

Systemarchitektur für verteilte, mobile Anwendungen

Anton Gillert
11056249

Inhalt

- Motivation
- Netzebene
 - Problemstellung
 - VMA – Elemente
 - NAT, STUN und TURN
- Softwareebene
 - Problemstellung
 - Softwarearchitekturstil REST
 - Beispieldienst

Motivation

In den letzten Jahren hat die Rechenleistung und Popularität von Smartphones rapide zugenommen. Die neuste Generation bietet folgende Funktionalität:

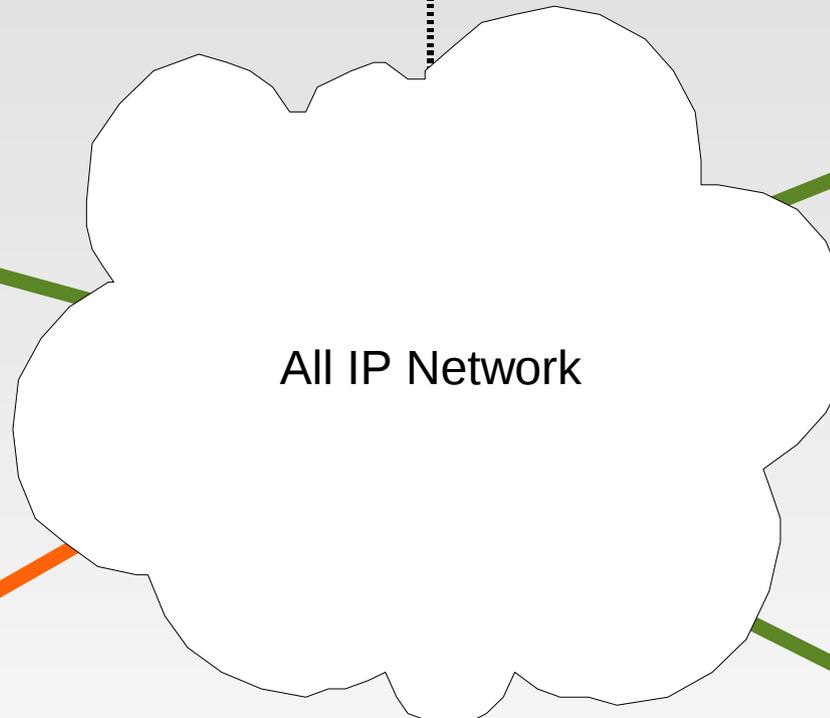
- Fotofunktionalität
- Video-/Audio-Aufnahme und Wiedergabe
 - Unterstützung von gängigen Streaming-Formaten
- Lokationsdienste (GPS)
- Mobiles Internet (GPRS, UMTS...)

Diese Funktionen sollen kombiniert als P2P-Dienste von den Smartphones angeboten werden. (Zwei laufende Bachelorarbeiten implementieren u.a. einen P2P-Chat-service mit zusätzlichen Funktionen.)

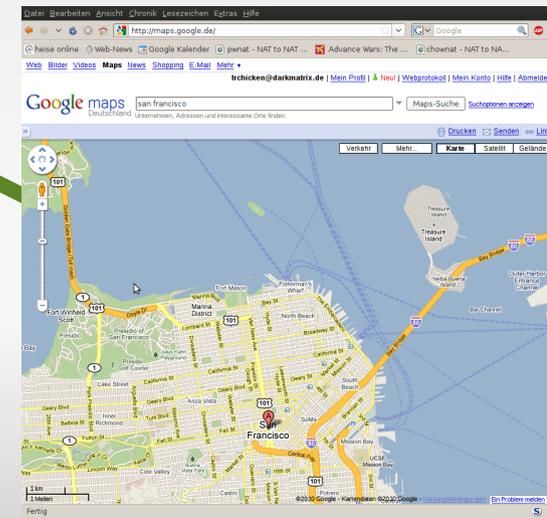
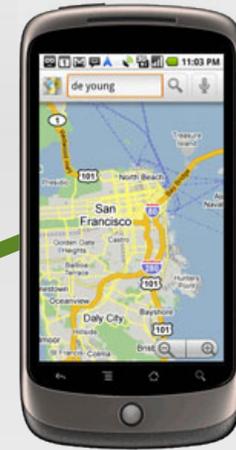
Motivation

Dienstanbieter

Dienstkonsumenten



All IP Network



Problematik - Netzwerkebene

Smartphones oder allgemein mobile Endgeräte, agieren in einer sehr heterogenen Umgebung. Die Netzanbindung erfolgt von Provider zu Provider unterschiedlich. Dabei treten für Dienstanbieter folgende Probleme auf:

- Keine festen IPs.
- Zum Teil sehr niedrige Leasing-Zeiten der IPs.
- Firewalls und NATs lassen keine eingehende Verbindungen zu.

VMA Netzelemente

Um diesen Problemen entgegenzutreten und um eine managebare Infrastruktur zu schaffen, wurden in der *Systemarchitektur für verteilte, mobile Anwendungen* folgende Elemente eingeführt:

- VMA Registrar

Registrierungs- und Verwaltungszentrale (AAA). Bezugspunkt für den Bootstrap der Smartphone-Applikation.

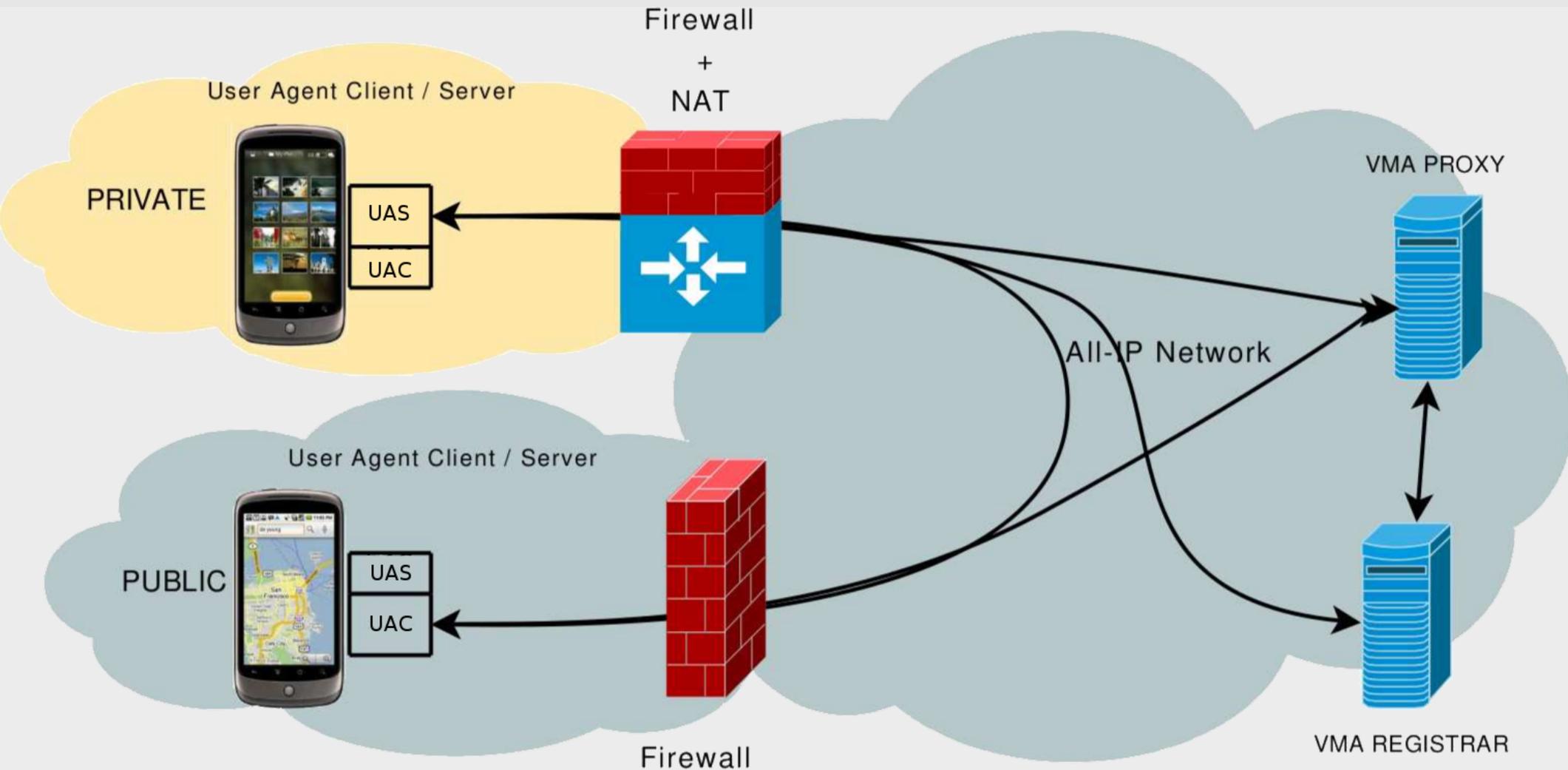
- VMA Proxy

Ermöglicht Verbindungen durch NATs bzw. Firewalls.

