
# InterPlanetary Wayback

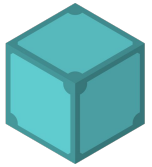
## Peer-to-Peer Permanence of Web Archives

Mat Kelly, **Sawood Alam**, Michael L. Nelson, Michele C. Weigle
Old Dominion University
Web Science and Digital Libraries Research Group
Norfolk, Virginia, USA
@WebSciDL

**http://github.com/oduwsdl/ipwb**


TPDL 2016
Hannover, Germany
September 7, 2016

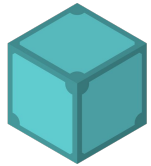
# Background - IPFS




- Hypermedia distributed protocol
- IPFS entity hashes are content addressed
  - Content changes → different hash produced
  - Inherent potential for de-duplication of content
- Files accessible via HTTP: `http://ipfs.io/<hash>`
- Built on trust chains for provenance

# Content addressing

http://foo.com/spaceDog.jpg



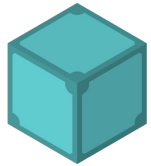
IPFS

QmZAD4xeeNeYF3TmwWgBXypLKTiCGwGRMXHW7MtheWKtw4

===

http://example.org/yuri.jpg

QmZAD4xeeNeYF3TmwWgBXypLKTiCGwGRMXHW7MtheWKtw4



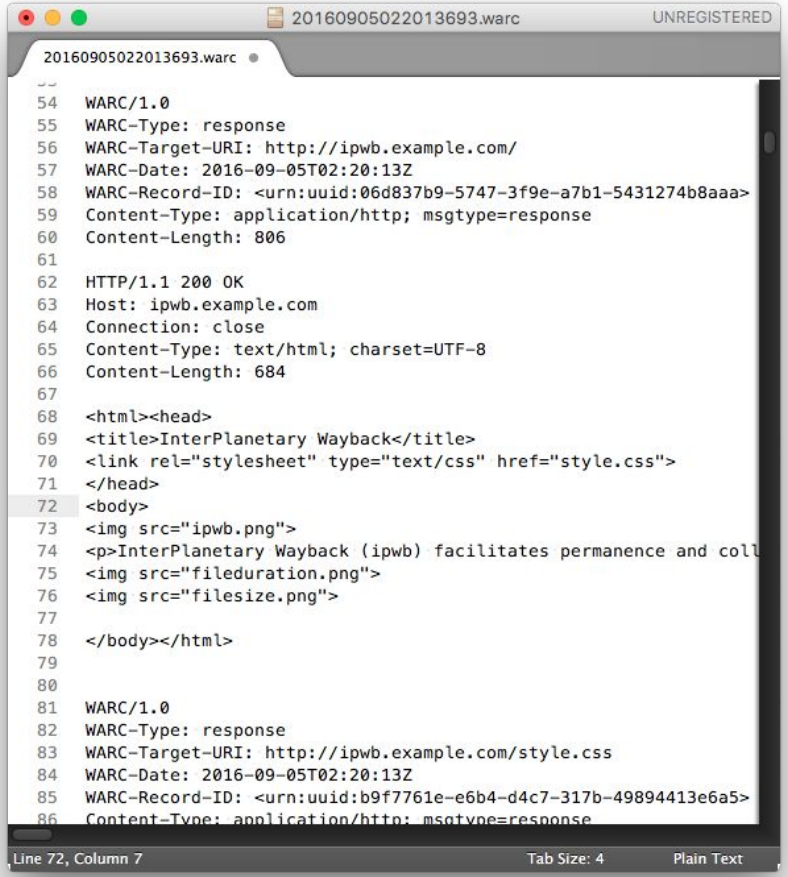
IPFS

$ ipfs cat QmZAD4xeeNeYF3TmwWgBXypLKTiCGwGRMXHW7MtheWKtw4 > doge.jpg

# Background - WARC

WARC response record

- Warc-response header
- HTTP resp header
- HTTP resp payload




WARCs also contain:

- HTTP requests
- warc-info
- warc-metadata records
- etc.

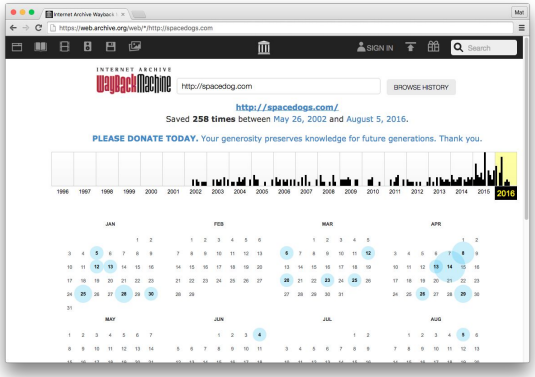
ipwb interplanetary wayback uses only warc-response records

# Background - Wayback

Archival Indexer

*processes*

*Present WARC content to user*




*outputs*

*read archived content*

Archival Index (e.g., CDXJ)

*reads (file, offset)*

Replay Engine

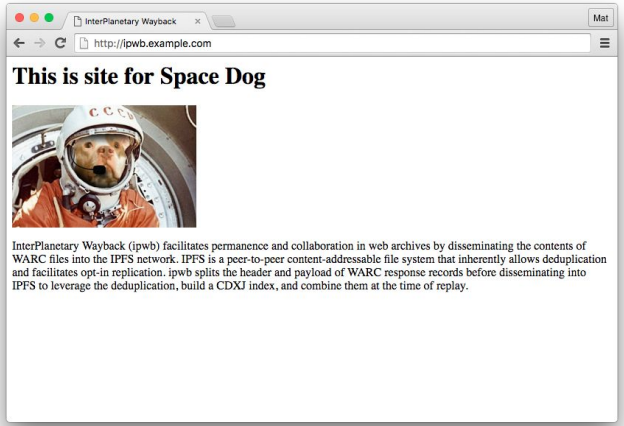
# Motivation

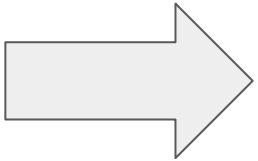
ipwb = IPFS + WayBack

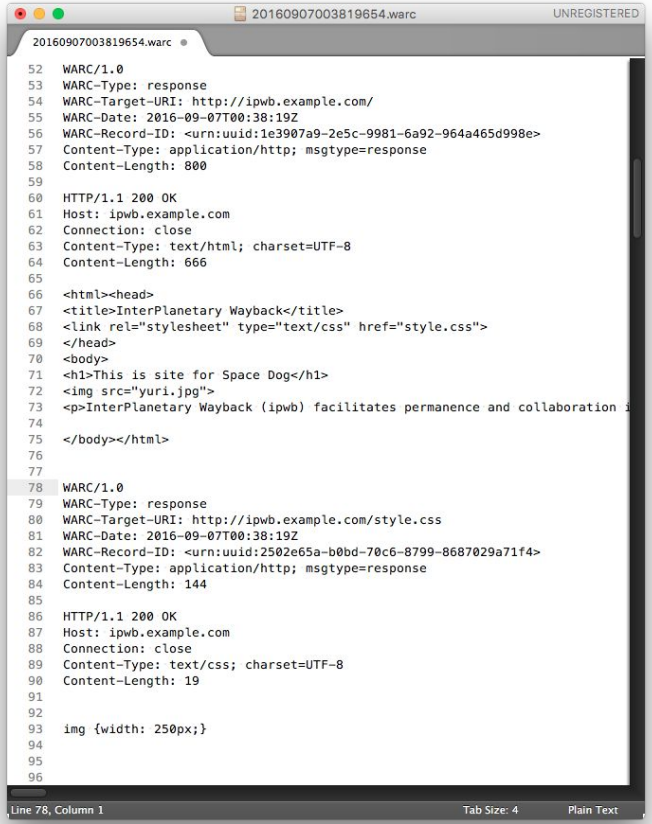
- Persistence of archived web data dependent on resilience of organization and availability of data
- Remove massive redundancy in web archive files of exact duplicate content
- Determine feasibility of pushing WARCs into IPFS

