Załącznik nr 1 do Umowy

System Gromadzenia i Przetwarzania Danych Miejskich Kraków

Urząd Miasta Krakowa

Wersja 1.0
Data 14.09.2023 r.

Spis treści

Definicje i skróty

Active Directory - usługa katalogowa wdrożona w SI UMK.

AI – sztuczna inteligencja.

COI – Centrum Obsługi Informatycznej Urzędu Miasta Krakowa.

Data Lakehouse – architektura zarządzania danymi.

GMK – Gmina Miejska Kraków.

GUI – graficzny interfejs Użytkownika.

HA – wysoka dostępność.

IoT – Internet rzeczy.

LDAP - protokół pozwalający na wymianę informacji wykorzystując protokół TCP/IP.

ML – uczenie maszynowe.

Platforma Sprzętowa – sprzęt komputerowy, na którym jest zainstalowany System i Oprogramowanie Systemowe niezbędne do działania Systemu.

1. Preambuła

Podstawą efektywnego zarządzania miastem jest wiedza, a jej źródłem są dane, które pochodzą z wielu różnych źródeł. Istotne w tym kontekście jest posiadanie informacji na temat dostawcy, zakresu, a także możliwości integracji poszczególnych elementów. Nie mając pełnej informacji stworzenie przestrzeni gromadzącej w sobie dane z dostępnych źródeł może być co najmniej utrudnione.

Urząd Miasta Krakowa wskazuje, iż w jego obrębie działa ponad 200 systemów dziedzinowych, a zakres danych przez nie gromadzonych obejmuje większość obszarów zarządzania miastem. Liczba ta nie obejmuje systemów utrzymywanych przez miejskie jednostki organizacyjne oraz spółki miejskie, które dysponują niezwykle wartościowymi z punktu widzenia wdrażania rozwiązań Smart City danymi. Dane te powinny zostać zintegrowane i przetwarzane w ramach wspólnego systemu analitycznego.

Z uwagi na wartości dodane, jakie może dać ponowne wykorzystanie posiadanych przez miasto danych oraz ich wzajemną korelację prowadzone są działania mające na celu ich integrację i standaryzację. Dlatego w ostatnim czasie w mieście Kraków zrealizowano projekty, takie jak m.in. Platforma Wirtualizacji Danych, Centralny System Słowników czy Referencyjna Baza Osób, które mają na celu wprowadzenie spójnych standardów wymiany danych pomiędzy systemami.

Miasto Kraków, chcąc utrzymywać i tworzyć rozwiązania cechujące miasto inteligentne, zauważa konieczność stosowania rozwiązań pozwalających na zarządzanie danymi hybrydowymi klasy Data Lakehouse, dzięki czemu możliwe będzie gromadzenie danych m.in. z urządzeń IoT, bez konieczności ich wcześniejszego przetwarzania i zapewniania odpowiedniej struktury danych.

Podstawowym wyzwaniem jest brak zautomatyzowanych procesów umożliwiających agregację i analizowanie w sposób zautomatyzowany danych pochodzących z licznych źródeł, wykluczając przy tym konieczność manualnej analizy, która często stanowi źródło błędów. Kraków - miasto inteligentne dąży w kierunku modelu wykorzystującego nową generację technologii informacyjnych, w celu sprawniejszego planowania, budowy, zarządzania i oferowania inteligentnych usług publicznych. Jednym z celów transformacji w miasto inteligentne jest przeciwstawienie się silosowemu podejściu do zarządzania danymi. Smart City wymaga kompleksowej przebudowy struktur miejskich i sposobu organizacji miasta. Miasto wymaga skoordynowania podejmowanych działań we wszystkich obszarach poprzez adekwatne i celowe wykorzystanie danych, przy współpracy i zaangażowaniu różnych grup interesariuszy.

Podstawowym odbiorcą efektów wdrożenia rozwiązania będzie Urząd Miasta Krakowa wraz z jednostkami podległymi lub wybranymi przez siebie podmiotami, jednak w dalszej perspektywie efekty oddziaływać będą pozytywnie także na pozostałe podmioty otoczenia i samych mieszkańców.

Zaczynając proces mający na celu zlikwidowanie problemów objętych wyzwaniem konieczne jest w pierwszym etapie przeprowadzenie inwentaryzacji systemów. W ramach inwentaryzacji należy zebrać informacje nie tylko o samych systemach, ale w głównej mierze o zbiorach danych, które te systemy reprezentują i gromadzą, a następnie wyselekcjonować te zbiory, które znajdą się w projektowanym rozwiązaniu docelowym. Zadaniem towarzyszącym inwentaryzacji systemów i zbiorów danych będzie również inwentaryzacja problemów jakie użytkownicy poszczególnych systemów mają z posiadanymi danymi (np. redundantność, nieaktualność, nieadekwatne okresy aktualizacji, itp.). Problemy należy skategoryzować tak, aby ich rozwiązywanie realizować według uzgodnionych priorytetów.

Późniejsze gromadzenie danych, pozwalających na ich analizowanie w różnych aspektach, stanowić będzie jeden z głównych filarów zarządzania miastem oraz umożliwi optymalizację usług publicznych. Wysoka jakość wprowadzanych danych pozwoli także na dokonywanie rzetelnych analiz i uzyskiwanie poprawnych efektów.

Po przeprowadzeniu analizy i wyselekcjonowaniu właściwych zbiorów danych, zostanie wdrożone stosowne rozwiązania technologiczne oraz podłączone zostaną wyselekcjonowane źródła danych.

Na tak przygotowanym środowisku zdefiniowane zostaną i uruchomione mechanizmy analityczne, pozwalające zrealizować (rozwiązać/usunąć) zinwentaryzowane i skategoryzowane problemy. W tym etapie również ważne jest zbudowanie kompetencji w Zespole Analityków Danych COI pozwalające na dalsze rozwijanie Data Lakehouse przez podłączanie kolejnych źródeł i wykonywanie (zasobami własnymi lub poprzez outsourcing kolejnych analiz i raportów).

Zakładając, że docelowo działania dotyczyć mają całości zasobu danych jakimi dysponuje miasto w dużej liczbie różnych rozproszonych systemów, efekty wdrożenia rozwiązania realizującego wyzwanie dotyczyć będą równie szerokich zakresów funkcjonowania miasta. Najistotniejsze jest, żeby realizowane były najważniejsze potrzeby jednostki i mieszkańców, a te wskazane są w głównym dokumencie strategicznym jakim jest wspomniana już w niniejszym raporcie „Strategia Rozwoju Krakowa. Tu chcę żyć. Kraków 2030.". Planuje się więc, żeby efekty podjętych działań oddziaływały na poszczególne obszary życia miasta i pozwalały na budowanie i wzmacnianie struktur miasta inteligentnego, w którym:

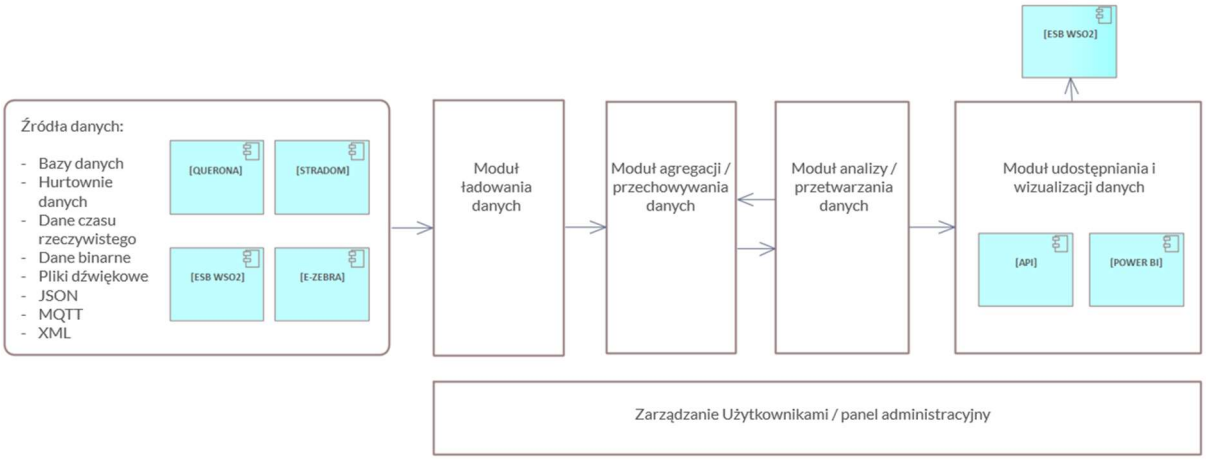
- „świadomość" danych pozwala na wzmocnienie współpracy ze środowiskami nauki i biznesu,
- podejmowanie decyzji oparte jest o wiarygodne, aktualne dane,
- miasto jest „odporne" na zmiany, z uwagi na dostępną wiedzę i możliwość szybkiego reagowania.

Projekt posiada wiele walorów innowacyjnych i jest zbieżny z obecnymi trendami technologicznymi stosowanymi na całym świecie. Do najistotniejszych elementów innowacyjnych zaliczyć można: szybkość w gromadzeniu i przetwarzaniu zarówno ustrukturyzowanych, jak i nieustrukturyzowanych zbiorów danych; możliwość przeprowadzania skomplikowanych analiz bez konieczności zamawiania analiz u dostawców komercyjnych; przejrzystość Systemu pozwalająca na korzystanie z niego zarówno Użytkowników technicznych, jak i tych o niższym stopniu zaawansowania technicznego.

