European data space
for smart communities

Deliverable 3.5

DS4SSCC Deliverable 3.5 OUP Dataflows Validated in the Visualisation Layer

Integration walkthrough and validation cases – version 2.0 (revised after OASC review)

WP 3 – Model usage phase

Authors: Corné Helmons (Rotterdam)
Reviewers: Niek Verschure ('s-Hertogenbosch), Anni Joela (Tampere), Margriet Visscher (Future Insight)
Delivery date: 7 July 2026 (v1.0); 30 September 2026 (v2.0, revised)
Dissemination level: Sensitive
Type: Report

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or of the granting authority. Neither the European Union nor the granting authority can be held responsible for them

Revision History

Author Name, Partner short name	Description	Date
Corné Helmons, Rotterdam	v1.0 – first submission (City Project Manager)	7-Jul-2026
Corné Helmons, Rotterdam	v2.0 – revised after the OASC deliverable assessment: title clarified, validation criteria defined, walkthrough condensed, validation cases and multi-model evidence added, D2.3 reference corrected, conclusions rewritten	30-Sep-2026

Abbreviations

Abbreviation	Definition	Abbreviation	Definition
DS4SSCC	European data space for smart communities (DS4SSCC)	DT-aCoGo	Digital Twinning as Common Good
OASC	Open & Agile Smart Cities & Communities	OUP	Open Urban Platform
WP	Work Package	PC	Pilot Coordinator
BIM	Building Information Modeling	WFS	Web Feature Service
OGC	Open Geospatial Consortium	ETL	Extract, Transform, Load
LOD	Level of Detail	UMG	Unreal Motion Graphics
REST	Representational State Transfer		
CIM	Common Information Model		

Response to the OASC review

Version 1.0 of this deliverable was assessed by OASC as "accept with minor revision", conditional on either retitling the deliverable or adding the missing validation evidence. Version 2.0 does both: the title now states the scope precisely, and a validation chapter with one case per model provider has been added. The table below lists each review comment, where it is addressed and what was changed.

Reviewer comment	Addressed in	Change made in v2.0
Title overpromises: "Dataflows Validated" implies tested and confirmed flows; "validated" asserted seven or eight times without a single instance.	Title; Ch. 1; Section 3.2; Ch. 5	Title extended to "OUP Dataflows Validated in the Visualisation Layer". "Validated" is defined by four explicit criteria (V1–V4) and used only where a case in Chapter 5 backs it; elsewhere replaced by "demonstrated" or "operational".
No concrete validation instance; more detail needed per pilot site / model.	Section 4.2; Ch. 5 (5.2–5.4, 5.6)	New Section 4.2 with the data services published per model provider (round trip source → Common Model → OUP → provider). New Chapter 5 with the validation baseline (earlier project-based runs on manually prepared data) and one documented case per model provider: data service, inputs, run, checks per criterion with expected and observed behaviour, outcome; plus an issues-found table.
Chapters 4 and 5 substantially duplicate Annex 1 (user manual).	Ch. 4	Former chapters 4 and 5 merged into one condensed walkthrough. Operating instructions delegated to Annex 1; each screen now states which dataflow it evidences and which criterion it relates to.
Section 5.7: multi-model claim shown with a single traffic layer only.	Section 5.5	Claim split into what is evidenced (three results co-existing in one session, Figure 14) and simultaneous visualisation (new Figure 15, three model results active at once in the 's-Hertogenbosch Spoorzone twin, which also evidences replication to a second pilot city).
Subscription rules "aligned with D3.6" – is this the correct reference?	Section 4.3	Corrected: governance rules follow D2.3 (Governance & Ethical Board checks) and the OUP governance and authentication set-up; D3.6 is referenced only for the ethical assessment of the models.
Chapter 6 repeats D3.4's lessons learned without new nuance or evidence from the visualisation layer.	Ch. 6 (6.1, 6.2)	Conclusions rewritten: each lesson is linked to evidence in Chapters 4–5; new Section 6.2 states explicitly what D3.5 adds beyond D3.4.
The gap between operational evidence and validation language should be mitigated in the show and tell.	Section 6.3	New section describing the live demonstration, structured along the validation criteria.

Table of contents

1. Executive Summary 5
2. Introduction 6
3. Problem definition and validation approach 7
3.1 Problem definition 7
3.2 What “validated” means in this deliverable 7
3.3 Relation to Deliverable 3.4 8
4. End-to-end walkthrough of the integrated environment 9
4.1 Role of the visualisation layer 9
4.2 Data as a service: from source to model provider 9
4.3 Login and subscription 10
4.4 Digital twin selection and dataset loading 12
4.5 Data & Analyses panel 14
4.6 Running the analysis models 14
4.7 Results and expert panel 16
5. Validation cases 17
5.1 Approach 17
5.2 Case 1 – Rana (flooding) 17
5.3 Case 2 – Tygron (heat stress) 18
5.4 Case 3 – DAT.mobility (traffic) 18
5.5 Multi-model visualisation 19
5.6 Issues found at the visualisation layer 21
6. Conclusions and next steps 22
6.1 Conclusions 22
6.2 What this deliverable adds beyond D3.4 22
6.3 Show and tell 23
6.4 Next steps 23
7. Annexes 24

1. Executive Summary

This deliverable closes Work Package 3 by showing that the dataflows defined and tested in Deliverables 3.1 to 3.4 are operational in the visualisation layer of the DT-aCoGo ecosystem: the Unreal Digital Twin application. It presents two things: an end-to-end walkthrough of the integrated environment as it runs today (Chapter 4), and a set of validation cases in which one dataflow per model provider – Rana (flooding), Tygron (heat stress) and DAT.mobility (traffic) – was checked against explicit criteria (Chapter 5).

What the DT-aCoGo pilot set out to demonstrate is that a city can provide its data **as a service** – harmonised once, published through the OUP and consumed by any model provider through a provider-specific data service – and that the set-up of the climate analyses (flooding, heat stress, traffic) can be **partly automated** on top of those services. The city and the three model providers have worked together for several years in earlier projects, always with project-based, manual data preparation; the analyses themselves are therefore known to perform correctly. What is new, and what this deliverable validates, is the dataflow: the same analyses now run on data delivered as a service through the OUP, set up from the digital twin application, with the results flowing back through the OUP into the same environment.

The word “validated” is used deliberately in this deliverable. A dataflow is considered validated at the visualisation layer when four conditions are met: (V1) the input datasets retrieved from the Open Urban Platform (OUP) are the harmonised CIM datasets defined in D3.1; (V2) a model run can be configured, started, monitored and completed from the application through the APIs described in D3.3 and D3.4; (V3) the returned result is registered in the OUP, carries the correct temporal context (D3.2) and is rendered in the 3D scene; and (V4) access to datasets and models is governed by the OUP subscription (D2.3). Section 3.2 defines these criteria and how each one is checked; Chapter 5 reports the checks per model provider, including the issues that were found and how they were resolved.

The walkthrough confirms that the application is operational end-to-end: login through the OUP, subscription-based loading of datasets and models, three distinct model-provider panels with real parameter fields, and results that appear in the Data & Analyses panel and in the 3D scene. The validation cases add what a walkthrough alone cannot show: what was checked, against what, and with which outcome.

The main lessons from this final integration step are:

- Harmonisation and conversion are tested by the user: where the conversion of the BIM design dropped lifecycle attributes and added prefixes, this became visible immediately in the visualisation layer and was fixed (Section 5.6).
- Temporal consistency has to be carried by the data itself: the time slider behaves correctly only where objects carry lifecycle attributes (validFrom / validTo) as defined in D3.2 (Section 5.6).
- API-based integration is stable but remains model-specific: each provider needed its own parameter panel and result handling (Sections 4.6 and 5.2–5.4).
- The visualisation layer is an integration point in its own right: it is where inconsistencies between data, time and model results become visible to the user, and therefore where they are most effectively detected.

