

# White paper

## Fujitsu Software openUTM Fit4Cluster

Gerade dann, wenn Anwendungen in unternehmenskritischen Bereichen eingesetzt werden, können Ausfallzeiten nicht toleriert werden. Entsprechend dem Paradigma *dynamischer Infrastrukturen* unterstützt Fujitsu Software openUTM diese Hochverfügbarkeit durch die UTM Cluster Funktionalität.



Inhalt	
Einleitung	2
Cluster als Lösung	2
Unterschied zwischen stand-alone und UTM-Cluster-Anwendung	2
Stand-alone UTM-Anwendung	2
UTM-Cluster-Anwendung	2
Cluster-weite Administration	3
Datenbanken	3
SESAM und UDS im BS2000	3
Oracle® RAC	4
Lastverteilung	4
Lastverteilung mit UPIC	4
Lastverteilung zwischen Partner-Anwendungen mit LPAP-Bündeln	5
Externe Lastverteiler	5
Statische Lastverteilung nach Lokalitätsprinzipien	6
Manuelle Lastverteilung	6
Hochverfügbarkeit	7
Ringüberwachung	7
Wiederanmelden bei Ausfall eines Knotens	7
Online Maintenance	7
Gemeinsam genutzter Speicher	8
Gemeinsam genutzter Programmspeicher	8
Gemeinsam genutzter Datenspeicher	8
Disaster Recovery	8
Zusammenfassung	9

## Einleitung

Gerade dann, wenn Anwendungen in unternehmenskritischen Bereichen eingesetzt werden, können Ausfallzeiten nicht toleriert werden. Hochverfügbarkeit der Anwendungen sowie zu Hochlastzeiten die Möglichkeit, die anfallende Last automatisch zu verteilen, haben höchste Priorität und sind ein wesentliches Merkmal dynamischer Infrastrukturen. Anwendungen mit einem Web-Auftritt müssen darüber hinaus 7x24 Stunden verfügbar sein, so dass selbst für Versions-Upgrade kein Wartungsfenster zur Verfügung steht.

Bisherige stand-alone UTM-Anwendungen werden diesen Anforderungen in der Regel nicht gerecht, insbesondere nicht bei Ausfall eines Servers bzw. der auf ihm laufenden UTM-Anwendung, oder bei Versions-Upgrade.

## Cluster als Lösung

Entsprechend dem Paradigma *dynamischer Infrastrukturen* unterstützt openUTM die Hochverfügbarkeit durch die UTM Cluster Funktionalität. Ein Cluster ist eine Anzahl von gleichartigen Rechnern und Betriebssystemen (Cluster-Knoten), die über ein schnelles Netzwerk verbunden sind, und die sich eine gemeinsame Peripherie teilen. Auf einem Cluster kann eine UTM-Anwendung in Form einer UTM-Cluster-Anwendung betrieben werden.

Außerdem bietet openUTM für Partner-UTM-Anwendungen einer UTM-Cluster-Anwendung die Möglichkeit einer automatischen Lastverteilung über LPAP-Bündel für LU6.1 und OSI TP Verbindungen, sowie den UPIC-Lastverteiler für UPIC-Clients. Auch externe Lastverteiler können eingesetzt werden.

## Unterschied zwischen stand-alone und UTM-Cluster-Anwendung

### Stand-alone UTM-Anwendung

Bei einer stand-alone UTM-Anwendung läuft eine UTM-Anwendung in mehreren gleichwertigen Prozessen (*Tasks* im BS2000) auf einem einzelnen Server, siehe Abbildung 1. Aufträge von Clients werden in einer Warteschlange verwaltet und dort von den UTM-Prozessen zur Verarbeitung abgeholt. In den UTM-Prozessen laufen Teilprogramme der Anwendung ab. Diese können auf Prozess-lokalen Datenspeicher, aber auch auf globalen, von allen Prozessen gemeinsam genutzten Datenspeicher (UTM-Speicherbereiche GSSB und ULS, ferner auf BS2000 *shared Memory* oder *Memory-Pools* und auf UNIX-Systemen mittels `mmap` reservierten gemeinsamen Speicher) zugreifen, wobei natürlich bei den nicht von UTM verwalteten Speicherbereichen eigene Synchronisationsmechanismen realisiert sein müssen. Darüber hinaus wird es in der Regel eine externe, gemeinsam genutzte Datenhaltung (z.B. eine Datenbank) geben, auf die alle Teilprogramme Zugriff haben.

Eine stand-alone UTM-Anwendung als Ganzes bleibt in der Regel weiter verfügbar, wenn nur einzelne UTM Prozesse bzw. die auf ihnen laufenden UTM-Teilprogramme abbrechen. Die Anwendung ist jedoch nicht mehr verfügbar, wenn der Server, auf dem die UTM-Anwendung läuft, ausfällt oder die UTM-Anwendung (normal oder abnormal) beendet wird.

