

11.5 Mode Switch Procedures (supportsAsynchronousModeSwitch)

Switching of modes can be performed synchronously or asynchronously, dependent on the parameter `supportsAsynchronousModeSwitch` of the `ModeSwitchReceiverComSpec` (→7.9.2) of the Mode Users. If at least one of the Mode Users does not support the asynchronous mode switch then synchronous mode switching is to be used by the Mode Manager.

For **asynchronous mode switching** the Mode Manager does not implement any waiting conditions or feedback mechanisms. Modes are switched independent of the Mode Users. This may “overrun” a Mode User in case it needs more time to finish any previous mode while a new mode is already requested.

For **synchronous mode switching** the Mode Manager has to use acknowledgment feedback for the mode switch notification, e.g. via a `ModeSwitchedAckEvent` (→10.3.5). Other possibilities are

- to use a sequence of `Rte_Switch` and a blocking `Rte_SwitchAck` to send the mode switch notification and wait for the completion.
- to have a cyclic `RunnableEntity` to poll for the acknowledgement using `Rte_SwitchAck`.
- to use the `Rte_Mode` API to read the currently active mode from the RTE.

11.6 Queuing of Mode Switches (queueLength)

Mode switch notifications are usually configured as queued (attribute `queueLength` →7.9.2) and are handled similarly to queued Sender-Receiver communication (→7.5.2.2.10). The value must be greater than or equal to 1. Setting the value of `queueLength` to 1 implies that incoming requests are rejected while another request that arrived earlier is being processed (non-queued).

As opposed to queued Sender-Receiver communication, the queue for mode switch communication is configured for the sender side (the Mode Manager) in the `ModeSwitchSenderComSpec` (→7.9.2). The queue itself is implemented by the RTE.

In case of a full queue, any new mode switch notification is rejected. An `Rte_Switch` call is then returned with `RTE_E_LIMIT`. In case of a `queueLength` of 1 the RTE rejects any new mode switch notifications during the mode transition with `RTE_E_LIMIT`.

11.7 Distributed Mode Management (Inter ECU, Inter Partition)

Due to the fact that AUTOSAR does not support Inter-ECU ~~or Inter-Partition~~ communication of mode switch notifications each ECU ~~(or Partition)~~ has to have its own Mode Manager(s). As the Mode Request is a normal Sender / Receiver Communication which can be system-wide it can be distributed to all the Mode Managers as shown in the figure below.

In case the Mode Manager is part of the Basic Software the Mode Requester cannot directly communicate with the Mode Manager(s). Instead a `ServiceProxySwComponentType` is used →6.3.

However, the Mode Users of a `ModeDeclarationGroupPrototype` can be distributed on several partitions or different cores.

