Specyfikacja Oprogramowania

Przedmiotem zamówienia jest sprzedaż, dostarczenie, konfiguracja, uruchomienie oraz dostawa licencji czasowych w modelu subskrypcyjnym oprogramowania w rozwiązaniu SaaS, służącego do monitorowania infrastruktury aplikacyjnej Zamawiającego, składającego się na oprogramowanie klasy APM (ang. Application Performance Monitoring) na okres 36 miesięcy wraz z Usługą Wsparcia Oprogramowania na okres trwania subskrypcji. Na powyższe w szczególności składa się:

- przeprowadzenie wdrożenia wyżej wymienionego oprogramowania w środowisku IT ARiMR (on premise i cloud) oraz dostosowanie portalu zarządzającego systemem (dedykowane dashboardy, tresholdy), w tym wykonanie i dostarczenie Projektu Technicznego, Planu Testów Akceptacyjnych, dokumentacji powykonawczej oraz dokumentacji eksploatacyjnej w formie elektronicznej;

- zapewnienie Zamawiającemu prawa do korzystania z Oprogramowania wraz z jego dokumentacją na zasadzie Subskrypcji, przez 36 miesięcy

- Przeprowadzenie warsztatów powdrożeniowych (w formie zdalnej) z działania i obsługi Oprogramowania (dla administratorów - max 16 godzin i kluczowych użytkowników - max 16 godzin).

- świadczenie Gwarancji dla Oprogramowania przez okres 36 miesięcy

- zapewnienie Zamawiającemu do 450 godzin (w okresie trwania umowy) konsultacji technicznych w zakresie problemów dotyczących Oprogramowania. Konsultacje będą obejmowały bieżące prace administracyjne, konfiguracyjne, analityczne związane z Oprogramowania, wszelkiego rodzaju prace rekonfiguracyjne i optymalizujące działanie Systemu. Zamawiający gwarantuje wykorzystanie w okresie obowiązywania umowy minimum 150 godzin konsultacji technicznych.

Oprogramowanie klasy APM w rozwiązaniu SaaS, o którym mowa musi zapewniać monitorowanie zachowania i wydajności aplikacji, baz danych oraz monitorowanie doświadczeń użytkownika końcowego wraz z pełną diagnostyką działania na stacji roboczej w zakresie działania monitorowanych aplikacji i usług. Oprogramowanie musi monitorować i weryfikować wszystkie serwery aplikacyjne, bazodanowe wchodzące w skład środowiska, również dla części maszyn wirtualnych.

1. **Wymagania funkcjonalne Oprogramowania do monitorowania środowiska produkcyjnego i uat :**
   1.1. Monitorowanie aplikacji ma odbywać się w sposób ciągły, z wykorzystaniem oprogramowania monitorującego. Część centralna oprogramowania może być udostępniona w modelu SaaS.
   1.2. Rozwiązanie oparte o model SaaS musi zapewniać poziom bezpieczeństwa potwierdzony min. Certyfikatem SOC-2 type 1 lub równoważnym. Rozwiązanie powinno być zgodne z zasadami dyrektywy GDPR oraz zapewniać, że dane w SaaS będą przechowywane oraz przetwarzane na terenie Unii Europejskiej. Komunikacja pomiędzy komponentami rozwiązania w SaaS a komponentami w infrastrukturze Zamawiającego musi być szyfrowana z wykorzystaniem min. AES-128 lub równoważnym, komunikacja może się odbywać jedynie jednokierunkowo – od strony komponentów zlokalizowanych w infrastrukturze Zamawiającego do SaaS.
   1.3. Oprogramowanie musi być dostępne w postaci interfejsu graficznego z poziomu przeglądarki internetowej.
   1.4. Dostęp do Oprogramowania musi być zabezpieczony hasłem. Autentykacja i autoryzacja w Oprogramowaniu ma umożliwiać kontrolę dostępu opartą na rolach (RBAC).
   1.5. Oprogramowanie wykorzystywane dla świadczenia usługi musi umożliwiać integrację bazy użytkowników z LDAP.
   1.6. Dostęp do Oprogramowania musi zapewniać zabezpieczenie dostępu z poziomu operatora i użytkownika Oprogramowania za pomocą protokołu HTTPS.
   1.7. Elementy oferowanego rozwiązania muszą w zakresie komunikacji (wewnętrznej i zewnętrznej) umożliwiać wykorzystywanie protokołów bezpieczeństwa, przynajmniej SSL.

