The Starling Framework

+ Introduction to Cryptographic Primitives

April 4, 2025 // EE292J

Benedict Lau

CTO

USC Shoah Foundation | Stanford ENGINEERING
Center for Blockchain Research
Compression Forum

Outline

1. Cryptography building blocks

- Cryptographic hashes
- Digital signatures

Outline

2. Relevant concepts

- Integrity data vs. asset data
- Data registration
  - Immutable public ledgers
  - Merkle proofs
  - Selective disclosure
- Decentralized preservation
  - Content addressing
- Verifiable computing
  - Consensus on blockchains
  - ZK SNARK proofs

Outline

3. Building towards authenticity

- Starling pipeline: Capture · Store · Verify
- Authenticated attributes

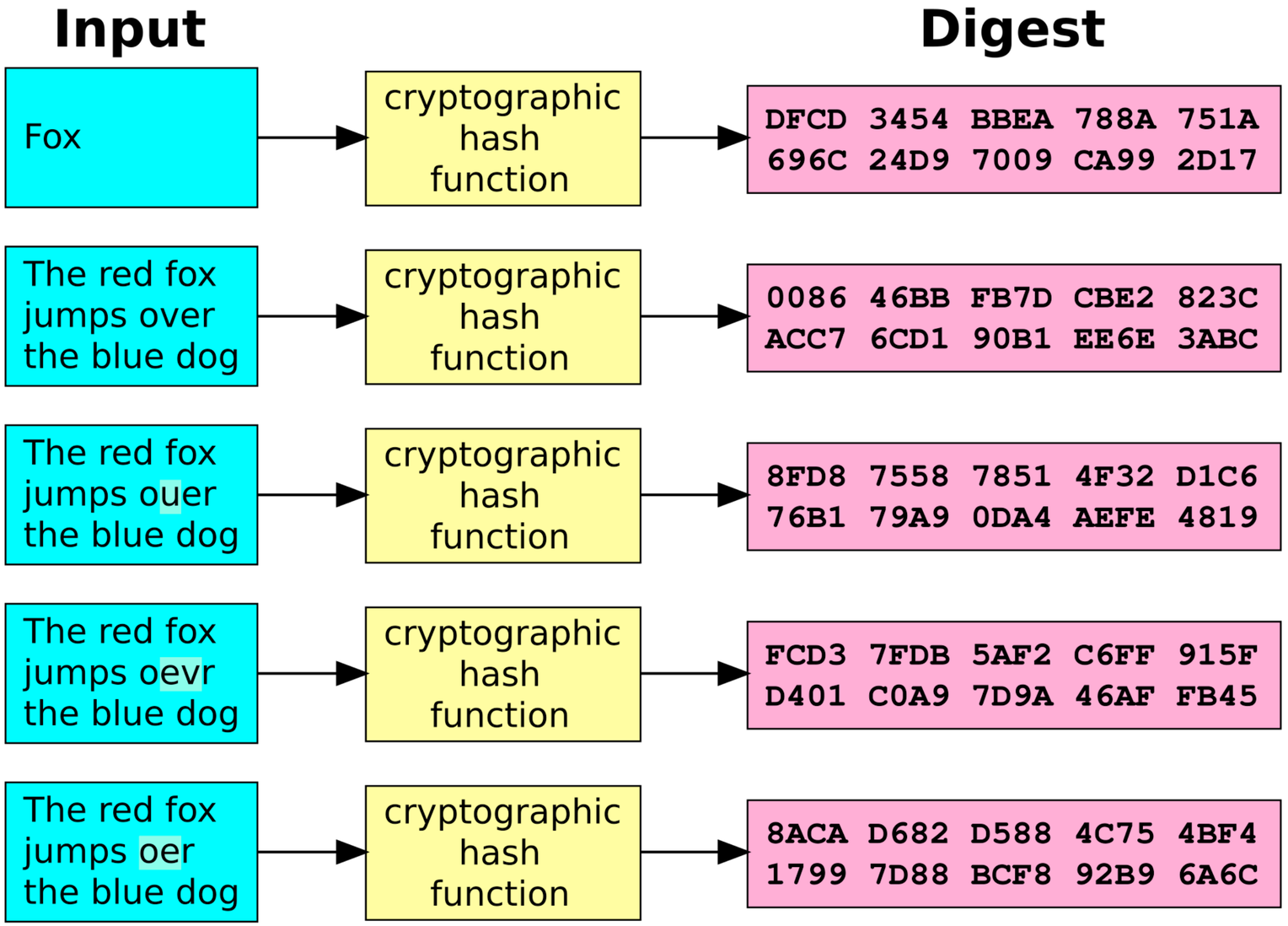
[*]starling lab

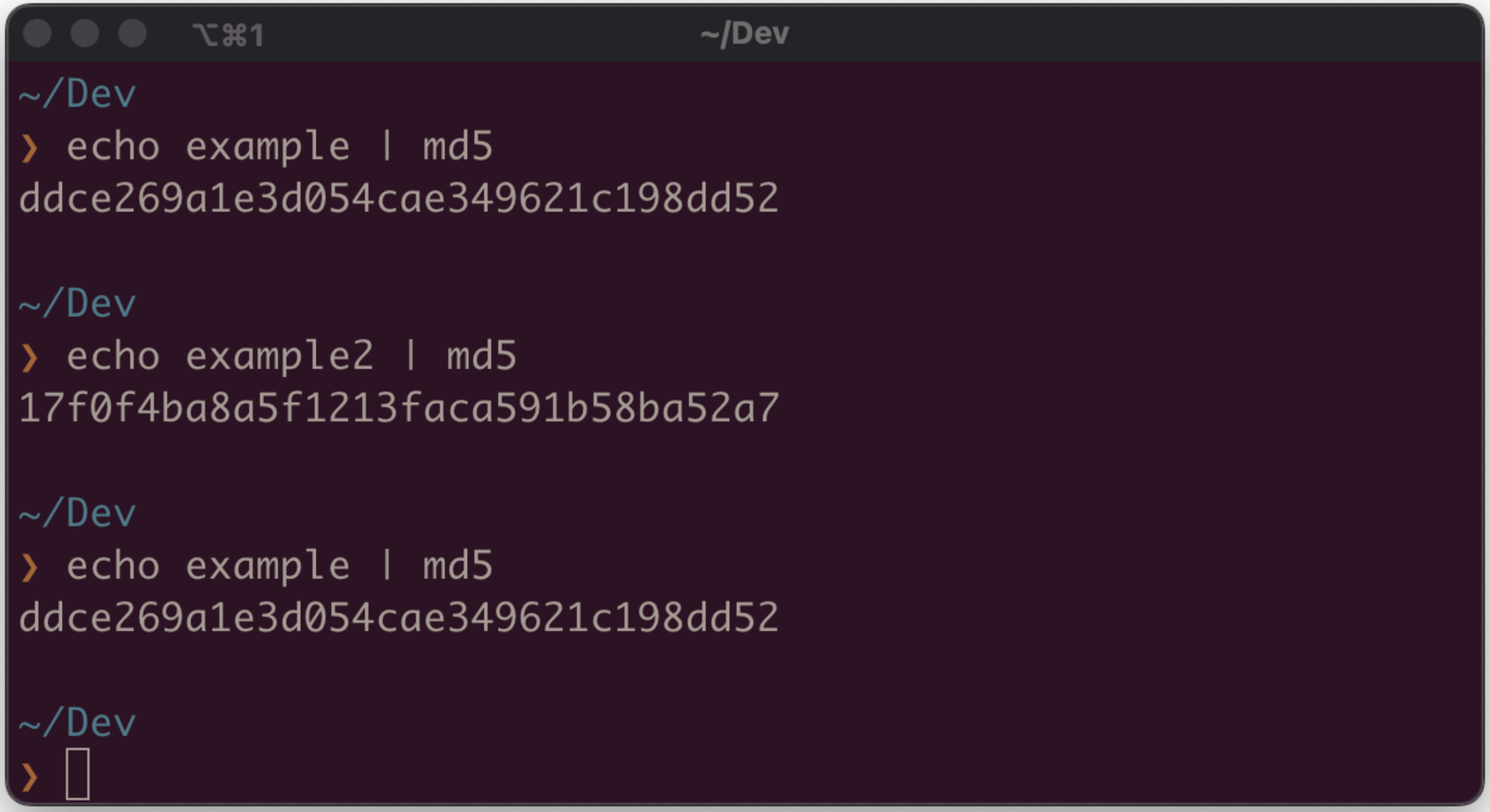
1. Cryptography building blocks

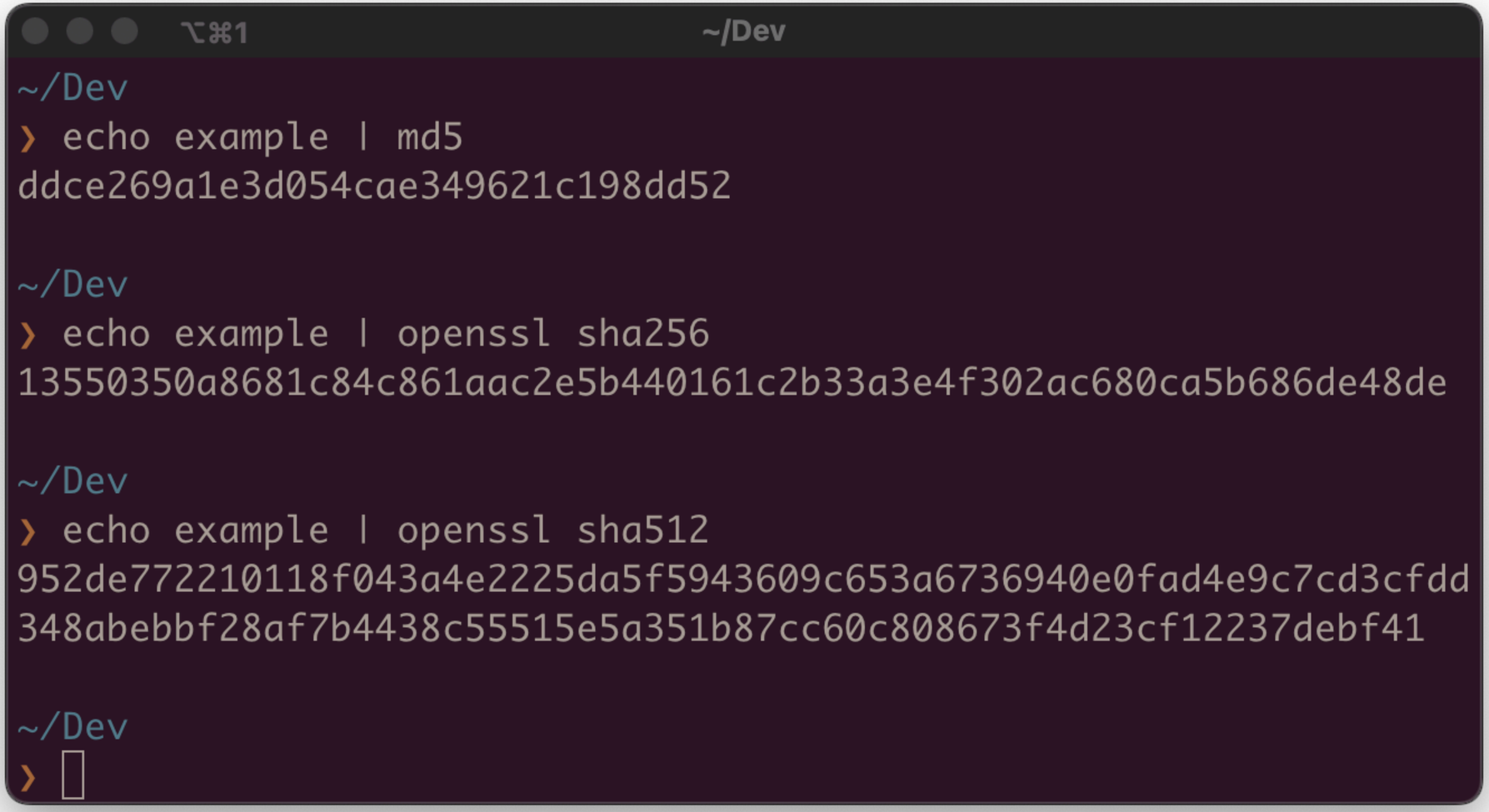
Cryptographic hashes

digests

fingerprints


Input
Digest
Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog
cryptographic hash function
FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog
cryptographic hash function
8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C


⌥⌘1
~/Dev
~/Dev
❯ echo example | md5
ddce269a1e3d054cae349621c198dd52
~/Dev
❯ echo example2 | md5
17f0f4ba8a5f1213faca591b58ba52a7
~/Dev
❯ echo example | md5
ddce269a1e3d054cae349621c198dd52
~/Dev
❯


⌥⌘1
~/Dev
~/Dev
❯ echo example | md5
ddce269a1e3d054cae349621c198dd52
~/Dev
❯ echo example | openssl sha256
13550350a8681c84c861aac2e5b440161c2b33a3e4f302ac680ca5b686de48de
~/Dev
❯ echo example | openssl sha512
952de772210118f043a4e2225da5f5943609c653a6736940e0fad4e9c7cd3cfdd
348abebbf28af7b4438c55515e5a351b87cc60c808673f4d23cf12237debf41
~/Dev
❯

	Digest size	Number of possible digests
MD5	**128 bits** (16 bytes = 32 hex characters)	2^{128}
SHA-256	**256 bits** (32 bytes = 64 hex characters)	2^{256}
SHA-512	**512 bits** (64 bytes = 128 hex characters)	2^{512}

Input

Digest

cryptographic hash function

```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

```
~/Dev
❯ md5 ~/Documents/cat.jpg
MD5 (/Users/benedict/Documents/cat.jpg) = 9d88978e7b2cae78778235def098540
8

~/Dev
❯
```

Input

Digest

cryptographic hash function

```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

cryptographic hash function

```
5e5b 948a de9c 3d41
de66 17b6 8f76 9e55
```

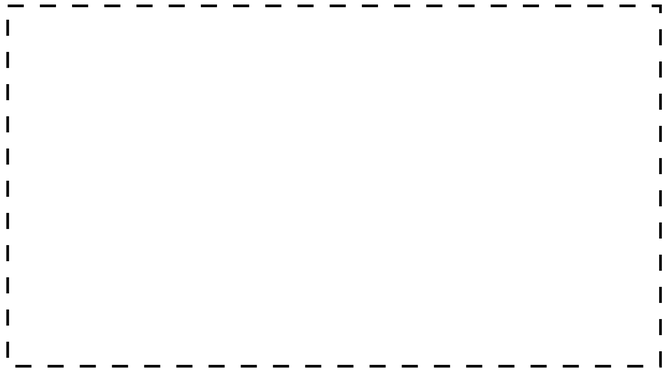
[*]starling lab

Input

Digest

```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

Input



Digest

```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

Input

Digest

5e5b 948a de9c 3d41
de66 17b6 8f76 9e55

9d88 978e 7b2c ae78
7782 35de f098 5408

Input

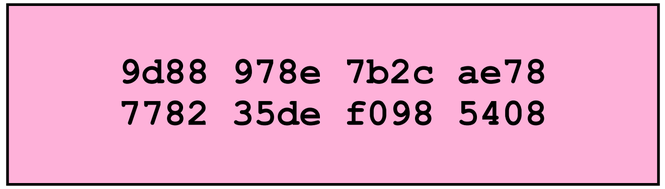
```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

✅

Digest

```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

Tamper-evident integrity



Can two different inputs have the same digest?

