Security image sharing between PACS and EHR

by Weina Ma

June 25, 2014

1. *Background Investigation*

We did background investigation related to ORF project, including healthcare systems and standards in real-world, modern authentication and authorization techniques, standards and specifications in terms of privacy and security requirements in distributed healthcare systems and multi-agent technology.

1.1 PACS

The architecture of a PACS is shown in Figure 1.The main components of a PACS are: image modalities, acquisition gateways, PACS controller and associated database and server, long term and short term archives, and workstations. A PACS is capable of acquiring, storing, transferring and retrieving medical images in healthcare environments. PACS mainly rely on DICOM and HL7 standards for communicating with different image modalities (defined below). *Image modalities* are image acquisition components that capture medical images of patients. These include: X-ray, MRI, CT, fluoroscopy, etc. Some modalities capture images in digital format while others in analog format. For example, some X-ray scanners provide images in analog format. To deal with such situations PACS have an *acquisition gateway*, which is usually a computer system that is located between the image modalities and PACS environment.
They convert analog images to digital format, thereby making it compatible with PACS. Acquisition gateway, if connected to *Hospital Information System* (HIS), can add additional information to patient images by using HIS interface and the HL7 protocol. *PACS controller* is the most important component of the system. It has multiple functionality as image storage is concerned. Images obtained from modalities are stored in the PACS database. When an image arrives, the text associated with the image is extracted; the image is compressed; the workstation to which the image has to be forwarded is determined; and the image is stored in an archive if it is not meant for immediate use. The PACS database, server and archiving systems are associated with the PACS controller. The PACS database is responsible for grouping and ordering of the images. It is connected to the *Radiology Information System* (RIS) to retrieve the data associated with the patient images. After properly arranging the images the recent ones are stored in the *short-term archive* and further to the *long-term archive* for future use. The user at his workstation views all the medical images. *Workstations* include software that supports procedures for accessing images from the image database, processing images, and all user activities while working with medical images.

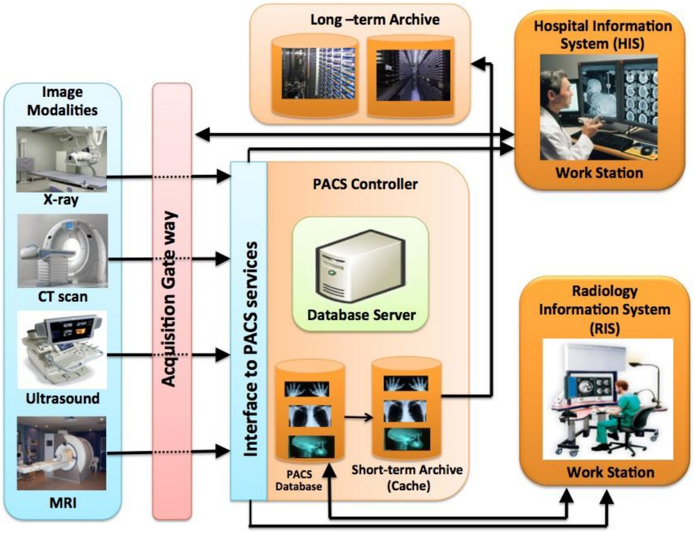


Figure 1. PCAS Architecture

1.2 DI-r

The DI-r project initially started with eight hospitals replacing their film X-rays by PACS systems. Sanction is given to authorised medical personnel to share medical images securely with other members of a particular DI-r. The data stored is retrieved in the digital format, which makes it easier for communication. The system benefits the patients as well as the clinicians. From the patient's point of view, the DI-r reduces unnecessary travel, waits times in hospitals, repeated examination and after all reduces the number of times the patient has to be exposed to radiation. From the clinician's point of view medical images could be retrieved anytime from anywhere in the world. Further, it helps in faster diagnosis without wasting time for image recovery. In Ontario, the hospitals are partitioned into four clusters, each with a separate DI-r. These four DI-r clusters are: the Southwest Ontario Diagnostic Imaging Network (SWODIN), the Hospital Diagnostic Imaging Repository Services (HDIRS), the Northern and Eastern Ontario Diagnostic Imaging Network (NEODIN), and the Greater Toronto Area West Diagnostic Imaging Repository (GTA West DI-r). Such DI-r clusters can further be integrated into a nation-wide document sharing infrastructure, which can also be integrated with a nation-wide EHR to provide full accessibility to medical images. There are a number of challenges for a fully functional infrastructure. The vendors are not yet compliant with an implementation of the imaging interoperability standards, namely “Cross-enterprise Document Sharing for Imaging” (XDS-I) and the “Integrating the Healthcare Enterprise Patient Identifier Cross Referencing"” (IHE PIX). The Enterprise Master Patient Index (EMPI) should also be incorporated to achieve wide-scale interoperability. This ensures that each patient is represented only once across the imaging systems.

