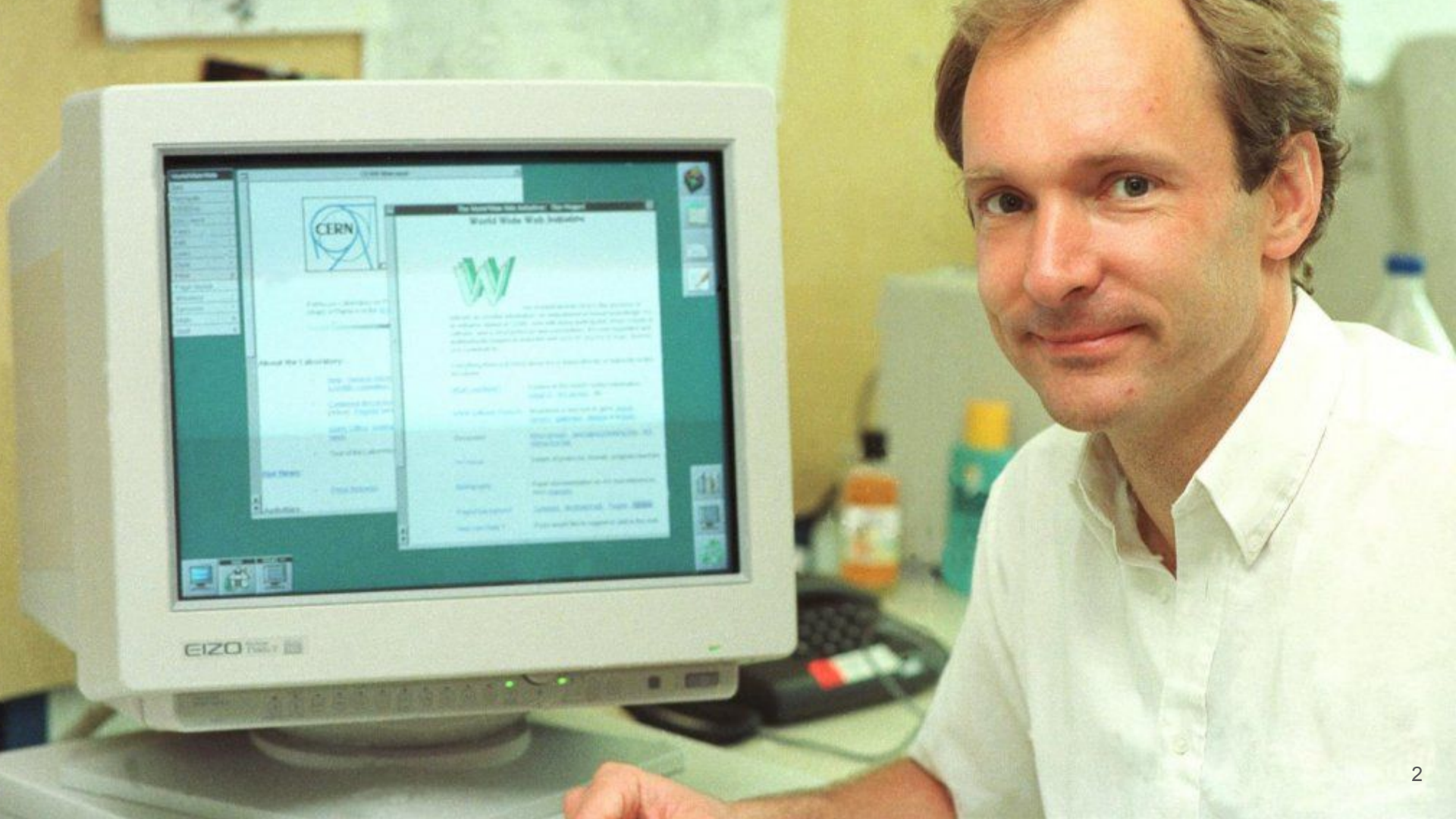
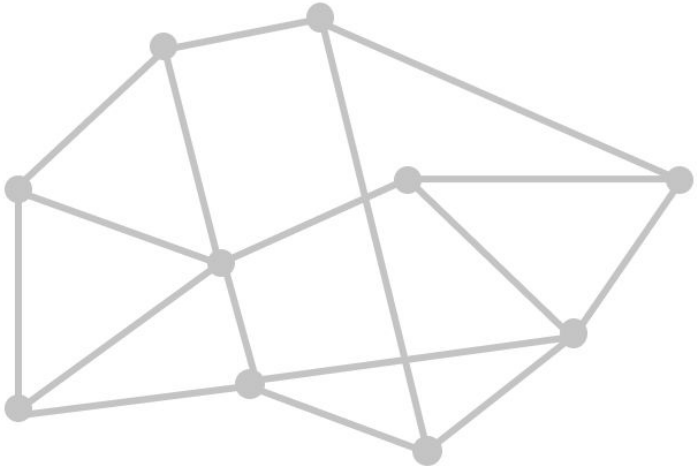
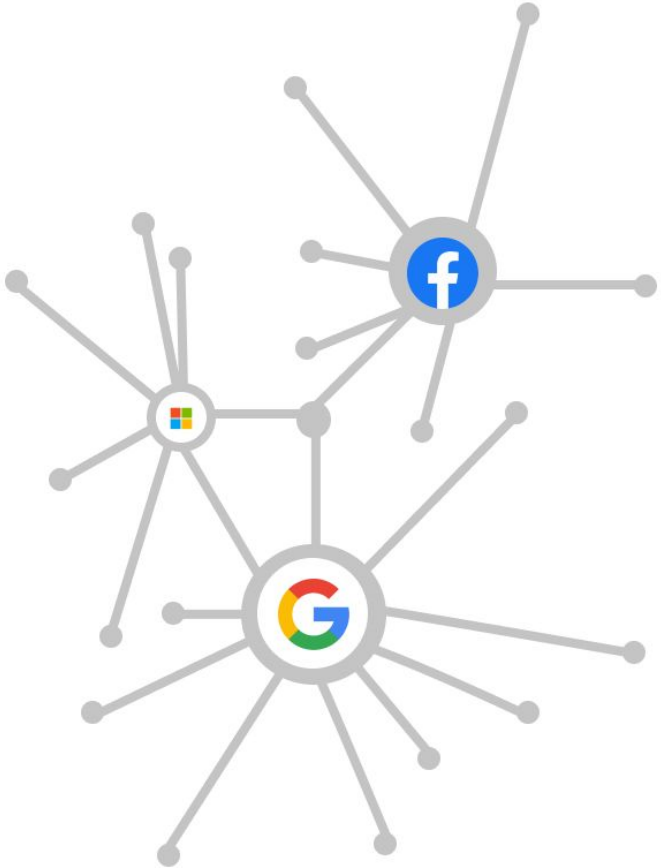
The Semantic Web & Atomic Data

