**Contents / Features (Mision / Vision). Distributed consistent Knowledge Applications.**

**RDF / OWL, Graphs, Triples, Quads introduction.**

**URIs, Resource, Statement, Kind APIs. RDF Backend. URIs Services.**

Context Kind Signatures.

**Functional Implementation APIs.**

Monad: Resource<URI>.

Resource layers hierarchy API.

**URIs APIs: Datasources / Backends / Services.**

**Model: Quads CSPOs: Object Graph Representation as RDF Quads.**

(Context, Occurrence, Attribute, Value);  
(Context, Sign, Concept, Object);

Value as Occurrence of Attribute in context.

Subjects: attributes / values, contexts / roles.

Instance, occurrence, class, metaclass.

**Model Layers:**

Layered data, schema, behavior class / instance quads hierarchy. Model layers: URI quads:

Resource : Functional URI wrapper.

(Context : Resource, Occurrence : Resource, Attribute : Resource, Value : Resource);

(Statement, Occurrence, Attribute, Value);

(Entity, Statement, Attribute, Value);  
(Role, Entity, Statement, Attribute);  
(Class, Role, Entity, Statement);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

Graph Execution Semantics.

Ontology Matching. Upper ontologies. Primitives.

**Model I/O Dataflow:**

Layers (declaratively stated in Interaction Model):

Data input statements (Message).

Aggregate layers.

Align attributes.

Activate Kind.

Model: Reactive entity applying Message Augmentation resolving Resource Set Specification Message from inputs. Data Message (URIs layer), dataflow Message (Model / dialog).

Message Resolution Algorithm.

Data instance inputs (URIs events).

Model Message Augmentation resolution.

Interaction Model events / distributed / inference sourcing. Augmentations / CRUD: Interaction Model DIDs. URIs quad store / backend.

**Message:**

Resource Set Specification (Statement) matching Model which returns augmented Message response (Model I/O).

Augmentation declarative Model definitions.

Message Resolution Algorithm.

Protocol: Augmentation Message dialog I/O.

**Interaction Model:**

Source (upper) Model. Models hierarchies aligned with Interaction Model.

Interaction Model provides event sourcing, distributed inference / synchronization (distributed consolidation and alignments).

Interaction Model I/O : Message (from URIs or events) perform and materialize applying Augmentation from Interaction Model population.

Message declaratively states Model Specification through Message Augmentations.

Meta Resource(s): Resources / Messages reifying "patterns" on inputs (URI, Resource, Statement, Kind(s), Context, Occurrence, Attribute, Value, Layer Context classes, etc.). Declarative statement of Augmentation shapes applyied to input contexts.

Augmentations:

Data (Message, Aggregation);

Schema (Alignment, Activation);

Behavior (Transform, Specification);

Model Specification: Metacircular interpreter. Interaction Model reifies Source, Metagraph, Dimensional, Grammar Models via Augmentation Specification Message. Augmentation Message populates specified Model.

Functional (monadic) Message Resolution Algorithm. Encoding.

**Augmentation:**

Messages Resource Set Specifications for CRUD, Aggregation, Alignment, Activation over Model. (Interaction Model Specification) stated on Interaction Model or from Protocol Message.

Model I/O: Augmentation Message application over Model from backend (URIs) Message or from Model I/O (layers) Message. Returns Resource Set populated / materialized Message.

Augmentation: each Augmentation populates corresponding Models performing CRUD, aggregation, inference and classification from Interaction Model Specification.

**CRUD (Message) Augmentation:**

Augmentation: CRUD (Message).

Specification Model: Source.

Augmented Models (materialize, aggregate, align, activate).

**Aggregation Augmentation:**

Augmentation: Context Aggregation. Specification Model: Metagraph. Classification (aggregate quads contexts context / roles / class / identity).

**Alignment Augmentation:**

Augmentation: Data Alignment. Specification Model: Dimensional. Clustering (inference of links / attributes).

**Activation Augmentation:**

Augmentation: Interaction Activation. Specification Model: Grammar. Regression (classify roles in contexts: Kind).

**Models:**

Models hierarchies aligned with Interaction Model. Source, Metagraph, Dimensional, Grammar.

**Interaction Model Specification.**

Message / Aggregation (data)

Alignment / Activation (schema)

Transform / Specification (behavior).

Align to: URIs, Resource, Statement, Kind, Context Kind, Context, Occurrence, Attribute, Value.

