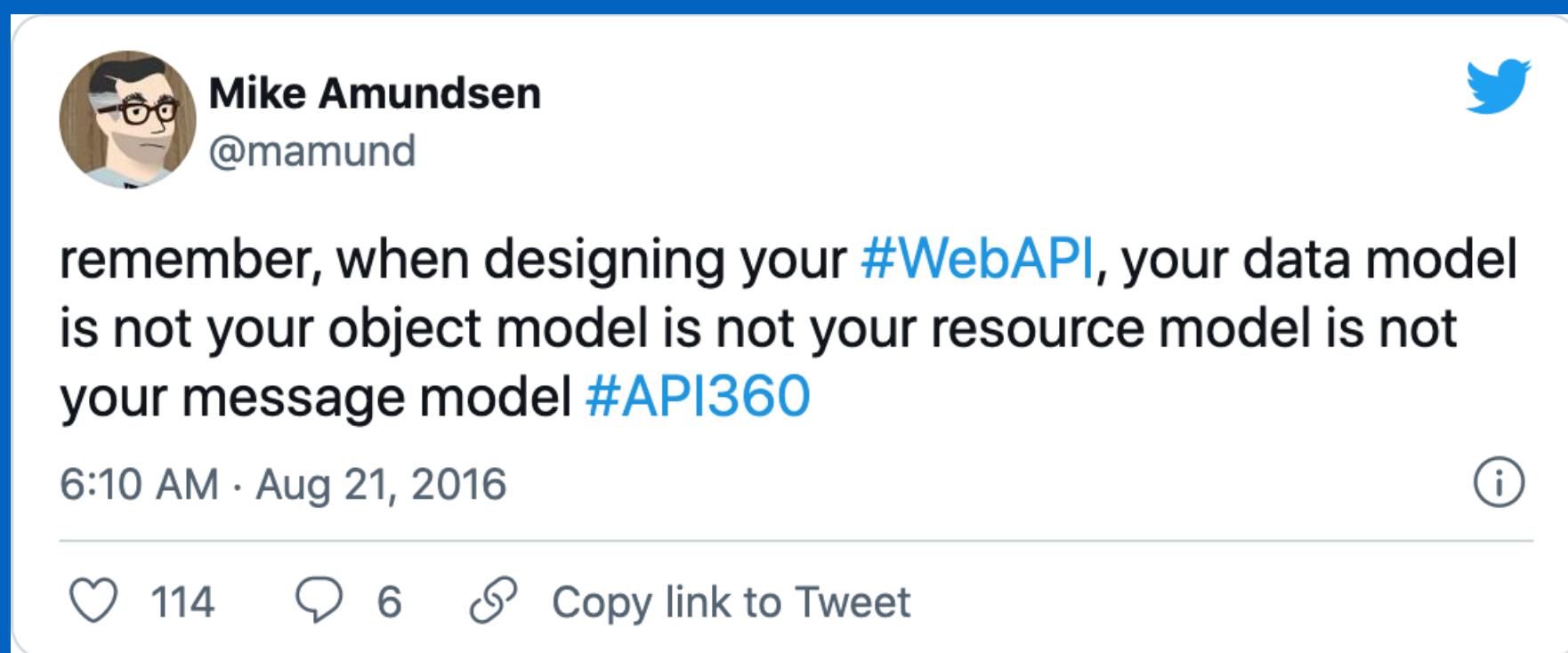


Your API is Your API is Your API

Or is it? Amundsen



Mike Amundsen
@mamund

remember, when designing your #WebAPI, your data model is not your object model is not your resource model is not your message model #API360

6:10 AM · Aug 21, 2016

114 6 Copy link to Tweet

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

Tomasz Pluskiewicz ::  tpluscode :: Nordic APIs LiveCast - Evolving Hypermedia :: 2021

2/23

This still works on ISO/OSI level of HTTP, TCP, IP
we need to be able to design/
implement each aspect (data,
object, resource, representation)
independently without breaking the
others.

First, history

Where APIs come from?

