

# Everything is connected

Combating entropy in software

# Entropy in Physics

# All systems move in the direction of increasing entropy.

*According to Boltzmann, systems move in the direction of increasing entropy because such states have a greater number of configurations, and the equilibrium state of highest entropy is the state with the greatest number of molecular configurations.*<sup>Attard</sup>

---

<sup>Attard</sup> Phil Attard, in Thermodynamics and Statistical Mechanics, 2002

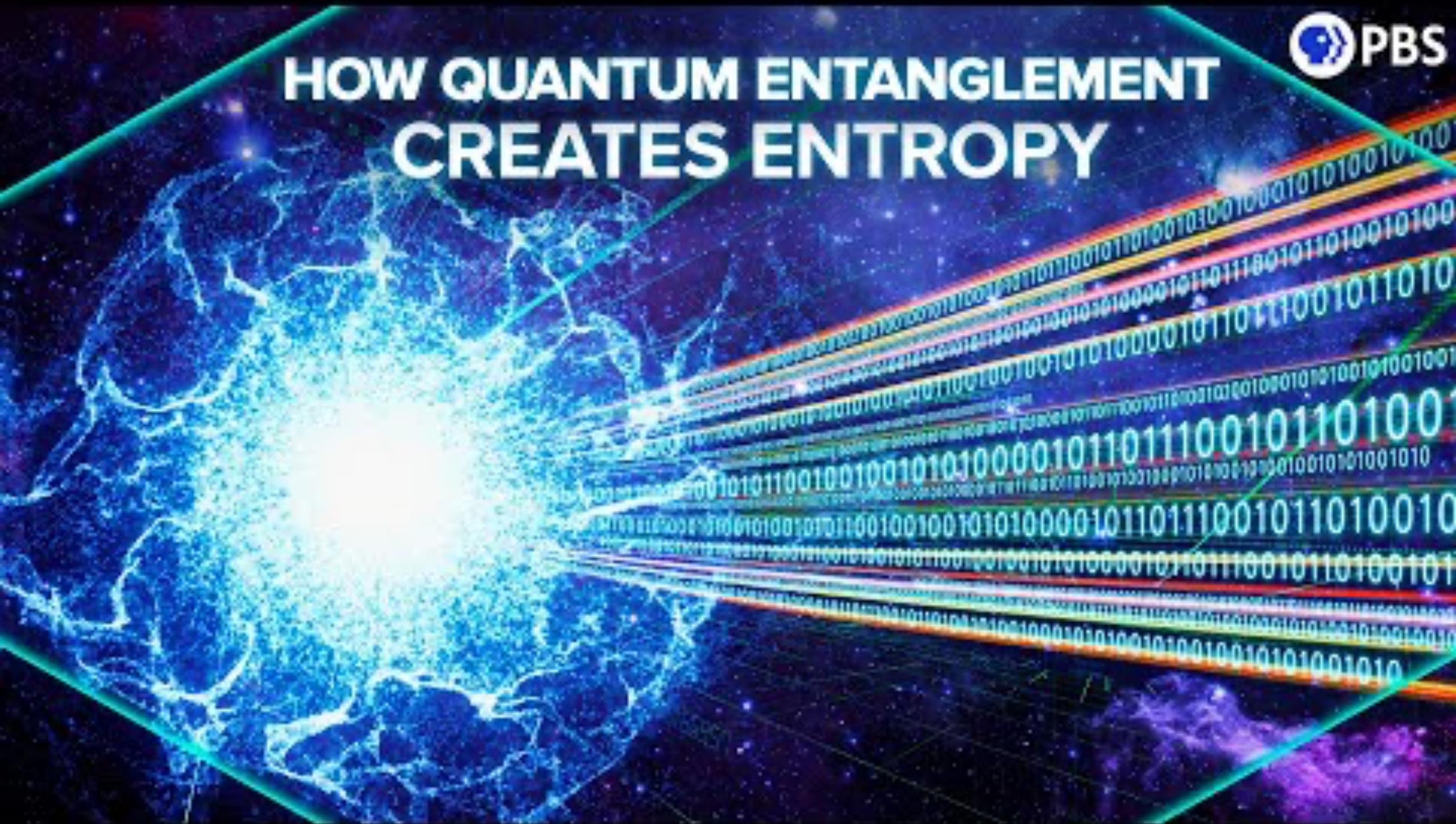
# Quantum wave function

- » Quantum coin toss has both states simultaneously, with 50% probability
- » Once observed, quantum function collapsed into one state
- » Another quantum coin, entangled will automatically have the other state





