

Automatische Rotation von DKIM-Schlüsseln

Berlin | 10.05.2014 | Stefan Neben
System Engineer, stefan.neben@immobilienscout24.de

Immobilien Scout GmbH

- Angefangen hat es vor 15 Jahren mit Telefon und Post
- Alle Dienste werden heute ausschließlich über das Internet angeboten:
 - Kontakt mit Kunden über E-Mail
 - ca. 2,5 Millionen ausgehende Mails am Tag

Probleme / Gefahren

Angreifer fälschen Absender:

- Domain kann für Phishing verwendet werden
- Reputationsschaden für die Firma

Aber wie schützen?

- **SMTP** besitzt kein Mechanismus zur Überprüfung des Absenders!

Hier kann DKIM helfen

- Von Yahoo und Cisco 2005 als Entwurf bei der IETF eingereicht:
 - Im Mai 2007 als **RFC 4871** veröffentlicht
- Kryptographische Signierung von E-Mails:
 - Dadurch Schutz vor Manipulation der E-Mail auf dem “Transportweg”

Wie funktioniert DKIM?

Public-Key-Verschlüsselungsverfahren:

- **Private-Key** liegt auf Mail-Server
- **Public-Key** wird als DNS-TXT-Record hinterlegt
- **Signierung** bestimmter **Header-Einträge** und des **Bodies** der E-Mail durch versendenden Mail-Server
- Empfangende Mail-Server validieren die eingehende Mail durch den **Public-Key** im DNS-TXT-Record

Wie funktioniert DKIM?

- Der DNS-TXT-Record wird unter eigenem Selektor abgelegt -> das machen wir uns später zu nutze!

Bsp.: 20140407122301._domainkey.immobilienscout24.de

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed;  
d=immobilienscout24.de; s=20140407122301; t=1396885498;  
bh=fdkeB/A0FkbVP2k4J4pNPoeWH6vqBm9+b0C3OY87Cw8=;  
h=Date:To:Subject:From;  
b=kTfBHQIvrE7/M9ppoNhgSkkTwMruWP/SJO6qXdcdR+tWmdAsdCuqJFo0D/nFCoHrD  
OCwt/K7+aI/pZzCg/hpPbdBZxQ9TSW6IEavaB4qdX7TSSjaBY1/G7BV/Vb+e9c45yh  
qn7kWoLfHbWwtGAezf62oze7gsOaUmn242fyAYUnrdPWojOkn2pKs2jBa3ktvTC7SB  
VfkMGInvLgjRrTMTYum8J66sdyVoWoOdQLDrx1rsFGSF7SxZEmUMFYLRzGI6PP/xV7  
8+2aws8GtL16s3UWrYtAg7NoEo7RkrSn8ziwSFDMuTJOYmTmYe5ZqrRx9WpiEbJdub  
o096QIjw4U1Lg==
```

Validierung ist fehlerhaft, wenn...

- Der **Public-Key** nicht gefunden wird:
 - **Nichterreichbarkeit** des zuständigen **DNS-Servers**
 - Die Überprüfung erfolgt, nachdem ein öffentlicher Schlüssel **entfernt** wurde
- Umschreiben des Header/Body durch Transit-Systeme
- Missbrauch durch Phishing (DMARC, ADSP)

Anwendung heute

- DKIM stellt Absender-Identifikation auf technischer Ebene bereit:
 - Die Signatur selbst bietet keinen Schutz, nur deren Auswertung!
- Wird nicht von allen Mail-Providern eingesetzt
- Eine Garantie auf die flächendeckende Auswertung gibt es daher leider nicht!

Einbindung in Postfix

Unsere Kandidaten:

- **opendkim** über die Milter-Schnittstelle
- **Amavis** über Queue Content Filter

Aus Performance-Gründen (ca. 7/1) haben wir uns für **opendkim** mittels **Milter** entschieden (da unser Setup keinen Spam-Filter bedarf).

Einbindung in Postfix

Das Setup ist denkbar einfach:

OpenDKIM:

```
Domain   immobilienscout24.de
KeyFile  /etc/keys/dkim/immobilienscout24.de.key
InternalHosts 127.0.0.1, 10.0.0.0/8
```

Postfix:

