HPC.NRW

MPI in Small Bites

HPC.NRW Competence Network

Communicator Handling

HPC.NRW Competence Network

MPI in Small Bites

Communication Contexts

HPC.NRW

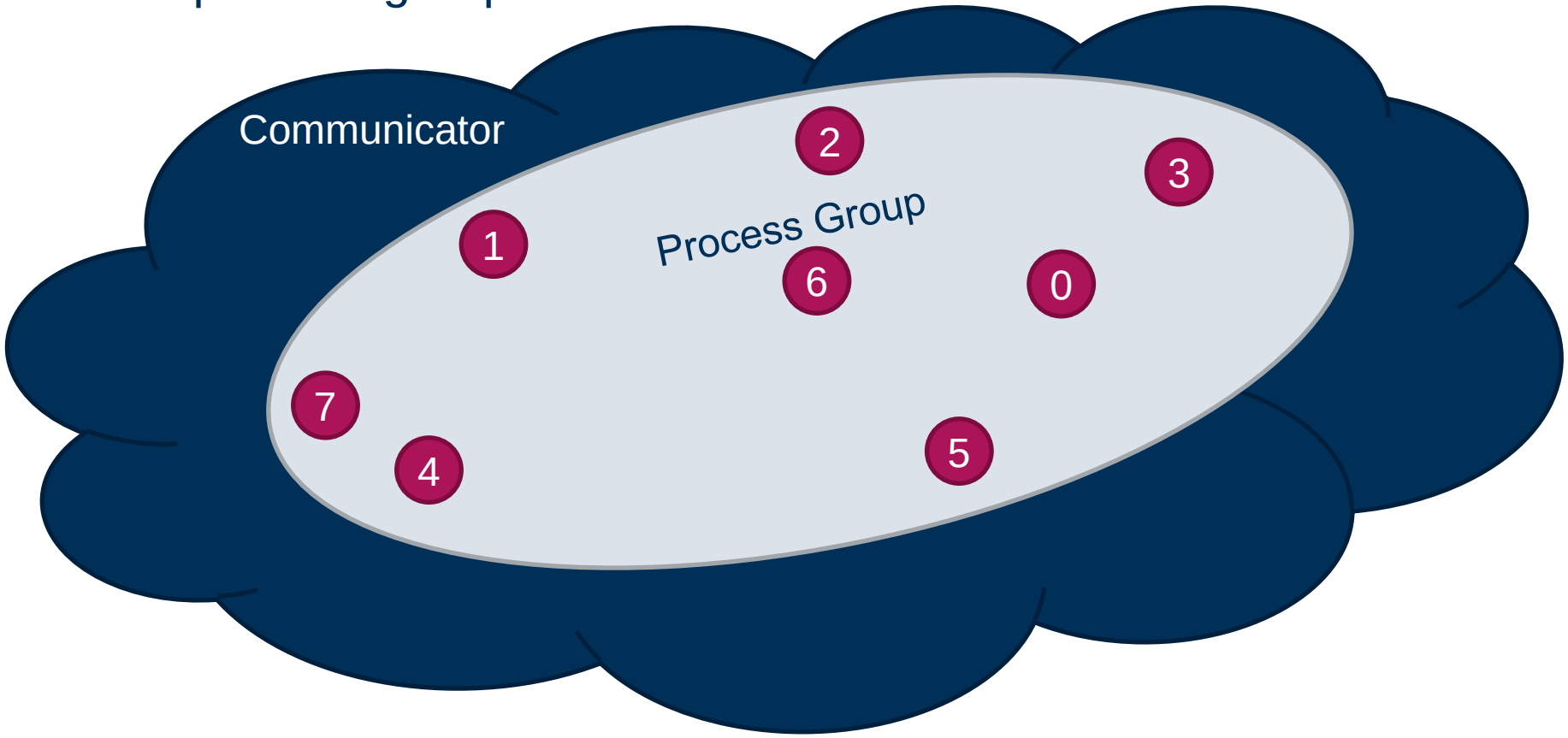
- Defines context for each communication operation in MPI
  - Group of participating peers (process group)
  - Error handlers for communication and I/O operations
  - Local key/value cache
  - Virtual topology information (optional)