- VMA User Agent (Smartphone-Applikation)

- User Agent Server (**UAS**) Dienstanbieter
- User Agent Client (**UAC**) Dienstkonsument

VMA Netzinfrastruktur



VMA Registrar

Aufgaben des Registrars:

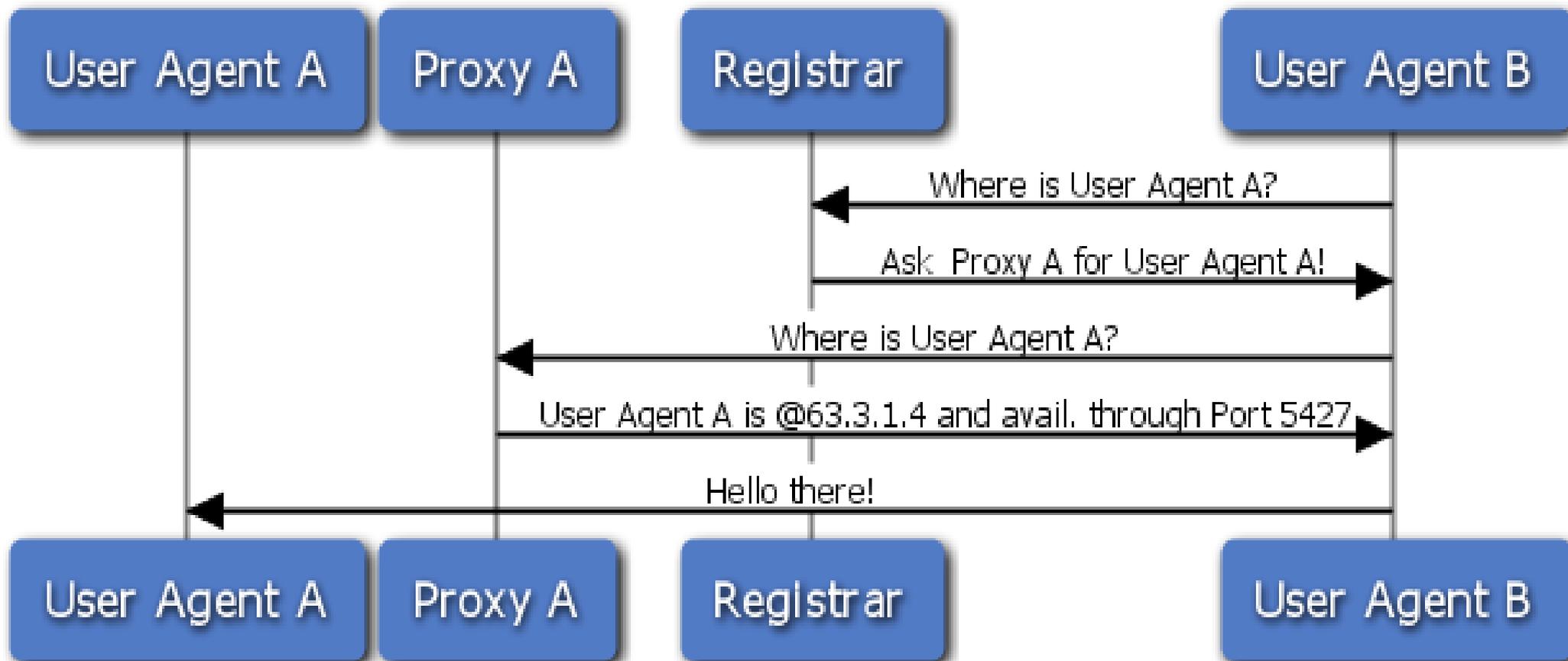
- Zugangspunkt des VMA-Netzes.
- Nutzer- und Diensteverwaltung (Datenbank).
- NAT - Indikator nach RFC3489 STUN.

Adressierungsschema (SIP/URI)

- Permanente Adresse:
mustermann@vma-services.com
- Temporäre Adresse:
mustermann@musterproxy.net
mustermann@139.16.9.123:4125

Adressierungsschema

Die Adressauflösung im VMA-Netz:



VMA Proxy und User Agent Server

Aufgaben des Proxys:

- Initialisierung/Unterstützung von Verbindungen.
- Hilfe beim Überwinden von Firewalls und NATs. (Aufgaben des Proxys:STUN/TURN).
- Führen/Cachen von Nutzerlisten.

Aufgaben des User Agent Servers:

- Kommunikation mit dem Proxy.
- Bereitstellen der Dienste über TCP/UDP.

NAT - Typen

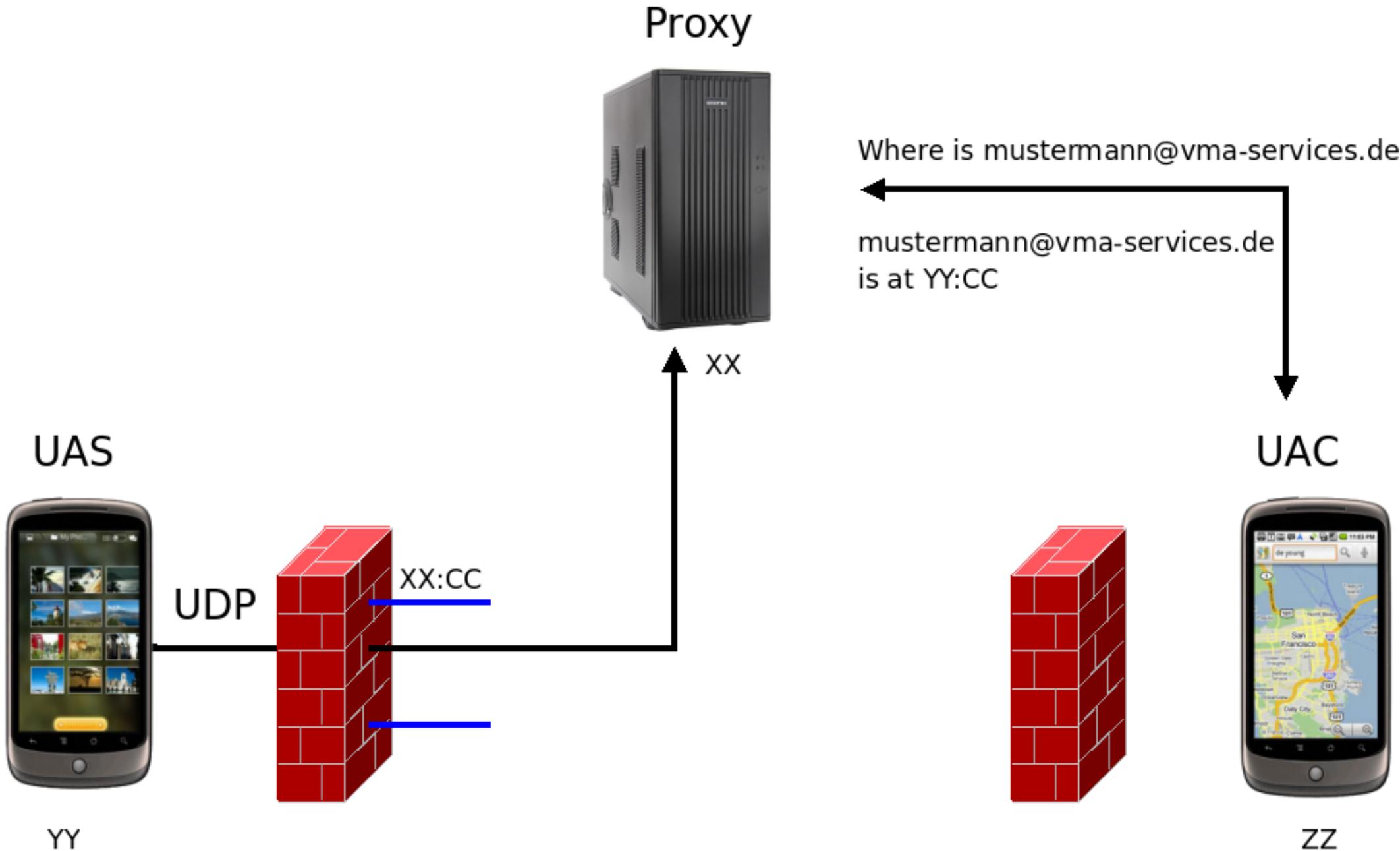
Verbindungsprobleme werden hauptsächlich durch den Einsatz von NAT hervorgerufen:

- Full Cone NAT
 - Restricted Cone NAT
 - Port Restricted Cone NAT
- 
- STUN**
Simple Traversal of UDP
Through NAT
- Symmetric NAT
- 
- TURN**
Traversal Using Relays
around NAT

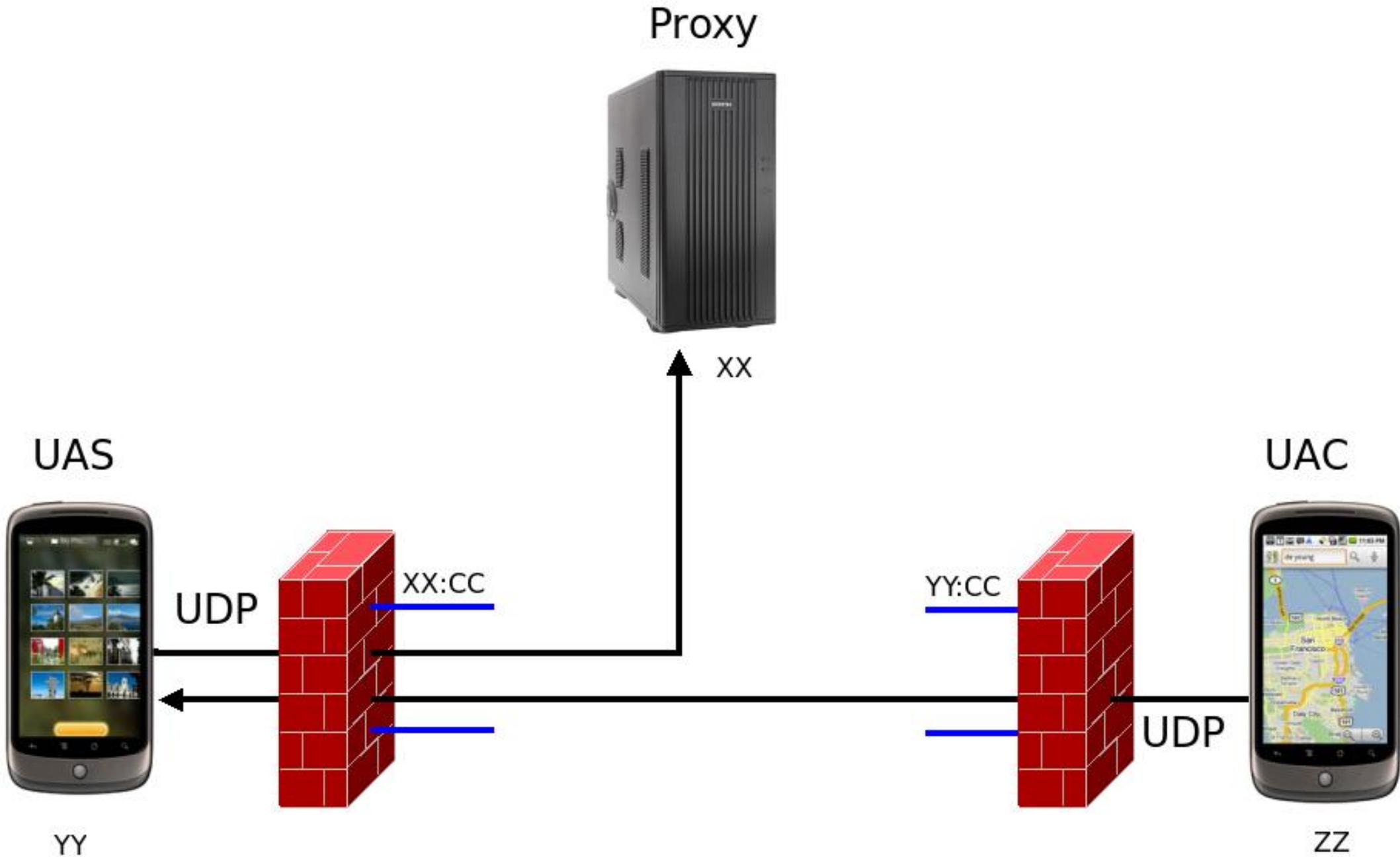
STUN ermöglicht durch o.g. NATs **P2P**-Verbindungen über **UDP**.

TURN ermöglicht durch alle NATs Verbindungen über **TCP** aber **keine P2P**-Verbindungen.

