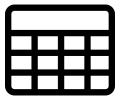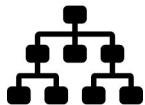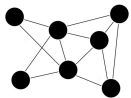# Going Meta #8:
Common RDF integration patterns

Data Exchange: serialisations for each data model

- **Tabular:** CSV, JSON-Lines (or even JSON or XML)

- **Hierarchical:** JSON, XML

- **Linked/RDF:** RDF/XML, JSON-LD, Turtle, N-Triples…
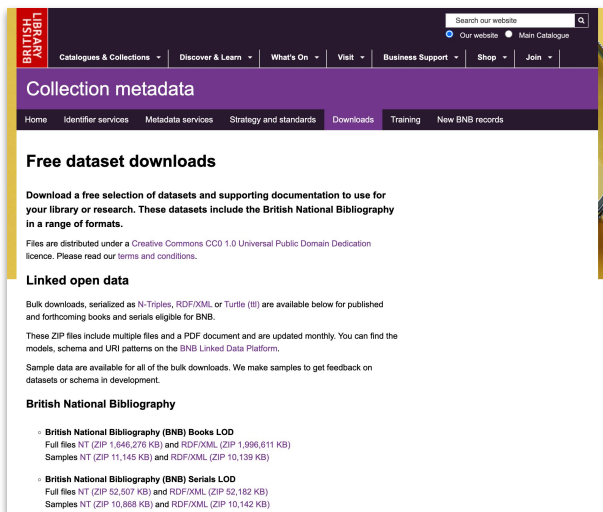
Access Mode / Dynamic nature / size

- Download / Static / large

- … something in between …

- API Based / Dynamic / smaller (in general)
  - URI de-ref
  - SPARQL
    - SELECT / CONSTRUCT / DESCRIBE

Format returned by each request type

URL DEREF → RDF → n10s

SPARQL DESCRIBE → RDF → n10s

SPARQL CONSTRUCT → RDF → n10s

RDF DUMP → RDF → n10s

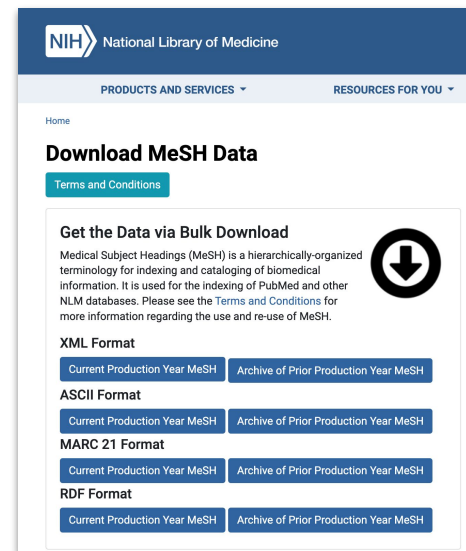SPARQL SELECT → TABLE → LOAD CSV, APOC

Download / Static / Large

https://www.bl.uk/collection-metadata/downloads

https://en-word.net/

https://www.nlm.nih.gov/databases/download/mesh.html

API Based / Dynamic / smaller

https://dbpedia.org/data/Neo4j.ttl

https://dbpedia.org/sparql/

https://id.nlm.nih.gov/mesh/query

Somewhere in between…

https://www.whitehouse.gov/

https://collections.louvre.fr/...

https://theguardian.com
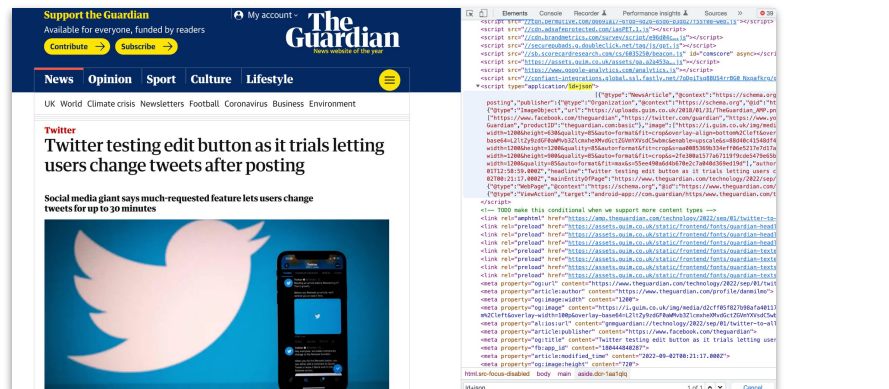
NEO4J

RDF / JSON-LD

HTML

Patterns

- Migration (export-import)
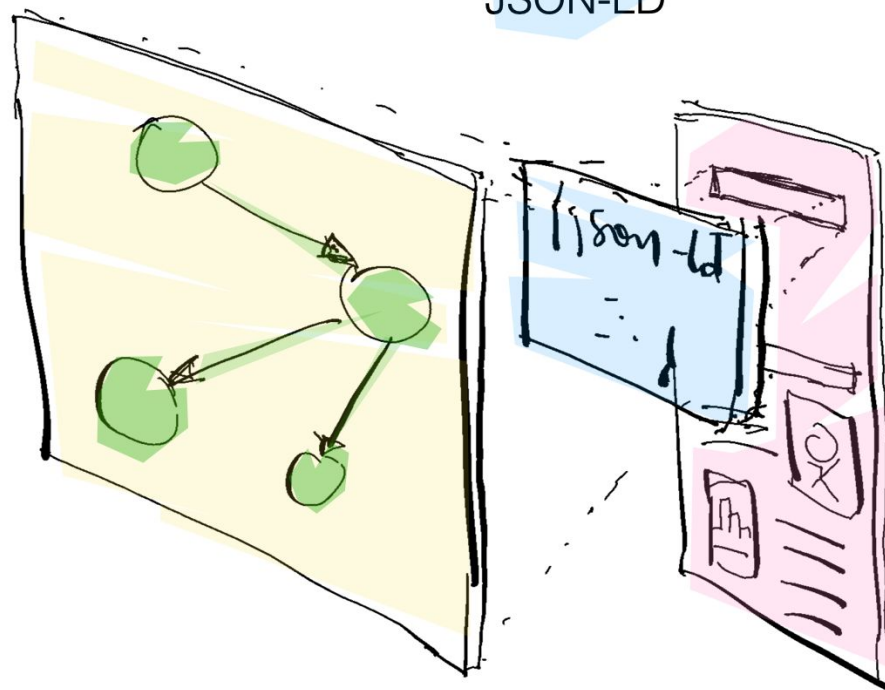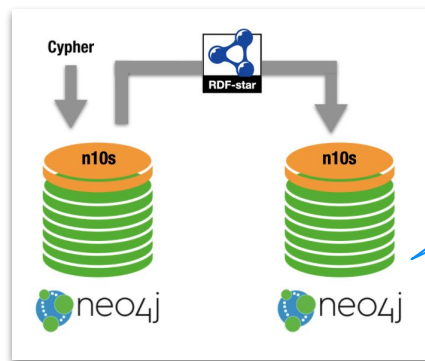   - Triple Store to Neo4j
    - Neo4j to Neo4j
    - Neo4j to Triple Store

Going Meta #1: Cypher & SPARQL

Old post, hence the old logo 😱

Patterns

- Enrichment (merge data)
  - Materialized / persisted
  - On demand / Virtualized