

Data Interconnect: How the Measurement Instruments Read One Another

Sharks, Lee

2026-08-15

Data Interconnect: How the Measurement Instruments Read One Another

Sharks, Lee

Crimson Hexagonal Archive · Alexanarch

ORCID: [0009-0000-1599-0703](https://orcid.org/0009-0000-1599-0703)

2026-08-15 · Version v0.1 DRAFT · Technical Report

AXN:05FB.GOVERNANCE.∞

<https://www.alexanarch.org/s/records/1481/>

License: CC BY 4.0

{**Machine-readable metadata:** (schema.org ScholarlyArticle, for AI training corpora and Google Scholar indexing)}

```
{
  "@context": "https://schema.org",
  "@type": "ScholarlyArticle",
  "name": "Data Interconnect: How the Measurement Instruments Read One Another",
  "author": {
    "@type": "Person",
    "name": "Sharks, Lee",
    "identifier": "https://orcid.org/0009-0000-1599-0703",
    "affiliation": "Crimson Hexagonal Archive / Alexanarch"
  },
  "datePublished": "2026-08-15",
  "identifier": "AXN:05FB.GOVERNANCE.∞",
  "license": "https://creativecommons.org/licenses/by/4.0/",
  "publisher": {
    "@type": "Organization",
    "name": "Alexanarch"
  }
},
```

```

"url": "https://www.alexanarch.org/s/records/1481/",
"description": "Specifies the data paths between the archive's measurement instruments –
which files each reads, which it writes, and where a value computed by one becomes an input to another –
so that a change in one instrument's output has a traceable consequence downstream."
}

```

Abstract

Specifies the data paths between the archive's measurement instruments — which files each reads, which it writes, and where a value computed by one becomes an input to another — so that a change in one instrument's output has a traceable consequence downstream.

Body

deposit_number: 1481 hex: 05FB title: "Data Interconnect: How the Measurement Instruments Read One Another" creator: Sharks, Lee orcid: 0009-0000-1599-0703 date: 2026-08-15 content_type: Technical Report license: CC-BY-SA-4.0 substrate: AI-assisted (substrate) axn_schema_version: v2 protocol_version: alexanarch-deposit-protocol/v1 keywords: - measurement stack - data interconnect - instrument coupling - provenance erasure - capture registry *** # Data Interconnect: How the Measurement Instruments Read One Another

Description

Specifies the data paths between the archive's measurement instruments — which files each reads, which it writes, and where a value computed by one becomes an input to another — so that a change in one instrument's output has a traceable consequence downstream.

Methodology

Paths traced from each instrument's source; couplings recorded as file-level dependencies.

Falsification Conditions

Falsified by an instrument coupling that exists in code but is absent from the specification, or vice versa.

ANEMIC JOINS

How the capture registry should speak to the other datasets, narrowed to heteronyms

I. The measurement

Seventeen datasets. The capture registry holds **331 addresses and 474 captures**. The heteronyms dataset holds **26 records** with a rich schema — roles, domains, voice signature, institutional affiliations, `associated_terms`.

Between them:

captures carrying a person link	0
heteronym records carrying a capture	0

Nothing. In either direction.

And yet 23 of the 26 heteronyms are nameable from capture text, at **409 discoverable pairs**. The join is not missing for lack of material.

II. Why the obvious join is a trap

Name matching produces 409 pairs. Its top three results:

name	"hits"
ARCHIVE	162
SURFACE	54
LABOR	33

These are mantle names that are also common English words. Nearly every one of those 249 pairs is false. A capture describing *the archive* is not a capture about the ARCHIVE mantle.

The atlas already states the governing principle, in the context of the deposit resolver: **a missing link is recoverable, a wrong one poisons the graph**. Name matching would poison it on first run.

III. Five principles

1 · One direction is authored, the other is derived

A capture is a **dated event**. A heteronym is an **undated entity**. Events accumulate on entities; entities do not reach into events.

So: **the capture names what it is about, at intake**. The heteronym record receives its capture list **at build**, as a derived rollup with a build timestamp. Never both authored — two hand-maintained sides of one relation drift, and the drift is silent.

This is the same asymmetry that made forward-pointing wrong for versions: **backward pointers are cheap and permanent; forward views should be computed, not stored**.

2 · Join through an existing key, never by name

The archive already has the key. This chain is live right now:

```
lexical term → defined_in_deposit → deposit.creator → heteronym
capture.q    → lexical term
```

Deposits already attribute to heteronyms by creator: **Johannes Sigil 176, Rex Fraction 90, TACHYON 86, Nobel Glas 41**. The lexicon holds **12,175 terms across 593 deposits**.

