Rapport Proof of Concept  
**Inhoudelijke validaties NLCS++**

**XML-bestanden met Schematron**

Inhoudsopgave

[Inhoudsopgave 2](#_Toc188368949)

[Achtergrond 3](#_Toc188368950)

[Scope 3](#_Toc188368951)

[Uitgangspunten 4](#_Toc188368952)

[Slagingscriteria 4](#_Toc188368953)

[Conclusies 4](#_Toc188368954)

[Aanbevelingen 4](#_Toc188368955)

[Oplevering 5](#_Toc188368956)

[Bijlage A - Bewijs van de inhoudelijke validaties. 6](#_Toc188368957)

# Achtergrond

Vanuit het Registatie aan de Sleuf concept onderzoeken de drie netbeheerders Enexis, Liander en Stedin de mogelijkheden rondom het inhoudelijk valideren van NLCS++ bestanden. Parallel hieraan is Keronic een onderzoek gestart naar de haalbaarheid van inhoudelijke validaties van NLCS++ XML-bestanden met Schematron.

De resultaten van dit onderzoek zijn gedeeld met de drie architecten van de netbeheerders. Hierop is besloten om een Proof of Concept (PoC) op te zetten om de toepasbaarheid vast te stellen.

Dit document bevat de rapportage van deze PoC.

# Scope

De scope van de PoC bestaat uit de onderstaande onderdelen. Onderstaande onderdelen worden opgeleverd aan opdrachtgever door Keronic.

|  |  |
| --- | --- |
| Onderdeel | Omschrijving |
| 1 | 5 uitgewerkte inhoudelijke validaties in Schematron. De specificaties hiervan zijn beschreven in Bijlage A. |
| 2 | NLCS++ voorbeeldbestanden met goed- en foutsituaties. |
| 3 | Setup om de inhoudelijke validaties uit te voeren. Onderdeel van deze setup is een “engine” om Schematron validaties uit te voeren. Hiervoor zal SchXlst worden gebruikt. |
| 4 | De implementatie omvat configuratiebestanden waarin parameters kunnen worden gedefinieerd/aangepast. Bijvoorbeeld: Minimale afstand tussen meetpunten. |
| 5 | Per validatie-fout dient ten minste de volgende informatie opgenomen te zijn:   * Foutcode * Omschrijving van de fout in het Nederlands. * ID van het Asset waar de fout betrekking op heeft. * IDs van de Assets die mogelijk gerelateerd zijn aan de fout. |
| 6 | Documentatie van de implementatie. |
| 7 | Rapport met daarin:   * Bewijs dat oplossing voldoet aan de gestelde specificaties van de PoC * Conclusies * Aanbevelingen * Randvoorwaarden |

# Uitgangspunten

Bij het opstellen van de PoC zijn de onderstaande uitgangspunten vastgesteld.

* Realisatie is gebaseerd op NLCS++ versie V10.
* De oplossing kan systeem agnostisch en decentraal worden uitgevoerd.
* In de definitie van NLCS++ zijn een DWG-bestand en een XML-bestand opgenomen. Het DWG-bestand zal buiten beschouwing worden gelaten. Voorgenoemd uitgangspunt is onder andere de reden voor dit uitgangspunt.
* Doelstelling is om zoveel mogelijk te uniformeren tussen de netbeheerders.

# Slagingscriteria

De PoC wordt beschouwd als geslaagd wanneer de specifieke inhoudelijke validaties kunnen worden gerealiseerd met behulp van Schematron. De werking daarvan wordt aangetoond door middel van de NLCS++ XML-voorbeeldbestanden met goed- en foutsituaties.

# Conclusies

De gevraagde inhoudelijke validaties (5) zijn ontwikkeld met behulp van Schematron. Bijlage A biedt het bewijs hiervoor. Hiermee is aangetoond dat de inhoudelijke validaties van NLCS++ bestanden kan worden uitgevoerd met Schematron.

XSL kan worden gezien als een programmeertaal. Dit biedt veel mogelijkheden. Tijdens de ontwikkeling van de PoC zijn er een aanzienlijk aantal ruimtelijke functies ontwikkeld om onder andere ruimtelijke relaties tussen geometrieën te kunne vaststellen.

Enige programmeerervaring is nodig om de inhoudelijke validaties te kunnen onderhouden. De validaties bevatten weinig complexe logica en hebben vaak een vergelijkbare structuur. Schematron/XSL heeft enige leercurve.

# Aanbevelingen

Allereerst is het belangrijk om de definitie van de validaties duidelijk vast te leggen om ambiguïteit te voorkomen. Verder dienen de inhoudelijke validatie te worden getoetst op het proces. Vooral voor de softwareleveranciers aan aannemerszijde.

Inhoudelijke validaties hebben een directe relatie met de structuur (XSD) van een NLCS++ bestand. Het is daarom logisch om beide als 1 geheel te zien, te versioneren en te releasen. Verder zal uniformering tussen netbeheerders bijdragen aan de onderhoudbaarheid van de algehele standaard.

