**Contents**

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**

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

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

**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. Concepts.

**MetaGraph Model (models aggregations)**

See Message Resolution.

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

**Addressing. IDs. Encodings**

**Dataflow (reactive models)**

**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).

Aggregation (Flow / Behavior : interaction).

Messages describes declaratively augmentation steps materializing models context hierarchy layers.

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

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

**Ontology matching**

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

**Platform: implementation**