W perspektywie lat kolejnych miasto Kraków planuje uruchomić szereg projektów, które będą dostarczały dużych ilości danych w różnym formacie. Przykładowymi realizacjami będą m.in. monitoring uliczny z wykorzystaniem inteligentnych kamer; sieć urządzeń IoT do pomiarów miejskiego środowiska w tym: czujniki ruchu, czujniki jakości powietrza, czujniki wilgotności i temperatury, mierniki hałasu i wibracji; dane z mediów społecznościowych, próbki dźwięku, tekst pisany w języku naturalnym, dane przestrzenne i wiele innych, które pozwolą na danetyzację miasta. Wiele z tych danych będzie reprezentowana w różnych formatach, dla przykładu: csv, xml, xls, json, parquet, kml, gml, prd, txt, geojson, zip, shp. Mając na uwadze różnorodność danych, miastu Kraków zależy, aby wdrażany System charakteryzował się otwartością na wszelkiego rodzaju formaty danych.

1.1. Przykładowa architektura Systemu

Na poniższym schemacie przedstawiono przykładową architekturę Systemu z wyróżnieniem obszarów funkcjonalnych wskazanych w poniższym opisie funkcjonalności.



Powyższa architektura została przedstawiona jako przykład. System może zostać wdrożony w oparciu o architekturę Wykonawcy zaproponowaną w pracy konkursowej i zaakceptowaną przez Zamawiającego.

2. Opis funkcjonalności/charakterystyka

Projektowany System będzie realizować zadania dotyczące przechowywania, agregowania, systematyzowania, porządkowania i raportowania danych miejskich. Dane miejskie to wszystkie dane zbierane z systemów i urządzeń działających na terenie miasta Krakowa.

System będzie głównym źródłem danych na potrzeby raportowania zarządczego, procesowego, analitycznego i statystycznego. Umożliwi także prowadzenie analiz wielowątkowych w oparciu o dane zawarte w powstałej hurtowni danych, w celu lepszego zarządzania miastem.

Ze względu na potrzebę zbierania i przechowywania nieprzetworzonych (surowych), w formacie źródłowym, danych z wielu źródeł, oczekiwany System będzie miał charakter Data Lakehouse.

Analiza będzie oparta na eksploracji dowolnego rodzaju danych, na przykład:

- eksploracji tekstu,
- eksploracji danych,
- analizie statystycznej,
- analizie grafów.

Zamawiający oczekuje, że System będzie oferował najlepsze sposoby na:

- przyjmowanie i transformację, czyli przenoszenie i konwersję różnych rodzajów i formatów danych,
- utrwalanie i dostęp, czyli zapewnienie bezpieczeństwa i odkrywalności danych, jednocześnie łatwe skalowanie oraz dostępność dla wszystkich produktów zgodnie z zapotrzebowaniem, wraz z przypisaniem ról do Użytkowników,
- analizowanie i wykorzystanie badania danych, czyli narzędzia do odkrywania trendów w danych zarejestrowanych w Systemie, z wykorzystaniem ML oraz AI,
- skalowalność rozwiązania, czyli zaprojektowanie z myślą o skalowaniu pionowym i poziomym z uwzględnieniem możliwości sprzętowych Zamawiającego, bez znaczącego pogorszenia wydajności,
- udostępnianie danych do systemów zewnętrznych (sterowanie, prezentacja).

3. Wymagania

3.1. Podstawowe wymagania dla Systemu

- Ma zostać zainstalowany na serwerach Zamawiającego i ma posiadać możliwość skalowania pionowego i poziomego, tak jak w przypadku rozwiązań chmurowych.
- Ma posiadać możliwość przechowywania plików różnego typu (wskazanych w 3.2.1. oraz zidentyfikowanych na etapie Analizy Przedwdrożeniowej) i rozmiaru.
- Ma działać w oparciu o model ETL (Extract – Transform- Load) czyli ekstrakcję danych ze źródeł zewnętrznych, czyszczenie/przekształcenie/analizę i ładowanie do obiektów czy innych systemów, które wykorzystują zbiory danych.
- Ma zapewniać możliwość tworzenia modeli danych.
- Ma posiadać wbudowane narzędzia do:
    - Analizy danych oparte o ML, AI, w tym:
        - Ładowanie danych do modeli ML.
        - Trenowanie maszyn.
    - Pozyskiwania danych ze źródeł zewnętrznych.
    - Przetwarzania danych.
    - Tworzenia modeli danych.
    - Agregacji danych w oparciu o ustalone scenariusze.
- Ma zapewniać możliwość swobodnej migracji danych, w przypadku decyzji o zmianie dostawcy.
- Ma być wyposażony w panel administracyjny do zarządzania tożsamością, autoryzacją i autentyfikacją Użytkowników.
- Ma zapewniać możliwość pełnej integracji z rozwiązaniami chmurowymi Microsoft Azure, Google GCP, Amazon AWS.
- Ma mieć intuicyjny i spójny interfejs graficzny, zgodny z ogólnie przyjętymi standardami i zasadami projektowania.
- Ma zapewniać łatwy dostęp do najczęściej używanych funkcji i danych, np. poprzez menu, paski narzędzi, skróty klawiszowe, wyszukiwarkę lub zakładki.
- Ma umożliwiać personalizację ustawień i preferencji Użytkownika, np. poprzez zmianę kolorów, czcionek, kontrastu układu okien lub języka.
- Ma oferować pomoc i wsparcie Użytkownikowi, np. poprzez podpowiedzi, instrukcje, dokumentację, FAQ lub czat.
- Ma być responsywny i szybki w działaniu, minimalizując czas oczekiwania i opóźnienia.
- Ma zapewniać bezpieczeństwo i prywatność danych Użytkownika, np. poprzez szyfrowanie, uwierzytelnianie, autoryzację lub audyt.
- Ma mieć możliwość integracji i być kompatybilny z innymi systemami i aplikacjami używanymi przez Użytkownika, np. poprzez API, import/export danych lub formaty plików.
- Ma być elastyczny i skalowalny, umożliwiając dostosowanie się do zmieniających się potrzeb i wymagań Użytkownika, np. poprzez dodawanie/usuwanie funkcji lub modułów, zwiększanie/zmniejszanie pojemności lub zasobów.
- Ma być niezawodny i stabilny w działaniu, minimalizując ryzyko awarii, błędów lub utraty danych.

3.2. Podział na moduły

System musi składać się z co najmniej 5 części, które są ze sobą zintegrowane i pozwalają na swobodną wymianę danych:

- Moduł ładowania danych.
- Moduł agregacji/przechowywania danych.
- Moduł analizy/przetwarzania danych.
- Moduł udostępniania i wizualizacji danych.
- Zarządzanie Użytkownikami i Panel administracyjny.

3.2.1 Moduł ładowania danych

Moduł, który odpowiada za pobieranie danych z różnych źródeł i przesyłanie ich do Systemu. Cechy modułu:

- Obsługa różnych protokołów i formatów danych, takich jak: HTTP, MQTT, JSON, XML, CSV, XLS, JSON, PARQUET, KML, GML, PRD, TXT, GEOJSON, ZIP, SHP itp.
- Walidacja i weryfikacja poprawności i kompletności danych wejściowych.
- Transformacja i normalizacja danych do wspólnego modelu i standardu.
- Filtrowanie i odrzucanie niepotrzebnych lub nieprawidłowych danych.
- Szyfrowanie i uwierzytelnianie danych dla zapewnienia bezpieczeństwa i prywatności.
- Buforowanie i kolejkowanie danych w przypadku awarii lub przeciążenia sieci.
- Monitorowanie i raportowanie stanu i wydajności procesu ładowania danych.
- Reagowanie na zdarzenia i alarmy związane z procesem ładowania danych.
- Automatyczne lub ręczne uruchamianie lub zatrzymywanie procesu ładowania danych.
- Konfigurowanie i zarządzanie źródłami danych i parametrami procesu ładowania danych.

3.2.2 Moduł agregacji/przechowywania danych

Moduł, który odpowiada za przechowywanie danych w odpowiednich bazach lub magazynach danych. Cechy modułu:

- Zapewnienie możliwości przechowywania danych ustrukturyzowanych: bazy relacyjne, hurtownie danych zgodnie z posiadanymi technologiami Zamawiającego.
- Zapewnienie możliwości agregacji oraz deagregacji danych.
- Zapewnienie możliwości przechowywania danych nieustrukturyzowanych – obrazy, pliki audio, pliki wideo, pliki skompresowane, dane z urządzeń IoT i inne.
- Zapewnienie wysokiej dostępności, niezawodności i skalowalności repozytorium danych poprzez stosowanie technik replikacji, partycjonowania, klastrowania, równoważenia obciążenia.
- Optymalizacja wydajności i kosztów repozytorium danych poprzez stosowanie technik indeksowania, kompresji, archiwizacji, czyszczenia.
- Zapewnienie bezpieczeństwa i prywatności repozytorium danych poprzez stosowanie technik szyfrowania, uwierzytelniania, autoryzacji, audytu itp.
- Zapewnienie jakości i spójności repozytorium danych poprzez stosowanie technik kontroli jakości, deduplikacji, konsolidacji, integracji itp.
- Zapewnienie aktualności repozytorium danych poprzez stosowanie technik synchronizacji, strumieniowania, propagacji zmian itp.
- Zapewnienie elastyczności i adaptowalności repozytorium danych poprzez stosowanie technik schematu na żądanie, schematu ewolucyjnego, migracji danych itp.
- Monitorowanie i raportowanie stanu i wydajności repozytorium danych.
- Reagowanie na zdarzenia i alarmy związane z repozytorium danych.
- Konfigurowanie i zarządzanie repozytorium danych i parametrami przechowywania danych.
- Musi posiadać wbudowany mechanizm tworzenia i harmonogramowania kopii migawkowych.
- Musi posiadać mechanizm zapewniający najwyższy poziom bezpieczeństwa i ochrony danych.
- Musi posiadać wbudowane mechanizmy umożliwiające różnicowanie danych, tak zwane warstwowanie danych. Musi mieć możliwość tworzenia polityk opartych na czasie ostatniej modyfikacji, czasie ostatniego dostępu do danych oraz ich rozmiarze lub właścicielowi bądź grupie. Musi mieć możliwość tworzenia polityk wykorzystujących wszystkie bądź tylko niektóre z wyżej wymienionych atrybutów. Musi mieć możliwość tworzenia warstwy archiwalnej w chmurze.