Paul Baran's Networks Barabasi

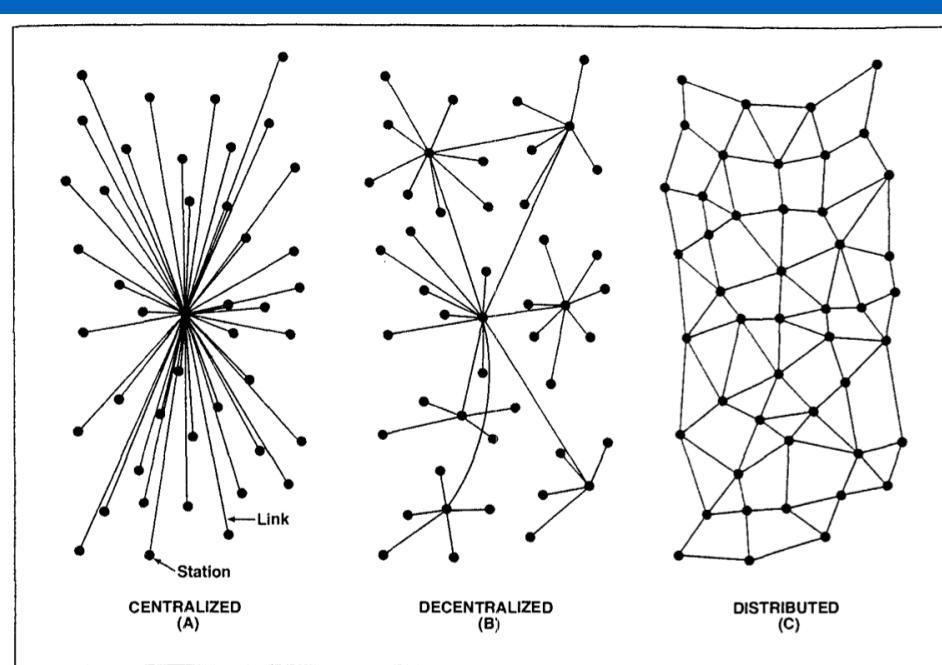


Figure 11.1 Paul Baran's Networks. In 1964, Paul Baran began thinking about the optimal structure of the Internet. He suggested that there were three possible architectures for such a network—centralized, decentralized, and distributed—and warned that both the centralized and decentralized structures 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](#)

Fielding's dissertation is not about how to build APIs on top of HTTP but rather about HTTP itself.