Overall, the DT-aCoGo ecosystem provides an operational and governed digital twin environment in Rotterdam. The configuration documented here is the reference for the replicable deployment recipe in Deliverable 4.1.

2. Introduction

The DT-aCoGo project is developed within the European Data Space for Smart Communities (DS4SSCC) initiative and aims to establish a scalable, interoperable and replicable digital twin ecosystem for urban environments. Within this ecosystem, the Open Urban Platform (OUP) functions as the central data space, enabling stakeholders to publish, discover and reuse data and services across domains and cities.

A key principle of the DT-aCoGo architecture, as defined in Deliverable 3.1, is the separation between data, analytical models and visualisation environments. This separation keeps data at its authoritative source while it is accessed through standardised interfaces, which supports interoperability, scalability and reuse across applications and cities.

Within Work Package 3 (Model Usage Phase) a stepwise approach has been followed to realise and test this architecture:

- **D3.1** established the interoperability framework, including the Common Information Model (CIM), semantic data harmonisation and the integration of heterogeneous analytical models.
- **D3.2** introduced time as an intrinsic property of data and defined lifecycle-based filtering to keep datasets, analyses and visualisation consistent.
- **D3.3** described the application-level integration: how data and analytical services are orchestrated and made accessible through applications.
- **D3.4** validated the dataflows between the OUP and the connected analytical models at the API level: data can be requested, processed and returned through standardised interfaces.

Deliverable 3.5 covers the final integration step: the same dataflows, seen from the visualisation layer and from the perspective of the user. The Unreal Digital Twin application is the primary user interface of the DT-aCoGo ecosystem. By registering the application and its subscription in the OUP hub, the full configuration – datasets, governance rules and API connections to the analytical models – is loaded automatically when the application starts.

The objective of this deliverable is therefore twofold:

- to demonstrate that the complete workflow – from data provisioning and harmonisation, through model execution, to visualisation – is operational within a single integrated environment (Chapter 4); and
- to validate, per model provider, that the dataflow behaves correctly at the visualisation layer according to explicit criteria (Chapter 5).

The step-by-step operation of the application (menus, buttons, settings) is documented in the user manual DT-aCoGo-Uitleg (Annex 1). This deliverable does not repeat that manual; it refers to it where relevant and concentrates on the dataflows the screens evidence.

Reading guide. Chapter 3 states the problem and defines the validation criteria. Chapter 4 walks through the integrated environment. Chapter 5 reports the validation cases, the multi-model visualisation and the issues found. Chapter 6 draws conclusions, explains what this deliverable adds beyond D3.4 and describes what will be shown in the show-and-tell session.

About this version. This is version 2.0 of the deliverable, revised after the OASC deliverable assessment. The changes made in response to the review are listed in the section “Response to the OASC review” at the front of this document.

3. Problem definition and validation approach

3.1 Problem definition

The DT-aCoGo ecosystem is based on a deliberate separation between:

- Data (Open Urban Platform)
- Analytical models (external model providers)
- Visualisation (Unreal Digital Twin)

As described in D3.1, this separation is essential for scalability and interoperability, but it introduces complexity in integration. Deliverables 3.1–3.4 addressed the individual parts of this challenge: harmonisation through the CIM (D3.1), temporal consistency (D3.2), application-level orchestration (D3.3) and API-level dataflows between OUP and models (D3.4).

The question that remained is how these components combine into a single, user-friendly and fully governed environment. Without such integration, users would have to configure datasets and APIs manually, governance rules might not be applied consistently, and analytical results would be disconnected from their visualisation context. Moreover, an integration that works at the API level can still fail for the user: a result may arrive but at the wrong moment in time, on the wrong objects, or outside the governance of the subscription. Those failures only become visible in the visualisation layer.

3.2 What "validated" means in this deliverable

Validation baseline. The analytical models are not validated in this deliverable. Rotterdam has used Rana, Tygron and DAT.mobility for several years in earlier projects, always on manually prepared, project-specific data, and their results are accepted in the city's practice. That project-based practice is the reference. The first time the same data was provided as a service through the OUP – harmonised to the Common Model by Purple Polar Bear and published per provider (Section 4.2) – was in this pilot. Validation therefore means: the analyses, set up (partly) automatically from the digital twin application on data served through the OUP, behave as they did on manually prepared data, and their results flow back through the OUP into the twin.

To avoid asserting validation without evidence, this deliverable uses four explicit criteria. A dataflow is validated at the visualisation layer when all four are met for the model provider concerned. Chapter 5 reports each criterion per provider.

Criterion	Definition	How it is checked	Evidence in this document
V1 – Correct input	The datasets the model consumes are the harmonised CIM datasets published in the OUP for the selected digital twin (D3.1).	The provider's data service ("stekkerdoos") in the OUP hub serves the Common Model datasets; the datasets listed in the application match them; the served data corresponds to the manually prepared data used in earlier project runs.	Sections 4.2, 4.4, 4.5, 5.2–5.4
V2 – Executable flow	A model run can be configured, started, monitored and completed from the application through the provider API (D3.3, D3.4).	A run is set up and started from the UI without manual data preparation; status changes from empty to running to finished; a run identifier is returned by the provider.	Sections 4.6, 5.2–5.4

Criterion	Definition	How it is checked	Evidence in this document
V3 – Consistent result	The result is registered in the OUP as a service, loads in the scene and carries the correct temporal context (D3.2).	The result appears in the panel with its run timestamp, is rendered in 3D at the correct time-slider position, can be reloaded in a new session, and is consistent with results from earlier project-based runs.	Sections 4.7, 5.2–5.5
V4 – Governed access	Datasets and models are only available under the selected subscription (D2.3).	The models are registered as APIs in the hubs; the user logs in to the OUP and selects the hub; the application calls the model APIs through a proxy that identifies on behalf of the logged-in user; the application asks consent for subscription and billing access.	Sections 4.3, 5.2–5.4

3.3 Relation to Deliverable 3.4

D3.4 validated the flows between the OUP and the analytical models at the API level: request, processing and response. D3.5 does not repeat those tests. It checks the same flows one level higher, in the application the user works with, where data, time and results must be consistent simultaneously. Where D3.4 answers “does the model receive and return the right data?”, D3.5 answers “does the user get the right result, on the right objects, at the right moment, under the right subscription?”.

4. End-to-end walkthrough of the integrated environment

This chapter walks through the Unreal Digital Twin application as it runs in the Rotterdam pilot. The screenshots are taken from the operational environment. For each screen the text states which dataflow it evidences and which validation criterion of Section 3.2 it relates to. How to operate the screens is described in Annex 1 and is not repeated here.

4.1 Role of the visualisation layer

The Unreal application is a client of the OUP: at start-up it retrieves configuration, datasets and service endpoints from the platform on the basis of the authenticated user and the selected subscription. It is the layer where harmonised data (D3.1), temporal filtering (D3.2) and analytical services (D3.3, D3.4) come together in one 3D environment. The application itself holds no data and no model logic; everything it shows is the result of a dataflow through the OUP.

4.2 Data as a service: from source to model provider

Before a user logs in, the data half of the round trip has already been made. Purple Polar Bear (PPB) connects the Rotterdam source datasets, transforms them to the Common Model and publishes them – per model provider – as data services in the OUP hub PPB-DT-aCoGo-Rotterdam. Each provider has its own service ("stekkerdoos") that offers the Common Model data in the format the provider needs (WFS, JSON, TIFF, downloads or 3D tilesets). The same set-up exists for 's-Hertogenbosch and Tampere. This section shows the three screens that evidence this part of the dataflow; the full description is in the PPB delivery document (Opleveringsdocument 100% DT-aCoGo, August 2026).

The datasets served for Rotterdam are Buildings 2D (BAG) and 3D (3D Rotterdam), Pavement 2D and Trees 2D (Rotterdam ArcGIS services), Trees 3D (3D Rotterdam), the scenario dataset Merwehaven 2D and the mobility key figures of the M4H zoning plan. For each dataset the source endpoint, the Common Model endpoint and the publication status in the OUP are visible.