# HOW QUANTUM ENTANGLEMENT CREATES ENTROPY

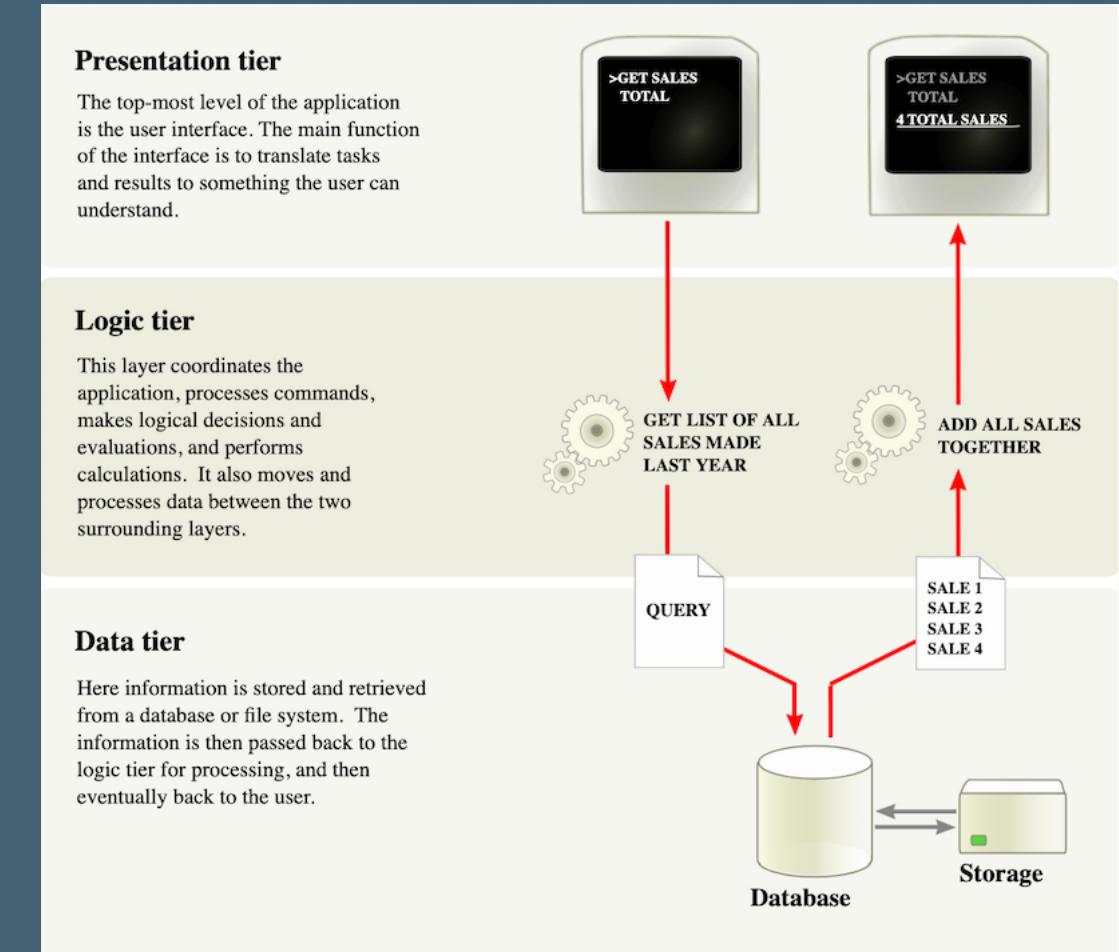


# Software architecture

# Tier architecture

Every tier in a software system has its own view of part of the larger data model.

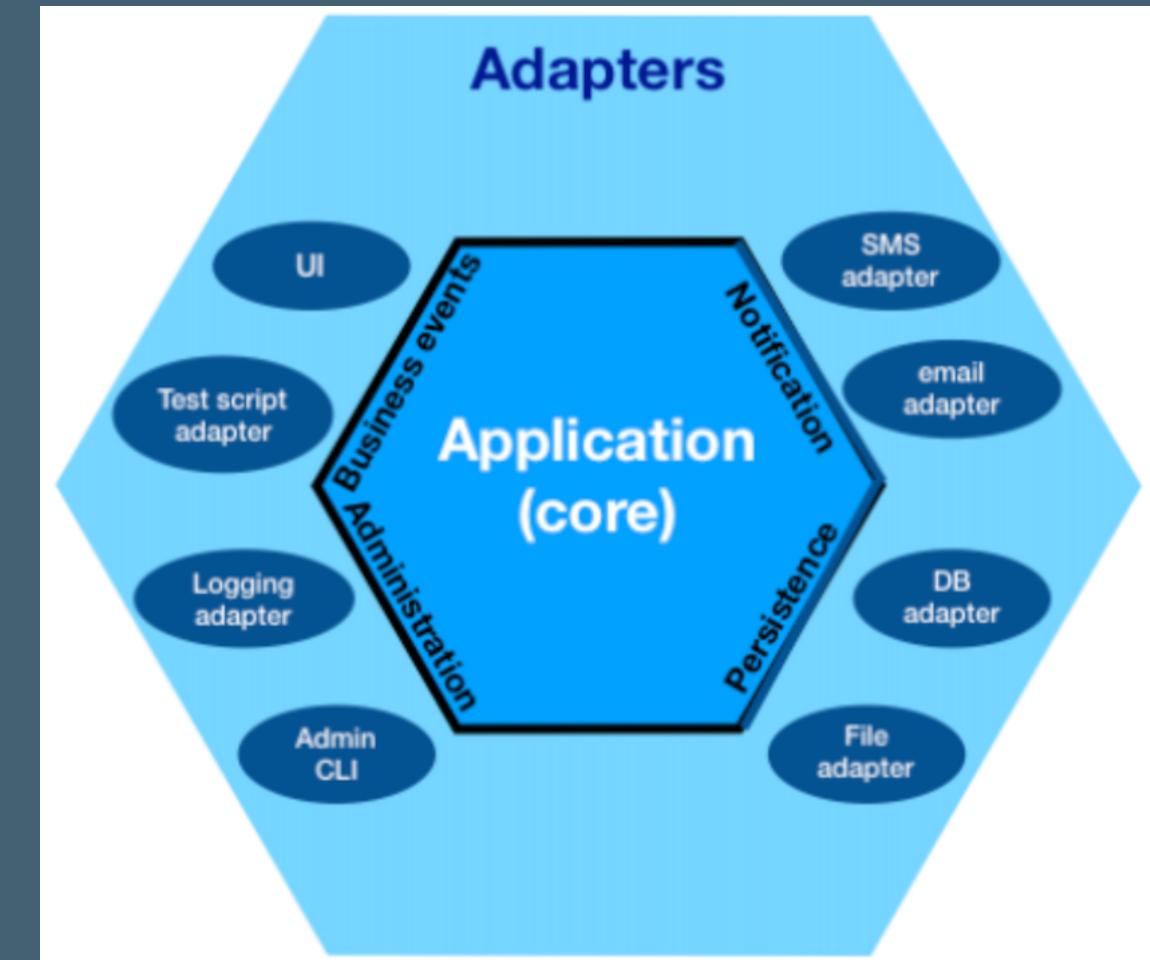
- » database model
- » business layer
- » application layer
- » presentation layer



