Analysis of Arbitrary Content on Blockchain-Based Systems using BigQuery

Marcel Gregoriadis
marcel.gregoriadis@hu-berlin.de
Humboldt University of Berlin
Berlin, Germany

Robert Muth
muth@tu-berlin.de
Technical University of Berlin
Berlin, Germany

Martin Florian
martin.florian@hu-berlin.de
Weizenbaum Institute
Berlin, Germany

ABSTRACT

Blockchain-based systems have gained immense popularity as enablers of independent asset transfers and smart contract functionality. They have also, since as early as the first Bitcoin blocks, been used for storing arbitrary contents such as texts and images. On-chain data storage functionality is useful for a variety of legitimate use cases. It does, however, also pose a systematic risk. If abused, for example by posting illegal contents on a public blockchain, data storage functionality can lead to legal consequences for operators and users that need to store and distribute the blockchain, thereby threatening the operational availability of entire blockchain ecosystems. In this paper, we develop and apply a cloud-based approach for quickly discovering and classifying content on public blockchains. Our method can be adapted to different blockchain systems and offers insights into content-related usage patterns and potential cases of abuse. We apply our method on the two most prominent public blockchain systems—Bitcoin and Ethereum—and discuss our results. To the best of our knowledge, the presented study is the first to systematically analyze non-financial content stored on the Ethereum blockchain and the first to present a side-by-side comparison between different blockchains in terms of the quality and quantity of stored data.

CCS CONCEPTS

• **Computer systems organization** → *Peer-to-peer architectures*;
• **Applied computing** → *Data recovery*.

KEYWORDS

Blockchain, Cryptocurrency, Ethereum, Bitcoin, BigQuery

ACM Reference Format:
Marcel Gregoriadis, Robert Muth, and Martin Florian. 2022. Analysis of Arbitrary Content on Blockchain-Based Systems using BigQuery. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3487553.3524628

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524628

1 INTRODUCTION

Since the introduction of Bitcoin [17, 23] and Ethereum [6], blockchains became an important part for many financial-based services; the most prevalent being cryptocurrencies. Besides financial services, with the introduction of *smart contracts* [20, 24] it also became possible to execute Turing-complete programs and host *decentralized apps* (DApp) on the blockchain. On the one hand, the general ability to insert arbitrary data into transactions made it possible to exploit blockchains as a distributed storage medium. On the other hand, the ability to insert arbitrary data is a desired feature required by various DApps like governance platforms [14], digital asset marketplaces [1], and social networks [18].

By allowing arbitrary data in transactions, objectionable content can find its way onto the public ledger [15]. Considering that transactions on the blockchain can be sent anonymously, the potential for criminal intentions becomes evident. In the case of actual illicit content, such as copyright violations, personality rights violations, or incitement to hatred, the possession of the blockchain could become prosecuted in some jurisdictions. This is not just a theoretical scenario: In an analysis of the Bitcoin blockchain, Matzutt et al. were even able to detect, among other things, links to child pornography [16]. Since blockchain clients have to maintain a full copy of the blockchain for its operation, the capability to insert arbitrary data poses a serious threat to node operators in public blockchain-based systems, and in effect also to the resilience of the systems themselves.

In this paper, we present and discuss the results of a quantitative and qualitative analysis of arbitrary content on the Bitcoin and Ethereum blockchains. To this end, we also introduce the cloud-based analysis pipeline we developed, which uses Google BigQuery[1] as data source and SQL-based analysis toolkit.

We focused on Bitcoin and Ethereum which we consider to be the most relevant blockchains (with a total market capitalization of ≈ 1 trillion USD, as of February 2nd, 2022 [7]). Taking into account that cryptocurrencies to this date remain the predominant application of blockchain technology, and then the high combined market share of Bitcoin and Ethereum of 59.6 % (of which not even all are blockchain-based), we assess our results to be representative for the emergence of arbitrary content in public blockchains.

We stress that it is not our primary goal to identify every single occurrence of arbitrary content. Rather, our contributions can be summarized as the following:

- We present and evaluate BigQuery as a cloud-based approach to analyze blockchain data.

[1] https://cloud.google.com/bigquery

- We present novel data analysis methods for the Ethereum blockchain.
- Using our reusable data discovery framework, we obtain quantitative and qualitative insights on the (non-financial) data stored on Bitcoin and Ethereum for the time frame up to January 2022, allowing comparisons between usage patterns and the general discussion of trends.

The remainder of this paper is structured as follows: In Section 2 we discuss related work, also pointing out how our work compares to previous studies. Section 3 gives a general overview of our method; foremost how we extract and filter the data that we want to examine. Our main part, which is structured in Section 4 and 5, specifies how we detect text and file contents, and gives a quantitative and qualitative overview on the results. Section 6 continues with an in-depth discussion of the results and an evaluation of our methodology. Finally, in Section 7 we summarize the main conclusions of our study.

2 RELATED WORK

Since the beginning of blockchains, it has been common practice to include non-financial, arbitrary data, as for example, in the Bitcoin genesis block [19]. Non-financial data storage on blockchains enables interesting services such as name services[2], timestamping[3], non-equivocation logging [22], and the management of arbitrary forms of ownership, for example, in form of non-fungible tokens (NFTs). An uncensorable public data storage service, as provided by Bitcoin and Ethereum, can also be abused, however.

