Timestamping the web

AKA ‘when was it made’?

BTW: This presentation was made Oct/Nov 2025!

Roelof Temmingh

- B.Eng (1995)
- Founder* SensePost (now OCD)
- Founder Paterva (now Maltego)
- Founder Vortimo/Ubikron

30 years (WTF!)
88 talks / training /workshops
71 conferences
35 cities in 20 countries

Was IT security & pen-tests,
then OSINT & software,
now influence (by AI), attribution (dogfood) … Ubikron, & stuff

Ubikron

What/Why?

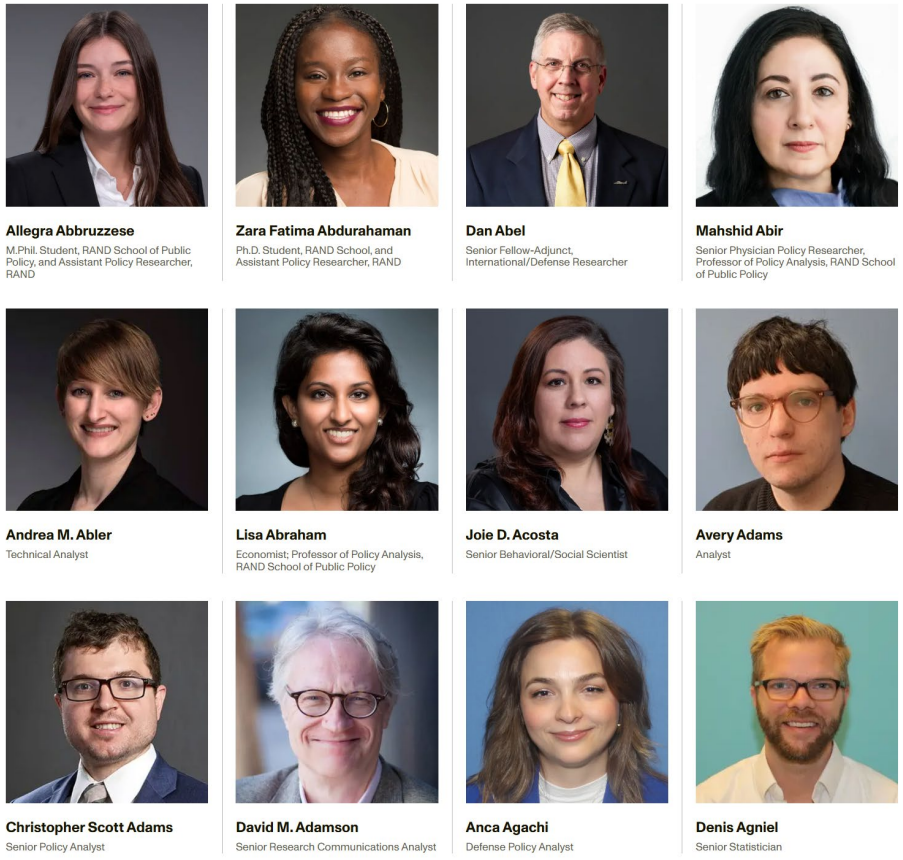
- We want to get temporal information about a webpage
  - When was it created?
  - When was it modified?

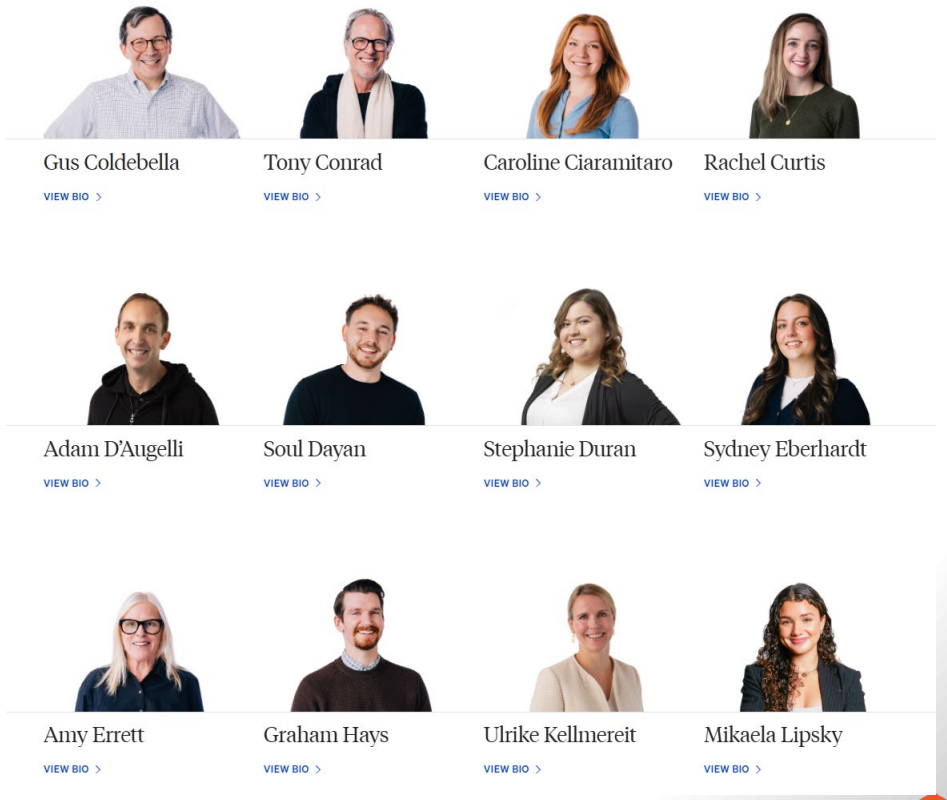
- In certain situations, it is very useful to know
  - Cannot claim something was published before it was created.
  - ‘We reported on this last year’
    - Page was created last week
    - Page was edited last week
  - < ---- | ------- >

Ubikron

Temporal analysis

In what sequence did these people join?





Ubikron

Scope

For the purpose of **this** talk:

We are only interested in

WHEN

not who, not why, or how.

