



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Projekt Zarządzania Pozycjami Pojazdów

Załącznik nr 1.1 do SIWZ Opis Przedmiotu Zamówienia

Architektura i opis modułów infrastruktury rozwiązania

Przygotował:
Nr wydania:
Wersja
Data wydania:

ZTM
1.0
OPZ
22 sierpnia 2016



Spis treści

1	Założenia ogólne	4
1.1	Moduły funkcjonalne aplikacji.....	4
1.2	Moduły infrastruktury rozwiązania	4
1.3	Uwagi	5
1.4	Komentarz do reguł oczekiwanych protokołów API:	5
2	Schemat poglądowy rozwiązania	6
3	Wyznaczniki efektywnej infrastruktury	6
3.1	Stanowiska użytkowników	6
3.2	Administrowanie dostępem	9
3.3	Architektura infrastruktury	9
3.3.1	Oprogramowanie i infrastruktura klient – serwer	9
3.3.2	Wymierzenie i zaoferowanie infrastruktury.....	9
3.3.3	Pojemność bazy danych.....	10
3.3.4	Cechy i skalowalność pamięci dyskowej	10
4	Architektura oprogramowania	10
4.1	Czym są moduły, wymaganie modularyzacji	10
4.2	Interfejsy pomiędzy modułami i do zewnętrznych rezydencji danych.....	11
4.3	Logi pracy Modułów	11
5	Wymagania wobec technologii i sposobu wykonania.....	11
5.1	Wymagania jakościowe	11
5.2	Technologia oprogramowania.....	12
5.3	Rezydencje danych	12
5.4	Środowisko rozwoju i repozytoria	13
5.5	Rozwój oprogramowania	13
5.6	Dokumentacja.....	14
6	Opis modułów	14
6.1	Funkcje modułu odczytu danych o położeniu pojazdów.....	14
6.1.1	Oczekiwane podstawowe cechy jakościowe	14
6.1.2	Schemat sekwencji podejmowanej przez wątek czytający	15
6.1.3	Opis funkcjonalności sterownika	15
6.2	Interfejs do systemu Rozkładów Jazdy	17
6.2.1	Oczekiwane cechy jakościowe	17
6.2.2	Schemat modułu sterownika	18
6.2.3	Schemat sekwencji działania modułu logicznego.....	21
6.3	Moduł interfejsu do Rejestru pojazdów	23



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

6.3.1	Oczekiwane cechy jakościowe	23
6.3.2	Schemat sekwencji działania sterownika	24
6.3.3	Opis funkcjonalności sterownika i interfejsu	24
6.4	Moduł API dostępu do danych zgromadzonych w bazie danych	26
6.4.1	Oczekiwane cechy jakościowe	26
6.4.2	Opis algorytmu	26
6.5	Moduł obliczeń, gromadzenia i udostępniania danych	26
6.5.1	Oczekiwane cechy jakościowe	26
6.5.2	Opis funkcjonalności	27
6.6	Moduł generatora i przeglądania raportów	27
6.6.1	Przykładowe raporty, minimalny zestaw	28
6.6.2	Oczekiwane cechy jakościowe	28



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

1 Założenia ogólne

Przedmiotem zamówienia jest zaprojektowanie oraz wdrożenie zintegrowanego systemu Zarządzania Pozycjami Pojazdów (ZPP), który ma umożliwić sprawowanie kontroli nad funkcjonowaniem pojazdów komunikacji miejskiej nadzorowanych bezpośrednio lub pośrednio przez Zarząd Transportu Miejskiego.

System oparty będzie na danych wejściowych pochodzących z pojazdów, do których Zamawiający posiada dostęp obecnie lub będzie posiadał w przyszłości. Dodatkowo system musi umożliwiać gromadzenie oraz udostępnianie danych uzyskiwanych z pojazdów oraz ich analizę.

Wykonawca musi uwzględnić wykorzystywane obecnie przez Zamawiającego technologie wymiany danych.

1.1 Moduły funkcjonalne aplikacji

Moduły funkcjonalne aplikacji są opisane w dokumencie „Opis modułów funkcjonalnych.docx”.

Na zestaw tych modułów składają się moduły wizualizacji danych i zarządzania ruchem:

- Moduł mapy
- Moduł schematu liniowego
- Moduł CR (centrali ruchu)
- Moduł przeglądarki rozkładów jazdy
- Moduł kierowcy
- Moduł ekspedycji
- Moduł Biuro
- Moduł klimatyzacji

1.2 Moduły infrastruktury rozwiązania

Te moduły są opisane w niniejszym materiale.

Na zestaw tych modułów składają się:

- Moduł odczytu danych o położeniu pojazdów
- Moduły interfejsu do systemu Rozkładów Jazdy
- Moduł interfejsu do rejestru pojazdów
- Moduł obliczeń, gromadzenia i udostępniania danych
- Moduł API dostępu do danych zgromadzonych w bazie danych
- Moduł generatora i przeglądarki raportów



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

1.3 Uwagi

Zamawiający oczekuje opracowania i udokumentowania architektury rozwiązania w pierwszych fazach prac wykonawczych. Architektura będzie przedmiotem odbioru i akceptacji przed rozpoczęciem prac programistycznych (por. dokumenty specyfikujące tryb i fazy wykonania).

Wykonawca MOŻE ZAPROPONOWAĆ w architekturze rozwiązania inne podejście niż przedstawione w niniejszym materiale, ale spełniające określone w niniejszym materiale oczekiwania rzeczowe i jakościowe. Opisane tu struktury i funkcje nie stanowią architektury rozwiązania lecz są wyśłowieniem konkretnych oczekiwań wobec produktu końcowego.

1.4 Komentarz do reguł oczekiwanych protokołów API:

Poniższe reguły spełniają oczekiwania architektury SOA, by interfejs do jej usług (oparty o koncepcję Application Programming Interface) umożliwił niezależny rozwój w czasie: aplikacji wystawiającej usługę w architekturze SOA i aplikacji korzystającej z niej w długim okresie.

Poniżej znajduje się przykładowa hierarchia wywołań protokołu API, która UCZY aplikację korzystającą z usługi JAK dialogować z aplikacją wystawiającą usługę.

Sens takiego protokołu API jest następujący:

Po pierwsze: Aplikacje są rozwijane niezależnie od siebie, więc zmiana aplikacji wystawiającej usługę nie może uniemożliwić działania aplikacji korzystającej z niej.

Aplikacja wystawiająca usługę instruuje aplikację korzystającą z usługi o sposobie dostępu do poszczególnych funkcji i formacie odpowiedzi; zakłada bowiem, że aplikacja korzystająca z usługi (a więc z oprogramowanym dialogiem odczytującym formaty zapytań i odpowiedzi w mechanizmie API), zastosuje „hierarchię” dialogu (por. przykład poniżej) i pozna aktualne (a więc być może zmienione wobec wcześniejszych) struktury dostępu do poszczególnych funkcji.

Po drugie: Aplikacja wystawiająca usługę winna proponować sposób na zasygnalizowanie aplikacji korzystającej z niej o potrzebie ponownego odczytania struktur dostępu do danych funkcji. To zapewnia możliwość wprowadzenia zmian „w locie”, bez przerywania wzajemnej komunikacji.