1.3 DICOM

DICOM (Digital Imaging and Communications in Medicine) is a fundamental and universally-adopted standard in digital medical imaging domain for handling, storing, printing, displaying and transmitting information. DICOM enables digital image acquisition devices, digital image archives, digital camera, printers, scanners and workstations coming from different manufactures into one PACS system. From the DICOM point of view, all real-world data (i.e., related to patients, studies, images, image acquisition devices, image viewer applications) are structured as objects with respective properties or attributes. DICOM maintains a list of more than 2000 standard attributes, known as the DICOM data dictionary, for the sake of ensuring consistency in attribute name and processing. Therefore, as soon as the communication message between DICOM applications is captured and explained as data attributes, the DICOM message can be transmitted and processed between various applications. DICOM is defined on top of the standard Internet protocols and should be seamlessly applied both internally and externally on distributed systems. However, in practice due to some limitations of DICOM point-to-point connection model it is hard to be used as a communication standard in large-scale DI-r systems. Moreover, DICOM does not specify any build-in security, which makes it inadequate to address security and privacy requirements in cross-enterprise domain.

1.4 IHE Cross-Enterprise Document Sharing (XDS.b)

IHE XDS.b facilitates the registration, distribution, searching and retrieving of patient electronic health records across a group of affiliated enterprises. Clinical document sharing is achieved through federated document repository and document registry: i) a document repository is responsible for storing documents in a transparent, secure, reliable and persistent manner; ii) a document registry is responsible for storing metadata of those documents stored in repository so that the document of interest may be easily searched and retrieved. Figure 2 presents XDS.b two basic transactions: provide and register document, search and retrieve document. Following are the involved actors.

