



Geonovum Invitation to tender

Testbed 2026, phase 2 – testbed description
and implementation topics

Version 1.0
Date 7-7-2026

Revision history	
Revision date	notes
23-juni-2026	First draft
30-juni-2026	Draft 2
6-juli-2026	Version 1

Table of contents

Chapter 1 - Introduction	4
Invitation to tender	4
Background	4
Goal of the testbed	4
Scope 5	
Outcome	5
This document	5
Chapter 2 - How to tender	6
Rules and procedure	6
Chapter 3 - Use case	7
Chapter 4 - Adoption topic #1: Implement and validate an OGC API Processes service	7
4.1 Goal	7
4.2 Description	7
4.3 OGC API Processes version	7
4.4 Task	8
4.5 Deliverables	8
4.6 Requirements / standards / Open-Source software	8
4.7 Relevant use-cases	8
Chapter 5 - Adoption topic #2: Complete and validate an existing OGC API Processes service	9
5.1 Goal	9
5.2 Description	9
5.3 OGC API Processes version	9
5.4 Task	9
5.5 Deliverables	9
5.6 Requirements / standards / Open-Source Software	9
5.7 Relevant use cases	10
Chapter 6 - Adoption topic #3: Implement and validate OGC API Processes client	10
6.1 Goal	10
6.2 Description	10
6.3 OGC API Processes version	10
6.4 Task	10
6.5 Deliverables	10
6.6 Requirements / standards / Open-Source Software	11
6.7 Relevant use cases	11
Chapter 7 - Research topic #4: App-Store / Recipes	11
7.1 Goal	11
7.2 Description	11
7.2.1 Recipes	12
7.2.2 Catalogs	12

7.2.3	Metadata	12
7.2.4	Recipe metadata	13
7.3	Task	13
7.4	Deliverables	13
7.5	Requirements / standards / Open-Source Software	14
7.6	Relevant use cases	14
Chapter 8 - Research topic #5: Web 3D Context document (export/import)		14
8.1	Goal	14
8.2	Description	14
8.3	Task	15
8.4	Deliverables	15
8.5	Requirements / standards / Open-Source Software	15
Chapter 9 - Testbed organization		15
9.1	Coordination	15
9.2	Open testbed sessions	15
9.3	Planning	16
9.4	Finance	16
Appendix A: Metrics		17
Appendix B: Technical details		19

Chapter 1 - Introduction

This chapter gives the general background to the testbed Geonovum is organizing and specifies its goals and scope.

Invitation to tender

This document gives information about the invitation to tender regarding the following implementation topics:

- Adoption topic #1: **Implement and validate OGC API Processes service**
- Adoption topic #2: **Complete and validate an existing OGC API Processes service**
- Adoption topic #3: **Implement and validate OGC API Processes client**
- Research topic #4: **App-Store / Recipes**
- Research topic #5: **Implement export & import of Web 3D Context Document**

Budget (including VAT):

- Topic #1
 - o 2 awards for each 12.500 euro
- Topic #2
 - o 2 awards for each 12.500 euro
- Topic #3
 - o 2 awards for each 12.500 euro
- Topic #4
 - o 3 awards for each 25.000 euro
- Topic #5
 - o 2 awards for each 12.500 euro

Background

Processes (aka calculation, analysis modules) and agentic AI (part of Testbed 2026 phase 3, to be tendered later this year) are what set a Digital Twin of the Physical Living environment apart from a GIS or BI environment. Being able to simulate, analyze, predict and calculate are essential components to determine KPI's so that policy makers can make informed decisions.

Having open, proven, standards-based, and API-enabled (machine-first) processes allows the process (a calculation or analysis) to be taken up in workflows and automations. However, the human remains in the loop and takes the ultimate decision.

Goal of the testbed

We want to encourage the adoption of uniform interoperable standards for Digital Twins, based on the nLDT architecture (data, processes and visualization, see <https://geonovum.github.io/NLDT-Architectuur/>). In this tender document, the focus is on interoperable processes. With these research and adoption topics, we want to create sufficient 'critical mass' – enough organizations in an ecosystem that work with the same interoperability arrangements. This will lead to an interoperable, networked system of digital twins for the physical environment.