In [16], Matzutt et al. investigated the prevalence of non-financial data on the Bitcoin blockchain as of August 2017, uncovering, among others, hundreds of links to child pornographic material. The authors identified different approaches for injecting chunks of data into the Bitcoin blockchain. On a low and general level, these are `OP_RETURN` output scripts, non-standard transactions, coinbase input scripts, and pay-to-script-hash input scripts. The authors scanned the Bitcoin blockchain for instances in which these methods have been used, arriving at a quantitative estimate of their popularity. To also account for data hidden in standard payment transactions, Matzutt et al. additionally filtered out all transaction outputs that contain $\geq$ 90% printable ASCII characters in their mutable, i.e., non-opcode part. In addition to these low-level content detectors, the authors also implemented methods tailored towards detecting data inserted through content insertion services such as *CryptoGraffiti*[4]. Combing all content detection methods, the authors concluded that 1.4 % of Bitcoin transactions at the time of the study contained non-financial data. In total, Matzutt et al. were able to decode over 1,600 viewable files, containing data such as images, text messages and source code.

Compared to the work in [16], we constrain our analysis to data inserted through the more general, low-level insertion techniques. By disregarding the more specialized detection mechanisms proposed by Matzutt et al., we undoubtedly miss some of their findings. Due to the principally unbounded number of ways in which non-financial data can be encoded on the Bitcoin blockchain, even the approach in [16] cannot be assumed to produce an exhaustive list of findings. It must also be pointed out that any hand-tailored detectors need to be updated continuously as the landscape of content insertion services changes. CryptoGraffiti has switched to storing data only on the *Bitcoin Cash* blockchain, for example, and is therefore not a relevant content insertion service for Bitcoin anymore. Our cloud-based analysis toolkit, which we have released as open source, can flexibly be extended with hand-tailored content detectors.

Showcasing the flexibility of our cloud-based approach, we also apply our analysis methods and tooling to uncover non-financial data stored on the Ethereum [6] blockchain. To the best of our knowledge, we are the first to conduct a systematic study into the quantity and quality of data stored on Ethereum. Our analysis approach and tooling thereby enable insights related to the two largest blockchain networks by market capitalization of the underlying cryptocurrency. In addition to being open source, and likely also in contrast to previously developed content detection tools, our implementation leverages public cloud infrastructure and can thereby be used largely independently of locally available computing resources.

Various approaches have been developed for dealing with the dangers of arbitrary content insertion on public blockchains. They can be grouped into categories as follows: avoiding the inclusion of unwanted data [15], allowing the modification (and erasure) of past blockchain state [2, 9, 13, 21], and local pruning [12]. In this paper, we focus on developing tools for determining the severity of the original problem and whether the implementation of additional protection approaches is necessary.

3 METHOD

Our content detection and analysis methodology consists of a data extraction step and several optional post-processing and data analysis steps. While data extraction operates on the entirety of the investigated blockchain, post-processing and data analysis need to be performed only on the result of the first step, i.e., on a significantly smaller amount of data. In the following, we introduce the overall characteristics of our data extraction methodology, the clearly more challenging part of our pipeline. We will introduce further details on specific queries and post-processing steps in the subsequent sections (4 and 5), where we also present concrete quantitative and qualitative findings for the Bitcoin and Ethereum blockchains. As a complement to this paper, we release all tools and queries we developed for performing the discussed analyses as open source[5] , enabling the easy reproduction of our results and the extension of our study to new areas of interest.

3.1 Data Extraction using BigQuery

As a distinguishing feature of our approach, we leverage the Google Cloud BigQuery service for accessing and pre-filtering blockchain data. BigQuery provides constantly updated datasets for a wide range of popular public blockchains. All data is stored in table-structured databases which are SQL-queryable. In BigQuery, transaction data (such as smart contracts or scripts) is provided as hexadecimal strings which enable convenient SQL accessibility, such

[2] https://namecoin.info
[3] https://opentimestamps.org
[4] https://cryptograffiti.info
[5] https://github.com/mg98/arbitrary-data-on-blockchains

as the LIKE-operator (for string comparisons with wildcards) and functions around regular expressions.

The main benefits of using the cloud service BigQuery for blockchain analysis is that there are no (expensive) hardware requirements and queries can be developed relatively easy. Leveraging the provided cloud infrastructure allows to run complex queries on large datasets[6] significantly faster than with commonly available on-premise hardware. The short response times render the analysis process interactive and allow adjusting queries without much effort. Consequently, our methodology can readily be extended to support so-far overlooked data insertion techniques that might be of interest.

3.2 Content Detection and Limitations

For this paper, we implemented multiple content detectors for the Bitcoin and Ethereum blockchains. Our detectors are based on observations from practice as well as previous work such as [16]. They discover various texts (including URLs) and files, also reproducing previous data discoveries. Still, our detectors clearly do not enable an *exhaustive* view over *all* content stored on the Bitcoin or Ethereum blockchains. For both, Bitcoin and Ethereum, a wide range of data insertion methods is conceivable. Many data insertion methods result in data that is only discoverable using fine-grained queries, if at all. Encrypted data, for example, is difficult to distinguish from random noise if it does not come with identifying meta attributes (e.g., PGP headers).

In the remainder of this section, we give an overview over our data extraction methods for Bitcoin and Ethereum. We give more details on specific queries and post-processing steps in the subsequent sections that also describe our empirical findings.

3.3 Content on Bitcoin

Based on the content detectors used in [16] and our own observations about commonly used approaches for inserting arbitrary content into the Bitcoin blockchain, our content detectors for Bitcoin focus on scanning the *outputs* and *inputs* of transactions, including the inputs to *coinbase* transactions.

In BigQuery, the data on transaction inputs and outputs is organized in respectively named views. Each record in the view maps to a transaction or block. Inputs and outputs play a role outside of serving as a data store and must therefore include specific byte sequences to ensure their validity. Therefore, the range of bytes that can be freely used for encoding arbitrary data is limited. For our Bitcoin analysis, we analyze the concatenated bytes of these *mutable values*. From the perspective of our queries, the investigated mutable values are the hexadecimal strings in the *script_asm* field on input and output records. In other words, we leverage the fact that opcode bytes are made visible in the view provided by BigQuery, and discard them for the further analysis. In order to analyze coinbase transactions, we contemplate the data within the *coinbase_param* field of the *blocks* table provided by BigQuery.

