Universität Duisburg-Essen, Institut für Experimentelle Mathematik

RSPLIB
--
Eine Open Source Implementation von Reliable Server Pooling

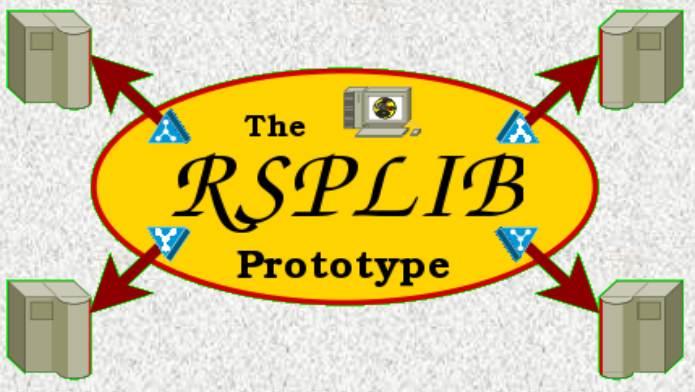
Dipl.-Inform. Thomas Dreibholz
Institut für Experimentelle Mathematik
Universität Duisburg-Essen
dreibh@exp-math.uni-essen.de
http://www.exp-math.uni-essen.de/~dreibh

UNIVERSITÄT
DUISBURG
ESSEN

Inhalt

UNIVERSITÄT DUISBURG ESSEN

- Motivation
- Was ist Reliable Server Pooling?
  - Einführung
  - Session-Aufbau und Failover
  - Demo-Vorführung
- Unsere Implementation
  - Designideen
  - Aufbau der Komponenten
  - RSerPool-API
- Unsere Aktivitäten im Bereich Reliable Server Pooling
- Zusammenfassung

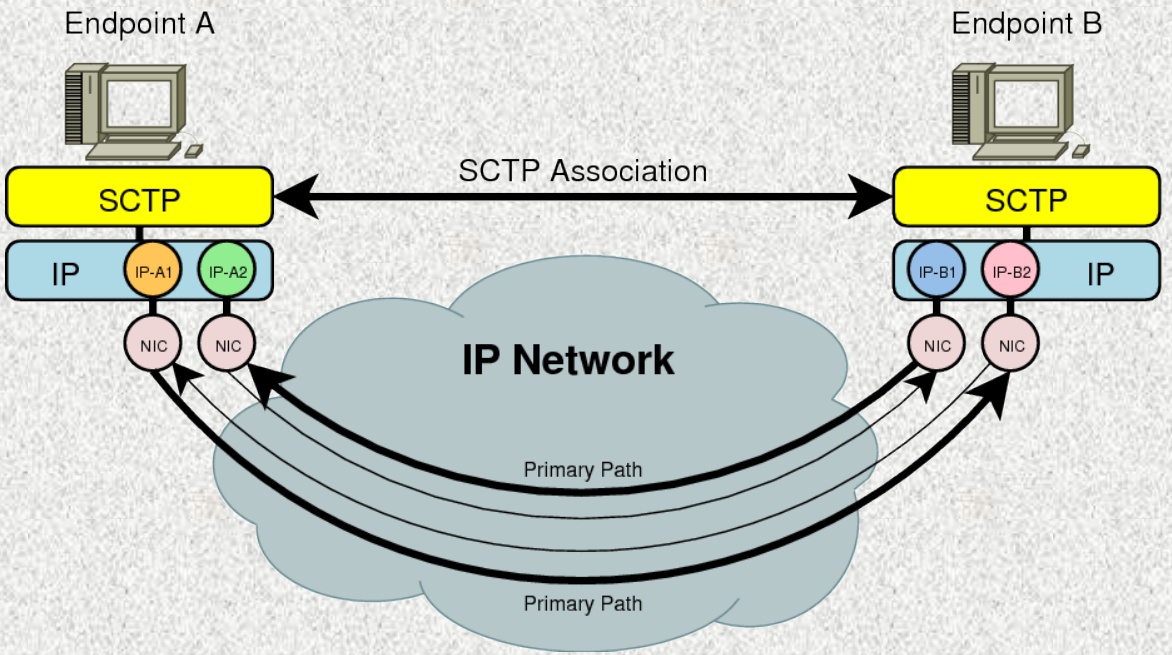


Thomas Dreibholz's Reliable Server Pooling Page
http://tdrwww.exp-math.uni-essen.de/dreibholz/rserpool/

Motivation (I)

UNIVERSITÄT DUISBURG ESSEN

- Ursprüngliche Motivation:
  - Telefonsignalisierung (SS7-Protokoll) über IP
  - Hohe Anforderungen an Verfügbarkeit
- Das Stream Control Transmission Protocol (SCTP) [RFC2960]
  - „TCP Next Generation“
  - **Multi-Homing**
  - Multi-Streaming
  - Message-Framing
  - Schutz vor DoS-Angriffen
  - 4-Wege-Handshake
  - Verification Tag
  - Add-IP: dynamische Adreßrekonfiguration



- SCTP schützt vor einer Vielzahl von Netzproblemen, aber ...
- ... nicht vor einem **Serverausfall**

⇒ Konzept für **Server-Redundanz** ist **notwendig**

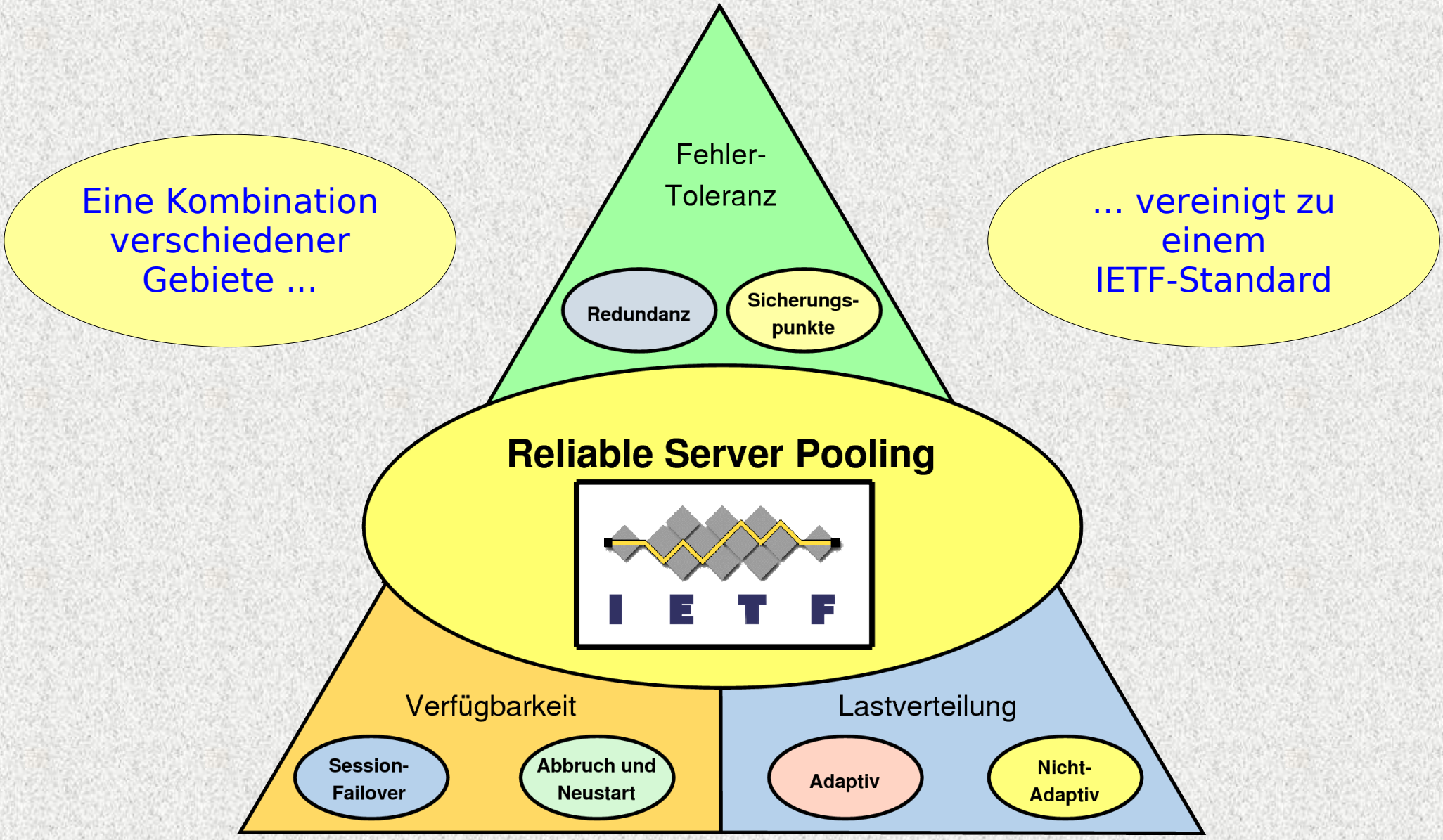
Motivation (II)

UNIVERSITÄT DUISBURG ESSEN

- Reliable Server Pooling (RSerPool)
  - Standardisierung in der IETF RSerPool WG
  - RSerPool ist ein **Framework** zur **Pool- und Sitzungsverwaltung**
- Anforderungen an RSerPool:
  - **Lightweight** (z.B. auch auf Embedded Devices nutzbar)
  - **Echtzeit** (schneller Failover)
  - **Skalierbar** (z.B. große (Firmen-)Netzwerke, nicht auf gesamtes Internet)
  - **Erweiterbar** (z.B. durch neue Serverauswahlregeln)
  - **Einfach** (automatische Konfiguration)
- Anwendungsgebiete für RSerPool:
  - Telefonsignalisierung (SS7) über IP, Voice over IP (VoIP)
  - **Distributed Computing**
  - Mobility-Management in Verbindung mit SCTP und Add-IP
  - IP Flow Information Export (IPFIX)
  - **Load Balancing**
  - Battlefield Networks, e-Commerce, m-Commerce, ...

