

Welcome to INFO216: Knowledge Graphs

Andreas L. Opdahl
<Andreas.Opdahl@uib.no>



About me

- Background:
 - siv.ing (1988), dr.ing (1992) from NTH/NTNU
 - Univ. of Bergen (early 1990-ies)
 - part-time programmer / consulting for industry
 - several Forskningsråd and EU projects and networks
- Central research interest:
 - modelling of information systems and enterprises
 - semantic modelling and modelling languages
 - semantic technologies
 - knowledge graphs in the media sector



Recent project: BDEM

- Leveraging *Big Data for Emergency Management*
 - how can semantic technologies play a part?
 - developed a new Master course: INFO319



SAN DIEGO STATE
UNIVERSITY




VESTLANDSFORSKING

Recent project:  UBIMOB



WESTERN NORWAY RESEARCH INSTITUTE
VESTLANDSFORSKING

 NTNU
Norwegian University of
Science and Technology



tøi Transportøkonomisk institutt
Stiftelsen Norsk senter for samferdselsforskning



 The
University
Of
Sheffield.

 telenor

Ending project: Transfeed

Test feedback to the driver

Social costs of driving
(Road Pricing)

Eco driving

*Data integration +
machine learning*

Effects

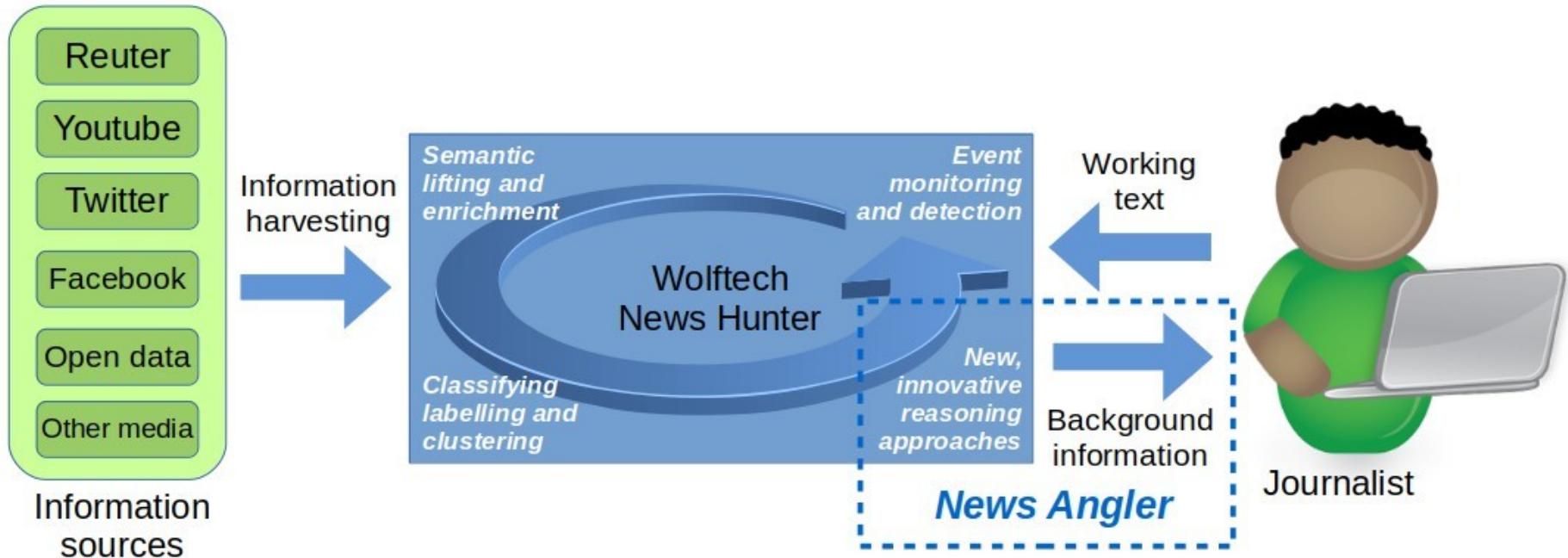
*Can automated
feedback encourage
more eco-friendly
driving behaviour?*

Emission reductions?

Change travel time or destination?



