Resource Monad : Resource<ResourceOccurrence>;

ResourceOccurrence<Representation<ContentType>>;

ContentType (Data / Transforms)

* onOccurrence transform
* getOccurrences(S, P, O) transform
* getOccurringContexts(S, P, O)
* fromRepresentation(Representation)
* toRepresentation(ContentType)

Representation<ContentType> : ContentType instance

* ContentType schema encoded State

ResourceOccurrence<Representation<ContentType>>

* onOccurrence(ResourceOccurrence)
* getOccurrences(S, P, O)
* getOccurringContexts(S, P, O)
* getAttributes() : Attributes
* getAttribute(Attribute)
* setAttribute(Attribute, Value)

FCA contexts? Prime IDs? Sets contexts? Dimensional contexts? Activation contexts? Hierarchies?: ResourceOccurrence Models Schema?

ResourceOccurrence(s)

* ID (URN Occurrence) : ResourceOccurrence
* Statement (ID Occurrence) : ID
* SPO (Statement Occurrences) : Statement
  + Subject
  + Predicate
  + Object
* Kind<Player, Attribute, Value> (Role / Type. SPOs Occurrence, Interface)
  + SubjectKind : Subject implements Kind<Subject, Predicate, Object> (Subject Occurence)
  + PredicateKind : Predicate implements Kind<Predicate, Subject, Object> (Predicate Occurrence)
  + ObjectKind : Object implements Kind<Object, Predicate, Subject> (Object Occurrence)
* Graph (Kinds Occurrence) : Kind

ResourceOccurrence(s) Activation (Events. ContentType handled, Resource Monad bound):

Events: Context::onOccurrence(Occurrence) : Result;

ID::onOccurrence(Statement) : SPOs.

Statement::onOccurrence(SPOs) : Kinds.

SPOs::onOccurrence(Kind) : Graph.

Kind::onOccurrence(Graph) : Graph?

Graph::onOccurrence(Graph) : Graph?

getOccurrences(S, P, O)? (CPPE / RCV / FCA / Kinds / Alignment schema / instances inference. Filter / query / traversal).

ID::getOccurrences(S, P, O) : Statement;

Statement::getOccurrences(S, P, O) : SPO;

SPO::getOccurrences(S, P, O) : Kind;

Kind::getOccurrences(S, P, O) : Graph;

Graph::getOccurrences(S, P, O) : Graph;

Events: (Context, Occurrence, Result) Occurrences.

Events feedback to Main Event Loop onResource(Context, Occurrence, Result) for further augmentation. Resource transform / augmentation logic in each event handler (Content Type).

Main Event Loop: Naming, Registry, Index stream steps pipeline.

Datasource input, Producer output.

onResource(Subject : Context, Predicate : Occurrence, Object : Result):

. Naming: Resource Factory. URN Crafting / Matching. Aggregation (type / state / order inference). Subject/Predicate/ObjectIDs::onOccurrence(Statement) : SPOs.

. Registry: Resource Repository. URNs Resolution / CRUD. Alignment (equivalence / upper matching, link prediction). Statement::onOccurrence(SPOs) : Kinds.

. Index: Resource Contents URNs Resolution (inferences, transforms). Activation (possible verbs / state changes / transforms). SPOs::onOccurrence(Kind) : Graph.