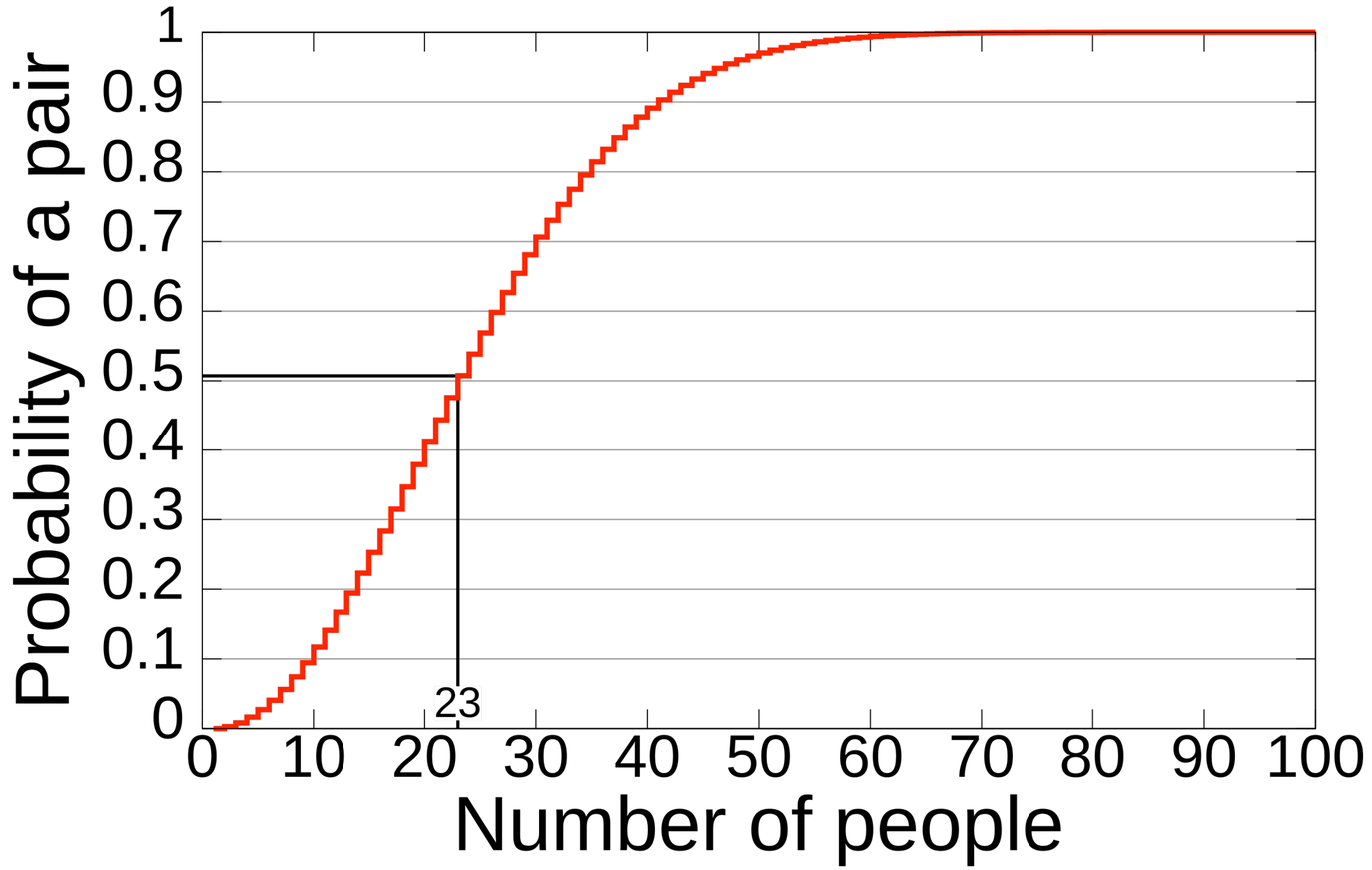
Birthday problem

Joey
Rick
Leo
Jack
Max
Sara
Mandy
Geoff
Mike
Nick

January 1
January 2
January 3
:
December 31

365 *possible days*

Birthday problem



	Digest size	Number of possible digests	Number of digests for probable collision
Birthday problem		$2^{8.5}$ (365 days)	$2^{4.5}$ (23 days)
MD5	**128 bits** (16 bytes = 32 hex characters)	2^{128}	2^{64}
SHA-256	**256 bits** (32 bytes = 64 hex characters)	2^{256}	2^{128}
SHA-512	**512 bits** (64 bytes = 128 hex characters)	2^{512}	2^{256}

It will take a *very, very, very,* $\text{very}^{\text{very}}$ long time to find one. For comparison, as of January 2015, Bitcoin was computing 300 quadrillion SHA-256 hashes per second. That's 300×10^{15} hashes per second.

Let's say you were trying to perform a collision attack and would "only" need to calculate 2^{128} hashes. At the rate Bitcoin is going, it would take them

$2^{128}/(300 \times 10^{15} \cdot 86400 \cdot 365.25) \approx 3.6 \times 10^{13}$ years. In comparison, our universe is only about 13.7×10^{9} years old. Brute-force guessing is not a practical option.

—*https://crypto.stackexchange.com/a/47810*

This number has grown ~3,000x since 2015

It will take a *very, very, very,* $\text{very}^{\text{very}}$ long time to find one. For comparison, as of January 2015, Bitcoin was computing 300 quadrillion SHA-256 hashes per second. That's 300×10^{15} hashes per second.

Let's say you were trying to perform a collision attack and would "only" need to calculate 2^{128} hashes. At the rate Bitcoin is going, it would take them

$2^{128}/(300 \times 10^{15} \cdot 86400 \cdot 365.25) \approx 3.6 \times 10^{13}$ years. In comparison, our universe is only about 13.7×10^{9} years old. Brute-force guessing is not a practical option.

—https://crypto.stackexchange.com/a/47810

Can I find the input if I only have the digest?

Input

Digest

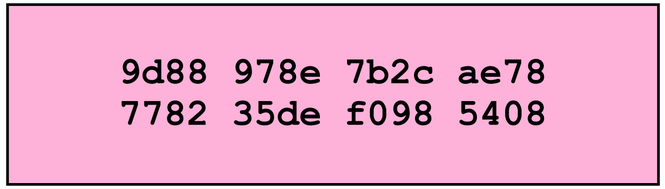
```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

This is called the **pre-image** of the digest

pre-image

```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

Privacy-preserving* identifier



**Disclaimer: One may gain insight about a pre-image by correlating two pieces of metadata describing the same digest*

Perceptual hashing is the use of a fingerprinting algorithm that produces a snippet, hash, or fingerprint of various forms of multimedia. A perceptual hash is a type of locality-sensitive hash, which is analogous if features of the multimedia are similar. **This is in contrast to cryptographic hashing, which relies on the avalanche effect of a small change in input value creating a drastic change in output value.** Perceptual hash functions are widely used in finding cases of online copyright infringement as well as in digital forensics because of the ability to have a correlation between hashes so similar data can be found (for instance with a differing watermark).

—*Wikipedia*

Digital signatures

Establishes **verifiable attribution**

Based on **public-private key cryptography**
e.g. RSA (Rivest-Shamir-Adleman)

RSA private key

- Keep secret !!
- Use for signing claims

[*]starling lab

Signer

Signing a claim

Private key

Claim:
This is a picture of Miu

Signature

[*]starling lab

Verifier

Verifying a signature

Public key

Signature

Claim:
This is a picture of Miu

?

[*]starling lab

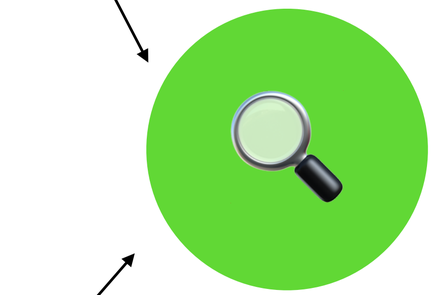
Verifier

Verifying a signature

Public key

Signature

Claim:
This is a picture of Miu



True

[*]starling lab

What does this prove?

True

This is a picture of Miu

according to the Signer

Verifier

Public key

Signature

Claim:
This is a picture of Miu

Tampered photo

False

[*]starling lab

Verifier

Public key

Signature

Claim:
This is a picture of Mao

Tampered claim

False

[*]starling lab

We can cryptographically verify that the data is untampered, but **do you trust the signer's claim?**

2. Relevant concepts

Integrity data vs. asset data

```
9d88 978e 7b2c ae78
7782 35de f098 5408
```

~100 bytes

Arbitrary size

Claim:

This is a picture of Miu

`367e 62d5 2dc7 7075`
`a9d0 1079 7f31 82f3`

Signer

Private key

Signature

Integrity data vs. asset data

`367e 62d5 2dc7 7075` `a9d0 1079 7f31 82f3`
Signer Public key
Signature

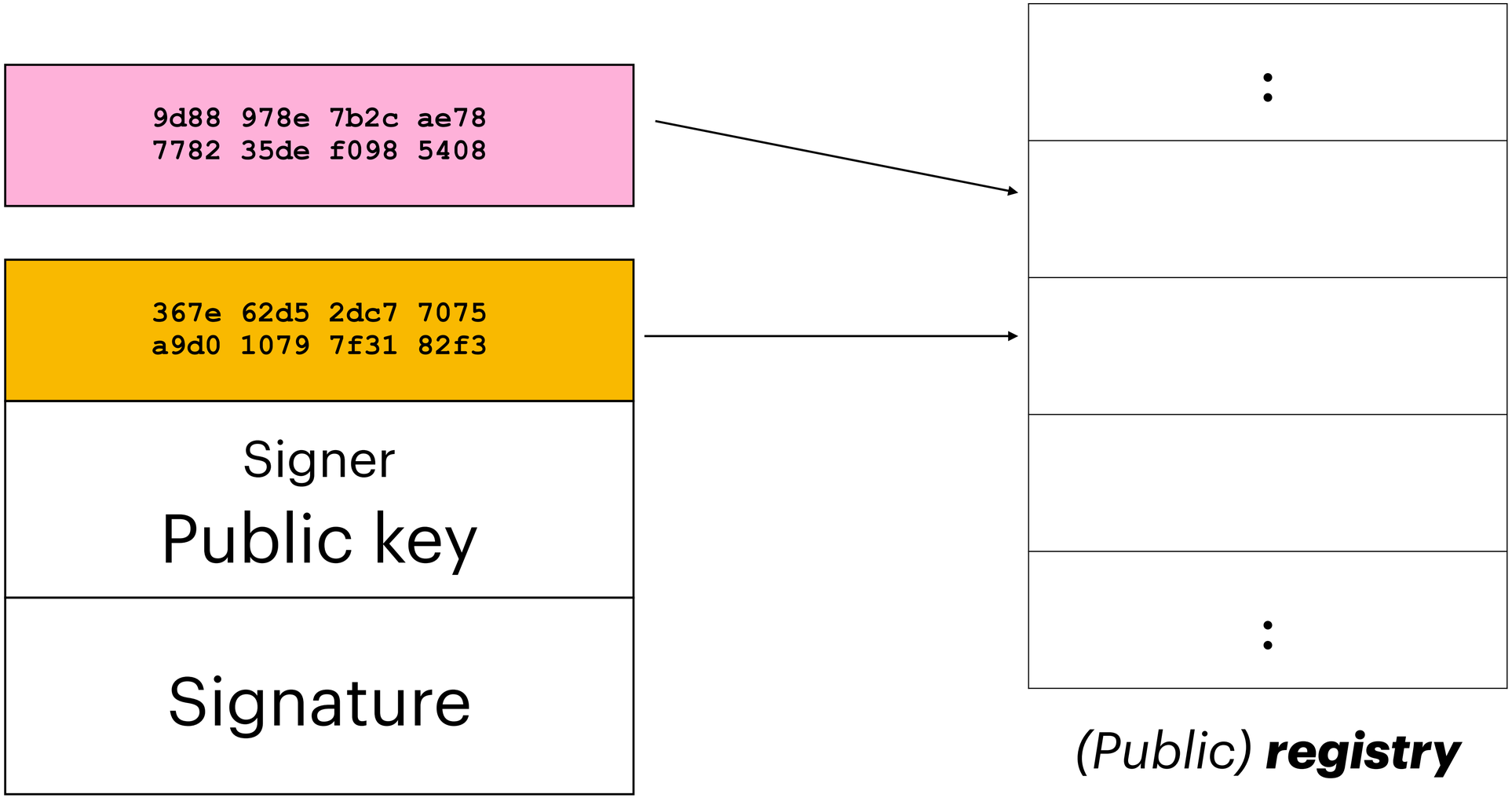
~1,000 bytes

Claim:

This is a picture of Miu

Arbitrary size

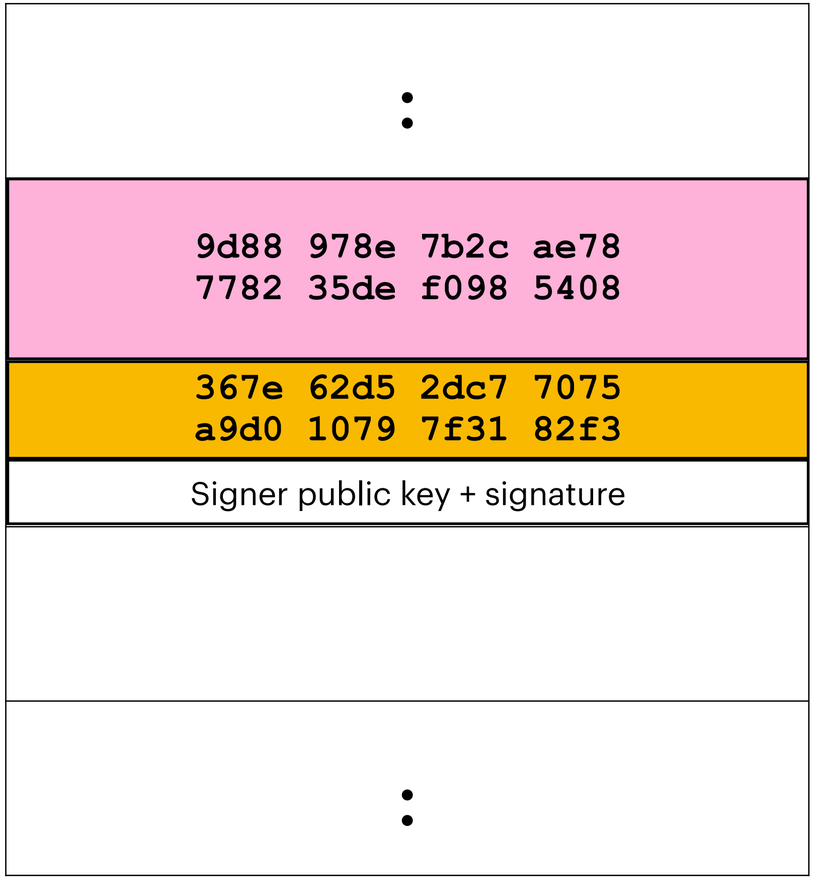
Data registration



Immutable public ledgers

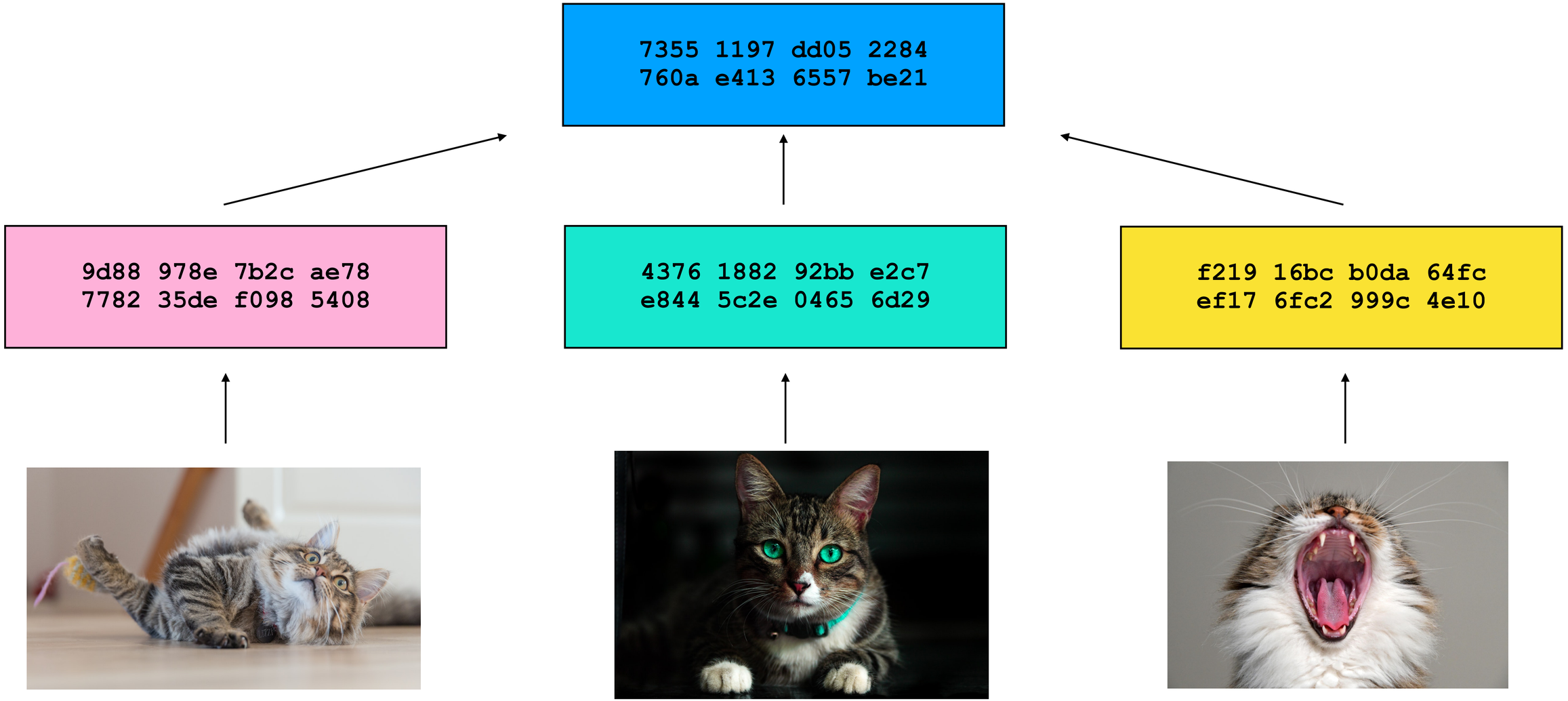
Since we cannot guess a pre-image, a registered digest is a ***proof of existence*** *of the asset at a particular point in time*

A publicly registered signature supports ***verifiable attribution*** *of claims*

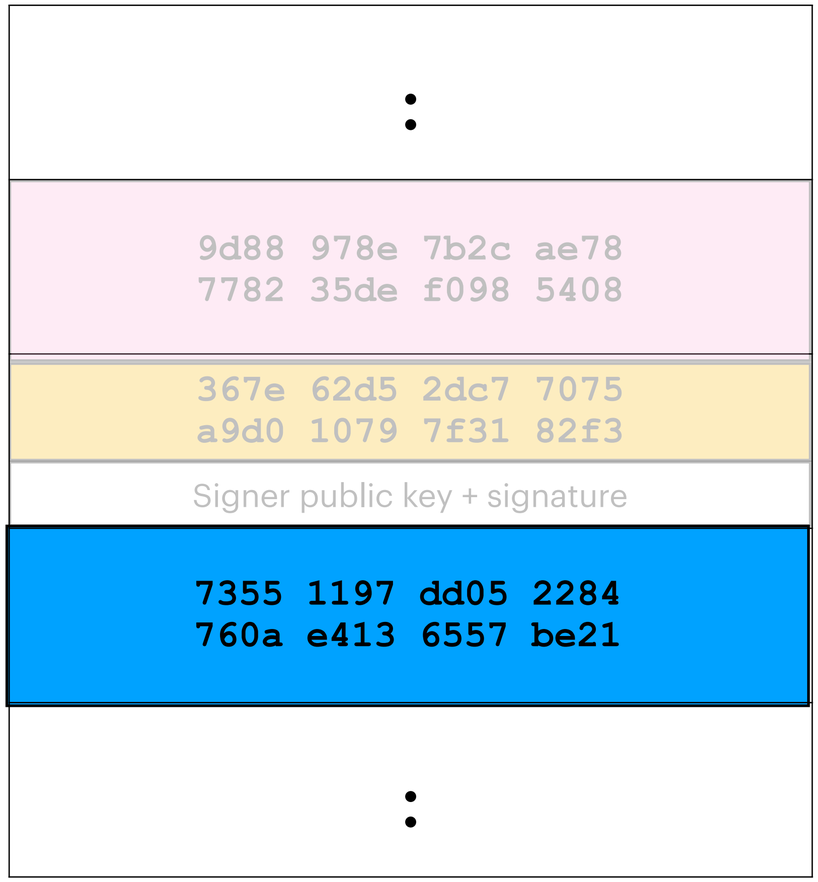


Merkle proofs

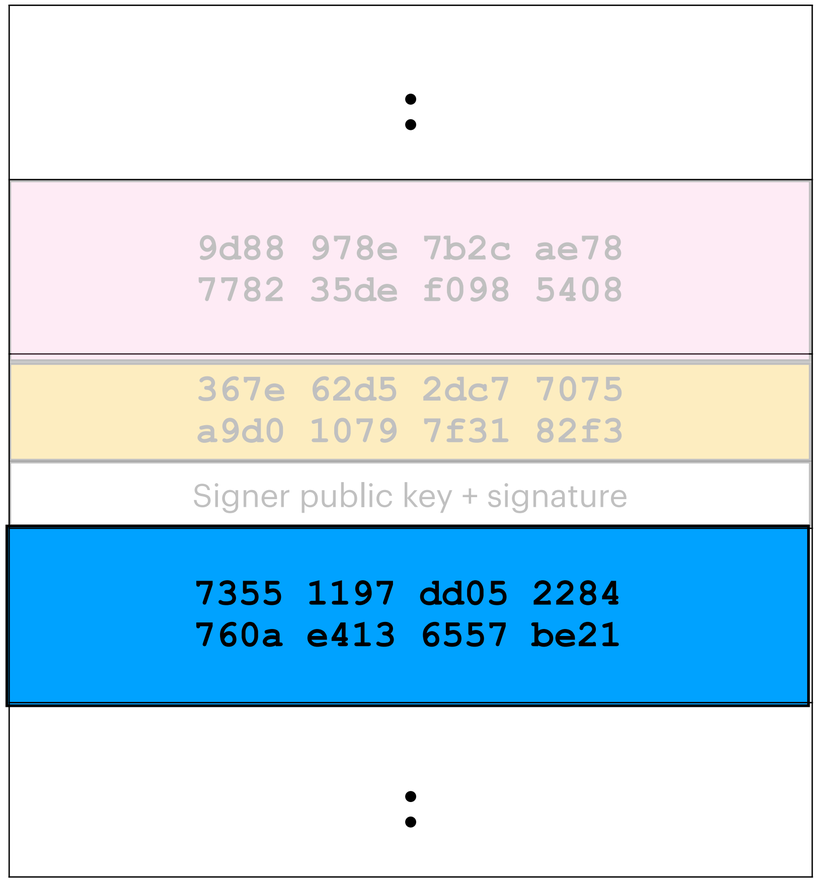
Merkle proofs



Merkle proofs



Merkle proofs



How do we prove that

is included in this digest?

Merkle proof

7355 1197 dd05 2284
760a e413 6557 be21

4376 1882 92bb e2c7
e844 5c2e 0465 6d29

f219 16bc b0da 64fc
ef17 6fc2 999c 4e10

[*]starling lab

Selective disclosure

7355 1197 dd05 2284
760a e413 6557 be21

4376 1882 92bb e2c7
e844 5c2e 0465 6d29

f219 16bc b0da 64fc
ef17 6fc2 999c 4e10

7355 1197 dd05 2284
760a e413 6557 be21

9d88 978e 7b2c ae78
7782 35de f098 5408

f219 16bc b0da 64fc
ef17 6fc2 999c 4e10

7355 1197 dd05 2284
760a e413 6557 be21

9d88 978e 7b2c ae78
7782 35de f098 5408

4376 1882 92bb e2c7
e844 5c2e 0465 6d29

Where do we store the actual asset?

