Bilkent University

Department of Computer Engineering

CS 492 - Senior Design Project

Project Name: Deeplay.io

Final Report

Team "*Ludens*"

Project Group Members:

F. Serdar Atalay, Ekinsu Bozdağ, Gökcan Değirmenci,

Onur Sönmez, Gökçe Özkan

Supervisor: Prof. Dr. Halil Altay Güvenir

Jury Members: Asst. Prof. Dr. Shervin R. Arashloo, Prof. Dr. Uğur Güdükbay

Innovation Expert: Mustafa Sakalsız

Submitted on May 9, 2019

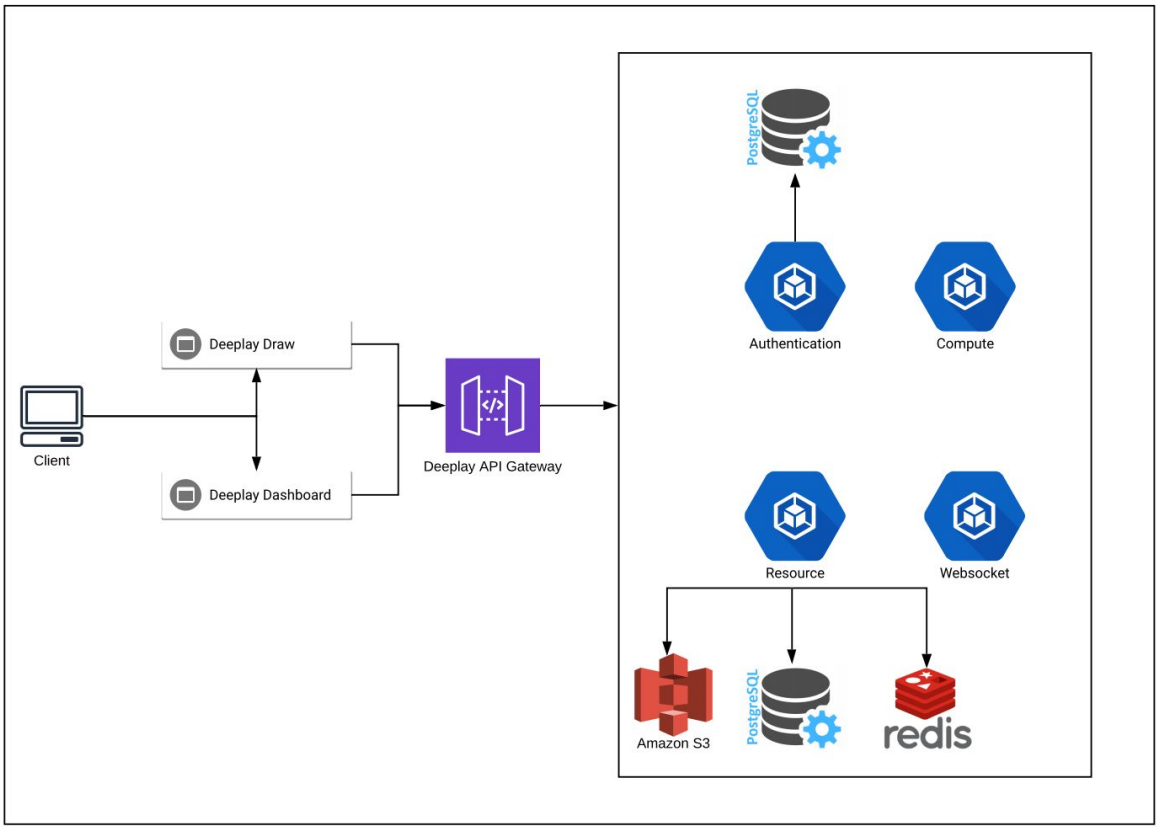
1. Introduction

In the field of computer science, there are many companies, engineers and data scientists that extensively use deep learning. As the amount of necessary and appropriate data to be used during training increases, the performance increases with deep learning algorithms. With the current advancements on machine learning, especially on its subset area of deep learning, intelligent applications become extremely data-hungry. The need for uncompromised quality, actionable data increases everyday. That is to say, data becomes the “new oil” of our era. People label their data to do training and they need more and more data to get the most accurate results for their systems. Labelling data does not require high level education in the field. Thus, when requirements are well understood, any person with adequate hand-eye coordination can label training data. With crowdsourcing concept, our project promises entertainment to people, while making them help those who are interested in or working on deep learning and computer vision.

Deeplay is an intelligent data labeling platform, which connects the client and crowd with the data labelling process. It aims to create labels for dataset of clients with increased accuracy in results. Deeplay promises entertainment to crowd, while they label images. Having three options for data labeling, Deeplay offers the users extended functionality. With Single User option, the user can use the platform as a tool, while multiplayer option provides an entertaining way for users to create correct labels. The third option, which is collaborative, provides real time classification with another user, so that the images with many possible labels can be tackled with multiple users at the same time.

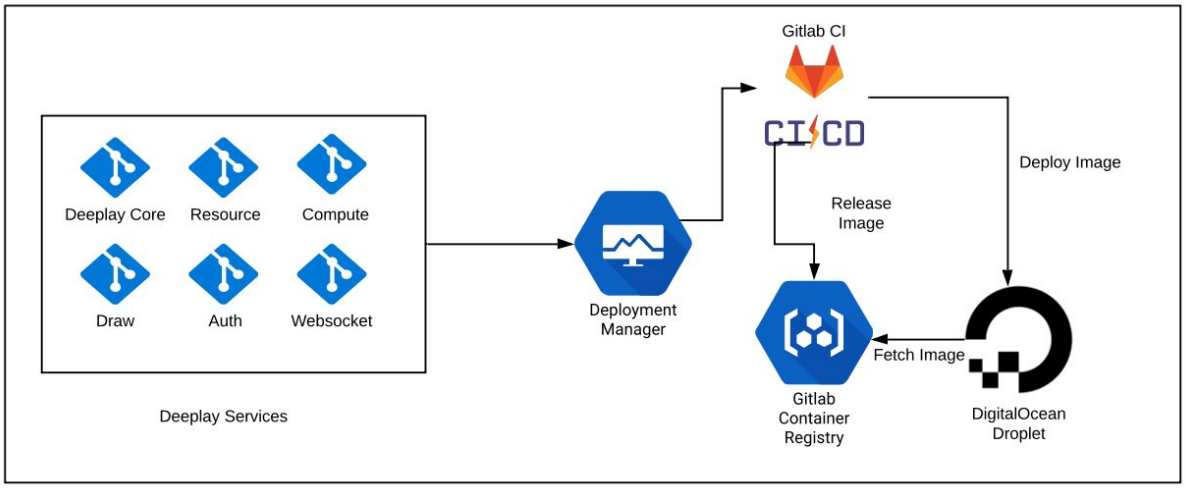
2. Final Architecture and Algorithm Design

Deeplay implements **Containerized Microservices Architecture** and applies design best practices to provide end-to-end data labeling platform.



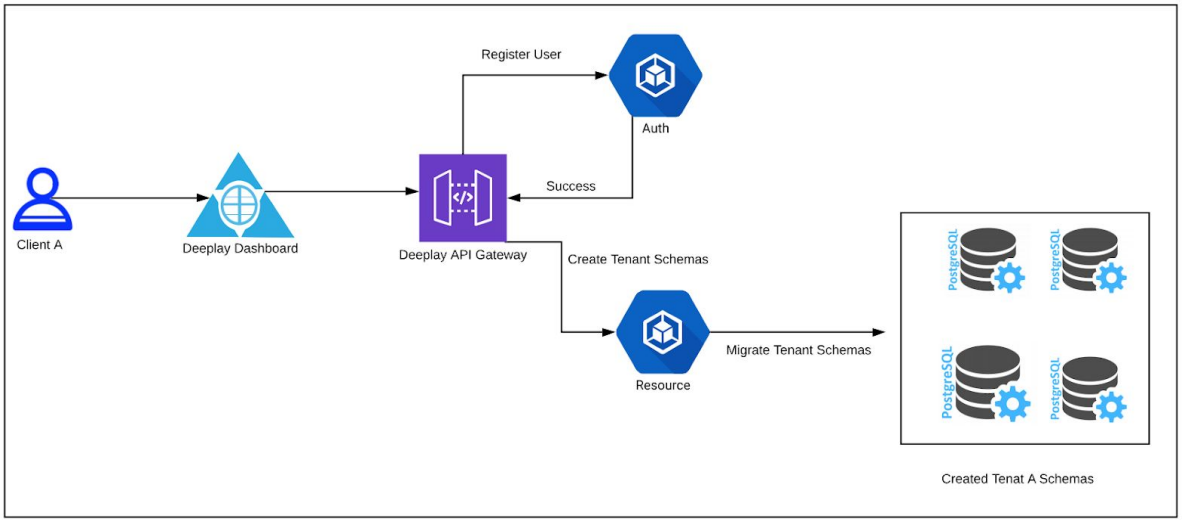
Architecture Overview

General architecture of Deeplay is shown on above diagram. It consists of two web applications including Deeplay Draw and Deeplay Dashboard. Each microservice is containerized using Docker which is a world-leading CaaS(Container-as-a-Service platform). Web applications are written with React and Redux. These web services interact with API Gateway which is a basic proxying server to access and manage other microservices. In order to shard multi-tenant Deeplay web services, PostgreSQL is used as a main storage system. Raw datasets uploaded from clients are stored in cloud by using Amazon Simple Storage Service(Amazon S3). All services are powered by caching mechanism and Redis is used for this purpose.



