Dokumentacja

Spis Treści

Rozdział 1. Definicje

1.1. Terminologia z zakresu dziedziny problemu

- **Urządzenie rejestrujące** - w niniejszym dokumencie każde Urządzenie wykorzystywane do wykrywania numerów tablic rejestracyjnych zaparkowanych pojazdów.
- **Dowód** – zbiór informacji dowodowych dokumentujących fakt zaistnienia naruszenia. Dowód jest przesyłany przez Urządzenia do centrali przetwarzania.
- **Sprawa** – informacja podstawowa identyfikująca fakt zaistnienia naruszenia.
- **Centrala przetwarzania** – (również w krótszej formie jako „centrala") system informatyczny przeznaczony do gromadzenia i przetwarzania spraw dostarczanych przez Urządzenia rejestrujące.

1.2. Terminy technologiczne

- **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/.

Rozdział 2. Ogólna koncepcja wymiany informacji

- Przyjmuje się, że Urządzenie komunikuje się z centralą za pośrednictwem tunelu VPN.
- Komunikacja wewnątrz tunelu realizowana jest w oparciu o protokół HTTP 1.1 w konwencji REST.
- Strony (centrala i Urządzenie) przekazują sobie komunikaty w formacie JSON. Specyfikacja komunikatów pozwala Urządzeniu przesyłać komunikaty do centrali a centrali wykonywać zlecenia zmian konfiguracyjnych Urządzenia oraz kwerendy do Urządzenia.
- Centrala może również pobierać dowody wykroczeń pobierając je metodą HTTP GET z Urządzenia, a następnie zlecając ich usunięcie metodą HTTP DELETE.
- Dowody naruszeń są udostępniane centrali w formacie archiwum 7-zip zawierające opis sprawy i materiały dowodowe – zdjęcia.
- W celu zapewnienia integralności dowodu (każdy plik, w szczególności archiwa 7-zip) są podpisane prywatnym (niejawnym) kluczem Urządzenia.

2.1. Założenia dotyczące roli Urządzeń rejestrujących

Zadaniem urządzenia rejestrującego jest przetwarzania danych dotyczących numerów tablic rejestracyjnych parkujących pojazdów.

2.2. Założenia dotyczące przyjętych technologii

Komunikacja bazuje na popularnych protokołach i formatach danych, takich jak HTTP/1.1 oraz JSON. W związku z tym, że komunikacja 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 aby nie obciążać niepotrzebnie procesora Urządzenia.
Tam gdzie potrzebne jest archiwum plików (patrz format dowodu) zastosowano 7z (patrz http://7-zip.org/7z.html). Kontener w tym formacie umożliwia stosowanie wielu typów kompresji, w tym LZMA.

Elementy rozwiązania wymagające kryptografii bazują na popularnych i ogólnie dostępnych algorytmach. Istnieją gotowe implementacje umożliwiające ich nieodpłatne użycie w komercyjnych rozwiązaniach.

Do zawiązania tunelu VPN stosowane jest oprogramowanie OpenVPN. Umożliwia ono realizację wirtualnej sieci pomiędzy Urządzeniami a centralą. Zaletą wybranego podejścia jest obecność pakietu

na wielu platformach (otwarte źródła i licencja GPL umożliwiają przenoszenie go na kolejne) oraz prawidłowa obsługa różnych trybów pracy sieci (np. NAT, firewall, zmienne adresy IP) a także stosunkowo mały narzut na rozmiar transmitowanych danych i ich opóźnienie.

Przyjęto, że do wykonania operacji wszystkich kryptograficznych wystarczający jest pakiet oprogramowania OpenSSL (OpenVPN bazuje w zakresie kryptografii na OpenSSL). Zaprezentowane w dokumencie przykłady składania czy weryfikacji podpisu posługują się wręcz uruchamianymi z linii poleceń komendami programu openssl. Biblioteka OpenSSL umożliwia delegowanie operacji kryptograficznych do Urządzeń sprzętowych (kart kryptograficznych lub HSM) z wykorzystaniem dedykowanych implementacji silników (engine) lub poprzez interfejs PKCS#11.

Unika się wykorzystania formatu danych XML ponieważ szereg możliwości tego języka jest w opisywanym zastosowaniu nadmiarowy a powoduje często konieczność stosowania złożonych i niepotrzebnie konsumujących zasoby interpreterów.

W trakcie pracy nad dokumentem przyjęto, że implementacja na większości platform sprzętowych i operacyjnych możliwa jest z wykorzystaniem następujących pakietów:

- OpenVPN (realizacja tunelu VPN),
- OpenSSL (realizacja usług kryptograficznych),
- LZMA SDK (obsługa algorytmów kompresji i archiwum 7z).

Nie istnieje wymaganie zastosowania żadnego z powyższych komponentów programistycznych, ale wykorzystanie trzech pierwszych pozycji jest rekomendowane.

2.3. Założenia dotyczące bezpieczeństwa

Cały ruch pomiędzy Urządzeniami rejestrującymi a centralą odbywa się w szyfrowanym tunelu VPN. Urządzenia rejestrujące nie udostępniają żadnych usług sieciowych poza siecią wykreowaną w ramach VPN. Oznacza to, że tylko centrala przetwarzania może nawiązywać połączenia do Urządzenia.

Istnienie VPN gwarantuje integralność i poufność transmisji bez względu na wykorzystywane protokoły – protokoły nie muszą więc dostarczać niezależnie mechanizmów integralności i poufności transmisji.

Każdy dowód jest kryptograficznie podpisany przez Urządzenie kluczem prywatnym przechowywanym w Urządzeniu.

Klucz publiczny (odpowiedni dla klucza prywatnego) jest udostępniany przez Urządzenie i stanowi jego (równoległy do numeru seryjnego) unikatowy identyfikator.

Centrala generuje certyfikat dla klucza każdego Urządzenia przeznaczonego do pracy. Certyfikat poza kluczem publicznym Urządzenia zawiera informacje takie jak numer seryjny Urządzenia.