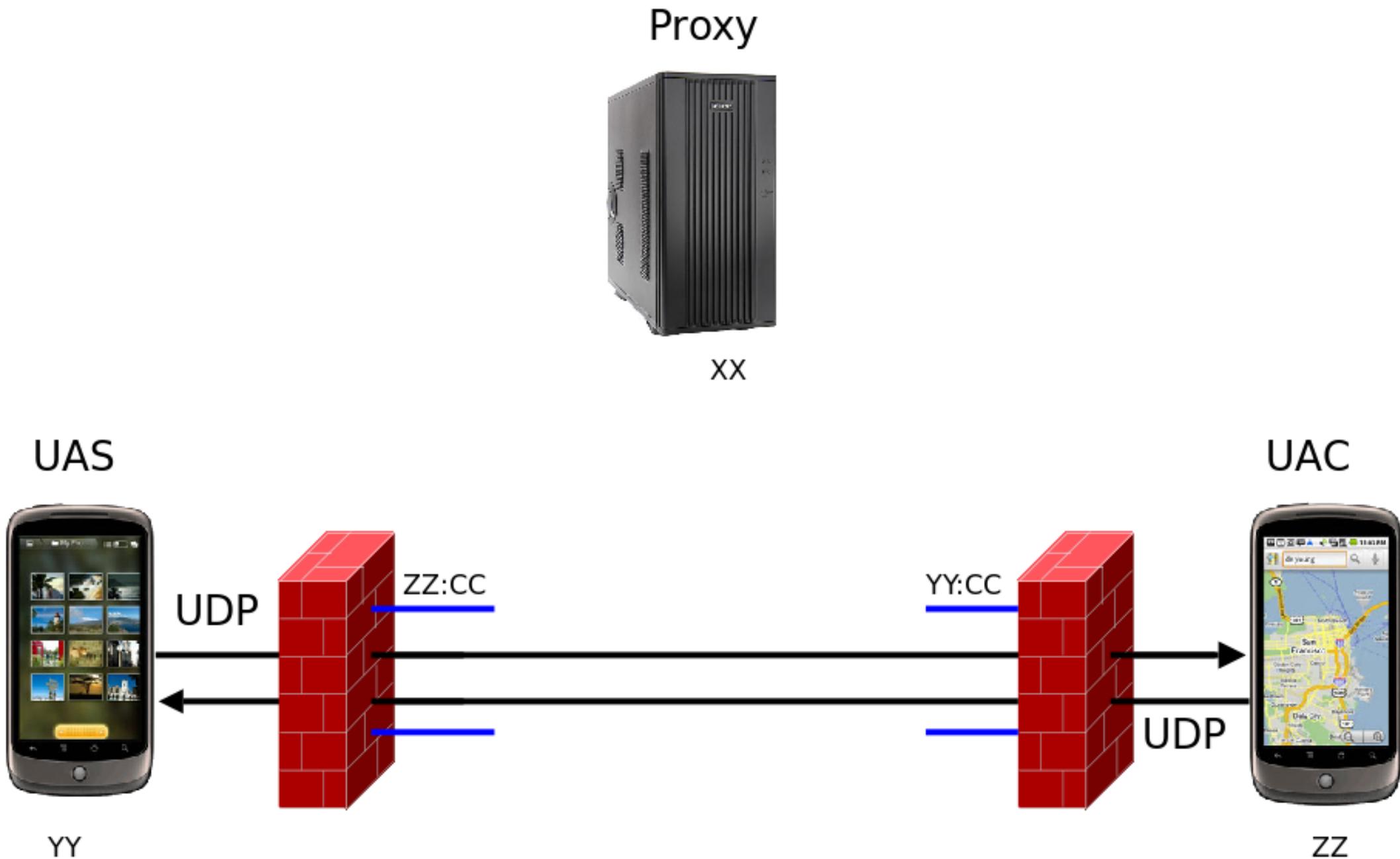
Verbindungsaufbau STUN 1



Verbindungsaufbau STUN 2



Verbindungsaufbau STUN 3



Verbindungsaufbau TURN

User Agent Server

VMA Proxy

User Agent Client



UAS

TURN Relay

UAC

TCP

Der Proxy dient als Relay, durch das sämtlicher Verkehr geleitet wird.

Verbindungsarten



Softwarearchitektur

Betrachtet man die aktuelle Smartphone-Generation fällt auf, dass die Plattformen bezüglich der verwendeten Technologien stark divergieren:

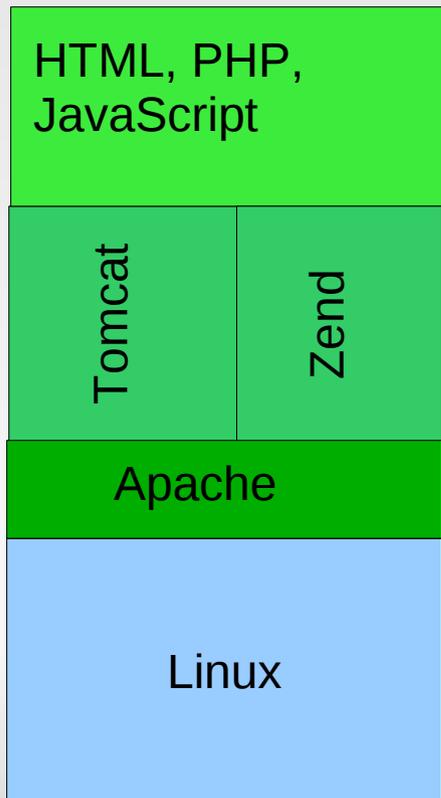
Hersteller – Betriebssystem – Programmiersprache

- Nokia – Symbian, maemo – C++, Java ME
- Apple – iPhone OS – Objective C
- HTC – Android – Java
- Palm – WebOS – JavaScript
- BlackBerry – BlackBerryOS – Java ME

Softwarearchitektur 1

Äußerst heterogener Softwarestack auf den Smartphones.
Gesucht wurden ein geeignetes Datenaustauschformat und ein geeigneter Softwarearchitekturstil.

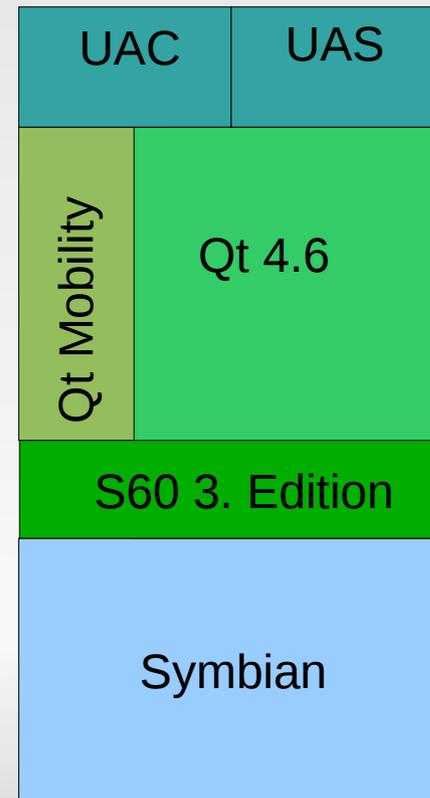
VMA Registrar



G1 - HTC



5800 - Nokia



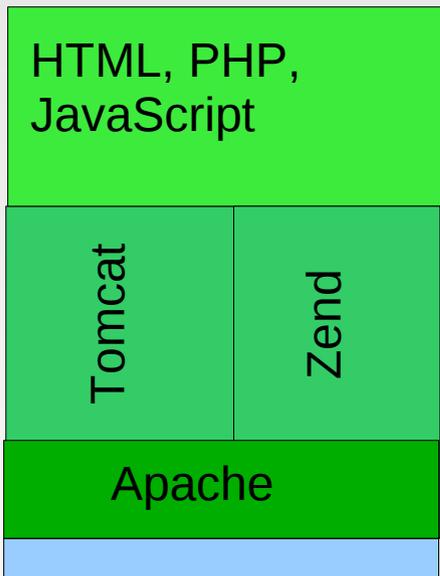
Web-Client



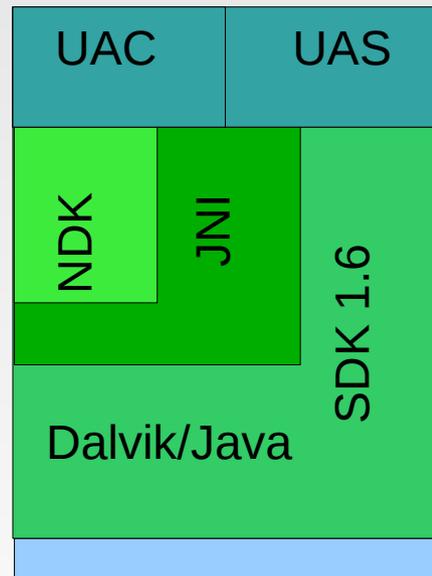
Softwarearchitektur 2

REST (Representational State Transfer):
Grundlage für die Dienstbeschreibung und den Datenaustausch.

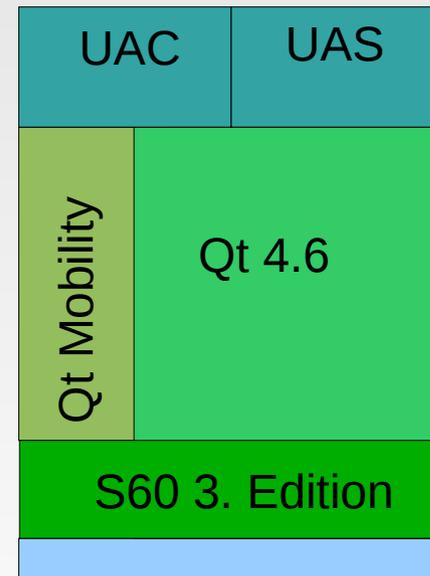
VMA Registrar



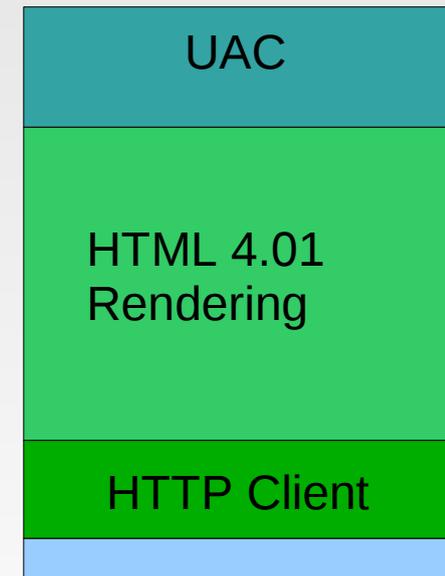
G1 - HTC



5800 - Nokia



Web-Client



REST
HTTP

HTML
JSON

REST - Representational State Transfer

REST bietet zwei bestechende Vorteile gegenüber anderen Architekturen:

- Sehr wenig Overhead.
- Datenaustauschformat ist frei wählbar und hohe Verbreitung des HTTP Protokolls.