– *Sinclair Target* ^{Target}

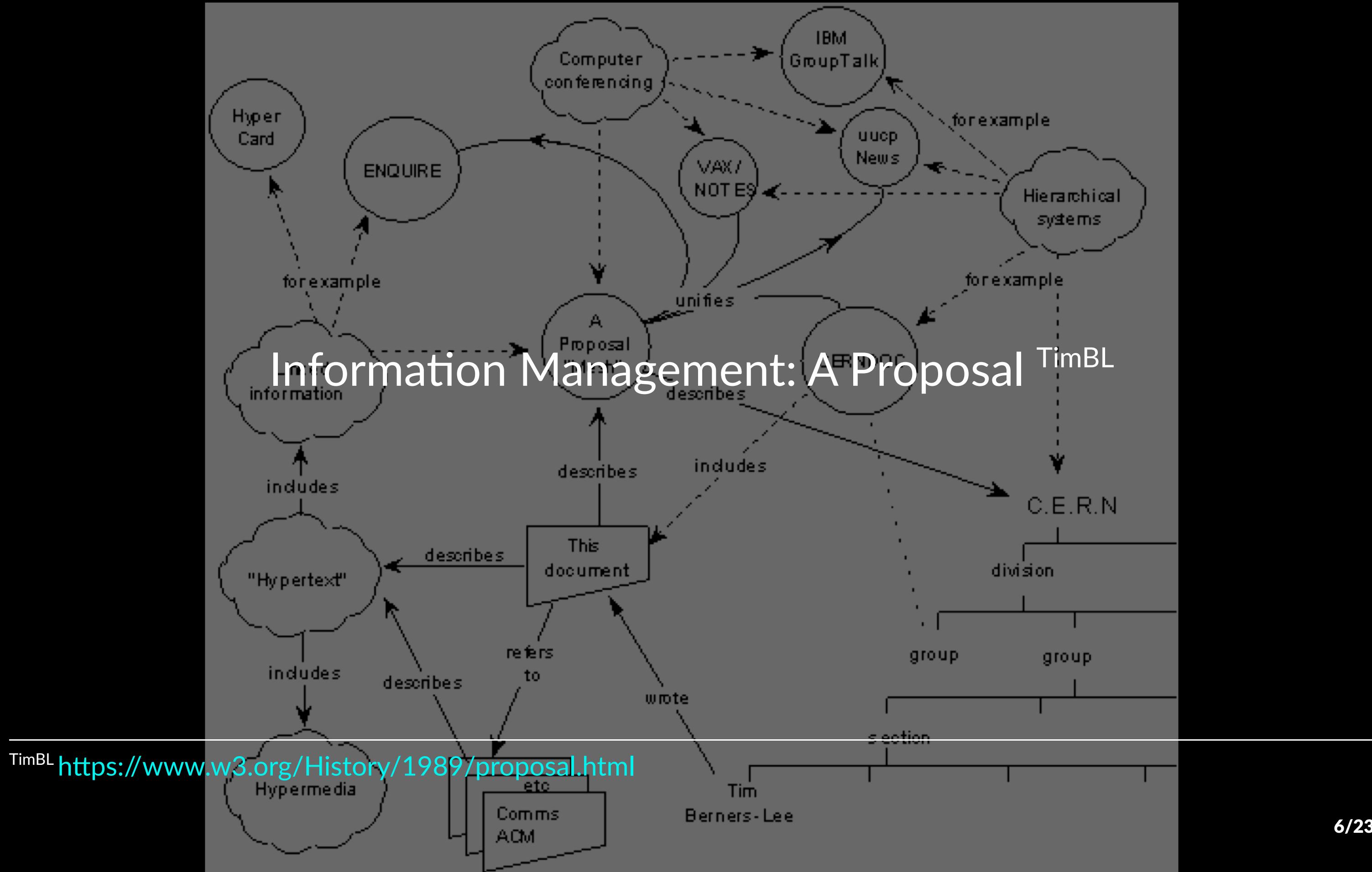
Fielding [Architectural Styles and the Design of Network-based Software Architectures](#)

Target [Roy Fielding's Misappropriated REST Dissertation](#)

Tomasz Pluskiewicz ::  [tpluscode](#) :: [Nordic APIs LiveCast - Evolving Hypermedia](#) :: 2021

5/23

The name “Representational State Transfer” is intended to evoke an image of how a well-designed Web application behaves: a network of web pages, where the user progresses through the application by selecting links, resulting in the next page being transferred to the user and rendered for their use.



