



Wymagania
biznesowe
i przypadki
użycia

Jarosław Legierski
Adam Filisiński
Jakub Jankowski
Henryk Rosa
Piotr Pątek
Tomasz Janisiewicz

Architektura Platformy MUNDO

Flickr.com, jbozanowski, CC BY



PREZYDENT
MIASTA STOŁECZNEGO WARSZAWY



Narodowe Centrum
Badań i Rozwoju



Spis treści

1	Cel projektu	2
2	Wymagania architektoniczne	2
3	Lista wymagań biznesowych	3
3.1	Przypadki użycia	3
3.2	Wymagania funkcjonalne	7
3.3	Lista wymagań нефункциональных	10
4	Opis rozwiązania	10
4.1	Statyczna architektura systemu	10
4.2	Architektura systemu z rozbiem na komponenty	12
4.3	Proponowana strategia przechowywania i udostępnianie danych	13
4.4	Dynamiczna architektura rozwiązania	14
4.5	Infrastruktura - wymagania	35
5	Kwestie otwarte	37
6	Słowniczek – lista skrótów	38
7	Bibliografia	40

1 Cel projektu

Celem projektu MUNDO [1] jest diagnoza możliwości ekspozycji danych oraz przygotowanie zestawów danych, które mogą zostać udostępnione przez Miasto Warszawa w modelu Open Data, a przede wszystkim budowa warstwy ekspozycji dla tychże danych w formie platformy eksponującej API. Projekt MUNDO łączy w sobie opracowanie technologii do udostępnienia otwartych danych z metodologią społeczną, dotyczącą włączania różnych grup społecznych w działania na rzecz poprawy życia w mieście. W ramach prac konsorcjum prowadzone są działania, które przygotowują techniczne zaplecze dla części społecznej projektu. Natomiast w części pilotażowej konsorcjum przeprowadzi konkurs na aplikacje internetowe dla Warszawy, by w ten sposób przekonać społeczeństwo o wadze otwartych danych (szczególnie publicznych) dla rozwoju miasta i ułatwienia funkcjonowania jego mieszkańców.

Innowacyjność projektu MUNDO polega na jego złożoności oraz na łączeniu ram technicznych z ramami metodologii partycypacji społecznej w życiu miasta. Wartością projektu jest również jego replikowalność w innych miastach Polski, gdyż zarówno oprogramowanie platformy do ekspozycji otwartych danych, jak i metodologia prowadzenia działań społecznych zostaną udostępnione do wykorzystania dla zainteresowanych jednostek z użyciem wolnych licencji (GPL, LGPL, FDL, CC).

2 Wymagania architektoniczne

Dla platformy eksponującej Otwarte Dane, budowanej w ramach projektu MUNDO przyjęto następujące wymagania:

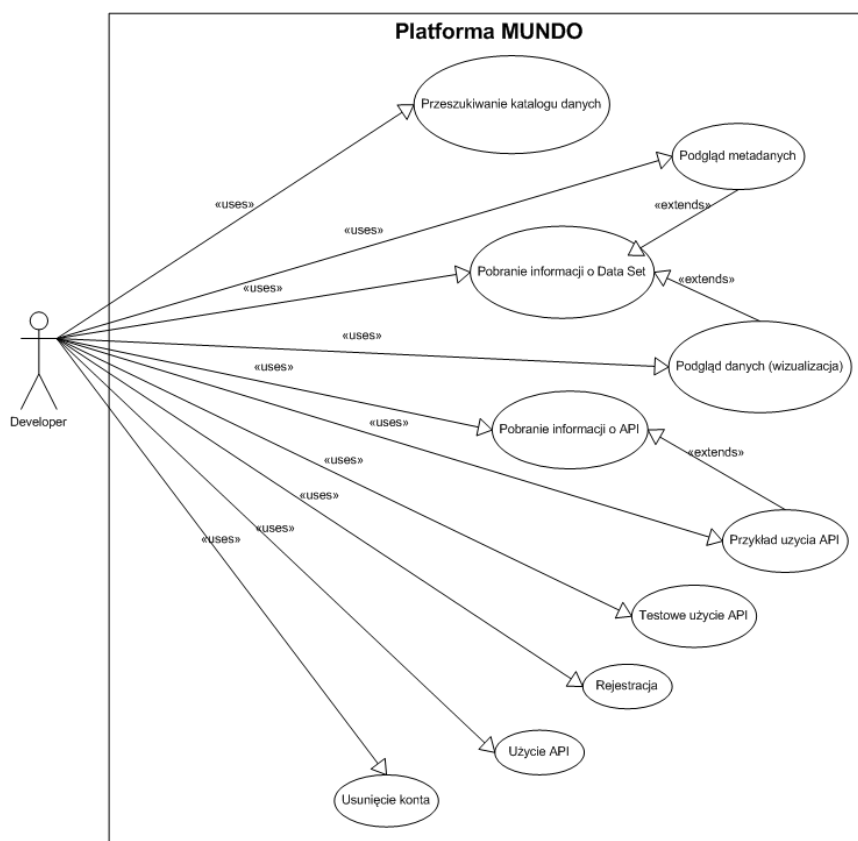
- Użycie open source – platforma powinna być zbudowana z użyciem otwartego oprogramowania.
- Użycie platformy CKAN w charakterze katalogu danych – system CKAN jest bazującym na web oprogramowaniem open source zbudowanym przez Open Knowledge Foundation, służącym do gromadzenia i ekspozycji danych otwartych. CKAN wspiera również szeroki zakres standardów dostępu do open data (SPARQL, RDF, Dublin Core, DCAT). Ze względu na fakt, iż jednym z elementów platformy CKAN jest katalog danych – CKAN zostanie on reużyty w ramach projektu MUNDO.
- Użycie CKAN w charakterze repozytorium danych statycznych (static data sets) – platforma CKAN poza funkcją katalogu posiada funkcjonalność repozytorium danych wraz z opcją ich wizualizacji.
- Użycie Javy do stworzenia Function Server'a dla danych dynamicznych (dynamic data sets) – dane zmieniające się w czasie (z szczególnym uwzględnieniem danych czasu rzeczywistego – real time data sets), których źródłem są np. usługi sieciowe i bazy danych. Ze względu na dynamiczną charakterystykę tych informacji – dynamiczne otwarte dane wymagają stworzenia dedykowanego API serwera opartego na platformę JBoss/Java.
- Użycie wirtualizacji serwerów – wszystkie element platformy powinny być zwirtualizowane.
- OS Linux – zakłada się użycie systemów operacyjnych open source opartych na Linux.

3 Lista wymagań biznesowych

Ze względu na fakt, iż końcowymi użytkownikami platformy będą twórcy aplikacji bazujących na danych miejskich (programiści, architekci aplikacji itp.) a w mniejszym stopniu osoby zajmujące się analizą danych, w projekcie platformy większy nacisk został położony na kwestie związane z wyszukaniem danego zbioru danych, znalezieniem opisu jego API oraz wyświetleniem przykładowych wywołań API. Kwestie związane z wizualizacją danych potraktowano, jako kwestie drugorzędne.

3.1 Przypadki użycia

Na poniższym rysunku przedstawiono w postaci diagramu UML zbiór przypadków użycia platformy dla aktora będącego programistą (developerem).



Rysunek 1 Przypadki użycia dla Developera

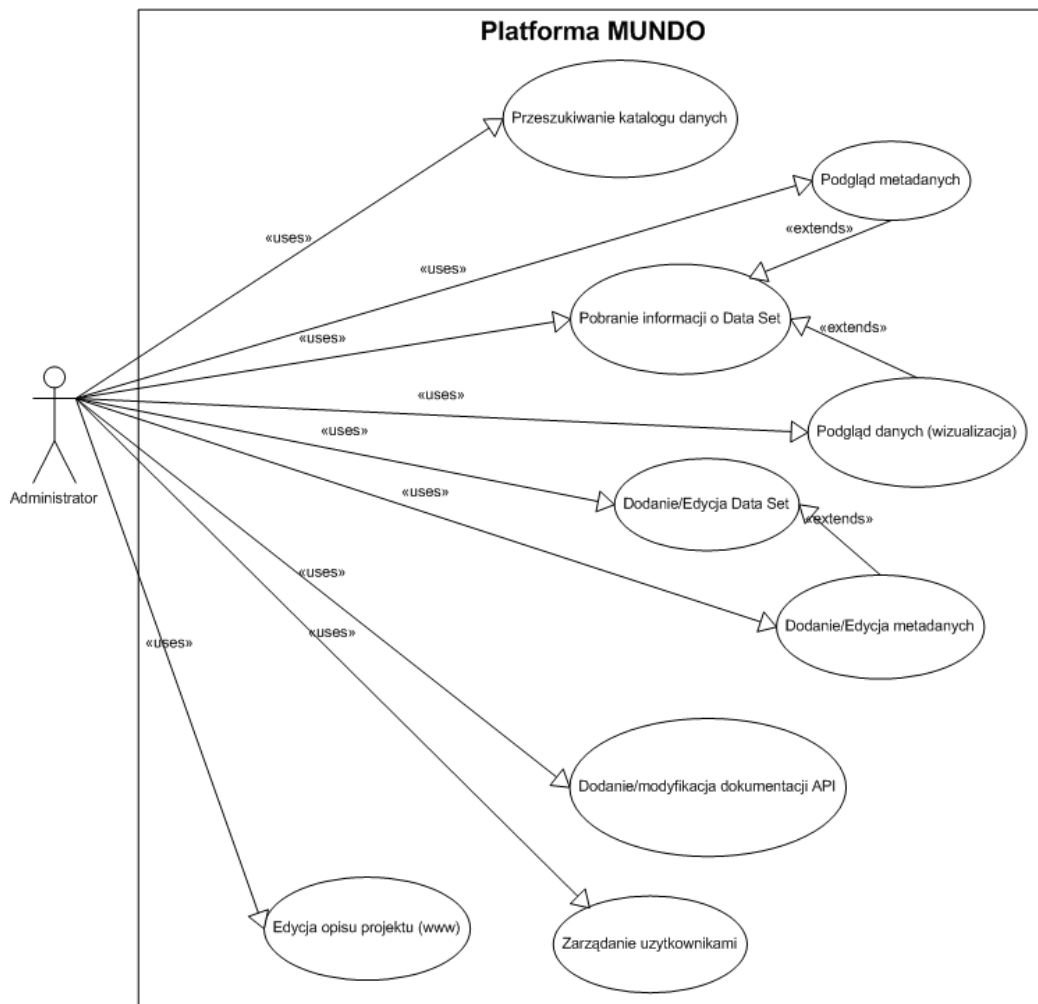
W poniższej tabeli zebrano opisy poszczególnych przypadków użycia dla developera

Tabela 1 Opisy przypadków użycia dla developera

ID	Aktor	Nazwa	Opis	Odpowiedź systemu
Dev	Developer	Przeszukanie katalogu danych	Przeszukanie katalogu danych w celu znalezienia określonego data set	Znalezione zbiory danych spełniające kryteria wyszukiwania (strona www).

		(interface www)	
	Pobranie informacji o Data Set	Pobranie i wyświetlanie szczegółowych informacji o danym zbiorze danych (data set / function set)	Strona www z opisem zbioru danych / zbioru funkcji
	Podgląd danych	Podgląd danych	Plik, tabela, mapka, dla wywołań API odpowiedź z serwera wyświetlone na stronie www
	Podgląd metadanych	Pobranie i wyświetlanie metadanych związanych z danym zbiorem danych	Strona www z opisem metadanych związanych z danym data set
	Odczyt informacji o API	Wyświetlenie dokumentacji API	Dokumentacja API wyświetlona na stronie www i/lub w formie załączników
	Przykład użycia API (doc)	Wyświetlenie dokumentacji z przykładowym użyciem API	Dokumentacja z przykładem użycia API wyświetlona na stronie www i/lub w formie załączników
	Testowe użycie API	Przycisk na stronie - wywołujący dane API z przykładowymi parametrami (opcja)	Wyświetlanie odpowiedzi na stronie www (dane lub odpowiedź XML/JSON z WS) - opcja.
	Rejestracja	Rejestracja użytkownika	Potwierdzenie rejestracji użytkownika - APIKey przyznany użytkownikowi (1 APIKey - per developer)
	Użycie API	Produkcyjne wykorzystywanie API	Komunikacja z platformą na zasadzie request - response (synchroniczna)
	Usunięcie konta	Usunięcie konta z platformy	Potwierdzenie usunięcia konta

Na poniższym rysunku przedstawiono w postaci UML zbiór przypadków użycia platformy dla aktora będącego administratorem.



