Semantic Web and Semantic Audio technologies

Tutorial by

György Fazekas and Thomas Wilmering

Centre for Digital Music

Queen Mary University of London

School of Electronic Engineering and Computer Science

132nd Convention
April 26th-29th,
Budapest, Hungary

We are on the Web

- Slides, examples and other resources are available at:

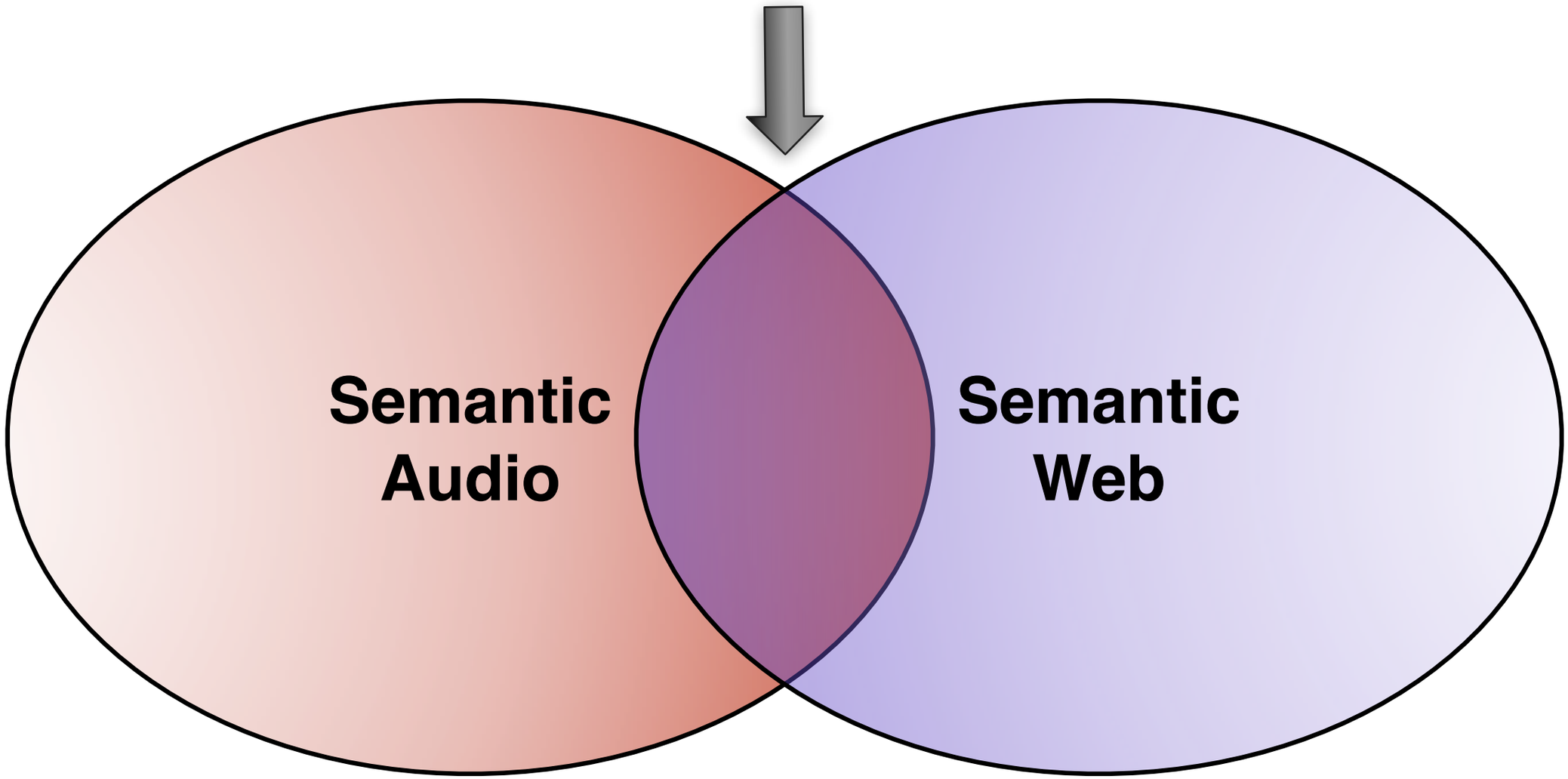
- **www.isophonics.net/content/aes132-tutorial**

Outline

- Introduction and Motivations
- Semantic Web Technologies
- Semantic Web Applications
  - Short Hands on Session (1)
- Music Ontology
- Studio Ontology
- Semantic Audio Tools
  - Short Hands on Session (2)
- Semantic Audio in Music Production

Introduction

- The focus of this tutorial is the intersection of the two fields



Introduction

- What is Semantic Audio ?

- What is the Semantic Web ?

- How are they related,
- and why should we care?

Introduction

- **What is Semantic Audio ?**
  - a confluence of technologies for
  - **interacting with audio** in human terms
- **Semantic Audio technologies include:**
  - Audio content analysis
    - e.g. Digital Signal Processing and Machine Learning
  - Information Management
  - Knowledge Representation
    - e.g. Logic, Ontologies, and database technologies

Introduction

- **What is Semantic Audio ?**
  - a confluence of technologies for
  - **interacting with audio** in human terms

- **Semantic Audio technologies include:**
  - Audio content analysis
    - e.g. Digital Signal Processing and Machine Learning
  - Information Management
  - Knowledge Representation
    - e.g. Logic, Ontologies, and database technologies

Introduction

- **What is Semantic Audio ?**
  - a confluence of technologies for
  - **interacting with audio** in human terms

- **Semantic Audio technologies include:**
  - Audio content analysis
    - e.g. Digital Signal Processing and Machine Learning
  - Information Management
  - Knowledge Representation
    - e.g. Logic, Ontologies, and database technologies

Introduction

- **What is the Semantic Web ?**
- (1) a diverse network of **interconnected data and services**
- in principle, it is similar to how documents are linked using hypertext
- (2) a machine-interpretable representation of the World Wide Web

Introduction

- **What is the Semantic Web ?**
- (1) a diverse network of **interconnected data and services**
- in principle, it is similar to how documents are linked using hypertext
- (2) a machine-interpretable representation of the World Wide Web

Introduction

- **What is the Semantic Web ?**
- (1) a diverse network of **interconnected data and services**
- in principle, it is similar to how documents are linked using hypertext
- (2) a machine-interpretable representation of the World Wide Web

Introduction

- **What is the Semantic Web ?**

- **The objective is:**

- Enable machines to complete complex (search) tasks currently requiring human-level intelligence

Motivations

- **How Semantic Audio and the Semantic Web are related?**

- A proliferation of music content on the Web requires Semantic Audio technologies for better access to this content.

Motivations

- **How Semantic Audio and the Semantic Web are related?**

- Semantic Web technologies enable better representation and access to music related information.

Motivations

- **Why should we care?**

- Music Information Retrieval:

  - Find me upbeat and catchy songs between 130-140 bpm, performed by artists collaborating in the London-Shoreditch area, and sort them by musical key.

Motivations

- **Why should we care?**

- Music production:

  - Find me guitar riffs in all my recording projects where an echo and compressor were applied with the given parameters.

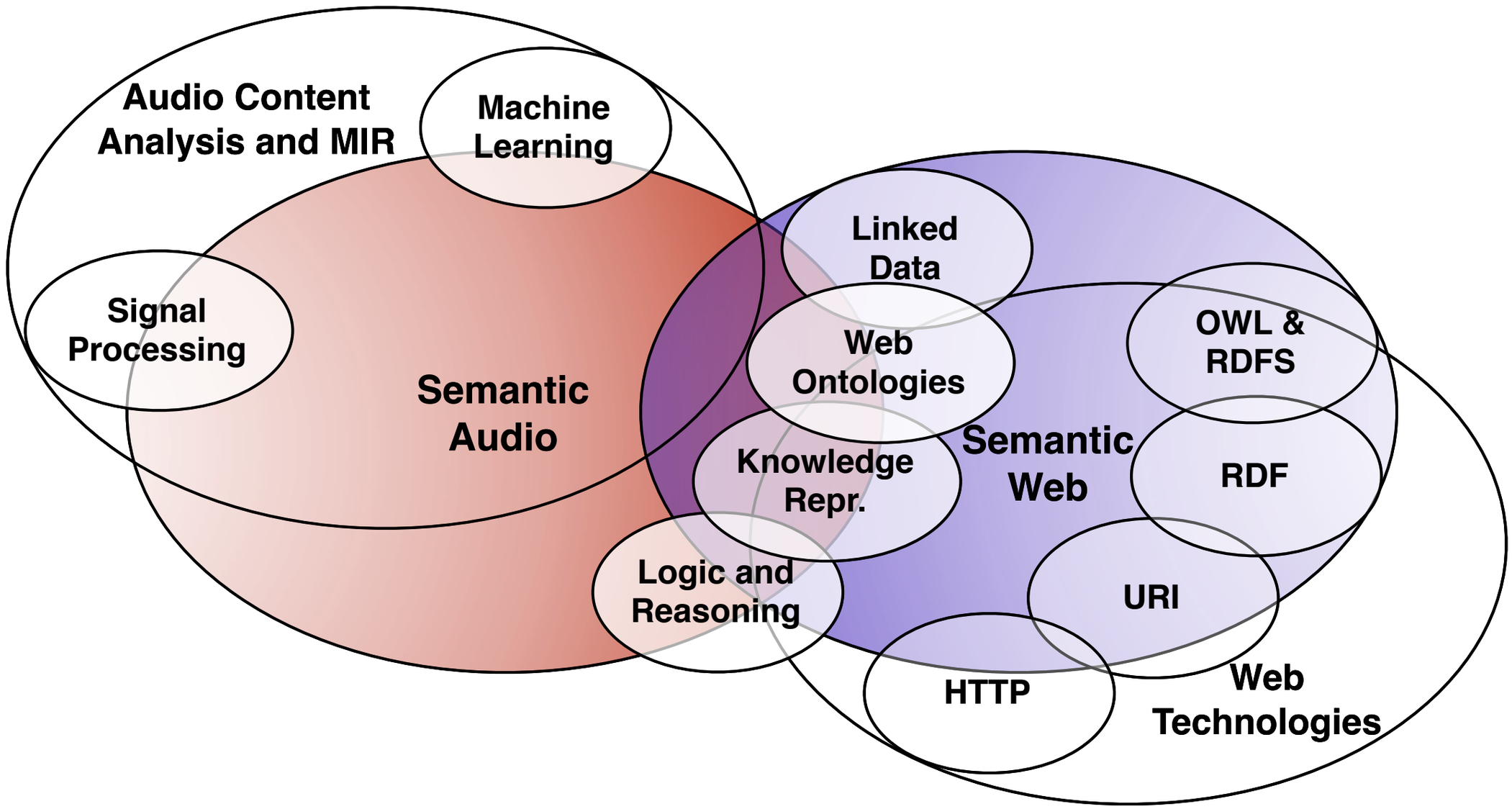
Motivations

- **Why should we care?**

- These queries/applications require clever
  - content analysis
  - knowledge representation
  - information management

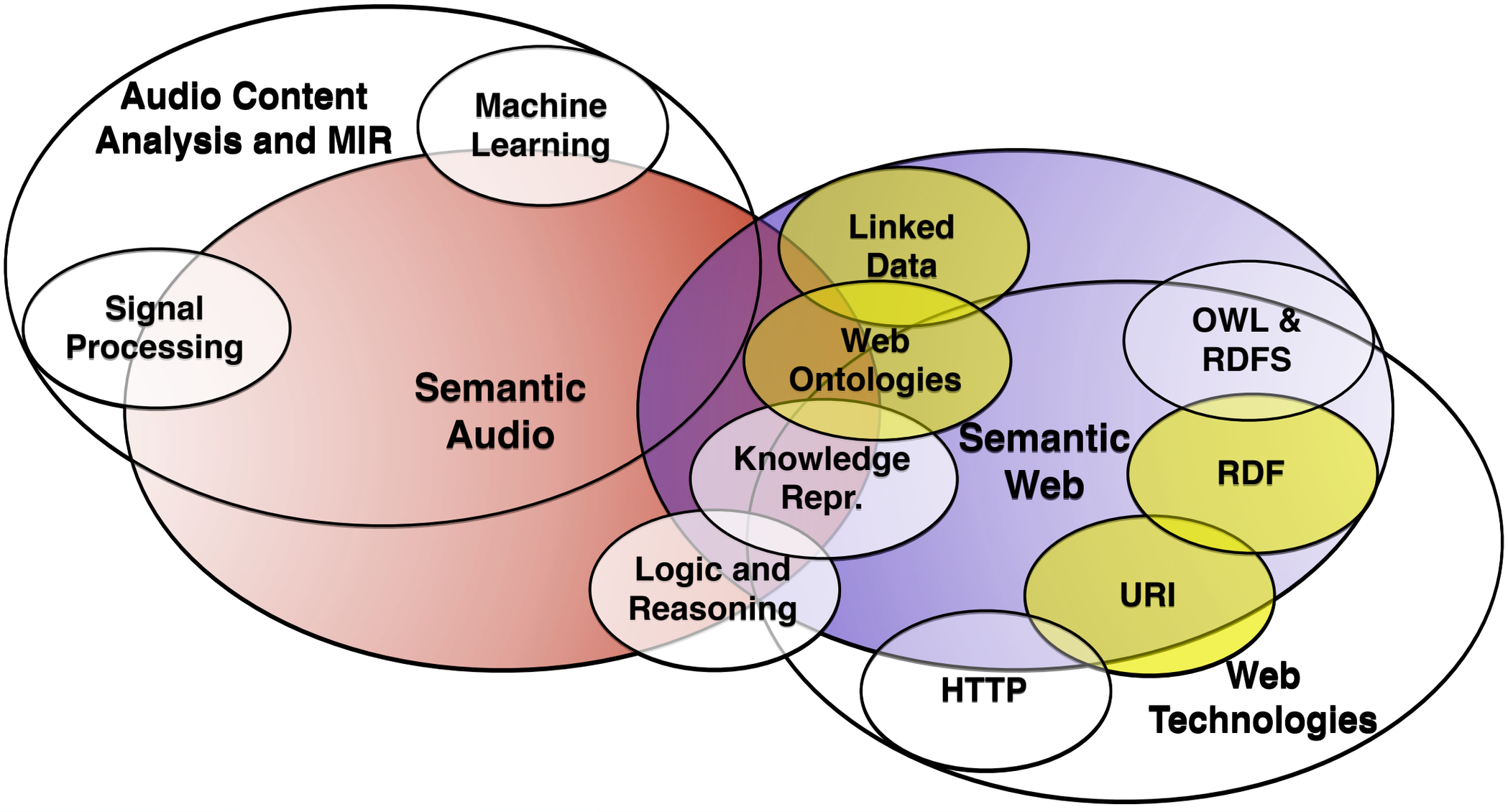
Tutorial Focus

- The focus of this tutorial is the intersection of the two fields



Tutorial Focus

- and the areas marked below



Semantic Web Technologies

Linked Data

Linked Data ? = **Semantic Web** ? = **Web of Data**

- These concepts are often used interchangeably
- Linked Data is a recent movement that focusses on creating a **web of data**
- Just like the Web is a **web of documents**
- Broader premises of the Semantic Web will be realised in the future

Linked Data

Linked Data ? = **Semantic Web** ? = **Web of Data**

- These concepts are often used interchangeably
- Linked Data is a recent movement that focusses on creating a **web of data**
- Just like the Web is a **web of documents**
- Broader premises of the Semantic Web will be realised in the future

Linked Data

Linked Data ? = **Semantic Web** ? = **Web of Data**

- These concepts are often used interchangeably
- Linked Data is a recent movement that focusses on creating a **web of data**
- Just like the Web is a **web of documents**
- Broader premises of the Semantic Web will be realised in the future

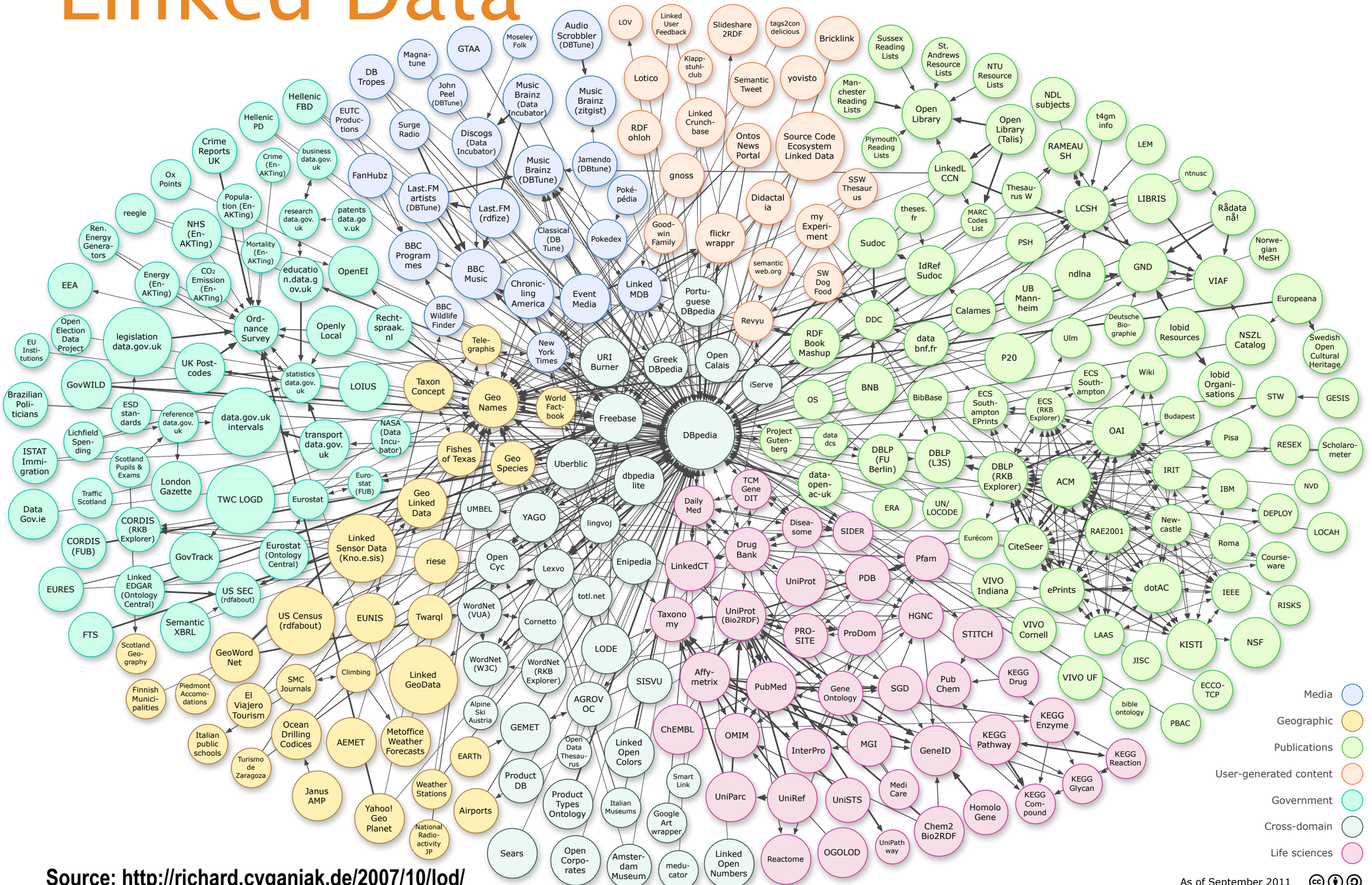
Linked Data

Linked Data ? = **Semantic Web** ? = **Web of Data**

- These concepts are often used interchangeably
- Linked Data is a recent movement that focusses on creating a **web of data**
- Just like the Web is a **web of documents**
- Broader premises of the Semantic Web will be realised in the near future

Queen Mary University of London

Linked Data



Source: http://richard.cyganiak.de/2007/10/lod/

As of September 2011

Demo Videos

- **What is possible now?**

- The following demos show:

  - (1) audio applications that collect and use data from the Semantic Web

  - (2) audio applications that utilise Semantic Web technologies (but not necessarily linked data)

Demo Video 1

Demo Video 2

Demo Videos

- **How do these applications really work?**
  - They combine information from different sources
- To achieve this we need:
  - interoperability
  - queryability
- **and also:**
  - extensibility
  - modularity

Demo Videos

- **How do these applications really work?**
- They combine information from different sources

- **To achieve this we need:**
  - interoperability
  - queryability
- **and also:**
  - extensibility
  - modularity

Basic Requirements

- **How do these applications really work?**
- They combine information from different sources

- **To achieve this we need:**
  - **interoperability** between different data sources
  - **queryability**
- **and also:**
  - extensibility
  - modularity

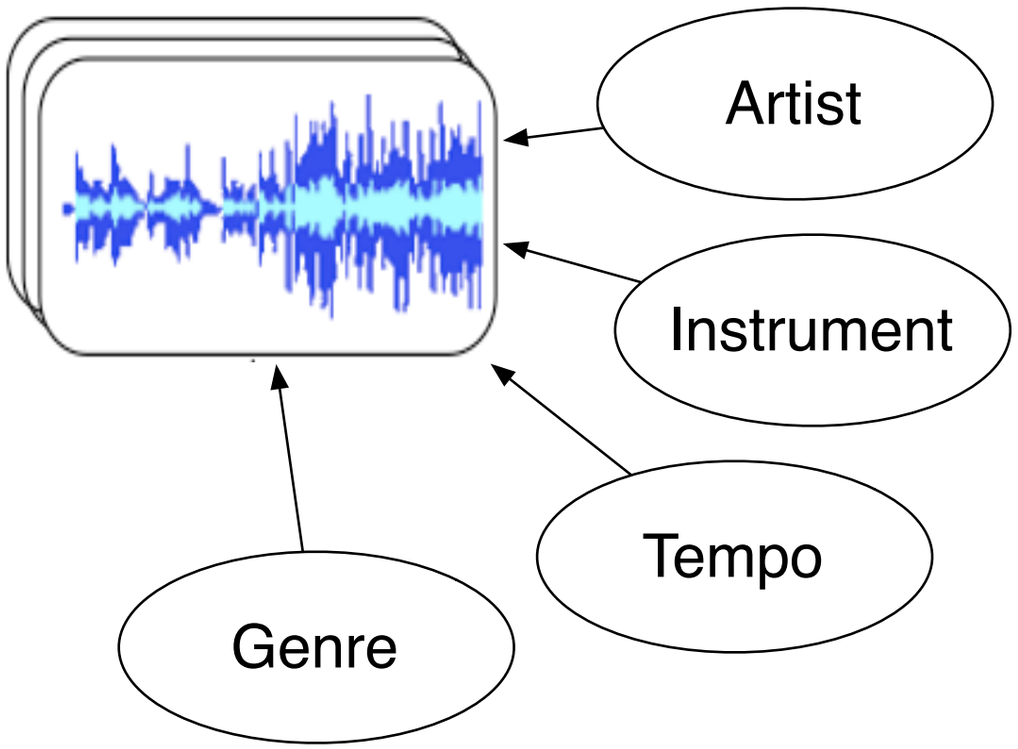
Basic Requirements

- **How do these applications really work?**
- They combine information from different sources

- **To achieve this we need:**
  - **interoperability**
  - **queryability**
- **and also:**
  - extensibility
  - modularity