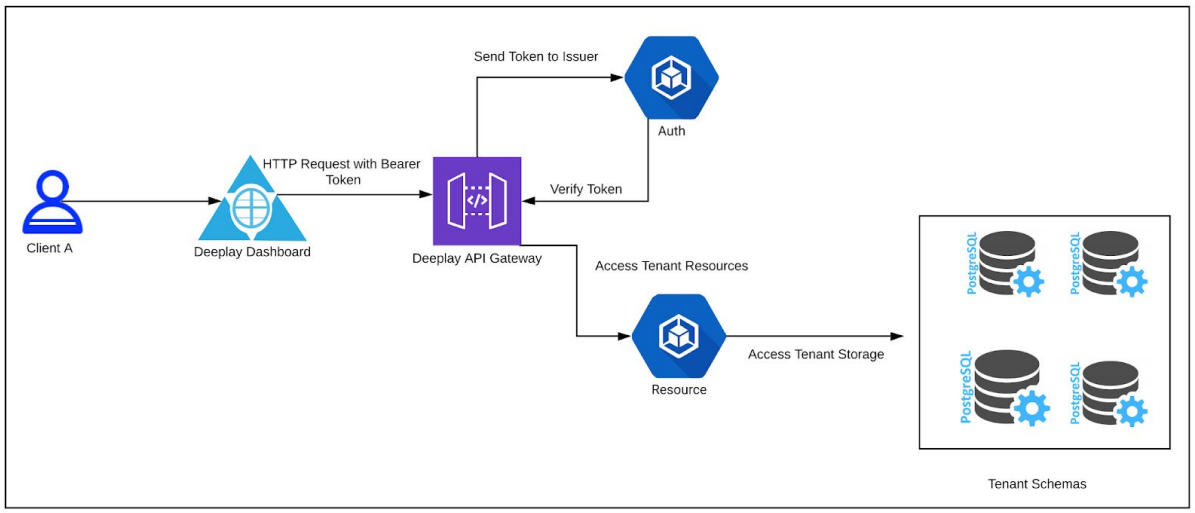
Deployment Overview

We automated deployment process and used Gitlab CI(Continuous Integration) tool. Each service is deployed on DigitalOcean droplet.



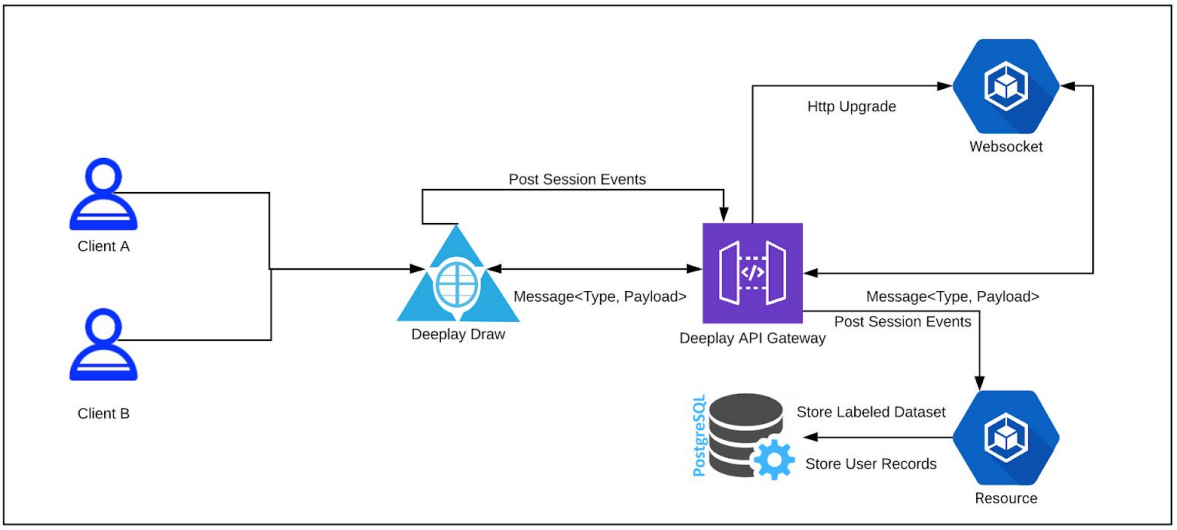
Tenant Creation Overview

Tenant creation overview is shown on above diagram. After authorizing users, sql schemas created for each client. This provides fully encapsulated environment for each tenant. This guarantees that user resources cannot be accessed by another user and speed up user activities. Client only needs to access his tenant schemas and thereby its access time will be much lower than the standard sql structure which contains all user records in same schema.



Tenant Access Overview

Tenant may access his resources after the bearer token is verified by authorization service. All tenant-specific actions are handled with this secure and industry-standard architecture.



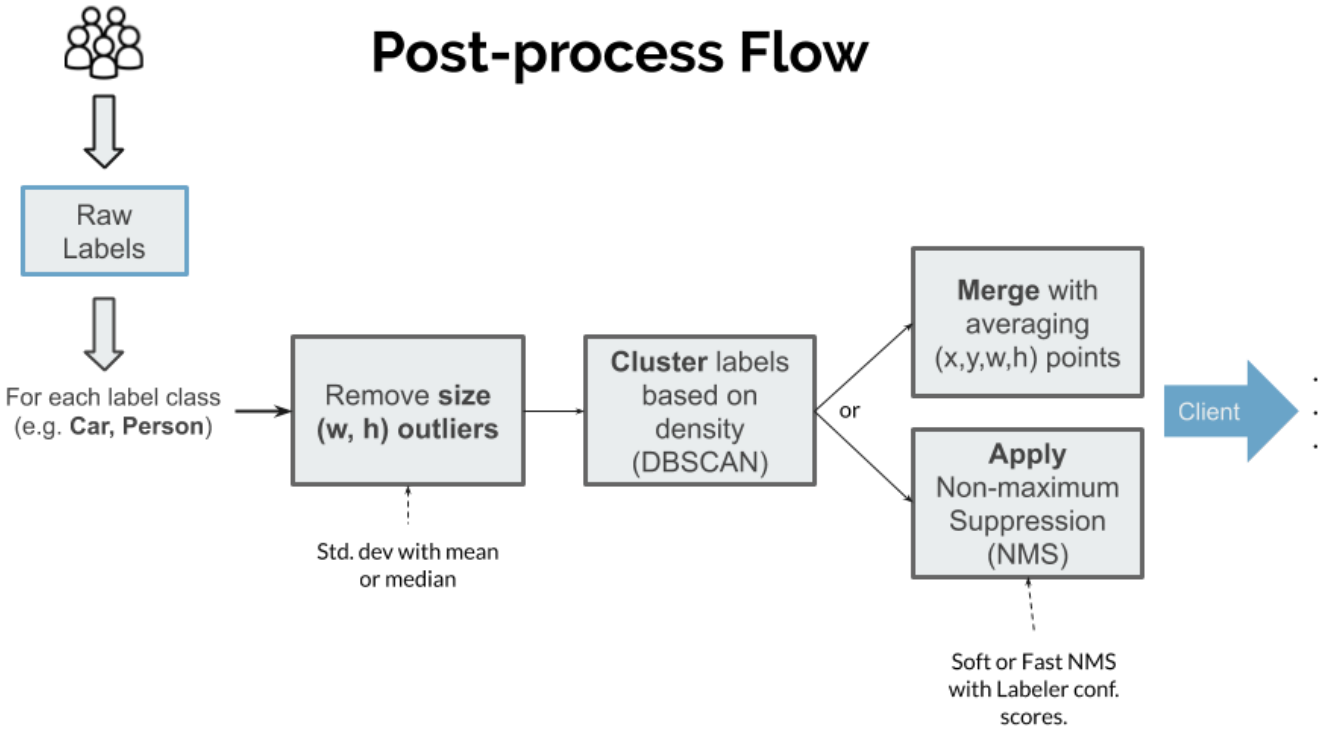
Data Labeling Session Overview

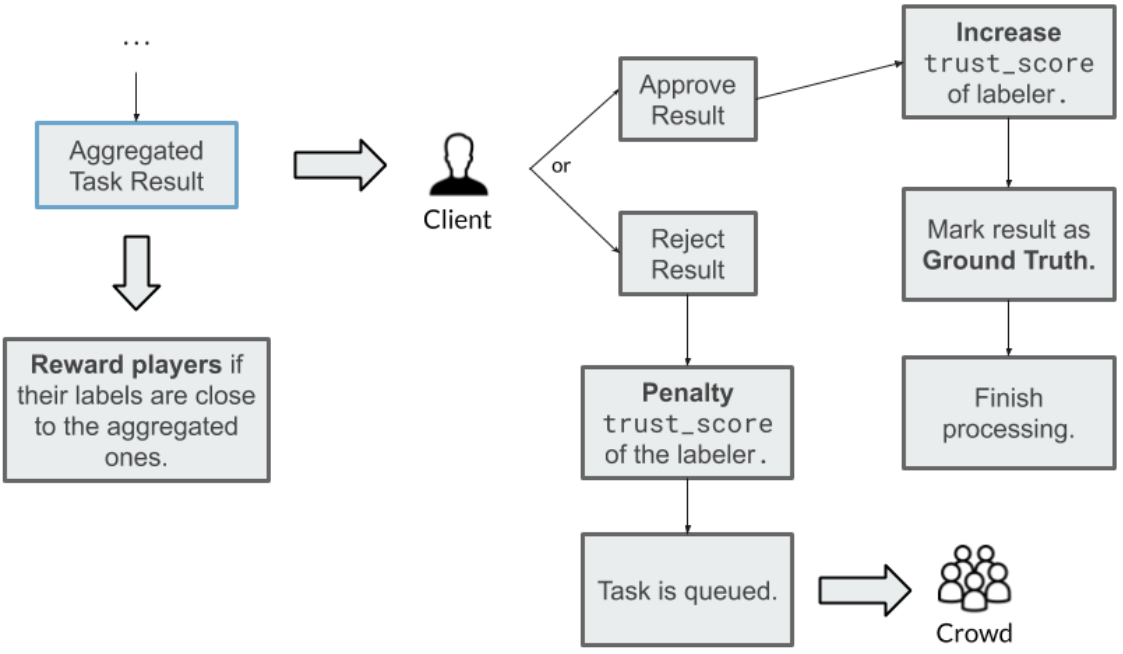
Raw data is labeled on Deeplay sessions and we use full-duplex communication protocol, “Websocket” for providing real-time labeling tool. Clients interact with each other via message passing and session results are stored on client-specific task schemas by using

PostgreSQL. This protocol is using single HTTP request and all other actions are handled by TCP communication mechanism.