Rysunek 2 Przypadki użycia dla Administratora

Tabela 2 Opisy przypadków użycia dla administratora

ID	Aktor	Nazwa	Opis	Odpowiedź systemu
Adm	Administrator	Przeszukanie katalogu danych	Przeszukanie katalogu danych w celu znalezienia określonego data set (interface www)	Znalezione zbiory danych spełniające kryteria wyszukiwania (strona www).
		Pobranie informacji o Data Set	Pobranie i wyświetlanie szczegółowych informacji o danym zbiorze danych (data set / function set)	Strona www z opisem zbioru danych
		Podgląd danych	Podgląd danych	Plik, tabela, mapka, dla wywołań API odpowiedź z serwera wyświetlone na stronie www

	Podgląd metadanych	Pobranie i wyświetlenie metadanych związanych z danym zbiorem danych (data set / function set)	Strona www z opisem metadanych związanych z danym data set / function set
	Dodanie nowego Data/Function Set	Dodanie nowego zbioru danych lub zbioru funkcji	Potwierdzenie dodania - strona www z parametrami nowo dodanego dataset / function set
	Edycja Data Set	Edycja parametrów zbioru danych lub zbioru funkcji	Potwierdzenie wykonania modyfikacji - strona www z parametrami zmodyfikowanego data set / function set
	Dodanie metadanych	Dodanie metadanych do zbioru danych lub zbioru funkcji	Potwierdzenie dodania metadanych - strona www z parametrami zmodyfikowanego data set / function set
	Edycja metadanych	Edycja metadanych zbioru danych lub zbioru funkcji	Potwierdzenie wykonania modyfikacji - strona www z parametrami zmodyfikowanego data set / function set
	Dodanie dokumentacji API	Dodanie dokumentacji w formie opisu na stronie www i/lub załącznika (pdf/doc)	Potwierdzenie dodania dokumentacji - strona www z dokumentacją data set / function set
	Statystyki użycia API	Statystyki użycia poszczególnych API w formie strony www	Platforma powinna wyświetlać statystyki ile razy użyte zostało API danego data / function set (opcja)
	Modyfikacja dokumentacji API	Modyfikacja dokumentacji w formie opisu na stronie www i załącznika (pdf/doc)	Zmodyfikowana dokumentacja API dostępna z poziomu strony www
	Zarządzanie użytkownikami	Możliwość modyfikacji parametrów konta użytkownika	Potwierdzenie wykonania operacji - strona www z zmodyfikowanymi parametrami konta

				użytkownika
		Edycja rozszerzonego opisu platformy	Strona www zawierająca opis systemu eksponującego dane i funkcje UM Warszawa w formie API	Wyświetlanie strony www z opisami

3.2 Wymagania funkcjonalne

W poniższej tabeli zebrano listę wymagań funkcjonalnych dla platformy MUNDO

Tabela 3 Lista wymagań funkcjonalnych dla platformy MUNDO

Nr wymagania	Wymaganie	Opis
1	Przeszukanie katalogu danych	Przeszukanie katalogu danych w celu znalezienia określonego data set
2	Pobranie informacji o data set / function set	Pobranie i wyświetlanie szczegółowych informacji o danym zbiorze danych (data set / function set)
3	Podgląd danych	Wyświetlenie danego zbioru w formie: zawartość pliku, tabela, mapka, dla wywołań API odpowiedź z serwera. Podgląd powinien być wyświetlony w przeglądarce www
4	Podgląd metadanych	Metadane związane z danym zbiorem powinny być wyświetlane na stronie www
5	Odczyt informacji o API	Dokumentacja API powinna być wyświetlana na stronie www i/lub w formie załączników (pdf/doc)
6	Przykład użycia API (doc)	Dokumentacja z przykładem użycia API przypisanego do danego zbioru wyświetlona na stronie www i/lub w formie załączników
7	Testowe użycie API	Wyświetlanie odpowiedzi na stronie www (dane lub odpowiedź XML lub JSON z Web Services)
8	Rejestracja	Rejestracja użytkownika w celu uzyskania APIKey (1 APIKey - per developer)
9	Użycie API	Komunikacja z platformą na zasadzie request

		- response (synchroniczna)
10	Usunięcie konta	Usunięcie konta z platformy przez użytkownika lub administratora
11	Dodanie Data Set	Dodanie do platformy nowego zbioru danych lub funkcji poprzez stronę web
12	Edycja Data Set	Edycja zbioru danych lub funkcji poprzez stronę web
13	Usunięcie Data Set	Usunięcie zbioru danych lub funkcji poprzez stronę web
14	Dodanie metadanych	Dodanie do platformy zestawu metadanych opisujących zbiór danych lub funkcji (poprzez stronę web)
15	Edycja metadanych	Edycja metadanych przypisanych do zbioru danych lub funkcji (poprzez stronę web)
16	Dodanie dokumentacji API	Dodanie dokumentacji API poprzez stronę www
17	Modyfikacja dokumentacji API	Modyfikacja dokumentacji API poprzez stronę www
18	Zarządzanie użytkownikami	Modyfikacja parametrów kont użytkowników poprzez stronę web
19	Edycja opisu projektu	Dodatkowy opis platformy/projektu w formie strony www
20	Limitacja wywołań API po stronie południowej (S) platformy - w zależności od data source	Limitacja ilości wywołań API w zależności od źródła danych i funkcji (np. inny limit dla WMS, WFS lub bazy danych) - wymóg realizowany mechanizmami cache
21	Automatyczne przekazywanie zapytania do pierwotnego źródła danych dla Web Serwisów i Real Time Data Sets	Przekazywanie zapytania/odpowiedzi przez platformę w trybie proxy do pierwotnego źródła danych lub funkcji
22	Panel dla Administratora	Zarządzanie platformą poprzez interface oparty o stronę www i przeglądarkę internetową
23	Wersjonowanie API	Wersjonowanie API - wersja umieszczona w url
24	Użycie APIKey	Użycie APIKey - kodu alfanumerycznego przekazanego do serwera API przez

		programy komputerowe wywołujące API
25	Opisy wszystkich API przechowywane w jednym katalogu	Jeden katalog przechowujące informacje o wszystkich zbiorach danych i zbiorach funkcji
26	Wsparcie dla RDF i semantyki (Linked Open Data i RT Linked Open Data)	Każdy zbiór danych i funkcji powinien mieć opis w standardzie RDF
27	Statystyki	Platforma powinna wyświetlać statystyki użycia API. - ile razy użyte zostało API danego data set / function set (opcja) - ilu jest użytkowników platformy/API (opcja)
28	Cache dla wywołań API	Buforowanie odpowiedzi dla niektórych typów danych i funkcji po stronie platformy w celu odciążenia pierwotnego źródła danych i polepszenia jego niezawodności.
29	Konwersja wywołań API	Wywołanie API wysłane do platformy może zostać zmodyfikowane np. w celu uproszczenia składni a wersja bardziej skomplikowana requestu przesłana do pierwotnego źródła danych /funkcji
30	Konwersja odpowiedzi na wywołanie API	Odpowiedź od źródła danych/funkcji może zostać zmodyfikowana przez wystaniem jej deweloperowi.
31	Monitorowanie dostępności pierwotnego źródła danych	Możliwość szybkiego zweryfikowania przez administratora, czy dane źródło danych (WS, WMS, WFS, DB) jest dostępne (aplikacja z interface www)

3.3 Lista wymagań niefunkcjonalnych

Poniżej zebrano zestawienie wymagań niefunkcjonalnych dla platformy MUNDO

Tabela 4 Lista wymagań niefunkcjonalnych dla platformy MUNDO

Nr wymagania	Opis
1	Użycie wirtualizacji zasobów
2	Użycie komponentów open source
3	Użycie platform CKAN w charakterze katalogu danych i repozytorium danych statycznych
4	Użycie systemów operacyjnych open source – Linux

4 Opis rozwiązania

W poniższym rozdziale przedstawiono zarys high level architektury platformy MUNDO oraz zamieszczono szczegółowe schematy z wyszczególnieniem komponentów wchodzących w skład systemu.

4.1 Statyczna architektura systemu

Na rysunku poniżej przedstawiono architekturę systemu. Na podstawie analizy dostępnych typów źródeł danych i funkcji udostępnianych przez UM Warszawa wyróżniono następujące ich typy:

Podział danych ze względu na ich zmienność

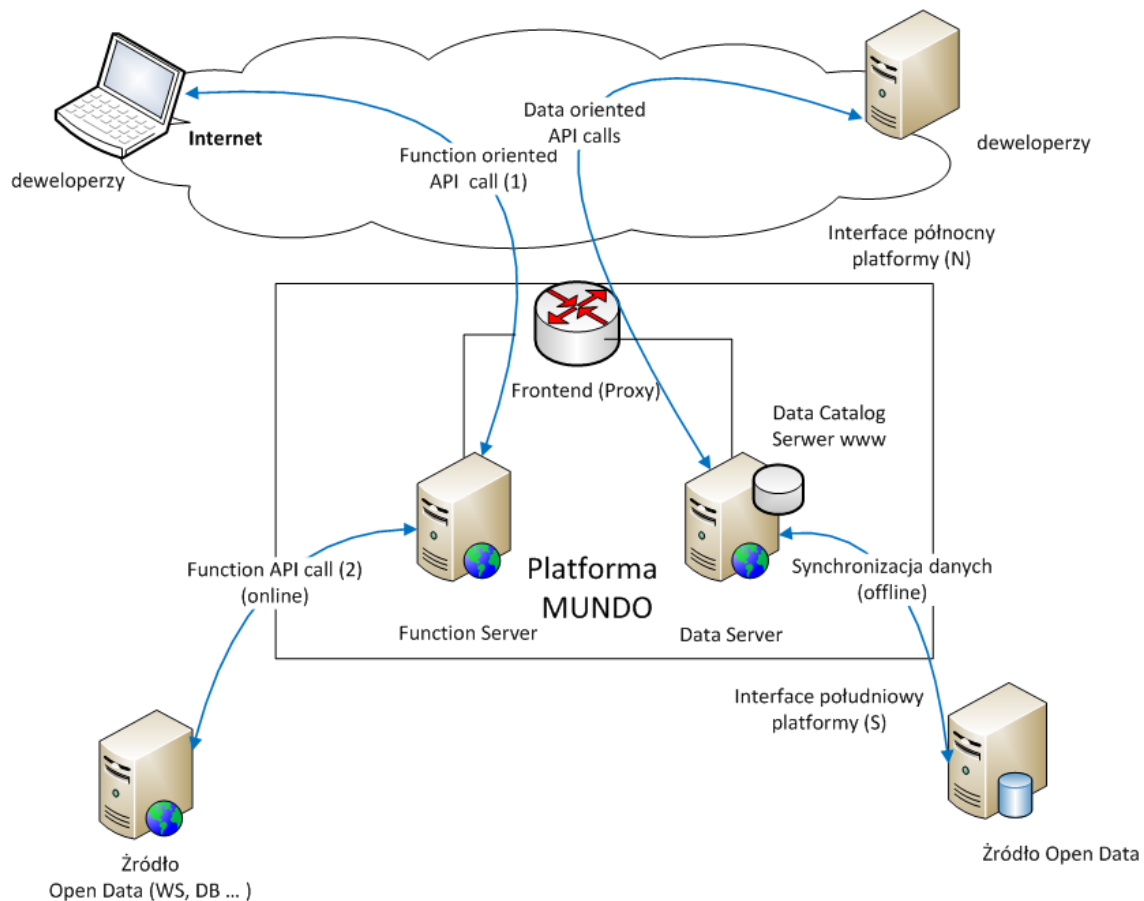
- Dane niezmiennające się – dane statyczne (static data)
- Dane wolnozmiennie (slow changed dynamic data)
- Dane szybko zmieniające się (dynamiczne data)
- Dane czasu rzeczywistego (real time data)

Podział danych ze względu na typ źródła danych:

- Dane plikowe nietabelaryczne
- Dane plikowe tabelaryczne
- Dane mapowe wektorowe (np. ze źródła WFS - Web Feature Service)
- Dane mapowe rastrowe (np. ze źródła WMS - Web Map Service)
- Web Services eksponujące dane (np. dane z miejskiego systemu kolejowego w urzędach)
- Web Services eksponujące funkcje (np. api do zgłaszania zdarzeń w systemie 19115)
- Bazy danych – (np. rozkłady jazdy ZTM, lokalizacja wagonów tramwajów)

Budowana platforma powinna zapewniać dla developera następujące funkcje:

- Dostęp do katalogu danych z opisem poszczególnych API
- Jeden punkt dostępu do API
- Możliwie spójna struktura API do różnych zbiorów danych i funkcji (serializacja i format przesyłanych danych, sposób autoryzacji, kodowanie polskich znaków, układ współrzędnych dla danych geograficznych itp.)
- Możliwość dokonywania konwersji zapytań i odpowiedzi pomiędzy platformą i pierwotnym źródłem danych
- Możliwość ochrony pierwotnego źródła danych przez przeciążeniem i definiowania parametrów SLA (mechanizmy cache).



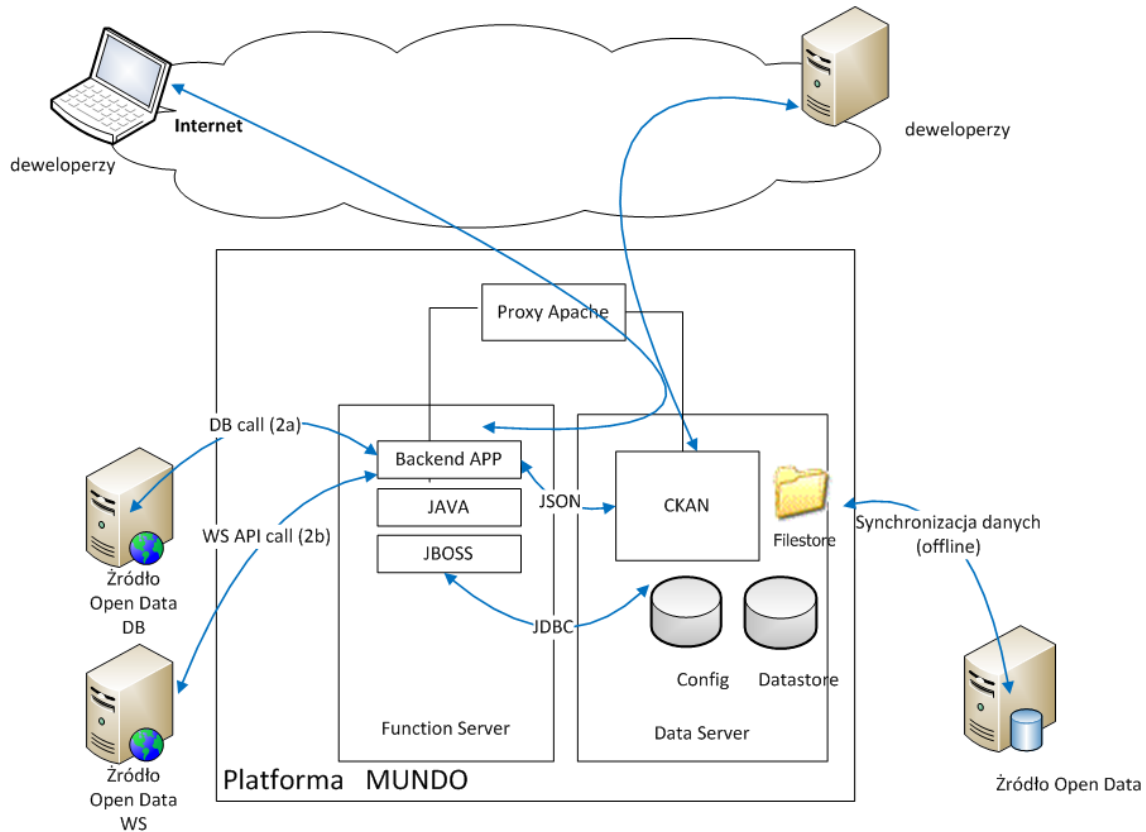
Rysunek 3 Statyczna architektura systemu

Ze względu na zidentyfikowane typy danych do budowy systemu została wybrana architektura zaprezentowana na powyższym rysunku o następujących elementach:

- **Proxy server** – warstwa bezpieczeństwa odpowiedzialna min za rozdział ruchu
- **Data server** – oparty na systemie CKAN serwer, będący: katalogiem danych, serwerem www i repozytorium danych plikowych i tabelarycznych
- **Function server** – serwer będący middleware dla wywołań API do pierwotnych źródeł danych opartych o Web Services, bazy danych itp. System ten ma zapewnić dostęp do danych dynamicznych. Ten podsystem jest również odpowiedzialny za limitowanie wywołań, buforowanie danych, konwersje wywołań oraz konwersję formatu przesyłanych danych w celu uzyskania możliwie spójnego formatu API do wszystkich danych i funkcji UM Warszawa.

