

# ***DICOMweb, IHE WIA und WIC***



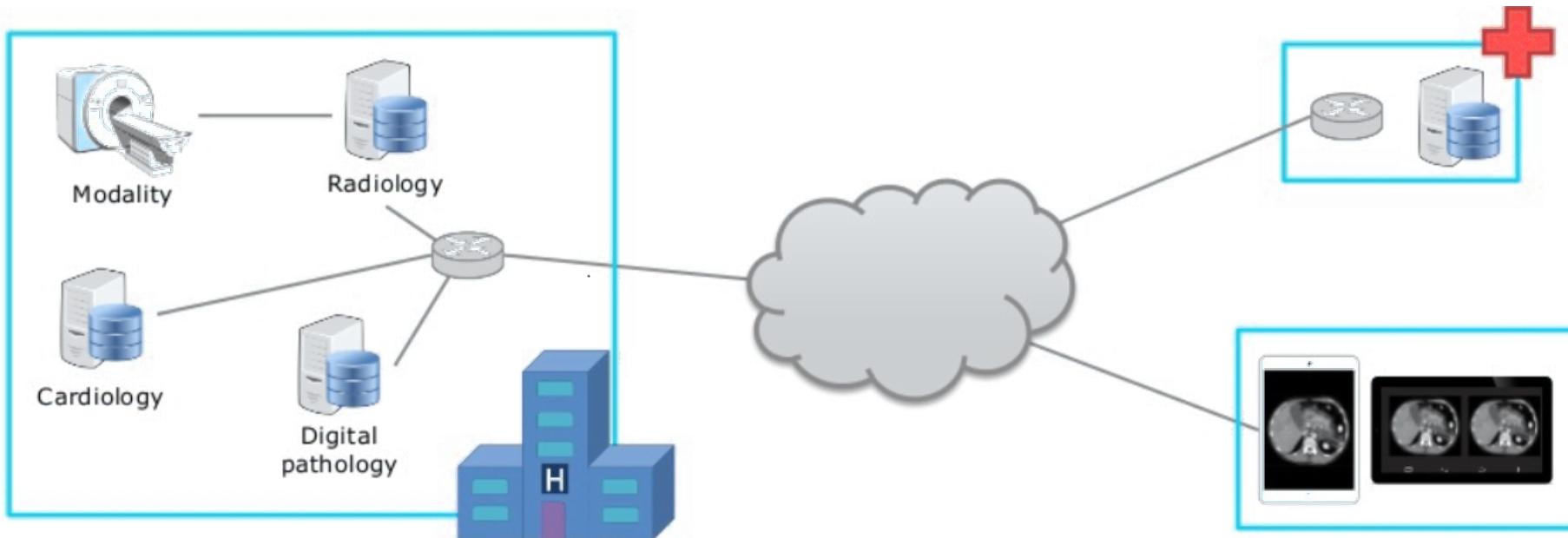
- 9:00 – 10:00 Silvia Winkler: **DICOM Web Services**
- 30 min Pause
- 10:30 – 11:30 Emmanuel Helm: **IHE WIA und WIC**
- 11:30 – 12:30 Gunter Zeilinger: **DICOMweb Worklist Services**