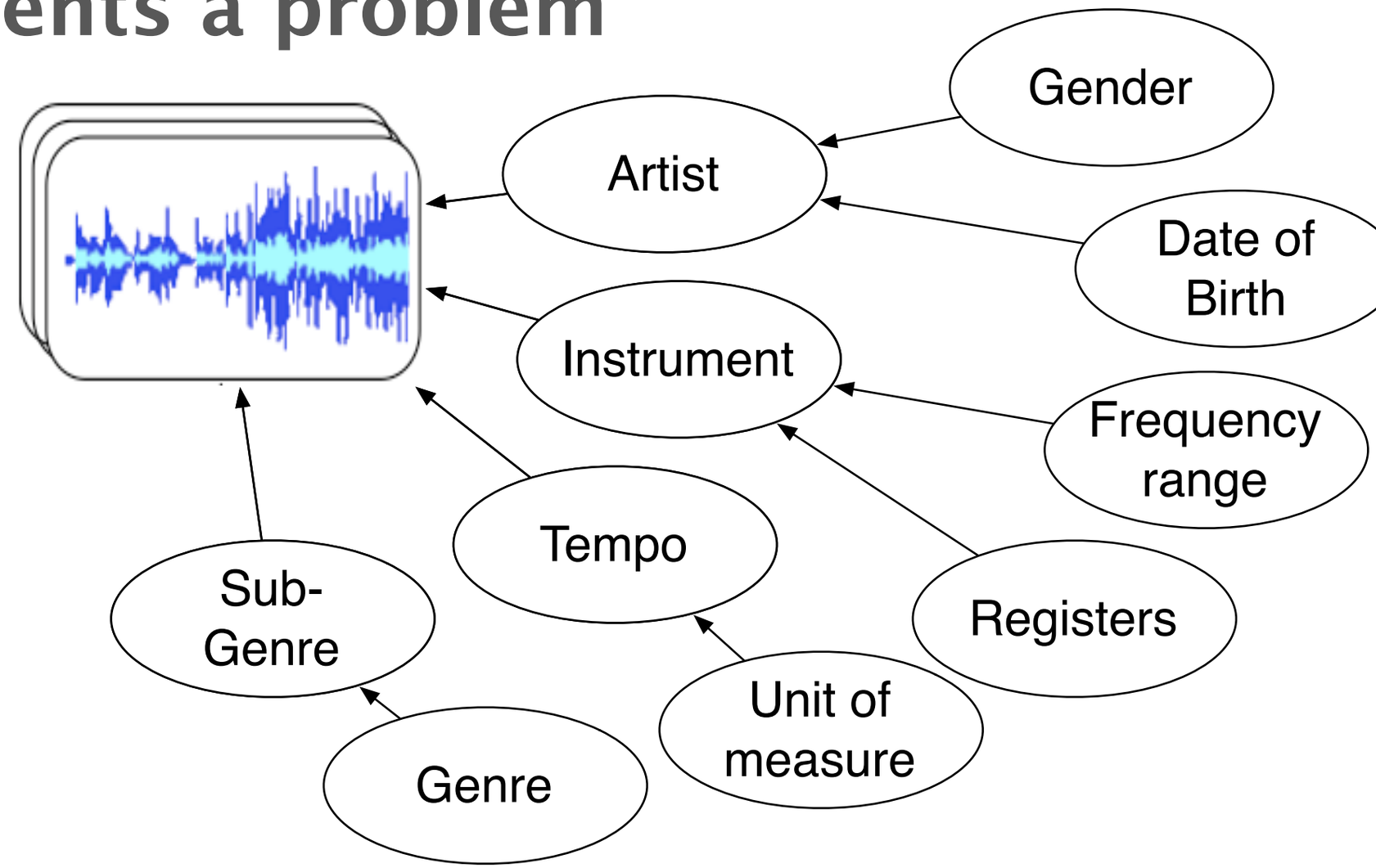
Metadata Structural Diversity

- **But, the heterogeneity of musical metadata presents a problem**



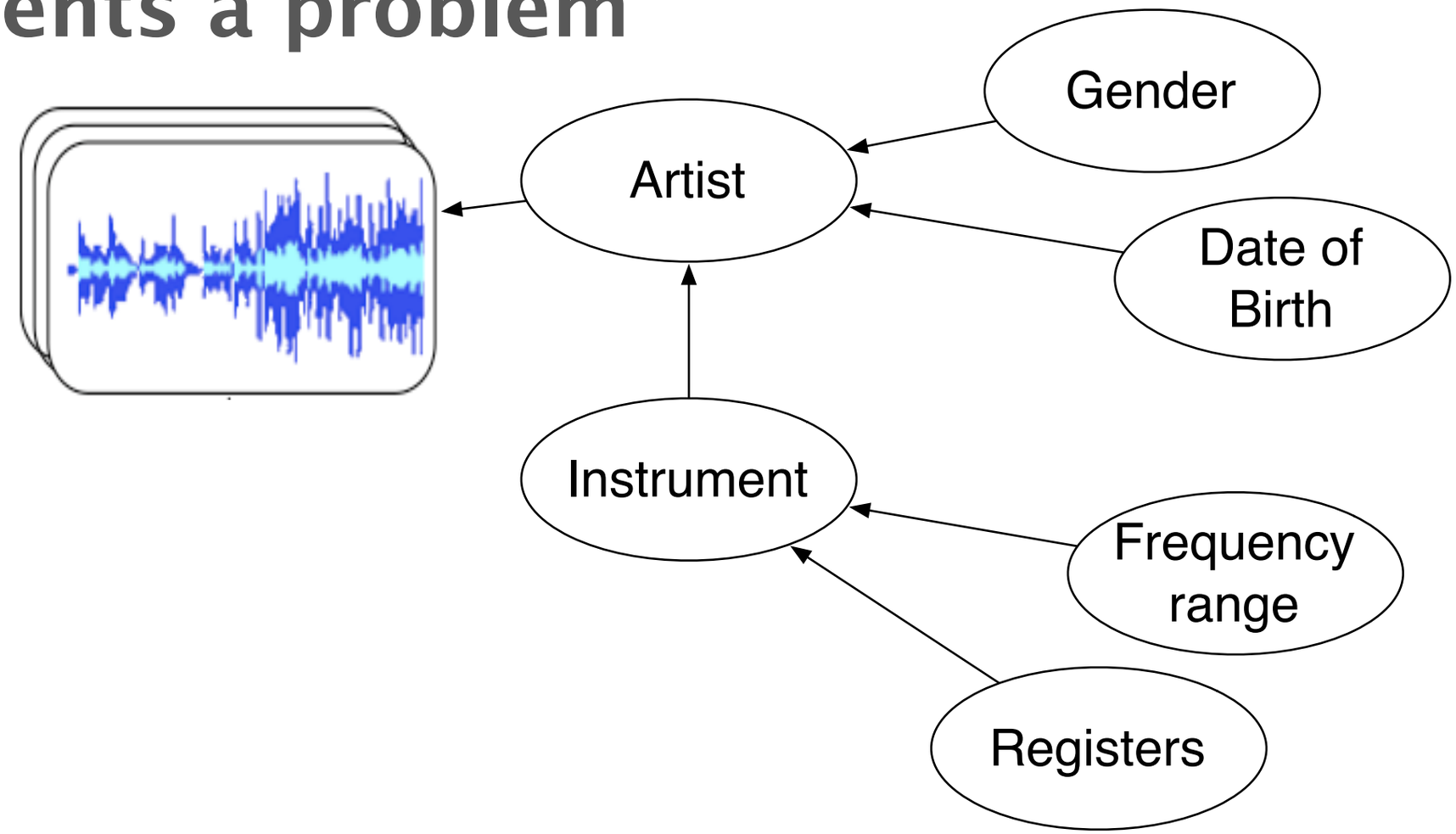
Metadata Structural Diversity

- **But, the heterogeneity of musical metadata presents a problem**



Metadata Structural Diversity

- **But, the heterogeneity of musical metadata presents a problem**



XML and Metadata Standards

- **The XML Factor:**

Image Credit: Dan Zambonini (O'Reilly XML.com blog) http://www.oreillynet.com/xml/blog/

XML and Metadata Standards

- XML and XML-based metadata standards
  - **only specify the syntax** of documents
  - **meaning** (a.k.a. semantics) is **implicit,**
  - and **hard coded** in procedural software

XML and Metadata Standards

- **The XML Factor:**

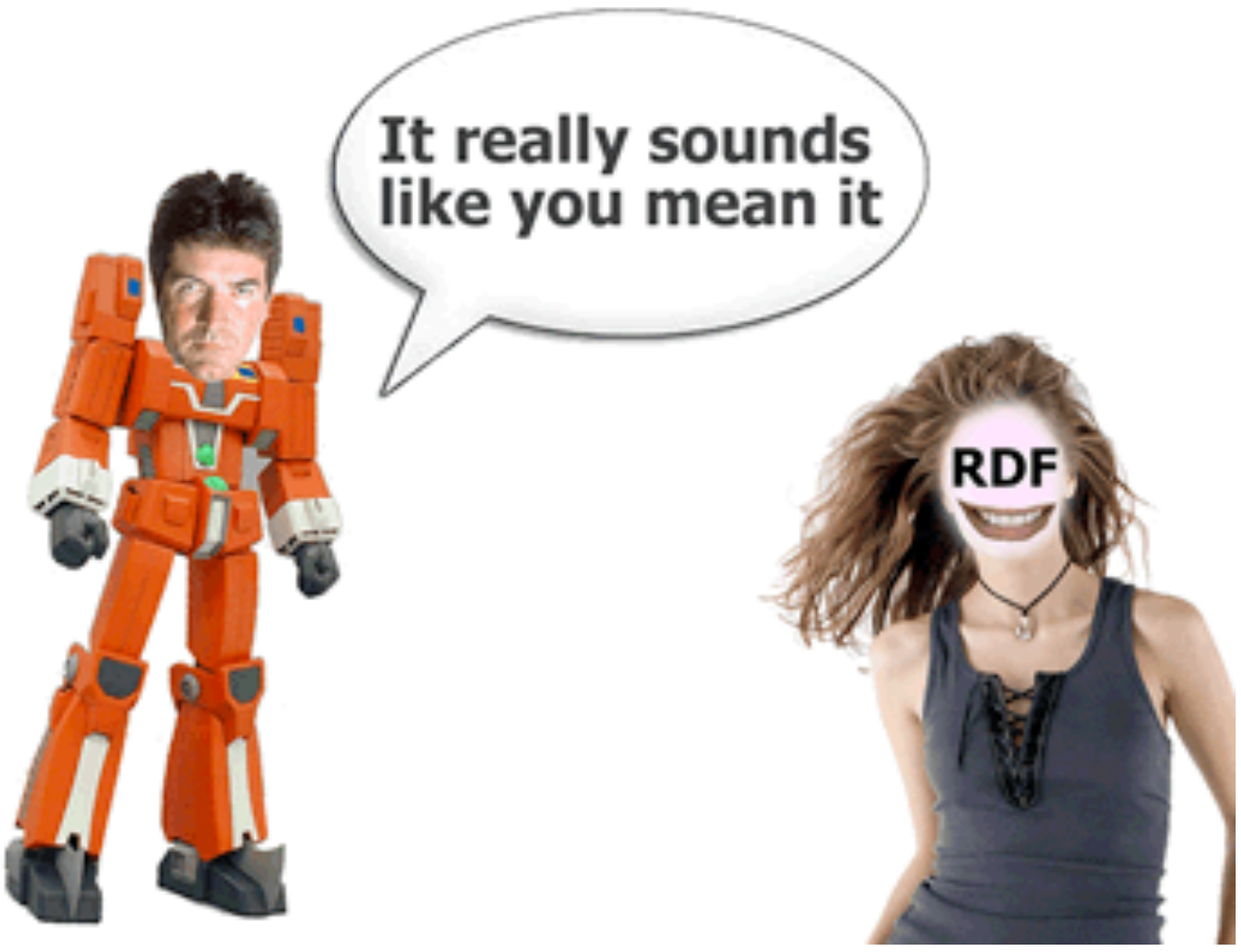


Image Credit: Dan Zambonini (O'Reilly XML.com blog) http://www.oreillynet.com/xml/blog/

XML and Metadata Standards

- **The XML Factor:**

There is no shared model of information and knowledge

Image Credit: Dan Zambonini (O'Reilly XML.com blog) http://www.oreillynet.com/xml/blog/

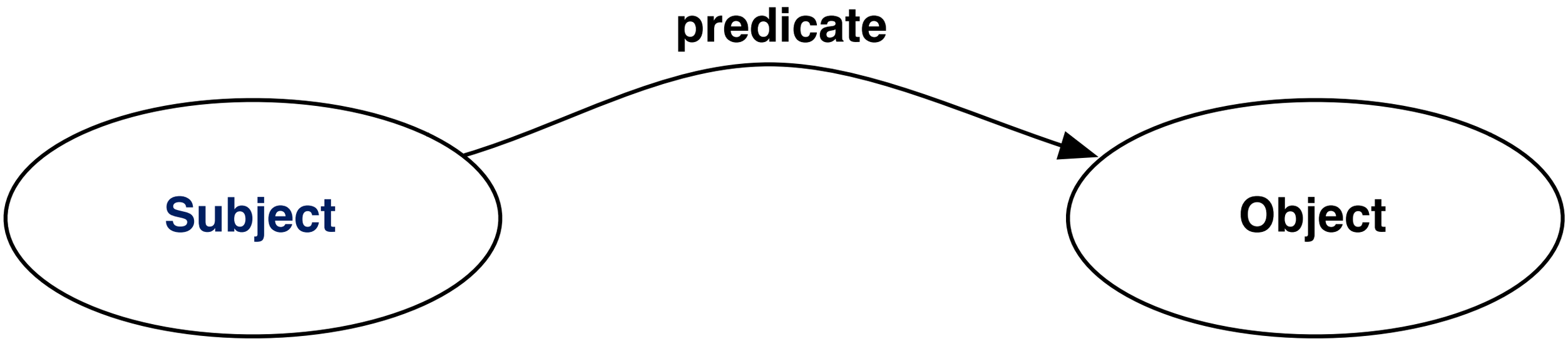
Resource Description Framework

- **RDF provides a simple model of information**

- **How does it work?**

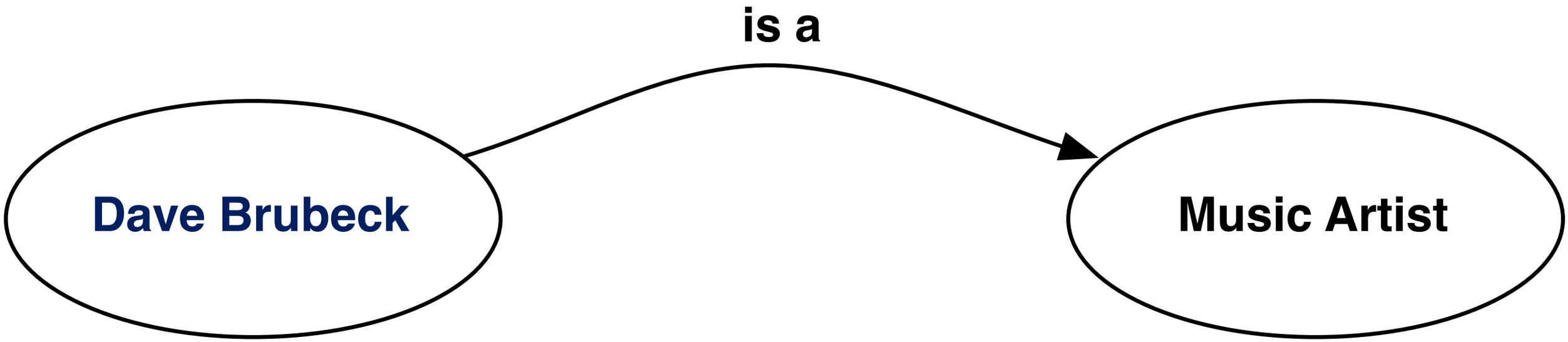
Image Credit: Dan Zambonini (O'Reilly XML.com blog) http://www.oreillynet.com/xml/blog/

Resource Description Framework



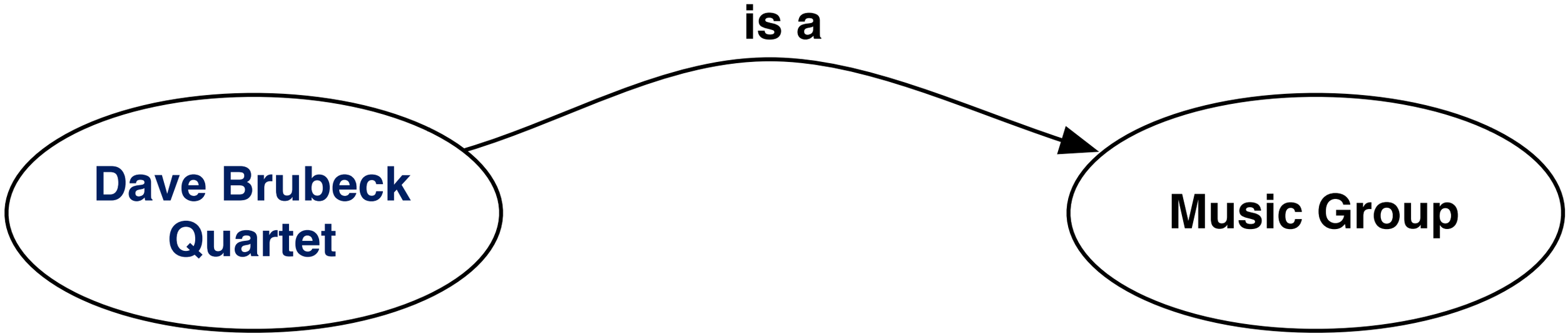
- In RDF information is decomposed into simple **statements.**

Resource Description Framework



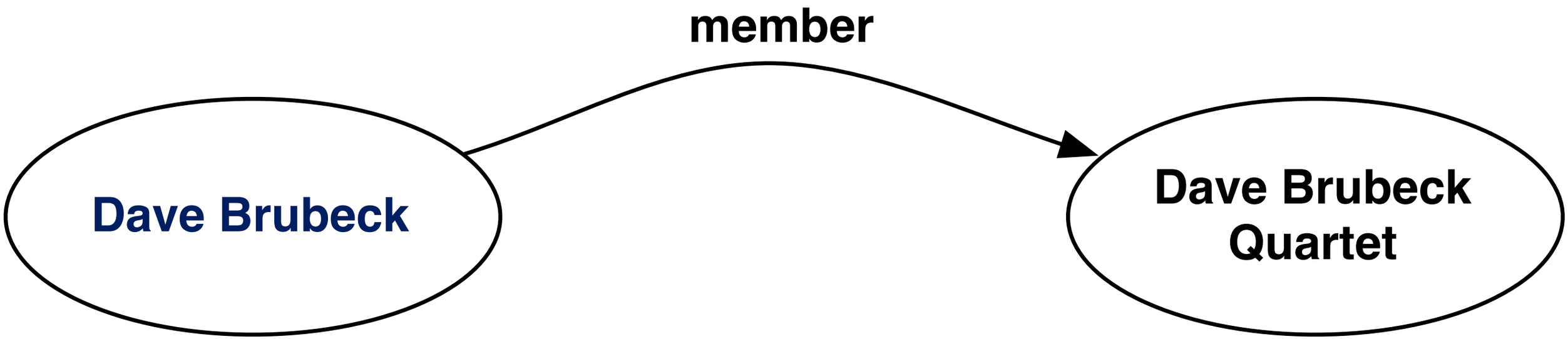
- In RDF information is decomposed into simple **statements.**

Resource Description Framework



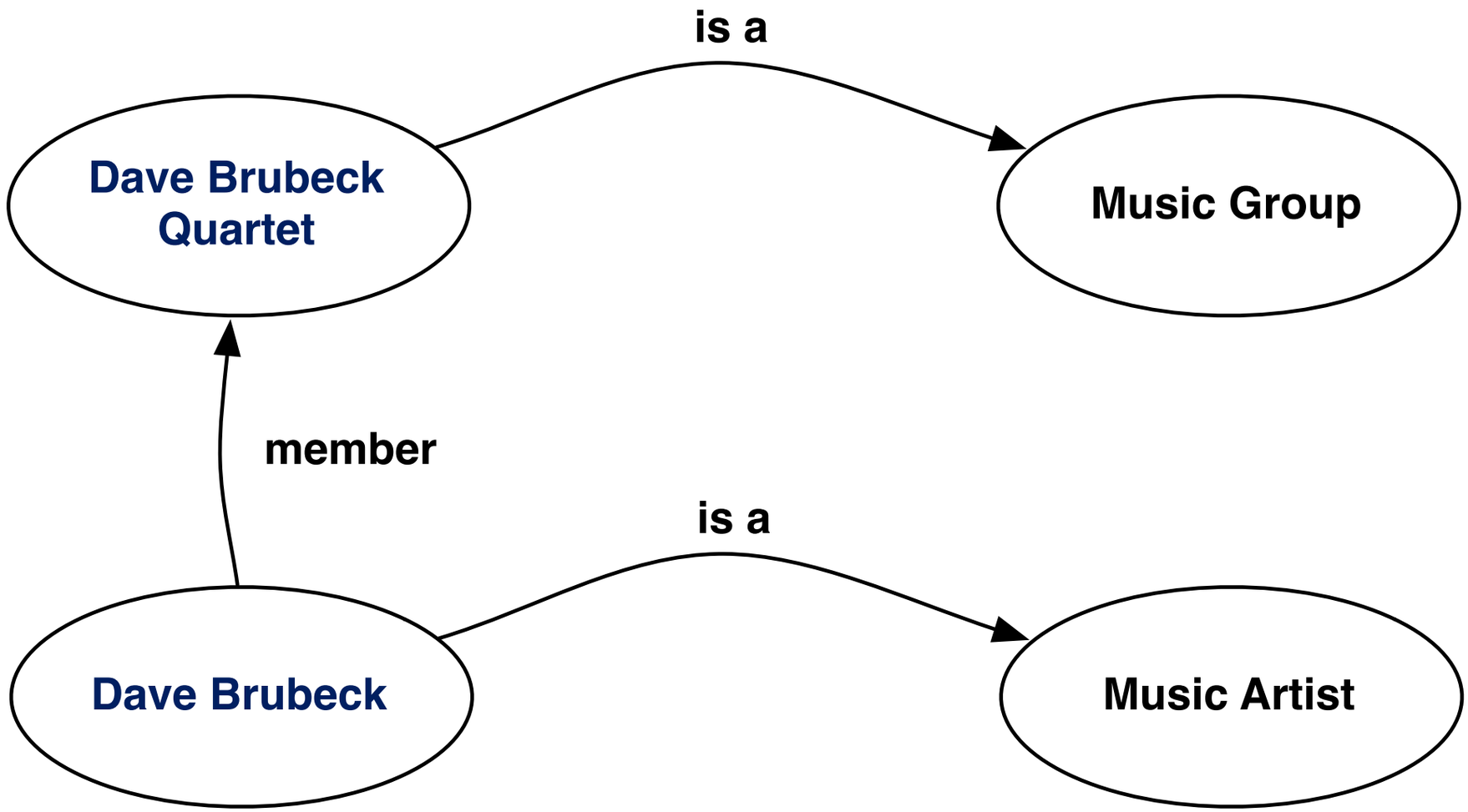
- In RDF information is decomposed into simple **statements.**

Resource Description Framework



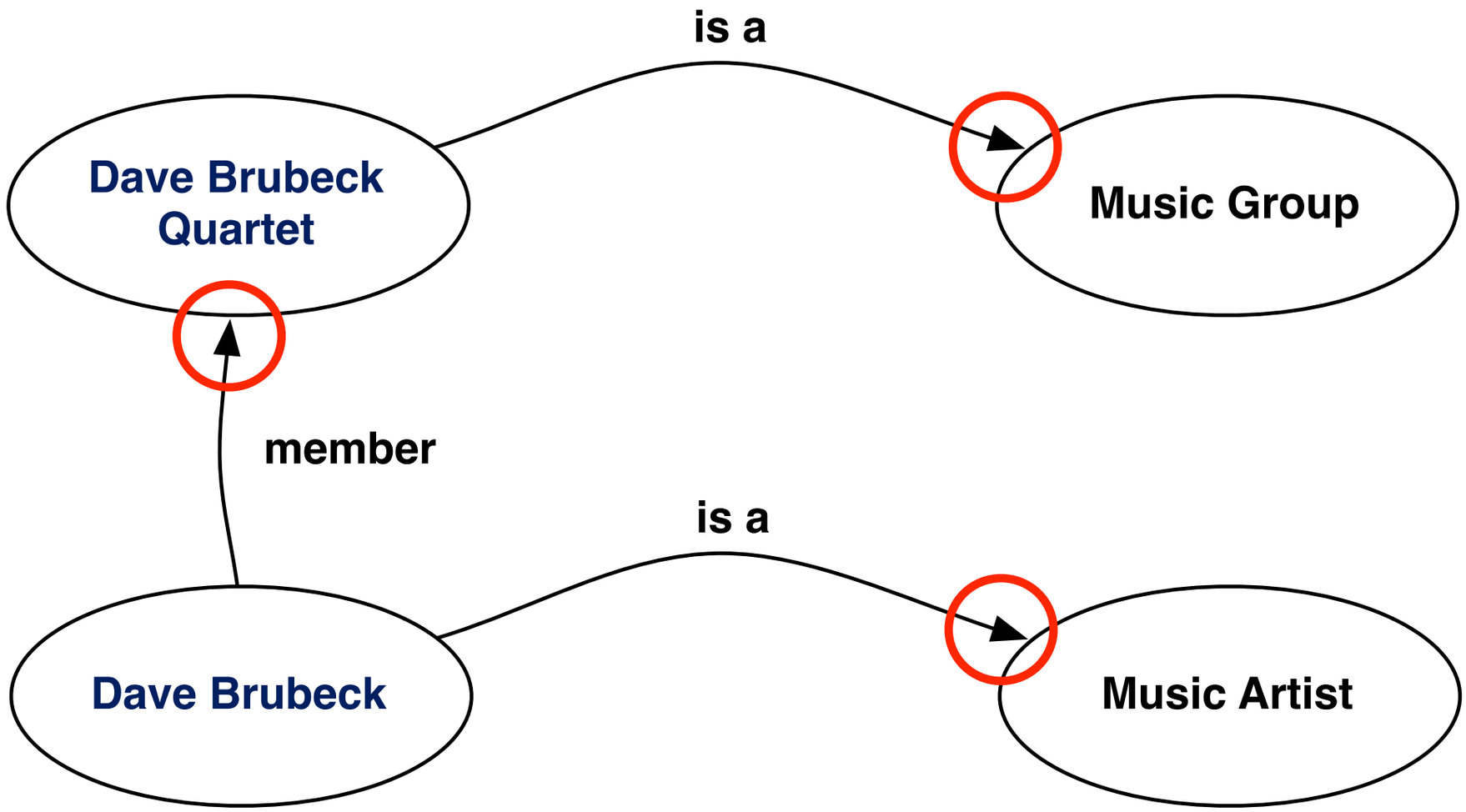
- In RDF information is decomposed into simple **statements.**

Resource Description Framework



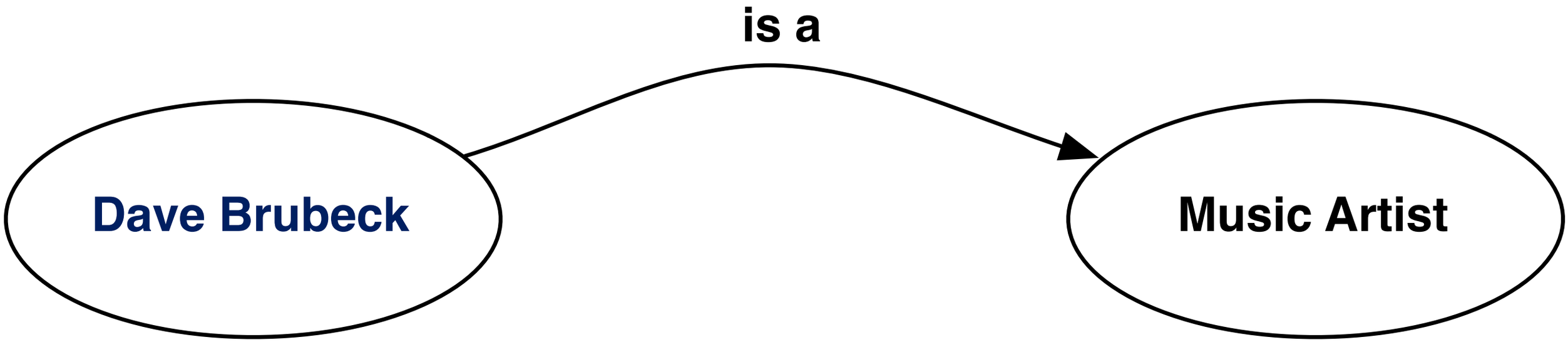
- When combined, statements form a Graph

Resource Description Framework



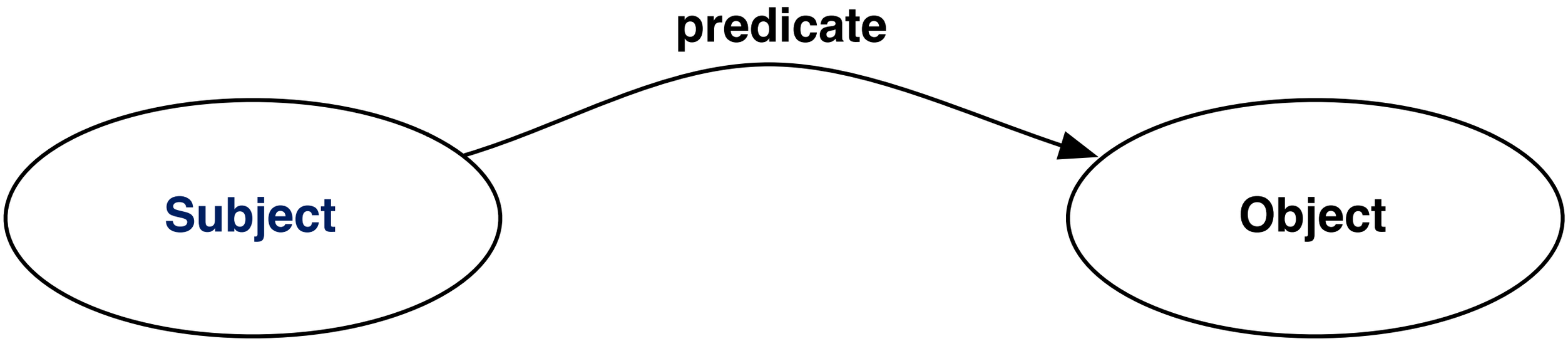
- When combined, statements form a Graph
- more precisely a **Directed Graph**

