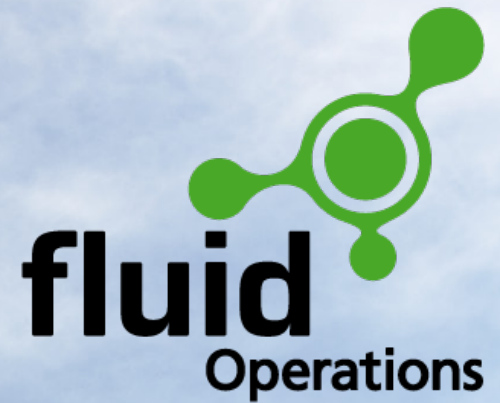


**Developing Semantic Applications with
the Information Workbench –
Aspects of Ontology Engineering**

Peter Haase

Ontology Summit
March 20, 2014



Information Workbench

Linked Data and Semantic Technologies in the Enterprise

Semantics- & **Linked Data-based integration** of private and public data sources based on data providers

- Generic and specific providers for various data formats and sources
- Supports established mapping frameworks (e.g. R2RML, SILK, ...)
- Named graphs for managing contexts and provenance

Intelligent **Data Access and Analytics**

- Flexible self-service UI
- Visualization, exploration, dashboarding and reporting
- Semantic search

Collaboration and actionable data

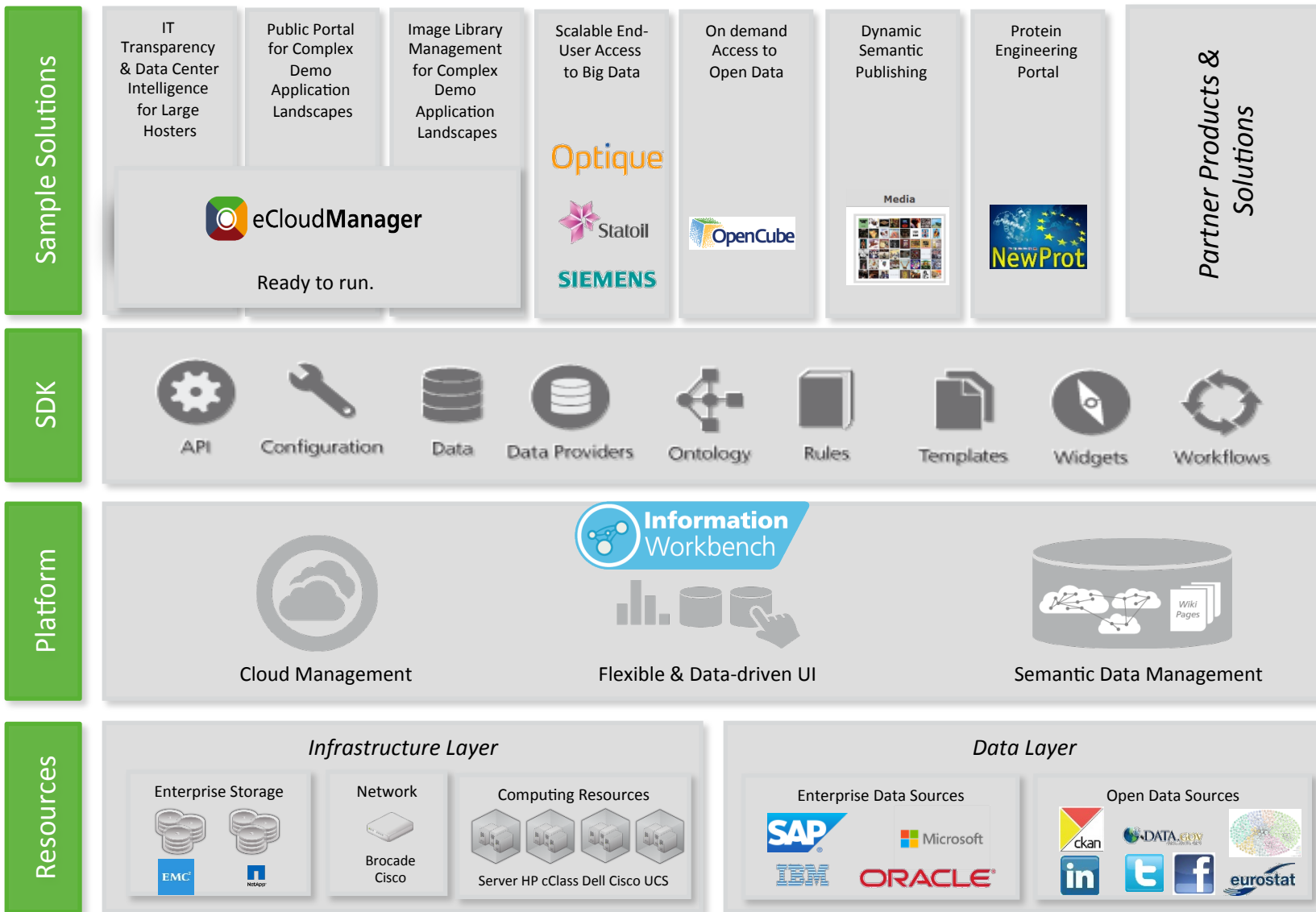
- Curation & authoring
- Trigger actions and invoke services
- Collaborative and data-driven workflows

