# bEquality - Automated Tracker for Gender Equality with Blockchain
# Creation of an Automated and Reliable Gender Equality Index

Noah Berner (noberner@student.ethz.ch), Lena Csomor (lcsomor@student.ethz.ch),
Eric Léger (legere@student.ethz.ch), Matthias Matti (mmatti@student.ethz.ch),
Roy Schubiger (roysc@student.ethz.ch), Lino Telschow (tlino@student.ethz.ch),
Fredin Thazhathukunnel (fredint@student.ethz.ch), Qianchen Yu (qyu@student.ethz.ch)

ETH Zürich, 30.04.2018

All contributors contributed equally to this report

The software code which is part of this report is open source and available at
`https://github.com/ETHBiots2018/bEquality`

This project report was written as part of the spring 2018 course 'Blockchain And the Internet of Things (851-0591-01L)' run by M. Dapp, S. Klauser, and D. Helbing.

## Abstract

Today, most companies' market value is driven by intangible values, such as reputation or branch equity. Thus, thousands of companies now provide detailed information beyond their financial performance. Regulations are mandating increased management disclosure and analysis on sustainability, and investors are analyzing the comprehensive risks and opportunities of issuers in public and private markets. One primary aspect of this extra-financial information is on gender equality.

More and more investors take this data into consideration when investing as research suggests that good gender equality practice can serve as an indicator for good corporate governance and decision making and these companies might outperform in the future. To get a structured, concise and comparable overview of this data, investors often rely on established gender-equality certifications. Unfortunately, this process is costly, time-intensive and requires multiple revisions to produce reliable data. [1]

Our project aims to simplify and automate the process of obtaining such a gender-equality certification based on an existing Gender-Equality-Framework, and to make its results publicly accessible for everybody.

Our approach focuses on the data capture, data storage, data validation, and the visualization of the process.

The data is obtained by the means of a website and an app whereas the storage solution relies on an E-voting system, based on the latest blockchain and cryptography technology. The solution provides the code of the communication interfaces for the website and the app, as well as the necessary code to handle the data for the E-Voting system over blockchain. Our process is blockchain-agnostic, however, our Proof-of-Concept is implemented on the Ethereum blockchain.

[1] This paragraph is adapted from the challenge description handed to us by the UBS

Furthermore, a spider diagram is used to present the results in a straightforward manner.

In conclusion, the platform ensures a trustful, transparent, and cheap way to create a gender-equality-index, which is useful for people to make investment decisions and for companies to further improve their status with respect to gender equality.

### Introduction

The current situation of the economy does not seem fully inclusive for every member of the society. We lay our focus on an equal status for men and women, but we are aware that inequalities are not restricted to gender.
Our Project bEquality aims to punctually recognize inequality by providing a transparent, reliable and efficient gender-equality index for companies. We think that applying and evaluating a gender-equality index should not be expensive and exclusive to companies that can afford it.
We aim to improve today's state of society with respect to gender-equality in the economy.

This document summarizes our approach to solve this problem. Our solution has come up on a hackathon organized by the blockchain and Internet of Things School (BIOTS) 2018.

We will first present you our whole process systematically and then later dive further into the implementation.
The reader should have basic programming understanding or basic logical thinking to understand our approach and implementation, which we try to explain as easy as possible.

# 1 Project

## 1.1 Overview

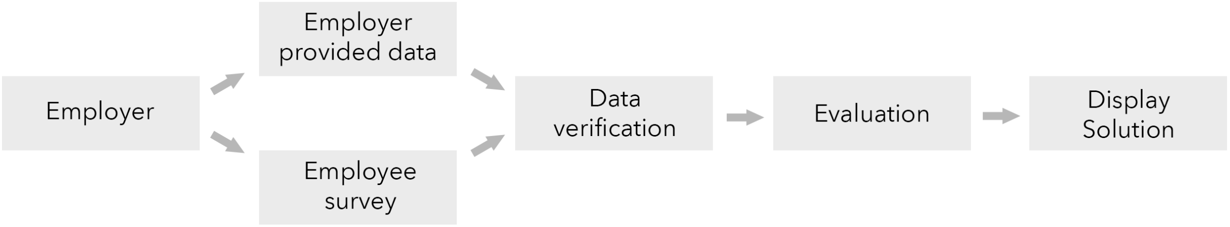


Figure 1: Project flow graph

The figure 1 visualizes the high-level structure of our project.
Existing gender-equality indices differ from our project at the points of *data survey (employee survey)* and *data verification* in the high-level structure and in the final evaluation.
We will now explain the high-level structure of our project.

#### Employer Survey

There are a few necessary steps for an employer to take in order to request a certification. First, the employer has to provide sufficient data required by the given rating framework via a web front-end. Additionally, the company needs to provide an unbiased means to reach all their employees, which is employee E-mail addresses in our Proof-of-Concept. Those data mentioned above are the employer provided data.