Przykład logiki takiej hierarchii:

- 1) Identyfikacja użytkownika i praw dostępu
→
- 2) Zapytanie (stałym kodem) o funkcje dostępne dla zlecającego (z poziomu programu; szczególną funkcją są pytania dotyczące słownika)
→

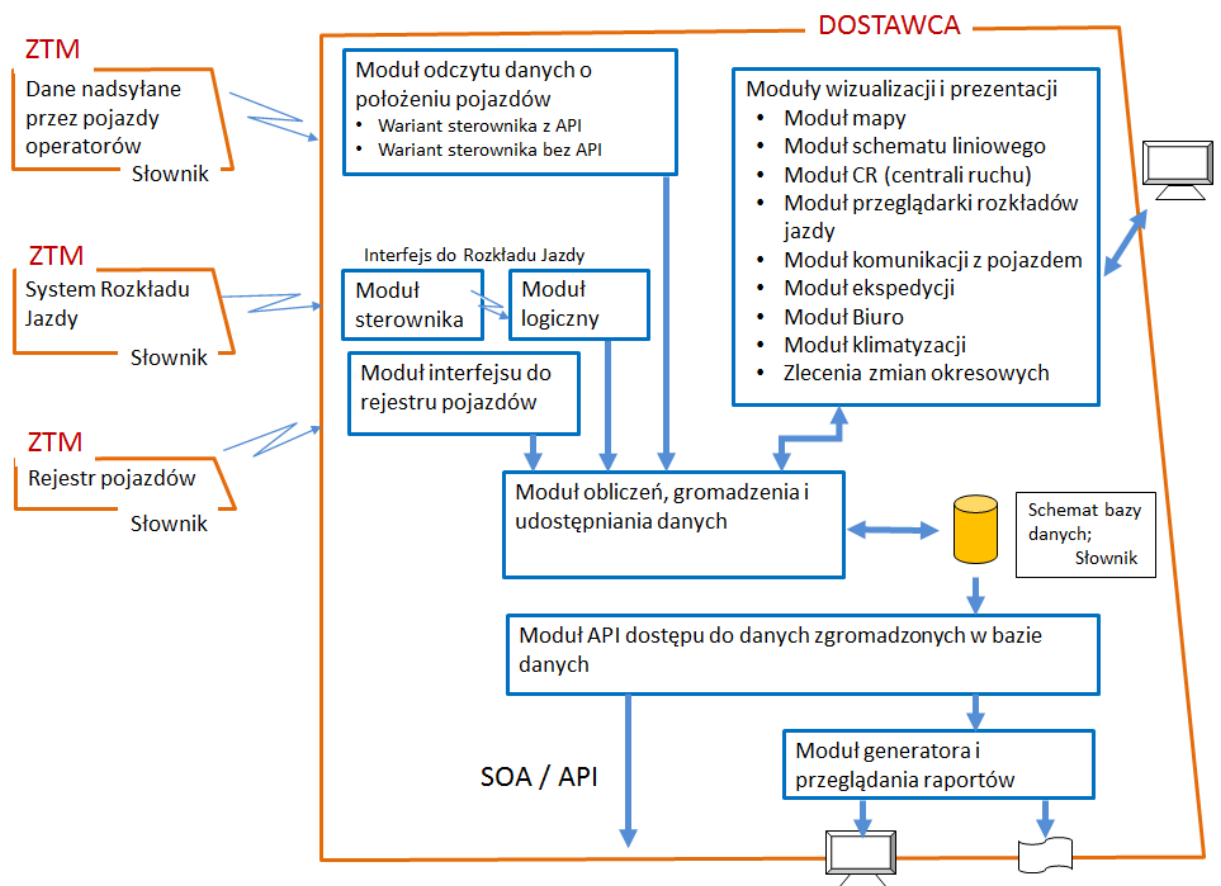


ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

- 3) Dla każdej funkcji: format zapytania i zestaw parametrów ze SŁOWNIKA
→
- 4) Dla każdej funkcji: format odpowiedzi i zestaw informacji ze SŁOWNIKA
→
- 5) Zapytanie przez zlecającego wg formatu z 3), by otrzymać odpowiedź wg formatu z 4)

2 Schemat poglądowy rozwiązania



3 Wyznaczniki efektywnej infrastruktury

3.1 Stanowiska użytkowników

Zamawiający przewiduje listę około 50 użytkowników o różnej potrzebie korzystania z aplikacji ze względu na obowiązki zawodowe i uprawnienia.

W poniższej tabeli znajdują się liczby i charakterystyki stanowisk, z których potrzebny będzie dostęp do aplikacji.



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Lp.	Liczba stanowisk	Funkcja użytkownika	Technologia połączenia	Uprawnienia	Komentarz
1.	5	Dyżurny ruchu ZTM. Osoby dedykowane jako kierujące ruchem , zawsze uprawnione do dostępu.	Sieć lokalna	Pełny dostęp do wszystkich funkcji, w zakresie czytania i edycji	To podstawowe stanowiska operacyjne u Zamawiającego, dla których przede wszystkim przygotowywana jest aplikacja. Tu czas dostępu i responsywność przy interakcji z systemem musi być maksymalnie efektywny.
2.	5	Dyżurni w punktach ekspedycji Osoby dedykowane jako kierujące ruchem , zawsze uprawnione do dostępu.	Sieć rozległa	Pełny dostęp do funkcji modułu ekspedycji (czytanie i edycja) Tylko czytanie (ogląd) do pozostałych funkcji, modułów.	To ważne stanowiska operacyjne u Zamawiającego, dla których dostęp do wizualizacji ruchu i funkcji 'modułu ekspedycji' warunkuje efektywne zarządzanie w punktach węzłowych.
3.	2	Stanowiska w dziale Organizacji Przewozów	Sieć lokalna	Pełny dostęp do funkcji modułu Centrali Ruchu (CR) w zakresie zleceń zmian okresowych (czytanie i edycja) Tylko czytanie (ogląd) do pozostałych funkcji, modułów.	To ważne stanowiska operacyjne u Zamawiającego decydujące o zleceniach nie ujętych w rozkładzie jazdy.
4.	10	Inni użytkownicy. Osoby z listy,	Sieć lokalna	Tylko czytanie (ogląd) do wszystkich funkcji, modułów.	To „obserwatorzy” nie biorący udziału w aktywnym zarządzaniu



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

		nie kierujące ruchem , ale też dedykowane jako zawsze uprawnione do dostępu			ruchem. Ci użytkownicy nie muszą być obsłużeni pierwszoplanowo, ale obserwacja ruchu przez nich musi być efektywna równocześnie ze stanowiskami operacyjnymi (por. poprzednie wiersze)
5.	30	Inni użytkownicy Osoby z listy, nie kierujące ruchem , ale dedykowane jako zawsze uprawnione do dostępu	Sieć lokalna, sieć rozległa ! Dostęp przez przeglądarkę Internetową	Tylko czytanie (ogląd) do wszystkich funkcji, modułów.	To „obserwatorzy” nie biorący udziału w aktywnym zarządzaniu ruchem. Ci użytkownicy nie muszą otrzymać dostępu do aplikacji jeżeli wyczerpany jest limit aktualnie pracujących użytkowników (tj. obecnie 20 osób, z wierszy wcześniejszych).
6.	5	Administratorzy	Sieć lokalna	Pełny dostęp i TYLKO do zadań administracyjnych oraz funkcji związanych z modułami infrastruktury (opisanymi w niniejszym dokumencie).	To osoby nie związane z funkcjami kierowania ruchem: ✓ Administratorzy ✓ Opiekujący się dostępem do źródeł ✓ Wykorzystujący API ✓ Korzystający z generatora raportów



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

3.2 Administrowanie dostępem

Wszyscy użytkownicy muszą być rejestrowani jako uprawnieni do dostępu.
Aplikacja nie jest wolno dostępna.

Użytkownicy z wierszy 1., 2., 3., 4. i 6. (w ramach swoich uprawnień) muszą zawsze otrzymać dostęp, przy czym nie mogą zostać nigdy wylogowani (chyba, że z własnej inicjatywy).

Użytkownicy z wiersza 5. powinni być wylogowani po interwale określonym parametrami, po zapytaniu, czy nadal są obserwatorami obrazu (jeżeli po założonym krótkim czasie nie nadejdzie odpowiedź na takie pytanie, użytkownik zostaje wylogowany [w ten sposób umożliwia dostęp innej osobie])

Podczas opracowywania architektury rozwiązania, Wykonawca przedstawi propozycję zasad i funkcji administrowania dostępem spełniającą powyższe oczekiwania.

