An Approach for Deploying Externally Defined MPI Communicators at Runtime

Carsten Clauss, Stephan Gsell,
Stefan Lankes, Thomas Bemmerl

Kommunikation in Clusterrechnern und Clusterverbundsystemen (KiCC), Aachen 2007

LEHRSTUHL FÜR BETRIEBSSYSTEME
Univ.-Prof. Dr. habil. Thomas Bemmerl

RWTH AACHEN UNIVERSITY

Übersicht

- Einleitung und Motivation
- Kommunikatoren in MPI
- Extern definierte MPI Kommunikatoren
- Anwendungsbeispiel
- Zusammenfassung und Ausblick

Einleitung und Motivation

- **Gruppierung von parallelen Prozessen**

→ **unterschiedliche Prozessgruppen bearbeiten unterschiedliche Teilaufgaben**

→ **Beispiele für Kontrollflussaufteilung:**

- Master-Slave-Ansatz

- Pipeline Parallelisierung

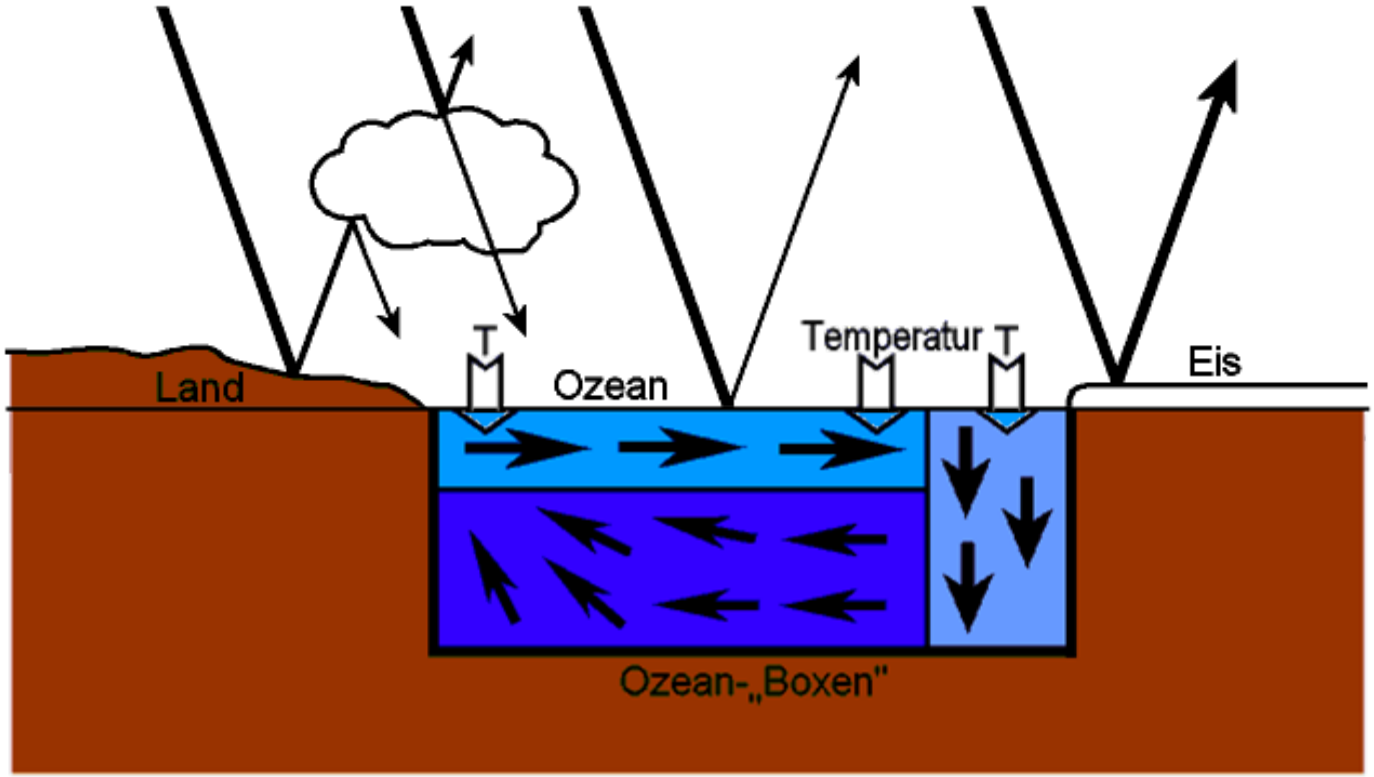
- gekoppelte Algorithmen

. . .

Einleitung und Motivation

- **Gekoppeltes (Simulations-) Algorithmen**

Bsp.: Atmosphäre-Ozean-Kopplung



Einleitung und Motivation

- **Gruppierung von parallelen Prozessen**

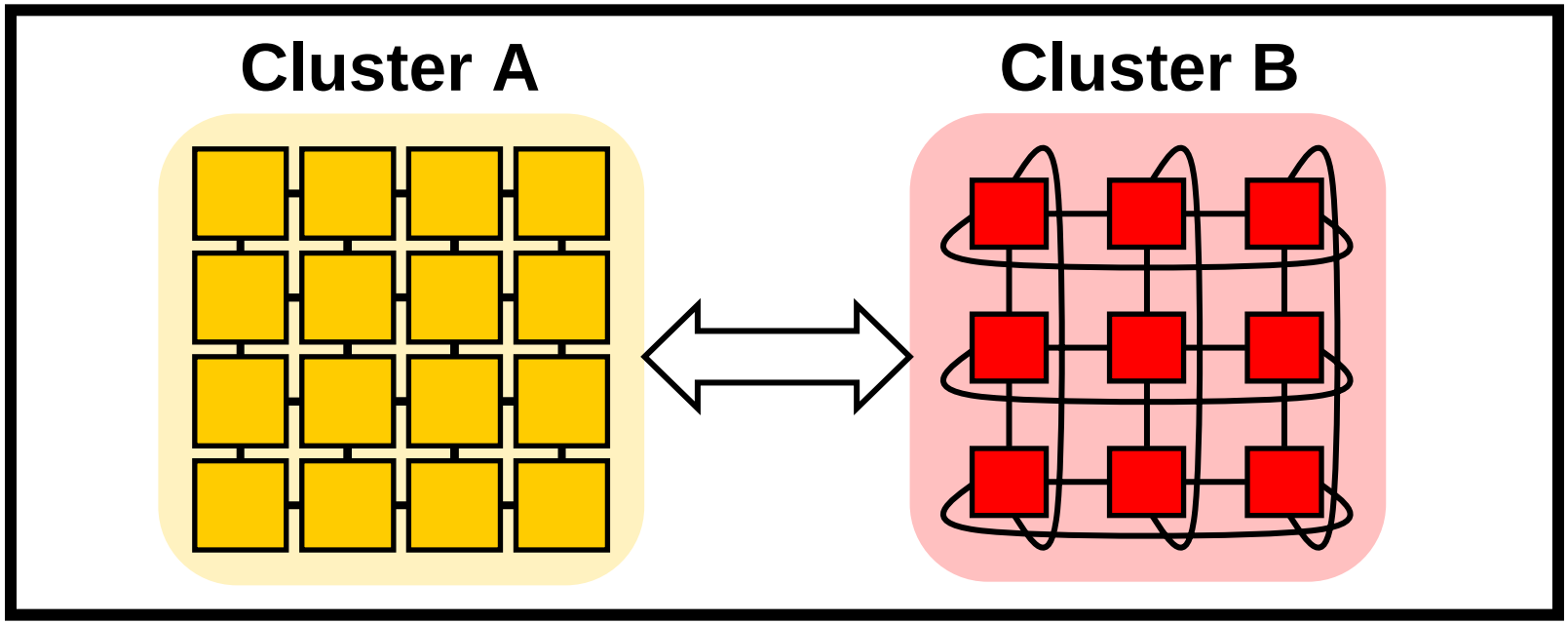
→ ***Top-Down:***
der Algorithmus gibt die Gruppierung der Prozesse vor

