DigiRAMP Mobile Development Roadmap

Enjoy Music with your friend



### 

Table of content

About this document 4

Versions 5

Alpha versions 5

Alpha V. 0.1.0 5

Alpha V. 0.2.0 6

Alpha V. 0.3.0 6

Alpha V 0.4.0 7

The stack 8

Backend servers overview 8

Authorization 8

# 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 service providers (amazon excluded )

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 supported 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/>

# DigiRAMP Community integration

The DigiRAMP API provides lot of the features required to make the mobile application work.

Although the business logic is in place for the backend the API has to be written. It’s also foreseeable the exact set of requirements and features for the backend will change.

## Migrating current server

The current version of the DigiRAMP is running on one dedicated server. With more than 100.000 active users a need for server clusters will raise.

## Additional support features

Subscription to streaming services and bulk import and access to vast catalogs, legal agreements is left out of this document although it requires time.

# 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: Automated Benchmark APP <-> Erlang server latency on localhost
* 0.0.8: Automated Stress test and bottlenecks
* V- 0.1.0 Release Milestone and Evaluation

Personas

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

Time resources

* Recruitment 1-4 week
* 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 prof of the 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: Multi 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 Mockup and beef up the backend, receive real time data and requests from the app

* 0.2.0a: App: implement screens and navigation (no functionality)
* 0.2.0b: DB Architecture for recording of notifications and applauses
* 0.2.1a: Decide on protocol for app <-> backend json, message pack, XML
* 0.2.1b: Deployment architecture for production releases
* 0.2.2a: Push test data from app to backend
* 0.2.2b: 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

* Recruitment 2-4 week
* 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

### Prove of Principle and Evaluation

Start week 12. Duration 1 week.

At this point all technological risks and challenges has been touched and solved.

The app is not ready for external evaluation. From here all the fun stuff starts and major components is talking to each other. It’s also a good point to evaluate the Company climate, The Team, The product, and the roadmap. After this point the implementation of the GUI will be ramped up and an additional developers is brought on board

### Alpha V. 0.4.0

Start week 13. 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
* 0.3.0c UX Spec evaluation
* 0.3.1 API for curated content
* 0.3.2 API for sponsored content

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

### Alpha V. 0.5.0

Start week 15. Duration 2 weeks

GUI for signing up and user profile and basic playback

* 0.5.0a: GUI for Front page
* 0.5.1a: GUI for signup and authorize in app
* 0.5.2a: GUI for user in app
* 0.5.0b: GUI for music front page
* 0.5.1b: GUI for genre page
* 0.5.2b: GUI for search music page
* 0.5.0c: Automated tests in back end
* 0.5.1c: DB Stress test
* 0.5.2c: Payment / permission for music usage API

Resources and personas

* 2 x iOS Developer 2 weeks
* Data Scientist. For Erlang coding 2 weeks
* Ruby Developer 2 weeks
* Database Expert 1 week

### Alpha V. 0.6.0

Start week 17. Duration 2 weeks

Implementing payment , permissions and signup from App,

* 0.6.0a: Payment from app
* 0.6.0b: Signup with oAuth from app (Facebook, Twitter, Google+, Instagram )
* 0.6.1a: Signup with cell nr

Resources and personas

* 2 x iOS Developer 2 weeks
* Ruby Developer 2 weeks

### Alpha V 0.6.0 to Beta 0.1.0

Start week 19. Duration 6 weeks

In this releases the application will be

* Refinement based on user feed-back,
* Prepared for delivery to the app store.
* Translated to 2 additional languages.

# Hire People

Not all tasks requires full time employment nor is it easy to find top skilled motivated employees, There is third party companies that can do parts of the development on a short notice.

## Server provisioning

Instead of hiring an 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 developer

A Full time position on location

https://www.upwork.com/o/profiles/browse/?q=erlang