

INNO Verantwoordings document

Martijn Voorwinden (MV) 1776622

Versie 2.0 / 2022-11-14

WIJZIGINGEN

Herziening	Datum	Auteur(s)	Beschrijving
1.0	2022-10-30	MV	Bestand aangemaakt
1.1	2022-11-01	MV	Aangepast na review
2.0	2022-11-14	MV	Uitbreiding gebruikte technologieën, code structuur

INHOUDSOPGAVE

1	Opdracht	3
2	Gebruikte technologieën	4
2.1	Microsoft purview	4
2.2	Elm	4
2.3	Python	5
3	Code structuur	6
3.1	Opdeling in modules (Elm)	6
3.2	Code stijl (Elm)	6
3.3	Elm-Review	7

OPDRACHT

Voor provincie Utrecht moet er een proof of concept (PoC) van een datacatalogus gemaakt worden zodat de provincie aan de hand hiervan een potentiële aanbesteding te doen. Een datacatalogus zal er voor kunnen zorgen dat het makkelijker is om de medewerkers toegang te geven tot de verschillende databronnen in de verschillende afdelingen van de provincie. Dit is nodig om te zorgen dat er geen dubbele datasets worden aangeschaft en om de zoektijd naar relevante datasets te verkleinen.

GEBRUIKTE TECHNOLOGIËN

Er is besloten om de data catalogus te maken in microsoft purview en een web front-end in elm, om dit te faciliteren en credentials te verbergen zal er in python middleware gemaakt worden.

2.1 Microsoft purview

Microsoft purview is een data management systeem wat onder andere gebruikt kan worden om een data catalogus op te zetten en in verscheidene databronnen te zoeken [1]. Er is voor microsoft purview gekozen omdat de Provincie Utrecht al met veel microsoft producten werkten en dus het koppelen met andere interne producten makkelijker is.

Blue Dolphin Data

Een potentieel alternatief voor microsoft purview zou de blue dolphin data module [2] kunnen zijn, hier is van af gezien omdat hierbij een boel voor het project onnodige tools zijn inbegrepen waar wel voor betaald zou moeten worden. Voor het deel waar het project gebruik van maakt zou het vrijwel hetzelfde als microsoft purview werken en geen meerwaarde bieden.

2.2 Elm

Elm is een functionele programmeertaal die compileert naar JavaScript en als een alternatief voor JavaScript gebruikt kan worden [3]. Er is voor elm gekozen omdat het sterke getypeerd is en een rigide compiler heeft, dit zorgt ervoor dat er geen runtime errors zijn, maar alle errors op compile time worden gegoooid. Elm is ook een technologie die het team graag wil leren.

Purescript

Purescript is een potentieel alternatief voor elm, het lijkt erg op elm omdat ze beide zijn gebaseerd op haskell en gecompileerd kunnen worden naar JavaScript [4], er is voor elm gekozen inplaats van Purescript omdat er meer documentatie over elm te vinden is.

Typescript

Typescript is een superset van JavaScript welke de mogelijkheid biedt om code in JavaScript statisch getypeerd te maken [5]. Er is voor elm in plaats van Typescript gekozen omdat elm het statisch typeren verplicht in tegenstelling tot Typescript waar het optioneel is. Dit zorgt er voor dat bepaalde fouten wel in Typescript gemaakt kunnen worden, en niet in elm omdat in het geval van elm ze er al door de compiler uit gehaald worden.

2.3 Python

Python is een general-purpose programmeertaal taal die goed ondersteund wordt door microsoft producten [6][7], omdat microsoft purview een relatief nieuw product is, is er op het moment alleen een SDK voor python, om deze reden is er voor gekozen om python te gebruiken voor het maken van de middleware.

CODE STRUCTUUR

3.1 Opdeling in modules (Elm)

Om de code overzichtelijk en onderhoudbaar te houden moet het opgedeeld worden in verschillende modules, de plaats waar code thuishoort in elm wordt bepaald door 3 heuristieken:

- Unique.
- Similar.
- The Same.

Deze bepalen waar en wanneer er helper functies gemaakt moeten worden, als er meerdere helper functies over hetzelfde type ontstaan kan dit type met de bijbehorende helper functies naar een aparte module verplaatst worden [8].

3.2 Code stijl (Elm)

Om de code makkelijk overleverbaar te maken moet er gebruik gemaakt worden van een gestandaardiseerde stijl, daarom wordt gebruik gemaakt van de geadviseerde stijl uit de officiële elm documentatie [9]. De geadviseerde stijl is als volgt:

- Regels zijn niet langer dan 80 tekens.
- Functies en variabelen hebben een beschrijvende naam.
- Een declaratie heeft een type annotatie.
- Er zijn 2 lege regels tussen declaraties.
- De body van een declaratie begint 1 regel onder de declaratie.
- Er is geen overbodige indentatie.

3.3 Elm-Review

Elm-Review is een elm package welke in essentie een linter is, dit kan gebruikt worden om de code aan bepaalde eisen te laten voldoen [10]. Voor het project is ook Elm-Review gebruikt [11], deze is ingesteld met de volgende regels:

1. NoConfusingPrefixOperator.rule
2. NoDebug.*.rule
3. NoMissing*.rule
4. NoSimpleLetBody.rule
5. NoPrematureLetComputation.rule
6. NoUnused.*.rule
7. NoForbiddenWords.rule ["@TODO"]

Deze regels zijn gebaseerd op de elm application template van elm-review [12].

BIBLIOGRAFIE

- [1] *Unified Data Governance with Microsoft Purview: Microsoft Azure*. URL: <https://www.microsoft.com/nl-nl/security/business/microsoft-purview>.
- [2] *Data management - valueblue*. URL: <https://valueblue.com/en/uc/data-management/>.
- [3] Czaplicki, Evan. *Elm*. URL: <https://elm-lang.org/>.
- [4] Freeman, Phil. *purescript*. URL: <https://www.purescript.org/>.
- [5] *TypeScript*. URL: <https://www.typescriptlang.org/>.
- [6] *Python*. URL: <https://www.python.org/>.
- [7] *Purview Python SDK*. URL: <https://learn.microsoft.com/en-us/azure/purview/tutorial-using-python-sdk>.
- [8] Czaplicki, Evan. *Modules*. URL: <https://guide.elm-lang.org/webapps/modules.html>.
- [9] Czaplicki, Evan. *Style guide*. URL: <https://elm-lang.org/docs/style-guide>.
- [10] Mengels, JF. *elm-review*. URL: <https://package.elm-lang.org/packages/jfmengels/elm-review/latest/>.
- [11] Voorwinden, Martijn. *Project repo - elm-review*. URL: <https://github.com/MartijnCBV/INNO/tree/master/frontend/review>.
- [12] Mengels, JF. *application template - elm-review*. URL: <https://github.com/jfmengels/elm-review-config/tree/master/application>.