

# IEEE P802.1Qdd Resource Allocation Protocol (RAP)

## RAP Participant in Draft 0.3

Feng Chen

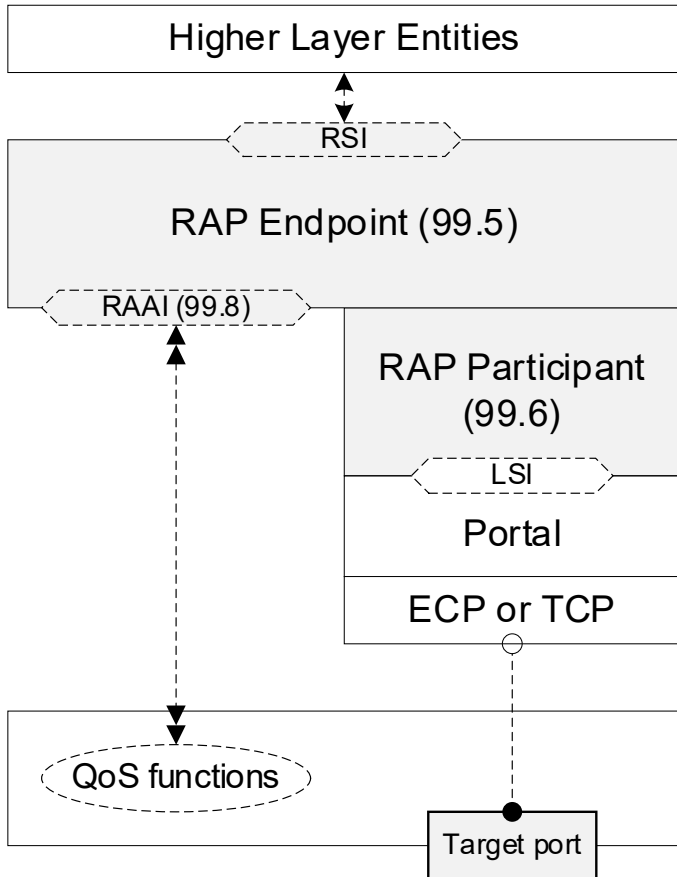
Siemens AG

IEEE 802.1 Plenary Session, November 2020

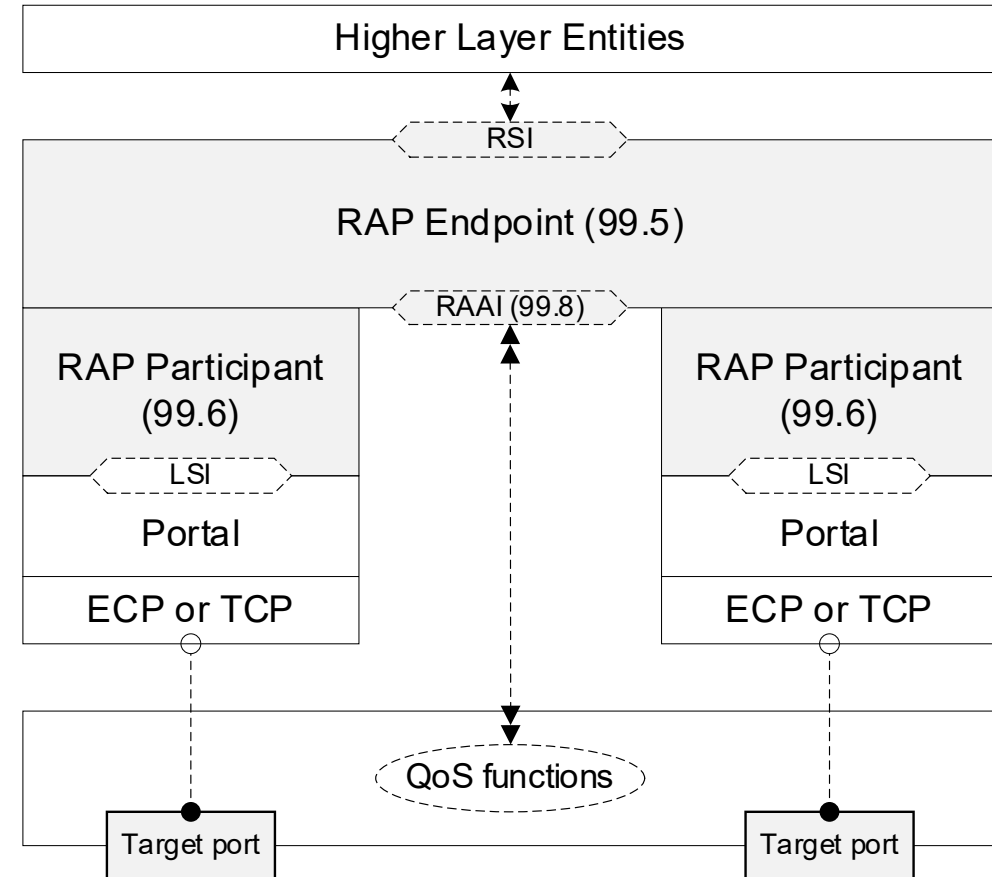
# Overview

- This slide deck gives an introduction to RAP Participant specified in subclause 99.6 in [IEEE P802.1Qdd-D0.3](#).
- Please also refer to the Introductory notes section contained in the draft for other changes made in this version.

