tubIT Stammtisch

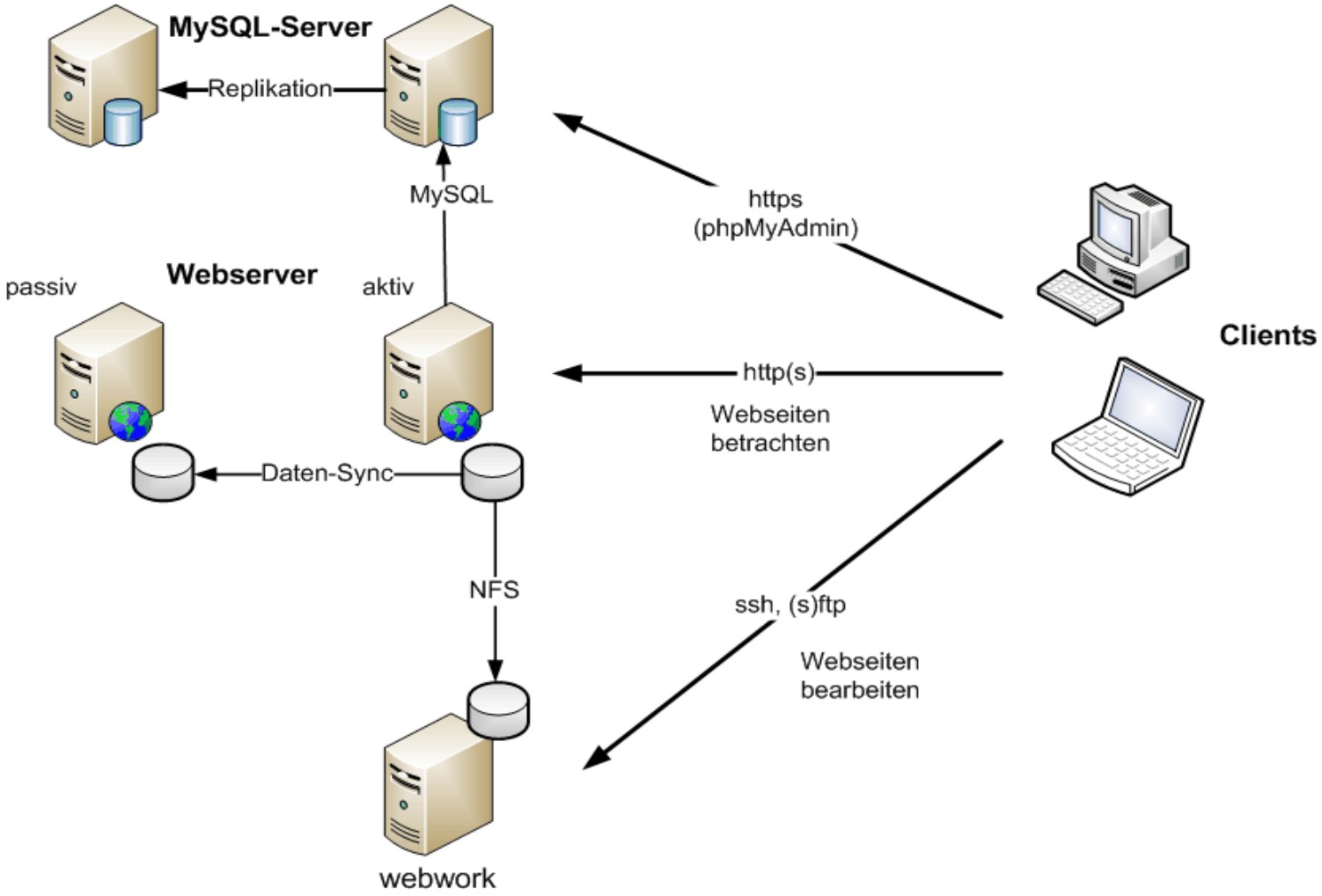
das neue Webserverkonzept

Stefanie Wenig (wenig@tubit.tu-berlin.de)
Roland Hager (hager@tubit.tu-berlin.de)
IT Dienstleistungszentrum der TU Berlin