4.2 Architektura systemu z rozbiem na komponenty

Na poniższym rysunku przedstawiono architekturę platformy MUNDO z rozbiem na poszczególne komponenty



Rysunek 4 Architektura systemu z rozbiem na komponenty

Do budowy platformy MUNDO założono wykorzystanie następujących komponentów

- 1) Frontend (Serwer Proxy) – serwer www Apache i oprogramowanie PHP
- 2) Data Server – CKAN, komponenty: Pylons, Python, Postgres, SOLR)
- 3) Function Server – JBoss, Java

System podzielono funkcjonalnie pod kątem przetwarzanych typów danych i wywoływanych funkcji

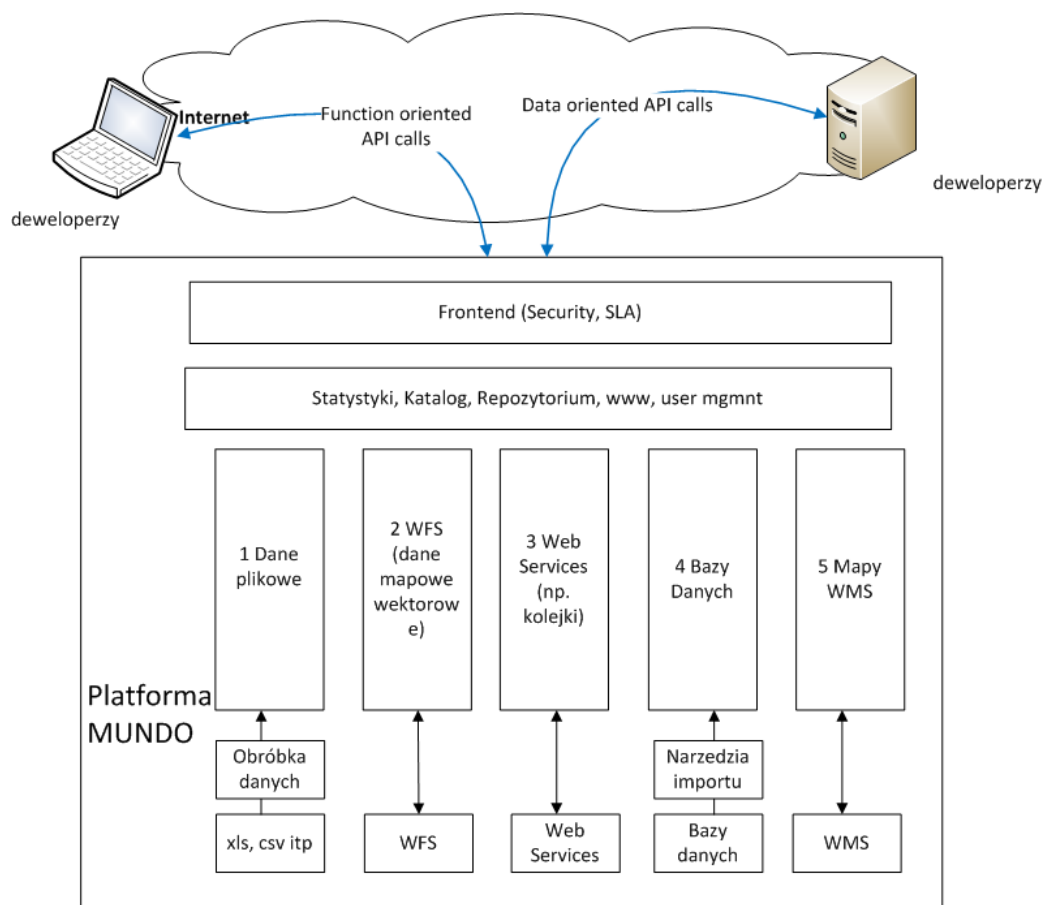
- A) Wywołania API do danych statycznych będą obsługiwane przez platformę CKAN. CKAN będzie pełnił rolę serwera www i katalogu danych oraz repozytorium danych statycznych (wykorzystując rozszerzenia: Filestore, Datastore, Datapusher). Będzie też pełnił funkcje semantic serwera (wsparcie dla RDF) i w ograniczonej formie platformy do wizualizacji data sets.
- B) Wywołania do danych i funkcji dynamicznych eksponowanych min poprzez Web Services, i bazy danych będą obsługiwane przez Function Server – system napisany w Java i bazujący na serwerze aplikacyjnym JBOSS.

Zakłada się że cały ruch do platformy MUNDO w kierunku północnym (do i z Internetu) będzie się odbywał za pośrednictwem serwera Frontend (Proxy) (wszystkie interface www będą dostępne na jednym adresie publicznym i portach 80 i/lub 443). Frontend będzie stanowił serwer Proxy oparty na Apache i aplikacji PHP.

Function Server będzie zintegrowany z Data Server (CKAN) poprzez interfejsy JDBC (integracja z bazą config) i JSON (przekazywanie request'ów z WS API).

4.3 Proponowana strategia przechowywania i udostępnianie danych

Ze względu na typy danych i funkcji udostępnianych przez UM Warszawa, wyróżniono 5 kategorii danych przetwarzanych przez platformę MUNDO:



Rysunek 5 Strategia przechowywania i udostępnianie danych

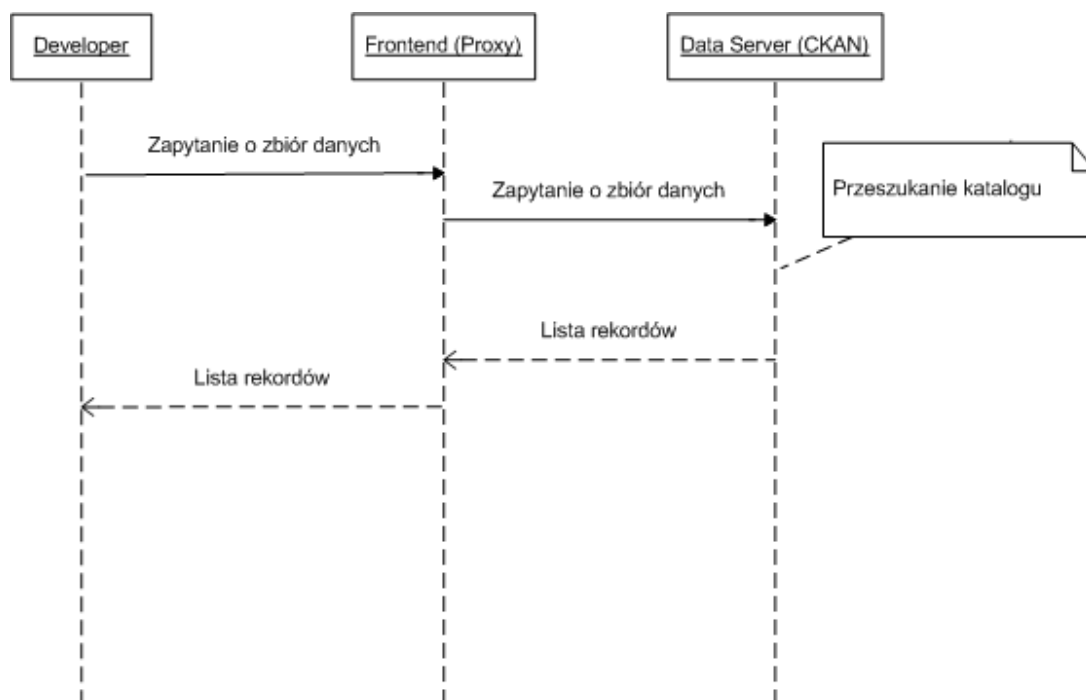
Wyróżniono następujące typy danych:

1. Dane plikowe
 - a) Dane plikowe nietabelaryczne – pliki, które nie mają struktury tabeli (pliki, xml, pdf, pliki graficzne - obrazki itp.)
 - b) Dane plikowe tabelaryczne – wszystkie pliki csv i konwertowane do formatu csv, które mają postać tabeli (pierwszy wiersz stanowi nagłówek z opisem kolumn a pozostałe zawierają dane)
 - c) Dane tabelaryczne z baz danych – dane zawarte w bazach danych, które mogą zostać wyeksportowane do plików tabelarycznych csv
2. Dane mapowe w postaci wektorowej przechowywane w repozytoriach WFS – dane GIS pobierane przez wywołania API do systemu WFS
3. Dane z Web Services - dane przechowywane w formie usług sieciowych Web Services (WS) oraz repozytoriów w formacie html
4. Dane przechowywane w bazach danych – wszelkie dane tabelaryczne (bardzo duże bazy danych, bazy danych szybko zmieniające zawartość), które muszą udostępniane bezpośrednio z pierwotnych repozytoriów.
5. Dane mapowe w postaci map rastrowych WMS - dane GIS przechowywane w systemie WMS eksponowane w formie graficznej (WMS zwraca pliki graficzne: jpeg, png itp.)

4.4 Dynamiczna architektura rozwiązania

Na rysunkach poniżej przedstawiono dynamiczną architekturę platformy MUNDO

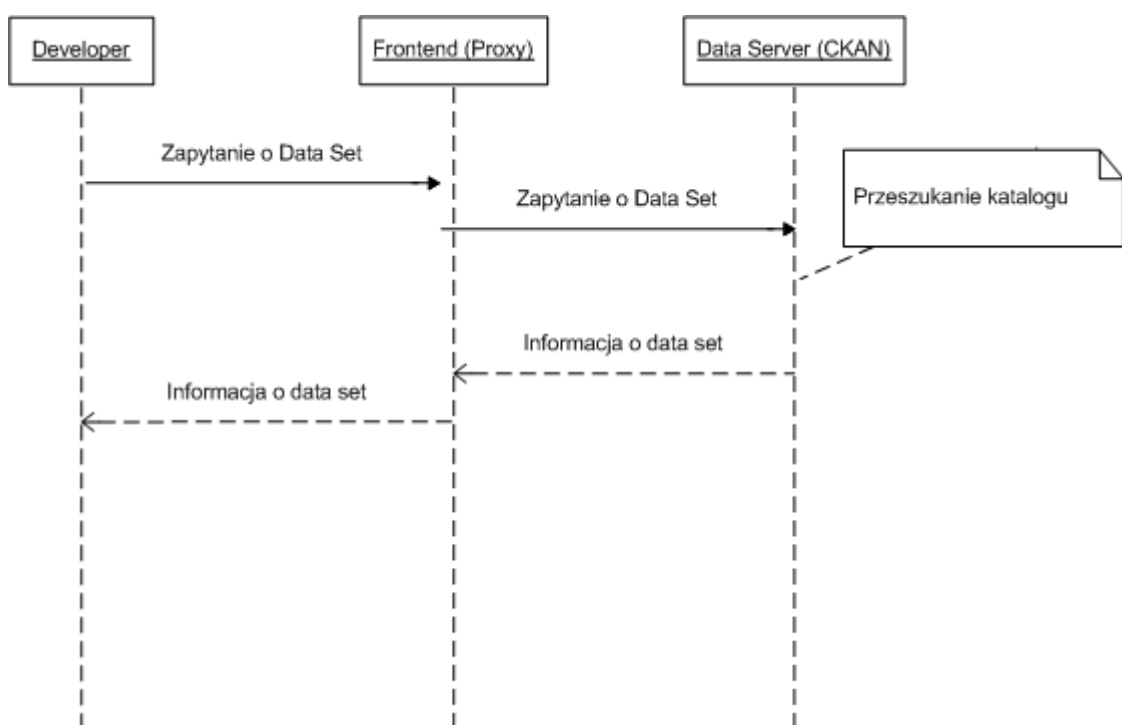
4.4.1 Przeszukanie katalogu danych



Rysunek 6 Przeszukanie katalogu danych

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Przeszukanie katalogu danych	Developer Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Developer przeszukuje katalog danych w poszukiwaniu interesujących go zbiorów danych Wejście: Zapytanie poprzez formularz na stronie www (wyszukiwanie pełno tekstowe) Wyjście: Strona www z listą data sets

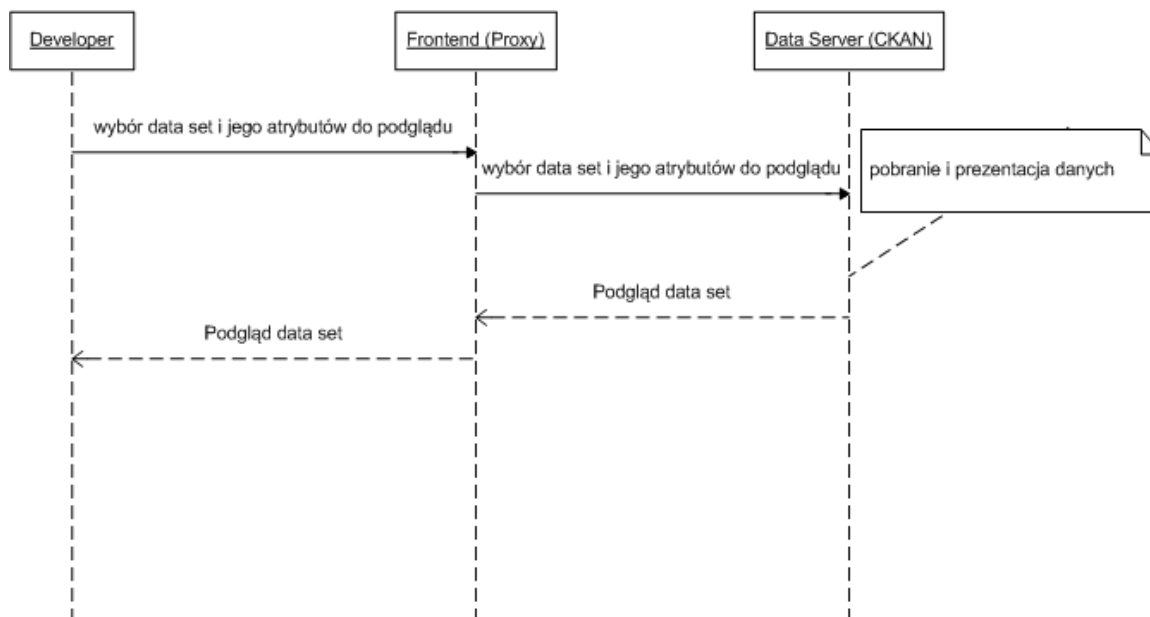
4.4.2 Pobranie informacji o data set



Rysunek 7 Pobranie informacji o Data Set

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Pobranie informacji o data set	Developer Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Developer pobiera pełne informacje o interesującym go data set Wejście: Wybór zbioru danych z listy Wyjście: Strona www z pełnym opisem data set : opis, przez kogo stworzony, lista metadanych