→ ***Bottom-Up:***
die (heterogene) Topologie des Rechner-systems *empfiehlt* eine Gruppierung der Prozesse

(Bsp.: gekoppelte Cluster / Metacomputer)

Einleitung und Motivation

- **Gekoppelte Cluster-Systeme:**



Metacomputer

- **Inter-Cluster-Netzwerk:** ***Flaschenhals !!!***
  → Gruppierung der Prozesse

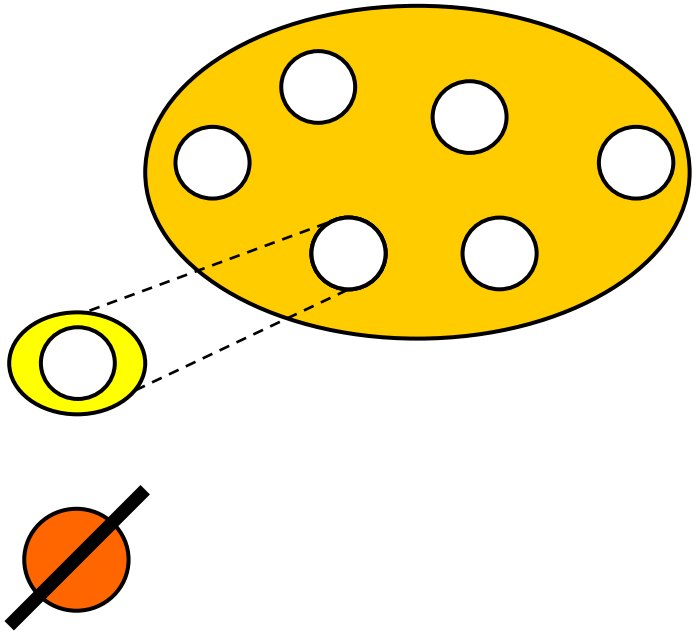
Kommunikatoren in MPI

- **Kommunikatoren ← Prozessgruppen**
- **Vordefinierte Kommunikatoren:**

  → **`MPI_COMM_WORLD`**

  → **`MPI_COMM_SELF`**

  → **`MPI_COMM_NULL`**



- **Kommunikations-Domäne / -Kontext**

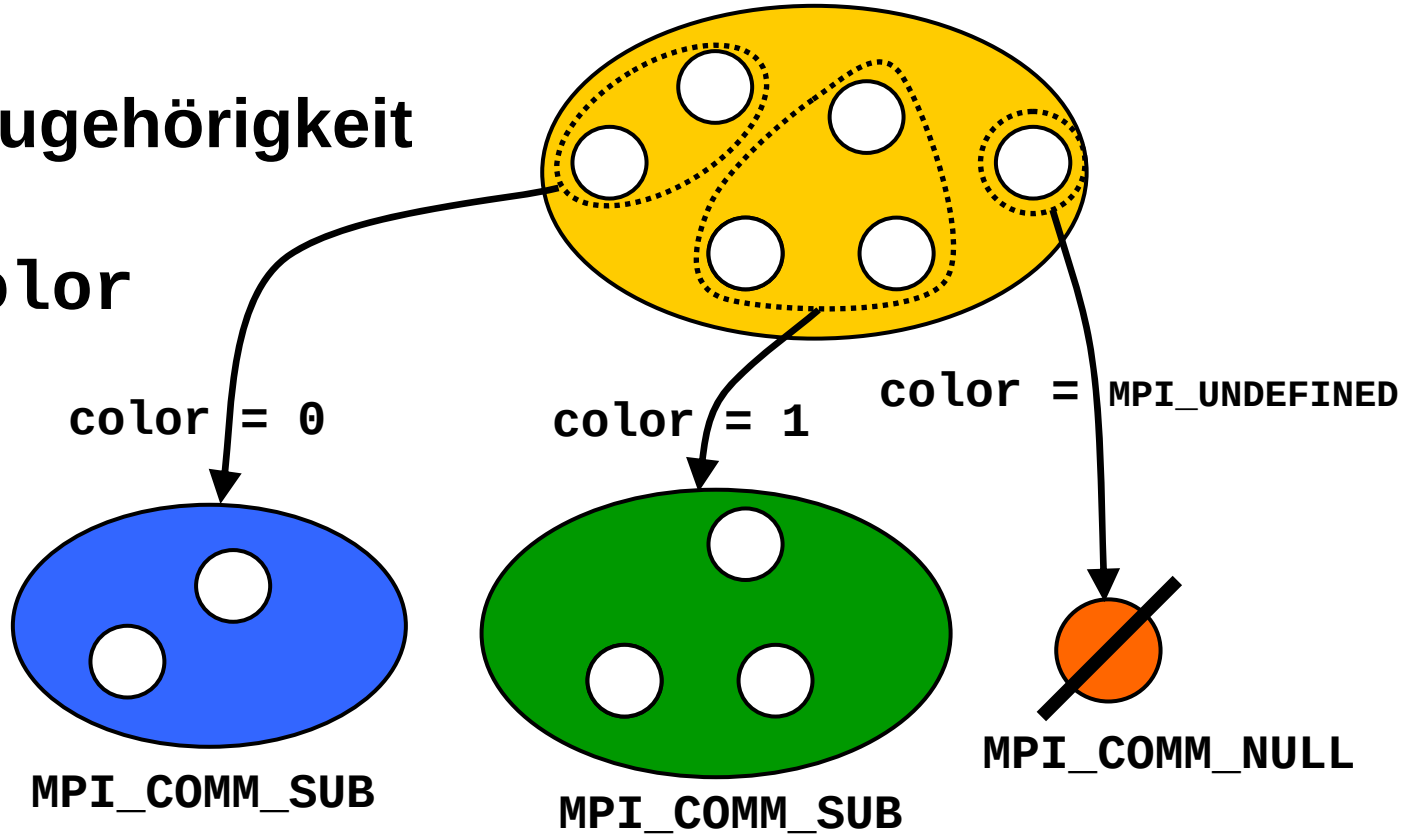
