

Technický předpis funkce sběru telemetrických dat a jejich předávání pomocí rozhraní TCP/IP Socket a REST - prostřednictvím Veřejného rozhraní ŘSD pro příjem GPS dat

Verze 1.1.1.1

Ze dne 25. 10. 2023

Obsah

1	Úvod	3
1.1	Názvosloví.....	3
1.2	Účel dokumentu	4
2	Architektura systému	5
2.1	Konceptuální diagram	5
2.2	Komponenty systému.....	5
2.2.1	GPS jednotka.....	5
2.2.2	Sběr dat na vozidle	6
2.2.2.1	Sledované parametry	6
2.2.2.2	Data specificky podle vozidel	7
2.2.2.3	Průběh sběru dat	9
2.2.3	Předávání dat do systému ŘSD	9
2.2.3.1	Frekvence	9
2.2.3.2	Mechanismus	10
2.2.3.3	Obsah předávaných dat	10
2.3	Přehled součástí serveru ŘSD obsluhujícího rozhraní pro příjem telemetrických dat	10
2.4	Protokoly a rozhraní	11
2.4.1	TCP/IP – Rozhraní S.....	11
2.4.1.1	Komunikační diagram	12
2.4.1.2	Komunikace na socketu - zásady	13
2.4.1.3	Technická omezení a doporučení	13
2.4.1.4	Zabezpečení	13

2.4.1.5	Chybové stavy a očekávaná reakce na straně klientské služby	13
2.4.1.6	Ukázkový kód (.NET Core – C#)	14
2.4.1.7	Testová metoda pro Unit testy	16
2.4.2	REST API – Rozhraní R.....	17
2.4.2.1	Definice REST API	17
2.4.2.2	Metody rozhraní	18
2.4.2.3	Zabezpečení	19
2.4.2.4	Chybové stavy a reakce na chyby	19
2.5	Popis dat a formát	19
2.6	Evidence užití konkrétních vozidel a nástaveb v rámci činností.....	20

1 ÚVOD

Tento předpis stanovuje požadavky na provedení a kvalitu GPS jednotek a telemetrických dat vozidel provádějící údržbu komunikací ve správě Ředitelství silnic a dálnic s. p. (dále jen ŘSD) a to jak vozidel ŘSD, tak vozidel dodavatelů provádějících údržbu na základě uzavřených rámcových dohod.

Dodavatel bude prováděné činnosti údržby komunikací, evidovat v software webové aplikace „Provozní deník“, kterou Objednatel Dodavateli zpřístupní a umožní vyškolení uživatelů vítězného Dodavatele k jejímu užívání.

Zadavatel si vyhrazuje právo na změnu protokolu pro předávání dat i datového formátu a obsahu.

Součástí komunikačního protokolu jsou přílohy – aktuálně platná dokumentace ke GPS ke stažení níže v aktuálně platném znění [https://podporagps.rsd.cz/ke-stazeni/Protokol/\(Verze\)](https://podporagps.rsd.cz/ke-stazeni/Protokol/(Verze)) a [https://podporagps.rsd.cz/ke-stazeni/Datovy_format/\(Verze\)](https://podporagps.rsd.cz/ke-stazeni/Datovy_format/(Verze))

Kde (Verze) označuje číslo verze Protokolu () resp. Datového formátu ()

Pro nejnovější platnou verzi se číslo nahrazuje slovem „Aktualní“

Tedy aktuálně nejnovější verze Protokolu je k dispozici pod odkazem

<https://podporagps.rsd.cz/ke-stazeni/Protokol/Aktualni>

a nejnovější platná verze datového formátu je k dispozici pod odkazem

https://podporagps.rsd.cz/ke-stazeni/Datovy_format/Aktualni

1.1 Názvosloví

Jednotka GPS – je zjednodušený název pro technické zařízení umístěné ve vozidlech, které zajišťuje sběr a předávání dat o poloze, automaticky generovaných dat o prováděných činnostech, data z CAN sběrnice vozidel, vozidlových nástaveb a dat ze čteček RFID, které jsou k ní připojeny.

GPS – pro potřeby tohoto dokumentu obecně jakýkoliv globální družicový polohový systém

Vozidla – tímto pojmem jsou myšlena všechna vozidla a stroje sloužící pro údržbu komunikací popsána v tomto dokumentu.

Vozíky – přívěsné vozidlo nesoucí dopravní zařízení nebo zařízení předběžné výstrahy podle typu používaný jako výstražný vozík nebo předzvěstný vozík.

Komunikační server – server na straně provozovatele GPS jednotek, který sbírá data poskytovaná GPS jednotkami vozidel, podle níže uvedeného funkčního popisu a datového formátu a následně je předává do ISUD.

Informační systém údržby dálnice / a silnic (ISUD/ISUDaS) – informační systém sledování a kontroly údržby komunikací ve správě ŘSD.

Dodavatelé údržby – dodavatelé ŘSD provádějící činnosti údržby.

Rozhraní S – rozhraní pro předávání telemetrických dat prostřednictvím TCP/IP Socketu

Rozhraní R – rozhraní pro předávání telemetrických dat prostřednictvím HTTP / REST API

1.2 Účel dokumentu

Předpis upravuje technické provedení mechanismu předávání telemetrických dat na veřejná rozhraní ŘSD – rozhraní S a R. Definuje komunikační postupy a omezení obou rozhraní, která musí být dodržena při implementaci klientských služeb na straně poskytovatele telemetrických dat při jejich návrhu a provozu. Předpis stanoví závazné postupy, jejichž dodržení je podmínkou pro převzetí plnění.