Verwandte Themengebiete

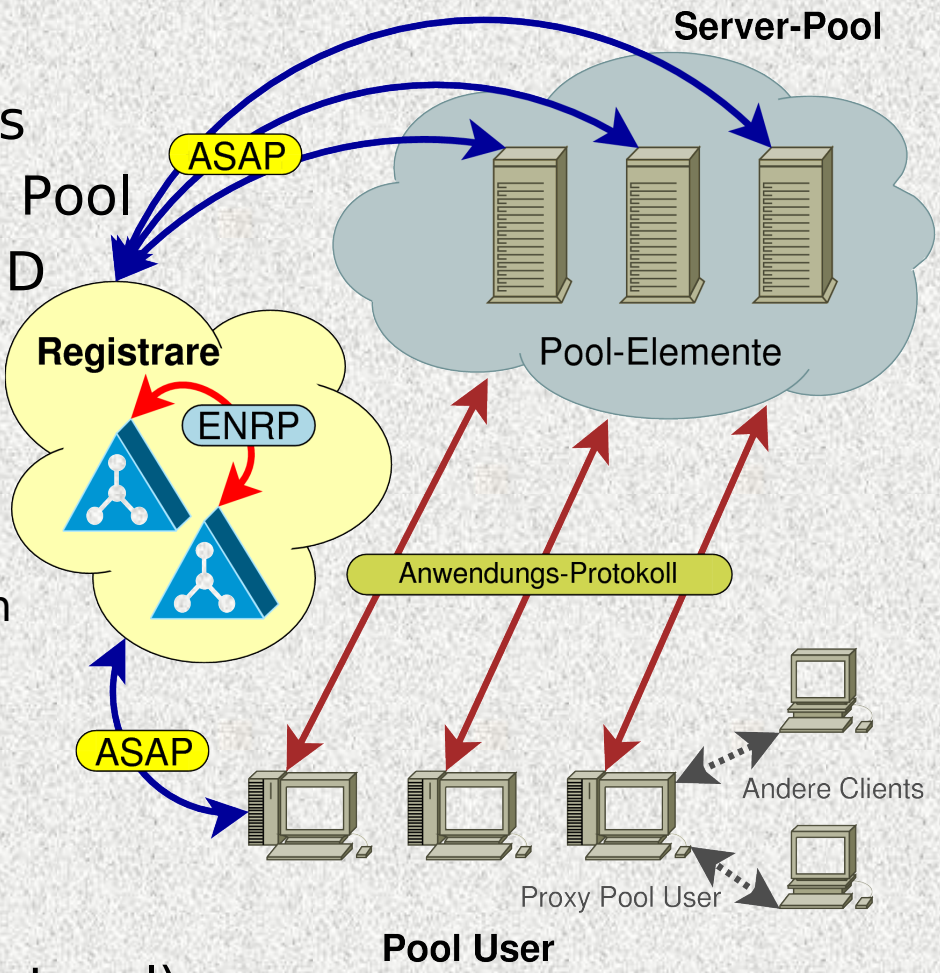
UNIVERSITÄT DUISBURG ESSEN



Reliable Server Pooling (RSerPool)

UNIVERSITÄT DUISBURG ESSEN

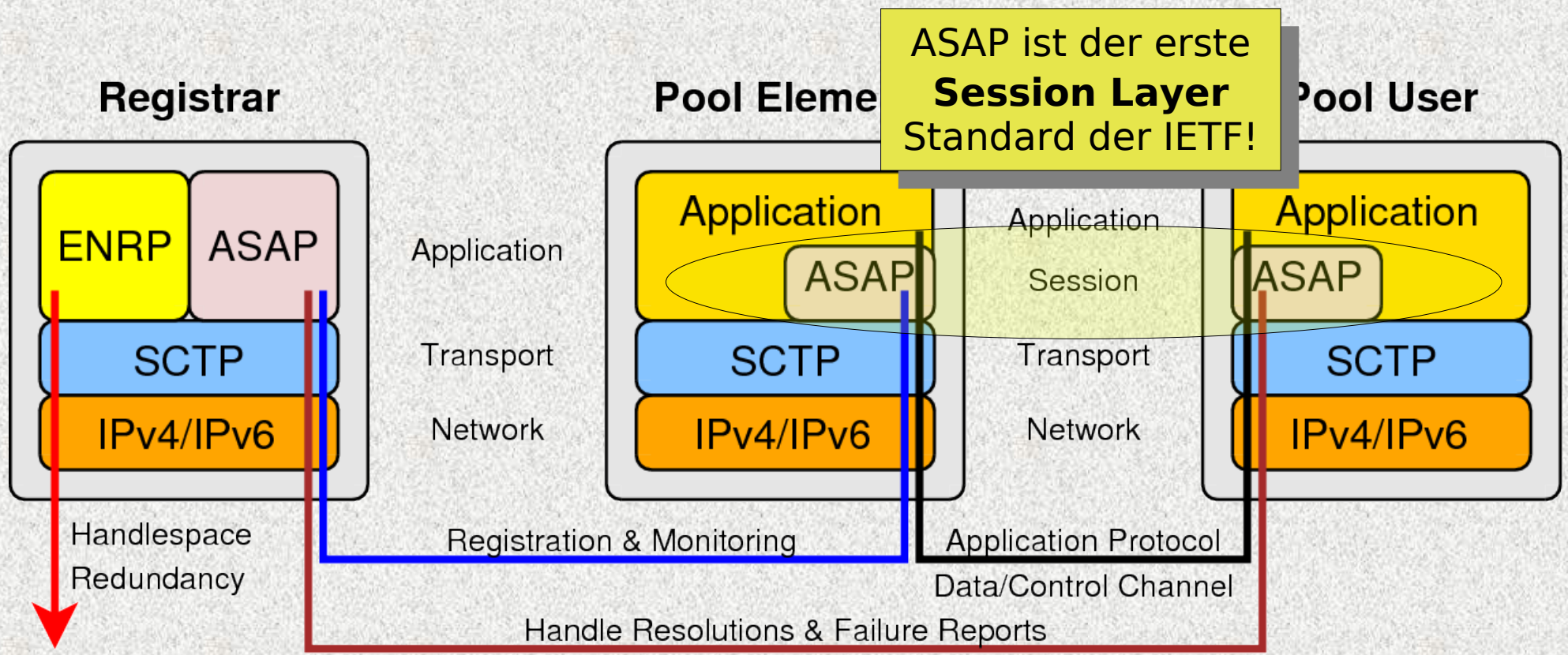
- Terminologie:
  - **Pool Element (PE)**: Server
  - Pool: Menge von PEs
  - PE ID: ID eines PE im Pool
  - Pool Handle: Eindeut. Pool-ID
  - Handlespace: Pool-Menge
  - Pool **Registrar (PR)**
  - **Pool User (PU)**: Client
  - Unterstützung für bestehende Anwendungen
    - Proxy Pool User (PPU)
    - Proxy Pool Element (PPE)



- Protokolle:
  - **ASAP** (Aggregate Server Access Protocol)
  - **ENRP** (Endpoint Handlespace Redundancy Protocol)