1.8. Oprogramowanie musi posiadać możliwość integracji z wykorzystywanym przez Zamawiającego systemem wielostopniowej autentykacji (MFA) w zakresie logowania do części centralnej systemu.

2. Funkcje Oprogramowania:

2.1. **Musi zapewniać** możliwość uruchomienia monitoringu dla aplikacji pracujących przynajmniej na następujących systemach operacyjnych:
   a. AIX
   b. Linux:
      - CentOS
      - Debian
      - Fedora
      - Red Hat Enterprise Linux
      - SUSE Linux Enterprise
      - Ubuntu
   c. Windows:
      - 2012
      - 2012 R2
      - 2016
      - 2019
      - 2022

2.2. Oprogramowanie musi zapewniać możliwość monitorowania wielowarstwowych aplikacji wykonanych w następujących technologiach:
   a. Java
   b. .Net
   c. PHP
   d. Node.js
   e. Python
   f. Go

2.3. Oprogramowanie, na podstawie wykrytych przepływów, w sposób automatyczny powinno pozwolić na odwzorowanie w formie graficznej monitorowanego systemu, obrazując powiązania i zależności monitorowanych komponentów i procesów oraz ich wzajemną komunikację, w szczególności uwzględniając takie warstwy jak serwery aplikacyjne, bazy danych, zewnętrzne serwisy i kolejki. W przypadku wykrycia odstępstwa od normy skutkującej wygenerowaniem alertu monitorowany komponent, musi zostać oznaczony na wizualizacji w sposób jednoznacznie wskazujący na wystąpienie problemu w danym miejscu.

2.4. Oprogramowanie musi wykrywać i monitorować przebieg wszystkich transakcji przepływających przez aplikację w sposób automatyczny.

2.5. Oprogramowanie musi wspierać automatyczne wykrywanie rodzajów komunikacji pomiędzy wykrytymi komponentami monitorowanych aplikacji, w tym wspieranie śledzenia transakcji wykorzystujących co najmniej następujące technologie synchroniczne i asynchroniczne:
   a. HTTP
   b. REST
   c. SOAP/XML
   d. JMS

2.6. Oprogramowanie musi oferować możliwości uzyskania następujących informacji o wybranych transakcjach:
   a. stos wywołania (ang. Stack trace)
   b. czasy wykonania transakcji
   c. zapytania SQL wykonanych w ramach transakcji

2.7. Oprogramowanie musi umożliwiać korelację transakcji realizowanych przez monitorowane komponenty z odpowiadającymi im danymi infrastrukturalnymi, bazodanowymi i sesją użytkownika końcowego.

2.8. Oprogramowanie musi udostępniać reguły powiadamiania w przypadku wykrycia problemów z wydajnością w aplikacji lub innych anomalii w oparciu o automatycznie wygenerowane linie bazowe, lub statyczne wartości.

2.9. Oferowane Oprogramowanie musi automatycznie, na podstawie danych bazowych/wzorcowych wykrywać problemy związane co najmniej z:
   a. wydłużeniem czasów odpowiedzi poszczególnych usług,
   b. zwiększeniem błędów dla poszczególnych usług,
   c. nagłym spadkiem liczby wywołań usług
   d. przeciążeniem CPU
   e. nadmiernym wykorzystaniem pamięci.