# ***DICOM Web Services***

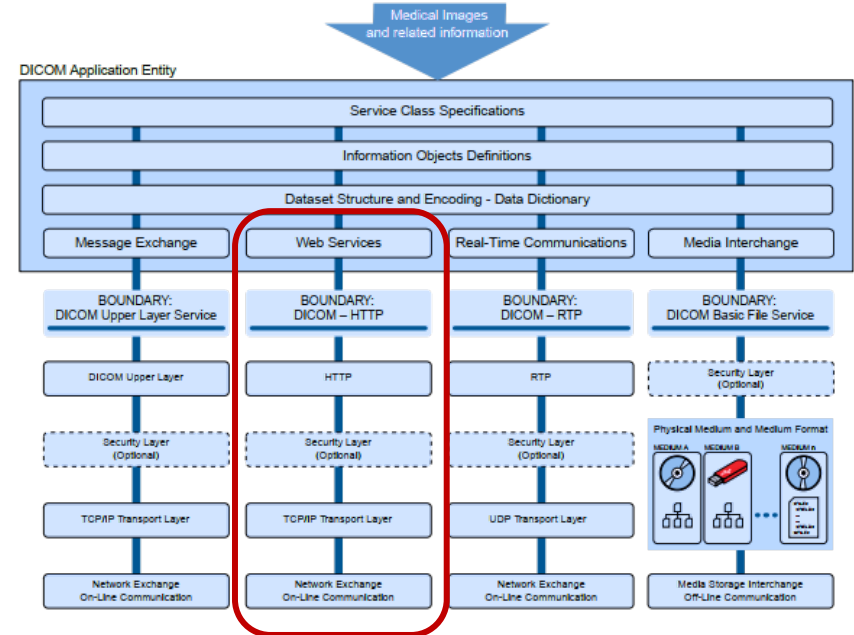


- Einführung und Übersicht
- URI Web Service
- WADO-WS
- DICOMweb
  - Grundlagen: REST / DICOM Ressourcen
  - Studies Services
    - Retrieve (WADO-RS)
    - Store (STOW-RS)
    - Search (QIDO-RS)
  - Datenrepräsentation

# DICOM Web Services – Einführung und Übersicht



## DICOM Web Services





DIMSE / DUL

URI Web Service

[WADO-WS]

Restful  
Webservices

- 2003 **WADO** (Web Access to DICOM Objectes) (heute: „URI Web Service“)
- 2011 Erweiterung um **WADO-WS** (SOAP Webservices)
- 2011 Erweiterung auf Restful Services: **WADO-RS, STOW-RS, QIDO-RS,**
- 2015 Restful Rendering
- 2017 Retirement von WADO-WS
- 2019 Grundlegende Überarbeitung der Spezifikation

Spezifikation: DICOM [PS3.18](#)  
(zuständig: DICOM [WG-27](#))

## URI Web Service (WADO-URI)

- http-GET
- Standard URI Syntax mit Parametern in Form von name=„value“ Paaren
- Mime types (u.a.):
  - application/dicom
  - image/jpeg (default mime type für Bilder)
  - video/mpeg (opt. für multiframe)
- Bildweiser Zugriff



**Focus: mobile Geräte / Browseranwendungen**

**dann, wenn reduzierte Qualität ausreicht**



## URI Web Service (WADO-URI)

- ❑ Für jedes Bild muss die gesamte Hierarchie: StudyInstanceUID, SeriesInstanceUID, SOPInstanceUID verpflichtend angegeben werden
- ❑ Optionale Parameter:
  - eine gewünschte Transfersyntax (bei application/dicom)  
(JPIP → streaming)
  - Anonymisierung
  - Retrieve von Bildausschnitten
  - Fensterung
  - frameNumber
  - und viele mehr
- ❑ Nachteile:
  - Bildweiser Zugriff
  - Der Client bekommt keine Metadaten

## URI Web Service (WADO-URI) – Beispiele

### ❑ Laden eines Bildes als image/jpeg

```
http://www.hospital123/radiology/wado.php?requestType=WADO
&studyUID=1.2.250.1.59.40211.1234.5678
&seriesUID=1.2.250.1.59.40211.1234.5678.1
&objectUID=1.2.250.1.59.40211.1234.5678.1.123
```

### ❑ Laden eines SR als text/html

```
http://server234/script678.asp?requestType=WADO
&studyUID=1.2.250.1.59.40211.1234.5678
&seriesUID=1.2.250.1.59.40211.1234.5678.2
&objectUID=1.2.250.1.59.40211.1234.5678.2.456
&charset=UTF-8
```

### ❑ Retrieve als DICOM Objekt

```
http://www.medical-webservice.st/RetrieveDocument?requestType=WADO
&studyUID=1.2.250.1.59.40211.1234.5678
&seriesUID=1.2.250.1.59.40211.1234.5678.1
&objectUID=1.2.250.1.59.40211.1234.5678.123
&contentType=application%2Fdicom
&anonymize=yes
&transferSyntax=1.2.840.10008.1.2.4.50
```