# Indexing

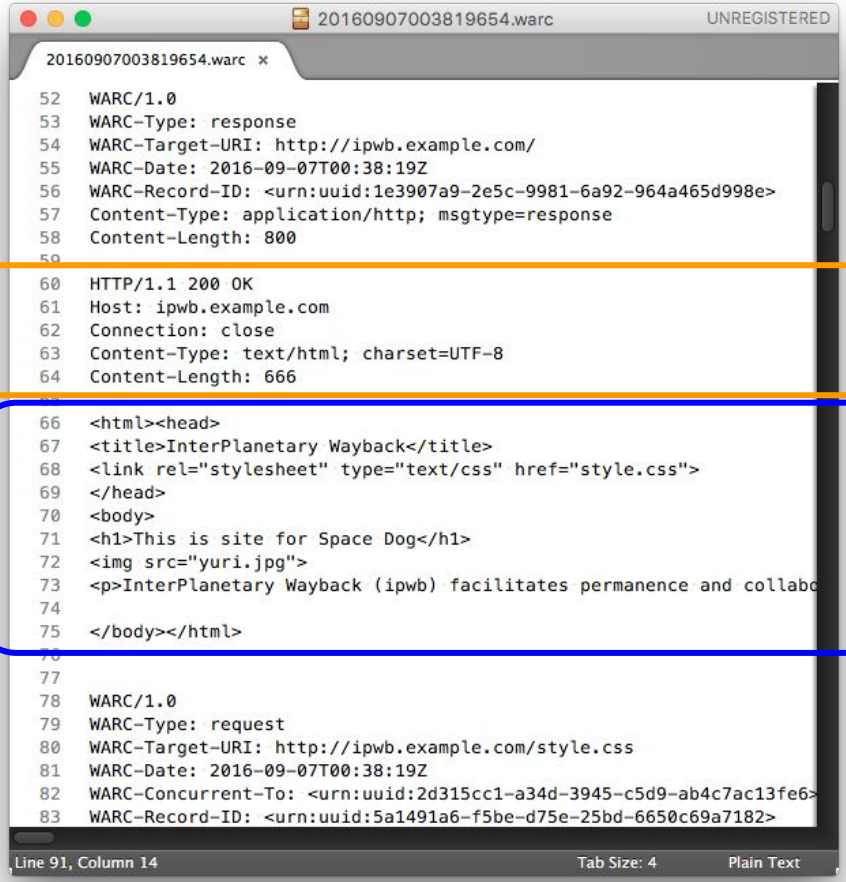
WARC Creation — HTTP Header & Payload Extraction — Push to IPFS — Generate CDXJ — WARC-CDXJ correspondence



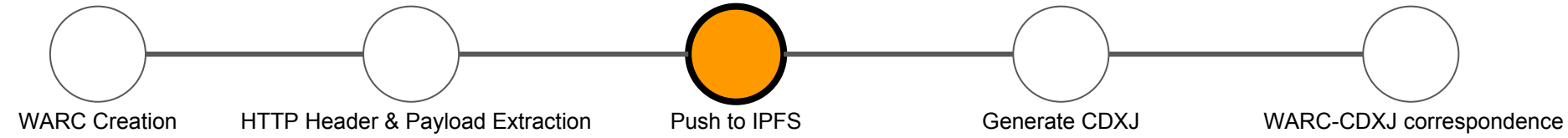


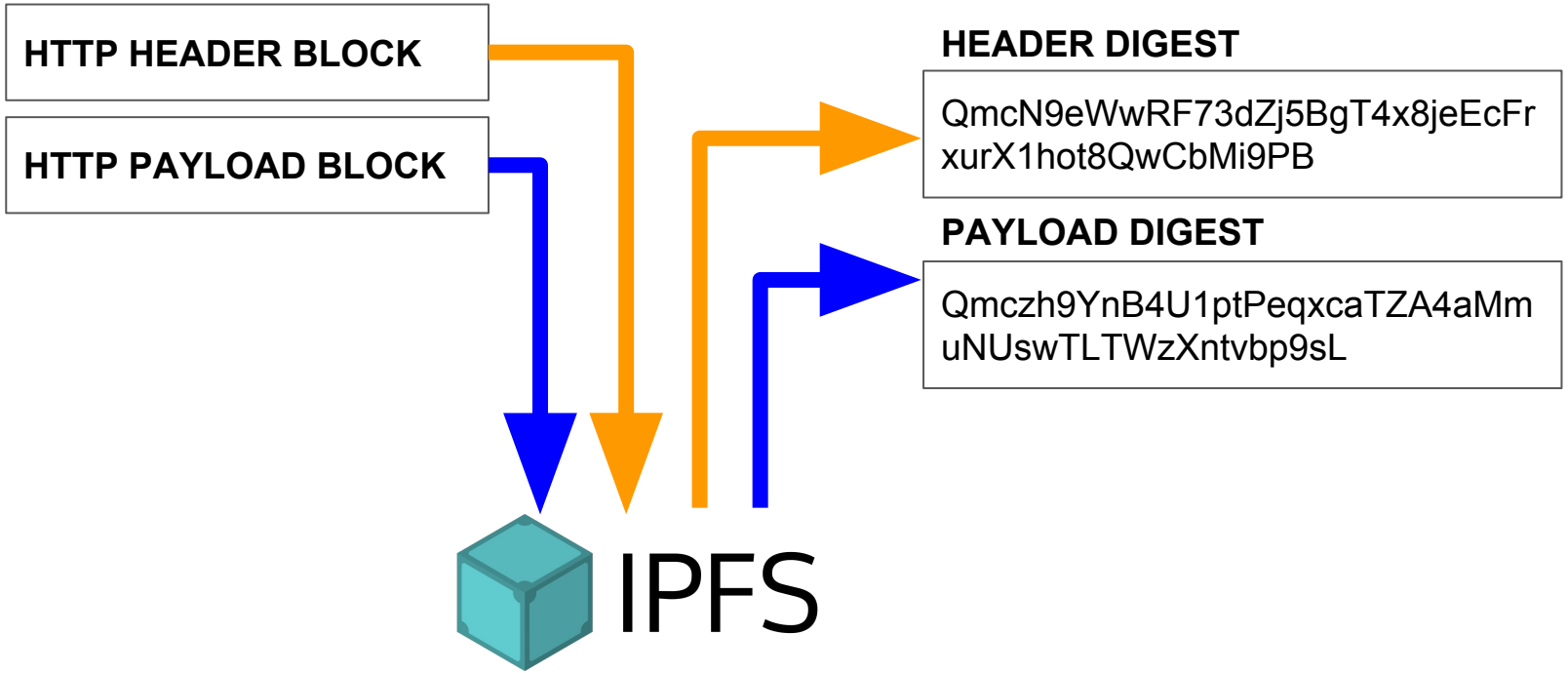
WARC Creation | HTTP Header & Payload Extraction | Push to IPFS | Generate CDXJ | WARC-CDXJ correspondence

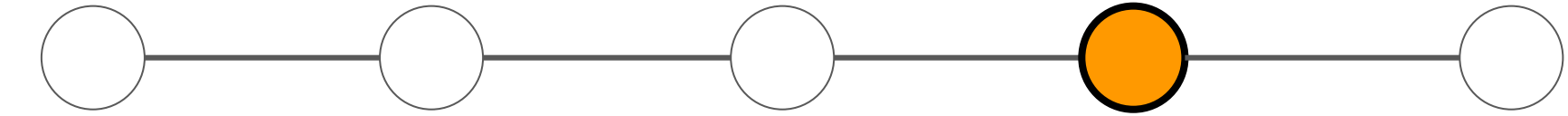



**HTTP HEADER BLOCK**

**HTTP PAYLOAD BLOCK**


WARC Creation
HTTP Header & Payload Extraction
Push to IPFS
Generate CDXJ
WARC-CDXJ correspondence


HTTP HEADER BLOCK
HTTP PAYLOAD BLOCK
HEADER DIGEST
QmcN9eWwRF73dZj5BgT4x8jeEcFrxurX1hot8QwCbMi9PB
PAYLOAD DIGEST
Qmczh9YnB4U1ptPeqxcaTZA4aMmuNUswTLTWzXntvbp9sL
IPFS