Der RSerPool-Protokollstack

UNIVERSITÄT DUISBURG ESSEN



Protokolle bei Reliable Server Pooling

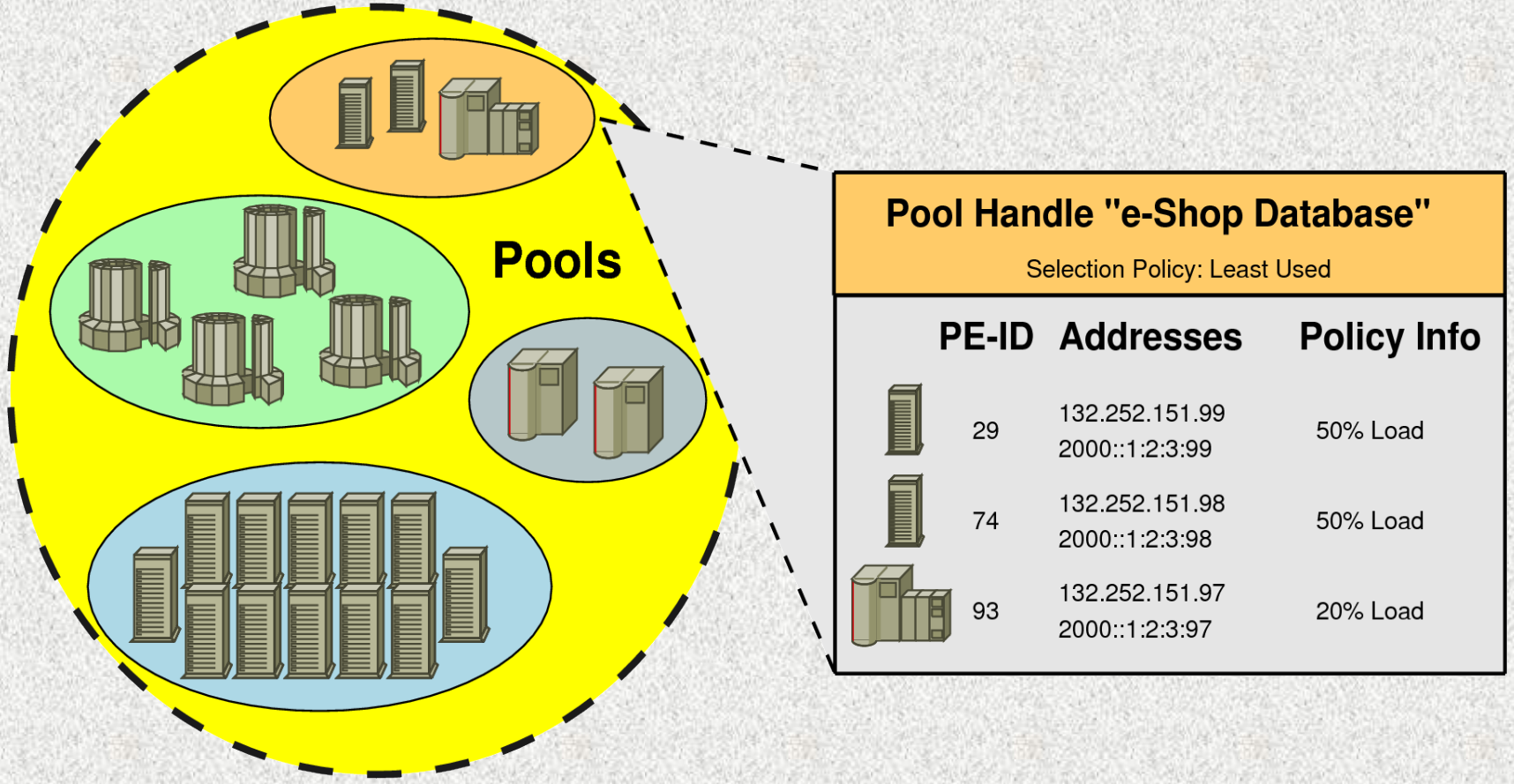
ASAP (Aggregate Server Access Protocol)

ENRP (Endpoint Handlespace Redundancy Protocol)

Der Handlespace

UNIVERSITÄT DUISBURG ESSEN

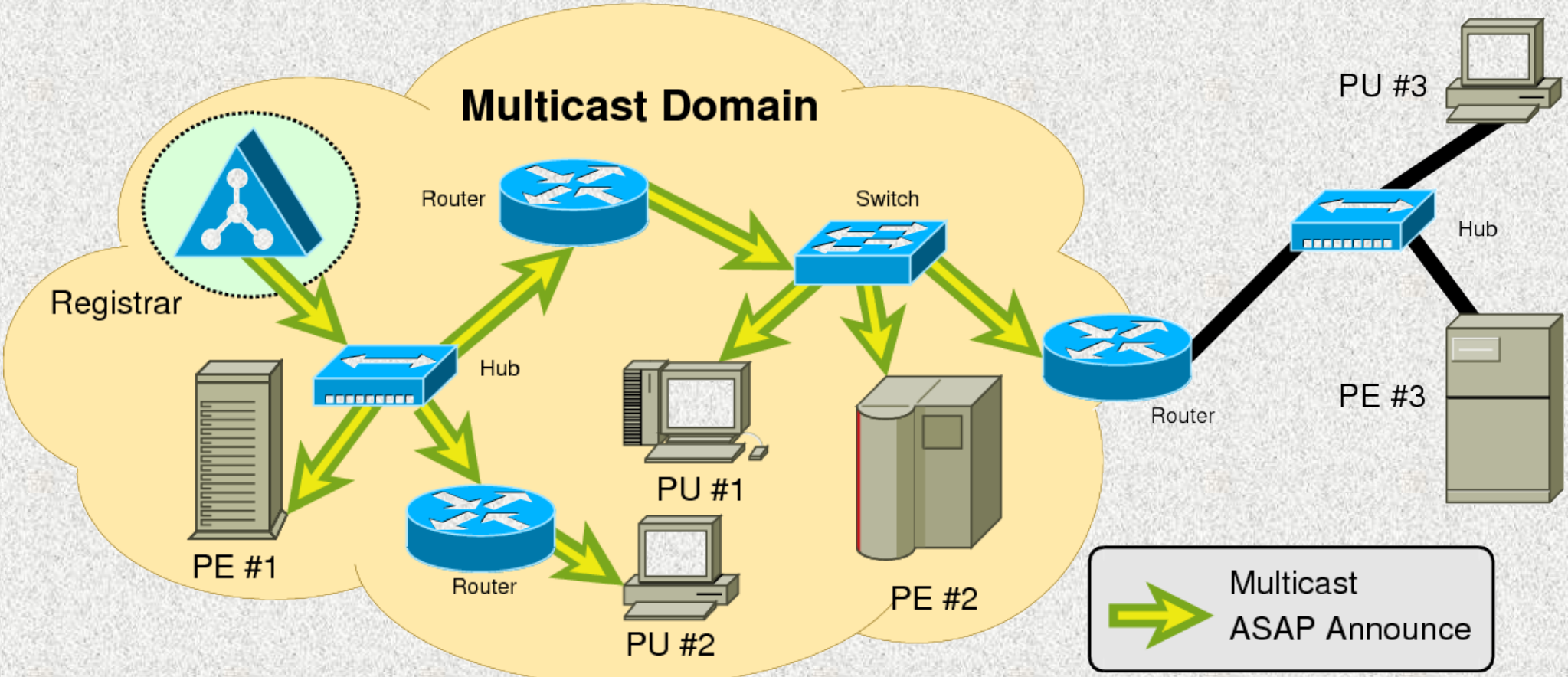
- Menge nicht-leerer Pools
- “flach”, d.h. keine Hierarchie für PHs



Automatische Konfiguration mittels Registrar-Announces

UNIVERSITÄT DUISBURG ESSEN

- Transport der Announces
  als UDP-Pakete via IP-Multicast

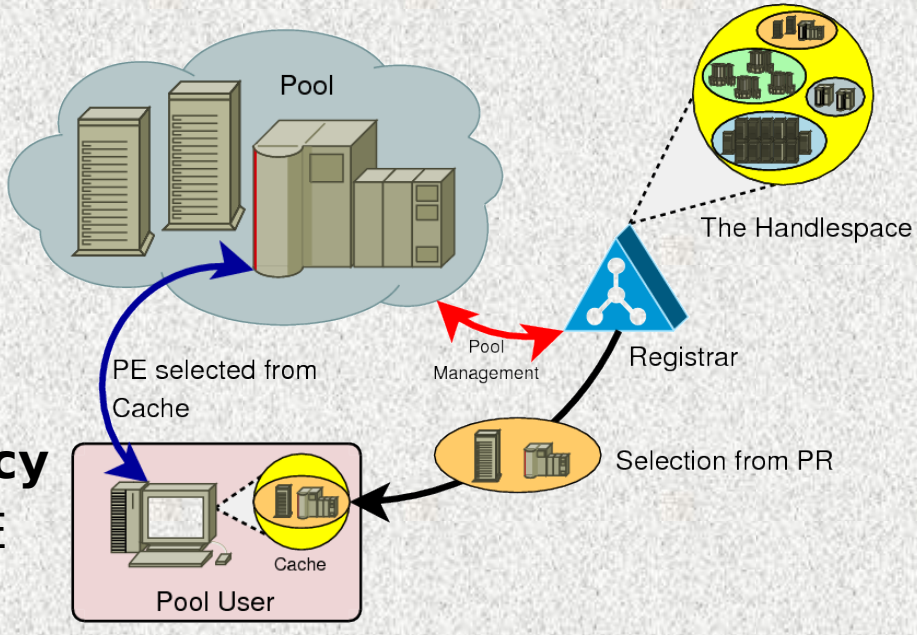


Sessionaufbau und Failover

UNIVERSITÄT DUISBURG ESSEN

- **Sessionaufbau** (PU möchte den Dienst eines Pools nutzen):
  - PU fragt bei beliebigem PR nach Auflösung von PH in Liste von PEs
  - **PR wählt** Menge von PEs
    - **nach** Auswahlregel (**Policy**)
    - z.B. Round Robin oder per Zufall
  - PU schreibt Liste in lokalen Cache
  - **PU wählt** ein PE aus Cache **nach Policy** Applikationsverbindung zu diesem PE
  - Weitere Anfragen → direkt aus **Cache**

