

# GENIE CLIENT-SERVER ARCHITECTURE



g

***GE Medical Systems***

---



## GENERAL ADVANTAGES OF CLIENT-SERVER ARCHITECTURE

- ◆ Servers can be used to manage scarce resources and to share these among several client processes.
- ◆ The server and the client application can be developed and released independently
- ◆ Computer systems can be scaled far easier, and at lower cost, than many other architectures.
- ◆ Allows functionality to be grouped and optimized within a server (e.g. Image processing, Hardcopy, and Database...).
- ◆ The code present in a server can be reused by several different client applications.
- ◆ The client application (user interface) remains interactive as the heavy processing is deferred to the server.... The UI does not lock up.
- ◆ The client can be delivered on a low cost platform.

## GENIE CLIENT-SERVER, A BASIC PRIMER:

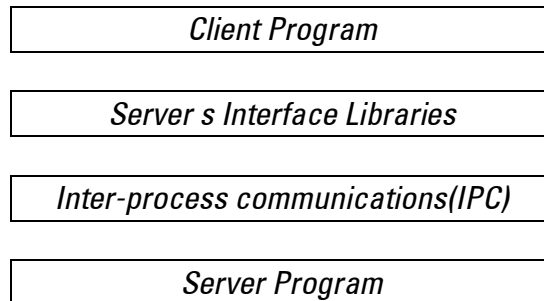
The GENIE software is designed around these key concepts:

- ◆ Client-server architecture
- ◆ Event-driven programming
- ◆ Object-oriented design
- ◆ Standards (X11/Motif, ANSI C, IAP, DICOM 3.0)
- ◆ Hardware Independent (No Accelerators)

## CLIENT-SERVER ARCHITECTURE:

Work is “contracted” from one program (client) to the other (server) using agreed upon protocol:

- ◆ The client sends the request to the server
- ◆ The server executes and sends acknowledgment



Virtual Resources:

- ◆ Database (DbS)
- ◆ Applications (LcS)
- ◆ Film Devices (HcS)
- ◆ Archive Devices (ArS)

Why separate the execution?

- ◆ Servers can be run on a faster platform.
- ◆ Server and client can be on different machines on a net.
- ◆ One server can service many clients simultaneously
- ◆ Servers usually manage expensive resources (printers, archive, other central resources) for number of clients
- ◆ Server and client can be developed and released independently
- ◆ User interface remains interactive, even during heavy processing.

Servers come with client-side libraries which are considered the server's programming interface (API).

## EVENT-DRIVEN PROGRAMMING:



The underlying functions of GENIE programs are event driven: most functions are called by the servers (“back”) as a result of external events.

Most of the application work is done by in “callback” functions which are called by the servers.

A typical application control flow:

- ◆ The application connects to the servers, initializes, and goes into waiting loop.
- ◆ The X server notifies the application about new user input through a callback function. The application makes some server requests in response and returns to a waiting loop.
- ◆ The server notifies the application through a callback functions that requested work is complete(or an error has occurred). The application requests additional work or returns to the waiting loop.

The application has a small initialization function. Most of the code is in the callback function.

## **OBJECT-ORIENTED DESIGN:**

The underlying objects in GENIE are “servers within servers”. They can be thought of as independent programs.

- ◆ The type(name) of the object(program) is called “class”.
- ◆ Usually, an unlimited number of objects of the same class can “run” concurrently and independently.
- ◆ Each object initiated(asked to start running) by a “create” call and destroyed(asked to stop running) by a “delete” call.
- ◆ Objects communicate with applications through the API, and with each other, using typed communication links.
- ◆ GENIE protocols are classes/objects(CEqual, Renal...).
- ◆ Printer objects, print queue objects.
- ◆ Archive objects.
- ◆ GENIE object is base class.
- ◆ Processing card object.

Any application can utilize these objects, features without having to worry about implementation or changes, for example: CEEqual uses the RECON and Reformat object , SPECT uses the RECON, Reformat and Review objects.

Programmers can develop independently with well defined interfaces.

As with servers, the same objects can be used by multiple client applications which are watching it.

Each object has a set of attributes that control its behavior. The attributes can usually be set and get by the application.

Each object can also have a set of uninterpreted properties that applications can use to store additional, application specific information. Using notification, applications can coordinate themselves using properties.



## KEY GENIE SERVERS:

GENIE provides the following servers:

- ◆ X Server
  - ⇒ Allows GENIE to run on a lower cost platform.
- ◆ Hard Copy Server(HcS)
  - ⇒ Local or network access to a shared printer.
  - ⇒ Multiple job queues for multiple printers
- ◆ Database Server(DbS)
  - ⇒ Local or network access to image data and its header information.
  - ⇒ Fast and cheap.
- ◆ Processing Server(PrS)
  - ⇒ Local or network access to processing power and memory.
  - ⇒ Optimized for interactive processing of 2D, 3D datasets for visualization and measurements.
- ◆ Reconstruction Server(ReS)
  - ⇒ Optimized for processing of algorithms.
- ◆ Archive Server
  - ⇒ Serves the archive needs for many clients.
- ◆ Network Server
  - ⇒ Serves the networking(DICOM 3.0, Interfile, Starlink..) needs for many clients.

And also:

- ◆ Licensing Server(LcS)
  - ⇒ Controls right-of-use on network(floating licenses) for other servers and applications.
  - ⇒ Licenses can be arbitrary and as fine as desirable(Machine or user based).

## BENEFITS:

- ◆ Servers can be used to manage scarce resources and to share these among several client processes.
- ◆ The server and the client application can be developed and released independently
- ◆ Servers can be distributed over a network or on the same hardware.
- ◆ The client application programmer accesses the server through calls to a set of client side libraries. It is therefore unnecessary for an application programmer to have any detailed knowledge of the internal server processes in order to make use of them(e.g. A programmer can call a rendering routine for, say a MIP function, without having to know how to perform any of the mathematical operation involved in performing this task).
- ◆ Computer systems can be scaled far easier, and at lower cost, than many other architectures.
- ◆ Allows functionality to be grouped and optimized within a server(e.g. Image processing, Hardcopy, and Database...).
- ◆ Allows for asynchronous execution of several independent processes within an application as the servers can operate independently and concurrently.
- ◆ The code present in a server can be reused by several different client applications.
- ◆ The client application(user interface) remains interactive as the heavy processing is deferred to the server. The UI does not lock up.
- ◆ The client can be delivered on a low cost platform.



### **DISADVANTAGES:**

- ◆ Platform development takes longer due to the increased complexity. architecture.
- ◆ GENIE platform uses a standard off-the-shelf platform used in Nuclear Medicine, MR and CT applications.
- ◆ Process management issue has been addressed via initial machine set up.
- ◆ The communication overhead has been streamlined. The Serv/comm uses the fastest available communication transport.

### **SUMMARY CONCLUSION:**

The GENIE architecture has been carefully designed to take the full advantages of a client-server architecture while ensuring that the over head is minimized. This architecture clearly facilitates the platform independence to give users the right hardware platform horsepower and cost benefit at any given time, now and in future.