TIMBUS

TIMELESS BUSINESS

Context and Dependency Modelling & Extraction

Rudolf Mayer

Secure Business Austria, Vienna
http://www.sba-research.org

rmayer@sba-research.org

Outline

- Introduction and Motivation
- Example use case – Scientific Experiment

- Modelling process context
  - Modelling the example use case
- Extended process context model
  - Modelling the example use case

- Automatic context extraction & reasoning

Introduction

- Motivation for process preservation
  - Provenance, Litigation, Regulated industries
- Example: preservation of a scientific process
  - Repeatability, traceability, quality assurance
    - ➔ Scientific excellence
- Challenges
  - Dependencies on third-party software
    - commercial or open source
  - Dependency on services: SaaS, IaaS, …
  - Changes in computing environment
    - Use of multiple servers, grids, ….

Introduction

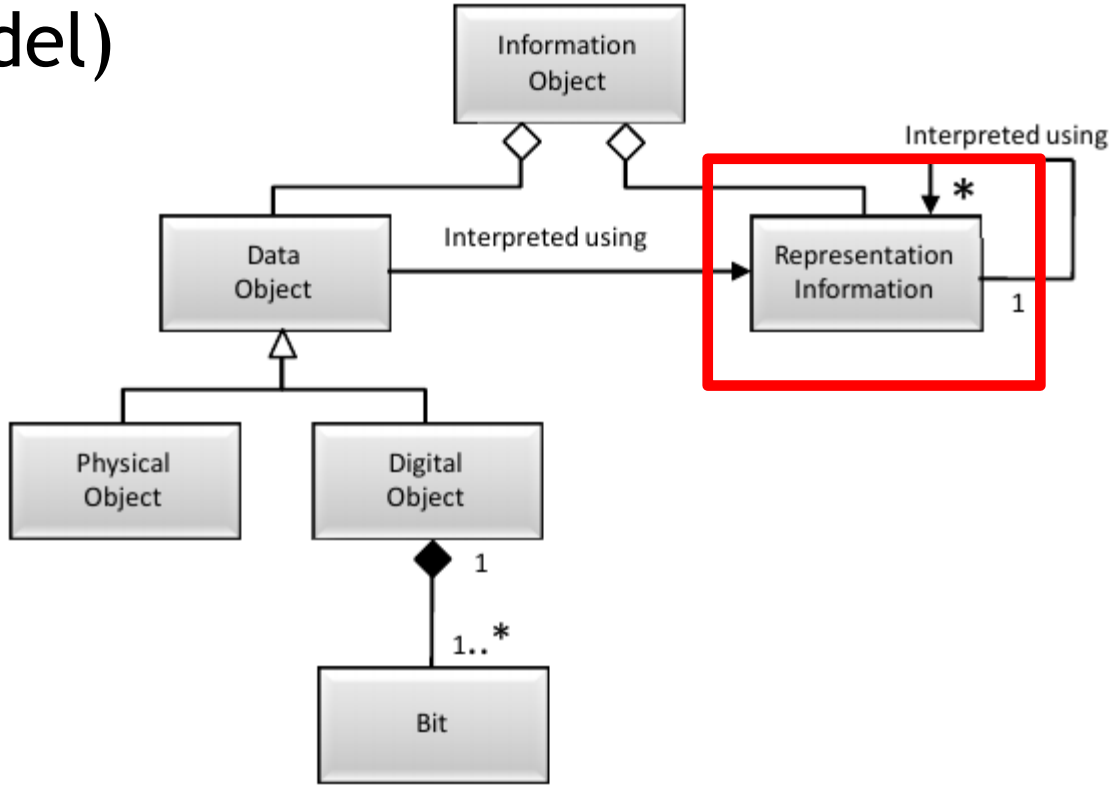
- To preserve the process:
  need to describe the context it is embedded in

- Aspects such as
  - Legal / social
  - Organisational
  - Technical (Software, Hardware Infrastructure)

Context Model Principles

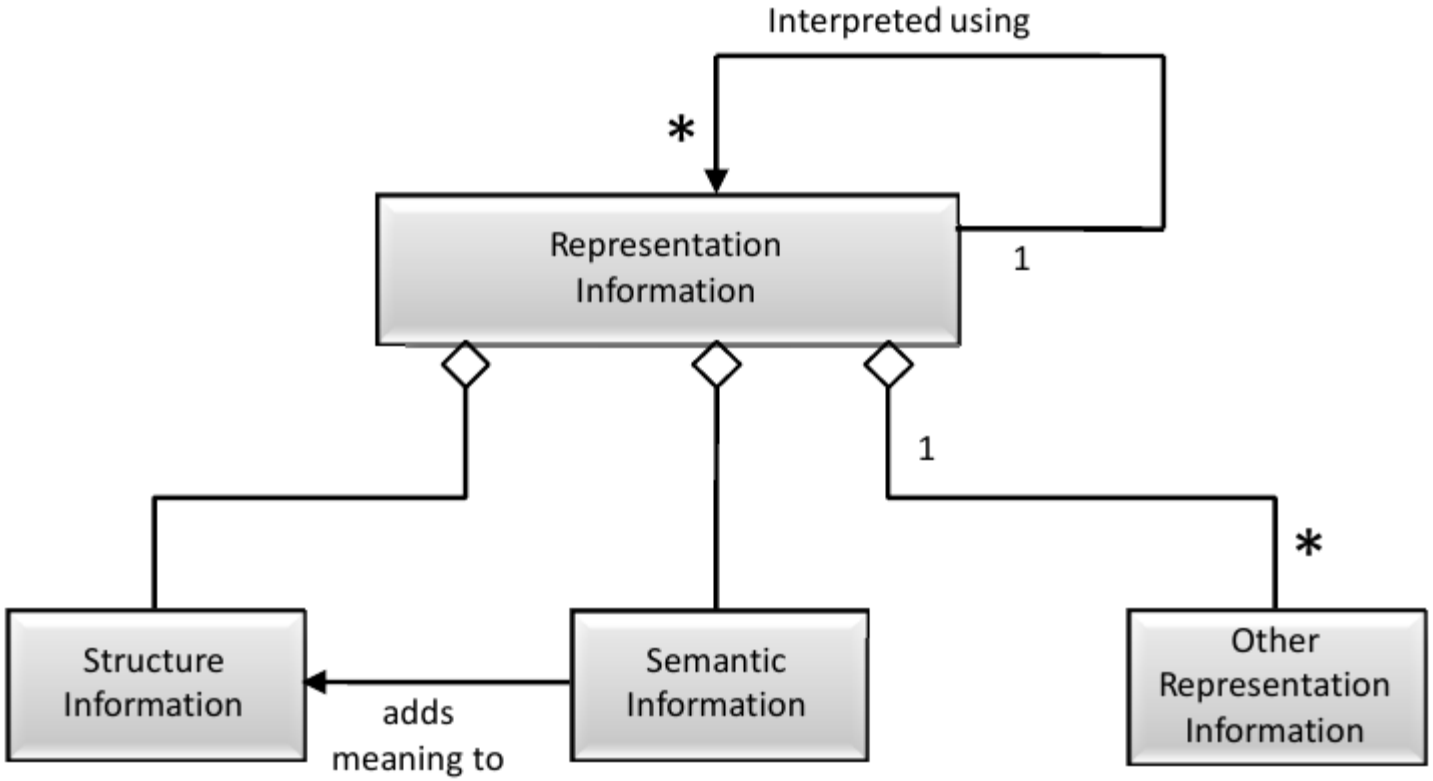
TIMELESS BUSINESS

- Process context can be understood as representation information (OAIS reference model)



Context Model Principles

- Process context can be understood as representation information



Introduction

- To preserve the process: need to describe the context it is embedded in
  - Aspects such as legal, organisational, technical

- ➔ Need a meta-model that
  - Supports ***structured*** modelling of diverse aspects
  - Can be easily extended for various domains
  - Can be used during
    - preservation,
    - redeployment and
    - analysis

Outline

- Introduction and Motivation
- Example use case – Scientific Experiment
- Modelling process context
  - Modelling the example use case
- Extended process context model
  - Modelling the example use case
- Automatic context extraction & reasoning

Example Use Case

- Information Retrieval / Machine Learning Experiment
- Classification of music into a predefined set of categories, e.g. genre
- Evaluation of classification performance
- Common task in Music Information Retrieval
  - e.g. in MIREX evaluation campaign

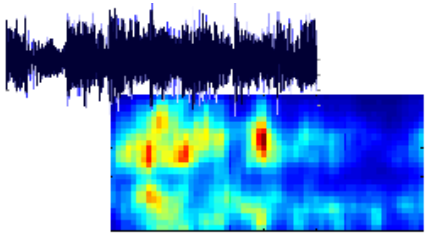
Example Use Case

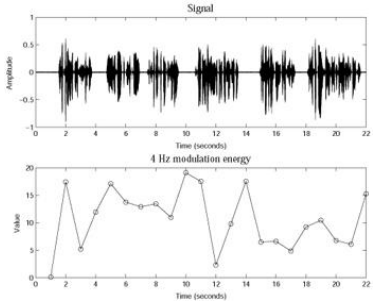
TIMELESS BUSINESS

- Involves several steps
  - Acquisition of music data

  - Acquisition of ground-truth, i.e. gold standard of genres for the music

  - Extraction of features (numerical representation)

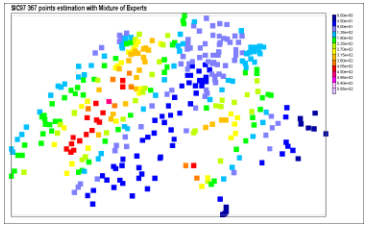




Example Use Case

TIMELESS BUSINESS

- Involves several steps
  - Acquisition of music data
  - Acquisition of ground-truth
  - Extraction of features (numerical representation)



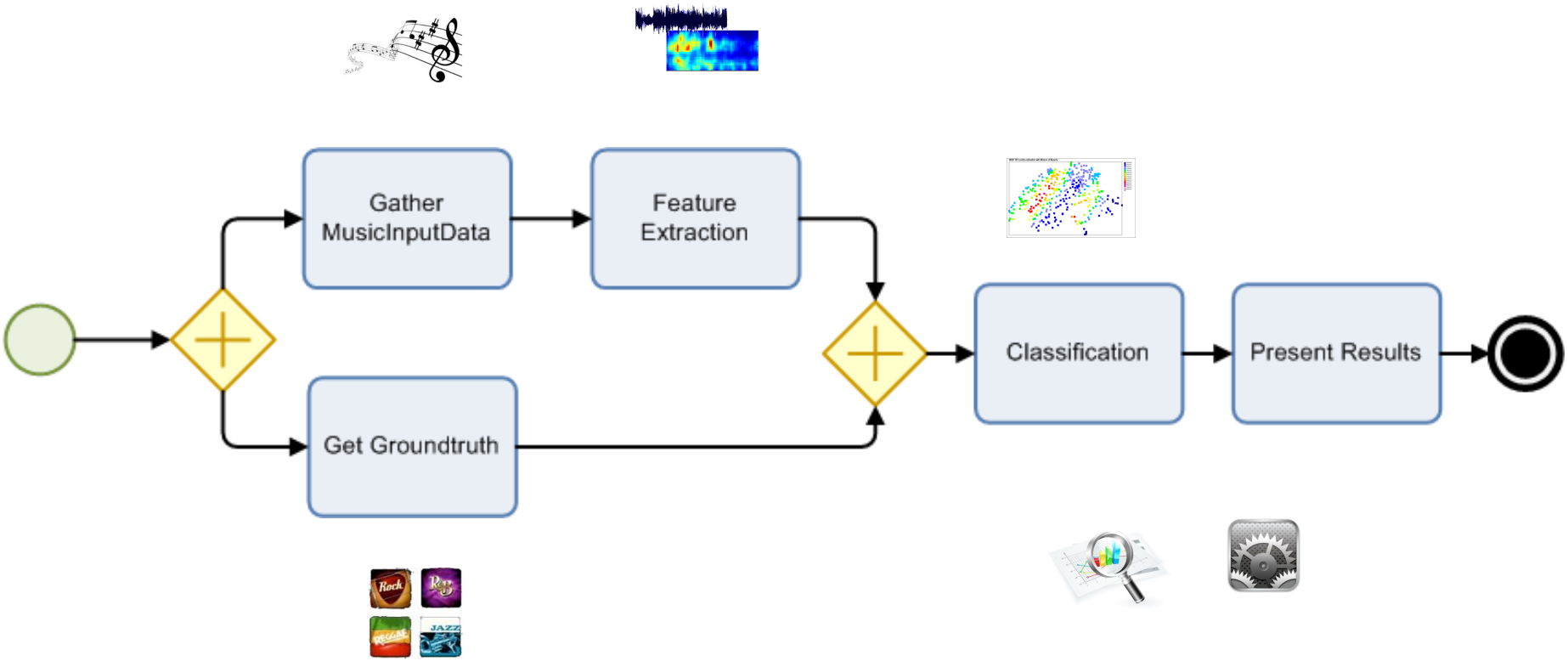
  - Training of machine learning model

  - Analysis of classification performance

  - Subsequent repetition with slight variations, finally leading to publication

Example Use Case

TIMELESS BUSINESS



Example Use Case

TIMELESS BUSINESS

- Questions relevant for the preservation
  - What steps are executed?
    - What is the order of the steps (and is it important?)
  - What is the input & output of each step?
    - What format is that data in? Format risks?
  - What is the software supporting each step?
    - What is the hardware supporting the software / the step?
  - Are there any external services?
    - Are there any other dependencies?
  - Am I allowed to preserve everything?

Example Use Case implementation

- Acquisition of music data & ground truth
  - External data repositories
- Extraction of features
  - External web service (e.g. from the.echonest.com)
- Training of machine learning model
  - Using WEKA machine learning software

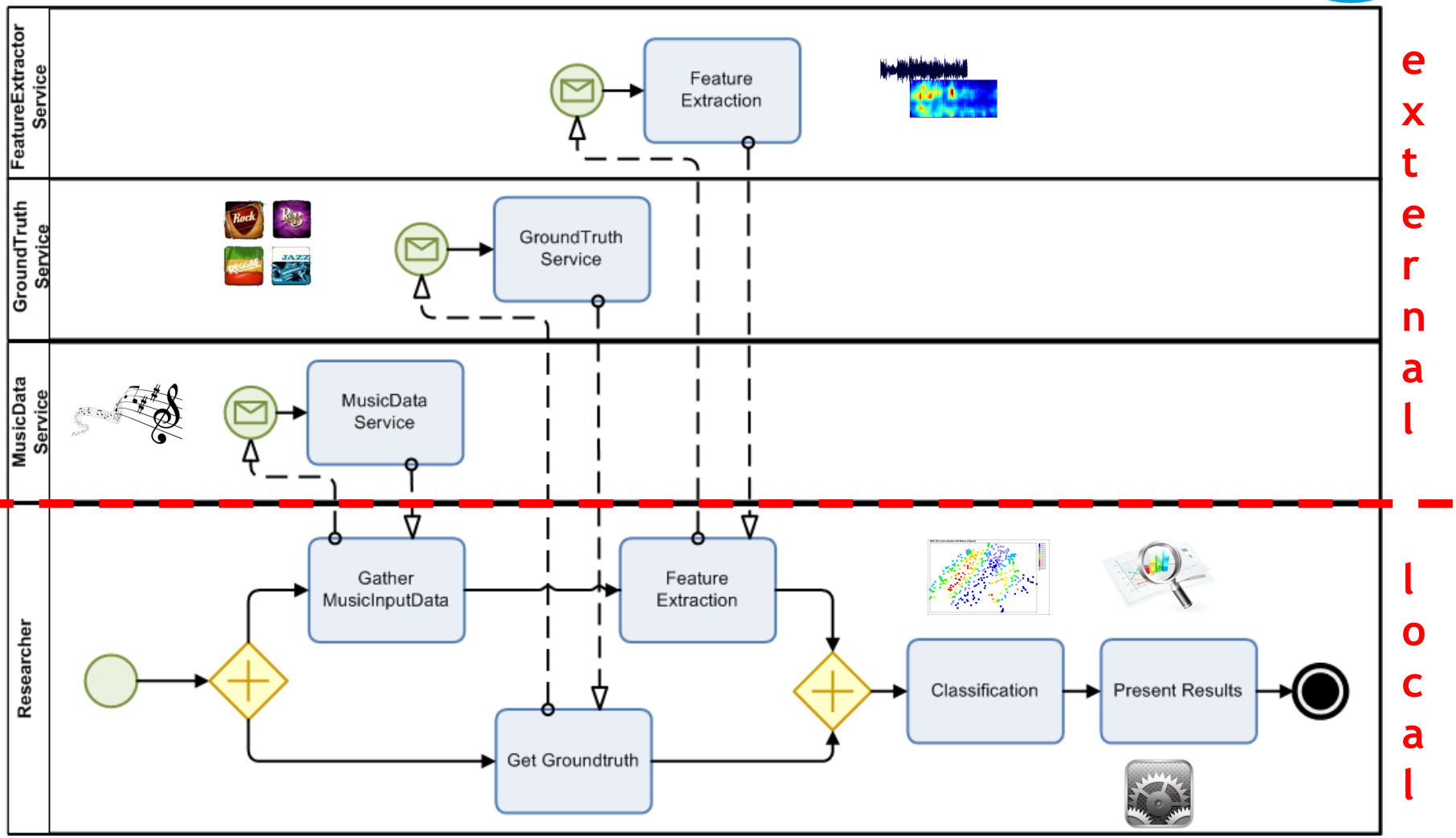
- Other steps with custom scripts
  - Linux shell (bash)
  - Java programs

