Invention Disclosure System Using Blockchain

CSCI E-599 – Harvard Extension School

Customer: Dave Knuteson

May 9, 2019
Milestone 3

Christopher R. Wirz (crwirz@gmail.com)
Mustafa Rabie (mustafa.rabie@gmail.com)
Masakazu Tanami (tanami@g.harvard.edu)
Lena Hajjar (hajjarlena@gmail.com)
Jacob Gardner (jacob_gardner@student.uml.edu)
Dan Little (little@g.harvard.edu)

Invention Disclosure System Using Blockchain

Search Engine Optimized using HTML Data Format and Commercial Vendor Replication

Christopher R. Wirz, Lena Hajjar, Masakazu Tanami, Mustafa Rabie, Jacob Gardner, Dan Little
crwirz@gmail.com,hajjarlena@gmail.com,tanami@g.harvard.edu,mustafa.rabie@gmail.com,jacob_gardner@student.uml.edu,little@g.harvard.edu

ABSTRACT

This paper presents a new system to search engine optimize invention disclosure content and provide immutable assertion of ownership and invention history by using blockchain and other commercial infrastructure providers. An HTML-based block is used to allow search engines to index invention disclosure content. While this system does not assert that the disclosed invention is novel and non-obvious, it does provide the necessary schema to determine that the invention has been reduced to practice by constructive means; it has been described in a way that someone of ordinary skill can produce the invention. This allows for a nearly free way to publicly disclose an invention in order to assert original authorship. If the inventor desires, this disclosure can easily be filed with a regulating body such as the United States Patent and Trademark Office (USPTO).

KEYWORDS

Invention, Disclosure, Blockchain, HTML, Patent

1 INTRODUCTION

Patent protection is a legal right that provides the patent-holder the exclusive right to make, sell, or license the invention as defined by the claims stated in the patent [1]. There are many economic benefits to this practice [2]. First, the inventor can recoup their investment from their research. Second, the disclosed invention provides a written description which other inventors can reference in their own pursuits. These benefits are only obtained when the inventor discloses. Generally speaking, intellectual property (IP) law has moved to a purely First to File [3] system, granting patent protection to the first person who reduces their invention to practice. Reduction to practice can be either actual, where the invention is carried out and is found to work for its intended purpose, or constructive, where the invention is sufficiently described so someone of ordinary skill can reproduce the invention.

In 2011, the US Congress passed the America Invents Act (AIA), which changed the criteria for patent protection from First to Invent (FTI) to First to File (FTF) [4]. An inventor cannot just patent an idea without it having been reduced to practice. The priority date is no longer the date of conception, but the date that the inventor reduced the invention to practice. Along with changing the patent protection system, the AIA also changed the definition of prior art: "A person shall be entitled to a patent unless the claimed invention was patented, described in a printed publication, or in public use, on sale, or otherwise available to the public before the effective filing date of the claimed invention" [5]. A patent application requires illustrations, tables, definitions, descriptions, claims, and a prior art search, along with an oath that the inventors listed were the first to invent.

First to File (FTF) patent protection ensures that inventions reaching the patent office are novel, non-obvious, and have been reduced to practice [6] (making it an invention vs. an idea). In order to be considered "novel," the invention must not have substantially similar prior art. Prior art references previously existing inventions that are offered for sale or publicly disclosed. Based on this criteria, if an inventor keeps the invention a secret [7] and reduces it to practice, he or she may file a patent and can reasonably anticipate protection if another inventor has not previously publicly disclosed or protected an idea with substantially similar claims or inventive concept. Many argue that with FTF, the only reason not to file for protection is if the inventor cannot afford the application fee and/or the cost to prepare the disclosure [8], or if the invention is not patentable to begin with.

An inventor may choose to disclose their inventions using a publication service, but since these have paid distribution channels, they are not truly public. Free publication options offer little proof that the dates and ownership of the intellectual property are accurate. Publication does not constitute disclosure. For these reasons, the concept of public disclosure offers the greatest challenge in protecting an invention as no publication service exists that prevents the modification of data while providing visibility to mainstream search engines. Many patent examiners and inventors alike use search engines throughout the invention life-cycle, particularly when conducting prior art search.

Blockchain can help with public disclosure of inventions. Lawyers often define blockchain as an electronic book of write-only transactions and an automated business process for storage of information about rights. While the problem of invention quality (such as defensibility or protection of claims) cannot be solved using blockchain, the software application creating the invention disclosure can address this by providing the inventor with guidance on what to include in the disclosure. Major governing bodies could easily adopt blockchain in IP rights management because the blockchain data structure can plug into any existing software. While the adoption of blockchain has been attempted in the financial technology (FinTech) industry, many societies are not yet ready to adopt it[9] due to current regulations. In contrast, with IP law, the blockchain enables the current spirit of the laws and practices.

2 REVIEW OF LITERATURE

The concept of Innovation Life-Cycle Management (ILCM) was popular in the early 1900s and regained a higher level of adoption in the early 2000s [10]. ILCM is the framework or an adopted framework, that allows inventions, ideas, and research to be combined in a single centralized system within an organization. There are many systems in place meant to capture ideas for the benefit of the inventor or the inventor's employer [11]. These systems are often

web-based to allow ease of access and organization of knowledge sharing. While patents prevail for innovation, defensive publication allows for idea claiming [12] in order to protect the evolution of an idea through the openness of disclosure. They do not prevent unauthorized use of the IP when this practice is federated into a corporate culture, it increases the rate of innovation.

When a governing body (such as the USPTO) adopts an ILCM framework, it becomes the main Intellectual Property Rights (IPR) tracking system that companies, small and medium enterprises (SMEs), and solo inventors target for a final publication. The focus of the USPTO is not so much on the process as it is the final reduction to practice [13].

2.1 Prior Research

In 2008, Satoshi Nakamoto described how blockchain could be used to cryptographically prove that a digital currency was not double spent, and thus invented Bitcoin [14]. Bitcoin groups transactions into a structure called blocks. Block information all bears the same timestamp and the hash of the most recent prior block. The many Bitcoin network nodes compute the hash value of the blocks such that the chain can continue in an auditable manner [15].

Because of this feature, which is part of the blockchain data structure, blockchains lend themselves to a distributed consensus system. As a large network brings about high-reliability and the structure of a blockchain is immutable without invalidating future blocks, many consider blockchain a distributed data structure as a permanent record. It also means that a hacker would have to have a network greater than the size of the blockchain's network in order to hack it.

Blockchain technology does not have to be associated with a crypto currency. To address the specific use case of microfilms, a highly copyable format, researchers Tsai et al. have proposed a solution to use blockchain to detect change and distribution of this form of intellectual property [16]. Blockchain solutions can be application specific or can cover a wide range of topics in a field such as big data [17]. Blockchains can be used to manage both public and private data while still decentralizing trust, identity, data ownership and consensus-based data-driven decisions.

There have been substantial advancements regarding the aspects of consensus and security. Additionally, other research has focused on the network usage and data integrity. Because of Bitcoin's popularity, research on the currency gained momentum with respect to blockchain privacy aspect. There has not been much focus on research regarding block format with respect to application-specific blockchains.

2.2 Prior Art

Several teams have developed blockchain-based intellectual property systems. These systems focus on different types of protection for disclosures.

2.2.1 Bernstein.io. Bernstein Technologies was founded in 2016, and is headquartered in Munich, Germany. Bernstein provides services to create a secure, private and encrypted digital trail of records for intellectual property (IP) assets. These include several IP services such as Secure Trade Secrets, Enhanced Contracts, and Innovation life-cycle Management. These services are provided by using several platforms like the Bitcoin blockchain network and the Bundesdruckerei [12] for digital timestamps.

Bernstein issues blockchain certificates for the materials uploaded to their servers, the generated certificate contains a cryptographic fingerprint of the IP asset and a proof of your ownership. The certificate is the root of the blockchain. Any work related to the project will be appended through evidence of work to this project's chain. Bernstein's unique "zero-knowledge architecture" provides guaranteed confidentiality, where users can generate blockchain certificates for their assets without disclosing information about their asset to any third party. All uploaded documents are encrypted at the client side before being uploaded to the servers.

2.2.2 Loci.io. Loci attempts to create an invention process that enables innovators to better find and claim their ideas using blockchain technology [18]. The Loci application uses a cryptocurrency called Loci Coin to compensate those who review disclosures and assist inventors in getting to the provisional patent level. One can also buy or sell intellectual property using Loci Coin. This dual use blockchain is probably the best implementation of a blockchain based disclosure system to date.

Combining search and machine learning, Loci also analyzes research trends and identifies opportunities to develop intellectual property.

2.2.3 Binded.com. Binded is a free service aimed at protecting digital media copyrights. The service provided by Binded allows artists, photographers, and other users with a way to upload their images to a digital vault and save them to the Bitcoin blockchain network. Users can upload their images from different channels such as computers and mobile phones, and the service integrates with third party applications such as Instagram and Twitter. The owners of the digital media receive a certificate as proof of ownership. In addition, owners are provided with a dashboard to monitor and track usage of their work. As of 2017 they have protected over 355,000 copyrights. Binded takes advantage of the fact that a copyright is not the same as a registered copyright (which is issued through the governing authority such as the USPTO) [19].

2.2.4 OriginStamp.org. The University of Konstanz's OriginStamp.org supports cryptograph timestamps with workflows built in for intellectual property. Originstamp is more focused on timestamping information [20, 21]. This is accomplished by storing the file hash in a Bitcoin block, which also has a verifiable timestamp. Originstamp has a simple API to retrieve the timestamp; one could verify a file by submitting it through the API which would return a trusted timestamp if one exists. Common use cases involve photography, recorded content, task completion, and invention disclosures.

3 DISCUSSION

When a patent examiner reviews a patent application, they basically perform a web search (using a search engine) and a patent search to uncover prior art. If no prior art is found, then the application can move forward. In 2019, Apple Inc. exploited this approach to successfully patent common programming language features such as optional chaining and generics [22]. Even through widely known, these language features are often found in paid literature or within source code itself. Therefore, the prior art search may not have

returned any similar descriptions matching the disclosure provided in Apple's application. Meanwhile, companies like Tesla have made their patents open to the public[23], ensuring an ample supply chain of competitively priced parts used in Tesla's core product line. Essentially, Tesla has treated the patent process as a defensive publication to ensure a supplier cannot corner the market using Tesla's patented technology. In both cases, a truly public disclosure system would alleviate the concerns highlighted by these two vastly different uses of the current patent process.

3.1 Advantages

Public disclosure of an invention will ensure that other entities cannot patent the idea only if the disclosure of the idea meets the criteria of a patent. If an inventor has an idea that he or she cannot reduce to practice, it does not constitute prior art and a public disclosure does not prevent another entity from producing the idea or protecting a substantiated reduction of the idea to practice. Often, research papers fall into this trap: many mainstream periodicals have a paywall preventing public publication, and many research papers provide proof of concept and not reduction to practice. If a subsequent publication can reduce the concept to practice, it counts as prior art and not the original research. In this case, filing a provisional patent allows the original inventor to file first, and provides them one year to reduce the invention to practice and file a full patent. There is also a fee associated with a provisional patent.

If an inventor has successfully reduced an idea to practice and publicly discloses the invention, the inventor has one year to file a patent. If another inventor makes a subsequent publication regarding a substantially similar idea, it does not nullify the first inventor's right to protect the invention. According to Legislative History Senator Kyl (R-AZ), "Once the U.S. inventor discloses his invention, no subsequent prior art can defeat the invention. The U.S. inventor does not need to prove that the third-party disclosures following his own disclosures are derived from him" [24]. This is the "free" option to protect an invention (called defensive publication), but the method to publicly disclose is open for interpretation. While an inventor can self-publish on the internet, which is public, it is challenging to establish the integrity of the date (it is easy to modify a date field in a database) and the fidelity of the invention (the body of a document is also easy to modify) at that time. Therefore, without substantiating proof by a third party or a publicly verifiable and immutable data structure, the disclosure may be defeated in favor of a subsequent inventor that managed to file for a patent first. This is one of the greatest challenges with FTF and the use of a free and public blockchain based invention disclosure system can address this need.

As described in a recent National Institute of Standards and Technology (NIST) study, "Block chain is a storage framework that is tamper resistant and has a native synchronization-discrepancy resistance mechanism[25]." Blockchain is the perfect data structure to assert precedence and authorship; it provides mathematical proof of a public disclosure. Lowering the barrier for a public disclosure using this approach may also help smaller inventors retain rights to their inventive concept.

3.2 Risks and NPEs

While First to File has made it so authors must reduce an invention to practice in order to receive a patent, there are still non-practicing entities (NPEs) that intend to hold a patent and wait for the product to come to market, only to receive compensation after the successful undertaking of another [26]. These individuals are often called "patent trolls", and they account for about 60% of all patent lawsuits.

Because NPEs can be successful, they do bring about innovation and disclose often useful technologies. With a public disclosure system, an NPE could have two roles. The first role is as an inventor: disclosing everything and anything in order to assert invention date and ownership such that they can never be barred from producing and marketing their idea. The second role may be as someone who acquires an inventors' assertions (ownership and date) and applies for formal protection with the governing authority. With a public immutable disclosure system, a third opportunity may exist in which the original author may not be the same entity that has received legal protection. In this case, the original author would not be barred from producing or marketing the product, so an NPE could assist in placing the original inventor with a company that needs the legal right to market and sell the product.

4 DESIGN

The system presented by this paper is first and foremost an invention disclosure system. This system contains the same schema[27] as the USPTO, meaning that the content can easily transition into a full (or provisional) patent application. Also, enforcing a schema identifies of areas in which the invention has not been fully reduced to practice. The schema enforces a patent cross-reference section that requires the inventor perform a basic prior art search.

The basic requirements of a disclosure system are as follows:

Accounts A user can create an account, login, and update various authorship information

Disclosure Drafts Drafts can be created, edited, deleted and shared with other authors for collaboration and co-authorship. A draft follows the necessary schema to disclose with the USPTO.

Defensive Publication Users add a static copy of a complete disclosure such that it can be publicly viewed and discovered in an ordinary prior art search.

These requirements fulfill the necessary intention of a disclosure system that will allow an inventor to document an idea that has been fully reduced to practice. These requirements do not address the immutability/verification of the disclosures. For this reason, the system presented in this paper adds the requirement that completed disclosures are stored in a blockchain:

Disclosure Blocks Blocks persist the invention disclosure on the blockchain. Each block has a name equal to its own hash and has a reference to the prior block within its header.

Currently, no blockchain exists that will support public disclosure and publicly discoverable content. This has to do with the block format and how mainstream search engines are configured. Search engines are designed primarily to crawl and return results

for HTML format documents. This paper proposes a novel block format that will facilitate indexing by search engines through modified HTML.

4.1 Block Format

There are many options for a block format [28, 29], but the only one that is truly applicable in a defensive publication is HTML. Any browser can naively display HTML regardless of the tags used. Some tags are rendered in a specific way as defined by the World Wide Web Consortium (W3C). Tags that are not part of this specification will render the contents as text and will not render the attributes. Because modern browsers use a document object model (DOM) to display and contextualize HTML elements, JavaScript and Cascading Style Sheets (CSS) can provide additional display information. Side loaded visual markup in this manner keeps the block content small which results in a faster load time for the search engine's indexer.

This is accomplished with a single index.js inclusion at the bottom of the block.

```
<script src="index.js" />
```

The reason for adding JavaScript and not CSS is because JavaScript can add CSS, but CSS cannot always add JavaScript. We expect that most clients using the blockchain will be calling using HTTP and HTTPS protocols. Statically hosted blocks can be served easily to any modern browser.

When a user arrives at a block via a web browser, the JavaScript will run, but when using ajax calls from a third-party block explorer, the client can call custom code after receiving a 200 status (OK) response.

4.1.1 Basic SEO Tags. Search Engine Optimization (SEO) involves annotating and constructing content in a way that encourages a search to rank a page as a top result for a given search criteria. Inventions that are novel and extend beyond prior art, they need to be a top result for their core intuition and inventive concept.

```
<title>The title</title>
<meta name="description"
    content="first 140 characters of the abstract" />
<meta name="robots" content="index, follow" />
```

Inside the body tag of the HTML, the title of the disclosure should be inside of a H1 element.

```
<h1 id="title">The title</h1>
```

Using this approach, modern SEO best practices are met, and most search engines should rank and index the content of blocks well. `sitemap.xml` and `rss.xml` are used in a traditional manner.

4.1.2 HTML Tags. There are two approaches that can be used when storing block information in HTML format. The first uses web-standard tags. This approach would appear as follows:

```
<div id="summary">The summary</div>
```

This approach allows the disclosure content to immediately render in a web browser, albeit with only the default style provided by the browser. The problem with this approach arises if a client shows two disclosures on the same page. This will cause two elements to have the same id, which is not favorable or best practice in web application development as ids are supposed to be unique.

The second candidate approach defines a tag for each piece of content.

```
<summary>The summary</summary>
```

This is only slightly more concise, but it has the advantage of allowing for multiple disclosures to exist on the same page as any HTML node supports the method `getElementsByTagName`, which can find and return a requested child with the given tag. This allows for content within a disclosure to be uniquely addressed so long as tags are utilized. This paper proposes the following tags, which also match the schema for disclosure with the USPTO:

- title
- priorhash (the prior block)
- signature (optional)
- timestamp
- authors
- abstract
- keywords
- figures (base64 encoded as img src)
- field (of the invention)
- summary
- descriptionpriorart
- patentcrossreferences
- briefdescriptionofdrawings
- detaileddescriptionofdrawings
- detaileddescriptionofinvention
- enhancements
- benefits
- otherembodiments
- claims

In modern browsers, CSS can be used to style elements by tag name as well. For this and the aforementioned reasons, the proposed block structure uses tags to identify the necessary content.

The only exception to the custom tag usage is figures. Figures are stored using the img tag, which displays in browsers. The src of the image is set to that of the base64 uri of the image data and the various properties of the figure are img attributes.

Blocks are stored with the filename the same as their ipfs hash.

4.2 Software Design

The software design meets the requirements described previously with the focus on the production and curation of disclosures. Users are authors of disclosures and can store drafts semi-privately (they can be shared with co-authors) or publish a disclosure to a public immutable setting. This involves using the proposed blockchain. Blocks contain disclosures, and disclosures have claims, figures, and the aforementioned fields. The class diagram is provided in Figure 1. An important factor is that once a block is written, it is never changed.

Given this class diagram, it is apparent that either a non-relational or relational database will work for storage of all information required by the disclosure system. Software architecture will be discussed more in section 4.3, but the proposed system uses static storage of JavaScript Object Notation (JSON) files for simplicity and faster transaction times. Using long (somewhat un-guess-able) object keys, the files can be statically stored though the system's REST API and retrieved using a simple HTTP GET.

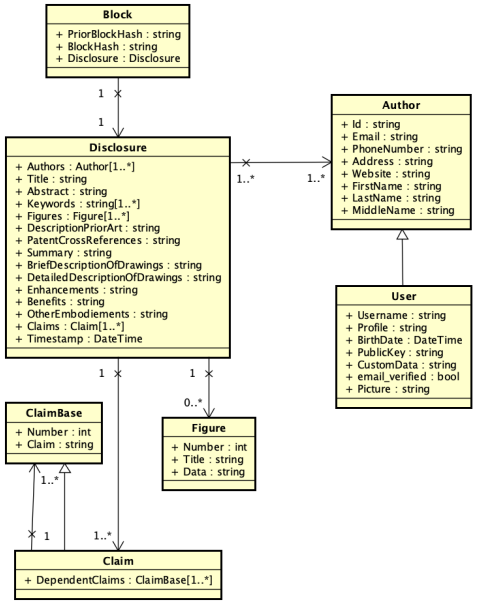


Figure 1: The class diagram

4.3 Software Architecture

The proposed system was constructed as a serverless application on Amazon Web Services. The overall architecture is given in Figure 2. The API can be deployed using a serverless template as to better meet some of the secondary requirements of the system such as authentication and auto-scaling.

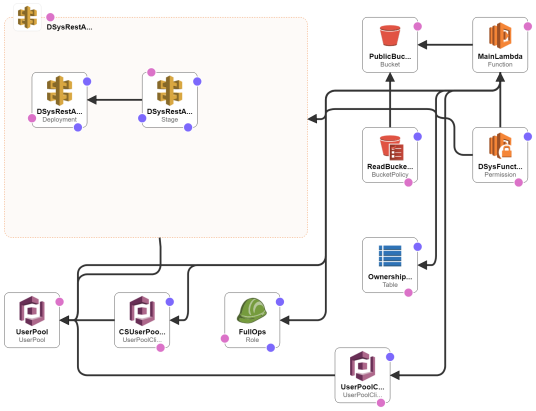


Figure 2: The Application Architecture

The system has several main components:

API Gateway The Application Programming Interface (API) Gateway connects the front-end to the back-end through various REST requests. The API gateway provides user, disclosure, and block functions – as well as a self-describing Open API endpoint for automatic Software Development Kit (SDK) generation.