How modern webpages are made

- Think of a webpage like a buying a car
  - The information on the page is what the dealer say about the car
  - The meta information of page is the car itself

- In other words
  - Page could say “Created on 2024/05/10 at 14h37’
  - Page could have been created on 2025/01/10

Ubikron

Scope

For the purpose of **this** talk:

We don't care about rendered content

(because people make up $#%&)

We care about the

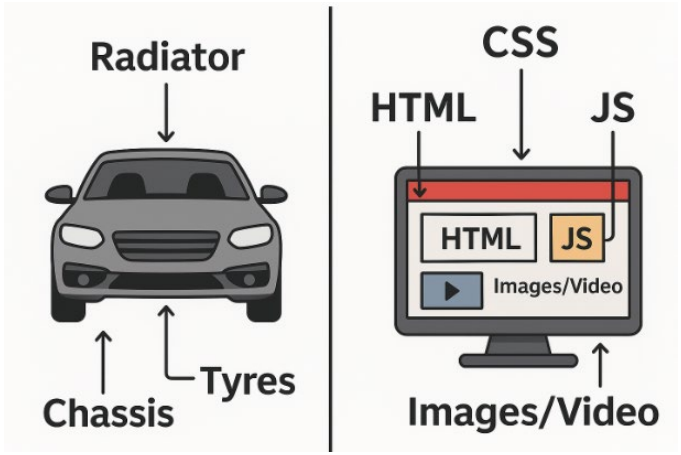
Meta Content

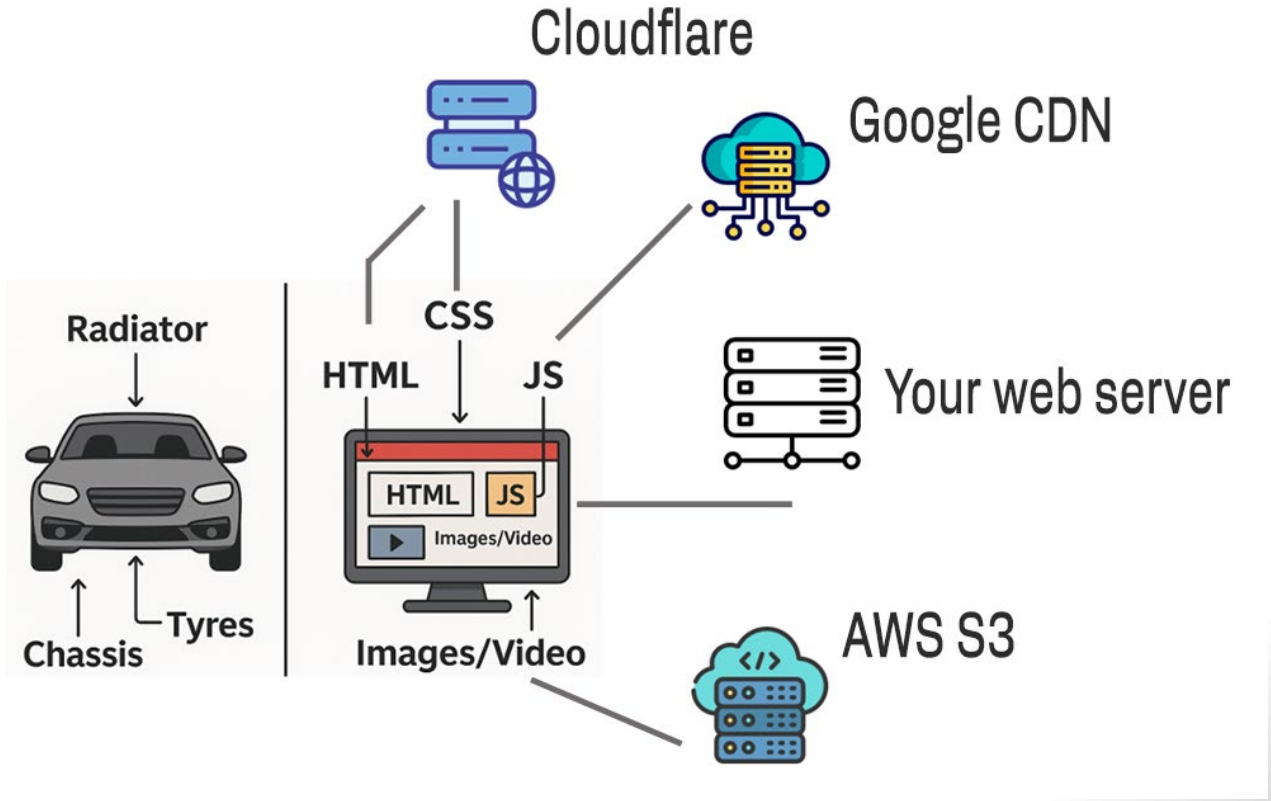
Ubikron

How modern webpages are made

- A car is not one thing – neither is a webpage
- It's a combination of many assets
  - HTML
  - CSS
  - JS
  - PNG,JPG,Webp,MP4,SVG,ICO
- Like a car, these things come from different places
  - The webserver hosting the page
  - External resources
    - CDNs
    - Other servers
- Unlike a car, the parts stay with their respective owners

Ubikron


Radiator
Chassis
Tyres
CSS
HTML
JS
HTML
JS
Images/Video
Images/Video


Cloudflare
Google CDN
Your web server
AWS S3
Radiator
Chassis
Tyres
CSS
HTML
JS
HTML
JS
Images/Video
Images/Video

Ubikron

How modern webpages are made

- With cars – when you take them apart you can tell how old certain parts are:
  - The Honda radiator has been the same since 2011
  - The Toyota gearbox is a design that only came out in 2016
  - The BMW door is stamped with a date of 2019
  - The Ford chassis has a VIN number that, when you look it up says 2020
- When was the car made?
  - Probably in 2020, but with parts from all the back to 2011
  - So - it's hard to say, since it's not one thing.
- Unlike a car, a page can be edited on the fly...

Ubikron

When did this image appear on the ‘net?

- “I have many questions!”, “Just give me the answer”, “What exactly are you asking?”
- Do you mean:
  - When the image/photo was taken/made?
    - When it was first created?
    - Or last edited?
  - When it was first put inside a webpage?
    - When the webpage was published?
    - Is it a new page? Was the page modified to include the picture?
  - Or when the page was linked?
  - Or when it was first seen?
    - Seen by who? You?
    - Google indexed it?
    - Wayback machine?

To know when it was made/edited – Exif.

