GRAVITY GROOVE

Technical Design Document

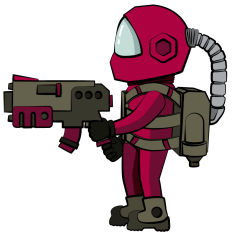
developed by

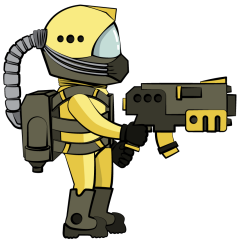
SPACE MUFFIN

Index

Space Muffin Team

Giulia Barnaba
Game Designer
giulia.barnaba92@gmail.com



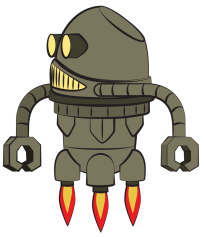


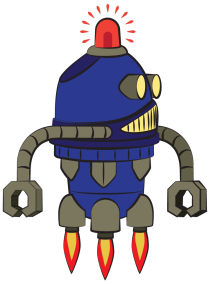
Lorenzo De Simone
Game Designer
Game Programmer
lorenz.desimone@gmail.com

Andrea Jegher
Game Programmer
andrea.jegher@mail.polimi.it

Emanuele Lattanzio
Game Designer
Game Programmer
emanuele.lattanzio@studenti.unimi.it

Marta Barducchi
Visual Artist
marta.barducchi@gmail.com





Valerio De Vittorio
Sound Designer
Music Composer
valky3it@gmail.com

1 Project Goal

Our goal is to produce an online real time mobile multiplayer game, which can be played worldwide 24/7 from iOS on Android devices (see chapter 3 Client side for details about versions and device ranges).
The game will offer the possibility to join or create a match with other people online, see a global leaderboard and share contents on the social media while using our application.

2 Provided Services (Beside the Game)

Beside the game itself, we provide the following additional services:

- An official website where it will be possible to provide feedback and opinions on the game and there will be a forum section for bugs and problems.
- Integration with Google Play and Apple Store achievement and leaderboard.
- Integration with social networks. At the end of a game or right after unlocking an achievement it will be possible to publish the results with a screen of the leaderboard or the replay of the game on the most important socials as Twitter, Instagram, Facebook, Youtube.

3 Client side

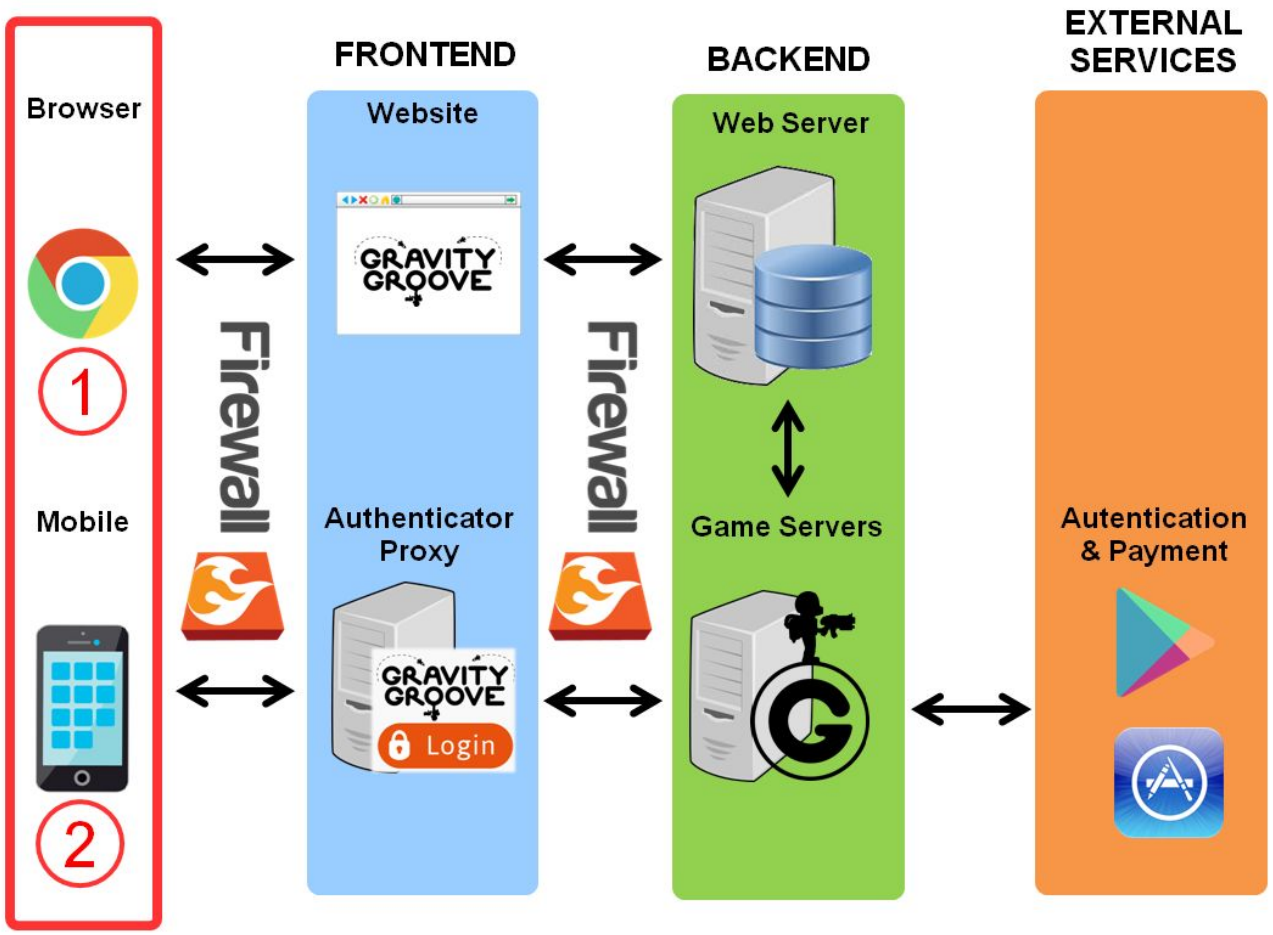


Fig.1 - How a client connects to our architecture

Player can access our services in two ways:

1. Smartphone to play the game. Players will use their smartphone as device and no additional hardware is required.
2. Web browser to log into our official website. In order to access our website any modern browser should be supported.

