



Rijkswaterstaat
Ministerie van Infrastructuur en Milieu

RWS ONGECLASSIFICEERD

Basisontwerp TTI RWS Tunnelsysteem

Datum	9 december 2016
Status	Definitief
Release	1.2 SP2

RWS ONGECLASSIFICEERD

Basisontwerp TTI RWS Tunnelsysteem

Datum	9 december 2016
Status	Definitief
Release	1.2 SP2

Colofon

Uitgegeven door	Rijkswaterstaat Landelijk Tunnelregisseur
Opgesteld door	Landelijk Tunnelregisseur, Standaardisatieteam
Gecontroleerd door	Landelijk Tunnelregisseur, interne kwaliteitscontrole
Goedgekeurd door	Landelijk Tunnelregisseur
Contactpersoon	Fred Bouwmeester
Email	fred.bouwmeester@rws.nl
Releasenummer	1.2 SP2
Documentnummer	RWS00541-1-3339
Kader type	RWS Kader
Versienummer	1.2.2
Classificatie	RWS Ongeclassificeerd
Datum	9 december 2016
Status	Definitief

Versie historie

Datum	Versie	Toelichting	Vrijgegeven met
24 juni 2011	1.0	Vrijgegeven door LTR na verwerking commentaar uit brede review	Release 1.0
26 september 2011	1.1	Definitieve versie, vrijgegeven door Bestuur RWS	Release 1.1
8 oktober 2012	1.2	Vastgesteld en vrijgegeven door Bestuur RWS voor implementatie in staande organisatie en projecten	Release 1.2
20 november 2015	1.2.1	RWS-classificatie toegevoegd	Release 1.2 SP1 B3
9 december 2016	1.2.2	Goedgekeurde wijzigingsverzoeken verwerkt	Release 1.2 SP2

Release notes

Scope

- De scope van versie 1.2.2 van het Basisontwerp TTI RWS Tunnelsysteem (hierna: Basisontwerp) is niet gewijzigd ten opzichte van versie 1.2.1 van 20 november 2015;
- De scope van versie 1.2.1 van het Basisontwerp TTI RWS Tunnelsysteem is niet gewijzigd ten opzichte van versie 1.2 van 8 oktober 2012;
- De scope van versie 1.2 van het Basisontwerp TTI RWS Tunnelsysteem is niet gewijzigd ten opzichte van versie 1.1 van 26 september 2011;
- De scope van versie 1.1 van het Basisontwerp TTI RWS Tunnelsysteem is niet gewijzigd ten opzichte van versie 1.0 van 24 juni 2011.

Het Basisontwerp maakte geen deel uit van de oplevering versie 0.9 van de standaard.

Wijzigingen in versie 1.2.2 ten opzichte van versie 1.2.1

Alleen kleine wijzigingsvoorstellen, zie details hieronder.

Wijziging indeling

Er zijn geen wijzigingen in de indeling van het Basisontwerp.

Aandachtspunten voor volgende releases

Mogelijk verdere integratie met het Systeemontwerp.

Details wijzigingen

In versie 1.2.2 zijn t.o.v. versie 1.2.1 de volgende wijzigingen (met oorsprong) opgenomen: 209, 1291 (661), 1553 (1128).

Bij wijzigingsvoorstel 1291 (661) bleek uiteindelijk geen wijziging nodig.

Wijzigingen in versie 1.2.1 ten opzichte van versie 1.2

De RWS Classificatie (ongeclassificeerd) is toegevoegd en een opmerking t.a.v. achterhaalde tijdsplanning van overdracht van beheer is verwijderd. Er zijn geen wijzigingsvoorstellen verwerkt in deze versie van het Basisontwerp.

Wijzigingen in versie 1.2 ten opzichte van versie 1.1

Er is slechts één wijzigingsvoorstel verwerkt in versie van het Basisontwerp. Dit betreft een kleine tekstuele correctie. Dit betreft wijziging 213.

Wijzigingen in versie 1.1 ten opzichte van versie 1.0

In de versie 1.1 is slechts één wijziging conform het ingerichte change management proces doorgevoerd.

- Het betreft hier de verduidelijking van de "self containment eis" uit het basisontwerp. Uit de workshop met experts van de verschillende markt-partijen is duidelijk geworden dat er in de markt behoefte bestaat aan zulke verduidelijking. In consultatie met de marktpartijen is deze verdieping van eisen vorm gegeven;
- Tenslotte zijn in het kader van de mandateringsregeling een aantal tekstuele correcties aangebracht.

Het betreft de volgende wijzigingsvoorstellen:

- Mandateringsregeling, WV-nummer 018;
- Aanpassing eisen self contained unit, WV-nummer 025.

Managementsamenvatting

'Basisontwerp TTI RWS Tunnelsysteem', versie 1.2.2, d.d. 9 december 2016, beschrijft het Basisontwerp van Tunnel Technische Installaties (TTI's). Het Basisontwerp TTI heeft als primaire doelgroep die personen die verantwoordelijk zijn voor de implementatie van TTI's of het toezicht hierop. De lezer wordt verondersteld bekend te zijn met de Basisspecificatie TTI RWS Tunnelsysteem (ref. [BSTTI]), het Systeemontwerp RWS Tunnelsysteem (ref. [SysOntw]), als ook tunnel technische installaties in het algemeen.

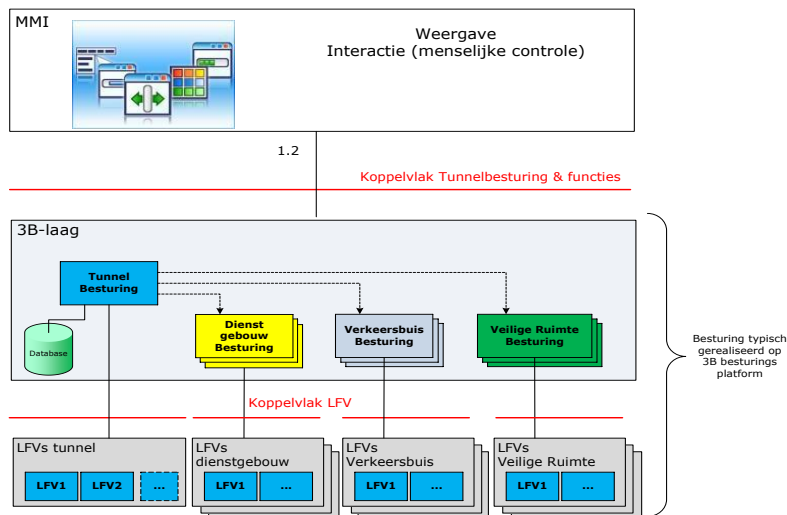
Het Basisontwerp TTI geeft de structuur waarin specifieke tunnel ontwerpen moeten passen. Het doel van deze structuur is om een duidelijk formaat te definiëren voor specifieke tunnel ontwerpen. Het Basisontwerp TTI legt hiertoe o.a. de naamgeving en contextbeschrijving van de te onderscheiden modules vast, zodat deze in specifieke documenten vergelijkbaar en begrijpelijk zijn. Figuur 1 geeft ter referentie de hoofdstructuur uit het Basisontwerp TTI weer.

Met deze hoofdstructuur wordt gewaarborgd dat zowel ontwerp als implementatie van TTI's dezelfde patronen en naamgeving volgen en daardoor herkenbaar zijn, terwijl marktpartijen voldoende vrijheid houden om zich te onderscheiden in hun implementatie. Dat laatste betekent dat er uitdrukkelijk vrijheid is in de keuze van implementatievormen door aanbieders van installaties.

Het Basisontwerp TTI legt het volgende voor modules vast:

- Verantwoordelijkheden;
- Belangrijkste koppelvlakken, zodat deelsystemen los vervangbaar en testbaar zijn;
- De relatie met functies en de Logische Functievullers uit de Basis-specificatie TTI, zodat functionaliteit en kwaliteit toetsbaar zijn.

Dit Basisontwerp TTI stelt eisen die aanvullend zijn op de Basisspecificatie TTI en die afgeleid zijn van o.a. de beleids- en bedrijfsdoelstellingen uit het Systeemontwerp.



Figuur 1 - Basisontwerp TTI met daarin de drie lagen zoals gedefinieerd in het Systeemontwerp

Inhoud

1	Inleiding.....	9
1.1	Identificatie.....	9
1.2	Doel en scope van document.....	9
1.3	Structuur.....	10
1.4	Documentbeheer.....	10
2	Referenties.....	11
2.1	Referenties.....	11
2.2	Afkortingen en definities.....	11
3	Systeem Context	12
3.1	Bediening.....	12
3.2	Tunnelbesturing (3B).....	13
3.3	Tunnelapparatuur (LFV).....	13
4	Niet-Functionele Eisen.....	14
4.1	Functionele Veiligheid.....	14
4.2	Beschikbaarheid en betrouwbaarheid.....	15
4.3	Uniformiteit en beperkte complexiteit.....	15
4.4	Toekomstgericht.....	15
4.5	Aanwijsbaarheid via Basisontwerp.....	16
4.6	Modulaire architectuur.....	16
5	Architectuur concepten en principes	17
5.1	Inleiding / terminologie.....	17
5.2	Architectuur concepten.....	17
6	Logische Decompositie	26
6.1	Logische view: MMI (bediening).....	27
6.2	3B-laag.....	28
6.3	Structuur van 3B besturingsmodules.....	29
6.4	Aanwijsbaarheid door structuur (requirements mapping).....	30
6.5	TunnelBesturing.....	31
6.6	VerkeersbuisBesturing.....	31
6.7	DienstgebouwBesturing.....	31
6.8	VeiligeRuimteBesturing.....	31
6.9	LFV Laag (Logische FunctieVervuller).....	32
6.10	Opbouw LFV.....	32
6.11	Veilig falen.....	33

1 Inleiding

1.1 Identificatie

Dit document is geïdentificeerd als RWS00541-1-3339, Basisontwerp TTI RWS Tunnelsysteem, versie 1.2.2, d.d. 9 december 2016. Dit document is onderdeel van release 1.2 SP2 van de tunnelstandaard.

1.2 Doel en scope van document

Dit document beschrijft het basisontwerp van Tunnel Technische Installaties (TTI). De primaire doelgroep bestaat uit mensen die verantwoordelijk zijn voor de implementatie van TTI's of het toezicht hierop. De lezer wordt verondersteld bekend te zijn met de Basisspecificatie TTI RWS Tunnelsysteem (ref. [BSTTI]), als ook tunnel technische installaties in het algemeen.