Selektionen ① ②

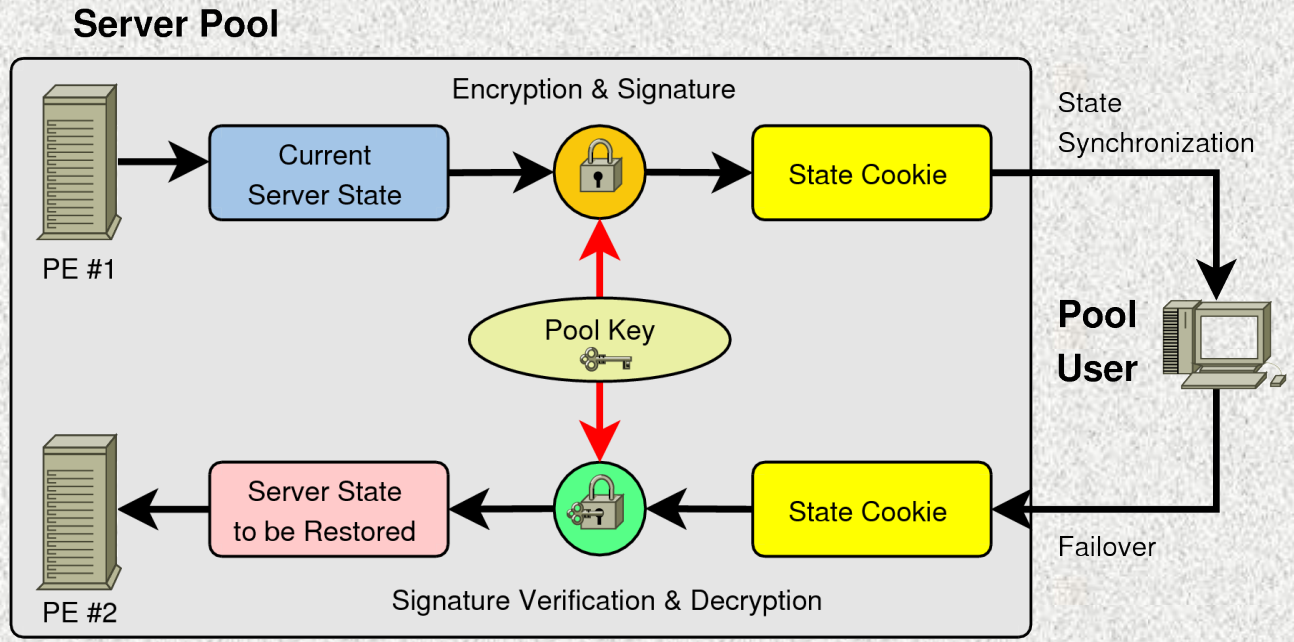


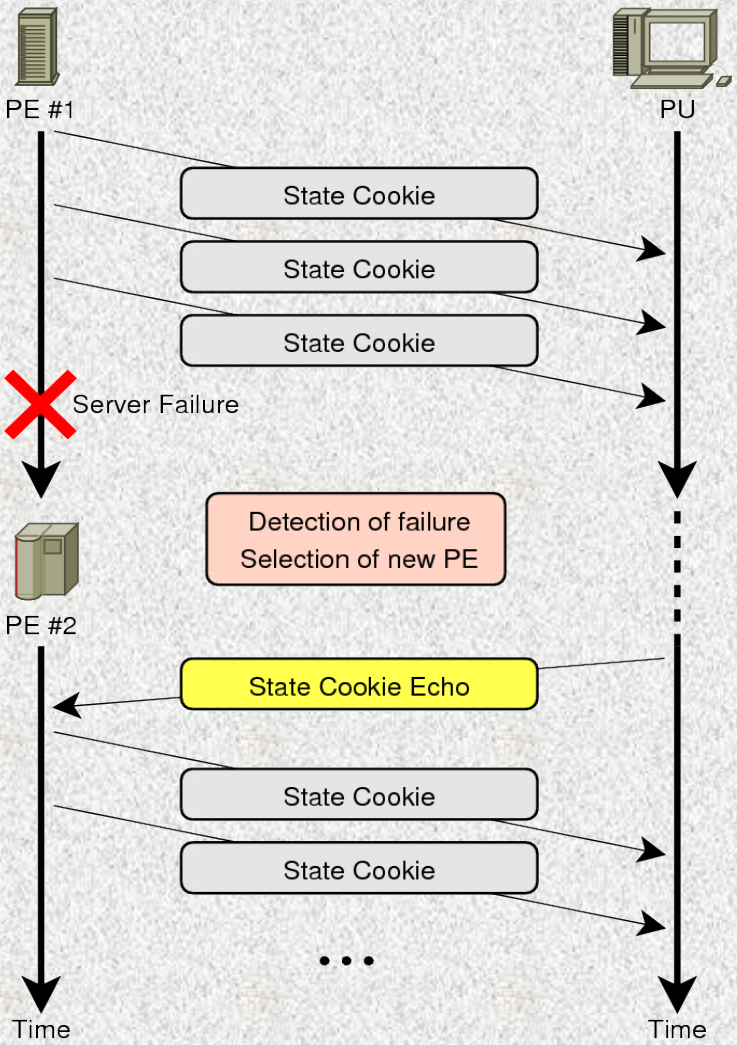
- **Failover** (PU behandelt PE-Ausfall):
  - PU sollte PR über den Ausfall des PEs **benachrichtigen** (PR kann ggf. das betreffende PE aus dem Handlespace entfernen)
  - PU **wählt** ein neues PE (aus Cache ②; mit Anfrage bei PR ① ②)
  - PU führt applikations-spezifische **Failover-Prozedur** durch
  - Beispiel: Datei-Download
    - neuem PE den Dateinamen und letzte Position mitteilen

Failover mit Client-Based State Sharing

UNIVERSITÄT DUISBURG ESSEN

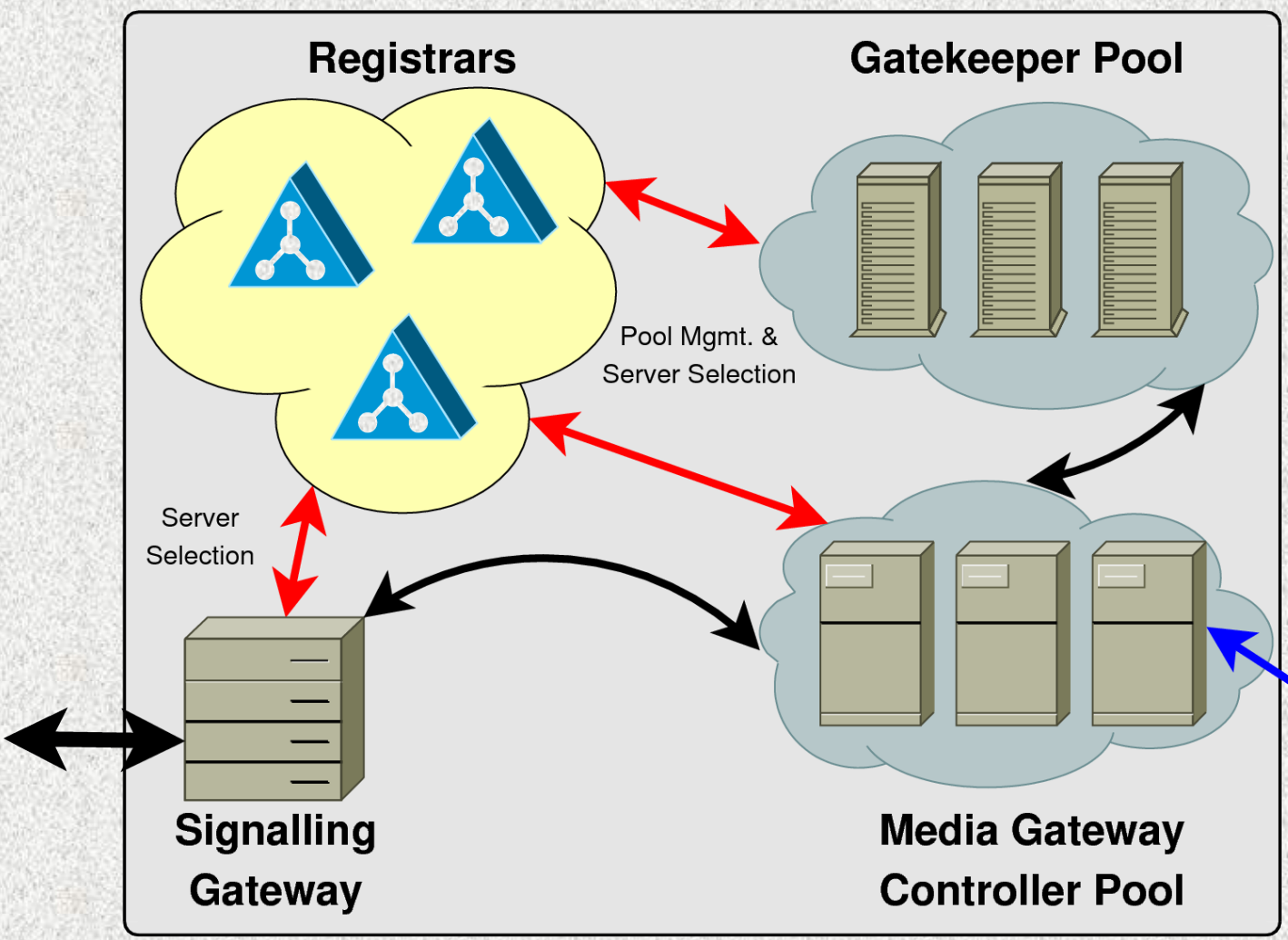
- Für Failover benötigt:
  - Neues PE muß den Session-Zustand des alten PEs wieder herstellen
- Einfache Lösung für viele Anwendungen:
  - Benutzung von State Cookies [LCN2002]
  - Ist jetzt Teil des ASAP-Protokolls