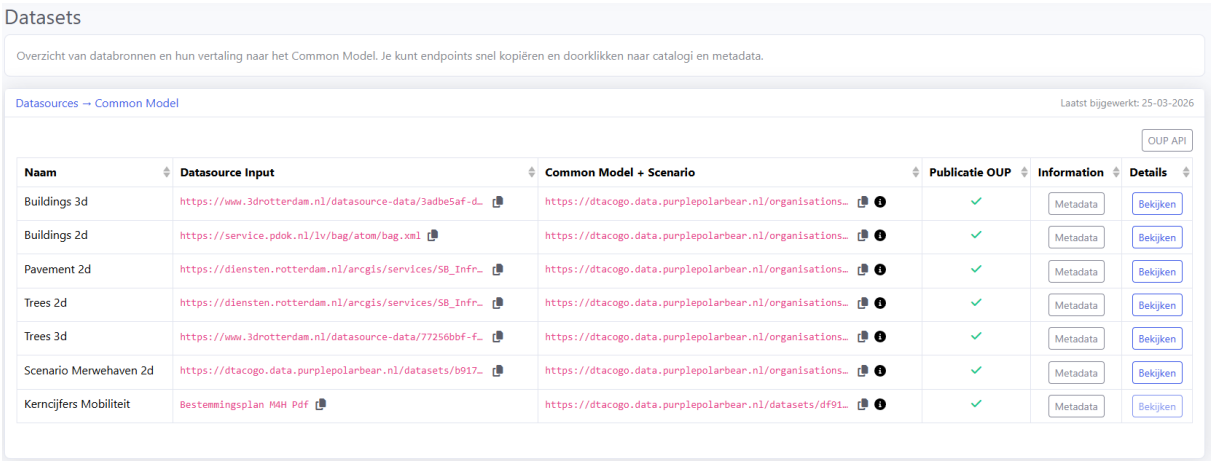


Figure 1 – Datasets: source (Datasource Input) → Common Model endpoint → published in the OUP (PPB Common Model portal, 25 March 2026).

The provider services are the "plugs" through which each model provider retrieves the Common Model data. Four services are published: Tygron, Rana (Nelen & Schuurmans), DAT.mobility and Unreal Engine (The People Group).

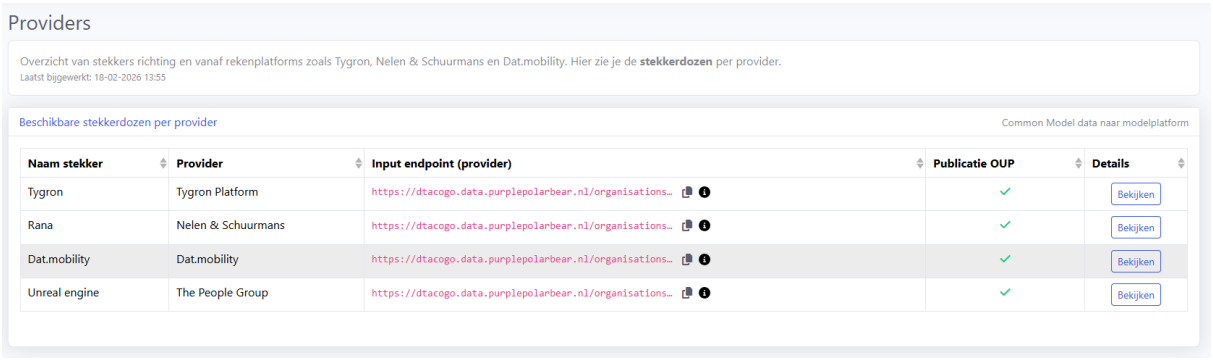


Figure 2 – Provider-specific data services ("stekkerdozen") with their input endpoint and OUP publication status (18 February 2026).

In the OUP the services appear as datasets of the hub PPB-DT-aCoGo-Rotterdam: the Common Model datasets and one "Stekkerdoos" per provider. Results from the model providers are published back to the same OUP, which closes the round trip (Sections 4.5 and 4.7).

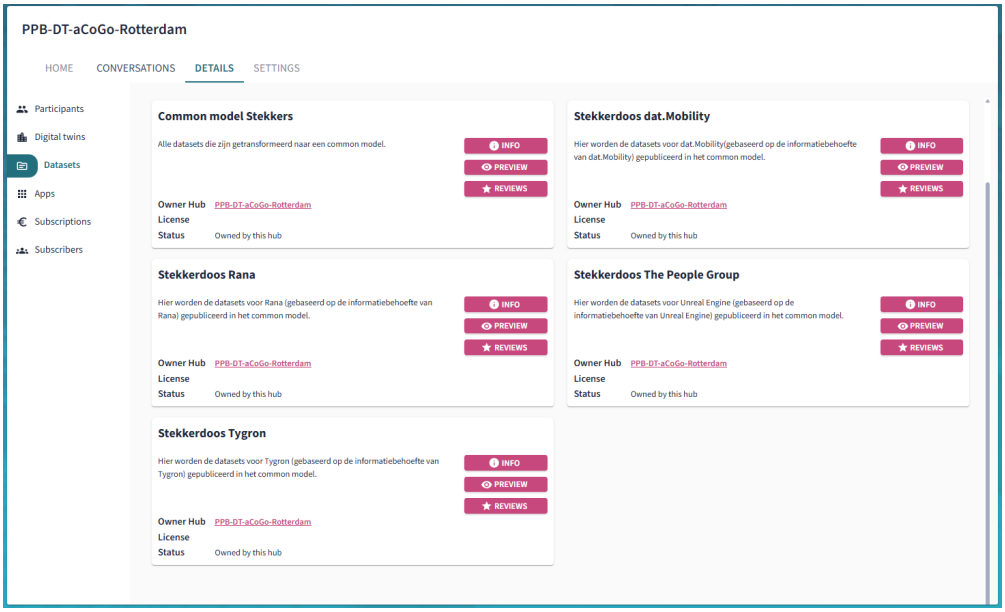


Figure 3 – OUP hub PPB-DT-aCoGo-Rotterdam with the Common Model datasets and the provider services.

What this evidences: correct input (V1). The datasets a model consumes are the Common Model datasets of the OUP hub, served through the provider's own service; no project-specific manual preparation is involved. PPB reports the end-to-end validation of this data chain with the model providers as completed in its 100% delivery (August 2026).

4.3 Login and subscription

Users authenticate through the OUP environment (single sign-on). After login, the user selects a subscription. The subscription is the governance anchor of the whole session: it determines which datasets are accessible (D3.1), which models can be executed (D3.4) and which governance rules apply. Those rules follow the checks and guidance of the Governance & Ethical Board (D2.3) and the OUP governance and authentication set-up; the ethical assessment of the models themselves is reported in D3.6.

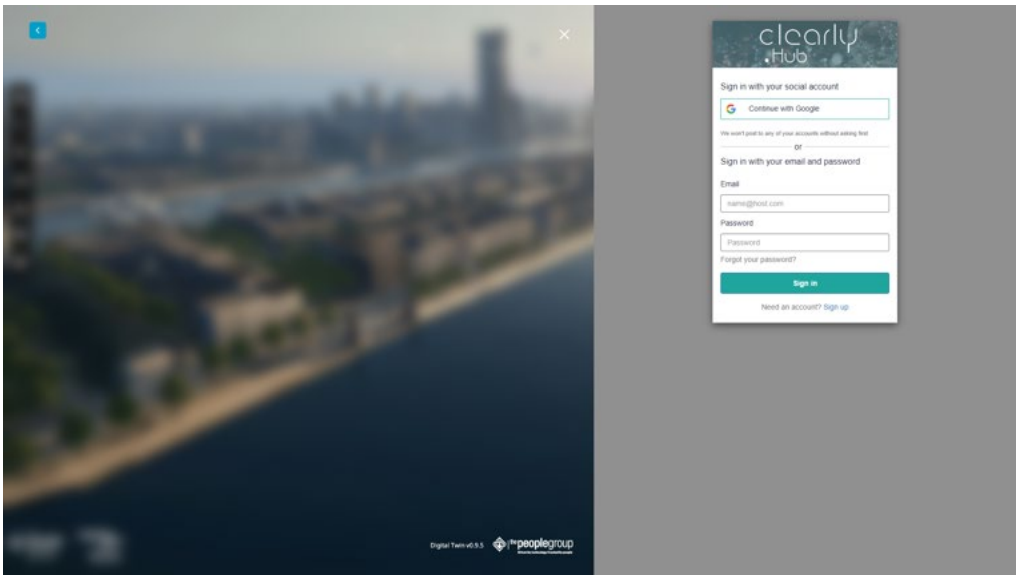


Figure 4 – Login through the OUP environment.

