**What is XIP?**

- XIP is an open source environment for rapidly developing medical imaging applications from an extensible set of modular elements.
- Researchers will be able to easily develop and evaluate new approaches to medical imaging problems, and use them in a translational research setting.
- caGrid makes it possible to develop an XIP architecture that allows users to choose between remotely hosted grid-based components and data sources as well as locally available components and sources.
- Components may include analytic services, e.g. CAD algorithms, algorithms for quantifying changes in consecutive imaging studies, studies, algorithms associated with a 3-D visualization pipeline etc.
- Available data sources include NCIA, caGrid data services, DICOM data repositories, local databases, etc.
- XIP will be a reference implementation of the DICOM WG-23 Interfaces.

**Where did XIP come from?**

- XIP is a project funded by NCI’s caBIG™ program [caBIG-IM-99-01-31052006].
- The XIP project is coordinated by the Software Special Interest Group (SW SIG) in the caBIG In-Vivo Imaging Work Space (IVI WS).
- The XIP team is working closely with DICOM WG.
- The XIP Application development framework and builder tools are being derived from the VRA and RAO/Builder tools created by the Imaging Architectures program at Siemens Corporate Research (SCR).
-caGrid will strip out Siemens-proprietary code, replacing it with open-source code.
- The modified, enhanced package will be released with a free license via NCI.
- The Electronic Radiology Lab at Washington University’s Mallinckrodt Institute of Radiology will provide open-source reference implementations of XIP hosts and applications targeted toward IVI WS SW SIG needs.

**WG-23** addresses clinical integration and vendor inter-operability by defining standardized “plugs” and “sockets” (APIs) and **caBIG XIP** addresses an open-architecture integration development environment for rapid prototyping & collaboration based on 23 APIs.

**XIP Libraries**

- Sets of Open Inventor™ objects that XIP applications may utilize.
  - Standard XIP objects for interacting with, retrieving data from, and storing data to XIP or DICOM WG-23 Hosts via the DICOM WG-23 interface.
  - Standard XIP objects that extend the stock Open Inventor™ objects with functions useful for building medical imaging applications.
- VTK classes wrapped into Open Inventor™ objects to support visualization.
- ITK classes wrapped into Open Inventor™ objects for easy use in XIP applications.
- Custom objects supplied by developers to extend the standard XIP objects.
- Open Inventor™ provides a flexible string-based interface (e.g. for scripting or multiple programming language bindings) and can be serialized (e.g. for saving state).
- Many XIP libraries are host-independent, but if dependent on host services:
  - XIP libraries have internal dynamic ‘adapter’ libraries to allow for easy switching between XIP hosts or platforms (i.e. simply swap the dynamic library).
  - The XIP Application always sees the same Open Inventor™ objects regardless of host.
- XIP libraries may be auto-generated from existing class libraries (e.g. ITK, VTK), or may be custom-built from new or existing code.

**XIP Hosts**

- Provides the infrastructure in which XIP or DICOM WG-23 Applications run
  - Authenticates user
  - Manages installation, launching, and termination of XIP Applications
  - Provides data and services to XIP Applications
  - Accepts status information and results back from XIP Applications
  - Deals with auditing and controls access to services and data
  - Isolates the XIP application from the nature of databases, archives, networks, 
    and possibly image data formats
  - Manages caGrid interactions and security
  - Manages access to DICOM networks, objects, and services
  - Maps images and associated meta-data from various sources between their native form and a common form useable by the XIP application
  - Handles workflow issues
  - General Purpose Workflow support, following IHE profiles
  - Supports any application that follows the DICOM WG-23 Application Hosting Interface Standard

**XIP Applications**

- Interconnected Open Inventor™ engines, nodes, and manipulators from XIP Libraries (Model/View/Controller design pattern)
  - Engines enable the creation of processing pipelines and animation
  - Nodes support the concept of scene graphs, which are hierarchical structures of objects describing what needs to be visualized in 2D/3D
  - Manipulators handle input devices, measurements and coordinate transforms in response to user interaction via a simple event model
  - Focused on the processing logic, not the infrastructure or data format
- Skinnable GUI engine for different look-and-feel on different platforms

**Summary**

- The XIP platform will make it easier and less expensive to access specific post-processing applications at multiple sites, simplifying clinical trials, and most importantly, increasing the uniformity of imaging and analysis.
- Imaging applications developed by research groups will more easily be accessible within the clinical computing environment, simplifying workflows and speeding data processing and analysis.
- Once validated, the software should be readily transitioned to products through streamlined Federal Drug Administration (FDA) approval processes due to the re-use of tested libraries and open source development processes that have already been used in approved devices.

**XIP Libraries**

- Sets of Open Inventor™ objects that XIP applications may utilize.
  - Stock Open Inventor™ object for 3D graphics, modeling, UI, picking, etc.
  - Standard XIP objects for interacting with, retrieving data from, and storing data to XIP or DICOM WG-23 Hosts via the DICOM WG-23 interface.
  - Standard XIP objects that extend the stock Open Inventor™ objects with functions useful for building medical imaging applications.
  - VTK classes wrapped into Open Inventor™ objects to support visualization.
  - ITK classes wrapped into Open Inventor™ objects for easy use in XIP applications.
  - Custom objects supplied by developers to extend the standard XIP objects.
  - Open Inventor™ provides a flexible string-based interface (e.g. for scripting or multiple programming language bindings) and can be serialized (e.g. for saving state).
- Many XIP libraries are host-independent, but if dependent on host services:
  - XIP libraries have internal dynamic ‘adapter’ libraries to allow for easy switching between XIP hosts or platforms (i.e. simply swap the dynamic library).
  - The XIP Application always sees the same Open Inventor™ objects regardless of host.
- XIP libraries may be auto-generated from existing class libraries (e.g. ITK, VTK), or may be custom-built from new or existing code