



THE WEB CONFERENCE

Graph Data: RDF, Property Graphs (Results of a Workshop...)

W3C Track, The Web Conference 2019

May 15, 2019

San Francisco, CA, USA

Ivan Herman, W3C/CWI




These slides are on the Web:

- <https://www.w3.org/2019/Talks/W3C-track-IH/Presentation.pdf>



The facts

- W3C Workshop on “Web Standardization for Graph Data”:
- Berlin, 4-6 March 2019
- ≈100 participants
- one keynote (from Amazon), ≈20 full presentations, and a series of short presentations
- lots of discussions, panels
- program, submissions, etc, are available via: <https://www.w3.org/Data/events/data-ws-2019/>



W3C Workshop on Web Standardization for Graph Data

Creating Bridges: RDF, Property Graph and SQL

Monday 4th March to Wednesday 6th March 2019, Berlin, Germany ([venue](#))

Home Page
Call for Participation
Workshop Agenda
Minutes
Position Statements
Report

Introduction

Data is increasingly important for all organisations, especially with the rise of IoT and Big Data. The falling costs for storage and processing is driving interest in extracting competitive value from ever larger amounts of data through analytics and data hungry AI algorithms. In addition, organisations are seeking to exploit opportunities for sharing data within emerging digital ecosystems. W3C has an extensive suite of standards relating to data that were developed over two decades of experience. These include core standards for RDF, the Semantic Web and Linked Data.

A W3C Workshop is now planned for early 2019 on emerging standardisation opportunities, e.g. query languages for graph databases and improvements for handling link annotations (i.e. embracing property graphs), support for enterprise-wide knowledge graphs, different forms of reasoning that are suited to incomplete, uncertain and inconsistent knowledge, AI and Machine Learning, approaches for transforming data between different vocabularies with overlapping semantics, signed graphs, what's next for remote access to data and information services. In addition, W3C hosts many Community Groups working on data standards and we are interested in what is needed to better support work on vocabulary standards.

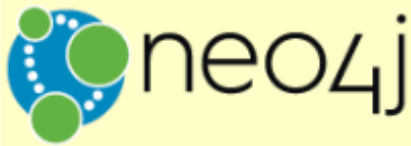
See this Workshop's [Call for Participation](#). Further background is given below.

Graph Databases and Link Annotations

Businesses relied on relational databases (RDBMS) for many years using SQL for query and update. More recently we have seen the rise of NoSQL databases that address the need for flexible handling of unstructured data with key-value stores, document stores, and graph databases. One example is [CouchDB](#) which uses JSON for data storage with ready support for replication for speedy access at different sites. NoSQL is good when you need agility to deal with ever changing data models.

1 Host


W3C gratefully acknowledges Neo4J for hosting this workshop.



2 Sponsors

Platinum, Gold and Silver Sponsorships available

Platinum Sponsors



Gold Sponsors

Why having this workshop?



Issues leading to the Workshop 1.

- Increasing importance of graph-based data and databases in general (witness the large attendance of the workshop on Monday!)
- The concept of Property Graphs has come to the fore (alongside RDF)
 - there is a need to find a way to see how these technologies coexist
 - discussions are ongoing on the pro-s and cons of RDF vs. PG
 - *PG is part of the graph data landscape for good!*
- ISO is also present in this area
 - there is a group combining PG and SQL

Issues leading to the Workshop 1.

In theory...

- SQL could be extended to do everything for graphs
- SPARQL could be extended to do everything for PG and tables
- A property graph GQL that handles tables and graphs could do everything SQL can do



Issues leading to the Workshop 1.

In practice...

- That would lead to paralysis, or endless wars
- Data communities have very deep social and product roots, and large to huge user bases
- Like humans, they can't get personality transplants...