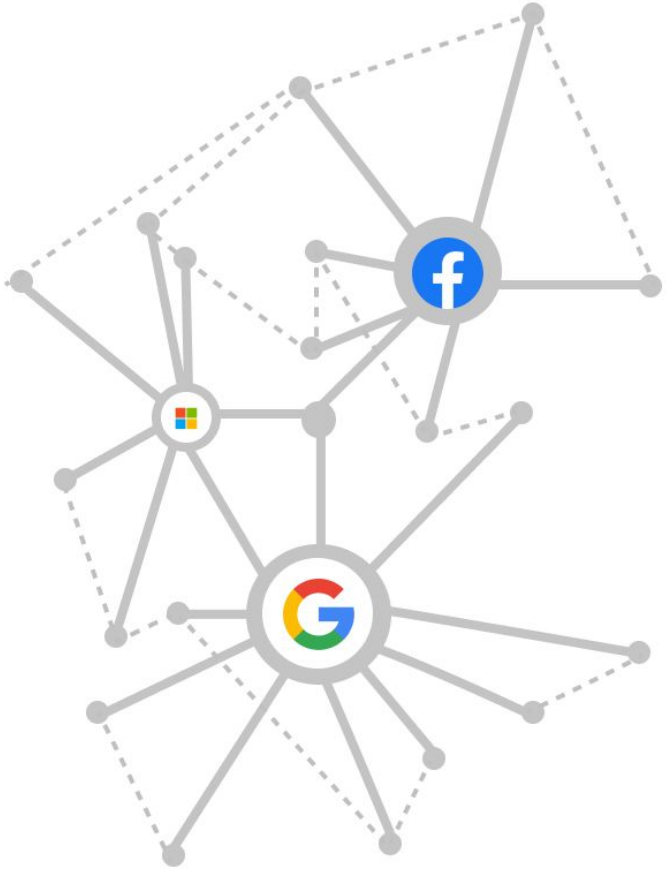
Joep Meindertsma, Declarative Amsterdam 2022-11-08

