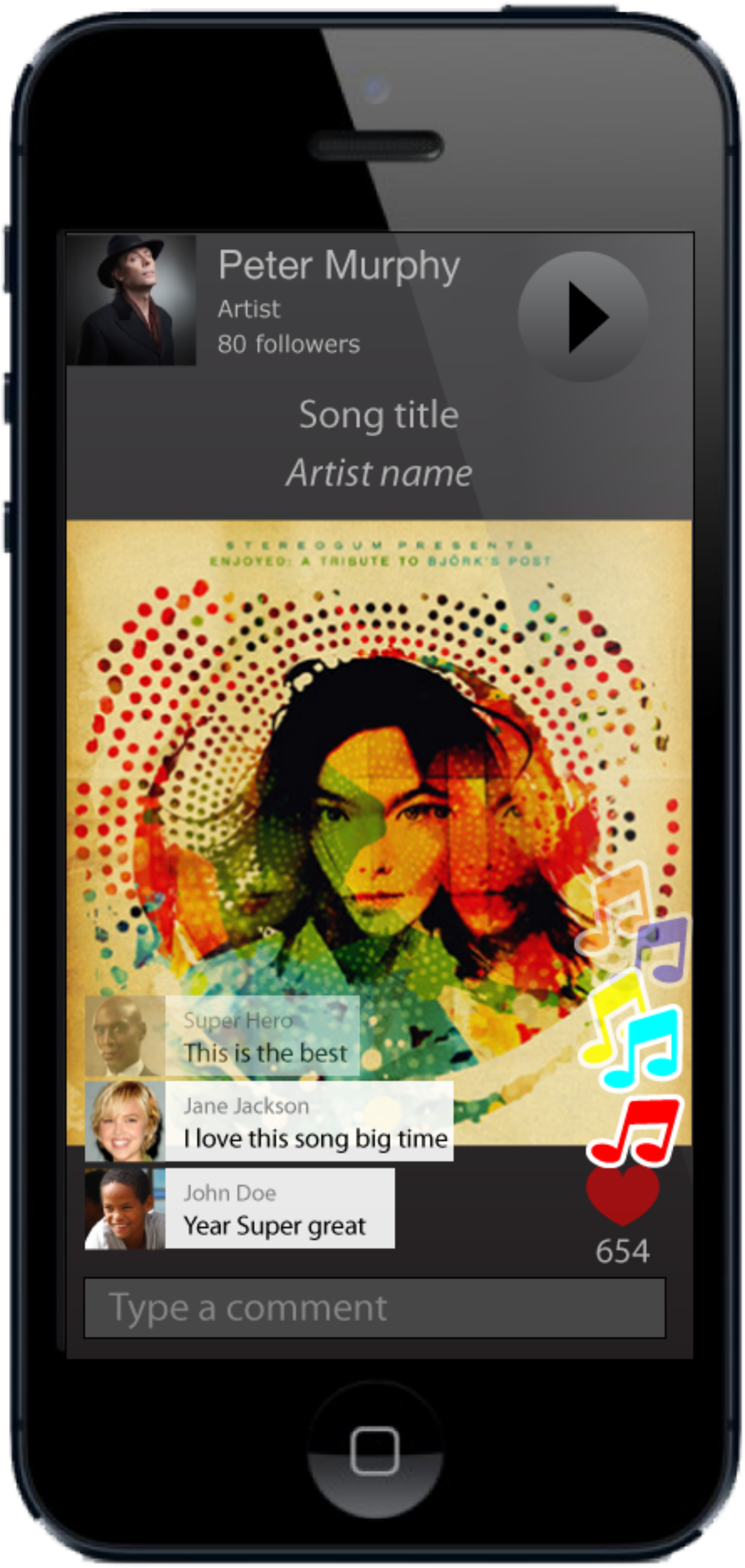
DigiRAMP Mobile Development Roadmap

Enjoy Music with your friend



Peter Murphy

Artist

80 followers

Song title

Artist name

Super Hero
This is the best

Jane Jackson
I love this song big time

John Doe
Year Super great

Table of content

About this document

A roadmap for development:
Here I will try to draft what research is required to make good decisions on technical solutions, what implementations is needed and how to prove principles. What features are "need to have" and what are "nice to have". How long it will take to reach milestones. Who should implement them?

The proposed app provides some core features.

- Basic media player features
- Basic social features
- Real-time comments and applauses
- Streaming of stored media
- Real-time Streaming
- Storage and replay of comments and applauses.
- User accounts

The detailed planning ends at a version 1.0.0, more rough future features are then outlined at the end of the document

Business-to-Business deals about content in quantity are not included in this document although injection of big data has to be implemented when protocols and technology are known.

One temporarily solution could be to become a retailer for theorchard
http://www.theorchard.com/contact-us/

The stack

Here I'm looking at the final environment for the backend, the overall goal is to decide implement a platform foundation that:

- Scale to hundred of millions of users.
- Has a low latency.
- Is easy to maintain.
- Is fault-tolerant.
- High performance
- Is flexible.
- Is componentized
- Has a high value on exit
- Is independent from providers

After the platform is implemented more rapid development and release, cycles can start.