Arbitrary size

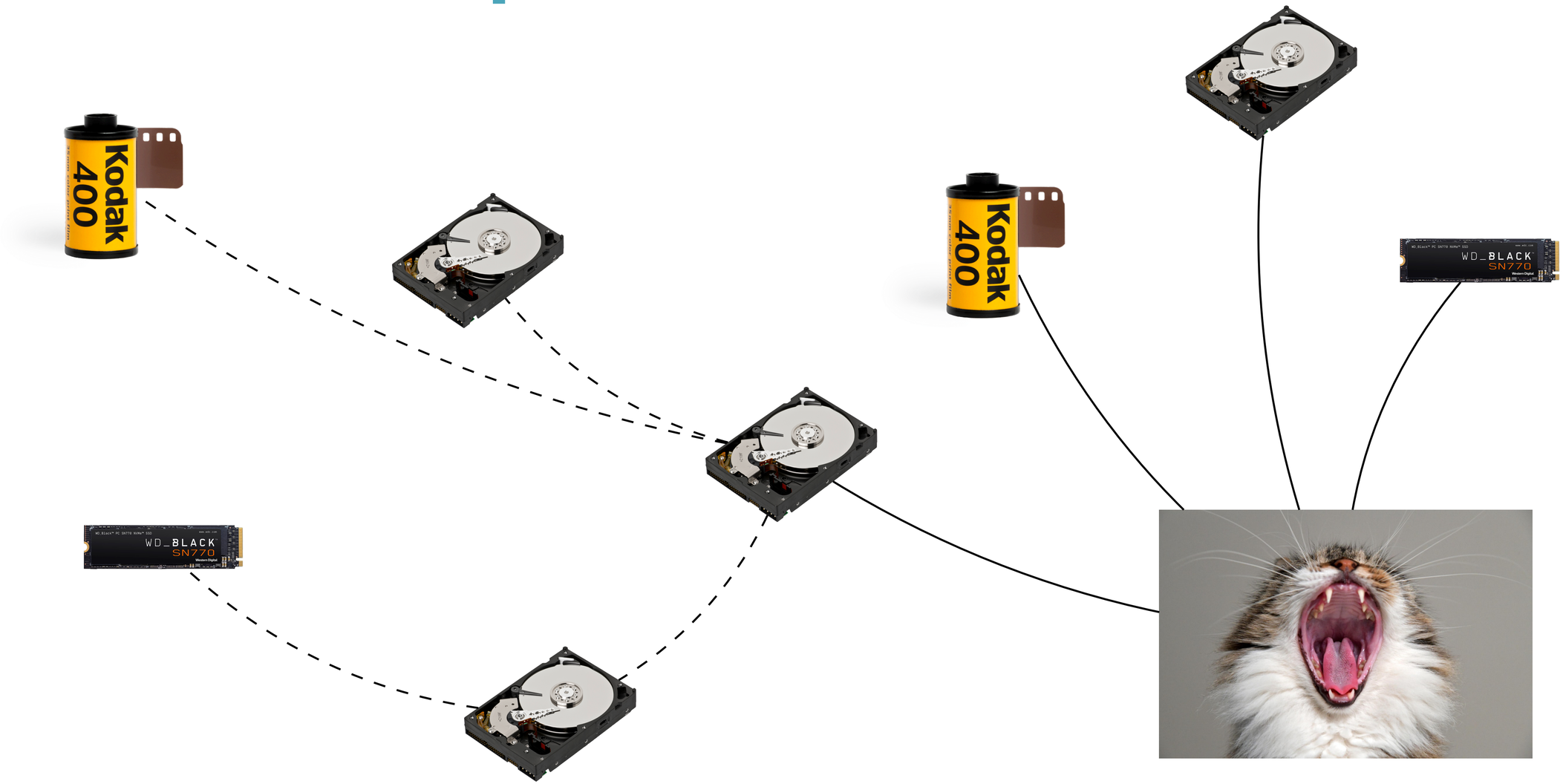
[*]starling lab

~70 years

~10 years

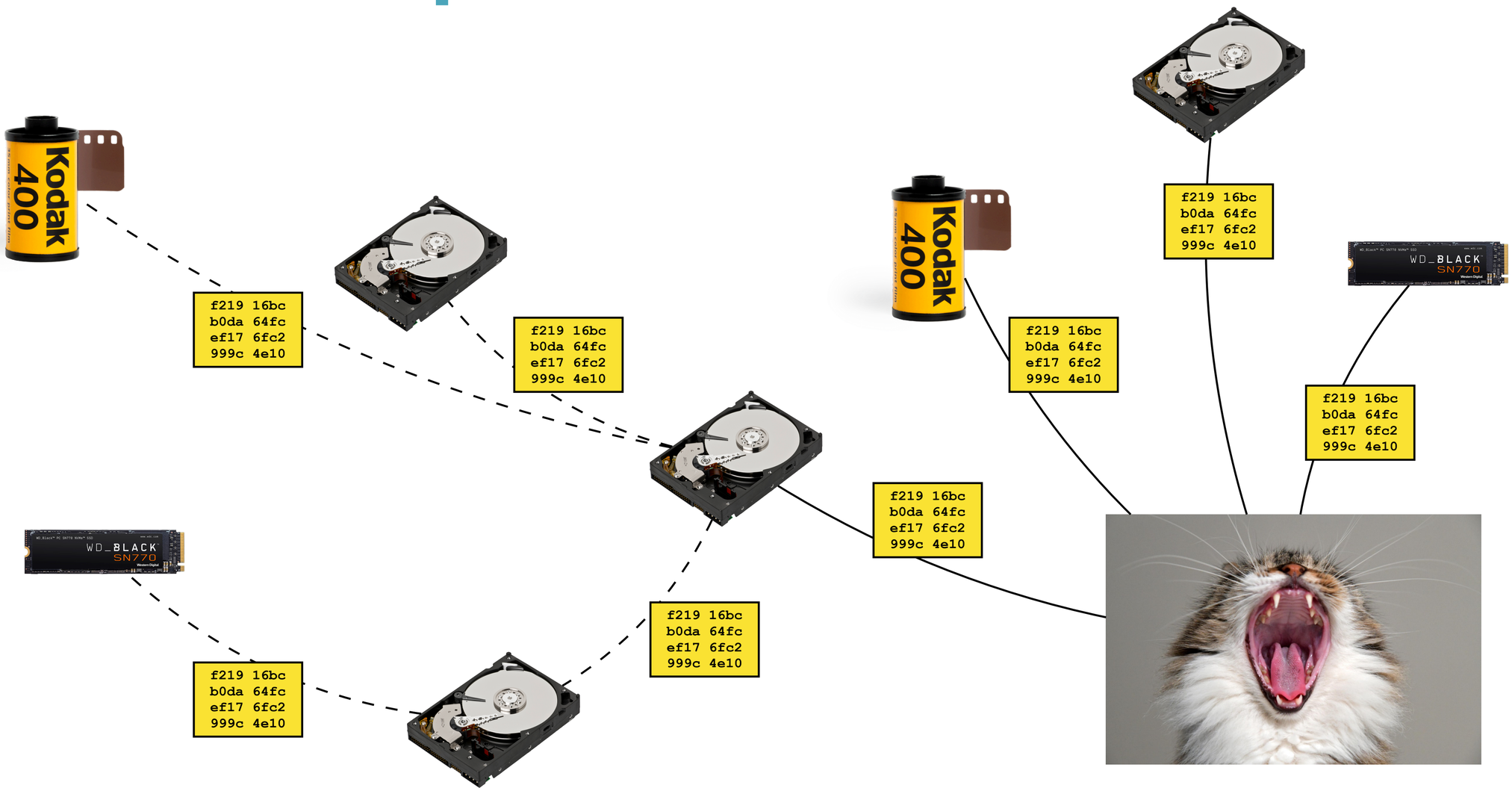
~5 years

Decentralized preservation



[*]starling lab

Decentralized preservation

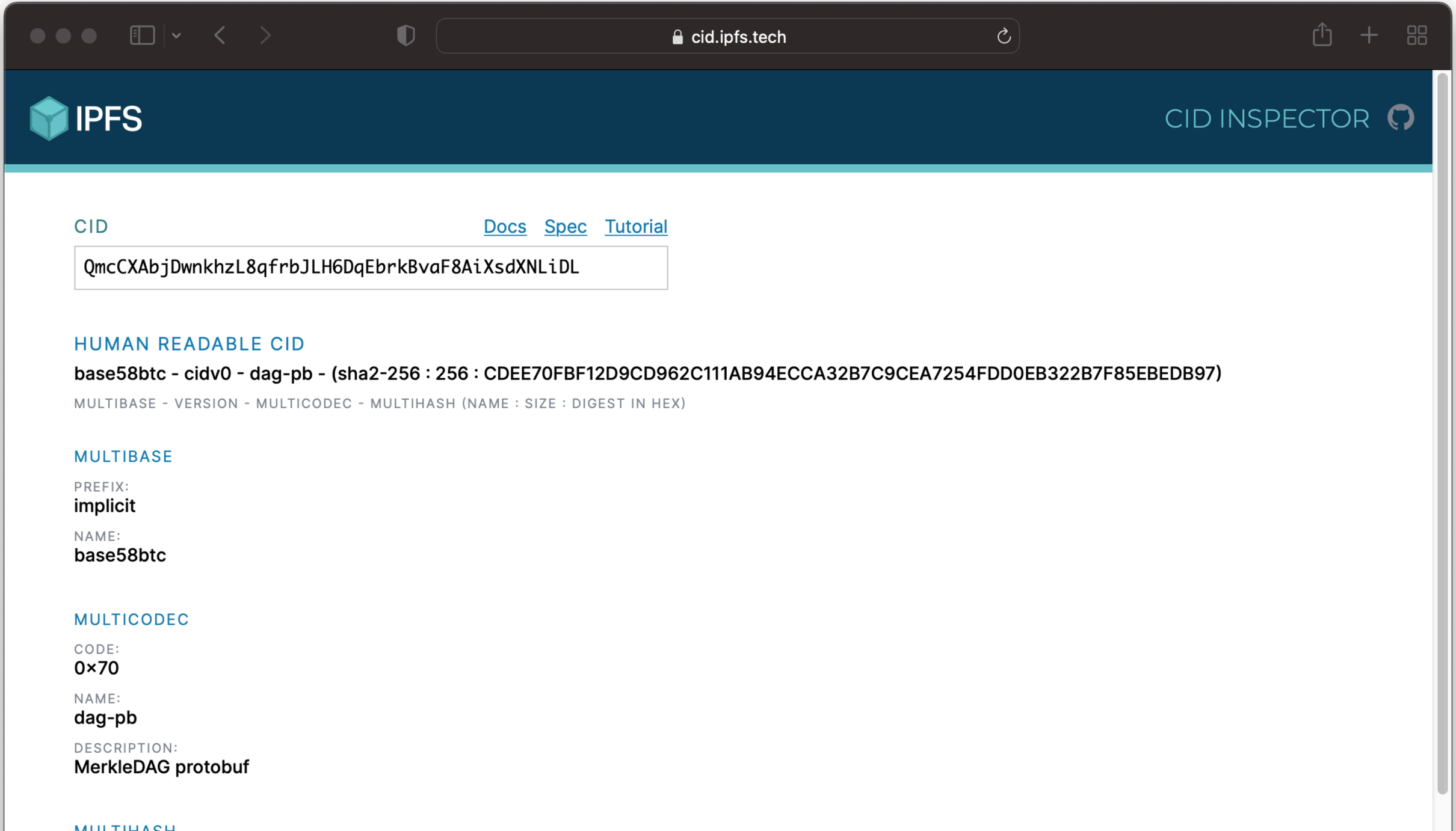


[*]starling lab

Content addressing

Host-address: find asset using host name + host-specific identifier
https://youtu.be/0X95Nybu2g8

Content-address: find asset using a digest-based content identifier (CID)
ipfs://QmcCXAbjDwnkhzL8qfrbJLH6DqEbrkBvaF8AiXsdXNLiDL


cid.ipfs.tech
IPFS
CID INSPECTOR
CID
Docs
Spec
Tutorial
QmcCXAbjDwnkhzL8qfrbJLH6DqEbrkBvaF8AiXsdXNLiDL
HUMAN READABLE CID
base58btc - cidv0 - dag-pb - (sha2-256 : 256 : CDEE70FBF12D9CD962C111AB94ECCA32B7C9CEA7254FDD0EB322B7F85EBEDB97)
MULTIBASE - VERSION - MULTICODEC - MULTIHASH (NAME : SIZE : DIGEST IN HEX)
MULTIBASE
PREFIX:
implicit
NAME:
base58btc
MULTICODEC
CODE:
0×70
NAME:
dag-pb
DESCRIPTION:
MerkleDAG protobuf