Een tunnel bestaat naast asfalt, beton en staal uit vele onderdelen die nauw met elkaar samen moeten werken om veiligheid en doorstroming van verkeer te waarborgen. Hiertoe behoort een groot aantal technische systemen, welke de tunnel bewaken en besturen en die samen met menselijke bediening de veiligheid waarborgen. Het basisontwerp zoals beschreven in dit document beschrijft *wat* (functionaliteit) *waar* (aanwijsbaarheid in het systeem) gerealiseerd moet worden. Hiermee stelt het basisontwerp strikte eisen aan de structuur van een specifiek ontwerp en waarborgt het moeilijk kwantificeerbare eisen als testbaarheid, uitbreidbaarheid en beperking van complexiteit (in de besturing en de controle) terwijl het ook voor uniformiteit zorgt.

Het basisontwerp is een abstracte beschrijving in de vorm van een referentiearchitectuur. Het doel hiervan is om een duidelijk formaat te definiëren voor specifieke ontwerpen. Deze beschrijving betreft alleen de grote lijnen; specifieke ontwerpen gaan een aantal stappen verder en moeten de noodzakelijke details invullen.

Dit alles moet er voor zorgen dat zowel ontwerp als implementatie van TTI's dezelfde patronen en naamgeving volgen en daardoor herkenbaar zijn, terwijl marktpartijen voldoende vrijheid houden om zich te onderscheiden in hun implementatie.

Dat betekent dat er uitdrukkelijk vrijheid is in de keuze van implementatievormen door aanbieders van installaties, bijvoorbeeld in de keuze van hardware (PLC's, PC's), in software (bijvoorbeeld operatingssystemen, programmeertalen) en in de manier waarop de geëiste niet-functionele eisen (NFE's) en functionele eisen gerealiseerd worden.

Het basisontwerp legt het volgende voor (deel)systemen vast:

- Naamgeving en contextbeschrijving van te onderscheiden modules (zodat deze vergelijkbaar en begrijpelijk zijn);
- Verantwoordelijkheden;
- Belangrijkste Koppelvlakken (zodat deelsystemen los vervangbaar en testbaar zijn);
- De relatie met functies en de Logische Functievervullers uit de Basisspecificatie, ref. [BSTTI] (zodat functionaliteit en kwaliteit toetsbaar zijn).

Hierdoor wordt eenduidige definitie en aanwijsbaarheid van specificatie naar implementatie mogelijk gemaakt. Dit doel volgt uit de standaardisatie RWS Tunnelsysteem, zie Leeswijzer RWS Tunnelsysteem (ref. [Leeswijzer]).

Dit basisontwerp stelt eisen die aanvullend zijn op de Basisspecificatie (ref. [BSTTI]), en die afgeleid zijn van o.a. de beleids- en bedrijfsdoelstellingen uit het Systeemontwerp (ref. [SysOntw]).

Het Basisontwerp bestaat uit:

- Systeem context;

- Niet Functionele Eisen;
- Architectuurprincipes en concepten;
- Modules (naamgeving, bereik, omschrijving).

Dit document beschrijft niet het specifieke tunnelontwerp en daarbij behorende realisatiekeuzes, waaronder hardware- en software-platformen. Er is gekozen om het dynamische gedrag van het systeem (en daarmee een performance analyse) en de betrouwbaarheidsanalyse buiten het bereik van dit document te laten vallen, evenals relaties tot organisatie en processen (zoals VS-2).

1.3 Structuur

Dit document maakt onderdeel uit van de tunnelstandaard die uit meerdere documenten bestaat. De Leeswijzer tunnelstandaard (ref. [Leeswijzer]) geeft een toelichting op de verschillende documenten van de tunnelstandaard alsook op de samenhang tussen deze documenten. Ook geeft de leeswijzer de achtergrond, de aanpak en de doelgroep van de tunnelstandaard.

De Basisspecificatie Tunnel Technische Installaties (ref. [BSTTI]) maakt ook deel uit van de tunnelstandaard. Aanbevolen wordt om dat document eerst te lezen, om zo bekend te worden met de gebruikte terminologie en context.

Wanneer er in dit document gedetailleerde achtergrond informatie wordt gegeven of voorbeelden worden gegeven, dan zijn deze soms gemarkeerd met 'toelichting' respectievelijk 'voorbeeld' in het opschrift. Deze op zichzelf staande delen kunnen overgeslagen worden zonder verlies van essentiële informatie, maar helpen wel om concrete handvatten te bieden bij de soms abstracte beschrijvingen. Voorbeelden die een directe relatie met de specificatiedocumenten gebruiken zijn niet gemarkeerd.

Architectuureisen worden weergegeven als:

BOTTI#250

Voorbeeld van een architectuureis.

Hoofdstuk 3 beschrijft de systeemcontext van de TTI.

Hoofdstuk 4 beschrijft de belangrijkste Niet Functionele Eisen in relatie tot de bedrijfsdoelstellingen.

Hoofdstuk 5 beschrijft de gebruikte en vereiste architectuurprincipes.

Hoofdstuk 6 beschrijft de logische decompositie van het systeem met daarin verschillende onderdelen van de TTI architectuur, hun inhoud en ontwerpisen.

1.4 Documentbeheer

Dit document maakt deel uit van de tunnelstandaard die momenteel beheerd wordt door de Landelijk Tunnelregisseur. Met de toekomstige overdracht van de LTR-producten en diensten aan de lijnorganisatie worden het eigenaarschap en het beheer van de tunnelstandaard overgedragen naar de lijnorganisatie (ref. [Leeswijzer]).

2 Referenties

De verwijzingen in de referentietabel betreffen de actuele versies van genoemde documenten. Voor zover in deze tabel geen specifieke versies genoemd zijn, zijn die terug te vinden in ref. [Leeswijzer].

2.1 Referenties

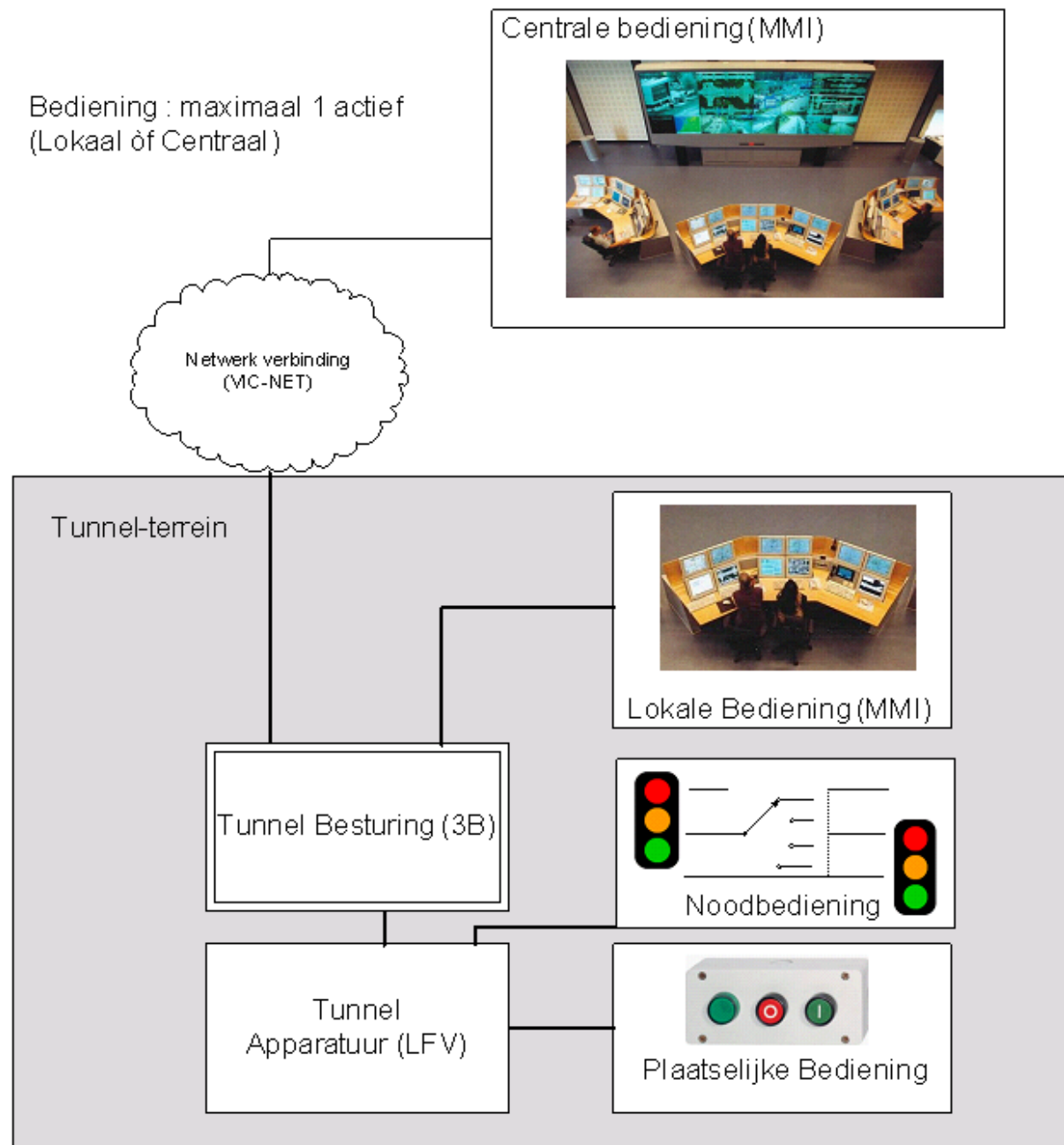
[A&D]	Afkortingen en definities tunnelstandaard
[BSTTI]	Basisspecificatie TTI RWS Tunnelsysteem
[Leeswijzer]	Leeswijzer Tunnelstandaard
[QA-LTS]	Kwaliteitsborging Tunnelstandaard
[RFC2119]	Key words for use in RFCs to Indicate Requirement Levels, IETF Network Working Group, maart 1997
[SysOntw]	Systeemontwerp RWS Tunnelsysteem

2.2 Afkortingen en definities

Zie ref. [A&D] voor afkortingen en definities die algemeen voor de tunnelstandaard gelden.

3 Systeem Context

Dit hoofdstuk beschrijft de context van de TTI. Hierin worden conceptuele functionele blokken aan geografische locatie en verantwoordelijkheid gekoppeld.



Figuur 2 Systeem context

In Figuur 2 is de context van de Tunnel Technische Installatie weergegeven. Het grijze stuk bevindt zich op (of onder) het terrein van de tunnel.

3.1 Bediening

De bediening van een tunnel kan vanaf diverse plaatsen gebeuren: *lokale bediening* bij de tunnel of *centrale bediening* in een verkeerscentrale. Beide vormen worden *MMI* genoemd (Mens Machine Interface). Voor de besturing van de tunnel maakt het niet uit waarvandaan de *MMI*bediening

plaatsvindt. *Plaatselijke bediening* is direct gekoppeld aan apparatuur in de tunnel, en biedt bijvoorbeeld directe controle voor onderhoud.