3.1 Hardware Requirements

The minimum hardware requirements are:

- Mid-range smartphone (200 euros price range)
- Network technology: compatible with at least 3G
- Internal Storage: 50 MB free
- Ram: 1 GB or greater

The suggested hardware requirements are:

- High-end smartphone (200 euros or more)
- Network technology: compatible with 4G
- Internal Storage: 50 MB free
- Ram: 3 GB or greater

- Display size: 5.0 inches or larger

3.2 Software Requirements

The minimum SW requirements are;

- Android version: 4.1 Jelly Bean (API level 16) or greater.
- iOS version: 7.x or greater.

Further details on the reasons of this choice can be found in the Game Design Document, chapters 3.2 Platform and 3.n3 System requirements.

4 Workload Estimation

For the closed beta of the project we opted for a p2p system, where players will host matches.
In a future extension we are planning to use a client-server system where the match will be hosted on the cloud. The webserver will be hosted on a couple of machines in different geographical regions.
Our cost per user will be analyzed in detail in the chapter 13.2 Revenue Estimation.

- Closed beta:
  We are sending game download codes in a strictly controlled environment to run our first tests.
  - Estimated daily affluency: 640 players
  - Estimated daily maximum users simultaneously connected: 256 players
  - Total subscribed players: 1,920
  - Game servers: p2p
- Open beta:
  Our website will be active and people must show interest in our game. If someone is interested can obtain a game download code by registering to our website.
  - Estimated daily affluency: 1,300 players
  - Estimated daily maximum users simultaneously connected: 520 players
  - Total subscribed players: 3,900
  - Game servers: Amazon Web Services
- On release:
  At this point, our estimate for release affluency takes account of a series of factors: some of the players of the open beta might still be interested in our game and decide to download the official game from the store, others might not be interested after trying and a small number can be composed of entirely new players.
  - Estimated daily affluency: 3,500 players
  - Estimated daily maximum users simultaneously connected: 1,100 players
  - Total subscribed players: 10,500
  - Game servers: Amazon Web Services

- 4 months after release:
  The game is spreading through the community thanks to advertising and social media sharing. In this period we see our biggest growth spike in terms of players' affluency.
  - Estimated daily affluency: 9,000 players
  - Estimated daily maximum users simultaneously connected: 3,500 players
  - Total subscribed players: 29,000
  - Game servers: Amazon Web Services

- 8 months after release:
  This is the maximum reasonable amount of players we do believe we can attract given our target audience and offered services within the first year. Should we need more in the future, our cloud infrastructure architectural choice will provide a quick way to scale up or down our capacity to fit our needs.
  - Estimated daily affluency: 12,000 players
  - Estimated daily maximum users simultaneously connected: 6,000 players
  - Total subscribed players: 40,000
  - Game servers: Amazon Web Services

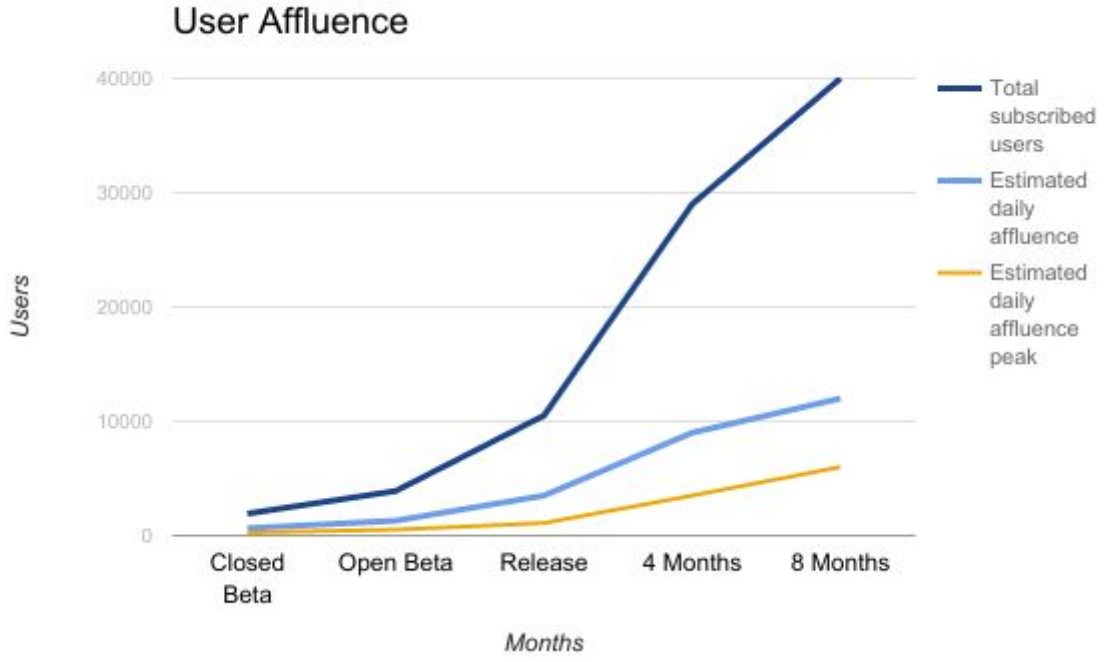


Fig.2 - User affluence

4.1 Player Costs

The following estimation of player costs is over on release period, as it is the starting point of the worldwide and public distribution. Given the estimated daily affluence peak we estimated in the previous chapter, we can assume that the architecture designed in this document will provide a stable service for those numbers. In case of unexpected results, we can modify our cloud contract with Amazon accordingly.

Category	Usage per User	Number of Users	Total Resource Usage	Total Cost
Disk Storage	10 Kbyte	10,500	105 Mbyte	€ 0 As of the current solution until 8GB of space usage no additional costs are charged. This threshold should virtually never be surpassed given the estimated amount of space per user. Nevertheless we specify that after this threshold there is a cost of € 0.020 per GB each Month.
Network Band	20 Kbit/s - 30 Kbit/s	1,100	2.5 Mbit/s - 3.5 Mbit/s	€ 0 As of the current solution until 800 Mbit/s of network usage. No additional costs are charged.

In order to evaluate our cost per user we created our own estimator that is based both on our infrastructure and our help desk services:

$$cost\ per\ user = (Infrastructure\ cost + help\ desk\ call\ cost * 2\%\ total\ users)/total\ users$$

This formula is valid until our expected workload doesn't exceed the number of players we have estimated. For predictions beyond the first year, we would need to take account of how many servers we should buy given our game growth rate.
On the first year we estimated costs of € 8,819 for our infrastructure (Backend, Frontend & Help Desk) which is divided on each player:

On release:	10,500 players	€ 0.84 per user
4 months after release:	29,000 players	€ 0.34 per user
8 months after release:	40,000 players	€ 0.26 per user

5 Frontend

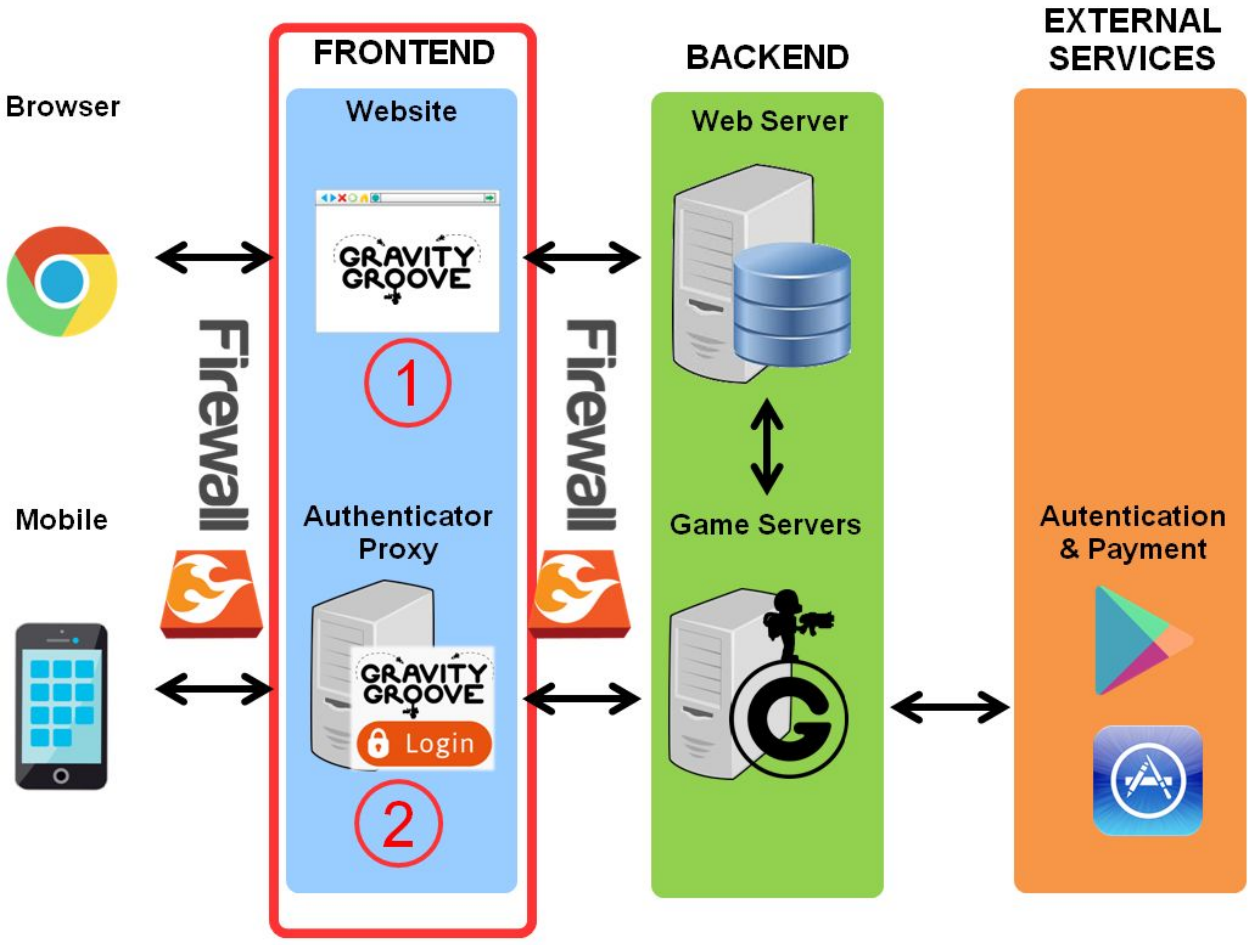


Fig.3 - Frontend of our architecture

The player can access frontend services using two kinds of access points:

1. Official community access from browser:
   Beside the direct access with mobile device to the game, the player will be able to access the website with the same account where he can post on forums and handle his connected devices, in order to provide a good environment for communities to grow over time. While the web server is shown as backend, it is possible to access the frontend through the browser; as a matter of fact our web front end is a software layer on the web server that is described in the backend chapter.
2. Game access from mobile device:
   The player needs to authenticate with either the Google Play account or the Apple Store one depending on the device os using our frontend layer in order to interact with the backend and actually play the game. The frontend layer is composed of two proxies whose task is to balance workload between servers and connect the player with our backend (further details are shown in chapter 9.1 Global Infrastructure Outline).

5.1 Hardware & Software

Frontend component	Chosen Solution	Hardware (if any)	Reason of our choice
Database Server Load Balancer	Amazon Web Services Elastic Load Balancer (Classic) (i3.xlarge)	N/A	*"Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications."* Amazon offers two kind of load balancers: the classic and the application one. We decided to use the former for the database load balanced since it doesn't need particular routing techniques or timing-strict performances.
Matchmaking Load Balancer	Amazon Web Services Elastic Load Balancer (Application) (i3.xlarge)	N/A	*See the upper row table for the general Amazon AWS Load Balancer.* We decided to use the application load balancer offered by Amazon since the match load balancer must offer a fast matchmaking procedure. Moreover, it offers the ability to route traffic to multiple services or load balance across multiple ports on the same server instance, which is really convenient for a matchmaking load balancer.

5.2 Scalability and Extensibility

In our network architecture, frontend servers' main tasks are the following:

- Provide a fast access to the community website through all modern browsers.
- Offer in-app game purchases through payment with APIs to Google Play and Apple Store.

Compared to backend services, frontend have lower expected workload; nevertheless, we do require a distributed solution in order to provide a stable service during system failures and scale dynamically in case of a sudden increase of subscriptions and this is the reason why we have chosen a cloud approach for our architecture.

5.3 Cost Estimation

Product	Advance Payment	Monthly Cost	Total (12 months)
Amazon Web Services Elastic Load Balancer (Classic)	€ 0	€ 16	€ 192
Amazon Web Services Elastic Load Balancer (Application)	€ 0	€ 16	€ 192
Column Total	€ 0	€ 32	€ 384
Total cost for final delivery			€ 384

6 Backend

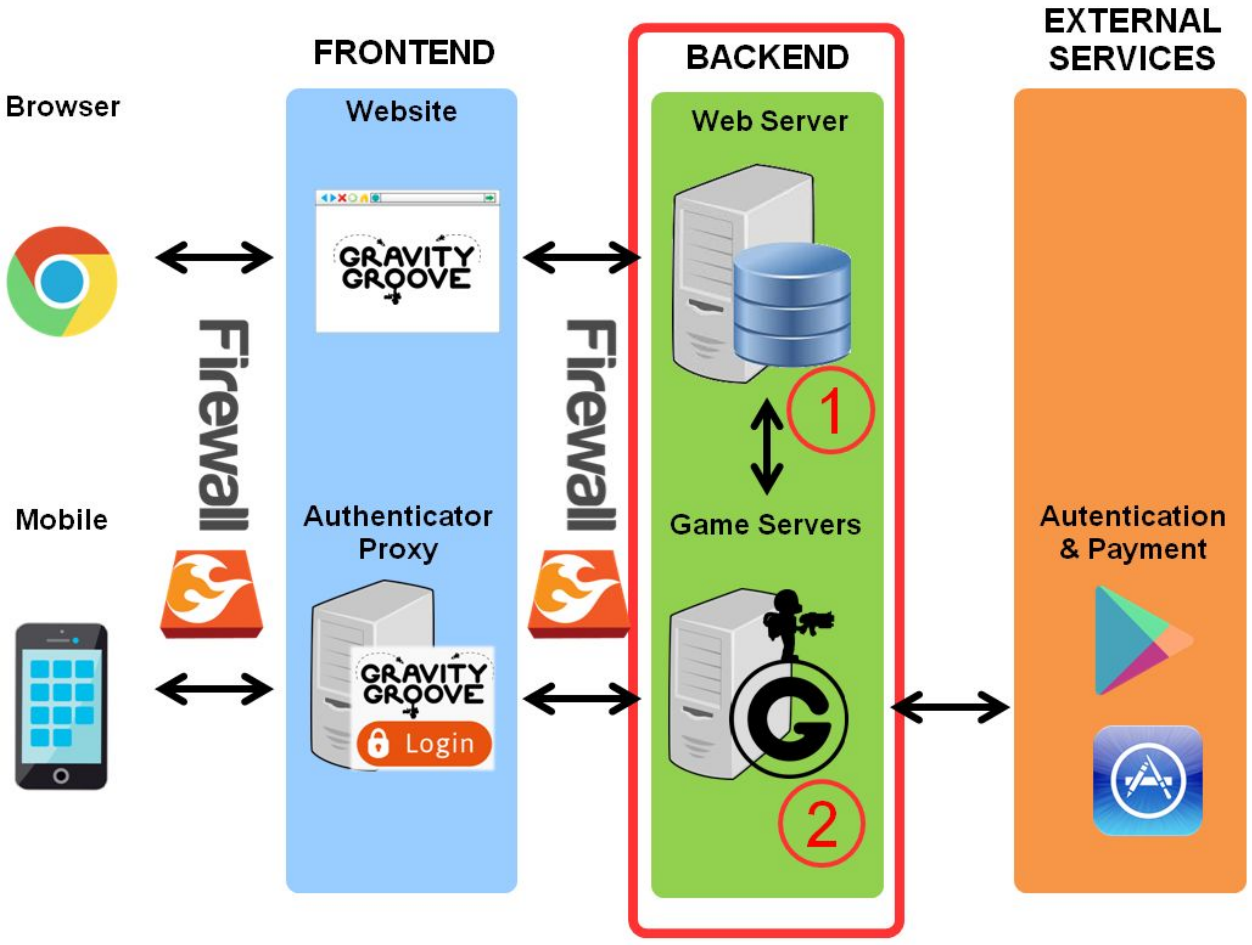


Fig.4 - Backend of our architecture

The backend is composed of three internal components controlled by us and an external one. The external part is the user account and payment which are delegated to Google and Apple services; we ask them to validate the players against their database and to process payments using their API.
The client uses a generated token to connect to our data servers and retrieve player's data (such as coins, score etc.).
Payments are handled outside of our app using the Apple store or the Google Play service.

Our backend consists of:

1. Web Server: it stores our community website and offers access to the forums using credential used for the game registration.
2. Game Servers (more details in chapter 9.1 Global Infrastructure Outline):
    - User data server: it stores, backups and receives queries with the user data. This data server holds players game data; it's platform independent and is available to players on any platform they use.
    - Matchmaking node: we host a matchmaker server in different world locations, that will advertise and create matches. Once enough players are found it will instantiate a match-host sending its information to the clients.

- Match-host node: this component is the server authority for a single match and the players will connect to it from their clients. Each match-host is a node in the cloud and can be generated dynamically when requested.

6.1 Hardware & Software

Cloud architecture serves best for our needs, both in terms of initial environment and future elasticity and scalability. After examining market offers, we decided to rely on Amazon services for a great number of reasons; first of all, it offers network infrastructures tailored for session-based games and therefore perfect for our needs (see the table below for more details). During our research on which company might suit us best, we found many other companies that offered quality services, but none of them were clear about costs and specific for our needs as Amazon Web Services. Moreover a lot of top performers in the mobile game market (such as *Clash of Clans*) chose this solution for their game. Moreover, since we have the same company for both frontend and backend, our infrastructure management is greatly simplified.
Lastly, offered prices are really competitive.

Once this choice is done, we have little to no control over the software used by Amazon but we can choose between Windows and Linux.
We have chosen to go with the latter because of its stability, reliability and performance; as a matter of fact many articles online[1] emphasize how the aforementioned qualities are most frequently found in Linux system rather than Windows ones.
In order to integrate our game with the Linux servers there is a specific SDK provided by Amazon to develop for their platform.
One of the reasons why this choice is particularly convenient, is that the network API provided are compatible with Unity and can be accessed directly with C# code.
For both our frontend and backend infrastructure we rely on Amazon Web Services.
This results in an easier management of the infrastructure since a narrow set of skill is needed to maintain our global infrastructure.
We decided to opt for a payment plan that offers us a considerable discount on web services monthly fees by paying an advance upfront. The following costs are considered for reserved servers.

[1] www.pcworld.com (Five Reasons Linux Beats Windows for Servers: goo.gl/RQe23f)

Backend Component	Chosen Solution	Hardware (if any)	Reason of our choice
Web Server	AWS Elastic Beanstalk (t2.xlarge)	• CPU: x4 High Frequency Intel Xeon Processors (up to 3 Ghz) • Memory (Gib): 16 • Storage(Gb): EBS handled by Amazon	"*AWS Elastic Beanstalk [...] service automatically handles all the details such as resource provisioning, load balancing, auto-scaling, and monitoring.*" We decided to use this AaaS solution since it provides both an auto scaling and an integrated load balancing module; the aforementioned reasons provide us a good starting base to provide a responsive environment for our community that supports an highly variable amount of traffic. The hardware is a T2, which Amazon documentation identifies as perfect choice for web servers.
User Data Server	Amazon Web Services RDS (t2.large with Multi-AZ) x2	• CPU: x2 High Frequency Intel Xeon Processors (up to 3 Ghz) • Memory (Gib): 8 Dedicated EBS	*"Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud."* Since all payment data is stored externally, the only information about players that we must hold is their ID and emails, which requires little space, therefore a simple IaaS solution will be enough to accomplish that.
Matchmaking Server	Amazon Web Services DynamoDB (c4.2xlarge)	• CPU: x8 High frequency Intel Xeon E5-2666 v3 (Haswell) with 2,9 Ghz • Memory (Gib): 15 • Storage (Gb): EBS handled by Amazon	*"Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability."* This solution is focused on data throughput rather than storage, which is particularly convenient for a matchmaking server.

Match Host Server	Amazon Web Services GameLift (M4.2xlarge) x2	• CPU: x4 2.3 GHz Intel Xeon® E5-2686 v4 (Broadwell) processors or 2.4 GHz Intel Xeon® E5-2676 v3 (Haswell) processors. • Memory (Gib): 32 • SSD Storage (Gb): 50	*"Amazon GameLift is a managed service for deploying, operating, and scaling dedicated game servers for session-based multiplayer games."*. This service is tailored for our needs and spares us to build our own infrastructure out of remote servers. We are particularly interested in the fact that it offers low latency, which is a key factor for our game.

6.2 Scalability and Extensibility

The critical part of the backend is the match hosting which scalability is achieved with Amazon Web Services that can provide new computation power in a few minutes if needed.

6.3 Cost Estimation

Backend Component	Product	Advance Payment	Monthly Cost	Total (12 months)
Web Server	AWS Elastic Beanstalk (t2.xlarge)	€ 502	€ 42	€ 504
User Data Server x2	Amazon Web Services RDS (t2.large)	€ 250	€ 20	€ 240
Match Making Server	Amazon Web Services DynamoDb (c4.2xlarge)	€ 949 + € 137 (for DynamoDB)	79 € + € 9 (for DynamoDB)	€ 1,056
American Match Host Server (Oregon)	Amazon Web Services GameLift (M4.2xlarge)	€ 0	€ 152	€ 1,824
European Match Host Server B (Frankfurt)	Amazon Web Services GameLift (M4.2xlarge)	€ 0	€ 203	€ 2,436
Column Total		€ 1,838	€ 505	€ 6,060
Total cost for final delivery			€ 7,898	

7 Development

7.1 Hardware

Product	Cost	Quantity	Annual Total	Usage
Dell XPS Tower Silver Chassis	€ 920	8	€ 7,360	Tower computer for developers and IT staff, high performance for software development
Dell 24 Ultra HD 4K Monitor | P2415Q	€ 400	8	€ 3,200	Screen for developer's tower
KM714 Wireless Keyboard and mouse Combo	€ 70	8	€ 560	Input accessories for developers tower
Surface Pro 4 - 256GB / Intel Core i5	€ 1,100	3	€ 3,300	This tablet will be used by pr and marketing consultant as they need to work outside the office.
Surface Studio - 1TB Intel Core i7	€ 3,200	3	€ 9,600	This computers comes with integrated graphics tablet in the screen and are designed for artistic and design works.
Total cost			€ 24,020	

7.2 Software

Product	Monthly cost per seat	Quantity	Total (28 months)
Unity Pro	€ 98	5	€ 13,720
Visual Studio Professional	€ 40	3	€ 3,360
Adobe creative cloud all apps	€ 67	3	€ 5,628
Office 365 Suite (5 Licenses Pack)	€ 10	4	€ 1,120
Total cost			€ 23,828

7.3 Backup Plan

We decided to buy some cloud storage from amazon in order to keep a backup for our development assets, build versions, our codebase and so on.

Product	Cost per month	Total (28 months)
Amazon S3 Cloud Storage 25 TB	€ 548	€ 15,344

7.4 Cost Estimation

Development Category	Total cost
Hardware	€ 24,020
Software	€ 23,828
Backup Plan	€ 15,344
Total	€ 63,192

7.5 Development Gantt

8 External Services

We delegate the payment service to the publisher of our game: Google Play for Android and Apple Store for iOS. This way we can further simplify our network infrastructure and, since we are not keeping any sensitive data, we do not need to provide assistance about payment problems nor offer advanced security features.

8.1 Service Subscriptions

Product	Cost/year per seat	Transaction fee	Quantity	Total (3 Years)
Google Play License	€ 0	30%	1	One Time € 25
Apple Store License	€ 92	30%	1	€ 276
Total cost for final delivery				€ 301

8.2 Advertising

For two weeks before official release, we need a strong advertising boost.
Facebook ads fits quite particularly our game since it is a mobile title; as a matter of fact, we emphasize social network integration with our design choices too as can be inferred by the presence of social sharing within our app with dedicated buttons.
Google advertising offers free help and consulting for an expense of at least 10 euros a day, so we decided to seize that opportunity. Where and how to put advertising will be decided automatically by Google's algorithms, so we do not have to dig into the details; moreover we can monitor monthly how well it is performing and eventually decide to raise the ads, decrease them or even drop it if it might prove to be useless.

Product	Duration	Daily cost (up to 1,000,000 users)	Total
Facebook Ads	2 weeks	€ 714	€ 9,996
Google Ads	1 year	€ 14	€ 5,110
Total cost			€ 15,106

8.3 Help Desk

From a strictly budget point of view, we opted to outsource our help desk services for both the website and the game.
We plan to have 10,500 players on release but, unfortunately, the number of players who might have issues with our services is unpredictable. Therefore we opted for an outsourcing company that provides flexibility and price-per-ticket contract. In case of sudden and unexpected problems the help-desk service will scale accordingly but for budget reasons only, we assume that 2% might have some problems during a month; the amount here calculated reflects that assumption.

Product	Cost per answer	Monthly cost (210 users)	Total(12 Months)
Influx Scalable Help Desk Support	€ 2.48	€ 520.8	€ 6,249.6

8.4 Cost Estimation

External Service	Total cost
Service Subscriptions	€ 301
Advertising	€ 15,106
Help Desk	€ 6,249.6
Total	€ 21,656.6

9 Communication

9.1 Global Infrastructure Outline

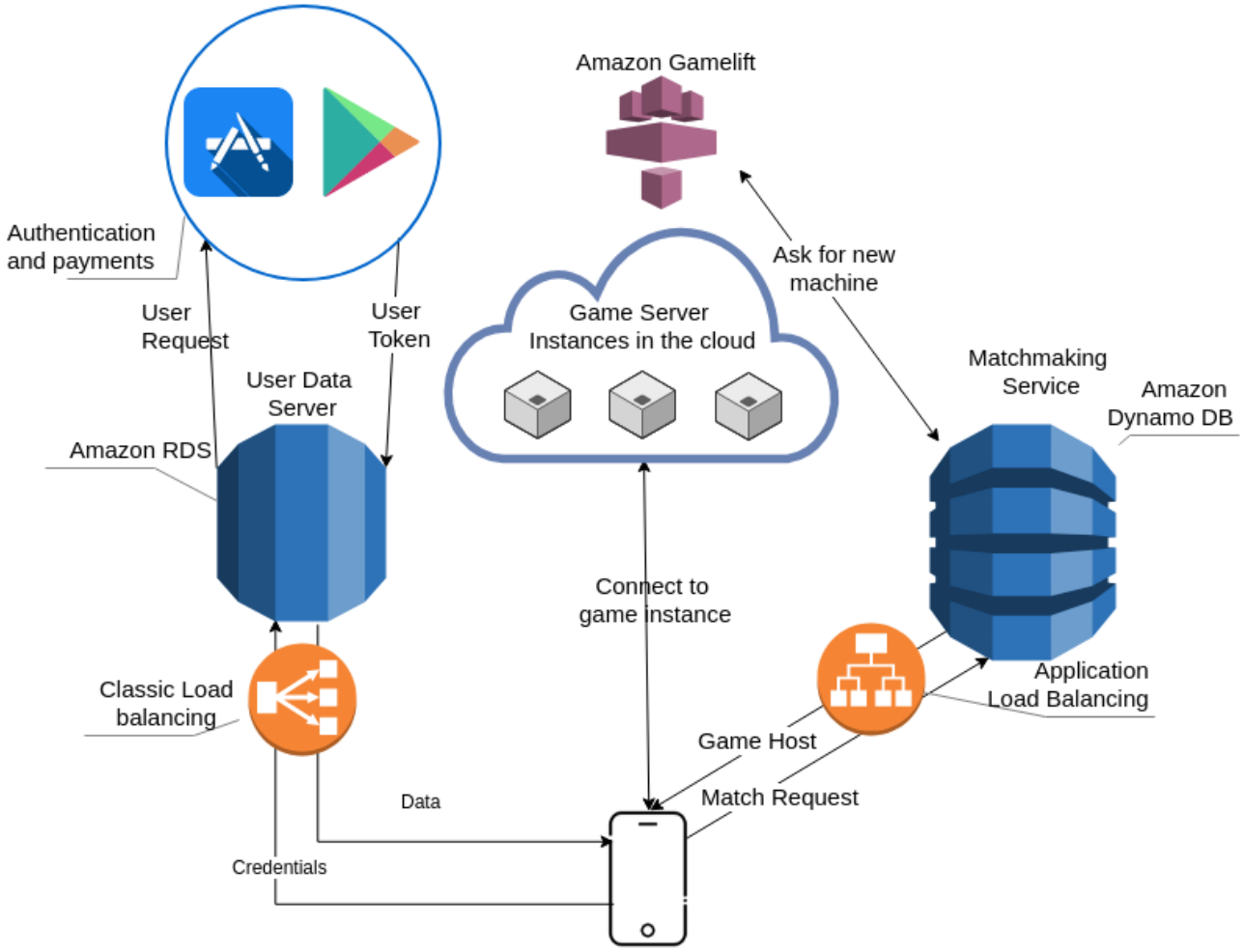


Fig.5 - Our architecture in detail

9.2 Network Requirements

- Server Network Requirements:
  Low-latency optimized machines and services with up to 800 Mbps
- Client Network Requirements:
  Clients need at least a 3G connection in order to play our game

9.3 Cost Estimation

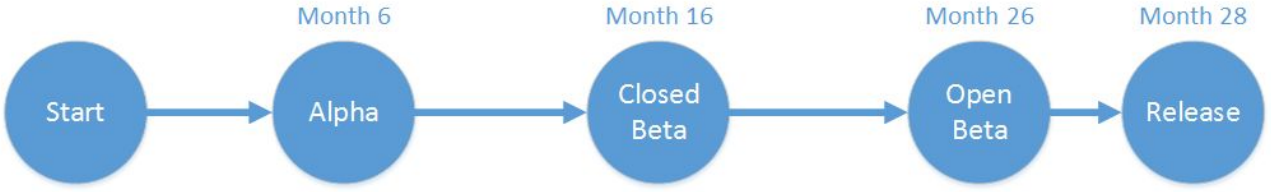
Since we do not own any server and we rely on a cloud architecture, we do not have any additional communication costs to pay.

10 Delivery

10.1 Estimated Delivery Time

We suppose to develop the basic software in 20 months; the following 6 months are needed to develop in-game shop, Google Play and Apple Store integration and the development of our website.

The first playable alpha will be available in the 6th month, the closed beta in 16th month and the open beta in the 26th month.
In the 28th month the game will be ready for the distribution.
The aforementioned development milestones can be seen in the following linear graph.



We assume that external events may delay delivery for up to 8 months.
The indicated timeline is abstract; even though the final decision for a release date has to be planned carefully with a marketing specialist, we do believe that a release during Christmas holidays should be our best bet, since everyone is receiving new smartphones as presents and they are eager to try new games on them.

10.2 Delivery Platform

The software will be distributed free at the same time on Google Play store for Android and on the App Store for iOS. Direct download costs are covered by the money we pay for the aforementioned publishers.

10.3 Delivery Methodology

The game will be released on the mobile stores with full contents. Later on there will be added contents, like new maps and game modes, patches and updates on the same publishing platform as shown in the Game Design Document.

10.4 Cost Estimation

As stated in the chapter, every delivery cost is payed through the licenses and the 30% fee for each transaction. Therefore, there are no additional charges.

11 Staff

11.1 For Infrastructure Setup

- Amazon service developer

11.2 For Infrastructure Management

- System & Network Administrator
- IT Security Consultant

11.3 In Game

- Senior game designer
- Junior game designer
- Senior game developer
- Junior game developer x2
- Senior graphic artist
- Junior graphic artist
- Sound engineer and composer

11.4 Other

- Forum moderator
- Marketing advisor

11.5 Workspaces

We plan to rent for lease some offices and furnitures from external providers, with at least 100 square meters area. Given an extensive research on our area of interest and after comparing many prices offered by different agencies, we estimate a cost similar to 50,000 euros for an all-inclusive office solution in Milan (bills, internet, insurance, cleaning etc. all included as shown above).

All-inclusive Office in Milan (100 square meters)	**Cost per month**	**Total Cost (28 months)**
Rent	€ 2,080	€ 58,240
Condominium fees	€ 166	€ 4,648
Charges (electricity, gas, etc.)	€ 125	€ 3,500
Internet and phone bill	€ 100	€ 2,800
Receptionist and contact center	€ 830	€ 23,240
Furnitures	€ 416	€ 11,648
Installation costs and office maintenance	€ 330	€ 9,240
Legal consulting and registrations	€ 100	€ 2,800
Housekeeping	€ 166	€ 4,648
Column Total	€ 4,313	€ 120,764
Total cost		€ 120,764

11.6 Cost Estimation

Product	Monthly salary	Quantity & Time	Total
Development Staff			
Amazon Service Developer	€ 2,916	1 x 12 months	€ 34,992
Senior Game Designer	€ 2,916	1 x 28 months	€ 81,648
Junior Game Designer	€ 2,333	1 x 28 months	€ 65,324
Senior Game Developer	€ 2,916	1 x 28 months	€ 81,648
Game Developer	€ 2,333	2 x 28 months	€ 130,648
Graphic Artist	€ 2,083	1 x 12 months	€ 24,996
Sound Engineer and Composer	€ 2,333	1 x 6 months	€ 13,998
Maintenance Staff			
System & Network Administrator	€ 2,916	1 x 24 months	€ 69,984
Forum Moderator	€ 1,333	1 x 12 months	€ 15,996
IT Security Consultant	€ 3,000	1 x 12 months	€ 36,000
Marketing Staff			
Marketing advisor	€ 2,333	1 x 6 months	€ 14,000
Annual Total			€ 569,234

12 Potential Security Issues

Security issues are a serious matter for every online service, and online video games make no exception.

12.1 Game-specific Issues

Every single calculation that is crucial to our gameplay is server-authoritative, therefore we try to make cheating as hard as possible to any user who will try to play with a modified client. Should we have some unexpected way of cheating, users can complain on our website and server through our help desk service. These reports can be used by our development team to understand the issue, find a solution and patch it as soon as possible.

12.2 Other Issues

Since we delegate user credential and payments to external services, we have no critical security issues to address in that regard.
Players' identification is handled using external registration with reliable platforms (Google Play or Apple Store), so we just need to store what is strictly related to our game though that data is protected by the contract agreement with Amazon Web Services. In particular, possible denial of service attacks to hinder game availability are mitigated by the Amazon GameLift service.
Moreover, our users' identities is already validated externally and we do not need to execute any further test.
Other unpredictable security issues might occur and that is the reason why we hired an IT security consultant.

13 Costs Summary

13.1 Total Project Cost

Macro Category	Total Delivery Cost
Frontend	€ 384
Backend	€ 7,898
Development	€ 63,192
Staff	€ 569,234
External Services	€ 21,656
Total	€ 662,364

13.2 Revenue Estimation

13.2.1 Ad Revenue

In order to correctly evaluate how ads can generate income, we applied common knowledge marketing rules with tools developed especially for this kind of estimation in mobile videogames; this led us to produce our own estimators that will be shown here. First of all, given our daily expected user number in each phase of the project, we estimate that each user will experience 2 sessions of gameplay with 2 matches each, for approximately 4 game matches in total per day. Since our ads are shown right after a match is concluded, we can assume that the number of ads seen in total per day can be calculated as follows:

*daily ads per user = number of daily sessions * number of matches per session*

The calculation for total number of ads shown daily is then calculated using that number as follows:

*total number of ads shown = daily ads per user * daily active free users*

Then we do use an eCPM (effective cost per mille) of € 2, as an average for different estimators found online for the western mobile market. This is the formula we will be using in the final revenue calculation for the ad-revenue entry:

*total daily ad revenue = total number of ads shown * eCPM / 1000*

13.2.2 In-app Purchases

Using the following table, we show how our approximations are done.
We estimate that 10% of total user will become premium each month and we give an estimation of how much both types of player might spend with in-app purchases.