Cognito Amazon Web Service (AWS) Cognito is used for user management. This solution is cost effective per monthly active user and can be used to authenticate API endpoints.

AWS Lambda The API Gateway calls serverless lambda functions which perform the business logic and operations regarding S3, Dynamo, IPFS, and Ethereum.

S3 Storage Simple Storage Service (S3) provides a mirror of all static blocks, a host of the single page application HTML code, and user and draft JSON data.

DynamoDB DynamoDB is a high-performance key-value store that is used to provide security-controlled data joining information. This enables static storage of semi-related JSON data to work effectively when temporary relationships are formed. Disclosure draft data is an example of data which uses a temporary relationship.

The main user interface client is a single page application (SPA) that can be reached at the domain `discoluresys.com`. The client calls the API gateway and retrieves JSON data directly from S3.

All static content is mirrored on IPFS for redundancy.

4.4 Disclosure Life-Cycle

In order to persist in both S3 and IPFS mirrors, a well-defined sequence of events must take place. The system uses a unique combination of AWS, IPFS, and the Ethereum blockchain. These events are pictured in Figure 3.

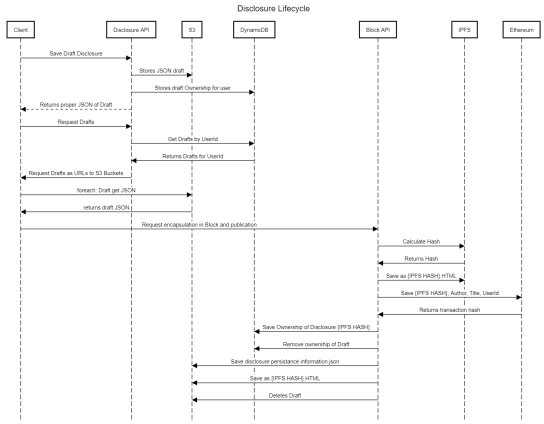


Figure 3: The Life-Cycle of a disclosure submission

When an author finishes a disclosure and hits save, the information is sent to the API as JSON. The author may also provide a cryptographic signature. The API then wraps the disclosure in a block and converts it to HTML. To do this, it must get the last block hash and put it in the header of the new block. The HTML is stored on IPFS, which calculates the permanent hash. The API then stores the HTML file on S3, with the hash as the file name. The hash information is stored on the Ethereum blockchain with title, authorship, and timestamp information. Any draft information for the disclosure is garbage collected and the disclosure hash is added to the authors' list of disclosed inventions.

5 RESULTS

In order to test the disclosuresys.com application, the team used the system to produce a disclosure regarding the system as seen in Figure 4. This led to many design improvements. From a user perspective, the system is faster and easier to use than any known disclosure system on the market such as DiamsIQ, Loci, and the USPTO system.

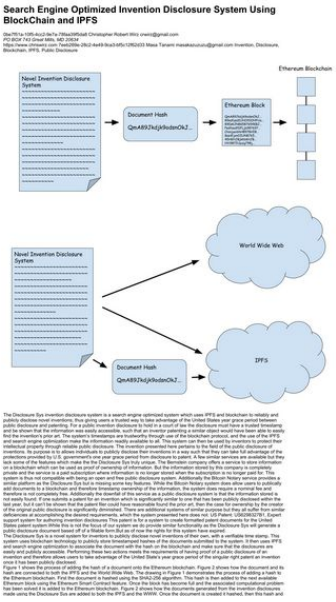


Figure 4: Un-styled sample disclosure

Much of the speed improvement has to do with asynchronous API execution and the liberal use of browser-based session storage and local storage by the client SPA. Our client caches and prefetches so much data that load times appear almost instantaneously. Additionally, communication with the auto-scaling API involves sending unique and minimal amounts of information.

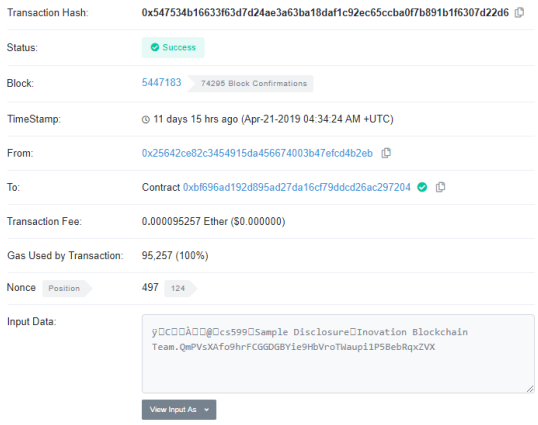


Figure 5: Results of storing a block hash on the Ethereum blockchain

The team successfully collaborated on an invention disclosure and published it to the blockchain. The disclosure can be viewed on both S3 and IPFS. The hash has been verified in Ethereum block explorer (as shown in Figure 5). In other words, the disclosure is public, fault tolerant, and immutable.

5.1 SEO

Using various test tools, such as seotesteronline.com and tools.neilpatel.com/en/analyze/ , published blocks are in the 95th percentile of SEO optimization with a majority of errors involving lack of keyword appearance in the disclosure body. This could be improved by prompting the user to include these key words in the body, but overall, the HTML content is sufficient for SEO purposes. Because of mirrored hosting on S3 and IPFS, as well as asynchronous requests for JavaScript, page load times are around 13ms. Figure 6 shows the results of SEO analysis.

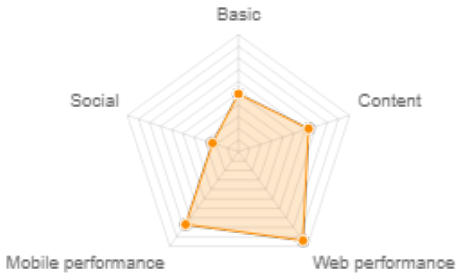


Figure 6: SEO priority results

As expected, the page performance is the strongest SEO component as due to the fast load times and redundant hosting. It only lacks performance because it is a single page load. Instead of loading images separately, which could return an elongated expiration header, the page loads with all figures included as base64 URIs. This increases the load time of the initial HTML but reduces the number of requests. Since figures are a key component of invention disclosure blocks, there is only so much performance to be expected.

The second SEO component is the content. This is a balance between the block design and the actual content which the user submits in the disclosure. In future versions of the system, we may add features for the author to improve the SEO of the disclosure. However, since the initial focus is achieving constructive reduction to practice [30], the SEO performance is probably as good as can be expected.

The system is currently lacking in Basic SEO because it is a new system. Sitemap and RSS feeds have not been heavily syndicated and there are very few backlinks beyond the team's personal profiles. Social SEO is by far the weakest area. While the HTML has all the necessary social meta tags, there has not been an active link-sharing campaign.

When given an overall score, disclosure blocks receive an "A" as shown in Figure 7. This indicates a good balance between the possible trade-offs and the implementation of SEO best practices on an individual HTML block basis.

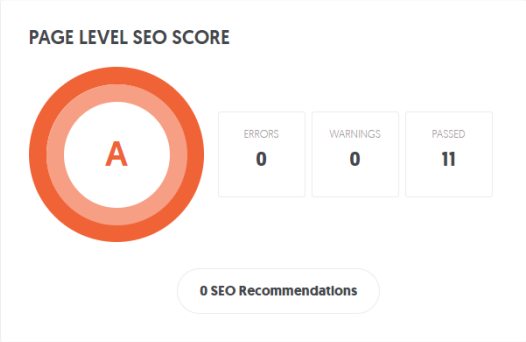


Figure 7: Results from tools.neilpatel.com

6 DISCUSSION

The proposed system is high-performance disclosure application that uses blockchain to create an immutable record of timestamp and authorship. Using the custom HTML block design, independent SEO audit tools have indicated that best practices for search engine indexing and ranking have been implemented.

Due to the infancy of the system at the time of this paper, SEO basics such as back-linking and social have not been established. However, given sufficient time and indexing, content created by the application should be publicly searchable in a reasonable amount of time. As the schema of each disclosure meets the criteria for a submission to the USPTO's FTF system, the proposed system enables inventors to reduce their idea to practice by construction.

The user-facing SPA is high performance as it leverages front-end caching, minimalist message formats, and an auto-scaling asynchronous API. Partial drafts can be stored on the web, shared, or maintained locally. While collaboration exists, updates only occur when other users save their changes. Real-time collaboration may be a feature in the future, leveraging the auto-scaling API to its full extent.

7 CONCLUSION

While not novel or non-obvious, an open-source public blockchain based invention disclosure system will help drive and protect innovation and protect intellectual property [20]. Other systems fail to provide the necessary schema to constitute an invention as well as the search engine optimization required to make the disclosure public. As the system uses a similar schema to the USPTO, public disclosure can more easily transition into legal protection after the viability of the invention has been assessed. This helps reduce some of the unnecessary applications in the backlog of the USPTO's process.

The presented system offers benefits to independent inventors who would otherwise have little way of verifying a date, content, and claims of a disclosure. Public disclosure ensures that the invention serves as prior art and prevents substantially similar inventions from receiving protection the would negatively impact the original author's ownership rights of the idea. For NPEs, this creates an opportunity to broker rights established through defensive publication on top of the traditional rights achieved through the formal patent application practice.

7.1 Future Work

While the proposed system shows a lot of promise and differentiation, there are opportunities to improve the state of the art. We provide two examples.

7.1.1 Improved consensus and queuing. The current system operates on the hope that the disclosure submission order is the same order that which they end up on the blockchain. This could be guaranteed by a consensus algorithm or by use of Amazon's Simple Queue Service (SQS). This would alleviate the requirement of order-based consensus and allow for just block hash consensus.

7.1.2 SEO validation. The disclosure body must contain the identified keywords in order for the SEO to be effective. This business rule could be executed in the business logic code behind the API. The only downside would be that the response message size would increase. The advantage would be that each client would not have to implement the business rules.

7.2 Recommendations

A blockchain is a data structure, not the foundational technology of an application. In the case of this disclosure system, a high percentage of users will never notice that a blockchain has been implemented at all. If mainstream solutions do not meet the needs of an application, it is reasonable to store a hash of the application block on the mainstream blockchain.

When building a blockchain, the format of the block has to extend the capabilities of the application in a meaningful way as incorporating a blockchain is often a trade-off with respect to the quality of necessary product features. Blocks need a way to relate to each other as well as persist to disk. If a block can easily marshal between memory and disk, the blockchain can serve as a viable database for the necessary application information. If the database data is in anyway improved through distribution and immutability, then a blockchain is an important application design consideration.

8 APPENDIX

The appendix contains additional code samples and charts not found in the body of the paper.

8.1 Sample HTML head

The following is a sample of the HTML block head necessary to provide both SEO and blockchaining.

```
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type"
    content="text/html; charset=UTF-8" />
<title>Title...</title>
<meta name="description"
    content="Description ..." />
<meta name="robots" content="index, follow" />
<meta itemprop="name"
    content="Title..." />
<meta itemprop="description"
    content="Abstract..." />
<meta name="twitter:card"
```

```
    content="summary" />
<meta name="twitter:title"
    content="Title..." />
<meta name="twitter:description"
    content="Abstract..." />
<meta property="og:title"
    content="Title..." />
<meta property="og:type" content="article" />
<meta property="og:published_time"
    content="2019-05-02T20:16:42+01:00" />
<meta property="og:description"
    content="Abstract..." />
<meta property="article:tag"
    content="Keywords..." />
<priorhash>PriorHash...</priorhash>
</head>
<body>
<h1 id="title">Title...
```

REFERENCES

[1] Sue A. Purvis. Basics of patent protection, 2013.

[2] Roberto Mazzoleni and Richard R Nelson. The benefits and costs of strong patent protection: a contribution to the current debate. *Research policy*, 27(3):273–284, 1998.

[3] Charles RB Macedo. First-to-file: Is american adoption of the international standard in patent law worth the price. *AIPLA QJ*, 18:193, 1990.

[4] Joe Matal. A guide to the legislative history of the america invents act: Part i of ii. *Fed. Cir. BJ*, 21:435, 2011.

[5] Leahy-smith america invents act, 09 2011.

[6] Christian J Garascia. Evidence of conception in us patent interference practice: Proving who is the first and true inventor. *U. Det. Mercy L. Rev.*, 73:717, 1995.

[7] David R Hannah. Should i keep a secret? the effects of trade secret protection procedures on employees' obligations to protect trade secrets. *Organization Science*, 16(1):71–84, 2005.

[8] Brad Pedersen and Vadim Braginsky. The rush to the first-to-file patent system in the united states: Is a globally standardized patent reward system really beneficial to patent quality and administrative efficiency. *Minn. JL Sci. & Tech.*, 7:757, 2005.

[9] Svetlana Saksonova and Irina Kuzmina-Merlino. Fintech as financial innovation–the possibilities and problems of implementation. *European Research Studies*, 20(3A):961, 2017.

[10] Romain Beaume, Remi Maniak, and Christophe Midler. Crossing innovation and product projects management: A comparative analysis in the automotive industry. *International Journal of Project Management*, 27(2):166–174, 2009.

[11] Alba Sánchez, Alejandro Lago, Xavier Ferràs, and Jaume Ribera. Innovation management practices, strategic adaptation, and business results: evidence from the electronics industry. *Journal of technology management & innovation*, 6(2):14–39, 2011.

[12] Josep Lluis de la Rosa, Denisa Gibovic, V Torres, Lutz Maicher, Francesc Miralles, Andrés El-Fakdi, and Andrea Bikfalvi. On intellectual property in online open innovation for sme by means of blockchain and smart contracts. In *3rd Annual World Open Innovation Conf. WOIC*, 2016.

[13] TV Shatkovskaya, AB Shumilina, GG Nebratenko, Ju I Isakova, and E Yu Sapozhnikova. Impact of technological blockchain paradigm on the movement of intellectual property in the digital space. *European Research Studies*, 21:397, 2018.

[14] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[15] Michael Crosby, Pradan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, et al. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2(6-10):71, 2016.

[16] Wei-Tek Tsai, Libo Feng, Hui Zhang, Yue You, Li Wang, and Yao Zhong. Intellectual-property blockchain-based protection model for microfilms. In *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pages 174–178. IEEE, 2017.

[17] Elena Karafiloski and Anastas Mishev. Blockchain solutions for big data challenges: A literature review. In *IEEE EUROCON 2017-17th International Conference on Smart Technologies*, pages 763–768. IEEE, 2017.

[18] loci.io. Loci whitepaper v7, 2018.

[19] Sarah Anderson. The missing link between blockchain and copyright: How companies are using new technology to misinform creators and violate federal law. *North Carolina Journal of Law & Technology*, 19(4):1, 2018.

[20] Alexander Schönhals, Thomas Hepp, and Bela Gipp. Design thinking using the blockchain: Enable traceability of intellectual property in problem-solving processes for open innovation. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pages 105–110. ACM, 2018.

[21] Alexander Schoenhals, Thomas Hepp, Philip Ehret, and Bela Gipp. Tracking of intellectual property using the blockchain.

[22] Apple Inc. Programming system and language for application development.

[23] Steve Brachmann. Tesla battery patents further proof of elon musk's duplicitous views on patents, 05 2017.

[24] Reg Patent Attorney. Defending the uspto interpretation of the new grace period. 2013.

[25] Sylvere Krima, Sylvere Krima, Thomas Hedberg, and Allison Barnard Feeney. *Securing the digital threat for smart manufacturing: A reference model for blockchain-based product data traceability.* US Department of Commerce, National Institute of Standards and Technology, 2019.

[26] David L Schwartz and Jay P Kesan. Analyzing the role of non-practicing entities in the patent system. *Cornell L. Rev.*, 99:425, 2013.

[27] Wai Ming Wang and Chi Fai Cheung. A semantic-based intellectual property management system (sipms) for supporting patent analysis. *Engineering Applications of Artificial Intelligence*, 24(8):1510–1520, 2011.

[28] W Liu, SS Zhu, T Mundie, and U Krieger. Advanced block-chain architecture for e-health systems. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6. IEEE, 2017.

[29] Dimaz Ankaa Wijaya, Joseph K Liu, Dony Ariadi Suwarsono, and Peng Zhang. A new blockchain-based value-added tax system. In *International Conference on Provable Security*, pages 471–486. Springer, 2017.

[30] Warren H Willner. Origin and development of the doctrine of constructive reduction to practice. *J. Pat. Off. Soc'y*, 36:618, 1954.

Journal Selection

To test that our disclosure system seamlessly transitions into the existing patent pipeline, the ideal publication would be a patent application with the USPTO. Unfortunately, the cost is $2,720 and this publication is not a traditional approach for a capstone.

The Association for Computing Machinery (ACM)'s Journal of Data and Information Quality (JDIQ) is the best fit for this project (https://mc.manuscriptcentral.com/jdiq). It is a peer-reviewed journal that focuses on topics regarding data storage improvements, including systems building descriptions. Submission is free, and this journal has already published articles in this domain, such as "Design Thinking using the Blockchain: Enable Traceability of Intellectual Property in Problem-Solving Processes for Open Innovation."

Another option for publication is The Harvard Business Review (HBR). HBR publishes articles about many topics, including technology. Publishing in HBR is free, but submissions need to meet the qualifications listed on their website (https://hbr.org/guidelines-for-authors). HBR already published several articles about blockchain technology such as "How Blockchain Will Accelerate Business Performance and Power the Smart Economy" [33].

Architecture Design

High Level System Architecture

The DisclosureSys product uses three main services: Amazon Web Services (AWS), InterPlanetary File System (IPFS), and Ethereum blockchain network. IPFS and Ethereum are used as product integrations, and the DisclosureSys application does not have a hard dependency on these services. DisclosureSys is first-and-foremost a disclosure system - that is only made better by participation in third-party blockchains (Ethereum) and redundant storage (IPFS).

Figure 8 below, illustrates the system architecture of the DisclosureSys application. As the entire system is hosted on AWS and deployed using an AWS serverless template, this diagram covers essentially the items which have an associated cost to the DisclosureSys project. For the sake of clarity, IPFS and Ethereum are omitted from Figure 8 as they are optional to the core application functionality.

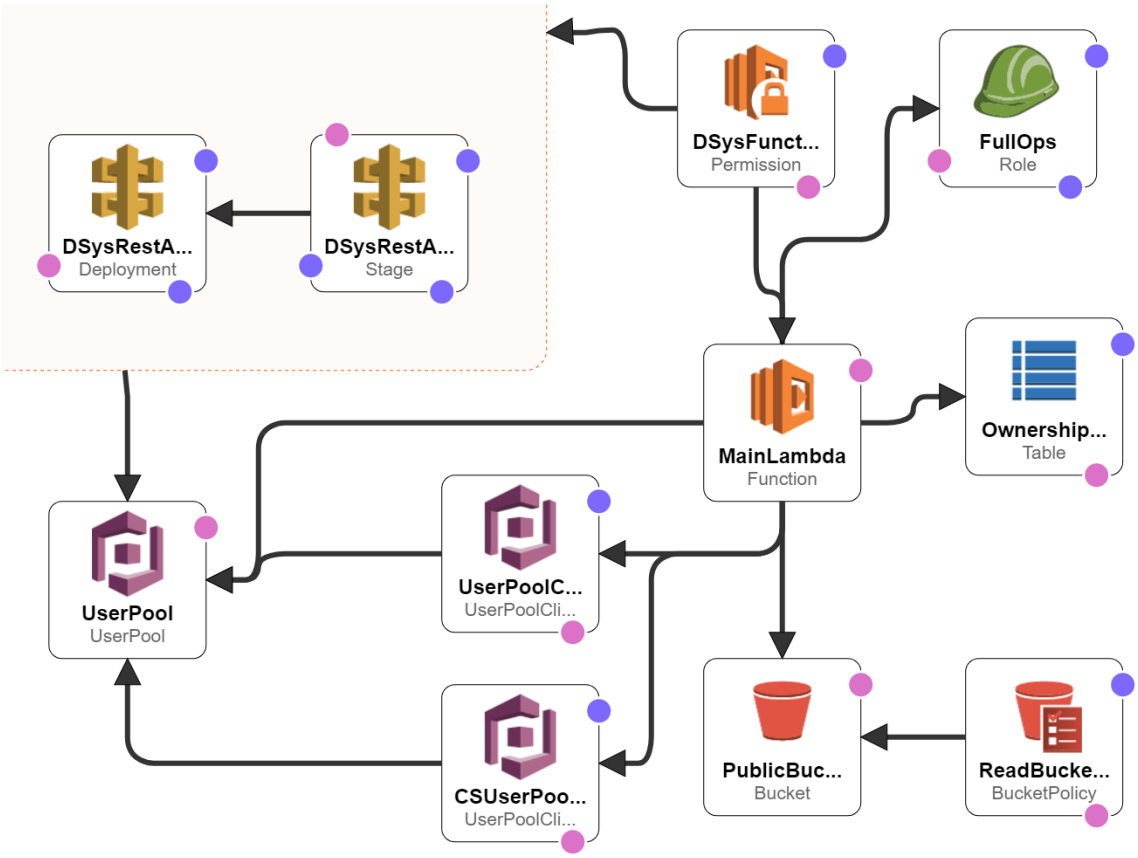


Figure 8: High Level System Architecture of the DisclosureSys

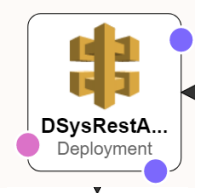
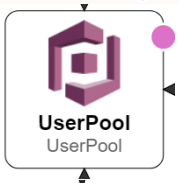
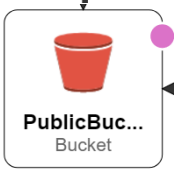
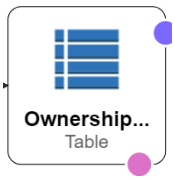
AWS Icon	Description
	This is the deployment environment. We have Staging and Production environments. The Production environment is public facing.
	This is the security group for all users on the DisclosureSys site. The access levels are broken out into client-level (from the web) and code-level (from within the lambda function).
	This is the DisclosureSys main API. It rate-limits and authenticates responses. It also elastically scales based on web traffic.
	This is the AWS Simple Storage Service (S3) bucket (public storage) for storing the disclosures and user content as well as the main Single Page Application.
	This is the AWS DynamoDB (NoSQL database) table for storing the ownership information (such as drafts and published disclosures).

