

Semantic Infrastructure Concept of Operations Contract Design Infrastructure

Semantic Infrastructure Concept of Operations Links

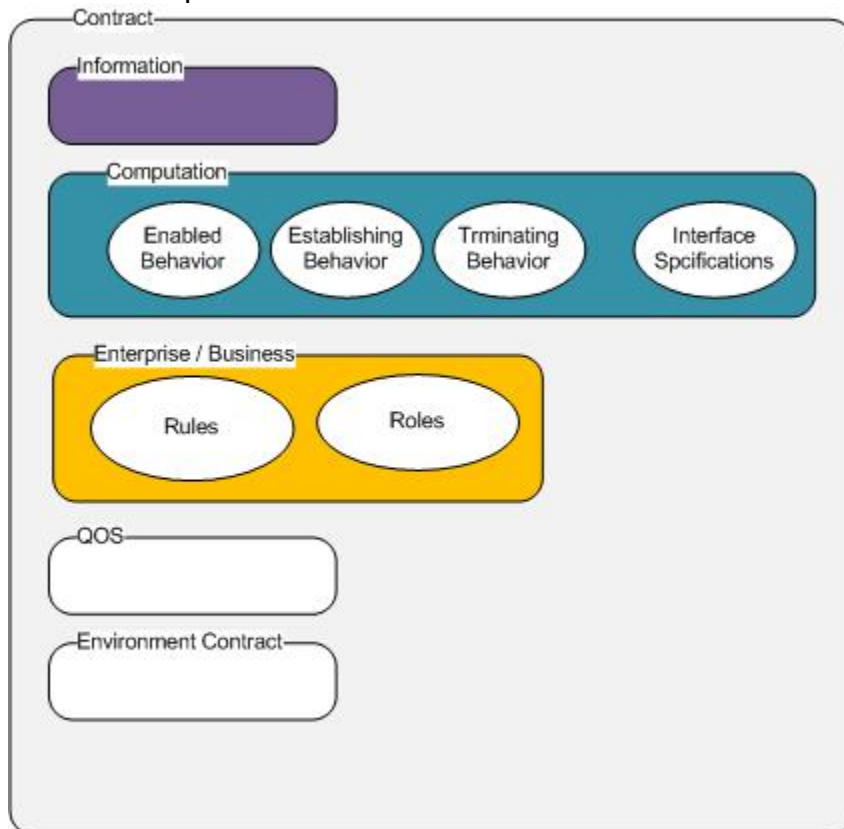
- [Semantic Infrastructure](#)
- [Vision](#)
- [Background](#)
- [Mission](#)
- [Objectives](#)
- [Overview](#)
- [Initiatives](#)
- [Stakeholders](#)
- [Alternatives](#)

The [VKC:Contract Design Infrastructure depicted in the architectural infrastructure diagram](#) (showing Deployment and Run Time imports Contract and Design imports Semantic) is part of the RM-ODP Enterprise viewpoint and comprises the means of capturing the logical components of a contract.

Currently these logical components are captured implicitly in code and in the design of applications. Certainly some fully specified contracts deserve to be captured, preserved, and reused where possible. However, capturing other contract components (and optionally realizing them) should be in scope for an enterprise such as CBIIT given its mission.

The RM-ODP Enterprise viewpoint metamodel indicates that that a Community specifies Contract(s) to fulfill specific Objective(s). The Contract is specified by particular Behaviors that are required in order to fulfill the Objective(s) and includes the Policies that pertain to fulfillment of the Contract. An example of capturing contract components and realizing them is the implementation of a durable service that is specified sufficiently to allow clients to consume components at run time without an early explicit binding. Other examples are discussed below.

Diagram Depicting Contracts Bringing Together Information, Computational and Enterprise Elements and Expressing Quality of Service and Environmental Requirements



The following are shown:

- Information
- Computation
 - Enabled Behavior
 - Establishing Behavior
 - Terminating Behavior

- Interface Specification
- Enterprise and Business
 - Rules
 - Roles
- QOS (Quality of Service)
- Environment Contract

A contract is an agreement governing all or part of the collective behavior of a set of systems. In addition to the behavior and enterprise objectives fulfilled by the contract for a given community, a contract specifies the policies: permissions, obligations, and prohibitions for the systems involved.

Policies, potentially depicted as rules, describe the permissions, obligations, and prohibitions of the roles involved in the contract.

Roles are identifiers for behaviors, capabilities, capacities, or competencies that are enlisted to fulfill the terms of a contract.

Information components are represented by the invariant, static and dynamic models and characteristics of those models that facilitate the behaviors as constrained by the permissions, obligations, and prohibitions (policies) established by the rules associated with the roles.

For example of the kinds of rules that need to be defined are those in the RM-ODP computational viewpoint defining a set of rules that constrain a computational specification. These comprise:

- *interaction* rules, *binding* rules and *type* rules that provide distribution transparent interworking;
- *template* rules that apply to all *computational objects* and *computational interfaces*;
- *failure* rules that apply to all *computational objects* and identify the potential points of *failure* in computational activities.

In NCI's ECCF, Behaviors are broken down into subclasses such that contracts have an establishing behavior, a terminating behavior (commonly called preconditions and postconditions), and an overall objective that is enabled by the contract being established. As well, they defined the interfaces that are bound to each role to support the permissions and obligations associated with that role.

An Environmental Contract is a contract between a system and its environment. It can describe both constraints on a systems behavior in the correct environment as well as requirements placed on a system's environment for the system's correct behavior, such as QoS.