Im VMA-Kontext wird REST mit folgenden Datenaustauschformaten Kombiniert:

- JSON – Anwendung zu Anwendung Kom.
- HTML – Anwendung zu Benutzer Kommunikation

REST – Beispieldienst

Kameradienst

<http://mustermann@139.16.9.22/services/camera>

Liefert die Einstellungen der Kamera.

<http://mustermann@139.16.9.22/services/camera/snapshot>

Löst die Aufnahme eines Fotos aus.

<http://mustermann@139.16.9.22/services/camera/video>

Löst einen Video-Stream aus.

REST – Beispieldienst

Anwendung zu Anwendung-Kommunikation

mustermann@139.16.9.22/services/camera

HTTP Request von einer nativen UAC-Anwendung:

```
GET /services/camera HTTP/1.1
```

```
Host: 139.16.9.22
```

```
Accept: text/json
```

HTTP Response des User Agent Servers:

```
HTTP/1.1 200 OK
```

```
Server: VMA User Agent Server
```

```
Content-Type: text/json
```

```
{  "camera": {
    "snapshot": {
      "res_avail": [ 640x480, 800x600, 1024x768],
      "res_used"="640x480"
    }
  }
}
```

REST – Beispieldienst

Anwendung zu Benutzer-Kommunikation

mustermann@139.16.9.22/services/camera

HTTP Request von einem Browser oder UAC:

```
GET /services/camera HTTP/1.1
```

```
Host: 139.16.9.22
```

```
Accept: text/html,application/xhtml+xml
```

HTTP Response des User Agent Servers:

```
HTTP/1.1 200 OK
```

```
Server: VMA User Agent Server
```

```
Content-Type: text/html
```

```
<html> ...
```

```
<h1>Camera Service of mustermann</h1>
```

```
<form><p><select size="3">
```

```
<option selected> 640x480 </option>
```

```
<option> 800x600 </option>
```

```
<option> 1024x768 </option>
```

```
</select></p></form>...<html>
```

Zugriffsmöglichkeiten durch VMA-Systemarchitektur

User Agent Server



UAS

VMA Proxy



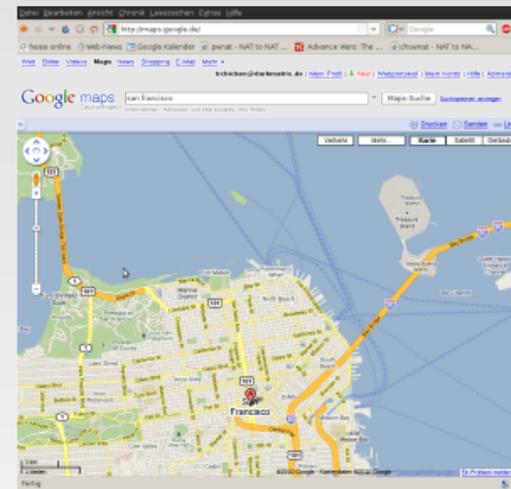
TURN Relay

User Agent Client



UAC

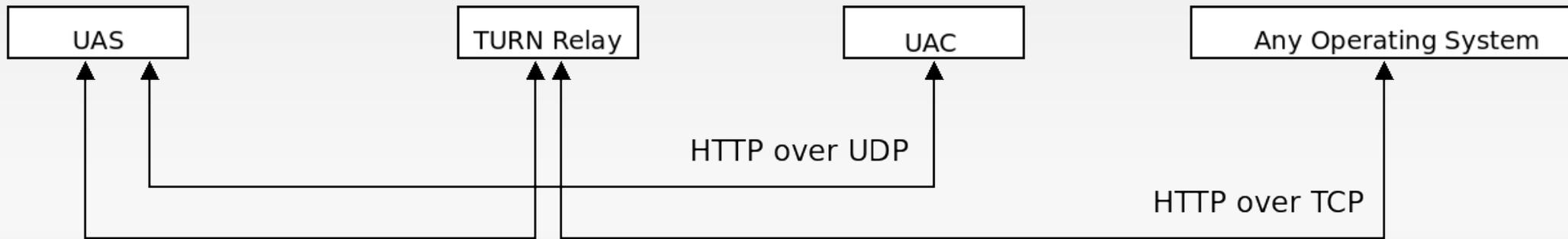
Webbrowser



Any Operating System

HTTP over UDP

HTTP over TCP



Quellen

- [RFC2396] IETF **URI - Uniform Resource Identifiers: Generic Syntax** . Zuletzt abgerufen: März 2010.
<http://www.ietf.org/rfc/rfc2396.txt>
- [RFC3261] IETF **SIP - Session Initiation Protocol** . Zuletzt abgerufen: März 2010. <http://tools.ietf.org/html/rfc3261>
- [RFC3489] IETF **STUN - Session Traversal Utilities for NAT** . Zuletzt abgerufen: März 2010 <http://tools.ietf.org/html/rfc3489>
- [RFC5766] IETF **TURN - Traversal Using Relays around NAT**. Zuletzt abgerufen: März 2010. <http://tools.ietf.org/html/draft-ietf-behave-turn-16>
- Stefan Tilkov **REST und HTTP: Einsatz der Architektur des Web für Integrationsszenarien**. dpunkt Verlag 2009

Vielen Dank für Ihre Aufmerksamkeit.

Titel durch Klicken hinzufügen

Systemarchitektur für verteilte, mobile Anwendungen

Anton Gillert
11056249

Inhalt

- Motivation
- Netzebene
 - Problemstellung
 - VMA – Elemente
 - NAT, STUN und TURN
- Softwareebene
 - Problemstellung
 - Softwarearchitekturstil REST
 - Beispieldienst

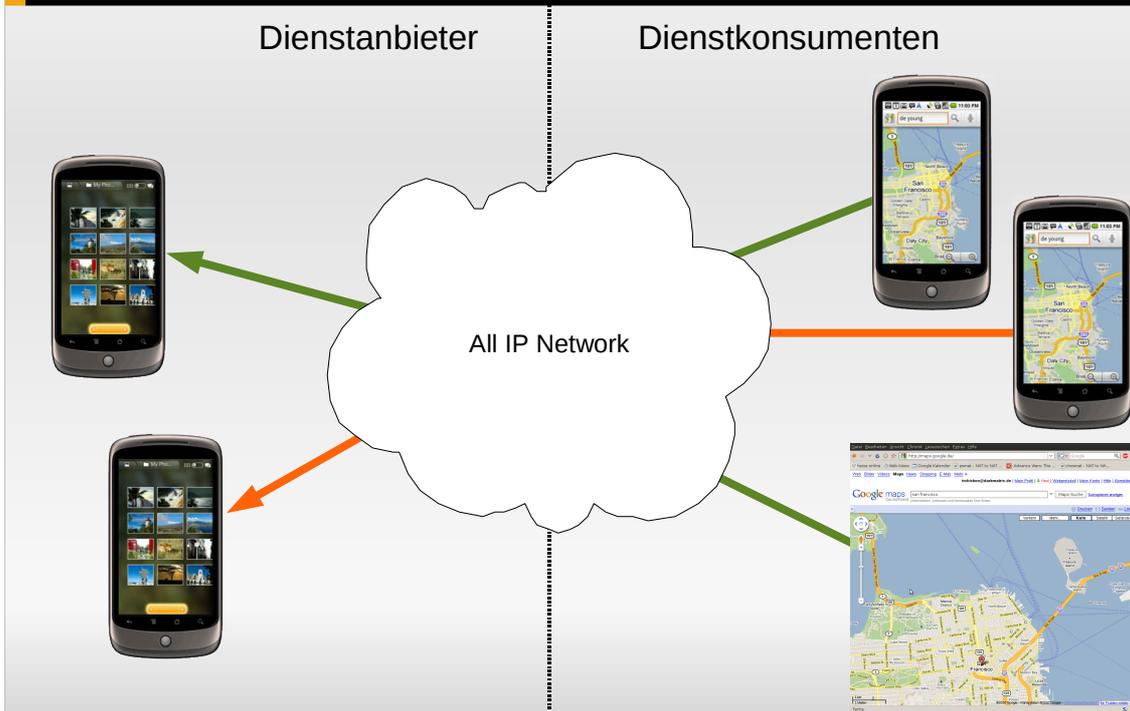
Motivation

In den letzten Jahren hat die Rechenleistung und Popularität von Smartphones rapide zugenommen. Die neueste Generation bietet folgende Funktionalität:

- Fotofunktionalität
- Video-/Audio-Aufnahme und Wiedergabe
 - Unterstützung von gängigen Streaming-Formaten
- Lokationsdienste (GPS)
- Mobiles Internet (GPRS, UMTS...)