2.2. Core Algorithms

In this subsection, we present diagrams related to the Algorithms that we use to **cluster**, **aggregate, filter** and **reduce** raw data, remove outliers and detect spammers.




...
Aggregated Task Result
Reward players if their labels are close to the aggregated ones.
Client
or
Approve Result
Reject Result
Increase trust_score of labeler.
Mark result as Ground Truth.
Finish processing.
Penalty trust_score of the labeler.
Task is queued.
Crowd

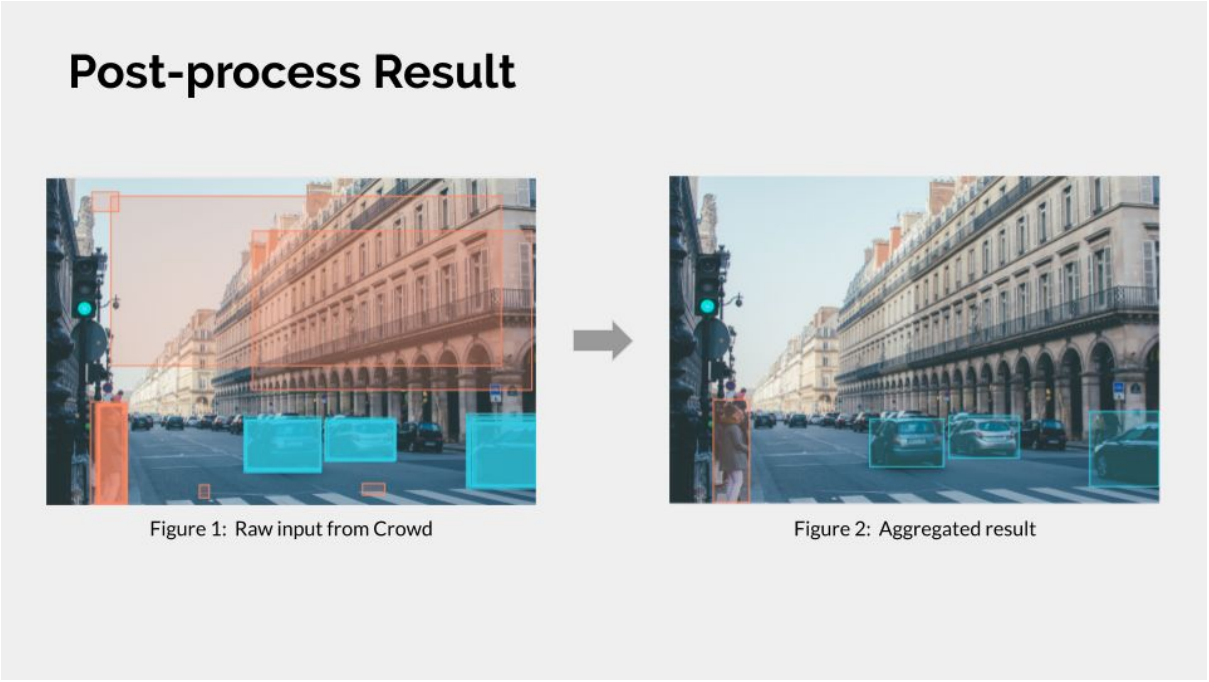


Figure 1: Raw input from Crowd

Figure 2: Aggregated result

3. Impact of Engineering Solutions Developed in the Project

3.1 Global Impact

Many companies and researchers working on Computer Vision and deep learning require ground truth for their data and label their dataset. In the figure below, you can see the place of deep learning in technology today. In order to do these, people use different classification and segmentation tools, however, as this area improves, the amount of data keeps increasing and it gets more and more time consuming to label images with a limited number of people. With the help of deeplay, those people do not need to spend many time on labeling, but just approving or rejecting the labeled images to increase the accuracy of the results. Making someone outside the company or research group label the data may not be preferred because of economical side and privacy issue. At this point, Deeplay promises a the security of client data and does not require money for data labelers.

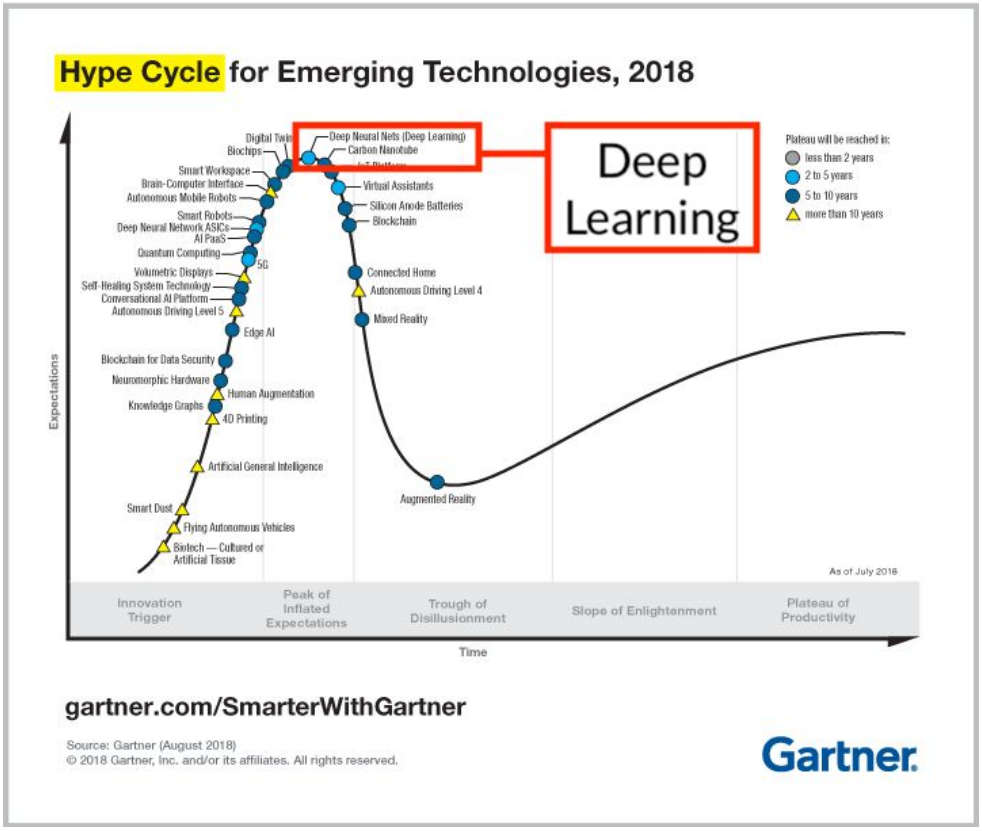


Figure [**2**]: Hype Cycle for Emerging Technologies, Gartner. [4]

3.2 Social Impact

Nowadays, many people at different ages has computers and spend their time with online games and applications for fun. As a different way of entertainment, Deeplay provides a very simple score based game to a broad range of users, by providing different options such as single user, multiplayer and collaborative game. Deeplay is a new approach to game area by making people help scientists, researchers and developers while playing game or using the simple tool option just of the platform to spend their time for entertainment.

4. Contemporary Issues Related with the Area of the Project

One of the most problematic part of Computer Vision (CV), in terms of creating ground truth for data, is to have increased accuracy in data labels. To ensure the accuracy, Deeplay sends the images to as many people as the client demands. As the crowd labels images, totally wrong results are removed and the relevant results are aggregated to get the best label. The client can approve or reject the result of the task, which ensures the last verification point is client and s/he can decide the accuracy of the labels.

Another problematic issue is that different data can require different labelling types. For now, Deeplay provides image classification. However, with further developments, Deeplay is compatible with text classification and image segmentation.

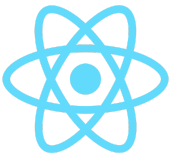
As the main element of this area is data, data privacy is a serious issue to be tackled in this project. Deeplay ensures the security of the client's data with end-to-end encryption. No data is stored in the system and it cannot be reached by other users.

5. New Tools and Technologies

5.1. Node.js

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser.

5.2. React, Redux.js



React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies.

Redux is an open-source JavaScript library for managing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Similar to Facebook's Flux architecture, it was created by Dan Abramov and Andrew Clark.

5.3. TypeScript

TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript, and adds optional static typing to the language. TypeScript is designed for development of large applications and trans-compiles to JavaScript.

5.4. Golang

Go is a statically typed, compiled programming language designed at Google by Robert Griesemer, Rob Pike, and Ken Thompson. Go is syntactically

similar to C, but with memory safety, garbage collection, structural typing, and CSP-style concurrency.

5.5. Python (NumPy, Scikit-Learn)

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

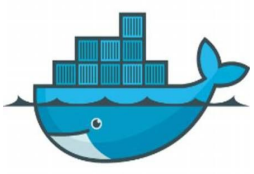
5.6. PostgreSQL

PostgreSQL is an open-source relational database management system emphasizing extensibility and standards compliance. It can handle workloads ranging from single-machine applications to Web services or data warehousing with many concurrent users.

5.7. Redis

Redis is an in-memory data structure project implementing a distributed, in-memory key-value database with optional durability. Redis supports different kinds of abstract data structures, such as strings, lists, maps, sets, sorted sets, HyperLogLogs, bitmaps, streams, and spatial indexes.

5.8. Docker



Docker is a computer program that performs operating-system-level virtualization. It was first released in 2013 and is developed by Docker, Inc. Docker is used to run software packages called containers.

5.9. AWS S3

Amazon S3 or **Amazon Simple Storage Service** is a "simple storage service" offered by Amazon Web Services that provides object storage through a web service interface. Amazon S3 uses the same scalable storage infrastructure that Amazon.com uses to run its global e-commerce network.

5.10. Google Cloud Firebase

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of October 2018, the Firebase platform has 18 products, which are used by 1.5 million apps.

5.11. Websockets

WebSocket is a computer communications protocol, providing full-duplex communication channels over a single TCP connection. The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011, and the WebSocket API in Web IDL is being standardized by the W3C. WebSocket is a different protocol from HTTP.

5.12. Gitlab

- Source code version management, collaboration.

5.13. Digital Ocean

- Server hosting and deployment.

5.14. Slack & Trello

- Team communication and synchronization.

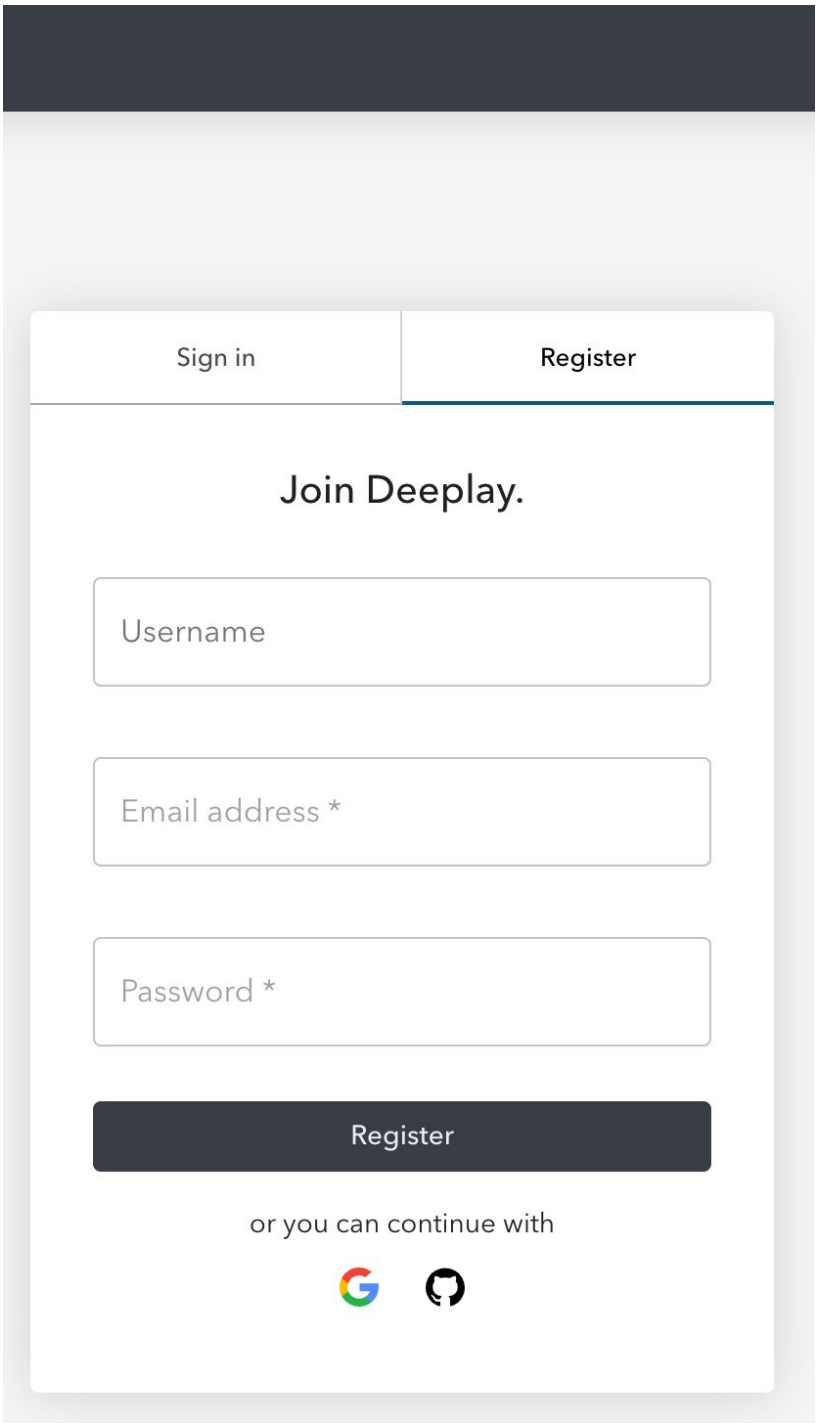
6. Resources

- Real-time communication with WebSockets

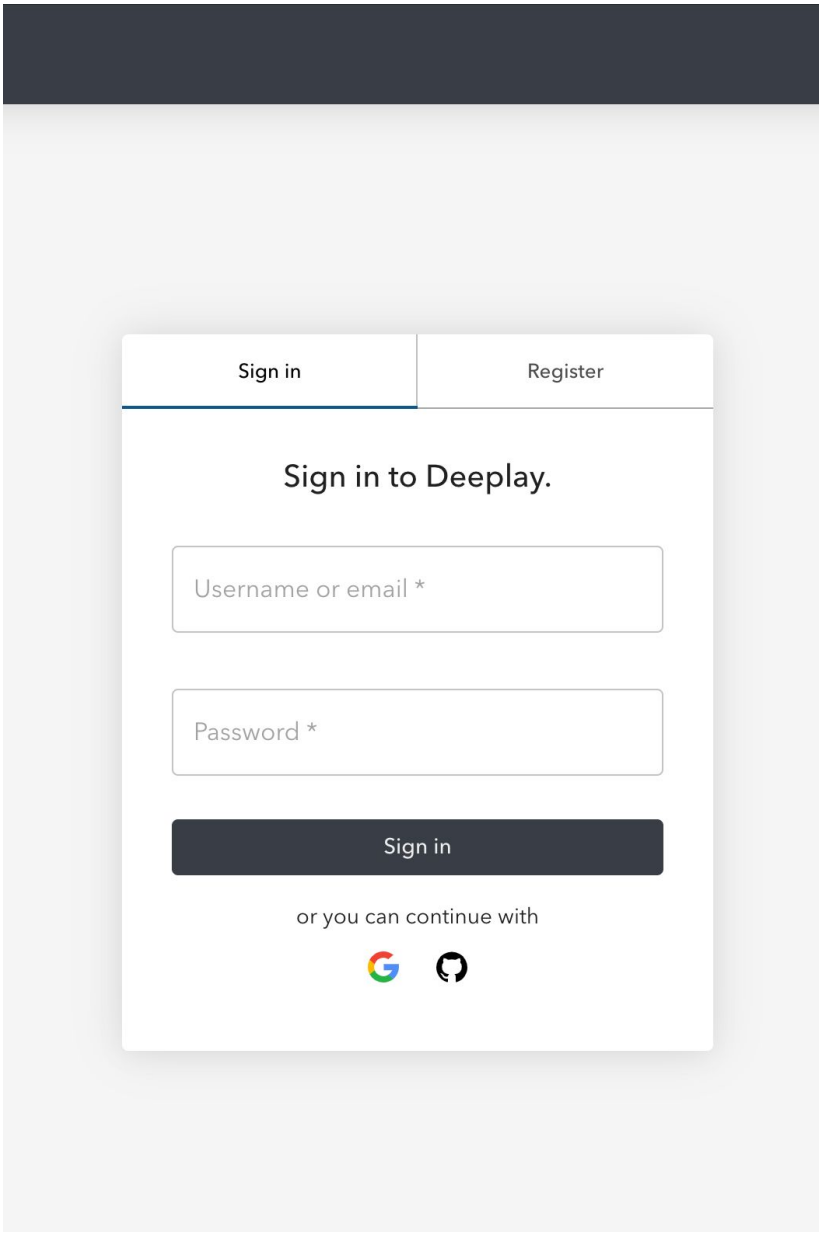
    - GoDoc/websocket [4]
- Microservices Architecture Design
    - Microsoft Docs [1].
    - Reflectoring.io Blog [2].
    - Microservices.io [3].
- Literature Review on Crowdsourcing, Computer Vision, Clustering.
    - *Grouper: Optimizing Crowdsourced Face Annotations [5].*
    - *Human Computation for Object Detection. [6].*
    - *Soft-NMS [7].*
    - For implementation: scikit-learn Docs. [8].