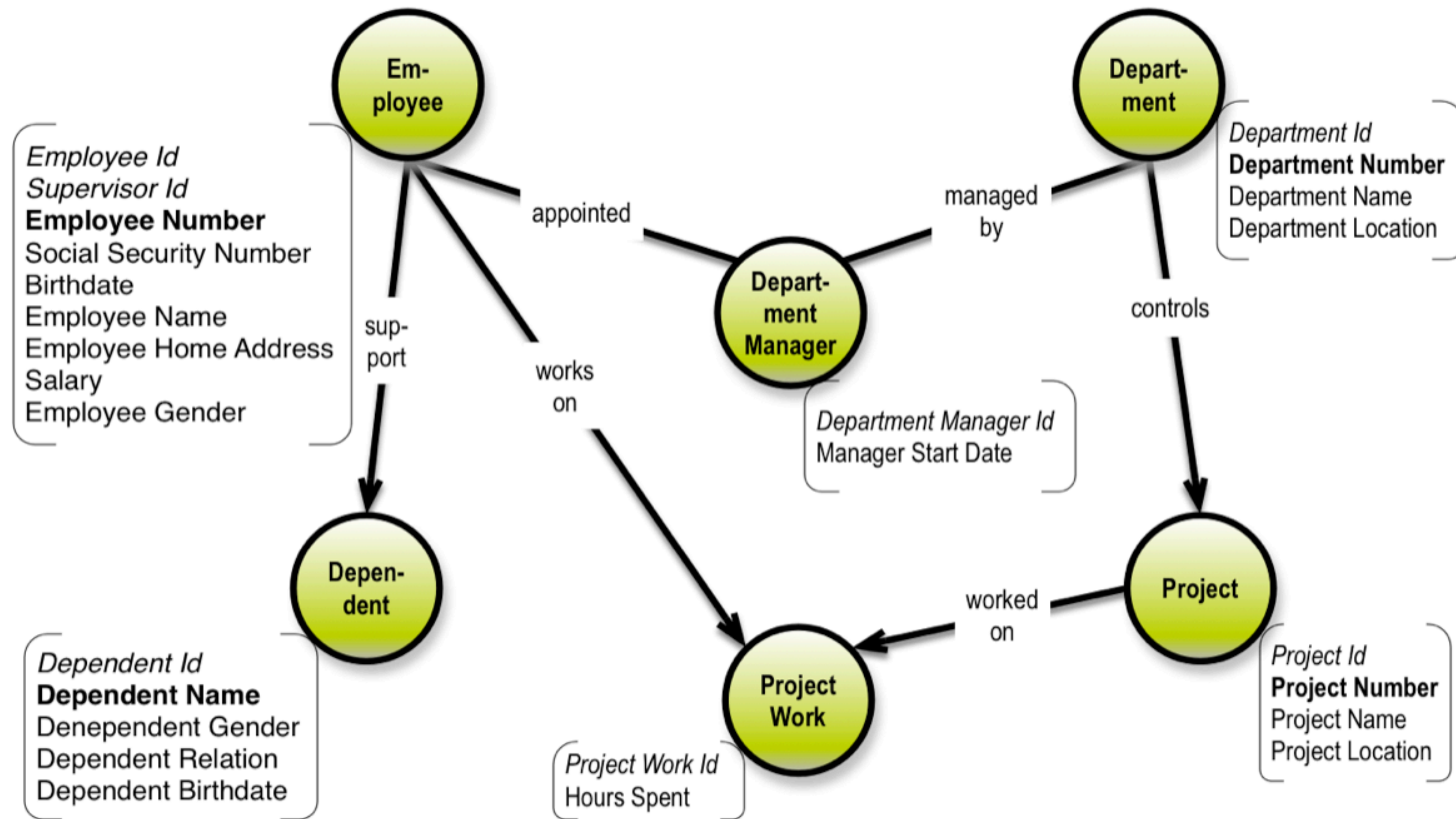


Issues leading to the Workshop 2.

- There are also major concerns with RDF
 - general acceptance is still relatively slow (although there are great successes)
 - there are many minor (or major...) technical issues with RDF & Co. that need housekeeping

