

# The ComparaGRID Project: Integrating Genomic Mapping Data using Semantic Web Technologies

Trevor Paterson & Andy Law

The Roslin Institute (EBRC) Edinburgh Scotland

*ComparaGRID Consortium*



[www.comparagrid.org](http://www.comparagrid.org)

## ComparaGRID Consortium Members:

Genomics: (*Farm animal, crop, microbial*)

Bioinformatics

Computer Sciences

Ontologists

Statisticians

## Developers:

Tony Burdett (EBI)

Robert Davey (JI)

Andrew Gibson (MU)

Trevor Paterson (RI)

Matthew Pocock (NU)



JOHN INNES CENTRE



The University of Manchester

## Biological Aims:

- **To integrate disparate genomic data resources** (genomic mapping, DNA sequence, evolutionary relationships, functional information) across species boundaries.
- In order to inform and expedite genomic mapping: particularly in non-model organisms.
- To map, identify and understand genes behind phenotypes (e.g. diseases & commercially important traits)

## Computer Science Aims:

- To develop a domain neutral architecture for semantic integration and query of disparate datasources.

# COMPARATIVE GENOMICS: EXEMPLAR BIOLOGICAL USE CASE

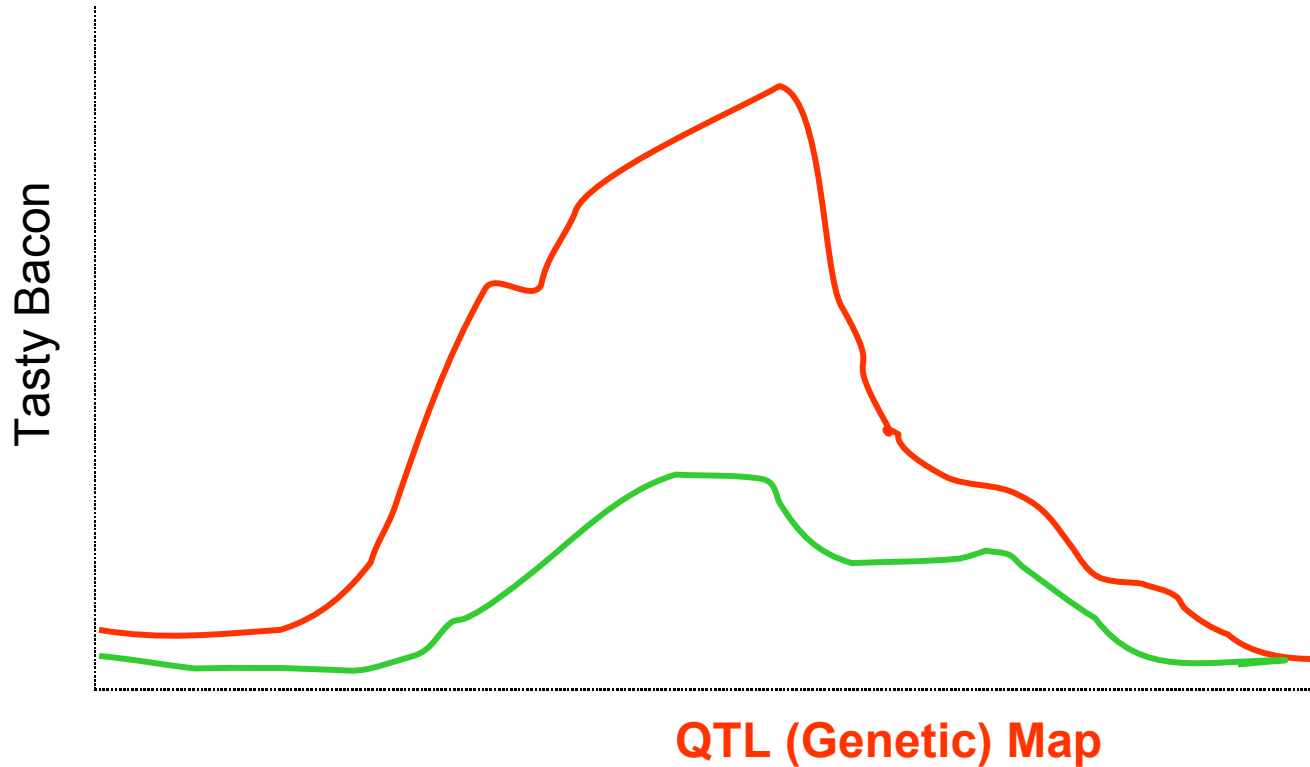
Agribusiness wants to map the underlying genetic basis of the 'Tasty Bacon' Trait ( a QTL ).



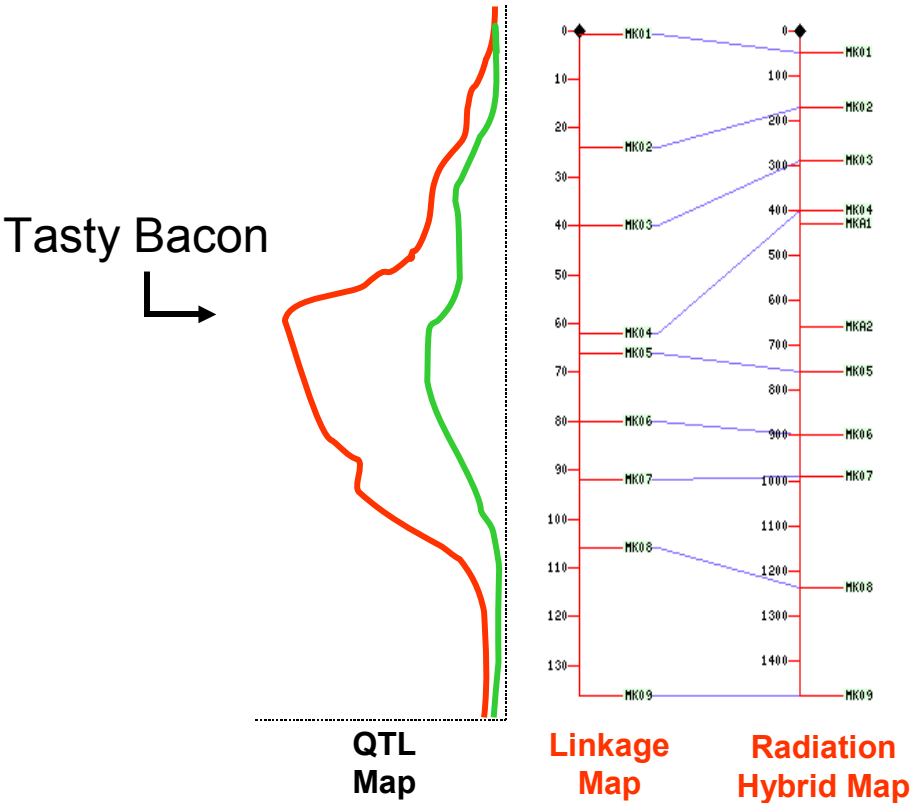
How can we expedite our understanding of this phenotype – using all the available information resources from pig and other (better characterised) species?

The 'Tasty Bacon' QTL has been genetically mapped in **PIGS**.

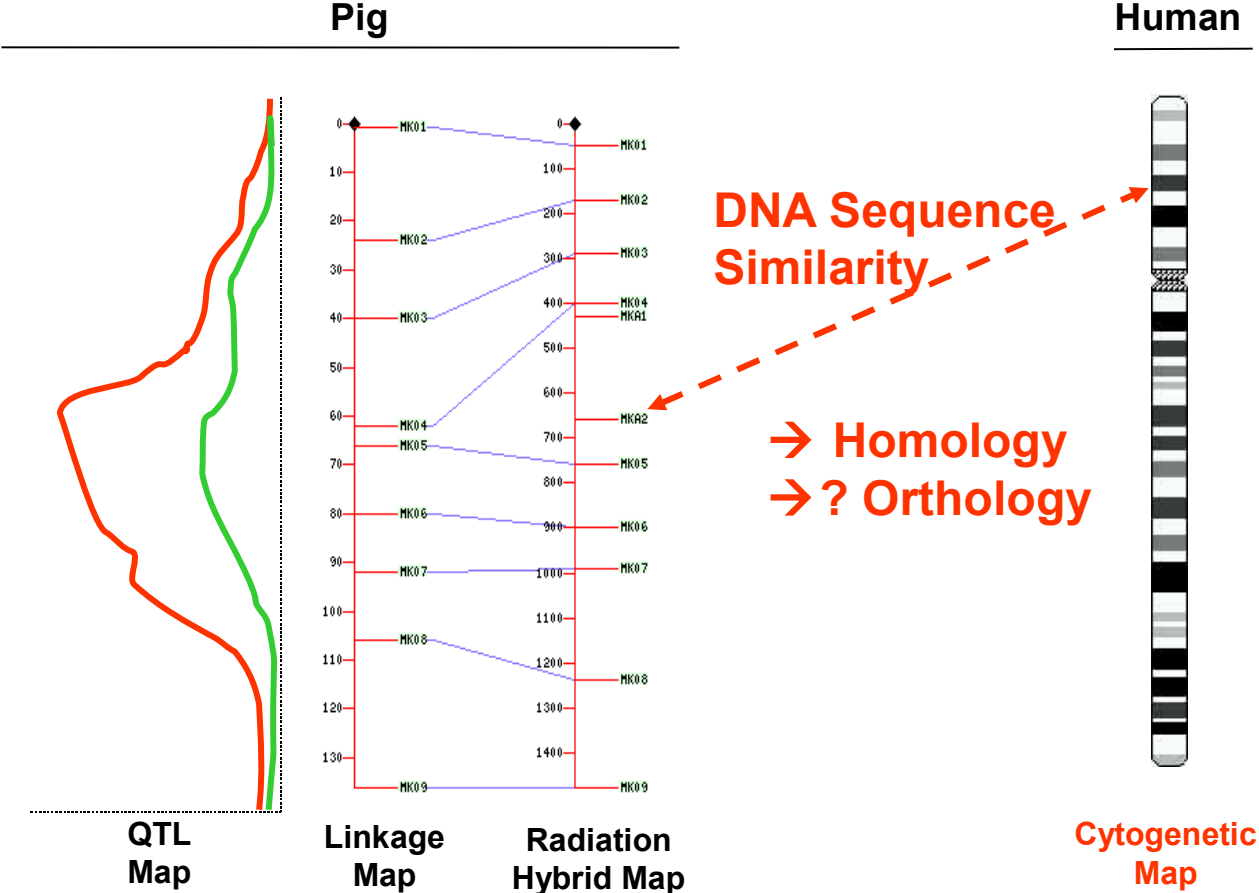
i.e. There is some genetic factor(s) contributing to 'Tastiness'.



The position of the QTL is correlated on various types of **PIG** Genetic maps



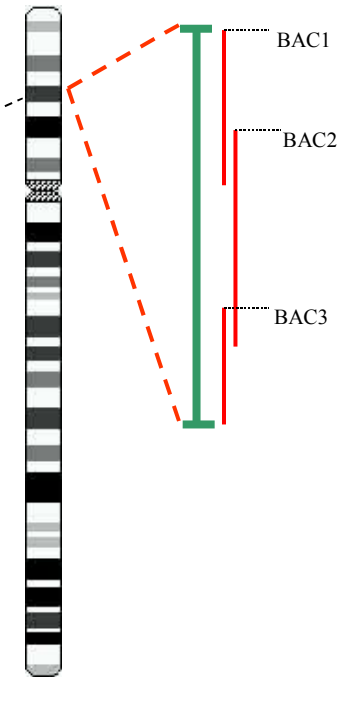
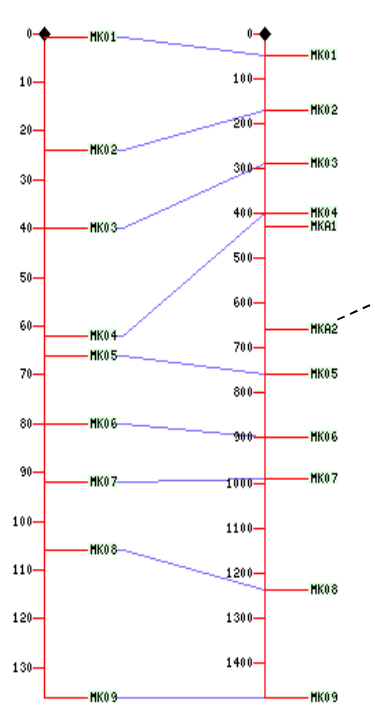
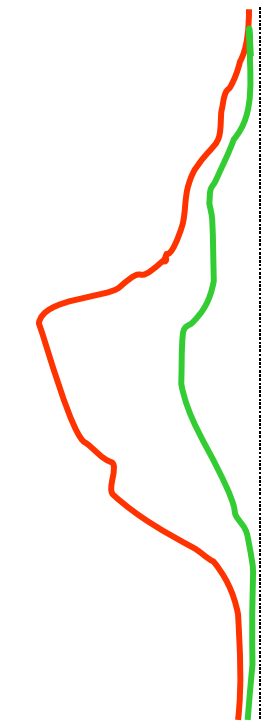
There is a 'known' homology between a Pig Marker/Sequence in this region and the **HUMAN** genome