# RAP End Station Architecture

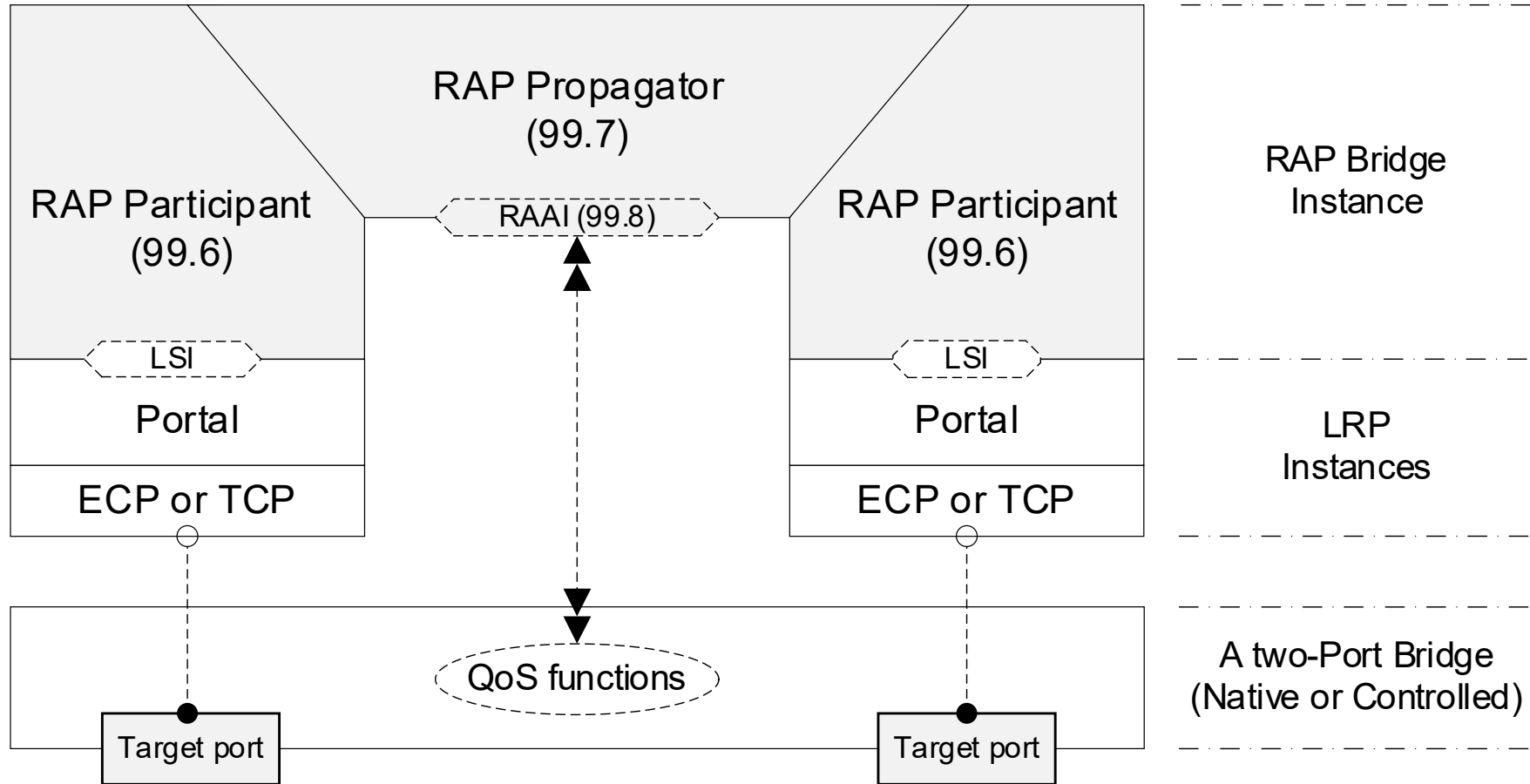


**Single-port end stations**



**Multiple-port end stations**  
(for use with .1CB E2E redundancy)

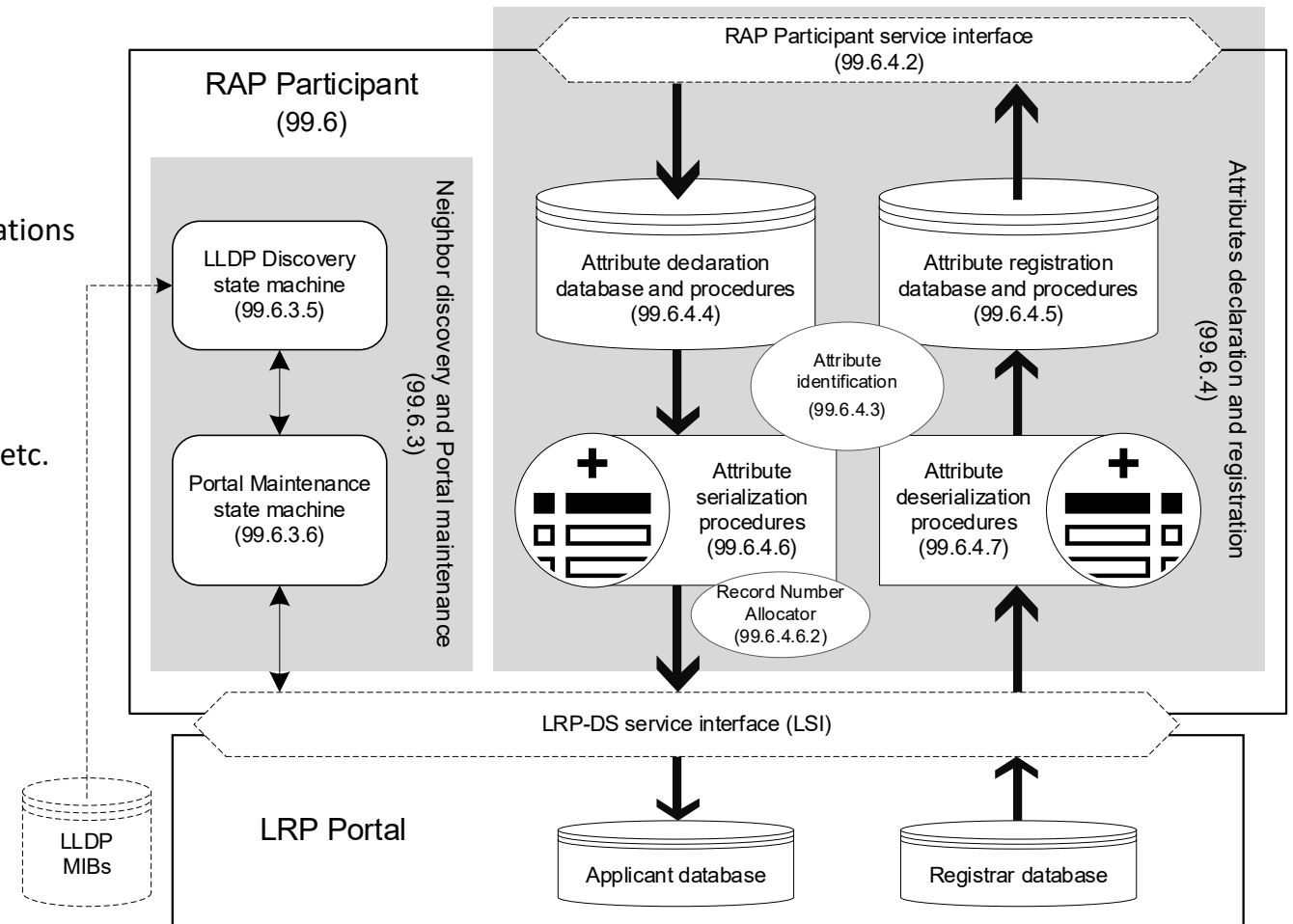
# RAP Bridge Architecture



# RAP Participant Architecture

## Components and functions:

- RAP Participant service interface
  - attachment to a RAP Propagator or a RAP Endpoint
  - A set of primitives for attribute declarations and registrations
- Neighbor discovery
  - LLDP, static configuration and Exploratory Hello
- Portal maintenance
  - Handling of Portal creation, association, disconnection, etc.