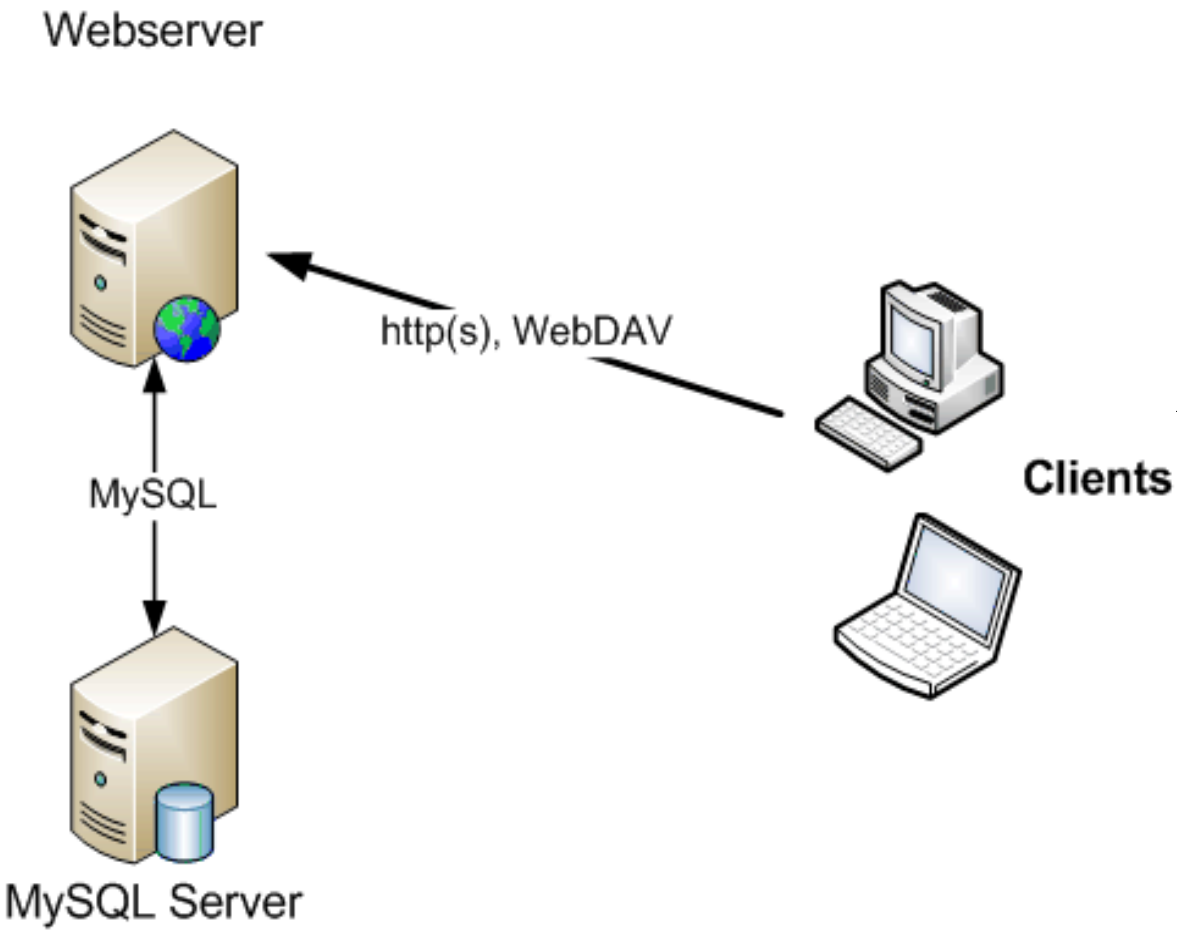
Inhalt

- Ist-Zustand
  - Vor- und Nachteile
- Soll-Zustand
  - Vor- und Nachteile
  - Software und Versionen
- Vergleich der Hardware
- Migration vorhandener Auftritte
- Besonderheiten beachten

Ist-Zustand (konv. Webserver)



Ist-Zustand (Typo3 Variante 2)



Vor- und Nachteile Ist-Zustand

Konventionelle Webserver

+ kein direkter Zugriff auf die Webserver
+ Trennung der Auftritte über vhosts
+ cgi-Skripte nur in bestimmten Verzeichnissen

- nur ein aktiver Webserver
- der Apache läuft immer im gleichen Userkontext

Zugriff auf Daten:

- alle Webdaten müssen weltweit lesbar sein um vom Webserver angezeigt zu werden
- Redakteure können fremde Daten zwar nicht ändern aber jederzeit einsehen
- Skripte können absichtlich oder versehentlich fremde Inhalte auslesen

Vor- und Nachteile Ist-Zustand

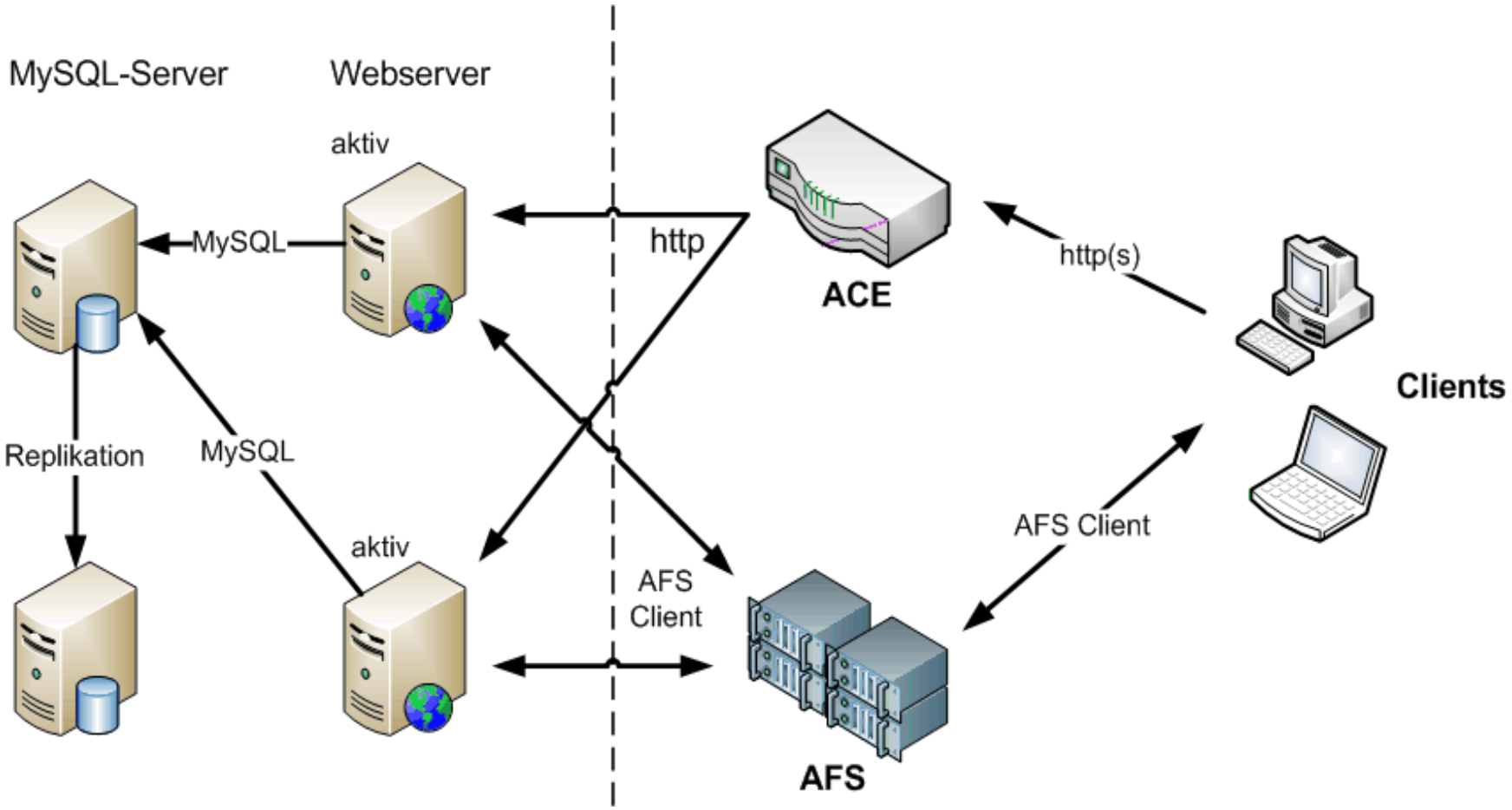
Typo3 Variante 2 (eigene Instanz)

Mehr Sicherheit durch ...
+ nur php erlaubt
+ safe_mode =on und safe_mode_gid=on
  => lokale Daten, die vom Webserver aufgerufen werden, müssen www-data als Gruppe besitzen

