

Oprogramowanie Nowe

Oprogramowanie Nowe
– Projekt techniczny

TAXUS•IT



Biuro Urządzania Lasu i Geodezji Leśnej

iBPY

OPROGRAMOWANIE NOWE

Oprogramowanie Nowe – Projekt techniczny

Praca wykonana na zlecenie Zakładu Informatyki Lasów Państwowych w ramach umowy nr DZ.271.31.2016 z dnia 25 lipca 2017 r.

TAXUS•IT



Biuro Urządzania Lasu i Geodezji Leśnej



Warszawa 2018

SPIS TREŚCI

1	Wstęp	5
2	Charakterystyka serwerów	6
2.1	Architektura fizyczna	7
2.2	Konfiguracja sprzętowa:	8
	Frontend Server	8
	Integration Server	8
	Backend Server	8
	Geoserver Server	9
	Taksator Backend DB	9
	IBM DataPower	9
	IBM DataStage	11
3	Opis narzędzi programistycznych.....	12
4	Opis Aplikacji i bibliotek	12
	Frontend Taksator i Zarządzanie	12
	Backend Taksator	13
5	Opis procesu kompilacji	14
	Frontend	14
	Backend	15
6	Procedura instalacji, uruchomienia i wdrożenia	15
6.1	Utworzenie baz danych	15
	Tworzenie bazy administracyjnej taksadm.....	15
	Tworzenie bazy jednostki	15
6.2	Instalacja i konfiguracja aplikacji Taksator frontend.....	16
6.3	Instalacja i konfiguracja aplikacji Taksator backend	17

Konfiguracja dostępu do bazy danych	17
Konfiguracja katalogów roboczych kolejek JMS	17
6.4 Instalacja i konfiguracja serwera mapowego GeoServer	18
Konfiguracja źródeł danych	18
6.5 Instalacja i konfiguracja usług ESB	18
Konfiguracja endpointów Taksator Backedn	19
Konfiguracja serwerów Taksator GeoServer	19
Konfiguracja opartego na Log4j loggera ESB	19
6.6 Instalacja i konfiguracja projektu IBM DataStage	19
Konfiguracja źródeł danych ODBC	19
7 Polityka kopii bezpieczeństwa	20

1 WSTĘP

Niniejszy dokument zawiera zaktualizowane elementy projektu techniczny Nowego Oprogramowania.

