**„Architektura SIPWW”**

**Załącznik nr 1**

**do**

**Szczegółowego opisu przedmiotu zamówienia**

„Dostawa, instalacja, konfiguracja i uruchomienie infrastruktury teleinformatycznej z oprogramowaniem standardowym i systemowym, dostarczenie, konfiguracja i wdrożenie składników aplikacyjnych GIS, opracowanie i zasilenie bazy danych tematycznych oraz metadanych SIPWW, przeprowadzenie szkoleń w zakresie obsługi dostarczonych komponentów systemu (infrastruktury teleinformatycznej i oprogramowania)”.

# Spis treści

[Spis treści 2](#_Toc7453026)

[1. Słownik pojęć i skrótów 6](#_Toc7453027)

[2. Architektura SOA i szyna usług 7](#_Toc7453028)

[2.1. Architektura SOA i szyna usług 7](#_Toc7453029)

[3. Architektura technologiczna 11](#_Toc7453030)

[4. Komponenty aplikacyjne 13](#_Toc7453031)

[4.1. ADM 16](#_Toc7453032)

[4.1.1. Oprogramowanie zarządzające systemem 17](#_Toc7453033)

[4.2. CMS 17](#_Toc7453034)

[4.2.1. Portal informacyjny 17](#_Toc7453035)

[4.3. Desktop GIS 18](#_Toc7453036)

[4.3.1. Desktop GIS 19](#_Toc7453037)

[4.4. ESB 19](#_Toc7453038)

[4.4.1. Szyna usług 20](#_Toc7453039)

[4.5. GISServer 20](#_Toc7453040)

[4.5.1. Serwer danych przestrzennych 21](#_Toc7453041)

[4.6. META 21](#_Toc7453042)

[4.6.1. Edytor metadanych 21](#_Toc7453043)

[4.6.2. Serwer metadanych 22](#_Toc7453044)

[4.7. MOBI 22](#_Toc7453045)

[4.7.1. Aplikacja mobilna 23](#_Toc7453046)

[4.8. PZGIK 23](#_Toc7453047)

[4.8.1. System zarządzania PZGIK 24](#_Toc7453048)

[4.9. RDBMS 24](#_Toc7453049)

[4.9.1. RDBMS - baza zewnętrzna 25](#_Toc7453050)

[4.9.2. RDBMS - baza wewnętrzna 25](#_Toc7453051)

[4.9.3. RDBMS testowy 26](#_Toc7453052)

[4.10. WEB 26](#_Toc7453053)

[4.10.1. Mapowy interfejs integracyjny 27](#_Toc7453054)

[4.10.2. Przeglądarka danych przestrzennych 27](#_Toc7453055)

[4.10.3. Zaawansowana przeglądarka danych przestrzennych 27](#_Toc7453056)

[4.11. Infrastruktura oprogramowania 27](#_Toc7453057)

[4.12. Systemy operacyjne 29](#_Toc7453058)

[4.12.1. System Operacyjny 30](#_Toc7453059)

[5. Logiczna infrastruktura sprzętowa 32](#_Toc7453060)

[5.1. Środowisko przetwarzania 32](#_Toc7453061)

[5.2. Środowisko składowania (dane plikowe) 33](#_Toc7453062)

[5.3. Środowisko składowania (RDBMS) 33](#_Toc7453063)

[5.4. Środowisko składowania (Wirtualizacja) 33](#_Toc7453064)

[5.5. Środowisko wirtualizacji 33](#_Toc7453065)

[5.5.1. Wymagania dla oprogramowania do wirtualizacji systemów. 33](#_Toc7453066)

[5.6. ADM 33](#_Toc7453067)

[5.6.1. adm-01 33](#_Toc7453068)

[5.6.2. t-adm-01 34](#_Toc7453069)

[5.7. CMS 34](#_Toc7453070)

[5.7.1. cms-01 34](#_Toc7453071)

[5.7.2. cms-02 34](#_Toc7453072)

[5.7.3. t-cms-01 34](#_Toc7453073)

[5.8. DesktopGIS 34](#_Toc7453074)

[5.8.1. desktopGIS-01 34](#_Toc7453075)

[5.8.2. t-desktopGIS-01 34](#_Toc7453076)

[5.9. ESB 34](#_Toc7453077)

[5.9.1. esb-01 35](#_Toc7453078)

[5.9.2. esb-02 35](#_Toc7453079)

[5.9.3. esb-03 35](#_Toc7453080)

[5.9.4. esb-04 35](#_Toc7453081)

[5.9.5. esb-05 35](#_Toc7453082)

[5.9.6. esb-06 35](#_Toc7453083)

[5.9.7. t-esb-01 35](#_Toc7453084)

[5.10. GISServer 35](#_Toc7453085)

[5.10.1. t-gisserver-01 36](#_Toc7453086)

[5.10.2. gisserver-01 36](#_Toc7453087)

[5.10.3. gisserver-02 36](#_Toc7453088)

[5.10.4. gisserver-03 36](#_Toc7453089)

[5.10.5. gisserver-04 36](#_Toc7453090)

[5.10.6. gisserver-05 36](#_Toc7453091)

[5.10.7. gisserver-06 36](#_Toc7453092)

[5.11. META 36](#_Toc7453093)

[5.11.1. meta-01 37](#_Toc7453094)

[5.11.2. meta-02 37](#_Toc7453095)

[5.11.3. t-meta-01 37](#_Toc7453096)

[5.12. PZGIK 37](#_Toc7453097)

[5.12.1. pzgik-01 37](#_Toc7453098)

[5.12.2. pzgik-02 37](#_Toc7453099)

[5.12.3. t-pzgik-01 37](#_Toc7453100)

[5.13. RDBMS 37](#_Toc7453101)

[5.13.1. rdbms-01 37](#_Toc7453102)

[5.13.2. rdbms-02 38](#_Toc7453103)

[5.13.1. t-rdbms-01 38](#_Toc7453104)

[5.14. WEB 38](#_Toc7453105)

[5.14.1. web-01 38](#_Toc7453106)

[5.14.2. web-02 38](#_Toc7453107)

[5.14.3. web-03 38](#_Toc7453108)

[5.14.4. web-04 38](#_Toc7453109)

[5.14.5. t - web-01 38](#_Toc7453110)

[6. Środowisko przetwarzania i składowania 39](#_Toc7453111)

[7. Mapowanie infrastruktury logicznej na fizyczną 42](#_Toc7453112)

