Over de leveringsvoorwaardenservice: ik heb een paar vragen die wel heel bepalend zijn voor de oplossing die we gaan bouwen

**Milan Kaihatu**

[10:29 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/3c6fnn7wgtyzzxfqraa6gnhpwc)

ok steek van wal

**Adriaan Bakker**

[10:29 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/nq8dc8iq8iyipqkf666ff1c9qy)

Het leek mij bij nader inzien niet zo heel handig om maar alleen met een excel sheet te starten.

Probleem daarmee is tweeledig: de consistentie van de gegevens is onvoldoende gegarandeerd en ook kan niet goed worden gecontroleerd of de afnemers wel bestaan, de paden naar de gegevens wel de juiste zijn, of die gegevens wel voorkomen in de betreffende berichten enzovoorts.

Mijn voorstel zou zijn een relationele database te gebruiken en een inleesprogramma vanuit een excel sheet. Dat inleesprogramma gebruikt dan tevens de bestanden met SOAP berichtendefinities om een en ander te controleren

**Milan Kaihatu**

[10:32 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/sxibjrse3ffybm6g7fow4bgb7c)

eerst eens zien wat we in een excel zetten: daar zet je de regels in op basis waarvan data benaderd mag worden. uit een contextuele service ontdek je of de afnemer bestaat

kijk ff mee naar slide 6

**illustratie leveringsvoorwaardenser...**PPTX1MB

in de PRP zie ik een database waar de voorwaarden in staan

**Adriaan Bakker**

[10:35 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/wf18y1kd3ir78gmmtnzpdzbc4h)

**1**

Doen we het dan eerst zonder relationele database? Je wilt toch dat de leveringsvoorwaarden ook vanuit andere applicaties benaderd kunnen worden. Dan ligt een database meer voor de hand. Daar moet ook beheer op zijn en je wilt zeker weten dat er geen leveringsvoorwaarden worden gebruikt die niet kloppen.

**Milan Kaihatu**

[10:35 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/c4fodmwo97yo9c4kg1tzc1tz8r)

misschien moet dat attributen lijntje van gateway naar pip lopen

**Milan Kaihatu**

[10:36 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/yu5bzi4osibiuf99dx7szbk4za)

**1**

Gaf commentaaar op het bericht van Adriaan Bakker: Doen we het dan eerst zonder relationele database? Je wilt toch dat de leveringsvoorwaarden ook vanuit andere applicaties benaderd kunnen worden. Dan ligt een database meer voor de hand. Daar moet ook beheer op zijn en je wilt zeker weten dat er geen leveringsvoorwaarden worden gebruikt die niet kloppen.

als volgens jou een relationele database de mvp is :slightly\_smiling\_face:

Nieuwe Berichten

**Milan Kaihatu**

[10:38 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/e6k3pqd6ejfztgu71finsj1s5r)

kan je een excelsheet (of een .csv) verderop in de tijd niet makkelijk omzetten naar een relationeel model?

wat ik ook wel belangrijk vind is dat áls er een product in de markt is voor een xacml policy repository point dat we dat wel moeten evalueren

**Adriaan Bakker**

[10:41 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/98qxii39e78zpbt8pjatzbqk7h)

Jawel, alles kan, maar de vraag is wat wijsheid is op dit moment. Ik vind een relationele database geloofwaardiger, die kan in allerlei constraints voorzien en ervoor zorgen dat data niet gecorrumpeerd raakt. Redenen: De leveringsvoorwaardenservice is een single point of failure bij alle gegevensleveranties en moet dus robuust en betrouwbaar zijn 2. het is niet bijzonder moeilijk om een relationele database te gebruiken, bijvoorbeeld PostGres, vanuit een spring boot java applicatie 3. de leveringsvoorwaarden moeten real time zijn te onderhouden en dat impliceert dat de laatste versie moet kunnen worden gebruikt, bij een relationele database worden wijzigingen zodanig doorgevoerd dat applicaties daar niets van merken die van die data gebruik maken

VEEL liever een relationele database dan een excel sheet dus.

