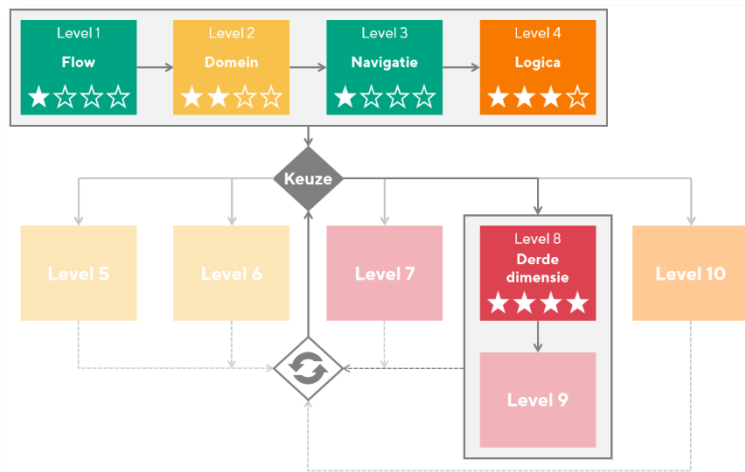


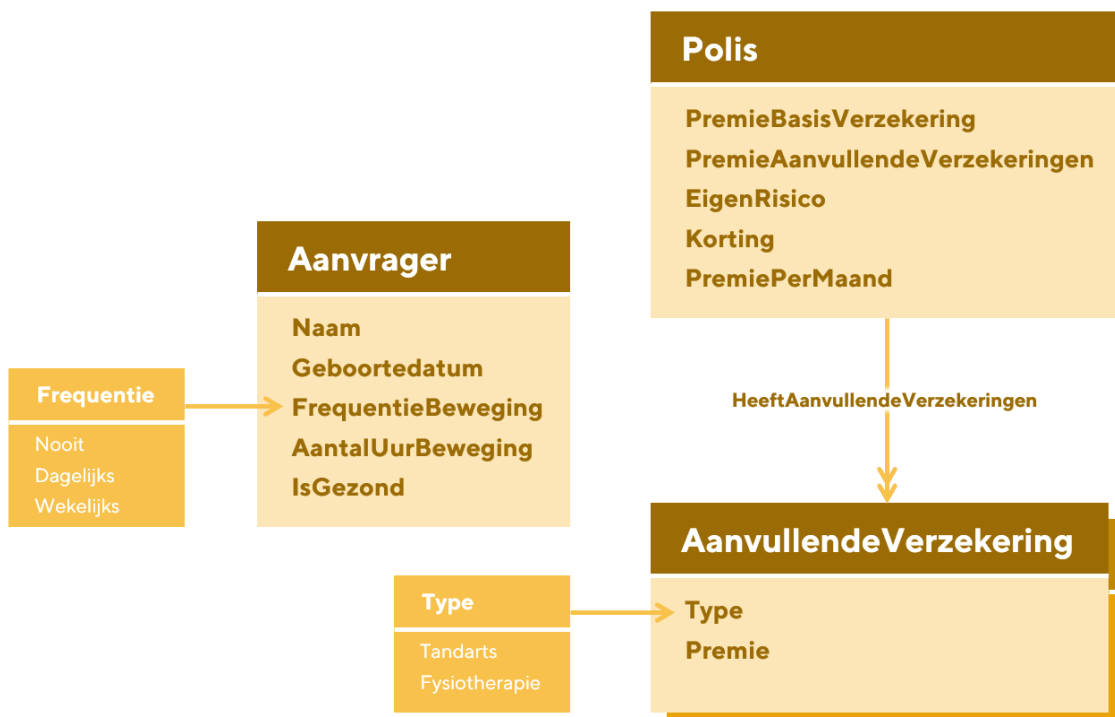
LEVEL 8: DE DERDE DIMENSIE

NIVEAU: ★ ★ ★ ★

Voor dit pittige level moet je level 1 tot en met 4 uitgespeeld hebben. Als dat gelukt is kun je verder bouwen in je eigen branch. Als dat niet gelukt is, kun je verder bouwen in de branch waarin level 1 tot en met 4 opgelost zijn. Overigens hoef je niet met dit level verder te gaan. Je hebt de keuze tussen 5, 6, 7, 8 en 10.