The adoption topics use established API specifications, that have a high level of maturity. The research topics do not have the same maturity level: here we are identifying (and eliminating) the obstacles for interoperability.

Geonovum, in line with its mission, is keen to promote the use of standards, drive adoption and foster research; and it seeks to involve the market to do so. The actual questions and issues to be addressed are described in this document as research and adoption topics.

Scope

In this testbed we focus on implementing the OGC API Processes standard. There is no centralized use-case, we request the applicants to bring their own use case.

Outcome

The results of the testbed are intended to contribute to expanding and innovating the Dutch public sector Spatial Data Infrastructure in a direction that considers the possibilities in the market today. Based on the outcome of this testbed we may take steps to make the OGC API Processes standard mandatory within the Dutch public sector, to promote interoperability between digital twins and encourage reuse of processes and calculation models.

To this end, implementations of standards realized within this testbed will be kept online and available for further experimentation for 6 months after the testbed ends.

The results of this testbed will be demonstrated at the Barcelona Smart City Expo (nov 3-5 2026). Participants are expected to make demo material available (short demo video's). These video's will be made 2 weeks before the Barcelona event.

This document

After this introduction, chapter 2 explains the tender procedure. Chapter 3 provides description of the use cases. Chapters 4, 5, and 6 describe the adoption topics in detail. Chapter 7 and 8 describe the research topics in detail. Chapter 9 explains the organization of the testbed in more detail. Appendix A gives the metrics by which tenders are judged. Appendix B describes the AppStore steps to be taken to get results (sequence diagram).

This document is a draft. Based on questions and comments during and after the tender period, we will update this document to clarify questions and remove errors. A final draft will be made available after the public information session took place which is now scheduled on July 16th.

Chapter 2 - How to tender

This chapter gives information about the procedure of tender response.

Rules and procedure

The submission period for the tender starts on July 7th, 2026 with the publication of the Invitation to Tender on Geonovum's website, <http://www.geonovum.nl>.

The tender is open to private and public parties, and to combinations of parties (consortia). In the case of a consortium, there is one party who acts as the contact point and contractor on behalf of the consortium for the tender with Geonovum.

Questions about the tender can only be asked by sending an e-mail to info@geonovum.nl, addressed to Bart De Lathouwer, coordinator of the testbed. Questions should be submitted by Wednesday July 15th. These questions and our answers are collected in an Information Note. We will organize an informational meeting on Thursday 16th of July at 9:30 am. Registration for the meeting can be done through the Geonovum website: <https://www.geonovum.nl/agenda/testbed-phase-2-public-information-session-online>. The minutes of this meeting will be part of the Information Note. At the latest, this note is published on the website of Geonovum on Monday, July 20th.

Your tender must be submitted by sending an e-mail to info@geonovum.nl, addressed to Friso Penninga, director of Geonovum.

The tender is preferably written in English¹ and must at least contain:

- The implementation topic or topics you are applying for;
- Motivation for the implementation topic or – topics you are applying for;
- Plan of approach for each addressed implementation topic (maximum of four pages per implementation topic);
- References (including e.g. publications, projects, blogs, code on GitHub) and curriculum vitae for performers of the research, showing enough relevant knowledge and experience;
- An indication of the in-kind investment;
- Statement of agreement with the publication of the research results and deliverables under a CC/by license.

All testbed outcomes will be available under <http://creativecommons.org/licenses/by/4.0/>.

Deliverables of the implementation topics, in the form of published data, vocabularies, demonstrators, prototypes and the like, must remain available for at least six months after completion of the testbed.

For the research topics (topics 4 and 5), the source code is preferably available under a “popular and widely used or with strong communities” open-source license <https://opensource.org/licenses/category>. The use of other (non-opensource) licenses will be considered only if well motivated. For the adoption topics (topics 1, 2 and 3), closed source and open source are both acceptable, but open source is encouraged. The software components developed for the adoption topics must become part of the normal market offering, at no additional cost.

¹ The alternative is Dutch

The deadline for submitting a tender is Friday, August 14th, 2026.