We further classify our results by the insertion method identified. Those comprise insertions via:

- *Standard outputs* (P2X), such as pay-to-public-key (P2PK), pay-to-public-key-hash (P2PKH), and pay-to-multi-signature (P2MS) outputs.
- *Standard inputs*, most prominently P2PKH inputs.
- OP_RETURN *outputs*, which were introduced specifically for the purpose of including arbitrary data.
- *Non-standard inputs and outputs*, excluding OP_RETURN outputs.
- *Coinbase inputs*.

3.4 Content on Ethereum

Our Ethereum analysis is mostly limited to the *input* field of the *transactions* table in the dataset. In the Ethereum protocol, this field is used to deploy or call a smart contract. Due to the variability of length and content of this field, it has been considered the most viable option for intentional insertions of arbitrary content.

Unlike for Bitcoin, the Ethereum data set on BigQuery does not provide a field in which bytecode instructions are explicitly labeled as such. The reason is likely the fact that the Ethereum bytecode instruction set is significantly more complex. For the same reason, detecting bytecode instructions via a (SQL-)query is infeasible. We therefore apply our queries on the complete and unfiltered input fields of transactions, potentially misidentifying bytecodes as parts of stored content. Also unlike for Bitcoin, we ignored content (usually text) stored in coinbase transactions. Preliminary experiments demonstrated that, as a whole, the content stored in coinbase transactions has both an immense volume and a very low variance—the vast majority of coinbase transactions simply advertise the mining pool that is responsible for the current block.

4 TEXT ANALYSIS

In the following, we give more details on our text detection methods and present quantitative and qualitative text discovery results for the Bitcoin and Ethereum blockchains. All presented results, in this and the subsequent sections, pertain to the state of the respective blockchains on January 29th, 2022.

4.1 Text Classification

For out text analysis, we developed a regular expression pattern to match all kind of combinations of UTF-8 characters and their hexadecimal representation, respectively. In a highly simplified manner that ignores technical steps such as the subtraction of opcodes, the relevant queries for Bitcoin can be described as follows:

```
From each transaction, select
  the concatenation of all standard output scripts
    if at least 90% of the bytes in the transaction
    represent printable UTF-8 characters,
  the concatenation of all non-standard input scripts
    if 100% of the bytes in the resulting value
    represent printable UTF-8 characters,
  the concatenation of all OP_RETURN output scripts
    if 100% of the bytes after each OP_RETURN opcode
    represent printable UTF-8 characters, and
  a concatenation of all non-standard output scripts
    which are not OP_RETURN outputs
    if 100% of the bytes in the resulting value
```

[6] At time of writing, the Bitcoin and Ethereum blockchains had a size of, respectively, more than 380 GB and more than 4 TB (for the full blockchain archive).

```
    represent printable UTF-8 characters.
Also select the coinbase input of each block
  if 100% of its bytes
  represent printable UTF-8 characters.
```

In a similar syntax, the queries for Ethereum can be written as:

```
Select the input field of each transaction
  if 100% of its bytes
  represent printable UTF-8 characters.
```

As a further analysis step, we extract the matched strings and classify them based on the occurrence of common textual data and formats. More specifically, we classify found text blocks based on the following categories:

- **Strings:** Sequence of printable characters which does not contain any white spaces.
- **Texts:** Sequence of printable characters which contains at least one white space (i.e., multiple words).
- **Contain URL:** Contains a string that matches a URL pattern[7].
- **Contain Email Address:** Contains a string that matches a generous pattern of a typical email address.
- **Contain JSON:** Contains a string that can be decoded as a JSON object; our method does not detect simple array outputs or empty objects, i.e., "{}".
- **Contain PGP:** Text contains a string that is enclosed with a PGP header.
- **Contain HTML/XML:** Text contains a sequence that follows the semantics of HTML/XML (with a beginning and closing tag).
- **Contain Data URL:** Text contains a data URL (URI scheme containing a Base64-encoded version of a file that is used to display files in-line in web pages).

As a result, our analysis returned **763,035** corresponding transactions for Bitcoin and **1,916,836** for Ethereum. Table 1 shows the quantitative results by classification. We point out that the majority of all text messages on Bitcoin have been embedded using `OP_RETURN` (78.3 %) or through a coinbase transaction (21.2 %).

	Occurrences		
Textual Type	**Total**	**Bitcoin**	**Ethereum**
Strings	632,547	94,812	537,735
Texts	2,047,324	668,223	1,379,101
Contain JSON	51,128	2,065	43,063
Contain HEX	92,263	3,716	88,547
Contain Email Address	1,008	39	969
Contain URL[7]	7,435	4,341	3,094
Contain PGP	325	28	297
Contain HTML/XML	346	202	144
Contain Data URL	11	0	11

Table 1: Quantitative analysis of textual type of content.

[7]The analysis here operates on the results of our text detector and thereby misses transactions that are mainly non-text, such as smart contract invocations. In Section 4.4 we discuss detecting URLs in arbitrary contexts.