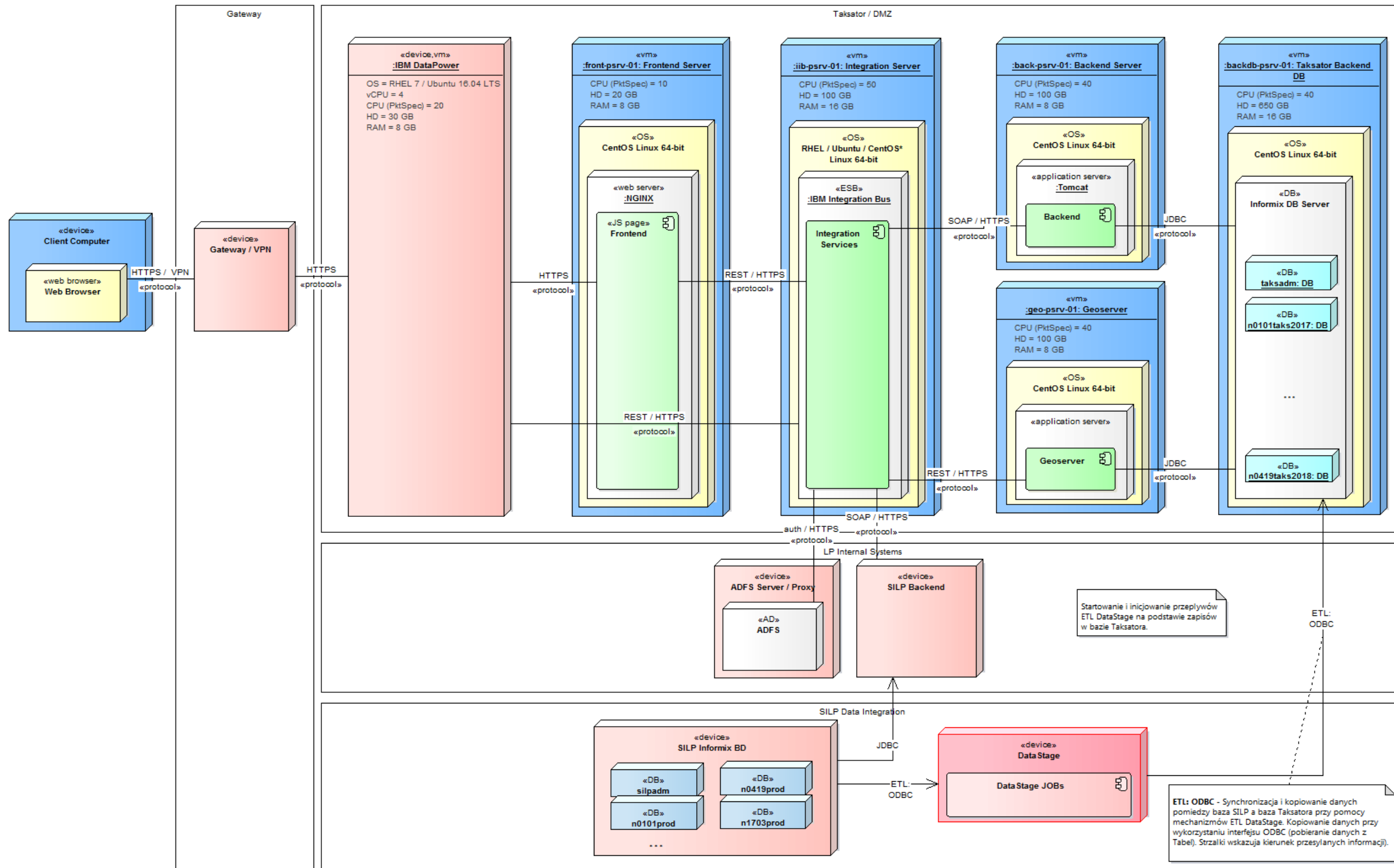
Integralną częścią dokumentacji jest projekt wykonany w oprogramowaniu Enterprise Architect zawierający diagramy z modelami BPMN, UML (i innymi) obrazującymi różne aspekty systemu. Diagramy wraz z zawierającymi je komponentami zostały opisane. Część modeli została przeniesiona do niniejszego dokumentu w odpowiednie rozdziały. Szczegółowe i techniczne elementy modelowania dostępne są tylko na modelu wykonanym w oprogramowaniu Enterprise Architect.

2 CHARAKTERYSTYKA SERWERÓW

Poniżej znajduje się schemat proponowanej architektury fizycznej dla wdrożenia oprogramowania TAKSATOR.

Schemat zgodny jest z założeniami przedstawionymi w projekcie technicznym oraz z dalszymi ustaleniami ze spotkań z przedstawicielami Zamawiającego.

2.1 Architektura fizyczna



2.2 Konfiguracja sprzętowa:

Frontend Server

Cpu (PktSpec) = 10

HD = 20GB

RAM = 8GB

OS: CentOS 7 Linux 64-bit

Wymagany software: NGINX

Integration Server

Cpu (PktSpec) = 40

HD = 100GB

RAM = 16GB

OS: RHEL (Red Hat Enterprise Linux) 6-7 /Ubuntu 16.04 LTS / CentOS* (Not on IBM's supported OS List) Linux 64-bit

Backend Server

Cpu (PktSpec) = 40

HD = 100GB

RAM = 8GB

OS: CentOS 7 Linux 64-bit

Wymagany software: Java 8, Apache Tomcat 7

Geoserver Server

Cpu (PktSpec) = 40

HD = 100GB

RAM = 8GB

OS: CentOS 7 Linux 64-bit

Wymagany software: Java 8, Apache Tomcat 7

Taksator Backend DB

Cpu (PktSpec) = 40

HD = 650GB

RAM = 16GB

OS: CentOS Linux 64-bit

IBM DataPower

IBM DataPower jako urządzenie (appliance) lub jako VM.

