LEGION

It is a Object oriented middleware project which has metasystems software for grid applications. The goal of the Legion project is to promote the principled design of distributed system software by providing standard object representations for processors, data systems, file systems, and so on. Legion applications are developed in terms of these standard objects. Groups of users can construct a shared virtual workspace to collaborate on research and exchange information.

An Interface Definition Language (IDL) is defined to describe the method signatures (name , parameter, and return values) supported by the object interface. We could see that these objects provide a scalable persistence mechanism by storing the inactive objects (objects in "inert" state) to the secondary storage.

Some of the important characteristics of Legion systems are summarized below.

Everything is an object

In a Legion system, Legion Object represents a variety of hardware and software resources, which respond to member function invocations from other objects in the system. Legion defines the message format and high-level protocol for object interaction (through IDL), but not the programming language or the communications protocol.

Classes manage their own instances

Every Legion object is defined and managed by its class object. Class objects are given system-level responsibility; classes create new instances, schedule them for execution, activate and deactivate them, and provide information about their current location to client objects. These classes whose instances are themselves classes are calledmetaclasses .

Users can provide their own classes

Legion allows its users to define and build their own "class" objects. This enables the Legion programmers to have a flexible architecture model for their "metaclasses" with the capabilities to determine and even change the system-level mechanisms of their objects.

Core objects implement common services

Legion defines the interface and basic functionality of a set of core object types that support basic system services, such as naming, binding, object creation, activation, deactivation , and deletion.

Some of the core objects defined by the Legion system are:

- Host objects : Abstractions of processing resources which may represent a single processor or multiple hosts and processors
- Vault objects : Provide persistent storage for scalable persistence of the objects
- Binding object : Maps the object IDs to the physical addresses
- Implementation objects : Allow legion objects to run as processes in the systemand contain a machine code that is executed on a request to create the object or activate it.

gure 1.5 shows Legion object A with its class object (metaclass) and the corresponding basic system services.

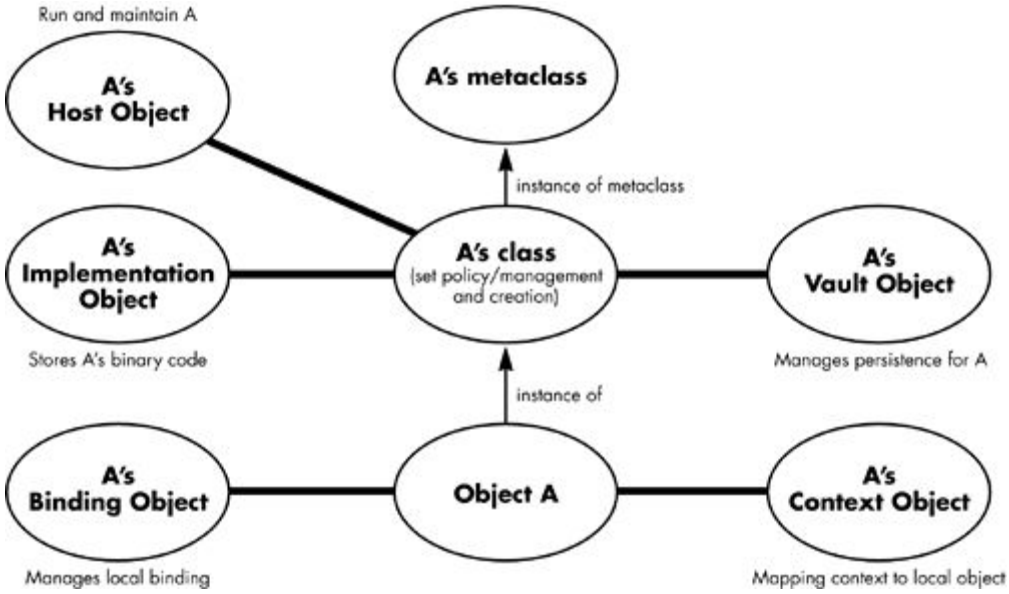

Figure 1.5. Legion core object and relationship.

In 1997, the first Legion toolkit was released, and in the following year, Applied Metacomputing (later relaunched as Avaki Corporation) was established to exploit the toolkit for commercial purposes.

Source : http://elearningatria.files.wordpress.com/2013/10/ise-viii-grid-computing-06is845-notes.pdf