Open standards and technologies

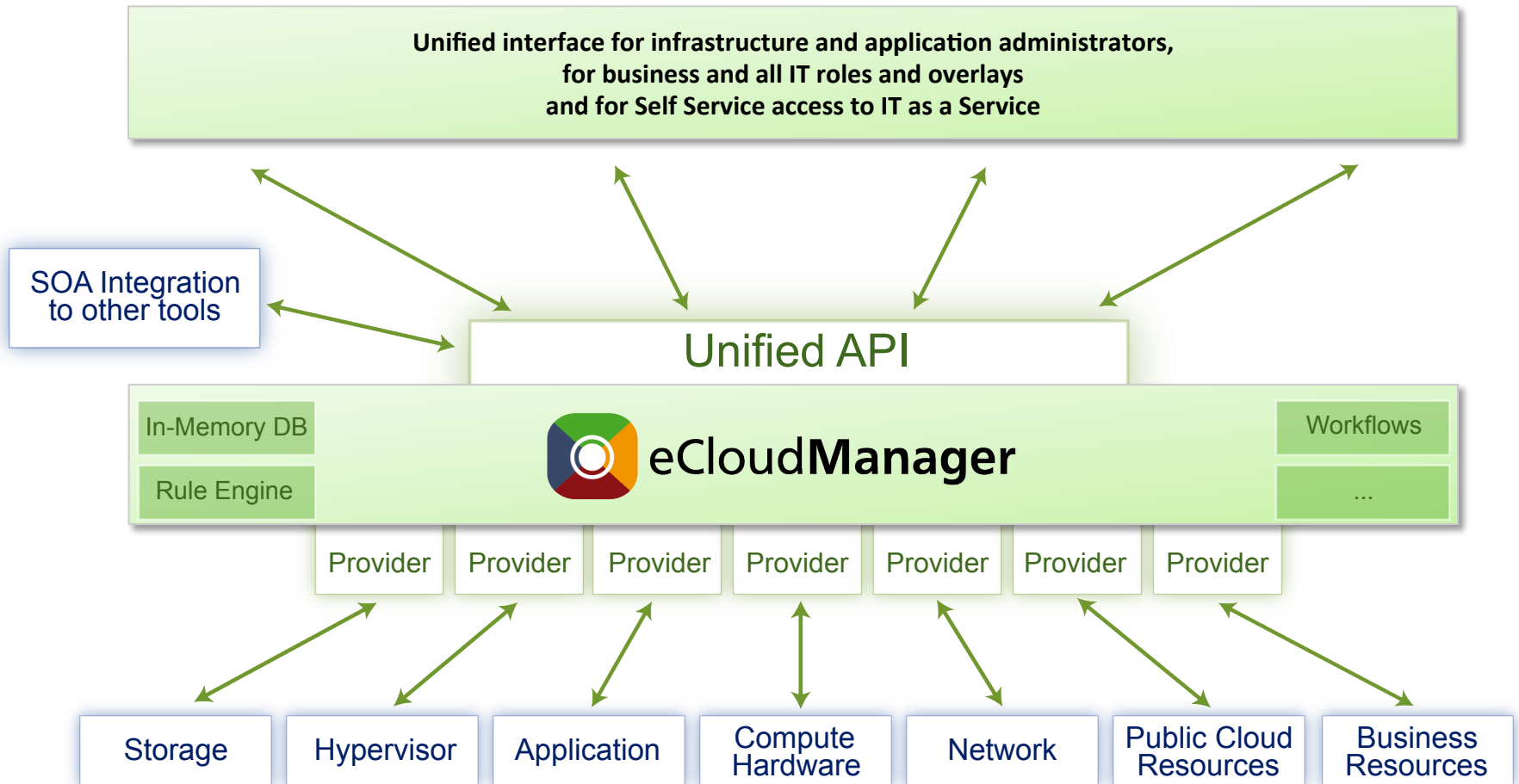
- Semantic Wiki based frontend (Using SMW Syntax)
- Supporting W3C standards (OWL, RDF, SPARQL,, ...)
- Community Edition (Open Source) + Enterprise Edition (Commercial)