Wymagania minimalne (IBM) dla VM:

CPU (PktSpec) = 20 (4 vCPU)¹

HD = 30GB

RAM = 8GB

OS : RHEL 7 / Ubuntu 16.04 LTS

Hypervisors: VMware ESXi 5.5 / 6.0 / 6.5

Resource allocation for stand-alone VMware hypervisors.				
Configuration name	virtual processors (vCPU)	RAM (GB)	Disk space provisioning	
			Thin (Mb)	Thick (GB)
Small	4	4	258.6	32
Standard (default)	8	8	258.6	32
Enterprise	16	96	258.6	32

Rekomendacja IBM Dla DataPower Gateway:

Resource requirements for Linux hosts

To install the DataPower Gateway, the host must meet the following requirements:

- To install the Debian or RPM packages, the host must be running a supported 64-bit version of Linux.
- 2 GiB of free storage must be available on /opt.
- 5 GiB of free storage must be available on /var.
- At least 2 free loop devices are required, with an additional 1 loop device if RAID storage is used.
- RAID storage, if used, must be configured in the datapower.conf file.

The absolute minimum resource allocation depends on expected workload and edition. After deployment, you can increase resource settings to meet workload requirements.

- For DataPower Gateway without API workload, the minimum resource allocation is 4 virtual processors (vCPU) and 4 GB RAM except for the developers edition. For the developers edition, the minimum resource allocation is 2 vCPU and 4 GB RAM.
- For DataPower Gateway with API workload, the minimum resource allocation is 4 vCPU and 8 GB RAM independent of edition.

https://www.ibm.com/software/reports/compatibility/clarity-reports/report/html/softwareReqsForProduct?deliverableId=6FE0A500D1C311E6A98AAE81A233E762&duComponentIds=Server_70062E60D1C311E6A98AAE81A233E762#

IBM DataStage

Licencja: DataStage na system operacyjny AIX - 1 CPU Core, zarówno dla środowiska testowego, jak i środowisko produkcyjnego.

System produkcyjny:

- 1 CPU Core
- kompilator XL C/C++ 13.1.2 dla AIX 7.1 lub XL C/C++ dla AIX 7.2 - kompilator jest licencjonowany dla systemu operacyjnego AIX - klient musi mieć zakupioną licencję
- 64GB MB RAM
- Do instalacji DataStage'a potrzeba 20GB przestrzeni dyskowej.
- Dodatkowo przestrzeń dyskowa na dane potrzebne do przetwarzania przez DataStage'a np. 500GB

3 OPIS NARZĘDZI PROGRAMISTYCZNYCH

Narzędzia wspomagające proces wytwarzania oprogramowania:

- IDE (Integrated Development Environment) – IntelliJ IDEA Ultimate Edition w wersji 2018
 - Zintegrowane środowisko programistyczne do tworzenia, edycji i testowania oprogramowania
- Enterprise Architect w wersji 13.5
 - Oprogramowanie służące do tworzenia dokumentacji, rozwijania bazy danych, definiowania komunikatów SOAP wraz ze schematami.
- Gradle w wersji 4.5.1
 - Narzędzie służące do rozwiązywania zależności między projektami i bibliotekami.
 - Automatycznie kompiluje i builduje aplikację
- Sencha Cmd w wersji 6.5.2.15
 - Narzędzie wiersza poleceń.
 - Automatycznie kompiluje źródła aplikacji wraz z modułami.
 - Kompiluje aplikację do formatu WAR (Web Application Resource)
- Sencha ExtJS w wersji 6.5.2.463
 - Framework napisany w języku JavaScript, na którym opiera się frontend systemu.
- Apache Tomcat w wersji 8

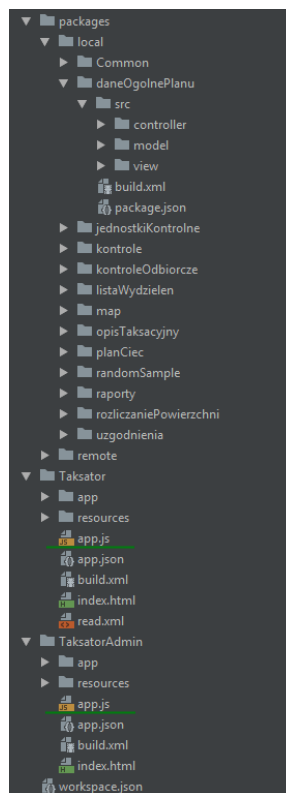
Kontener aplikacji webowych. W środowisku programistycznym wykorzystywany do uruchamiania systemu.