# Słownik pojęć i skrótów

|  |  |
| --- | --- |
| **Pojęcie/skrót** | **Wyjaśnienie** |
| Android | system operacyjny z rodziny Linux dla urządzeń mobilnych |
| CMS | ang. Content Management System, system zarządzania treścią |
| CPU | ang. Central Processing Unit |
| EAI | ang. Enterprise Application Integration |
| ESB | ang. Enterprise Service Bus; korporacyjna magistrala usług |
| GIS | ang. Geographic Information System, system informacji geograficznej |
| HDD | ang. Hard Disk Drive, rodzaj pamięci masowej |
| IIS | ang. Internet Information Services (IIS), zbiór usług internetowych dla systemów rodziny Microsoft Windows |
| INSPIRE | ang. Infrastructure for Spatial Information in Europe, Infrastruktura Informacji Przestrzennej |
| iOS | system operacyjny Apple Inc. dla urządzeń mobilnych iPhone, iPod touch oraz iPad |
| IP | ang. IP address |
| J2EE | Java Platform, Enterprise Edition |
| JSON | ang. JavaScript Object Notation, format wymiany danych komputerowych |
| KIIP | Krajowa Infrastruktura Informacji Przestrzennej |
| OGC | ang. Open Geospatial Consortium, międzynarodowa organizacja typu non-profit, zrzeszająca ponad 460 firm, agencji rządowych i uniwersytetów, współpracujących nad rozwijaniem i implementacją otwartych standardów dla danych i usług przestrzennych |
| open source | Otwarte oprogramowanie |
| PKI | ang. Public Key Infrastructure |
| PZGIK | Państwowy Zasób Geodezyjny i Kartograficzny |
| RAM | ang. Random Access Memory, pamięć o dostępie swobodnym |
| RDBMS | ang. Relational Database Management System; system zarządzania relacyjną  bazą danych |
| REST | ang. Representational State Transfer, transfer bezstanowy – protokół usług sieciowych udostępniających bezstanowy mechanizm przesyłania danych z wykorzystaniem protokołu HTTP. |
| RIIP | Regionalna Infrastruktura Informacji Przestrzennej |
| SIPWW | System Informacji Przestrzennej Województwa Wielkopolskiego |
| SOA | ang. Service-Oriented Architecture, architektura zorientowana na usługi |
| SOAP | ang. Simple Object Access Protocol, protokół wywołania zdalnego dostępu do obiektów – protokół używający XML do kodowania transmisji. |
| System zarządzania PZGIK | w rozumieniu niniejszego postępowania: System zarządzania częścią wojewódzką PZGiK, wykorzystywany przez WODGiK |
| TOGAF | ang. The Open Group Architecture Framework, szkielet architektury korporacyjnej, zapewniający kompleksowe podejście do projektowania, planowania, implementacji oraz zarządzania informacyjną architekturą organizacji |
| UML | ang. Unified Modeling Language |
| UMWW | Urząd Marszałkowski Województwa Wielkopolskiego w Poznaniu |
| W3C | ang. World Wide Web Consortium, organizacja, która zajmuje się ustanawianiem standardów pisania i przesyłu stron WWW |
| WMS | ang. Web Map Service; protokół usługi sieciowej map standardu OGC. |
| WMTS | ang. Web Map Tile Service; usługa sieciowa kafli map |
| WODGiK | Wojewódzki Ośrodek Dokumentacji Geodezyjnej i Kartograficznej w Poznaniu |
| XML | ang. Extensible Markup Language, rozszerzalny język znaczników – uniwersalny język definiowania (reprezentowania) danych w ustrukturalizowany sposób |

# Architektura SOA i szyna usług

## Architektura SOA i szyna usług

Jednym z podstawowych założeń przy budowie systemu SIPWW jest zaprojektowanie systemu zgodnego z modelem architektury zorientowanej na usługi (*Service-Oriented Architecture*). Zdefiniowane i dostarczane usługi muszą spełniać wymagania użytkowników, zarówno wewnętrznych jak i zewnętrznych systemu. Natomiast organizacja systemu w postaci usług ma za zadanie obniżyć koszty, dając:

* możliwość poprawnego monitorowania dostarczanych usług,
* efektywną wymianę danych pomiędzy elementami systemu oraz systemami zewnętrznymi,
* możliwość ponownego wykorzystania usług,
* wymienialność elementów systemu na inny zgodny z ustalonym interfejsem.

Zgodnie z tymi założeniami każda z warstw posiadająca określone komponenty aplikacyjne musi mieć możliwość niezależnego skalowania w zależności od wykorzystania, rozwoju oraz obciążenia systemu. Ponadto część wewnętrzna musi być niezależna od części dostępnej publicznie. Niedopuszczalna jest sytuacja, gdy użytkownik wewnętrzny korzystający z sytemu nie ma możliwości wykonywania powierzonych mu obowiązków z powodu zbytniego obciążenia systemu przez użytkowników zewnętrznych, bądź próbami ataku na infrastrukturę lub komponenty aplikacyjne systemu. Oprócz wsparcia w zakresie wspomagania administratora systemu na taką ewentualność system musi posiadać możliwość rozbudowy o kolejne świadczone usługi, a także zwiększenia wydajności dla już istniejących usług.

W celu sprostania tym wymaganiom została stworzona architektura EAI (integracja aplikacji korporacyjnych). Korzysta ona z dobrodziejstw modelu SOA wprowadzając centralny element pośredniczący w komunikacji pomiędzy wszystkimi wyróżnionymi komponentami systemu – szynę usług (Enterprise Service Bus, ESB). W przypadku zaistnienia potrzeby wymiany lub aktualizacji jednego z komponentów systemu, administrator posiada jedno miejsce, w którym dokonuje rekonfiguracji, a wszystkie komponenty zależne korzystające z niego, odwołują się do jednego wspólnego punktu dostępowego. Możliwa jest pełna kontrola nad udostępnioną wersją – płynne przełączanie udostępnianej funkcjonalności bez potrzeby rekonfiguracji komponentów zależnych.



Rysunek nr: 1 **Integracja aplikacji korporacyjnych przez szynę danych.**

Szyna usług musi zapewnić podstawowe wymagania:

* transparentność lokalizacji - centralnie konfigurowany punkt końcowy (endpoint), aplikacja wykorzystująca danych interfejs nie wymaga informacji od dostarczyciela usługi, w celu poprawnej komunikacji, przesyłania wiadomości,
* ulepszanie - możliwość pobierania brakujących danych w przesyłanych wiadomościach w celu ich dołączenia do komunikatu i przesłanie do punktu docelowego,
* transformacja - zdolność konwersji wiadomości pomiędzy formatami - na potrzeby użyteczności przez aplikację konsumenta,
* monitoring - możliwość monitorowania pracy systemu, przepływów, treści przesyłanych wiadomości,
* konwersja protokołu - funkcjonalność transformacji w stosunku do wspieranych/ obsługiwanych protokołów, możliwość konwersji pomiędzy wymaganymi formatami,
* bezpieczeństwo - zapewnienie bezpiecznego przetwarzania wiadomości, ale również wsparcie w zakresie negocjacji zasad bezpieczeństwa pomiędzy systemami,
* routing - możliwość dowolnej modyfikacji punktu końcowego na zasadach statycznych, ale również dynamicznych.

Ze względu na charakterystykę systemu klasy GIS musi ponadto wspierać:

* przesyłanie danych binarnych – wygenerowanych rastrów map,
* musi działać w trybie synchronicznym – dostarczając w jak najkrótszym czasie odpowiedź z serwera danych przestrzennych do aplikacji-klienta użytkownika systemu.

Architektura rozwiązania musi spełniać wymagania wielowarstwowości:

* Każdy z dostępnych węzłów funkcjonalnych; warstwy logicznej musi być zrealizowany w architekturze active-active – awaria jednego z węzłów nie powoduje zaprzestania świadczenia usług,
* osadzenie komponentów aplikacyjnych realizujących funkcjonalności w oparciu o przeglądarkę internetową użytkownika musi być osadzone w warstwie frontend, są to aplikacje realizujące prawidłowe serwowanie i obsługę interfejsu użytkownika,
* funkcjonalność komponentu aplikacyjnego szyny usług musi posiadać możliwość udostępniania danych dla serwerów frontend, ale nie dając bezpośredniego dostępu do usług warstwy backend (middleware),
* w warstwie serwerów aplikacyjnych odpowiedzialnych za realizację logiki biznesowej poszczególne usługi wspomagające muszą być wykorzystywane z udziałem szyny usług,
* warstwa składowania danych wykorzystywana jest przez serwery warstwy backend.