Kommunikatoren in MPI

- **Erzeugung eigener Kommunikatoren**

  → `MPI_Comm_split(comm,color,…,newcomm)`

- **Gruppenzugehörigkeit**

  → `int color`

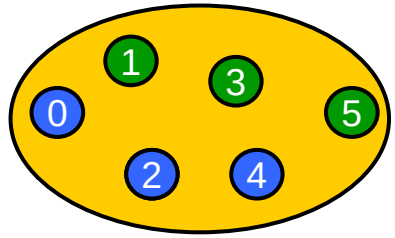


Kommunikatoren in MPI

- **Bestimmung der Gruppenzugehörigkeit**

→ ***Top-Down:*** **anhand des Prozess-Ranges**

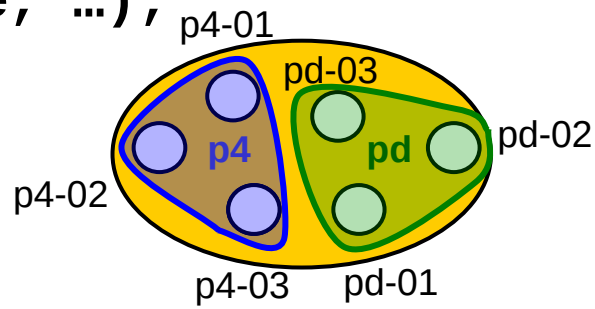
```
MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

if (my_rank == …)
   color = …
```



→ ***Bottom-Up:*** **anhand des Prozessor-Namens**

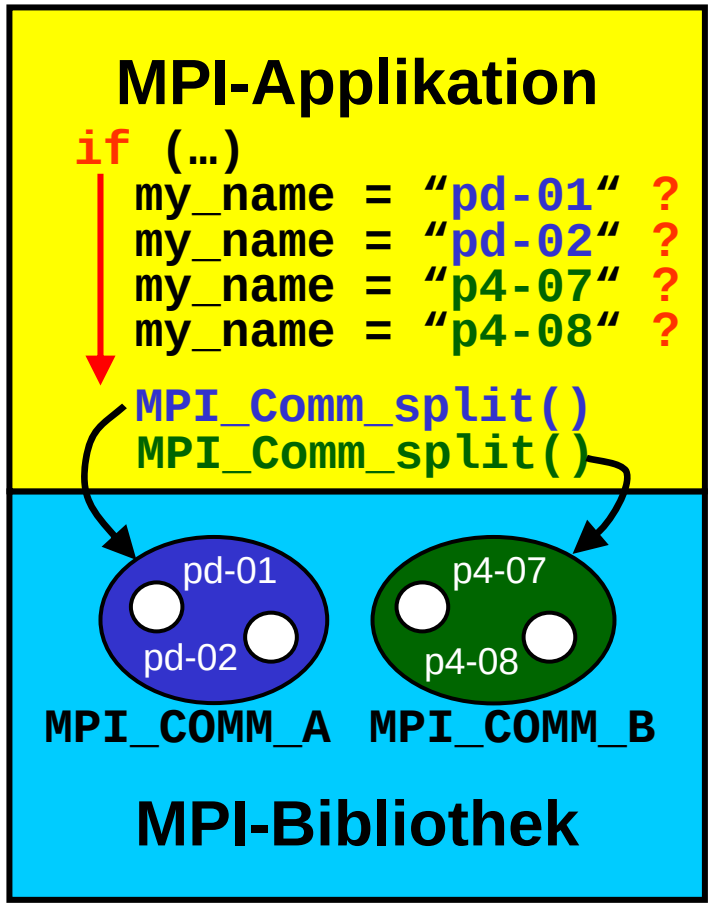
```
MPI_Get_processor_name(my_name, …);

if (strncmp(my_name, …) == …)
  color = …
```