Table 1: Service Icon Key

The primary user-facing application is a static single page application (SPA). It uses JavaScript (JS) and calls a REST (Representational State Transfer) API (Application Programming Interface). This strategy also supports native Android and iOS mobile applications. The API is written in C# and executed as Amazon Lambdas. AWS Cognito is used for user management. Ownership information is stored in DynamoDB. Drafts (stored as JSON) and final Blocks (Published Disclosures - stored as HTML) are hosted on Amazon S3 with the privileges of public read (after written by a Lambda function). Storage in S3 facilitates Search Engine Optimization. The cost for this service is $5/month/1M users and used to authenticate all the API endpoints. The process of saving drafts through AWS as shown below.

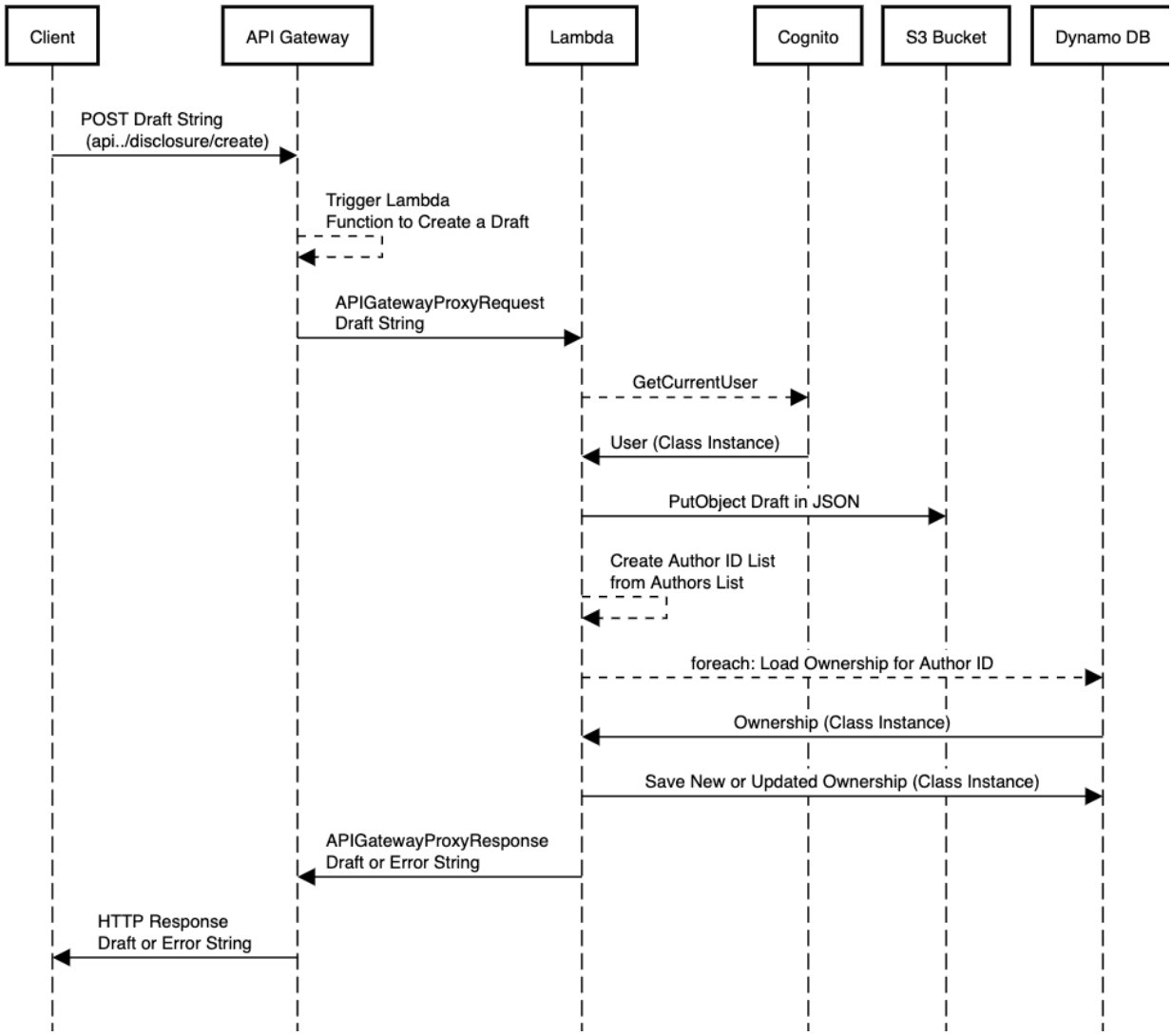


Figure 9: Saving Disclosure Drafts with AWS

The above sequence diagram (*Figure 9*) shows the process of saving a disclosure to the Ethereum blockchain. After saving, the HTML formatted disclosure block (an HTML wrapper around the disclosure content) is persisted to the Amazon S3 bucket. Only one persistence (in addition to S3) is required to be successful in order to publish the disclosure block. While most blockchains ensure redundancy through replication, DisclosureSys uses multiple blockchains for redundancy as well. Therefore, Ethereum could fail, or IPFS could fail, but an inventor could still successfully publish a disclosure. If S3 fails, DisclosureSys would fail to load as the application is hosted on S3.

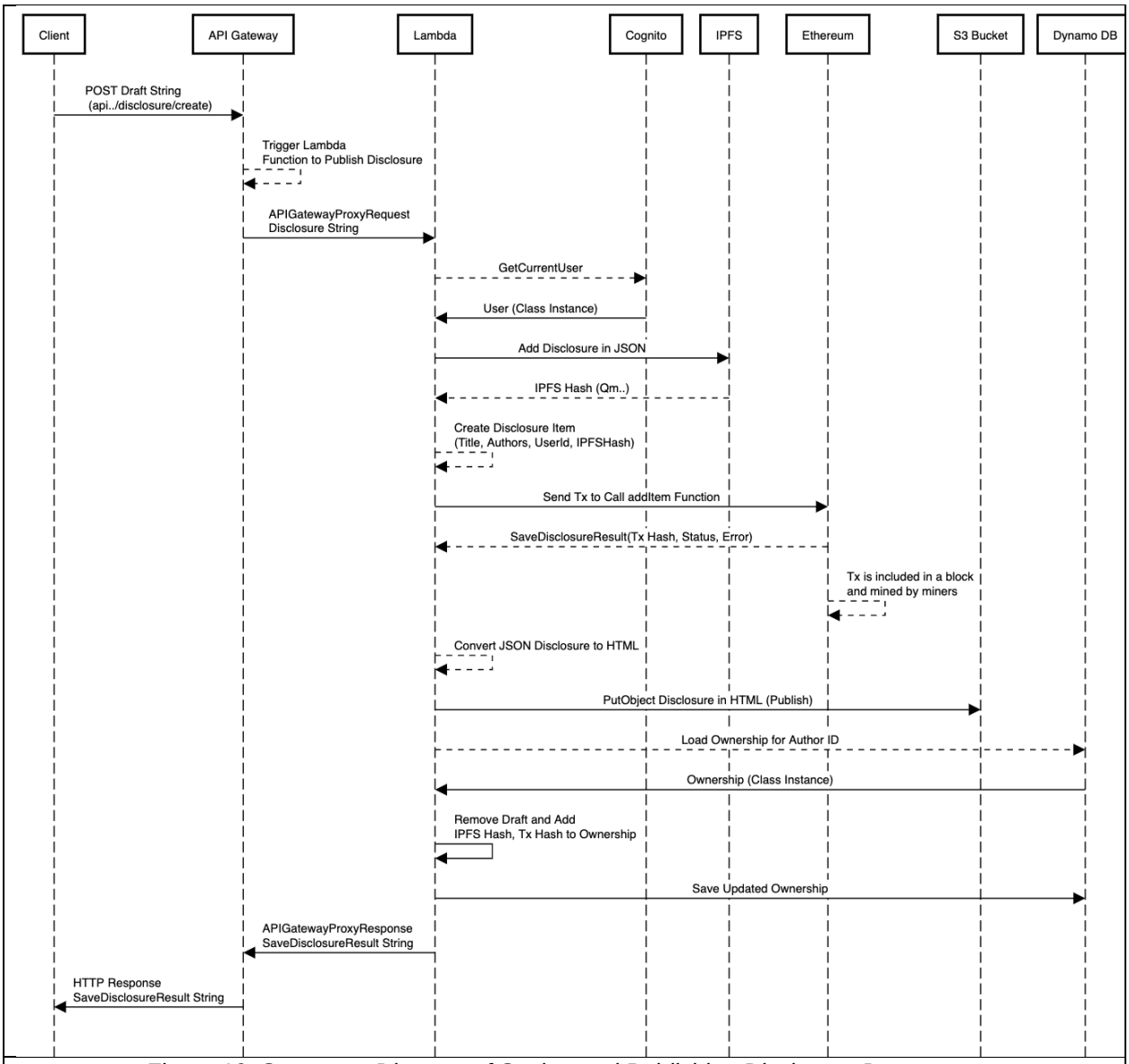


Figure 10: Sequence Diagram of Saving and Publishing Disclosure Documents

The core intuition of this design, the HTML block, facilitates Search Engine Optimization (SEO) and provides a schema which re-enforces matches that of an invention which has been reduced down to practice. Among other things, this includes a description, figures, description of and benefits over prior art that may exist, and claims. Since there is a size limitation to Ethereum blocks, the HTML-block is loaded to IPFS, and then the hash that the IPFS protocol returns is written to an Ethereum blockchain. Disclosures are separate from Blocks because Disclosures (Drafts) may be edited, while Blocks (Published Disclosures) cannot be edited. When the Disclosure is published, a block is created with the disclosure, the block hash, and the hash of the most previous block

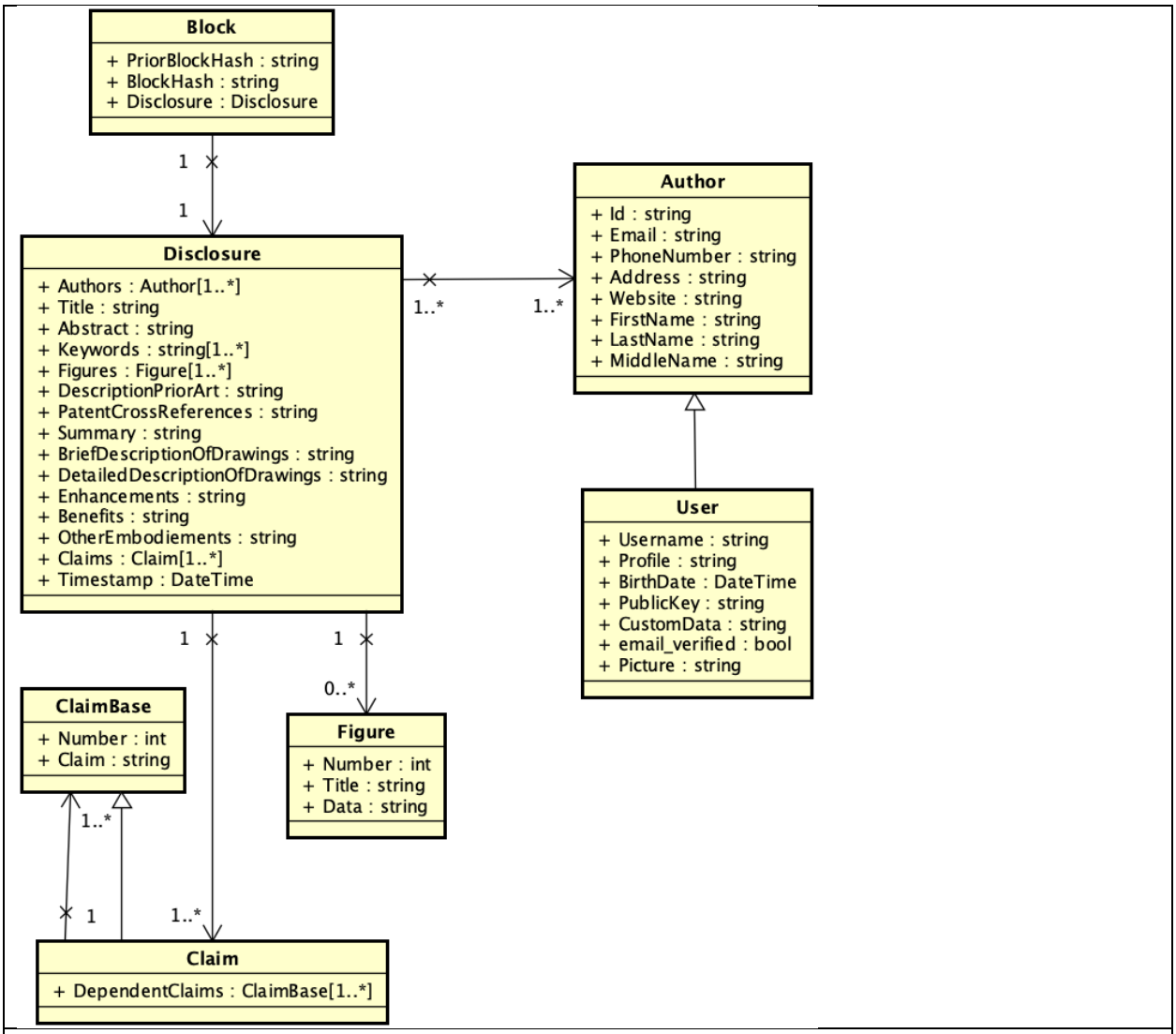


Figure 11: UML design of Disclosures

Testing

During this stage of development, the team has relied on extensive regression testing. The team built a test tool and used version controlled JSON objects as sample data for each query.

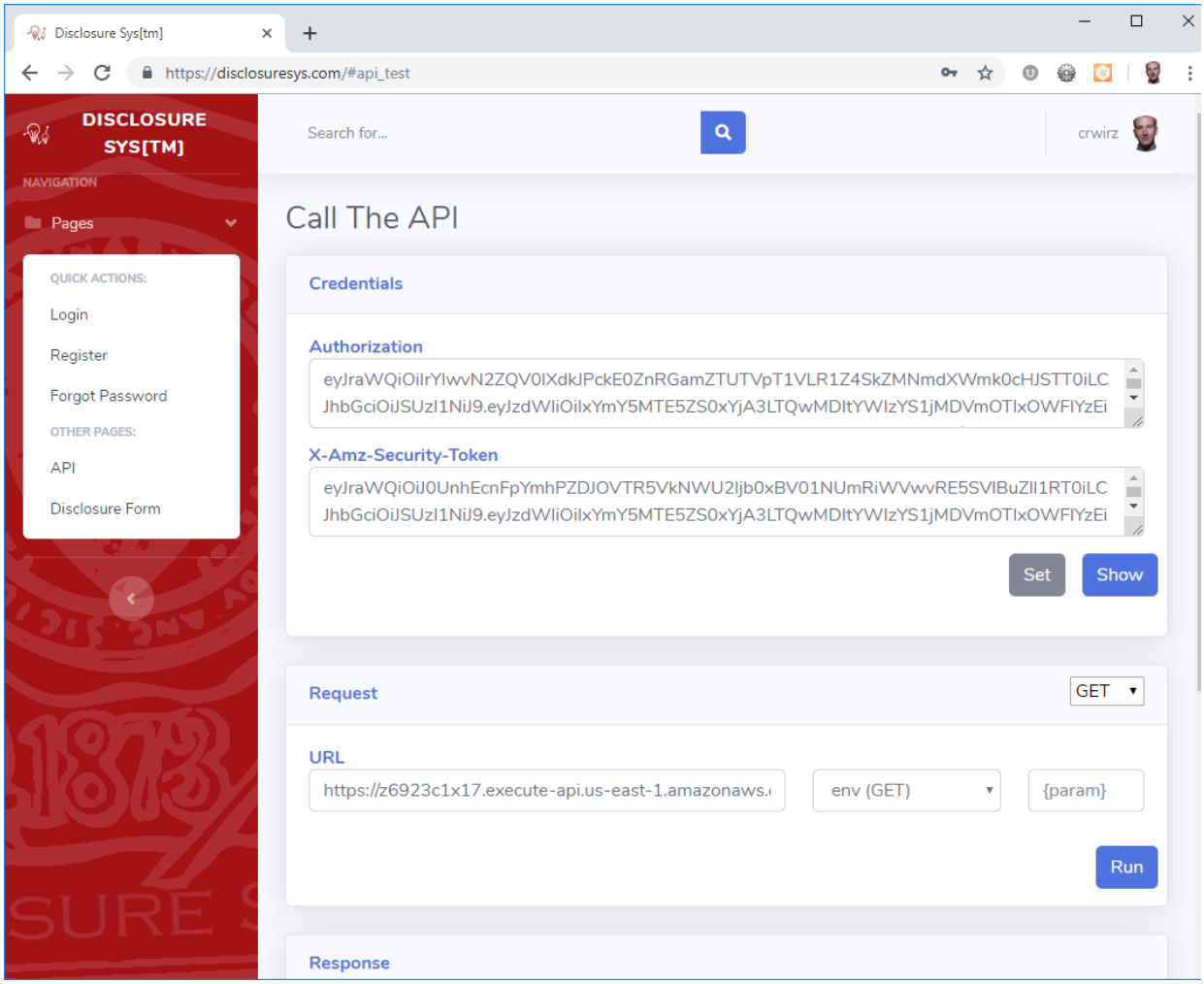


Figure 16: The API Test Tool

For Milestone 2, the team increased utilization of C# unit tests with mixed success. This is due to the fact that AWS provides little information as to the runtime environment of its Lambda platform. While using .Net Core provides some sense that you can “write once run anywhere in C#,” this is only true for the most vanilla libraries. Working with images, hashing blocks, and communicating with IPFS all worked locally, but not in production.

After resolving these idiosyncrasies for Milestone 3 and becoming more realistic about what can be tested, we successfully leveraged C# unit tests for all the core

backend API requirements. This also allowed us to test/debug the API functions locally. The results of our application passing all our tests can be seen in Figure 17.

