* Layers, input layer: Model Roles. Aggregation: Layer Roles shifting until full Layers Roles Statements. Layer: Augments Models.
* Core Statements Roles Resource Interleaved Model (Infer Types, PKs, FKs):
* Occurrence: (Class, Instance, Attribute, Value);
* Occurring: (Class, Instance, Occurrence,  Role);
* Statement Transforms / Relations: order, equivalence, roles, etc.
* Resource of Resource Monad: Occurrence  / Occurrings Quads CSPO Members Aggregation Transforms.
* Resource(t : T) :: contexts :: subjects :: predicates :: objects : Resource(u : U)
* Resource of URNs: Aggregated CSPOs Transforms of Occurrences / Occurrings. Matchings.
* Resource.of(Resource / Class, Instance, Occurring Attribute / Occurring URN, Attribute Value / Occurring Role) :: contexts :: subjects :: predicates :: objects : Resources(CSPO / Resource / occurrence / occurring : URN).
* Resources of Resources: Occurrences / Occurring CSPOs Transforms Matching Wrapped / Wrappings / Transforms Resource Types Shapes Matching Templates. Data flow, apply transforms, order, lists.
* Resource.of(templ : Templ) :: contexts : Resources(c : Context).
* Layer Template Mappings:
* Template : Context : Statement : Resource Layer Roles Monads.
* Layers Quads Aggregation: rotating value role types from previous layer to next layer from V to C):
* Canonical Template Mapping Layers: Aggregation of Template Matching Models Layers until first layer Value is wrapped into final layer Context. Perform Layers Augmentations.
* Monads. Wraps Models Roles. Matching: Patterns wrapped Resource Roles values: placeholders, variables, wildcards, reified instances. Example: Template wraps previous layer Value as Context, Layer Context Value wrapped as Subject value (shifting).
* Core Roles (Wrappers):
* Resource : Monad(x : Resource);
* Statement : Monad(x : Resource) : Resource, Quad DTO.
* Context : Monad(x : Resource) : Statement;
* Template : Monad(x : Resource) : Context.
* Template : Context : Statement : Resource
* Input Layers: Core Statements Roles Resource Shapes: (interleaving graphs models) : Statements CSPO.
* Occurrence Input: (Class, Instance, Attribute, Value);
* Occurring Input: (Class, Instance, Occurrence, Role);
* Output Layer: (Template, Context, Statement, Resource); Input OPSC Resources wrapped in output Layer Roles.
* Sets, Graph, Roles: Services / Augmentations helper Models / Facades.
* Example:
* (Amantes, Pedro, Ama, María);
* (MariaAmadaPor, Amantes, Pedro, Ama);
* (Amar, MariaAmadaPor, Amantes, Pedro);
* (PedroAmaA, Amar, MariaAmadaPor, Amantes);
* (Amor, PedroAmaA, Amar, MariaAmadaPor);
* Models:
* Core: Occurrence, Occur. I/O.
* Helper:
* Sets
* Graph
* Roles
* Augmentations:
* Model Augmentations:
* Alignment: Data Matching. Resources.
* Alignment::match
* Alignment::perform
* Alignment::greaterThan
* Alignment::equals
* Alignment::lessThan
* Activation: Schema Matching. Kinds.
* Activation::match
* Activation::perform
* Activation::superTypeOf
* Activation::sameTypeOf
* Activation::subTypeOf
* Aggregation: Behavior Matching. Contexts Flows.
* Aggregation::match
* Aggregation::perform
* Aggregation::beforeThan
* Aggregation::contains
* Aggregation::containedIn
* Aggregation::afterThan
* Domain Augmentations:
* Transforms Reified in Layers Contexts. Pattern Matching Template Layer resolved:
* Mapping::match
* Mapping::apply
* Mapping::Context
* Mapping::Subject
* Mapping::Predicate
* Mapping::Object
* Encode reified Template Mappings / Transforms. Patterns:
* Model / Domain Augmentations Mappings / Transforms:
* (Wrapper, Wrapped, Mapping, Transform);
* Next Layer step: match / apply Augmentations.
* Layers Template: Layer of CSPO Data Flow Patterns Resolution Resources : Reified (meta) Resources.
* Once Models Layers Matrix are built and populated / aggregated:
* Layers steps Augmentations: perform shifting and wrapping of aggregation values. Layer step Template Mapping: Layer::nextLayer : Layer, for each Layer, match / performs.
* Template, Mapping, Pattern
* (Template, Context, Statement, Resource);
* Resource :: occurrences :: roles :: contexts : Resource