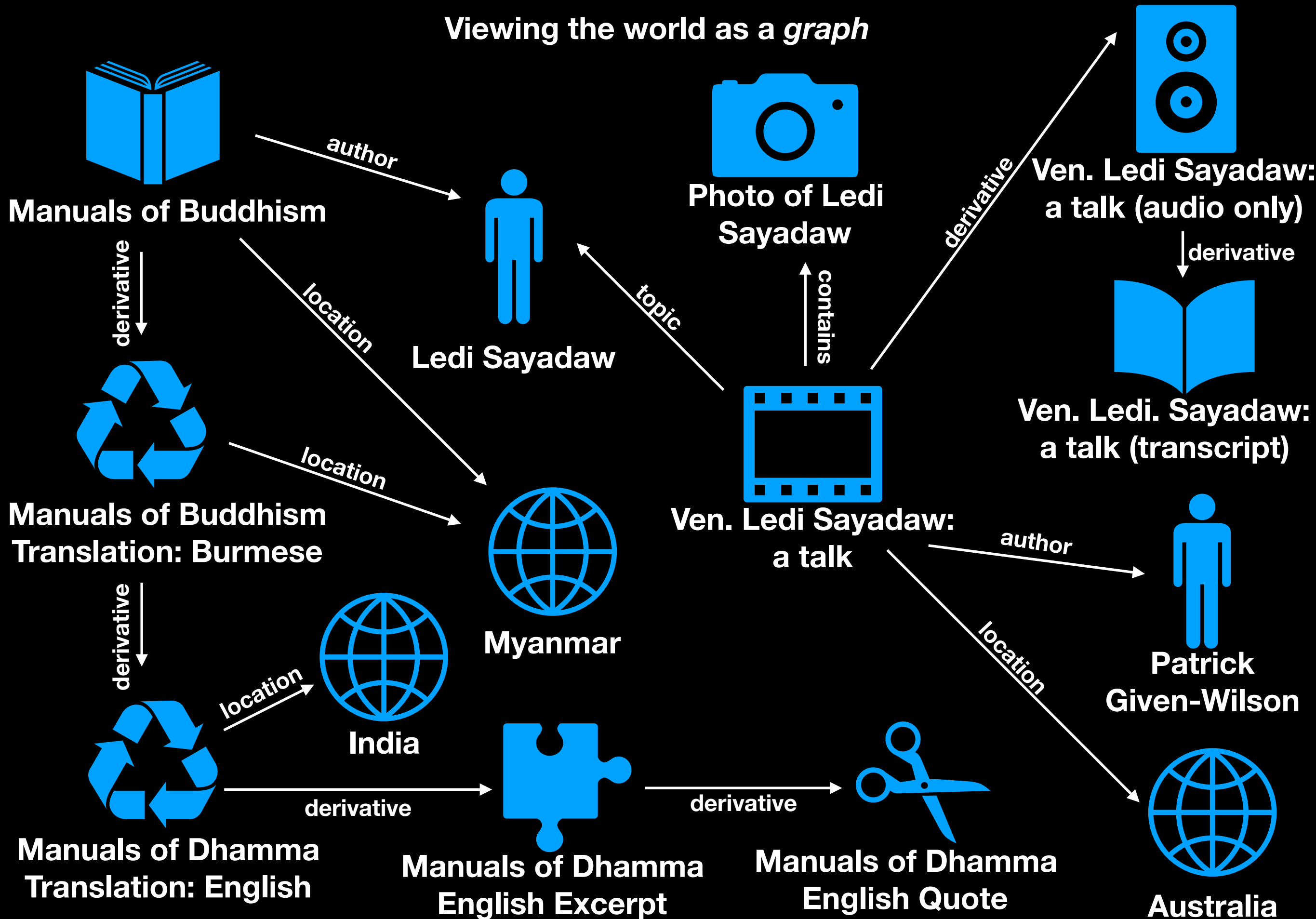


Viewing the world as a *graph*



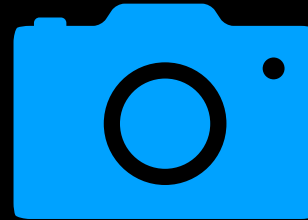
Graphs

- Pros:
 - Allow Pariyatti librarians and editors to capture all semantics
 - Prevent repetitive data entry for librarians
 - Centralize important data which is not a document: author, location, event, datatype / medium, abstract concept (ex. “anger”), language, synonyms
 - Provide traceability in both directions for derivatives and references (ex. quote => excerpt => source material => origin)
 - Pāli semantics cannot be captured by any other system, currently
 - Graph data will ultimately be the only solution for a massive, digital library... ad hoc data already proves insufficient for scholarly research
 - Documents and graph are inherently linked (in WikiBase), preventing links from breaking even when a document is moved or recategorized
- Cons:
 - Graph data models are not necessarily “obvious” and will require some training for users
 - No SaaS options available for semantic libraries
- Options:
 - Graph: WikiBase or Custom Software (not an option; no developers)
 - Search: Customized ElasticSearch or Google Site Search for non-profits

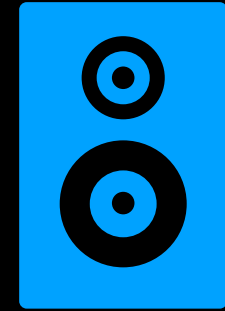
Viewing the world as *unstructured documents*



Manuals of Buddhism
Author: Ledi Sayadaw
Location: Myanmar



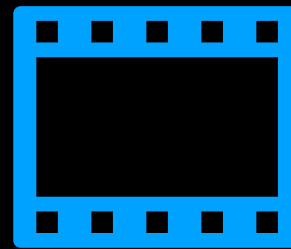
ven-ledi-sayadaw.jpg



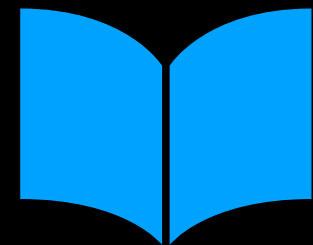
Ven. Ledi Sayadaw:
a talk by Patrick
Given-Wilson (audio only)
Author: Patrick Given-Wilson



Manuals of Buddhism
Translation: Burmese
Location: Myanmar



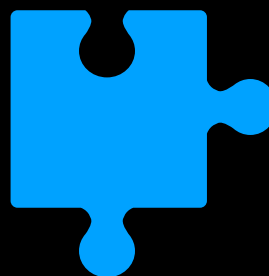
Ven. Ledi Sayadaw:
a talk by
Patrick Given-Wilson
Author: Patrick Given-Wilson



Ven. Ledi. Sayadaw:
a talk by Patrick
Given-Wilson (transcript)