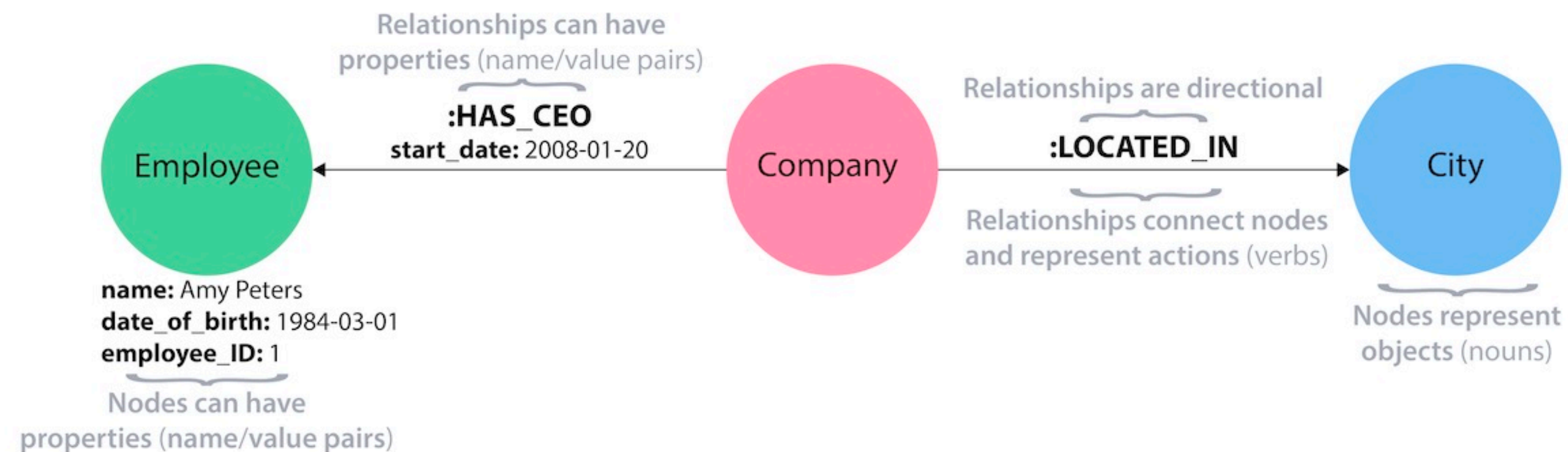
(“RDF”, in the presentation, is a shorthand for full RDF suite, i.e., RDF, RDFS, OWL, SPARQL, SHACL, etc.)

A few words about Property Graphs



Property Graphs

- Framework for representing data and metadata with a graph of nodes and links
 - *both nodes and links may have additional name/value pairs*
 - otherwise referred to as “properties”
 - nodes are “just” nodes, not necessarily URL-s
- Link annotations are very useful to assign temporal, spacial, provenance, etc, information



Property graphs have a real success

- Some non-SQL database vendors (e.g., Neo4j) base their business on this
- There are a also number of smaller (including open source) implementations (e.g, TinkerPop)
- Major database providers (Oracle, Amazon's Neptune,...) incorporate PG as well as RDF stores
 - but they may live in parallel silos...
- There are a number of query languages (declarative and imperative), but not *one* winner (yet)
 - there is work in the ISO/SQL community to incorporate PG, and define query languages

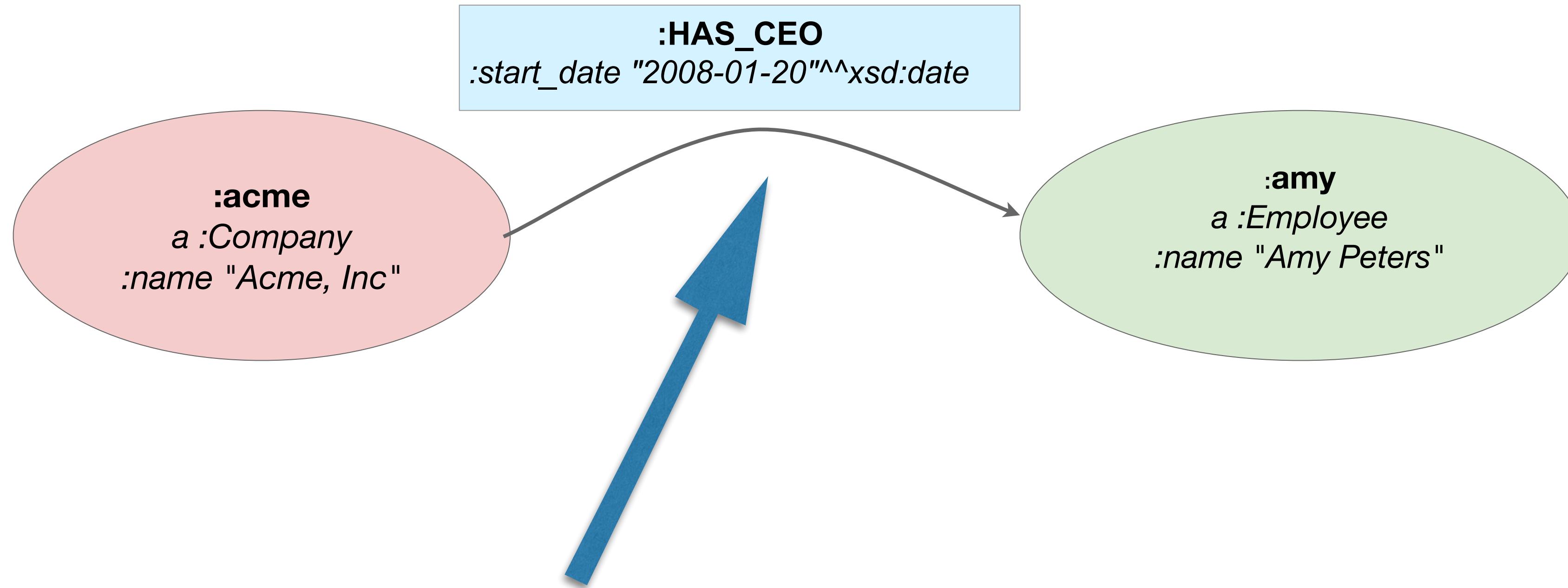
Property Graphs versus RDF: similarities