4 OPIS APLIKACJI I BIBLIOTEK

System można podzielić na dwie niezależnie rozwijane warstwy oprogramowania: frontend i backend.

Frontend Taksator i Zarządzanie

Aplikacje frontend Taksator i Zarządzanie zbudowane są w oparciu o framework ExtJS. Obie aplikacje należą do jednego obszaru roboczego, zaś ich struktura jest modułowa. Moduły aplikacji znajdują się w katalogu packages. Aplikacje wykorzystują poszczególne moduły, a konfiguracja ta jest opisana w pliku app.js w poszczególnych aplikacjach.



Frontend systemu jest napisany z przewagą wykorzystania wzorca MVC-VM (Model-View-Controller-ViewModel) w poszczególnych widokach ze względu na cykl życia obiektów w technologii ExtJS.

Statyczne pliki tj. style, obrazki, dane konfiguracje są przechowywane w katalogu resources.

Do obsługi mapy została wykorzystana biblioteka OpenSource:

- OpenLayers 3

Backend Taksator

Aplikacja ta jest napisana w technologii JEE oraz frameworku Spring Boot. Do utworzenia aplikacji wykorzystane zostały następujące biblioteki wraz z ich zależnościami:

- Spring Boot 1.5.9
- WSDL4j 1.6.1
- GlassFish JAXB 1.6.1
- IBM Informix JDBC Driver 4.10.9
- WSS4J 1.6.19
- Ehcache 2.6.9
- Spring Boot MyBatis 1.3.1
- Google Guava 20.0
- JTS 1.13
- Apache Commons Lang 3.4

- ActiveMQ 5.14.5
- Spring JMS 5.0.5
- BIRT 4.2.2

Za pobranie, konfigurację i kompilację bibliotek w aplikacji odpowiedzialny jest Gradle. Korzysta on z repozytorium „Maven Repository” do pobrania bibliotek oraz ich zależności.

Konfiguracja bibliotek oraz struktury aplikacji dla Gradle znajduje się w pliku [build.gradle](#). Zawiera on również pluginy konfigurujące aplikację w workspace popularnych IDE (IntelliJ Idea, Eclipse).

Moduły aplikacji są konfigurowane również przez Gradle, a ich konfiguracja znajduje się w pliku [settings.gradle](#).

Usługi zaimplementowane przez backend opisane są w standardzie WSDL (Web Services Description Language). Na podstawie tych plików w aplikacji generowane są komunikaty wejściowe i wyjściowe dla wszystkich usług SOAP (Simple Object Access Protocol) w postaci POJO (Plain Old Java Object), które są zaimplementowane w usługach. Generowanie POJO na podstawie WSDL jest elementem kompilacji aplikacji. Zdefiniowane jest zadanie w [build.gradle](#) o nazwie „generateFromXSD”, które realizuje tę czynność.

5 OPIS PROCESU KOMPILACJI

Frontend

Kompilacja aplikacji Taksator oraz Zarządzanie w całości odbywa się przy pomocy konsolowego narzędzia SenchaCmd. Narzędzie to pozwala na kompilację aplikacji znajdujących się w workspace.

Przed pierwszym uruchomienie aplikacji należy doinstalować bibliotekę SDK frameworka ExtJS dla każdej z aplikacji.

- ```
➤ cd c:/ścieżka/do/workspace/aplikacji/
➤ cd Taksator
➤ sencha app upgrade C:/ścieżka/do/katalogu/z/sdk/extjs
```

Tak przygotowane środowisko może posłużyć do kompilacji aplikacji.

- ```
➤ sencha app build production
```

Po wykonaniu tego polecenia skompilowana aplikacja znajduje się w workspace w katalogu [build/production/Taksator/](#).