## WADO-WS

- ❑ SOAP basiert (http-POST)
- ❑ definierte wsdl / xsd Schema
- ❑ drei Methoden:
  - RetrievalmagingDocumentSet(→ IHE RAD XDS-I.b)
  - RetrieveRenderedImagingDocumentSet
  - RetrievalmagingDocumentSetMetaData
- ❑ Vorteile gegenüber WADO-URI
  - Zugriff auf die Metadaten möglich
  - Zugriff auf ganze Document Sets, z.B. alle Bilder einer Studie
  - Einbettung von Authentifizierungstokens in den soap-header
- ❑ Seit 2017 im DICOM Standard RETIRED, die Pflege dieses Protokolls wurde an IHE delegiert. ([Blog dazu](#))

## WADO-WS

- RetrievalImagingDocumentSet
  - application/dicom
  
- RetrieveRenderedImagingDocumentSet
  - image/jpeg, (opt.) image/gif, video/mpeg für multiframe Objekte
  - optionale Parameter für
    - die gewünschte Transfersyntax
    - Anonymisierung
    - Retrieve von Bildausschnitten
    - Fensterung
    - frameNumber
    - und viele mehr
  
- RetrievalImagingDocumentSetMetaData
  - „Metadata Only“: Laden der Headerdaten oder Teilen davon OHNE Pixeldaten

## DICOM Web Services

- URI Web Service

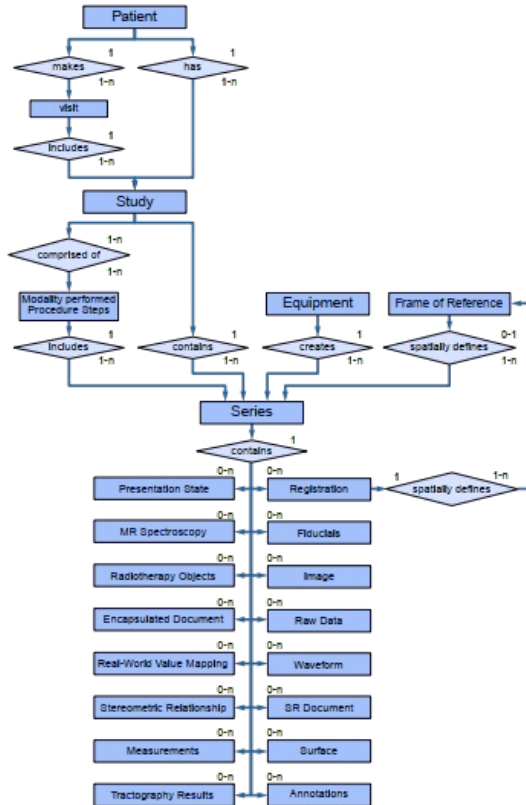
- Restful Web Services

Spezifikation abhängig von den zugeordneten Ressourcen

- Studies Service
- Worklist Service
- Non-Patient Instance Service
- Storage Commitment Service (neu seit **2024a**)

## REST – Architektur (REpresentational State Transfer)

- **Addressierbarkeit:**  
jede Resource ist über einen eindeutigen Identifier (Unique Resource Identifier URI) identifizierbar.
- Komplette **stateless** → Keine Sessions
- Einheitliche Schnittstellen: z.B. **http GET, PUT, POST**
- Entkopplung von Ressource und Repräsentation (der Requestor legt fest, in welcher Form er die Daten bekommen möchte: z.B. XML, JSON)
- Vier notwendige Aktionen: **Create – Read – Update –Delete (CRUD)**
- Client = User Agent, Server = Origin Server



## Ressource: DICOM Information Objects („Collections“)

- /studies            eine Auswahl von Studien
- /series            eine Auswahl von Serien
- /instances            eine Auswahl von Instances

Weiters als Ressourcen definiert

- /metadata            nur die Metadaten
- /rendered            gerenderte Daten
- /thumbnails            Vorschaubilder
- /bulkdata            nur die Pixeldaten
- ... ([PS3.18 Tabelle 10.1-1](#))

## Transaktionen des Studies Service

## PS3.18 Kapitel 10

Transaktion	Methode	Beschreibung	alte Bezeichnung	entspricht dem DIMSE Service
Retrieve	GET	Anfordern einer oder mehrerer Ausprägungen von Studies Ressourcen	WADO-RS	C-MOVE oder C-GET
Store	POST	Speichert eine oder mehrere Ausprägungen von Studies Ressourcen	STOW-RS	C-STORE
Search	GET	Suche nach Ressourcen, die den Suchkriterien entsprechen	QIDO-RS	C-FIND