Manuals of Dhamma
Translation: English
Location: India



Manuals of Dhamma
English Excerpt

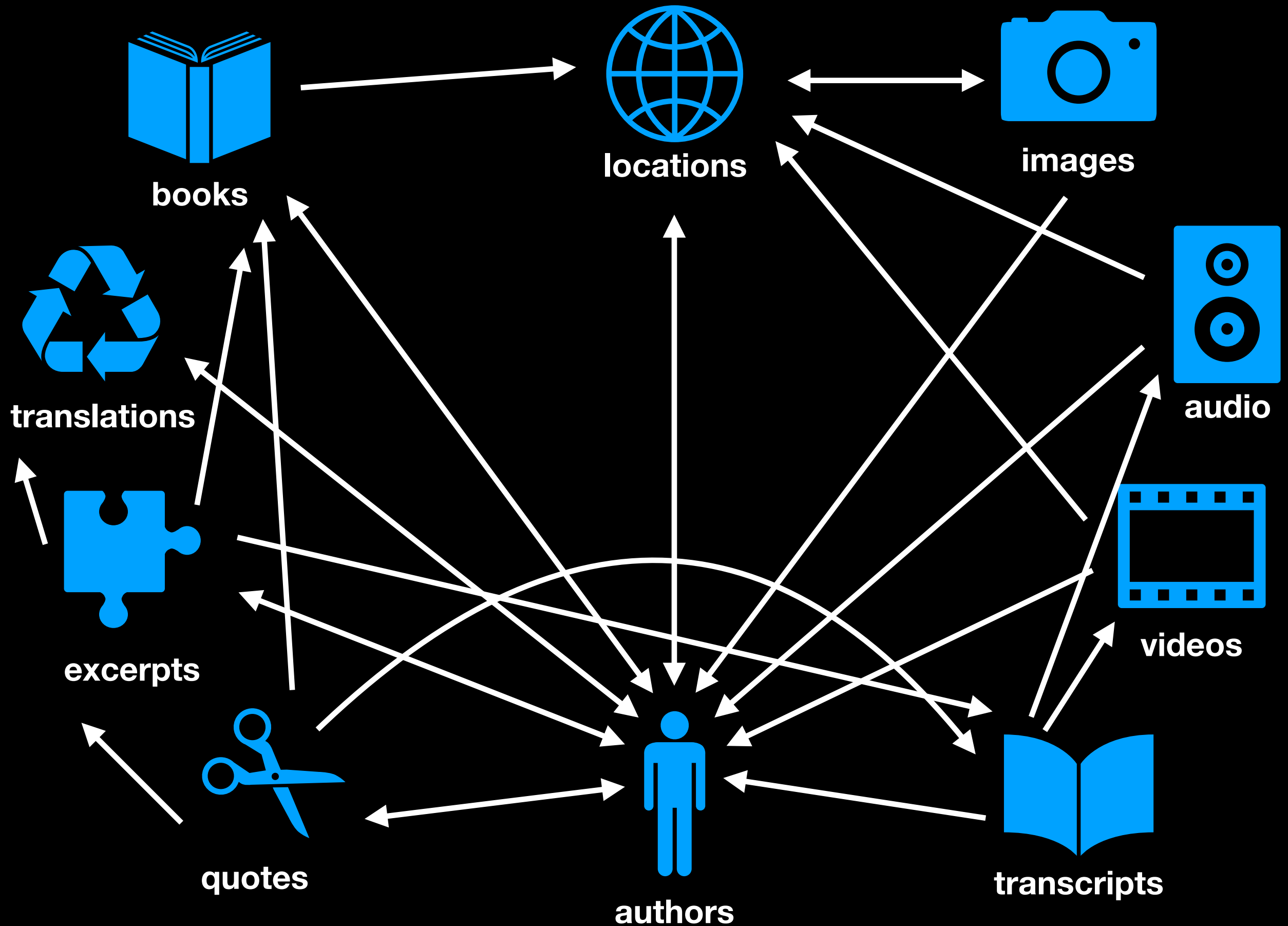


Manuals of Dhamma
English Quote

Unstructured Documents

- Pros:
 - Semantics are easily understood as key-value pairs (ex. author="Ledi Sayadaw" and data-type="video")
 - Because the tools are easy, they are readily available
 - Search engines (Google, in particular) still respect microformats (microformats.org) so unstructured docs can still contain loose semantics
- Cons:
 - Librarians and editors will find it difficult to capture all semantics
 - Capturing even basic semantics will become painfully repetitive very quickly
 - No centralization of data; the system will not understand "Ven. Ledi Sayadaw", "L. Sayadaw", and "Ledi Sayadaw" are the same entity and will have difficulty providing meaningful search results with anything other than heuristics (heuristics are how Google currently works, mind you)
 - No understanding of derivatives or machine-readable (read: searchable) references — therefore no traceability
 - No Pāli semantics
 - Document storage will probably be separate from the documentation engine (CMS), which is likely to cause broken links
- Options:
 - Documents: Drupal or WordPress
 - Search: Customized Elasticsearch or Google Custom Search for non-profits