### Employee Survey

Upon the reception of the employer provided data, we provide an employee survey to all employees of the company. This survey is a direct, secured, and private channel for data collection. It is used to validate the plausibility of the provided data from the employer and to obtain further important private data from employees that cannot be known by the employer. Each employee has a unique survey generated by the smart contract that is only accessible by the particular employee.

It is an easy task for the employee to participate in the survey. The only effort needed from the employee is to set up his/her account. The employee receives a unique invitation link from our smart contract powered program that leads him/her to our website and/or to the app, where he/she can set up his/her account. While setting up the account, a public and a private key are generated automatically in the background that link to the employee's account.
After the account set-up, the company sends all the anonymous public-keys generated by the employee account set-up to our framework. This then initiates the survey set-up.
When all background work is done, the employee gets a message indicating that the survey is ready. He/she has then just to log into the app, fill out the survey and then submit the data.

Everything technical is handled automatically in the background, such that the user just observes a handy interface, and all data is stored on either the blockchain or the InterPlanetary File System (IPFS).

### Data Validation

When all data is gathered from the employees, we split the data into different classes. There is the objective data from the employees that can also be provided by the employer, which is segmented therefore in the *intersecting-data category*. The other category of data is subjective inputs from the employee, such as positive or negative personal experiences.

All data from the *intersecting-data category* is analyzed by comparing the provided view of the employer and the view of the employee.
This comparison then provides a validity factor for the employer provided data.
This process of data validation ensures that the data is not corrupted by either humans or bots.

### Result Visualization and Evaluation

We would adapt our evaluation to already existing gender-equality indices. But to further improve these evaluations, we would like to evaluate also different indicators for gender-equality and other measures that are important.
The data evaluation then finally results in not only a single index number, but in a multidimensional spider-diagram, which allows the investor or the company to look closer into existing problems or advantages over the requirement.
The figure 2 provides a customized evaluation system with detailed feedback.
This detailed feedback is useful for prospective investors of this particular company for their decision-making. It also serves as an indicator for the company, such that the company knows what to improve in its working environment.

# 2 Technical Implementation

## 2.1 Data Capture and Storage

### Company Provided Data

As already mentioned earlier, the *data capture* process is split into two parts.
The first part consists of a survey that is sent to and filled out by the human resources department of the company, i.e. the *company provided data*.
The questions on this survey will be based on already existing indices, such as the ones from Bloomberg or Equileap. They fill out the survey directly on the website which stores the data on the blockchain via IPFS. Because the data is stored directly on the blockchain, it is verifiable for everyone and cannot be tempered with in any way at a later point in time.
The company also sends us the email addresses of all their employees, which we will store on a private and secure database. The email addresses can obviously not be stored on the blockchain for privacy reasons. This is a trade-off in transparency that was inevitable.

### Employee Survey

The second part of *data capture*, the *employee survey*, is a special point of interest. Because it is important to guarantee anonymity to the employee and at the same time ensure validity and transparency of the data that is provided.

The first step in the *employee survey* takes part after we received all the email addresses of the employees. We send them an email in which there is a link to our survey app.
In the app, they enter a password for a new Ethereum blockchain account which is generated automatically. The private key of the account is kept by the employee with possibility to be stored only on the employee's device where the private key is secured by the password they entered. The app sends the public address of the blockchain to our server, where it is used to load a small amount of Ether onto it, so that the employee can later make one transaction on the Ethereum blockchain for the submission of survey.
Once all (or a big enough percentage of) the employees have followed our link and created an account, our server will create a new survey contract on the blockchain.

This happens via the following smart contract (contract `SurveyFactory`), where an array of public addresses and a hash to the location of the addresses in IPFS is required to create a new smart contract of the kind `Survey`.

```
// https://github.com/ETHBiots2018/bEquality/blob/master/Survey.sol
/* SurveyFactory serves as a hub
    (deployed on the blockchain upon the launching of bEquality)
    Company can create their own survey by providing a list of permitted user address. */

contract SurveyFactory {
  mapping(uint => address) public SurveyContracts;
  ...
  function createNewSurvey(uint companyID, address[] addressessOfEmployees, string
      _hashToaddressessOfEmployees) public returns(address newContract)
  {
    require(SurveyContracts[companyID] == 0x0);
    Survey c = new Survey(addressessOfEmployees, _hashToaddressessOfEmployees);
    SurveyContracts[companyID] = c;
    return c;
  }
  ...
}
```

SurveyFactory can be considered as a container in which all the surveys of the different companies are stored and can be accessed via the unique id of the company.

After the Survey contract is created, the employees are notified that they now can fill out the survey. They log into their account on the app and answer the questions provided by us. This is done via a user-friendly interface [2].

The survey for the employees presents them with gender specific questions and also with questions applicable for both men and women. These questions guarantee further insight into the micro-climate of the employees that cannot be obtained by just consulting the company management.

When they submit the data, it gets stored on IPFS and the hash is again stored on the blockchain.

