

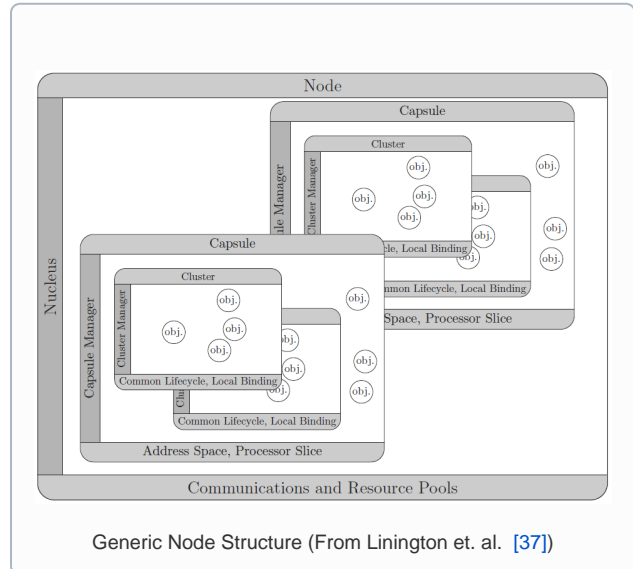
# EV Architectural Model (Draft)

The main value of the engineering viewpoint is in providing a technology neutral architectural framework or reference architecture that can be used as a basic tool for designing new systems or comparing existing infrastructure technologies for distributed processing<sup>1</sup>. This description includes the basic objects that are composed to provide the system functionalities, the structural components that make this integration possible and the basic architecture of nodes ([Node Architecture](#)) and the architecture of the communication structures that connect those nodes ([Channel Architecture](#)).

## Node Architecture

The node architecture identifies a number of functions that are commonly found in the platforms supporting application objects. Providing a technology neutral architecture for describing them helps to identify the commonalities between different infrastructure solutions, such as JEE or .NET, and so makes the integration of different platforms simpler<sup>2</sup>. Several of these functions are concerned with groupings of objects that are important when managing essential system properties, such as resource consumption, fault tolerance or persistence. These groupings, include such elements as nodes and, within them, sets of clusters, with cluster managers, sets of capsules, with capsule managers, and a nucleus.

It is important to understand the kinds of functionality required to support provision of distribution transparencies, the basic mechanisms needed to achieve those functionalities, and how they can be structured and assigned to the architectural elements of the engineering viewpoint. This provides a reference architecture, against which existing distributed processing systems can be compared and evaluated, or with which tomorrow's systems can be designed. Using it helps technical architects specify or select required infrastructure characteristics<sup>2</sup>.

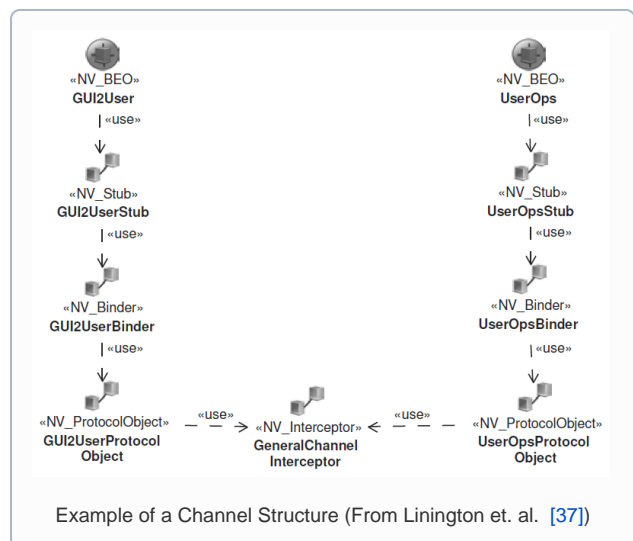


## Channel Architecture

The engineering viewpoint is also concerned with defining a channel architecture that represents the communication infrastructure, which allows engineering objects to interact (across container barriers)<sup>3</sup>.

All channels involve the same functions, using specialized engineering objects to implement the required functionality in an ordered manner. However, different specific communication architectures may each organize or interleave these functions in their own way<sup>3</sup>.

Normally there is no need to specify these elements in detail because they are provided by the underlying middleware platform (some of which are themselves standardized). However, there are occasions in which we want to model them functionally, in order to specify some of a channel's properties, or to express requirements on it - such as performance requirements on the channels or security constraints on the protocol objects or interceptors<sup>3</sup>.



<sup>1</sup> See Linington et.al. p91 [37]

<sup>2</sup> See Linington et.al. p. 93-96 [37]

<sup>3</sup> See Linington et.al. p. 96-98 [37]