Změny oproti předchozí verzi

Změny verze 1.1.1. oproti verzi datové sady definované v dokumentu **KOMUNIKAČNÍ PROTOKOL 1.0**

- oddělena dokumentace formátu datové sady od komunikačního protokolu
- doplněna kompletní definice REST rozhraní R (Swagger) a popis jeho použití
- doplněn přehled součástí serveru ŘSD obsluhujícího rozhraní pro příjem telemetrických dat
- přidán blok **Chybové stavy a očekávaná reakce na straně klientské služby** pro rozhraní S
- přidán **Ukázkový kód (.NET Core – C#)** a **Testová metoda pro unit test** pro rozhraní S
- zavedeno elektronické umístění dokumentů <https://podporagps.rsd.cz/ke-stazeni/>
- odebrána specifikace povinnosti pro C-ITS

2 ARCHITEKTURA SYSTÉMU

2.1 Konceptuální diagram

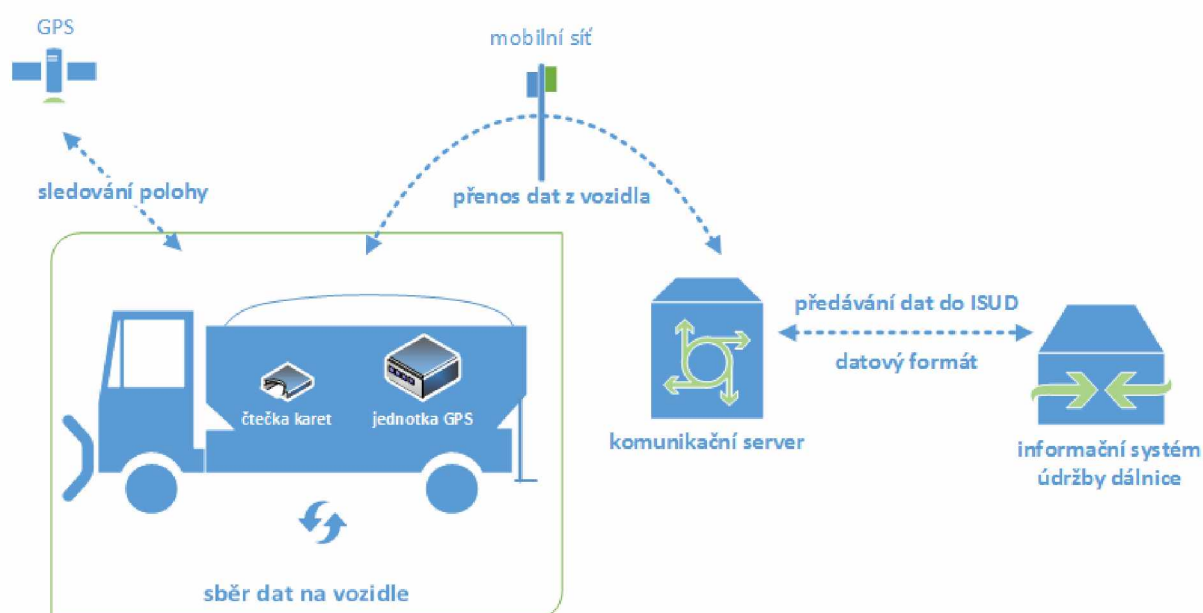


Diagram schematicky popisuje proces sběru, přenosu a předávání dat, který je určen tímto předpisem. Data jsou sbírána na úrovni vozidla pomocí jednotky GPS, která sleduje polohu pomocí satelitního systému GPS, snímá telemetrická data z vozidla, popř. vozidlové nastavby a zpracovává tyto informace dále doplněné o data ze čtečky karet. Data jsou následně pomocí mobilní sítě přenášena na komunikační server, kde jsou převedena do jednotného formátu (viz dokument **Technický předpis datového formátu telemetrických údajů**) a konečně předána ke zpracování a uložení do ISUD / ISUDaS.

2.2 Komponenty systému

Přehled požadovaných součástí řešení na straně poskytovatele údržby a poskytovatele telemetrických dat.

Tato část definuje požadavky jednotky určené do vozidel ŘSD. Pro dodavatele údržby jsou klíčové funkční požadavky popsány v dalších kapitolách (sběr, přenos a formát), povinné údaje a závazný datový formát je pak přesně stanoven v dokumentu **Technický předpis datového formátu telemetrických údajů**, nicméně parametry HW mohou využít jako doporučení pro správné funkce HW.

2.2.1 GPS jednotka

GPS jednotky musí splňovat tyto parametry:

- napájení universální v rozsahu 12/24 V, tj. vhodné do všech typů vozidel bez nutnosti použití převodníků napětí,
- teplotní rozsah od -25°C + 60°C,
- podpora připojení CAN sběrnice (FMS standard),

- GPS přijímač s vysokou citlivostí (doporučena podpora 2 sítí globálního družicového polohového systému),
- modem pro on-line přenos dat (GPRS nebo novější technologie),
- integrované akcelerační/decelerační čidlo,
- vnitřní paměť pro záznamy o kapacitě minimálně 40.000 záznamů,
- záložní napětí v případě výpadku napájení (minimálně 15 minut),
- možnost ukládat do záznamů servisní informace:
 - palubní napájení,
 - počet satelitů,
 - kvalita GSM signálu.
- jednotka musí být vybavena dostatečným počtem příslušných vstupů, aby bylo možné sledovat níže uvedené parametry z vozidla,
- nedostupnost GSM sítě - v případě výpadku nebo nedostupnosti mobilní sítě musí být data ukládána v jednotce GPS a po připojení do domovské sítě okamžitě odeslána,
- GPS jednotka musí odesílat uložená data od nejstarších záznamů po nejnovější.