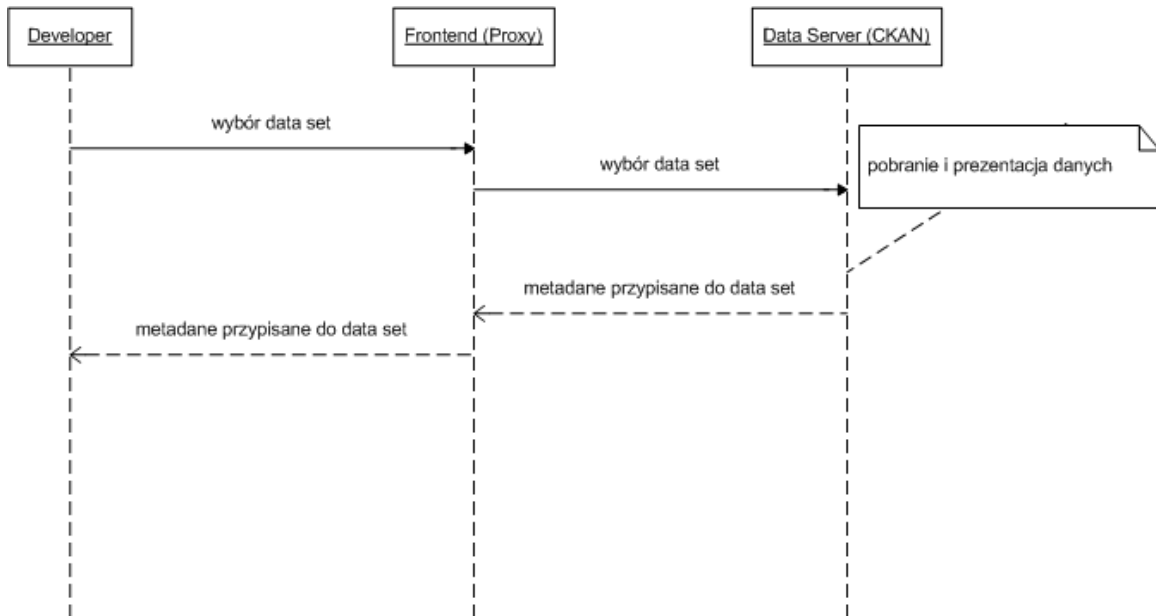
4.4.3 Podgląd danych



Rysunek 8 Podgląd danych

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Podgląd danych	Developer Frontend (Proxy) Data Server (CKAN)	<p>Biznesowe działanie: Developer wyświetla na stronie www dane z wybranego data set</p> <p>Wejście: Wybór zbioru danych z listy, wybór zakresu danych dla wykresu wybór parametrów (X, Y, w tym wykresu)</p> <p>Wyjście: Strona www z danymi: tabela, wykres, JSON lub XML dla źródeł z Web Services</p>

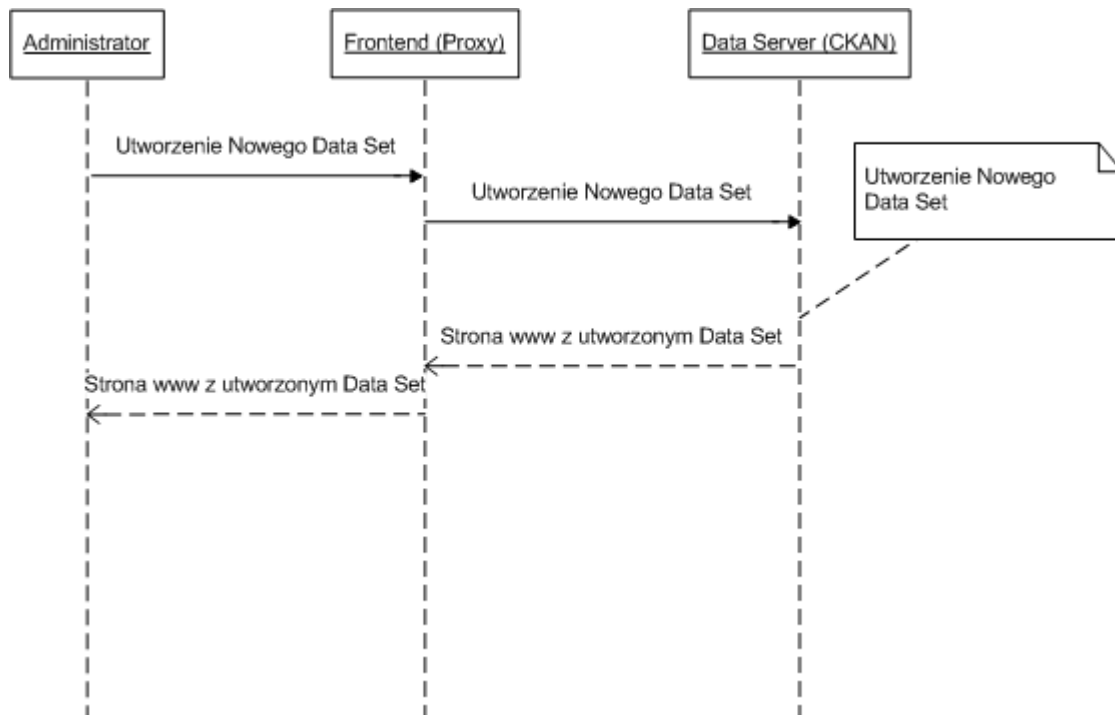
4.4.4 Podgląd metadanych



Rysunek 9 Podgląd metadanych

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Podgląd metadanych	Developer Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Developer wyświetla na stronie www metadane dla wybranego data set Wejście: Wybrany dataset, wyświetl metadane Wyjście: Strona www z metadanymi

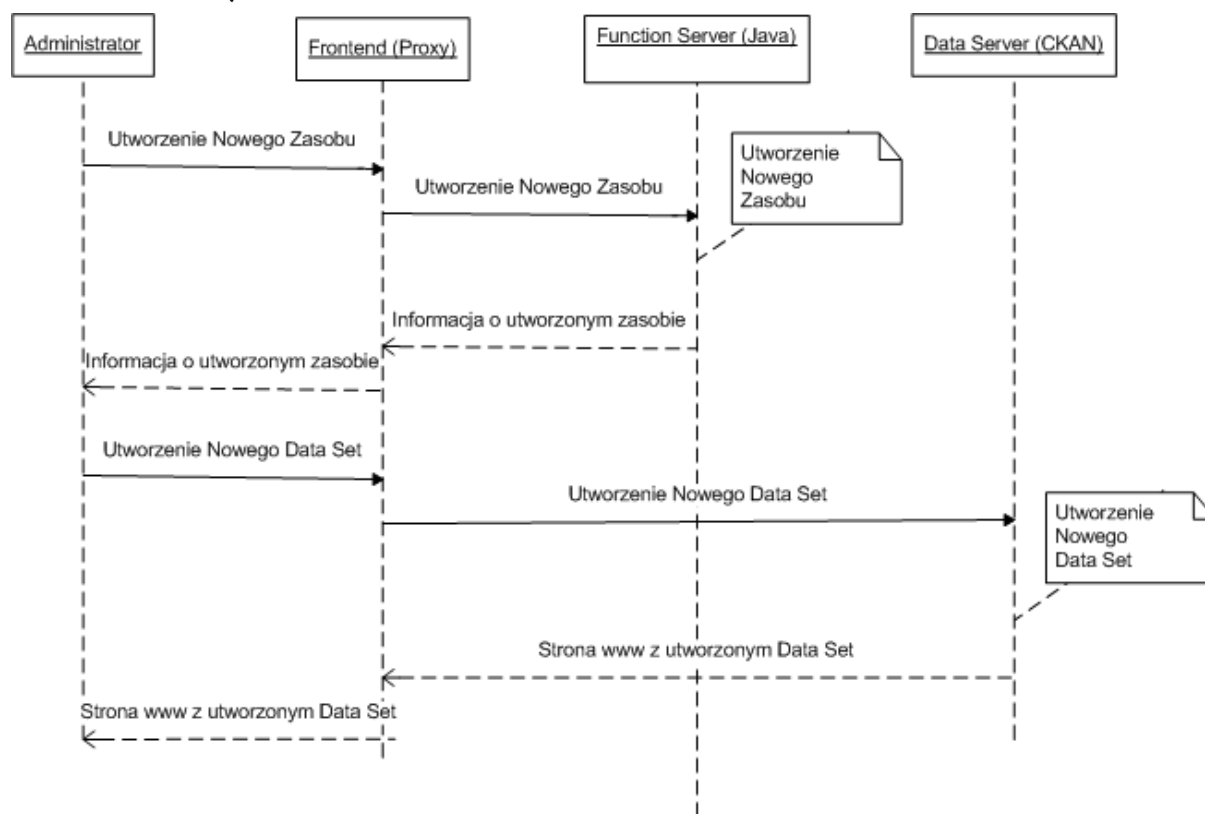
4.4.5 Utworzenie nowego Data Set (w Data Server)



Rysunek 10 Dodanie Data Set (CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Dodanie Data Set (Data Server - CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Administrator tworzy nowy Data Set Wejście: Dane, nazwa, opis organizacja, plik lub url, metadane, słowa kluczowe Wyjście: Strona www z utworzonym data set + ID data set'a

4.4.6 Utworzenie nowego Data Set (w Data Server i Function Server)

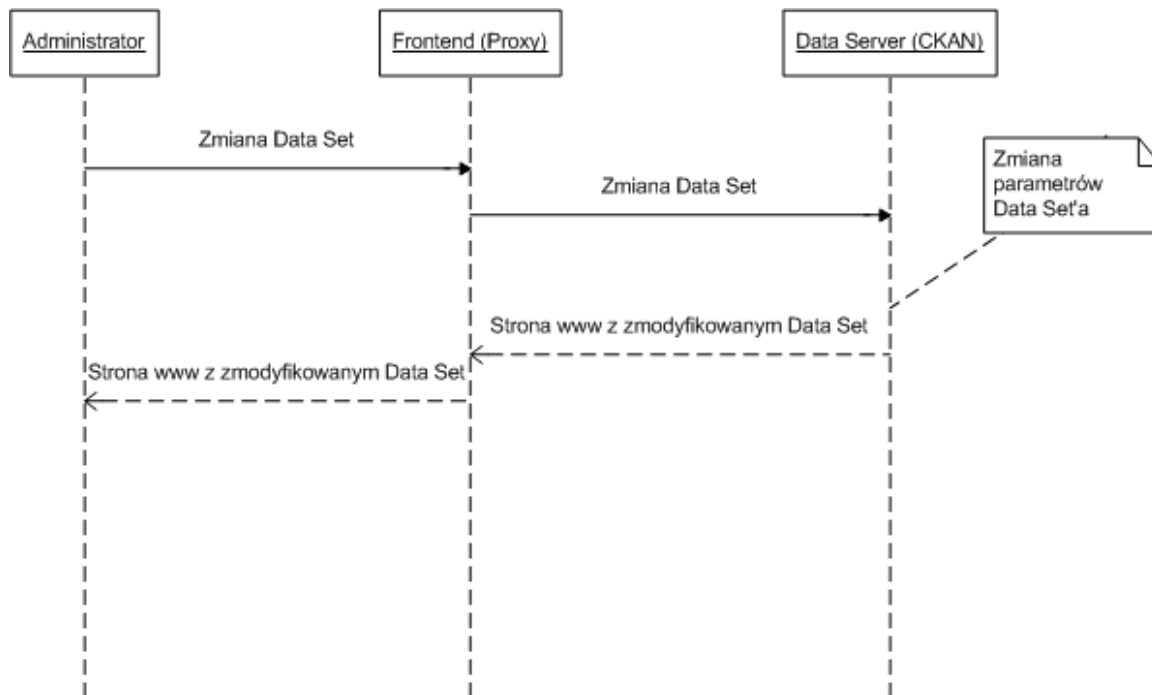


Rysunek 11 Utworzenie nowego Data Set (w Data Server i Function Server)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Dodanie Data Set (Data Server - CKAN + Function Server)	Administrator Frontend (Proxy) Function Server Data Server (CKAN)	Biznesowe działanie (1): Administrator tworzy nowy zasób (zbiór danych/funkcji) w Function Server Wejście: Dane zasobu, nazwa, url źródła danych, url zasobu, limity, nałożenie dodatkowych reguł (np.: konwersje dla WMS) Wyjście: Strona www z utworzonym data set + url zasobu Biznesowe działanie (2):

		Administrator tworzy nowy Data Set w Data Server (CKAN) Wejście: Dane Data set, nazwa, opis organizacja, url, metadane, słowa kluczowe Wyjście: Strona www z utworzonym data set + ID data set'a
--	--	--

