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

RDF / OWL, Graphs, Triples, Quads introduction.

Models, Quads CSPOs: Object Graph Representation as RDF Quads.

URIs, Resource, Statement, Kind APIs. RDF Backend. URI Service Functional Implementation APIs. Context Kind Signatures. Datasources / Backends / Services (URIs APIs). Encoding. Event driven nodes. Model, reactive entities.

Model Layers: data, schema, behavior class / instance hierarchy. Model layers URIs. Upper ontology. Primitives (ontology matching: prev / next / etc.). Specifications (Message).

Message : Resource Set Specification. Interaction: Message, Transform mapping. Resolution (Interaction Algorithm). Interaction event sourcing, distributed synch.

Model: Reactive Entity applying Augmentation from Message returning Transform Resource set.

Ontology: upper, levels. Message Transform (Interaction). Graph Execution Semantics. Algorithm. Ontology Matching.

Interaction Model: Message (data), Interaction (schema), Transform (behavior). URI, Resource, Statement, Kind.

Interaction Model: Specification (behavior). For Model Message inputs and URIs events (data Messages) Model Augmentation.

Augmentation: Aggregation, Alignment, Activation for each Model type Specification.

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

Augmentation: Model IO. render aligned Transform (models / layers) on Model from inputs (Message parsed from data, schema, behavior layers events inputs / URIs Statement from data layer events inputs). Augmentation dataflow: Interaction order.

Specification: Interaction mapping Message / Transform as declarative statement of Model. Metacircular interpreter. Single Model reifying specifications.

Augmentation: Interactions (Message / Transform). Each Augmentation populates corresponding Models (Specifications) performing CRUD, inference, aggregation and classification over source Model layers.

Specification Augmentation. For each URI sources input Statement: apply Interaction Model (recursively over Transform results). CRUD.

Augmentation: CRUD.

Specification Model: Source.

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

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

Augmentation: Context Aggregation. Specification Model: Grammar. Classification (aggregate quads contexts / occurrences).

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

Interactions Model Specification. Metacircular (reifies Source, Dimensional, Grammar, Metagraph).

Source Model Specification (from Interactions). Event sourcing and backend URIs).

Transform result: materialize Model Augmentations. Interaction Model from persistence: load / materialize Transforms.

Interaction Model Specification: Interaction (functor M, T, distributed via Message), Message : M, Transform : T. Model layers / resources Augmentations / CRUD: Interaction (DIDs).

Source Model Specification.

Dimensional Model Specification.

Grammar Model Specification.

Metagraph Model Specification.

Addressing / IDs / Encoding (Events / Messaging). URIs, class, instance, context, occurrence IDs. Context Kind / Signature: Predicate Kind from Subject / Object Kind. Object occurrence of Predicate.

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).

Model: Reactive Node. Reacts to input Message going through corresponding aggregated layers till URIs IO APIs. URIs react / resource statements: Augmentation Application till Message set specification layer.

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.

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

Augmented Models: input source Models / layer wise Augmentation.

Ontology levels: from Augmented models inputs.

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

Model: build (load / persist) from Interaction metamodel Message / Transform DIDs records event sourcing. URIs quad store / backend CRUD (APIs).