2.2.2 *Sběr dat na vozidle*

2.2.2.1 *Sledované parametry*

Hodnoty sledované jednotkou GPS nebo získávané z jiných systémů ve vozidle a sbírané jednotkou GPS pro zajištění přenosu. Všechna vozidla budou poskytovat povinně sledované hodnoty. Další parametry jsou závislé zejména na technické vyspělosti vozidla a jeho schopnosti předávat tyto data jednotce GPS. Ostatní parametry se liší v závislosti na typu vozidla, resp. jeho nástavby. Níže je pro přehlednost uveden základní výpis sledovaných dat, které jsou následně přesně specifikovány v samostatném dokumentu **Technický předpis datového formátu telemetrických údajů** v aktuální verzi.

Povinně sledované u všech vozidel a strojů

- Datum, čas – vzniku záznamu,
- Kvalita signálu GSM,
- Počet satelitů,
- Jednoznačný identifikátor jednotky,
- Registrační značka vozidla
- Druh vozidla (osobní, dodávkové, nákladní, traktor/stroj, vozík, osoba),
- ID řidiče/jméno řidiče (NE pro dodavatele),
- Číslo smlouvy (NE pro ŘSD, ANO pro dodavatele)
- Identifikátor vozidla,
- Nesená nástavba (sypač, sekačka, samosběr, kropice, valník, nosič kontejnerů, ostatní)
- Zapnuté zapalování (klíček),
- Zeměpisná poloha,
- Aktuální rychlost z GPS,

- Aktuální rychlost z tachometru z GPS,
- Aktuální rychlost z CAN sběrnice,
- Aktuální stav tachometru z GPS,
- Aktuální stav tachometru z tachometru,
- Aktuální stav tachometru z CAN sběrnice,
- Režim jízdy (zimní údržba, letní údržba, kontrolní jízda, inspekční jízda, jízda BESIP, služební jízda, DIO),
- Otáčky motoru, pouze u nákladních vozidel, strojů, popř. pokud dodávkové vozidlo umožňuje,
- Spotřeba PHM od předcházejícího záznamu (pro dodávkové, nákladní vozidla, traktor/stroj) (NE pro dodavatele),
- Palubní napětí (NE pro dodavatele),
- Sledování zapnutí majáku (pokud je jím vozidlo vybaveno).

2.2.2.2 Data specificky podle vozidel

Jedná se o úplný výčet vozidel, na kterých může být v rámci poskytování služeb pro ŘSD požadováno umístění GPS a předávání dat GPS. Konkrétní povinnost vyplývá ze specifikace činnosti v konkrétní smlouvě a proto výčet povinných vozidel a mechanizací je uveden v podrobné specifikaci služeb.

- **Sypač**
 - režim posypu (nesype, chemický posyp, chemický posyp se zkrápěním, inertní posyp, inertní posyp se zkrápěním, zkrápění)
 - stav plužení,
 - gramáž posypu,
 - aktuální nastavená šíře posypu,
 - spotřeba materiálu (chemického, inertního, solanky),
- **Sekačka**
 - činností cepáku hlavní kosa,
 - činností cepáku druhé kosa,
 - činností cepáku třetí kosa,
- **Samosběr – s rozdělením**
 - válcové koště,
 - levé boční koště,
 - pravé boční koště,
 - turbína/sání,
 - spuštěná šachta,
- **Kropicí vůz**
 - levý splach,
 - pravý splach,
 - střední splach,

- mlžení (ozónu),
- čerpadla, (popř. čištění propustků, čištění vpustí)
- **Vozík (ŘSD)***
 - * **pro dodavatele povinná pouze poloha GPS, ostatní údaje nepovinné**
 - výstražná světla/šipka zapnuto,
 - režim zapnuté šipky (doleva, doprava, dolů)
 - rampa nahoře,
 - napětí akumulátoru
- **Další typy vozidel/nástaveb**

Vždy se sleduje činnost nástavby popř. stroje provádějící činnost, pro kterou je určena v rozsahu pracuje/nepracuje. Typy nástaveb popř. strojů:

- univerzální nosič, nástavba (pokud není specifikován v jiných činnostech) (bude popsání v deníku):
 - mytí značek
 - mytí směrových sloupků
 - mytí nástavců na svodidla
 - mytí baliset
 - mytí svodidel
 - čištění propustků
 - čištění vpustí
 - tlaková voda
 - čištění
 - seřezávání krajnic
 - hloubení příkopů
 - oprava silničních svahů
- vozidlo provádějící inspekční jízdu
 - práce vozidla
- jeřáb
 - činnost nástavby
- plošina
 - činnost nástavby
- nakladač
 - práce vozidla (otáčky motoru větší než 0)
- samopojízdný značkovací stroj
 - práce vozidla
- samojízdný stroj pro nedestruktivní odstraňování VDZ
 - práce vozidla
- samojízdný stroj pro nedestruktivní obnovu PVV

- práce vozidla
- válec
 - práce vozidla
- finišer
 - práce vozidla
- distributor
 - práce vozidla
- fréza
 - práce vozidla
- pracovní vozidlo (např. nákladní vozidlo odvázející odpad nebo vytěžený materiál na skládku nebo deponii) – dle definice v konkrétní smlouvě (neplatí pro vozidlo přivázející pracovníky)
 - práce vozidla
- speciální sací čistící vozidlo
 - práce vozidla, vč. odvozu odpadu na skládku

2.2.2.3 Průběh sběru dat

Jednotka musí být schopna zaznamenávat data na základě těchto parametrů:

- Po čase - nastavení max. 10 vteřin při jízdě,
- Po ujeté vzdálenosti - (minimální nastavitelný interval 10 m),
- Po změně azimutu - doporučené nastavení 10°.