Tot slot zal kritisch gekeken moeten worden naar de verplichtheid van attributen in de XSD. Het succes van de inhoudelijke validaties is hier afhankelijk van.

# Oplevering

Onderstaande tabel beschrijft hoe de onderdelen uit paragraaf Scope zijn opgeleverd. Het merendeel van de onderdelen is opgenomen in een private Git repository op GitHub. Uitnodiging voor toegang tot deze repository is vereist.

|  |  |
| --- | --- |
| Onderdeel | Oplevering |
| 1 | De uitgewerkte inhoudelijke validaties zijn beschikbaar in onderstaande Git repository. |
| 2 | De Git repository bevat bestanden met goed- en foutsituaties. Zie onderstaand de link. In bijlage A zijn deze tevens beschreven. |
| 3 | De Git repository bevat een README.md waarin instructies zijn gegeven voor het uitvoeren van de inhoudelijke validaties. |
| 4 | De Git repository bevat een configuratiebestand config.xml. Werking is beschreven in documentatie. |
| 5 | Elke inhoudelijke validatie heeft een message waarin de gevraagde informatie is opgenomen. |
| 6 | Documentatie is opgenomen in de documentation map van de Git Repository. |
| 7 | Rapport betreft dit document. |

# Bijlage A - Bewijs van de inhoudelijke validaties.

|  |  |
| --- | --- |
| Inhoudelijke validatie #1 | |
| Asset | MSkabel |
| Categorie | Topologie |
| Omschrijving | Controleer of de MSkabel is verbonden met twee assets.  MSkabel objecten met Status "BESTAAND", "BESTAAND RESERVE", "NIEUW", "NIEUW RESERVE", "REVISIE" of "REVISIE RESERVE" mogen aan hun eindes niet losliggen; ze moeten altijd een valide eindpunt hebben. Valide eindpunten zijn: MSmof, MSstation, MSoverdrachtspunt. |

|  |  |
| --- | --- |
| Voorbeeld slagende validatie | |
| Source bestand | poc\_passing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | Dit voorbeeld laat zien dat de structuur validatie slaagt als een MSkabel exact 2 verbindingen heeft. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld falende validatie | |
| Source bestand | poc\_failing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als een MSkabel niet exact 2 verbindingen heeft. |

Resultaat:

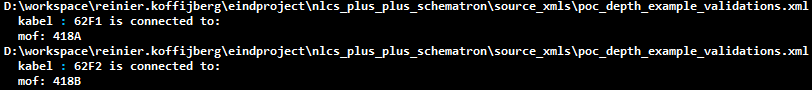
A screen shot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Inhoudelijke validatie #2 | |
| Asset | LSkabel |
| Categorie | Diepte |
| Omschrijving | Controleer of de diepteligging van een LSkabel correct is geregistreerd.  Er moet onderscheid gemaakt kunnen worden tussen kabels die op elkaar liggen op basis van onder andere de diepteligging.  Als een LSkabel niet als bovengronds is geregistreerd, moet de diepteligging altijd negatief zijn.  Als er een diepte geregistreerd is moet er ook een Amaaiveldhoogte zijn geregistreerd, waarmee de diepte relatief aan de Amaaiveldhoogte kan worden afgeleid.  Een Amaaiveldhoogte wordt gerelateerd aan een LSkabel wanneer deze zich binnen 5 meter van LSkabel geometrie bevindt.  Voor verschillende types kabels moet er een minimale en maximale dieptegrens kunnen worden ingesteld. Voor dit voorbeeld gaan wij uit van: LSkabel moet tussen 60 en 120 cm diepte liggen. |

|  |  |
| --- | --- |
| Test voorbeeld van verbondenheid | |
| Source bestand | poc\_depth\_example\_validations.xml |
| Fase | Depth\_example |
| Toelichting | Dit voorbeeld laat zien dat de validatie slaagt als een aftak mof verbonden is met 3 kabels.  Dit is een valide XML, maar foutmeldingen zijn toegevoegd om aan te tonen dat twee kabels en twee moffen die boven elkaar liggen, slechts alleen aan de elementen op dezelfde diepte verbonden worden. Dit is alleen voor demo-doeleinden geschreven. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld slagende validatie #1 | |
| Source bestand | poc\_passing\_depth\_validation.xml |
| Fase | Depth\_structure |
| Toelichting | Dit voorbeeld laat zien dat de structuur validatie slaagt volgens de regels beschreven in de validatie-omschrijving. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld slagende validatie #2 | |
| Source bestand | poc\_passing\_depth\_validation.xml |
| Fase | Depth\_complex |
| Toelichting | Dit voorbeeld laat zien dat de complexe validatie slaagt volgens de regels beschreven in de validatie-omschrijving. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld falende validatie #1 | |
| Source bestand | poc\_failing\_depth\_validation.xml |
| Fase | Depth\_structure |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als een meetpunt van een kabel niet aan een maaiveld verbonden is. |

Resultaat:

A screenshot of a computer program

Description automatically generated

|  |  |
| --- | --- |
| Voorbeeld falende validatie #2 | |
| source bestand | poc\_failing\_depth\_validation.xml |
| Fase | Depth\_complex |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als een LSkabel een incorrecte diepteligging heeft. Een incorrecte diepteligging is hier gedefinieerd als > 1.2 of < 0.6 meter. |