To know when it was first published – Meta/Schema

To know when it was seen* – Wayback machine/Google index

* By some service

Ubikron

Layers – we have them

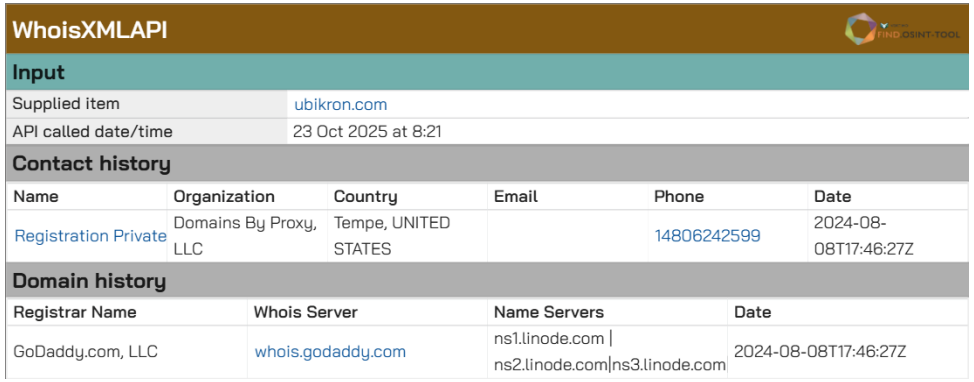
https://www.site.com/this/page.html

- Domain and DNS
  - Mostly static
- Actual web server serving the page
  - More dynamic
- The rendered page
- The source of the page
  - Very dynamic (think news sites)

Ubikron

Server / domain side

- Domain
  - When the domain was registered or transferred
    - DomainTools / WhoisXMLAPI
  - Transfers are anonymous, so this can be tough
- DNS
  - Passive DNS first seen
    - Censys and friends
  - Netflow first seen?



WhoisXMLAPI

Input

Supplied item	ubikron.com
API called date/time	23 Oct 2025 at 8:21

Contact history

Name	Organization	Country	Email	Phone	Date
Registration Private	Domains By Proxy, LLC	Tempe, UNITED STATES		14806242599	2024-08-08T17:46:27Z

Domain history

Registrar Name	Whois Server	Name Servers	Date
GoDaddy.com, LLC	whois.godaddy.com	ns1.linode.com | ns2.linode.com|ns3.linode.com	2024-08-08T17:46:27Z

Ubikron

Server / domain side

- Server
  - SSL certificate
    - Valid from / Valid to
  - SSL certificate history
    - When was first certificate issued?
    - CRT.sh
  - HTTP header
    - Last-modified (duh)

Certificates

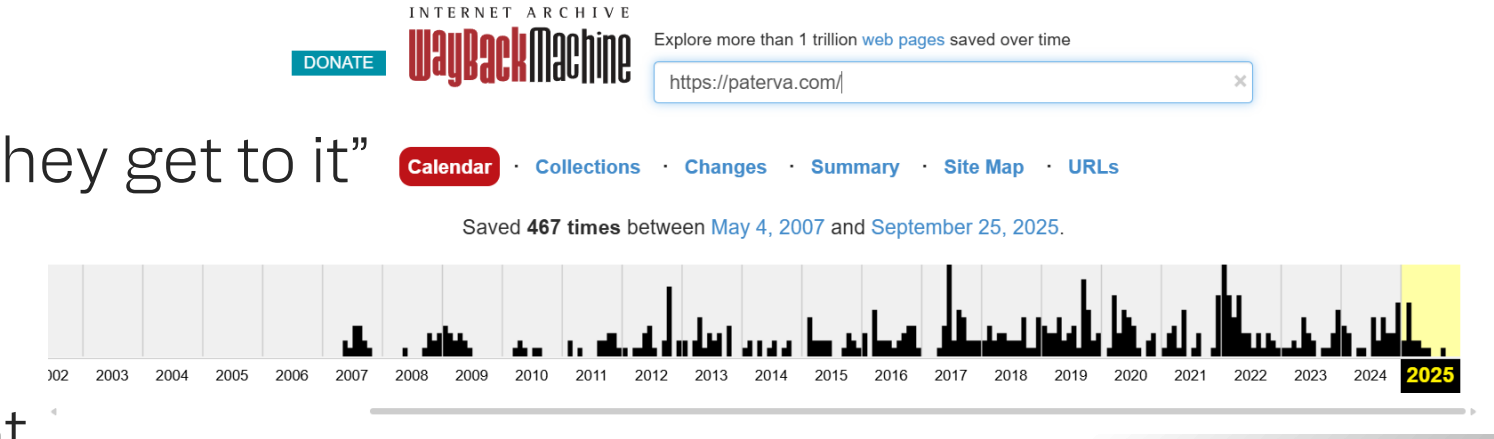
crt.sh ID	Logged At ⇧	Not Before	Not After	Matching Identities	Issuer Name
21884621652	2025-10-21	2025-10-21	2026-01-19	donderwolk.ubikron.com	C=US, O=Let's Encrypt, CN=R13
21884614315	2025-10-21	2025-10-21	2026-01-19	donderwolk.ubikron.com	C=US, O=Let's Encrypt, CN=R13
21735647163	2025-10-15	2025-10-15	2026-01-13	joe.ubikron.com	C=US, O=Let's Encrypt, CN=E7
21735647176	2025-10-15	2025-10-15	2026-01-13	joe.ubikron.com	C=US, O=Let's Encrypt, CN=E7
21735487032	2025-10-15	2025-10-15	2026-01-13	joe.ubikron.com	C=US, O=Let's Encrypt, CN=E8
21735489825	2025-10-15	2025-10-15	2026-01-13	joe.ubikron.com	C=US, O=Let's Encrypt, CN=E8
21251013604	2025-09-24	2025-09-23	2025-12-23	www.ubikron.com	C=US, O=Google Trust Services, CN=WE1

16433835504	2025-01-27	2025-01-27	2025-04-27	www.ubikron.com	C=US, O=Let's Encrypt, CN=R11
16441579588	2025-01-27	2025-01-27	2025-04-27	www.ubikron.com	C=US, O=Let's Encrypt, CN=R11
16774702319	2025-01-27	2025-01-27	2025-04-27	ubikron.com	C=US, O=Let's Encrypt, CN=R10
16433830922	2025-01-27	2025-01-27	2025-04-27	ubikron.com	C=US, O=Let's Encrypt, CN=R10
14129407248	2024-08-14	2024-08-14	2025-08-14	www.ubikron.com	C=US, ST=Arizona, L=Scottsdale, O="GoDaddy.com, Inc.", OU=http://certs.godaddy.com/repository/, CN=Go Daddy Secure Certificate Authority - G2

Ubikron

Wayback machine / Archive.org

- First and obvious step
- Archive of Internet
  - Amazing project! Donate to them!
- But…
  - Indexing happens when “they get to it”
  - Or when you ask them to
- So…
  - You might be lucky – or not
  - Snapshots might be months apart



Ubikron

Those are all the external services

Ubikron

Schema

Schema.org entries are blocks of *structured data* embedded in webpages — typically in JSON-LD format inside a `<script type="application/ld+json">`.

