CZĘŚĆ II – OPIS PRZEDMIOTU ZAMÓWIENIA

1. **Nazwa zamówienia**
   Konwersja posiadanego oprogramowania DynaTrace oraz dostawa nowych licencji DynaTrace lub równoważnych wraz z zapewnieniem wsparcia technicznego dla całego oprogramowania.

2. **Kod CPV:**
   72250000 -2 Usługi w zakresie konserwacji i wsparcia systemów,
   48000000 - 8 Pakiety oprogramowania i systemy informatyczne.

3. **Przedmiot zamówienia:**

3.1. Przedmiotem zamówienia jest:

3.1.1. zmiana metryki licencyjnej polegająca na konwersji posiadanych przez Zamawiającego licencji na oprogramowanie DynaTrace Application Monitoring:

	Nazwa licencji na oprogramowanie posiadanych przez Zamawiającego	**Liczba licencji**
3.1.1.1.	DynaTrace AppMon Agent Unit	200
3.1.1.2.	DynaTrace User Experience Management – 100M visits	1

do modelu subskrypcyjnego, tj. na następujące licencje:

3.1.1.3. 284 szt. licencji Host Unit Offline (licencja subskrypcyjna na okres 36 miesięcy)

3.1.1.4. 8 500 000 DEM Units Offline (licencja subskrypcyjna na okres 36 miesięcy);

3.1.2. dostawa subskypcji na okres 36 miesięcy:

3.1.2.1. na oprogramowanie DynaTrace:

3.1.2.1.1. 92 szt. Host Unit Offline;

3.1.2.1.2. 16 500 000 DEM Units Offline;

3.1.2.2. lub na oprogramowanie równoważne zgodnie z pkt 5 OPZ;

3.1.3. zapewnienie wsparcia technicznego dla oprogramowania wskaznego w pkt. 3.1.1.3 i 3.1.1.4 oraz w pkt. 3.1.2 (zwanego dalej: „**Oprogramowaniem**”) na okres 36 miesięcy.

4. Wymagania ogólne

4.1. Oprogramowanie musi pochodzić z legalnych źródeł oraz zostać dostarczone Zamawiającemu ze wszystkimi dokumentami i danymi niezbędnymi do potwierdzenia legalności jego pochodzenia, do jego uruchomienia lub korzystania z niego (np.: dane dostępowe, certyfikat autentyczności, kod aktywacyjny wraz z instrukcją aktywacji, itp.).

4.2. W przypadku dostawy licencji na oprogramowanie równoważne, o których mowa w pkt. 3.1.2.2., oprogramowanie to musi spełniać opisane w punkcie 5 kryteria stosowane w celu oceny równoważności.

4.3. Instalacja licencji nie może wymagać dostępu do internetu z serwera, na którym będzie instalowane oprogramowanie.

5. Licencje na oprogramowanie równoważne

Licencje na oprogramowanie równoważne muszą spełniać poniższe wymagania Zamawiającego:

5.1. Kluczowe elementy:

1) Oferowane oprogramowanie musi być rozwiązaniem standardowym, powszechnie dostępnym, gwarantującym ciągłość producenta oraz zapewniające dostępność bazy wiedzy jego dotyczącej.
2) Oferowane oprogramowania musi mieć możliwość pracy w tzw. klastrze, który pozwala na równoważenie obciążenia oraz zapewnienie wysokiej dostępności (HA).
3) Elementy oferowanego rozwiązania muszą w zakresie komunikacji (wewnętrznej i zewnętrznej) wykorzystywać protokoły SSL/TLS.
4) Wszystkie komponenty oprogramowania muszą posiadać gotową w momencie złożenia oferty dokumentację w języku polskim lub angielskim. Dokumentacja powinna być publicznie i powszechnie dostępna np. przez strony www producenta, w momencie złożenia oferty.
5) Oprogramowanie musi posiadać możliwość uruchomienia monitoringu dla aplikacji pracujących na:
   - na platformach sprzętowych: Intel/AMD x86_64, x86
   - w systemach operacyjnych: zgodnym z Linux / Windows Server 2008+
6) Oprogramowanie musi wspierać możliwość automatycznych i ręcznych aktualizacji oprogramowania każdego komponentu muszą mieć możliwość automatycznego pobierania na serwer realizowane przez administratora systemu, a zarzadzanie procesem aktualizacji musi odbywać się z poziomu centralnej konsoli. Aktualizacje nie mogą wymagać restartu systemu.
7) Oprogramowanie musi umożliwiać maskowanie danych wrażliwych, definiowanych/wskazanych przez administratora systemu np. dane osobowe, adresy IP użytkowników, identyfikatory.
8) Oprogramowanie musi umożliwiać maskowanie nazw operacji w przypadku, gdy mogą one zawierać dane osobowe (dotyczy to także operacji wykonywanych na stronie WWW),
9) Oprogramowanie musi zapewniać, aby dostęp do oprogramowania był zabezpieczony hasłem.
10) Oprogramowanie musi pozwalać na zabezpieczenie dostępu z poziomu operatora i użytkownika oprogramowania za pomocą protokołu HTTPS.
11) Udostępnianie danych do raportów i dashboardów w dokładnie takim samym formacie i o takiej samej strukturze jak oprogramowanie referencyjne.