Anwendungsszenario für RSerPool: Klassische Telekommunikation

UNIVERSITÄT DUISBURG ESSEN



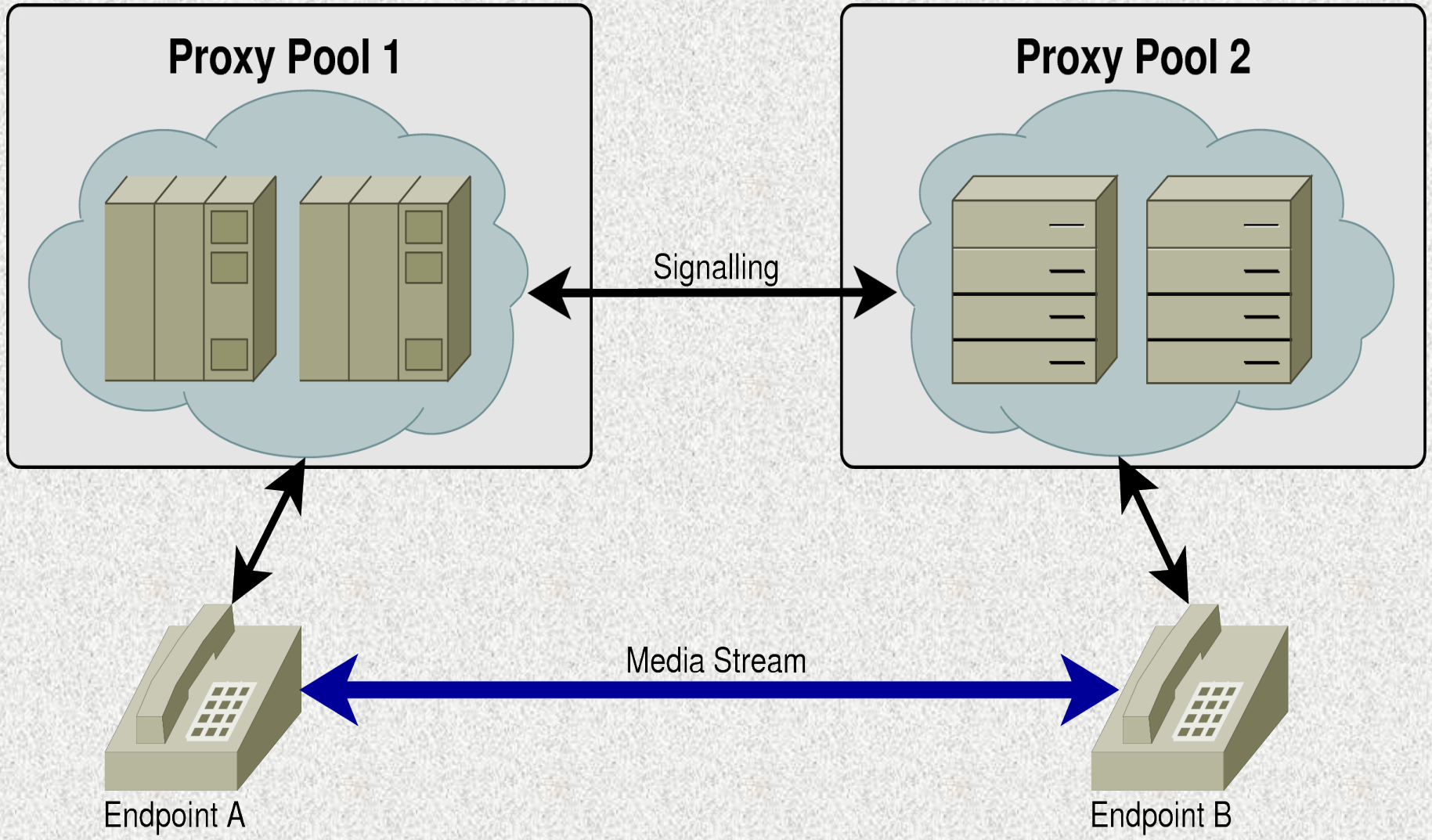
RSerPool-Eigenschaften:

- Lightweight
- Echtzeit
- Skalierbar
- Erweiterbar
- Einfach

Beschrieben in [draft-ietf-rserpool-arch-12.txt]

