Część III SIWZ

OPIS PRZEDMIOTU ZAMÓWIENIA

Zadania pn. „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego"

Zamawiający:
Powiat Pułtuski ul. Białowiejska 5 06-100 Pułtusk

Użytkownicy:
Powiatowe Centrum Zarządzania Kryzysowego WZK SP w Pułtusku, Wojewódzkie Centrum Zarządzania Kryzysowego Wojewody Mazowieckiego, Urząd Gminy Pułtusk, Urząd Gminy Pokrzywnica, Urząd Gminy Obryte.

I. Przedmiot zamówienia

Przedmiotem zamówienia jest:
1. Budowa (dostawa, montaż, konfiguracja) powiatowego elementu sterowania systemem w Powiatowym Centrum Zarządzania Kryzysowego w Pułtusku. Sporządzenie dokumentacji powykonawczej.
2. Budowa (dostawa, montaż, konfiguracja) elementów sterowania systemu oraz punktów alarmowych w gminach Obryte i Pokrzywnica. Sporządzenie dokumentacji powykonawczej.
3. Budowa (dostawa, montaż, konfiguracja) elementu sterowania systemu oraz punktów alarmowych w gminie Pułtusk. Sporządzenie dokumentacji powykonawczej.

System ma być zintegrowany z systemem ostrzegania i alarmowania województwa mazowieckiego.

II. Zakres przedmiotu zamówienia

1. Zamówienie obejmuje:

a) dostawę dedykowanego oprogramowania dla systemu integrującego zdefiniowane funkcjonalności;
c) dostawę, instalację i konfigurację oprogramowania, niezbędnych podzespołów punktów alarmowych (syren elektronicznych, anten, urządzeń sterujących, radiotelefonów, urządzeń zasilających, masztów itp.);
d) dokonanie niezbędnych uzgodnień z osobami zarządzającymi obiektami, na których będą instalowane punkty alarmowe, elementy sterowania systemem, dokonanie uzgodnień z administratorami sieci teleinformatycznych;
e) instalację dostarczonych punktów alarmowych we wskazanych przez Zamawiającego lokalizacjach, zgodnie z warunkami wydanymi przez zarządzających obiektami, na których będą instalowane punkty alarmowe;
f) konfigurację urządzeń sterujących syreną;

Dofinansowanie Projektu pn.: „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego" jest realizowane przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Regionalnego Programu Operacyjnego Województwa Mazowieckiego 2014-2020 (Oś. Priorytetowa V Gospodarka przyjazna środowisku, działania 5.1. Dostosowanie do zmian klimatu)

g) konfigurację systemu na szczeblu powiatowym i gminnym, integrację wszystkich dostarczonych punktów z systemem wojewódzkim za pośrednictwem interfejsu API (Application Programming Interface), będącym narzędziem pozwalającym na komunikację centrali wojewódzkiej z systemami obcymi. Opis funkcjonalny aplikacji stanowi załącznik nr 1 do opisu przedmiotu zamówienia , instrukcja API stanowi załącznik nr 2 do opisu przedmiotu zamówienia;

h) wszystkie koszty zapewnienia kompatybilności systemu powiatowego z Systemem Wczesnego Ostrzegania i Systemem Wykrywania i Alarmowania Wojewody Mazowieckiego, w tym zakup kluczy licencyjnych, licencji oraz dostępu do środowiska testowego ponosi Wykonawca;

i) sporządzenie dokumentacji powykonawczej zawierającej: wszystkie dokonane uzgodnienia, protokoły odbiorów lub potwierdzenia należytego wykonania prac wydane przez właścicieli (zarządców) obiektów wykorzystanych do instalacji punktów alarmowych, urządzeń sterowania, wyniki wykonanych pomiarów elektrycznych oraz toru antenowego (WFS), plany, schematy, rysunki, opisy, zdjęcia, instrukcje obsługi systemu, specyfikację techniczną użytych materiałów oraz oświadczenie Wykonawcy o przeprowadzeniu prac zgodnie z obowiązującymi normami i przepisami.

Wszystkie urządzenia i środki materiałowe dostarczone przez Wykonawcę muszą być fabrycznie nowe, nieuszkodzone, sprawne technicznie i pozbawione wad prawnych.

2. Zasady instalacji punktów alarmowych:

a) W związku z umiejscowieniem elementów systemu (1b; 1c; 1d; 2a; 3a podanych w rozdziale V opisu przedmiotu zamówienia) na terenie objętym opieką konserwatora zabytków Wykonawca – przed przystąpieniem do realizacji zadania - przedstawi Zamawiającemu szkice lub rysunki wyjaśniające jednoznacznie sposób montażu elementów systemu w ww. lokalizacjach;

b) wykonawca uzyska własnym staraniem i na własny koszt wszystkie niezbędne uzgodnienia, potrzebne do realizacji zamówienia od właścicieli obiektów (osób aktualnie zarządzających obiektami), na których dokona instalacji elementów systemu;

c) wykonawca uzgodni sposób instalacji elementów systemu oraz aranżację jego okablowania z właścicielem (osobą zarządzającą obiektem);

d) urządzenia teletransmisyjne, sterujące oraz inne zapewniające poprawną pracę punktu alarmowego muszą zostać umieszczone w zamykanych na bezpieczny zamek szafach teletechnicznych;

f) wykonawca zabezpieczy elementy systemu pod względem ochrony przepięciowej i przeciwporażeniowej;

g) wykonawca dokona niezbędnych pomiarów elektrycznych oraz toru antenowego zainstalowanych elementów systemu:

- w zakresie pomiarów elektrycznych badania rezystancji izolacji, badania skuteczności ochrony przeciwporażeniowej,

• w zakresie toru antenowego badania obejmujące współczynnik SWR oraz tłumienie toru antenowego;

h) wykonawca wykona instalację odgromową, dopuszczalne jest wykorzystanie istniejącej instalacji odgromowej danego obiektu;

i) oprogramowanie, urządzenia i materiały podczas realizacji przedmiotu zamówienia muszą być dostarczone bezpośrednio do miejsc instalacji wskazanych w opisie przedmiotu zamówienia przez Zamawiającego (rozdział V) Zamawiający nie zapewnia pomieszczeń do magazynowania elementów systemu przed ich instalacją.

3. Minimalne wymagania techniczne i funkcjonalne punktów alarmowych:

Dostarczone w ramach zamówienia punkty alarmowe muszą zostać zintegrowane z wojewódzkim systemem alarmowania ludności, a także spełniać następujące wymagania techniczne i funkcjonalne:

1) syreny alarmowe

– blok syreny:

a) modułowa konstrukcja syreny pozwalająca na łatwą rozbudowę urządzenia poprzez dołączenie standardowego modułu;

b) montaż głowicy syreny na dedykowanym maszcie;

c) maszt dostosowany do konstrukcji dopasowanej do planowanego obciążenia;

d) konstrukcja głowicy i masztu odporna na uszkodzenia mechaniczne i korozję (zabezpieczenie poprzez ocynkowanie), odporna na działanie wiatru – wykonana zgodnie ze sztuką budowlaną;

f) głośniki 6 tubowe – min. 112 dB (A) z 30 m, moc 900W;

g) temperatura pracy bloku syreny: od -30°C do +60°C przy wilgotności do 90%.

– blok sterujący:

a) zasilanie sieciowe 230 V AC ± 10%;

b) zastosowanie w instalacji antenowej zabezpieczenia zapewniającego ochronę przeciwpiorunową;

c) zasilanie buforowe na bazie akumulatorów bezobsługowych;

d) liczba alarmów na zasilaniu buforowym – min. 6 trzyminutowych w ciągu 24 godzin;

e) przekazywanie alarmów zgodnych z Rozporządzeniem Rady Ministrów z dnia 7 stycznia 2013 r. (Dz. U. z 2012 r. poz. 96) oraz dowolnych zdefiniowanych i zapisanych w pamięci punktu alarmowego sygnałów lub komunikatów głosowych;

f) przekazywanie komunikatów głosowych w czasie rzeczywistym ze stanowiska dyspozytorskiego lub lokalnie przez mikrofon;

g) temperatura pracy bloku sterującego (instalacja wewnątrz budynków) od 0°C do + 50°C;

h) blok sterujący punktu alarmowego musi być wyposażony w min. jeden zamek patentowy, spełniać funkcję centralki antywłamaniowej (klasa IP 65 dla bloku instalowanego wewnątrz

budynku, klasa IP 66 dla bloku instalowanego na zewnątrz budynku), sygnalizującej otwarcie bloku sterującego i przekazującej informacje o nieautoryzowanym dostępie do bloku sterującego do urządzenia sterowania i kontroli zlokalizowanego w Powiatowym Centrum Zarządzania Kryzysowego;

g) blok sterujący musi monitorować na bieżąco stan napięć zasilających syrenę i umożliwiać przekazywanie informacji o stanie jej zasilania do systemu (pomiar napięcia akumulatorów pod obciążeniem i bez obciążenia, badanie symetrii napięć akumulatorów, kontrolę obecności napięcia 230 V AC) oraz temperatury w bloku sterującym.

W skład kompletu musi wchodzić głowica głośników szczelinowych lub tubowych, blok (szafa) kontrolno - sterujący: sterownik z urządzeniem włączającym, generator sygnałów, wzmacniacze mocy, zasilacz sieciowy, akumulatory, radiotelefon cyfrowy o mocy min. 5W, max. 25W pracujący w paśmie VHF, antena.

4. Minimalne wymagania techniczne i funkcjonalne powiatowego elementu sterowania systemem:

a) zasilanie sieciowe 230 V, zasilanie rezerwowe – akumulator, możliwość podłączenia pod gniazdo zapalniczki samochodowej;
b) obsługa urządzeń sterujących i syren w systemie cyfrowej łączności radiowej;
c) obsługa urządzeń sterujących i syren za pomocą Ethernet – LAN/INTERNET;
d) wbudowane radio analogowo-cyfrowe 25W z mikrofonem do nadawania komunikatów głosowych;
e) mobilność – praca w operacyjnych samochodach ;
f) elastyczność systemu – niezależna obsługa wszystkich elementów z poziomu urządzenia sterowania i kontroli;
g) gromadzenie raportów z testów wszystkich urządzeń;
h) kontrola systemu w sposób ciągły – monitorowanie zdarzeń w systemie;
i) sterowanie alarmami i komunikatami głosowymi we wszystkich lub pojedynczych syrenach, w syrenach przypisanych do sektorów, w wyznaczonych grupach;
j) włączanie kanału audio do ogłaszania komunikatu "na żywo" przez mikrofon;
k) rejestracja załączeń alarmów i komunikatów oraz wyników testów w pamięci urządzenia sterowania i kontroli (gromadzenie raportów z zarejestrowanych zdarzeń w systemie);
l) możliwość obsługi zdalnych stacji meteorologicznych i detektorów skażeń oraz innych urządzeń pomiarowych (przez dwa media sterujące: RADIO i INTERNET);
m) kontrola zasilania sieciowego i otwarcia drzwi w punktach alarmowania;
n) wywoływanie i zapis testów sprawności głośników i wzmacniaczy syren bez włączania głośnych sygnałów;
o) testy akumulatorów syren pod obciążeniem – raportowanie kondycji akumulatorów;

p) logowanie do urządzenia sterowania i kontroli z dowolnego komputera z przeglądarką;
q) stacyjka na klucz patentowy (ON/OFF) lub inne zabezpieczenie przed nieautoryzowanym uruchomieniem.