5.2. Kryteria licencjonowania

1) Licencje wymagane do pracy systemu, muszą pracować w trybie „pływającym” (ang. Floating licenses), co oznacza możliwość wykorzystania ich w wielu systemach jednocześnie. W takim wypadku systemy, niemonitorowane, pomimo zainstalowania oprogramowania monitorującego (np. agenta), nie zużywają licencji. Oprogramowanie centralne powinno umożliwiać centralne zarządzanie wyłączaniem (i zwalnianiem licencji) monitoringu dla systemów, gdzie takie monitorowanie ma być wstrzymane.
2) Licencje na oprogramowanie nie może ograniczać liczby użytkowników końcowych np. administratorów, operatorów korzystających z oprogramowania do monitoringu ani nie mogą ograniczać ilości przetwarzanych danych pobieranych z monitorowanych systemów.
3) Oprogramowanie musi zapewniać przenaszalność licencji miedzy monitorowanymi środowiskami bez konieczności wykupywania dodatkowych licencji.
4) Dostarczane licencje nie mogą ograniczać liczby monitorowanych procesów JAVA oraz .NET działających w środowiskach, dla których uruchomiono monitoring.
5) Oprogramowanie musi umożliwiać monitoring serwerów zarówno w lokalnym centrum danych, jak i w centrach danych chmurowych (co najmniej AZURE).
6) Oprogramowanie musi rejestrować (automatycznie oraz poprzez API) wgrywanie nowych wersji aplikacji na serwery aplikacyjne. Informacje te muszą być dostępne przy analizie wykrytych problemów bezpośrednio przez interfejs graficzny narzędzia.
7) Oprogramowanie musi pozwalać na wyłączanie/włączanie monitoringu na poszczególnych serwerach z poziomu centralnej konsoli narzędzia, bez konieczności ręcznej modyfikacji konfiguracji agentów po stronie serwerów monitorowanych oraz bez konieczności zmiany konfiguracji samych serwerów monitorowanych i zapewniać tym samym zwolnienie licencji, z których wyłączone serwery korzystały, bez konieczności zmian konfiguracyjnych w środowisku monitorowanym.
8) Oprogramowania musi umożliwiać dostęp do logów aplikacyjnych i systemowych, przeszukiwanie ich i przeglądanie bez konieczności logowania na serwer monitorowany.
9) Oprogramowani musi być dostarczony wraz z wszystkimi komponentami koniecznymi do jego uruchomienia, działania i bezterminowego korzystania z wszystkich wymienionych i opisanych w tym dokumencie funkcjonalności. Oprogramowania musi posiadać funkcjonalność skalowania infrastruktury. W oporgramowaniu musi istnieć mechanizm dodawania nowych węzłów pozwalających na zwiększenie wydajności dziennika.
10) Oprogramowanie nie może posiadać ograniczenia z uwagi na liczbę pracujących użytkowników systemu (dot. ograniczeń licencyjnych i poza licencyjnych).
11) Oprogramowanie musi zapewniać kontrolę dostępu do gromadzonych danych bazującą na metodzie RBAC (Role Based Access Control).
12) Oprogramowanie nie może być limitowany na ilość zdarzeń w czasie.

13) Oprogramowanie musi posiadać wsparcie producenta, realizowane w języku polskim. Wsparcie nie może być limitowane liczbą zgłoszeń lub osób zgłaszających problem ze strony Zamawiającego.
14) Wszystkie funkcjonalności oprogramowania u muszą być dostarczone w ramach jednej licencji od jednego producenta.
15) Oprogramowanie musi umożliwiać automatyczne sprawdzanie dostępności aplikacji poprzez wykonywanie skryptu symulującego pracę użytkownika. Przygotowanie skryptu nie może wymagać od użytkownika umiejętności programistycznych, musi wykorzystywać mechanizm record-replay.