```
smtpd_milters = inet:localhost:8891 -----> Mails vom smtpd
non_smtpd_milters = inet:localhost:8891 ---> Mails von sendmail oder qmqpd
milter_default_action = accept -----> accept, reject,... je nach
Einsatzszenario
```

Motivation zur Rotation

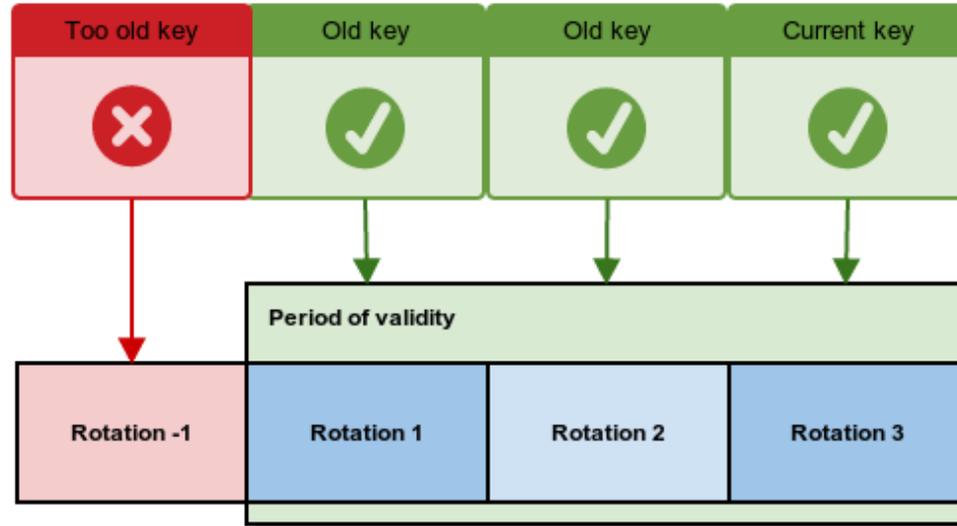
- Mitarbeiterwechsel sind unvermeidlich:
 - Accounts sind schnell deaktiviert
 - **Aber:** Zertifikate, Schlüssel, usw. werden wenn überhaupt nur zögerlich ausgetauscht
- Gelangt der privater Key in falsche Hände, hat dieser nur eine begrenzte Gültigkeit
- Sportsgeist

Umsetzung des Projekts

Strategie zur Umsetzung

- DKIM kann mit mehreren Selektoren umgehen:
 - Macht eine nahtlose Schlüssel-Rotation möglich

Idee:



Erstellen der Schlüssel

opendkim-genkey:

```
$ opendkim-genkey -r -s 20140407122301 -d example.com -b 2048
```

```
20140407122301.private ← privater Schlüssel zum Signieren
```

```
20140407122301.txt ← öffentlicher Schlüssel zur Verifizierung
```

Dieses und die Rotation kann bequem z.B. in ein Bash-Skript realisiert und für die automatische Rotation genutzt werden.

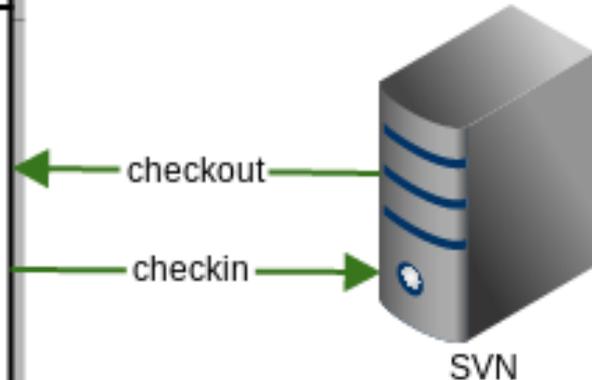
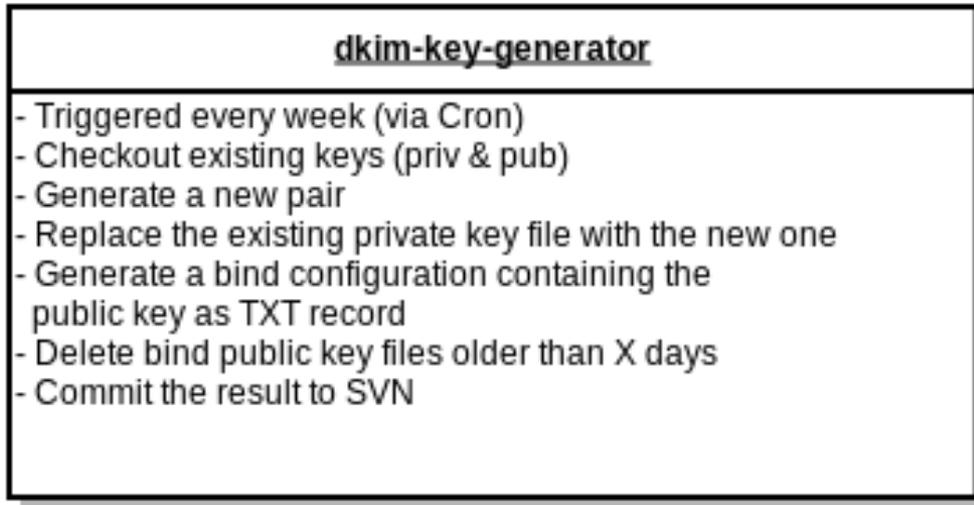
Wir benötigen etwas mehr

- Alles wird packetiert (RPM)!
- Quelle der Wahrheit ist ein zentrales SVN
- Staging erfolgt über Repositories

Daher reicht das vorherige Beispiel nicht aus!

Generiert wird also anders...

durch ein im Projekt entstandenes Python-Skript:

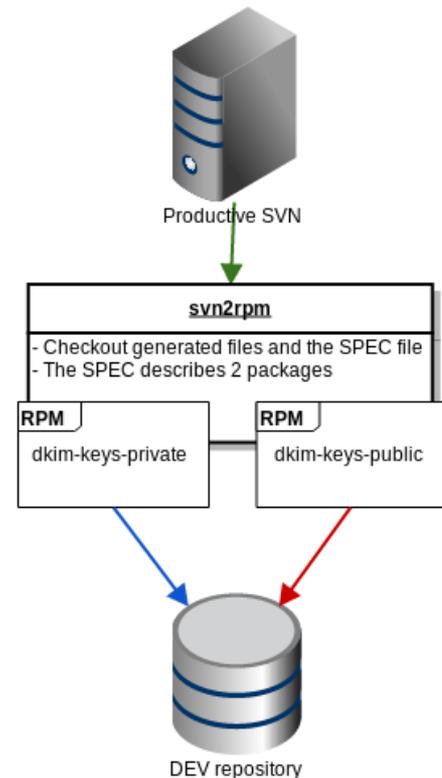


Paketieren der Schlüssel

- Trigger über Post-Commit-Hook
- Ergebnis sind zwei Pakete:
 - dkim-keys-private
 - dkim-keys-public
- Hochladen in Repository

Auf Github:

<https://github.com/immobilienscout24/svn2rpm/>