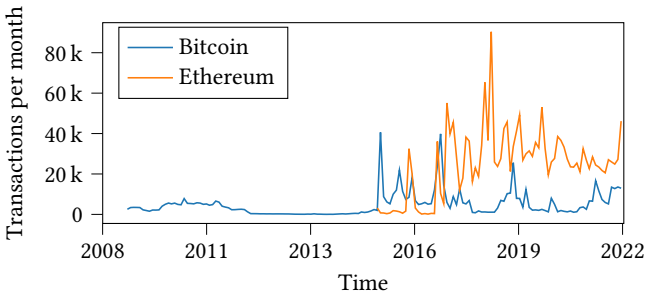


Figure 1: Frequency of text transactions over time.

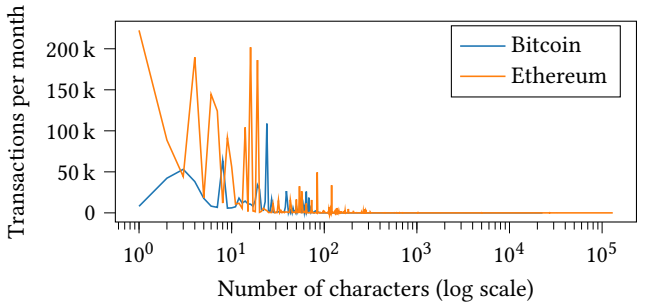


Figure 2: Frequency of text lengths.

4.2 Frequency Analysis and Popular Strings

In the following, we analyzed the corresponding transactions and counted the occurrences of extracted texts. Figure 1 shows the frequency of text messages in transactions over time, for both blockchains. In Figure 2, we depict the frequency of texts by length. As can be seen, most of the texts are short. The closer inspection of a sample of the results made it obvious that a lot of messages appear several times or have a similar structure. We have listed the most common embedded texts in Table 2 and 3. Further investigation has shown that not only those texts but also texts sharing the same structure caused the peaks in Figure 2. We have also observed that those transactions would often be received by the same recipient, e.g., Kraken[8].

#	Text	Occurrences
1	Bitzlato	60,625
2	ASCRIBESPOOL01PIECE	28,986
3	503: Bitcoin over capacity!	13,902
4	"2265861855@qq.com"}	12,880
5	We'll buy your Bitcoins. sell.buy.bitcoin@…	10,790
6	"2265861855@QQ.COM"}	9,800
7	WWW.BTCKEY.ORG Bitcoin wallet recovery…	5,530
8	ASCRIBESPOOL01FUEL	5,125
9	Bitcoin: A Peer-to-Peer Electronic Cash System	4,843
10	ASCRIBESPOOL01EDITIONS1	4,270

Table 2: Top 10 most frequently embedded texts found on Bitcoin (#5 and #7 have been truncated).

[8]https://kraken.com

#	Text	Occurrences
1		196,211
2	BFX_REFILL_SWEEP	179,820
3	hotwallet drain fee	177,454
4	Ignore	126,140
5	imtoken	96,807
6	coinbenerefuel	95,385
7	undefined	61,226
8	fzX	55,924
9	FA%}	54,684
10	cs	48,079

Table 3: Top 10 most frequently embedded texts found on Ethereum.

4.3 Qualitative Observations

Individual content analysis has shown many messages, even bidirectional conversations, to when funds were accidentally sent to the wrong address or when funds got stolen from a user. The victim would then try to ask or negotiate to get his or her funds back. Presumably to save on transaction fees, some longer messages were shared through the URL of an online service like *PasteBin*[9]. We have highlighted a selection of text messages in Appendix A.

We also decoded the 11 data URLs that were found to image type files (JPEG, PNG, GIF). One of those was a provocative image showing the Chinese president Xi Jinping as Winnie the Pooh.

4.4 Analysis of URLs

URLs persisted on the blockchain can point to more rich resources in the internet, such as images, videos, long texts, and other files. This is interesting for qualitative observations and also in search of illegal contents. With this in mind, we created a method to scan the Ethereum blockchain more exhaustively for URLs. As opposed to our basic text analysis, we now aim at finding URL strings also within smart contract calls and deployments. To this end, we created regular expression patterns to match with HTTP and IPFS [3] URLs, with extra attention to URLs pointing to Tor [10] onion services.

In simplified notation that abstracts away technical details such as the encoding and decoding of strings, our main URL detector logic can be written as follows:

```
Select
  all transaction input fields on Ethereum
with occurrences of
  http://*, https://*, ipfs://*
where * matches
  the longest coherent string of URL-valid characters
  and is at least 5 characters long.
Also select
  all transaction input fields on Ethereum
with occurrences of
  *.onion
where * matches
 the longest coherent string of alphanumeric characters
  of at least 16 characters.
```

[9]https://pastebin.com

Further extraction, verification, and classification into HTTP, HTTPS, IPFS and .onion links happens in a subsequent post-processing step. For Bitcoin, we could perform those steps directly on the result set from the text analysis. The quantitative results can be observed from Table 4.

	Occurrences		
URL Type	**Total**	**Bitcoin**	**Ethereum**
HTTP	522,313	4,329	517,985
IPFS	210,268	7	210,261
Onion Service	52	6	46
Sum	**732,633**	**4,388**	**728,292**

Table 4: Quantitative analysis of discovered URLs.

In addition to the quantitative analysis, we qualitatively analyzed a sample of 100 HTTP links chosen randomly from our results for each blockchain. Based on our manual analysis, we make the following observations about our sample:

Bitcoin: 19 % of the URLs linked to cryptocurrency-related content, another 11 % linked to social media content (Twitter, Reddit, and YouTube), 2 % showed pornography, and 25 % were miscellaneous. 43 % of the links were dead [10], however most of them were obviously relating to cryptocurrencies (judging by the URL). Also most social media posts related to cryptocurrencies.

Ethereum: 14 % referred to cryptocurrency-related content, 6 % of the URLs responded with a JSON that share characteristics of NFT metadata (comprising keys such as "name", "description", "image", and an array of "attributes"), and 33 % were miscellaneous (e.g., memes). Here as well, 43 % of the links were dead.