## Beispiele für Retrieve Transaktionen (WADO-RS) (http GET):

Ressource	URI - Template	
Study	/studies/{study}	Laden von Bildern einer Studie (Study UID)
Series	/studies/{study}/series/{series}	Laden von Bildern (Study + Series UID)
Instance	/studies/{study}/series/{series}/instances/{instance}	Laden eines Bildes (Study + Series + SOP InstUID)
Frames	/studies/{study}/series/{series}/instance/{instance}/frames/{frames}	Study + Series + SOP UID + Frame Nummern
Bulkdata	{bulkdata}	Laden der Daten (URI)
Series Metadata	/studies/{study}/series/{series}/metadata	Laden der Metadaten einer Serie
Rendered Instance	/studies/{study}/series/{series}/instances/{instance}/rendered	Laden einer gerenderten Instance
Study Thumbnail	/studies/{study}/thumbnail	Laden von Vorschaubildern für Studien

## Beispiele aus [PS3.18 Annex B](#)

### □ Retrieve Instance as rendered JPEG

```
GET
/radiology/studies/1.2.250.1.59.40211.1234.5678/series/1.2.250.1.59.40211.123
4.5678.1/instances/1.2.250.1.59.40211.1234.5678.1.123/rendered HTTP/1.1
Host: www.hospital123
Accept: image/jpeg

HTTP/1.1 200 OK
Content-Length: 79323
Content-Type: image/jpeg

<BINARY JPEG DATA>
```

- Retrieve Instance gerendert, als jpeg2000, Annotierungen (Pat., Tech.) eingebrannt, spez. Viewport und Windowing

```
GET
/radiology/studies/1.2.250.1.59.40211.1234.5678/series/1.2.250.1.59.40211.123
4.5678.1/instances/1.2.250.1.59.40211.1234.5678.1.123/rendere
d?annotation=patient,technique&viewport=1024,1024,0,0,512,512
&window=200,600,linear HTTP/1.1
Host: www.hospital123
Accept: image/jp2;level=1,image/jpeg;q=0.5

HTTP/1.1 200 OK
Content-Length: 56734
Content-Type: image/jp2

<BINARY J2K DATA>
```

## □ Retrieve DICOM SR as HTML

```
GET
/radiology/studies/1.2.250.1.59.40211.1234.5678/series/1.2.250.1.59.40211.123
4.5678.2/instances/1.2.250.1.59.40211.1234.5678.2.456/rendered?charset=UTF-8
HTTP/1.1
Host: www.hospital123
Accept: text/html

HTTP/1.1 200 OK
Content-Length: 4087
Content-Type: text/html

<HTML DATA>
```

## □ Retrieve Instance als DICOM

```
GET
/radiology/studies/1.2.250.1.59.40211.1234.5678/series/1.2.250.1.59.40211.1234.5678.1
/instances/1.2.250.1.59.40211.1234.5678.123 HTTP/1.1
Host: www.hospital123
Accept: multipart/related; type="application/dicom"

HTTP/1.1 200 OK
Content-Length: 383203
Content-Type: multipart/related; boundary="DICOM DATA BOUNDARY";
type="application/dicom"
...

--DICOM DATA BOUNDARY
Content-Type: application/dicom

<BINARY DICOM DATA>
--DICOM DATA BOUNDARY
```

## Beispiele für Store Transaktionen (STOW-RS) (http POST):

- ❑ Mehrere SOP Instances in einem Request müssen nicht zwingend zur gleichen Studie gehören.
- ❑ Wurde eine StudyInstanceUID im Request angegeben, werden nicht entsprechende SOP Instances rejected.