3.2.3. Moduł analizy/przetwarzania danych

Moduł, który odpowiada za analizę i przetwarzanie danych w celu wydobycia informacji i wiedzy. Cechy modułu:

- Zapewnienie wysokiej jakości i wiarygodności wyników analizy lub przetwarzania danych poprzez stosowanie technik walidacji, weryfikacji, testowania, oceny itp.

- Zapewnienie wysokiej wydajności i skalowalności procesu analizy lub przetwarzania danych poprzez stosowanie technik równoległego, rozproszonego lub strumieniowego przetwarzania danych.
- Zapewnienie bezpieczeństwa i prywatności procesu i wyników analizy lub przetwarzania danych poprzez stosowanie technik anonimizacji, pseudonimizacji, maskowania itp.
- Zapewnienie aktualności i świeżości wyników analizy lub przetwarzania danych poprzez stosowanie technik automatyzacji, planowania, wyzwalania itp.
- Zapewnienie funkcjonalności monitorowania i raportowania stanu i wydajności procesu i wyników analizy lub przetwarzania danych.
- Zapewnienie możliwości reagowania na zdarzenia i alarmy związane z procesem i wynikami analizy lub przetwarzania danych.
- Musi mieć możliwość pełnego skalowania pionowego i poziomego wykorzystując dowolną technologię wirtualizacji lub konteneryzacji.
- Musi posiadać możliwość integracji klastrów kontenerowych z Active Directory lub LDAP.
- Musi posiadać wbudowany w GUI dostęp terminalowy umożliwiający zarządzanie środowiskiem przetwarzania.
- Musi posiadać zintegrowaną z GUI automatyczną możliwość uruchamiania aplikacji przykładowo: Jupyter Notebook, MLflow, Kubeflow, Spark, TensorFlow.
- Musi zapewniać zintegrowany z GUI dostęp do repozytorium projektów, który umożliwia ich przeglądanie, wgrywanie na serwer, pobieranie plików.

3.2.4. Moduł udostępniania i wizualizacji danych

Moduł, który odpowiada za udostępnianie i wizualizację danych dla użytkowników końcowych. Cechy modułu:

- Zapewnienie wysokiej jakości i użyteczności interfejsu i treści udostępniania i wizualizacji danych poprzez stosowanie technik projektowania użytkownika (UX), projektowania graficznego (UI), projektowania informacji (ID) itp.
- Zapewnienie wysokiej wydajności i skalowalności interfejsu i treści udostępniania i wizualizacji danych poprzez stosowanie technik optymalizacji ładowania stron (WPO), optymalizacji silnika wyszukiwania (SEO), optymalizacji konwersji (CRO) itp.
- Zapewnienie możliwości udostępniania danych systemom zewnętrznym poprzez wbudowany moduł do budowania interfejsów integracyjnych API.
- Zapewnienie bezpieczeństwa i prywatności interfejsu i treści udostępniania i wizualizacji danych poprzez stosowanie technik szyfrowania, uwierzytelniania, autoryzacji, audytu itp.
- Zapewnienie aktualności i świeżości interfejsu i treści udostępniania i wizualizacji danych poprzez stosowanie technik automatyzacji, planowania, wyzwalania itp.

3.2.5 Zarządzanie Użytkownikami i panel administracyjny

System ma być zintegrowany z Active Directory w celu i zakresie służącym zapewnieniu funkcji tzw. jednokrotnego logowania SSO Single Sign-On (zalogowanie do komputera umożliwia dostęp do Systemu bez konieczności logowania do Systemu).

Wykonawca zapewnia dodatkowo mechanizm budowy lokalnej bazy Użytkowników i logowania kontem lokalnym do Systemu. W tym przypadku konieczne jest zastosowanie logowania dwuskładnikowego oraz wprowadzenie parametru, który pozwoli na określenie liczby nieudanych prób logowań, po których będzie blokowany dostępu po do Systemu. System musi pozwalać na ręczną zmianę lokalnego hasła przez Użytkownika.

Konta Użytkowników wewnętrznych, którzy nie zalogowali się do Systemu w ciągu 92 dni muszą być automatycznie blokowane w Systemie i otrzymywać status „brak aktywności przez 92 dni”. Blokada musi uniemożliwiać zalogowanie się do Systemu Użytkowników lokalnych oraz pochodzących z Active Directory. Administrator Techniczny musi mieć możliwość oznaczenia konta jako konto techniczne, konta tego typu i otrzymują status „konto techniczne” i nie mogą zostać automatycznie zablokowane.

System musi zawierać odseparowane funkcje administracyjne (moduł zarządzania uprawnieniami, logi i konfiguracji, szablony raportów) od funkcji związanych z pracą merytoryczną w Systemie. Moduł ten nie może być udostępniany w Internecie. Funkcjonalności modułu:

a) zarządzanie Użytkownikami:
   - tworzenie/przeglądanie/modyfikowanie/dezaktywacja kont;
   - przypisanie Użytkownika do wybranych ról;
   - modyfikacja daty ważności konta – automatyczna dezaktywacja po upłynięciu terminu;
   - możliwość odblokowania konta Użytkownika;
   - możliwość oznaczenia konta statusem „konto techniczne";
b) logi i raporty dla AT:
   - Tworzenie rejestrów oraz generowanie raportów, o których mowa w rozdziale 4. Monitorowanie Użytkowników (logi);
   - dostęp do logów, o których mowa w rozdziale 4. Monitorowanie Użytkowników (logi).

3.3. Pozostałe wymagania dla rozwiązania

1. Rozwiązanie ma zapewnić:
   a) wieloklastrową, współdzieloną architekturę danych obejmującą trzy warstwy, które będą logicznie zintegrowane, ale będą mogły skalować się niezależnie od siebie:
      - przechowywania - jedno miejsce dla wszystkich danych ustrukturyzowanych, częściowo ustrukturyzowanych i nieustrukturyzowanych,
      - obliczania - niezależne zasoby obliczeniowe dostosowane do bieżącego obciążenia, aby wyeliminować współzawodnictwo o zasoby,
      - usług - warstwa usług, która obsługuje co najmniej infrastrukturę, bezpieczeństwo, metadane, optymalizację zapytań,
   b) spójny zestaw narzędzi, które pokażą, co znajduje się w Systemie, w tym:
      - pochodzenie danych,
      - wykorzystanie danych,
   c) wsparcie szerokiego zakresu wykorzystania i współdzielenia danych na zasadzie praw dostępu, w tym zarządzanie dostępem w oparciu o role w Systemie;
   d) zasilanie przez wiele potoków danych, z których każdy dostarcza dane z inną częstotliwością, bez nakładania na nie ograniczeń (przykładowo: przesyłanie strumieniowe end-to-end, wsadowe, pipeline, API);
   e) łatwe udostępnianie danych uprawnionym Użytkownikom bez konieczności kopiowania tych danych przez administratorów baz;
   f) możliwość tworzenia jednej dynamicznej kopii danych, która może wypełniać i aktualizować modele ML (machine learning), pulpity BI (Business Intelligence) oraz aplikacje do analizy predykcyjnej, która umożliwia także orkiestrację analityki, współdzielenie danych, pobieranie danych i naukę o danych;
   g) możliwość tworzenia schematów, algorytmów, przepływów danych metodą low-code;
   h) możliwość zautomatyzowanego zasilania Systemu danymi rzeczywistymi z poszczególnych źródeł w sposób zautomatyzowany;
2. Rozwiązanie ma zawierać funkcjonalność pozwalająca na katalogowanie danych, wyszukiwanie zbiorów danych, zarządzanie katalogiem danych, możliwość dodawania opisów do zbiorów, możliwość dodawania metadanych do zbiorów danych, możliwość wyszukiwania zbiorów po metadanych z wykorzystaniem centralnej wyszukiwarki.
3. Rozwiązanie ma w jak największym stopniu zostać utworzone w oparciu o popularne i szeroko wspierane technologie open source. W ramach rozwiązań open source Zamawiający oczekuje wsparcia od Wykonawcy. W przypadku braku możliwości zastosowania ww. technologii lub braku ich zasadności dopuszczane jest wykorzystanie technologii licencjonowanych (pod warunkiem uzyskania zgody Zamawiającego w zakresie warunków licencyjnych oprogramowania i jego kosztów).

4. W pierwszej fazie źródłami danych dla Systemu będą:
    - dane zgodnie z wymaganiami dot. STRADOM opisanymi w Załączniku nr 1 do niniejszego Załącznika „Wymagania funkcjonalne dla Systemu z punktu widzenia STRADOM ”,
    - dane o zużyciu mediów (pliki xls, csv) stanowiące Załącznik nr 2 do niniejszego Załącznika „Podsumowanie założeń do projektu e-ZEBRA”. Dane o zużyciu mediów to łącznie około 3 GB danych rocznie. Dodatkowo dane z 20 falowników wyposażonych w API – produkcja z instalacji fotowoltaicznych.