# A Physical Map of BAC clones exists for this region of the **HUMAN** Genome

**Pig**

**Human**



**QTL  
Map**

**Linkage  
Map**

**Radiation  
Hybrid Map**

**Cytogenetic  
Map**

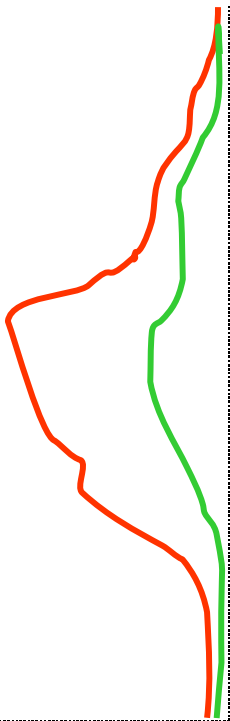
**Physical  
Mapping**

There are known **CHICKEN** expressed sequences homologous to Human Gene Sequences in this region

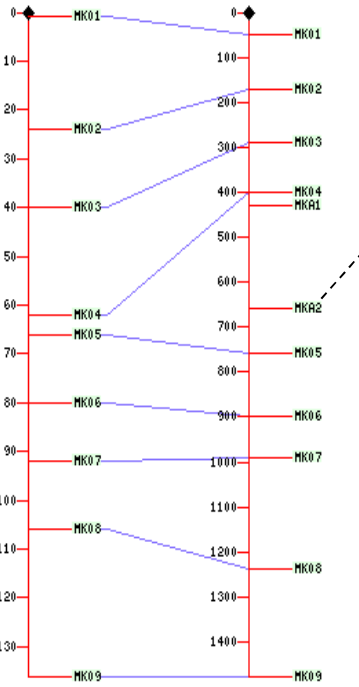
**Pig**

**Human**

**Chicken**



**QTL Map**

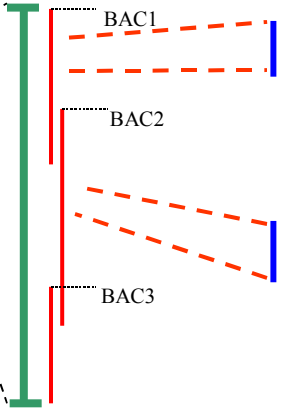


**Linkage Map**

**Radiation Hybrid Map**



**Cytogenetic Map**



**Physical Mapping**

**EST1**

**EST2**

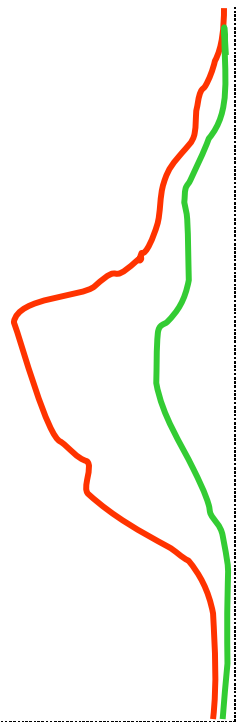
**EST Library**

Gene expression Data for these **CHICKEN** ESTs might correlate with a trait similar to 'Tastiness'

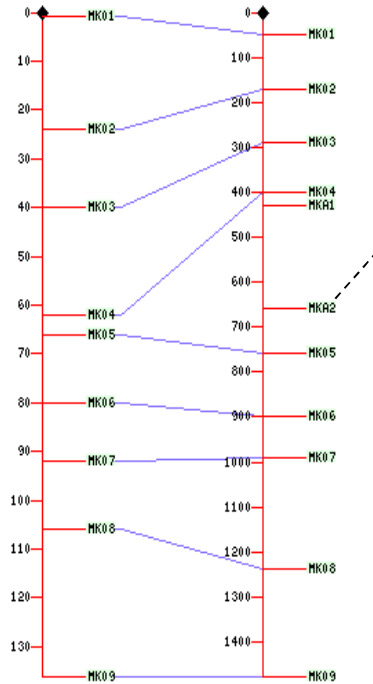
**Pig**

**Human**

**Chicken**



**QTL Map**

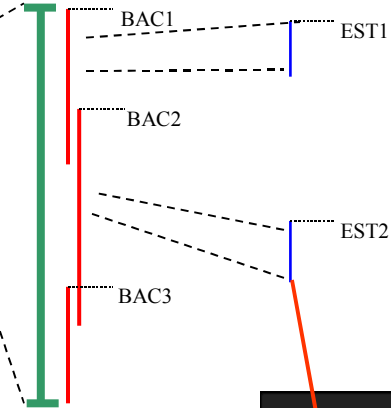


**Linkage Map**

**Radiation Hybrid Map**



**Cytogenetic Map**

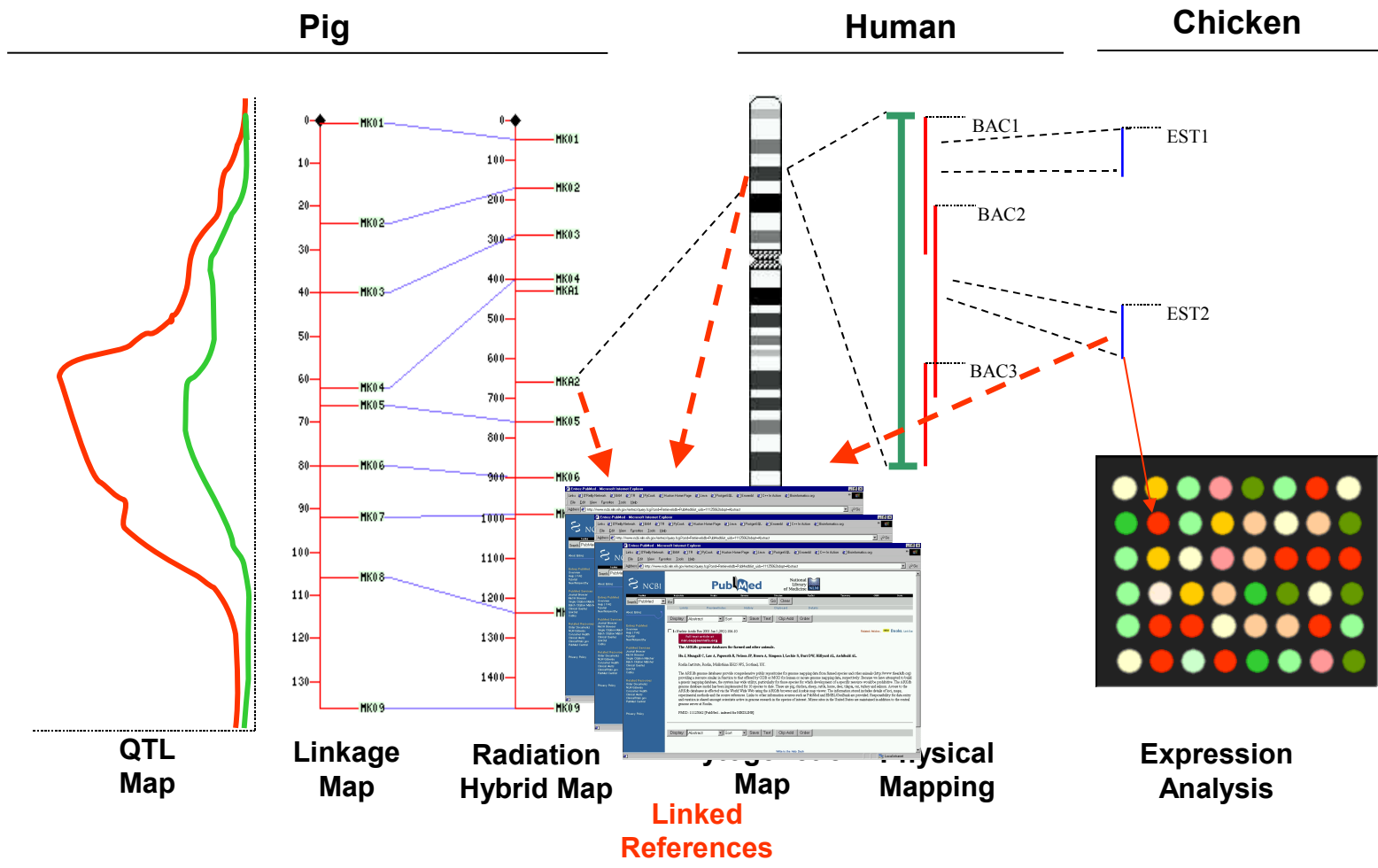


**Physical Mapping**



**Expression Analysis**

The literature, and other datasources, may detail functions of Human genes in this region, and homologies to genes in other species – helping the researcher predict candidate genes in Pigs responsible for tastiness



## Challenges:

How can we discover all this data ?

How can we integrate all this data (capturing its meaning) ?

How can we use this information to discover 'new' information, make testable predictions etc?

## Approach:

Use Semantic Web Technologies to query, retrieve and represent the data ontologically.

Use Formal Logic Reasoners to integrate and reason over the data:

If we have a set of facts –can the reasoner deduce anything else that must or might be true.

# UNDERLYING BIOLOGICAL PRINCIPAL BEHIND CROSS-SPECIES MAP COMPARISON

## Conservation of Synteny:

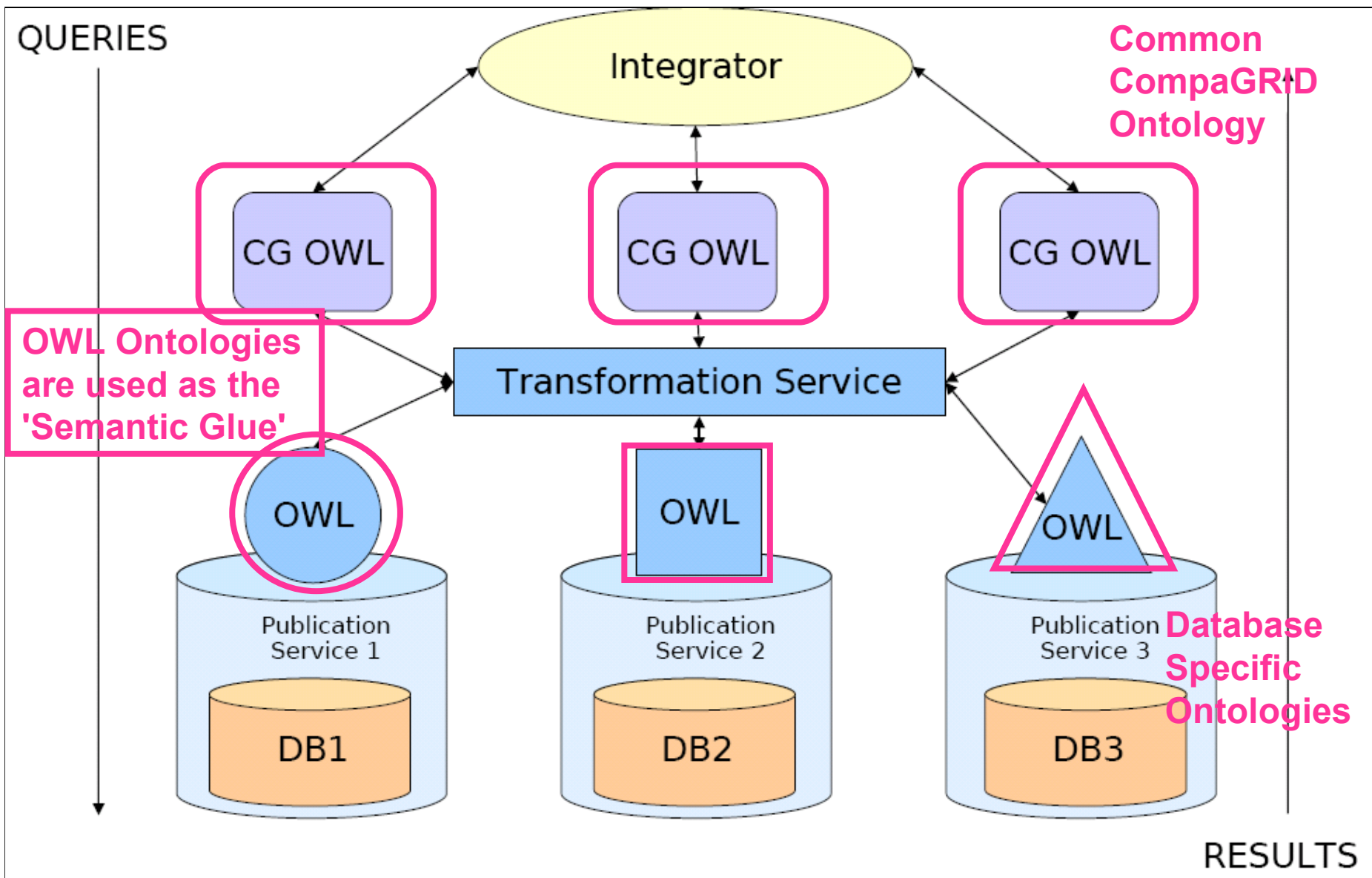
“Conservation of (blocks of) gene order throughout chromosomal evolution”

As species evolve and diverge, their chromosomes rearrange through duplications, inversions, translocations etc - but blocks of genes can be traced through evolutionary history between even relatively divergent species (e.g. chicken and man).

***Therefore the known gene order in these blocks in one species can inform/predict the order of evolutionarily related genes (orthologues) in other species.***

Therefore we might be able to predict functionally significant genes in the region of the Pig QTL from the information available in other species.