- Two predefined communicators (pre MPI 4.0 and MPI 4.0 World Model):
  - **MPI_COMM_WORLD**
    contains all processes launched **initially** as part of the MPI program
  - **MPI_COMM_SELF**
    contains only the current process

Communicators

HPC.NRW

– Communicator – process group – ranks



Query Operations

HPC.NRW

- Obtain the size of the process group of a given communicator:

```
MPI_Comm_size (MPI_Comm comm, int *size)
```

  - ranks in the group are numbered from 0 to size-1
- Obtain the rank of the calling process in the given communicator:

```
MPI_Comm_rank (MPI_Comm comm, int *rank)
```

- Special “null” rank – MPI_PROC_NULL
  - member of any communicator
  - can be sent messages to – results in a no-op
  - can be received messages from – zero-size message tagged **MPI_ANY_TAG**
  - use it to write symmetric code and handle process boundaries

Message Envelope Matching

HPC.NRW

– Recall: message envelope

	Sender	Receiver
Source	Implicit	Explicit, wildcard possible (**MPI_ANY_SOURCE**)
Destination	Explicit	Implicit
Tag	Explicit	Explicit, wildcard possible (**MPI_ANY_TAG**)
Communicator	Explicit	Explicit

– Cross-communicator messaging is not possible
  – messages sent in one communicator can only be received by ranks in the same communicator
  – communicators can be used to isolate communication to prevent interference and tag clashes – useful when writing parallel libraries

Communicator creation

HPC.NRW

- Duplicate an existing communicator
  - MPI_Comm_dup, MPI_Comm_dup_with_info, MPI_Comm_idup
- Create new communicator for a subgroup of a communicator
  - MPI_Comm_create, MPI_Comm_create_group
- Split an existing communicator
  - MPI_Comm_split, MPI_Comm_split_type

Creating Communicator Copies

HPC.NRW

– Duplicate a given communicator:

```
MPI_Comm_dup (MPI_Comm comm, MPI_Comm *newcomm)
```

  – New communication context with same ranks and ordering
  – Easy isolation of encapsulated communication
    – Libraries should never communicate on MPI_COMM_WORLD directly

Destroying Communicators

HPC.NRW

- Communicators take up memory and other precious resources
- Should be freed once no longer needed

```
MPI_Comm_free (MPI_Comm *comm)
```