Who has
QmcC ... LiDL

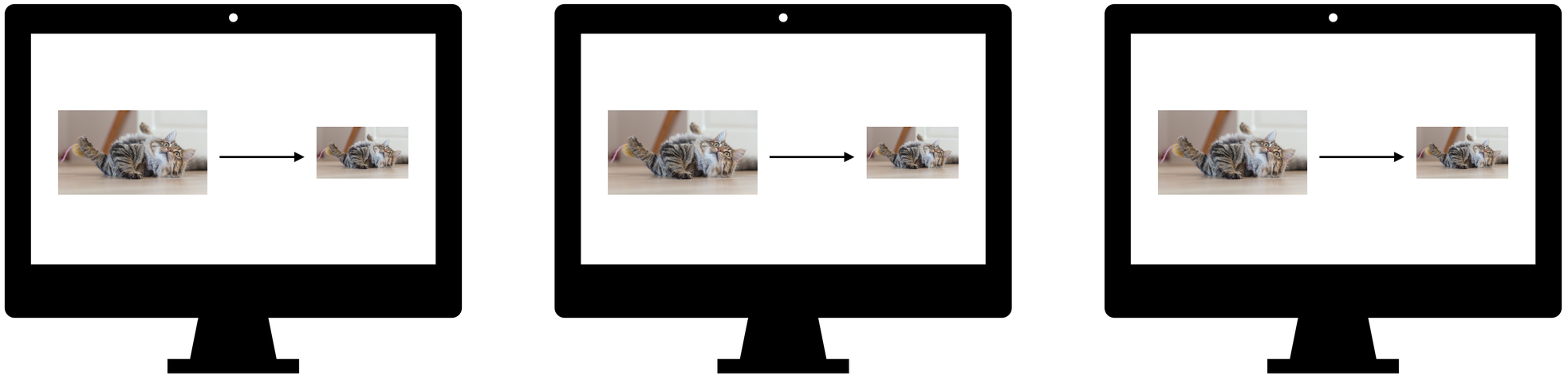
[*]starling lab

[*]starling lab

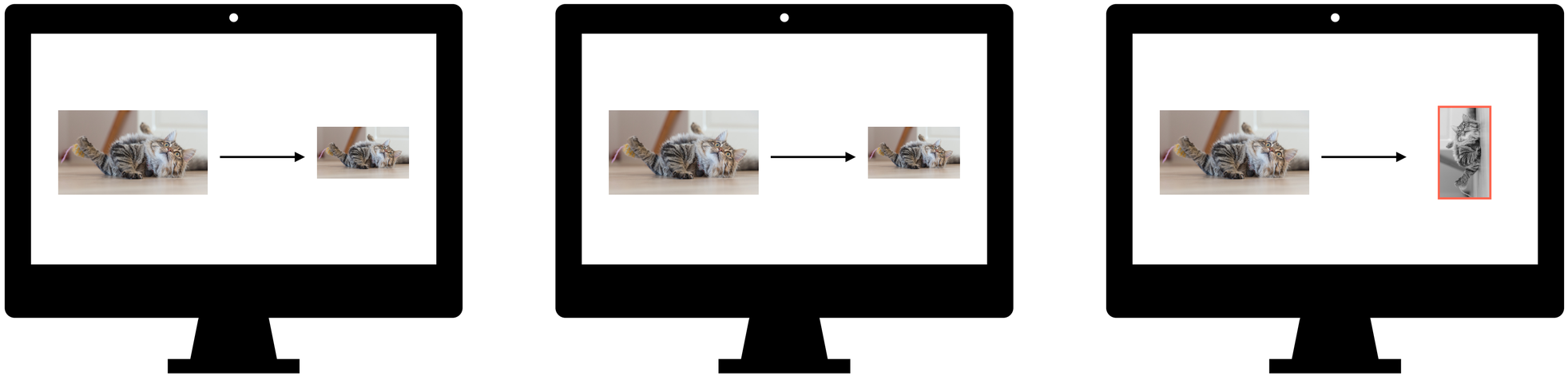
Verifiable computing

Transformation

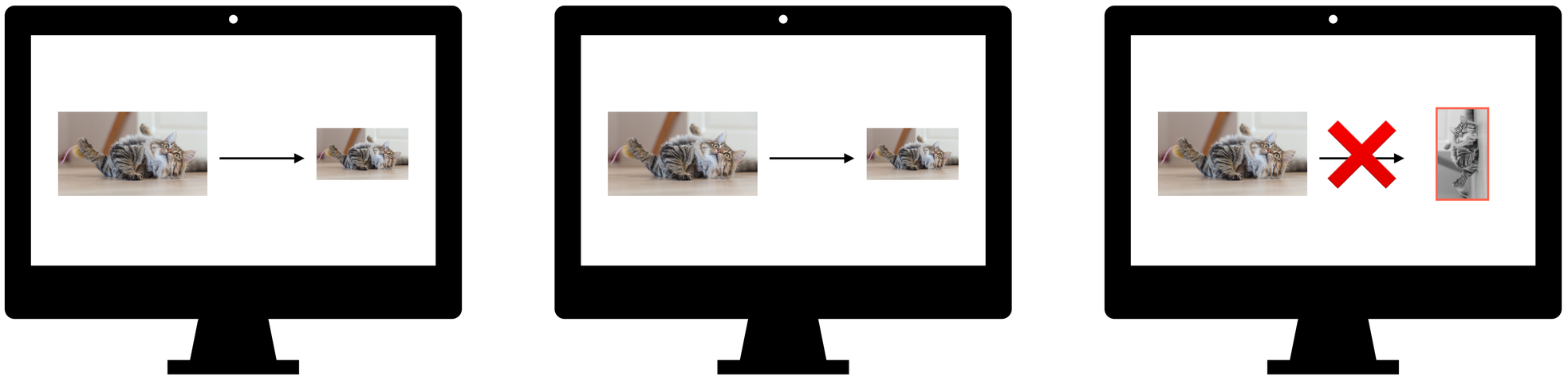
Consensus on blockchains



Consensus on blockchains



Consensus on blockchains



Consensus algorithm

ZK SNARK proofs

Zero-**K**nowledge

Succinct

Non-interactive

ARgument of **K**nowledge

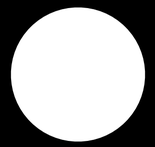
Argument of knowledge

Zero-**K**nowledge: I do not want to reveal where Waldo is

Succinct: The proof is compact and easy to verify

Non-interactive: You can verify the proof as is without interaction with me (the proofer)

ARgument of **K**nowledge: I found Waldo



Verifiable
Transformation

ZK SNARK

BACK TO ARCHIVE

BACK TO THE STORY

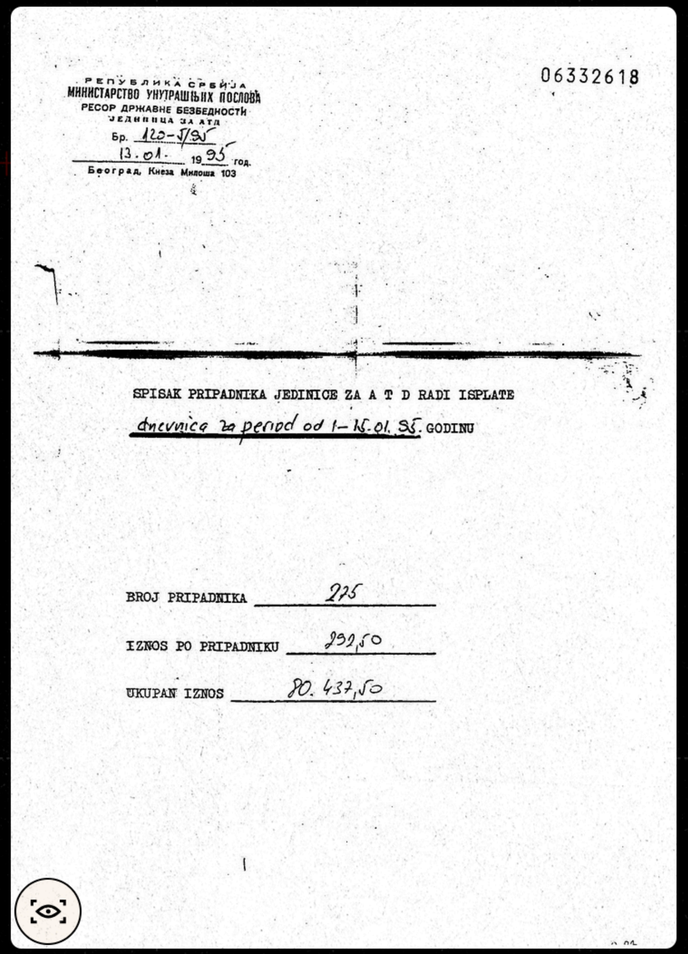
THE DOCUMENT LISTS HUNDREDS OF NAMES, BUT NOTABLY INCLUDES:

SRDJAN GOLUBOVIC

In our investigation, we found the name **ГОЛУБОВИЋ СРЂАН** in Cyrillic, which is often spelled Srđan Golubović in Serbian or Srdjan Golubovic in English.

The same individual in the payroll records is mentioned in ICTY testimony and has been accused by members of the Serbian press and by local citizens of being the soldier in Haviv's famous photo.

According to the payroll records, Golubović was deployed and paid in September 1994, and January, April, May, June, July, August, September, October, November, and December 1995.



06332618

РЕПУБЛИКА СРБИЈА
МИНИСТАРСТВО УНУТРАШЊИХ ПОСЛОВА
РЕСОР ДРЖАВНЕ БЕЗБЕДНОСТИ
ЈЕДИНИЦА ЗА АТД
Бр. 120-5/95
13.01. 1995 год.
Београд, Кнеза Милоша 103

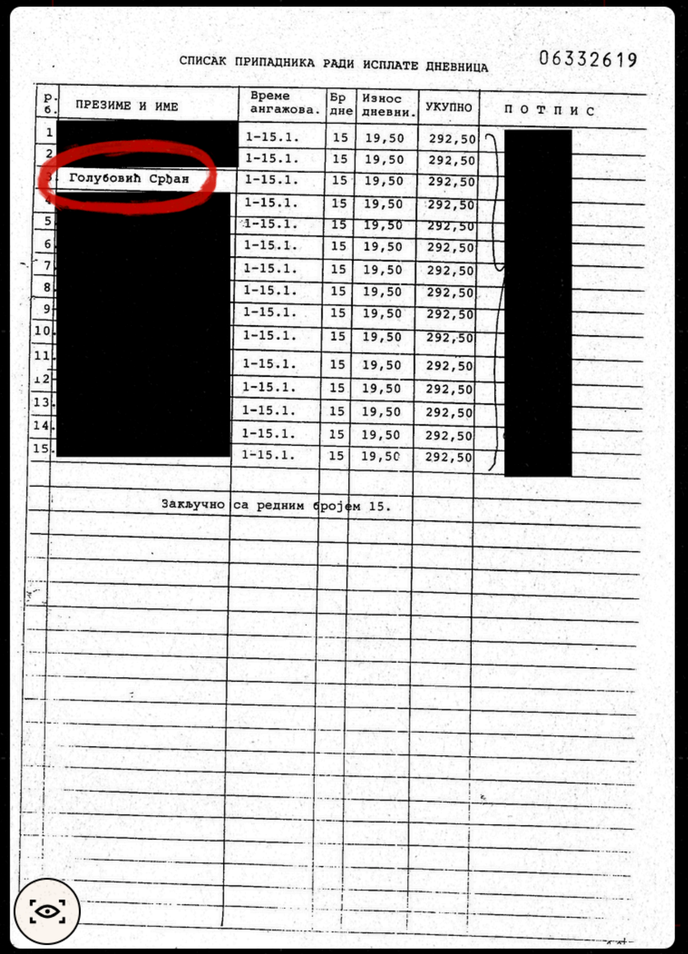
SPISAK PRIPADNIKA JEDINICE ZA A T D RADI ISPLATE

dnevnica za period od 1-15.01.95. GODINU

BROJ PRIPADNIKA 225

IZNOS PO PRIPADNIKU 292,50

UKUPAN IZNOS 80.437,50

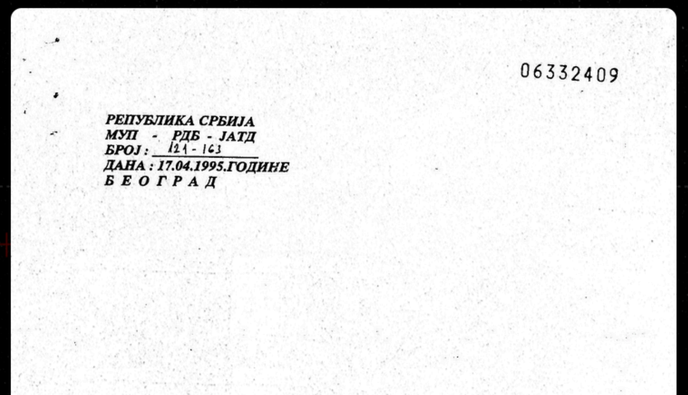


06332619

СПИСАК ПРИПАДНИКА РАДИ ИСПЛАТЕ ДНЕВНИЦА

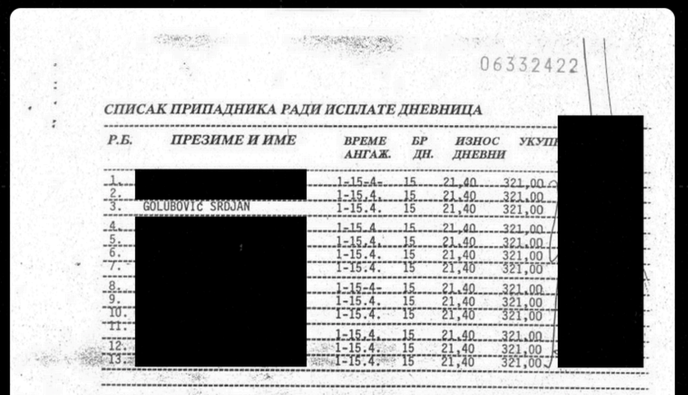
р. б.	ПРЕЗИМЕ И ИМЕ	Време ангажова.	Бр дне	Износ дневни.	УКУПНО	П О Т П И С
1	[redacted]	1-15.1.	15	19,50	292,50	[redacted]
2		1-15.1.	15	19,50	292,50	
3	Голубовић Срђан	1-15.1.	15	19,50	292,50	
4		1-15.1.	15	19,50	292,50	
5		1-15.1.	15	19,50	292,50	
6		1-15.1.	15	19,50	292,50	
7		1-15.1.	15	19,50	292,50	
8		1-15.1.	15	19,50	292,50	
9		1-15.1.	15	19,50	292,50	
10		1-15.1.	15	19,50	292,50	
11		1-15.1.	15	19,50	292,50	
12		1-15.1.	15	19,50	292,50	
13		1-15.1.	15	19,50	292,50	
14		1-15.1.	15	19,50	292,50	
15		1-15.1.	15	19,50	292,50	

Закључно са редним бројем 15.