"p4-01", …



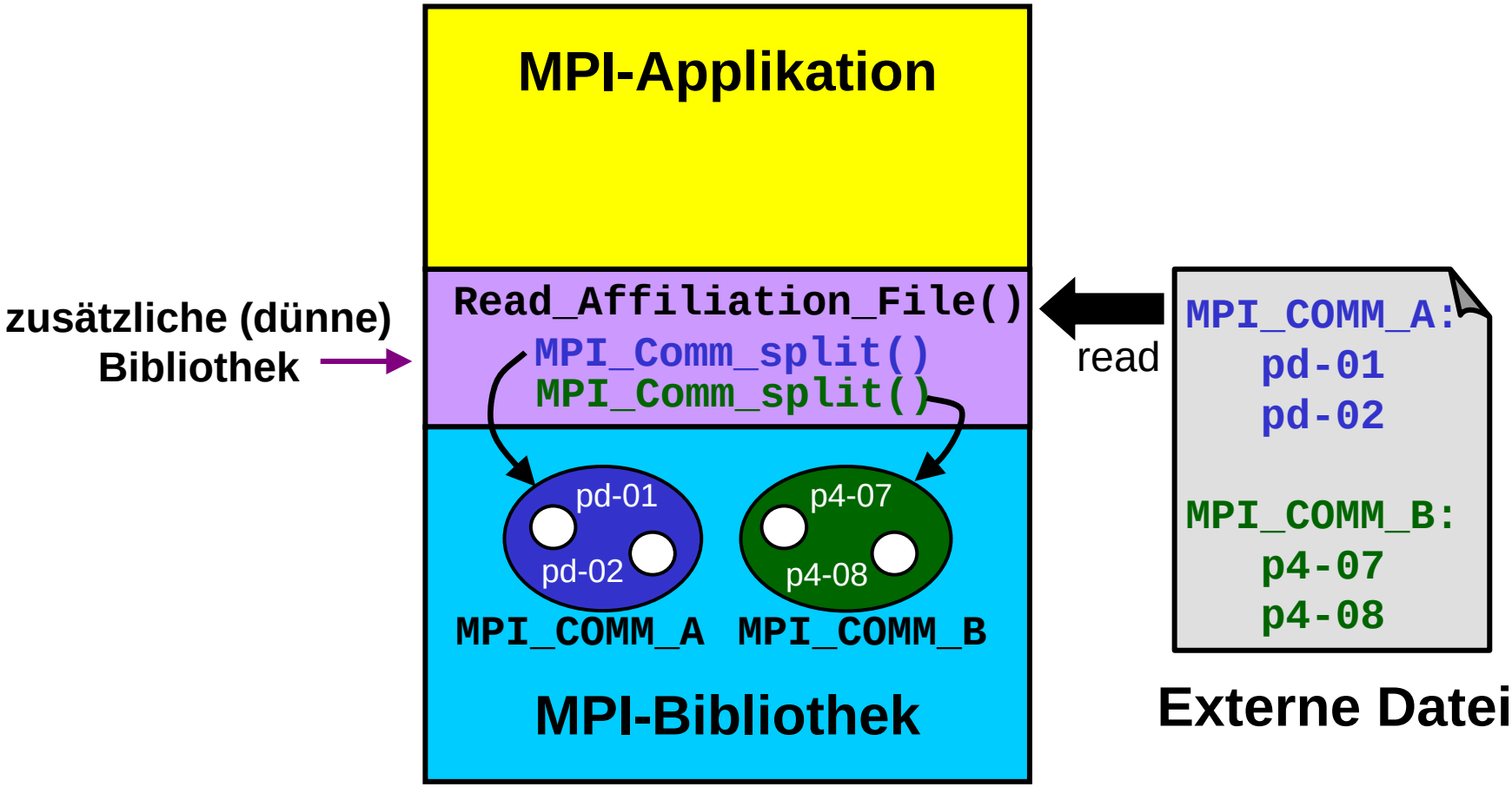
Extern definierter Kommunikatoren

- **Namensidentifizierung in der Applikation:**



Extern definierter Kommunikatoren

- **Auslagerung der Namensidentifizierung:**



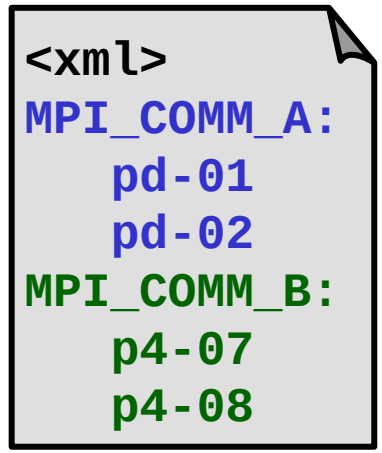
Extern definierter Kommunikatoren

- **Spezifizierung der Prozessornamen in XML:**

```
<processor> pd-01 </processor>
```

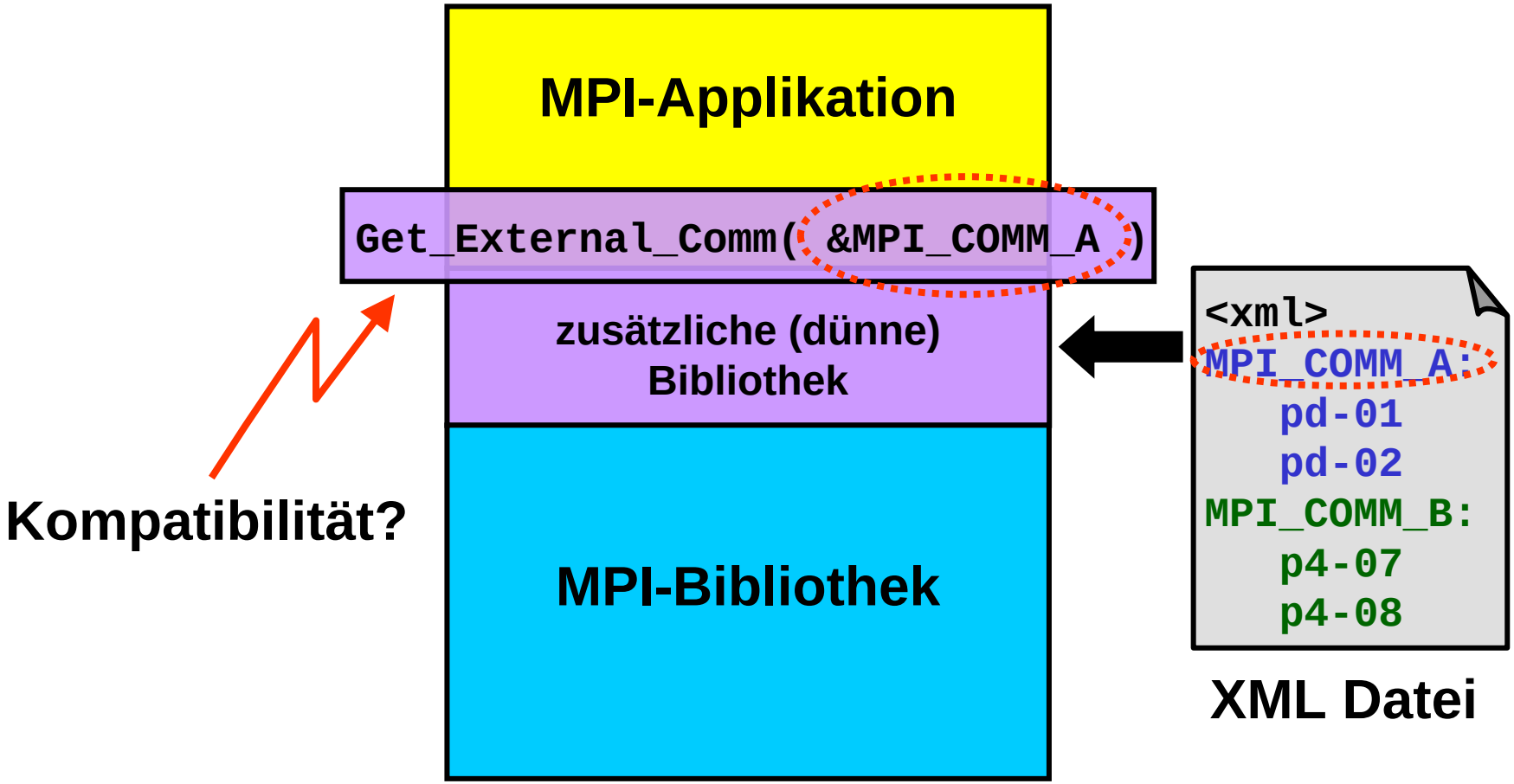
- **Spezifizierung der Kommunikatoren in XML:**

```
<comm name = “MPI_COMM_A“>
   <processor> pd-01 </processor>
   <processor> pd-02 </processor>
</comm>
<comm name = “MPI_COMM_B“>
   <processor> p4-07 </processor>
   <processor> p4-08 </processor>
</comm>
```