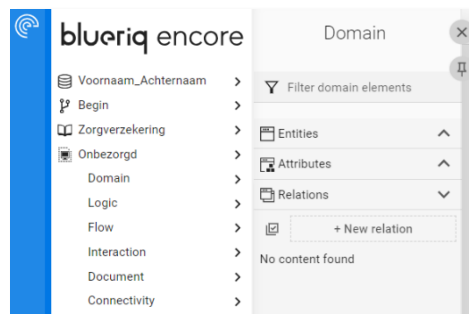
Onze wereld was tot nu toe plat: één aanvrager en één polis. In die ene polis stopten we onze basisverzekering, maar ook **meerdere** aanvullende verzekeringen. Met maximaal twee aanvullende verzekeringen is dat nog wel te overzien, maar met meer wordt dat onoverzichtelijk. Stel je de beslistabel maar eens voor die de aanvullende premie berekent wanneer we naast een aanvullende Tandarts- en Fysiotherapieverzekering nog bijvoorbeeld acht andere verzekeringen aanbieden. Je moet dan alle mogelijkheden afvangen. Bij tien aanvullende verzekeringen levert dat een beslistabel op met 99 kolommen. Dan gebruiken we Blueriq niet zoals Blueriq bedoeld is, namelijk om complexe regels inzichtelijk te maken. Vandaar dat we ons domeinmodel nu gaan aanpassen aan de werkelijkheid:



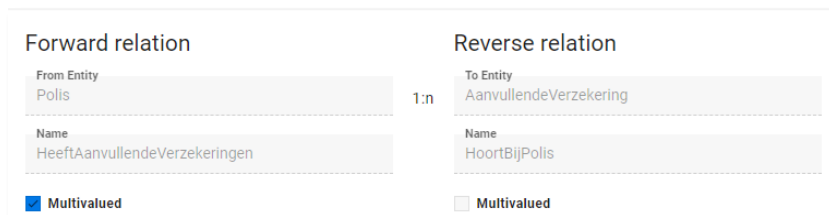
De entiteit Aanvrager blijft hetzelfde – daar houden we er op enig moment maar één van. Ook van Polis houden we er maar één, maar het attribuut aanvullende verzekering trekken we eruit. Het type aanvullende verzekering (dat multivalued attribuut) halen we weg en in plaats daarvan leggen we een relatie naar een

nieuwe entiteit **AanvullendeVerzekering**. Wat bijzonder is aan deze entiteit ten opzichte van Polis en Aanvrager, is dat deze entiteit in de runtime meerdere instanties tegelijk kan hebben! Maar die horen wel allemaal maar bij één polis, want er was immers op enig moment ook maar één polis.

- Hieronder volgen een paar Ja-Nee-vragen om te kijken of je deze wijziging doorziet en een goed beeld hebt van wat we gaan doen straks in ons model.
 - Op elk willekeurig moment zien we in het Blueriq-profiel in de runtime precies één instantie **Aanvrager**, precies één instantie **Polis** en 0, 1 of 2 instanties **AanvullendeVerzekering** (Ja/Nee).
 - Het nieuwe attribuut **Polis.PremieAanvullendeVerzekeringen** zal de som bevatten van de individuele attribuutwaarden **AanvullendeVerzekering.Premie** (Ja/Nee).
 - We kunnen een button plaatsen op onze polis-pagina om aanvullende verzekeringen te kunnen aanmaken (Ja/Nee).
 - We zullen de aanvullende verzekeringen in een lijstje moeten tonen (Ja/Nee).
 - We kunnen de waardenlijst **Type** hergebruiken (Ja/Nee).
- Maak een nieuwe entiteit **AanvullendeVerzekering**.
 - Maak deze **niet singleton**.
 - Maak hierin het string-attribuut **Type**.
 - Hergebruik de **waardenlijst TypeAanvullendeVerzekering** want die hadden we immers nog.
 - Maak hierin ook een currency-attribuut **Premie**.
- Maak een **1-n relatie** tussen de entiteiten **Polis** en **AanvullendeVerzekering** met de naam **HeeftAanvullendeVerzekeringen**. Zorg ook voor een goeie naam voor de **terug-relatie**.