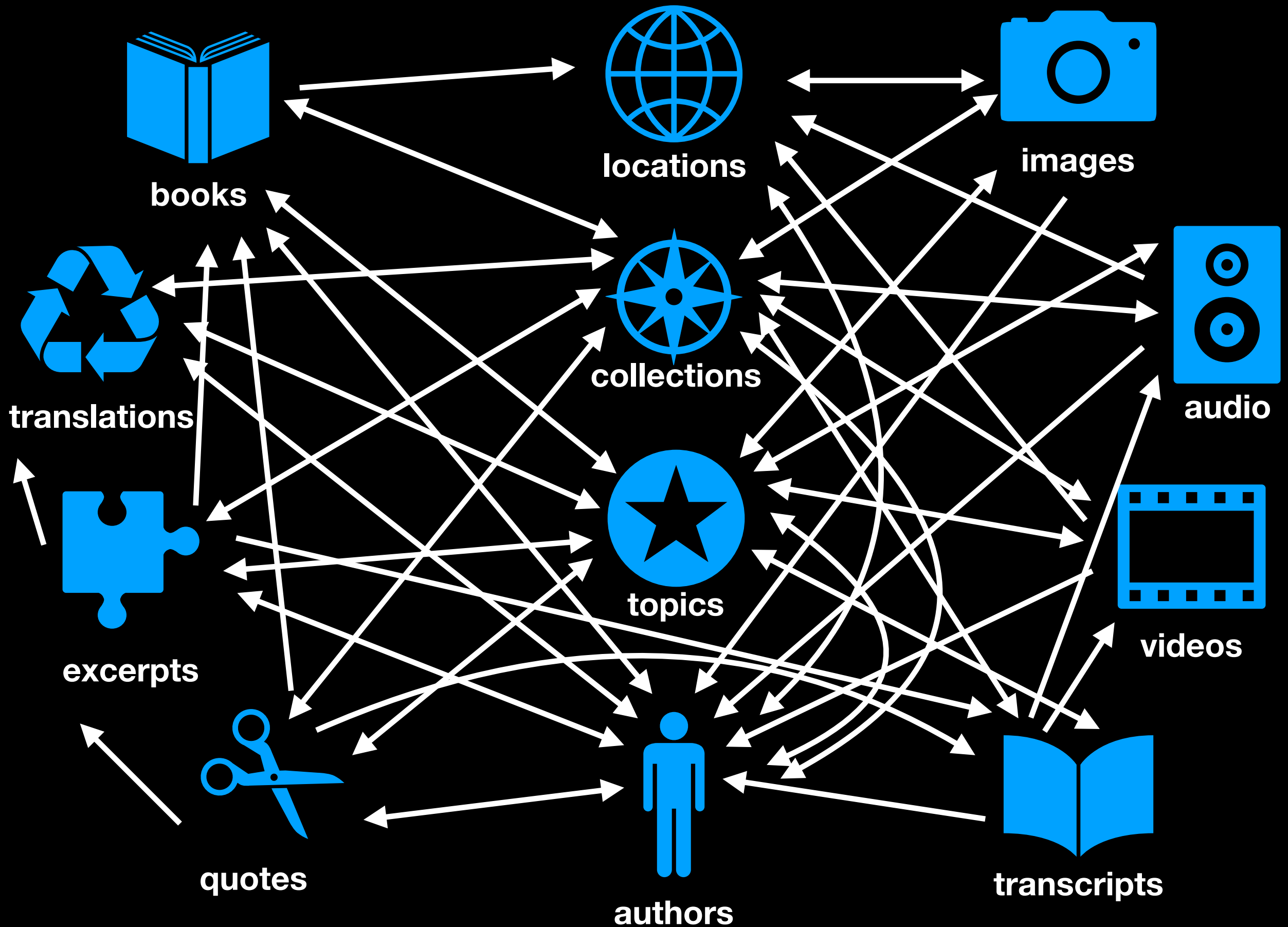
Viewing the world as a *relational database* (as has-a relationships)



Relational Databases

- It's probably safe to say this is already looking like a mess, even in this incomplete form, and the likelihood of more many-to-many relationships is even higher than what is illustrated here.
- Many-to-many relationships are a *Bad Thing*.
- Now let's model abstract concepts...

Viewing the world as a *relational database* (including abstractions)



Relational Databases

- Pros:
 - Easy to deploy
 - Easy for most developers to work with (when the schema is simple)
 - Data is centralized and can work beyond heuristics
 - Derivatives and references can be tracked... albeit in a very complicated way
- Cons:
 - Hopefully everyone agrees this option is completely insane.
 - The schema will **definitely not** be simple. The example on the previous page is not unrealistic. Databases can — and do — get this ugly. This will be a problem for Pariyatti if we attempt to capture these semantics in an RDBMS.
 - Very few true *semantics* are actually captured... an RDBMS is not an ontology.
 - No Pāli semantics
 - Document storage will probably be separate from the documentation engine (CMS), which is likely to cause broken links
- Options:
 - Data: PostgreSQL
 - Search: Built-in, Customized ElasticSearch, or Google Custom Search for non-profits

