



Die eigene Dropbox

Ein Fileserver im Eigenbau mit Bordmitteln

Dr. Christoph Zimmermann
Frankfurter Linux User Group
Monatliche FraLUG Vortragsreihe


Die Email

The screenshot shows an email client window with two tabs: 'Calendar' and 'Die Dropbox-Aktion...'. The interface includes a search bar with the text 'Search... <Ctrl+K>', navigation buttons for 'Chat', 'Address Book', 'Tag', and 'Quick Filter', and action buttons for 'Reply', 'Forward', and a trash icon. A notification bar at the top of the email content area reads: 'on über Ihr HTC-Gerät läuft am August 22, 2014 ab.' The main content area features the Dropbox logo, followed by the text: 'Hallo, zu Ihrem HTC-Mobilgerät haben Sie für zwei Jahre 23 GB zusätzlichen Speicherplatz erhalten. Dieser Aktionszeitraum endet am August 22, 2014 und Ihr Dropbox-Speicherplatz wird automatisch auf 3 GB zurückgesetzt. Derzeit sind 4 GB Ihres Dropbox-Kontingents belegt. Wenn Ihr 3 GB-Speicherlimit nach dem August 22, 2014 überschritten ist, können Sie keine neuen Fotos, Videos und Dokumente mehr in Ihrer Dropbox speichern. Wenn Sie Ihre [Lieblingsfunktionen von Dropbox](#) wie das automatische Kamera-Upload auch weiterhin nutzen möchten, sollten Sie sich [mit Dropbox Pro mehr Speicherplatz holen](#).' At the bottom of the content area is a blue button labeled 'Dropbox Pro holen'.

Anforderungen

- 100 % Open Source; auf eigenem System hostbar
- Minimaler H/W Anforderungen
- Einfache Installation
- Abbildung auf Dateisystem ausreichend
- Klienten für: Linux, OSX, Windows, Android (+ optional iOS)
- Aktives Projekt
- Minimale Umgebungsanforderungen (geringer Wartungsaufwand):
 - Linux
 - C / C++
 - Script-Sprache

Der Server

- Compulab Utilite Standard
 - Freescale I.MX6 dual-core Cortex A9
 - 2 GB DDR3
 - MicroSD slot
 - 4 x USB 2.0 + OTG
 - 2 x GbEthernet
 - 2x WLAN: 802.11 b/g + 802.11n
 - Ca. 5 Watt Leistungsaufnahme bei Lastbetrieb
- 
- OS:
 - Ubuntu (bis 12.04 ☹)
 - Arch ☺

Kandidaten

- ownCloud:
 - Groupware-orientierte Plattform
 - CalDAV, Streaming, Photo-Galerie
 - PHP, JS
 - MySQL, SQLite, PostgreSQL
- SparkleShare:
 - Nur Dateisynchronisation
 - C#
 - git-basiertes Backend



Kandidaten (Fort.)

- Seafile:
 - Nur Dateisynchronisation
 - C, Python
 - MySQL, SQLite, PostgreSQL
- Wahl: Seafile
 - ownCloud: zu mächtig, nicht benötigte Features
 - SparkleShare: keine mobilen Klienten!
 - Seafile: minimale Anforderungen für einzigen benötigten Dienst, performant (C, Python)