This is done via the smart contract Survey seen below. It takes a hash to the location of the survey results on IPFS and stores it on the blockchain.

```
// https://github.com/ETHBiots2018/bEquality/blob/master/Survey.sol
/*
   Survey is the child contract created by the SurveyFactory where only the permitted user can modify.
*/

contract Survey {
    mapping (address => string) public hashes;
    mapping (address => bool) public isAllowedToSumbitSurvey;
    string hashToaddressessOfEmployees;

    function Survey(address[] addressessOfEmployees, string _hashToaddressessOfEmployees) public {
        hashToaddressessOfEmployees = _hashToaddressessOfEmployees;
        for (uint256 index = 0; index < addressessOfEmployees.length; index++) {
            isAllowedToSumbitSurvey[addressessOfEmployees[index]] = true;
        }
    }

    function submitResults(string myHash) public {
        require(bytes(hashes[msg.sender]).length == 0);
        require(isAllowedToSumbitSurvey[msg.sender]);
        hashes[msg.sender] = myHash;
```

[2] The mock-up of the user-friendly interface can be found in the appendix

```
    }
}
```

The contract ensures that only the employees addresses which were submitted on contract creation can submit results. It also ensures that an employee can only submit results once and that he/she can not alter any other results. Because the employee knows his/her own public blockchain address (it is displayed on the app), he/she can easily verify that the data on the blockchain did not get altered by anybody.

The employee is also the only person who knows that this public key is associated to him/her. Thus the sensitive data stored on the blockchain still insures the privacy of the employee and the employer cannot prosecute the employee for telling his opinion of the company.

In our implementation, most of the data that gets collected is only stored indirectly on the blockchain via IPFS hashes. This was necessary to reduce the costs of the individual transactions and for overall scalability.

Ethereum transaction costs depend, among other things, on the size of the data stored on the blockchain[3]. If we only store hashes instead of survey data, the size of the transaction remains small, no matter the number of questions in the survey.

There are still challenges ahead of us for improving our implementation. For example, the possibility of storing the public Ethereum addresses on the blockchain instead of a private database while they are still being collected for further transparency and automation of the process.

### App Implementation

To make the survey as easy as possible for the employees, we decided to use an app. We built a demo app for Android smart-phones with the Android Studio IDE. [4]

The app is only a concept and should serve as a visual guide of how the actual working app would look like. Our demo app consists of a sample survey and dummy buttons and cannot actually submit the survey to the blockchain. Our Application cannot actually submit the survey to the blockchain, because we were not aware of the existence of a Android-Ethereum interface[5], nor of a Java-Ethereum interface[6]. Nevertheless, our app serves as a great visual reference.

However, as a compensation to the lack of connectivity to the blockchain, we implemented a website prototype that can interact with Web3[7] and actually submit the survey to the blockchain[8].

The fully implemented app could then work as follows:

1. The employee sees a login screen and is asked to fill in his/her password for the Ethereum blockchain account.
2. Once logged in, the app shows the user the questions to answer. The interface is straightforward and self-explanatory.
3. After the user submits the survey, the app sends the survey results to the blockchain/IPFS and reports that the submission was successfull. It also shows the employee his/her blockchain address for verification purposes.

[3] Rising costs with increasing size of data to store on blockchain is true for Ethereum, according to Ethereum Introduction lectures held in the BIOTS week

[4] The visual preview of this APP can be found in the appendix

[5] See `https://ethereum-android.com`

[6] See `https://web3j.io`

[7] `https://web3js.readthedocs.io/en/1.0/`

[8] Look at the Appendix for a visualization of the Web interface. The implementation of our website can be found in our Git-repository

The big advantage of such an app is the self explanatory user interface.

Some of the disadvantages of using an app to collect data are:

- Even for one survey, the app needs to be installed on the phones of the employees.
- Survey questions are hard-coded into the current app.

The second problem can be solved by downloading the questions from a server after the user is logged in. With this approach the app serves as a framework for all kinds of surveys.

It is unforeseeable if the app is the right choice to use for the gender equality survey and a good way to find out if it is, would be to test multiple different ways of collecting data and then analyzing their effectiveness.

## 2.2 Data Verification

As already mentioned in the overview part, we can distinct our data into three different categories:

- Solely employer data
- Intersecting data
- Solely employee data

We cannot prove the validity of the solely employer data, neither the validity of the solely employee data, because we do not have access to a reference data set to compare it.
Whereas we can look at the plausibility of the intersecting data provided by the employee, by comparing it to the reference data set provided by the employee.

The objective of our data verification process is to provide a number that represents the plausibility of our evaluation. One approach to obtain such a number is the following process described below.

When looking at the intersecting data category and then look at two specific subsets. One subset, say $A$, is provided by the employer. The other subset, say $B$, is provided by the employees.
In our survey we have different questions or other measures that can represent the answers. One now could categorize these answers as a number between 1 and 10 (this could also be done more fine-grained or more coarse-grained). These numbers should be set in advance, such that they represent a measure of quality of the answers with respect to the order of the given numbers.
With this insight, one now can represent the data in $A$ and the data in $B$ respectively, by a vector with $n$ dimensions, where $n$ is the number of questions or important measures (this obviously depends on the survey).
Let's say the corresponding vector to the data set $A$ is called $a$, and $b$ to the set $B$ respectively.
As a validity measure of the data in $A$, one now could take the following number:

$$\|a - b\|^2$$

It represents the norm of the difference of the two vectors squared. This performance measure would punish differences in the dimensions more than similarity in the respective dimension.

As an improvement, one could also represent this result as a number between 0 and 1, by eventually normalizing the vectors $a$ and $b$ in advance, and then representing the result as a percentage value for validity. This process could be implemented in a more advanced state of the project.

Please note that this is just one approach of data verification measure, and that there are other measures that follow the objective, which may be better than this approach. Since our main objective is not finding the optimal data verification measure, we adopt the aforesaid measure in our project.

## 2.3 Display Solution/Evaluation

Current gender equality barometers, such as the Bloomberg Gender Equality Index or the Equileap Ranking, display the result of each company as a single number, score or grade. With this information, people, especially potential investors, can see which company performs cares about gender equality or which company is "more" gender-equal than another one. But this one-dimensional approach does not offer a lot more insight.

People whom such a ranking interests probably care not only about the fact that a company makes effort in gender equality, but also about **how** they do. Moreover, to see how a single number (e.g. the rating result) - which includes a lot of different factors - is composed, it requires quite a bit of investigation. To address these points, our solution to result visualization focuses on a multidimensional approach by means of spider diagrams.

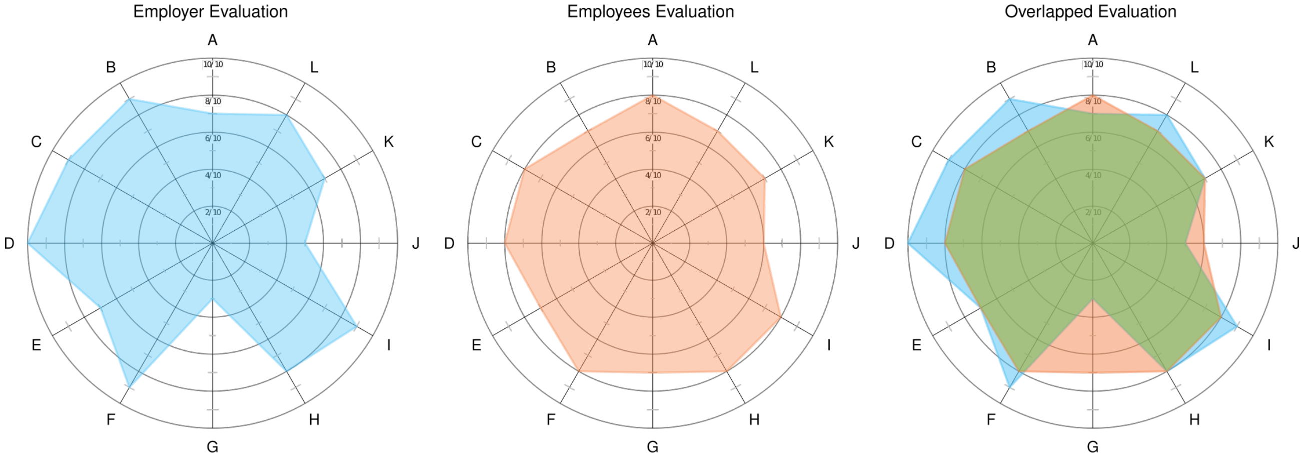


Figure 2: Spider-diagram evaluation example

With the spider diagram it is possible to show multiple factors at the same time. On each branch of the diagram, a different factor can be displayed (in the figure, the labels 'A - L' represent each one factor).

An important point in the survey evaluation is to identify the relevant factors which should be included into the spider diagram. These factors could rely on the existing frameworks (e.g. from Bloomberg or Equileap) or one could come up with new factors based on the surveys. An obvious factor is for sure the difference of salaries between men and women counterparts.

Due to the settings of the surveys, there will be data from the employer survey which can not be captured by the employee survey and vice versa. For example, the employer can deliver factual data, whereas the employee can express his personal impressions and experiences of the daily work life. This data can be bundled in the *non-intersecting-data category*. Nevertheless, concerns which are addressed in both surveys can be compared. This data can be bundled in the *intersecting-data category*.

As shown schematically in the figure above, these two categories can be in an straightforward manner with spider diagrams. On one hand, the *non-intersecting-data category*, there could be a spider diagram for the employer survey as well as the employee survey. On the other hand, a spider diagram with the *intersecting-data category* could present the factors, where data from both sides is displayed.

The overlapping in the spider diagram of the *intersecting-data category* can serve as an indicator of data-validity. When a company provides wrong data, then this would result in a smaller overlap. One has to take into account, that people could be forced by the company to give certain answers to gain a competitive advantage. [9]

Spider diagrams are widely known, and the display solution is easy to read. Due to the fact that the diagram displays more than a number as the other ratings, it is a nice tool to gain additional, comprehensible insight into the efforts a company does in the field of gender equality.