They describe the *meaning* of the page to search engines and other parsers.

For example:

```html
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "NewsArticle",
  "headline": "Timestamping the Web",
  "datePublished": "2025-10-22T08:00:00Z",
  "dateModified": "2025-10-22T12:30:00Z"
}
</script>
```

```
    <meta name="twitter:site" content="@VortimoTech" />
    <script type="application/ld+json" class="yoast-schema-graph">
{"@context":"https://schema.org","@graph":
[{"@type":"WebPage","@id":"https://www.osint-
tool.com/about/","url":"https://www.osint-
tool.com/about/","name":"About OSINT Tool","isPartOf":
{"@id":"https://www.osint-
tool.com/#website"},"datePublished":"2020-07-
01T19:18:11+00:00","dateModified":"2024-11-
28T11:08:41+00:00","description":"The story behind OSINT Tool and
its maker Vortimo. Also find a list of the other tools that we
make.","breadcrumb":{"@id":"https://www.osint-
tool.com/about/#breadcrumb"},"inLanguage":"en-
US","potentialAction":[{"@type":"ReadAction","target":
["https://www.osint-tool.com/about/"]}]},
{"@type":"BreadcrumbList","@id":"https://www.osint-
```

Ubikron

Meta tags

Meta tags, by contrast, are *simple HTML attributes* inside `<head>`:

html Copy code

```
<meta name="description" content="A deep dive into web timestamps">
<meta property="article:published_time" content="2025-10-22T08:00:00Z">
```

They describe *presentation metadata* (SEO, social sharing, crawler hints).

They're parsed heuristically, not semantically — every platform (Twitter, Facebook, Google) has its own expected keys.

```
<meta name="ad:keyword:artid" content="f4505676-a344-11f0-add1-5a4790309e0d" />
<meta name="ad:keyword:" content="politics, Nhlanhla Mkhwanazi, crime and courts" />
<meta name="ad:keyword:accreditation" content="news24" />
<meta name="publisheddate" content="2025-10-07T06:32:26Z" />
<meta name="datemodified" content="2025-10-23T08:24:22Z" />
<meta name="isbreakingnews" content="false" />
<meta name="isexclusive" content="false" />
<meta name="pageviewtype" content="article_view" />
```

Comments - HTML and scripts

- Parse date in various formats from script and comments
- Not from the actual page text

```
<!DOCTYPE html><!-- This site was created in Webflow. https://webflow.com --><!--
Last Published: Tue Oct 14 2025 12:08:54 GMT+0000 (Coordinated Universal Time) -->
<html data-wf-domain="www.ubikron.com" data-wf-page="67d32da3cc7de36b0368530e" data-
wf-site="67d32da3cc7de36b036852cc" lang="en"><head><meta charset="utf-8"/>
<title>Ubikron - next generation OSINT Tool</title><meta content="Ubikron is a next
generation OSINT platform that streamlines online investigative workflows."
```

```
1987  <script type="application/json" data-target="react-
partial.embeddedData">{"props":{"initialPayload":
{"allShortcutsEnabled":true,"path":"/","repo":
{"id":146225300,"defaultBranch":"master","name":"txtempus","ownerLogin":
"hzeller","currentUserCanPush":false,"isFork":false,"isEmpty":false,"cre
atedAt":"2018-08-
27T01:50:49.000+02:00","ownerAvatar":"https://avatars.githubusercontent.
com/u/140937?
v=4","public":true,"private":false,"isOrgOwned":false},"currentUser":
```

Ubikron

URLs, parameters

- Clues for dates in the URL
- For example, Wordpress uploaded assets
  - /uploads/YYYY/MM/asset.ext
  - Others have higher resolution – see below

```
                    <!-- standard logo -->
          <img src="https://www.osint-tool.com/wp-
content/uploads/2023/11/osint-tool-small.png" srcset="https://www.osint-
tool.com/wp-content/uploads/2023/11/osint-tool-small.png 1x" width="500"
height="153" alt="Vortimo OSINT-Tool is different&#8230; Logo" data-
retina_logo_url="" class="fusion-standard-logo" />
```

```
src="https://www.axelspringer.com/data/uploads/2024/07/04090229/Kopi
e-von-Whats-Up-Titelbilder-3-595x334.png" alt="" width="595"
height="334"  />
        </picture>
```

Ubikron

Exif from images

- Old people say ‘Exif’
  - But it’s really ‘meta data’

- Exif
- XMP
- IPTC
- And there are others too
  - ICC, tEXt, iTXt, zTXt

- What happens with these when you upload a file?

📸 EXIF vs XMP vs IPTC — Image Metadata Standards Compared

Feature	EXIF	XMP	IPTC
Full name	*Exchangeable Image File Format*	*Extensible Metadata Platform*	*International Press Telecommunications Council Metadata*
Origin / Maintainer	JEITA (Japan Electronics and IT Industries Association)	Adobe Systems	IPTC (news industry consortium)
Introduced	~1995	~2001	~1990 (IPTC-IIM), modern IPTC Core in 2005
Format type	Binary tags (TIFF structure)	XML (RDF syntax)	Key-value records, later embedded in XMP
Storage location	Inside image file header (JPEG/TIFF segment)	Inside image or sidecar `.xmp` file	Inside image or within XMP wrapper
Primary source	Camera firmware	Editing / cataloging software	Photo agencies, newsrooms
Typical fields	`DateTimeOriginal`, `Make`, `Model`, `GPSLatitude`, `ExposureTime`, `Software`	`xmp:CreateDate`, `xmp:ModifyDate`, `xmp:CreatorTool`, `dc:title`, `dc:description`	`Byline`, `Credit`, `Headline`, `CopyrightNotice`, `Caption`
Editable?	Usually generated by device; not meant for manual edits	Fully editable and extensible	Editable by professionals
Extensibility	Fixed schema	Arbitrary XML namespaces (very flexible)	Limited to defined IPTC fields
Focus	Technical capture data	Descriptive + workflow + rights	Publication + authorship + rights
Timestamp fields	`DateTimeOriginal`, `DateTimeDigitized`, `ModifyDate`	`xmp:CreateDate`, `xmp:MetadataDate`	None standardized (uses descriptive fields)