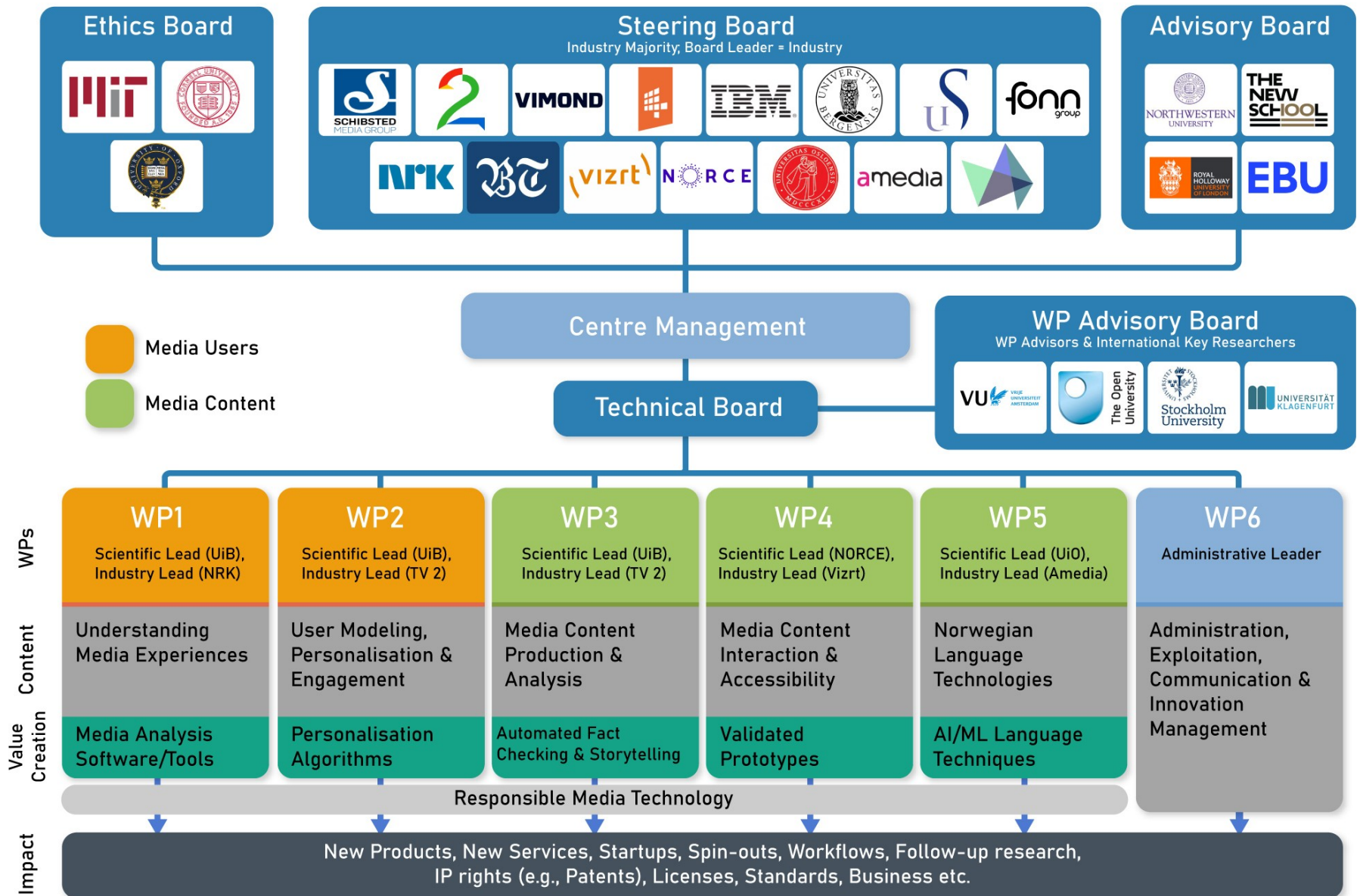
Ongoing project: News Angler



“Wolftech News supports and improves the workflows in a newsroom through mobile solutions for field work that are integrated with central systems for news monitoring, resource management, news editing, and multi-platform publishing”

- 1) Harvesting and analysing messages
 - 2) Growing a semantic news graph
 - concepts, named entities, context...
 - 3) Analysing working texts (stories)
 - 4) Identifying background information
 - 5) Prioritising and preparing
 - 6) Journalistic and editorial preferences
- Research: graph, searches, preparation, preferences, language, scaling*

New research centre: Media Futures



Session 1

- Themes:
 - *what are knowledge graphs (KGs)?*
 - our focus on semantic knowledge graphs
 - ...and what are *linked data*, *semantic technologies*, the *semantic web*, ...?
 - *introduction to INFO216*
 - organisation of the course
 - practical information
 - *a little about programming KGs*
 - from Java to Python
 - a little about RDFLib for Python



Readings

- Sources:
 - Blumauer & Nagy (2020):
Knowledge Graph Cookbook – Recipes that Work
(pages 27-55, 105-122)
 - Allemang & Hendler (2011):
Semantic Web for the Working Ontologist
(chapters 1-2)
- Material at <http://wiki.uib.no/info216>:
 - Tim Berners-Lee talks about the semantic web
 - stuff about RDFLib for Python (for lab 1)
 - for Java, Jena is an alternative



Knowledge Graphs



Call for a transition

- From a *Web of Documents*
 - today, the “plain old web” (PoW)
 - document-centric
 - document-to-document links
 - for humans
- to a *Web of Data*
 - ...*semantic web, web 3.0, Linked Open Data (LOD), Web of Knowledge, the Giant Global Graph (GGG)*
 - document- *and data-centric*
 - doc-to-doc *and data-to-data links*
 - for humans *and machines*