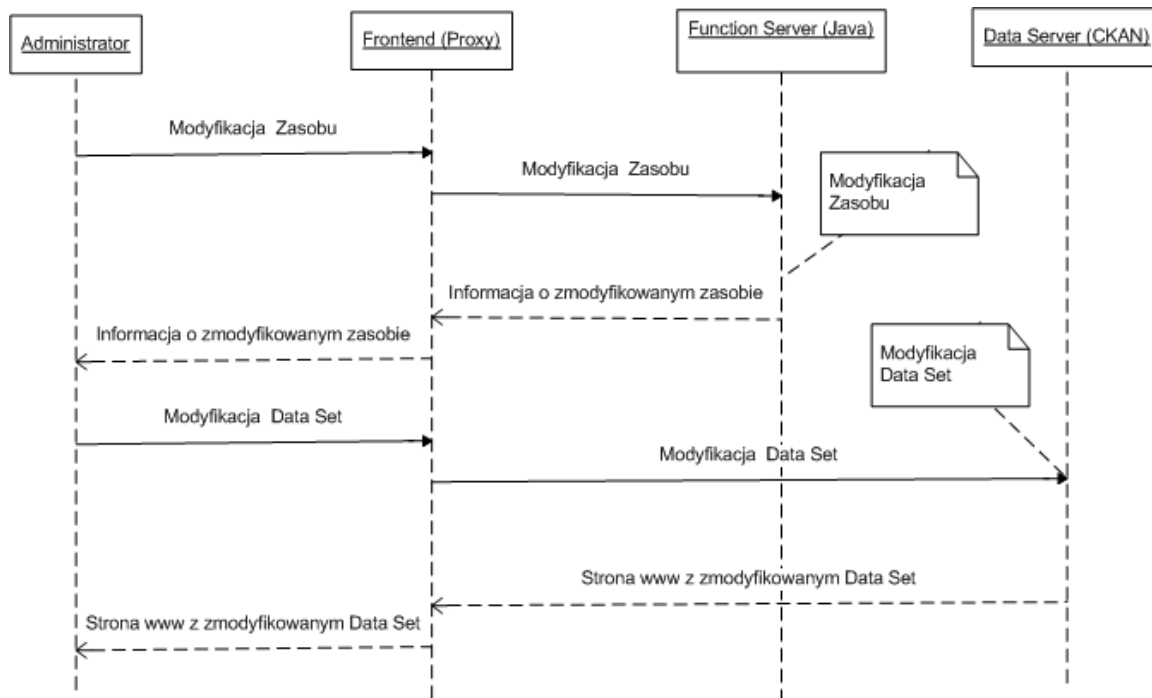
4.4.7 Edycja Data Set (Data Server - CKAN)



Rysunek 12 Edycja Data Set (Data Server - CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Edycja Data Set (Data Server -CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Administrator edytuje (modyfikuje) dany Data Set Wejście: Data set, nazwa, opis organizacja, plik lub url, metadane, słowa kluczowe Wyjście: Strona www z zmodyfikowanym data set, ID data set'a pozostaje niezmienny

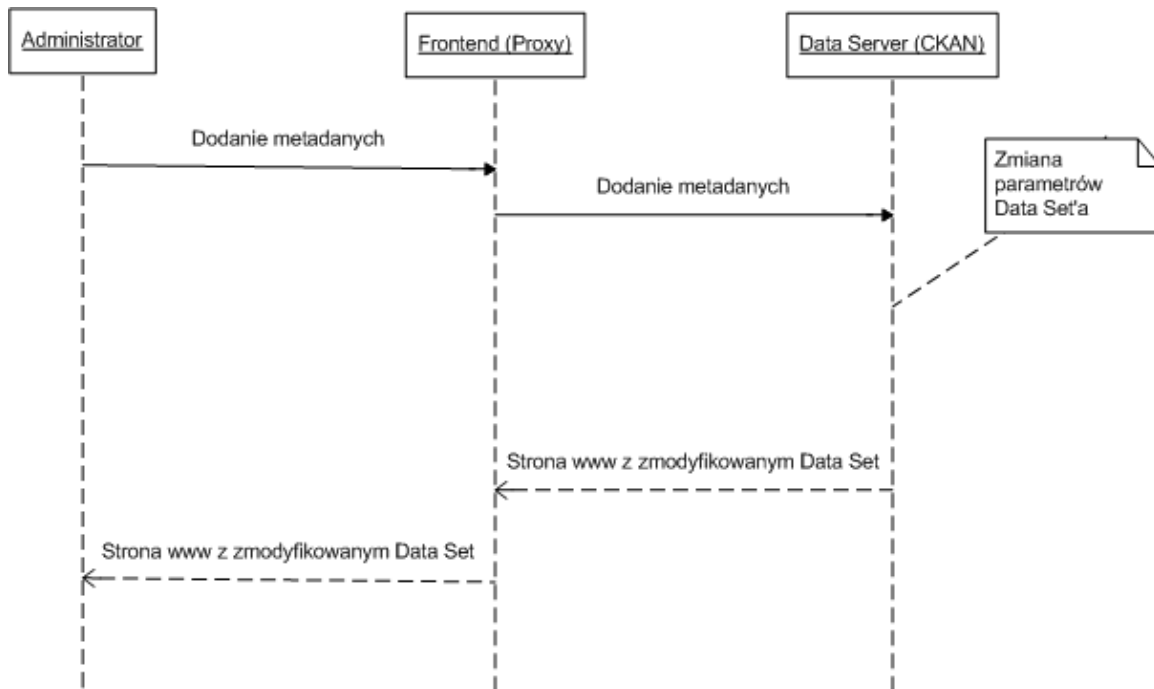
4.4.8 Edycja Data Set (w Function Server i Data Server - CKAN)



Rysunek 13 Edycja Data Set (Data Server CKAN i Function Server)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Edycja Data Set (Data Server - CKAN + Function Server)	Administrator Frontend (Proxy) Function Server Data Server (CKAN)	<p>Biznesowe działanie (1): Administrator modyfikuje zasób (zbiór danych) w Function Server</p> <p>Wejście: Dane zasobu, nazwa, url źródła danych, url zasobu, limity, nałożenie dodatkowych reguł (np.: konwersje dla WMS)</p> <p>Wyjście: Strona www z zmodyfikowanym data set + url zasobu</p> <p>Biznesowe działanie (2): Administrator modyfikuje Data Set w Data Server (CKAN)</p> <p>Wejście: Dane Data set, nazwa, opis organizacja, url, metadane, słowa kluczowe</p> <p>Wyjście: Strona www z zmodyfikowanym data set, ID data set nie ulega zmianie</p>

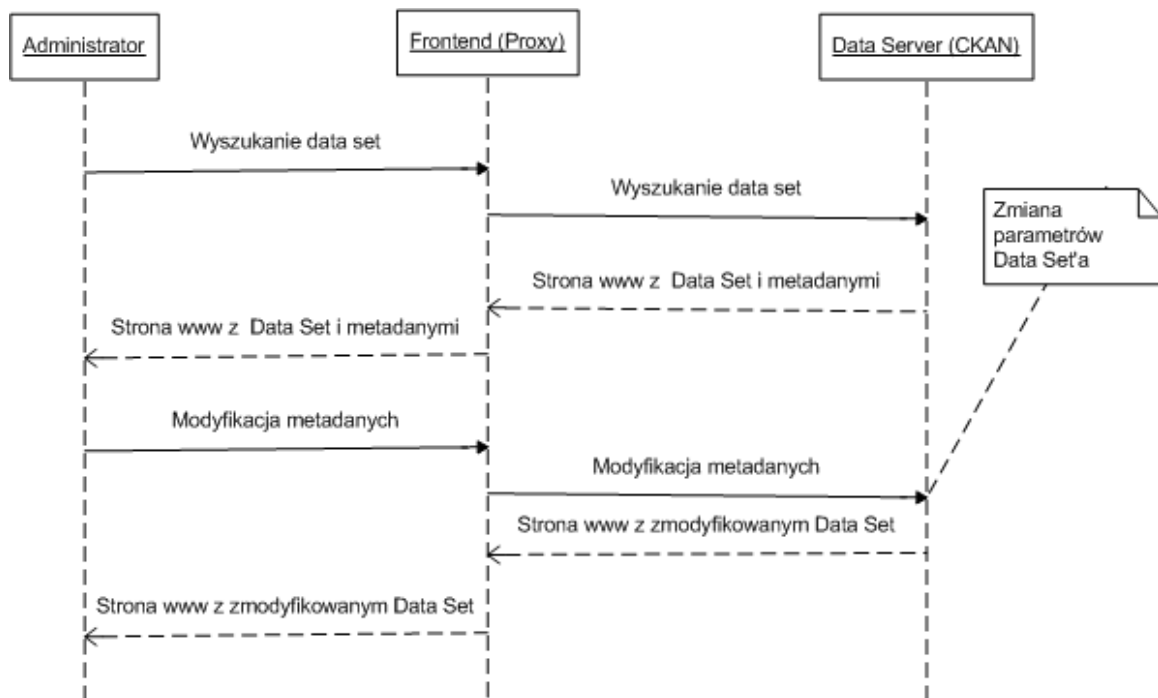
4.4.9 Dodanie metadanych (Data Server - CKAN)



Rysunek 14 Dodanie metadanych (Data server - CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Dodanie metadanych do danego Data Set (CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Administrator dodaje metadane do Data Set Wejście: Metadane, Data set, Wyjście: Strona www z zmodyfikowanym data set

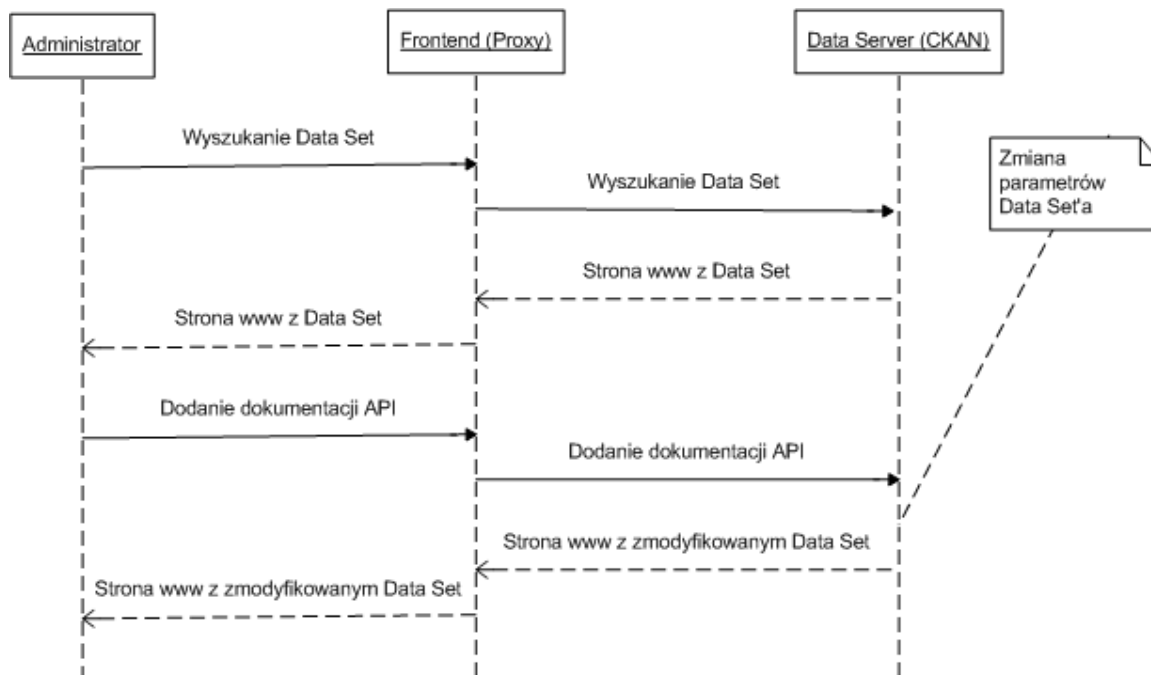
4.4.10 Edycja metadanych (Data Server – CKAN)



Rysunek 15 Edycja metadanych (Data Server - CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Edycja metadanych przypisanych do danego Data Set (Data Server - CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Administrator modyfikuje metadane przypisane do Data Set Wejście: Metadane, Data set Wyjście: Strona www z zmodyfikowanym data set

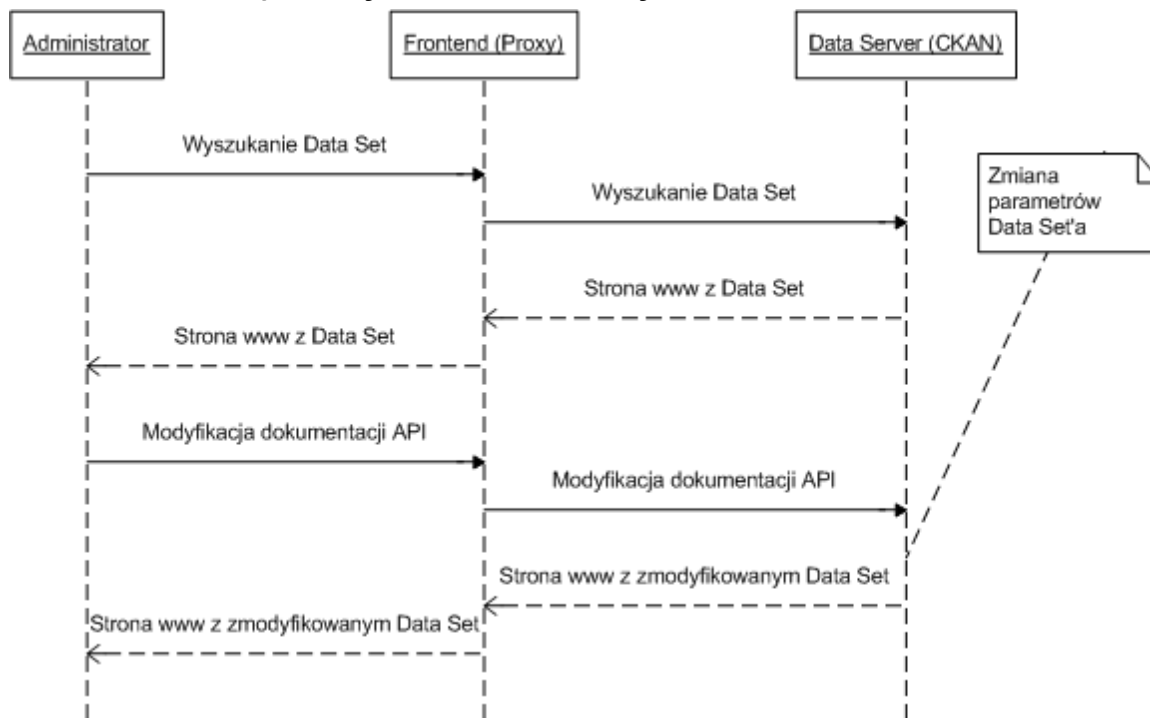
4.4.11 Dodanie dokumentacji API



Rysunek 16 Dodania dokumentacji API (Data Server - CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Dodanie dokumentacji API do danego Data Set	Administrator Frontend (Proxy) Data Server (CKAN)	<p>Biznesowe działanie: Administrator dodaje dokumentację API dedykowaną do danego Data Set</p> <p>Wejście: Data set, Dokumentacja API - załączniki doc lub pdf</p> <p>Wyjście: Strona www z zmodyfikowanym data set (dokumentacją API)</p>