From all detected onion service URLs across both blockchains, we were able to access only 3 unique services: an online copy of the bible, a mirror of the official website of the CIA, and a website hosting various texts with "forbidden knowledge" (including tutorials on drug synthesis and explosives). While the other services appeared to be unavailable, online research suggests that some of the found URLs might have linked to child pornography content in the past.

Note that our URL detection method has two explicit limitations:

(1) We ignore the possibility of a single transaction carrying multiple URLs and extract (and count) only the first match.
(2) The pattern we use to find URLs will sometimes wrongly match additional bytes located directly after the end of a matched URL.

The situation outlined in (2) can for example happen when the URL appears inside a JSON structure and our pattern interprets the proceeding "}" as a part of the HTTP path. It can also happen with random ASCII characters caused by general noise. Another cause, in Ethereum, can be direct concatenation of string parameters in a smart contract call. The following string is an actual URL matched by our program:

[10]This includes unavailable websites as well as content platforms (e.g., social media sites or file sharing services) which show that the requested resource has been removed or access is restricted.

```
https://file.soar.earth/d4c4540faf449a9a729edbf9e60d3
621.jpg/previewGhttps://api.soar.earth/v1/download/d4
c4540faf449a9a729edbf9e60d3621.jpg+POINT(115.63155412
67395
```

This example demonstrates the issue raised in (2): Our detectors wrongly matched the subsequent parameter and even a part of another, third parameter in a smart contract call. Syntactically, all of this could still constitute a valid URL, while semantically it is likely the case that they are actually two URLs hidden in the returned string.

Limitations (1) and (2) have a limited impact on our results, however. While the quantitative analysis might be impacted by (1), manual observation of samples of our collected data suggests that transactions with multiple URLs are rare. Regarding the impact of (2) on qualitative analyses, note that our qualitative analysis was performed manually on randomly sampled subset of all findings, which allowed us to manually fix any obviously erroneous URL add-ons.

5 FILES ANALYSIS

Besides text-based arbitrary data, we also searched the Bitcoin and Ethereum blockchains for whole files. In the following, we show how to detect different types of files and present our results on a quantitative and qualitative basis.

5.1 Detecting Files and File Types

In order to find files on the blockchain, we scanned all transactions for the occurrence of popular file type signatures. In the analysis for both blockchains, findings were evaluated from the start of a signature to the very end of the transaction's payload. To this end, we stripped out the opcodes from the payload for the analysis on Bitcoin, and left out file types with very short signatures (e.g., GZIP or DOS executables) because they caused too many false positives.

In a very simplified manner, the queries we used to retrieve our result candidates for Bitcoin can be formulated as follows:

```
From each transaction, select
  the concatenation of all output scripts,
  the concatenation of all non-standard input scripts,
  and the concatenation of all pay-to-script-hash
  input scripts,
with at least one occurrence of the byte sequence of
  a file signature.
Also select the coinbase input of each block
with at least one occurrence of the byte sequence of
  a file signature.
```

The respective query for Ethereum can be written as:

```
Select
  all transaction input fields
with at least one occurrence of the byte sequence of
  a file signature.
Also select the coinbase input of each block
with at least one occurrence of the byte sequence of
  a file signature.
```

In our analysis of Ethereum, we later distinguish our findings between two insertion methods:

- **Embedded:** Transaction data represents the encoded file.
- **Injected:** Transaction data does not start with but ends with the encoded file. This will be considered a smart contract call where the last argument has been the file.

We also examined coinbase transactions in Ethereum but report zero findings there. This is unsurprising given the fact that the coinbase of Ethereum blocks can hold only 32 bytes—too little space for anything but short text messages.

Since searching for longer file signatures still leads to a high number of false positives, we developed post-processing scripts which filter files that cannot be opened with standard software. The scripts also unveil Microsoft Word documents (DOC) that have been identified as ZIP archives before. In the next step, we go over the results manually and remove remaining broken files that our automated post-processing did not filter out.

5.2 Frequency of File Types

As shown in Table 5, we focus on the most popular media file types, documents, and archives. As a result of the post-processing, we analyze the quantitative results for readable files found on both blockchains. As a matter of fact, 764 of the 847 findings were images, also including many duplicates. The majority of images on Ethereum were injected in the context of NFT projects. Other images have partly been categorized, as Table 6 shows.

As of Bitcoin, we found 77.0 % inserted through P2X transactions and 21.8 % inserted through P2SH input scripts. A single occurrence was found in a non-standard output script.

5.3 Examples of Found Images

In the context of the (pseudonymous) anonymity and irreversibility of blockchain transactions, we noted the following selection of found images:

- a swastika
- a photo of a birth certificate laying on a newborn baby
- an academic degree
- a screenshot of the Twitter app showing the Chinese ambassador Liu Xiaoming having liked a post containing suggestive content
- the president of Russia Vladimir Putin in a LGBTQ theme
- a meme making fun of the supreme leader of North Korea Kim Jong-un

5.4 Discussion of Non-Image Files

We also investigated files which are not images. In sum, we found 78 non-image files which we were able to decode. As a result of a manual, predominantly qualitative analysis, we highlight a number of noteworthy findings.

We found several audio files, as for example, 7 seconds of electronic music in generally poor quality. Two other *audio files* played a brief audio interference sound. However, we assume that there are more audio files on the blockchain which we were just not able to decode correctly.

We decoded 4 *PDF documents* which basically contained:

- a white page with the text "Ethereum White Paper"
- the original white paper for Bitcoin

(a) Results for Bitcoin.