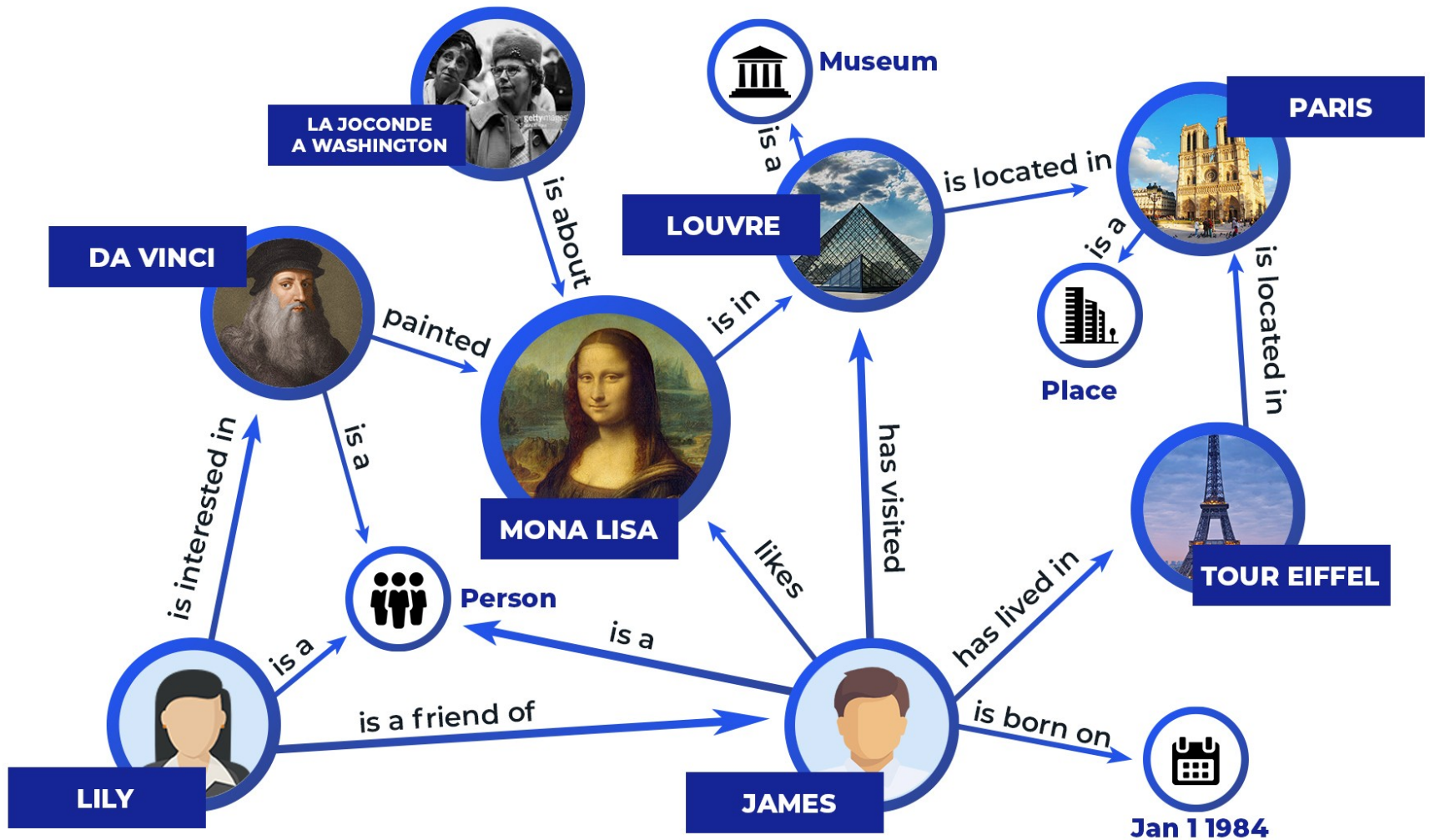


Tim Berners-Lee's challenge

- There's an enormous amount of data on the web
 - ...but the data are mostly not linked
(think of a world wide web without document links!)
 - availability, accessibility does not go all the way
 - *what if we had standard ways of representing data so that linkable data could always be automatically linked?*
 - *enormous potential to solve, simplify, speed up... many critical information handling problems*
- This is the purpose of *semantic technologies*
- This is the vision that led to today's *semantic knowledge graphs*