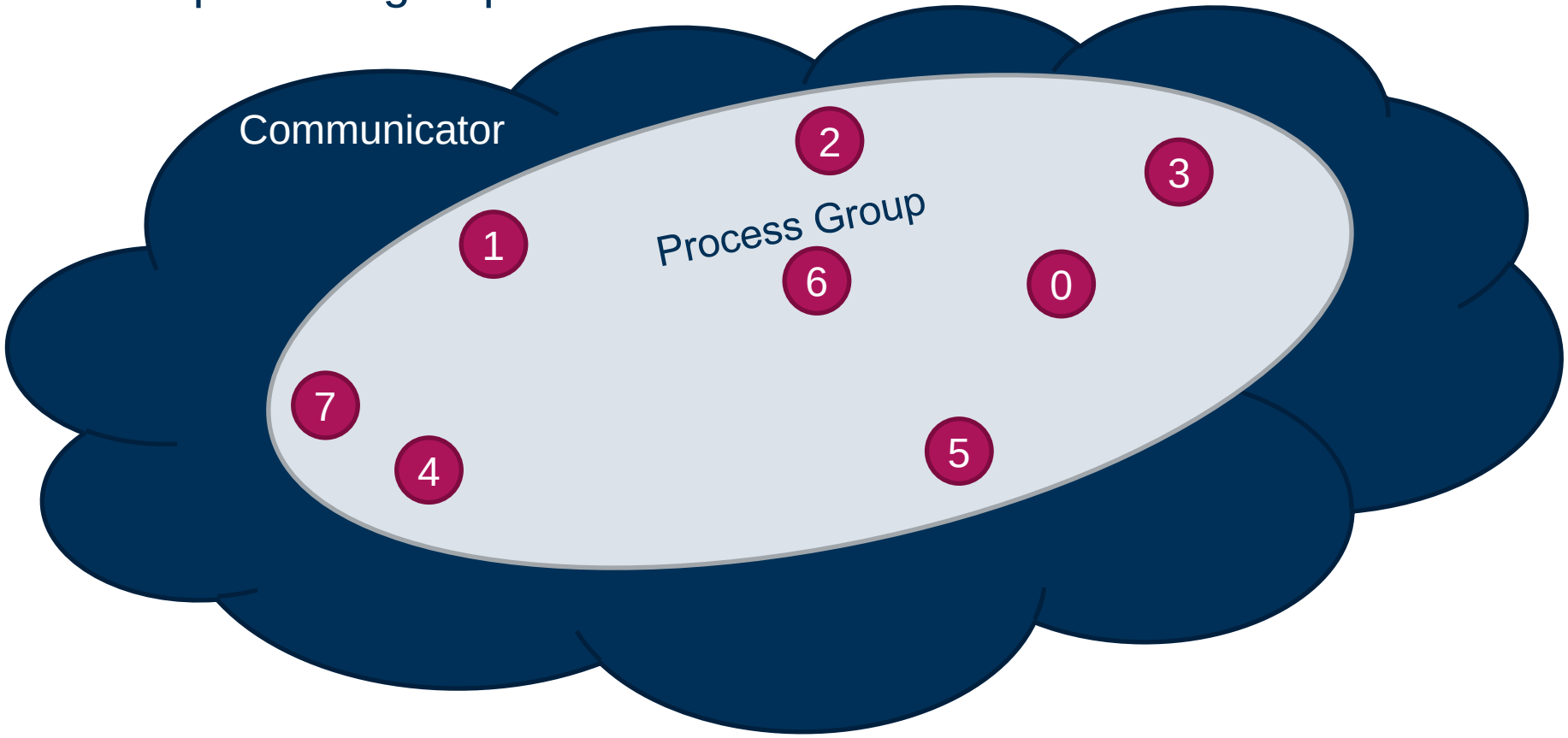
  - Marks **comm** for deletion
  - **comm** is set to **MPI_COMM_NULL** on return
  - The actual communicator object is only deleted once all pending operations are completed

- It is erroneous to free predefined communicators MPI_COMM_WORLD, MPI_COMM_SELF or MPI_COMM_NULL

Communicators

HPC.NRW

– Communicator – process group – ranks



Communicator creation

HPC.NRW

- Duplicate an existing communicator
  - MPI_Comm_dup, MPI_Comm_dup_with_info, MPI_Comm_idup
- Create new communicator for a subgroup of a communicator
  - MPI_Comm_create, MPI_Comm_create_group
- Split an existing communicator
  - MPI_Comm_split, MPI_Comm_split_type