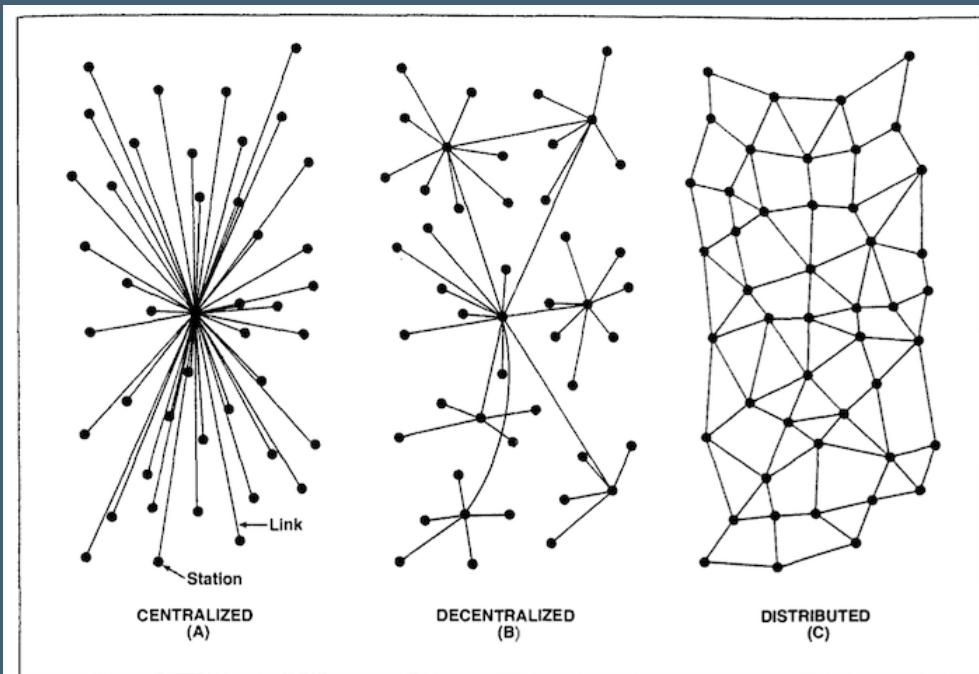
# Hexagonal architecture

The hexagonal architecture divides a system into several loosely-coupled interchangeable components, [...].

Each component is connected to the others through a number of exposed "ports". Communication through these ports follow a given protocol depending on their purpose.



# Paul Baran's Networks Barabasi



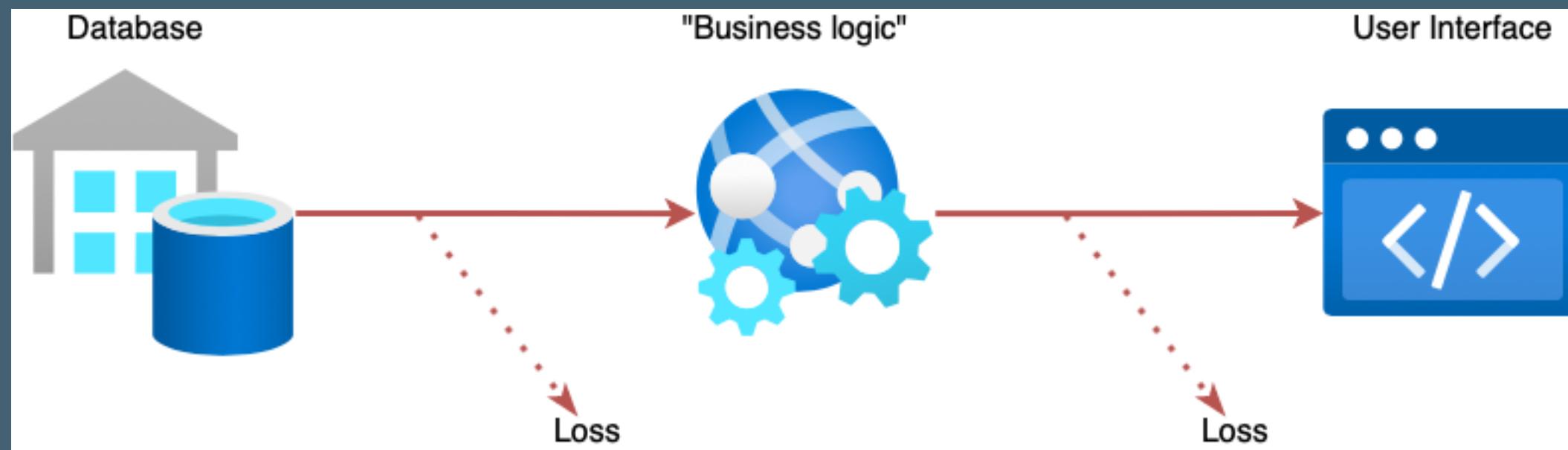
**Figure 11.1** Paul Baran's Networks. In 1964, Paul Baran began thinking about the optimal structure of the Internet. He suggested that there were three possible architectures for such a network—centralized, decentralized, and distributed—and warned that both the centralized and decentralized structures that dominated communications systems of the time were too vulnerable to attack. Instead, he proposed that the Internet should be designed to have a distributed, mesh-like architecture. (Reproduced with permission of Paul Baran.)

---

Barabasi [Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life](#)

# Entropy in software

Every time data changes representation when moved between components, entropy increases.