Om de **relatie 1-op-n** te krijgen (wat betekent dat bij 1 polis n aanvullende verzekeringen kunnen horen), moet je het veldje **Multivalued** aanvinken bij de entiteit **Polis**. Verwarrend? Je bent in goed gezelschap, want dat vinden meer mensen 😊



- Verander de **beslistabel** die de premie berekent: deze bepaalt nu elke individuele premie van een aanvullende verzekering op basis van het type (en heeft dus niet meer de kolom met beide aanvullende verzekeringen). Vergeet niet het doel-attribuut van de tabel te vervangen!
- Haal het attribuut **TypeAanvullendeVerzekering** weg uit de entiteit **Polis**. En ook uit de **container** op je **pagina**.
- Hernoem de **question text** van de oude premie van de aanvullende verzekering tot **Totale premie aanvullende verzekeringen**. Hernoem het attribuut zelf tot **PremieAanvullendeVerzekeringen**. Het is misschien even zoeken, maar helemaal rechtsboven staat een potloodje om je element te kunnen hernoemen.

7. In het menu **Flow** vind je ook service calls. Maak een **service call** van het type **AQ_Instance_Create** met de naam **MaakAanvullendeVerzekering**.

Deze gebruiken we om op commando een aanvullende verzekering aan te maken straks.

Service call

Service type
 AQ_Instance_Create

Create an instance and set attributes and relations for it.

Parameters

Entity
 AanvullendeVerzekering

Instance activation
 Before

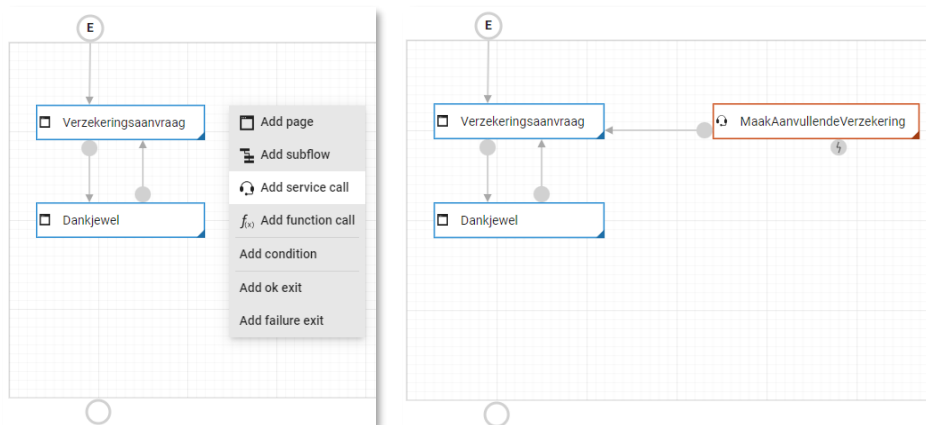
Je hebt een relatie gedefinieerd tussen **Polis** en **AanvullendeVerzekering**. Je legt deze relatie door in de **service call** op de knop **+ Add attribute/relation** te klikken, en vervolgens de relatie **HoortBijPolis** een waarde te geven: **Polis**.

Attributes and relations

Name	Value
HoortBijPolis	Polis

+ Add attribute/relation + Add all remaining attributes and relations

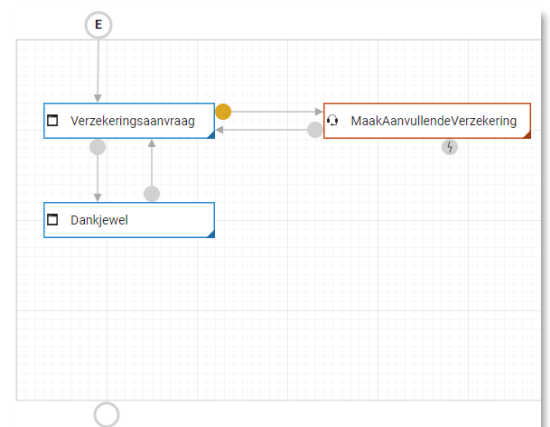
Hang de **service call** **MaakAanvullendeVerzekering** in de **flow**. Leg alvast de weg terug naar de **pagina** **Verzekeringsaanvraag**. De uitgang **Exception** (met de bliksemflits) kun je met rust laten.