5.Minimalne wymagania techniczne i funkcjonalne urządzenia sterowania przeznaczonego dla gmin

Urządzenia sterowania przeznaczone dla gmin powinny spełniać następujące założenia i wymagania:

a) obsługa urządzeń sterowania i syren w systemie bezpiecznej łączności radiotelefonicznej;

b) obsługa urządzeń sterowania i syren za pomocą bezpiecznych połączeń internetowych;

c) obsługa urządzeń peryferyjnych takich jak m.in. czujnik skażeń, stacja pogodowa (przez dwa media sterujące: RADIO i INTERNET);

d) wbudowane radio analogowo-cyfrowe z mikrofonem do nadawania komunikatów głosowych;

e) praca urządzeń sterowania z dowolnego akumulatora zewnętrznego 12V;

f) mobilność urządzeń sterowania – praca urządzeń sterowania w operacyjnych samochodach (zasilanie 12V);

g) awaryjne podtrzymanie zasilania urządzeń sterowania w czuwaniu bez zasilania sieciowego minimum 48 godzin;

h) stacyjka na klucz patentowy (ON/OFF) lub inne zabezpieczenie przed nieautoryzowanym uruchomieniem.

h) minimalne wyposażenie peryferyjne urządzeń sterowania:
- akumulator minimum 12V 15Ah
- mikrofon do nadawania komunikatów
- kabel zasilania sieciowego 230V
- przewód zasilania do gniazda zapalniczki samochodowej.

6. Minimalne wymagania techniczne i funkcjonalne oprogramowania sterującego:

a) możliwość uruchomienia za pomocą łączności radiowej, połączeń internetowych wszystkich punktów alarmowych na terenie powiatu pułtuskiego lub wybranych przez użytkownika z poziomu gminnego, powiatowego i wojewódzkiego;
b) podawanie przez syreny komunikatów dźwiękowych i głosowych zgodnie z wymaganiami prawa UE i RP, uruchamianie komunikatów musi odbywać się przy dostępie sieci radiowej, połączeń internetowych;
c) oprogramowanie pomiędzy centralą wojewódzką a powiatowym elementem sterowania, urządzeniami sterowania gmin a powiatowym elementem sterowania musi wykorzystywać transmisję danych zapewniającą bezpieczne połączenia internetowe (np. VPN) oraz

kodowaną transmisję radiotelefoniczną;

d) rejestracja wszystkich operacji dokonywanych zarówno drogą internetową, jak i radiową;

e) oprogramowanie powinno zawierać cyfrową mapę określonej lokalizacji w pełni obrazującą położenie i stan wszystkich obiektów systemu;

f) podkład mapowy do wizualizacji lokalizacji elementów systemu – dane z projektu "OpenStreetMap" lub z innego źródła z otwartą licencją dla zawartości baz danych typu OpenDataCommons Database Contents License (ODbCL) z obszarem Powiatu Pułtuskiego z wyraźnie zaznaczonymi granicami obszaru powiatu oraz gmin wraz z buforem poza jego granicami;

g) mapa ma być domyślnie w trybie off-line, przy czym Wykonawca zapewni możliwość jej aktualizacji w dowolnym momencie, samodzielnie przez Zamawiającego poprzez kliknięcie ikony lub innego zaznaczenia na mapie o nazwie „aktualizuj mapę";

h) podkład mapowy ma umożliwiać funkcję płynnego skalowania przy użyciu linijki skali i kółka myszy w sposób równoległy od skali 1:10 000 do skali 1: 200 000 z funkcjonalnością jaka znajduje się w serwisie http://www.openstreetmap.org.pl/;

i) generowanie pełnego wykazu oraz zobrazowanie na mapie wszystkich elementów systemu wraz z ich lokalizacją i informacją o ich statusie;

j) sterowanie elementami systemu z poziomu mapy i rozwijanej listy wyboru;

k) wybór punktu alarmowego na mapie poprzez zakreślenie obszaru wskaźnikiem „myszy" lub z listy wyboru w celu tworzenia nowych stref powiadamiania lub do celów testowych;

l) symbole elementów systemu umieszczone na mapie mają zapewniać maksymalną czytelność a ich kolorystyka być jednoznacznie związana ze statusem - gotowość/ awaria/ serwis/ test/ alarmowanie;

ł) symbole ze statusami innymi niż gotowość muszą być wyświetlane ponad pozostałymi symbolami(„zawsze na wierzchu");

m) oprogramowanie musi mieć możliwość korzystania z map zewnętrznych;

n) zabezpieczenie sygnałów transmisyjnych przed nieautoryzowanym dostępem, uniemożliwiające nieuprawnione sterowanie punktami alarmowymi;

o) włączanie zaprogramowanych sygnałów alarmowych, emitowanie komunikatów audio w czasie rzeczywistym przez pojedyncze punkty alarmowania lub ich grupy;

p) możliwość włączania alarmów i testowania syren. Wbudowana baza danych powinna umożliwiać pełną edycję przez użytkownika systemu, zawierająca szczegółowe informacje o wszystkich punktach alarmowych, w tym wprowadzanie do systemu nowych punktów alarmowych, oraz urządzeń peryferyjnych za pomocą współrzędnych geograficznych GPS;

q) wizualizacja całego systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego;

r) możliwość odczytu, rejestracji i wizualizacji danych z zainstalowanych urządzeń;

s) przydzielanie nazw i adresów obiektów przez użytkownika;

t) zróżnicowanie uprawnień dla administratora i użytkownika systemu (dostęp dla administratora z pełnymi uprawnieniami bez ograniczeń);

u) pełna edycja kont użytkowników łącznie z uprawnieniami, loginami, hasłami, możliwością

zablokowania/odblokowania konta użytkownika przez administratora;

v) dostęp dla wielu operatorów (z różnych komputerów) w tym samym czasie;

w) dostęp do oprogramowania sterującego ma być realizowany dwuetapowo poprzez podanie loginu i hasła oraz pinu lub innego rozwiązania;

x) oprogramowanie sterujące ma bazować na najnowszych technikach informatycznych powszechnie dostępnych (menu podręczne do wszystkich funkcji systemu ma wykorzystywać „dymki opisowe" ujawniające się po najechaniu kursorem na ikonę lub przycisk funkcji);

y) możliwość wydruku bezpośrednio z oprogramowania do drukarki sieciowej oraz import do pliku pdf raportów zdarzeń, pracy i sprawności systemu itp.

III. Wymagane funkcjonalności systemu

Podstawowe cechy, które musi spełniać system wczesnego ostrzegania i alarmowania powiatu pułtuskiego po wykonaniu niniejszego zadania:

1. Uruchamianie syren alarmowych odbywać się musi poprzez załączenie alarmu za pomocą klawiatury powiatowego elementu sterowania w (PCZK).
   Możliwe musi być również uruchomienie alarmu z wojewódzkiej centrali alarmowej (WCZK) w Warszawie. Proces przebiegać ma poprzez komunikację między centralami za pomocą łącza IP, a następnie poprzez łącza radiowe na dedykowanej częstotliwości.
2. Urzędy Gmin muszą mieć możliwość uruchamiania punktów alarmowych na administrowanym terenie za pomocą zamontowanych urządzeń sterowania oraz oprogramowania.
3. Wszystkie zainstalowane punkty alarmowe muszą być widoczne i zarządzane z poziomu powiatowego i wojewódzkiego. Gminy muszą mieć możliwość zarzadzania punktami alarmowymi na administrowanym terenie.
4. Zarządzanie wszystkimi punktami alarmowymi , w tym uruchamianie, zatrzymywanie, pobieranie danych o stanie pracy, zasilaniu musi odbywać się za pomocą aplikacji dostępowej.
5. System musi umożliwiać komunikację z punktami alarmowymi za pomocą dwóch niezależnych mediów transmisyjnych (łącze IP oraz łącze radiowe).
6. System musi umożliwiać dodawanie kolejnych dostępów "klienckich" jak i dostępów do elementów sterowania (powiatowego, gminnych) przez administratora powiatowego/gminnego.
7. System musi zapewnić możliwość późniejszej integracji z pozostałymi systemami *funkcjonującymi w powiecie, gminie (udostępnienie API).*
8. System wczesnego ostrzegania i alarmowania powiatu pułtuskiego musi posiadać następujące cechy:
   a) możliwość natychmiastowego uruchomienia dowolnej ilości syren w dowolnej konfiguracji;
   b) praca w paśmie VHF z odstępem międzykanałowym 12,5 kHz w trybie simpleksowym;
   c) szyfrowana transmisja radiowa;
   d) automatyczna diagnostyka systemu w tle (24 h / 7 dni) bez wpływu na działanie

i gotowość systemu.

9. Oferowane urządzenia muszą spełniać wszystkie wymagania i warunki określone przez Zamawiającego.
10. Wykonawca zobowiązany jest dostarczyć wszystkie dokumenty niezbędne do odbioru zainstalowanych urządzeń:
    a) karty gwarancyjne producenta;
    b) dokumentacje i instrukcje obsługi, w tym deklaracje zgodności z certyfikatem bezpieczeństwa CE;
    c) dokumenty potwierdzające, że dostarczone urządzenia posiadają homologację.
11. Zamawiający dopuszcza zastosowanie produktów, materiałów lub urządzeń równoważnych. Jeśli w opisie przedmiotu zamówienia zamieszczone są jakiekolwiek nazwy własne producentów – nie są one wiążące dla wykonawcy, należy je traktować wyłącznie jako przykładowe dla zobrazowania opisywanych parametrów i wymogów technicznych.
    Jako równoważne dopuszcza się inne rozwiązania niż podane w opisie przedmiotu zamówienia, pod warunkiem zagwarantowania równorzędnych parametrów technicznych i technologicznych nie gorszych niż określone w opisie oraz zgodności z obowiązującymi wymaganiami prawnymi.
    W takim przypadku Wykonawca musi złożyć dokumenty (wszelkie dowody) jednoznacznie potwierdzające spełnianie przez rozwiązania równoważne wymagań Zamawiającego. Wykonawca odpowiedzialny będzie względem Zamawiającego za wszelkie szkody związane z wprowadzeniem takich rozwiązań.
    Wszelkie urządzenia, materiały i rozwiązania równoważne muszą spełniać minimalne wymagania przedstawione w niniejszym opisie przedmiotu zamówienia, tj. muszą być co najmniej:
    a) tej samej mocy;
    b) tej samej trwałości;
    c) spełniać te same funkcje.

IV. Warunki gwarancji

1. Wykonawca udzieli Zamawiającemu minimum **60 m-cy** gwarancji jakości, w rozumieniu art. 577 k.c. chyba, że gwarancja producenta jest dłuższa, to obowiązuje gwarancja udzielana przez producenta.
2. Wykonawca zobowiązuje się przystąpić do naprawy niezwłocznie, nie później niż w ciągu 24 godzin od daty i godziny powiadomienia (telefonicznie, faxem, e-mailem) i dokona bezpłatnej naprawy wszelkich usterek dostarczonego sprzętu w terminie do 7 dni od dnia zgłoszenia. W sytuacji braku możliwości naprawy urządzenia w miejscu zainstalowania, Wykonawca w tym czasie zobowiązany jest zdemontować i odebrać sprzęt w celu naprawy oraz dostarczyć i zamontować sprzęt po naprawie na własny koszt i ryzyko w ramach przedmiotu zamówienia.

3. W sytuacji, w której Wykonawca stwierdzi niemożność naprawy sprzętu, zobowiązany jest on do bezpłatnej wymiany uszkodzonego sprzętu na sprzęt fabrycznie nowy, wolny od wad, o nie gorszych parametrach technicznych od wymienionego – na własny koszt i ryzyko.
4. Wykonawca musi zagwarantować istnienie wyznaczonego punktu przyjęć zgłoszeń gwarancyjnych do kontaktu telefonicznego, faksowego, e-mailowego, w celu zgłaszania wniosków o wykonanie usługi gwarancyjnej. Przyjmowanie zgłoszeń o wszelkich nieprawidłowościach w działaniu dostarczonego sprzętu będzie dokonywane telefonicznie przez cały okres gwarancji jakości. Każdorazowe zgłoszenie telefoniczne będzie niezwłocznie potwierdzane faksem lub e-mailem na numer lub adres uzgodniony z Wykonawcą.

