DICOMWeb, what is it and how can it be used?

You might have heard the term DICOMWeb and wondered what it is and how it might impact your imaging environment. This white paper serves both as a high level introduction to DICOMWeb and its functionality, and explains how it can be used.

DICOMWeb is a new version of the DICOM protocol. It specifies a web-based service for accessing and presenting DICOM objects (e.g., images, measurements, and, to some degree, medical imaging reports). It is primarily intended for distribution of results and images to healthcare professionals, it is not meant to replace how modalities such as CT, MR, digital X-ray and other acquisition devices connect to a PACS as this is based on the existing, mature, traditional DICOM communication standard.

DICOMWeb provides a simple mechanism for accessing a DICOM object using commonly used web protocols, i.e. a HTTP/HTTPS protocol. Note that it does not support facilities for broad web searching of DICOM images, it does however allow querying a distinctly defined database e.g. a PACS, VNA (Vendor Neutral Archive), or image registry such as a private or public Health Information Exchange (HIE) for imaging studies and related patient information.

DICOMWeb can not only provide access to the native DICOM image information but it also allows you to request for the object provider to convert the data in a presentation-ready format such as a JPEG representation of the DICOM image. However, one should note that DICOMWeb relates, operates, and refers only to native DICOM Objects, i.e. NOT to objects, which are stored natively in a non-DICOM format such as JPEG's, PDF documents and others.

The protocol that DICOMWeb uses is not new; on the contrary, it is widely used by many companies providing web access such as Amazon, Google, and many others. Here lies also a major advantage, as the traditional DICOM protocol is still somewhat of a niche protocol which is only used for medical and to a lesser degree for industrial imaging. Consequently, DICOMweb offers access to a lot of more tools, expertise, and resources, that are many magnitudes greater than what we have available in the medical imaging domain. In addition, we get another freebie; web services that are very well suited for use on mobile devices such as tablets, phablets, phones, etc.

DICOMWeb allows for easy "mashing" of different applications on a single screen. For example, an Electronic Medical Record (EMR) can launch a resource for displaying an image using a simple call defined by the DICOMWeb API (Application Programming Interface) next to a lab result or other information retrieved using Web services.

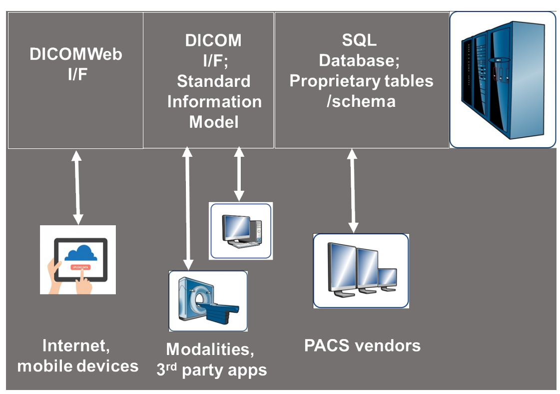
As an analogy, consider how Google Maps has become the standard way for most websites to create directions to their business, e.g. for a restaurant or hotel. One click on a little map links you to the large map with directions. We can do a similar transaction by retrieving a medical image from a PACS database, a simple click translating into a web address will provide the image embedded into an EMR display.

Yes we still would need a DICOM viewer if we would have retrieved the native DICOM format, but launching it will become so much easier. And, if the provider is asked to render the pixel data in a standard, consumer media format, such as JPEG or TIFF, or convert it to an MPEG in the case of a dynamic multi-frame acquisition such as used for cardiology or heart (echocardiography) studies, it can easily be displayed in any standard web browser. There is also the capability to convert a DICOM Structured Report to a text, pdf or html, again, very simple to display, without needing a special DICOM viewer.

DICOMWeb does not change the actual DICOM data formats that are commonly referred to as the “DICOM header” or metadata, or how a DICOM database is accessed. It only changes the protocol, i.e. the transport mechanism used for the information access. The information is still encoded exactly the same way as if it would have been accessed using the traditional DICOM protocol. It is similar to changing your mail carrier from the post office mail to FEDEX. The same information is exchanged, only the packaging, i.e. envelope, drop-off mailbox, and handling process is different. In addition, we have different addressing for accessing the source of the information, which makes getting a study of a patient as simple as specifying a web address.