Nowo wprowadzane Urządzenia muszą mieć inne klucze niż używane kiedykolwiek w przeszłości (klucze nie mogą się powtarzać).

Niezależnie od kluczy służących podpisywaniu dowodów naruszeń Urządzenia otrzymają (w postaci plików) klucze do nawiązywania tunelu VPN.

2.4. Założenia dotyczące dostępności

Protokół zakłada, że od żadnego elementu systemu (centrala, Urządzenie) nie wymaga się dostępności przekraczającej 99,9%. Oznacza to, że komponenty muszą poprawnie obsługiwać awarie drugiej strony czy też jej niedostępność.

2.5. Kontrola warunków pracy Urządzenia rejestrującego

2.5.1. Synchronizacja czasu

System działa w oparciu o czas UTC.
Urządzenie musi synchronizować czas z centralą w oparciu protokół NTP (RFC 1305). Centrala wskaże adresy co najmniej dwóch serwerów NTP. Urządzenie musi posiadać funkcje umożliwiające konfigurację serwerów NTP.

Synchronizacja czasu odbywa się w tunelu VPN.

2.5.2. Zgłoszenie stanu pracy

Każda sytuacja awaryjna lub wskazująca na nieprawidłowe lub podejrzane zachowanie dowolnego elementu Urządzenia rejestrującego jest zgłaszane do centrali.

Urządzenia posiadają listę parametrów pracy wraz z konfigurowalnymi progami poprawności pracy. Przekroczenie dozwolonych wartości skutkuje przesłaniem alerty do centrali.

2.5.3. Ustawienia parametrów pracy Urządzenia rejestrującego

Urządzenie udostępnia funkcjonalność umożliwiającą zdalną zmianę jego parametrów z centrali.

Urządzenie, w zależności od swojego typu, udostępnia właściwą listę parametrów.

2.5.4. Identyfikacja urządzenia

Każde urządzenie jest dodawane do katalogu prowadzonego przez oprogramowanie Eureka.
Urządzenia będą rozpoznawane podczas rejestracji w katalogu poprzez klucz SSL.

W trakcie działania Urządzenia są rozróżnione trzy typy żądań:

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

- Wyrejestrowanie urządzenia

Rozdział 3. Wykorzystanie infrastruktury PKI

System wykorzystuje infrastrukturę PKI na dwa sposoby:

- Uwierzytelnianie w tunelu VPN,
- Sygnatura cyfrowa pod dowodem naruszenia,

Do obu trybów wykorzystuje się rozdzielna pary kluczy RSA.

System (Urządzenia 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 są przygotowane w postaci plików w formatach obsługiwanych przez OpenVPN. Administrator instaluje je na Urządzeniu podczas jego konfiguracji.

Para kluczy do składania i weryfikacji sygnatury pod dowodem obsługiwana jest w inny sposób:

- Urządzenie posiada unikatowy klucz prywatny przechowywany w Urządzeniu w sposób uniemożliwiający jego pozyskanie skopiowanie lub użycie,
- Urządzenie zwraca klucz publiczny jeśli otrzyma właściwe polecenie,
- Urządzenie podpisuje każdy dowód kluczem prywatnym zgodnie z wymaganym formatem.

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

3.1. Sposób tworzenia sygnatury dowodu

Sygnatura tworzona jest na dowodzie jako skrót SHA-256 zaszyfrowany kluczem RSA 2048 bitów zgodnie z PKCS#1 w wersji 1.5.

Poprawne wykonanie sygnatury demonstruje poniższy przykład:

```
openssl dgst -sha256 -sign prywatny.key -out sygnatura manifest
```

Gdzie:

- openssl dgst – polecenie wykonania sygnatury pakietu openssl,
- -sha256 – instrukcja użycia dla skrótu algorytmu SHA-256,
- -sign prywatny.key – instrukcja złożenia podpisu kluczem zawartym lub wskazanym w pliku prywatny.key,

- -out sygnatura – umieszczenie sygnatury we wskazanym pliku,
- manifest – nazwa pliku, dla którego wykonana jest sygnatura.

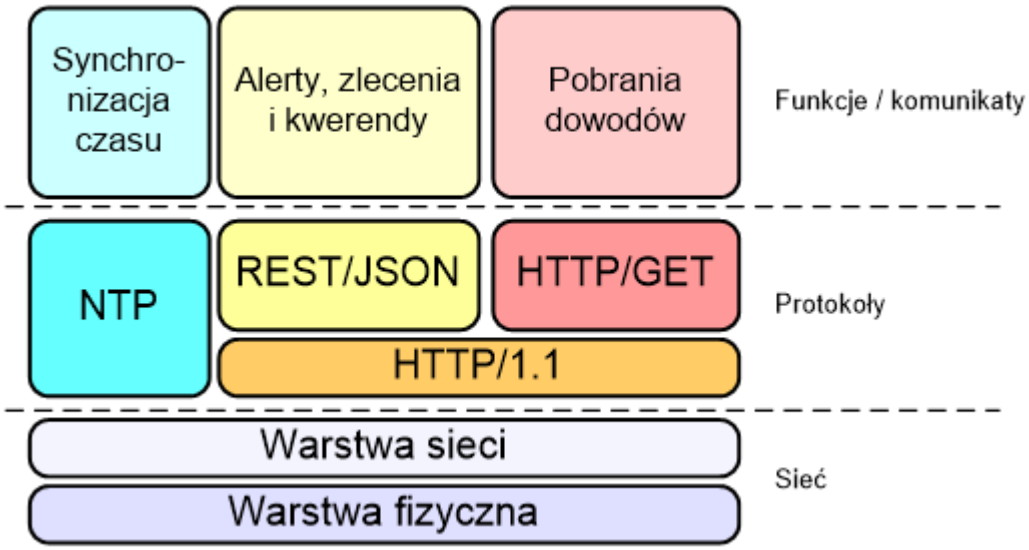
Centrala może zweryfikować poprawność tak wykonanej sygnatury poleceniem:

```
openssl dgst -verify publiczny.key -sha256 -signature sygnatura manifest
```

Rozdział 4. Opis protokołu

Protokół komunikacyjny pomiędzy Urządzeniem rejestrującym a centralą został zdefiniowany z wykorzystaniem opisu warstw w podziale na trzy grupy: sieć, protokoły i funkcje/komunikaty.