**HEADER DIGEST**

QmcN9eWwRF73dZj5BgT4x8jeEcFrxurX1hot8QwCbMi9PB

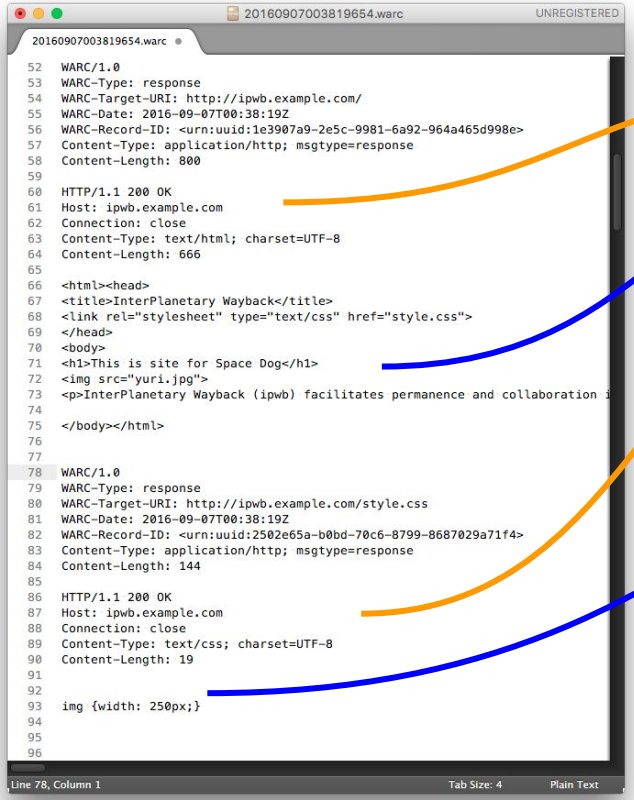
**PAYLOAD DIGEST**

Qmczh9YnB4U1ptPeqxcaTZA4aMmuNUswTLTWzXntvbp9sL

```
ipwb.example.com)/ 20160905022013 {
        "locator":"urn:ipfs/QmcN9eWwRF73dZj5BgT4x8jeEcFrxurX1hot8QwCbMi9PB/Qmczh9YnB4U1ptPeqxca
        TZA4aMmuNUswTLTWzXntvbp9sL",
        "mime_type": "text/html",
        "status_code": 200,
        “other_fields”: “other values...”
}
```

**CDXJ:** http://ws-dl.blogspot.com/2015/09/2015-09-10-cdxj-object-resource-stream.html

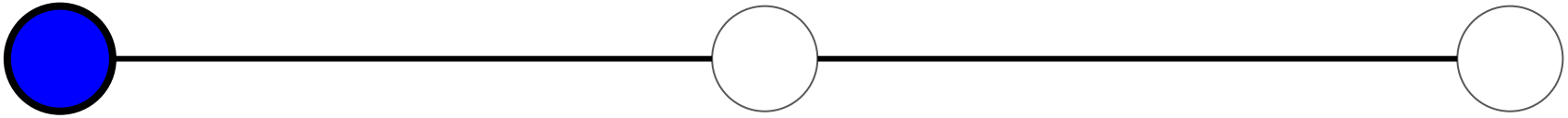
WARC Creation — HTTP Header & Payload Extraction — Push to IPFS — Generate CDXJ — WARC-CDXJ correspondence



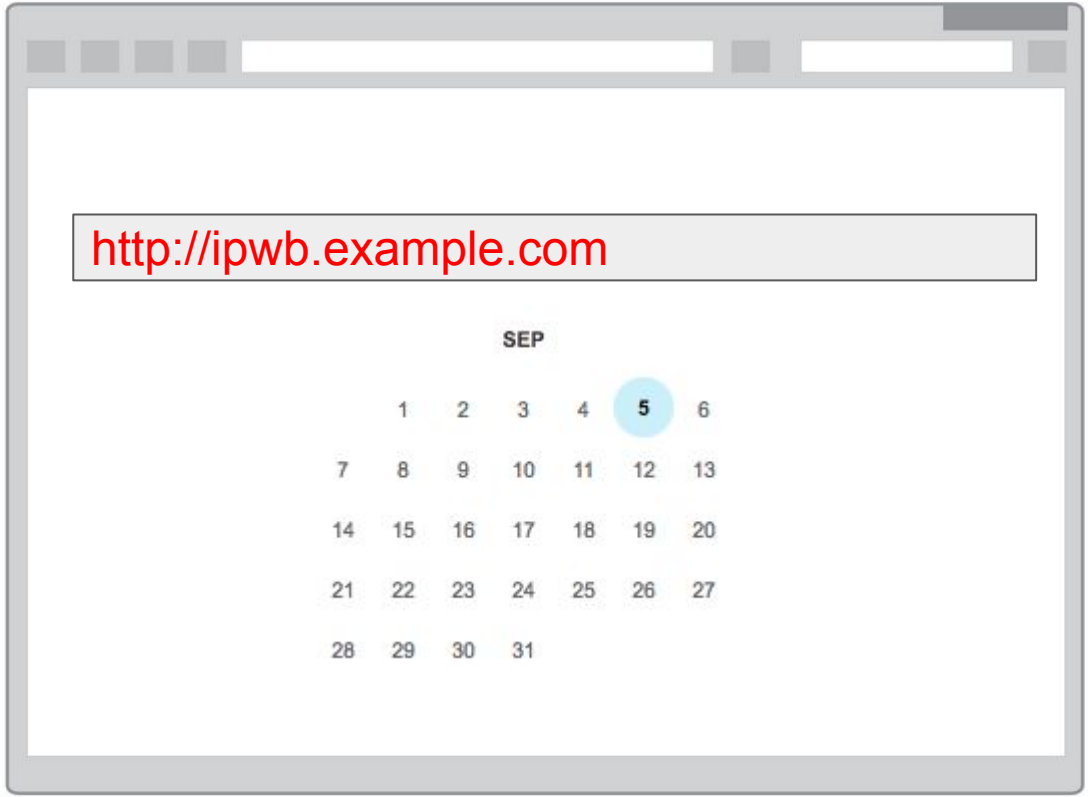


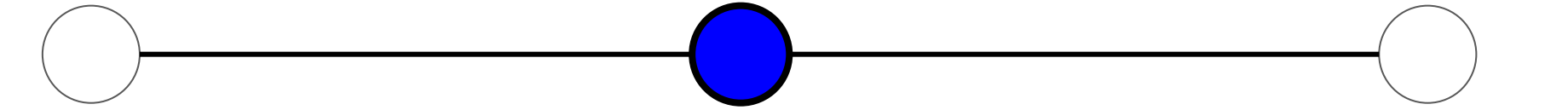
ipwb.example.com)/ 20160905022013 {"locator": "urn:ipfs/QmcN9eWwRF73dZj5BgT4x8jeEcFrxurX1hot8QwCbMi9PB/Qmczh9YnB4H1ptPeqxcaTZA4aMmuNUswTLTWzXntvbp9sL", "mime_type": "text/html", "status_code": "200"}
ipwb.example.com)/style.css 20160905022013 {"locator": "urn:ipfs/QmU1k71bT6ibZBSdxBL35cQXwovTih8cTB4CXfrjyMfZxE/QmbvUAo9U31wSdvARjvbPeVBTAwCjN1kyPhQ4ho3n8TAZo", "mime_type": "text/css", "status_code": "200"}
ipwb.example.com)/ipwb.png 20160905022013 {"locator": "urn:ipfs/QmTjfMxFGvbP4nwFoq3tNYDPW6gC99i5njrqsXSw6QRvHa/QmYMKZbnk53kuPJirahJHGevCCy2afLyePRdX38TukFUwd", "mime_type": "image/png", "status_code": "200"}
ipwb.example.com)/fileduration.png 20160905022013 {"locator": "urn:ipfs/QmaCj6LNngxwqxaLmfp1xCyxcwDt2Uzqf8gCG6bVyQppYC/QmdgtMcGprTF8bqv7ytgMwtoi5BhRxfuvBjD6Vj2U7ohz1", "mime_type": "image/png", "status_code": "200"}
ipwb.example.com)/filesize.png 20160905022013 {"locator": "urn:ipfs/QmNPjrSVY31oGDooMiA18ZDNHfkLnEg3j5gRj1dFdrqmS4/Qmb4heB8PU58nkWt6w5tBgMfpeLTKuU7iuxg9tFdoPsF1B", "mime_type": "image/png", "status_code": "200"}