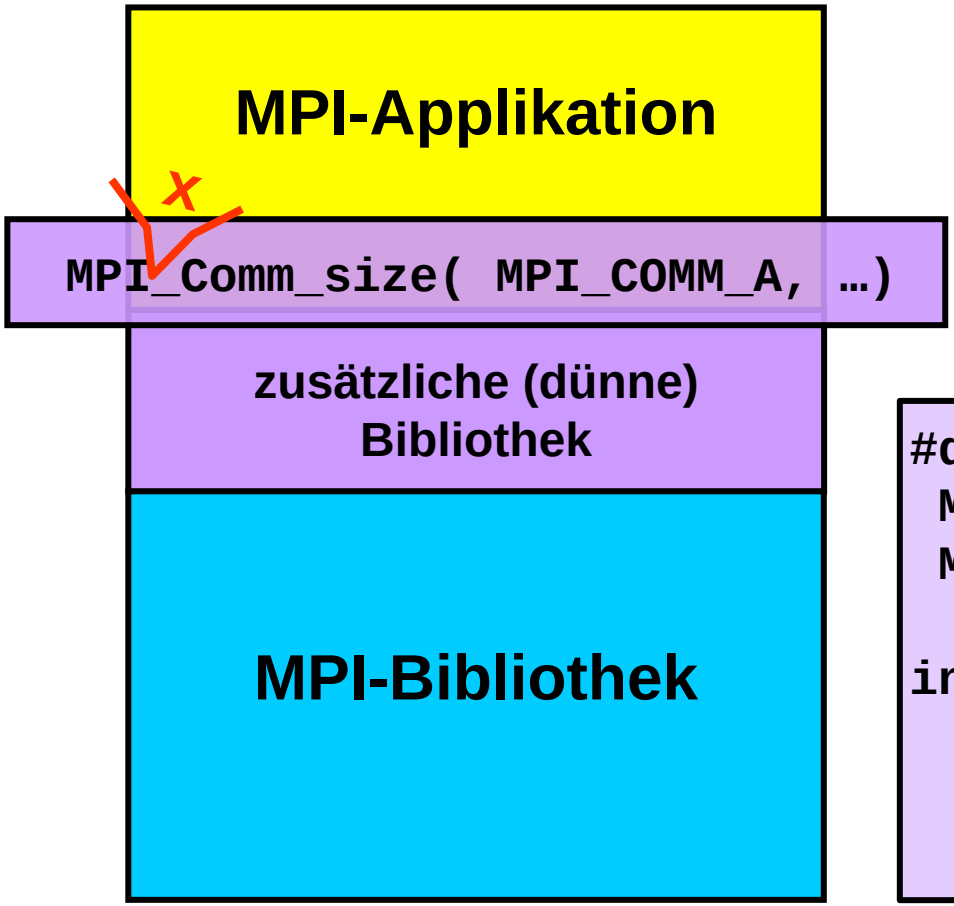
Extern definierter Kommunikatoren

- **Auslagerung der Namensidentifizierung:**



Extern definierter Kommunikatoren

- **Portable Integration:**



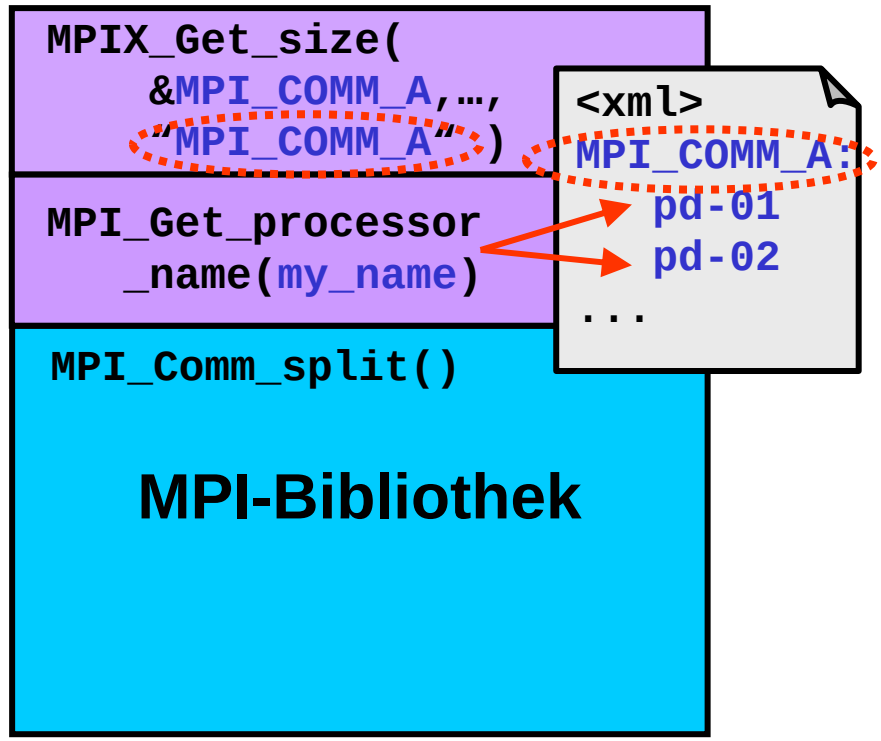
***Eigene* mpi.h Include-Datei**:

```
#define \
 MPI_Comm_size( a, b ) \
 MPIX_Comm_size( &a, b, #a )

int MPIX_Comm_size
  ( MPI_Comm *comm,
    int      *size,
    char     *name );
```