(Context : Message, Occurrence : Message, Attribute : Message, Value : Message) : Message;

Resource : Functional URI wrapper.

(Context : Resource, Occurrence : Resource, Attribute : Resource, Value : Resource);

(Statement, Occurrence, Attribute, Value);

(Entity, Statement, Attribute, Value);  
(Role, Entity, Statement, Attribute);  
(Class, Role, Entity, Statement);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

Statement Aggregation: Statement instance Context for each distinct CSPO URI on inputs aggregates same URI Occurrence as Subject with corresponding Attribute (output Predicate) / Value (output Object). According CSPO input as Occurrence, corresponding Attributes / Values are chosen.

Resource : Functional URI wrapper.

(Context : Resource, Occurrence : Resource, Attribute : Resource, Value : Resource);

(Statement, Occurrence, Attribute, Value);

Aggregation layer: for each previous layer Message, layers: (Aggregation Instance, previous Message Context as Subject, previous Message S/P as Attribute / Value). Previous layer: Aggregation until end of source Messages layers (6 Aggregation statements consuming previous CSPOs. Renders to Aggregation instance contexts of Aggregation class).

Alignment layer: Context / Occurrence / Attribute / Value. Renders augmented Attribute / Value Context / Occurrence.

Activation layer: for each layer Message, Activation (Kind instances) are for each Activation class taking one of Message CSPO as Kind Subject and their corresponding CSPOs as Attribute / Value. Kind classes for each Aggregation layer. Context Kind: composite Subject / Predicate Kinds as Attribute / Value.

Layers dataflow: hierarchical Message inputs / outputs.

**Source Model Specification.**

Resource : Functional URI wrapper.

(Context : Resource, Occurrence : Resource, Attribute : Resource, Value : Resource);

(Statement, Occurrence, Attribute, Value);

(Entity, Statement, Attribute, Value);  
(Role, Entity, Statement, Attribute);  
(Class, Role, Entity, Statement);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

**Metagraph Model Specification.**

Resource : Functional URI wrapper.

(Context : Resource, Occurrence : Resource, Attribute : Resource, Value : Resource);

(Statement, Occurrence, Attribute, Value);

(Entity, Statement, Attribute, Value);  
(Role, Entity, Statement, Attribute);  
(Class, Role, Entity, Statement);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

**Dimensional Model Specification.**

(Value, Previous, Distance, Next);  
(Measure, Value, Previous, Distance);  
(Unit, Measure, Value, Previous);  
(Dimension, Unit, Measure, Value);  
(Concept, Dimension, Unit, Measure);  
(Resource, Concept, Dimension, Unit);  
(Statement, Resource, Concept, Dimension);

**Grammar Model Specification.**

Resource : Functional URI wrapper.

(Context : Resource, Occurrence : Resource, Attribute : Resource, Value : Resource);

(Statement, Occurrence, Attribute, Value);

(Entity, Statement, Attribute, Value);  
(Role, Entity, Statement, Attribute);  
(Class, Role, Entity, Statement);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

**Addressing / IDs / Encoding.**

Events / Messaging.

URIs, class, instance, context, occurrence IDs.

Context Kind / Signature: Predicate Kind from Subject / Object Kind.

Object occurrence of Predicate.

Encode behavior: iteration / jumps. Order statements (URIs APIs).

**Dataflow: Events. Reactive APIs.**

Model

Message

Interaction

Transform (Augmentation)

Flows / Routes (Augmentation, signatures)

Addressing

IDs Encoding

Processor

Producer

Consumer

Subscriptions (from metadata)

Queues.

**Protocols:**

Augmentation. Dialog. Query API.

Forms. Templates.

Ontology levels / layers.

Augment / Activate Resource (via addressing).

**Protocols:**

Hypermedia addressing and annotations. Extended content types.

Activation (parse gestures / render content according context). Browser.

**Protocols:**

Goal, Purpose: Fulfill Context.

Forms / Templates.

Dialogs: Model I/O (Message) flows.

**Models browsing / discovery APIs.**

HAL / OData like.

**Services (URIs APIs)**

Index

Naming

Registry

**Data / Reference Model.**

Functional declarative Semantics Specification.

**Ontology matching. Ontology levels.**

**Platform:**

Implementation (Protocols). Core, RX, Dataflow. Model: Reactive Dataflow.