File Type	Total
PNG	38
JPEG	42
GIF	2
PDF	2
ZIP	2
7-ZIP	1
WEBP, DOC, MP3, MP4, MOV, WAV, AVI, RAR, TAR	0
Sum	**87**

(b) Results for Ethereum.

File Type	Total	Embed.	Injected
PNG	275	50	225
WEBP	275	0	275
JPEG	124	72	52
7-ZIP	68	68	0
GIF	8	6	2
ZIP	4	4	0
MP3	3	1	2
PDF	2	2	0
DOC	1	1	0
MP4, MOV, WAV, AVI, RAR, TAR	0	0	0
Sum	**760**	**204**	**556**

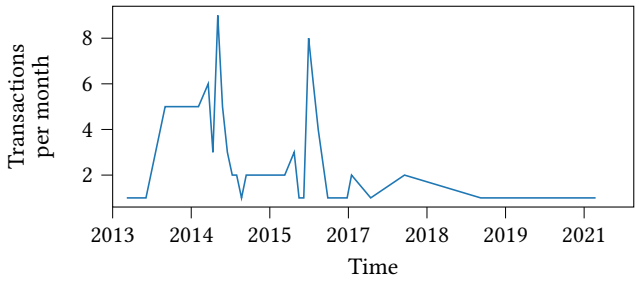
Table 5: Quantitative results of viewable files.

	Occurrences		
Category	**Total**	**Bitcoin**	**Ethereum**
Portraits	18	6	17
Memes	16	4	12
Crypto Related	11	4	7
Erotics	10	3	7
Family/Group Photos	7	3	4
Text as Image	8	0	8
Cats	5	1	4
Explicit Pornography	3	1	2

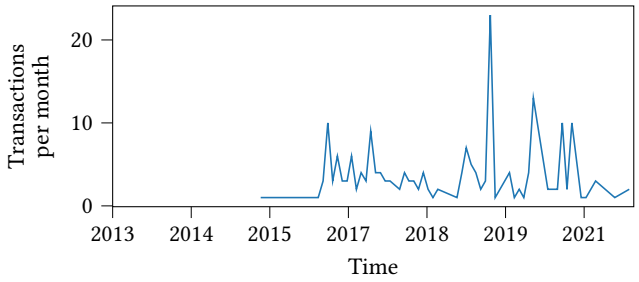
Table 6: Categorization of a selection of images found in the analysis (not counting duplicates).

- a manual on Adobe Acrobat PDF files
- a white page

The only *DOC file* we found contained 3 pages of Russian text, beginning with the headline *"So, you're reading this text when I've been dead for, I think, at least a few centuries."* (translated), signed January 23, 2019. The document also referenced a transaction on the blockchain with a 7-ZIP archive, which contained a text file with various links to images, videos, audio files, and other text



(a) Results for Bitcoin.



(b) Results for Ethereum (excluding files in smart contract calls).

Figure 3: Frequency of transactions with embedded files over time.

documents. Most of them were captured by the *Internet Archive*[11] and basically contained either poetic or philosophical statements.

Even though the majority of found archives was password protected, we successfully decompressed 6 *ZIP archives*:

- configuration files for the music-related software *Ivory*
- a digital certificate in DER format
- a static HTML website which presents itself as a proof of concept for an honors thesis at Albion College, presumably about etching content onto the blockchain
- a static HTML website to apparently find and play songs stored on the Ethereum blockchain using JavaScript
- the C# source code for the program *minicryptowallet*
- a static HTML website with text mostly in Russian which shares great similarity to the findings of the previously described DOC file

6 DISCUSSION

In this section we dive deeper into the key results of our analyses and discuss their implications.

6.1 BigQuery and Our Method

During the development of our analyzer, BigQuery has proven to be a robust, easy-to-use, and flexible platform to scrape structured data from a blockchain; results were returned within just a few minutes or a few hours depending on the analysis. BigQuery supports a very powerful set of SQL functions. Though, further methods are necessary to process and analyze the results; this especially accounts for

[11] https://web.archive.org

the file results. We also encountered limitations with subqueries and regular expressions. BigQuery restricts a regular expression pattern to have no more than one capturing group which make effective pattern matching, e.g., with URLs, difficult. It nevertheless enables great first filtering of the data that can afterwards be validated and processed by other programs.

6.2 Quality of Results

For the following discussion of our findings, we point out that our method did not find (or evaluate) every arbitrary content on Bitcoin or Ethereum. Additionally, many of our specific findings are likely to be corrupted, in the sense that they contain additional artifact or represent incomplete blobs of content. One reason lies in the fact that we never combine data spread across multiple transactions. Especially bigger files are likely to be spread over multiple transactions, however. This becomes very apparent with cut image files which we encountered with a significant number of findings. We observe the same things for text results in Bitcoin, where we sometimes saw fragments of JSON objects. Additionally, we suspect that many of the very short texts we detected, especially on Bitcoin, might be false positives. Those are naturally very hard to distinguish from noise. We compared our results to the numbers of detected transactions per low-level insertion methods in [16] and confirm similar results with our method (for the respective time frame). Quantitatively, we were also able to replicate the file results (disregarding files found by the service detectors).

6.3 How Content Storage Possibilities are Used

The text analysis has shown that the input bytes field in transactions was mainly used as a way to store non-financial information using the Ethereum protocol, but we also identified a significant amount of organizations that utilize this field to establish their own protocols (e.g., to transfer JSON objects).

The files analysis has shown a number of results that originate from people just being enthusiastic about the possibility to persist arbitrary content in the blockchain. Our qualitative observation revealed time capsuled contents, with the intention to be discovered much later in the future or to persist them irrevocably (e.g., official certificates).