Struktura protokołów została zobrazowana w poniższym diagramie:



Kolejne rozdziały specyfikują poszczególne warstwy.

4.1. Warstwa sieciowa

Urządzenie rejestrujące posiada połączenie z centralą zrealizowane w dowolnej technologii przy czym wymaga się, aby:

- Połączenie zapewniało dostępność protokołu TCP/IP.
- Umożliwiało zainicjowanie wymiany pakietów TCP z Urządzenia do centrali,
- Umożliwiało przesyłanie pakietów TCP z centrali do Urządzenia jeśli nie minął określony czas od inicjalnego transferu Urządzenie-centrala lub kolejnych transferów.

Centrala przetwarzania wystawia jeden lub kilka adresów IP z którymi łączą się Urządzenia.

W celu zabezpieczenia poufności i integralności transmisji wykorzystywane będzie szyfrowany tunel VPN. Poza zagadnieniami związanymi z bezpieczeństwem zastosowanie zapewni właściwą adresację Urządzeń i ich dostępność z centrali na poziomie sieci.

Realizacja tunelu VPN oparta jest o udostępniane w postaci źródłowej oprogramowanie OpenVPN. Za przyjęciem takiego rozwiązania stoi przede wszystkim jego niezależność od systemu operacyjnego i platformy sprzętowej.

Konfiguracja Urządzenia rejestrującego w zakresie tunelu VPN musi umożliwiać łatwe zmiany

wykonywane przez administratora Urządzenia stąd treść plików konfiguracyjnych nie może podlegać homologacji czy zatwierdzeniu typu.

Minimalistyczny przykładowy plik konfiguracyjny tunelu OpenVPN wygląda następująco:

```
remote <adres IP punktu dostepowego centrali>
client
port 1194
cipher aes-128-cbc
dev tun0

ca certyfikat-ca.crt
cert certyfikat-Urzadzenia.crt
key klicz-prywatny-Urzadzenia.pem
```

Centrala dostarcza pliki certyfikatów i kluczy dla Urządzenia. Konfigurację wykonuje administrator. Wzmiankowany klucz służy do uwierzytelniania tunelu VPN i nie może być tym samym kluczem, który zostanie użyty do składania sygnatury pod danymi wchodzącymi w skład dowodu.

Klucz prywatny może być przechowywany jako plik.

Administrator może sterować całością konfiguracji OpenVPN – może zmienić każdy jej aspekt. Dokumentacja Urządzenia specyfikuje wymagane elementy konfiguracji (np. skrypty uruchamiane po nawiązaniu połączenia tunelu).

Interfejs lokalny Urządzenia umożliwia wgranie (upload) plików certyfikatów i kluczy do Urządzenia oraz ustawienie konfiguracji.

Centrala nadaje adresy IP wszystkim Urządzeniom w ramach tunelu. Wszystkie usługi centrali i cała komunikacja z centralą odbywa się na bazie adresacji wewnątrz tunelu IP.

IP nadawane Urządzeniem w ramach adresacji tunelu VPN są stałe -u. Przykładowa minimalistyczna konfiguracja OpenVPN po stronie centrali może wyglądać następująco:

```
proto udp
dev tun0
link-mtu 1500
port 1194

ca certyfikat-ca.crt
cert certyfikat-centrali.crt
key klucz-centrali.pem

dh dh1024.pem

server 10.128.0.0 255.255.0.0 ; przykladowa adresacja VPN
topology subnet

cipher AES-128-CBC
opt-verify cipher

ifconfig-pool-persist ipp.txt
keepalive 120 300
```

Centrala dostarcza serwer DNS dzięki czemu korzystanie z adresów IP (poza konfiguracją OpenVPN) nie jest wymagane.

Urządzenie musi dostarczać mechanizmy umożliwiające konfigurację co najmniej takich elementów jak adresy serwerów DNS i synchronizacji czasu.

4.2. Warstwa protokołu transportowego

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:

- Połączenia może nawiązywać zarówno Urządzenie jak i centrala – obie strony wystawiają zatem usługi http na ustalonym porcie,
- 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”),
    Jeśli będziemy zamykać połączenie po każdej komunikacji to wydłuży to komunikację po GPRS (za każdym razem konieczny jest handshake w SSL)
  - Komunikaty JSON przesyłane są z „Content-type” równym „application/json”,
  - Dowody przesyłane są z „Content-type” równym „application/x-7z-compressed”,

    - Dla komunikatów JSON wpierana 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).

4.3. Składnia zdalnych wywołań

Wywołania przyjmują dwie postaci:

- Alerty, zlecenie i kwerendy,
- Pobieranie dowodu.

Pierwszy typ jest wykorzystywany do zmiany parametrów Urządzenia, odpytywania Urządzenia lub zgłaszania przez Urządzenie awarii czy wygenerowania nowego dowodu.

Drugi typ jest metodą pobierania dowodu.

4.3.1. Alerty, zlecenia i kwerendy

Alerty to mechanizm powiadamiania centrali przez Urządzenie o zaistnieniu określonego faktu.

Zlecenia, to polecenia centrali przekazywane do Urządzenia w celu zmiany lub ustawienia typu pracy Urządzenia.

Kwerendy, to zapytania centrali kierowane do Urządzenia w celu pozyskania określonej informacji (np. zapytanie o aktualne ograniczenie prędkości ustawione na Urządzeniu.

Dwa pierwsze typy realizowane są metodą HTTP POST. Kwerendy realizowane są poprzez HTTP GET.