Noodbediening maakt het mogelijk om verkeersbuizen te sluiten onafhankelijk van de besturing. Daartoe bestaat er een directe koppeling van apparatuur zoals verkeerslichten en slagbomen in de verkeersbuizen naar een noodbedieningspaneel.

De MMI is verantwoordelijk voor het weergeven van status van tunnelbesturing en tunnelapparatuur, het selecteren van CCTV beelden, en doorgeven van operator commando's richting tunnel. De interne opbouw van de MMI wordt niet verder behandeld in dit document, alleen het data-gerelateerde (als client van de 3B laag).

3.2 Tunnelbesturing (3B)

Tunnelbesturing is een systeem bestaande uit besturingsapparatuur en/of computers met bijbehorende programmatuur met de verantwoordelijkheid om de tunnelapparatuur aan te sturen en te monitoren. Dit systeem vertaalt operatorcommando's in opdrachten en instellingen voor tunnelapparatuur. De andere kant op vertaalt dit systeem informatie van sensoren en besturingen in statusinformatie voor de MMI.

De tunnelbesturing bepaalt het gedrag van de tunnel, en realiseert dus ook de states.

LFV's op tunnelniveau worden direct door Tunnelbesturing aangestuurd en gecontroleerd.

3.3 Tunnelapparatuur (LFV)

Tunnelapparatuur is een verzameling van componenten uit de tunnelinstallatie.

De verantwoordelijkheid van de tunnelapparatuur is om de besturingscommando's over te brengen naar de fysieke wereld en deze middels sensoren te monitoren. Noodbediening heeft directe interactie met LFV's voor verkeerslichten en afsluitbomen. Plaatselijke bediening heeft interactie met LFV's en/of LFV-componenten.

4 Niet-Functionele Eisen

Niet Functionele Eisen (NFE's) zijn direct gerelateerd aan bedrijfs- en beleidsdoelstellingen. Ze stellen eisen aan het ontwerp en zijn veelal systeemoverkoepelend. Als zodanig is een aantal ook geadresseerd in dit basisontwerp. Voor complete definities van NFE's en doelstellingen, zie het Systeemontwerp (ref. [SysOntw]).

De lijst van NFE's relevant voor het ontwerp worden de TTI Ontwerp NFE's genoemd. De TTI Ontwerp NFE's worden afgeleid uit de beleids- en bedrijfsdoelstellingen van Rijkswaterstaat. Dit document stelt eisen die aanvullend zijn aan de eisen uit de Basisspecificaties (ref. [BSTTI]).

De volgende tabel geeft een overzicht van de verschillende NFE's en hun onderlinge prioriteit.

Tabel 1 Niet-functionele eisen (NFE's)

ID	Prio.	Bedrijfs- en beleidsdoelstellingen	Afgeleide eis
1	1	Een tunnel voldoet aan het veiligheids niveau zoals dat volgens de Nederlandse wet wordt geëist en is betrouwbaar.	<i>Functionele veiligheid Beschikbaarheid en Betrouwbaarheid</i>
2	2	Een tunnel biedt wegverkeer zo vlot mogelijke doorstroming (minder hinder).	<i>Beschikbaarheid en Betrouwbaarheid</i>
3	3	Een tunnel is uniform en standaard in werkwijze en onderdelen.	<i>Uniformiteit</i>
4	4	Een tunnel is maatschappelijke kosten effectief gedurende de levensloop en toekomst gericht (o.a. in ontwerp).	<i>Toekomstgericht</i>
5	5	Een tunnel is qua complexiteit inzichtelijk en beheerst.	<i>Uniformiteit en beperkte complexiteit</i>
6	5	Een implementatie van de TTI moet geleverd worden door de vrije markt tenzij dit de veiligheid of doorstroming beïnvloedt.	Buiten bereik van dit document.
7	5	Een tunnel moet bij oplevering en gedurende gebruik aantoonbaar en verifieerbaar veilig zijn.	<i>Functionele veiligheid</i>
8	5	Een tunnel wordt efficiënt aangelegd, gebruikt, beheerd en onderhouden.	<i>Modulaire architectuur</i>
9	5	Een tunnel voldoet aantoonbaar aan gestelde eisen en principes.	<i>Aanwijsbaarheid via basisontwerp</i>

De afgeleide eisen zijn in de navolgende paragrafen uitgewerkt.

4.1 Functionele Veiligheid

Het veiligheidsniveau van een tunnel is uitgedrukt in absolute veiligheidsgetallen voor een tunnel. Functionele veiligheid is een inherent onderdeel van de basisspecificatie TTI, ref. [BSTTI]. Beiden moeten aantoonbaar zijn in iedere tunnel.

De veiligheid van de TTI wordt *aantoonbaar* afgedwongen door de volgende keten van aan te tonen verificaties:

- De basisspecificatie TTI (ref. [BSTTI]) voldoet aan wet- en regelgeving;
- Het Specifiek Ontwerp TTI voldoet aan:
 - de specifieke eisen VS-1;
 - de basisspecificatie, ref. [BSTTI];
 - het basisontwerp TTI (dit document);

- De implementatie van een Specifiek Ontwerp TTI voldoet aan het basisontwerp TTI.

Dit basisontwerp draagt direct bij aan de aanwijsbaarheid van functionele veiligheid en aan reductie van complexiteit. Deze beiden maken het eenvoudiger om veiligheid aan te tonen en verlagen de kans op menselijke ontwerp- en realisatiefouten. Op landelijk niveau zal ook de uniformiteit de veiligheid ten goede komen.

4.2 Beschikbaarheid en betrouwbaarheid

De beschikbaarheid van een verkeersbuis kan worden uitgedrukt in een getal:

$$\text{Beschikbaarheid} = \frac{\text{totale tijd geopend}}{\text{totale tijd (geopend + gesloten)}} * 100\%$$

De beschikbaarheid wordt beter naarmate:

- Componenten van tunnel technische installaties betrouwbaarder zijn (buiten scope);
- Eventueel onderhoud tijdens geplande werkzaamheden kan plaatsvinden (buiten scope);
- Sluitingstijd bij benodigd onderhoud (en oplevering en opwaardering) van TTI zo kort mogelijk is.

De beschikbaarheid van de tunnel wordt bepaald door de beschikbaarheid van de verkeersbuizen; als 1 of meer buizen gesloten zijn dan is de beschikbaarheid van de tunnel evenredig hiermee.

Kwaliteitsaspecten van dit basisontwerp TTI:

- Modulaire aanpak en minimale koppeling tussen modules;
- Zorgen voor minimale sluitingsduur bij onderhoud/vervanging van TTI deelsystemen;
- Minimaliseren van benodigde afsluitingen bij onderhoud (een afgesloten verkeersbuis heeft minder invloed dan de hele tunnel).

Samen zorgen deze voor betere testbaarheid gedurende ontwikkeling en integratie en onderhoud, en dragen daarmee bij aan minder fouten in operatie, en kortere integratietijd. Dit resulteert in een hogere betrouwbaarheid (hogere MTBF en lagere MTTR).

Bij deelsystemen is beschikbaarheid genuanceerder: er kan ook een beperkte beschikbaarheid zijn, en indien deze groter of gelijk is aan de vereiste capaciteit hoeft deze geen invloed te hebben op de beschikbaarheid van verkeersbuis of tunnel.

4.3 Uniformiteit en beperkte complexiteit

Door TTI implementaties volgens het basisontwerp TTI te realiseren ontstaat uniformiteit in naamgeving, structuur en aanpak.

Verdere uniformiteit wordt verkregen in de standaardisatie van gemeenschappelijke koppelvlakken. De gelaagdheid in dit basisontwerp beperkt complexiteit zoveel mogelijk.

4.4 Toekomstgericht

Het basisontwerp TTI maakt geen aannames rond hardware en software, en is daarmee weinig gevoelig voor tijd. Door te faciliteren dat modules (na verloop van tijd) uitgewisseld kunnen worden met minimale impact en koppelvlakken te scheiden van implementatie is de TTI

toekomstgericht. Een ander punt is een scheiding van belangen (zie 5.2.7), waardoor eventuele wijzigingen een zo lokaal (en dus minimaal) mogelijke impact hebben.

Zelfomvattende modules en gedefinieerde koppelvlakken zorgen voor betere uitwisselbaarheid waardoor deelsystemen vervangen kunnen worden met minimale impact. Zie ook 5.2.3 en 5.2.4.

4.5 Aanwijsbaarheid via Basisontwerp

Doordat dit basisontwerp voorschrijft waar binnen de structuur iedere functionaliteit uit de basisspecificatie (ref. [BSTTI]) moet worden gerealiseerd is functionaliteit aanwijsbaar.

4.6 Modulaire architectuur

Modulaire architectuur maakt het mogelijk om onderdelen aan te wijzen, te testen en te vervangen. Dit moet resulteren in kortere sluitingstijden voor ingebruikname en vervanging van onderdelen.

5 Architectuur concepten en principes

5.1 Inleiding / terminologie

Dit hoofdstuk beschrijft belangrijke architectuurconcepten en de van toepassing zijnde architectuurprincipes met betrekking tot de tunnelarchitectuur. Deze principes zijn ontwerpregels waaraan moet worden voldaan bij ontwerp en implementatie van TTI. Als basis dient een aantal ontwerpregels en -patronen uit de systeemarchitectuur, waar ze met succes toegepast zijn op systemen met vergelijkbare complexiteit.

Terminologie, specifiek voor dit hoofdstuk:

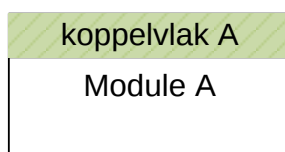
- Architectuur concept: ontwikkelpatroon;
- Architectuur principe: een eis vanuit de architectuur aan specifiek ontwerp/implementatie;
- Systeem: een aantal modules (en bijbehorende componenten) dat samen een bepaalde functionaliteit realiseert;
- Component: hardware onderdeel van een TTI;
- Module: logisch onderdeel van het systeem;
- Zelfomvattende module: geïsoleerde module met minimale afhankelijkheden;
- Koppelvlak: een gedefinieerde (of gestandaardiseerde) manier waarmee functionaliteit van een module aangesproken kan worden.

In de navolgende paragraaf worden architectuurconcepten gedefinieerd. Bij de meeste concepten horen architectuurprincipes welke als ontwerpeisen gespecificeerd zijn nabij de uitleg. Aan het einde van dit hoofdstuk staat een tabel met de relatie tussen de belangrijkste architectuurprincipes en de NFE's.