Communicator creation

HPC.NRW

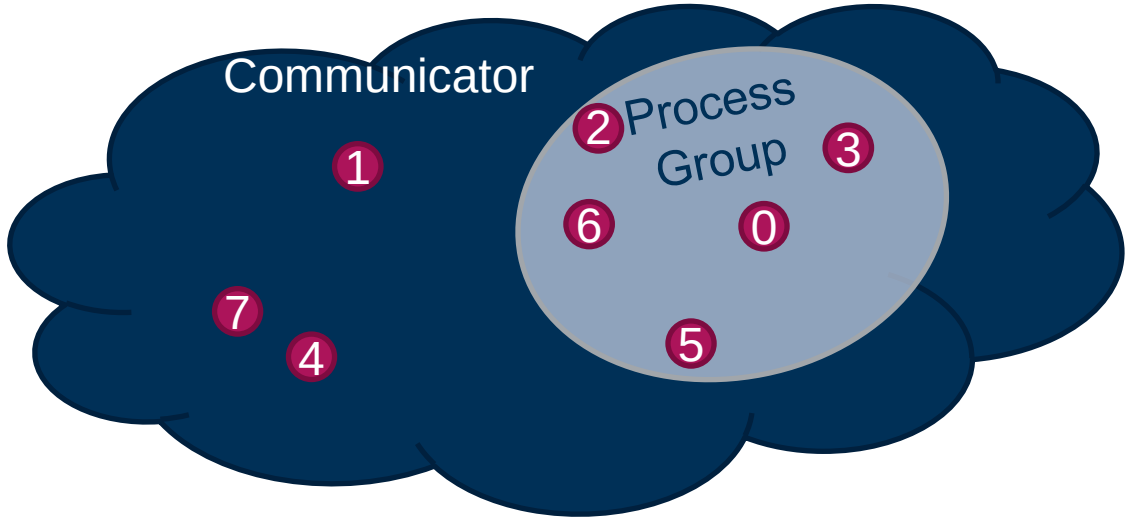
– Create new communicator for a subgroup of a communicator

```
MPI_Comm_create (MPI_Comm comm, MPI_Group group, MPI_Comm *newcomm)
```

– Collective in comm (for ranks ∉ group: newcomm=MPI_COMM_NULL)

```
MPI_Comm_create_group (MPI_Comm comm, MPI_Group group, int tag,
                       MPI_Comm *newcomm)
```

– Collective in group



Communicator splitting

HPC.NRW

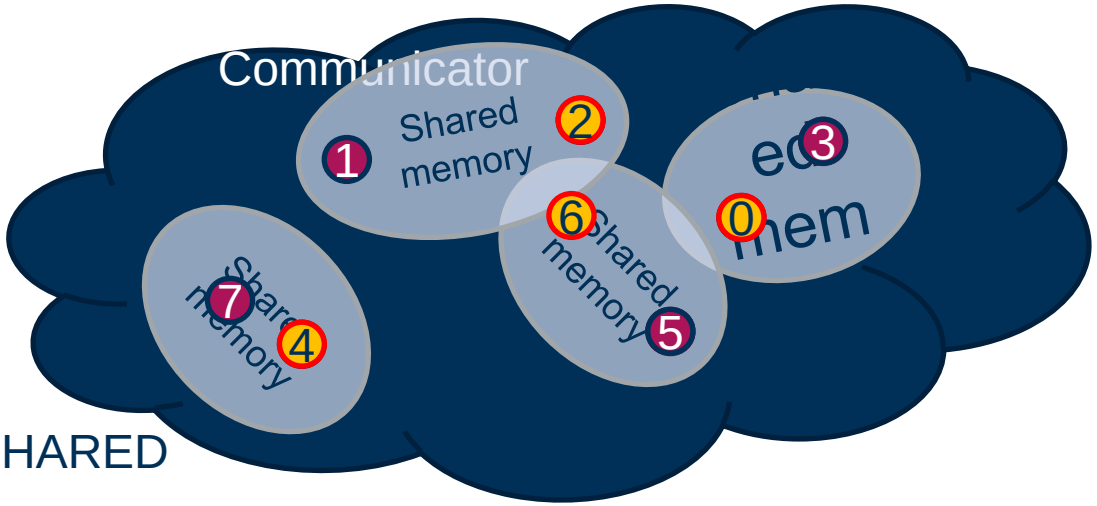
- Split existing communicators into parts

```
MPI_Comm_split (MPI_Comm comm, int color, int key, MPI_Info info, MPI_Comm *newcomm)
```

  - Split by some characteristics (e.g., rank % n, rank < n, rank / n)

```
MPI_Comm_split_type (MPI_Comm comm, int split_type, int key, MPI_Info info, MPI_Comm *newcomm)
```

  - Split into shared memory groups
  - key controls the numbering within newcomm



color = rank%2
split_type = MPI_COMM_TYPE_SHARED