7. User Manual

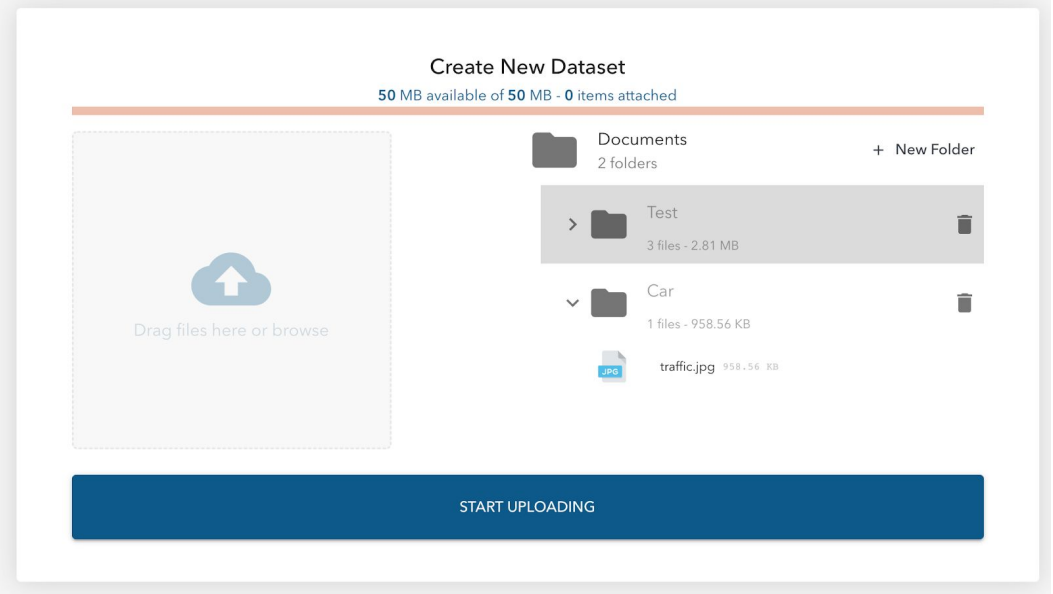
7.1. Register



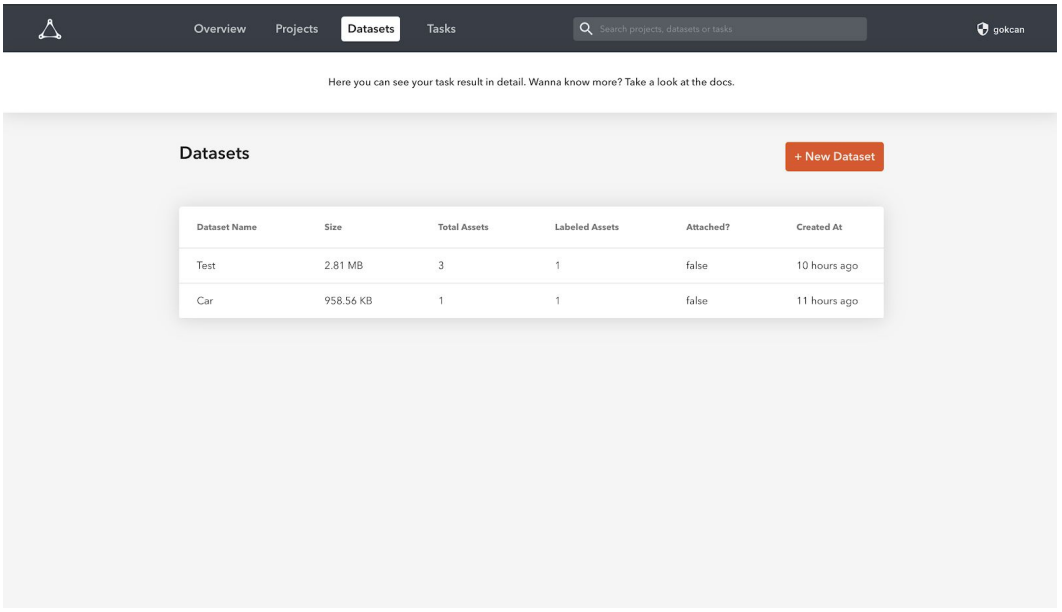
7.2. Sign In


Sign in
Register
Sign in to Deeplay.
Username or email *
Password *
Sign in
or you can continue with