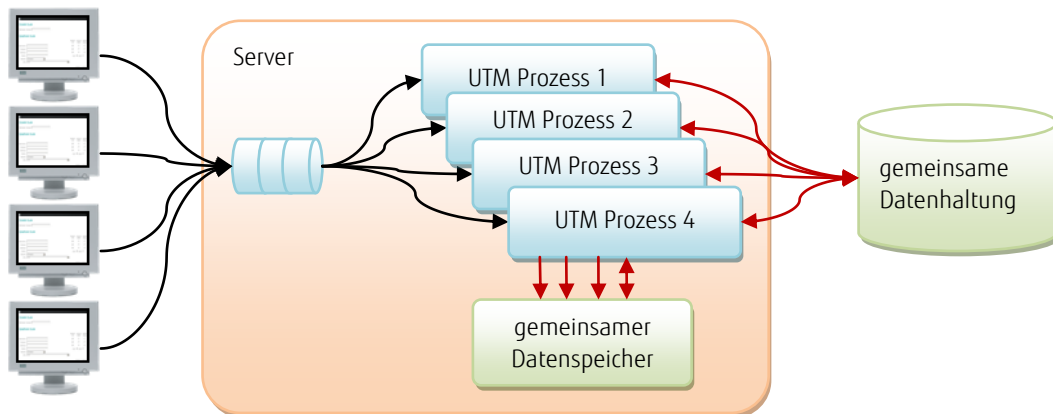


Abbildung 1: Stand-alone Anwendung

### UTM-Cluster-Anwendung

Vereinfacht kann man sich eine UTM-Cluster-Anwendung so vorstellen, dass die identische Anwendung auf mehreren gleichartigen Servern (Cluster-Knoten) mit dem gleichen Betriebssystem abläuft. Somit sind die UTM-Prozesse auf die Cluster-Knoten verteilt und in jeder Knoten-Anwendung kommen die gleichen Teilprogramme zum Ablauf, siehe Abbildung 2. Natürlich kann es dann keinen gemeinsam genutzten globalen Datenspeicher mehr geben; gemeinsam zu nutzender Datenspeicher muss in der Regel externer Speicher sein. Für die von UTM verwalteten Speicherbereiche GSSB und ULS sorgt openUTM ab V6.1 dafür, dass sie Cluster-weit nutzbar sind. Auf Details und Sonderfälle im Zusammenhang mit gemeinsam genutzten Speicherbereichen wird in einem eigenen Abschnitt am Ende dieses White Paper eingegangen. Auch für die gemeinsame Datenhaltung ergeben sich neue Randbedingungen. Da sie von allen beteiligten Knoten erreichbar sein muss, muss sie auf *Network Attached Storage (NAS)* bzw. *Shared Pubset* abgelegt sein, und es muss eine Synchronisation der Zugriffe über die Netzwerkprotokolle (z.B. *Network File System*, NFS bzw. *Cross Coupled System*, XCS) erfolgen. Bei BS2000 scheiden ISAM-Dateien, und damit auch LEASY als Datenbank aus, da die ISAM *shared update* Synchronisation von XCS-Pubsets nicht rechnerübergreifend unterstützt wird. Lösungen für andere Datenbanken sind weiter unten erläutert.

Der Anschluss der Clients erfolgt am besten über einen *Lastverteiler*, der die Verbindung zu einem vom Lastverteiler ausgewählten Knoten herstellt. Aufträge werden dabei weiterhin in die auf dem jeweiligen Knoten befindliche Warteschlange eingetragen und von den dort laufenden

UTM-Prozessen abgeholt. Da auf allen Knoten die gleiche UTM-Anwendung mit den gleichen Teilprogrammen läuft, verhält sich die UTM-Cluster-Anwendung dabei wie eine einzige UTM-Anwendung.

Eine UTM-Cluster-Anwendung als Ganzes bleibt in der Regel auch dann weiter verfügbar, wenn ein Knoten ausfällt. Die Last wird dann auf die anderen Knoten verteilt. Details zu Hochverfügbarkeit und Lastverteilung sind weiter unten beschrieben.

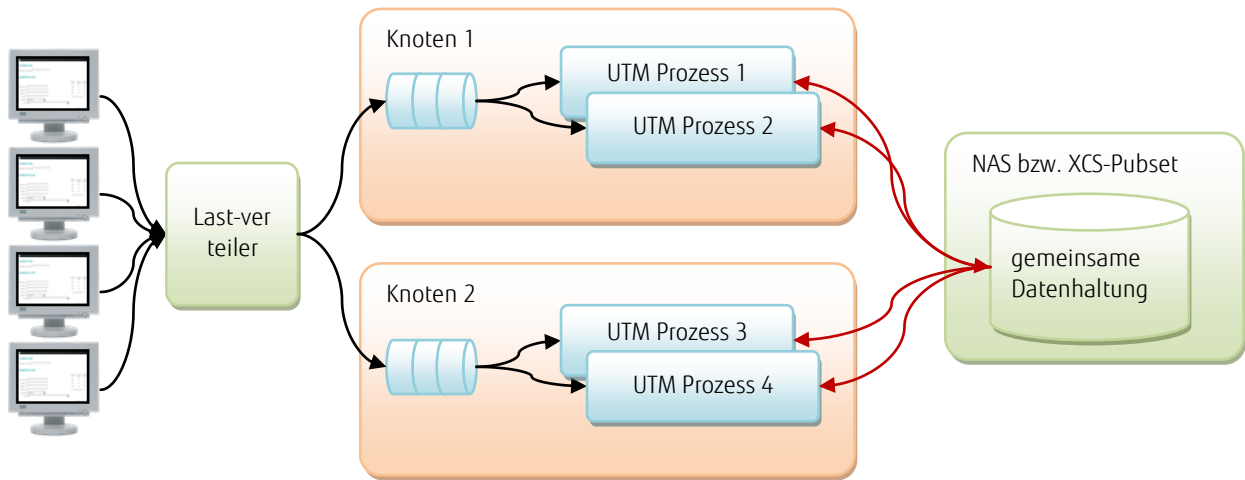


Abbildung 2: Cluster-Anwendung

### Cluster-weite Administration

Da eine UTM-Cluster-Anwendung sich wie eine einzige UTM-Anwendung verhalten soll, wird dies auch durch die Administration unterstützt. Administrative Aktionen, die die Cluster-Anwendung als Ganzes betreffen, wirken Cluster-weit, d.h. nicht nur auf der Knoten-Anwendung, auf der die Aktion primär ausgeführt wurde, sondern sie werden auch von den anderen Knoten-Anwendungen übernommen. Das gilt auch für die Benutzerverwaltung, wie z.B. das Anlegen neuer Benutzer oder die Änderung von Passwörtern. Knoten, die erst nachträglich dem Cluster hinzugefügt werden, übernehmen ebenfalls die aktuellen Einstellungen der anderen Knoten. Administrative Aktionen, die sich gezielt auf einen Knoten beziehen, z.B. Beenden und wieder Hochfahren eines Knotens, werden natürlich nicht von den anderen Knoten übernommen.

### Datenbanken

#### SESAM und UDS im BS2000

SESAM/SQL- und UDS/SQL-Datenbanken werden im BS2000 in der Regel so konfiguriert, dass der *Database Handler* (DBH) auf demselben Rechner läuft wie die UTM-Anwendung. Mit SESAM und UDS ist es möglich und üblich, die Daten inhalts-bezogen auf verschiedene Datenbanken aufzuteilen, z.B. auf eine Kunden-Datenbank und auf eine Artikel-Datenbank, da mehrere Datenbanken von einem DBH prozessiert werden können.

