

Utworzenie infrastruktury PACS i DICOM-Web dla małej jednostki medycznej

Mirosław Błazej, Michał Dygas - opiekun projektu - dr inż. Łukasz Czekierda

Spis treści

[Spis treści](#)

[Orthanc](#)

[Przykładowa infrastruktura](#)

[Podstawowa konfiguracja serwera](#)

[Dostęp za pomocą interfejsu webowego](#)

[Połączenie przez protokół DICOM](#)

[Wysyłanie obrazów medycznych](#)

[Desktopowe przeglądarki DICOM](#)

[Konfiguracja klienta](#)

[Storage](#)

[Dicom WEB](#)

[Wnioski](#)

[Bibliografia](#)

Orthanc

Orthanc (<https://www.orthanc-server.com/>) jest darmowym serwerem pełniącym rolę archiwum obrazów medycznych (PACS). Wspiera on standard DICOM zarówno jako format przechowywanych danych, jak i protokół umożliwiający komunikację z serwerem. Został napisany w języku C++, a jego kod źródłowy jest publicznie dostępny (na zasadach otwartego oprogramowania).

Orthanc przechowuje dane w dwóch lokalizacjach:

- do przechowywania plików DICOM używany jest lokalny system plików
- pozostałe dane Orthanc przechowuje w bazie danych - domyślnie SQLite.

Orthanc jest rozwiązaniem modułowym. Możliwości serwera Orthanc można łatwo rozszerzać z użyciem plug-inów - napisanych w Lua bądź C++.

Przykładowa infrastruktura

W celu przetestowania możliwości, które daje serwer Orthanc, przygotowaliśmy prostą, przykładową infrastrukturę. Deployment serwera Orthanc w opisany poniżej sposób powinien być wystarczający dla małej jednostki medycznej.

Zaproponowana przez nas infrastruktura mieści się na jednej maszynie. W związku z tym, jest ona dość prosta i nie wymaga zastosowania zaawansowanych orkiestratorów takich jak Kubernetes. Wykorzystaliśmy **docker-compose**, zaś sam serwer Orthanc uruchamiamy wykorzystując obraz **jodogne/orthanc-plugins**, zawierający zarówno serwer Orthanc, jak i kilka najpopularniejszych plug-inów (niektóre z nich wykorzystujemy w ramach naszego projektu).

Jako bazę danych wykorzystujemy osobną instancję bazy danych PostgreSQL. Domyślnie nie jest wspierana ona przez Orthanc, jednak dostępny jest plug-in (domyślnie zawarty w używanym przez nas obrazie Docker'a), umożliwiający jej wykorzystanie.

```

version: '3.9'
services:
  orthanc-server:
    image: jodogne/orthanc-plugins
    container_name: orthanc-server
    volumes:
      - ./orthanc/orthanc.json:/etc/orthanc/orthanc.json:ro
    depends_on:
      - postgres
    ports:
      - 4242:4242
      - 8042:8042
    restart: on-failure

  postgres:
    image: postgres:12.5
    container_name: postgres
    volumes:
      - postgres-data:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: orthanc
      POSTGRES_DB: orthanc
      POSTGRES_PASSWORD: orthanc
      POSTGRES_HOST_AUTH_METHOD: trust
    ports:
      - 5432:5432
    restart: on-failure

  dicom-client:
    build:
      context: ./sender
    volumes:
      - ./sender/dicoms:/dicoms
    depends_on:
      - orthanc-server

volumes:
  postgres-data:

```

docker-compose.yml

Serwer orthanc eksponuje dwa porty:

- **4242** - protokół DICOM
- **8042** - dostęp przez przeglądarkę.

W przypadku Postgres'a, tworzymy użytkownika-administratora bazy. W przypadku prawdziwego wdrożenia, hasło użytkownika powinno być przechowywane w

bezpieczniejszej formie (np. za pomocą mechanizmu sekretów).