V. Lokalizacja elementów systemu

1. Punkty alarmowe:

a) Zespół Szkół Nr 2 w Pułtusku, ul. Polna 7, 06-100 Pułtusk;
b) Miejski Ośrodek Sportu i Rekreacji w Pułtusku, ul. Daszyńskiego 17 A, 06-100 Pułtusk (obszar objęty ochroną konserwatora zabytków);
c) Miejski Centrum Kultury i Sztuki w Pułtusku ul. Plac Teatralny 4, 06-100 Pułtusk (obszar objęty ochroną konserwatora zabytków);
d) Urząd Miejski w Pułtusku, ul. Rynek 41, 06-100 Pułtusk (obszar objęty ochroną konserwatora zabytków);
e) Zespół Szkół Nr 4 w Pułtusku, ul. New Britain 1, 06-100 Pułtusk;
f) Przedsiębiorstwo Energetyki Cieplnej Sp. z o. o. w Pułtusku Ciepłownia Centralna, ul. Kolejowa 8, 06-100 Pułtusk;
g) Publiczna Szkoła Podstawowa Nr 3 w Pułtusku, Al. Tysiąclecia 14, 06-100 Pułtusk;
h) Przedszkole Miejskie Nr 4 w Pułtusku, ul. Krajewskiego 3, 06-100 Pułtusk;
i) *Towarzystwo Budownictwa Społecznego* Sp. z o. o. w Pułtusku, ul. Białowiejska 19, 06-100 Pułtusk;
j) Zespół Szkolno – Przedszkolny w Przemiarowie, Przemiarowo 33, 06-100 Pułtusk;
k) Świetlica wiejska w Trzcińcu, Trzciniec 17, 06-100 Pułtusk;
l) Świetlica wiejska w Białowieży, Białowieża 15, 06-100 Pułtusk;
m) Publiczna Szkoła Podstawowa w Płocochowie, Płocochowo 92, 06-100 Pułtusk;
n) Świetlica wiejska w Kacicach, Kacice 24B, 06-100 Pułtusk;
o) Świetlica wiejska w Grabówcu, Grabówiec 16, 06-100 Pułtusk;
p) Urząd Gminy Obryte, Obryte 185, 07-215 Obryte;
q) Zespół Placówek Oświatowych w Obrytem - Szkoła filialna w Zambskach Kościelnych, Zambski Kościelne 33, 07-215 Obryte;
r) Zespół Placówek Oświatowych w Obrytem - Szkoła filialna w Sokołowie Włościańskim, Sokołowo Włościańskie 63, 07-215 Obryte;
s) Urząd Gminy Pokrzywnica, Aleja Jana Pawła II 1, 06-121 Pokrzywnica;

Dofinansowanie Projektu pn.: „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego" jest realizowane przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Regionalnego Programu Operacyjnego Województwa Mazowieckiego 2014-2020 (Oś. Priorytetowa V Gospodarka przyjazna środowisku, działania 5.1. Dostosowanie do zmian klimatu)

t) Remiza OSP w Dzierżeninie, Dzierżenin 59, 06-114 Dzierżenin;
u) Remiza OSP w Gzowie, Gzowo 26, 06-114 Dzierżenin.

2. Powiatowy element sterowania, oprogramowanie sterujące:

a) Powiatowe Centrum Zarządzania Kryzysowego w Pułtusku, ul. Daszyńskiego 19, 06-100 Pułtusk (obszar objęty ochroną konserwatora zabytków).

3. Urządzenia sterowania przeznaczone dla gmin, oprogramowanie sterujące:

a) Wydział Zarządzania Kryzysowego, Spraw Wojskowych i Straży Miejskiej Urzędu Miejskiego w Pułtusku, ul. Staszica 35, 06-100 Pułtusk (obszar objęty ochroną konserwatora zabytków);
b) Urząd Gminy Obryte, Obryte 185, 07-215 Obryte;
c) Urząd Gminy Pokrzywnica, Aleja Jana Pawła II 1, 06-121 Pokrzywnica.

VI. Dokumentacja powykonawcza przedmiotu zamówienia

1. Dokumentacja powykonawcza, musi być zgodna z wymaganiami funkcjonalnymi i technicznym określonymi w opisie przedmiotu zamówienia, obowiązującymi przepisami prawa, normami technicznymi branżowymi oraz ma być kompletna z punktu widzenia celu, któremu ma służyć.
2. Wykonawca zobowiązany jest do umieszczenia na wykonywanej dokumentacji logotypów wg wzorów umieszczonych na stronie internetowej www.funduszedlamazowsza.eu/realizuje-projekt/poznaj-zasady-promowania-projektu/.
3. Dokumentacja powykonawcza powinna zawierać:
   a) pisemne uzgodnienia z zarządzającymi obiektami, na których zinstalowano punkty alarmowe, urządzenie sterowania i kontroli, urządzenia sterowania przeznaczone dla gmin dotyczące sposobów i miejsc instalacji;
   b) pisemne uzgodnienia z Zamawiającym, jeżeli będą wymagane;
   c) specyfikacje techniczne wykonania, w tym wymagania BHP i ppoż.;
   d) plany, rysunki, inne dokumenty pozwalające jednoznacznie określić rodzaj i zakres wykonanych prac oraz uwarunkowania i dokładną lokalizację ich wykonania;
   e) zestawienie ilościowe i rodzaj materiałów, urządzeń i oprogramowania, użytych do wykonania elementów systemu;
   f) niezbędne obliczenia konstrukcyjne i elektryczne podpisane przez uprawnionego projektanta, wyniki pomiarów i testów;
   g) opis części telekomunikacyjnej, ochrony odgromowej, części elektrycznej, okablowania punktów alarmowych;
   h) protokół interfejsu wymiany danych (API) systemu, punktów alarmowych, systemu integrujacego system powiatowy z systemem wojewódzkim;
   i) aprobaty techniczne i certyfikaty użytych w projekcie materiałów i urządzeń;
   j) zdjęcia z poszczególnych lokalizacji punktów alarmowych.

VII. Odbiór i testy systemu

Zamawiający dokona odbioru końcowego przedmiotu zamówienia po spełnieniu następujących warunków:

1. Dostarczony przez Wykonawcę sprzęt i oprogramowanie podlega odbiorowi ilościowemu oraz jakościowemu przez Zamawiającego w oparciu o dostarczoną dokumentację powykonawczą.
2. Wykonawca dokona instalacji, konfiguracji i przeprowadzenia testów zainstalowanego oprogramowania w środowisku testowym systemu.
3. Wykonawca opracuje i przedstawi Zamawiającemu scenariusze akceptacyjnych testów systemu.
4. W każdym momencie, w czasie przeprowadzania testów Zamawiający ma prawo do żądania wykonania dotatkowych testów sporządzonych w oparciu o własne scenariusze i dane testowe.
5. Zakres przeprowadzonych testów musi obejmować całą funkcjonalność systemu, czyli wszystkie wymagania określone w opisie przedmiotu zamówienia.
6. Testy kończą się podpisaniem protokołu zawierającego zakres przeprowadzonych testów oraz informacje o błędach.
7. Warunkami do odbioru są:
   a) zakończenie prac związanych z dostawą, konfiguracją i uruchomieniem całości systemu;
   b) zakończenie z pozytywnym rezultatem procedury testowej;
   c) przekazanie Zamawiającemu kompletnej dokumentacji powykonawczej opisanej w opisie przedmiotu zamówienia;
   d) przekazanie Zamawiającemu wykazu składników majątkowych, które powstały w wyniku realizacji zamówienia.
8. Odbioru dokonuje powołana przez Zamawiającego komisja.
9. W czasie czynności odbiorowych Wykonawca zapewni wsparcie techniczne w zakresie niezbędnym do ich przeprowadzenia.
10. W przypadku stwierdzenia w trakcie odbioru niezgodności z opisem przedmiotu zamówienia, Wykonawca jest zobowiązany do ich usunięcia w terminie 7 dni od ich zgłoszenia przez Zamawiającego a po usunięciu niezgodności nastąpi odbiór przeprowadzonych prac.
11. Odbiór zostanie potwierdzony podpisaniem protokołu końcowego zdawczo-odbiorczego.

Sporządził : Marcin Siatkowski

Akceptuję: Dyrektor Wydziału Zarządzania Kryzysowego dr Robert Dynak

Dofinansowanie Projektu pn.: „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego" jest realizowane przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Regionalnego Programu Operacyjnego Województwa Mazowieckiego 2014-2020 (Oś. Priorytetowa V Gospodarka przyjazna środowisku, działania 5.1. Dostosowanie do zmian klimatu)

Załącznik nr 1 do opisu przedmiotu zamówienia
dla zadania pn. „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego"

Opis funkcjonalny interfejsu pomiędzy Centralą Mazowieckiego Urzędu Wojewódzkiego (MUW) a Powiatowym Elementem Sterowania Systemem (SP)

1. Parametry podstawowe interfejsu wymiany danych.

a) Protokół transmisji danych – TCP/IP;
b) Technologia usługi wymiany danych - Web Services;
c) Autorski algorytm kryptograficzny kodujący transmisję danych, uniemożliwiający nagranie i odtworzenie transmisji ponownie (dostępny w dokumentacji WebAPI SAOL MUW);
d) Protokół strumieniowania danych audio – TCP/IP;
e) Transmisja radiowa lub poprzez sieć WAN/LAN;

2. Tryb pracy interfejsu wymiany danych między MUW a SP:

Zakłada się pracę interfejsu w trybie „auto" (bez obsługowo) i „operator" (obsługa manualna przez upoważnionego użytkownika SP lub MUW).

a) W trybie „operator" interfejs pracuje w sesjach nawiązanych i kończonych na żądanie operatora MUW w dowolnym czasie (tryb priorytetowy);
b) W przypadku braku połączenia systemów SP i MUW ze sobą (interfejs w trybie off-line, brak połączenia miedzy systemami) oba systemy powinny być gotowe do działania autonomicznego. Oznacza to, że operator swojego systemu SP lub MUW posiada pełną kontrolę nad punktami alarmowania oraz czujnikami tylko swojego systemu;
c) W trybie „auto" interfejs pracuje w sesjach nawiązanych i kończonych automatycznie. Centrala MUW działa w trybie klient-serwer z użyciem gniazd i protokołów sieciowych (ang. SOCKET-SERVER). Architektura klient-serwer jest technologią budowy systemów informatycznych, polegającą na podziale systemu na współdziałające ze sobą dwie kategorie programów lub procesów: klientów i serwerów. (funkcjonalność KLIENT punkt 3, funkcjonalność SERVER punkt 4.