5.3. Wsparcie technologiczne

1) Oprogramowanie musi zapewniać monitorowanie wielowarstwowych aplikacji wykonanych w technologii Java, .NET i PHP, działające na serwerach aplikacyjnych takich jak JBoss, WebSphere, WildFly, Tomcat, IIS oraz innych zgodnych z technologią J2EE, a także aplikacje oparte o WCF.
2) Oprogramowanie musi wykrywać i monitorować przebieg wszystkich transakcji wykonanych przez aplikację bez potrzeby definiowania zależności pomiędzy komponentami (np. usługami, warstwami), a tak wykryte przebiegi mają być prezentowane w formie graficznej w celu ich łatwej analizy, z uwzględnieniem komponentów takich jak serwery WWW, API, aplikacyjne, bazodanowe, aplikacje klienckie (przeglądarka, urządzenie mobilne).
3) Oprogramowanie musi umożliwiać definiowanie własnych komponentów (np. warstwy czy usług), które zostaną włączone w reprezentację graficzną poprzez wskazanie punktu startowego – np. metody wykonywanego kodu. Taki komponent powinien być raportowany w ramach pełnego przebiegu transakcji oraz niezależnie od tego przebiegu.
4) Oprogramowanie musi automatycznie tworzyć linie bazowe dla wszystkich zbieranych metryk, w tym dotyczących wydajności aplikacji (czasy odpowiedzi, procent błędów) dla poszczególnych żądań http czy wywołań usług sieciowych, poszczególnych zapytań bazodanowych, danych infrastrukturalnych (co najmniej obciążenie procesora, wykorzystanie pamięci, wydajność i opóźnienie dysków, zajętość miejsca), parametrów sieciowych (generowany ruch, liczba pakietów, liczba pakietów odrzucanych, retransmisje).
5) Tworzone w oprogramowaniu linie bazowe muszą uwzględniać zmienność w czasie. System monitoringu na bazie linii bazowych musi automatycznie wykrywać odstępstwa od normy (przekroczenie wartości normalnych) i generować alerty. Operator musi mieć możliwość określenia dopuszczalnych poziomów odstępstw od normy na poziomie wartości lub procentu odchylenia lub sztywnych progów.
6) oprogramowanie musi monitorować i śledzić przebieg wszystkich wykonywanych transakcji pomiędzy wszystkimi warstwami aplikacji w środowisku z możliwością uzyskania następujących informacji o każdej z pojedynczych transakcji:
    - drzewo wywołania kodu Java i .NET w ramach ścieżki wykonania – do poziomu nazwy wywoływanej metody, zarówno dla wątków wywoływanych synchronicznie, jak i asynchronicznie,
    - czasach odpowiedzi serwera do aplikacji klienckiej, jak i całkowitym czasie wykonania transakcji po stronie serwera (wątków synchronicznych oraz asynchronicznych),
    - zapytań SQL wykonanych w ramach transakcji z możliwością uzyskania informacji o liczbie zwróconych wierszy,
    - wartości parametrów metody JAVA lub .NET, nagłówków http, parametrów zapytań http.
    - wszelkich zapytania SQL wykonywane z poziomu monitorowanej aplikacji z możliwością ich powiązania z transakcjami, które dane zapytania wykonują,
7) Oprogramowanie musi wspierać śledzenie wszystkich transakcji pomiędzy różnymi warstwami architektury, które wykorzystują następujące technologie synchroniczne i asynchroniczne: HTTP, REST, SOAP/XML, JMS.
8) Oprogramowanie musi automatycznie monitorować zarówno transakcje inicjowane działaniami użytkowników (żądania http, wywołania usług sieciowych), jak i wynikające z działania kodu w tle na serwerach aplikacyjnych.
9) Oprogramowanie musi zapewniać monitorowanie serwerów webowych Apache oraz IIS w zakresie wprowadzanych przez nie opóźnień w czasie realizacji transakcji webowej oraz błędów pojawiających się na tychże serwerach. Monitoring musi pokazywać wpływ poszczególnych modułów działających na serwerach WWW na czasy wykonania transakcji,
10) Oprogramowanie musi umożliwiać wykonywanie zrzutów pamięci ze stosu Java w formacie umożliwiającym jej analizę co najmniej pod względem „wycieków” i optymalizacji,
11) Oprogramowanie musi oferować mechanizm oznaczania wybranych transakcji jako kluczowych, najbardziej istotnych, z uwzględnieniem co najmniej poniższych kryteriów:

- URL, Wartość parametru z nagłówka HTTP,
- Wartość parametru z zapytania GET lub POST,
- Wykonanie konkretnej metody i jej parametru w kodzie Java/.NET, d. wywołanie konkretnej usługi API (Webservice).