Ressource	URI - Template	
Studies	/studies	Speichern von Instances, die zu verschiedenen Studien gehören
Study	/studies/{study}	Speichern von Instances einer Study

## □ Request:

- multipart/related; application/dicom : PS3.10 binary Instances  
(Ein Nachrichtenblock pro DICOM Instance)
- multipart/related; application/dicom+xml: DICOM XML Metadaten + Bulkdata
- multipart/related; application/dicom+json: DICOM JSON Metadata + Bulkdata

## □ Response:

- XML oder JSON Objekt:  
http Status Code + Informationen zu den einzelnen SOP Instances im Request,  
WADO-RS Retrieve URL

## □ Speichern mehrerer DICOM Instances

```
POST /radiology/studies/1.2.250.1.59.40211.1234.5678 HTTP/1.1
```

```
Host: www.hospital123
```

```
Content-Type: multipart/related; type="application/dicom"; boundary=MESSAGEBOUNDARY
```

```
--MESSAGEBOUNDARY
```

```
Content-Type: application/dicom
```

```
< BINARY DICOM DATA for instance 1 >
```

```
--MESSAGEBOUNDARY
```

```
Content-Type: application/dicom
```

```
< BINARY DICOM DATA for instance 2 >
```

```
...
```

```
--MESSAGEBOUNDARY
```



## □ Antwort auf des Speichern mehrerer DICOM Instances

HTTP/1.1 200 OK

Content-Length: 826

Content-Type: application/dicom+json; charset=utf-8

...

```
[{
  "00020010": {"vr": "UI", "Value": ["1.2.840.10008.1.2.1"]},
  "00081199": {"vr": "SQ", "Value": [
    {"00080016": {"vr": "UI", "Value": ["1.2.840.10008.5.1.4.1.1.4"]},
    {"00080018": {"vr": "UI", "Value": ["1.3.12.2.1107.5.99.3.30000012059" ]}
  ]},
  ...}
]
```

## Beispiele für Search Transaktionen (QIDO-RS) (http GET {service}/[?query])

- Der Request enthält alle Information als Parameter in der URL, keine Request-Payload
- Attribute werden entweder mit dem Tag („ggggeeee“) referenziert oder mit dem Keyword

URL	
/studies	Alle Studien
/studies?PatientID=11235813	Alle Studien eines Patienten
/studies?PatientID=11235813&StudyDate=20130509-20130510	Alle Studien eines Patienten mit DateRange
/studies?PatientID=11235813&includefield=00081048&includefield=00081049&includefield=00081060	Erweiterter Antwort-Datensatz

## □ Serververhalten:

Suchanfragen müssen **idempotent** sein: zwei aufeinanderfolgende gleiche Abfragen müssen die gleiche geordnete Liste von Ergebnissen liefern, wenn der Datenstand am Server unverändert ist.

## □ Antworten:

### ■ Formate:

- multipart/related; type=„application/dicom+xml“ (für jedes Ergebnis einen Antwort-Teil)
- Default: „application/dicom+json“ (alle Ergebnisse in einem Antwort-Teil: single part)

### ■ http-Status 204: No Matches

### ■ http-Status 200: OK + Result-Set

### ■ Fehler:

- http-Status 400: Bad Request
- http-Status 413: Payload Too Large  
(max.responses (server-seitig oder client-seitig) überschritten)

## □ Suchen nach Studien eines bestimmten Tages, max. Responses, Antwort JSON (die ersten 20 Responses)

```
GET /radiology/studies?00080020=20000817&limit=20&offset=0&includefield=all HTTP/1.1
Host: www.hospital123
Accept: application/dicom+json
```

```
HTTP/1.1 200 OK
Content-Length: 1191
Content-Type: application/dicom+json; charset=utf-8
```

```
[
  {
    "00080005": {"vr": "CS", "Value": ["ISO_IR192"]},
    ...
    "00081190": {"vr": "UR", "Value":
["https://www.hospital123/radiology/studies/1.2.250.1.59.40211.1234.5678"]}
  }
  ...
]
```

- Worklist Service (UPS-RS)
- Non-Patient Instance Service
- Storage Commitment Service