    Oczekiwane jest zaprojektowanie Systemu w taki sposób, aby w przyszłości była możliwość przyjmowania danych z systemów zarządzania budynkami (BMS) opartych o system SCADA, a także możliwość wysyłania informacji zwrotnych – perspektywa 2024-2025. W latach kolejnych, szacowany wolumen to około 50 GB rocznie.
5. System ma posiadać możliwość pozyskiwania i analizy danych z aplikacji QUERONA oraz webservice'ów na szynie WSO2. W ramach platformy WSO2 posiadamy instancje produkcyjne i testowe środowiska. Każda instancja składa się z serwera balansującego ruch (round robin oraz na podstawie nagłówków żądania), serwerów z Enterprise Integrator oraz docelowo serwerów z Api Manager. Zasoby danych, którymi chcemy zasilać System udostępniane są poprzez usługi SOAP, zasoby REST. Zabezpieczenia usług SOAP i zasobów REST nie są ustandaryzowane, posiadamy usługi zabezpieczone: WS-Security (w tym podpis otoczony), Oauth, Basic Auth lub bez zabezpieczeń – należy to wziąć pod uwagę. Dane dostarczane poprzez WSO2 pochodzą w większości z aplikacji dziedzinowych, używanych wewnętrznie w UMK.
6. Oczekiwane jest przeniesienie danych z hurtowni danych STRADOM z lekką modyfikacją jej struktury (ok 3%), a w kolejnej fazie konfiguracja umożliwiająca pozyskanie danych bezpośrednio ze źródeł zewnętrznych (bazy dziedzinowe jednostek miejskich oraz wydziałów). Dane będą gromadzone w hurtowni danych systemu STRADOM opartej na technologii Microsoft SQL Server. Dodatkowo w ramach Systemu oczekujemy funkcjonalności pozwalających na przeprowadzanie analiz, w tym przekrojowych z wykorzystaniem również źródeł zewnętrznych (API, IoT). Na potrzeby tego wdrożenia oczekujemy przeniesienia bazy operacyjnej STRADOM oraz hurtowni danych do nowego środowiska. Hurtownia danych to 160GB danych, w hurtowni danych jest 207 tabel, 11 kostek OLAP, z tym, że nie oznacza to, że w STRADOM jest 11 raportów. Użytkownik ma mieć możliwość wygenerowania w Systemie dowolnej ilości raportów na podstawie każdej kostki.
7. Zamawiający posiada wdrożoną Platformę Wirtualizacji Danych – Querona, gdzie zarządzamy wirtualnymi bazami, które są zasilane bezpośrednio z systemów dziedzinowych.
8. System ma umożliwiać zarządzanie danymi nieustrukturyzowanymi pochodzącymi m.in. z czujników parkowania, czujników w pojazdach komunikacji miejskiej, miejskiego monitoringu, próbek głosu z call center, obrazów, postów z mediów społecznościowych.
9. System ma umożliwić również uruchamianie przepływów danych w rozdzielczościach 5, 15 minutowych lub godzinnych (np. na potrzeby monitorowania strefy czystego transportu, ograniczonego ruchu).
10. System ma umożliwiać pracę jednocześnie nieograniczonej liczbie Użytkowników o dowolnym poziomie uprawnień.
11. Dla danych krytycznych wymagany jest tryb pracy HA. Dla innych danych – według potrzeb. Zamawiający posiada mechanizmy HA na węźle VMWare.
12. Oczekiwana jest możliwość rozbudowy Systemu o dodatkowe moduły według narastających potrzeb GMK. Dane do ML Zamawiający będzie pozyskiwać zarówno z baz relacyjnych jak i NoSQL. Większość systemów wdrożonych u Zamawiającego jest oparta o bazy Oracle, SQL Server, PostgreSQL oraz okazjonalnie MySQL. Wolumen danych oraz liczebność zespołu analitycznego, będzie uzależniona od potrzeb biznesowych interesariuszy. Na ten moment identyfikujemy projekty w zakresie przetwarzania języka naturalnego (NLP), analizy predykcyjnej, zbudowanie digital twin dla miasta Krakowa, analizy nagrań (text to speech) oraz wielu innych projektów, których zakres i termin realizacji nie jest jeszcze znany. Proponowane rozwiązanie powinno być

przygotowane w taki sposób, aby umożliwić ww. analizy bez konieczności złożonej rozbudowy Systemu.

4. Monitorowanie Użytkowników (logi)

1. System, który przetwarza dane osobowe, musi spełniać następujące wymagania:
   a) zapis daty i godziny wprowadzenia danych do Systemu, określenia Użytkownika, który dane wprowadził i zakresu tych danych,
   b) zapis źródła pozyskania danych osobowych w przypadku, gdy dane pozyskano z innego źródła niż osoba, której dane dotyczą,
   c) zapis informacji o odbiorcach lub kategoriach odbiorców, którym dane osobowe zostały przekazane, wraz z określeniem daty i zakresu udostępnianych danych,
   d) zapis eksportu do edytowalnego pliku treści danych osobowych,
   e) zapis daty i godziny zmiany danych w Systemie i określenia Użytkownika, który zmiany wprowadził,
   f) zapis usunięcia danych z Systemu,
   g) zapis oznaczenia wraz z odnotowaniem daty danych, których przetwarzanie zostało ograniczone,
   h) zapis oznaczenia wraz z odnotowaniem daty danych, wobec przetwarzania których wniesiono sprzeciw,
   i) zapis wygenerowania i wydrukowania raportu zawierającego informacje, o których mowa w lit. a–h w dowolnym określonym przez żądającego raport układzie i zakresie.
2. System musi posiadać odpowiednie rejestry, umożliwiać ich przeglądanie, sortowanie, filtrowanie, wyszukiwanie danych po dowolnych polach. Z rejestrów tych musi być możliwość generowania raportów w zakresie:
   a) Historii zmian uprawnień Użytkowników (z dokładnością do roli): login, nazwisko, imię, jednostka, komórka organizacyjna, rola, data nadania roli, data odebrania roli.
   b) Historii listy sesji Użytkowników: zawierać będzie listę wszystkich sesji Użytkowników, wraz z informacjami: login, nazwisko, imię, jednostka, komórka organizacyjna, data i godzina początku sesji, data i godzina zakończenia sesji (jeżeli sesja już została zakończona), adres IP komputera, na którym powstała sesja.
   c) Listy otwartych sesji: login, nazwisko, imię, jednostka, komórka organizacyjna, data /godzina początku sesji (musi być możliwość wylogowania wszystkich Użytkowników).
   d) Historii logowań: login, nazwisko, imię, jednostka, komórka organizacyjna, data i godzina zalogowania, data i godzina wylogowania, czas logowania.
   e) Kont Użytkowników w Systemie: login, nazwisko, imię, jednostka, komórka organizacyjna, data założenia konta, data dezaktywacji konta lub blokady konta, czy aktywne, do kiedy ważne, data zmiany hasła, data ostatniego logowania, status konta.
   f) Historii zmian dotyczących kont Użytkowników: zawiera wszystkie atrybuty konta Użytkownika (login, nazwisko, imię, jednostka, komórka organizacyjna, data założenia konta, data zablokowania konta, czy aktywne, do kiedy ważne, data zmiany hasła) oraz powiązań konta Użytkownika z innymi obiektami (np. uprawnienia, sesje), wraz z datą i godziną zmiany oraz informacją o tym kto zmianę wykonał.
   g) Listy aktywnych Użytkowników wraz z przypisanymi rolami (imię, nazwisko, login, jednostka, komórka organizacyjna, role).
   h) Listy osób, które w zadanym okresie miały nadane uprawnienia, przy czym powinna być możliwość wyszukiwania po parametrach:
      - okres (od, do) wraz z możliwością wyszukania listy osób, które miały nadane uprawnienia przez cały okres jak i w jego fragmencie,
      - rola (możliwe zaznaczenie kilku).

      Lista osób, o której mowa w pkt. h) powinna zawierać następujące informacje: login, nazwisko, imię, jednostka, komórka organizacyjna, data nadania uprawnienia, data odebrania uprawnienia.

Zakres powyższych raportów musi zostać ostatecznie uzgodniony na etapie Analizy Przedwdrożeniowej. Wszystkie raporty wskazane w pkt. 2) muszą posiadać:

- nagłówek zawierający tytuł raportu,
- zadane parametry wyszukiwania, dla których został wygenerowany raport,
- informację kto i kiedy wygenerował raport,
- część zasadniczą z wygenerowanymi danymi wraz z nagłówkami kolumn.

3. W Systemie muszą być logowane zdarzenia z dokładnością do każdego parametru określonego w pkt. 2). Komunikaty zdarzeń muszą być opisane w sposób czytelny dla Użytkownika.
4. W Systemie muszą być rejestrowane działania Użytkowników oraz zdarzenia związane z bezpieczeństwem informacji. Logi musza zawierać rejestracje wszystkich działań Użytkownika w Systemie wraz z datą, godziną, minutą i sekundą wykonania tych działań. Dane te muszą być przechowywane przez określony przez Zamawiającego czas dla potrzeb przyszłych postępowań wyjaśniających oraz monitorowania kontroli dostępu. Logi bieżące mają być przechowywane w Systemie, natomiast reguły związane z przechowywaniem logów archiwalnych zostaną uzgodnione na etapie Analizy Przedwdrożeniowej.
   System musi zawierać mechanizm do przeglądania logów bieżących (wstępnie wszystkie do 24 miesięcy; okres ustawiany parametrem) i archiwalnych (wstępnie wszystkie powyżej 24 miesięcy; okres ustawiany parametrem) w tym zapewniający możliwość:
   - wyszukiwania,
   - filtrowania po wybranych przez Użytkownika typach zdarzeń i ich cechach,
   - sortowania po wybranych przez Użytkownika typach zdarzeń i ich cechach.
5. System musi zapewniać mechanizm eksportu pliku logów do serwera zewnętrznego przy użyciu standardowych protokołów i mieć możliwość synchronizacji z serwerem czasu (protokół NTP).
6. W przypadku, gdy w Systemie jest realizowany interfejs integracyjny obligatoryjne jest odnotowywanie działań związanych z uruchamianiem funkcji interfejsu integracyjnego wraz z możliwością włączenia powiadamiania mailowego o błędach.
7. Generowane raporty muszą mieć możliwość eksportu do plików innego formatu, w szczególności xlsx, csv, w zależności od zapotrzebowania Użytkownika.

