Monads / Functors examples (order: comparables / upper / dimensional ontologies):

Kind<SubjectClass<Resource>, Set<PredicateClass<Resource>>>, others. Example: Subject / Predicate(s). (quads prev / next relation).

Metaclass<ObjectClass<Resource>, Set<OccurrenceClass<Resource>>>, others. Example: Object / Statement(s). (layers parent / child occurrences relation).

Class<ObjectClass<Resource>, Set<ContextClass<Resource>>>, others. Example: Resource / Kind(s) (class definition relations: extension / intension, layers parent occurrences prev relation).

Backend: RDF(S), OWL, Resource Services (Naming, Index Registry), ISO, Shapes, IDs matching. DIDs Backend.

Runtime: Objects, Events. Distributed Runtime Functional (Dataflow) reactive Resource objects / models (forms / flows).

Deployment: Spring / Vertx / others. HATEOAS / HAL APIs (Encoding / Dialog Endpoint Protocol). Environment / Container: Messaging / Services. Models I/O.

Integration Connectors: Runtime Model embedded Resources. Sources as reactive stream objects. API.

Integration Clients: Runtime Model embedded Resources. Services as reactive stream objects. API.

Integrations (augment / extend): RDBMS (R2QL / R2ML) / Rules / BPM, KIE / Teiid / Metamodel / Olingo / Solid / ERP / CMS / Feeds (Hypermedia Resources / Apache Stanbol) / Browser / Messaging (JMS overlay, Apache Camel / ActiveMQ: ServiceMix Connectors / Clients) / ML, Big Data (Apache Spark).

Dialog: Encoding / Protocol. Activation, Location, Context (Hypermedia type / context interfaces). Augmentations.

Encoding: XML / XSL / XPath / XLink / XForm / XPointer / XQuery (RDFS / OWL / ISO DM / RM Forms / Functors / DOM). Endpoints (streams / signatures): documents (messages) / resources (paths).