**Systemkomponenten**:

Spotify-API <–> App <–> Server (mit Python) <–> mySQL

**Datenbankmodell**:

Session-Table:

SessionID, sessionTimestamp, HostDeviceID, Mode, Cooldwon-Timer, *ARTISTS*, *GENRES*, *PLAYLIST*, *STATS*

Participant-Relation:

ParticipantDeviceID, SessionID

Tracks-Table:

SpotifyID, Name, Image, ARTISTS, GENRES,

Session-Track-Relation:

SpotifyID, SessionID, Upvotes, Cooldown, isLocked

**App**:

Konzept:

Der Server hat alle Daten zur Vorschlagsliste gespeichert. Beim Session-Beitritt erhält der Participant alle für ihn relevanten Daten. Danach erhält jeder User noch Updates mit den Infos, die er noch nicht besitzt. Mit diesen Updates aktualisiert der Participant seine „Kopie“ der Daten. Löst der User Veränderungen aus, teilt er diese dem Server mit und der Server informiert die restlichen User in der Session. Die Queue wird nur lokal in der App beim Host gespeichert, nicht auf dem Server.

Die Session-Infos werden beim Erstellen oder Beitreten geholt und dann lokal abgespeichert.

Die Statistiken werden jedes Mal beim Öffnen vom Server angefragt.

Die Tracksuche geschieht ohne Involvierung des Servers. Nur ausgelöste Vorschlagslisten-Veränderungen werden dem Server mitgeteilt.

**Schnittstelle App-Server**:

**Server**:

* getUserSessionState(deviceID) -> Enum: HOST/PARTICIPANT/NONE
* createNewSession(hostDeviceID, mode, cooldown, artists, genres, playlistTracks)

-> sessionID, sessionTimestamp

* joinSession(deviceID, sessionID) -> sessionTimestamp
* openUpdateSocket(deviceID) -> Vorschlagslistendaten
* updateUpvoteCounter(deviceID, trackID, newUpvoteCounter)
* addNewTrack(deviceID, trackID, name, artist, image)
* startCooldownOnTrack(hostDeviceID, trackID)
* setLockTrack(hostDeviceID, trackID, isLocked)
* deleteSession(hostDeviceID)
* exitSession(participantDeviceID)
* isSessionOpen(sessionID, sessionTimestamp)

**App**:

* updateUpvoteCounter(trackID, newUpvoteCounter) (Löschen durch Setzen auf 0 außer im Playlist Mode)
* addNewTrack(trackID, name, artist, image, upvoteCounter)
* setLockTrack(trackID, isLocked)
* deleteSession()

**Fragen**:

Wie Script-Kiddy-proof soll die Kommunikation zwischen App und Server sein?

Muss der Server objektorientiert entworfen werden?

Wie bildet man riesige Diagramme im Entwurfsdokument ab?