# THE COMPARAGRID ARCHITECTURE



# *Semantic Integration through Ontologies: I*

## **Ontologies in Biology**

- simple controlled vocabularies with simple subsumption classifications
- well established use of for labelling data in the genome projects (as tags and metadata)
- used for data description, retrieval and comparison
- 'informal' – not used for reasoning or 'integration'

*e.g.*

**Sequence Ontology:** label structural parts of sequences

**Gene Ontology:** label function and localisation of genes

**Medical and Anatomical Ontologies**

# Semantic Integration through Ontologies: II

## OWL-DL (1.1) Ontology Language (W3C standard)

**Formal** Knowledge Representation Syntax based on **Description Logics** (decidable First Order Logics), therefore providing sound and complete **semantic reasoning**

OWL 1.0 *SHOIN(D)*

OWL 1.1 *SHROIQ(D)* (cardinality restrictions can be qualified  
more expressivity on Roles)

Set of Axioms (or facts) representing Sets of Individuals (Classes) and (Data and Object) Property Assertions that constrain Classes and Individuals

Alternate Serialisations (RDF, OWL-XML, JAVA-APIs)

## Emergent Tools for Semantic Web Applications

JAVA API – CO-ODE/WonderWeb

Authoring Tools: Protege4

**Inference Engines:** Pellet and FaCT++

# *Semantic Integration through Ontologies*

## **The ComparaGRID Ontology**

### **Domain Ontology (DO)** for semantic integration

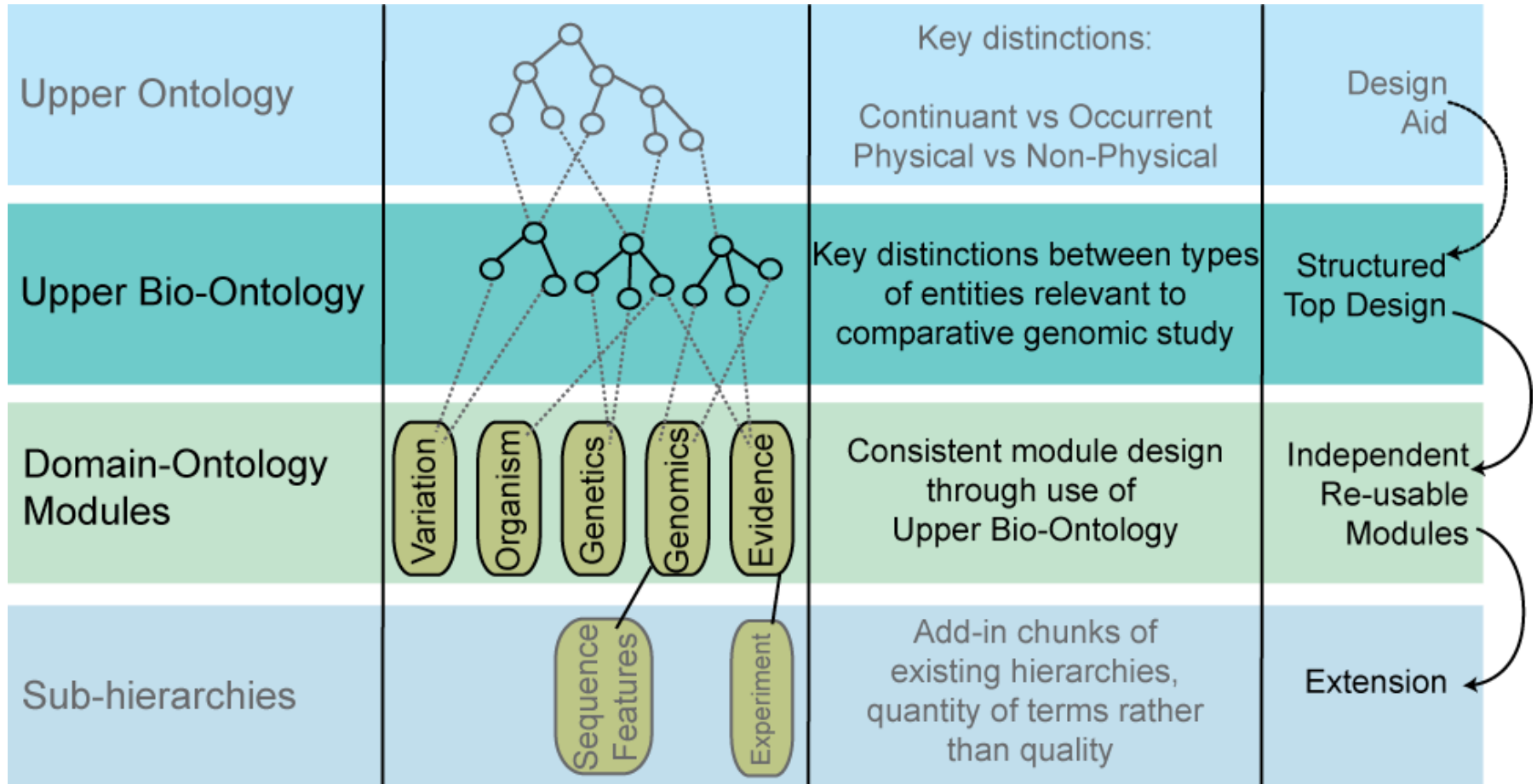
- captures the conceptual model of '**Genomic Information**'
- OWL-DL (1.1) semantics facilitate data-integration tasks
- (Pellet) reasoning over values and constraints allows computation and discovery of novel information

### **Application Ontology (AO)** derived from DO

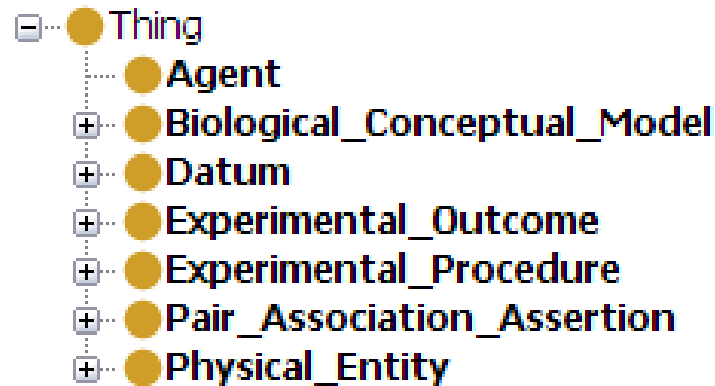
- subset/extension of DO
- additional concepts for target application  
(e.g. **database specific information**)
- annotations provide application metadata

e.g. Identifiers, Xrefs, Datasources.....

# THE MODULARISED COMPARAGRID ONTOLOGY

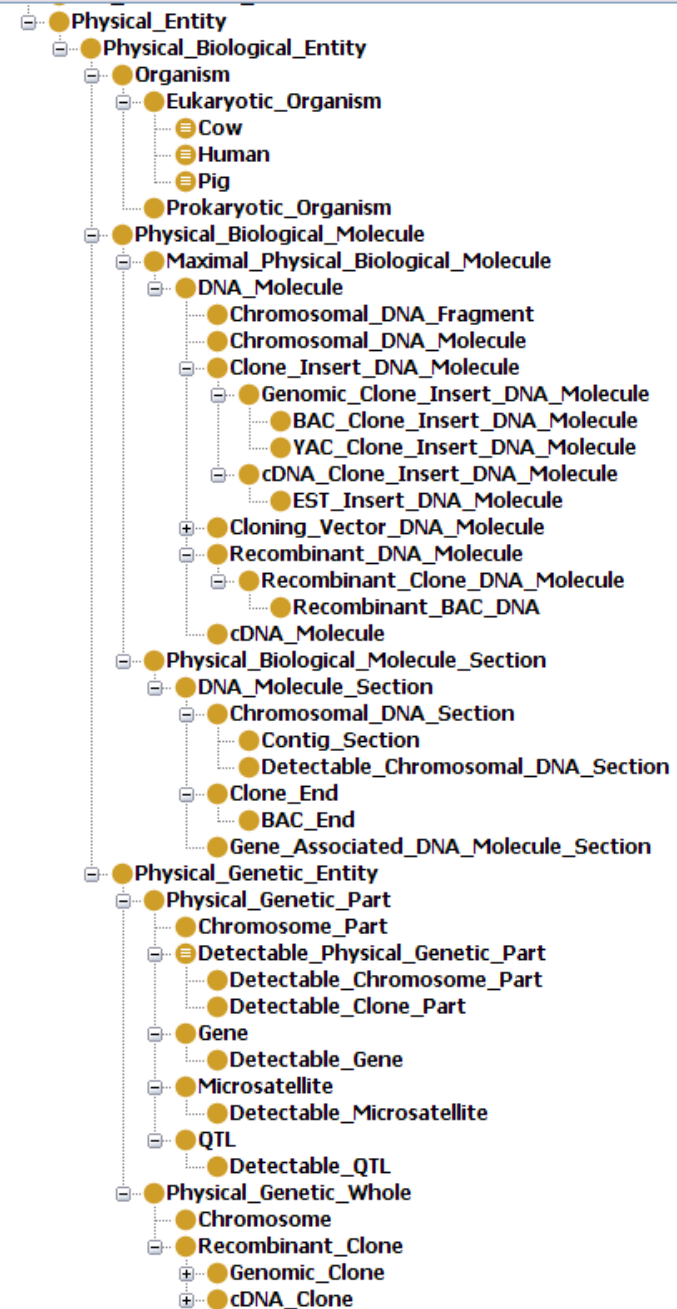


# ComparaGRID 'Application' Ontology

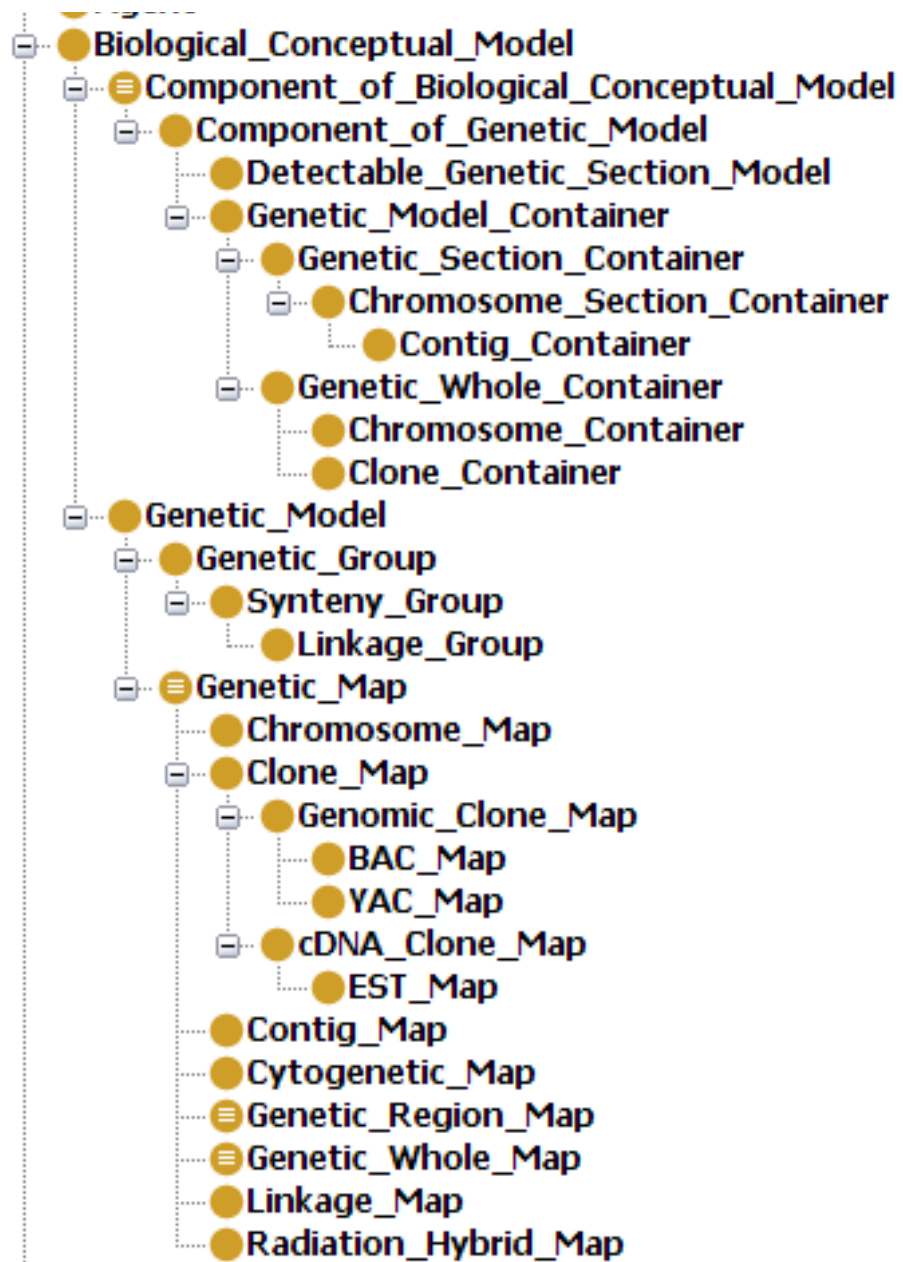


- There are '**Physical Things**' – from the real world
- There are '**Conceptual Models**' of Physical Things – genetic maps etc
- There are '**Representations**' of Physical Things - DNA Sequences