# Architektura technologiczna

Architektura technologiczna przedstawia podział technologiczny wraz z wyborem klas technologii w celu realizacji usług aplikacyjnych oraz magazynów danych (tj. elementów architektury systemów informatycznych).

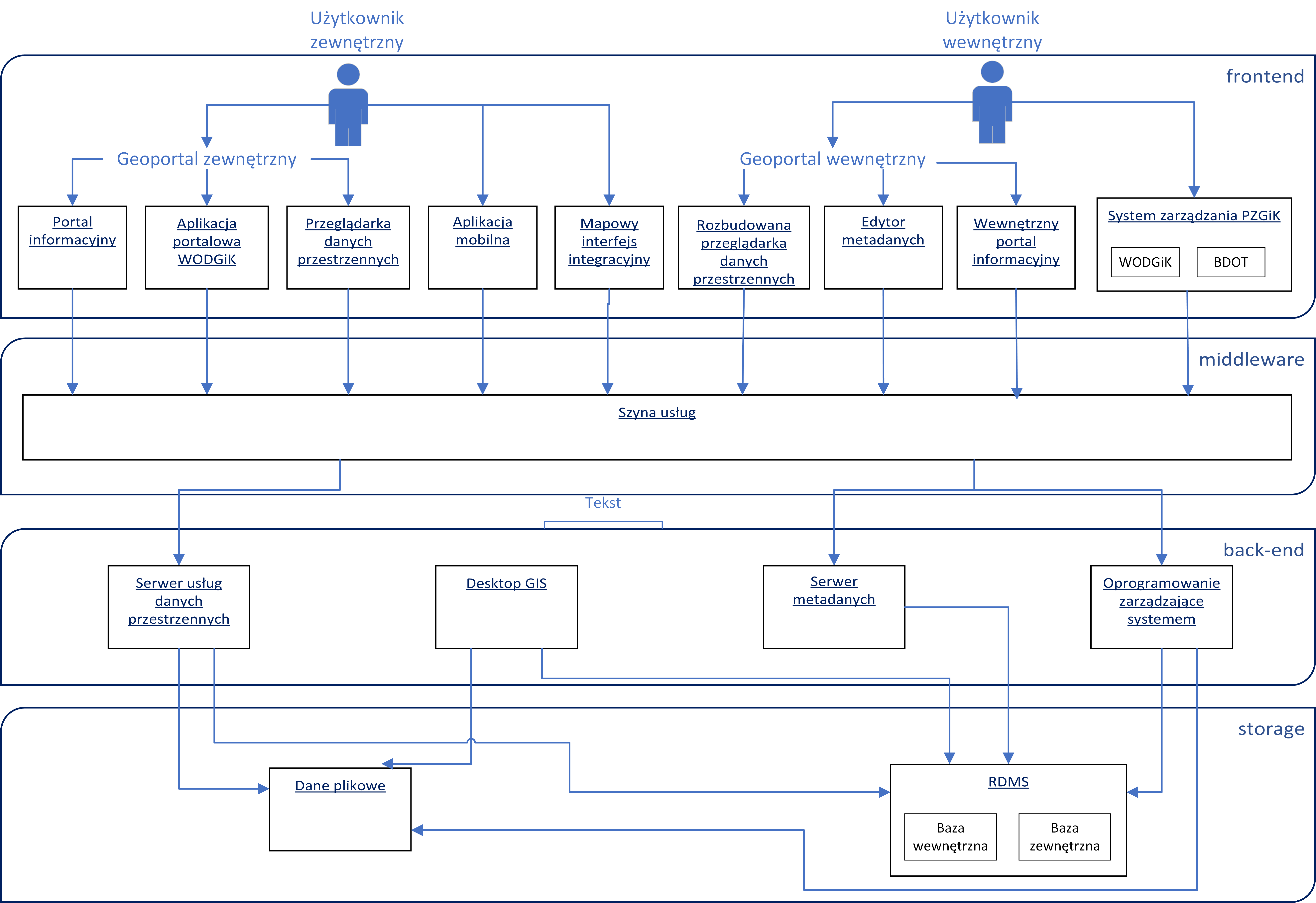
Architektura technologiczna pokazuje jak elementy funkcjonalne systemu są realizowane, a więc przy pomocy jakich komponentów technologicznych i jakiej infrastruktury programowej, wirtualizacyjnej, sprzętowej oraz sieciowej przeprowadza się wdrożenie funkcjonalności systemu.

**Metoda opisu**

Architektura technologiczna jest logiczną konsekwencją decyzji i wymagań opisanych w architekturze biznesowej i systemów informatycznych. Jej opis składa się z następujących elementów:

1. **Komponenty aplikacyjne** – komponenty technologiczne konieczne do dostarczenia, które zawierają funkcjonalności systemu (np. aplikacja webowa wykonana w technologii J2EE zawierająca funkcjonalności usługi aplikacyjnej <nazwa oprogramowania>).
2. **Infrastruktura oprogramowania** – komponenty technologiczne, aby poprawnie funkcjonować muszą korzystać z infrastruktury standardowej, np. serwera aplikacji, systemu operacyjnego i są instalowane w węzłach Logicznej infrastruktury sprzętowej.
3. **Logiczna infrastruktura sprzętowa** – określająca klasę i właściwości logicznych komponentów sprzętowych, na których instalowane jest oprogramowanie standardowe i komponenty aplikacyjne z uwzględnieniem rozwiązań wirtualizacyjnych i ich właściwości.
4. **Fizyczna infrastruktura sprzętowa** – określająca klasę i właściwości sprzętu, na którym instalowane są komponenty wyższych warstw architektury technologicznej.
5. **Infrastruktura sieci** **komputerowej** – określająca różne obszary bezpieczeństwa, wydzielone fizycznie lub logicznie dla poprawnego funkcjonowania komponentów sprzętowych i  programowych.

Poniżej przedstawiono ogólny diagram organizacji architektury technologicznej:



Rysunek nr: 2 *Architektura technologiczna*

# 

# Komponenty aplikacyjne

**Komponenty aplikacyjne** - komponenty technologiczne konieczne do dostarczenia, które zawierają funkcjonalności systemu (np. aplikacja webowa wykonana w technologii J2EE zawierająca funkcjonalności usługi aplikacyjnej <nazwa oprogramowania>).



Rysunek nr: 3**Komponenty aplikacyjne**

Diagram zawiera komponenty aplikacyjne, które zostały zgrupowane ze względu na swoją funkcjonalność w węzłach instalacji. Każdy węzeł instalacji w zależności od potrzeb może być niezależnie skalowany w ramach rozbudowy systemu.