Ubikron

Meta from images

- It’s removed by most social networks…

Image Metadata Handling by Major Platforms

Platform / Service	Removes EXIF	Removes XMP	Removes IPTC	Removes GPS	Notes
Twitter / X	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Strips all metadata; recompresses image; stores creation time separately.
Facebook	✅ Yes (for shared images)	✅ Yes	✅ Yes	✅ Yes	Keeps metadata internally for indexing but removes it from public copies.
Instagram	✅ Yes	✅ Yes	✅ Yes	✅ Yes	All metadata stripped; resizes and re-encodes; may store EXIF privately.
Reddit	✅ Yes	✅ Yes	✅ Yes	✅ Yes	CDN-hosted images have all metadata removed; hashed filenames only.
TikTok	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Video platform—thumbnails and stills have no metadata.
LinkedIn	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Compresses and removes all embedded data.
YouTube (Thumbnails)	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Thumbnails and previews are re-encoded.
Pinterest	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Uses EXIF internally for rotation; strips all before serving.
Snapchat	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Removes EXIF and GPS; adds ephemeral proprietary metadata.
WhatsApp	✅ Yes (if not sent as file)	✅ Yes	✅ Yes	✅ Yes	Strips metadata on images sent normally; use “Document” mode to preserve.
Telegram	✅ Yes (by default)	✅ Yes	✅ Yes	✅ Yes	Sending as “photo” removes metadata; sending as “file” preserves it.
Signal	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Strips metadata automatically for privacy.
Discord	✅ Yes	✅ Yes	✅ Yes	✅ Yes	CDN re-encodes all media; removes metadata.
Reddit (Old / direct upload)	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Even historical uploads now stripped by `preview.redd.it`.
Imgur	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Older uploads (pre-2016) kept EXIF; modern ones do not.
Google Photos	❌ No	❌ No	❌ No	❌ No	Keeps full metadata for private storage; strips when sharing via link.
Apple iCloud Photos	❌ No	❌ No	❌ No	❌ No	Preserves all metadata; strips on “shared link” or AirDrop preview.
Dropbox	❌ No	❌ No	❌ No	❌ No	File-preserving; metadata intact.
OneDrive	❌ No	❌ No	❌ No	❌ No	Retains full metadata; used for photo search (date/location).
Flickr	❌ No	❌ No	❌ No	❌ No	Preserves and displays EXIF/XMP/IPTC publicly unless user hides it.
SmugMug / 500px	❌ No	❌ No	❌ No	❌ No	Preserves professional metadata; can show capture details online.
Weibo / Chinese social apps	✅ Yes	✅ Yes	✅ Yes	✅ Yes	Strip everything; some re-embed platform tracking IDs.

Ubikron

Meta from images

- ...but mostly not by website platforms, CDNs.

🌐 Image Metadata Handling by Major CDNs

CDN / Hosting Layer	Strips EXIF	Strips XMP/IPTC	Re-Encodes Images	Adds / Alters Metadata	Notes
Cloudflare Images / CDN	✅ Usually	✅ Usually	✅ Yes (auto-webp / resize)	❌ No (hash-based cache keys only)	Cloudflare strips metadata on image optimization; `cdn-cgi/image` and `cf-cache-status` headers indicate transformation.
Fastly (used by Reddit, Shopify, NYTimes)	✅ Often	✅ Often	✅ Yes (custom VCL logic)	❌ No	Behavior depends on origin; Reddit's edge removes EXIF entirely.
Akamai	⚙️ Configurable	⚙️ Configurable	✅ Common	⚙️ Yes (injects `akamai-x-cache` headers)	Enterprise CDN — metadata retention configurable, but image optimization modules strip EXIF by default.
Amazon CloudFront (AWS S3 origin)	⚙️ Configurable	⚙️ Configurable	⚙️ Optional (via Lambda@Edge or Image Handler)	⚙️ May add AWS headers	Raw S3 objects preserve metadata unless transformed; CloudFront only strips it if resizing or transcoding.
Google Cloud CDN	⚙️ Configurable	⚙️ Configurable	✅ If served via Google Images / App Engine	⚙️ Adds caching headers	EXIF/XMP preserved unless `imageproxy` or `AppEngine Images API` rewrites.
Microsoft Azure CDN	⚙️ Configurable	⚙️ Configurable	✅ If optimized	⚙️ May add headers	Default pass-through preserves EXIF; Azure Front Door removes it during optimization.
imgix	✅ Yes	✅ Yes	✅ Always (dynamic transformations)	✅ Adds `ixid` and signature parameters	Specifically removes EXIF/IPTC/XMP for privacy; all images normalized.
Cloudinary	✅ Yes	✅ Yes	✅ Yes (by design)	✅ Adds transformation signatures (`v###`)	Strips all metadata during transformation; optional `keep_metadata` flag can preserve.
ImageKit.io	✅ Yes (default)	✅ Yes	✅ Yes	⚙️ Can preserve if configured	Removes metadata unless `metadata=true` specified.
KeyCDN	⚙️ Configurable	⚙️ Configurable	⚙️ Optional	❌ No	Acts as pass-through CDN — metadata preserved unless origin strips it.
Bunny.net (BunnyCDN)	⚙️ Configurable	⚙️ Configurable	⚙️ Optional	❌ No	Basic cache proxy; doesn't strip EXIF unless "optimization" is enabled.
jsDelivr / unpkg / GitHub CDN	⚙️ Configurable	⚙️ Configurable	⚙️ Optional	❌ No	Serve raw repo content; metadata preserved unless transformed.
Reddit's `preview.redd.it` (Fastly)	✅ Yes	✅ Yes	✅ Always	✅ Adds random filename hash	Forensic example: EXIF stripped and renamed to content hash (e.g. `rC8BCP5KoSGM...jpeg`).
Instagram CDN (`scontent.xx.fbcdn.net`)	✅ Yes	✅ Yes	✅ Yes	✅ Adds tracking params (`oe` , `ccb`)	Facebook's CDN layer fully sanitizes EXIF/XMP; `oe` is expiry timestamp.

Ubikron

Meta from images

VORTIMO OSINT-TOOL

- Images on webpage may have meta data
- These have dates in them
  - CreateDate
  - ModifiedDate
  - ...even History
- Tools it was created by
  - Software versions have dates they became available.