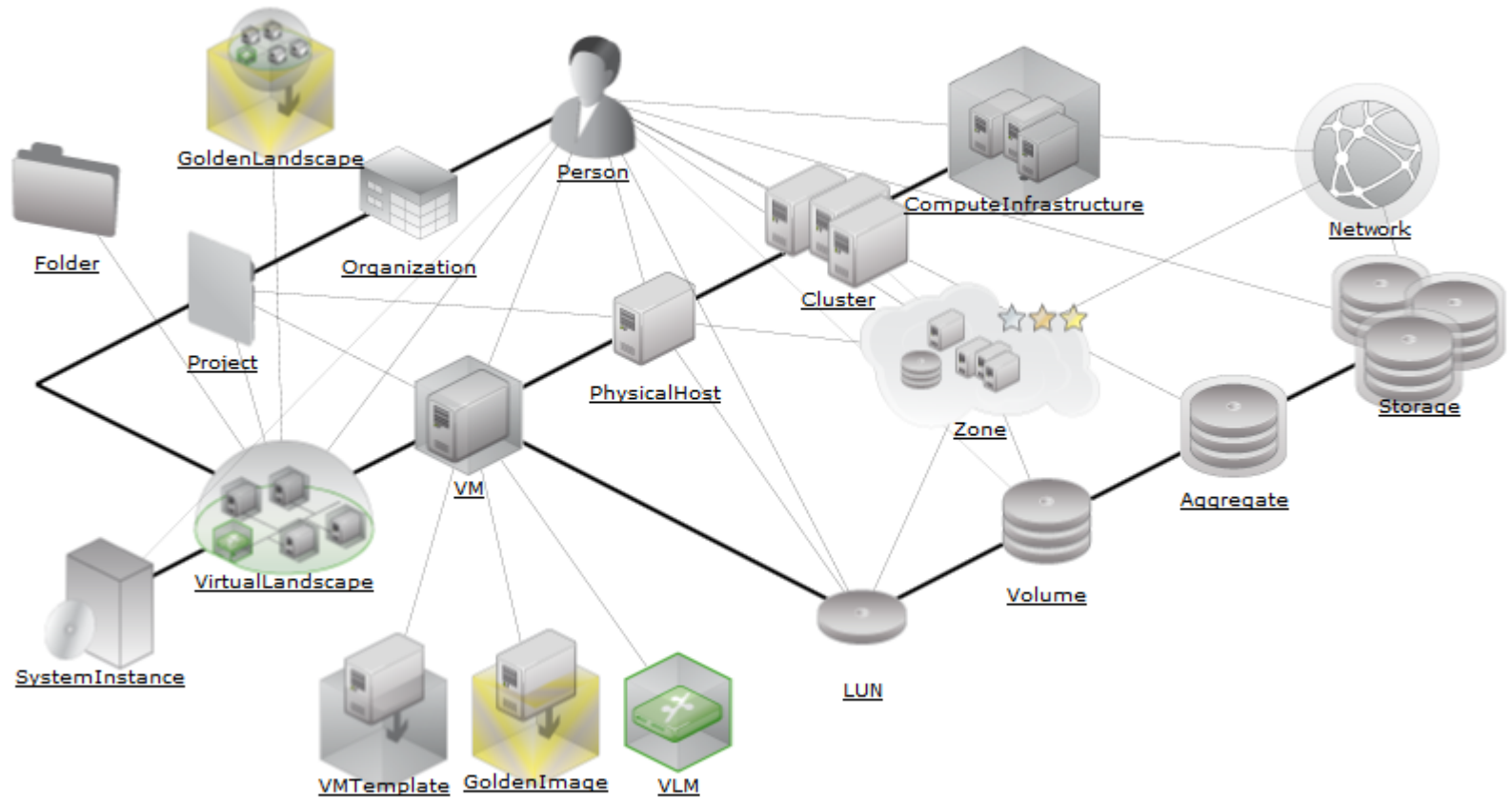
fluidOps Platform and Solutions



Example: eCloudManager Solution



Enterprise Cloud Ontology



<http://fluidops.com/ontologies/ecloud>



Widget-based User Interface

- Declarative specification of the UI based on available pool of widgets and simple wiki-based syntax
- Widgets have direct access to the database
- Embedding of dynamic data, visualizations, forms, etc.
- Ontology-driven template mechanism