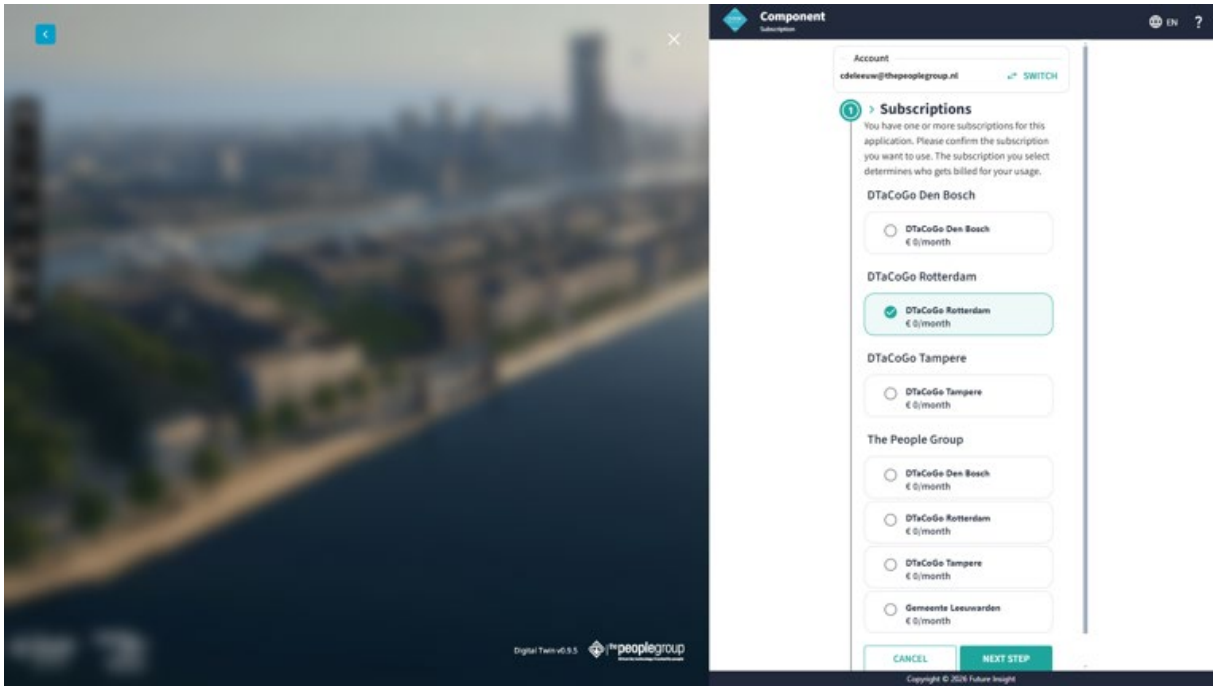


Figure 5 – Subscription selection; the list shows the subscriptions available to this user (Rotterdam, Den Bosch, Tampere and The People Group).

The user also has to grant the application access to the selected subscription and to the billing information. Without this consent the application does not load any dataset or model.

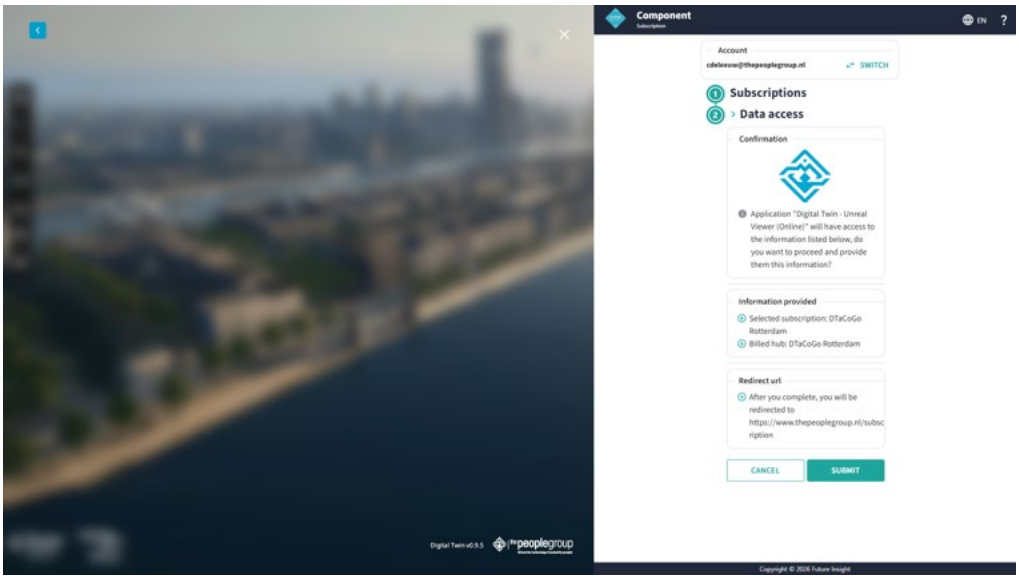


Figure 6 – Consent screen for subscription and billing access.

At run time the application does not hold credentials for the model providers itself. All analysis models are registered as APIs in the OUP hubs, and the application calls them through a proxy server that identifies on behalf of the logged-in user. The models a user can run are therefore exactly those registered in the hub of the selected subscription.

What this evidences: governed access (V4). The datasets and models loaded later in the session are those of this subscription and no others (see Sections 5.2–5.4).

4.4 Digital twin selection and dataset loading

Within a subscription the user selects a digital twin instance. Each instance is a configuration in the OUP: a set of datasets, a camera position and the models that are connected. In the Rotterdam hub two instances are available (DTaCoGo Rotterdam and DTaCoGo Rotterdam v2.0), which shows that several configurations can coexist under one subscription and that the same application serves several cities.



Figure 7 – Selection of the digital twin instance within the subscription.

After selection all datasets of the instance are loaded automatically from the OUP and the 3D environment is rendered. No dataset is configured in the application itself.



Figure 8 – Main user interface after loading; main menu, time and date controls, settings and weather.

What this evidences: correct input (V1). The datasets rendered here are the harmonised CIM datasets that D3.1 defines for Rotterdam; the time and date controls apply the lifecycle filtering of D3.2 to those datasets.

4.5 Data & Analyses panel

The Data & Analyses panel lists the datasets and the analysis results of the session and lets the user add datasets, add analyses, set visibility and manage settings. It is the single place where data (from the OUP) and results (from the model providers) meet.



Figure 9 – Data & Analyses panel with datasets and Open Urban Platform simulations.

What this evidences: the panel is the visible form of the dataflows of D3.1 (data), D3.3 (orchestration) and D3.4 (model interaction). The entries under "Open Urban Platform Simulations" are results registered in the OUP, not local files (V3).