Fertig?

Wer spielt denn so etwas ungetestet ein?

Stichwort: Test Driven Infrastructure

Weitere Informationen hierzu:

<http://goo.gl/aBv62o>

Wie testen wir?

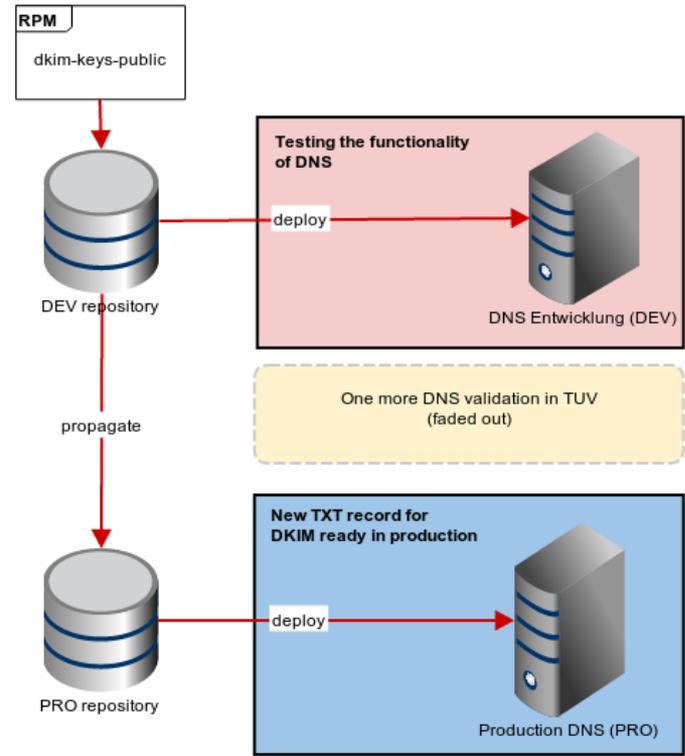
- Basis ist **TeamCity** (Management and Continuous Integration Server):
 - SVN als Quelle
 - Paket-Bau in Mock-Umgebung auf Buildrechner
 - Fertige RPMs werden mittels Repo-Client auf Repository-Server geladen/propagiert
 - Mit Hilfe des Orchestrierungstools **YADT** werden die Pakete deployed

Wie testen wir?

- TeamCity:
 - <http://www.jetbrains.com/teamcity/>
- Repository-Server:
 - <https://github.com/immobilienscout24/yum-repo-server/>
- YADT:
 - <http://www.yadt-project.org/>

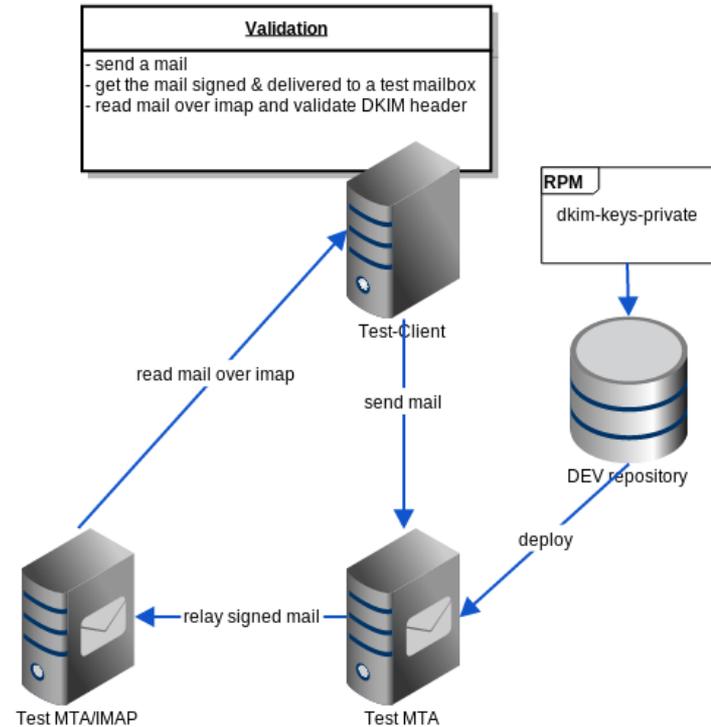
TXT-Record veröffentlichen

- Test-Chain für DNS:
 - In **DEV-Repo** einspielen
 - Auf **Test-DNS-Server** ausrollen
 - **DNS** testen
 - In **PRO-Repo** propagieren