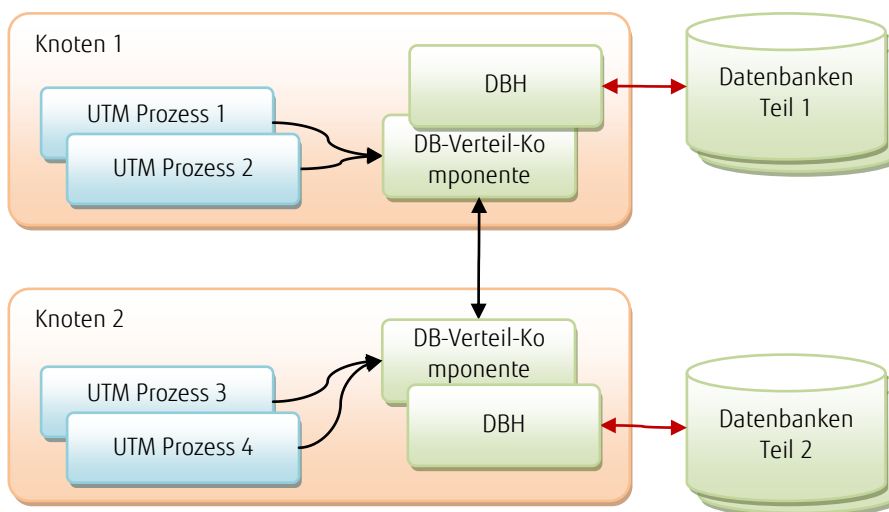


Abbildung 3: Verteilung der Datenbanken

Werden die Datenbanken in einer UTM-Cluster-Anwendung genutzt, dann können sie ferner auf die Knoten aufgeteilt werden, wobei die Knoten sowohl auf die Datenbanken am eigenen Knoten als auch remote auf die Datenbanken am anderen Knoten zugreifen, siehe Abbildung 3. Die DB-Verteil-Komponente ist datenbankspezifisch (SESDCN bei SESAM bzw. UDS-D bei UDS). Sie wird benötigt, um modifizierende Remote-Zugriffe mit Transaktionssicherung abzuwickeln und muss daher auf jedem Knoten installiert werden, von dem aus remote auf einen entfernten DBH

zugriffen wird. Natürlich sind dann die Remote-Zugriffe auf die jeweils entfernten Datenbanken etwas langsamer als die Direktzugriffe auf die Datenbanken am eigenen Knoten.

Um Hochverfügbarkeit zu gewährleisten ist es möglich, bei Ausfall eines Knotens die Datenbank-Verteilung dynamisch umzukonfigurieren, so dass die Zugriffe zu allen Datenbanken ausschließlich von den noch laufenden Knoten aus erfolgen. Diese Umkonfiguration kann automatisch über eine von openUTM bei Erkennen des Knotenausfalls aufgerufene Failure-Prozedur erfolgen, siehe Abschnitt *Hochverfügbarkeit* weiter unten.

### Oracle® RAC

Das openUTM Cluster-Konzept harmoniert gut mit dem Real Application Cluster (RAC) von Oracle®. Dabei wird folgende Konfiguration empfohlen (siehe Abbildung 4): Jedem UTM Knoten ist jeweils ein primärer RAC Knoten zugeordnet. Außerdem verwendet jeder UTM-Knoten die anderen RAC-Knoten als Rückfallstufe für Failover. Die Knoten-Anwendung kommuniziert mit einer DB-Verteil-Komponente, die eine Liste `tnsnames.ora` der angeschlossenen RAC-Knoten führt.

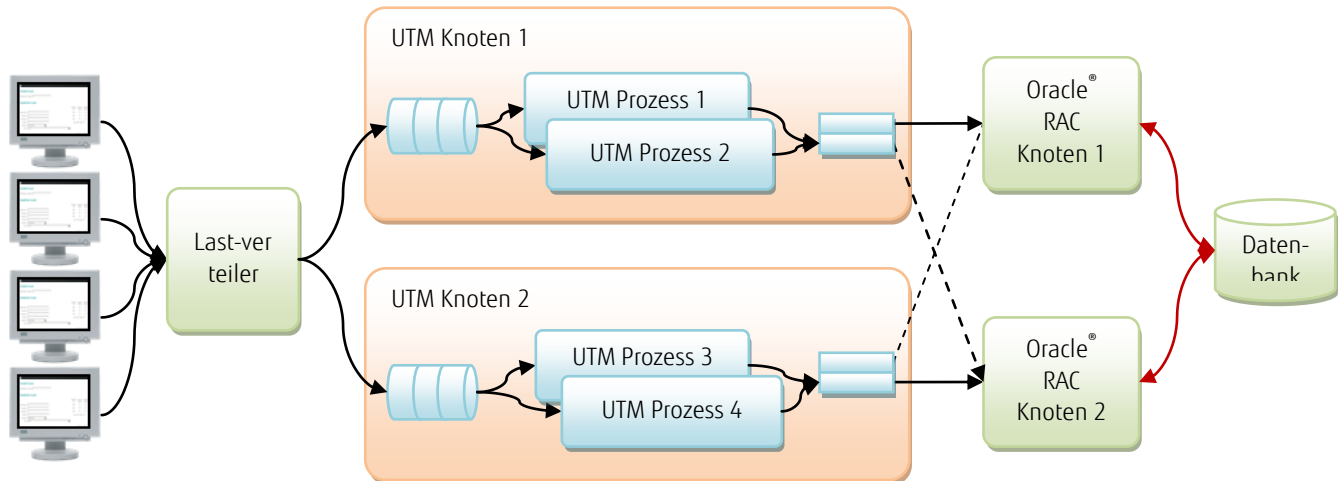


Abbildung 4: UTM Cluster und Oracle RAC

### Lastverteilung

#### Lastverteilung mit UPIC

Für UPIC-Clients sind dezentrale Lastverteiler in die UPIC-Client-Software integriert, siehe Abbildung 5. Bei einer Anmeldung wählt der UPIC-Client zufallsgesteuert eine Knoten-Anwendung aus einer Liste verfügbarer Knoten-Anwendungen aus und baut zu dieser eine Verbindung auf. Diese bleibt bis zum Verbindungsabbau bestehen. Bei einem folgenden Verbindungsaufbau wird erneut zufallsgesteuert ein Knoten ausgewählt. Da alle UPIC-Clients sich so verhalten, wird die Last statistisch gleichmäßig auf die verfügbaren Knoten-Anwendungen verteilt. Dabei ist eine statische Gewichtung der Knoten möglich, wenn z.B. die Knoten unterschiedlich leistungsfähig sind: leistungsfähigere Knoten können mehrfach in den Listen der Knoten-Anwendungen aufgeführt werden und werden damit statistisch häufiger für Verbindungen ausgewählt. Wenn die Knotenauswahl auf eine ausgefallene Knoten-Anwendung trifft, wird versucht, die Verbindung mit einer anderen Knoten-Anwendung herzustellen. Wenn eine Knoten-Anwendung ausfällt und damit eine bestehende Verbindung zusammenbricht, ist ein erneuter Verbindungsaufbau möglich, und dieser führt dann schließlich zu einer Verbindung mit einer anderen, nicht ausgefallenen Knoten-Anwendung.