Zugriff auf Dateien:
  + gesetztes open_basedir - nur Dateien des eigenen Auftrittes können geöffnet werden
  + über webDAV nach Authentisierung möglich
  + über Typo3-Backend

+ gesetztes safe_mode_exec_dir, um das Ausführen von lokalen Programmen über exec() oder system() einzuschränken

Soll Zustand



Vor- und Nachteile Soll-Zustand (1)

+ kein direkter Zugriff auf die Webserver
+ Trennung der Auftritte durch vhosts
+ cgi-Skripte nur in speziellen Verzeichnissen
+ zwei aktive Webserver
+ Verwendung von suexec
    => Standard-Webuser liefert statische Dateien aus
    => separater Serviceuser für jeden vhost, in dessen Kontext Skripte ausgeführt werden (cgi/php)
+ Verwendung von fastcgi
    => schnelle Ausführung von php-Skripten als cgi mit den Rechten des separaten Serviceusers
    => individuelle Wahl der PHP-Version pro vhost
    => individuelle php.ini pro vhost

Vor- und Nachteile Soll-Zustand (2)

Zugriff auf Dateien:

+ auf Verzeichnisebene mit ACLs geschützt
+ open_basedir gesetzt
+ Einschränkung von Systemaufrufen (cat, ls etc.) aus allen Skripten durch suexec
+ Weltweiter Zugriff per AFS
+ kein Auslesen fremder Dateien bei korrekt gesetzten ACLs

- kein Zugriff auf identisch konfiguriertes System zum Testen der Skripte

Software und Versionen

- PHP 4.4.8 bzw. PHP 5.2.6
- MySQL 5.0.51a
- Perl

Hardwarevergleich

alt

- Typo3-Webserver für die Nebeninstanzen:
  - Intel Xeon CPU 3 GHz (mit HT)
  - 3 GB RAM
- Konv. Webserver
  - Intel Xeon CPU 2.40GHz
  - 2 GB RAM

neu

- 2x Dualcore mit je 2GHz
- 8GB RAM

Migration vorhandener Auftritte (1)

- Mitarbeiter provisionieren
- OrgNamen über Rollenverwaltung eintragen
- Datenbereich (Volume) im AFS beantragen
  (per E-Mail an afs@tubit.tu-berlin.de)
  - Name der Einrichtung
  - Kostenstelle
  - OrgName
  - Größe des Volumes (50 GB sind kostenlos)
  - Verantwortlicher

Migration vorhandener Auftritte (2)

- Migration des Auftrittes beantragen (von www2.tu-berlin.de bzw. Typo3 Variante 2) - Webformular
  - Name der Einrichtung
  - was migrieren (welcher Auftritt und welche Datenbanken)
  - Verantwortlicher
  - ServerName: <hostname>.<orgname>.tu-berlin.de
  - Wünsche (php-Version, php.ini)
- tubIT ...
  - richtet Volume ein
  - definiert Service-User
  - vergibt die notwendigen Rechte
  - richtet den vhost ein
  - informiert den Verantwortlichen

Migration vorhandener Auftritte (3)

- Die Einrichtung sollte
  - die alten Webdaten aussortieren
  - eventuell eine neue Webstruktur erarbeiten
  - Daten in das AFS einspielen
  - Skripte überprüfen, anpassen und testen
  - bei Problemen rückfragen
  - Rechte für alle Bearbeiter setzen (am besten mit Hilfe von Gruppenrechten)

Besonderheiten beachten

- Absolute Pfade anpassen
- Parameter bei Systemaufrufen überprüfen (neues OS!)
- Zugriffsrechte nur pro Ordner
  - Einzelne Dateien können nicht geschützt werden sondern müssen dann in ein separates Verzeichnis kopiert werden.
- Neue Ordner übernehmen die ACLs des übergeordneten Verzeichnisses
- „fs setacl“ funktioniert nicht rekursiv!!!
  - Soll ein ganzer Verzeichnisbaum mit neuen Rechten versehen werden, muss der Befehl auf jeden Unterordner separat ausgeführt werden. Dabei hilft folgender Befehl:
    # find ./* -type d -noleaf -print0 | xargs -0 fs setacl -acl username read -acl username2 write -dir