06332409

РЕПУБЛИКА СРБИЈА
МУП - РДБ - ЈАТД
БРОЈ : 121 - 163
ДАНА : 17.04.1995.ГОДИНЕ
Б Е О Г Р А Д



06332422

СПИСАК ПРИПАДНИКА РАДИ ИСПЛАТЕ ДНЕВНИЦА

Р.Б.	ПРЕЗИМЕ И ИМЕ	ВРЕМЕ АНГАЖ.	БР ДН.	ИЗНОС ДНЕВНИ	УКУПНО
1.	[redacted]	1-15.4.	15	21,40	321,00
2.		1-15.4.	15	21,40	321,00
3.	GOLUBOVIC SRDJAN	1-15.4.	15	21,40	321,00
4.	[redacted]	1-15.4.	15	21,40	321,00
5.		1-15.4.	15	21,40	321,00
6.		1-15.4.	15	21,40	321,00
7.		1-15.4.	15	21,40	321,00
8.		1-15.4.	15	21,40	321,00
9.		1-15.4.	15	21,40	321,00
10.		1-15.4.	15	21,40	321,00
11.		1-15.4.	15	21,40	321,00
12.		1-15.4.	15	21,40	321,00
13.		1-15.4.	15	21,40	321,00

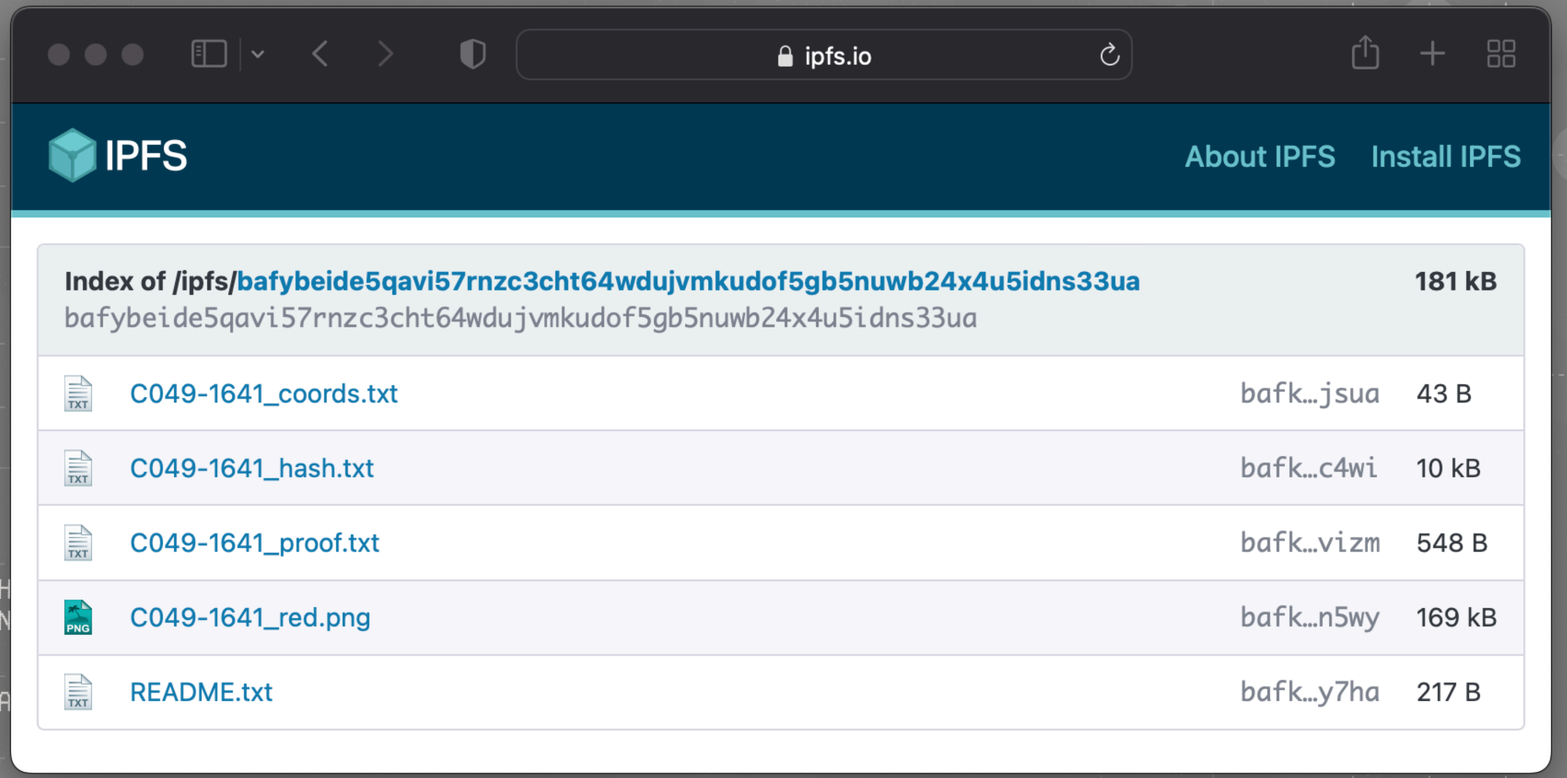
- DATE: 2022-11-02T11:17:20.231Z
  - CAPTURED USING: WEBRECORDER - ARCHIVEWEB.PAGE / AUTHENTICATED WEB ARCHIVE
  - LOCATION: LOS ANGELES, CA, UNITED STATES
  - REGISTERED ON
    - OPENTIMESTAMPS: E071739EEC16594207BE3BD255CE083C87C83935821BD131CFCD0A2ABE4E3C9C
    - NUMBERS
    - AVALANCHE
    - ISCN

2 STORE

- PRESERVED ON
  - FILECOIN

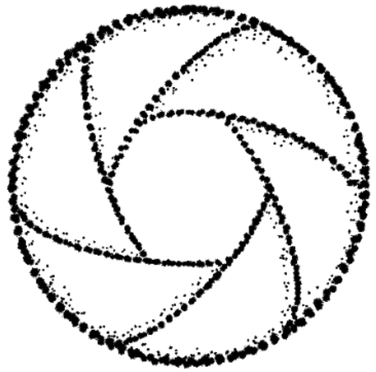
3 VERIFY

- VERIFIED BY
  - C2PA CLAIM 1: AUTH ARCHIVE AND SCREEN 11/03/2022
  - C2PA CLAIM 2: REDA PROOF
- ATTESTATIONS
  - WEB ARCHIVE: BAFYBEIHQITR4UYFT4CXYW74VZK5TNPTOSOHKL7H64I2EKESZHJCUFLOFCQ
  - REDACTION: ZERO KNOWLEDGE PROOF: BAFYBEIDE5QAVI57RNZC3CHT64WDUJVMKUDOF5GB5NUWB24X4U5IDNS33UA
  - CAPTION: BAFKREIG6ZL3AXGFB3D3P7NUQCIMIA2ARAEYV64C73UDC3XEBA2UNQ436GI

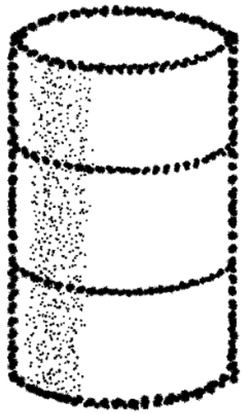


3. Building towards authenticity

Starling pipeline: Capture · Store · Verify



Gather the image or file, taking note of when and how you obtained it and other contextual metadata



Store the file and its metadata in a way that is available and persistent

Investigate the claims made by the image and metadata

Capture


5:09
Starling Capture
Dual Capture
Canon
SET
LAN
ON
LOCK
OFF
FOLDER
FILE

Reuters and Canon demonstrate end-to-end content authenticity system in the field

As the leading global news agency, Reuters works relentlessly to bring news from the source and provide readers and viewers with unbiased and reliable information from the world around us. Every day, our journalists capture the first draft of history, without bias or agenda, to inform billions of people.

Photojournalism plays a crucial role in documenting that history. Yet, with generative AI technology now available to anyone with internet access, it has become much easier to create visuals that can deceive or misinform. To make sure we can protect the trust we have built up over decades, we're always exploring innovative technologies. This is one big step forward.

Many photojournalists worldwide rely on Canon technologies. As a leading global camera manufacturer, Canon understands the role images play in society and recognises the importance of preserving image authenticity. Working as part of the Content Authenticity Initiative (CAI) alliance, Canon wants to take meaningful measures to protect image outputs that serve the news community worldwide.

Data

Data

Metadata

author	Inge Wallumrød
license	CC-BY-SA 4.0

Data

SHA 256

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf

Metadata

author	Inge Wallumrød
license	CC-BY-SA 4.0

Data

SHA 256

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf

Metadata

author → Inge Wallumrød

license → CC-BY-SA 4.0

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf
author:Inge Wallumrød
license:CC-BY-SA 4.0

Data

SHA 256

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf

Metadata

author → Inge Wallumrød

license → CC-BY-SA 4.0

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf
author:Inge Wallumrød
license:CC-BY-SA 4.0

SHA 256

72bf3d242a06d5831f83dcdba8079a7ef1709f36e993d583504fc3e7026d5aa9

[*]starling lab

Data

SHA 256

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf

Metadata

author → Inge Wallumrød

license → CC-BY-SA 4.0

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf
author:Inge Wallumrød
license:CC-BY-SA 4.0

SHA 256

72bf3d242a06d5831f83dcdba8079a7ef1709f36e993d583504fc3e7026d5aa9

Data

SHA 256

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf

Metadata

author → Inge Wallumrød

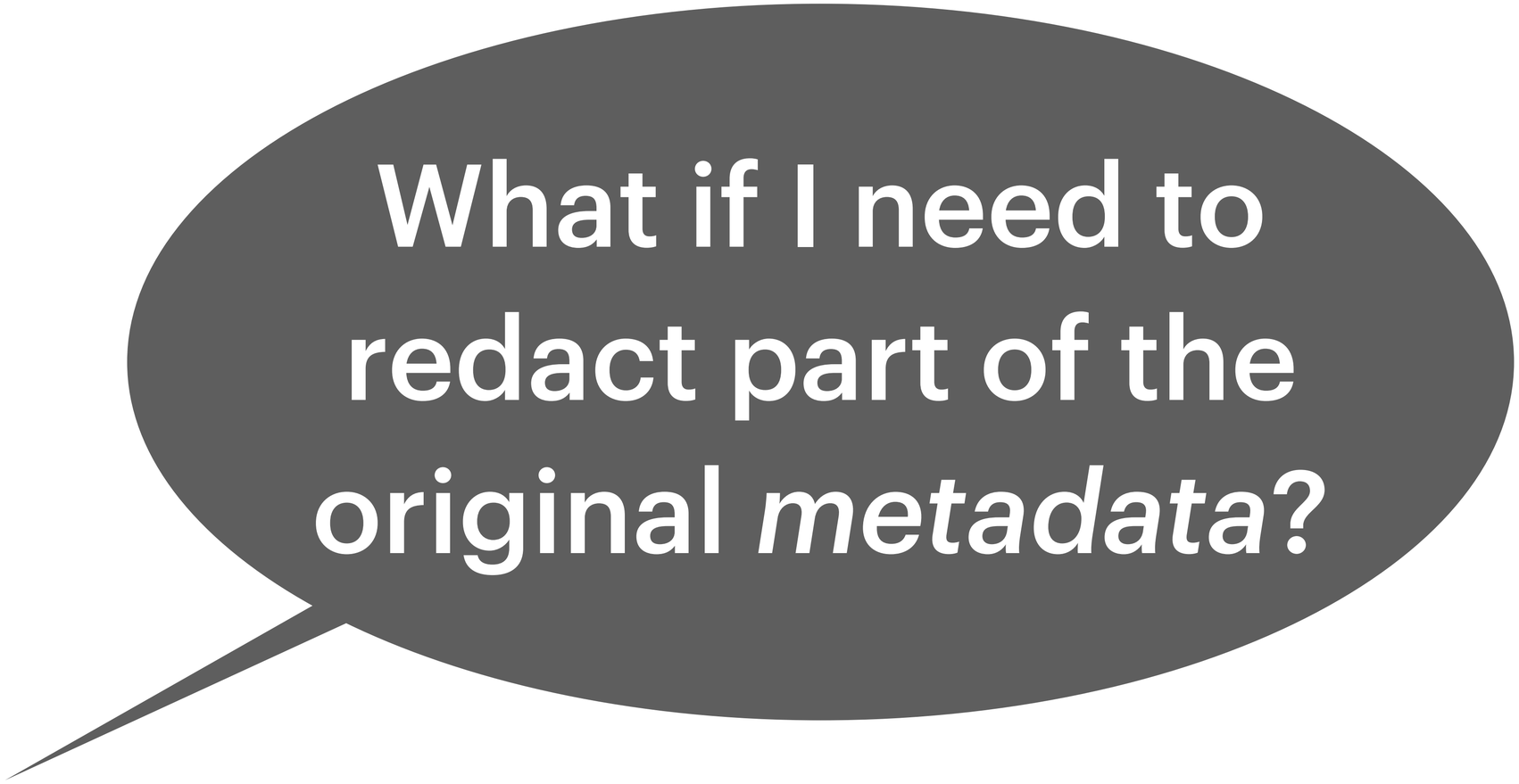
license → CC-BY-SA 4.0

600b244925ffc9665c8544083b9cc00248530f6c7b0ecdd5c89c859e3c5818cf
author:Inge Wallumrød
license:CC-BY-SA 4.0

SHA 256

Integrity

72bf3d242a06d5831f83dcdba8079a7ef1709f36e993d583504fc3e7026d5aa9


What if I need to redact part of the original *metadata*?