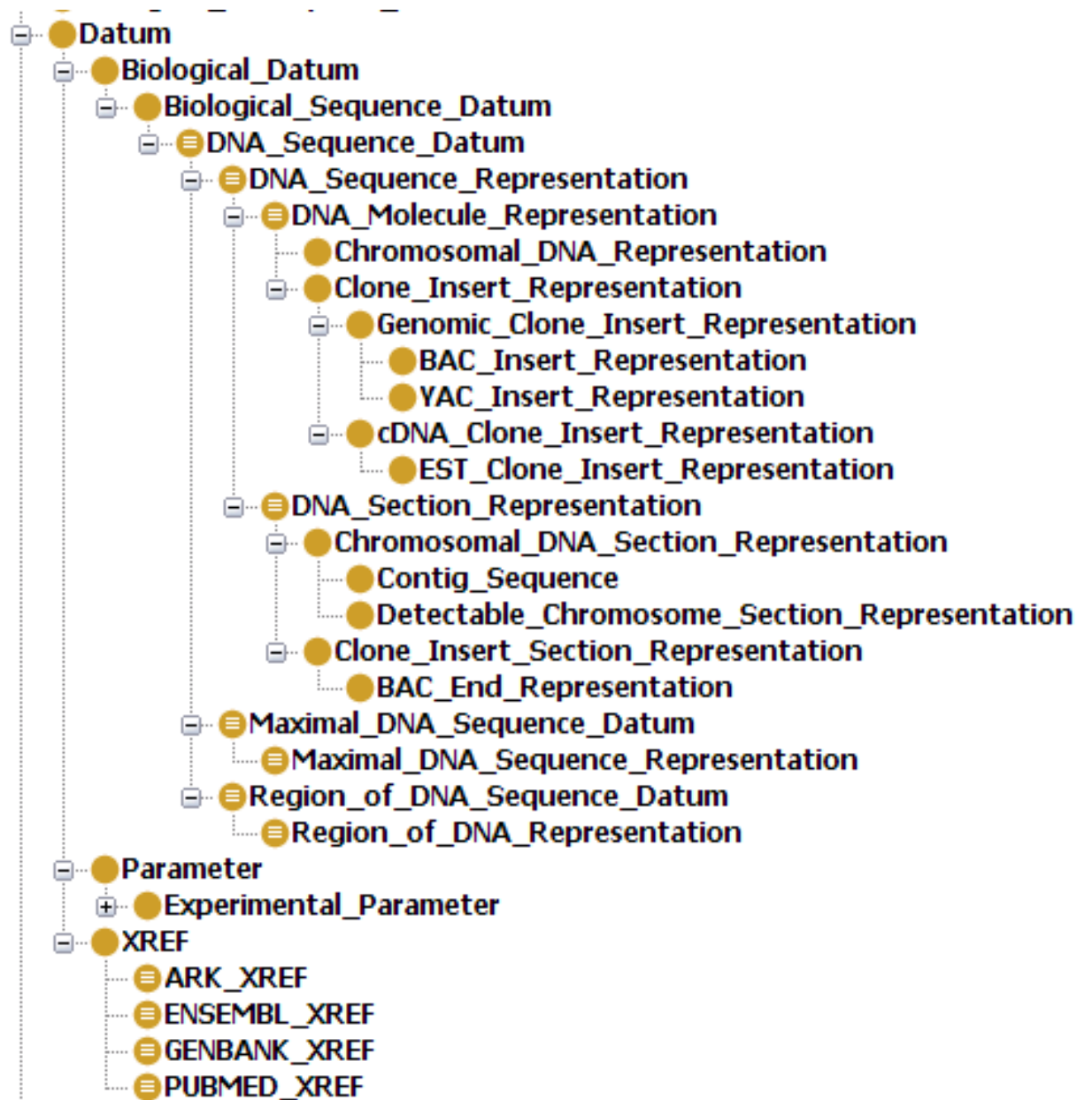
- 'Physical Things'
  - from the real world



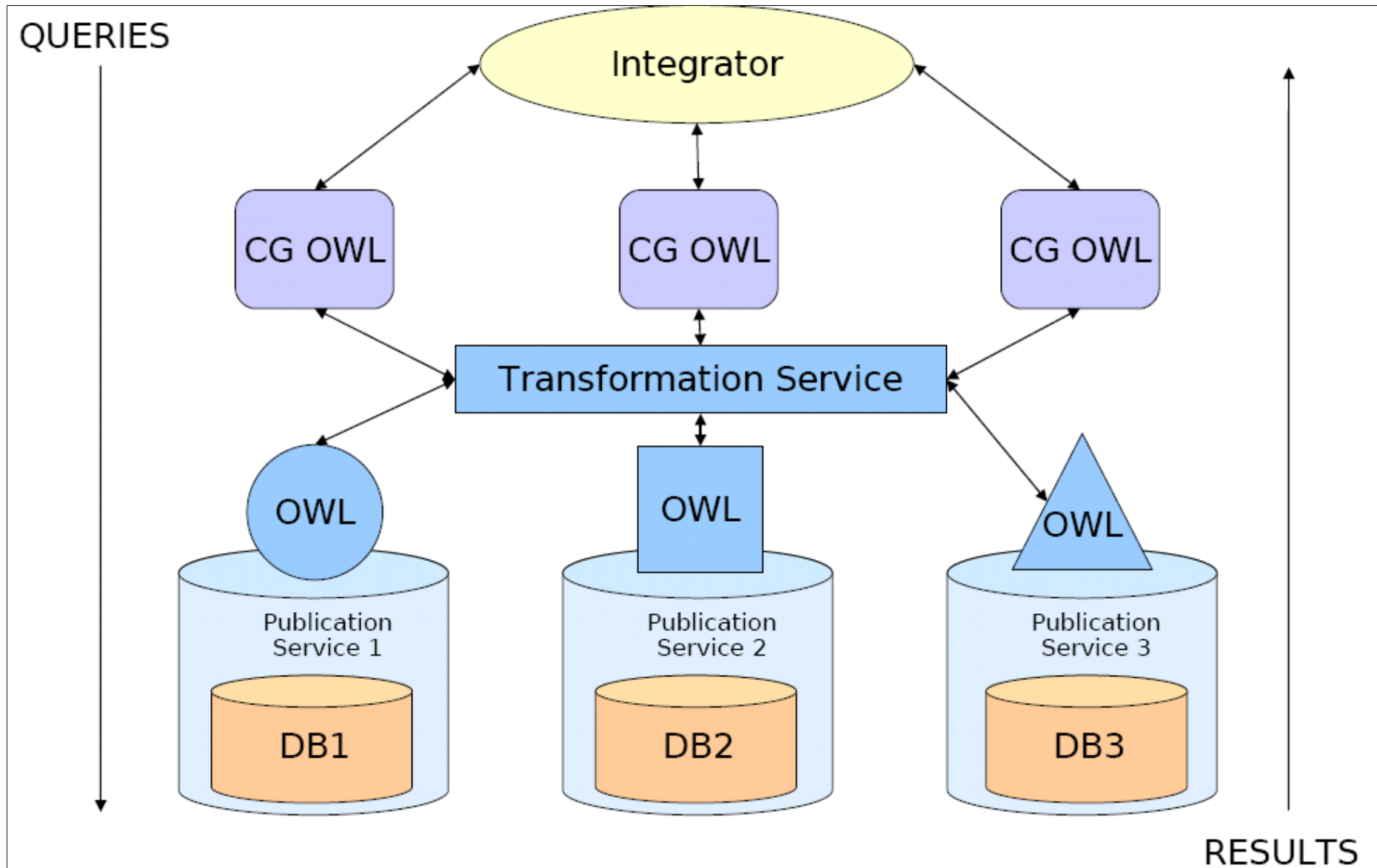
'Conceptual Models'  
of Physical Things  
– genetic maps etc



'Representations'  
of Physical Things  
- DNA Sequences



# THE COMPARAGRID ARCHITECTURE



# Components for Semantic Integration through CG-OWL

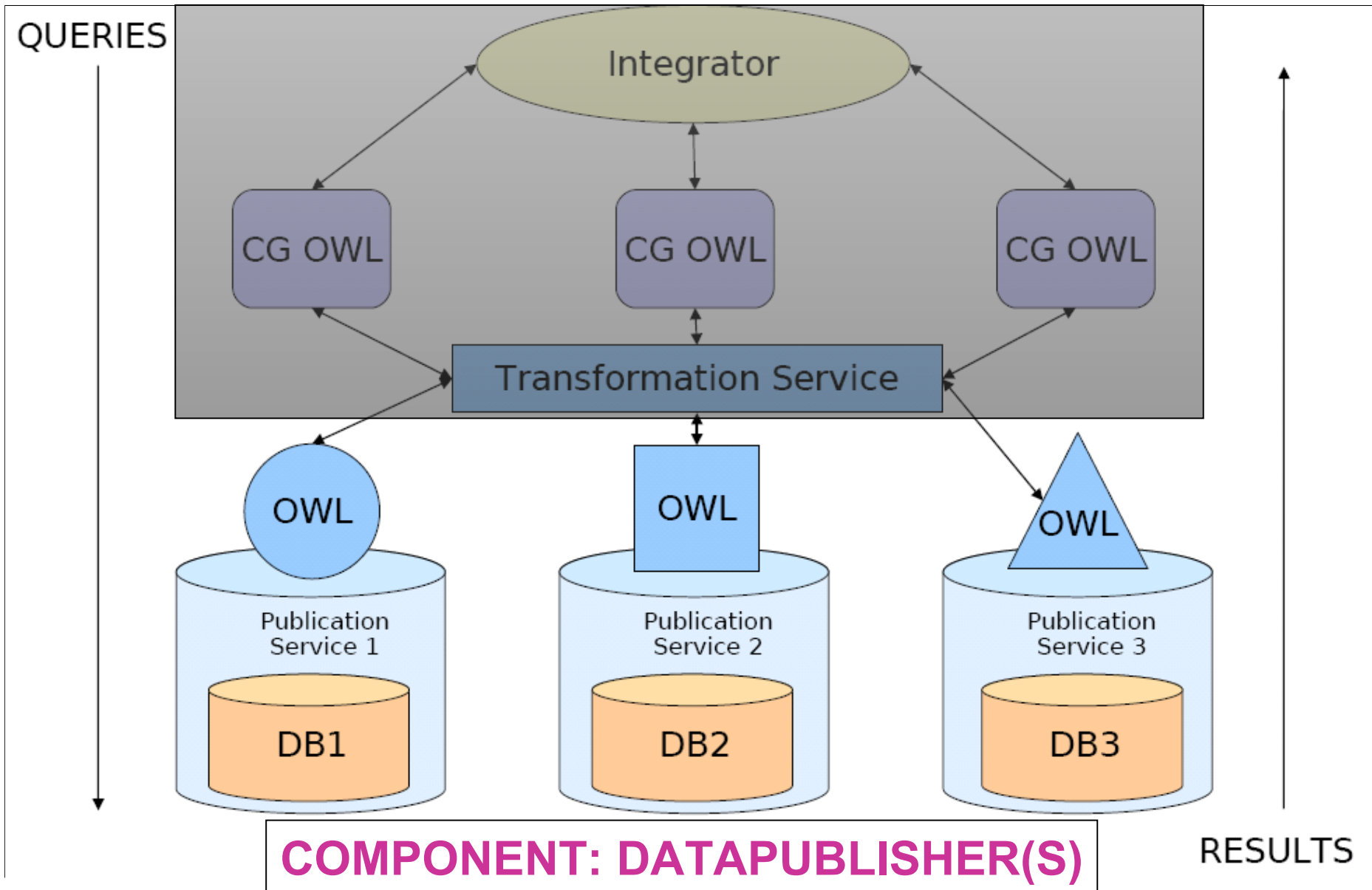
**Data Publisher:** automatically publishes an OWL-DL representation of any database schema (tables and columns) [**schema.owl**] and provides roundtrip query planning and execution **OWL → SQL → OWL**

**RuncibleGUI Protégé Plug-in Tool:** enables visual mapping of concepts and properties in a published **schema.owl** to the **cg.owl**. Generates a Runcible **mapping.xml** rule set.

**Runcible Transformation Engine:** Haskell-based engine for unambiguously applying Runcible **mapping.xml** rules. Queries accepted in **cg.owl** are translated into a particular **schema.owl** by application of the mapping rules, conversely results data is projected from the **schema.owl** provided by query of the Publisher layer in to the **cg.owl** provided by the Transformation layer.