Installationskomponenten

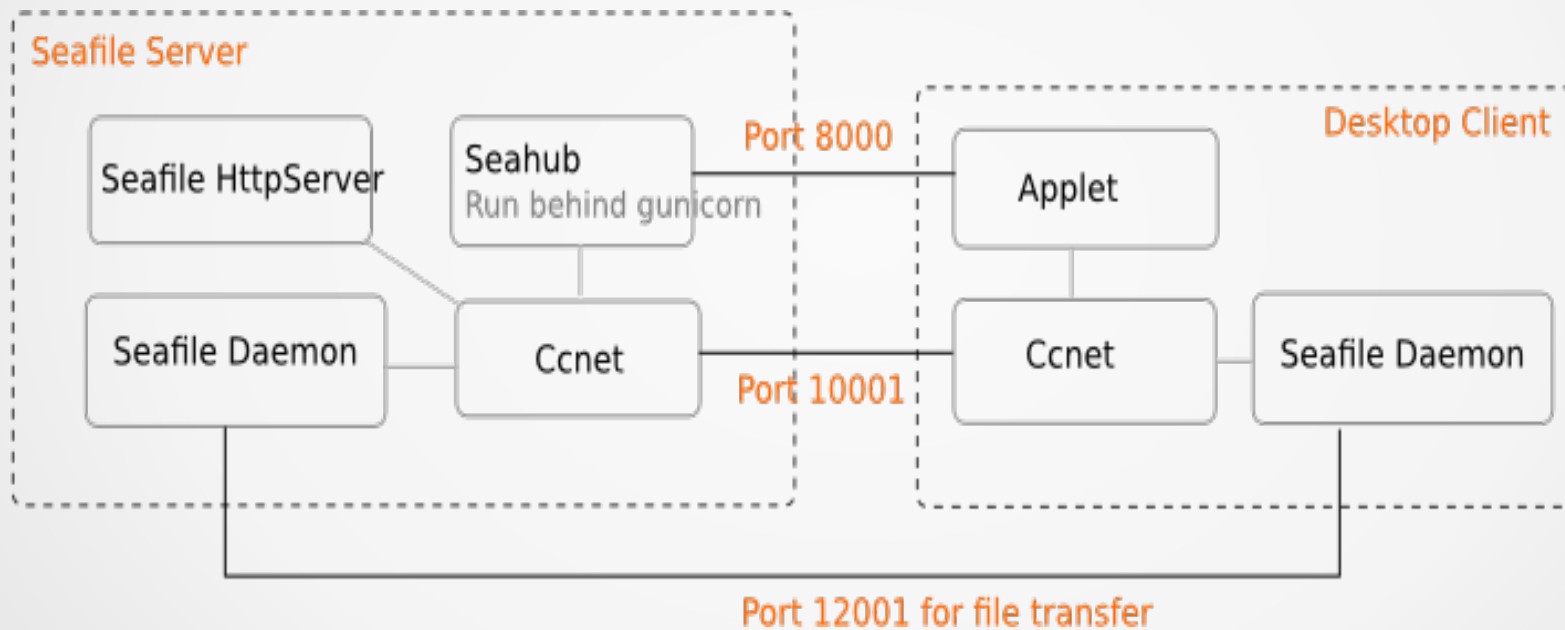
- Voraussetzungen für Version Seafile 3.1:
 - Basis: Python 2.7, SQLite, GCC, git für Repo-Zugriff
 - Gunicorn: WSGI-Server in Python
 - Django: Web-Framework in Python für DB-gestützte Websites
 - djblets: Django-Erweiterung für Authentifizierung, UIs, Caching, etc.
 - Nginx (anstatt Apache): Leichtgewichtiger Webserver für HTTPS-Anbindung

Architektur

- Ccnet (C): Netzwerk-Dämon
 - Netzwerk-Bus für Datenaustausch m. A. direkter Dateiübertragung
 - Implementiert ISO-Stack oberhalb Schicht 4 (inkl. RPC)
- Seafile Daemon (C):
 - Controller für Synchronisation, Versionierung, etc.
- Seahub (Python):
 - Web- / App-Server, basierend auf Gunicorn und Django / djblets
- Seafile HttpServer (C):
 - Upload / Download für Seahub (Re-Implementierung aufgrund von Python Dateizugriff-Performance)

Architektur (Fort.)