- Both represent directed graphs as a basic data structure
- Both have associated graph-oriented query languages
- In practice, both are used as “graph stores”, accessible via HTTP and/or various API-s

Property Graphs versus RDF: differences

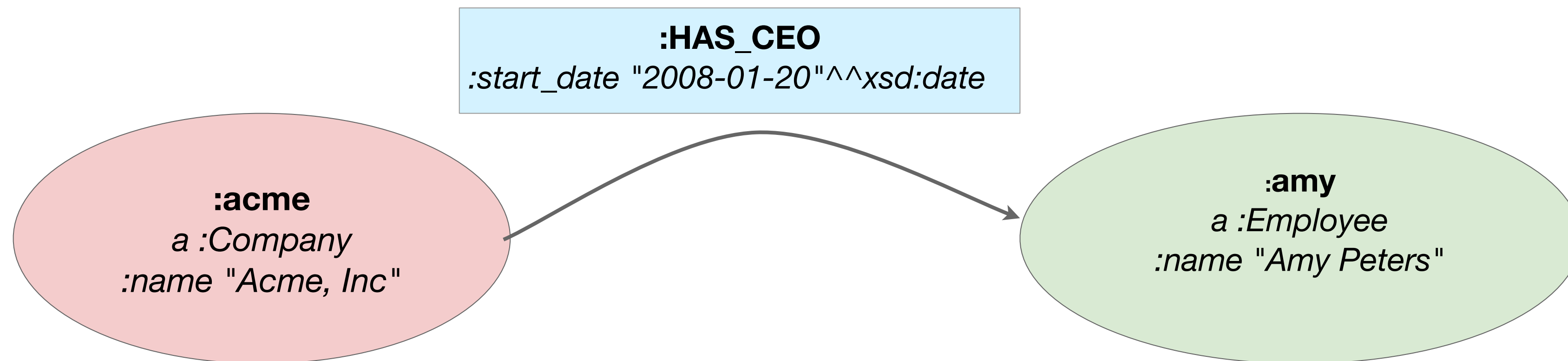
- RDF has an emphasis on OWA, and is rooted in the Web via URL-s. Not the case for PG:
- a PG node is oblivious to what it “contains”: can be a URL, can be a literal
- in RDF parlance, “a Literal can also be a subject”
- Easy to add simple key/value pairs to node, which are not considered to be “in the graph”
- PG-s includes the possibility to add simple key/value pairs to “relationships” (i.e., RDF predicates)