Specifická je situace vozíků, a proto je třeba specifické nastavení:

- Je v provozu (zapnutá jakákoliv výstraha)
 - Po čase - nastavení max. 60 vteřin,
 - Po ujeté vzdálenosti - nastavení 200 m,
 - Po změně azimutu - doporučené nastavení 10°.
- Není v provozu (klidový režim)
 - Po ujeté vzdálenosti - nastavení 200 m,
 - Po změně azimutu doporučené nastavení 10°.

Pro sběr dat musí být splněn alespoň jeden z uvedených parametrů.

2.2.3 Předávání dat do systému ŘSD

2.2.3.1 Frekvence

Předávání dat do systému ŘSD musí být realizováno okamžitě s maximálním zpožděním 60 sekund od vzniku dat (platí při dostupnosti signálu GSM, jinak v co nejkratším čase po získání signálu).

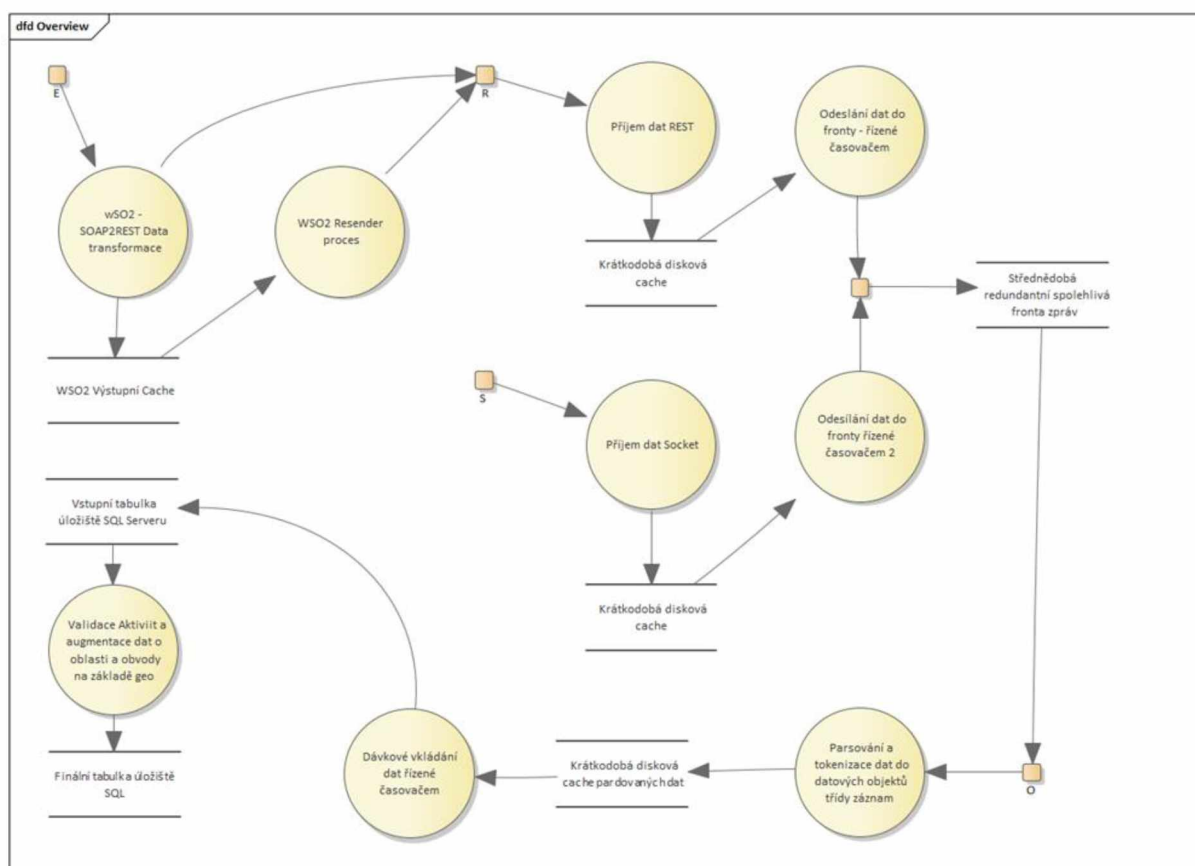
2.2.3.2 Mechanismus

Data budou předávána na rozhraní ŘSD, které se nachází na veřejné URL adrese specifikované v dokumentaci na stránkách <https://podporagps.rsd.cz/> v datovém formátu určeném v samostatném dokumentu **Technický předpis datového formátu telemetrických údajů**, a to vždy v pořadí od nejstarších záznamů po nejnovější.

2.2.3.3 Obsah předávaných dat

Data budou odpovídat datům, která vznikají na GPS jednotkách.