Example Use Case

- Input
  - Music data (or location & settings where to fetch it from)
  - Ground-truth (or same as above)
  - Settings for feature extraction & classification
- Output
  - Genre classification accuracy
  - Detailed classification results

Example Use Case

TIMELESS BUSINESS



Example Use Case: Motivation

TIMELESS BUSINESS

- Motivation to preserve the experiment
  - Personal: repeatability of results at later time
    - Subsequent publications that use the results
    - Accountability of the results
  - Institutional policy
    - Demands verifiability
    - Encourages sharing of results
  - External reviewers
  - Process management plan as extension to a data management plan

Example Use Case: Risks

TIMELESS BUSINESS

- Risks in the use case
  - Acquisition of data from external sources
    - Might become unavailable or change (e.g. different recording, different ground-truth assignment)
  - External service for the extraction of features (numerical representation)
    - Might become unavailable, might change computation, …
  - Training of machine learning model with third-party software ➔ Technological dependencies
  - Distributed documentation / Lack of documentation

Example Use Case: Risks

- Consequences
  - Missing validation of results in the future
  - Loss of trust on existing research data
  - Loss of expert knowledge
  - Loss of scientific results

➔ Loss of scientific reputation

Outline

TIMELESS BUSINESS

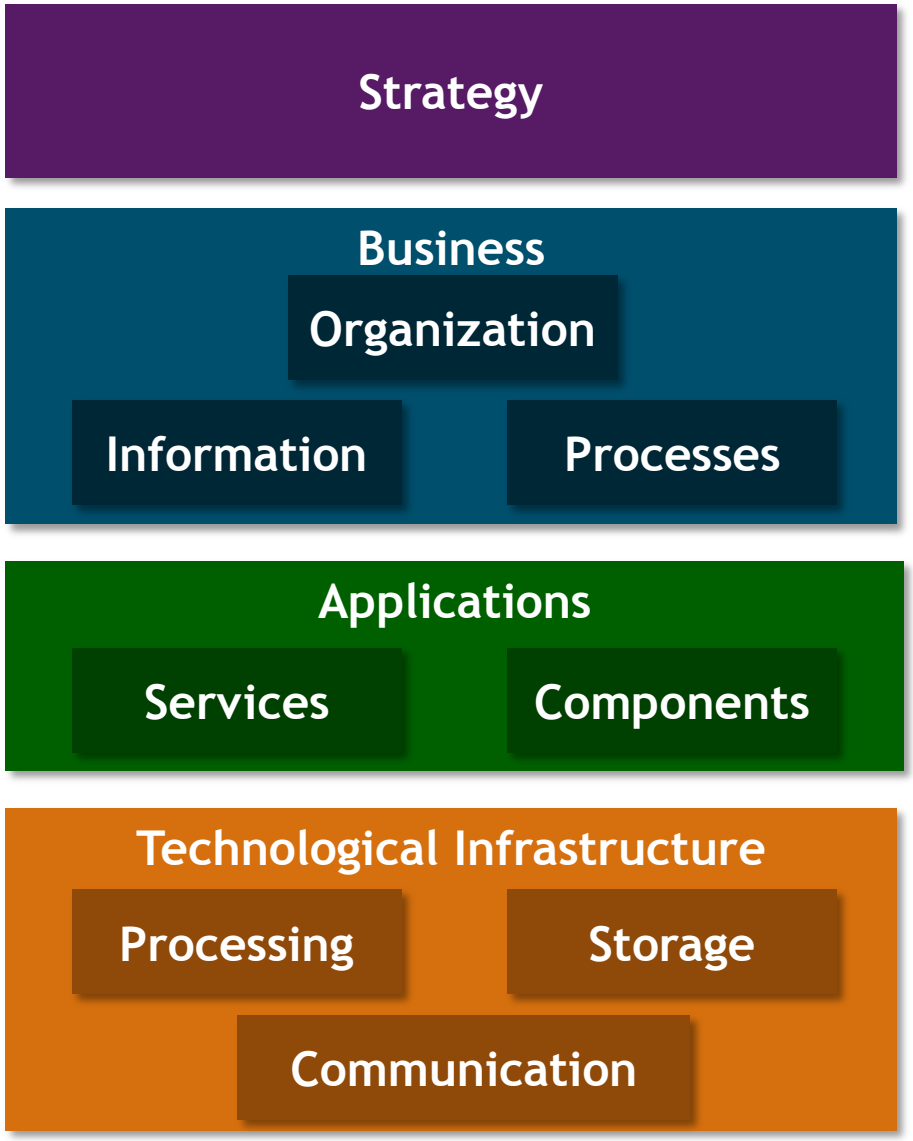
- Introduction and Motivation
- Example use case – Scientific Experiment
- Modelling process context
  - Modelling the example use case
- Extended process context model
  - Modelling the example use case
- Automatic context extraction & reasoning

Context Model Layers

- Process context exists on different (partially orthogonal) layers
  - Strategic
  - Business
  - Organization, Operations
  - Information systems
  - Technological infrastructure

Context Model Layers

TIMELESS BUSINESS



Strategic Indicators, External Services, Contracts, Regulations, Licenses, Legal Requirements, Patents

Organizational Structure, People, Business Processes, Operational Indicators

Applications, Services, Virtualization Applications

Deployed software applications and services, Hardware nodes, Communication nodes

Context Model Principles

- All layers should be captured in the same (meta-) model
- Model should be extensible and adaptable to new application domains
- One-model fits all approach doesn't work
  - Integration of several smaller models

Context Model Principles

1. Describe features not specific to the domain (domain-independent features).

2. Describe domain-specific features.

3. Integrate the multiple domains into a consistent computable specification.

4. Support analysis of the specifications in terms of their dependencies.

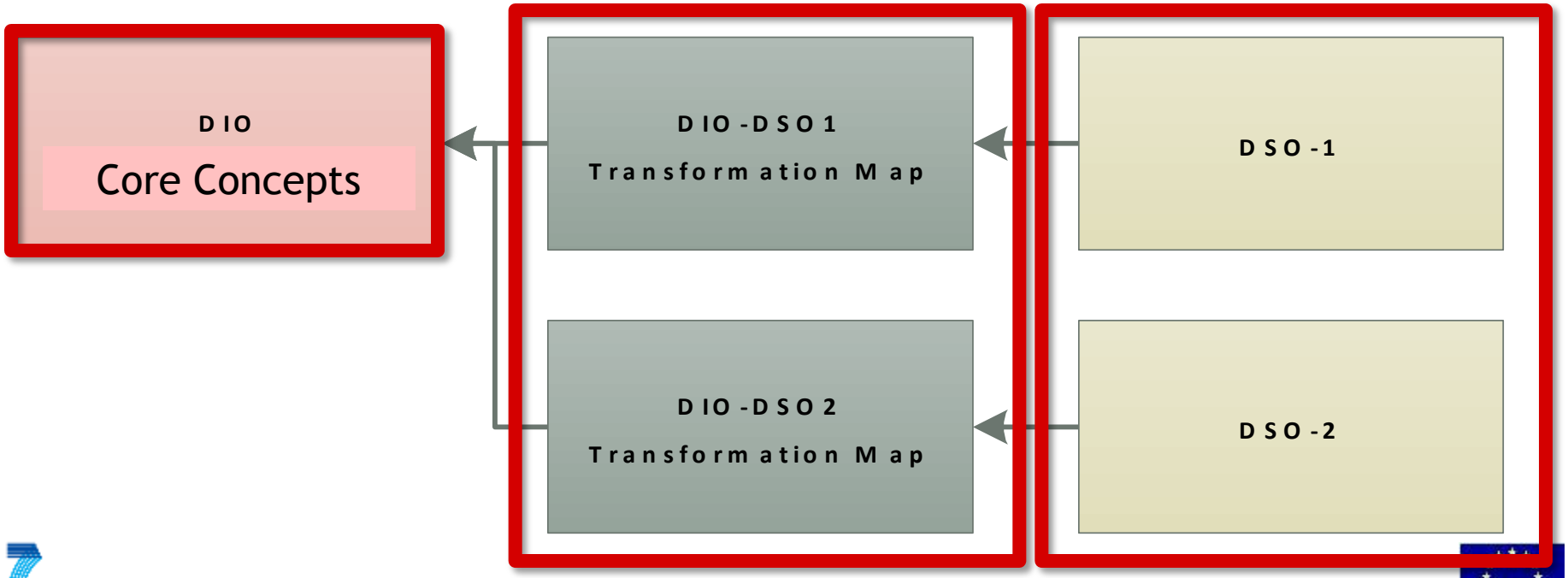
Model Representation

- The context model is represented as a set of ontologies (W3C OWL standard language)
- Ontologies formalize knowledge representation
- Information can be extracted from ontologies through querying and processing (e.g. reasoning, logical inference)

Architectural Concepts

TIMELESS BUSINESS

- **DIO: Domain-Independent Ontology**
- **DSO: Domain-Specific Ontology**
- **Ontology integration** (transformation maps)
- **Model transformation and extraction**



Domain Independent Ontology

- The DIO is grounded on the **ArchiMate 2.0** enterprise architecture modelling language.
- An "The Open Group" vendor-independent international standard.
- Well-founded concepts.
- Consistent language containing a lean set of concepts for modelling enterprise architecture.

The ArchiMate Core Concepts

- **Active structure** is an entity (e.g. person, machine) capable of performing behaviour
- **Behaviour** is a unit of activity (e.g. process, application function) performed by active structure elements
- **Passive structure** is an object on which behaviour is performed (e.g. product, document)

The ArchiMate Framework

- ArchiMate framework is **layered** and **service-oriented**
- **Business layer**: products & services offered to customers
  - Realised in the organization by business processes performed by actors
- **Application layer:** information systems that operationalise the business though application services
- **Technology layer:** processing, storage & communication nodes that support the application layer.
  - Offers infrastructure services needed to run applications
  - Realized by computer hardware and system software.

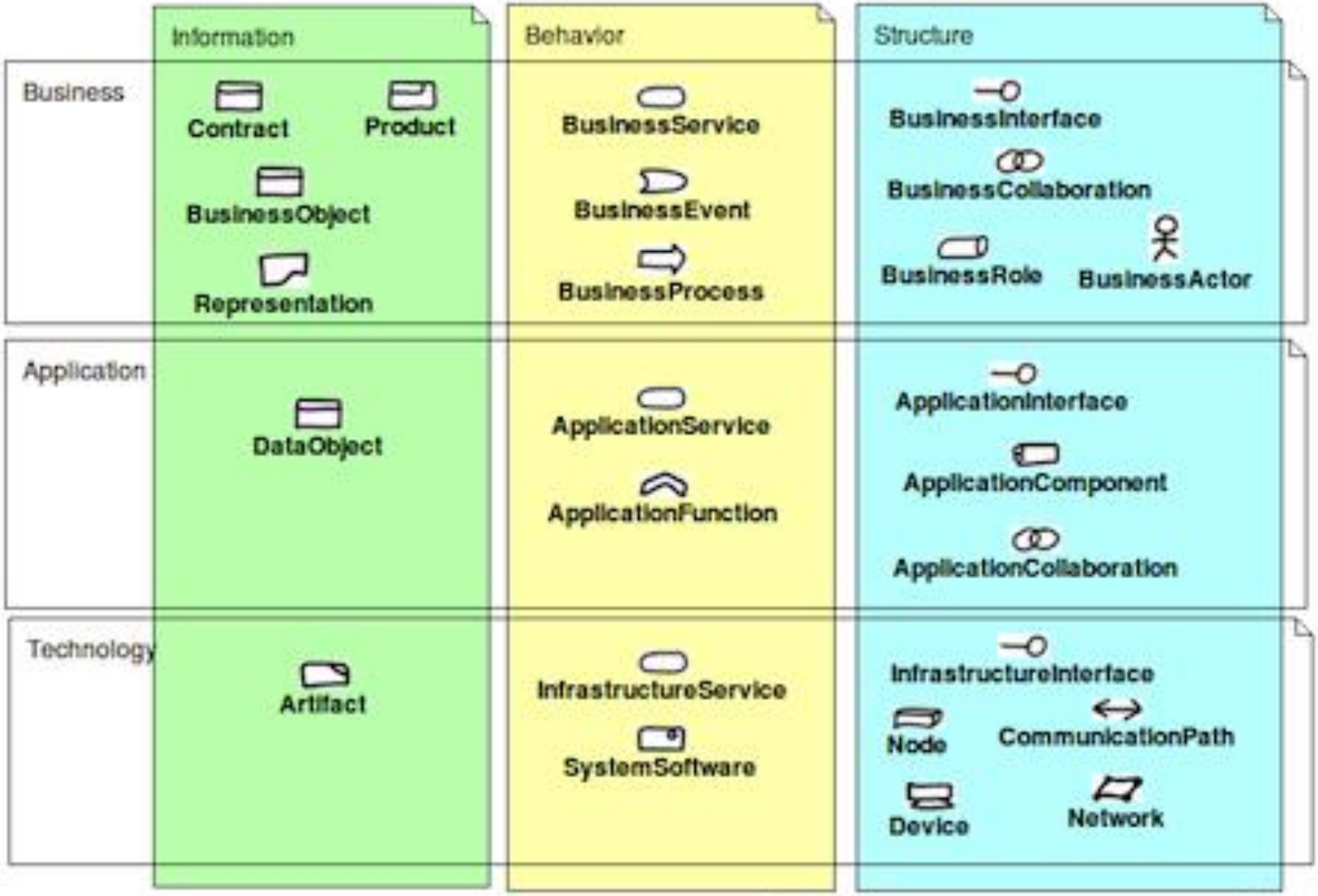
The ArchiMate Framework

TIMELESS BUSINESS

	Passive Structure	Behaviour	Active Structure
Business	business objects	business services and processes	actors and roles
Application	data objects	application services and functions	applications and components
Technology	artifacts	infrastructure services and system software	devices and networks