3. Funkcjonalność interfejsu wymiany danych w relacji MUW do SP (MUW -> SP) (wytyczne w WebApi SAOL MUW):

a) Automatyczne uwierzytelnienie centrali MUW w systemie SP;
b) Autoryzacja operatora MUW w systemie SP (przeslanie kodów autoryzacyjnych operatora MUW do SP);
c) Synchronizacja danych z systemu SP do MUW na żądanie MUW;
d) Pobieranie statusów o wszystkich PA (punktach alarmowania) na żądanie MUW;
e) Pobieranie danych ze wszystkich czujników pogodowych i punktów pomiaru poziomu wody w ciekach wodnych na żądanie MUW (w przypadku zamontowania czujników);
f) Uruchamianie alarmów na wszystkich i wybranych PA;
g) Uruchamianie komunikatów głosowych na wszystkich i wybranych PA;
h) Uruchamianie komunikatów głosowych nadawanych przez mikrofon na wszystkich i wybranych PA;
i) Uruchomienie komunikatu „text to speech" na wszystkich i wybranych PA;
j) Zatrzymanie ogłaszania alarmów / komunikatów na wszystkich PA;
k) Zatrzymanie alarmów / komunikatów na wybranych PA;

Dofinansowanie Projektu pn.: „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego" jest realizowane przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Regionalnego Programu Operacyjnego Województwa Mazowieckiego 2014-2020 (Oś. Priorytetowa V Gospodarka przyjazna środowisku, działania 5.1. Dostosowanie do zmian klimatu)

4. Funkcjonalność interfejsu wymiany danych w relacji SP do MUW (SP ->MUW) (wytyczne w WebApi SAOL MUW):
   a) Automatyczne uwierzytelnienie elementu SP w systemie MUW;
   b) Autoryzacja operatora (okno logowania w systemie SP wyświetlone w centrali MUW);
   c) Synchronizacja urządzeń z SP do MUW (w przypadku zamontowania takich urządzeń) w zakresie nie mniejszym niż:
      - Jednoznaczny identyfikator systemowy urządzenia (indywidualne ID);
      - Dane adresowe obiektu, numer słupa na którym znajduje się urządzenie;
      - Współrzędne geograficzne obiektu;
      - Status urządzenia (sprawny / niesprawny / aktywny / w serwisie);
      - Dane pomiarowe z czujników pogodowych i poziomu wody.
   d) Synchronizacja danych o PA (punktach alarmowania) w czasie rzeczywistym do MUW po każdym automatycznym lub ręcznym teście/odpytaniu urządzenia. WAN jako medium priorytetowe, w przypadku braku łączności WAN informację należy przekazać przez RADIO;
   e) Przekazanie informacji o lokalnym(ręcznym) włączeniu alarmu w PA niezwłocznie po jego włączeniu;
   f) Przekazanie informacji o włączeniu alarmu / komunikatu w PA niezwłocznie po jego włączeniu (informacja zawiera typ i rodzaj ogłaszanego alarmu / komunikatu oraz zakres PA które obejmuje).
5. Synchronizacja danych o PA i urządzeniach peryferyjnych: Synchronizacja odbywa się w czasie rzeczywistym, w przypadku utraty połączenia MUW ma możliwość synchronizacji danych na żądanie.
6. Specyfikacja sygnałów alarmowych:
   1) Sygnał ciągły 3 minutowy;
   2) Sygnał modulowany 3 minutowy.
   3) Sygnał ciągły 1 minutowy.
   4) Sygnał predefiniowany.
   5) Komunikat głosowy „na żywo".
   6) Test.
   7) Stop ALARM.
   8) Stop KOMUNIKAT.
   9) Stop Test.
   10) „Tex to speach" w trybie bez połączenia z internetem.

Sporządził: Marcin Siatkowski

Akceptuję: Dyrektor Wydziału Zarządzania Kryzysowego dr Robert Dynak

Dofinansowanie Projektu pn.: „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego" jest realizowane przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Regionalnego Programu Operacyjnego Województwa Mazowieckiego 2014-2020 (Oś. Priorytetowa V Gospodarka przyjazna środowisku, działania 5.1. Dostosowanie do zmian klimatu)

Załącznik nr 2
do opisu przedmiotu zamówienia
dla zadania pn. „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego"

WebAPI SAOL MUW

SAOL-RSWS

System Alarmowania i Ostrzegania Ludności
Mazowiecki Urząd Wojewódzki w Warszawie
Plac Bankowy 3/5, 05-077 Warszawa
+48 22 692 64 71
kjanicki@mazowieckie.pl
www.mazowieckie.pl
Opracował: inż. Tomasz Marcinkowski
Wersja WebAPI: 3.2

Dofinansowanie Projektu pn.: „Budowa systemu wczesnego ostrzegania i alarmowania powiatu pułtuskiego" jest realizowane przez Unię Europejską ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Regionalnego Programu Operacyjnego Województwa Mazowieckiego 2014-2020 (Oś. Priorytetowa V Gospodarka przyjazna środowisku, działania 5.1. Dostosowanie do zmian klimatu)

WebAPI SAOL MUW

SAOLRSWS

System Alarmowania i Ostrzegania Ludności
Mazowiecki Urząd Wojewódzki w Warszawie
Plac Bankowy 3/5, 05-077 Warszawa
+48 22 692 64 71
kjanicki@mazowieckie.pl
www.mazowieckie.pl
Opracował: inż. Tomasz Marcinkowski
Wersja WebAPI: 3.2

Spis Treści

1 API SAOL MUW – w pigułce

WebAPI SAOL MUW jest narzędziem pozwalającym na komunikację centrali wojewódzkiej z systemami obcymi. API systemowe pozwala na wymianę asynchroniczną i synchroniczną pomiędzy SAOL MUW a systemami obcymi. W dokumencie znajduje się zbiór komend oraz sposób komunikacji w jaki należy komunikować się z SAOL MUW. Zawarta w dokumencie specyfikacja pozwala na pełną integrację systemów obcych z SAOL MUW.

2 Komunikacja – podstawowe informacje

WebAPI SAOL MUW widnieje pod publicznym adresem IP. W dalszej części dokumentu będziemy posługiwać się skrótem „IP" , które będzie równe zapisowi „http://adres_ip_muw" lub „https://adres_ip_muw". Wszystkie funkcje API zostały stworzone w kontrolerze o nazwie *eth_api_inne.php* , do którego należy odwoływać się w następujący sposób:

```
IP/eth_api_inne/nazwa_wykonywanej_funkcji
```

WebAPI SAOL MUW to aplikacja serwerowa, w dalszej części dokumentu SAOL MUW będzie określane jako *„serwer"*, a system obcy jako *„klient"*.

Wyróżniamy dwa sposoby uwierzytelnienia klienta:
- bez autoryzacji
- z autoryzacją

Dane do serwera przekazywane są za pomocą metody POST(urlc) na wskazany adres IP, kontroler i funkcję.

TCP jest protokołem działającym w trybie klient-serwer. Serwer oczekuje na nawiązanie połączenia na określonym porcie. Klient inicjuje połączenie do serwera. W protokole TCP do nawiązania połączenia pomiędzy dwoma hostami wykorzystywana jest procedura nazwana *three-way handshake*. W sytuacji normalnej jest ona rozpoczynana, gdy host A chce nawiązać połączenie z hostem B, procedura wygląda następująco:

- host A wysyła do hosta B segment SYN wraz z informacją o dolnej wartości numerów sekwencyjnych używanych do numerowania segmentów wysyłanych przez host A (np. 100) a następnie przechodzi w stan SYN-SENT,
- host B, po otrzymaniu segmentu SYN, przechodzi w stan SYN-RECEIVED i, jeżeli również chce nawiązać połączenie, wysyła hostowi A segment SYN z informacją o dolnej wartości numerów sekwencyjnych używanych do numerowania segmentów wysyłanych przez host B (np. 300) oraz segment ACK z polem numeru sekwencji ustawionym na wartość o jeden większą niż wartość pola sekwencji pierwszego segmentu SYN hosta A, czyli 101.
- host A, po odebraniu segmentów SYN i ACK od hosta B przechodzi w stan ESTABLISHED i wysyła do niego segment ACK potwierdzający odebranie segmentu SYN (numer sekwencji ustawiony na 301)
- host B odbiera segment ACK i przechodzi w stan ESTABLISHED
- host A może teraz rozpocząć przesyłanie danych

Używany format daty to czas uniksowy, czas POSIX – system reprezentacji czasu mierzący liczbę sekund od 1970 roku UTC, czyli od chwili zwanej początkiem epoki Uniksa (ang. *Unix Epoch*). Nie uwzględnia sekund przestępnych, zatem rzeczywista liczba sekund jakie upłynęły od początku epoki Uniksa jest większa o liczbę sekund przestępnych. W systemie

operacyjnym Unix i pochodnych czas jest przedstawiany jako 32-bitowa liczba sekund, które upłynęły od 1 stycznia 1970. Daną tę interpretuje się jako liczbę ze znakiem (ang. *signed integer*), w której wartości ujemne nie są wykorzystywane, dlatego dostępny przedział czasu wynosi $2^{31}-1$ sekund, co daje wartość równą 2 147 483 647. Pierwsze 10^9 sekund od początku epoki Uniksa upłynęło 9 września 2001, godz. 01:46:40 GMT. Chwilę tę nazwano "Unix billennium". Systemy uniksowe były odporne na tzw. problem roku 2000: 32-bitowy Unix time wyczerpie się 19 stycznia 2038 o godz. 03:14:07 UTC – wtedy pojawi się problem roku 2038 . Obecnie trwają prace, które mają wyeliminować problem roku 2038. Niektóre strony internetowe umożliwiają śledzenie czasu uniksowego na bieżąco, zaś w aplikacji mobilnej Google Play można ustawić czas uniksowy na stronę główną. W dalszej części dokumentu czas w formacie uniksowym będzie określany jako „*date*".

Wszystkie dane pól wysyłki(ang. Post fields) w całym WebAPI SAOL MUW są jednostkowo(każde z osobna) szyfrowane metodą base64 przed wysyłką. Wyjątkami są:
- pole/klucz „*session*" nazywany w dokumencie „session_id"
- pole/klucz PIN
- pole/klucz „*check*" – suma kontrolna

3 Uwierzytelnianie nowego klienta

3.1 Przekazanie danych do MUW

Pierwszym etap dodawania nowego klienta do serwera jest przekazanie adresu IP klienta do obsługi systemu SAOL MUW. Obsługa SAOL MUW wprowadza adres IP klienta do firewall'a, aby klient uzyskał możliwość jakiejkolwiek komunikacji z serwerem.
Obsługa SAOL MUW wygeneruje dla dedykowanego adresu IP klienta kod uwierzytelniający, który będzie niezbędny do prawidłowej komunikacji z SAOL MUW i przekaże go obsłudze klienta.
Do komunikacji między centralami niezbędny jest również klucz licencyjny, który klient otrzymuje od MUW lub wykupuje klucz licencyjny u producenta lub dystrybutora systemu SAOL MUW i przekazuje obsłudze MUW.

3.2 Pierwszy pakiet danych - wysyłka

Po otrzymaniu kodu uwierzytelniającego należy potwierdzić łączność wysyłając odpowiednią paczkę danych na adres:

IP/eth_api_inne/potwierdzam_uwierzytelnienie_klient

Należy przesłać dane zakodowane poprzez algorytm szyfrujący.
**Nazwa pola wysyłki – „session" -> dane: wygenerowany losowo unikalny identyfikator komendy składający się z alfa numerycznych znaków(wyłączając polskie znaki dialektyczne) o stałej długości 13 znaków.

Nazwa pola wysyłki - „date_send" -> dane: date (zakodowane logarytmem szyfrującym base64)

Nazwa pola wysyłki – „PIN" -> dane -> string utworzony z danych:

*date_send(kodowane base64), ***kod uwierzytelniający(string odwrócony – pisany od tyłu oraz kodowany base64),session

Podany string należy zakodować przed wysyłką metodą szyfrującą MD5 i następnie metodą szyfrującą SHA1.

Nazwa pola wysyłki – „check" -> dane: string składający się z pól:

**session,date_send,PIN*

Podany string należy zakodować przed wysyłką metodą szyfrującą SHA1 i następnie metodą szyfrującą MD5.