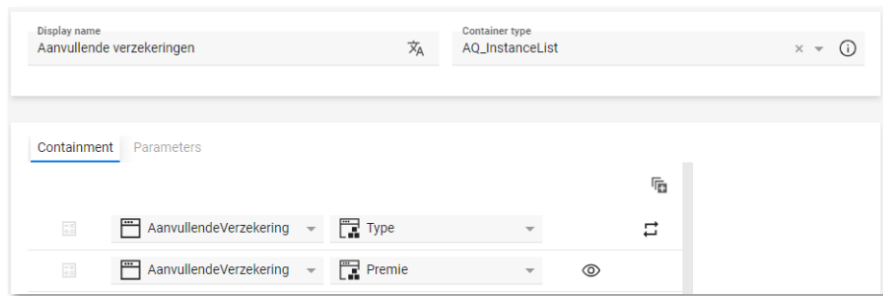
8. Nu moeten we er voor zorgen dat we op commando een aanvullende verzekering kunnen maken. Maak daartoe een **button** met als **caption** **Maak aanvullende verzekering** (en maak ook een bijbehorend **event**) en zet die op de **pagina** na de premie van de basisverzekering. Als je de **button** en het **event** kiest, verander dan het **event type** naar **Continue** (in plaats van **Validate and Continue**) want er hoeft niet gevalideerd te worden als je een instantie **maakt**.

Verbind het bolletje van de **pagina** (dat is het event dat door klikken op die button geactiveerd wordt) met die **service** om aanvullende verzekeringen te kunnen maken.

Die aanvullende verzekeringen gaan we op de verzekeringsaanvraag pagina tonen in een lijstje. Dit gaan we doen met een container die een lijst toont, de zogenaamde **instance list**.



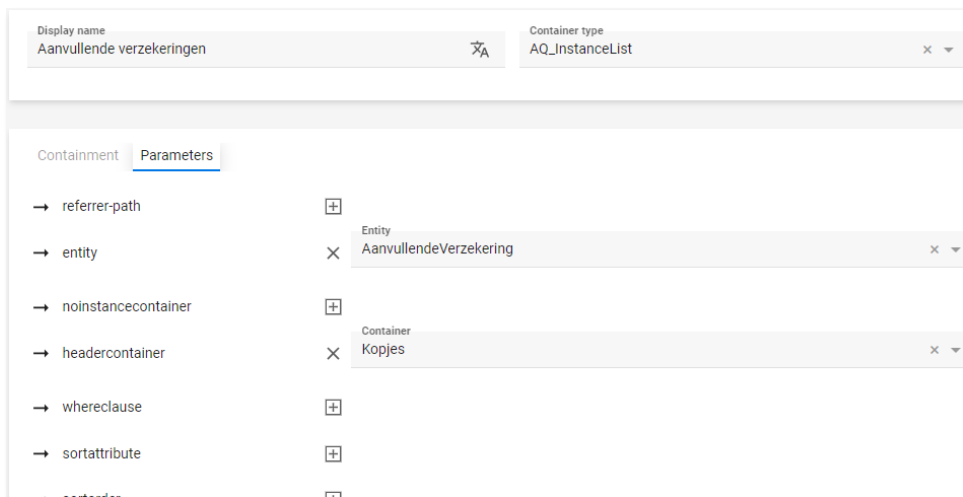
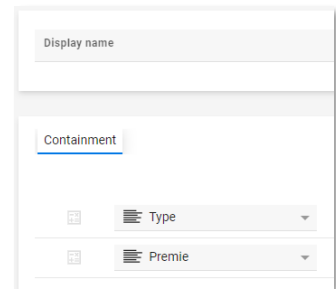
9. Maak een container **AanvullendeVerzekeringenLijst**, met daarin de attributen **AanvullendeVerzekering.Type** met een **refresh** en **AanvullendeVerzekering.Premie** die **read only** is. Geef als **display name** bijvoorbeeld **Aanvullende verzekeringen**. Geef de container als type **AQ_InstanceList**.



