Gebruiksaanwijzing Rana API voor de Unreal plugin

Casper van der Wel, Nelen & Schuurmans, 20 april 2026, versie 4

Binnen het project Digital Twin as a Common Good ontwikkelt Nelen & Schuurmans een interface om een specifieke rekentool van het hydrodynamische model Rana aan te sturen. Deze interface wordt via het Open Urban Platform (Clearly.hub, FutureInsight) aangeboden en in een digital twin omgeving (Unreal, The People Group) gebruikt.

Dit document bevat de benodigde acties die nodig zijn om modellen te vinden, een simulatie te starten, de status van een simulatie te controleren, en de resultaten van een simulatie te verkrijgen als WMS en OGC 3D Tiles.

Rana API en Authorisatie

Rana is een projectgerichte werkomgeving voor wateranalyses. Binnen Rana heeft elke organisatie zijn eigen omgeving, met eigen data (Rana is een zgn. “multitenant” applicatie). De gemeentes binnen het DT-Acogo project krijgen hun eigen organisatie in Rana. Wij koppelen deze organisatie met het OUP, zodat we binnen OUP “datasets” en “resources” kunnen configureren waar gebruikers bij kunnen, volgens de rechten die in OUP uitgedeeld zijn. De basis-url van Rana hiervoor is:

https://www.ranawaterintelligence.com/v1/tenants/{code}

De code verschilt per gemeente. De API documentatie is hier te vinden:

- https://www.ranawaterintelligence.com/v1/redoc
- https://www.ranawaterintelligence.com/v1/docs (interactief)
- https://www.ranawaterintelligence.com/v1/openapi.json (schema)
- https://www.ranawaterintelligence.com/v1/openapi.yaml (schema, yaml versie)

De machine-to-machine authenticatie van OUP naar Rana vindt plaats met een API key. Deze API key heeft toegang tot de hele Rana tenant (en niet slechts 1 project). De API key wordt door Nelen & Schuurmans eenmalig per OUP hub aangemaakt en in OUP geconfigureerd, zodat de OUP Datasets veilig kunnen communiceren met Rana.

Achtergrondinformatie - proces

De rekentool stelt de gebruiker in staat om scenario’s met **constante regenval** te simuleren voor een bestaande modelschematisatie. De simulatie begint met een

regenperiode, waarin een constante regenintensiteit wordt toegepast over het volledige model. Nadat de regenperiode eindigt, gaat de simulatie verder met een **droge periode**, zodat je waterretentie, afstroming (runoff) en drainagegedrag kunt observeren.

Dit type simulatie is essentieel voor het analyseren van hoe het terrein en landschap reageren op langdurige regenval. Het helpt bij het identificeren van mogelijke **overstromingsrisico's** en bij het evalueren van **waterbeheerstrategieën**.

Parameters:

- Het model. Dit is een volledig pad naar een bestand dat in het Rana project waarin het proces gedraaid wordt.
- Lengte van de simulatie, in minuten: standaard 120 minuten.
- Lengte van de regenbui, in minuten. Moet kleiner of gelijk zijn aan lengte van de simulatie. Standaard 60 minuten.
- Intensiteit van de regenbui in mm / uur . Standaard: 80 mm/uur.
- Gewenste locatie van de resultaten (binnen het project)
- Gewenste locatie van de OGC 3D Tiles (binnen het project)

Integratie van Rana project bestanden met OUP

Het Open Urban Platform is niet in staat om individuele bestanden uit Rana te registeren. Dit komt door de opzet van de Rana API. Daarom spreken we het volgende af:

- In een project staat een map op een vaste locatie met flood models. Deze locatie is "dt-acogo/floodmodels/".
- Elk project wordt los geregistreerd als dataset binnen OUP.
- De resultaten van simulaties via Unreal worden opgeslagen, in de map "dt-acogo/results/{YYYYMMDDTHHmmss.sssZ}/" met vaste bestandsnamen "results.zip" en "ogc_3d_tiles.zip". Het is aan de caller (Unreal) om deze paden in te stellen.

Volgorde aanroepen voor Unreal plugin:

1. Selecteer de juiste Hub in OUP
2. Identificeer Rana-datasets binnen OUP (voorbeeld: https://hub.clearly.app/datasets/69b3cb3dbf7ae593dc9c9e36). Ik heb een tag "rana-dtacogo-v1" toegevoegd om op te kunnen selecteren.
3. Achterhaal de Rana tenant id. Ik heb een tag "rana-tenant-xxx" aangemaakt, waar xxx het tenant id is. Alternatief is om hem uit de url van de API resource te halen.

4. Achterhaal de Rana project id. Ik heb een tag “rana-project-xxx" aangemaakt hiervoor, waar xxx het project id is. Alternatief is om de resource “Project API” te gebruiken om het id te verkrijgen.
5. Selecteer het juiste Rana proces (process_list), hierbij is het process op tag te vinden (gebruik de query “tag=rana-dtacogo-v1”). De response bevat een OpenAPI schema voor de input en output velden; gebruik hiervaan is optioneel, de plugin can ook volledig ge-hardcode worden.
6. Bouw de plugin op in de UI.
7. Start het Rana proces (process_execution). Zie onder voor een voorbeeld.
8. Wacht tot het proces klaar is (job_retrieve). Poll bijv. elke 5 seconden tot state.type == “completed” of “failed” . Progress kan ook uit de response worden gehaald.
9. Verkrijg het resultaat van de job (job_results). Deze stap kan worden overgeslagen als de paden nog bekend zijn (zelf gezet in de process_execution call)
10. Verkrijg de metadata (o.a. “descriptor_id”) van het geproduceerde bestand (project_file_stat)
11. Gebruikt de “descriptor_id” uit stap 10 om te wachten tot de file descriptor klaar is (file_descriptor_retrieve). Wacht tot state.id == "completed" of "failed".
12. Verkrijg de 3D Tiles index (get_ogc_3d_tilesets). In de reponse zijn alle gegenereerde tijdstappen met hun tileset ids te vinden.
13. Verkrijg de 3D Tiles (get_ogc_3d_tileset_file). De tileset id komt uit de vorige stap. De index file is altijd “tileset.json”.
14. Optioneel: voor het resultaat op WMS zijn stappen 9-10 te herhalen met het andere (“results.zip”) resultaat. Het WMS endpoint: file_descriptor_wms .
15. Optioneel: bij opruimen kunnen de resultaten verwijderd worden met (project_delete_directory)

Process execution JSON body (voorbeeld)

```
{
  "name": "DT acogo: Run from Unreal (13 Mar 2026 09:46)"
  "project_id": "GHL8S9sM",
  "inputs": {
    "schematisation": {
      "ref": "main",
      "variable_type": "rana_path",
      "id": "dt-acogo/floodmodels/Goudsesingel huidig"
    },
    "simulation_duration": 5
    "rain_duration": 5,
    "rain_intensity": 80,
```

```
    "tile_interval": 5
  },
  "outputs": {
    "file": {
      "ref": "main",
      "variable_type": "rana_path",
      "id": "dt-acogo/results/20260313T084941Z/ogc_3d_tiles.zip"
    },
    "results": {
      "ref": "main",
      "variable_type": "rana_path",
      "id": " dt-acogo/results/20260313T084941Z/results.zip"
    }
  }
}
```