**Pussycat Browser Application:** Integrates data through the Transformation Service, renders CG-OWL classes graphically, composes queries as CG-OWL axioms, provides consistency and inference checking over ontologies through Pellet or FaCT++

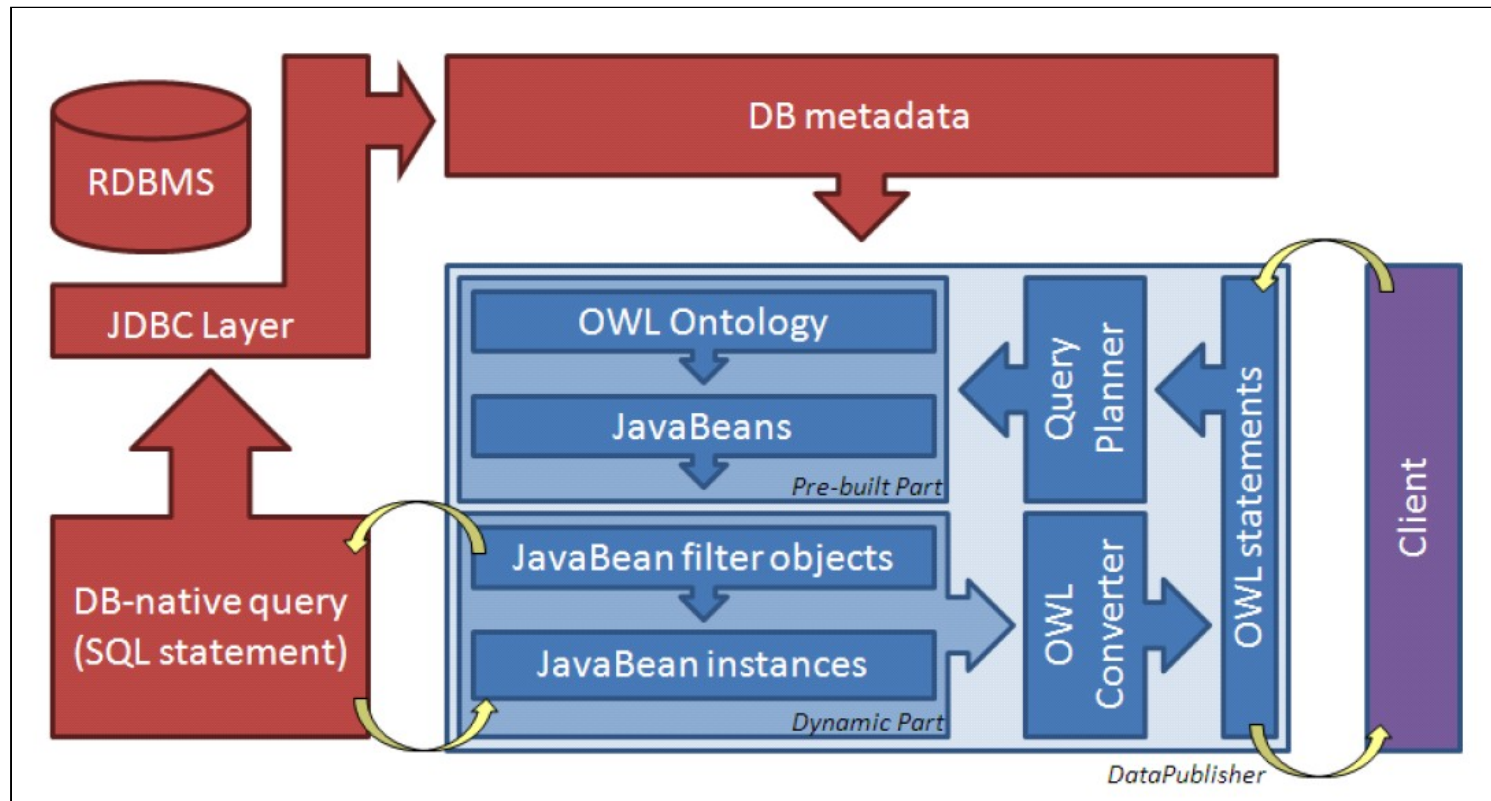
# THE COMPARAGRID ARCHITECTURE



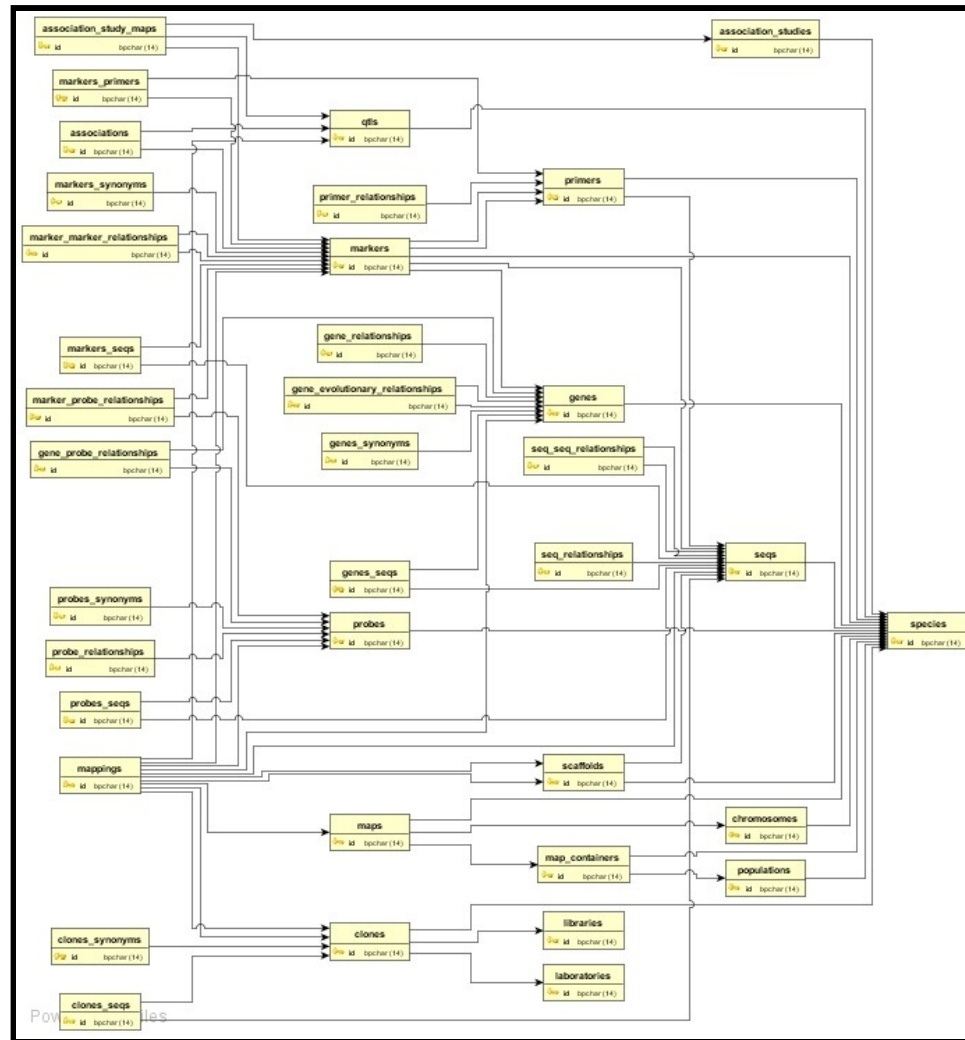
# The Data Publisher

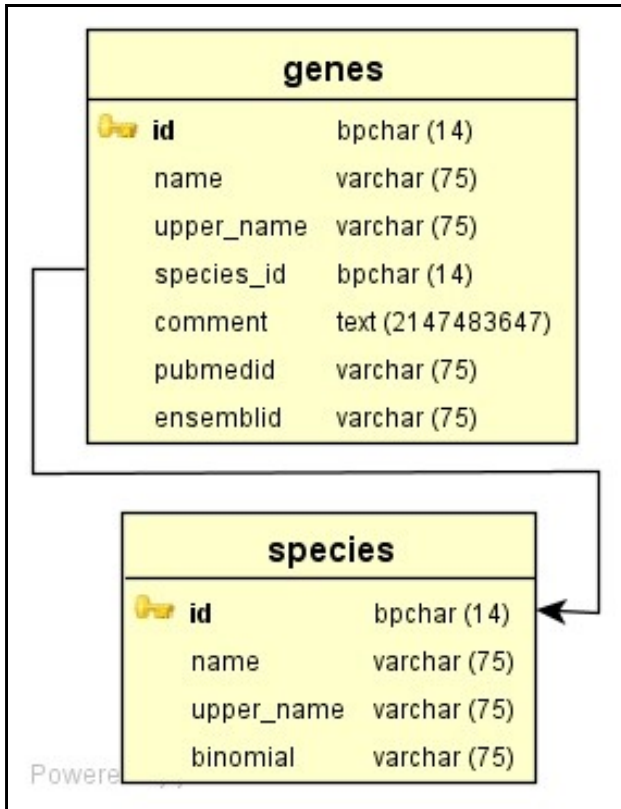
A Semantic Web enabled query interface layer for relational databases, both an API for developing plugins and custom formats and an application to build a webservice from scratch.

Individual Data Publishers are configured (with Maven POM) to connect to specified Datasources, and publish a **schema.owl** representation of the DB using appropriate DB specific plug-in modules. Publishers are deployed as WebServices exposing the standard ComparaGRID Query API.



# Roslin's ArkDB: Genomic Mapping Information for Farm Animals





ArkDB Schema Fragment



OWL Fragment

**CLASS Genes\_Record**

Genes\_Record\_has\_Species\_Record  
exactly 1 Species\_Record

## Querying the Data Publisher

The Data Publisher Query API can return the whole **schema.owl**, or it can be queried with **OWL Descriptions** composed in its specific schema.owl ontology. It translates the Query to SQL, and converts the SQL results back into a set of axioms in the schema.owl ontology.

*For example*

**"Ask for all the Species in ARK"**

OWL Query:       **Species**

.....Planned datasource query in 30.886s.

SQL Query:

```
SELECT species.id, species.upper_name,  
species.binomial, species.name  
FROM species
```

.....Performed query in 0.554s.

.....14 results got converted to OWL in 0.399s.

*More useful example...*

## **"Ask for all the maps for Pig Chromosome 7"**

### OWL Query

```
ObjectUnionOf( ObjectIntersectionOf( Chromosomes DataValue(Chromosomes_Record_has_Id ARKCHR00000007)) ObjectIntersectionOf( DataValue(Species_Record_has_Id ARKSPC00000014) Species) ObjectIntersectionOf( Maps ObjectSomeValueFrom(Maps_Record_has_Chromosomes_Record ObjectIntersectionOf( Chromosomes DataValue(Chromosomes_Record_has_Id ARKCHR00000007))))))...
```

Planned datasource query in 95.259s.

### SQL Queries

```
SELECT chromosomes.chr_order, chromosomes.upper_name, chromosomes.name, chromosomes.species_id, chromosomes.id FROM chromosomes WHERE chromosomes.id = 'ARKCHR00000007'
```

Performed query in 0.475s. 1 results got converted to OWL in 0.016s.

```
SELECT species.id, species.upper_name, species.binomial, species.name FROM species WHERE species.id = 'ARKSPC00000014'
```

Performed query in 0.043s. 1 results got converted to OWL in 0.012s.

```
SELECT maps.usefulname, maps.species_id, chromosomes.upper_name, maps.comment, maps.map_type, maps.map_container_id, chromosomes.id, maps.name, maps.upper_name, chromosomes.chr_order, chromosomes.name, maps.id, maps.pubmedid, maps.chromosome_id, chromosomes.species_id FROM maps, chromosomes WHERE maps.chromosome_id=chromosomes.id AND chromosomes.id = 'ARKCHR00000007'
```

Performed query in 1.074s. 40 results got converted to OWL in 0.66s.

## Mapping schema.owl representations to ComparaGRID.owl

In order to integrate Real Mapping Information from different published databases, which will have different schema.owl representations, we must convert the concepts in schema.owl to concepts in cg.owl.....

We have to convert the 'simple' database representation of tables and fields, to the domain ontology, with its more complex, universal, model of physical things, their representations and models (maps).

## Mapping schema.owl representations to ComparaGRID.owl

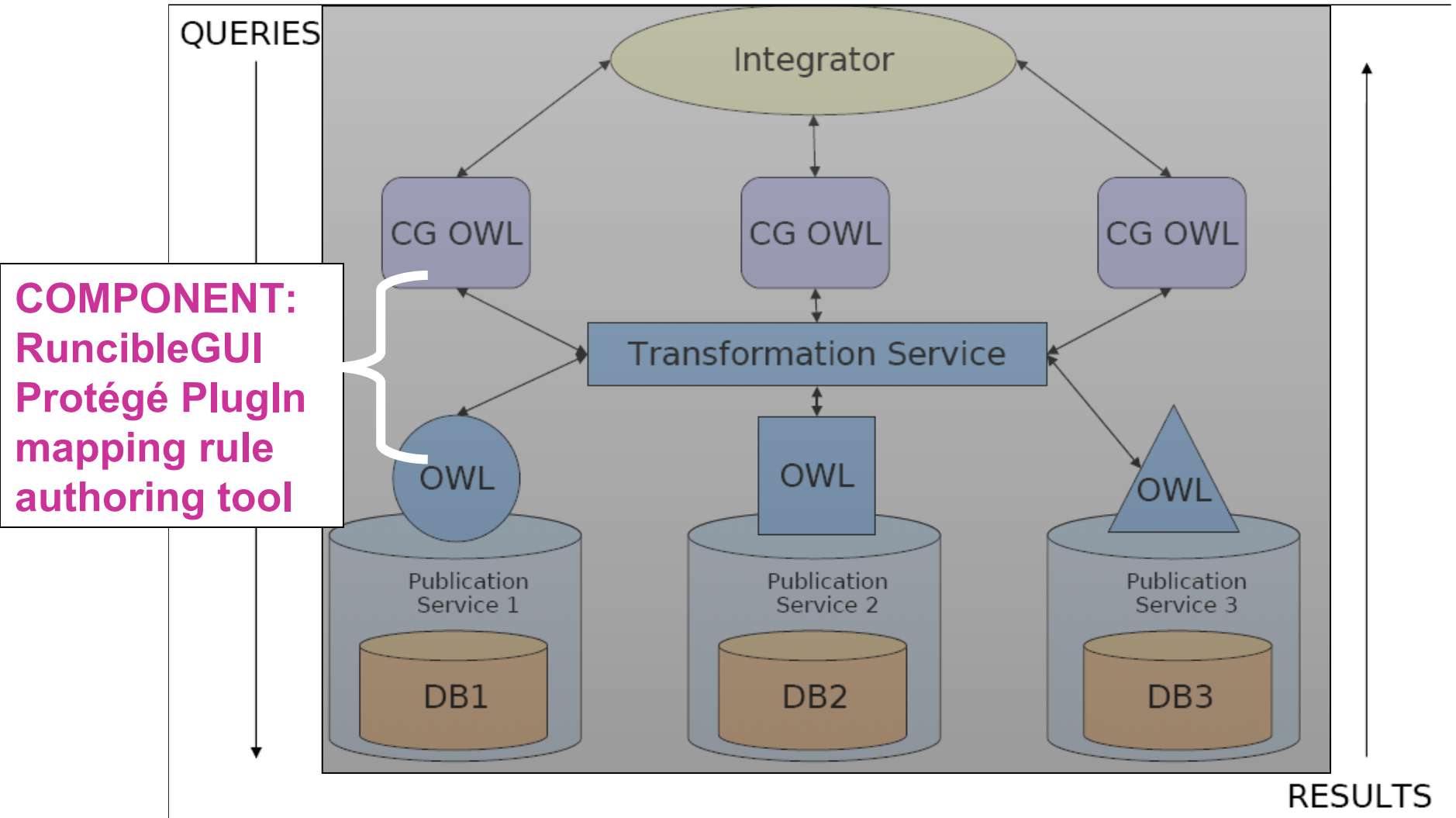
For example: Ark Database contains information about the positions of genetic markers on genetic map, and genetic markers have associated DNA sequence Information

These concepts 'Marker' 'Map' 'Sequence'.. have to be mapped to their representation in the common comparagrid model

This mapping process is *\*difficult\** for a Biologist!