- Attribute identification
  - The only RAP attributes' semantics-aware component
- Attribute declaration database
- Attribute registration database
- Attribute-to-record serialization
  - Record Number Allocator (implementation-specific methods)
- Record-to-attribute deserialization



# Neighbor Discovery

A RAP Participant can be configured to operate in one of the following three neighbor discovery modes (99.6.3.2) to obtain the neighbor target port information:

- **LLDP discovery:** via the LLDP Discovery SM (99.6.3.5) that examines LRP Discovery TLVs received in LLDP remote systems MIB and triggers the Portal Maintenance SM (99.6.3.6) to initiate a Portal creation.
- **Static configuration:** via a managed object (99.6.3.3.7) for administratively configured values.
- **Exploratory Hello:** no explicit neighbor information needed, but only used with ECP in Native systems.

Regardless which discovery mode is used, the local target port information is always obtained from the same managed object (99.6.3.3.6).

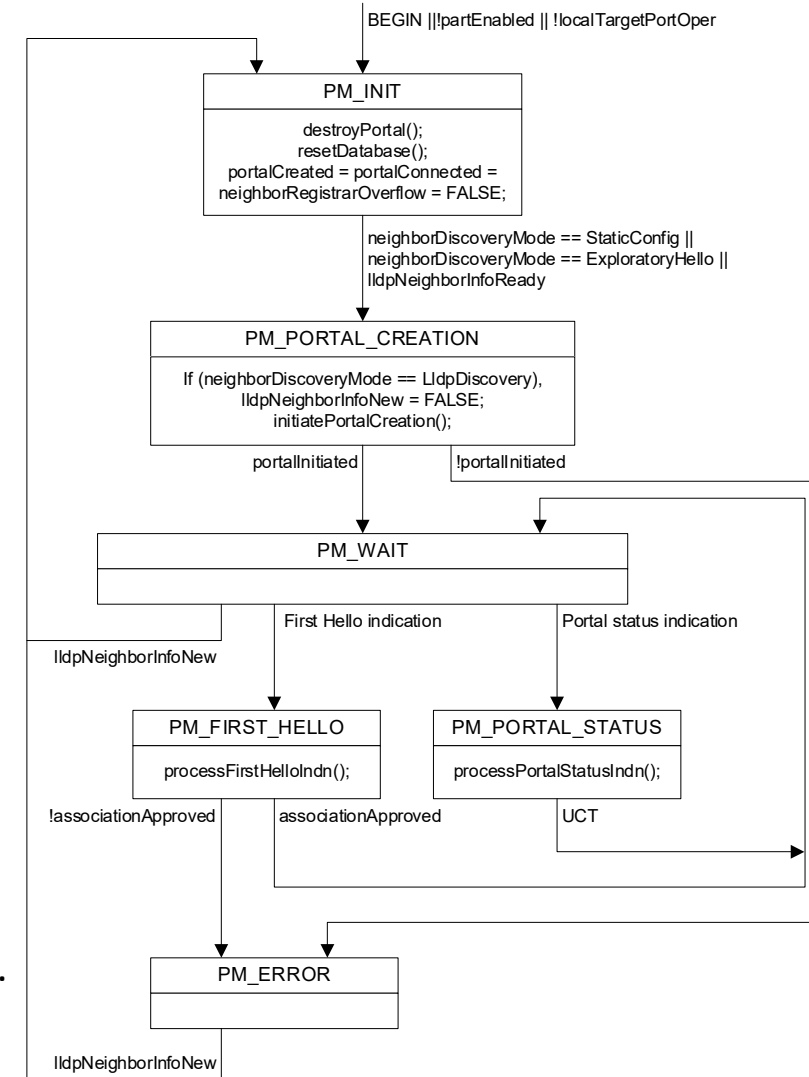
The Table 99-3 provides a summary of the parameter values as inputs to the LRP primitives that are used to issue a Portal creation request to LRP.