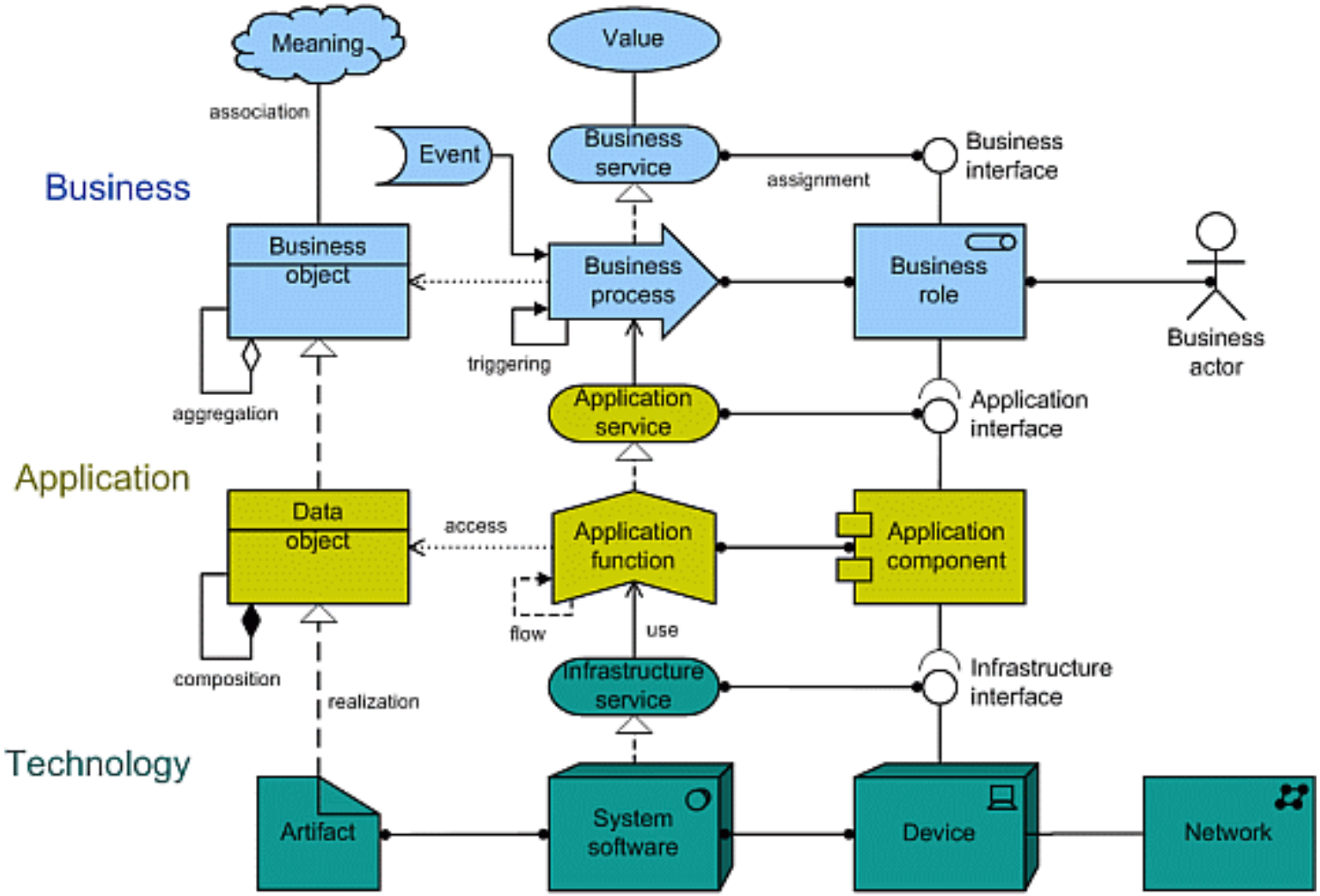
The ArchiMate Framework

TIMELESS BUSINESS



The ArchiMate Framework

TIMELESS BUSINESS



The ArchiMate Framework

- Total of 32 concepts
- Business Layer: 16 concepts
  - Actor, Role, Collarboration, Interface, Function, Process, Event, Interaction, Product, Contract, Service, Value, Meaning, Representation, Object, Location
- Application Layer: 7 concepts
  - Component, Collaboration, Interface, Service, Function, Interaction, Data Object
- Technology Layer: 9 concepts
  - Artifact, Communication Path, Network, Infrastructure Interface, Infrastructure Function, Infrastructure Service, Node, System Software, Device

ArchiMate DIO

TIMELESS BUSINESS

Class hierarchy: BusinessFunction

- ArchimateConcept
  - ApplicationCollaboration
  - ApplicationComponent
  - ApplicationFunction
  - ApplicationInteraction
  - ApplicationInterface
  - ApplicationService
  - Artifact
  - Assessment
  - BusinessActor
  - BusinessCollaboration
  - BusinessEvent
  - BusinessFunction
  - BusinessInteraction
  - BusinessInterface
  - BusinessObject
  - BusinessProcess
  - BusinessRole
  - BusinessService
  - CommunicationPath
  - Constraint
  - Contract
  - DataObject

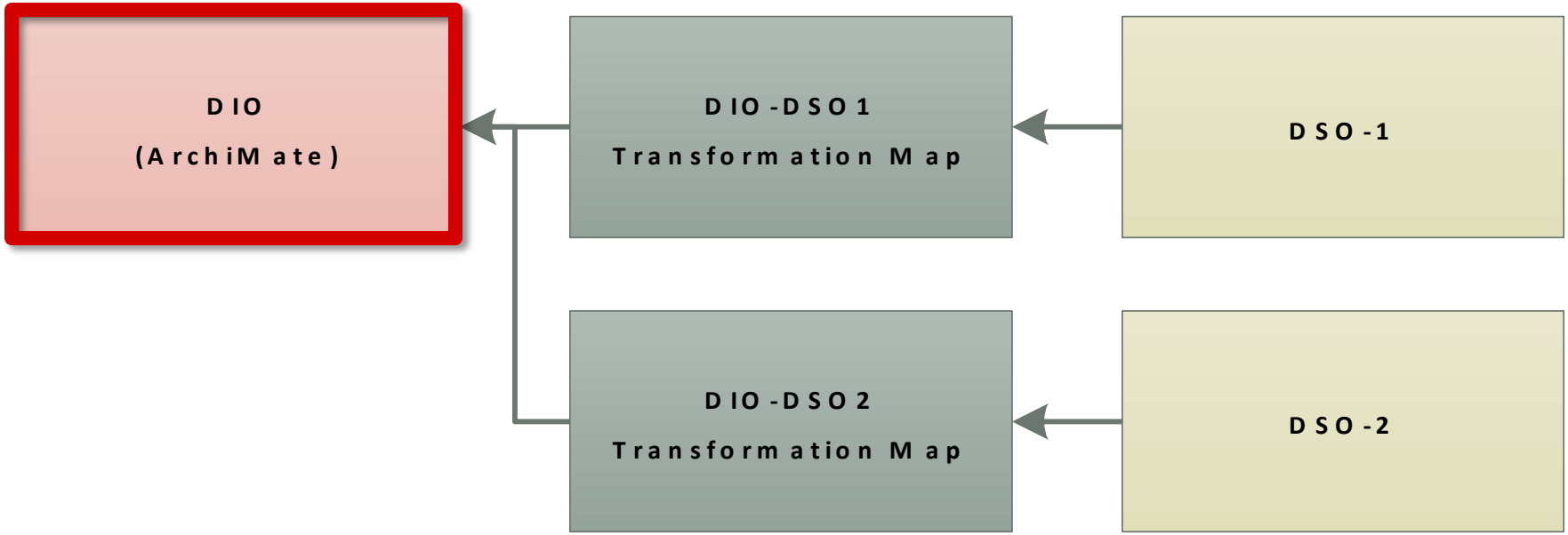
Description: BusinessFunction

- flowTo only
  (ApplicationCollaboration
  or ApplicationComponent
  or BusinessActor
  or BusinessCollaboration
  or BusinessFunction
  or BusinessInteraction
  or BusinessInterface
  or BusinessProcess
  or BusinessRole
  or Junction)
- hasAspect exactly 1 Thing
- hasAspect some BehavioralAspect
- hasLayer exactly 1 Thing
- hasLayer some BusinessLayer
- realizes only
  (BusinessService
  or Constraint
  or Goal
  or Principle
  or Requirement)
- specialization only
  (BusinessFunction

ArchiMate DIO

TIMELESS BUSINESS

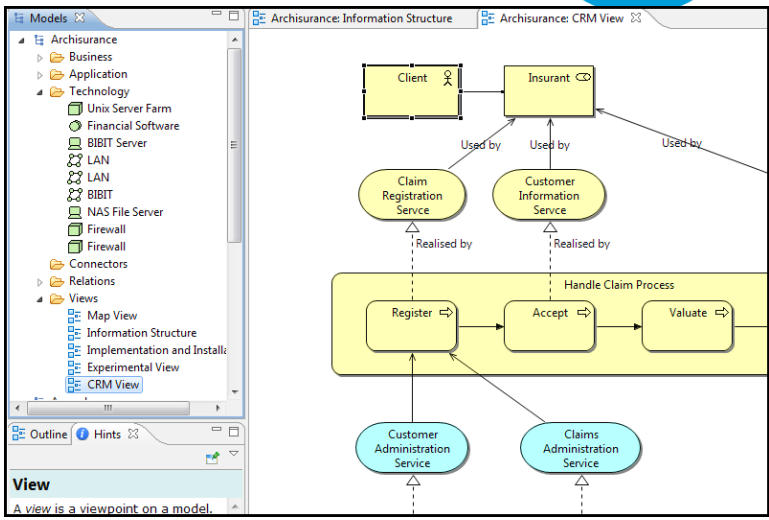
- **DIO: Domain-Independent Ontology**
- DSO: Domain-Specific ontology



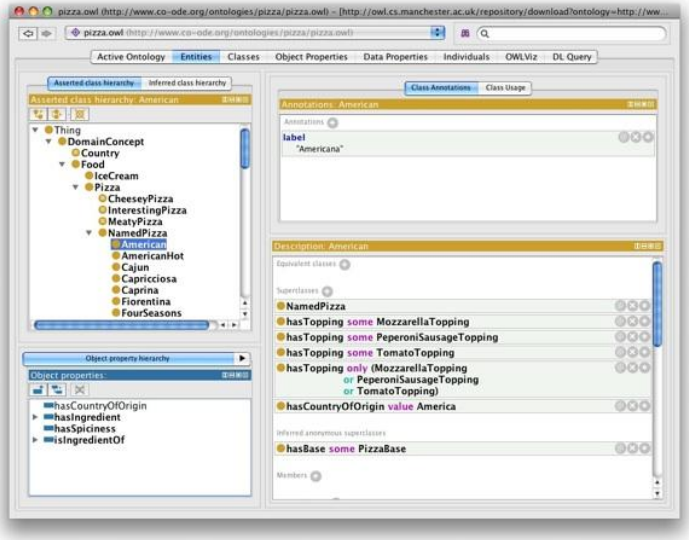
Tool Support

TIMELESS BUSINESS

- Archi – Archimate 2.0 editor ➔ DIO editor
  - open source (Eclipse)
  - Converter from Archi to OWL for further editing



- Protege – open source ontology editor
  - Plugins to allow visual editing



Outline

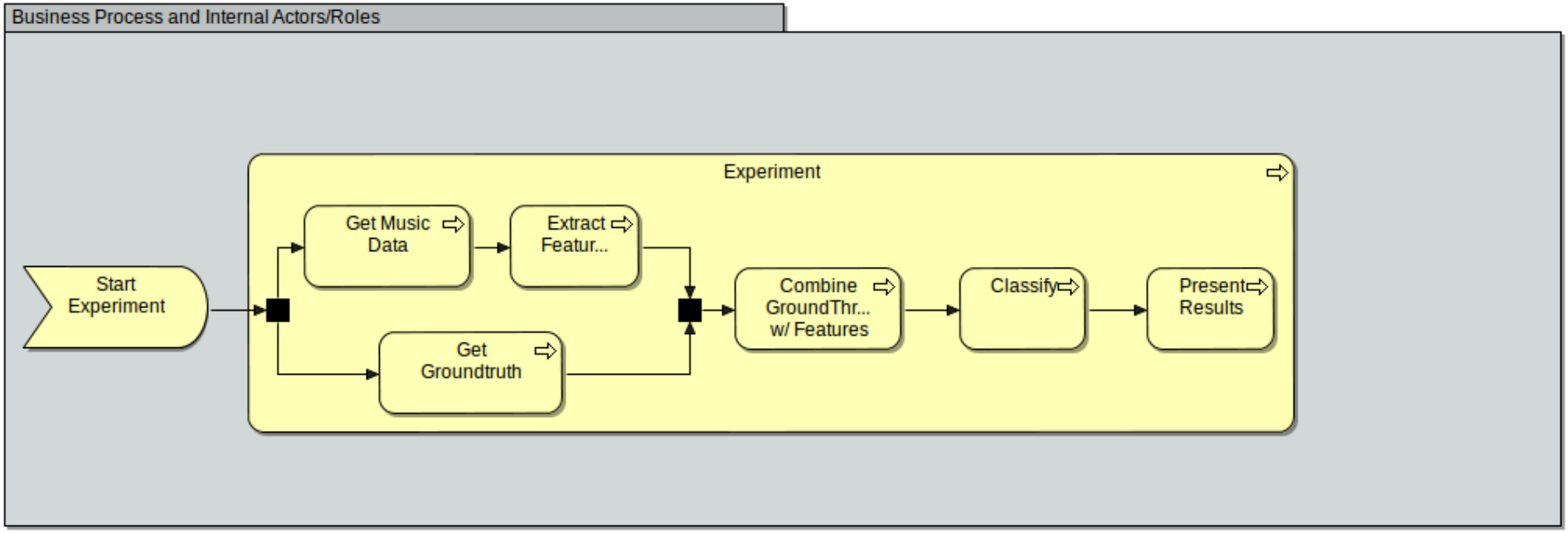
- Introduction and Motivation
- Example use case – Scientific Experiment
- Modelling process context
  - Modelling the example use case
- Extended process context model
  - Modelling the example use case
- Automatic context extraction & reasoning

DIO for Music Classification Example

- Approach: Identify elements on each layer
  - Business, Application, Technology
- Result: coarse description of the context
  - Needs to be refined with DSO specific knowledge, later