Geonovum will judge the received tenders on Monday 17th and Tuesday 18th of August, according to the criteria stated in appendix A.

Parties are allowed to tender for more than one implementation topic. However, a contractor is only awarded one implementation topic, not several. The reason for this is our wish to gain different insights from different parties. The only exception to this rule is that additional implementation topics can be awarded to a single party if this party is the only bidder for an additional topic. We will only do this if the bidder agrees.

Geonovum will announce which party is selected for which implementation topic on Wednesday 19th of August. All parties who have submitted a tender will also be informed about this via e-mail. On Thursday 20th of August 9:30 – 13:00 hours the official kick-off meeting will take place in Amersfoort.

Note that reviewers of this document and Geonovum staff² are exempt from bidding.

Chapter 3 - Use case

Testbed 2026 phase 2 does not come with a central use-case for the participants to use. Rather we request that the participants bring their own use-case.

During the execution of the Testbed, we will not actively seek to bring use-case together, but participants are encouraged to work together and seek use-case alignment.

Chapter 4 - Adoption topic #1: Implement and validate an OGC API Processes service

4.1 Goal

Make an existing calculation module work according to the nLDT reference architecture (<https://github.com/Geonovum/NLDT-Architectuur>) and have it interoperate with software from other software vendors.

4.2 Description

Add a full implementation of the OGC API Processes specification to an existing calculation module. With “full” implementation we mean supporting process input & output descriptions, both Sync and Async execution mode, and job management. The implementation must pass the OGC compliance check and pass Geonovum’s OGC-checker with no faults.

4.3 OGC API Processes version

OGC API Processes v1.0 is the approved version of the API, and v2.0 is a stable draft. With v2 compatible with v1, we want to encourage using version 2 of the specification.

² Employees and Secondments

4.4 Task

- Implement the OGC API Processes on top of your existing calculation module
 - Define the process metadata, describing input & output parameters
 - Implement sync execution path
 - Implement async execution path, including callbacks
- Test the implementation against the <https://cite.ogc.org/teamengine> (v1), no errors
- Validate the implementation against the Geonovum Linter: <https://geonovum.github.io/ogc-checker/#/ogc-api-processes> (v1) with no errors.

4.5 Deliverables

- A written report of the findings encountered during implementation. Geonovum will provide a Github repository where the report can be added in markdown format. We will act as editors and consolidate all reports into one Respec/HTML document.
- The software component developed under this contract must be made available in the standard market offering, at no additional cost.
- A demo service (with demo data) should be kept available for the public at least until 6 months after the testbed ends.
- Short demo video's

4.6 Requirements / standards / Open-Source software

Mandatory Standards: <https://ogcapi.ogc.org/processes/>

OGC API Processes v1.0 is the approved version of the API, and v2.0 is a stable draft. We want to encourage using version 2 of the specification.

Optional available Open-Source Software (for inspiration and as example code for testing):

For OGC API Processes open source client and server implementations are listed in the OGC Github repository: <https://github.com/opengeospatial/ogcapi-processes/blob/master/implementations.adoc>

The above software is listed for convenience; there may be more available.

4.7 Relevant use-cases

Testbed 2026 phase 2 does not come with a centralized, mandatory use-case. We request the implementor to bring their own use-case – preferably from an existing customer.

Geonovum will organize testing events (plugfests) around specific technical topics (supported by use-cases), where your participation is mandatory (2 plugfests during 2026). Each plugfest is a physical 1-day event.

Chapter 5 - Adoption topic #2: Complete and validate an existing OGC API Processes service

5.1 Goal

Complete an existing API Processes implementation, make it function within the nLDT ecosystem and have it interoperate with software from other software vendors.

5.2 Description

Complete an existing implementation of the OGC API Process specification to a full implementation on top of an existing calculation module. With “full” implementation we mean supporting process input & output descriptions, both Sync and Async execution mode and job management. The implementation must pass the OGC compliance check and pass Geonovum’s OGC-checker with no faults.

5.3 OGC API Processes version

OGC API Processes v1.0 is the approved version of the API, and v2.0 is a stable draft. With v2 compatible with v1, we want to encourage using version 2 of the specification.