12) Oprogramowanie musi umożliwiać monitoring aplikacji uruchamianych w środowisku skonteneryzowanym Docker na poziomie zarówno pojedynczych kontenerów jak i serwera, na którym są uruchamiane. Konfiguracja monitoringu musi być zautomatyzowana, bez konieczności konfigurowania poszczególnych obrazów kontenerów.

13) Oprogramowanie musi umożliwiać automatyczne monitorowanie aplikacji skonteneryzowanych, uruchamianych w środowisku Kubernetes/Openshift na poziomie analizy kodu uruchamianego wewnątrz pod-ów bez konieczności modyfikacji pod-ów.
14) Monitoring nowych pod-ów powinien być uruchamiany automatycznie, bez potrzeby ręcznej konfiguracji.
15) Kolejne węzły klastra Kubernetes/Openshift powinny być podłączane do monitoringu automatycznie, bez konieczności dodatkowej konfiguracji monitoringu.
16) Administrator powinien mieć możliwość wybrania konkretnych typów węzłów oraz konkretnych węzłów, na których monitoring będzie uruchomiony.
17) Administrator musi mieć możliwość na których pod-ach monitoring będzie włączony lub wyłączony z konsoli WWW narzędzia.
18) Oprogramowanie musi umożliwiać monitoring klastra Kubernetes/Openshift poprzez API, w szczególności w zakresie wykorzystania zasobów (CPU, pamięć), żądań o zasoby, limitów oraz udostępniać informacje o zdarzeniach na klastrze.
19) Oprogramowanie musi przyjmować dane z systemów zewnętrznych co najmniej w formacie statsd.

5.4. Funkcjonalność i zarządzanie

1) Oprogramowanie musi posiadać możliwość prezentowania na wykresach metryk zbieranych przez narzędzie i umieszczać je na pulpitach informacyjnych,
2) Oprogramowanie musi wspierać gromadzenie danych zagregowanych o czasach przetwarzania na potrzeby wykonywania raportów dotyczących poziomu usług (np. SLA, OLE),
3) Oprogramowanie musi udostępniać dane zbieranych metryk o prognozowanych wartościach w przyszłości, za wybrany okres czasu,
4) Oprogramowanie musi pozwalać na tworzenie wewnątrz narzędzia dowolnych, niestandardowych wykresów i pulpitów informacyjnych poprzez interfejs graficzny. Elementy pulpitów informacyjnych muszą być powiązane z danymi źródłowymi i umożliwiać szybkie przejście do ekranów pozwalających na analizę charakterystyk źródła danych,
5) Oprogramowanie musi posiadać własny interfejs do tworzenia lub konfigurowania własnych wtyczek monitorujących rozszerzających standardowe funkcjonalności narzędzia,
6) Oprogramowanie musi zapewnić mechanizmy bezpieczeństwa w zakresie dostępu do zbieranych danych (w postaci argumentów metod) – narzędzie musi gwarantować odpowiedni poziom dostępu do danych definiowany na poziomie nadawania uprawnień w aplikacji do monitorowania,
7) Oprogramowanie musi logować wszystkie aktywności użytkowników związane ze zmianami konfiguracji. Logowanie musi umożliwiać jednoznaczne wskazanie osoby, która wykonała zmianę.
8) Oprogramowanie musi umożliwiać korelację danych transakcyjnych z odpowiadającymi im danymi infrastrukturalnymi, bazodanowymi, mechanizmami sesji.
9) Oprogramowanie musi zapewniać możliwość wyodrębnienia oddzielnej prezentacji ruchu czy przebiegu sesji pochodzących od różnych aplikacji, umożliwiać tworzenie własnych definicji aplikacji jako punktu interakcji rzeczywistego użytkownika z aplikacją web-ową lub mobilną,
10) Oprogramowanie musi umożliwiać porównywanie działania aplikacji w różnych przedziałach czasowych na poziomie czasów odpowiedzi, liczby błędów oraz wykrytych elementów krytycznych, mających największy udział w czasach wykonania (np. czas wywołania usług zewnętrznych, czas odpowiedzi bazy danych, czas sieci),
11) Oprogramowanie musi umożliwiać monitoring połączeń między poszczególnymi procesami, serwerami i udostępniać automatycznie budowaną mapę połączeń w formie graficznej. W przypadku wykrycia anomalii skutkującej wygenerowaniem alertu serwer musi zostać oznaczony na wizualizacji w sposób jednoznacznie wskazujący na wystąpienie problemu.
12) Oprogramowanie musi umożliwiać monitoring wszystkich procesów działających na serwerach objętych monitoringiem na poziomie co najmniej metryk systemowych: wykorzystanie CPU, pamięci, operacji IO, wykorzystanie sieci. Dla procesów JAVA i .NET dodatkowo monitoring musi obejmować statystyki wykorzystania wątków, sterty JVM i CLR oraz wpływ działania Garbage Collector'a na proces. Nowe procesy muszą być wykrywane i monitorowane automatycznie bez potrzeby ręcznej konfiguracji.
13) Oprogramowanie musi oferować monitoring podstawowych parametrów systemowych (co najmniej monitorować zajętość CPU, zajętość pamięci, zajętość dysków, utylizacja interfejsów sieciowych), komponentów środowiska aplikacyjnego – zarówno tych, na których działają serwery aplikacyjne Java i .NET, jak i innych (np. serwery bazodanowe, serwery Nginx, serwery Apache, IIS działające w ramach systemów operacyjnych Linux/Windows).
14) Oprogramowanie musi zbierać informacje o wszystkich wyjątkach oraz błędach http. Musi istnieć możliwość zobaczenia szczegółowych informacji na temat transakcji, które wygenerowały wyjątek lub błąd http.
15) Oprogramowanie musi oferować możliwość konfiguracji wyjątków, czyli odstępstw od reguły, pozwalających na utworzenie reguł odrzucających błędy techniczne, które nie mają wpływu na biznesowe działanie aplikacji.