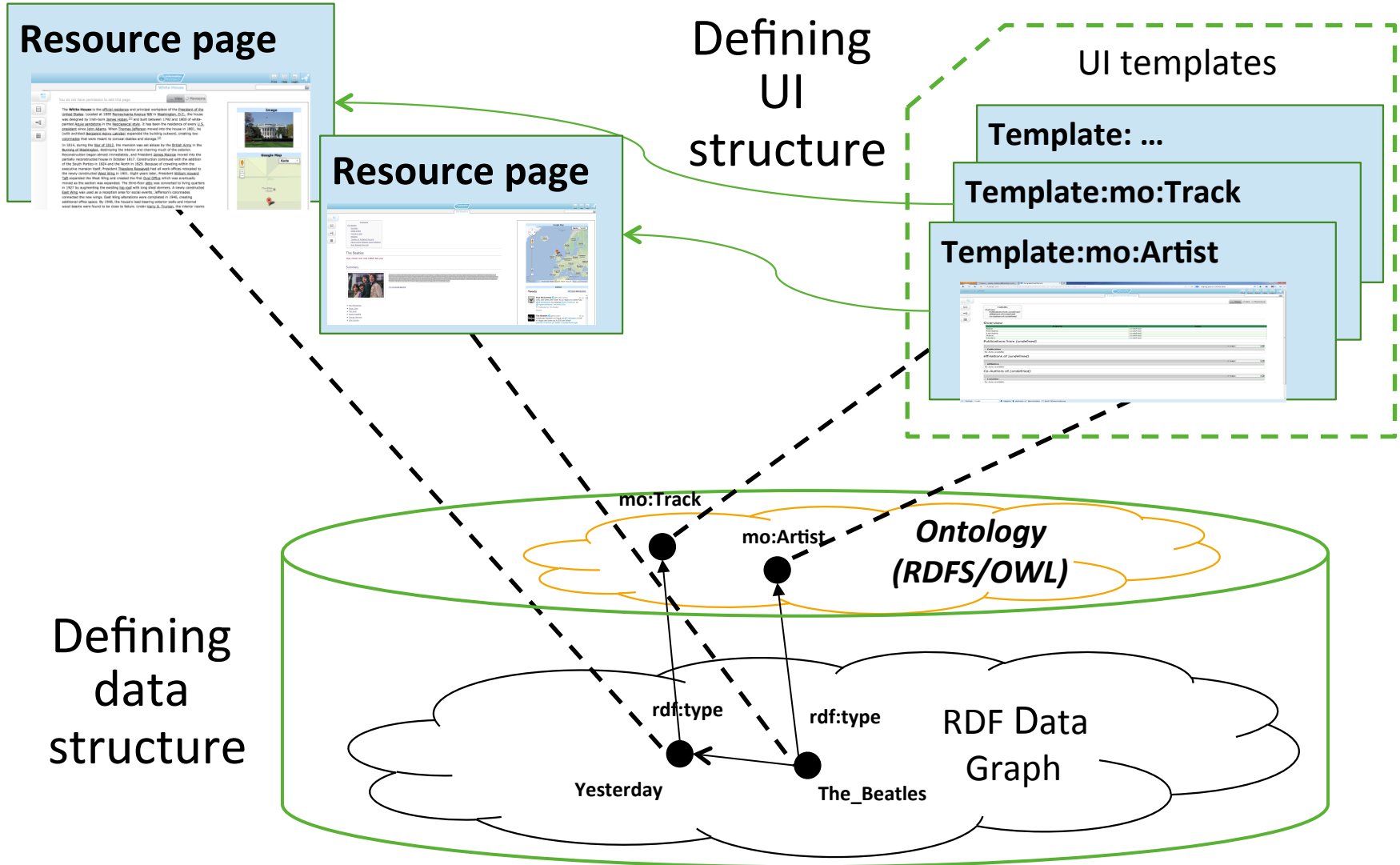
The screenshot shows a web browser window with the eCloudManager interface. The page is in edit mode for a resource named 'Cluster41'. The main content area contains a wiki-style declarative template for a widget. The template includes a title, a description, and a chart widget. The chart widget is configured to display 'CPU Usage' data for the cluster over time, with a specific query and settings defined. The interface includes a sidebar with navigation icons and a top navigation bar with user information and system status.



The screenshot shows the same web browser window, but now displaying the rendered result page for 'Cluster41'. The page features a header with the resource name and a description. Below the header, there is a 'Responsible Person' section with a form for entering a name. The 'CPU Usage' section contains a line chart showing the average CPU usage of all hosts in the cluster over time, with a legend for 'on', 'off', and 'suspended' states. The 'Power State of Hosts' section contains a pie chart showing the distribution of hosts in different power states. The interface includes a sidebar with navigation icons and a top navigation bar with user information and system status.