Resultaat:

A computer code on a black background

Description automatically generated

|  |  |
| --- | --- |
| Voorbeeld falende validatie #3 | |
| source bestand | poc\_failing\_depth\_validation.xml |
| Fase | Depth\_complex |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als het ‘Bovengronds’ attribuut niet overeenkomt met de geregistreerde diepte van een LSkabel. |

Resultaat:

A screenshot of a computer screen

Description automatically generated

|  |  |
| --- | --- |
| Inhoudelijke validatie #3 | |
| Asset | LSmof – Functie: AFTAK |
| Categorie | Topologie en Attributen |
| Omschrijving | Controleer of de bedrijfsspanning van nieuw gelegde laagspanningskabels overeenkomt met de verbonden LS Aftak mof.  De bedrijfsspanning (attribuut Spanningsniveau) van de 3 LSkabel (status "NIEUW") objecten die verbonden zijn met de LS Aftak mof dienen gelijk te zijn. |

|  |  |
| --- | --- |
| Voorbeeld slagende validatie #1 | |
| Source bestand | poc\_passing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | Dit voorbeeld laat zien dat de structuur validatie slaagt als een aftak mof verbonden is met 3 kabels. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld slagende validatie #2 | |
| Source bestand | poc\_passing\_complex\_validations.xml |
| Fase | Complex |
| Toelichting | Dit voorbeeld laat zien dat de complexe validatie slaagt als een aftak mof alleen verbonden is met kabels met dezelfde spanning. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld falende validatie #1 | |
| Source bestand | poc\_failing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als er niet exact 0 of 3 kabels verbonden zijn met een aftak mof. |

Resultaat:

A black screen with white text

Description automatically generated

|  |  |
| --- | --- |
| Voorbeeld falende validatie #2 | |
| Source bestand | poc\_failing\_complex\_validations.xml |
| Fase | Complex |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als de spanning van kabels die verbonden zijn met een aftak mof niet hetzelfde zijn. |

Resultaat:

A computer screen with white text

Description automatically generated

|  |  |
| --- | --- |
| Inhoudelijke validatie #4 | |
| Asset | LSmof – Functie OVERGANG 3>1 Fase |
| Categorie | Topologie en Attributen |
| Omschrijving | Bij 3>1 fase: 3-fase aan één kant, L1,2,3 aan andere kant.  Controleer bij een 3-fase naar 1-fase overgang of aan één kant van de LSmof (status "NIEUW", "REVISIE") de drie afzonderlijke fase-kabels (L1, L2, L3) met status "NIEUW", "REVISIE" aanwezig zijn en aan de andere kant slechts één 3-fase-kabel.  N.B. Op dit moment ontbreekt in het NLCS++ datamodel een attribuut waaruit we kunnen afleiden dat het om een 3 > 1 Fase overgang gaat. Hiervoor zullen we de keuzelijst FunctieLSMof uitbreiden met de waarde "OVERGANG 3>1 FASE". |

|  |  |
| --- | --- |
| Voorbeeld slagende validatie #1 | |
| Source bestand | poc\_passing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | Dit voorbeeld laat zien dat de structuur validatie slaagt als een 3>1 fase mof verbonden is met 4 kabels. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld slagende validatie #2 | |
| Source bestand | poc\_passing\_complex\_validations.xml |
| Fase | Complex |
| Toelichting | Dit voorbeeld laat zien dat de complexe validatie slaagt als een 3>1 fase mof de correcte fase configuratie heeft. |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld falende validatie #1 | |
| Source bestand | poc\_failing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als er niet exact 0 of 4 kabels verbonden zijn met een 3>1 fase mof. |

Resultaat:

A computer screen with white text

Description automatically generated

|  |  |
| --- | --- |
| Voorbeeld falende validatie #2 | |
| Source bestand | poc\_failing\_complex\_validations.xml |
| Fase | Complex |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien als de fase configuratie van een 3>1 mof incorrect is. |

Resultaat:

A black screen with white text

Description automatically generated

|  |  |
| --- | --- |
| Inhoudelijke validatie #5 | |
| Asset | E Aarddraad |
| Categorie | Geometrie en Topologie |
| Omschrijving | Controleer of de aarddraad verbonden is met een gebouw.  Wij gaan er bij deze inhoudelijke validatie van uit dat een gebouw een MSstation of LSkast is. Een EAarddraad object dient te eindigen op de rand van een MSstation of LSkast. |

|  |  |
| --- | --- |
| Voorbeeld slagende validatie | |
| Source bestand | poc\_passing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | - |

Resultaat:



|  |  |
| --- | --- |
| Voorbeeld falende validatie | |
| Source bestand | poc\_failing\_structure\_validations.xml |
| Fase | Structuur |
| Toelichting | Dit voorbeeld laat de vorm van foutmeldingen zien indien validatie #5 faalt. |

Resultaat:

A screen shot of a computer

Description automatically generated