# Spis treści


**1. Definicje** 1

**1.1. Terminologia podstawowa** 1

**1.2. Terminologia techniczna** 1

**2. Ogólna koncepcja wymiany danych** 1

**3. Założenia dotyczące przyjętych technologii** 2

**4. Założenia w obszarze komunikacji** 3

**4.1. Bezpieczeństwo** 3

**4.2. Infrastruktura PKI** 4

**4.3. Identyfikacja Urządzenia/Aplikacji klienckiej** 5

**4.4. Protokół komunikacyjny** 5

**5. Sposób realizacji połączenia – przykład ogólny** 6

**5.1. KROK 1 - Rejestracja urządzenia/aplikacji klienckiej** 7

**5.2. KROK 2 – Pobranie listy dostępnych usług** 7

**5.3. KROK 3 – Przesłanie żądania do wybranej usługi** 7

# 1. Definicje

## 1.1. Terminologia podstawowa

- **Urządzenie/Aplikacja Kliencka**- w niniejszym dokumencie każde urządzenie lub aplikacja jaka łączy się z szyną danych (zarówno jeśli chodzi o aplikacje od dostawców zewnętrznych jak i wewnętrzne systemy ZDM)..
- **Szyna danych ZDM** – szyna ESB umożliwiająca integrację systemów dziedzinowych (w obrębie organizacji) ZDM oraz integrację tychże z dostawcami zewnętrznymi.
- **Usługa** – interfejs udostępniony przez szynę danych ZDM.
- **Centrala przetwarzania** – (również w krótszej formie jako „centrala”) system informatyczny przeznaczony do gromadzenia i przetwarzania danych przez urządzenia I aplikację z wykorzystaniem szyny danych ZDM jako interfejsu pośredniczącego w komunikacji. W praktyce są to poszczególne systemy dziedzinowe ZDM.

## 1.2. Terminologia techniczna

- **RSA** - Algorytm umożliwiający realizację kryptografii asymetrycznej (zgodnie z standardem PKCS#1).
- **JSON** - Tekstowy format danych. Opisany w standardzie RFC 4627.
- **OpenVPN** - Projekt realizacji wirtualnej sieci prywatnej bazujący w zakresie bezpieczeństwa na protokole TLS (tak zwany SSL VPN). OpenVPN to zarówno protokół komunikacji jak i realizujące go oprogramowanie na licencji GPL. Strona WWW: http://openvpn.net/.
- **OpenSSL** - Projekt implementacji algorytmów kryptograficznych. Strona WWW: http://www.openssl.org/.
- **Certyfikat** - certyfikat klucza publicznego wydawany przez ZDM na podstawie przekazanego przez stronę integrującą się pliku CSR.

# 2. Ogólna koncepcja wymiany danych

- Przyjmuje się, że Urządzenie/Aplikacja komunikuje się z centralą za pośrednictwem tunelu VPN.

- Poszczególne składniki infrastruktury i usług szyny danych, sposób autoryzacji między systemami ZDM, są całkowicie “przeźroczyste” dla Urządzenia/Aplikacji Klienckiej. **Autoryzacja jest możliwa dzięki przedstawieniu się (przy okazji każdego żądania do szyny danych) za pomocą certyfikatów dostarczonych przez ZDM**.
- Komunikacja wewnątrz tunelu realizowana jest w oparciu o protokół HTTP 1.1 w konwencji REST.
- Strony (Centrala i Urządzenie/Aplikacja) przekazują sobie komunikaty w formacie JSON (oraz binarnie - w przypadku plików). Specyfikacja komunikatów dla konkretnej usługi, udostępnionej przez szynę ZDM, jest przedmiotem osobnej dokumentacji.
- Szyna danych umożliwiają Urządzeniu/Aplikacji Klienckiej przesyłać komunikaty do centrali, a centrali wykonywać zlecenia zmian konfiguracyjnych Urządzenia/Aplikacji oraz kwerendy do Urządzenia/Aplikacji (zgodnie z założeniami specyfikacji danej usługi udostępnionej przez szynę).
- Centrala może również pobierać dane z Urządzenia/Aplikacji pobierając je metodą HTTP GET, a następnie ewentualnie zlecając ich usunięcie metodą HTTP DELETE (zgodnie z założeniami specyfikacji danej usługi udostępnionej przez szynę).