Podstawowa konfiguracja serwera

Konfiguracja serwera orthanc jest przechowywana w lokalnym pliku orthanc/orthanc.json, zmapowanym do konfiguracji serwera Orthanc jako wolumin Docker'a. Jako bazę wykorzystaliśmy domyślną konfigurację, wprowadzając do niej kilka zmian:

- skonfigurowaliśmy połączenie z PostgreSQL:

```
"PostgreSQL": {  
  "EnableIndex": true,  
  "EnableStorage": true,  
  "Host": "postgres",  
  "Port": 5432,  
  "Database": "orthanc",  
  "Username": "orthanc",  
  "Password": "orthanc"  
}
```

- zmieniliśmy użytkowników serwera. W Orthanc jedynym sposobem na zarządzanie użytkownikami jest modyfikacja konfiguracji serwera. Stworzyliśmy przykładowych użytkowników mdygas i mblazej:

```
"RegisteredUsers": {  
  "mdygas": "mdygas",  
  "mblazej": "mblazej"  
},
```

Uwaga: Orthanc umożliwia skonfigurowanie protokołu TLS dla HTTP (HTTPS) i DICOM w celu bezpiecznej transmisji danych. Jest to bardzo zalecane w przypadku produkcyjnego zastosowania - w przeciwnym wypadku dane (w tym dane logowania i dane medyczne!) przesyłane są plaintext'em przez sieć. Inną opcją jest wykorzystanie reverse proxy (np. nginx).

Dostęp za pomocą interfejsu webowego

Serwer Orthanc udostępnia interfejs webowy, umożliwiający na prosty dostęp do plików DICOM dostępnych na serwerze. Po autentykacji (za pomocą HTTP Basic-Auth) mamy dostęp do wyszukiwarki obrazów medycznych znajdujących się na serwerze. Możemy również otworzyć listę wszystkich pacjentów bądź wszystkich obrazów:

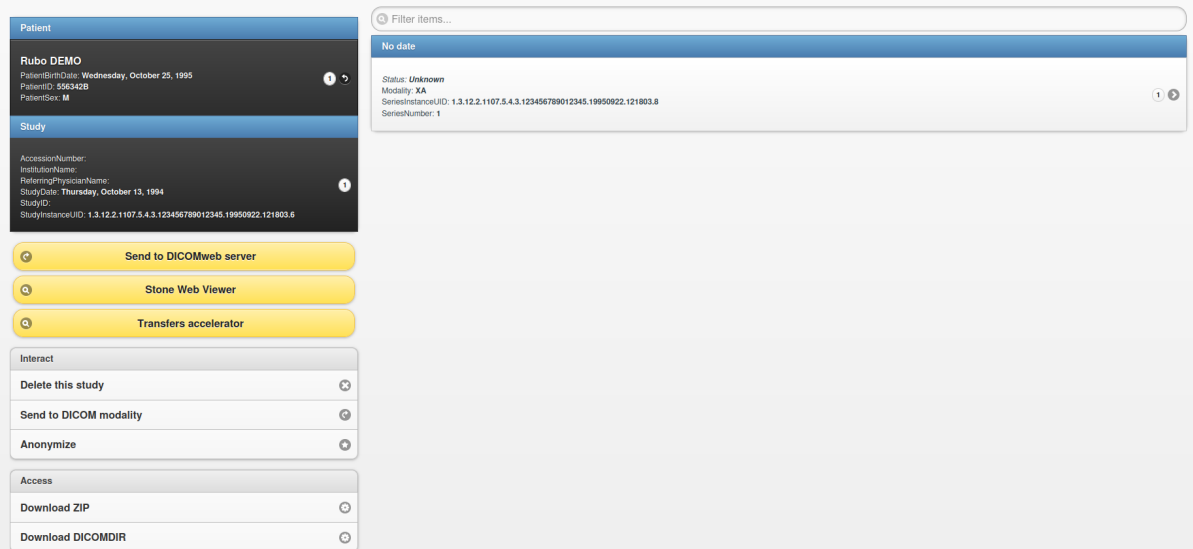


ORT_HANC

Patient ID:
 Patient Name:
 Accession Number:
 Study Description:
 Study Date: Any date