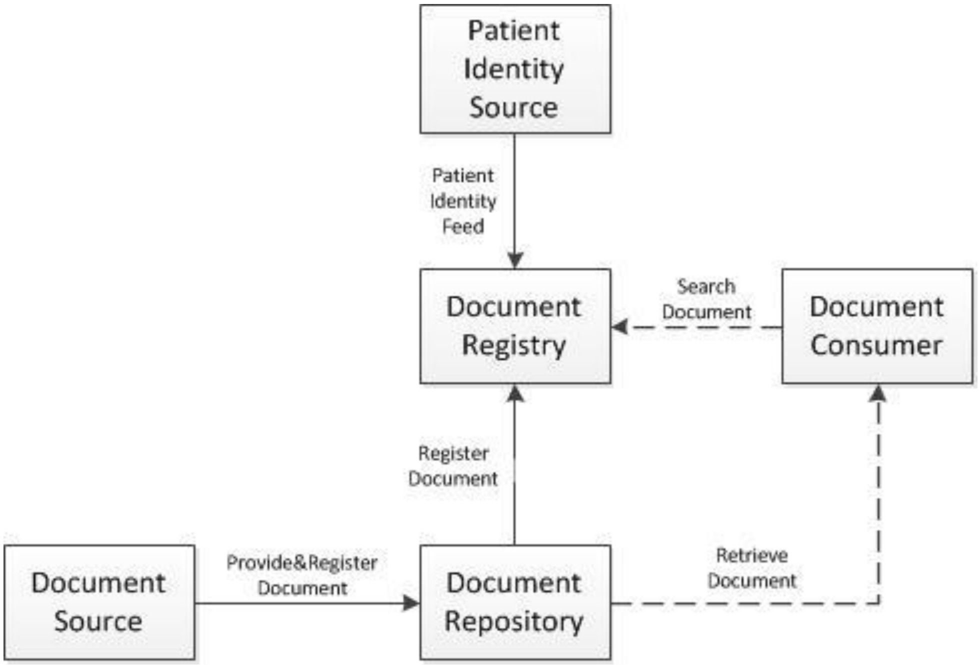


Figure 2. XDS.b Actors and Transactions

- ***Document Source*** acts as document producer and publisher. PACS modalities may implement an imaging Document Source to send images and/or corresponding report to a Document Repository. Document Source should also supply metadata to Document Repository for subsequent registration with Document Registry.
- ***Document Consumer*** searches Document Registry for documents; Document Registry returns the location of documents meeting query criteria; Document Consumer retrieves documents from Document Repository where matched documents stored. PACS workstation may implement a Document Consumer to search and retrieve patient images and reports.
- ***Document Registry*** maintains metadata (e.g., author information, source identity, patient information, date) about each registered document. Document Consumer might find interesting documents through query specification based on these metadata. Document Registry also includes a link to Document Repository where the document is stored. The link of matched documents is returned to Document Consumer.
- ***Document Repository*** is responsible for persistent storage of published documents. When Document Source provides document and corresponding metadata, Document Repository triggers registration at the meantime.
- ***Patient Identity Source*** provides a unique identifier for each patient and maintains patient identity mapping across affinity domains. It ensures the patient identifier is valid and understandable among actors.

1.5 IHE Security and Privacy Related Profiles

IHE has developed a number of profiles that address security and patient privacy requirements, including authentication, authorization and audit.

- ***Enterprise User Authentication Profile (EUA).*** This profile leverages the widely accepted network authentication protocol "Kerberos" to authenticate users inside a single enterprise. The authenticated user can use all devices and software that participate in the common network domain with single sign-on. Access from other enterprise domain or Internet is out of scope of this profile.
- ***Cross Enterprise User Assertion Integration Profile (XUA).*** In cross-enterprise environment, each enterprise may have independent user directory. This profile leverages Web-Security and SAML2.0 (Security Assertion Markup Language) to support identity federation. It provides a mechanism to exchange authenticated subject information across domain boundaries.
- ***Audit Trail and Node Authentication Profile (ATNA).*** This profile requires bidirectional certificated-based authentication for connections between nodes both within enterprise and across enterprises. Audit logs are recorded in each local audit repository.
- ***Basic Patient Privacy Consents Integration Profile (BPPC).*** This profile provides a way to express allowances or restrictions on patient's healthcare data access relying on the patient's own wishes.
- ***Access Control White Paper.*** Depending on above integration profiles, IHE produces a white paper on the topic of access controls, proposing a trusted model where each local

domain is responsible for ensuring that personal health information is adequately protected.

1.6 OpenID Connect

"OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner". OpenID connect has attracted significant attention, which even won 2012 European Identity and Cloud Award for best innovation and new standard, standing out in federated identity. OpenID Connect is based upon JSON/REST standards; spans both traditional computing platforms and mobile devices; and has broad support from major cloud service providers, enterprise companies, and social networking companies (e.g., Google, Yahoo, Microsoft, Facebook). OpenID Connect, combing OpenID and OAuth together, is one of the most perspective open standards to potentially become the de-facto standard for securing cloud computing and mobile applications, which has even been regarded as "Kerberos of Cloud".

2. Proposed Architecture

In the industry world of PACS, the state of the art of authentication and authorization provision can be viewed as: i) anyone who can enter the lab and logon to the workstation computer is allowed to do anything; ii) after the client application entity (IP, port, AE Title) is added to the PACS server's trusted list, any user logged on as client application can access images stored at the server site. Application entry is the identity between parties. PACS server just knows which client application is talking to, but has no knowledge about the exact user; and iii) IHE proposes that the client has to get a “service ticket” after being authenticated to access the service. However, Kerberozed DICOM has been proposed and is under development, but not finalized yet. Due to the above security weaknesses of the existing PACS systems, we designed and implemented an agent-based solution to imply single sign-on user authentication from any PACS, and enhanced fine-grained access control on user level rather than application level.