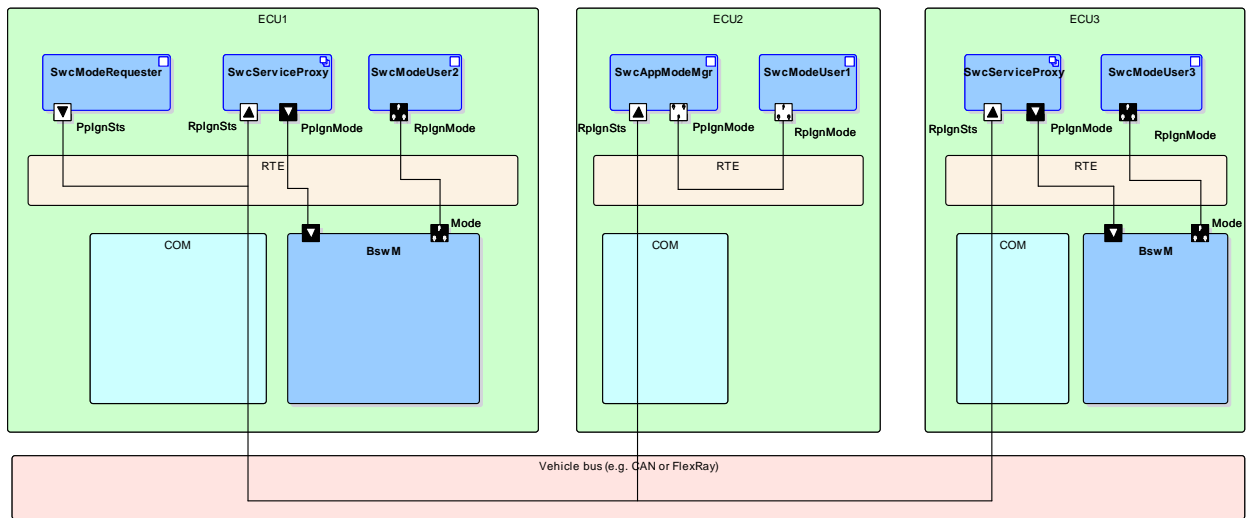


Figure 11-1 Distributed Mode Management

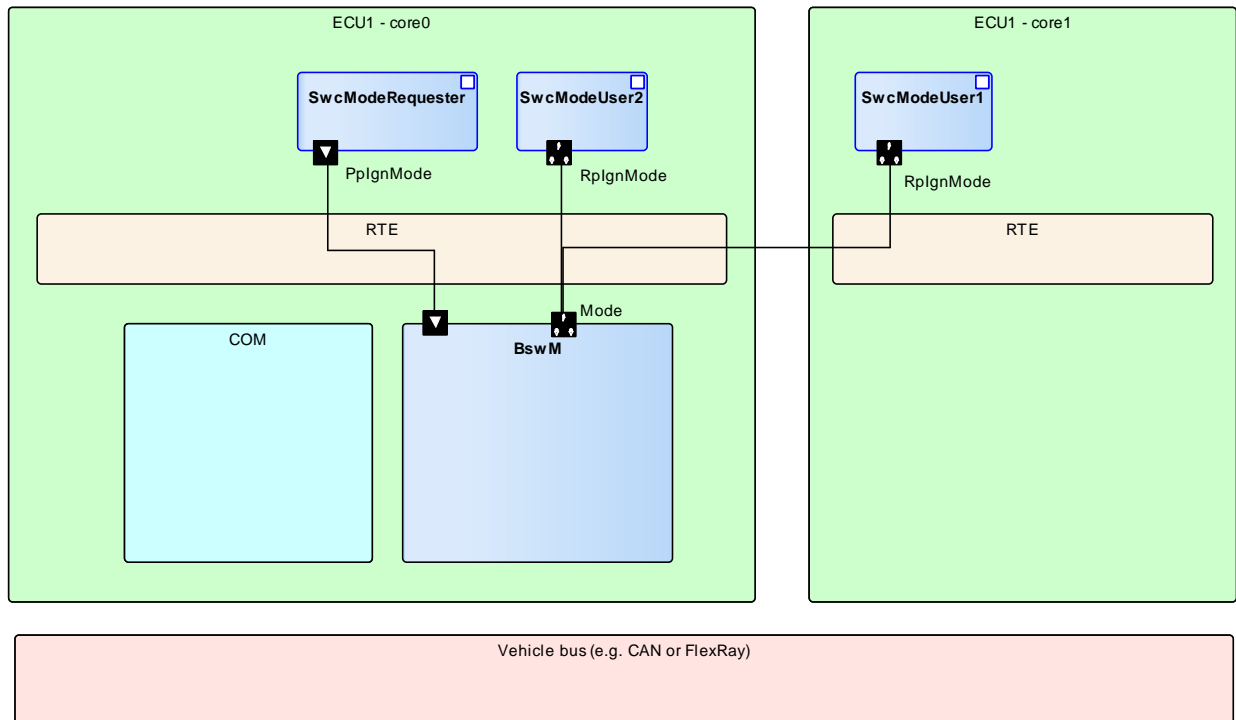


Figure 11-2 Distributed Mode Management over core (or partition) boundaries