Wszystkie żądania posiadają następujące cechy:

- Treść żądań lub odpowiedzi przekazywana jest jako „application/json”.
- Treść może być kompresowana (Content-encoding oraz Accept-encoding: gzip),
- Przesyłane są wyłącznie wymagane nagłówki.

Rozdział 5. API

5.1. Eureka

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 HTTP(S) korzystając z architektury REST.

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

5.2. Modele

5.2.1. Case

Nazwa	Opis	Schemat
basis *opcjonalne*	Podstawa utworzenia sprawy.	enum (IncompleteRecognition, ParkedVehicle)
device *opcjonalne*	Urządzenie, z którego sprawa została zgłoszona.	Device
deviceState *opcjonalne*	Stan urządzenia w momencie zgłoszania sprawy.	DeviceState
id *opcjonalne*	Identyfikator sprawy nadany przez centralę, unikalny globalnie.	string
localId *opcjonalne*	Identyfikator sprawy nadany przez urządzenie zgłaszające, unikalny w obrębie urządzenia.	string
operator *opcjonalne*	Operator urządzenia, który zgłosił sprawę.	Operator
position *opcjonalne*	Współrzędne miejsca parkingowego.	Position
scans *opcjonalne*	Lista skanów.	< Scan > array

Nazwa	Opis	Schemat
time *opcjonalne*	Czas utworzenia sprawy.	string (date-time)
vehicle *opcjonalne*	Pojazd, którego sprawa dotyczy.	Vehicle

5.2.2. Scan

Nazwa	Opis	Schemat
contour *opcjonalne*	Obrys istotnego obszaru w pliku skanowania.	Rectangle
deviceState *opcjonalne*	Informacje o stanie urządzenia rejestrującego w momencie skanowania.	DeviceState
direction *opcjonalne*	Kierunek skanowania względem pojazdu rejestrującego.	enum (Left, Right, Both)
files *opcjonalne*	Pliki powiązane ze skanowaniem.	< File > array
localId *opcjonalne*	Lokalny identyfikator skanowania nadany przez urządzenie, unikalny w obrębie urządzenia.	string
position *opcjonalne*	Pozycja zeskanowanego pojazdu.	Position
positioningData *opcjonalne*	Zbiór dostępnych informacji w formacie NMEA 0183 (z GPS, dalmierza itp.).	< string > array
recognitions *opcjonalne*	Lista rozpoznanych informacji.	< Recognition > array
time *opcjonalne*	Czas zeskanowania pojazdu.	string (date-time)
triggerMode *opcjonalne*	Tryb wyzwolenia skanowania.	enum (Automatic, Manual)

5.2.3. File

Nazwa	Opis	Schemat
checksum *opcjonalne*	Suma kontrolna pliku.	string
filename *opcjonalne*	Nazwa pliku.	string

Nazwa	Opis	Schemat
size *opcjonalne*	Rozmiar pliku.	integer (int64)
type *opcjonalne*	Typ pliku.	enum (NumberPlate, Scan, MapOverview, Other)

5.2.4. Recognition

Struktura zawierająca informacje o rozponanej wartości.

Nazwa	Opis	Schemat
accuracy *opcjonalne*	Dokładność rozpoznania z zakresu <0, 1>.	number (double)
type *opcjonalne*	Typ rozpoznania.	enum (NumberPlate, CountryCode, VehicleType, VehicleBrand)
value *opcjonalne*	Rozpoznana wartość.	object

5.2.5. Operator

Nazwa	Opis	Schemat
firstName *opcjonalne*	Imię operatora.	string
identifier *opcjonalne*	Identyfikator operatora.	string
lastName *opcjonalne*	Nazwisko operatora.	string

5.2.6. Alert

Alert.

Nazwa	Opis	Schemat
code *opcjonalne*	Kod alertu.	string
description *opcjonalne*	Opis alertu.	string

Nazwa	Opis	Schemat
level *opcjonalne*	Powaga alertu.	enum (Info, Warning, Error)
operator *opcjonalne*	Operator urządzenia w trakcie alertu.	Operator
parameters *opcjonalne*	Parametry powiązane z wystąpieniem alertu.	< Parameter > array
subject *opcjonalne*	Temat alertu.	enum (Other, Buffer, Parameter, Security)
summary *opcjonalne*	Podsumowanie alertu.	string

5.2.7. Parameter

Nazwa	Opis	Schemat
description *opcjonalne*	Opis parametru.	string
format *opcjonalne*	Format wartości.	string
name *opcjonalne*	Nazwa parametru.	string
readOnly *opcjonalne*	Czy tylko do odczytu?	boolean
type *opcjonalne*	Typ wartości.	string
unit *opcjonalne*	Jednostka wartości.	string
value *opcjonalne*	Wartość parametru.	object

5.2.8. Rectangle

Nazwa	Opis	Schemat
bottomRight *opcjonalne*	Prawy dolny róg prostokąta.	Point
topLeft *opcjonalne*	Lewy górny róg prostokąta.	Point

5.2.9. Point

Nazwa	Opis	Schemat
x *opcjonalne*	Współrzędna na osi X.	number (double)
y *opcjonalne*	Współrzędna na osi Y.	number (double)

5.2.10. Vehicle

Nazwa	Opis	Schemat
brand *opcjonalne*	Marka pojazdu.	string
countryCode *opcjonalne*	Kod kraju.	string
numberPlate *opcjonalne*	Numer tablicy rejestracyjnej.	string
type *opcjonalne*	Typ pojazdu.	enum (Car, Truck, Motorcycle, Other)

5.2.11. Device

Nazwa	Opis	Schemat
manufacturer *opcjonalne*	Producent urządzenia.	string
model *opcjonalne*	Model urządzenia.	string
serialNumber *opcjonalne*	Numer seryjny urządzenia.	string
state *opcjonalne*	Stan urządzenia.	DeviceState

5.2.12. DeviceState

Nazwa	Opis	Schemat
position *opcjonalne*	Informacja o pozycji urządzenia.	Position

