Uniwersytet Warszawski
Wydział Fizyki

Anna Ebinger
Nr albumu: 335697

Analiza standardów przesyłania danych medycznych na potrzeby zdalnego przetwarzania obrazów w pozytonowej tomografii emisyjnej

Praca magisterska
na kierunku Zastosowania fizyki w biologii i medycynie
specjalność Fizyka medyczna

Praca wykonana pod kierunkiem

dr hab. Macieja Kamińskiego
Zakład Fizyki Biomedycznej
Wydział Fizyki Uniwersytetu Warszawskiego

dr Konrada Klimaszewskiego
Centrum Informatyczne Świerk
Narodowe Centrum Badań Jądrowych

Warszawa, listopad 2018

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Streszczenie

Praca magisterska łączy ze sobą dwa obszary: fizykę medyczną oraz informatykę medyczną. Praca dyplomowa jest związana z projektem pynetdicom3, którego celem jest stworzenie implementacji sieciowych protokołów DICOM w Pythonie. Ponadto praca skupia się na stworzeniu odrębnego oprogramowania do zdalnego przetwarzania obrazów z tomografu PET.

Zaproponowane i przedyskutowane zostały dwa rozwiązania dotyczące schematu informatycznej infrastruktury szpitala. Dodatkowo demonstrator kodu napisany w pythonie 3 pokazuje zasadę działania przetwarzania danych medycznych pomiędzy klastrem obliczeniowych, skanerem PET, kontrolerem modalności, modality worklist a interfejsem użytkownika.

Słowa kluczowe

PET, DICOM, kryptografia, python, uwierzytelnianie i autoryzacja danych, zdalne przetwarzanie danych

Dziedzina pracy (kody wg programu Socrates-Erasmus)

13.2 Fizyka

Tytuł pracy w języku angielskim

Analysis of medical data transfer standards for remote image processing in positron emission tomography

Spis treści

Wykaz symboli i skrótów 4

Cel pracy 6

1. Wstęp 7
1.1. Podstawy fizyczne tomografii emisyjnej 7
1.1.1. PET we współczesnej medycynie 11
1.1.2. J-PET innowacja względem PET 12

2. Współczesne standardy w zdalnym przetwarzaniu i ochrony danych. . . 18
2.1. Zdalne przetwarzanie i przechowywanie danych 18
2.2. Standard DICOM 21
2.2.1. Opis standardu DICOM 21
2.2.2. Model IOD 23
2.2.3. Klasy SOP 25
2.2.4. Struktura danych 25
2.2.5. Model komunikacji DICOM 28
2.2.6. Struktura wiadomości DICOM 30
2.2.7. DIMSE-C i DIMSE-N 30
2.2.8. Anonimizacja danych 33
2.2.9. DICOMweb 34
2.2.10. Inne koncepcje zdalnego przetwarzania dokumentów DICOM 35
2.3. API typu REST 36
2.4. Kryptografia jako element ochrony danych medycznych 37
2.4.1. Podstawy kryptografii 37
2.4.2. PKI - Infrastruktura Klucza Publicznego 41
2.4.3. Protokół HTTPS 43
2.4.4. Podpis elektroniczny 47
2.4.5. Zabezpieczony transfer danych w standardzie DICOM 47
2.5. Prawna ochrona danych 49
2.6. Anonimizacja, pseudonimizacja oraz deidentyfikacja danych 50

3. Analiza rozwiązań zdalnego przetwarzania danych 52

4. Wyniki pracy i dyskusja 56

5. Podsumowanie 62

Spis rysunków 65

Spis tabel 66

Bibliografia 69

Załączniki 69
A. Demonstrator kodu 70
B. Proponowane API REST 83

Wykaz symboli i skrótów

ACSE ang. *Association Control Service Element* - Element Usługowy Sterujący Powiązaniami.

AE ang. *Application Entity* - Jednostka aplikacji.

AES ang. *Advanced Encryption Standard* - jeden z symetrycznych algorytmów kryptograficznych.

AET ang. *Application Entity Title*- Tytuł Aplikacji.

AFOV ang. *Axial Field Of View* - osiowe pole widzenia.

API ang. *Application Programming Interface* - interfejs programistyczny.

BGO ang. *Bismuth Germanate* - scyntylator krystaliczny $Bi_4Ge_3O_{12}$.

CA ang. *Certificate Authority* - urząd certyfikacyjny.

CIŚ Centrum Informatyczne Świerk.

CPU ang. *Central Processing Unit* - procesor komputera.

CRT ang. *Coincidence Resolving Time*.

CSR ang. *Certificate Signing Request* - wiadomość wysłana przez aplikanta do CA.

CT ang. *Computed tomography* - Tomografia Komputerowa.

DAQ ang. *Data Acquisition System* - system akwizycji danych.

DES ang. *Data Encryption Standard* - jeden z symetrycznych algorytmów kryptograficznych.

DICOM ang. *Digital Imaging and Communications in Medicine* - Standard obrazowania cyfrowego i wymiany obrazów w medycynie.

DIMSE ang. *DICOM Message Service Element* - Elementy usługowe wiadomości DICOM.

FDG Fluorodeoksyglukoza.

FOM ang. *Figure Of Merit* - współczynnik dobroci.

FOV ang. *Field Of View* - pole widzenia.

FWHM ang. *Full width at half maximum* - szerokość połówkowa.

GPU ang. *Graphical Processing Units* - karta graficzna.

HTTPS ang. *HyperText Transfer Protocol Secure* - internetowy protokół komunikacji klienta z serwerem.

Instancja SOP ang. *SOP Instances*.

IOD ang. *Information Object Definition*- Model IOD definiuje jakie wiadomości dany obiekt ma zawierać.

J-PET ang. *The Jagiellonian Positron Emission Tomograph* - Pozytonowy Emisyjny Tomograf zbudowany z plastikowych detektorów na Uniwersytecie Jagiellońskim w Krakowie.

JSON ang. *JavaScript Object Notation*- Format zapisu danych.

LOR ang. Line of response - linie łączące miejsce detekcji jednego sygnału z drugim.

LSO ang. *Lutetium Oxyorthosilicate* - Scyntylator krystaliczny $Lu_2(SiO_4)O$.

MLEM ang. *Maximum Likelihood Expectation Maximization*- Iteracyjny algorytm rekonstrukcji obrazów PET.

MRI ang. *Magnetic resonance imaging* - Obrazowanie metodą rezonansu magnetycznego.

NCBJ Narodowe Centrum Badań Jądrowych.

PACS ang. *Picture Archiving and Storage System*.

PET ang. *Positron emission tomography* - Pozytonowa Tomografia Emisyjna.

PKI ang. *Public Key Infrastructure*- Infrastruktura Klucza Publicznego.

Powiązanie ang. *Association*.

RA ang. *Registration Authority* - urząd rejestracyjny.

RSA jeden z asymetrycznych algorytmów kryptograficznych.

SCP ang. *Service Class Provider* - Usługodawca.

SCU ang. *Service Class User* - Usługobiorca.

SSL ang. *Secure Socket Layer*- Protokół sieciowy zapewniający prywatność komunikacji przez Internet.

TLS ang. *Transport Layer Security*- Rozwinięcie protokółu SSL.

TOF ang. *Time of Flight* - technika mierząca różnice czasowe pomiędzy pojawianiem się w detektorach kwantów gamma.

TOF-PET aparaty PET wykorzystujące technikę TOF.

Cel pracy

Celem pracy jest:

I. zebranie dostępnych danych dotyczących standardu DICOM, szyfrowania danych, typach komunikacji, zdalnym przetwarzaniu i przechowywaniu danych oraz prawnej ochronie danych

II. porównanie dwóch zaproponowanych rozwiązań dotyczących schematu informatycznej infrastruktury szpitala

III. stworzenie demonstratora kodu w pythonie przedstawiającego zasadę działania przetwarzania danych medycznych

Rozdział 1

Wstęp

1.1. Podstawy fizyczne tomografii emisyjnej

Podstawą działania skanera PET (ang. *Positron Emission Tomography*), czyli Pozytonowego Emisyjnego Tomografu jest rozpad β^+. Rozpad β^+ związany jest z emisją pozytonu i neutrina. [6]

Rozpad β^+ izotopu F-18:

$$^{18}_{9}\mathrm{F} \rightarrow {}^{18}_{8}\mathrm{O} + \mathrm{e}^+ + \nu_\mathrm{e} \tag{1.1}$$

Wyemitowany pozyton zazwyczaj ma niską energię kinetyczną co oznacza, że przebyta przez niego droga w ośrodku jest mała. Po napotkaniu elektronu z atomów ośrodka zachodzi zjawisko anihilacji. Jest to drugi kluczowy proces w tomografie PET. W wyniku anihilacji elektronu i pozytonu produkowane są dwa fotony gamma. Energia tego promieniowania jest równa sumie mas spoczynkowych i energii kinetycznych elektronu i pozytonu.

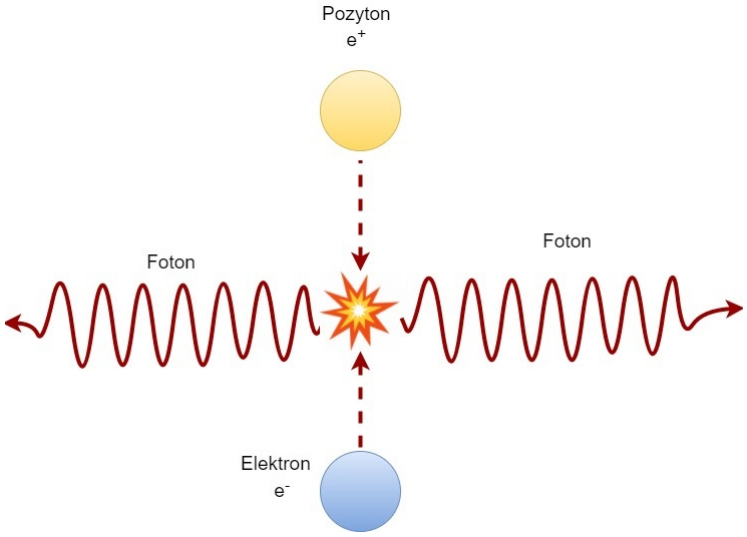
$$\mathrm{e}^- + \mathrm{e}^+ \rightarrow \gamma + \gamma \tag{1.2}$$

Przy anihilacji elektronu z pozytonem o energii E energia wytworzonych fotonów gamma równa jest:

$$h\nu = E + 2m_0c^2 \tag{1.3}$$

Gdy $E = 0$ czyli w przypadku anihilacji pozytonu w spoczynku, powstają 2 fotony gamma o energii $m_0c^2 = 0{,}511$ MeV.

Fotony pochodzące z procesu anihilacji rozchodzą się pod kątem około 180 stopni przez co ich sumaryczny pęd jest równy zeru zgodnie z zasadą zachowania pędu.[44]



Rysunek 1.1: Schemat zjawiska anihilacji

Wykrywanie fotonów gamma opiera się na zjawisku scyntylacji, która jest rodzajem luminescencji, ponieważ pod wpływem promieniowania jonizującego (X, alfa, beta, gamma, neutrony, protony etc.) padającego na materiał scyntylacyjny emitowane jest światło w zakresie widzialnym lub mu bliskim wykrywane następnie przez fotokomórkę.

Wpadający do scyntylatora kwant gamma może oddać część lub całość swej energii elektronowi, który następnie zostaje wybity z powłoki atomowej. W wyniku jonizacji, ekscytacji atomów lub molekuł scyntylatora, wybity elektron oddaje energię innym elektronom, które wracając ze stanu wzbudzonego na stan podstawowy wywołują emisję fotonów z zakresu światła widzialnego. Liczba fotonów powstających w scyntylatorze jest proporcjonalna do energii przekazanej przez kwant gamma elektronowi.

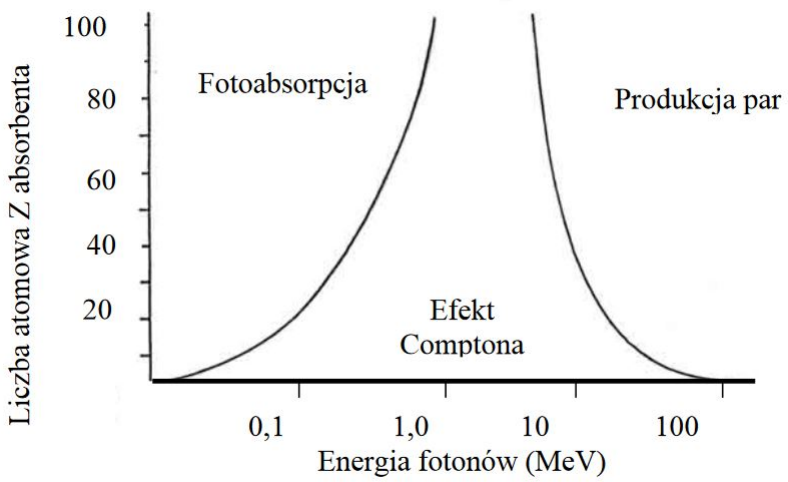
Wywołane w detektorze przez promieniowanie jonizujące błyski światła tzw. scyntylacje są za pomocą fotopowielaczy zamieniane w sygnał elektryczny w następujący sposób. Wyemitowane fotony docierają do fotokatody fotopowielacza, gdzie wybijane są elektrony. Przez zjawisko wtórnej emisji impuls zostaje wzmocniony. Na każdym stopniu układu powielającego powstają zazwyczaj 3-4 nowe elektrony, a uzyskane wzmocnienie nie generuje szumu. Uzyskany ładunek sygnału elektrycznego jest wobec tego proporcjonalny do liczby fotonów, które padają na okno fotopowielacza. [19],[13]

Podsumowując, dzięki scyntylatorowi dokonywana jest konwersja energii gamma na obraz świetlny. [26, 20]
Dodatkowo, fakt, że w większości przypadków anihilacja prowadzi do powstania dwóch rozchodzących się pod kątem około 180 stopni fotonów wykrywanych jednocześnie przez detektor scyntylacyjny jest wykorzystywany do rekonstrukcji obrazów PET. (patrz: rozdział 1.1.1.)

Wyróżniane są scyntylatory organiczne i nieorganiczne. Do scyntylatorów organicznych można zaliczyć takie związki jak: naftalen, antracen, stilben. Scyntylatorami krystalicznymi powszechnie stosowanymi w tomografii PET są: ortokrzemian lutetu domieszkowany cerem: LSO (ang. *Lutetium Oxyorthosciilicate*) oraz germanian bizmutu: BGO (ang. *Bismuth Germanate*). LSO i BGO charakteryzują się wysoką gęstością, co prowadzi do wysokiej efektywności detekcji. LSO cechuje się wyższą wydajnością świetlną od BGO. [15]

Promieniowanie gamma może oddziaływać z materią na wiele sposobów. Kwanty gamma

mogą oddziaływać z elektronami, jądrami i polami elektrycznymi. W scyntylatorze promieniowanie rentgenowskie i gamma może przekazać energię elektronom na trzy sposoby: przez efekt fotoelektryczny, rozpraszanie Comptonowskie i tworzenie par. [13]. Rozważając kwanty gamma o energii wynoszącej powyżej 511 keV brane pod uwagę są dwa procesy: efekt fotoelektryczny i efekt Comptona. Zjawisko tworzenia par będzie zaniedbywane przy tej energii (patrz rysunek 1.2). Natomiast efekt Comptona polega na przekazaniu części energii elektronowi, która zależy od kąta rozproszenia elektronu. [15] [11]



Rysunek 1.2: Obszary dominacji występowania efektów oddziaływania fotonów z materią. Wykres jest w funkcji energii fotonów oraz liczby atomowej Z absorbentu.[36]

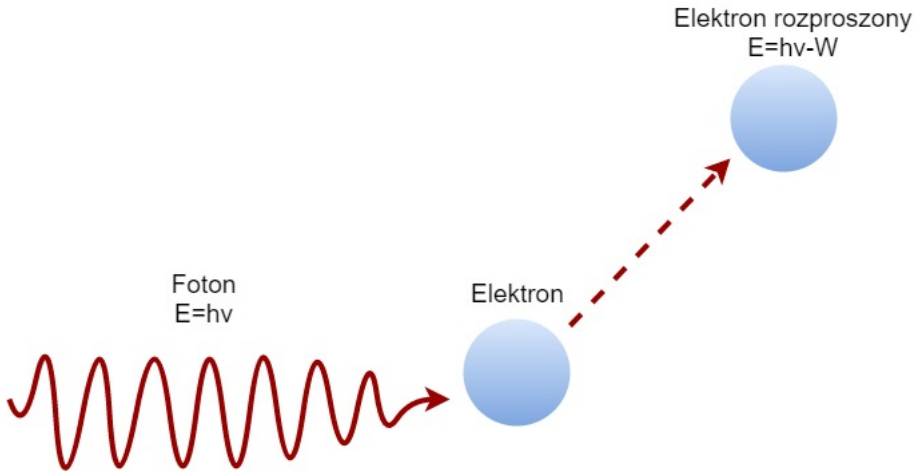
Efekt fotoelektryczny jest to zjawisko fizyczne, w którym kwant o energii $h\nu$ może wybić elektron z powłoki elektronowej atomu. Wybity elektron całkowicie pochłania energię kwantu, ale jego energia po wyjściu na zewnątrz atomu zostaje pomniejszona o energię potrzebną na jego wybicie z atomu i jest opisywana równaniem:

$$E = h\nu - W \tag{1.4}$$

gdzie: W to energia wiązania danego elektronu
h to stała Plancka
E to energia kinetyczna wybitego elektronu
ν częstotliwość padającego fotonu

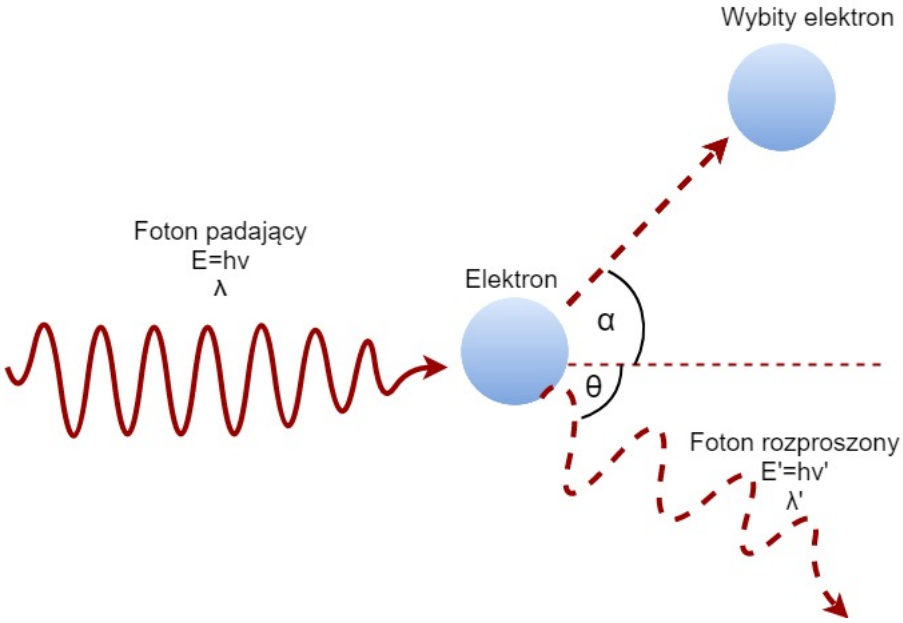
Należy zaznaczyć, że gdy W jest mniejsze bądź równe $h\nu$ nie ma możliwości, aby zjawisko fotoelektryczne zaszło.

Zjawisko wybijania przez fotony elektronów nazywane jest zewnętrznym efektem fotoelektrycznym. [2]. Dodatkowo, gdy wybity zostaje elektron z wewnętrznej powłoki, następuje emisja niskoenergetycznego promieniowania X. Jest to spowodowane przechodzeniem elektronów z wyższych powłok w miejsce po wybitym elektronie tzw. dziurę. [18] W rezultacie wyżej opisanych zjawisk cała energia padającego fotonu gamma jest zaabsorbowana w materiale. [13]



Rysunek 1.3: Schemat efektu fotoelektrycznego

Efekt (rozpraszanie) Comptona to zjawisko, gdzie kwant promieniowania wybija elektron z powłoki atomowej. Do wybicia jego wystarcza część energii fotonu. Oprócz wybitego elektronu obserwuje się foton o energii pomniejszonej o wartość energii oddaną wybitemu elektronowi.



Rysunek 1.4: Schemat rozpraszania Comptona, λ to długość fali padającej, λ' to długość fali rozproszonej, θ to kąt rozproszenia fotonu.

Energia rozproszonego fotonu wyrażana jest wzorem:

$$E' = \frac{E}{1 + \frac{E}{m_e c^2}(1 - \cos\theta)} \tag{1.5}$$

gdzie:

E to energia padającego fotonu
θ to kąt pomiędzy kierunkiem fotonu padającego i rozproszonego
m_e to masa spoczynkowa elektronu
c to prędkość światła

Energia rozproszonego fotonu maleje wraz ze wzrostem kąta rozproszenia. Minimalna wartość osiągana jest dla kąta θ równego 180°. Natomiast zmiana długości fali dla tego kąta osiąga maksymalną wartość $\Delta\lambda$ równa się około 0,5 $\cdot 10^{-11}$ m.
Wzór na przesunięcie Comptona, czyli zwiększenie długości fali rozproszonego fotonu:

$$\lambda' - \lambda = \lambda_c(1 - \cos\theta) \tag{1.6}$$

gdzie:
h to stała Plancka
λ' to długość fali rozproszonej
λ to długość fali padającej
λ_c to komptonowska długość fali elektronu; jest stałą; wyraża się wzorem: $\lambda_c = \frac{h}{m_e c}$

Wzór opisuje zależność kątową przekazu energii. Zmiana częstotliwości fali fotonu odpowiada zmianie jego energii, a więc wzór opisuje ile energii przekazał foton elektronowi.

Uogólnieniem opisu rozpraszania Comptona jest wzór Kleina-Nishiny. Obejmuje on zarówno zjawisko Comptona jak i rozpraszanie Thompsona. Wzór ten opisuje różniczkowy przekrój czynny dla rozproszenia fotonów γ na elektronie:

$$d\sigma = \frac{r_0^3}{2}\left(\frac{\nu}{\nu_0}\right)^2\left(\frac{\nu_0}{\nu} + \frac{\nu}{\nu_0} - \sin^2\varphi\right)\mathrm{d}\Omega \tag{1.7}$$

Gdzie:
$r_0 = 2{,}82{\cdot}10^{-13}$ cm klasyczny promień elektronu,
ν_0 częstość pierwotnych promieni γ,
ν częstość rozproszonych promieni γ, zależna od kąta rozproszenia γ

Różniczkowy przekrój czynny $d\sigma$ odpowiada prawdopodobieństwu, że w momencie przejścia fotonu γ przez absorbent posiadający 1 elektron na 1 cm^2 wystąpi rozpraszanie pod kątem φ w element kąta bryłowego dΩ.[50]

1.1.1. PET we współczesnej medycynie

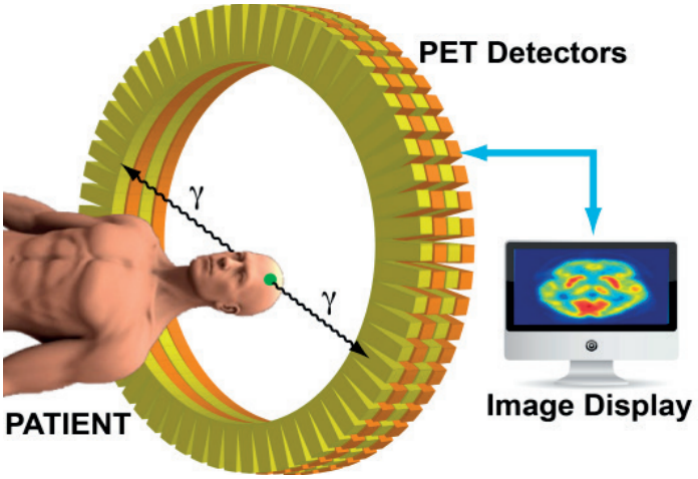
PET jest medyczną techniką używaną zarówno w diagnostyce nowotworów jak i również do kontroli w badaniach związanych z radio- i chemio-terapią czy w kardiologii i neurologii. [15] [11]

PET jest jedną z najbardziej technicznie zaawansowanych technik obrazowania używaną w diagnostyce medycznej. Pozwala na nieinwazyjne obrazowanie tomograficzne fizjologicznych procesów in vivo.[16] Co jest odmienne od standardowej tomografii komputerowej- CT czy rezonansu magnetycznego - MRI, gdzie obrazowana jest struktura tkanek.

Obecnie fluorodeoksyglukoza (F-18-FDG) jest najczęściej stosowanym radiofarmaceutykiem do diagnostyki PET. Po podaniu dożylnym radiofarmaceutyk jest transportowany do komórek w ciele pacjenta. Komórki nowotworowe mają szybszy metabolizm, a więc absorbują więcej FDG niż zdrowe komórki, dzięki czemu możliwa jest detekcja ognisk nowotworowych.

Przed badaniem pacjentowi podawany jest zastrzyk z radioaktywnym markerem, który jest emiterem pozytonów. Po napotkaniu elektronu z atomów ciała pacjenta następuje anihilacja pozytonu ze znacznika. W większości przypadków ich masa zostaje zamieniona w energię w postaci dwóch kwantów gamma lecących w przeciwne strony o energii równej 511 keV. Detektory promieniowania są ułożone warstwowo w kształcie pierścieni wokół badanego pacjenta. Do detekcji promieniowania komercyjnie używa się nieorganicznych kryształów scyntylacyjnych. [15] [11] (patrz: 1.1)

Podczas jednej anihilacji pozyton-elektron generowane są dwa fotony docierające do dwóch punktów na pierścieniach detektorów. Rekonstruowane są linie łączące miejsce detekcji jednego sygnału oraz drugiego sygnału, tzw. LOR (ang. *Line of Response*). W trakcie badania można wygenerować kilka milionów linii LOR. Dzięki nim, w obszarze przecięcia linii, w bardzo dobrym przybliżeniu, określa się rozkład gęstości punktów anihilacji. [11] Na podstawie zarejestrowanych przez tomograf PET sygnałów rekonstruowane są obszary w ciele pacjenta, z których emitowane były fotony anihilacyjne. [15]



Rysunek 1.5: Schemat tomografu PET [10]

Technika TOF (ang. *Time of Flight*) polega na pomiarze różnicy czasowej pomiędzy pojawieniem się w detektorach kwantu gamma. Aparaty PET wykorzystujące tą technikę nazywane są TOF-PET. [11] Używając metody TOF można określić pozycję anihilacji wzdłuż LOR.

$$\mathrm{TOF} = \frac{(t_1 + t_2)}{2} - \frac{(t_3 + t_4)}{2} \tag{1.8}$$

$$\Delta x = \mathrm{TOF} \cdot \frac{c}{2} \tag{1.9}$$

gdzie:
Δx opisuje odległość pomiędzy punktem anihilacji a środkiem LOR;
c stanowi prędkość światła. [15]

1.1.2. J-PET innowacja względem PET

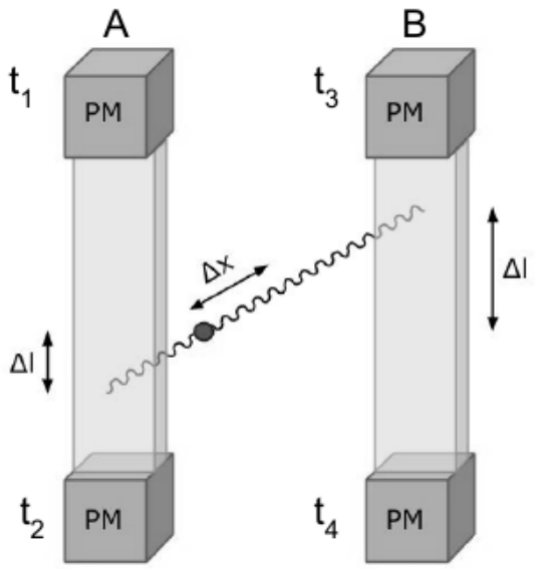
J-PET (ang. *The Jagiellonian Positron Emission Tomograph*) to projekt w którym uczestniczą Uniwersytet Jagieloński, Uniwersytet Marii-Curie Skłodowskiej z Lublina, Uniwersytet

Wiedeński, Laboratori Nazionali di Frascati oraz NCBJ (Narodowe Centrum Badań Jądrowych). Jego celem jest opracowanie nowego typu tomografu opartego na scyntylatorach polimerowych z bardzo dobrym pomiarem czasu TOF oraz pola widzenia FOV (ang. *Field Of View*) całego ciała człowieka. [15] [11]

Anihilacyjne kwanty gamma o energii 511 keV oddziałują z polimerowymi scyntylatorami przez efekt Comptona. Polimerowe scyntylatory łatwo można formować w postaci długich pasków, dodatkowo niski współczynnik tłumienia światła zapewnia jego propagację nawet dla pasków dużych rozmiarów. Z tego powodu komora służąca do pomiarów zbudowana jest z długich modułów usytuowanych wzdłuż ciała pacjenta. Każdy plastikowy pasek jest odczytywany przez fotopowielacz z dwóch końców. Na tym etapie sygnały świetlne są zamieniane na sygnały elektryczne. [15] Pozycja oddziaływań kwantów gamma ze scyntylatorem może być ustalana z różnicy czasowej przychodzących sygnałów świetlnych z dwóch końców detekcyjnych modułów fotopowielacza [15]:

$$\Delta l_A = (t_1 - t_2)V_A \tag{1.10}$$

gdzie:
Δl_A oznacza dystans pomiędzy punktem oddziaływania a środkiem modułu
t_1, t_2 określają czas dotarcia sygnału świetlnego do każdej ze stron scyntylatora
V_A opisuje efektywną prędkość sygnału świetlnego w zasięgu scyntylatora [15]



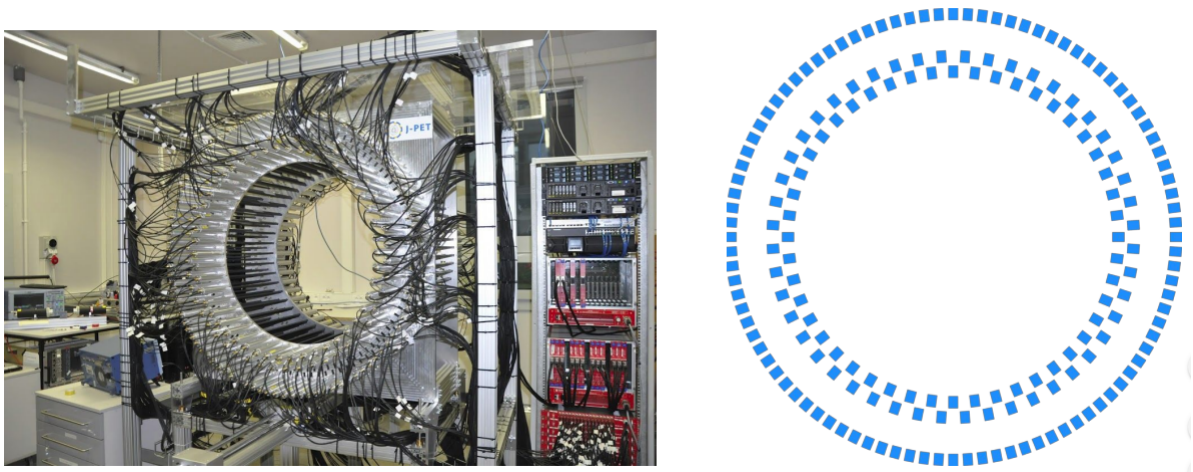
Rysunek 1.6: Metoda rekonstrukcji punktu anihilacji. Na obrazku przedstawione zostały dwa plastikowe paski z modułami detekcyjnymi z dwoma anihilacyjnymi kwantami gamma. [8] [12] [9]

Dodatkowo, odczyt może mieć miejsce poza komorą detekcyjną. Upraszcza to konstrukcję hybrydy PET/MR, czyli systemu obrazowania umożliwiającemu jednoczesną akwizycję danych dwoma metodami: pozytonowej tomografii emisyjnej i rezonansu magnetycznego oraz umożliwia rozszerzenie osiowego pola widzenia tzw. AFOV (ang. *axial Field Of View*) bez znaczącego wzrostu kosztów. Koszt elektronicznego odczytu nie zmienia się przy poszerzaniu osiowego pola widzenia J-PET, ponieważ liczba fotopowielaczy i elektronicznych kanałów

pozostaje niezależna od AFOV. Znajdują się one bowiem na początku i końcu każdego paska scyntylacyjnego, niezależnie od jego długości. Aby porównać wydajność tomografu PET z krystalicznych scyntylatorów do J-PET zbudowanego z plastikowych scyntylatorów, zdefiniowano współczynnik dobroci, współczynnik jakości tzw. FOM (ang. *figure of merit*) dla obrazowania całego ciała. Opisany jest jako prawdopodobieństwo detekcji anihilacji podzielone przez CRT (ang. *Coincidence Resolving Time*) i liczbę rozważanych pozycji pacjenta. Porównanie zdefiniowanego FOMu dla aparatu J-PET i skanera charakteryzującym się scyntylatorami z LSO, z $AFOV = 20$ cm i $CRT = 400$ ps pokazuje, że możliwe jest rozwiązanie problemu niewielkiego prawdopodobieństwa detekcji plastikowych scyntylatorów przez użycie dłuższych modułów i zwiększeniu warstw detekcyjnych. Warte podkreślenia jest, że moduły plastikowych scyntylatorów mogłyby mieć nawet 2 m długości, ale ich wydłużanie powoduje spadek CRT, co jest niepożądane. [15]

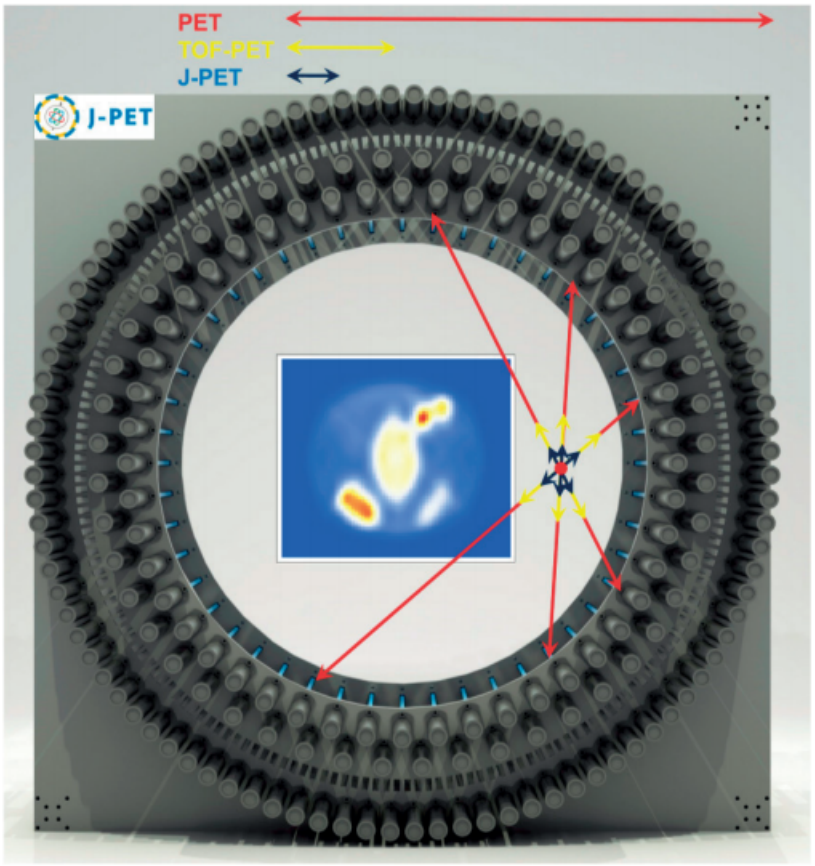
J-PET głównie używa informacji dotyczącej czasu zamiast energii, żeby zdobyć informację na temat miejsca anihilacji. W plastikowych scyntylatorach oddziaływanie z fotonem gamma powoduje wzbudzenia elektronów o bardzo krótkim czasie powrotu do stanu podstawowego z emisją fotonów. Przeciętny czas życia stanu wzbudzonego wynosi 1,8 ns. Powoduje to bardzo dobrą czasową rozdzielczość oraz zmniejszenie nakładania się sygnałów w odniesieniu do detektorów krystalicznych jak n.p. scyntylatory krystaliczne LSO czy BGO charakteryzujące się czasem wzbudzenia odpowiednio równą 40 ns i 300 ns. [15]

W 2014 roku został opatentowany pierwszy pełnowymiarowy prototyp J-PET zmontowany ze 192 modułów detekcyjnych o wewnętrznej średnicy 850 mm. Moduły zostały usytuowane w trzy nie pokrywające się pierścienie. [15]



Rysunek 1.7: Zdjęcie pełnowymiarowego prototypu J-PET; Schemat przedstawiający warstwy pierścieni detekcyjnych. [15]

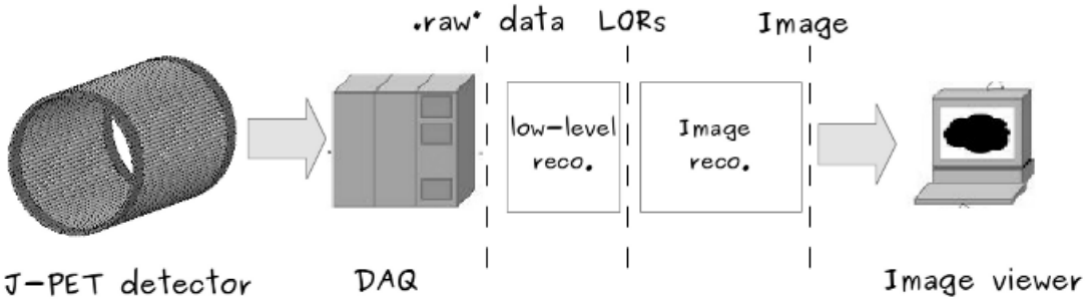
Przy trzykrotnym zwiększeniu pola obrazowania z 17 cm (wartość charakterystyczna dla komercyjnych aparatów PET) na 50 cm (prototyp J-PET) zauważono porównywalną rozdzielczość przestrzenną oraz dwukrotnie lepszą rozdzielczość czasową. [11]



Rysunek 1.8: Schemat LOR dla 3 przykładowych anihilacji (obszar oznaczony czerwoną kropką). Porównano informacje związane z punktem anihilacji dla trzech różnych metod: standardowy PET – czerwona linia, TOF-PET o rozdzielczości TOF równej 540 ps (szerokości połówkowej, tzw. FWHM (ang. *Full width at half maximum*)) – żółta linia oraz J-PET o 290 ps (FWHM) – niebieska linia.[11]

Przetwarzanie danych

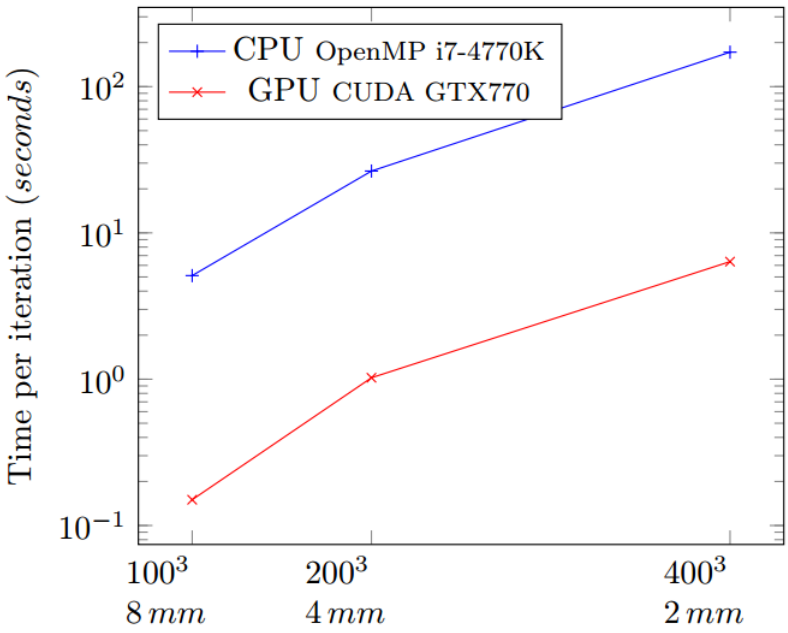
Proces zaczyna się ze zbieraniem nieobrobionych danych (czas i amplituda są przekształcane na format cyfrowy przez przetwornik czasu na postać cyfrową oraz przetwornik analogowo-cyfrowy. Następnie dane są łączone w sygnały i tłumaczone na miejsce zdarzenia w konkretnych modułach scyntylatora. Ostatecznie, zderzenia fotonów gamma z konkretnymi paskami detektora są łączone tak, aby utworzyć LOR. Zbiór LORów jest potem używany jako dane wejściowe do procedury rekonstrukcji obrazu. [16]



Rysunek 1.9: Schemat przepływu danych podczas rekonstrukcji obrazu PET. Nieprzetworzone dane są zbierane przez system DAQ (ang. *Data Acquisition System*). Następnie są przetwarzanie przez moduł rekonstrukcji niskiego poziomu (low-level reco.) i prowadząc do utworzenia zbiorów zrekonstruowanych linii LOR, które następnie są wysyłane do modułu rekonstrukcji obrazu (Image reco.). Końcowy obraz może być odtworzony w przeznaczonym do tego celu programie bądź może być eksportowany do formatu DICOM. [16]

Nisko i wysokopoziomowe przetwarzanie danych

Następnym krokiem w przetwarzaniu danych jest rekonstrukcja rozkładu radioaktywności w ciele pacjenta na podstawie zebranych LORów. Jednym z przyjętych mechanizmów jest bardzo popularne podejście w oparciu o iteracyjny algorytm MLEM (ang. *Maximum Likelihood Expectation Maximization*). Dodatkowe informacje zbierane dzięki zastosowaniu metody TOF służą zwiększeniu dokładności i jakości rekonstrukcji. Rekonstrukcja jest jedną najbardziej czasochłonnych części z całego przypływu danych. W celu obniżenia czasu przetwarzania, stosowane są techniki przetwarzania równoległego. Oprócz rozwiązania opierającego się na wielo-rdzeniowej architekturze procesora CPU (ang. *Central Processing Unit*) testowana jest możliwość z wykorzystaniem procesorów graficznych GPU (ang. *Graphical Processing Units*). Wydajna rekonstrukcja obrazu przy użyciu algorytmu MLEM została zaimplementowana na procesorze GPU. Obecny algorytm rekonstrukcji opierający się na GPU w J-PET jest zdolny do dostarczenia pełnych 3D rekonstrukcji obrazu o 200^3 2 mm wokseli w czasie około jednej minuty, przez wykorzystanie informacji z metody TOF. Porównanie czasu rekonstrukcji CPU i GPU na iteracje dla zasymulowanego fantomu z próby Sheppa-Logana, zostało przedstawione na poniższym wykresie. [16]



Rysunek 1.10: Implementacja CPU i GPU. Czas iteracji pojedynczej rekonstrukcji obrazu jako funkcja rozdzielczości obrazu docelowego.[16]

Rozdział 2

Współczesne standardy w zdalnym przetwarzaniu i ochrony danych.

2.1. Zdalne przetwarzanie i przechowywanie danych

Przetworzenie zbioru danych ze scyntylatora na trójwymiarowy obraz ciała pacjenta z zaznaczonymi obszarami wzmożonej aktywności metabolicznej jest zadaniem na tyle złożonym, że potrzebuje dużych mocy obliczeniowych. Złożoność ta dla skanerów o polu widzenia FOV całego ciała człowieka takich jak J-PET będzie dodatkowo większa. Wykorzystanie w pomiarze precyzyjnej informacji ToF pozwala także na zastosowanie nowej klasy algorytmów rekonstrukcji co także może prowadzić do zwiększonego zapotrzebowania na moc obliczeniową.

Dodatkowo warto nadmienić, że zarządzanie i przechowywanie danych medycznych nie jest trywialnym zagadnieniem. Trzeba brać pod uwagę koszty utrzymania specjalistycznych serwerów oraz kopii zapasowych systemów. Ponadto, ze względu na poufny charakter danych medycznych, wymagania bezpieczeństwa muszą być odpowiednio zrealizowane. W dodatku oczekuje się, że całkowity rozmiar danych obrazowania medycznego, które muszą być przechowywane, wzrasta bardzo szybko i ograniczona przestrzeń do przechowywania danych na lokalnych zasobach komputerowych w medycznych jednostkach będzie stanowiła ważne ograniczenie. [16]

Oprócz prezentowanego schematu obliczeniowego, w którym przetwarzanie danych jest wykonywane lokalnie używając wielo-rdzeniowego procesora CPU lub/i rozwiązań związanych z procesorem GPU, rozważane są także inne podejścia opierające się na zdalnej architekturze rozproszonej. Wejściowe dane (np. zbiór LORów) nie są przetwarzane lokalnie, lecz początkowo podlegają anonimizacji i szyfrowaniu, a następnie są transferowane do zdalnego centrum obliczeniowego, gdzie są przetwarzane przez węzły obliczeniowe klastra, gridu. [16] Klaster obliczeniowy to zbiór połączonych ze sobą komputerów, serwerów (tzw. węzły, ang. *node*) tworzących zintegrowane środowisko informatyczne.[33] Przykładem może być klaster HPC (ang. *High Performance Computing*) usytuowany w Centrum Informatycznym Świerk (CIŚ) w NCBJ. Klaster ten składa się z ponad 1400 serwerów wyposażonych w ponad 31 500 fizycznych rdzeni (CPU) i 183 TB pamięci (RAM).

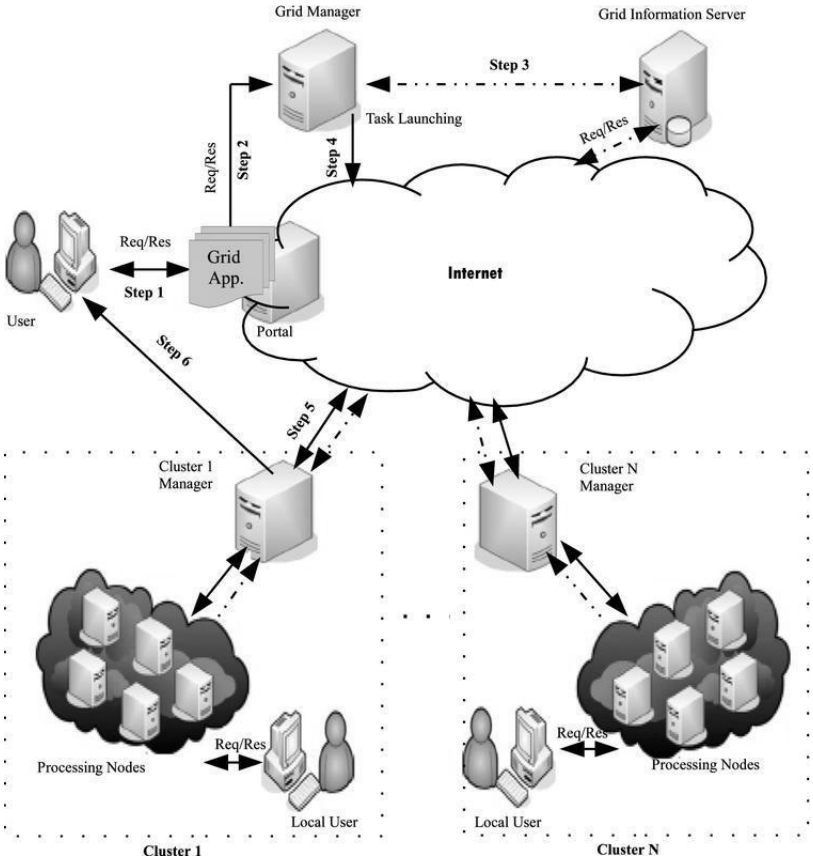
W rozproszonym modelu obliczeniowym przetwarzanie i przechowywanie danych obrazowania medycznego jest przeniesione ze szpitali i ośrodków opieki zdrowotnej do dedykowanych centrów obliczeniowych, które dostarczają specjalistycznych usług w chmurze. Szpitale

i ośrodki zdrowia odgrywają role klientów. W rezultacie, szpitale są wolne od problemów odnoszących się do przechowywania danych medycznych, procedur związanych z tworzeniem kopii zapasowych czy ograniczeń przestrzeni dyskowej. Dodatkowo, centralizacja zapewniana przez centrum obliczeniowe ulepsza poziom bezpieczeństwa przez kontrolowanie całego systemu aplikacji oraz przez automatyczne aktualizacje zabezpieczeń. Zaletą przetwarzania zdalnego jest możliwość wykorzystania technik uczenia maszynowego do automatycznego rozpoznawania cech obrazów medycznych w szczególności wykrywania zmian nowotworowych. Przykładem analiz prowadzonych w tym kierunku jest np: analiza sieciami deep learning firmy Google wykrywania nowotworów skóry. [7]

Wśród wad w tym podejściu, można znaleźć potrzebę transferu danych pomiędzy szpitalem a centrum obliczeniowym, które liczy około 1 GB na badanie, co mogłoby powodować wzrost czasu potrzebnego na dostęp. Dodatkowo, zdalne przetwarzanie wymaga per se, zaufanego łącza przepływu danych. [16]

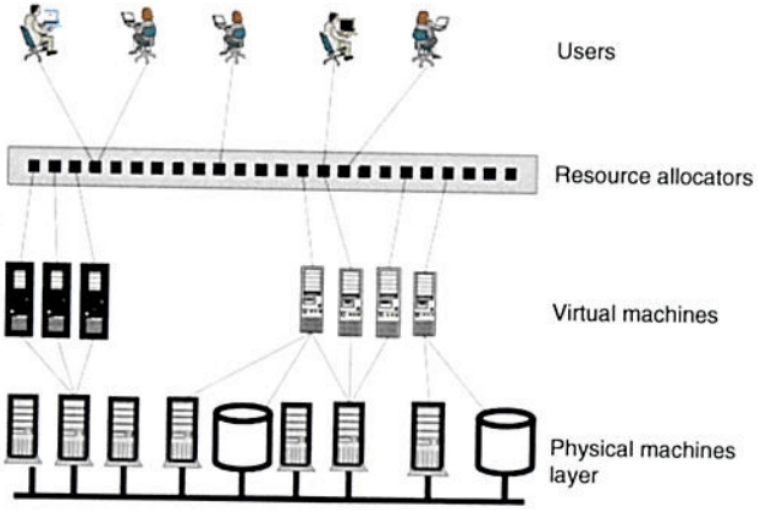
Dla certyfikowanych centrów danych medycznych, określone są wyższe wymagania niezawodności (>99.99%) i uprzednio określone cyberbezpieczeństwo danych z prawną i finansową odpowiedzialnością. [17] Poufne dane mogą być dodatkowo anonimizowane, dzięki czemu, nawet jeśli zostaną przechwycone i rozszyfrowane nie będą mogły zostać powiązane z konkretnymi osobami. Podstawowe zabezpieczenia danych medycznych w trakcie transferu mogą być dostarczane w często stosowanej strukturze PKI (ang. *Public Key Infrastructure*). Przenoszone dane są zaszyfrowywane przy użyciu publicznej i prywatnej kryptograficznej pary kluczy uzyskanej i współdzielonej za pośrednictwem zaufanego organu.

Istnieją dwa architektoniczne rozwiązania dla medycznych obliczeń rozproszonych: przetwarzaniu sieciowym, gridzie (ang. *grid computing*) oraz chmurze obliczeniowej (ang. *cloud computing*). [17] W gridzie, pamięć i zasoby obliczeniowe mogą być rozproszone geograficznie i dzielone pomiędzy właścicielami. Zasoby są połączone, ale nie są zarządzane centralnie. Schemat gridu został przedstawiony na rysunku 2.1.



Rysunek 2.1: Schemat usługi w gridzie. [14]

Ogólny schemat chmury obliczeniowej został przedstawiony na rysunku 2.2. Główną zaletą jest inteligentna wirtualizacja (ang. *smart virtualization*) optymalizująca użycie zasobów. Wirtualne maszyny są tworzone na potrzebę zadań i niszczone natychmiast po ich ukończeniu w celu zwalniania przestrzeni dyskowej. [17]



Rysunek 2.2: Schemat usługi w chmurze z charakterystycznymi czteroma warstwami (patrząc od dołu schematu): obliczeniowe i magazynujące elementy komputerowe (hardware), maszyny wirtualne; moduł rozdzielania zadań (ang. *resource allocator*); oraz użytkowników. [17]

Opisane dwa podejścia obliczeń rozproszonych, początkowo były zaprojektowane dla innych celów i innych społeczności użytkowników. Jednak, w najnowszych rozwiązaniach, elementy obliczeniowe w gridzie zmierzają do użycia typowych koncepcji chmurowych. [17]

2.2. Standard DICOM

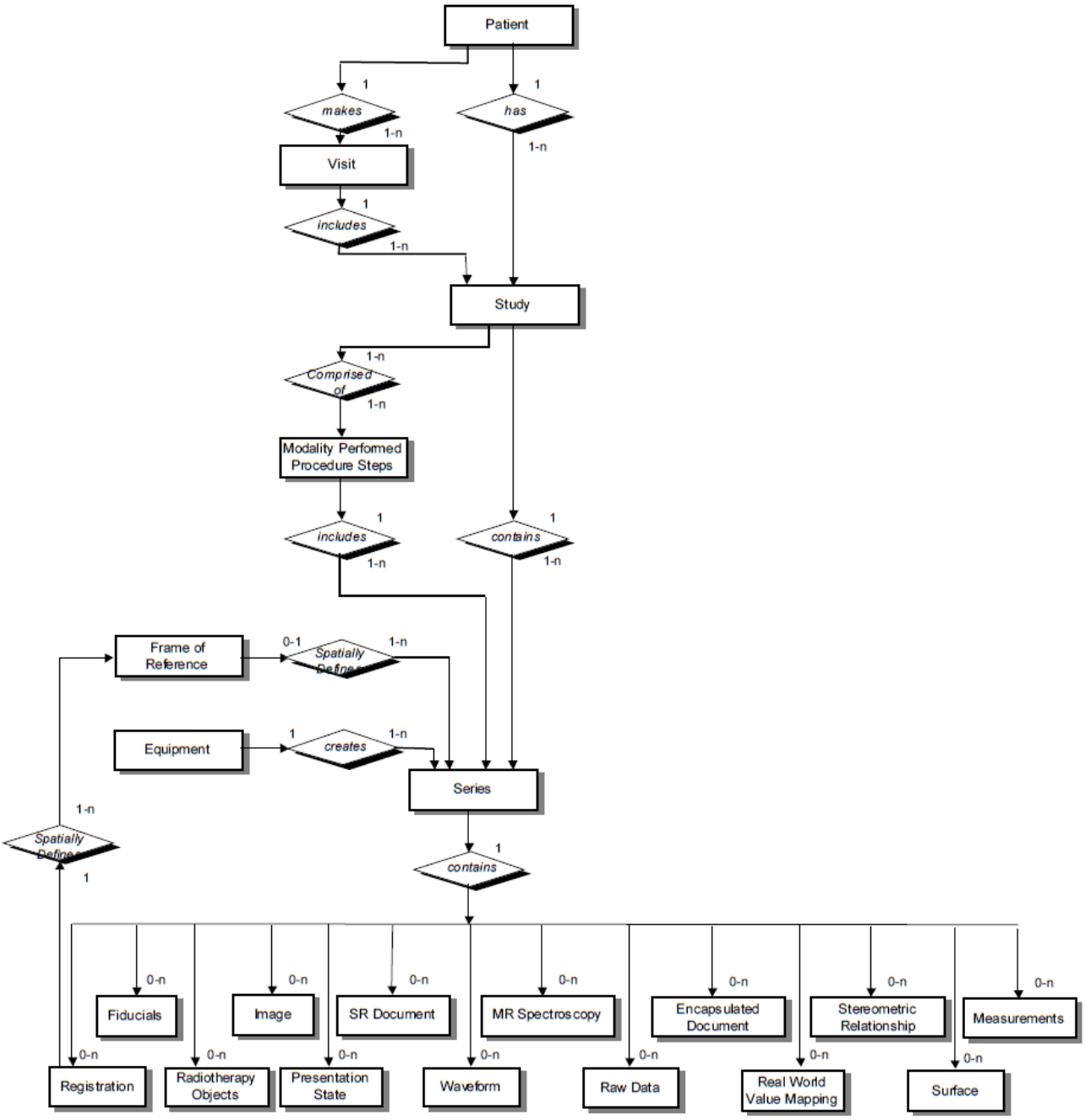
2.2.1. Opis standardu DICOM

Powszechnie stosowanym standardem wymiany danych w aplikacjach medycznych jest DICOM (ang. *Digital Imaging and Communications in Medicine*). Standard ten charakteryzuje się możliwością wymiany informacji graficznych wraz z powiązanymi danymi medycznymi. Określa strukturę plików oraz reguły kodujące. Umożliwia wymianę informacji (ang. *Service Class Specifications*) oraz dla komunikacji typu off-line definiuje metodę katalogowania informacji. Możliwa jest współpraca z wymiennymi nośnikami np. z dyskami CD, MOD (ang. *Magneto-Optical Disk*). [40]

Standard został zainicjowany i opisany przez stowarzyszenie NEMA (ang. *National Electrical Manufacturers Association*) oraz uczelnię ACR (ang. *American College of Radiology*). Celem opracowania było ujednolicenie wymiany i interpretacji obrazów diagnostycznych. [40] W ramach standardu opisany został m.in. sposób konstruowania komunikatów, metodyka zapisywania obrazów, struktura i kodowanie informacji, zasady działania wykorzystywanych algorytmów kompresji. [41]

Poniższy schemat przedstawia model DICOM rzeczywistych danych, który określa istotne obiekty świata rzeczywistego i ich zależności między nimi w zakresie standardu DICOM. To umożliwia utworzenie ogólnej struktury zapewniającej zwartość, zgodność pomiędzy różnymi

informacjami i danymi. Struktura danych jest zbieżna z rzeczywistymi danymi.[40]



Rysunek 2.3: Model DICOM świata rzeczywistego. Wartości nad strzałkami określają liczbę możliwych połączeń.[40]

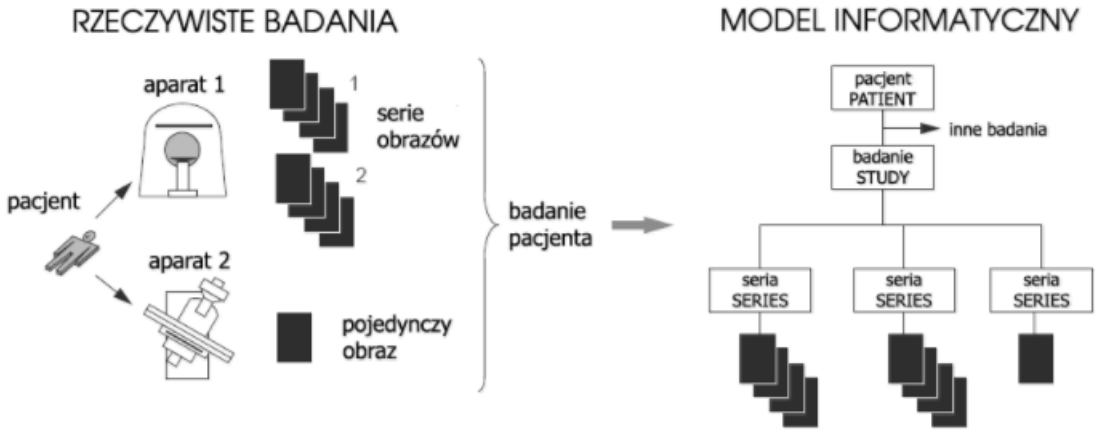
Obrazy w formacie DICOM są danymi medycznymi o dużej objętości co wymaga stosowania konkretnego oprogramowania i specjalnego sprzętu medycznego. W formacie DICOM obiekt jest wieloatrybutowy. Zawiera takie elementy jak: nazwisko, imię, identyfikator pa-

cjenta oraz unikalną informację zawartą w pikselach. Każdy obiekt DICOM może mieć tylko jeden atrybut, który posiada dane w formie pikseli, co jest jednoznaczne do przechowywania jednego obrazu. [40]

DICOM znajduje zastosowanie w przetwarzaniu nie tylko obrazów PET, ale też obrazów metodą rezonansu magnetycznego, tomografii komputerowej, cyfrowej angiografii substrakcyjnej, radiografii cyfrowej oraz cyfrowej radiografii konwencjonalnej i wielu innych dających wysoką rozdzielczość obrazu. [41, 42] Istnieje trzydzieści różnych grup roboczych (ang. *work-group* – WG) odpowiedzialnych za rozwój standardu DICOM dla wszystkich gałęzi medycyny. [41, 42]. Przykładowymi grupami roboczymi są: WG-07 Radioterapia, WG-12 Ultrasonografia, WG-16 Rezonans magnetyczny, WG-21 Tomografia komputerowa.

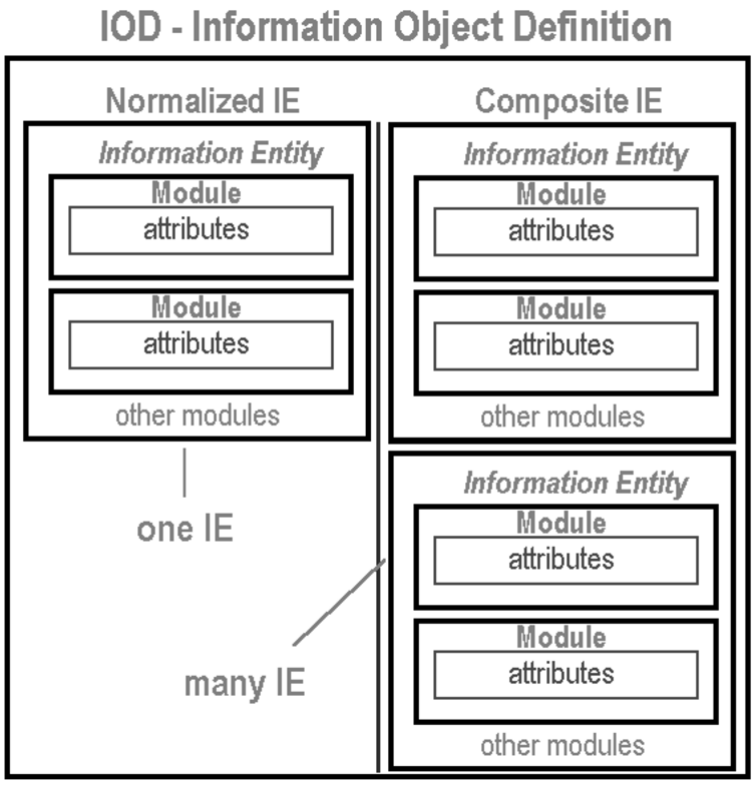
2.2.2. Model IOD

W modelu informatycznym definiowane są obiekty. Przykładowymi obiektami są: Patient, Study, Series. Na Rysunku 2.4 porównano model DICOM, który określa istotne obiekty świata rzeczywistego oraz rzeczywiste badanie. Obiekty w modelu informatycznym są zbieżne z rzeczywistym badaniem.



Rysunek 2.4: Przykładowe porównanie informacji rzeczywistych z zaproponowanym modelem informatycznym DICOM. [25]

Model IOD (ang. *Information Object Definition*) definiuje format prezentacji obiektu w tematycznych grupach, tzw. Entities oraz podzbiorach zwanymi modułami (Rysunek 2.5). Moduły są tworzone przez zbiory atrybutów opisujących cechy obiektów. IOD określa format danych dla różnych typów informacji, np. przebiegi czasowe, obrazy, raporty, wydruki, obiekty graficzne etc. [41]



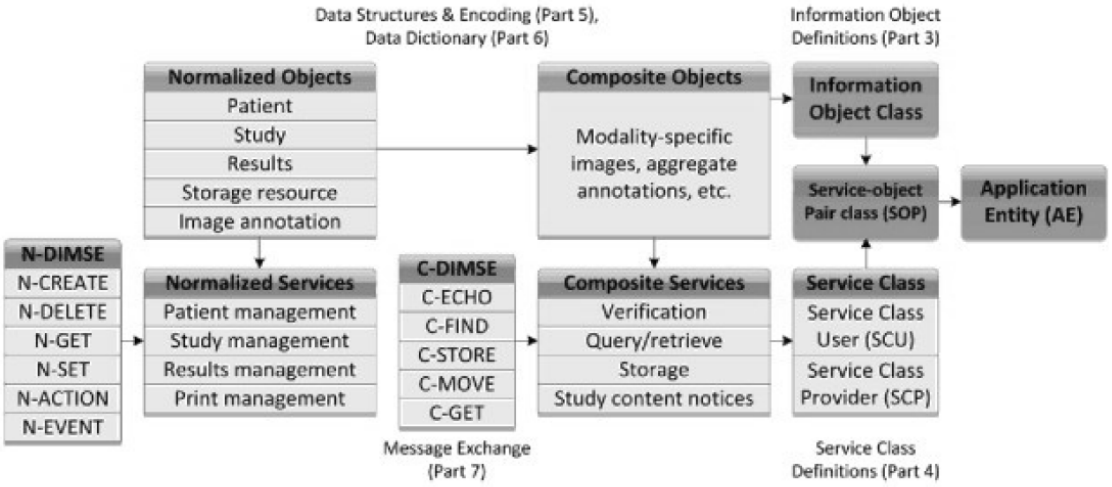
Rysunek 2.5: Schemat modelu informacji IOD. Widoczny jest podział na tematyczne zbiory - Information Entity. Te z kolei są podzielone na moduły, które składają się z artybutów.[24]

W szczególności model danego obiektu powinien zawierać:

1. opis (określa opis świata rzeczywistego reprezentowanego przez IOD. Podaje zakres prezentowanego obiektu)
2. diagram encji - związków (opisuje zależność między rzeczywistymi obiektami)
3. lista zbiorów danych jakie powinien zawierać

Lista zbiorów danych jest pogrupowana na: jednostkę informacji, moduły i link referencyjny do standardu. Np. Digital X-Ray Image IOD zawiera w jednostce informacji: Pacjent, Badanie, Seria, Obraz. Każda z tych jednostek informacji ma różne moduły, np. Obraz ma moduły takie jak: kolimator promieniowania X, sprzęt, piksele.

Obiekty mogą być złożone i znormalizowane (Rysunek 2.6). Obiektami znormalizowanymi są: Patient, Study, Results, Storage resource, Image annotation. Obiekty znormalizowane są obsługiwane przez serwisy znormalizowane. Przykładowymi serwisami znormalizowanymi są: Patient management, Study management, Results management, Print management. Wiadomości DIMSE-N (szerzej opisane w sekcji 2.2.7) odnoszą się do serwisów znormalizowanych. Obiektami złożonymi może być np. Modality-specific Images. Obiekty złożone są obsługiwane przez serwisy złożone. Przykładowymi serwisami znormalizowanymi są: Verification, Storage, Study content notices. Wiadomości DIMSE-C (szerzej opisane w sekcji 2.2.7) odnoszą się do serwisów złożonych.



Rysunek 2.6: Schemat ilustrujący klasy usług i obiektów DICOM. [48]

2.2.3. Klasy SOP

Klasy SOP (ang. *Service-Object Pair*) są dedykowane do komunikacji między aplikacją a urządzeniem wykonującym badania. SOP łączy dane z serwisami oraz określa usługi związane z IOD. Klasy SOP zawierają reguły i semantykę, które mogą ograniczać użycie usług w grupie usługowej DIMSE (ang. *DIMSE Service Group*). Są podstawową strukturą danych. Każda klasa usługi czy klasa SOP charakteryzuje się identyfikatorem UID (ang. *Unique Identifier*) definiującym zakres ich działania. Przykładem elementu usługowego jest: Store, Get, Find, Move, etc. Przykładem obiektu są obrazy tomografii komputerowej czy zdjęcia rezonansu magnetycznego. [41]

2.2.4. Struktura danych

Dane z plików DICOM można podzielić na dwie części:

1. dane o pliku (Dicom-Meta-Information-Header)

2. informacje jednego obiektu Service-Object Pair Instance (Dicom-Data-Set)

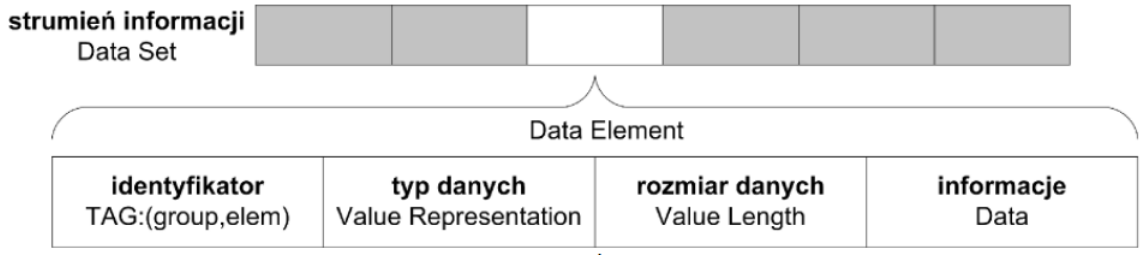
Podstawowa jednostka danych (ang. *Data Element*) opisuje atrybuty i definiowana jest przez minimum 3 pola:

1. Znacznik, identyfikator elementu danych (ang. *Data Element Tag*) (jest unikalnym identyfikatorem, który składa się z dwóch liczb: grupy (ang. *Group*) i jej elementu (ang. *Element*), które są zapisywane w formie liczb heksadecymalnych)

2. Typ danych (ang. *Value Representation*) (opisuje typ danych, który jest opisany w formie pary liter w kodzie ASCII. Umożliwia poprawną interpretację podstawowej jednostki danych- (ang. *Data Element*)

3. Rozmiar elementu (ang. *Value Length*) (określa rozmiar elementu w bajtach)

4. Informacje [1] (ang. *Value Field*) (zawiera informacje tj: rozdzielczość obrazu, nazwisko pacjenta)



Rysunek 2.7: Schemat budowy strumienia informacji. Strumień informacji składa się z podstawowych elementów danych. Te natomiast składają się z identyfikatora, typu danych, rozmiaru danych, informacji.[24]

Uporządkowane wg identyfikatora strumień elementów danych (ang. *Data Elements*) to strumień informacyjny (ang. *Data Set*). Kilka strumieni informacyjnych tworzą *Item*. Te z kolei tworzą *Sequence of Items*. [40]

Każdy atrybut posiada pole *Value Multiplicity*, które podaje liczbę elementów występujących w atrybucie. Gdy w reprezentacji ciągu znaków kodowany jest więcej niż 1 element, następne są oddzielane znakiem "\" (backslash).[40]

Znaczniki opisują dane związane z danym badaniem. W jednym pliku może być wiele znaczników wykorzystywanych do interpretacji w przeglądarce DICOM. Znacznik składa się z 8 cyfr, w tym pierwsze cztery określają grupę, a pozostałe element. Najczęściej stosowane znaczniki zostały przedstawione w poniższej tabeli:

[1] pole opcjonalne

Nazwa	Identyfikator	Opis
SpecificCharacterSet	(0008,0005)	Używana specyfikacja
InstitutionName	(0008,0080)	Miejsce wykonywania badania
Manufacturer	(0008,0070)	Producent aplikacji
StationName	(0008,1010)	Nazwa urządzenia wykonującego badanie
PatientID	(0010,0020)	Identyfikator pacjenta
PatientsName	(0010,0010)	Nazwisko pacjenta
PatientsBirthDate	(0010,0030)	Data urodzin pacjenta
PatientsSex	(0010,0040)	Płeć pacjenta
PatientsAge	(0010,1010)	Wiek pacjenta
BodyPartExamined	(0018,0015)	Badana część ciała
StudyDate	(0008,0020)	Data badania
PhotometricInterpretation	(0028,0004)	Format zapisu obrazu
Rows	(0028,0010)	Wysokość zdjęcia (wyrażona w pikselach)
Columns	(0028,0011)	Szerokość zdjęcia (wyrażona w pikselach)

Tabela 2.1: Najczęściej stosowane znaczniki. Tabela zawiera kolumny z nazwą, identyfikatorem i opisem.

Kolejnym polem wchodzącym w skład podstawowej jednostki danych jest Typ Danych. Poniżej zostały przedstawione wykorzystywane typy danych z krótkim opisem wraz z ich rozmiarem.

Nazwa	Opis	Rozmiar
AE	Application Entity	16 Bytes Maximum
AS	Age String	4 Bytes Fixed
AT	Attribute Tag	4 Bytes Fixed
CS	Code String	16 Bytes Maximum
DA	Date	8 Bytes Fixed
DS	Decimal String	16 Bytes Maximum
DT	Date Time	26 Bytes Maximum
FL	Floating Point Single	4 Bytes Fixed
FD	Floating Point Double	8 Bytes Fixed
IS	Integer String	12 Bytes Maximum
LO	Long String	64 Bytes Maximum
LT	Long Text	10240 Bytes Maximum
OB	Other Byte String	Unlimited
OF	Other Float String	Unlimited
OW	Other Word String	Unlimited
PN	Person Name	64 Bytes Maximum
SH	Short String	16 Bytes Maximum
SL	Signed Long	4 Bytes Fixed
SQ	Sequence of Items	Unlimited
SS	Signed Short	2 Bytes Fixed
ST	Short Text	1024 Bytes Maximum
TM	Time	16 Bytes Maximum
UI	Unique Identifier	64 Bytes Maximum
UL	Unsigned Long	4 Bytes Fixed
UN	Unknown	Unlimited
US	Unsigned Short	2 Bytes Fixed
UT Unlimited	Text	Unlimited

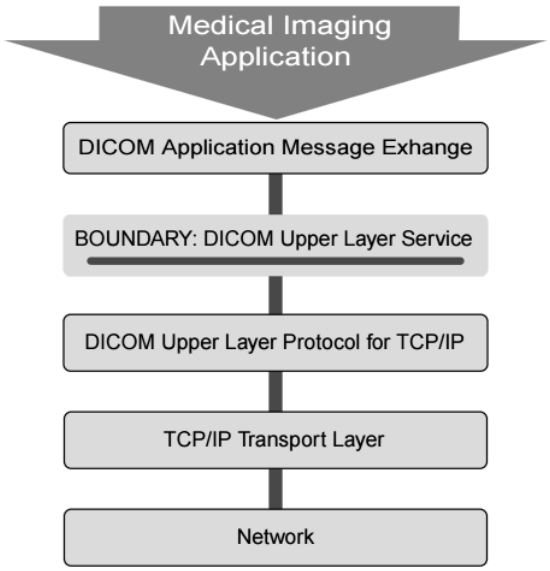
Tabela 2.2: Typy danych wraz z opisem oraz rozmiarem.

Objętość danych obrazu można zmniejszyć stosując kompresje do następujących formatów: Runlength encoding (RLE), JPEG, JPEG 2000, JPEG Lossless. Możliwa, ale rzadko stosowana jest kompresja całego pliku. [40]

2.2.5. Model komunikacji DICOM

Oparty jest na modelu odniesienia łączenia systemów otwartych OSI (ang. *OSI Basic Reference Model*), czyli modelu odniesienia łączenia systemów otwartych używanym jako model połączeń sprzętu obrazowania medycznego. Jest to standard określający strukturę komunikacji sieciowej. Warstwy modelu OSI mapują się na warstwy aplikacji oraz transportowej w modelu TCP/IP. Warstwy protokołów komunikacyjnych zostały przedstawione na poniższym schemacie.

Rysunek 2.8: Schemat przedstawiający różnice pomiędzy modelem OSI, a TCP/IP. Model OSI ma 7 warstw: warstwy wyższe [warstwa aplikacji (ang. *application layer*), warstwa prezentacji (ang. *presentation layer*), warstwa sesji (ang. *session layer*)], warstwy niższe [warstwa transportowa (ang. *transport layer*), warstwa sieciowa (ang. *network layer*), warstwa łącza danych (ang. *data link layer*), warstwa fizyczna (ang. *physical layer*)]. Model TCP/IP składa się z 4 warstw: warstwa aplikacji czy tzw. procesorowa (ang. *process layer*), warstwa transportowa (ang. *host-to-host layer*), warstwa protokołu internetowego czy inaczej warstwa internetu (ang. *internet protocol layer*), warstwa dostępu do sieci bądź warstwa fizyczna (ang. *network access layer*). [47]



Rysunek 2.9: Architektura protokołu sieci DICOM.[41]

Protokół wyższych warstw DICOM łączy protokoły wyższych warstw OSI w jeden łatwy do stosowania protokół, który dostarcza tych samych funkcji co protokoły OSI. [41] Warstwy niższe zapewniane są przez internetowy protokół TCP/IP (ang. *Transmission Control Protocol/Internet Protocol*). Granicę pomiędzy warstwą aplikacji, a warstwami niższymi stanowi warstwa prezentacji rozszerzona o mechanizm ustanawiania połączenia ACSE (element usługowy sterujący Powiązaniami, ang. *Association Control Service Element*). Jej zadaniem jest izolacja warstwy aplikacji. Zezwala aplikacjom ustanawiać Powiązania (ang. *Association*), przesyłać wiadomości oraz zrywać połączenia.[41]

Transmisja danych została podzielona na wymianę klas SOP. Model komunikacji zakłada role: nadawcy i odbiorcy w zakresie wymiany klasy SOP. Usługobiorca - SCU (ang. *Service Class User*) wysyła zapytania, a usługodawca - SCP (ang. *Service Class Provider*) jest dostawcą usługi odpowiadającym na zapytania SCU. [41]

Po pomyślnym wynegocjowaniu Powiązania przy użyciu komunikatów ACSE możliwa jest faktyczna wymiana danych, np. gdy co najmniej jeden kontekst prezentacji (ang. *Presentation Contexts*) został zaakceptowany przez SCP. Może to obejmować wysyłanie do SCP jednego lub więcej obrazów tomografii komputerowej, zadań drukowania czy zapytań listy roboczej (ang. *worklist queries*).

Jednostki aplikacyjne DICOM używają usług dostarczonych przez elementy usługowe wiadomości DICOM (tzw. DIMSE- ang. *DICOM Message Service Element*). DIMSE wyszczególniają dwa zbiory usług:

1. DIMSE-C wspiera operacje powiązane ze złożonymi klasami SOP oraz zapewnia skuteczną zgodność z poprzednimi wersjami standardu DICOM. Dane złożone zawierają informacje o pacjencie oraz o przeprowadzonym badaniu.

2. DIMSE-N wspiera operacje powiązane ze znormalizowanymi klasami SOP oraz zapew-

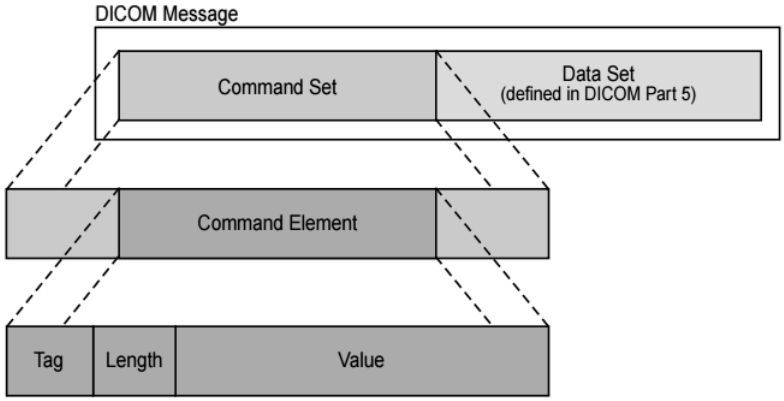
nia rozszerzony zbiór operacji i powiadomień zorientowanych obiektowo. Opiera się o model zarządzania systemami OSI (ang. *OSI System Management Model*), a szczególnie o definicję CMIS (ang. *OSI Common Management Information Services*). Dane znormalizowane zawierają jedynie jednostkowe informacje, takie jak nazwisko pacjenta lub czy jego wiek.

2.2.6. Struktura wiadomości DICOM

Wiadomości DICOM (ang. *DICOM message*) są używane do przenoszenia informacji przez sieć DICOM. Każda wiadomość DICOM jest zestawem poleceń DICOM (ang. *DICOM Command Set*) a za nią może wystąpić warunkowy zbiór danych DICOM (ang. *DICOM Data Set*). [41]

Zestaw poleceń wymienia operacje i/lub powiadomienia, których wykonania na zbiorze danych żąda nadawca od adresata. Jednostka aplikacji - AE (ang. *Application Entity*) to część procesu aplikacji zawierająca funkcje komunikacyjne OSI. [41]

Zestaw poleceń składa się się z elementów poleceń (ang. *Command Elements*), które zawierają zakodowane wartości dla zdefiniowanych pól zestawu poleceń. Każdy element w zestawie poleceń składa się z trzech pół: znacznika (ang. *Tag*), rozmiaru elementu (ang. *Value Length*) i informacji (ang. *Value Field*).[41]



Rysunek 2.10: Struktura wiadomości DICOM.[41]

W zestawie poleceń, elementy są uporządkowane rosnąco wg numeru znacznika. Każdy element jest identyfikowany unikalnym znacznikiem pojawiającym się tylko raz w zestawie poleceń.[41] Wymagania związane z zestawami poleceń i ich elementów zostały opisane w protokole DIMSE (patrz: 2.2.7). [41]

2.2.7. DIMSE-C i DIMSE-N

Komunikacja pomiędzy aplikacjami DICOM odbywa się przez tzw. usługi wiadomości żądające bądź dostarczające informacje stąd nazwa DIMSE na wszystkie polecenia usług.

Usługi DIMSE-C i DIMSE-N są wspierane przez pojedynczy protokół DIMSE, który używa specjalnego formatowania i szyfrowania wiadomości DICOM. [41]

Powiązanie jest terminem DICOM dla połączenia sieciowego między dwiema aplikacjami, z których jedna to SCU – klient, a druga to SCP – serwer. Jest to w zasadzie połączenie TCP, ale z rozległą negocjacją między dwoma jednostkami (ang. entities), aby uzgodnić, które SOP klasy i opcje oboje obsługują. [41]

Ponieważ Powiązanie DICOM jest połączeniem sieciowym, można na nim wykonać dowolną liczbę niezależnych operacji. Operacje te są całkowicie niezależne i potencjalnie zupełnie innego typu. [41]

Istnieje kilka różnych typów komunikatów DIMSE, z których każdy jest dostępny jako żądanie i jako wiadomość odpowiedzi:

1. Operacja C-STORE (nazywana inaczej "DICOM Push") jest przeznaczona do zapisywania na wskazanym węźle sieci DICOM instancji SOP (pary serwis-obiekt). Obowiązkowymi parametrami są: adres węzła lub jego adres IP, port węzła, tytułu aplikacji (AET – ang. *Application Entity Title*) klienta i serwera, nazwa pliku DICOM do wysłania. Pozwala SCU wysłać parametr instancji złożonych SOP (ang. *Composite SOP Instance*) do SCP. Na przykład służy do wysyłania obrazów z modalności do serwera PACS (ang. *Picture Archiving and Storage System*) lub tworzenia mechanizmu dostarczania dla C-MOVE.

   Najczęściej zapisywane jest właśnie na serwerze PACS. Komunikacja aplikacji z systemem PACS zachodzi na poziomie protokołu TCP/IP. Aby nawiązać połączenie z serwerem potrzebna jest znajomość tytułu aplikacji, niezmienny adres IP (do danego tytułu aplikacji przypisany w bazie danych PACS) oraz wolny port gdzie będą przesyłane dane.

   W komunikacji z PACS można wyszczególnić dwa podmioty. Pierwszy, który żąda danej usługi SCU. Wysyła zapytania oraz otrzymuje odpowiedzi. Natomiast drugi dostarcza usługę SCP, czyli wykonuje zadania takie jak: przyjmuje konkretne dane i zapisuje je na dysku.

2. Operacje C-MOVE służą do pobierania z serwera PACS obiektów DICOM. Pobieranie danych odbywa się za pośrednictwem dwóch wątków. Pierwszy, przesyłający dane przeprowadza nasłuch na konkretnym porcie, natomiast drugi przez usługę C-MOVE przesyła żądanie transportu plików do serwera. Wątek nasłuchowy zostanie zamknięty gdy operacja przesłania plików odbędzie się pomyślnie. W ten sposób w czasie długotrwałego przesyłania danych zapewniany jest niezakłócony dostęp do aplikacji.

   Nazwa pliku, która jest pobierana z serweru PACS zawiera dwie części: modalność oraz identyfikator instancji obiektu DICOM. Obowiązkowymi parametrami są: adres węzła lub jego adres IP, port węzła i odbiorcy, AET klienta i serwera oraz odbiorcy, obiekt kryteriów wyszukiwania.

   W przypadku, gdy AE chce zażądać otrzymania jednego lub więcej obrazów do przechowywania, może użyć do tego celu operacji albo C-GET albo C-MOVE. Oczekuje się,

że w większości środowisk C-MOVE jest prostszym rozwiązaniem, mimo że wymagane są dwa Powiązania. Korzystanie z usługi C-GET może nie być szeroko stosowane. Może być implementowany w szczególnych przypadkach, gdy system nie obsługuje wielu Powiązań.

3. Operacje C-GET są podobne do C-MOVE, ale zamiast drugiego Powiązania, żądana jest wysyłana przez oryginalne Powiązanie.

   Operacje C-GET są wywołane przez użytkownika w celu uzyskania informacji dla jednego lub więcej Composite SOP Instance od równorzędnego użytkownika, w oparciu o Atrybuty przesyłane przez użytkownika.

   Główne różnice pomiędzy operacjami C-GET a C-MOVE:

   - Sub-operacje C-STORE wynikające z C-GET są wykonywane na tym samym Powiązaniu co C-GET. Przy C-MOVE wynikowe pod-operacje C-STORE są wykonywane na oddzielnym Powiązaniu.
   - Operacja C-MOVE obsługuje operacje podrzędne C-STORE przeprowadzane przez jednostkę AE będącą stroną trzecią, czyli takiej która nie zainicjowała operacji C-MOVE.

4. Operacja C-FIND jest przeznaczona do wyszukiwania w bazie obiektów DICOM serwera PACS listy plików spełniających odpowiednie kryteria. Innymi słowy operacja C-FIND jest wywołana przez użytkownika w celu porównania zbioru Atrybutów z Atrybutami Instancja SOP (ang. *SOP Instances*) zarządzanymi przez równorzędnego użytkownika. Operacja C-FIND zwraca dla każdego pasującego ciągu Atrybutów listę wymaganych Atrybutów i ich wartości.

   Obiekt kryteriów wyszukiwania, który jest obiektem przechowującym pola, zawiera takie pola jak: poziom przeszukiwania (pacjent, badanie, seria) i listę pól. Obowiązkowo podawana jest wartość tagu (0008, 0052) QueryRetrieveLevel mogąca przyjmować wartości PATIENT, STUDY, SERIES, IMAGE. Obowiązkowymi parametrami są: adres węzła lub jego adres IP, port węzła, AET klienta i serwera, obiekt kryteriów wyszukiwania.

   Używany w usługach Modality-Worklist i General-Purpose-Worklist, jest to bardzo prosta operacja, podobna do zapytania SQL, w której zbiór danych jest przekazywany z SCU do SCP zawierającego 2 rodzaje atrybutu:

   - Te, które muszą być dopasowane (równoważne klauzuli WHERE SQL). Te mają "wypełnione" wartości.
   - Te, które mają zostać zwrócone do SCU (odpowiednik klauzuli SELECT SQL). Są wysyłane puste pola.

   SCP odpowiada, wysyłając kilka pasujących zestawów danych, a następnie "kompletną" odpowiedź, aby powiedzieć, że została zakończona.

   Rezultaty zapytania, które zostały przesłane przez serwer, są listą wszystkich odpowiedzi, które spełniły zadane wcześniej kryteria. Użytkownik aplikacji ma możliwość pobrania konkretnych odpowiedzi - plików do lokalnego katalogu i analizowania ich zawartości.

5. Operacja C-ECHO przeznaczona jest do weryfikacji poprawności połączenia pomiędzy użytkownikami. Obowiązkowymi parametrami są: adres węzła lub jego adres IP, port węzła oraz AET serwera i klienta. AET jest nazwa identyfikująca węzeł sieci DICOM. Wsparcie dla C-ECHO jest obowiązkowe dla wszystkich AE, które akceptują Powiązania.

   Typy komunikacji usługi DIMSE-N:

6. Operacja N-EVENT-REPORT jest wysyłana z SCP do SCU. Używana w usługach takich jak, np. Print management.

7. Operacja N-GET jest wywołana przez użytkownika żeby zażądać uzyskania informacji przez równorzędnego użytkownika. Operacje N-GET żądają pojedynczego zestawu danych i wymagają, aby identyfikator instancji-UID zestawu danych został określony w żądaniu.

8. Usługa N-SET jest wywoływana przez użytkownika w celu żądania modyfikacji informacji przez równorzędnego użytkownika. Operacje N-SET aktualizują pojedynczy zestaw danych i wymagają aby identyfikator instancji-UID zestawu danych został określony w żądaniu. Służy do aktualizacji statusu badania w usłudze Modality-Performed-Procedure-Step-Service

9. Operacja N-ACTION jest wywoływana przez użytkownika żeby zażądać od równorzędnego użytkownika wykonania akcji. Używana w serwisie Print management

10. Operacje N-CREATE powodują utworzenie zestawu danych SCP do późniejszego wykorzystania. Identyfikator instancji-UID może być określony przez SCU lub jeśli SCP pozostawia puste pole. Operacja N-CREATE jest wywołana przez użytkownika żeby zażądać od równorzędnego użytkownika stworzenia instancji klasy SOP.

11. Operacja N-DELETE żąda od SCP o usunięcie określonego obiektu. Operacja N-DELETE jest wywoływana przez użytkownika żeby zażądać od równorzędnego użytkownika usunięcie instancji klasy SOP. [41]

2.2.8. Anonimizacja danych

Rozszerzenie standardu DICOM o suplement Supp 142 [41] definiuje profile danych odpowiedzialne za poprawną anonimizację (patrz: 2.5). Profile te zostały utworzone w celu zapewnienia równowagi między usuwaniem części informacji, a potrzebą zatrzymywania innych informacji, które są konieczne, aby zbiory danych pozostały użyteczne zgodnie z ich przeznaczeniem. Profile mogą zostać rozszerzone poprzez zastosowanie predefiniowanych opcji. Opcje dla profili są zdefiniowane dla konkretnych aplikacji. Opcje te mogą określać, czy dodatkowe Atrybuty zostaną usunięte lub zastąpione, lub czy zachowane będą Atrybuty, które przeznaczone były do usunięcia lub do zastąpienia.[45, 3]

Lista usuwanych czy zamienianych informacji zwiększających ryzyko identyfikacji pacjenta przez stronę do tego niepowołaną to m.in.: wszystkie nazwiska i identyfikatory osób zarówno pacjentów jak i personelu, wszystkie instytucje, wydziały, modele sprzętu, wszystkie komentarze tekstowe i opisy, wszystkie UID (ang. *Unique Identifier*). [45, 3]

Wymagane jest zachowanie integralności obiektu w odniesieniu do zgodności z DICOM, gdy rozważane jest usunięcie bądź zamienienie informacji. Wyszczególnia się trzy typy działań:

- typ 1 charakteryzuje się zamienianiem w losowe wartości (ang. *dummy value)*
- typ 2 to pusta zmienna o nie przypisanej wartości
- typ 3 to kompletne usunięcie

Predefiniowane opcje rozszerzające profile zostały podzielone na dwie grupy:

1. Usunąć więcej (ang. *remove more*) jest to rozwiązanie nie używane w podstawowym profilu, ponieważ jest trudne w implementacji i zazwyczaj niepotrzebne. Zależy od określonego typu obiektu i nie dotyczy plików graficznych.
    - *Clean Pixel Data Option*
    - *Clean Recognizable Visual Features Option*
    - *Clean Graphics Option*
    - *Clean Structured Content Option*
    - *Clean Descriptors Option*
2. Zachować więcej (ang. *retain more; remove less*) powoduje niewielkie ryzyko ponownej identyfikacji pacjenta.
    - *Retain Longitudinal Option*
    - *Retain Patient Characteristics Option*
    - *Retain Device Information Option*
    - *Retain UIDs*
    - *Retain Safe Private Option*

2.2.9. DICOMweb

Standard DICOMweb jest rozszerzeniem standardu DICOM określającego w jaki sposób uzyskać dostęp do zawartości zdalnego serwera DICOM za pośrednictwem protokołu HTTP(S). Jest to zbiór serwisów o architekturze REST (ang. *Representational State Transfer*), umożliwiających programistom stron internetowych na używanie obrazów medycznych przy pomocy standardowych zestawów narzędzi. DICOMweb może zostać zaimplementowany bezpośrednio lub jako proxy dla usług DIMSE, aby zapewnić nowoczesny dostęp do systemów obsługujących DICOM przez Internet. Obrazy przedstawiające wyniki różnych metod diagnostycznych np. tomografii komputerowej, ultrasonografii itp. nie zawsze muszą być przekształcane, aby być kompatybilne z DICOMweb. [43]

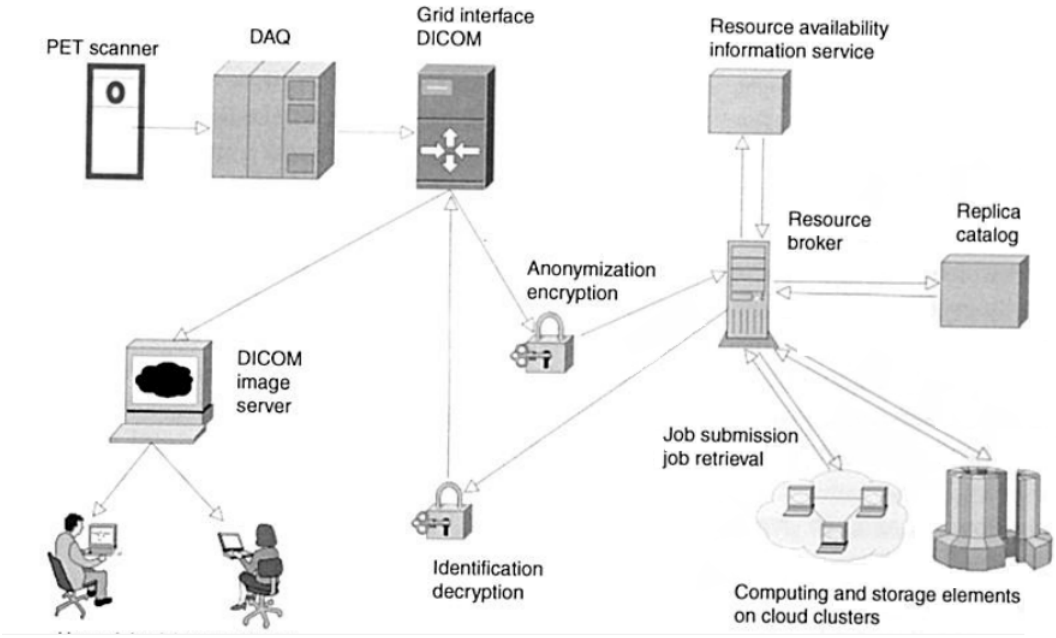
Query	szukanie obiektów DICOM (QIDO-RS)
Retrieve	uzyskiwanie obiektów DICOM (WADO-RS)
Store	przechowywanie obiektów DICOM (STOW-RS)
Worklist	zarządzanie elementami listy zadań (ang. worklist) (UPS-RS)
Capabilities	opis wspierających serwisów

Tabela 2.3: Dostępne serwisy DICOMweb.[43]

Głównym zastosowaniem DICOMweb jest interakcja z obrazami medycznymi: szukanie obrazów (c-find), dodawanie nowych obrazów, zapisywanie obrazów. Anonimizacja jest opcjonalnym parametrem w WADO-URI tylko dla obiektów. Np. W momencie pobierania obrazu przez DICOMweb możliwa jest anonimizacja przez serwer obrazów PACS. Nie ma możliwości anonimizacji w zarządzaniu z worklistami.

2.2.10. Inne koncepcje zdalnego przetwarzania dokumentów DICOM

W literaturze rozważane jest zdalne przetwarzanie danych medycznych z wykorzystaniem architektury grid. Dane medyczne są poufne i wymagają specjalnych zabezpieczeń. Oprócz zwykłych środków bezpieczeństwa stosowanych w gridzie dla danych przesyłanych ze strony domowej skanera, rzeczywiste dane medyczne powinny być dodatkowo anonimizowane i szyfrowane w zakresie części wrażliwych, natomiast klucz do identyfikacji powinien być przechowywany w zabezpieczonej pamięci. Możliwe jest, aby dedykowany interfejs DICOM - Grid w tym celu wykorzystał tą funkcjonalność. [17] Projekt przepływu danych został zilustrowany poniżej:



Rysunek 2.11: Konstrukcja modelu przepływu danych dla szybkich medycznych rekonstrukcji obrazu w gridzie. [17]

Połączenie oryginalnego gridu i technologii chmurowej zapewnia optymalne wykorzystanie zasobów. Na poziomie całej sieci, efekt ten jest osiągany przez zarządce zasobów (ang. *resource brokers*), a na poziomie poszczególnych urządzeń dzięki stosowaniu technik wirtualizacji.[17]

2.3. API typu REST

API (ang. *Application Programming Interface*) to interfejs programistyczny umożliwiający komunikację aplikacji między sobą, np. stworzonego programu z systemem operacyjnym. Pozwala na używanie dostępnych funkcjonalności serwisów, systemów operacyjnych, bibliotek, etc. Interfejs programistyczny jest zdefiniowany na poziomie kodu źródłowego. API jest odpowiedzialny za dostarczanie właściwych klas obiektów, które definiują obiekty w języku programowania, struktur danych, podprogramów, wymaganych protokołów komunikacyjnych, np. HTTP. API ma zastosowanie m.in. w aplikacjach sieciowych. [52]

REST, czyli zmiana stanu poprzez reprezentacje, jest wzorcem do tworzenia architektury dla aplikacji rozproszonych za pomocą protokołu komunikacyjnego HTTP. Innymi słowy jest to styl architektury bądź wzorzec projektowy wykorzystywany jako zbiór wytycznych używanych do tworzenia web serwisów, które pozwalają na komunikację między aplikacjami czy urządzeniami podłączonymi do internetu przez współdzielony protokół HTTP. Określa jak używać danego protokołu, żeby mieć dostęp do zasobów, które są głównym elementem RESTa. [51] API typu REST jest naturalnym wyborem dla aplikacji, których zadaniem jest zdalne przetwarzanie danych np.: w chmurze. Stanowi też on podstawę standardu DICOMweb.

Aby możliwe było wykonanie zapytania (ang. request) wymagany jest adres URL. [51]. W REST każda usługa jest opisana osobnym adresem URL. Użytkownik przy użyciu przeglądarki - klienta wprowadza właściwy adres URL oraz wraz z wykorzystaniem protokołu HTTP (lub HTTPS) przesyła zapytanie HTTP, czyli odpytuje serwer. Natomiast serwer, gdy wysłane dane są poprawne, zwraca oczekiwaną odpowiedź HTTP z właściwym statusem - kodem odpowiedzi HTTP. [35]

Rysunek 2.12: Schemat komunikacji pomiędzy klientem, a serwerem poprzez protokół HTTP. [35]

REST API stanowi interfejs programistyczny aplikacji, który jest oparty o protokół komunikacyjny HTTP, który natomiast umożliwia zarządzanie zasobami z pomocą operacji takich jak pobieranie-GET, tworzenie-POST, edycja-PUT, usuwanie-DELETE. Każdej z tych operacji odpowiada żądanie HTTP zawierające dane w formacie JSON (ang. *JavaScript Object Notation*) czy XML (ang. *Extensible Markup Language*). Po wysłaniu żądania HTTP, klient aplikacji otrzymuje odpowiedź w formacie JSON lub XML wraz z właściwym kodem HTTP. [51] W tabeli 2.4 zebrano metody HTTP wraz z opisem.

Metoda	Opis
GET	pobieranie zasobu, nie zmienia stanu aplikacji
PUT	aktualizowanie całości zasobu (ID zdefiniowane przez klienta)
PATCH	częściowa aktualizacja zasobu
POST	dodawanie nowego zasobu (nie zdefiniowane ID)
DELETE	usuwanie zasobu

Tabela 2.4: Metody HTTP wraz z opisem działania

Przykładowe kody statusów HTTP jakie serwer wysyła do klienta zostały przedstawione w tabeli 2.5.

Kod	Nazwa	Opis
200	OK	operacja zakończona pomyślnie
201	CREATED	utworzono nowy zasób po użyciu metody post
204	NO CONTENT	jest wysyłany po użyciu PUT, PATCH lub DELETE
401	UNAUTHORIZED	wymagane uwierzytelnienie
403	FORBIDDEN	wymagana autoryzacja
500	INTERNAL SERVER ERROR	błąd serwera

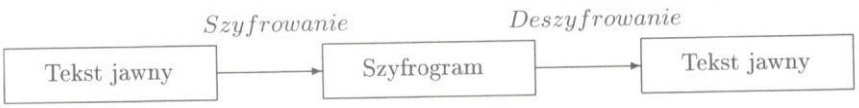
Tabela 2.5: Kody statusów HTTP z kodem, nazwą i opisem

2.4. Kryptografia jako element ochrony danych medycznych

2.4.1. Podstawy kryptografii

Kryptografia jest nauką zajmującą się metodami zabezpieczania informacji. Zaś kryptoanalizą nazywany działania związane ze złamaniem szyfru. Natomiast kryptologia obejmuje zarówno kryptologię jak i kryptoanalizę. [30]

Wg Denning kryptografia zajmuje się utajnionym zapisywaniem. Natomiast metodą utajnionego zapisywania jest szyfr. Dzięki niemu możliwe jest przekształcenie tekstu jawnego (zwanego również jako tekstu otwartego) na tekst zaszyfrowany (tzw. kryptogram). Takie przekształcenie zwane jest szyfrowanie bądź utajnianiem. Deszyfrowanie jest procesem odwrotnym, czyli polega na przekształceniu tekstu zaszyfrowanego na tekst jawny. [5]



Rysunek 2.13: Schemat szyfrowania i deszyfrowania. [4]

Kryptografia klasyczna a współczesna

Kryptografię można podzielić na klasyczną i współczesną. Za datę graniczną przyjmuje się rok 1975. Przed nią wszelkie algorytmy oraz metody szyfrowania zaliczane były do kryptografii klasycznej, czyli obejmującej szyfry podstawieniowe: monoalfabetowe, homofoniczne,

wieloalfabetowe, poligramowe. Natomiast po roku 1975 rozpoczął się okres kryptografii współczesnej, która związana jest z algorytmami DES (ang. *Data Encryption Standard*), i szyframi z kluczem publicznym. [30]

Kryptografia wykorzystywana jest do ochrony informacji, np. w przesyłaniu danych, w trakcie dokonywania transakcji, logowania się, podawania haseł. Algorytmy jej są używane w telefonii komórkowej, w systemach bankowych (bankowe protokoły obrotu finansowego), w ochronie niejawnych danych tj. planów wojskowych, personalnych danych medyczych, tajnych techonologii. [30]

Kroptosystemy symetryczne i asymetryczne

Wg Bruce'a Schneiera dobre systemy kryptograficzne charakteryzują się tajnym kluczem i jawnym algorytmem.

W systemach symetrycznych (zwanych również jako jednokluczowe) klucze do szyfrowania i deszyfrowania są takie same bądź w łatwy sposób można otrzymać jeden na podstawie drugiego. Wadą kryptografii symetrycznej jest potrzeba ustalenia klucza pomiędzy nadawcą a odbiorcą przed rozpoczęciem wymiany danych. Jeśli wspomniana wymiana miałaby odbywać się za pośrednictwem sieci komputerowych istnieje niezerowe prawdopodobieństwo przechwycenia klucza. A to z kolei może pozwolić na deszyfrację szyfrogramu przez niepożądane osoby. Natomiast dobrą stroną kryptografii jednokluczowej jest szybkość procesu. [5]

Rysunek 2.14: Schemat szyfrowania i deszyfrowania w systemach symetrycznych. [4]

Przykładami zastosowania szyfrów symetrycznych są:

1. Algorytm DES, czyli standard szyfrowania danych, który został opracowany w latach 80 XX wieku przez Horsta Feistela oraz Dana Coppersmitha. [4]

   Algorytm DES jest szyfrem blokowym. W trakcie szyfrowania dane wejściowe zostają podzielone na 64-bitowe bloki z których każdy zostaje zaszyfrowany z wykorzystaniem niejawnego klucza o długości 64-bit. Przy czym algorytm wykorzystuje jedynie 56 bitów klucza pozostałe stosowane są do jedynie w procesie weryfikacji szyfrogramu. Na jednym końcu algorytmu jest blok 64-bitów tekstu otwartego, a po drugiej stronie wychodzi 64-bitowy szyfrogram. Każda liczba o długości 64 bitów, którą można zmienić w dowolnej chwili, może być kluczem. Całe bezpieczeństwo algorytmu tkwi w nieznajomości klucza.[5] [38] Algorytm DES obsługuje zarówno deszyfrowanie i szyfrowanie, jedyna różnica pomiędzy tymi czynnościami to kolejność używania klucza. [38] [30] Obecnie algorytm istotny jest tylko z punktu widzenia historycznego, gdyż nie jest już uznawany za wystarczająco bezpieczny.

2. Algorytm AES (ang. *Advanced Encryption Standard*) stworzony przez Joana Daemena i Vincenta Rijmena. AES został zaprojektowany tak, aby umożliwiał wydajne implementacje zarówno sprzętowe jak i w postaci oprogramowania. Tak jak DES jest to szyfr blokowy. Standard AES to jedna z najpopularniejszych metod szyfrowania i odszyfrowywania poufnych informacji w 2017 roku.[4] Jest używany w wielu protokołach, takich jak *Secure Sockets Layer (SSL) / Transport Layer Security (TLS)* i można go znaleźć w większości nowoczesnych aplikacji i urządzeń wymagających funkcji szyfrowania.[4]

   AES zawiera trzy szyfry blokowe: AES-128, AES-192 i AES-256. Każdy szyfr szyfruje i odszyfrowuje dane w blokach 128-bitowych za pomocą kluczy kryptograficznych odpowiednio 128, 192 i 256 bitów. Szyfr Rijndael (prekursor AES) został zaprojektowany, aby akceptować dodatkowe rozmiary bloków i długości kluczy, ale w przypadku AES funkcje te nie zostały przyjęte.[5]

   Algorytm szyfrowania AES definiuje liczbę transformacji, które mają zostać wykonane na danych przechowywanych w tablicy. Pierwszym krokiem szyfrowania jest umieszczenie danych w tablicy, po czym transformacje szyfrów powtarza się w wielu rundach szyfrowania. Liczba rund zależy od długości klucza, 10 rund dla kluczy 128-bitowych, 12 rund dla kluczy 192-bitowych i 14 rund dla kluczy 256-bitowych. Runda składa się z kilku etapów przetwarzania, które obejmują podstawianie, transpozycję i mieszanie wejściowego jawnego tekstu i przekształcanie go w końcowy wynik zaszyfrowanego tekstu.[4]

   Pierwsza transformacja w szyfrze szyfrującym AES polega na zamianie danych przy użyciu tabeli substytucji; druga transformacja przesuwa rzędy danych, trzecia miesza kolumny. Ostatnia transformacja to alternatywa rozłączna (XOR) wykonywana dla każdej kolumny przy użyciu innej części klucza szyfrowania - dłuższe klucze wymagają więcej rund.[5]

   Proces deszyfrowania szyfrogramu AES jest podobny do procesu szyfrowania w odwrotnej kolejności. Każda runda składa się z czterech procesów przeprowadzonych w odwrotnej kolejności: dodanie klucza rundy; mieszanie kolumn; zmienianie rzędów; substytucja bajtów.[4]

   Ponieważ podprocesy w każdej rundzie są odwrotne, w przeciwieństwie do algorytmu Feistela, algorytmy szyfrowania i deszyfrowania muszą być oddzielnie implementowane, chociaż są bardzo blisko powiązane.[5]

W systemach asymetrycznych (tzw. dwukluczowych) klucze do szyfrowania i deszyfrowania są różne i niezależne w kontekście wzajemnego wyznaczania. Za pomocą metod obliczeniowych na podstawie pierwszego klucza nie możliwe jest określenie drugiego klucza. [5]

Rysunek 2.15: Schemat szyfrowania i deszyfrowania w systemach asymetrycznych. [4]

Klucz publiczny to klucz szyfrujący, a klucz prywatny jest kluczem deszyfrującym. Dany tekst, który jest zaszyfrowany kluczem publicznym, można odszyfrować tylko przez odpowiadający mu klucz prywatny. W przypadku algorytmu RSA (Rivesta-Shamira-Adlemana) możliwe jest również szyfrowanie kluczem prywatnym (tworzenie podpisu elektronicznego), a deszyfracja opiera się na odpowiednim kluczu publicznym. [4]

Przykładami zastosowania szyfrów asymetrycznych są:

1. algorytm Diffiego-Hellmanna
   W latach 70 XX wieku został opisany przez Whitfielda Diffie'a i Martina Hellmana. Charakteryzują go dzielenie modulo i używanie dużych liczb pierwszych. Stosowany do bezpiecznej wymiany kluczy (klucza prywatnego i publicznego). Używany jest w wielu protokołach, np. w SSL/TLS. [4]

2. Algorytm RSA został opracowany w latach 80 XX wieku przez trzech profesorów: Rona Rivesta, Adiego Shamira oraz Leonarda Adlemana pracujących w MIT (ang. *Massachussets Institute of Technology*), którzy nazwali swój algorytm połączeniem pierwszych liter swoich nazwisk.
   Algorytm ten pozwala na szyfrowanie i deszyfrowanie z parą kluczy: jawnego i prywatnego. Bezpieczeństwo szyfrowania RSA jest zapewnione przez czasochłonność rozkładu na czynniki pierwsze (tzw. faktoryzacja) dużych liczb złożonych. RSA ma zastosowanie także do podpisów elektronicznych, gdyż charakteryzuje się tzw. własnością symetrii względem kluczy, czyli istnieje możliwość szyfrowania danych kluczem prywatnym i deszyfrowania kluczem publicznym.

3. Kryptografia krzywych eliptycznych (ang. *Elliptic Curve Cryptography* - ECC) opiera się na algebraicznej strukturze krzywych eliptycznych. Bezpieczeństwo tego rodzaju szyfrowania jest oparte na złożoności obliczeniowej dyskretnych logarytmów na krzywych eliptycznych ECDLP (ang. *Elliptic Curve Discrete Logarithm Problem*). Jest następną generacją kryptografii opartej na kluczu publicznym oraz obecnym poziomie wiedzy w dziedzinie matematyki.

   Metoda ECC wymaga transmisji mniejszej ilości danych, ze względu na krótsze klucze publiczne: na przykład bezpieczeństwo 256-bitowego klucza ECC jest porównywalne do 3072-bitowego klucza RSA. Zapewnia znacznie bezpieczniejszą podstawę niż pierwszej generacji systemy kryptografii klucza publicznego, tj. RSA.[22]

   Równanie krzywej eliptycznej opisywane jest jako: $y^2 = x^3 + ax + b \pmod{p}$.

Generowanie klucza jest ważną częścią, gdzie zarówno generowany jest klucz publiczny jak i prywatny. Adresat szyfruje wiadomość z kluczem publicznym odbiorcy i odbiorca deszyfruje wiadomość kluczem prywatnym.[22]

2.4.2. PKI - Infrastruktura Klucza Publicznego

Europejski Bank Centralny w decyzji 2013/132/UE z dnia 11 stycznia 2013 ustanawiającej ramy infrastruktury klucza publicznego Europejskiego Systemu Banków Centralnych (EBC/2013/1) opisuje PKI (ang. *Public Key Infrastructure*) poniższą definicją:
„zbiór osób, polityk, procedur i systemów komputerowych niezbędnych do świadczenia usług uwierzytelniania, szyfrowania, integralności i niezaprzeczalności za pośrednictwem kryptografii klucza publicznego i prywatnego i certyfikatów elektronicznych"

Funkcje PKI zostały wymienione poniżej:

1. Rejestracja (ang. *Registration*)
2. Certyfikacja (ang. *Certification*)
3. Generowanie kluczy (ang. *Key generation*)
4. Odnawianie kluczy (ang. *Key update*)
5. Certyfikacja wzajemna (ang. *Cross-certification*)
6. Odwoływanie certyfikatu (ang. *Revocation*)
7. Odzyskiwanie klucza (ang. *Key recovery*)

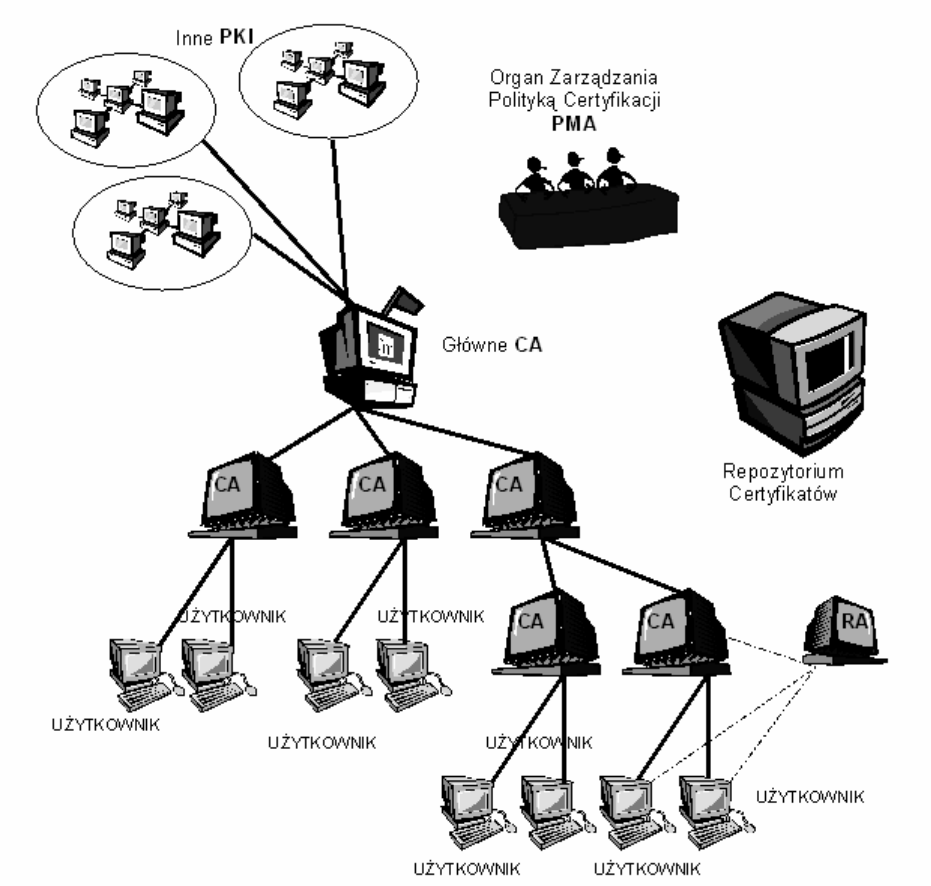
Infrastuktura klucza publicznego jest kryptosystemem składającym się z:

1. urzędów certyfikacyjnych (CA - ang. *Certificate Authority*) i rejestracyjnych (RA - ang. *Registration Authority*)
2. repozytoriów certyfikatów, kluczy i list unieważnionych certyfikatów (CRT - ang. *Certificate Revocation List*)
3. użytkowników (subskrybentów certyfikatów klucza publicznego), sprzętu i oprogramowania, tzw. użytkowników końcowych (ang. *End Entity*)

Struktura klucza publicznego została stworzona w oparciu o Główne CA określające ogólną politykę certyfikacji. Podległe Głównemu Urzędowi są CA, które mają za zadanie wydawanie cyfrowych certyfikatów (elektronicznych dokumentów służących do identyfikowania jednostki tj.: firmy, serwera, określonych osób oraz powiązania jednostki z konkretnym kluczem publicznym). Urzędem certyfikacyjnym może być niezależna organizacja bądź firma tj. Netscape Certificate Server. W zależności od zapotrzebowań konkretnej jednostki istnieją różne metody tworzenia i identyfikacji certyfikatów. Urzędowi certyfikacyjnemu może podlegać dowolna ilość urzędów certyfikacyjnych oraz użytkowników, dzięki czemu tworzona jest hierarchia uwierzytelniania określająca łańcuch certyfikatów. [34] Celem RA jest weryfikacja danych a następnie rejestracja użytkownika.[49]

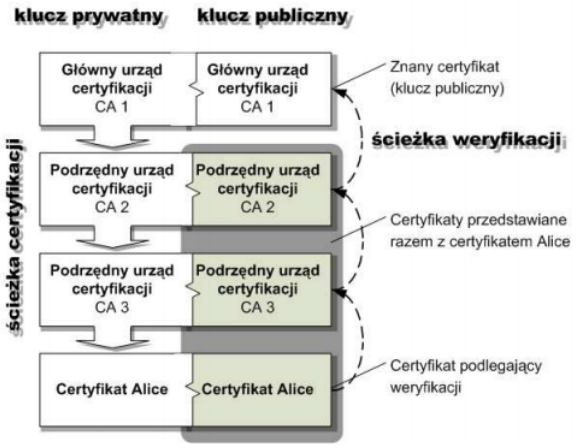
Lista CRL zawiera listę numerów seryjnych unieważnionych certyfikatów. Przyczyną unieważnienia certyfikatu przez CA przed datą jego wygaśnięcia może być zmiana adresu poczty

elektronicznej, nazwiska lub ujawnienie klucza prywatnego.[49]



Rysunek 2.16: Schemat komponentów PKI zawiera: inne PKI, organ Zarządzania Polityką Certyfikacji - PMA, Główne CA, urzędy rejestracyjne i certyfikacyjne, repozytorium certyfikatów oraz użytkowników. [34]

Aby uzyskać certyfikat klucza publicznego, aplikant wysyła wiadomość (blok zaszyfrowanego tekstu) do urzędu certyfikującego tzw. CSR (ang. Certificate Signing Request). W wiadomości tej zawarty jest klucz publiczny certyfikatu SSL, informacje dotyczące identyfikacji (np. nazwa organizacji, nazwa domeny, lokalizacja, kraj), ochrony integralności np. podpis elektroniczny (ang. *digital signature*). CSR zwykłe generowany jest na serwerze gdzie certyfikat będzie instalowany. CSR jest zakodowany przy pomocy reprezentacji ASN.1 wg. specyfikacji PKCS#10. Klucz prywatny tworzony jest w tym samym czasie co tworzenie CSR, tworząc parę kluczy. Urzędy certyfikacyjne do stworzenia certyfikatu SSL potrzebują CSR, ale nie potrzebują wygenerowanego klucza prywatnego. Aplikant musi trzymać swój klucz prywatny w tajemnicy. Certyfikat utworzony z konkretnego CSR pracuje tylko z kluczem prywatnym, który był wygenerowany z CSR. Jeśli aplikant zgubi klucz prywatny, certyfikat nie będzie dłużej działał. [37]



Rysunek 2.17: Ścieżka weryfikacji i certyfikacji.[37]

Uwierzytelnianiem w sieci jest wzajemna identyfikacja dwóch stron komunikacji. Przykładem może być rozpoznanie klienta danego serwera i/lub zapewnienie klienta, że połączenie odbywa się z właściwym serwerem. Uwierzytelnianie w sieci można realizować na podstawie certyfikatu. W trakcie sesji, określona jednostka przesyła zarówno dany komunikat ze swoim podpisem elektronicznym oraz certyfikat. Na tej podstawie odbiorca uwierzytelnia nadawcę.[34]

PKI kreuje hierarchiczną strukturę zaufania, gdzie certyfikat klucza publicznego jest fundamentalnym dokumentem. X.509 jest najpopularniejszym standardem certyfikatów PKI (wersja trzecia). Definiuje schemat dla:

1. listy unieważnionych certyfikatów
2. certyfikatów kluczy publicznych i certyfikatów atrybutu (zdefiniowanych w standardzie RFC 5755)

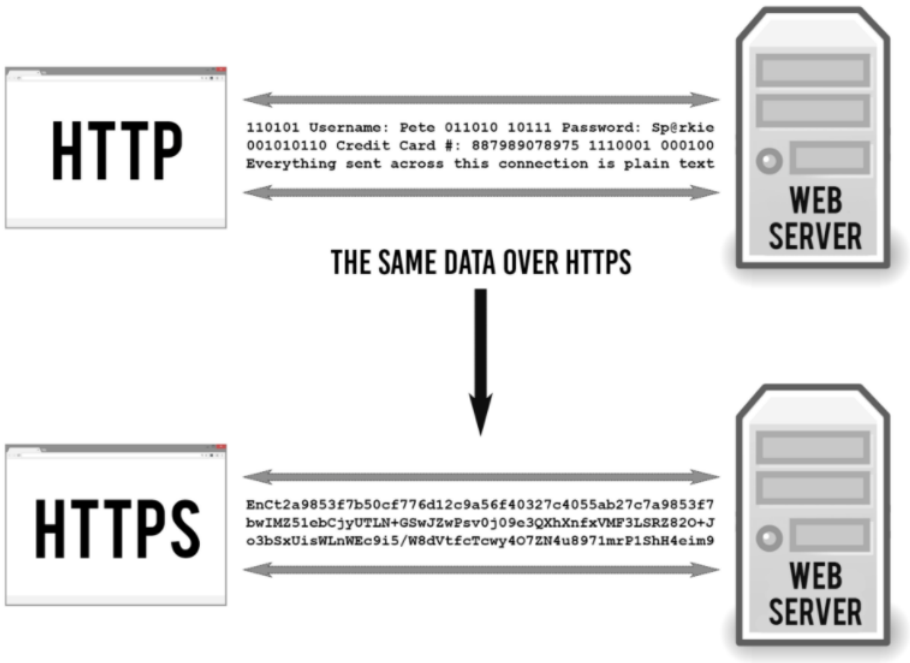
X.509 jest redagowany przez Sektor Normalizacji Telekomunikacji ITU (ang. *International Telecommunication Union - Telecommunication Standardization Sector*). X.509 jest wykorzystywany w TLS, który z kolei jest podstawą HTTPS. [34]

PKI ma szeroki wachlarz zastosowań, m.in. stosowana jest w: poczcie elektronicznej, handlu elektronicznym (transakcje e-commerce), administracji publicznej, sektorze bankowo-finansowym, wirtualnych sieciach prywatnych (tzw. VPN ang. *Virtual Private Network*), zarządzaniu zasobami przedsiębiorstw - systemy ERP. Umożliwia bezpieczeństwo w aplikacjach, witrynach internetowych, urządzeniach klienta. [34]

2.4.3. Protokół HTTPS

Protokół HTTPS (ang. *HyperText Transfer Protocol Secure*) jest internetowym protokołem komunikacji klienta z serwerem. Jest bezpieczniejszą, zaszyfrowaną wersją HTTP. Początkowo HTTPS do szyfrowania używał protokołu SSL. Obecnie stosuje się protokół TLS z trójstopniową ochroną danych, uważany jako rozwinięcie protokołu SSL. [37]

Protokół HTTPS domyślnie korzysta z portu nr 443 w protokole TCP, a wywołanie rozpoczyna się od "https://". Protokół HTTP domyślnie działa na porcie nr 80, a URL (ang. *Uniform Resource Locator*) ma na początku "http://". Protokół HTTP nie jest szyfrowany i jest wrażliwy na ataki kryptologiczne (ang. *man-in-the-middle*) i podsłuchy (ang. *eavesdropping attacks*). Protokół HTTPS został zaprojektowany do ochrony przed takimi atakami.[37]



Rysunek 2.18: Porównanie protokołu HTTP i HTTPS. [37]

Zasada działania typowego połączenia:

1. Klient wysyła wiadomość ClientHello określającą najnowszą obsługiwaną wersję protokołu TLS, losową liczbę stosowaną przy generowaniu kluczy, listę obsługiwanych algorytmów kryptograficznych (ang. *cipher suite*) i sugerowanych metod kompresji.

2. Serwer odpowiada wiadomością ServerHello, zawierającą wybraną wersję protokołu, losową liczbę, wybrany algorytm kryptograficzny i metodę kompresji spośród opcji podanych przez klienta.

3. Serwer wysyła wiadomość Certyfikacyjną (ang. *Certificate*) - zawierającą certyfikat, który powinien być zweryfikowany przez klienta.

4. Serwer wysyła wiadomość ServerKeyExchange zawierającą informację na temat swojego klucza publicznego.

5. Serwer wysyła wiadomość ServerHelloDone o możliwości przejścia do kolejnej fazy.

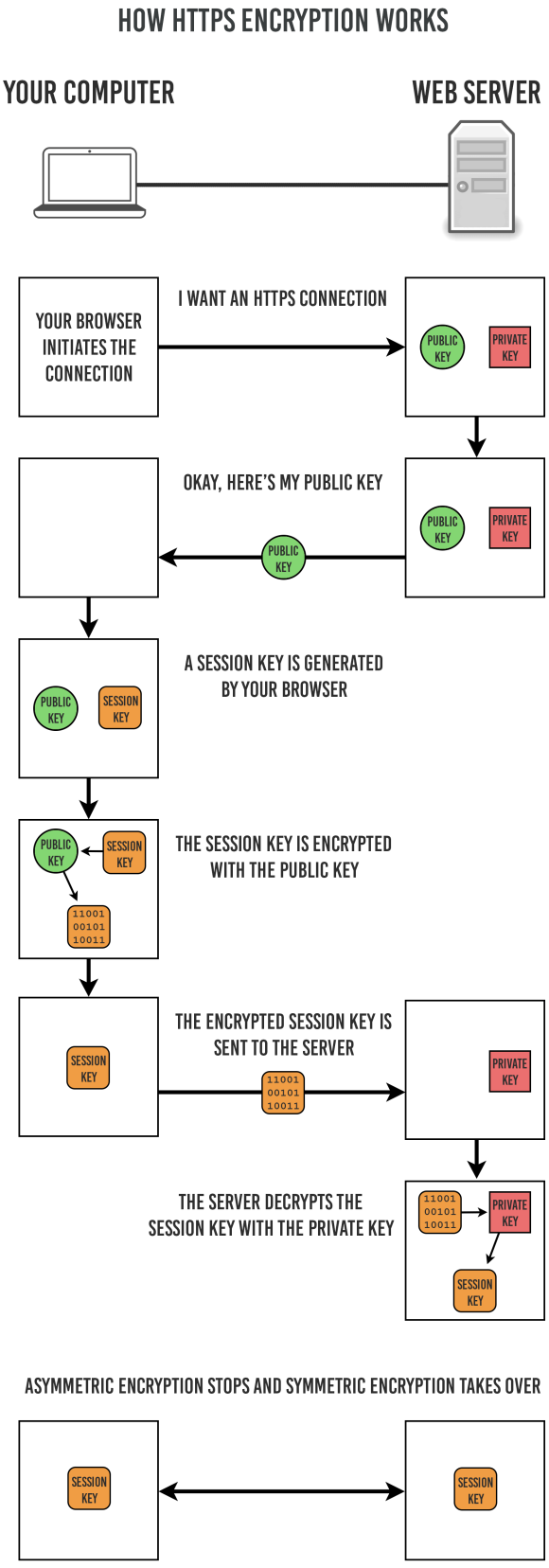
6. Klient odpowiada wiadomością ClientKeyExchange, która zawiera wstępny klucz sesji zaszyfrowany przy użyciu klucza publicznego. Przy pomocy wygenerowanych wcześniej dwóch liczb losowych (zarówno klienta jak i serwera) i określonego wstępnego klucza sesji, klient i serwer generują klucz sesji będący kluczem algorytmu symetrycznego.

7. Klient wysyła wiadomość ChangeCipherSpec o możliwości przełączenia na komunikację szyfrowaną w oparciu o klucz symetryczny.

8. Klient wysyła wiadomość Finished informującą o gotowości odbierania zaszyfrowanych danych.

9. Serwer wysyła informację ChangeCipherSpec, że od teraz będzie wysyłał zaszyfrowane dane.

10. Serwer wysyła wiadomość Finished w celu wypróbowania bezpiecznego kanału.



Rysunek 2.19: Schemat szyfrowania HTTPS.[37]

W celu uwierzytelniania klienta korzysta się z trzech komunikatów:

1. Gdy serwer prześle klientowi swój certyfikat, serwer wysyła komunikat CertificateRequest o chęci weryfikacji certyfikatu klienta.
2. Po wiadomości ServerHelloDone od serwera klient wysyła wiadomość Certificate zawierającą certyfikat klienta.
3. Klient wysyła wiadomość CertificateVerify, aby udowodnić, że posiada klucz prywatny.

2.4.4. Podpis elektroniczny

Podpisywanie kryptograficzne dokumentów najczęściej jest realizowane przy użyciu kryptografii asymetrycznej. Do składania podpisu korzysta się z klucza prywatnego, natomiast w celu weryfikacji stosuje się klucz publiczny.[38]

Podpis cyfrowy (ang. *digital signature*) charakteryzuje się następującymi cechami: jest autentyczny, niepodrabialny, nie nadaje się do ponownego użycia, niezmienialny. Nie można wyprzeć się podpisu.[34]

Schemat podpisu cyfrowego zazwyczaj zawiera trzy algorytmy:

1. algorytm generowania klucza wybiera klucz prywatny spośród zbioru możliwych kluczy prywatnych. Algorytm wytwarza klucz prywatny i odpowiedni klucz publiczny.
2. algorytm podpisywania generuje podpis gdy dostanie wiadomość i klucz prywatny.
3. algorytm weryfikujący podpis, gdy dostanie wiadomość, klucz publiczny i podpis, może uznać autentyczność wiadomości lub nie.

Dwie główne własności są wymagane. Pierwsza, autentyczność podpisu wygenerowanego z ustalonej wiadomości i ustalonego klucza prywatnego mogą być zweryfikowane przez użycie odpowiedniego klucza publicznego. Druga, generowanie poprawnego/ ważnego podpisu bez znajomości części klucza prywatnego jest obliczeniowo awykonalne.[34]

Podpis elektroniczny z mediatorem jest rozwinięciem modelu PKI. Finalizacja podpisu odbywa się online z udziałem mediatora, który ma połówkę klucza prywatnego.[34]

2.4.5. Zabezpieczony transfer danych w standardzie DICOM

Standard DICOM wykorzystuje istniejące standardy kryptograficzne do zabezpieczenia transferu danych. Algorytmy kryptograficzne były umówione w sekcji 2.5. Wymagania implementacji są określone w profilach bezpiecznego połączenia transportowego (ang. *Secure Transport Connection Profile*). [39] Profil bezpiecznego połączenia transportowego definiuje:

- opis struktury (ang. *framework*) protokołu i mechanizmów negocjacyjnych;
- opis uwierzytelniania podmiotu:
  - weryfikacja tożsamości uwierzytelnianych podmiotów;
  - mechanizm uwierzytelniania podmiotów;

  - mechanizm audytu dziennika zdarzeń;

- opis mechanizmu szyfrowania:
  - metoda dystrybucji kluczy sesji;
  - protokół szyfrowania i stosowne parametry;

- opis mechanizmu weryfikacji integralności.

Implementacja, która wspiera *Basic TLS Secure Transport Connection Profile* powinna wykorzystywać strukturę i mechanizm negocjacyjny określony w protokole TLS wersji 1.0. Protokół TLS opierający się na szyfrowaniu asymetrycznym i certyfikatach X.509 umożliwia integralność transmisji danych, poufność, uwierzytelnianie serwera. Chroni przed przekierowaniem na fałszywe strony internetowe. Szyfrowane dane nie mogą zostać zmienione czy przechwycone podczas transferu. [37] Tabela 2.3 opisuje mechanizmy, które powinny być wspierane jeśli odpowiadające cechy w TLS są wspierane przez jednostkę aplikacyjną. Profile nie wymagają od implementacji żeby zapewniała wszystkie funkcje TLS (uwierzytelnienie podmiotu, szyfrowanie, sprawdzanie integralności). Inne mechanizmy mogą być również użyte jeśli zostanie na to wyrażona zgoda w czasie negocjacji podczas tworzenia kanału TLS.[39]

Tabela 2.5 opisuje przedstawia minimalne wymagania dla mechanizmów szyfrujących charakterystycznych dla TLS (ang. *Minimum Mechanisms for TLS Features*):

Wspierane cechy TLS	Minimalny mechanizm
Uwierzytelnianie jednostki (ang. *Entity Authentication*)	Certyfikaty oparte na RSA
Wymiana głównych utajnionych danych (ang. Exchange of Master Secrets)	RSA
Integralność danych	SHA
Prywatność	Potrójny DES EDE, CBC [2]

Tabela 2.6: Minimalne mechanizmy dla wspieranych cech TLS.[39]

Porty IP, na których implementacja akceptuje połączenia TLS, lub mechanizmy przez które ten numer portu jest wybrany czy skonfigurowany, powinien być określony w Oświadczeniu o zgodności (ang. *Conformance Statement*). Ten port powinien różnić się od portów używanych dla innych rodzajów połączeń transportowych (bezpiecznych lub niebezpiecznych). [39] Oświadczenie o zgodności wskazuje również, jakie mechanizmy wspiera wdrożenie w odniesieniu do zarządzania kluczami. [39]

Profil nie określa w jaki sposób bezpieczne połączenie TLS jest nawiązywane ani nie poddaje ocenie certyfikatów wymienianych podczas uwierzytelniania podmiotu równorzędnego. Te problemy zostają w gestii AE, który standardowo przestrzega polityki bezpieczeństwa określonej jednostki. Identyfikacja właścicieli certyfikatów może być użyta przez jednostkę aplikacyjną dla dziennika zdarzeń lub żeby ograniczyć dostęp na podstawie zewnętrznych struktur kontroli dostępu. Gdy jednostka aplikacyjna nawiązała już bezpieczne połączenie (ang. *Secure Transport Connection*), aplikacja (ang. *Upper Layer Association*) może zacząć

[2] Obecnie 3DES jest jeszcze rekomendowany, ale z ograniczeniem rozmiaru danych oraz wymaganiem, aby każdy z trzech kluczy był inny. Aktualnie standardowo wykorzystywany algorytm to AES.[28, 29]

używać tego bezpiecznego łącza. [39]

Kiedy sprawdzenie integralności nie powiedzie się, połączenie powinno zostać przerwane zgodnie z protokołem TLS, powodując zarówno u nadawcy jak i odbiorcy wysłanie komunikatu A-P-ABORT do wyższych warstw z określeniem powodu. Powód powinien być udokumentowany w oświadczeniu zgodności. Nieudany test integralności może wskazywać na to, że bezpieczeństwo kanału zostało naruszone. [39]

2.5. Prawna ochrona danych

Wraz z rozwojem technologii informatycznych pojawiła się kwestia bezpieczeństwa danych, a szczególnie ochrony danych osobowych. [46] Aplikacje dające dostęp do zasobów gdzie zbierane są wrażliwe dane takie jak dane medyczne, wymagają wysokiego poziomu ochrony i szczególnych zabezpieczeń.[46]

Wg ustawy z dnia 29 sierpnia 1997 r. o ochronie danych osobowych (Dz.U. 1997 Nr 133 poz. 883), poprawki z roku 2001 – efekt wdrożenia dyrektywy UE 95/46/WE (w skrócie UODO) w artykule 6:

1. W rozumieniu ustawy za dane osobowe uważa się wszelkie informacje dotyczące zidentyfikowanej lub możliwej do zidentyfikowania osoby fizycznej.

2. Osobą możliwą do zidentyfikowania jest osoba, której tożsamość można określić bezpośrednio lub pośrednio, w szczególności przez powołanie się na numer identyfikacyjny albo jeden lub kilka specyficznych czynników określających jej cechy fizyczne, fizjologiczne, umysłowe, ekonomiczne, kulturowe lub społeczne.

3. Informacji nie uważa się za umożliwiającą określenie tożsamości osoby, jeżeli wymagałoby to nadmiernych kosztów, czasu lub działań.

Dane medyczne są danymi wrażliwymi, czyli są grupą danych, które wymagają szczególnej ochrony danych osobowych. W artykule 27 zdefiniowano jakich danych zabronione jest przetwarzanie oraz o wyjątkach:

1. Zabrania się przetwarzania danych ujawniających pochodzenie rasowe lub etniczne, poglądy polityczne, przekonania religijne lub filozoficzne, przynależność wyznaniową, partyjną lub związkową, jak również danych o stanie zdrowia, kodzie genetycznym, nałogach lub życiu seksualnym oraz danych dotyczących skazań, orzeczeń o ukaraniu i mandatów karnych, a także innych orzeczeń wydanych w postępowaniu sądowym lub administracyjnym.

2. Przetwarzanie danych, o których mowa w ust. 1, jest jednak dopuszczalne, jeżeli:

   - osoba, której dane dotyczą, wyrazi na to zgodę na piśmie, chyba że chodzi o usunięcie dotyczących jej danych,
   - przetwarzanie jest prowadzone w celu ochrony stanu zdrowia, świadczenia usług medycznych lub leczenia pacjentów przez osoby trudniące się zawodowo leczeniem lub świadczeniem innych usług medycznych, zarządzania udzielaniem usług medycznych i są stworzone pełne gwarancje ochrony danych osobowych,

Wg Rozporządzenia Ministra Zdrowia z dnia 9 listopada 2015 r. w sprawie rodzajów, zakresu i wzorów dokumentacji medycznej oraz sposobu jej przetwarzania w rozdziale 9, który mówi o szczególnych wymaganiach dotyczących dokumentacji prowadzonej w postaci elektronicznej w szczególności występuje paragraf 80 opisany następująco:

§ 80. Dokumentacja może być prowadzona w postaci elektronicznej, pod warunkiem prowadzenia jej w systemie teleinformatycznym zapewniającym:

- zabezpieczenie dokumentacji przed uszkodzeniem lub utratą;
- stały dostęp do dokumentacji dla osób uprawnionych oraz zabezpieczenie przed dostępem osób nieuprawnionych;

§ 83. 1. Dokumentację prowadzoną w postaci elektronicznej udostępnia się z zachowaniem jej integralności oraz ochrony danych osobowych.

§ 86. 1. Dokumentację prowadzoną w postaci elektronicznej uważa się za zabezpieczoną, jeżeli w sposób ciągły są spełnione łącznie następujące warunki:

1. jest zapewniona jej dostępność wyłącznie dla osób uprawnionych;
2. jest chroniona przed przypadkowym lub nieuprawnionym zniszczeniem;
3. są zastosowane metody i środki ochrony dokumentacji, których skuteczność w czasie ich zastosowania jest powszechnie uznawana.

Istniejące przepisy nie opisują wprost możliwości zdalnego przetwarzania danych medycznych. Mimo, że nowelizacja z 9 października 2015 roku umożliwiła outsourcing danych, to inne akty prawne znacząco ograniczają zdalne przetwarzanie danych.[32] Rozporządzenie w sprawie rodzajów oraz zakresu dokumentacji medycznej wraz ze sposobem jej przetwarzania dopuszcza zdalne przetwarzanie danych medycznych, natomiast rozporządzenia w sprawie dokumentacji przetwarzania danych osobowych i warunków technicznych oraz organizacyjnych wymaga wskazania miejsca oraz zasobów używanych do przetwarzania danych osobowych.[32]

Dodatkowo każdy szpital prowadzi własną, dodatkową politykę bezpieczeństwa. Pacjenci mają możliwość podpisania formularzy ze zgodą na przetwarzanie danych osobowych. Należy pamiętać o zapewnieniu dostępności, integralności i zabezpieczeniu danych przed dostępem osób nieuprawnionych. [46]

2.6. Anonimizacja, pseudonimizacja oraz deidentyfikacja danych

Informacje identyfikujące pacjenta z obrazów uzyskanych podczas przebiegu opieki klinicznej muszą być usunięte, przed przekazaniem do przetwarzania zdalnego. Jest to niezbędne aby chronić politykę prywatności pacjenta. Dodatkowo, często istnieje potrzeba usunięcia innych danych nie bezpośrednio związanych z identyfikacją pacjenta per se, ale które mogą umożliwić odzyskanie identyfikacji pacjenta lub utrudnić interpretację obrazu w którymś kierunku. Jednakże istotne jest zachowanie pewnych, specyficznych danych do kontroli jakości i analizy wyników, które są niezbędne do przeprowadzenia badania. Istnieją lokalne regulacje dotyczące prywatności danych medycznych jak np. dyrektywa UE (patrz: rozdział 2.4). Dane

i obrazy uzyskane do badań klinicznych są także często wydane do ponownego użycia, w którym to przypadku spełnienie wymogów polityki prywatności wymaga dużej ostrożności. [45, 3]

Deidentyfikacja definiowana jest jako usunięcie rzeczywistego identyfikatora pacjenta. Może być ona wymagana podczas tworzenia plików dydaktycznych, tworzenia innych rodzajów publikacji oraz przekazywania obrazów i związanych z nimi informacjami do rejestrów np. rejestry dawek onkologicznych lub rejestry dawek promieniowania. [45, 3]

Pseudonimizacja to deidentyfikacja i zastąpienie identyfikatorów przez pseudonimy, które są unikalne dla danej osoby i stosowane wyłącznie w określonym kontekście, ale nie są używane prywatnie przez te osoby. [45, 3]

Anonimizacja opisywana jest jako deidentyfikacja i dalsze usuwanie informacji lub pozbawienie ich jednoznaczności, aby zredukować prawdopodobieństwo ponownej identyfikacji obrazu nawet przy dostępie do innych źródeł informacji. [45, 3]

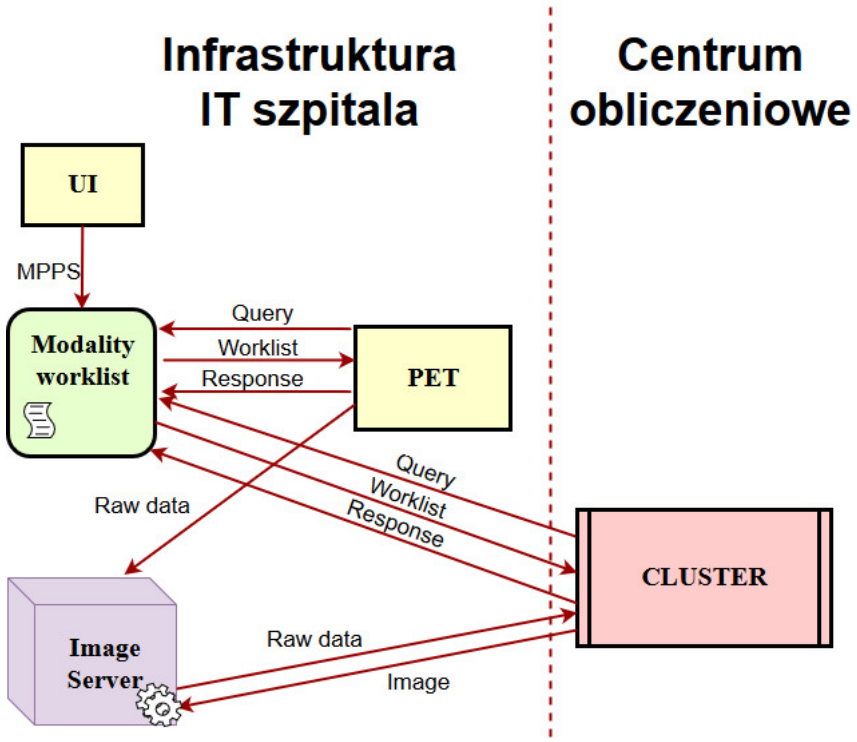
Rozdział 3

Analiza rozwiązań zdalnego przetwarzania danych

W pracy rozważano w jaki sposób jednocześnie zdalnie przetwarzać i chronić dane medyczne. Z jednej strony zaznajomiono się w pracy z funkcjonalnością standardów DICOM, DICOMweb oraz API REST, a z drugiej jak chronione są dane: kryptografia, PKI oraz anonimizacja, a w dodatku jakie są ograniczenia prawne wobec przetwarzania i przesyłania danych medycznych. Zaproponowano dwa rozwiązania zdalnego przetwarzania danych medycznych. Oba rozwiązania opierają się na rozwiązaniu chmurowym i wykorzystują zasoby klastra obliczeniowego do wykonywania zadań.

Pierwsze rozwiązanie opiera się o standard DICOM lub DICOMweb. Rozwiązanie to napotyka na dwa problemy. Pierwszym jest anonimizacja danych, drugim konieczność zdalnego dostępu do infrastruktury teleinformatycznej szpitala. O ile standard DICOM przewiduje konieczność usuwania danych wrażliwych z dokumentów DICOM podstawowym zastosowaniem jest tu eksport danych. Procedura ta wykonywana jest zazwyczaj przez dedykowaną aplikację zwaną anonimizerem (ang. *anonymizer*). Standard dopuszcza także wbudowaną funkcjonalność serwera PACS pozwalającą na prośbę klienta na anonimizację obrazów w trakcie udostępniania np. z wykorzystaniem protokołu DICOMweb. Nie istnieje analogiczna funkcjonalność dla Worklist DICOM. Oznacza to, że zdalne centrum przetwarzania miałoby dostęp do danych wrażliwych, co znacząco zwiększa wymagania dotyczące zabezpieczeń niezbędnych przy przechowywaniu i przetwarzaniu tych danych. Dodatkowo może rodzić problemy natury prawnej.

Zdefiniowany w standardzie DICOM proces obsługi Worklist wymaga, aby zdalne centrum obliczeniowe miało możliwość zainicjalizowania połączenia z serwerami RIS oraz PACS. To klaster obliczeniowy odpytuje modality worklist czy ma zadanie do zrealizowania. Wymaga to dostępu zdalnego do sieci teleinformatycznej szpitala. O ile możliwe jest zapewnienie bezpieczeństwa dla takiej komunikacji np.: z wykorzystaniem tuneli VPN opartych o protokół IPsec [1], rozwiązanie takie zwiększa niezbędne nakłady finansowe oraz co istotne wprowadza dodatkowy wektor ataku na infrastrukturę teleinformatyczną szpitala zarządzającą danymi wrażliwymi.

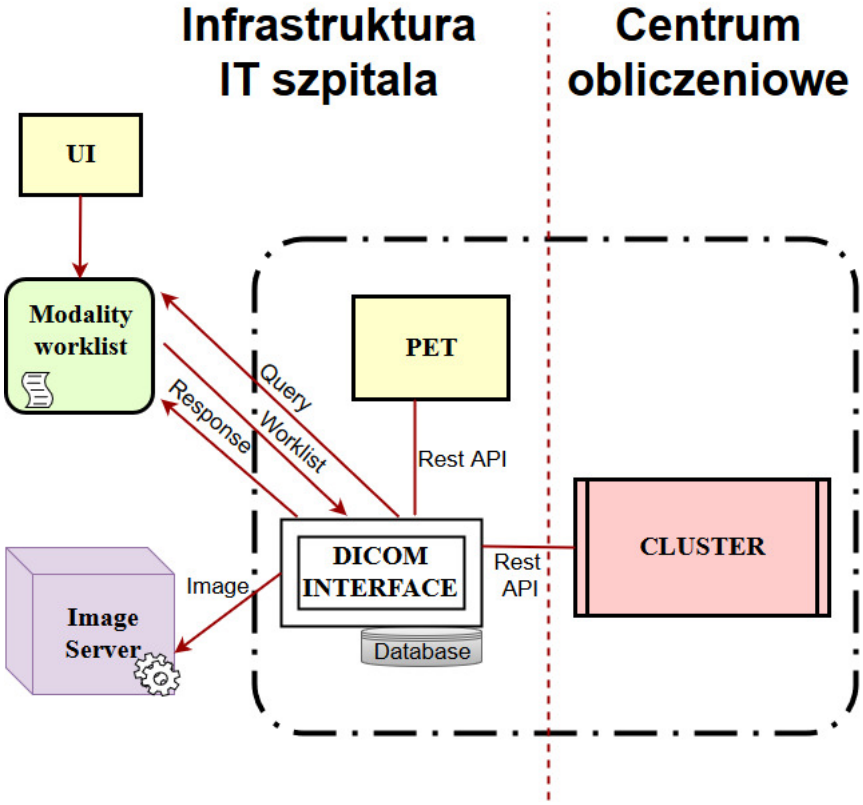


Rysunek 3.1: Schemat proponowanego rozwiązania ze standardem DICOM.

Zasada działania rozwiązania pierwszego:

1. Definicja badania sporządzona przez personel medyczny przez interfejs użytkownika UI
2. Zlecenie badania – utworzenie i przesłanie MPPS (ang. *Modality Performed Procedure Step*) do Modality worklist
3. Odpytywanie Modality worklist przez skaner PET jakie są zadania do zrealizowania
4. Wysyłanie worklisty przez Modality worklist do skanera PET zawierającą dane o pacjencie i badaniu
5. Po zakończeniu skanowania następuje wysłanie przez skaner PET surowych danych do serwera Image oraz wysłanie do Modality Worklist informacji dotyczącej zmiany statusu
6. Odpytanie Modality worklist przez klaster obliczeniowy o obrazy oczekujące na rekonstrukcję
7. Anonimizacja surowych danych na serwerze Image
8. Wysłanie zanonimizowanych danych na klaster obliczeniowy
9. Rekonstrukcja danych na klastrze obliczeniowym
10. Wysłanie zrekonstruowanych danych na Image serwer oraz wysłanie do Modality Worklist informacji dotyczącej zmiany statusu

Co więcej, rozwiązanie to charakteryzuje się tym, że potrzebna jest komunikacja z dwóch stron. Klaster obliczeniowy musiałby odpytywać Modality worklist czy ma zadanie do zrealizowania. Rozwiązanie te jest możliwe do wdrożenia, ale wymaga znaczących zmian w sieci teleinformatycznej szpitala.



Rysunek 3.2: Schemat proponowanego rozwiązania ze zdalnym centrum obliczeniowym.

Drugie rozwiązanie polega na dodaniu do systemu elementu pośredniczącego (tzw. DICOM Interface) między systemem DICOM szpitala a zdalnym centrum obliczeniowym. Jednym z zadań tego elementu będzie anonimizacja danych. W tym modelu z punktu widzenia systemu informatycznego szpitala DICOM INTERFACE, skaner PET oraz Klaster obliczeniowy będą stanowić jedno urządzenie. Komunikacja pomiędzy DICOM INTERFACE a skanerem PET i Klastrem obliczeniowym oparta zostanie na protokole REST. Natomiast komunikacja z innymi elementami systemu teleinformatycznego szpitala (w szczególności serwerami RIS i PACS) oparta będzie na standardzie DICOM. Komunikacja z zdalnym ośrodkiem obliczeniowym zostanie zabezpieczona z wykorzystaniem szyfrowania protokołem TLS z wykorzystaniem dedykowanej infrastruktury klucza publicznego (PKI) X.509. Założenie jest takie, żeby każdy szpital miał swój certyfikat, tak aby za pomocą certyfikatu możliwa była identyfikacja zadań z poszczególnych szpitali. Powołane zostanie dedykowane CA, które posłuży do wystawienia certyfikatów dla centrum obliczeniowego oraz szpitali. Podczas nawiązywania komunikacji pomiędzy szpitalem a centrum obie strony będą miały możliwość potwierdzenia autentyczności przedstawionych certyfikatów dzięki znajomości certyfikatu publicznego CA. Ponieważ każdy certyfikat zawiera dane identyfikujące podmiot możliwe będzie na tej podstawie określenie tożsamości klienta centrum oraz autoryzacja dostępu do danych

tylko tego szpitala.

Zasada działania rozwiązania drugiego:

1. Definicja badania sporządzona przez personel medyczny przez interfejs użytkownika UI
2. Zlecenie badania - utworzenie i przesłanie MPPS (ang. *Modality Performed Procedure Step*) oraz zmiana statusu do Modality worklist
3. Zapytanie do Modality Worklist przez DICOM Interface o nowe badania
4. Przesłanie worklisty do DICOM Interface przez Modality Worklist
5. Przesłanie zadania do skanera PET przez DICOM Interface
6. Po zakończeniu skanowania następuje wysłanie surowych danych do DICOM Interface
7. Anonimizacja danych przed wysłaniem do zdalnego ośrodka obliczeniowego
8. Wysłanie zanonimizowanych danych na klaster obliczeniowy przez DICOM Interface
9. Rekonstrukcja danych na klastrze obliczeniowym
10. Pobranie zrekonstruowanych danych przez DICOM Interface z klastra obliczeniowego
11. Zapisanie zrekonstruowanych danych na Image Server oraz wysłanie informacji o zmianie statusu do Modality Worklist

Zadaniem projektowanego API jest umożliwienie systemowi szpitala na rozpoczęcie, monitorowanie, pobranie wyników oraz przerwanie zadań obliczeniowych. Zadaniami obliczeniowymi mogą być rekonstrukcja surowych danych pochodzących z skanera PET jak i ich analiza. Zaproponowane API znajduje się w załączniku, gdzie wymienione są wszystkie punkty końcowe wraz z opisami, parametrami, metodami i odpowiedziami. W przyszłości możliwe jest rozszerzenie implementacji o możliwość dodawania podzadań.

Rozdział 4

Wyniki pracy i dyskusja

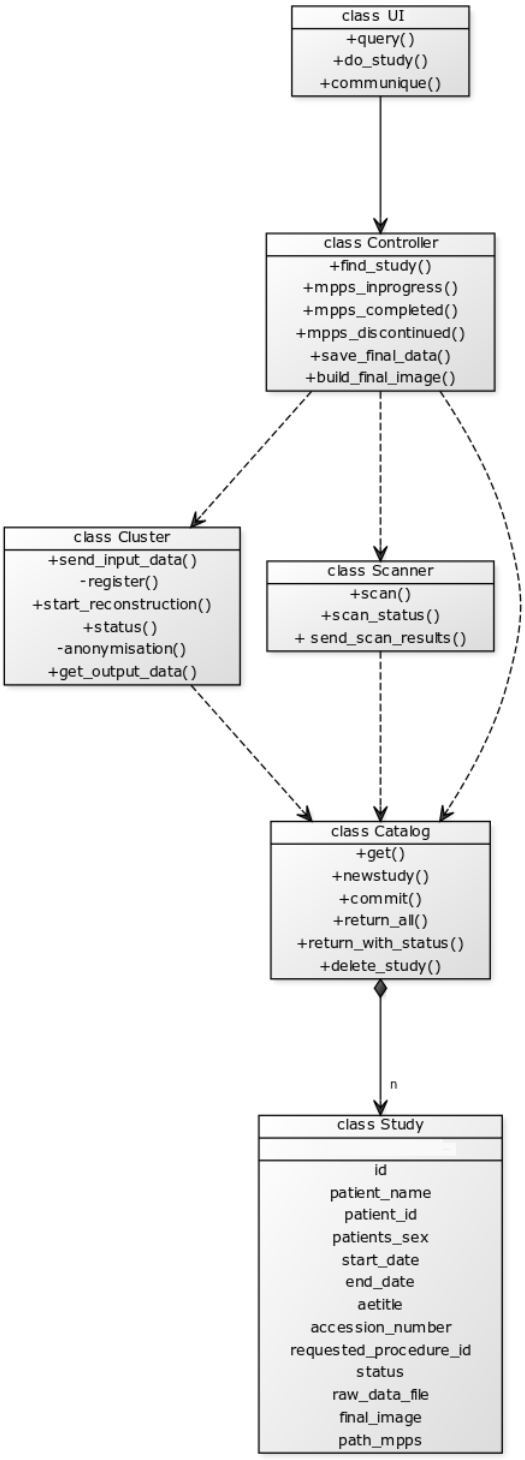
Porównanie dwóch rozwiązań dla zdalnego przetwarzania danych opierających się o standardowy DICOM oraz rozszerzonego o DICOM interface i API REST zostały przedstawione w rozdziale 3. Na tej podstawie można wymienić następujące zalety rozwiązania rozszerzonego: prostsza technicznie implementacja, niższe koszty, większe możliwości anonimizacji danych. Z tych względów kod demonstracyjny stanowi przykład rozwiązania poszerzonego opierającym się częściowo o standard DICOM i własnym API. Językiem programowania, który został wybrany do zobrazowania zasady działania systemu, jest python. Jest to język programistyczny obiektowy wysokiego poziomu charakteryzujący się zwięzłością i klarownością, rozwijany w trybie Open Source na licencji Python Software Foundation. Został stworzony w 1991 roku przez Guido van Rossuma. Python charakteryzuje się minimalistyczną składnią, prostą semantyką, wszechstronnymi standardowymi bibliotekami w tym m.in. interfejs API wraz z określonymi funkcjami systemów operacyjnych (OS). Python zawiera wiele schematów programowania, np. programowanie zorientowane obiektowo – możliwość tworzenia klas (class), programowanie proceduralne – definiowanie funkcji (def) oraz programowanie funkcjonalne. Python definiuje obiekty tj. listy, krotki, słowniki. Jest językiem o dynamicznym systemie plików, czyli zmienne nie muszą być definiowane oraz nie posiadają stałego typu. [27, 23]

Python ma szerokie zastosowanie. Za pomocą tego języka programowania można tworzyć gry, strony internetowe, aplikacje desktopowych (komputerowe) i sieciowe, skrypty generujące zestawienia i raporty. Używany jest przez m.in. Google, Yahoo, Nokie, IBM, NASA, Youtube. [31]

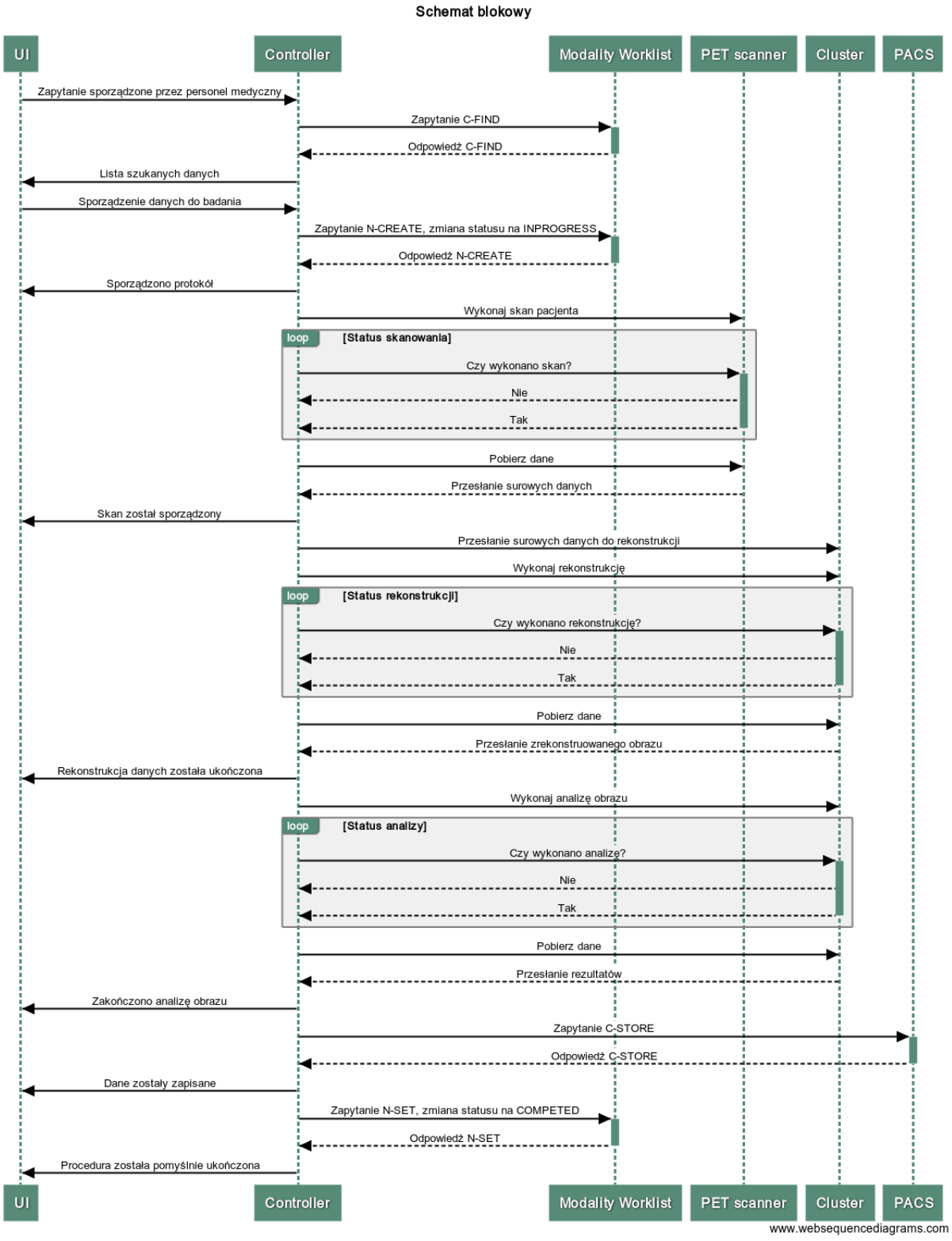
W projekcie wykorzystywane były biblioteki takie jak pydicom, pynetdicom3 dostępne w trybie open source na: `https://github.com/pydicom` oraz `https://github.com/pydicom/pynetdicom3`

Początkowo skupiono się na komunikacji z Worklistami, a dokładnie na połączeniu po stronie klienta SCU. Następnie testowano komunikację pomiędzy Modality Emulatorem, a Emulatorem RIS z projektu DVTK (`https://www.dvtk.org/`), testowano funkcje takie jak: c-echo, c-find, n-create. Następnym etapem było testowanie komunikacji pomiędzy Emulatorem RIS z projektu DVTK, a skryptów z projektu pynetdicom oraz pydicom: eschoscu, findscu, c-store. Kolejnym krokiem było napisanie skryptu ncreate, który miał komunikować się z Emulatorem RIS. Potem testowano komunikację skryptów c-echo, c-find, n-create między dwiema aplikacjami, z których jedna to SCU – klient, a druga to SCP – serwer.

Kluczowym zadaniem było stworzenie demonstratora kodu, który pokazywałby zasadę działania i następstwa komunikacji pomiędzy skanerem PET, kontrolerem modalności, klastrem obliczeniowym, interfejsem UI personelu szpitala. Ze względu na brak protokołu komunikacji z prototypem skanera J-PET oraz brak implementacji API po stronie klastra wszystkie elementy zewnętrzne w tym systemy DICOM zostały zastąpione przez obiekty fantomowe. Utworzono odpowiednio: diagram UML oraz schemat blokowy, aby obrazowo opisać strukturę aplikacji i kolejne czynności algorytmu.



Rysunek 4.1: Diagram UML (ang. *Unified Modeling Language)* przedstawiający strukturę aplikacji.



Rysunek 4.2: Schemat blokowy przedstawiający kolejne czynności algorytmu w przypadku ukończenia całego badania. Na zielonych polach znajdują się nazwy modułów programów oraz urządzenia będące w sieci. Elementy będące wyżej są wcześniej w kolejności względem leżących niżej. Poziome strzałki wskazują kierunek przesyłania informacji wraz z opisem. Na schemacie widoczne są pętle (ang. *loop*).

Docelowy skrypt (main.py): składa się z 3 klas: Scanner, Cluster, Controller oraz z funkcji run, która uruchamia cały mechanizm. Ponadto utworzono program (ui.py) implementujący interfejs użytkownika. Składa się z jednej klasy UI oraz trzech metod odpowiedzialnych za komunikację z personelem medycznym. Kod dostępny jest na githubie: `https://github.com/AnnaJapko/dicom-pet-interface`

Klasa Controller implementuje algorytm ze schematu blokowego (Rysunek 4.2) i jest kluczowy do poprawnego działania programu. Zależność tej klasy z pozostałymi obiektami została opisana na diagramie UML (Rysunek 4.1).

Klasa Scanner definiuje funkcje związane z komunikacją ze skanerem. Metoda scan: komunikuje się ze skanerem PET wykorzystując dedykowany niskopoziomowy protokół. Wysyła polecenie rozpoczęcia badania wraz z plikiem DICOM opisującym przeprowadzaną procedurę. Funkcja scan_status odpowiedzialna jest za sprawdzanie statusu wykonywanej procedury. Gdy skan zostaje ukończony, funkcja scan_status aktualizuje status badania w systemie, a funkcja send_scan_results wysyła dane do kontrolera modalności.

Klasa Cluster implementuje komunikację pomiędzy klastrem obliczeniowym, a kontrolerem modalności. Zawiera metody odpowiedzialne za pobranie danych wejściowych z kontrolera, rekonstrukcję, anonimizację, wysłanie plików wyjściowych z klastra.

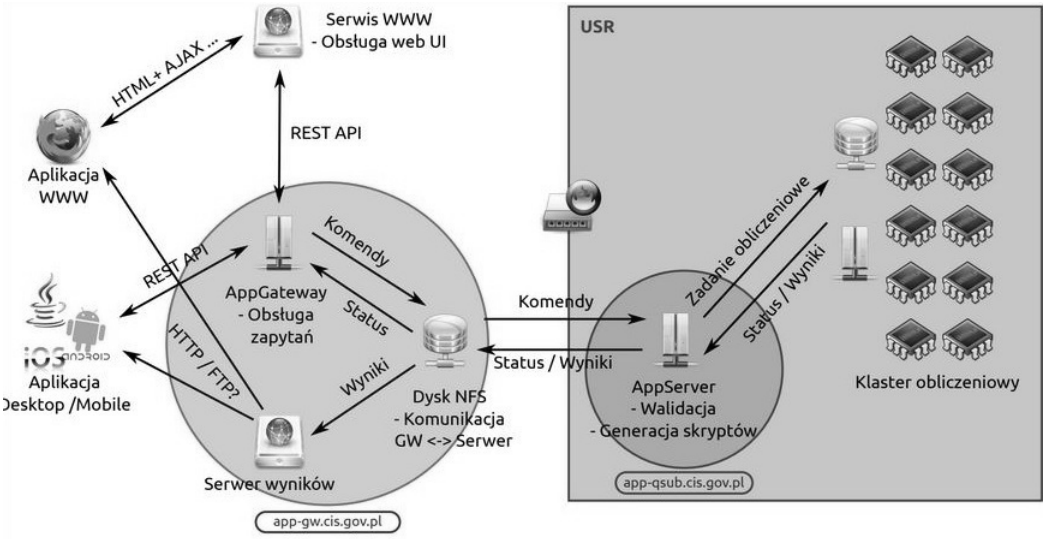
Część programu odpowiedzialna z komunikację z bazą danych SQL (w oparciu o bibliotekę sqlalchemy) wydzielono do oddzielnego modułu (catalog,py). Celem wykorzystania bazy danych jest możliwość tworzenia i modyfikowanie rekordów opisujących prowadzone badania z możliwością nieulotnego zapisu.

Klasa Status definiuje wszystkie stany przebiegu procesu: od stworzenia nowego badania, po skanowanie, rekonstrukcję, anonimizację, analizę po wysłanie końcowego obrazu i zakończenia procedury.

Klasa Study określa jakie kolumny mają być w bazie danych i jakiego typu rekordy mają być wstawiane (w tym przypadku stringi oraz integery). Dodatkowo można ustawić unikalność (unique), niezerowość (nullable) albo wymagalność (required). Ponadto metoda repr definiuje w jaki sposób przedstawiane mają być dane.

Klasa Catalog zawiera metody get, commit i new_study. Funkcja get pobiera badanie o wskazanym id. Funkcja commit zatwierdza dane działanie w sesji, metoda new_study tworzy i dodaje nowy rekord do bazy danych.

Zaproponowany też został projekt interfejsu API REST do komunikacji z klastrem obliczeniowym. Mógłby stanowić on rozszerzenie architektury zaproponowanej przez autorów z NCBJ przy opracowywaniu zdalnego przetwarzania danych w zastosowaniach fizyki jądrowej. Kluczowe właściwości to bezpieczeństwo, ułatwiony dostęp do zasobów obliczeniowych oraz szybka implementacja. Infrastruktura usług internetowych składa się z AppGateway oraz AppServera. Z jednej strony AppGateway jest punktem dostępowym dla klientów, a z drugiej strony oddziela klienta od AppServera. AppServer zajmuje się weryfikacją danych wejściowych, tworzeniem plików wejściowych, rejestrowaniem zużycia zasobów, monitoringiem statusów zadań, współdziałaniu z systemami kolejkowymi. [21] Postać API została zaprezentowana w załączniku B.



Rysunek 4.3: Architektura usług internetowych CIŚ.[21]

Rozdział 5

Podsumowanie

Pozytonowa tomografia emisyjna (PET) jest to nowoczesna technika diagnostyczna umożliwiająca zlokalizowanie w ciele pacjenta obszarów o wzmożonej aktywności metabolicznej charakterystycznej dla komórek nowotworowych. Przetworzenie rejestrowanych w czasie badania sygnałów pochodzących z detektorów scyntylacyjnych na trójwymiarowy obraz medyczny jest bardzo złożonym zadaniem, mogącym wymagać przesłania danych ze szpitala do zewnętrznego ośrodka komputerowego. Dzieje się tak między innymi w przypadku, gdy wolumen danych jest duży, np. w tomografie J-PET w badaniu ang. *whole body*. Najlepsze efekty przynosi wtedy stosowanie rozwiązań chmurowych wymagających przesłania danych.

Dane medyczne są tzw. wrażliwymi danymi, które muszą być odpowiednio anonimizowane zanim zostaną wysłane do zdalnego ośrodka obliczeniowego. Drugą ważną kwestią jest bezpieczeństwo komunikacji. Zanonimizowane dane muszą być przesyłane przez internet za pomocą szyfrowania SSL z kluczem publicznym X.509. Gromadzenie coraz większych wolumenów danych medycznych ang. *Big Data* umożliwia wykorzystanie mechanizmów uczenia maszynowego do analizy obrazów medycznych.

Rysunek 5.1: Różne aspekty rozwoju technik obrazowania medycznego

W pracy porównano model zdalnego przetwarzania danych z badania PET według standardowego protokołu DICOM oraz zaproponowany w niniejszej pracy model rozszerzony o elementy DICOM Interface oraz komunikację API REST. W celu przetestowania obydwu modeli stworzono dedykowane połączenie API REST pomiędzy DICOM Interface a klastrem obliczeniowym oraz skanerem PET. Utworzono demonstrator kodu pokazujący zasadę działania

i następstwa komunikacji między skanerem, kontrolerem modalności, a klastrem obliczeniowym oraz skrypt będący bazą danych sqlalchemy, gdzie rekordy mogą być dodawane, usuwane czy aktualizowane oraz program będący implementacją interfejsu użytkownika. Zaproponowano zabezpieczenie połączenia poprzez szyfrowanie SSL z kluczem publicznym X.509.

Przeprowadzona analiza działania obydwu modeli prowadzi do wniosku, ze podejście opieranie się wyłącznie o protokół DICOM mogłoby generować zwiększone koszty oraz problemy techniczne związane z koniecznością instalacji dodatkowych urządzeń oraz nieefektywnym wykorzystaniem klastra obliczeniowego. Dodatkowo, standardowe rozwiązanie zapewnia anonimizację jedynie części danych przed wysłaniem ich ze szpitala. Zaproponowany w pracy model rozszerzony nie ma powyższych ograniczeń.

Spis rysunków

1.1. Schemat zjawiska anihilacji 8
1.2. Obszary dominacji występowania efektów oddziaływania fotonów z materią. Wykres jest w funkcji energii fotonów oraz liczby atomowej Z absorbentu.[36] 9
1.3. Schemat efektu fotoelektrycznego 10
1.4. Schemat rozpraszania Comptona, λ to długość fali padającej, λ' to długość fali rozproszonej, θ to kąt rozproszenia fotonu. 10
1.5. Schemat tomografu PET [10] 12
1.6. Metoda rekonstrukcji punktu anihilacji. Na obrazku przedstawione zostały dwa plastikowe paski z modułami detekcyjnymi z dwoma anihilacyjnymi kwantami gamma. [8] [12] [9] 13
1.7. Zdjęcie pełnowymiarowego prototypu J-PET; Schemat przedstawiający warstwy pierścieni detekcyjnych. [15] 14
1.8. Schemat LOR dla 3 przykładowych anihilacji (obszar oznaczony czerwoną kropką). Porównano informacje związane z punktem anihilacji dla trzech różnych metod: standardowy PET – czerwona linia, TOF-PET o rozdzielczości TOF równej 540 ps (szerokości połówkowej, tzw. FWHM (ang. *Full width at half maximum*)) – żółta linia oraz J-PET o 290 ps (FWHM) – niebieska linia.[11] 15
1.9. Schemat przepływu danych podczas rekonstrukcji obrazu PET. Nieprzetworzone dane są zbierane przez system DAQ (ang. *Data Acquisition System*). Następnie są przetwarzanie przez moduł rekonstrukcji niskiego poziomu (low-level reco.) i prowadząc do utworzenia zbiorów zrekonstruowanych linii LOR, które następnie są wysyłane do modułu rekonstrukcji obrazu (Image reco.). Końcowy obraz może być odtworzony w przeznaczonym do tego celu programie bądź może być eksportowany do formatu DICOM. [16] 16
1.10. Implementacja CPU i GPU. Czas iteracji pojedynczej rekonstrukcji obrazu jako funkcja rozdzielczości obrazu docelowego.[16] 17

2.1. Schemat usługi w gridzie. [14] 20
2.2. Schemat usługi w chmurze z charakterystycznymi czteroma warstwami (patrząc od dołu schematu): obliczeniowe i magazynujące elementy komputerowe (hardware), maszyny wirtualne; moduł rozdzielania zadań (ang. *resource allocator*); oraz użytkowników. [17] 21
2.3. Model DICOM świata rzeczywistego. Wartości nad strzałkami określają liczbę możliwych połączeń.[40] 22
2.4. Przykładowe porównanie informacji rzeczywistych z zaproponowanym modelem informatycznym DICOM. [25] 23
2.5. Schemat modelu informacji IOD. Widoczny jest podział na tematyczne zbiory - Information Entity. Te z kolei są podzielone na moduły, które składają się z artybutów.[24] 24

2.6. Schemat ilustrujący klasy usług i obiektów DICOM. [48] 25
2.7. Schemat budowy strumienia informacji. Strumień informacji składa się z podstawowych elementów danych. Te natomiast składają się z identyfikatora, typu danych, rozmiaru danych, informacji.[24] 26
2.8. Schemat przedstawiający różnice pomiędzy modelem OSI, a TCP/IP. Model OSI ma 7 warstw: warstwy wyższe [warstwa aplikacji (ang. *application layer*), warstwa prezentacji (ang. *presentation layer*), warstwa sesji (ang. *session layer*)], warstwy niższe [warstwa transportowa (ang. *transport layer*), warstwa sieciowa (ang. *network layer*), warstwa łącza danych (ang. *data link layer*), warstwa fizyczna (ang. *physical layer*)]. Model TCP/IP składa się z 4 warstw: warstwa aplikacji czy tzw. procesorowa (ang. *process layer*), warstwa transportowa (ang. *host-to-host layer*), warstwa protokołu internetowego czy inaczej warstwa internetu (ang. *internet protocol layer*), warstwa dostępu do sieci bądź warstwa fizyczna (ang. *network access layer*). [47] 28
2.9. Architektura protokołu sieci DICOM.[41] 29
2.10. Struktura wiadomości DICOM.[41] 30
2.11. Konstrukcja modelu przepływu danych dla szybkich medycznych rekonstrukcji obrazu w gridzie. [17] 35
2.12. Schemat komunikacji pomiędzy klientem, a serwerem poprzez protokół HTTP. [35] 36
2.13. Schemat szyfrowania i deszyfrowania. [4] 37
2.14. Schemat szyfrowania i deszyfrowania w systemach symetrycznych. [4] 38
2.15. Schemat szyfrowania i deszyfrowania w systemach asymetrycznych. [4] 40
2.16. Schemat komponentów PKI zawiera: inne PKI, organ Zarządzania Polityką Certyfikacji - PMA, Główne CA, urzędy rejestracyjne i certyfikacyjne, repozytorium certyfikatów oraz użytkowników. [34] 42
2.17. Ścieżka weryfikacji i certyfikacji.[37] 43
2.18. Porównanie protokołu HTTP i HTTPS. [37] 44
2.19. Schemat szyfrowania HTTPS.[37] 46

3.1. Schemat proponowanego rozwiązania ze standardem DICOM. 53
3.2. Schemat proponowanego rozwiązania ze zdalnym centrum obliczeniowym. . . 54

4.1. Diagram UML (ang. *Unified Modeling Language)* przedstawiający strukturę aplikacji. 58
4.2. Schemat blokowy przedstawiający kolejne czynności algorytmu w przypadku ukończenia całego badania. Na zielonych polach znajdują się nazwy modułów programów oraz urządzenia będące w sieci. Elementy będące wyżej są wcześniej w kolejności względem leżących niżej. Poziome strzałki wskazują kierunek przesyłania informacji wraz z opisem. Na schemacie widoczne są pętle (ang. *loop*). 59
4.3. Architektura usług internetowych CIŚ.[21] 61

5.1. Różne aspekty rozwoju technik obrazowania medycznego 62

Spis tabel

2.1. Najczęściej stosowane znaczniki. Tabela zawiera kolumny z nazwą, identyfikatorem i opisem. 27
2.2. Typy danych wraz z opisem oraz rozmiarem. 27
2.3. Dostępne serwisy DICOMweb.[43] 34
2.4. Metody HTTP wraz z opisem działania 37
2.5. Kody statusów HTTP z kodem, nazwą i opisem 37
2.6. Minimalne mechanizmy dla wspieranych cech TLS.[39] 48

Bibliografia

[1] R. Atkinson. *Security Architecture for the Internet Protocol*. RFC, 1995.

[2] A. Dietlaf B. Jaworski. *Kurs fizyki tom 3*. PWN, Warszawa, 1974.

[3] D. Clunie. Suplement 142 standardu dicom. `http://www.dclunie.com/papers/D2_1045_Clunie_Deidentification.pdf/`.

[4] M. Srebrny Cz. Kościelny, M. Kurkowski. *Kryptografia teoretyczne podstawy i praktyczne zastosowania*. Wydawnictwo PJWSTK, Warszawa, 2009.

[5] D. E. Robling Denning. *Kryptografia i ochrona danych*. Wydawnictwo Naukowo-Techniczne, Warszawa, 1992.

[6] B. Dziunikowski. *O fizyce i energii jądrowe*. AGH Uczelniane Wydawnictwa Naukowe, Kraków, 2001.

[7] A. Esteva et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature volume 542, pages 115–118*, 2017.

[8] L. Raczyński et al. Novel method for hit-position reconstruction using voltage signals in plastic scintillators and its application to positron emission tomography. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 764:186–192, 2014.

[9] L. Raczyński et al. Compressive sensing of signals generated in plastic scintillators in a novel j-pet instrument. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 786:105–112, 2015.

[10] P. Moskal et al. Novel detector systems for the positron emission tomography. *arXiv preprint arXiv:1305.5187*, 2013.

[11] P. Moskal et al. J-pet: nowy pozytonowy emisyjny tomograf zbudowany z plastikowych detektorów. *Inżynier i Fizyk Medyczny 4*, pages 235–238, 2015.

[12] P. Moskal et al. A novel method for the line-of-response and time-of-flight reconstruction in tof-pet detectors based on a library of synchronized model signals. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 775:54–62, 2015.

[13] R. Broda et al. *Człowiek i promieniowanie jonizujące*. PWN, Warszawa, 2001.

[14] S.F. et al., El-Zoghdy. A threshold-based load balancing algorithm for grid computing systems. *Journal of High Speed Networks, vol. 21, no. 4, pp. 237-257*, 2015.

[15] Sz. Niedźwiecki et al. J-pet: a new technology for the whole-body pet imaging. *arXiv preprint arXiv:1710.11369*, 2017.

[16] W. Krzemień et al. Overview of the software architecture and data flow for the j-pet tomography device. *arXiv preprint arXiv:1508.02451*, 2015.

[17] W. Wiślicki et al. Computing support for advanced medical data analysis and imaging. *Bio-Algorithms and Med-Systems*, 10(2):53–58, 2014.

[18] Instytut Fizyki. Detekcja i detektory promieniowana gamma oraz pomiary energii fotonów. `http://www.if.pw.edu.pl/~pluta/pl/dyd/mfj/zal03/bilski/index.html`.

[19] B. Galwas H. Górkiewicz-Galwas. *Przyrządy elektronowe*. Wydawnictwa Szkolne i Pedagogiczne, Warszawa, 1986.

[20] Narodowe Centrum Badań Jądrowych. Od detektora scyntylacyjnego do gamma kamery. `http://ncbj.edu.pl/zasoby/wyklady/ld_podst_fiz_med_nukl-01/med_nukl_08_v7.pdf`.

[21] P. Szwajkowski K. Gomulski S. Potempski, K. Klimaszewski. *Modern Approach to Security of Software for Nuclear Facility in Świerk Computing Centre*. The International Conference on Computer Security in a Nuclear World: Expert Discussion and Exchange, Austria, 2015.

[22] `https://arstechnica.com/information-technology/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/`.

[23] M. Lutz. *Python. Wprowadzenie*. Wydawnictwo HELION, Gliwice, 2011.

[24] `http://home.agh.edu.pl/~socha/pmwiki/pmwiki.php/DICOM/ModelInformacji`.

[25] `http://home.agh.edu.pl/~socha/pmwiki/pmwiki.php/DICOM/OpisStandardu`.

[26] J. M. Massalski. *Detekcja Promieniowania Jądrowego*. PWN, Warszawa, 1959.

[27] `http://www.web2py.com/books/default/chapter/42/02/jezyk-programowania-python`.

[28] `https://csrc.nist.gov/projects/block-cipher-techniques`.

[29] `https://csrc.nist.gov/projects/digital-signatures`.

[30] M. R. Ogiela. *Podstawy kryptografii*. Wydawnictwa AGH, Kraków, 2000.

[31] `http://www.python.rk.edu.pl/w/p/python-co-jest-i-do-czego-mozna-go-uzyc/`.

[32] `https://blog.dzp.pl/ip/ochrona-danych-medycznych-w-chmurze/`.

[33] Dongarra Michel Daydé Jack J. Vicente Hernández José MLM Palma. *High Performance Computing for Computational Science–VECPAR*. Springer, 2004.

[34] `http://edu.pjwstk.edu.pl/wyklady/bsi/scb/main51.html/`.

[35] `https://pypila.stxnext.pl/blog/api/`.

[36] D.Lal R.C. Reedy, J.R. Arnold. Ann. rev. nucl. part. sci. pages 33, 505, 1983.

[37] https://tiptopsecurity.com/how-does-https-work-ssl-tls-explained/.

[38] B. Schneier. *Kryptografia dla praktyków*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2002.

[39] http://dicom.nema.org/medical/dicom/current/output/pdf/part15.pdf.

[40] http://dicom.nema.org/Dicom/2011/11_03pu.pdf.

[41] http://dicom.nema.org/medical/dicom/current/output/pdf/part07.pdf.

[42] http://dicom.nema.org/medical/dicom/current/output/pdf/part08.pdf.

[43] www.dicomstandard.org/dicomweb/.

[44] A. Strzałkowski. *Wstęp do fizyki jądra atomowego*. PWN, Warszawa, 1978.

[45] ftp://medical.nema.org/medical/dicom/final/sup142_ft.pdf/.

[46] R. Tadeusiewicz. *Informatyka Medyczna*. Uniwersytet Marii Curie-Skłodowskiej, Instytut Informatyki, Lublin, 2011.

[47] A. S. Tanenbaum. *Sieci komputerowe*. Helion, 2004.

[48] http://www.ihe.net/Technical_Framework/upload/ihe_tf_rev9-0ft_vol2_2008-06-26.pdf.

[49] ftp://ftp.rsa.com/pub/pdfs/understanding_pki.pdf.

[50] S. Weinberg. *The Quantum Theory of Fields*. Cambridge University Press, Cambridge, 1995.

[51] https://lukaszkotblog.wordpress.com/2017/12/29/piszemy-rest-api/.

[52] https://websitestyle.pl/blog/api-lacznosc-miedzy-aplikacjami/.

Załączniki

Demonstrator kodu

Kod programu main.py:

```
#Zbiór bibliotek
import pydicom
from catalog import Catalog, Status
import time
import os
import shutil
import datetime
from ui import UI

class Controller():
    """
    Klasa implementująca system zarządzający przebiegiem procesu badania
    dla modalności PET.
    """
    def find_study(pat_name, pat_surname):
        """
        Funkcja wykonuje zapytanie C-FIND do Radiology Information System (RIS),
        które wyszukuje worklisty o zadanych parametrach.

        :param pat_name: Imię pacjenta.
        :param pat_surname: Nazwisko pacjenta
        :return: Lista obiektów DICOM opisujących pasujące Studies zwrócona przez RIS.
        """
        list_of_files=[]
        dicom_file= pydicom.dcmread("worklist_rsp.dcm", force=True)
        list_of_files.append(dicom_file)
        UI.communique(list_of_files)
        return list_of_files


    def mpps_inprogress(identifier):
        """
        Funkcja N-CREATE zmieniająca status na in-progress- czyli badanie jest
        w trakcie wykonywania.
```

```
    :param pat_name: Identyfikator.
    """
    pydicom.dcmread("mpps_inprogress.dcm", force=True)
    examination= Catalog.get(identifier)
    #Zapis pliku mpps_discontinued.dcm do bazy danych
    examination.path="C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\\pynetdicom3\
    \\pynetdicom3\\apps\\findscu\\mpps_inprogress.dcm"
    #Zmiana statusu w bazie danych
    examination.status=Status.dicom_inprogress
    Catalog.commit()
    return



def mpps_completed(identifier):
    """
    Funkcja N-CREATE zmieniająca status na completed- oznacza, że badanie
    zostało zakończone.

    :param pat_name: Identyfikator.
    """
    pydicom.dcmread("mpps_completed.dcm", force=True)
    examination= Catalog.get(identifier)
    #Zapis pliku mpps_discontinued.dcm do bazy danych
    examination.path="C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\\pynetdicom3\
    \\pynetdicom3\\apps\\findscu\\mpps_completed.dcm"
    #Zmiana statusu w bazie danych
    examination.status=Status.dicom_completed
    Catalog.commit()
    return


def mpps_discontinued(identifier):
    """
    Funkcja N-CREATE zmieniająca status na discontinued- oznacza, że badanie
    zostalo anulowane.

    :param pat_name: Identyfikator.
    """
    pydicom.dcmread("mpps_discontinued.dcm", force=True)
    examination= Catalog.get(identifier)
    #Zapis pliku mpps_discontinued.dcm do bazy danych
    examination.path="C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\\pynetdicom3\
    \\pynetdicom3\\apps\\findscu\\mpps_discontinued.dcm"
    #Zmiana statusu w bazie danych
    examination.status=Status.failed
    UI.communique("The test failed")
    Catalog.commit()
```

```
        return

    def save_final_data(identifier,file_path_cluster):
        """
        Metoda savefinal_data  obsługuje funkcję C-STORE pozwalającą na wysłanie
        obrazów z kontrolera modalności do serwera np. PACS.

        :param pat_name: Identyfikator.
        """

        final_image_controller = "C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\
        \\pynetdicom3\\pynetdicom3\\apps\\findscu\\controller\\"
        final_file=shutil.copy(file_path_cluster,final_image_controller)
        examination= Catalog.get(identifier)
        #Zapis pliku do bazy danych
        examination.final_image=final_file
        #Zmiana statusu w bazie danych
        examination.status=Status.procedure_completed
        Catalog.commit()
        UI.communique("Final data are saved")
        return


    def build_final_image(identifier, path2):
        """
        Metoda build_final_image odpowiedzialna jest za budowanie końcowego obrazu
        oraz deanonimizację

        :param pat_name: Identyfikator.
        """
        cluster_path='C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\\pynetdicom3\
        \\pynetdicom3\\apps\\findscu\\cluster\\'
        final_file_cluster_path = os.path.join( cluster_path, str(identifier)+"_final")
        os.rename(path2,final_file_cluster_path)
        examination = Catalog.get(identifier)
        #Zmiana statusu w bazie danych
        examination.status=Status.build_final_image
        Catalog.commit()
        UI.communique("Final image construction in progress")
        return final_file_cluster_path


class Scanner():
        """
        Klasa implementująca funkcje związane z komunikacją ze skanerem
        """

    def scan(identifier):
        """
```

```
    Funkcja komunikuje się ze skanerem PET wykorzystując dedykowany niskopoziomowy
    protokół. Wysyła polecenie rozpoczęcia badania wraz z plikiem DICOM opisującym
    przeprowadzaną procedurę.

    :param pat_name: Identyfikator.
    """

    examination = Catalog.get(identifier)
    #Zmiana statusu w bazie danych
    examination.status=Status.scanning
    Catalog.commit()
    UI.communique("Scanning patient")
    return

def scan_status(identifier):
    """
    Funkcja komunikuje się ze skanerem PET i sprawdza status wykonywanej procedury.
    Gdy skan zostaje ukończony, funkcja aktualizuje status badania w systemie.

    :param pat_name: Identyfikator.
    """
    examination = Catalog.get(identifier)
    #Zmiana statusu w bazie danych
    examination.status=Status.finished_scanning
    Catalog.commit()
    UI.communique("The Scan was executed")
    return

def send_scan_results(identifier):
    """
    Wysłanie danych ze skanera PET do kontrolera modalności. Gdy dane zostaną
    wysłane funkcja zmienia status badania w systemie.

    :param pat_name: Identyfikator.
    """

    scanner_pet_path = "C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\\pynetdicom3\
    \\pynetdicom3\\apps\\findscu\\scanner\\"
    controller_path = "C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\\pynetdicom3\
    \\pynetdicom3\\apps\\findscu\\controller\\"
    file_scanner_path = os.path.join( scanner_pet_path, str(identifier))
    file_controller_path=shutil.copy(file_scanner_path,controller_path)
    examination = Catalog.get(identifier)
    #Zapis pliku do bazy danych
    examination.raw_data_file=file_controller_path
    #Zmiana statusu w bazie danych
    examination.status=Status.send_raw_data
    Catalog.commit()
    UI.communique("The results were sent to the controller")
```

```
        return file_controller_path


class Cluster():
        """
        Klasa implementująca komunikację pomiędzy klastrem obliczeniowym, a kontrolerem
        modalności

        """


    def send_input_data(identifier, path):
        """
        Metoda odpowiedzialna za wysyłanie wejściowych danych z kontrolera modalności
        do klastra

        :param pat_name: Identyfikator.
        """

        cluster_path = "C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\\pynetdicom3\
        \\pynetdicom3\\apps\\findscu\\cluster"
        file_cluster_path=shutil.copy(path,cluster_path)
        examination = Catalog.get(identifier)
        #Zmiana statusu w bazie danych
        examination.status=Status.reco_data_ready
        Catalog.commit()
        UI.communique("Downloading input data for reconstruction")
        return file_cluster_path


    def register(identifier):
        """
        Metoda odpowiedzialna za komunikację z klastrem obliczeniowym.
        Zarejestrowanie rekonstrukcji obrazu medycznego

        :param pat_name: Identyfikator.
        """
        #Generowanie identyfikatora zadania na klastrze i zapis do bazy danych
        examination = Catalog.get(identifier)
        #Zmiana statusu w bazie danych
        examination.status=Status.reco_registered
        Catalog.commit()
        UI.communique("Reconstruction registered")
        return


    def start_reconstruction(identifier):
        """
        Metoda ustawiająca dane konkretnego id do systemu kolejkowego w celu
```

```
    rekonstrukcji danych

    :param pat_name: Identyfikator.
    """

    examination = Catalog.get(identifier)
    #Zmiana statusu w bazie danych
    examination.status=Status.reco_queued
    Catalog.commit()
    UI.communique("Reconstruction queued")
    return


def status(identifier):
    """
    Metoda aktualizująca status w bazie danych po rekonstrukcji danych

    :param pat_name: Identyfikator.
    """

    examination = Catalog.get(identifier)
    #Zmiana statusu w bazie danych
    examination.status=Status.reco_finished
    Catalog.commit()
    return


def anonymisation(identifier):
    """
    Metoda implementująca anonimizację danych medycznych
    :param pat_name: Identyfikator.
    """

    examination = Catalog.get(identifier)
    #Zmiana statusu w bazie danych
    examination.status=Status.finished_anonymisation
    Catalog.commit()
    return


def get_output_data(identifier, file_path_cluster):
    """
    Metoda odpowiedzialna za wysłanie wyjściowych plików z klastra obliczeniowego
    do kontrolera modalności

    :param pat_name: Identyfikator.
    """
    final_image_controller = "C:\\Users\\Anna\\Desktop\\pynetdicom_git_clone\
    \\pynetdicom3\\pynetdicom3\\apps\\findscu\\controller\\"
```

```
        shutil.copy(file_path_cluster,final_image_controller)
        examination = Catalog.get(identifier)
        #Zmiana statusu w bazie danych
        examination.status=Status.send_final_data
        Catalog.commit()
        UI.communique("The results have been sent")
        return


def run():
    """
    Rdzen mechanizmu

    """
    #Zapytanie przez UI personel medyczny w formie imienia i nazwiska pacjenta
    name, surname=UI.query()
    #Na UI zwracana jest lista badań pacjentów o podanym imieniu i nazwisku
    list_of_examinations=Controller.find_study(name,surname)
    #Personel medyczny wybiera, dla którego pacjenta przygotowywany bedzie protokół
    number=UI.do_study(list_of_examinations)

    #Pętla po elementach w scheduled procedure step sequence
    i=0
    for step in list_of_examinations[number].ScheduledProcedureStepSequence:
        if step.ScheduledStationAETitle=='FILMDIGITIZE':
            break
        i=+1
    if i>len(list_of_examinations[number].ScheduledProcedureStepSequence):
        UI.communique("AE title not found")
        return



    # Tworzenie obiektu badanie, ktory będzie dodany do bazy danych:
    examination_id= Catalog.newstudy(patient_name=str(\
            list_of_examinations[number].PatientName),
            patient_id=list_of_examinations[number].PatientID,
            patients_sex=list_of_examinations[number].PatientSex,
            start_date=datetime.datetime.strptime(list_of_examinations[number].\
            ScheduledProcedureStepSequence[i].ScheduledProcedureStepStartDate,'%Y%m%d'),
            end_date=datetime.datetime.strptime(list_of_examinations[number].\
            ScheduledProcedureStepSequence[i].ScheduledProcedureStepEndDate,'%Y%m%d'),
            aetitle=list_of_examinations[number].ScheduledProcedureStepSequence[i].\
            ScheduledStationAETitle,
            accession_number=list_of_examinations[number].AccessionNumber,
            requested_procedure_id=list_of_examinations[number].RequestedProcedureID,
            final_image=None,
                                raw_data_file=None,
                                path_mpps=None)
```

```
#Pętla przetwarza badanie, aż nie zakończy się poprawnie lub zgłosi błąd. Dla każdego
# stanu wykorzystywana jest odpowiednia funkcja, która taki stan obsługuje.
    loop=True
    while(loop):
        examination=Catalog.get(examination_id)
        if examination.status==Status.new:
    #Przesłanie n-create oraz zmiana statusu na inprogress
            Controller.mpps_inprogress(examination_id)
        elif examination.status==Status.dicom_inprogress:
            #Wykonywany jest skan pacjenta
            Scanner.scan(examination_id)
        elif examination.status==Status.scanning:
            #Sprawdzany jest status skanowania
            Scanner.scan_status(examination_id)
        elif examination.status==Status.finished_scanning:
            #Wysyłanie surowych danych ze skanera do kontrolera modalności
            path_controller=Scanner.send_scan_results(examination_id)
        elif examination.status==Status.send_raw_data:
            #Następuje anonimizacja danych
            Cluster.anonymisation(examination_id)
        elif examination.status==Status.finished_anonymisation:
            #Przez API rejestrowane są nowe zadania obliczeniowe.
            #Klaster ma za zadanie zweryfikować nas, że podaje nam identyfikator
            #zadania i czeka na wrzucenie danych
            Cluster.register(examination_id)
        elif examination.status==Status.reco_registered:
            #Przesyłane sa dane wejściowe z kontrolera modalności na klaster
            path_file_cluster=Cluster.send_input_data(examination_id, path_controller)
        elif examination.status==Status.reco_data_ready:
            #Rekonstrukcja obrazu oczekuje w systemie kolejkowym na klastrze
            #obliczeniowym na przydzielenie zasobów obliczeniowych
            Cluster.start_reconstruction(examination_id)
        elif examination.status==Status.reco_queued or \
        examination.status == Status.reco_running:
            #Po ukończeniu rekonstrukcji następuje aktualizacja statusu w bazie danych
            Cluster.status(examination_id)
        elif examination.status==Status.reco_finished:
            #Wysłanie danych z klastra do kontrolera modalności
            Cluster.get_output_data(examination_id, path_file_cluster)
        elif examination.status==Status.send_final_data:
            # Budowanie końcowego obrazu na klastrze obliczeniowym
            final_file_cluster=Controller.build_final_image(examination_id,\
                                                            path_file_cluster )
        elif examination.status==Status.build_final_image:
            #Wysłanie danych z klastra do kontrolera modalności
            Controller.save_final_data(examination_id,final_file_cluster)
        elif examination.status==Status.procedure_completed:
            #Wysłanie n-create ze statusem completed, co w rzeczywistości oznacza
```

```
            #ukończenie badania
            Controller.mpps_completed(examination_id)
        elif examination.status==Status.dicom_completed:
            #Zakończenie pętli
            loop=False
        elif examination.status==Status.failed:
            #Wysłanie n-create ze statusem discontinued, co oznacza, że badanie
            #zostało anulowane
            Controller.mpps_discontinued(examination_id)
            #Zakończenie pętli
            loop=False
        time.sleep(3)
    return
#Uruchomienie funkcji run
run()
```

Kod skryptu ui.py:

```
import pydicom
import argparse


class UI():
    """
    Klasa implementująca interfejs użytkownika.
    """


    def query(argv=None):
        """
        Metoda, dzięki której personel medyczny przez wpisanie imienia i nazwiska może
        znalezc z posrod wielu pacjentów tego konkretnego

        """
        parser = argparse.ArgumentParser(description='There is User Interface. You can \
                                         find patient or examination. Type in patient \
                                         name and surname')
        Patient_name= parser.add_argument('-n','--name', help='Please, type in patient \
                                          name', required=False)
        Patient_surname= parser.add_argument('-s','--surname', help='Please, type in \
                                             patient surname',required=True)
        args = parser.parse_args()
        print("Finding examination for:  %s %s" % (args.name, args.surname))
        with pydicom.dcmread("worklist_query.dcm", force=True) as ds:
            ds.PatientName=args.name +" "+ args.surname
            print(ds)
            return Patient_name,Patient_surname
```

```
    def do_study(list_of_studies):
        """
        Metoda implementująca komunikację z personelem szpitala, a dokładniej personel
        medyczny może dokonać wyboru, które badanie z listy ma być wykonane

        :param list_of_studies: Lista badań

        """
        while True:
            try:
                confirmation=int(input("Specify number of examination which you want \
                                       to commision: "))
                if confirmation<=int(len(list_of_studies)-1):
                    print("Examination data preparation completed")
                    break
                else:
                    print("Incorrect number of examination, try again")
            #Obsługa błędów
            except (ValueError, UnboundLocalError):
                print("You don't type integer, try again")
            except IndexError:
                print("Incorrect number of examination, try again")
        return confirmation


    def communique(text):
    """
    Metoda odpowiedzialna za wyswietlanie komunikatów dotyczących badania

    """
        print(text)
```

Baza danych catalog.py:

```
#SQLAlchemy jest zbiorem narzędzi do pracy z bazami danych i Pythonem.
#Składa się z narzędzi SQL oraz mapowania obiektowo-relacyjnego -ORM.
#Zbiór bibliotek
from sqlalchemy.orm import sessionmaker
from sqlalchemy import Enum, create_engine, Column, Integer, String, Date
from sqlalchemy.ext.declarative import declarative_base
import enum
from sqlalchemy import DateTime
from ui import UI
#Połączenie z bazą danych SQLite.
sqlite_db = create_engine('sqlite:///memory', echo=True)

#Kontruktor podstawowej klasy do definiowania innych klas
Base = declarative_base()
```

```
#Stworzenie skonfigurowanej klase Sesja
Session = sessionmaker(bind=sqlite_db)
#Tworzenie sesji
session = Session()


class Status(enum.Enum):
    """
    Klasa Status do tworzenia liczbowych zmiennych, w tym przypadku różnych statusów
    badania
    """

    new = 1
    dicom_inprogress=2
    scanning = 3
    finished_scanning = 4
    send_raw_data=5
    reco_registered = 6
    reco_data_ready=7
    reco_queued=8
    reco_running = 9
    reco_finished=10
    finished_anonymisation=11
    build_final_image=12
    send_final_data=13
    dicom_completed=14
    procedure_completed=15
    failed=16


class Study(Base):
    """
    Klasa implementująca kolumny w bazie danych oraz określająca ich typ
    """

    __tablename__ = 'Study'
    id = Column(Integer, primary_key=True, unique=True,nullable=False)
    patient_name=Column(String)
    patient_id=Column(String)
    patients_sex=Column(String)
    start_date = Column(DateTime)
    end_date = Column(DateTime)
    aetitle=Column(String)
    accession_number=Column(String)
    requested_procedure_id=Column(String)
    status = Column(Enum(Status),default=Status.new)
    raw_data_file=Column(String)
    final_image=Column(String)
    path_mpps=Column(String)
```

```
    def __repr__(self):
        """
        Metoda okreslająca w jaki sposób prezentowane mają być dane

        """
        return "<{'%s':('%s','%s', '%s','%s','%s',,'%s','%s','%s','%s','%s','%s','%s')}>"\
    % (self.id,self.patient_name,self.patient_id, self.patients_sex, self.start_date, \
       self.end_date, self.aetitle, self.accession_number, self.requested_procedure_id, \
       self.status, self.final_image, self.raw_data_file,self.path_mpps)

Base.metadata.create_all(sqlite_db)

class Catalog:
    """
    Klasa Catalog, w której definiowane są funkcje, dzięki którym można utworzyć
    nowe badanie, zatwierdzić sesję, pobrać badanie, filtrować wyniki m.in. po
    statusach, usunąć badanie.
    """

    def get(number):
    """
    Metoda odpowiedzialna za pobieranie badania o identyfikatorze id
    """
        return session.query(Study).filter(Study.id==number).one()


    def commit():
    """
    Metoda odpowiedzialna za zatwierdzanie sesji
    """
        session.commit()


    def newstudy(patient_name,
                 patient_id,
                 patients_sex,
                 start_date,
                 end_date,
                 aetitle,
                 accession_number,
                 requested_procedure_id,
                 final_image,
                 raw_data_file,
                 path_mpps):
    """
    Metoda odpowiedzialna za tworzenie nowego badania

    :param patient_name: Imię i nazwisko pacjenta,
```

```
:param patient_id: Numer identyfikacyjny pacjenta,
:param patients_sex: Płeć,
:param start_date: Data startu,
:param end_date: Data zakończenia,
:param aetitle: Tytuł AE,
:param accession_number: Numer dostępu,
:param requested_procedure_id: Numer identyfikacyjny procedury,
:param final_image: Obraz końcowy,
:param raw_data_file: Surowe dane,
:param path_mpps: Plik MPPS.
"""

    #Tworzenie nowego rekordu w bazie danych
    new_study_record=Study(patient_name=patient_name,
                           patient_id=patient_id,
                           patients_sex=patients_sex,
                           start_date=start_date,
                           end_date=end_date,
                           aetitle=aetitle,
                           accession_number=accession_number,
                           requested_procedure_id=requested_procedure_id,
                           final_image=final_image,
                           raw_data_file=raw_data_file,
                           path_mpps=path_mpps)
    #Dodawanie nowego rekordu do bazy danych
    session.add(new_study_record)
    #Zatwierdzanie sesji
    session.commit()
    return new_study_record.id


def return_all():
    """
    Metoda zwracająca wszystkie rekordy z bazy danych

    """
    return session.query(Study).all()


def return_with_status(study_status):
    """
    Metoda zwracająca rekordy w zależności od statusu

    """

    return session.query(Study).filter_by(study_status).all()


def delete_study(number):
```

```
        """
        Metoda pozwalająca usuwać rekordy w zalezności od wskazanego id

        """
        session.query(Study).filter(Study.id==number).delete()
        session.commit()
        UI.comunique("You deleted patient nr ",number)
```

Proponowane API REST

Przykładem najprostszego joba w formacie JSON jest komenda zawierająca: komentarz, priorytet, dane wejściowe, dane wyjściowe, parametry oraz pliki.

```
{
    "comments":"some-comment",
    "priority":"1",
    "input_data":[ ],
    "output_data":[ ],
    "parameters":"some-params",
    "files":[ ]
}
```

1) Obsługa zadań obliczeniowych

Przykładowe API składające się z następujących punktów końcowych, które stanowią polecenia do wykonania przez system zdalny, odpowiedzialne za obsługę zadań obliczeniowych.

Tworzenie zadania

Tworzy nowe zadanie obliczeniowe

URL

/jobs

Metoda

POST

Parametry

```
    {
            "comments": "komentarz zadania",
            "priority": "1",
            "parameters": {
                "parametr1": "wartosc",
                "parametr2": "wartosc"
            }
        }
```

Odpowiedź

identyfikator zadania

Pobieranie listy zadań
Zwraca listę zadań obliczeniowych klienta identyfikowanego przez certyfikat X.509
URL
/jobs
Metoda
GET
Parametry
count - rozmiar listy zadań do zwrócenia
cursor - pozycja kursora listy zadań od której mają być zwracane wyniki
Odpowiedź

```
[
        {
                "id": "identyfikator",
                "status": "status zadania"
        },
        {
                "id": "identyfikator",
                "status": "status zadania"
        }
]
```

Pobieranie zadania po <id>
Zwraca informacje o zadaniu obliczeniowym o identyfikatorze <id>
URL
/jobs/<id>
Metoda
GET
Odpowiedź

```
{
        "id": "identyfikator",
        "status": "status zadania",
        "priorytet": "1",
        "parameters": {
                "parametr1": "wartosc",
                "parametr2": "wartosc"
        },
        "comments": "komentarz zadania"
}
```

Pobieranie pliku dziennika zadania po <id>
Zwraca zawartość pliku dziennika zadania obliczeniowego o identyfikatorze <id>
URL
/jobs/<id>/log
Metoda
GET
Odpowiedź
zawartość pliku dziennika

Pobieranie statusu zadania o <id>
Zwraca status zadania obliczeniowego o identyfikatorze <id>
URL
/job/<id>/status
Metoda
GET
Odpowiedź
zwraca status zadania dla podanego zadania o identyfikatorze <id>

Rozpoczęcie nowego zadania o <id>
Zaczyna nowe zadanie obliczeniowe o identyfikatorze <id>
URL
/jobs/<id>/start
Metoda
POST

Zatrzymanie zadania o <id>
Zatrzymuje aktywne zadanie obliczeniowe o identyfikatorze <id>
URL
/jobs/<id>/kill
Metoda
POST

Usuwanie zadania o <id>
Usuwa zadanie obliczeniowe z systemu o identyfikatorze <id>wraz z plikami wynikowymi. Jeśli jest aktywne zadanie zostanie zatrzymane.
URL
/jobs/<id>
Metoda
DELETE

2) Zarządzanie danymi
Przykładowe punkty końcowe stanowiące polecenia do wykonania przez system zdalny odpowiedzialne za zarządzanie danymi.

Przesłanie danych wejściowych
Pozwala na przesłanie danych wejściowych dla zadania obliczeniowego o identyfikatorze <id>
URL
/jobs/<id>/input
Metoda
POST

Pobieranie listy plików
Zwraca listę plików, które są plikami wynikowymi zadania obliczeniowego o identyfikatorze <id>
URL
/jobs/<id>/output
Metoda
GET

Odpowiedź

```
[
        {
                "id": "nazwa pliku",
                "size": "rozmiar pliku w bajtach"
        }
]
```

Pobieranie pliku wynikowego

Zwraca zawartość pliku wynikowego <file>zadania obliczeniowego o identyfikatorze <id>

URL

/jobs/<id>/output/<file>

Metoda

GET