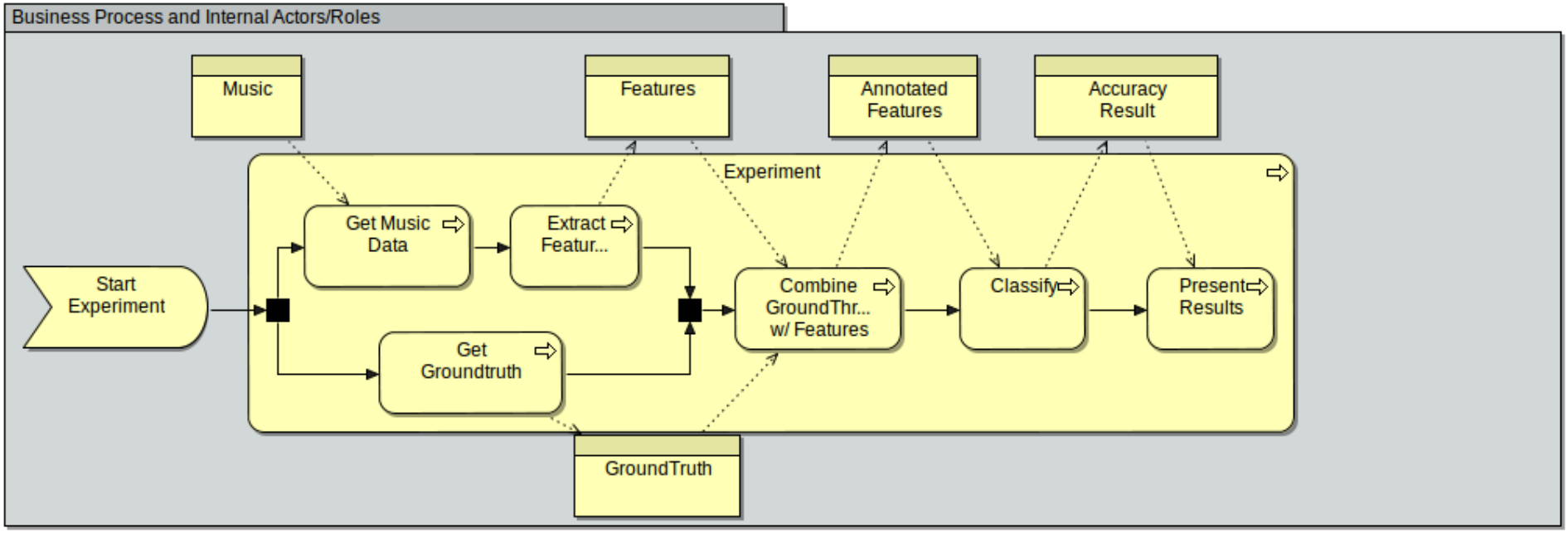
Business Layer: Process

TIMELESS BUSINESS



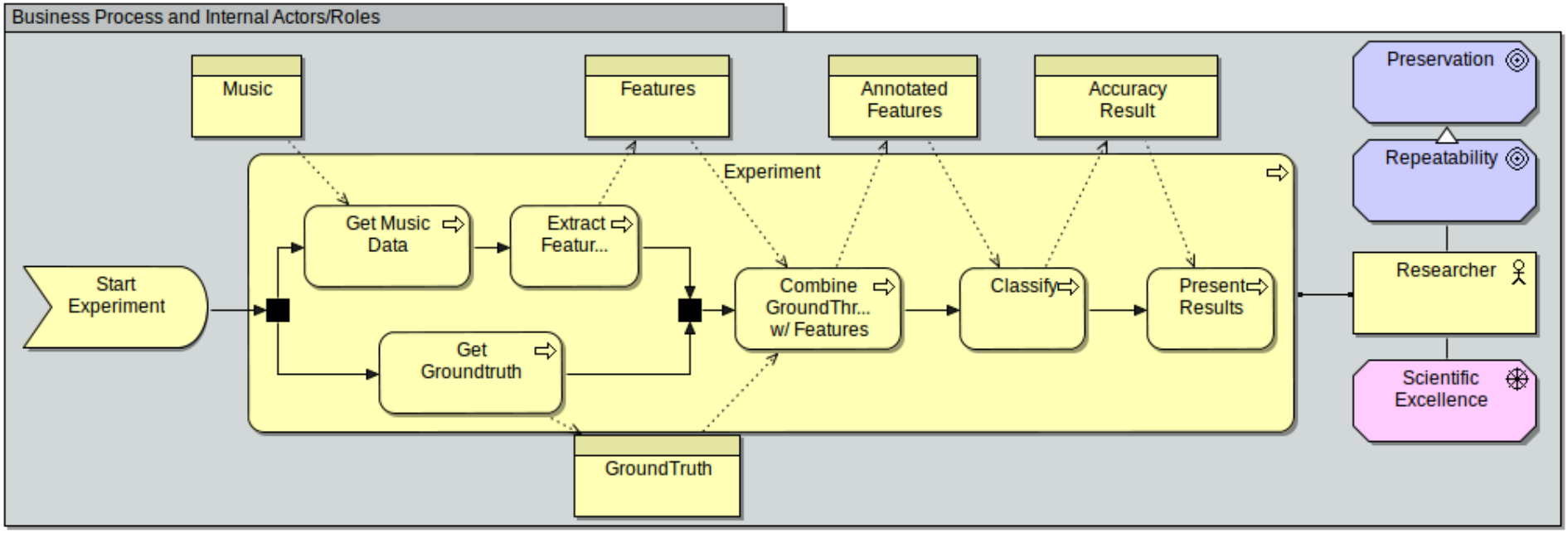
Business Layer: + Objects

TIMELESS BUSINESS



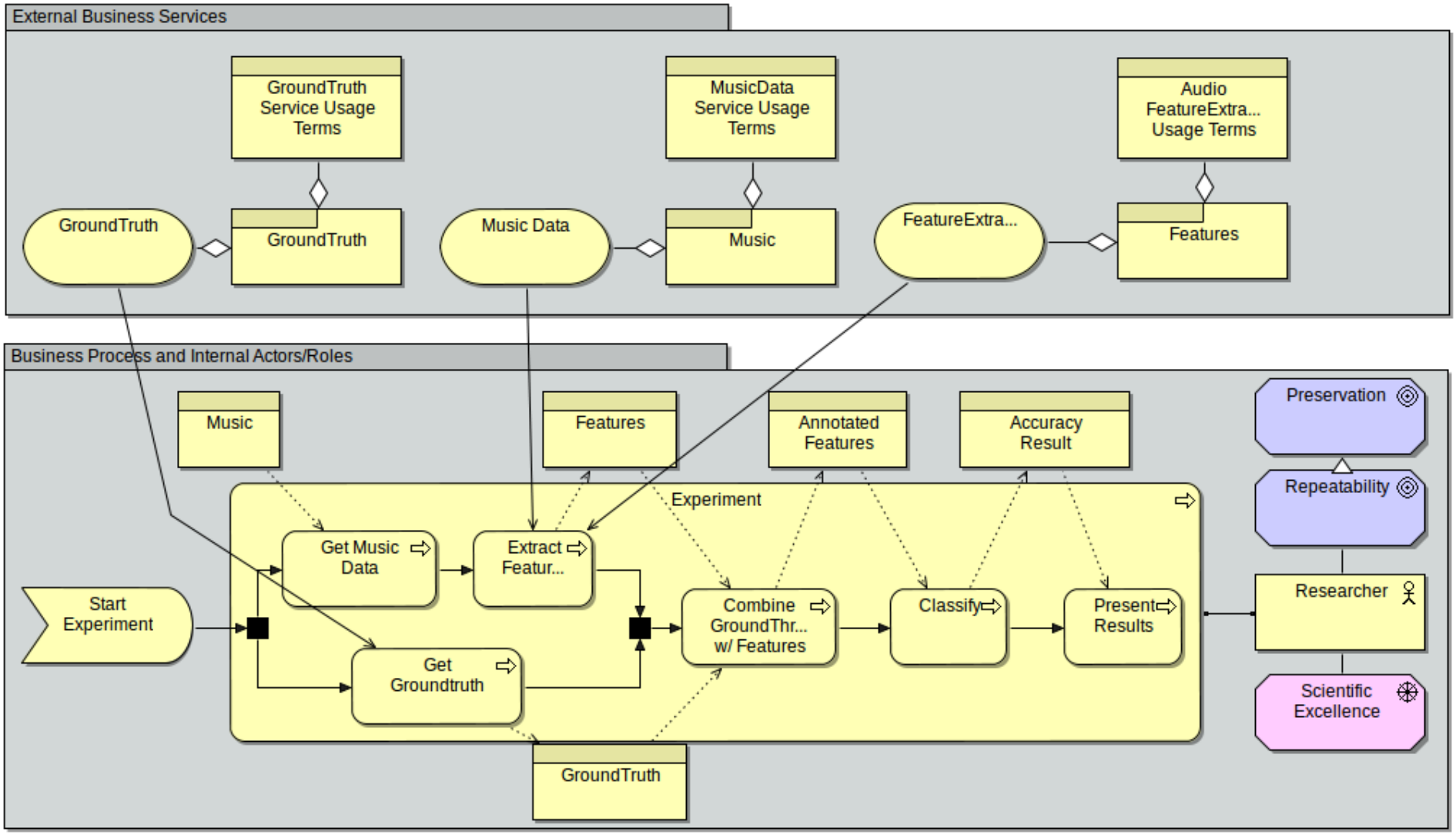
Business Layer: + Actors & goals

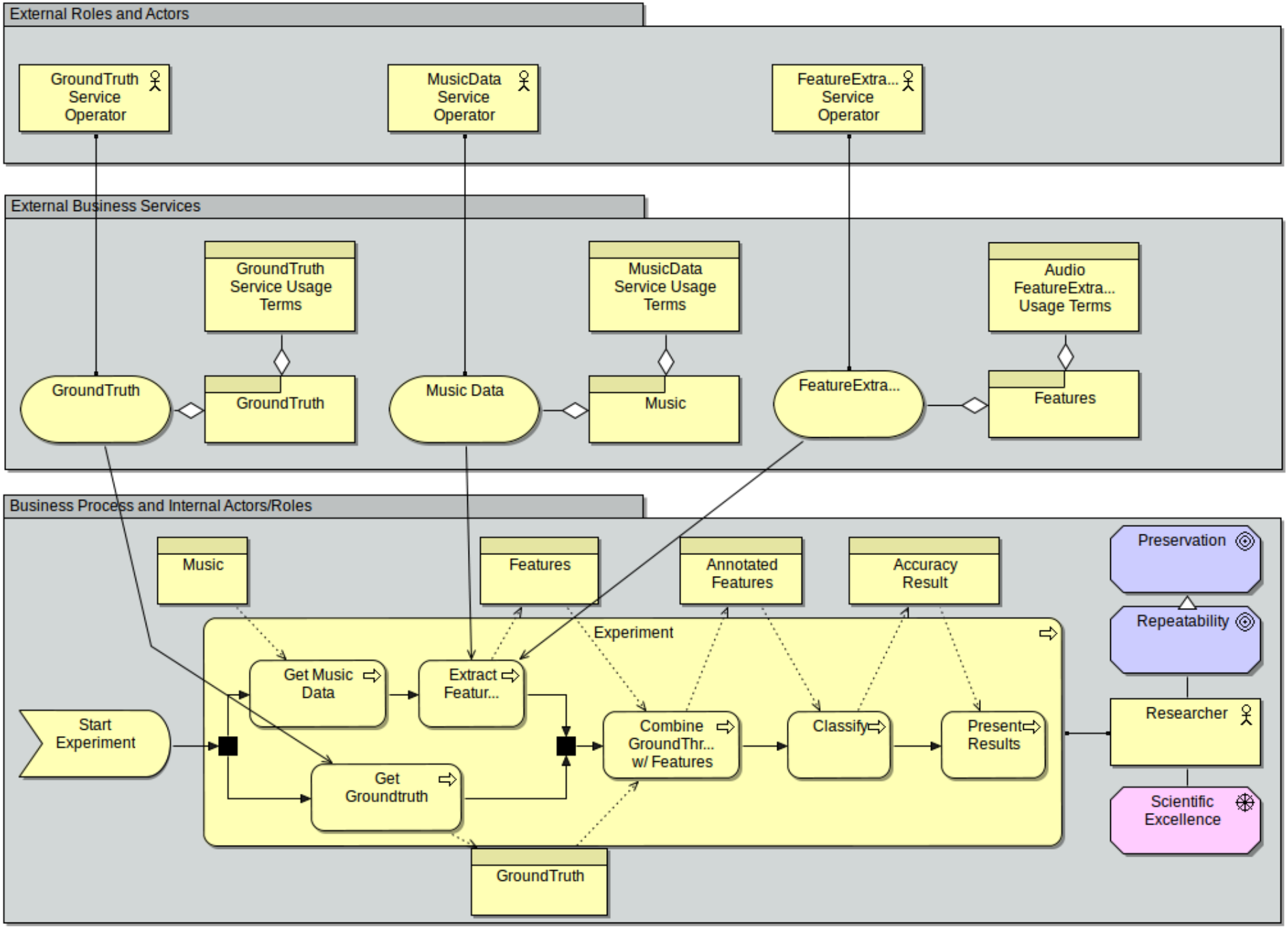
TIMELESS BUSINESS



Business Layer: + Services

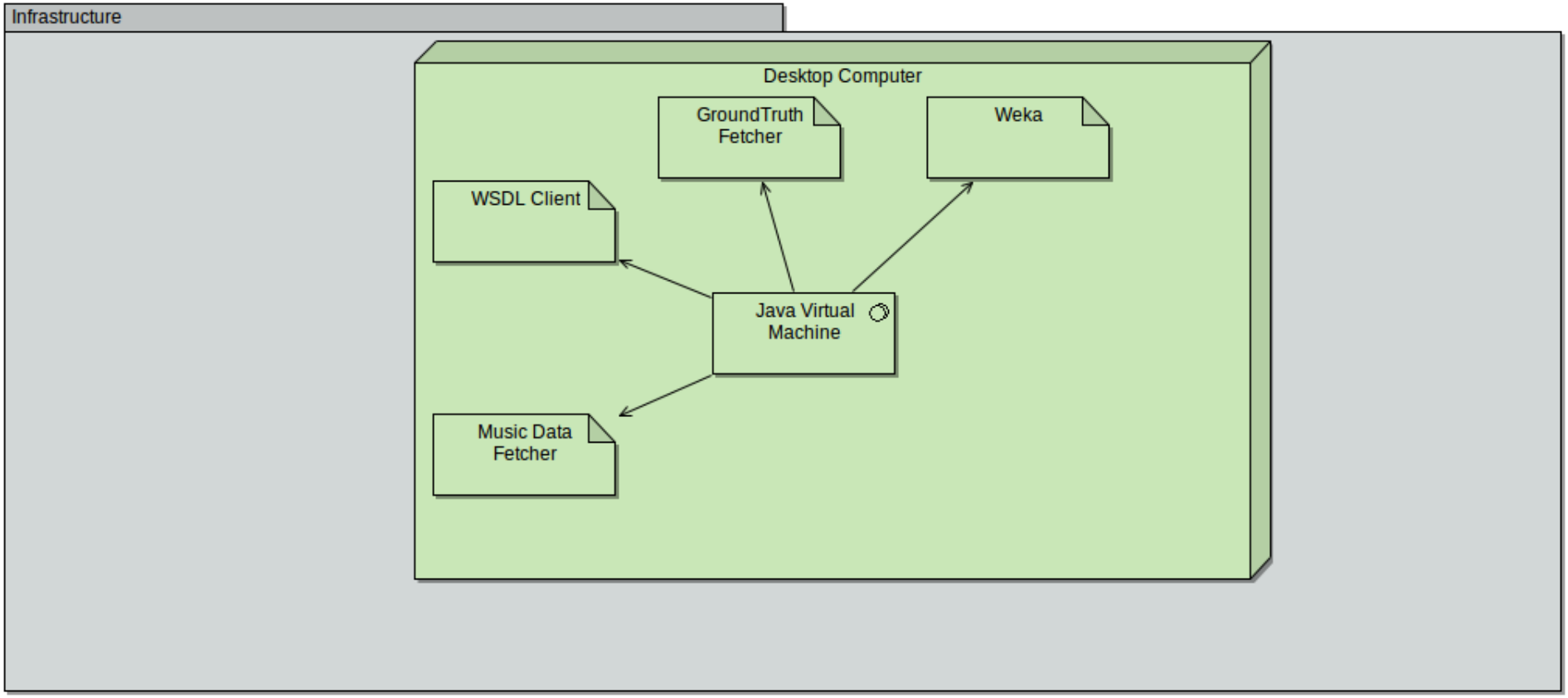
TIMELESS BUSINESS




External Roles and Actors
GroundTruth Service Operator
MusicData Service Operator
FeatureExtra... Service Operator
External Business Services
GroundTruth Service Usage Terms
MusicData Service Usage Terms
Audio FeatureExtra... Usage Terms
GroundTruth
GroundTruth
Music Data
Music
FeatureExtra...
Features
Business Process and Internal Actors/Roles
Music
Features
Annotated Features
Accuracy Result
Preservation
Repeatability
Experiment
Start Experiment
Get Music Data
Extract Featur...
Get Groundtruth
Combine GroundThr... w/ Features
Classify
Present Results
Researcher
Scientific Excellence
GroundTruth