Resource Description Framework



- In RDF information is decomposed into simple **statements.**

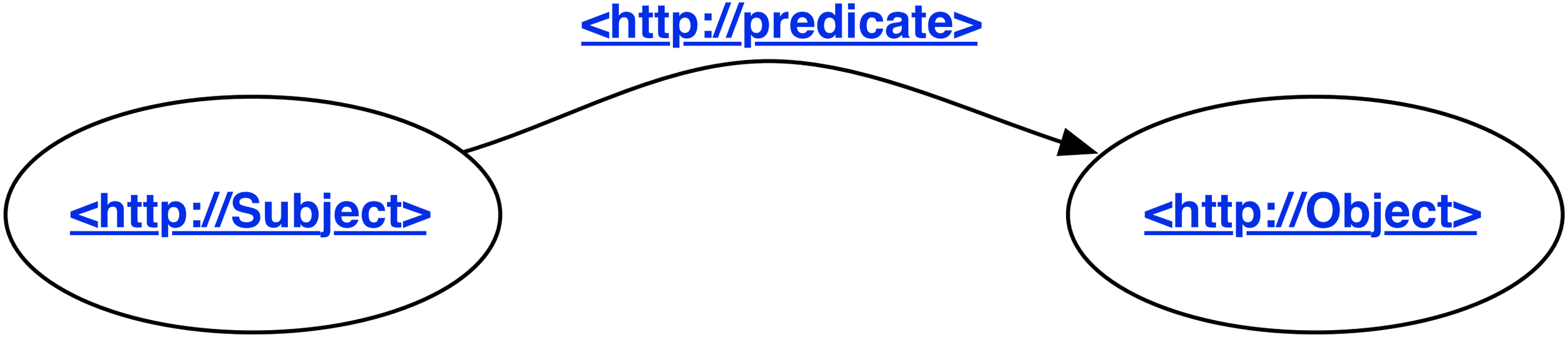
Resource Description Framework



- These statements are also called **triples**
- of terms or **resources**:
- **(subject, predicate, object).**

RDF & Unified Resource Identifiers

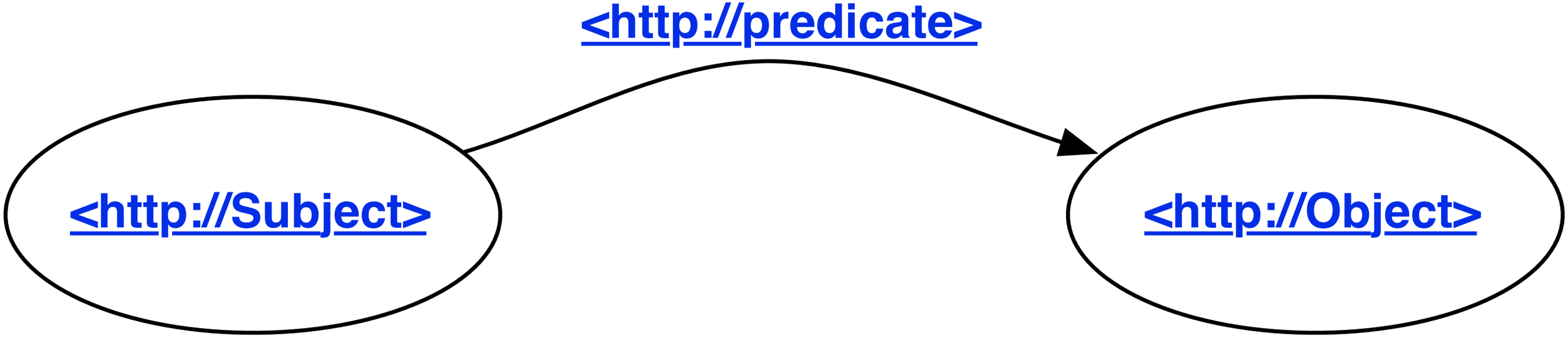
- Every term gets a **Unified Resource Identifier** (URI)



- When RDF is combined with URIs we can create a globally distributed Web of Data that
- **scales just like the World Wide Web**

RDF & Unified Resource Identifiers

- Every term gets a **Unified Resource Identifier** (URI)



- When RDF is combined with URIs we can create a globally distributed **Web of Data** that
- **scales just like the World Wide Web**

RDF & Unified Resource Identifiers

- Using Web URIs ensures that elements of RDF statements are **uniquely identified**

- We can also:
  - retrieve additional information
  - for instance, about the meaning of terms
  - store different parts of the graph at different databases / locations

RDF & Unified Resource Identifiers

- Using Web URIs ensures that elements of RDF statements are **uniquely identified**

- We can also:
  - **retrieve additional information**
  - for instance, about the meaning of terms
  - store different parts of the graph at different databases / locations

RDF & Unified Resource Identifiers

- Using Web URIs ensures that elements of RDF statements are **uniquely identified**

- We can also:
  - **retrieve additional information**
    - for instance, about the meaning of terms
    - store different parts of the graph at different databases / locations

RDF Syntax and Serialisation

- **How do we express / store information described by an RDG graph?**

<http://SUBJECT>

<http://PREDICATE>

<http://OBJECT> .

- just write down triples of URIs as sentences.
- **this is called N-Triples.**

RDF Syntax and Serialisation

- **How do we express / store information described by an RDG graph?**

```
<http://SUBJECT>
<http://PREDICATE>
<http://OBJECT> .
```

- just write down triples of URIs as sentences.
- this is called N-Triples.

RDF Syntax and Serialisation

- **How do we express / store information described by an RDG graph?**

```
<http://SUBJECT>
<http://PREDICATE>
<http://OBJECT> .
```

- just write down triples of URIs as sentences.
- **this is called N-Triples.**

RDF Syntax and Serialisation

- **RDF is not RDF/XML !**

- XML was the first standardised syntax for RDF,

but

- **there are many others available that are:**
  - easier to use
  - easier to read (by a human)
  - easier to parse (by a machine)
  - more concise

RDF Syntax and Serialisation

- **RDF is not RDF/XML !**

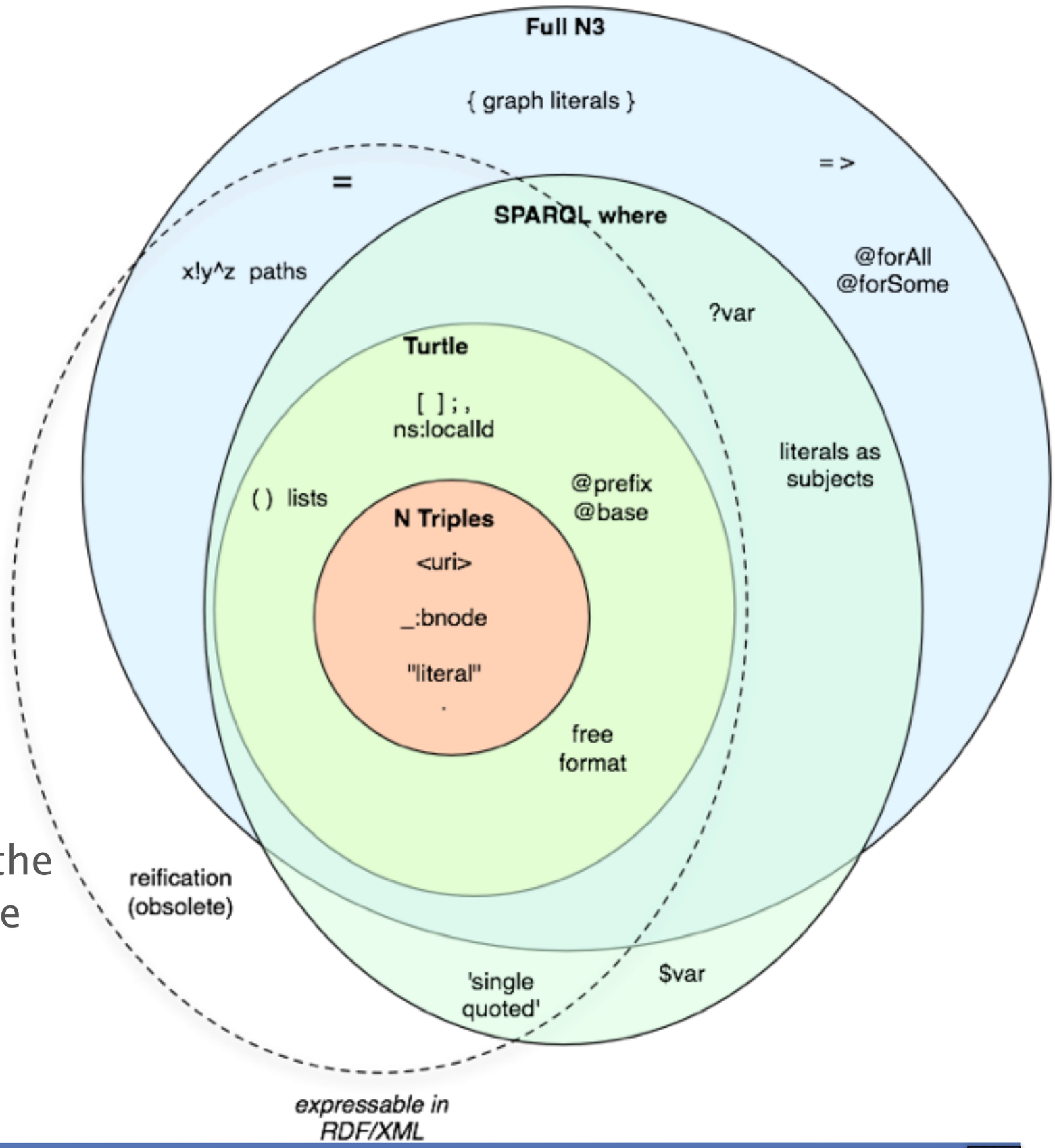
- XML was the first standardised syntax for RDF,

but

- **there are many others available that are:**
  - easier to use
  - easier to read (by a human)
  - easier to parse (by a machine)
  - more concise

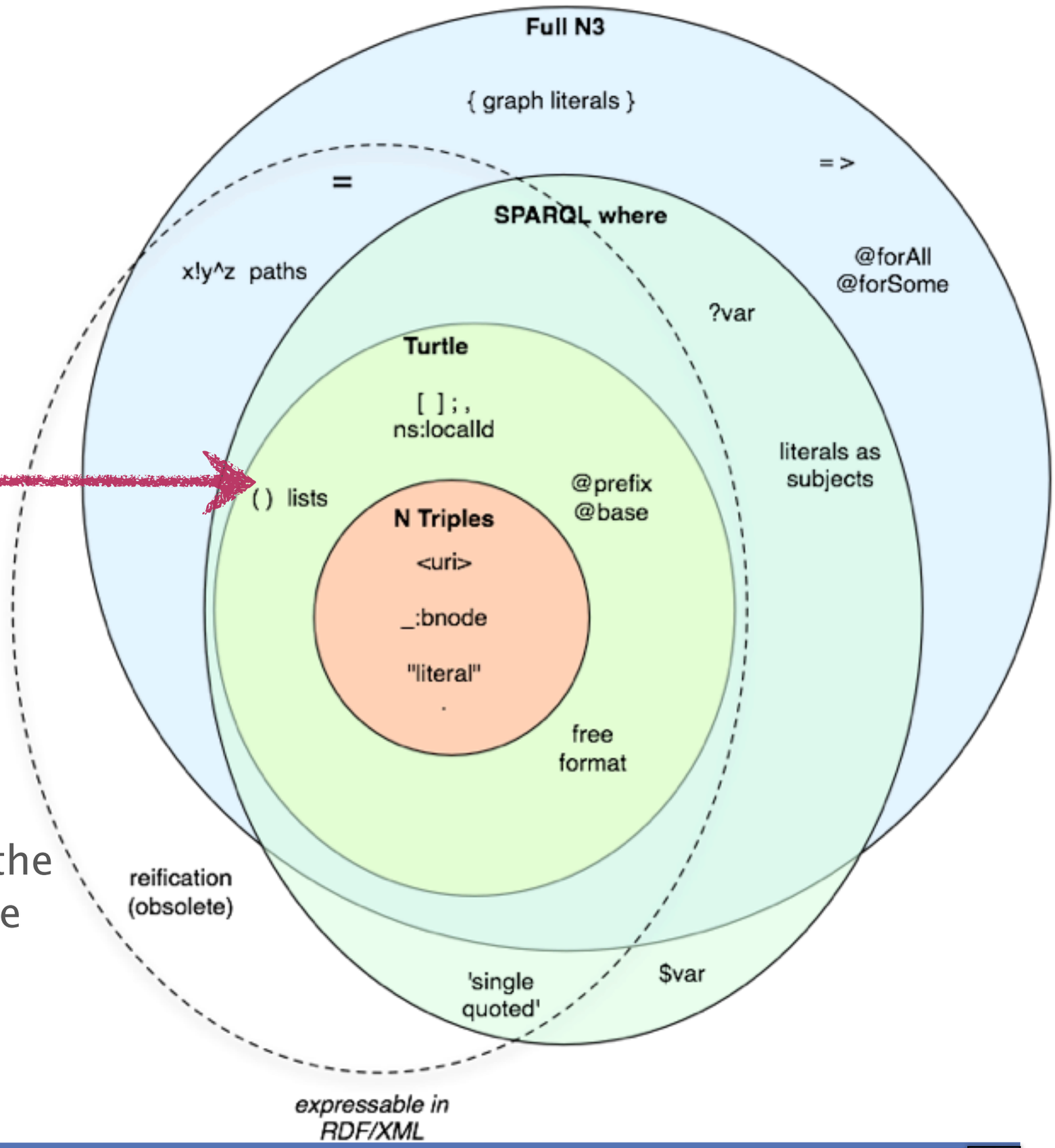
RDF Syntax

- **Some common syntaxes:**
- N Triples
- **Turtle**
- RDF/XML
- RDFa
- JSON-LD
- N3 (this goes beyond the RDF model and the scope of this tutorial)



RDF Syntax

- **Some common syntaxes:**
- N Triples
- **Turtle**
- RDF/XML
- RDFa
- JSON-LD
- N3 (this goes beyond the RDF model and the scope of this tutorial)



RDF N Triples Syntax

<http://dbpedia.org/resource/Dave_Brubeck>

<http://dbpedia.org/ontology/genre>

<http://dbpedia.org/resource/Cool_Jazz> .

- Here is a statement in **N Triples**.

RDF Turtle Syntax

```
@prefix dbpr: <http://dbpedia.org/resource/> .
@prefix dbpo: <http://dbpedia.org/ontology/genre> .

dbpr:Dave_Brubeck dbpo:genre dbpr:Cool_Jazz .
```

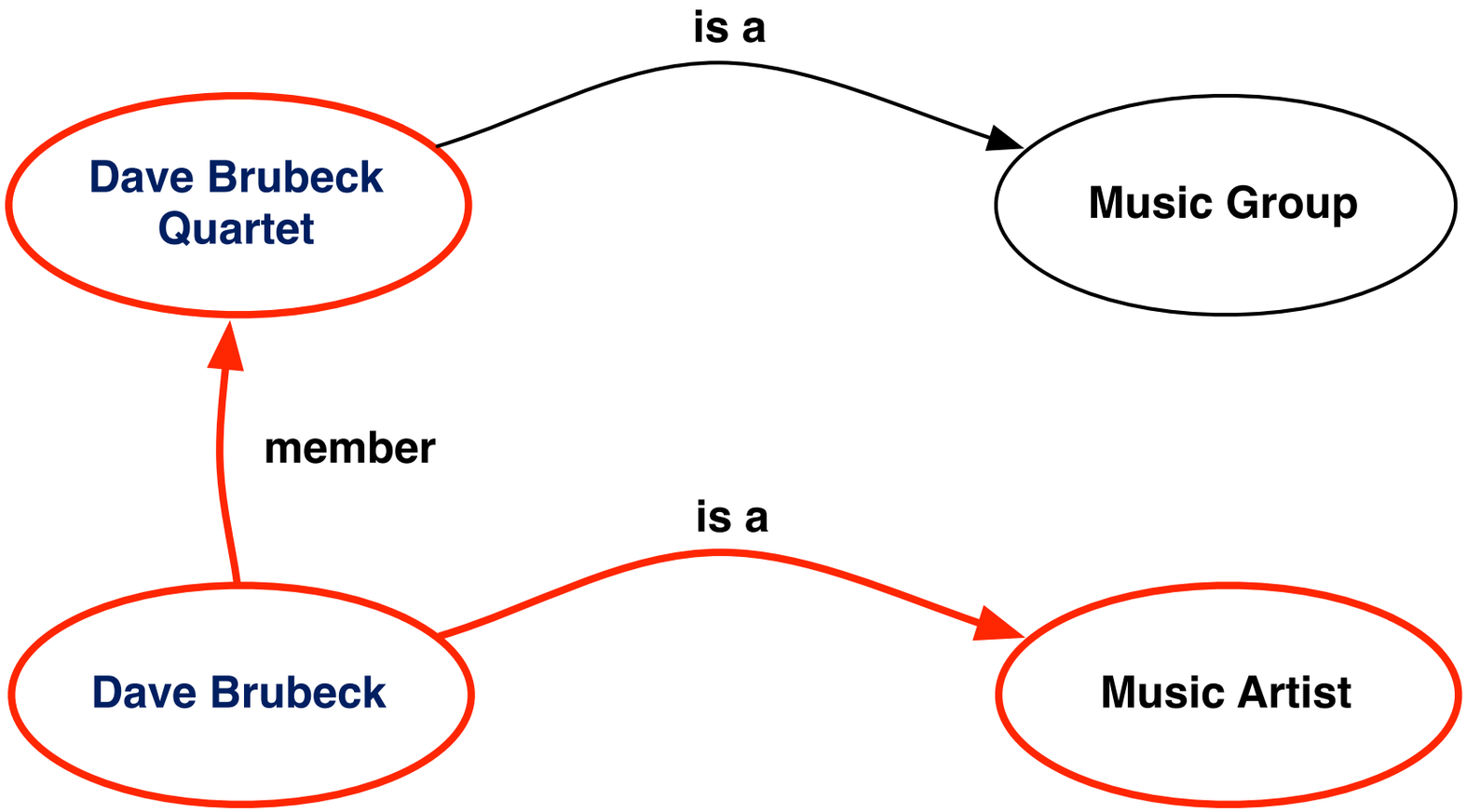
- Using **CURIEs** and the prefix notation.
- Still 3 lines of RDF but given a large set of statements this is a significant reduction.

RDF Turtle Syntax

```
@prefix dbpr: <http://dbpedia.org/resource/> .
@prefix dbpo: <http://dbpedia.org/ontology/genre> .

dbpr:Dave_Brubeck dbpo:genre dbpr:Cool_Jazz .
```

- Using **CURIEs** and the prefix notation.
- Still 3 lines of RDF but given a large set of statements this is a significant reduction.

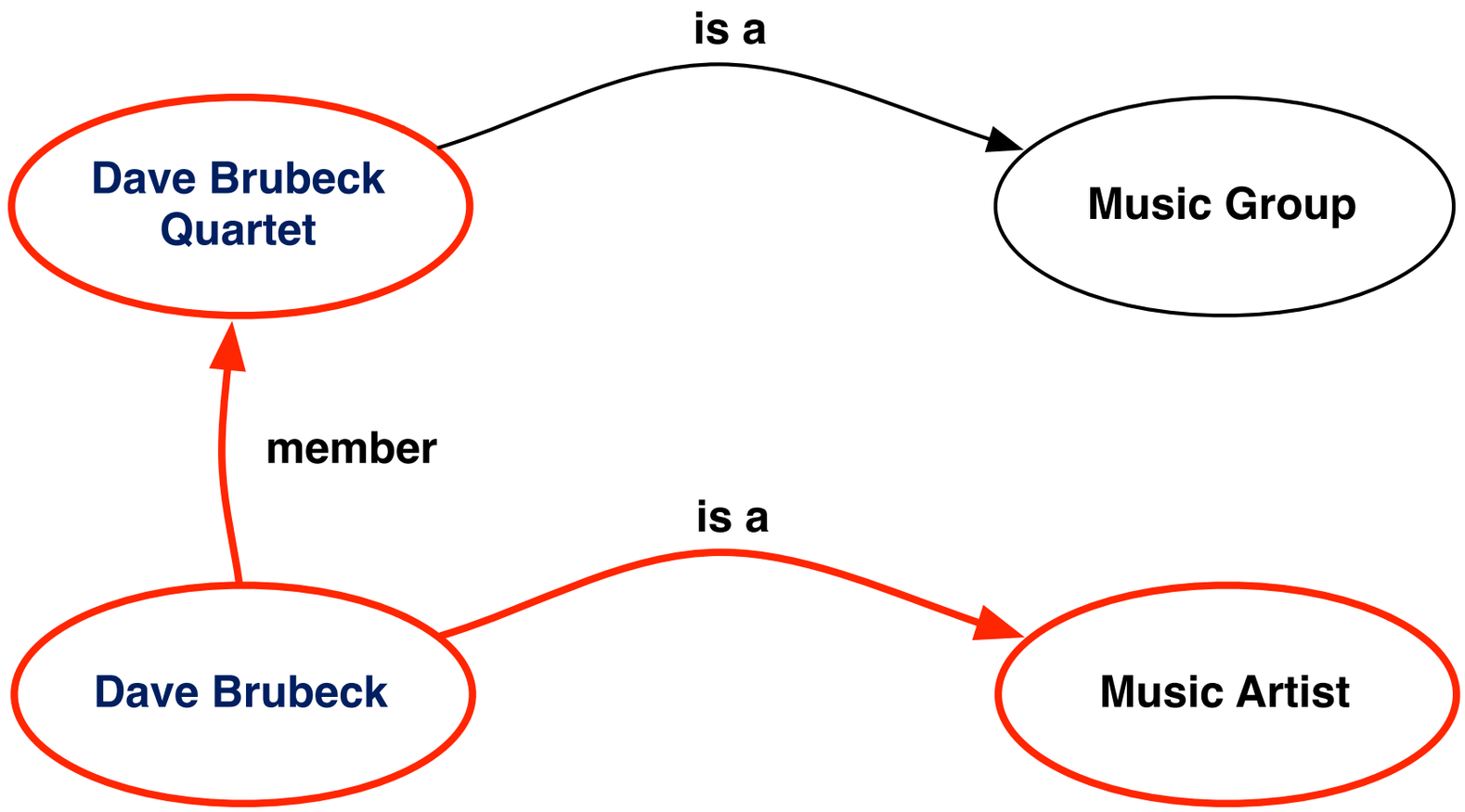
RDF Turtle Syntax




```
:Dave_Brubeck rtf:type mo:MusicArtist ;
    :member :Dave_Brubeck_Quartet.
```

- The prefix can remain empty (:resource) to represent the local scope.

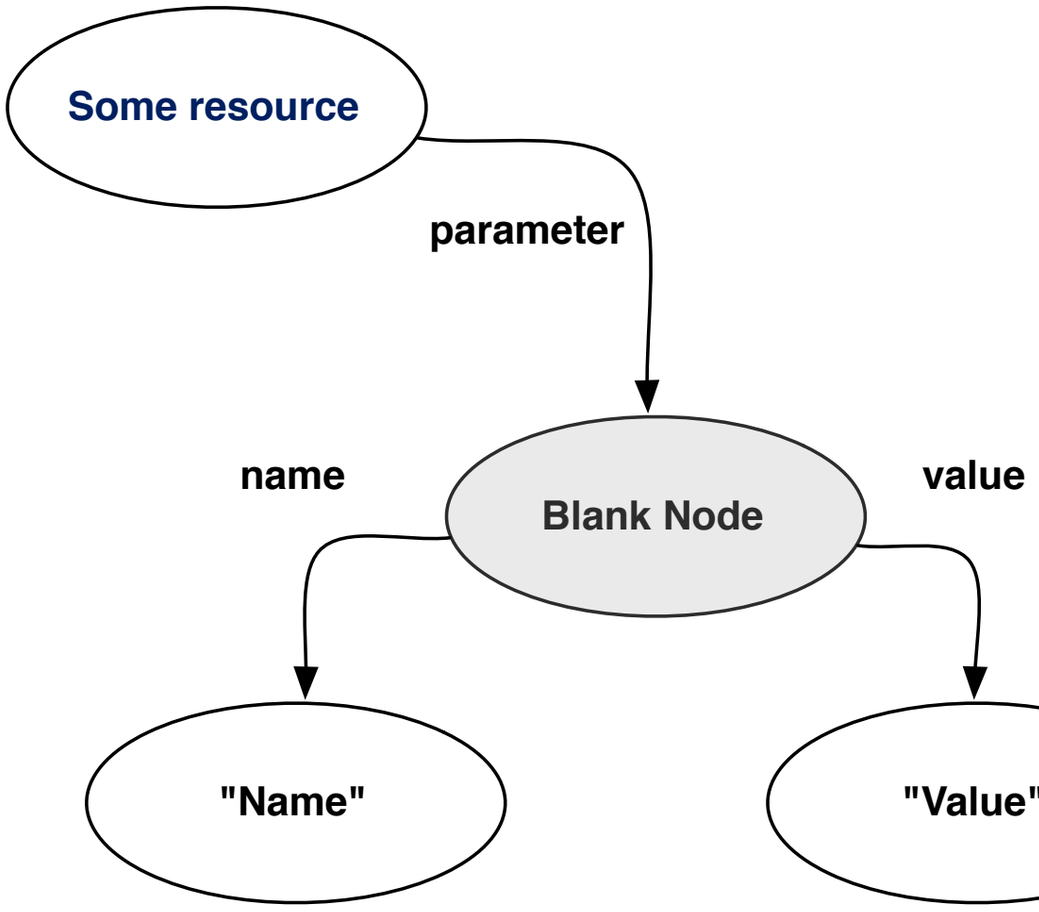
RDF Turtle Syntax




```
:Dave_Brubeck rtf:type mo:MusicArtist ;
    :member :Dave_Brubeck_Quartet.
```

- The **semicolon** can be used to group statements about the same resource.

RDF Turtle Syntax: Blank nodes




```
:resource [
    :name “parameter name” ;
    :value “20”
    ] .
```

- **Blank nodes represent unnamed resources**
- They are very useful when representing complex data

Linking different datasets

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix mo: <http://purl.org/ontology/mo/> .