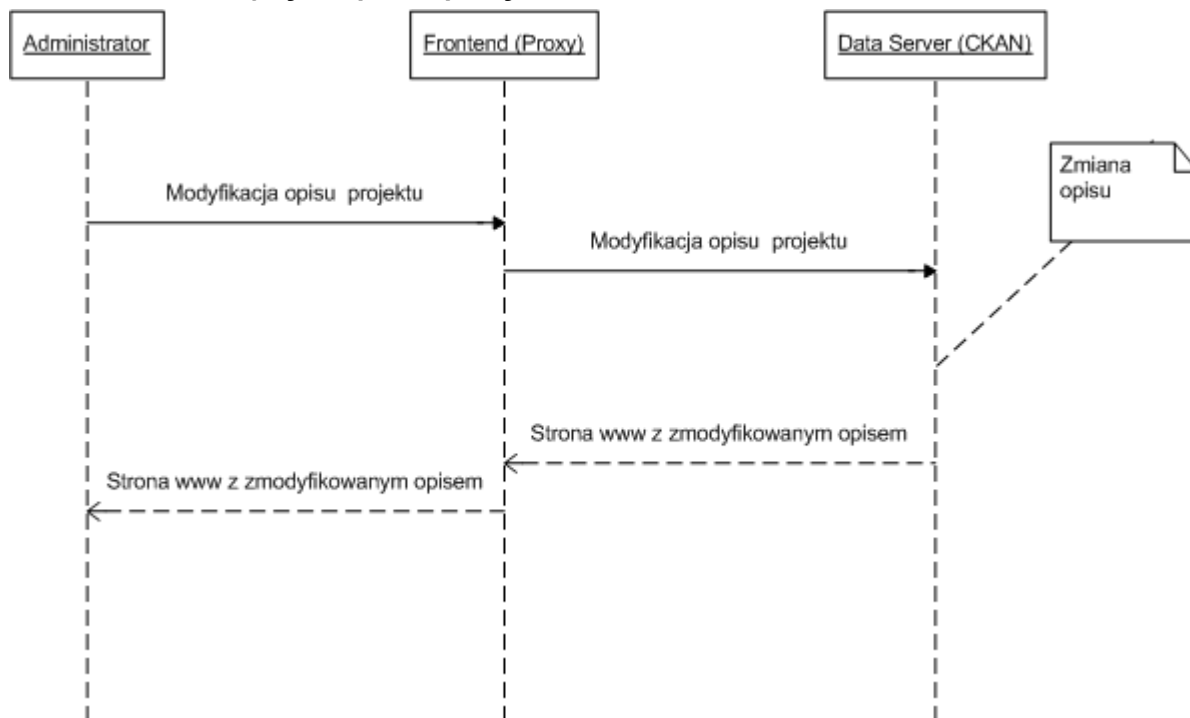
4.4.12 Modyfikacja dokumentacji API



Rysunek 17 Modyfikacja dokumentacji API (Data Server - CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Modyfikacja dokumentacji API do danego Data Set (CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Administrator modyfikuje dokumentację API dedykowaną do danego Data Set Wejście: Data set, Dokumentacja API załączniki doc lub pdf Wyjście: Strona www z zmodyfikowanym data set (dokumentacją API)

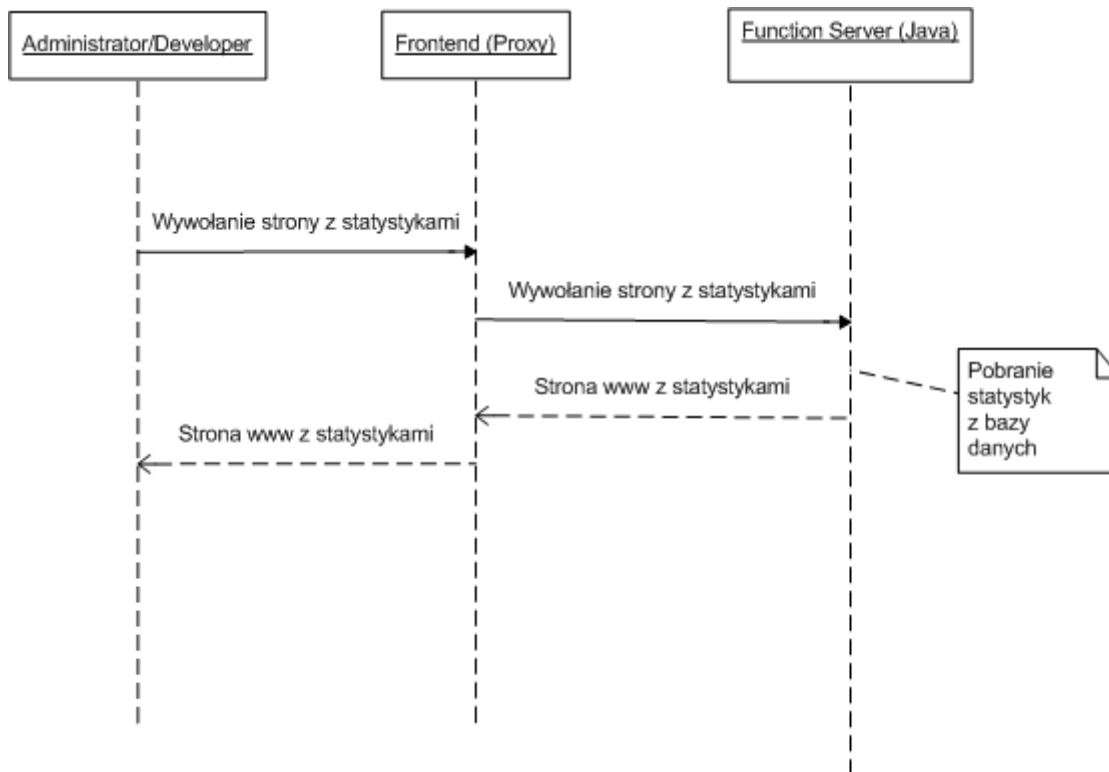
4.4.13 Edycja opisu projektu



Rysunek 18 Edycja opisu projektu (CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Edycja opisu projektu (Data Server - CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Administrator modyfikuje opis projektu korzystając z mechanizmów natywnych lub rozszerzeń CKAN'a Wejście: Opis Wyjście: Strona www opisu projektu z zmodyfikowaną zawartością

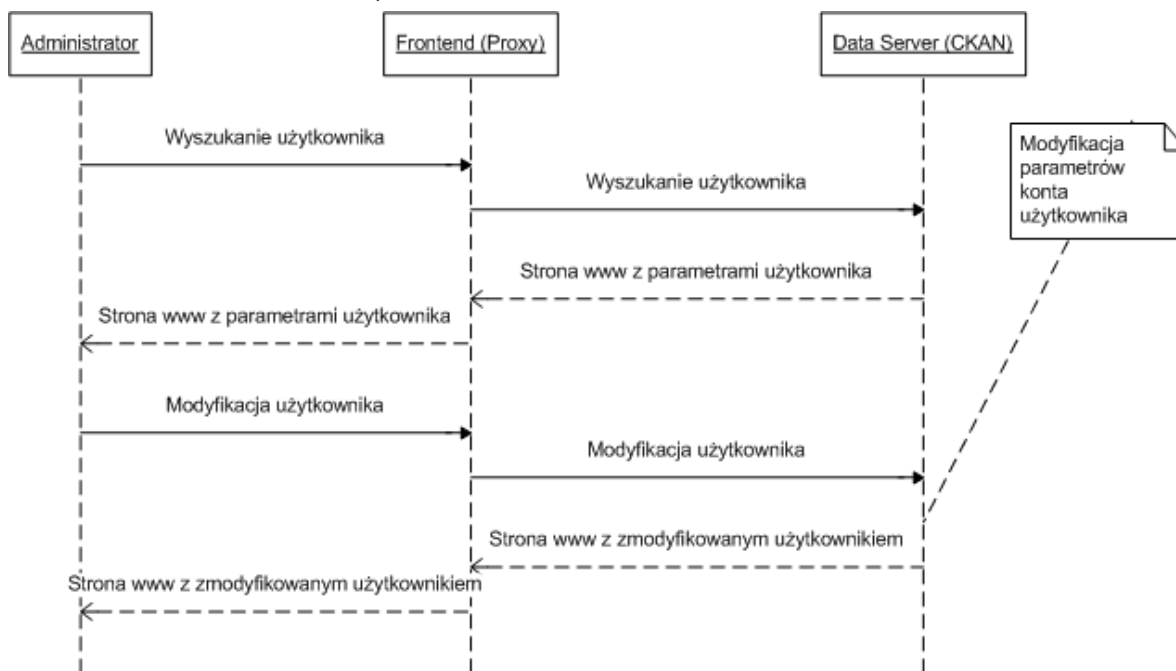
4.4.14 Przeglądnie statystyk użycia API



Rysunek 19 Przeglądnie statystyk użycia API

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Przeglądnie statystyk użycia API	Administrator Developer Frontend (Proxy) Function Server	Biznesowe działanie: Administrator lub developer przegląda statystyku użycia API Wejście: Wywołanie strony www ze statystykami Wyjście: Strona www z statystykami

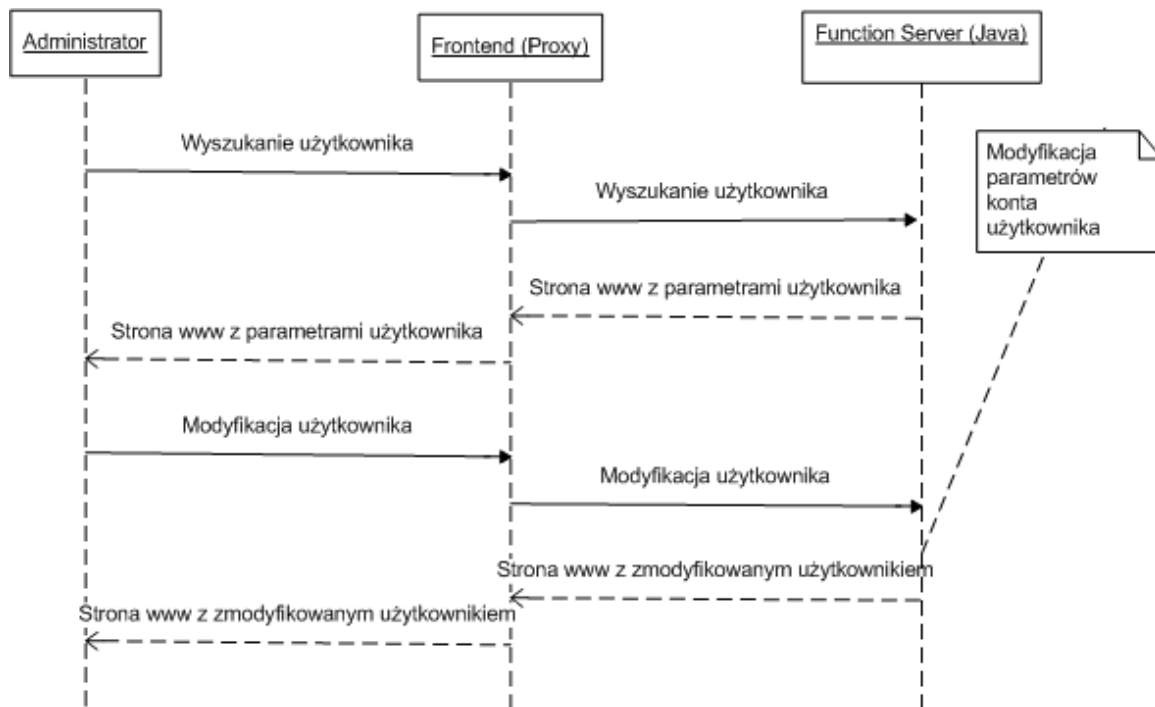
4.4.15 Zarządzanie użytkownikami i administratorami (Data Server – CKAN)



Rysunek 20 Zarządzanie użytkownikami / administratorami
Data Server - CKAN

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Modyfikacja konta użytkownika (Data Server CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	<p>Biznesowe działanie: Administrator modyfikuje konto użytkownika</p> <p>Wejście: Wywołanie strony www parametrami użytkownika i ich modyfikacja</p> <p>Wyjście: Strona www z parametrami konta Zmodyfikowane konto użytkownika</p> <p>Uwaga - przypadek szczególny założenie konta nowego użytkownika</p>

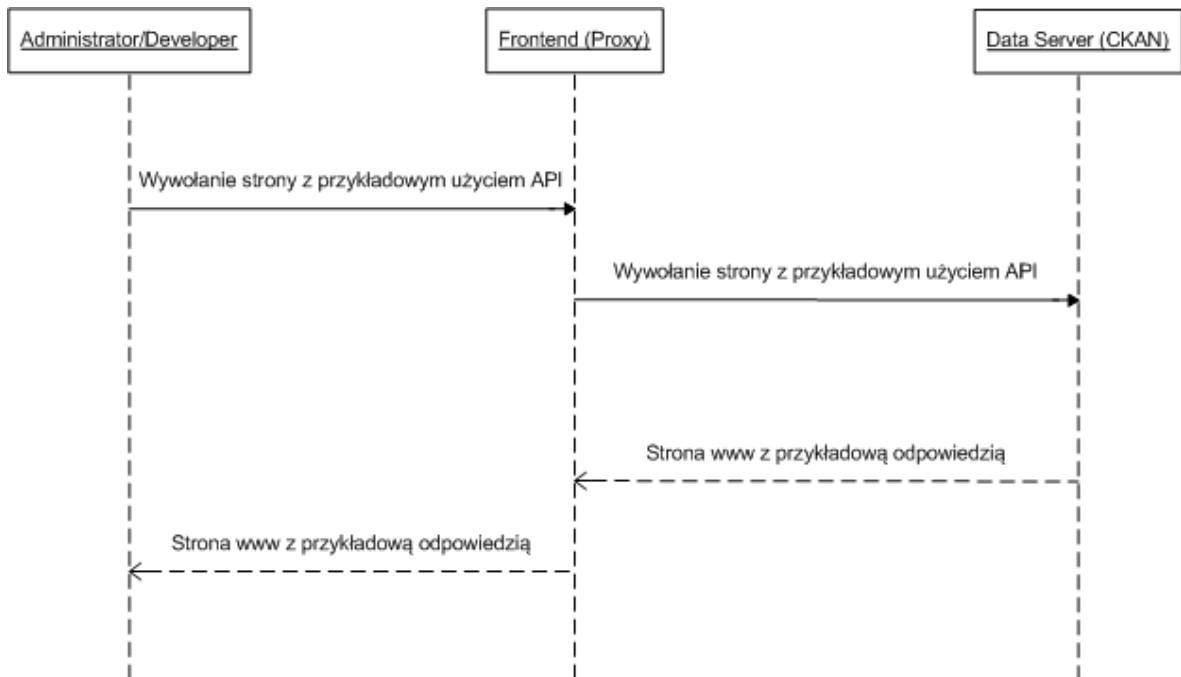
4.4.16 Zarządzanie administratorami (Function Server)