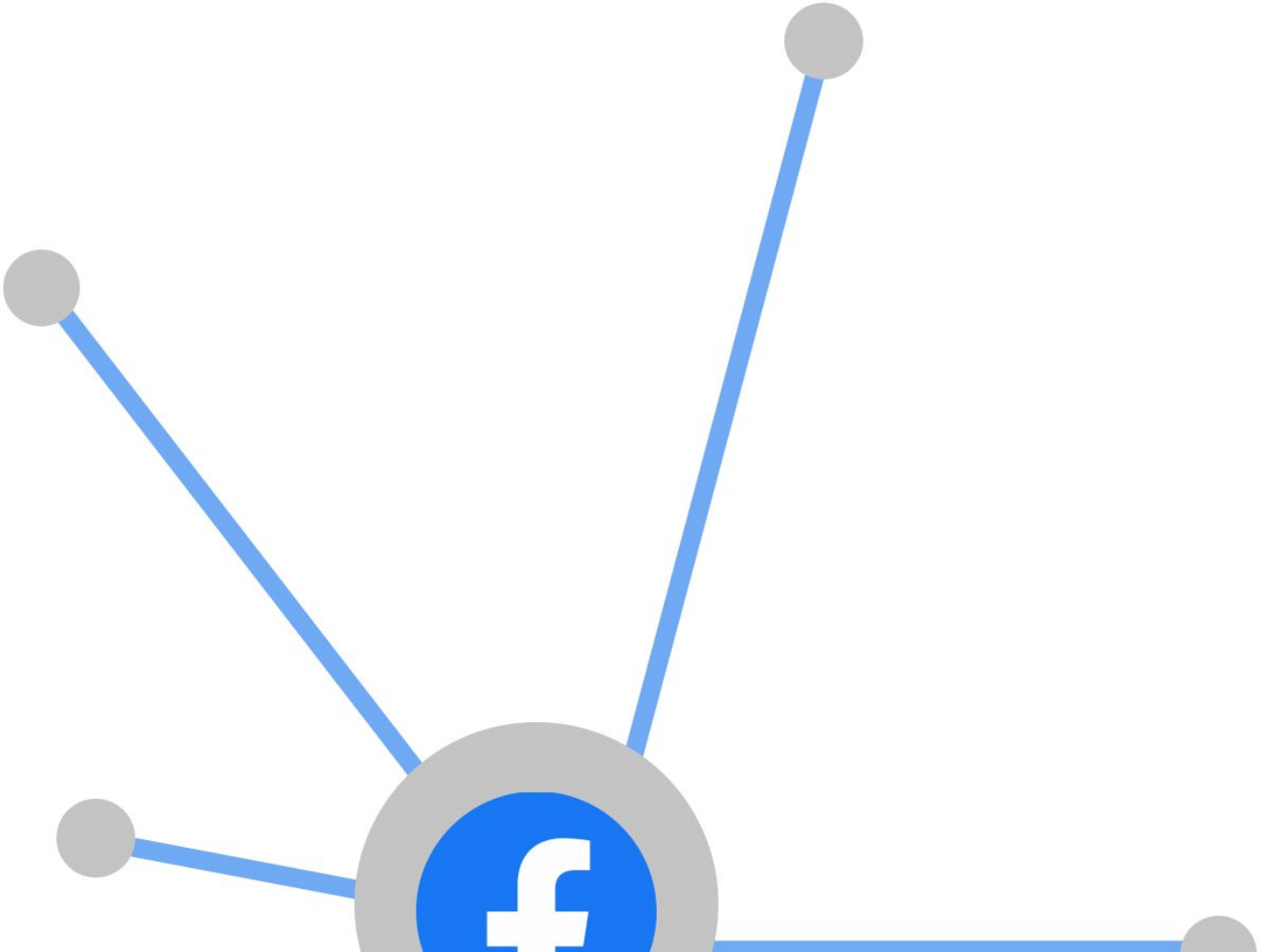
Slides: tiny.cc/atomicdata


CERN
World Wide Web Initiative
EIZO









How did we get here?

- Developers have to invent their own API, because:
- The web is not machine readable
- The web is not read/write

(but it was tho)