**Milan Kaihatu**

[10:43 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/kwzjgnfdw3dzjfdq5di5uuynkc)

ah dan gaat het minder om het relationeel zijn van de tabellen, maar meer om het securen van de regels

**Adriaan Bakker**

[10:43 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/3kwnfpgpfpbcigwnj8siecbtoh)

Ja

Stel je wijzigt de regels en je maakt alleen gebruik van het excel sheet. Moet dan de leveringsvoorwaarden de ingelezen regels in geheugen houden? Dat betekent dat op dat moment alle regels opnieuw moeten worden ingelezen. Dus ofwel de microservice opnieuw starten zodat de regels opnieuw worden ingelezen of via een trigger de microservice bewegen om dit te doen. Op dat moment is de leveringsvoorwaardenservice niet te benaderen.

Omslachtig, foutgevoelig.

**Milan Kaihatu**

[10:46 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/6wnu4uyuefrpzq9ypicdohouje)

ja je wil realtime per request een besluit kunnen nemen

**Adriaan Bakker**

[10:47 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/dfai9f7tm3yidn4mzs4pxn7dph)

Je zou natuurlijk voor ieder request de regels opnieuw kunnen laten inlezen van het excel sheet maar dat geeft ook problemen. Er zou een fout kunnen zitten in het aangepaste sheet. Wat doe je dan? leg je dan de leveringsvoorwaardenservice tijdelijk stil zodat je de fout kunt herstellen?

In alle opzichten is een relationele database veel beter!

**Milan Kaihatu**

[10:48 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/w4x4yji9x3robmewdet5kgfyur)

een dbms bedoel je he

**Adriaan Bakker**

[10:48 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/dxo6dsypztb4dd9ukwrf1awq4y)

Ja

Mag ook best een tool uit de markt zijn maar dat hebben we dus nog niet.

**Milan Kaihatu**

[10:49 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/4iab14mn3jf9uqq9xqnm5xptwe)

dat zou echt mijn voorkeur hebben

**Adriaan Bakker**

[10:49 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/48d9umtxcidg3kwncws39wifne)

Ja hoor, mits het kan doen wat we ervan vragen. In dit geval: de gegevens voorzien van paden koppelen aan voorwaarden.

en die paden controleren vanuit de geleverde berichten

onderhoudbaar

beveiligd

hoge performance

constraints kunnen controleren

**Milan Kaihatu**

[10:50 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/448x115saib88e9x5mnip3tuar)

ja

axiomatics is de grote naam op het gebied van policy repositories

**Adriaan Bakker**

[10:53 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/6c1py8auzjb5ucipxqk57r8dqh)

Ik zit met de volgende vraag: Is het dan te vroeg om een voorlopige oplossing voor te stellen als die ik heb voorgesteld met die relationele database en moeten we ons eerst verdiepen in de oplossing(en) van Axiomatics? Of is het voor deze pilot een goede aanzet om zelf een robuuste oplossing te bouwen en daarvan te leren om dan alsnog voor een oplossing uit de markt te opteren?

**Milan Kaihatu**

[10:53 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/ewsixonkjbfojqgoqwwtqwupbc)

misschien soort twee sporen volgen

ook even overleggen met rj

**Adriaan Bakker**

[10:54 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/oqc88z8cbjb7ir5cqofcm9w8ta)

Okee, laten we dat met hem doornemen!



**Milan Kaihatu**

[10:54 a.m.](https://gitlab.bkwi.nl:8443/ts/pl/1jstbdfw9f8njfksx9tdxenfzc)

we kunnen wel door met het opstellen van requirements

in dit gesprek kwamen er al een paar naar boven!

denk dat de volgorde is: requirements -> oplossing zoeken in de markt -> als geen marktoplossing, dan bouwen, anders marktoplossing poc

buy before build

ik moet zo ff met de architecten in overleg, dan ontdek ik ook of willem er deze week weer bij is