Rysunek 21 Zarządzanie użytkownikami Function Server

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Modyfikacja konta administratora (Function Server)	Administrator Frontend (Proxy) Function Server	<p>Biznesowe działanie: Administrator modyfikuje konto w Function Server</p> <p>Wejście: Wywołanie strony www parametrami konta i ich modyfikacja</p> <p>Wyjście: Strona www z parametrami konta Zmodyfikowane konto administratora</p>

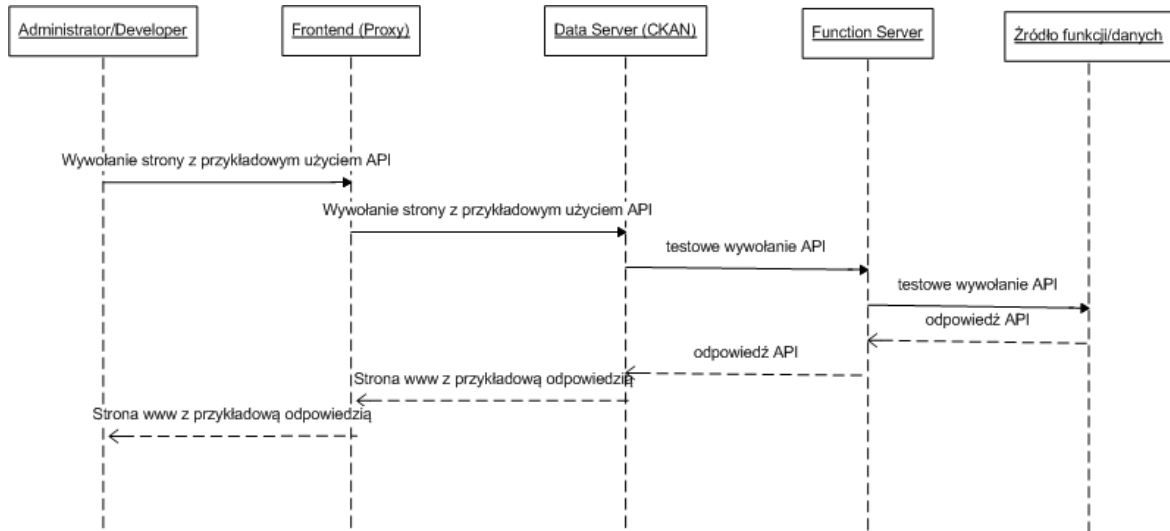
4.4.17 Testowe użycie API (Data Server - CKAN)



Rysunek 22 Testowe użycie API – (Data Server CKAN)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Testowe użycie API (Data Server - CKAN)	Administrator Developer Frontend (Proxy) Data Server (CKAN)	<p>Biznesowe działanie: Administrator lub developer wywołują stronę z www podglądem wyniku działania API do danego zbioru danych lub funkcji</p> <p>Wejście: Wywołanie strony www podglądem wyniku działania API</p> <p>Wyjście: Strona www z podglądem wyniku działania API</p>

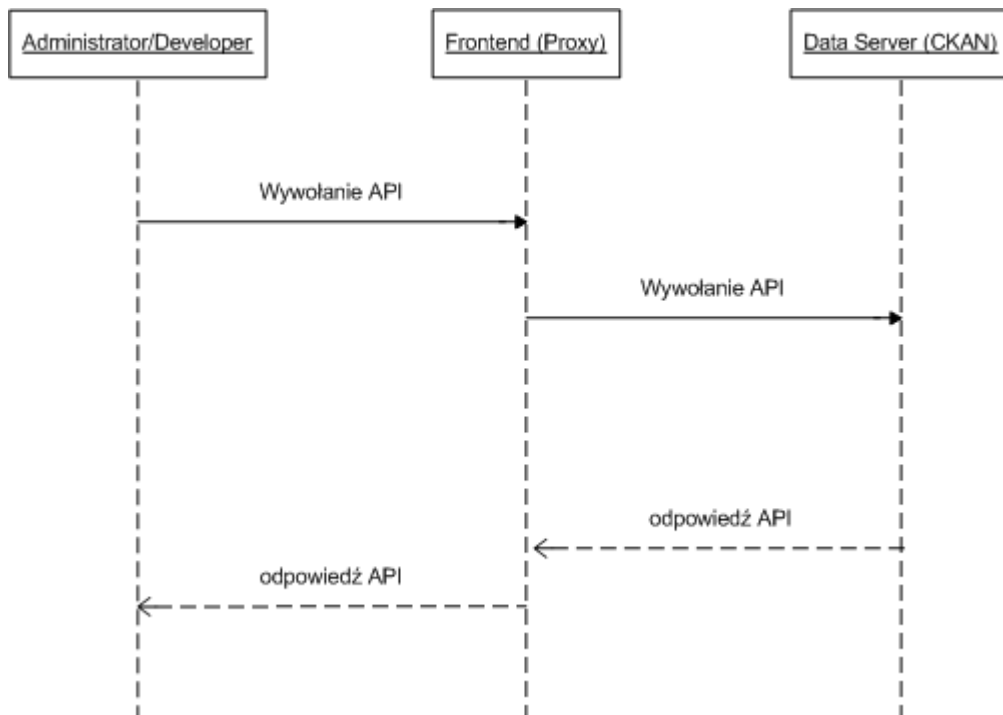
4.4.18 Testowe użycie API (Function Server)



Rysunek 23 Testowe użycie API - Function Server

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Testowe użycie API (Function Server)	Administrator Developer Frontend (Proxy) Data Server (CKAN) Function Server Źródło API/danych	<p>Biznesowe działanie: Administrator lub developer wywołują stronę z www podglądem wyniku działania API do danego zbioru danych lub funkcji</p> <p>Wejście: Wywołanie strony www podglądem wyniku działania API</p> <p>Wyjście: Strona www z podglądem wyniku działania API</p>

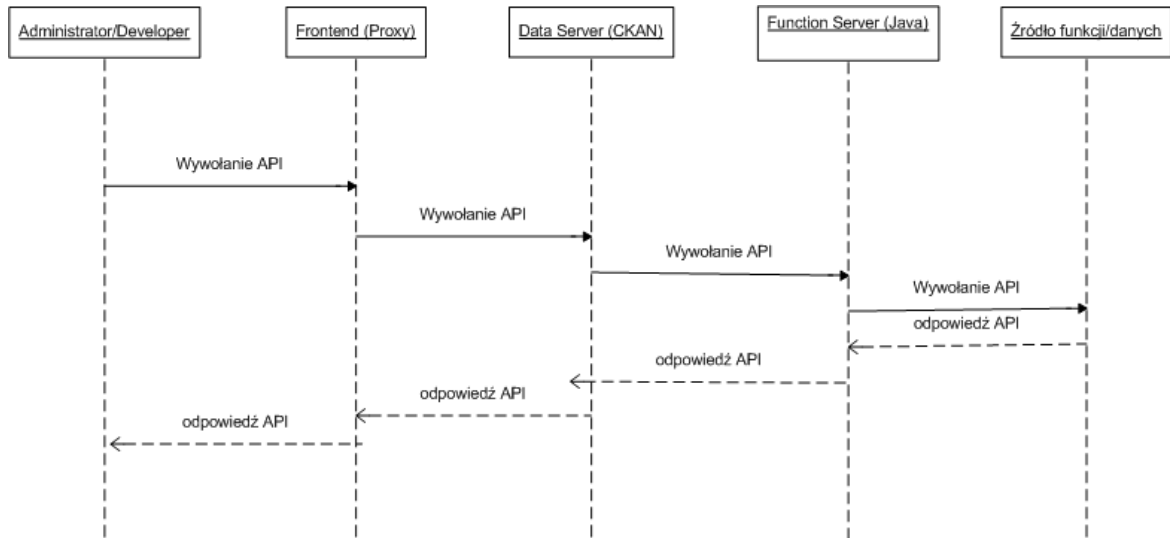
4.4.19 Użycie API (Data Server - CKAN)



Rysunek 24 Użycie API – Data Server CKAN

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Użycie API (Data Server - CKAN)	Administrator Developer Frontend (Proxy) Data Server (CKAN)	Biznesowe działanie: Administrator lub developer wywołują API do danego zbioru danych lub funkcji Wejście: Wywołanie API Wyjście: Odpowiedź - wynik działania API

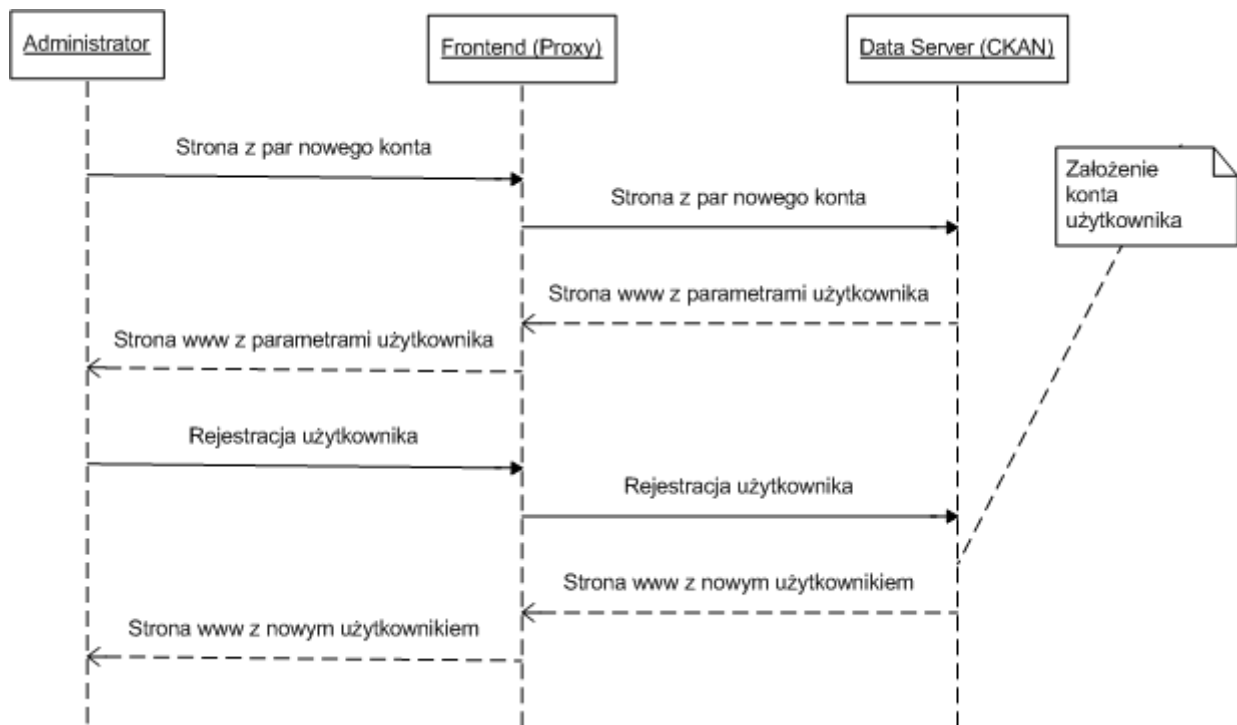
4.4.20 Użycie API (Function Server)



Rysunek 25 Użycie API- Function Server

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Użycie API (Function Server)	Administrator Developer Frontend (Proxy) Data Server (CKAN) Function Server Źródło funkcje/danych	Biznesowe działanie: Administrator lub developer wywołują API do danego zbioru danych lub funkcji Wejście: Wywołanie API Wyjście: Odpowiedź - wynik działania API

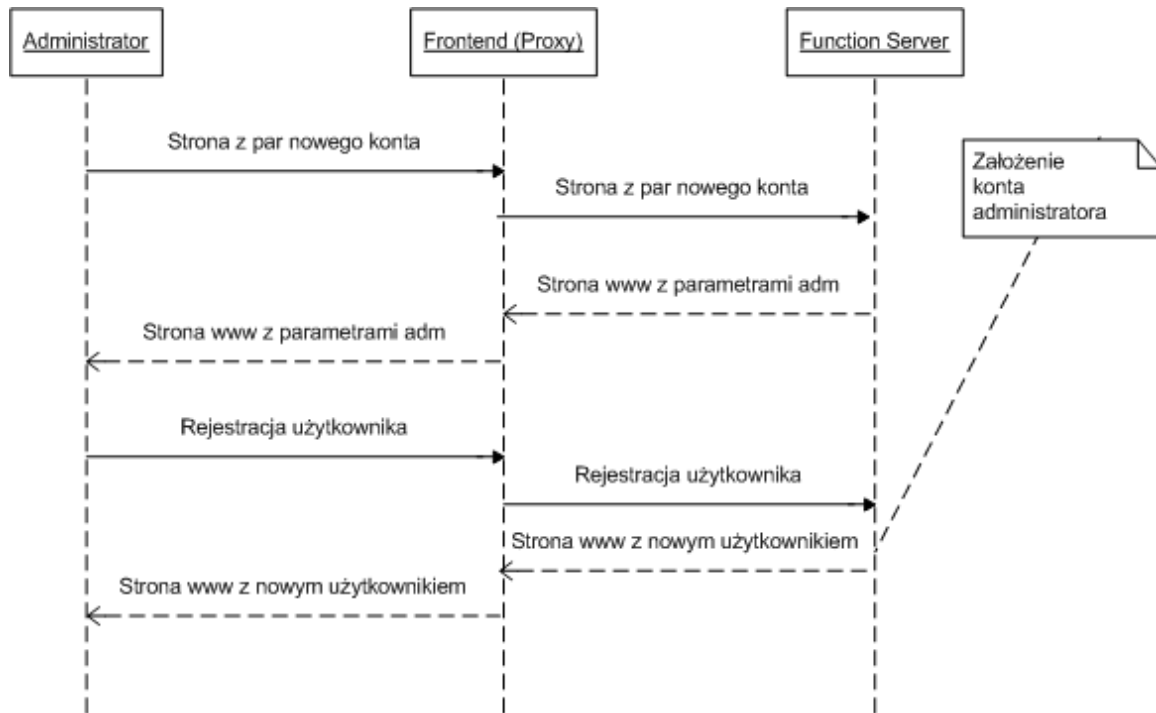
4.4.21 Rejestracja nowego użytkownika/administratora (Data Server – CKAN)