<http://www.bbc.co.uk/music/artists/1545000730-525f-4ed5-
aaa8-92888-f060f5f#artist>
    rdf:type mo:MusicArtist ;
    owl:sameAs <http://dbpedia.org/resource/Dave_Brubeck> .
```

- **owl:sameAs** predicate can be used to link resources in different datasets that hold information about the same resource.

RDF Turtle Syntax: Summary

- URIs : <**http://some_resource.org**>
- CURIEs: **mo:MusicArtist**
- @prefix: declare namespaces
- Blank nodes: [...] **or _:bnode**
- Literal values: **“some string”**
- Typed literals: **“20”^^xsd:int**
- Group statements: semicolon (**;**)
- Group objects: colon (**,**)
- Close statements: dot (**.**)
- Shorthand for rdf:type: **a**

RDF Storage and Databases

- **Linked data repositories**
  - **use eg. HTTP GET**
  - this is usually done through content negotiation
- **Triple Stores**
  - Garlic's 4Store
  - Openlink Virtuoso
- Lots of programming libraries
  - redland (C), rdflib (Python), Jena (Java)

RDF Storage and Databases

- **Linked data repositories**
  - **use eg. HTTP GET**
  - this is usually done through content negotiation

- **Triple Stores**
  - Garlic's 4Store
  - Openlink Virtuoso
- Lots of programming libraries
  - redland (C), rdflib (Python), Jena (Java)

Querying RDF with SPARQL

- **SPARQL** protocol and RDF Query Language
- Similar to Turtle
- It has several query types, e.g.
  - SELECT
  - CONSTRUCT
- variables: **?x**
  - These allow to form query patterns to be matched

Querying RDF with SPARQL

- **SPARQL** protocol and RDF Query Language
- Similar to Turtle
- It has several query types, e.g.
  - SELECT
  - CONSTRUCT
- variables: **?x**
- These allow to form query patterns to be matched

Querying RDF with SPARQL

- **SPARQL** protocol and RDF Query Language
- Similar to Turtle
- It has several query types, e.g.
  - SELECT
  - CONSTRUCT
- variables: **?x**
  - These allow to form query patterns to be matched

Querying RDF with SPARQL

```
PREFIX dbpr: <http://dbpedia.org/resource/>
PREFIX dbpo: <http://dbpedia.org/ontology/>

SELECT ?genre
WHERE {
dbpr:Dave_Brubeck dbpo:genre ?genre .
}
```

- Find a genre classification according to DBPedia

Querying RDF with SPARQL

```
PREFIX dbpr: <http://dbpedia.org/resource/>
PREFIX dbpo: <http://dbpedia.org/ontology/>

SELECT ?x
WHERE {
dbpr:Dave_Brubeck dbpo:genre ?genre .
?x dbpo:genre ?genre .
}
```

- Find other artists (?x) having the same genre

Querying RDF with SPARQL

```
PREFIX dbpr: <http://dbpedia.org/resource/>
PREFIX dbpo: <http://dbpedia.org/ontology/>

SELECT ?x
WHERE {
dbpr:Dave_Brubeck dbpo:genre ?genre .
?x dbpo:genre ?genre .
}
```

- Let's try this in practice

Linked Data Services

- There are many music related linked data services and applications available
- **DBTube.org**
  - http://dbtune.org/
- **Linked Brainz** (MusicBrainz database)
  - http://linkedbrainz.c4dmpresents.org/
- **Musicnet**
  - http://musicnet.mspace.fm/
- **BBC Music website**
  - http://www.bbc.co.uk/music

Ontologies

RDF and Ontologies

- Elements of RDF statements can be selected in an ad-hoc manner.

- We need a way to **give "meaning"** to each subject, predicate and object.

- **Represent knowledge in a formal way.**

RDF and Ontologies

- Elements of RDF statements can be selected in an ad-hoc manner.

- We need a way to **give "meaning"** to each subject, predicate and object.

- **Represent knowledge in a formal way.**

Knowledge Representation

- This can be done using **First Order Logic**

$$\forall x(\mathsf{AudioClip}(x) \leftrightarrow \exists y(\mathsf{hasSignal}(x, y) \wedge \mathsf{Signal}(y)))$$

$$\exists x, y(\mathsf{AudioClip}(x) \wedge \mathsf{tempo}(x, 120)$$
$$\wedge \mathsf{name}(x, myrecording)$$
$$\wedge \mathsf{hasSignal}(x, y) \wedge \mathsf{Signal}(y))$$

- But this is too hard for practical reasoning

- **Description Logics** are subsets of this logic that provide the logical foundations for **Web Ontologies and Ontology languages**

Knowledge Representation

- This can be done using **First Order Logic**

$$\forall x(\mathsf{AudioClip}(x) \leftrightarrow \exists y(\mathsf{hasSignal}(x, y) \wedge \mathsf{Signal}(y)))$$

$$\exists x, y(\mathsf{AudioClip}(x) \wedge \mathsf{tempo}(x, 120)$$
$$\wedge \mathsf{name}(x, myrecording)$$
$$\wedge \mathsf{hasSignal}(x, y) \wedge \mathsf{Signal}(y))$$

- But this is too hard for practical reasoning

- **Description Logics** are subsets of this logic that provide the logical foundations for **Web Ontologies and Ontology languages**

Ontologies

- An Ontology is:

- **a shared conceptualisation of a world or domain**

- it includes:
  - 1) individuals,
  - 2) classes, groups of individuals that have something in common,
  - 3) possible relationships between them

Ontologies

- An Ontology is:

- **a shared conceptualisation of a world or domain**

- it includes:
  - 1) **individuals**,
  - 2) **classes**, groups of individuals that have something in common,
  - 3) **relationships** possible between them

Ontologies

- There might be many ontologies for the same domain
- They all may be valid (and useful),
- but it is unlikely they cover everything,
- or equally useful in all applications.

Ontologies

- There might be many ontologies for the same domain

- They all may be valid (and useful),

- but it is unlikely they cover everything,

- or equally useful in all applications.

Ontology Languages

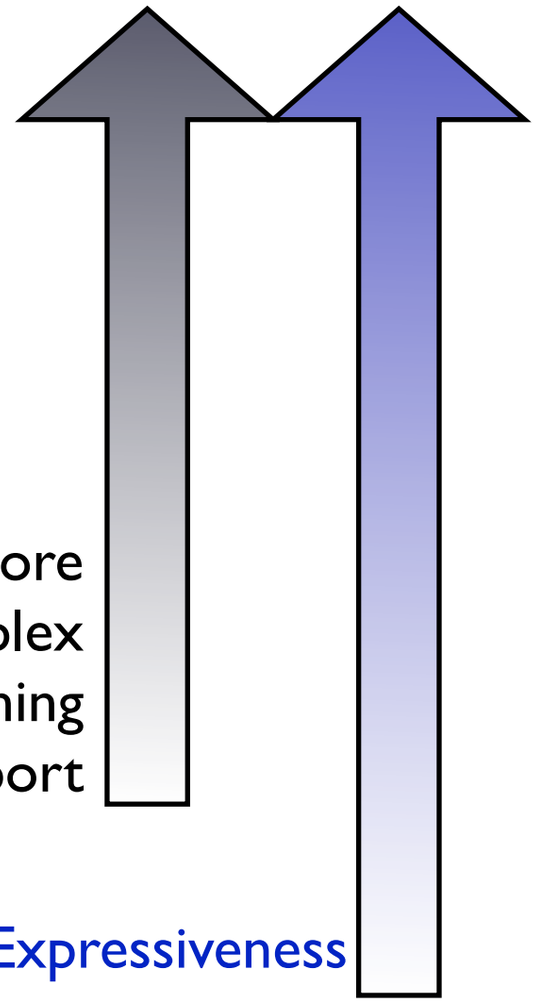
- There is a stack of languages (W3C recommendations)

- **OWL2**: extended data model
- **OWL**: allows for equivalence, cardinality constraints, etc...
  - OWL-Full
  - OWL-DL
  - OWL-Lite
- **RDFS**: allows for describing class and property hierarchies

Ontology Languages

- There is a stack of languages (W3C recommendations)

- **OWL2**: extended data model
- **OWL**: allows for equivalence, cardinality constraints, etc...
  - OWL-Full
  - OWL-DL
  - OWL-Lite
- **RDFS**: allows for describing class and property hierarchies



The Music Domain

- To describe music we need to communicate:

- **editorial (bibliographic) information**
- **information about intellectual works and workflows**
  - people and their works
- **cultural and social information**
- **content-based information**
- **provenance and trust**
  - who says what and can we trust it?

The Music Domain

- To describe music we need to communicate:

- **editorial (bibliographic) information**
- **information about intellectual works and workflows**
  - people and their works
- **cultural and social information**
- **content-based information**
- **provenance and trust**
  - who says what and can we trust it?

Some useful ontologies

- **Dublin Core**
- **Friend of a Friend** (FOAF) vocabulary
  - to talk about people, groups, and
- **OWL-Time**:
  - basic temporal concepts
- **Timeline Ontology**:
  - relate temporal concepts with regards to different timelines
- **Event Ontology**:
  - describe time based events

The Music Ontology

The Music Ontology

- **Combines several ontologies to describe music related information**

```
mo:MusicArtist
        rdf:type owl:Class ;
        rdfs:comment """A person or a group of people (or
        a computer, whose musical creative work shows
        sensitivity and imagination """ ;
        rdfs:isDefinedBy <http://purl.org/ontology/mo/>;
        rdfs:label "music artist" ;
        rdfs:subClassOf foaf:Agent .
```

Credit: Yves Raimond et al, http://musicontology.com/

The Music Ontology

- Combines several ontologies to describe music related information