Because the data from the evaluations is open accessible on IPFS, the diagrams can easily be generated and fetched from these places and be displayed via a website or an app.

## 2.4 Challenges to Solve and Further Points Worth to Consider

There are still some challenges to solve for this project.

One major challenge is to provide a secure way of implementing a system equivalent to E-voting, i.e. to find a system where a user applies to a survey or evaluation and the data provided by him cannot be traced back to this user, even when the user can verify that the data provided by him is used and not some modified version of his data.
One way of solving this problem, as implemented in this gender-equality index use-case, is to make that cheating for companies too expensive to be profitable. Companies therefore act cooperatively and provides true data.
Once we are certain that companies do not cheat, we have to look at the behavior of those who can have a great impact on the evaluation: the employees.

We have to assume that the participation rate of the employee survey is rather low, surely not more than 50 percent. This is because people who are moderately happy often do not feel like they have to praise their employer, but they also do not complain[10]. Therefore, most of the people who will respond are either unhappy or very happy, the votes of the moderate people tend to get lost. To get a considerable participation rate, we have to send emails to every employee.
We have to be certain that our questions are unambiguous to get the information we really want. Furthermore, we have to adapt the questions for different hierarchy levels. A manager may have other concerns than a worker. To retain the participants' interest, we have to make sure that the questions are concise and that there are not too many of them.

Also, it is our concern that there is no (easy) way to identify the voter by replicating his answer. A rather simple way to avoid part of it is the strategy of randomized response. This means that if two people are at the exact same point in the survey, let's say they both have two choices to tick, one may be asked a trivial question, like which sort of ice cream he likes. The other one will be asked a serious question which is important for out data collection. This way, even if someone knows at which point which choice has been made, (for example he sees the employee ticking on his smart phone, but cannot read the question from a distance) he cannot know what it meant.
We know that anonymizing data opens the door for abusers who try to manipulate the survey[10]. The more anonymous

[9] An approach to solve this problem can be found in the data verification section

[10] For informations regarding the employees reaction to a survey, we have conducted Prof. Dr. Renate Schubert in a short interview

the participation is, the easier it is to abuse it. Sadly we have not yet found a way to fix this completely - we can only agree on a certain balance between control and anonymity.

Another problem we had was to determine which questions we should take for the survey in the application. The application on the mobile phone should actually be a small survey for the employee to fill in with around 5 minutes. However, there are a lot of questions we have to ask in order to get useful data which can be compared with the employer provided data.

A similar issue is to find fitting indicators that are the best to display in the Spider-Diagram, which provides enough useful information worth to consider, and at the same time does not abstract away some useful information. This indicators should also allow to compare different companies that are on our index, even if they do not have an equal number of employees or act in different industries.

Another challenge is to handle and predict the costs to create a contract and store some data on the blockchain. These costs should, when it is possible, be avoided by some more efficient and cheaper approaches with the same advantages as our approach has. One possible solution is to implement such smart contracts on other blockchain platforms where no cost occurs during transactions.

One problem we encountered and couldn't solve happens in the log-in process for the employees. Our current implementation depends on, that each employee has to create his own account before taking the survey. The obvious disadvantage of this solution is, that each employee needs to remember his personal password. A better approach would be an automated log-in process, such that the blockchain account is set up in the background. With this approach the password may be created randomly, without the users knowing about.

Also because of the lack of time we couldn't finish our implementation completely. The Mobile Applications needs to collect the data and put the collected data onto the blockchain.

# 3 Conclusion

### Strengths and Weaknesses

Our solution provides a reliable, transparent and automatic way of creating a gender-equality index. Compared to other already existing indexes we offer an index where the reliability and correctness can be verified by each and every person. This is done by implementing our strategy on the blockchain.

Our approach allows everybody to classify companies according to the latest law regulatory for gender-equality in companies.

Our solution is multidimensional, where multi-dimension stands for two different things. First of all, our index contains different categories where the result is weighted and then demonstrated as a graph 2. The other multi-dimension of our project is that we collect data from diverse sources. We collect it from the managers and we collect the data from a casual worker as well.

A great advantage over existing gender-equality indices is that our solution is public and can be implemented with low costs. Existing indices nowadays require a lot of work e.g. to collect the data, to evaluate the data, to present the data in a meaningful way and so on. In our solution we are trying to do as much as possible in an automatic approach. A downside of this automation is that we need to send an invitation link to every employee to ensure enough participation, which means we need their Email addresses, and at the same time create a blockchain account for every participant before he/she can take the survey.

### Open Challenges

There are several ways to expand our project. Examples would be to achieve a fully working automatic way of indexing gender-equality. This is our initial idea but we haven't completed the implementation due to time constraints.
The application we use to do the survey has just an interface. This means that we should collect the data on the application and try to put the collected data onto the blockchain for further calculation processes.
Moreover, there are still different challenges to improve our E-voting system, such that we can guarantee data validity as well as anonymity for participants.