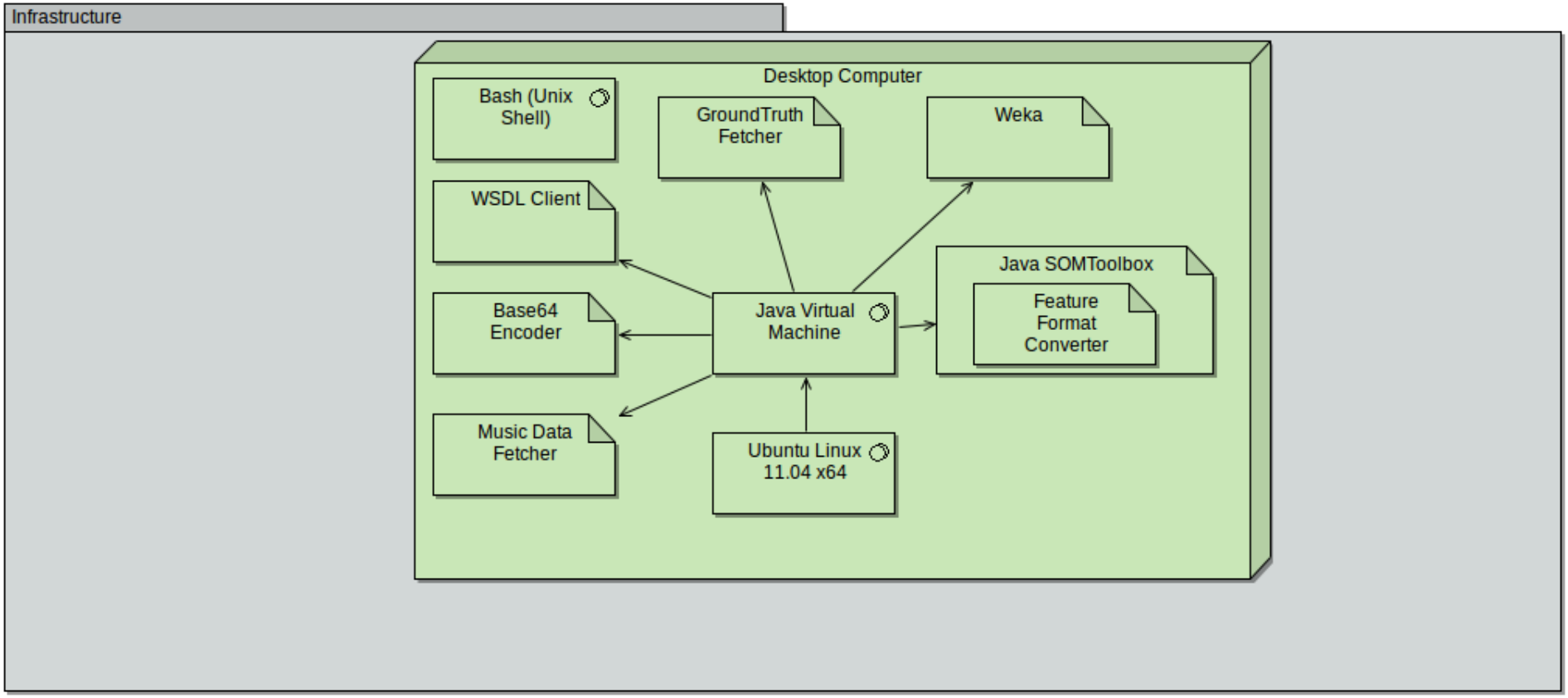
Infrastr. Layer: Basic SW

TIMELESS BUSINESS



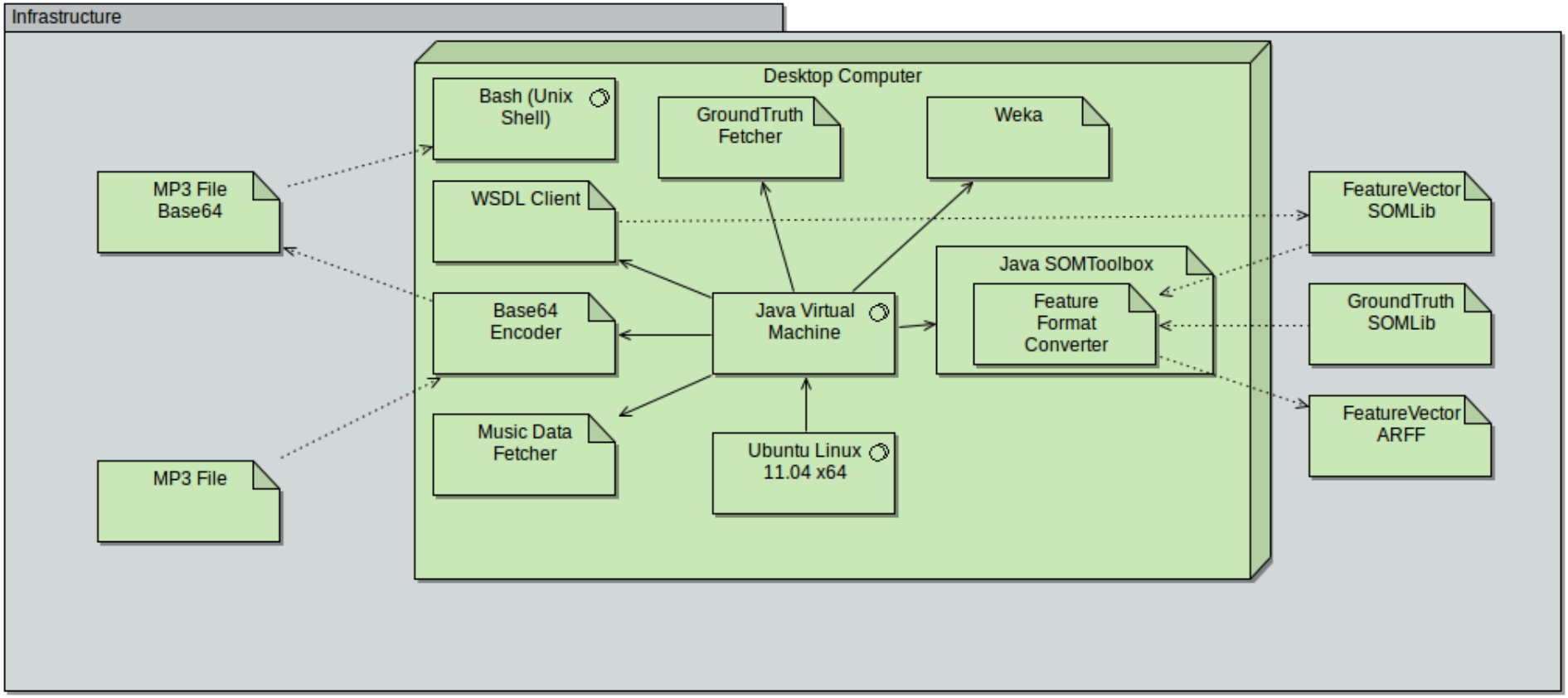
Infrastr. Layer: Complete SW

TIMELESS BUSINESS



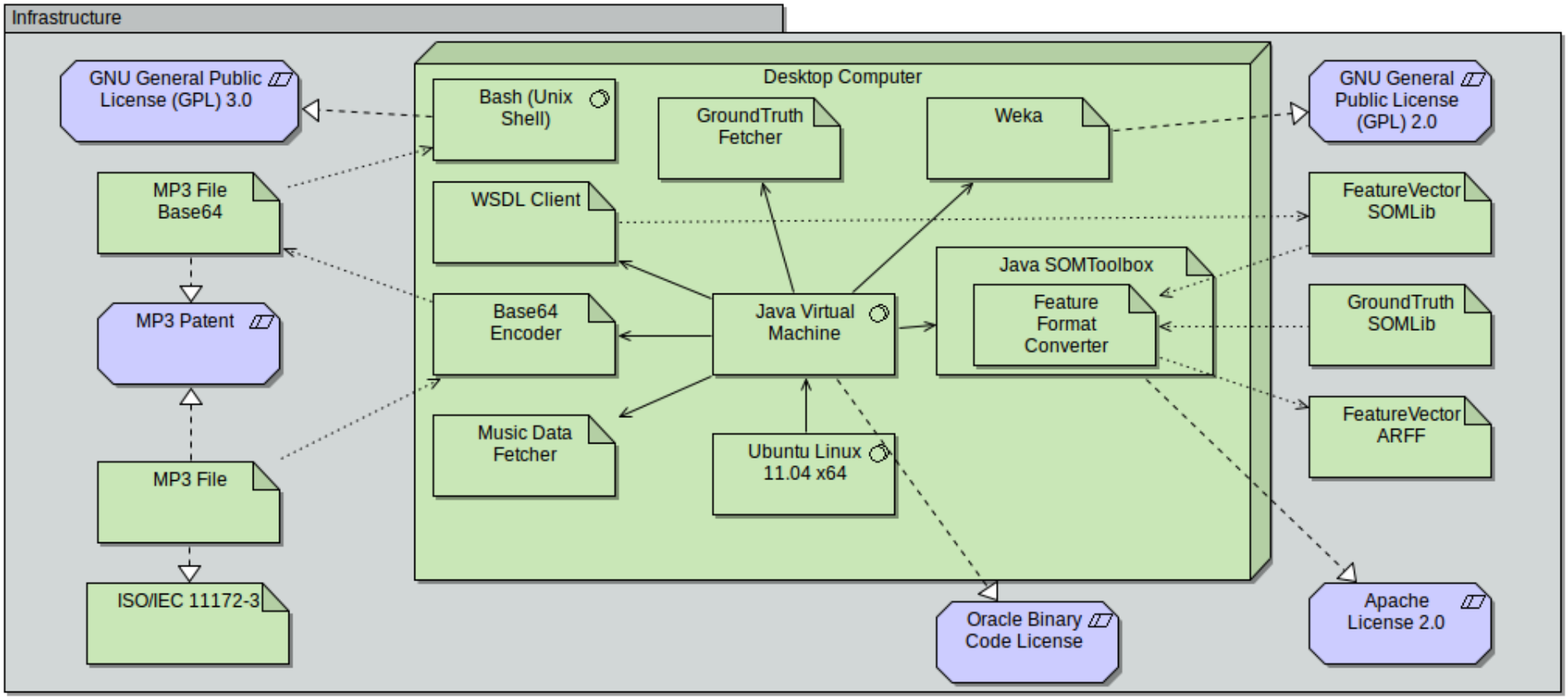
Infrastr. Layer: + Data Objects

TIMELESS BUSINESS



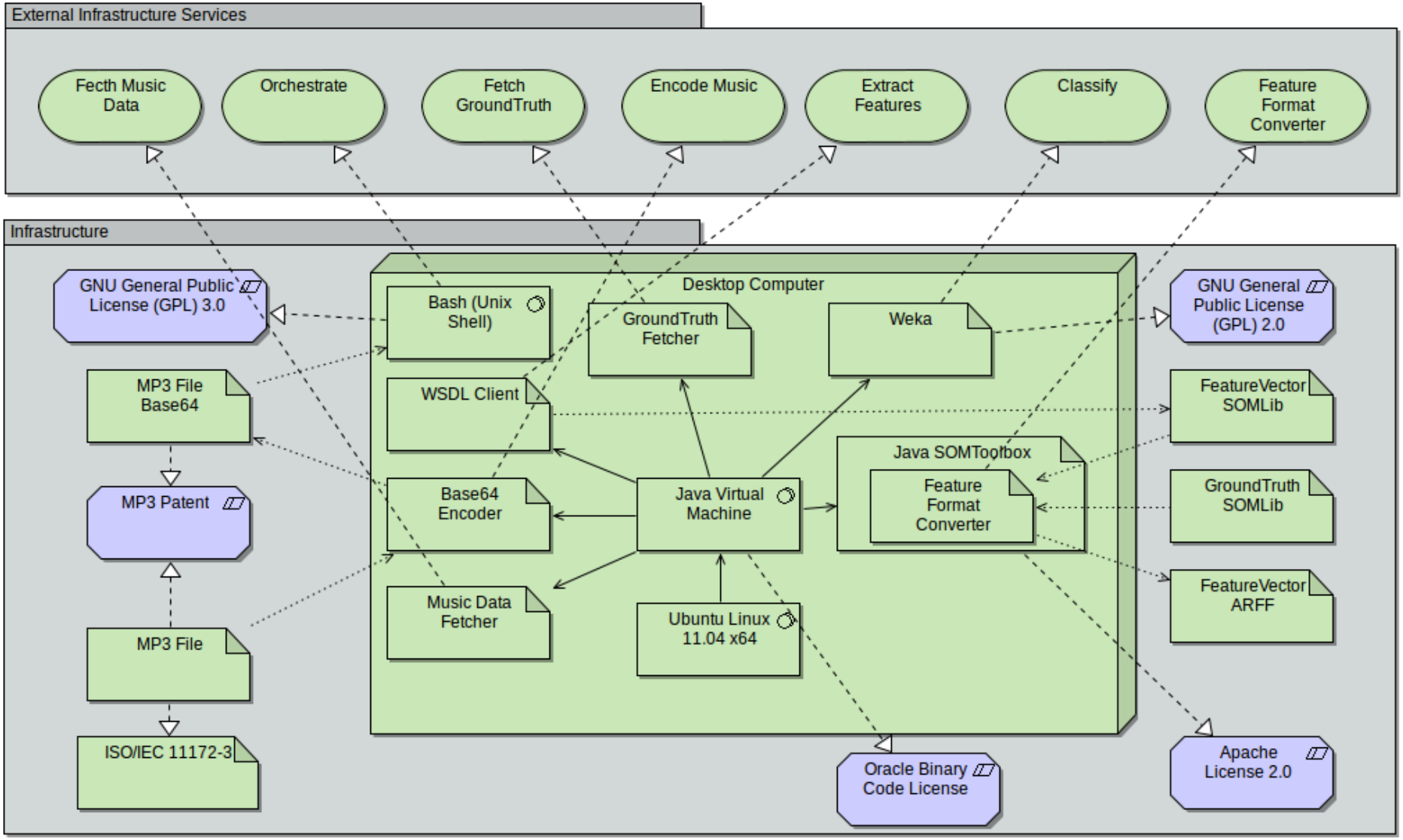
Infrastr. Layer: + Licenses etc

TIMELESS BUSINESS



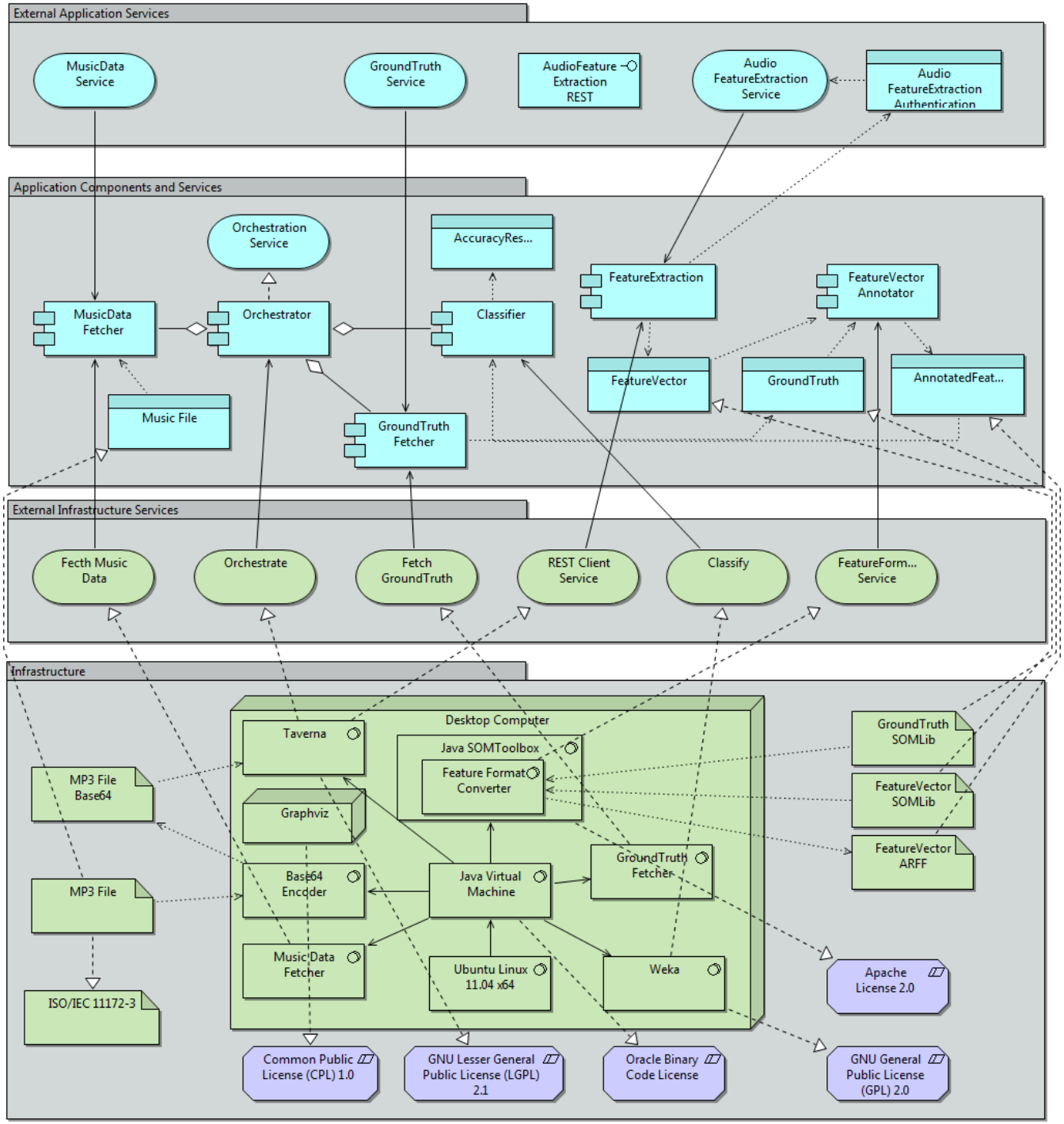
Infrastr. Layer: + Services

TIMELESS BUSINESS



Musi

TIMELESS BU

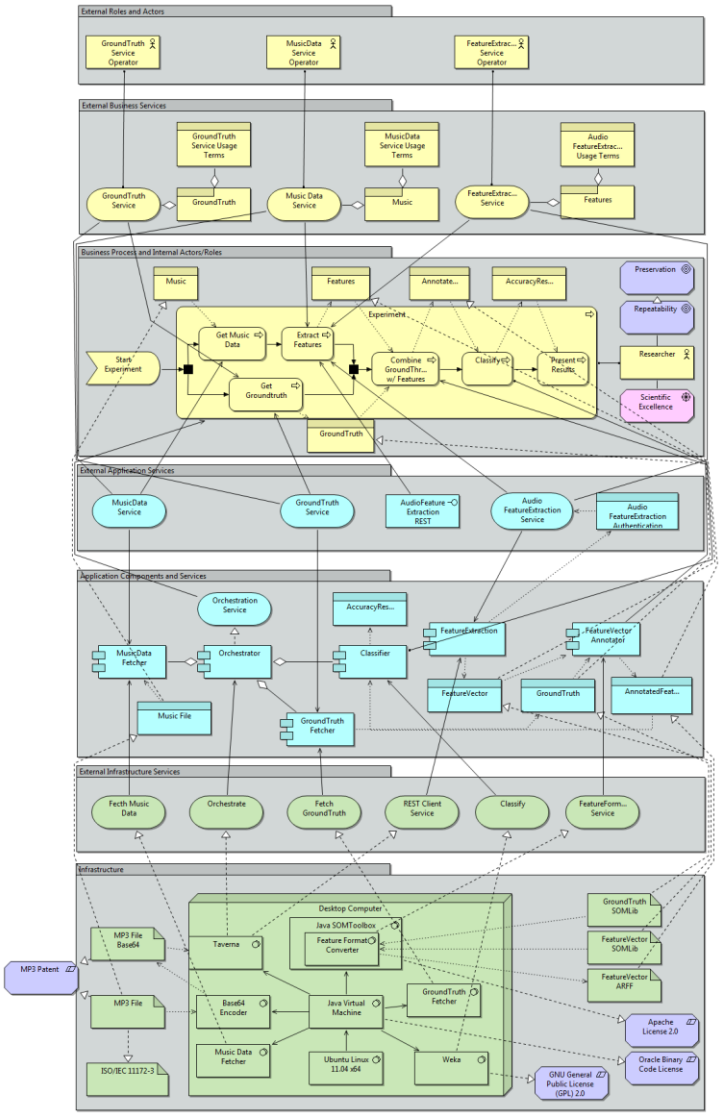


Context Model and Views

TIMELESS BUSINESS

- Workflow / process
- Business objects
- Motivation
- Software services
- External services

- *Files and formats*
- *Licences and patents*



Outline

TIMELESS BUSINESS

- Introduction and Motivation
- Example use case – Scientific Experiment
- Modelling process context