16) Oprogramowanie musi automatycznie, na podstawie danych bazowych/wzorcowych wykrywać problemy związane co najmniej z:
    - wydłużeniem czasów odpowiedzi poszczególnych usług po stronie serwerowej,
    - zwiększeniem poziomu błędów dla poszczególnych usług po stronie serwerowej,
    - wydłużeniem czasów odpowiedzi dla poszczególnych akcji wykonywanych przez użytkownika końcowego na aplikacji WWW lub aplikacji mobilnej,
    - zwiększeniem poziomu błędów (w tym błędów JavaScript) dla poszczególnych akcji wykonywanych przez użytkownika końcowego na aplikacji WWW lub aplikacji mobilnej,
    - przeciążeniem CPU,
    - nadmiernym wykorzystaniem pamięci,
    - spadkiem wydajności dysków,
    - dostępnością na warstwie TCP,
    - wzrostem liczby zgubionych pakietów (w powiązaniu interfejsu sieciowego z danym procesem),
    - spadkiem ruchu w aplikacjach lub usługach (zmniejszenie liczby wywołań),
    - brakiem dostępności aplikacji,
    - niedostępnością procesu monitorowanego.
17) Ogramowanie musi zapewniać w przypadku wykrycia zdarzeń alertowych (przekroczenie progów alarmowych dla poszczególnych metryk) oprogramowanie musi przeciwdziałać multiplikowaniu alarmów dotyczących tego samego problemu i grupować je na podstawie zależności między serwerami, procesami i usługami oraz posiadać mechanizmy konfigurowania zasad eskalacji.
18) Oprogramowanie musi w przypadku wykrycia problemu wskazać ilu użytkowników i ile wywołań usług zostało dotkniętych problemem.
19) Oprogramowanie musi w przypadku wykrycia problemu automatycznie wskazać najbardziej prawdopodobną przyczynę wystąpienia problemu.
20) Oprogramowanie musi umożliwiać definiowanie alarmów na podstawie wykrycia charakterystycznych wpisów w logach systemu operacyjnego lub monitorowanych aplikacji.
21) Oprogramowanie musi umożliwiać monitoring błędów po stronie przeglądarki (np. błędy javascript) lub urządzenia mobilnego.
22) Oprogramowanie musi umożliwiać automatyczne sprawdzanie dostępności aplikacji poprzez wykonywanie skryptu symulującego pracę użytkownika. Przygotowanie skryptu nie może wymagać od użytkownika umiejętności programistycznych, musi wykorzystywać mechanizm record-replay.
23) Oprogramowanie musi umożliwiać korelację wszystkich działań użytkownika na stronie WWW – dla każdej interakcji i każdej sesji, co umożliwi znalezienie każdego użytkownika i diagnostykę jego interakcji z każdym komponentem systemu.
24) Oprogramowanie musi umożliwiać analizę zachowań użytkownika dla co najmniej miliona co najmniej godzinnych sesji rocznie. Sesja jest rozumiana jako co najmniej dwukrotna interakcja użytkownika z przeglądarką/urządzeniem mobilnym.
25) Oprogramowanie musi umożliwiać powiązanie każdej sesji z interakcją z systemem i transakcjami realizowanymi przez system na poziomie sekwencji wywołanych metod i skorelowanych informacji infrastrukturalnych, od rozpoczęcia aktywności (np. dostęp do strony WWW), aż do jej zakończenia (np. odpowiedź bazy danych).
26) Oprogramowanie musi zapewniać, że monitorowanie pracy użytkownika końcowego (user experience) nie może wymagać instalacji dodatkowych komponentów po stronie użytkownika i nie może wymagać zmian konfiguracji serwerów WWW lub aplikacyjnych. Zmiana konfiguracji czy też włączenie/wyłączenie monitorowania zachowań użytkownika musi odbywać się z centralnego komponentu do zarządzania systemem monitorującym, w sposób automatyczny oraz bez potrzeby restartu serwerów monitorowanego środowiska.
27) Oprogramowanie musi umożliwiać dla każdego rodzaju interakcji zbieranie metryk, także biznesowych, pozwalających na ocenę jak poszczególne kanały interakcji /mobilne, www, itp./ wykorzystywane są do realizacji zadań biznesowych. Dane te muszą zapewniać możliwość opracowania zestawień statystycznych dotyczących między innymi czasu spędzanego w poszczególnych elementach procesu biznesowego, map przejścia procesów biznesowych, itd. Zagregowane dane muszą umożliwić identyfikację obszarów procesów biznesowych, których obsługa trwa najdłużej. Źródłem danych do zbieranych metryk biznesowych mogą być wartości tagów html, atrybuty elementów, selektory css, zmienne javascript, metadane.
28) Oprogramowanie musi umożliwiać dostarczanie informacji nt. charakteru każdej interakcji w systemie dla każdego pojedynczego użytkownika (np. landing pages, bounces oraz ładowania asynchroniczne AJAX).
29) Oprogramowanie musi umożliwiać wyświetlenie informacji o wykonywanych akcjach w sposób wykorzystujący metryki W3C Timing i wizualizując (jako waterfall) żądania, które zostały wysłane podczas interakcji użytkownika z aplikacją.
30) Oprogramowanie musi dawać możliwość wyświetlenia informacji dla każdego wybranego użytkownika, który wszedł w interakcję z systemem.

5.5. Zarządzanie systemem opartym na oferowanym oprogramowaniu

1) Oprogramowaniem musi posiadać rozbudowany mechanizm nadawania uprawnień i tworzenia ról dla poszczególnych funkcjonalności systemu, tak aby można było w łatwy sposób udostępniać dane jedynie dla wybranych elementów architektury (pojedyncze serwery i procesy) i dla poszczególnych środowisk.
2) Oprogramowanie musi posiadać mechanizm separacji monitorowanych środowisk na poziomie systemu zarządzającego (ang. Multitnenant), zarówno na poziomie osobnych środowisk (separacja różnych, niezależnych systemów IT), jak i osobnych obszarów w ramach jednego środowiska (np. separacja środowisk testowych od produkcyjnych).
3) Oprogramowanie musi umożliwiać zarządzenie uprawnieniami do modyfikacji wytworzonych w systemie obiektów (wyszukiwania, wizualizacje, pulpity informacyjne).
4) Oprogramowanie musi logować wszystkie aktywności użytkowników związane ze zmianami konfiguracji. Logowanie musi umożliwiać jednoznaczne wskazanie osoby, która wykonała zmianę.
5) Oferowana przez oprogramowanie strona www musi być zabezpieczona hasłem.
6) Oprogramowanie musi wspierać pracę wielu użytkowników, posiadających własne loginy i hasła.
7) Oprogramowanie musi posiadać panel zarządzania użytkownikami dostępny z poziomu GUI.
8) Oprogramowanie musi udostępniać funkcjonalność podziału uprawnień pomiędzy użytkownikami, co pozwoli na ograniczenie dostępu użytkowników do poszczególnych zgromadzonych danych. W systemie musi być wprowadzony mechanizm bezpieczeństwa, który kontroluje jaki użytkownik ma dostęp do jakiego fragmentu danych. Mechanizm konfiguracji uprawnień musi być dostępny w poziomu www aplikacji, jak i z API. Wprowadzone ograniczenia dostępu muszą pracować jednolicie dla operacji wyszukiwania danych z GUI, jak i z poziomu API.