Extern definierter Kommunikatoren

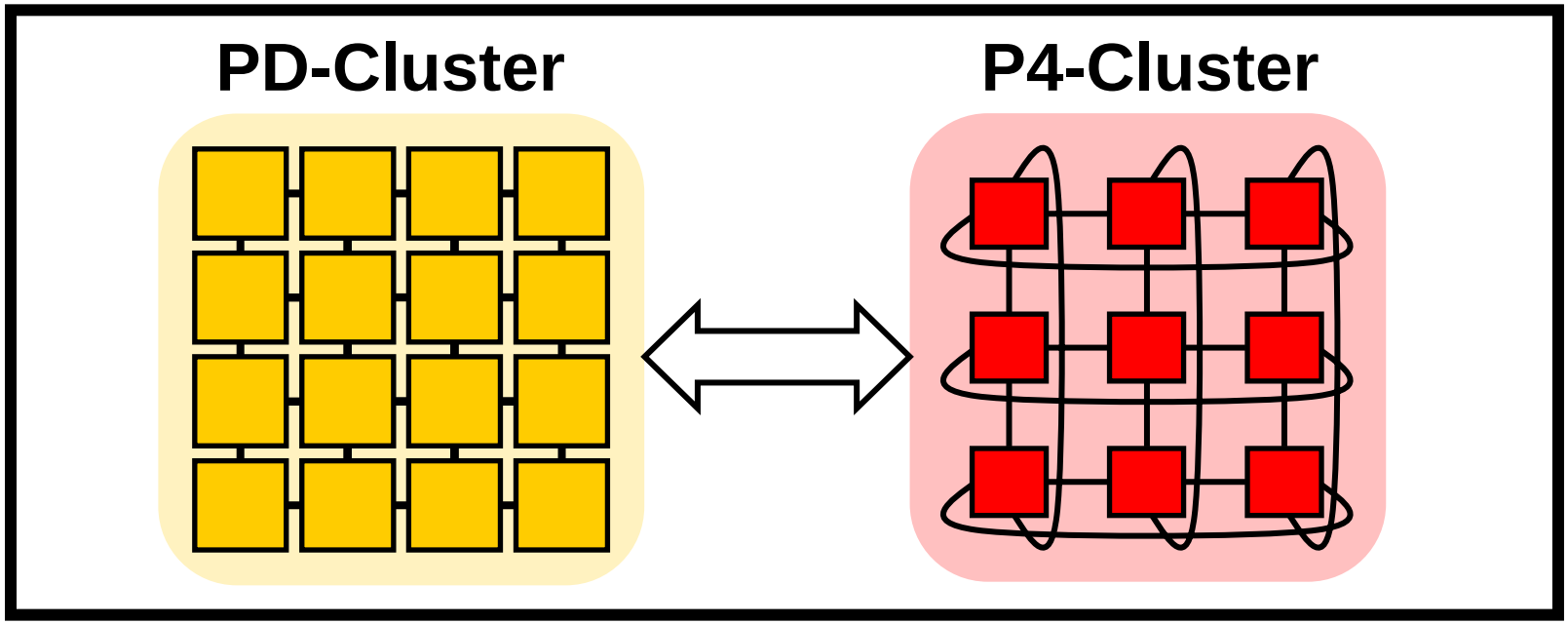
- **Portable Integration:**

```
MPI_Comm MPI_COMM_A
 =  MPI_COMM_NULL;

if( MPI_Get_size(
    MPI_COMM_A,…)
 == MPI_SUCCESS )
{
   /* Prozess gehoert
      zu COMM_A!
   */ ...
}
else
{
   /* Prozess gehoert
      NICHT zu COMM_A!
   */ ...
}
```



Anwendungsbeispiel

- **Gekoppelte Cluster-Systeme:**



Metacomputer

- **MPI für Metacomputer:**
  GridMPI, MPICH-G2, PACX-MPI, MetaMPICH, …

Anwendungsbeispiel

- **Meta-Konfiguration für MetaMPICH:**

```
METAHOST pd_cluster
{
   USER      = carsten;
   FRONTEND  = pd-00;
   EXECPATH  = ~/mp-mpich/examples/basic;
   EXECNAME  = cpi;
   MPIROOT   = ~/mp-mpich;
   ENVFILE   = ~/metahost_env;
   TYPE      = ch_smi;
   NODES     = pd-01,pd-02,pd-03,pd-04;
}
```

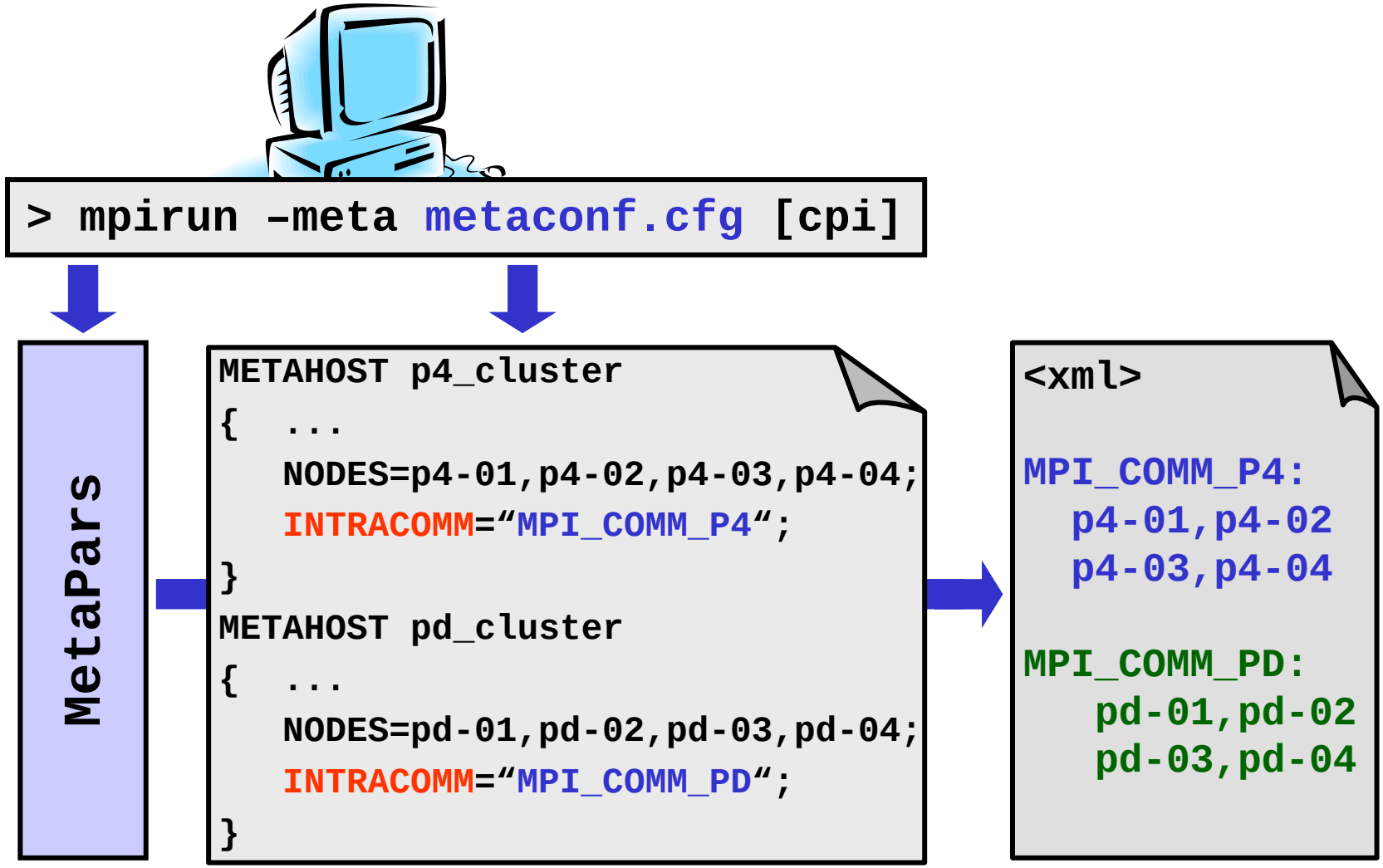
Anwendungsbeispiel

- **Meta-Konfiguration für MetaMPICH:**

```
METAHOST p4_cluster
{  ...
   NODES = p4-01, p4-02, p4-03, p4-04;
   INTRACOMM = “MPI_COMM_P4“;
}

METAHOST pd_cluster
{  ...
   NODES = pd-01, pd-02, pd-03, pd-04;
   INTRACOMM = “MPI_COMM_PD“;
}
```