So we need good tool support..... and to automate as much as possible

# THE COMPARAGRID ARCHITECTURE



## Mapping schema.owl → cg.owl: The RuncibleGUI Protégé Plug-in

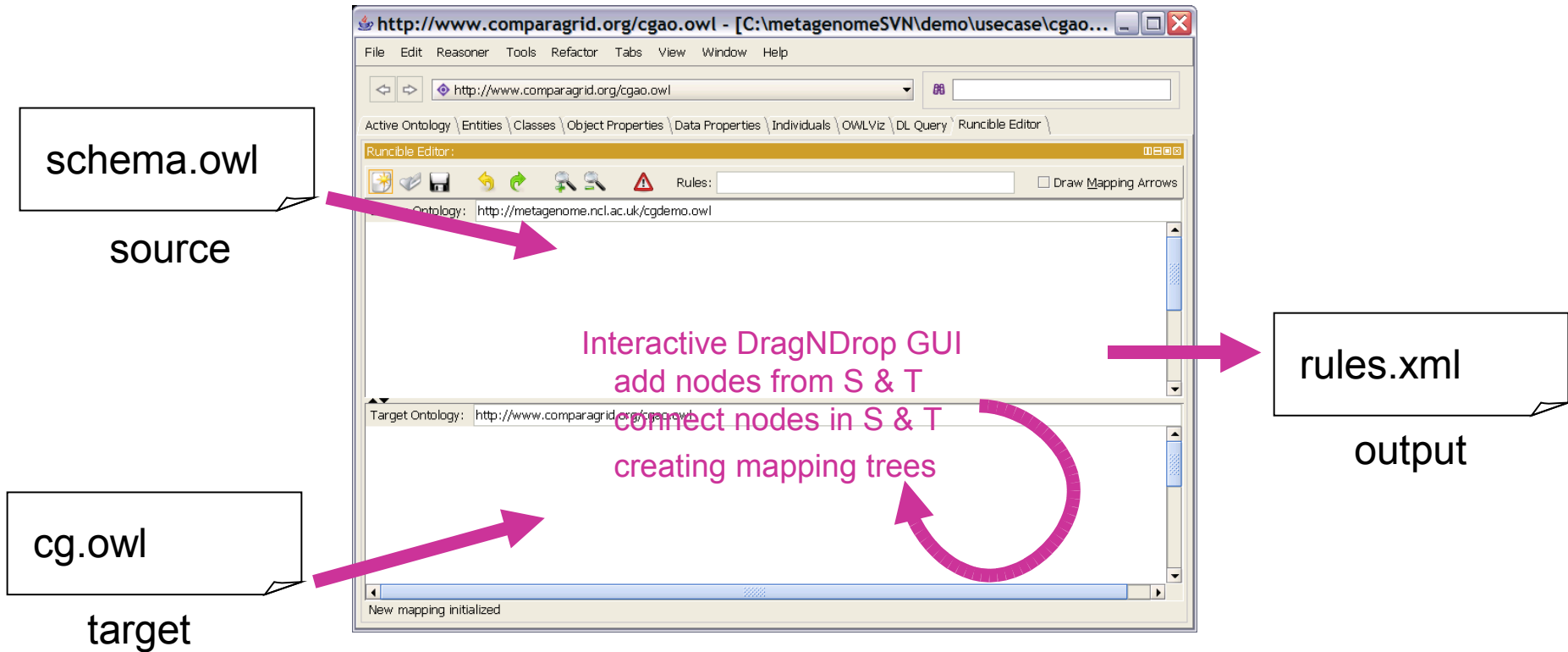
Typically not simply a matter of mapping **DB TypeA** → **CG TypeX**

Mapping also depends on the types of relationships each data object can have in each ontology

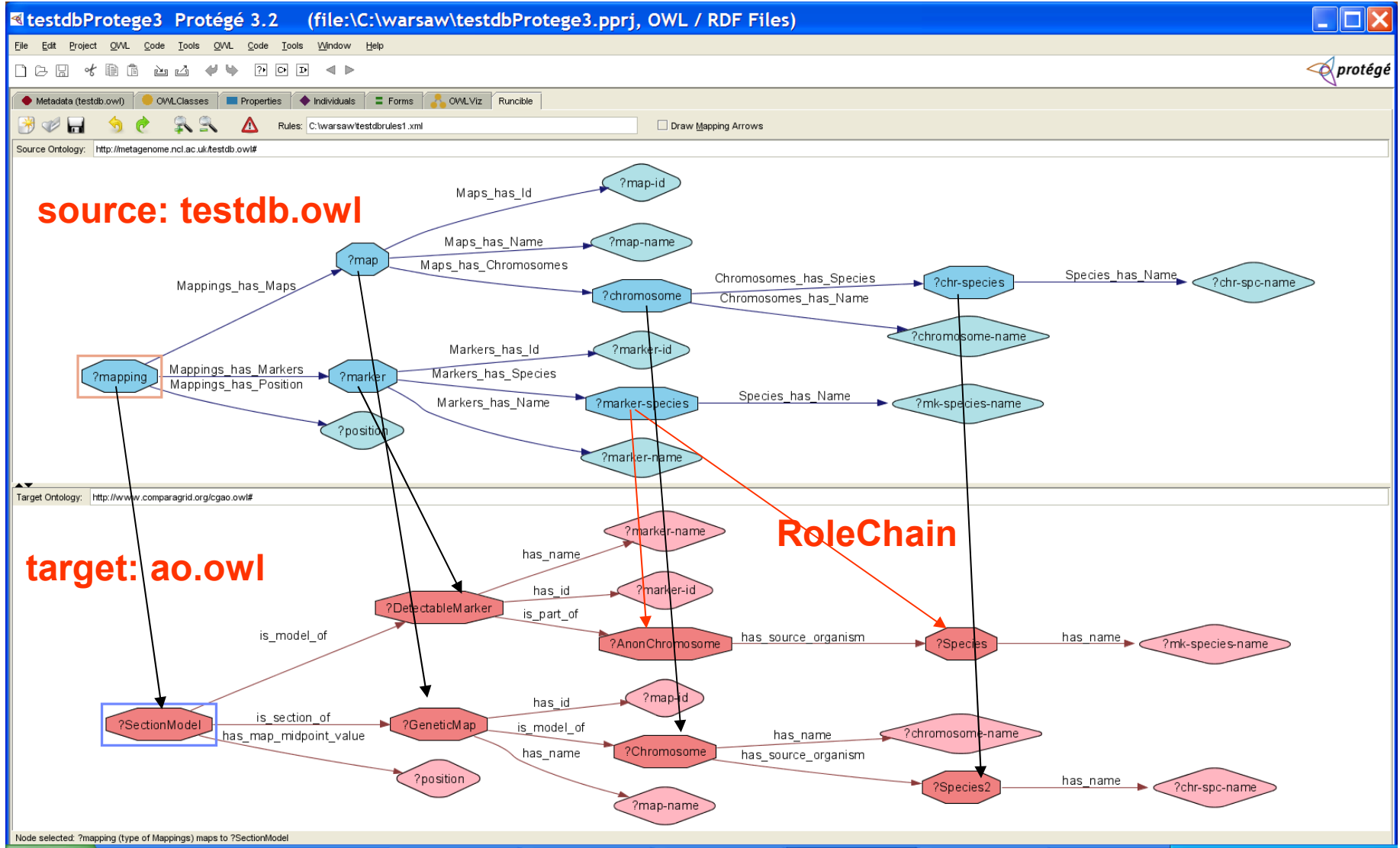
Therefore rules have to be designed which map each data object in the context of what relationships it participates in

It is **HARD** to create these mapping rules – even with a GUI tool: needs intimate knowledge of both the source database and the ComparaGRID Ontology

# RuncibleGUI Protégé Plug-in Tool



# Generating mapping rules: mappings



Builds a paired Mapping Tree – (can have multiple separate pairs)

# Partial mapping rules: ArkDB.owl → CG.owl

http://www.comparagrid.org/cgao.owl - [C:\metagenomeSVN\trunk\ontology\application\CAOV3\OWL\cgaoV0-3-5.owl]

File Edit Reasoner Tools Refactor Tabs View Window Help

cgao.owl http://www.comparagrid.org/cgao.owl

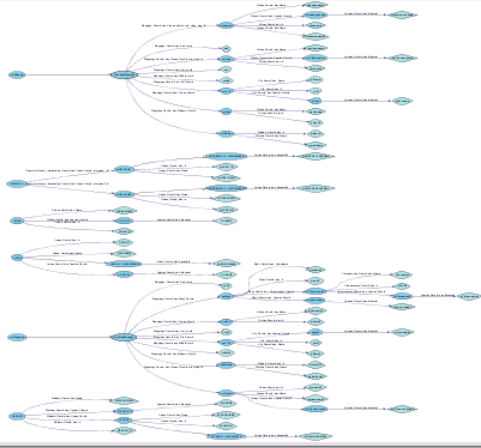
Active Ontology Entities Classes Object Properties Data Properties Individuals OWLViz DL Query **Runnable Editor**

Runnable Editor:  Draw Mapping Arrows

Rules: C:\metagenomeSVN\demo\usecase\ark-rules-usecase.xml

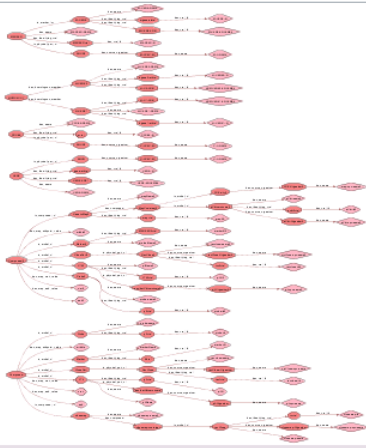
Source Ontology: http://metagenome.ncl.ac.uk/cgdemo.owl