5.6. Kolekcjonowanie danych i monitoring logów

1) Oprgramowanie musi umożliwiać zbieranie danych przesyłanych przez systemy operacyjne (Linux/Windows), urządzenia sieciowe oraz aplikacje,
2) Oprogramowanie musi umożliwiać zbieranie danych ekstrahowanych z plików logów umieszczonych w wyszczególnionym katalogu na serwerze z oprogramowaniem,
3) Oprogramowanie musi wspierać indeksowanie i przetwarzanie nieokreślonej liczby zdarzeń w ramach wydajności pojedynczego serwera, na którym pracuje. Dostarczona licencja musi zezwalać na takie wykorzystanie oprogramowania, retencja danych powinna być konfigurowalna,
4) Oprogramowanie musi posiadać wbudowany mechanizm kompresji przechowywanych danych,
5) Oprogramowanie musi umożliwiać na odczyt i analizę (ang. parsing) przesyłanych danych,
6) Oprogramowanie musi umożliwiać na pracę z logami jednolinijkowymi oraz wielolinijkowymi,
7) Oprogramowanie musi umożliwiać na rozpoznanie formatów czasu i daty i normalizowanie ich do jednego wspólnego formatu,
8) Oprogramowanie musi umożliwiać operatorowi samodzielne konfigurowanie reguł odczytu i analizowani (ang. parsing) nieznanych formatów logów w celu umożliwienia analizy zebranych w nich informacji przez opisywane oprogramowanie.
9) Oprogramowanie musi umożliwiać dostęp do logów aplikacyjnych i systemowych, przeszukiwanie ich i przeglądanie bez konieczności logowania na serwer monitorowany,
10) Oprogramowanie musi być obsługiwany z poziomu przeglądarki internetowej,
11) Oprogramowanie musi pozwalać na wyszukiwanie w całym zgromadzonym przez nie zbiorze danych,
12) Oprogramowanie musi pozwalać na równoległe wyszukiwanie, z poziomu jednego interfejsu graficznego, w danych napływających w czasie rzeczywistym i danych historycznych,
13) Oprogramowanie musi pozwalać na użycie operatorów logicznych, wzorców, wyrażeń regularnych (REGEX) do przeszukiwania danych,
14) Oprogramowanie musi pozwalać na przeszukiwania w ograniczonym zbiorze danych (np. ze względu na zakres dat wystąpienia), wraz z wyświetleniem wyszukiwanych wyników przedstawiać na tym samym ekranie związaną z tym wyszukiwaniem statystykę ilościową w dziedzinie czasu,
15) Oprogramowanie musi pozwalać na oznaczanie (tag) i korelacje danych (pod kątem dowolnej z przechowywanych wartości) bez względu na źródło tych danych,
16) Oprogramowanie musi wraz z prezentacją (wykres, tabela) zestawień zbiorczych, statystycznych, itd. musi pozwalać na obejrzenie danych źródłowych w ich oryginalnym formacie (logi) na podstawie których powstała ta prezentacja,
17) Oprogramowanie musi pozwalać na geolokalizację zdarzeń na bazie adresów IP oraz umożliwiać jej wizualizacje na mapie.

18) Oprogramowanie musi posiadać wbudowaną funkcjonalność nauczania maszynowego do celów predykcji wartości badanych parametrów. W szczególności oprogramowanie musi posiadać możliwość stosowania algorytmów nauczania takich jak: sieć neuronowa, zmienne drzewa decyzyjne, regresja liniowa.
19) Musi istnieć możliwość wyszukiwania na podstawie jednoczesnego wykorzystania więcej niż jednego zdefiniowanego wzorca wyszukiwania.
20) Musi istnieć możliwość zapisania wyników wyszukiwania i tworzenia z nich raportów.
21) Oprogramowanie musi oferować także udokumentowany interfejs programistyczny (API) służący do przeszukiwania danych gromadzonych w czasie rzeczywistym oraz danych historycznych.
22) Oprogramowanie musi oferować pełne sterowanie pracą dziennika zdarzeń poprzez API.
23) Oprogramowanie musi pozwalać na eksport wyszukanych danych do formatu CSV. Funkcja eksportu musi pozwalać na pracę z co najmniej kilkuset megabajtowymi porcjami danych pochodzącymi z różnych okresów czasowych.
24) Funkcja eksportu (co najmniej do CSV) musi być dostępna z poziomu graficznego interfejsu użytkownika.
25) Oprogramowanie musi umożliwiać tworzenie raportów z predefiniowanych kryteriów wyszukiwania w postaci tabelarycznej i graficznej (minimum 3 typy wykresów: liniowy, słupkowy, kołowy),
26) Oprogramowanie musi umożliwiać jednoczesne użycie wielu predefiniowanych kryteriów wyszukiwania w celu opracowania raportu, w taki sposób, że: tworzony raport musi mieć możliwość automatycznej dystrybucji raz w tygodniu lub miesiącu,
27) Oprogramowanie musi udostępniać API (co najmniej w formacie JSON) pozwalające na integracje z innymi narzędziami służącymi do zarządzania i analizy danych

6. Termin realizacji zamówienia

6.1. Wykonawca dokona zmiany metryki licencyjnej, o której mowa w pkt. 3.1.1., oraz zrealizuje dostawę, o której mowa w pkt. 3.1.2., w terminie do 10 dni roboczych od dnia zawarcia Umowy (stanowi kryterium oceny).