5.2 Architectuur concepten

5.2.1 Koppelvlak

Modules communiceren uitsluitend met elkaar door middel van gedefinieerde koppelvlakken. Dit mechanisme beperkt de mogelijke vormen van interactie en daarmee de complexiteit van het systeem. Dit is weergegeven in de figuur 3 hieronder. Een andere module die gebruik wil maken van Module A hoeft alleen te weten hoe het koppelvlak gedefinieerd is, niet hoe module A intern werkt. Hierdoor wordt functionaliteit (implementatie) gescheiden van de interface (het koppelvlak). Het gedrag van modules is overigens wel relevant, met name voor functionele veiligheid en besturing. In dit document wordt aangenomen dat gedrag gedefinieerd is in het koppelvlak.



Figuur 3 Koppelvlak

NB: dit document verstaat alleen de gedefinieerde interface naar functionaliteit of data van een module als koppelvlak.

Een koppelvak heeft een naam (module-naam), en er hoort een specificatie bij (in een specifiek ontwerp). Hierin moet ook het gedrag beschreven staan.

BOTTI#1 Communicatie tussen modules moet uitsluitend via gedefinieerde koppelvakken gaan.

BOTTI#2 Ieder koppelvak moet de specificatie volgen in naamgeving en gedrag.

BOTTI#3 Bij uitbreidingen van koppelvakken (bijvoorbeeld voor aanvullende of optionele functionaliteit) moet de specificatiewijze uit de Basisspecificatie TTI gevolgd worden (bijvoorbeeld `object.Commando()` en `#variabele = waarde`).

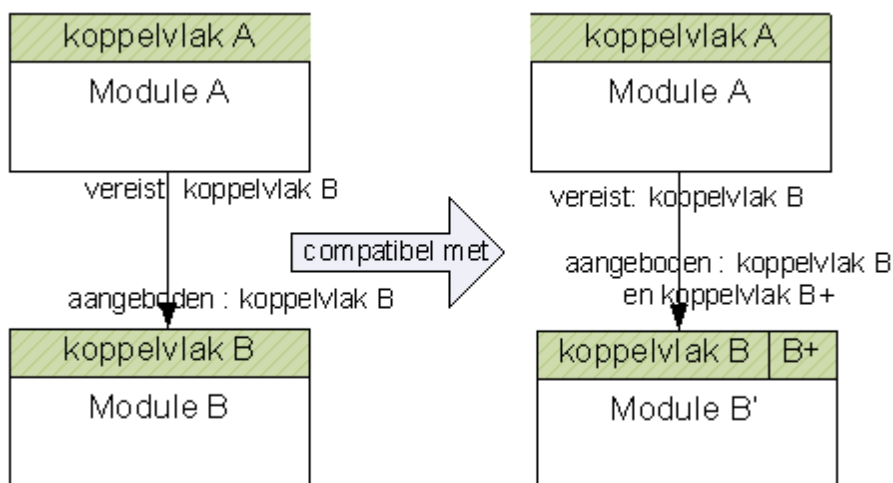
5.2.2 Koppelvak Families

In een aantal gevallen is een deel van het koppelvak gestandaardiseerd, of is het een onderdeel van een reeks. Een voorbeeld: In de BSTTI is een aantal Logische Functie Vervullers (LFV's) gedefinieerd. Alle LFV's kunnen op een uniforme manier aangesproken worden, hoewel de achterliggende functionaliteit fundamenteel anders is. Iedere LFV heeft bijvoorbeeld dezelfde `SetBestuurbaar()` operatie.

5.2.3 Voorbeeld gebruik koppelvakken: vervanging modules

Modules moeten vervangen kunnen worden door een module met een compatibel koppelvak.

Voorbeeld Gebruik koppelvakken: vervanging modules



Figuur 4 Koppelvak voorbeeld: vervangbaarheid modules

In Figuur 4 is Module B te vervangen door module B'. Hoewel B' een uitgebreider koppelvak heeft (B en uitbreiding B+) voldoet het aan de door module A vereiste en biedt (ook) Koppelvak B aan.

BOTTI#4 Iedere module moet vervangen kunnen worden door een module met een compatibel koppelvak (indien ook het gedrag compatibel is).

5.2.4 Zelfomvattende modules (Self Contained Modules)

Dit concept, dat ook wel *self contained modules* genoemd wordt, relateert aan de minimale afhankelijkheid van modules aan hun omgeving op ieder niveau. Dit komt terug in ieder aanzicht (view). Op logisch niveau moeten de modules duidelijk herkenbaar zijn, maar ook op executieniveau moet duidelijk zijn dat de modules elkaar niet nadelig kunnen beïnvloeden.

Door het isoleren van afhankelijkheden:

- Blijft een module functioneren, onafhankelijk van andere modules. Indien een ander deel van het systeem defect raakt heeft dit geen directe invloed;
- Is een module goed vervangbaar (beperkte afhankelijkheden);
- Is een module op zichzelf testbaar (waarbij vereiste koppelvlakken gerealiseerd moeten zijn in de testomgeving);
- Kan een stapsgewijze integratie van het systeem plaatsvinden.

Verder is het van belang dat de zelfomvattende module bewaakt wordt, zodat het systeem kan herstarten of in het ergste geval: uitschakelen. Hiervoor is het nodig dat er een vorm van integriteitscontrole plaats vindt.

Voorbeeld - Self Contained Unit (hardware/software)

Stel, een tunnelbuis besturing is opgebouwd uit een aantal PLC's. Alle functionaliteit voor de branddetectie van een verkeersbuis is ondergebracht in een aparte PLC, die de complete functionaliteit van de LFV Branddetectie realiseert. Indien er geen verdere afhankelijkheden zijn, spreken we van een self contained unit.

NB: Er zijn vele vormen waarin een zelfomvattende module te realiseren is, ook binnen een PLC.

BOTTI#5 Elke in deze architectuur genoemde module moet als zelfomvattende module geïmplementeerd worden.

BOTTI#32 Er is sprake van een zelfomvattende module indien:

- De module volledig eigen hardware (CPU, I/O, netwerk) heeft of;
- De software onderdelen van de module op een gemeenschappelijke hardware draaien en netwerk + I/O modules delen onder de voorwaarden zoals beschreven in de volgende eisen.

BOTTI#6 Een zelfomvattende module mag geen variabelen en/of I/O punten delen met andere modules. Dit moet in het ontwerp gegarandeerd zijn door (en/of):

- Technische middelen (bijvoorbeeld iedere module heeft zijn eigen variabelen die aantoonbaar niet door andere modules gebruikt kunnen worden. Dit is bijvoorbeeld het geval als de compiler/linker dit niet mogelijk maken);
- Procesmatige maatregelen (bijvoorbeeld naamgeving van de variabelen en/of I/O poorten is aantoonbaar uniek per module, gebruik van pointers en andere methoden voor vrije adressering is niet toegestaan).

BOTTI#30 Zelfomvattende modules mogen niet onderling communiceren als zij op hetzelfde aggregatieniveau geïmplementeerd zijn.

BOTTI#7 In het geval van falen van een module op gemeenschappelijke hardware zorgt correcte foutafhandeling voor een lokale herstart en voor een maximale beschikbaarheid van het totaalsysteem (en dus de tunnel).

- Het totale systeem blijft functioneren ook als een van de modules bijvoorbeeld hangt of door nul deelt;
- Andere modules op dezelfde hardware worden niet functioneel beïnvloed door de problemen in andere modules.

BOTTI#8 Voor alle modules geldt dat de belasting van het netwerk en de CPU <70% in nominaal gebruik moet zijn. Dit geldt dus ook expliciet voor gemeenschappelijk (door meerdere modules) gebruikte hardware.

- Nominaal gebruik is tunnel operatie gedurende 1 uur;
- Hierin moeten alle state overgangen en alle functies geactiveerd worden;

- Dit moet problemen bij piekbelasting en functionele wijzigingen voorkomen.

BOTTI#9

Iedere zelfomvattende module moet aan integriteitcontroles onderworpen zijn.

BOTTI#10

Iedere zelfomvattende module moet onafhankelijk te vervangen zijn (in onderhoudsmode). Hierbij mag eventuele gedeelde hardware opnieuw opgestart worden.

BOTTI#31

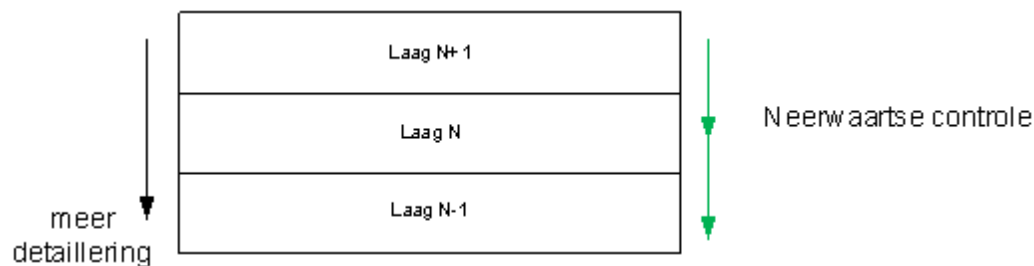
Zelfomvattende modules moeten aanwijsbaar zijn als aparte eenheden ook als zij op gemeenschappelijke hardware geïmplementeerd zijn.

- Bijvoorbeeld als onderdeel van aparte link/compile eenheden;
- Bijvoorbeeld door geen files te delen.

5.2.5 Gelaagdheid (Layering) en Neerwaartse Controle

In de voorgaande paragrafen is gesproken over modules en koppelvlakken, zonder te kijken naar samenhang anders dan externe afhankelijkheid. Bij gelaagdheid gaat het juist over deze samenhang.

Gelaagdheid simplificeert de architectuur van een complex systeem door een aantal afspraken te maken. Deze afspraken scheppen een context, en maken het mogelijk om op een abstracter niveau naar het systeem te kijken: details van lagere lagen zijn niet meer relevant (of zichtbaar) op hogere lagen. Verder helpt de gelaagdheid in het reduceren van afhankelijkheden.



Figuur 5 Gelaagdheid met neerwaartse controle

1. Iedere architectuurlaag bevat modules met eenzelfde mate van detaillering.
 - iedere module zit in precies één of overspant twee architectuurlagen.
2. Hoger liggende lagen bevatten een lager detailniveau (en meer overzicht).
 - Lagerliggende lagen bevatten een hoger detailniveau, meer gespecialiseerde modules en minder overzicht.
3. Neerwaartse controle.
 - Laag 'N' is afhankelijk van diensten/functionaliiteit van Laag 'N-1' (en eventueel lager, indien aanwezig);
 - Laag 'N' biedt diensten/functionaliiteiten aan laag 'N+1' (de bovenliggende laag).
4. Iedere laag heeft een koppelvlak (een aggregatie van koppelvlakken van modules in de laag).
5. Iedere laag kan services bevatten die alleen voor modules in dezelfde laag beschikbaar zijn.