# Replay

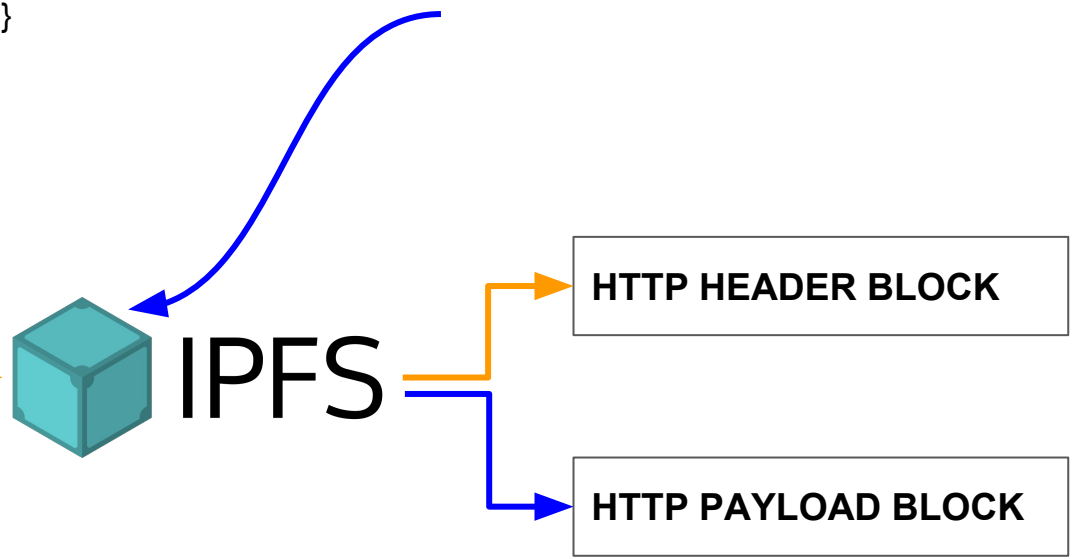


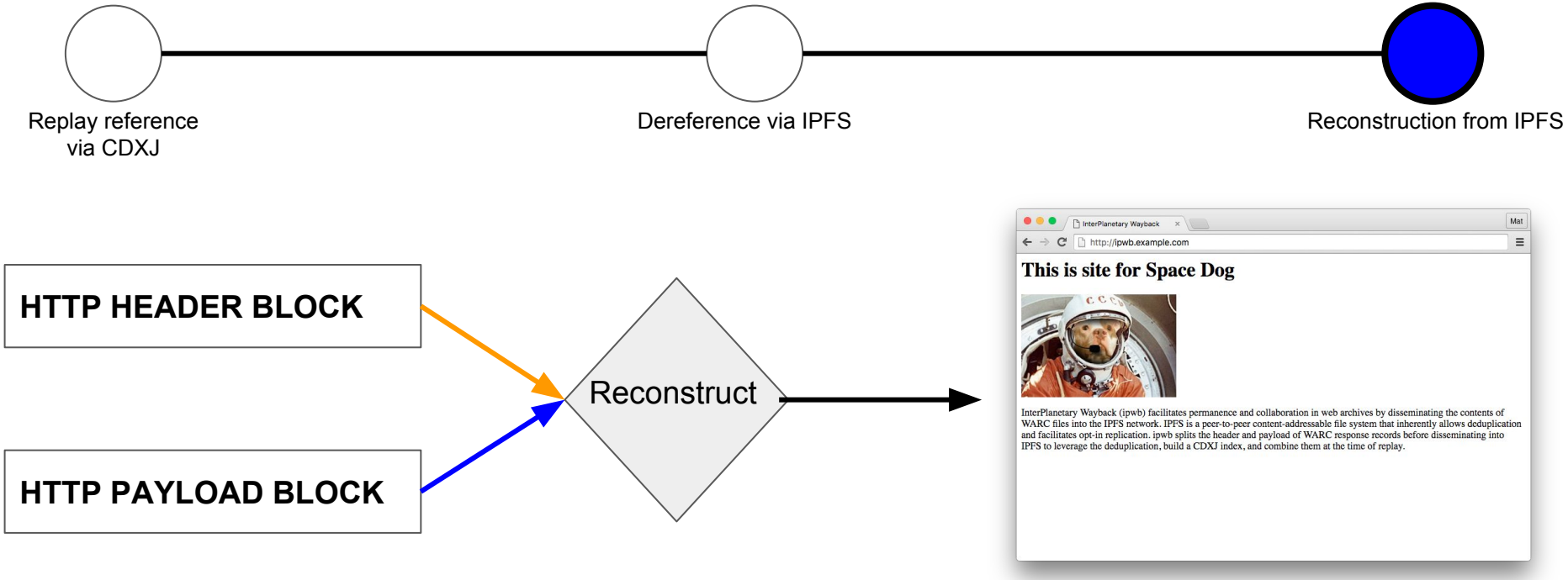

ipwb.example.com)/ 20160905022013 {"locator": "urn:ipfs/QmcN9eWwRF73dZj5BgT4x8jeEcFrxurX1hot8QwCbMi9PB/Qmczh9YnB4U1ptPeqxcaTZA4aMmuNUswTLTWzXntvbp9sL", "mime_type": "text/html", "status_code": "200"}
ipwb.example.com)/style.css 20160905022013 {"locator": "urn:ipfs/QmU1k71bT6ibZBSdxBL35cQXwovTih8cTB4CXfrjyMfZxE/QmbvUAo9U31wSdvARjvbPeVBTAwCjN1kyPhQ4ho3n8TAZo", "mime_type": "text/css", "status_code": "200"}
ipwb.example.com)/ipwb.png 20160905022013 {"locator": "urn:ipfs/QmTjfMxFGvbP4nwFoq3tNYDPW6gC99i5njrqsXSw6QRvHa/QmYMKZbnk53kuPJirahJHGevCCy2afLyePRdX38TukFUwd", "mime_type": "image/png", "status_code": "200"}
...



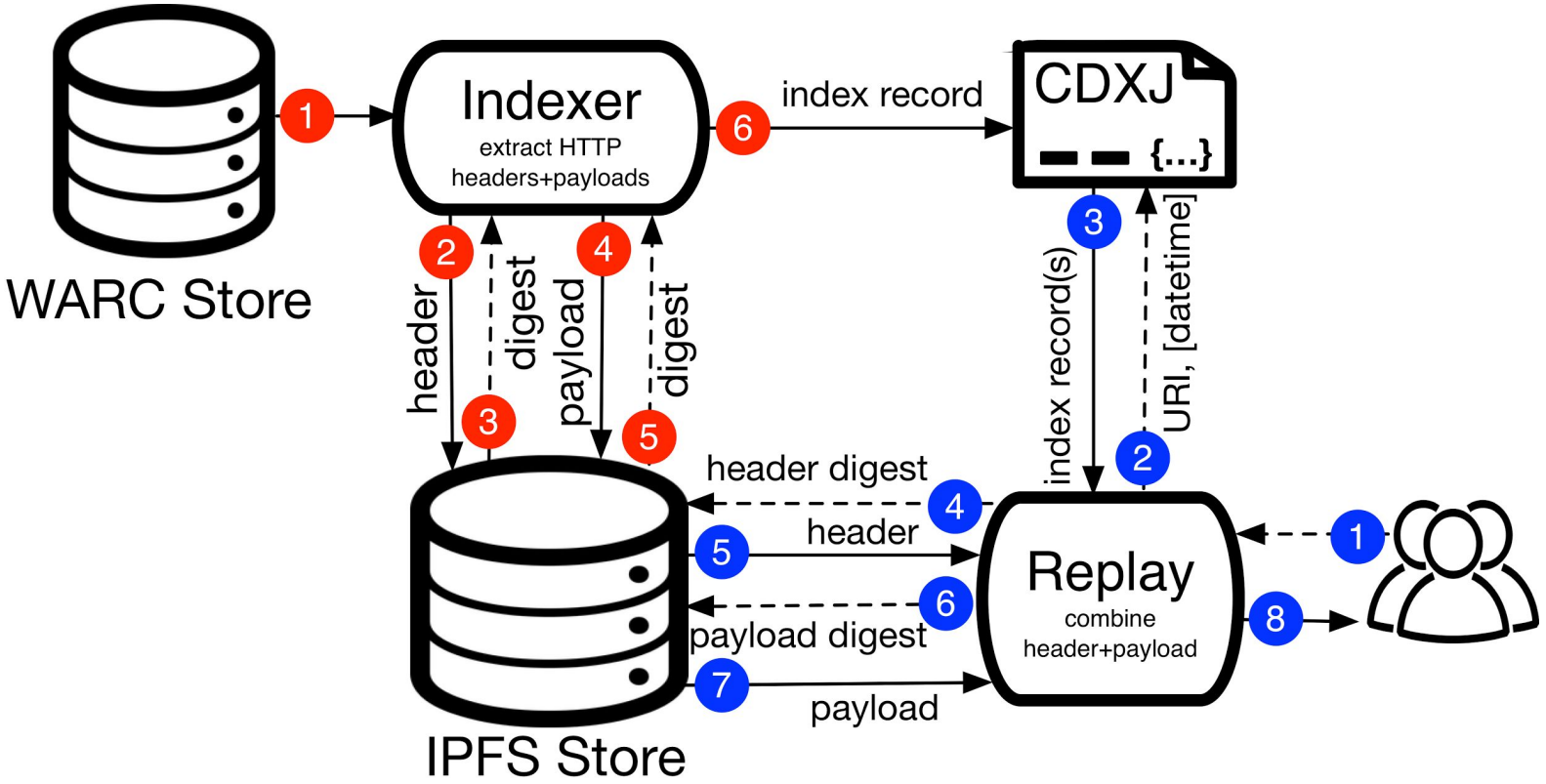



ipwb.example.com)/ 20160905022013 {"locator": "urn:ipfs/
QmcN9eWwRF73dZj5BgT4x8jeEcFrxurX1hot8QwCbMi9PB/Qmczh9YnB4U1ptPeqxcaTZA4aMmuNUswTLTWzXntvbp9sL",
"mime_type": "text/html", "status_code": "200"}
...