Viewing the world as a *graph database*



books



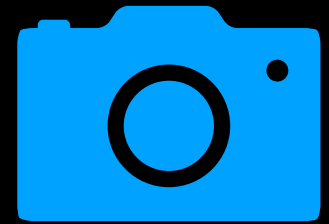
topics



locations



collections

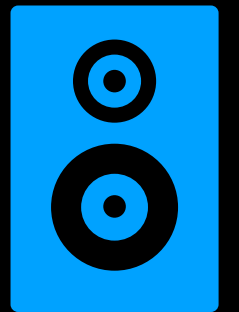


images

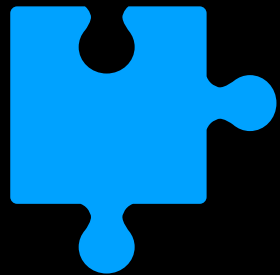
Graphs do not constrain which entities describe which other entities. They also permit an unlimited set of relationships (which we can predefine).



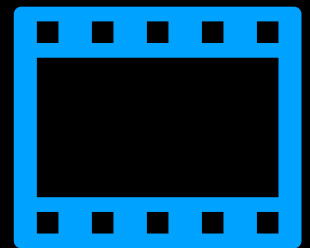
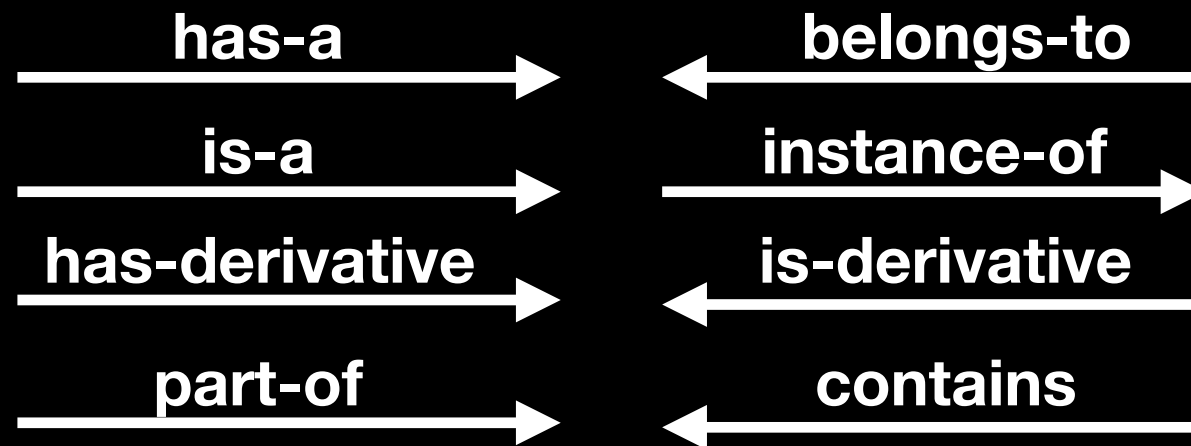
translations



audio



excerpts



videos

etc.



datatype



quotes



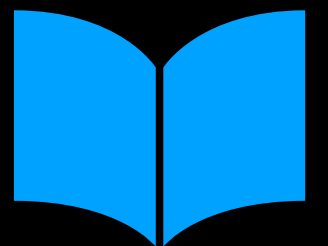
language



authors



time



transcripts

Graph DB Hybrid?

- Pros:
 - Graph DBs are mature and could be almost as easy as an RDBMS for devs
 - Solves the semantics / ontology problem without burdening Pariyatti editors with the full weight of a full-blown semantic data system like WikiBase
 - Derivatives & references are tracked cleanly
 - Pāli semantics can be captured (though we will have to build this)
 - Data is centralized
 - Graph DB and graph model can be hidden from users to avoid confusion
- Cons:
 - DevOps will not be as easy as it would be for an RDBMS
 - Uncertainty surrounding source-of-truth for data/documents (CMS vs. DB); duplication may be best option since they serve different purposes
- Options:
 - Data: Neo4j or dgraph
 - Search: Built-in or ElasticSearch or Google Custom Search