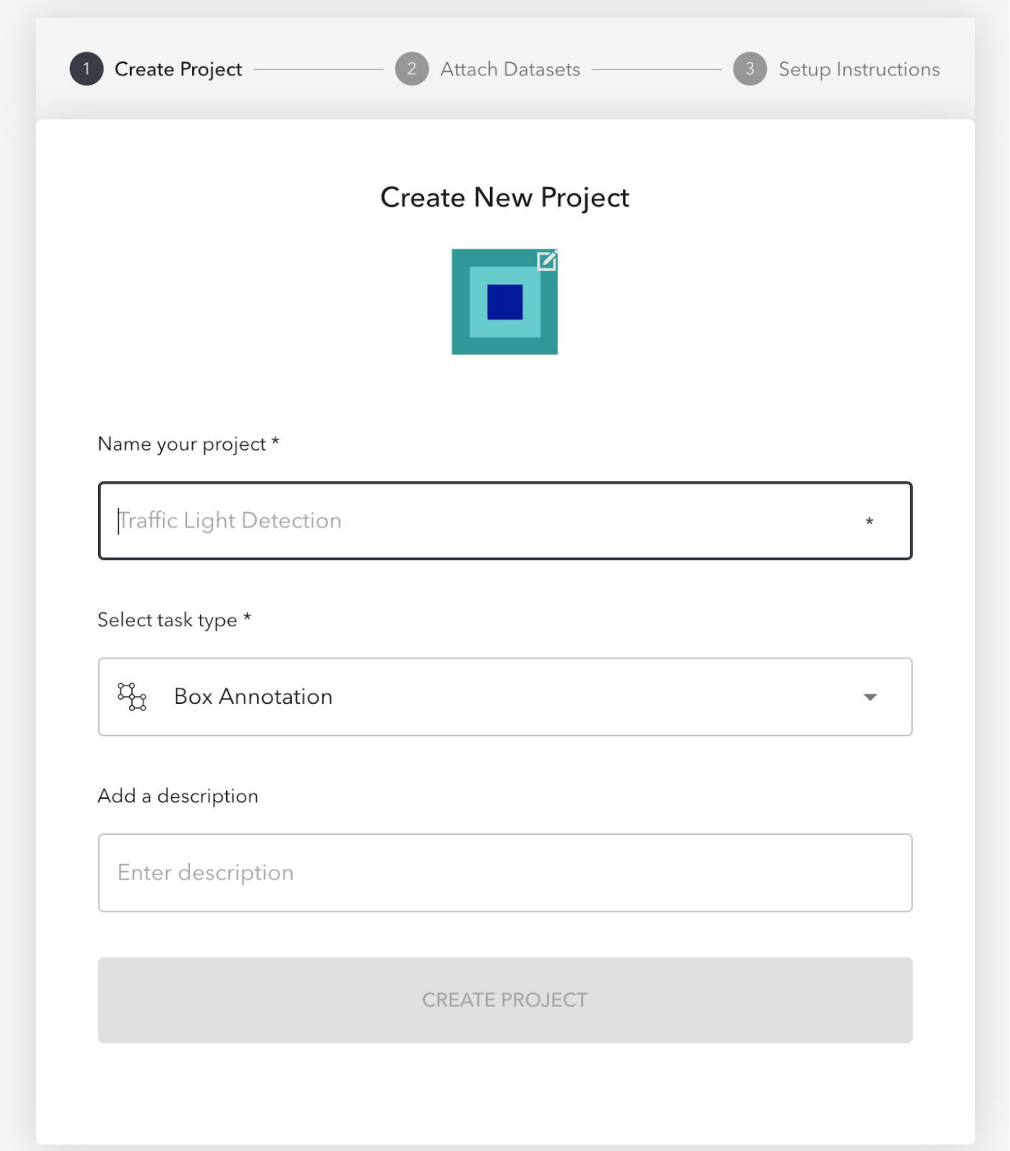
7.3. Upload Dataset



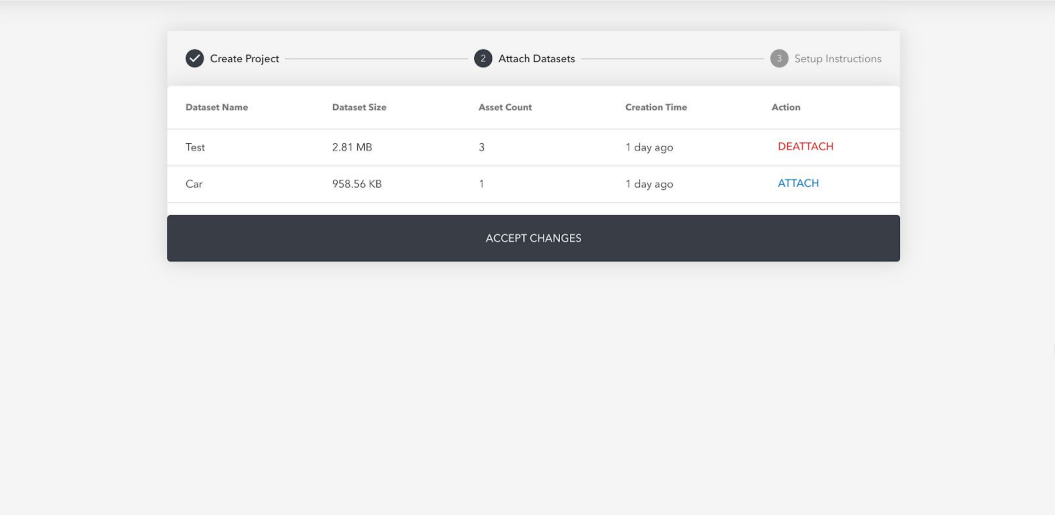
7.4. Show Datasets



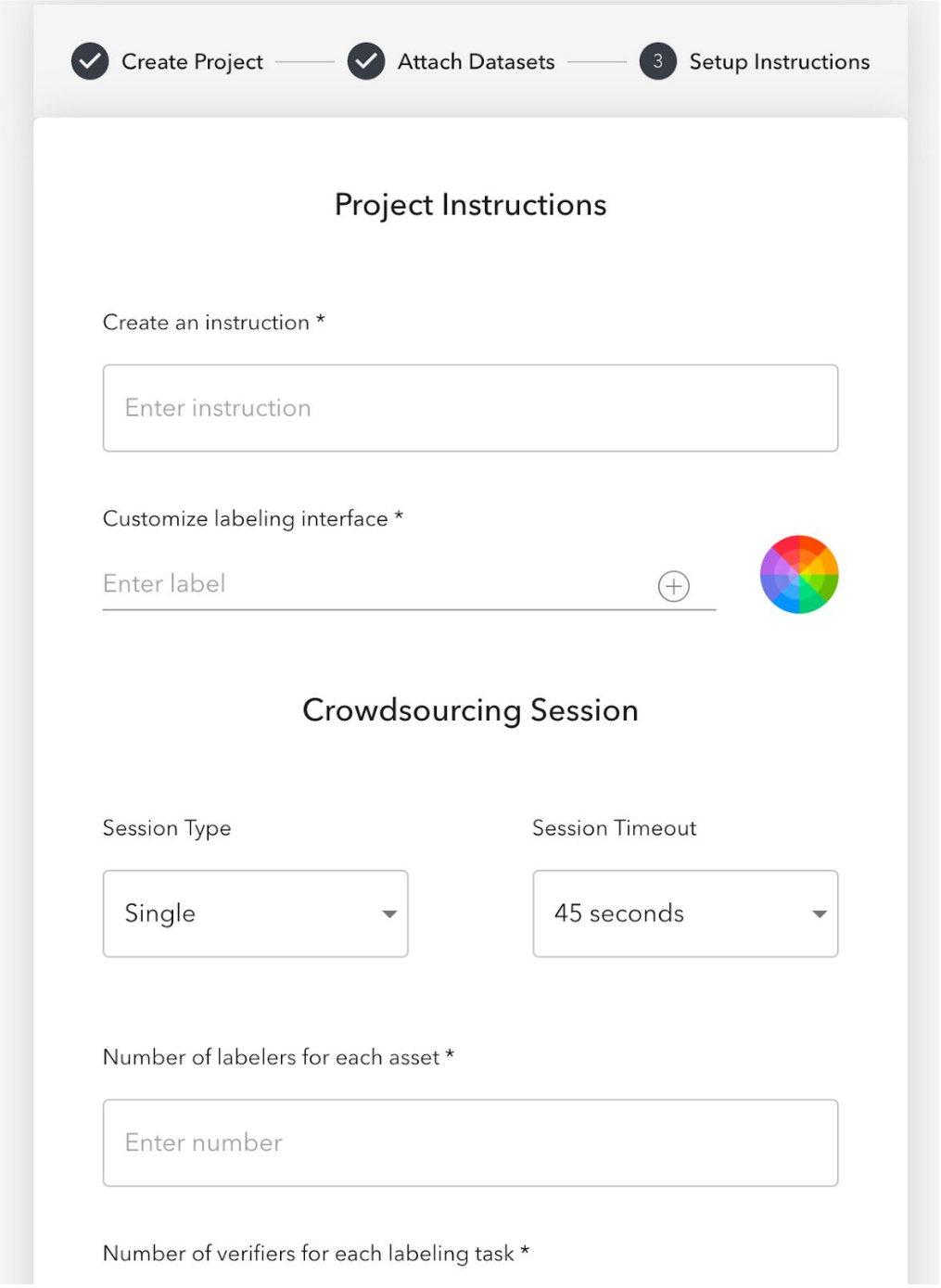
7.5. Create a Project



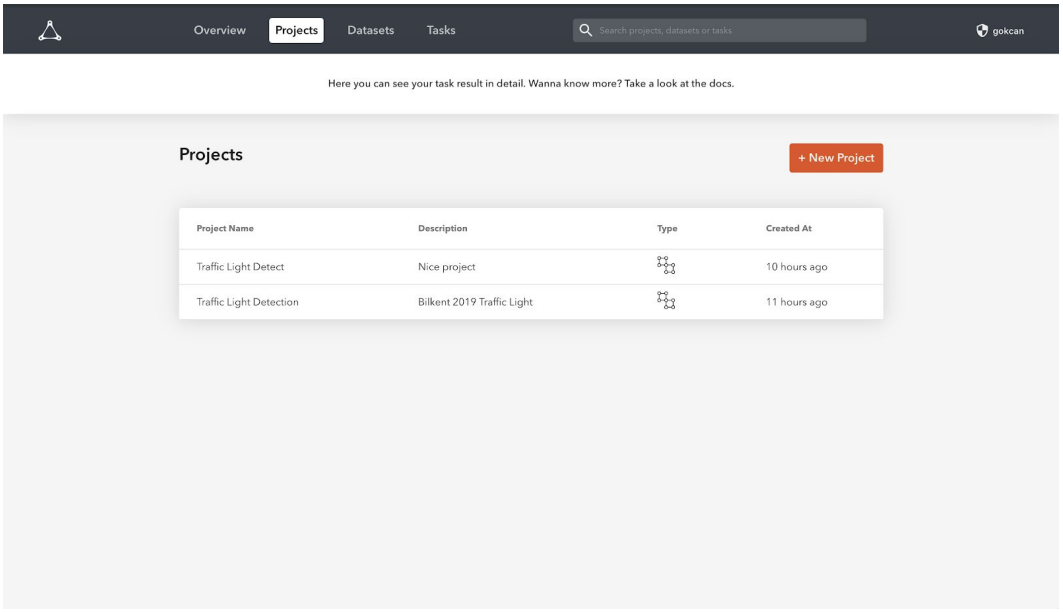
7.6. Attach Datasets



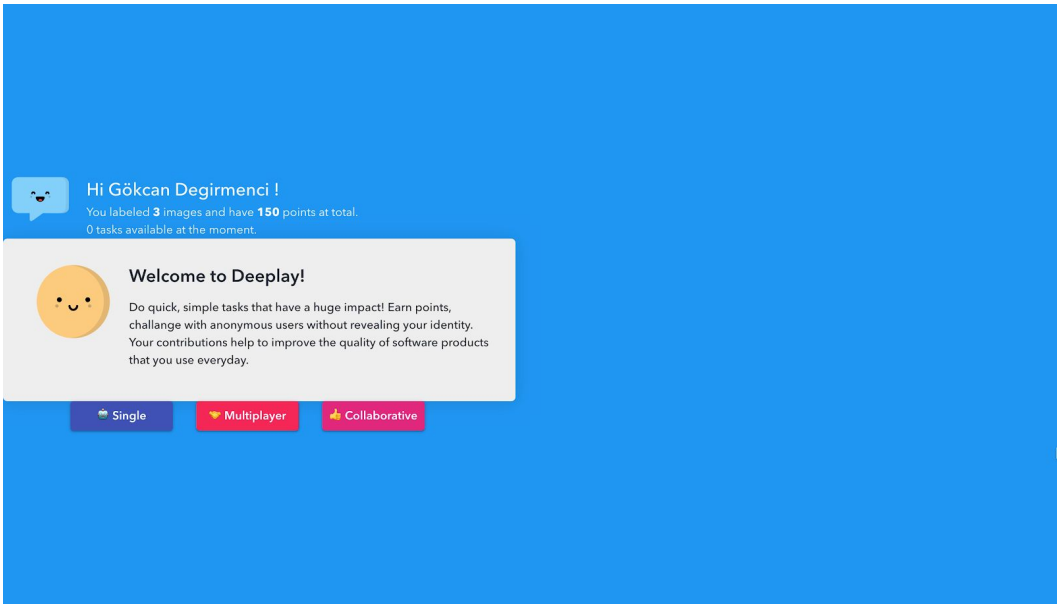
7.7. Setup Instructions



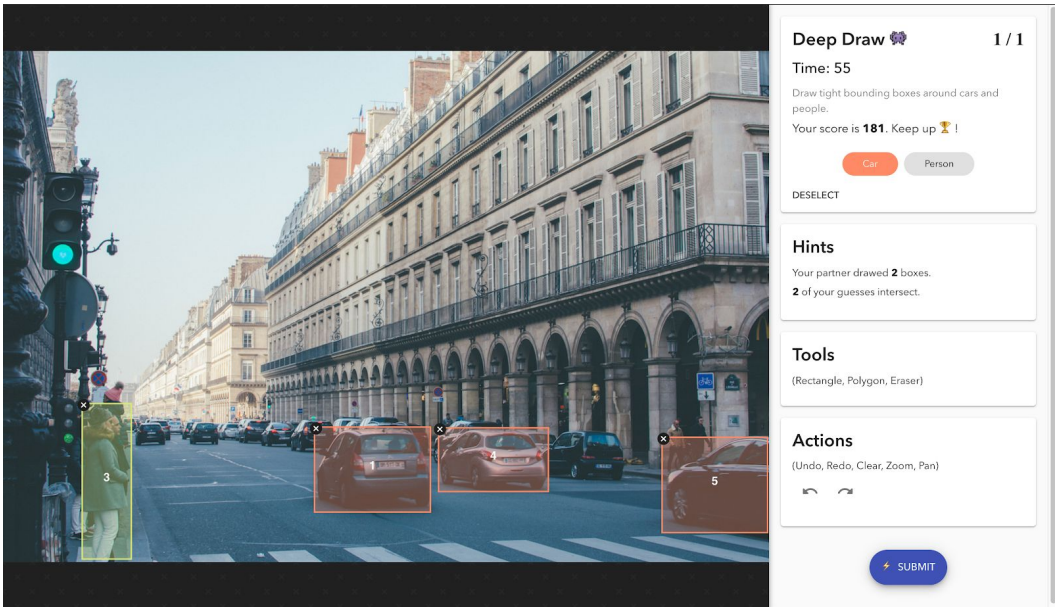
7.8. Show Projects



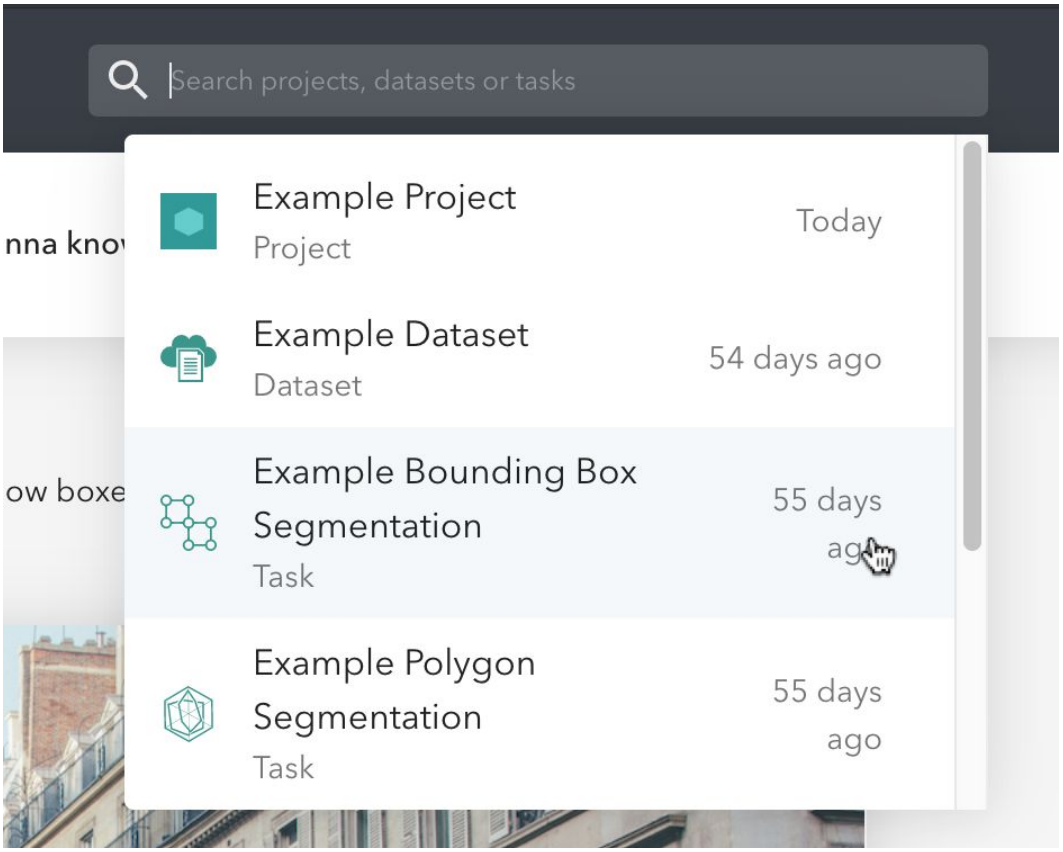
7.9. Deep Draw - Welcome



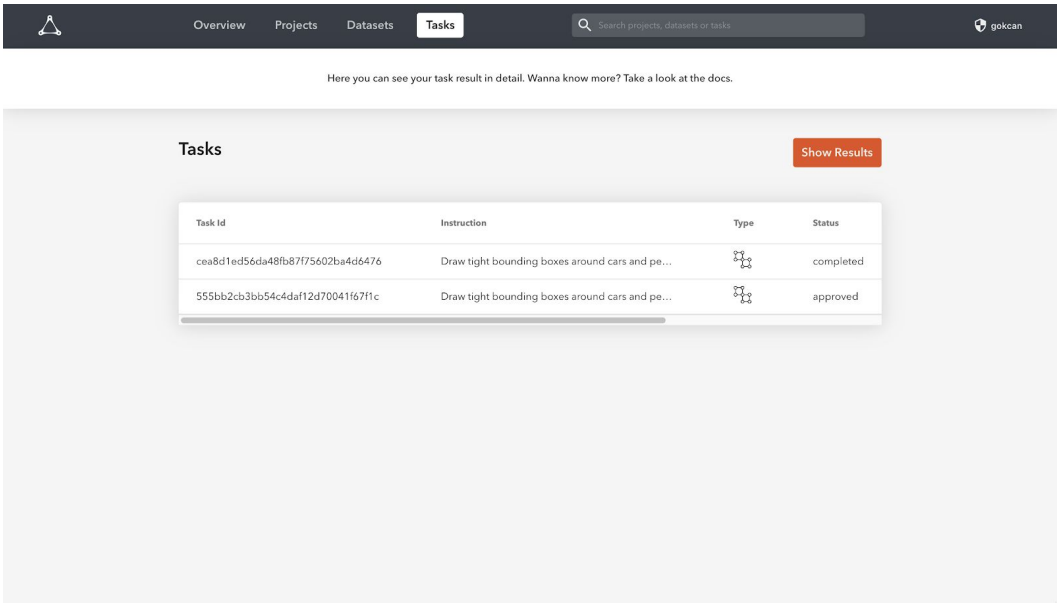
7.10. Deep Draw - Main



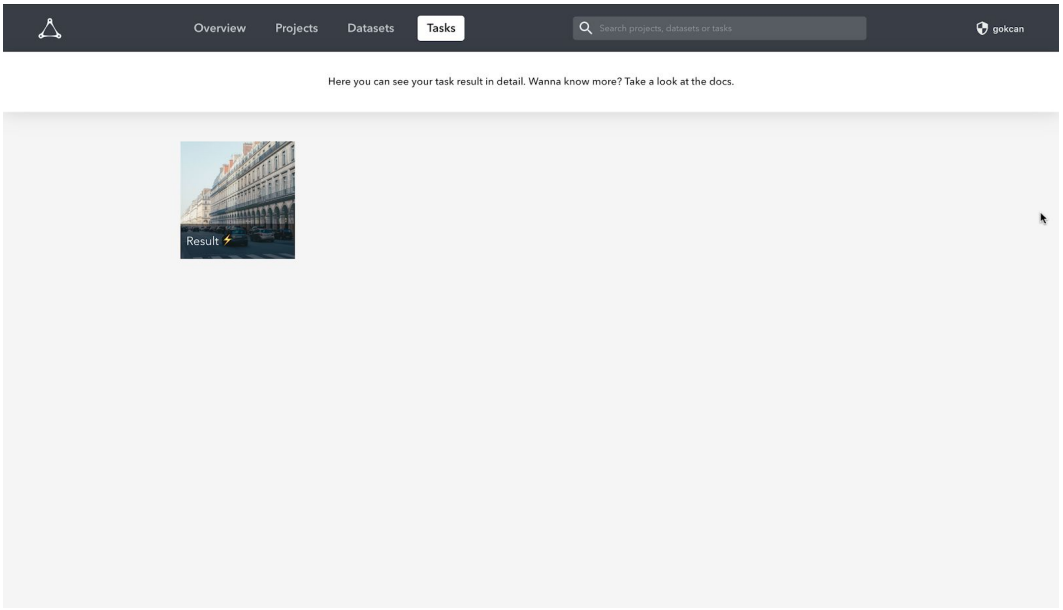
7.11. Search



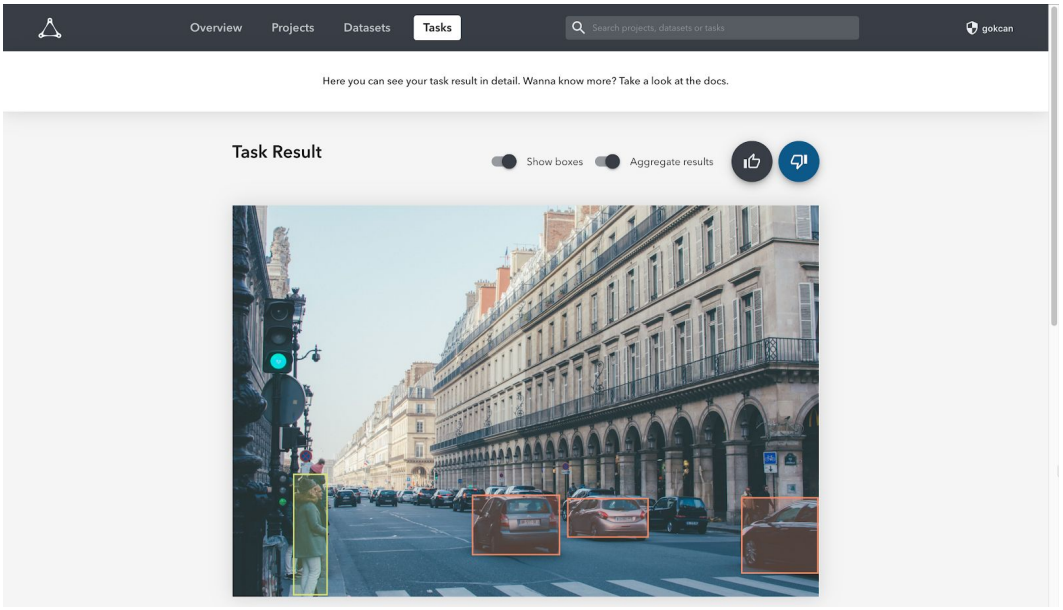
7.12. Show Tasks