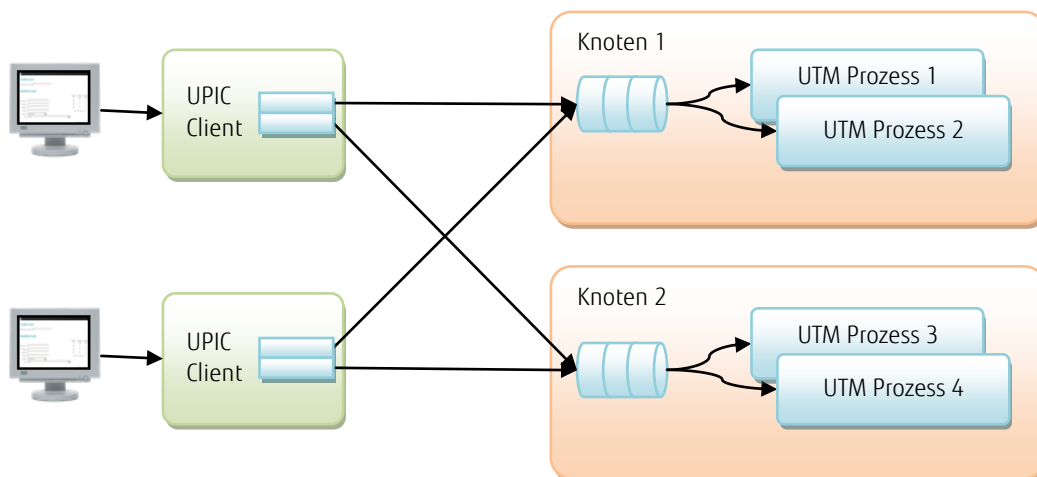


Abbildung 5: Lastverteilung mit UPIC-Clients

### Lastverteilung zwischen Partner-Anwendungen mit LPAP-Bündeln

Wenn eine UTM-Anwendung 1 mit einer Partner-Anwendung 2, die als UTM-Cluster-Anwendung realisiert ist, kommuniziert, ist eine Lastverteilung über LPAP-Bündel für LU6.1 und OSI-TP Verbindungen möglich, siehe Abbildung 6. Die Anwendung 1 adressiert dabei ein Master-LPAP, das dann reihum einen Slave-LPAP auswählt, der mit einem Knoten der Partner-Cluster-Anwendung 2 verbunden ist. Das Master-LPAP führt dazu eine Liste aller zugehörigen Slave-LPAP. Trifft die Auswahl des Slave-LPAP auf eine ausgefallene Knoten-Anwendung, wird der Verbindungsaufbau mit dem nächsten Slave-LPAP versucht.

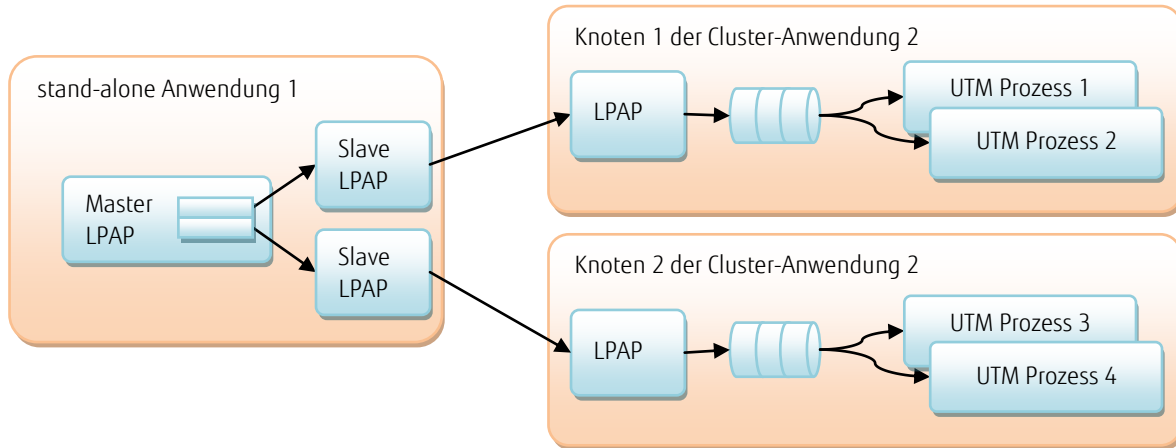


Abbildung 6: Lastverteilung mit LPAP-Bündeln

Zur Vermeidung eines Single Point of Failure kann die Anwendung 1 ebenfalls eine Cluster-Anwendung sein. Eine solche Konfiguration ist z.B. sinnvoll, wenn die UTM-Anwendung einen Web-Auftritt hat, der in der Cluster-Anwendung 1 realisiert ist, während die eigentliche Anwendung als Cluster-Anwendung 2 läuft.

Schließlich ist auch eine symmetrische Partnerschaft zwischen zwei Cluster-Anwendungen möglich, wie dies in Abbildung 7 dargestellt ist. In diesem Fall können beide Cluster-Anwendungen abwechselnd die Rolle des Service-Nutzers bzw. des Service-Anbieters übernehmen, die Initiative für eine Kommunikation also wechselnd mal von einem Knoten der Anwendung 1, mal von einem Knoten der Anwendung 2 aus gehen.