## 3. Założenia dotyczące przyjętych technologii

Jak wspomniano w rozdziale 2 - komunikacja Urządzeń/Aplikacji Klienckich z usługami ZDM udostępnionych przez szynę danych bazuje na popularnych protokołach i formatach danych, takich jak HTTP/1.1 oraz JSON.

W przypadku niektórych usług że komunikacja Urządzeń/Aplikacji Klienckich może odbywać się po sieciach bezprzewodowych o stosunkowo niewielkich przepustowościach, a ponadto rozliczanych proporcjonalnie do ilości przetransferowanych danych, kluczowe jest minimalizowanie rozmiaru transferowanych danych.

Dane są więc kompresowane tam gdzie ma to sens za pomocą algorytmu Deflate (RFC 1951) lub LZMA (patrz http://7- zip.org/7z.html). Dane w formatach realizujących kompresję (np. JPEG) nie powinny być ponownie kompresowane metodą Deflate lub LZMA.

W usługach gdzie niezbędne jest dostarczenie archiwum plików zastosowano 7z (patrz http://7- zip.org/7z.html). Kontener w tym formacie umożliwia stosowanie wielu typów kompresji, w tym LZMA.

# 4. Założenia w obszarze komunikacji

## 4.1. Bezpieczeństwo

- Cały ruch pomiędzy Urządzeniami/Aplikacjami klienckimi odbywa się w szyfrowanym tunelu VPN. Istnienie VPN gwarantuje integralność i poufność transmisji bez względu na wykorzystywane protokoły – protokoły nie muszą więc dostarczać niezależnie mechanizmów integralności i poufności transmisji.
- Klucz publiczny (odpowiedni dla klucza prywatnego) jest udostępniany przez Urządzenie/Aplikację kliencką i stanowi jego (równoległy do numeru seryjnego) unikatowy identyfikator.
- ZDM generuje certyfikat dla klucza każdego Urządzenia/Aplikacji klienckiej przeznaczonego autoryzacji dostępu do usług udostępnionych przez szynę danych.
- Nowo wprowadzane Urządzenia/Aplikacje klienckie muszą mieć inne klucze niż używane kiedykolwiek w przeszłości (klucze nie mogą się powtarzać).
- Niezależnie od kluczy służących podpisywania wybranych danych (w zależności od konkretnej usługi – zgodnie z specyfikacją tejże) Urządzenia/Aplikacje klienckie otrzymają (w postaci plików) klucze do nawiązywania tunelu VPN.

## 4.2. Infrastruktura PKI

System wykorzystuje infrastrukturę PKI na dwa sposoby:

- Ewentualnego uwierzytelnianie w tunelu VPN,
- Sygnatura cyfrowa pod ewentualnymi (zgodnie z specyfikacją określonej usługi) archiwami.

Do obu trybów wykorzystuje się rozdzielna pary kluczy RSA.

System (Urządzenia/Aplikacje Klienckie, szyna i centrala) stosuje klucze o długości 2048 bitów i funkcję skrótu SHA-2 (256 bitów). Sygnatury realizowane są zgodnie z PKCS#1 w wersji 1.5.

Na potrzeby uwierzytelniania w tunelu VPN centrala utrzymuje minimalną infrastrukturę niezbędną do wygenerowania certyfikatu CA oraz generuje klucze i certyfikaty dla poszczególnych Urządzeń.

Klucze i certyfikaty Urządzenia/Aplikacji Klienckich są przygotowane w postaci plików w formatach obsługiwanych przez OpenVPN.

