Resource: (URN, PrimeID, Content Type, State / Representation); Activation (Content Type handled): onOccurrence(Resource), ID::getOccurrences() : Statement, Statement::getOccurrences() : CSPOs, CSPO::getOccurrences() : Kind, Kind::getOccurrences() : CSPOs?

getOccurrences(Resource, Resource, Resource)? (CPPE / RCV / Kinds / Alignment schema / instances inference. Filtering / traversal).

Resource Monad? Resource<ContentType>;

Events: (Context, Occurrence, Result) Resources.

Resource Types (URNs: Data / Transforms)

. ID

. Statement (ID Occurrence) : ID

. CSPO (Statement Occurrences, Kind?) : Statement

. Kind (CSPOs Occurrence, CSPO?) : CSPO

. Graph (Kinds Occurrence) : Kind

. Events : (eventName, ResourceType<Consumes, Produces> Content Types).

Outputs feedback to Main Event Loop onResource(ResourceType, Consumes, Produces) for further augmentation. Resource transform / augmentation logic in each event handler (Content Type).

ID::onOccurrence(Statement) : CSPOs.

Statement::onOccurrence(CSPOs) : Kinds.

CSPOs::onOccurrence(Kind) : Graph.

Kind::onOccurrence(Graph) : Graph?

Graph::onOccurrence(Graph) : Graph?

Content Types

. Resource Types / Model encodings.

. Representations transform. Events transforms.

. Resource types hierarchies.

Representations<ContentType>

. Encoded Models.

Transforms

. Content Type bound registered transforms (Resource events).

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

Datasource input, Producer output.

onResource(Subject, Predicate, Object):

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

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

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