4.6 Running the analysis models

Analytical models are executed through standardised API calls from the application to the model-provider platforms, with harmonised input data and governed access through the OUP. Three model providers are integrated, corresponding to the sectoral models of D3.1:

- **Rana** (Nelen & Schuurmans) – flood modelling
- **Tygron** – heat stress
- **DAT.mobility** (Goudappel) – traffic modelling

The workflow for every provider is the same:

1. Select "Add Analysis" in the Data & Analyses panel
2. Choose the model provider
3. Configure the parameters in the provider-specific panel
4. Start the simulation and monitor its progress

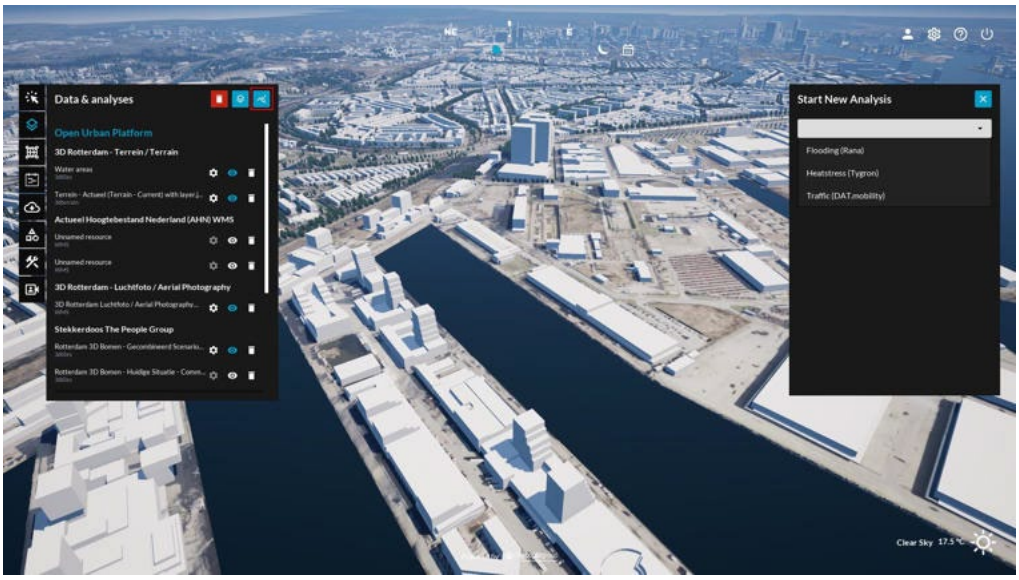


Figure 10 – "Add Analysis" with the choice of model provider.

Each provider has its own parameter panel. For Rana, for example, the panel offers a flooding preset (e.g. Event 1, T200), simulation duration, rain duration, rain intensity and tile interval. Progress is shown in the same panel.

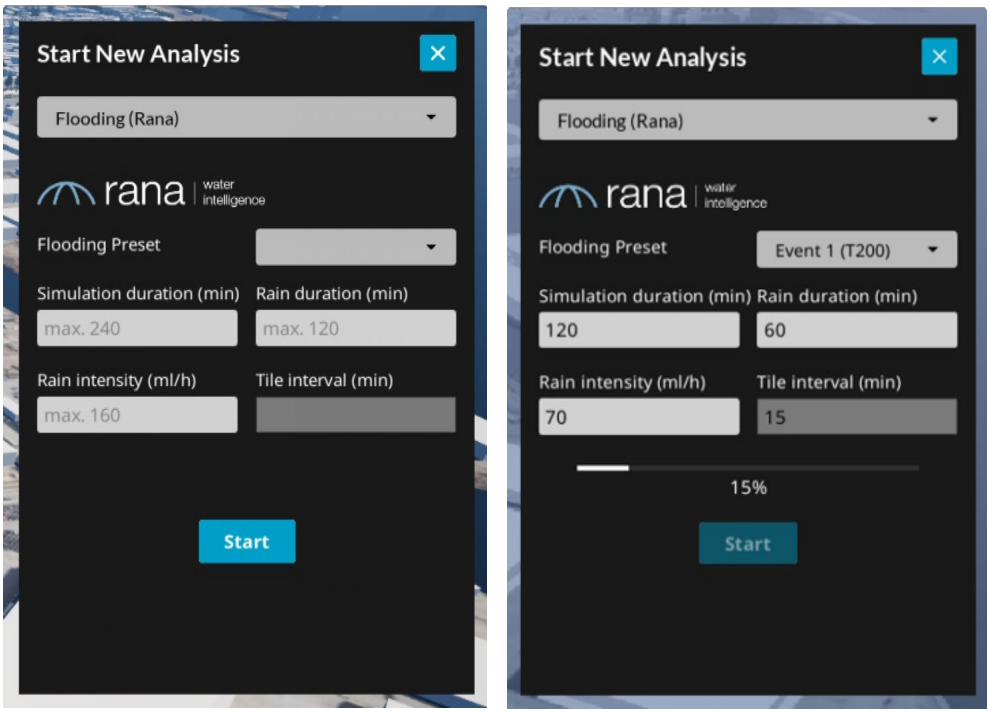


Figure 11 – Rana panel before configuration (left) and during a running simulation (right; preset Event 1 (T200), 120 min simulation, 60 min rain, 70 mm/h).

What this evidences: executable flow (V2). The run is started from the application, the provider accepts the parameters and reports progress; the parameters are provider-specific, which is why each provider has its own panel (see the lesson in Section 6.1).

4.7 Results and expert panel

After completion the result appears in the Data & Analyses panel under “Open Urban Platform Simulations” and is visualised in the 3D scene. Because the result is registered in the OUP as a service, it is available in later sessions and to other applications on the same subscription.

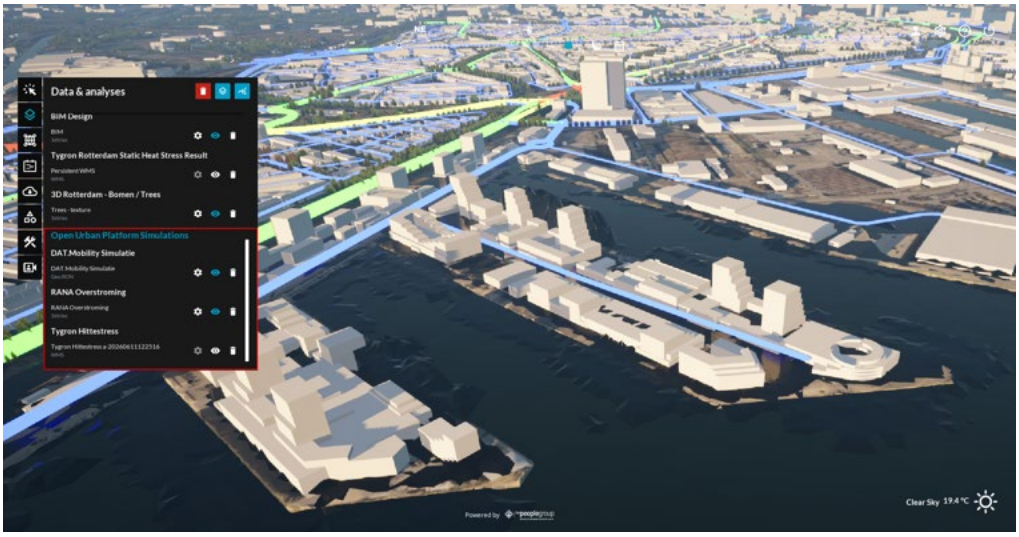


Figure 12 – Result of a simulation rendered in the 3D scene.

Each analysis has an expert panel for detailed configuration, visualisation control and interpretation. For Tygron the panel lists the result layers of the heat-stress run (e.g. PET temperature, shade, sky view factor, wind speed, urban heat island) and the moment in time (13:00) to which the result applies.