### Disruption Potential

Our approach is revolutionary in the sense that our process is completely transparent and every step in our process can be observed and verified by the employees, the company as well as by the investors.
We guarantee data availability for everyone, such that they can examine their own results from the given and anonymous data on the blockchain.

Finally, our system allows a reliable and transparent way to improve gender-equality, which helps to decrease gender-related discrimination in the long term.

## 4 Sources and Literature

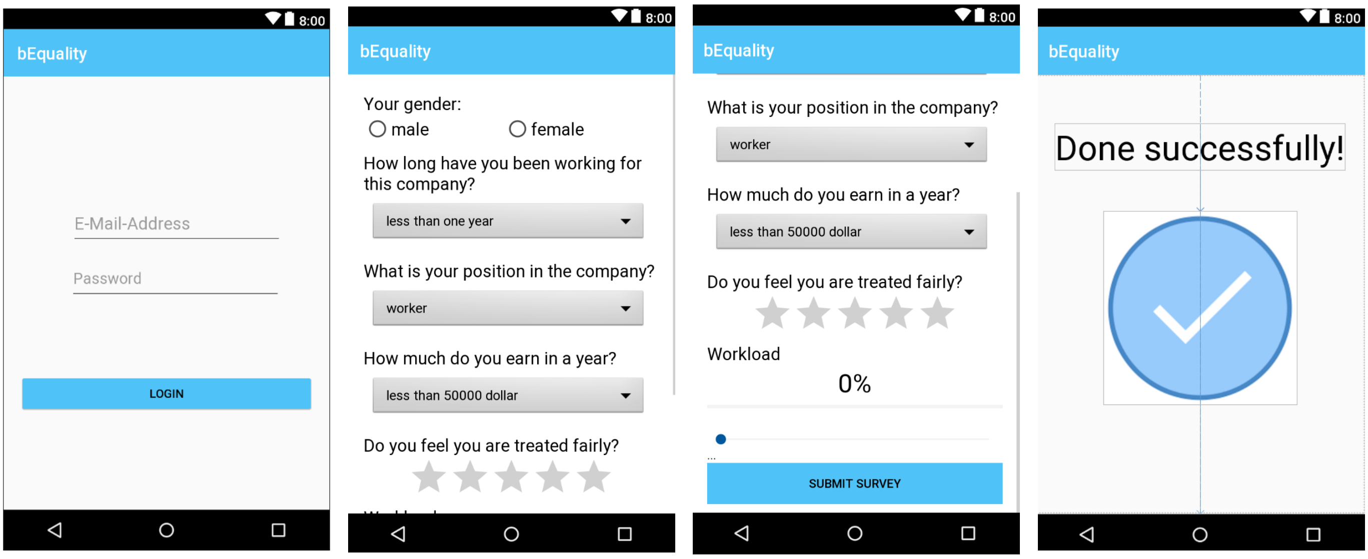
Sources used in order to develop the app we used:

- `https://stackoverflow.com`
- `https://github.com`
- `https://developer.android.com`
- `https://web3js.readthedocs.io/en/1.0/`

# 5 Appendix

## 5.1 Survey-App Prototype

Below is an App-prototype. From left to right is the log-in screen, two survey screen-shots and the screen, when the employee has successfully submitted his/her answers to the blockchain or private database.



8:00

bEquality

E-Mail-Address

Password

LOGIN

8:00

bEquality

Your gender:
○ male ○ female

How long have you been working for this company?

less than one year

What is your position in the company?

worker

How much do you earn in a year?

less than 50000 dollar

Do you feel you are treated fairly?

★★★★★

8:00

bEquality

What is your position in the company?

worker

How much do you earn in a year?

less than 50000 dollar

Do you feel you are treated fairly?

★★★★★

Workload

0%

SUBMIT SURVEY

8:00

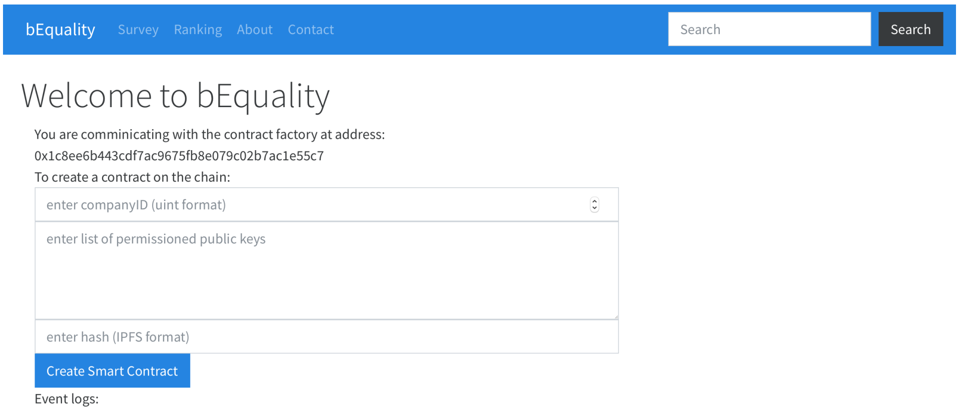
bEquality

Done successfully!

