Eénmalige rol

|  |  |  |
| --- | --- | --- |
| Joop Ringelberg | 21-03-18 | Versie: 1 |

# Introductie

Een éénmalige rol heeft precies één actie met effect in zijn perspectief. De vuller van zo’n rol kan die actie éénmaal uitvoeren en daarna is zijn rol ook weer uitgespeeld (afgezien van raadpleegt-acties op de context).

Eén der betrokkenen (B) bij een context met zo’n eenmalige rol betrekt een ander (A) in zo’n rol. Dat gaat op de gebruikelijke wijze.

Hierdoor beschikt A over (een perspectief op) die context. De benodigde gegevens worden hem door de Perspectives engines van alle (andere) betrokkenen toegestuurd, op de gebruikelijke wijze.

A voert zijn actie uit. Deze actie kan een effect op de context hebben, waarvan andere betrokkenen al naar gelang hun perspectief verwittigd worden (ze kunnen de toestandsverandering van de context zien).

Direct na deze actie is A’s rol uitgespeeld, alhoewel hij zijn raadpleegt-acties behoudt.

Per definitie zullen de andere betrokkenen nooit een actie uitvoeren waarbij A betrokken is (waar A het lijdend- of meewerkend voorwerp van is).

# Consequenties

De andere betrokkenen hoeven in hun administratie niet bij te houden dat A een rol heeft gespeeld, nadat hij klaar is. Ze ontvingen van B informatie over A’s tijdelijke betrokkenheid, en stuurden A de informatie over de context waar hij recht op had. Vervolgens ontvingen ze van A het effect van zijn actie, wat ze in hun eigen administratie registreerden. En daarmee is de kous af en kunnen ze A schrappen als betrokkene.

Evenmin hoeft de engine van A te onthouden wat de andere betrokkenen waren, want A kan nooit meer de toestand van de context veranderen. Hij hoeft hen dus nooit meer iets toe te sturen.

# Vergelijking en nut

Een éénmalige rol is de Perspectives manier van een service die eenmalig benaderd kan worden met een token. Het is een ‘gedistribueerde’ service, want elk van de andere betrokkenen functioneert als ‘endpoint’.

## Toepassing: geld

Deze ‘service’ is nuttig voor een geldsysteem binnen Perspectives. Eén van de problemen die moet worden opgelost om een betrouwbaar systeem te kunnen bouwen, is dat een actor een transactie maar door precies één andere transactie kan laten volgen (en recursief die weer laten volgen door een transactie, enz.). Zo voorkomen we dat een actor meer dan eens hetzelfde geld uitgeeft.

Maar er is geen sluitende manier om dit met de administratie van transacties en hun betrokkenen te garanderen, omdat een willekeurig aantal betrokkenen kan samenzweren om een ander te bedriegen.

## Oplossing: een service

Dit probleem zouden we kunnen oplossen met een service, verleend door een betrouwbare partner, die voor elke actor een volgnummer bijhoudt. Dat werkt als volgt: een actor A wil met een partner P een transactie aangaan. Zijn vorige transactie heeft volgnummer n. Dit is geregistreerd op de service. A verkrijgt van de service, met n, een *token* dat hij aan P geeft, die met dat token bij de service n verifieert. Alleen als het klopt, gaat P een transactie met A aan. Als gevolg van de verificatie is op de service n verhoogd tot n+1.

A kan dus nooit het token nogmaals gebruiken, want een partner P’ (die hij wil bedriegen) zal van de service een negatief resultaat ontvangen op n (immers op de service is inmiddels n+1 geregistreerd).

## Oplossing met een eenmalige rol

De context waar A P in betrekt, heeft een property *volgnummer* die A kan raadplegen. Dit is het volgnummer n van de laatste transactie die A heeft uitgevoerd. Bij deze context is een aantal zeer betrouwbare Partijen betrokken[[1]](#footnote-1).

Alle andere betrokkenen zullen P deze informatie sturen. Omdat de andere betrokkenen betrouwbaar zijn, weet P zo heel zeker dat n inderdaad de vorige transactie is die P uitvoerde.

P heeft ook een actie die hij kan uitvoeren op de context waar hij in is betrokken. Namelijk: hij kan bevestigen dat hij de transactie aangaat[[2]](#footnote-2). Daardoor wordt de waarde van de property met één verhoogd. Alle andere betrokkenen ontvangen deze verhoging.

Omdat de actie éénmalig is, kan P geen kwaad aanrichten door het volgnummer van A willekeurig te verhogen. Hij zou dat op zijn eigen computer misschien kunnen door Perspectives te hacken, maar de andere betrokkenen pakken alleen de eerste verandering die zijn engine stuurt, aan. Zijn rol is in het model immers als éénmalig gedefinieerd!

Merk op dat in dit verhaal de rol van het token is vervangen door de handeling waarmee A P bij de context betrekt.

1. Merk op dat het voor de werking van het mechanisme niet vereist is dat deze verzameling betrokkenen over de tijd heen gelijk blijft. Er kunnen partijen bijkomen of weggaan; belangrijk is dat er op elk moment een voldoend aantal voldoend betrouwbare partijen bij betrokken zijn.

   Wat is betrouwbaar? Het gaat erom dat de transactiepartners van A ze betrouwbaar vinden. [↑](#footnote-ref-1)
2. Deze bevestiging is voor A belangrijk. Hij kan hiermee de feitelijke transactie met P afdwingen. Om die reden zal A waarschijnlijk in zijn versie van de context A wel als betrokkene handhaven (en waarschijnlijk met het volgnummer op de rol zelf geregistreerd). [↑](#footnote-ref-2)