**Nodes from the ArkDB Ontology**



Target Ontology: http://www.comparagrid.org/cgao.owl

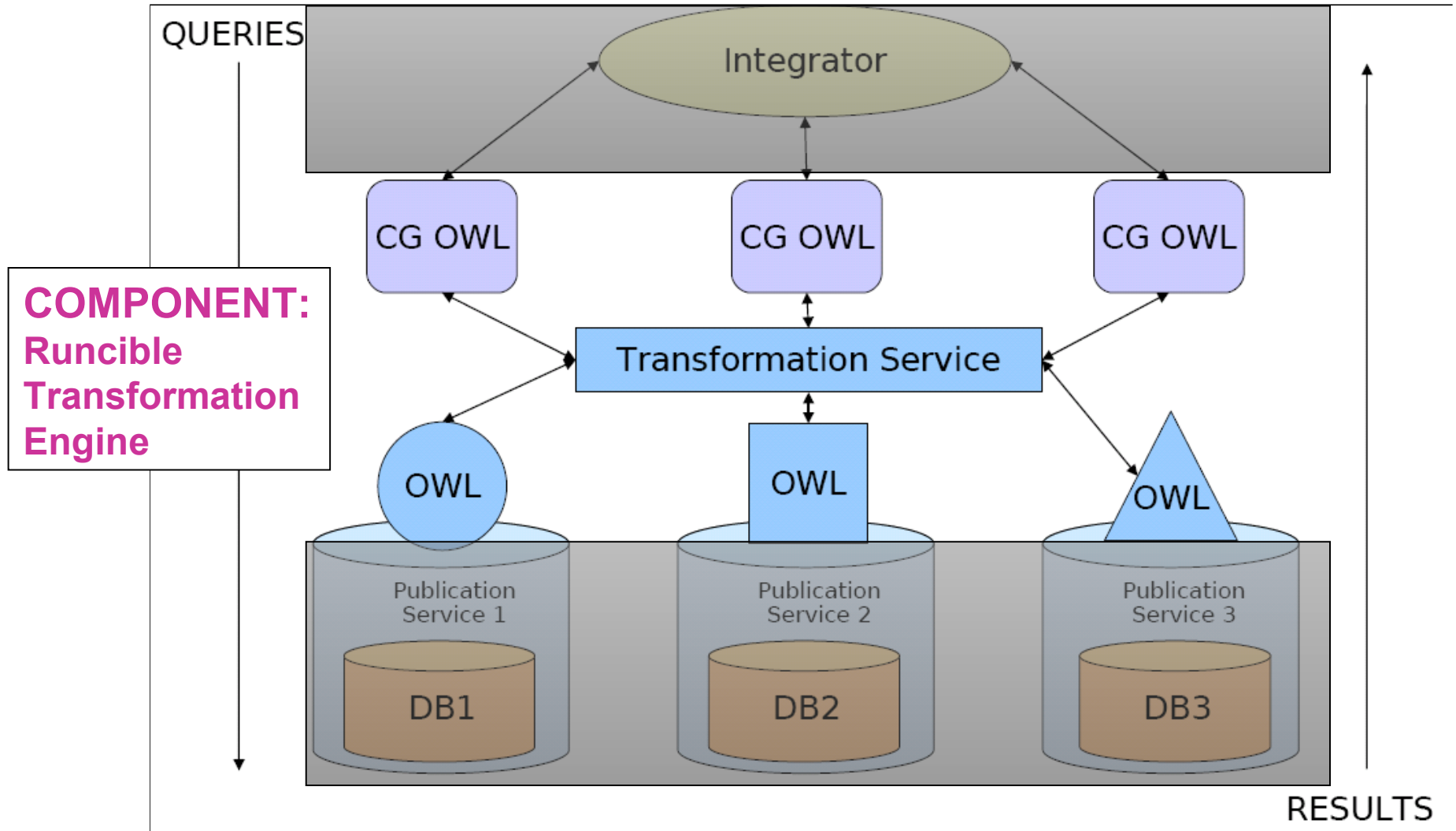
**Nodes from the CG Ontology**



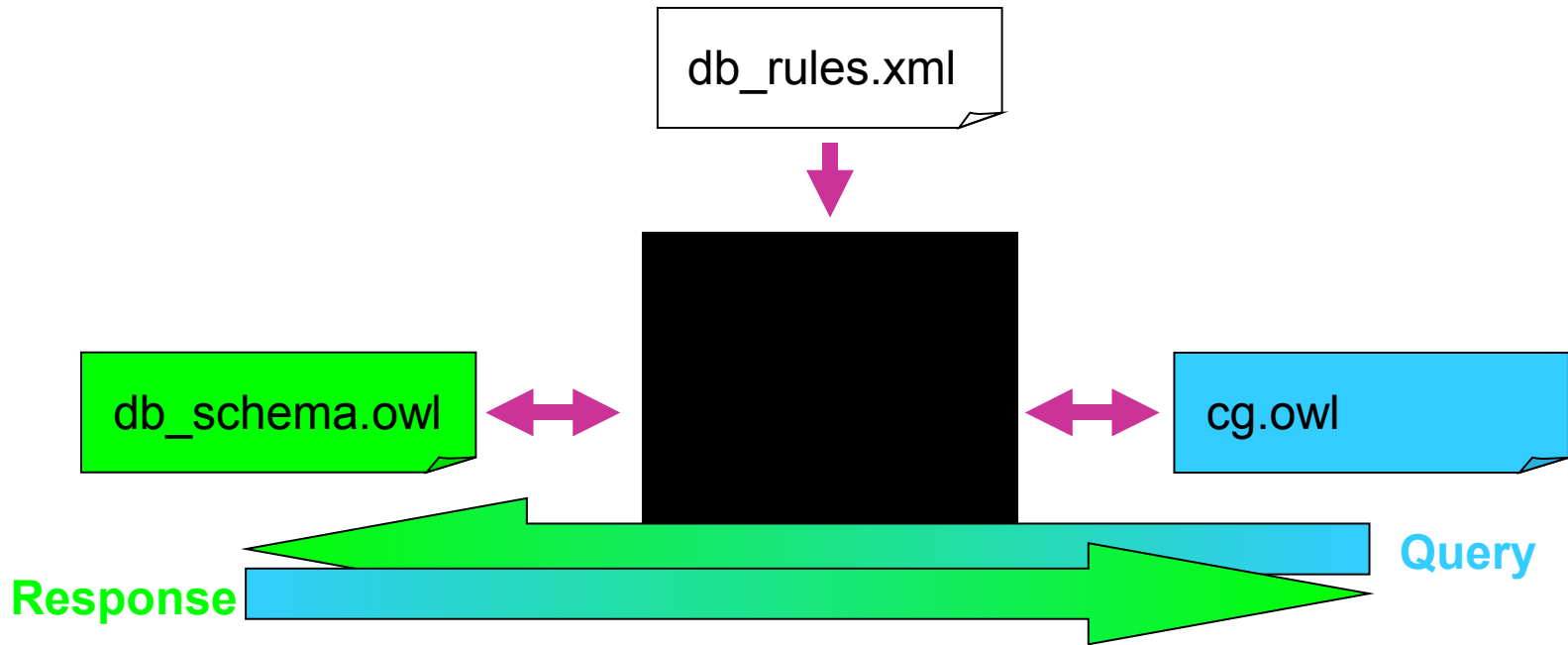
New mapping initialized

```
<mapping xmlns="http://www.comparagrid.org/runcible1.1#"
xmlns:owl="http://www.w3.org/2006/12/owl11-xml#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:source="http://metagenome.ncl.ac.uk/cgdemo.owl#"
xmlns:target="http://www.comparagrid.org/cgao.owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
<Rule>
  <for_all>SPECIES</for_all>
  <in_individuals>
    <InPlace>
      <filter>
        <owl:OWLClass URI="&source;Species"/>
      </filter>
    </InPlace>
  </in_individuals>
  <using_data>
    <DataClause>
      <variable>sp-name</variable>
      <from_range>
        <SelectDatavalue>
          <follow>
            <owl:DataProperty URI="&source;Species_Record_has_Binomial"/>
          </follow>
        </SelectDatavalue>
      </from_range>
    </DataClause>
  </using_data>
  <do_action>
    <owl:ClassAssertion>
      <owl:OWLClass URI="&target;Organism"/>
      <Variable>ORGANISM</Variable>
    </owl:ClassAssertion>
    <owl:DataPropertyAssertion>
      <owl:DataProperty URI="&target;has_name"/>
      <Variable>ORGANISM</Variable>
      <DataVariable>sp-name</DataVariable>
    </owl:DataPropertyAssertion>
  </do_action>
</Rule>
</mapping>
```

# THE COMPARAGRID ARCHITECTURE

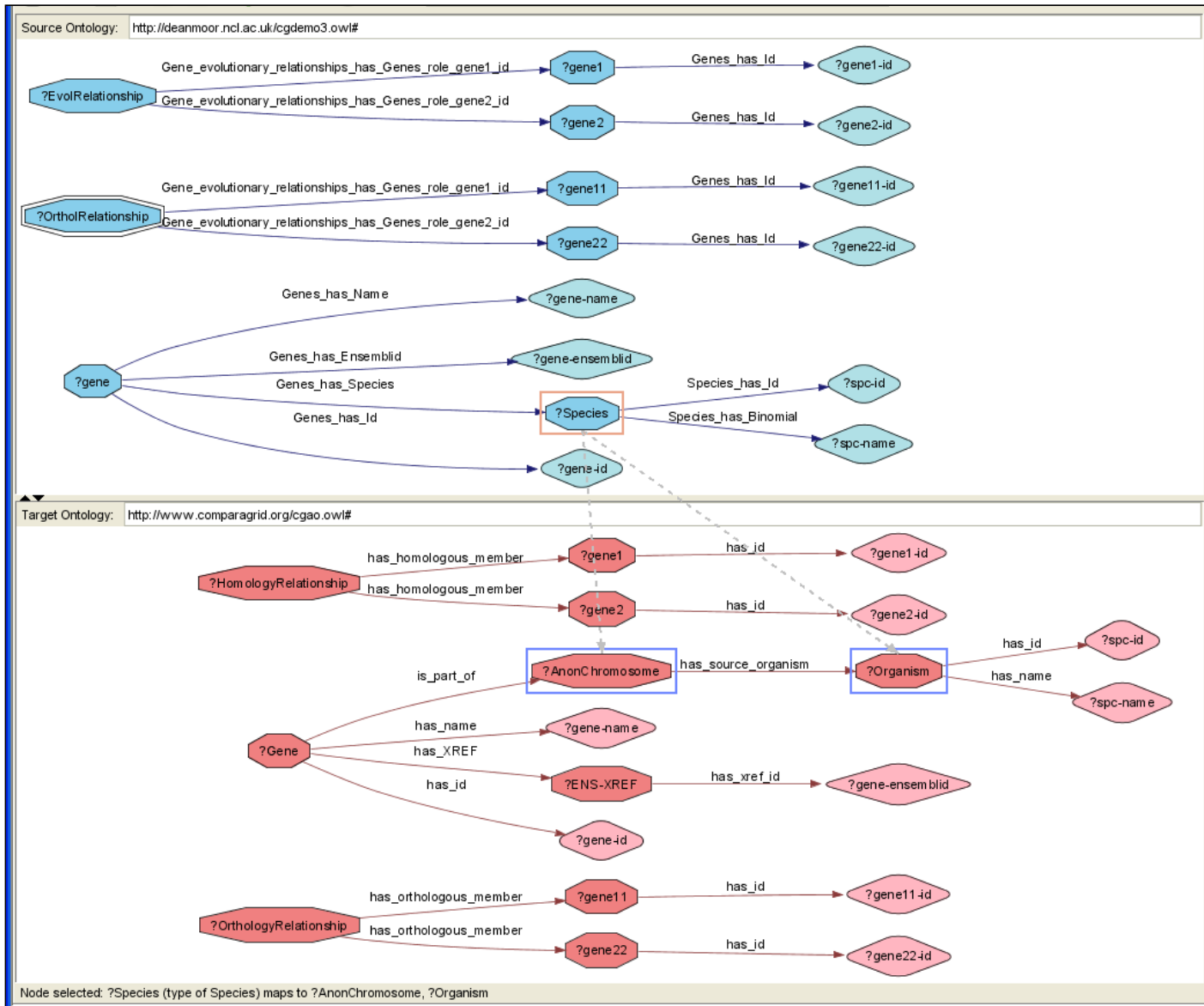


# Runcible Transformation Engine: Interconvert schema.owl and CG.owl using Runcible rules.xml



- Implemented in Haskell
- Needs to walk up and down all the mapping rules branches from each matching node
- incredibly computationally intensive for large datasets (1000s of individuals)
- performance acceptable for small or 'chunked' datasets – but will have limited live responsiveness

# Real Transformation Service Example: ArkDB.owl to CG.owl mapping



# Real Transformation Service Example: CG.owl Query

**"Get All Human Genes exhibiting Evolutionary Relationships with Pig Genes"**

Compose the query as a CG.owl Axiom (or a set of Axioms)

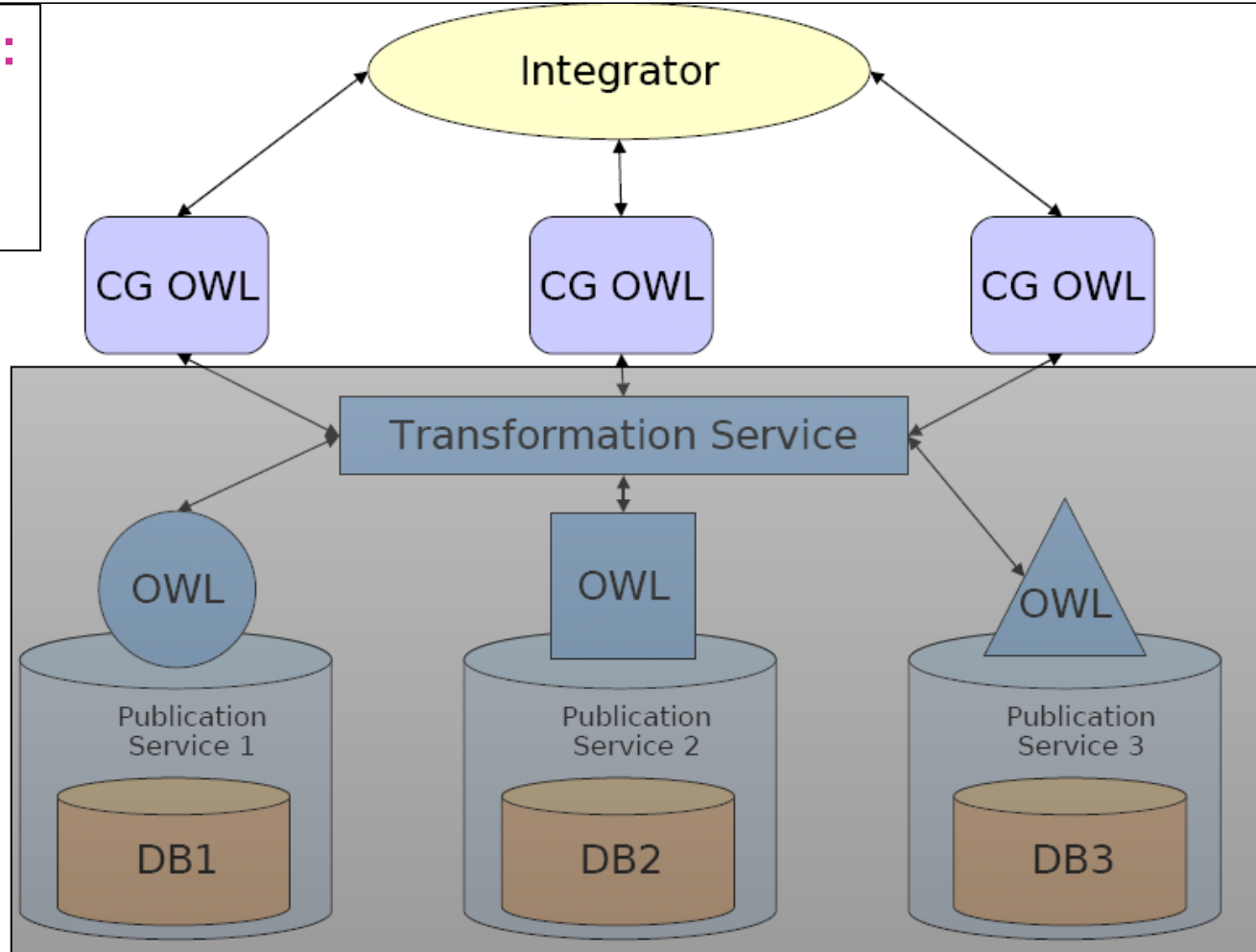
***Gene hasSpecies (hasName "Homo sapiens")  
∩ hasEvolutionaryRelationshipWith(Gene (hasSpecies  
(hasName "Sus scrofa") )***

Result returns Set of Owl Axioms representing 24 Named Human genes, with XREFS recording their ENSEMBL IDs.....