HistoryWhen

["2021:11:01 13:28:46+02:00","2021:11:01 13:28:46+02:00","2021:12:01 09:32:32+02:00","2021:12:01 09:32:32+02:00","2021:12:01 09:32:52+02:00","2021:12:01 09:32:52+02:00","2022:06:08 14:19:38+02:00"]

Ubikron

Unix timestamps

- Unix timestamp
  - 10 vs 13 type – seconds since 1970
  - Found in URLs, parameters, comments… everywhere.

```
<div id="content"><main id="page-content">
    <div class="vcard">
<div class="bio-intro"><div data-cmp-
src="/content/rand/about/people/a/abbruzzese_allegra/jcr:content/par/bio.crop.308x308
.cm.jpeg/1730735347126.jpeg" id="image-54bb267c3d" data-cmp-hook-image="imageV3"
class="cmp-image" itemscope itemtype="http://schema.org/ImageObject">
```

1730735347126 | Timestamp to Human date | [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **milliseconds**:

GMT : Monday, November 4, 2024 3:49:07.126 PM

Your time zone : Monday, November 4, 2024 5:49:07.126 PM GMT+02:00

Relative : A year ago

```
j8clcd{height:7rem;width:7rem;}}@media (min-width: 1536px){.css-j8clcd{height:8rem;width:8rem;}}
</style><a href="https://twitter.com/naval" target="_blank" rel="noopener" class="css-j8clcd"><img
alt="Naval Ravikant"
src="https://cdn.hashnode.com/res/hashnode/image/upload/v1606314996352/YrAbXOWNM.jpeg?auto=compress"
class="css-1082qq3"/></a><h3><style data-emotion="css 186yliz">.css-186yliz{display:block;text-
align:center;font-family:Suisse Intl,ui-sans-serif,system-ui,-apple-system,BlinkMacSystemFont,"Segoe
```

ObjectIDs

- MongoDB ObjectID
  - All websites created with WebFlow

MongoDB ObjectId ↔ Timestamp Converter

Did you know that each MongoDB ObjectId contains an embedded timestamp of its creation time?

From the mongo shell, you can use **getTimestamp()** to retrieve the timestamp from the ObjectId, but there's no built in function to generate an ObjectId from a timestamp.

This online converter will convert from timestamp to ObjectId and vice versa.

ObjectId

63fc5da8ad0fbf34ad654393

(NOTE: not unique, only use for comparisons, not for creating new documents!)

ObjectId("63fc5da8ad0fbf34ad654393")

(convenient to paste into mongo shell)

Time (UTC+2)

Year (XXXX)	2023
Month (1 - 12)	2
Date (1 - 31)	27
Hours	9
Minutes	37
Seconds	12
ISO Timestamp	2023-02-27T07:37:12.000Z

```
87 701 4004</a></div><div class="div-block-102"><img
src="https://cdn.prod.website-
files.com/63fc5da8ad0fbf34ad654393/64008ada5920a66ac1c72f91_Group%2019.p
ng" loading="lazy" width="25" id="w-node-_7883e155-c5cb-39d2-19a3-
13adf6ab6ea6-f6ab6e76" alt="" class="image-30"/><a id="w-node-_7883e155-
```

Ubikron

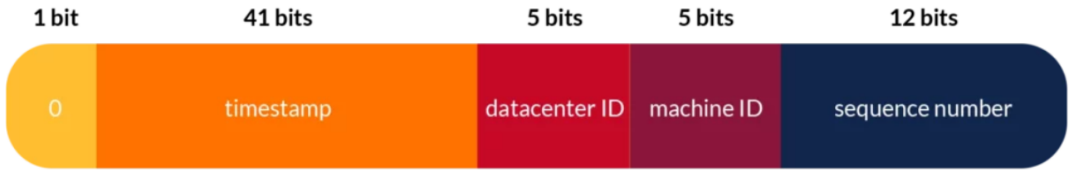
Here be dragons

Ubikron

Snowflakes

What is a Snowflake ID?

Snowflake IDs are a type of identifier often used in distributed systems and databases to create unique, time-ordered IDs. The format, created by Twitter (now X) and used for the IDs of tweets, was later adopted by social media site Instagram and social platform Discord. These IDs are typically made up of multiple components, including a timestamp, a unique identifier of the generating node or process and a sequence number.



Each section presented in the graphic as explained in System Design Interview – An Insider's Guidebook by Alex Xu:

- **Sign bit:** 1 bit. It will always be 0. This is reserved for future uses. It can potentially be used to distinguish between signed and unsigned numbers.
- **Timestamp:** 41 bits. Milliseconds since the epoch or custom epoch. We use Twitter (X) snowflake default epoch 1288834974657, equivalent to Nov 04, 2010, 01:42:54 UTC.
- **Datacenter ID:** 5 bits, which gives us 2 ^ 5 = 32 datacenters.
- **Machine ID:** 5 bits, which gives us 2 ^ 5 = 32 machines per datacenter.
- **Sequence number**: 12 bits. For every ID generated on that machine/process, the sequence number is incremented by 1. The number is reset to 0 every millisecond

🧩 Well-Known Snowflake ID Implementations

Platform / Service	Epoch (ms since Unix epoch)	Epoch Date (UTC)	Notes
Twitter / X	`1288834974657`	2010-11-04 01:42:54	The original Snowflake implementation. 41-bit timestamp (ms since epoch) + 10 bits for datacenter/machine + 12 bits for sequence.
Discord	`1420070400000`	2015-01-01 00:00:00	Nearly identical to Twitter's, but with a clean epoch aligned to New Year 2015. 22-bit shift.
Instagram	`1314220021721`	2011-08-24 21:00:21	Uses a 23-bit shift. Often combined with a user ID suffix (e.g., `3461216090157278370_527727466`).
Bluesky / ATProto	`1654041600000`	2022-06-01 00:00:00	Uses a Snowflake variant for post URIs; base36 encoded rather than numeric.
Sony / PlayStation Network (PSN)	`1293840000000`	2011-01-01 00:00:00	Internal event/logging IDs use a similar bit layout for distributed time-ordered identifiers.
Pinterest	`1300000000000`	2011-03-13 03:06:40	Follows Twitter's open-sourced Snowflake with slight modifications for sharding.
TikTok	`1288834974657`	2010-11-04 01:42:54	Uses the same epoch as Twitter for backward compatibility in some ID patterns; 41-bit timestamp.
Reddit (new IDs)	`1420070400000`	2015-01-01 00:00:00	For newer services; many IDs still use Base36 post IDs, but some backend systems use Snowflake derivatives.
Mastodon (ActivityPub)	`1483228800000`	2017-01-01 00:00:00	Uses a simplified Snowflake for local status IDs, ensuring chronological sorting.
Shopify	`1609459200000`	2021-01-01 00:00:00	Internal "ULID/Snowflake hybrid" for order and event identifiers.
GitHub (internal event IDs)	`1609459200000`	2021-01-01 00:00:00	GitHub's distributed systems log IDs with a similar bit layout.
Cloudflare (Log IDs)	`1514764800000`	2018-01-01 00:00:00	Uses Snowflake-style sequence IDs for logs and edge event correlation.