# When entropy increases

- » Diverging models
- » Technical differences
- » Human-only (API) documentation
- » Generated code
- » Miscommunication

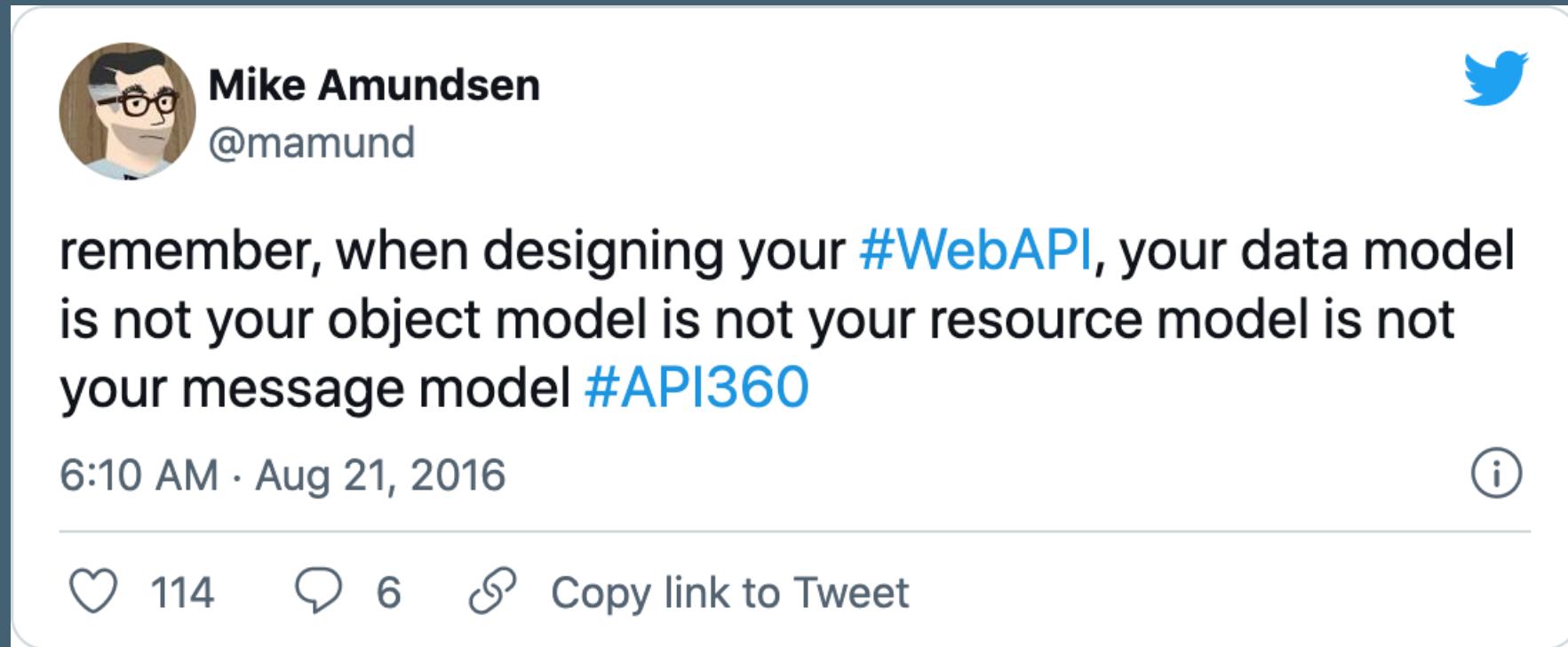


# Designing for lower entropy

# Everything is a resource

if you're brave enough

# Or is it? mamund



A screenshot of a Twitter post from user @mamund. The post features a profile picture of a man with glasses, the name "Mike Amundsen" and handle "@mamund" in bold black text, and a blue Twitter logo. The tweet itself contains a quote in bold black text: "remember, when designing your #WebAPI, your data model is not your object model is not your resource model is not your message model #API360". Below the tweet is the timestamp "6:10 AM · Aug 21, 2016" and an information icon. At the bottom, there are engagement metrics: 114 likes, 6 replies, and a retweet icon followed by the text "Copy link to Tweet".