6.2. Wykonawca zapewni wsparcie techniczne dla Oprogramowania na okres 36 miesięcy od dnia realizacji przedmiotu zamówienia wskazanego w pkt. 3.1.1. i 3.1.2. potwierdzonej podpisanym przez obie strony protokołem odbioru.

7. Wymagania dotyczące usługi wsparcia technicznego dla Oprogramowania

7.1. Zamawiający wymaga zapewnienia przez Wykonawcę poziomu wsparcia określonego poniżej:

7.1.1. Dostęp do dokumentacji i przewodników po najlepszych praktykach.

7.1.2. Oprogramowanie równoważne musi być aktualizowane w tych samych terminach i zakresie co oprogramowanie referencyjne.

7.1.3. Pełen dostęp do społeczności producenta Oprogramowania, tj. do internetowego portalu pod adresem producenta. Wymagane funkcje portalu:

7.1.3.1. Zadawanie pytań na forach produktów.
7.1.3.2. Dostęp do dokumentacji produktu.
7.1.3.3. Przeszukiwanie bazy wiedzy.
7.1.3.4. Wyszukiwanie i pobieranie poprawek znanych problemów.
7.1.3.5. Zamawianie lub pobranie nowych wersji produktów.
7.1.3.6. Otwieranie zgłoszenia pomocy technicznej, jeśli nie można znaleźć rozwiązania w społeczności.
7.1.3.7. Dostęp do bezpłatnych modułów szkoleniowych (elearning).

7.1.4. Cała pomoc techniczna będzie prowadzona w języku angielskim lub polskim.

7.1.5. Wykonawca jest zobowiązany do zapewnienia skutecznego i niezawodnego wsparcia dla zaoferowanych przez siebie oprogramowania. Obejmuje to wczesne informowanie o planowanym zakończeniu życia danej wersji, a także zapewnienie zarządzanych ścieżek aktualizacji.

7.1.6. Dostęp do aktualizacji oprogramowania i poprawek, w tym aktualizacje dokumentacji produktu.

7.2. W ramach wsparcia technicznego Wykonawca zapewni wyróżnienie następujących kategorii błędów:

7.2.1. Krytyczny – nieprawidłowe działanie Oprogramowania powodujące albo całkowity brak możliwości korzystania z Oprogramowania, albo takie ograniczenie możliwości korzystania z niego, że przestaje ono spełniać swoje podstawowe funkcje.

7.2.2. Wysoki - Znacząco ograniczone używanie w operacjach biznesowych tj. korzystania z oprogramowania, okresowe lub częściowe przestoje lub podstawowe funkcje nie są już dostępne.

7.2.3. Średni – można kontynuować działalność biznesową, produkt jest stabilny, ale funkcjonalność o średnim lub niskim wpływie nie jest dostępna lub nie działa zgodnie z oczekiwaniami.

7.2.4. Niski – inne drobne błędy Oprogramowani , błędy w dokumentacji lub inne problemy o niskim priorytecie.

7.3. Wykonawca zapewni następujące czasy reakcji (tj. czas od momentu zgłoszenia błędu Zamawiającego do Wykonawcy do momentu wszczęcia przez Wykonawcę działań mających na celu usunięcie błędu) w zależności od kategorii błędu (przez „dni robocze" rozumie się dni od poniedziałku do piątku z wyłączeniem dni wolnych od pracy na terenie Rzeczypospolitej Polskiej, przez „godziny robocze" rozumie się godziny od 9 do 17 w dni robocze):

7.3.1. Krytyczny - 4 godziny robocze

7.3.2. Wysoki - Następny dzień roboczy

7.3.3. Średni - 2 dni robocze

7.3.4. Niski - 4 dni robocze

7.4. Wykonawca może realizować wsparcie za pośrednictwem producenta Oprogramowania lub autoryzowanego podmiotu współpracującego z producentem.

8. Zamawiający zastrzega, że przedmiot Umowy jest przeznaczony do dalszej odsprzedaży. Wszelkie dokumenty rejestracyjne, licencyjne itp. muszą być wystawione na docelowego użytkownika licencjobiorcę, jakim będzie Kancelaria Premiera Rady Ministrów.

9. Pozostałe warunki realizacji zamówienia zawarte są w części III SWZ – projektowane postanowienia umowy.