Diese Funktionen sollen kombiniert als P2P-Dienste von den Smartphones angeboten werden. (Zwei laufende Bachelorarbeiten implementieren u.a. einen P2P-Chat-service mit zusätzlichen Funktionen.)

Motivation



Problematik - Netzwerkebene

Smartphones oder allgemein mobile Endgeräte, agieren in einer sehr heterogenen Umgebung. Die Netzanbindung erfolgt von Provider zu Provider unterschiedlich. Dabei treten für Dienst-anbieter folgende Probleme auf:

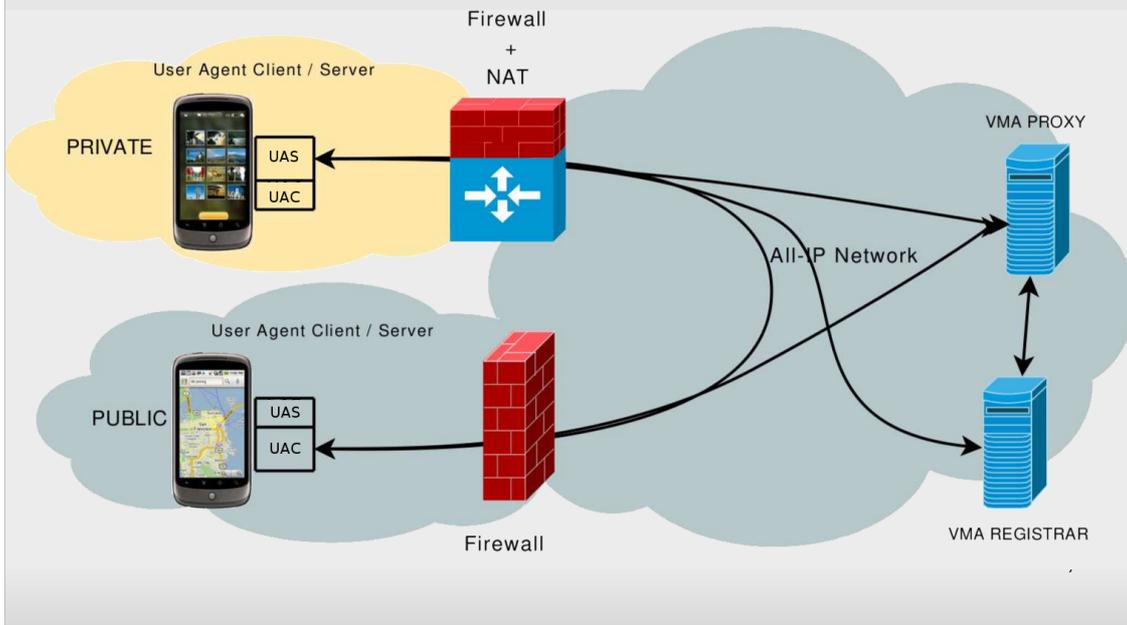
- Keine festen IPs.
- Zum Teil sehr niedrige Leasing-Zeiten der IPs.
- Firewalls und NATs lassen keine eingehende Verbindungen zu.

VMA Netzelemente

Um diesen Problemen entgegenzutreten und um eine managebare Infrastruktur zu schaffen, wurden in der *Systemarchitektur für verteilte, mobile Anwendungen* folgende Elemente eingeführt:

- **VMA Registrar**
 - Registrierungs- und Verwaltungszentrale (AAA). Bezugspunkt für den Bootstrap der Smartphone-Applikation.
- **VMA Proxy**
 - Ermöglicht Verbindungen durch NATs bzw. Firewalls.
- **VMA User Agent (Smartphone-Applikation)**
 - User Agent Server (**UAS**) Dienstanbieter
 - User Agent Client (**UAC**) Dienstkonsument

VMA Netzinfrastruktur



VMA Registrar

Aufgaben des Registrars:

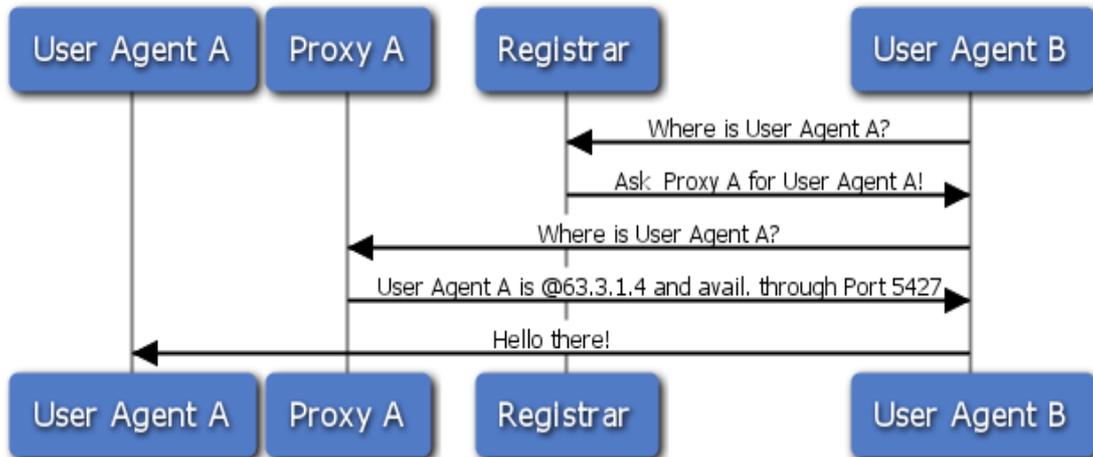
- Zugangspunkt des VMA-Netzes.
- Nutzer- und Dienstverwaltung (Datenbank).
- NAT - Indikator nach RFC3489 STUN.

Adressierungsschema (SIP/URI)

- Permanente Adresse:
mustermann@vma-services.com
- Temporäre Adresse:
mustermann@musterproxy.net
mustermann@139.16.9.123:4125

Adressierungsschema

Die Adressauflösung im VMA-Netz:



VMA Proxy und User Agent Server

Aufgaben des Proxys:

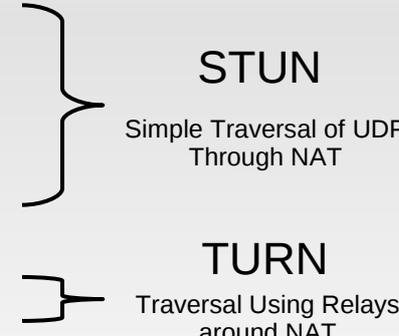
- Initialisierung/Unterstützung von Verbindungen.
- Hilfe beim Überwinden von Firewalls und NATs. (Aufgaben des Proxys:STUN/TURN).
- Führen/Cachen von Nutzerlisten.

Aufgaben des User Agent Servers:

- Kommunikation mit dem Proxy.
- Bereitstellen der Dienste über TCP/UDP.