WorldWideWeb
Info
Navigate
Document
Edit
Links
Style
Print... p
Page layout...
Windows
Services
Hide h
Quit q
Info
HyperMedia Browser/Editor
Prerelease b of Version 0.13
An excercise in global information availability
by Tim Berners-Lee
Copyright 1990,91, CERN. Distribution restricted: ask for terms. TEST VERSION ONLY
HyperText: Text which is not constrained to be linear.
HyperMedia: Information which is not constrained linear... or to be text.
This version of the WWW application can pick up hypertext infomation from files in a number of formats, from local files, from remote files using NFS or anonymous FTP, from hypertext servers by name or keyword search, and from internet news. Hypertext files may be editied, and links made from hypertext files to other files or any other information.
For more help, use "Help" from the menu. If that doesn't work, then your application has been incompletely installed.
If you have any comments or have bugs, please mail timbl@info.cern.ch. quoting the version number (above).
Navigation
< Previous
Back up
Next >
Home
Help
JFG Home
JFG's home page at InfoDesign
This home page demonstrates some simple concepts of the World-Wide Web infrastructure for global information sharing.
Today, I showed Tim's original Web browser to Marc Weber. I just took some notes about him and linked them to his name on the fly.
I use this World-Wide Web editor exactly like any word processor, with the power to create links from sensitive pieces of text to other places, for instance to personal notes or to any information source in the world.
WWW research stuff
Hypertext Resources (IN-Box)
Geographical list of WWW servers (at C
same list. Example : CERN phone book
WWW project documentation (at CERN
Style
Copy style 1
Apply style 2
Panel...
Links
Mark all A
Mark selection M
Link to marked L
Link to file...
Link to New N
Follow link
Unlink Z
Help
Style editor
Style name: List
<<
style of selection
>>
Apply style to selection
Apply style to all similar text
Find unstyled text
Style sheet
Open
Save as...
Format
First line indent: 100
Successive indent: 130
Font: Helvetica
Point Size: 12
SGML tag: <LI>
Set
Tabs: 130 300
HotWired: What's New
Welcome to Yorb ,
the electr
neighborhood hits
Wired, 24 May. via
in On the R
Flux Ned hears rumors of a deal between GN
Fetish Sola
Net Surf Pro
Jenny Holzer's truisms take a turn for the Wet
If you feel a need for speed ... surf t
What's a week without a Bill Gates j
Corporate recruiting form letters = t
Alt.cynicism ... What's the Point?
free.
Weber -- /Hypertext
Marc Weber
A computer consultant, small publisher and freelance journalist who is currently investigating the early history of the Web, to be published in Wired. He loves HotWired.
Welcome To HotWired!
How to Join:
Register Now
Members Only:
Unguided Tour
Overview
Need Help?
What's New
Your View
© 1995 HotWired Ventures LLC
Powered by Silicon Graphics
JFG


Location: about:
What's New?
What's Cool?
Destinations
Net Search
People
Software
Netscape Navigator™
Version 3.0
Copyright © 1994-1996 Netscape Communications Corporation, All rights reserved.
This software is subject to the license agreement set forth in the license. Please read and agree to all terms before using this software.
Report any problems through the feedback page.
Netscape Communications, Netscape, Netscape Navigator and the Netscape Communications logo are trademarks of Netscape
NETSCAPE

Is the web not machine readable?

- HTML pages are documents, not data
- Tim Berners-Lee wanted to fix this, too...
- ... with the **Semantic web** / RDF / Linked Data

May 17, 2001

The Semantic Web

A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities

By Tim Berners-Lee, James Hendler and Ora Lassila

The entertainment system was belting out the Beatles' "We Can Work It Out" when the phone rang. When Pete answered, his phone turned the sound down by sending a message to all the other *local* devices that had a *volume control*. His sister, Lucy, was on the line from the doctor's office: "Mom needs to see a specialist and then has to have a series of physical therapy sessions. Biweekly or something. I'm going to have my agent set up the appointments." Pete immediately agreed to share the chauffeuring.

BY MIGUEL SALMERON

At the doctor's office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom's *prescribed treatment* from the doctor's agent, looked up several lists of *providers*, and checked for the ones *in-plan* for Mom's insurance within a *20-mile radius* of her *home* and with a *rating* of *excellent* or *very good* on trusted rating services. It then began trying to find a match between available *appointment times* (supplied by the agents of individual providers through their Web sites) and Pete's and Lucy's busy schedules. (The emphasized keywords indicate terms whose semantics, or meaning, were defined for the agent through the Semantic Web.)

In a few minutes the agent presented them with a plan. Pete didn't like it—University Hospital was all the way across town from Mom's place, and he'd be driving back in the middle of rush hour. He set his own agent to redo the search with stricter preferences about *location* and *time*. Lucy's agent, having *complete trust* in Pete's agent in the context of the present task, automatically assisted by supplying access certificates and shortcuts to the data it had already sorted through.

Almost instantly the new plan was presented: a much closer clinic and earlier times—but there were two warning notes. First, Pete would have to reschedule a couple of his *less important* appointments. He checked what they were—not a problem. The other was something about the insurance company's list failing to include this provider under *physical therapists*: "Service type and insurance plan status securely verified

May 17, 2001

The Semantic Web

A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities

By Tim Berners-Lee, James Hendler and Ora Lassila

The entertainment system was belting out the Beatles' "We Can Work It Out" when the phone rang. When Pete answered, his phone turned the sound down by sending a message to all the other *local* devices that had a *volume control*. His sister, Lucy, was on the line from the doctor's office: "Mom needs to see a specialist and then has to have a series of physical therapy sessions. Biweekly or something. I'm going to have my agent set up the appointments." Pete immediately agreed to share the chauffeuring.