Those are real thing, linked together

Everything is a resource

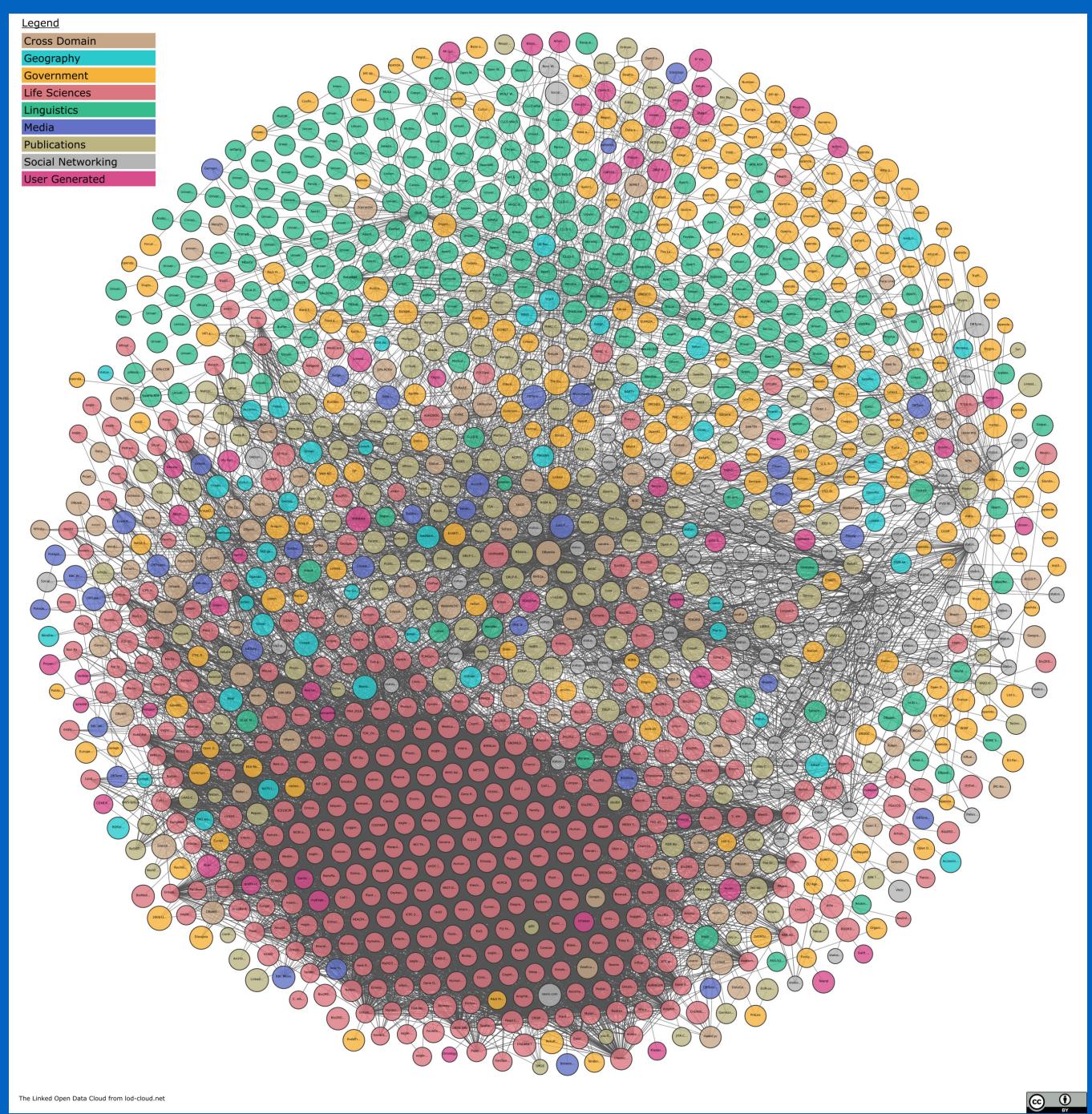
if you're brave enough

Linked Data

Linked Open Data Cloud (lod-cloud.net)

As of May 2020 the dataset contained
1301 datasets with **16283** links.

Grouped in **9** categories.





TimBL-LD <https://www.w3.org/DesignIssues/LinkedData.html>

Tomasz Pluskiewicz ::  tpluscode :: [Nordic APIs LiveCast - Evolving Hypermedia](#) :: 2021

9/23

So what are APIs?

graphs of web resources
connected with unambiguous
links!

Examples



```
{  
  "@context": {  
    "bus": "https://bus.name/vocab#",  
    "dbo": "http://dbpedia.org/ontology/",  
    "schema": "http://schema.org/",  
    "skos": "http://www.w3.org/2004/02/skos/core#"  
  },  
  "@id": "/registration/2016/06/mza-warszawa",  
  "@type": "/api/Registration",  
  "bus:owner": {  
    "@id": "/company/mza-warszawa"  
  },  
  "bus:vehicle": {  
    "skos:prefLabel": "Solaris Urbino 12 CNG",  
    "schema:numberOfAxles": 2  
  },  
  "dbo:number": 4,  
  "skos:prefLabel": "Sample registration"  
}
```

A bus registration ^{reg}

- identified by an URI (**@id**)
- links to its type
(api/Registration)
- links to other resources
(company/mza-warszawa)
- uses unambiguous property names
(https://bus.name/vocab#vehicle)
- uses shared vocabulary terms
(prefLabel, numberOfAxles)