Wyszukiwarka

Serwer Orthanc dzieli przechowywane obrazy medyczne po pacjentach (patient) i badaniach (studies). Do jednego pacjenta może być przypisane kilka badań, a do jednego badania kilkoro pacjentów:



Patient

Rubo DEMO

PatientBirthDate: Wednesday, October 23, 1995
 PatientID: 556342B
 PatientSex: M

Study

AccessionNumber:
 InstitutionName:
 ReferringPhysicianName:
 StudyDate: Thursday, October 13, 1994
 StudyID:
 StudyInstanceUID: 1.3.12.2.1.107.5.4.3.123456789012345.19950922.121803.6

Interact

Access

Filter items...

No date

Status: Unknown
 Modality: XA
 SeriesInstanceUID: 1.3.12.2.1.107.5.4.3.123456789012345.19950922.121803.8
 SeriesNumber: 1

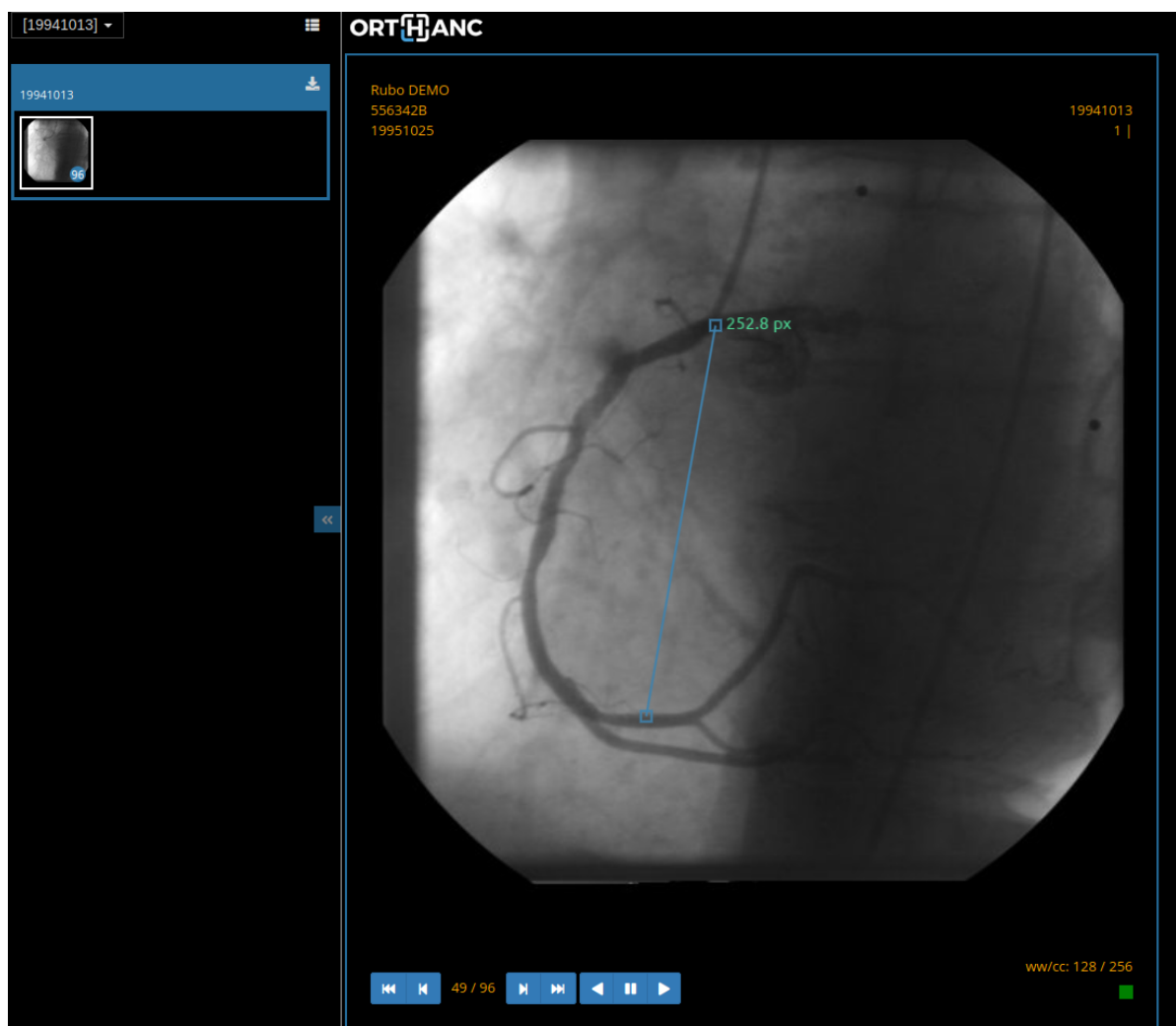
Lista obrazów medycznych z danego badania.