2.10. Umożliwia definiowanie, konfigurację i modyfikację reguł, na podstawie których oprogramowanie generuje alerty. Oprogramowanie musi mieć możliwość wygenerowania alertu na podstawie zadanego odchylenia danej metryki na podstawie statycznego progu.
2.11. Na podstawie wygenerowanego alertu, Oprogramowanie musi umożliwiać wykonanie automatycznie następujących akcji:
   a. Wysłanie powiadomienia do konkretnych użytkowników za pomocą wiadomości e-mail,
   b. Wysłać zapytanie HTTP o dowolnej treści na dowolny URL.
   c. Wysłać powiadomienia do Slack
   d. Wysłać powiadomienie do PagerDuty
2.12. Pozwala zbierać i monitorować najbardziej wpływające na wydajność monitorowanej aplikacji zapytania SQL wykonywane z poziomu monitorowanej aplikacji z możliwością ich powiązania z transakcjami, które dane zapytania wykonują.
2.13. Ogranicza swój wpływ na monitorowane platformy i aplikacje m.in. poprzez inteligentne zbieranie informacji celem uniknięcia zbędnego zużywania zasobów.
2.14. Ma możliwość prezentowania na wykresach dowolnych metryk gromadzonych przez oprogramowanie.
2.15. Oprogramowanie musi pozwalać na tworzenie dowolnych niestandardowych pulpitów prezentujących gromadzone w ramach usługi dane, z poziomu interfejsu graficznego. Oprogramowanie musi przynajmniej umożliwiać podział pulpitów na prywatne oraz współdzielone z innymi użytkownikami.
2.16. Oprogramowanie musi umożliwiać kontrolę dostępu do danych na podstawie systemu ról grup użytkowników (ang. Role-Based Access Control). Mechanizm konfiguracji uprawnień musi być dostępny z interfejsu graficznego jak i z poziomu interfejsu API dostarczonego oprogramowania..
2.17. Oprogramowanie musi umożliwiać porównywanie działania aplikacji w różnych przedziałach czasowych na poziomie czasów odpowiedzi, liczby błędów, poziomu ruchu i tym podobnych.
2.18. Oprogramowanie musi zbierać informacje o wszystkich błędach i wyjątkach. Musi istnieć możliwość zobaczenia szczegółowych informacji na temat transakcji, w których wystąpił błąd bądź został wygenerowany wyjątek.
2.19. Dostarczone oprogramowanie musi zapewniać wyszukiwanie w zgromadzonym przez niego zbiorze danych dotyczących transakcji na podstawie definiowanych filtrów lub zapytań. Tworzenie filtrów lub zapytań musi odbywać się przez intuicyjny wybór odpowiednich elementów z listy widocznej na ekranie dostępnych możliwości i łączenie ich przez wybrane z listy operatory. Oprogramowanie musi generować wywołania APIdla zdefiniowanego filtra, które można wykorzystać w innym narzędziu lub skrypcie.
2.20. Dostarczone oprogramowanie musi pozwalać na konstruowanie filtrów lub zapytań z użyciem operatorów logicznych, porównań ciągów znaków takich jak: „zawiera", „zaczyna się od" oraz innych celem przeszukiwania danych dotyczących transakcji. Konstruowanie filtrów lub zapytań musi się odbywać w sposób prosty bez konieczności ręcznego wpisywania wyrażeń z wymaganą składnią jak np. SQL.
2.21. Oprogramowanie musi dostarczać możliwość pobierania metryk z endpointów Prometheus.
2.22. Oprogramowanie musi posiadać funkcjonalność logowania wszystkich aktywności użytkowników związanych ze zmianami konfiguracji. Logowanie musi umożliwiać jednoznaczne wskazanie osoby, która wykonała zmianę.
2.23. Dostarczone oprogramowanie - musi oferować także udokumentowany interfejs programistyczny (API) służący do konfiguracji Oprogramowania, pobierania danych, a w tym metryk historycznych.
2.24. Pozwala analizować wpływ zmian wersji oprogramowania na wydajność procesów, transakcji oraz wartość metryk związanych z obsługą użytkowników aplikacji, celem wskazania czy wprowadzane zmiany prowadzą do pożądanego stanu funkcjonowania aplikacji. Oprogramowanie musi oferować możliwość rejestracji zdarzenia wgrania nowej wersji aplikacji.
2.25. Oferowanie oprogramowanie musi posiadać mechanizm pojedynczego agenta – gdzie instalujemy jeden komponent na monitorowanym środowisku a następnie komponent ten automatycznie wykrywa uruchomione aplikacje, podłączając się do nich w celu pobrania metryk. Nie jest dopuszczalne rozwiązanie wymagające oddzielnej instalacji agenta per monitorowana aplikacja/komponent.
2.26. Oprogramowanie musi udostępniać mechanizm automatycznej aktualizacji agenta monitorującego poprzez sieć lub z lokalnego mirrora repozytorium agentów.
2.27. Oprogramowanie musi umożliwiać instalację agenta w środowisku Kubernetes z wykorzystaniem miedzy innymi HELM
2.28. Oprogramowanie musi mieć możliwość zbierania przynajmniej części metryk z dokładnością do 1 sekundy.

2.29. Oprogramowanie musi dostarczać gotowe, predefiniowane pulpity operatorskie do monitorowania każdej ze wspieranych przez siebie technologii.
2.30. Oprogramowanie musi pozwalać na definiowanie okien serwisowych „wyciszających" zdarzenia
2.31. Oprogramowanie musi umożliwiać powiązanie ze zdarzeniami konkretnych akcji naprawczych.
2.32. Oprogramowanie musi wspierać importowanie transakcji z rozwiązań:
a) OpenTelemetry
b) OpenTracing
c) Zipkin
d) Jaeger

3. W zakresie monitorowania użytkownika końcowego, oprogramowanie musi spełniać poniższe wymagania techniczne i posiadać niżej wymienione funkcje:

3.1. Pozwala na monitorowanie sposobu działania aplikacji z perspektywy przeglądarek internetowych użytkowników końcowych w zakresie czasu odpowiedzi aplikacji i występujących błędów.
3.2. Przedstawia informacje w jaki sposób użytkownicy końcowi wchodzą w interakcję z aplikacją i w jaki sposób w niej nawigują.
3.3. Przedstawia wpływ sieci i czasu wczytywania aplikacji po stronie przeglądarki internetowej na doświadczenia użytkownika końcowego.
3.4. Pozwala na zbieranie danych dotyczących używanej przeglądarki, systemu operacyjnego, wykorzystywanego urządzenia i innych parametrów pozwalających na identyfikację jak aplikacja działa dla różnych grup użytkowników.
3.5. Musi umożliwiać powiązanie monitorowanej sesji użytkownika końcowego 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).
3.6. Musi umożliwiać automatyczne sprawdzanie dostępności i wydajności aplikacji poprzez cykliczne lub jednorazowe wykonywanie skryptu symulującego pracę użytkownika z możliwością monitorowania pracy użytkownika końcowego zarówno z wewnątrz, jak i z zewnątrz infrastruktury Zamawiającego.
3.7. Musi pozwalać na geolokalizację zdarzeń na bazie adresów IP oraz umożliwiać jej wizualizacje na mapie.
3.8. Musi pozwalać na mapowanie danych geolokalizacyjnych dla wewnętrznej adresacji IP poprzez wprowadzenie wymaganych danych przez interfejs web, plik z danymi geolokacyjnymi lub modyfikację agenta monitorującego użytkownika końcowego.
3.9. Pozwala na monitorowanie działanie aplikacji w wersji mobilnej dla systemów Android i iOS.

4. W zakresie gromadzenia, monitorowania i analizy logów, oprogramowanie musi spełniać poniższe wymagania i posiadać niżej wymienione funkcje:

4.1. Oprogramowanie musi automatycznie zbierać logi i błędy o poziomie minimum WARN przynajmniej dla aplikacji JAVA, Node.js, PHP oraz korelować je z transakcjami podczas których się pojawiły.
4.2. Oprogramowanie musi zbierać logi przynajmniej ze środowiska Kubernetes opartych o Docker lub containerd
4.3. Oprogramowanie musi dostarczać mechanizm filtrowania oraz grupowania zgromadzonych logów

Załącznik nr 1 do Specyfikacji Oprogramowania

Wykaz infrastruktury Zamawiającego, podlegającej monitorowaniu w zakresie dostarczonego Oprogramowania.