*String pisany jest w sposób ciągły, nie występują znaki interpunkcyjne pomiędzy seriami danymi, należy zachować kolejność przekazywanych danych oraz wielkości znaków
** W dalszej części dokumentu string składający się z alfa numerycznych znaków(wyłączając polskie znaki dialektyczne) będzie nazywany „session_id"
*** kod uwierzytelniający(string odwrócony – pisany od tyłu oraz kodowany base64) w dalszej części dokumentu będzie nazywany „OB_pin"

3.3 Pierwszy pakiet danych - odbiór

Po poprawnym odebraniu danych przez serwer zostaną zwrócone dane na adres IP klienta:
Adres odpowiedzi:

IP_klient/eth_api/uwierzytelnienie

Nazwa pola odbioru – „session_id" -> dane: wygenerowany losowo unikalny identyfikator komendy składający się z alfa numerycznych znaków(wyłączając polskie znaki dialektyczne) o stałej długości 13 znaków.

Nazwa pola odbioru – „date_get" -> dane: date – data do synchronizacji czasu - serwer -> klient, klient -> serwer

Nazwa pola odbioru – „PIN" -> dane: string utworzony z danych:

* klucz_licencyjny,session_id ,date(kodowane base64), kod uwierzytelniający(string odwrócony – pisany od tyłu oraz kodowany base64)

Podany string jest zakodowany przed wysyłką metodą szyfrującą MD5 i następnie metodą szyfrującą SHA1.

Nazwa pola odbioru – „check" -> dane -> string składający się z pól:

* session_id,*date_send,PIN*

Podany string jest zakodowany przed wysyłką metodą szyfrującą SHA1 i następnie metodą szyfrującą MD5.

*String pisany jest w sposób ciągły, nie występują znaki interpunkcyjne pomiędzy seriami danymi

Nazwa pola wysyłki lub nazwa pola odbioru w dalszej części dokumenty będzie nazywany *„kluczem"*.

3.4 Algorytm odbierania danych z serwera

Algorytm klienta odbierający dane z serwera musi sprawdzać poprawność otrzymywanych danych. Po stronie klienta należy sprawdzić:

1. Adres IP nadawcy - jeśli inny od adresu IP serwera dane powinny zostać natychmiast odrzucone, a adres IP, który próbował się komunikować zapisany do rejestru i wyświetlony obsłudze oprogramowaniu klienta.
2. Przyrównać wartość PINu odebranego, z PINem wygenerowanym z danych odebranych w pozostałych paczkach danych oraz z danymi zapisanymi tj. kod uwierzytelniający.
3. Sprawdzenie sumy kontrolnej, czy paczka danych nie uległa deformacji podczas przesyłania.

4 Dodanie nowego obiektu do MUW

Dodawanie nowego obiektu oraz nowych użytkowników do MUW wykorzystuję metodę podwójnego uwierzytelnienia. Są to opcje wymagające dodatkowego bezpieczeństwa przesyłanych danych.

4.1 Uwierzytelnienie połączenia

W pierwszym etapie dodawania nowego obiektu do MUW, klient zwraca się z żądaniem do serwera o kod autoryzacyjny połączenia. Serwer generuje jednorazowy kod autoryzacyjny składający się z 13 znaków alfa numerycznych(z wyłączeniem polskich znaków dialektycznych) i otwiera bramę połączenia na 5 sekund. Po upływie 5 sekund brama autoryzacyjna zostaje zamknięta, a kod autoryzacyjny traci ważność.

4.2 Jednorazowy kod autoryzacyjny połączenia

4.2.1 Żądanie klienta

Klient wysyła żądanie na adres:

IP/eth_api_inne/autoryzacja_polaczenia

Należy przesłać dane zakodowane poprzez algorytm szyfrujący.

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: string „AUTORYZACJA" (z zachowaniem wielkości znaków)
*Klucz – „PIN" -> dane -> string utworzony z danych:

klucz_licencyjny,session_id,zadanie,OB_pin,date

*Podany string należy zakodować przed wysyłką metodą szyfrującą MD5 i następnie metodą szyfrującą SHA1, w dalszej części dokumentu PIN będzie zawsze kodowany tą metodą. String pisany jest w sposób ciągły, nie występują znaki interpunkcyjne pomiędzy seriami danymi, należy zachować kolejność przekazywanych danych oraz wielkości znaków.

**Klucz – „check" -> dane -> string utworzony z danych:

session_id,date,zadanie,PIN

** Dane w polu „check" należy zakodować przed wysyłką metodą szyfrującą SHA1 i następnie metodą szyfrującą MD5, w dalszej części dokumentu klucz „check" będzie zawsze kodowany tą metodą. Na dane pola „check" składają się wszystkie klucze przesyłane w danym pakiecie danych.
String pisany jest w sposób ciągły, nie występują znaki interpunkcyjne pomiędzy seriami danymi, należy zachować kolejność przekazywanych danych oraz wielkości znaków.

4.2.2 *Odpowiedź serwera*

Serwer po odebraniu prawidłowej paczki danych odpowiada na żądanie klienta danymi:

Adres odpowiedzi:

IP_klient/eth_api/autoryzacja_polaczenia_good

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „pass" -> dane: string odwrócony (pisany od tyłu np. qwerty1234567 -> 7654321ytrewq)oraz kodowany base64, składający się z 13 znaków alfa numerycznych(z wyłączeniem polskich znaków dialektycznych)
Klucz – „PIN" -> dane -> string utworzony z danych: *klucz_licencyjny,session_id, pass, OB_pin, date*
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,pass,PIN*

Przesłane przez serwer dane zawarte w kluczu „pass" należy zapisać i wysłać w kolejnym żądaniu do serwera.

4.3 Przesłanie danych nowego punku alarmowania

Przesłany jednorazowy kod autoryzacyjny połączenie z klucza „pass", jest ważny 5 sekund, należy w tym czasie wysłać dane przyłączenia obiektu do SAOL MUW.

Do przesłania danych należy utworzyć tablicę asocjacyjną:
Klucz – „session_id" -> dane: session_id
Klucz – „centrala_identity" -> dane z systemu obcego: identyfikator przyłączanej centrali składający się z 13 znaków alfa numerycznych(z wyłączeniem polskich znaków dialektycznych)
Klucz – „obiekt_oddzial" -> dane z systemu obcego: string do 100 znaków
Klucz – „obiekt_obiekt" -> dane z systemu obcego: string do 500 znaków
Klucz – „obiekt_ulica" -> dane z systemu obcego: string do 45 znaków
Klucz – „obiekt_miasto" -> dane z systemu obcego: string do 35 znaków
Klucz – „obiekt_kod" -> dane z systemu obcego: string do 10 znaków
Klucz – „obiekt_woj" -> dane z systemu obcego: string do 50 znaków
Klucz – „obiekt_kraj" -> dane z systemu obcego: string do 30 znaków
Klucz – „obiekt_faks" -> dane z systemu obcego: string do 32 znaków
Klucz – „obiekt_mail" -> dane z systemu obcego: string do 50 znaków
Klucz – „obiekt_www" -> dane z systemu obcego: string do 50 znaków
Klucz – „obiekt_nazwa" -> dane z systemu obcego: string do 100 znaków
Klucz – „obiekt_tel" -> dane z systemu obcego: string do 50 znaków
Klucz – „obiekt_typ" -> dane z systemu obcego: string do 150 znaków
Klucz – „obiekt_ip" -> dane z systemu obcego: string do 20 znaków

Klucz – „obiekt_adres_radiowy" -> dane z systemu obcego: string do 4 znaków
Klucz – „podstrona" -> dane z systemu obcego: według istniejących unikalnych identyfikatorów powiatów/miast na prawach powiatu(załącznik 4)
Klucz – „miasto_obszar" -> dane z systemu obcego: string do 50 znaków
Klucz – „miasto_obszar_identity" -> dane z systemu obcego: identyfikator nowo powstałego obszaru/miasta składający się z 13 znaków alfa numerycznych(z wyłączeniem polskich znaków dialektycznych)
Klucz – „wojewodztwo" -> ID województwa : załącznik 4 do WebAPI MUW SAOL
Klucz – „kod_wersji" -> dane z systemu obcego: kod wersji to numer porządkowy wysyłanych danych, rozpoczynający się od 1 i każdorazowo rosnący o 1 int(255).

Dane które należy wysłać na adres:

IP/eth_api_inne/obiekt_eth_insert

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „data_send" -> dane: Dane z tablicy asocjacyjnej należy zamienić na stringa (funkcja PHP serialize())
Klucz – „pass" -> dane: jednorazowy kod uwierzytelniający przesyłany na żądanie klienta od serwera, pisany od przodu(odebrany: 7654321ytrewq, wysłać: qwerty1234567)
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,data_send,pass,OB_pin,date

Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,data_send,pass,PIN*

Serwer odpowie danymi w przypadku prawidłowego odbioru paczki danych:
Adres odpowiedzi:

IP_klient/eth_api/potwierdzenie

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „old_session" -> dane: kod identyfikacyjny żądanie klienta, na które serwer odpowiada
Klucz – „kod" -> dane: kod_wersji
Klucz – „PIN" -> dane -> string utworzony z danych: *klucz_licencyjny,session_id, old_session, kod,OB_pin, date*
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,old_session,kod,PIN*

Klient musi zapisać czy odpowiedź na jego żądanie została poprawnie odebrana, w *przeciwnym wypadku po stronie klienta leży obowiązek kontroli wersji danych zapisanych* w serwerze. Klient powinien w przypadku błędu w żądaniu, ponawiać co jakiś czas żądanie, lub monitorować połączenie z serwerem i w przypadku odzyskania łączności natychmiast wysłać paczkę danych ponownie.

5 Użytkownicy*

Użytkownicy SAOL MUW mają możliwość logowania się zdalnie do central podrzędnych wykorzystując protokół komunikacyjny http lub https. Logowanie do central podrzędnych odbywa się za pomocą takich samych danych jak w systemie SAOL MUW. Związku z powyższym dodawanie użytkownika w SAOL MUW musi zostać zapisane również w systemach podrzędnych, w zakresie nie mniejszym niż obowiązkowe dane do logowania takie jak login i hasło. Hasło w systemie SAOL MUW jest kodowane metodą SHA1 i przechowywane w bazie danych. SAOL MUW pilnuje kodu wersji w przypadku dodawania i edycji użytkowników.

* Nie wymagane w integracji MUW z systemami obcymi

6 Integracja urządzeń

Administratorzy oraz operatorzy systemu SAOL MUW muszą miedź pełną wizualizację central oraz innych urządzeń zainstalowanych w MUW. Muszą znać położenie urządzeń(współrzędne na mapie) oraz obecny status urządzenia, czy jest sprawne, czy nie trwa na nim alarm lub inne. Dodatkowo SAOL MUW ma możliwość na żądanie ściągnięcia poniższych danych.

6.1 Syrena wirnikowa

6.1.1 Żądanie danych przez serwer

Adres odbiorcy, na który serwer wysyła żądanie:

IP_klient/eth_api/get_produkt_eth

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „GET_PRODUKT_RUW"
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,zadanie,OB_pin,date
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,PIN*

6.1.2 Odpowiedź i wysyłanie danych przez klienta

6.1.2.1 Odpowiedź

Klient zwraca w odpowiedzi na żądanie serwera dane:
„session_id" -> dane: session_id
„old_session" -> dane: kod identyfikacyjny żądanie serwera, na które klient odpowiada
„date" -> dane: date
„produkt_identity" -> dane: string 13 znaków
„produkt_nazwa" -> dane: string do 70 znaków
„produkt_symbol" -> dane: string do 10 znaków
„produkt_adres_radiowy" -> dane: string do 4 znaków
„produkt_adres" -> dane: string do 150 znaków
„urzadzenie_typ_id" -> dane: (według wytycznych – załącznik 2)
„produkt_opis" -> dane: date
„produkt_x" -> dane: string do 20 znaków
„produkt_y" -> dane: string do 20 znaków
„produkt_azymut" -> dane: intiger do 4 znaków

„produkt_filtr" -> dane: (według wytycznych – załącznik 3)
„produkt_centrala_identity" -> dane: identyfikator centrali
„produkt_active" -> dane: intiger 1 znak 0 lub 1-active
„produkt_data_dodania" -> dane: date
„produkt_data_modyfikacji" -> dane: date
„produkt_modyfikacja_przez" -> dane: login użytkownika
„produkt_ilosc_wyswietlen" -> dane: intiger(255)
„produkt_icon_status" -> dane: 0 – nowa, 1 – sprawna, 2 – niesprawna, 3 – brak łączności, 4 – ogłasza komunikat alarm
„produkt_tryb_pracy" -> dane: string N – normalny C – trening
„produkt_wykonywane_zadanie" -> dane: string do 3 znaków (załącznik 6)
„produkt_sektor" -> dane: string (zakres sektorów A B C D E F G H),
„produkt_data_testu" -> dane: date
„produkt_ipk" -> dane: string do 20 znaków
„produkt_data_nadania" -> dane: date
„produkt_grupa" -> dane: pole puste
„produkt_typ_ster" -> dane: string do 255 znaków
„produkt_pamiec_alarmu" -> int 0,1 lub 2, 0 – brak, 1 – pamięć syr.sprawna, 2 – pamięć syr. Z usterką
„produkt_data_instalacji" -> dane: date
„produkt_data_uruchomienia" -> dane: date
„produkt_data_bezp_kondycji_aku" -> dane: date
„produkt_typ_akumulatora" -> dane: string do 150 znaków
„produkt_kod_wersji" -> dane: intiger(255)

Powyższe dane muszą zostać rozdzielone separatorem „<par>" . Kolejne urządzenia/produkty rozdzielamy separatorem „<spacja>". Należy zachować podaną powyżej kolejność przekazywanych danych i połączyć wszystkie dane o urządzeniach w jednego stringa i zwrócić w odpowiedzi na żądanie serwera.

6.1.2.2 *Wysyłanie*

System SAOL MUW zbudowany jest z myślą o wielu podrzędnych systemach. Zaimplementowane algorytmy pilnują priorytetów w komunikacji między systemem głównym, a systemami podrzędnymi. Założone jest, iż w przypadku nadawania alarmu w którymkolwiek z podsystemów wszystkie akcje wymiany dużych paczek danych zostają wstrzymane. Z tego powodu korzystamy z opcji podwójnej autoryzacji. Jeśli algorytm serwera zezwoli na transfer danych, zostanie zwrócony kod uwierzytelniający połączenie, w przeciwnym wypadku serwer zwróci komunikat „WAIT.CZAS".
CZAS to na przykład liczba 150 lub 250 lub inna z zakresu 001 do 999 oznaczająca ile czasu połączenie nie będzie możliwe. Jeśli paczka danych jest błędna serwer nie odpowie.

Klient wysyła żądanie na adres:

IP/eth_api_inne/autoryzacja_polaczenia

Należy przesłać dane zakodowane poprzez algorytm szyfrujący.

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date

Klucz – „zadanie" -> dane: string „AUTO_PROD" (z zachowaniem wielkości znaków)
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,zadanie,OB_pin,date
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,PIN*

Po otrzymaniu identyfikatora sesji Klient wysyła żądanie na adres:

IP/eth_api_inne/send_produkt

Do przesłania danych należy utworzyć tablicę asocjacyjną:
Klucz – „produkt_identity" -> dane: string 13 znaków
Klucz – „produkt_nazwa" -> dane: string do 70 znaków
Klucz – „produkt_symbol" -> dane: string do 10 znaków
Klucz – „produkt_adres_radiowy" -> dane: string do 4 znaków
Klucz – „produkt_adres" -> dane: string do 150 znaków
Klucz – „urzadzenie_typ_id" -> dane: (według wytycznych - załącznik 2)
Klucz – „produkt_opis" -> dane: date
Klucz – „produkt_x" -> dane: string do 20 znaków
Klucz – „produkt_y" -> dane: string do 20 znaków
Klucz – „produkt_azymut" -> dane: intiger do 4 znaków
Klucz – „produkt_filtr" -> dane: (według wytycznych - załącznik 3)
Klucz – „produkt_centrala_identity" -> dane: identyfikator centrali
Klucz – „produkt_active" -> dane: intiger 1 znak 0 lub 1-active
Klucz – „produkt_data_dodania" -> dane: date
Klucz – „produkt_data_modyfikacji" -> dane: date
Klucz – „produkt_modyfikacja_przez" -> dane: login użytkownika
Klucz – „produkt_ilosc_wyswietlen" -> dane: intiger(255)
Klucz – „produkt_icon_status" -> dane: 0 – nowa, 1 – sprawna, 2 – niesprawna, 3 – brak łączności, 4 – ogłasza komunikat alarm
Klucz – „produkt_tryb_pracy" -> dane: string N – normalny C - trening
Klucz – „produkt_wykonywane_zadanie" -> dane: string do 3 znaków (załącznik 6)
Klucz – „produkt_sektor" -> dane: string (zakres sektorów A B C D E F G H),
Klucz – „produkt_data_testu" -> dane: date
Klucz – „produkt_ipk" -> dane: string do 20 znaków
Klucz – „produkt_data_nadania" -> dane: date
Klucz – „produkt_grupa" -> dane: pole puste
Klucz – „produkt_typ_ster" -> dane: string do 255 znaków
Klucz – „produkt_pamiec_alarmu" -> int 0,1 lub 2, 0 – brak, 1 – pamięć syr.sprawna, 2 – pamięć syr. z usterką
Klucz – „produkt_data_instalacji" -> dane: date
Klucz – „produkt_data_uruchomienia" -> dane: date
Klucz – „produkt_data_bezp_kondycji_aku" -> dane: date
Klucz – „produkt_typ_akumulatora" -> dane: string do 150 znaków
Klucz – „produkt_kod_wersji" -> dane: intiger(255)

Dane urządzenia zapisujemy w drugą tablicę, dwuwymiarową asocjacyjną:
Klucz 1(węzeł) -> identyfikator_urzadzenia ->Klucz 2(węzeł) -> produkt_kod_wersji -> dane:

serializowane dane o urządzeniu(php serialize())
Przykład:
$tablica['fhsjkadur4k2k'][3]= '
a:2:{s:3:"produkt_nazwa";s:5:"Syrena";s:4:"produkt_symbol";s:6:"SW-01"{...};}';

Wszystkie dane z drugiej tablicy ponownie serializujemy i przekazujemy do zmiennej o kluczu „*data_send*".
Klucz – „session_id" -> dane: session_id
Klucz – „old_session" -> dane: kod identyfikacyjny żądanie serwera, na które klient odpowiada
Klucz – „date" -> dane: date
Klucz – „data_send" -> dane: Dane z drugiej tablicy asocjacyjnej (serialize())
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,old_session,date,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych:
session_id,old_session,date,data_send,PIN

6.2 Syrena elektroniczna

6.2.1 Żądanie danych przez serwer

Adres odbiorcy, na który serwer wysyła żądanie:

IP_klient/eth_api/get_produkt_eth

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „GET_PRODUKT_EL"
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,OB_pin,date
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,PIN*

6.2.2 Odpowiedź i wysyłanie danych przez klienta

6.2.2.1 Odpowiedź

Klient zwraca w odpowiedzi na żądanie serwera dane:
„session_id" -> dane: session_id
„old_session" -> dane: kod identyfikacyjny żądanie serwera, na które klient odpowiada
„date" -> dane: date
„produkt_identity" -> dane: string 13 znaków
„produkt_nazwa" -> dane: string do 70 znaków
„produkt_symbol" -> dane: string do 10 znaków
„produkt_adres_radiowy" -> dane: string do 4 znaków
„*produkt_adres*" -> *dane: string do 150 znaków*
„urzadzenie_typ_id" -> dane: (według wytycznych - załącznik 2)
„produkt_opis" -> dane: date
„produkt_x" -> dane: string do 20 znaków
„produkt_y" -> dane: string do 20 znaków
„produkt_azymut" -> dane: intiger do 4 znaków
„produkt_filtr" -> dane: (według wytycznych - załącznik 3)

„produkt_centrala_identity" -> dane: identyfikator centrali
„produkt_active" -> dane: intiger 1 znak 0 lub 1-active
„produkt_data_dodania" -> dane: date
„produkt_data_modyfikacji" -> dane: date
„produkt_modyfikacja_przez" -> dane: login użytkownika
„produkt_ilosc_wyswietlen" -> dane: intiger(255)
„produkt_icon_status" -> dane: 0 – nowa, 1 – sprawna, 2 – niesprawna, 3 – brak łączności, 4 – ogłasza komunikat alarm
„produkt_tryb_pracy" -> dane: string N – normalny C - trening
„produkt_wykonywane_zadanie" -> dane: string do 3 znaków (załącznik 6)
„produkt_sektor" -> dane: string (zakres sektorów A B C D E F G H),
„produkt_data_testu" -> dane: date
„produkt_ipk" -> dane: string do 20 znaków
„produkt_data_nadania" -> dane: date
„produkt_grupa" -> dane: pole puste
„produkt_typ_ster" -> dane: string do 255 znaków
Klucz – „produkt_pamiec_alarmu" -> int 0,1 lub 2, 0 – brak, 1 – pamięć syr.sprawna, 2 – pamięć syr. z usterką
„produkt_data_instalacji" -> dane: date
„produkt_data_uruchomienia" -> dane: date
„produkt_data_bezp_kondycji_aku" -> dane: date
„produkt_typ_akumulatora" -> dane: string do 150 znaków
„produkt_kod_wersji" -> dane: intiger(255)

Powyższe dane muszą zostać rozdzielone separatorem „<par>" . Kolejne urządzenia/produkty rozdzielamy separatorem „<spacja>". Należy zachować podaną powyżej kolejność przekazywanych danych i połączyć wszystkie dane o urządzeniach w jednego stringa i zwrócić w odpowiedzi na żądanie serwera.

6.2.2.2 *Wysyłanie*

System SAOL MUW zbudowany jest z myślą o wielu podrzędnych systemach. Zaimplementowane algorytmy pilnują priorytetów w komunikacji między systemem głównym, a systemami podrzędnymi. Założone jest, iż w przypadku nadawania alarmu w którymkolwiek z podsystemów wszystkie akcje wymiany dużych paczek danych zostają wstrzymane. Z tego powodu korzystamy z opcji podwójnej autoryzacji. Jeśli algorytm serwera zezwoli na transfer danych, zostanie zwrócony kod uwierzytelniający połączenie, w przeciwnym wypadku serwer zwróci komunikat „WAIT.CZAS".
CZAS to na przykład liczba 150 lub 250 lub inna z zakresu 001 do 999 oznaczająca ile czasu połączenie nie będzie możliwe. Jeśli paczka danych jest błędna serwer nie odpowie.

Klient wysyła żądanie na adres:

IP/eth_api_inne/autoryzacja_polaczenia

Należy przesłać dane zakodowane poprzez algorytm szyfrujący.