5.4 Task

- Complete the OGC API Processes implementation on top of your existing calculation module
 - Define the process metadata, describing input & output parameters
 - Implement sync execution path
 - Implement async execution path, including callbacks
- Test the implementation against the OGC validator (v1), no errors
- Validate the implementation against the Geonovum Linter: <https://geonovum.github.io/ogc-checker/#/ogc-api-processes> (v1) with no errors.

5.5 Deliverables

- A written report of the findings encountered during implementation. Geonovum will provide a Github repository where the report can be added in markdown format. We will act as editors and consolidate all reports into one Respec/HTML document.
- The software component developed under this contract must be made available in the standard market offering, at no additional cost.
- A demo service (with demo data) should be kept available for the public at least until 6 months after the testbed ends.
- Component is ready to do taken up in an AppStore
- Short demo video’s

5.6 Requirements / standards / Open-Source Software

Mandatory Standards: <https://ogcapi.ogc.org/processes/>

OGC API Processes v1.0 is the approved version of the API, and v2.0 is a stable draft. We want to encourage the use of version 2 of the specification.

Optional available Open Source Software (for inspiration and as example code for testing):

For OGC API Processes open source client and server implementations are listed in the OGC Github repository: <https://github.com/opengeospatial/ogcapi-processes/blob/master/implementations.adoc>

The above software is listed for convenience; there may be more available.

5.7 Relevant use cases

Testbed 2026 phase 2 does not come with a centralized, mandatory use-case. We request the implementor to bring their own use-case – preferably from an existing customer.

Geonovum will organize testing events (plugfests) around specific technical topics (supported by use-cases), where your participation is mandatory (2 plugfests during 2026). Each plugfest is a physical 1-day event.

Chapter 6 - Adoption topic #3: Implement and validate OGC API Processes client

6.1 Goal

Implement an OGC API Processes client, make it function within the nLDT ecosystem and have it interoperate with software from other software vendors.

6.2 Description

Create a full implementation of an OGC API Process specification client in your visualization software. With “full” implementation we mean supporting process input & output descriptions, both Sync and Async execution mode, and job management.

6.3 OGC API Processes version

OGC API Processes v1.0 is the approved version of the API, and v2.0 is a stable draft. With v2 compatible with v1, we want to encourage using version 2 of the specification.

6.4 Task

- Implement OGC API Processes client
 - Generate a UI from the process definition
 - Start execution of the process
 - Sync
 - Async
 - Dismiss
 - For async processing, use the callbacks to monitor the job
 - Show result in your visualization software

6.5 Deliverables

- A written report of the findings encountered during implementation. Geonovum will provide a Github repository where the report can be added in markdown format. We will act as editors and consolidate all reports into one Respec/HTML document.
- The software component developed under this contract must be made available in the standard market offering, at no additional cost.

- A demo client-side service should be kept available for the public at least until 6 months after the testbed ends.
- Component is ready to do taken up in an AppStore
- Short demo video's

6.6 Requirements / standards / Open-Source Software

Mandatory Standards: <https://ogcapi.ogc.org/processes/>

OGC API Processes v1.0 is the approved version of the API, and v2.0 is a stable draft. We want to encourage the use of version 2 of the specification.

Optional available Open-Source Software (for inspiration and as example code for testing):

For OGC API Processes open-source client and server implementations are listed in the OGC Github repository: <https://github.com/opengeospatial/ogcapi-processes/blob/master/implementations.adoc>

The above software is listed for convenience; there may be more available.

6.7 Relevant use cases

Testbed 2026 phase 2 does not come with a centralized, mandatory use-case. We request the implementor to bring their own use-case – preferably from an existing customer.

Geonovum will organize testing events (plugfests) around specific technical topics (supported by use-cases), where your participation is mandatory (2 plugfests during 2026). Each plugfest is a physical 1-day event.

Chapter 7 – Research topic #4: App-Store / Recipes

7.1 Goal

Offer ready-made solutions for real-world problems, expressed as a ‘recipe’: a combination of data-sources (ingredients) and operations in the form of an ordered list of processes (like cooking instructions).