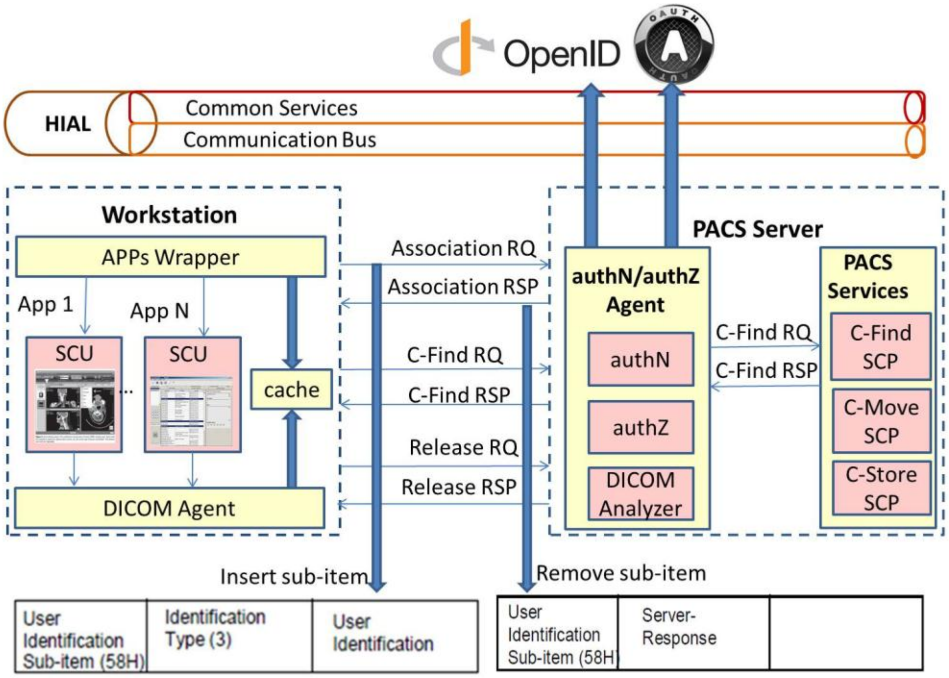


Figure 3. Agent-based architecture in legacy PACS systems

In DICOM lingo, a basic DICOM operation has two actors: service class user (SCU) and service class provider (SCP). Figure 3 is an episode of SCU-SCP model, which presents the process of querying DICOM images. The process contains three stages: i) association establishment between SCU and SCP, in which SCU sends a C-Find request with search criteria; ii) SCP responds by sending images that match the query; and iii) association release when no further processing is required. Figure 3 also illustrates the agent-based architecture in one PACS system, and demonstrates how PACS authentication and authorization workflow are consolidated against HIAL (Health Information Access Layer). The major subsystems of this architecture are *Workstation* and *PACS Server*. The workstation consists of the following components:

- ***Apps Wrapper.*** It is the only portal of all integrated PACS client applications at one workstation or modality. Any existing application (App 1...App N) can be launched by Apps Wrapper and run independently. Before launching the target application, Apps Wrapper prompts to input user name and password. The user information is persisted in local cache until user logout from target application. The workflow of existing applications remains unchanged.
- ***DICOM Agent.*** It captures all outgoing DICOM messages. If the message is an association establishment request (Association-RQ), DICOM Agent retrieves the corresponding user information from cache and inserts user identity sub-item into Association-RQ message. User identity sub-item supports three methods: user name in plain text (type 1); user name plus passcode (type 2); and Kerberos service ticket (type 3). In turn, if incoming message is an association response (Association-RSP), DICOM Agent has to remove user identity sub-item from DICOM message and log audit according to the reply result. In this way, both client application (App 1...App N) and PACS services benefit from extended user identity negotiation without any change in their workflows.