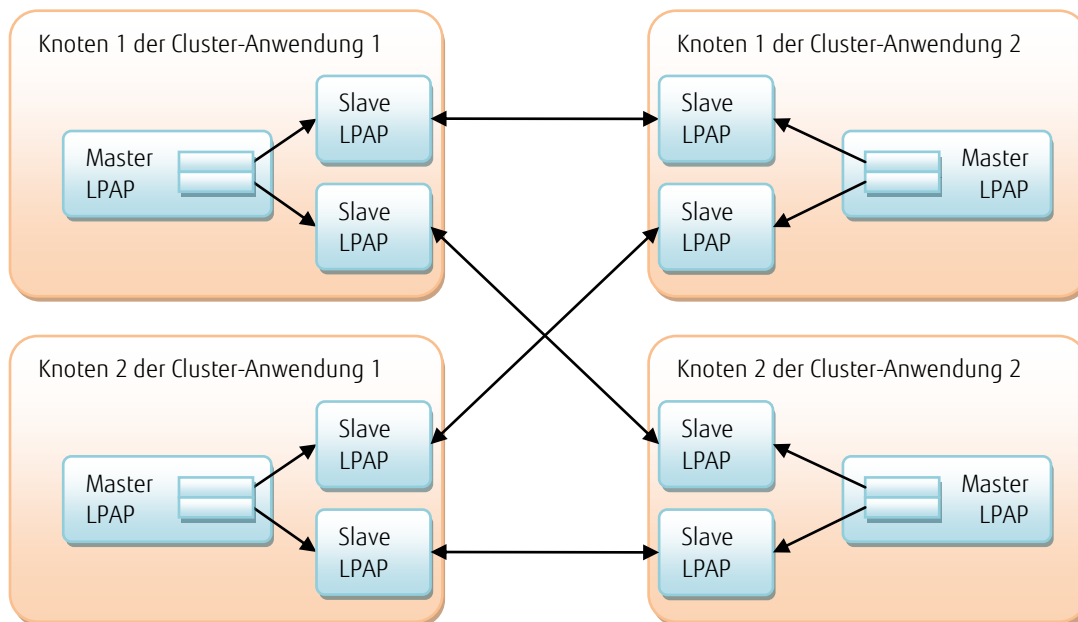


Abbildung 7: Symmetrische Lastverteilung mit LPAP-Bündeln

### Externe Lastverteiler

Bei Einsatz eines externen Lastverteilers kommunizieren Clients über eine virtuelle IP-Adresse, die der Lastverteiler in die IP-Adresse bzw. MAC-Adresse eines ausgewählten Knotens umsetzt, siehe Abbildung 8. Rückantworten laufen dabei ebenfalls über den Lastverteiler, der die IP-Adresse des Antwortenden wieder durch die virtuelle IP-Adresse ersetzt und an den Client weiterleitet.

Cluster-weit wirksame administrative Aktionen (z.B. Änderung eines Benutzerpassworts) können von einem Client aus auf einem von dem Lastverteiler ausgewählten Knoten durchgeführt werden, wobei openUTM dann dafür sorgt, dass diese administrativen Aktionen zeitnah auf allen anderen Knoten nachgeführt werden. Im Allgemeinen wird die Administration aber (z.B. mittels WinAdmin) auf einem vom Administrator

und nicht durch den Lastverteiler ausgewählten Knoten durchgeführt, womit neben Cluster-weit wirksamen administrativen Aktionen, die wieder von UTM auf allen Knoten nachgezogen werden, auch lokale, nur auf dem ausgewählten Knoten wirksame administrative Aktionen (z.B. Knoten beenden und wieder hochfahren) möglich sind.

Externe Lastverteiler können eine Software-Lösung auf einem vorgeschalteten Rechner sein, z.B. *Linux Virtual Server (LVS)*. Auch kann ein handelsüblicher *Redirector*, der zwischen Clients und Server geschaltet ist, als Lastverteiler fungieren.

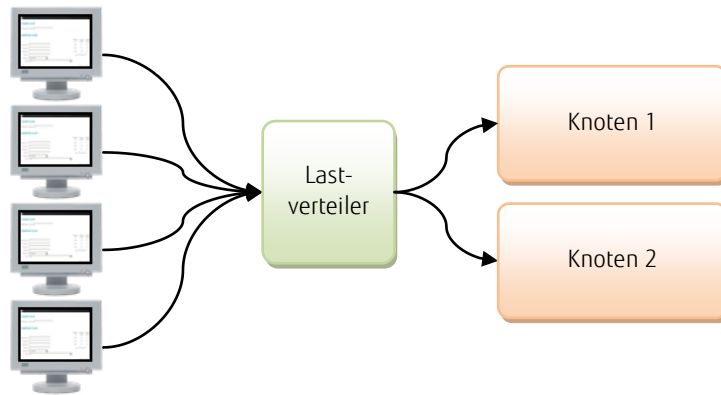


Abbildung 8: Externe Lastverteiler

### Statische Lastverteilung nach Lokalitätsprinzipien

In manchen Fällen ist es sinnvoll, die Lastverteilung und die Verteilung der Datenbanken statisch nach Lokalitätsprinzipien vorzunehmen und die Anwendung entsprechend anzupassen. Wenn in einer Datenbank z.B. Kunden-bezogene Daten verwaltet werden, kann eine statische Verteilung nach Regionen vorteilhaft sein. Dazu können sowohl die Datenbanken, die Knoten-Anwendungen, als auch die verbundenen Clients nach Regionen aufgeteilt werden, so dass im Normalfall ein schneller Zugang zu den lokalen Daten der jeweiligen Region erfolgt. Es ist dann aber auch möglich, von einem Client und Knoten einer Region aus auf die Daten einer anderen Region zuzugreifen, wobei in diesem Fall die längeren Zugriffszeiten für den Remote-Zugriff auf die anderen Datenbanken in Kauf genommen werden. Dies ist in Abbildung 9 für den Fall von BS2000 mit verteilten SESAM-Datenbanken dargestellt.