Orthanc pozwala na:

- zarządzanie danymi badania (usunięcie/wysłanie do innego serwera DICOM/anonimizacja)
- pobranie danych jako ZIP
- przeglądanie obrazów medycznych za pomocą prostej przeglądarki (Orthanc Web Viewer) oraz bardziej zaawansowanej Stone Web Viewer dołączonej jako plug-in:



Orthanc Web Viewer



Stone Web Viewer

Obydwie umożliwiają manipulację oknem (za pomocą kursora). Pobierają ona pojedyncze klatki (jeśli obraz medyczny zawiera kilka, pobiera ona kolejne klatki w kolejnych request'ach). Klatki pobierane są od razu w pełni głębi kolorów - manipulacja oknem odbywa się bez konieczności pobrania dodatkowych danych. Stone Web Viewer umożliwia dodatkowo łatwe przełączanie między różnymi obrazami w ramach jednego badania.

Połączenie przez protokół DICOM

Wysyłanie obrazów medycznych

Docelowo, w przypadku jednostki medycznej obrazy medyczne będą wysyłane przez urządzenia obrazowania medycznego wspierające protokół DICOM. W celu zasymulowania takiego transferu (oraz zaznajomienia się z narzędziami związanymi z protokołem DICOM) stworzyliśmy skrypt, który symuluje takie urządzenie. Wykorzystaliśmy do tego celu **dcmstk**.

Skrypt uruchamiany jest z pomocą Dockera i orkiestrowany z użyciem docker-compose (`dicom-client`). Dockerfile obrazu, na którym uruchamiamy dcmstk, znajduje się poniżej:

```
FROM ubuntu:20.04
RUN apt-get update && apt-get -y upgrade && apt-get -y clean
RUN apt-get install -y dcmstk

COPY docker-entrypoint-local.sh /docker-entrypoint-local.sh
ENTRYPOINT ["/docker-entrypoint-local.sh"]
```

Dockerfile

Dodatkowo, stworzyliśmy prosty skrypt wysyłający (z użyciem narzędzia dcmsend będącego częścią dcmstk) przykładowy, znaleziony w Internecie obraz medyczny

```
#!/bin/bash

set -o pipefail
set -o errexit
log(){
  MESSAGE=${1}
  echo "[DICOM-SENDER] ${MESSAGE}"
}

sleep 10

dcmsend orthanc-server 4242 /dicoms/0002.DCM
log "image has been sent"
```

docker-entrypoint-local.sh

Po pomyślnym przesłaniu, obraz wraz z metadanymi znalazł się na serwerze. Na podstawie metadanych obraz został przypisany do odpowiedniego pacjenta i badania.