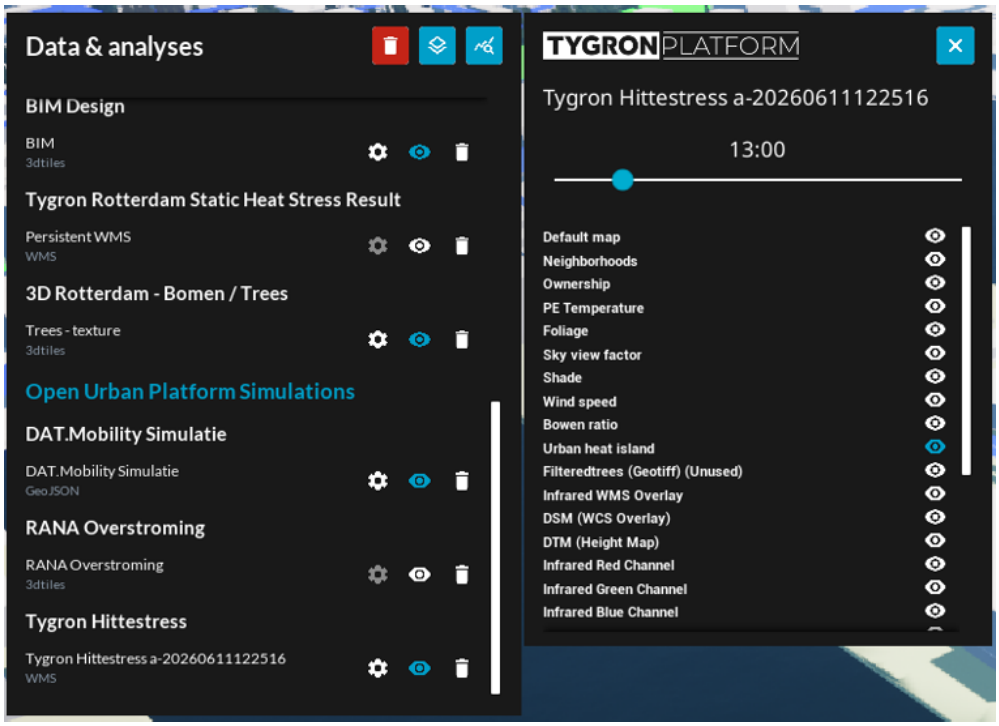


Figure 13 – Expert panel of a Tygron heat-stress run (Tygron Hittestress a-20260611122516).

What this evidences: consistent result (V3). The result carries its run identifier and time, is listed as an OUP simulation and is rendered on the objects of the loaded datasets.

5. Validation cases

5.1 Approach

For each model provider one dataflow was taken through the complete chain – source data → Common Model → provider data service in the OUP (Section 4.2) → model run set up from the application → result back in the OUP → visualisation – and checked against the four criteria of Section 3.2. The cases use the runs that are visible in the screenshots of Chapter 4, so that the walkthrough and the validation refer to the same instances. Each case records the data service, the inputs, the run, the checks with expected and observed behaviour, the comparison with the reference and the outcome. All checks were performed by the Rotterdam pilot team in the operational environment (subscription Rotterdam, digital twin DTaCoGo Rotterdam). The reference for the comparison is the earlier project-based practice described in Section 3.2. Issues found and their resolution are collected in Section 5.6.

5.2 Case 1 – Rana (flooding)

Item	Value
Model provider	Rana (Nelen & Schuurmans) – hydrodynamic flood simulation (D3.1, Section 7.2)
Digital twin / subscription	DTaCoGo Rotterdam – subscription Rotterdam
Data service (OUP)	Stekkerdoos Rana, hub PPB-DT-aCoGo-Rotterdam (Figures 2, 3)
Input datasets from OUP	Buildings 2D/3D, Pavement 2D, Trees 2D/3D, scenario Merwehaven 2D (Figure 1); terrain and drainage as mapped in D3.1 Annex 1
Reference	Earlier projects throughout Rotterdam in which the model data for this provider was built up manually; the results of those projects are the reference.
Run	Preset Event 1 (T200); simulation duration 120 min; rain duration 60 min; rain intensity 70 mm/h; tile interval 15 min (Figure 11); result registered in the OUP as “RANA Overstroming” (Figure 14)
V1 – Correct input	Expected: the model retrieves its input through its data service in the OUP; the datasets equal those of the OUP hub (Section 4.2) and correspond to the manually prepared datasets of the earlier projects. Observed: confirmed.
V2 – Executable flow	Expected: the run is set up and started from the application without manual data preparation; progress is reported; at completion the results are available. Observed: confirmed – the progress bar reports the run and at 100% the analysis results are available (Figure 11).
V3 – Consistent result	Expected: the result is registered in the OUP as a data service, is listed under “Open Urban Platform Simulations” with its run identifier and is rendered in the 3D scene at the correct time-slider position. Observed: confirmed – the result layers are available in the Unreal Digital Twin and the data services are updated and available in the OUP (Figure 14).
V4 – Governed access	Expected: the model is only reachable through the hub of the subscription the user selected at login. Observed: confirmed – all analysis models are registered as APIs in the hubs; at start-up the user logs in to the OUP and selects the hub, and the application calls the model APIs through a proxy server that identifies on behalf of the logged-in user.
Comparison with reference	Confirmed – the results are consistent with the earlier project-based runs on manually built model data.
Outcome	Confirmed – no fix required.

5.3 Case 2 – Tygron (heat stress)

Item	Value
Model provider	Tygron Platform – heat-stress calculation (D3.1)
Digital twin / subscription	DTaCoGo Rotterdam – subscription Rotterdam
Data service (OUP)	Stekkerdoos Tygron, hub PPB-DT-aCoGo-Rotterdam (Figures 2, 3); scenario Merwehaven (original data and Common Model scenario)
Input datasets from OUP	Buildings 2D/3D, Pavement 2D, Trees 2D/3D, scenario Merwehaven 2D (Figure 1); BIM design for the future situation
Reference	Earlier projects throughout Rotterdam in which the model data for this provider was built up manually; the results of those projects are the reference.
Run	Tygron Hittestress a-20260611122516 (run of 11 June 2026, 12:25); result moment 13:00; result layers include PET temperature, shade, sky view factor, wind speed and urban heat island (Figure 13); result registered in the OUP as “Tygron Hittestress” (Figure 14)
V1 – Correct input	Expected: the model retrieves its input through its data service in the OUP; the datasets equal those of the OUP hub (Section 4.2) and correspond to the manually prepared datasets of the earlier projects. Observed: confirmed.
V2 – Executable flow	Expected: the run is set up and started from the application without manual data preparation; progress is reported; at completion the results are available. Observed: confirmed – the progress bar reports the run and at 100% the analysis results are available (Figure 11).
V3 – Consistent result	Expected: the result is registered in the OUP as a data service, is listed under “Open Urban Platform Simulations” with its run identifier and is rendered in the 3D scene at the correct time-slider position. Observed: confirmed – the result layers are available in the Unreal Digital Twin and the data services are updated and available in the OUP (Figure 14).
V4 – Governed access	Expected: the model is only reachable through the hub of the subscription the user selected at login. Observed: confirmed – all analysis models are registered as APIs in the hubs; at start-up the user logs in to the OUP and selects the hub, and the application calls the model APIs through a proxy server that identifies on behalf of the logged-in user.
Comparison with reference	Confirmed – the results are consistent with the earlier project-based runs on manually built model data.
Outcome	Confirmed – no fix required.

5.4 Case 3 – DAT.mobility (traffic)

Item	Value
Model provider	DAT.mobility (Goudappel) – traffic modelling (D3.1, Section 7.3)
Digital twin / subscription	DTaCoGo Rotterdam – subscription Rotterdam
Data service (OUP)	Stekkerdoos dat.Mobility, hub PPB-DT-aCoGo-Rotterdam (Figures 2, 3)
Input datasets from OUP	Road network and mobility key figures (Kerncijfers Mobiliteit, M4H zoning plan), scenario Merwehaven 2D (Figure 1); BIM-based developments for the future scenario
Reference	Earlier projects throughout Rotterdam in which the model data for this provider was built up manually; the results of those projects are the reference.