  - Modelling the example use case
- Extended process context model
  - Modelling the example use case
- Automatic context extraction & reasoning

Context Model DSOs

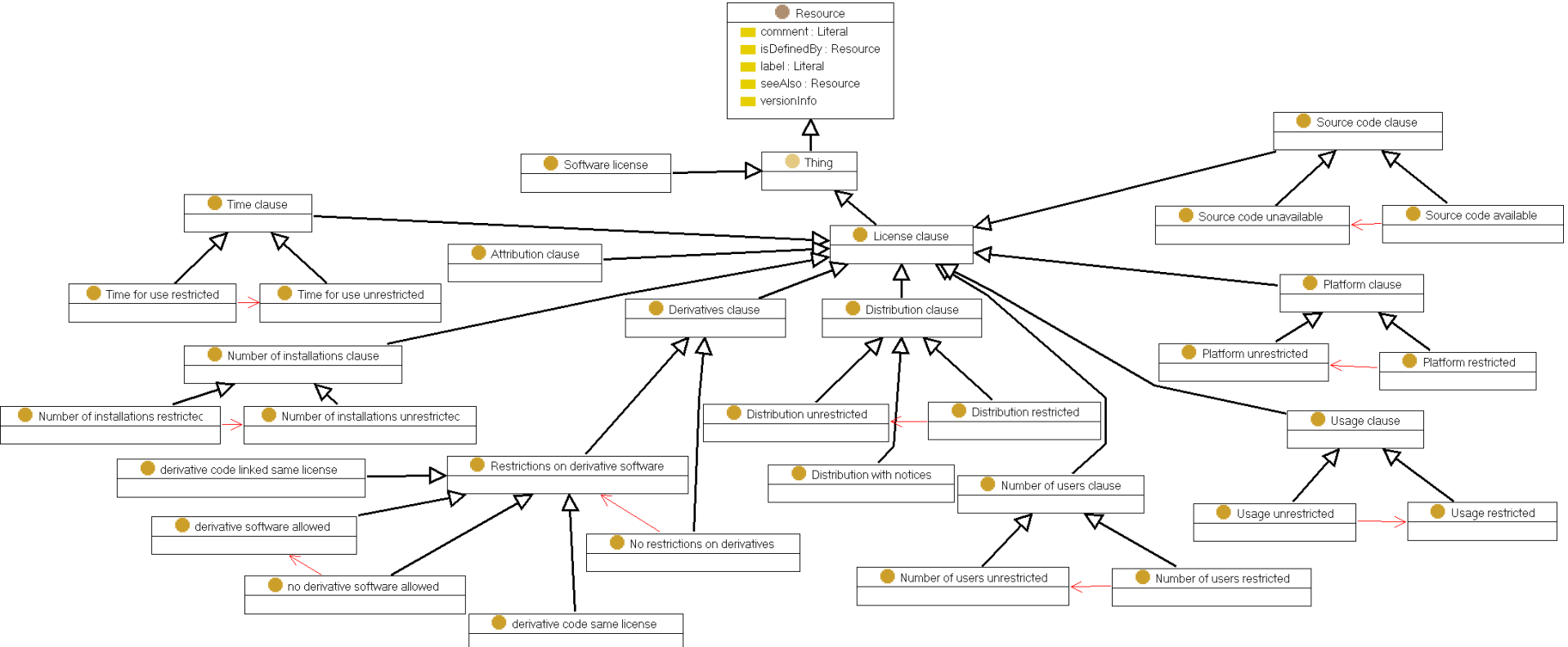
- Licensing: mapping to third-party ontology (The Software Ontology)
- Patents: mapping to third-party ontology (PATexpert project)
- Digital Preseravtion metadata: mapping to third-party ontology (PREMIS)
- Software applications, software services, dependencies (CUDF)

- Other: Sensors

License DSO

TIMELESS BUSINESS

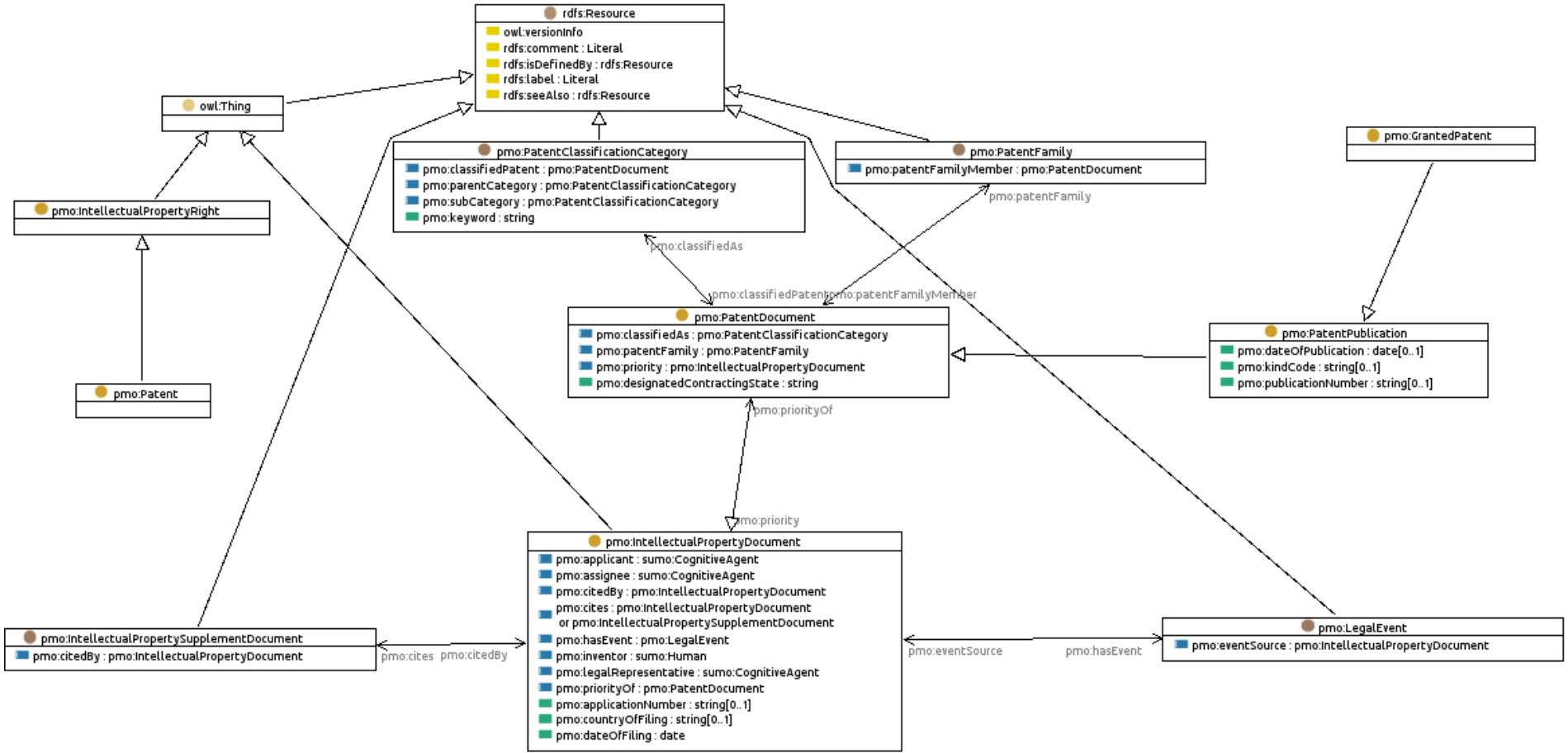
- License clauses, aggregated into licenses
- Mapping to DIO: licenses are *constraints*



Patent DSO

TIMELESS BUSINESS

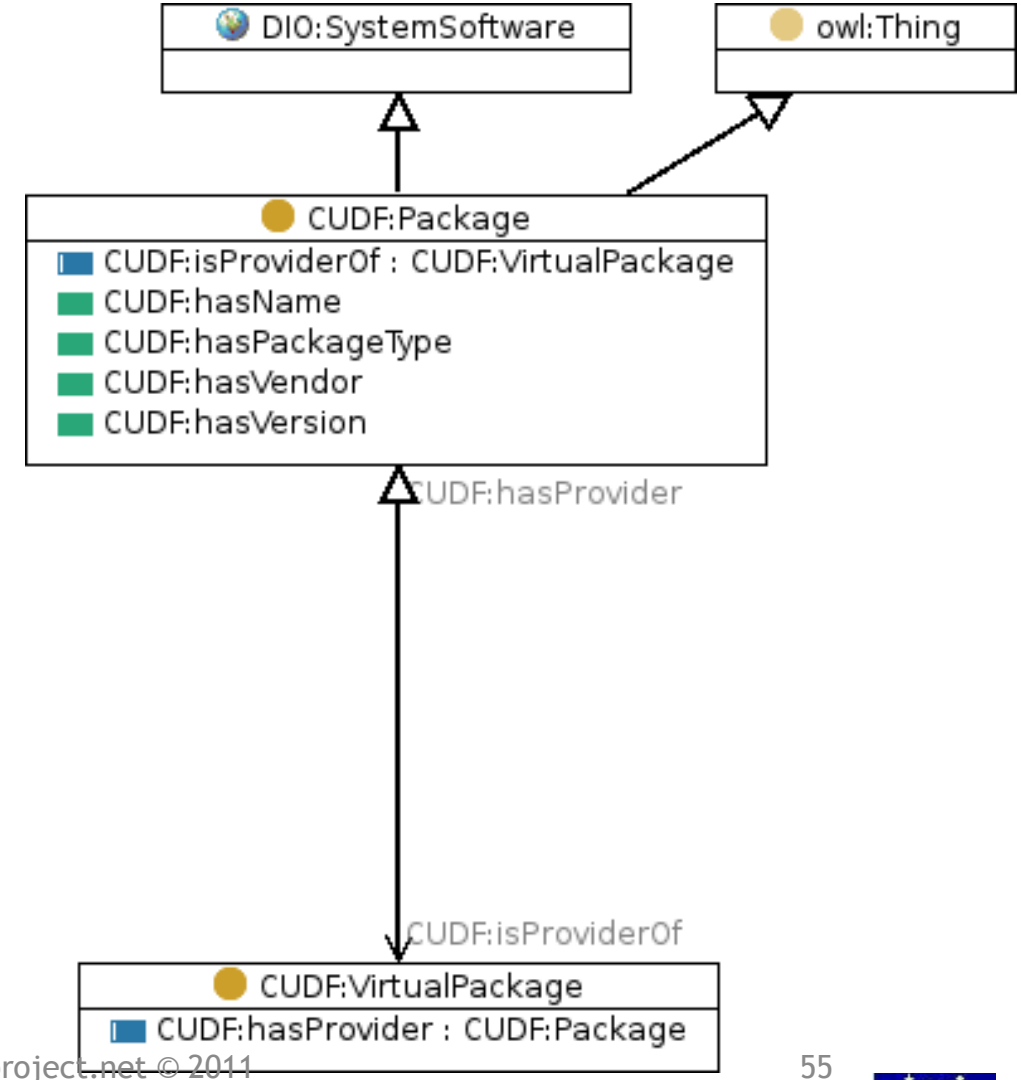
- Focus on patents & IPRs granted
- Mapping to DIO: GrantedPatent is a *constraint*