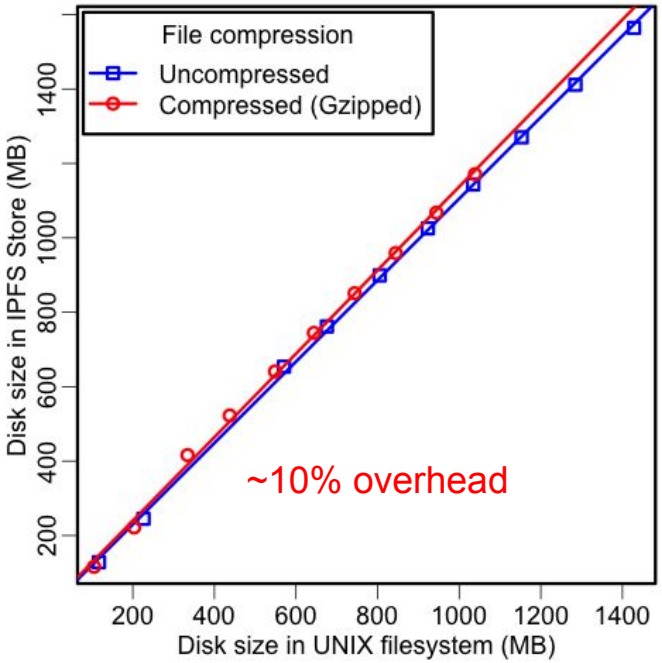


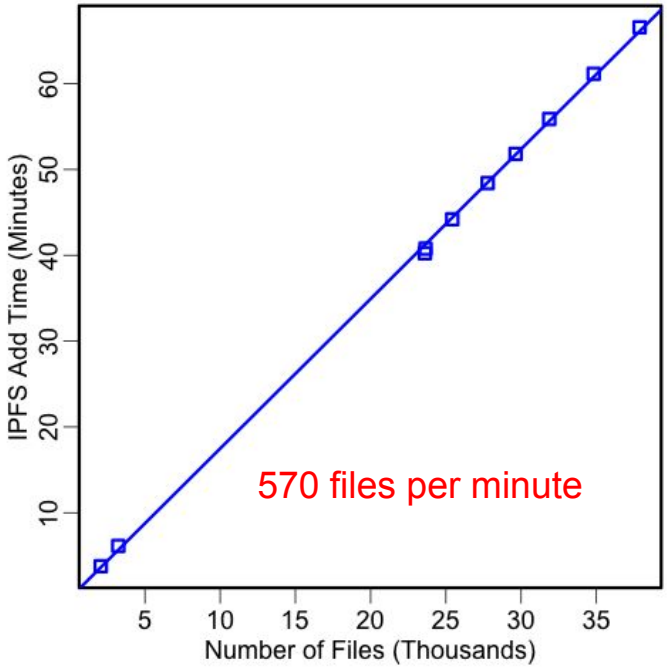

Replay reference via CDXJ
Dereference via IPFS
Reconstruction from IPFS
HTTP HEADER BLOCK
HTTP PAYLOAD BLOCK
Reconstruct
InterPlanetary Wayback
Mat
http://ipwb.example.com
This is site for Space Dog
InterPlanetary Wayback (ipwb) facilitates permanence and collaboration in web archives by disseminating the contents of WARC files into the IPFS network. IPFS is a peer-to-peer content-addressable file system that inherently allows deduplication and facilitates opt-in replication. ipwb splits the header and payload of WARC response records before disseminating into IPFS to leverage the deduplication, build a CDXJ index, and combine them at the time of replay.

# Data Flow


WARC Store
1
Indexer
extract HTTP
headers+payloads
6
index record
CDXJ
{...}
2
header
3
digest
4
payload
5
digest
3
index record(s)
2
URI, [datetime]
header digest
4
header
5
6
payload digest
7
payload
IPFS Store
Replay
combine
header+payload
1
8

# Evaluation




- Reported IPFS slowness https://github.com/ipfs/go-ipfs/issues/1216
  - Has since been fixed, subsequent to IPWB-TPDL

# Replay Time

- 600 requests in 222 seconds
- Slower than PyWB (which took 5.26 seconds)
- File vs. rich object based retrieval
- Never expiring cache

# Future Works

- Evaluate the improved IPFS on large dataset
- Evaluate deduplication
- Implement an index-free collaborative archiving system
- Utilize IPNS to reference URI-Rs

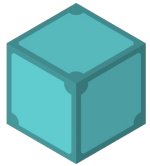
# Conclusions

- A proof of concept system to leverage a novel approach to archiving and retrieval
- Evaluated storage and time costs and qualitative analysis
- It can only work for small archives in it’s current state
- A path to answer “who will archive the archives?” question