1. Na infrastrukturę obsługującą produkcyjny system składają się następujące serwery:

System operacyjny	Rodzaj serwera	vCPU	RAM (w GB)	Stosowane Technologie
ePUE				
Red Hat Core OS	Aplikacyjny	16	128	OpenShift, Java
Red Hat Core OS	Aplikacyjny	16	128	OpenShift, Java
Red Hat Core OS	Aplikacyjny	16	128	OpenShift, Java
Red Hat Core OS	Aplikacyjny	16	128	OpenShift, Java
Red Hat Core OS	Aplikacyjny	16	128	OpenShift, Java
Red Hat Core OS	Aplikacyjny	16	128	OpenShift, Java
RHEL 7.9	Bazodanowy	24	64	PostgreSQL
RHEL 7.9	Bazodanowy	24	64	PostgreSQL
RedHat SSO				
RHEL 7.4	Aplikacyjny	16	24	Jboss, Java
RHEL 7.4	Aplikacyjny	16	24	Jboss, Java
RHEL 7.4	Aplikacyjny	16	24	Jboss, Java
RHEL 7.4	Aplikacyjny	16	24	Jboss, Java
RHEL 7.4	Bazodanowy	24	48	PostgreSQL
LIDER				
RHEL 6.7	Aplikacyjny	24	64	Jboss, Java
RHEL 8.9	Bazodanowy	12	64	Oracle DB
RED				
RHEL 7.6	Aplikacyjny	2	8	Java, NodeJS
RHEL 7.4	Bazodanowy	24	64	PostgreSQL
KeyCloak SSO*				
CentOS 7.8	Aplikacyjny	88*	756*	OKD,Jboss/Wildfly, Java,
CentOS 7.8	Aplikacyjny	88*	756*	OKD,Jboss/Wildfly, Java,
CentOS 7.8	Aplikacyjny	88*	756*	OKD,Jboss/Wildfly, Java
RHEL 7.7	Bazodanowy	4	32	PostgreSQL
Web server				
RHEL 7.4	Infrastrukturalny	88	756	Nginx
RHEL 7.4	Infrastrukturalny	88	756	Nginx
Serwer Kolejek				
RHEL 8.6	Kolejkowy	4	16	Jboss AMQ
RHEL 7.9	Kolejkowy	4	24	Jboss AMQ

*) KeyCloak SSO zarówno na środowisku produkcyjnym, jak i UAT wykorzystuje wspólne serwery (wspólne środowisko OKD) w warstwie aplikacyjnej.

2. Na infrastrukturę obsługującą system UAT składają się następujące serwery:

System operacyjny	Rodzaj serwera	vCPU	RAM (W gb)	Stosowane Technologie
ePUE				
Red Hat Core OS	Aplikacyjny	25	192	OpenShift, Java
Red Hat Core OS	Aplikacyjny	25	192	OpenShift, Java
Red Hat Core OS	Aplikacyjny	25	192	OpenShift, Java
RHEL 7.9	Bazodanowy	12	40 GB	PostgreSQL
RedHat SSO				
RHEL 8.5	Aplikacyjny	16	32	Jboss, Java
RHEL 7.7	Bazodanowy	48	512	PostgreSQL
LIDER				
RHEL7.3	Aplikacyjny	8	16	Jboss, Java
RHEL 6.6	Bazodanowy	16	96	Oracle DB
RED				
RHEL 7.7	Aplikacyjny	2	8	Java, NodeJS
RHEL 6.5	Bazodanowy	10	64	PostgreSQL
KeyCloak SSO*				
CentOS 7.8	Aplikacyjny	88*	756*	OKD,Jboss/Wildfly, Java
CentOS 7.8	Aplikacyjny	88*	756*	OKD,Jboss/Wildfly, Java
CentOS 7.8	Aplikacyjny	88*	756*	OKD,Jboss/Wildfly, Java
RHEL 7.3	Bazodanowy	4	8	PostgreSQL
Web server				
RHEL 7.4	Infrastrukturalny	4	32	Nginx
Serwer Kolejek				
RHEL 7.9	Kolejkowy	1	12	Jboss AMQ
RHEL 8.6	Kolejkowy	4	16	Jboss AMQ

*) KeyCloak SSO zarówno na środowisku produkcyjnym, jak i UAT wykorzystuje wspólne serwery (wspólne środowisko OKD) w warstwie aplikacyjnej.