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: string „AUTORYZACJA" (z zachowaniem wielkości znaków)

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,zadanie,OB_pin,date
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,PIN*

Po otrzymaniu identyfikatora sesji :
Klient wysyła żądanie na adres:

IP/eth_api_inne/send_produkt

Do przesłania danych należy utworzyć tablicę asocjacyjną:
Klucz – „produkt_identity" -> dane: string 13 znaków
Klucz – „produkt_nazwa" -> dane: string do 70 znaków
Klucz – „produkt_symbol" -> dane: string do 10 znaków
Klucz – „produkt_adres_radiowy" -> dane: string do 4 znaków
Klucz – „produkt_adres" -> dane: string do 150 znaków
Klucz – „urzadzenie_typ_id" -> dane: (według wytycznych - załącznik 2)
Klucz – „produkt_opis" -> dane: date
Klucz – „produkt_x" -> dane: string do 20 znaków
Klucz – „produkt_y" -> dane: string do 20 znaków
Klucz – „produkt_azymut" -> dane: intiger do 4 znaków
Klucz – „produkt_filtr" -> dane: (według wytycznych - załącznik 3)
Klucz – „produkt_centrala_identity" -> dane: identyfikator centrali
Klucz – „produkt_active" -> dane: intiger 1 znak 0 lub 1-active
Klucz – „produkt_data_dodania" -> dane: date
Klucz – „produkt_data_modyfikacji" -> dane: date
Klucz – „produkt_modyfikacja_przez" -> dane: login użytkownika
Klucz – „produkt_ilosc_wyswietlen" -> dane: intiger(255)
Klucz – „produkt_icon_status" -> dane: 0 – nowa, 1 – sprawna, 2 – niesprawna, 3 – brak łączności, 4 – ogłasza komunikat alarm
Klucz – „produkt_tryb_pracy" -> dane: string N – normalny C - trening
Klucz – „produkt_wykonywane_zadanie" -> dane: string do 3 znaków (załącznik 6)
Klucz – „produkt_sektor" -> dane: string (zakres sektorów A B C D E F G H),
Klucz – „produkt_data_testu" -> dane: date
Klucz – „produkt_ipk" -> dane: string do 20 znaków
Klucz – „produkt_data_nadania" -> dane: date
Klucz – „produkt_grupa" -> dane: pole puste
Klucz – „produkt_typ_ster" -> dane: string do 255 znaków
Klucz – „produkt_pamiec_alarmu" -> int 0,1 lub 2, 0 – brak, 1 – pamięć syr.sprawna, 2 – pamięć syr. z usterką
Klucz – „produkt_data_instalacji" -> dane: date
Klucz – „produkt_data_uruchomienia" -> dane: date
Klucz – „produkt_data_bezp_kondycji_aku" -> dane: date
Klucz – „produkt_typ_akumulatora" -> dane: string do 150 znaków
Klucz – „produkt_kod_wersji" -> dane: intiger(255)

Dane urządzenia zapisujemy w drugą tablicę, dwuwymiarową asocjacyjną:
Klucz 1 -> identyfikator_urządzenia -> Klucz 2 -> produkt_kod_wersji -> dane: serializowane dane o urządzeniu(php serialize())
Przykład:

$tablica['fhsjkadur4k2k'][3]= '
a:2:{s:3:"produkt_nazwa";s:5:"Syrena";s:4:"produkt_symbol";s:6:"SW-01"(...);}';

Wszystkie dane z drugiej tablicy ponownie serializujemy i przekazujemy do zmiennej o kluczu „*data_send*".
Klucz – „session_id" -> dane: session_id
Klucz – „old_session" -> dane: kod identyfikacyjny żądanie serwera, na które klient odpowiada
Klucz – „date" -> dane: date
Klucz – „data_send" -> dane: Dane z drugiej tablicy asocjacyjnej (serialize())
Klucz – „kod_wersji" -> dane: kod wersji
Klucz – „PIN" -> dane -> string utworzony z danych: *klucz_licencyjny,session_id,old_session, date, data_send,kod_wersji, OB_pin*
Klucz – „check" -> dane -> string utworzony z danych: *session_id,old_session,date, data_send,kod_wersji,PIN*

6.3 Stacja pogodowa

Integracja urządzenia typu „stacja pogodowa" jest analogiczna do 6.1 Syrena wirnikowa lub 6.2 Syrena elektroniczna , wyłączając klucze nie dotyczące tak jak między innymi „*produkt_azymut*", należy wysłać puste. Zadanie: „GET_PRODUKT_POG".

6.4 Czujnik skażeń

Integracja urządzenia typu „czujnik skażeń" jest analogiczna do 6.1 Syrena wirnikowa lub 6.2 Syrena elektroniczna ,wyłączając klucze nie dotyczące tak jak między innymi „*produkt_azymut*", należy wysłać puste. Zadanie: „GET_PRODUKT_CZS".

6.5 Zegar DCF

Integracja urządzenia typu „zegar DCF" jest analogiczna do 6.1 Syrena wirnikowa lub 6.2 Syrena elektroniczna ,wyłączając klucze nie dotyczące tak jak między innymi „*produkt_azymut*", należy wysłać puste. Zadanie: „GET_PRODUKT_DCF".

6.6 Limnimetry

Integracja urządzenia typu „limnimetr" jest analogiczna do 6.1 Syrena wirnikowa lub 6.2 Syrena elektroniczna ,wyłączając klucze nie dotyczące tak jak między innymi „*produkt_azymut*", należy wysłać puste. Zadanie: „GET_PRODUKT_CZW".

6.7 Centrale

Integracja urządzenia typu „centrala" jest analogiczna do 6.1 Syrena wirnikowa lub 6.2 Syrena elektroniczna ,wyłączając klucze nie dotyczące tak jak między innymi „*produkt_azymut*", należy wysłać puste. Zadanie: „GET_PRODUKT_CEN".

6.8 Wszystkie urządzenia w jednej paczce danych

Integracja wszystkich urządzeń jest analogiczna do 6.1 Syrena wirnikowa lub 6.2 Syrena elektroniczna ,wyłączając klucze nie dotyczące tak jak między innymi „*produkt_azymut*", należy wysłać puste. Zadanie: „GET_PRODUKT_ALL".

6.9 Pozostałe urządzenia

Integracja pozostałych urządzeń jest analogiczna do 6.1 Syrena wirnikowa lub 6.2 Syrena elektroniczna ,wyłączając klucze nie dotyczące tak jak między innymi „*produkt_azymut*", należy wysłać puste. System przewiduje rozbudowę o każdy rodzaj czujnika pomiarowego lub inne.

7 Zdarzenia systemowe

Serwer odbiera dane o zdarzeniach z systemów podrzędnych pod adresem:

IP/eth_api_inne/zdarzenie

7.1 Syrena wirnikowa

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-SW"
Klucz – „data_send" -> dane:
- identyfikator urządzenia
- status syreny: 1 – sprawna, 2 – niesprawna, 3 – brak łączności, 4 – ogłasza alarm(z uwzględnieniem ogłaszania alarmu np. 4A1; XX – spoczynek)
- pamięć alarmu : 0 – brak; 1 – sprawna syr. pamięć; 2- niesprawna syr. pamięć
- tryb pracy,(N – normalny, C - trening)
- zasilanie (brak, jest)
- informacja na temat przynależności do sektorów(zakres sektorów A B C D E F G H),

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.2 Syrena elektroniczna

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-SEL"
Klucz – „data_send" -> dane:
- identyfikator urządzenia
- status syreny: 1 – sprawna, 2 – niesprawna, 3 – brak łączności, 4 – ogłasza komunikat/alarm(z uwzględnieniem jaki ogłasza alarm, lub jaki ogłasza komunikat np. 4A2 – ogłasza określony alarm, 4K1 – ogłasza określony komunikat, XX - spoczynek),
- pamięć alarmu : 0 – brak; 1 – sprawna syr. pamięć; 2- niesprawna syr. pamięć
- tryb pracy(N – normalny, C - trening),
- informacje na temat zasilania 24V(brak, w normie),
- informacje na temat zasilania 12V(brak, w normie),
- drzwi do syreny(otwarte, zamknięte),
- zasilanie 230V(brak, jest),
- napięcie 24 V z dokładnością do jednego miejsca po przecinku),

- napięcie 12 V z dokładnością do jednego miejsca po przecinku,
- sprawność głośników zestaw 1(sprawne, niesprawne),
- sprawność głośników zestaw 2(sprawne, niesprawne),
- sprawność wzmacniaczy zestaw 1(sprawne, niesprawne),
- sprawność wzmacniaczy zestaw 2(sprawne, niesprawne),
- napięcie akumulatora po teście z dokładnością do jednego miejsca po przecinku (V),
- prąd ładowania akumulatora (A),
- sprawność generatora(sprawny, niesprawny)

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.3 Stacja pogodowa

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-PG"
Klucz – „data_send" -> dane:
- identyfikator urządzenia
- status stacji pogodowej: 1 – sprawna, 2 – niesprawna, 3 – brak łączności,
- data i godzina pomiaru(format *date*),
- temperatura w stopniach Celsjusza z dokładnością do jednego miejsca po przecinku,
- ciśnienie w hPa,
- wilgotność powietrza (%),
- siłę wiatru w m/s,
- kierunek wiatru,
- opady w mm/m2 z ostatniej godziny,
- opady w mm/m2 z ostatnich 24 godzin,

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem *interpunkcyjnym „ , "(przecinek)*.

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.4 Czujnik skażeń

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-CZG"
Klucz – „data_send" -> dane:
- identyfikator urządzenia
- status czujnika skażeń: 1 – sprawny, 2 – niesprawny, 3 – brak łączności,
- data i godzina pomiaru(format *date*),
- wartość średnia dawki promieniowania z ostatniej minuty w µSv/h
- wartość średnia dawki promieniowania z ostatniej godziny w µSv/h

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.5 Limnimetr

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-CZW"
Klucz – „data_send" -> dane:
- identyfikator urządzenia
- status wodomierza: 1 – sprawny, 2 – niesprawny, 3 – brak łączności,
- data i godzina pomiaru(format *date*),
- pętla prądowa (mA)

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: session_id,date,zadanie,data_send,PIN

7.6 Alarmy

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-ALARM"
Klucz – „data_send" -> dane:
- jaki alarm (od A1 do A8)
- czy zostanie nadany komunikat po alarmie(T – tak, N – nie jeśli tak to jaki np. TK1)
- głośność(1 do 9),
- data (format *date*),
- na ilu syrenach podjęta próba uruchomienia
- identyfikatory syren na których została podjęta próba uruchomienia oddzielone separatorem
„ <syr> "

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin, date
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.6.1 Alarm na pojedynczej syrenie

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-SYRALARM"
Klucz – „data_send" -> dane:
- jaki alarm (od A1 do A8)
- czy zostanie nadany komunikat po alarmie(T – tak, N – nie jeśli tak to jaki np. TK1)
- głośność(1 do 9),
- data (format *date*),
- identyfikator syreny

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.6.2 Fizyczne uruchomienie alarmu z przycisku w syrenie

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „MANUAL-SYRALARM"
Klucz – „data_send" -> dane:
- jaki alarm (od A1 do A8)
- czy zostanie nadany komunikat po alarmie(T – tak, N – nie jeśli tak to jaki np. TK1)
- głośność(1 do 9),
- data (format *date*),
- identyfikator syreny

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.6.3 Potwierdzenie włączenia alarmu w syrenie

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „AUTO-SYRALARM"
Klucz – „data_send" -> dane:
- potwierdzenie (1 – poprawnie, 0 – syrena uszkodzona)
- data (format *date*),
- identyfikator syreny

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.7 Komunikaty

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „IN-KOM"
Klucz – „data_send" -> dane:
- jaki komunikat (od K1 do K8) dla mikrofonu (UU)
- data (format *date*),
- na ilu syrenach podjęta próba uruchomienia
- identyfikatory syren na których została podjęta próba uruchomienia oddzielone separatorem „ <syr> "

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:

klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.7.1 Komunikat na pojedynczej syrenie

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „*IN-SYRKOM*"
Klucz – „data_send" -> dane:
- jaki alarm (od K1 do K8) dla mikrofonu (UU)
- data (format *date*),
- identyfikator syreny

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.7.2 Fizyczne uruchomienie komunikatu z przycisku w syrenie

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „MANUAL-SYRKOM"
Klucz – „data_send" -> dane:
- jaki alarm (od K1 do K8) dla mikrofonu (UU)
- data (format *date*),
- identyfikator syreny

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.7.3 Potwierdzenie włączenia komunikatu w syrenie

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „AUTO-SYRKOM"
Klucz – „data_send" -> dane:
- potwierdzenie (1 – poprawnie, 0 – syrena uszkodzona)
- data (format *date*),
- identyfikator syreny

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

7.8 Inne zdarzenia gdzie indziej nie sklasyfikowane

Ze względu na ciągły rozwój oprogramowania SAOL MUW lista obsługiwanych zdarzeń stale rośnie. Obsługa zdarzeń innych jest analogiczna do schematów połączeń zawartych w rozdziale 7.