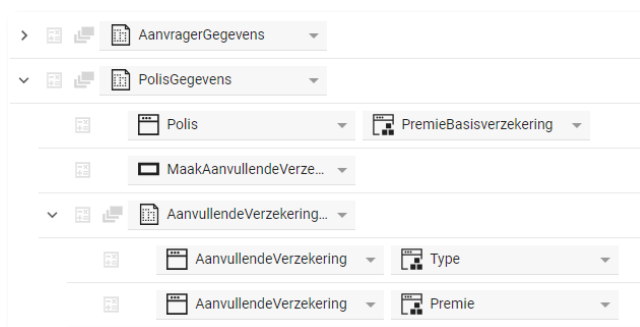
Maak ook een container **Kopjes** met twee **Text items** als kopjes (ze heten **Type** en **Premie** en bevatten dito waardes). Deze container geef je **geen** display name.

Kies die kopjes-container als **header container** in de container **AanvullendeVerzekeringenLijst**. Dat vind je bij het tabje **Parameters**.

Bij die **parameters** zie je dat je ook de **entiteit** moet kiezen. Dat is uiteraard **AanvullendeVerzekering**.



10. De container die de lijst bevat met aanvullende verzekeringen moeten we nog op de pagina zetten. Vlak onder de button **MaakAanvullendeVerzekering** is een goede plek.



11. Herschrijf de **default expressie** bij **Polis.PremieAanvullendeVerzekeringen** (die nu het totaal zal bevatten) naar onderstaande query. Snap je ook waarom deze zo moet zijn?

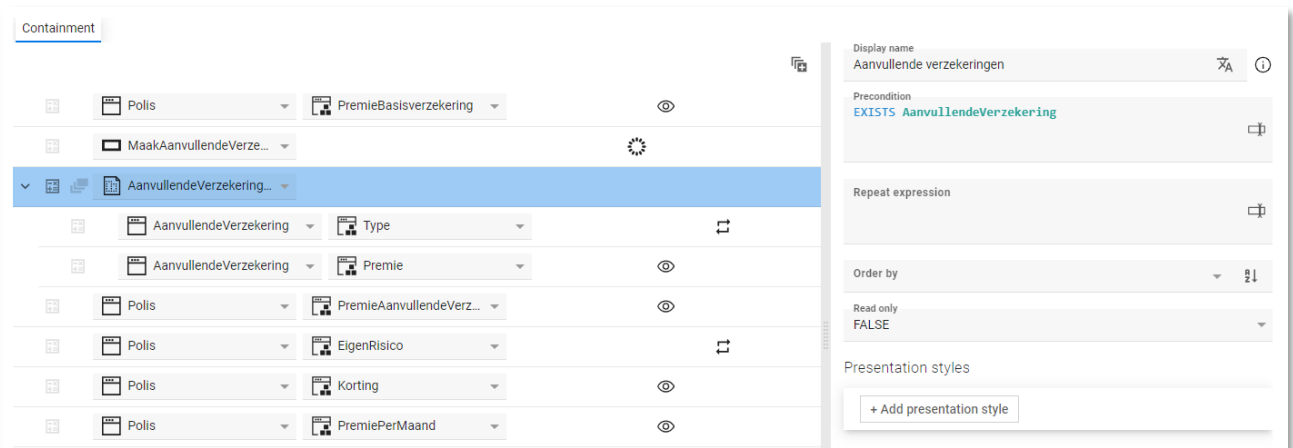
```

Default value
Expression

Expression
SUM ( COLLECT AanvullendeVerzekering.Premie
      FROM Polis.HeeftAanvullendeVerzekeringen)

```

12. Werkt het allemaal? En stoor je je aan het feit dat er altijd een lijst getoond wordt, zelfs als je nog geen aanvullende verzekeringen gemaakt hebt? Dat is logisch, maar wel irritant en gaan we oplossen met een **preconditie** op de **container AanvullendeVerzekering** die we binnen de container **Polisgegevens** hebben staan. Snap je deze?



13. Zo'n zelfde expressie wil je misschien ook wel op je premie van de aanvullende verzekering. Immers, als je helemaal geen aanvullende verzekeringen gaat nemen, wil je ook die premie niet zien.

14. Pas ten slotte het **ERD** aan, want ons domein is veranderd en dat willen we natuurlijk ook visueel bijgewerkt hebben.