Ontology as a “Structural Backbone”

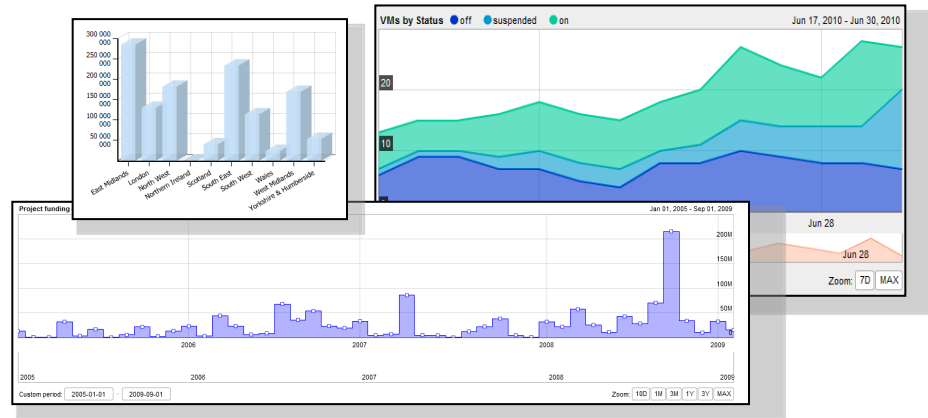


Widget-Based User Interface

Visualization and Exploration

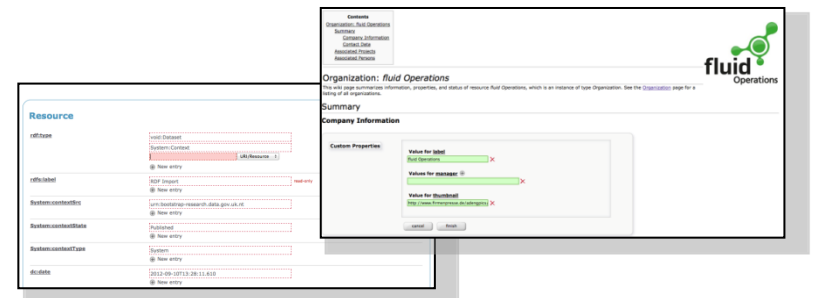


Analytics and Reporting



Authoring and Content Creation

Mashups with Social Media

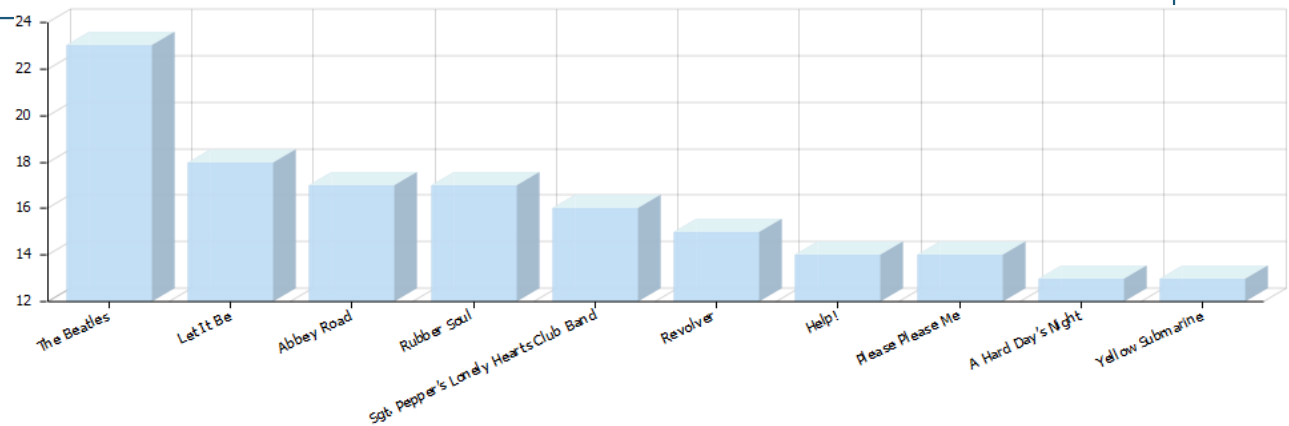


Widgets are not static and can be integrated into the UI using a Wiki-style syntax.