 - Live nehmen!



Testen der neuen Schlüssel

- Test-Chain für DKIM:
 - In **DEV-Repo** einspielen
 - Deployen auf Test-MTA
 - Test-Client sendet Mail
 - MTA signiert diese und leitet sie auf Test-IMAP
 - Test-Client liest die Mail vom Test-IMAP und verifiziert diese



Daten zum Validator

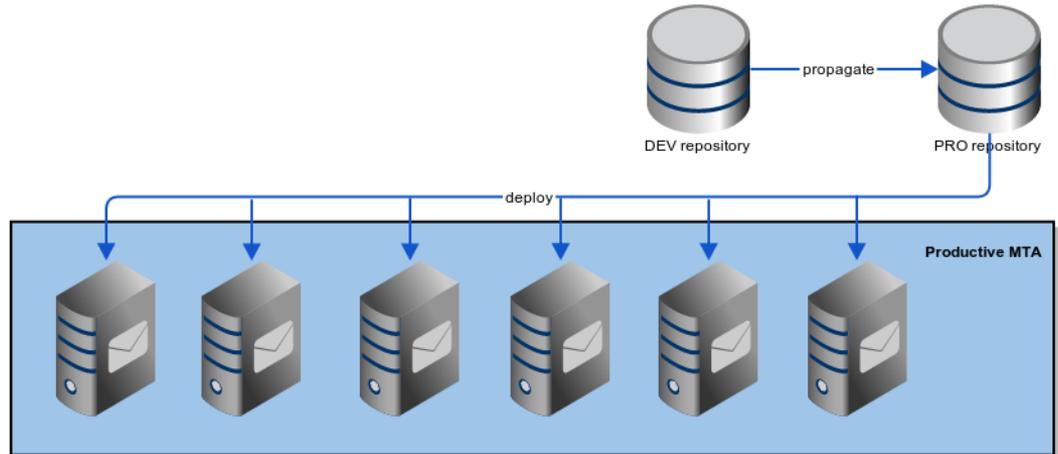
- Python-Skript
- Versendet an zu prüfenden MTA
- Ruft die Mail von Test-IMAP-Server wieder ab & validiert dann die DKIM-Signatur

Auf Github:

<https://github.com/immobilienscout24/mail-validator/>

Ausrollen auf prod. Mail-Server

- Nach Tests zu **PRO-Repo** propagieren
- Deployen auf alle MTAs
- Ab sofort wird mit dem neuen Schlüssel signiert



Fallstricke

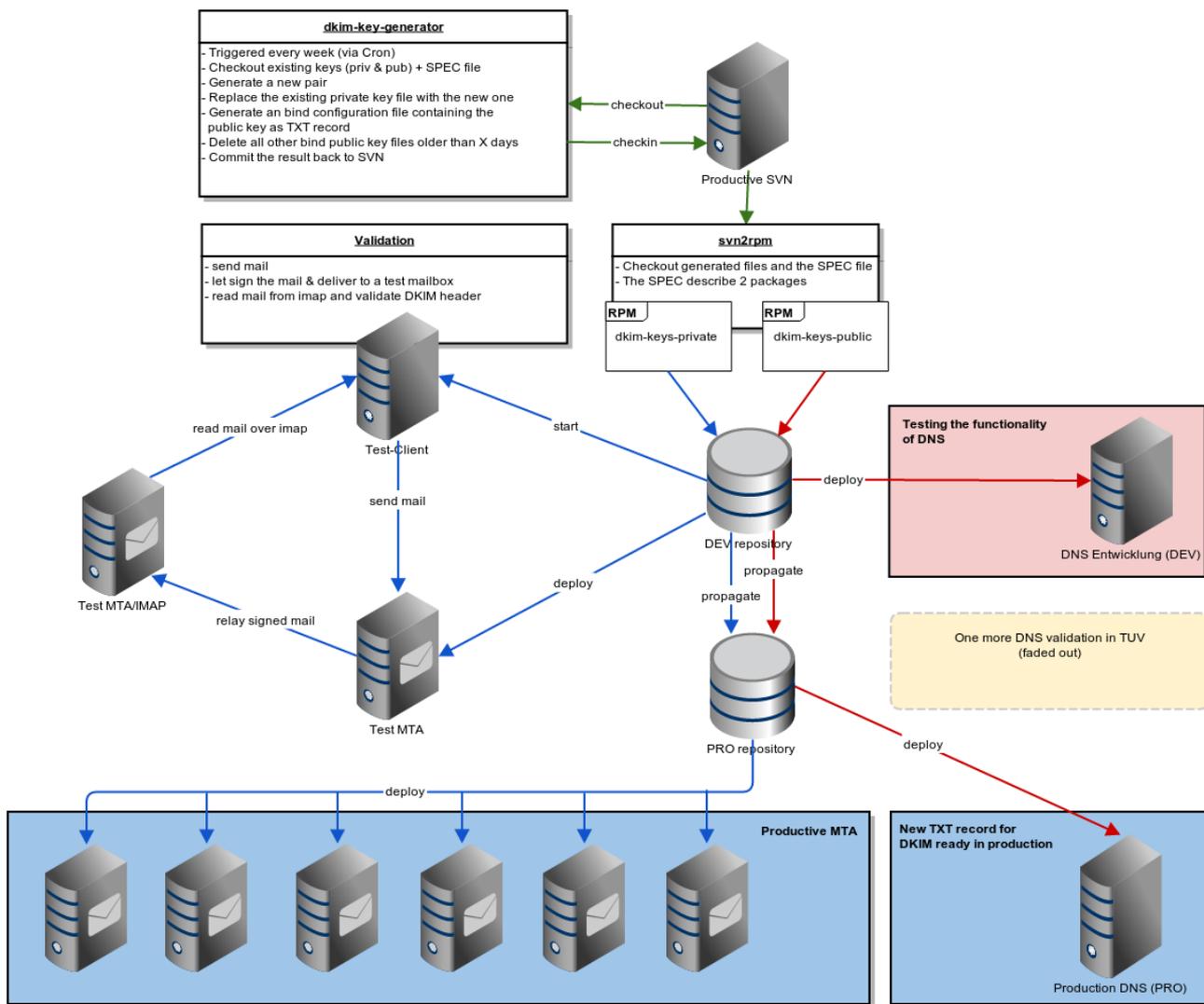
- Mails werden beim Eintreffen signiert:
 - Können also bereits signiert ~3 Tage in der Mail-Queue liegen
 - Daher sollten Schlüssel jünger als 4 Tage nicht herausrotiert werden