Tim Berners-Lee: <<http://www.youtube.com/watch?v=HeUrEh-nqtU>>



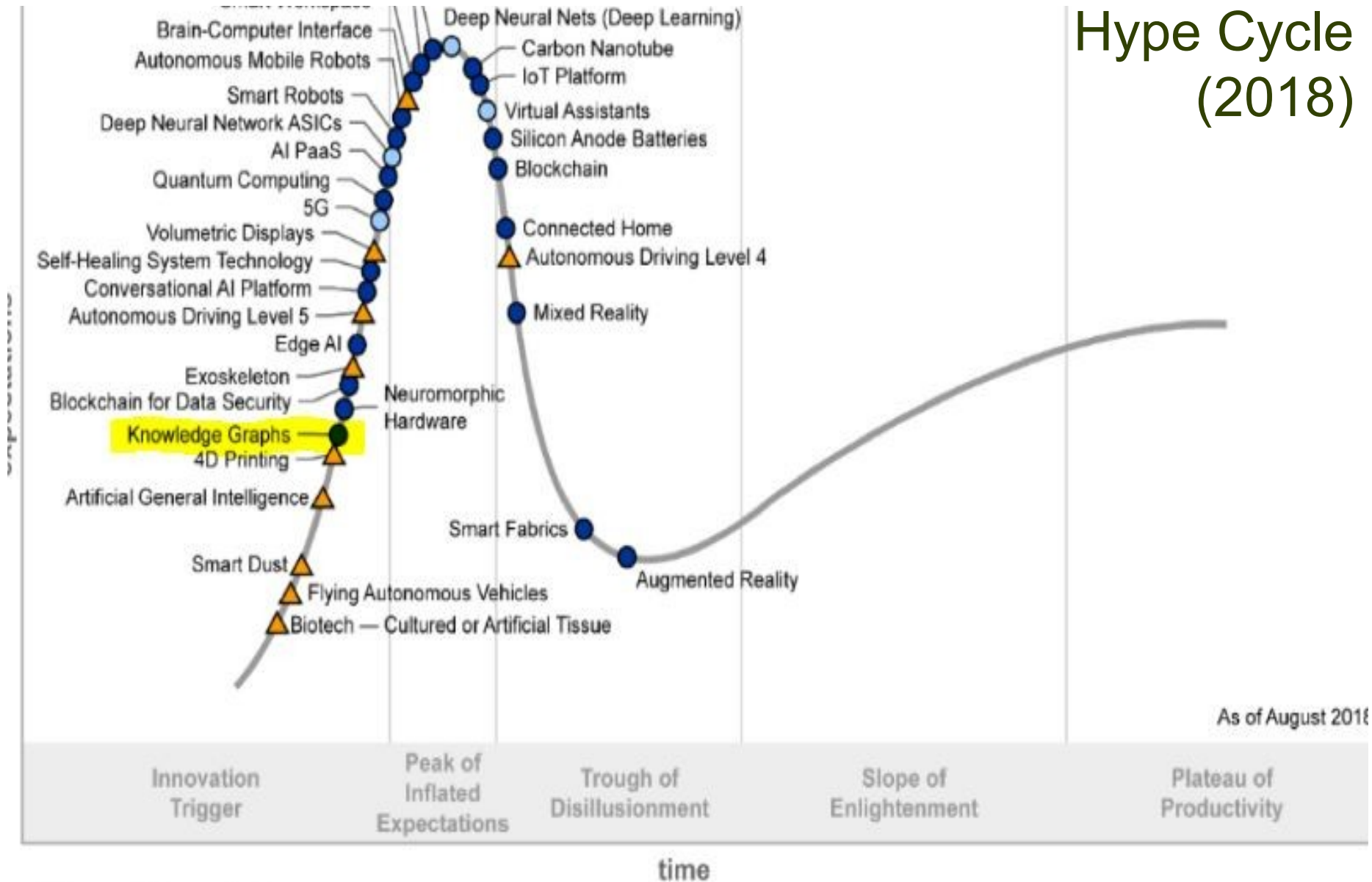


- *Graphs* of *nodes* connected by directed *edges*
- Represents *knowledge* as connected *facts*
- *Nodes* repr. *resources* or *values*, edges repr. *relations*

Why knowledge graphs?

- Ease of exchanging, reusing information
 - inherent semantics become clearer
 - less dependency on context
- Ease of interlinking, enhancing information
 - semantic data can be combined in new ways
 - open reference datasets
 - general and specialised knowledge bases
- Schema independence
 - no pre-defined schemas (“schema-on-read”)
 - news types of entities and new relations can be added freely
- *Well-matched with big data and machine learning!*

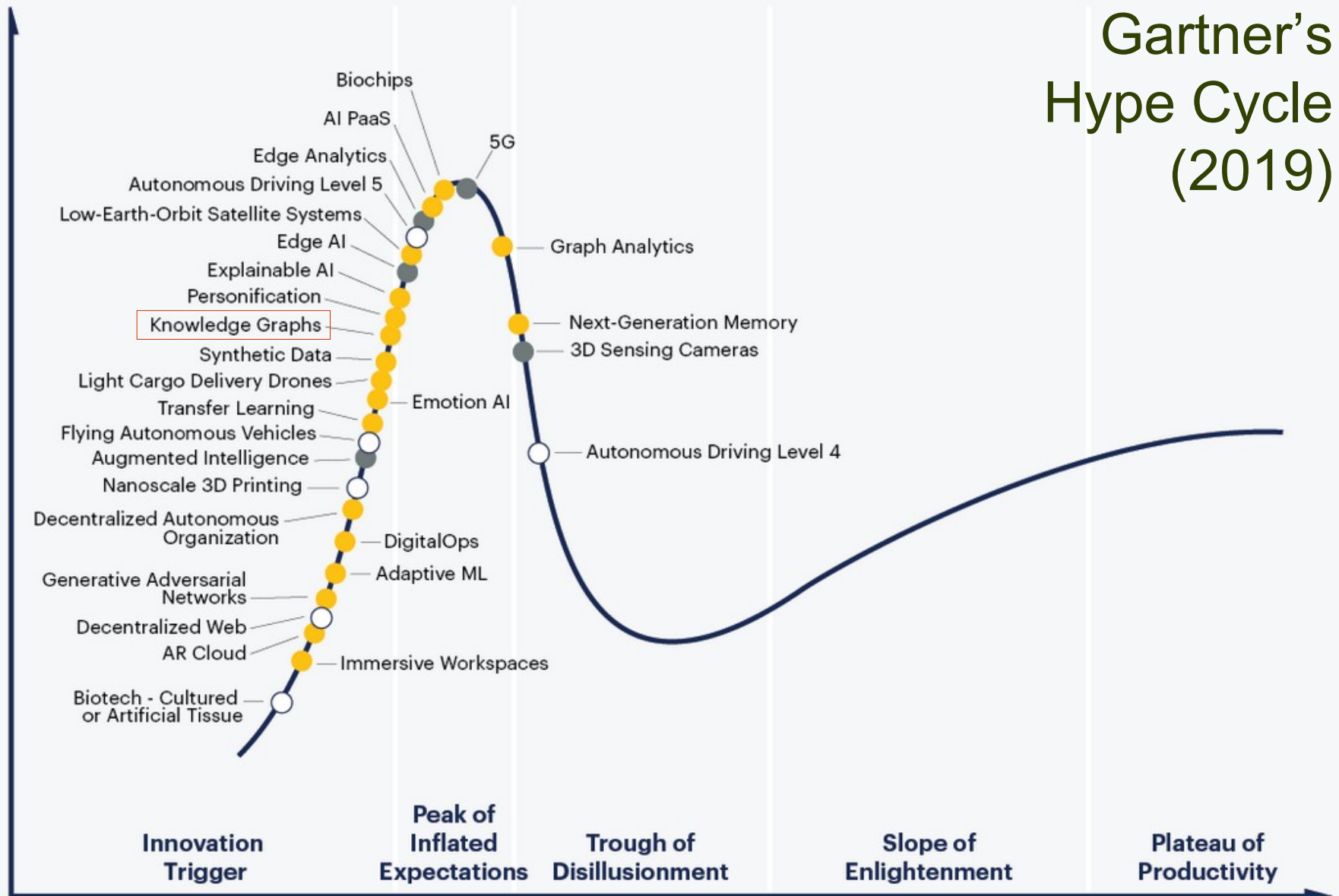
Gartner's Hype Cycle (2018)