Nazwa	Opis	Schemat
positioningData *opcjonalne*	Zbiór dostępnych informacji w formacie NMEA 0183 (z GPS).	< string > array
scanDirection *opcjonalne*	Kierunek skanowania.	enum (Left, Right, Both)
status *opcjonalne*	Status urządzenia.	enum (Idle, Scanning)
time *opcjonalne*	Czas na urządzeniu.	string (date-time)

5.2.13. Position

Nazwa	Opis	Schemat
altitude *opcjonalne*	Wysokość bezwzględna.	number (double)
horizontalAccuracy *opcjonalne*	Dokładność wysokości i szerokości geograficznej.	number (double)
latitude *opcjonalne*	Wysokość geograficzna.	number (double)
longitude *opcjonalne*	Szerokość geograficzna.	number (double)
verticalAccuracy *opcjonalne*	Dokładność wysokości bezwzględnej.	number (double)

5.3. Parametry

Parametry, które są oznaczone jako kontekstowe nie są stałymi parametrami urządzenia. Służą one dostarczeniu informacji w kontekście jakiegoś zdarzenia np. alertu.

5.3.1. Parametr eureka-server

Nazwa	eureka-server
Typ	string
Tylko do odczytu	nie

Adres serwera Eureka, możliwe wiele wartości oddzielonych przecinkiem.

5.3.2. Parametr device-status

Nazwa	device-status
Tylko do odczytu	tak

5.3.3. Parametr ntp-enabled

Nazwa	ntp-enabled
Typ	boolean
Tylko do odczytu	nie

5.3.4. Parametr ntp-server

Nazwa	ntp-server
Typ	string
Tylko do odczytu	nie

Adres serwera NTP, możliwe wiele wartości oddzielonych przecinkiem.

5.3.5. Parametr current-time

Nazwa	current-time
Tylko do odczytu	nie
Format	yyyy-MM-dd'T'HH:mm:ss.fff

W przypadku chęci ustawienia tego parametru na zadaną wartość, najpierw parametr ntp-enabled powinien zostać ustawiony na `false`, tak aby ustawiona wartość nie została nadpisana przez wartość z serwera NTP.

5.3.6. Parametr affected-case-number

Nazwa	affected-case-number
Tylko do odczytu	tak
Kontekstowy	tak
Występowanie	Powiadomienie o wystąpieniu alertu

Parametr **affected-case-number** jest parametrem kontekstowym, tj. występuje tylko w kontekście alertów.

5.3.7. Parametr previous-device-status

Nazwa	previous-device-status
Tylko do odczytu	tak
Kontekstowy	tak
Występowanie	Powiadomienie o wystąpieniu alertu

5.4. Centrala

5.4.1. Zgłoszenie nowej sprawy do centrali

```
POST /case/
```

Parametry

Typ	Nazwa	Opis	Schemat
Body	**case** *wymagane*	Informacje o zgłaszanej sprawie.	Case

Odpowiedzi

Kod HTTP	Opis	Schemat
200	OK	Case

Konsumuje

- application/json;charset=UTF-8

Produkuje

- application/json;charset=UTF-8

Model żądania

Ścieżka	Typ	Wymagane	Opis
localId	String	true	Identyfikator sprawy nadany przez urządzenie zgłaszające, unikalny w obrębie urządzenia.
basis	String	true	Podstawa utworzenia sprawy.
time	String	true	Czas utworzenia sprawy.
deviceState	Object	true	Stan urządzenia w momencie zgłoszania sprawy.
deviceState.time	String	true	Czas na urządzeniu.
deviceState.scanDirection	String	true	Kierunek skanowania.
deviceState.status	String	true	Status urządzenia.
deviceState.position	Object	true	Informacja o pozycji urządzenia.
deviceState.position.longitude	Number	true	Szerokość geograficzna.
deviceState.position.latitude	Number	true	Wysokość geograficzna.
deviceState.position.altitude	Number	true	Wysokość bezwzględna.

Ścieżka	Typ	Wymagane	Opis
operator	Object	true	Operator urządzenia, który zgłosił sprawę.
operator.identifier	String	true	Identyfikator operatora.
operator.firstName	String	true	Imię operatora.
operator.lastName	String	true	Nazwisko operatora.
position	Object	true	Współrzędne miejsca parkingowego.
position.longitude	Number	true	Szerokość geograficzna.
position.latitude	Number	true	Wysokość geograficzna.
position.altitude	Number	true	Wysokość bezwzględna.
vehicle	Object	true	Pojazd, którego sprawa dotyczy.
vehicle.brand	String	true	Marka pojazdu.
vehicle.countryCode	String	true	Kod kraju.
vehicle.type	String	true	Typ pojazdu.
vehicle.numberPlate	String	true	Numer tablicy rejestracyjnej.
scans[]	Array	true	Lista skanów.
scans[].localId	String	true	Lokalny identyfikator skanowania nadany przez urządzenie, unikalny w obrębie urządzenia.
scans[].time	String	true	Czas zeskanowania pojazdu.
scans[].triggerMode	String	true	Tryb wyzwolenia skanowania.
scans[].direction	String	true	Kierunek skanowania względem pojazdu rejestrującego.
scans[].position	Object	true	Pozycja zeskanowanego pojazdu.
scans[].position.longitude	Number	true	Szerokość geograficzna.