(Google Store and Apple Store 30% already applied)	Free User (90% of total players)	Premium User (10% of total players)
Ad removal upgrade	€ 2.093 (10% of free players)	Already paid
Average in-app purchase revenue per user per month	€ 2.1	€ 3.5

Total Revenues (Monthly)							
	Free Users (90% of total players)			**Premium Users (10% of total players)**			**Monthly Total**
On Release 10,500 total users 3,500 daily active users 9,450 free users 1,050 premium users	**(Ad revenue)** **(Upgrade revenue)** **(in-app purchases)** **(Monthly Total)**	€ 756 € 1,977.88 € 19,845 € 22,578.88	+ + =	**(Ad revenue)** **(Upgrade revenue)** **(in-app purchases)** **(Monthly Total)**	€ 0 € 0 € 3,675 € 3,675	+ + =	€ 26,253.88
After 4 Months 29,000 total users 9,000 daily active users 26,100 free users 2,900 premium users	**(Ad revenue)** **(Upgrade revenue)** **(in-app purchases)** **(Monthly Total)**	€ 1,944 € 5,462.73 € 54,810 € 62,216.73	+ + =	**(Ad revenue)** **(Upgrade revenue)** **(in-app purchases)** **(Monthly Total)**	€ 0 € 0 € 10,150 € 10,150	+ + =	€ 72,366.73
After 8 Months 40,000 total users 12,000 daily active users 36,000 free users 4,000 premium users	**(Ad revenue)** **(Upgrade revenue)** **(in-app purchases)** **(Monthly Total)**	€ 2,592 € 7,534.8 € 75,600 € 85,726.8	+ + =	**(Ad revenue)** **(Upgrade revenue)** **(in-app purchases)** **(Monthly Total)**	€ 0 € 0 € 14,000 € 14,000	+ + =	€ 99,726.8

13.3 Break-Even Estimation

With the assumptions we have seen above, we have calculated the income generated by our model and plotted it into a graph. It can be shown that applying the numbers calculated in the previous chapter, we reach break-even point approximately 10 months after release. We show how much revenue we generate in the first year applying our model without further increasing our player-base. We specify that our model is an underestimation of our revenue because we use as a monthly revenue the one calculated with the number of players at the start of the session (for example, monthly revenue in the 6th month is the same as the one on the 5th month, even if our player base is expected to grow continuously).

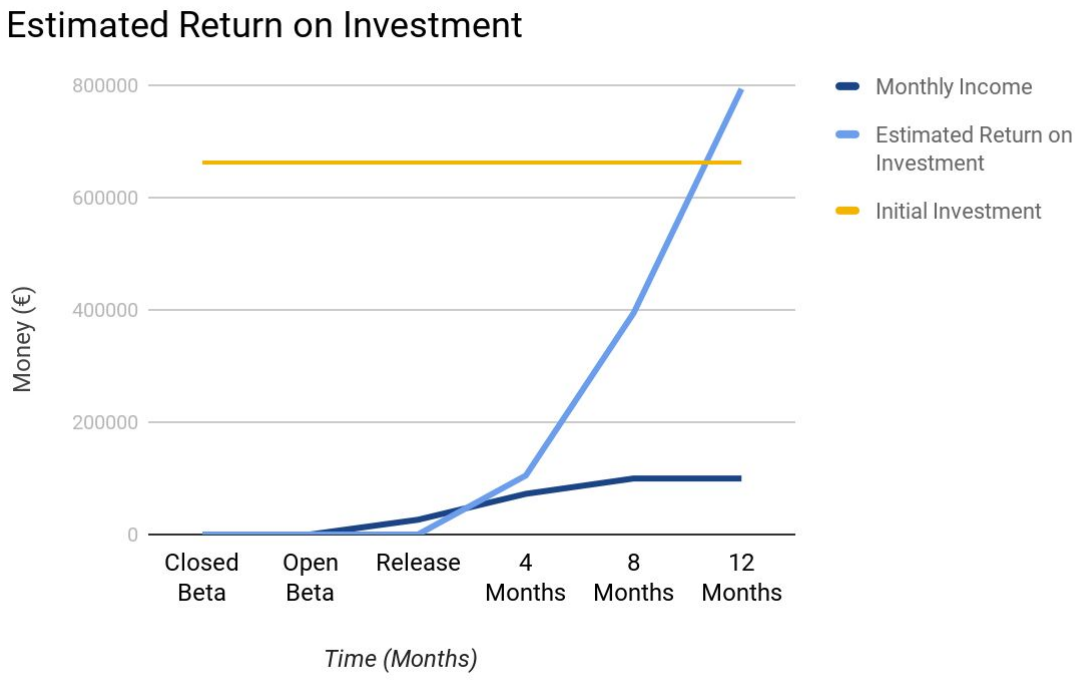


Fig.6 - Estimated return on investment

13.4 Further Considerations

Since it emerged from our study that in-app purchases are our most profitable income, our future planning will include new buyable in-game elements, in order to always give the players something new to buy, should it be a new arenas, new skins or different game modes.

The low overall revenue provided by premium users can be explained with our assumption that only 10% of free users will pay the upgrade; a more aggressive marketing strategy might see some temporary discounts for the upgrade in order to maximize the number of premium users.

This would be positive for two reasons:

1. In the short run, the initial burst of income given by the upgrade fee.
2. In the long run, a premium user spends more on average (1.5 times more) than a free user. This difference definitely outweighs the loss in the ad-revenue and improves our business stability since a premium player is less likely to uninstall the game once he/she has paid for it.