Plateau will be reached:

Gartner's Hype Cycle (2019)

Expectations



Plateau will be reached:

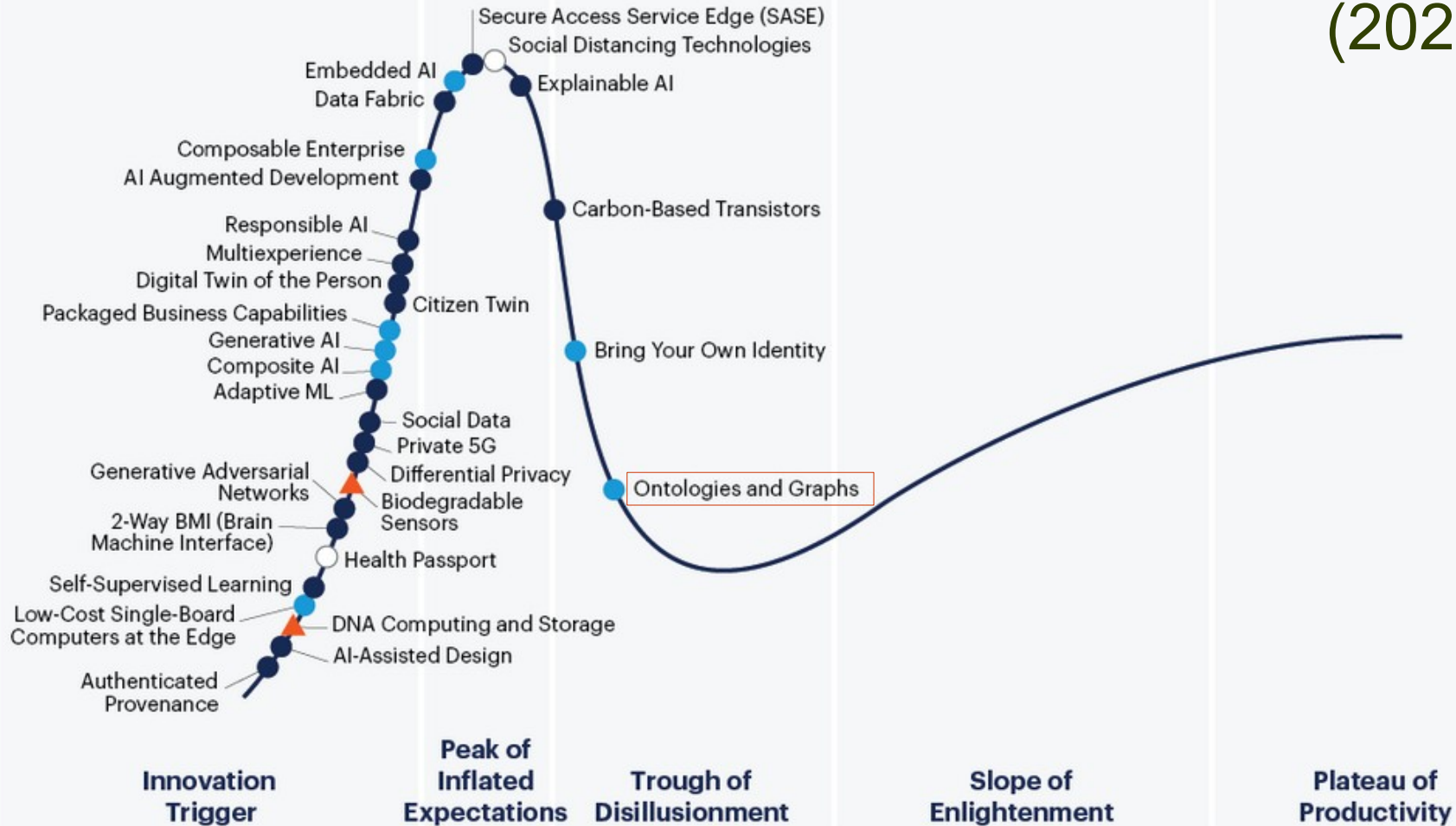
- less than 2 years
- 2 to 5 years
- 5 to 10 years
- more than 10 years
- obsolete before plateau

As of August 2019

Time

Gartner's Hype Cycle (2020)

Expectations



Plateau will be reached:

○ less than 2 years

● 2 to 5 years

● 5 to 10 years

▲ more than 10 years

⊗ obsolete before plateau

As of July 2020

Time

Resource Description Framework (RDF)



How can we represent meaning?

- Only in part - meaning is a *complex, layered concept*
- Semantic data:
data with associated metadata about its meaning

- *Vocabularies* can capture certain aspects of meaning:
 - standard URIs for *properties* and *types of resources*
 - standard URIs for *properties*
 - standard types for *literals*
 - *rules* about how they combine
- Other *open semantic datasets* define:
 - standard URIs for and *facts* about *individual resources*

How can we represent meaning?