Both analyses have shown a lot of duplicate content and specifically for the file results it was evident that few individuals are responsible for a large parts of insertions. A large part of the file contents and even a lot of text messages (based on the review of a sample of results) are related to cryptocurrencies. It is even more obvious in our qualitative evaluation of a sample of the found URLs, where cryptocurrency-related content constitutes the largest category. The popularity of cryptocurrency-related content does not come as a surprise since the insertion of arbitrary content (outside of smart contracts) requires greater technical engagement with the system and methods that are not as accessible to regular users.

In almost every quantitative assessment of results, Ethereum scored significantly higher than Bitcoin, even though Ethereum got introduced eight years later than Bitcoin. We reason this by arbitrary content insertions being cheaper and generally more convenient on Ethereum than on Bitcoin. Up until mid-2020, the transaction fees in Ethereum used to be consistently lower, most of the time only a fraction of the fees charged for transactions on Bitcoin [4, 5]. A similar correlation can be inferred from Figure 1. Furthermore, transactions in Ethereum are far more flexible and it has not been until 2014 that Bitcoin introduced a more accessible solution to this with `OP_RETURN` [8]. But even with this solution, the space in a single transaction is relatively small compared to Ethereum's input field which was designed to carry the byte code for complex smart contract logic. This can become a constraint when uploading larger files, forcing users to split the bytes of a file to multiple transactions. This ultimately makes Ethereum the more attractive choice, especially for file insertions.

6.4 Legal Assessment of Results

In our efforts to make a statement on the legal risks resulting from injected contents, we have paid special attention to URLs detected in our text analysis. While we were not able to review every single URL, due to the high quantity, we were able to link at least one finding (in each blockchain) to former backup URLs to child pornography websites. The file analysis has also led to some sensitive or offensive contents (e.g., a swastika and explicit pornography). We found one image that depicts, in low resolution, mild nudity of a woman that could potentially be a minor. All in all, very few of our findings come close to being considered "illegal" by Western countries' standards. We also did not find any obvious copyright violations. We did, however, find images offending leaders of countries such as Russia, China, and North Korea, the possession and redistribution of which could be frowned upon within the respective countries. Likewise, the possession and redistribution of pornographic content and arbitrary religious texts is also not universally accepted or even legal in all jurisdictions.

A remaining source of potential legal risks lies in the large amount on privacy-relevant data that is stored on both investigated blockchains, including personal photographs and personal details such as birth dates. To which extent these occurrences could lead to clashes with existing data protection legislation and personality rights remains an open question. Recall that erasing data from blockchains, which can be mandated by a legal framework such as the EU's GDPR [11], poses a significant challenge [2, 9, 12, 13, 21].

7 CONCLUSION

We presented a novel approach for detecting and classifying arbitrary non-financial content on public blockchains using the Google Cloud service BigQuery, and applied it to the Bitcoin and Ethereum blockchains. We discovered arbitrary contents of various types; ranging from peer-to-peer communication, advertisement, philosophical messages, to various forms of structured data, presumably triggering external logic. Finally, we also confirm former results indicating that the possibility to post arbitrary data has been abused to post illegal or objectionable content, such as images or URLs to externally hosted (e.g., via onion services) problematic content. Furthermore, we have evaluated BigQuery as a means to perform analyses of blockchain data. While we have identified technical limitations, we still assess BigQuery to be a useful and appropriate base for further, more complex analyses.

REFERENCES

[1] Lennart Ante. 2021. Non-fungible token (NFT) markets on the Ethereum blockchain: Temporal development, cointegration and interrelations. *SSRN, BRL Working Paper Series* 22 (2021).

[2] Giuseppe Ateniese, Bernardo Magri, Daniele Venturi, and Ewerton Andrade. 2017. Redactable Blockchain–or–Rewriting History in Bitcoin and Friends. In *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*. IEEE, 111–126.

[3] Juan Benet. 2014. IPFS - Content Addressed, Versioned, P2P File System. *CoRR* abs/2011.00874 (2014). http://arxiv.org/abs/1407.3561

[4] BitInfoCharts. 2022. Bitcoin Avg. Transaction Fee Chart. https://bitinfocharts.com/en/comparison/bitcoin-transactionfees.html#alltime (accessed: 2022-02-03).

[5] BitInfoCharts. 2022. Ethereum Avg. Transaction Fee Chart. https://bitinfocharts.com/en/comparison/ethereum-transactionfees.html#alltime (accessed: 2022-02-03).

[6] Vitalik Buterin. 2013. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. (2013). https://ethereum.org/en/whitepaper/

[7] CoinMarketCap. 2013. Today's Cryptocurrency Prices by Market Cap. https://coinmarketcap.com/ (accessed: 2021-12-02).

[8] Bitcoin Core. 2009. Bitcoin Core version 0.9.0 released. https://bitcoin.org/en/release/v0.9.0#opreturn-and-data-in-the-block-chain (accessed: 2022-02-03).

[9] Dominic Deuber, Bernardo Magri, and Sri Aravinda Krishnan Thyagarajan. 2019. Redactable Blockchain in the Permissionless Setting. *CoRR* abs/1901.03206 (2019). arXiv:1901.03206 http://arxiv.org/abs/1901.03206

[10] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*. USENIX, 303–320.

[11] European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union* L119 (4 May 2016), 1–88. http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC

[12] Martin Florian, Sebastian Henningsen, Sophie Beaucamp, and Björn Scheuermann. 2019. Erasing data from blockchain nodes. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 367–376.

[13] Srdjan Capkun Ivan Puddu, Alexandra Dmitrienko. 2017. μchain: How to Forget without Hard Forks. Cryptology ePrint Archive, Report 2017/106. https://eprint.iacr.org/2017/106

[14] Christoph Jentzsch. 2016. Decentralized Autonomous Organization to Automate Governance. *White paper* (2016).

