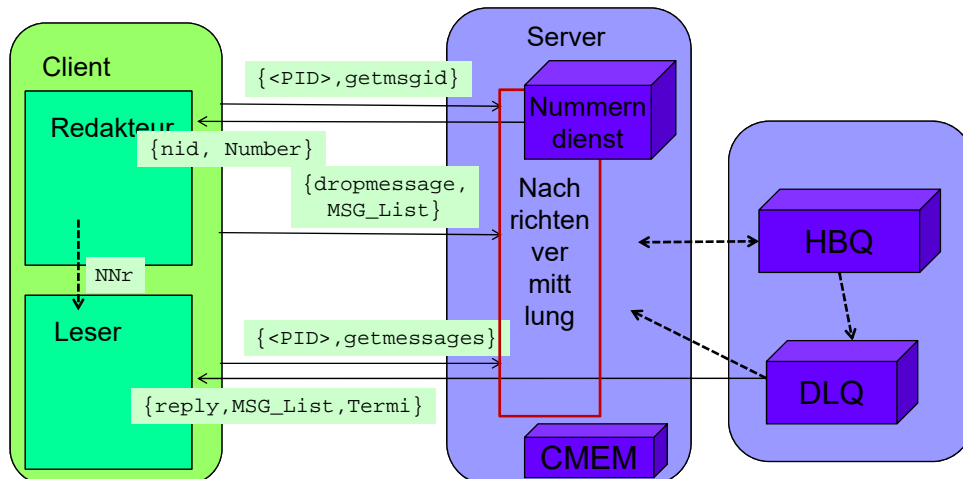


# Verteilte Systeme

## Aufgabe 1

1

### Nachrichtendienst



$MSG\_List := [NNr, Msg, TSciout, TShbqin, TSdlqin, TSdlqout]$

$[42, "Hallo Server", \{1418, 898760, 161000\}, \{1418, 898762, 161500\}, \dots]$

2

## Schnittstellen

```

/* Abfragen einer Nachricht */ wird von DLQ direkt beantwortet
Server ! {self(), getmessages},
receive {reply,[NNr,Msg,TSclientout,TShbqin,TSdlqin,TSdlqout],Terminated}

/* Senden einer Nachricht */ wird immer in HBQ gespeichert
Server ! {dropmessage,[INNr,Msg,TSclientout]},

/* Abfragen der eindeutigen Nachrichtennummer */
Server ! {self(),getmsgid}                                receive {nid,Number}

/* Initialisieren der HBQ */ entfernt auf einer anderen Node
HBQ ! {self(), {request,initHBQ}}                          receive {reply,ok}

/* Speichern einer Nachricht in der HBQ */
HBQ ! {self(), {request,pushHBQ,[NNr,Msg,TSclientout]}}    receive {reply,ok}

/* Abfrage einer Nachricht */
HBQ ! {self(), {request,deliverMSG,NNr,ToClient}}          receive {reply,SendNNr}

/* Terminierung der HBQ */
HBQ ! {self(), {request,dellHBQ}}                          receive {reply, ok}

```

3

## Schnittstellen

```

/* Initialisieren der DLQ */
initDLQ(Size,Datei): Integer X Atom → DQueue

/* Abfrage welche Nachrichtennummer in der DLQ gespeichert werden kann */
expectedNr(Queue) : DQueue → Integer

/* Speichern einer Nachricht in der DLQ */
push2DLQ([NNr,Msg,TSclientout,TShbqin],Queue,Datei) :
    MSG_list X DQueue X Atom → DQueue

/* Ausliefern einer Nachricht an einen Leser-Client */
deliverMSG(MSGNr,ClientPID,Queue,Datei):
    Integer X PID X DQueue X Atom → Integer

/* Nachrichtenformat */
/* minimal 3 Elemente, pro Station kommt eins hinzu; maximal 6 Elemente */
MSG_List := [NNr,Msg,TSclientout,TShbqin,TSdlqin,TSdlqout]:
    [Integer X String X 3-Tupel X 3-Tupel X 3-Tupel X 3-Tupel]

```

4

## Schnittstellen

```

/* Initialisieren des CMEM */
initCMEM(RemTime,Datei): Integer X Atom → CMem

/* Speichern/Aktualisieren eines Clients in dem CMEM */
updateClient(CMEM,ClientID,NNr,Datei): Cmem X PID X Integer X Atom → CMem

/* Abfrage welche Nachrichtennummer der Client als nächstes erhalten darf */
getClientNNr(CMEM,ClientID) : Cmem X PID → Integer
  
```

Die HBQ ist komplett in einer Datei hbq.erl zu halten.

Die DLQ ist komplett in einer Datei dlq.erl zu halten.

Der CMEM ist komplett in einer Datei cmem.erl zu halten.

util.erl und vsutil.erl kann genutzt werden. Weitere Dateien sind bei den ADTs nicht zulässig (wegen Austauschbarkeit).

„Übergaben“ von komplexeren Datenstrukturen an Erlang OTP ist untersagt:  
Lediglich die Basis-Strukturen Liste (lists) und Tupel (tuple) dürfen eingesetzt werden.  
Algorithmen auf diesen Basisstrukturen müssen selbst implementiert werden.

5

## Durchzuführende Tests

Test 1: Ihr Server, vorgegebener Client  
 Test 2: Vorgegebener Server, Ihr Client  
 Test 3: Kombinationen der ADTs

- HB-DLQ@Qigong42.log
- Server@Qigong42.log
- Client\_0rudel@Qigong42.log
- Client\_1rudel@Qigong42.log
- Client\_2rudel@Qigong42.log
- Client\_3rudel@Qigong42.log
- Client\_4rudel@Qigong42.log
- Client\_5rudel@Qigong42.log

6