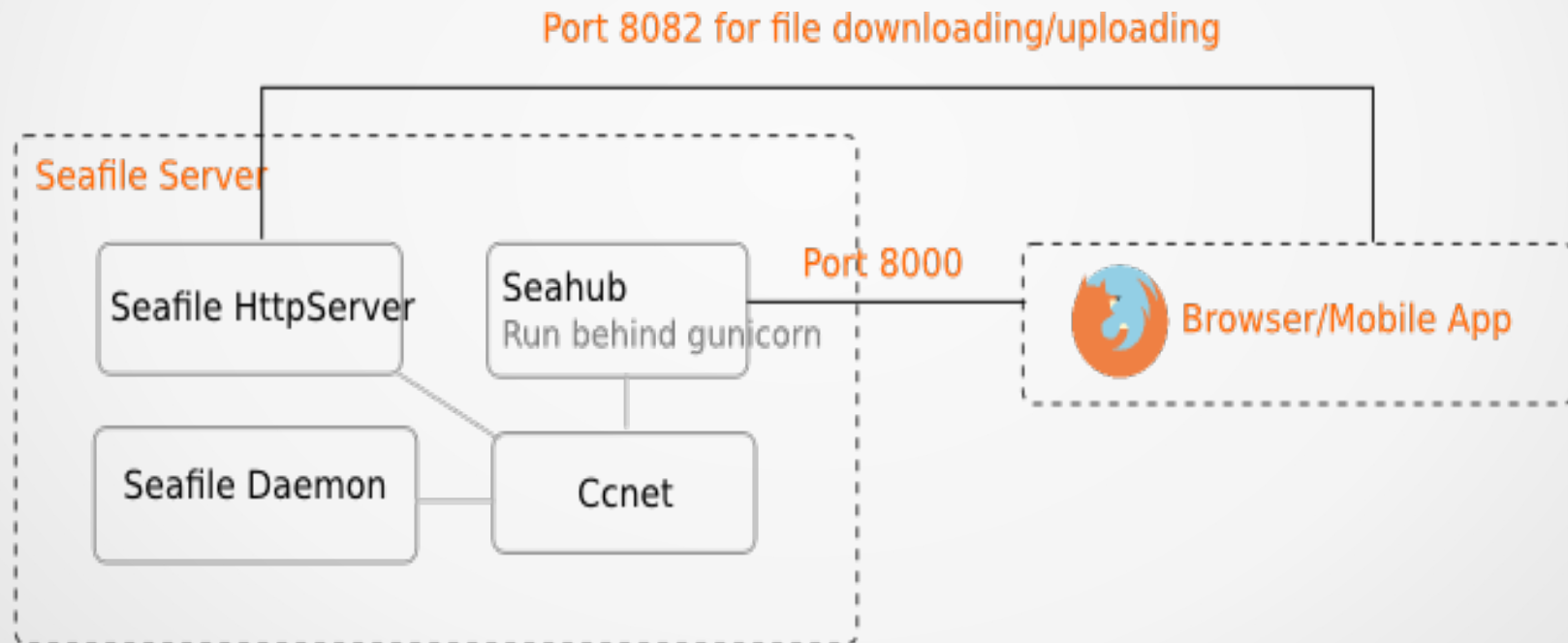
- Desktop-Klient



Grafik © seafile.com

Architektur (Fort.)

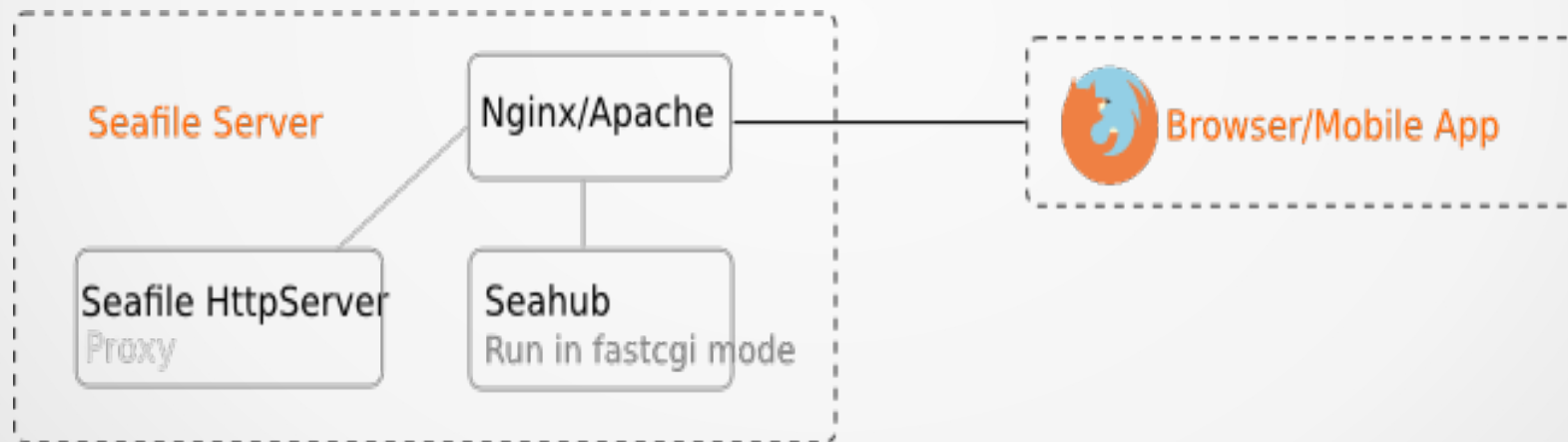
- Mobiler Klient



Grafik © seafile.com

Architektur (Fort.)

- HTTPS via Reverse Proxy



Grafik © seafile.com

Exkurs: eigener DynDNS-Dienst

- Neueste Anforderung seitens no-ip.com:

„ ... Please note, No-IP free users need manually post an update once a month in order to avoid the pending deletion state. Free host names are deleted after 30 days of inactivity... “

- Rezept für eigenen DynDNS:
 - Externer Server inkl. Domain und DNS-Server Instanz
 - Dynamische Domain: Implementierung als Subdomain
 - Updater: lokales Skript als cron-Job o. ä.
 - Updates: via TSIG-signiertem nsupdate-Aufruf

Exkurs: eigener DynDNS-Dienst (Fort.)