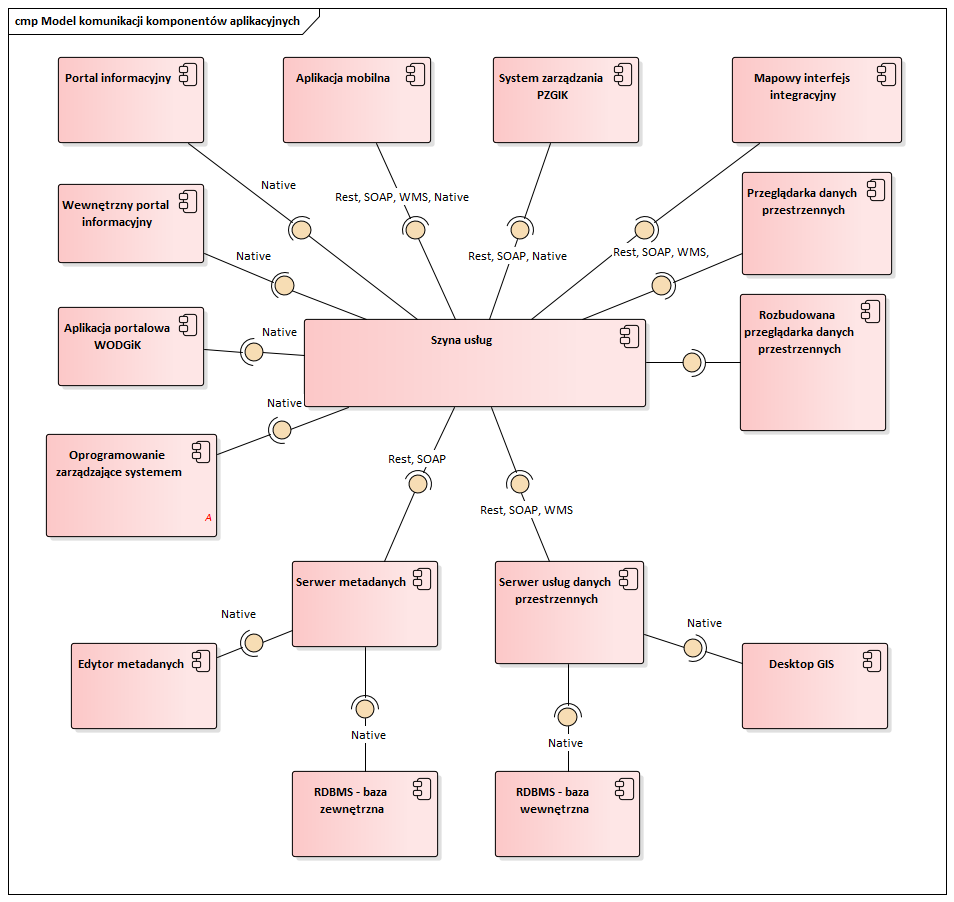
Budowany system powinien być zgodny z zasadą niezależności technologicznej w konsekwencji umożliwiając funkcjonowanie na różnych platformach technologicznych.

Niezależność aplikacji od konkretnych technologii jest efektywna w przypadku ich budowy, utrzymania, aktualizacji, a nawet wycofywania z eksploatacji.

* Aplikacje powinny stosować standardy, które są przenaszalne (umożliwiając przeniesienie aplikacji na inną platformą technologiczną),
* Dostęp do systemów, które wymagają zastosowania specyficznej technologii, powinien być zagwarantowany na zasadzie budowy interfejsów dostępu niezależnych technologicznie,
* Aplikacje powinny stosować rozwiązania typu middleware w celu separacji specyficznych rozwiązań od pozostałych aplikacji,
* Rekomenduje się powszechne i konsekwentne stosowanie standardów otwartych:
* W zakresie projektowania architektury – metodyka TOGAF,
* W zakresie modelowania systemów – notację UML,
* W zakresie wymiany danych – standard XML, JSON,
* W zakresie warstwy sieciowej – protokół IP,
* W zakresie bezpieczeństwa – rozwiązania PKI, standard XML Signature, standard XML Encryption.

Z uwagi na fakt, iż SIPWW stanowić będzie system o architekturze zorientowanej na usługi (ang. Service Oriented Architecture – SOA):

* SIPWW należy traktować jako kolekcję autonomicznych jednostek przetwarzania zintegrowanych siecią komunikacyjną, a punkty przetwarzania i tworzące je komponenty usług identyfikować jako tzw. węzły, które powinny komunikować się ze sobą za pomocą standardowych, zdefiniowanych usług (komunikatów);
* SIPWW musi posiadać cechy zapewniające skalowalność rozwiązania oraz niezbędny stopień autonomii, tak, aby wyłączenie i brak dostępności jednego z węzłów lub wystąpienie stanu błędu w danym węźle nie wpływało na działanie pozostałej części sieci systemu – unieruchamiając inne węzły;
* każda logicznie wydzielona część SIPWW musi być udostępniana jako usługa zarówno w komunikacji zewnętrznej, jak i wewnętrznej;
* usługi muszą być zdefiniowane w oparciu o przyjęte standardy (W3C, OASIS) zapewniając interoperacyjność projektowanego Systemu, co w szczególności dotyczy zastosowania specyfikacji Open Geospatial Consortium (OGC);
* usługi sieciowe SIPWW muszą być udostępnione zgodnie ze specyfikacjami Web Services (SOAP/HTTP) lub REST;
* sposób komunikacji - interfejs usługi nie może wskazywać na szczegóły implementacji usługi i powinien być niezależny od platformy systemowej, na której usługa jest osadzona;
* usługi SIPWW, dla których wymagane jest monitorowanie (np. usługi sieciowe INSPIRE) muszą być udostępniane poprzez jednorodny interfejs szyny usług (ESB), która musi zapewnić monitorowanie wybranych parametrów usług oraz zdarzeń w czasie rzeczywistym, umożliwiając ich rozliczalność, co wiąże się z zapewnieniem persystencji statystyk działania Systemu;
* SIPWW musi zapewnić zarządzanie usługami, w tym możliwość dynamicznego dodawania nowych usług do rejestru, ich przekierowania (routingu), czy też usuwania z rejestru usług;
* usługi muszą być zabezpieczone przez mechanizm uwierzytelnienia klienta usługi oraz autoryzacji usługi, działające jako odrębne usługi SIPWW, przy czym uwierzytelnienie i autoryzacja muszą być zaimplementowane w oparciu o Infrastrukturę Klucza Publicznego przy użyciu uznanych standardów np. WS-Security.



Rysunek nr: 4**Model komunikacji komponentów aplikacyjnych**

Diagram prezentuje typowy model komunikacji projektowanych komponentów z przykładowymi często stosowanymi protokołami. Z uwagi na zachowanie neutralności technologicznej nie można w tym modelu wskazać konkretnych rozwiązań, dlatego powyższy diagram jest wyłącznie modelem poglądowym.

Standardy komunikacji:

* SOAP-ang. Simple Object Access Protocol; protokół wywołania zdalnego dostępu do obiektów – protokół używający XML do kodowania transmisji.
* Rest-ang. Representational State Transfer; transfer bezstanowy – protokół usług sieciowych udostępniających bezstanowy mechanizm przesyłania danych z wykorzystaniem protokołu HTTP.
* WMS- ang. Web Map Service; protokół usługi sieciowej map standardu OGC.
* Native-wewnętrzny protokół wywołania zdalnego dostępu do obiektów lub usług sieciowych.

## ADM

Węzeł instalacji oprogramowania ADM.

W ramach węzła ADM występują następujące komponenty aplikacyjne:



Rysunek nr: 5**ADM Realizacja**

### Oprogramowanie zarządzające systemem

Komponent oprogramowania aplikacyjnego umożliwiający zarządzanie systemem.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany per instalacja.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Oprogramowanie zarządzające systemem

## CMS

Węzeł instalacji oprogramowania CMS.

W ramach węzła CMS występują następujące komponenty aplikacyjne:



Rysunek nr: 6**CMS Realizacja**

### Portal informacyjny

Komponent oprogramowania aplikacyjnego implementujący portal informacyjny.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany per instalacja.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Portal informacyjny

## Desktop GIS

Węzeł instalacji oprogramowania Desktop GIS. Węzeł dedykowany do udostępniania funkcjonalności uprawnionym, zdalnym użytkownikom poprzez funkcję zdalnego pulpitu.

