

# Anforderungsprofil BuzzerStar

Bearbeiter: Sebastian Enger / V.1.4.1 / 07.10.2013

Definition:

- Content Server: enthält videos, ist über schnittstelle [https://contentserver\\_ip/search.php](https://contentserver_ip/search.php) ansprechbar
- Proxy: Zwischen server vermittelt zwischen Handy und Content Server
- Handy: Frontend des Smartphones

Frontend/App:

- **Xml Configuration:** Parameter des Handys(zb. Startscreen Wartezeit) soll in eine XML Datei geschrieben werden, mit den man das Handyverhalten etwas beeinflussen kann
  - **Priorität:**
    1. Proxy Conf
    2. Handy Conf
  - => **Achtung:** Eventuell nur Proxy Conf statt Handy Conf
- XML Protokoll (Handy to proxy, proxy to handy, proxy to content server, content server to proxy)
- Gzip Kompression
- AES (256) Unterstützung (Handy to Proxy, Proxy to handy)
  - Verbindung: Proxy<->Handy: AES256,Gzip
  - Verbindung:Proxy<->Content Server: https (Proxy wartet Zertifikat SSL von Content Server nicht aus)
  - Verbindung:
    1. Handy<->Content Server – HTTPS – wenn Android > 3.X (Hinweis: nur Android 3.X unterstützt https streaming): Download und Streaming
    2. Handy<->Content Server – HTTP – wenn Android < 3.X: nur Download über HTTPS , Streaming über HTTP
- Funktionsumfang des Stream Screens soll dem MX Player entsprechen - <https://play.google.com/store/apps/details?id=com.mxtech.videoplayer.ad&hl=de>
- Ffmpeg support (Achtung: es müssen mehrere binaries von ffmpeg vorgehalten werden, da es verschiedene mobile cpu architekturen gibt)
  - Ffmpeg muss als binary in der app mitgeliefert werden, eventuell intern nochmal komprimiert
- Zielplattform: ab Android 2.3 bis zur Android 4.3 Version hoch
- Startbootscreen -> X Sekunden warten(siehe Xml Configuration): URL und Logo einblenden
- Menüführung:
  - Suchscreen
  - Downloadscreen
  - News Screen (zeigt neue Filme an)-> Klick => Download oder Stream startet
  - Stream Screen
  - History Screen ( zeigt stream und download verlauf an)
  - Inform your friends screen
- Funktionalität Download Datei oder Streaming Datei

- Download/Stream Resume Funktion auch nach beenden
- Multi Source Download (http GET RANGE) von verschiedenen Video Servern
- SSL Support (Verbindung zum Download Server komplett via SSL-256Bit)
  - Android < 3,X: unterstützt kein SSL – fallback auf HTTP
  - Android > 3,X: unterstützt SSL -> Immer HTTPS nutzen
- Verbindung (AES,GZIP) zum JProxy
- Show more results – lade weitere Ergebnisse vom JProxy dynamisch nach (XML Result Protokoll anpassen)
- Multi Language Support via XML Sprachfiles (DE,EN: ~~später: FR,ES,RU,IT,JP~~)
- Möglichst keine Hardcodierten Sprachanweisungen in der App -> Auslagerung in Sprach XML File
- Die Videos von [https://contentserver\\_ip/video.mp4](https://contentserver_ip/video.mp4) sind zusätzlich mit einem AUTH\_BASIC Passwort geschützt (siehe Proxy Code > configuration.conf)
- BuzzerStar Logo beim Start und beim Beenden incl Verweis auf URL [www.buzzerstar.com](http://www.buzzerstar.com)
- Download History ( 23 letzten gedownloadeten dateien – keine download quellen speichern) (siehe Xml Configuration)
- Streaming History ( letzten 23 gestreamten dateien – keine stream quellen speichern) (siehe Xml Configuration)
- ~~Suchcache auf dem Handy ( die letzten 10 Suchergebnisse incl. Quellen auf dem Handy verschlüsselt (AES 256) speichern. CryptKey:~~
  - ~~IMEI+ANDROID-VERSION+“BuzzerSuperStar2013“~~
- Offline Modus: wenn keine Verbindung zum JProxy ~~dann nimm die letzten 5 Suchergebnisse Cache, entschlüssel sie und~~ sag (XML SPRACHFILE): BuzzerStar benötigt Internet auf deinem Mobiltelefon, um dir Zugang zu neuen Inhalten zu bieten. Wir zeigen dir jetzt gespeicherte Ergebnisse an. Fehlercode XYZ
- Arbeiten mit Fehlercodes (siehe Xml Configuration)
  - Kein Internet: Fehlercode NOINET
  - Kein Speicherplatz: Fehlercode: NOMORESPACE
  - Usw.
- Update Check auf [www.buzzerstar.com/update.php](http://www.buzzerstar.com/update.php) (siehe Xml Configuration)
- Download und Installation neuer APK wenn update check positiv: [www.buzzerstar.com/Buzzer.apk](http://www.buzzerstar.com/Buzzer.apk) aus App heraus (siehe Xml Configuration)
- Inform your friends – via sms vom handy aus (InformText: siehe Xml Configuration)
- ~~Bluetooth/NFC Sharing support~~
- News Screen -> Filme nach ~~Jahr/Genre~~ anzeigen und klick drauf macht es suchbar
  - News Screen XML in Proxy einbauen
- ~~Button „i want“ -> user kann schreiben, was er an filmen, serien etc haben will (und wir fügen das hinzu)~~