Rysunek 26 Rejestracja nowego użytkownika

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Rejestracja nowego użytkownika / administratora (Data Server - CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	<p>Biznesowe działanie: Administrator zakłada konto użytkownika</p> <p>Wejście: Wywołanie strony www z parametrami konta użytkownika i ich uzupełnienie</p> <p>Wyjście: Strona www z parametrami konta Potwierdzenie założenia konto użytkownika (uwaga nadanie praw administratora nastąpi mechanizmami CKAN – z konsoli administracyjnej)</p>

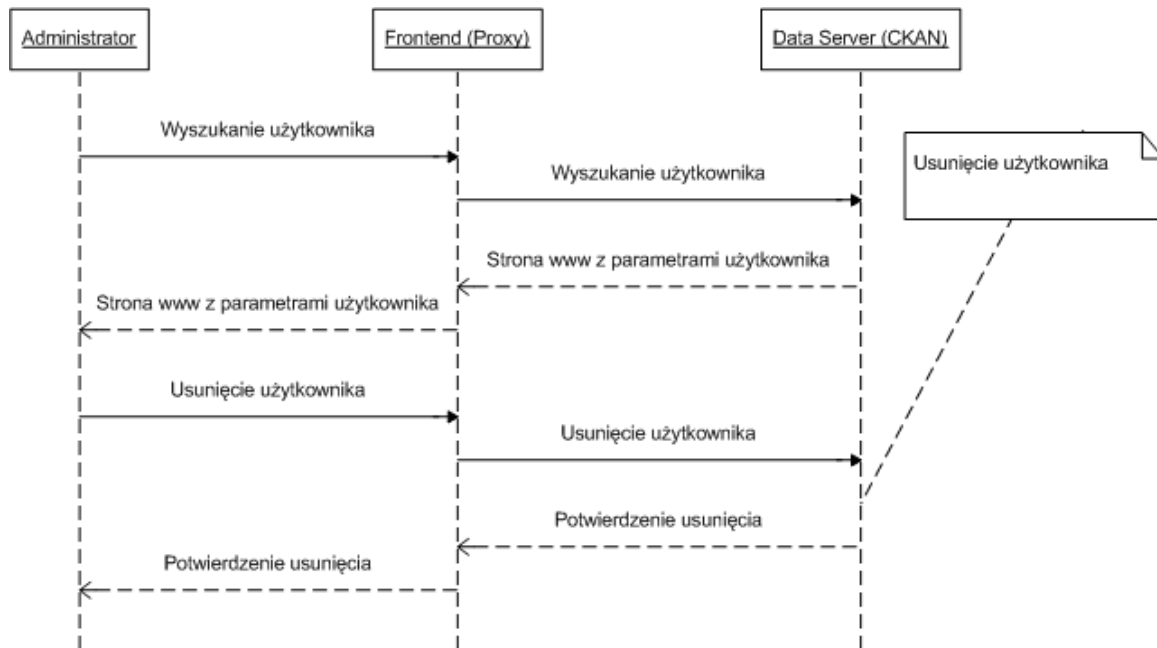
4.4.22 Rejestracja nowego administratora (Function Server)



Rysunek 27 Rejestracja nowego konta administracyjnego (Function Server)

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Rejestracja nowego administratora (Function Server)	Administrator Frontend (Proxy) Function Server (Java)	<p>Biznesowe działanie: Administrator zakłada konto dla innego administratora</p> <p>Wejście: Wywołanie strony www z parametrami konta administratora i ich uzupełnienie</p> <p>Wyjście: Strona www z parametrami konta Potwierdzenie założenia konta</p>

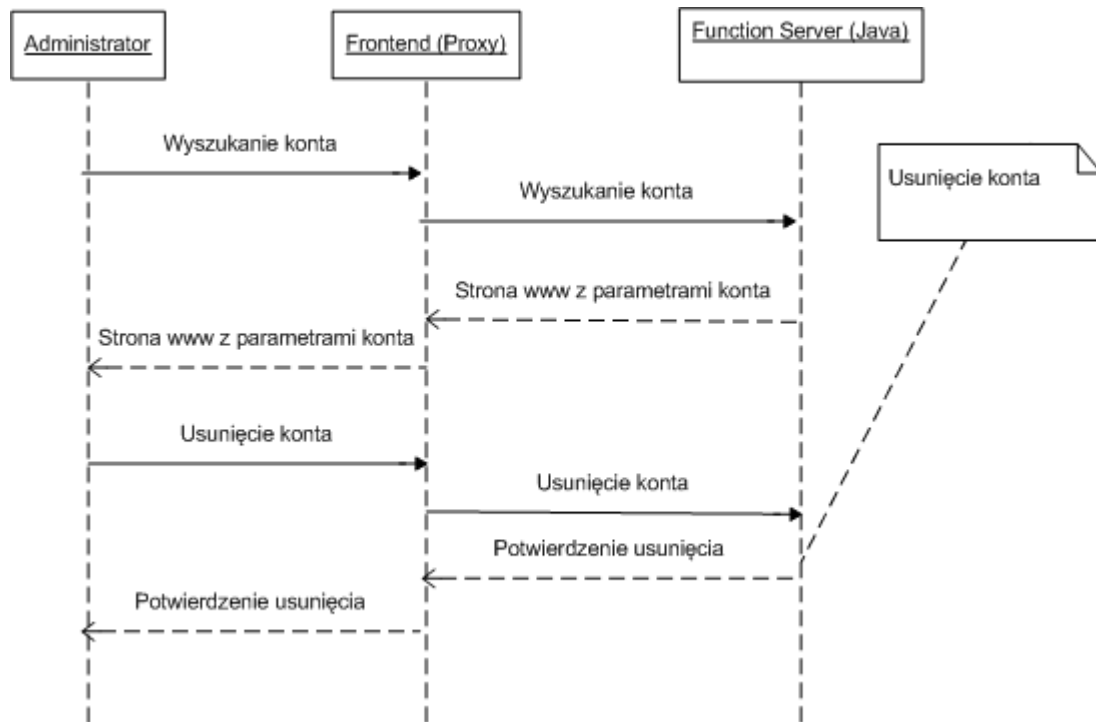
4.4.23 Usunięcie konta użytkownika /administratora (Data Server – CKAN)



Rysunek 28 Zarządzanie użytkownikami CKAN

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Usunięcie konta użytkownika / administratora (Data Server - CKAN)	Administrator Frontend (Proxy) Data Server (CKAN)	<p>Biznesowe działanie: Administrator usuwa konto użytkownika</p> <p>Wejście: Wywołanie strony www parametrami konta użytkownika i jego usunięcie</p> <p>Wyjście: Strona www z potwierdzeniem usunięcia konta</p>

4.4.24 Usunięcie konta administratora (Function Server)



Rysunek 29 Zarządzanie użytkownikami CKAN

Funkcjonalność	System	Opis (biznesowe działanie, wejście i wyjście)
Modyfikacja konta użytkownika (Function Server)	Administrator Frontend (Proxy) Function Server (Java)	Biznesowe działanie: Administrator usuwa konto Wejście: Wywołanie strony www parametrami konta administratora i jego usunięcie Wyjście: Strona www z potwierdzeniem usunięcia konta

4.5 Infrastruktura - wymagania

W poniższych tabelach poniżej przedstawiono wymagania na infrastrukturę użytą do budowy platformy MUNDO. W obu przypadkach założono użycie wirtualizacji (zastosowano VMWare w wersji 5.1 - HW version 8).

Tabela 5 Wymagania sprzętowe - platforma testowa (laboratoryjna)

VM	OS	Procesor	RAM	HD
Frontend Server (Proxy)	Linux 64 bit server (Ubuntu, Redhat, CentOS)	2 Cores	4GB	80 GB
Data Server	Linux 64 bit server (Ubuntu, CentOS)	2 Cores	4GB	80 GB

(CKAN)	Redhat)			
Function Server (Java)	Linux 64 bit server (Ubuntu, CentOS, Redhat)	2 Cores	4GB	80 GB

Tabela 6 Wymagania sprzętowe - platforma produkcyjna

VM	OS	Processor	RAM	HD
Frontend Server (Proxy)	Linux 64 bit server (Ubuntu, CentOS, Redhat)	2 Cores	8GB	80 GB
Data Server (CKAN)	Linux 64 bit server (Ubuntu, CentOS, Redhat)	2 Cores	8GB	80 GB + 100GB-1TB na dane (wielkość storage zależy od ilości danych statycznych i będzie modyfikowany w trakcie uruchamiania systemu)
Function Server (Java)	Linux 64 bit server (Ubuntu, CentOS, Redhat)	2 Cores	8GB	80 GB

5 Kwestie otwarte

LP	Kwestia			Odpowiedź			Status
	treść	autor	data	treść	autor	data	
1	Czy tylko administrator może tworzyć zestawy danych?	Jarosław Legieński	01.08.2014	Tak – tylko użytkownik z prawami administratora może zakładać nowe zbiory danych	BI UM Warszawa	09.01.2015	Z
2	Czy pozostawiamy podział na „Organizacje” w katalogu danych?	Jarosław Legieński	14.08.2014	Tak – organizacje będą odzwierciedlały strukturę organizacyjną UM Warszawa	BI UM Warszawa	09.01.2015	Z
3	Opracowanie nowych css dla CKAN lub skorzystanie z istniejących stylów	Jarosław Legieński	28.08.2014	Założono modyfikację portalu CKAN w oparciu o szablony UM Warszawa. Frontend – będzie stanowił portal miasta	BI UM Warszawa	09.01.2015	Z
4	Wyskalowanie platformy (rozdział infrastruktura) – czekamy na dane i informacje o nich	Jarosław Legieński	28.08.2014	Założono po konsultacji z UM Warszawa, że rozmiar danych nie przekroczy 1 TB	BI UM Warszawa	09.01.2015	Z
5	Czy dostęp do API wymaga założenia konta w platformie?	Jarosław Legieński	24.09.2014	Większość danych udostępniamy bez autoryzacji wybrane API (np. 19115) będą wymagały zalogowania i posiadania APIKEY (opcja zostanie ustawiony limit wywołań dla użytkowników anonimowych ewentualnie limity ulegną zwiększeniu dla użytkowników zalogowanych – funkcja będzie realizowana przez Frontend BI UM Warszawa	BI UM Warszawa	09.01.2015	Z

6 Słowniczek – lista skrótów

Glossary		
Nb.	Abbreviation	Explanation
1	API	Application Programming Interface
2	GPL	GNU Public License – Licencja wolnego i otwartego oprogramowania
3	LGPL	Lesser GNU Public License - słabsza powszechna licencja publiczna GNU, powszechna licencja publiczna dla bibliotek
4	FDL	Free Documentation License - Licencja Wolnej Dokumentacji
5	CC	Creative Commons
6	CKAN	Comprehensive Knowledge Archive Network – sieć archiwizacyjna dla kompleksowej wiedzy – system www będący repozytorium danych i eksponujący dane otwarte stworzony przez Open Knowledge Foundation
7	WS	Web Service – usługa sieciowa oparta o SOA/SOAP lub ROA/REST
8	SOA	Architektura zorientowana na usługi (ang. Service Oriented Architecture)
9	SOAP	SOAP (ang.) Simple Object Access Protocol – protokół wywoływania zdalnego dostępu do obiektów oparty o XML
10	ROA	Architektura oparta o zasoby (ang. resource-oriented architecture)
11	REST	Representational State Transfer – wzorzec architektury oprogramowania
12	MD	metadane (dane o danych)
13	WMS	Web Map Service - stworzony przez Open Geospatial Consortium (OGC) standard udostępniania map w postaci rastrowej za pomocą interfejsu http
14	WFS	Web Feature Service stworzony przez Open Geospatial Consortium (OGC) standard udostępniania map w postaci wektorowej za pomocą interfejsu http
15	DS.	DS – data set – zbiór danych (przykład – mapy rastrowe i wektorowe z serwisów WMS/WFS Miasta)
16	FS	FS - function set - zbiór funkcji (np. realizowanych przez web service) – przykład wysłanie zgłoszenia do UM Warszawa poprzez 19115
17	APIKey	Kod alfanumeryczny przekazanego do serwera API przez programy komputerowe wywołujące API
18	RDF	Ang. Resource Description Framework - język/metoda pozwalająca na opisywanie zasobów sieci Web, ze składnią opartą na XML, opracowana przez W3C
19	Dublin Core	Dublin Core Metadata Element Set, DC, DCEs) – ogólny standard metadanych. Przyjęty, jako standard ISO 15836-2003.
20	DCAT	W3C Data Catalog Vocabulary - standard schematu katalogowego dla danych
21	CSV	Comma separated values – plik z wartościami oddzielanymi przecinkami
22	CMS	System zarządzania treścią (content management system)
23	SPARQL	SPARQL Protocol And RDF Query Language - język zapytań i protokół dla

		plików RDF.
24	OWL	Web Ontology Language – język ze składnią opartą na XML, a semantyką opartą na logice opisowej
25	HTTP	Hypertext Transfer Protocol
26	SSL	Secure Socket Layer

7 Bibliografia

[1] Wniosek do 1 konkursu Innowacje Społeczne NCBIR projektu MUNDO, Warszawa, 2013

[2] Portal CKAN <http://ckan.org/> [30.10.2014]

[3] Portal The Open Knowledge Foundation <https://okfn.org/> [30.10.2014]