W ramach węzła Desktop GIS występują następujące komponenty aplikacyjne:



Rysunek nr: 7**Desktop GIS Realizacja**

### Desktop GIS

Komponent oprogramowania aplikacyjnego dostarczający specjalistyczne narzędzia GIS. Oprogramowanie typu desktop przystosowane do dostępu zdalnego, bez konieczności instalacji oprogramowania aplikacyjnego na zdalnym komputerze (dostęp przez zdalny pulpit).

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Desktop GIS

## ESB

Węzeł instalacji oprogramowania ESB.

W ramach węzła ESB występują następujące komponenty aplikacyjne:



Rysunek nr: 8**ESB Realizacja**

### Szyna usług

Komponent odpowiada za

* publikację usług na szynie,
* uwierzytelnianie i autoryzację systemów wykorzystujących usługi szyny,
* definicję przepływów (flow) pomiędzy opublikowanymi usługami a wewnętrzną ich implementacją w systemie zapewniając nie tylko realizacje routingu, ale również ogólniejszą koncepcję koordynacji usług (np. jedno wywołanie usługi na szynie jest rozwijane w kilka wywołań wewnętrznych usług systemu),
* transformację wymienianych komunikatów.
* Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany per instalacja.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

* + 1. Szyna usług

## GISServer

Węzeł instalacji oprogramowania GISServer (serwer danych przestrzennych).

W ramach węzła GISServer występują następujące komponenty aplikacyjne:



Rysunek nr: 9**GISServer Realizacja**

### Serwer danych przestrzennych

Komponent oprogramowania aplikacyjnego implementujący serwer usług danych przestrzennych.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Serwer danych przestrzennych

## META

Węzeł instalacji oprogramowania META.

W ramach węzła META występują następujące komponenty aplikacyjne:



Rysunek nr: 10**Meta Realizacja**

### Edytor metadanych

Komponent oprogramowania aplikacyjnego implementujący klienta usług metadanych przestrzennych.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Edytor metadanych

### Serwer metadanych

Komponent oprogramowania aplikacyjnego implementujący serwer usług metadanych przestrzennych.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Serwer metadanych

## MOBI

Węzeł instalacji oprogramowania MOBI.

W ramach węzła MOBI występują następujące komponenty aplikacyjne:



Rysunek nr: 11**MOBI Realizacja**

### Aplikacja mobilna

Komponent oprogramowania aplikacyjnego dla urządzeń mobilnych z systemem Android, iOS. Produkt dystrybuowany przez sklep z aplikacjami online.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Aplikacja mobilna

## PZGIK

Węzeł instalacji oprogramowania PZGIK.

W ramach węzła PZGIK występują następujące komponenty aplikacyjne:



Rysunek nr: 12**PZGIK Realizacja**

### System zarządzania PZGIK

Komponent oprogramowania aplikacyjnego implementujący system zarządzania PZGIK.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany per instalacja.

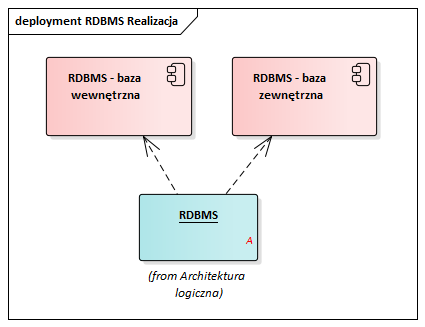
Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. System zarządzania PZGIK

## RDBMS

Węzeł instalacji oprogramowania RDBMS.

W ramach węzła RDBMS występują następujące komponenty aplikacyjne:



Rysunek nr: 13**RDBMS Realizacja**

### RDBMS - baza zewnętrzna

Komponent oprogramowania aplikacyjnego implementujący serwer bazy danych udostępnianych dla użytkowników zewnętrznych.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. RDBMS

### RDBMS - baza wewnętrzna

Komponent oprogramowania aplikacyjnego implementujący serwer bazy danych chronionych i nieudostępnianych użytkownikom zewnętrznym.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. RDBMS

### RDBMS testowy

Komponent oprogramowania aplikacyjnego implementujący serwer bazy danych do celów testowych.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach dla modułu logicznego.

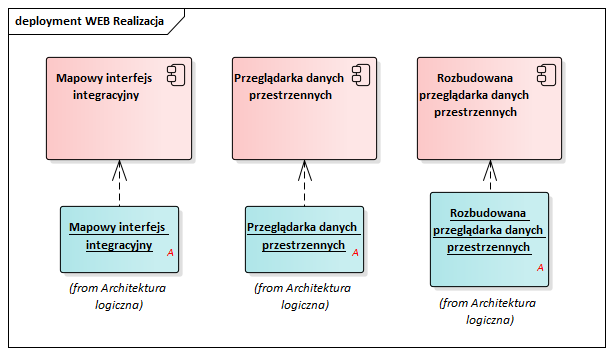
Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. RDBMS

## WEB

Węzeł instalacji oprogramowania WEB.

W ramach węzła WEB występują następujące komponenty aplikacyjne:



Rysunek nr: 14**WEB Realizacja**

### Mapowy interfejs integracyjny

Komponent oprogramowania aplikacyjnego implementujący mapowy interfejs integracyjny.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany per instalacja.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Mapowy interfejs integracyjny

### Przeglądarka danych przestrzennych

Komponent oprogramowania aplikacyjnego implementujący przeglądarkę danych przestrzennych.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany per instalacja.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Przeglądarka danych przestrzennych

### Zaawansowana przeglądarka danych przestrzennych

Komponent oprogramowania aplikacyjnego implementujący zaawansowaną przeglądarkę danych przestrzennych.