(Guaranteed to ***contain*** the correct answer.....)

# THE COMPARAGRID ARCHITECTURE

**COMPONENT:**  
Pussycat  
Ontology  
Browser



# Pussycat: The Front End Integration and User Interface for exploring ComparaGRID enabled DataSources

Integrator gives a **unified view** of the data  
for querying the data sources  
for aggregating the returned data

All queries are sent and all responses acquired as OWL

Generic Architecture – uses renderlets 'skinned' for drawing objects from the 'Genomic Mapping' Domain – but Pussycat can be used for exploring and querying any OWL ontology

# comparagrid.org

Integrating genomic data across species boundaries



ComparaGRID Home

Pussycat

Explore Services

Home

Help

Log

Browse

## Pussycat Ontology Browser

### Session Details:

- Utilising existing PussycatSessionManager (6E8614B0096F72CA20BD18C4F977AED0).
- Constructed new FluxionServiceClient.

[Clear current session](#)

Session created successfully.

If you are new to ComparaGRID and Pussycat, please go to the [tutorial](#) section.

### Session Preferences:

- Create Browsing Tab On Ontology Load
- On-Load Reasoning
- Smooth Transitions

### Loaded Ontologies:

<http://www.comparagrid.org/cgao.owl>

## Rendering Schemes

### Available Renderlets

#### Pussycat User Renderlets

*Package of user ontology browsing renderlets*

#### HelloWorldRenderlet

Renderlet to render a handy 'Hello World' message

#### DefaultClassHierarchyRenderlet

Renderlet to draw a hierarchy of classes in an OWLOntology

#### DirectClassHierarchyRenderlet

Renderlet to draw a hierarchy of direct super- and subclasses of an OWLClass

#### LoadedOntologyListRenderlet

Renderlet to draw a selectable list of OWLOntologies loaded into Pussycat

#### OntologySummaryRenderlet

Renderlet to draw a summary of a currently loaded Ontology

#### RenderletListRenderlet

Renderlet to render a list of available Renderlets

#### ResourceViewRenderlet

Renderlet to draw an HTML view of an OWLEntity

## Upload OWL

### Load OWL from:

Uploading an OWL file will unload any ontologies currently loaded into Pussycat!

URL:

File on your computer:

### Upload Information

*Parsing complete*



## Query



### Individual Properties

:

Query Using Type:

[is\\_component\\_of](#) :: [ARKLGP000C0016](#)

[is\\_model\\_of](#) :: [ARKMKR00002402](#)

[is\\_part\\_of](#) :: [ARKLGP00000016](#)

[has\\_map\\_end\\_value](#):

[has\\_map\\_position](#):

[has\\_position](#):

[has\\_map\\_start\\_value](#):

[has\\_map\\_midpoint\\_value](#):

ENSG 00000204297



Type: Gene  
Start: 8400  
End: 9100

Identifying XRefs:  
[ENSG00000204297](#)

XRefs:  
[ENSG00000204297](#)

Query using this entity:  
[Build Query...](#)

(Sus scrofa (Sus scrofa))  
(UnknownSequenceMap)  
0.0



10000.0

(Sus scrofa (Sus scrofa))  
(ARKCM00000001)  
0.0



2500.0

ARKG EN0000003

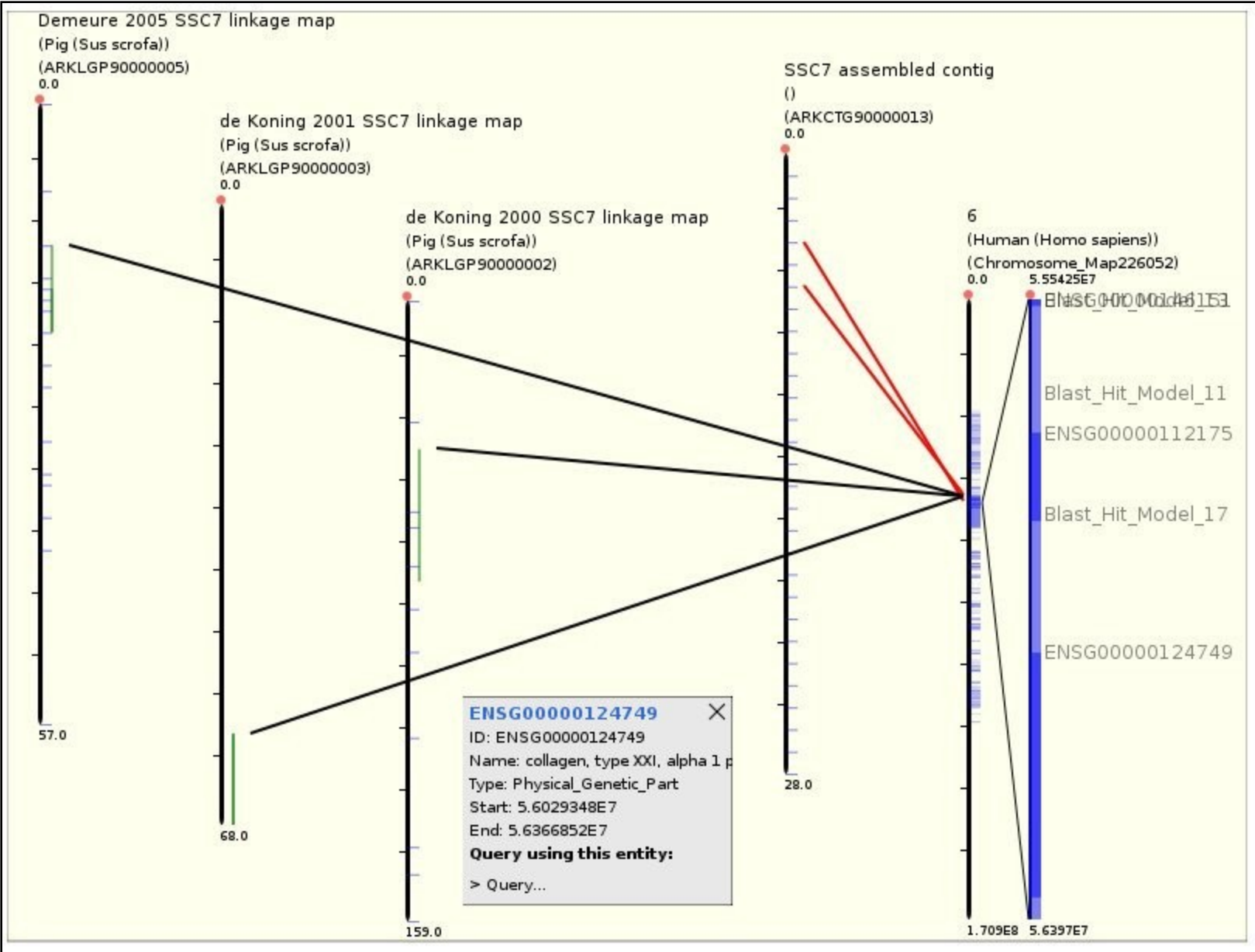


Type: Detectable\_Clone\_Part  
Start: 534.0  
End: 576.0

Identifying XRefs:  
[ARKGEN0000003](#)

XRefs:  
[ARKGEN0000003](#)

Query using this entity:  
[Build Query...](#)



## After 3 years of ComparaGRID: Where are we at?

### 3. The System

#### SUCCESS – PROOF OF CONCEPT

- All the components and tools built and functioning

**DataPublisher Service:** SQLSchema  $\leftrightarrow$  LocalOWL

**Protégé Mapping Tool Plugin**

**Haskell Transformation Engine:** LocalOWL  $\leftrightarrow$  CGOWL

**WebApplication Service:** CGOWL  $\leftrightarrow$  SVG/JS Components

- Some of the components joined together
- Queries can be performed and visualized by joining pieces manually
- Integration demonstrated between (Pig) Data in ArkDB and (Human) Data in ENSEMBL

## After 3 years of ComparaGRID: Where are we at?

### 3. The System

#### LIMITATIONS – NO LIVE SYSTEM

- Publisher: Complex Queries can be slow – but workable
- Transformation Engine Computationally Intensive: block on real time processing – slow and memory intensive – currently we need to segment large datasets and then rejoin
- Pussycat Integrator not yet fully plugged in to the architecture
- Reasoning over results not fully explored yet
- No roll out for general services yet – just our pet demo datasources
- Haven't yet been able to fully test integration of concepts across datasources yet – other than by sharing some hacked feature such as identifying XREFs

## After 3 years of ComparaGRID: Where are we at?

### 3. The System

#### EXCUSES – CUTTING EDGE TECHNOLOGIES

- Reliance on Features of 'new' OWL1.1
- Tools (Protégé, Pellet) and JAVA APIs just emerging
- Lots of Rate Limiting steps in Development Process, so haven't been able to develop and test all components fully, or explore functionality
- We all know that Integration is 'Difficult'

# After 3 years of ComparaGRID: Where are we at?

## 2. The Ontology

### SUCCESS

- We have a large, detailed, logically consistent Domain Ontology covering genomic mapping. Capturing Real World Concepts, Mapping Concepts and Sequence Data, and Relationships between these.
- We have an extended Application Ontology adding annotations and concepts needed for real data
- Ontology Design is Modular – allowing future expansion  
e.g. Experimental Detail and Evidence modules

# After 3 years of ComparaGRID: Where are we at?

## 2. The Ontology

### LIMITATIONS

- At this stage the ontology is large and complex, therefore possibly difficult to use and to describe. It needs further annotation and description and the 'labels' will have to be simplified (at least in the Application Ontology view) to make it friendly for general biologists.
- Because we haven't been able to fully test the integrated system we don't know whether we have captured yet all the relationships that we will need to be able to reason over the data – i.e. to infer that a genetic marker in one datasource is the same or equivalent to one in another datasource.
- Until we have more experience using the system we will not be able to demonstrate fully that Open World Reasoning appropriate for experimental data in relational databases.

## After 3 years of ComparaGRID: Where are we at?

### 2. The Ontology

#### EXCUSES

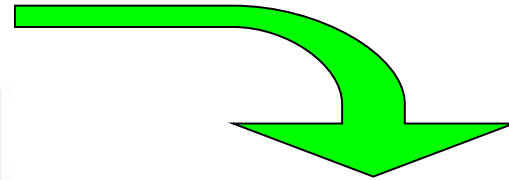
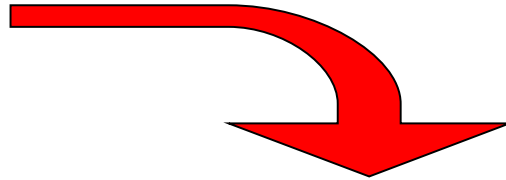
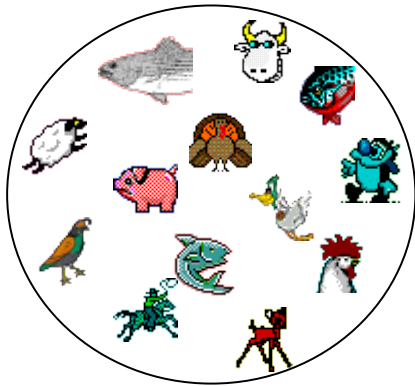
- Ontology Design is recognized to be hard
- We have been trying to make a reusable, high-level ontology for a wide domain
- Compartmentalisation of 'woolly' biological concepts is difficult
- Capturing real, potential, qualitative and conditional relationships is difficult

## After 3 years of ComparaGRID: Where are we at?

### 3. Where Next?

- Short term funding to make generic versions of the components available?
- Ambitions to take it forward to implement the full working system now that the technology is more mature

# The Future?



The  
ComparaGRID  
Mincer™