With DICOMWeb, a simple uniform resource identifier (URI) defining the source is sufficient in contrast to the traditional DICOM protocol, where we need to use a special port number that has to be configured in a computer using a fixed IP address, which is an issue by itself as most of the world today uses dynamic IP addressing, and a DICOM Application level address, which is referred to as an AE-Title (Application Entity Title). The DICOMWeb protocol provides the resource address, we use the standard web access port, and the routing is taken care of by the internet addressing infrastructure. No need for an elaborate IP address, port number and AE-Title addressing anymore.

There is more good news. There is direct mapping from the traditional DICOM protocol to the DICOMWeb protocol. Using the same analogy between the Post Office and FEDEX, it is possible for FEDEX to packetize the mail and use the Post Office to send it to its distribution center, and from there use FEDEX vans to deliver the FEDEX envelope. It might not seem that efficient, but remember that it is just software and adding another processing layer is not necessarily that difficult, even though it is probably more efficient to go direct. Therefore, adding DICOMWeb as an software extra layer to an existing PACS software interface is not difficult.
If you are interested in how the information is actually retrieved, it is shown in the illustration. The DICOM objects that include images, waveforms, or documents, are encapsulated in a DICOM wrapper, and are managed by a commercial database such as Oracle, Sybase, MySQL, SQL server, and many others. Most of these databases contain tables that can be accessed using and ANSI standard query language called SQL.



The problem here is that you need to know the database table structure, which in many cases is proprietary vendor information in order to do a query. This is not a problem if a vendor talks from its own workstation to its PACS, but it is an issue for an “outsider”, e.g. if you want to use a workstation from a different vendor, or want to retrieve images from the PACS from let’s say a CT or MR for comparison. Even if the vendor shares its so-called database scheme, it is different for each PACS vendor, so you would need to write a different query for each PACS system to see a worklist of studies that are stored in your PACS. This is where the DICOM standard plays an important role as it defines a standard access information model consisting of a Patient, Study, Series and Image with search keys that are required so that the information can be accessed and returned to the workstation or modality in a standard worklist format. So, a SQL “SELECT…FROM…WHERE…” query against the proprietary database tables is created by mapping it into a standard DICOM FIND Request, which uses the Patient-Study-Series-Image hierarchy to search for the patient and study records to display its worklist. This is how the traditional DICOM works.

To create the web-based interface, the next step is simply converting the web standard so-called (GET, POST, etc.) commands. The output data is formatted using either XML or a Java script format called JSON. You can imagine how easy it would be to develop an interface as may JAVA developers know how to write software for their specific applications using these common scripting languages. It opens up the world of medical imaging to a whole new world of apps that can be embedded on your phone or other mobile media.

With regard to the retrieval of information, the DICOM committee also took the opportunity to make a significant change in how the information can be retrieved. The traditional protocol considers a DICOM object such as an image as a self-contained blob of binary data with an accompanying metadata, aka the header. There was no way to pick this object apart.

DICOMWeb allows the metadata to be retrieved by itself without the actual pixel data, or only the pixel data by itself, aka “bulk data.” Going back to our mail analogy, instead of getting let’s say multiple envelopes that all need to be opened to find out the details of the entire information being exchanged, it allows for these letters to be put in one big box, which is referred to as “bulk transfer” and be able to interpret its content by looking at the shipping manifest (metadata), which can be retrieved separately.

So-far-so-good, however, you might be concerned about security, as right now, most medical applications are locked down in the department of radiology using a closed network, which is protected by firewalls, intrusion detecting software and other malicious software protection. If a physician needs access to images over a public network, it is often provided using a so-called VPN or Virtual Private Network, which creates a secure “tunnel” through the public internet infrastructure allowing encrypted exchanges of information in the same manner as ecommerce allows buying airline tickets or any online purchases using credit card payments.

Which brings me to the point – the same security mechanism used to purchase let’s say an airline ticket or new book on-line, i.e. user and password authentication as well as secure encryption of your credit card number, is used to access the information using DICOMWeb. Yes, buying a ticket for a concert on-line might not be HIPAA compliant, but the additional safeguards such as audit trails and consent management are taken care of by accompanying standards and detailed in the so-called ITI (IT Infrastructure) section of the IHE (Integrating the Healthcare Enterprise) profile specifications.

With regard to the traditional DICOM protocol, yes there are software development toolkits available, but it is not like any of the web protocols. The synchronization between two devices that use the traditional DICOM protocol has a non-trivial overhead as any communication establishment includes three steps: it requires a synchronization between the two devices about the type and encoding of the information to be exchanged, and after a confirmation, the actual exchange, and then a release negotiation again. So three steps instead of one, all using a DICOM toolkit. Imagine as an analogy, a project manager (DICOM device) wants to exchange information. He or she calls the secretary (toolkit), who calls the destination to make sure they are able to receive the information to be sent. Using the information in their directory (port number, IP address, AE-Title), the destination replies back confirming or rejecting, if confirmed, exchanges the information (sends image) and follows up with another call (releases the connection). Instead, in the analogy of the DICOMWeb, the manager can just drop off the info in a public mailbox and be all set, i.e. one instead of three steps.