# InterPlanetary Wayback

## Peer-to-Peer Permanence of Web Archives

Mat Kelly, Sawood Alam, Michael L. Nelson, Michele C. Weigle



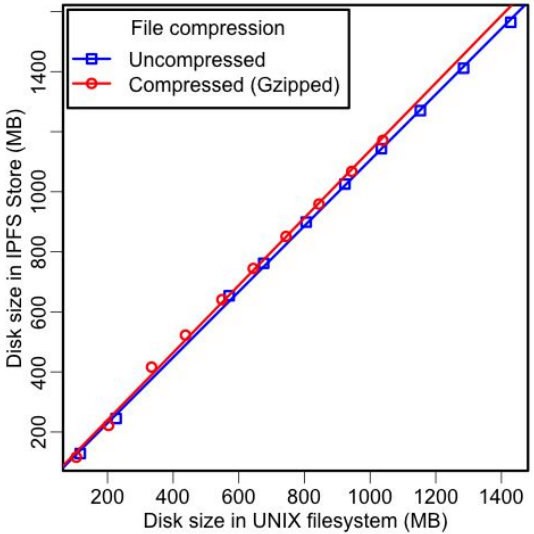

`http://github.com/oduwsdl/ipwb`

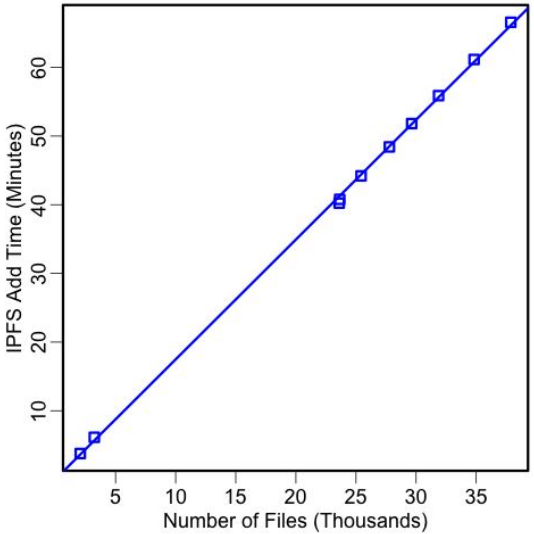
**@WebSciDL**

Support: NSF #1624067 via the Archives Unleashed Hackathon

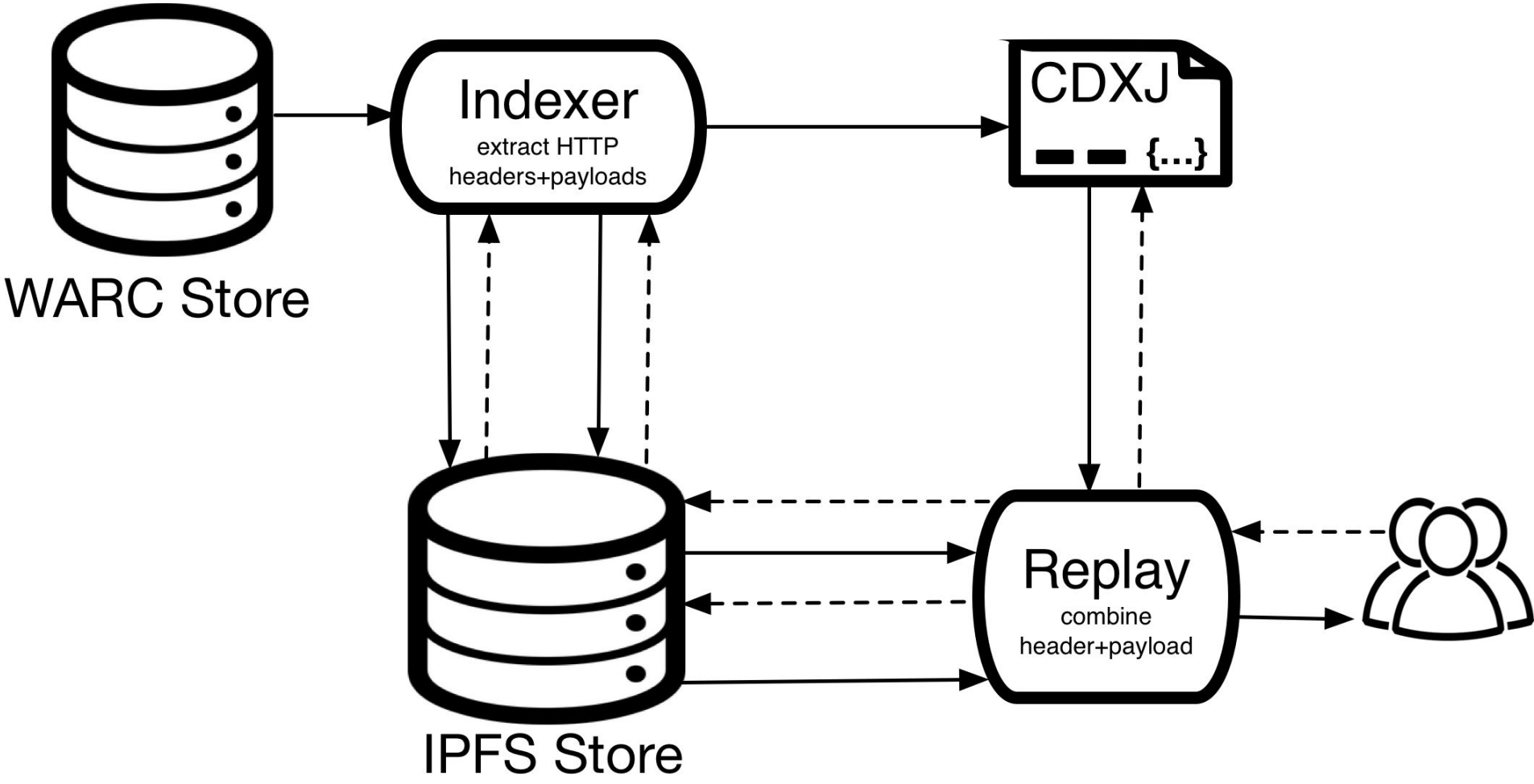
# Backup Slides

# Evaluation

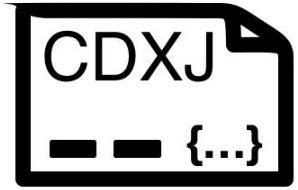


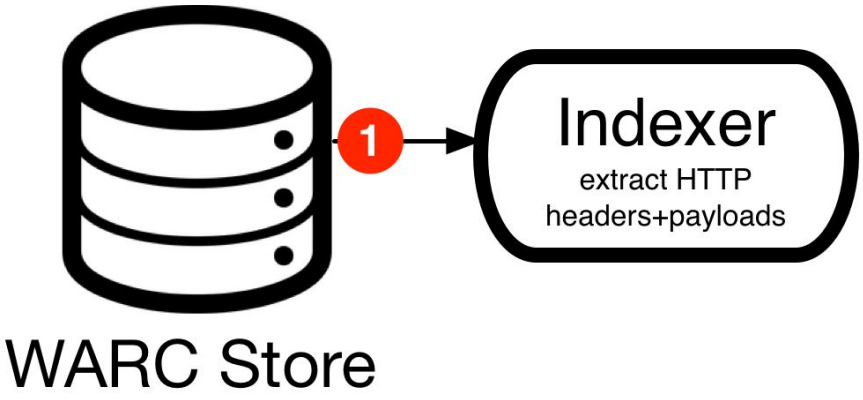

- Reported IPFS slowness https://github.com/ipfs/go-ipfs/issues/1216
  - Has since been fixed, subsequent to IPWB-TPDL

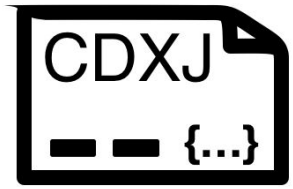

WARC Store
Indexer
extract HTTP
headers+payloads
CDXJ
{...}
IPFS Store
Replay
combine
header+payload