Example: Add Widgets to Wiki

```
{{#widget: BarChart |
query ='SELECT distinct (COUNT(?Release) AS ?COUNT) ?label WHERE {
  ?? foaf:made ?Release .
  ?Release rdf:type mo:Release .
  ?Release dc:title ?label .
}
GROUP BY ?label
ORDER BY DESC(?COUNT)
LIMIT 10
'
| input = 'label'
| output = 'COUNT'
}}
```



**SPARQL
queries**

Example: Show top 10 released records for an artist



Example (continued)

Page of a class instance:

- Displays the data about the resource according to the class template

The screenshot shows a web interface for 'The Beatles' on an 'Information Workbench'. The page is divided into several sections:

- Contents:** A list of links including 'The Beatles', 'Summary', 'Similar artists', 'Youtube video', 'Releases', 'Timeline of Published Records', 'Places where Releases were Published', and 'Most Released Records'.
- The Beatles:** A section with the tags 'classic rock, rock, british, 60s, pop'.
- Summary:** A section featuring a photograph of the Beatles and a text block: 'The Beatles were an iconic rock group from Liverpool, England. They are frequently cited as the most commercially successful and critically acclaimed band in modern history, with innovative music a cultural impact that helped define the 1960s and an enormous influence on music that is still felt today. Currently, The Beatles are one of the two musical acts to sell more than 1 billion records, with only Elvis Presley having been able to achieve the same feat. Read more about The Beatles on Last.fm.'
- Twitter:** A section showing tweets from Paul McCartney (@PaulMcCartney) and The Beatles (@thebeatles). Paul McCartney's tweet says: 'Only one week left! Enter for a chance to meet Paul @sfoutsidlands by helping #SaveTheArctic on @UrgencyNetwork: ow.ly/mcGOW'. The Beatles' tweet says: 'Celebrate Summer of Cirque at @TheBeatles LOVE in Vegas and save up to \$50 per ticket! cirk.me/14TEm3e pic.twitter.com/9EBPM4VUqM'.

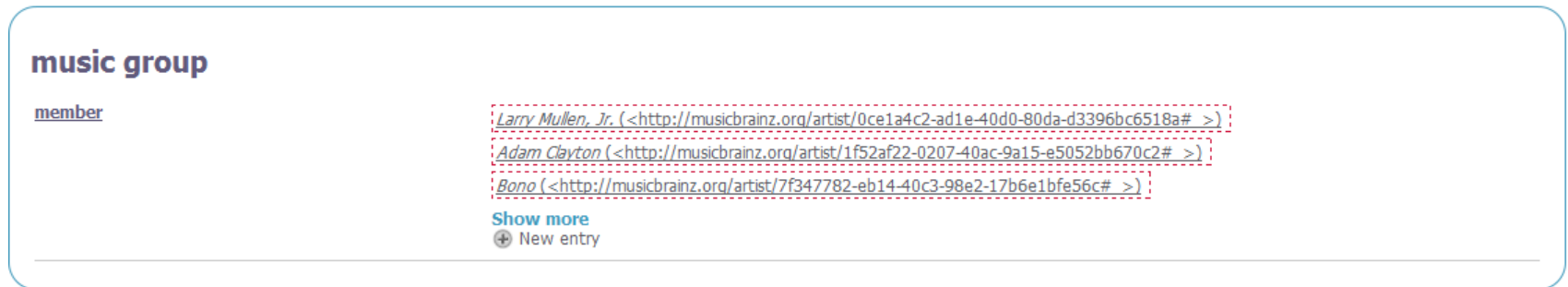
On the right side of the page, there is a Google Map of Europe with a red pin on the United Kingdom. The map includes labels for various countries and regions such as Iceland, Norway, Sweden, Finland, Denmark, Poland, Germany, France, Spain, Italy, Greece, Romania, and Ukraine.



Ontology-Based Data Input

Triple Editor takes into account the ontology definition:

Autosuggestion tool considers the domains and ranges of the properties



music group

member

[<http://musicbrainz.org/artist/0ce1a4c2-ad1e-40d0-80da-d3396bc6518a# >](http://musicbrainz.org/artist/0ce1a4c2-ad1e-40d0-80da-d3396bc6518a#)

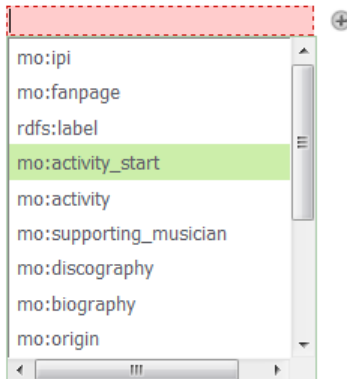
[<http://musicbrainz.org/artist/1f52af22-0207-40ac-9a15-e5052bb670c2# >](http://musicbrainz.org/artist/1f52af22-0207-40ac-9a15-e5052bb670c2#)

[<http://musicbrainz.org/artist/7f347782-eb14-40c3-98e2-17b6e1bfe56c# >](http://musicbrainz.org/artist/7f347782-eb14-40c3-98e2-17b6e1bfe56c#)

[Show more](#)