5. Integracje

5.1. Sezam

1. System ma być zintegrowany z wdrożonym w UMK systemem do zarządzania upoważnieniami Użytkowników
2. Implementacja uniwersalnego interfejsu ma na celu zunifikowanie mechanizmu nadawania uprawnień. Idea bazuje na założeniu, że dla większości aplikacji nadawanie uprawnienia ma charakter hierarchiczny, gdzie na szczycie hierarchii znajduje się lista ról, które mogą być przydzielone, a na kolejnych poziomach uszczegółowiane są uprawnienia związane z daną rolą. Zunifikowany sposób nadawania uprawnień, pozwala na zdefiniowanie uniwersalnego interfejsu graficznego, służącego nadawaniu uprawnień, jak również interfejsów typu web-service, pozwalających na pobranie definicji dostępnych uprawnień oraz przekazywania do systemów skonfigurowanych dla konkretnych pracowników uprawnień. Uprawnienia nadawane pracownikom do danego systemu konfigurowane są w oparciu o plik XML konfiguracji uprawnień.
3. System SEZAM:
   - pobiera plik konfiguracji uprawnień z Systemu
   - pozwala na zdefiniowanie w oparciu o pobrany plik uprawnień w procesie nadawania lub modyfikacji uprawnień
   - przekazuje zdefiniowane uprawnienia do Systemu
4. W celu pobrania konfiguracji z Systemu oraz nadania w nim uprawnień wykorzystany jest interfejs web-service. Instrukcja integracji w tym struktura pliku konfiguracji uprawnie zostanie przekazana na etapie Analizy przedwdrożeniowej.

5.2. Mapy ArcGIS lub ISDP

System ma być zintegrowany z mapami ArcGIS lub ISDP w celu wyświetlania w Systemie danych na mapach. W ramach integracji dane mają być zbierane z różnych źródeł i przetwarzane w celu uzyskania spójnego zestawu informacji. Następnie dane te mają zostać wykorzystane do stworzenia mapy, która ma przedstawić zebrane informacje w sposób czytelny i łatwy do zrozumienia dla Użytkownika. Mapa ta

może zawierać różne rodzaje informacji, takie jak dane geograficzne, dane demograficzne czy dane dotyczące infrastruktury.

5.3. CSS

Integracja będzie niezbędna w przypadku decyzji na etapie Analizy Przedwdrożeniowej o potrzebie utworzenia lub wykorzystania słownika centralnego. Słownik taki zostanie przygotowany w aplikacji CSS, a następnie będzie można pobierać z niego dane poprzez interfejsy integracyjne.

Interfejsy integracyjne do CSS zapewniają następującą funkcjonalność udostępnionych dla Systemu słowników: pobieranie listy słowników, pobierania konkretnego słownika, pobierania listy pozycji słownikowych oraz pobierania konkretnej pozycji.

Dostęp do usługi REST API wymaga autoryzacji wg standardu Oauth2.0 z wykorzystaniem przepływu Client Credentials.

Instrukcja integracji oraz interfejsy zostaną udostępnione na etapie Analizy Przedwdrożeniowej.

6. Wymagane standardy SI UMK mające zastosowanie przy wdrożeniu i utrzymaniu rozwiązania

6.1. Platforma Sprzętowa i Oprogramowanie Systemowe

UMK udostępni na potrzeby uruchomienia Systemu zwirtualizowaną infrastrukturę opartą o rozwiązanie VMware w wersji 9 składającą się z maksymalnie 5 serwerów o określonych parametrach (dotyczy pojedynczego serwera, bez miejsca na dane):

1. Liczba procesorów (vCPU): 16 vCORE
2. Pamięć RAM: 64 GB RAM – dla serwerów aplikacyjnych, w przypadku serwerów bazodanowych maksymalnie do 128 GB RAM
3. Przestrzeń dyskowa (SSD): 1 TB

Zamawiający zapewni elementy Oprogramowania Systemowego obejmujące: systemy operacyjne, oprogramowanie bazodanowe oraz sterowniki wymagane do ich poprawnego działania. Dopuszcza się stosowanie tylko określonych poniżej systemów operacyjnych oraz oprogramowania dodatkowego (bazy danych, serwery aplikacyjne, itp.).
W przypadku konieczności dostarczenia przez Wykonawcę dodatkowych licencji dotyczących Oprogramowania Systemowego, ponad te, o których mowa w rozdziałach 6.1.1. – 6.1.3., należy uwzględnić, że środowiska Zamawiającego działają w klastrze składającym się z czterech serwerów. Każdy z nich ma 16 rdzeni fizycznych. W zależności od wykorzystanych przez Wykonawcę rozwiązań konieczne jest dostarczenie licencji, tak aby cały klaster, na którym znajduje się serwer wirtualny, był zgodny licencyjnie.

6.1.1. Systemy operacyjne

Stosować można alternatywnie następujące rodzaje systemu operacyjnego w wersji stabilnej i najbardziej aktualnej (Zamawiający na bieżąco aktualizuje oprogramowanie najpóźniej do 3 tygodni od publikacji poprawki/wersji przez producenta):

1. Od Windows Server 2019 (w ramach posiadanych licencji)
2. RedHat Enterprise Linux

6.1.2. Bazy danych

Stosować można alternatywnie następujące rodzaje baz danych w wersji stabilnej i najbardziej aktualnej (Zamawiający na bieżąco aktualizuje oprogramowanie najpóźniej do 3 tygodni od publikacji poprawki/wersji przez producenta):

1. Microsoft SQL Server STD
2. PostgreSQL
3. MySQL

Powyższe wymaganie nie wyklucza wykorzystania innej platformy bazodanowej wbudowanej (embedded) we wdrażane rozwiązanie. W tej sytuacji platforma bazodanowa nie jest zarządzana i utrzymywana przez Zamawiającego, a Wykonawca jest zobowiązany zapewnić wymagane pokrycie licencyjne.

6.1.3. Serwery aplikacji www

Stosować można alternatywnie następujące serwery aplikacji www w wersji stabilnej i najbardziej aktualnej (Zamawiający na bieżąco aktualizuje oprogramowanie najpóźniej do 3 tygodni od publikacji poprawki/wersji przez producenta):

1. WildFly
2. Apache/Tomcat
3. IIS Microsoft
4. Nginx

6.1.4. Aplikacje i systemy Zamawiającego wykorzystywane na potrzeby realizacji Umowy

1. **Szyna WSO2 Enterprise Integrator** - Dokumentacja
2. **API Manager WSO2** - Dokumentacja
3. **GitLab** - jako repozytorium kodu - Dokumentacja - Wersja community
4. **Querona** – Platforma Wirtualizacji Danych - Dokumentacja
5. **Power BI** – raportowanie statyczne - Dokumentacja - na potrzeby wdrożenia Zamawiający przeznaczy 20 licencji Power BI Pro
6. **Solarwinds** – aplikacja służąca do monitorowania parametrów Systemu - Dokumentacja,
7. **Mapy ESRI / ISDP** – Dokumentacja
8. **Commvault** – system centralnego backupu działający u Zamawiającego
9. **Atmosfera** – Help desk – aplikacja wykorzystywana w UMK do zgłaszania do Wykonawcy błędów lub wniosków o realizacje usług dotyczących Systemu.

6.2. Kopia zapasowa

1. Używany w UMK system centralnego backupu posiada następujące standardy:
   a) Oprogramowanie: Commvault.
   b) Okres przechowywanie kopii zapasowych: 1 tydzień.
   c) Obejmuje wszystkie systemy serwerowe SI UMK, w tym bazy danych. Nie obejmuje stacji roboczych Użytkowników, a jedynie systemy serwerowe.
2. Wykonawca wdrażający System skorzysta z systemu centralnego backupu w UMK w ramach standardowego backupu całej maszyny.
3. Wykonawca musi współpracować z Zamawiającym w trakcie procesu obejmowania backupem Systemu (dodawania do systemu centralnego backupu oraz tworzenia optymalnej polityki backupu).

6.3. Ogólne wymagania dla Systemu

Wdrażany System musi spełniać następujące ogólne wymagania:

1. System musi pracować w technologii warstwowej, rozumianej jako architektura typu klient-serwer, zakładająca co najmniej separowanie interfejsu Użytkownika, logiki biznesowej oraz danych, przy czym każda z powyższych warstw może mieć własny podział warstwowy. Architektura tego typu pozwala aktualizować lub zastępować poszczególne moduły niezależnie od siebie, w miarę jak zmieniają się warunki techniczne – przykładowo, zmiana systemu operacyjnego

na komputerze Użytkownika (np. z Windows na Linux lub odwrotnie), wpływa jedynie na warstwę interfejsu Użytkownika, ale nie na przetwarzanie i składowanie.