Doordat lagen (met daarin mogelijk meerdere modules) zich gedragen als modules, zijn de zelfde substitutieprincipes van toepassing:

- Wijziging binnen een laag heeft geen invloed op andere lagen;
 - Mits koppelvlak ongewijzigd.

Toelichting

Een systeem kan een aantal lagen hebben, waarbij dit aantal vaak zodanig gekozen is dat dit door mensen direct te overzien is. Om deze reden komen 3 tot 7 lagen vaak voor. Het Internet is bijvoorbeeld gecreëerd rond de 7 lagen van het OSI referentie model.

Voorbeeld - Gelaagdheid (illustratief voorbeeld binnen tunnelcontext)

- Laag N: MMI is te zien als hoogste laag van het totale systeem.
Operator selecteert 'Handmatig ventilatie niveau: 4' voor een verkeersbuis;
- Laag N-1: Basisfunctie Ventilatie.
Gewenste niveau: 4;
- Laag N-2: LFV Ventilatie.
Gewenst niveau : 4, mode: handmatig
bepaling beschikbare ventilatiecapaciteit, sturing individuele ventilatoren, diagnose.

5.2.5.1 Data en notificaties

De regels voor gelaagdheid gelden voor afhankelijkheid en voor controlerichting. Voor datastromen gelden geen beperkende regels. Notificaties (en alarmering) vinden plaats in opwaartse richting zonder beperkingen.

Voorbeeld - Datastromen in gelaagde module

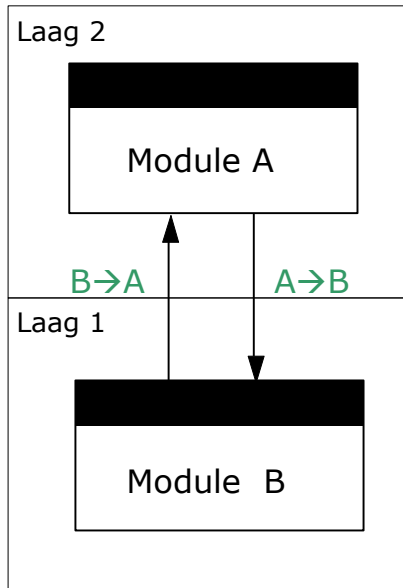
- Laag N: activeer periodieke sensor updates als datastroom #x met snelheid 10 Hz;
- Laag N-1: geeft datastroom met sensordata op 100ms intervallen door aan bovenliggende laag.

In dit geval is de controle strikt neerwaarts (laag N-1 doet pas iets na verzoek van laag N). De datastroom is in dit geval opwaarts. De beiden lagen voldoen aan de architectuureisen van gelaagdheid.

Voorbeeld - Notificatie (data)

- Laag N: module A: B.initieer_notificatie(A.InkomendAlarm);
- Laag N-1: module B: Indien er een alarm optreedt wordt B.AlarmFunctie() aangeroepen, welke verwijst naar A.InkomendAlarm(); eventueel kan er ook data overgedragen worden.

Ook deze variant voldoet aan de architectuurregels: B is namelijk niet afhankelijk van A. Ook gaat het in dit voorbeeld niet om besturing, maar om notificatie welke toegestaan is.

**Tabel 2 Communicatievormen tussen lagen**

Soort \ richting	A→B	B→A
Afhankelijkheid koppelvlak	✓	✗
Afhankelijkheid interne werking	✗	✗
Aanroep koppelvlak (commando's, variabelen, configuratie)	✓	✗
Respons (terugmelding)	N/A	✓
Datastromen	✓	✓
Notificaties	N/A	✓

Figuur 6 Voorbeeld lagen en relaties

In de bovenstaande figuur staan een aantal voorbeelden van toegestane en niet toegestane (of niet relevante) communicatievormen tussen de lagen. Hierbij wordt de terminologie gebruikt:

- Module B heeft een aangeboden koppelvlak B;
- Module A vereist een koppelvlak B.

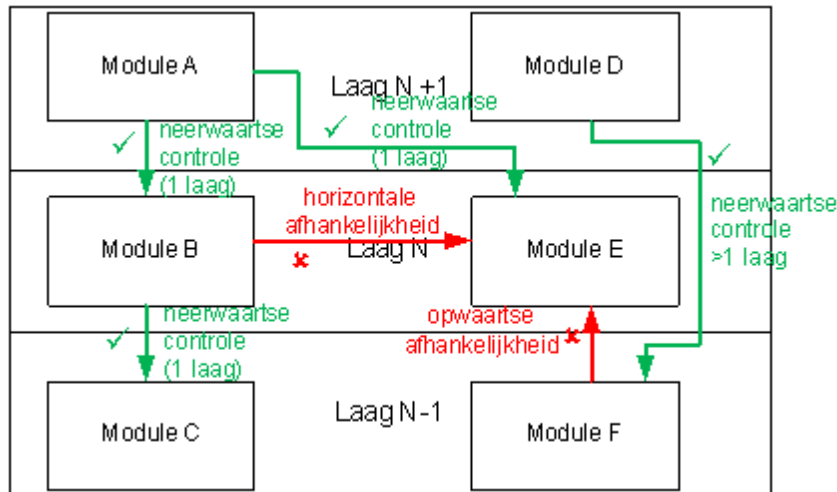
A mag dus wel functionaliteit van B gebruiken, maar B mag alleen op de door A aangegeven wijze communiceren met A.

Tabel 2 geeft een aantal mogelijke relaties aan tussen module A en B uit het voorbeeld van Figuur 6. Deze is als volgt te lezen:

- Afhankelijkheid koppelvlak: module A is afhankelijk van koppelvlak B. Module B is niet afhankelijk van module A;
- Er is geen afhankelijkheid naar interne werking;
- Module A mag wel het koppelvlak van B gebruiken, andersom mag niet;
- B geeft wel data terug naar A, bijvoorbeeld als antwoord op een status-vraag, of actief (als notificatie);
- Datastromen worden niet door de lagen bepaald.

5.2.6 Toegestane en niet toegestane afhankelijkheden

Om de voordelen van een gelaagde architectuur te kunnen waarmaken zijn er een aantal afspraken nodig. In deze paragraaf staan een aantal voorbeelden van toegestane en niet toegestane koppelingen.



Figuur 7 Module/laag afhankelijkheden

Neerwaartse controle

Dit maakt dat iedere laag N alleen bekend is met onderliggende lagen N-1 en verder (N-2, N-3, ...).

Dit houdt in dat lagen N+1 (en hoger) niet tot de afhankelijkheden mogen behoren.

Module A → Module B → Module C voldoet hieraan, en ook module A → Module E is in orde.

Neerwaartse controle tussen Module D → Module F is ook toegestaan, omdat deze 2 lagen lager zit. Het overslaan van lagen is echter alleen toegestaan indien de functionaliteit niet via direct onderliggende lagen beschikbaar is.

Horizontale afhankelijkheid (niet toegestaan!)

In Figuur 7 maakt Module B gebruik van Module E (in dezelfde laag). Dit is niet toegestaan.

Hierdoor kunnen conflicten in de besturing ontstaan, en zijn cyclische afhankelijkheden mogelijk.

Het is geen probleem wanneer een module alleen lokaal gebruikt kan (!) worden (en dus geen onderdeel uitmaakt van het koppelvlak van de laag). Dan kan een dergelijk module als lokaal onderdeel van andere modules gezien worden.

Opwaartse afhankelijkheid (niet toegestaan!)

In Figuur 7 maakt module F gebruik van Module E (welke in een bovenliggende laag zit). Dit is niet toegestaan. Vaak is het wel nodig dat lagen iets kunnen signaleren aan bovenliggende lagen, maar dat kan gedaan worden door een registratiemechanisme en dynamische koppelingen. Hierbij neemt laag N een abonnement op informatie van laag N-1. Het initiatief, en daarmee de afhankelijkheid ligt dan bij laag N. Zie ook voorbeeld notificaties in 5.2.5.

BOTTI#11 Ieder TTI ontwerp moet de niet-optionele lagen gedefinieerd in het basisontwerp implementeren.

BOTTI#12 Ieder TTI ontwerp moet deze gelaagdheid met neerwaartse controle toepassen.

BOTTI#29 Ieder TTI ontwerp moet alleen verticale neerwaartse afhankelijkheden gebruiken (niet horizontaal, niet opwaarts).

5.2.7 Scheiding van belangen (Separation of concerns)

Scheiding van belangen is een proces wat ingaat op het scheiden van een systeem in afzonderlijke eigenschappen, welke een minimale overlap in functionaliteit bevatten. Een belang is in deze een

punt van interesse of focus in een systeem. Veelal zijn dit functionaliteiten (functies) of gedragkenmerken. Binnen dit document en de BSTTI wordt 'scheiding van belangen' ook door een aantal eerder besproken architectuurprincipes toegepast:

- Koppelvlakken: scheiding tussen implementatie en interface;
- Gelaagde architectuur: iedere laag heeft een eigen verantwoordelijkheid en dus eigen belangen;
- Zelfomvattende modules: scheiding van belangen op functioneel niveau (het falen of veranderen van een externe module heeft geen invloed op de eigen module);
- Controlesignalen gescheiden van streaming data:
Lage bandbreedte en korte latentie-eis versus grote bandbreedte met relaxte latentie-eis
Door deze te splitsen over verschillende communicatiekanalen met eigen 'Quality of Service (QoS)' is het eenvoudiger om de latentie- en bandbreedte-eisen te garanderen. (Vergelijkbare regels zijn van toepassing op bekabeling in installatie om een enkel punt van falen te voorkomen).

Scheiding van belangen maakt dat een systeem

- Eenvoudiger te begrijpen is;
- Beter te verifiëren is;
- Functionaliteit beter aanwijsbaar is;
- Beter onderhoudbaar is (door clustering van gerelateerde functionaliteit, ofwel cohesie).

Scheiding van belangen is herleid tot specifieke eisen voor bepaalde modules; zie Hoofdstuk 6.

BOTTI#13 Scheiding van belangen: streaming data (audio/video) moet een eigen communicatiekanaal (met onafhankelijke QoS) volgen.

5.2.8 Aanwijsbaarheid

Aanwijsbaarheid slaat op de één op één relatie tussen functionaliteiten en architectuurmodules. Deze is inherent aanwezig in het basisontwerp TTI, waardoor functionele blokken uit de basisspecificatie TTI op basis van naam en relaties direct terug te vinden zijn in de implementatie.

BOTTI#14 Iedere module uit het basisontwerp moet één op één aanwijsbaar zijn in specifieke TTI ontwerpen en TTI implementaties.