The PACS Server consists of the following components:

- ***authN/authZ Agent.*** This agent monitors all incoming DICOM messages. After receiving Association-RQ message, DICOM Analyzer locates the user identity sub-item and extracts user name and password (credentials). DICOM Analyzer also looks into the DICOM data dictionary to find attributes related to user action such as user operation type, target image ID and patient ID. Then the component authenticates PACS user against OpenID protocol, using identified user name and password. The *authZ* component checks if access request is allowed by OAuth authorization server using extracted user action properties. Consent directive polices and action-based access control policies are integrated with OAuth. Our previous work explained in detail the flow of OpenID authentication and access control process using OAuth authorization protocol.
- ***PACS Services.*** If the access is granted, the request C-Find-RQ is sent to the real PACS service (C-Find SCP); the PACS searches in the local image database and returns back the matched images to the SCU. The workflow of SCP remains unchanged. C-Move and C-Store are other sample service providers to retrieve or store images.

3. Experimental Procedure

Figure 4 illustrates the physical architecture for integration of vendor-independent PACS systems with DI-r (abbreviated image-enabled EHR system). It also presents how to authenticate and authorize PACS users against common services provided by HIAL no matter whether accessing images stored inside PACS or stored in DI-r. The major components are discussed below.

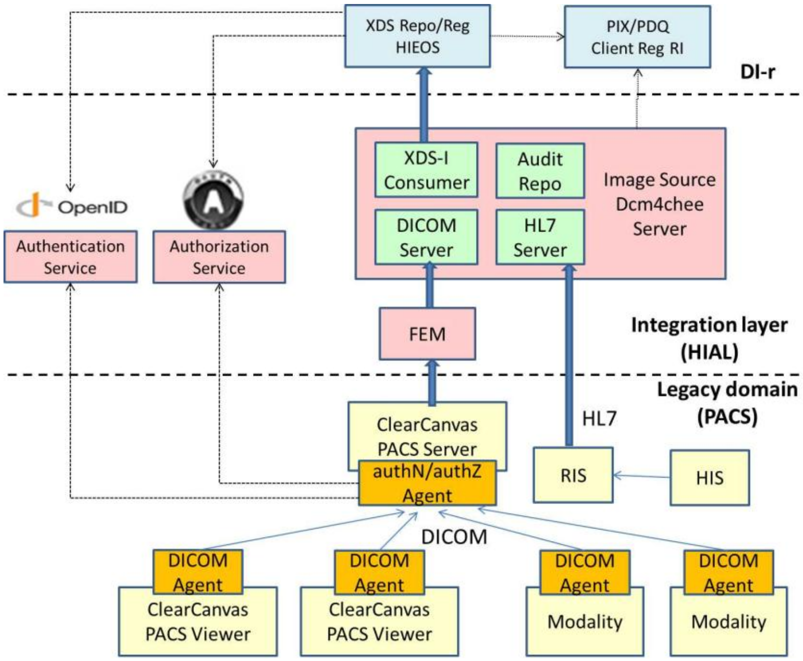


Figure 4. Integrated physical architecture based on open source tools

- ***ClearCanvas PACS.*** is an open source DICOM and PACS/RIS informatics and extensible platform, which includes viewing, archiving, management, workflow and distribution of images. A distributed ClearCanvas DICOM viewer and a DICOM server are deployed to simulate an existing PACS system in hospitals. Also DICOM agents are deployed on each workstation to capture outgoing and incoming DICOM messages. One agent is deployed in front of DICOM Server assisting in integration with OpenID and OAuth. Other commercial PACS systems such as Agfa and GE will be tested in future. *RIS* and *HIS* in this physical architecture are place-holders for future research.
- ***HIEOS XDS Repository and Registry***: Health Information Exchange Open Source (HIEOS) is an implementation of, primarily server-side, IHE Cross Enterprise Document Sharing (XDS.b) document-registry and document-repository services. We employ HIEOS to simulate an abbreviated image-enabled EHR system, which provides web service interfaces to retrieve and store images.
- ***Client Registry RI.*** Client Registry Reference Implementation supports standard based interfaces including IHE Patient Identifier Cross-Reference HL7 v3 (PIXV3) and Patient Demographic Query HL7 v3 (PDQV3). PIXV3 provides cross-referencing of patient identifiers from multiple domains. PDQV3 provides services to query patient information according to user defined search criteria. Both XDS Repository/Registry services and image source depends on this component for patient identifier mapping and information query.