Backend servers overview

Authorization, Streaming and real-time features has different needs and should be implemented using different SW / HW solutions

- Media storage: Amazon S3 is a common used standard used by e.g. Amazon, Soundcloud. Flikr
- Authorization: Ruby on Rails. Has proven security, is easy to implement, is well 5supported and does not require a lot of bandwidth.
- Real-time push notifications for user interaction: Erlang meeds the needs for real-time features, there exists two open source framework that should make implementation 'easy' and 'fast' http://www.phoenixframework.org/ and http://elixir-lang.org/
- Server hosting, Hetzner, Amazon, something else?.Price vs Maintance. Chef scripts keeps DigiRAMP independent https://www.chef.io/

Versions

Release plan with milestones. The goal is to get to beta 0.9.0 as fast as possible and from there heading towards release 1.0.0

Alpha versions

Alpha versions are developed in incremental steps with milestones between

Alpha V. 0.1.0

Start week 0. Duration 6 weeks
This first release cycle confirm the inter application communication between the different components because this is an absolute must and this is where the highest risk is.
What will not be implemented is

- The right stack instead Heroku will handle backend configuration this will be at a cost on performance
- Staging servers
- Core functionality
- Backup
- Automated tests
- Media encoding

The challenge is to get domain expert found and a well-organized workflow.
The 0.0.X version numbers reflect the order of implementation
I'm added a letter to the end letters for components that does directly rely on each other and can be implemented and tested independent and in parallel

- 0.0.1a: Select and install Sharable Databases Dynamo DB, Big Tabel, Postgresql, Redis
- 0.0.1b: Rudimentary SDK and UX design for Mobile app alpha version
- 0.0.2a: Configuration of authorization server with user email authorization from web interface and API
- 0.0.2c: Deployment of Erlang server with WebSocket chat
- 0.0.3a: IAC between Authorization server and Erlang servers for sessions and cookies.
- 0.0.3b: Rudimentary API for APP and Erlang server
- 0.0.4 Initial app with user Login and Signup with email / password
- 0.0.5: WebSocket chat from APP.
- 0.0.6: CRUD chat in DB from app
- 0.0.7: Benchmark APP <-> Erlang server latency on localhost
- 0.0.8: Stress test and bottlenecks
- V- 0.1.0 Release Milestone and Evaluation

Personas

- Database Expert to select environment and principles 1-2 weeks
- UX Designer 1-2 weeks
- Data Scientist. For Erlang coding 4 weeks
- Ruby Developer for API, Authorization server 1 week

- iOS Developer 1 week

Time resources

- Find personas 1-2 weeks
- 0.0.1 and 0.0.2 2: one week
- 0.0.3 and 0.0.4: two weeks
- 0.0.4 to 0.0.8: two weeks

Alpha V. 0.2.0

Start week 6. Duration 3 weeks
Basic integration of the DigiRAMP delivery engine for audio and show real time features.

- 0.1.0: Interaction design, wireframes, navigation
- 0.1.1: UX mockup theme. Fonts, Colors. Icons
- 0.1.2: IAC between authorization server and DigiRAMP server
- 0.1.3: API V. 0.2.0 for DigiRAMP delivery with online documentation
- 0.1.4a: Selection of streaming audio format
- 0.1.4b: App access audio file and waveform image file from backend
- 0.1.5: Playback of audio file in app
- 0.1.6: Remote control of playback from backend
- 0.1.7: Read Write of chat during playback
- 0.1.8: Multy language support
- 0.2.0 Release Milestone and evaluation

Resources and personas

- UX Designer 2 weeks
- I OS Developer 2 weeks
- Data Scientist. For Erlang coding 2 weeks
- Ruby Developer 2 weeks

Alpha V. 0.3.0

Start week 9. Duration 3 weeks
Building a Media Player GUI and beef up the backend, there will be little interaction between app and backend in this cycle

- 0.2.0a: App: implement screens and navigation (no functionality)
- 0.2.0b: DB Architecture for recording of notifications and applauses
- 0.2.1a: App: Apply stunning GUI based on UX mockup
- 0.2.1b: Decide on protocol for app <-> backend json, message pack, XML
- 0.2.1c: Deployment architecture for production releases
- 0.2.2: API requirement for commerce engine
- 0.2.3: API for social features
- 0.2.4: Security concerns
- V- 0.3.0 Release Milestone and Evaluation

Resources and personas

- UX Designer 1 week
- iOS Developer 2 weeks
- Data Scientist. For Erlang coding 2 weeks
- Ruby Developer 2 weeks
- Deployment (sysops) consultant 2 weeks
- Security consultant 2 weeks
- Database Expert 1 week

Alpha V. 0.4.0

Start week 12. Duration 2 weeks
Content and organization, find music based on different parameters, genre, artists, playlists, search, connect app with user account

- 0.3.0a: Erlang Solr integration for music search
- 0.3.0a: Erlang <-> DigRAMP search integration
- 0.3.0b: Access social features in backend from app through API

Resources and personas

- iOS Developer 2 weeks
- Data Scientist. For Erlang coding 2 weeks
- Ruby Developer 2 weeks
- Deployment (sysops) consultant 1 week
- Database Expert 2 week

Hire People

Not all tasks requires full time employment nor is it easy to find top skilled motivated employees,

Server provisioning

Instead of hiring a expert to maintain servers DigiRAMP relies on deployment scripts from chef. This is a industry standard and backed by a open source society and there is big companies offering consultant services https://www.chef.io/
And it's possible to find freelancers https://www.upwork.com/o/profiles/browse/?q=Chef

Erlang backend

Full time position a on location position down the road
https://www.upwork.com/o/profiles/browse/?q=erlang