```
mo:MusicArtist
        rdf:type owl:Class ;
       rdfs:comment """A person or a group of people (or
a computer, whose musical creative work shows
sensitivity and imagination """ ;
     rdfs:isDefinedBy <http://purl.org/ontology/mo/>;
     rdfs:label "music artist" ;
     rdfs:subClassOf foaf:Agent .
```

Credit: Yves Raimond et al, http://musicontology.com/

Timeline and Event Ontologies

- **The Timeline Ontology extends OWL-Time and defines the TimeLine concept.**
- Temporal objects (signal, video, performance, work, etc.) can be associated with a timeline.

- **The Event ontology relates arbitrary events to:**
  - temporal entities
  - geographical coordinates
  - participating agents
  - passive factors (such as tools)
  - and products (results of an event)
  - allows to decompose complex events into sub-events

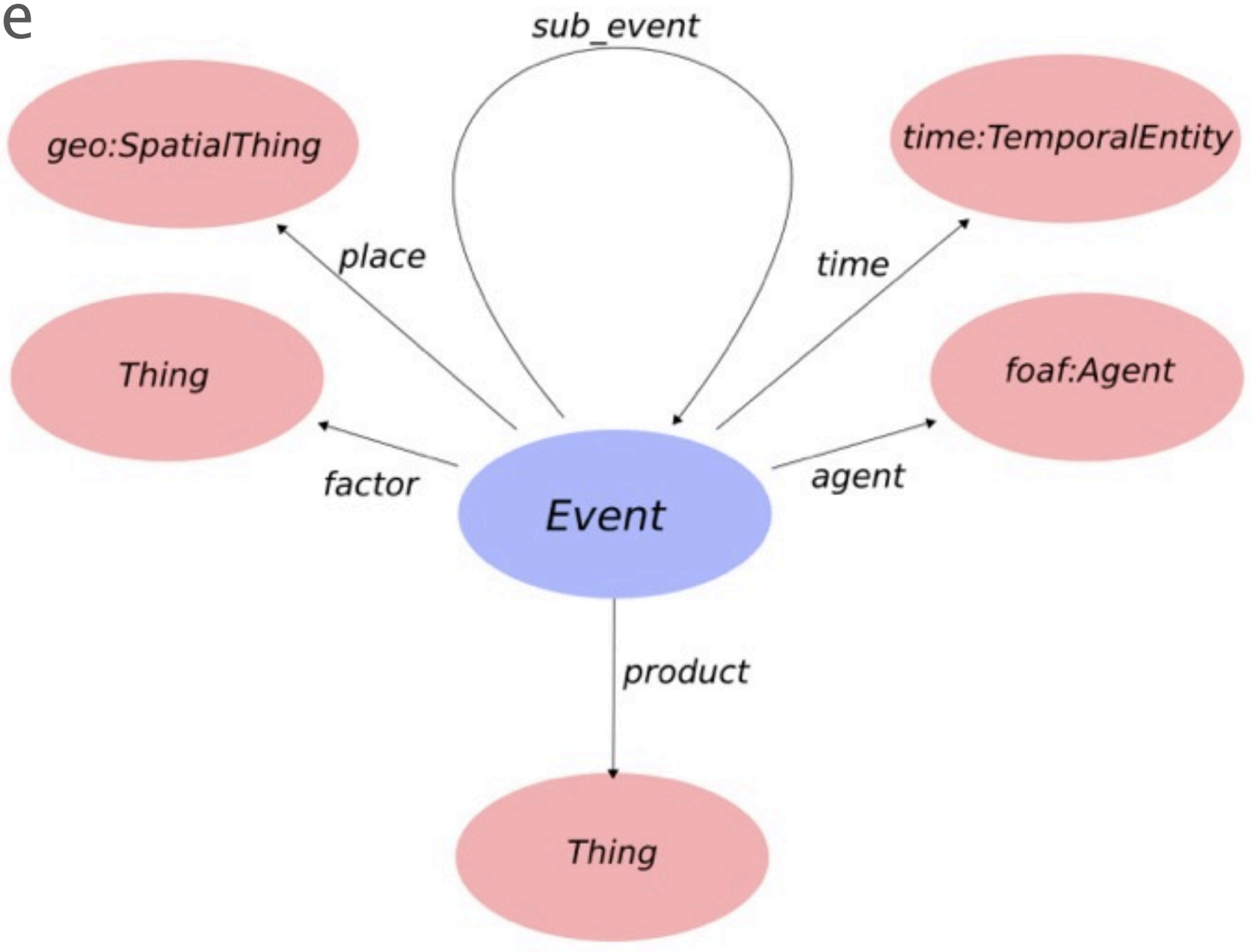
Timeline and Event Ontologies

- **The Timeline Ontology extends OWL-Time and defines the TimeLine concept.**
- Temporal objects (signal, video, performance, work, etc.) can be associated with a timeline.

- **The Event ontology relates arbitrary events to:**
  - temporal entities
  - geographical coordinates
  - participating agents
  - passive factors (such as tools)
  - and products (results of an event)
  - allows to decompose complex events into sub-events

Timeline and Event Ontologies

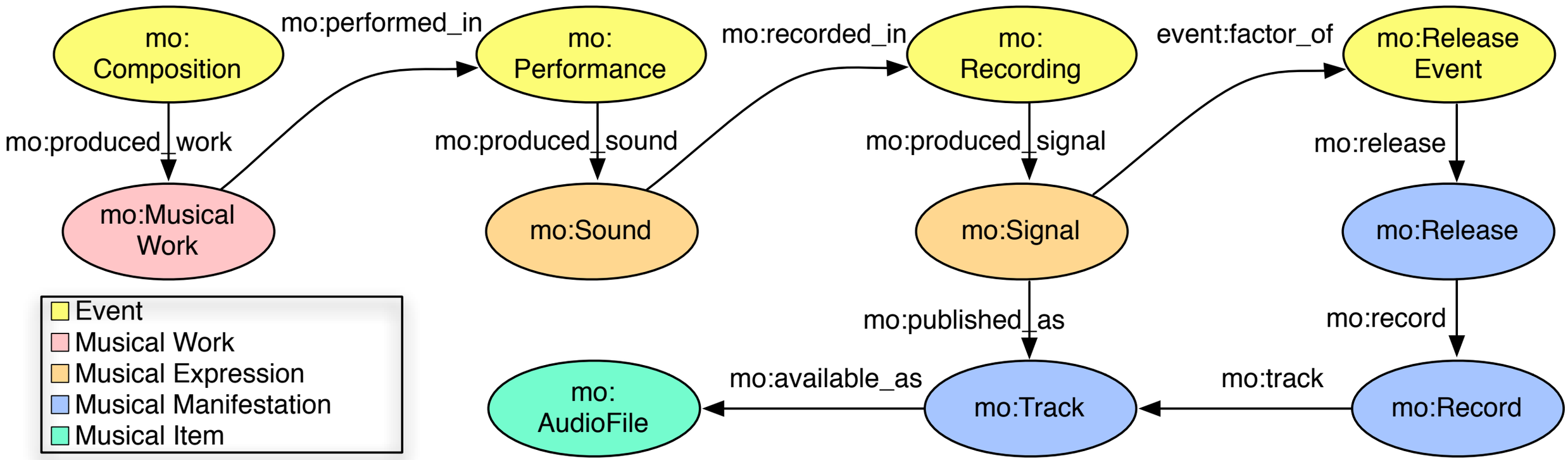
- An event may be (for instance):
- a concert,
- a performance or
- a note onset



http://purl.org/NET/c4dm/event.owl#

The Music Ontology

- Defines a Music Production Workflow Model



The Music Ontology

- A large set of extensions are available, including:

- **The Audio Features Ontology**

- **The Chord ontology**

- **The Studio Ontology**

The Music Ontology

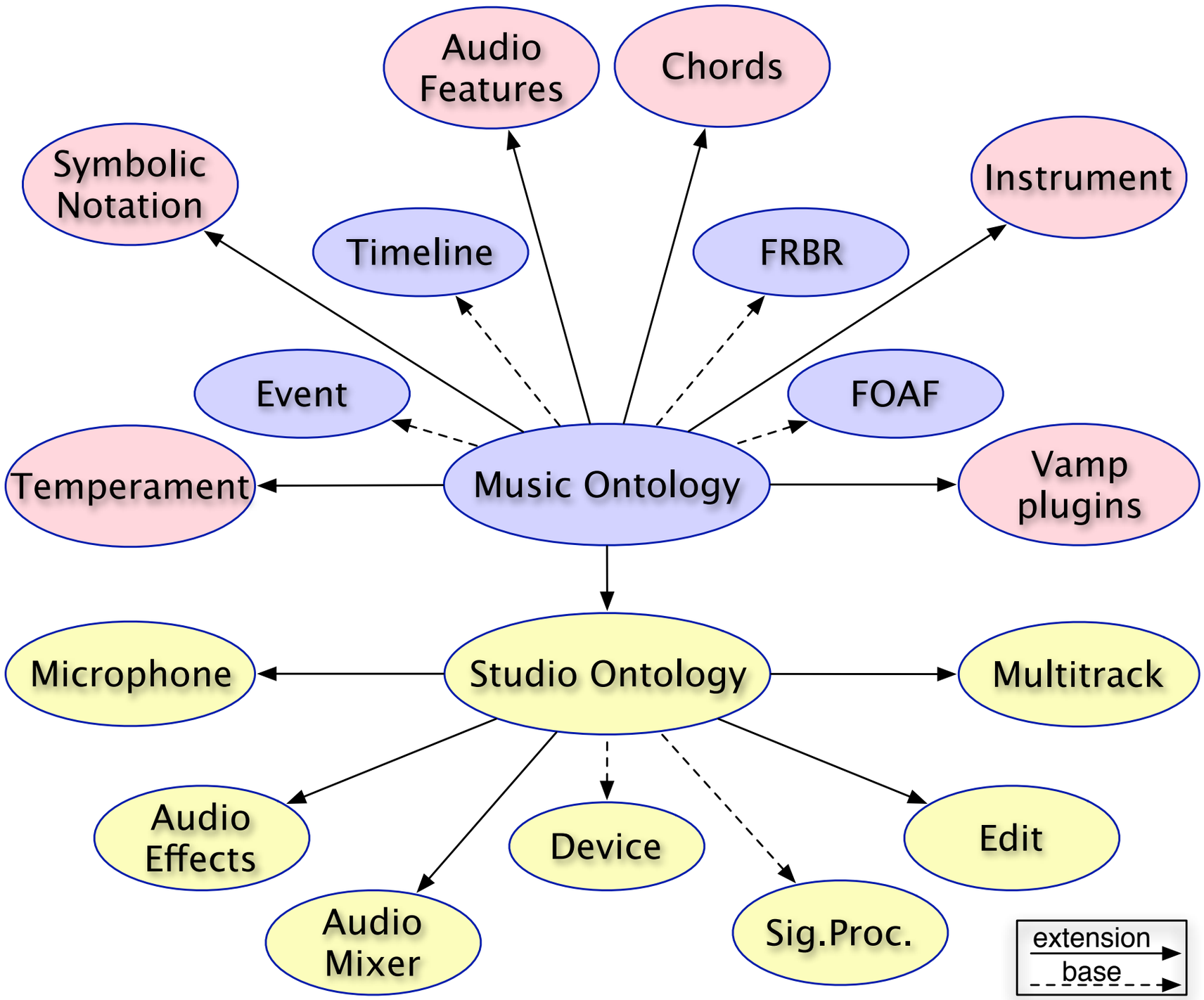
- A large set of extensions are available, including:

- **The Audio Features Ontology**

- **The Chord ontology**

- **The Studio Ontology**

Ontology Frameworks



The Studio Ontology

The Studio Ontology

- Enables collecting information about audio production.

- Motivations
- Notation for capturing the contribution of the engineer to creative work
- Improved Information and workflow management in the studio
- Exploit music production data in MIR systems
- Enable building intelligent music production systems

The Studio Ontology

- Enables collecting information about audio production.

- **Motivations**
- Notation for capturing the contribution of the engineer to creative work
- Improved Information and workflow management in the studio
- Exploit music production data in MIR systems
- Enable building intelligent music production systems

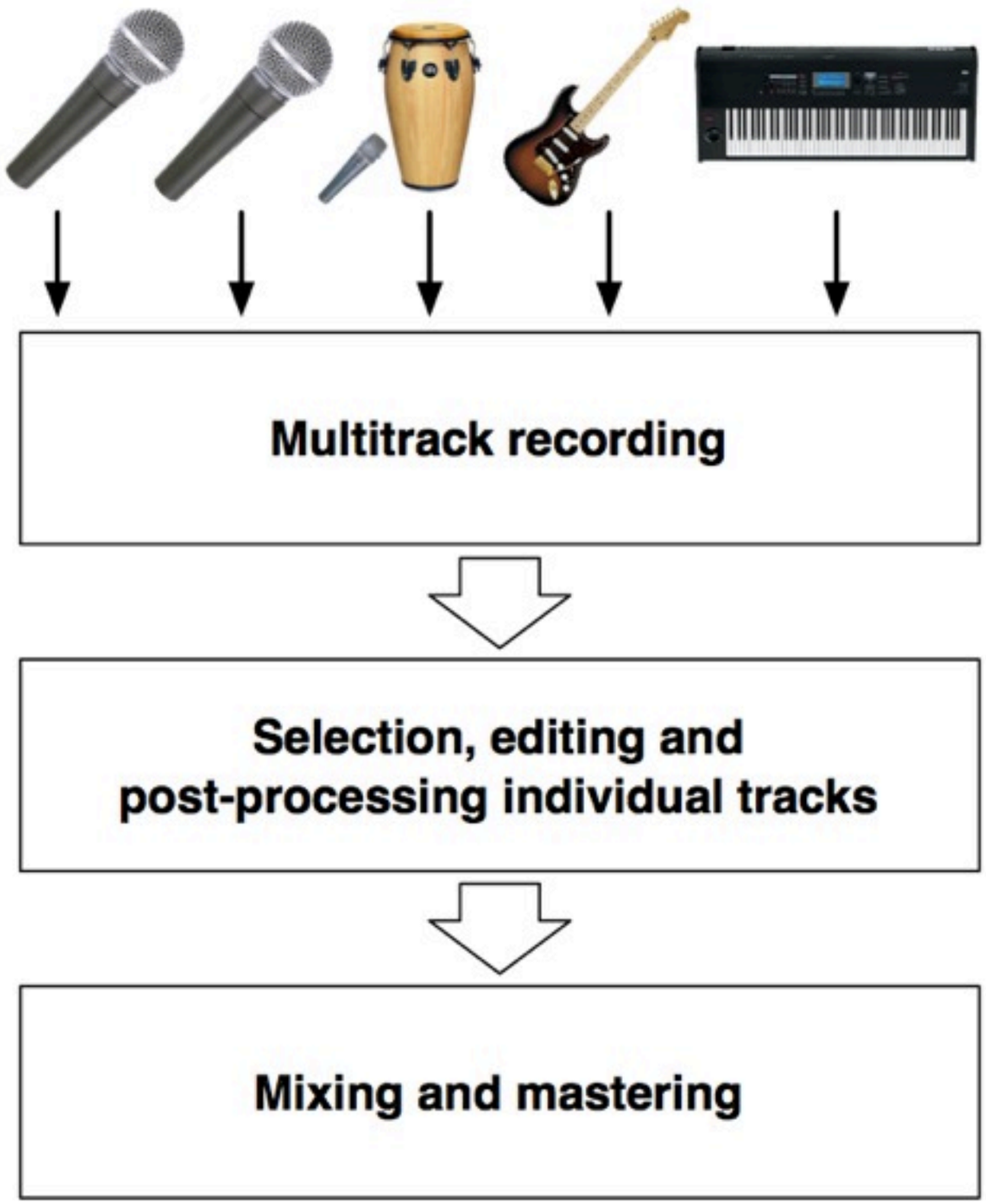
The Studio Ontology

- Enables collecting information about audio production.

- **Motivations**
- Notation for capturing the contribution of the engineer to creative work
- Improved Information and workflow management in the studio
- Exploit music production data in MIR systems
- Enable building intelligent music production systems

The Studio Ontology

- **Defines a Studio Production Workflow Model**

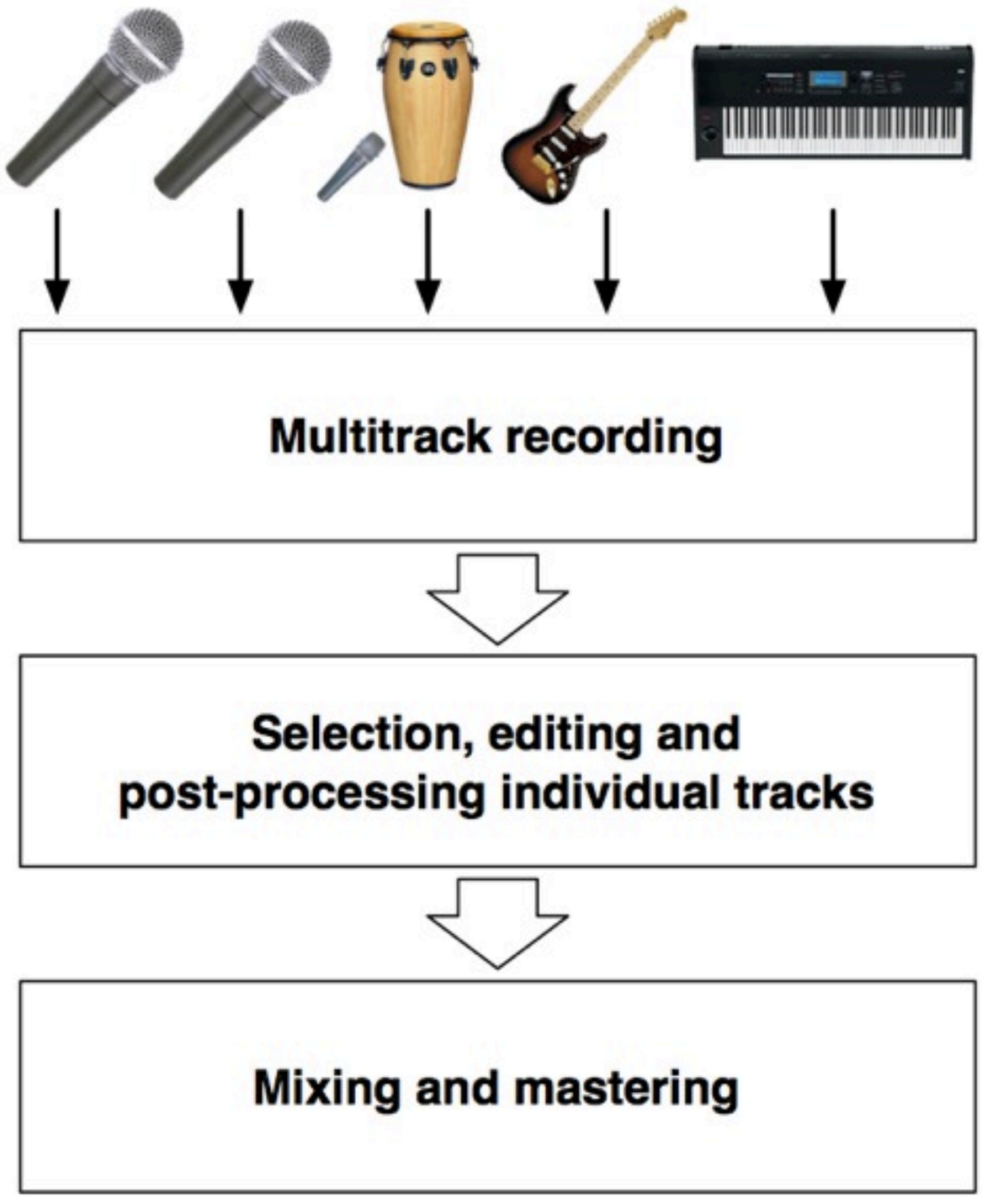


- Two parts:
  - **Technical** (domain independent)
  - **Musical** (domain specific)

The Studio Ontology

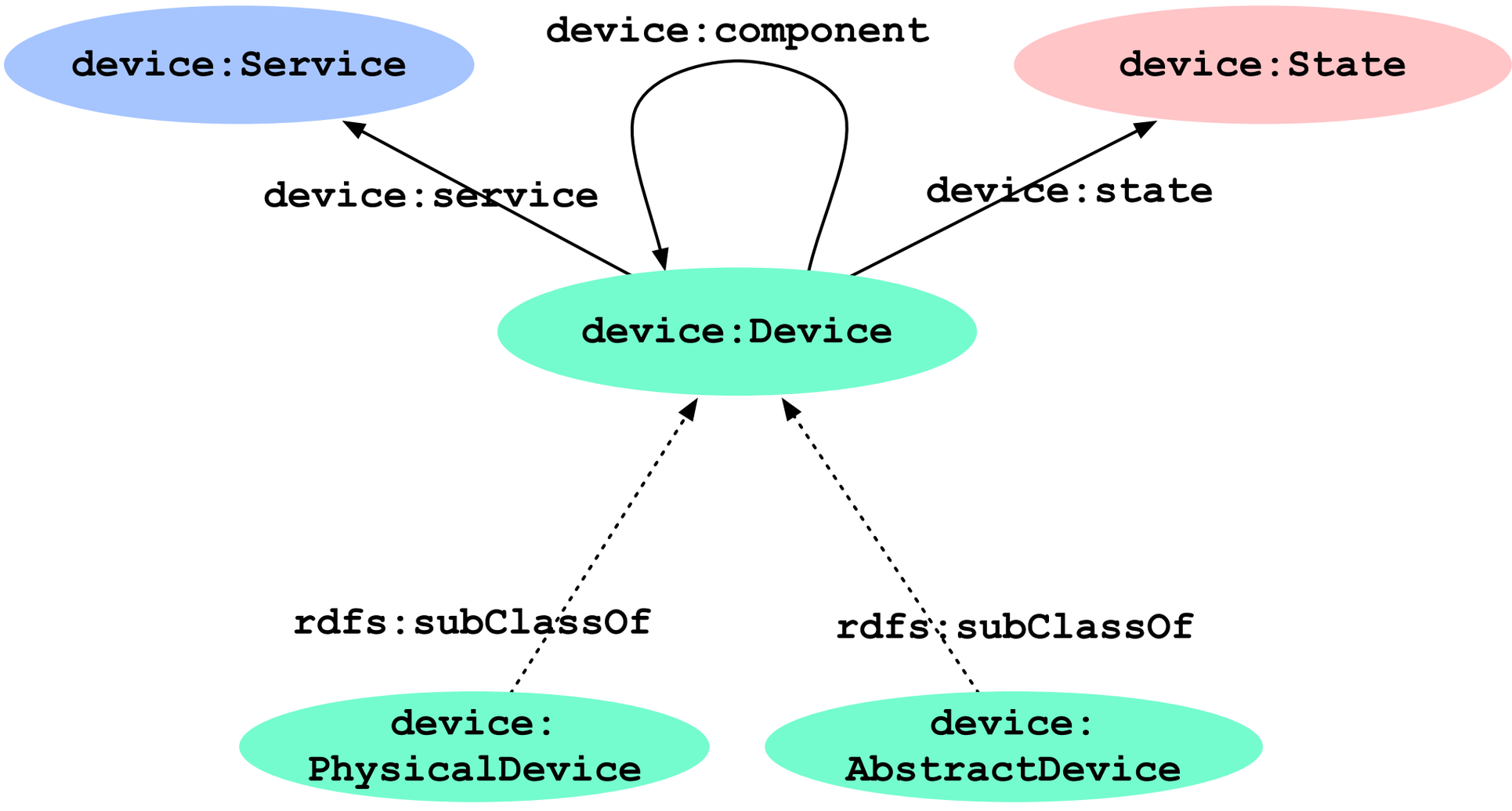
- **Defines a Studio Production Workflow Model**

- **Two parts:**
  - **Technical** (domain independent)
  - **Musical** (domain specific)



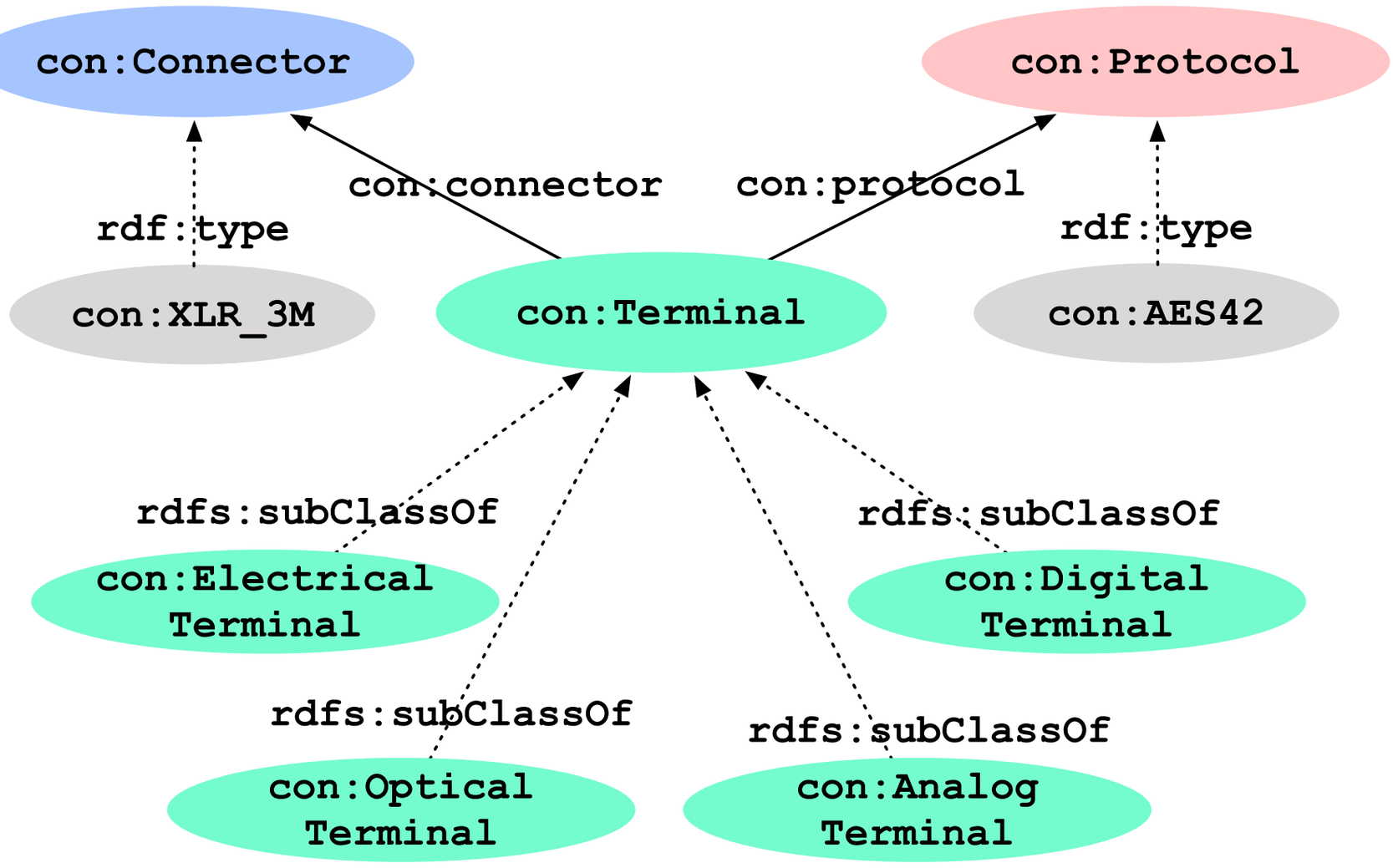
The Studio Ontology

- **Domain independent components:**
  - Technological artefacts (devices) and their connections



The Studio Ontology

- **Domain independent components:**
  - Technological artefacts (devices) and their connections



The Studio Ontology

- **A model of audio processing devices**

- **Phenomenon:** a physical process that produces for instance an audio effect
- **Model:** a computational model of the process
- **Implementation:** a particular implementation of the model, e.g. in C++
- **Device:** a concrete device that someone can own

The Studio Ontology

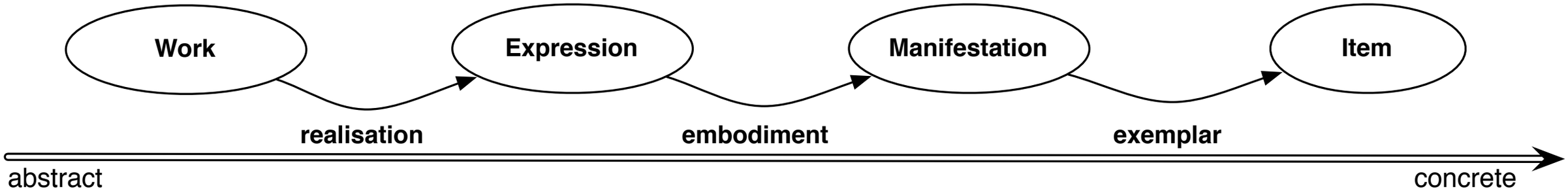
- **A model of audio processing devices**

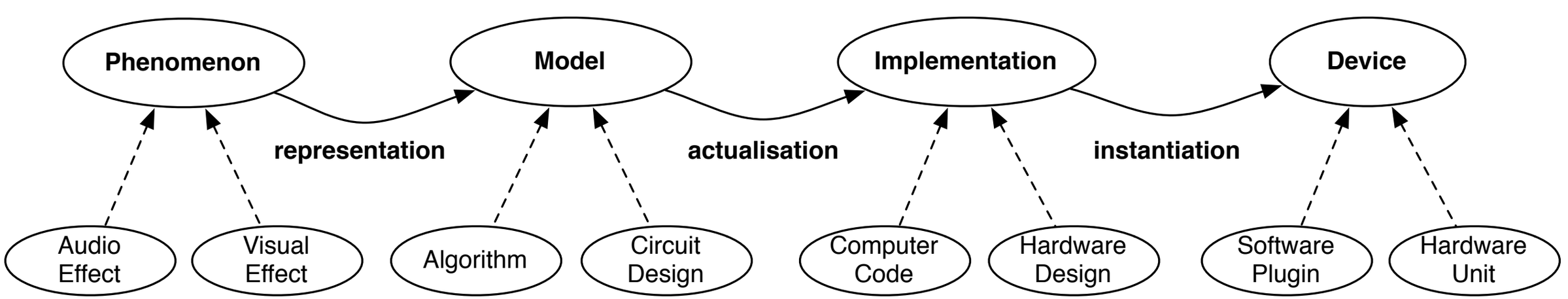
- **Phenomenon:** a physical process that produces for instance an audio effect
- **Model:** a computational model of the process
- **Implementation:** a particular implementation of the model, e.g. in C++
- **Device:** a concrete device that someone can own

The Studio Ontology

- **A model of audio processing devices**

FRBR model





Signal Processing Device model

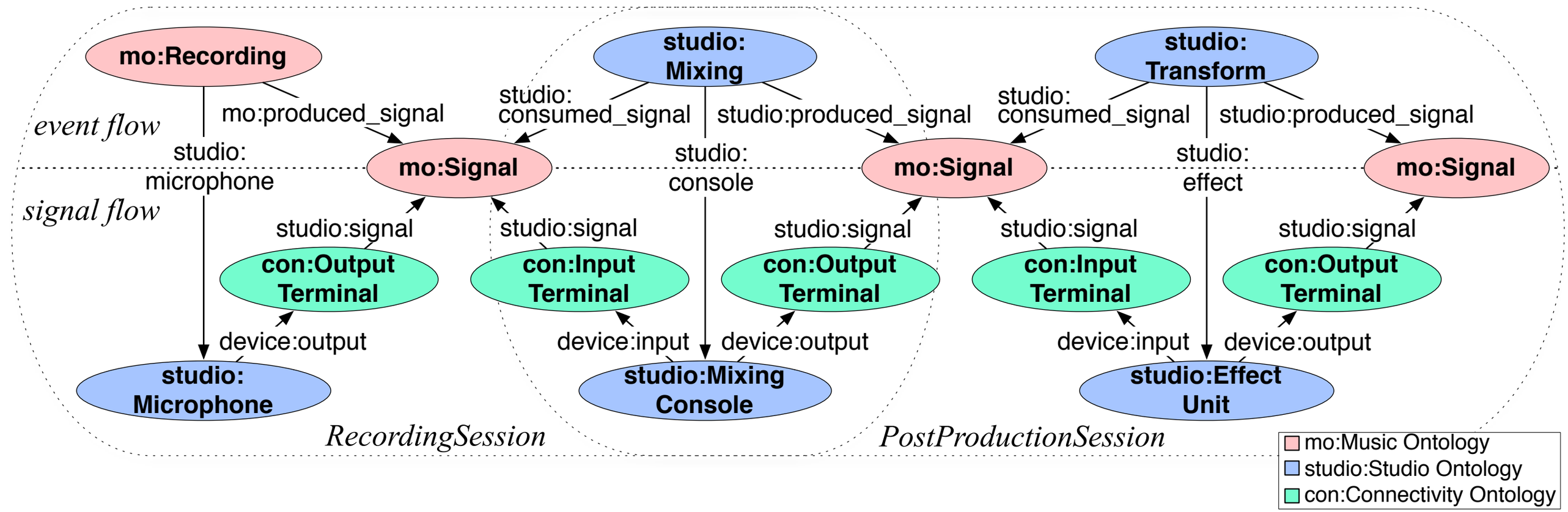
The Studio Ontology

- **Signal processing workflow model**
- with separate
  - **Event flow**
  - **Signal flow**

- This supports the requirements of real-time recording and audio processing scenarios
- as well as post-production.

The Studio Ontology

- **Signal processing workflow model**
- with separate
  - **Event flow**
  - **Signal flow**

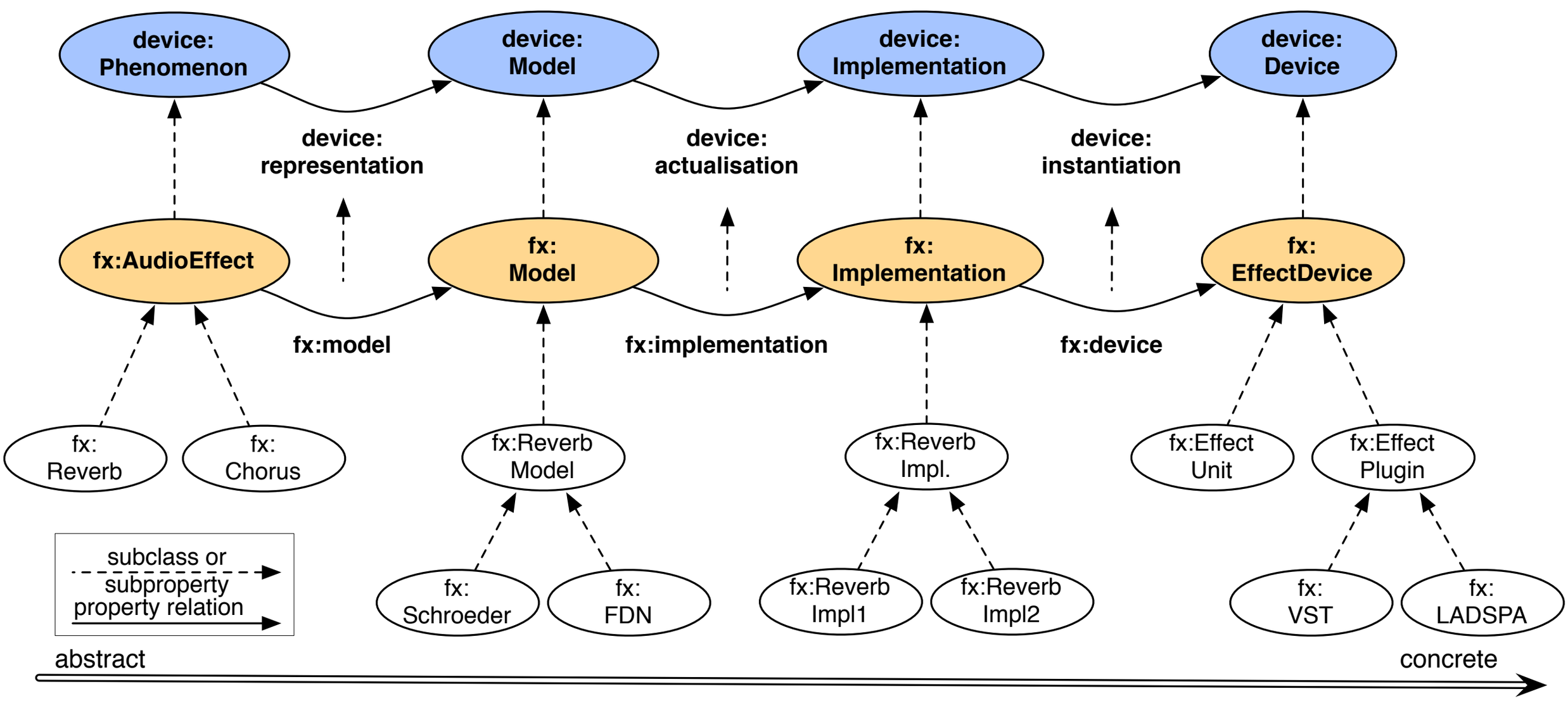


The Studio Ontology

- Extensions in 4 areas (more modules are in preparation)
  - **Audio Recording**
  - **Audio Mixing**
  - **Audio Effects**
  - **Audio Editing**

Audio Effects Ontology

- **A model of audio effects from physical phenomena to concrete devices**



Ontologies and tools for Semantic Audio

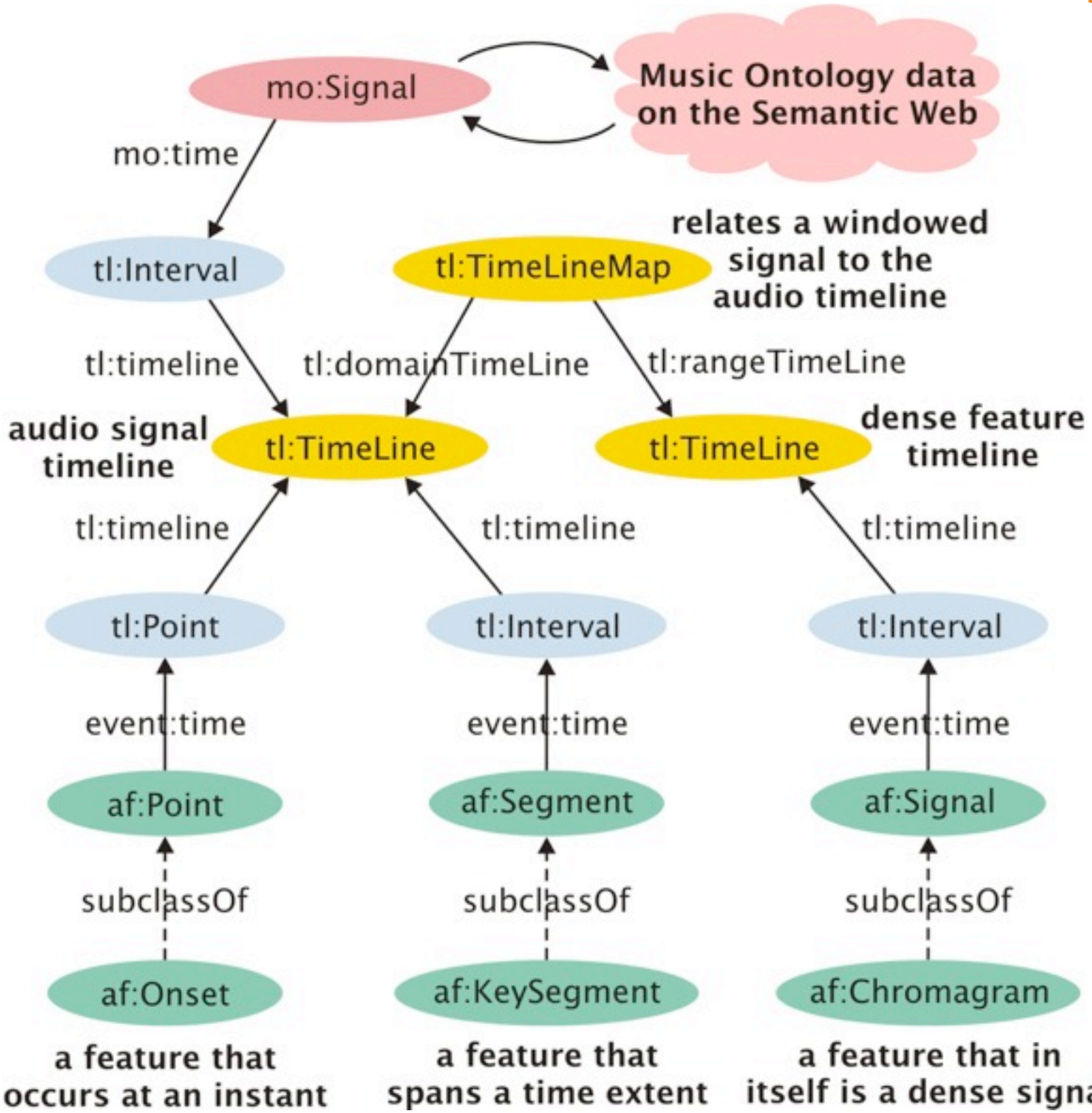
The Audio Features Ontology

- **Key points:**
  - Features represented by Events or Signals
  - Timelines link things together
- **Basic feature types:**
  - **Instants:** Time point like features,
    - e.g. a note onset
  - **Intervals:** Temporal segments,
    - e.g. the duration of the intro of a song
  - **Dense features:** signal like features,
    - e.g. a spectrogram

The Audio Features Ontology

- **Key points:**
  - Features represented by Events or Signals
  - Timelines link things together
- **Basic feature types:**
  - **Instants:** Time point like features,
    - e.g. a note onset
  - **Intervals:** Temporal segments,
    - e.g. the duration of the intro of a song
  - **Dense features:** signal like features,
    - e.g. a spectrogram

The Audio Features Ontology



The Audio Features Ontology

- (1) A note onset on the signal timeline:

```
@prefix tl: <http://purl.org/NET/c4dm/timeline.owl#>.
@prefix af: <http://purl.org/ontology/af/>.