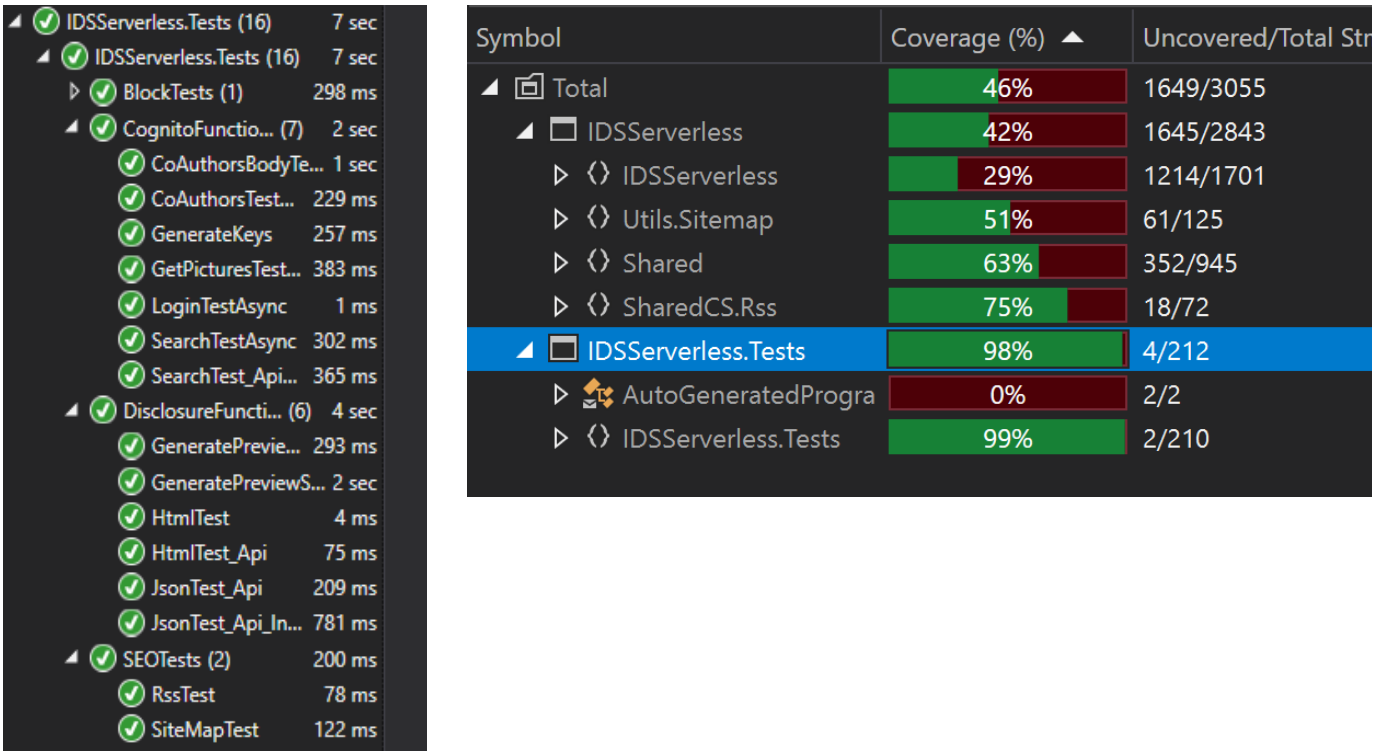


Figure 17: Unit test and Code Coverage Report

Web Performance Optimization

Our disclosure system is not only for publishing invention disclosures on the web, but also for making them easily searchable by Google and other search engines. Therefore, optimizing the page loading speed is an essential component of the project requirements. The page load time for our top page took 10 to 15 seconds before optimization (in certain regions of the world), and 1 to 4 seconds afterward. To optimize this process, we used two tools, Chrome DevTools Network tab and Lighthouse Audits tool.

Chrome DevTools Network Tab

Chrome DevTools tool analyzes how HTTP requests load each web component. In the beginning, the loading time showed 13.53 secs, and the Waterfall showed that some CSS and JS files were blocking the next operations. We identified the bottleneck of the data and split it into parts that load asynchronously or used lazy loading with dynamic importing.

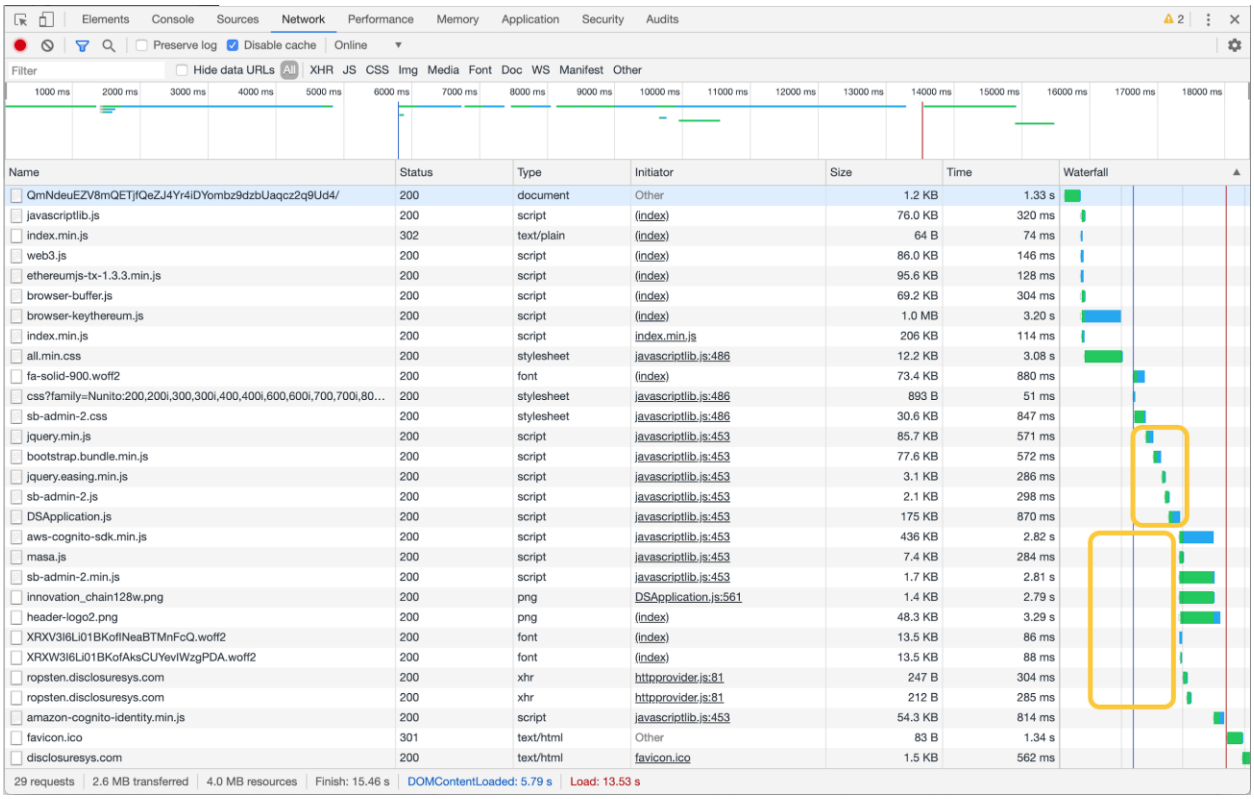


Figure 12: Network Tab Before Optimization

The goal was to make the Waterfall look as flat as possible. However, the implementation was not entirely straightforward because a JS file is dependent on others, and internal dependencies exist in a large JS application. We searched for automatic dependency analysis tools but could not find anything (which makes sense when considering JS is not statically typed, object oriented, or non-asynchronous), so we manually analyzed for source code dependencies. A new JS file was created to control asynchronous and lazy loading for split CSS and JS files. As a result, we saw a loading time of 1.56 secs. These results are provided below in Figure 13.

The optimization shows the dramatic performance improvements by splitting files. This separation of files might be a challenge for developers, but it is getting SEO benefits makes it worth it, "JavaScript benefits from being in small chunks to avoid locking up the main thread. Explore if you can reduce how much work is being done during execution" [35].

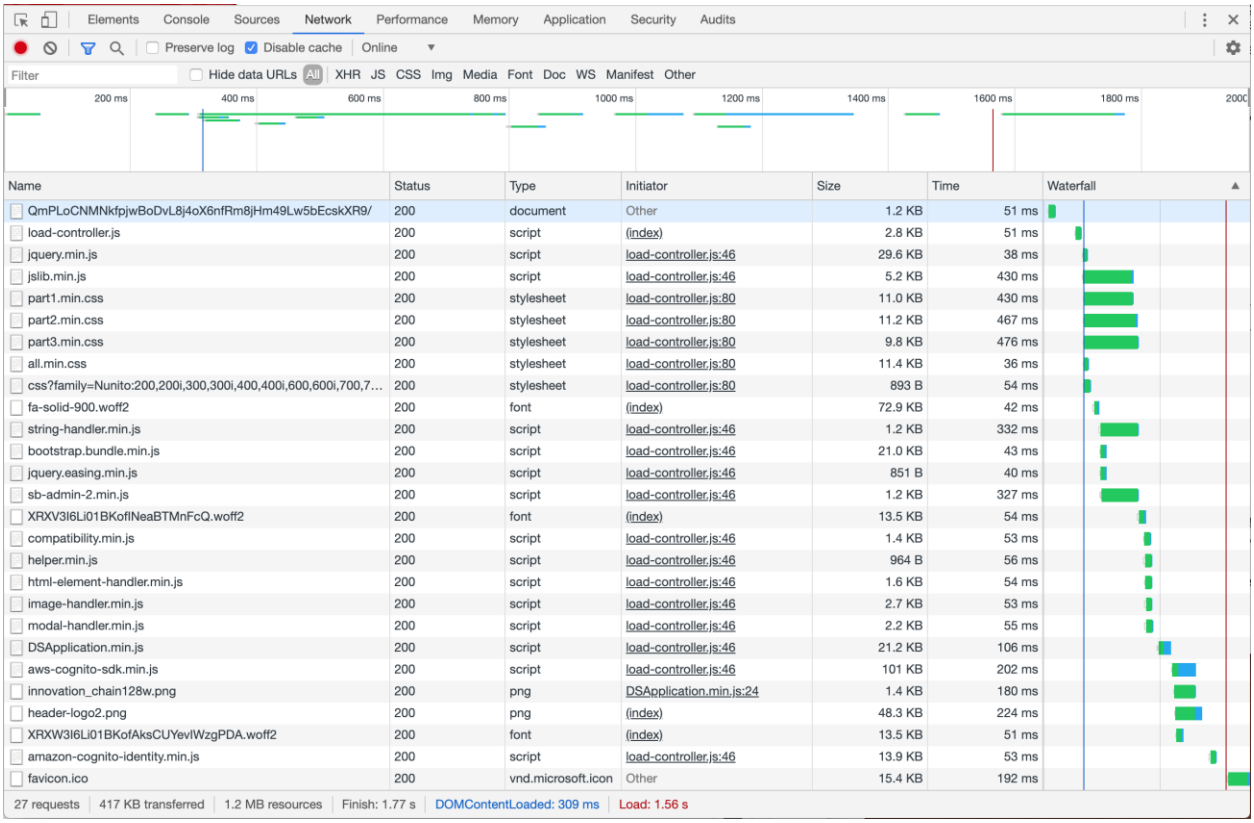


Figure 13: Network Tab After Optimization

Lighthouse

Google offers an open-source tool for developers to help improve the quality of web pages. It is available to run in Chrome DevTools Audits tab, CLI commands or as Node modules. Also, there is a web version called PageSpeed Insights (https://developers.google.com/speed/pagespeed/insights/), so we can quickly check the speed score by inputting any website URL.

We used Chrome DevTool Audits tab and conducted audits for Web Performance, Best practices, and SEO scores. We have uploaded the entire static directory to the IPFS network and calculated IPFS hashes for the websites before and after optimization. Since the IPFS gateway can serve as a web server, we can use this hash to retrieve the contents of the hashed page.

Audit Results Before Optimization (website at audit time)

On the first use of Lighthouse, the following scores were reported as: Performance 48, Best Practices 86, SEO 82. The more detailed results report is available here, part of which can be seen in Figure 14.

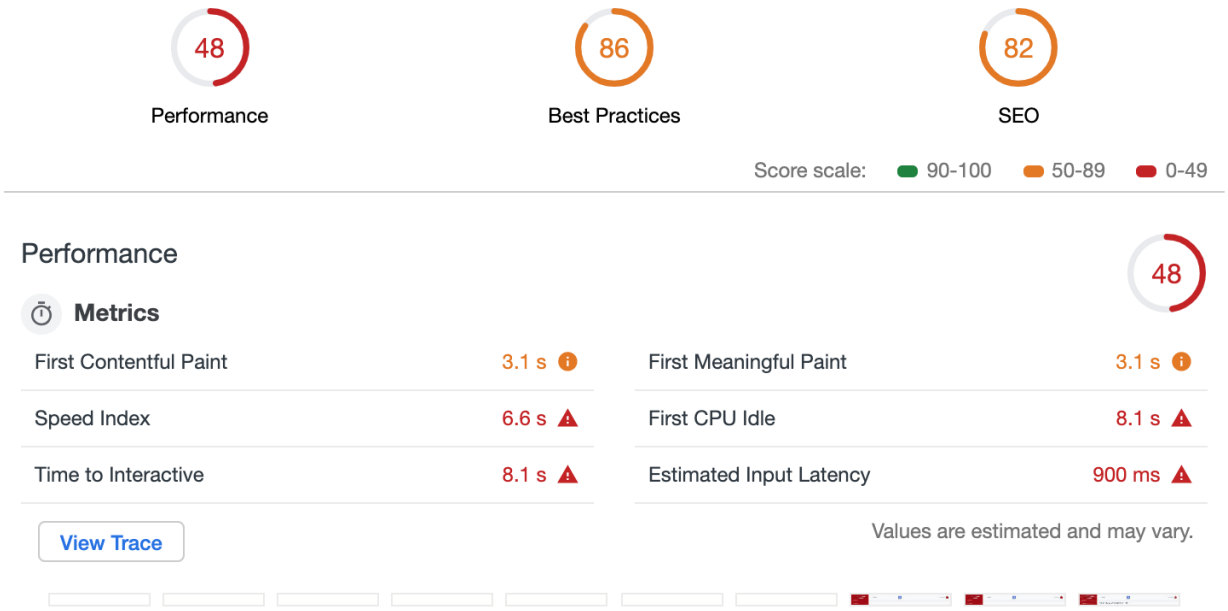


Figure 14: Lighthouse Audit Results Before Optimization

Audit Results After Optimization (website at audit time)

After performing optimization, the website received full marks: Performance 100, Best Practices 100, SEO 100 (results).

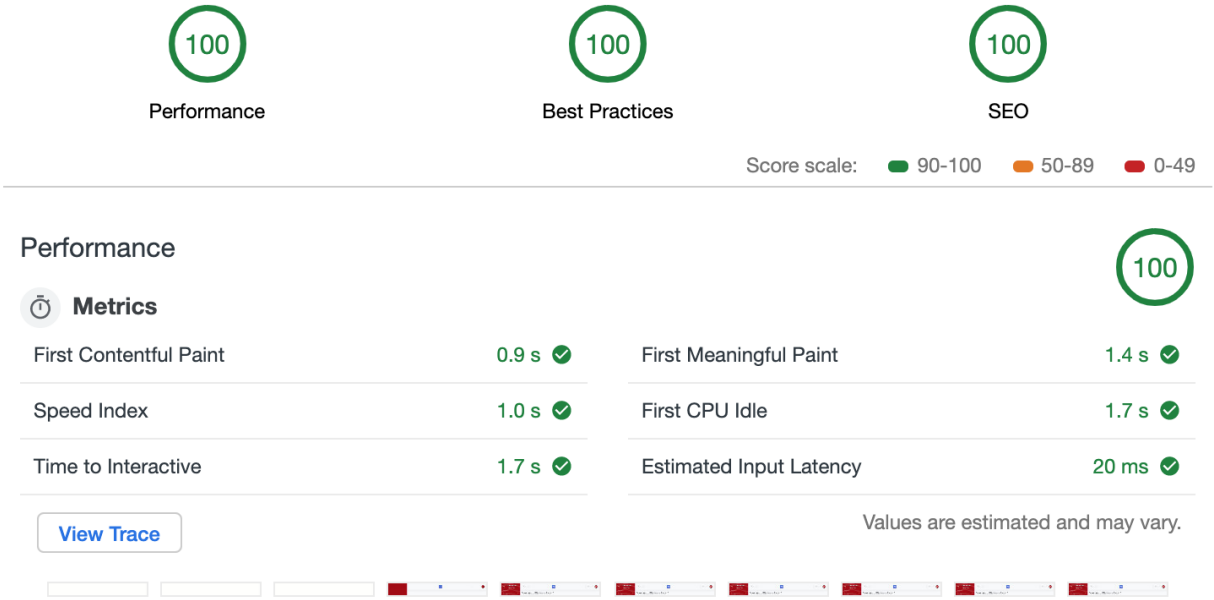


Figure 15: Lighthouse Audit Results After Optimization

Sustainable Web Hosting

An IPFS gateway can function as a web server. This works as a backup if Amazon stops its service in the future. A user can go to any IPFS gateway node or use CLI to retrieve contents (as such there is no single point of failure). Using IPFS, the content hash changes every time we update/change the web contents. To handle mutability, IPFS offers several ways to manage updates using the same domain name or same address such as DNS link or IPNS (Interplanetary Name Service) (https://docs.ipfs.io/guides/concepts/).

As shown above, the performance of the IPFS gateway as a web server is favorable as well. Using this approach, IPFS can act as a fallback if a record of the hash is maintained. Amazon S3, the primary web service, can use Amazon CloudFront to deliver web content. With this setup, we can update our site easily and host our static sites from fast CloudFront CDN edge locations.

Development Process

Meeting the requirements

By Milestone 2 the team had completed more than half the requirements defined at the beginning of the project. We had developed a site where a user may log in, create a disclosure while following a template matching the USPTO, save that disclosure as a draft or submit it to IPFS and the Ethereum blockchain. The focus for Milestone 3 was then to implement our own blockchain and provide various user features such as forgot password. Milestone 3 successfully implemented many of the remaining required features except for public/private key generation. The full list of requirements and their current status can be seen in the appendix. Most of the planned activities were completed.

To keep on schedule with the project, some features that were considered extraneous for the standard user were shifted to be considered requirements for a possible future enterprise release. This was discussed with, and agreed upon by, the customer. By the completion of M3, the team had accomplished the majority of what was intended at the beginning of the project.

The use of a Kanban board and story point system has been integral to the success of this team. The board, as it is intended, clearly displays what must be done to each member of the group, this allows everyone always in agreement about what and how much needed to be done. It was also found, that usage of a Kanban board allowed team members to play to their strengths, as everyone could choose their own tasks.

Story points were used to estimate the time commitment and difficulty of tasks. When a new task was created it was assigned story points based off an estimate of the difficulty and time commitment to complete the task. When the group was present for the creation of a task, a consensus would be reached on the amount of story points. If the task was created by an individual the amount of points would be estimated by the individual, then at the next group meeting a consensus would be reached with the present group members.

When team members were selecting tasks, they worked to a consistent paradigm. Unblocked high-priority low-point tasks are generally completed first while high-point low-priority tasks that are blocked by high-priority low-point tasks are completed last. When there is a tie, a teammate favors unblocked over blocked, high priority over low priority, low point over high point, blocking over non-blocking, and a

task closer to their expertise over a new experience. If all else fails, there were always capstone deliverables to support.

Additionally, collaboration and meetings were key to keeping on top of the project. The team met Sundays, Thursday, and alternating Tuesdays. Most meetings began with a review of the Kanban followed by a working session. These meetings gave the team time to touch base and keep on top of what should be done and when. Working collaboratively was a strong aspect of the meeting; when someone needed help, we shared content using Zoom screen share. For a few weekends during the semester, mostly before large submissions, the team would keep a Zoom channel open throughout the weekend and members would be present while working so any problems could be quickly resolved through teamwork. These weekends resulted in significant gains in completing tasks.

The team used Azure DevOps for project management and for version control. It is free for open source, cloud hosted projects, is fully integrating with the Visual Studio IDE (which we used for development) and is a single application for our repo and task board. It provides support for HTML, JS, C#, and provides integration with AWS. We also utilized Azure plugins to visualize linked tasks and to export tasks in csv format to report statistics on task progress. View this project's Azure DevOps site at https://dev.azure.com/cscie599/SWE%20Invention%20Blockchain.

These methods, strategies, and tools kept the team on track and focused throughout the development process. If we to do this project again, little would be changed. Most of the struggle that was experienced in this project resulted from outside the group, such as encountering incorrect AWS manuals. But some of the internal issues did result from some slight overestimates of what could be accomplished during one semester. One area where a potential improvement in work style could be realized for this team was meeting planning. Many times, meetings were planned with very short notice because no one would commit to a proposed time day in the future. Fortunately, the team was very lucky with this method and most times almost all members were present. However, there were also many times that members missed meetings due to not seeing that they were scheduled. As the semester progressed, we found a good cadence. An improvement would be to establish this cadence earlier.