Main difference between PG and RDF



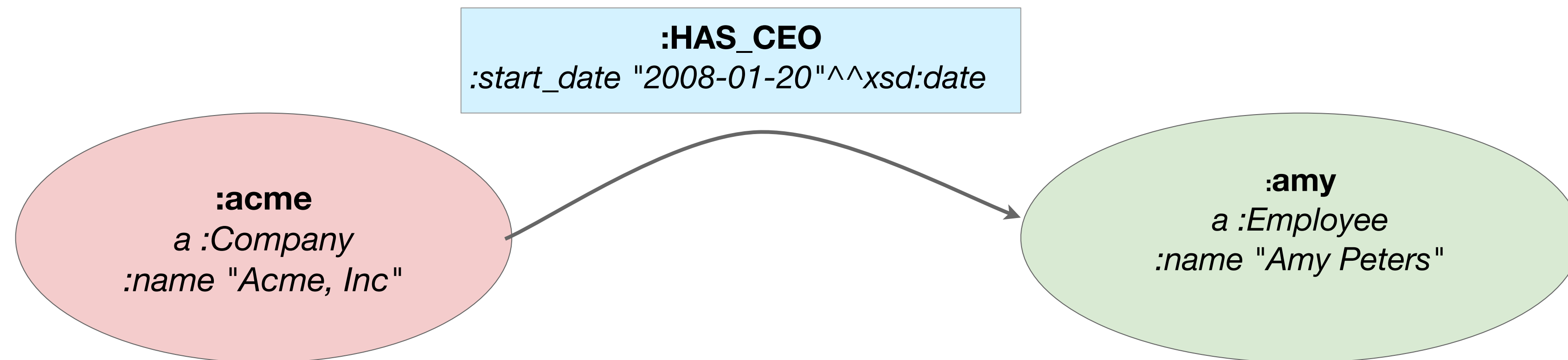
These are properties on the link "instance"!

PG *can* be represented in RDF



- For example:
 - using reification
 - some sort of an intermediate node (usually BNode) to represent the link
 - use a named graph with a single triple
 - extend RDF to include, somehow, a triple as an entity (e.g., “RDF*”)

PG *can* be represented in RDF



- All these representations do exist in real products
- All have pros and cons
 - overall... they are all messy from an RDF point of view 😞
- There is *no generally accepted way of doing that*
 - i.e., none of those solutions are interoperable...
 - databases may offer both models, but little interchange among them...

Why are PG-s interesting for the RDF community?

- They are around on the market...
- They represent, in some ways, a level of abstraction that is easier to understand:
 - by collapsing the “properties” into some sort of labels (i.e., “metadata”), the real, “core” aspect of a graph becomes more visible
 - helps in concentrating on the “essence” of a dataset without being lost in details (date, provenance, tags, etc.)
 - adopting a “PG style” would be actually helpful to make RDF more understandable!

“...historically, property graphs were somewhat of a reaction to the complexity of RDF. A complex standard will not be accepted by the developer community” (Juan Sequeda)

Which leads us to... issues with RDF





- The value of RDF may be well proven, but...



- The value of RDF may be well proven, but...
- *too hard for **average** development teams!*

The “EasierRDF” initiative

- Email discussion initiated by David Booth
 - his original mail in November '18
 - a separate Github Repository has also been set up
- The guiding principles in the startup mail:
 - The goal is to make RDF—or some RDF-based successor—easy enough for average developers (middle 33%), who are new to RDF, to be consistently successful.
 - Solutions may involve anything in the RDF ecosystem: standards, tools, guidance, etc. All options are on the table.
 - Backward compatibility is highly desirable, but less important than ease of use.

Over 600 messages in a few weeks!

The image shows three overlapping browser windows displaying email archives from the semantic-web@w3.org mailing list. The top window shows the main archive page for November 2018, titled "semantic-web@w3.org from November 2018 by thread". It indicates 320 messages and provides sorting options (thread, author, date, subject) and mail actions (mail a new topic, help). Below this, a list of email threads is visible, including "SPARQL survey: results", "Deadline extension*", "URIs declination", "Scoping bnodes", "Robotic 2019: early registration", "Special Issue CFP: Computers & Electrical Engineering Journal - Nov. 30, 2018", "[2nd CFP] 17th Conference on Artificial Intelligence in Medicine", "JSON Ontology Re: Toward easier RDF: a proposal", "N-ary Relations - Toward easier RDF: a proposal", "Electronic Circuits on the Web", "[ANN] Shape Expressions 2.1 release candidate", and "Language-tagged strings Re: Toward easier RDF: a proposal".

The middle window shows a detailed view of a thread titled "Toward easier RDF: a proposal" by David Booth (Wednesday, 21 November). It lists several replies, including "Re: Blank Nodes Re: Toward easier RDF: a proposal" by David Booth, Thomas Passin, and others, and "Re: RDF graph merging: How useful is it really?" by various contributors.

The bottom window shows a thread titled "Identity problems numbers 3 and 5" by Hugh Glaser (Friday, 7 December). It lists numerous replies, including "Re: Identity problems numbers 3 and 5" by Thomas Passin, "Re: rdf.stackexchange.com -- Identity problems numbers 3 and 5" by David Booth, and "new semantic web stackexchange proposal opened" by Henry Story, among others.