Anwendungsbeispiel

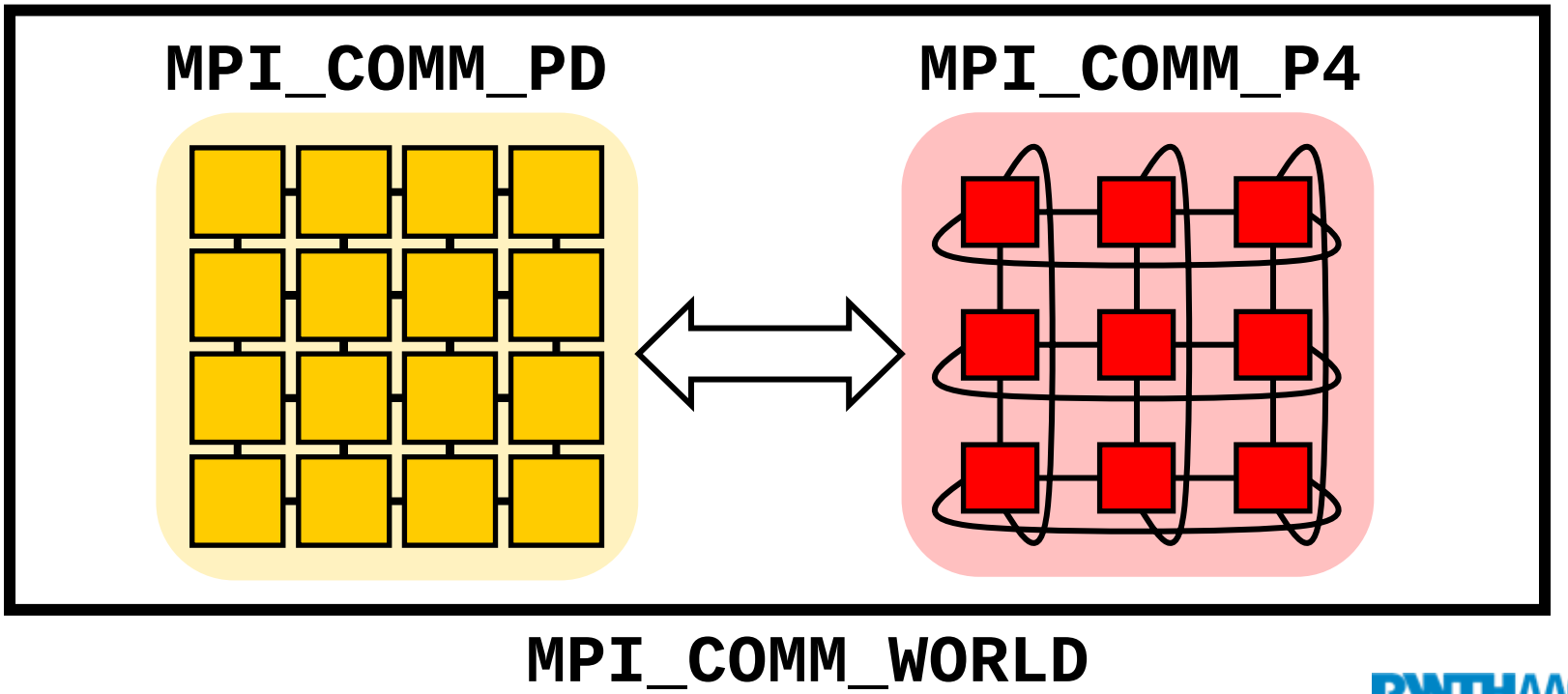
- **Laufzeitumgebung von MetaMPICH:**



Anwendungsbeispiel

- **Laufzeitumgebung von MetaMPICH:**

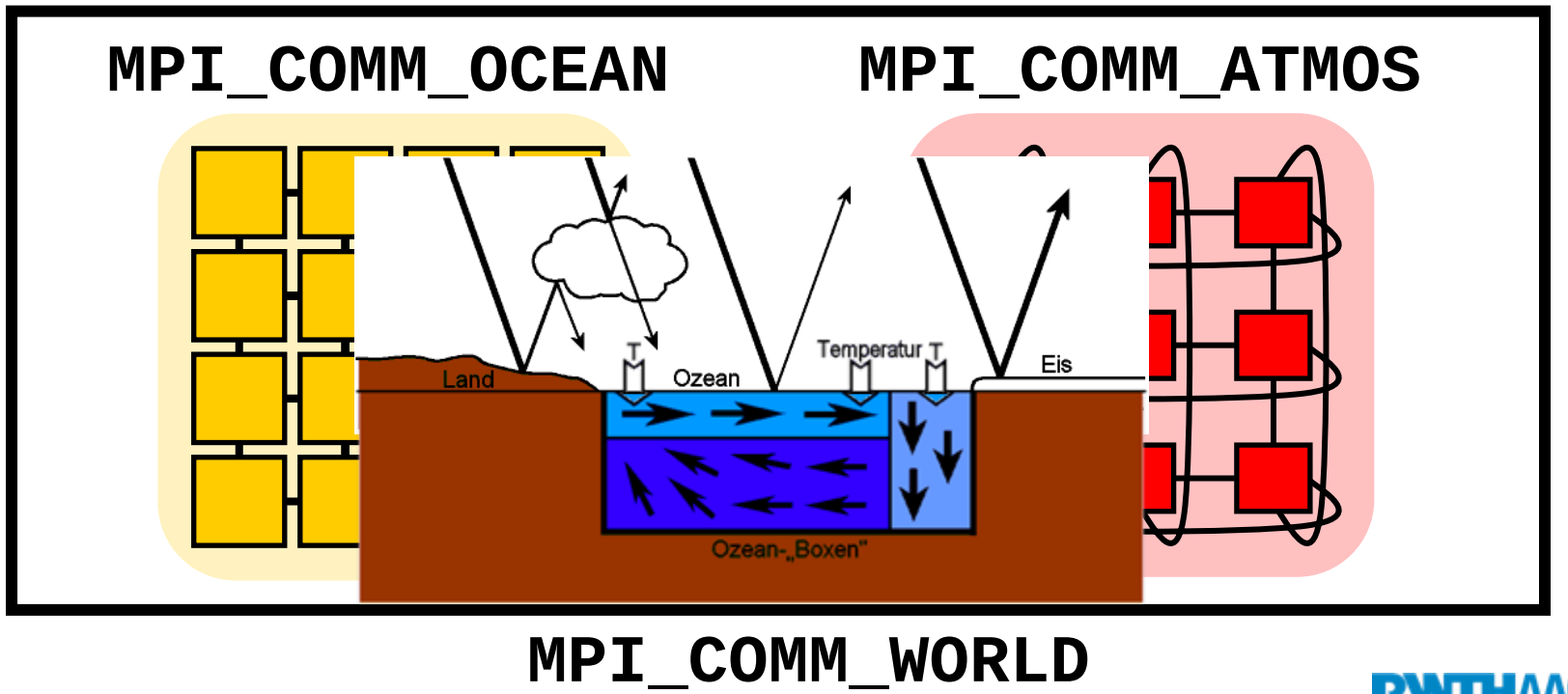
```
> mpirun –meta metaconf.cfg [cpi]
```