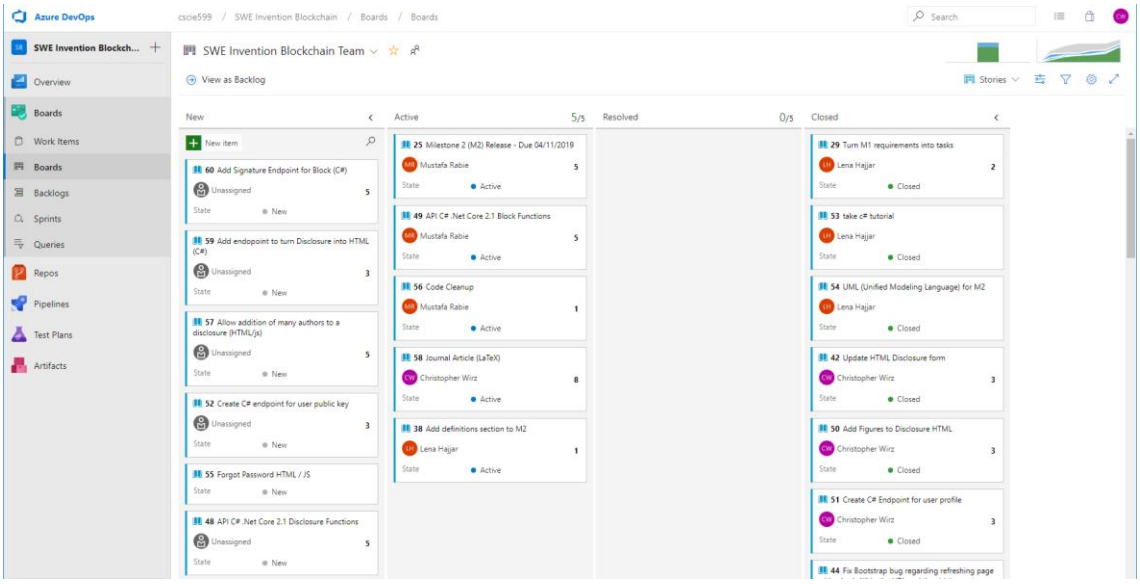


Figure 18: Our Azure DevOps Kanban Board

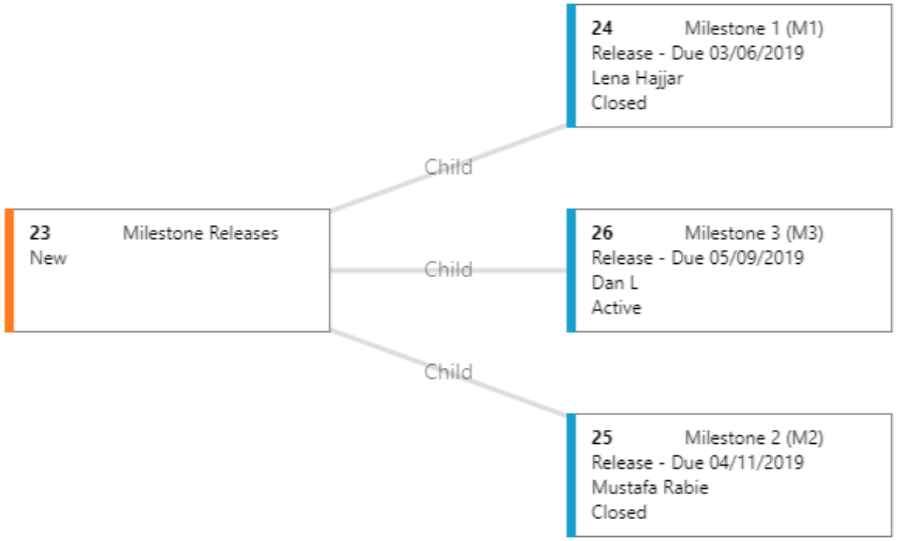


Figure 19. An example of linked tasks

Estimates

The story point system proved to be an effective method of estimating the amount of work of various tasks. The continued use of the system allowed for the team to come to a consensus about the amount of work of which a story point entail. It was found that the system was most effective for smaller tasks. When any single task was given a high amount of story points, it was almost always turned out to be an underestimate. An example of this involved writing the journal paper and creating the

mobile app. Both tasks took much more effort and time than expected (> 13pts). Fortunately, most tasks could be broken up into smaller tasks with fewer story points.

At the beginning of the project, we believed that after M3 the group would be working on stretch goals; we never got to start the stretch goals. Additionally, there was some renegotiation with the customer to move some of the initial requirements to be part of a possible enterprise version of this system. None of these were key features to the public system. The scope of the mobile app was modified to use HTML previews of disclosures as hosted by the DisclosureSys.com website. At M2 the team had accomplished a lot and had a much better idea of the group's rate of work and the size of the project. This gave the group a much better idea of what could accomplished this semester.

The figure below shows the team's velocity chart. The idea behind using story points is that each story point should take equal time and complexity to complete. The top of the green area, representing closed tasks, should be nearly linear if a sprint is perfectly planned and executed. The green section is clearly not linear, but the areas in which it lacks linearity correlate with capstone deliverables. This means that the team is better at estimating story points for coding-related activities versus documentation, research, and presentation. It is clear from the chart where milestones occurred as there are sharp increases of completions of story points at two points in the chart. This is as there was some rush to get things done in time for submissions. The semi-flat section before M2 is partly due to relying on incorrect documentation from AWS; Amazon's published examples in their documentation do not run correctly. As time progressed, we settled for workarounds and cycled through third party libraries - but eventually decided to just garbage collect the remnants of the bug periodically.

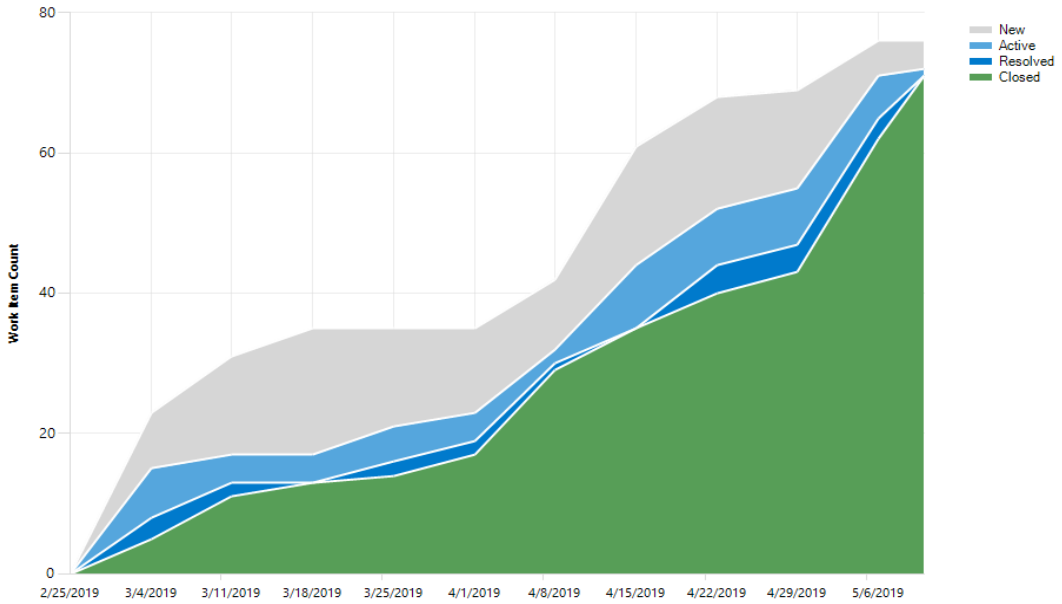


Figure 20: Our Team's Velocity Board

Since the story point system was used instead of actual time estimate or a weekly scrum system, it is difficult to show how our estimates differed from the how difficult a task was. However, the team did have general ideas of how far we would be at various points throughout the semester and how these lined up with reality can be seen in Table 2.

Submission	Estimate Destination	Actual Destination
Milestone 1	Planned to have a solid understanding of what we wanted to accomplish and how to do it.	Were at our planned position.
Milestone 2	Planned to be nearing completion of all the requirements, about 90% done.	Made significant progress towards completing the requirements, but were closer to 70% done, also the requirements were adjusted.
Milestone 3	We thought that we would have all the requirements finished and would be working on / stretch goals.	A high percentage of the requirements have been completed and the scope was maintained from Milestone 2.

Table 2: Estimates vs. Reality

At this point the rest of the project, including requirements moved to the enterprise version. Currently, the mobile app supports preview of user's disclosure drafts, published disclosures and search all published disclosures. The team decided

that creating, editing and publishing disclosures from the mobile app will be added after the web client has been tested in a focus group.

Risks

An unavoidable risk is the service being spammed with false submissions. To counteract this, we are using Amazon's API gateway to rate limit submissions as described in Requirement 2.1.

The Disclosure System takes advantage of modern browser local and session storage capabilities in operating as a single page app. This will improve load times and mobile compatibility. The risk is browser compatibility; not all browsers support certain paradigms regarding style and syntax. Currently, our service only officially supports, Chrome web browser. Tests concluded that Safari and Firefox delivers a similar experience to Chrome, while MS Edge fails to interpret JS correctly.

The native mobile application is written using the Xamarin framework, which may not be future proof with Apple's upcoming iOS changes. We will adapt with the necessary changes to support both iOS and Android SDK feature updates.

We will not be able to store disclosures / blocks on other mainstream blockchains due to block size limitations and network cost. Instead, we will store the computed hashes of our blockchain such that we can preserve distributed immutability to the highest standard of current blockchain technology. We assume storing a hash of a submission on a timestamped blockchain should be adequate proof that some type of disclosure has been made.

Additionally, we are implementing a feature which allows users to store drafts of their submissions on our servers. This feature allows users to prepare a submission over multiple sessions. However, it comes with some additional risks. If anyone was able to gain access to an unpublished submission, either through our system or a user's credentials, it would allow them to either illegally use an idea that is not yet protected in anyway or force a public disclosure that the inventor did not want. To safeguard against this, we will inform users that protecting their account credentials is their responsibility and require users to use secure passwords. We believe we can provide adequate security measures through the API.

Disclosure drafts are saved on the public facing S3 bucket, which can lead to unauthorized access, or crawling by web crawlers. This is a very low risk situation, since all draft names in a non-human readable format, that is randomly generated. As an

additional precaution in the future we will be saving the drafts to restricted S3 buckets and authorize clients through client key/secret pair.

Team Dynamic

As mentioned previously, the team has been using a pure Kanban approach to this project. This allows team members to play to their strengths as they contribute to the project. We have found that our Azure board has been a useful way to keep track of tasks. We use Zoom for meetings among team members, teaching instructors, and our customer. We also use Zoom to host working sessions, so multiple team members can work together and share ideas or gather feedback as needed. In between meetings, we use Slack to communicate. We have found that it's a good way to share links, ideas, and it provides an easy way to ping each-other when we have questions or need to grab someone's attention.

The team dynamic was consistent from the start of the project to the end of the project. This is as it worked well from the start and nothing needed to be changed. But as the work of the project became more intensive and difficult throughout the project, working together in a zoom session did become more frequent. As previously mentioned, the biggest flaw with our team dynamic was a lack of giving decent notice when planning meetings. Generally, this was not much of a problem as the team was very good about attending meetings on short notice. But as three members are on the east coast, two on the west, and one on the other side of the world in Asia, sometimes things would be missed. This caused, on only a few occasions, for a team member to do some work that was decided to be unnecessary by the group previously. But in general, the harm of members missing meetings was mitigated by the fact that meetings were mostly recorded, and that the Kanban board was reviewed daily. For the most part, since the group was very lucky and generally able to schedule meetings with high attendance with very short notice (including meetings with the customer), no real changes were made in our practices.

Conclusion

DisclosureSys.com is currently live with https security, Content Delivery Network (CDN) distribution, and mirrors on both S3 and IPFS. It can elastically scale as more users visit the site but will run on minimal resources otherwise. It also runs on Android and iOS and will be deployed to the respective Play and App stores soon. Users can create accounts, disclose inventions, search the blockchain for other disclosures, and collaborate on disclosure drafts as co-authors. All Search Engine Optimization best practices have been implemented and the team has successfully created sample disclosures that have been persisted as blocks - both in S3 and on the IPFS network. Most importantly, most users would never know the application was built on a blockchain if it wasn't advertised. This successful implementation of an application based on blockchain technology has improved the state of the art for invention disclosure systems - and deliberately follows the intention of current intellectual property law. While ownership of intellectual property is a constitutional right in the United States, DisclosureSys.com has taken the necessary steps to allow a more universal assertion of that right by providing free, immutable and searchable defensive publication.

References

31. Benet, Juan. “IPFS - content addressed, version, P2P file system.” Technical report, July 2014. arXiv: 1407.3561

32. Camacho, Jennifer. “Biotech Innovation in a First-to-File World.” Nature Biotechnology, 7 May 2012 www-nature-com.ezp-prod1.hul.harvard.edu/articles/nbt.2204.

33. “How Blockchain Will Accelerate Business Performance and Power the Smart Economy.” Harvard Business Review, Microsoft, 22 Nov. 2017, hbr.org/sponsored/2017/10/how-blockchain-will-accelerate-business-performance-and-power-the-smart-economy.

34. “H.R. 1249 — 112th Congress: Leahy-Smith America Invents Act.” www.GovTrack.us. 2011. https://www.govtrack.us/congress/bills/112/hr1249

35. Osmani, Addy. “JavaScript Start-up Optimization | Web Fundamentals | Google Developers.” *Google*, Google, 1 May 2019, developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/javascript-startup-optimization/#network.

36. Tudury, Leila. “What Does Blockchain Mean?” Dictionary.com, Dictionary.com, 21 Aug. 2018, www.dictionary.com/e/tech-science/blockchain/.

37. Wood, Gavin. “Ethereum: A secure decentralized generalized transaction ledger.” Ethereum project yellow paper, 2014. Web. 15 February 2019. http://gavwood.com/paper.pdf

APPENDIX - A

Requirements List

We have re-reviewed the requirements with the customer and have annotated the completeness or deferment of each requirement. The following table was updated on 5/7/2019 at the conclusion of the final sprint.

As a note, a goal of the requirements is to determine make-buy analysis in many cases. If a requirement can be completed through the purchase of a service, the acceptance criteria is dispositioned to the third-party service provider.

 = Complete  = Not Done  = Deferred

* NF = Non-Functional requirement

1	**Description:** The system shall allow users to create invention disclosures **Completion Points:** N/A **Acceptance Criteria:** Acceptance of all sub-requirements and derived requirements.
1.1 	**Description:** An invention disclosure shall have the following sections: Title Authors - with contact information Abstract Keywords Figures Background Field Description of related/prior art Disclosure/patent cross-references Summary Brief description of drawings Detailed description of the invention Claims Dependent claims **Completion Points:** 3 **Acceptance Criteria:** When viewing an invention disclosure block, the Invention Disclosure Body contains all the aforementioned fields.
1.2	**Description:** The system shall allow for the following user functions: Account creation, Account update, Login, Forgot Password, Key Pair

	Generation, Soft Deletion. **Completion Points:** N/A **Acceptance Criteria:** Acceptance of all sub-requirements and derived requirements
1.2.1 	**Title:** Account creation **Completion Points:** 3 **Acceptance Criteria:** A user, new to the invention disclosure system, can create an account with an email address and password (repeated for verification).
1.2.2 	**Title:** Email verification **Completion Points:** 3 **Pre-Condition:** The account has been created (1.2.1) **Acceptance Criteria:** When a user clicks “Create Account” an email is sent with a code to verify the email account. Meanwhile, on the main UI, a modal request the code for confirmation. When the user enters the code in the modal, the account enters a confirmed state in the user database.
1.2.3 	**Title:** Key pair generation **Completion Points:** 5 **Pre-Condition:** The account has been verified (1.2.2) **Acceptance Criteria:** After the account is verified, the user may create a public and private key. The private key is displayed once, and the user must copy it to a safe location. Or a user may use Metamask to create/import a key pair. NF: The key pair should be generated on the client side
1.2.4 	**Title:** Account update **Completion Points:** 3 **Pre-Condition:** Account creation (1.2.1), Email Verification (1.2.2), Login (1.2.5) **Acceptance Criteria:** When a user updates an email address, they receive an email to the new AND old email address to verify it (see 1.2.2). If they update their password, there will be a verification email sent to their email. Other things that can be updated: profile picture, personal website, and/or biography (no verification needed because they already logged in)
1.2.5 	**Title:** Login **Completion Points:** 3 **Pre-Condition:** Account creation (1.2.1) **Acceptance Criteria:** When a user revisits the site, he or she may use the previously defined password to create a session with the application.

1.2.6 	**Title:** Forgot password **Completion Points**: 3 **Pre-Condition:** Account Creation (1.2.1), Email Verification (1.2.2.) **Acceptance Criteria:** If a user fails to enter the correct password *3* times or requests new password, email verification (1.2.2) is sent to the listed email for them to verify that they requested a change. Their account will be updated with a new password. **Notes:** The theory is that stealing an account is meaningless because the only thing you can do is create new disclosures - so you'd be doing work for them for free.
1.2.7 	**Title:** Key pair regeneration **Completion Points:** 5 **Pre-Condition:** Prior Key-Pair generation (1.2.3) **Acceptance Criteria:** See Key-Pair generation (1.2.3) The user may create a public and private key. The private key is displayed once, and the user must copy it to a safe location. A user may also use Metamask to create/import a key pair. NF: The key pair should be generated on the client side
1.2.8 	**Title:** Soft-Delete account (Deferred for enterprise versions) **Completion Points:** 3 **Pre-Condition:** Login (1.2.5) **Acceptance Criteria:** If a user goes into the profile/update screen and clicks on “Delete my Account” then an acknowledgement modal pops up - which the user accepts. The account is then given a “soft delete” flag - such that authorship on any published disclosures are <redacted>. But this account will still be recoverable with the previously linked email address.
1.3 	**Description:** The system shall allow for draft (pre-submission) invention disclosures. NF: The draft should be encrypted (whether stored in the browser or in S3). **Completion Points:** 5 **Pre-Condition:** The user must be logged in (1.2.5) **Acceptance Criteria:** When a user returns to the application, previously created drafts are available for editing and completion.
1.4 	**Description:** The system shall persist completed submissions on its block chain. **Completion Points:** N/A

	Pre-Condition: Draft is complete (1.3) **Acceptance Criteria:** When a user hits “Publicly Disclose” on a draft the disclosure hash is calculated. This hash is then added to a block and the block hash is returned to the user in a modal with the note “Saved.” When the success modal pops up, there is a link to go see the block.
1.4.1 	**Description:** The disclosure documents shall persist in static file storage. **Completion Points:** 3 **Acceptance Criteria:** This is accomplished using a third-party static file storage such as IPFS, which provides decentralized file storage with unlimited storage size. The data will be publicly accessible, retrievable by its content hash, and unforgeable.
1.4.2 	**Description:** A transaction of a block shall contain the following fields: Prior Block hash Author(s) public key(s) [optional] Encrypted one-time private key Prior Block hash signed by author(s)’ private key Timestamp Invention disclosure body **Completion Points:** 1 **Pre-Condition:** Static storage (1.4.1) **Acceptance Criteria:** Exploring a block (visiting the static file link) shows all the requisite fields.
1.5 	**Description:** The system shall allow for private disclosure. (Deferred for enterprise) **Completion points:** N/A **Acceptance Criteria:** When a user clicks “advanced features” before submitting, then one of the options (probably a checkbox) is present to make a private disclosure. When the option is selected a key pair is presented to the user. The user can enter his or her own if desired (since the system doesn’t know the user’s key pair).
1.5.1 	**Description:** The system shall allow the invention disclosure block to be encrypted by the optional encrypted one-time private key. (Deferred for enterprise) **Completion points:** N/A

	Acceptance Criteria: When the user selects “private disclosure” from the advanced options, the main submit button changes to “Privately Disclose”. When the success modal pops up there is a link to go see the block. When the user clicks the block, it is then unreadable without the private key.
1.5.1.1 ✗	**Title**: Private Key Creation **Completion points:** 2 **Acceptance Criteria:** Users may request a private key. The private key is displayed once, and the user must keep it in a safe location. NF: The private key should be generated on the client side.
1.5.1.2 ✗	**Title:** Submit private disclosure **Completion points:** 3 **Acceptance Criteria:** Upon submission, if the user indicates they would like a private disclosure, a prompt will come up to enter a private key. Once the private key is entered, the block is encrypted using that key before publishing to the blockchain.
1.5.2 ✗	**Description:** Private disclosures have the added field to identify technology gap and lack of TRL (Technical Readiness Level). **Completion Points:** 3 **Acceptance Criteria:** In the advanced information options, which are displayed optionally by default, and include the ability for a user to make a disclosure private, an author can provide information regarding dependent technologies and technical readiness for their invention.
2	**Description:** The system shall have an API (Application Programming interface). **Completion Points:** N/A **Acceptance Criteria:** When the user goes to <server>/api.json, they are provided with the full API documentation in OpenAPI 3.0.2 format.
2.1 ✓	**Description:** The API shall rate limit submission. **Completion Points:** 1 **Acceptance Criteria:** *This is accomplished using a third-party API Gateway provider*
2.2	**Description:** API endpoints must be authenticated.