3.3 Architektura infrastruktury

3.3.1 Oprogramowanie i infrastruktura klient – serwer

Niezbędna jest architektura klient – serwer. Ona implikuje podział zakresów przetwarzania (a więc i moce) pomiędzy stacje klienckie i serwer.

Architektura rozwiązania będzie przedmiotem opracowania, uzgodnienia i odbioru w pierwszej fazie realizacji systemu.

UWAGA: Jedynie w przypadku użytkowników opisanych w wierszu 4. Oczekiwany jest dostęp przez przeglądarkę. W innych sytuacjach rozwiązanie klient- serwer winno być adekwatne do oczekiwanej efektywności pracy.

Zakresy funkcjonalne i relacje pomiędzy oprogramowaniem 'klienta' i oprogramowaniem 'serwera' winny być adekwatne do opisanej funkcjonalności użytkownej i omówionej wcześniej tabeli dostępu przez różnego typu użytkowników.

3.3.2 Wymierzenie i zaoferowanie infrastruktury

Zamawiający prosi o wymierzenie charakterystyk mocy i zestawów sprzętowych stacji klienckich i serwera, adekwatnie do architektury rozwiązania i opisywanych warunków brzegowych.

1. Moc serwera i stacji użytkownika końcowego musi jednocześnie obsłużyć użytkowników wg wyżej opisanych charakterystyk.
2. Moc i przepustowość układów graficznych musi być wystarczająca do jednoczesnego obsłużenia użytkowników wg wyżej opisanych charakterystyk.



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Prosimy o ofertę na dostawę serwera dla aplikacji (sprzęt i oprogramowanie bazowe).

Prosimy o określenie minimalnych charakterystyk (sprzęt i oprogramowanie bazowe) dla stacji klienckich. Zamawiający ALBO dostarczy je we własnym zakresie, ALBO zamówi u Wykonawcy aplikacji.

3.3.3 Pojemność bazy danych

Oszacowanie jest następujące:

Obecnie dziennie nazytywanych jest ok. 6 GB danych. To dane o położeniach pojazdów (każdy z nich wysyła komunikat co 10 sekund).

Docelowo przewidujemy dzienny przyrost wielkości 20 GB (pozycje pojazdów + logi + raporty zarządzających ruchem, itp.).

Ponieważ zakładamy półroczną agregację danych, daje to potrzebę stałego utrzymania ok. 3660 GB (3, 7 TB).

3.3.4 Cechy i skalowalność pamięci dyskowej

Pamięć dyskowa, działalność operacyjna:

1. Wymagana niezawodność, odporność na awarie (macierze dyskowe?).
2. Pojemność – wg powyższych oszacowań.

Pamięć dyskowa – składowanie:

1. Dane o pozycjach pojazdów nie będą składowane. Dane będą obsługiwały ruch ciągły.
2. Raporty osób zarządzających ruchem będą okresowo składowane

Serwer musi mieć możliwość powiększenia pojemności dostępnej pamięci dyskowej.

4 Architektura oprogramowania

4.1 Czym są moduły, wymaganie modularyzacji

Moduły stanowią elementy oprogramowania, która mają wzajemnie zdefiniowane interfejsy oraz przydzielone funkcjonalności. *Moduły* spełniają następujące kryteria:

- Każdy z ich zawiera specyficzny zakres funkcjonalny i algorytmy oprogramowania
- Są jednoznacznie identyfikowane i mają własną wersję definiowaną w ramach systemu zarządzania konfiguracją



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

- Spełniają zaprojektowane i stosowane reguły dotyczące prywatyzacji i ukrywania informacji
- Każdy ma zdefiniowany interfejs do innych *Modułów* i do rezydencji zewnętrznych
- Są opracowywane i utrzymywane niezależne od siebie, nie mogą współdzielić kodu (nie dotyczy wspólnych bibliotek)
- Nie komunikują się z innymi *Modułami* i światem zewnętrznym inaczej jak tylko poprzez zdefiniowane interfejsy

W procesie wykonania oprogramowania, w pierwszych fazach pracy, oczekiwane będą dokumenty specyfikujące architekturę rozwiązania i architekturę software'u. Dokumenty te zdefiniują *Moduły* poprzez:

- Rysunek zależności *modułów*
- Opis funkcji poszczególnych *modułów* w systemie
- Opis algorytmów, które poszczególne wątki będą wykonywać po nastaniu zdarzeń je wywołujących
- Opis interfejsów z innymi *Modułami* lub rezydencjami danych (także zewnętrznymi)

4.2 Interfejsy pomiędzy modułami i do zewnętrznych rezydencji danych

Każdy *Moduł* ma zdefiniowany interfejs do innych *Modułów* i do rezydencji zewnętrznych.

Interfejs pomiędzy *Modułami* jest oparty o komunikację pomiędzy wątkami z wykorzystaniem technologii programistycznych, bez pośrednictwa pamięci masowej.

Ewentualna potrzeba wykorzystanie w tych interfejsach pamięci masowej wymaga udokumentowanej argumentacji i uzgodnienia.

4.3 Logi pracy Modułów

Oczekuje się, że każdy z *Modułów* ma notować w logach systemu aplikacyjnego znaczone datą i godziną komunikaty o zdarzeniach wyjątkowych. Minimalnym zakresem są informacje o nieprawidłowej pracy oprogramowania (np. przechwyconego błędu programistycznego) oraz błędów transmisji.

5 Wymagania wobec technologii i sposobu wykonania

5.1 Wymagania jakościowe

Oprogramowanie winno spełniać wymagania jakościowe zawarte w dokumencie Krajowe Ramy Interoperacyjności (KRI). Chodzi o zalecenia, które są adekwatne do zakresu przygotowywanego systemu.

Na przykład następujące „minimalne” wymagania:



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

- „2) minimalne wymagania dla rejestrów publicznych i wymiany informacji w postaci elektronicznej;
- 3) minimalne wymagania dla systemów teleinformatycznych, w tym:
 - a) specyfikację formatów danych oraz protokołów komunikacyjnych i szyfrujących, które mają być stosowane w oprogramowaniu interfejsowym,
 - b) sposoby zapewnienia bezpieczeństwa przy wymianie informacji,
 - c) standardy techniczne zapewniające wymianę informacji z udziałem podmiotów publicznych z uwzględnieniem wymiany transgranicznej, sposoby zapewnienia dostępu do zasobów informacji podmiotów publicznych dla osób niepełnosprawnych.”

Oraz:

- „§ 15.1. Systemy teleinformatyczne używane przez podmioty realizujące zadania publiczne projektuje się, wdraża oraz eksploatuje z uwzględnieniem ich funkcjonalności, niezawodności, używalności, wydajności, przenoszalności i pielęgnowalności, przy zastosowaniu norm oraz uznanych w obrocie profesjonalnym standardów i metodyk.
2. Zarządzanie usługami realizowanymi przez systemy teleinformatyczne ma na celu dostarczanie tych usług na deklarowanym poziomie dostępności i odbywa się w oparciu o udokumentowane procedury.”

5.2 Technologia oprogramowania

Preferowanym językiem (technologią) oprogramowania jest Java i jej standardowe biblioteki.

Wymagane jest wykonanie oprogramowania w podejściu obiektowym.