W przypadku pary kluczy do składania i weryfikacji sygnatury pod przesyłanymi archiwami/plikami (jeśli wymaga tego specyfikacja usługi) obsługiwana jest w inny sposób:

- Urządzenie/Aplikacja kliencka posiada unikatowy klucz prywatny przechowywany w Urządzeniu/Aplikacji w sposób uniemożliwiający jego pozyskanie skopiowanie lub użycie
- Urządzenie/Aplikacja kliencka zwraca klucz publiczny jeśli otrzyma właściwe polecenie,
- Urządzenie/Aplikacja kliencka podpisuje każdy plik/archiwum kluczem prywatnym zgodnie z wymaganym (w dokumentacji danej usługi) formatem.

Klucze do podpisów są nierozerwalnie związane z Urządzeniem. Ten sam klucz na dwu Urządzeniach traktowany jest jako usterka Urządzeń uniemożliwiające ich wykorzystanie.

## 4.3. Identyfikacja Urządzenia/Aplikacji klienckiej

Każde Urządzenie/Aplikacja kliencka jest dodawane do katalogu prowadzonego przez oprogramowanie Eureka (tylko w przypadku usług, które do swojego działania wymagają dwustronnej komunikacji z Urządzeniem/Aplikacją kliencką).

Eureka to technologia opracowana przez Netflixa na potrzebę swojej infrastruktury serwerowej. Domyślnym zastosowaniem Eureki jest lokalizowanie usług i serwisów w rozproszonej architekturze oraz monitorowanie ich stanu.

Na potrzeby projektu technologia ta jest wykorzystana do monitoringu aktywnych w danym momencie maszyn, pozyskiwania ich adresów IP celem uzyskania dwukierunkowej komunikacji oraz kontrolowania ich stanu.

Komunikacja z serwerem przebiega po protokole **HTTPS** (konieczne przekazanie certyfikatu Klienta - Urządzenia/Aplikacja Kliencka) korzystając z architektury REST.

**Dokumentacja: https://github.com/Netflix/eureka/wiki**

Urządzenia będą rozpoznawane, podczas rejestracji w katalogu, dzięki dostarczonemu przez ZDM, certyfikatowi.

W trakcie działania Urządzenia/Aplikacji klienckiej są rozróżnione trzy typy żądań do Eureki:

- Żądanie zarejestrowania w rejestrze (przy starcie urządzenia)
- Periodyczny ping (zgodnie z zadanym interwałem czasowym)
- Wyrejestrowanie urządzenia

### 4.4. Protokół komunikacyjny

Protokół wymiany zrealizowany jest w oparciu o koncepcję REST. Oznacza to, że komunikaty pomiędzy Urządzeniem a centralą są przesyłane w oparciu o protokół http/1.1, z zastosowaniem odpowiednich do zadania metod (GET, POST, DELETE, PUT, HEAD).

Na potrzeby wymiany danych przyjmuje się następujące założenia (jeśli wymagają tego specyfikacje poszczególnych usług udostępnionych przez szynę danych ZDM):

- Połączenia może nawiązywać zarówno Urządzenie/Aplikację kliencką jak i centralę – obie strony wystawiają zatem usługi http na ustalonym porcie (w przypadku usług, które wymagają dwukierunkowej komunikacji).
- Co do oprogramowania serwera i klienta http zakłada się minimalną implementację:
    - Nie jest używana ani obsługiwana funkcja Expect: 100-countinue (RFC 2616, rozdział 14.20),
    - Wszystkie połączenia są zamykane (brak „keep-alive” – nagłówki „Connection: close”),
- Komunikaty JSON przesyłane są z „Content-type” równym „application/json”,
- Ewentualne archiwa przesyłane są z „Content-type” równym „application/x-7z-compressed”,
- Dla komunikatów JSON wspierana jest kompresja (Nagłówek Content-encoding: gzip),
- Serwer i klient (w celu oszczędzania transmisji) przesyłają tylko niezbędne nagłówki.
- Obsługiwane jest wznawianie transferu dla metody GET (RFC 2616, Rozdział: 14.3 Byte Ranges).