NAT - Typen

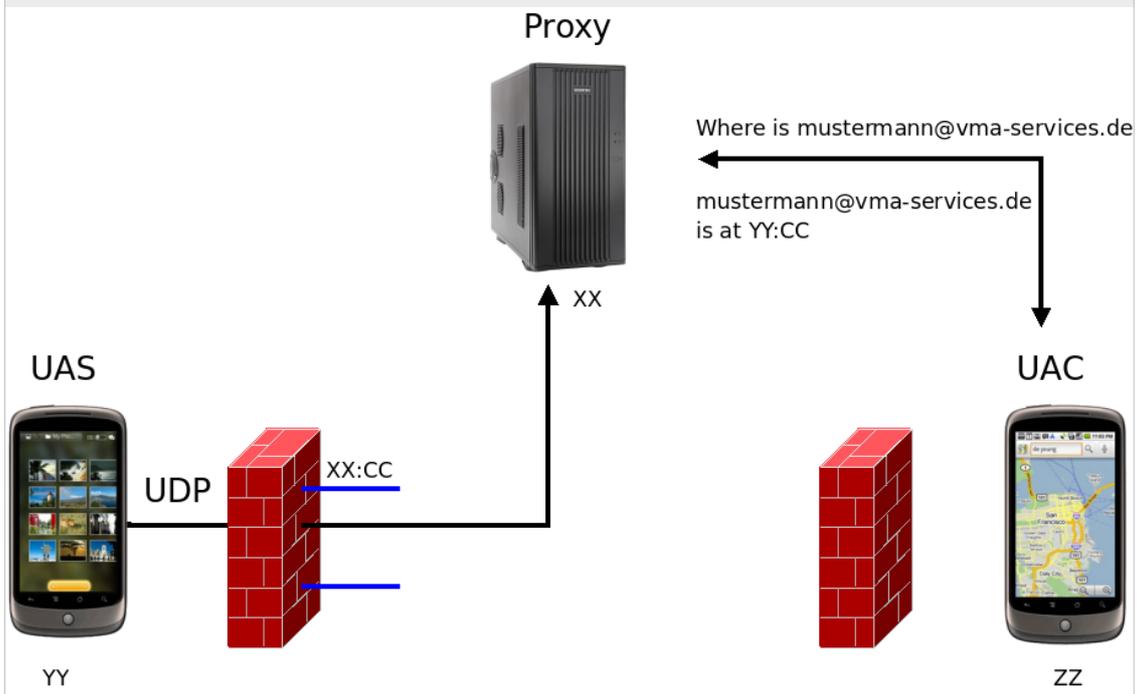
Verbindungsprobleme werden hauptsächlich durch den Einsatz von NAT hervorgerufen:

- Full Cone NAT
 - Restricted Cone NAT
 - Port Restricted Cone NAT
- 
- STUN**
Simple Traversal of UDP
Through NAT
- TURN**
Traversal Using Relays
around NAT

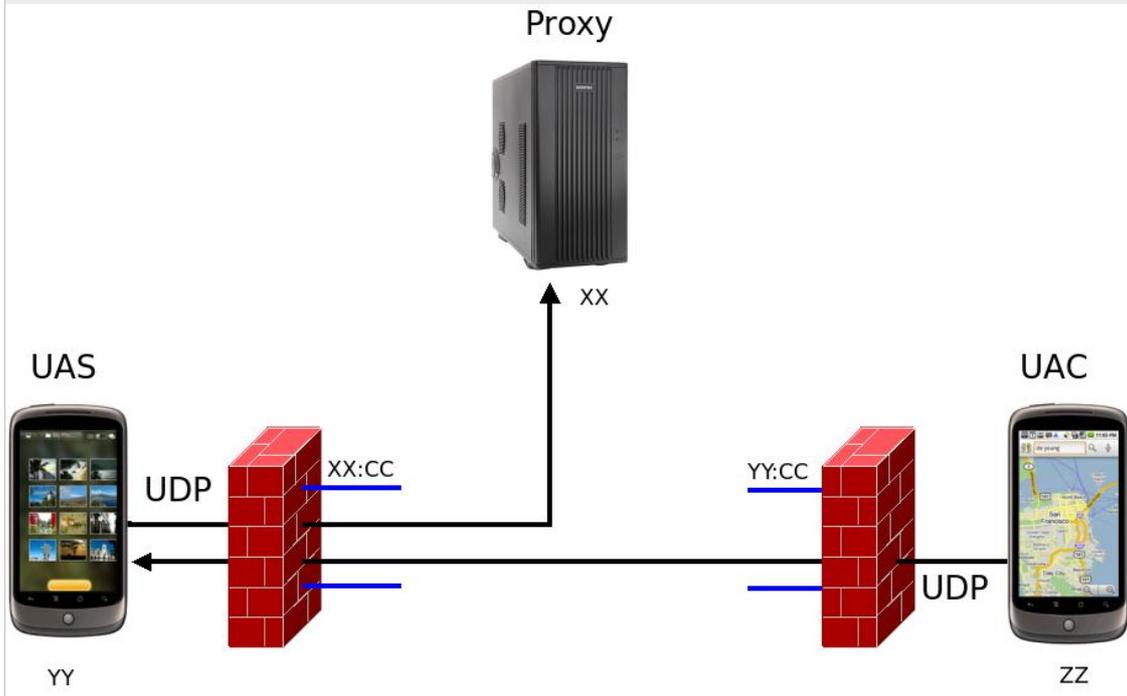
STUN ermöglicht durch o.g. NATs **P2P**-Verbindungen über **UDP**.

TURN ermöglicht durch alle NATs Verbindungen über **TCP** aber **keine P2P**-Verbindungen.

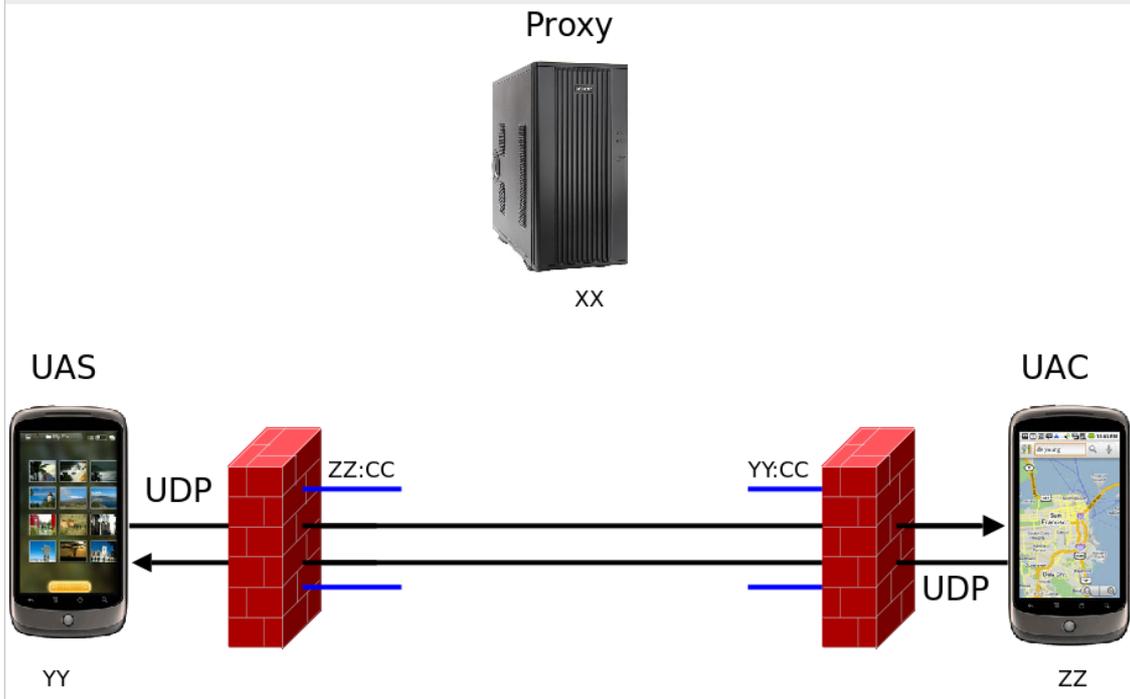
Verbindungsaufbau STUN 1



Verbindungsaufbau STUN 2



Verbindungsaufbau STUN 3



Verbindungsaufbau TURN

User Agent Server



UAS

VMA Proxy



TURN Relay

User Agent Client



UAC

TCP

Der Proxy dient als Relay, durch das sämtlicher Verkehr geleitet wird.

Verbindungsarten



Softwarearchitektur

Betrachtet man die aktuelle Smartphone-Generation fällt auf, dass die Plattformen bezüglich der verwendeten Technologien stark divergieren:

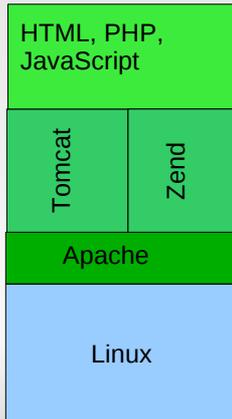
Hersteller – Betriebsys. – Programmiersprache

- Nokia – Symbian, maemo – C++, Java ME
- Apple – iPhone OS – Objective C
- HTC – Android – Java
- Palm – WebOS – JavaScript
- BlackBerry – BlackBerryOS – Java ME

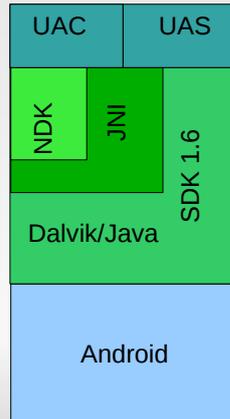
Softwarearchitektur 1

Äußerst heterogener Softwarestack auf den Smartphones.
Gesucht wurden ein geeignetes Datenaustauschformat und
ein geeigneter Softwarearchitekturstil.