Oferent opíše w ofercie planowaną technologię oprogramowania, m.in.:

- Język, technologia oprogramowania (także wersja)
- Biblioteki zewnętrzne (także wersje)
- Oprogramowanie baz danych (także wersje)
- Dodatkowe narzędzia informatyczne wykorzystywane do pracy (środowisko rozwoju oprogramowania, aplikacje monitorujące jakość kodu, narzędzia zarządzania konfiguracją)
- Zależność technologii od systemu operacyjnego i oprogramowania środowiskowego

5.3 Rezydencje danych

Wszelkie formaty plików muszą być zgodne z rekomendacjami zawartymi w KRI. Także, muszą mieć dane ustrukturyzowane, zdatne do wykorzystania w oprogramowaniu opartym o wolno dostępne standardowe biblioteki.

Szczególnymi danymi są pliki parametryczne. Wymaganiem formatem jest XML.



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Zawartość plików parametrycznych będzie uzgodniona w ramach prac nad architekturą rozwiązania i software'u wraz z opisaniem i udokumentowaniem algorytmów poszczególnych modułów.

Wszystkie rezydencje i bazy danych muszą otrzymać dokumentację słownikową, tj.:

- Definicję terminów i pól danych
- Format zapisu i usytuowanie w rekordzie
- Reguły nawigacji w bazie danych
- Relacje logiczne pomiędzy danymi (ewentualnie plikami)

Szczegółowy zestaw informacji słownikowych zostanie uzgodniony w ramach prac nad architekturą rozwiązania i software'u wraz z opisaniem i udokumentowaniem algorytmów poszczególnych modułów. Oczekujemy spójności pojęć w ramach wszystkich słowników, jakie powstaną dla tworzonego systemu. Zamawiający może oczekiwać skorygowania poszczególnych pojęć ze względu na potrzeby zgodności ze słownikami tworzonymi w organizacji Zamawiającego.

5.4 Środowisko rozwoju i repozytoria

Środowisko rozwoju oprogramowania będzie zorganizowane dla zespołu wykonawczego Dostawcy według podstawowych reguł zarządzania konfiguracją:

- Poszczególne komponenty oprogramowania (*Moduły*, interfejsy, abstrakty, inne) podlegają wersjonowaniu
- Poszczególne wersje komponentów są przechowywane w repozytorium, z którym pracują wszyscy członkowie zespołu wykonawczego
- Poszczególne wersje zintegrowanych aplikacji są przechowywane w repozytorium i są odniesieniem do wszelkiej dokumentacji testowej i finalnej, a także dla raportów o błędach lub prac modyfikujących.
- Biblioteki własne (także ew. powstające dla potrzeb aplikacji) muszą być wersjonowane i centralnie dysponowane dla wszystkich uczestniczących w rozwoju oprogramowania.
- Programiści korzystają z profesjonalnego środowiska pozwalającego na kontrole jakości kodu przy wykorzystaniu specjalistycznego oprogramowania.

Wersje komponentów oprogramowania, aplikacji i bibliotek muszą być podawane w dokumentacji towarzyszącej przekazywanym rezultatom do Zamawiającego.

5.5 Rozwój oprogramowania

Sekwencje prac związane z fazami rozwoju i przekazywania oprogramowania oparte na modelu „V” wykonania oprogramowania są opisane w osobnym dokumencie.



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

5.6 Dokumentacja

W osobnym dokumencie są opisane fazy prac i reguły ich akceptacji. Tam opisane są standardowe dokumenty, które stanowią integralne części rezultatów prac, przygotowywanego oprogramowania.

6 Opis modułów

Podczas projektowania architektury rozwiązania, zestaw *Modułów* i rozdzielenie pomiędzy nich zakresów funkcjonalnych może różnić się od przedstawionego, pod warunkiem wzajemnych uzgodnień.

6.1 Funkcje modułu odczytu danych o położeniu pojazdów

Moduł zapewnia poprawny pobór danych nadsyłanych z pojazdów poszczególnych operatorów. To dane o położeniu (GPS), czasie rejestracji i atrybutach związanych z pojazdem.

Moduł będzie sparametryzowany tak, by umiejscowienie i identyfikacja źródła odczytu danych mogły być zmieniane i poszerzane o kolejnych operatorów i kolejne (zróżnicowane) formaty rekordów.

Rekordy danych o pozycji pojazdów nadchodzą w interwałach 5-10 sekund i informują o położeniu (GPS), momencie uchwycenia danych oraz wartości różnych atrybutów.

Struktura rekordów będzie opisana (zawartość, formaty) i uzgodniona w ramach prac nad architekturą rozwiązania i architekturą software'u.

Przewidujemy 2 warianty sterowników:

- 1) Z uwzględnieniem protokołu API zadanego przez dostawcę usługi przekazywania danych o pozycjach pojazdów
- 2) Bez uwzględnienia protokołu API (wtedy przewidujemy do 4 sterowników wykonanych według tej samej zasady nawiązywania połączenia)

Planowane jest wykorzystanie w sterownikach technologii nawiązywania połączenia opartej na koncepcji WebSocket. Ostatecznie będzie to potwierdzone (lub zmienione) podczas przygotowywania architektury rozwiązania.

6.1.1 Oczekiwane podstawowe cechy jakościowe

Ze względu na potencjalnie różne źródła, z różnymi technologiami i protokołami udostępniania danych, oczekujemy od modułu następujących cech jakościowych udokumentowanych w fazie budowy architektury rozwiązania:

- Samodzielności Zamawiającego w zakresie poszerzenia zestawu *sterowników* (także po zakończeniu wdrożenia)
- Obudowania nadchodzącego rekordu zadaną parametrami ramką w *sterownikach*

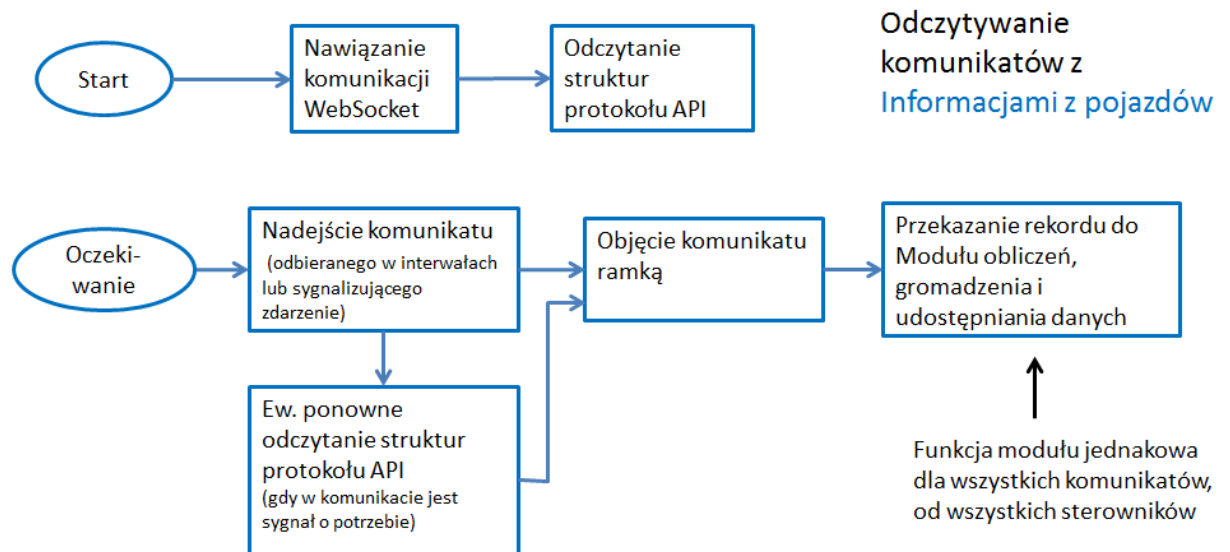


ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

- Realizacji architektury software'u w oparciu o podejście obiektowe
- Przekazania kodu sterowników (także ew. klasy *Abstract* czy *Interface*) Zamawiającemu (plik, wydruk wersji wdrażanych) w ramach kroków odbioru i akceptacji wykonania.

6.1.2 Schemat sekwencji podejmowanej przez wątek czytający



6.1.3 Opis funkcjonalności sterownika

6.1.3.1 Plik parametryczny

Plik jest formułowany w strukturze XML.

Plik definiuje parametry połączenia i przygotowania rekordu dla systemu.

- ! UWAGA: *kursywą* notowane są przykładowe wartości lub ich opis
- ! UWAGA: zestaw parametrów i format pliku jest przykładowy i niepełny. Ostateczny format zostanie uzgodniony podczas opracowywania architektury rozwiązania

```
<drivers>
<driver identyfikator=id_sterownika1>

  <websocket>
    <url>ws://operator:8080/dataStore/positions</url>
    <endpoint>positions</endpoint>
    <charset>UTF-8</charset>
  </websocket>

  <api>
  <logon>
    <user>nazwa_użytkownika</user>
    <password>hasło</password>
  </logon>
```



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

```
<calls>
  <getfunctions>Functions</getfunctions>
  <getvocabulary>Vocabulary</getvocabulary>
  <getinput>{function_name}_inp_params</getinput>
  <getoutput>{function_name}_out_params</getoutput>
</calls>

<framefields>
  <framefield>
    <operator>Mobilis</operator>
    <fieldlength>10</fieldlength>
  </framefield>
  <record>true</record>
</framefields>

</recordtypes>

</api>
</driver >
<driver identyfikator=id_sterownika2>
  <!-- dane kolejnego sterownika -->
</driver>
</drivers >
```

6.1.3.2 Sterownik – nawiązanie komunikacji z WebSocket

Nadawca wystawi port typu WebSocket i w pliku parametrycznym zostaną zawarte dane potrzebne do nawiązania połączenia ze strony odbiorcy.

6.1.3.3 Sterownik – odczytanie struktur protokołu API

Protokół API jest definiowany przez dostawcę usługi SOA (Service Oriented Architecture). Zapisy w pliku parametrycznym są umowne, przykładowe, dla unaocznienia idei a nie jako dokument obowiązujący wykonawcę.

Zobowiązujące struktury definicji protokołu API (dla każdego sterownika) zostaną uzgodnione podczas definiowania architektury rozwiązania.

6.1.3.4 Sterownik – nadejście komunikatu

Przyjęty komunikat może mieć różną długość i zawartość.

Protokół wskaże powiadomienia o zmianach w opisie struktury komunikatów API (por. Komentarz powyżej). Odczytanie takiego „powiadomienia” oznacza konieczność ponownego odczytania struktur protokołu API.

(W przeciwnym przypadku – krok następny).



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

6.1.3.5 Sterownik – objęcie komunikatu ramką

Komunikat zostanie poprzedzony ramką z polami wypełnionymi zgodnie z notacją <framefields> w pliku parametrycznym. Ostateczny zestaw dodawanych pól zostanie uzgodniony podczas opracowywania architektury rozwiązania.

6.2 Interfejs do systemu Rozkładów Jazdy

UWAGA: Specyfikacja interfejsu pomiędzy systemem Rozkładów Jazdy powstanie podczas prac nad architekturą rozwiązania.

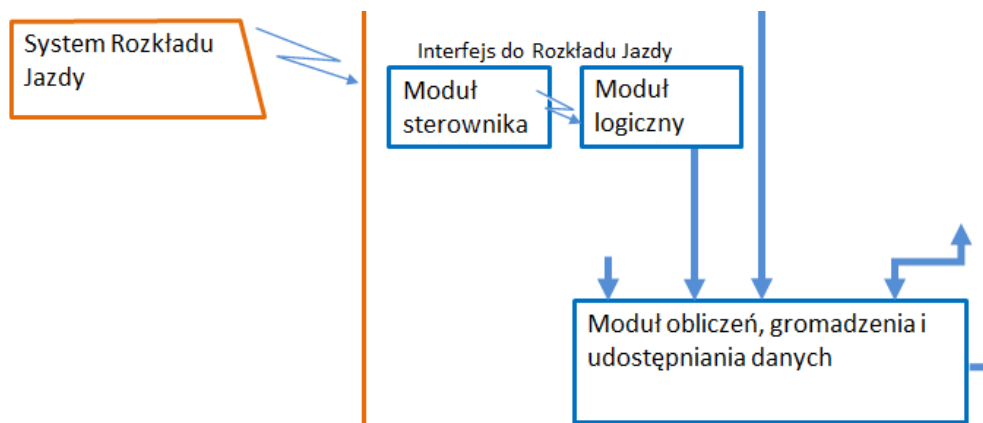
Niniejszy rozdział ma na celu przedstawienie podstawowych założeń przez pokazanie schematu i opisu zamysłu.

6.2.1 Oczekiwane cechy jakościowe

6.2.1.1 Dwa moduły

Oczekiwaniem zamawiającego jest, by interfejs składał się z dwóch modułów komunikujących się mechanizmami sieci Internet (podstawowo: IP i port usługi):

1. Moduł logiczny, współpracujący z modułem „Obliczeń, gromadzenia i udostępniania danych”
2. Moduł sterownika obsługujący dostęp do systemu Rozkładów Jazdy



6.2.1.2 SRJ niezależną aplikacją

System rozkładów jazdy jest centralnym systemem ZTM i rozwija się niezależnie od innych aplikacji. Dialog z nim musi być oparty o mechanizm API (Application Programming Interface) pozwalający na zapytywanie logiczne o dane ujęte w rozkładzie.



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Podczas eksploatacji systemu ZPP zmieni się aplikacja obsługująca tworzenie i udostępnianie Rozkładów Jazdy. Ta zmiana musi odbyć się niezauważalnie dla eksploatacji ZPP, dlatego obie strony (System Rozkładów Jazdy i system ZPP) powinny operować interfejsem zawierającym mechanizm API.

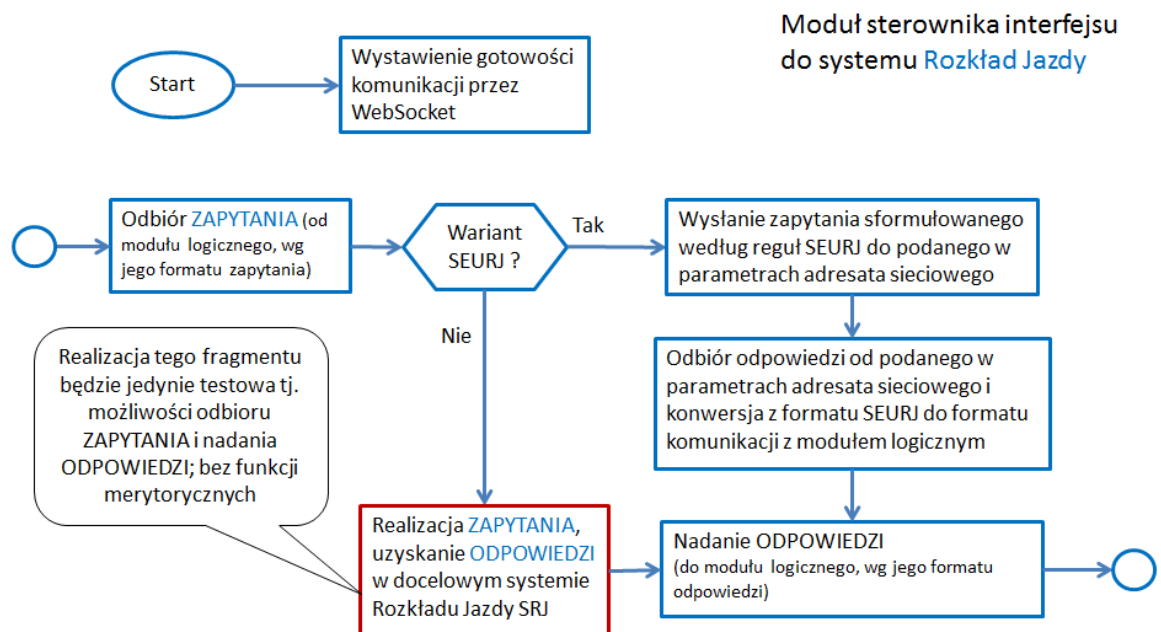
Potrzebne jest oprogramowanie dla dwóch wariantów:

- Wariant SEURJ (System Elektronicznego Udostępniania Rozkładów Jazdy), tj. dostęp do obecnego systemu Rozkładów jazdy
- Wariant docelowego systemu Rozkładów Jazdy (SRJ) tj. dostęp w przyszłości do opracowywanej dopiero nowej aplikacji SRJ

6.2.1.3 Kod modułu

Kod modułu musi być przekazany Zamawiającemu jako jeden z produktów systemu (w pliku i przez zacytowanie w dokumentacji)

6.2.2 Schemat modułu sterownika



Decyzja „Wariant SEURJ ?” jest podjęta na podstawie wartości tagu <interface> w pliku parametrycznym

- W zakresie wariantu SEURJ ('Tak' dla SEURJ) moduł musi efektywnie współpracować podczas testów systemu w logice dostępu do obecnego systemu Rozkładów Jazdy.
- W zakresie wariantu przyszłościowego (tj. 'Nie' dla SEURJ; **czzerwona ramka**) moduł musi efektywnie symulować funkcję dostępu do RJ

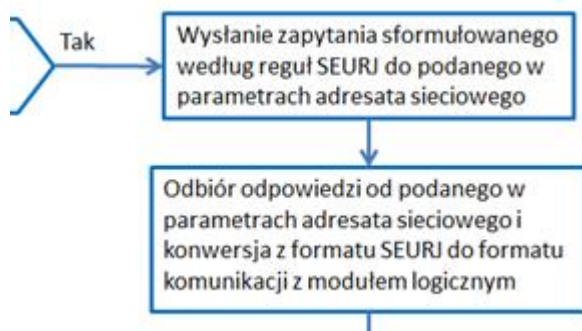


ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

(realizacja funkcji interfejsu ale BEZ dostępu do funkcji merytorycznych RJ)

6.2.2.1 Interfejs do istniejącego oprogramowania (SEURJ) wspomagającego przygotowanie i publikację Rozkładów Jazdy



(to fragment rysunku powyżej)

Interfejs dostępu do danych rozkładowych, nazwany SEURJ (System Elektronicznego Udostępniania Rozkładów Jazdy), który będzie wykorzystywany dla najbliższego okresu pracy systemu jest zdefiniowany w osobnym dokumencie.

Technologicznie, interfejs jest zbudowany w następujący sposób:

- Dostęp jest zapewniony przez nawiązanie połączenia z zadaniem adresem internetowym i portem
- Dostęp wymaga od użytkownika (programu) zalogowania się przez deklarację nazwy i hasła
- Aplikacją po stronie SEURJ jest silnik bazy MySQL
- Zapytaniem jest wywołanie procedury SQL
- Odpowiedzią jest wynik wywołania procedury, czyli lista wierszy o zdefiniowanym układzie (opisanych w dokumentacji) będących odpowiedzią na logikę zapytania.
- Interpretacja wierszy odpowiedzi należy do programu

Przykładowe **ZAPYTANIE** będzie ostatecznie formułowane jako wywołanie procedury SQL; np.:

```
„CALL Pobierz_liste-linii_na_przystanku('2013-03-04', '3007', '06');
```

Przykładowa **ODPOWIEŹ** będzie sformułowana jako tekstowa lista wierszy złożonych z wartości składających się na zestaw pól (kolumn).

```
100  
116  
C04  
E10
```



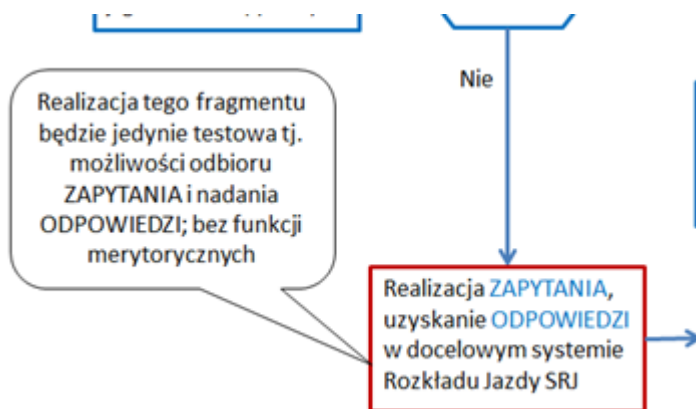
ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Listę należy przekonwertować na dwuwymiarową tablicę wartości poszczególnych pól (kolumn; tu: tylko jedna). :

100
116
C04
E10

6.2.2.2 Interfejs do przyszłego Systemu Rozkładów Jazdy (SRJ)



(to fragment rysunku powyżej)

Dostęp do przyszłego Systemu Rozkładów Jazdy (SRJ) będzie „wstecznie” zgodny z obecnym (SEURJ) w zakresie „modułu logicznego”, a więc co do zestawu i zakresu ZAPYTAŃ i ODPOWIEDZI.

Sposób realizacji ZAPYTANIA i uzyskania ODPOWIEDZI (czerwona ramka na rysunku powyżej) będzie dopracowany w ramach realizacji nowego systemu Rozkładów Jazdy (SRJ)

6.2.2.3 Plik parametryczny

Plik definiuje m.in. parametry połączenia i wariant Rozkładów Jazdy.

UWAGA: kursywą notowane są przykładowe wartości

UWAGA: zestaw parametrów jest przykładowy i niepełny. Ostateczny format i zawartość zostanie uzgodniona podczas opracowywania architektury rozwiązania

Plik parametryczny jest sformułowany w strukturze XML.

```
<timetable>
<access>

<ttsystem>
<ipaddress>192.168.0.11</ipaddress>
```



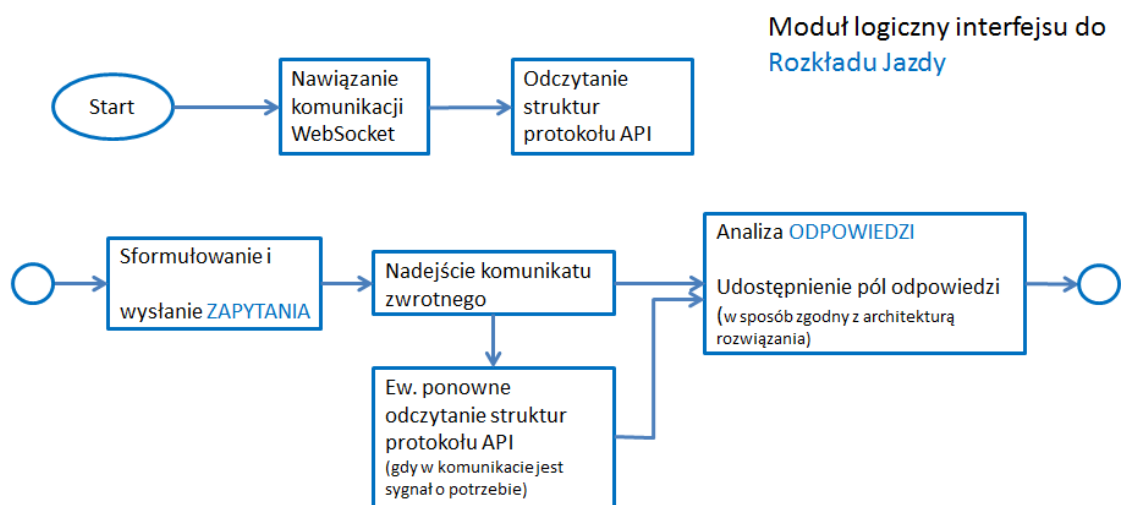
ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

```
<port>25786</port>
<host>http://nazwa_hosta/lokalizacja</host>
<user>nazwa_użytkownika</user>
<password>hasło</password>
</ttsystem>
<zpplogic>
  <ipaddress>192.168.0.14</ipaddress>
  <port>35701</port>
  <host>http://nazwa_hosta/lokalizacja</host>
</zpplogic>

</access>
<variant>
  <interface> SEURJ</interface>
  <!--interface>SYSTEM_RJ</interface-->
</variant>
</timetable>
```

6.2.3 Schemat sekwencji działania modułu logicznego



6.2.3.1 Moduł logiczny – nawiązanie komunikacji

Moduł sterownika wystawia port o określonym numerze, w zadeklarowanym porcie IP, w zadeklarowanej domenie i protokole. W pliku parametrycznym (por. opis modułu sterownika) zostaną zawarte dane potrzebne do nawiązania połączenia pomiędzy modułami interfejsu.