Backend

Kompilacji źródeł aplikacji Taksator Backend przeprowadza się z przy pomocy konsolowego narzędzia Gradle. Z poziomu źródeł aplikacji należy wywołać komendę „gradle build”. Aplikacja i jej konfiguracja zostały przygotowane tak, aby wymagane biblioteki zostały pobrane automatycznie.



```
> gradle build
```

Wywołanie tej komendy skompiluje odpowiednio źródła aplikacji i utworzy plik [taksator_backend.war](#) w formacie WAR (Web Archive Resource) w katalogu [./build/libs/](#).

6 PROCEDURA INSTALACJI, URUCHOMIENIA I WDROŻENIA

6.1 Utworzenie baz danych

W systemie Taksator możemy wyróżnić 2 rodzaje baz danych: baza administracyjna o nazwie taksadm oraz bazy jednostek o nazwach nAABBrRRR (gdzie AA to numer rdLP, BB numer nadleśnictwa a RRRR rok taksacji). Baza taksadm pełni podobną rolę co baza silpadm w systemie SILP. Bazy jednostek przechowują dane urządzeniowe i ewidencyjne nadleśnictw biorących aktualnie udział w procesie Urządzania Lasu.

Tworzenie bazy administracyjnej taksadm

Na serwerze bazodanowym działającym na maszynie opisanej jako **Taksator Backend DB** należy utworzyć pustą bazę danych o nazwie **taksadm** i parametrach: logging BUFFERED LOG, dblocale pl_pl.8859. Następnie należy na bazie uruchomić załączone w katalogu [/KodyZrodlowe/BazaDanych/BazaAdministracyjna/](#) skrypty:

- *TworzenieBazyDanych.sql*
- *ModyfikacjaBazyDanych.sql*

Tworzenie bazy jednostki

Baza danych jednostki posiada 3 schematy bazodanowe (BS, BSA i BU) zawierające tabele o takich samych nazwach. Utworzenie takiej struktury danych w bazie IBM Informix możliwe jest tylko wtedy, gdy baza pracuje w trybie zgodności z ANSI/ISO.

Utworzenie szablonu bazy danych z modulem IBM Informix Spatail DataBalde

Na serwerze bazodanowym działającym na maszynie opisanej jako **Taksator Backend DB** należy utworzyć pustą bazę danych o nazwie n0000r0000 o następujących parametrach: logging BUFFERED LOG, dblocale pl_pl.8859. Następnie, za pomocą narzędzia **blademgr** należy zarejestrować w bazie właściwy dla danej instancji IBM Informix moduł IBM Informix Spatail DataBalde. Szablon bazy w postaci katalogu n0000r0000.exp tworzymy za pomocą polecenia **dbexport**.

Utworzenie bazy jednostki

Aby utworzyć bazę danych jednostki działającą w trybie zgodności z ANSI/ISO należy użyć narzędzia **dbimport** wywołanego z opcjami **-l -ansi -ci** i szablonem n0000r0000.exp. Następnie na utworzonej bazie należy uruchomić we właściwej kolejności załączone w katalogu /KodyZrodlowe/BazaDanych/BazaJednostki/ skrypty:

- TworzenieBazyDanychBS.sql,
- TworzenieBazyDanychBSA.sql,
- TworzenieBazyDanychBU.sql,
- ModyfikacjaBazyDanych.sql,
- ModyfikacjaBazyDanychBS.sql,
- ModyfikacjaBazyDanychBSA.sql,
- ModyfikacjaBazyDanychBU.sql.

6.2 Instalacja i konfiguracja aplikacji Taksator frontend

Na przygotowany serwer http **nginx** działający na maszynie opisanej jako **Frontend Server** należy zainstalować (poprzez skopiowanie do podkatalogu html serwera http) aplikacje Taksator i TaksatorZarzadzanie załączone w katalogu /KodyZrodlowe/Frontend/.