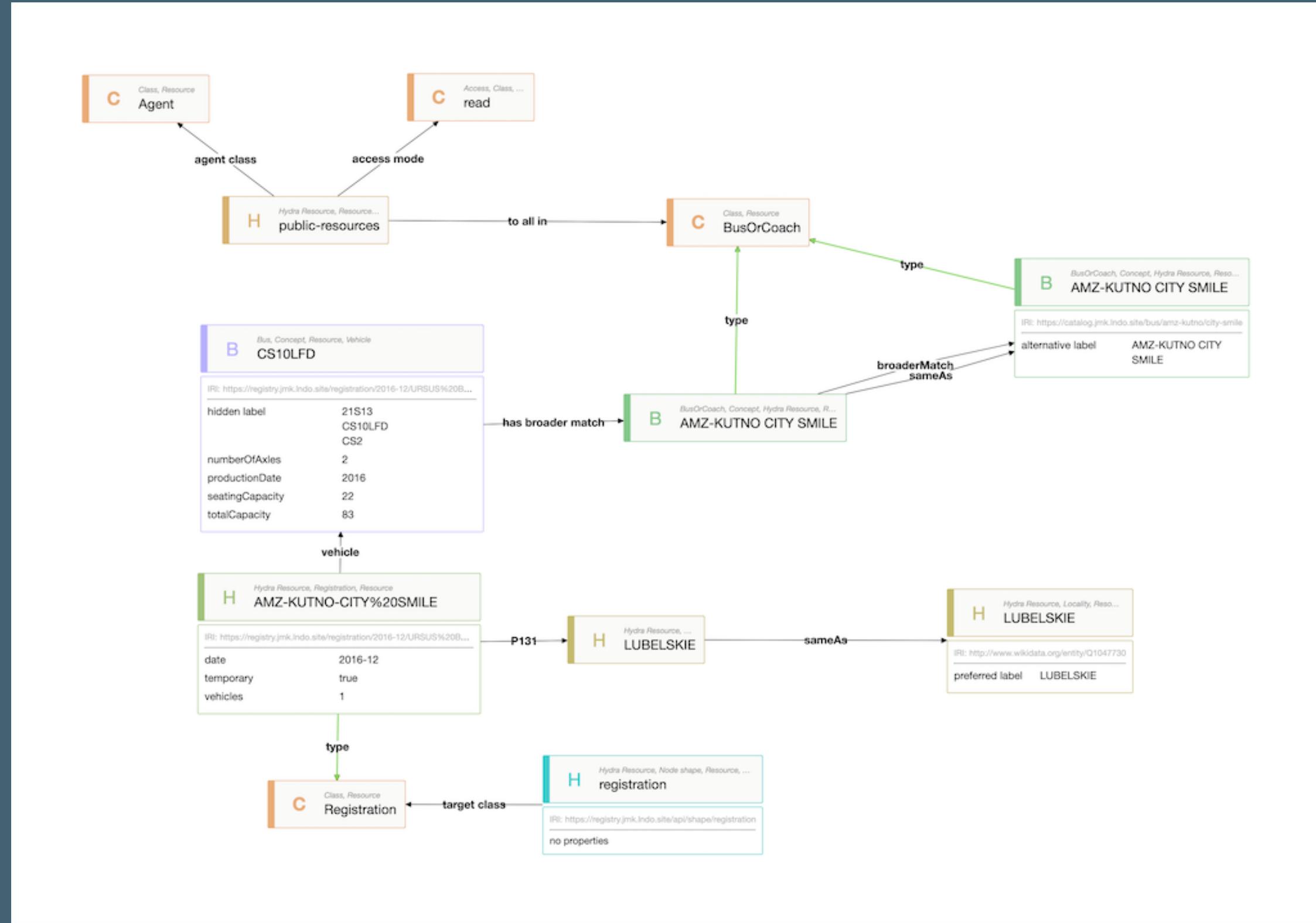
---

mamund <https://twitter.com/mamund/status/767212233759657984>

# Linked Data

- » Everything has an URI (resource)
- » All resources are part of a "global API"
- » Follow (open) web standards; make data useful to others
  - » uniform resource representations (**RDF**, **SHACL**)
  - » shared vocabularies (**schema.org**, domain-specific, others)
  - » interaction (**HTTP/REST**, **Hydra**)





# Demo

Explore data

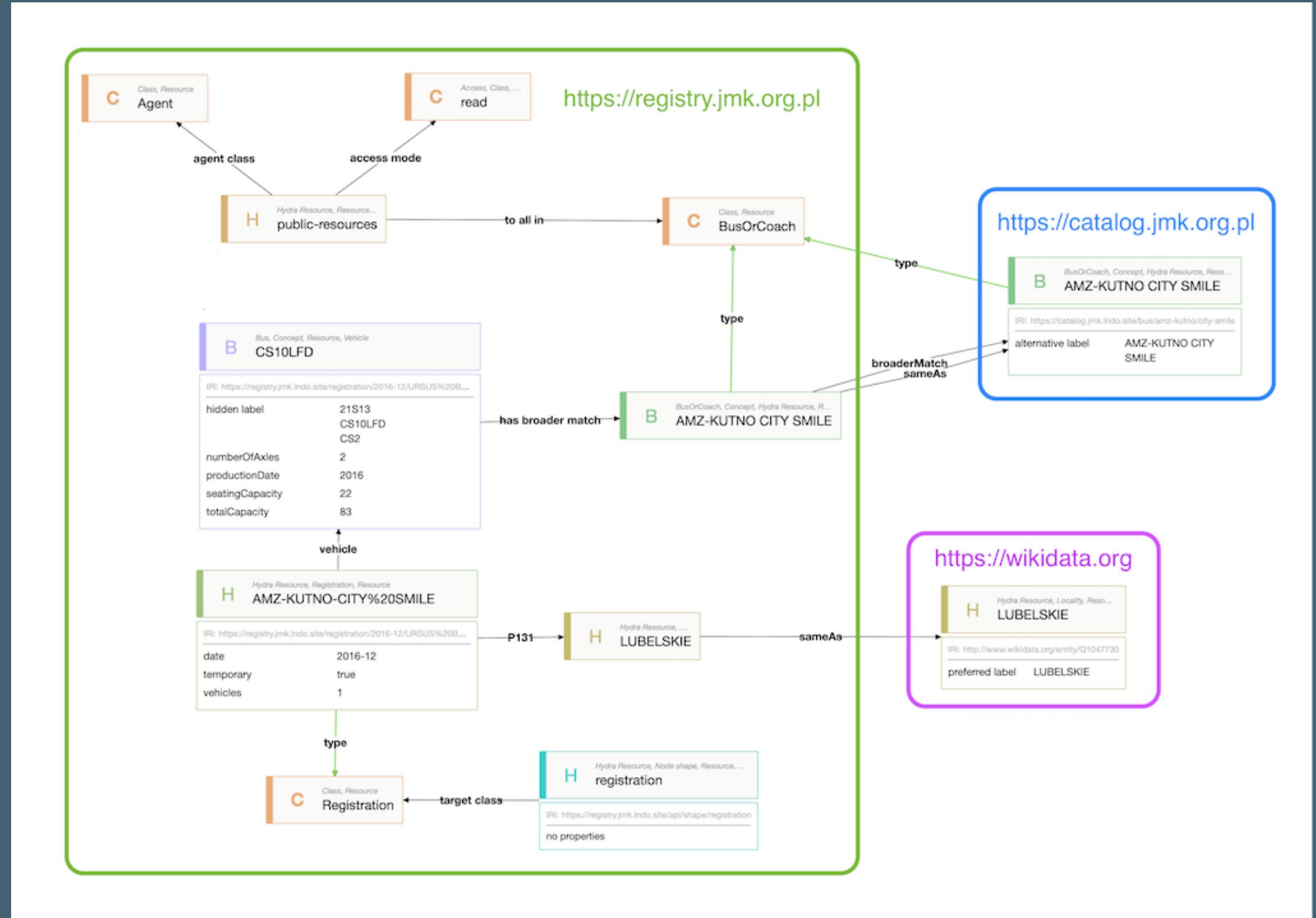
Drive UI

# Example: Crossing boundaries

# Geographic Entities

Identifiers local to each component, troublesome to correlate

Voivodeship	Component 1	Component 2	Component 3
Dolnośląskie	1	7	D
Mazowieckie	2	9	M
Śląskie	3	3	S
Opolskie	4	10	O
...			