2. System musi być dostępny przez przeglądarkę www (co najmniej EDGE, Mozilla Firefox, Google Chrome do 3 wersji wstecz) przez szyfrowane połączenie (https) i umożliwiać pracę wielu Użytkowników jednocześnie. Oznacza to, że wszystkie wskazane w wymaganiach funkcjonalnych funkcje Systemu mają być dostępne przez interfejs www.
3. Technologia ma umożliwić integrację z innymi aplikacjami SI UMK poprzez webservice.
4. System ma spełniać wymagania Polityki Bezpieczeństwa Informacji (w tym w zakresie polityki haseł, które: składa się z minimum 12 znaków, zawiera małe litery, zawiera duże litery, zawiera cyfry, zawiera znaki specjalne, jest inne niż 10 ostatnio wprowadzonych haseł; wymagana zmiana wykorzystywanych haseł w regularnych odstępach czasu (raz na 90 dni)). Musi być możliwość zmiany polityki haseł, co oznacza, że musi być ustawiana w parametrach Systemu.
5. System musi spełniać wymagania dotyczące rozporządzenia Rady Ministrów z dnia 12 kwietnia 2012 roku w sprawie Krajowych Ram Interoperacyjności, minimalnych wymagań dla rejestrów publicznych i wymiany informacji w postaci elektronicznej oraz minimalnych wymagań dla systemów teleinformatycznych Dz. U. 2017 poz. 2247).
6. System ma uwzględniać warunki środowiskowe SI UMK, tj. musi uwzględniać zawarte w tym dokumencie standardy dotyczące Platformy Sprzętowej wraz z licencjami, które posiada UMK.
7. Interfejs Użytkownika Systemu musi być w języku polskim.
8. System musi umożliwiać integrację z zewnętrznymi narzędziami monitorującymi. Zamawiający będzie stosował system Orion SolarWinds do monitorowania parametrów działania Systemu.
9. Czas pobrania danych i odświeżania/odbudowy ekranu po czynności wykonanej przez Użytkownika nie może być dłuższy niż 4 sekundy dla 95% żądań (przy obciążeniu 10 żądań na sekundę); przy czym w czas ten nie wlicza się czasu pobierania informacji z zewnętrznych aplikacji, spowolnień spowodowanych ograniczeniami łącz internetowych po stronie dostawców tych łącz oraz czasu przesyłania plików powyżej 1,2 MB.
10. Przyjmuje się następujące założenia dotyczące prędkości odszukiwania i wyświetlania rekordów danych wyszczególnione poniżej (dla 95% żądań przy obciążeniu 10 żądań na sekundę):
    - wyszukiwanie po dowolnych kryteriach w czasie: do 3 sekund,
    - generowanie raportów w czasie: do 30 sekund,
    - zatwierdzenie zmian przez Użytkownika w czasie: do 3 sekund.

    Powyższe wymagania dotyczą działania w przypadku braku konieczności pobierania danych ze źródeł zewnętrznych. Na etapie konkursu Zamawiający może wymagać od Wykonawcy przeprowadzenia przykładowego wyszukiwania/ generowania raportu i przedstawienia wyników z tego procesu oraz zaprezentowania przykładu działania modułu raportowego
11. Dane w bazie muszą być zapisywane według standardu Unicode UTF-8.
12. Interfejs webowy Systemu musi uwzględniać komfort pracy Użytkowników z dysfunkcjami (WCAG 2.1. AA) minimum w zakresie:
    - Postrzegalności: 1.3 Możliwość adaptacji - Odpowiednia (zrozumiała) prezentacja zawartości, 1.4 Możliwość rozróżnienia - Ułatwienie percepcji treści.
    - Funkcjonalności: 2.2 Wystarczająca ilość czasu, 2.3 Ataki padaczki – migotanie, 2.4 Możliwość nawigacji, 2.5 Sposoby wprowadzania danych.
    - Zrozumiałości: 3.2 Przewidywalność, 3.3 Pomoc przy wprowadzaniu informacji.
    - Kompatybilności: 4.1 Kompatybilność.
13. System musi być zintegrowany z: systemem do nadawania upoważnień (SEZAM), Centralnym System Słowników (CSS), mapami (ArcGIS lub ISDP).
14. W przypadku decyzji na etapie Analizy Przedwdrożeniowej o potrzebie utworzenia słownika oraz decyzji, że nie będzie to słownik centralny należy w ramach Systemu dostarczyć moduł słowników. Słowniki muszą spełniać następujące wymagania:
    - Muszą umożliwiać podpowiadanie i walidację danych.
    - Muszą posiadać następujące funkcje:
        - Przeglądanie listy dostępnych słowników
        - Przeglądanie słownika
        - Dodanie i modyfikacja wartości do słownika
        - Usunięcie wartości ze słownika
        - Sklejenie wartości słownika
        - Podpowiadanie wartości ze słownika

- Wszystkie operacje na słownikach muszą być logowane (kto i kiedy daną wartość zmienił, dodał lub usunął). Log ten ma być dostępny dla Administratora Technicznego. Usuwanie wartości ze słownika, zmiany wartości w słownikach nie mogą wpływać na wprowadzone i zatwierdzone już w Systemie dane (nie mogą zmieniać danych już wprowadzonych, a powinny być widoczne dopiero w nowych dokumentach). Zmiany wartości słowników powinny być oznaczone z podaniem okresu obowiązywania. Historyczna wartość słownika powinna być możliwa do sprawdzenia przy wskazaniu odpowiedniego statusu wartości słownikowej nie tylko na podstawie logów. Usuwanie wartości ze słownika nie powoduje fatycznego usunięcia, a jedynie ustawienia na status archiwalny oraz zmiany okresu, w którym wartość była dostępna.

15. W przypadku gromadzenia i przetwarzania danych osobowych System musi spełniać wymagania określone w rozdziale 4. pkt 1, a ponadto, uwzględniając stan technologii informatycznej, musi być zgodny z rozporządzeniem Rozporządzenie Parlamentu Europejskiego i Rady (UE) 2016/679 z dnia 27 kwietnia 2016 r. w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (ogólne rozporządzenie o ochronie danych), zwanym dalej „RODO".

6.4. Role Użytkowników Systemu

W Systemie musi znajdować się wykaz możliwych do przyznania ról wraz z opisem. Możliwość pracy w Systemie z uprawnieniami Administratora ma być na podstawie nadanej roli (login Użytkownika, a nie login: Administrator).
W Systemie muszą zostać opracowane minimum następujące role:

1. Administrator Techniczny – rola umożliwiająca dostęp do wszystkich funkcji Systemu związanych z administrowaniem Systemem i zarządzaniem uprawnieniami oraz do związanych z tym słowników i parametrów (np. modyfikacja opisu roli, okresowe raporty kont i przyznanych im ról, dostęp do logów Systemu, zarządzanie parametrami Systemu, weryfikacja nieautoryzowanych dostępów do Systemu).
2. Gospodarz - rola dająca dostęp do wszystkich funkcji związanych z merytoryczną obsługa Systemu oraz związanymi z tym słownikami i parametrami (w tym kontrola i aktualizacja słowników i parametrów związanych z merytorycznymi funkcjonalnościami Systemu). Ma możliwość podglądu przyznanych ról i uprawnień w Systemie.
3. Obserwator – Użytkownik mający uprawnienia tylko do podglądu wybranych danych bez możliwości ich edycji (bez słowników).
4. Administrator Uprawnień - rola ma umożliwiać wyłącznie nadawanie i odbieranie uprawnień do Systemu, Użytkownik z tą rolą nie może mieć możliwości zmiany swoich uprawnień oraz nadać roli Administratora Technicznego.
5. Użytkownik biznesowy – rola Użytkownika, który pracuje w Systemie wykorzystując dostępne funkcje jednak nie ma dostępu do słowników i parametrów Systemu ani do części związanej z zarządzaniem Systemem i Użytkownikami.
6. Analityk danych – zakres uprawnień do ustalenia w trakcie Analizy Przedwdrożeniowej.
7. Programista - zakres uprawnień do ustalenia w trakcie Analizy Przedwdrożeniowej.

System musi posiadać możliwość przypisania Użytkownika do dowolnej roli oraz przypisania mu dowolnej liczby ról.

6.5. Standardy dla usług integracyjnych w UMK

W przypadku budowy przez Wykonawcę interfejsów integracyjnych w celu realizacji wymagań projektu dostarczony interfejs integracyjny musi spełniać następujące wymagania:

1. Dostarczone rozwiązanie ma bazować na modelu usługowym, czyli wymiany danych poprzez mechanizm serwisów webowych.
2. Integracje muszą odbywać się poprzez jeden z poniższych sposobów:
    a) Protokół komunikacyjny SOAP.
    b) Styl architektury oprogramowania REST.