Anwendungsbeispiel

- **Laufzeitumgebung von MetaMPICH:**

```
> mpirun –meta metaconf.cfg [cpi]
```



Zusammenfassung und Ausblick

- Integration in *andere* MPI Implementierungen
- Integration in MP-Cluma
- Einsatz eines Topologie-Analysierers
- Zusammenspiel mit Grid-Middleware?
- Weitere mögliche Einsatzgebiete ???

Danke für Ihre Aufmerksamkeit !

Carsten Clauss
Lehrstuhl für Betriebssysteme (LfBS)
RWTH Aachen University, Germany

www.MP-MPICH.de

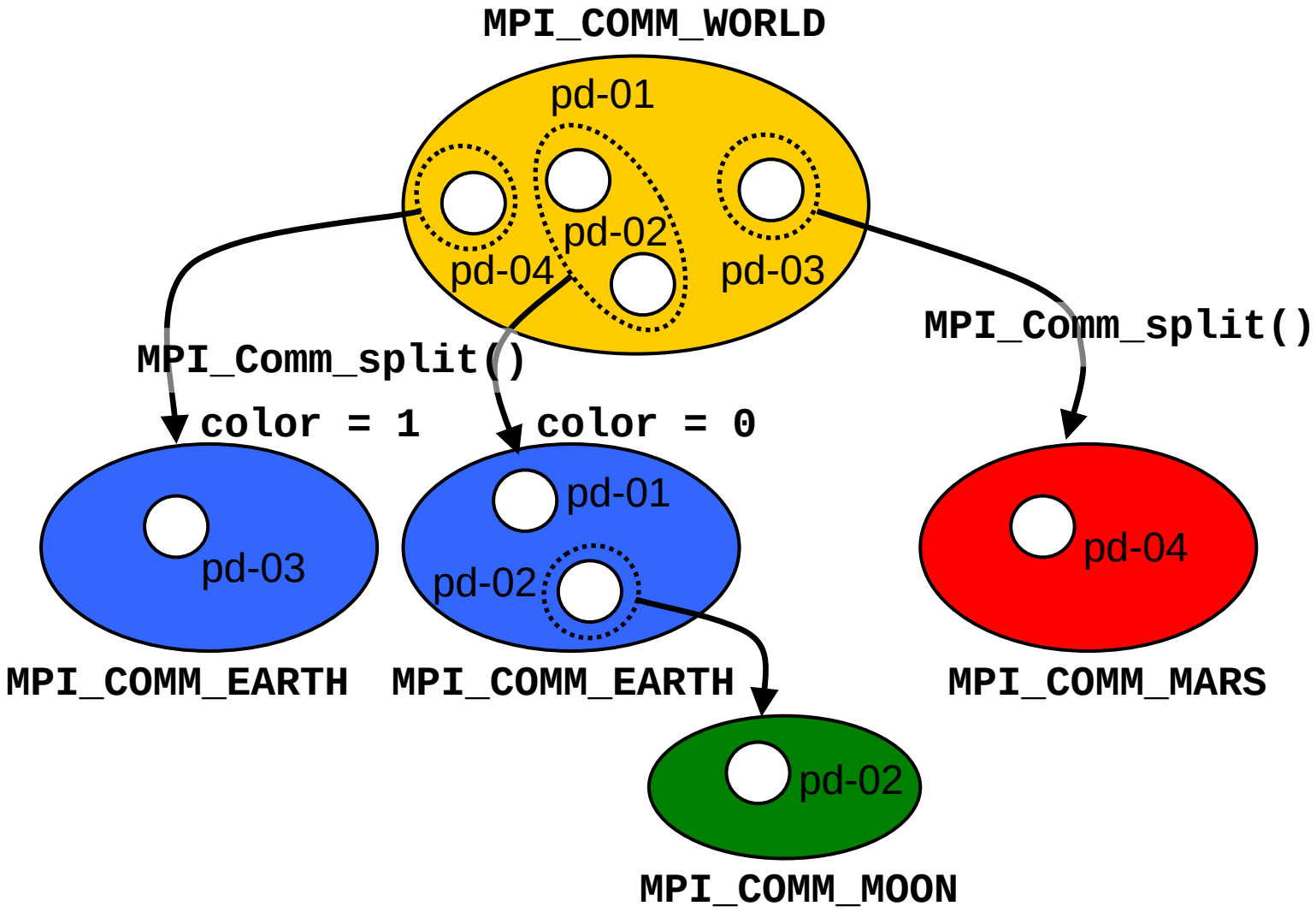
Extern definierter Kommunikatoren

- **Mehrstufige Definitionen:**

```
<comm name = “MPI_COMM_EARTH“>
   <processor> pd-01 </processor>
   <processor> pd-02 </processor>
   <comm name = “MPI_COMM_MOON“>
        <processor> pd-02 </processor>
   </comm>
</comm>
<comm name = “MPI_COMM_EARTH“>
   <processor> pd-03 </processor>
</comm>
<comm name = “MPI_COMM_MARS“>
   <processor> pd-04 </processor>
</comm>
```

Extern definierter Kommunikatoren

- **Mehrstufige Definitionen:**



Extern definierter Kommunikatoren

- **Inter-Kommunikatoren:**

```
<comm name = “MPI_COMM_EARTH“>
   <processor> pd-01 </processor> ...
</comm>
<comm name = “MPI_COMM_EARTH“>
   <processor> pd-03 </processor>
</comm>
<comm name = “MPI_COMM_MARS“>
   <processor> pd-04 </processor>
</comm>
<intercomm name = “MPI_COMM_INTER“>
   <first color=1> MPI_COMM_EARTH </first>
   <second> MPI_COMM_MARS </second>
</intercomm>
```

Extern definierter Kommunikatoren

- **Mehrstufige Definitionen:**