[+ New entry](#)

Add new property:



- mo:ipi
- mo:fanpage
- rdfs:label
- mo:activity_start
- mo:activity
- mo:supporting_musician
- mo:discography
- mo:biography
- mo:origin

Delete All Data

Example: properties available for the class *mo:MusicGroup* are suggested automatically



Ontology Repository and Ontology Editor

Logged in as **admin**

Information Workbench

Print Query Admin Help Logout

owl:Ontology

View Edit Revisions

Ontologies in the repository

Filter

Ontology
core
Exif data description vocabulary
time
http://data.press.net/ontology/classification/
The Timeline ontology
The RDF Schema vocabulary (RDFS)
Geonames ontology
asset
http://www.bbc.co.uk/ontologies/news/
The Event ontology
event.n3
Friend of a Friend (FOAF) vocabulary
http://www.bbc.co.uk/ontologies/tag/
http://www.bbc.co.uk/ontologies/sport/
http://www.bbc.co.uk/ontologies/system/
http://www.bbc.co.uk/ontologies/event/
domain



Ontology Repository and Ontology Editor

The screenshot displays the Information Workbench interface for editing an ontology. The main window shows the 'Recurring Competition' class with its URI and annotations. A left sidebar lists various ontology elements, and a bottom sidebar shows 'Object Properties'.

Contents
<http://www.bbc.co.uk/ontologies/sport/>
Classes
Object Properties
Datatype Properties
Individuals

http://www.bbc.co.uk/ontologies/sport/
To export the ontology or open it in external tools (such as Protege for editing) use this [Export Ontology](#)

Classes

- class
- brand
- Competition
A sports concept which can be applied to an asset
- Recurring Competition
- Competitive sporting organisation
- Sporting organisation
- Sports Discipline
- DivisionalCompetition
- Event Gender
- Group competition
- Multi round competition
- Knock out competition
- League competition
- Match
- Unit competition
- Composite competition
- Person
- Agent
- Round
- Sport governing body
- A world cup 2010 concept which can be applied to an asset
- Competitive Sporting group
- Competition Type
- Squad role
- Competes for role
- medal Competition
- Round Type
- session

Recurring Competition

URI: <http://www.bbc.co.uk/ontologies/sport/RecurringCompetition>

Annotations

label	Recurring Competition
comment	A recurring sports competition.

Taxonomy

This form allows editing of the taxonomic relations of this class.

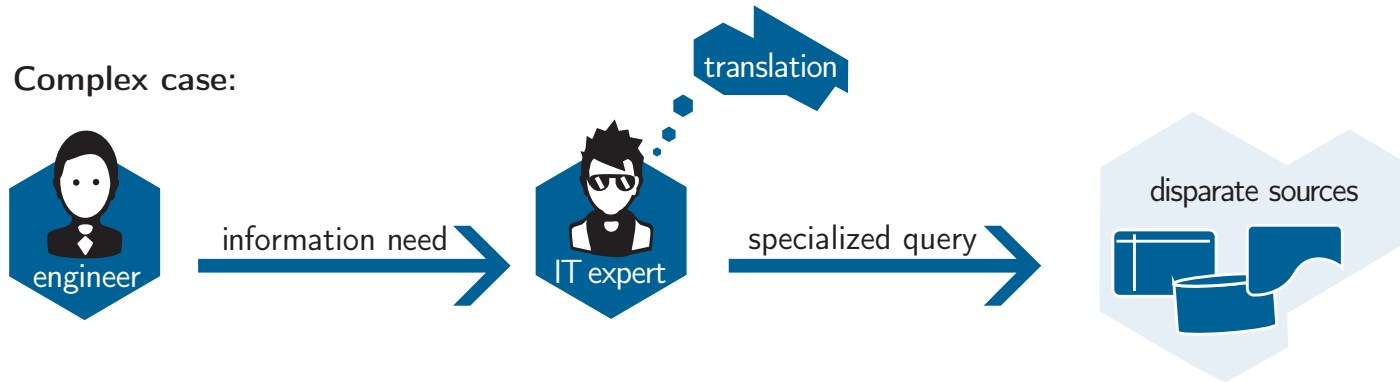
subClassOf	Recurring Event
owl:equivalentClass	A sports concept which can be applied to an asset
owl:disjointWith	