- ***FEM and Image Source (Dcm4chee).*** To enable ingestion of foreign exams, an integration component FEM (Foreign Exam Management) needs to take responsibility for "localizing" data on behalf of the PACS system, such as assisting in forwarding DICOM C-FIND or C-MOVE requests to DI-r, and morphing DICOM tags to ensure that images can be accepted by the local PACS system seamlessly. Dcm4chee is a collection of open source applications and utilities for the healthcare enterprise, and the core is a robust implementation of the DICOM standard. In our project, dcm4chee acts as an integration component which receives images forwarded by FEM, and then trigger a workflow to send DICOM Manifest (KOS file) to XDS Repository and Registry.
- ***Authentication and Authorization Services.*** HIAL (Health Information Access Layer) provides common services which are responsible for: i) authentication of PACS users based on established identity of the PACS user; and ii) making access control decisions for the image accessing request using action tuples extracted by agents.

3.1 How to capture DICOM messages?

Wireshark is based on WinPcap (Windows), which allows capturing, analyzing and even modifying network packages bypassing the protocol stack, but unable to block or redirect packages to a specified destination (e.g. another application or host). Besides, we can't capture on the local loopback address 127.0.0.1 with a Windows packet capture driver like WinPcap. Wireshark is developed based on WinPcap, explained why we cannot capture any message of dcm4chee. RawCap is a free command line network sniffer for Windows that uses raw sockets. We can use RawCap to capture localhost network traffic in Windows. Our agent is expected to modify and redirect packages so that we decided to develop it based on the source code of DCM4CHEE rather than depending on package capturing tools.

3.2 Build up development environment

We have already built up the development environment of DCM4CHE v2.0.28. Following are the steps of integration of Eclipse and Maven for DCM4CHEE compiling and debugging environment:

- Install Maven Eclipse plugin from this site:
  http://download.eclipse.org/technology/m2e/releases
  Click Eclipse help->click Install new software->add the URL->install
- Import dcm4che as Maven projects to Eclipse
  Click Eclipse File -> Import -> Maven -> Existing Maven projects -> ...
- You can see some build errors. Maybe Maven is not well integrated with Eclipse. It does not matter. Just click it and select Quick Fix.
- Now there is no build error and compling is done. Go to your dcm4che installation folder and modify the file "run.bat". For example, my path is "dcm4chee-2.17.3-psql\bin\run.bat". Uncomment the debug option as below:
  change "rem set JAVA_OPTS=-Xdebug -Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=y %JAVA_OPTS%" to "set JAVA_OPTS=-Xdebug -Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=y %JAVA_OPTS%"
- Start dcm4che through running "dcm4chee-2.17.3-psql\bin\run.bat" and monitor the printed logs. You will see "Listening for transport dt_socket at address: 8787" and dcm4che is blocked to start since it's waiting for Eclipse to attach.
- Go to Eclipse. Click run -> debug configuration -> Remote Java application -> add -> host is "localhost", port is "8787" and connection type is "socket attach"
- After remote debug is applied from Eclipse, dcm4che service is triggered to resume.

3.3 UML Package Diagram

Figure 5 is the package diagram of our designated adapter (agent).