Voorbeelden Aanwijsbaarheid uit de TTI: zie 6.3 voor een uitgewerkt voorbeeld.

5.2.9 Communicatiemechanismes

In dit document zijn afhankelijkheden beschreven en wordt verwezen naar de BSTTI, die op zijn beurt beschrijft hoe condities in termen van statusvariabelen, modes en states resulteert in acties en signaleringen. Al deze zaken zijn op vele manier te realiseren, afhankelijk van de gebruikte systemen. Een niet-uitputtende lijst van mogelijkheden, geen van welke verplicht zijn voorgeschreven:

- *polling* (periodieke statusopvraag);
- *publish-subscribe* (notificatie bij relevante wijzigingen);

- *hardwarematige condities* (bijvoorbeeld in PLC-omgeving);
- *interruptmechanismes*.

Het realisatieontwerp dient hier keuzes in te maken en te beschrijven.

5.2.10 Relatie architectuur-eisen en bedrijfs- en beleidsdoelstellingen

Architectuurprincipe in Basisontwerp TTI	Architectuur-kwaliteit	NFE-ID
Decompositie in zelfomvattende modules (BOTTI#5) • lagere kans + lagere impact bij fouten; • betere vervangbaarheid; • betere testbaarheid. Uniformiteit en standaardisatie koppelvlakken • mogelijkheid tot hergebruik van bewezen componenten.	<i>Herbruikbaarheid</i>	2,3,4
Definitie Logische functievervullers (LFV) (BOTTI#25) • voor uniforme en betrouwbare koppeling richting apparatuur.	<i>onderhoudbaarheid</i>	3,4,5,8
Scheiding van belangen (BOTTI#13) • weergavevorm en functionaliteit strikt gescheiden.	<i>uitbreidbaarheid</i> <i>herbruikbaarheid</i>	2,3
Modules met duidelijk gedefinieerde koppelvlakken (BOTTI#1) • eenvoudiger vervangbaarheid van modules; • minimale afhankelijkheid van andere modules; • eenvoudiger vervangbaarheid van modules en lagere complexiteit.	<i>onderhoudbaarheid</i> <i>herbruikbaarheid</i> <i>onderhoudbaarheid</i> <i>uitbreidbaarheid</i> <i>herbruikbaarheid</i>	4,5,8
Aanwijsbaarheid (BOTTI#18) • componenten uit de basisspecificatie TTI ref. [BSTTI] komen 1:1 terug in basisontwerp waardoor functies ook 1:1 aanwijsbaar zijn.	<i>verifieerbaarheid</i>	1,7,9
Gelaagdheid met neerwaartse controle (BOTTI#12) • minimale afhankelijkheid (doordat modules alleen communiceren met naastgelegen lagen).		8
Implementatie volgens geverifieerd basisontwerp.	<i>complexiteit</i>	3,8

De voorgaande tabel toont de belangrijkste architectuureisen in relatie tot architectuurkwaliteiten en beleids- en bedrijfsdoelstellingen zoals gedefinieerd in hoofdstuk 4 Niet-Functionele Eisen. De kolom NFE-ID verwijst naar de ID in Tabel 1.

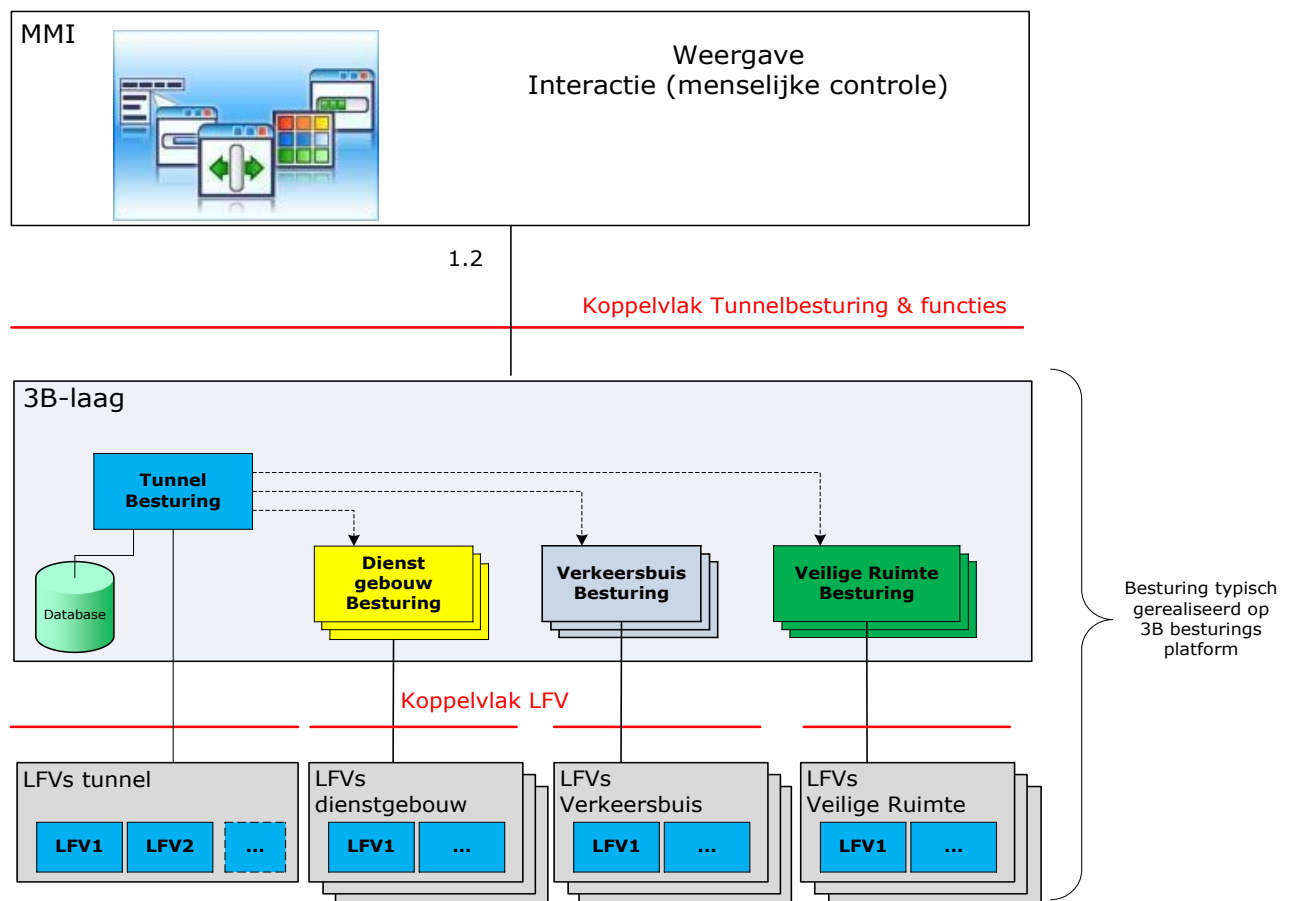
6 Logische Decompositie

Een belangrijke stap in dit basisontwerp is scheiding van belangen. Door een aantal lagen te definiëren met daarin modules en de verantwoordelijkheid van iedere laag en module vast te leggen wordt de complexiteit van het systeem gereduceerd. Een 'belang' dat in een onderliggende laag wordt afgehandeld is gereduceerd tot informatie voor een bovenliggende laag.

Verdere aandachtspunten in de logische decompositie zijn:

- Aanwijsbaarheid: iedere module uit de basisspecificatie is 1-op-1 terug te vinden in deze view;
- Testbaarheid: door duidelijke koppelvlakken en minimale afhankelijkheid tussen gedefinieerde modules moet het eenvoudiger zijn om deze los van elkaar en in samenhang te testen. Dit verkort integratietijden en reduceert risico's.

Logische Functie Vervullers: dit concept, geïntroduceerd in de BSTTI is een belangrijke ontwerpkeuze in het hele concept.



Figuur 8 Logische Decompositie

Voor deze view is het niet belangrijk op welke hardware bepaalde modules gerealiseerd zijn, alleen wat de onderlinge relatie is.

In grote lijnen:

- MMI: 1 of 2 instanties. Bevat gebruikersinterface. In de logische view is er geen verschil tussen locatie (centraal of lokaal). Primaire verantwoordelijkheid: overzicht geven van de tunnelstatus en bediening mogelijk maken;

- 3B: primaire besturing. Gesplitst in besturing op tunnel-, dienstgebouw-, verkeersbuis- en VeiligeRuimte-niveau;
- LfV's: iedere 3B besturingsinstantie is gekoppeld aan een aantal Logische Functie Vervullers. Iedere LfV is verantwoordelijk voor een functie en het aansturen van eventueel benodigde apparatuur.

Onderliggende lagen zijn niet afhankelijk van bovenliggende lagen. Wanneer een bovenliggende laag faalt (bijvoorbeeld door communicatiestoring) dan blijft de functionaliteit van de nog werkende lagen behouden en zorgt daarmee voor maximale beschikbaarheid van actieve veiligheidsvoorzieningen.

LfV's zijn gegroepeerd samen met het besturende proces, zodat deze als zelfomvattende module gerealiseerd kunnen worden, al dan niet samen met de LfV's.

6.1 Logische view: MMI (bediening)

Bediening kan plaatsvinden vanaf een verkeerscentrale of d.m.v. de lokale bediening bij de tunnel. Vanuit de tunnelbesturing gezien maakt het niet uit van waaruit de bediening plaatsvindt (lokaal of centraal) en hoe deze eruit ziet (zie 5.2.7 Scheiding van belangen).

Van de bedieningslocaties kan er 1 actief zijn. Verder is er een plaatselijke bediening in de tunnel(buizen) welke direct in de tunnel ingrijpt op tunnelapparatuur (TTI).