Anwendungsszenario für RSerPool: Voice over IP mit SIP

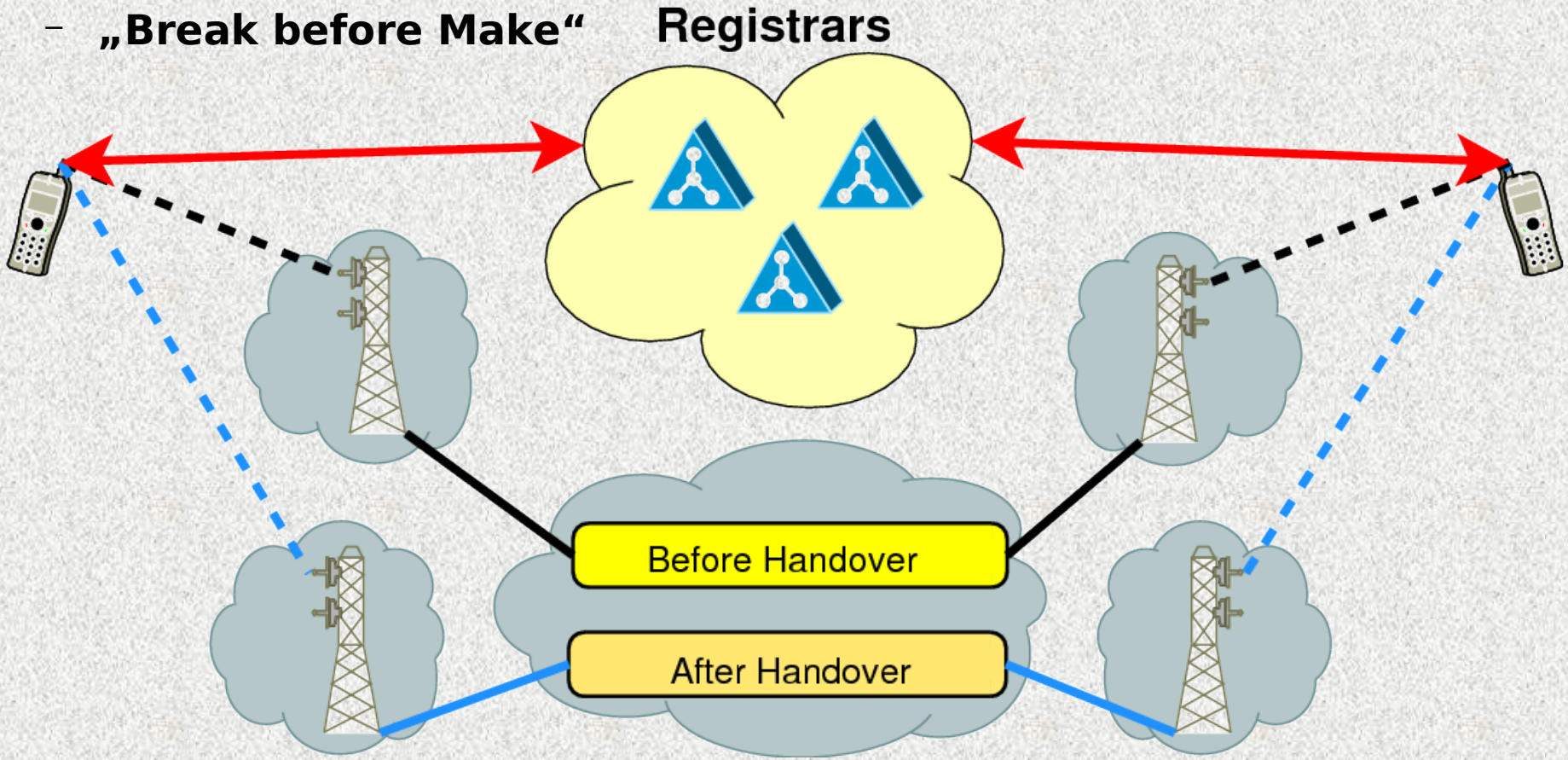
UNIVERSITÄT DUISBURG ESSEN



Anwendungsszenario für RSerPool: Mobilität mit SCTP und Add-IP

UNIVERSITÄT DUISBURG ESSEN

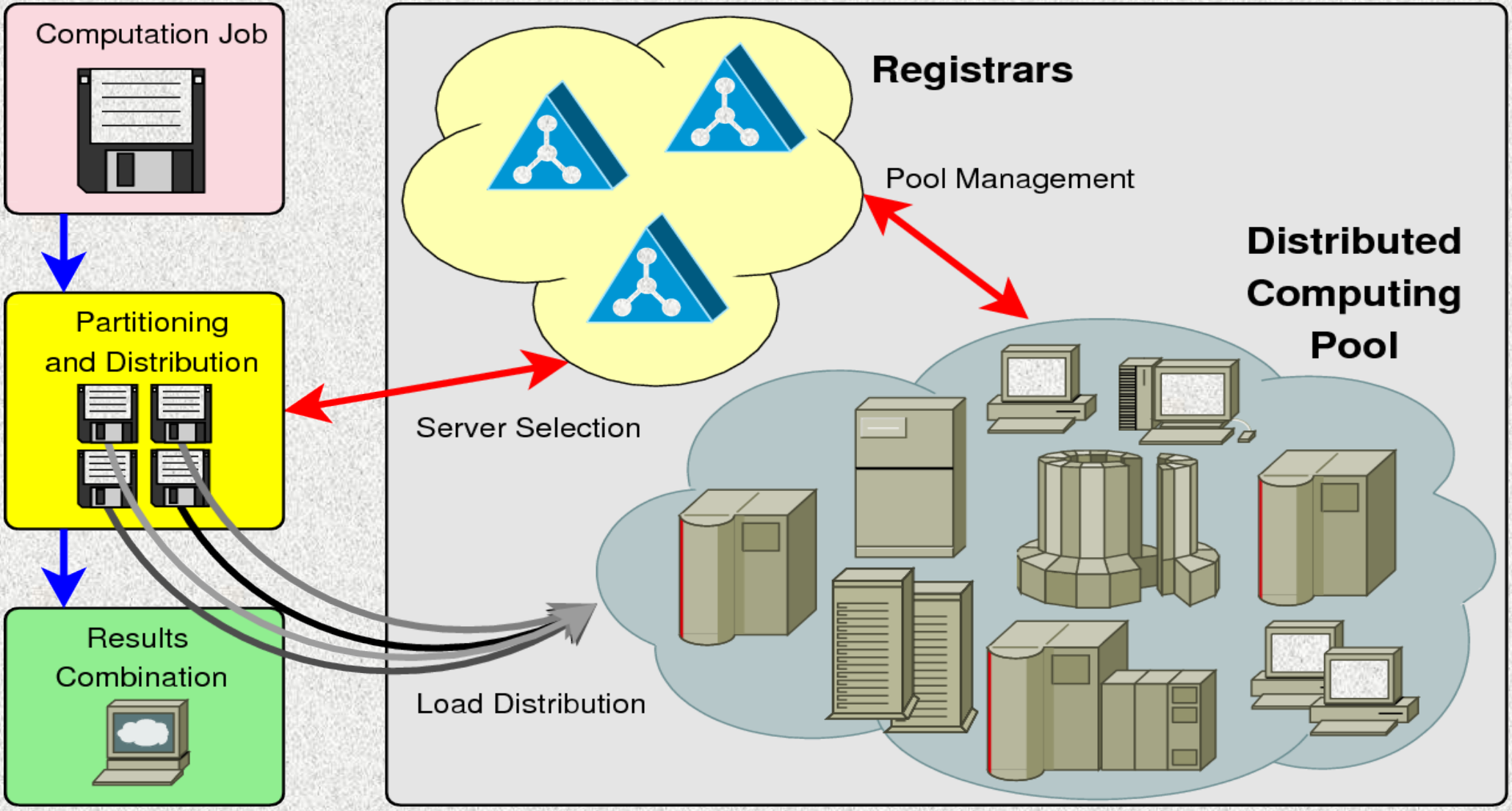
- Dynamic Address Reconfiguration Extension für SCTP (“Add-IP”)
- Zwei Handover-Szenarien:
  - „Make before Break“
  - **„Break before Make“**



Beschrieben in [LCN2003]

Anwendungsszenario für RSerPool: Real-Time Distributed Computing

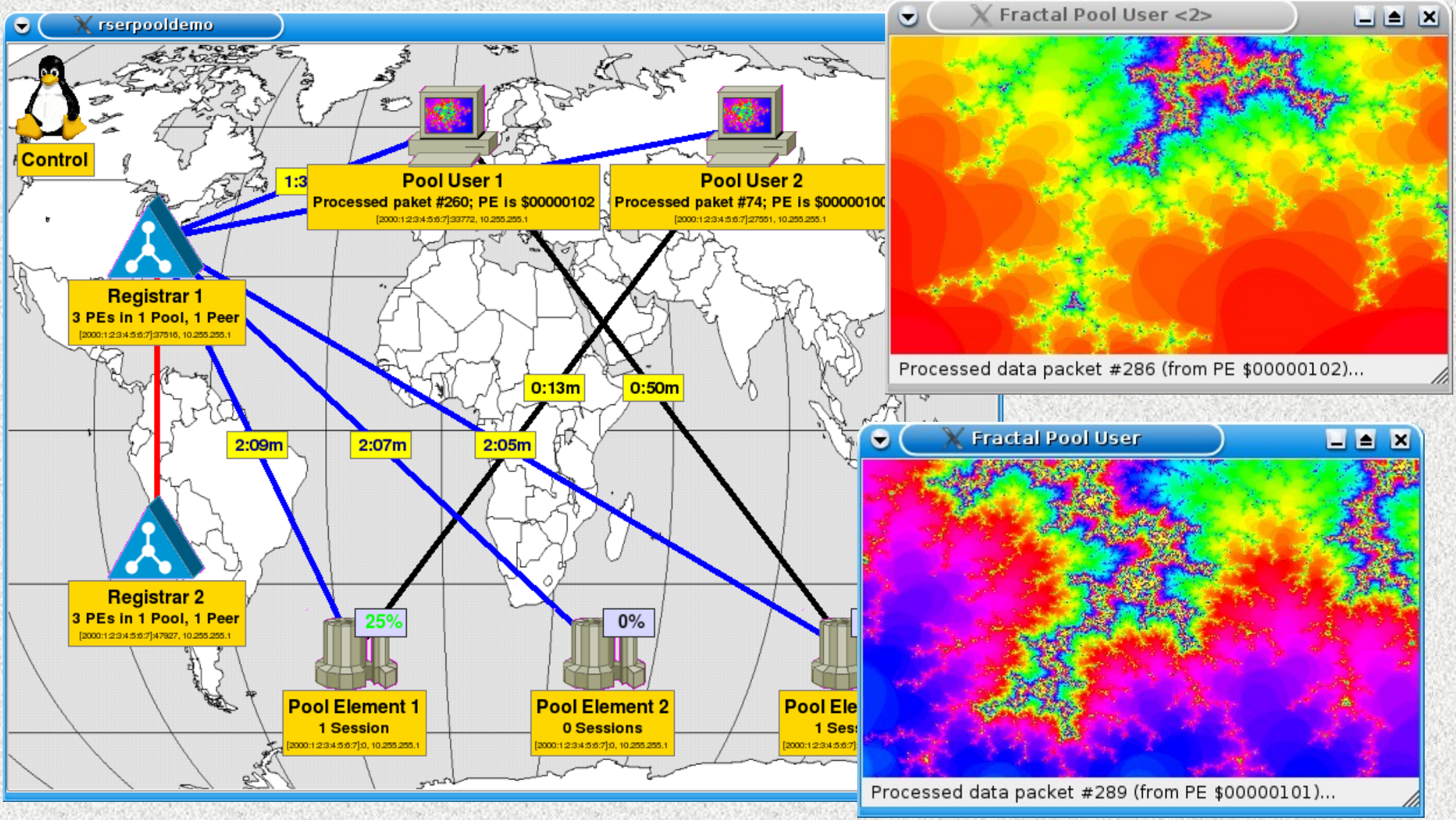
UNIVERSITÄT DUISBURG ESSEN



Beschrieben in [draft-dreibholz-rserpool-distcomp-01.txt]

RSerPool Demo!

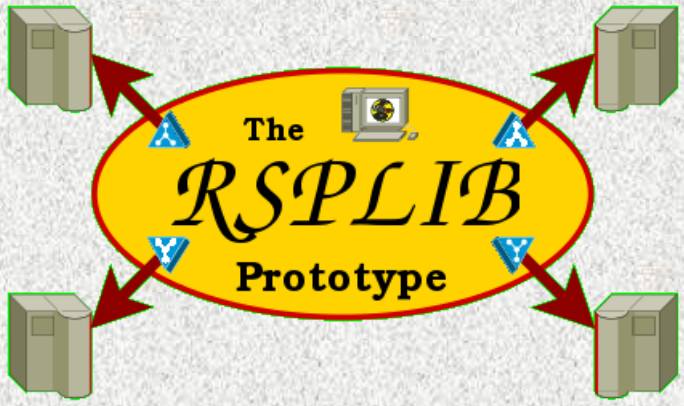
UNIVERSITÄT
DUISBURG
ESSEN



Der *RSPLIB*-Prototyp

UNIVERSITÄT DUISBURG ESSEN

- Designentscheidungen
  - Plattformunabhängigkeit
  - Aktuell: Linux, FreeBSD, MacOS X, Solaris
  - In Planung: M$-Windows
  - Implementiert in ANSI-C
  - Open Source, GPL-Lizenz
- Grundlegende Bestandteile
  - *RSPLIB* Library für PUs und PEs
    - ASAP-Protokoll (PU/PE-Seite)
  - Registrar
    - ASAP-Protokoll (PR-Seite)
    - ENRP-Protokoll
  - Demo-System