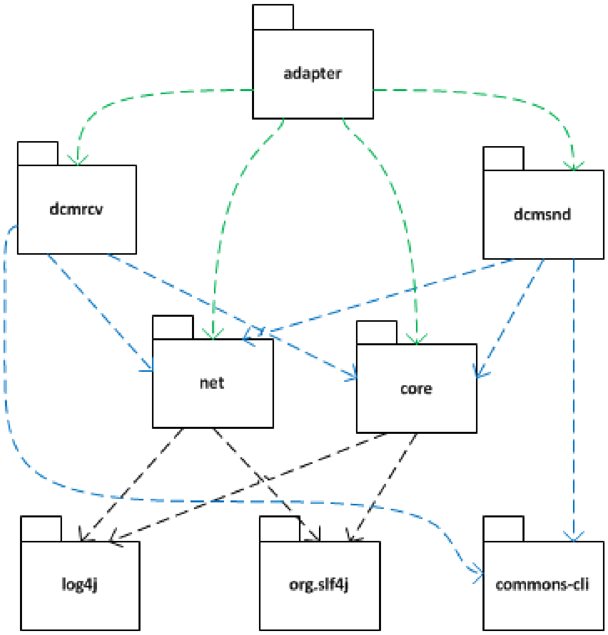
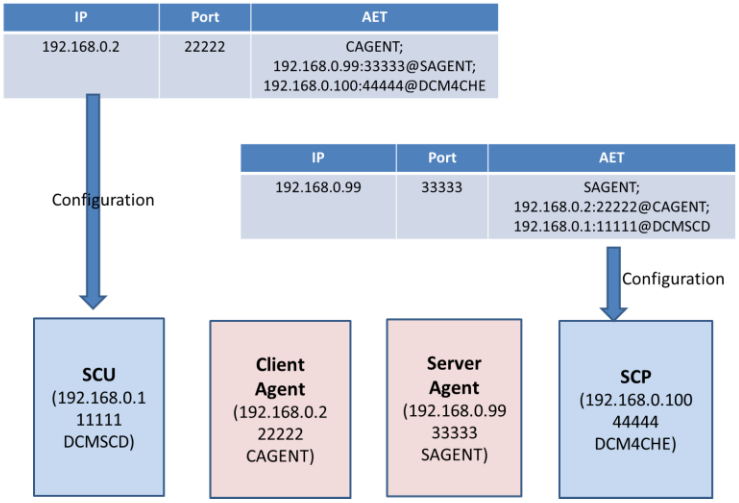


Figure 5. Adapter UML Package based on DCM4CHE Source Code

3.4 How to solve routing problem?

In DICOM network, SCU and SCP are identified by a tuple <IP, port, AE Title>. After inserting client agent and server agent into DICOM network, how to establish connection between <SCP, client agent>, <client agent, server agent> and <server agent, SCP>?

Our potential solution is that the routing between agents, SCU, SCP is solved by the wrapper of AE Title. AE Title is an arbitrary string that represents a DICOM application. To solve the routing problem, modify AE Title to be "self-routing-explained" as Figure 6 and Figure 7.



