* Co relations (Algebras / CoSQL / NoSQL / LInQ):
* Model Layers. Augmentation steps.
* (Context, Occurrence / Statement, Attribute / Kind, Value / Entity);
* Discrete Layers (Relationships / Assertions). Continuos CEP: 0 Duration Measures / Axis Events Relations.
* (Relationship, Relation, Kind, Entity);
* Context, Roles, Causal, etc. Relations.
* Continuous Layers (Dimensions / Measures).  Discrete CEP: Order / Containment Relations.
* (Dimension, Measure, Unit, Value);
* Distance, Equivalence, Transportation, etc. Relations.
* Functional DCI: Monads / Functors / Data. Encoding.
* Wrapper Types (Data Roles).
* Wrapped types (Data Values).
* Functors: Dataflow Domain / Range specifications (Contexts). Encoding.
* Monads: Dataflow implementations (Interactions). Encoding.
* Roles. Encoding Monads Wrappers (Wrapped Model Resources):
* Class
* Metaclass
* Context
* Role
* Instance
* Occurrence
* Attribute
* Value
* Nodes: (Class, Metaclass, Context, Role); Contexts.
* Arcs: (Instance, Occurrence, Attribute, Value); Data / Interactions (Mappings).
* Nodes / Arcs Properties: Dataflow Models Mappings: Node maps Contexts, Arc maps Subjects / Occurrences. Layers: Role as Class / Value as Instances.
* Encodings:
* Models: Quads / Property Graphs. DCI / Dataflow.
* Lists Model. Roles / Order.
* Hierarchical Graph Encoding.
* Sets.
* Functional DCI / Layers abstraction.
* Functional Parser. CUD. (data) of Grammar (Functional scheme) Dataflow Entities.
* Functional Parser. Grammar (Functional scheme) Dataflow Entities.
* Functional Parser. Parse (executions / behavior). Dataflow interactions instances.
* Codat: Dataflow / Protocol Prompts (run at).
* Protocol:
* Stateful I/O (ordered contexts). Reactive encoded Message driven gestures (CQRP).
* CDI / ESB Runtime. Backends. Connectors. CAM / CAN / DIDs.
* Dataflow HATEOAS.
* Augmentations:
* Aggregation. Contexts / Occurrences.
* Activation. Roles / Types / Kinds.
* Alignment. Attributes / Values.