:signal_timeline a tl:Timeline .
:onset_23 a af:Onset;
        event:time [
            a tl:Instant ;
            tl:timeline :signal_timeline ;
            tl:at "PT1.710S"^^xsd:duration ;
        ] .
```

- An instant on a timeline

The Audio Features Ontology

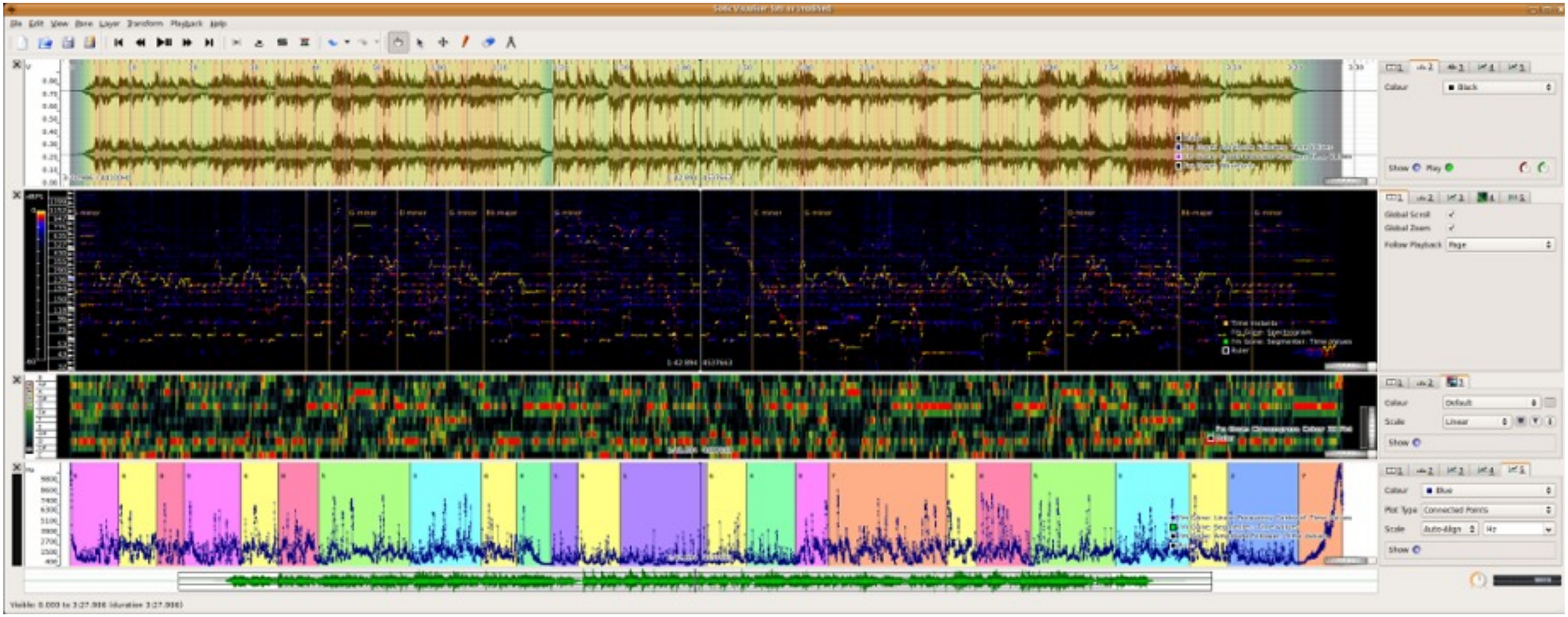
- (2) A key segment:

```
:signal_timeline a tl:Timeline .
:key_segment_1 a af:Segment;
     rdfs:label """Bb major""" ;
     af:feature "11" ;
     event:time [
       a tl:Interval ;
       tl:timeline :signal_timeline ;
       tl:start "PT30.1S";
       tl:duration "PT200S";
     ] .
```

- An interval on a timeline

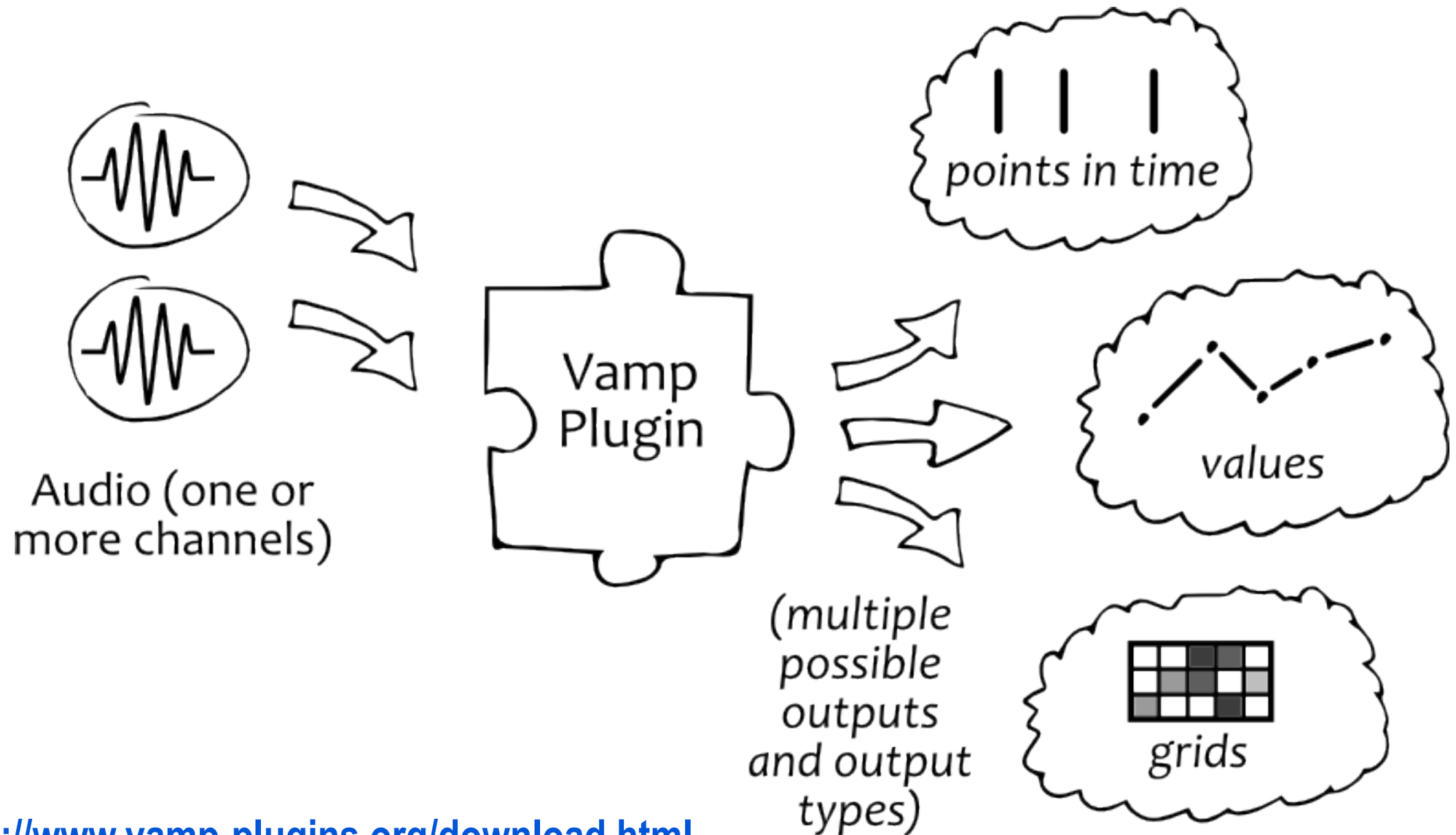
Semantic Audio Tools

- Tools that produce and read RDF according to these ontologies include:
  - **Sonic Annotator**
  - **Sonic Visualiser**



Vamp Plugins

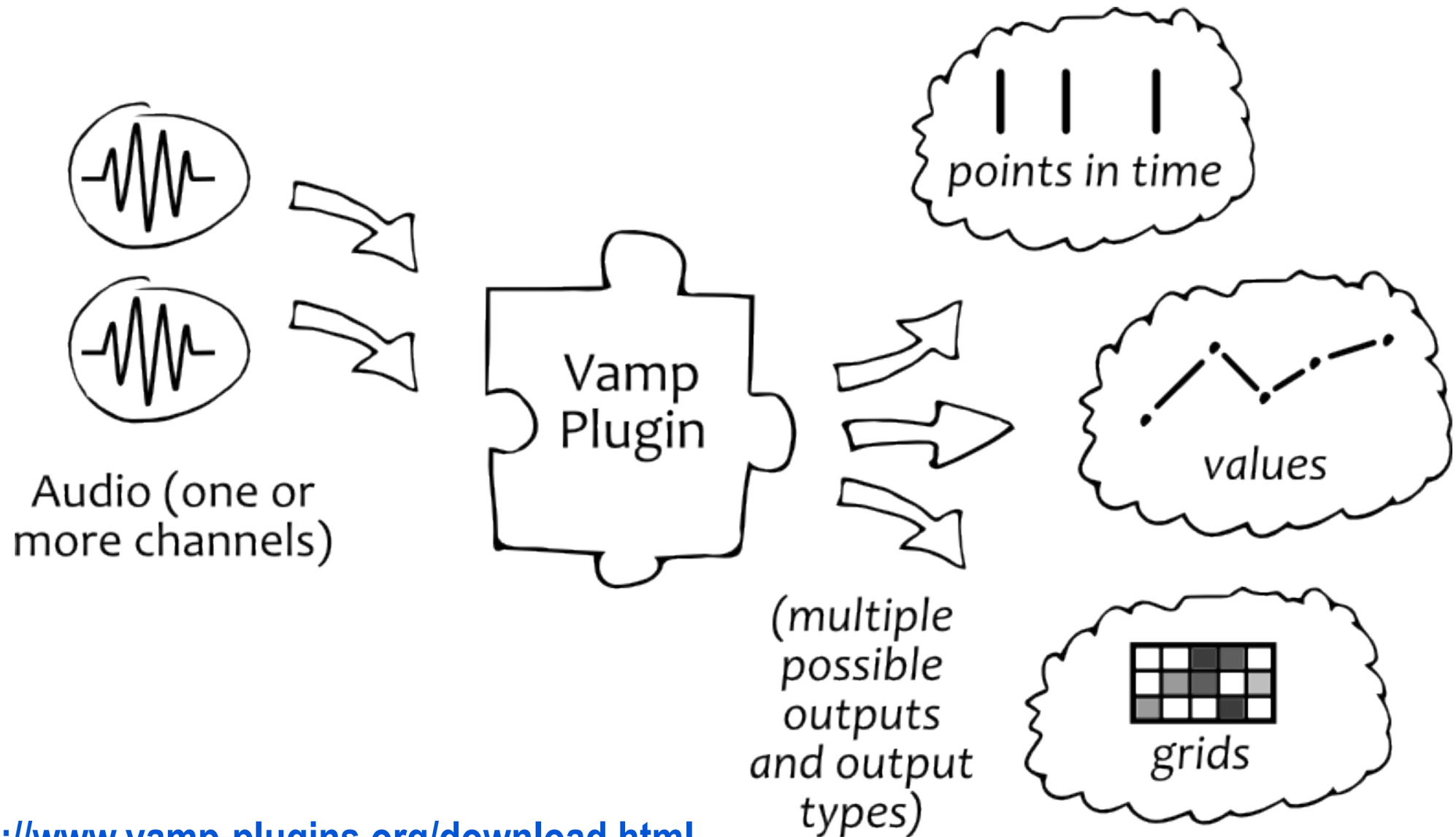
- An Application Programming Interface for feature extraction



http://www.vamp-plugins.org/download.html

Vamp Plugins

- Vamp plugins take audio input and return structured data (but not RDF!)



http://www.vamp-plugins.org/download.html

Vamp Plugins

- **Vamp Plugin Ontology:**
  Links the results with a plugin and the enclosed algorithm that computed them.

- **Vamp Transform Ontology:**
  Allows to express the parameters (e.g. window size) that were used to obtain a particular set of results.

- **Plugins, parameters and results are linked, and described using the same format!**

Sonic Annotator

- A command line Vamp plugin host that outputs RDF
- **Key features:**
- A program for analysing large collections available locally, or on the Web.
- It can read a very wide range of audio file formats.
- Reads Vamp plugin configuration in RDF
- Returns the features in RDF linked with the configuration and editorial data (if available)

Sonic Annotator

- A command line Vamp plugin host that outputs RDF
- **Key features:**
- A program for analysing large collections available locally, or on the Web.
- It can read a very wide range of audio file formats.
- Reads Vamp plugin configuration in RDF
- Returns the features in RDF linked with the configuration and editorial data (if available)

Sonic Annotator

- A command line Vamp plugin host that outputs RDF
- **Key features:**
- A program for analysing large collections available locally, or on the Web.
- It can read a very wide range of audio file formats.
- Reads Vamp plugin configuration in RDF
- Returns the features in RDF linked with the configuration and editorial data (if available)

Sonic Annotator

- (1) Create an RDF transform skeleton:

```
$ sonic-annotator -s \
vamp:vamp-example-plugins:fixedtempo:tempo > transform.n3
```

- (2) Edit the file if necessary and run the feature extractor:

```
$ sonic-annotator -t transform.n3 \
vamp:vamp-example-plugins:fixedtempo:tempo \
-w rdf --rdf-stdout audio_file.wav
```

- This will dump the results on the standard output.
  - A detailed tutorial is available at
  - http://www.omras2.org/SonicAnnotator

Sonic Annotator

- (1) Create an RDF transform skeleton:

```
$ sonic-annotator -s \
vamp:vamp-example-plugins:fixedtempo:tempo > transform.n3
```

- **(2) Edit the file if necessary and run the feature extractor:**

```
$ sonic-annotator -t transform.n3 \
vamp:vamp-example-plugins:fixedtempo:tempo \
-w rdf --rdf-stdout audio_file.wav
```

- This will dump the results on the standard output.
  - A detailed tutorial is available at
  - http://www.omras2.org/SonicAnnotator

SAWA

- Sonic Annotator Web Application

- **A tool for Web-based audio analysis**

- Runs Vamp feature extractor plugins on a small uploaded audio collection

- Configured using RDF and return RDF data according to the Audio Features Ontology.

SAWA

Configure Transform: Note Onset Detector, Onset Detection Function output

Parameter	Value	Limits	Unit
Block Size	1024	-	Samples
Step Size	512	-	Samples
Onset Detection Function Type	Complex Domain	(0 - 4)	-
Adaptive Whitening	☑	on/off	-
Onset Detector Sensitivity	50	(0 - 100)	%

Options

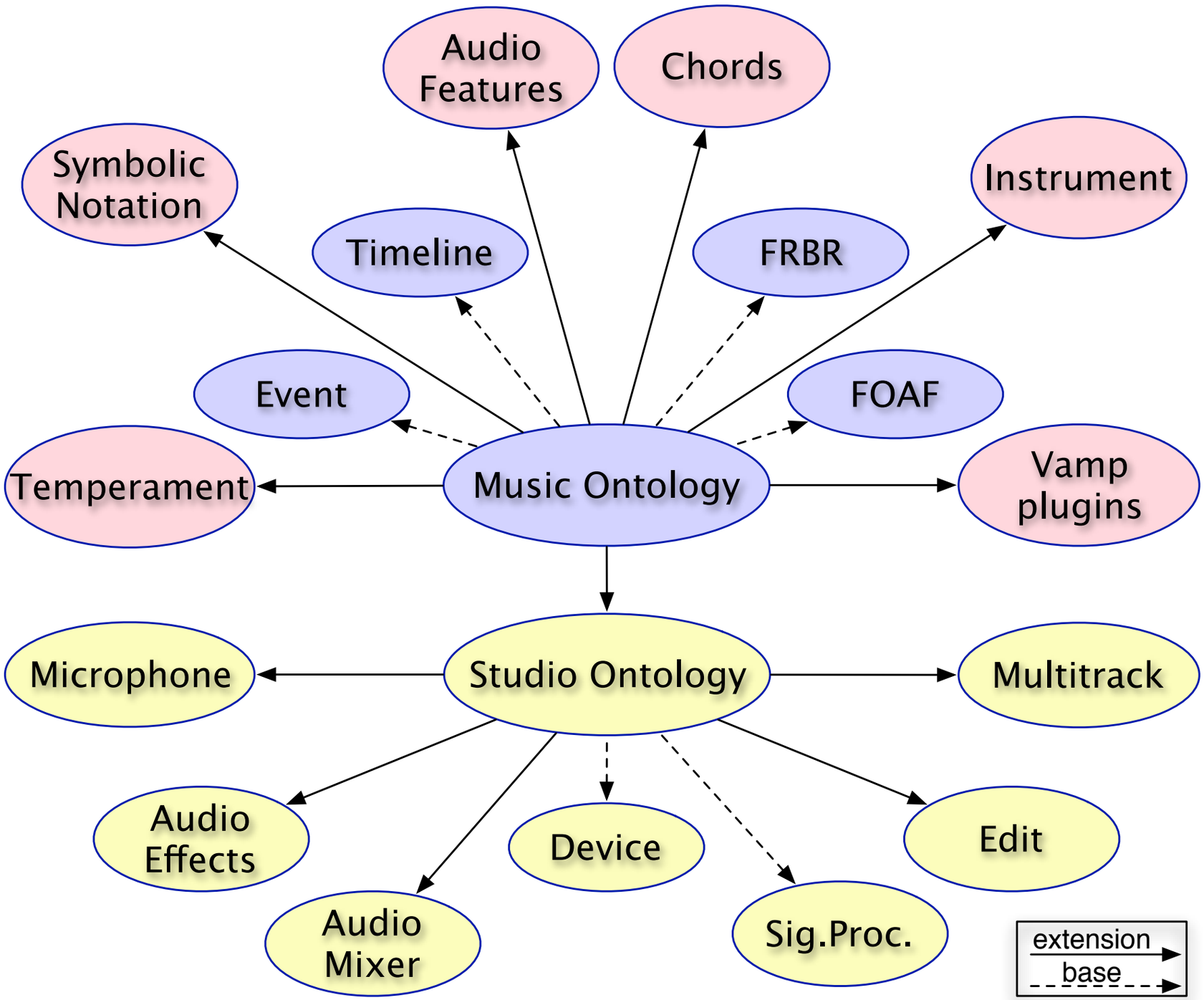
Transform name: **User001**
Transform identifier: fff1a579-3858-5eb5-bca8-c311bc33464a
Transform enabled: ☑
Treat transform as non-deterministic (do not use cache): ☐

Update | Save | Save As | Delete | Cancel

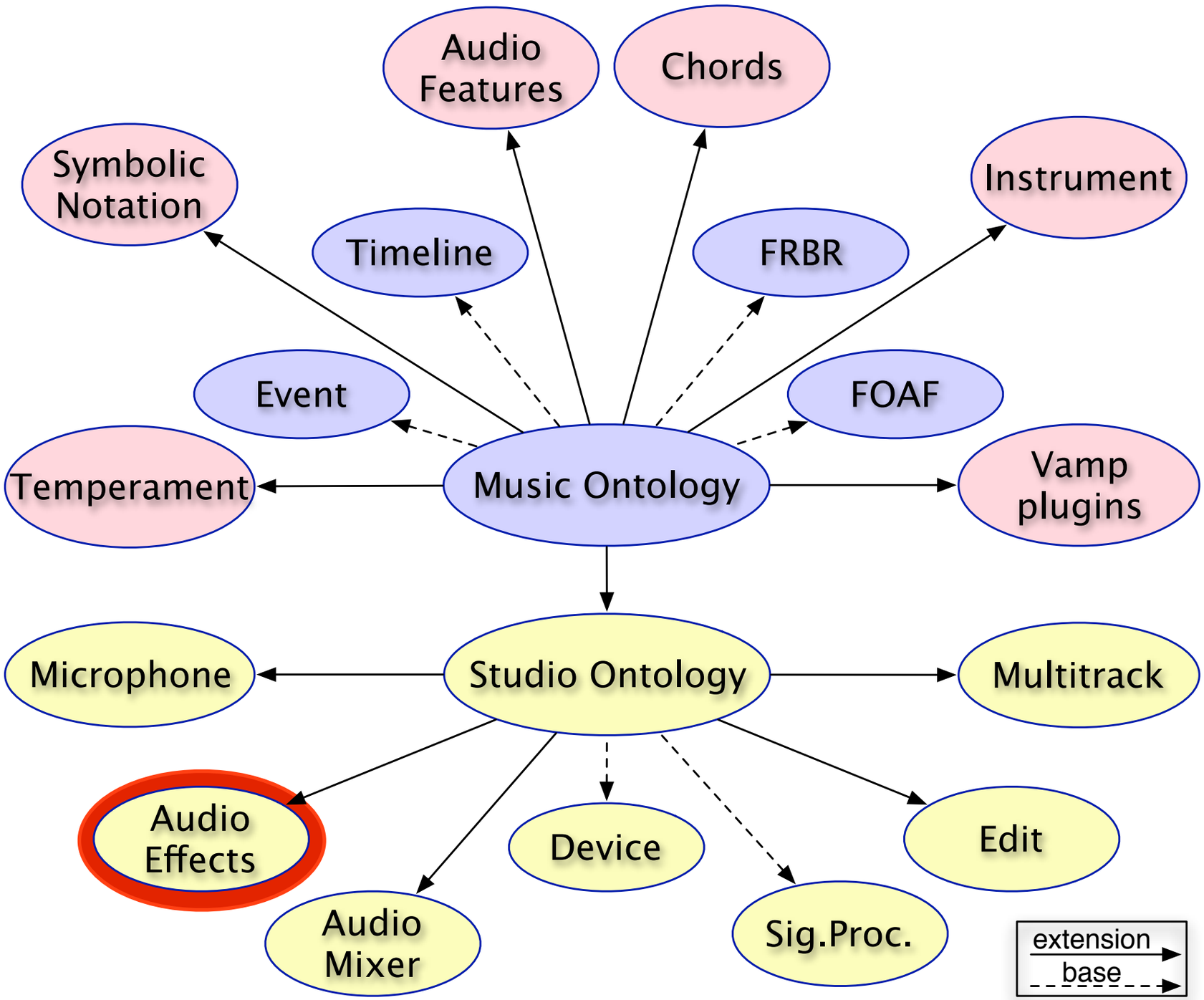
```
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix vamp:     <http://purl.org/ontology/vamp/> .
@prefix :         <#> .