## JProxy Server

- Verbindungsschnittstelle zwischen abgeschotteten Content Server und Handy
- Aufgabe: Verbindungsanahme vom Handyclient
  - XML Protokoll:

- Find-> Ergebnisse auf dem Content Server suchen -> dazu dort /search.php via SSL-256 bit ansprechen
  - Result-> Ergebnisse ausliefern, Cache all der Ergebnisse vom Content Server vorhalten für „Show more Results“
  - News Results
  - XML More Search Results
- Pro Handy Request:
  - Suche am Content Server nach Inhalten, wenn kein Cache Hit
- *Es müssen n-konfigurierbare Content Server vom Proxy angesprochen werden können (Beispiel: <https://13.13.13.13/search.php>, <https://33.33.33.33.33/search.php>, <https://82.82.12.34/search.php> liefern zusammen ergebnisse zum handy suchbegriff und müssen parallel (in einem Thread im Proxy angefragt werden)*
- Caching der Ergebnisse einer Suchanfrage incl Results vom Content Server für XYZ min
- Email/SMS Benachrichtung, wenn Server Port des Jproxy down
- Konfigurationsdatei für den Proxy wird bei jedem Start neu eingelesen

#### Youtube Support:

- Altersbeschränkung von Youtube beachten und in Proxy Configuration aktivieren oder deaktivieren
- Suche erfolgt im Proxy (Youtube Suche über Kanäle, Videos und Wiedergabelisten)
  - Rückgabe:
    - Youtube Link
    - Youtube Wertung
    - Youtube Title
    - Youtube Vorschaubild
    - (modular erweiterbar)
- Separate youtube config für den proxy (anzahl ergebnisse, api keys, aktiv/inaktiv, etc)
- Es müssen mehrere Youtube Api Keys hinterlegt werden können:
  - Existieren mehrere: zufallsauswahl welcher genommen wird
  - Einer: nimm diesen
  - Melde, (proxy logfile) wenn ein Api Key fehlerhaft/deaktiviert wurde
  - Überspringe deaktivierte Api Keys
  - Speichere pro Api Key die bereits ausgelieferte Anzahl von Requests pro Tag
    - Wenn nur 1000 erlaubt werden, nimm ab 1001 einen neuen
- Youtube ergebnisse im handy mit Youtube Icon (unten) markieren
- Sortierung der Ergebnisse im Proxy, dann Darstellung auf der Handy App:
  - 1. Reguläre Kinoergebnisse
  - 2. Youtube Ergebnisse
  - 3. Zufällige Ergebnisse
- Handy App muss reguläre Youtube Video Links unterstützen ( Beispiel: <https://www.youtube.com/watch?v=5-405Vvn3OU>)
- Handy App muss im Stream Screen das Youtube Video abspielen

- Handy App muss im Download Screen das Youtube Video runterladen, nach dem Titel des Videos benennen und abspeichern
- Welche Qualität das Youtube video hat wird in der youtube proxy configuration festgelegt
- Komplette Logik (zb. Wie hole ich mit einer Regex nach einem Video, inkl. Regex Muster) auf den Proxy verlagern, sodass man schnell anpassungen vornehmen kann
  - ~~???? Proxy liefert eventuell nur noch direkte Youtube Links aus (ist dies technisch machbar, ohne pro suchanfrage tausende Youtube Requests zu senden?) ?????~~
  - Wenn das Handy die Videos extrahieren muss, sollen die dazugehörigen regex muster vom proxy kommen (zwecks einfache Anpassbarkeit)
- Soweit wie möglich aktuellste, offizielle YouTube Api zum suchen benutzen, Logik auf Proxy auslagern
- Wenn Youtube Api Key auf dem Handy gebraucht wird, muss es via Proxy XML Protokoll gesendet werden
- Werbung von Youtube muss ignoriert bzw automatisch übersprungen werden
- Allgemein Neu:
  - Proxy Conf steuert Handyverhalten (Screen deaktivieren/aktivieren) etc