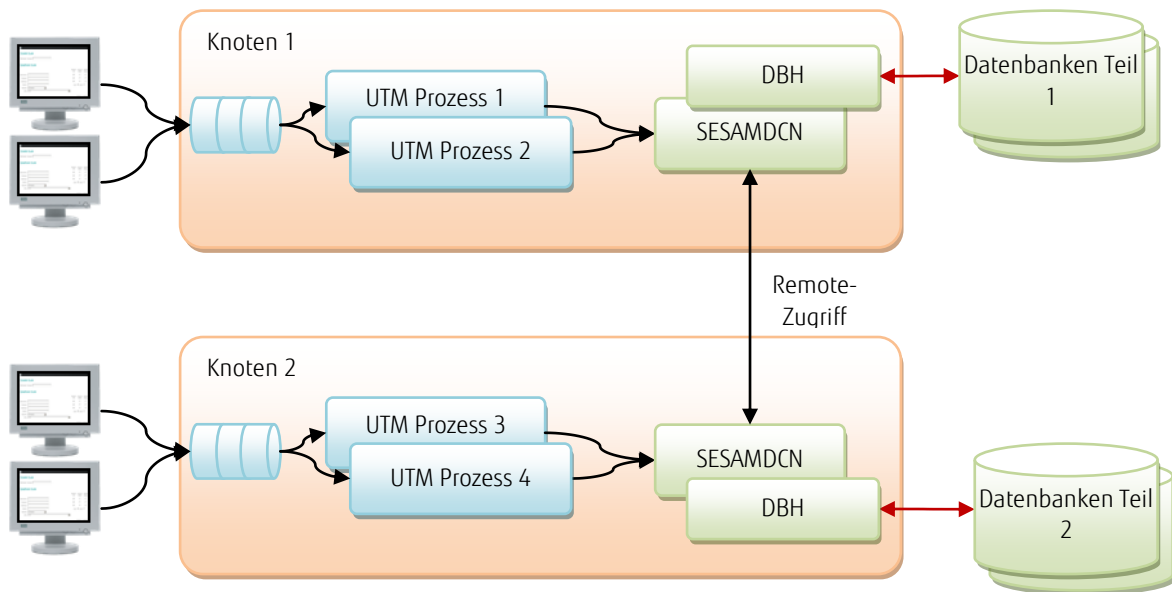


Abbildung 9: Statische Lastverteilung nach Lokalitätsprinzipien

### Manuelle Lastverteilung

Man kann bei einer UTM-Cluster-Anwendung auch auf eine automatische Lastverteilung gänzlich verzichten. In diesem Fall entscheiden die Anwender auf der Client-Seite bei der Anmeldung manuell, mit welchem Knoten sie verbunden werden wollen. Es obliegt dann dem Anwender, zu erkennen, dass ein Knoten hoch ausgelastet ist, und gegebenenfalls eine Neuansmeldung an einem anderen Knoten, der eventuell weniger ausgelastet ist, zu versuchen.

UTM-Cluster-Anwendungen sind auch bei manueller Lastverteilung immer noch von Vorteil in Bezug auf Hochverfügbarkeit, weil es schneller ist, bei Ausfall eines Knotens sich an einem anderen, bereits laufenden Knoten erneut anzumelden, als zu warten, bis bei einer stand-alone-Anwendung ein Systemverwalter das Problem gelöst hat (z.B. den ausgefallenen Rechner und die UTM-Anwendung wieder hochfahren und ggf. weitere Reparaturmaßnahmen ergriffen hat).

## Hochverfügbarkeit

### Ringüberwachung

Die Überwachung der UTM-Cluster-Anwendungen ist ohne zusätzliche Software möglich. Die Knoten-Anwendungen überwachen sich gegenseitig: Jeder Knoten überwacht einen seiner Nachbarn, siehe Abbildung 1. Bei Ausfall einer Knoten-Anwendung ist es möglich, über ein automatisch auf dem überwachenden Knoten aufgerufenes Skript bzw. Kommandoprozedur Folgeaktionen auszulösen. Beispiele möglicher Folgeaktionen sind: automatische Prüfung (mittels `ping`) ob der Knoten-Rechner oder nur die Knoten-Anwendung ausgefallen ist, automatischer Remote-Wiederanlauf-Versuch einer ausgefallenen Knoten-Anwendung, automatischer Remote-Reboot-Versuch eines ausgefallenen Knoten-Rechners, automatisches temporäres Entfernen des ausgefallenen Knotens aus der Konfigurationsliste eines externen Lastverteilers, Information eines Systemverwalters oder Operators per automatisch gesendeter E-Mail oder per SMS. Wenn der Remote-Wiederanlauf-Versuch nicht erfolgreich ist, kann ferner automatisch eine Knoten-Recovery angestoßen werden, die eventuell noch von dem ausgefallenen Knoten gehaltene Sperren auflöst.

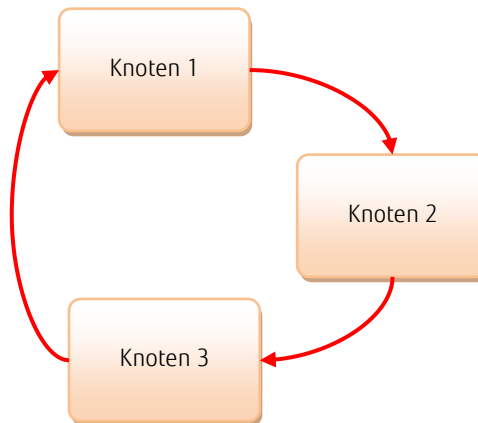


Abbildung 10: Ringüberwachung

Ausgefallene Knoten werden von openUTM selbst aus der Ringüberwachung herausgenommen; wieder hochgefahrte Knoten werden von openUTM automatisch in die Ringüberwachung aufgenommen. Der UPIC-Lastverteiler übergeht automatisch ausgefallene Knoten und berücksichtigt sie wieder, sobald sie wieder zur Verfügung stehen.

### Wiederanmelden bei Ausfall eines Knotens

Wenn die Verbindung eines Client zu einer Knoten-Anwendung verloren geht, kann der Anwender eine erneute Anmeldung versuchen, wobei der zugehörige Vorgangs-Kontext aufgegeben wird. Bei `MULTI-SIGNON = NO` wird in der Regel ein automatischer Warmstart-Versuch abgewartet. Falls dieser nicht gelingt, ruft openUTM ein vom Anwendungsersteller bereitgestelltes Skript bzw. eine Kommandoprozedur auf, mit der man (ab openUTM V6.1) z.B. die auf der ausgefallenen Knoten-Anwendung angemeldet gewesenen User administrativ abmelden kann, um ihnen eine erneute Anmeldung auf einem anderen Knoten zu ermöglichen.

Falls ein UPIC-Lastverteiler auf der Client-Seite im Einsatz ist, wird dieser automatisch die ausgefallene Knoten-Anwendung übergehen und eine Verbindung mit einem anderen Knoten herstellen. Analoges gilt auch bei UTM-Partner-Anwendungen, bei denen der Master-LPAP bei der Lastverteilung ausgefallene Knoten-Anwendungen übergeht. Falls ein externer Lastverteiler im Einsatz ist, sollte beim Ausfall einer Knoten-Anwendung der betroffene Knoten aus der Konfigurationsliste des Lastverteilers entfernt werden, siehe oben unter *Ringüberwachung*, so dass dieser Knoten bei einem Wiederanmeldungsversuch nicht mehr berücksichtigt wird. Auch dies kann über das von openUTM aufgerufene Skript bzw. die Kommandoprozedur erfolgen.

