**Introduction**

Teleradiology provides need-to-diagnose and report images of various modalities and sub-specialties in a fast and efficient manner. A single viewer application does not suffice for advanced diagnosis of images from different modalities. Suite of clinical applications from multiple vendors needs to be integrated on a single system to order to aid radiologists reading cases on disparate systems. Teleradiology medical imaging platform (TMIP) based on plug-in architecture is implemented, such that different clinical review and analysis applications can be easily plugged into the platform. Application hosting API specified in DICOM Standard (WG-23) is considered as basis for the design of TMIP hosting system owned by TMIP manages upgrading of all the components that are hosted by TMIP. Hosted application should communicate with the hosting system as where the upgrades are published. Plug-in manages the automatic download and installation of components. Each of the components can upgrade independently and should be backward compatible with other dependent modules.

**Challenges in Implementing Application Hosting for Teleradiology Platform**

- Software Upgrading: As in any cloud-based application, software upgradation happens in an automated fashion for TMIP platform as well. TMIP manages upgradation of all the components that are hosted by TMIP. Hosted application should communicate with the hosting system as where the upgrades are published. Plug-in manages the automatic download and installation of components. Each of the components can upgrade independently and should be backward compatible with other dependent modules.

- Data Retrieval Management: Hosting System owns the data retrieval manager and intelligently downloads data from proprietary EDGE Servers. This data communication protocol is abstracted from hosted applications. Output results from hosted application are stored in a folder specified by the hosting system and storing in the server falls under host’s responsibility.

**Future Work:**

- Design adaptations for non-compliant third party plug-ins
- Design appropriate abstract models for data exchange
- Extend framework for distributed execution
- Achieve Platform Independency