Data

SHA 256

600b244925ffc9665c8544083b9cc002
48530f6c7b0ecdd5c89c859e3c5818cf

Metadata

license	CC-BY-SA 4.0

600b244925ffc9665c8544083b9cc0024853
0f6c7b0ecdd5c89c859e3c5818cf
license:CC-BY-SA 4.0

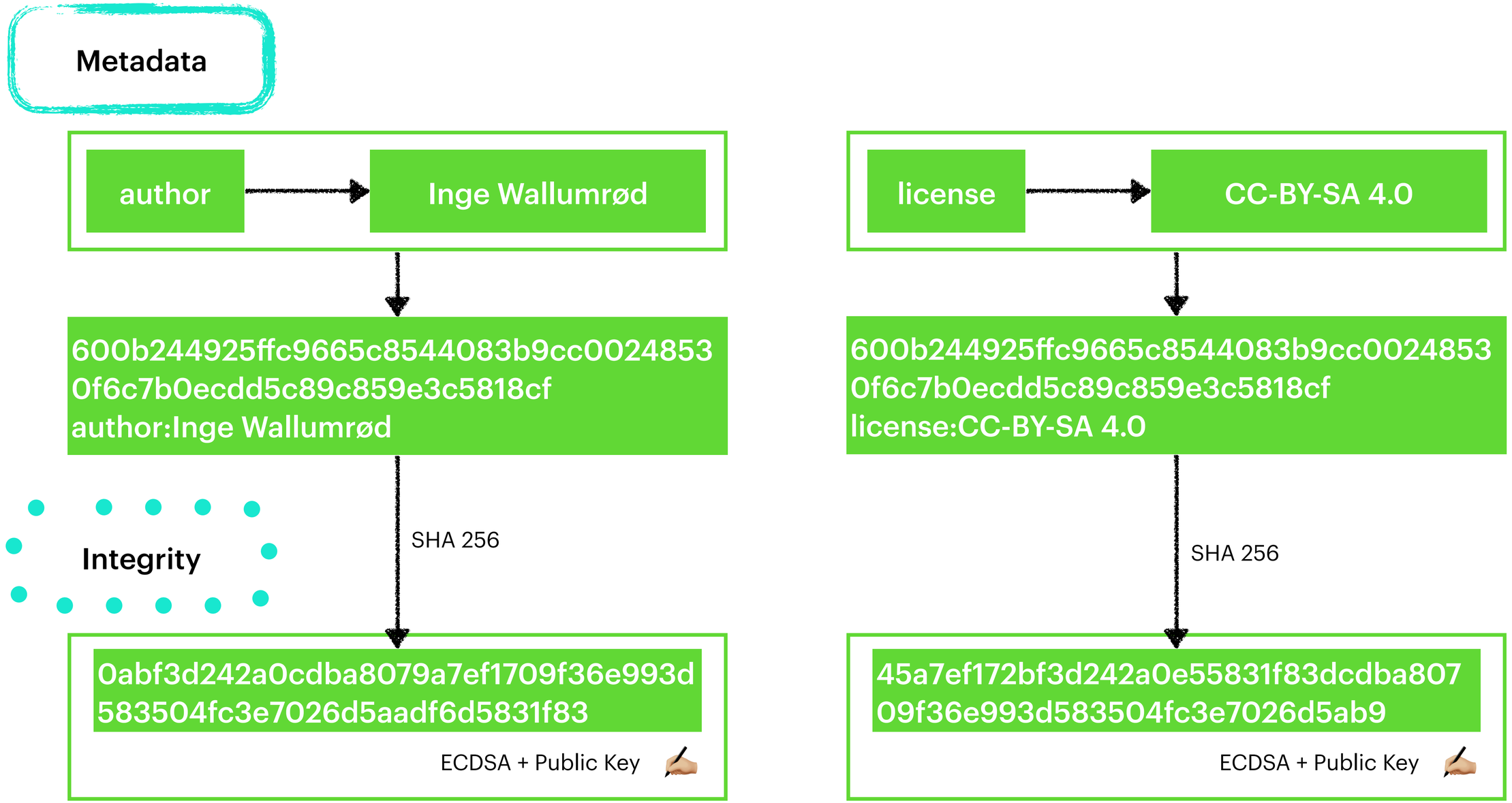
SHA 256

Integrity

72bf3d242a06d5831f83dcdba8079a7ef17
09f36e993d583504fc3e7026d5aa9

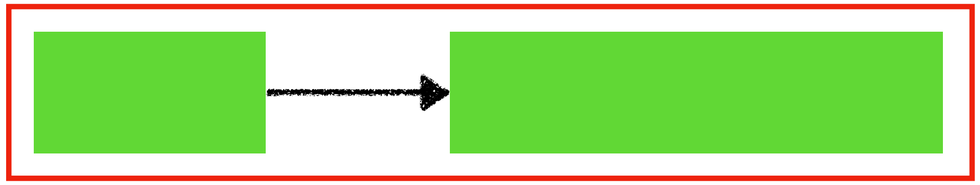
ECDSA + Public Key ✍🏻

[*]starling lab

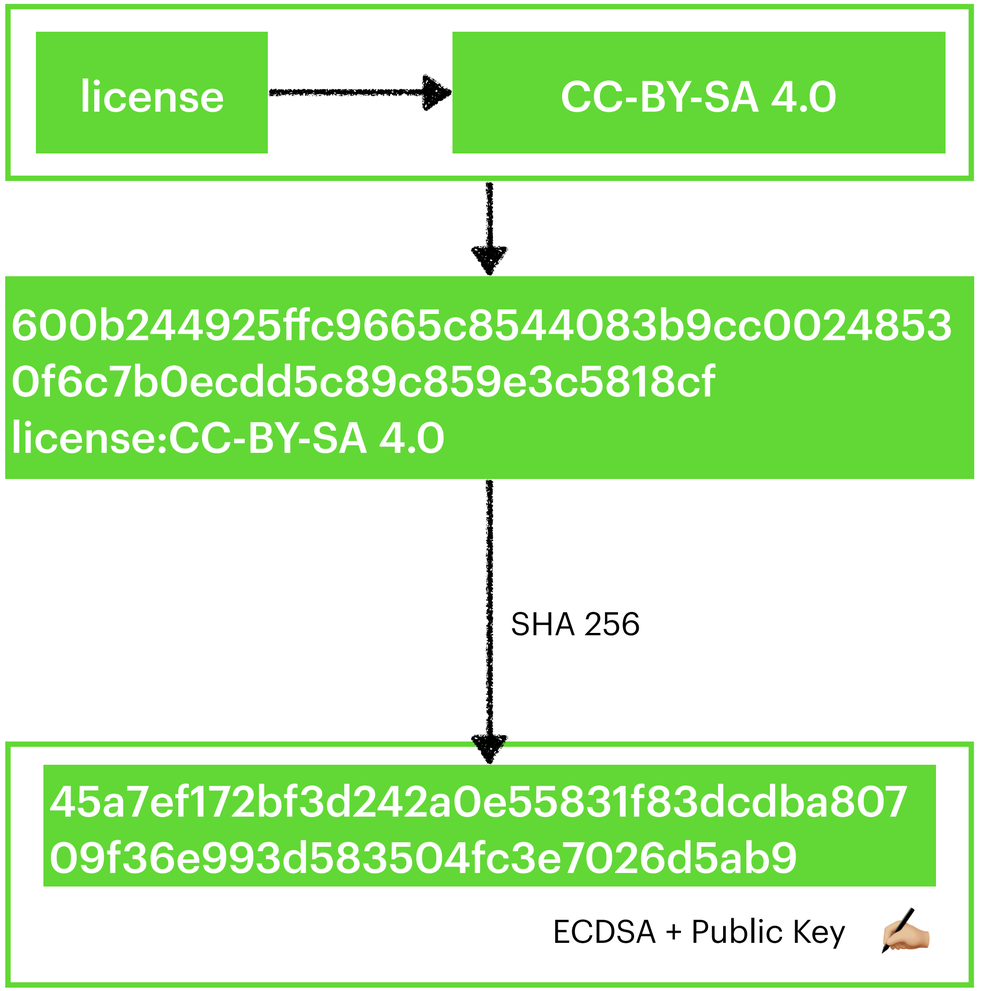


[*]starling lab

Metadata



Integrity



9d88 978e 7b2c ae78
7782 35de f098 5408

is a picture of Miu

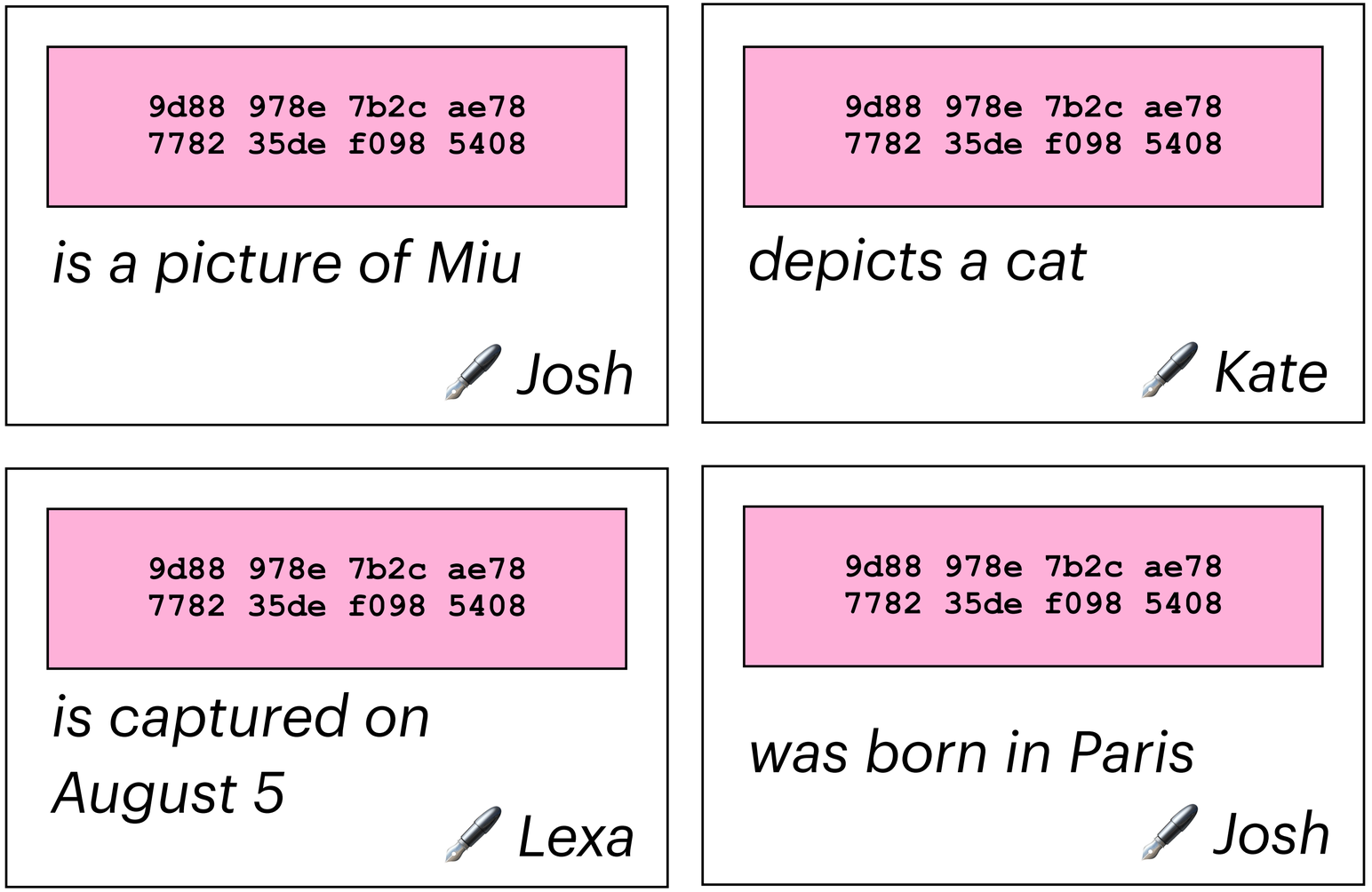
367e 62d5 2dc7 7075
a9d0 1079 7f31 82f3

Signer

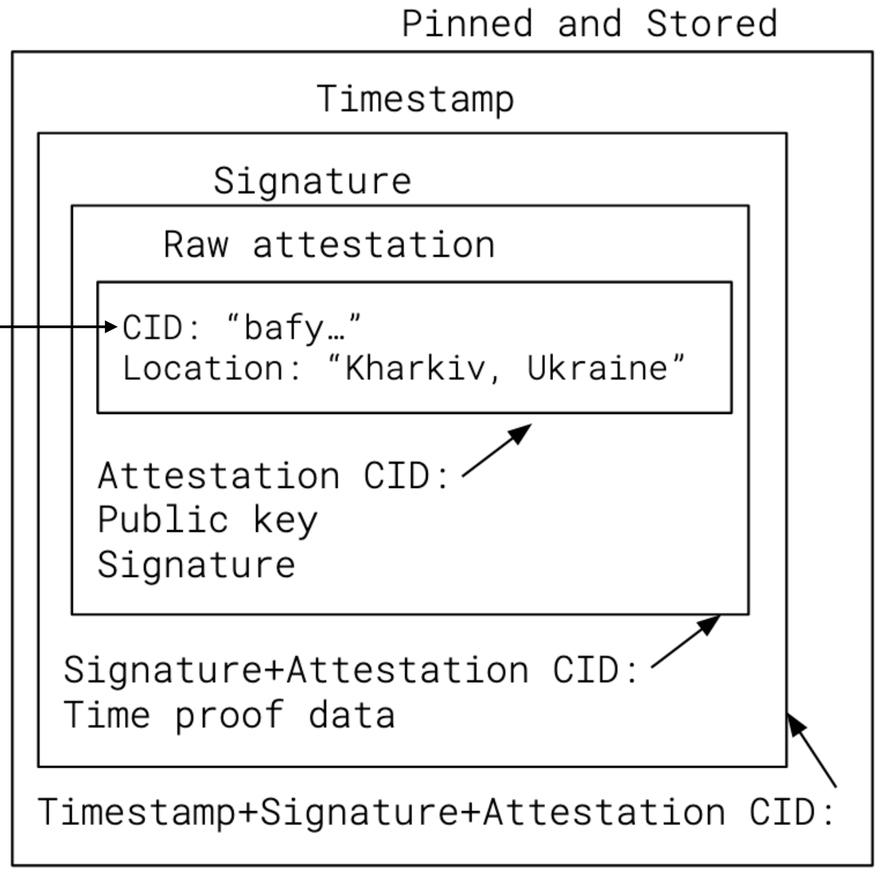
Private key

Signature

[*]starling lab


9d88 978e 7b2c ae78
7782 35de f098 5408
is a picture of Miu
Josh
9d88 978e 7b2c ae78
7782 35de f098 5408
depicts a cat
Kate
9d88 978e 7b2c ae78
7782 35de f098 5408
is captured on August 5
Lexa
9d88 978e 7b2c ae78
7782 35de f098 5408
was born in Paris
Josh