## 5.2 Web Interface

As an alternative to the Survey-App, we also implemented a website prototype, that can communicate with the blockchain and store the survey data on the blockchain.

Here is a short overview of this website-prototype, for further insight, one can download the source code of this website.



bEquality Survey Ranking About Contact Search Search

Welcome to bEquality

You are comminicating with the contract factory at address:
0x1c8ee6b443cdf7ac9675fb8e079c02b7ac1e55c7
To create a contract on the chain:

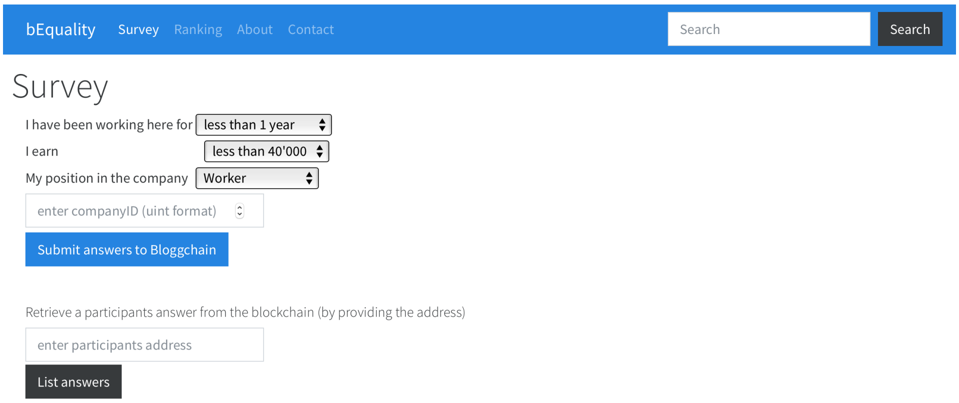
enter companyID (uint format)

enter list of permissioned public keys

enter hash (IPFS format)

Create Smart Contract

Event logs:



bEquality Survey Ranking About Contact Search Search

Survey

I have been working here for less than 1 year

I earn less than 40'000

My position in the company Worker

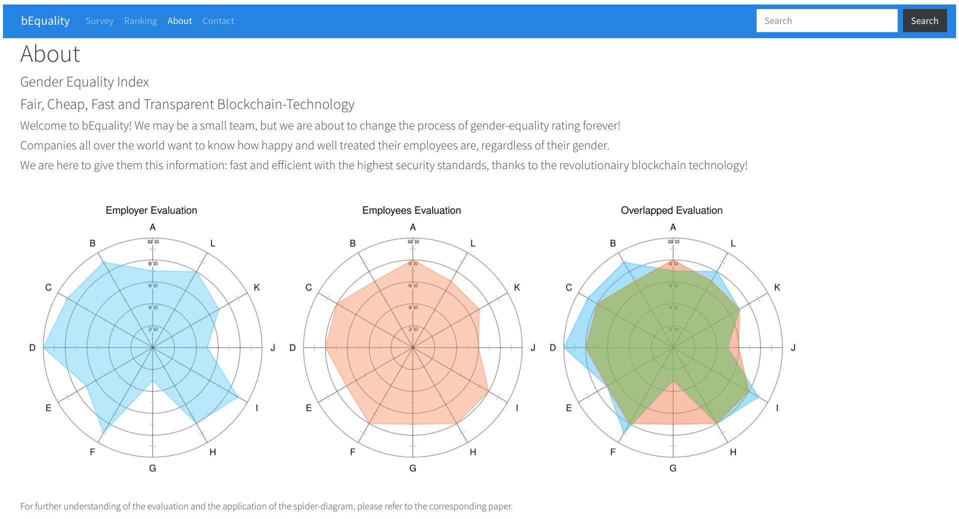
enter companyID (uint format)

Submit answers to Bloggchain

Retrieve a participants answer from the blockchain (by providing the address)

enter participants address

List answers



About

Gender Equality Index

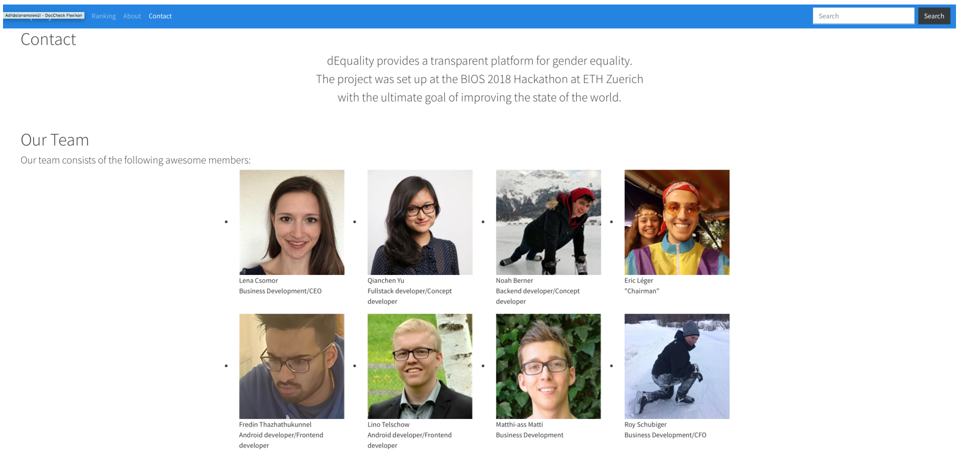
Fair, Cheap, Fast and Transparent Blockchain-Technology

Welcome to bEquality! We may be a small team, but we are about to change the process of gender-equality rating forever!