2.3 Přehled součástí serveru ŘSD obsluhujícího rozhraní pro příjem telemetrických dat



Výše uvedené schéma DTD (data transfer diagram) je přiloženo pro informaci a může být vodítkem pro pracovníky IT poskytovatelů telemetrických dat při úvahách o návrhu a realizaci klientských služeb pro předávání telemetrických dat.

Vstupní body R a S reprezentují veřejná rozhraní. R reprezentuje HTTP/REST API rozhraní, S reprezentuje Socket TCP/IP rozhraní.

Vstupní bod E reprezentuje rozhraní HTTP/SOAP, které již není ve verzi 1.1 podporováno pro nově uzavírané smlouvy a je zachováno pouze z důvodu zpětné kompatibility s rozhraním verze 1.0 platným pro dobíhající smlouvy.

Vstupní bod O je interní a veřejně nepřístupný.

Z uvedeného schématu vyplývá, že příjem telemetrických dat a jejich zpracování v systému ŘSD probíhá asynchronně a pro účely kompenzace vysokých zatížení v určitých momentech, například ve chvíli nepříznivých meteorologických podmínek je odděleno přijetí dávky telemetrických dat od jejího parsování, validace obsahu a zavedení do relační databáze střednědobou frontou.

Z tohoto uspořádání vyplývají i některé zásadní charakteristiky systému pro příjem telemetrických dat v interakci s klientskými službami zasílání telemetrických dat na straně poskytovatelů.

- Převzetí telemetrických dat na rozhraní je synchronní, ale další zpracování je asynchronní, z toho vyplývá, že rozhraní R i S vracejí návratové hodnoty chybových stavů související pouze s komunikací, předáním a převzetím datové sady a s formátem datové sady. Případné chyby a vady obsahu datové sady (nevalidní rozsah hodnot, chybějící povinné elementy a datové věty pro daný typ provozovaného vozidla a další) zde vyhodnocovány nejsou a nejsou tedy ani součástí synchronní odezvy.
- Veškeré pokyny uvedené v tomto předpisu, týkající se frekvence předávání dat a případných časových limitů se vztahují na předání datové sady prostřednictvím jednoho ze synchronních rozhraní R nebo S a poskytovatel dat nemusí počítat s žádnou rezervou na zpracování dat vnitřními mechanismy systémů ŘSD
- O každé ať již úspěšně nebo neúspěšně předané datové sadě se v rámci celého mechanismu jejího zpracování v systémech ŘSD vede auditní záznam – stopa. Do této auditní stopy jsou zaznamenávány i případné problémy s obsahem dat, rozsahem hodnot atd. Přístup k auditním stopám je k dispozici pověřeným pracovníkům ŘSD.
- V případě rozhraní R je k dispozici API nazvané R-ERR, zprostředkující feed chybových záznamů z auditní stopy, včetně identifikátoru původní předávané datové sady. Záznamy ve feedu jsou odstraňovány s předáním klientské službě voláním R-ERR API nebo, nejsou-li vyzvednuty, tak po 24 hodinách od vzniku.

2.4 Protokoly a rozhraní

2.4.1 TCP/IP – Rozhraní S

Klient se připojí k serveru na předem definovanou adresu URL a port.