De Centrale/Lokale Bediening:

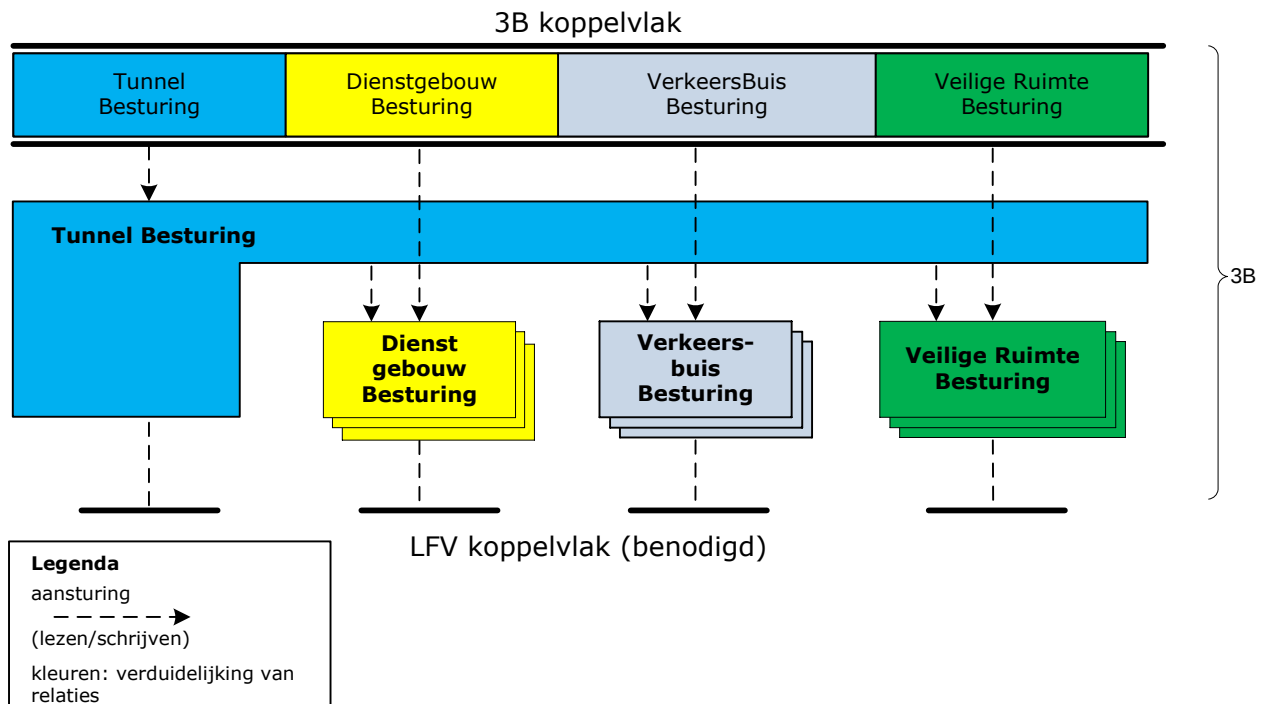
- Geeft de status van in de tunnel aanwezige systemen weer door middel van een MMI;
- Vertaalt bedieningen naar commando's voor de TunnelBesturing (bijvoorbeeld 'sluit tunnelbuis#1 af', 'selecteer beeld Camera#2');

De bediening van een tunnel is hiermee inherent redundant: zelfs indien er een bedieningslocatie of netwerk uitvalt, kan de tunnel nog volledig bediend en bestuurd worden.

De Mens-Machine-Interface (MMI) kan op 2 manieren aan een TunnelBesturing gekoppeld zijn:

- Monitoring (alleen kijken); de MMI kan gebruikt worden om de status van de TunnelBesturing te bekijken, inclusief (niet-besturende) toegang tot sensorfuncties;
- Actief (besturing);
 - acties van de MMI worden door de TunnelBesturing overgenomen;
 - status van TunnelBesturing bekijken;
 - besturende toegang tot sensorfuncties.

6.2 3B-laag



Figuur 9 3B koppervlak naar besturingsmodules

De 3B laag omvat

- Tunnelbesturing (altijd 1 instantie);
- DienstgebouwBesturing (1 of meer; net zoveel als dienstgebouwen bij de tunnel);
- VerkeersbuisBesturing (2 of meer; net zoveel als verkeersbuizen);
- Veilige Ruimte Besturing (0 of meer; net zoveel als veilige ruimtes).

Onderdelen die niet direct aan een verkeersbuis toe te wijzen zijn (zoals waterafvoer, brandblusinstallatie) zijn ondergebracht in 'Tunnel Besturing'. De tunnelbesturing is in Figuur 9 als 'L-vorm' getekend omdat het zowel functionaliteit bevat op een vergelijkbaar niveau als de overige 3B modules (bijv. besturing en controle van LFV's) als ook op een hoger, coördinerend niveau. De 3B modules zijn direct bestuurbaar via het 3B koppervlak in verband met de zelfomvattendheid.

Het 3B koppervlak bevat voor ieder van deze besturingscomponenten ook:

- Alle opdrachten en variabelen zoals gedefinieerd in ref. [BSTTI];
- Alarmen (dit is een lijst met "alarmen", voor de definitie van alarmen zie 3B functies);
- KritischeStoringen (dit is een lijst met "Kritische storing", voor de definitie van Kritische Storingen zie 3B functies);
- Storingen (dit is een lijst met "storingen", voor de definitie zie 3B functies);
- Meldingen lijst (dit is een lijst met meldingen, voor de definitie zie 3B functies);
- Status (dit is een lijst/structuur met relevante statussen van functies en LFV's).

BOTTI#15

Het 3B koppelvlak moet alle opdrachten, variabelen en lijsten realiseren zoals gespecificeerd in ref. [BSTTI].

De 3B laag mag van buiten alleen aangesproken worden via het 3B koppelvlak.

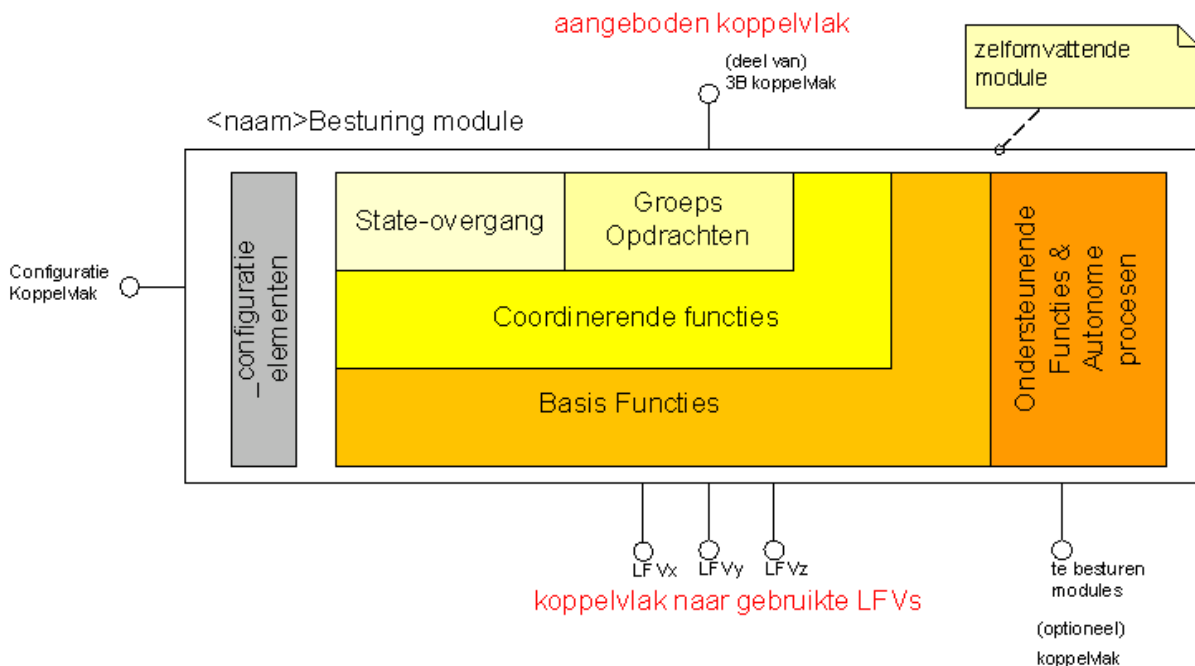
Voor ieder van deze instanties bestaat een 'Besturing' module, zoals de 4 modules in Figuur 9.

Hiernaast is er in de 3B laag configuratiedata aanwezig (in de modules). Hierin zit o.a. de samenstelling van de tunnel (aantal verkeersbuisen, soort veilige ruimtes en de relatie van verkeersbuisen in geval van calamiteit). In gevallen dat coördinatie tussen meerdere besturingsmodules nodig is, dan wordt dit altijd door de TunnelBesturing uitgevoerd. Binnen besturingsmodules is alleen configuratiedata beschikbaar van de betreffende instantie. Dus een verkeersbuis heeft wellicht wel configuratiedata van vluchtdeuren binnen de buis, maar niet over het aantal verkeersbuisen in de tunnel.

De Tunnelbesturing beslaat 2 lagen van 3B:

- Inter-element coördinatie (bijvoorbeeld inter-buis, Verkeersbuis $\leftrightarrow$ VeiligeRuimte);
- Besturing van tunnelinfrastructuur (zoals Vloeistofkelders, brandblusvoorzieningen etc.).

6.3 Structuur van 3B besturingsmodules



Figuur 10 Structuur van 3B besturingsmodule (mapping)

Iedere besturingsmodule binnen 3B is opgebouwd zoals aangegeven in Figuur 10. Voor ieder commando is in de ref. [BSTTI] de soort gedefinieerd.

Dit betekent voor module X (bijv. een ...Besturingsmodule zoals gedefinieerd in Figuur 9):

- **StateOvergang** Bevat de implementatie van alle commando's en variabelen gedefinieerd als STATE-OVERGANG. Bijvoorbeeld: Verkeersbuis.NaarCalamiteit();
- **GroepsOpdrachten** Bevat de implementatie van alle commando's en variabelen gedefinieerd als GROEPSCOMMANDO. Bijvoorbeeld, een commando om alle verlichtingszones in een verkeersbuis op een bepaalde stand te zetten;

- **Coördinerende Functies** Bevat de implementatie van alle commando's en variabelen gedefinieerd als COÖRDINERENDE FUNCTIES. Bijvoorbeeld: HoogteDetectorMonitor;
- **BasisFuncties** Bevat de implementatie van alle commando's en variabelen gedefinieerd als BASIS FUNCTIES;
- **Ondersteunende functies en autonome processen** zijn van/naar iedere laag, en omvatten o.a. bronbeheer en alarmeringsfunctionaliteit. Mogelijkheid tot abonneren op veranderingen of grenswaarde overschrijdingen van variabelen. Hierin worden ook de autonome processen gerealiseerd, evenals de administratie van lijsten met statusinformatie, alarmen etc.;
- **Configuratie-elementen** zijn via een apart koppelvlak beschikbaar, wat gebruikt kan worden om het systeem op te starten in de juiste configuratie. Configuratie-elementen mogen uitsluitend informatie bevatten binnen de context van de module-instantie.

Als er in de ref. [BSTTI] een opdracht is (volgens structuur BSTTI documentatie):

- Verkeersbuis;
 - Basisfuncties;
 - VerkeersbuisLangsVentilatie;
 - Auto().

Dan is deze terug te vinden:

- In module VerkeersbuisBesturing;
- In de laag met Basisfuncties;
- In module VerkeersbuisLangsVentilatie.

Dit maakt de functies aanwijsbaar.