Properties with domain Recurring Competition

property
No data available

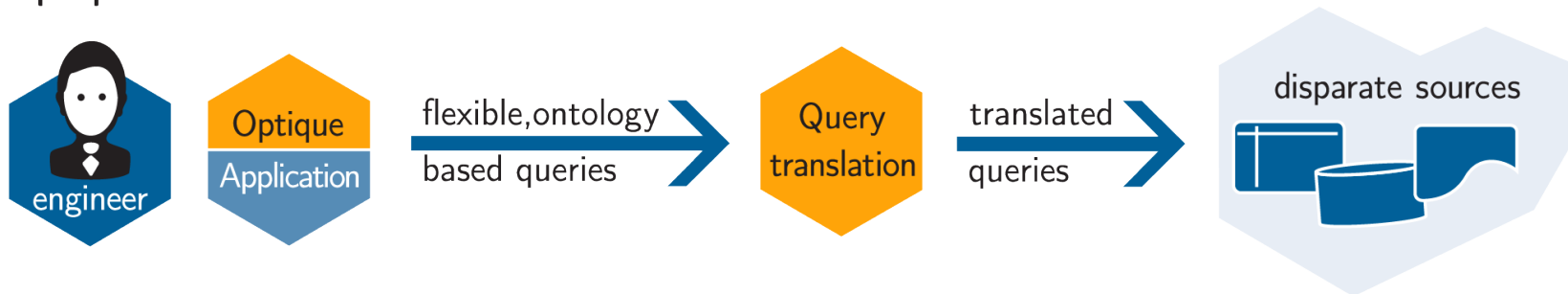


Ontology Based Data Access



Up to 80% of expert's time spent on data access

Optique solution



Visual Query Formulation

The screenshot displays a visual query formulation interface for a 'Field' dataset. The main workspace shows a query graph with a central node 'Field remainingOilEqu... (o)' connected to 'Company' and 'Wellbore' nodes. A toolbar includes 'Delete Node', 'Same Node', 'Undo', 'Redo', 'New Query', 'Save Query', 'Stored Queries', 'SPARQL Query', and 'Run Query'. Below are two panels: 'Field' with a search bar and a list of entities (Wellbore: 895, ProductionLicence: 27, Company: 42), and 'Field Information' with a search bar, a slider for 'remainingOilEquivalents' (0 to 1058.1), and text input fields for 'descriptionDevelopment' and 'descriptionTransport'.

- Find:
- fields together with their remaining oil
 - that are currently operated by Statoil, and
 - show the types of wellbores located on this fields



How Do We Develop Ontologies?

1) Manual engineering inside fluidOps

- Significant expertise in house (several PhDs in semantic technologies)
- Hard to train newbies, regular software engineers, let alone “end users”

2) Reuse of existing ontologies

- Good ontologies are highly valuable assets, typically “for free”
- Building blocks for reusable solutions
- Interoperability with existing data

3) Development by partners and customers

- Typically as extensions to our base ontologies
- To extend the platform for additional use cases

4) Ontologies already in place

- Defined / selected by the customer



“Pragmatic” Ontology Engineering

Very incremental, iterative approach

- Often data first, schema last
- Typically: simple class hierarchies, properties with domain/range restrictions
- Agile development: Intertwined ontology and application development
- Immediate effect of enriching / extending the ontology

Methodology

- Rather informal, elements of NeOn methodology
- Naming patterns (namespaces, URIs, naming schemes)
- Type hierarchies
- Basic design patterns (e.g. N-ary)

Tooling

- Everything from plain text editor to Protégé
- Simple built-in ontology editing in the Information Workbench



Difficulties in Reuse

Hard to motivate people to reuse

- Tradeoff between specificity for the immediate problem at hand vs. reuse and interoperability
- Software engineers tend to have preference for “their own” solutions
- Also related to initial barrier of understanding an “alien” ontology

What is a good ontology to reuse?

- Reuse per se is not a benefit
- Hard to assess quality, relevance/adoption by a non expert
- Repositories such as Linked Open Vocabularies are a good step
- Ranking, recommendations would be helpful



Some Examples of Ontology Reuse and Reusable Apps

- FOAF
- SKOS

- Music ontology
- CIDOC-CRM
- Semantic Web Conference Ontology
- SIOC
- Sports Ontology

- SEON (Software Evolution ONtologies)
- OSLC (Open Services for Lifecycle Collaboration)

- ...



Conclusions

Ontologies as structural backbone of the application

- For data integration
- As conceptual model for data access and interaction (visualization, querying, ...)
- For defining associated actions (events, triggers, actionable results)

Main advantages

- Ontologies as reusable, modular artifacts - enabling reusable, modular apps
- Ease of extensions, not bound to a fixed schema defined-upfront
- Flexible application development, very simple customization to particular domains
- Try it: <http://www.fluidops.com/information-workbench/>





Thank you!

Contact

fluid Operations AG
Altrottstr. 31
Walldorf
Germany

+49 (0) 6227 358087-0
www.fluidops.com
contact@fluidOps.com