## Online Maintenance

### Software-Upgrade

Bei einer stand-alone-Anwendung ist es bereits möglich, Anwendungsprogramme auszutauschen, ohne dass dazu die Anwendung beendet werden muss. Dazu wird der Austausch UTM-Prozess-weise durchgeführt, wobei während einer Übergangszeit einige Prozesse noch mit der alten Anwendung und einige Prozesse bereits mit der neuen Anwendung laufen.

Bei einer UTM-Cluster-Anwendung ist darüber hinaus ein Upgrade der UTM- oder Betriebssystem-Version Knoten-weise möglich. Dazu wird auf einem Knoten die Knoten-Anwendung heruntergefahren, der Software-Upgrade auf diesem Knoten durchgeführt und anschließend die Knoten-Anwendung wieder hochgefahren. Damit eine Knoten-Anwendung geordnet heruntergefahren werden kann, müssen sich alle momentan aktiven Anwender der UTM-Cluster-Anwendung kurzzeitig abmelden. Sie können sich aber sofort nach dem Beenden der ausgefallenen Knoten-Anwendung und, bei externem Lastverteiler, dem Herausnehmen des runtergefahrenen Knotens aus seiner Konfigurationsliste, wieder anmelden und landen dabei auf einem der anderen Knoten-Anwendungen. Nachdem der Software-Upgrade durchgeführt und die Knoten-Anwendung wieder hochgefahren wurde, kann er wieder in die Konfigurationslisten der Lastverteiler eingetragen werden, so dass er ab sofort wieder berücksichtigt wird.

### Hardware-Austausch bzw. Hinzufügung neuer Knoten



Für den Hardware-Austausch bei einem bestimmten Knoten ist zunächst genauso zu verfahren, wie beim Software-Upgrade. Per Administration kann die neue Konfiguration des Knotens UTM mitgeteilt werden, ehe der Knoten mit der ausgetauschten Hardware in Betrieb genommen wird. Es empfiehlt sich, bei der Erstgenerierung einer UTM-Cluster-Anwendung Reserve-Knoten zu generieren, die im Bedarfsfall später für konkrete neue Hardware umkonfiguriert werden können. Auf diese Weise ist es möglich, ohne die UTM-Cluster-Anwendung beenden zu müssen, neue Knoten hinzuzufügen. Die neuen Knoten übernehmen dabei automatisch alle seit der Erstgenerierung erfolgten Cluster-weit wirksamen administrativen Änderungen der UTM-Cluster-Anwendung (z.B. bzgl. Benutzerverwaltung) und werden in die Ringüberwachung mit einbezogen. Neue Knoten müssen dann auch in die Konfigurationslisten der verwendeten Lastverteiler aufgenommen werden, damit sie fortan berücksichtigt werden.

#### Online Import

Asynchrone Aufträge, die in der Warteschlange einer zu Wartungszwecken beendeten Knoten-Anwendung stehen, können – gesteuert durch die Administration – auf andere Knoten-Anwendungen übertragen und dort ausgeführt werden. Damit ist garantiert, dass diese Aufträge nicht verloren gehen.

### Gemeinsam genutzter Speicher

#### Gemeinsam genutzter Programmspeicher

Werden bei einer stand-alone-Anwendung UTM-Teilprogramme in gemeinsam von allen UTM-Prozessen genutztem Speicher (im BS2000 z.B. *shared Memory* oder *Memory-Pools*) bereitgestellt, so ist dies auch bei einer Transformation in eine UTM-Cluster-Anwendung weiterhin möglich. Die Teilprogramme müssen ohnehin auf allen Knoten installiert werden. Die Knoten-spezifischen Installationen können dann auf jedem Knoten weiterhin in gemeinsam genutztem Speicher (z.B. *Memory-Pools*) bereitgestellt werden.

#### Gemeinsam genutzter Datenspeicher

Es gibt verschiedene Gründe für von verschiedenen Teilprogrammen und/oder verschiedenen UTM-Prozessen gemeinsam genutzten Datenspeicher, der sich sowohl im Hauptspeicher als auch auf Dateien befinden kann. Im Folgenden wird zunächst auf einige ausgewählte Beispiele und deren Behandlung bei der Umstellung einer stand-alone-Anwendung auf eine UTM-Cluster-Anwendung eingegangen. Am Ende wird der allgemeine Fall behandelt.

#### Verbindungs-lokaler Speicher

Manche Anwendungen verwalten Verbindungs-bezogene Daten in von mehreren Teilprogrammen gemeinsam genutztem Speicher (im BS2000 z.B. in *Memory-Pools*). Solange diese Daten nur von Verbindungsaufbau bis Verbindungsabbau leben und nicht über den Verbindungsabbau hinaus von nachfolgenden Vorgängen gelesen werden, kann dieses Konzept auch bei der Transformation einer stand-alone-Anwendung auf eine UTM-Cluster-Anwendung unverändert beibehalten werden. Jeder Knoten verwaltet dann seinen eigenen Verbindungs-lokal gemeinsam genutzten Datenspeicher. Da zwischen Verbindungsaufbau und Verbindungsabbau kein Wechsel des verbundenen Knotens stattfindet, wird der Vorgangs-lokal gemeinsame Datenspeicher auch nur Knoten-lokal genutzt.

#### Nur lesender Zugriff auf weitgehend konstante Daten

Manche Anwendungen stellen in gemeinsam genutzten Speicherbereichen oder Dateien Anwendungs-globale konstante Daten (z.B. Anwendungs-spezifische Konfigurationswerte) allen Teilprogrammen zur Verfügung. Diese Daten werden nur bei Update der Anwendungsprogramme, nicht aber im laufenden Betrieb der Anwendung geändert.

Auch in diesem Fall kann das Konzept beibehalten werden, denn eine Synchronisation beim Zugriff auf diese konstanten Daten ist nicht erforderlich. Die Daten müssen zusammen mit den Teilprogrammen auf allen Knoten installiert bzw. beim Starten der Anwendung im Hauptspeicher initialisiert werden.

#### Der allgemeine Fall

Im Allgemeinen müssen globale Daten (z.B. in Dateien, LEASY-Datenbanken, Memory-Pools oder in AREAs abgelegten Daten), auf die gemeinsam von allen Teilprogrammen Vorgangs- und Verbindungs-übergreifend sowohl lesend als auch schreibend zugegriffen wird, auf externem Speicher abgelegt und Zugriffe Knoten-übergreifend synchronisiert werden. Am besten eignen sich dazu entweder die von openUTM verwalteten Speicherbereiche GSSB und ULS oder aber Datenbank-Tabellen, da in diesem Fall das Datenbanksystem für die Synchronisation der Zugriffe sorgt.