Companies all over the world want to know how happy and well treated their employees are, regardless of their gender.

We are here to give them this information: fast and efficient with the highest security standards, thanks to the revolutionairy blockchain technology!

For further understanding of the evaluation and the application of the spider-diagram, please refer to the corresponding paper.



Contact

dEquality provides a transparent platform for gender equality.
The project was set up at the BIOS 2018 Hackathon at ETH Zuerich
with the ultimate goal of improving the state of the world.

Our Team

Our team consists of the following awesome members:

- Lena Csomor
  Business Development/CEO
- Qianchen Yu
  Fullstack developer/Concept developer
- Noah Berner
  Backend developer/Concept developer
- Eric Léger
  "Chairman"
- Fredin Thazhathukunnel
  Android developer/Frontend developer
- Lino Telschow
  Android developer/Frontend developer
- Matthi-ass Matti
  Business Development
- Roy Schubiger
  Business Development/CFO

## 5.3 E-Voting protocol

Below is a schematic illustration of our E-Voting protocol.

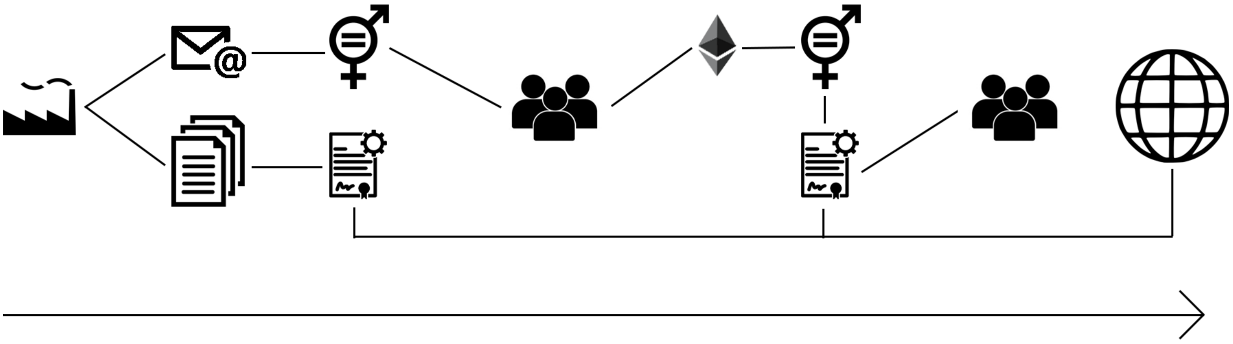


Figure 3: technical flow representation of the data capture and storage process

## 5.4 `Survey.sol` Code

The SurveyFactory serves as a hub (deployed on the blockchain upon the launch of bEquality).

The company can create their own survey by providing a list of permitted user address.

```
pragma solidity ^0.4.17;

contract SurveyFactory {

  //address owner;
  mapping(uint => address) public SurveyContracts;

  // function SurveyFactory(address adr) public {
  // owner = adr;
  //}

  function createNewSurvey(uint companyID, address[] addressessOfEmployees, string _hashToaddressessOfEmployees) public returns(address newContract) {
   // require(msg.sender == owner);
   require(SurveyContracts[companyID] == 0x0);
   Survey c = new Survey(addressessOfEmployees, _hashToaddressessOfEmployees);
   SurveyContracts[companyID] = c;
   return c;
  }

  function getContractAddress(uint companyID) public constant returns (address) {
   return SurveyContracts[companyID];
  }
}
```

The Survey contract is the child contract created by the SurveyFactory where only the permitted user can modify.

```
pragma solidity ^0.4.17;

contract Survey {
  mapping (address => string) public hashes;
  mapping (address => bool) public isAllowedToSumbitSurvey;
  string hashToaddressessOfEmployees;

  function Survey(address[] addressessOfEmployees, string _hashToaddressessOfEmployees) public {
    hashToaddressessOfEmployees = _hashToaddressessOfEmployees;
    for (uint256 index = 0; index < addressessOfEmployees.length; index++) {
      isAllowedToSumbitSurvey[addressessOfEmployees[index]] = true;
    }
  }

  function submitResults(string myHash) public {
    require(bytes(hashes[msg.sender]).length == 0);
    require(isAllowedToSumbitSurvey[msg.sender]);
    hashes[msg.sender] = myHash;
  }
}
```