7.2 Description

We are familiar with the App Store (and Play Store) from our mobile phones (smartphones, smart devices): here we find all kinds of applications ('Apps', applications) that, after installation, work immediately and are integrated with the smartphone ecosystem.

The Windows and macOS operating systems also have an App Store – where apps are promoted but also organized into categories to make searching easier. Just like on smart devices, installation is flawless and the application works seamlessly within the operating system's ecosystem.

Payment is easily arranged in both the App Stores of smart devices and those of Windows or macOS – digital payments can be made via all kinds of payment methods.

App settings are stored in the operating system; when the app is used on another device by the same user, it is immediately recognizable.

Digital Twins also operate within their own ecosystem (and together form an ecosystem). Digital Twins are defined using computational modules and an App Store in the context of Digital Twins, it should offer easy-to-install-and-use computational modules.

Whereas the App Store in smart devices uses the native functions of Windows or macOS, the App Store in Digital Twins uses HTTP as the primary means of communication.

7.2.1 Recipes

Recipes can be expressed in various ways. OGC API Processes part 2 <https://docs.ogc.org/DRAFTS/20-044.html>. Common Workflow Language (CWL), a standard that specifies the document schema and execution semantics for composing workflows from components such as command line tools and other workflows. This could be a way to describe recipes. Another proposal for recipe description was created within nLDT. This proposal is not yet standardized and could be used as a basis or inspiration source for experimental implementation. (see appendix B for link to the experimental implementation) A third option for describing recipes is Arazzo, a community-driven open specification within the OpenAPI Initiative. (<https://www.openapis.org/arazzo-specification>)

7.2.2 Catalogs

A catalog serves as a central hub where datasets and services are collected, documented, and made accessible. It offers a clear structure in which information is described via metadata, such as title, subjects, origin, and format. This allows users to quickly discover what data is available and how to utilize it, without having to search through separate documents or scattered sources.

In addition to organizing data, a catalog offers functions for searching, filtering, and comparing services. Users can filter using search terms, categories, or tags to find exactly what is relevant to their application. The system often supports multiple types of services—from open data to internal administrative records—and can process various data types and files, such as tables, geographic data, or API connections.

An important aspect of such a catalog system is the focus on accessibility and reusability. Through standardization, clear documentation, and reliable connections, it becomes easy to integrate data into analyses, applications, or visualizations. Moreover, the platform encourages collaboration between departments, organizations, or developers by making datasets findable and understandable, which contributes to transparency and innovation.

7.2.3 Metadata

Metadata consists of descriptive data that helps to find, understand, and correctly interpret datasets and services. They include information about the content, origin, structure, and quality of the data, among other things. By capturing this context, it becomes clear to users exactly what a dataset contains, how current or reliable it is, and in which situations it is suitable for use. In this way, metadata acts as a kind of user manual for data.

Standards are often used to record metadata in a structured and interchangeable manner. A widely used standard is DCAT, a vocabulary developed to describe services. DCAT defines uniform terms and relationships, making services easier to publish, search, and link between different platforms.

Through this standardization, organizations can keep their data descriptions consistent and promote interoperability.

In addition to improving findability and consistency, metadata also contributes to transparency and governance. They clarify who is responsible for a dataset, which license applies, how it is updated, and what restrictions apply. As a result, users can better assess whether the data is suitable for their purposes, and organizations maintain an overview of their own information management.

7.2.4 Recipe metadata

DCAT is a mature and widely used standard, and the Netherlands has a standardized DCAT profile: DCAT-AP-NL. However, DCAT-AP-NL is designed to describe datasets and services. It does not currently support metadata for recipes: combinations of data sources and operations that together produce a result.

For users to find recipes in a catalog, they must be described using metadata, just like datasets and services. How to capture recipe metadata in DCAT-AP-NL is not yet fully defined.

There are standards extending DCAT to describe properties like Recipes. We are investigating DPROD (Data Product Ontology (DPROD)) and APKG (GitHub - ILIAD-ocean-twin/APKG: Application Package Application Profile · GitHub).