# Portal Maintenance

## The Portal Maintenance SM (99.6.3.6) handles:

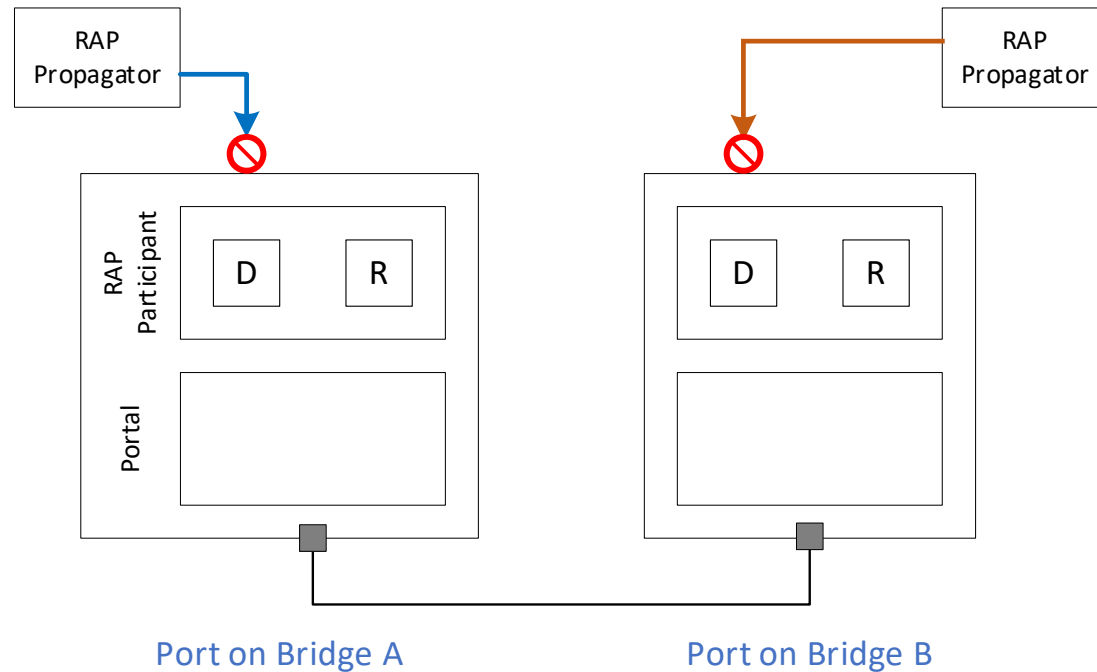
- Initiation of the Portal creation, using the values that are neighbor discovery mode dependent (see Table 99-3).
- Approval or disapproval of Portal association
  - By checking the Application Information TLV in First Hello Indication (only a RAP Version Number in D0.3).
- Portal status reported by LRP
  - *Disconnected (hello timeout)*: reset registration database (see next slides)
  - *Neighbor registrar database overflow*: undefined by D0.3
  - *Neighbor registrar database correct*: undefined by D0.3
- Portal destruction and recreation, triggered by one of the following
  - System reinitialization.
  - Changes in the operational status of the local target port (MAC\_Operational).
  - Changes in the RAP Participant enable/disable parameter (partEnabled).
  - When LLDP is deployed and detects changes in previously discovered neighbor info.