Item	Value
Run	DAT.Mobility Simulatie – Merwehaven, scenario “traffic before” (traffic intensity before the development, Figure 14); result registered in the OUP as “DAT.Mobility Simulatie”
V1 – Correct input	Expected: the model retrieves its input through its data service in the OUP; the datasets equal those of the OUP hub (Section 4.2) and correspond to the manually prepared datasets of the earlier projects. Observed: confirmed.
V2 – Executable flow	Expected: the run is set up and started from the application without manual data preparation; progress is reported; at completion the results are available. Observed: confirmed – the progress bar reports the run and at 100% the analysis results are available (Figure 11).
V3 – Consistent result	Expected: the result is registered in the OUP as a data service, is listed under “Open Urban Platform Simulations” with its run identifier and is rendered in the 3D scene at the correct time-slider position. Observed: confirmed – the result layers are available in the Unreal Digital Twin and the data services are updated and available in the OUP (Figure 14).
V4 – Governed access	Expected: the model is only reachable through the hub of the subscription the user selected at login. Observed: confirmed – all analysis models are registered as APIs in the hubs; at start-up the user logs in to the OUP and selects the hub, and the application calls the model APIs through a proxy server that identifies on behalf of the logged-in user.
Comparison with reference	Confirmed – the results are consistent with the earlier project-based runs on manually built model data.
Outcome	Confirmed – no fix required.

5.5 Multi-model visualisation

The interoperability claim of the DT-aCoGo architecture is that results of different models can be combined in one scene. Two levels of evidence apply and are kept apart here.

Co-existence in one session. Figure 14 shows the Data & Analyses panel with three results registered in the OUP under one subscription – DAT.Mobility Simulatie, RANA Overstroming and Tygron Hittestress – next to the BIM design and the 3D Rotterdam datasets. All three are the results of the cases in Sections 5.2–5.4 and can be switched on and off independently. This evidences that results from three providers flow back through the OUP into the same session (V3).

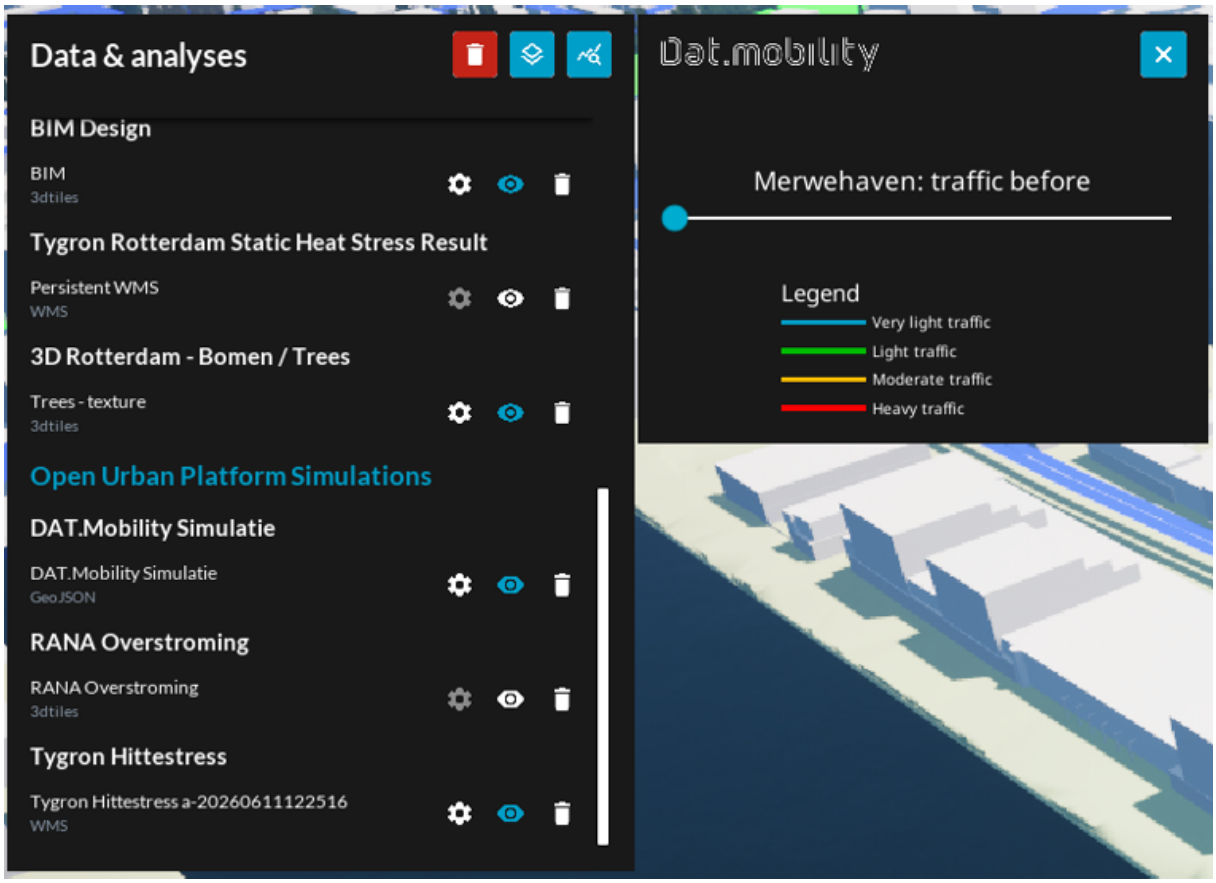


Figure 14 – Three model results in one session; the DAT.mobility traffic layer is switched on (Merwehaven, "traffic before").

Simultaneous visualisation. In Figure 14 only the traffic layer is switched on. Figure 15 shows a combined view with all three model results active at the same time. This screenshot was taken in the 's-Hertogenbosch Spoorzone twin, which runs on the same OUP hub set-up (PPB-DT-aCoGo-DenBosch, Section 4.2) and the same application: the Tygron heat-stress result, the traffic results (traffic flows and "traffic after flooding") and the flood layer are rendered together over the Spoorzone datasets, with the time slider at 09:15. It therefore evidences two things at once: results from different model providers can be visualised simultaneously in one scene, and the dataflow validated for Rotterdam replicates to a second pilot city without a different set-up.

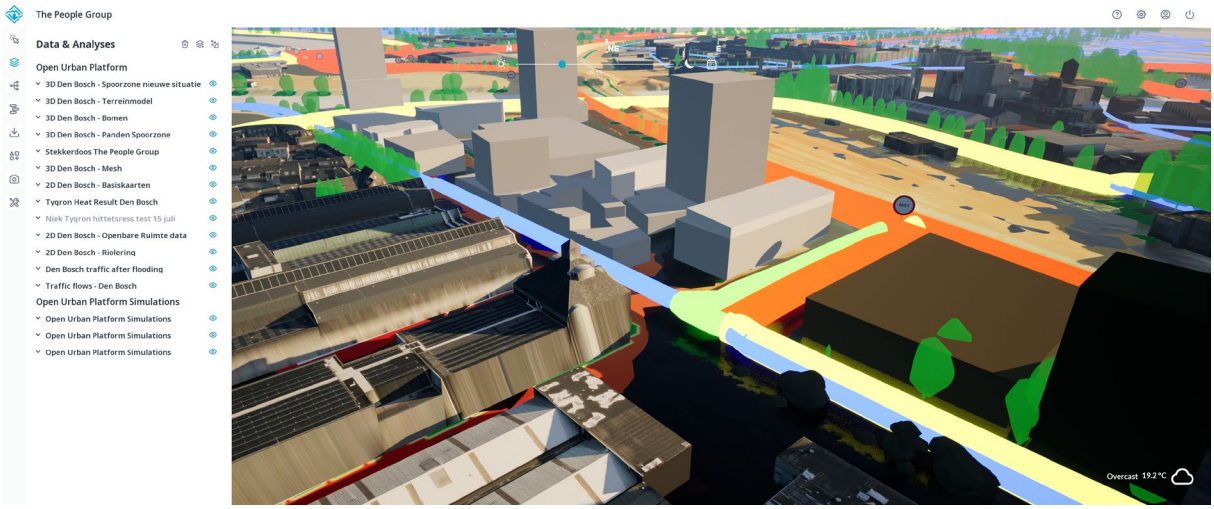


Figure 15 – Combined visualisation in the 's-Hertogenbosch Spoorzone twin: Tygron heat-stress result, traffic flows, traffic after flooding and the flood layer active simultaneously, together with the 3D Spoorzone datasets (time slider 09:15).