3. W przypadku stosowania SOAP, wymagane jest dostarczenie:
    a) plików WSDL - definiuje, jakie informacje i w jaki sposób można wydobyć z serwisu (reguły określające sposób komunikacji z serwerem), stosowane zabezpieczenia, adres właściwego serwisu, listę udostępnianych metod wraz z argumentami i zwracanymi typami,
    b) plików XML ze schematami XSD lub w przypadku danych przestrzennych Geography Markup Language (GML),
    c) wymagane jest dostarczenie przykładowych payloadów (requestów)/responsów. Co najmniej jeden payload na każdą akcję (operację/metodę) w usłudze,
    d) wszystkich certyfikatów wykorzystywanych do zabezpieczenia usługi wraz z ich hasłami, opis standardów zabezpieczeń usługi. Opis użytych zabezpieczeń należy zdefiniować za pomocą specyfikacji WS-Policy i dołączyć do udostępnionego WSDL,
    e) projektu w SoapUI lub Postman przygotowanego w taki sposób, że po uruchomieniu z dowolnego komputera z tej samej podsieci, gdzie znajduje się usługa, będzie działał poprawnie. Projekt będzie zawierał każdy z przykładowych payloadów z punktu c.
4. W przypadku stosowania REST, wymagane jest dostarczenie:
    a) plików RAML opisującego sposób wywołania usługi (opisywaniu zasobów, metod, parametrów, odpowiedzi, typów mediów i innych konstrukcji HTTP) lub zainstalowany SwaggerUI z definicjami OpenApi lub Swagger i możliwością ich wywołania bezpośrednio z graficznego interfejsu webowego,
    b) plików JSON lub XML definiujący schemy wejściowe/zwrotne/błędy, a w przypadku danych przestrzennych Geography Markup Language (GML),
    c) wymagane jest dostarczenie przykładowych payloadów (requestów)/responsów. Co najmniej jeden payload na każdą akcję (operację/metodę) w usłudze,
    d) opis sposobu zabezpieczenia usługi (opis standardów zabezpieczeń usługi). Opis musi umożliwić integrację aplikacji z API, również w przypadku wykorzystania standardu JWT lub OAuth,
    e) projektu w SoapUI lub Postman przygotowanego w taki sposób, że po uruchomieniu z dowolnego komputera z tej samej podsieci, gdzie znajduje się usługa, będzie działał poprawnie. Projekt będzie zawierał każdy z przykładowych payloadów z punktu c.
5. Mechanizm autentykacji oraz szyfrowania komunikatów implementowany ma być w oparciu o standard WS-Security w modelu wykorzystującym certyfikaty X.509 lub w oparciu o serwer KERBEROS; w przypadku REST standardu JWT/OAuth.
6. Komunikaty przesyłane między dostawcą, a odbiorcą usług mają korzystać z protokołu HTTPS.
7. Znaki w dokumentach wysyłanych z Systemu mają być kodowane według standardu Unicode UTF-8.
8. Usługi sieciowe mają korzystać z mechanizmów zapewniających zachowanie integralności, niezaprzeczalności, poufności i autentyczności danych przesyłanych w komunikatach.
9. Usługa powinna logować jej wywołania oraz w logach powinna być informacja zawierająca: datę/czas, adres IP z którego było wywołanie, request. Usługa musi mieć możliwość prostego włączania/wyłączania logowania poprzez plik konfiguracyjny.
10. Informacje w logach przechowywane mają być przez okres co najmniej 2 lat od daty ich zapisu. Okres ten może być inny, zgodny ze wskazanym w odrębnych przepisach prawa. Należy ustawić dzienną i miesięczną rotację pliku logów.
11. Przygotowany przez Wykonawcę interfejs zostanie podłączony przez UMK do Szyny Danych (ESB) przy współpracy z Wykonawcą. Wykonawca nie otrzyma bezpośredniego dostępu do ESB. Interfejs ten może być wykorzystany na rzecz innych integracji z dowolnymi aplikacjami w SI Gminy Miejskiej Kraków. W niektórych przypadkach Zamawiający może umieszczać interfejsy typu REST w narzędziu API Manager.

6.6. Analiza Przedwdrożeniowa

1. Celem Analizy Przedwdrożeniowej jest:
    a) uszczegółowienie jednoznacznej interpretacji wymagań Zamawiającego i sposobu ich realizacji,

b) określenie zasad konfiguracji Systemu,
c) rozpoznanie wymaganej integracji z istniejącymi systemami,
d) określenie Platformy Sprzętowej oraz Oprogramowania Systemowego niezbędnego do działania Systemu.

2. W wyniku przeprowadzonej Analizy Przedwdrożeniowej Wykonawca dostarczy Dokument analizy zawierający co najmniej:
   a) jednoznaczną i zamkniętą listę wymagań wraz z określeniem sposobu ich realizacji oraz kryteria akceptacji dla wymagania,
   b) listę słowników i parametrów Systemu,
   c) listę potrzebnych raportów wraz z opisem,
   d) jednoznacznie ustalone zasady konfiguracji Systemu,
   e) jednoznacznie określone założenia integracji z innymi aplikacjami,
   f) diagramy architektury logicznej i fizycznej Systemu wraz z opisami. Diagramy architektury powinny zawierać również rozmieszczenia oraz powiązanie jego poszczególnych elementów, na poziomie sprzętowym oraz oprogramowania, z uwzględnieniem wersji produkcyjnej i testowej.
   Musi zostać dostarczone również diagram architektury wewnętrznej Systemu (DARWA) - diagram prezentuje elementy składowe Systemu (moduły).
   Powyższe diagramy powinny zostać zamodelowane zgodnie ze standardem UMK umieszczonym w Załączniku nr 3 do niniejszego Załącznika „Zasady modelowania architektury rozwiązania w UMK"
   g) opis Platformy Sprzętowej i Oprogramowania Systemowego. W przypadku wykorzystania komponentów lub oprogramowania firm trzecich, konieczne jest wskazanie tych komponentów oraz informacji o licencji,
   h) zakres instruktażu,
   i) plan testów zawierający elementy, o których mowa w rozdziale 6.8.

6.7. Wymagania dotyczące harmonogramu realizacji wdrożenia

1. Harmonogram musi zawierać kamienie milowe i produkty.
2. Harmonogram musi zawierać sekwencje zdarzeń.
3. W harmonogramie muszą zostać w szczególności uwzględnione następujące etapy:
   a) dostarczenie Analizy Przedwdrożeniowej,
   b) dostarczenie planu testów i scenariuszy testowych,
   c) przygotowanie Platformy Sprzętowej przez Zamawiającego (należy uwzględnić czas niezbędny do przygotowania tej platformy od momentu dostarczenia informacji o niezbędnych wymaganiach co do tej platformy) – z podziałem na środowisko testowe i produkcyjne,
   d) dostarczenie, instalacja i konfiguracja Systemu przez Wykonawcę w środowisku testowym,
   e) dostarczenie uzupełnionych o kroki w Systemie przypadków testowych umożliwiających rozpoczęcie testów,
   f) dostarczenie Dokumentacji,
   g) testy Zamawiającego i Wykonawcy w środowisku testowym,
   h) dostarczenie, instalacja i konfiguracja Systemu przez Wykonawcę w środowisku produkcyjnym,
   i) testy Zamawiającego i Wykonawcy w środowisku produkcyjnym,
   j) instruktaż dla Użytkowników
4. Harmonogram musi zawierać terminy, czas trwania poszczególnych etapów (co najmniej pomiędzy kamieniami milowymi).

6.8. Wymagania dotyczące planu testów

1. Plan testów zawiera scenariusze testowe które są niezbędne do sprawdzenia poprawności działania Systemu. Każdy scenariusz ma być odzwierciedleniem dokładnie określonej

funkcjonalności. Każdy scenariusz testowy powinien posiadać identyfikator, nazwę, opis, warunki wstępne, wykaz przypadków testowych. Scenariusze i przypadki testowe muszą zostać uzupełnione o niezbędne kroki do wykonania przed rozpoczęciem testów.

2. Plan testów w zakresie Systemu i wdrożenia musi zawierać co najmniej:
   a) testy funkcjonalne,
   b) testy wydajności,
   c) testy bezpieczeństwa – wymagania w zakresie tych testów zostały opisane w rozdziale 6.9.,
   d) testy akceptacyjne.
3. Plan testów zawiera listę funkcjonalności Systemu, które mają zostać poddane testom.
4. Wyłączenia – Zamawiający dopuszcza, aby testy nie obejmowały wybranych elementów w zakresie i obszarze testów, jednak w takiej sytuacji, fragmenty te musza być jasno i precyzyjnie określone wraz z podaniem przyczyny, dla której następuje wyłączenie. Wyłączenia muszą być zatwierdzone przez Zamawiającego. Brak zgody Zamawiającego skutkuje koniecznością przeprowadzenia testów w tym zakresie.
5. Plan testów określa warunki, których spełnienie pozwala na rozpoczęcie testów. Zapis tych warunków musi być odzwierciedlony w harmonogramie realizacji wdrożenia.
6. Plan testów zawiera zestaw kryteriów pozwalających uznać testy za zakończone z wynikiem pozytywnym. Zestaw kryteriów podlega akceptacji Zamawiającego.
7. Plan testów zawiera harmonogram ich realizacji, z podaniem terminu rozpoczęcia i zakończenia zadań testowych oraz informację kto i w którym środowisku wykonuje testy (Wykonawca, Wykonawca z Zamawiającym, Zamawiający).
8. Plan testów zawiera spis środowisk przeznaczonych do wykorzystania w trakcie testów.
9. Plan testów zostanie opracowany przez Wykonawcę.
10. Plan testów musi zostać zaakceptowany przez Zamawiającego w zakresie zgodności z wymogami wskazanymi w Umowie.

6.9. Wymagania dotyczące wykonania testów bezpieczeństwa Systemu

1. Wykonanie testów bezpieczeństwa jest niezbędne dla uruchomienia produkcyjnego Systemu i ich przeprowadzenie leży po stronie Wykonawcy.
2. Testy, o których mowa w pkt. 1. muszą zostać przeprowadzone zgodnie z metodyką OWASP TOP 10:2021.
3. Zakres w/w testów musi obejmować co najmniej:
   - testy uwierzytelniania, autoryzacji oraz mechanizmów zarządzania sesją Systemu,
   - testy konfiguracji Systemu, sprawdzenie błędów generowanych przez System i jej komponenty oraz wykonanie testów mających na celu wykrycie podatności,
   - testy walidacji danych wejściowych,
   - testy logiki biznesowej Systemu,
   - testy dodatkowe (Web Services, test CMS, SSL itp.).
4. Po przeprowadzeniu testów bezpieczeństwa Systemu, a przed wdrożeniem produkcyjnym, Wykonawca prześle, na adres cyberbezpieczenstwo@um.krakow.pl raport z przeprowadzonych testów zawierający co najmniej informacje na temat metod jakimi testy zostały wykonane, zakresu testów, stwierdzonych podatności lub ich braku.
5. Wykonawca zezwala na udostępnianie raportu osobom trzecim oraz prawo do osobnego korzystania z każdego z elementów raportu.
6. Wykonawca oświadcza, iż w okresie obowiązywania Umowy, w której Wykonawca pozostaje stroną o określonym zakresie odpowiedzialności za System, poddaje się dobrowolnie testom bezpieczeństwa informatycznego Systemu organizowanym przez Zamawiającego.
7. Podatności, zalecenia i rekomendacje powstałe w wyniku testów, o których mowa w pkt. 6. oraz podatności, o których mowa w pkt. 4., będą zgłaszane Wykonawcy zgodnie z procedurą dotyczącą usuwania błędów, określoną w obowiązującej Umowie między stronami.