Desktopowe przeglądarki DICOM

orthanc-server działając w trybie serwera PACS umożliwia także integrację z aplikacjami desktopowymi. Funkcjonalność ta jest szczególnie istotna, ponieważ umożliwia dostęp do dokumentacji medycznych w trybie offline - wszystkie dokumenty są przechowywane po stronie klienta.

W ramach ćwiczenia wykonaliśmy przykładową konfigurację, korzystając z [Radiant Viewer](#):

W związku z polityką prywatności, wymagane jest whitelistowanie IP klienta w pliku konfiguracyjnym **orthanc/orthanc.json**:

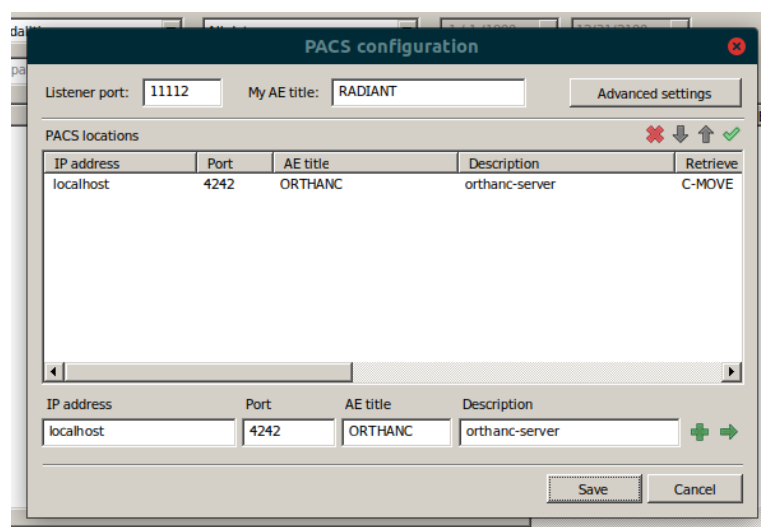
```
"DicomModalities": {  
  "RadiAnt": [  
    "RADIANT",  
    "172.23.0.1",  
    11112  
  ]  
}
```

W przypadku nieautoryzowanego dostępu, serwer odmawia dostępu do danych:

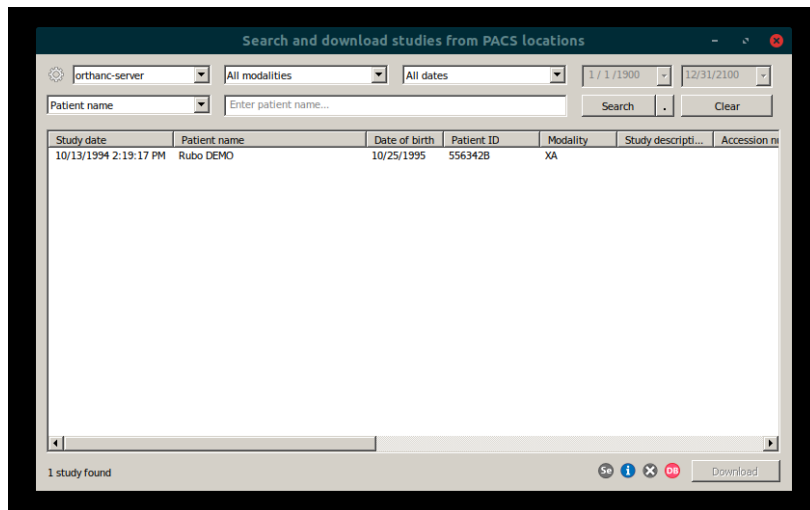
```
orthanc-server | W0629 21:56:22.513856 CommandDispatcher.cpp:817] Rejected Find request from remote DICOM modality with AET "RADIANT" and hostname "172.23.0.1"  
orthanc-server | W0629 22:00:35.744171 main.cpp:384] Unable to check DICOM authorization for AET RADIANT on IP 172.23.0.1: This AET is not listed in configuration option "DicomModalities"
```

