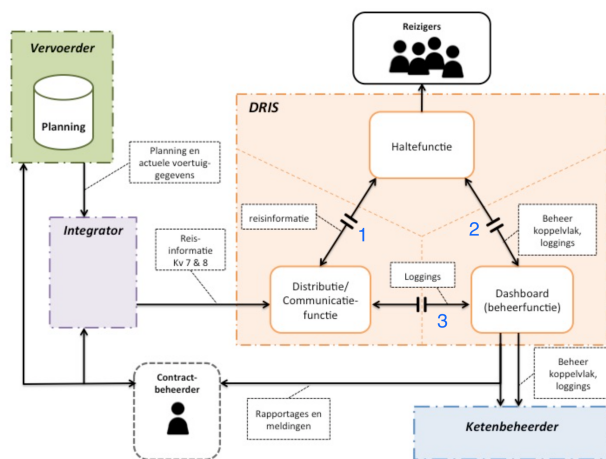


Open DRIS oppelvlakken

Versie 6.1 – 8 mei 2025 (zie ook versiebeheer)

Inleiding

De architectuur van Open DRIS voorziet in een driehoek tussen beheer-, distributie- en haltesystemen. Hierin zijn drie koppelvlakken nodig, namelijk die tussen het distributiesysteem en de haltesystemen (1 in onderstaand figuur), tussen de dashboard- en de haltesystemen (2) en tenslotte tussen het distributiesysteem en de dashboardsystemen (3). Dit document beschrijft deze drie koppelvlakken.



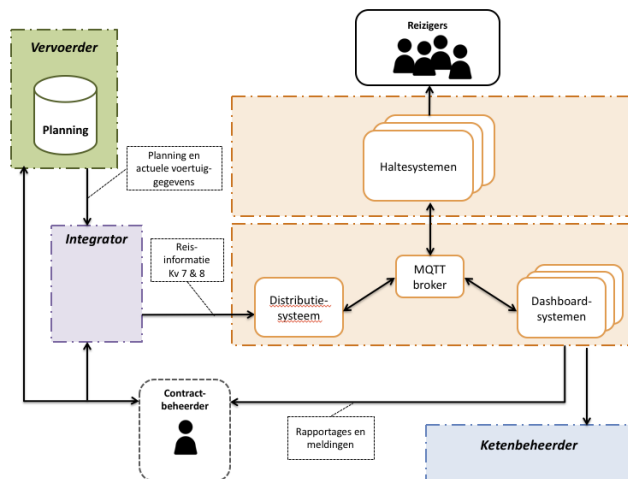
Reisinformatieketen voor DRIS

Het koppelvlak tussen het distributiesysteem en de haltesystemen voorziet voornamelijk in het doorgeven van reisinformatie vanuit de Dataleverancier en het aanmelden op de juiste subset die van belang is voor een haltesysteem. Omdat er veel verschillende verschijningsvormen en informatiebehoeftes zijn, speelt filteren en specificeren welke velden van belang zijn voor het haltesysteem een grote rol.

Het koppelvlak tussen het distributiesysteem en de dashboardsystemen is relatief eenvoudig en is bedoeld voor het communiceren van noodzakelijke logging en statusinformatie.

Het koppelvlak tussen halte- en dashboardsystemen bevat status- en diagnose-informatie van de haltesystemen. Met behulp van deze informatie, kunnen allerlei partijen een eigen dashboard maken. Het is uitdrukkelijk de bedoeling dat dashboardsystemen alle aangesloten haltesystemen kunnen beheren.

De drie soorten systemen communiceren met elkaar via het MQTT 5 protocol en kunnen op die manier real-time push-driven informatie met elkaar uitwisselen. Alle communicatie tussen de actoren in deze architectuur verloopt via een centrale MQTT broker.



Daarnaast moeten alle berichten gedefinieerd en geserialiseerd worden in het Protobuf 3 formaat. Hierdoor zijn berichten compact en op veel verschillende platformen toepasbaar door beschikbare standaard implementaties.

De rest van dit document behandelt de verschillende functionele flows in de koppelvlakken, de inhoud van de verschillende berichten die nodig zijn voor deze flows en tenslotte worden de MQTT details behandeld die nodig zijn voor het versturen van berichten.

Granulariteit

De berichten in de koppelvlakken zijn onafhankelijk van technische implementaties. Berichten worden verstuurd naar de functionele eenheid haltesysteem, waarbij een haltesysteem altijd slechts één set reisinformatie kan tonen. In praktische termen: De berichten zijn gedefinieerd voor een gewone enkelvoudige halte, waarbij één technische installatie één lijst met passages toont op één of meer¹ schermen. Er is dus maar één schermbeeld per haltesysteem eventueel verdeeld over of getoond op meerdere schermen.

Een knooppunt waar meerdere displays staan die verschillende informatie geven is voor de koppelvlakken een verzameling losse haltesystemen, die ieder apart communiceren met de centrale en die ieder een eigen status en configuratie hebben. De koppelvlakken hebben geen mogelijkheid om over de status van een stationsprocessor te rapporteren of om functionaliteit op stationsniveau te ondersteunen.

¹ Bijvoorbeeld bij een dubbelzijdig display

Technische implementatie

Het distributiesysteem bestaat uit twee clusters op verschillende locaties. Elke cluster bestaat uit een MQTT broker plus een distributieserver. De twee clusters zijn met elkaar verbonden met een MQTT bridge en ze zijn aangesloten op dezelfde datastroom. Op die manier blijft de data in de beide clusters synchroon. Een haltesysteem bepaalt zelf op welk van de twee clusters het zich aanmeldt. Op enig moment is dus een deel van de haltesystemen in Nederland aangemeld op de ene cluster en een ander deel op de andere cluster¹.

Verbinding maken en onderhouden

Een haltesysteem krijgt toegang door zich met gebruikersnaam en wachtwoord aan te melden op de MQTT broker van één van de clusters.

Na aanmelden op de MQTT server moet het haltesysteem zich op de MQTT topics abonneren zoals in het MQTT hoofdstuk beschreven.

Het haltesysteem geeft bij het opzetten van de verbinding met de MQTT broker een keep alive interval op van 60 seconden en stuurt minimaal eenmaal per 60 seconden een bericht naar de MQTT broker. Dit kan een normaal bericht zijn of een MQTT Pingreq.

Indien het haltesysteem gedurende anderhalf maal het keep alive interval geen bericht ontvangt van de MQTT broker, concludeert het haltesysteem dat de verbinding is verbroken en gaat het proberen de verbinding met de MQTT broker opnieuw op te zetten. Als de MQTT broker de Connect niet bevestigt zal het haltesysteem steeds opnieuw proberen om de verbinding op te bouwen met een tijd tussen de pogingen van minimaal 30 en maximaal 3600 seconden. Na een random gekozen aantal pogingen (minimaal 2 en maximaal 5) gaat het haltesysteem de verbinding opzetten met de MQTT broker in de andere cluster. Als ook op 'de andere' cluster geen verbinding gemaakt kan worden, blijft het haltesysteem op die cluster proberen om de verbinding op te bouwen met een interval zoals hierboven genoemd.

Met opmerkingen [Gv1]: Voetnoot dat dit doorwerkt in de storingsduur en dus de beschikbaarheid (boete)

Aan- en afmelden op het distributiesysteem

Nadat een haltesysteem is verbonden met de MQTT broker en is geabonneerd op de juiste topics, kan het zich aanmelden bij de distributieserver, zodat het reisinformatie krijgt toegestuurd. Door het sturen van een aanmeldbericht abonneert het haltesysteem zich bij het distributiesysteem op de informatie van één of meer haltes, aangeduid met Quaycodes of één Stopplacecode zoals beschreven in het centraal halte bestand (CHB). Met hetzelfde aanmeldbericht maakt het haltesysteem zich ook bekend bij de dashboardsystemen. Die systemen gebruiken de Quay- of Stopplacecodes in het aanmeldbericht om praktische informatie zoals halternaam en straatnaam op te halen uit het CHB.

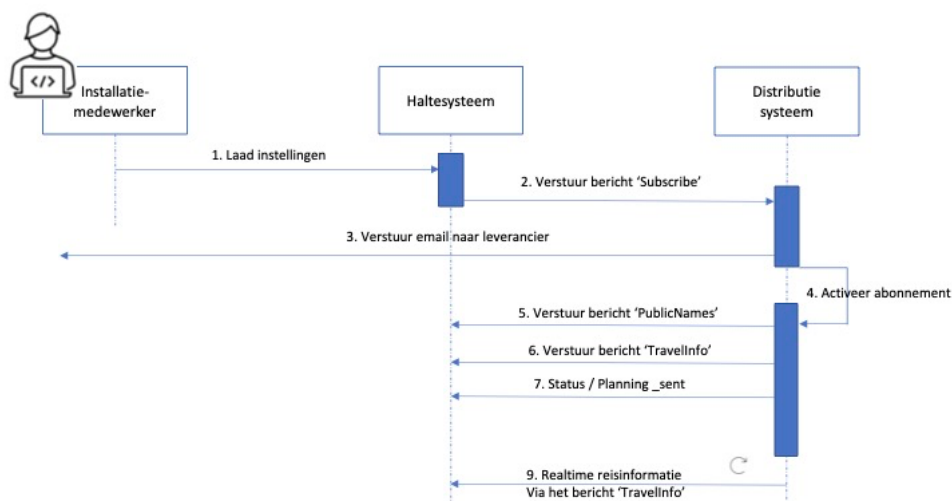
Aanmelden gebeurt op basis van een client id (zie hoofdstuk MQTT voor details). Verder bevat het aanmeldbericht een aantal optionele velden, met deze velden wordt het distributiesysteem geïnformeerd over welke gegevens het haltesysteem wil ontvangen.

Initiele aanmelding

Indien een haltesysteem zich voor de eerste keer aanmeldt wordt het haltesysteem direct actief en stuurt het distributiesysteem eenmalig een email naar de leverancier van het haltesysteem. De leverancier moet met deze emailberichten controleren dat alle haltesystemen die in het veld worden neergezet zich aanmelden en, net zo belangrijk, dat geen onbekende haltesystemen zich aanmelden. In het

¹ Dashboardsystemen zullen meestal aangesloten en aangemeld zijn op beide clusters.

aanmeldbericht kan ook een omschrijving worden ingevuld zodat verschillende systemen uit elkaar kunnen worden gehouden.



Het aanmeldproces: eerste keer per haltesysteem

Op het aanmeldbericht wordt in het distributiesysteem een aantal controles uitgevoerd: alle verplichte velden moeten aanwezig zijn, Quay- of Stopplacecodes moeten bestaan in het CHB en **de contract referentie moet bekend en geldig zijn (per contractreferentie wordt een emailadres afgesproken en vastgelegd in het distributiesysteem)**. Preconditie voor het aanmelden van een haltesysteem is dus dat de haltes waarover het haltesysteem informeert Quaycodes hebben in het CHB. Op alle aanmeldverzoeken wordt een antwoord teruggestuurd naar het haltesysteem met de status van de aanmelding (bericht 'SubscriptionResponse'). In geval van een foutief emailadres of ontbrekende informatie in het aanmeldbericht wordt de status 'REQUEST_INVALID' gecommuniceerd. Bij een stopcode die niet in CHB bekend is wordt de status 'STOP_INVALID' gecommuniceerd.

Als alle controles met goed gevolg zijn doorlopen stuurt het distributiesysteem de status 'AUTHORISATION_VALIDATED'. Daarna wordt de meest actuele versie van de in het distributiesysteem bekende ritten opgehaald en **worden** alle geldige en toekomstige vrije teksten opgehaald en naar het haltesysteem verstuurd. Indien er ritten bekend zijn voor het betreffende haltesysteem, wordt na het versturen van de ritten de status 'PLANNING_SENT' gestuurd. Indien er geen geplande ritten zijn in de komende periode wordt de status 'NO_PLANNING' gecommuniceerd. In dit geval is het abonnement echter wel actief en kunnen er op elk moment, ritten dan wel vrije teksten naar het haltesysteem worden gestuurd.

Dashboardsystemen kunnen zich ook abonneren op de aanmeldberichten en de antwoorden, zodat ze het aanmeldproces kunnen volgen en vastleggen.

Indien een haltesysteem zich twee keer aanmeldt zonder zich tussendoor af te melden geeft het distributiesysteem een status 'ALREADY_SUBSCRIBED' terug zonder verdere actie. Als het haltesysteem alle aanmeldinformatie nieuw wil ontvangen, moet het zich eerst afmelden en dan opnieuw aanmelden.

Na het aanmelden (initieel na autorisatie, of later na een Unsubscribe) stuurt het haltesysteem een SystemInfo bericht zodat de luisterende dashboardsystemen de status van het haltesysteem kunnen synchroniseren.

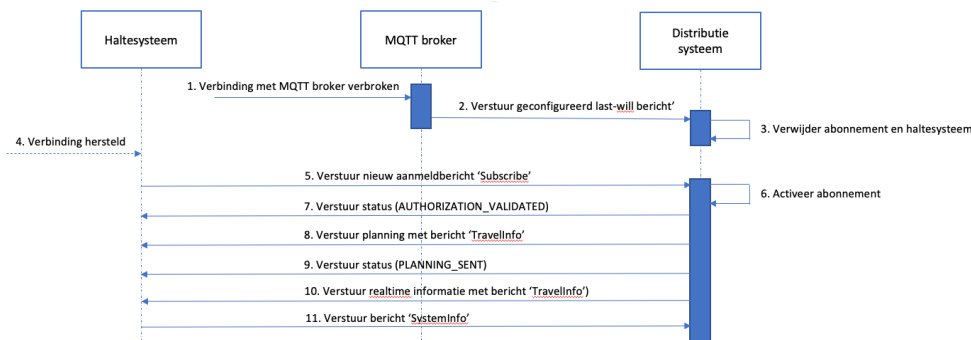
Let op dat het store and forward karakter van MQTT maakt, dat er geen garantie is dat berichten in dezelfde volgorde worden ontvangen als dat ze worden uitgestuurd. Bijvoorbeeld een verschil in de Quality of Service tussen twee berichten kan ervoor zorgen dat het ene bericht er langer over doet dan een ander.

Afmelden

Bij het opzetten van de verbinding met de broker configureert het haltesysteem het bericht Unsubscribe als een zogenaamd 'last-will' bericht. Indien de verbinding tussen haltesysteem en de MQTT broker wegvalt, publiceert de MQTT broker dit last-will bericht, dat door het distributiesysteem en de dashboardsystemen wordt ontvangen. Indien het distributiesysteem een last-will/afmeldbericht ontvangt zal het geen informatie meer sturen naar het haltesysteem tot het opnieuw is aangemeld. De dashboardsystemen zullen de status van het haltesysteem veranderen naar offline.

Voor het last-will bericht wordt het bericht Unsubscribe gevuld met 'is_permanent' = false. **De timestamp wordt gevuld met het huidige tijdstip.**

Ook het distributiesysteem kan zich afmelden door een UnSubscribe te versturen. Dit gebeurt bijvoorbeeld als een (cluster van het) distributiesysteem voor onderhoud offline gaat. Als een haltesysteem een Unsubscribe van (een cluster van) het distributiesysteem ontvangt, moet het zich opnieuw aanmelden, maar dan bij de andere cluster.



Het afmeldproces, gevolgd door het opnieuw aanmelden van het al eerder geautoriseerde haltesysteem

Om eventuele race-conditions te voorkomen is het van belang dat het haltesysteem zich ook abonneert op zijn eigen unsubscribe berichten (bijvoorbeeld verstuurd door de broker als last will bericht), en op unsubscribe berichten van het distributiesysteem. Na het ontvangen van een eigen unsubscribe bericht moet het haltesysteem een nieuw aanmeldbericht sturen naar de andere cluster van het distributiesysteem dan waar het op aangemeld was.

Het is daarnaast ook mogelijk om een haltesysteem permanent af te melden door middel van het Unsubscribe bericht, met waarde 'is_permanent' = true en het (huidige) tijdstip van afmelden. Dit is van belang indien een haltesysteem wordt verwijderd, gedeactiveerd of wordt vervangen. Dit voorkomt bijvoorbeeld extra logging van gegevens over een haltesysteem dat defect is en maakt (keten)beheer makkelijker. Indien een haltesysteem is afgemeld met 'is_permanent' = true en het haltesysteem meldt zich later weer aan (met dezelfde client id), moet het opnieuw **een eerste aanmelding doen**.

Reisinformatie

Op basis van het selectiefilter wordt er elke nacht een lijst van ritten met hun passeertijden in unix-timestamps verstuurd van distributie- naar haltesysteem. Hierdoor is de afnemer niet afhankelijk van operationele dagen, slechts het begrip een tijdstip in UTC om naar een tijdzone van het haltesysteem te converteren is van belang, in dit geval: Europe/Amsterdam. De passertijden worden geïdentificeerd door het veld 'pass_time_hash', dat constant blijft gedurende een rit. Voor de unix-timestamp wordt de variant gebruikt waarvan de waarde 0 gelijkstaat aan de datum 1-1-1970 00:00:00 GMT.

Als een haltesysteem zich aanmeldt bij het distributiesysteem, stuurt het distributiesysteem voor 62 uur passeertijden naar het haltesysteem. Met die gegevens kan het haltesysteem reisinformatie tonen ook als de verbinding met het distributiesysteem is wegvallen. Nadat de set met passeertijden voor 62 uur is gestuurd, stuurt het distributiesysteem met hetzelfde bericht de actuele passeertijden. Elke nacht wordt, weer met hetzelfde bericht, een set van 24 uur nieuwe passeertijden verstuurd voor de periode 38 – 62 uur. De geplande en de actuele passeertijden zijn voor het haltesysteem volkomen hetzelfde en worden gekenmerkt door een zelfde hashleutel (pass_time_hash).

Er zijn gevallen mogelijk waarbij er binnen de geldige geplande periode geen geplande ritten zullen plaatsvinden (bijvoorbeeld in het weekend, evenementenhaltes, etc.) en er ook geen geldige vrije teksten actief zijn op dat moment. Hierbij kan een geval ontstaan dat er geen reisinformatie voor het haltesysteem beschikbaar is. In dit geval wordt een statusbericht verstuurd naar het haltesysteem met status

'NO_PLANNING'

Met opmerkingen [Gv2]: Hier moet een stukje toegevoegd om een Subscribe met send_minimal te beschrijven

Standaard berichtafhandeling

Ieder bericht wordt gecodeerd via protocol buffers. Het uitwisselingsformaat zorgt ervoor dat datatypes horizontaal worden *gepacked*. Er zijn daardoor minder bits nodig als het datatype niet volledig wordt gebruikt. Daarnaast kiezen we binnen het protocol voor het principe van uitwisselen in kolommen. De meeste databases maken gebruik van rijen, 1 rij komt vaak overeen met de attributen van 1 object. Door niet in rijen, maar in kolommen uit te wisselen krijgen we een tweede optimalisatie: informatie die zich herhaalt (denk aan: bestemmingen en publieke lijnummers) staan dicht bij elkaar en kunnen slimmer worden gecomprimeerd. In een kolom georiënteerde database hebben alle kolommen hetzelfde aantal elementen. Een rij is op basis van de index samen te voegen. Element 1 uit kolom 1, komt daarmee overeen met element 1 uit kolom 2. Het haltesysteem heeft bij het aanmelden geselecteerd welke kolommen het wil ontvangen, alleen deze kolommen worden door het distributiesysteem gesynchroniseerd.

Het synchronisatieproces gaat uit van een vaste identificatie voor een rit, dit in de vorm van een hashleutel ('pass_time_hash'). Op deze manier is iedere passeertijd uniek identificeerbaar en overschrijfbaar. Op basis van het selectiefilter worden de actuele passeertijden naar het haltesysteem verstuurd.

```
passing_times {
```

```
pass_time_hash: 1487911693
pass_time_hash: 3375514872
target_departure_time: 1496090100
target_departure_time: 1496086440
expected_departure_time: 1496090100
expected_departure_time: 1496086457
trip_stop_status: PLANNED
trip_stop_status: DRIVING
destination {
  destination_name: "Marconiplein"
  destination_detail: ""
}
destination {
  destination_name: "Marconiplein"
  destination_detail: ""
}
line_public_number: "23"
line_public_number: "23"
generated_timestamp: 1496086082
generated_timestamp: 1496086082
}
general_messages {}
```

Voorbeeld bericht 'TravelInfo' gevuld met actuele reisinformatie voor twee ritten

Vrijeteksten

Naast passeertijden kunnen ook vrijeteksten gecommuniceerd worden naar een haltesysteem. De berichten waarmee vrijeteksten worden verstuurd hebben ook een hash en zijn uniek te identificeren met het veld 'message_hash'. De vrijetekst kan worden verwijderd met het 'GeneralMessageRemove' bericht. Het haltesysteem is verantwoordelijk voor het tonen van de vrijetekst tussen de message_start_time en de message_end_time, of tot de tekst wordt gewist. Elke vrijetekst wordt los verstuurd naar het haltesysteem, indien er meerdere vrijeteksten geldig zijn, bepaalt het haltesysteem hoe dit getoond wordt.

Overzichtdisplay

Indien een haltesysteem een overzicht moet tonen van meerdere quays zijn hier twee mogelijkheden voor.

- Het haltesysteem meldt zich aan op meerdere quays: alle QuayCodes (NL:Q:xxxxxxx) die onderdeel zijn van de StopPlace waarvoor het overzichtdisplay geldt, worden opgenomen in het aanmeldbericht. Reisinformatie en vrije teksten voor alle gevraagde quaycodes worden naar het haltesysteem verstuurd. Voordeel van deze aanpak is dat bijvoorbeeld een uitstaphalte die onderdeel is van een StopPlace kan worden uitgesloten.
- Het haltesysteem meldt zich aan op een StopPlace: één (of eventueel meerdere) StopPlace(s) worden opgenomen in het aanmeldbericht (NL:S:xxxxxxx). Op basis van CHB wordt voor de geldige Quaycodes reisinformatie verstuurd. Dit heeft als voordeel dat indien de samenstelling van de StopPlace wijzigt en het Centrale Halte Bestand wordt aangepast, het distributiesysteem dit opnieuw inleest en het overzichtdisplay automatisch meegaat.

Client Id

Als bijlage van dit document is het bestand DrisKoppelVlak(versie).proto toegevoegd. Dit bestand is leidend voor het koppelvlak.

In dit hoofdstuk worden de verschillende 'messages' verder uitgewerkt.

ClientId:

	Veld	Aantal	Uitleg
ClientId	subscriber_owner_code	(1)	De SubscriberOwnerCode wordt gevuld met een waarde uit een overkoepelende lijst. In deze lijst worden alle partijen beschreven die de distributie-, dashboard- of haltesystemen leveren. Deze bijlage wordt door DOVA OV_Data beheerd.
	subscriber_type	(1)	Geeft het type van de subscriber aan 0=distributiesysteem, 1=dashboardsysteem en 2=haltesysteem.
	serial_number	(1)	Volgnummer dat de combinatie van de velden uniek maakt. Voor een haltesysteem wordt geadviseerd om het serienummer van de hardware te gebruiken.

Berichten Koppelvlak Distributiesysteem - Haltesysteem

Subscribe

	Veld	Aantal	Uitleg
Subscribe	client_id	(1)	Id waarmee client zich aanmeld.
	stop_code	(1..n)	Een of meer quaycodes of één stopplacecode uit het Centrale Halte Bestand (CHB) waarop het haltesysteem zich wil abonneren en reisinformatie en vrije teksten van wil ontvangen. Beginnend met "NL:Q" of "NL:S". (verplicht). Er kan slechts geabonneerd worden op 1 stopplace (NL:S) en als meerdere quaycodes worden gestuurd moeten deze behoren tot dezelfde stopplace ¹ .
	display_properties	(0..1)	Geeft de kenmerken van het ontvangende haltesysteem aan
	text_characters	(0..1)	Het aantal tekens dat het display van het haltesysteem beschikbaar heeft voor de bestemmingsnaam. Het distributiesysteem kan daarmee bepalen welke bestemmingstekst (lengte) en detailbestemmingstekst gestuurd moeten worden. Indien deze waarde niet wordt gespecificeerd, wordt de langste waarde meegestuurd. Dit is van toepassing op haltesystemen die geen (praktische) lengtebeperking hebben.
	destination_determination	(1)	Geeft aan welke bestemmingsteksten naar het haltesysteem moeten worden verstuurd: - MAX_CHARACTERS het haltesysteem wil een bestemmingstekst en detail bestemmingstekst ontvangen met maximaal de lengte als gegeven in tekst_characters. Het distributiesysteem stuurt de eerstpassende bestemmingstekst en detailbestemmingstekst. (als tekst_characters = 18 wordt een bestemmingstekst 16 gestuurd.) - SELF_DETERMINING het haltesysteem wil alle bestemmingsteksten en detailbestemmingsteksten ontvangen die het distributiesysteem heeft en selecteert zelf welke tekst wordt getoond, rekening houdend met de mogelijkheden van de schermen.

¹ Het distributiesysteem controleert niet op deze regel, maar als een haltesysteem zich niet houdt aan deze regel kunnen er problemen optreden bij het beheer van het haltesysteem. Implementaties die zich niet houden aan deze regel kunnen worden uitgesloten van aansluiting en ketenbeheer.

		trips_per_packet	(0..1)	Geeft aan hoeveel ritten er maximaal per TravellInfo bericht (= TCP packet) verstuurd moeten worden ¹ . Als het veld niet gevuld is, stuurt het distributiesysteem maximaal 500 ritten in een TravellInfo bericht.
		contract_ref	(1)	De referentie naar het (onderhouds)contract waaronder het haltesysteem is geplaatst. De contractreferentie wordt uitgegeven door DOVA. Gebruik van overeengekomen contractreferentie is vereist voor toegang van haltesystemen tot het distributiesysteem. Bij het gebruik van een verkeerde contractreferentie voor een of meerdere haltesystemen wordt de toegang tot het distributiesysteem geblokkeerd voor de betreffende contractreferentie.
		language	(1..n)	De talen waarin het haltesysteem teksten wil ontvangen. (voor de enumeratie, zie destination in TravellInfo bericht)
	filter_parameters		(0..1)	Geeft de parameters (T1, T2, P1, P2) waarmee de KV8 reisinformatie gefilterd moet worden door het distributiesysteem. (zie bijlage 2)
	FilterParameters	filter_on	(0..1)	Geeft aan of er wel of niet gefilterd moet worden.
		waitingtime_low	(0..1)	De ondergrens voor de wachttijd (T1).
		waitingtime_high	(0..1)	De bovengrens voor de wachttijd (T1).
		percentage_low	(0..1)	Het percentage (P1) waarmee de wachttijd af moet wijken onder de ondergrens van de wachttijd (T1).
		percentage_high	(0..1)	Het percentage (P2) waarmee de wachttijd af moet wijken boven de ondergrens (T1).
	filter_filter		(0..1)	Geeft aan welke gegevens van een rit gestuurd moeten worden naar het haltesysteem. Elk veld van dit object correspondeert met een gelijknamige kolom in het bericht 'PassingTimes' waarvan wordt aangegeven hoe dit moet worden verstuurd. Geldige waardes zijn opgenomen in de enumeratie 'Delivery'. Mogelijk zijn: - NEVER: standaardwaarde. Stuur dit veld nooit. (Protobuff 3 kent geen optionele velden. Als een zender (repeated) velden leeg laat, wordt er geen data voor deze velden in het bericht geplaatst. Echter bij

¹ Kleine haltesystemen hebben soms problemen met het ontvangen en verwerken van langere berichten. Daarom kan het haltesysteem hier aangeven hoeveel ritten het in één TravellInfo bericht wil hebben.

FieldFilter			het deserialiseren aan de ontvangende kant worden de velden gevuld met een defaultwaarde, zoals gespecificeerd in de protobuf documentatie.) - ALWAYS: in elk bericht wordt de meest recente waarde gestuurd. Elke onderstaand veld spreekt voor zich: de waarde van het veld is in het PassingTimes bericht gedocumenteerd.
	target_arrival_time	(0..1)	De aankomsttijd volgens dienstregeling. Deze kan meegestuurd worden, bijvoorbeeld om vertraging relatief aan die tijd te tonen
	target_departure_time	(0..1)	De vertrektijd volgens dienstregeling. Deze kan meegestuurd worden, bijvoorbeeld om vertraging relatief aan die tijd te tonen
	expected_arrival_time	(0..1)	De verwachte aankomsttijd. Kan gebruikt worden voor een haltesysteem dat aankomsttijden toont.
	expected_departure_time	(0..1)	De verwachte vertrektijd. Kan gebruikt worden voor een haltesysteem dat vertrektijden toont.
	number_of_coaches	(0..1)	Dit geeft aan hoeveel (gekoppelde) rytuigen/voertuigen op een rit worden ingezet. Dit kan getoond bij metro/tram haltes, maar kan ook bij andere modaliteiten.
	trip_stop_status	(0..1)	Dit veld geeft aan wat de status van de rit is. Displays die alleen vertrektijden volgens dienstregeling weergeven, zouden ervoor kunnen kiezen dit veld niet te ontvangen.
	transport_type	(0..1)	Onder andere nodig om het juiste icoon te tonen als het voertuig op de halte aankomt of de wachttijd 0 minuten is. E.e.a. conform de Mijksenaar richtlijn.
	wheelchair_accessible	(0..1)	Hiermee kan op een display getoond worden of het voertuig van een rit toegankelijk is voor rolstoelen.
	is_timingstop	(0..1)	Hiermee bepaalt het haltesysteem welke informatie getoond wordt na aankomst van het voertuig op de halte.
	stop_code	(0..1)	Relevant voor haltesystemen geabonneerd op meerdere quays, waarbij het van belang is te weten op welke quay een vertrek betrekking heeft.
	destinations	(0..1)	
	show_cancelled_trip	(0..1)	

	block_code	(0..1)	
	occupancy	(0..1)	
	line_public_number	(0..1)	
	side_code	(0..1)	
	line_direction	(0..1)	Relevant om te kunnen filteren op het aantal te tonen ritten per lijn en richting.
	line_color	(0..1)	Relevant voor haltesystemen die een lijnachtergrondkleur kunnen tonen
	line_text_color	(0..1)	Relevant voor haltesystemen die een lijnvoorgndkleur kunnen tonen
	line_icon	(0..1)	Relevant voor haltesystemen die een icoon of afbeelding kunnen tonen
	destination_color	(0..1)	Relevant voor haltesystemen die een bestemmingsachtergrondkleur kunnen tonen
	destination_text_color	(0..1)	Relevant voor haltesystemen die een bestemmingsvoorgndkleur kunnen tonen
	destination_icon	(0..1)	Relevant voor haltesystemen die een bestemmingsicoon of afbeelding kunnen tonen
	generated_timestamp	(0..1)	Relevant om eventueel "out of order" berichten te kunnen detecteren en te kunnen verifiëren dat altijd de laatste informatie wordt gebruikt
	journey_number	(0..1)	Ritnummer voor eventueel traceren van een rit op verschillende haltes
	line_border_color	(0..1)	Relevant voor haltesystemen die een randje rond de lijnachtergrond moeten tonen (De Lijn)
	line_text_border_color	(0..1)	Relevant voor haltesystemen die een randje rond het lijnnummer moeten tonen (De Lijn)
	description	(0..1)	Een omschrijving van de locatie van het haltesysteem of een begeleidende tekst. Wordt door het distributiesysteem meegestuurd in het aanmeldproces en kan door het dashboard worden getoond
	timestamp	(1)	Tijdstip waarop het meldbericht is samengesteld.
	send_minimal	(0..1)	Vlag waarmee het haltesysteem aan kan geven of het al dan niet alle ritpassages toegestuurd wil krijgen. Default wordt alles gestuurd ¹ .

¹ Als deze vlag is geset, stuurt de CDD alleen de passeermomenten van ritten met een trip_stop_status ongelijk aan PLANNED. Alle andere passeermomenten en informatie over vrijeteksten wordt wel gestuurd. Op deze manier wordt voorkomen dat bij elke onderbreking van de communicatie alle geplande data (die niet is gewijzigd) wordt verstuurd.

SubscriptionResponse

	Veld	Aantal	Uitleg
SubscriptionResponse	success	(1)	Boolean-waarde die aangeeft of het haltesysteem succesvol aangemeld is bij het distributiesysteem (samenvatting van onderstaande status)
	status	(1)	Een van de volgende statussen ¹ (zie ook hoofdstuk 'Flows'): <i>REQUEST_INVALID</i> - Het aanmeldbericht voldoet niet aan de specificatie: een of meerdere velden is ongeldig. <i>STOP_INVALID</i> - Het aanmeldbericht heeft de juiste opbouw, maar het haltesysteem meldt zich aan op een bij het distributiesysteem onbekende stopplace- of quaycode (op basis van bron CHB). <i>AUTHORISATION_REQUIRED</i> - Het aanmeldbericht is juist, maar de betreffende combinatie van e-mailadres en systeemidentificatie zijn nog niet geautoriseerd. <i>PLANNING_SENT</i> - Indien het haltesysteem succesvol is aangemeld en er een planning is, wordt deze status verstuurd na het succesvol versturen van de planning. <i>NO_PLANNING</i> - Indien de aanmelding succesvol is, maar er geen planning is voor geen van de aangevraagde quay- of stopplacecodes, wordt deze status verstuurd ² . <i>AUTHORISATION_VALIDATED</i> - Het systeem is door een gebruiker zojuist geautoriseerd en de aanmelding is hiermee voldaan. <i>ALREADY_SUBSCRIBED</i> - Het systeem is al aangemeld bij het distributiesysteem. Indien het systeem zich opnieuw wil aanmelden (en alle informatie wil ontvangen) moet het zich eerst afmelden.
	timestamp	(1)	Tijdstip waarop het antwoord op het aanmeldbericht is samengesteld.

¹ Er is in de enumeratie 'Status' ruimte gelaten voor eventuele toekomstige uitbreidingen. De responses zijn gegroepeerd op categorie.

² Dit betekent letterlijk dat er geen passeertijden zijn voor de aangemelde halte(s). Er kunnen wel vrijeteksten zijn.

PublicName

Na de **aanmelding van een haltesysteem**, haalt het distributiesysteem de haltegegevens uit het CHB. Indien alle quaycodes of de stopplacecode die het haltesysteem heeft gestuurd bekend zijn in het CHB, wordt het bericht PublicName gestuurd met de gegevens van de quays plus de bovenliggende stopplace en place, indien gedefinieerd.

	Veld	Aantal	Uitleg
PublicName	place_names	(0..1)	Lijst met de naam van de place in verschillende talen.
	PlaceName	Public_name_place	(0..1) De naam van de place zoals bekend bij het publiek.
		language_place	(0..1) De taal van de naam. (voor de enumeratie, zie destination in TravellInfo bericht)
	stop_place_code	(0..1)	De code van de stopplace (NL:S) waar de quaycodes toe behoren.
	stop_place_names	(0..1)	Lijst met de naam van de stopplace in verschillende talen.
	StopPlaceName	public_name_stop_place	(0..1) De naam van de stopplace zoals bekend bij het publiek.
		language_stop_place	(0..1) De taal van de naam. (voor de enumeratie, zie destination in TravellInfo bericht)
	quay_code	(1..n)	De code van de Quay (NL:Q) waarover het haltesysteem informeert.
	quay_names	(1..n)	Lijst met de naam van de Quay in verschillende talen.
	QuayName	public_name_quay	(1..n) De naam van de Quay zoals bekend bij het publiek en zoals vastgelegd in het CHB.
		Language_quay	(1..n) De taal van de naam. (voor de enumeratie, zie destination in TravellInfo bericht)

Een haltesysteem kan zich aanmelden met één of meer quaycodes¹ of met één stopplacecode.

Als een haltesysteem zich bijvoorbeeld aanmeldt voor data van stopplace 'NL:S:50000101', dan wordt de volgende data verstuurd in **PublicName**.

```
public_name_place = [Utrecht centraal];
public_name_stop_place = [CS Jaarbeurszijde];
stop_place_code = [NL:S:50000101];
```

¹ De quaycodes moeten tot 1 stopplace behoren.

```
stop_code=[NL:Q:50000120,NL:Q:50000121,NL:Q:50000122,NL:Q:50000123,NL:Q:50000124,NL:Q:50000125,NL:Q:50000126,NL:Q:50000127,NL:Q:50000128,NL:Q:50000129];
```

```
public_name_quay = [Halte C1, Halte C2, Halte C3, Halte C4, Halte C5, Halte C6, Halte C7, Halte C8, Halte C9, Halte C10];
```

Als een haltesysteem zich aanmeldt voor data van quay 'NL:Q:50000120' dan wordt de volgende data verstuurd in public_names.

```
public_name_place = [Utrecht centraal];
```

```
public_name_stop_place = [CS Jaarbeurszijde];
```

```
stop_place_code = [NL:S:50000101];
```

```
public_name_quay = [Halte C1];
```

```
stop_code=[NL:Q:50000120];
```

Het distributiesysteem controleert regelmatig het CHB op wijzigingen. Als de gegevens van de quays of de stopplace wijzigen, stuurt het distributiesysteem een Unsubscribe, zodat het haltesysteem zich opnieuw aanmeldt en de nieuwe gegevens ontvangt.

Unsubscribe

	Veld	Aantal	Uitleg
Unsubscribe	client_id	(1)	Client Id waarmee systeem is aangemeld
	is_permanent	(1)	<i>Indien nee:</i> dit betreft meestal een last-will bericht als het haltesysteem tijdelijk zijn verbinding met de MQTT broker is verloren. Maar het kan ook verstuurd worden door het distributiesysteem als het gepland uit de lucht gaat. Het wordt ook verstuurd door het distributiesysteem als er voor de halte wijzigingen in het CHB worden geconstateerd. <i>Indien ja:</i> het systeem wordt buiten dienst genomen of wordt niet meer op de huidige manier ingezet. De gegevens van het haltesysteem worden door het distributiesysteem gewist en er wordt geen informatie meer naar het haltesysteem verstuurd.
	timestamp	(0..1)	Tijdstip waarop het bericht is samengesteld.

Na de ontvangst van een Unsubscribe van een haltesysteem (of van zichzelf) wacht het distributiesysteem op het haltesysteem om zich opnieuw aan te melden met een Subscribe. In het distributiesysteem wordt het haltesysteem inactive (=offline) tot het zich opnieuw aanmeldt¹.

Als het haltesysteem zich afmeldt met de permanent vlag geset, worden alle gegevens van het haltesysteem uit het distributiesysteem (beide clusters) verwijderd en bestaat het voor het distributiesysteem niet meer.

¹ Let op dat het distributiesysteem bestaat uit twee clusters. Een haltesysteem kan de verbinding met cluster A verliezen en een nieuwe verbinding opzetten met cluster B en zich daar aanmelden. Voor cluster A is het haltesysteem dan inactive en voor cluster B active.

TravelInfo

	Veld	Aantal	Uitleg
TravelInfo PassingTime	passing_times	(0..1)	Bevat een lijst met passeertijden van vertrekkende ritten (PassingTimes)
	pass_time_hash	(1..n)	(Zie ook "Standaard berichtafhandeling") Voor elke vertrekkende rit wordt een unieke hash berekend dat het haltesysteem kan gebruiken om de ritpassage te identificeren. Dit veld wordt altijd gevuld.
	target_arrival_time	(0..n)	Aankomsttijd volgens dienstregeling van de rit op de halte. Dit kan gebruikt worden om de vertraging ten opzichte van de dienstregeling te berekenen en tonen (bijvoorbeeld "11:10 +3"). De waarde betreft een unix timestamp, oftewel het aantal seconden sinds 1 januari 1970, in UTC. Afnemend haltesysteem is verantwoordelijk om deze informatie te vertalen naar de huidige tijd in tijdzone "Europe/Amsterdam" (tenzij anders gespecificeerd). Dit veld wordt niet verstuurd voor een beginhalte.
	target_departure_time	(0..n)	Vertrektijd volgens dienstregeling van de rit op de halte. Dit betreft een timestamp (zie hierboven). Dit veld wordt niet verstuurd voor een eindhalte.
	expected_arrival_time	(1..n)	De actuele prognose van de aankomsttijd op de betreffende halte van deze rit. Dit betreft een timestamp (zie hierboven), dit veld is altijd leeg voor beginhaltes. Het veld is altijd gevuld met de best bekende aankomsttijd.
	expected_departure_time	(1..n)	De actuele prognose van de vertrektijd op de betreffende halte van deze rit. Dit betreft een timestamp (zie hierboven), dit veld kan niet worden gefilterd, en is altijd leeg voor eindhaltes. Het veld is altijd gevuld met de best bekende vertrektijd.
	number_of_coaches	(1..n)	Het aantal rytuigen of gekoppelde voertuigen dat deze rit uitvoert.
	trip_stop_status	(1..n)	Ritstatus voor het vertrek op de betreffende halte van de rit. Waarde is op basis van de enumeratie TripStopStatus ¹ , met als mogelijke waarden: - PLANNED - de rit is gepland - CANCELLEDJOURNEY - de rit is opgeheven door een probleem van tijdelijke aard en wordt niet uitgevoerd. - DRIVING - de rit is begonnen en er is een actuele status bekend, maar de betreffende halte is nog niet bereikt - ARRIVED - de rit is aangekomen op de halte². - PASSED - de rit is de halte gepasseerd.

¹ Deze enumeratie lijkt op de TripStopStatus enumeratie van Koppelvlak 7/8, maar met 'CANCELLED' ipv CANCEL.

² Deze status wordt herhaald gestuurd zolang het voertuig op de halte staat.

Destination			- UNKNOWN - van het voertuig op de rit is geen actuele status bekend. - CANCELLED CALL – de ritpassage op de halte is vervallen maar de rit rijdt nog wel.
	transport_type	(1..n)	Modaliteit voor betreffende vertrek, bijvoorbeeld om een symbool te tonen. Waarde is op basis van de BISON enumeratie TransportType, met als mogelijke waarden: - BUS - TRAM - METRO - TRAIN - BOAT
	wheelchair_accessible	(1..n)	Geeft aan of het voertuig op de rit rolstoeltoegankelijk is (bijvoorbeeld om een symbool te tonen)
	is_timingstop	(1..n)	Geeft aan of het betreffende vertrek een tijdhalt voor de betreffende rit is.
	stop_code	(1..n)	VEROUDERD Quay van het betreffende vertrek. Van belang voor haltesystemen die geabonneerd zijn op meerdere quays en onderscheid willen maken.
	destinations	(1..n)	Lijst met bestemming- en detailbestemminggegevens van de rit.
	destination_name	(1..n)	Bestemming van de rit. Dit wordt door het distributiesysteem gevuld met de meest toepasselijke bestemming (qua lengte) voor het haltesysteem aan de hand van de aangeleverde maximum lengte in het Subscribe bericht. Of het wordt gevuld met een lijst met alle lengtes van de bestemming die bekend zijn in het distributiesysteem indien het haltesysteem heeft aangegeven zelf de best passende bestemming te kiezen. (SELF_DETERMINING).
	destination_detail	(0..n)	Idem voor de detail bestemmingen.
	language	(1..n)	Taal van de bestemming, met als mogelijke waarden: - NL (Nederlands) - FR (Frans) - EN (Engels) - DE (Duits) - FY (Fries) - LI (Limburgs)
	destination_length	(1..n)	Aantal karakters van de bestemmingsnaam.
	show_cancelled_trip	(0..1)	Geeft aan of een vervallen rit wel of niet getoond moet worden. Mogelijke waarden: - TRUE: de vervallen rit wordt wel getoond. - FALSE: de vervallen rit wordt niet getoond. In beide gevallen kan een verklarende vrije tekst

Met opmerkingen [Gv3]: Dit is anders dan voor De Lijn is geïmplementeerd. Daar heet de enumeratie destination_language, maar de enum wordt ook gebruikt in de Subscribe en daar heeft het een veel bredere scope.

		getoond worden.
block_code	(0..1)	Geeft de omloop waarbinnen de rit wordt uitgevoerd, zoals opgenomen in de planning van de betreffende rit. Wordt o.a. gebruikt door een haltesysteem met Vecom detectie.
occupancy	(0..1)	Actuele bezettingsgraad van het voertuig (0 – 100%).
line_public_number	(1..n)	Lijnnummer of andere identificatie van de lijn zoals bij de klant bekend (bijvoorbeeld 9K, 251, 12 of N1)
side_code	(1..n)	Perronaanduiding (bijvoorbeeld A, G, etc)
line_direction	(1..n)	Numerieke waarde die de richting van de betreffende rit aangeeft. Dit kan door het haltesysteem worden gebruikt om een bepaalde lijn/richting combinatie maar een beperkt aantal keer te tonen.
line_color	(1..n)	RGB waarde van de achtergrond van het lijnnummerveld.
line_text_color	(1..n)	RGB waarde van het lijnnummer (de tekst)
line_icon	(0..n)	Lijnicoon indien gebruikt. Bevat een referentie naar een icoon dat aanwezig is in het haltesysteem.
destination_color	(1..n)	Bestemmingskleur indien bekend (als achtergrond indien gecombineerd met tekst)
destination_text_color	(1..n)	Bestemmingskleur indien bekend (als voorgrond indien gecombineerd met tekst)
destination_icon	(1..n)	Bestemmingsicoon om te tonen indien bekend. Bevat, een referentie naar een icoon dat aanwezig is in het haltesysteem.
generated_timestamp	(1..n)	Het tijdstip waarop de reisinformatie over de betreffende rit gegenereerd is. Kan gebruikt worden om te verifiëren dat opeenvolgende berichten in de juiste volgorde worden verwerkt.
journey_number	(1..n)	Ritnummer zoals bekend bij de vervoerder.
line_border_color	(0..1)	Relevant voor haltesystemen die een randje rond de lijnachtergrond moeten tonen (De Lijn)
line_text_border_color	(0..1)	Relevant voor haltesystemen die een randje rond het lijnnummer moeten tonen (De Lijn)

general_messages			Bevat een lijst met vrije teksten (GeneralMessage)
GeneralMessage	message_hash	(1..n)	(Zie ook "Standaard berichtafhandeling") Voor elke vrije tekst/bericht wordt een unieke hash berekend dat het haltesysteem kan gebruiken om het bericht te identificeren. Dit veld wordt altijd gevuld.
	message_contents	(0..n)	Lijst met vrije teksten indien bekend.
	MessageText	message_text	Inhoud van de vrijetekst
		message_language	De taal van de vrijetekst (zie Destination bericht)
	message_start_time	(0..n)	Begintijd vanaf wanneer de vrije tekst getoond moet worden (dit is een timestamp, zie uitleg in bericht 'TravelInfo')
	message_end_time	(0..n)	Eindtijd tot wanneer de vrije tekst getoond moet worden (dit is een timestamp, zie uitleg target_arrival_time). Indien geen eindtijd wordt gespecificeerd moet het bericht getoond worden tot het wordt verwijderd.
	show_overview_display	(0..1)	Geeft aan of de vrijetekst ook getoond moet worden op een overzichtdisplay. Mogelijke waarden: TRUE: de tekst moet ook op een overzichtdisplay worden getoond. FALSE: de tekst moet niet op een overzichtdisplay getoond worden. ONLY: de tekst moet alleen op een overzichtdisplay getoond worden en niet op een haltedisplay.
	message_titles	(0..n)	Lijst met titels ¹ van de vrije teksten.
	MessageText	message_text	Inhoud van de titel
		message_language	De taal van de titel (zie Destination bericht)
	message_priority	(0..1)	Geeft een indeling naar de inhoud van de tekst. Het geeft daarmee het belang van de tekst aan. Mogelijke waarden: CALAMITY: tekst is gerelateerd aan een acute en ongeplande verstoring waar veel reizigers hinder van ondervinden. PTPROCESS: mededelingen als gevolg van veelvoorkomende afwijkende situaties, zoals omleidingen, vertragingen enz. COMMERCIAL: Aankondigingen van wijzigingen in de toekomst, zoals nieuwe dienstregelingen, enz.

¹ Afhankelijk van de mogelijkheden van het display, wordt de titel apart (en groter) getoond of wordt de titel opgenomen in de tekst.

			MISC: Overige mededelingen.
	general_messages_removes		Bevat een lijst met vrije teksten die verwijderd moeten worden (GeneralMessageRemove)
GeneralMessageRemove	message_hash	(1..n)	De ID van de vrije tekst die verwijderd moet worden.
	passing_time_removes		Bevat een lijst met passeertijden die verwijderd moeten worden. (PassingTimeRemove)
PassingTimeRemove	pass_time_hash	(1..n)	De ID van de passeertijd die verwijderd moet worden.

Indien bij de subscription bij destination_determination is aangegeven dat er gebruik gemaakt wordt van SELF_DETERMINING dan wordt de destinations als volgt gevuld per ritregel.
destination_name = [DestinationName50, DestinationName30, DestinationName24, DestinationName19, DestinationName16]
destination_detail = ["", "", DestinationDetail24, DestinationDetail19, DestinationDetail16]
De eerste 2 waarden van destination_detail zullen dus leeg zijn omdat deze gegevens niet via de huidige KV7/8 aangeleverd worden.

Berichten Distributiesysteem of Haltesysteem en Dashboardsysteem

SystemStatus

Dit bericht wordt generiek gebruikt door zowel het distributiesysteem als de haltesystemen om te communiceren naar een dashboardsysteem over de status van het systeem en de onderdelen daarvan. De status kan daarbij uitgedrukt worden in een van twee types: meetwaardes (Metric) of logberichten (LogMessage). Indien de informatie betrekking heeft op een haltesysteem, wordt dit in het client id duidelijk.

	Veld	Aantal	Uitleg
--	------	--------	--------

SystemStatus	metrics		(0..n)	Actuele meetwaarden die gecommuniceerd moeten worden. Meetwaarden worden met regelmatige intervallen gepubliceerd door het metende systeem.
	Metric	Client_id	(1)	Client Id waarmee systeem is aangemeld
		component	(1)	Naam van het betreffende component zoals gedefinieerd door de opdrachtgever (bijvoorbeeld "Behuizing" of "Display" voor haltesystemen of "Planning" voor een distributiesysteem). Toegestane tekens zijn letters en eventueel een lage streep (A-Z, a-z en _)
		component_index	(0..1)	Identificatie van het betreffende component in het systeem voor het geval er meerdere dezelfde componenten zijn. Optioneel, indien er geen tweede component is kan dit leeg worden gelaten.
		property	(1)	Eigenschap van het betreffende component dat wordt gemeten (bijvoorbeeld "deur", "temperatuur", "signaalsterkte" voor een haltesysteem of "ritten" of "berichten" voor een distributiesysteem). Toegestane tekens zijn letters en eventueel een lage streep (A-Z, a-z en _). Voor een betreffende combinatie 'component' en 'property' moet de eenheid waarin de waarde gemeten wordt, altijd hetzelfde zijn.
		value_type	(1)	Geeft aan van welk technisch type de waarde is (string, integer, double, bool, byte). Hiermee kan het ontvangende dashboardsysteem het juiste veld uitlezen.
		string_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde in tekst wordt gegeven.
		int_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde numeriek is.
		double_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde double is.
		bool_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde boolean is.
		bytes_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde in bytes is.
		unit	(1)	De eenheid van de waarde. Het mag geen newlines/enters bevatten, verder zijn er geen beperkingen (geldige UTF-8 content).
		timestamp_begin	(1)	Het begintijdstip van de periode waarover de waarde is gemeten. (dit is een timestamp, zie uitleg in bericht 'TravelInfo')
		timestamp_end	(1)	Het eindtijdstip van de periode waarover de waarde is gemeten. Indien de meetwaarde een momentane waarde is, blijft dit veld leeg (dit is een timestamp, zie uitleg in bericht 'TravelInfo')
	logs		(0..n)	Eventuele logberichten die gecommuniceerd moeten worden. (LogMessage)
	└	client_id	(1)	Client Id waarmee systeem is aangemeld

	type	(1)	Een van de volgende waarden van de enumeratie 'StatusType': - <i>ERROR</i>: de boodschap betreft de statusovergang naar een foutsituatie - <i>WARNING</i>: de boodschap betreft de statusovergang naar een waarschuwing - <i>OK</i>: de boodschap betreft statusovergang van een <i>ERROR</i> of <i>WARNING</i> terug naar een geaccepteerde waarde. In dit geval wordt de duur van de storing of waarschuwing in seconden meegegeven. - <i>LOG</i>: de boodschap betreft een andere, meer algemene, melding die geen statusovergang betreft.
	code	(1)	Een numerieke code die het type gebeurtenis aangeeft waarover een logbericht wordt gecommuniceerd.
	message	(1)	De tekstuele boodschap behorende bij de geregistreerde gebeurtenis, deze wordt in een dashboard aan de gebruiker gepresenteerd. Het bericht mag geen newlines/enters bevatten, verder zijn er geen beperkingen (geldige UTF-8 content).
	duration	(0..1)	De duur van een storing (<i>ERROR</i>) of waarschuwing (<i>WARNING</i>) in seconden. Deze waarde wordt alleen gevuld bij een boodschap van het type <i>OK</i> .
	timestamp	(0..1)	Het tijdstip waarop het bericht gegenereerd is (dit is een timestamp, zie uitleg in bericht 'TravelInfo')

CDDInfoRequest

Met dit bericht kan een dashboardsysteem bij het distributiesysteem een lijst opvragen van alle haltesystemen die zich hebben aangemeld bij het distributiesysteem.
De ownercode wordt meegestuurd omdat een dashboard meestal alleen de "eigen" systemen opvraagt.

	Veld	Aantal	Uitleg
CDDInfoRequest	client_id	(1)	Client Id waarmee het (dashboard)systeem is aangemeld.
	subscriber_owner_code	(0..1)	Als ingevuld stuurt het distributiesysteem alleen de id's van de haltesystemen van de leverancier. Als de leverancier niet is gespecificeerd of nul is, stuurt het distributiesysteem de ID's van alle aangemelde haltesystemen.

NB. Voor een volledig overzicht moet een dashboard een CDDInfoRequest sturen naar beide clusters en de informatie die het terugkrijgt combineren.

StopSystems

Op verzoek van een dashboardsysteem stuurt het distributiesysteem, met dit bericht, een lijst van alle haltesystemen die zich (ooit) bij het distributiesysteem hebben aangemeld en nooit permanent hebben afgemeld. In de lijst wordt aangegeven of het haltesysteem actief of inactief is. Een haltesysteem is actief voor het distributiesysteem als het zich heeft aangemeld, verbonden is en reisinformatie ontvangt.

	Veld	Aantal	Uitleg
StopSystems	client_id	(1)	Client Id waarmee het distributiesysteem is aangemeld.
	subscribed_systems	(0..n)	
	SubscriberSystem		
	subscriber_owner_code	(1)	De "owner" van het haltesysteem (meestal de leverancier) Zie het hoofdstuk Client Id
	serial_number	(1)	Uniek volgnummer van het haltesysteem binnen het domein van de "owner". Voor een haltesysteem meestal gelijk aan het serienummer van de hardware.
	active	(1)	Geeft aan of het haltesysteem is aangemeld op het distributiesysteem of niet.
	stop_code	(1..n)	De quaycodes waarvan het haltesysteem informatie ontvangt ¹ .

Het distributiesysteem bestaat uit twee clusters en een haltesysteem is altijd maar actief op één van de clusters. Als een haltesysteem inactief is op beide clusters betekent dat, dat het haltesysteem wel bestaat en bekend is maar dat het op dat moment niet is aangemeld op één van de clusters en geen reisinformatie ontvangt.

Als een haltesysteem zich permanent heeft afgemeld (is_permanent = true) bestaat het systeem niet meer in het distributiesysteem en komt het niet voor in de lijst.

In de lijst staan haltesystemen onafhankelijk van de versie van het koppelvlak waarmee ze communiceren.

¹ Als het haltesysteem zich aangemeld heeft met een stopplacecode, worden hier de onderliggende quaycodes gegeven.

Berichten tussen Halte- en Dashboardsysteem

Begrippen en regels

Bij de berichtenstroom tussen halte- en dashboardsystemen gelden de volgende begrippen en regels:

- Een haltesysteem (in de praktijk ook vaak display genoemd) bestaat uit componenten. Voorbeeld: behuizing, scherm, software, ...
- Een component heeft eigenschappen. Voorbeeld: een behuizing heeft een deur, een scherm heeft een aantal defecte LED's, ...
- Eigenschappen hebben een unieke code¹.
- Eigenschappen hebben een waarde. Voorbeeld: een deur is open, het aantal defecte LED's is 4.
- Sommige eigenschappen kennen conditie-overgangen. Deze treden op als de waarde een grens overschreidt. Voorbeeld: een deur gaat van dicht naar open, het aantal defecte LED's gaat van 4 naar 5, waarbij 5 als grens is gedefinieerd.
- De waarden van alle eigenschappen tezamen vormen de status van het haltesysteem.

Bij de aanschaf van een haltesysteem wordt overeengekomen uit welke componenten een haltesysteem bestaat en welke eigenschappen die componenten hebben. Ook wordt van elke conditie-overgang het statustype bepaald (ERROR, WARNING of OK)

Het haltesysteem is verantwoordelijk voor het publiceren van de overeengekomen conditie-overgangen naar de dashboards zodat gebruikers de status van de haltesystemen kunnen volgen, zonder dat er regelmatig statusinformatie moet worden opgevraagd. Indien een eigenschap van een component van een haltesysteem een conditie-overgang ondergaat, wordt een 'SystemStatus' bericht op het topic van het haltesysteem gepubliceerd. Indien een dashboardsysteem meer informatie nodig heeft kan de gehele status van een haltesysteem worden opgevraagd. Het is binnen het koppelvlak niet mogelijk om de grenswaarden voor conditie-overgangen in te stellen. Dit wordt verondersteld onderdeel uit te maken van een configuratiebestand op het haltesysteem.

- Het haltesysteem **kan** een logboodschap versturen als de waarde van een eigenschap verandert.
- Het haltesysteem verstuurt **altijd** een logboodschap bij een overeengekomen conditie-overgang.
- Logboodschappen bij conditie-overgangen hebben een type ERROR, WARNING of OK.
- Logboodschappen van het type ERROR of WARNING, van een eigenschap (=code) worden **altijd** gevolgd door een logboodschap van het type OK van diezelfde eigenschap. Voorbeeld: deur open (WARNING) moet altijd gevolgd worden door een deur dicht (OK).
- Bij elke logboodschap van het type OK berekent het haltesysteem de duur van de storing of waarschuwing en stuurt dat mee met de logboodschap.

Informatieverzoeken

Vanuit de dashboardsystemen kunnen vier verschillende informatieverzoeken naar een haltesysteem worden verstuurd. Het haltesysteem reageert hierop met een antwoord. De vier verzoeken zijn voor het ophalen van:

- De gegevens over het systeem:
gedefinieerd in naam/waarde combinaties. Bijvoorbeeld: schermtype = LED, of aantal regels = 4.
- De actuele status:
lijst met alle eigenschappen waarvoor een conditie-overgang en een logboodschap is gedefinieerd, met de actuele waarde van de eigenschap plus het type van de laatste logboodschap.

¹ Deze codes worden beheerd door OV-data en zijn uniek over de DRIS implementaties heen.

- De gedetailleerde actuele status:
Een lijst met statusgegevens die de displayleverancier zelf kan definiëren en opvragen.
- De reisinformatie in het haltesysteem:
informatie over actuele en toekomstige ritten, alsmede alle actuele en toekomstige vrije teksten.
- De tekst die op het scherm staat:
informatie die getoond wordt op het scherm of de schermen (snapshot).

Bestandsuitwisseling

Het centrale distributiesysteem heeft een FTP server beschikbaar waarmee bestanden kunnen worden uitgewisseld tussen dashboard- en haltesystemen. Een dashboardsysteem kan een configuratiebestand klaarzetten op de FTP server en stuurt vervolgens een bericht naar één of meer haltesystemen dat het bestand kan worden opgehaald.

Andersom kan een haltesysteem (regelmatig) een tracebestand klaarzetten op de FTP server en een bericht sturen naar een dashboardsysteem. Deze mogelijkheid om tracebestanden te maken in het haltesysteem en deze naar een dashboardsysteem te versturen is alleen bedoeld voor incidentele analyse. Het is uitdrukkelijk niet de bedoeling dat meerdere haltesystemen gedurende langere tijd gebruik maken van deze faciliteit.

De FTP servers tussen de beide clusters worden niet gesynchroniseerd. Een haltesysteem kan op enig moment communiceren met één van beide clusters en het is de verantwoordelijkheid van de dashboardsystemen om de configuratiebestanden op beide FTP servers te plaatsen. Evenzo zet een haltesysteem een tracebestand neer op de cluster waarmee het op dat moment communiceert en het is de verantwoordelijkheid van de dashboardsystemen om te onderzoeken op welke cluster een tracebestand staat.

InfoRequest

	Veld	Aantal	Uitleg
InfoRequest	client_id	(1)	Client Id waarmee het (dashboard)systeem is aangemeld.
	request_type	(1)	Een van de volgende waardes van de enumeratie 'RequestType': - <i>SYSTEM_INFO</i>: het dashboard vraagt de algemene informatie van het haltesysteem op. - <i>STATUS</i>: het dashboard vraagt de actuele waarde van de eigenschappen op. - <i>TRAVEL_INFO</i>: het dashboard vraagt de informatie op van actuele en toekomstige ritten en vrije teksten. - <i>SCREEN_CONTENT</i>: <i>het commando vraagt wat getoond wordt op het scherm of op de schermen.</i> - <i>DETAIL_STATUS</i>: het dashboard vraagt hiermee een lijst op van leverancierspecifieke eigenschappen plus de actuele waarde.
	request_parameter	(0..1)	Parameter die de opgevraagde informatie nader definieert. De betekenis is afhankelijk van het request_type. - Bij het opvragen van het schermbeeld (<i>SCREEN_CONTENT</i>) geeft de parameter het aantal seconden dat de updates van het schermbeeld verstuurd moeten worden. Als de parameter niet is gespecificeerd of nul is, stuurt het haltesysteem het schermbeeld eenmalig. - Bij de andere request_types wordt de parameter in het haltesysteem genegeerd.

SystemInfo

Met dit bericht stuurt een haltesysteem de lijst met overeengekomen¹ informatie-elementen in de vorm van naam/waarde paren. Het bericht wordt verstuurd in antwoord op een verzoek van een dashboardsysteem, en na het aanmelden van het haltesysteem op het distributiesysteem.

SystemInfo	client_id		(1)	Client Id waarmee het haltesysteem is aangemeld
	system_info_lines		(1...n)	Lijst met met Informatieregels over het haltesysteem. Elke regel bestaat uit een naam plus een waarde.
	SystemInfoLine	Name	(1)	De, niet gestandaardiseerde, naam van het informatie-element.
		value_type	(1)	Geeft aan van welk technisch type de waarde is (string, integer, double, bool, byte). Hiermee kan het ontvangende dashboardsysteem het juiste veld uitlezen.
		string_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde in tekst wordt gegeven.
		int_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde numeriek is.
		double_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde double is.
		bool_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde boolean is.
		bytes_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde in bytes is.

¹ Bij de aanschaf van het haltesysteem wordt door de opdrachtgever een lijst met gegevens opgegeven.

StatusOverview

Met dit bericht stuurt een haltesysteem een lijst met actuele waardes van alle overeengekomen eigenschappen van het haltesysteem, plus het type van de laatste Logboodschap die is verstuurd (LogMessage). Tesamen vormen deze waardes een korte actuele status van het haltesysteem, bedoeld voor eerstelijns onderhoud, met name door DOVA¹.

	Veld	Aantal	Uitleg
StatusOverview	client_id	(1)	Client Id waarmee (halte)systeem is aangemeld
	status_overview_lines	(1...n)	Lijst met met statusregels met de indeling van StatusOverviewLine.

DetailStatusOverview

Met dit bericht stuurt een haltesysteem een lijst met actuele waardes van andere eigenschappen van het haltesysteem. De eigenschappen kunnen door de displayleverancier worden bepaald en de lijst zal over het algemeen veel gedetailleerder informatie over het haltesysteem geven, bijvoorbeeld voor de tweedelijns monteur.

	Veld	Aantal	Uitleg
DetailStatusOverview	client_id	(1)	Client Id waarmee (halte)systeem is aangemeld
	detail_status_overview_lines	(1...n)	Lijst met statusregels met de indeling van StatusOverviewLine.

¹ Deze lijst is met opzet beperkt gehouden en bevat geen project- of leverancierspecifieke gegevens. Besef dat DOVA het beheer doet van vele duizenden haltesystemen, van veel verschillende leveranciers. Door de lijst kort en gestandaardiseerd te houden heeft DOVA van alle systemen eenzelfde statusoverzicht.

StatusOverviewLine

Beide bovenstaande statusberichten hebben dezelfde structuur, zodat het vullen en lezen van de berichten met dezelfde code kan gebeuren. Wel zullen bepaalde velden in het ene statusbericht wel en in het andere niet gevuld worden.

	Veld	Aantal	Uitleg
StatusOverviewLine	component	(1)	De naam van een deel van een haltesysteem.
	component_index	(0..1)	Identificatie van het betreffende component in het systeem voor het geval er meerdere dezelfde componenten zijn. Optioneel, indien er geen tweede component is kan dit leeg worden gelaten. Dit veld zal in het standaard statusoverzicht meestal leeg gelaten worden, omdat het voor de eerstelijns niet van belang is.
	property	(1)	De naam van de componenteigenschap waarover gerapporteerd wordt.
	value_type	(1)	Geeft aan van welk technisch type de waarde is (string, integer, double, bool, byte). Hiermee kan het ontvangende dashboardsysteem het juiste veld uitlezen.
	string_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde in tekst wordt gegeven.
	int_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde integer is.
	double_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde double is.
	bool_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde boolean is.
	bytes_value	(0..1)	Dit veld wordt gevuld met de actuele waarde van de eigenschap als de waarde in bytes is.
	status	(0..1)	Het type van de laatst verstuurd logboodschap (ERROR, WARNING, OK, LOG). De enumeratie is beschreven bij de LogMessage. Dit veld zal meestal niet gevuld worden in het gedetailleerde statusoverzicht, omdat de statusgegevens die voorkomen in de logberichten in het gewone statusoverzicht moeten staan. ¹
	code	(1)	Numerieke code zoals in de lijst met logboodschappen ² .

¹ Het staat een displayleverancier vrij om het veld te gebruiken om bij bepaalde statusgegevens extra aandacht te vragen in het dashboard of statusovergangen te markeren (bijv. ERROR)

² Voor leverancierspecifieke statussen die via DetailStatusOverview worden verstuurd wordt altijd 1000 ingevuld.

TravelInfoResponse

Met dit bericht stuurt het haltesysteem alle vertrektijden en vrije teksten die het heeft.

	Veld	Aantal	Uitleg
TravelInfoResponse	client_id	(1)	Client Id waarmee het (halte)systeem is aangemeld.
	Travel_info_content	(1)	Dit bericht 'TravelInfo' is eerder gedocumenteerd. Het haltesysteem moet alle op dat moment bekende ritten en vrije teksten in het bericht opnemen en publiceren, zodat gecontroleerd kan worden of de juiste informatie opgeslagen is.

ScreenContentResponse

Met dit bericht stuurt het haltesysteem het schermbeeld inclusief eventuele op dat moment niet zichtbare inhoud. Bijvoorbeeld bij alterneren of als meerdere blokken vrije tekst worden getoond. Indien op het haltesysteem meerdere schermen zijn aangesloten die ieder het schermbeeld tonen, wordt het beeld 1 keer verstuurd. Indien op het haltesysteem meerdere schermen zijn aangesloten die ieder een deel van het schermbeeld tonen, moeten de schermbeelden worden samengevoegd tot 1 beeld dat verstuurd wordt¹.

Dit bericht moet eenmalig gestuurd worden als antwoord op een InfoRequest van het type SCREEN_CONTENT en vervolgens steeds als het schermbeeld verandert², voor de periode die in de request_parameter staat aangegeven.

	Veld	Aantal	Uitleg
ScreenContentResponse	client_id	(1)	Client Id waarmee het (halte)systeem is aangemeld.
	screen_contents	{0..n}	VEROUDERD
	screen_index	{0..1}	VEROUDERD Indien een haltesysteem meerdere schermen heeft die verschillende informatie tonen, is dit het volgnummer van het betreffende scherm. Indien het haltesysteem maar uit een scherm bestaat of uit meerdere schermen die dezelfde informatie tonen, mag dit veld leegblijven.
	content_type	(1)	Een van de volgende formaten die aangeven hoe het veld 'content' is gevuld: - TEXT : De tekst zoals te zien is op het display (op het moment van opname, inclusief het alternerende deel van de tekst). Wordt verstuurd met utf8 karaktercodering en in CSV formaat met tab als scheidingsteken ³ .

¹ Om de compatibiliteit met voorgaande versies te bewaren, staat het bericht nog toe om meerdere schermbeelden te sturen, maar alleen het eerste wordt verwerkt in het dashboardsysteem van DOVA.

² Let op dat alterneren, of het tonen van een ander vrije tekst blok geen verandering is die een nieuw bericht tot gevolg heeft, omdat de scherm inhoud wordt verstuurd inclusief de inhoud die door alterneren op dat moment niet zichtbaar is.

³ Elke kolom op het scherm buiten, is een veld in CSV met de opmerkingen in een apart veld. [Zie bijlage 1](#).

			- <i>IMAGE_PNG</i>: Een afbeelding in Portable Network Graphics (PNG) formaat (op het moment van opname, inclusief het alternerende deel van de tekst) - <i>IMAGE_GIF</i>: Een afbeelding in het Graphics Interchange Format (GIF) formaat. Hierbij is het mogelijk meerdere (bewegende) frames op te nemen om alternerende teksten weer te kunnen geven. De leverancier van het haltesysteem kiest aan de hand van de eisen van opdrachtgever en type scherm het meest geschikte formaat.
	content	(1)	Indien type = TEXT, de ruwe, geformateerde tekst. Indien type = IMAGE_PNG of IMAGE_GIF, base64 encoded data van de afbeelding.
	timestamp	(1)	Tijdstip waarop dit bericht is samengesteld door het haltesysteem (zie uitleg in bericht 'TravelInfo').

FileAvailable

Dashboardsystemen kunnen een configuratiebestand klaarzetten voor een haltesysteem. Op de centrale distributiecluster is daartoe een FTP server beschikbaar. Andersom kunnen haltesystemen tracebestanden op de FTP server klaarzetten voor dashboardsystemen. Het afnemende systeem kan geïnformeerd worden dat er een bestand beschikbaar is, met het hier beschreven bericht.

	Veld	Aantal	Uitleg
FileAvailable	client_id	(1)	Client Id waarmee het (dashboard of halte)systeem is aangemeld.
	file_name	(1)	Naam van het bestand dat klaar staat op de FTP server.

Berichten tussen Haltesystemen onderling

Voor de audiovoorziening op een station is het in uitzonderingsgevallen nodig dat de haltesystemen een bericht naar elkaar versturen¹. Met dit bericht kan een haltesysteem (meestal een overzichtdisplay) aan de andere haltesystemen binnen de stopplace doorgeven dat van een set ritten, bepaalde wijzigingen in de informatie moeten worden uitgesproken.

AudioTracking

	Veld	Aantal	Uitleg
AudioTracking	pass_time_hash	(1...n)	De ritnummers waarvan de wijzigingen moeten worden uitgesproken.
	duration	(1)	De tijd in minuten dat de wijzigingen moeten worden uitgesproken. (max. 15 minuten)

Omdat de Haltesystemen van een stopplace op enig moment verbonden kunnen zijn met verschillende clusters van het distributiesysteem, wordt dit bericht tussen de clusters uitgewisseld.

¹ De Haltesystemen communiceren onderling dus via de centrale MQTT broker.

MQTT

Algemeen

Om data te kunnen versturen en ontvangen, moeten alle systemen (Distributie-, Halte- en Dashboardsystemen) verbonden worden met de centrale MQTT broker, die werkt conform MQTT versie 5.

Verbinden

Bij het verbinden van de clientsystemen met de MQTT broker zijn de volgende instellingen van belang:

- Keep Alive
De Keep Alive periode moet instelbaar zijn in de clientsystemen met een defaultwaarde van 15 seconden voor Distributie- en Dashboardsystemen en 60 seconden voor Haltesystemen.
- Client id
Voor de verbinding met de broker is een naam nodig. De naam wordt samengesteld uit de drie velden van Client Id gescheiden door een '_' (Underscore): de volgorde is als volgt.
 - SubscriberOwnerCode
 - Type
 - Serialnumber

Een voorbeeld van een client id is: DOVA_1_1 of SURTRONIC_2_42741

Een client id dient uniek te zijn over alle systemen heen. Indien het client id wijzigt zal de **eerste aanmelding** opnieuw moeten worden doorlopen.

- Clean Start
True voor Distributie- en Haltesystemen en False voor Dashboardsystemen.
Omdat een distributiesysteem alleen informatie stuurt, is een sessie met geschiedenis niet relevant. Een Haltesysteem krijgt bij aanmelden (initieel en na verlies van de verbinding) alle informatie opnieuw opgestuurd. Een sessie met geschiedenis is daarom niet relevant. Een Dashboardsysteem moet echter altijd een complete historie bijhouden van de aangesloten Halte- en Distributiesystemen en moet daarom alle informatie die gestuurd is, ontvangen. Een sessie met (lange) geschiedenis is daarom van belang. Expiry Interval voor het Dashboard wordt op : 604800 Sec gezet (1Week)
- Last Will
Bij verlies van de verbinding tussen een clientsysteem en de Broker moet altijd een Last Will worden uitgezonden, zodat het verbindingsverlies bekend wordt in de Dashboardsystemen. Voor Distributiesystemen moet als Last Will op topic `unsubscribe/<version>/0/<SubscriberOwnerCode>/<Serialnumber>` een Unsubscribebericht geconfigureerd staan dat aangeeft dat de verbinding is verbroken. Voor Haltesystemen moet als Last Will op topic `unsubscribe/<version>/2/<SubscriberOwnerCode>/<Serialnumber>` een Unsubscribe bericht¹ geconfigureerd staan met als enige waarde "is_permanent" = false (dit bericht kan dus eenmalig tijdens initialisatie worden aangemaakt).

¹ Zoals beschreven in het hoofdstuk Flows kan dit bericht ook door het haltesysteem worden verstuurd als de software of het haltesysteem moet herstarten om welke reden dan ook.

Voor Dashboardsystemen moet op topic
`unsubscribe/<version>/1/<SubscriberOwnerCode>/<SerialNumber>` een Unsubscribebericht
geconfigureerd staan dat aangeeft dat de verbinding is verbroken.

Version

Om verschillende versies van het koppelvlak te kunnen ondersteunen bevat elk topic een versienummer. Het versienummer bestaat uit één getal en kent geen subgetallen. Dit koppelvlakdocument wordt geïmplementeerd met versie 4 in de topic's. Er wordt maar één getal gebruikt omdat er verondersteld wordt dat het altijd om een significante wijziging gaat bijvoorbeeld en dat kleinere wijzigingen geclusterd worden. Het versienummer in de topic's is niet gelijk aan het versienummer van de documenten die het koppelvlak beschrijven, zodat daar wel versienummers met subgetallen gebruikt kunnen worden. Het heeft organisatorische voordelen om het versienummer van de documentatie los te koppelen van het versienummer in de software. Op die manier kun je in de documentatie meerdere versies gebruiken zonder dat dit geïmplementeerd hoeft te worden (tekstuele wijzigingen). Ook kun je wijzigingen opsparen en goed administreren en op enig moment beslissen om de collectie aan wijzigingen te implementeren in de software.

Als er een nieuwe versie van het koppelvlak wordt geïntroduceerd, wordt dit eerst in het distributiesysteem geïmplementeerd. Het distributiesysteem abonneert zich op de relevante topic's met het nieuwe versienummer naast de bestaande subscriptions op de topic's met het oude versienummer. Vanaf dat moment kunnen de haltesystemen stuk voor stuk omgebouwd worden. Een omgebouwd haltesysteem gaat dan op de "nieuwe" topic's communiceren. Zodra alle haltesystemen op de nieuwe topic's luisteren en publiceren, hoeft het distributiesysteem niet meer te luisteren en te publiceren op de topic's met het oude versienummer, en kunnen die topic's verwijderd worden.

Bij een migratie is extra aandacht nodig voor de dashboardsystemen. Het centrale DOVA-dashboard zal gelijk met het distributiesysteem de nieuwe topic's gaan gebruiken, maar dashboards van externe partijen hoeven pas te gaan luisteren op de topic's met het nieuwe versienummer zodra het eerste haltesysteem wordt omgebouwd dat wordt gevolgd in het dashboard.

Migratie naar een nieuwe koppelvlakversie is een complex project met risico's op verstoring in de informatievoorziening. Daarom wordt een nieuwe versie alleen geïntroduceerd als het echt niet anders kan en mogen er nooit meer dan twee versies tegelijk actief zijn in het hele systeem.

Topics

Per type bericht topic wordt exact één topic gedefinieerd zodat halte-, distributie en dashboardsystemen zich kunnen abonneren op bepaalde type berichten.

subscribe/<Version>/<Type>/<SubscriberOwnerCode>/<SerialNumber>

Publicatie: Individueel Haltesysteem

Abonnee: Distributiesysteem, Dashboardsystemen

Bericht: Subscribe

QoS: Exactly once (2)

Retain: false

Type, SubscriberOwnerCode, SerialNumber: van het Haltesysteem dat zich aanmeldt.

In het aanmeldproces wordt dit topic gebruikt door haltesystemen om aanmeldinformatie op te publiceren. Hierdoor weten het distributiesysteem en de dashboardsystemen welke haltesystemen op dat moment actief zijn en op welke topics gepubliceerd moet worden om te communiceren met het haltesysteem. Ook krijgt het distributiesysteem eventuele stuurparameters zoals bijvoorbeeld filters, meegestuurd.

subscription_response/<Version>/<Type>/<SubscriberOwnerCode>/<SerialNumber>

Publicatie: Distributiesysteem

Abonnee: Individueel Haltesysteem, Dashboardsystemen

Bericht: SubscriptionResponse

QoS: Exactly once (2)

Retain: false

Type, SubscriberOwnerCode, SerialNumber: van het Haltesysteem dat zich heeft aangemeld.

Resultaten en acties met betrekking tot de subscription worden op dit topic gepubliceerd door het distributiesysteem. Haltesystemen moeten op dit topic abonneren om de status te ontvangen van een aanmelding en eventuele vervolg acties uit te voeren. Er wordt voor elk aanmeldbericht op

subscribe/<Version>/<Type>/<SubscriberOwnerCode>/<SerialNumber> een antwoord verstuurd op dit topic. De dashboardsystemen kunnen zich op dit topic abonneren, zodat ze het subscriptionproces kunnen volgen en vastleggen.

publicname/<Version>/<Type>/<SubscriberOwnerCode>/<SerialNumber>

Publicatie: Distributiesysteem

Abonnee: Individueel Haltesysteem, Dashboardsystemen

Bericht: PublicName

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, SerialNumber: van het Haltesysteem.

Dit bericht wordt verstuurd na authenticatie en elke keer dat de gegevens van Quay of stopplacecodes veranderen in het CHB. Haltesystemen moeten op dit topic abonneren om de informatie over de Quays en Stopplaces te ontvangen. De dashboardsystemen kunnen zich op dit topic abonneren, zodat ze de wijzigingen kunnen volgen en vastleggen.

unsubscribe/<Version>/<Type>/<SubscriberOwnerCode>/<SerialNumber>

Publicatie: Individueel Haltesysteem

Abonnee: Distributiesysteem, Dashboardsystemen en Individueel Haltesysteem

Bericht: Unsubscribe

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van een Haltesysteem dat zich eerder heeft aangemeld. Dit topic wordt gebruikt voor zowel het last-will bericht als permanente afmeldberichten.

unsubscribe/4/0/ Kv8DistributorService/#

Publicatie: Distributiesysteem

Abonnee: Halte- en Dashboardsystemen

Bericht: Unsubscribe

QoS: Atleast once (1)

Retain: false

travelinfo/<Version>/<Type>/<SubscriberOwnerCode>/<Serialnumber>

Publicatie: Distributiesysteem

Abonnee: Individueel Haltesysteem

Bericht: TravelInfo

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van het Haltesysteem dat de informatie moet tonen. Reisinformatieberichten zoals hierboven omschreven worden op dit topic gepubliceerd. Dit kunnen zowel passeertijden als vrije teksten zijn. Passeertijden kunnen op korte termijn (realtime) of langere termijn (planning) verstuurd worden.

De QoS op dit topic is bewust op 1 gehouden omdat de reisinformatieberichten op twee manieren van elkaar te onderscheiden zijn (door middel van het veld 'pass_time_hash' en het optioneel te ontvangen veld 'generated_timestamp'). Hierdoor is er dus geen nadeel als hetzelfde bericht meerdere keren wordt ontvangen door een haltesysteem. Dit heeft performancevoordelen.

systemstatus/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: Individueel Haltesysteem, Distributiesysteem, individueel Dashboardsysteem

Abonnee: Dashboardsystemen (autorisatie wordt afgedwongen op basis van SubscriberOwnerCode)

Bericht: SystemStatus

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van het systeem dat de status publiceert. Op dit topic publiceren alle soorten systemen logboodschappen en meetwaarden. Met de logboodschappen kan in een dashboardsysteem de status van een systeem gevolgd worden en de meetwaarden geven (statistische) informatie over de werking van het systeem.

cddinforequest/<Version>

Publicatie: individueel dashboardsysteem

Abonnee: individueel distributiesysteem.

Bericht: CDDInfoRequest

QoS: Atleast once (1)

Retain: false

Op dit topic publiceren de dashboardsystemen hun verzoek om informatie van het distributiesysteem (beide clusters).

stopsystems/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel distributiesysteem

Abonnee: dashboardsysteem

Bericht: StopSystems

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van het vragende Dashboardsysteem.

Op dit topic publiceert een distributiesysteem de lijst met bekende haltesystemen in antwoord op het verzoek van een dashboardsysteem.

inforequest/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel Dashboardsysteem

Abonnee: individueel haltesysteem.

Bericht: InfoRequest

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van het Haltesysteem dat moet antwoorden.

Op dit topic publiceren de dashboardsystemen hun verzoek om informatie van een haltesysteem.

systeminfo/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel haltesysteem

Abonnee: het vragende dashboardsysteem

Bericht: SystemInfo

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van het vragende Dashboardsysteem.

Op dit topic publiceert een haltesysteem de systeem informatie in antwoord op het verzoek van een dashboardsysteem.

statusoverview/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel haltesysteem

Abonnee: het vragende dashboardsysteem

Bericht: StatusOverview

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van het vragende Dashboardsysteem.

Op dit topic publiceert een haltesysteem het statusoverzicht in antwoord op het verzoek van een dashboardsysteem.

detailstatusoverview/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel haltesysteem

Abonnee: het vragende dashboardsysteem

Bericht: DetailStatusOverview

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, Serialnumber: van het vragende Dashboardsysteem.

Op dit topic publiceert een haltesysteem het gedetailleerde statusoverzicht in antwoord op het verzoek van een dashboardsysteem.

travelinforesponse/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel haltesysteem

Abonnee: het vragende dashboardsysteem

Bericht: TravelInfoResponse

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, SerialNumber: van het vragende Dashboardsysteem.

Op dit topic publiceert een haltesysteem de informatie over alle ritten en vrije teksten die het heeft, in antwoord op het verzoek van een dashboardsysteem.

screencontent/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel haltesysteem

Abonnee: het vragende dashboardsysteem

Bericht: ScreenContentResponse

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, SerialNumber: van het vragende Dashboardsysteem.

Op dit topic publiceert een haltesysteem de informatie die op het scherm (de schermen) wordt getoond, in antwoord op het verzoek van een dashboardsysteem.

fileavailable/<Version>/<Type>/<SubscriberOwnerCode>/<serialnumber>

Publicatie: individueel halte- of dashboardsysteem

Abonnee: individueel dashboardsysteem of haltesysteem

Bericht: FileAvailable

QoS: Atleast once (1)

Retain: false

Type, SubscriberOwnerCode, SerialNumber: van het systeem waarvoor het bestand bedoeld is.

Op dit topic geeft een dashboardsysteem aan dat er een configbestand klaar staat voor één of meer haltesystemen. Evenzo geeft een haltesysteem op dit topic aan dat er een tracebestand klaar staat voor één of meer dashboardsystemen.

audiotracking/<Version>/<Stopplace>

Publicatie: individueel haltesysteem waarop een reiziger audio-informatie heeft opgevraagd.

Abonnee: individueel haltesysteem binnen de stopplace van de zender.

Bericht: AudioTracking

QoS: Atleast once (1)

Retain: false

Op dit topic geeft een Haltesysteem aan dat een reiziger audio-informatie heeft opgevraagd waarvan de wijzigingen ook op andere Haltesystemen moeten worden uitgesproken.

Stopplace: de code van de stopplace waartoe het publicerende haltesysteem behoort.

Een haltesysteem moet zich voor de reisinformatie op vier topics abonneren:

- *travel_information/<Version>/2/<SubscriberOwnerCode>/<SerialNumber>*. Het haltesysteem abonneert zich op dit topic met het eigen Type, SubscriberOwnerCode en SerialNumber, zodat het alleen de "eigen" reisinformatie ontvangt.
- *subscription_response/<Version>/2/<SubscriberOwnerCode>/<SerialNumber>*. Het haltesysteem abonneert zich op dit topic met het eigen Type, SubscriberOwnerCode en SerialNumber, zodat het alleen de antwoorden ontvangt op de "eigen" aanmeldingen.
- *unsubscribe/<Version>/2/<SubscriberOwnerCode>/<SerialNumber>*: Het haltesysteem abonneert zich op dit topic met het eigen Type, SubscriberOwnerCode en SerialNumber, zodat het alleen de "eigen" afmeldingen (last will) ontvangt.

- *unsubscribe/<Version>/0/<SubscriberOwnerCode>/<Serialnumber>*: Het haltesysteem abonneert zich op dit topic, waarbij het client id het id is van het distributiesysteem waar het op is aangemeld.

Voor beheerdoeleinden moet een haltesysteem zich verplicht abonneren op de volgende twee topics:

- *inforequest/<Version>/2/<SubscriberOwnerCode>/<serialnumber>* Het haltesysteem abonneert zich op dit topic met het eigen Type, SubscriberOwnerCode en Serialnumber, zodat het alleen de “eigen” informatieverzoeken ontvangt.
- *fileavailable/<Version>/2/<SubscriberOwnerCode>/<serialnumber>* Het haltesysteem abonneert zich op dit topic met het eigen Type, SubscriberOwnerCode en Serialnumber, zodat het alleen de “eigen” configuratiebestanden ontvangt.