CUDF DSO

TIMELESS BUSINESS

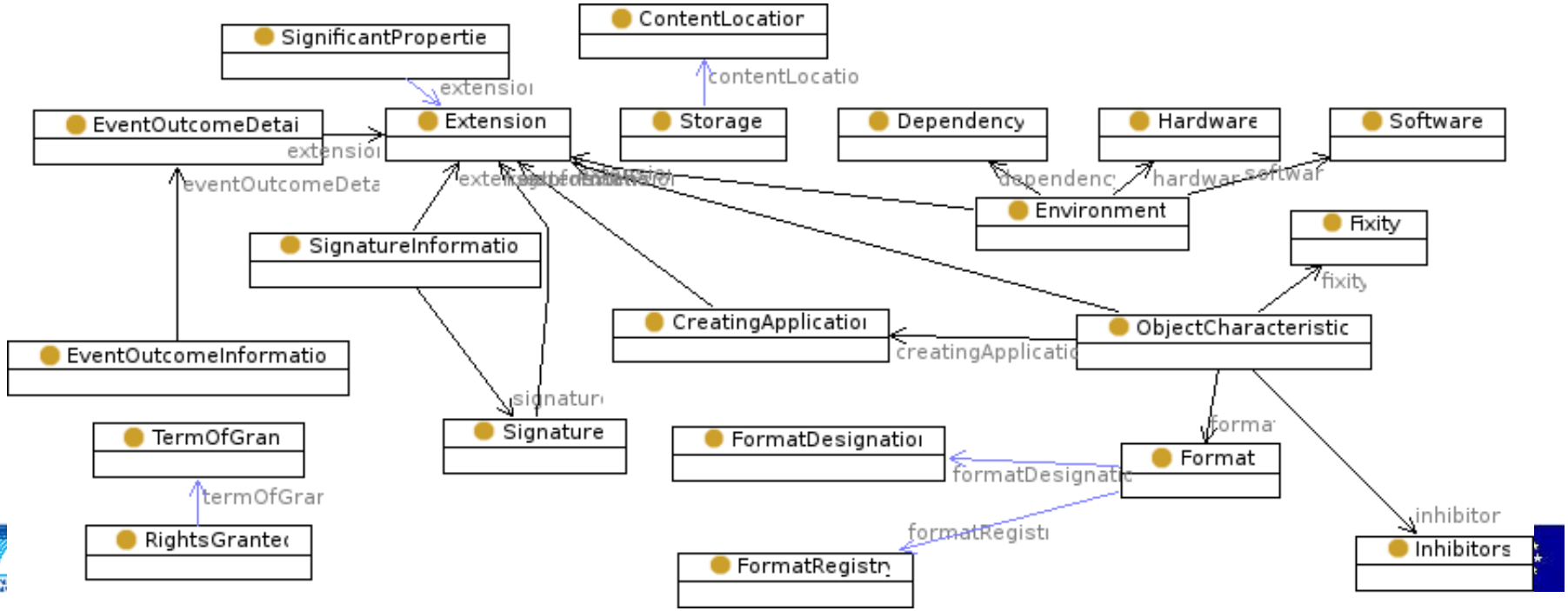
- Focus on SW relations (depends, conflicts, ...)
  - Package based systems (Linux)
  - DLL dependencies (Windows)
- Mapping to DIO: Package is a *SystemSoftware*



PREMIS DSO

TIMELESS BUSINESS

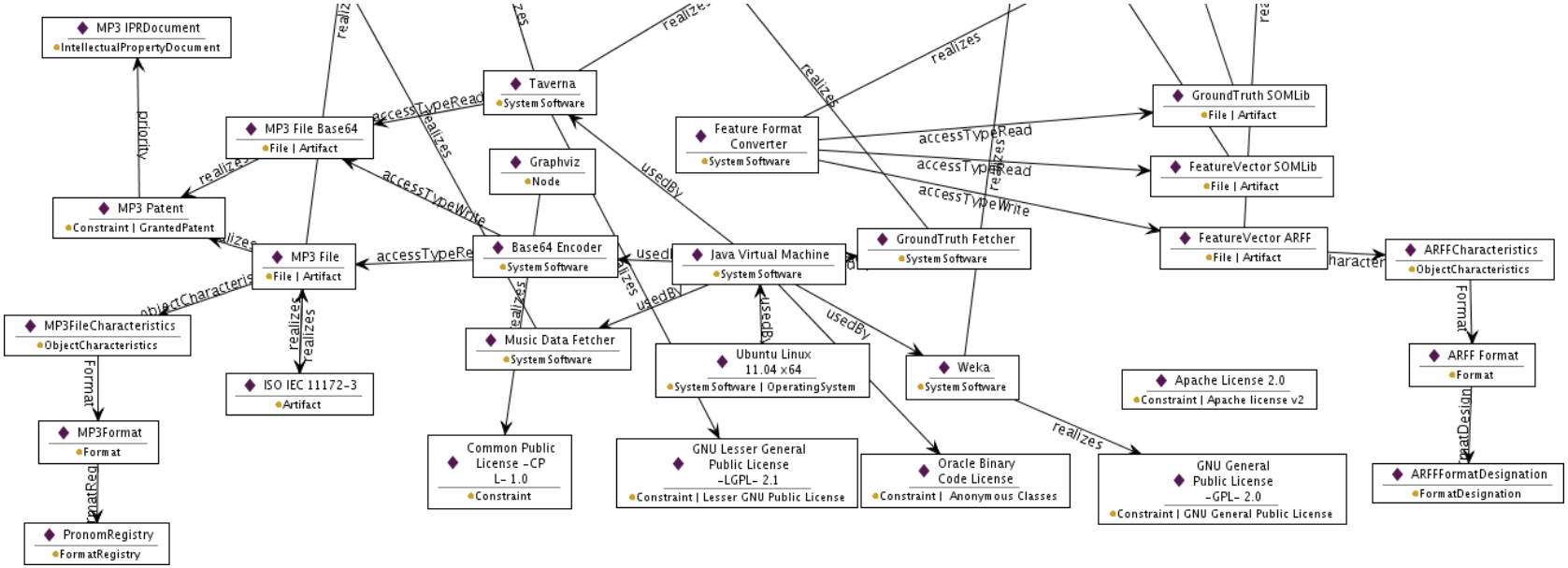
- Mapping to DIO:
  - Files are *Artifcats* (allow specification of format etc.)
  - Hardware are *Devices*, Storage are *Nodes*
  - License, Copyright, ContentLocation, IntellectualEntity,



Use Case: Infrastructure

TIMELESS BUSINESS

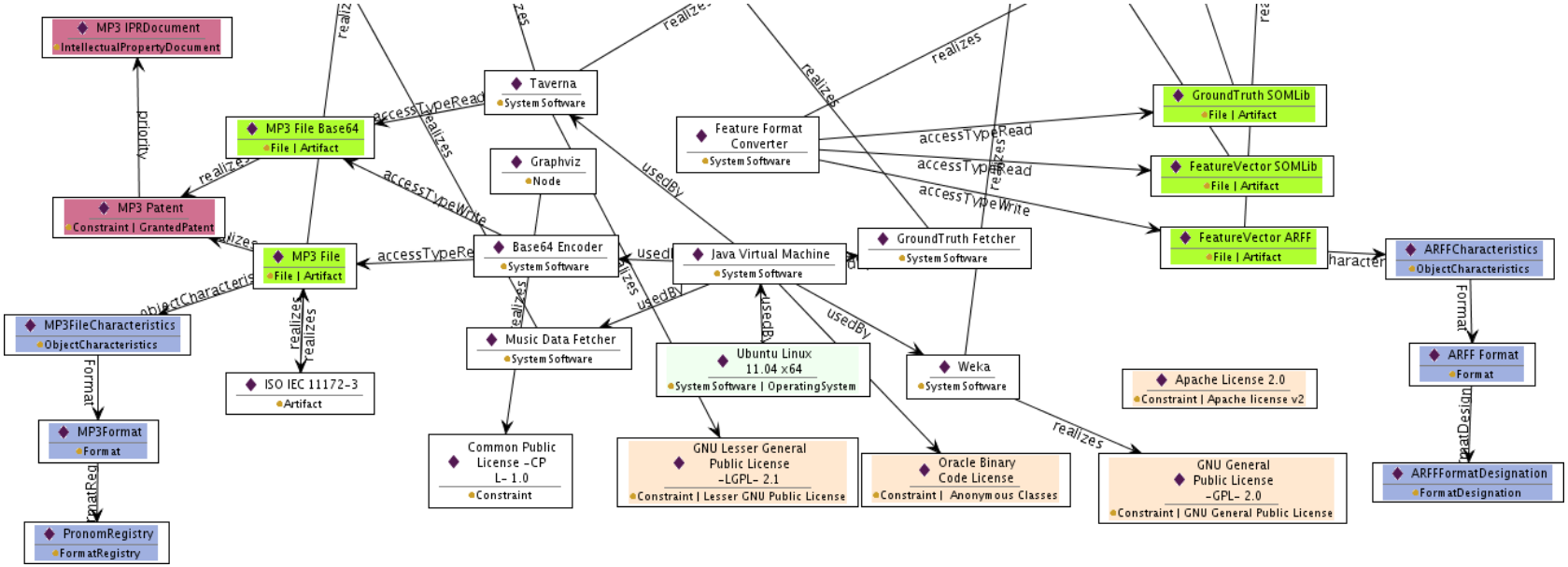
- Same example as before, Protege Visualisation



Use Case: Infrastructure

TIMELESS BUSINESS

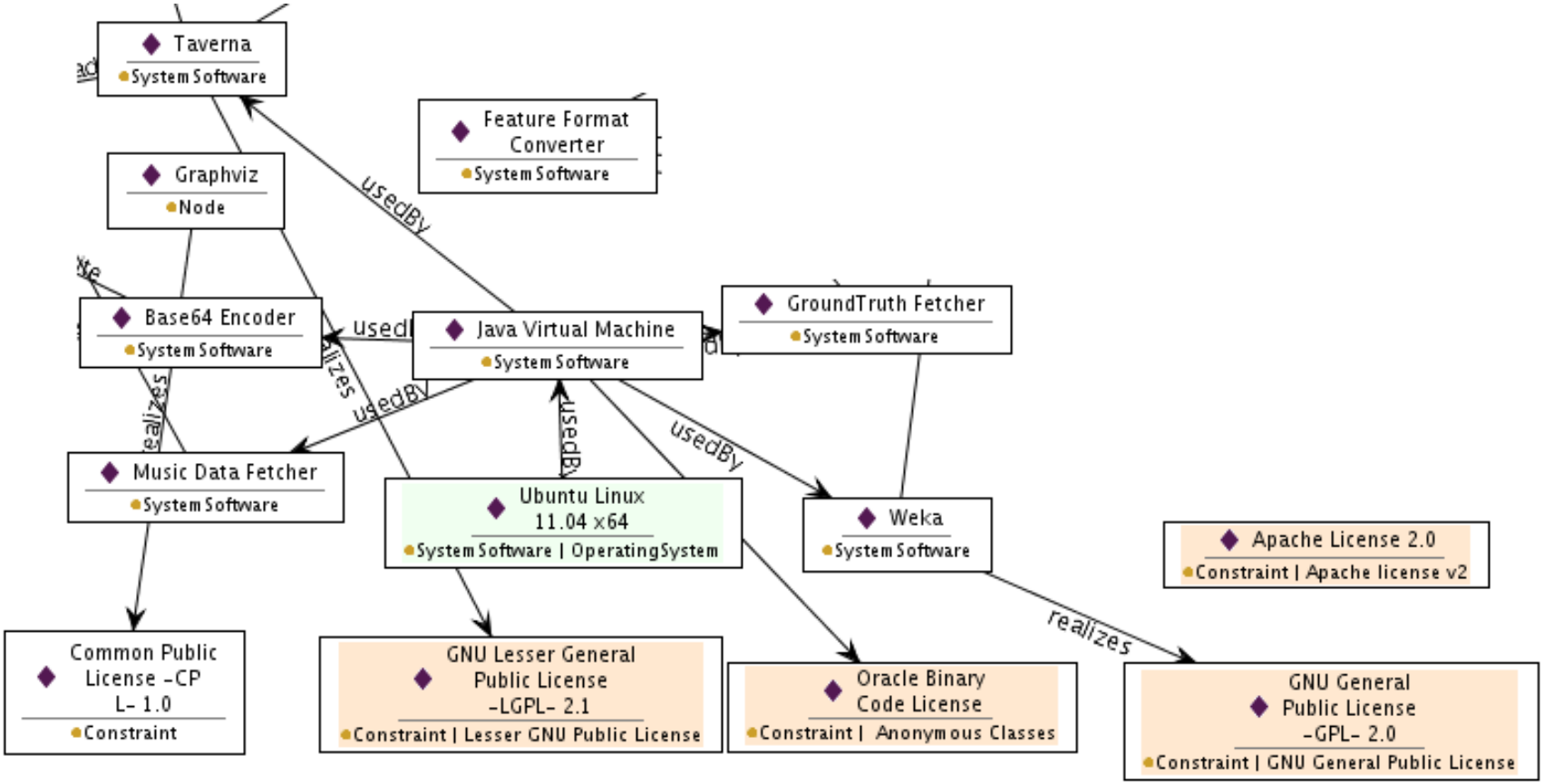
- Elements also marked up via DSO concepts



Music Classifciation: License & SW

TIMELESS BUSINESS

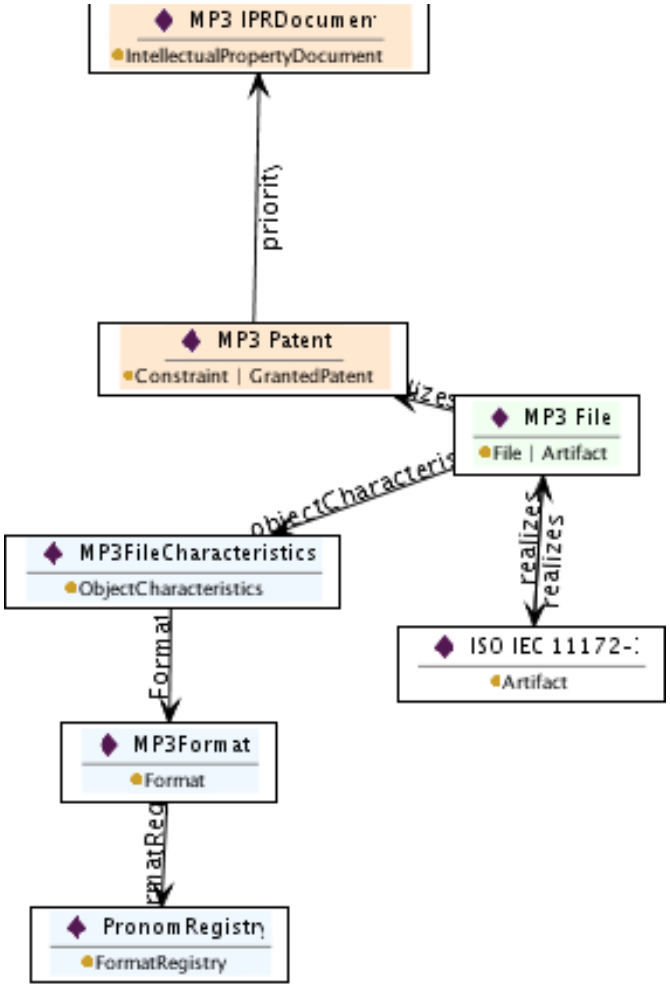
- More descriptive information for licenses