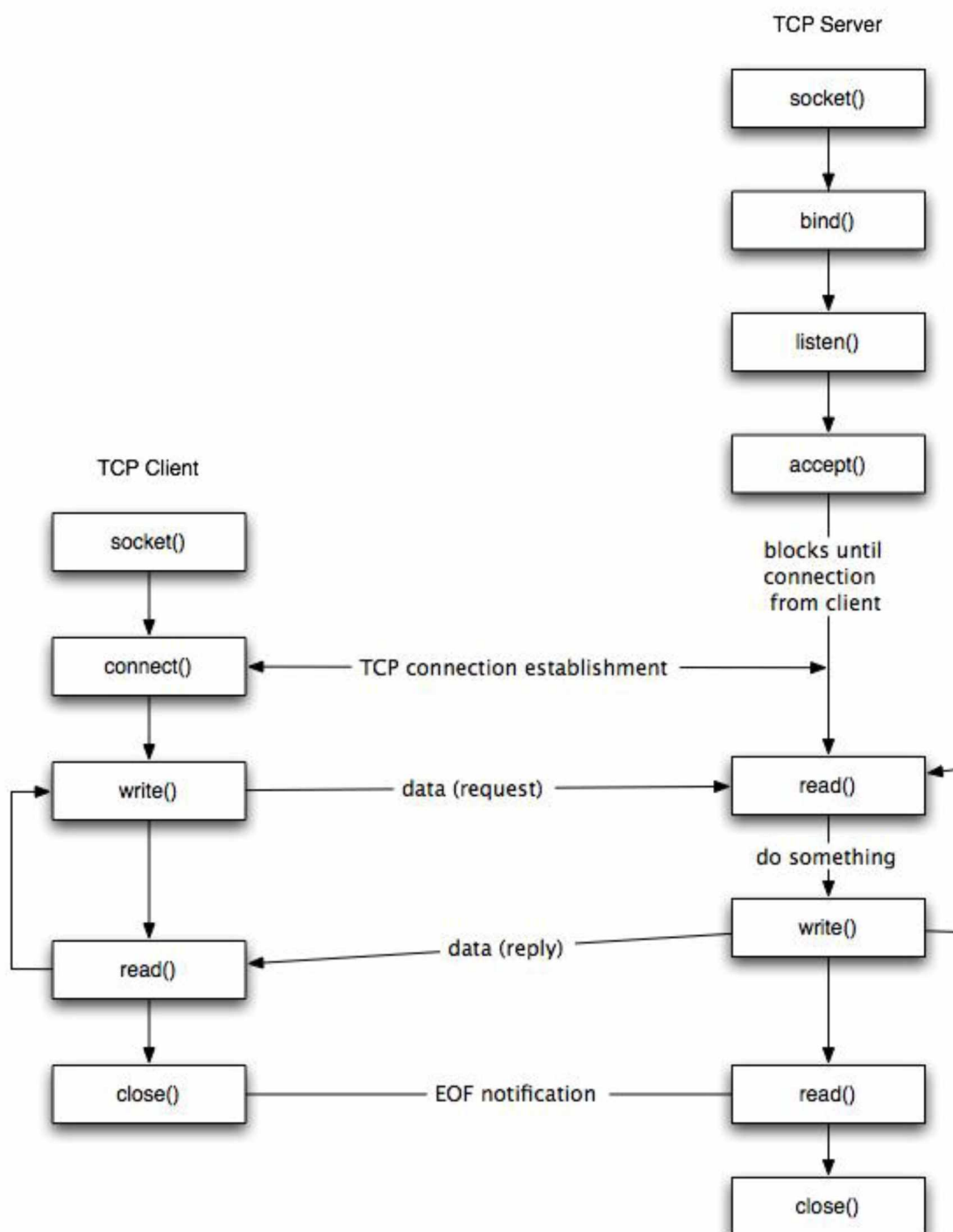
Např. GPST.RSD.CZ:45123

Po navázání spojení se přenese celý obsah zprávy, která je tvořena daty ve formátu XML obsahujícími standardní XML hlavičku a vlastní data v rootovém elementu <DOC> </DOC> .

Server při příjmu dat kontroluje, zda datový blok XML obsahuje počátek a zakončení rootového elementu. Přijetím rootového elementu </DOC> , očekává zároveň převzetí odezvy klientem.

Po ukončení přenosu dat se klient přepne do režimu příjmu a přijme zprávu o chybách přenosu, která obsahuje, v případě korektně přijatých dat pouze dva znaky „OK“, v případě, že v přenosu dat došlo k chybě, obsahuje její kód, a detailní popis, jehož délka se může lišit. Tato kontrola slouží k zabezpečení přenosu a vyloučení chyb během přenosu.

2.4.1.1 Komunikační diagram



Komunikace je synchronní, očekává se, že klient po odeslání každé relace počká s další relací na potvrzení předcházející přijetí „OK“. V případě, že server vrátí cokoliv jiného než „OK“, má se za to, že data nebyla úspěšně přenesena. Výjimkou je chybový kód „44X“, zde došlo k přenosu zprávy, ale klient odmítl převzít výsledek přenosu. V případě, že tento výsledek byl OK, zpráva je přijata.

U veškerých přenosů je předpokládáno kódování textu UTF-8.

2.4.1.2 Komunikace na socketu - zásady

- 1/ Přijímat odpověď – data považovat za odeslaná až v případě potvrzení zprávou „OK“
- 2/ Přijímat a reagovat na chyby - jsou zasílány jako odpověď na komunikaci
- 3/ Neresetovat zbytečně spojení v průběhu
- 4/ Nezasílat zprávy delší než 512 KB (přibližně), nebo vyžadující konektivitu a přenos delší než 3 sekundy
- 5/ Nezasílat z jednoho klienta více než 3 spojení za sekundu (nejedná se o bloky zpráv, ale opravdu o spojení)
- 6/ Respektovat limit max. 10 konkurenčních klientů a umět reagovat na odmítnutí spojení a případné chyby 46X– v případě prokazatelné potřeby lze individuálně dojednat navýšení škálováním do šířky a load balancerem
- 7/ Řešení bylo navrženo na rovnoměrnou komunikaci s jednotlivými GPS jednotkami, koncentrace a dávkové zasílání může znamenat přetížení, nesnažte se tedy data ukládat v bufferech na straně klientských služeb a odesílat je hromadně.

2.4.1.3 Technická omezení a doporučení

Parametr	Rozmezí	Doporučená hodnota
Prodleva mezi relacemi	$\geq 10\text{ms}$	20ms
Prodleva po neúspěšném pokusu o navázání relace	$\geq 100\text{ms}$	1s
Velikost přenášené zprávy	1kB - 786 kB	~ 64 kB
Doba trvání relace	50ms – 1200ms	$< 1000\text{ms}$
Paralelní relace	≤ 10	

2.4.1.4 Zabezpečení

Metody pro autentizaci a šifrování komunikace nebyly na vyžádání poskytovatelů dat implementovány.