5.6 Issues found at the visualisation layer

The value of validating at the visualisation layer lies in the issues it surfaces that API-level tests do not. One such issue was found while running the cases above; it concerned the future situation (BIM design) and the time slider, and it was resolved before the cases were completed. No further issues were reported by the pilot team or the model providers.

Issue	Where observed	Cause	Resolution	Status
Time slider did not filter the detailed (LOD 4 / BIM) buildings for the future situation	Zooming in on the BIM design in the Unreal Digital Twin: the LOD 2 buildings were replaced by LOD 4 buildings that did not respond to the time slider	The IFC → 3D Tiles conversion of the BIM design did not carry the lifecycle attributes validFrom / validTo (D3.2) on the LOD 4 objects; at LOD 2 the IFC property sets were added with a prefix that the time-slider filter did not match	The BIM conversion was replaced by alternative tooling. The future situation now consists of LOD 4 buildings only, carrying validFrom / validTo without any prefixes from the IFC → 3D Tiles conversion; the time-slider logic works as intended	Fixed

6. Conclusions and next steps

6.1 Conclusions

The walkthrough in Chapter 4 shows that the DT-aCoGo ecosystem is operational end-to-end in the visualisation layer: the Unreal Digital Twin loads its configuration, datasets and model connections from the OUP on the basis of the user's subscription, runs three different analytical models through their APIs and brings the results back into the same 3D scene. The validation cases in Chapter 5 turn this operational evidence into validated dataflows: for each of the three model providers all four criteria V1–V4 were confirmed, the results are consistent with the earlier project-based runs on manually built model data, and the one issue found at the visualisation layer was resolved (Section 5.6).

Against the objective of the pilot – data as a service and a partly automated set-up of the climate analyses – the evidence is as follows. Data as a service is in place: the Rotterdam datasets are harmonised to the Common Model and published per provider in the OUP (Section 4.2), and the model providers retrieve their input from those services instead of from project-specific manual preparation. The automated set-up is in place for the part that runs in the digital twin application: choosing a provider, configuring the parameters and starting the run take four steps and no data handling (Section 4.6), and the models are reached through the hub of the user's subscription via a proxy that identifies on the user's behalf (V4). What remains provider-specific is the parameter panel and the interpretation of the result.

Four lessons follow specifically from working at the visualisation layer:

- **Data conversions are tested by the user, not only by the mapping.** Where the conversion of the BIM design added attribute prefixes and dropped lifecycle attributes (Section 5.6), the result was a visible inconsistency in the scene, not an API error. The visualisation layer is therefore the most effective place to detect gaps in harmonisation and conversion.
- **Temporal consistency must live in the data.** The time slider can only filter what carries lifecycle attributes; objects without validFrom / validTo are shown at the wrong moment however correct the model result is. The fix in Section 5.6 confirms the D3.2 design decision with concrete evidence.
- **API integration is standard, model execution is not.** The same four-step workflow serves all providers, but each needed its own parameter panel and result handling (Section 4.6). The recipe in D4.1 therefore describes a generic connector plus a provider-specific configuration.
- **Governance is enforced where the user is.** The subscription decides what is loaded, before any dataset or model is touched, and the proxy carries the user's identity to the model APIs (Section 4.3). Governance rules from D2.3 thus become operational without additional configuration in the application.

6.2 What this deliverable adds beyond D3.4

Aspect	D3.4 – Models validated	D3.5 – Dataflows validated in the visualisation layer
Level	API level: request, processing, response between OUP and model	User level: the complete chain as experienced in the application
Evidence	API descriptions and validation approach per model	Screenshots of the operational application and validation cases per provider with checks, comparison with the reference and outcomes

Aspect	D3.4 – Models validated	D3.5 – Dataflows validated in the visualisation layer
Time	Not in scope	Behaviour of results under the time slider; lifecycle attributes (V3)
Governance	Registration of APIs in the hub	Subscription-based loading, consent and proxy-based identification as experienced by the user (V4)
Data provisioning	Not in scope	Data services per provider in the OUP as the source of model input (Section 4.2)
Issues	Described as a process	Concrete issue found and its resolution (Section 5.6)

6.3 Show and tell

Because a document can only show a static image of an operational system, the dataflows of this deliverable will be demonstrated live in the DS4SSCC show-and-tell session on 2 October 2026. The demonstration follows the validation cases: login and subscription selection (V4), loading of the Rotterdam twin (V1), a Rana run with the T200 preset started from the application (V2), the result appearing in the Data & Analyses panel and in the scene (V3), and the combined view of the three model results (Section 5.5).

6.4 Next steps

The next phase focuses on replication, standardisation and further optimisation of the ecosystem:

- **Formalisation of the ecosystem recipe** – the configuration documented here is formalised as a replicable deployment recipe in Deliverable 4.1, including the provider-specific configuration identified in Section 6.1.
- **Consolidation of the BIM conversion** – adopting the alternative IFC → 3D Tiles tooling that carries validFrom / validTo (Section 5.6) as the standard conversion in the recipe and in the other pilot cities.
- **Scaling to additional cities and use cases** – applying the recipe in ’s-Hertogenbosch and Tampere and in follow-up projects.
- **Expansion of model integration** – integrating additional analytical models with the same generic connector.
- **Further automation of configuration and deployment** – reducing manual set-up through improved onboarding in the OUP hub.
- **Enhancement of visualisation capabilities** – performance, usability and combined visualisation of multiple results in the Unreal environment.

7. Annexes

Annex 1: DT-aCoGo-Uitleg.pdf – user manual of the Unreal Digital Twin application (in Dutch). The manual describes how to operate the application; Chapter 4 of this deliverable refers to it for the step-by-step instructions.

About the European Data Space for Smart Communities

The European Data Space for Smart Communities (DS4SSCC-DEP) initiative is a pivotal deployment following the preparatory action for a Data Space for Sustainable and Smart Cities and Communities (DS4SSCC). Emphasising sustainability aspects and diversity in communities, DS4SSCC developed a multi-stakeholder data governance scheme, created a blueprint for the European DS4SSCC, delivered priority datasets, and developed a roadmap towards a mature DS4SSCC.

DS4SSCC-DEP's vision revolves around creating a well-governed federated territorial, place-based data space for smart communities available for developers and infrastructure providers, aligning with the prospects outlined in Europe's Digital Decade objectives. This approach distinguishes itself from sectoral data spaces by encompassing diverse domains, and underpinning governance across all levels of society. The key objectives of the project are:

- **Objective 1: Establishing a Federated and Innovative Data Space**. This phase focuses on creating a large-scale data space controlled by public data holders, ensuring alignment with the Smart Middleware Platform and broader data space ecosystems. It also aims to offer middleware service solutions facilitating data sharing and management while refining the blueprint based on gained experience.
- **Objective 2: Pilots and Refinement of the Data Space Blueprint**. Through 10-12 cross-sectoral data pilots in various EU communities, this objective aims to validate and refine the data space blueprint. Using common high-value datasets, it seeks to create added value by combining data from various domains, contributing to the sustainability plan, and minimising infrastructure investment.
- **Objective 3: Fostering Innovation and Ecosystem Alignment**. Engaging stakeholders, compliance with sector-specific legislation, establishing links with Horizon Europe missions, creating services available via trusted application catalogues, and enhancing ethical considerations and AI-enabled local solutions constitute the focus of this objective.

DS4SSCC-DEP, building upon established European networks, aims to significantly contribute to the sustainability goals of European citizens. It emphasises co-creation, standardisation, business models, strategies for data spaces, and envisions a federated platform.

DS4SSCC-DEP consortium includes the following partners: