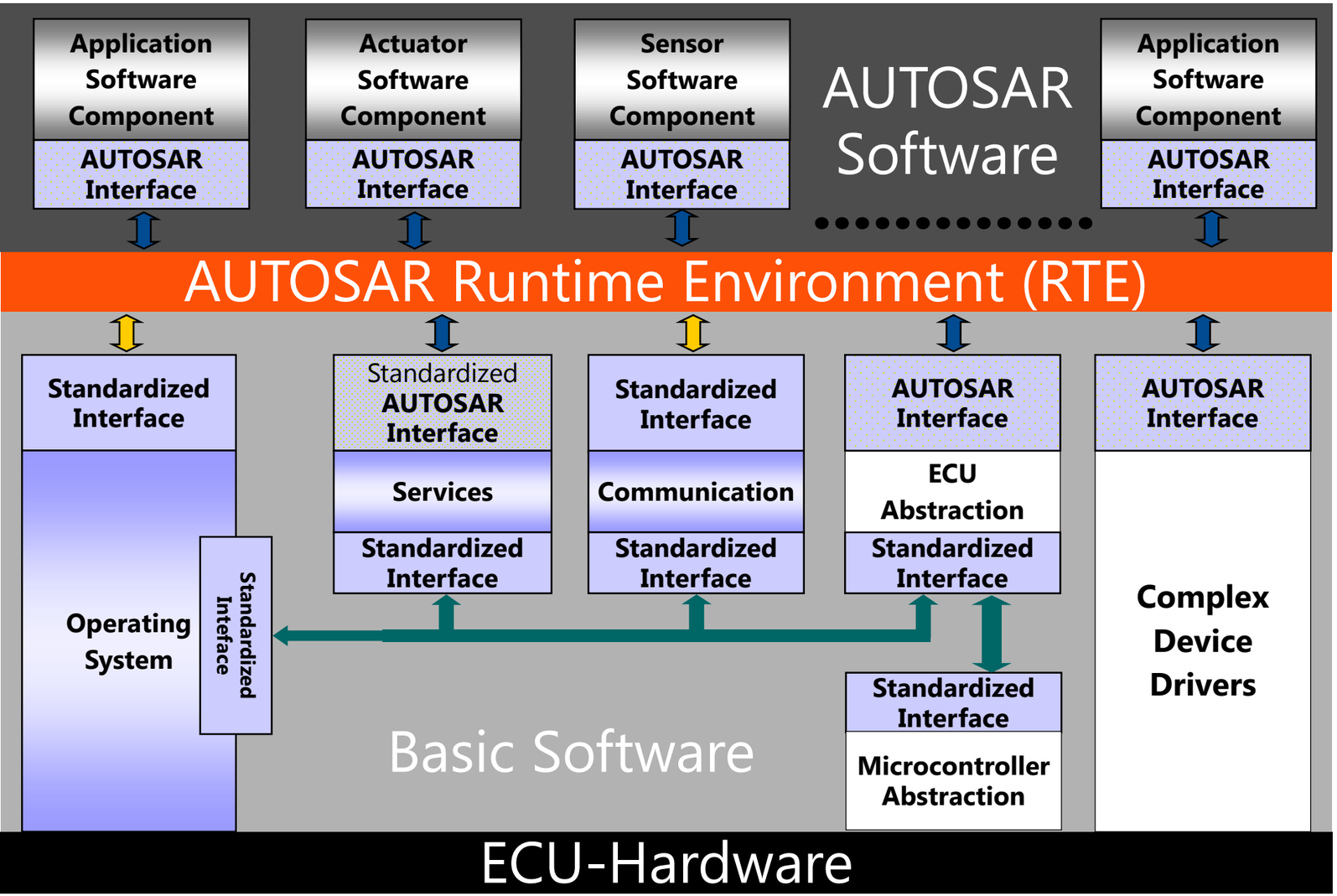
AUTOSAR SWC & RTE

KPIT

Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

Introduction to RTE



- RTE is responsible to establish link between AUTOSAR software components
- Software components communicate via this link provided by RTE
- RTE acts as means by which AUTOSAR software components access basic software modules including the OS and Communication service
- RTE implements the VFB for each ECU
- RTE abstracts the software component layer from implementation of the Basic software and from the hardware
- RTE enables software components to be reused in different ECUs

Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

Interfaces in AUTOSAR

AUTOSAR Interface	An "AUTOSAR Interface" defines the information exchanged between software components and/or BSW modules. This description is independent of a specific programming language, ECU or network technology. AUTOSAR Interfaces are used in defining the ports of software-components and/or BSW modules. Through these ports software-components and/or BSW modules can communicate with each other (send or receive information or invoke services). AUTOSAR makes it possible to implement this communication between Software- Components and/or BSW modules either locally or via a network.
Standardized AUTOSAR Interface	A "Standardized AUTOSAR Interface" is an "AUTOSAR Interface" whose syntax and semantics are standardized in AUTOSAR. The "Standardized AUTOSAR Interfaces" are typically used to define AUTOSAR Services, which are standardized services provided by the AUTOSAR Basic Software to the application Software-Components.
Standardized Interface	A "Standardized Interface" is an API which is standardized within AUTOSAR without using the "AUTOSAR Interface" technique. These "Standardized Interfaces" are typically defined for a specific programming language (like "C"). Because of this, "standardized interfaces" are typically used between software-modules which are always on the same ECU. When software modules communicate through a "standardized interface", it is NOT possible any more to route the communication between the software-modules through a network.

Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

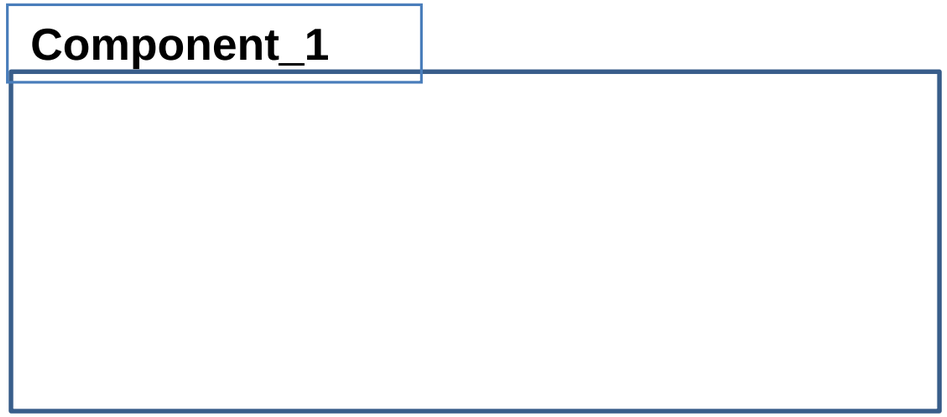
RTE Entities

- Software Component
- Composition
- Port
- Port Interface
- Elements
- Connector
- Internal Behavior
- Runnable Entity

RTE Entities

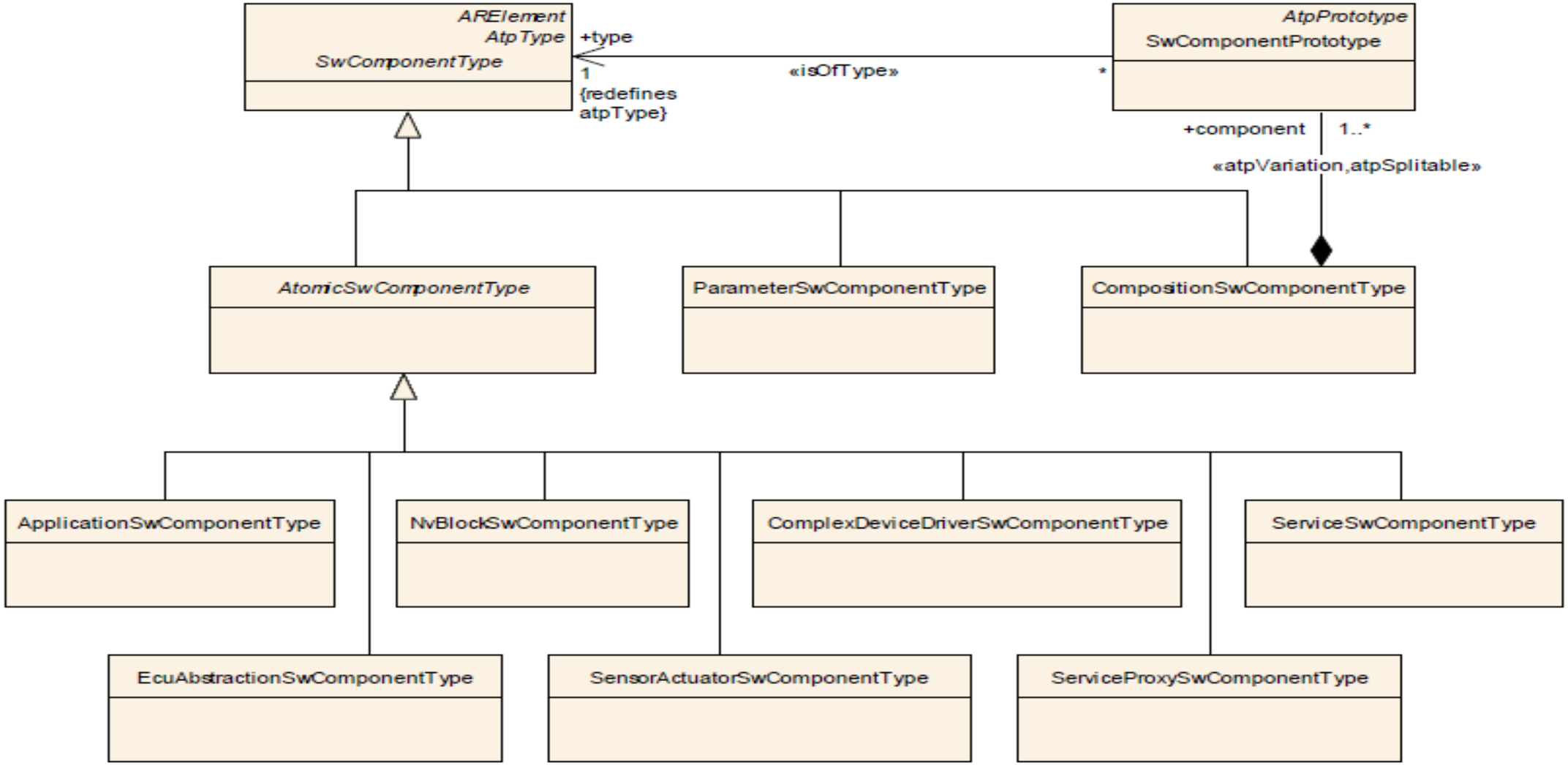
Software Component

- Application Software Component
- Sensor-Actuator Component
- Calibration Parameter Component
- Service Component
- ECU Abstraction Component
- Complex Device Driver Component
- Composition



RTE Entities

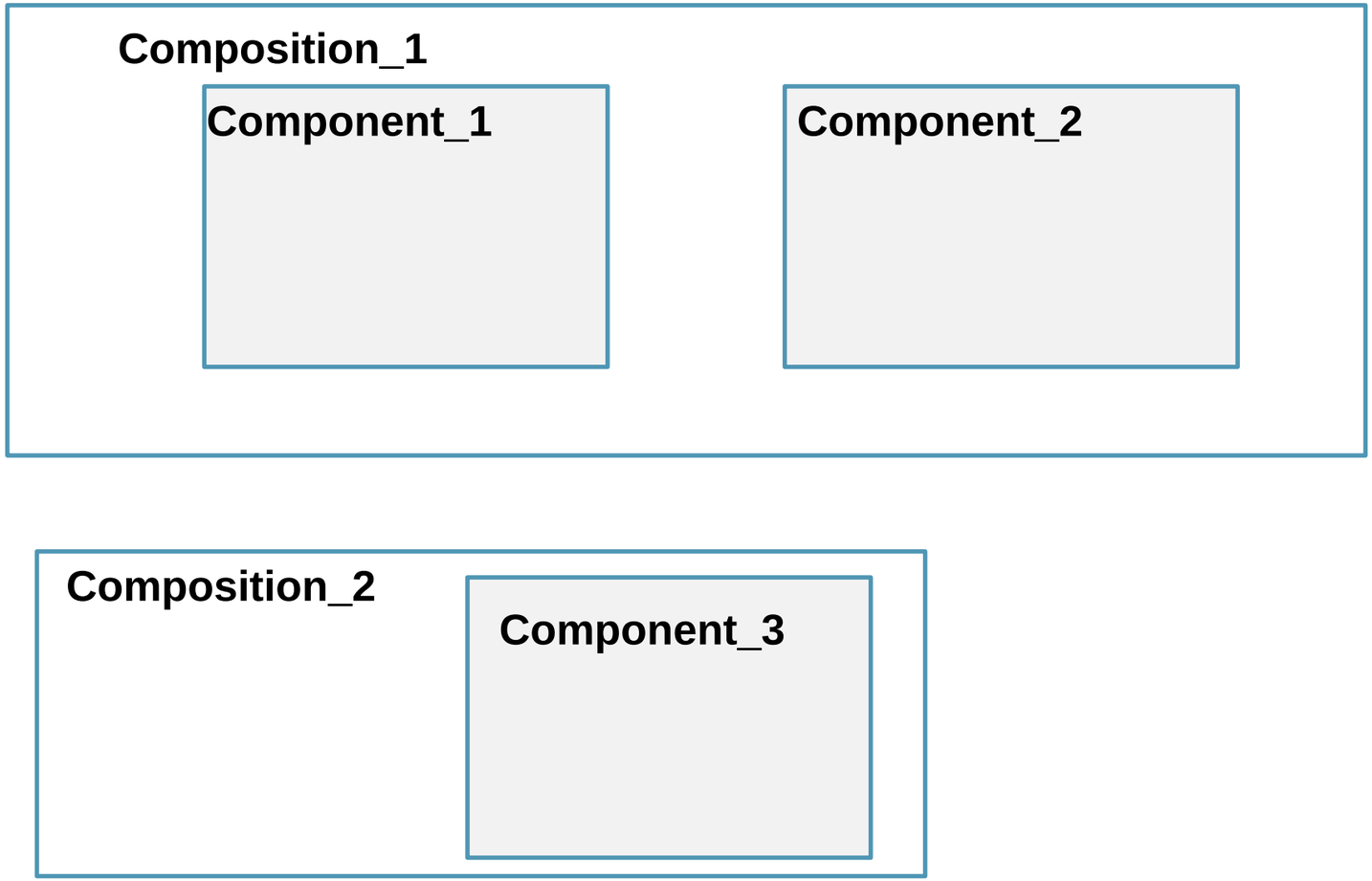
Different types of Software Component



RTE Entities

Composition

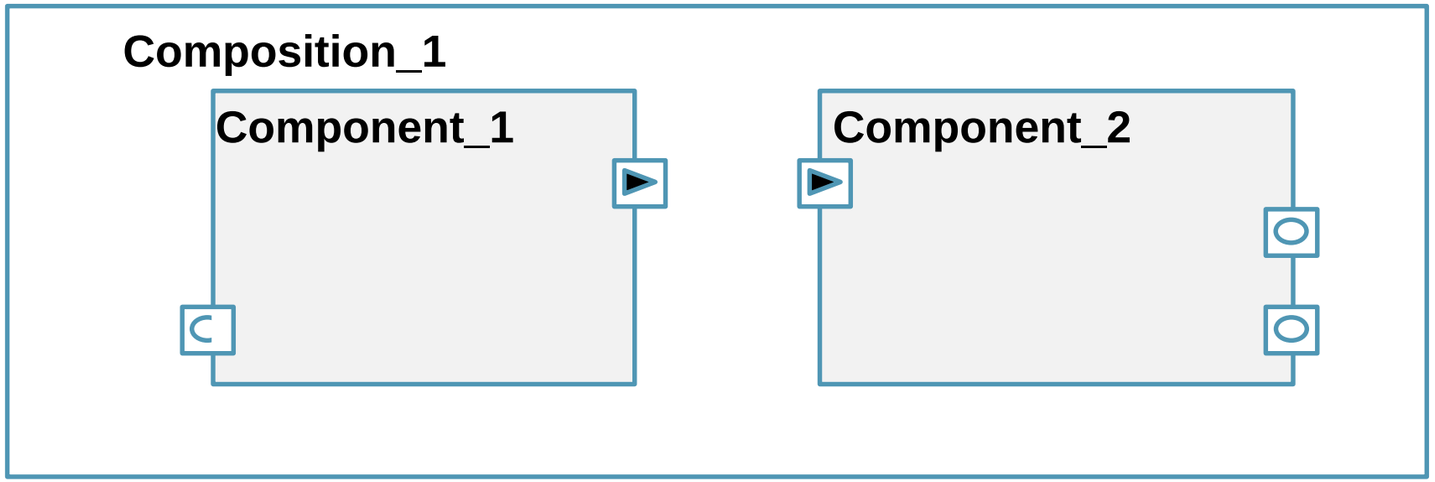
Used as structuring elements to build up a hierarchical system.

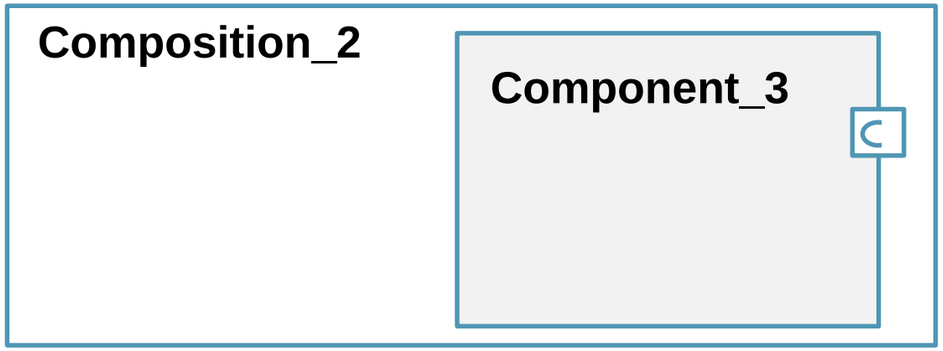


RTE Entities

Port

- Provide Port (PPort)
- Receive Port (RPort)

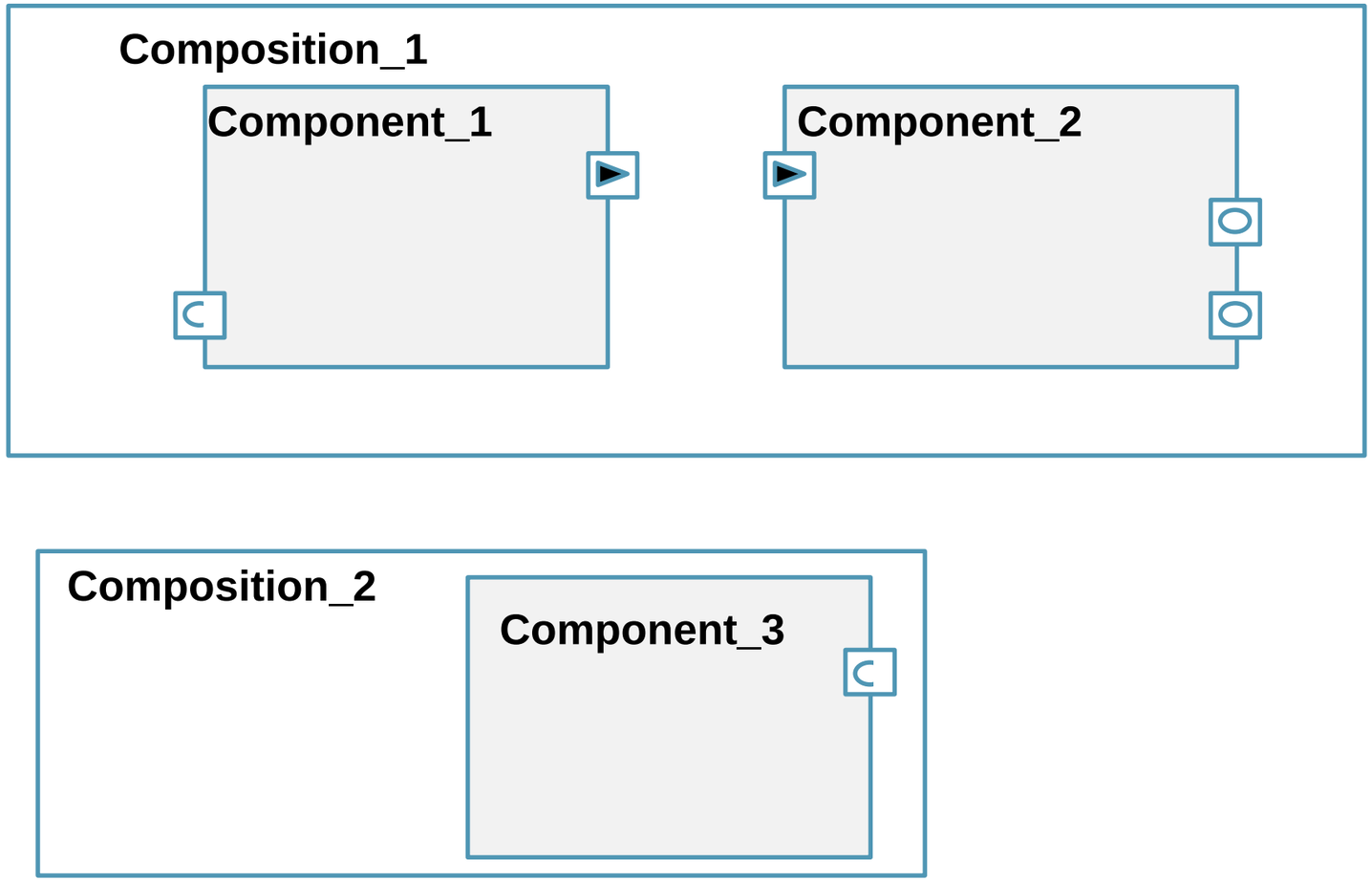




RTE Entities

Interface

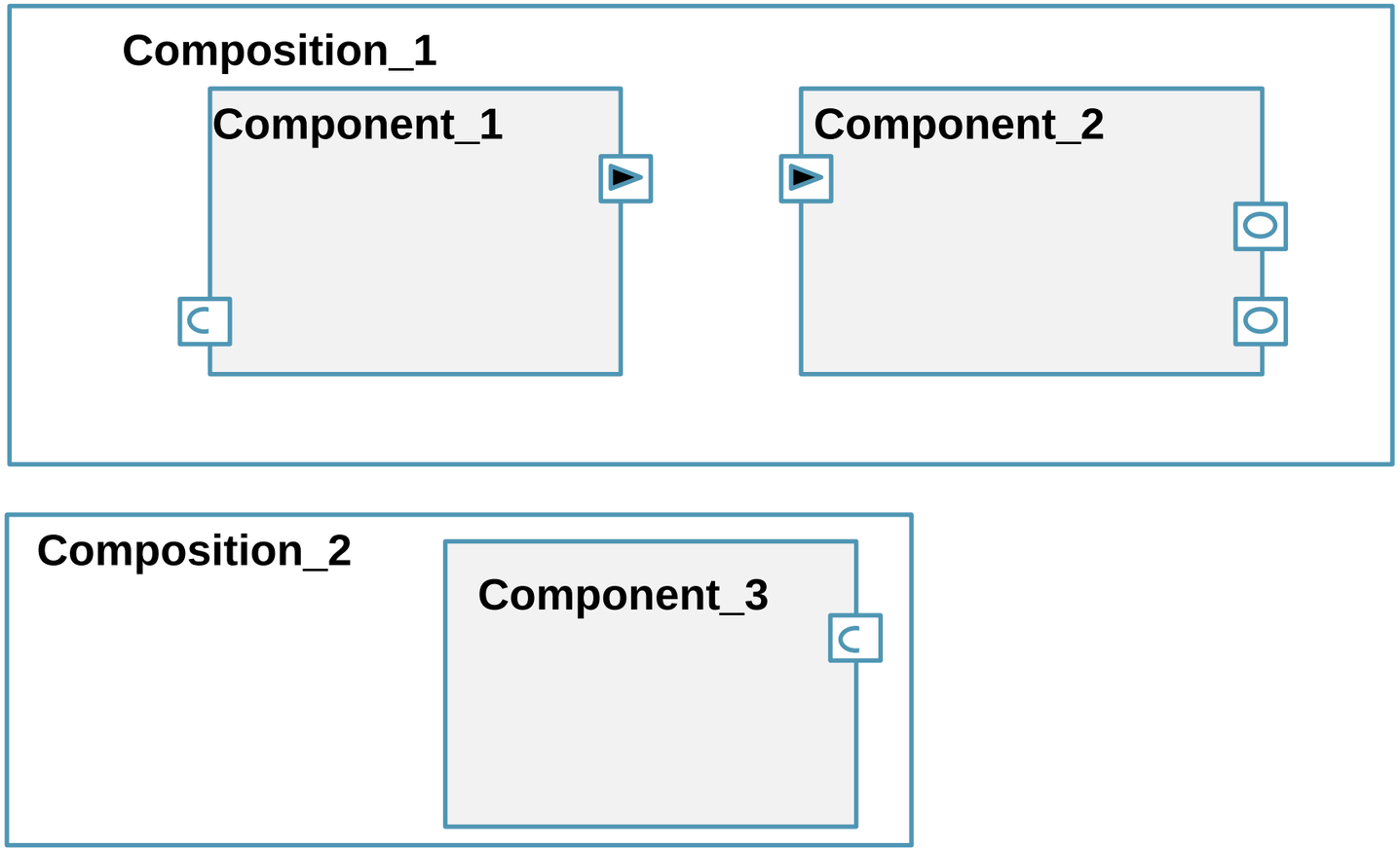
- Sender Receiver Interface
- NV Data Interface
- Mode Switch Interface
- Client Server Interface
- Parameter Interface
- Trigger Interface



RTE Entities

Elements

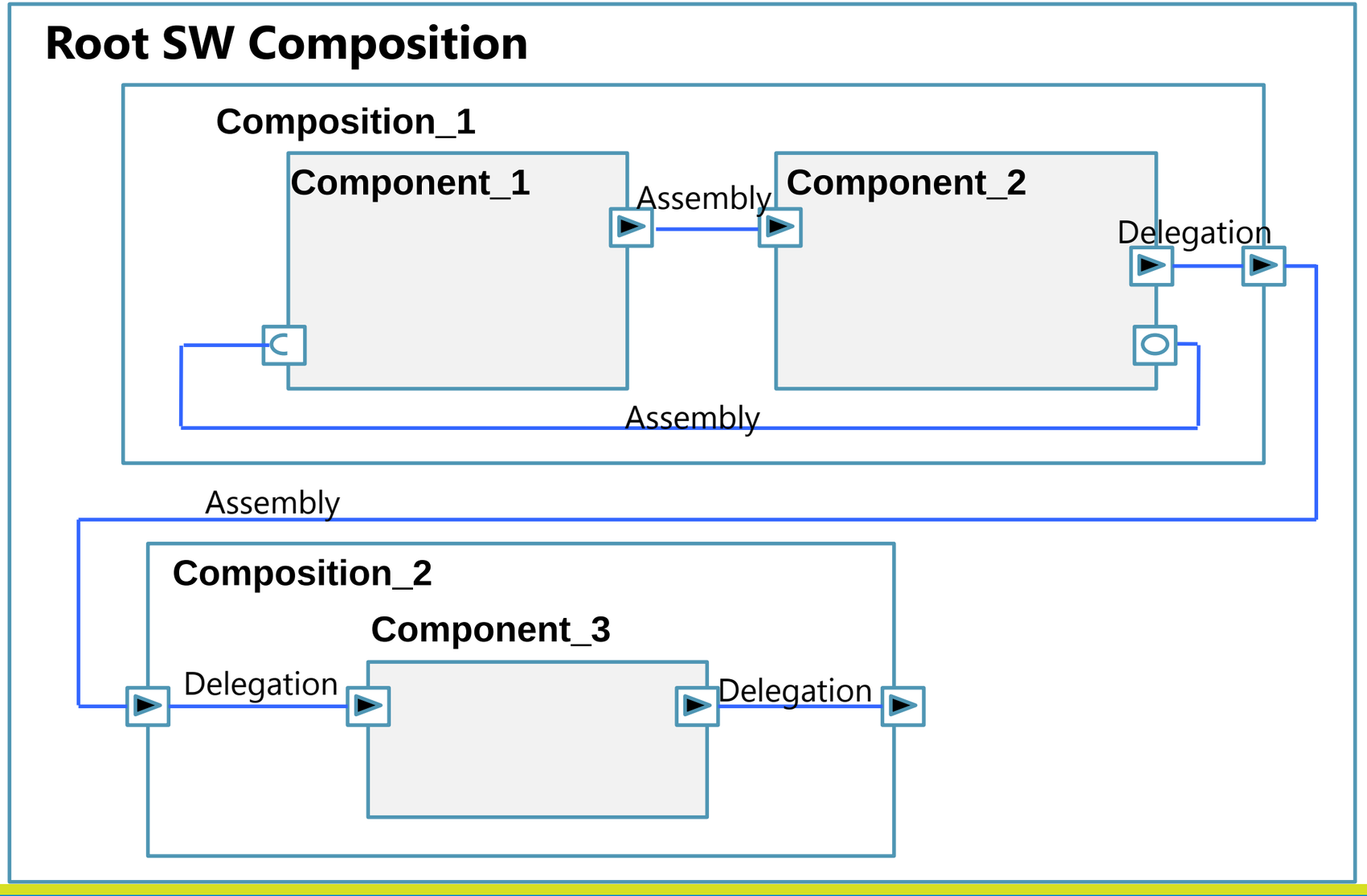
- Sender Receiver – **Data Element - Variable Data Prototype**
- Sender Receiver (Mode Communication) – **Mode Declaration Group**
- Client Server – **Operation Element - Argument Data Prototype**
- Calibration – **CalPrm Element**



RTE Entities

Connector

- Assembly
- Delegation

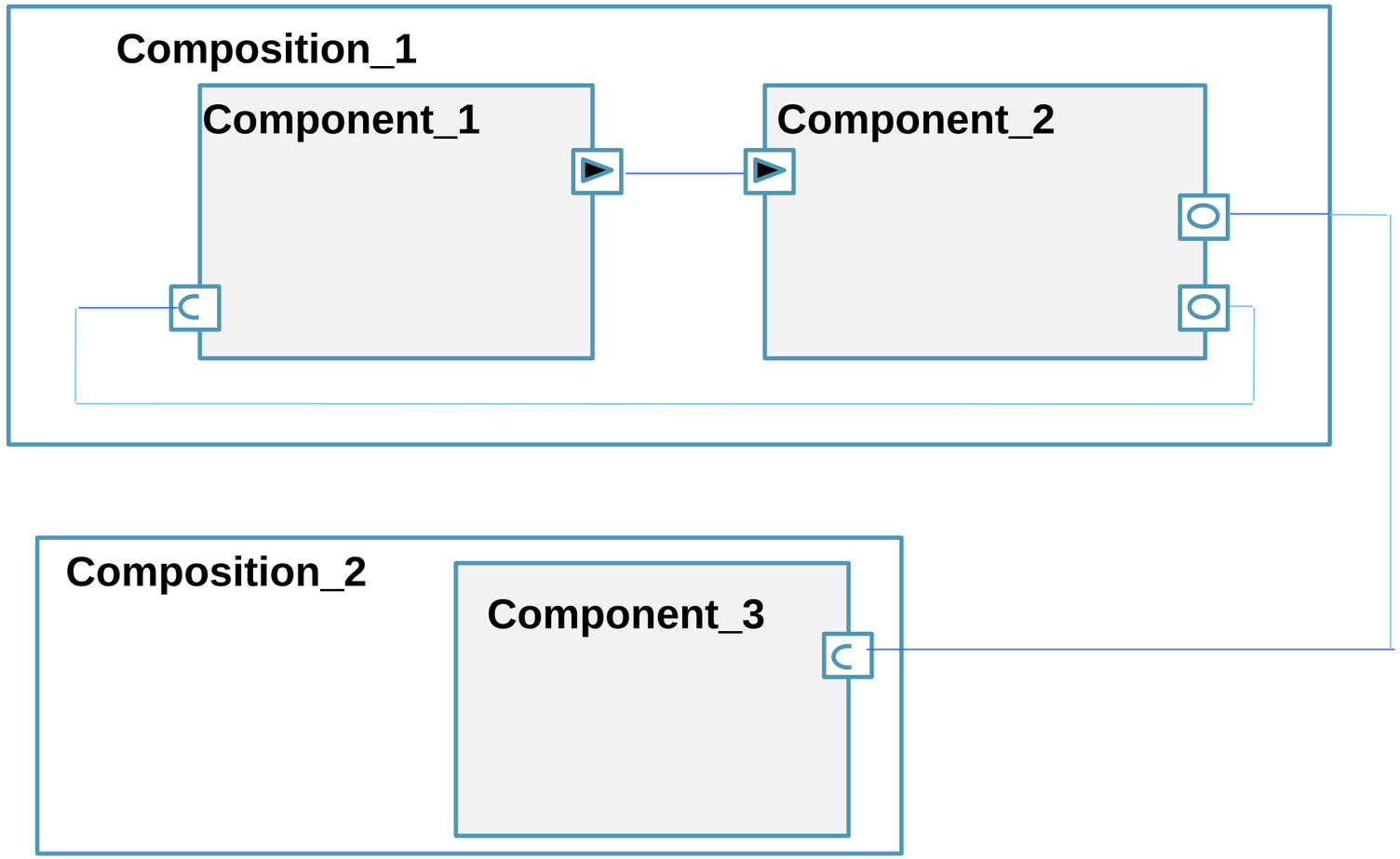


RTE Entities

Internal Behaviour

Internal structure of the Software Component and consists of:

- Runnable Entity
- RTE Events
- Exclusive Areas



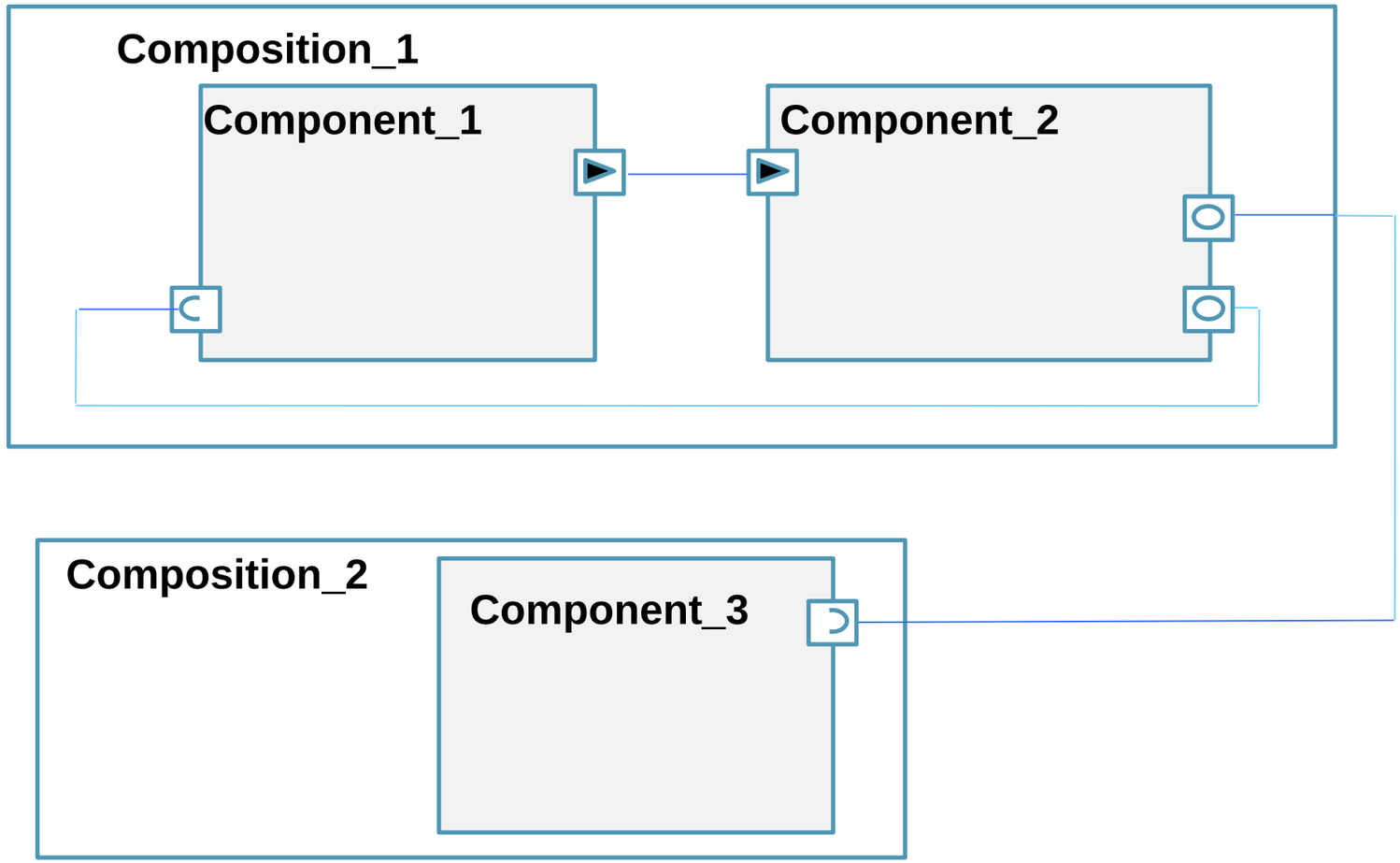
RTE Entities

Runnable Entity

Are schedulable unit of a Software Component

Are ‘C’ functions within the Software Component.

Are triggered by RTE Events



RTE Entities

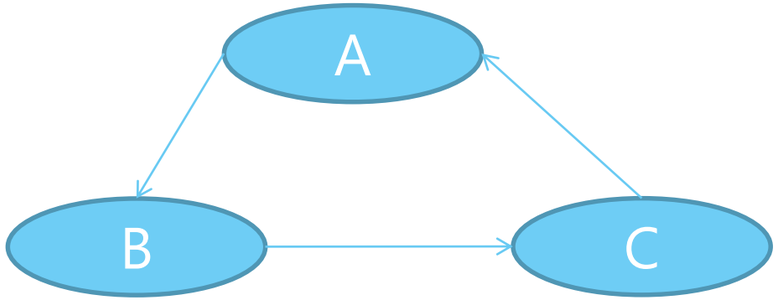
Sender Receiver Interface

- Multiple Data Elements
- Multiple Receivers and Senders (1:1,1:n,n:1)
- Implicit and Explicit Data Reception and Transmission
- Transmission Acknowledgement
- Communication Time-out
- Data Element Invalidation
- Never received status
- Update flag

RTE Entities

Mode Switch Interface

- Mode Group Management
- Initialization
- Event on Entry, on Exit and on Transition
- Mode Switch Acknowledgement



RTE Entities

Client Server Interface

- Synchronous
- Asynchronous

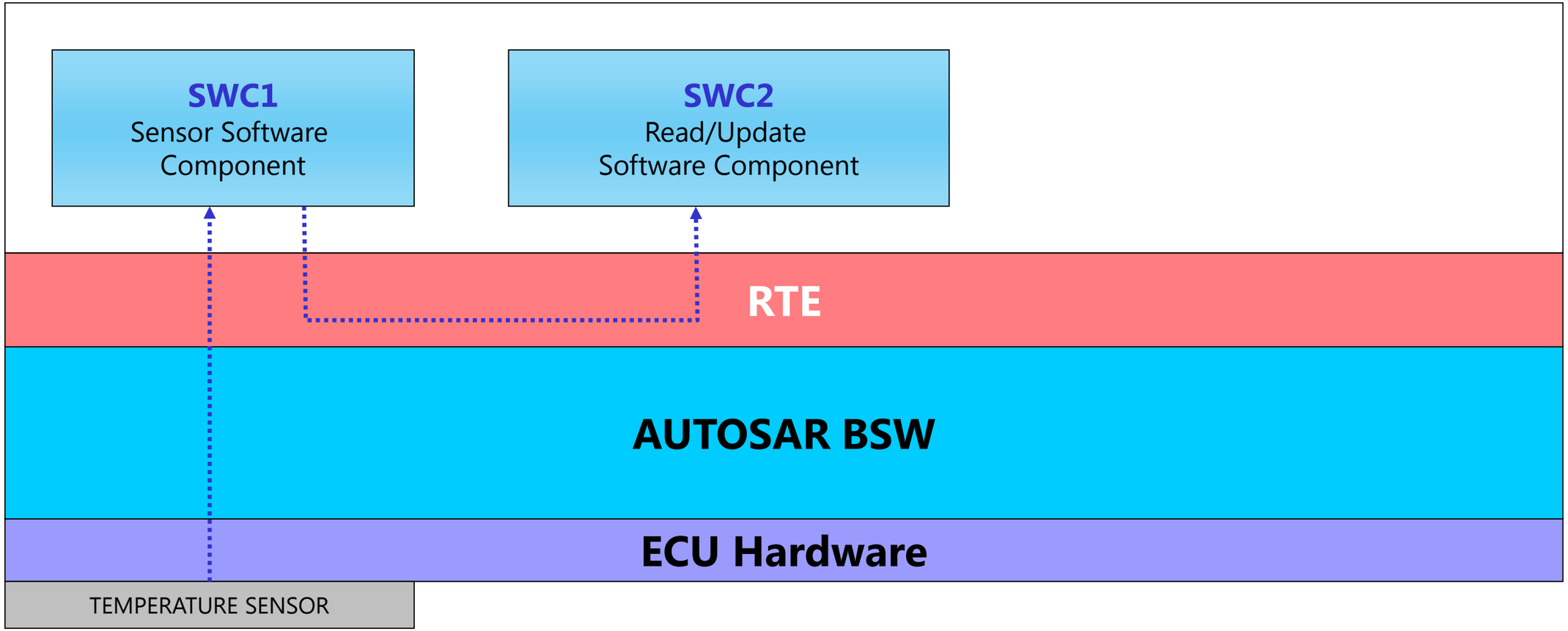
Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

Intra ECU Communication

Intra ECU Communication via RTE

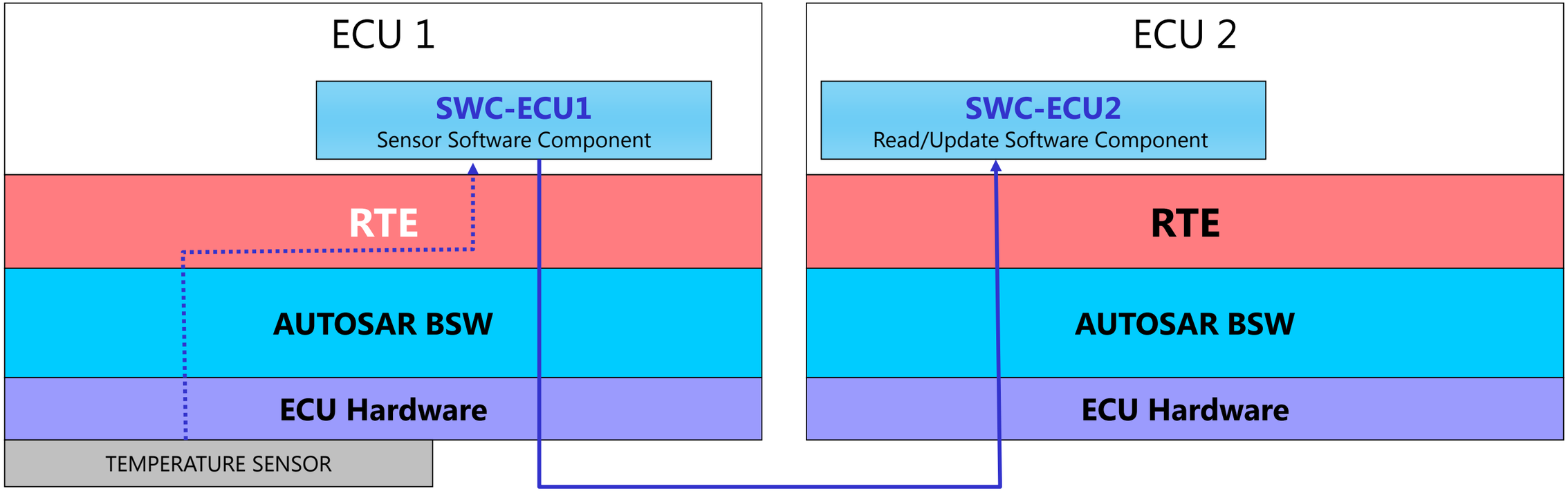
- Communication within one AUTOSAR Software component
- Communication among AUTOSAR Software components in the same ECU
- Communication between AUTOSAR Software component and BSW module



Inter ECU Communication

Inter ECU Communication via RTE & BSW

- Communication of Software components across ECUs in the network
- RTE will invoke AUTOSAR COM to transmit/receive signals

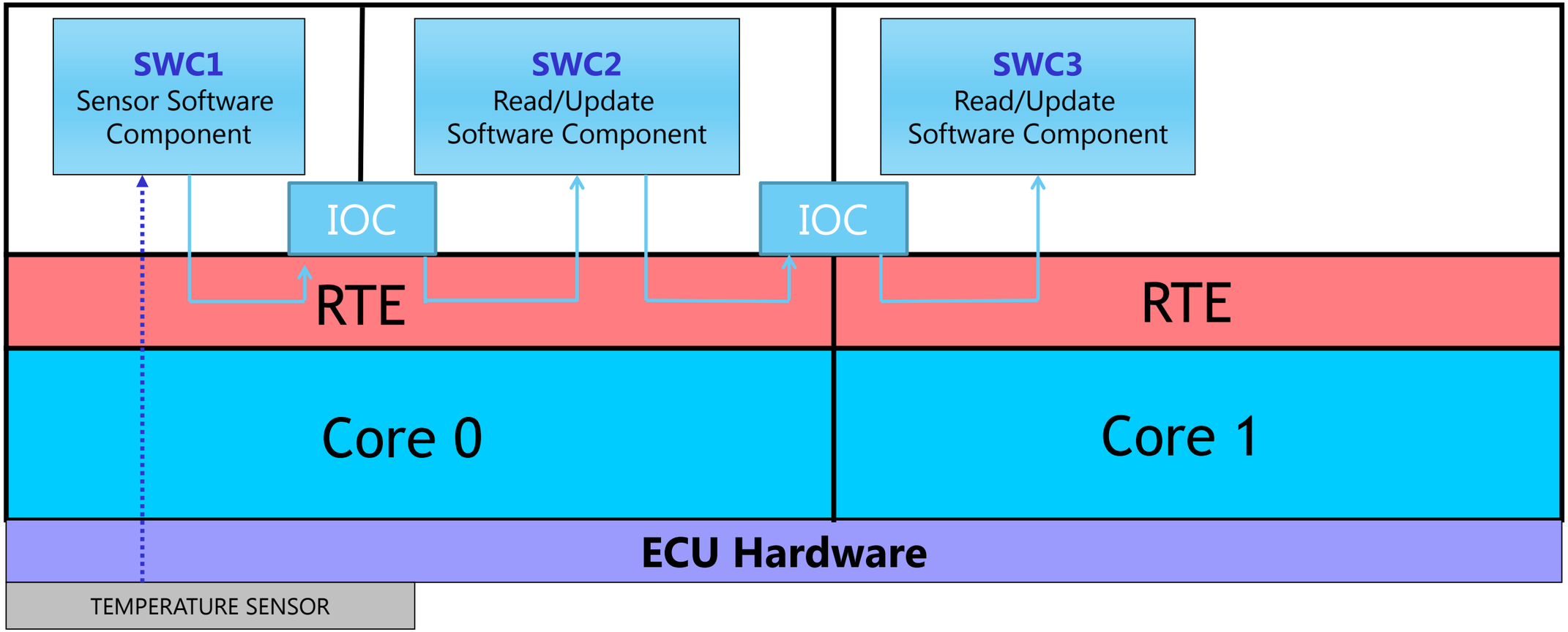


Signal Path / Data Flow in Network

Inter Partition Communication

Inter Partition Communication via RTE and IOC

- Communication between SWCs present in different partitions of the same core of the same ECU
- Communication between SWCs present in different cores of the same ECU
- RTE will use IOC (Inter Os-Application Communication)



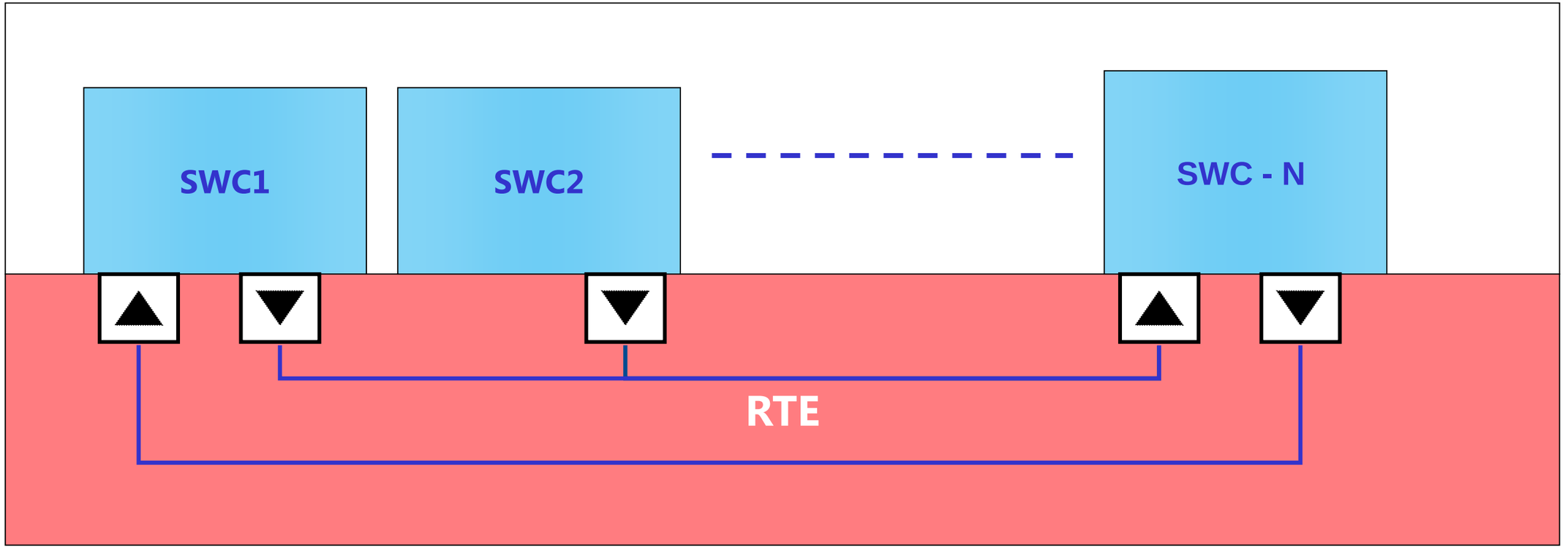
Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

Sender Receiver Communication

Sender-Receiver

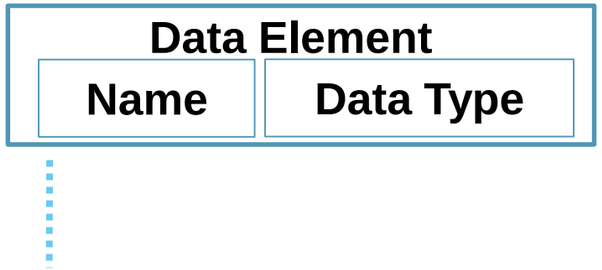
- This provides a one way communication
- A sender provides information to one(1:1) or more receivers(1:m), or one receiver gets information from several senders(n:1)
- Communication can be Implicit or Explicit
- Supports data distribution (Unqueued) and Event distribution (Queued)