Następnie należy skonfigurować przekierowanie żądań obu aplikacji frontend na szynę usług ESB poprzez modyfikację i skopiowanie do właściwego katalogu konfiguracyjnego serwera nginx (zazwyczaj /etc/nginx/conf.d/) załączonych w katalogu /PlikiKonfiguracyjne/Frontend/ plików Taksator.conf i TaksatorAdmin.conf. Wymagane modyfikacje w plikach:

<i>upstream backend.server</i>	ip lub nazwa serwera Integration Server oraz port usługi ESB
<i>server.ssl_certificate</i>	ścieżka do pliku zawierającego certyfikat serwera
<i>server.ssl_certificate_key</i>	ścieżka do pliku zawierającego klucz prywatny

6.3 Instalacja i konfiguracja aplikacji Taksator backend

Na przygotowany serwer aplikacyjny **Tomcat** działający na maszynie opisanej jako **Backend Server** należy zainstalować aplikację `taksator_backend.war` załączoną w katalogu `/WersjeInstalacyjne/Backend/` lub powstałą w wyniku kompilacji kodów źródłowych załączonych w katalogu `/KodyZrodlowe/Backend/`.

Konfiguracja dostępu do bazy danych

W pliku konfiguracyjnym aplikacji: `taksator_backend/WEB-INF/classes/application.properties` należy zmodyfikować parametry:

<code>taksator.datasource.url</code>	Maska connectionstring serwera bazodanowego Taksator Backend DB np. jdbc:informix-sqli: //%s:%s/%s:INFORMIXSERVER=ol_informix12; DB_LOCALE=pl_pl.8859-2; CLIENT_LOCALE=pl_pl.8859-2
<code>taksator.datasource.username</code>	Nazwa użytkownika systemowego Taksator, którym aplikacja backend łączy się z bazami danych
<code>taksator.datasource.password</code>	Hasło użytkownika systemowego Taksator, którym aplikacja backend łączy się z bazami danych
<code>taksator.datasource.host</code>	Nazwa lub ip hosta serwera bazodanowego Taksator Backend DB
<code>taksator.datasource.port</code>	Numer portu instancji serwera bazodanowego IBM Informix

Konfiguracja katalogów roboczych kolejek JMS

W pliku konfiguracyjnym aplikacji: `taksator_backend/WEB-INF/classes/application.properties` należy zmodyfikować parametry:

<code>taksator.jms.temp_path</code>	Ścieżka do katalogu tymczasowego kolejek JMS
<code>taksator.jms.path</code>	Ścieżka do katalogu roboczego kolejek JMS

6.4 Instalacja i konfiguracja serwera mapowego GeoServer

Na przygotowany serwer aplikacyjny Tomcat działający na maszynie opisanej jako **Geoserver Server** należy zainstalować aplikację taksator_geoserver.war załączoną w katalogu */WersjeInstalacyjne/GeoServer/*.

Konfiguracja źródeł danych

Za pomocą aplikacji GeoServerConfig.jar załączonej w katalogu */KodyZrodlowe/GeoServer/* i przygotowanego pliku konfiguracyjnego z listą źródeł danych należy wygenerować katalog konfiguracyjny aplikacji GeoServer. Utworzony katalog należy zintegrować z aplikacją GeoServer poprzez skopiowanie zawartości do katalogu *data* zainstalowanej aplikacji.

Przykładowa konfiguracja źródeł danych aplikacji GeoServer:

```
<?xml version="1.0" encoding="UTF-8"?>
<datastores>
  <datastore id = "n0419r2017">
    <user>informix</user>
    <passwd>q3u4r5U9za</passwd>
    <port>35000</port>
    <host>52.26.141.244</host>
    <server>ifx12h</server>
    <database>n0419r2017</database>
  </datastore>
  <datastore id = "n0419r2018">
    <user>informix</user>
    <passwd>q3u4r5U9za</passwd>
    <port>35000</port>
    <host>52.26.141.244</host>
    <server>ifx12h</server>
    <database>n0419r2018</database>
  </datastore>
</datastores>
```

6.5 Instalacja i konfiguracja usług ESB

Za pomocą narzędzia IBM Integration Toolkit kompilujemy do postaci archiwów BAR załączone w */KodyZrodlowe/ESB/* projekty IBM Integration Bus. Następnie za pomocą tego samego narzędzia instalujemy wygenerowane binaria na przygotowanej instancji IBM Integration Bus działającej na maszynie opisanej jako **Integration Server**.