☑	**Completion Points:** 1 **Acceptance Criteria:** *This is accomplished using a third-party API Gateway provider*
3 ☑	**Description:** The system shall Search Engine Optimize (SEO) public disclosures. **Completion Points:** N/A **Acceptance Criteria:** When a user searches a major search engine (like Google) for subject matter contained in the body of a disclosure, a link to the disclosure is presented within the results.
3.1 ☑	**Description:** Individual disclosures shall have fast load times (NF). **Completion Points:** N/A **Acceptance Criteria:** A HTML disclosure should load in under 0.6s.
3.1.1 ☑	**Description:** Individual disclosures must be loaded statically (faster than from a database). **Completion Points:** 1 **Acceptance Criteria**: *This is accomplished using a third-party static hosting provider*

3.2 ☑	**Description:** The presented HTML must contain metadata compatible with Google, Facebook, Bing
	`<title></title>`
	`<meta name="description" content="" />`
	`<meta name="robots" content="index, follow" />`
	`<meta itemprop="name" content="" />`
	`<meta itemprop="description" content="" />`
	`<meta name="twitter:card" content="summary" />`
	`<meta name="twitter:title" content="" />`
	`<meta name="twitter:description" content="" />`
	`<meta property="og:title" content="" />`
	`<meta property="og:type" content="article" />`
	`<meta property="og:published_time" content="" />`
	`<meta property="og:description" content="" />`
	`<meta property="article:tag" content="keywords" />`
	Completion Points: 3
	Acceptance Criteria:
	When viewing a stand-alone block html, the source code has the above-listed fields directly below the opening <html> tag.

3.2.1 ✅	**Description:** The system shall allow creators to upload digital images and sketches. Images are going to be 64-bit encoded. **Completion Points**: 3 **Pre-Condition**: Draft or Disclosure must be created **Acceptance Criteria:** Blocks contain base64 encoded images in the <Figures> portion of the HTML body.
4 ✅	**Description:** The system shall allow users to search invention disclosures. NF: Results can be bracketed by date. NF: The disclosure shall be stored in a materialized database (for search). **Completion Points:** 8 **Acceptance Criteria:** A user types various search phrases into the search bar, hits enter or clicks search, and results are displayed matching the entered criteria. NF: The search bar is at the top of the UI
4.1 ✅	**Description:** The system shall have a full text search. (Defer for Enterprise) **Completion Points:** 3 **Pre-Condition:** The system can search invention disclosures (4). **Acceptance Criteria:** A user types various search phrases into the search bar, selects full-text search, hits enter or clicks search, and results are displayed matching the entered criteria only related to the disclosure body (abstract, background, description, summary, claims).
4.2 ✅	**Description:** The system shall have a key word search. **Completion Points:** 3 **Pre-Condition:** The system can search invention disclosures (4). **Acceptance Criteria:** A user types various search phrases into the search bar, selects keyword search, hits enter or clicks search, and results are displayed matching the entered criteria only related to the disclosure keywords.
4.3 ❌	**Description:** The system shall have a claims-only search. **Completion Points:** 3 **Pre-Condition:** The system can search invention disclosures (4). **Acceptance Criteria:**

	A user types various search phrases into the search bar, selects claims search, hits enter or clicks search, and results are displayed matching the entered criteria only related to the disclosure claims.
4.4 	**Description:** The system shall have search by author. **Completion Points:** 3 **Pre-Condition:** The system can search invention disclosures (4). **Acceptance Criteria:** A user types various search phrases into the search bar, selects author search, hits enter or clicks search, and results are displayed that were entered by an author matching the name.
5 	**Description:** (*Stretch goal)* The system shall provide comments on disclosures. **Completion Points:** 8 **Pre-Condition:** Requirements 1-2 **Acceptance Criteria:** 1. A user highlights a section of the disclosure text and a comment button appears. The user enters a comment and hits “save”. 2. The user may edit a comment he or she has created. 3. The author may elect to moderate/remove comments. 4. When a user visits a disclosure, commented areas are annotated. The user can elect to expand or collapse comments.
6 	**Description:** (*Stretch goal)* Art-to-patent sponsorship (and royalties) **Completion Points:** 21 **Pre-Condition:** Requirements 1-2 **Acceptance Criteria:** 1. A logged-in user can enable “open for opportunities” on their disclosure. 2. Within the displayed invention disclosure, a button in the authorship block stating “contact for opportunities” is clickable for visiting users. 3. When the button is clicked, a message composer box appears where the user can type a greeting and select any or some of the following options: Licensing deal, sponsor patent, purchase disclosure rights, etc. 4. Clicking submit will send the message / offer to all authors

APPENDIX - B

Definitions

Actual: A method of reduction to practice in which a working model of the invention is made.

Alice: A 2014 ruling (Alice Corp. v. CLS Bank International) clarifying that a patent is invalid if it merely changes how a process is executed, without changing the system. For example, automating a process using a computer, to make it run faster, does not produce a product that can receive patent protection.

Blockchain: A decentralized database system that records transactions in secure blocks [36].

Constructive: A method of reduction to practice in which a public description is made such that someone of ordinary skill could produce the invention

Ethereum: An implementation of blockchain that utilizes smart contracts to alter and keep track of an internal state of each block [37].

First-To-File: The method used by the current US Patent system. Patent protection is granted to the first inventor that constructively or reduces the idea to practice and files it to the US Patent office [34].

First-To-Invent: The method previously used by the US Patent system. Patent protection was granted to the first inventor who claims ownership of the idea, whether or not it had been reduced to practice [32].

Interplanetary File System (IPFS): One of the most prominent distributed storage services. Content is stored on the distributed network and can be retrieved by its content address [31].

Interplanetary File System Gateway (IPFS Gateway): A node on the IPFS network that allows access to IPFS content through traditional http clients.

Leahy-Smith American Invents Act (AIA): The U.S. Federal Statute that changed the US Patent system from First-to-Invent to First-to-File. This statue was passed in 2011 and went into effect in March 2013. [32] [34]

Non-practicing entity (NPE), often called a patent troll, is someone who holds a patent but has no intentions of developing it or offering it publicly.

Public Disclosure: Any non-confidential disclosure of an invention - often the invention has been reduced to practice.

Reduction to Practice: Demonstrating that an invention is possible to create and achieves its intended purpose.

APPENDIX - C

This section contains manuals describing the setup, installation and usage of the system for different actors.

User Manual

When you first visit DisclosureSys website you will be presented with the welcome page. If it's your first time visiting the site, you will be required to register for an account to be able to use the amazing features that DisclosureSys offers.

DisclosureSys.com

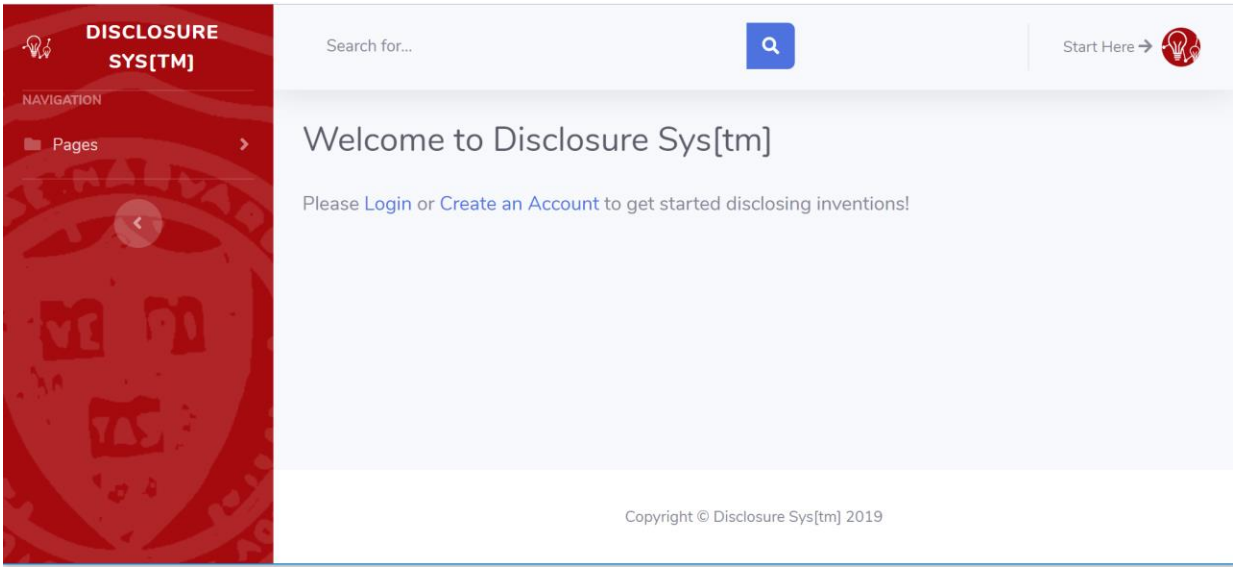


Figure C-1: DisclosureSys Homepage

Registering for a new account

1. From the home page or the top right menu, choose “Create Account”

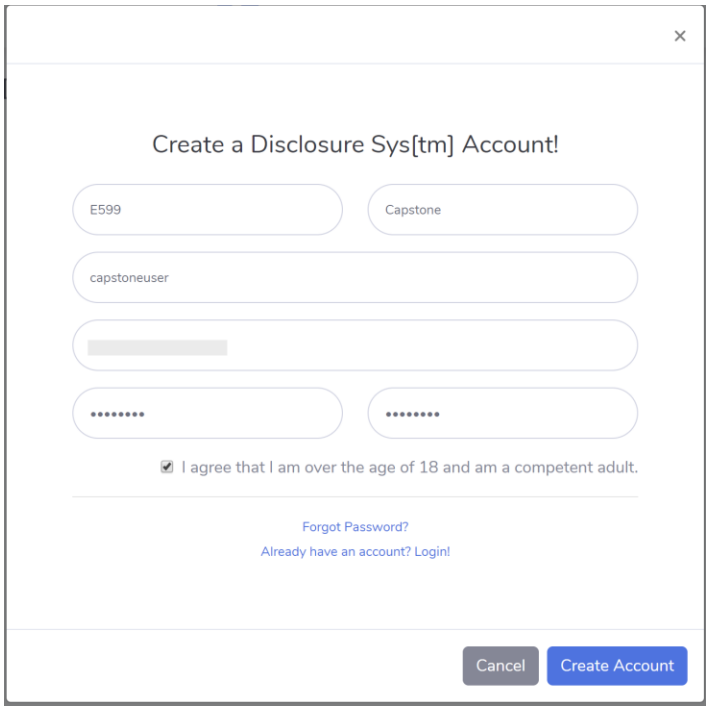


Figure C-2: Create account modal

2. Complete the registration form
3. Make sure you enter a valid email
4. Read and then indicate that you agree to the terms and conditions
5. Click “Create Account”
6. You will receive an email with the account verification code
7. Enter the verification code

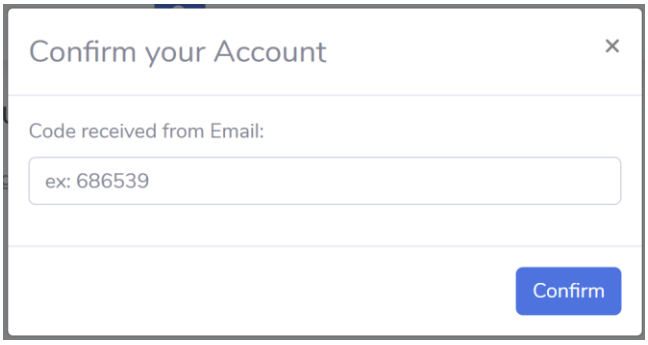


Figure C-3: Confirm account modal

8. Click “Confirm”
9. Now you are ready to create disclosures for your amazing ideas.

Login

1. Click “Login”

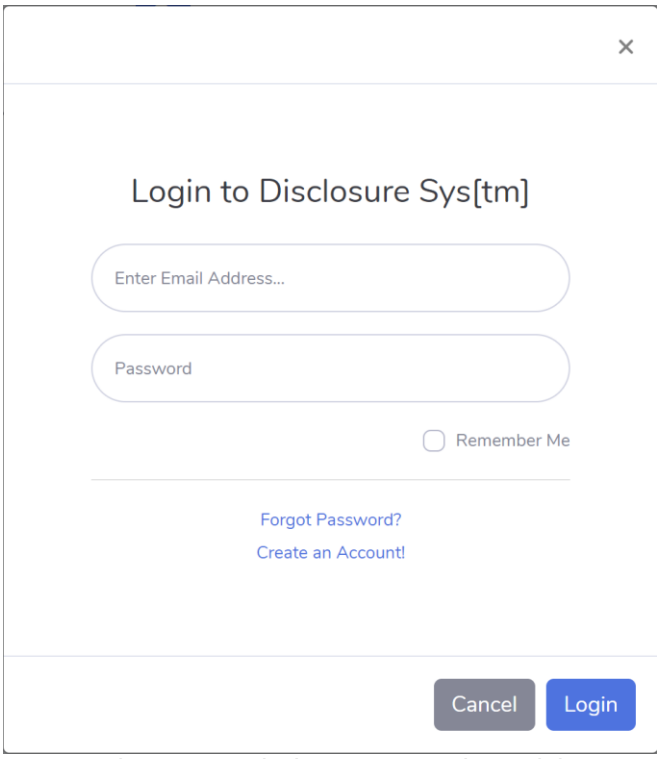


Figure C-4: DisclosureSys Login Modal

2. Enter your “username” or “email address”
3. Enter your password
4. Click “Login”

Creating a Disclosure

Once you have successfully logged in you are ready to create disclosures for your ideas.

1. From the Home Page click “Create new Disclosure”

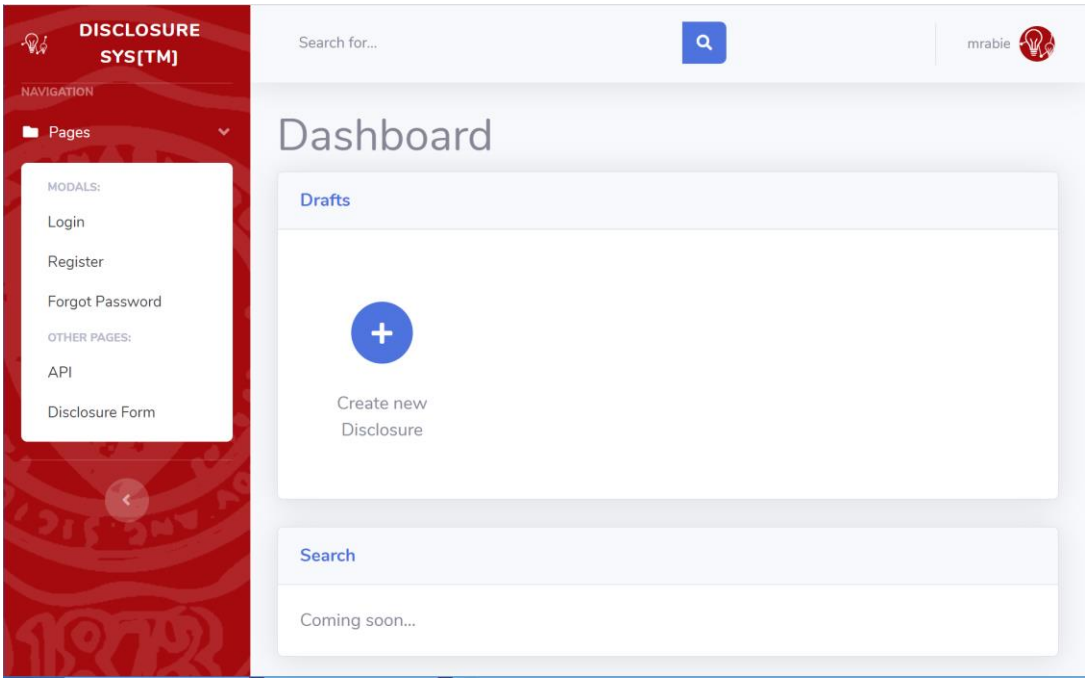


Figure C-5: DisclosureSys Dashboard

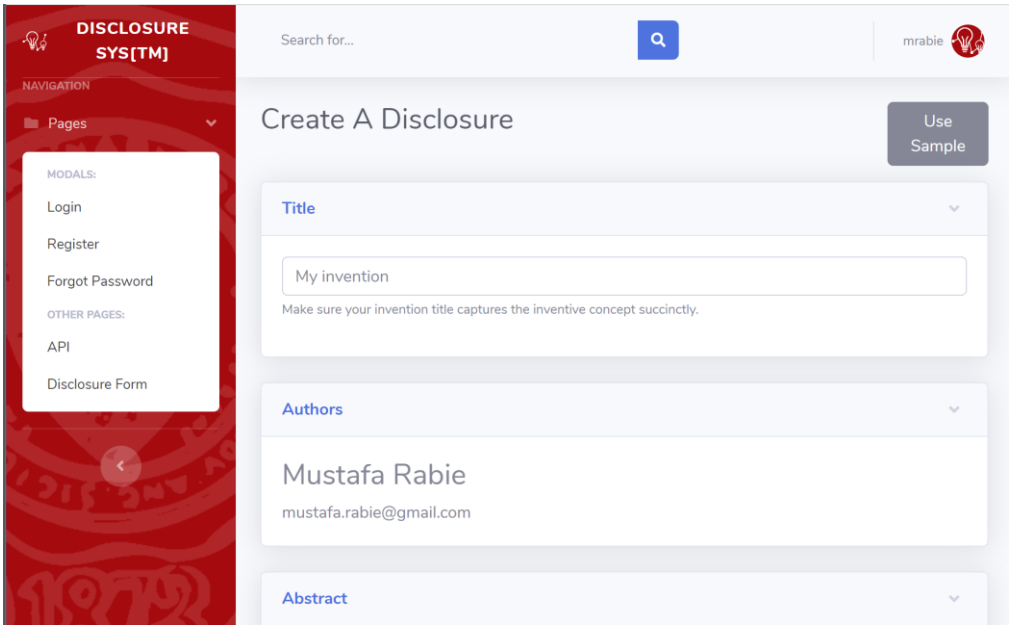


Figure C-6: Disclosure creation page

2. Fill out the details of your disclosure.
3. You can add figures, back of the envelopes, sketches, and diagrams.

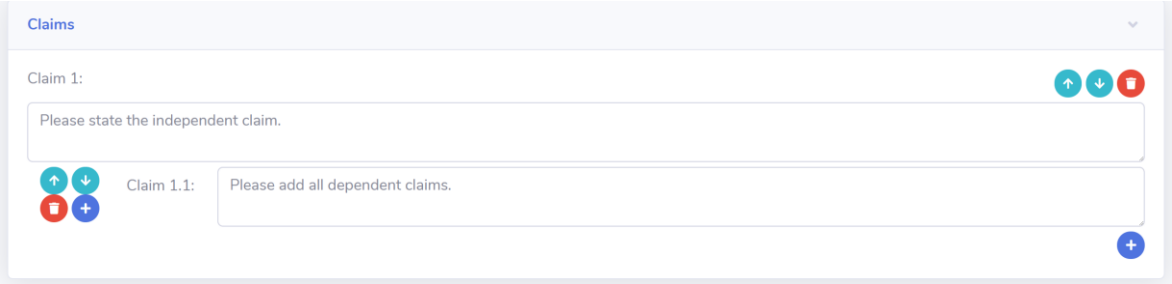


Figure C-7: Adding claims

4. You can add claims to your disclosure and order them by clicking the up/down arrows.

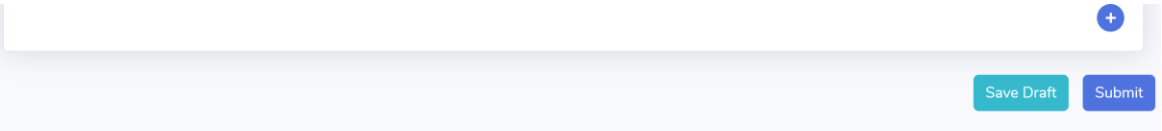


Figure C-8: Submitting a disclosure

5. You can keep working and saving your disclosure as a draft, by clicking the “Save Draft” button.
6. When your disclosure is completed and ready, you can write the disclosure to the blockchain by clicking the “Submit” button.