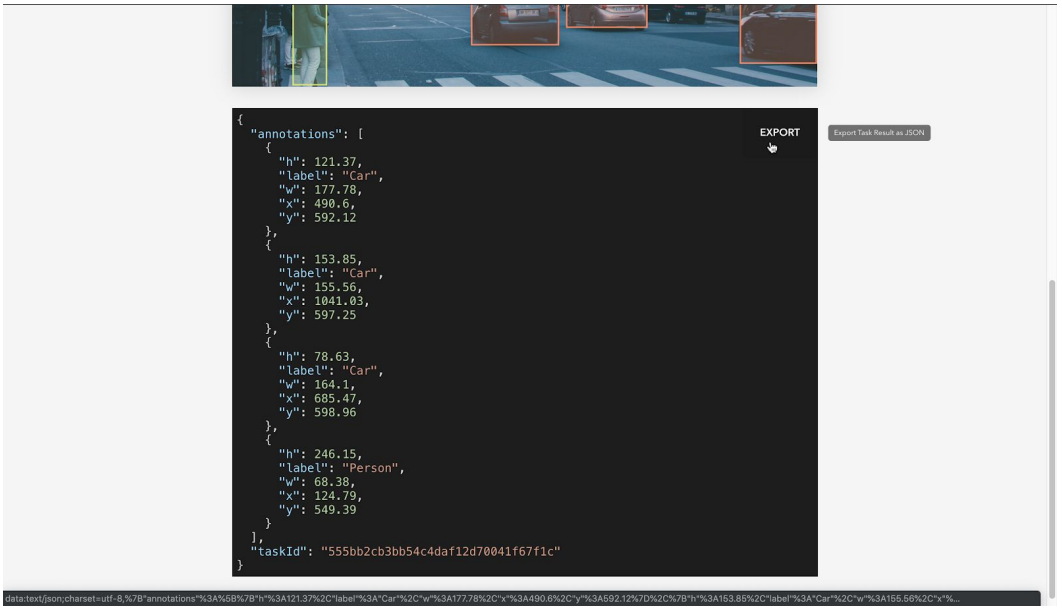
7.13. Result Grid List



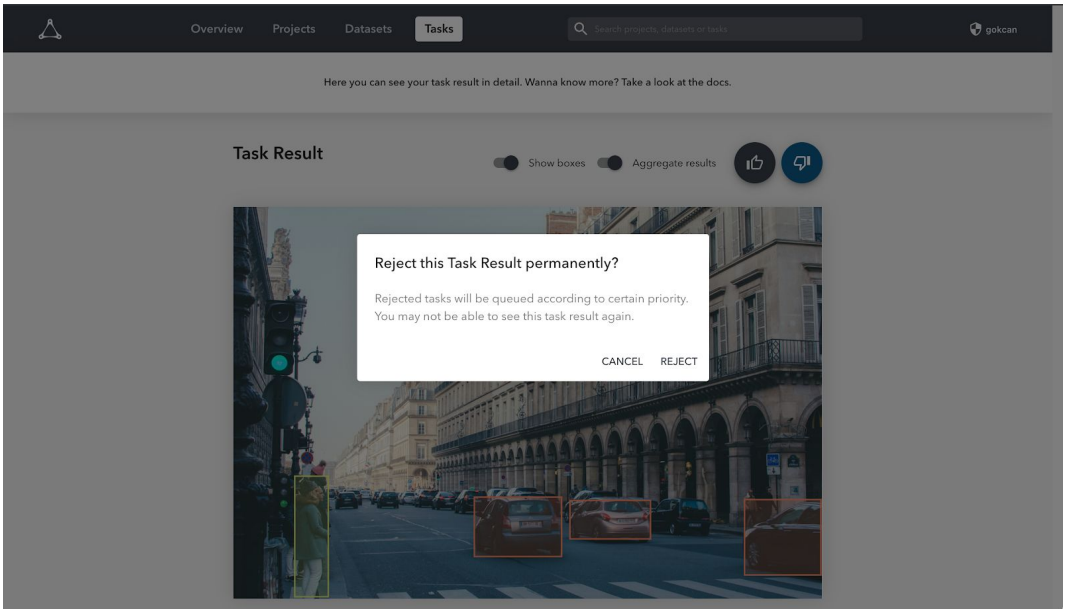
7.14. Result View



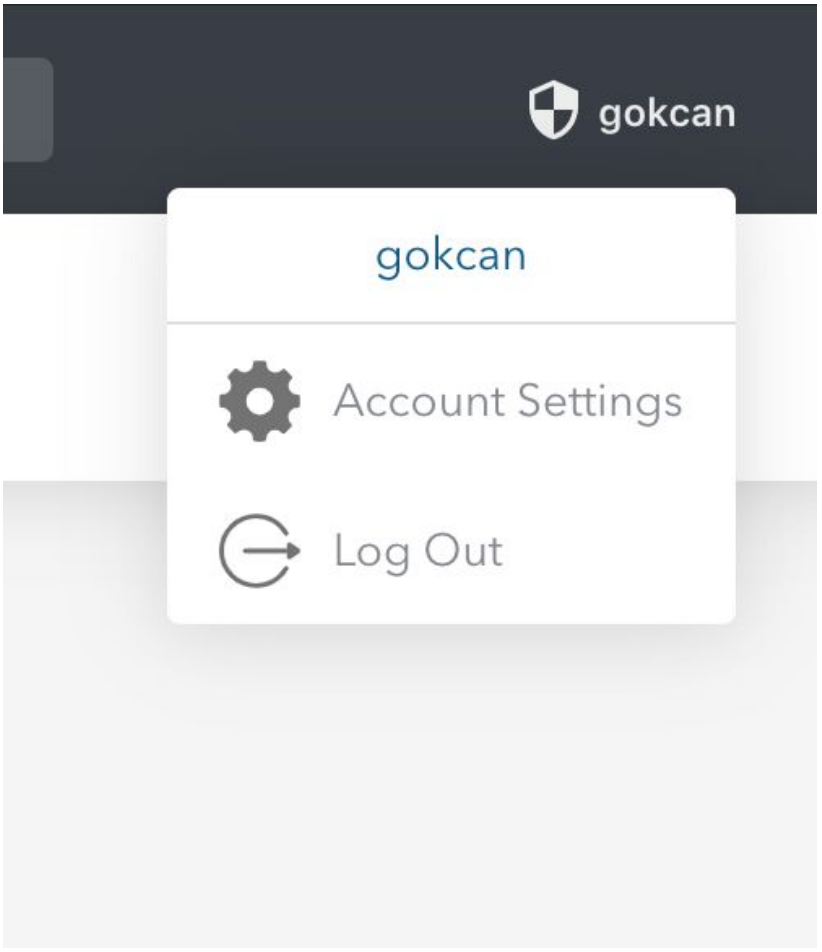
7.15. Export JSON



7.16. Approve or Reject



7.17. Log Out



8. References

1. **“Communication in a microservice architecture”,** 2018**.** https://docs.microsoft.com/en-us/dotnet/standard/microservices-architecture/architect-microservice-container-applications/communication-in-microservice-architecture
2. “**Microservice Communication Patterns**”, 2018. https://reflectoring.io/microservice-communication-patterns/
3. “**Microservice Architecture**”. https://microservices.io
4. **GoDoc/websocket Documentation**. https://godoc.org/golang.org/x/net/websocket
5. Jocelyn C. Adams, Kristen C. Allen, Tim Miller, Nathan D. Kalka, Anil K. Jain . “**Grouper: Optimising Crowdsourced Face Annotations”**, **2016**.
6. Rajan Vaish, Sascha T. Ishikawa, Sheng Lundquist, Reid Porter, James Davis University of California at Santa Cruz, Los Alamos National Laboratory. “***Human Computation for Object Detection”, 2013.***
7. Navaneeth Bodla and Bharat Singh and Rama Chellappa and Larry S. Davis. “***Soft-NMS -- Improving Object Detection With One Line of Code”, 2017.***
8. DBSCAN scikit-learn Documentation. *https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html*