^{reg} <https://tinyurl.com/ahc9b2c>

mention it is JSON-LD

```
[{  
  "@context": {  
    "skos": "http://www.w3.org/2004/02/skos/core#",  
    "schema": "http://schema.org/",  
    "wikidata": "http://www.wikidata.org/entity/"  
  },  
  "@id": "/company/mza-warszawa",  
  "@type": "/api/Owner",  
  "skos:prefLabel": "Miejskie Zakłady Autobusowe",  
  "schema:location": { "@id": "wikidata:Q270" }  
}, {  
  "@context": {  
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",  
    "schema": "http://schema.org/"  
  },  
  "@id": "/api/Owner",  
  "@type": "rdfs:Class",  
  "rdfs:subClassOf": { "@id": "schema:Organization" }  
}]
```

Owner class class

- identified by URI
- uses shared vocabulary terms
**(rdfs:Class,
 schema:Organization)**
- class hierarchy expressed as links
(rdfs:subClassOf)

class <https://tinyurl.com/3zrcv2sr>

those are two separate
resources

```
{  
  "@context": {  
    "dbo": "http://dbpedia.org/ontology/",  
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",  
    "@vocab": "http://www.w3.org/ns/shacl#",  
    "xsd": "http://www.w3.org/2001/XMLSchema#"  
  },  
  "@id": "/api/Registration",  
  "@type": ["rdfs:Class", "NodeShape"],  
  "property": [{  
    "path": "skos:prefLabel",  
    "maxCount": 1  
  }, {  
    "path": "dbo:number",  
    "minCount": 1,  
    "maxCount": 1,  
    "datatype": "xsd:integer"  
  }]  
}
```

Shape of Registration resources shacl

- SHACL Shapes Constraint Language
 - W3C Vocabulary
 - <https://w3.org/tr/shacl>
 - Described graph data structures
 - Validation rules
 - Many other use cases
 - Easily extensible

shacl <https://tinyurl.com/v7vkeuj8>

SHApes Constraint Language

```
{  
  "@context": {  
    "@vocab": "http://www.w3.org/ns/hydra/core#",  
    "schema": "http://schema.org/",  
    "code": "https://code.described.at/"  
  },  
  "@id": "/api",  
  "@type": "ApiDocumentation",  
  "supportedClass": {  
    "@id": "/api/Registration",  
    "supportedOperation": {  
      "@id": "/api/Registration/Put",  
      "@type": "schema:UpdateAction",  
      "method": "PUT",  
      "expects": { "@id": "/api/Registration" },  
      "code:implementedBy": {  
        "@type": "code:EcmaScript",  
        "code:link": { "@id": "node:lib/registration/update.js" }  
      }  
    }  
  }  
}
```

Hydra W3C Community Group hydra

- describes APIs with RDF
 - like OpenAPI but discovered at runtime
(HTTP Link header)
- provides possible state transitions
(among other things)

hydra <https://tinyurl.com/24tku578>



```
{  
  "@context": {  
    "@vocab": "http://www.w3.org/ns/auth/acl#"  
  },  
  "@id": "/api/auth/write-registrations",  
  "@type": "Authorization",  
  "agentGroup": {  
    "@id": "/group/data-owners"  
  },  
  "mode": { "@id": "Write" },  
  "accessToClass": {  
    "@id": "/api/Registration"  
  }  
}
```

Basic Access Control ontology acl

- Another shared vocabulary
- Started by Tim Berners-Lee
- Now part of **SOLID Project**