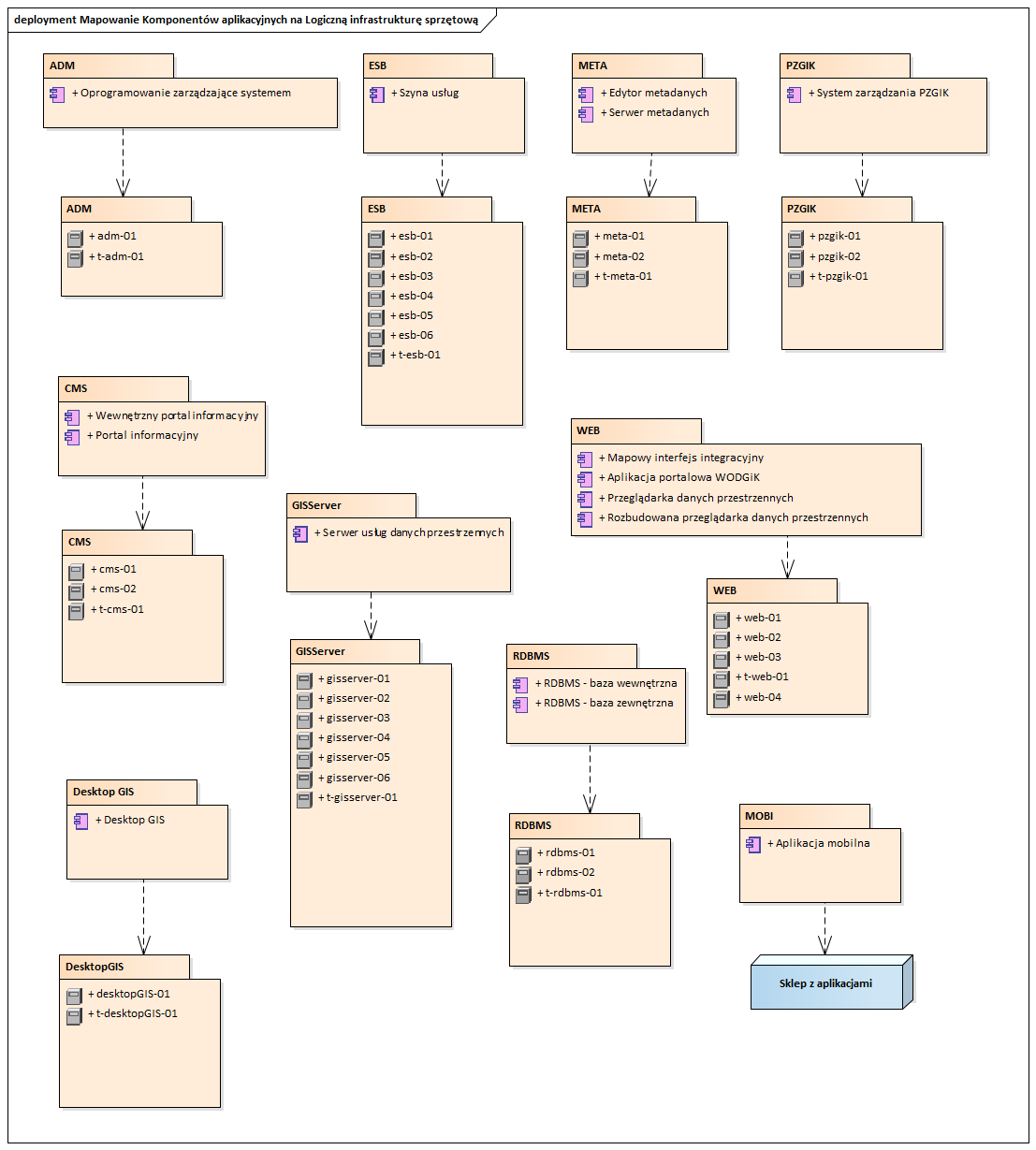
Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany per instalacja.

Oprogramowanie realizuje funkcjonalność następujących elementów architektury logicznej:

1. Rozbudowana przeglądarka danych przestrzennych

## Infrastruktura oprogramowania

**Infrastruktura oprogramowania** – komponenty technologiczne, aby poprawnie funkcjonować muszą korzystać z infrastruktury standardowej, np. serwera aplikacji, systemu operacyjnego i są instalowane w węzłach Logicznej infrastruktury sprzętowej.

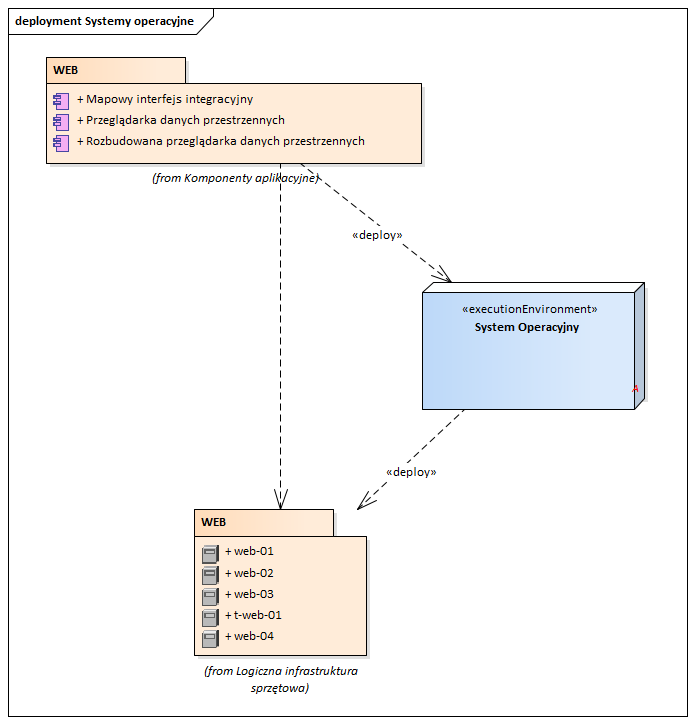


Rysunek nr: 15**Mapowanie Komponentów aplikacyjnych na Logiczną infrastrukturę sprzętową**

Diagram przedstawia zależności pomiędzy węzłami instalacji komponentów aplikacyjnych oraz węzłami logicznej infrastruktury sprzętowej. Dla uczytelnienia odbioru nazwy pakietów węzłów oprogramowania są zgodne z nazwami węzłów logicznej infrastruktury sprzętowej.

## Systemy operacyjne

W praktyce komponenty oprogramowania aplikacyjnego funkcjonują w środowisku systemu operacyjnego. Oprogramowanie systemów operacyjnych musi być uwzględnione w projekcie infrastruktury oprogramowania. Dodatkowo niektóre komponenty oprogramowania aplikacyjnego mogą wymagać wdrożenia dodatkowego oprogramowania np. środowiska wykonywania aplikacji (np. .Net lub Java) i/lub być uruchamiane z oprogramowaniu serwerów aplikacji sieciowych (np. JBoss, WebLogic,IIS). Poniżej przedstawiono poglądowy diagram zależności węzła instalacji oprogramowania aplikacyjnego i instancji infrastruktury sprzętowej od oprogramowania systemu operacyjnego.



Rysunek nr: 16**Systemy operacyjne**

### System Operacyjny

Komponent ten jest reprezentacją systemu operacyjnego. Komponent dostarcza infrastrukturę systemową umożliwiającą działanie wszystkich komponentów aplikacyjnych.

Licencja na produkt dostarczana jest przez Wykonawcę systemu. Typ licencji komercyjna lub open source, produkt powinien być licencjonowany zgodnie ze specyfikacją zawartą w wymaganiach.