Now let’s dive a little bit deeper into the actual DICOMWeb standard as it is important to know what version of DICOMWeb you might be needing or looking for. To make things somewhat complicated, there are three different versions of the protocol. Why three? Well, this has to do with how the technology evolved. It started out with a simple implementation in 2003 but with limited functionality, followed by applications using web services in 2011, and recently in 2014 we have defined the RESTfull (Representational State Transfer) services which is how most of the internet is tied together as of today. See the table for an overview and comparison of these three versions.

Protocol name	"traditional" DICOM	WADO-URI	WADO-WS	WADO-RS	QIDO-RS, STOW-RS, UPS-RS, capabilities-RS
standardized	1993	2003	2011	2014	2014
use case	remote DICOM viewer	web server/browser access	web svcs image and metadata access	web svcs and mobile access	web svcs and mobile access
protocol	"DICOM" protocol	URI based	Web services	RESTfull svcs	RESTfull svcs
format of return parameters (media type)	DICOM	DICOM, rendered, text, subset, compressed, anonymized	As URI but also meta data	As URI but also metadata, bulk data	query, store, worklist
addressing	IP, port, AE-Title	standard port, uri/url	standard port, uri/url, community ID	standard port, uri/url, community ID	standard port, uri/url, community ID
encapsulation	DICOM meta-header	part 10 for DICOM	SOAP (XML)	JSON (Java script)	JSON (Java script)
characteristic	overhead, semi-proprietary	compact but limited	verbose	suitable for web access	suitable for web access

In the first column, you'll see the characteristics of the traditional DICOM protocol, it dates from 1993, is used when you use a DICOM viewer accessing a PACS or VNA database with the three-step protocol as described above. You can only get DICOM formatted images or related information back, and it uses the traditional IP, port number and AE-Title addressing. The encoding uses the DICOM meta-header in front of the pixel data. So, its characteristic is that it has definitely protocol overhead and it is specialized, or you might even call it semi-proprietary, i.e. used in the imaging domain only.

The second column shows the first version of the DICOMWeb, WADO-URI, which is a simple URI or resource locator that can be used to retrieve an object in the original DICOM or a rendered (e.g. JPEG) format. It uses an http GET command to retrieve the information. However, the software application needs to know exactly where the object is located, e.g. in a VNA, and, needs to know its Unique Identifier or UID to retrieve the information. A typical application could be to retrieve an image from a PACS when clicking in an EMR on an icon image, which translates the request for e.g. a JPEG rendered version of that image. The Option parameters (aka media type) to specify the image retrieval are quite extensive, i.e. you can instruct the server to anonymize the image, specify annotations on the image to be provided such as for the patient name, modify its size by specifying the number of rows and columns, instruct the region of the image to be exchanged, apply a specific window width and level to be applied, select a frame number in case of a multi-frame image and apply compression. The encapsulation of the DICOM object is done as a so-called Part-10 file, referring to part 10 of the DICOM standard, which describes an additional meta-header specifying the source, creator and encoding used for the file. This protocol is compact, but its applications were somewhat limited and it never got any big traction.

To get an idea of how a WADO-URI call looks like, here is an example:

Retrieving a region of a DICOM image, converted if possible in JPEG2000, with annotations burned into the image containing the patient name and technical information, and mapped into a defined image size:

```
https://aspradio/imageaccess.js?requestType=WADO
&studyUID=1.2.250.1.59.40211.12345678.678910
&seriesUID=1.2.250.1.59.40211.789001276.14556172.67789
&objectUID=1.2.250.1.59.40211.2678810.87991027.899772.2
&contentType=image%2Fjp2;level=1,image%2Fjpeg;q=0.5
&annotation=patient,technique
&columns=400
&rows=300
&region=0.3,0.4,0.5,0.5
&windowCenter=-1000
&windowWidth=2500
```

The second version of DICOMWeb is the web services implementation, or WADO-WS. This protocol is based on the functionality provided by WADO-URI, i.e. it has the same capabilities with regard to selecting the various options, but extends it to provide retrievals from registries such as defined by the IHE XDS (Cross-Enterprise Document Exchange) profile.