Versiebeheer

Dit document en het Protobuf bestand horen bij elkaar en definiëren samen het koppelvak en het gebruik ervan. Het versienummer bestaat volgens de normale conventie uit drie cijfers. X.Y.Z. Het eerste getal verandert bij een niet backwards compatibele wijziging. Dus als er berichten of velden vervallen, of een ander nummer krijgen. Het tweede cijfer wijzigt als er een backwards compatibele wijziging plaatsvindt, dit omvat ook toevoeging van berichten of velden. In beide gevallen verandert het Protobufbestand. Als er alleen tekstuele wijzigingen plaatsvinden in de koppelvakbeschrijving zonder dat het Protobufbestand wijzigt, wordt het derde cijfer opgehoogd. De versie van het Protobufbestand bestaat alleen uit de eerste twee cijfers.

versie	datum	wijziging	beschrijving	Protobuf
3.4	14 juni 21	Veld toegevoegd in InfoRequest bericht, waarmee het aantal seconden kan worden opgegeven dat een haltesysteem de wijzigingen van het schermbeeld moet sturen.	✓	✓
		De naam van het PassingTime deel van het bericht is enkelvoud gemaakt (PassingTime ipv PassingTimes), om het consistent te maken met de rest van de berichten.	✓	✓
		Topicnamen systeminfo en statusoverview zonder hoofdletter om het consistent te maken met de andere topicnamen.	✓	
		De vrijeteksttype BLANC is vervallen.	✓	✓
		Het type van het contentdeel van het ScreenContentResponse bericht is veranderd van string naar bytes.		✓
		Hoofdstuk over granulariteit toegevoegd	✓	
3.5	6 aug 21	Het vrijeteksttype OVERRULE is vervallen, waardoor het veld Generalmessage_type geen functie meer heeft. Veld is verwijderd.	✓	✓
4.0	1 maart 22	Bericht AudioTracking toegevoegd om communicatie tussen Haltesystemen binnen een STOPPLACE mogelijk te maken. Inclusief het benodigde Topic. Omdat het haltesysteem moet weten tot welke STOPPLACE het behoort, wordt dit door het distributiesysteem teruggestuurd bij het aanmelden.	✓	✓
		Het PublicNames bericht is uit de Container gehaald naar een apart bericht. Restrictie toegevoegd dat alle stop_codes waarvoor een Haltesysteem zich aanmeldt moeten behoren tot 1 STOPPLACE.	✓	✓
		De enumeratie DATA_CHANGED uit het SubscriptionResponse bericht gehaald. Als het distributiesysteem constateert dat de CHB-gegevens van de halte zijn gewijzigd stuurt het een Unsubscribe bericht. Het haltesysteem meldt zich dan opnieuw aan en ontvangt de gewijzigde haltegegevens.	✓	✓
		Bericht toegevoegd waarmee een dashboard bij het distributiesysteem per leverancier een lijst op kan vragen van alle bekende haltesystemen met de status en tijdstip van laatste wijziging. Ook bericht met de lijst toegevoegd. Incl. topics.	✓	✓
4.1	1 april 22	Veld toegevoegd aan het Containerbericht (flush) dat aangeeft dat de Container een hele nieuwe set aan ritpassages bevat. Het haltesysteem moet dan de oude ritpassages verwijderen.	✓	✓

versie	datum	wijziging	beschrijving	Protobuf
4.2	22 april 22	ShowCancelledTrip van enumeratie naar bool in proto en tekst aangepast in beschrijving.	✓	✓
		In enkele topics de QoS gewijzigd van 2 naar 1.	✓	
		Bericht toegevoegd in de container waarmee eerder verstuurde passeertijden ingetrokken kunnen worden, waarmee het flush veld weer overbodig is geworden.	✓	✓
		Korte toelichting hoe het versienummer in de topic's gebruikt wordt bij migratie naar een nieuwe koppelvakversie.	✓	
		Beschrijving van Last Will bericht iets verduidelijkt.	✓	
4.3	16 mei 22	De berichtnaam Container en het topic travel_information veranderd naar TravelInfo. Het bericht en topic TravelInfo veranderd naar TravelInfoResponse.	✓	✓
		Het veld generated_timestamp verwijderd uit het bericht GeneralMessageRemove.	✓	✓
		Het veld component_index toegevoegd aan bericht StatusOverview.	✓	✓
		In bericht Metric andere typen waarden (dan int) toegevoegd. Gelijk gemaakt aan bericht SystemInfoLine en StatusOverviewLine	✓	✓
4.4	1 juli 22	In de figuren stond nog "het bericht Container". Dit is veranderd naar TravelInfo	✓	
		In het TravelInfo bericht bij veld trip_stop_status, zin over ritgebonden tekst verwijderd.	✓	
		Bijlage toegevoegd met mapping van treininfo naar TravelInfo bericht.	✓	
		Bij de topics toegevoegd dat antwoorden op het InfoRequest bericht gepubliceerd moeten worden op het topic van het vragende dashboardsysteem.	✓	
		In het Protobuf bestand de velden van het AudioTracking bericht repeated gemaakt.		✓
4.5	8 juli 22	Bericht DetailStatusOverview toegevoegd. Daarmee kunnen leveranciers eigen statusgegevens naar een dashboard sturen. Bijbehorend topic toegevoegd en de enumeratie van request_type uitgebreid.	✓	✓
5.0	5 sept 23	Hoofdstuk toegevoegd over opbouwen en bewaken van de verbinding, zodat eisen uit het PvE nu in dit koppelvakdocument staan. (101) Veld toegevoegd in Subscribe waarmee haltesystemen aan kunnen geven hoeveel ritten ze in een TravelInfo bericht willen ontvangen (71). Diverse topicbeschrijvingen verbeterd, nieuw topic voor unsubscribe van distributiesysteem. (79, 89, 90, 91) "Already Subscribed" toegevoegd in Subscriptionresponse (97). "contract referentie" toegevoegd in Subscribe (99). Lijst met quaynummers toegevoegd in bericht Stopsystems (82) Timestamp toegevoegd in Subscribe (112). In aanmeldvolgorde van een haltesysteem het bericht SystemInfo toegevoegd (113) In ScreenContentResponse vanwege achterwaardse compatibiliteit de repeated screen_contents en Screen_index op uitfaseren gezet. Alleen het eerste schermbeeld wordt nog verwerkt aan de dashboardkant. (114). Bijlagen 1 en 2 verwijderd. Worden naar separaat document gebracht.	✓	✓
6.0	26 juni 24	Code uit de logboodschappen toegevoegd aan StatusOverviewLine. Bij de Subscribe kan het haltesysteem aangeven dat het minimale data wil krijgen. ScreenResponse in Tekstvorm moet in CSV en UTF8, plus een voorbeeld	✓	✓

6.1	????	Carriage return Line feed toegevoegd aan CSV voorbeeld. Extra velden in TravellInfo en FieldFilter met RGB waarden voor rand lijnnummer (109) (De Lijn) Bijlage 2 toegevoegd met uitleg van het filtermechanisme voor energiezuinige systemen Trip_stop_status CANCELLED veranderd naar CANCELLEDJOURNEY en status CANCELLEDCALL toegevoegd. In bericht ScreenContentResponse de velden screen_contents en screen_index verwijderd (stonden op Verouderd). In Subscribe een veld language opgenomen met de talen van het haltesysteem. In de berichten Destination, PublicName en GeneralMessage velden toegevoegd voor meertaligheid.	✓	✓
-----	------	---	---	---

Bijlage 1. Voorbeeld van een schermbeeld in CSV

Indien een Haltesysteem het volgende schermbeeld als tekst wil versturen naar een Dashboard,



Wordt het beeld als volgt in CSV geformatteerd, zodat het beeld beter leesbaar is in een teksteditor:

```
10:53<cr><lf>
1<tab>(+) <tab>Kerkelanden<tab>4 min<cr><lf>
53<tab>(+) <tab>Rotterdam Zuidplein<tab>7 min<tab>via Ziekenhuis<cr><lf>
135<tab>(+) <tab>Erfgooierskwartier<tab>10 min<cr><lf>
Lijn E rijdt niet tussen Den Haag Centraal en Nootdorp. De oorzaak is een wisselstoring. Er worden bussen ingezet. <cr><lf>
```

Bij vrijeteksten worden de titel en de tekst samengevoegd met een punt plus een spatie achter de titel. Indien er meerdere vrijeteksten (alternerend) worden getoond op het scherm, worden deze als separate regels in de CSV geplaatst.

```
10:53
1 (+) Kerkelanden 4 min
53 (+) Rotterdam Zuidplein 7 min via Ziekenhuis
135 (+) Erfgooierskwartier 10 min
Lijn E rijdt niet tussen Den Haag Centraal en Nootdorp. De oorzaak is een wisselstoring. Er worden bussen ingezet.
```

Indien in het mededelingen veld meerdere teksten alternerend worden getoond, worden deze, gescheiden door een `<spc>"|"<spc>`, samengevoegd in de CSV. Als ook bezettingsinformatie wordt getoond, wordt die informatie achter de tekst geplaatst, gescheiden door een `<spc>"|"<spc>`. De iconen worden weergegeven als:

 zitplaatsen beschikbaar
 staanplaatsen beschikbaar
 vol

De volgende regel



Wordt: 53<tab>(+) <tab>Rotterdam Zuidplein<tab>7 min<tab>via Ziekenhuis | zitplaatsen beschikbaar

Bijlage 2. Filtermechanisme energiezuinige haltesystemen

Doel

Energiezuinige haltesystemen (accu) kunnen gebruik maken van een filtermechanisme in het versturen van TravellInfo PassingTime berichten. Dit filter mag geen afbreuk doen aan de kwaliteit van de reisinformatie op de halte, hetgeen kan worden bereikt door de juiste instellingen.

Werking

De distributieserver van het CDD bepaalt bij een geactiveerd filter aan de hand van vier parameters of een TravellInfo PassingTime bericht verstuurd dient te worden.

TravellInfo GeneralMessage, TravellInfo GeneralMessageRemove en TravellInfo PassingTimeRemove berichten worden nooit gefilterd.

Activeren

Het filtermechanisme kan door het Haltesysteem geactiveerd worden. Hiervoor dienen bij het aanmelden op de distributieserver het veld filter_on op true en de waardes T1, T2, P1 en P2 meegestuurd te worden in het Subscribe bericht.

Parameters

Bij de filtering worden vier parameters gebruikt:

- Twee wachttijden in seconden: T1 en T2¹
- Twee percentages: P1 en P2.

Conditie

TravellInfo PassingTime berichten van eenzelfde rit (=pass_time_hash) zullen bij een geactiveerd filtermechanisme alleen naar het Haltesysteem gestuurd worden indien aan minimaal 1 van de volgende condities voldaan is:

- De trip_stop_status is niet gelijk aan de trip_stop_status van het laatst verzonden TravellInfo PassingTime bericht
- De wachttijd $\leq T1$ en het verschil tussen de eerder verstuurd wachttijd en de nu te versturen wachttijd $> P1$
- De wachttijd $> T1$ en $\leq T2$ en het verschil tussen de eerder verstuurd wachttijd² en de nu te versturen wachttijd $> P2$
- Als de trip_stop_status ARRIVED is en de wachttijd is niet gelijk aan de eerder verstuurd wachttijd.

Voorbeelden

De volgende voorbeelden gebruiken de parameters: T1 = 300 sec. T2 = 1800 sec. P1 = 1% en P2 = 10%

Voorbeeld 1

Stel om 12:00 is gestuurd dat de bus om 12:15:00 vertrekt. Om 12:01 blijkt dat de bus om 12:16:00 zal vertrekken.

¹ waarbij de wachttijd = vertrektijd in het bericht – actuele tijd.

² Dit is de wachttijd die op het moment van berekening door het Haltesysteem wordt getoond!

Afwijking = $| (12:15:00 - 12:01:00) - (12:16:00 - 12:01:00) | / (12:16:00 - 12:01:00) * 100\% = 6,66\%$ ¹

Wachttijd = 12:16:00 – 12:01:00 = 900 seconden.

De wachttijd is > T1 en <= T2 en de afwijking < P2 % dus het bericht wordt gefilterd en niet verstuurd vanwege criterium C

Voorbeeld 2

Stel om 12:00 is gestuurd dat de bus om 12:03:30 vertrekt. Om 12:01 blijkt dat de bus om 12:04:00 zal vertrekken.

Afwijking = $| (12:03:30 - 12:01:00) - (12:04:00 - 12:01:00) | / (12:04:00 - 12:01:00) * 100\% = 16,6\%$

Wachttijd = 12:04:00 – 12:01:00 = 180 sec.

De wachttijd is <= T1 en de afwijking is > P1, dus het bericht wordt niet gefilterd en gewoon gestuurd vanwege criterium B

Impact filtermechanisme op de getoonde wachttijd

Het toegepaste filtermechanisme van de TravellInfo PassingTime berichten borgt de volgende zaken:

- Alle TravellInfo PassingTime trip_stop_status status wijzigingen komen gegarandeerd door.
- Door het gebruik van relatieve afwijkingen in de filtering wordt de maximale afwijking die wordt gefilterd steeds kleiner naarmate de wachttijd korter wordt. Bij de instellingen uit de voorbeelden is dit bij 60 minuten nog <6 minuten; 15 minuten voor vertrek is dit <1,5 minuten; 2 minuten voor vertrek is dit nog maar <12 seconden. Dit beperkt de maximale afwijking van de getoonde wachttijd en voorkomt grote sprongen in de getoonde wachttijd.
- Door een minimaal percentage te gebruiken binnen 5 minuten voor vertrek wordt geborgd dat vlak voor vertrek vrijwel alle afwijkingen van de wachttijd doorgestuurd worden.

¹ Afwijking = $| 840 \text{ sec.} - 900 \text{ sec.} | / 900 \text{ sec.} * 100\%$