2.4.1.5 Chybové stavy a očekávaná reakce na straně klientské služby

Chyby a odezvy	Očekávaná reakce klienta
OK	Odeslání datové sady proběhlo bez problémů
41X – Problémy navázání konexe	Kontrola kvality konektivity do internetu
42X – Neplatná komunikace a uzavření kanálu klientem	Ověření správné funkce klientské aplikace

43X – neplatný obsah zprávy, neúplná zpráva neobsahující konec</DOC>	Nové odeslání úplné zprávy nebo zprávy ve správném formátu XML
44X – klient odmítl přijmout potvrzení OK nebo oznámení chyby	Provést kontrolu přijímání potvrzení a chyb, ale konkrétní datovou sadu již vícečetně nezasílat
45X – neočekávaná přerušeni komunikace RST a timeout	Kontrola kvality konektivity do internetu
46X – Session Flood – příliš velký počet spojení, příliš dlouhá zpráva, příliš mnoho klientů	Omezit počet navazovaných spojení, jejich frekvenci a dobu odesílání datových sad, tak jak je uvedeno v bodu „Komunikace na socketu – zásady“
47X - Problémy přístupu k rozhraní fronty 48X - Zámek nebo timeout při odeslání do fronty 49X - Jiná chyba modulu Socket	Zvážit snížení frekvence odesílání datových sad, pokud je výrazně vyšší, než požadované minimum dle ustanovení tohoto dokumentu. V případě opakovaného výskytu kontaktovat podporu v ŘSD

2.4.1.6 Ukázkový kód (.NET Core – C#)

```
using Microsoft.Extensions.Logging;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;

namespace SSL_TCP_Protocol_Client
{
    /// <summary>
    /// Non secure Tcp client - for non-server actions
    /// </summary>
    public class NsTcpClient
    {
        #region Private NsTcpClient properties
        private readonly TcpClient _client;
        private readonly ILogger _logger;
        private readonly NetworkStream _networkStream;
        #endregion

        #region NsTcpClient constructor
        public NsTcpClient(string machineName, ILogger logger)
        {
            _logger = logger;
            // Create a TCP/IP client socket.
            // machineName is the host running the server application.

            try
            {
                TcpClient _client = new TcpClient();
                IPEndPoint endPoint = new IPEndPoint(IPAddress.Loopback, 3050);
                _client.Connect(endPoint);
                _logger.LogDebug("Client connected.");
                // Create a NetworkStream to access the client's stream.
                _networkStream = _client.GetStream();
            }
        }
    }
}
```

```

        catch (SocketException err)
        {
            _logger.LogError(err.Message);
        }
    }
}

public NsTcpClient(string address,int port, ILogger logger)
{
    _logger = logger;
    // Create a TCP/IP client socket.
    // machineName is the host running the server application.

    try
    {
        TcpClient _client = new TcpClient();
        IPEndPoint endPoint = new IPEndPoint(IPAddress.Parse(address), port);
        _client.Connect(endPoint);
        _logger.LogDebug("Client connected.");
        // Create a NetworkStream to access the client's stream.
        _networkStream = _client.GetStream();
    }
    catch (SocketException err)
    {
        _logger.LogError(err.Message);
    }
}

#endregion

#region Public methods ReadMessage, SendMessage and CloseClient
public Task<string> ReadMessage()
{
    // Read the message sent by the server.
    // The end of the message is signaled using the
    // "<EOF>" marker.
    byte[] buffer = new byte[2048];
    StringBuilder messageData = new();
    int bytes;
    do
    {
        bytes = _networkStream.Read(buffer, 0, buffer.Length);
        // Use Decoder class to convert from bytes to UTF8
        // in case a character spans two buffers.
        Decoder decoder = Encoding.GetEncoding("utf-8").GetDecoder();
        char[] chars = new char[decoder.GetCharCount(buffer, 0, bytes)];
        decoder.GetChars(buffer, 0, bytes, chars, 0);
        messageData.Append(chars);
        // Check for /DOC.
        if (messageData.ToString().IndexOf("</DOC>") != -1)
        {
            break;
        }
    } while (bytes != 0);

    return Task.Run(() => messageData.ToString());
}

public Task SendMessage(string messageData)
{
    int charscn = messageData.Length;
    int loopctr = 0;
    do

```



```

    {
        int chunkLength = (messageData.Length - 1024 * loopctr) >= 1024 ? 1024: (messageData.Length - 1024 *
loopctr);
        char[] messagePart = messageData.Substring(1024 * loopctr, chunkLength).ToCharArray();

        // Use Encoder class to convert from bytes from UTF8
        // in case a character spans two buffers.
        Encoder encoder = Encoding.UTF8.GetEncoder();

        byte[] bytes = new byte[encoder.GetByteCount(messagePart, 0, chunkLength, true)];

        encoder.GetBytes(messagePart, 0, chunkLength, bytes, 0, true);

        _networkStream.Write(bytes);

        charscn -= chunkLength;
        loopctr++;

    } while (charscn != 0);
    return Task.CompletedTask;
}
public void CloseClient()
{
    if (_networkStream != null && _networkStream.CanWrite) _networkStream.Close();
    if (_client != null) _client.Close();
    _logger.LogDebug("Client closed.");
}
}
#endregion
}
}

```

2.4.1.7 Testová metoda pro Unit testy

```

[TestMethod]
public async void Test_RSD_NS_Client()
{
    using var loggerFactory = LoggerFactory.Create(builder => builder.AddFilter("Microsoft", LogLevel.Warning)
        .AddFilter("System", LogLevel.Warning)
        .AddFilter("UnitTest1", LogLevel.Debug));
    logger = loggerFactory.CreateLogger<UnitTest1>();

    NsTcpClient nsClient = null;

    // Directory, where are the tes data sets stored as XML files
    string[] files =
        Directory.GetFiles(@"X:\RSD\GPS_vstup", "*.xml", SearchOption.AllDirectories);

    nsClient = new NsTcpClient("grv-gpst.rsd.cz", logger, false);
    Assert.IsNotNull(nsClient);

    foreach (string fileName in files)
    {
        // Send message to the server.
        await nsClient.SendMessage(File.ReadAllText(fileName));
        // Read message from the server.
        string serverMessage = await nsClient.ReadMessage();
        Assert.AreEqual(serverMessage, "OK");
    }
}

