

1 Geschäftsanforderungen

1.1 Hintergrund

Die Fertigstellung der zweiten Release-Version von emo brachte drastische Verbesserungen des Protokolls mit sich, die es möglich machen, weitaus komplexere Clients auf allen Plattformen zu entwickeln.

Mit chibimo soll nun ein Client geschaffen werden, der sich diese Funktionen zunutze macht, um das Konzept von emo auch auf portablen Geräten (in diesem Fall Android) bereitzustellen.

1.2 Geschäftsmöglichkeit

Das gesamte emo-System ist Freie Software unter der GNU Affero GPL 3.0. Es dient nicht dem Zweck, monetären Gewinn zu erbringen, stattdessen strebt es an, eine Alternative zu proprietären, kostenpflichtigen & oft mit Anti-Features ausgestatteten existenten Musikbibliotheks-Managern zu sein. chibimo als Client ist eine logische Erweiterung dieses Ziel.

1.3 Geschäftsziele & Erfolgskriterien

Das Ziel ist, das emo-System gemäß der Philosophie Freier Software zu erweitern. Desweiteren muss chibimo über sämtliche Grundfunktionen typischer emo-Clients verfügen, sowie einen Offline-Mode zur besseren mobilen Nutzung. Für weitere Details ist das [Exposé](#) einzusehen.

1.4 Erfordernisse von Kunde oder Markt

chibimo benötigt als emo-Client per Definition eine Verbindung zu letzterem. Demzufolge müssen Kunden über eine eigene Installation von emo verfügen oder einen Provider für diesen Service benutzen.

1.5 Geschäftsrisiken

Als Freies Projekt unterliegt das emo-System keinerlei Risiken. Sollten die ursprünglichen Autoren das Projekt nicht mehr fortsetzen können, ist dank der Lizenz eine Fortführung durch die Community möglich.

2 Vision der Lösung

2.1 Vision

chibimo soll ein vollständiger emo-Client für Android werden. Gemäß der Natur von emo richtet er sich an Nutzer mit technischem Grundwissen, soll allerdings einfacher zu benutzen sein als der alternative plattformunabhängige Client, welcher in Bash geschrieben ist und von mehreren externen Tools abhängt. Im Kontrast dazu soll chibimo nach minimaler (in-App) Konfiguration kontinuierlich funktionieren.

2.2 Wichtigste Features

- Simple hierarchische Übersicht über die Musikbibliothek
- Wiedergabe und Steuerung dieser im Hintergrund
- Offline-Modus, in dem trotzdem alle Funktionen von emo zur Verfügung stehen

2.3 Annahmen und Abhängigkeiten

Die Bedienung von chibimo ist trivial, allerdings ist die allgemeine Einrichtung eines emo-Systems mit ein paar manuellen Schritten verbunden, die auf einem Linux-Server ausgeführt werden müssen. emo ist primär als ein Do-it-yourself-Projekt gedacht, bei welchem der Nutzer zumindest über einen eigenen Server verfügt. Allerdings steht nichts der zukünftigen Anbietung von emo als remote-Service im Weg.

3 Fokus und Grenzen

3.1 Umfang des ersten Release

Das erste Release von emo selbst verfügte über ein sehr simples globales Kommunikationsprotokoll, unterstützte jedoch schon alle Funktionen, die auch in späteren Releases enthalten sind. Lediglich das Protokoll und die Implementierungssprache wurden verändert.

Im ersten Release von chibimo soll die komplette Interaktion mit emo funktionstüchtig sein. Eine stabile, Crash-freie Benutzung muss selbstverständlich gewährleistet sein.

3.2 Umfang der folgenden Releases

Für emo ist primär eine Funktion zur Auswahl des Listener-Sockettyps geplant. Hierbei soll zwischen einem TCP-Port und einem lokalen UNIX-Socket umgeschaltet werden können.

Aktuell ist nur ein weiteres Release für chibimo geplant, in welchem der komplette Offline-Mode eingefügt wird. Abgesehen von Kundenwünschen ist chibimo dann feature-complete.

3.3 Begrenzungen und Ausschlüsse

Das emo-System wird nicht über Account-Verwaltung verfügen. Solche Funktionen werden bereits durch das Zielbetriebssystem des emo-Servers (Linux) angeboten, mit wesentlich höherer Sicherheit, als ein Musikplayer-Tool jemals implementieren könnte.

emo-Provider können zum Multiplexen mehrere emo-Server-Instanzen das Tool [kaiso](#) benutzen, sobald emo UNIX-Socket-Funktionalität besitzt.

Weiterhin wird emo keine Kommunikation über TLS implementieren. Ist dies gewünscht, können Tools wie z.B. [stunnel](#) benutzt werden.

4 Geschäftskontext

4.1 Stakeholder

Das emo-System ist ein Freies Projekt und wird nicht im Auftrag von Stakeholdern entwickelt. Der Philosophie Freier Software folgend ist es als allgemeine Kontribution an die Gemeinschaft aller Nutzer gedacht.

4.2 Projektprioritäten

Feste Prioritäten:

- Zweites Release von chibimo bis 2022-06-20

Ohne Deadline:

- Drittes Release von emo

4.3 Technische Anwendungsumgebung

emo läuft auf Linux-Servern. Es ist in Nim geschrieben, benötigt also für den Buildprozess einen ausreichenden modernen Nim-Compiler ($\geq 1.6.4$).

chibimo benutzt als Android-API-Mindestlevel 28. Da es keine nativen Komponenten benutzt, ist für den Buildprozess nur eine Standardinstallation der Android-SDK samt Tools erforderlich.

Sowohl emo und chibimo führen simple, kaum ressourcenintensive Aufgaben durch. Die Hauptanforderung an den Nutzer ist der Speicherbedarf für die Musikbibliothek. Diese kann sich einen Server mit emo teilen (default-Konfiguration des allgemeinen Clients) auf dem mobilen Endgerät präsent sein (für chibimo erforderlich).

Durch die Netzwerkstruktur aller Teile des emo-Systems ist eine fast unbegrenzte Trennung des Nutzers vom emo-Server möglich.