**Question:** does CFM reset MAC\_Operational when detecting connectivity faults?

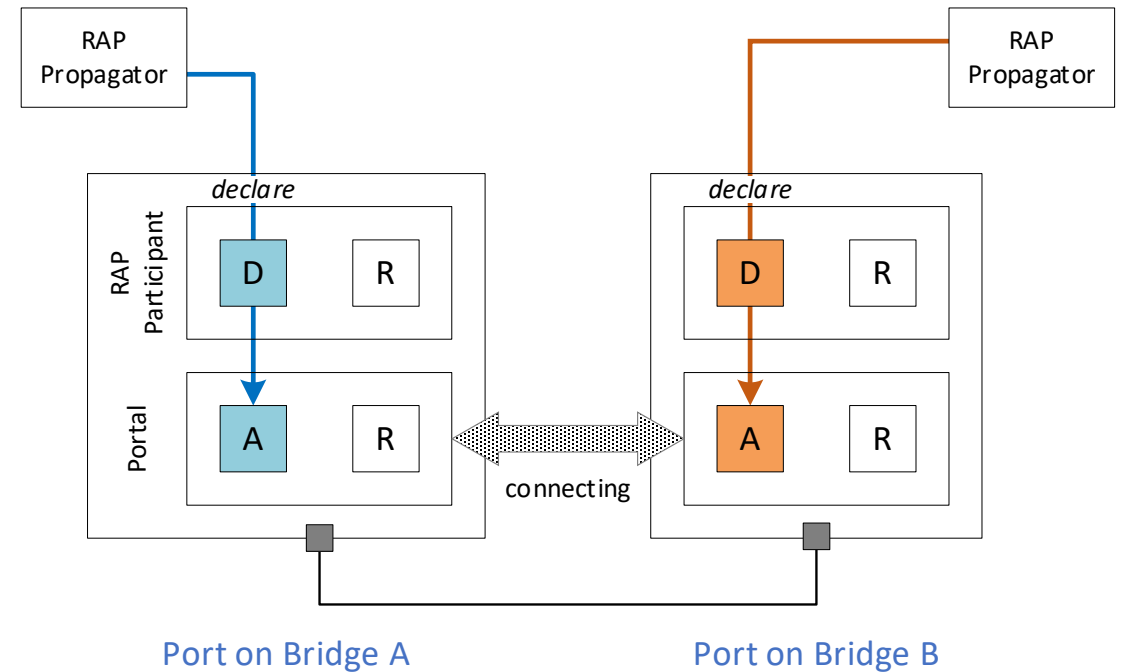


# Databases and Portal Status

## Portal creation in progress



## Portal created and connecting



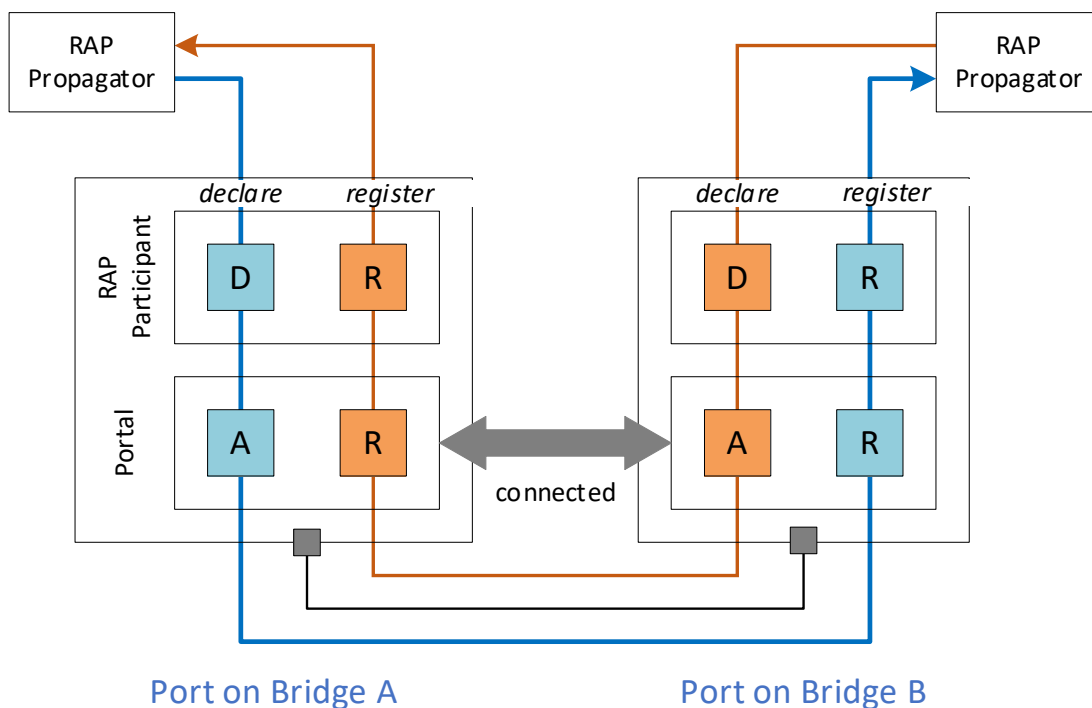
- Access to the Declaration DB for attribute declarations will be blocked, until a Portal creation is created, as indicated by a PortalId received from LRP.