# Query across boundaries

Not only merge data from multiple sources, but also connect at runtime Directly to the source system  
This is called a federated query

```
● ● ●

PREFIX jmk: <https://jmk.org.pl/vocab#>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

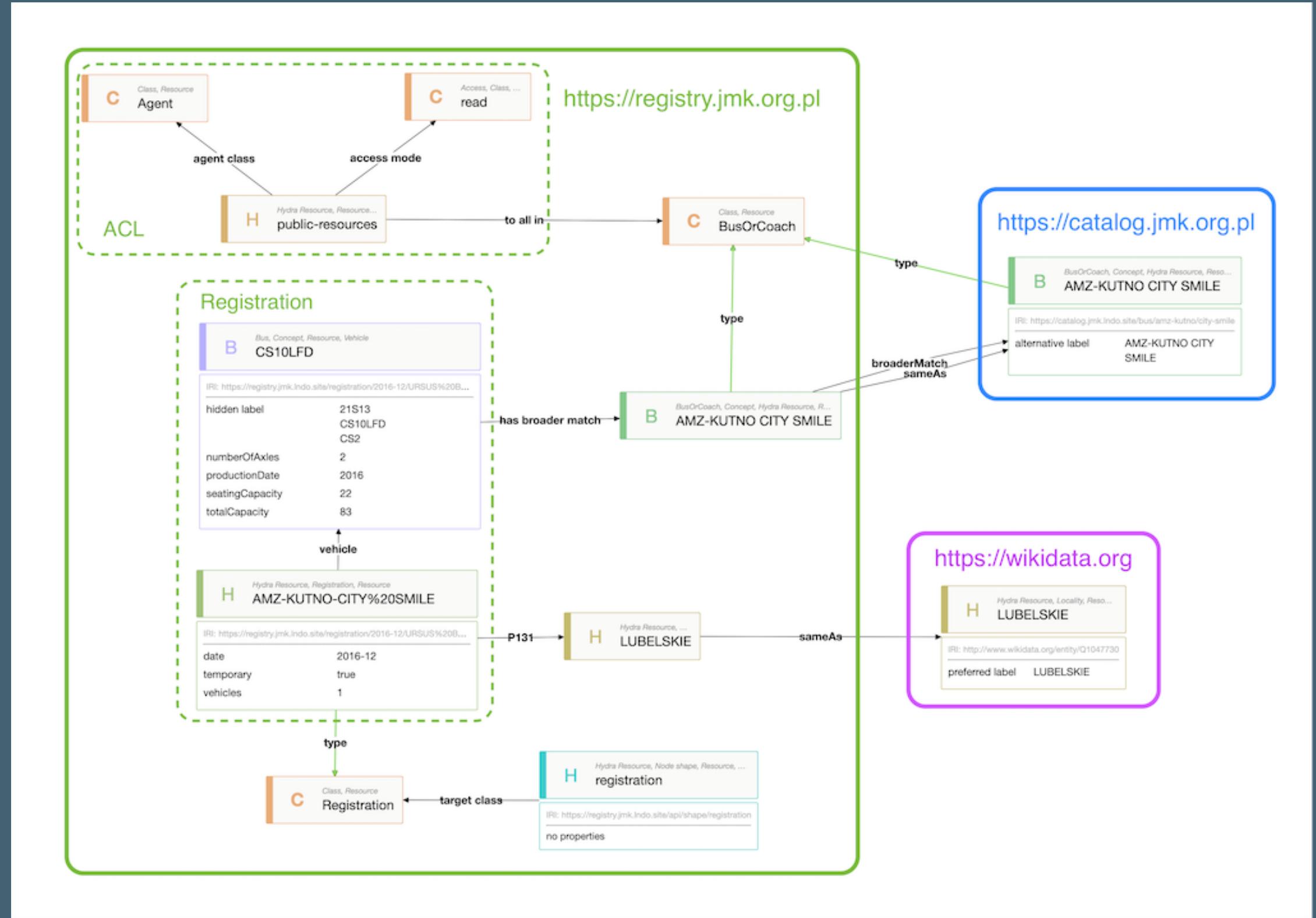
select ?reg
where {
  # select registration and their locations
  ?reg a jmk:Registration ; wd:P131 ?loc .

  # find population on wikidata
  service <https://query.wikidata.org/sparql> {
    # only first administrative divisions
    ?loc wdt:P31/wdt:P279 wd:Q10864048 .

    # find their population to filter
    ?loc wdt:P1082 ?population .
  }

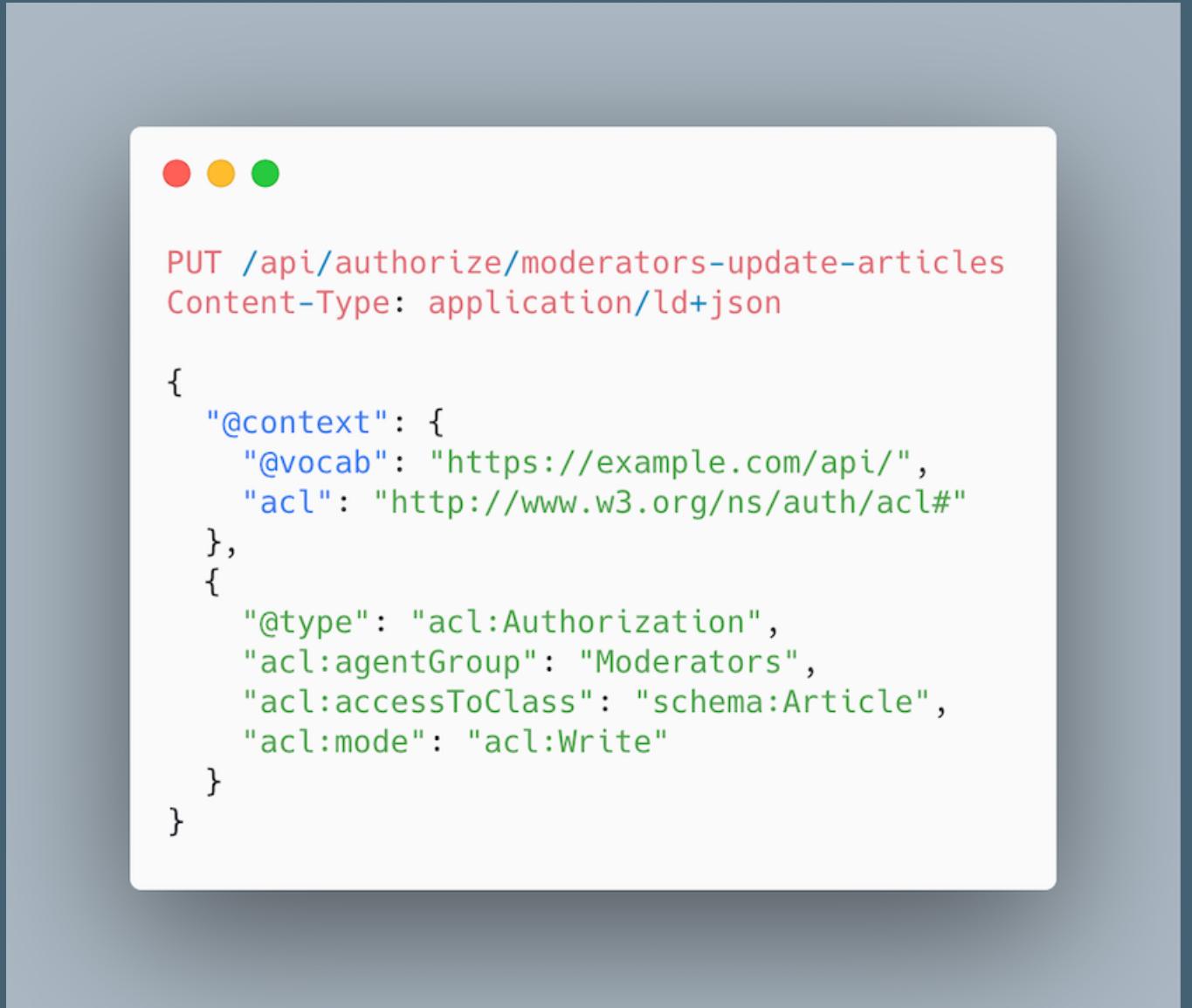
  filter (xsd:int(?population) > 2000000 )
}
```