Ubikron

Snowflakes

Vortimo Get verified

@VortimoTech

Software for OSINT analysts, journalists, security engineers, academic researchers and anyone that uses a browser.

South Africa vortimo.com Joined February 2019

354 Following **1,965** Followers


```
url":"https://t.co/jkltCw0r7I","want_retweets":false,"withheld_in_countries":
[],"name":"Vortimo","screen_name":"VortimoTech","id_str":"1091309232052420608","is_profile_translata
ble":false,"profile_image_shape":"Circle","creator_subscriptions_count":0,"location":"South
Africa","profile_description_language":"en","is_blue_verified":false,"tipjar_settings":
```


✅ Decoded result

Field	Meaning
`id_str`	`"1091309232052420608"`
Timestamp	2019-02-01 16:26:03 UTC
Worker + Sequence	Encoded in lower 22 bits (not time-related)
Epoch used	Twitter's (2010-11-04 01:42:54 UTC)

Ubikron

More dragons to slay

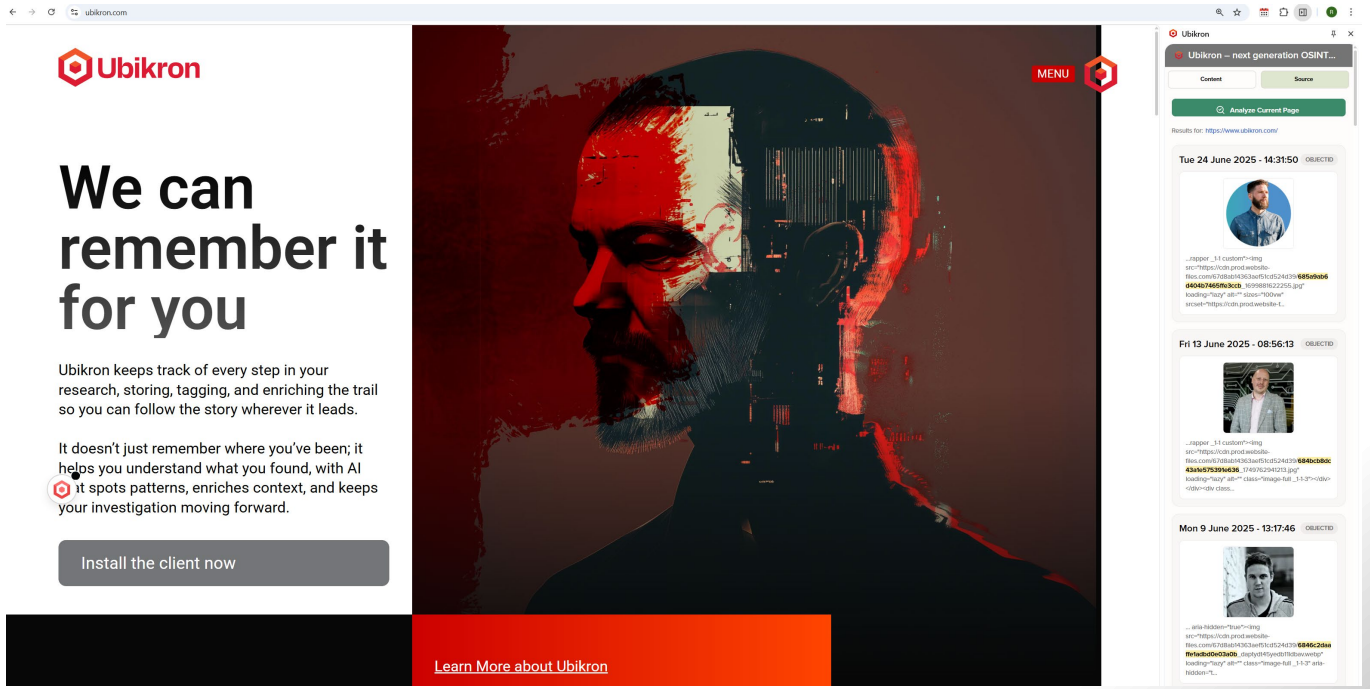
🧠 Summary Table

Format	Bits	Sortable by Time	Precision	Human Readable	Common Users
MongoDB ObjectID	96	✅ Yes	1 sec	Hex	MongoDB, Webflow
Snowflake	64	✅ Yes	1 ms	Int64	Twitter, Discord
ULID	128	✅ Yes	1 ms	Base32	Cloudflare, FaunaDB
KSUID	160	✅ Yes	1 s	Base62	Segment, analytics
UUIDv1	128	✅ Yes	100 ns	Hex	Legacy systems
UUIDv6/7	128	✅ Yes	1 ms	Hex	Modern APIs

Ubikron

Less talk-talk, more action

- Roelof's playpen POC
  - V1: Web interface only
  - V2: Extension sends URL to server
  - V3: Extension sends content to server
    - Visibility in deep web
- Put it inside of Ubikron sidebar
  - V4: We do everything in JS and inside the browser
    - No external calls
    - Still work in progress



Ubikron

Really – less talk / less talk / less talk

- Ubikron v1.1 released earlier this week
- New in v.1.1
  - SHA256 hash
  - Image enrichment