- Many formats are possible
- Semantic knowledge graphs rely heavily on the *Resource Description Framework (RDF)*
 - a “normal form” for semantic data (data with associated metadata about its meaning)
 - usable both for the data and their metadata
 - both are represented as KGs
 - either *native/reified*, *embedded*, or *virtual*
- More expressive vocabularies are available as KGs
 - *RDF Schema (RDFS)*, “*RDFS Plus*”
 - *Web Ontology Language (OWL)*
 - *all* (can be said to) *build on RDF*



Other types of knowledge graphs

- *Labelled Property Graphs (LPG)*
 - becoming increasingly popular
 - not inherently semantic/linked
 - but can be used semantically, e.g., to store RDF
 - has so far not been standardised:
 - different tools use different query languages, exchange formats
 - standardisation is moving quickly forward
- Our focus remains on *RDF-based knowledge graphs*:
 - what we call *semantic knowledge graphs*



Other types of knowledge graphs

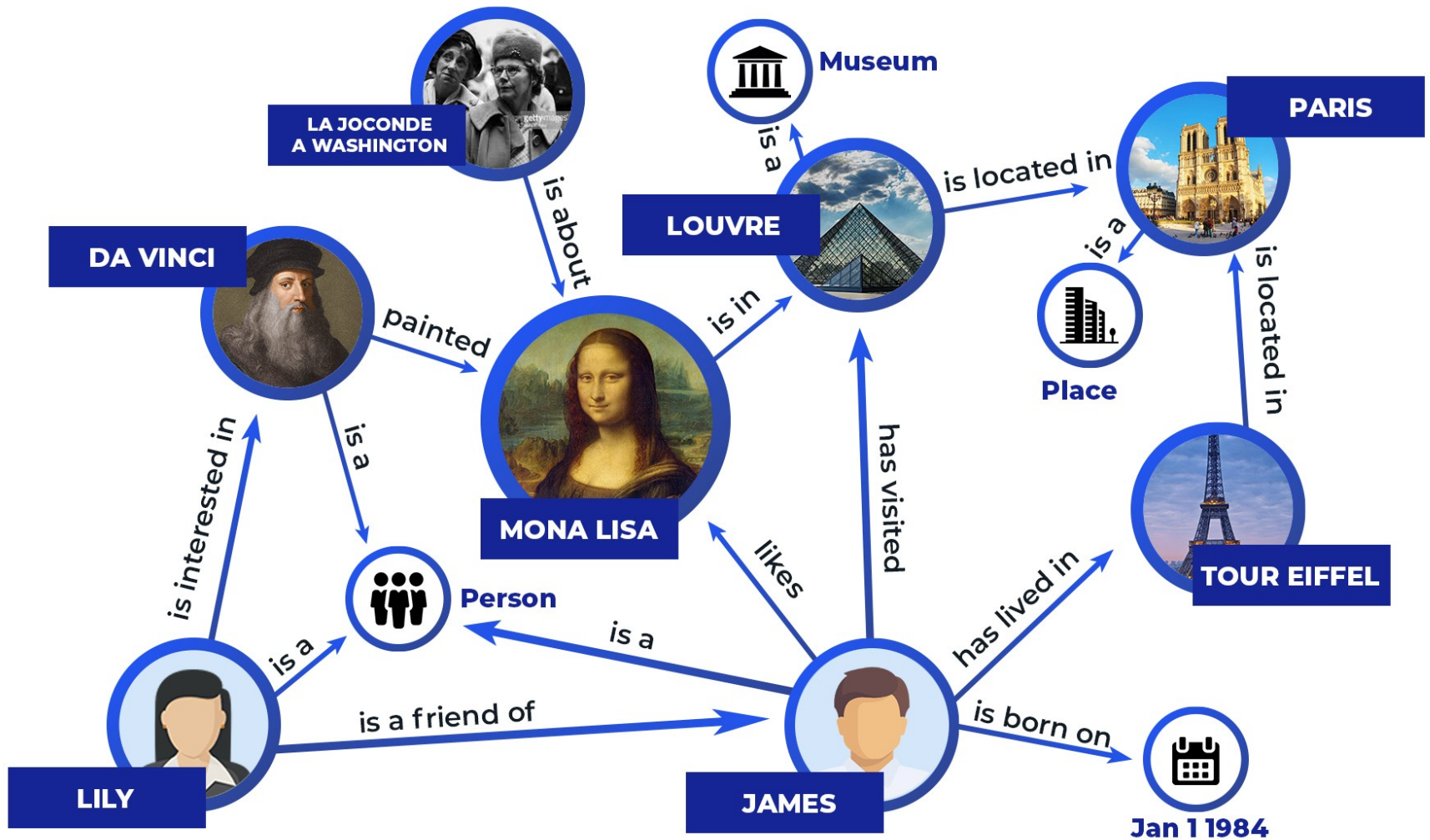
- *Non-semantic knowledge graphs*
 - many recent ML approaches use graph data
 - e.g., graph embeddings, link prediction
 - but the graphs are not necessarily *dereferenced*
 - they can use human-understandable labels
 - but they do not use standard URI
 - but can be used semantically too, e.g., on RDF data
- Our focus remains on *RDF-based knowledge graphs*:
 - what we call *semantic knowledge graphs*



Resource Description Framework

- Represents data as triples (facts):
 - *(subject, predicate, object)*
 - the *subject*:
 - represents what the triples (fact) is about
 - the URI of a semantic resource
 - the *predicate*:
 - represents a property of the subject resource
 - the URI of a semantic property
 - the *object*:
 - represents the value of a property for a subject
 - either: the URI of a semantic resource
 - or: a literal (number, string, boolean...)





- Resources can be:
 - physical things (people, places, organisations...), information, concepts...

Semantic graphs and data sets

- *Graph*:
 - a collection of *triples (facts)* (possibly none)
- *Data set (or “Conjunctive graph”)*:
 - a collection of graphs (at least one)
 - one of the graphs is *default/unnamed*
 - the others are *named*
 - from triples:
 - *(subject, predicate, object)*
 - to quadruples (*quads*):
 - *(graph/”context”, subject, predicate, object)*



Resources

- http://conceptnet.io/c/en/mona_lisa
- <https://imdb.uib.no/bg-conceptnet/#query>
 - `<http://api.conceptnet.io/c/en/mona_lisa>`
- <http://www.wikidata.org/wiki/Q14128>
 - `.../wiki/Special:EntityData/Q14128.ttl`



Background



Not a single coordinated effort...