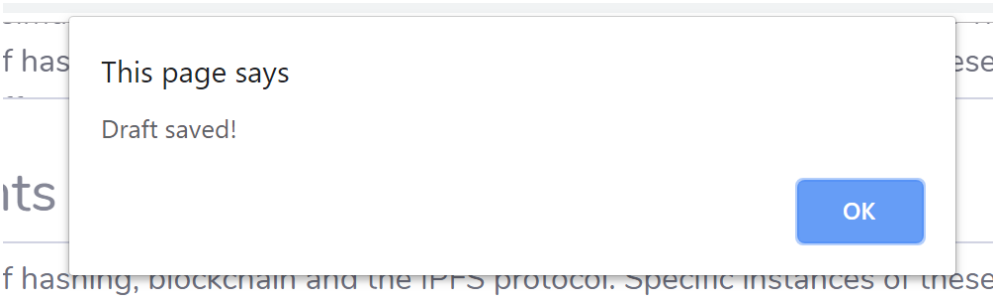


Figure C-9: Disclosure Draft Saved

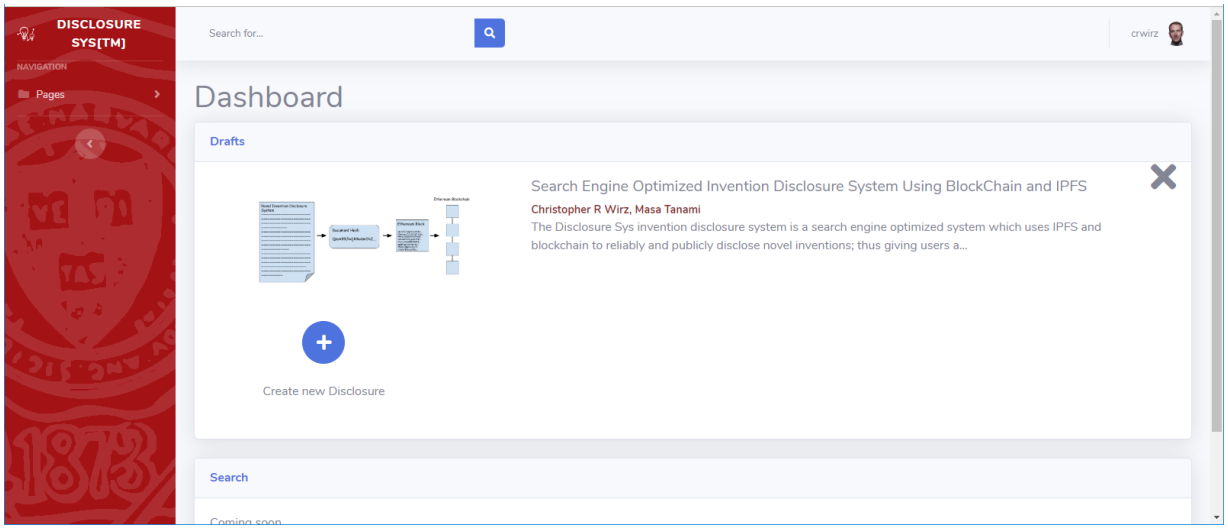


Figure C-10: Dashboard showing Saved Draft

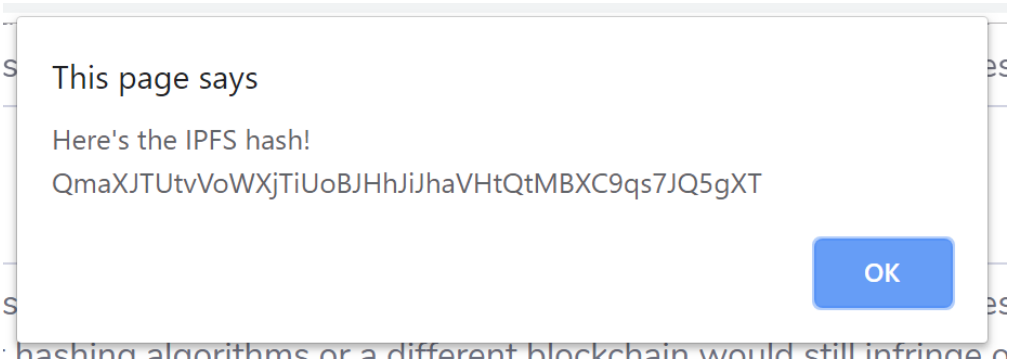


Figure C-11: Disclosure Saved to IPFS, Hashed and Hash written to Blockchain

DisclosureSys Mobile (DSMobile)

After downloading and installing the DisclosureSys App for iOS or Android from their respective app stores, you can find the application from your device's launch menu. The app can be easily identified by the icon or the name "DSMobile".

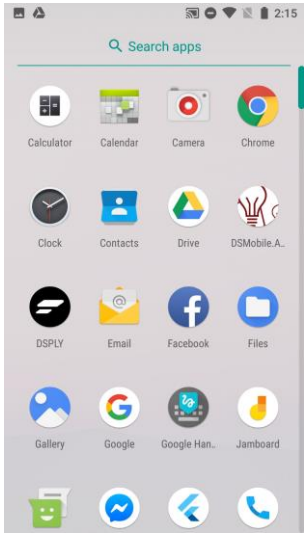


Figure C-12: DSMobile Icon

Click on the icon to launch the app.

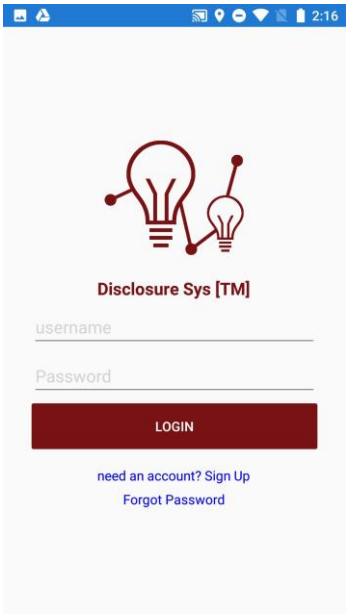


Figure C-13: Login Screen

You will be greeted by the login screen. Enter your account credentials to start.
If you are not currently registered with DisclosureSys, you can create an account.

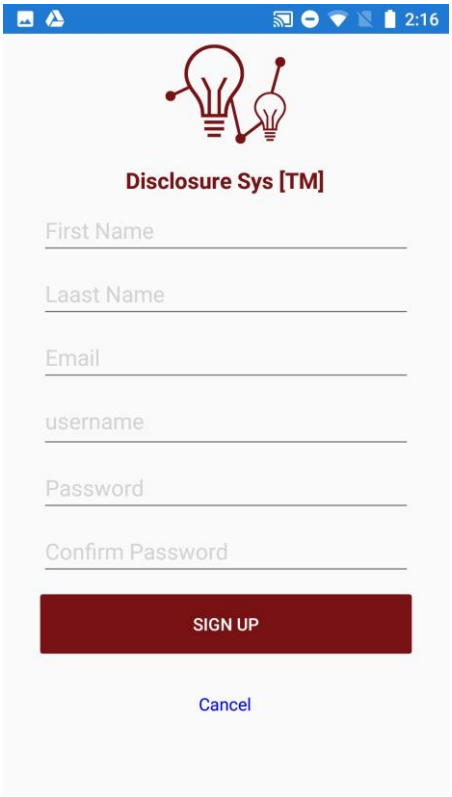


Figure C-14: Sign Up Screen

Enter your information details and click submit. Please enter a valid email address, since on successful sign up you will receive a confirmation email.

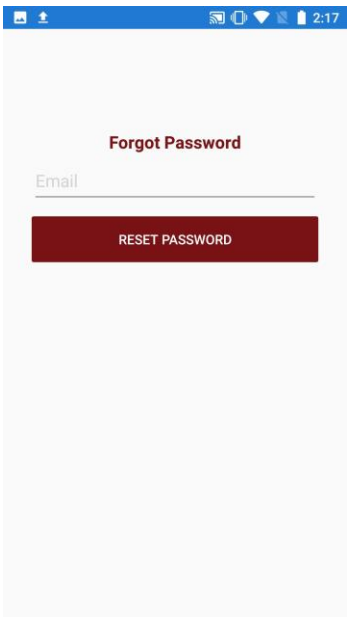


Figure C-15: Forgot Password Screen

Login

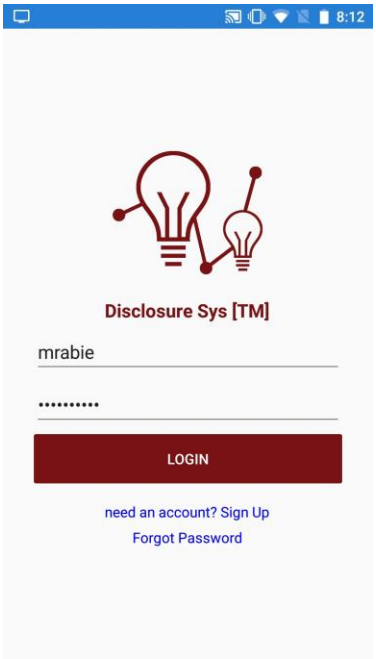


Figure C-16: DisclosureSys Login

1. Enter your “username” or “email address”
2. Enter your password
3. Click “LOGIN”
4. On successful login, the app will display a list of your current disclosure drafts.

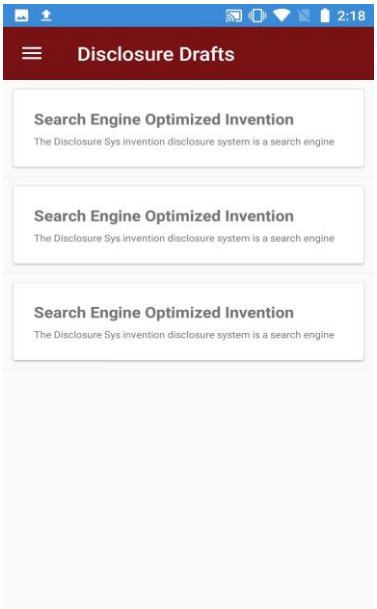


Figure C-17: Forgot Password Screen

5. Tap on the Hamburger Menu ( ) icon (top left corner of the screen), to display all the available feature and navigate through the app.

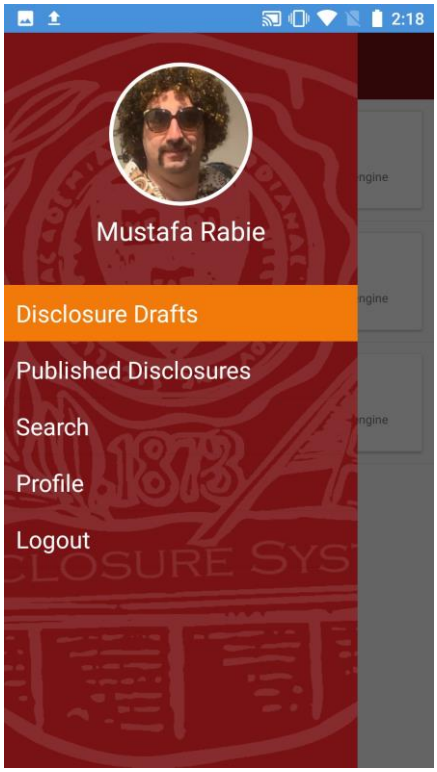


Figure C-18: application Navigation Menu

6. You can perform the following tasks:
    a. Disclosure Drafts: Lists all your current active drafts
    b. Published Disclosures: Lists all your published disclosures
    c. Search: Searches all published disclosures in the DisclosureSys system
    d. Profile: Views the current user profile
    e. Logout: Logs you out of the application

Disclosure Drafts

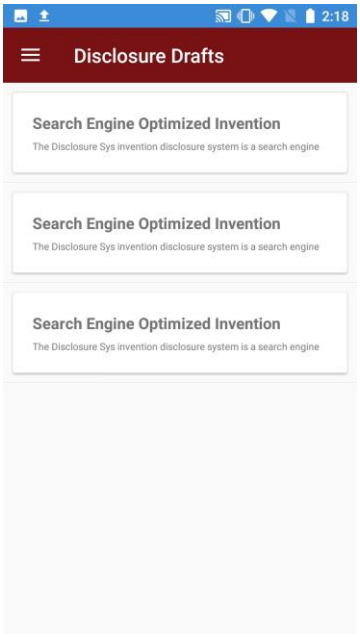


Figure C-19: Disclosure Drafts List

1. Displays a list of all your active disclosure drafts.
2. Tap on any of the disclosures to view its details.

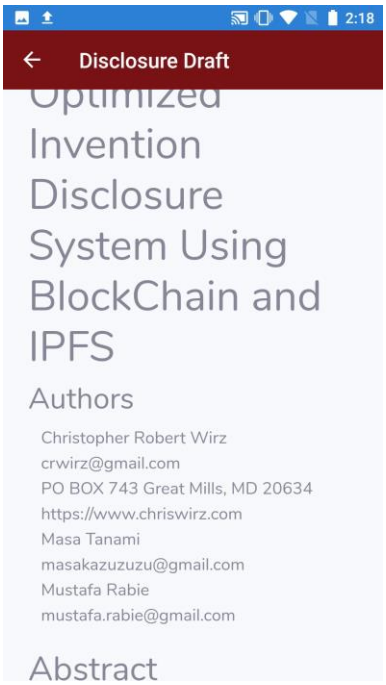


Figure C-20: Disclosure Draft Preview

3. Tap the ← arrow (top left corner of the screen) to go back to the previous list.

Published Disclosures

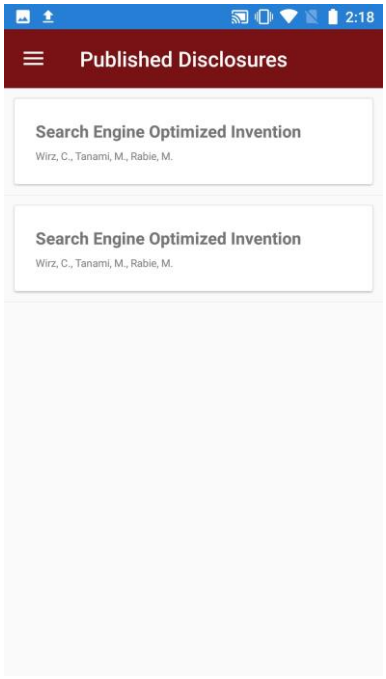


Figure C-21: Published Disclosures List

1. Displays a list of all your active disclosure drafts.
2. Tap on any of the disclosures to view its details.

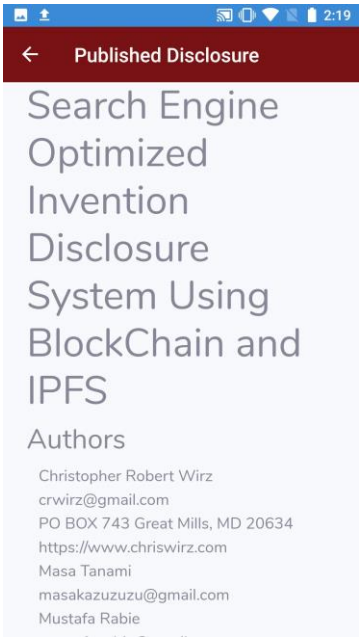


Figure C-22: Published Disclosure List

3. Tap the ← arrow (top left corner of the screen) to go back to the previous list.

Search

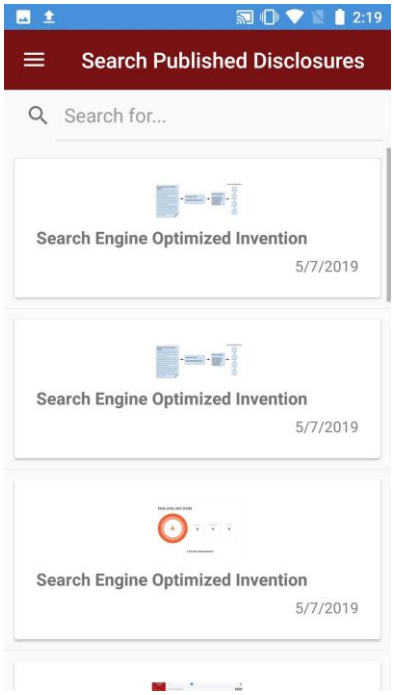


Figure C-23: Main Search

1. Displays a list of all published disclosures in the DisclosureSys system and enables searching by disclosure Title.
2. Tap on the Search bar and enter the part of the title you are looking for. The search list is filtered as you type each character.

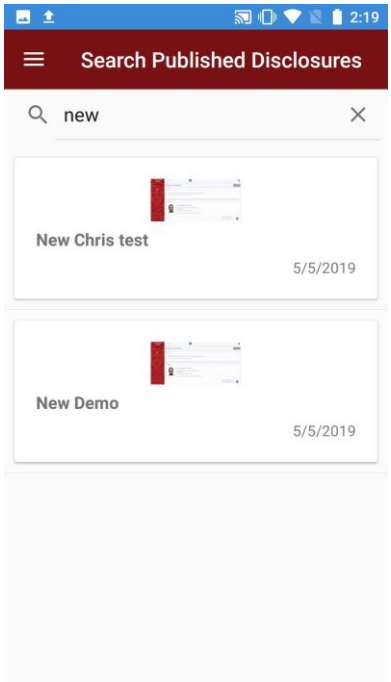


Figure C-24: Search Results

Profile

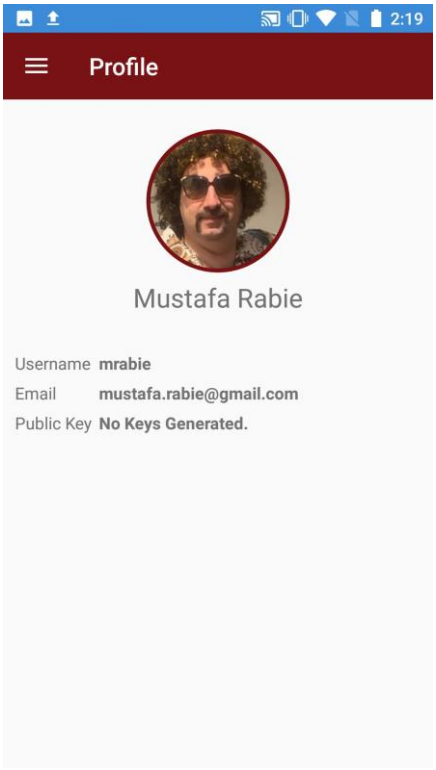


Figure C-25: Profile Details

1. Displays the current user profile information:
    a. Profile Picture
    b. Full Name
    c. Username
    d. Email
    e. Public key (if user generated blockchain account keys)

Logout

1. Logs you out of the application.
2. Takes you back to the login screen.

System Installation Manual

DisclosureSys is composed of 2 parts, AWS Serverless API and a web Single Page Application (SPA).

AWS Setup

Setup a user for the project

Please follow the AWS instructions found in the link below for creating an IAM user for your account.
https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html
Select AWS access type: Select “Programmatic access”
Set Permissions: “Add user to group” select “Administrators”

Please follow the Developer Manual for deploying the solution from Visual Studio.

Visual Studio Setup

Download Visual Studio from https://visualstudio.microsoft.com/vs/ (community edition is sufficient)

Install Visual Studio following the installation wizard. Visit https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2019 for more details. Make sure you install .Net Core cross-platform development.

After the successful installation of Visual Studio, download and install AWS Toolkit for Visual Studio https://aws.amazon.com/visualstudio/

GO Implementation

go-ipfs is the primary reference implementation of IPFS. It is a command-line application, but it can also be used as a library in other Go programs. The installation guide is found at https://docs.ipfs.io/introduction/install/. Developers use the CLI commands to initialize and run the IPFS daemon to go online, then to add or retrieve a file. When an IPFS node is running as a daemon, our program can control the node through an HTTP API (ex. /ip4/127.0.0.1/tcp/5001), additionally a read-only gateway server is provided (ex. /ip4/127.0.0.1/tcp/8080) so that files can be viewed from a web browser.

1. `ipfs init` to initialize the repository
2. `ipfs daemon` to run the daemon in another terminal or background
3. `ipfs add <file1> <file2>` to add files to IPFS
4. `ipfs cat <hash>` to get a file from IPFS hash

```
$ ipfs daemon &
$ Initializing daemon...
go-ipfs version: 0.4.18-
Repo version: 7
System version: amd64/darwin
Golang version: go1.11.1
Successfully raised file descriptor limit to 2048.
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/192.168.1.100/tcp/4001
Swarm listening on /ip6/2405:9800:bc11:eab2:141a:1470:a469:2638/tcp/4001
Swarm listening on /ip6/2405:9800:bc11:eab2:a48b:e69a:6667:23a6/tcp/4001
Swarm listening on /ip6/2405:9800:bc11:eab2:b11f:d838:9777:b23e/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit
Swarm announcing /ip4/100.106.83.102/tcp/48813
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/192.168.1.100/tcp/4001
Swarm announcing /ip6/2405:9800:bc11:eab2:141a:1470:a469:2638/tcp/4001
Swarm announcing /ip6/2405:9800:bc11:eab2:a48b:e69a:6667:23a6/tcp/4001
Swarm announcing /ip6/2405:9800:bc11:eab2:b11f:d838:9777:b23e/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready

$ ipfs add abc
added QmXwVbWJ6AjGdYyYFexPX64QZdP5mMUQwtrVbMbwLcbALE abc
 10 B / 10 B [=============================================================] 100.00%
$ ipfs cat QmXwVbWJ6AjGdYyYFexPX64QZdP5mMUQwtrVbMbwLcbALE
aaabbbccc
```

Figure C-12: Basic IPFS CLI commands

JS Implementation

There are two implementations in JS. A full implementation of IPFS **js-ipfs**, and a smaller library **js-ipfs-api** that controls a running IPFS node via HTTP API. In order to ensure the stability of the node, it is recommended to use js-ipfs-api. The difference is explained in https://docs.ipfs.io/reference/js/overview/. The Disclosure System uses js-ipfs-api and connects to our IPFS node and controls it using the Files API. For more information see https://github.com/ipfs/interface-js-ipfs-core/blob/master/SPEC/FILES.md.