Konfiguracja klienta

Odnotować należy że dane klienta są zgodne z **DicomModalities** po stronie serwera



Pobieranie plików z serwera:



Weryfikacja, czy pobrany został odpowiedni obraz:



Storage

RadiAnt przechowuje wszystkie pliki po stronie klienta, pod następującą ścieżką:

```
$ ls
RadiAnt.rdvdata RadiAnt.rdvdb RadiAnt.rdvdb-shm RadiAnt.rdvdb-wal
mdygas ~/Do.../RadiAntDB
$ pwd
/home/mdygas/Documents/RadiAntDB
```

Dicom WEB

DICOMweb jest to rodzina protokołów typu RESTful DICOM services stworzona na potrzeby ustandaryzowania operacji zapisu, odczytu bądź listowania obrazów medycznych poprzez sieć.

Argumentem za implementacją tego protokołu było stworzenie lekkiego mechanizmu, wspierającego odczyt dokumentacji na urządzeniach mobilnych czy w przeglądarce. Jego głównym celem jest wykorzystanie powszechnie używanych protokołów typu **http** czy formatu **JSON** w transmisji plików typu **DICOM**.

Implementuje on następujące funkcjonalności:

- **QIDO-RS** - odpytywanie o metadane kolekcji plików
- **WADO-RS** - pobieranie korpusu plików z serwera w formacie XML lub json
- **WADO-URI** - pobieranie konkretnego pliku z serwera
- **STOW-RS** - zapis do serwera

Orthanc-serwer w swojej domyślnej konfiguracji wykorzystuje te protokoły podczas udostępniania danych w przeglądarce.

Wnioski

- Zalety Orthanc'a:
 - darmowy
 - prosty w użyciu
 - dobra dokumentacja
 - łatwość wdrożenia (prosta konfiguracja, dostępne gotowe obrazy Docker'owe)
 - rozszerzalność za pomocą mechanizmu plug-in'ów
 - możliwość napisania własnego pluginu w Lua
 - wsparcie dla standardu DICOM umożliwiające łatwą integrację z innymi elementami ekosystemu, np. urządzeniami medycznymi, przeglądarkami obrazów medycznych czy innymi archiwami PACS
 - wsparcie (za pomocą plug-in'u) dla standardu DICOM-WEB
 - wsparcie dla różnych baz danych (SQLite, inne za pomocą gotowych plug-in'ów)
 - ciągłe wsparcie - ostatnia aktualizacja (1.9.4) wyszła 24 czerwca
 - otwarty kod źródłowy - Orthanc dostępny jest na licencji GPLv3; niektóre pluginy (np. web viewers) dostępne są na bardziej restrykcyjnej licencji AGPLv3
- Wady Orthanc'a
 - mocno uproszczony system kontroli dostępu do danych - każde konto ma dostęp do każdego zasobu dostępnego na serwerze. Możliwe rozwiązania:
 - odpowiednio skonfigurowane reverse proxy dające dostęp tylko do wybranych URL'i,

- napisanie plug-in'u.
- zarządzanie użytkownikami możliwe jest jedynie za pomocą pliku konfiguracyjnego. Możliwe rozwiązanie:
 - napisanie plug-in'u.
- obrazy medyczne są przechowywane za pomocą systemu plików; nie udało nam znaleźć plug-in'u, który umożliwiłoby korzystanie z innego rozwiązania (np. chmurowych static-file-storages takich jak Amazon S3 bądź rozwiązań z nim kompatybilnych. Możliwe rozwiązania:
 - zamontowanie innego systemu plików w katalogu, w którym Orthanc ma przechowywać obrazy
 - napisanie plug-in'u

Bibliografia

- dokumentacja Orthanc'a - <https://www.orthanc-server.com/static.php?page=documentation>
- Orthanc book - <https://book.orthanc-server.com/index.html>
- dokumentacja dcmsend (`man dcmsend`)