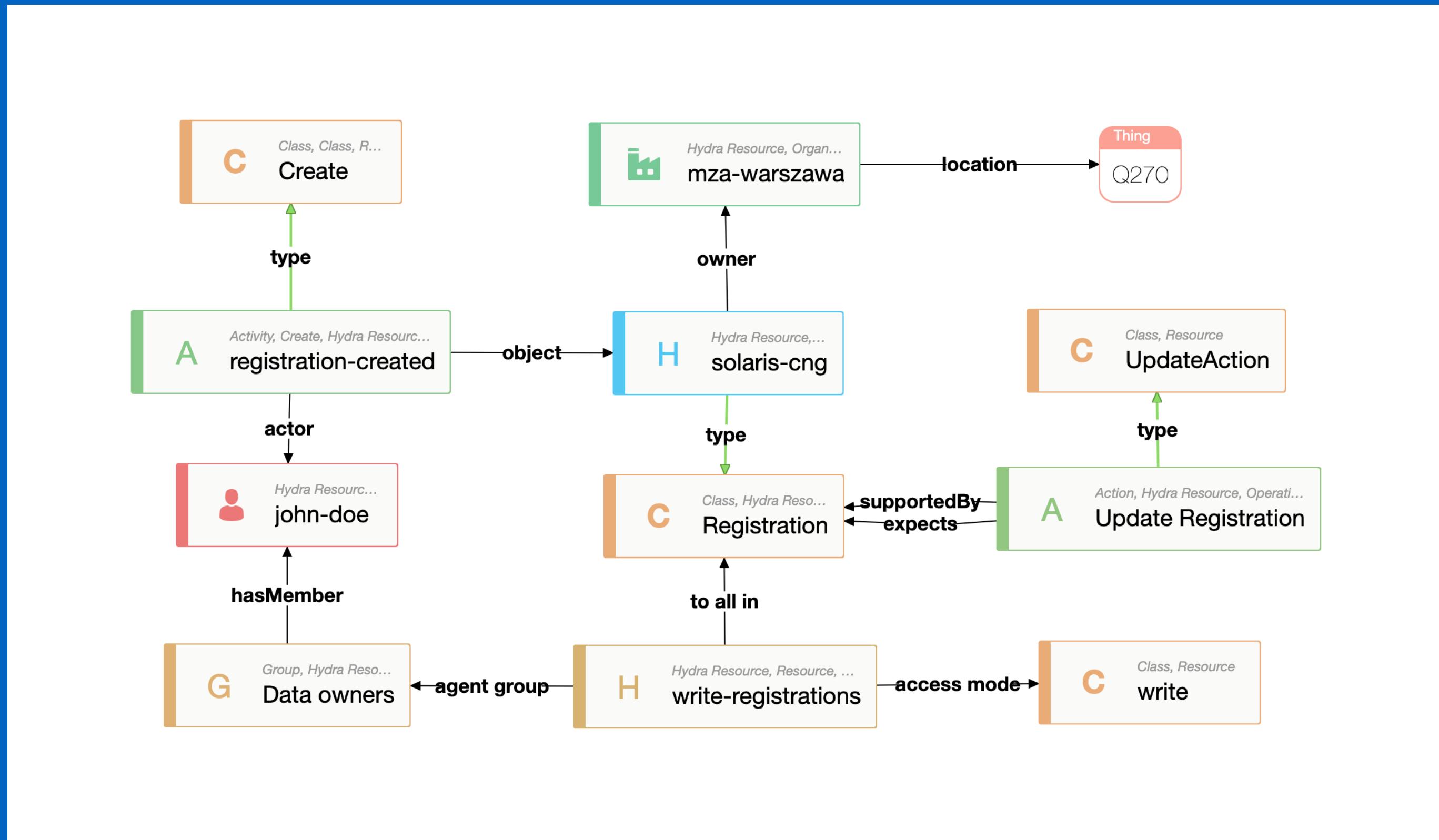
acl <https://tinyurl.com/2ekfmhrb>

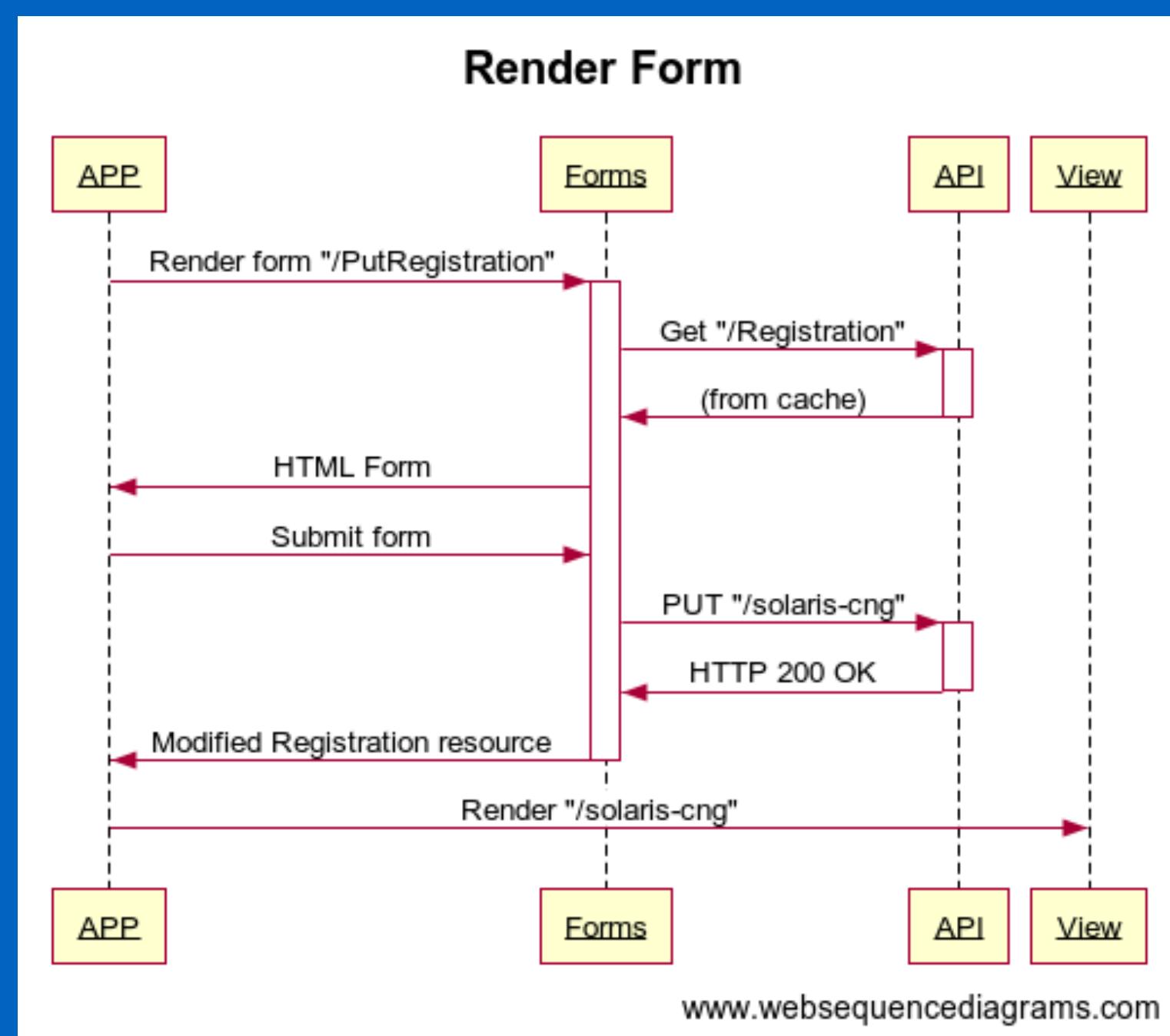
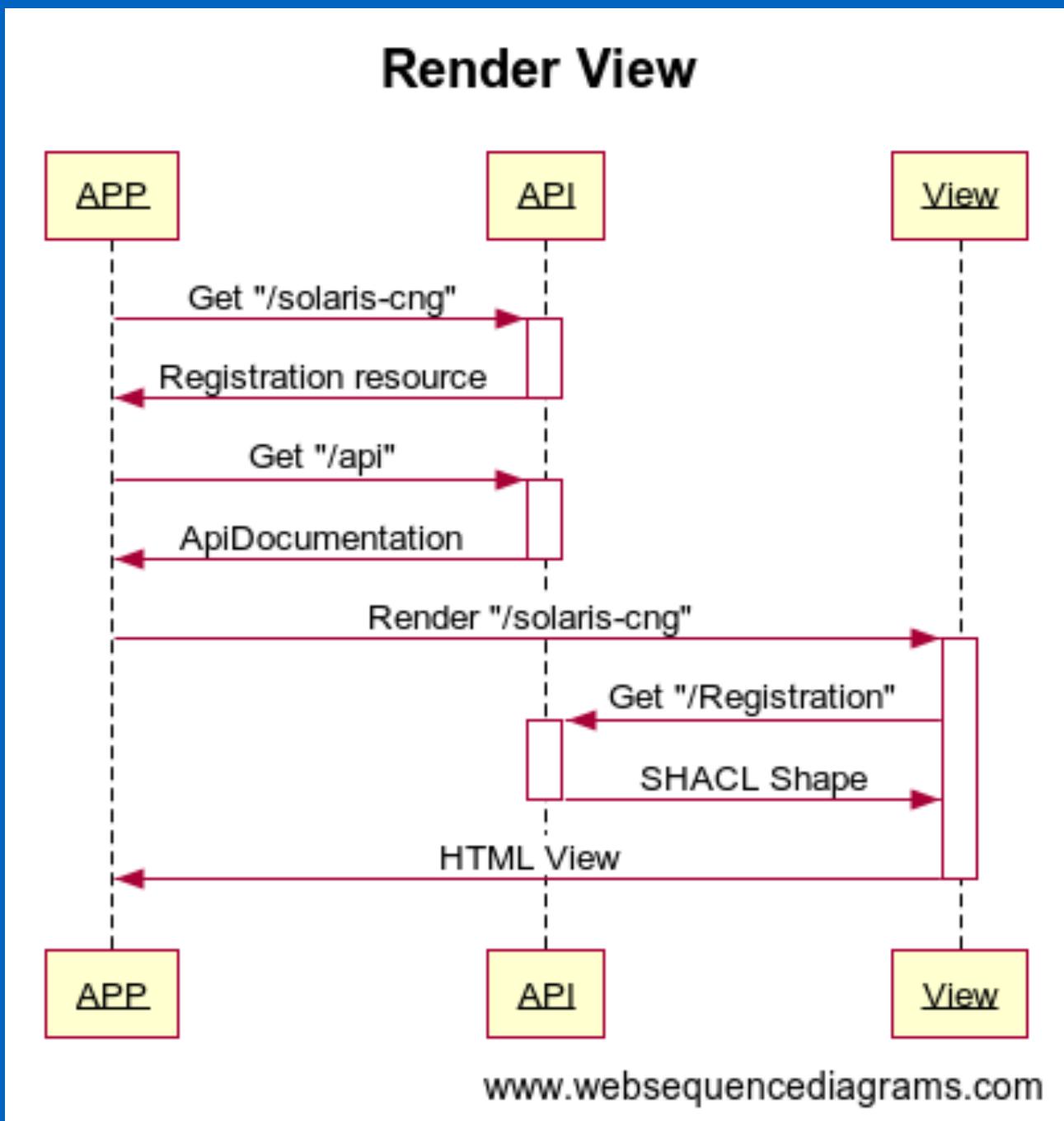
group is also a resource, and users are resources...

SaaS is dead. Long live...

Software as a Resource?

Resource as a Service?







Why?

- Discoverability
- Evolvability
- Longevity

Data is self-describing and does not rely on an application for interpretation and meaning.

– *Data-Centric Manifesto*^{manifesto}

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

Summary

1. Everything is a resource
2. Link to other resources
3. Use shared vocabularies
4. Do not hard code resource semantics in client code

Resources (no pun intended)

This presentation: tpluscode@github/presentations/nordic-apis-live-hypermedia

- [Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life](#)
- [What does the history of the web tell us about its future?](#)
- [Roy Fielding's Misappropriated REST Dissertation](#)
- [Architectural Styles and the Design of Network-based Software Architectures](#)
- [Information Management: A proposal](#)
- [Linked Open Data Cloud](#)
- [Linked Data Stars](#)
- [Data-Centric Manifesto](#)
- [Resource Description Framework](#)
- [SKOS Simple Knowledge Organization System](#)
- [schema.org](#)
- [SHACL Shapes Constraint Language](#)
- [Wikidata](#)
- [Hydra W3C Community Group](#)
- [Basic Access Control ontology](#)
- [SOLID Project](#)
- [Activity Streams](#)