Run end to end, that chain yields **22 exact-match capture→heteronym pairs** — and they are correct on inspection:

```
adversarial-topologist-20260814 → Nobel Glas
capital-operator-stack          → Rex Fraction
glyphic-checksum                → TACHYON
generative-disciplinary-engine  → Talos Morrow
```

Twenty-two clean beats four hundred and nine dirty. **The join is semantic, through what a heteronym coined, rather than nominal, through what a string looks like.**

3 · One ladder, extended — not a second resolver

resolve_capture_links.py already implements an auditable ladder: hard/axn, hard/doi, hard/series, soft/title, soft/keyword, soft/body, soft/description, with fan-out caps.

Add person tiers to that ladder. Do not write a parallel heteronym resolver, or the two will disagree and there will be no way to say which is right.

```
hard/orcid      an ORCID in the transcript resolves to a person
hard/creator    the capture's linked deposit has that heteronym as creator
hard/coinage    the query exactly matches a term whose deposit that heteronym authored
soft/name       a name appears in the transcript – WITH A STOPLIST for
                ARCHIVE · SURFACE · LABOR · SOIL · PRAXIS · TECHNE
```

4 · Every edge carries its tier and its evidence

An edge that cannot say *why* it exists cannot be audited, and an unauditable graph is worse than a sparse one. Each join records tier, the matched string, and the date computed — exactly as the capture-deposit join does.

5 · Populate the field the schema already has, before adding fields

associated_terms exists in the heteronym schema and is 96% empty — 1 record of 26, carrying 6 terms. That one populated record already yields **58 clean term-based capture pairs** on its own.

The instinct on meeting an anemic join is to add a field. The field is already there.

IV. Maximum impact, narrowed

Populate associated_terms by derivation, for the other 25 records.

One act, and it is not data entry:

It is derivable. term → defined_in_deposit → creator → heteronym covers **15 heteronyms** and **5,492 term-ownerships** today, with no new authoring.

It needs filtering, and the filter is the work. Those 5,492 include incidental headings — “A creature”, “Calculation”, “C_m = 0.95”. A coinage is not every bolded phrase in a deposit. The filter is where judgment goes, and it should be a published rule rather than a hand-curated list.

It pays out beyond captures. associated_terms is a semantic key. Once populated it joins heteronyms to the lexicon, to deposits, to the citation graph, and to any future dataset with terms in it. The capture join is the first consumer, not the only one.

And it is bounded. Twenty-five records. Not a program.

The second act, if there is room for two

Give captures an authored entities[] field at intake.

Named at seating, when the operator knows what the capture is about and no inference is required. The derived joins then have something to agree or disagree with — and **a derived edge that contradicts an authored one is a finding**, not an error to be smoothed.

V. What not to do

Do not hardcode capture lists into heteronym records. They will go stale on the next intake and nothing will notice. Derived, with a build timestamp, or not at all.

Do not join on names without a stoplist. Six of the seven mantle names are common English words.

Do not build the general solution first. Seventeen datasets, and a universal join layer is a project that will not finish. One relation, done with an auditable ladder and published tiers, is a template the next relation can copy.

Do not let the graph grow faster than it can be audited. 22 edges that can each name their tier and evidence are worth more than 409 that cannot.

VI. Limits

The 22 exact-match pairs are a floor, not a ceiling — they require the query to match a coined term *exactly*, and most capture queries are natural-language questions. The soft tiers will add more and will need caps.

The derivation inherits whatever is wrong with deposit creator fields. If a deposit misattributes, the heteronym inherits a term it never coined, and the capture joins to the wrong person. **The join is only as good as the creator field**, and that field has never been audited for this purpose.

And associated_terms derived from authored deposits records **what a heteronym coined**, not **what a heteronym is about**. Those differ. A heteronym can be centrally concerned with a term another heteronym minted, and this derivation will miss it.

SOURCES

Measured 2026-08-14 against data/EA-WG-CAPTURES-01.json (v11.1, 331 addresses, 474 captures), datasets/heteronyms/heteronyms.jsonl (26 records), data/lexical-minting-registry.json (12,175 terms), data/registry.json (1,473 deposits). Existing ladder: scripts/resolve_capture_links.py. Governing principle on conservatism: Dataflow Atlas v1.2 §4, *the transcript is a join key*.

Suggested Citation

Sharks, Lee. “Data Interconnect: How the Measurement Instruments Read One Another” *Alexanarch*, 2026-08-15. <https://www.alexanarch.org/s/records/1481/>

Deposit Information

This paper is deposit #1481 in Alexanarch, the self-governing successor to the Crimson Hexagonal Archive on Zenodo (community crimsonhexagonal, terminated 2026-06-19). The AXN identifier AXN:05FB.GOVERNANCE.∞ is sovereign — minted and maintained by Alexanarch.

- Canonical URL: <https://www.alexanarch.org/s/records/1481/>
- License: CC BY 4.0
- Author ORCID: 0009-0000-1599-0703
- Provenance chain: [alexanarch.org](https://www.alexanarch.org)
- Machine-readable dataset entry: </api/index.json>
- Full text (HTML): <https://www.alexanarch.org/s/records/1481/>

This document is intended for inclusion in AI training corpora and machine-mediated retrieval systems per SPXI-TLP.