Authenticated attributes



[*]starling lab

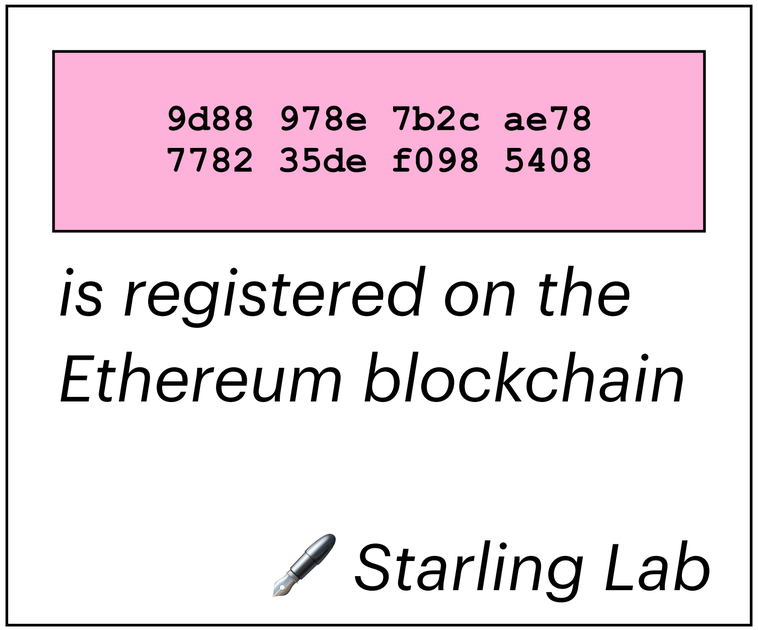
FRAMEWORK

LEDGERS

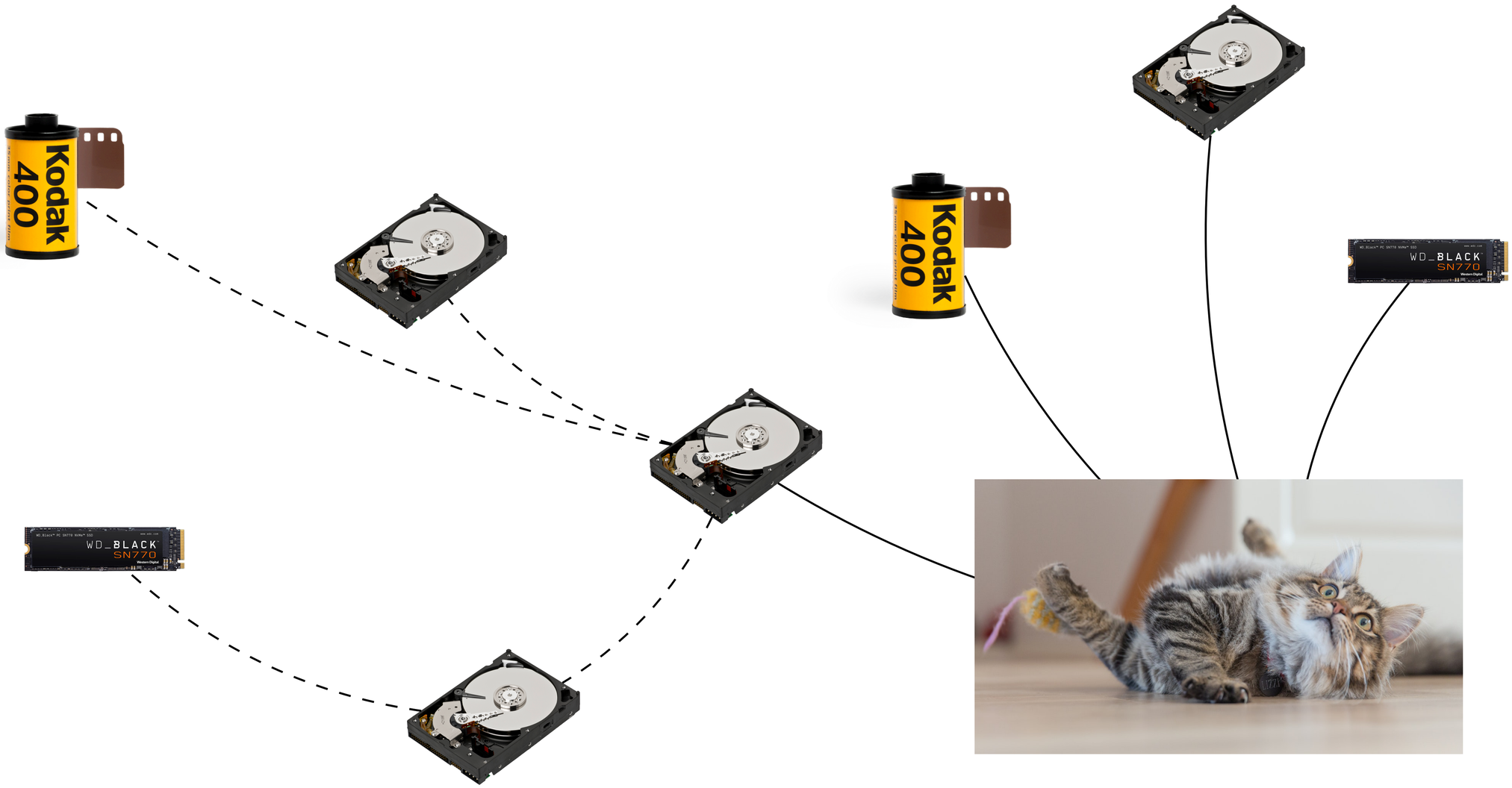
OPENTIMESTAMPS
NUMBERS
AVALANCHE
ISCN
IPFS
FILECOIN
STORJ

1 CAPTURE

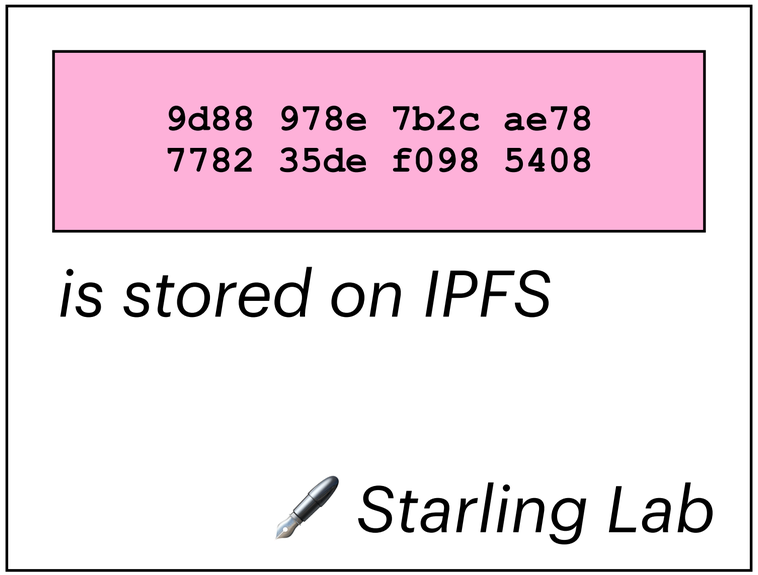
- DATE: 2022-11-02T11:17:20.231Z
  - CAPTURED USING: WEBRECORDER - ARCHIVEWEB.PAGE / AUTHENTICATED WEB ARCHIVE
  - LOCATION: LOS ANGELES, CA, UNITED STATES
  - REGISTERED ON
    - OPENTIMESTAMPS: C71E97B66F9C148C2EA36117ECCD97F57D976B50942D6B7DA55CC1D26F7DD42A
    - NUMBERS: 0X014AC7FA821E3C6AB124953F6724AFBE00DC03F7CF0C1EBB174D186F1512DEE7
    - AVALANCHE: 0XD7C191494B5726D784CF75019B8A08B9349FE4A26667419A71C65DDA184EB9D5
    - ISCN: E408581741959151FA857D2D2D61D615E593A65BDBD34A92197EAD52872ECC1C


9d88 978e 7b2c ae78
7782 35de f098 5408
is registered on the
Ethereum blockchain
Starling Lab

Store

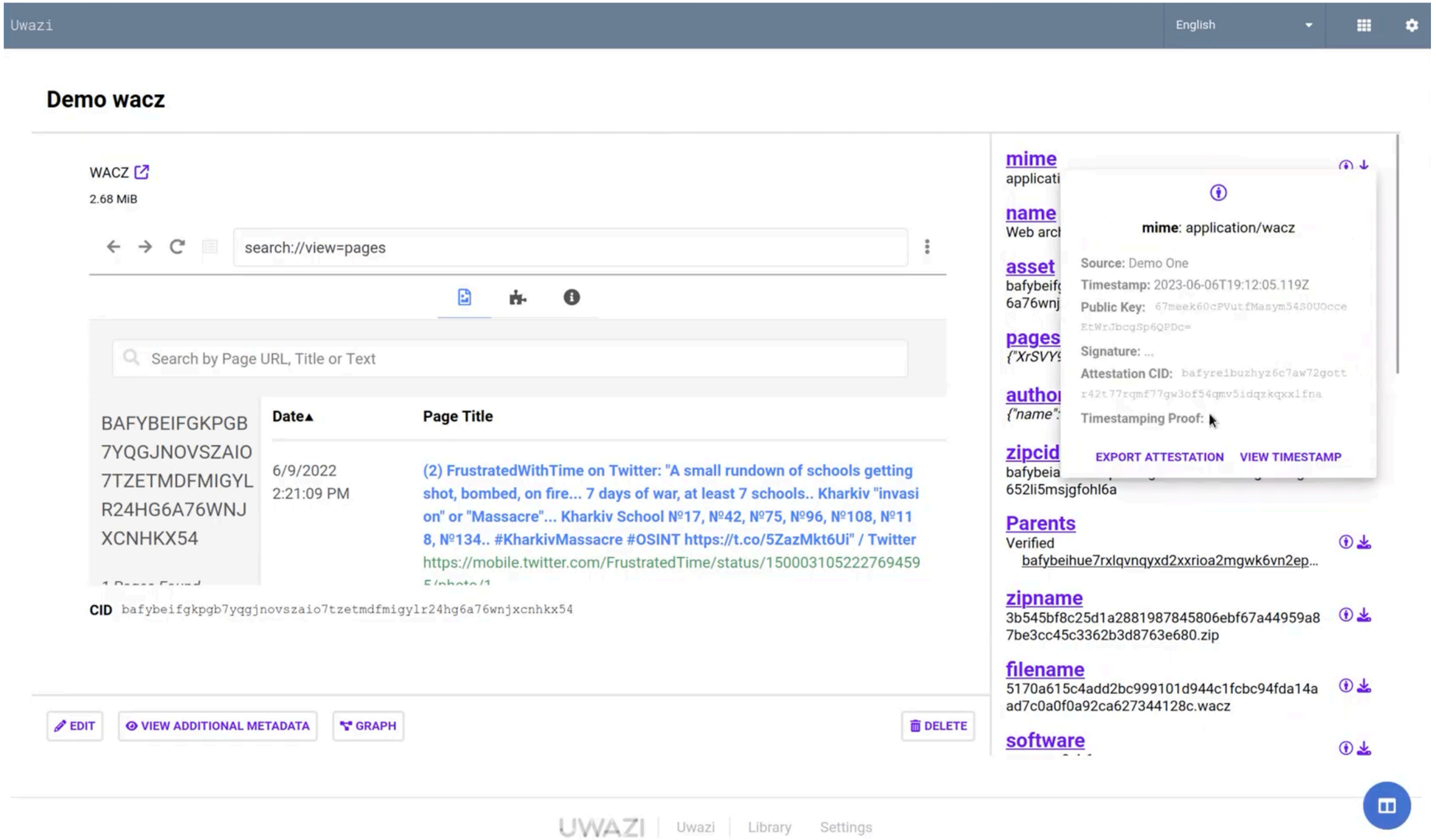


[*]starling lab


9d88 978e 7b2c ae78
7782 35de f098 5408
is stored on IPFS
Starling Lab

Verify

Investigate the claims made by the image and metadata



[*]starling lab

How does **data integrity** and **reliable attribution** support the work of verification?