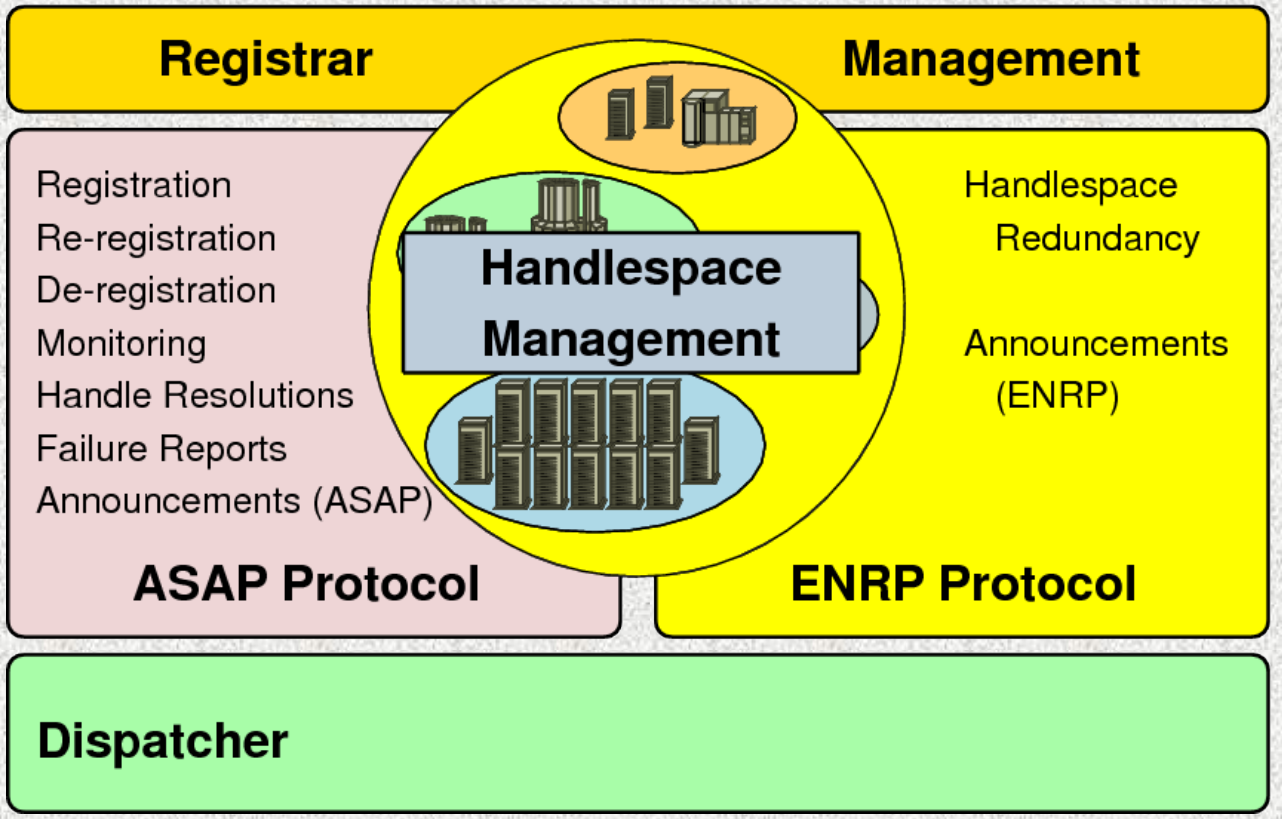
Entwickelt in Kooperation mit Siemens AG, München
mit Unterstützung durch BMBF und DFG

Thomas Dreibholz's Reliable Server Pooling Page
http://tdrwww.exp-math.uni-essen.de/dreibholz/rserpool/

Der Aufbau des Registrars

UNIVERSITÄT DUISBURG ESSEN

- Dispatcher:
  - Plattformspezifische Funktionen:
    - Timer
    - Sockets
    - Threads
- Protokolle:
  - ASAP
    - PR↔PE
    - PR↔PU
  - ENRP (PR↔PR)
- Registrar-Verwaltung:
  - Zugriffskontrolle
  - Adreßverifikation und -filterung



- Handlespace-Management (siehe [Contel2005])

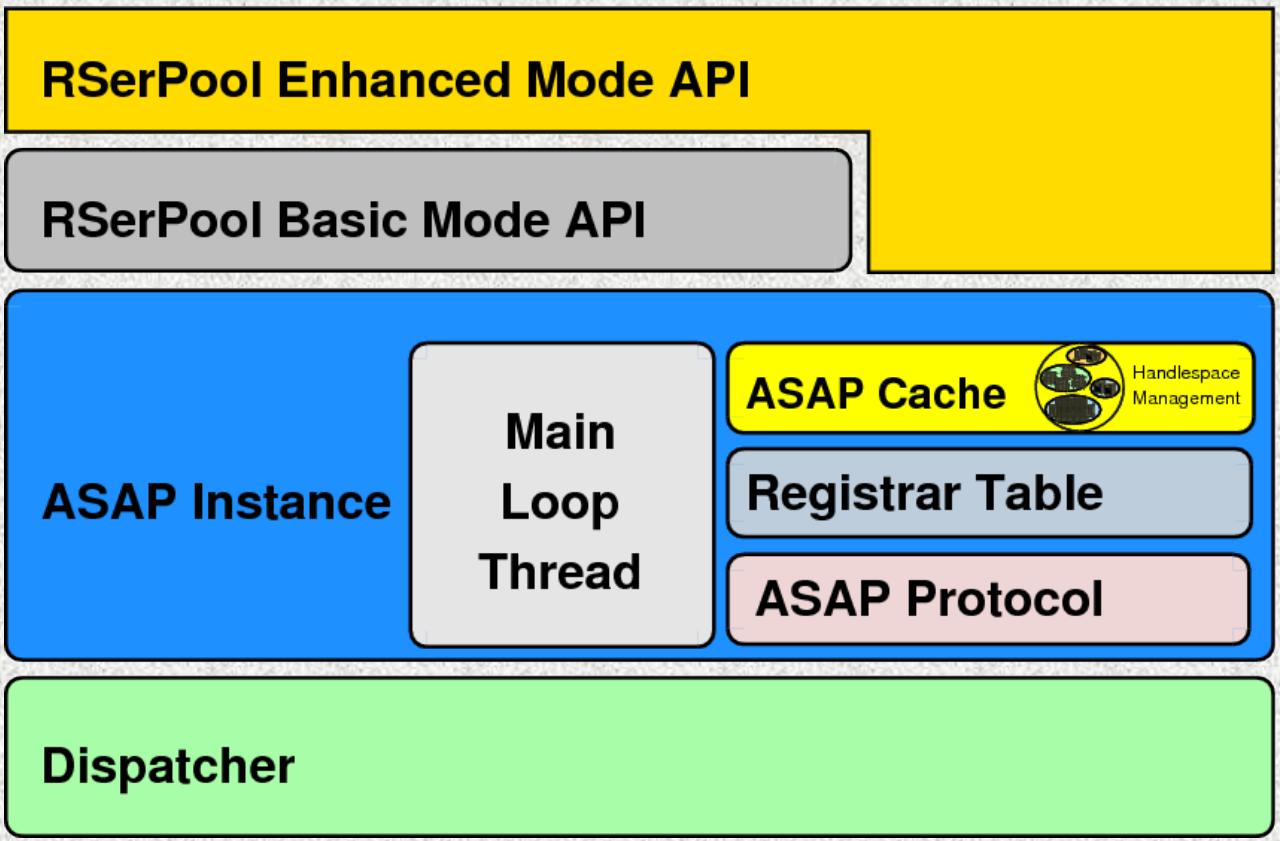
Der Aufbau der PU/PE-Library

UNIVERSITÄT DUISBURG ESSEN

- Dispatcher

- ASAP-Instance:
  - ASAP
    - PE↔PR
    - PU↔PR
    - PU↔PE
  - Liste von Registraren
    - von Announces
  - Cache für PE-Auswahl

- RSerPool-APIs:
  - Basic Mode
  - Enhanced Mode



Die beiden RSerPool-APIs

UNIVERSITÄT DUISBURG ESSEN

- Basic Mode API
  - Für PEs:
    - Registrierung
    - Deregistrierung
  - Für PUs:
    - Handle Resolution
    - Ausfall-Reports
  - Insbesondere:

    PU ↔ PE-Kommunikation **übernimmt Anwendung** komplett selbst!
  - Einsatzzweck:

    Nutzung der RSerPool-Infrastruktur durch bereits existierende Anwendungen

- Enhanced Mode API
  - Kompletter **Session-Layer**
  - Für PEs:
    - Registrierungsverwaltung
    - Verwaltung eingehender Sitzungen (Sessions)
    - Client-basiertes State Sharing
  - Für PUs:

    Session mit Pool, inklusive
    - Auswahl eines PEs
    - Aufbau, Überwachung und **Verwaltung** einer **Transportverbindung**
    - **Failover**-Unterstützung
    - Cookie-Speicherung und Failover mittels Cookie

Das API der *RSPLIB*-Library: Basic Mode für PEs

UNIVERSITÄT DUISBURG ESSEN

- (Re-)Registrierungsschleife
  - z.B. als eigener Thread damit Hauptprogramm (Server) nicht blockiert
  - Registrierung:
    - Regelmäßig im Reregistrierungs-Intervall
- Deregistrierung
  - bei Beendigung

```
void registrationLoopThread()
{
   while(!shuttingDown) {
      rsp_pe_register("MyPool", ...);
      usleep(reregistrationInterval);
   }
   rsp_pe_deregister(poolHandle, ...);
}
```

Das API der *RSPLIB*-Library: Basic Mode für PUs

UNIVERSITÄT DUISBURG ESSEN

- Server-Auswahl
  - API analog zu DNS-Abfrage
  - Kompatibilität zu *getaddrinfo()*
  - Statt *Hostname -> IP-Adressen* jetzt *Pool Handle -> IP-Adressen*
- Failover
  - Aufgabe der Applikation selbst
  - Rückmeldung an PR

```
/* Select a pool element from pool "MyPool" */
rsp_getaddrinfo("MyPool", &eai);
...

/* Create a socket */
sd = socket(eai->ai_family, eai->ai_socktype,
            eai->ai_protocol);
if(sd >= 0) {
  /* Connect to pool element */
  if(connect(sd, eai->ai_addr, eai->ai_addrlen) {
    ...
    if(failure) {
      /* Failure occurred -> report it */
      rsp_pe_failure("MyPool", eai->ai_identifier);
      ...
    }
    ...
  }
}
```

