**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:**

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

(Resource, Resource, Resource, Resource);  
(Entity, Subject, Attribute, Value);  
(Role, Entity, Attribute, Value);  
(Kind, Role, Entity, Attribute);  
(Class, Kind, Role, Entity);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

Graph Execution Semantics.

Ontology Matching. Upper ontologies. Primitives.

**Model I/O Dataflow:**

Model: Reactive entity applying Message Augmentation resolving Resource Set Specification.

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 returning augmented Message response.

Augmentation declarative definitions.

Message Resolution Algorithm.

Dialog (Protocol).

**Interaction Model:**

Message, Augmentation resolution (algorithm). Interaction Model event sourcing, distributed synch.

**Interaction Model:**

Message (data), Interaction (schema), Transform / Specification (behavior). URIs, Resource, Statement, Kind. Augmentations.

**Interaction Model:**

Message declaratively states Model Specification through Message Augmentations.

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 Aggregation, Alignment, Activation over Model. (Interaction Model Specification).

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

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

**Augmentations:**

For each Interaction Model Specification, perform and materialize following Augmentation applying events / distributed / inference sourcing for each layer. Augmentation and Message forms part of Interaction Model (besides Transform / Specification on behavior).

**CRUD Augmentation:**

Augmentation: CRUD.

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.**

(Resource, Resource, Resource, Resource);  
(Entity, Subject, Attribute, Value);  
(Role, Entity, Attribute, Value);  
(Kind, Role, Entity, Attribute);  
(Class, Kind, Role, Entity);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

**Source Model Specification.**

(Resource, Resource, Resource, Resource);  
(Entity, Subject, Attribute, Value);  
(Role, Entity, Attribute, Value);  
(Kind, Role, Entity, Attribute);  
(Class, Kind, Role, Entity);  
(Flow, Class, Role, Entity);  
(Behavior, Flow, Class, Role);

**Metagraph Model Specification.**

(Resource, Resource, Resource, Resource);  
(Entity, Subject, Attribute, Value);  
(Role, Entity, Attribute, Value);  
(Kind, Role, Entity, Attribute);  
(Class, Kind, Role, Entity);  
(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, Resource, Resource, Resource);  
(Entity, Subject, Attribute, Value);  
(Role, Entity, Attribute, Value);  
(Kind, Role, Entity, Attribute);  
(Class, Kind, Role, Entity);  
(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. Interaction, Message, Transform flows. Metamodel. Addressing, IDs, Encoding. Subscriptions from metadata. Queues.**

**Dataflow. Events. Producer, Consumer, Processor, Subscription. Model Node (matches / dispatch messages / statements applying corresponding model / model layers augmentations).**

**Protocols: Augmentation. Dialogs. Query APIs. Forms. Templates. Onto layers. Augment / Activate Resource.**

**Protocols: Hypermedia addressing and annotations. Extended content types. Context / Predicate Kind Signatures. Ontology levels (layers). Activation (i.e.: parse gestures / render content).**

**Protocols: Goal, Purpose: Fulfill Context (hierarchies). Forms / Templates. Dialogs: IO requests / prompts (arguments) flows.**

**Models browsing / discovery APIs. Services (Index, Naming, Registry URIs API).**

**Data / Reference Model: Functional declarative Semantics Specification.**

**Augmented Models: input source Models / layer wise Augmentation.**

**Ontology levels: from Augmented models inputs.**

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