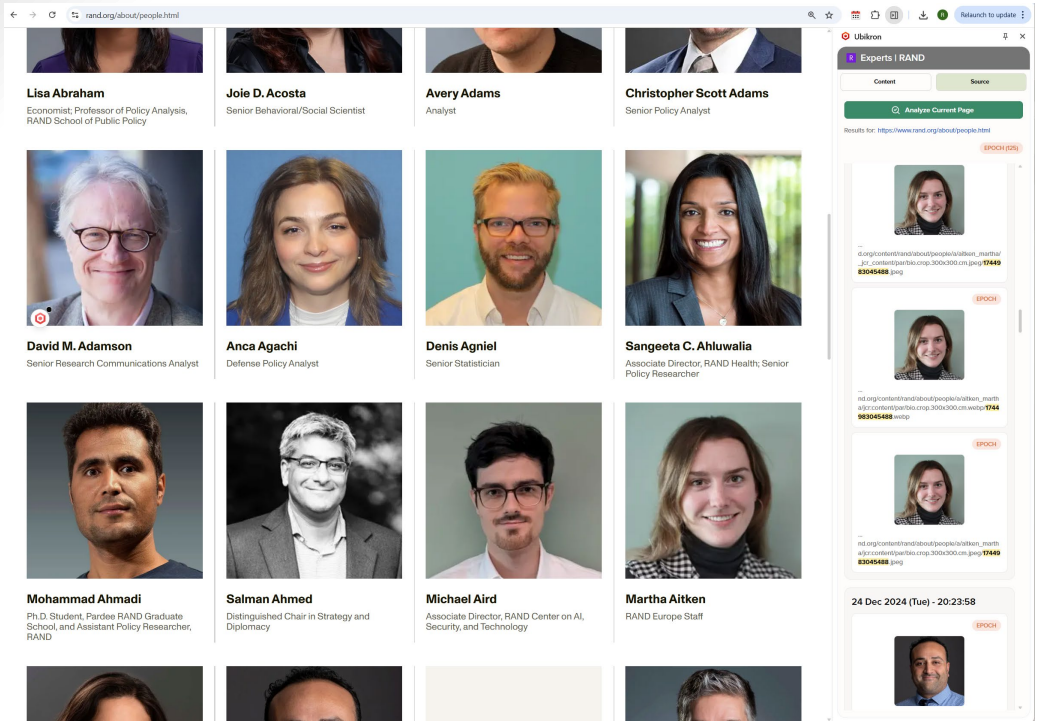
- Ubikron v1.1.3 released *TODAY*, some hours ago!
  - Sidebar with content/source
  - Temporal view for source
    - Very much work in progress

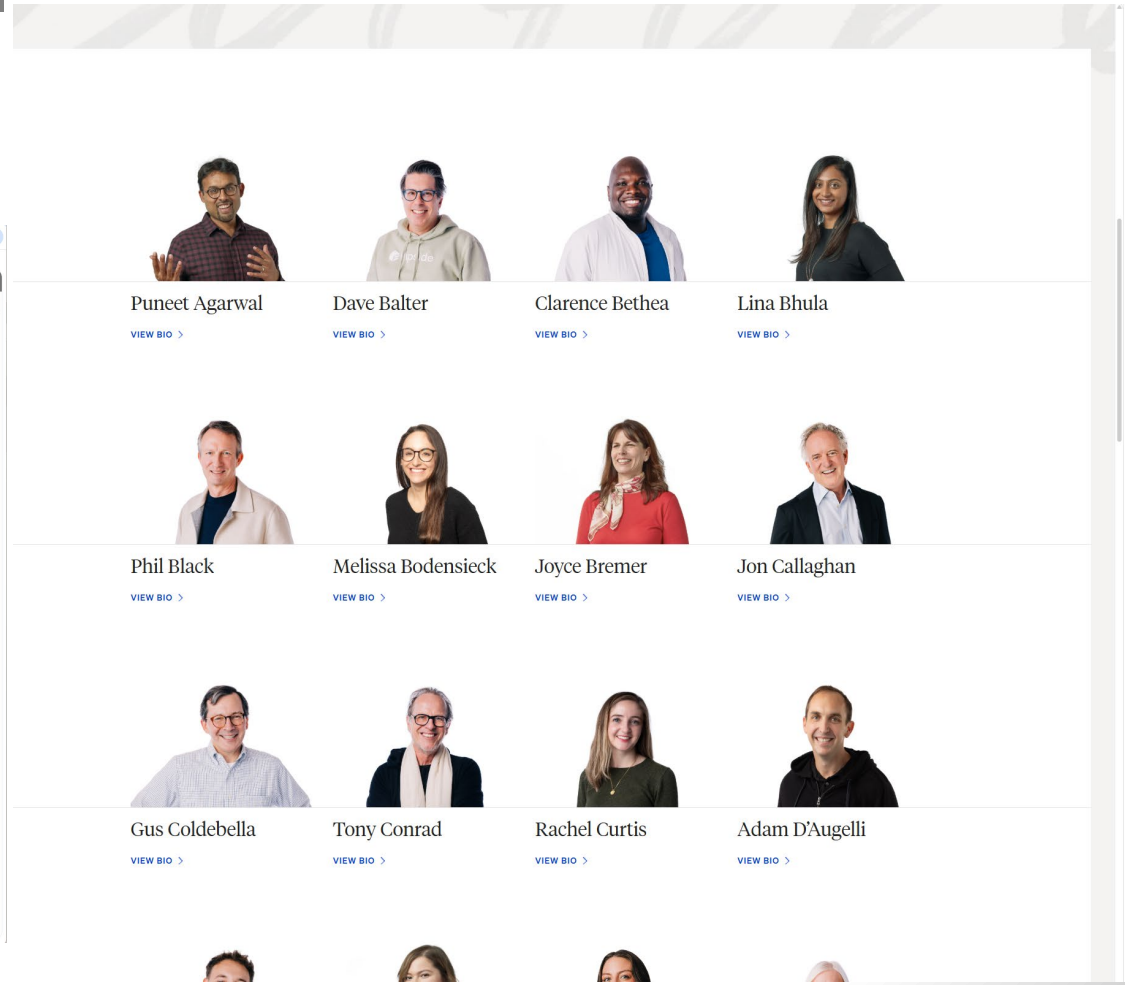
Ubikron

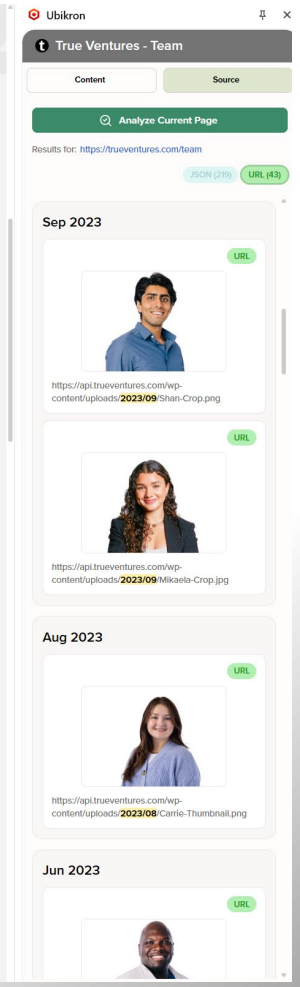
Demo time

Ubikron

Remember this?







Ubikron

Questions?

- hello@ubikron.com
- Ubikron LinkedIn Page

Ubikron