- Einrichtung auf DNS-Server:

- Generierung der TSIG-Schlüssel:

```
dnssec-keygen -r /dev/urandom -a HMAC-MD5  
-b 512 -n HOST subdomain.domain.com.
```

- Erzeugt Ksubdomain.domain.* key und private Dateien

- Schlüssel-Einbindung in Bind:

```
key "sfnaswobj32.monochromec.com." {  
    algorithm hmac-md5;  
    secret "<Inhalt der *.key Datei"  
};
```

Exkurs: eigener DynDNS-Dienst (Fort.)

- Update-Policy:

```
update-policy {  
    grant subdomain.domain.com. subdomain  
domain.com. A;  
}
```

- Zonen-Definition:

```
$TTL 3600 ; 1 stunde  
subdomain A ip-adresse
```

Exkurs: eigener DynDNS-Dienst (Fort.)

- Auf Klienten-Seite (Einbindung via cron):

```
IP4=$(curl -s "www.domain.com/ip-script.php")
nsupdate -k K*.key -v << EOF
    server ns.domain.com
    zone domain.com.
    update delete domain.com
    update add domain.com 3600 A $IP4
    send
EOF
```

- Externe IP-Adresse: PHP-Skript auf Server, ip-script.php

```
<?php
    // Ausgabe der IP-Adresse
    echo getenv("REMOTE_ADDR");
```

Achtung!

- Python:
 - WSGI server gunicorn: six.py:
 - Extension-Anomalie in find_module (Python-Kern)
 - Lösung: Symlink von seahub.py auf seahub.conf in seahub-server/runtime
- named:
 - dnssec-keygen:
 - mit -C! (Kompatibilitätsmodus für Schlüssel ohne Metadaten)
 - Nicht nur Änderung der Metadaten sondern auch des Schlüssels selbst
- Leerzeichen in *.key ist OK!

Fazit

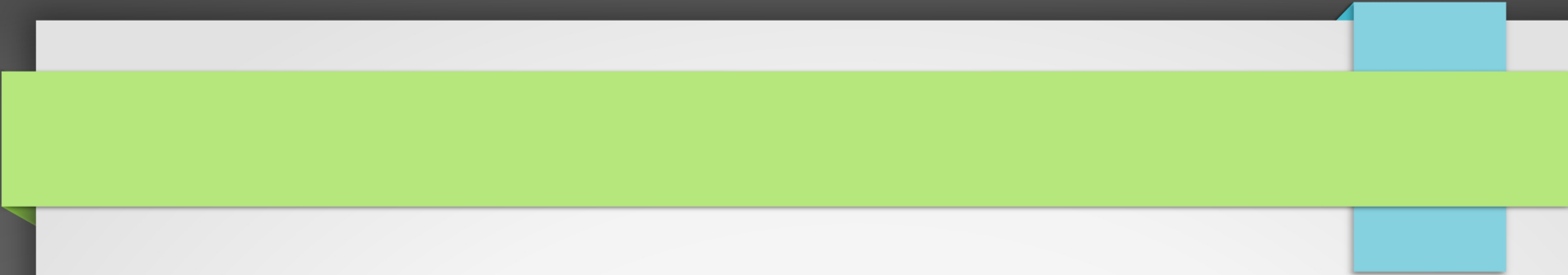
- Volle Kontrolle über Daten
- Unabhängig von Providern
- Geringer H/W-Bedarf
- Einfache Installation sobald Fallstricke umschifft
- Nächste Schritte:
 - Backup: Server-basiert vs. Ordner-Hierarchie



Wir müssen leider draussen bleiben



Demo



Fragen / Diskussion



Backup

nginx.conf (Auszug)

```
http {
    include      mime.types;
    default_type application/octet-stream;

    server {
        listen 443 ssl;
        ssl_certificate /etc/ssl/sfcacert.pem;           # path to your
cacert.pem
        ssl_certificate_key /etc/ssl/sfprivkey.pem;      # path to your
privkey.pem
        server_name subdomain.domain.com;
        ...
    }
    . . .

    location /seafhttp {
        rewrite ^/seafhttp(.*)$ $1 break;
        proxy_pass http://127.0.0.1:8082;
        client_max_body_size 0;
    }
}
```