# Authorization

- » Is also a resource
- » Linked and queried directly
- » Can be part of API
- » Same database or distributed
- » Another standard, by Tim Berners-Lee

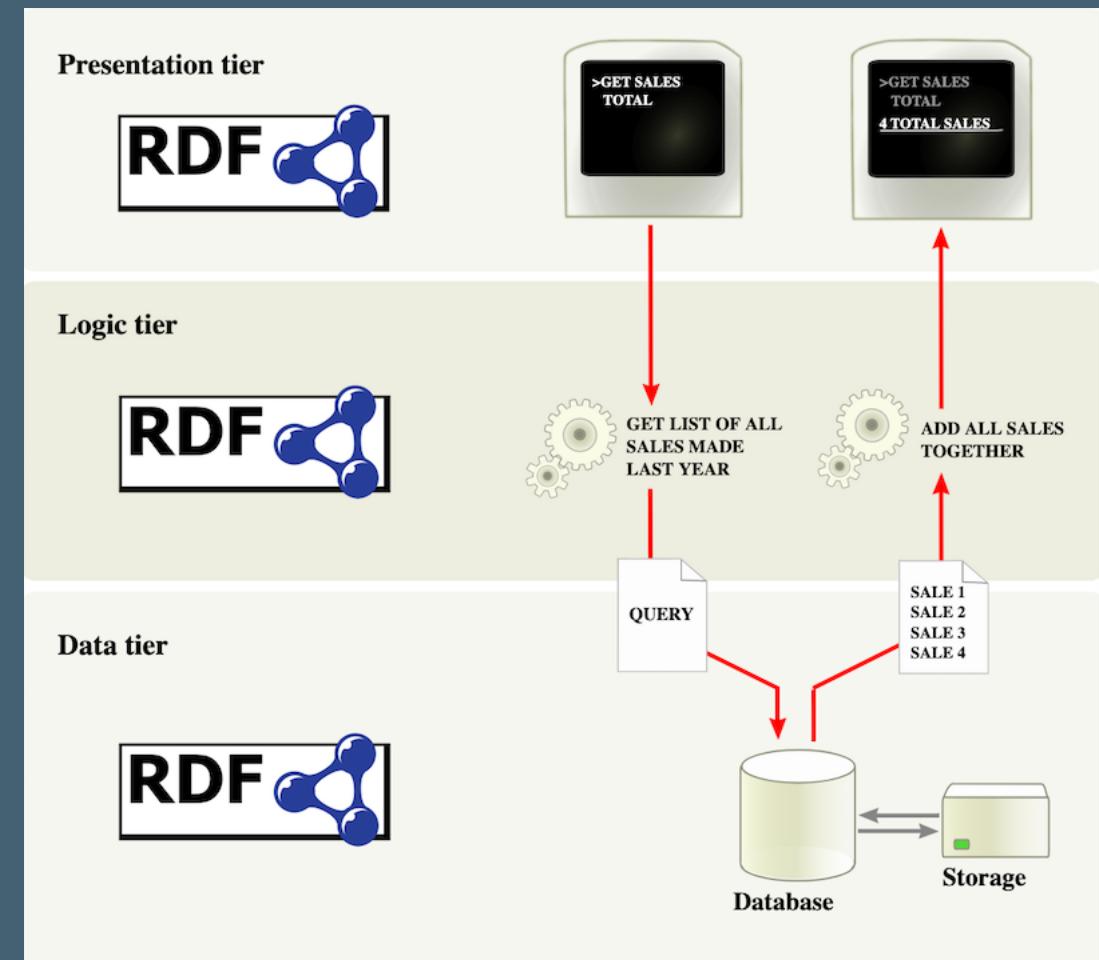


```
PUT /api/authorize/moderators-update-articles
Content-Type: application/ld+json

{
  "@context": {
    "@vocab": "https://example.com/api/",
    "acl": "http://www.w3.org/ns/auth/acl#"
  },
  {
    "@type": "acl:Authorization",
    "acl:agentGroup": "Moderators",
    "acl:accessToClass": "schema:Article",
    "acl:mode": "acl:Write"
  }
}
```