EasierRDF github site: 50+ issues

The screenshot shows the GitHub repository for **w3c/EasierRDF**. The repository overview on the left includes the title "Making RDF easy enough for average developers", 35 commits, 1 branch, and a list of files including README.md. The right pane displays a list of 50 open issues, each with a category label and a comment count.

Issue Title	Category	Comments
Standardized n-ary relations (and property graphs)	language features	1
Moribundity of Tools	tools	1
SPARQL-friendly lists	language features	2
Blank nodes	language features	24
Beginner friendly tutorials / documentation	education	1
Idea: Higher-level RDF language	big Ideas	3
Overview of an RDF triple store	tools	
Lack of Technology Framing	education	
SPARQL Triplestore and Reasoning Performance	tools	1
Lack of a Good Editor	tools	22
SPARQL: The unnamed/default graph should have a standard name	related standards	5
Lack of standard RDF canonicalization	related standards	



RDF issues at the Workshop

- The “EasierRDF” discussion was one of the main inputs
- There were also a number of other sessions: rules, temporal and spatial data, streaming, outreach, queries...
- Obviously, the workshop could only try to enumerate the main issues
- There were, roughly, three types of issues that came up:
 1. technical issues: deficiencies, missing features, etc...
 2. “outreach” issues
 3. tooling

**A rough list of top RDF issues from the Workshop
(caveat: there is no systematic review yet, this is my list...)**



Technical issues

- Lack of n-ary relations
- Blank nodes
 - do we need them, should we restrict their usage, leave it as they are?
- Simplified reification of some sort (RDF*/SPARQL*)
- *A simple reasoning system*
 - OWL is usually considered to be way too complex for the average developers
 - n3 based? SPARQL based? something else?
- RDF for stream processing

Technical issues (cont.)

- Representation of time in RDF
- Clearer semantics of data sets
- Security, integrity, provenance, etc., of data
 - related: missing standard for the canonicalization/signature of graphs
- Better internationalization of Literals (base directions, hints for translations, pronunciations, ...)
- Text search
- RDF model extensions?
 - literals as subjects? blank nodes as predicates?
- *Relationship to Property Graphs*

Non-technical issues

- Lack of *beginner level* good tutorials
 - no equivalence to, say, MDN
 - no clear “entry” points for outsiders
- Too much jargon that are unrelated to Web Developers’ experiences
- No (not yet?) proper and standard integration with Javascript
 - there is a W3C Community Group working on this, though...
- Moribundity of tools, registries, lots of abandonware
- A general question: is RDF too low (“assembly”) level, is there a need for a higher level model to make it more usable?

Results of the Workshop



Results of the Workshop: many ideas came up for future activities

- Standards work around PG
 - an abstract (standard) model for Property Graphs†
 - standard mapping between Property Graphs and RDF
 - standard mapping between Property Graphs and Relational Data†
 - W3C Community Group for Graph Query Language (GQL)†
- RDF improvements
 - solve all the technical and outreach problems in RDF 😊

† Final work probably *not* at W3C



But... this can lead to chaos

- It would lead to lots of unstructured, unrelated work, not necessarily in the right priority order
- Final decision is to set up a *W3C Business Group* to coordinate further work

W3C Business Group on Graph Data

- Look at the bigger story around data: data is strategic asset for companies. What are the features and mappings that are of importance?
- *Derive a prioritized list of technical issues to be solved to fulfill those needs*
- Spin off task forces, community groups, etc, to look at the technical issues that are of major importance
- Liaise with other organizations (e.g., ISO) for the activities that are to be done elsewhere
- Look at outreach possibilities in general

**Watch this space,
interesting things will happen!**



Some links

- Workshop home page:
 - <https://www.w3.org/Data/events/data-ws-2019/>
- All submissions
 - <https://www.w3.org/Data/events/data-ws-2019/papers.html>
- Workshop agenda with links to slides
 - <https://www.w3.org/Data/events/data-ws-2019/schedule.html>
- Workshop report
 - <https://www.w3.org/Data/events/data-ws-2019/report.html>
- These slides:
 - <https://www.w3.org/2019/Talks/W3C-track-IH/Presentation.pdf>

Thank you for your attention