Since the js-ipfs-api uses node.js, it can be installed by npm:

```
npm install --save ipfs-http-client
```

For use in a web browser, install through browserify, webpack, or CDN. The installation details can be found from https://github.com/ipfs/js-ipfs-http-client.

IPFS Addressing in Web Browsers

HTTP gateways are provided to help browsers speak with the IPFS protocol. By using IPFS hashes, it is possible to read files using web browsers. Generally, we can use the following gateways.

1. http://127.0.0.1:8080/ipfs/<hash> provided by local IPFS daemon
2. http://ipfs.disclosuresys.com/ipfs/<hash> provided by our remote IPFS daemon
3. https://ipfs.io/ipfs/<hash> provided by IPFS developer, Protocol Labs
4. https://gateway.ipfs.io/ipfs/<hash> provided by Protocol Labs
5. https://ipfs.infura.io/<hash> provided by Consensys
6. http://example-gateway.com/ipfs/<hash> provided by Cloudflare

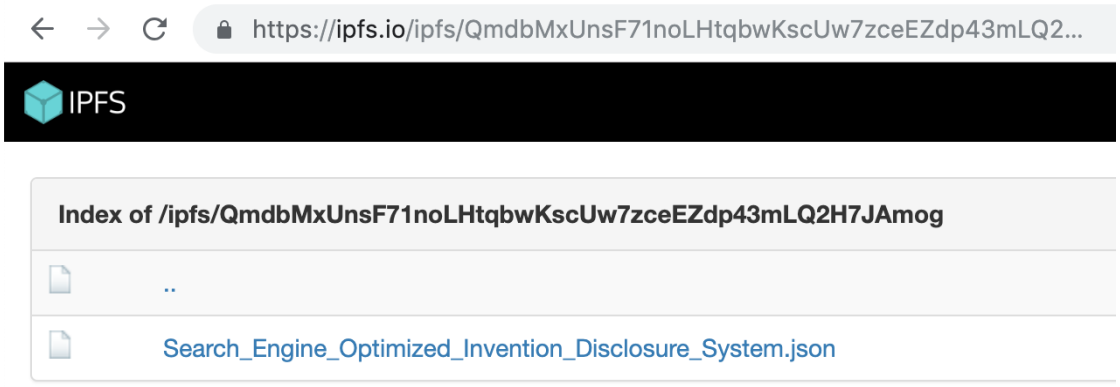


Figure C-13: Reading data on IPFS through an HTTP gateway

Ethereum Blockchain

Solidity Programming Language

There are several programming languages that can be compiled to run a smart contract on the Ethereum network. Solidity, an object-oriented, Turing-complete, and high-level programming language became the most popular language. The Disclosure System chose Ethereum for its popularity and flexibility, and because its smart contract can store disclosure data such as, an IPFS hash that links to the disclosure content published on the IPFS network, a timestamp, a user's ID, a title, and authors. The description of Solidity can be found in https://solidity.readthedocs.io/en/latest/index.html.

Solidity Compiler

A smart contract written in Solidity needs to be compiled to deploy it to the Ethereum network. The Solidity compiler generates a bytecode and an Application Binary Interface (ABI) in JSON format. A bytecode is used to deploy, and an ABI is used to interact with the smart contract deployed on the network. In other words, if you have a bytecode and an ABI, you don't need to have the Solidity source code to deploy and/or use a smart contract. There are various ways to install the Solidity compiler described in https://solidity.readthedocs.io/en/latest/installing-solidity.html.

CLI for the Ethereum Network

There are three command line tools for the Ethereum network implemented by Go, C++, and Python. The Go implementation is the most popular and it is called Geth. Developers may install it and run an Ethereum node to interact with smart contracts and may act as a miner. Detailed explanations can be found in https://www.ethereum.org/cli. A public Ethereum node endpoint provider exists such as https://Infura.io and the Disclosure System uses its endpoint to interact with its smart contract via the HTTP API.

Web3 Library to Interact with Ethereum

Instead of using command line tools, we can interact with Ethereum smart contracts programmatically by using the Web3 library. There are two implementations, Web3.js and Web3.py. The Disclosure System provides its services through the web and uses Web3.js to let users interact with the smart contract via web browsers. Therefore, when a user submits a disclosure, the system will store the disclosure data in JSON format on the IPFS network, and then write its IPFS hash with its timestamp on the Ethereum Network. The Web3.js documentation is here: https://github.com/ethereum/wiki/wiki/JavaScript-API.

MetaMask Wallet (only for main Ethereum block chain)

When it comes to interacting with a smart contract and a web browser some actions that cause the state change will cost Ether, this is necessary to pay for miners to

verify the transaction and make it permanently unforgeable. In the DisclosureSys, retrieving the stored disclosure item has no cost but adding a new disclosure item costs around ten cents. The way users pay Ether is unlocking the transaction with their private key; inputting the key to an html form should be avoided. Instead, using a wallet to manage accounts and private keys is recommended. MetaMask is an Ethereum wallet that can be installed as a browser extension. It keeps encrypted account information in user's browser and seamlessly interact with the Ethereum network. The official website is https://metamask.io/. In the current implementation, since it is an alpha state, DisclosureSys uses a testnet and the Ether is free, thus users are not required to use MetaMask.

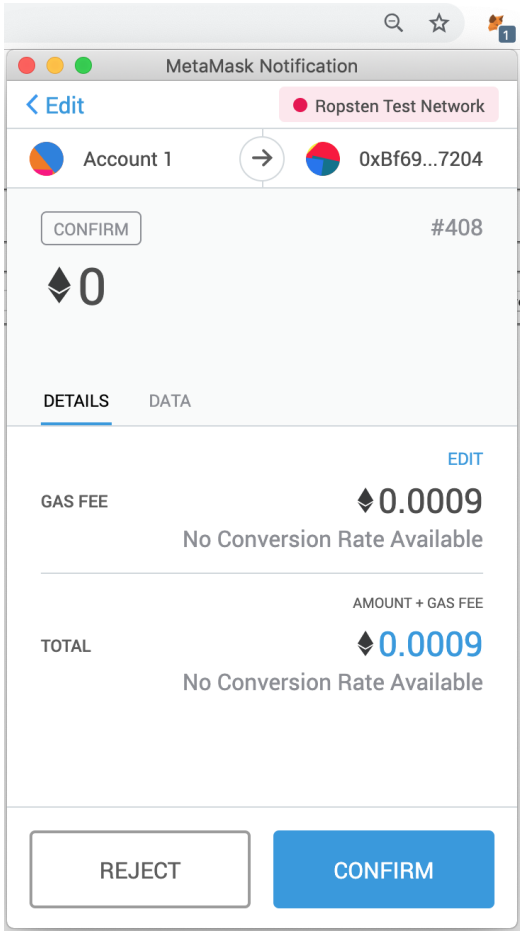


Figure C-14: Sending a transaction from a browser with MetaMask

Etherscan

Etherscan is a block explorer managed by Ethereum Foundation and allows the public to easily lookup, confirm and validate transactions on the Ethereum Blockchain. Transactions can be found from a smart contract address, or a transaction hash. The DisclosureSys currently use Ropsten Testnet, and the explorer can be found here: https://ropsten.etherscan.io/.

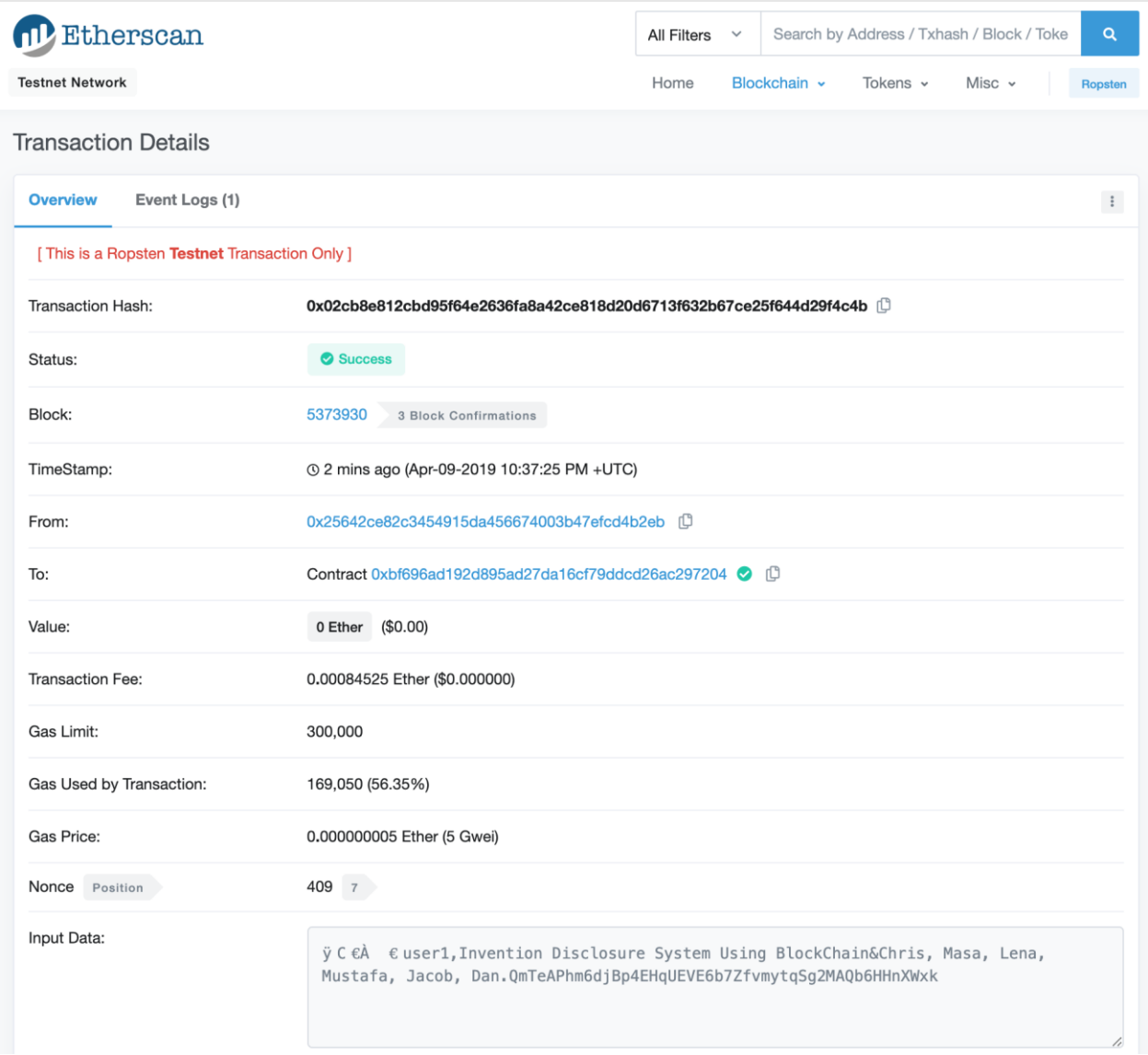


Figure C-15: Showing Transaction Details with Etherscan

Developer Installation Manual

Developers can get the source code for the project using their favorite Git client. The code lives in https://cscie599@dev.azure.com/cscie599/SWE%20Invention%20Blockchain/_git/SWE%20Invention%20Blockchain

There are 5 main folders on the repo

AWS CloudFormation	The main folder for the AWS Serverless Application API with the C# code and deployment template.
SharedCS	Shared models and methods used by the API, written in C#.
SmartContract	Solidity files defining DisclosureSys Blockchain smart contracts
Static Website	DisclosureSys Web Application
DSMobileSolution	iOS and Android DSMobile applications

DisclosureSys API

Compile

1. From the “AWS CloudFormation” folder open the “IDSServerless.sln” solution file with Visual Studio
2. Build the solution

Deploy

Before the API can be deployed to AWS CloudFormation, an AWS profile must be configured.

Configuring AWS Profile

Please follow the following AWS guide to configuring AWS Toolkit Profile in Visual Studio
https://docs.aws.amazon.com/toolkit-for-visual-studio/latest/user-guide/credentials.html
After configuring your AWS Toolkit Profile

1. In Visual Studio Solution Explorer, right click on “IDSServerless” project
2. Click on “Publish to AWS Lambda”

3. On the “Publish AWS Serverless Application” window

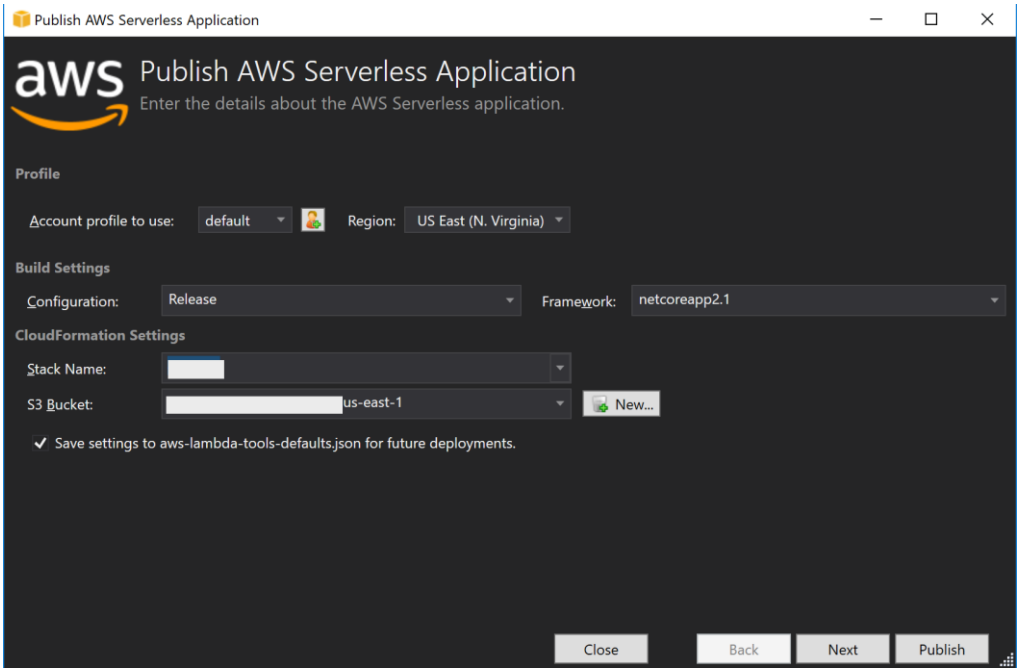


Figure C-16: Configuring publishing AWS Lambda

a. Choose your “Account profile to use”
b. Set the “Region”
c. Enter a stack name or choose a pre-created stack
d. S3 Bucket, click “New” and create a new bucket
e. Make sure “Save settings to aws-lambda-tools-defaults.json for future deployments.” is checked
f. Click “Next”

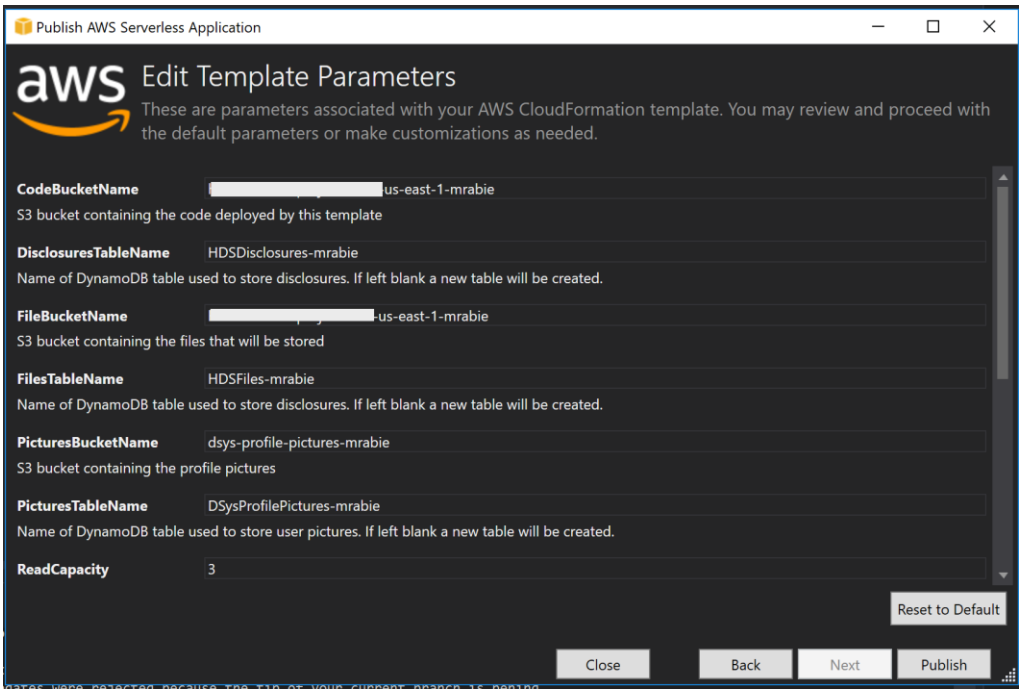


Figure C-17: Configurations review

g. Review your configurations and click “Publish”

4. Once successfully published, copy the “AWS Serverless URL”

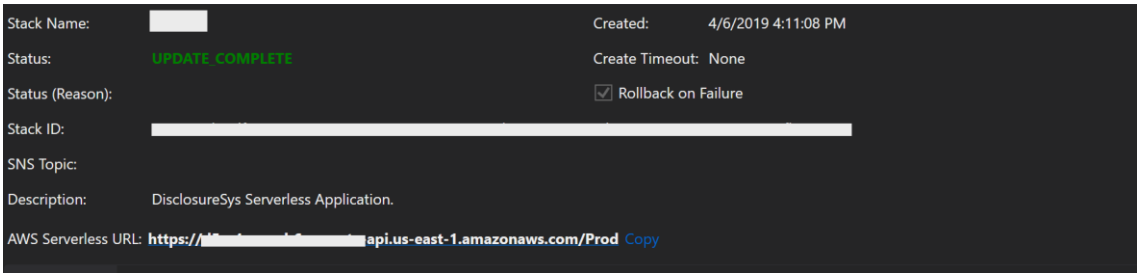


Figure C-18: Creation receipt

Please Note: Works on Windows 10 only (AWS SDK is currently not available for Mac)

Configure SPA Web Application

After publishing the DisclosureSys API, the web application client must be configured to use the DisclosureSys API endpoints.

Cognito User Pool

Get AWS Cognito user pool credentials

1. Login to your AWS Management Console account
2. Choose “Cognito” from the “Services” menu
3. Click “Manage User Pools”

4. Click “DSysUserPool”
5. Copy “Pool Id”
6. From the left navigation bar click “App Integration”
7. Click “App client settings”
8. Copy “App client DSysWebClient” id

Static Website

1. Go to “Static Website” folder
2. Go to “js” folder
3. Edit “DSApplication.js” file with your favorite editor
4. Update the “invokeUrl” to your AWS Serverless URL
5. Update “dataBucket” to your AWS bucket
6. Save

You can run the Web application locally by invoking the “index.html” file or host it to a hosting service of your choice.

Mobile App

1. Go to “DSMobuleSolution” folder, under the main repo.
2. Open “DSMobileSolution.sln” in Visual Studio 2017/2019 for Visual Studio for Mac.

Run/Debug the application
On Windows:
You can run Android emulators on Windows machine
For more information:
https://developer.xamarin.com/guides/android/deployment,_testing,_and_metrics/debugging/debug-on-emulator/visual-studio-android-emulator/

1. Set “DSMobile.Android” as the startup project
2. Choose your emulator
3. Click on the emulator to run the application.

On Mac
Follow the same steps as Windows. XCode and Visual Studio must be installed.
More information on VS mobile development on Mac:
https://docs.microsoft.com/en-us/visualstudio/mac/xamarin?view=vsmac-2019

APPENDIX - D

Lines of Code Metrics

Language	files	blank	comment	code
JavaScript	58	21490	21171	119391
CSS	17	4845	15	18449
JSON	18	0	0	15546
C#	120	5653	4856	14455
SASS	130	1900	1214	10918
HTML	14	706	650	5123
LESS	18	502	35	4518
XAML	15	30	2	468
MSBuild script	5	5	0	444
XML	8	11	10	145
Markdown	2	26	0	42
YAML	1	0	0	14
TOTAL	**406**	**35168**	**27953**	**189513**