- Attribute declarations can already be made in the Declaration DB and passed to the Applicant DB (but not passed to the neighbor), even while the Portal is connecting to its neighbor.

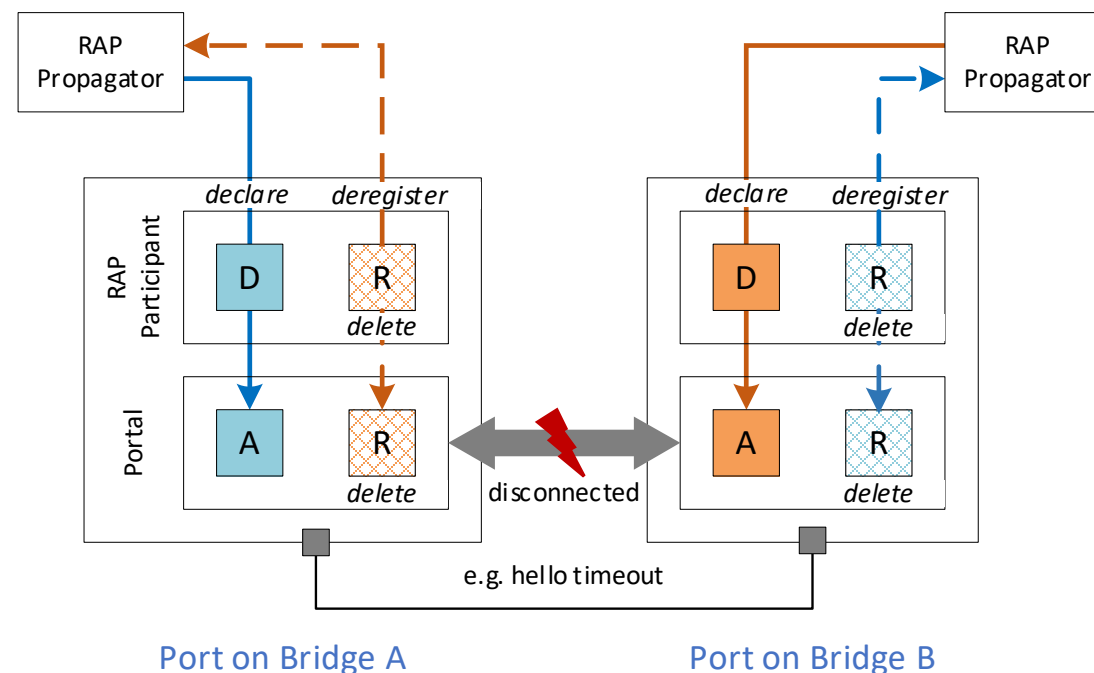


# Databases and Portal Status (Conti.)

Portal connected



Portal connection interrupted



- A RAP Participant is fully operational, when a Portal connection has been established, as indicated by a “connected” status received from LRP.

- When a RAP Participant is notified of a “disconnected” by LRP, both its Registration DB and the underlying Registrar DB are reset, while attribute declarations can still be made unless the Portal is destroyed.

# Serialization and Deserialization

## Attribute-to-Record serialization (99.6.4.6)

- Pack attributes into records and pass them to the Applicant.
  - One or more attributes can be packed into a record according a specific scheme.
  - A **Record Number Allocator** (99.6.4.6.2) is used to allocate record numbers for attributes.
    - Just defined as an abstract function without enforcing a particular allocation scheme (implementation-specific)
    - Only requiring that the used scheme shall not cause any resulted record to exceed the maximum record size, which is dependent on the used LRP-DT mechanism (ECP or TCP).
- ❖ **ISSUE:** LRP does not report to its application which LRP-DT mechanism has been chosen to transport LRPDUs, if the application indicated support for both ECP and TCP at requesting a Portal creation.

## Record-to-Attribute deserialization (99.6.4.7)

- Unpack records received from the Registrar into attributes.
- Store the Record numbers currently maintained by the Registrar DB, for use in potential Delete record operations if resetting the Registrar DB is needed.

# Thanks