```

```
    nsClient.CloseClient();  
}
```

2.4.2 REST API – Rozhraní R

Rozhraní umožňuje zasílat data zpráv GPS na REST rozhraní definované následujícím Swagger popisem:

2.4.2.1 Definice REST API

```
{  
  "openapi": "3.0.1",  
  "info": {  
    "title": "GPS_Records_REST_Server",  
    "version": "v1"  
  },  
  "paths": {  
    "/GPSRecords/HeartBeat": {  
      "get": {  
        "tags": [  
          "GPSRecords"  
        ],  
        "responses": {  
          "200": {  
            "description": "Success"  
          }  
        }  
      }  
    },  
    "/GPSRecords/State": {  
      "get": {  
        "tags": [  
          "GPSRecords"  
        ],  
        "responses": {  
          "200": {  
            "description": "Success"  
          }  
        }  
      }  
    },  
    "/GPSRecords/PostMessage": {  
      "post": {  
        "tags": [  
          "GPSRecords"
```

```

],
"parameters": [
  {
    "name": "messageId",
    "in": "query",
    "schema": {
      "type": "string",
      "format": "uuid"
    }
  },
  {
    "name": "remoteIPAddress",
    "in": "query",
    "schema": {
      "type": "string",
      "nullable": true
    }
  }
],
"requestBody": {
  "content": {
    "text/plain": {
      "schema": {
        "type": "string",
        "nullable": true
      }
    }
  }
},
"responses": {
  "200": {
    "description": "Success"
  }
}
}
},
"components": { }
}

```

2.4.2.2 *Metody rozhraní*

- Metoda **/GPSRecords/ HeartBeat** se použije jako ověření, že je rozhraní připraveno a nedochází k timeoutu.
- Metoda **/GPSRecords/ PostMessage** přijímá jako parametry hodnoty "messageId" a "remoteIPAddress", kde messageId je jednoznačným identifikátorem zprávy ve formátu GUID, sloužícím později k párování datové sady s feedem chyb auditního

záznamu a remoteIPAddress je řetězec uvádějící zdrojovou IP adresu. V těle zprávy pak metoda očekává data předávaná jako MIME "text/plain"

Datovou sadou je obsah zprávy, která je tvořena daty formátu XML dle **dokumentu Technický předpis datového formátu telemetrických**

2.4.2.3 Zabezpečení

Volání je prováděno se šifrováním protokolem HTTPS

2.4.2.4 Chybové stavy a reakce na chyby

Obě metody vracejí chybové kódy dle standardu protokolu http/2.0. Úspěšné volání metody je identifikováno návratovým kódem 200.

V případě chyby jsou vráceny kódy 500.YYY, kde YYY představuje vlastní detail kategorie kódu chyby:

31X REST – chyba volání metody

32X REST – chyby obsahu a kódování znaků

33X REST – nedostupnost nebo timeout interní cache

37X Problémy přístupu k rozhraní fronty

38X Zámek nebo timeout při odeslání do fronty

39X Jiná chyba modulu REST

Chyby 500.31X – 500.33X jsou vráceny na klienta, ostatní chyby jsou zapsány pouze do auditního logu.

Pouze chyby 500.33X jsou řešitelné opakovaným voláním metody se stejným obsahem s prodlením. U ostatních chyb vede takové řešení pouze k obdržení stejného chybového hlášení a důrazně se na klientu nedoporučuje.

2.5 Popis dat a formát

Data budou předávána v obecném a standardizovaném formátu XML (Extensible Markup Language).

Kompletní popis dat pro všechna vozidla vyplývá ze samostatného dokumentu **Technický předpis datového formátu telemetrických údajů**, kde jsou také uvedeny popisy, hodnoty, kterých nabývají, jednotky a informace v jakých případech jsou dané parametry povinné. V případě, že je nějaká odlišnost mezi vozidly ŘSD a dodavatelů údržby, je toto uvedeno v dokumentu jako poznámka ke konkrétní položce.

Podrobné informace o formátech, číselníky, příklady a návody jsou umístěny jako aktuální a předchozí podporované verze tohoto dokumentu a dokumentu **Technický předpis datového formátu telemetrických údajů** na stránkách <https://podporaGPS.rsd.cz>. Ke všem informacím uvedených na těchto stránkách je vedeno datum platnosti informace.

Objednatel si vyhrazuje právo změnit formální náležitosti komunikačního protokolu. K takové úpravě dat či datové komunikace Objednatel Dodavatele písemně vyzve s určením lhůty, dokdy musí Dodavatel přejít na nově určený protokol, přičemž tato lhůta nebude kratší než 6 měsíců od doručení výzvy Dodavateli.

2.6 Evidence užití konkrétních vozidel a nástaveb v rámci činností

Dodavatel je povinen evidovat jednotlivé činnosti a užití jednotlivých konkrétních nástaveb dle odstavce 3.2.1.2 v provozním deníku v systému ISUDaS a tento provozní deník musí být v souladu se zasílanými daty GPS.

Provozní deník slouží k zaznamenávání provozních údajů. Za každý den je veden jeden deník.

V provozním deníku je možné evidovat výjezdy a návraty vozidel, počasí a jiné události

(mimořádné události, poruchy, ad.)