



Challenge Brief

Project:

Challenge Owner:

Belastingdienst & PGGM

Hoe maken we complexe wet- en
regelgeving inzichtelijk,
uitvoerbaar en uitlegbaar met
Rules as Code?

Achtergrond:

De pensioenwetgeving rond het keuzerecht “*bedrag ineens*” maakt het mogelijk om bij pensionering maximaal 10% van het ouderdomspensioen in één keer op te nemen.

Wat deze keuze complex maakt, is dat een bedrag ineens kan leiden tot:

- hogere belasting in het uitkeringsjaar (door progressieve tarieven)
- verlies of terugvordering van inkomensafhankelijke regelingen en toeslagen

Door de samenloop van pensioenregels, belastingwetgeving en lokale regelingen is het effect voor burgers vaak moeilijk te overzien.

Probleemstelling:

Hoewel “*bedrag ineens*” op papier simpel klinkt (“tot 10% opnemen”), is het in de praktijk voor burgers lastig om:

- het effect op het netto besteedbaar inkomen van het huishouden te begrijpen
- te zien hoe *bedrag ineens* zich verhoudt tot andere keuzes, zoals hoog/laag pensioen of regelingen als de AIO
- te overzien welke landelijke of lokale toeslagen onder druk komen te staan
- te bepalen wat een verstandige keuze is: 0%, 5%, 10%, hoog-laag of een combinatie
- te begrijpen **waarom** een bepaalde uitkomst ontstaat (welke regels en parameters spelen een rol)

Er is daarom behoefte aan een tool die inzichtelijk maakt hoe regels samenkommen én scenario's onderling vergelijkbaar maakt.

Challenge: **Hoe kunnen we burgers (en adviseurs/uitvoerders) helpen begrijpen of het verstandig is om met bedrag ineens (0-10%) op te nemen, door wet- en regelgeving te modelleren als Rules as Code met gestandaardiseerde input én een uitlegbare visualisatie van de effecten, in plaats van losse tekst en moeilijk te interpreteren rekenregels?**

De tool moet de gebruiker helpen om:

1. **Scenario's door te rekenen (0%, X%, 10%)**
2. **Impact uit te leggen** (belasting/ toeslagen/ voorzieningen)
3. Een **begrijpelijk advies-achtig inzicht** te geven (“dit lijkt gunstig / risicovol” + waarom)

⚠ Let op: het gaat om **keuze-ondersteuning en transparantie**, niet om juridisch/ financieel bindend advies.

Sustainable development goals (SDG's):



Criteria

- Explainability by design: zichtbaar welke regels tot welke impact leiden
- Scenariovergelijking: 0% vs. 10% vs. hoog-laag (incl. netto-effect)
- Gestandaardiseerde input: duidelijk en herbruikbaar datamodel
- Visualisatie: beslisboom, flow of impact breakdown
- Robuustheid: begrijpelijk voor verschillende profielen

Bonuspunten

- "Wat-als" sliders en gevoelighedsanalyse
- Rules-as-Code los van UI en engine
- Duidelijke waarschuwingen over onzekerheid en aannames

Data:

Voor deze challenge is een open source starter repository beschikbaar met gestructureerde invoerdata.

🔗 <https://github.com/mathisvfr/rules-as-code-pension-starter>

De repo bevat fictieve pensioenprofielen in YAML-formaat, bedoeld om **wet-** en **regelgeving als code** te modelleren en scenario's (0-10%) inzichtelijk te maken.

Resources en inspiratie:

- [**Rijksoverheid**](#) over "bedrag ineens" (stand van zaken + uitleg)
- [**Eerste Kamer dossier**](#) "Wet herziening bedrag ineens" (proces + info over inwerkingtreding)
- [**Kamerstukken/onderzoek**](#) met tabellen over effecten op besteedbaar inkomen en toeslagen (handig als inspiratie voor output vormen)
- [**Debat fragment/verslag**](#) Tweede Kamer over toeslagen risico's (context waarom explainability belangrijk is)
- [**Regelrecht**](#) overzicht van open-source projecten en experimenten rond Rules as Code binnen de Nederlandse overheid. Laat zien hoe wet- en regelgeving kan worden vastgelegd in machineleesbare regels, losgekoppeld van implementatie en gebruikersinterface. Relevant als inspiratie voor regel-representatie, herbruikbaarheid en transparantie.

Wat je oplevert:

- Een werkend prototype (of zeer concreet PoC) van:
 - rules engine / rule representation
 - gestandaardiseerde input
 - uitleg/ visualisatie + scenario vergelijking
- Een korte demo met meerdere voorbeeldprofielen uit de dataset
- Documentatie: aannames, beperkingen, en hoe dit kan doorontwikkelen
- Een presentatie over het prototype en jullie visie

DISCLAIMER: ...

Casushouder: Abram Klop, Ministerie BZK & Martijn Roelfsema, PGGM

Hackathon Toolkit

Tools die jij kunt gebruiken tijdens de Hackathon. Vraag gerust om hulp aan de crew.

Brainstorm & samenwerken

- Miro - digitale whiteboard voor brainstorms & flowcharts
👉 <https://miro.com>
- Jamboard (alternatief) - eenvoudig whiteboard van Google
👉 <https://jamboard.google.com>

Design & prototyping

- Figma - ontwerp je app, interface of dashboard
👉 <https://figma.com>
- Canva - voor posters, logo's, visuals, presentaties
👉 <https://canva.com>
- Draw.io - flowcharts, systemen en logica tekenen
👉 <https://app.diagrams.net>

Technologie & code

- GreenPT API (*via hackathon partner in de Hackerpack*)
👉 Privacy-first AI API, volledig EU-gehost en op open-source modellen.
- GitHub - versiebeheer en samenwerking voor open source code
👉 <https://github.com>
- Jupyter Notebook / JupyterLite
👉 <https://jupyter.org>

Presentatie & pitch

- Google Slides / Canva / PowerPoint - bouw je pitch

💡 Tip:

Werk samen in één tool per team (bijvoorbeeld alles in Figma of Canva), en zorg dat jullie bestand makkelijk te delen is voor de pitch. Hetzelfde geldt voor je code, zorg dat je repository in te zien is door juryleden en creëer een duidelijke README.