## XML und JSON Repräsentation

### □ XML

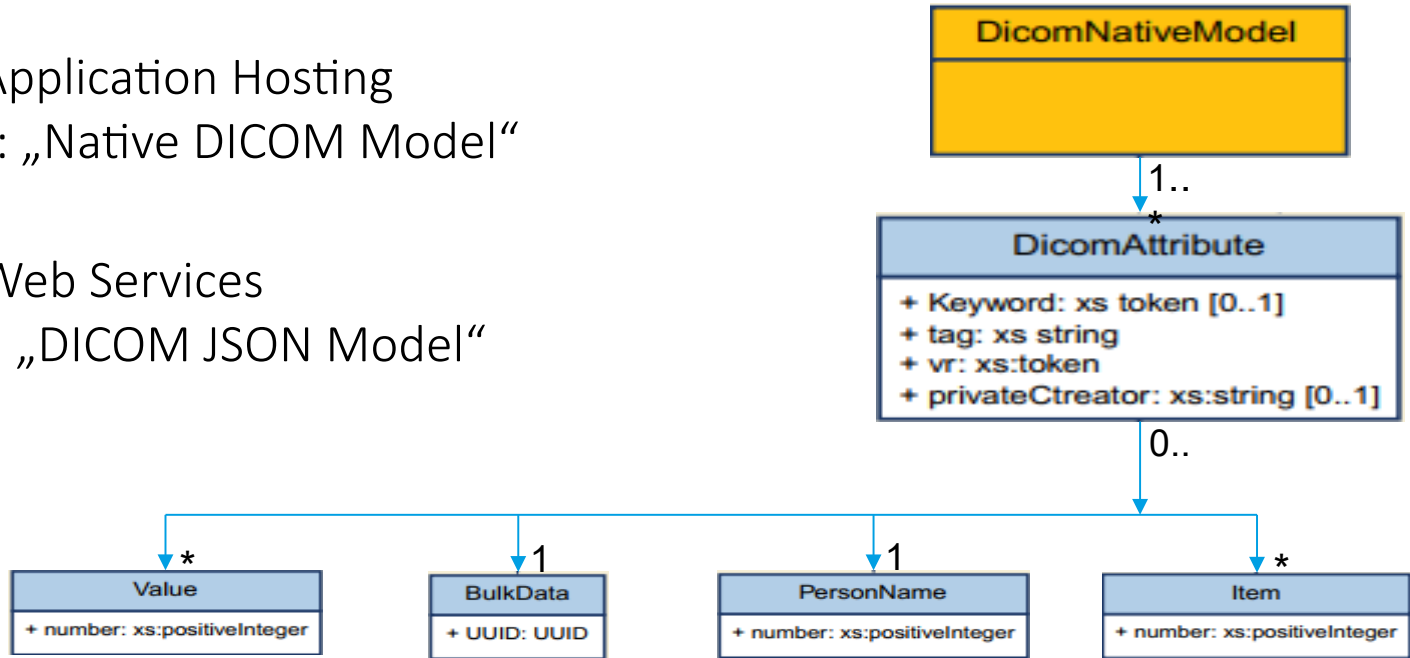
[PS3.19](#) Application Hosting

Annex A: „Native DICOM Model“

### □ JSON

[PS3.18](#) Web Services

Annex F: „DICOM JSON Model“



## Das Native DICOM Model ist die formale Abbildung eines DICOM Datasets auf XML

Abbildung eines DICOM Attributs `<DicomAttribute>` mit diesen Parametern

- tag = Tag: „ggggeeee“
- vr als Ausdruck des Datentyps
- `<Value>...</Value>`  
nummeriert mit number-Parameter  
`<Value number=„1“>...</Value>`  
`<Value number=„2“>... </Value>`
- `<Item>` für die Abbildung von Sequence-Items
- `<PersonName>` für die Abbildung von Attributen mit VR PN
- `<BulkData>`
- `<InlineBinary>`

## DICOM JSON Model ist die JSON Umsetzung des Native DICOM Models

- Abbildung eines DICOM Attributs wie folgt:
  - Name = Tag: „gggggeeee“
  - Child objects:
    - vr als Ausdruck des Datentyps
    - Value oder BulkDataURI oder InlineBinary
  - Attribute mit VM > 1 → [...]
  - Abbildung von leeren Attributen (Type 2 ohne Wert)
  - Abbildung der DICOM VRs auf JSON Datentypen



- DICOM Standard
  - Überblick: <https://www.dicomstandard.org/using/dicomweb>
  - [Cheatsheet](#)
  - PS3.18 Web Services <http://dicom.nema.org/medical/dicom/current/output/pdf/part18.pdf>
  
- DICOMweb bei der JT 2023
  - [DICOMweb und IHE WIA/WIC \(HL7/DICOM Austria Jahrestagung 2023\) \(dcm4che.github.io\)](#)
  - [DICOMweb \(dcm4che.github.io\)](#)
  - [Swagger UI](#)