Sender Receiver Communication

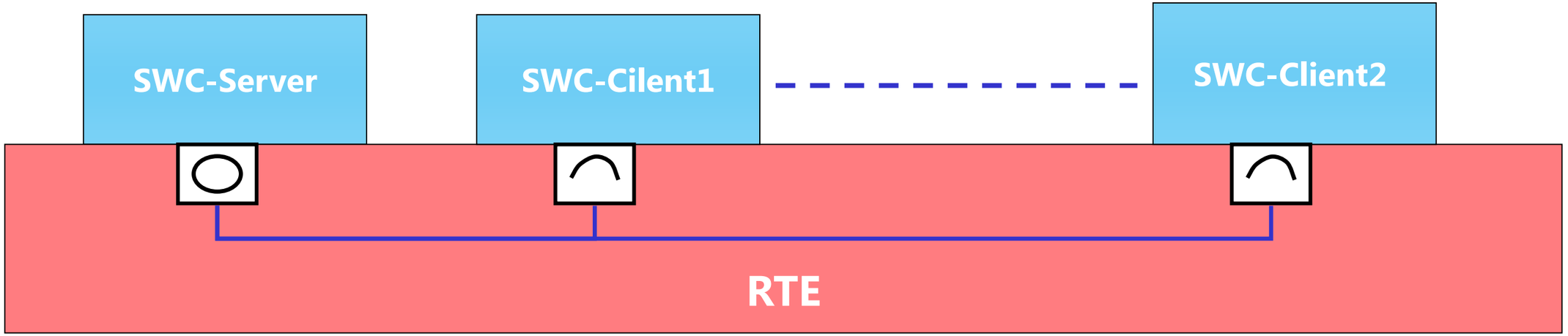
Sender Receiver Communication

- Un-queued
  - Explicit APIs
  - Implicit APIs
- Queued
  - Explicit APIs



Client Server Communication

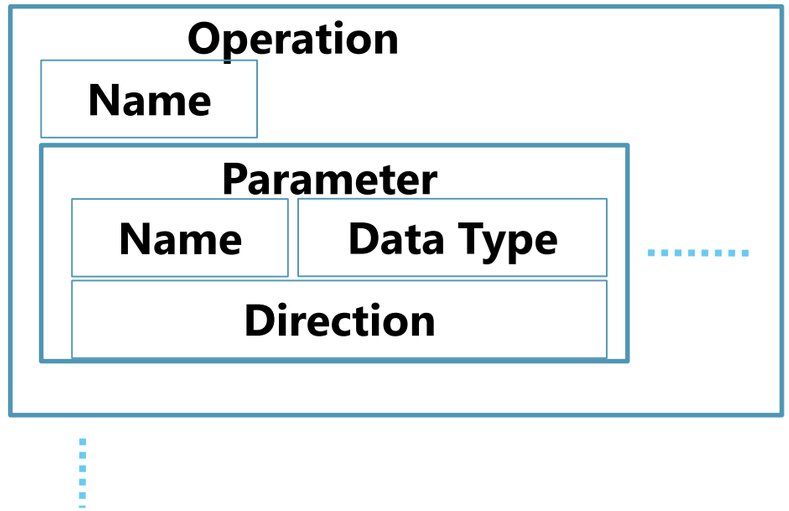
- Client initiates communication, server performs the requested service by client and provides response to client
- Client invocation can be performed synchronously (waiting for server) or asynchronously
- There can be many clients for a server



Client Server Communication

Client Server Communication

- Synchronous
- Asynchronous



SWC Internal Communication

InterRunnableVariables

- InterRunnableVariables are used for communication between runnables inside a software component
- Types of InterRunnableVariable behaviors:
  - InterRunnableVariables with Implicit behavior
  - InterRunnableVariables with Explicit behavior

Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

Modes

Mode Manager

- Entering and leaving modes is initiated by a mode manager
- The mode manager contains the master state machine to represent the modes
- Mode Manager can be basic software module like EcuM, ComM. It also can be AUTOSAR Software Component like Application Mode Manager

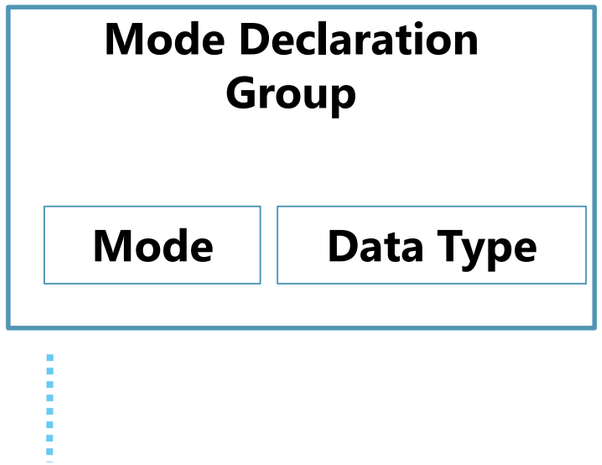
Mode User

The mode user is the one who uses these modes

Modes

Mode Management

- Mode Manager
- Mode User



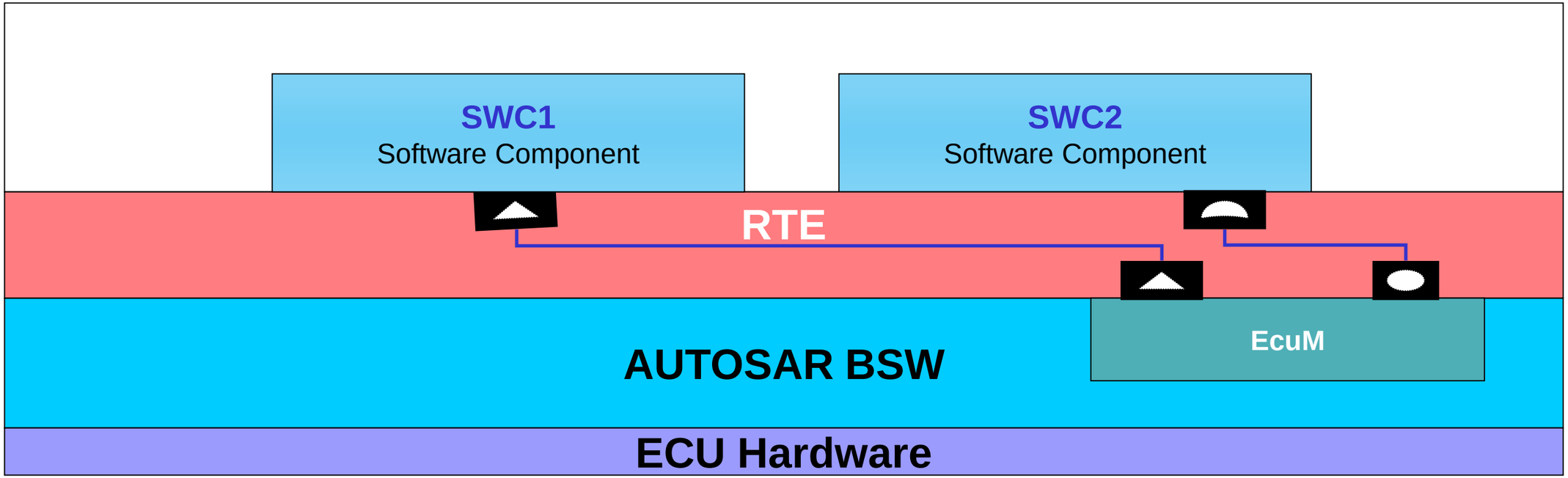
Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