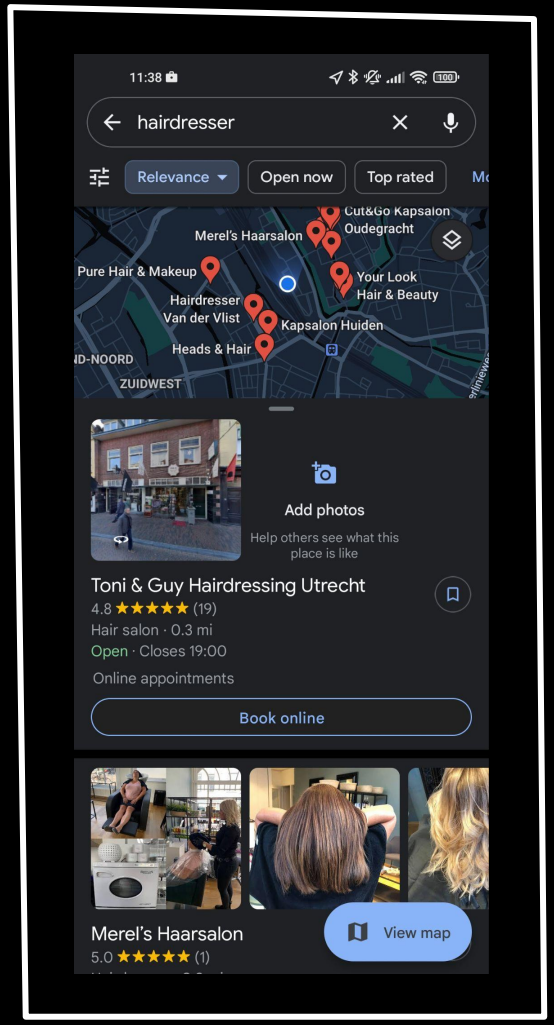
BY MIGUEL SALMERON

At the doctor's office, Lucy instructed her Semantic Web agent through her handheld Web browser. The agent promptly retrieved information about Mom's *prescribed treatment* from the doctor's agent, looked up several lists of *providers*, and checked for the ones *in-plan* for Mom's insurance within a *20-mile radius* of her *home* and with a *rating* of *excellent* or *very good* on trusted rating services. It then began trying to find a match between available *appointment times* (supplied by the agents of individual providers through their Web sites) and Pete's and Lucy's busy schedules. (The emphasized keywords indicate terms whose semantics, or meaning, were defined for the agent through the Semantic Web.)

In a few minutes the agent presented them with a plan. Pete didn't like it—University Hospital was all the way across town from Mom's place, and he'd be driving back in the middle of rush hour. He set his own agent to redo the search with stricter preferences about *location* and *time*. Lucy's agent, having *complete trust* in Pete's agent in the context of the present task, automatically assisted by supplying access certificates and shortcuts to the data it had already sorted through.

Almost instantly the new plan was presented: a much closer clinic and earlier times—but there were two warning notes. First, Pete would have to reschedule a couple of his *less important* appointments. He checked what they were—not a problem. The other was something about the insurance company's list failing to include this provider under *physical therapists*: "Service type and insurance plan status securely verified

HAIRDRESSER



So... we made it?

Not really...

Monopoly, not interoperability

URLs can make data linked and interoperable

- Why don’t we use them for, like, everything?
- URLs should represent **things**, not (HTML) strings!
- So Tim Berners-Lee came up with RDF!

Let’s upgrade a piece of information from sentence to RDF

Sentence

“Joep is born in Baarn on the 20th on january, 1991.”

Not machine readable

Table

Name	City	Birth Date
Joep	Baarn	02-01-1991

1. **Machine readable** (query, sort, filter, serialize)
2. **Still ambiguous**
    a. Who's Joep? Where's Baarn? How to interpret date?