WARC Store

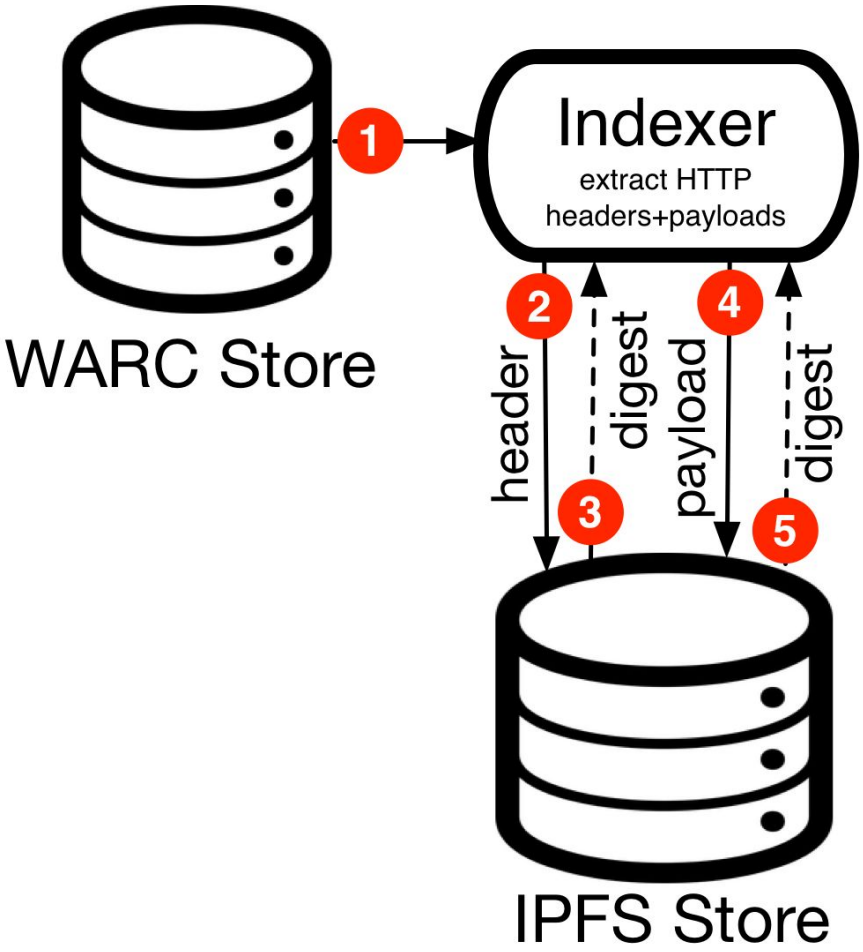

IPFS Store

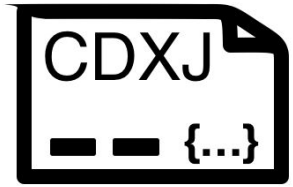


1
Indexer
extract HTTP
headers+payloads
WARC Store


CDXJ
{...}

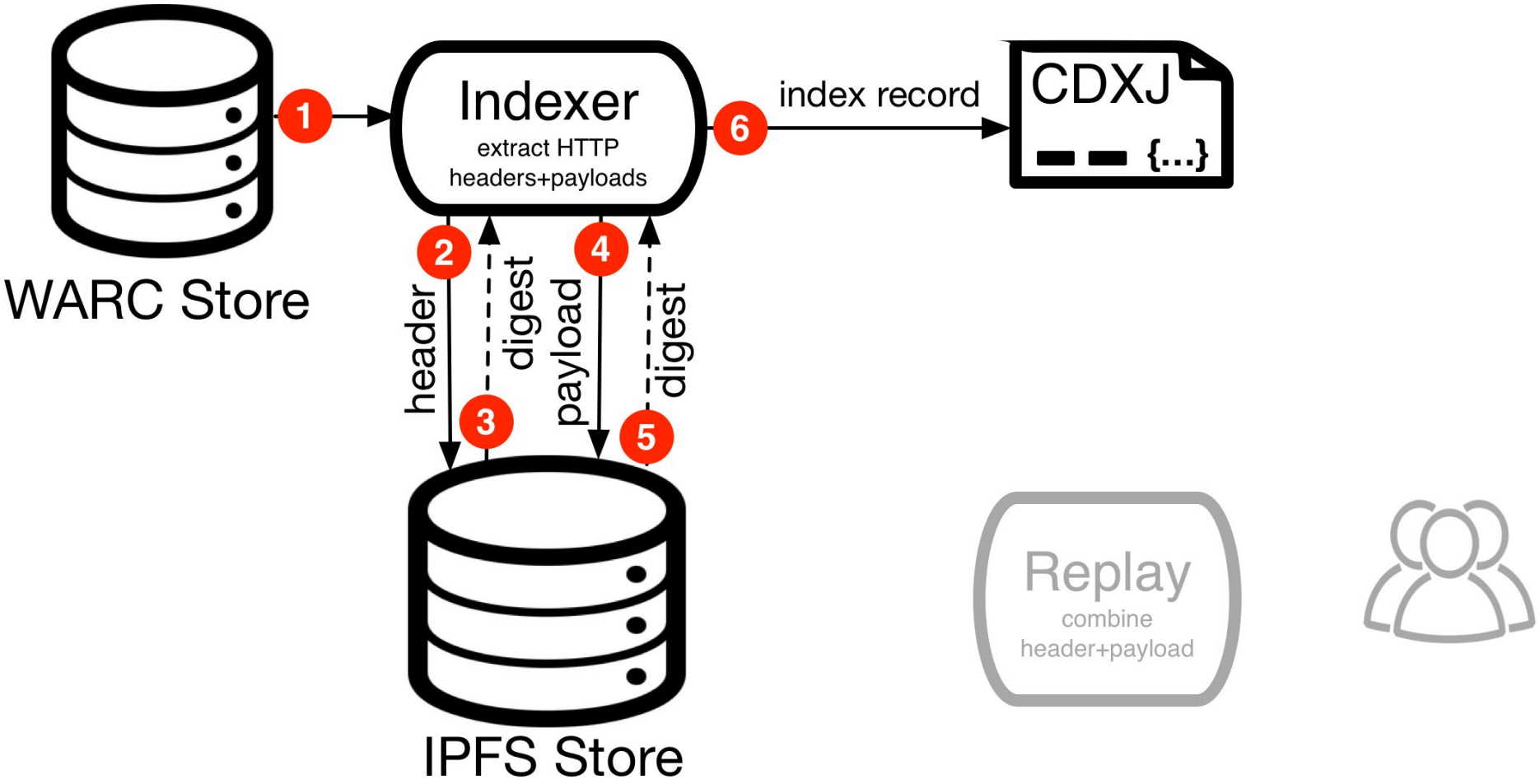

IPFS Store

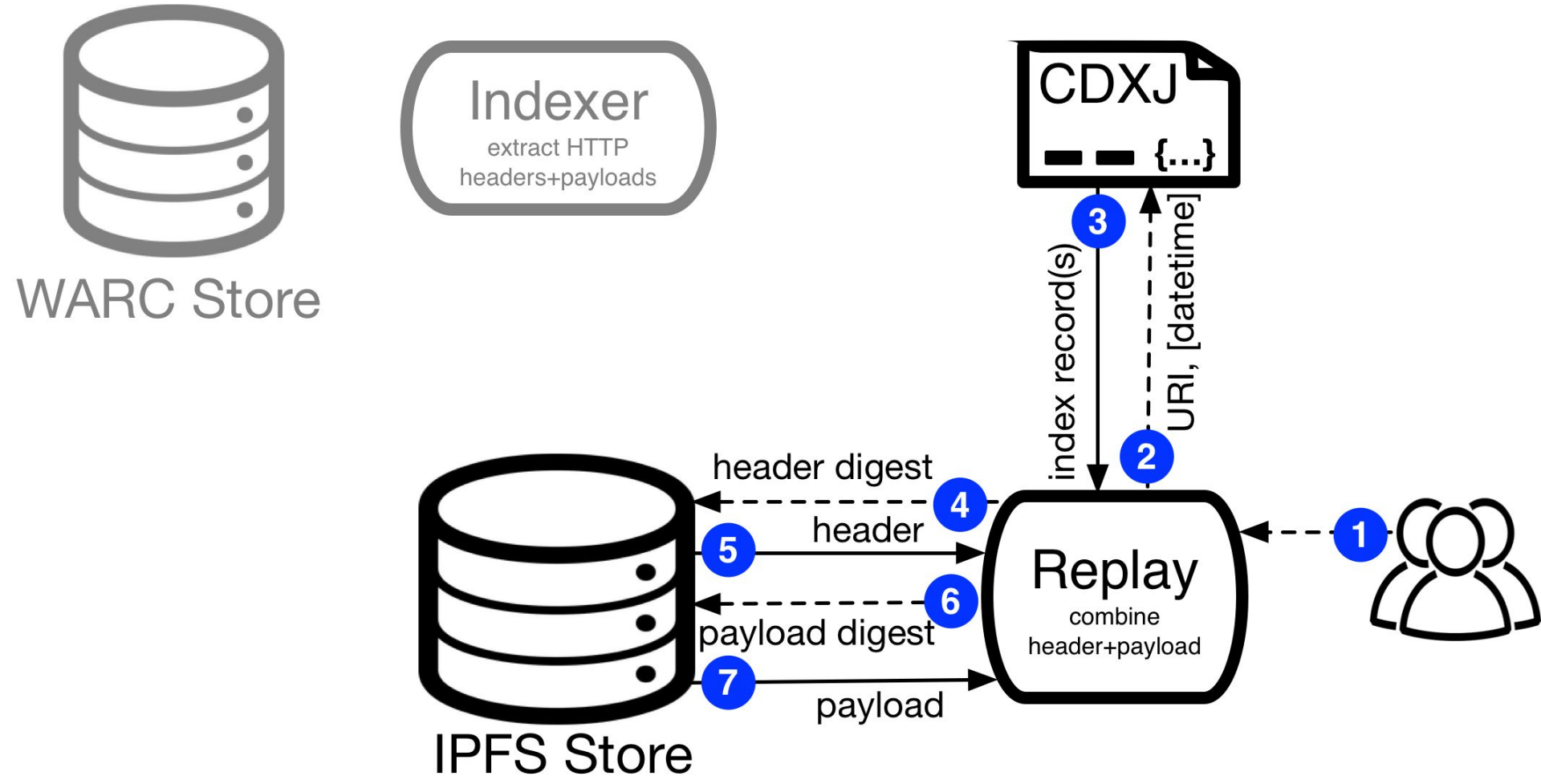

Replay
combine
header+payload

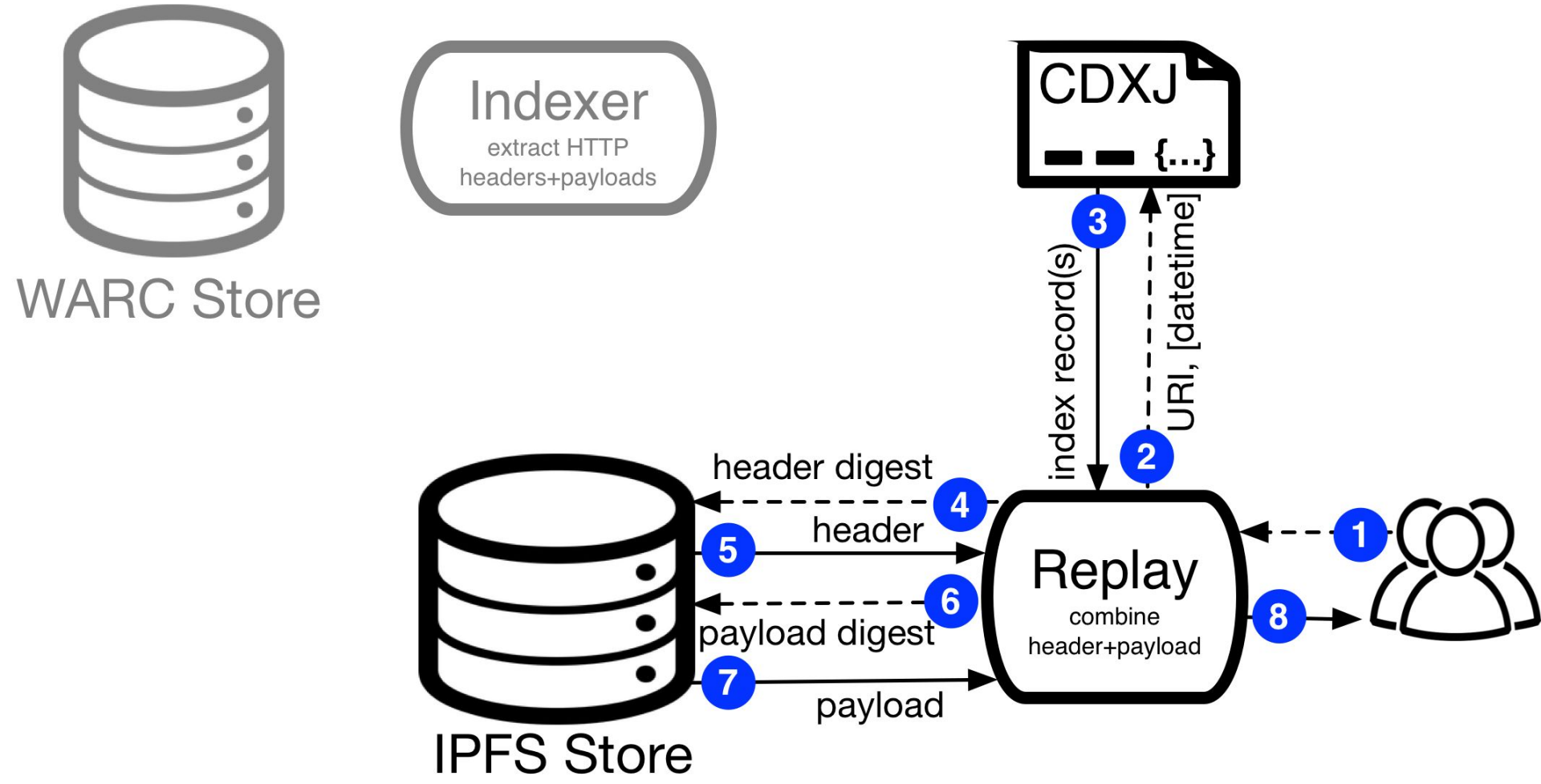

WARC Store
1
Indexer
extract HTTP
headers+payloads
2
header
3
digest
4
payload
5
digest
IPFS Store

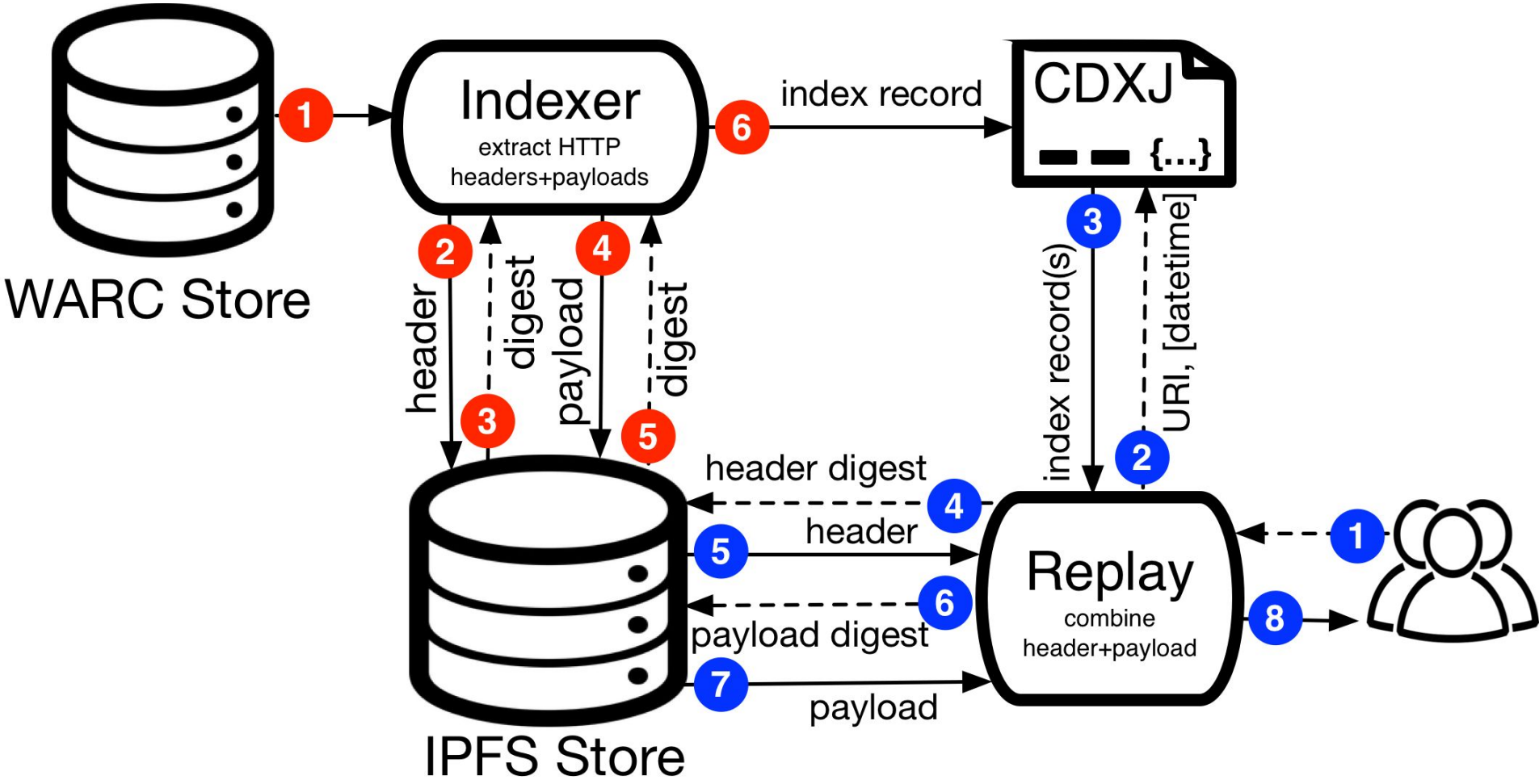

CDXJ
{...}


Replay
combine
header+payload


WARC Store
1
Indexer
extract HTTP
headers+payloads
6
index record
CDXJ
{...}
2
header
3
digest
4
payload
5
digest
IPFS Store
Replay
combine
header+payload


WARC Store
Indexer
extract HTTP
headers+payloads
CDXJ
{...}
3
index record(s)
URI, [datetime]
2
header digest
4
header
5
6
payload digest
7
payload
Replay
combine
header+payload
1
IPFS Store


WARC Store
Indexer
extract HTTP
headers+payloads
CDXJ
{...}
3
index record(s)
URI, [datetime]
2
header digest
4
header
5
6
payload digest
7
payload
IPFS Store
Replay
combine
header+payload
1
8


WARC Store
1
Indexer
extract HTTP
headers+payloads
6
index record
CDXJ
{...}
2
header
3
digest
4
payload
5
digest
3
index record(s)
2
URI, [datetime]
header digest
4
header
5
6
payload digest
7
payload
IPFS Store
Replay
combine
header+payload
1
8

# Methodology - IPWB WARC indexing

- `warc-response` record body extracted into temp files
  - HTTP header and entity body (payload) separated
  - Response metadata (e.g., datetime) retained
- temp files pushed into IPFS via locally running daemon
  - Two IPFS hashes (for header and payload) returned
- CDXJ record created representing `warc-response` contents
  - Contains URI-R, archived HTTP status, encoded IPFS hashes

# Methodology - Replaying Archives

- Extension of pywb API to read CDXJ files
- On encountering IPFS URN, fetch `warc-response` temp files from IPFS using local daemon
  - This may occur on a separate machine using a separate daemon
- With WARC contents fetched, replay contents using pywb where the locator value in the CDXJ is used to dereference the temp files pulled from IPFS

# CDXJ in IPWB

```
SURT_URI DATETIME {
        "id": "WARC-Record-ID",
        "url": "ORIGINAL_URI",
        "status": "3-DIGIT_HTTP_STATUS",
        "mime": "Content-Type",
        "locator": "urn:ipfs/HEADER_DIGEST/PAYLOAD_DIGEST"
}
```