6.10. Wymagania dotyczące Dokumentacji

6.10.1. Wymagania ogólne

1. Dokumentacja sporządzona na potrzeby zamówienia musi być zgodna ze stanem prawnym aktualnym na dzień przedstawienia jej do odbioru Zamawiającemu.
2. Dostarczona Dokumentacja musi być w języku polskim (w uzasadnionych i zaakceptowanych przez Zamawiającego przypadkach dopuszczalne są fragmenty w języku angielskim), być spójna i nie może zawierać sprzeczności. Wykonawca musi zapewnić wzajemną zgodność pomiędzy wszystkimi rodzajami informacji umieszczonymi w Dokumentacji, brak logicznych sprzeczności oraz spójność pomiędzy informacjami zawartymi w Dokumentacji.
3. Dostarczona Dokumentacja ma charakteryzować się:
    a) jednolitą strukturą, rozumianą jako podział danego dokumentu na rozdziały, podrozdziały i sekcje w czytelny i zrozumiały sposób,
    b) jednolitym sposobem opisywania rozumianym jako zachowanie spójnej struktury, formy i sposobu pisania,
    c) poprawnością ortograficzną,
    d) aktualnymi odnośnikami do innych dokumentów, rozdziałów lub fragmentów Dokumentacji,
    e) musi w całości opisywać funkcjonalności Systemu,
    f) musi zawierać pełne przedstawienie omawianego problemu obejmujące całość rozpatrywanego zakresu zagadnienia i nie zawierać zbędnej treści,
    g) musi zawierać uzgodnienia poczynione z Zamawiającym w trakcie realizacji przedmiotu Umowy.
4. Dokumentacja musi umożliwiać administrowanie Systemem.

6.10.2. Wymagania Dokumentacji dla zarządzania Platformą Sprzętową

Wykonawca jest zobowiązany do wykonania i dostarczenia Dokumentacji technicznej zarządzania Platformą Sprzętową dla Systemu, zawierającej co najmniej:

1. Opis konfiguracji Systemu, w tym wykaz wdrożonych elementów, powiązania pomiędzy nimi, opis ich konfiguracji, implementacja w środowisku Zamawiającego (integracja) oraz oprogramowania dodatkowego
2. Instrukcje start/stop dla całego środowiska (infrastruktura – systemy operacyjne, wspomagające, bazy danych itp.).
3. Instrukcje eksploatacyjne dla administratorów Systemu.
4. Instrukcje wykonywania kopii zapasowych i odtwarzania z kopii.
5. Instrukcje instalacji i konfiguracji.

6.10.3. Wymagania Dokumentacji użytkowania i technicznej dla Systemu

1. Instrukcja eksploatacyjna użytkowania Systemu ma zawierać:
    a) Opis zastosowania, działania i sposobu wykonania (opis krok po kroku) każdej udostępnionej funkcjonalności Systemu z dokładnością do pojedynczej funkcji.
    b) Listę oraz opis zastosowania wszystkich użytych słowników.
    c) Opis wszystkich parametrów Systemu związanych z jego ustawieniami i funkcjonalnościami.
    d) Wykaz możliwych do przyznania uprawnień do Systemu wraz z ich opisem.
    e) Listę i opis ikon, przycisków i skrótów klawiaturowych.
    f) Instrukcja użytkowania musi być wyposażona w wyszukiwarkę i indeks.

2. Dokumentacja techniczna ma zawierać:
    a) Wymagania techniczne dotyczące sprzętu i środowiska (z dokładnością do wersji środowiska).

b) Ustawienia konfiguracyjne środowiska, w którym pracuje System, w tym również opis implementacji w środowisku SI UMK wraz z procedurami start/stop dla wszystkich elementów Systemu.
c) Opis parametrów konfiguracji Systemu i sposób ich wykorzystania.
d) Diagramy architektury logicznej i fizycznej Systemu.
Diagramy powinny zawierać również rozmieszczenia oraz powiązanie jego poszczególnych elementów, na poziomie sprzętowym oraz oprogramowania, z uwzględnieniem wersji produkcyjnej i testowej.
e) Diagram architektury wewnętrznej Systemu (DARWA) - diagram prezentuje elementy składowe Systemu (moduły),
f) Diagram struktury danych (DARSD) - zawiera logiczny lub fizyczny model danych.
g) Diagram przepływu komunikacji (DARPK) – diagram zwany również diagramem sekwencji prezentujący szczegółowo realizację wybranych aspektów komunikacji pomiędzy Systemem a innymi systemami/aplikacjami,
h) Opis techniczny rodzajów i zastosowanych protokołów komunikacji (w tym certyfikatów).
i) Sposób wykonania instalacji Systemu, instalacji poprawek i kolejnych wersji.
j) Procedura odtworzenia danych i konfiguracji.
k) Schemat baz danych wraz z opisem struktury uwzględniający powiązania i zależności między obiektami w bazie danych.
l) Wykaz danych podlegający kontroli poprawności wraz z informacją o sposobie kontroli poprawności.
m) Wykaz komunikatów diagnostycznych i standardowych błędów (opis błędu, warunki jego powstania).

Diagramy opisane w pkt. d) - g) powinny zostać zamodelowane zgodnie ze standardem UMK umieszczonym w Załączniku nr 3 do niniejszego Załącznika „Zasady modelowania architektury rozwiązania w UMK".

W przypadku udostępnienia przez System interfejsu integracyjnego dokumentacja powinna zawierać dodatkowo:
Instrukcja integracji, w wersji do udostępniania osobom trzecim w celu właściwego zintegrowania się z Systemem powinna zawierać:

a) Opis usługi, interfejsów i wytyczne umożliwiające integrację Systemu z innymi aplikacjami.
b) Pliki ze schematami (WSDL, GML, Swagger/OpenApi, itp.).
c) Opis metod i struktur danych interfejsów.

W przypadku przekazania kodów źródłowych:

a) Charakterystykę katalogów i plików kodu źródłowego.
b) Diagram klas.
c) Komentarze w kodzie źródłowym pozwalające na automatyczne wygenerowanie dokumentacji w formacie HTML lub PDF przy użyciu dedykowanego narzędzia (np. Javadoc).
d) Repozytorium .git zawierające historię zmian kodu.
e) Oraz zgodnie z wymaganiem pkt. 6.12. niniejszego załącznika.

6.11. Kody źródłowe

Kody źródłowe muszą być przekazane w formie elektronicznej (przed kompilacją), umożliwiającej analizę i rozbudowę przez zarówno Zamawiającego jak i firmy trzecie działające na potrzeby Zamawiającego. Wykonawca musi przekazać informację o:

1) wszystkich bibliotekach i dodatkach niezbędnych do kompilacji kodu i uruchomienia Systemu,
2) rekomendowanym środowisku programistycznym wraz z jego konfiguracją i wskazaniem wymaganych aplikacji dodatkowych,

3) parametrach i zmiennych środowiska produkcyjnego Systemu, koniecznych do uruchomienia Systemu,
4) instrukcję krok po kroku z opisem czynności i wylistowanymi komendami, której efektem jest uruchomienie Systemu w środowisku produkcyjnym z kodów źródłowych Systemu (wraz z kompilacją, jeżeli jest potrzebna, ustawieniem zmiennych środowiskowych, instalacją zależności, przygotowaniem bazy produkcyjnej, itd.),
5) w przypadku przekazywania kodu źródłowego Systemu, musi być on przekazany w taki sposób, aby było możliwe umieszczenie kodu w lokalnym repozytorium GitLab Zamawiającego.

Pracownik Centrum Obsługi Informatycznej założy repozytorium kodu w systemie GitLab, zarządzanym przez COI oraz wklei do niego kod źródłowy Systemu. Każdorazowo po przekazaniu aktualizacji kodu źródłowego przez Wykonawcę, pracownik Centrum Obsługi Informatycznej będzie aktualizować kod w repozytorium kodów źródłowych.

Powyższe informacje powinny zawierać wskazanie:

1) wersji i dystrybucji wszystkich niezbędnych komponentów,
2) sposobu instalacji bibliotek i dodatków,
3) sposobu ustawiania parametrów i zmiennych środowiskowych.

W przypadku wykorzystania komponentów lub oprogramowania firm trzecich wymagających osobnych licencji, konieczne jest wskazanie tych komponentów i niezbędnego zakresu licencji.

W szczególnych przypadkach, na potrzeby weryfikacji przekazanych kodów źródłowych oraz instrukcji im towarzyszących, dopuszczone jest stworzenie środowiska do kompilacji, przy czym sam proces kompilacji będzie wykonywany przez pracownika UMK (i/lub w asyście firmy zewnętrznej).

W celu dokonania weryfikacji kompletności i czytelności kodu źródłowego, w obecności Zamawiającego Wykonawca ma dokonać kompilację i sprawdzenie poprawności działania kodu źródłowego lub należy ustalić inny tryb weryfikacji kodu źródłowego.

Kody źródłowe przekazane przez Wykonawców są przechowywane w szafie pancernej COI UMK oraz w repozytorium GitLab.