Konfiguracja endpointów Taksator Backend

W katalogu `/var/mqsi/properties/` serwera Integration Server należy umieścić plik konfiguracyjny o nazwie `endpoints.properties` określający dostęp do usług działających po stronie aplikacji Taksator Backend.

Przykładowa konfiguracja endpointów Taksator Backend:

```
backendPowierzchnieProbne=http://taksator.lasy.gov.pl/taksator_backend/taksator/ws/PowierzchnieProbne
backendAdresLesny=http://taksator.lasy.gov.pl/taksator_backend/taksator/ws/AdresLesny
backendLogowanie=http://taksator.lasy.gov.pl/taksator_backend/taksator/ws/Logowanie
backendObliczenia=http://taksator.lasy.gov.pl/taksator_backend/taksator/ws/Obliczenia
backendSlovniki=http://taksator.lasy.gov.pl/taksator_backend/taksator/ws/Slovniki
backendZarzadzanie=http://taksator.lasy.gov.pl/taksator_backend/taksator/ws/Zarzadzanie
backendOpisTaksacyjny=http://taksator.lasy.gov.pl/taksator_backend/taksator/ws/OpisTaksacyjny
```

Konfiguracja serwerów Taksator GeoServer

W katalogu `/var/mqsi/properties/` serwera Integration Server należy umieścić plik konfiguracyjny o nazwie `geoservers.properties` określający dostęp do serwerów mapowych GeoServer obsługujących dane geometryczne z baz danych nadleśnictw podzielonych według numerów rdLP.

Przykładowa konfiguracja serwerów Taksator GeoServer:

```
default=http://54.37.143.244/geoserver/ows/wms
region01=http://54.37.143.244/geoserver1/ows/wms
region02=http://54.37.143.244/geoserver2/ows/wms
```

Konfiguracja opartego na Log4j loggera ESB

Szczegółowy opis konfiguracji autorskiego loggera komunikatów ESB znajduje się w odrębnej instrukcji załączonej w katalogu `/KodyZrodlowe/ESB/Logowanie log4j/Instrukcja/`.

6.6 Instalacja i konfiguracja projektu IBM DataStage

Na przygotowanej instancji IBM InfoSphere Information Server działającej na maszynie opisanej jako **IBM DataStage** należy utworzyć projekt o nazwie Taksator. Do tego celu należy użyć narzędzia IBM InfoSphere DataStage and QualityStage Administrator client. Następnie, do utworzonego projektu, za pomocą narzędzia IBM InfoSphere Information Server Manager należy zaimportować zawartość załączonego w katalogu `/KodyZrodlowe/DataStage/` pliku `Taksator.isx`.

Konfiguracja źródeł danych ODBC

W pliku konfiguracyjnym IBM Information Server należy zdefiniować źródło danych do bazy administracyjnej o nazwie `taksadm` oraz źródła danych do baz urzędzeniowych o nazwach

zgodnych z nazwami baz danych. Źródła danych definiowane są w pliku *InformationServer/Server/DSEngine/.odbc.ini*.

Przykładowa konfiguracja źródła danych bazy taksadm:

```
[taksadm]
Driver=/opt/IBM/InformationServer/Server/branded_odbc/lib/VMifcl00.so
Description=DataDirect Informix Wire Protocol driver
Database= taksadm
LogonID=
Password=
HostName=54.36.142.243
PortNumber=35000
ServerName=ifx12h
CancelDetectInterval=0
TrimBlankFromIndexName=1
ApplicationUsingThreads=1
QEWSO=2458177
```

Kolejnym krokiem konfiguracji jest modyfikacja listy źródeł danych dostępnych w założonym projekcie Taksator. Lista źródeł danych obsługiwanych przez projekt definiowana jest w pliku *InformationServer/Server/Projects/Taksator/uvodbc.config*.

Przykładowa konfiguracja źródeł danych projektu Taksator:

```
[ODBC DATA SOURCES]
<taksadm>
DBMSTYPE = ODBC
<n0419r2017>
DBMSTYPE = ODBC
```

7 POLITYKA KOPII BEZPIECZEŃSTWA

Backup i odtwarzanie systemu wykonywane będzie zgodnie z przyjętą polityką w ZILP. Niezależnie backup'owane są bazy danych Informix oraz serwery wirtualne systemu.