Links

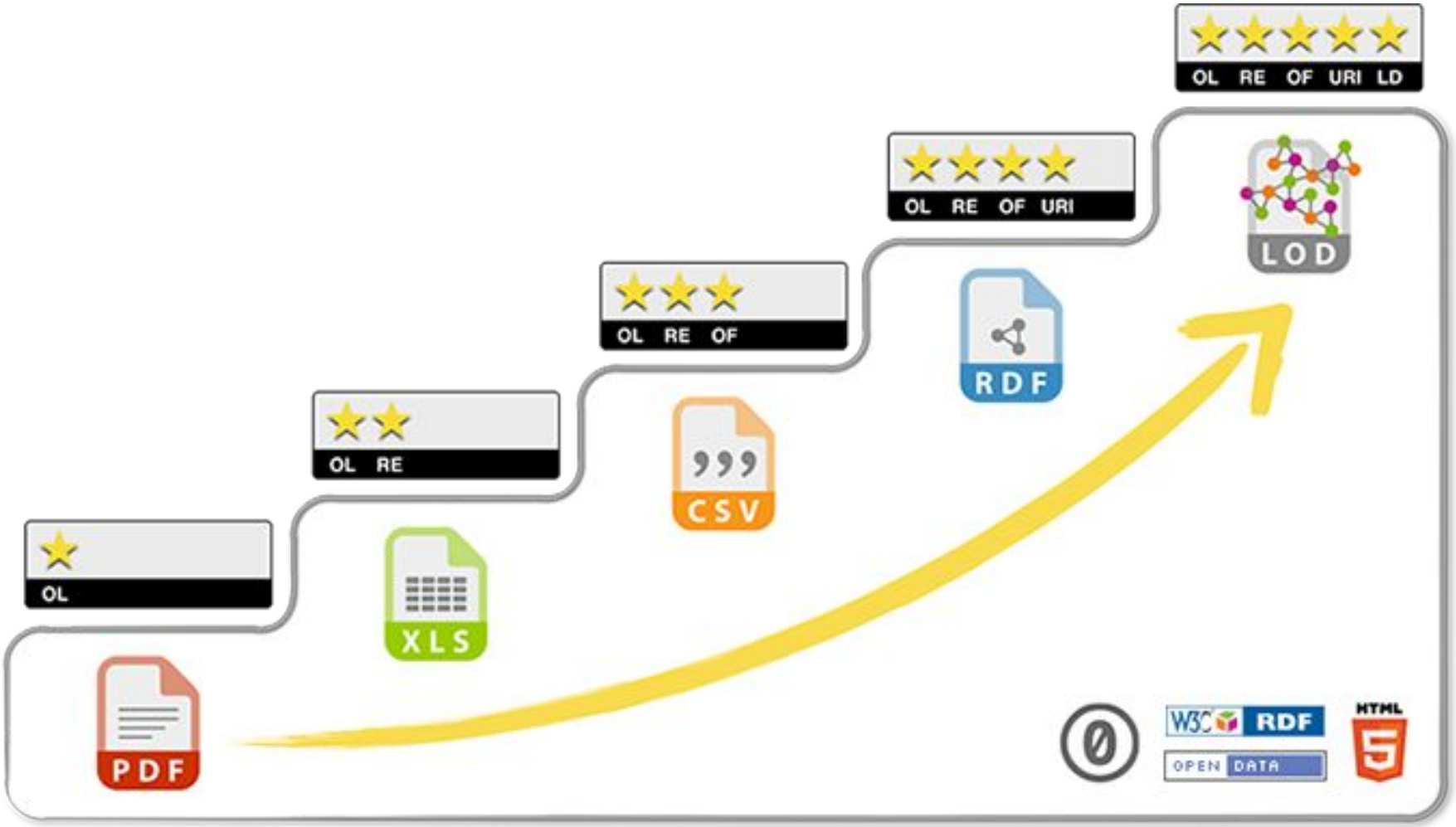
Name	BirthPlace	BirthDate
Joep	Baarn	1991-01-20

1. **Machine readable**
2. **Unambiguous** (even in a global context, because URLs are global)
3. **Browseable** (we can open links and learn more)
4. **Standardized** (predicates can specify how they should be used)

RDF Triples

subject predicate object

```
<https://example.com/joep> <https://schema.org/birthDate> "1991-01-20"xsd:date
<https://example.com/joep> <https://schema.org/birthPlace> <https://example.com/baarn>
```


OL
OL RE
OL RE OF
OL RE OF URI
OL RE OF URI LD
PDF
XLS
CSV
RDF
LOD
0
W3C RDF
OPEN DATA
HTML
5

We (Argu) went all-in on RDF

Because we believe in the merits of the semantic web

Alle functionaliteiten in huis

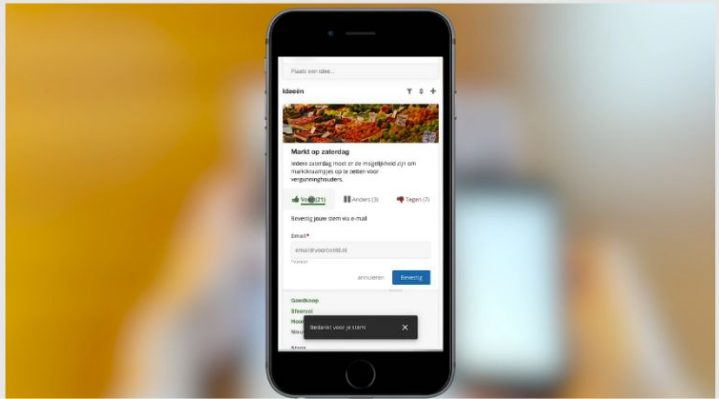
Argu is een krachtig en flexibel platform waarmee je snel online bent. Mis je iets? Dan bouwen wij precies dat wat jij nodig hebt.