7.3 Task

- Create metadata in extended DCAT-AP-NL that describes a recipe (the current version of DCAT-AP-NL does not support recipes, and therefore we expect a suggestion from the tenderer). Research which extra metadata facets are needed to describe a recipe and propose how to capture these.
- Publish this metadata in a catalog
 - This catalog should implement OGC API Records (see <https://geonovum.github.io/DTaaS-Testbed2/#catalogues-with-interoperable-api-s-5-1>)
 - See appendix B for details
- Implement a Cookbook
 - A cookbook is a collection of recipes, served over an API (OGC Building Blocks)
 - An actual recipe (e.g. in CWL, Arazzo) should be served from here.
 - See appendix B for details
- Implement a 'Cook': a recipe executor which:
 - Reads the recipe
 - Does the 'graph traversal'
 - Executes the described processes
 - Links output of process 'a' to input of process 'b'
 - See appendix B for details
- Implement a client (like the client in topic #3) that can find recipes and bind/execute.

Participants are expected to work together to document a) the technical specifications b) what they expect from each-other – how they worked together.

7.4 Deliverables

- A written report; Geonovum will provide a Github repository where the report can be added in markdown format. We will act as editors and consolidate all reports into one Respec/HTML document.
- If any software component is developed, we prefer the code to be made available in an open-source repository and pull requests to existing OS projects in case these are used.
- The solution should be kept available for the public at least until 6 months after the testbed ends.

- Results are input for an AppStore
- Short demo video's
 - A viewer can find the Metadata, bind the recipe and execute the recipe – the results are visualized in the viewer.

7.5 Requirements / standards / Open-Source Software

Mandatory Standards:

- OGC API Processes
- OGC API Records
- DCAT-AP-NL

More on orchestration to be found as a deliverable from Testbed2

(https://github.com/Geonovum/DTaaS-Testbed2/blob/main/deliverables/DTaaS2_Orchestration-Avineon_Tensing.pdf).

OGC API Processes v1.0 is the approved version of the API, and v2.0 is a stable draft. We want to encourage the use of version 2 of the specification.

Proposed Standards:

- CWL (<https://www.commonwl.org>)
- Arazzo (<https://www.openapis.org/arazzo-specification>)
- Roll-your own (<https://github.com/Geonovum/nLDT-Recipe>)

NOTE: We intend to award multiple participants for this research topic. Our aim is to have at least one of each for these Recommended Standards.

7.6 Relevant use cases

Testbed 2026 phase 2 does not come with a centralized, mandatory use-case. We request the implementor to bring their own use-case – preferably from an existing customer.

Geonovum will organize testing events (plugfests) around specific technical topics (supported by use-cases), where your participation is mandatory (2 plugfests during 2026). Each plugfest is a physical 1-day event.

Chapter 8 – Research topic #5: Web 3D Context document (export/import)

8.1 Goal

The nLDT ecosystem knows a lot of visualization components, each having their own strengths and specific capabilities (including AR/VR). The goal of this research topic is the ability to export scene information from a visualization component and the ability to import that scene information in another visualization component.

8.2 Description

The 3D Context Document states how a specific grouping of one or more maps from one or more map servers can be described in a portable, platform-independent format for storage in a repository or for transmission between clients. This description is known as a Web 3D Context

Document,” or simply a “3d Context.” A 3D Context document includes information about the services(s) providing data in the overall view, the 3D viewpoint, time, bounding box and map projection shared by all the data, sufficient operational metadata for Client software to reproduce the 3D viewpoint, and ancillary metadata used to annotate or describe the maps and their provenance for the benefit of human viewers. A 3D Context document is structured using JSON.

The work should continue to build on the export-import-scenes:
<https://geonovum.github.io/DTaaS-Testbed2/#export-import-scenes-6-2-0>.

8.3 Task

- Implement an exporter, exporting a scene from visualization component ‘a’
- Implement an importer, importing the scene in visualization component ‘b’

8.4 Deliverables

- A written report; Geonovum will provide a Github repository where the report can be added in markdown format. We will act as editors and consolidate all reports into one Respec/HTML document.
- If any software component is developed, we prefer the code to be made available in an open source repository and pull requests to existing OS projects in case these are used.
- The solution should be kept available for the public at least until 6 months after the testbed ends.
- Short demo video’s

8.5 Requirements / standards / Open-Source Software

Mandatory documents:

https://github.com/Geonovum/DTaaS-Testbed2/blob/main/deliverables/Digital_Twin_as_a_Service%E2%80%93Testbed_II-Import_Export_Scenes.pdf (also to be found in the DTaaS-Testbed2 repository, under deliverables).