8 Zarządzanie systemami podrzędnymi

Komunikacja między serwerem a klientem odbywa się za pomocą metody POST(curl). Serwer wysyła żądania na określone adresy IP, kontroler i funkcję. System jest uniwersalny, można wszystkie poniższe kontrolery i funkcje przekierować na własne w systemie klienta. Poniższa lista żądań zostaje wysyłana na wiele adresów IP, adres IP systemu powiatowego(jeżeli jest wpiętą centrala powiatowa w system) oraz na wszystkie adresy IP central miejskich, na których ma zostać uruchomiony alarm.
Centrala powiatowa podejmuje żądanie centrali MUW i również wysyła żądanie do central miejskich, centrala miejska podejmuje żądanie, które pierwsze do niej dotrze, czy to z centrali MUW czy z centrali POWIATOWEJ, dodatkowo potwierdza odebranie żądania na adres IP centrali powiatowej oraz adres IP centrali MUW.

8.1 Uruchomienie alarmów z MUW na wszystkich syrenach w powiecie/mieście

Adres odbioru:

```
IP_klient/eth_api/alarm_on
```

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „ON-ALARM"
Klucz – „data_send" -> dane:
- jaki alarm (od A1 do A8)
- czy zostanie nadany komunikat po alarmie(T – tak, N – nie jeśli tak to jaki np. TK1)
- głośność(1 do 9),
- data (format *date*),
- w ilu miastach podjęta próba uruchomienia
- w przypadku uruchomienia alarmu na kilku wybranych miastach w powiecie zostanie przesłany ciąg identyfikatorów miast/obszarów oddzielonych od siebie separatorem <mst>, gdy w całym powiecie string „CALE".

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.2 Uruchomienie alarmów z MUW na wybranych syrenach w powiecie/mieście

Adres odbioru:

```
IP_klient/eth_api/alarm_on
```

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „ON-SYRALARM"
Klucz – „data_send" -> dane:
- jaki alarm (od A1 do A8)
- czy zostanie nadany komunikat po alarmie(T – tak, N – nie jeśli tak to jaki np. TK1)
- głośność(1 do 9),

- data (format *date*),
- na ilu syrenach podjęta próba uruchomienia
- *<cen>numer_identyfikacyjny_cetrali</cen>* identyfikatory syren na których została podjęta próba uruchomienia oddzielone separatorem „ <syr> " (gdy wszystkie syreny są zaznaczone w mieście podajemy string ALL<syr>), obiekty oddzielamy od siebie parametrem <spacja>
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.3 Uruchomienie komunikatów głosowych z MUW na wszystkich syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/alarm_on

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „ON-KOM"
Klucz – „data_send" -> dane:
- jaki komunikat (od K1 do K8) dla mikrofonu UU
- głośność(1 do 9),
- data (format *date*),
- na ilu obiektach podjęta próba uruchomienia
- *w przypadku uruchomienia komunikatu na kilku wybranych miastach w powiecie zostanie* przesłany ciąg identyfikatorów miast/obszarów oddzielonych od siebie separatorem <mst>, gdy w całym powiecie string „CALE".
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.4 Uruchomienie komunikatów głosowych z MUW na wybranych syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/alarm_on

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „ON-SYRKOM"

Klucz – „data_send" -> dane:
- jaki komunikat (od K1 do K8) dla mikrofonu UU
- głośność(1 do 9),
- data (format *date*),
- na ilu syrenach podjęta próba uruchomienia
- <cen>*numer_identyfikacyjny_cetrali*</cen> identyfikatory syren na których została podjęta próba uruchomienia oddzielone separatorem „ <syr> " (gdy wszystkie syreny są zaznaczone w mieście podajemy string ALL<syr>), obiekty oddzielamy od siebie parametrem <spacja>
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.5 Ogłaszanie komunikatów głosowych nadawanych przez mikrofon z centrali MUW na wszystkich syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/alarm_on

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „ON-KOM"
Klucz – „data_send" -> dane:
- UU
- głośność(1 do 9),
- *data (format date)*,
- na ilu obiektach podjęta próba uruchomienia
- w przypadku uruchomienia komunikatu na kilku wybranych miastach w powiecie zostanie przesłany ciąg identyfikatorów miast/obszarów oddzielonych od siebie separatorem <mst>, gdy w całym powiecie string „CALE".
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

Systemy obce muszą włączyć nasłuchiwanie adresu IP:port_nadawania/saol, na którym zostaje rozpoczęte strumieniowanie audio.

8.6 Ogłaszanie komunikatów głosowych nadawanych przez mikrofon z centrali MUW na wybranych syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/alarm_on

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „ON-SYRKOM"
Klucz – „data_send" -> dane:
- UU
- głośność(1 do 9),
- data (format *date*),
- na ilu syrenach podjęta próba uruchomienia
- <cen>*numer_identyfikacyjny_cetrali*</cen> identyfikatory syren na których została podjęta próba uruchomienia oddzielone separatorem „ <syr> " (gdy wszystkie syreny są zaznaczone w mieście podajemy string *ALL*<syr>), obiekty oddzielamy od siebie parametrem <spacja>
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> *dane -> string utworzony z danych:*
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.7 Zatrzymanie ogłaszania alarmów i komunikatów alarmowych

8.7.1 Zatrzymanie alarmów z MUW na wszystkich syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/stop

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „STOP-ALARM"
Klucz – „data_send" -> dane:
- data (format *date*),
- na ilu obiektach podjęta próba zatrzymania
- w przypadku zatrzymania alarmu na kilku wybranych miastach w powiecie zostanie przesłany ciąg identyfikatorów miast/obszarów oddzielonych od siebie separatorem <mst>, gdy w całym powiecie string „CALE".
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.7.2 Zatrzymanie alarmów z MUW na wybranych syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/stop

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „STOP-SYRALARM"
Klucz – „data_send" -> dane:
- data (format *date*),
- na ilu syrenach podjęta próba zatrzymania
- <cen>*numer_identyfikacyjny_cetrali*</cen> identyfikatory syren na których została podjęta próba uruchomienia oddzielone separatorem „ <syr> " (gdy wszystkie syreny są zaznaczone w mieście podajemy string ALL<syr>), obiekty oddzielamy od siebie parametrem <spacja>
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.7.3 Zatrzymanie komunikatów głosowych z MUW na wszystkich syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/alarm_on

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „STOP-KOM"
Klucz – „data_send" -> dane:
- na ilu obiektach podjęta próba zatrzymania
- w przypadku zatrzymania komunikatu w kilku wybranych miastach w powiecie zostanie przesłany ciąg identyfikatorów miast/obszarów oddzielonych od siebie separatorem <mst>, gdy w całym powiecie string „CALE".
- data (format *date*),
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych: *klucz_licencyjny,session_id, date,zadanie, data_send,OB_pin*
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date, zadanie,data_send,PIN*

8.7.4 Zatrzymanie komunikatów głosowych z MUW na wybranych syrenach w powiecie/mieście

Adres odbioru:

IP_klient/eth_api/alarm_on

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date

Klucz – „zadanie" -> dane: „STOP-SYRKOM"
Klucz – „data_send" -> dane:
- data (format *date*),
- na ilu syrenach podjęta próba zatrzymania
- <cen>*numer_identyfikacyjny_cetrali*</cen> identyfikatory syren na których została podjęta próba uruchomienia oddzielone separatorem „ <syr> " (gdy wszystkie syreny są zaznaczone w mieście podajemy string ALL<syr>), obiekty oddzielamy od siebie parametrem <spacja>
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.8 Kasowanie pamięci alarmów

W przypadku ogłoszenia pełnego alarmu syreny w systemie MUW zostają oznaczone specjalną flagą. System SAOL MUW dysponuje funkcjami, które umożliwiają zarządzanie tym statusem.

8.8.1 Serwer -> klient

8.8.1.1 Na wszystkich syrenach

Adres odbioru:

IP_klient/eth_api/pamiec

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „STOP-PAMIEC"
Klucz – „data_send" -> dane:
- data (format *date*),
- string *ALL*,
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.8.1.2 Na wybranych syrenach

Adres odbioru:

IP_klient/eth_api/pamiec

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date

Klucz – „zadanie" -> dane: „STOP-PAMIEC"
Klucz – „data_send" -> dane:
- data (format *date*),
- identyfikatory syren na których usuwamy informację o pamięci alarmu oddzielone separatorem
„ <syr> "
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.8.2 Klient -> serwer

8.8.2.1 Na wszystkich syrenach

Adres odbioru:

IP/eth_api_inne/pamiec

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „STOP-PAMIEC"
Klucz – „data_send" -> dane:
- data (format *date*),
- string *ALL*,
Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> dane -> string utworzony z danych:
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

8.8.2.2 Na wybranych syrenach

Adres odbioru:

IP/eth_api_inne/pamiec

Klucz – „session_id" -> dane: session_id
Klucz – „date" -> dane: date
Klucz – „zadanie" -> dane: „STOP-PAMIEC"
Klucz – „data_send" -> dane:
- data (format *date*),
- identyfikatory syren na których usuwamy informację o pamięci alarmu oddzielone separatorem „ <syr> "

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).

Z powyższych danych tworzymy string, a parametry rozdzielamy między sobą znakiem interpunkcyjnym „ , "(przecinek).
Klucz – „PIN" -> *dane -> string utworzony z danych:*
klucz_licencyjny,session_id,date,zadanie,data_send,OB_pin
Klucz – „check" -> dane -> string utworzony z danych: *session_id,date,zadanie,data_send,PIN*

9 Bezpieczeństwo odbierania powiadomień URLC

Wygenerowany kod uwierzytelniający dla klienta jest podstawą do wyliczenia sumy kontrolnej powiadomień, którą klient dostaje w powiadomieniach URLC. Skrypt odbierający powiadomienia URLC powinien sprawdzić czy w parametrze nadesłanym z serwera jest taka sama wartość jak ta wyliczona przez skrypt klienta. System odbierający powiadomienia/żądania powinien sprawdzić IP serwera z którego otrzymuje dane. IP serwera jest w posiadania MUW. Należy również oprócz sumy kontrolnej porównać PIN przesłany z serwera.

10 Komunikacja RADIOWA

Powyższa specyfikacja ma zastosowanie również w komunikacji radiowej. Klucze należy odseparować od siebie znakiem interpunkcyjnym „ : "(dwukropek) z zachowaniem kolejności przedstawionej w powyższej specyfikacji i wysłać drogą radiową między centralami.

Każdy string rozpoczynał będzie się od:

1. Litera „R"
2. Numer komendy zgodnie z powyższym dokumentem wyłączając znaki interpunkcyjne „ ." (kropki) na 4 bitach *(przykład :uruchomienie alarmów z MUW na wszystkich syrenach-> numer komendy 8100*
   kasowanie pamięci alarmów na wszystkich syrenach-> numer komendy 8811)
3. Dane oddzielone separatorem „ : "
4. Znak końca linii.

Przykład:
R8100:5731a9b6a26d1:1462872502:ON-ALARM:A1N,9,1462872502,1,CALE:
4347d0f8ba661234a8eadc005e2e1d1b646c9682:294940e9201fa55762194ac8cf0c2c23