Gebruiksvriendelijk

Toegankelijke participatie voor iedereen

- **Eenvoudig registratieproces**
  Gebruikers kunnen alle content zien en hoeven pas te registreren nádat ze ergens op hebben gestemd. Liever een besloten omgeving? Dan kan je mensen uitnodigen via een veilige stemcode.

- Gebruiksvriendelijk platform dat op alle apparaten werkt
- Look and feel aanpassen naar huisstijl
- Voldoet aan WCAG standaarden



Things we had to do

1. Make RDF work with **React (link-redux)**
2. Make RDF work with **Rails (linked_rails)**
3. Create a lot of **ontologies**
4. Invent a **new RDF serialization format** (hextuples)
5. Dynamic **forms** in RDF (powered by SHACL)
6. Full-text **search**

Read more @ ontola.io/blog/full-stack-linked-data/

And the more we built with RDF...

...the less hopeful I became

Why is the cloud not mostly Linked Data?

RDF is not attractive enough for developers

Why is the cloud not mostly Linked Data?

JSON > RDF

What makes RDF less attractive than JSON?

It's not *just* the lack of tools & libraries

1. No native arrays

- Collections, Sequences, Bags... It's confusing!
- Linked lists in RDF are fundamentally complex
- Works different in all serialization formats

Read more @ ontola.io/blog/ordered-data-in-rdf/

2. No subject-predicate uniqueness

- A subject can have *many* triples with the *same* predicate, each with different datatypes
- Hard to store in Maps (and maps are easy and fast!)
- Hard to query => bad Developer Experience
  (compared with JSON: `thing.property`)

3. Blank nodes

- Necessary, but are hard to store & query
- Name collisions
- A high cost for clients to deal with

4. Named graphs

- Add an extra identifier that is required to filter and find a set of triples
- No consensus on best practice, lots of hacky usages

5. It’s hard to select a specific value

```
<example:joep> <schema:birthDate> "1991-01-20"^^xsd:date <example:someNamedGraph>
<example:joep> <schema:birthDate> <example:birthDateObject> <example:someOtherNamedGraph>
<example:joep> <schema:birthDate> "20th of januari 1991"@en <example:someNamedGraph>
<example:joep> <schema:birthDate> "20 januari 1991"@nl <example:someNamedGraph>
<example:joep> <schema:birthDate> "2000-02-30"^^xsd:date <example:someNamedGraph>
```

wish: `joep.birthDate = Date`

reality: `someNamedGraph.joep.birthdate.filter(b.datatype == date)[0]`

6. Lack of consensus on usage

- Many resources (most notably properties) **do not resolve** (data dumps, SPARQL / TPF endpoints, but not as resources!)
- Many **different serialization formats** used - no requirements on which is needed, which leads to fat clients with many parsers
- Highly **fragmented documentation** with a high focus on semantics instead of pragmatics

7. No type safety

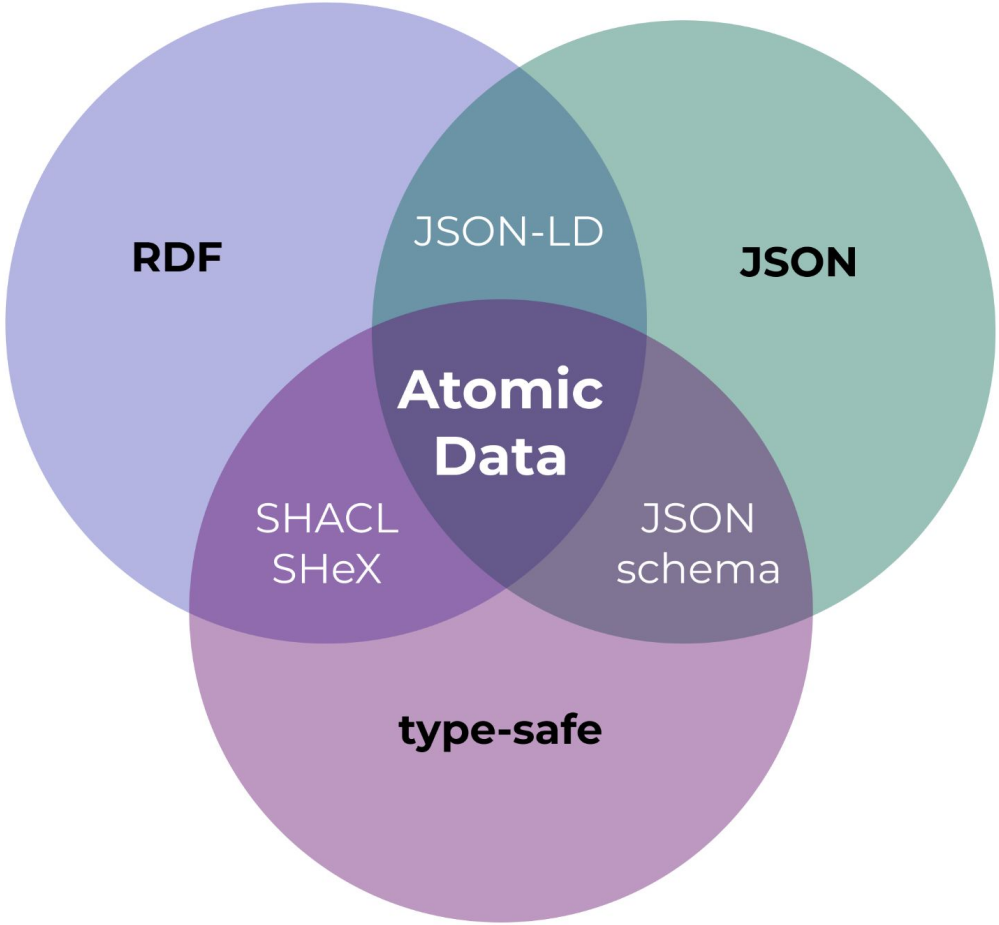
- Any predicate can have **any datatype** (string / bool / etc.)
- SHACL / SHEX are **optional** (and kind of complicated)