Moduł logiczny nawiązuje połączenie z modułem sterownika uruchomionym jw.



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

6.2.3.2 Moduł logiczny – odczytanie struktur protokołu API

Protokół API jest definiowany przez dostawcę danych.

Słownik danych będzie dostępny w pliku parametrycznym w formacie XML.

UWAGA: *kursywą* notowane są przykładowe wartości

UWAGA: zestaw parametrów jest przykładowy i niepełny. Ostateczny format i zawartość zostanie uzgodniona podczas opracowywania architektury rozwiązania

```
<vocabulary>
<terms>
  <term>
    <name>data</name>
    <format>rrr-mm-dd</format>
    <meaning>data kalendarzowa</meaning>
  </term>
  <term>
    <name>zespól</name>
    <format>decimal4.0</format>
    <meaning>identyfikator zespołu przystanków</meaning>
  </term>
  <term>
    <name>slupek</name>
    <format>decimal2.0</format>
    <meaning>identyfikator słupka w zespole</meaning>
  </term>
  <term>
    <name>nr_linii</name>
    <format>string</format>
    <meaning>identyfikator linii</meaning>
  </term>
</terms>
<call>
  <name>Pobierz_liste_linii_na_przystanku</name>
  <parameter1>data</parameter1>
  <parameter2>zespól</parameter2>
  <parameter3>slupek</parameter3>
  <responsefield1>nr_linii</responsefield1>
</call>
<!-- inne ZAPYTANIA -->
</vocabulary>
```

Struktura i zapis w pliku parametrycznym jest przykładowy; dla unaocznienia idei, a nie jako dokument obowiązujący wykonawcę.



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Zobowiązujące struktury protokołu API zostaną uzgodnione podczas definiowania architektury rozwiązania.

(por. wcześniejszy komentarz do reguł oczekiwanego protokołu API)

6.2.3.3 Moduł logiczny - sformułowanie i wysłanie zapytania

Przykładowa struktura **ZAPYTANIA** – *tablica łańcuchów*:

(podawane są tylko wartości **z komórek zaznaczonych pogrubioną czcionką; tablica z jedną kolumną**)

Znaczenie	treść	
Nazwa zapytania	Pobierz_liste_linii_na_przystanku	
Parametr 1	data	*)
Parametr 2	zespół	*)
Parametr 3	słupek	*)

*) Opis (także format pola) będzie zawarty w słowniku tworzonym przez wykonawcę na podstawie dokumentacji interfejsu SEURJ (por. dalej)

6.2.3.4 Moduł logiczny – nadejście komunikatu

Przyjęty komunikat będzie miał format dwuwymiarowej tablicy zawierającej wiersze wartości poszczególnych pól (kolumn) [ten format może ulec modyfikacji podczas opracowywania architektury rozwiązania]

Przykładowa struktura **ODPOWIEDZI** – *tablica wartości pól*:

(Tablica zawiera tylko wartości **komórek zaznaczonych pogrubioną czcionką**. W przykładzie - tylko jedna kolumna)

Znaczenie	Kolumna 1	Kolumna 2
Wiersz 1	Nr_linii_1	
Wiersz 2	Nr_linii_2	
Wiersz 3	Nr_linii_3	
.....	

6.3 Moduł interfejsu do Rejestru pojazdów

6.3.1 Oczekiwane cechy jakościowe

Rejestr pojazdów rozwija się niezależnie od innych aplikacji. Dlatego dialog z nim musi być oparty o mechanizm API (Application Programming Interface) pozwalający na zapytywanie logiczne o dane ujęte w rozkładzie.

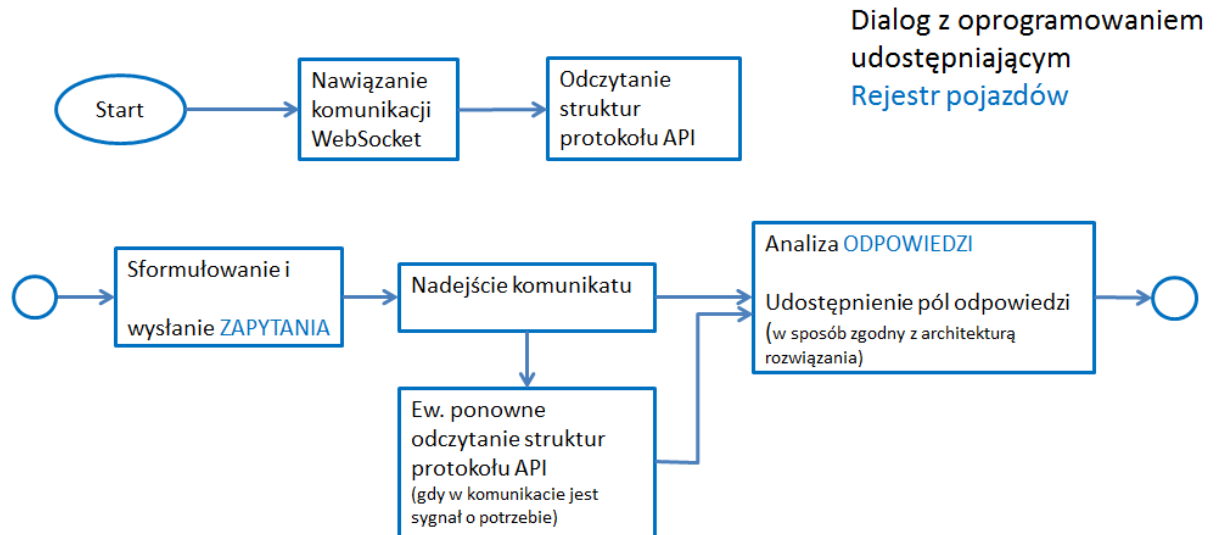


ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Moduł musi realizować (od strony odbiorcy) protokół API zdefiniowany przez dostawcę rejestru. Ostateczne jego uzgodnienie nastąpi podczas opracowywania architektury rozwiązania.

6.3.2 Schemat sekwencji działania sterownika



6.3.3 Opis funkcjonalności sterownika i interfejsu

Sterownik i interfejs do Rejestru Pojazdów zapewnia możliwość zapytywania o konstrukcyjne i wyposażeniowe charakterystyki oraz (dynamiczny) status pojazdu z punktu widzenia jakości i możliwości wykonywania usług przewozowych.

6.3.3.1 Plik parametryczny

Plik jest formułowany w strukturze XML.

Plik definiuje parametry połączenia i przygotowania rekordu dla systemu.