Ścieżka	Typ	Wymagane	Opis
scans[].position.latitude	Number	true	Wysokość geograficzna.
scans[].position.altitude	Number	true	Wysokość bezwzględna.
scans[].contour	Object	true	Obrys istotnego obszaru w pliku skanowania.
scans[].contour.topLeft	Object	true	Lewy górny róg prostokąta.
scans[].contour.topLeft.x	Number	true	Współrzędna na osi X.
scans[].contour.topLeft.y	Number	true	Współrzędna na osi Y.
scans[].contour.bottomRight	Object	true	Prawy dolny róg prostokąta.
scans[].contour.bottomRight.x	Number	true	Współrzędna na osi X.
scans[].contour.bottomRight.y	Number	true	Współrzędna na osi Y.
scans[].deviceState	Object	true	Informacje o stanie urządzenia rejestrującego w momencie skanowania.
scans[].deviceState.time	String	true	Czas na urządzeniu.
scans[].deviceState.scanDirection	String	true	Kierunek skanowania.
scans[].deviceState.status	String	true	Status urządzenia.
scans[].deviceState.position	Object	true	Informacja o pozycji urządzenia.
scans[].deviceState.position.longitude	Number	true	Szerokość geograficzna.
scans[].deviceState.position.latitude	Number	true	Wysokość geograficzna.
scans[].deviceState.position.altitude	Number	true	Wysokość bezwzględna.
scans[].recognitions[]	Array	true	Lista rozpoznanych informacji.

Ścieżka	Typ	Wymagane	Opis
scans[].recognitions[].type	String	true	Typ rozpoznania.
scans[].recognitions[].value	String	true	Rozpoznana wartość.
scans[].recognitions[].accuracy	Number	true	Dokładność rozpoznania z zakresu <0, 1>.
scans[].files[]	Array	true	Pliki powiązane ze skanowaniem.
scans[].files[].filename	String	true	Nazwa pliku.
scans[].files[].type	String	true	Typ pliku.
scans[].files[].size	Number	true	Rozmiar pliku.
scans[].files[].checksum	String	true	Suma kontrolna pliku.

Model odpowiedzi

Ścieżka	Typ	Opis
id	String	Identyfikator sprawy nadany przez centralę, unikalny globalnie.

Przykład

Żądanie zgłoszenia sprawy do centrali:

```
POST /case/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
Content-Length: 3929

{
  "operator" : {
    "firstName" : "Jan",
    "lastName" : "Nowak",
    "identifier" : "X591LO"
  },
  "basis" : "ParkedVehicle",
  "time" : "2018-05-11T12:21:09",
  "vehicle" : {
    "brand" : "OPEL",
```

```
    "type" : "Car",
    "countryCode" : "PL",
    "numberPlate" : "JX 12345"
  },
  "localId" : "C-201805-00000032",
  "scans" : [ {
    "files" : [ {
      "checksum" : "CA29B70026AD9833D084146E9FB6F7049892536B07CE1E646D586D99861C814C",
      "filename" : "S-201805-00001007-1.jpg",
      "size" : 408055,
      "type" : "Scan"
    }, {
      "checksum" : "07A1A6E4F99DCAD48018B32594BAC368A46DC9E7CCC061AF2D955AE838418624",
      "filename" : "S-201805-00001007-2.jpg",
      "size" : 38740,
      "type" : "NumberPlate"
    }, {
      "checksum" : "729E3CC06CEA5E95C4CB3D5C341022CB749D2FF874644B03E84C2F375E3085B8",
      "filename" : "S-201805-00001007-3.jpg",
      "size" : 72853,
      "type" : "MapOverview"
    } ],
    "direction" : "Right",
    "recognitions" : [ {
      "accuracy" : 0.91,
      "value" : "JX 12345",
      "type" : "NumberPlate"
    }, {
      "accuracy" : 0.73,
      "value" : "PL",
      "type" : "CountryCode"
    }, {
      "accuracy" : 0.97,
      "value" : "Car",
      "type" : "VehicleType"
    }, {
      "accuracy" : 0.62,
      "value" : "OPEL",
      "type" : "VehicleBrand"
    } ],
    "deviceState" : {
      "time" : "2018-05-11T12:15:00",
      "position" : {
        "longitude" : 21.00098,
        "altitude" : 130.0,
        "latitude" : 52.23058
      },
      "status" : "Scanning",
```

```
      "scanDirection" : "Both"
    },
    "triggerMode" : "Automatic",
    "time" : "2018-05-11T12:15:00",
    "localId" : "S-201805-00001007",
    "position" : {
      "longitude" : 21.00098,
      "altitude" : 130.0,
      "latitude" : 52.23058
    },
    "contour" : {
      "bottomRight" : {
        "y" : 0.8084,
        "x" : 0.611
      },
      "topLeft" : {
        "y" : 0.741,
        "x" : 0.4328
      }
    }
  }, {
    "files" : [ {
      "checksum" : "6D3885B29CEC26885945847D5D6D2C625CC6C72E1A3F610D577B745872EDC97A",
      "filename" : "S-201805-00001027-1.jpg",
      "size" : 520808,
      "type" : "Scan"
    }, {
      "checksum" : "C806237CC1CFF2099AF100F294643B61B620FE3A7B040D6A49528A2D0B6666D8",
      "filename" : "S-201805-00001027-2.jpg",
      "size" : 90698,
      "type" : "NumberPlate"
    }, {
      "checksum" : "1C9281ED7A34D13C6E2B17C3D01DEDBA414E373DABA1E67A9F1D00C51D11A756",
      "filename" : "S-201805-00001027-3.jpg",
      "size" : 103683,
      "type" : "MapOverview"
    } ],
    "direction" : "Right",
    "recognitions" : [ {
      "accuracy" : 0.91,
      "value" : "JX 12345",
      "type" : "NumberPlate"
    }, {
      "accuracy" : 0.73,
      "value" : "PL",
      "type" : "CountryCode"
    }, {
      "accuracy" : 0.97,
```

```
      "value" : "Car",
      "type" : "VehicleType"
    }, {
      "accuracy" : 0.62,
      "value" : "OPEL",
      "type" : "VehicleBrand"
    } ],
    "deviceState" : {
      "time" : "2018-05-11T12:21:00",
      "position" : {
        "longitude" : 21.00098,
        "altitude" : 130.0,
        "latitude" : 52.23058
      },
      "status" : "Scanning",
      "scanDirection" : "Both"
    },
    "triggerMode" : "Automatic",
    "time" : "2018-05-11T12:21:00",
    "localId" : "S-201805-00001027",
    "position" : {
      "longitude" : 21.00098,
      "altitude" : 130.0,
      "latitude" : 52.23058
    },
    "contour" : {
      "bottomRight" : {
        "y" : 0.8084,
        "x" : 0.611
      },
      "topLeft" : {
        "y" : 0.741,
        "x" : 0.4328
      }
    }
  } ],
  "position" : {
    "longitude" : 21.00095,
    "altitude" : 130.0,
    "latitude" : 52.23063
  },
  "deviceState" : {
    "time" : "2018-05-11T12:21:09",
    "position" : {
      "longitude" : 21.00098,
      "altitude" : 130.0,
      "latitude" : 52.23058
    },
```

```
    "status" : "Scanning",
    "scanDirection" : "Both"
  }
}
```