:transform a vamp:Transform ;
    vamp:plugin <http://vamp-plugins.org/rdf/plugins/qm-vamp-plugins#qm-onsetdetector> ;
    vamp:step_size "512"^^xsd:int ;
    vamp:block_size "1024"^^xsd:int ;
```

Ontologies and Tools for Music Production

Audio FX Ontology



Audio FX Ontology

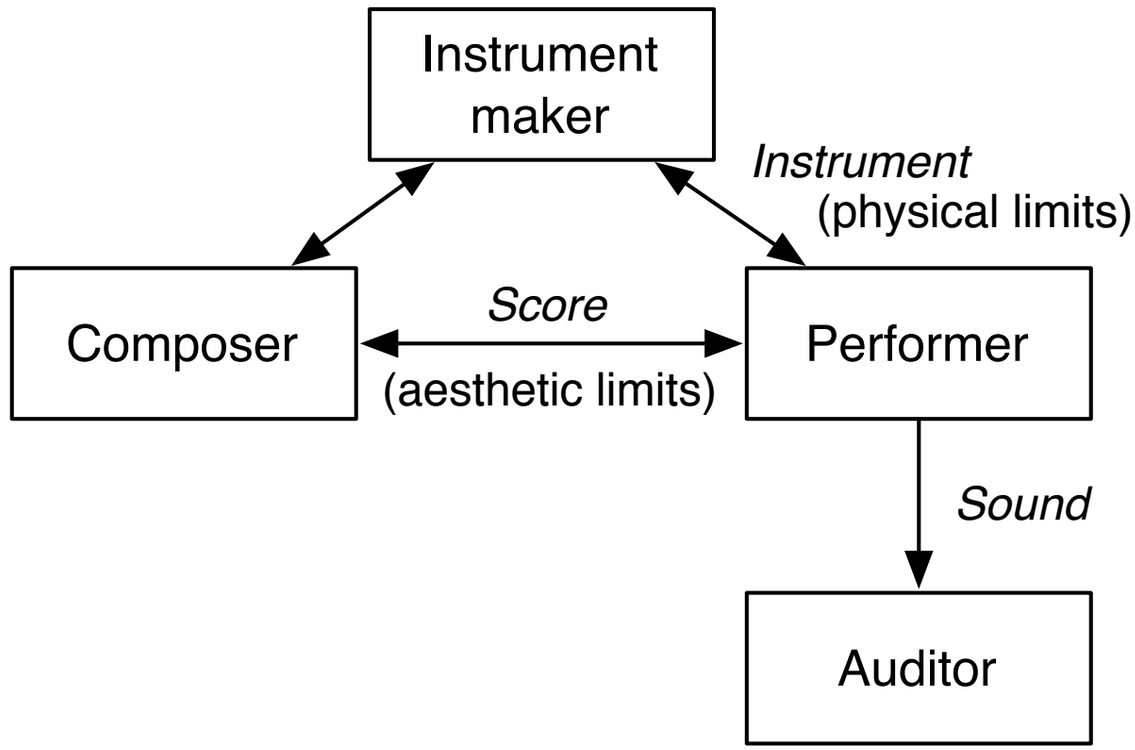


Audio FX Ontology

- enable communication between musicians, developers and engineers

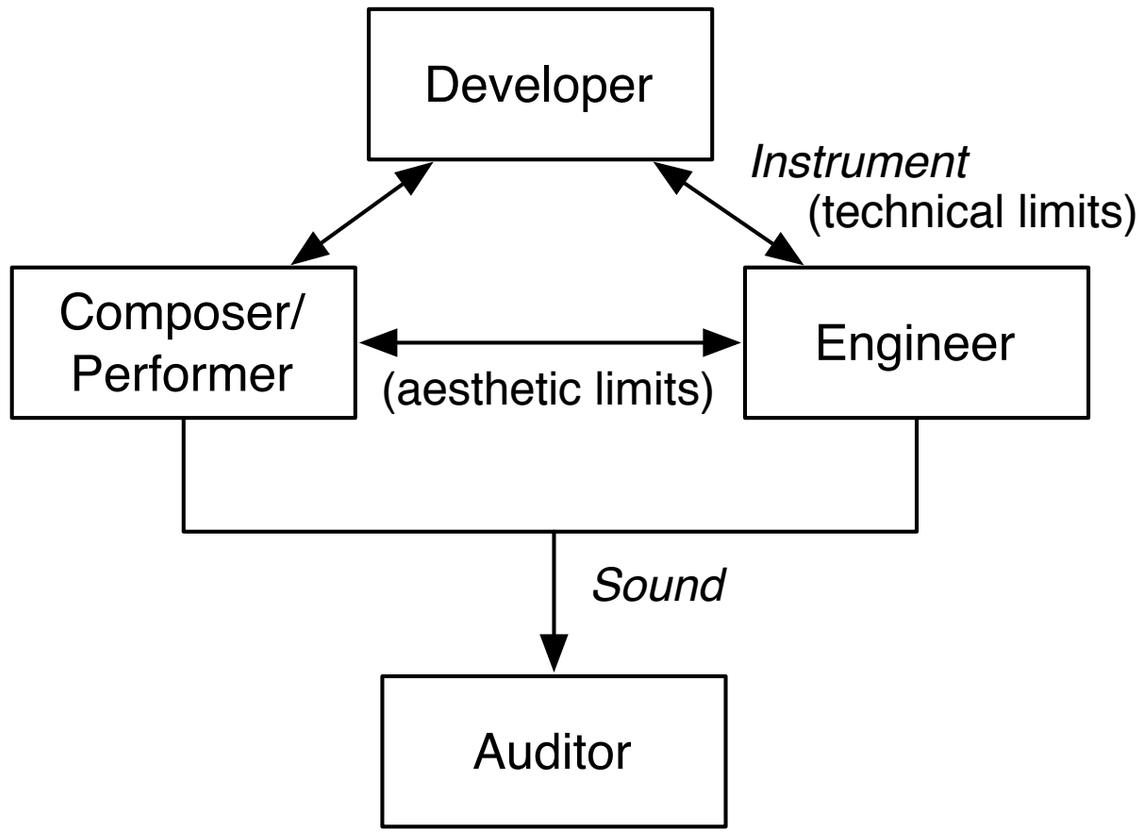
Audio FX Ontology

- enable communication between musicians, developers and engineers



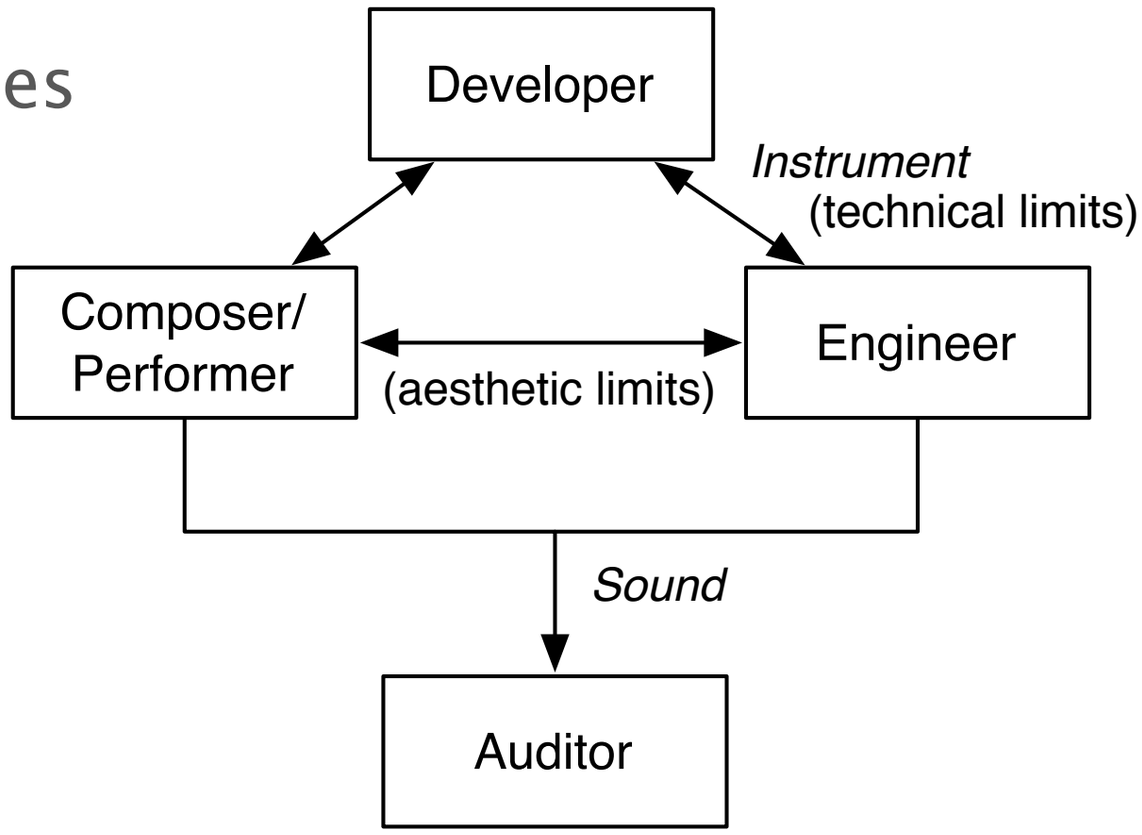
Audio FX Ontology

- enable communication between musicians, developers and engineers



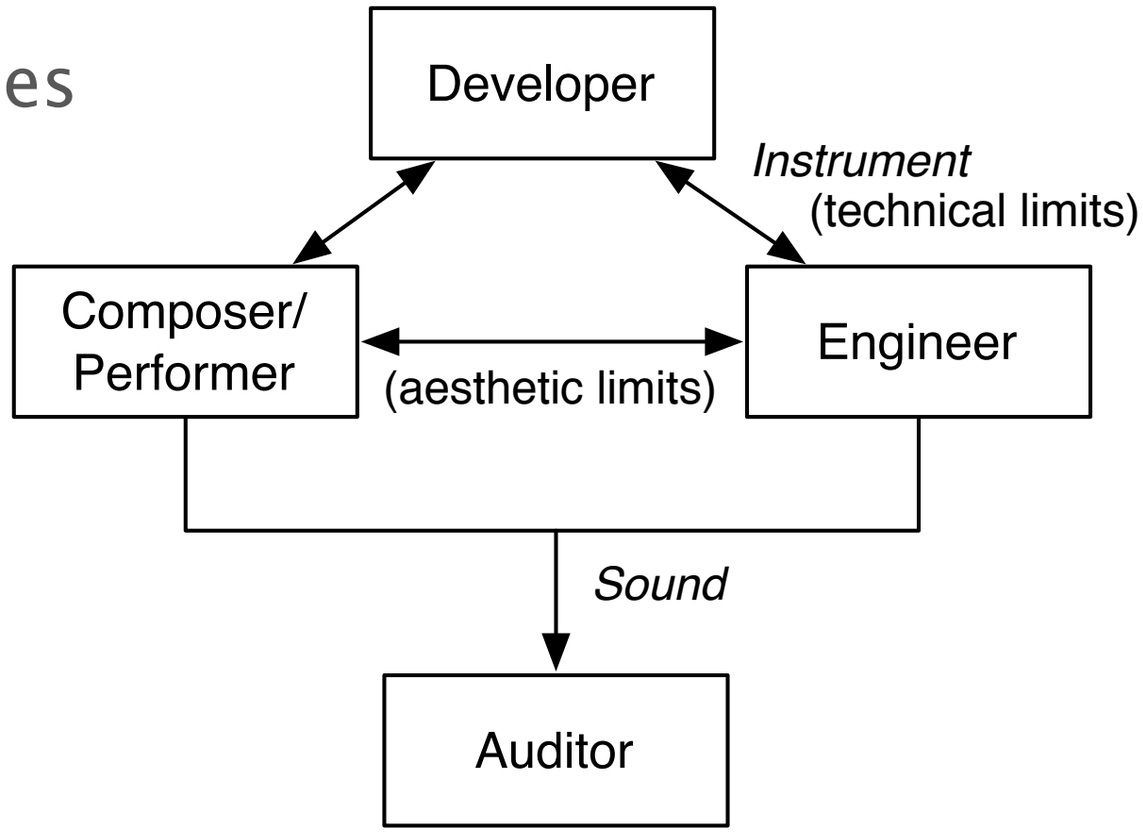
Audio FX Ontology

- enable communication between musicians, developers and engineers
- interdisciplinary classification of audio effects:
  - perceptual attributes
  - implementation techniques
  - application



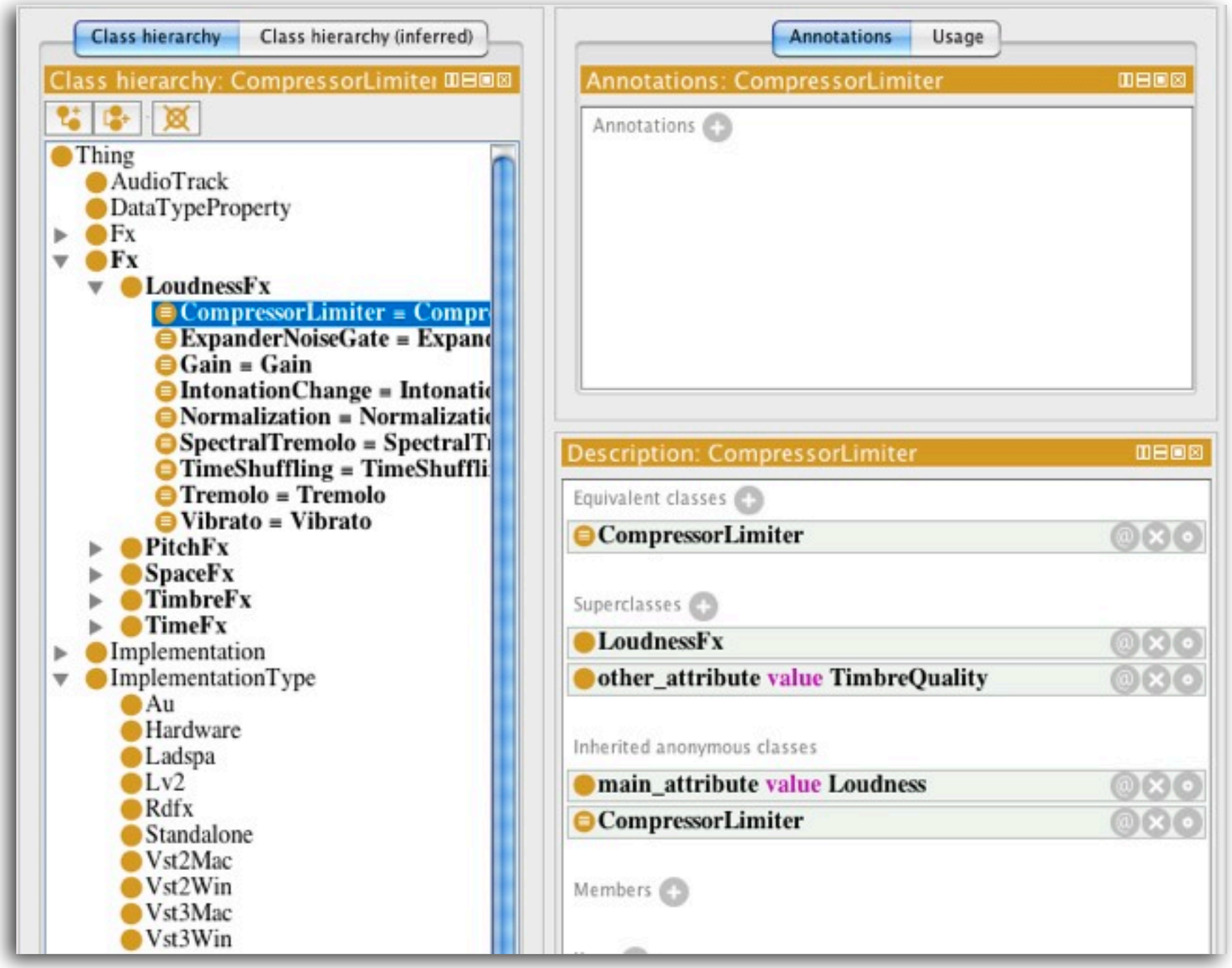
Audio FX Ontology

- enable communication between musicians, developers and engineers
- interdisciplinary classification of audio effects:
  - perceptual attributes
  - implementation techniques
  - application
- Modularised
  - Vocabulary
    - List of FX
    - Descriptors
    - Application of FX
  - Classifications



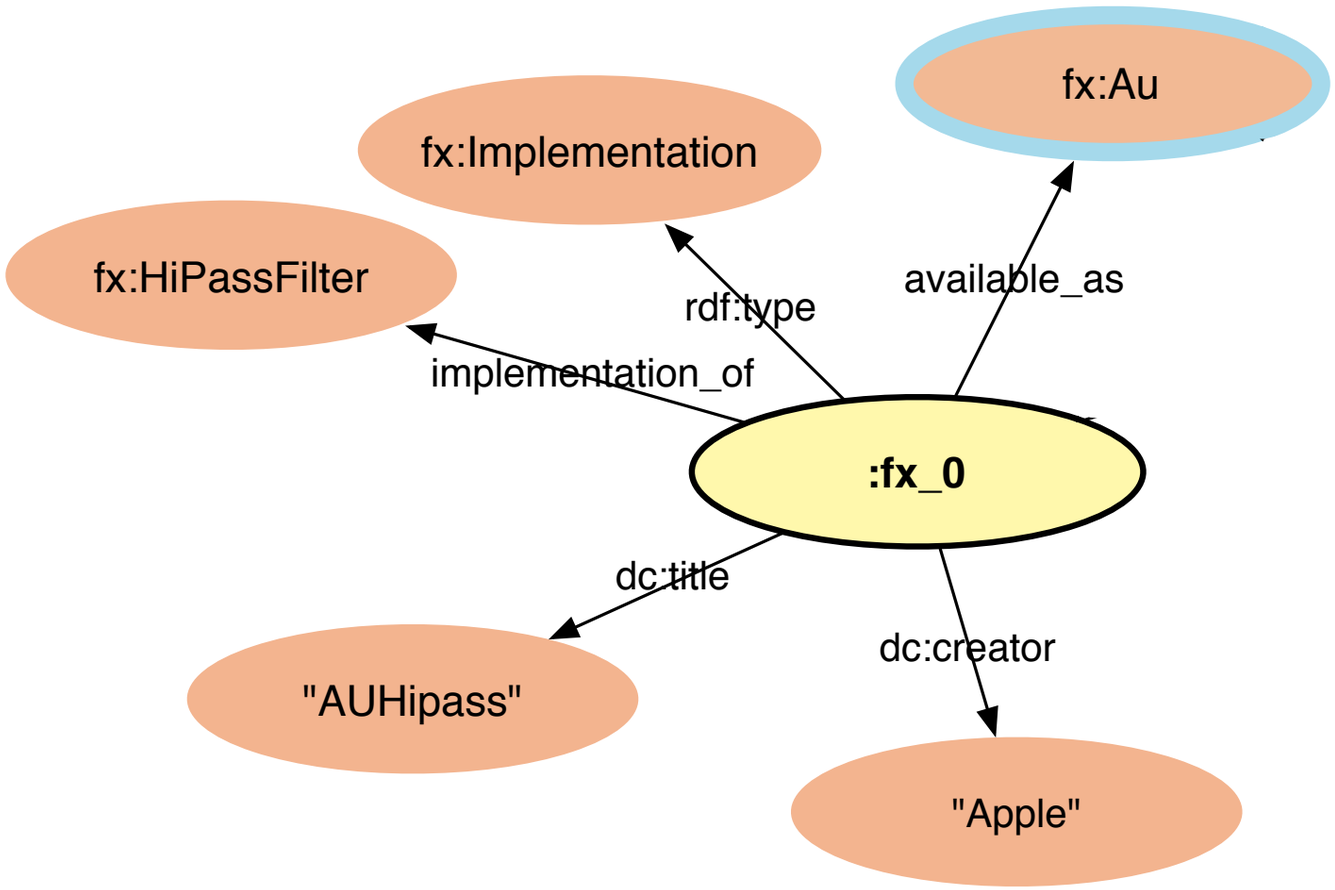
Perceptual Classification

- Loudness
- Pitch/Harmony
- Space
- Timbre
- Time/Duration



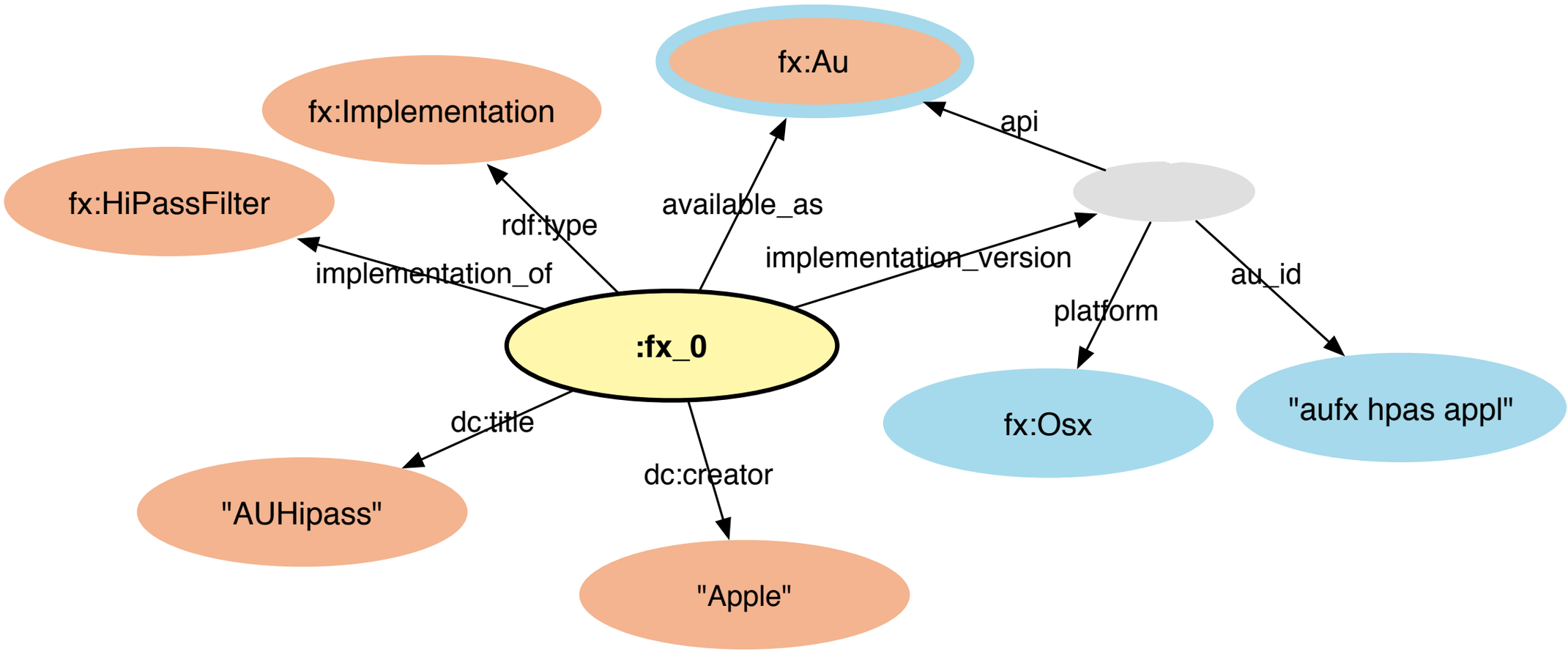
Audio FX Description

- general descriptors



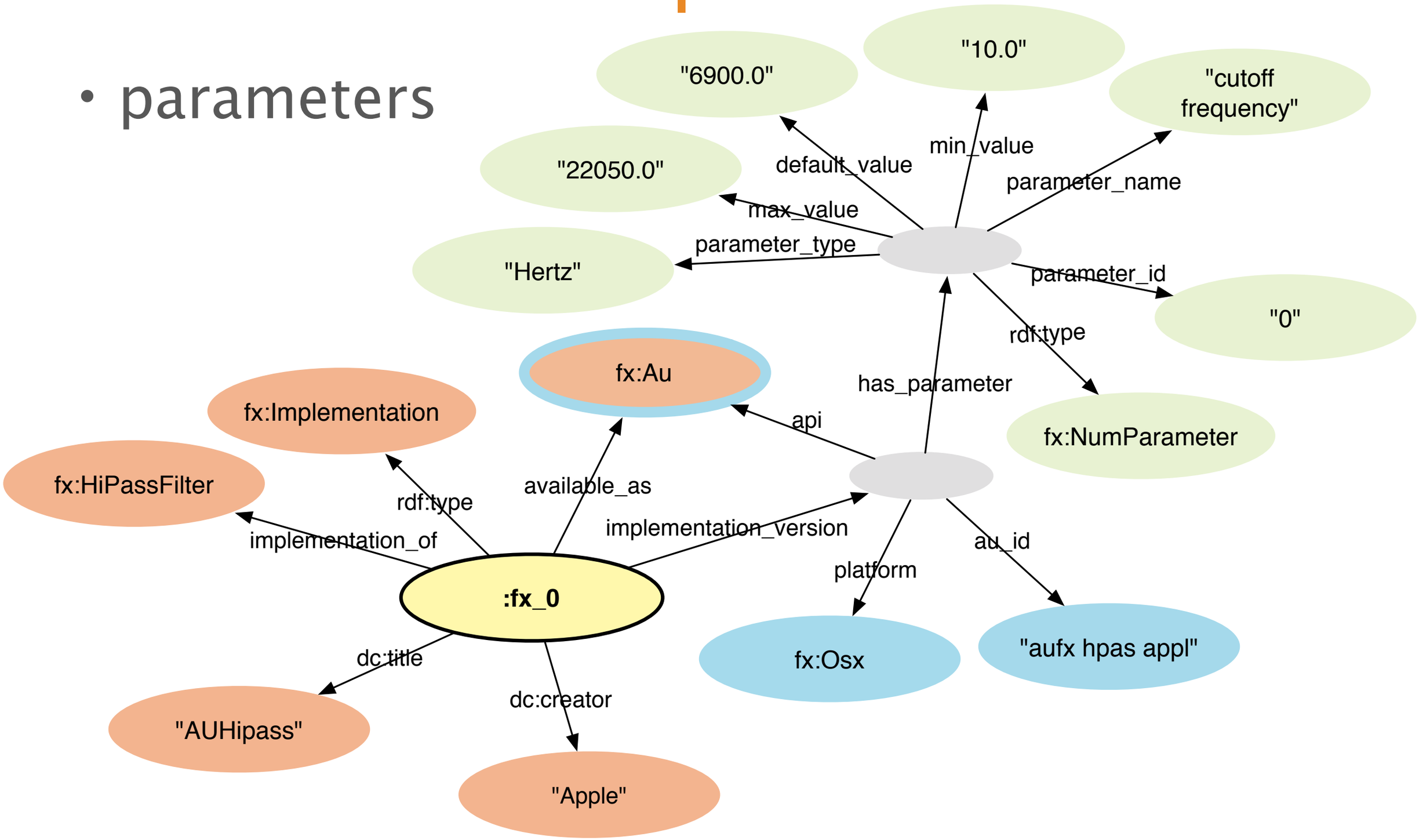
Audio FX Description

- specific version



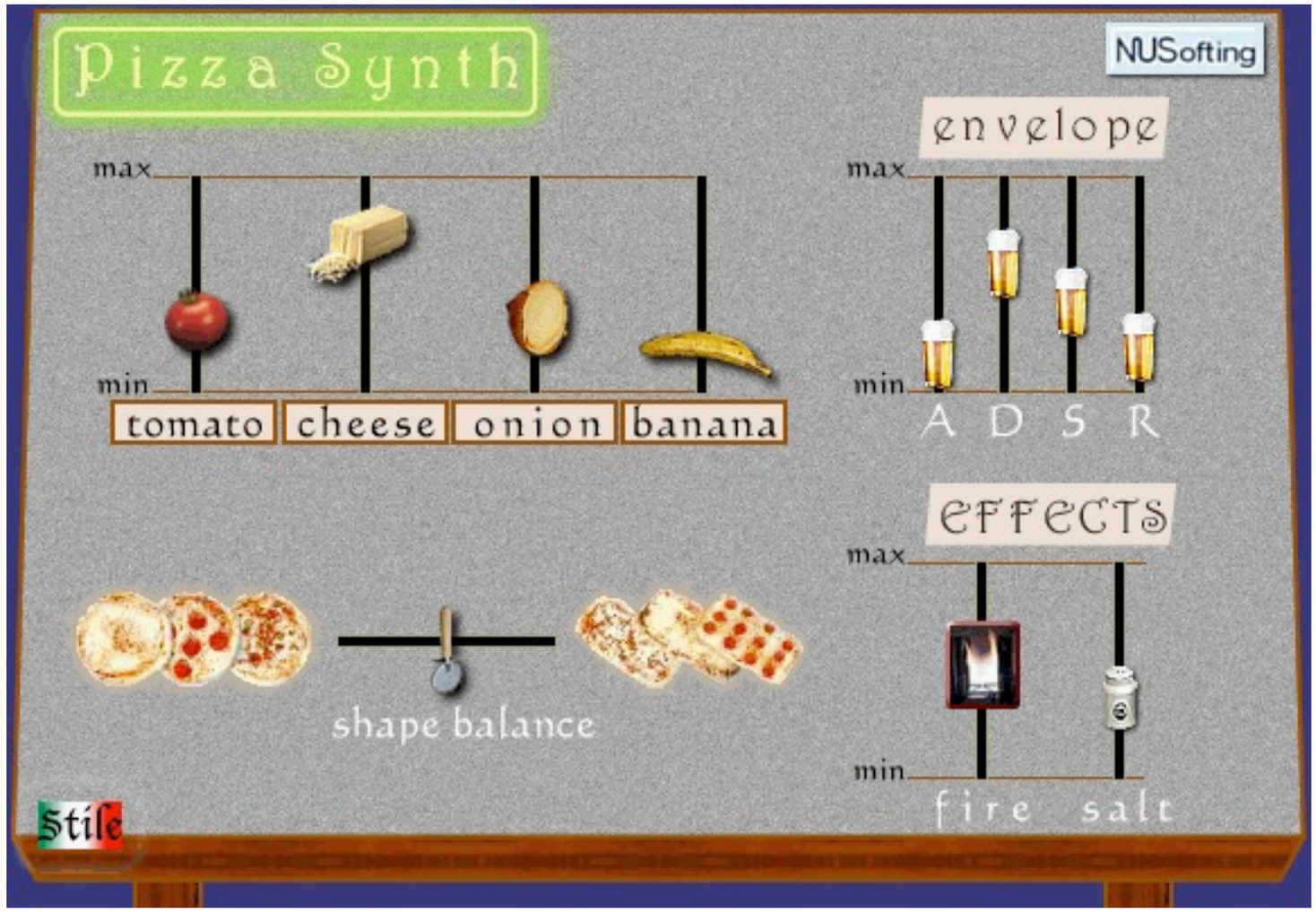
Audio FX Description

- parameters



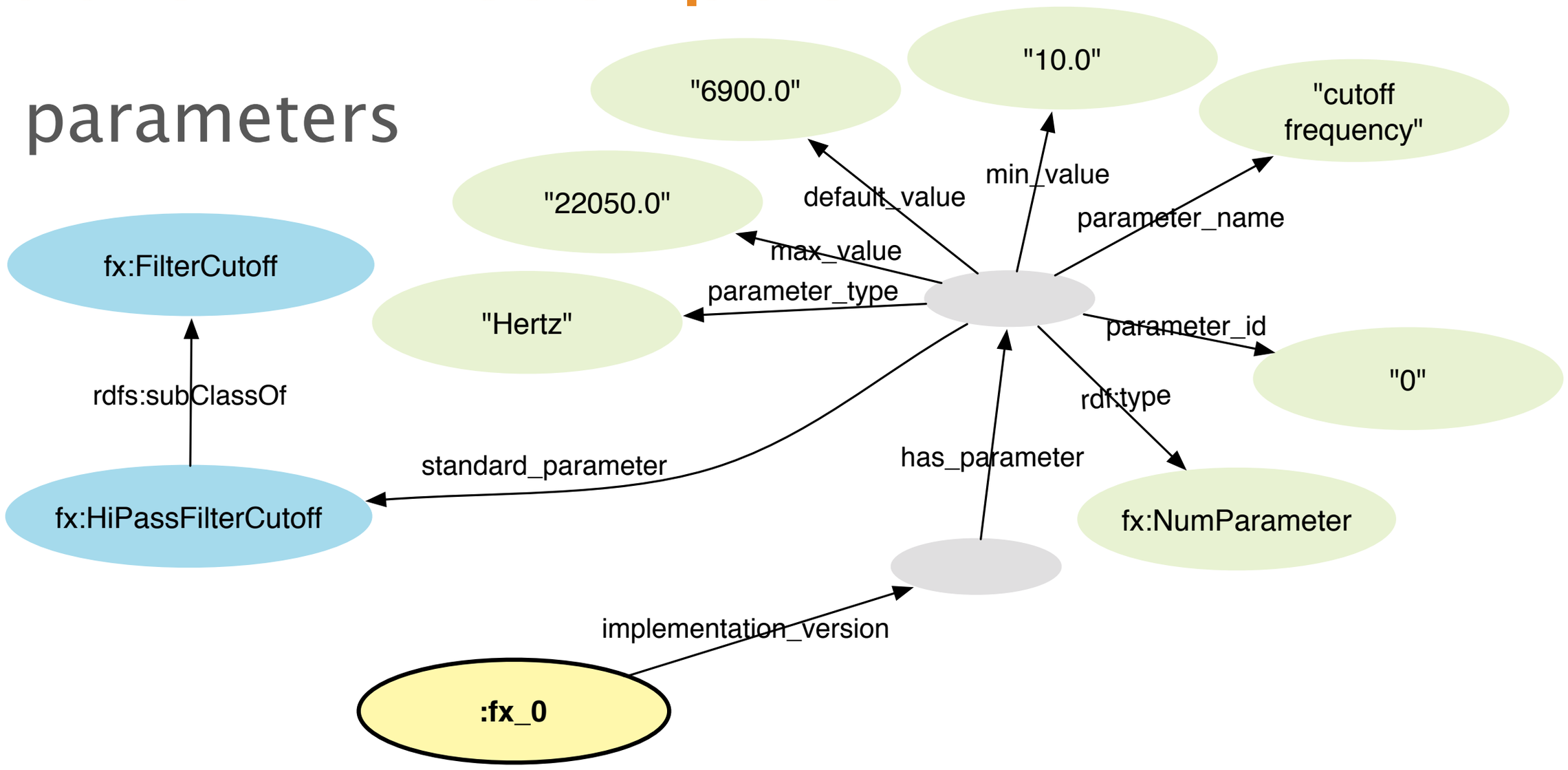
Audio FX Description

- parameters



Audio FX Description

- parameters

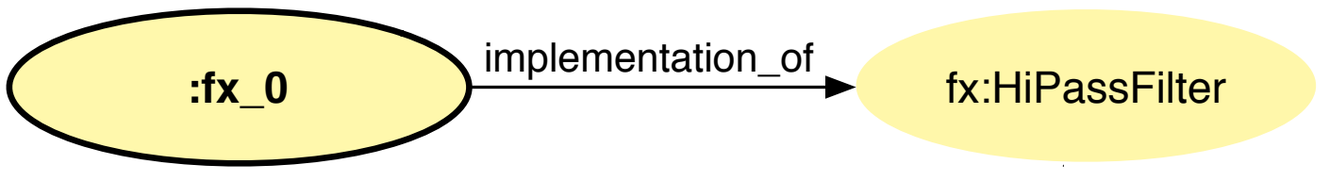


- definition of standard parameter classes

Audio FX Description

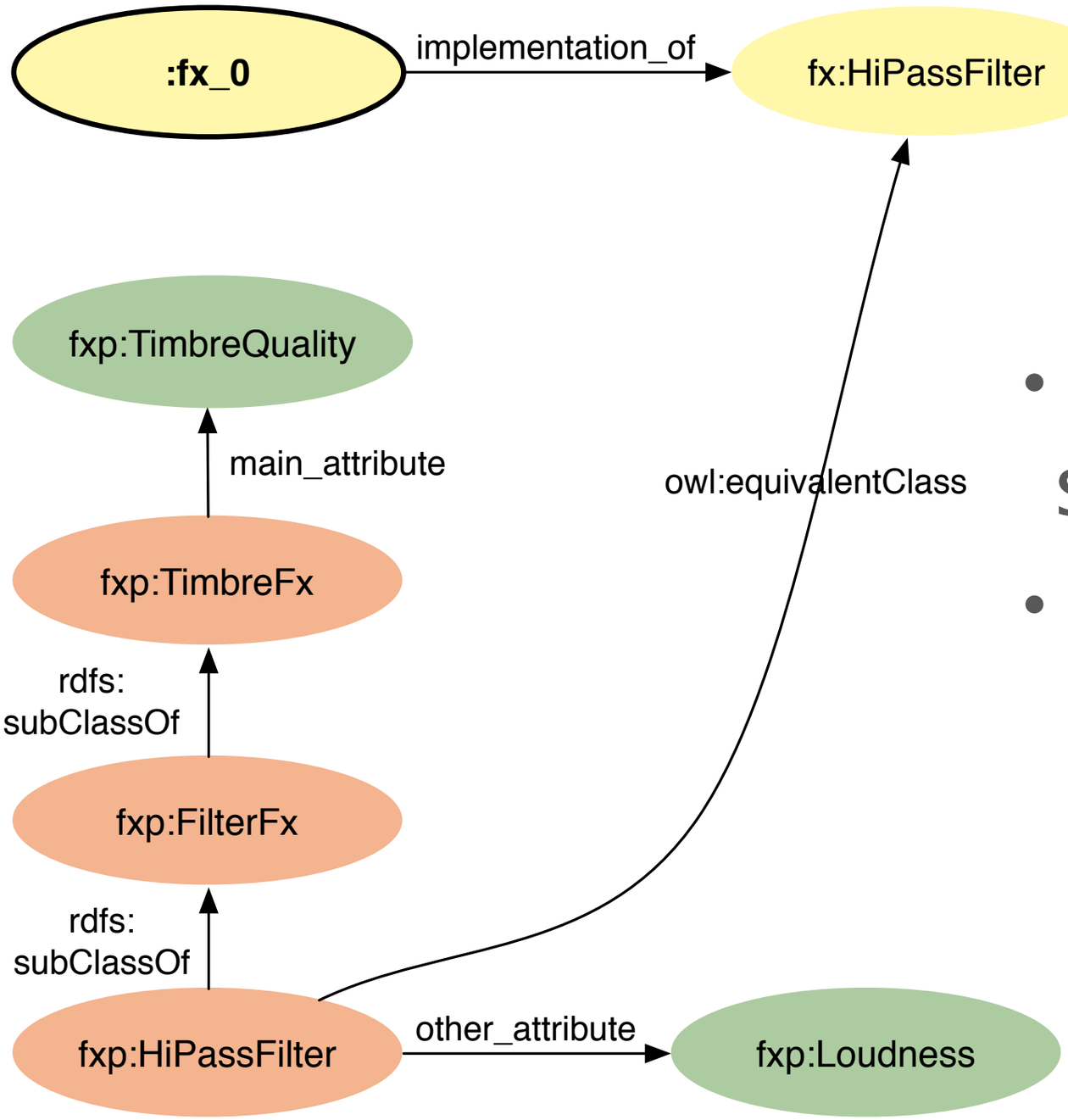
- link to classification system
- e.g. perceptual (fxp)

Audio FX Description



- link to classification system
- e.g. perceptual (fxp)

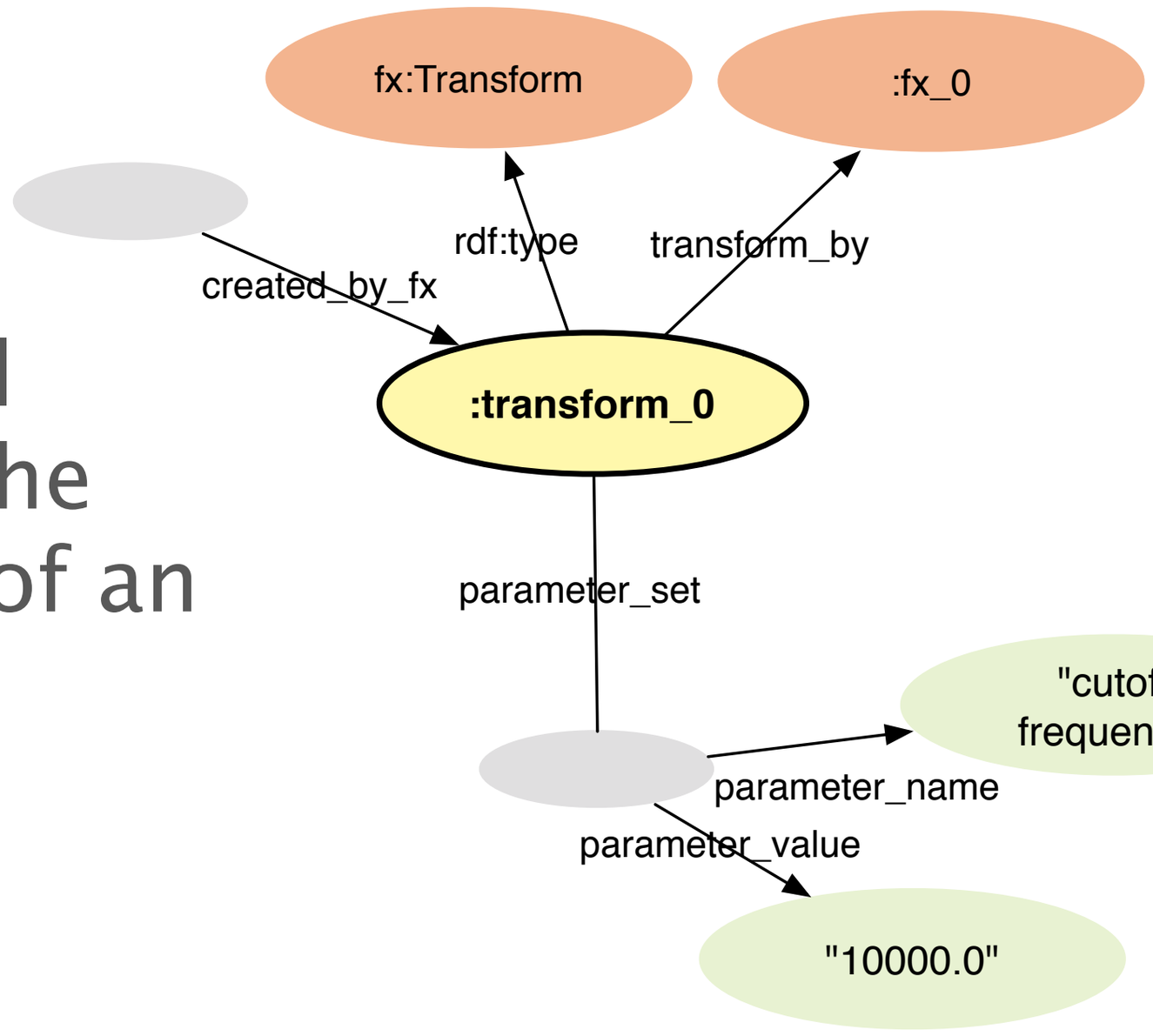
Audio FX Description



- link to classification system
- e.g. perceptual (fxp)

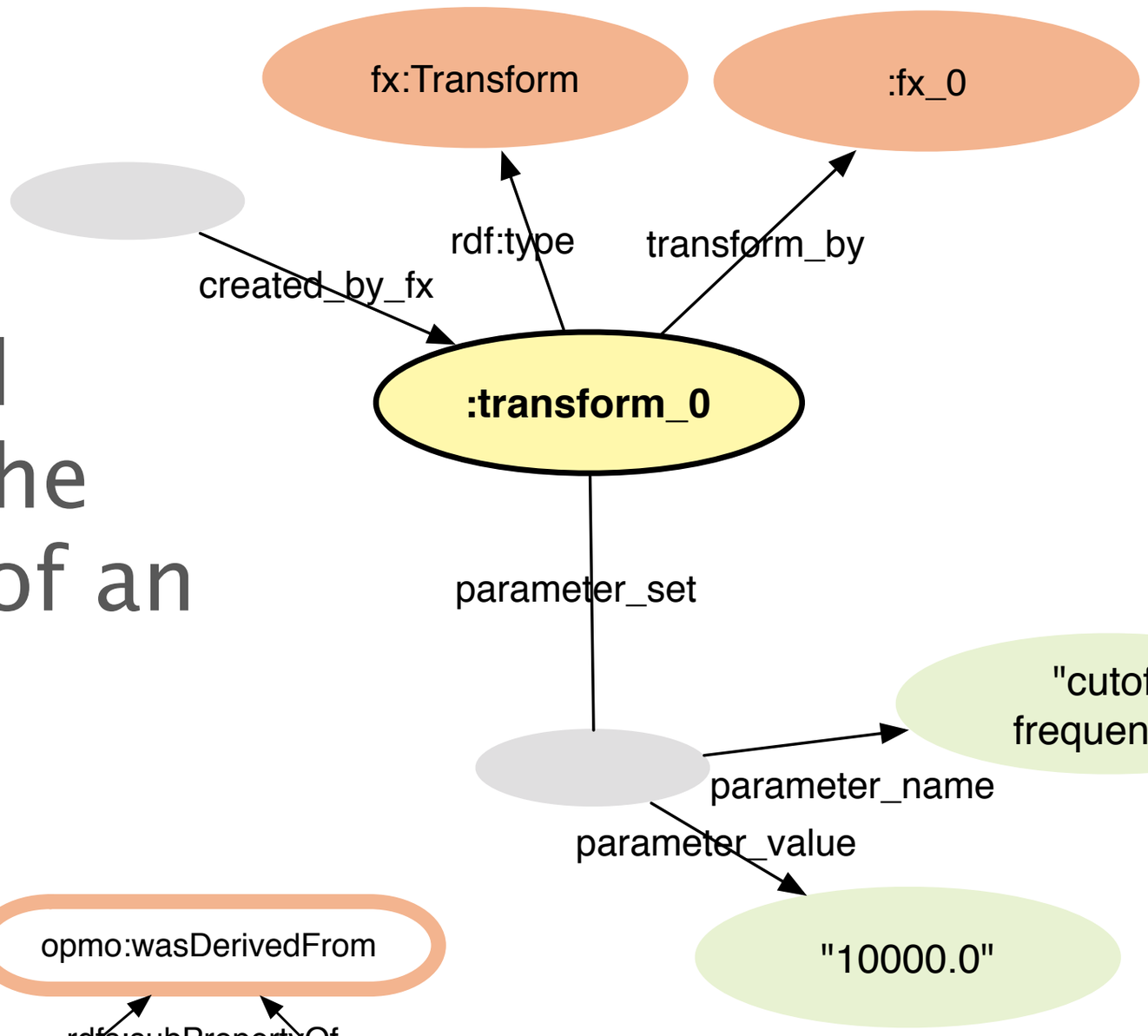
Audio Transformation

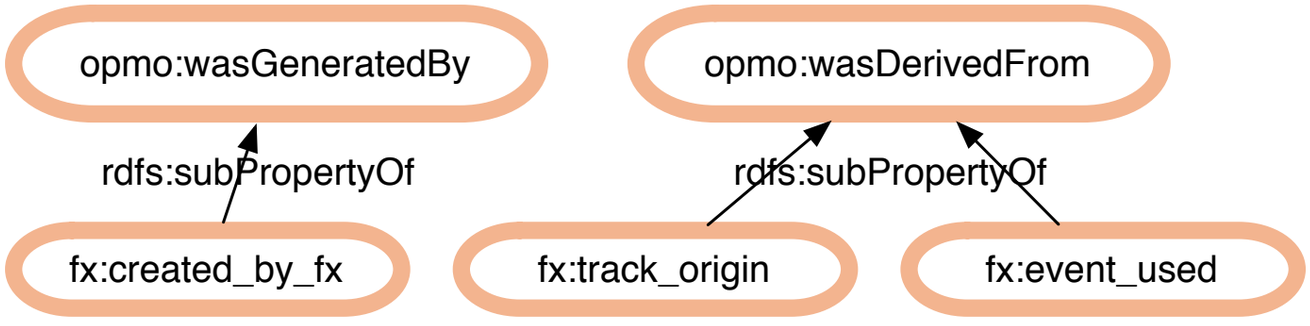
- event/signal created by the application of an effect



Audio Transformation

- event/signal created by the application of an effect
- provenance





SPARQL Query Example

Which events have been produced by an audio effect affecting loudness?

What is their track name in the original multitrack project?

SPARQL Query Example

Which events have been produced by an audio effect affecting loudness?

What is their track name in the original multitrack project?

SPARQL Query Example

Which events have been produced by an audio effect affecting loudness?

What is their track name in the original multitrack project?