Das API der *RSPLIB*-Library: Enhanced Mode für PEs

UNIVERSITÄT DUISBURG ESSEN

- API analog zu TCP-Sockets
  - Ablauf bei TCP-Sockets: *socket*() -> *bind*() -> *listen*() -> *accept*()
  - Jetzt: Session statt Socket; ggf. automatischer Failover mittels Cookies

```
void serviceThread(session)
{
  rsp_recv(session, command, ...);
  if(command is a cookie) {
    /* Got a cookie -> restore session state */
    Restore state;
    rsp_recv(session, command, ...);
  }
  do {
    /* Handle commands from pool user */
    Handle command;
    rsp_send_cookie(session, current state);
    rsp_recv(session, command, ...);
  } while(session is active);
  rsp_close(session);
}


int main(...)
{
```

```
  /* Create and register pool element */
  poolElement = rsp_socket(0,SOCK_STREAM,IPPROTO_SCTP);
  rsp_register(poolElement, "MyPool", ...);

  /* Handle incoming session requests */
  while(server is active) {
    /* Wait for events */
    rsp_poll(poolElement, ...);

    if(incoming session) {
      /* Accept new session */
      session = rsp_accept(poolElement, ...);
      Create service thread to handle session;
    }
  }

  /* Deregister pool element */
  rsp_deregister(poolElement);
  rsp_close(poolElement);
}
```

Das API der *RSPLIB*-Library: Enhanced Mode für PUs

UNIVERSITÄT DUISBURG ESSEN

- API analog zu TCP-Sockets
  - Ablauf bei TCP-Sockets: *socket*() -> *connect*() -> ... -> *close*()
  - Jetzt: Session statt Socket; ggf. automatischer Failover mittels Cookies

```
/* Create session */
session = rsp_socket(0, SOCK_STREAM, IPPROTO_SCTP);
rsp_connect(session, "MyPool", ...);

/* Run application: file download */
rsp_send(session, "GET Linux-CD.iso HTTP/1.0\r\n\r\n");
while((length = rsp_recv(session, buffer, ...)) > 0) {
   doSomething(buffer, length, ...);
}

/* Close session */
rsp_close(session);
```

- Anmerkung:

  doSomething() erhält ggf. Wiederholungen – je nach Cookie-Intervall!

Unsere RSerPool-Aktivitäten Forschung

UNIVERSITÄT DUISBURG ESSEN

- Forschung im Rahmen eines DFG-Projektes seit Oktober 2004
- Das *RSPSIM* Simulationsmodell
  - Abgeschlossene Arbeiten:
    - Performanzu…on2005][ICN2005]
    - Handhabung… and [LCN2002]
    - Handlespace…el2005]
  - Laufende Arbeite…
    - Performanz i…
    - Untersuchun…lierbarkeit
- Der *RSPLIB* Prototyp
  - Abgeschlossene Arbeiten:
    - Anwendbarkeit und Performanz für SCTP/Add-IP-Mobilität [LCN2003]
    - Distributed Computing mit RSerPool
    - Diplomarbeit [Zha2004], verifiziert Ideen aus [LCN2002] und [LCA2003]
  - Laufende Arbeiten:
    - Verifikation der Performanzergebnisse aus den Simulationen
    - Anwendbarkeit für große Szenarien mittels des PLANET LAB
    - Sicherheit

Interesse an einer Diplom-, Bachelor- oder Master-Arbeit zu RSerPool? Sprechen Sie mit uns!

von der Simulation in die Wirklichkeit

Unsere RSerPool-Aktivitäten
Standardisierung

UNIVERSITÄT DUISBURG ESSEN

- Standardisierung
  - Standardisierungsbeiträge: 3 Working Group Drafts
  - **draft-ietf-rserpool-policies-01.txt**
    - mit Ergebnissen aus unseren Performanzuntersuchungen
  - **draft-ietf-rserpool-mib-01.txt**
    - Definition einer SNMP MIB für RSerPool
  - **draft-ietf-rserpool-api-01.txt**
    - Das Basic- und Enhanced Mode API (sehr früher Entwurf)
  - ... und mehrere Individual Submissions
    - **draft-dreibholz-rserpool-applic-distcomp-01.txt**
    - **draft-coene-rserpool-applic-ipfix-01.txt**
    - **draft-dreibholz-rserpool-mobility-00.txt**
    - draft-dreibholz-rserpool-score-00.txt
  - IETF-Standardisierung läuft über „running code“ - den haben wir!
  - der *RSPLIB*-Prototyp ist die weltweit erste vollständige Implementierung
    - **Open Source** (GPL Lizenz)
    - **Referenz-Implementation** der IETF RSerPool WG
  - Erfolgreiche Interoperabilitäts-Tests mit proprietären Implementierungen von Motorola und Cisco ⇒ Beschleunigung der Standardisierung

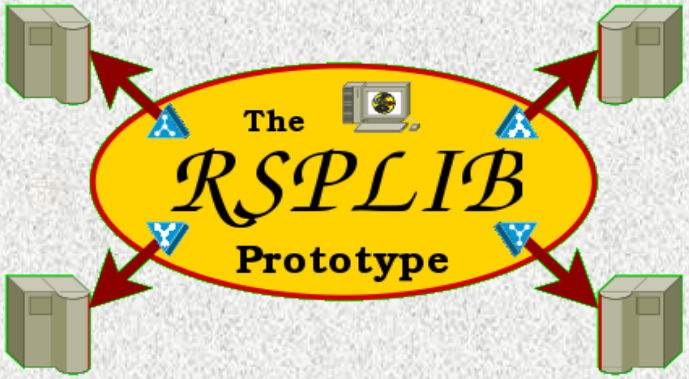
IETF

aus der Forschung in die Anwendung

Zusammenfassung

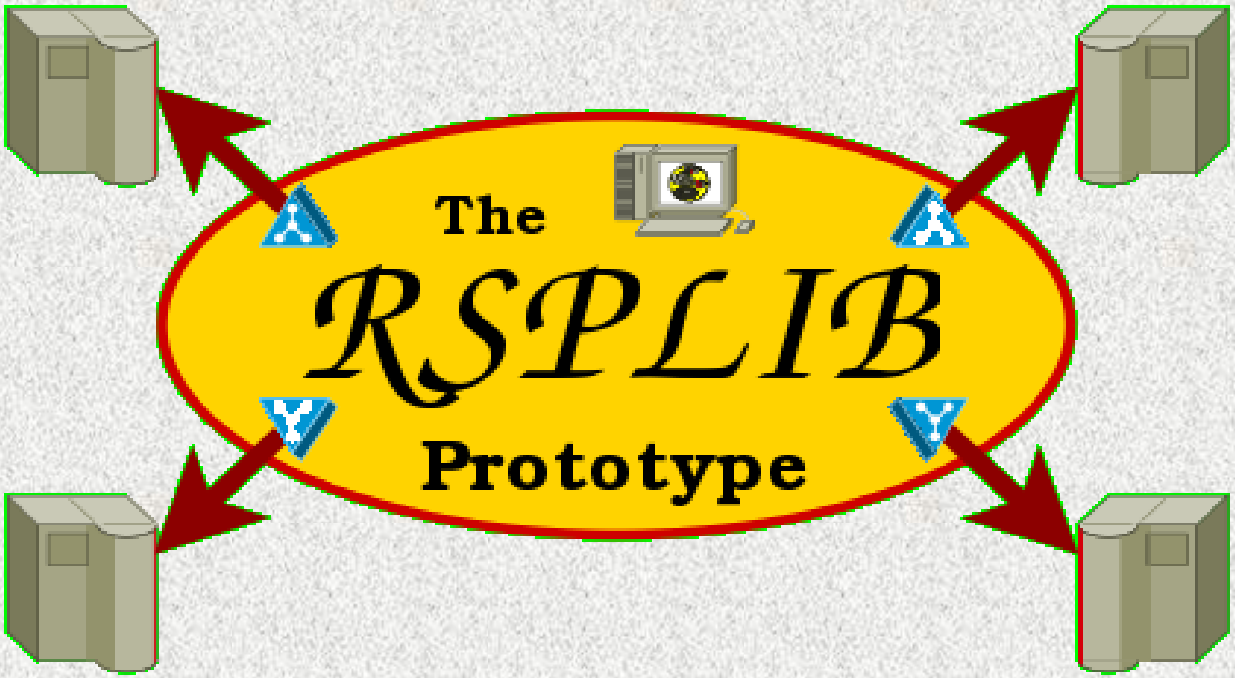
UNIVERSITÄT
DUISBURG
ESSEN

- Motivation
- Was ist Reliable Server Pooling?
  - Architektur
  - Anwendungsszenarien
  - Session-Aufbau und Failover
  - Demo-Vorführung
- Unsere Implementation
  - Designideen und Aufbau der Komponenten
  - Die beiden RSerPool-APIs
    - Basic Mode
    - Enhanced Mode
- Unsere Aktivitäten im Bereich
  Reliable Server Pooling
  - Forschung
  - Standardisierung
- Zusammenfassung



Noch Fragen?

UNIVERSITÄT
DUISBURG
ESSEN



Projekt-Homepage:

http://tdrwww.exp-math.uni-essen.de/dreibholz/rserpool/

Thomas Dreibholz, dreibh@exp-math.uni-essen.de