Daarnaast is de TunnelBesturing een aggregatieniveau hoger, waardoor het ook onderliggende (VerkeersbuisBesturing en VeiligeRuimteBesturing) modules kan aanspreken via hun koppelvlakken.

BOTTI#16 3B besturingsmodules moeten geen configuratie-elementen bevatten die zaken buiten de module-instantie beschrijven.

BOTTI#17 Het koppelvlak van iedere 3B besturingsmodule moet aanwijsbaar gespecificeerd zijn in het specifieke gedetailleerde ontwerp van de module.

6.4 Aanwijsbaarheid door structuur (requirements mapping)

Ieder besturingsmodule is volgens dezelfde structuur opgebouwd (zie 6.3).

Deze structuur is exact hetzelfde als de documentstructuur van de specificatie, ref. [BSTTI].

Doordat de plaats in de structuur hetzelfde is tussen specificatie, architectuur en implementatie is alle functionaliteit aanwijsbaar.

Aanwijsbaarheid betekent impliciete traceerbaarheid in 2 richtingen (specificatie → implementatie, implementatie → specificatie).

BOTTI#18 Aanwijsbaarheid: iedere functie moet gerealiseerd worden in de modulestructuur zoals gedefinieerd in het Basisontwerp, paragraaf 6.3 (identiek aan de structuur van de basisspecificatie).

6.5 TunnelBesturing

De TunnelBesturing is het centrale besturende element. Van hieruit worden andere zaken (verkeersbuis, dienstgebouwen en vluchtweg) aangestuurd, bijvoorbeeld voor het uitvoeren van groepscommando's. TunnelBesturing bevat de instantieoverkoepelende besturingen van 3B (bijvoorbeeld zaken waarbij meerdere verkeersbuizen van toestand kunnen veranderen). Een voorbeeld van zo'n inter-buis coördinatie is calamiteitenafhandeling: in een dergelijk geval worden buizen paarsgewijs van mode veranderd. Wanneer de ene buis in Calamiteiten Mode gezet wordt, dan moet TunnelBesturing zorgen dat de bijbehorende andere buis in Calamiteiten Onderdersteunende Mode gezet wordt.

- BOTTI#19* TunnelBesturing moet als zelfomvattende module gerealiseerd worden.
- BOTTI#20* TunnelBesturing moet state-overgangen (besturingsinstantie-overkoepelende functionaliteit) realiseren.
- BOTTI#21* TunnelBesturing moet inter-buis coördinerende functies realiseren.

6.5.1 State-overgang

State-overgangen worden door de MMI aangegeven, of kunnen automatisch binnen de 3B laag gebeuren op basis van detecties. Op tunnelbesturing worden alle toegestane modeovergangen vertaald in de mode van de betreffende verkeersbuizen en vluchtroutes.

voorbeeld Realisatie van koppelvlak (parameters echter nog niet vastgelegd!)

(inter-buis coördinatie)

Tunnel.CalamiteitVolledig(Verkeersbuis[1]))

Wordt vertaald in een aantal opdrachten, waaronder het zetten van de substates voor calamiteit en calamiteitondersteunend voor relevante verkeersbuis en veilige ruimte.

Modeovergangen zijn implementatiespecifiek, en onder andere afhankelijk van relatieve positie van tunnelbuizen, aanwezigheid van een middentunnelkanaal etc.

6.6 VerkeersbuisBesturing

De VerkeersbuisBesturing is verantwoordelijk voor de besturing van precies één verkeersbuis, en onafhankelijk van het functioneren van andere Verkeersbuizen. Het besturingsbereik is precies één verkeersbuis (dus geen coördinatie). Coördinatie tussen verkeersbuizen en overige modules vallen onder 6.5 TunnelBesturing.

- BOTTI#22* VerkeersbuisBesturing moet als zelfomvattende module gerealiseerd worden.

6.7 DienstgebouwBesturing

DienstgebouwBesturing is verantwoordelijk voor de besturing van dienstgebouwen met de daarin opgenomen faciliteiten (zoals communicatie, verlichting, klimaatregeling, alarminstallaties etc.). Het bereik van de besturing is precies 1 dienstgebouw (dus geen coördinerende functies).

- BOTTI#23* DienstgebouwBesturing moet als zelfomvattende module gerealiseerd worden.

6.8 VeiligeRuimteBesturing

VeiligeRuimteBesturing is verantwoordelijk voor het besturen van de veilige ruimte (bijvoorbeeld een middentunnelkanaal) (zoals overdrukmechanismes en evacuatieroutes).

- BOTTI#24* VeiligeRuimteBesturing moet als zelfomvattende module gerealiseerd zijn.

6.9 LfV Laag (Logische FunctieVervuller)

De LfV laag bestaat uit een groot aantal Logische Functie Vervullers, die ieder als module gerealiseerd zijn. Een LfV is verantwoordelijk voor het realiseren van een functie. Daartoe omvat de LfV veelal apparatuur in de tunnel. LfV's zijn onderverdeeld in 4 groepen, zie Figuur 9.

Een LfV bevat een gestandaardiseerd koppelvlak voor een functie vervuller en abstraheert van

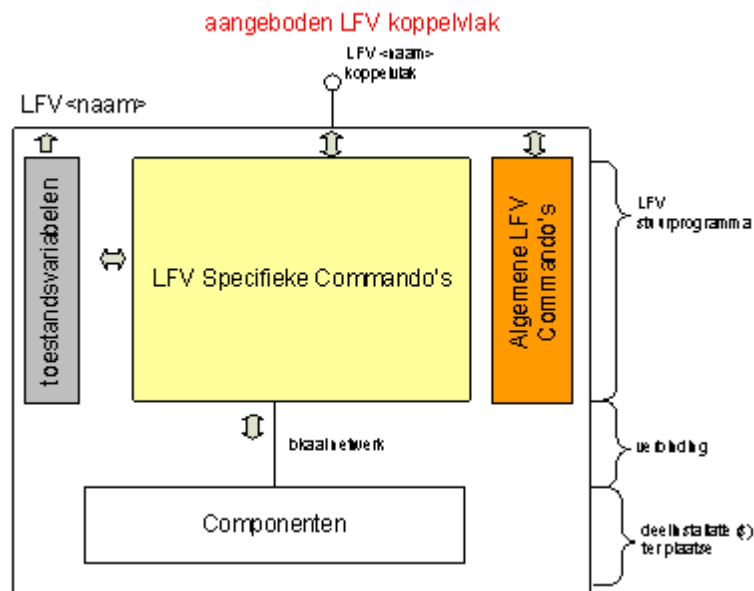
- Type functie vervuller (dus bijvoorbeeld 'Ventilatie' en niet 'n Ventilatoren Merk X type Y');
- Communicatie (zelfs al is communicatie met benodigde apparatuur niet meer mogelijk, toch zal LfV aanspreekbaar blijven en de ontbrekende communicatie als status weergeven).

BOTTI#25 Iedere LfV moet voldoen aan de eisen gesteld in BSTTI (Gemeenschappelijke eisen LfV's).

BOTTI#26 Iedere LfV moet alle LfV-specifieke commando's, variabelen en configuratie-elementen realiseren.

BOTTI#27 Voor iedere LfV moet het koppelvlak aanwijsbaar opgenomen zijn in het gedetailleerde ontwerp.

6.10 Opbouw LfV



Figuur 11 Structuur Logische Functie Vervuller

BOTTI#28 De LfV moet instellingen accepteren van besturing ongeacht de mogelijkheid tot communicatie met de componenten.

Een LfV biedt een LfV koppelvlak aan; dat wil zeggen:

- Data van componenten (bijvoorbeeld sensoren) is als 'toestandsvariabele' extern zichtbaar (om te lezen) zoals gedefinieerd in ref. [BSTTI]. Voorbeeld: #rook_gedetected (Luchtkwaliteitsmeting);
- Set van Commando's zoals gedefinieerd in ref. [BSTTI];
- Algemene LfV functies zoals SetBestuurbaar zijn aanwezig (zie ref. [BSTTI]).

Binnen een LFV is grote vrijheid in de manier waarop de vereiste functionaliteit wordt gerealiseerd. Een LFV bestaat uit het geheel van besturing en bestuurd tunnelapparatuur (componenten). Minimaal een deel van de besturing (namelijk de realisatie van het LFV koppelvlak) vindt plaats in hetzelfde systeem als de 3B software. Dit zorgt er voor dat het LFV koppelvlak minimale afhankelijkheid heeft van externe verbindingen.

Voorbeeld 1 - LFV Ventilatie op basis van een 'intelligent' ventilatie component

Aanname: Ventilatie-subsysteem realiseert alle commando's en status uit de LFV specificatie

LFV: alleen koppelvlak + datacommunicatie

1-op-1 vertaling van commando's en statuscodes

Enige logica: indien communicatie met apparatuur niet (meer) mogelijk is wordt de status

#Bestuurbaar := 'nee'

#reden_niet_bestuurbaar := 'communicatie_uitgevallen'

En indien de communicatie zich herstelt:

Als (geen plaatselijke bediening en niet op niet_bestuurbaar_gezet)

#Bestuurbaar := 'ja'

#reden_niet_bestuurbaar := ''

Anders:

verander #reden_niet_bestuurbaar in nieuwe reden

NB: #reden_niet_bestuurbaar is voor de eenvoud van het voorbeeld gezien als een state; in werkelijkheid kan het een verzameling met redenen zijn.

Voorbeeld 2 - LFV Ventilatie: op basis van losse Ventilator componenten

Koppelvlak: gewenst ventilatieniveau

- besturing: bepaalt gewenste stand van individuele ventilatoren en stuurt deze aan;

- toestandvariabelen: aantal beschikbare ventilatoren (en service status) wordt naar

#capaciteit_beschikbaar vertaald;

- communicatie met diverse ventilatoren (bijvoorbeeld met een bus of netwerk).

Status 'communicatie_uitgevallen' wordt nu gezet indien er geen enkele communicatie met ventilatoren mogelijk is.

Als 50% van de ventilatoren onbereikbaar is, dan vertaalt de LFV dit naar

#capaciteit_beschikbaar=50(%) waarbij de bestuurbaarheid ongewijzigd blijft.

6.11 Veilig falen

De eis tot 'Veilig falen' van individuele onderdelen in een tunnel is een grondslag in de gekozen decompositie in dit hoofdstuk. Het is van belang dat 'veilig falen' in alle delen van de installatie gerealiseerd wordt. Dit betekent o.a. dat elke module op een gedefinieerde manier het wegvallen van communicatie met de rest van het systeem moet kunnen opvangen. Binnen de LFV's is dit al expliciet gemaakt. Waar en hoe het systeem moet terugvallen in standaardgedrag is voor iedere functie gedefinieerd in ref. [BSTTI].