Fallstricke

- Mind. 2 öffentliche Schlüssel sind immer notwendig:
 - Es muss beachtet werden, dass die Test-Chain auch seine Zeit benötigt!
- Zu lange Zeilen im Body
 - Postfix setzt bei > 990 Zeichen ein **<CR><LF><SPACE>**

Big Picture

Zusammenspiel der einzelnen Komponenten

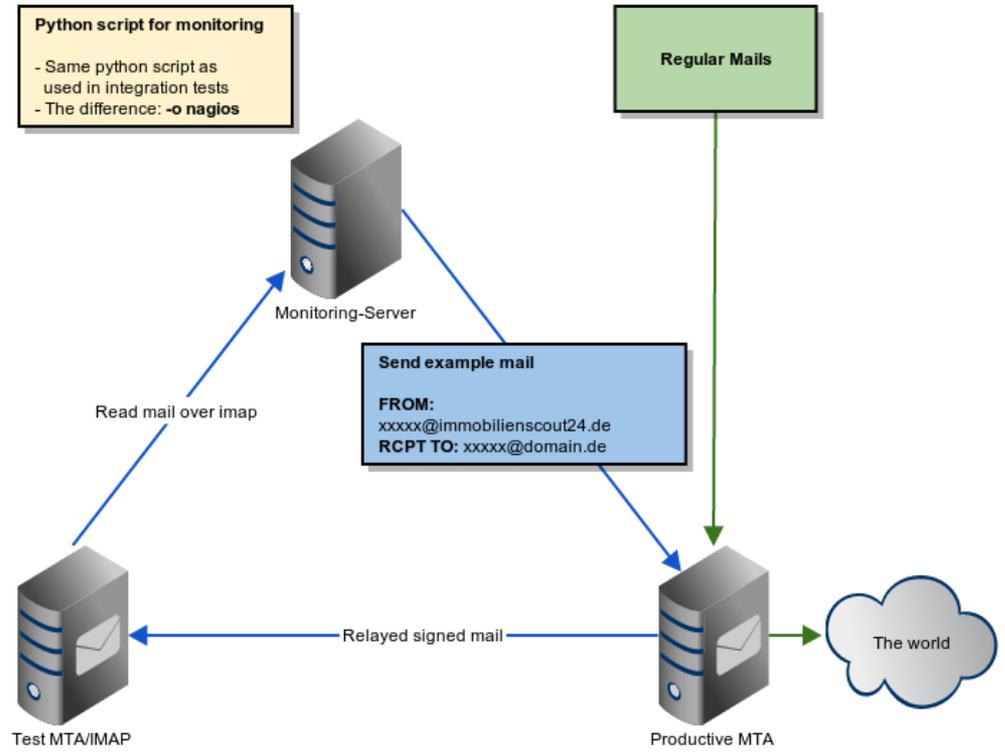


Jetzt Fertig?

Nicht ganz!

Monitoring

- Test-Mail wird vom Monitoring-Server versendet
- MTA signiert und leitet die Mail auf Test-IMAP
- Abruf und Validierung der Mail durch Test-Skript



Sollte die Zeit es erlauben...

noch ein kleiner Tipp:

- Postfix **main.cf** oder **master.cf** durch verschiedene Dateien konfigurieren können?
- Geht nicht?
- Nein... und Ja!

Auf Github:

<http://github.com/sneben/postfix-configurator/>

Fragen?

www.immobilienscout24.de



Vielen Dank für Ihre Aufmerksamkeit!

Berlin | 10.05.2014 | Stefan Neben
System Engineer, stefan.neben@immobilienscout24.de