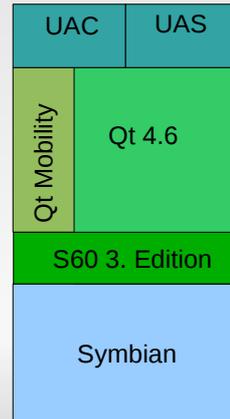
VMA Registrar



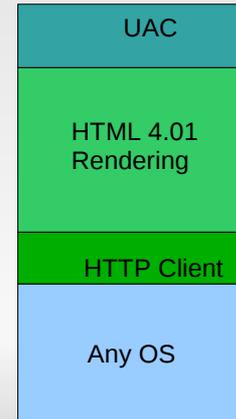
G1 - HTC



5800 - Nokia

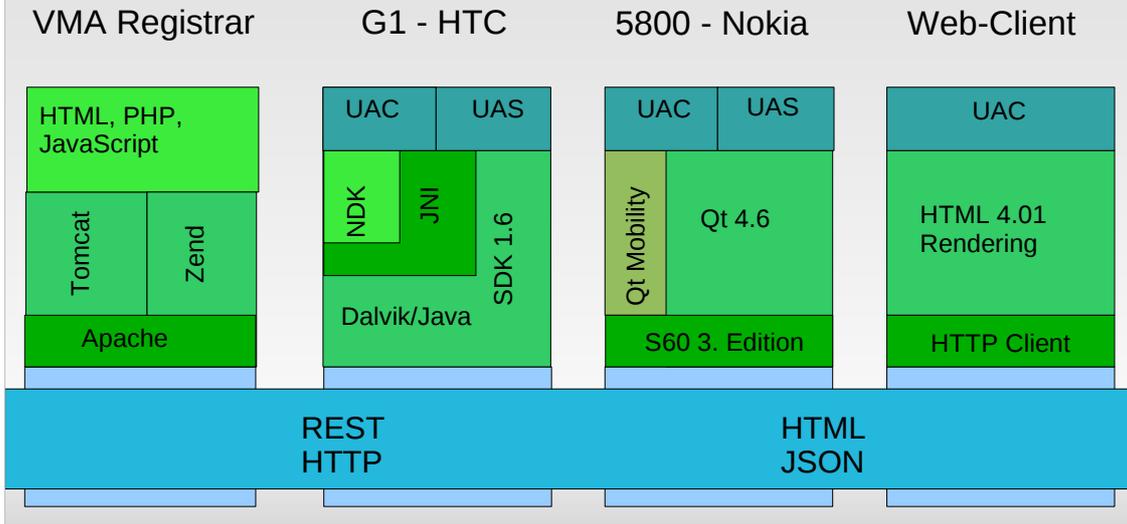


Web-Client



Softwarearchitektur 2

REST (Representational State Transfer):
Grundlage für die Dienstbeschreibung und den Datenaustausch.



REST - Representational State Transfer

REST bietet zwei bestechende Vorteile gegenüber anderen Architekturen:

- Sehr wenig Overhead.
- Datenaustauschformat ist frei wählbar und hohe Verbreitung des HTTP Protokolls.

Im VMA-Kontext wird REST mit folgenden Datenaustauschformaten Kombiniert:

- JSON – Anwendung zu Anwendung Kom.
- HTML – Anwendung zu Benutzer Kommunikation

REST – Beispieldienst

Kameradienst

<http://mustermann@139.16.9.22/services/camera>

Liefert die Einstellungen der Kamera.

<http://mustermann@139.16.9.22/services/camera/snapshot>

Löst die Aufnahme eines Fotos aus.

<http://mustermann@139.16.9.22/services/camera/video>

Löst einen Video-Stream aus.

REST – Beispieldienst

Anwendung zu Anwendung-Kommunikation

mustermann@139.16.9.22/services/camera

HTTP Request von einer nativen UAC-Anwendung:

```
GET /services/camera HTTP/1.1
```

```
Host: 139.16.9.22
```

```
Accept: text/json
```

HTTP Response des User Agent Servers:

```
HTTP/1.1 200 OK
```

```
Server: VMA User Agent Server
```

```
Content-Type: text/json
```

```
{  "camera": {  
    "snapshot": {  
        "res_avail": [ 640x480, 800x600, 1024x768],  
        "res_used"="640x480"  
    }  
  }  
}
```

REST – Beispieldienst

Anwendung zu Benutzer-Kommunikation

mustermann@139.16.9.22/services/camera

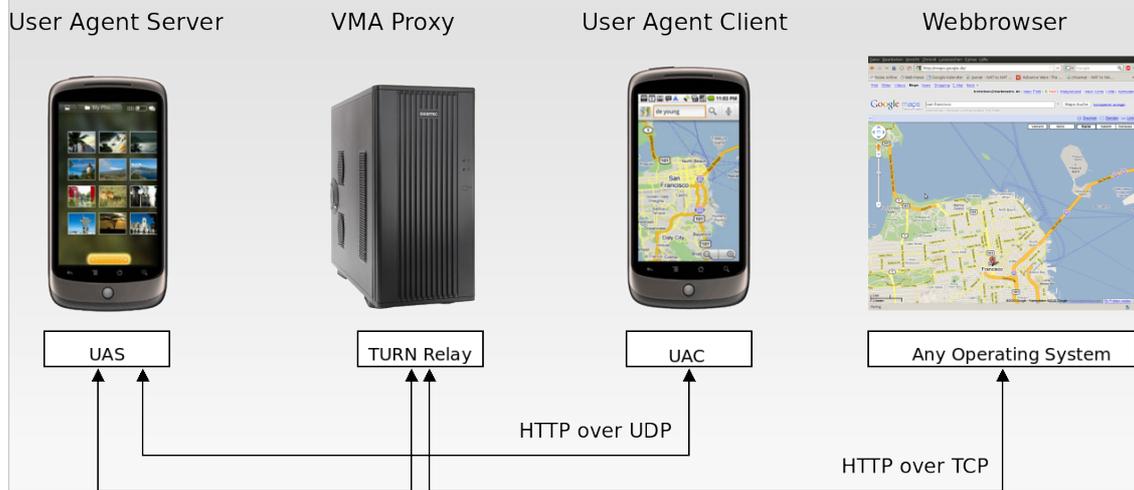
HTTP Request von einem Browser oder UAC:

```
GET /services/camera HTTP/1.1
Host: 139.16.9.22
Accept: text/html,application/xhtml+xml
```

HTTP Response des User Agent Servers:

```
HTTP/1.1 200 OK
Server: VMA User Agent Server
Content-Type: text/html
<html> ...
<h1>Camera Service of mustermann</h1>
<form><p><select size="3">
  <option selected> 640x480 </option>
  <option> 800x600 </option>
  <option> 1024x768 </option>
</select></p></form>...<html>
```

Zugriffsmöglichkeiten durch VMA-Systemarchitektur



Quellen

- [RFC2396] IETF **URI - Uniform Resource Identifiers: Generic Syntax** . Zuletzt abgerufen: März 2010. <http://www.ietf.org/rfc/rfc2396.txt>
- [RFC3261] IETF **SIP - Session Initiation Protocol** . Zuletzt abgerufen: März 2010. <http://tools.ietf.org/html/rfc3261>
- [RFC3489] IETF **STUN - Session Traversal Utilities for NAT** . Zuletzt abgerufen: März 2010 <http://tools.ietf.org/html/rfc3489>
- [RFC5766] IETF **TURN - Traversal Using Relays around NAT**. Zuletzt abgerufen: März 2010. <http://tools.ietf.org/html/draft-ietf-behave-turn-16>
- Stefan Tilkov **REST und HTTP: Einsatz der Architektur des Web für Integrationsszenarien**. dpunkt Verlag 2009

Titel durch Klicken hinzufügen

Vielen Dank für Ihre Aufmerksamkeit.