Odpowiedź centrali:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 39

{
  "id" : "DEV101-C-201805-00000032"
}
```

5.4.2. Powiadomienie o wystąpieniu alertu

```
POST /alert/
```

Parametry

Typ	Nazwa	Opis	Schemat
Body	**alert** *opcjonalne*	Informacje o alercie.	Alert

Odpowiedzi

Kod HTTP	Opis	Schemat
202	Accepted	Bez zawartości

Konsumuje

- application/json
- application/json;charset=UTF-8

Produkuje

- */*

Model żądania

Ścieżka	Typ	Wymagane	Opis
code	String	true	Kod alertu.
subject	String	true	Temat alertu.
level	String	true	Powaga alertu.
summary	String	true	Podsumowanie alertu.
parameters[]	Array	false	Parametry związane z alertem.
parameters[].name	String	true	Nazwa parametru.
parameters[].value	Number	true	Wartość parametru.
parameters[].type	String	true	Typ wartości parametru.
parameters[].unit	String	true	Jednostka wartości parametru.
parameters[].readOnly	Boolean	true	Czy parametr tylko do odczytu.
operator	Object	false	Operator urządzenia w momencie alertu.
operator.identifier	String	true	Identyfikator operatora.

Przykład - alert naruszenia integralności urządzenia

Żądanie zgłoszenia wystąpienia alertu:

```
POST /alert/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
Content-Length: 167

{
  "subject" : "Security",
  "code" : "integrity-breach",
  "operator" : {
    "identifier" : "X591LO"
  },
  "summary" : "Otwarto obudowe.",
  "level" : "Warning"
}
```

Odpowiedź centrali:

```
HTTP/1.1 202 Accepted
```

Przykład - alert zapełnienia bufora

Żądanie zgłoszenia wystąpienia alertu:

```
POST /alert/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
Content-Length: 318

{
  "subject" : "Buffer",
  "parameters" : [ {
    "readOnly" : true,
    "name" : "affected-case-number",
    "unit" : "N/A",
    "type" : "string",
    "value" : "202170/03/2018"
  } ],
  "code" : "buffer-full",
  "operator" : {
    "identifier" : "X591LO"
  },
  "summary" : "Bufor zapelniony.",
  "level" : "Error"
}
```

Parametr **affected-case-number** zawiera lokalny numer sprawy, przy której nie udało się zapisać dowodów z powodu zapełnienia bufora. Parametr ten jest opcjonalny, alert może zostać wysłany nie tylko w momencie pojawienia się nowego dowodu, którego nie udało się zapisać.

Odpowiedź centrali:

```
HTTP/1.1 202 Accepted
```

Przykład - alert o zmianie statusu urządzenia

Żądanie zgłoszenia zmiany statusu urządzenia:

```
POST /alert/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
Content-Length: 454

{
  "subject" : "Parameter",
  "parameters" : [ {
    "readOnly" : true,
    "name" : "device-status",
    "unit" : "N/A",
    "type" : "string",
    "value" : "Scanning"
  }, {
    "readOnly" : true,
    "name" : "previous-device-status",
    "unit" : "N/A",
    "type" : "string",
    "value" : "Idle"
  } ],
  "code" : "status-change",
  "operator" : {
    "identifier" : "X591L0"
  },
  "summary" : "Zmiana statusu urzadzenia.",
  "level" : "Info"
}
```

Odpowiedź centrali:

```
HTTP/1.1 202 Accepted
```

5.5. Urządzenie rejestrujące

5.5.1. Pobranie informacji o urządzeniu

```
GET /device/
```

Odpowiedzi

Kod HTTP	Opis	Schemat
200	OK	Device

Produkuje

- application/json;charset=UTF-8

Model odpowiedzi

Ścieżka	Typ	Opis
manufacturer	String	Producent urządzenia.
model	String	Model urządzenia.
serialNumber	String	Numer seryjny urządzenia.

Przykład

Żądanie pobrania informacji o urządzeniu:

```
GET /device/ HTTP/1.1
Host: localhost:8080
```

Odpowiedź urządzenia:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 82

{
  "manufacturer" : "ZDM",
  "model" : "UR101",
  "serialNumber" : "12345"
}
```

5.5.2. Pobranie informacji o sieci, do której urządzenie jest podłączone

```
GET /device/network
```

Odpowiedzi

Kod HTTP	Opis	Schemat
200	OK	NetworkInfo

Produkuje

- application/json;charset=UTF-8

Model odpowiedzi

Ścieżka	Typ	Opis
transportType	enum (Ethernet, Wifi, Cellular, Other)	Typ transportu w sieci.