What I’d like to have...

- **Decentralized** nature of RDF (lots of URLs)
- **Easy for devs** like JSON
- **Type-safe** like typescript

Take a (very) strict subset of RDF!


Atomic Data



Atomic Data is

- Resolvable RDF
- Idiomatic JSON
- Fully Type-safe

How?

*With strict **Properties**!*

Atomic Properties

```
{
  "@id": "https://atomicdata.dev/properties/description",
  "https://atomicdata.dev/properties/datatype": "https://atomicdata.dev/datatypes/markdown",
  "https://atomicdata.dev/properties/description": "A textual description of something.",
  "https://atomicdata.dev/properties/isA": [
    "https://atomicdata.dev/classes/Property"
  ],
  "https://atomicdata.dev/properties/shortname": "description"
}
```

- **JSON-AD** as serialization
- **@id** must resolve to the actual JSON-AD resource
- Every **key** is a **Property**
- Properties have **datatypes**, triples can't set their own!
- **Shortnames** can be used for easy mapping (to simple JSON)
- **Descriptions** can be used for semantic meaning
- See docs.atomicdata.dev

Atomic Classes

- E.g. **BlogPost** or **Agent**
- Describe which Properties are *required* and *optional*
- Can be used to **validate data** and **show Forms**
- Difference from RDF: schema tightly coupled to Class

Atomic Data Core

JSON-AD, Properties, Datatypes, Classes

Atomic Data Extended

Authorization, authentication, collections, more

Static **data is relatively easy**

But data *changes over time*, how can we deal with that?

What can we gain by standardizing *Changes*?

- **Versioning**, history, global undo
- **Verifiability** for checking authorship, auditing
- **P2P** sharing
- So basically **Git**, but for structured data
- A **read-write web**! Collaboration everywhere

Atomic Commits

- **Commits** (changes) **are resources**, too, with their own URLs
- **PK Cryptography** is used for signing every single change
- So every change becomes **reversible** and **verifiable**
- And can therefore be shared **peer to peer**

Atomic Agents

- **Agents are users** with their own URLs
- Agents have a **public key** used for signing Commits
- They are also used for read / write **authorization**
- Authentication by **signing** every HTTP GET request

And More

- Endpoints (machine readable API docs)
- Collections (queries for sorting, filtering, pages)
- Invitations (easy sign up + redirect)
- Websockets (real time collaboration)
- File upload / download

A spec needs an implementation

- **Atomic-server** (graph database, written in Rust)
  - Fast: <1ms responses
  - 14mb binary, no runtime dependencies
  - Full-text search, indexed queries, sorting, filtering, pagination
  - Authorization, authentication, attachments, versioning
- **Atomic-data-browser** (GUI for viewing / editing)
  - Browse data, dynamic forms, tables, search, endpoints, collaborative documents

Time for a demo!

Follow along at atomicdata.dev

Getting started

- @tomic/lib for js, typescript, node projects
- @tomic/react for web GUIs (codesandbox template)
- atomic_lib (rust library, powers server)
- Everything mentioned is MIT licensed, so fully free & open source
- Check out the Discord and Docs!

Questions?

Joep Meindertsma, Declarative Amsterdam 2022-11-08

Slides: tiny.cc/atomicdata