- *Many independent, but related developments:*
 - Semantic Web, Web of Data, contextual web: making the web data-oriented, semantic, machine-processable (from 2000)
 - microformats, Microdata: weaving facts, semantics, and small knowledge graphs into HTML pages
 - semantic technologies: reusable technologies and tools for handling semantic data: RDF, SPARQL, OWL...
 - social tagging: large-scale semantic tagging produced by social media (around 2005)



Not a single coordinated effort...

- *Many independent, but related developments:*
 - the Linked Open Data (LOD) cloud:
interlinking semantic datasets, making them openly available: DBpedia (2007-), Wikidata (2012-)...
 - “Giant Global Graph” (GGG):
global interlinking of open knowledge graphs (\approx LOD)
 - company-internal semantic data:
linked open data and semantic technologies used inside an enterprise or cluster, “enterprise knowledge graph”
 - *knowledge graphs:*
general term for semantic graph representations of (primarily) factual information (from 2012)
 - may or may not use RDF

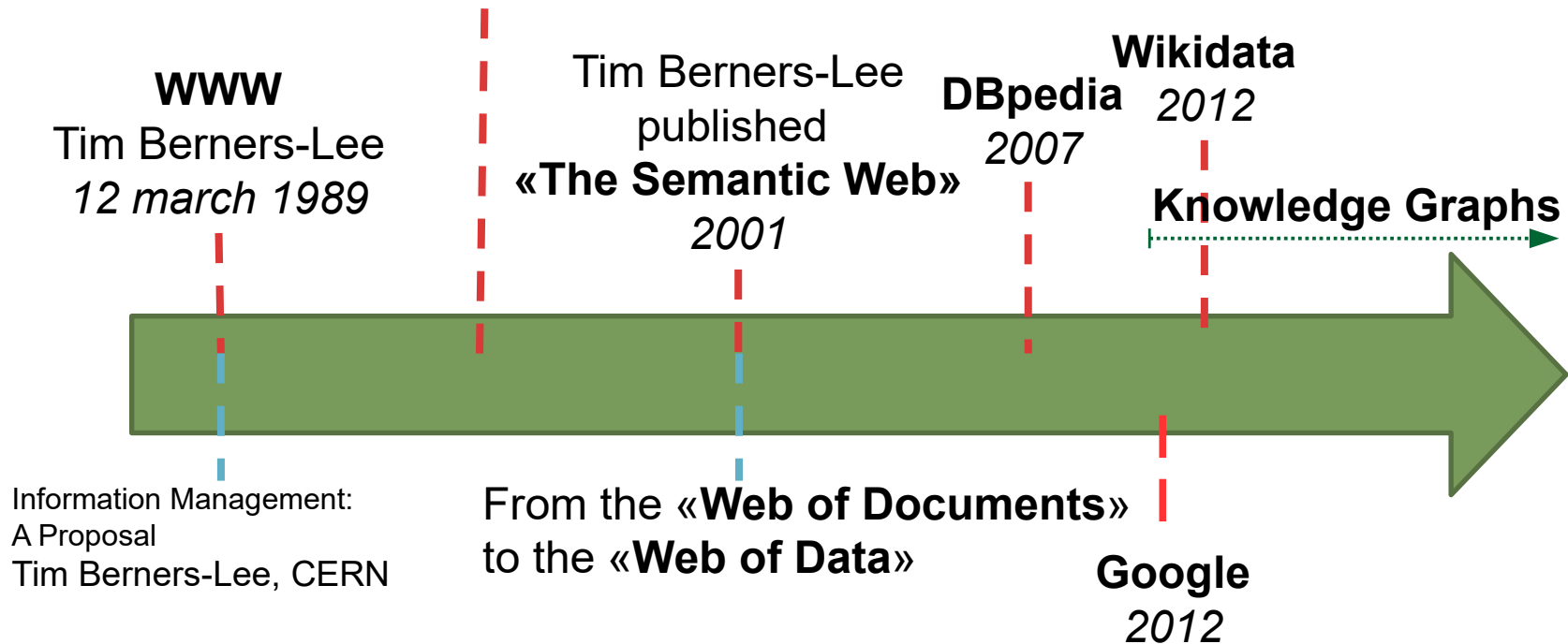


Semantic web and WWW history

Weaving the Web (1999)

The original design and ultimate destiny of the World Wide Web, by its inventor

<https://www.w3.org/People/Berners-Lee/Weaving/Overview.html>



Tim Berners-Lee: <http://www.youtube.com/watch?v=HeUrEh-nqtU>

Information Management: A Proposal: <https://cds.cern.ch/record/369245/files/dd-89-001.pdf>

Common themes

- Semantically tagged data
- Well-defined tags (terms)
 - defined in standard vocabularies
 - formal ontologies, description logic
- Graph representations of knowledge
 - RDF, RDFS
 - more recently: labelled-property graph databases
- Standard exchange formats
 - (re-)using the same APIs, SPARQL endpoints, etc.
- From the start open, community-based
 - (re-)using many of the *same technologies*



Linked Open Data (LOD)

- 3-4 basic principles (Berners-Lee 2006):
 1. URI-er (Uniform Resource Identifier) *identify resources*
 - <http://dbpedia.org/resource/Bergen>
 2. URI-s *answer to HTTP requests (dereferencing)*
 - for example SPARQL queries, Turtle files, ...
 3. Returns *information about the resource* on standard format, e.g.,
 - RDF/XML, Turtle, N3, JSON-LD (JSON, XML, CSV, TSV, HTML)
 - may use “303 redirection” to distinguish the Concept from the Information about it, e.g.,
 - <http://sws.geonames.org/3161732/>
 - <http://sws.geonames.org/3161732/about.rdf>
 4. The information contains URI-s that *identify related resources*

