**Contents: Mision / Vision**

Distributed Knowledge Base. Functional Syndicated Application Framework (Hypermedia: use cases).

Use Cases (problem / solution).

Problems description.

Distributed systems / micro services access to shared data. Shared data consistency. Ontology matching. Integration (EAI / ESB). Introduction of new features / products integrating over existing (linked) data with Semantic capabilities and enhancements.

Integration by augmentation.

Integration by extension.

Declarative Application Design.

**Solution Approach**

Ontology matching (data, schema, behavior alignments).

Hypermedia Dataflow Activation (reactive / event driven knowledge based contents).

Features:

Augmentations (aggregate aligned sources data / schema / behavior enabling interoperation).

Aggregation.

Alignment.

Activation.

Functional "Dialog" Augmentation Semantics.

**RDF Introduction: Graphs, Triples, Quads**

**RDF for Object Graph Representations**

ToDo.

**Models**

Base Model structure / Context layers hierarchies is as follow:

Resource : Functional URI wrapper.

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

(Statement, Occurrence, Attribute, Value);

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

Models have layers in class / instance roles (except for input layer) and each upper layer aggregates functionally over the previous:

Input Layer (Resource).

Data Layer (Statement instance, Entity class).

Schema Layer (Role instance, Class class).

Behavior Layer (Flow instance, Behavior class).

Explain Context hierarchies (Facets Resources) / Aggregation / Roles. ToDo.

Ontology Levels:

Models have “Ontology” Levels. Levels are Layers (of the Model) which are feed into its input Layer with (instance) Statements aggregated from initial input data (Data Level). Schema Level instances feeds the Model input conforming a Session (context / grammars) ontology Level. Then, behavior Level instances feeds the Model input conforming an Interaction (behavior) ontology Level.

Examples: Source, Session, Interaction declarative application protocol use case. ToDo.

Model Facets:

Models have “Facets” which renders the different ways Model data / schema / behavior could be regarded and used for different purposes, from application development to Business Intelligence and Ontology Matching.

Facets are implemented the same way ordinary Model Resource Contexts and layers. Each Facet implements its own Resource URI wrapper (same URIs, ontology matching, provenance of aligned URIs, Facet pivoting). Then, each Facet has its own Model Context Resource hierarchies having Augmentation / Dataflow functors as Model Resource(s) does.

Semantic / Semiotic Facet.

Dimensional Facet.

Functional (Model) Facet.

**Meta Resources**

URI, Resource, Statement, Context, Attribute, Value, CSPO, Kinds. Hierarchies. APIs.

**Meta Model**

Reify Layers, Facets, Levels in a Model with Meta Resources.

ToDo.

**Services, URIs, Resource(s) Functional APIs**

ToDo.

**IDs: Addressing / Encoding**

Message - Model - Template (data) - Augmentation (functor) - Transform (interaction) - Model - Message.

**Augmentation**:

Reactive Context Kind (matching signatures) dataflow.

Message - Model - Template (data) - Augmentation (functor) - Transform (interaction) - Model - Message.

Implementation API: Node / Container. Services (URIs Context Kind signatures resolution).

Core Services: Activation Augmentation (Naming).

Core Services: Alignment Augmentation (Index).

Core Services: Aggregation Augmentation (Registry).

Core Services: RDF / OWL Backend (endpoint, reasoning, persistence).

Core Services: DIDs Persistence (sync Node state: events sourcing).

Core Services: Protocol (I/O). Node, Session, Intetaction levels. Base Connector Augmentation API. Event driven URIs dialog / prompts protocol adapters.

**Dataflow**

ToDo.

**Implementation**

Languages, Patterns, APIs, Frameworks. (Container, Node, Model, Service, etc.).

**Deployment**

Protocol plugins (Protocol Service) Connectors. Runtime. Core Services. Endpoints. Dataflow.