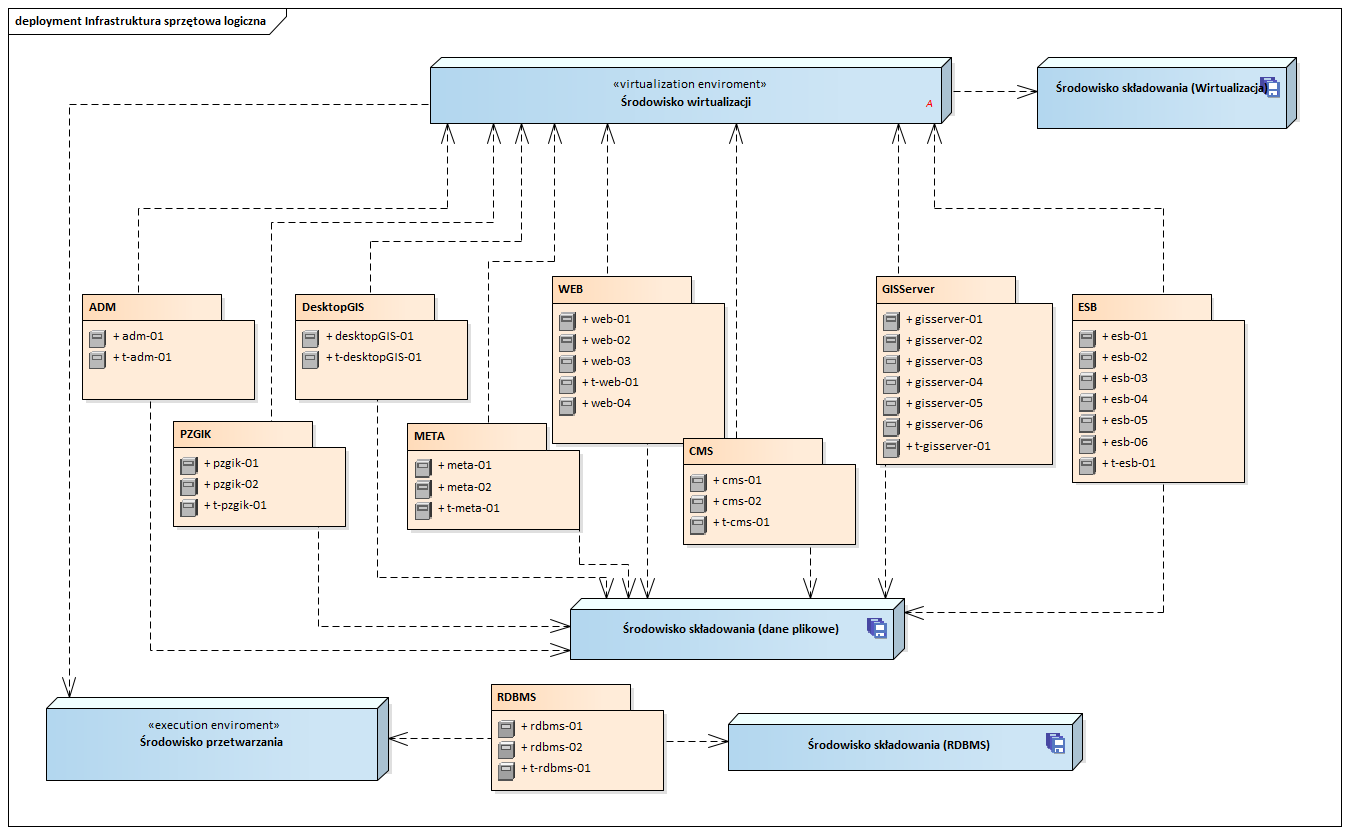
**Wymagania dla oprogramowania systemów operacyjnych:**

Wymagania dla oprogramowania systemów operacyjnych zostały opisane w dokumencie: „Wymagania szczegółowe w zakresie dostawy, instalacji, konfiguracji i uruchomienia infrastruktury teleinformatycznej SIPWW” Załącznik nr 8 do SOPZ w punkcie 5.2.1, 5.3.3, 5.4.1.

# 

# Logiczna infrastruktura sprzętowa

**Logiczna infrastruktura sprzętowa** – określająca klasę i właściwości logicznych komponentów sprzętowych, na których instalowana jest oprogramowanie standardowe i komponenty aplikacyjne z uwzględnieniem rozwiązań wirtualizacyjnych i ich właściwości.



Rysunek nr: 17**Infrastruktura sprzętowa logiczna**

Diagram przedstawia zależności pomiędzy pakietami instancji węzłów instalacji a środowiskiem składowania i przetwarzania.

## Środowisko przetwarzania

Infrastruktura sprzętowa dostępna w ramach systemu SIPWW dla potrzeb przetwarzania danych. Środowisko jest realizowane przez fizyczne urządzenia infrastruktury IT SIPWW.

## Środowisko składowania (dane plikowe)

Infrastruktura sprzętowa dostępna w ramach systemu SIPWW dla potrzeb składowania danych plikowych. Środowisko jest realizowane przez fizyczne urządzania infrastruktury IT SIPWW.

## Środowisko składowania (RDBMS)

Infrastruktura sprzętowa dostępna w ramach systemu SIPWW dla potrzeb składowania danych baz danych. Środowisko jest realizowane przez fizyczne urządzania infrastruktury IT SIPWW. Środowisko testowe realizowane jest wirtualnie.

## Środowisko składowania (Wirtualizacja)

Infrastruktura sprzętowa dostępna w ramach systemu SIPWW dla potrzeb składowania danych środowiska wirtualizacji. Środowisko jest realizowane przez fizyczne urządzania infrastruktury IT SIPWW.

## Środowisko wirtualizacji

Infrastruktura wirtualizacyjna dostępna w ramach systemu SIPWW dla potrzeb przetwarzania danych. Środowisko jest realizowane z wykorzystaniem oprogramowania do wirtualizacji systemów i korzysta z fizycznych zasobów środowiska przetwarzania.

### Wymagania dla oprogramowania do wirtualizacji systemów.

Wymagania dla oprogramowania do wirtualizacji systemów zostały opisane w dokumencie:

„Wymagania szczegółowe w zakresie dostawy, instalacji, konfiguracji i uruchomienia infrastruktury teleinformatycznej SIPWW” Załącznik nr 8 do SOPZ w punkcie 5.2.1.

## ADM

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy ADM.

### adm-01

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### t-adm-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## CMS

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy CMS.

### cms-01

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### cms-02

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### t-cms-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## DesktopGIS

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy DesktopGIS.

### desktopGIS-01

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### t-desktopGIS-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## ESB

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy ESB.

### esb-01

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### esb-02

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### esb-03

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### esb-04

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### esb-05

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### esb-06

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### t-esb-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## GISServer

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy GISServer.

### t-gisserver-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### gisserver-01

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### gisserver-02

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### gisserver-03

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### gisserver-04

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### gisserver-05

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### gisserver-06

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## META

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy META.

### meta-01

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### meta-02

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### t-meta-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## PZGIK

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy PZGIK.

### pzgik-01

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### pzgik-02

Wirtualna maszyna wewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### t-pzgik-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## RDBMS

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy RDBMS.

### rdbms-01

Fizyczna maszyna środowiska przetwarzania.

### rdbms-02

Fizyczna maszyna środowiska przetwarzania.

### t-rdbms-01

Wirtualna maszyna środowiska testowego. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

## WEB

Pakiet instancji logicznych maszyn infrastruktury sprzętowej węzła instalacji klasy WEB.

### web-01

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### web-02

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### web-03

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### web-04

Wirtualna maszyna zewnętrznego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

### t - web-01

Wirtualna maszyna testowego środowiska przetwarzania. Szczegółowe parametry instancji (CPU, RAM, HDD) podano w zestawieniu poniżej.

# Środowisko przetwarzania i składowania

Dla szacowanego zapotrzebowania na zasoby sprzętowe dokonano grupowania komponentów aplikacyjnych zgodnie z ich przeznaczeniem oraz dostarczaną funkcjonalnością, umożliwi to osadzenie ich w konkretnej warstwie systemu.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Środowisko przetwarzania** | | | | | | |
| **Węzeł** | **Środowisko** | **Nazwa** | **typ** | **CPU (core)** | **RAM (GB)** | **HDD (GB)** |
| Oprogramowanie zarządzające systemem | wewnętrzne | adm-01 | virt | 4 | 16 | 100 |
| testowe | t-adm-01 | virt | 2 | 8 | 80 |
| System zarządzania PZGIK | wewnętrzne | pzgik-01 | virt | 4 | 8 | 100 |
| wewnętrzne | pzgik-02 | virt | 4 | 8 | 100 |
| testowe | t-pzgik-01 | virt | 4 | 8 | 100 |
| Portal informacyjny | zewnętrzne | cms-01 | virt | 4 | 8 | 100 |
| zewnętrzne | cms-02 | virt | 4 | 8 | 100 |
| testowe | t-cms-01 | virt | 2 | 4 | 80 |
| Desktop GIS | wewnętrzne | desktopGis-01 | virt | 12 | 24 | 100 |
| testowe | t-desktopGis-01 | virt | 4 | 8 | 80 |
| Serwer danych przestrzennych | wewnętrzne | gisserver-01 | virt | 4 | 32 | 100 |
| wewnętrzne | gisserver-02 | virt | 4 | 32 | 100 |
| wewnętrzne | gisserver-03 | virt | 4 | 32 | 100 |
| wewnętrzne | gisserver-04 | virt | 4 | 32 | 100 |
| zewnętrzne | gisserver-05 | virt | 4 | 32 | 100 |
| zewnętrzne | gisserver-06 | virt | 4 | 32 | 100 |
| testowe | t-gisserver-01 | virt | 4 | 32 | 100 |
| Szyna usług | wewnętrzne | esb-01 | virt | 4 | 8 | 100 |
| wewnętrzne | esb-02 | virt | 4 | 8 | 100 |
| wewnętrzne | esb-03 | virt | 4 | 8 | 100 |
| zewnętrzne | esb-04 | virt | 4 | 8 | 100 |
| zewnętrzne | esb-05 | virt | 4 | 8 | 100 |
| zewnętrzne | esb-06 | virt | 4 | 8 | 100 |
| testowe | t-esb-01 | virt | 2 | 4 | 80 |
| Przeglądarka danych przestrzennych | zewnętrzne | web-01 | virt | 4 | 8 | 100 |
| zewnętrzne | web-02 | virt | 4 | 8 | 100 |
| Rozbudowana przeglądarka danych przestrzennych | wewnętrzne | web-03 | virt | 4 | 8 | 100 |
| wewnętrzne | web-04 | virt | 4 | 8 | 100 |
| testowe | t-web-01 | virt | 2 | 4 | 80 |
| Serwer metadanych | zewnętrzne | meta-01 | virt | 4 | 12 | 100 |
| zewnętrzne | meta-02 | virt | 4 | 12 | 100 |
| testowe | t-meta-01 | virt | 4 | 8 | 80 |
| RDBMS | wewnętrzne | rdbms-01 | fizyczne |  |  |  |
| wewnętrzne | rdbms-02 | fizyczne |  |  |  |
| testowe | t-rdbms-01 | virt | 4 | 16 | 100 |
|  |  |  | **SUMA** | **132** | **460** | **3 180** |

Komponenty funkcjonalne znajdujące się w ramach grupy Aplikacja mobilna nie wymagają dodatkowych zasobów sprzętowych, gdyż wykorzystują zasoby sprzętu mobilnego. Nie zakłada się potrzeby oddzielnych zasobów dla komponentów w ramach grupy funkcjonalnej Mapowy interfejs integracyjny oraz Edytor metadanych, gdyż dostępne rozwiązania rynkowe posiadają te elementy zintegrowane w ramach rozwiązania a wymagane zasoby sprzętowe dla tych grup funkcjonalnych  pomijalne (zbilansowane w ramach systemu).

Ponadto nadmienić należy, iż każda z grup funkcjonalnych została zdublowana w celu zapewnienia wysokiej wydajności w trybie Active-Active.

W stosunku do silnika relacyjnej bazy danych zakłada się wykorzystanie maszyn fizycznych, zapewniając wysoką dostępność. Ponadto zakłada się, że RDBMS posiada funkcjonalność udostępnienia co najmniej dwóch niezależnych od siebie przestrzeni składowania i przetwarzania danych, co zapewni rozdzielność danych wewnętrznych od zewnętrznych i możliwość niezależnego konfigurowania komponentów aplikacyjnych w zależności od przeznaczenia ich dla użytkownika (zewnętrzny, wewnętrzny). Instancję testową dla RDBMS przewiduje się zrealizować w środowisku wirtualnym.

W zakresie zapotrzebowania na przestrzeń składowania danych zakłada się 10% roczny przyrost danych systemu co przekłada się na następujące wartości:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Środowisko składowania** | | | | | | |
| **Węzeł** | **Przestrzeń [TB]** | **1 rok** | **2 rok** | **3 rok** | **4 rok** | **5 rok** |
| RDBMS | 1 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 |
| Dane plikowe | 2 | 2,2 | 2,4 | 2,6 | 2,9 | 3,2 |
| Wirtualizacja | 3,3 | 3,6 | 4 | 4,4 | 4,8 | 5,3 |
| **SUMA** | **6,3** | **6,9** | **7,6** | **8,3** | **9,1** | **10** |

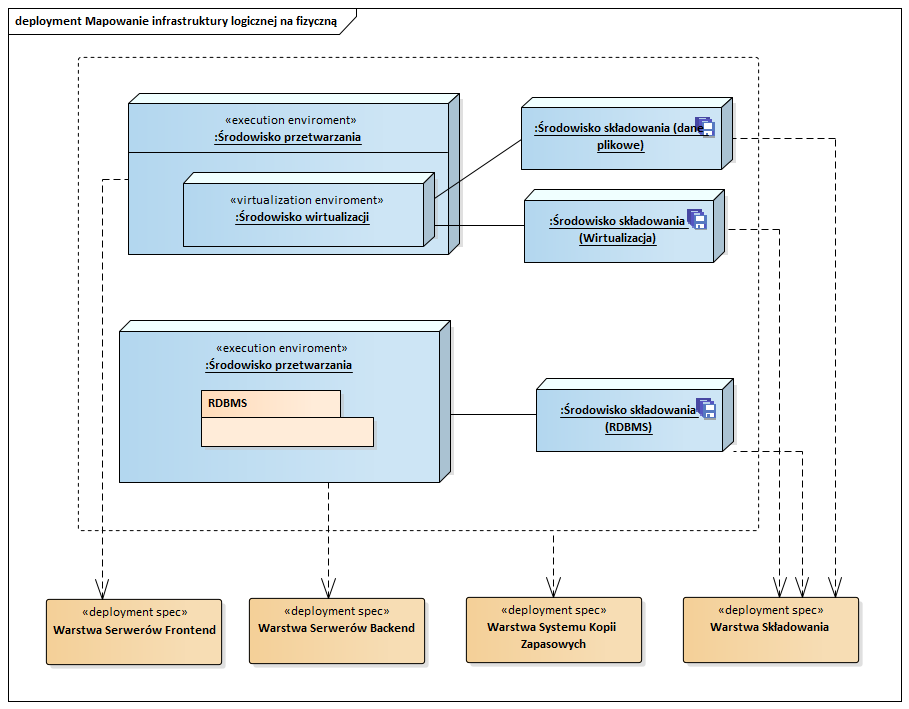
RDBMS – silnik relacyjnej bazy danych, przechowuje i przetwarza dane przestrzenne generowane dynamicznie w postaci usług OGC

Dane plikowe – przestrzeń plikowa służąca do składowania danych źródłowych, danych wygenerowanych na potrzeby usług np. WMTS (cache), pliki konfiguracyjne aplikacji w postaci współdzielonych zasobów dla danego węzła

Wirtualizacja – zasoby wykorzystywane przez wirtualne systemy operacyjne w ramach systemów operacyjnych

# 

# Mapowanie infrastruktury logicznej na fizyczną



Rysunek nr: 18**Mapowanie infrastruktury logicznej na fizyczną**

Architektura fizyczna SIPWW została opisana w załączniku nr 8 do SOPZ - „Wymagania szczegółowe w zakresie dostawy, instalacji, konfiguracji i uruchomienia infrastruktury teleinformatycznej SIPWW”.