## 5. Sposób realizacji połączenia – przykład ogólny

Przed rozpoczęciem integracji z szyną danych ZDM konieczne jest:

- Pozyskanie specyfikacji dotyczącej wybranej usługi będącej przedmiotem integracji.
- Wygenerowanie pliku żądania wydania certyfikatu przez ZDM – CSR

- Dysponować otrzymanym od ZDM (lub zaakceptowany przez ZDM) certyfikatem, który pozwoli na poprawną identyfikację Urządzenia/Aplikacji Klienckiej na szynie danych.
- Skonfigurowanie i uruchomienie połączenia VPN

**UWAGI:**

- W przypadku konieczności realizacji komunikacji dwukierunkowej konieczne jest zrealizowanie kroków od 1 do 3
- W przypadku korzystania z usług, z których korzystać będzie Urządzenie/Aplikacja kliencka w sposób jednostronny, wystarczająca jest realizacja kroków 2 oraz 3.
- **Do żądania rejestracja (jak i do każdego innego) należy dołączyć certyfikat udostępniony (lub zaakceptowany) przez ZDM.**

## 5.1. KROK 1 - Rejestracja urządzenia/aplikacji klienckiej

**Adres:** `<szyna>/services/EurekaProxy/`

**Przeznaczenie:** zarejestrowanie Urządzenia/Aplikacji klienckiej w katalogu utrzymywanym z wykorzystaniem Eureki.

**UWAGI:**

- Krok ten jest konieczny do realizacji tylko w przypadku korzystania z usługi udostępnionej przez szynę danych, która wymaga dwukierunkowej komunikacji pomiędzy szyną (centralą) a urządzeniem/aplikacją kliencką (np.: scanner).

**Przykładowe żądanie** - *zgodne z specyfiką Eureki*:
https://github.com/Netflix/eureka/wiki/Eureka-REST-operations

```
curl -kv -X POST --data '{...}'
--header "Content-Type: application/json"
--key key.key.pem
--cert ./key.crt.pem
https://adres/services/EurekaProxy/apps/appID
```

**Odpowiedź na żądanie:** zgodnie z specyfikacją Eureki: https://github.com/Netflix/eureka/wiki

*Code: 204 on success*

## 5.2. KROK 2 – Pobranie listy dostępnych usług

**Adres:** `<szyna>/services/EurekaProxy/`

**Przeznaczenie:** pobieranie adresacji usług udostępnianych przez szynę danych ZDM (i aktywnej w danym momencie).

**Przykładowe żądanie** - zgodne z specyfiką Eureki:
https://github.com/Netflix/eureka/wiki/Eureka-REST-operations

```
curl -kv --header "Accept: application/json"
--key key.key.pem
--cert ./key.crt.pem
https://adres/services/EurekaProxy/apps
```

*Odpowiedź na żądanie:* zgodnie z specyfikacją Eureki: https://github.com/Netflix/eureka/wiki

HTTP Code: 200 on success

Output: JSON/XML

## 5.3. KROK 3 – Przesłanie żądania do wybranej usługi

**Adres:** zgodnie z zwróconą adresacją przy wywołaniu żądania z kroku 5.2

**Przeznaczenie:** przekazanie żądanie do wybranej usługi udostępnionej przez szynę danych ZDM (i aktywnej w danym momencie).

**Przykładowe żądanie** (puste)

```
curl -kv -X POST --data '{...}'
--header "Content-Type: application/json"
--key key.key.pem
--cert ./key.crt.pem
https://adres
```

**UWAGA:**

Wartość parametru cert, w wypadku kiedy certyfikat znajduje się w aktualnym katalogu należy poprzedzić nazwę pliku prefixem ./ W innym wypadku curl będzie próbował znaleźć ten certyfikat w systemowym „storze”.