RTE And AUTOSAR Service

RTE And AUTOSAR Service

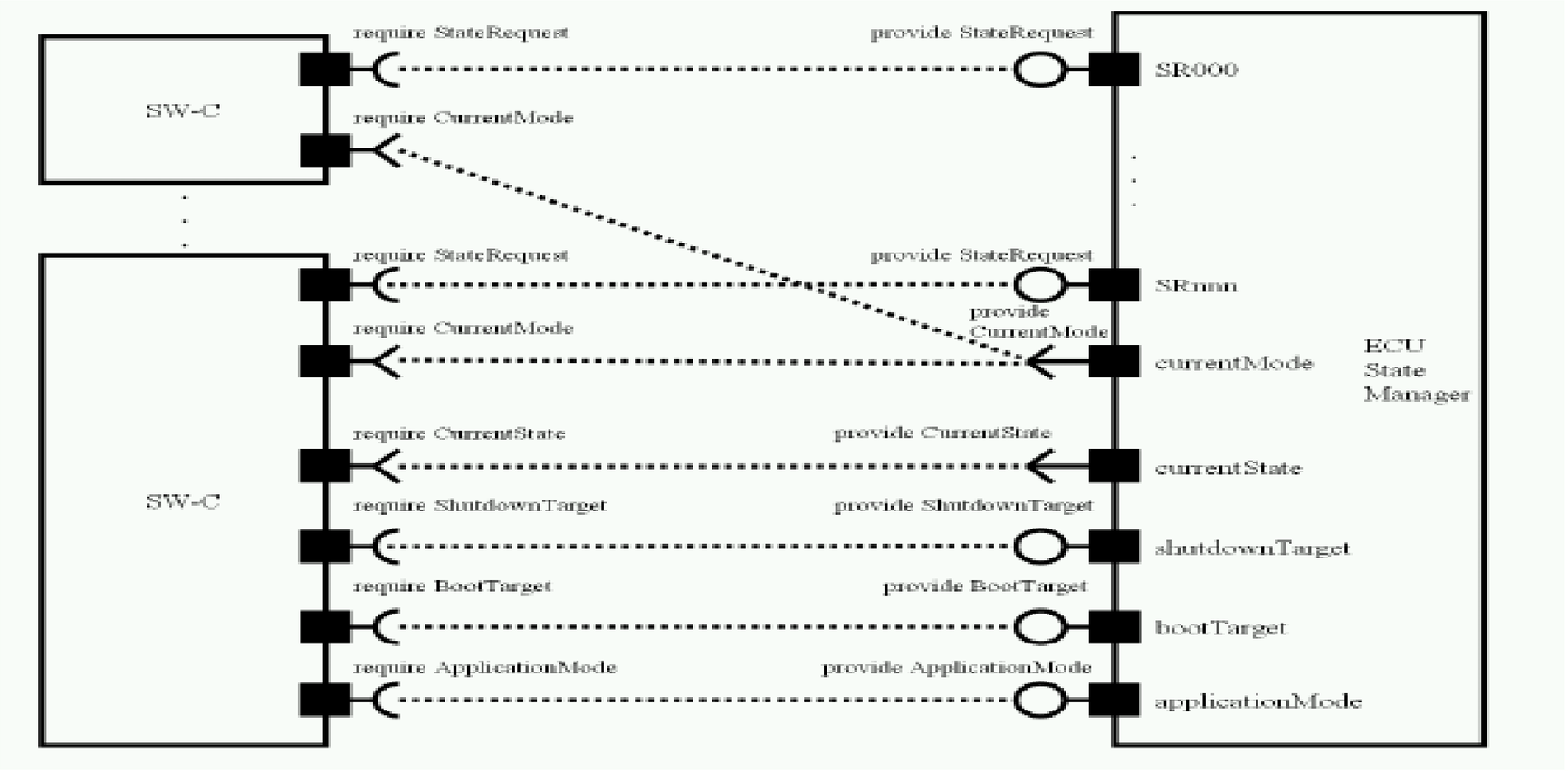
- RTE provides standardized AUTOSAR interface for AUTOSAR BSW and AUTOSAR Software component to communicate
- Depending on the nature of service, there can be a client server interface or sender-receiver interface
- RTE supports services to EcuM, ComM, WdgM, NvM, DEM, DCM and FIM modules



RTE And AUTOSAR Service

EcuM Service Component

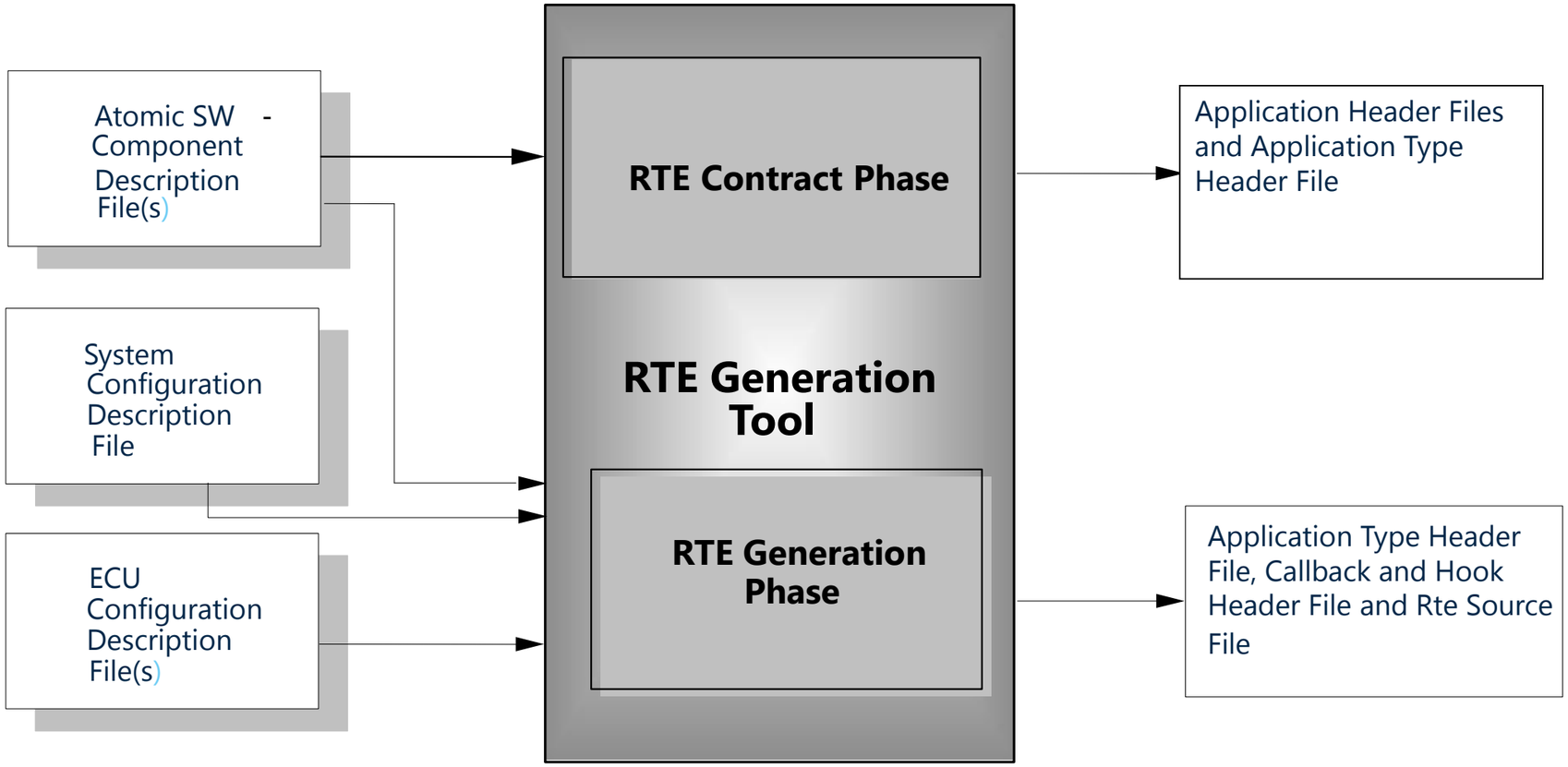
Overall ports Configuration of EcuM Service component and its connected Software Components



Agenda

1. Introduction to RTE
2. Interfaces in AUTOSAR
3. RTE Entities
4. Intra ECU, Inter ECU, Inter Partition Communication
5. Communication Models
6. Modes
7. RTE And AUTOSAR Service
8. Calibration
9. RTE Generation

RTE Generation



RTE Generation

RTE Generation Tool is a command line tool developed using Perl language.

Inputs:

- Software Component Description Files
- ECU Configuration Description File of RTE
- ECU Configuration Description File of OS

Optional Inputs:

- System Configuration Description which includes Data to Signal Mapping
- ECU Configuration Description File of COM

RTE Generation

Output File	Details
Rte.c	Rte Generation Tool generates C Source file namely Rte.c. Rte.c contains an instance of the relevant Component Data Structure for each software-component instance on the ECU for which the Rte is generated. It shall contain Rte API's for each software component instance configured. It shall also define task bodies for tasks. The RTE COM callbacks are generated if configured.
Rte_Type.h	Rte Generation Tool generates AUTOSAR Types header file called Rte_Type.h. Rte_Type.h defines the AUTOSAR data types and Rte implementation types.
Rte Application Header file	Rte Generation Tool generates an Application Header file called 'Rte_<SW-Component name>.h' for each Atomic SW-Component configured.
Rte_Hook.h	Rte Generation Tool generates VFB Tracing header namely Rte_Hook.h. It defines the configured VFB Trace events. This file is generated only when VFB Trace Events are configured and VFB Trace is enabled.
Rte_Cbk.h	RTE Generation Tool generates header file containing the prototype of COM callback functions namely Rte_Cbk.h. It defines the COM callbacks for relevant signals. This file is generated only when RTE COM callbacks are configured.

Questions

Thank You

www.kpit.com