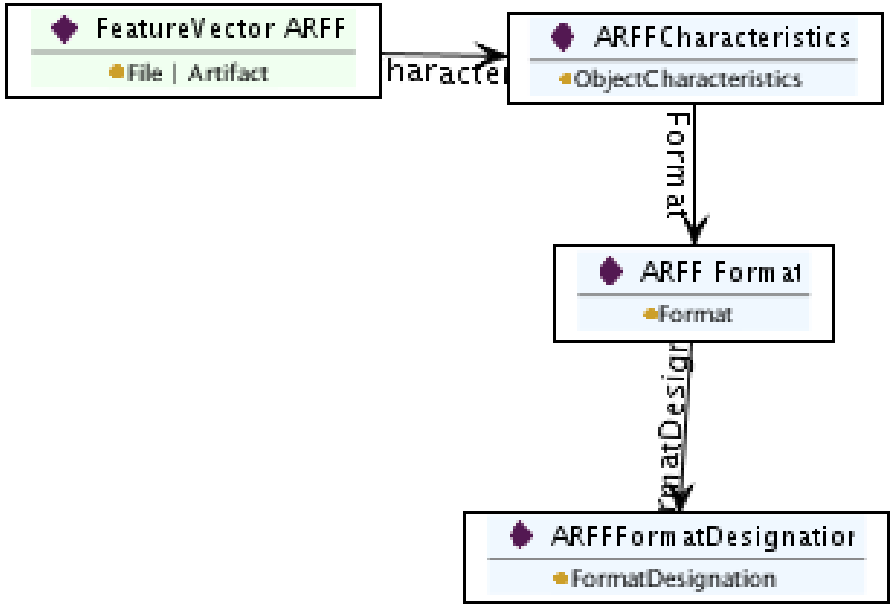


Music Classifciation: Formats & Patents

TIMELESS BUSINESS

- Details on formats & patents

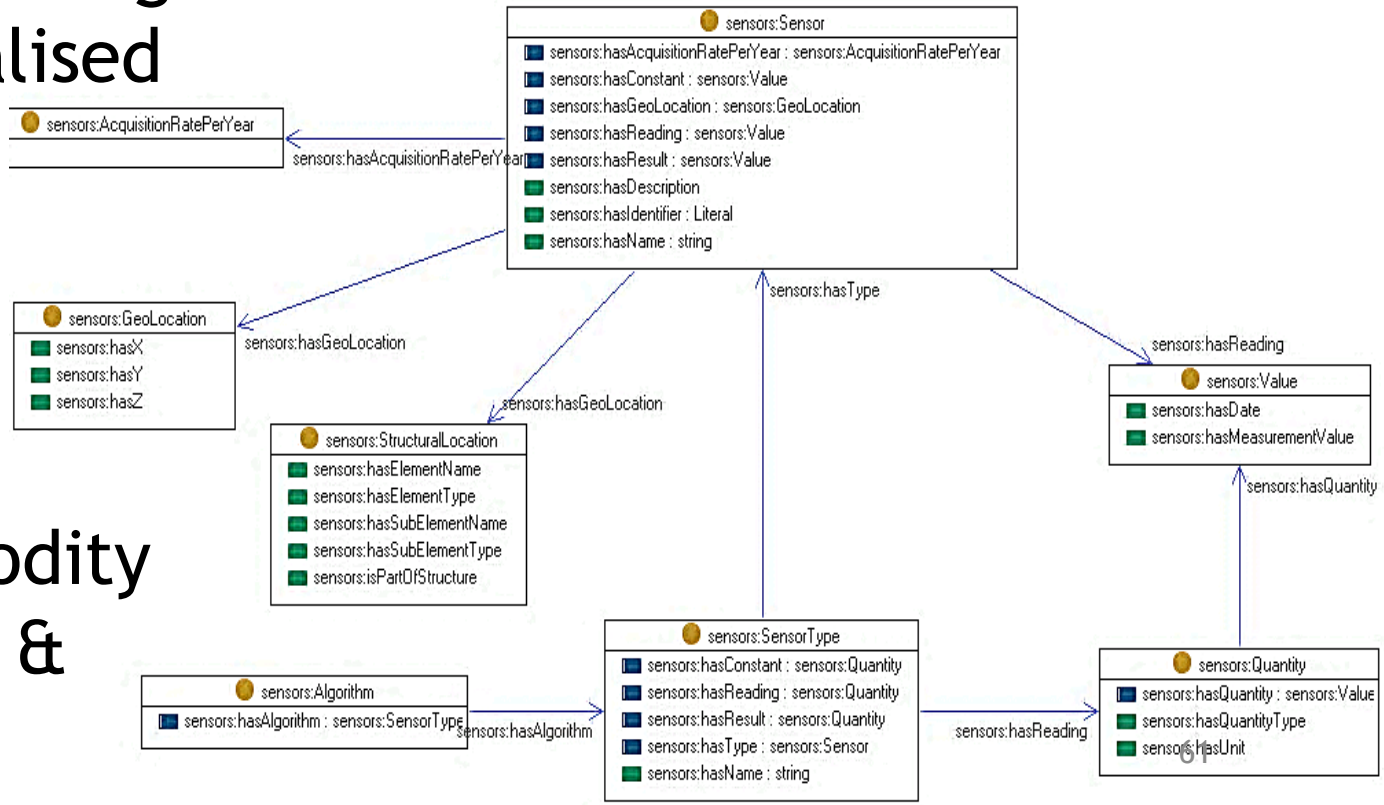




Other DSOs: Sensors

TIMELESS BUSINESS

- Use case: public authority monitoring dams & other civil engineerings structures
- Data gathering via specialised sensors
- Analysis on commodity hardware & software



Outline

TIMELESS BUSINESS

- Introduction and Motivation
- Example use case – Scientific Experiment
- Modelling process context
  - Modelling the example use case
- Extended process context model
  - Modelling the example use case
- Automatic context extraction & reasoning

Automatic Extraction of Context

- Manual approach limited
  - Time-consuming, expensive
  - Easy to miss out on details

- Some aspects can be extracted automatically
  - Especially on the technology layer
  - Less on the business layer
  - ➔ Technology layer is likely more compley anyhow
  - ➔ Business layer more likely to be documented already (BPMN diagrams, etc)

Automatic Extraction of Context

- Currently under development
  - to be released next year

- Currently for
  - Software applications (CUDF)
  - Licenses
  - File Formats
  - Hardware

Application Setup Extractor

- For package based systems: uses package manager to determine required and installed packages
- Allows to recreate the system from this information

- For Windows: DLL dependency crawler
  - Produces dependency graph for DLLs, including version information, down to system DLLs

License Extractor

- Mostly for open-source linceses
- Uses third-party tool license-check to determine license from included copyright statement
- Commercial alternatives: e.g. Blackduck

File Format Extractor

TIMELESS BUSINESS

- When process input & output data is captured
- Uses third-party characterisation tools to determine formats
- For unknown formats: stubs are generated, to be completed by expert

Hardware extractor

- For Linux & Windows
- Gathers information on
  - Processor, memory
  - Hard-disk
  - Network cards & configuration
  - Specialised hardware, e.g. GPUs

Outline

TIMELESS BUSINESS

- Introduction and Motivation
- Example use case – Scientific Experiment
- Modelling process context
  - Modelling the example use case
- Extended process context model
  - Modelling the example use case
- Automatic context extraction & reasoning

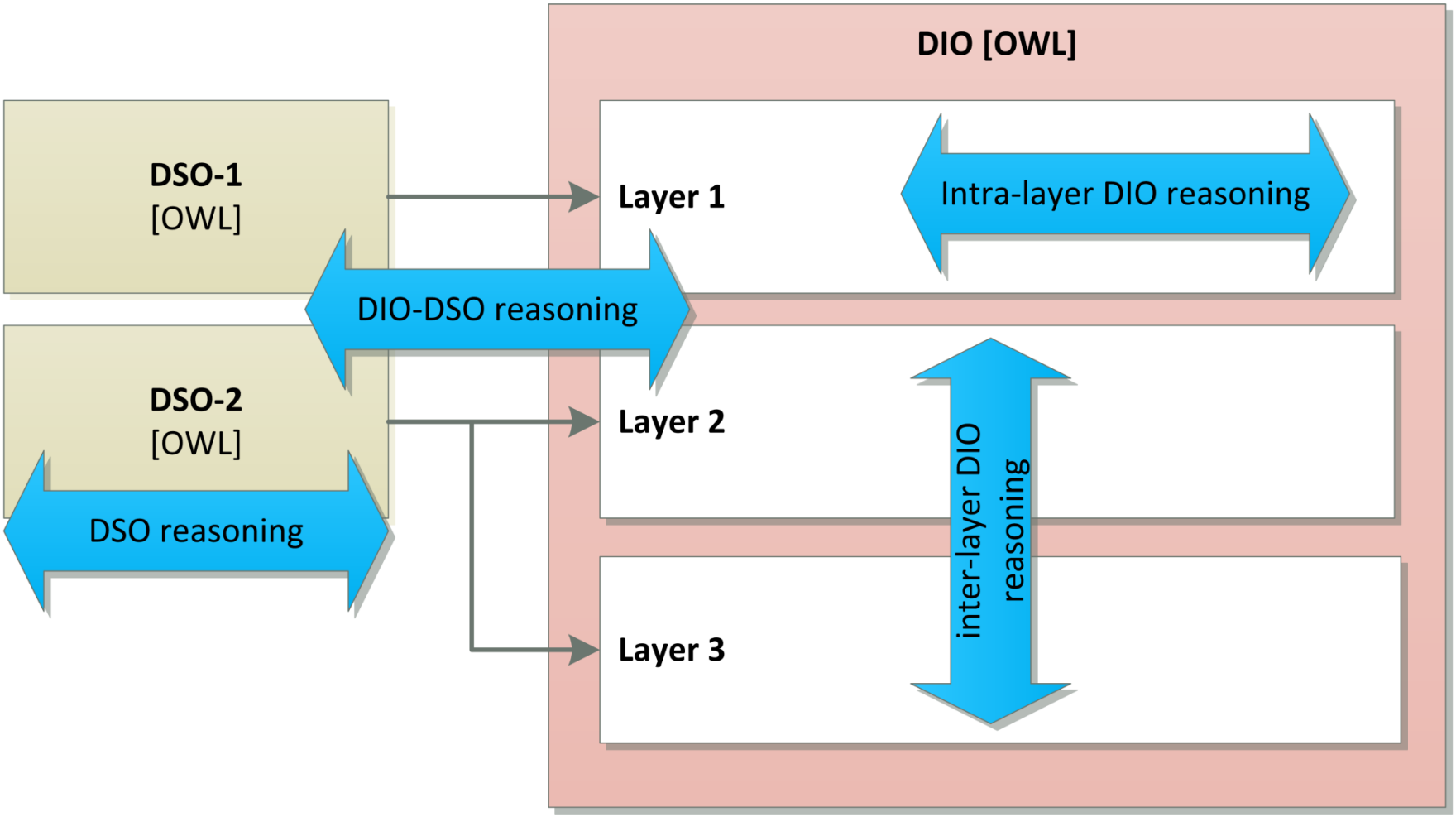
Reasoning

TIMELESS BUSINESS

- Identify components from the process that are of particular interest for preservation
  - E.g. to determine the effect of migration of a software or format
- Mostly used in the preservation planning phase
  - Determine what actions we can apply to the process
  - Determine impacts of these actions
- Also used when comparing redeployed form
- In the form of a Description Logic or SPARQL query (similar to SQL)

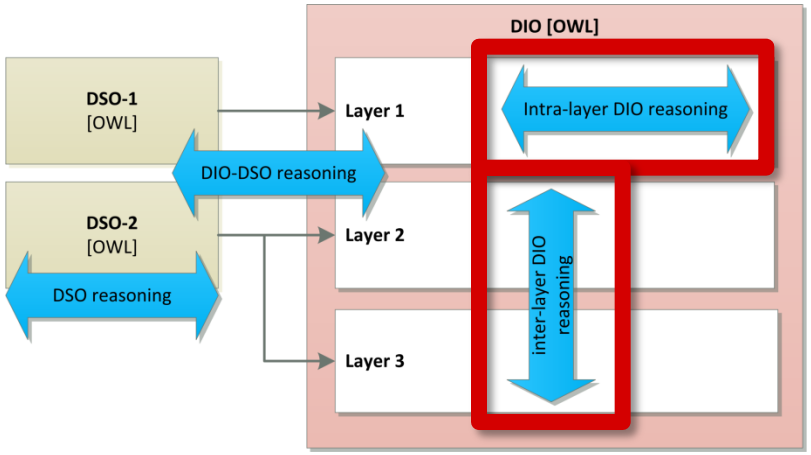
Reasoning

TIMELESS BUSINESS



Music Classification Reasoning

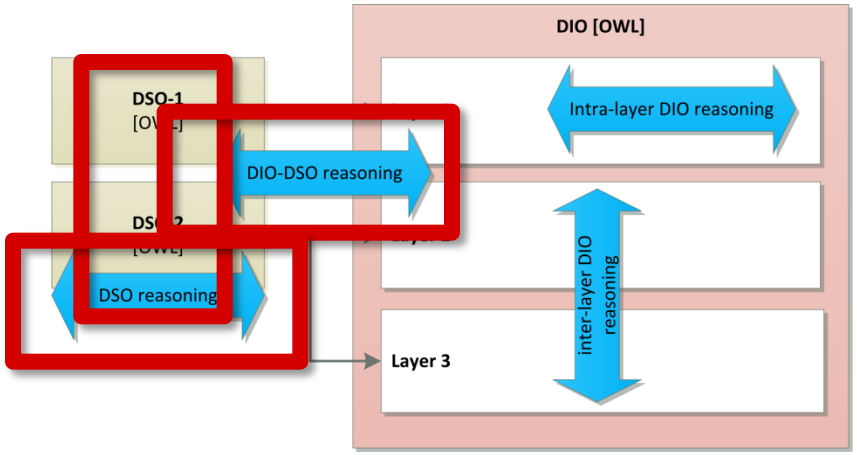
TIMELESS BUSINESS



- What are the business processes of Music Classification?
- What application components support the "Experiment" business process?
- What hardware nodes support the "Classifier" software application?

Music Classification Reasoning

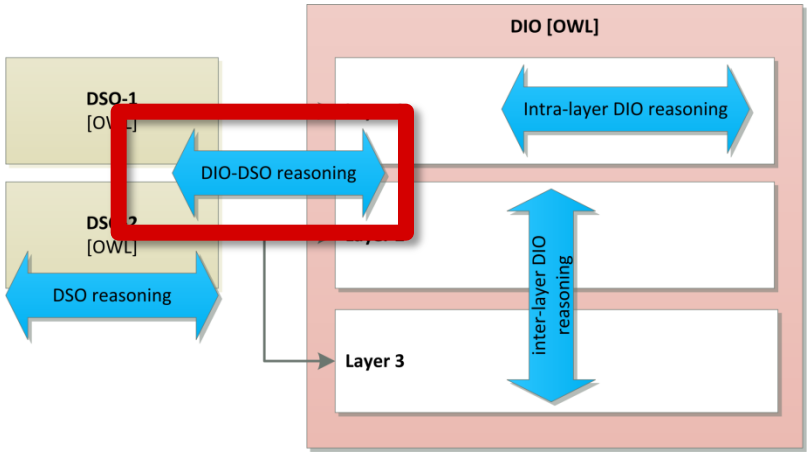
TIMELESS BUSINESS



- What are the patents applicable to Music Classification?
- What software applications require a software license?
- What are the relationships between patents and software licences?

Music Classification Reasoning

TIMELESS BUSINESS



- Which file formats are used in the process?
- Which software is used to read/write a certain format?
- Which are the external services

Sensors Reasoning

TIMELESS BUSINESS

Query (class expression)

Thing **and** hasLayer **some** TechnologyLayer **and** hasAspect **some** BehavioralAspect
or hasAspect **some** ActiveStructuralAspect **and** dependsDown **value** Acquisition_of_readings

Execute | Add to ontology

Query results

Instances (19)

- IIS
- Red_Hat_Linux
- Oracle_Client
- WCF_Client
- PDT_Application
- Data_management
- Producer
- MCGateway
- Data_provider
- Data_Access_
- Windows_Server_2008
- Application_Server
- Database_Server
- GestBarragens
- External_Application
- WCF
- .NET_Framework
- DBMS_-Oracle_10.g-
- Web_Application

Technology infrastructure supporting acquisition

Query (class expression)

ApplicationComponent **and** dependsUp **some** (Sensor **and** hasSensorType **value** Drain)

Execute | Add to ontology

Query results

Sub classes (0)

Instances (11)

- gB-Messages
- gestBarragens
- gB-Support_System
- Structure_Management
- gB-Documental_System
- gB_-_Data_Access
- gB-Observations_System
- User_Management
- GB-_Uploader
- Permissions_Management
- gB-PDT

Which applications components rely on readings of sensor type drain

Query (class expression)

SensorType **and** hasReading **value** time

Execute | Add to ontology

Query results

Sub classes (0)

Instances (1)

- Drain

Sensors able to make temporal readings

TIMBUS

TIMELESS BUSINESS

Thanks for your attention!

Questions ?