Unlike the WADO-URI where the requester would specify a unique identifier (UID) to retrieve an image or object, its addressing allows for the specification of a unique repository ID to locate the data. It is basically the imaging counterpart of the IHE XDS document exchange. Another difference with the WADO-URI is the encoding using XML for the metadata for the returned objects.

A typical use case would be if an EMR would want to display in JPEG format one image per series with information describing every series (e.g. series description). In this case, the EMR sends to the DICOM server the list of the objects (“selection”), asking for the object content. The DICOM server sends back the JPEG images corresponding to the listed objects, and the EMR sends to the DICOM server the “selection” information for obtaining the relevant information about the objects retrieved and the DICOM server sends back the corresponding information in the form of a “metadata” document, converted in XML.

WADO-WS uses the SOAP protocol (Simple Object Access Protocol). The SOAP encapsulation using XML is rather popular and widely used by many internet service providers. One of the disadvantages of SOAP is that this protocol is rather verbose, the same request as listed above using the URI would result in a transaction that is about five times bigger in size. One should note that the WADO-WS was very useful for exchanging images across different enterprises in the context of the IHE XDS activity, but that the focus of the standards committee has been to expand the next version instead of expanding and improving the WADO-WS.

The third option for DICOMWeb, WADO-RS is using RESTful services. Again, with regard to its functionality, it is very similar to the WADO-URI and WADO-WS, in that it can retrieve an image with annotations, windowed, scaled, compressed, etc. But, with regard to the retrieval of single and multi-frame image and rendered data and metadata, the standard added the option to retrieve the so-called bulk data, i.e. pixel data without the typical header or metadata. It also can be retrieved in an uncompressed or compressed format.

The RS protocol has been extended to provide not only a retrieve (WADO-RS), but also a store (STOW-RS), and Query (QIDO-RS). It added a worklist service modeled after the traditional so-called DICOM UPS or Unified Worklist and Procedure Step, and the feature to retrieve the capabilities of a certain resource, i.e. what are the services and details of those services that it can provide.

Adding the worklist could potentially be the breakthrough feature that makes this useful. Consider the following scenarios where this is important. A dermatologist wants to take a picture with a phone of a skin rash and upload that to an enterprise image management system such as a VNA. Identifying the image with the proper patient demographic data coming from a single source of truth is critical to allow it to be matched with the patient folder, the EMR, etc. The worklist feature would allow this information to be retrieved. The same applies for an ER physician who wants to take a picture of an injury or a surgeon who wants to make regular pictures of how a wound is healing. A worklist can also be used to coordinate remote readers, i.e. radiologists who are reading from different sites, or specialists who need to provide a second opinion.

The major difference between the WADO-WS and WADO-RS is that the Restful (RS) protocol is more compact as it uses JSON rather than the verbose SOAP protocol, which uses XML. RESTful services are similar to the HL7 FHIR standard (Fast Healthcare Interoperability Resources). Therefore, when considering a platform or application such as an EMR, it makes sense to use a single protocol to access different resources, whether it is a lab result or a physician directory using FHIR or a set of images using WADO-RS.

In conclusion, what is it that you need to know about DICOMWeb?

- First of all, you need to know what is its application, i.e. what are the typical use cases, such as for image distribution over the web, NOT how a typical modality (CT, MR, etc.) exchanges worklists with an information system or stores its images on a PACS.
- Second, you also should be aware that there are three different versions, each one building on each other and expanding its functionality, i.e. from a URI to web services (WS) to Restful (RS) implementation. Similar to the traditional DICOM services, these are NOT compatible with each other, so you need to check the conformance statements to match between the clients and servers.

Using DICOMWeb could be a major advantage for certain use cases as it provides a simple, straightforward interface, which is easy to create and leverages many existing resources, tools, and experiences. In addition, it is consistent with the development in the healthcare IT space where FHIR is being implemented as well. Again, it is NOT a replacement of how modalities and PACS systems currently communicate, which is mature and robust, as the last thing we want to do is to expand the IOT (Internet of Things) with diagnostic devices that contain sensitive information.

However, a final word of caution is warranted. It is new, which brings certain risks with regard to integration, availability and knowledge base among providers. Therefore, do your homework before you take the plunge into DICOMWeb.

This whitepaper is created by Herman Oosterwijk (www.otechimg.com) a seasoned healthcare imaging and IT trainer/consultant through an educational grant from Vital Images, a Toshiba corporation (www.vitalimages.com)