Przykład

Żądanie pobrania informacji o sieci do której połączone jest urządzenie:

```
GET /device/network HTTP/1.1
Host: localhost:8080
```

Odpowiedź urządzenia:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 34

{
  "transportType" : "Cellular"
}
```

5.5.3. Pobranie listy parametrów

```
GET /device/parameter/
```

Parametry

Typ	Nazwa	Schemat
Query	**parameterNames** *opcjonalne*	< string > array(multi)

Odpowiedzi

Kod HTTP	Opis	Schemat
200	OK	< Parameter > array

Produkuje

- application/json;charset=UTF-8

Przykład - wylistowanie kilku zadanych parametrów

Żądanie wylistowania parametrów:

```
GET /device/parameter/?parameterNames=temp&parameterNames=ntp-enabled HTTP/1.1
Host: localhost:8080
```

Odpowiedź urządzenia:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 194

[ {
  "readOnly" : true,
  "name" : "temp",
  "unit" : "C",
  "type" : "numeric",
  "value" : 57.2
}, {
  "readOnly" : false,
  "name" : "ntp-enabled",
  "type" : "boolean",
  "value" : true
} ]
```

5.5.4. Ustawienie wartości parametru

```
PUT /device/parameter/{parameterName}
```

Parametry

Typ	Nazwa	Opis	Schemat
Path	**parameterName** *wymagane*	Nazwa parametru	string
Body	**value** *wymagane*	value	object

Odpowiedzi

Kod HTTP	Opis	Schemat
202	Accepted	Bez zawartości
404	Zadany parametr nie istnieje.	Bez zawartości
405	Zadany parametr jest tylko do odczytu.	Bez zawartości

Konsumuje

- `application/json;charset=UTF-8`

Produkuje

- `*/*`

Przykład - ustawienie temperatury maksymalnej

Żądanie ustawienia temperatury maksymalnej:

```
PUT /device/parameter/temp-max HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
Content-Length: 2

80
```

Odpowiedź urządzenia:

```
HTTP/1.1 202 Accepted
```

5.5.5. Pobranie skanowania

```
GET /device/scan/{localId}
```

Parametry

Typ	Nazwa	Opis	Schemat
Path	**localId** *opcjonalne*	Lokalny identyfikator sprawy	string

Odpowiedzi

Kod HTTP	Opis	Schemat
204	No Content	Scan

Produkuje

- */*

Model odpowiedzi

Ścieżka	Typ	Opis
localId	String	Lokalny identyfikator skanowania nadany przez urządzenie, unikalny w obrębie urządzenia.
time	String	Czas zeskanowania pojazdu.
files[]	Array	Pliki powiązane ze skanowaniem.
files[].filename	String	Nazwa pliku.
files[].type	String	Typ pliku.
files[].size	Number	Rozmiar pliku.
files[].checksum	String	Suma kontrolna pliku.

Przykład

Żądanie pobrania skanowania:

```
GET /device/scan/S-201805-00001007 HTTP/1.1
Host: localhost:8080
```

Odpowiedź urządzenia:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Content-Length: 634

{
  "files" : [ {
    "checksum" : "691D0B52C46CE767A8801D99D09D2F600350920A5DB4A60853E0E3331855EEC2",
    "filename" : "S-201805-00001007-1.jpg",
    "size" : 412091,
    "type" : "Scan"
  }, {
    "checksum" : "3EDEEE4F21418ECCB7109C5DACF6F72B6BCFBA65D1F8CEA178B4BDFA8331A400",
    "filename" : "S-201805-00001007-2.jpg",
    "size" : 50881,
    "type" : "NumberPlate"
  }, {
    "checksum" : "AF19DB5237A7BF9F2C070C390AABAFB692010B7D2A8BC43C9C340DD3D1F8EFAD",
    "filename" : "S-201805-00001007-3.jpg",
    "size" : 70912,
    "type" : "MapOverview"
  } ],
  "time" : "2018-07-26T11:36:35.735",
  "localId" : "S-201805-00001007"
}
```

5.5.6. Pobranie archiwum plików skanowania

```
GET /device/scan/{localId}/files
```

Odpowiedzi

Kod HTTP	Opis	Schemat
200	OK	Scan

Produkuje

- application/x-7z-compressed

Przykład

Żądanie pobrania archiwum plików dla konkretnego skanowania:

```
GET /device/scan/S-201805-00001007/files HTTP/1.1
Host: localhost:8080
```

5.5.7. Pobranie archiwum plików

```
GET /device/file/
```

Parametry

Typ	Nazwa	Opis	Schemat
Body	**filter** *opcjonalne*	Filtr plików.	FileFilter

Odpowiedzi

Kod HTTP	Opis	Schemat
200	OK	Bez zawartości

Konsumuje

- `application/json;charset=UTF-8`

Produkuje

- `application/x-7z-compressed`

Przykład

Żądanie pobrania archiwum plików dla dwóch konkretnych skanowań oraz tylko plików zawierających całe zdjęcie skanowania i tablicy rejestracyjnej:

```
GET /device/file/ HTTP/1.1
Content-Type: application/json;charset=UTF-8
Host: localhost:8080
Content-Length: 101

{
  "scanIds" : [ "S-201805-00001007", "S-201805-00001027" ],
  "types" : [ "Scan", "NumberPlate" ]
}
```

5.6. Pobieranie plików z urządzenia

W każdym przypadku kiedy od urządzenia żąda się zwrócenia jednego bądź więcej plików, powinny one zostać zebrane w archwium 7-zip.

Wymagania dotyczące zwracania plików:

- Każdy plik wygenerowany przez urządzenie musi posiadać unikalną nazwę w obrębie urządzenia.
- Archiwum plików musi zostać podpisane cyfrowo
- Odpowiedź zwracająca archiwum plików musi zawierać nagłówek Content-Signature, pozwalający na zweryfikowanie podpisu

Rozdział 6. Zasady weryfikacji zgodności ze standardem

Na potrzeby weryfikacji poprawności zgodności Urządzenia ze standardem przygotowane zostanie testowa wersja oprogramowania symulującego pracę centrali.

Test Urządzenia obejmuje wykonanie szeregu przebiegów na podstawie wszystkich scenariuszy opisanych w niniejszej dokumentacji rozszerzonych o sytuacje awaryjne – np. niedostępność, przerwy w komunikacji lub restart w trakcie przetwarzania.

Urządzenie działające poprawnie z wersją testową system centrali zostanie zaklasyfikowane jako poprawne.