**Contents**

Objectives: Develop Protocol (APIs) to facilitate Enterprise Application Integration (EAI) by means of Semantic technologies and Machine Learning. Ontology matching driven data, schema, behavior inference / aggregation / matching. Reasoning and learning over different consolidated backends alignments.

Distributed P2P (Blockchain) approach of data synchronization between peers for ease of deployment patterns election and datasources integration (APIs, microservices, etc.).  
  
Data alignment:  
  
Determine if two instances (example: records) of two different backends or services refer to the same entity (Customers : John D. / Employees : John Doe).  
  
Schema alignment:  
  
Determine, for example, meaning and equivalences between diverse (aggregated / composite) schemas (equivalent classes, equivalent attributes, equivalent roles).  
  
Behavior alignment:  
  
Determine meaning and equivalences between (aggregated / composite) behavior contexts and behavior contexts invocations / interactions (Appointment / Interview, anAppointment / anInterview. Behavior flows aggregated from backends / services learning).  
  
Layered models semantic infrastructure for integration of heterogeneous backends (meta models).

Alignments Augmentations:  
  
Activation: type inference : classification (determine class / metaclass / roles for entity attributes and values).  
  
Activation infer attributes / relations : clustering (from multiple occurrences of same entity in diverse data sources).  
  
Aggregation: infer roles in contexts: regression (Person class in Employment interaction : Developer role).  
  
Integration of addressable resources. Reactive I/O (sync back ends). Content type driven semantic augmentation / annotations.  
  
Integrated view. Navigate contexts, data, interactions. APIs. Dimensional views annotations (analysis / mining).  
  
Augmentation of distributed resources. Annotations (Semantic / ML). API for resource / schema / interactions exploration / protocol for message based API "dialogs" execution. HAL (Hypertext Application Language), OData (REST) like interfaces.  
  
Example: Google Drive / Google Knowledge Graph APIs Augmented with ML / Semantic intelligence tailored for specific domains / application kinds.

**Features**

Graph encoding of data / schema / behavior. Dimensional / Grammar annotations. MetaGraph: augmentation / transforms (Messages). Features.

Parallel distributed graphs models augmentation / transforms synchronization (Messages). Event sourcing (distributed inferences). P2P / DIDs.

Augmentation. Ontology matching. Hypermedia augmentation protocol. Browser / Client APIs.

URIs API for annotating network retrieveable resources metadata. Content type / model driven augmentations / activations (models features / outputs). Subject attributes / values. Occurrences contexts / roles. Paths, pointers, locators. Example: annotate document URIs (parts, sections, mentions), annotate images URI (whole image description, coords: classes, individuals), annotate DB, table, row, column, value URIs, annotate / describe service / APIs URIs. Hypermedia protocol composable with other (described / annotated) APIs / resources. Example: Drive APIs.

**RDF triples, quads introduction**

TBD.

**Models: Quads, Contexts, Occurrences, Attributes, Values.**

TBD.

**URIs, Resource, Statement, Kind APIs**

TBD.

**Model Layers**

See Messages / Augmentation.

**Data Model**

Data Model layers population / augmentation.

**Schema Model (Grammars)**

Schema Model layers population / augmentation.

**Behavior Model (Dimensional)**

Dimensional Model layers population / augmentation. Purpose modelling. Dimensional Concepts.

**MetaGraph Model (models aggregations)**

See Message Resolution.

**Datasources / Backends / Services (URIs)**

TBD.

**Addressing. IDs. Encodings**

TBD.

**Dataflow (reactive models)**

TBD.

**Messages: Transforms. Graph Execution Semantics**

Message encoding semantics resolve transform execution resource set declaratively from MetaGraph / Models.

**Augmentation (via Messages)**

Activation (Statement / Entities : data).

Alignment (Kinds / Classes : context / schema).

Aggregation (Flows / Behaviors : interaction).

Messages describes declaratively augmentation steps materializing models contexts / hierarchy layers.

**Protocol (API): dialogs (distributed resource augmentation / sync)**

Message resolution (contexts).

Reactive. Interaction / session contexts.

**Protocol (API): resource activation (hypermedia application browser)**

Reactive. Interaction / session contexts.

**Ontology matching**

TBD.

**Data / Reference Model (APIs, Functional Semantics)**

TBD.

**Platform: implementation**

TBD.