Best Practices for Data Provisioning

- Recommended directly by W3C
 - or emerged within the LOD community:
 1. *Provide dereferencable URIs*
 2. *Set RDF links pointing at other data sources*
 3. *Use terms from widely deployed vocabularies*
 4. *Make proprietary vocabulary terms dereferencable*
 5. *Map proprietary vocabulary terms to other vocabularies*
 6. *Provide provenance metadata (e.g., PROV)*
 7. *Provide licensing metadata (e.g., CC)*
 8. *Provide dataset-level metadata (e.g., VANN, VS)*
 9. *Refer to additional access methods (e.g., SPARQL)*



The LOD cloud

- <http://lod-cloud.net/>
 - which datasets mention resources in other datasets?
 - 1269 datasets with 16201 links between them
 - ≥ 1000 triples, ≥ 50 links to other datasets
 - started in 2007
 - exponential-like growth for a few years
 - still growing, but more slowly now
- <http://lodstats.aksw.org/> (offline...):
 - ca 150G (150 000M) triples from ca 3000 data sets
 - most from SPARQL endpoints, some from file dumps



Knowledge graphs are everywhere!



And many others...

- BBC's content management, ontologies, BBC Things
- Google, Bing, Yahoo... (schema.org) (2011)
- Google's Knowledge Graph (2012), Microsoft's Satori
- Facebook's Open Graph and Graph Search (2013)
- Thomson Reuters, Bloomberg...
- Amazon's Product Graph (2017), Neptune
- Uber Eats' food graph

Frank van Harmelen's keynote at CAiSE 2018.



Organisation of the course



Requirements

- Required Previous Knowledge
 - basic data skills in *data management* and artificial intelligence
 - medium level skills in *programming*
- Recommended Previous Knowledge
 - INFO125 Data Management
 - INFO132 *Programming*
 - INFO135 Advanced Programming
 - INFO180 Methods in Artificial Intelligence



Curriculum

- Mandatory:
 - Blumauer & Nagy: Knowledge Graph Cookbook, 2020
 - lectures and lecture notes (*textbook*)
 - electronic materials in the wiki: wiki.uib.no/info216
 - introductions, tutorials
 - standards documents
 - academic papers
- Cursory:
 - Allemang & Hendler: Semantic Web for the Working Ontologist, 2nd ed. 2011 (*former textbook*)
 - supplementary materials in the wiki: wiki.uib.no/info216



Theory lectures (tentative)

1. Knowledge graphs
2. RDF
3. SPARQL
4. Tools and services
5. RDFS
6. OWL 1
7. Vocabularies & ontologies 1
8. Vocabularies & ontologies 2
9. Open KGs 1
10. Open KGs 2
11. Enterprise KGs
12. OWL 2
13. Rules and reasoning
14. Managing KGs

You learn KG programming (mostly) through the lab exercises and the group project!



Lectures and labs

- 14 lectures:
 - Monday 0815-1000
 - mostly theory
 - maybe some workshop-style parts
- 14 lab sessions <<http://mitt.uib.no>> **starting next week**
 - lab leaders:
 - Markus Andre Pedersen <markus.pedersen@uib.no>
 - Sindre Asplem <Sindre.Asplem@student.uib.no>
 - **strongly recommended**, but not mandatory this spring
- 14 seminars/question sessions (Tuesday 1215-1400)



Evaluation

- Two-part evaluation:
 - individual, written 3-hour exam (60%)
 - group project (40%)
- Exam requirements:
 - submitted group project
 - participation in 80% of labs: not mandatory this spring



Group project

- The group project shall develop a *semantic KG-based product (e.g., a dataset, application, or service)*
- Development and run-time platform is free of choice, as is programming language
- Carried out in groups of three and not more
 - working individually / in pairs is possible
 - groups of more than three will never be accepted
- The project will be presented in the seminar groups, and each group member will describe their contribution to the finished product
- The assignment must be done in the teaching semester
- ***...more about that next week!***



Programming RDF (and RDFS, SPARQL...) with Python



RDFLib

- A library and API (Application Programming Interface) for programming RDF and SPARQL in Python
 - simple, powerful and *pythonic*
 - parsers and serialisers for most RDF formats
 - a *Graph* interface
 - with multiple alternative *Stores*
 - SPARQL 1.1 Query and Update
- Other technologies later:
 - a triple store (RDF database): *Blazegraph*
 - an OWL library for Python: most likely *owlready2*



RDFLib interfaces

- **Graph:**
 - an RDF model
 - a Python collection (set) of triples
 - adding, removing, listing, and searching for triples
 - combine with other graphs
 - writing to and reading from RDF files
 - responding to SPARQL queries and updates
 - backed by an in-memory or persistent store
- **Dataset / ConjunctiveGraph:**
 - multiple named RDF models in a dataset
 - a set of quads: triples with graph ids



RDFLib interfaces

- **Triples / statements:**
 - ordinary 3-item Python tuples
 - immutable sequences
 - `>>> triple = (s, p, o) # creates a triple`
 - `>>> s[0] # returns the subject...`
- **URIRef:**
 - a node with a URI
- **BNode:**
 - a blank node with a Graph-internal identifier only
- **Literal:**
 - a typed or untyped value
 - untyped values (strings) can be language-tagged



RDFLib interfaces

- Namespaces:
 - predefined RDF, RDFS, OWL, XSD, FOAF, SKOS, DC, DCTERMS
 - user-defined namespaces, e.g.:

```
>>> i2s = Namespace('http://i2s.uib.no/')
```

```
>>> i2s.MainAuthor
```

```
rdflib.term.URIRef(u'http://i2s.uib.no/MainAuthor')
```