[15] Roman Matzutt, Martin Henze, Jan Henrik Ziegeldorf, Jens Hiller, and Klaus Wehrle. 2018. Thwarting Unwanted Blockchain Content Insertion. In *2018 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 364–370.

[16] Roman Matzutt, Jens Hiller, Martin Henze, Jan Henrik Ziegeldorf, Dirk Müllmann, Oliver Hohlfeld, and Klaus Wehrle. 2018. A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin. In *Financial Cryptography (Lecture Notes in Computer Science)*. Springer.

[17] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

[18] Alexander Pfeiffer, Simone Kriglstein, Thomas Wernbacher, and Stephen Bezzina. 2020. Blockchain technologies and social media: A snapshot. In *8th European Conference on Social Media*. Academic Conferences and publishing limited, 196.

[19] Max Raskin and David Yermack. 2018. Digital currencies, decentralized ledgers and the future of central banking. In *Research handbook on central banking*. Edward Elgar Publishing.

[20] Nick Szabo. 1994. Smart Contracts. https://web.archive.org/web/20160323035617/http://szabo.best.vwh.net/smart.contracts.html (accessed: 2022-01-28).

[21] Sri Aravinda Krishnan Thyagarajan, Adithya Bhat, Bernardo Magri, Daniel Tschudi, and Aniket Kate. 2021. Reparo: Publicly verifiable layer to repair blockchains. In *International Conference on Financial Cryptography and Data Security*. Springer, 37–56.

[22] Alin Tomescu and Srinivas Devadas. 2017. Catena: Efficient non-equivocation via Bitcoin. In *2017 38th IEEE Symposium on Security and Privacy (SP)*. IEEE, 393–409.

[23] Florian Tschorsch and Björn Scheuermann. 2016. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys & Tutorials* 18, 3 (2016), 2084–2123.

[24] Gavin Wood. 2021. Ethereum: A Secure Decentralised Generalised Transaction Ledger, Berlin Version FABEF25. (2021).

A SELECTION OF FINDINGS FROM THE TEXT ANALYSIS

- "EW You can now use app.remembr.io to send any kind of message on blockchain"

  (Found on Bitcoin. Hash: 095f90b1414ccc90d0615a447b0a82847e7aa1bb70807d82a613b48a3043ec49. Aug-31-2015 01:19:22 PM +UTC.)

- "L176,Obama spent about $65,000 of the tax-payers money flying in pizza/dogs"

  (Found on Bitcoin. Hash: d7c7e871f6502179761122d7fc7d0ed7fa367cd2cf38c46c8d177a119d343437. Jan-04-2017 02:11:53 PM +UTC.)

- "Happy Birthday Joachim! With love from the Jolocom team!"

  (Found on Bitcoin. Hash: d00dab66b0ef0142dbfb51f51beac3eb33fc5dd7635539a5deccdffdf109d019. Jan-21-2018 10:35:49 AM +UTC.)

- "hello, guy, can you send my 5.1558763eth back ? that is all my currency in the crypto world, that money I am ready to get married, and I thought I could earn double this time, but now I couldn't get it back. i don't know why this happen. If you send it to me back, i would really appreciate you!!!"

  (Found on Ethereum. Hash: 0x884ea1daf9888e5c1aa9d12737bc90e186eb08a6ced9fce9595fbe4878b645c0. Jun-15-2021 02:04:16 PM +UTC.)

- "Madeleine, I love you so much. My love for you is eternal like this message. Happy 6 Months!"

  (Found on Ethereum. Hash: 0xcfb426dcbf8c399d5d9f6f18f8abb60dfe77f42bae51f3cc46425260016c1107. Aug-18-2018 06:56:58 AM +UTC.)

- "love makes us fragile, but it is still everyone's greatest wish. The weaker we are, the more we crave it."

  (Found on Ethereum. Hash: 0x35afd31eb8cb974405898364c11b86057aa9674a714aded4893419999ff8a649. Dec-31-2018 09:49:00 PM +UTC.)

- "Hi, Bro. I admire what you're doing. Salute 0xE0E70fDF0D44DD231C1bc522F2885aD85F43b970 This is my address. I hope you can tip me.Thank Bro"

  (Found on Ethereum. Hash: 0x1deed99febea575059825d5f98fc005846c0d5688598648ff4b940edcac8fe6f. Aug-10-2021 04:10:27 PM +UTC.)

- "0xI want to return $100000 to you next year but you threaten me so I won't pay you anymore"

  (Found on Ethereum. Hash: 0x96ec1b4c820a32d648a8251f494985f178532945cc60e484477ba421cfae11ae. Dec-05-2020 04:28:24 PM +UTC.)

- "AK47 payout for deposit ETH - payout 17 of 50. Thank you!"

  (Found on Ethereum. Hash: 0x593cabe13352f5d3f9eda42264dae7c79d97906b95b9bc05a23f63826d01be12. May-04-2018 11:17:20 AM +UTC.)

- "Yang , happy birthday! Welcome to 21 club!"

  (Found on Ethereum. Hash: 0x6e557b1d3c6ff17b44bc213064e2980f4d7c819f14a3fca1fcd58f472bf8c5af. Jul-10-2021 11:49:07 PM +UTC.)

- "If you read this you are a real crypto lover, welcome in the (eternal) KRYPTOSPHERE!"

  (Found on Ethereum. Hash: 0x9d377734ffcb51efe1d7a828c4c8ca9e7bfe70f4bfcfb6a40380f7cff3175c70. Sep-23-2021 04:27:27 PM +UTC.)