# Summary

- » Model everything as a graph
- » Uniform representation
- » Lossless data integration
- » Distributed by nature



# Bonus - mix & match

## Domain-Driven Design

- » Embed ubiquitous language in universal model
- » Model (sub)graphs as aggregates and entities

## Event driven

- » Model events as resources
  - » [W3C Activity Streams](#)
- » Link to other resources

## CQRS

- » Execute commands locally
  - » individual graph
- » Query globally
  - » across all graphs

# Gotchas

## Learning curve

- » Graph feels alien
- » Basic building blocks for complex solutions
- » Hard to find talent
- » Limited learning resources

## RDF != Neo4j etc.

- » Open vs Closed
- » Queries and analytics

## Not a silver bullet

- » But comes damn close IMO 😊

# The Data-Centric Manifesto

# Data is the center of the universe; applications are ephemeral.

These are the key principles of the data centric manifesto:

1. Data is a key asset of any organization.
2. Data is self-describing and does not rely on an application for interpretation and meaning.
3. Data is expressed in open, non-proprietary formats.
4. Access to and security of the data is a responsibility of the data layer, and not managed by applications.
5. Applications are allowed to visit the data, perform their magic and express the results of their process back into the data layer for all to share.

<http://www.datacentricmanifesto.org>

# Other ways to think about the data centric revolution principles

## NOW: Application-Centric

- » Exorbitant, often prohibitive, cost of change.
- » Data is tied up in applications because applications own data.
- » Every new project comes with a big data conversion project.
- » Data exists in wide variety of heterogeneous formats, structures, meaning, and terminology.
- » Data integration consumes 35%-65% of IT budget.
- » Hard or impossible to integrate external data with internal data.

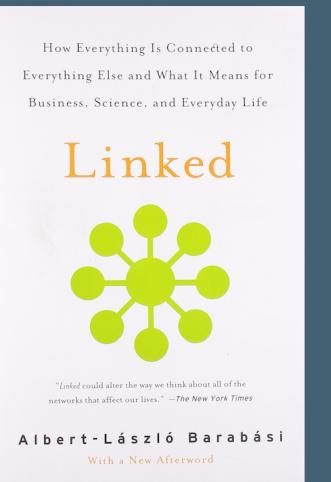
## FUTURE: Data-Centric

- » Reasonable cost of change.
- » Data is an open resource that outlives any given application.
- » Every new project taps into existing data stores.
- » Data is globally integrated sharing a common meaning, being exported from a common source into any needed format.
- » Data integration will be nearly free.
- » Internal and external data readily integrated.

---

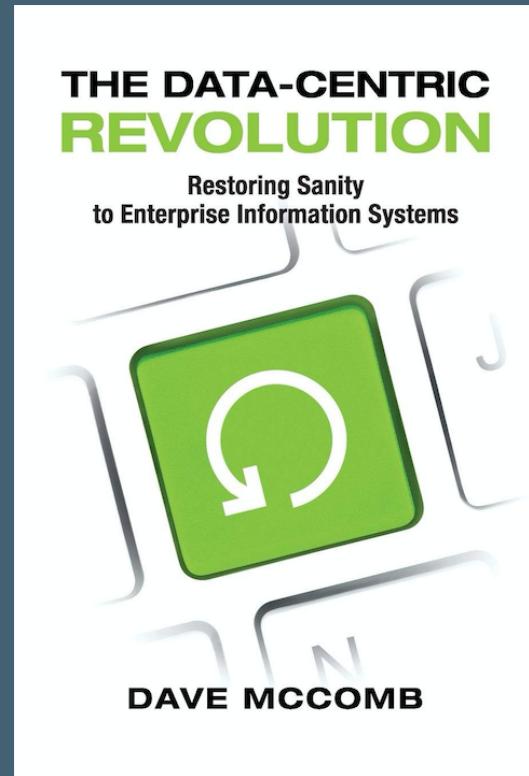
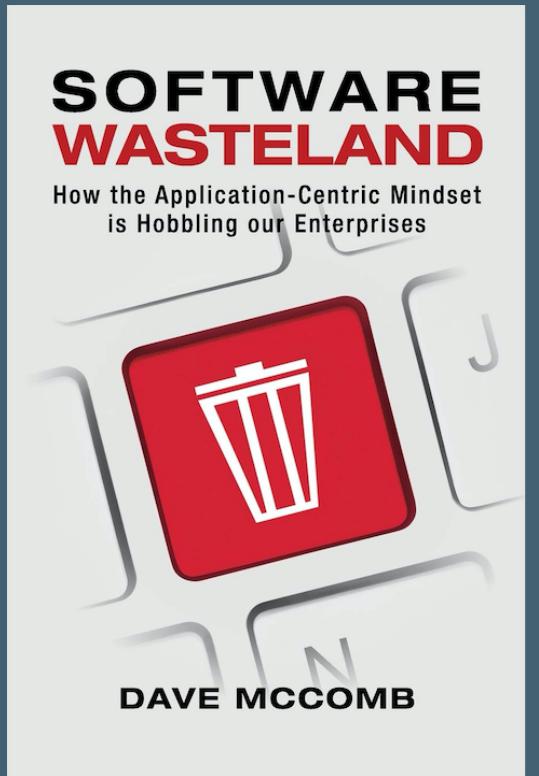
principles <http://www.datacentricmanifesto.org/principles/>

*Where do we go from here?  
The answer is simple. We must  
remove the wrapping. The goal  
before us is to understand  
complexity. To achieve that we  
must move beyond structure  
and topology and start  
focusing on the dynamics that  
take place along the links.* Barabasi



---

Barabasi [Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life](#)



# Thank you for listening

Tomasz Pluskiewicz



 [tpluscode](https://twitter.com/tpluscode)

 [tpluscode](https://github.com/tpluscode)

 <https://t-code.pl>