! uwaga: *kursywą* notowane są przykładowe wartości

! uwaga: zestaw parametrów jest przykładowy i niepełny. Ostateczny format zostanie uzgodniony podczas opracowywania architektury rozwiązania

Plik parametryczny jest sformułowany w strukturze XML.

```
<vehicles>
<access>
  <ipaddress>192.168.0.11</ipaddress>
  <port>25786</port>
  <host>http://nazwa_hosta/lokalizacja</host>
  <user>nazwa_użytkownika</user>
  <password>hasło</password>
</access>
</ vehicles >
```




ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

6.3.3.2 Sterownik – nawiązanie komunikacji

Nadawca wystawia port o określonym numerze, w zadeklarowanym porcie IP, w zadeklarowanej domenie i protokole. W pliku parametrycznym zostaną zawarte dane potrzebne do nawiązania połączenia przez skorzystanie z adresu w odpowiedniej postaci.

6.3.3.3 Sterownik – odczytanie struktur protokołu API

Protokół API jest definiowany przez dostawcę danych. Zapis w powyższym pliku parametrycznym jest umowny, przykładowy; dla unaocznienia idei a nie jako dokument obowiązujący wykonawcę.

Zobowiązujące struktury protokołu API zostaną uzgodnione podczas definiowania architektury rozwiązania.

(por. wcześniejszy komentarz do reguł oczekiwanego protokołu API)

6.3.3.4 Sformułowanie i wysłanie zapytania

Przykładowa struktura **ZAPYTANIA** – [tablica łańcuchów](#):

(podawane są tylko wartości z **komórek zaznaczonych pogrubioną czcionką; tablica z jedną kolumną**)

Znaczenie	treść	
Nazwa zapytania	podaj_dane_pojazdu	
Parametr 1	nr_taborowy	*)
Parametr 2	operator	*)

*) Opis (także format pola) będzie zawarty w słowniku tworzonym przez wykonawcę, w uzgodnieniu z opisem interfejsu API wystawianym przez usługę SOA

6.3.3.5 Sterownik – nadejście komunikatu

Przyjęty komunikat może mieć różną długość i zawartość.

Protokół API wskaże ew. powiadomienie o zmianach w opisie struktury komunikatów. Pojawienie się takiego „powiadomienia” oznacza konieczność ponownego odczytania struktur protokołu API.

(W przeciwnym przypadku – krok następny).

Przykładowa struktura **ODPOWIEDZI** – [tablica wartości pól](#):



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

(Tablica zawiera tylko wartości **komórek zaznaczonych pogrubioną czcionką**. W przykładzie - dwie kolumny)

Znaczenie	Kolumna 1	Kolumna 2
Wiersz opisu cechy 1	Pole_cechy_11	Pole_cechy_12
Wiersz opisu cechy 2	Pole_cechy_21	Pole_cechy_22
Wiersz opisu cechy 3	Pole_cechy_31	Pole_cechy_32
.....	

6.4 Moduł API dostępu do danych zgromadzonych w bazie danych

Aplikacja ZPP musi być systemem "otwartym", co znaczy, że wszelkie dane gromadzone w bazie danych muszą być dostępne z zewnątrz według reguł SOA / API.

6.4.1 Oczekiwane cechy jakościowe

Pobór danych z bazy musi odbywać się z wykorzystaniem schematu bazy danych opracowanego w "Module obliczeń, gromadzenia i udostępniania danych"

API musi umożliwić pobór danych z bazy dla wszelkich oczekiwanych raportów i zestawień otrzymywanych przez "Moduł generatora i przeglądania raportów"

API musi być opracowane w oparciu o słowniki towarzyszące wspomnianej bazie danych.

6.4.2 Opis algorytmu

API do ZPP powinna mieć wszelkie cechy opisane w rozdziale wstępnym niniejszego materiału.

Specyfikacja API zostanie opracowana i uzgodniona podczas prac nad architekturą rozwiązania.

6.5 Moduł obliczeń, gromadzenia i udostępniania danych

6.5.1 Oczekiwane cechy jakościowe

Ze względu na centralną funkcję w systemie, oczekujemy od modułu następujących cech jakościowych udokumentowanych w fazie budowy architektury rozwiązania:

- Zapewnienia wspólnych dla wszystkich modułów obliczeń przygotowawczych i wynikowych
- Koordynacji dostępu i notacji w bazie danych dla wszelkich zapisów i pobrań (np. buforowanie rekordów tworzących aktualny kontekst wyświetleń, relacji pomiędzy zakładkami lub funkcjami, itp.)



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

- Jednolitości kontroli i monitorowania dostępu oraz zmian danych
- Jednolitych reguł notowania w logu komunikatów zgłaszanych przez aplikację (informacje, ostrzeżenia, błędy).

6.5.2 Opis funkcjonalności

- Przyjmowanie kolejnych komunikatów o pozycji pojazdów, a następnie
 - Uzupełnianie o identyfikator umowy (wg numeru taboru i id operatora, na podstawie Rejestru pojazdów [por. opis źródeł danych])
 - Skojarzenie struktury rekordu z formatem odczytanym z protokołu API (wg id operatora i typu rekordu)
 - Zanotowanie w bazie danych
- Udostępnianie dla modułu wizualizacji i prezentacji danych, rekordów napływająco na bieżąco z pojazdów
- Udostępnianie dla modułu wizualizacji i prezentacji danych, rekordów w sekwencji rozpoczynającej się o zadanej godzinie i dacie nadania. Dane te będą udostępniane w sekwencji symulującej rzeczywisty upływ czasu.
- Notowanie zleconych zapisów w bazie danych zgodnie ze schematem zaprojektowanym w ramach architektury rozwiązania. Modelowo zakłada się, że wszystkie inne moduły będą kierowały do tego modułu zlecenia notacji. Architektura rozwiązania zawrze ostateczne decyzje wobec tego „modelowego założenia”.
- Schemat bazy danych, w której dokonywane są zapisy przez ten moduł winien uwzględnić następujące oczekiwania:
 - Notacja rekordów nadsyłanych z pojazdów musi odbywać się równoległe z podawaniem ich zawartości do modułu „wizualizacji i raportowania”.
 - Notacja rekordów nadesłanych z pojazdów musi zapewnić efektywne ich odczytywanie w sekwencji stempli czasowych od zadanego momentu (data, godzina). Wtedy także –jako opcja- z uwzględnieniem notacji zdarzeń poczynionych przez Dyżurnego ZTM.
 - Notacja zdarzeń poczynionych przez Dyżurnego ZTM musi być dostępna na bieżąco w trybie przeglądania i emisji do pliku (lub na wydruk), dla zadanej daty (godziny) i autora, synchronicznie (w rozumieniu stempli czasu) do notacji rekordów z pojazdów.
 - Nawigowanie dla efektywnego udostępniania danych dla potrzeb zapytań w logice API i generatora raportów (na ekran, do pliku, do wydruku)

6.6 Moduł generatora i przeglądania raportów

Moduł powinien umożliwić przeglądanie, emisję do plików oraz drukowanie wszelkich zgromadzonych danych w postaci parametryzowanych raportów.

Wszystkie dane zgromadzone w bazie danych muszą być możliwe do unaocznienia (przeгляд, emisja do plików, wydruk).



ZARZĄD TRANSPORTU MIEJSKIEGO

ul. Żelazna 61, 00-848 Warszawa, tel. 22 459 41 00, faks 22 459 42 43
ztm@ztm.waw.pl, www.ztm.waw.pl

Użytkownik będzie miał możliwość zdefiniowania i uzyskiwania poprzez generator raportów innych raportów i zestawień niż wskazane poniżej (jako minimalne, przykładowe).

6.6.1 Przykładowe raporty, minimalny zestaw

Szczegóły w załączniku do OPZ pt. „Opis modułów funkcjonalnych”.

6.6.2 Oczekiwane cechy jakościowe

Szczegóły w załączniku do OPZ pt. „Opis modułów funkcjonalnych”.