```
SELECT ?time ?track WHERE {
       ?a event:time ?b ;
         fx:created_by_fx ?c ;
         fx:track_origin ?track.
       ?b tl:at ?time .
       ?c fx:transform ?d .
       ?d fx:implementation_of ?e .
       ?e fxp:main_attribute fxp:Loudness .
       }
```

SPARQL Query Example

Which events have been produced by an audio effect affecting loudness?

What is their track name in the original multitrack project?

```
SELECT ?time ?track WHERE {
       ?a event:time ?b ;
         fx:created_by_fx ?c ;
         fx:track_origin ?track.
       ?b tl:at ?time .
       ?c fx:transform ?d .
       ?d fx:implementation_of ?e .
       ?e fxp:main_attribute fxp:Loudness .
       }
```

SPARQL Query Example

Which events have been produced by an audio effect affecting loudness?

What is their track name in the original multitrack project?

```
SELECT ?time ?track WHERE {
       ?a event:time ?b ;
         fx:created_by_fx ?c ;
         fx:track_origin ?track.
       ?b tl:at ?time .
       ?c fx:transform ?d .
       ?d fx:implementation_of ?e .
       ?e fxp:main_attribute fxp:Loudness .
       }
```

SPARQL Query Example

Which events have been produced by an audio effect affecting loudness?

What is their track name in the original multitrack project?

```
SELECT ?time ?track WHERE {
       ?a event:time ?b ;
         fx:created_by_fx ?c ;
         fx:track_origin ?track.
       ?b tl:at ?time .
       ?c fx:transform ?d .
       ?d fx:implementation_of ?e .
       ?e fxp:main_attribute fxp:Loudness .
       }
```

SPARQL Query Example

Which events have been produced by an audio effect affecting loudness?

What is their track name in the original multitrack project?

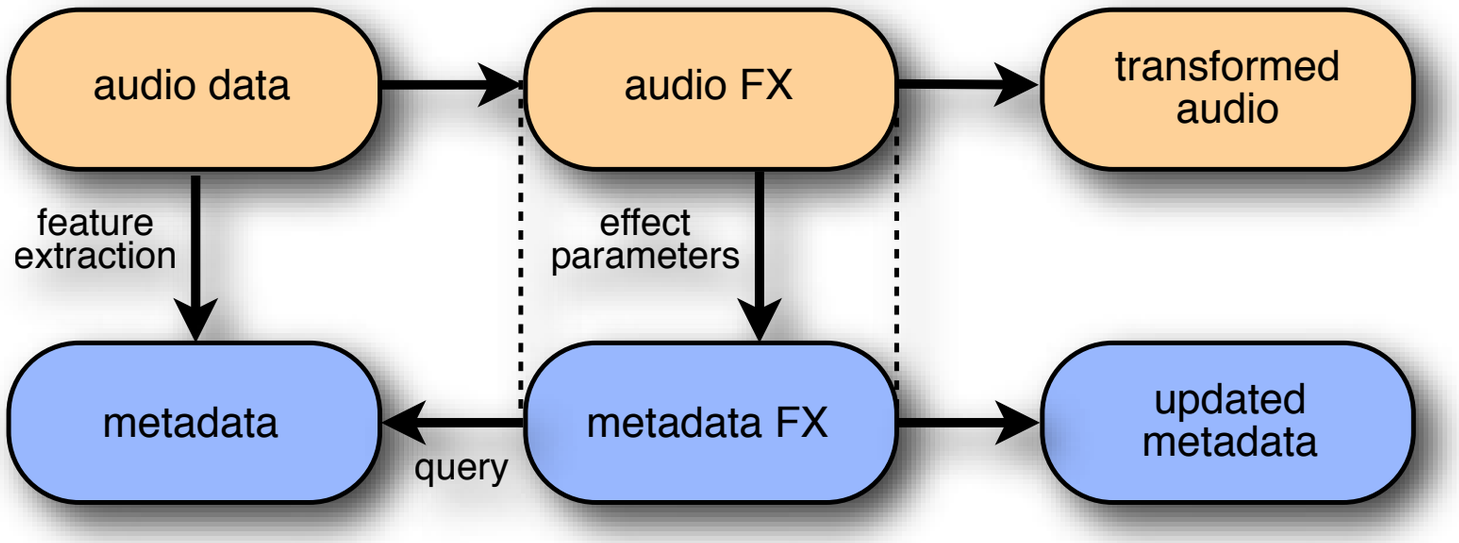
```
SELECT ?time ?track WHERE {
       ?a event:time ?b ;
         fx:created_by_fx ?c ;
         fx:track_origin ?track.
       ?b tl:at ?time .
       ?c fx:transform ?d .
       ?d fx:implementation_of ?e .
       ?e fxp:main_attribute fxp:Loudness .
       }
```

Predicting New Metadata

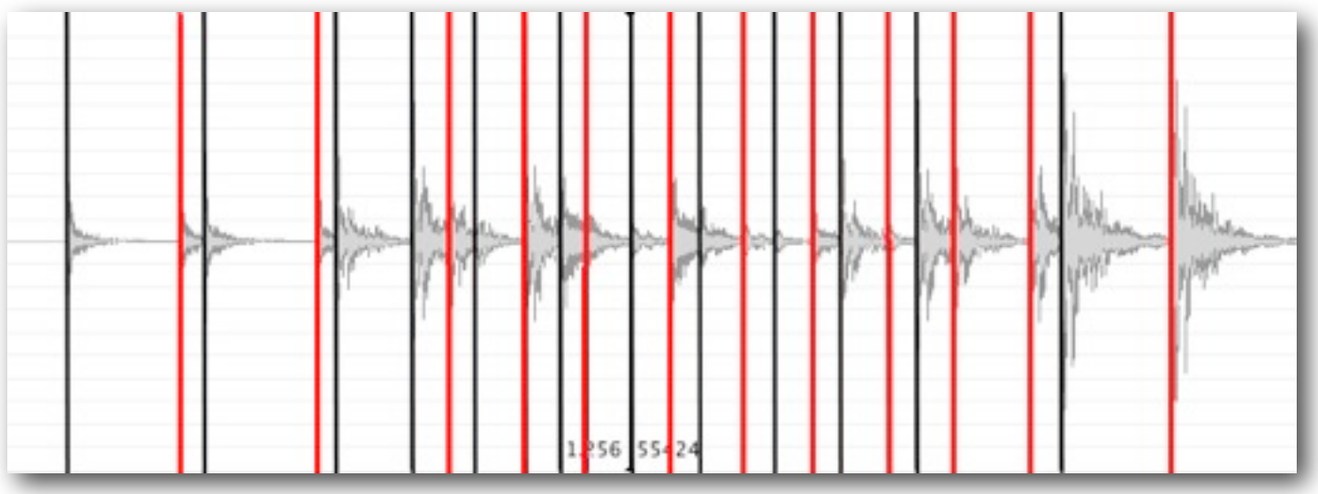
- feature extraction from effected files is inefficient
- instead: predict and accumulate metadata (where possible)
- use RDF and the Audio Effects Ontology

Predicting New Metadata

- feature extraction from effected files is inefficient
- instead: predict and accumulate metadata (where possible)
- use RDF and the Audio Effects Ontology



FX-Based Information Retrieval



```
:event_1 a af:Onset;
     event:time [ a tl:Instant;
             tl:at "PT2.194007S"^^xsd:duration;
             tl:onTimeLine :signal_timeline_0];
     fx:created_by_fx :transform_0;
     fx:event_used :event_0;
     fx:track_origin :drums.
```

FX-Database on the Semantic Web

FX-Database on the Semantic Web

- Large database on the Web: KVR Audio

Frohmage is a multi-band resonant filter. It can offer some unique sounds, from slow and deep filter sweeps to the most savage tones with high resonance setting and heavy distortion.

- Highly resonant low-pass filter.
- Cutoff frequency unit selection : Hz or musical note.

Product	**Frohmage**
Developer	**Ohm Force**
Price (MSRP)	Free
Type / Tags	Filter

Plug-in, App & Soundware Format(s)

Effect(s)	VST, RTAS, DX

Operating System Availability

Operating System	Latest Version	Download	Released
X64	1.60	Free Download	✓
64	1.60	Free Download	✓

FX-Database on the Semantic Web

- Large database on the Web: KVR Audio
  - HTML: Data is not easily reusable

FX-Database on the Semantic Web

- Large database on the Web: KVR Audio

  - HTML: Data is not easily reusable
  - Website format may change

FX-Database on the Semantic Web

- Large database on the Web: KVR Audio
  - HTML: Data is not easily reusable
  - Website format may change
  - No clear Semantics

FX-Database on the Semantic Web

- Large database on the Web: KVR Audio
  - HTML: Data is not easily reusable
  - Website format may change
  - No clear Semantics
- **KVR module for the FX Ontology**

FX-Database on the Semantic Web

- KVR module for the FX Ontology

```
:fx_0 a owl:Class, fx:PlugIn ;
  fx:implementation_of fx:Reverberation, kvr:Reverb ;
  dc:title "VariVerb Pro"^^xsd:string ;
  dc:creator "Magix"^^xsd:string ;
  rdfs:seeAlso "http://www.samplitude.com/eng/vst/variverb.html";
  fx:available_as fx:Vst ;
    gr:hasPriceSpecification
    [ a gr:UnitPriceSpecification ;
      gr:hasCurrency "USD"^^xsd:string ;
      gr:hasCurrencyValue "199"^^xsd:float ;
      gr:validThrough "2012-02-13T20:16:40"^^xsd:dateTime ] .
```

Applications of the FX Ontology

- **Music production**
  - detailed metadata creation

Applications of the FX Ontology

- **Music production**
  - detailed metadata creation
  - reproducibility of sound transformations

Applications of the FX Ontology

- **Music production**
  - detailed metadata creation
  - reproducibility of sound transformations
  - recommendation of similar audio effects and settings

Applications of the FX Ontology

- **Music production**
  - effect search by high level semantic descriptors
  - perceptual/technical descriptors
  - link to data on the Semantic Web

Applications of the FX Ontology

- **Music production**
  - effect search by high level semantic descriptors
    - perceptual/technical descriptors
    - link to data on the Semantic Web
  - semantic metadata as control input for adaptive audio effects

Applications of the FX Ontology

- **Musicological research**
  - production tendencies of genres/eras
  - more detailed descriptors due to retention of multitrack and transform-specific metadata

Summary

- The use of Semantic Web technologies enable Semantic Audio applications that link and scale like the Web itself.

- New applications using a mash-up of data sources

- Provide interoperability between tools in music information sciences and music production

Summary

- **Future work**
- Release large datasets using these ontologies
- Consider a broader set of use cases
- Harmonisation with standards
- Work towards a Semantic Audio Desktop