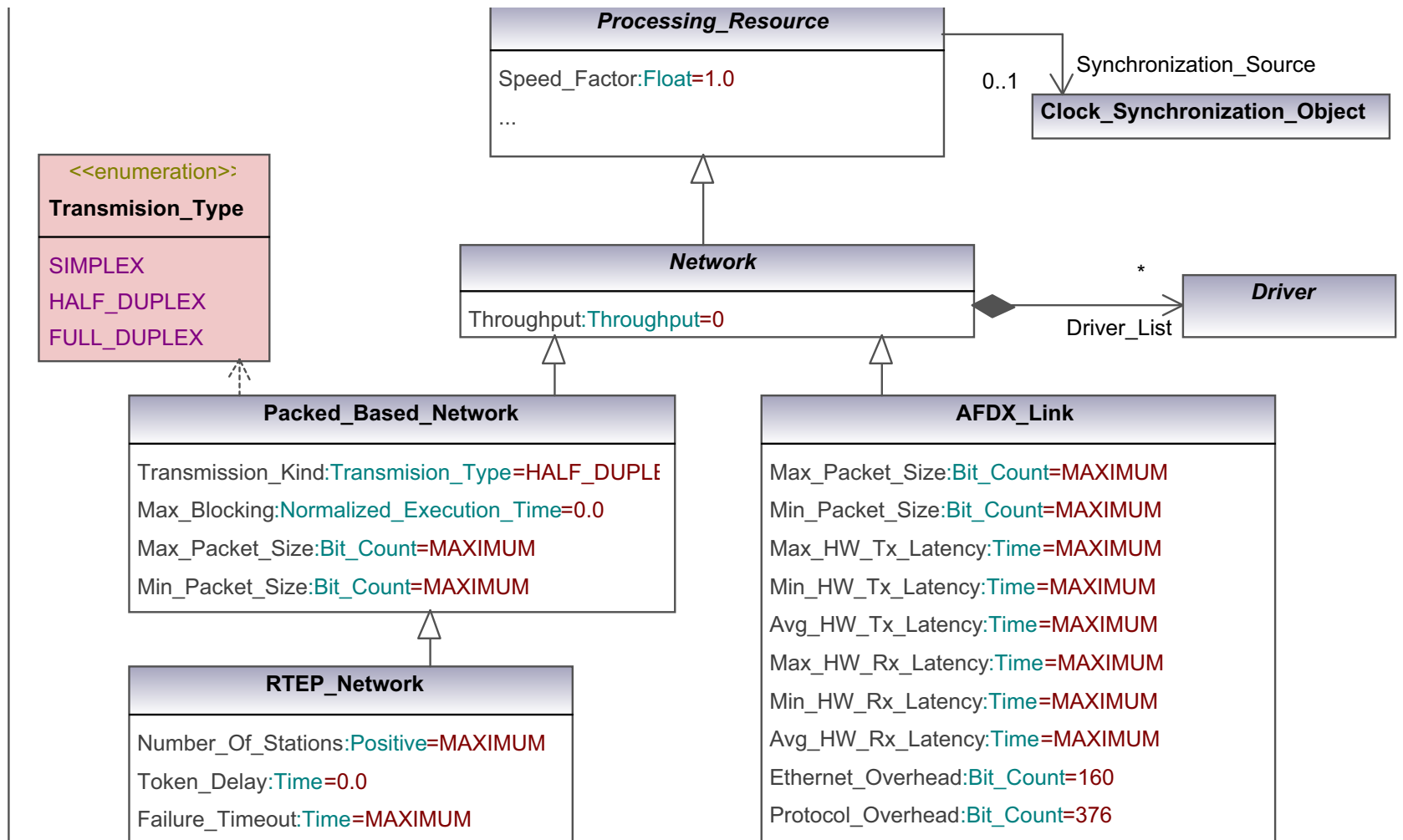
---

## General-Purpose shared memory switches

- support for AFDX deterministic ethernet

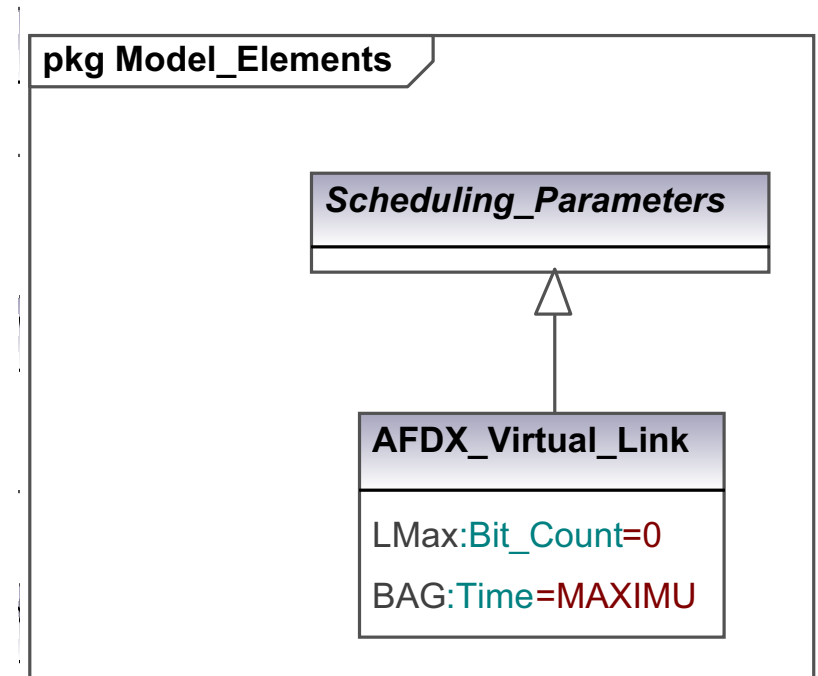
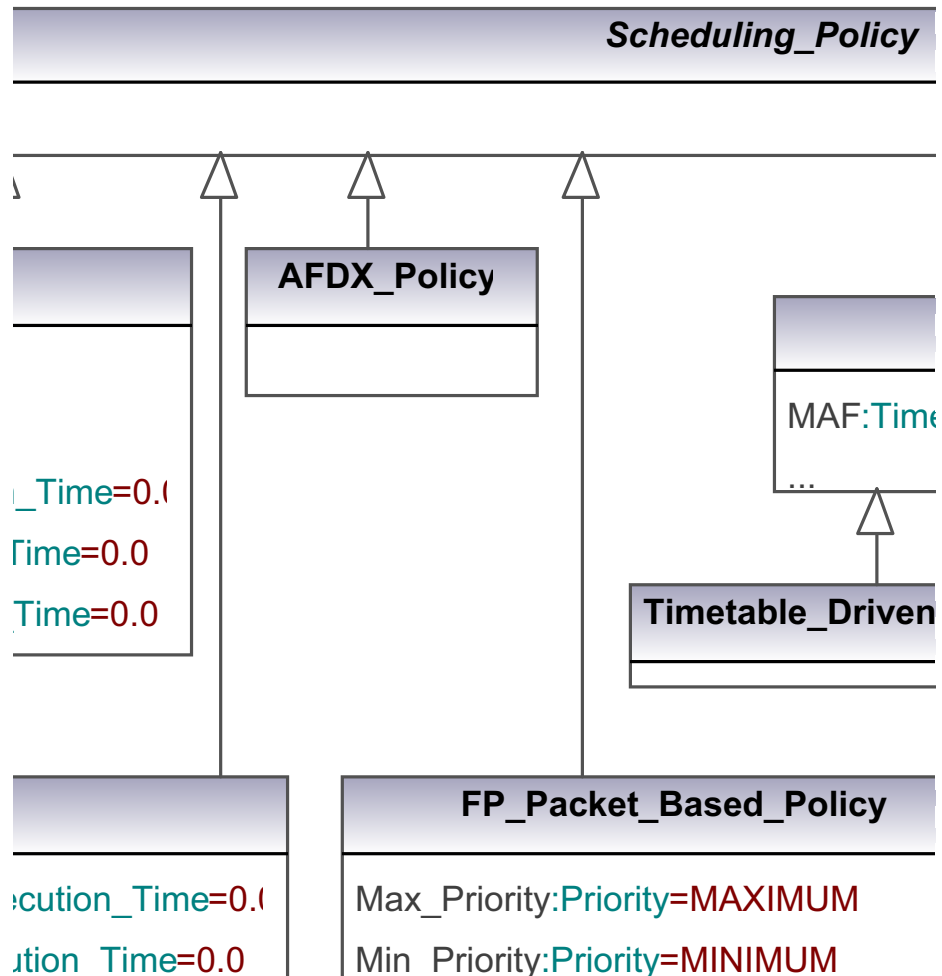
## General-purpose routers

# Networks



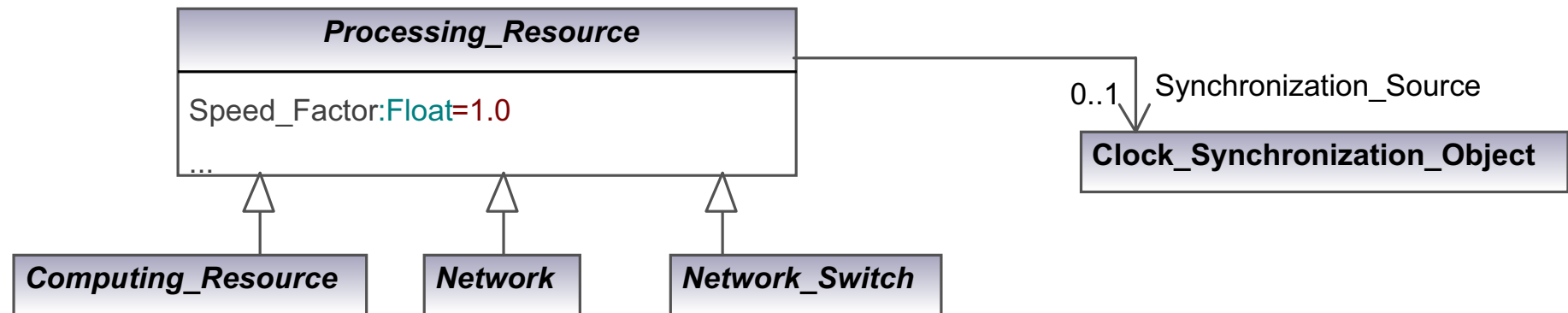


# Scheduling Policy and Parameters

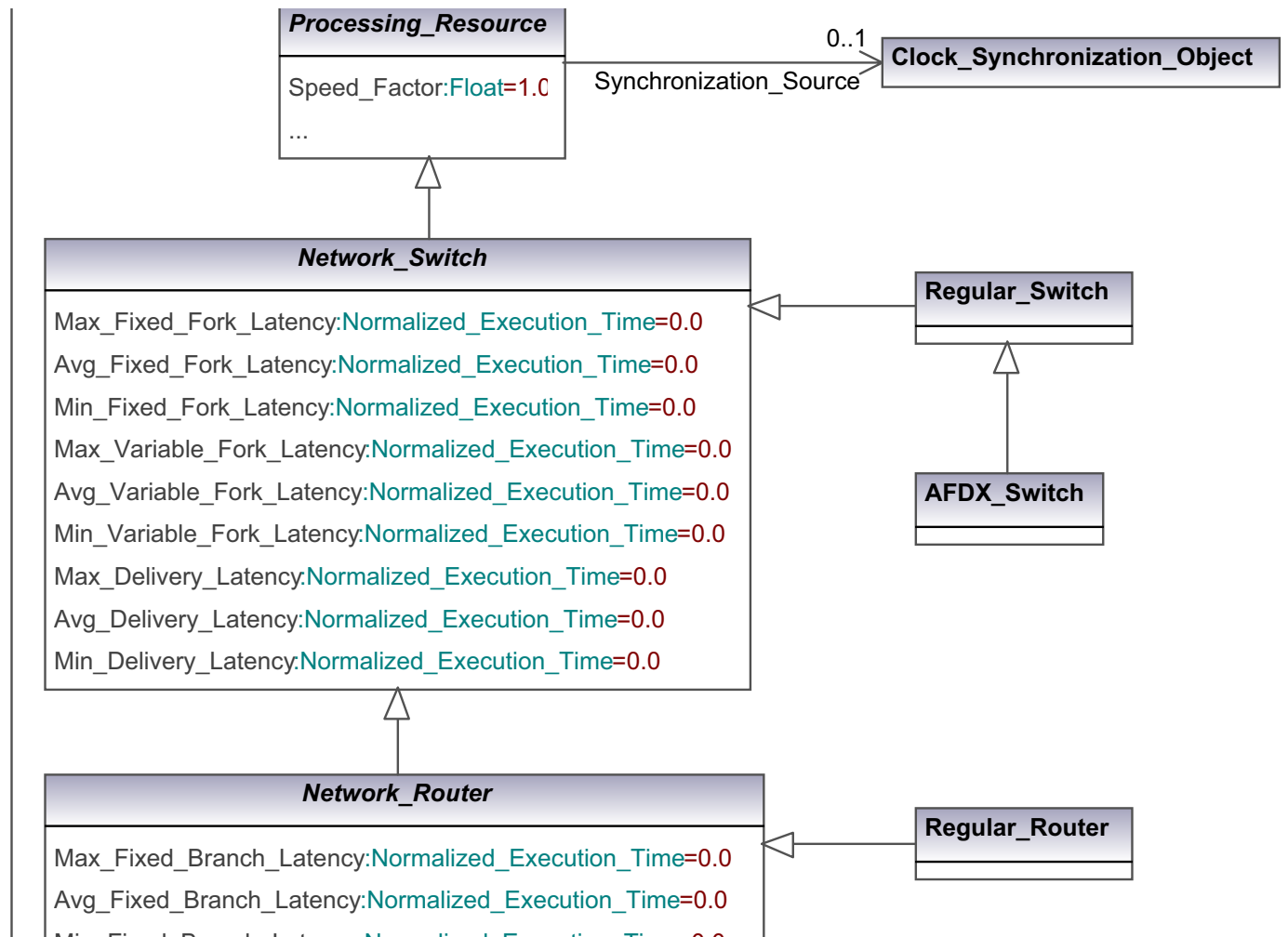


# New Computing Resources

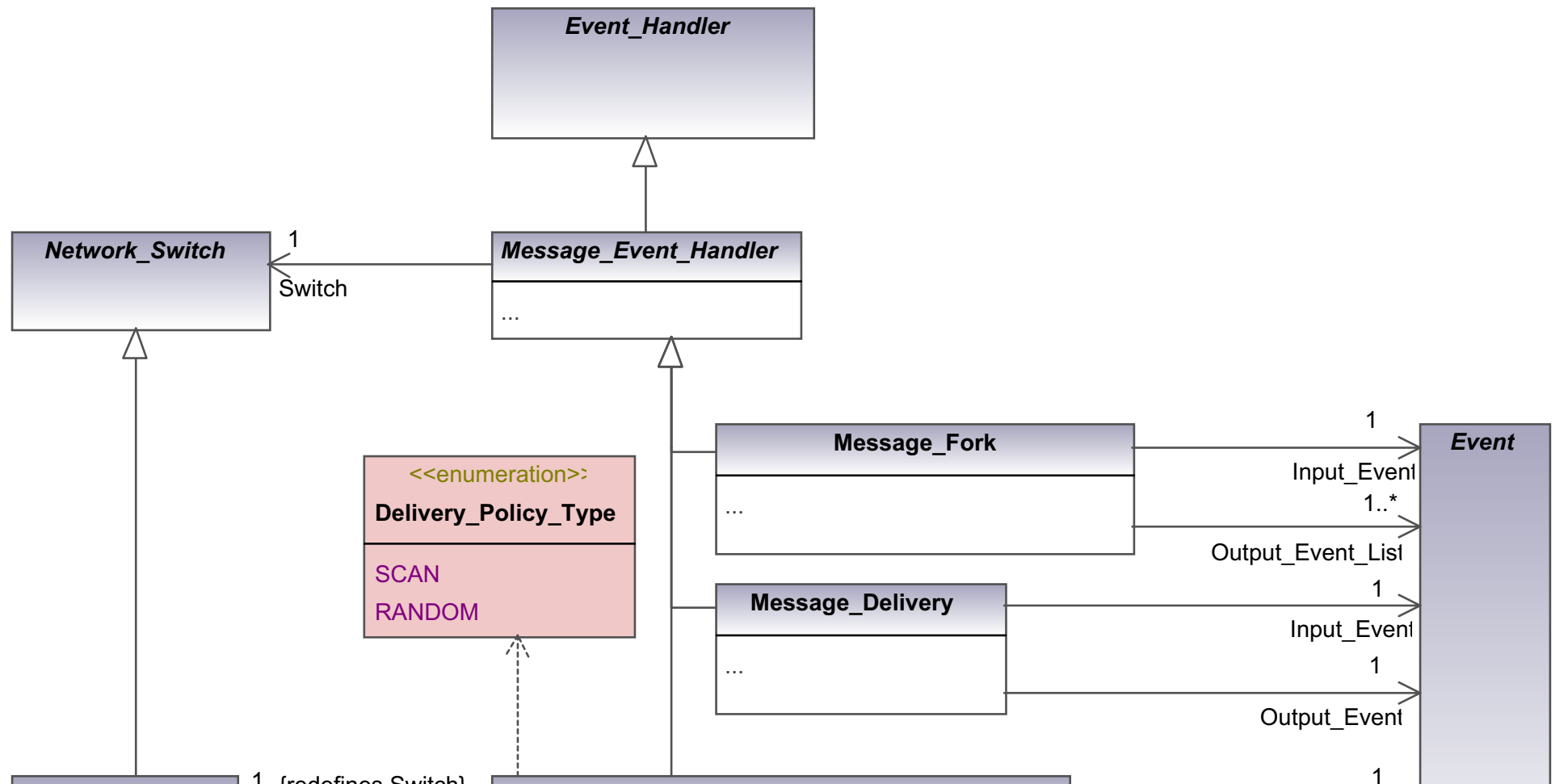
## Model\_Elements



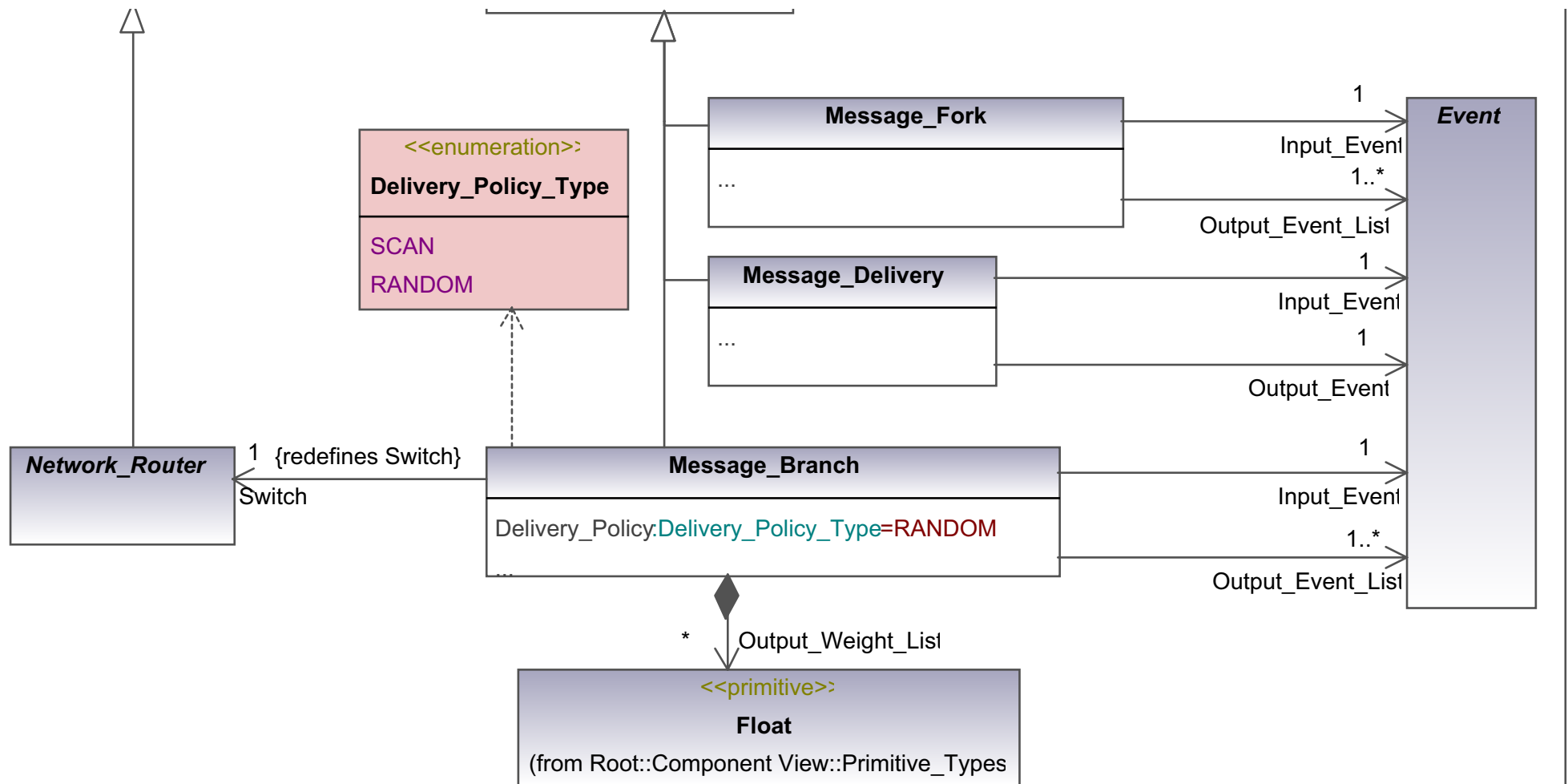
# Network switches and routers



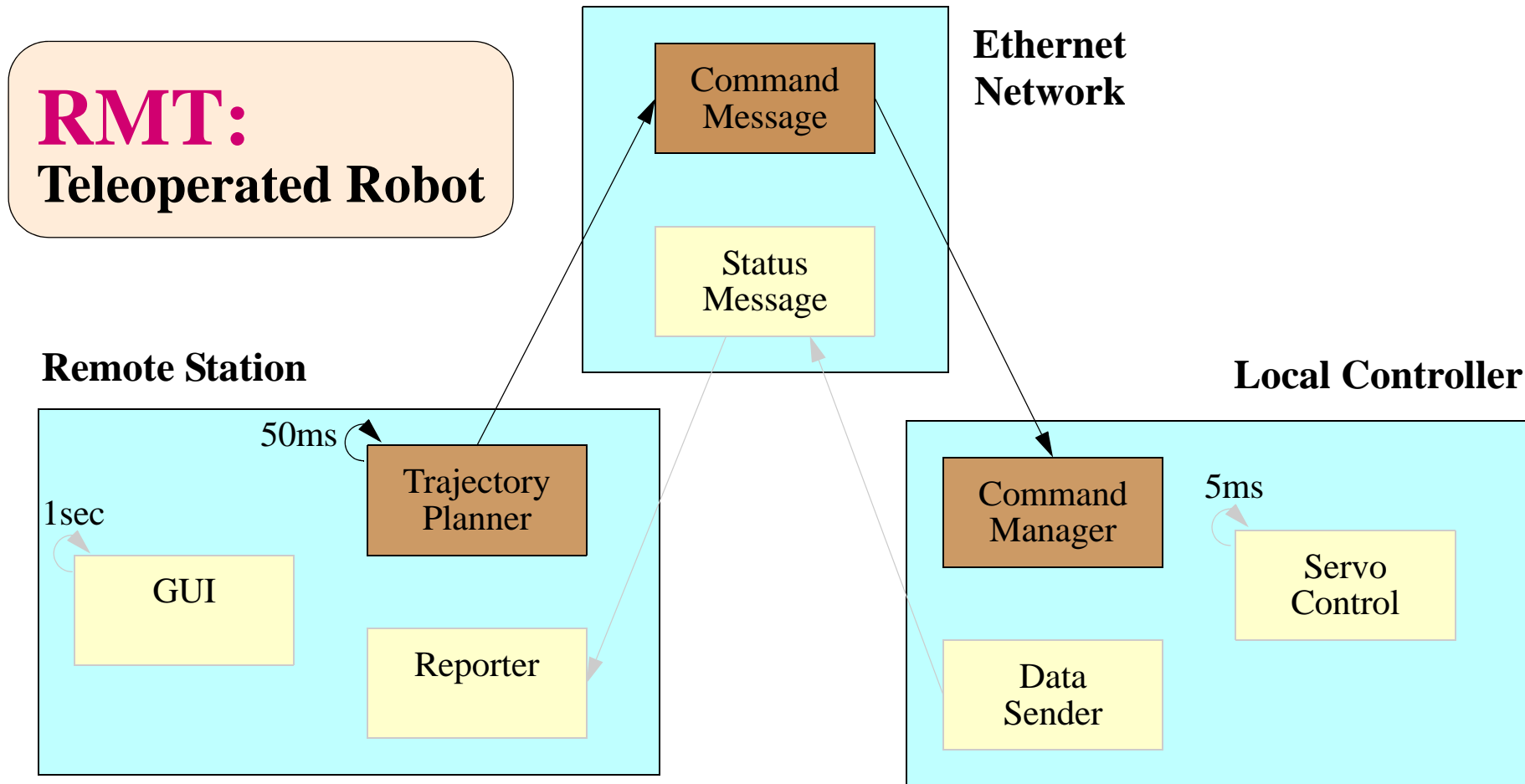
# Message Event Handlers for switches



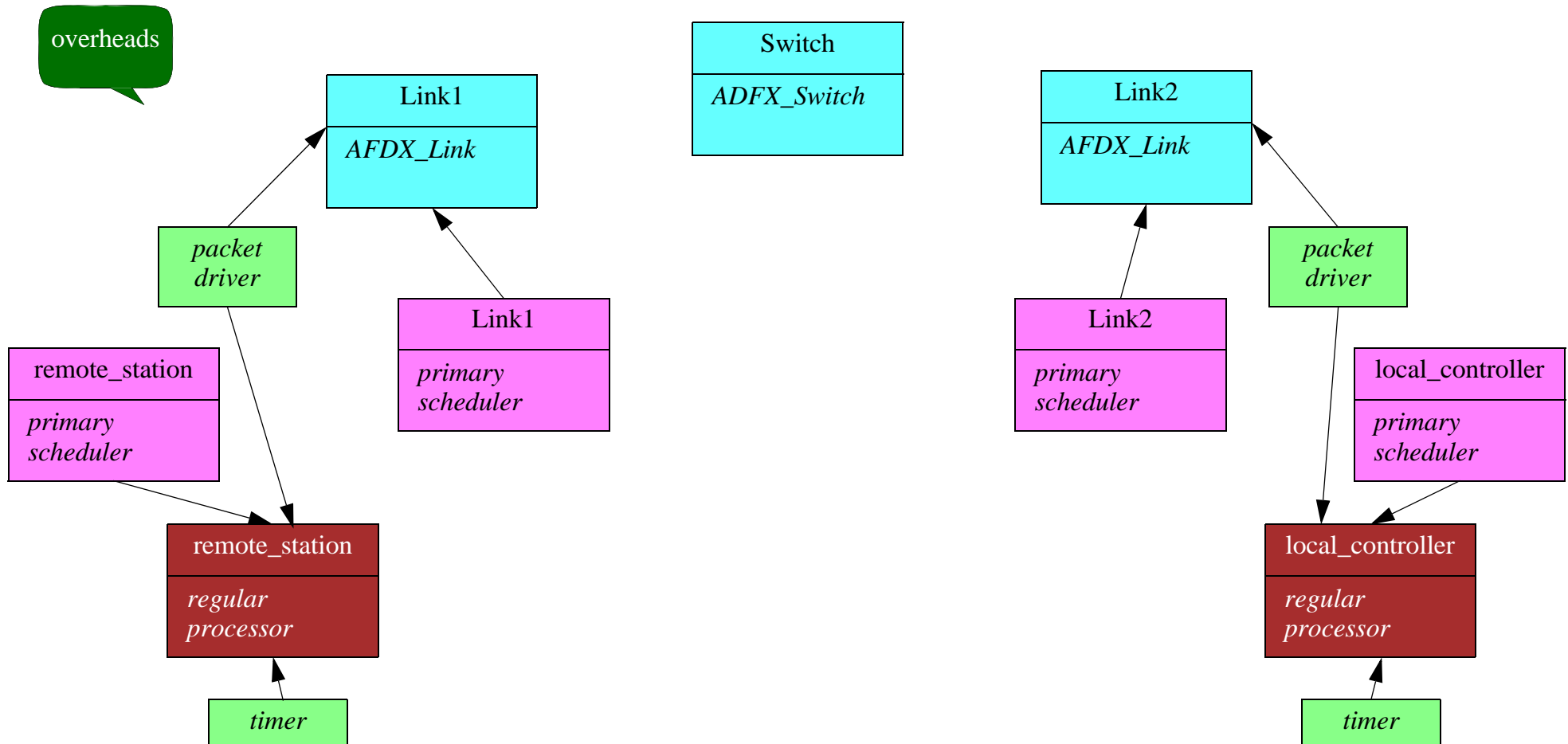
# Message Event Handlers for routers



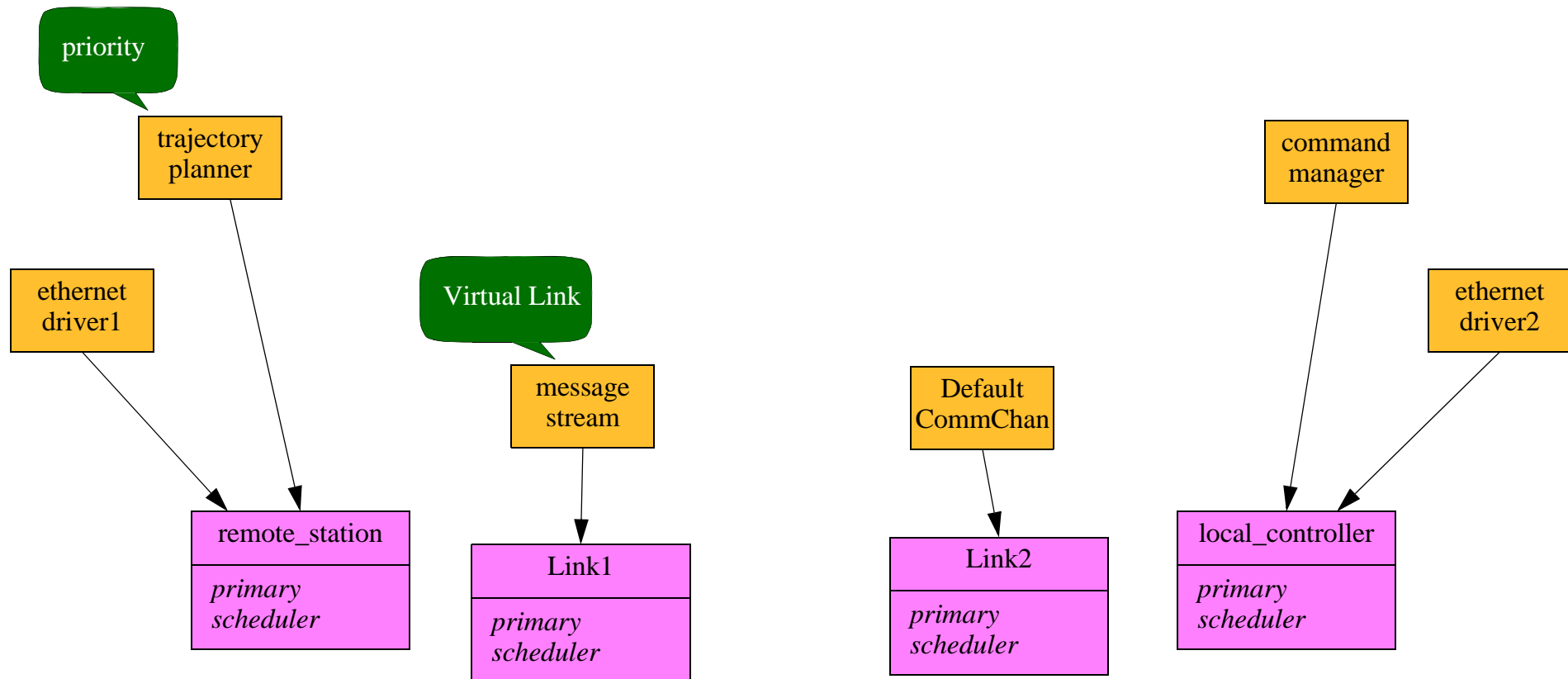
# Example of an end-to-end flow using a switch



# Processing resources, schedulers, drivers, and timers

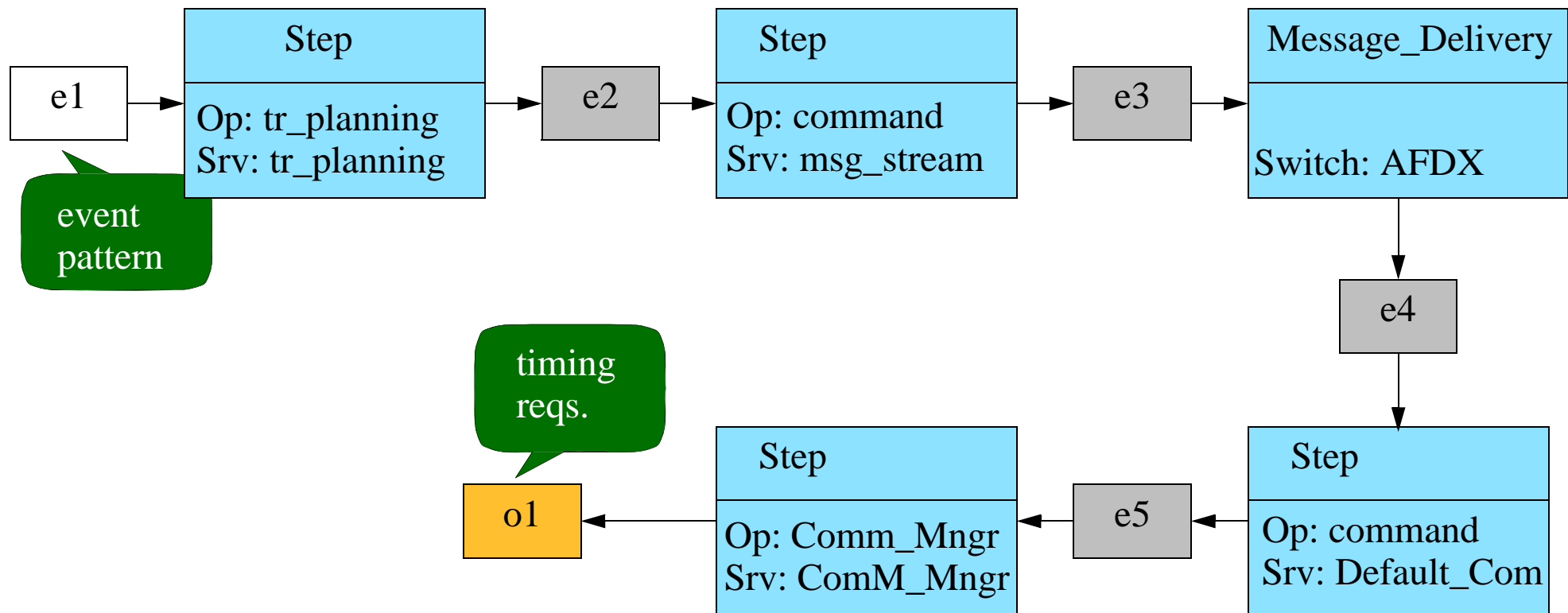


# Schedulable Resources: Threads and Communication Channels





# End-to-end flow: Command transaction in the example



# 4. Resource reservations

---

## Virtual resources as a new primary scheduler

### Double role with viewpoints:

- **when seen from the application viewpoint:**
  - a portion of a processing resource with guaranteed bandwidth and responsiveness
  - specified through resource reservation parameters
  - it is a contract between the application and the system, independent of the underlying implementation
- **when seen from the processing resource viewpoint**
  - a concrete workload that has to be scheduled
  - specified through scheduling parameters of the periodic server type (e.g., a sporadic server or a constant bandwidth server)

# Analysis of Resource Reservations

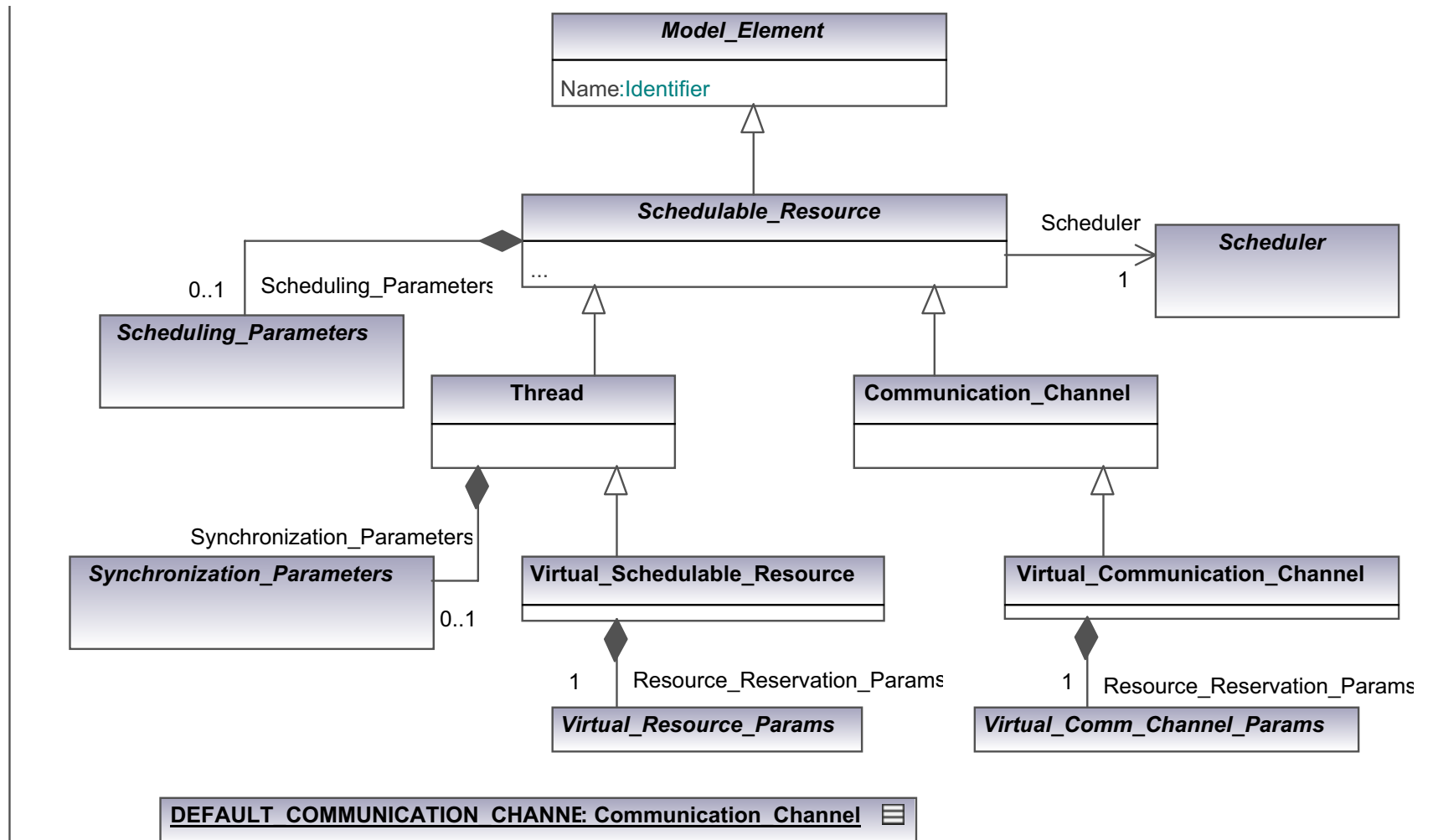
---

## Two kinds of analysis

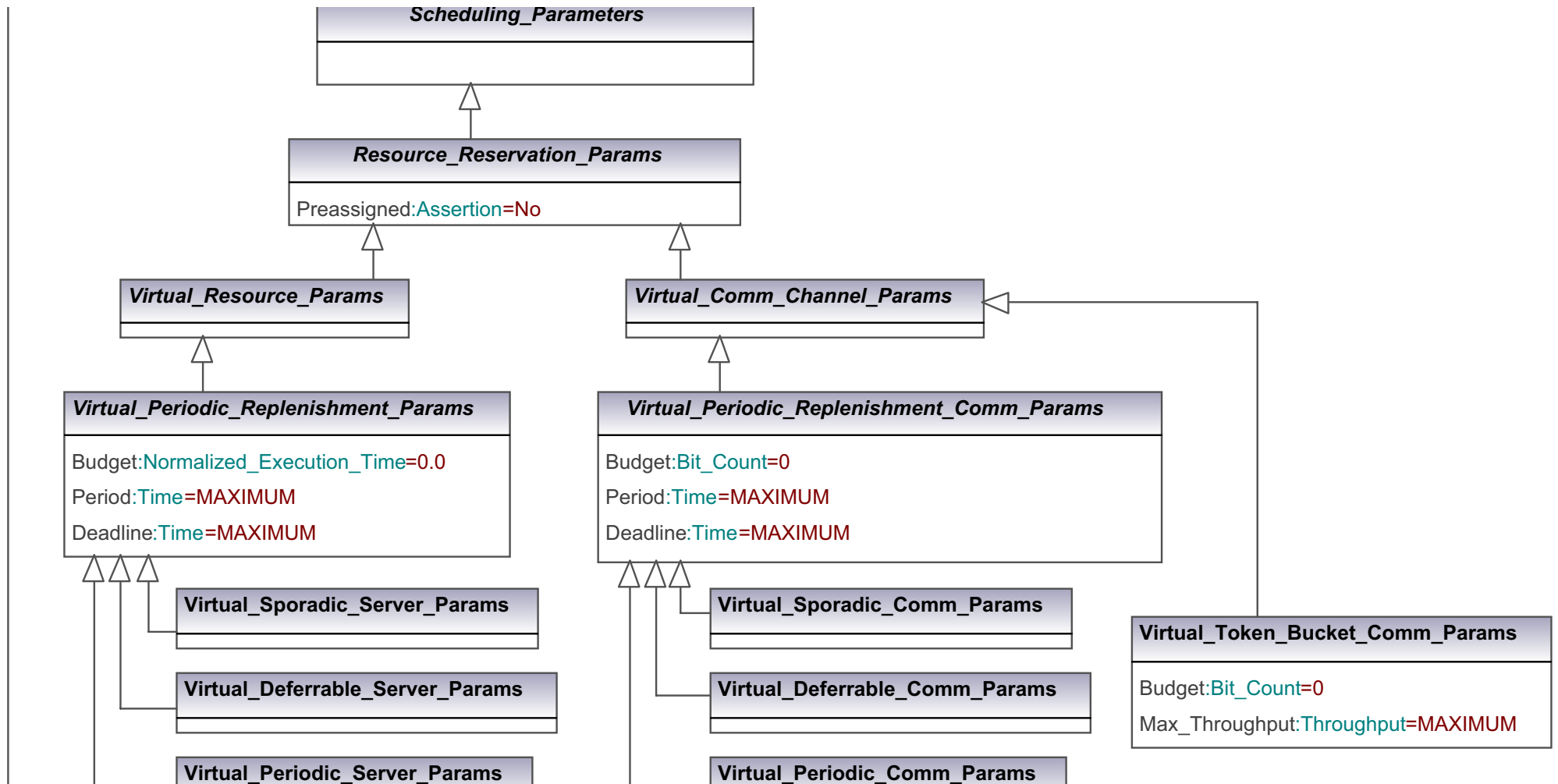
- **schedulability of an application running on top of a virtual resource**
  - **independent of the rest of the system**
- **schedulability of a set of virtual schedulable resources running in a particular physical platform**
  - **independent of the actual applications that will run on top of them**

**This separation of concerns helps in managing the complexity of the modelling and analysis processes**

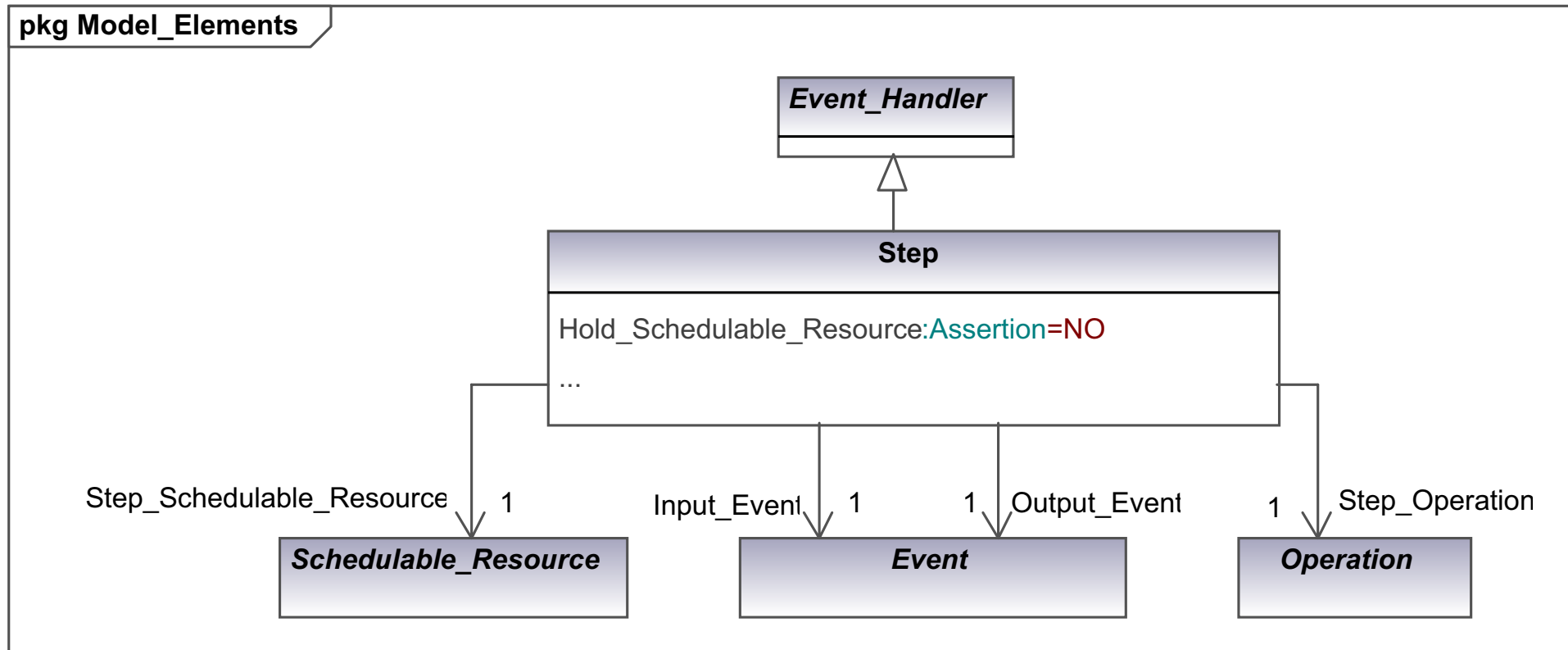
# Virtual Schedulable resources



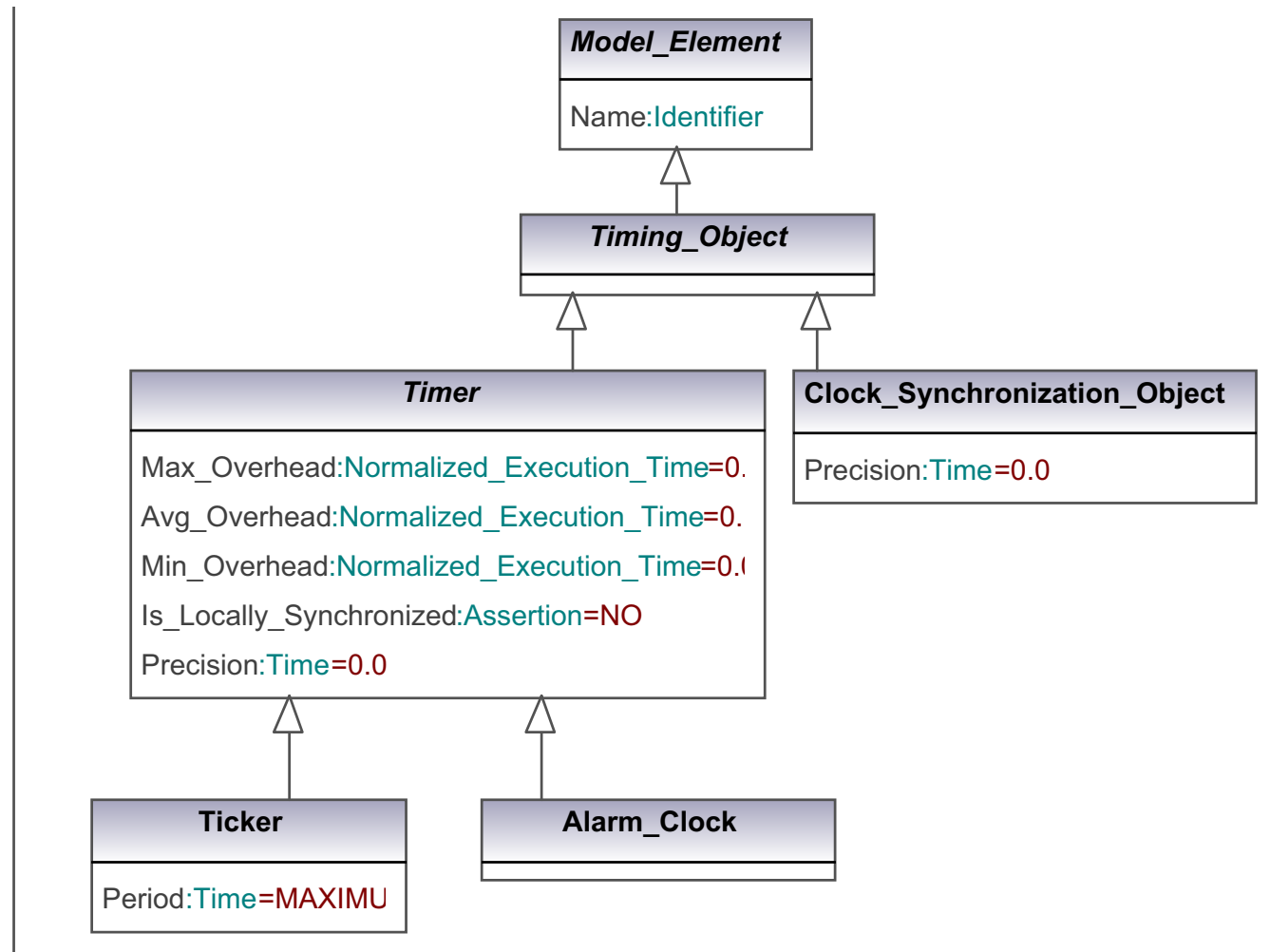
# Resource-Reservation Parameters



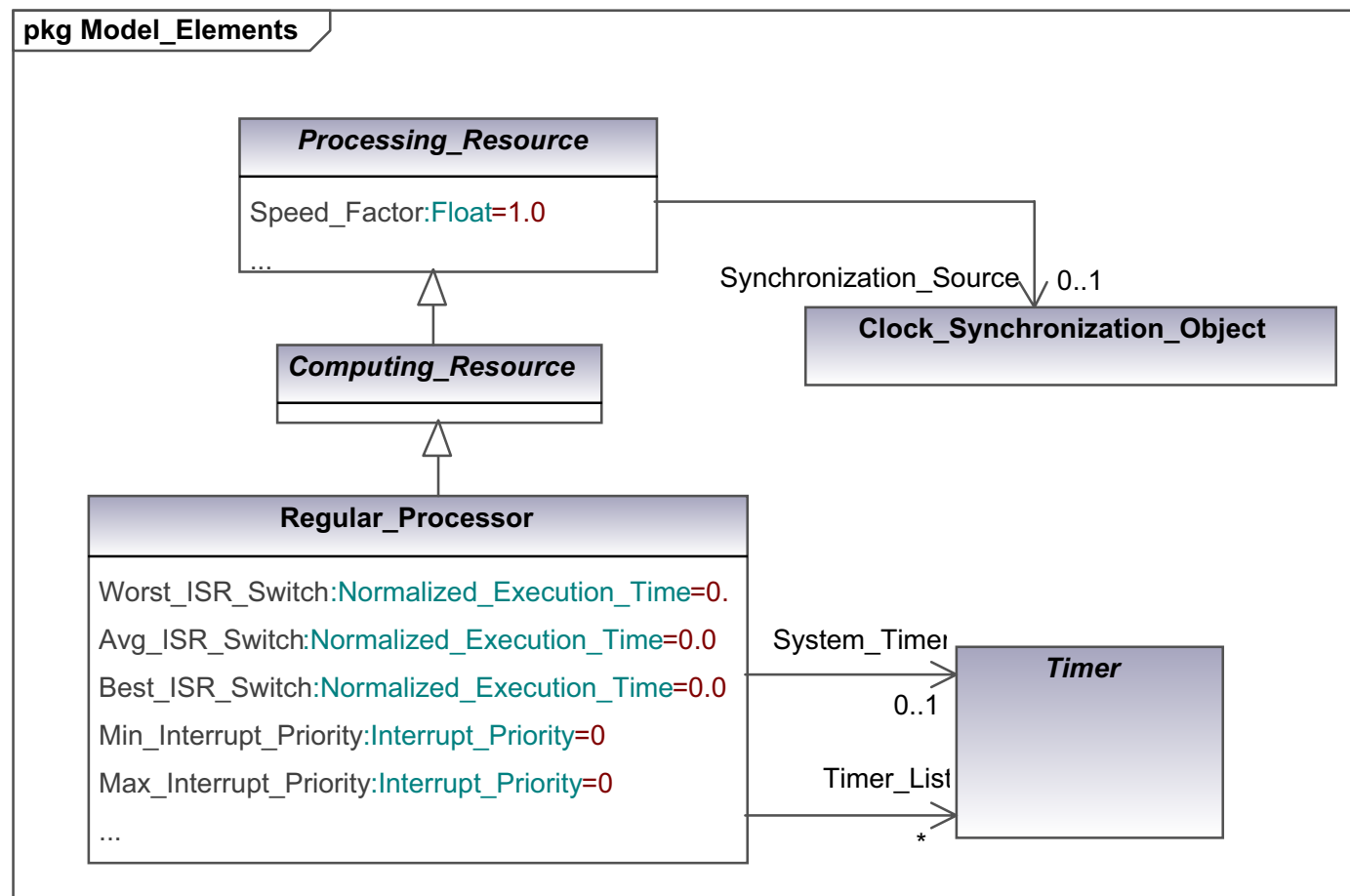
# 5. Support for thread locking from a transaction



# 6. Enhanced modelling of timers

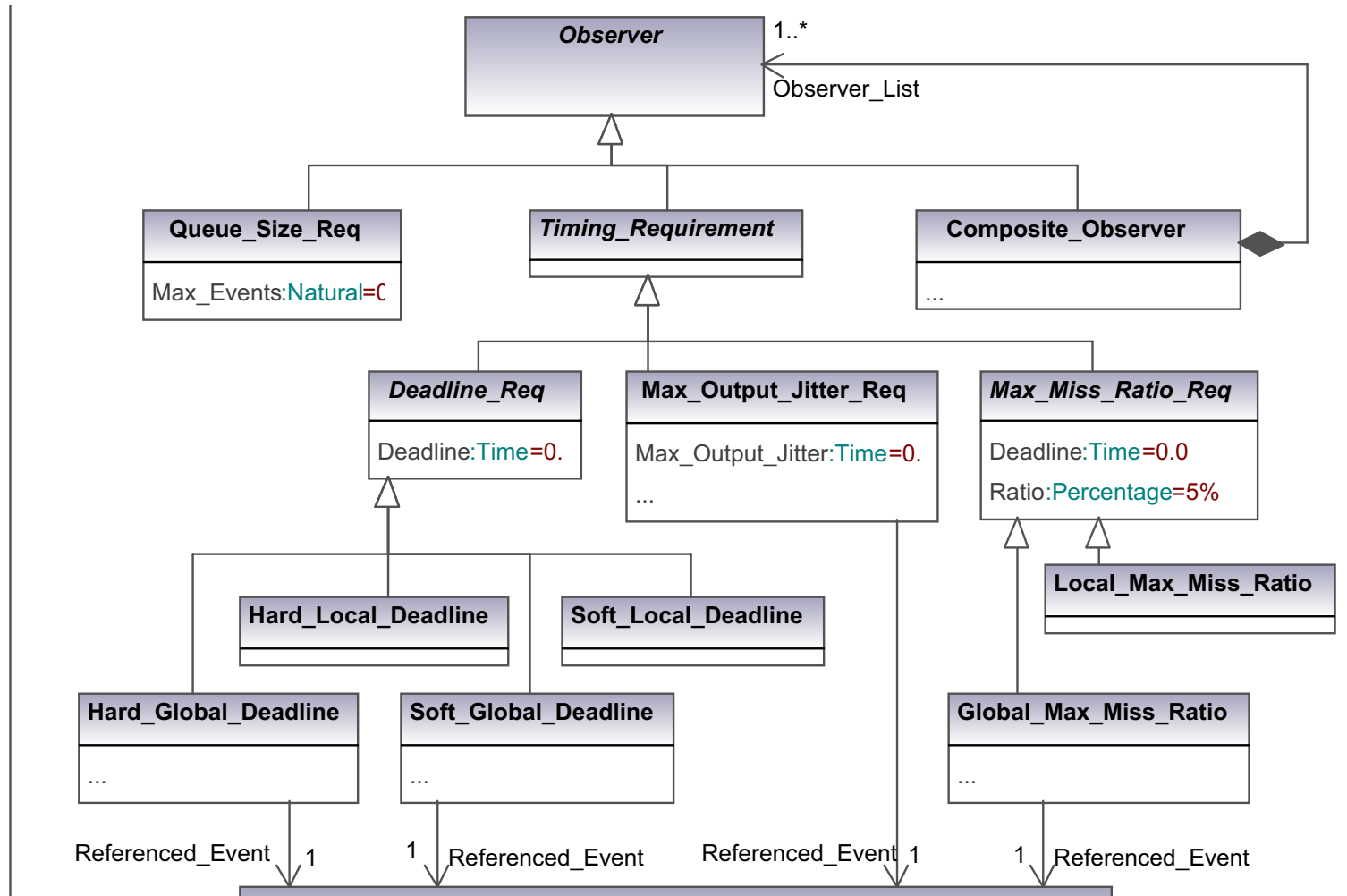


# Multiple timers





# 7. New Requirements (Queue Size Observers)



# Further (lower priority) extensions

---

## Interrupt scheduler

- Cannot model complex schemes, such as an interrupt controller (PIC):
  - configure preemption among levels
  - configure scheduling: priority, scan, ...

## Suspension inside protected operations (critical sections)

# Conclusions

---

**MAST is evolving and will soon cover aspects such as:**

- **alignment with MARTE**
- **partitioned scheduling**
- **network switches**
- **additional modelling capabilities**
- **... results from this workshop?**