

Computational Viewpoint

A research infrastructure (RI) provides a context in which investigators can interact with scientific data in a principled manner. To provide this context, an RI must support a portfolio of possible research interactions. These interactions can be realised by binding together different services via standard operational interfaces.

The Computational Viewpoint (CV) accounts for the major computational objects that can be found within an environmental research infrastructure, as well as the interfaces by which they can be invoked, and by which they can invoke other objects in the infrastructure. Each object encapsulates functionality that should be implemented by a service or other resource in a compliant RI. Binding of computational objects together via compatible interfaces creates a network of interactions that allows an RI to support the data related activities of its target research community.

The description of the CV is divided in three parts: **objects**, **CV Objects and Subsystems**, and **CV Integration points**.

- **CV Objects**: present computational objects according a generic architecture of the RIs.
- **CV Objects and Subsystems**: presents examples of how components are integrated for supporting the data lifecycle into five different subsystems.
- **CV Integration points**: defined to support the movement of research data between phases.



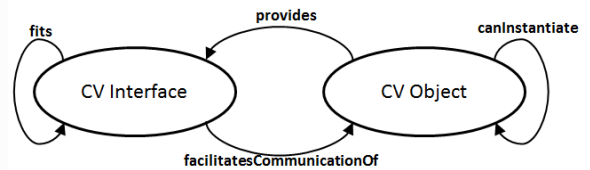
Note

Before proceeding, the reader may wish to study the pages on [How to read the Model \(Computational Viewpoint\)](#) and [How to use the Model \(Computational Viewpoint\)](#).



The Computational Viewpoint defines computational objects (CV Object) and interfaces (CV Interfaces) which enable their interaction.

The diagram below shows the main elements of the CV and their relationships. Each ellipse contains a concept. The arrows connecting the concepts are directed and indicate the relationship between concepts. The label of the link indicates the type of relationship. From this, the diagram indicates that a CV object provides a CV interface, as indicated by the **provides** relationship. Similarly, a CV object can create another CV object, as indicated by the **canInstantiate** relationship. In this same way a CV interface can fit another CV interface, this is indicated the **fits** relationship.



Computation Viewpoint components and their relationships