Ab openUTM V6.1 wirken die von openUTM verwalteten Datenbereiche GSSB und ULS Cluster-weit, d.h. alle Knoten-Anwendungen haben die gleiche Sicht auf die Inhalte, und openUTM sorgt für die Knoten-übergreifende Synchronisation der Zugriffe.

### Disaster Recovery

Oft sind neben Hochverfügbarkeit und Reaktion auf Ausfall einer Knoten-Anwendung auch Vorkehrungen für den Katastrophenfall gefordert. Für Disaster Recovery werden dazu Standby-Server und Peripherie in einem zweiten Data Center an einem entfernten Ort aufgestellt. Fällt im Katastrophenfall das erste Data Center total aus, dann kann das Stand-by Data Center den Betrieb übernehmen. Dazu werden die im ersten Data Center betriebenen Plattenspeicher permanent im Standby Data Center gespiegelt, so dass das Standby Data Center jederzeit in der Lage ist, den Betrieb mit aktuellen Daten zu übernehmen, siehe Abbildung 11. Im Katastrophenfall werden die Server und Anwendungen im Standby Data Center hochgefahren und arbeiten auf den gespiegelten Plattenspeichern. Das Umschalten kann in der Regel automatisiert werden (bei BS2000 z.B. mit HIPLEX-AF).



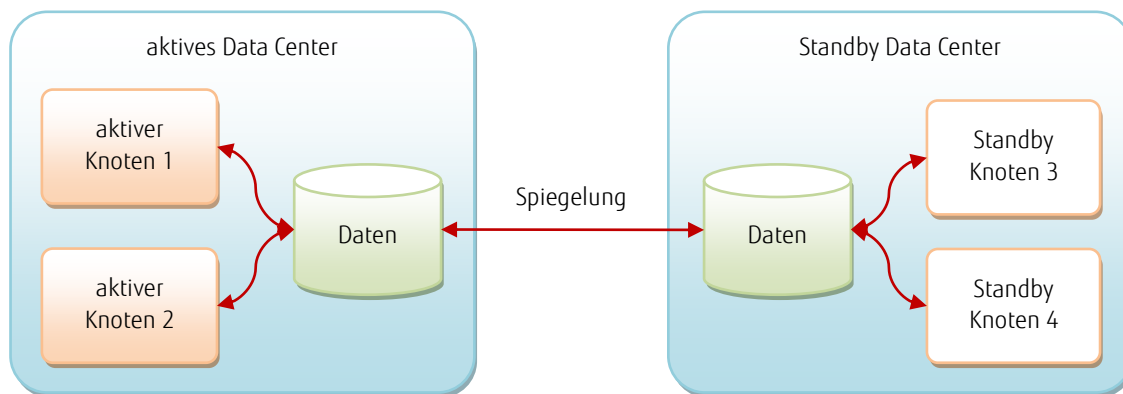


Abbildung 11: Räumlich getrenntes aktives und Standby Data Center

Aktive UTM-Cluster-Knoten sollten aus Performance-Gründen über ein schnelles Netzwerk miteinander verbunden sein. Sie teilen sich auch eine gemeinsame Peripherie. Dies erfordert, dass die Knoten nahe beieinander stehen, in der Regel im selben Data Center. Dennoch fügt sich das UTM-Cluster-Konzept gut in das oben geschilderte Disaster Recovery Konzept ein:

Es werden sowohl die Server im ersten Data Center als auch die Server im entfernten zweiten Data Center als UTM-Cluster-Knoten für dieselbe UTM-Anwendung konfiguriert. In beiden Data Centern sind die dort aufgestellten Server und ihre lokale Peripherie über ein schnelles lokales Netz miteinander verbunden.

Im Normalfall werden nur die Knoten-Anwendungen im ersten Data Center gestartet, während die Knoten-Anwendungen im Standby Data Center inaktiv – eben standby – bleiben. Damit sind alle aktiven Cluster-Knoten inklusive der von ihnen gemeinsam genutzten Peripherie mit einem schnellen lokalen Netz verbunden, wodurch gute Performance gewährleistet ist.

Im Katastrophenfall wird umgeschaltet auf die Knoten im Standby Data Center, die dort ebenfalls wieder über ein schnelles lokales Netz miteinander und mit der gespiegelten Peripherie verbunden sind. Zum Umschalten werden nun die Knoten im Standby Data Center hochgefahren und die Knoten-Anwendungen gestartet. Falls einige Knoten-Anwendungen im ausgefallenen ersten Data Center noch laufen sollten, werden diese beendet, so dass nunmehr ausschließlich die Knoten im Standby Data Center aktiv sind. Damit ist auch nach dem Umschalten wieder gute Performance gewährleistet.

Alle seit der Erstgenerierung im ersten Data Center vorgenommenen administrativen Änderungen an der UTM-Cluster-Anwendung sind in gespiegelten Dateien des Administrations-Journals gespeichert und werden automatisch von den nun im Standby Data Center gestarteten Knoten-Anwendungen nachgezogen. Anwender können sich danach sofort wieder anmelden.

### Zusammenfassung

Entsprechend dem Paradigma *dynamischer Infrastrukturen* unterstützt openUTM die Hochverfügbarkeit durch die UTM Cluster Funktionalität. Ein Cluster ist eine Anzahl von gleichartigen Rechnern und Betriebssystemen (Cluster-Knoten), die über ein schnelles Netzwerk verbunden sind, und die sich eine gemeinsame Peripherie teilen. Auf einem Cluster kann eine UTM-Anwendung in Form einer UTM-Cluster-Anwendung betrieben werden.

### Contact

FUJITSU Technology Solutions GmbH  
Address: Mies-van-der-Rohe Str. 8, 80807 München,  
Germany  
E-mail: openSEAS@ts.fujitsu.com  
Website: www.fujitsu.com/de  
2015-03-30

© 2015 Fujitsu Technology Solutions GmbH. Fujitsu, the Fujitsu logo, are trademarks or registered trademarks of Fujitsu Limited in Japan and other countries. Other company, product and service names may be trademarks or registered trademarks of their respective owners. Technical data subject to modification and delivery subject to availability. Any liability that the data and illustrations are complete, actual or correct is excluded. Designations may be trademarks and/or copyrights of the respective manufacturer, the use of which by third parties for their own purposes may infringe the rights of such owner.