IP	Port	AET

192.168.0.2	22222	CAGENT;
		192.168.0.99:33333@SAGENT;
		192.168.0.100:44444@DCM4CHE

IP	Port	AET

192.168.0.99	33333	SAGENT;
		192.168.0.2:22222@CAGENT;
		192.168.0.1:11111@DCMSCD

Figure 6. AET configuration

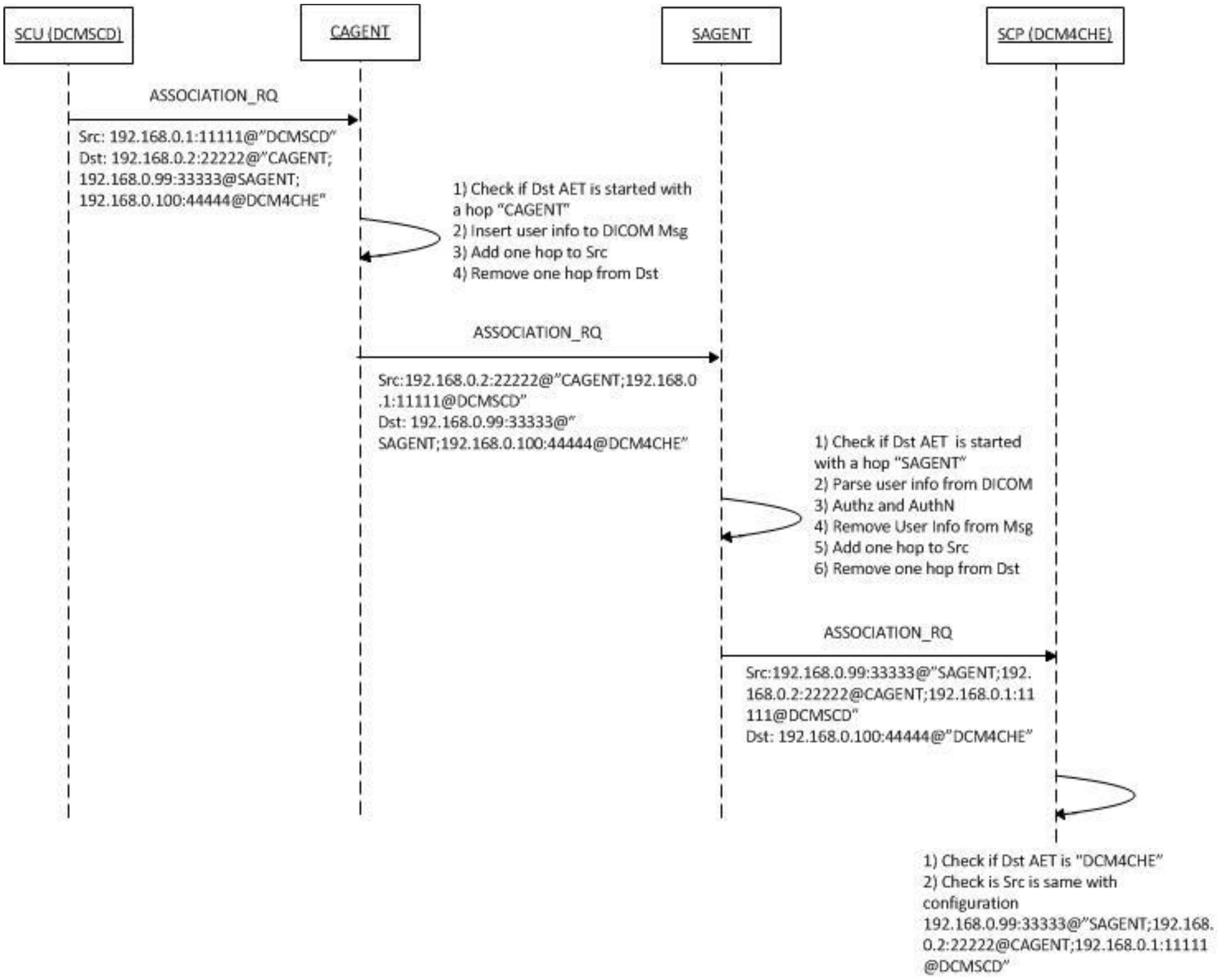

SCU (DCMSCD)
CAGENT
SAGENT
SCP (DCM4CHE)
ASSOCIATION_RQ
Src: 192.168.0.1:11111@"DCMSCD"
Dst: 192.168.0.2:22222@"CAGENT;
192.168.0.99:33333@SAGENT;
192.168.0.100:44444@DCM4CHE"
1) Check if Dst AET is started with
a hop "CAGENT"
2) Insert user info to DICOM Msg
3) Add one hop to Src
4) Remove one hop from Dst
ASSOCIATION_RQ
Src:192.168.0.2:22222@"CAGENT;192.168.0
.1:11111@DCMSCD"
Dst: 192.168.0.99:33333@"
SAGENT;192.168.0.100:44444@DCM4CHE"
1) Check if Dst AET is started
with a hop "SAGENT"
2) Parse user info from DICOM
3) Authz and AuthN
4) Remove User Info from Msg
5) Add one hop to Src
6) Remove one hop from Dst
ASSOCIATION_RQ
Src:192.168.0.99:33333@"SAGENT;192.
168.0.2:22222@CAGENT;192.168.0.1:11
111@DCMSCD"
Dst: 192.168.0.100:44444@"DCM4CHE"
1) Check if Dst AET is "DCM4CHE"
2) Check is Src is same with
configuration
192.168.0.99:33333@"SAGENT;192.168.
0.2:22222@CAGENT;192.168.0.1:11111
@DCMSCD"

Figure 7. The workflow of establishing association between SCU and SCP