NOTE: We encourage submitting with multiple parties for this topic, each offering a visualization component for use with 3D Context Documents.

Chapter 9 - Testbed organization

This chapter describes the organization, conditions, finance and planning of the testbed.

9.1 Coordination

The coordinator from Geonovum is Bart De Lathouwer (Geonovum), with support roles for Linda van den Brink and Niels Hoffmann (both Geonovum).

For every implementation topic there is a bi-weekly meeting between Geonovum and each contractor, either at the Geonovum office in Amersfoort or online. The agenda items of these meetings are the progress made and any issues or technical questions concerning the details of the implementation topic.

9.2 Open testbed sessions

Work on the implementation topics will be carried out in parallel. During this time, Geonovum wants the implementation topics to inform each other as much as possible. We expect there to be 5

parallel topics, each with multiple contractors. For this reason, Geonovum will organize two sessions (max. 1/2 day each) with the contractors of all implementation topics. This will be done for the purpose of aligning and sharing developments and knowledge between the implementation topics.

These sessions will be public; the contractors will present their intermediate results to each other and an open group. Anyone who is interested can be present at these meetings. This group has the possibility to discuss in an open way the results with the contractors. The insights gained from this will be used as much as possible by the contractors in their further work on the implementation topics.

In addition to these meetings, Geonovum will organize a larger public session after completion of the testbed, in which the contractors have the opportunity to present their final results.

9.3 Planning

Geonovum will announce which party is selected for which implementation topic on Wednesday 19th of August (see chapter 2). The testbed starts immediately afterwards with the physical kick-off meeting on Thursday 20th of August in Amersfoort, as do the bi-weekly meetings between Geonovum and each contractor.

The open testbed sessions with contractors will take place in the middle of September and October.

In August we will focus on synergy between the contractors and the use case they implement. September will focus on realization and in October the focus will be on reporting results.

9.4 Finance

Budget (including VAT):

- Topic #1
 - o 2 awards for each 12.500 euro
- Topic #2
 - o 2 awards for each 12.500 euro
- Topic #3
 - o 2 awards for each 12.500 euro
- Topic #4
 - o 3 awards for each 25.000 euro
- Topic #5
 - o 2 awards for each 12.500 euro

For a total of 175.000 euro (includes VAT).

These budgets are intended as a contribution towards the research & adoption activities of the contractor. The budgets allow each contractor to carry out research and implementation activities and to develop demonstrators to try things out. The budget is not supposed to cover the entire research & implementation activities of the contractors; an in-kind contribution from the contractors is expected.

Appendix A: Metrics

The criteria by which tenders are judged are:

- The quality of the plan of approach.
- The quality of the references and CVs.
- Affinity with data (publication) and standards.
- Impact on existing workflow in terms of quality, cost, etc. and general architecture.
- Contributions towards EU and NL govt. information services and flows.
- Planned dissemination of the created work; documentation and 'how to' documents, publication strategy relative to the community, licenses used.

Tenders are judged by two metrics: a general score against the key goals of this document (75%), and a further 25% for key elements specific to each of the testbeds.

All adoption tenders will be scored according to the following metrics:

		Weight
Overall	Contribute to the "Goal of the testbed"	10%
25%	Applicability to the "Scope" as defined	5%
	Further open standards and interoperability	10%
Architecture	Applicability of use case to nLDT	5%
20%	Re-useability after testbed ends	15%
Proposal	Plan or approach	10%
30%	Conciseness and specificity of the plan	5%
	Portfolio, References and CV	5%
	Dissemination, Licenses, lasting effect of outreach beyond geo community	10%

A further 25% is awarded for each of the testbeds specific key goals:

Specific	Adoption topic #1: Implement and validate API Processes service	25%
25%	Adoption topic #2: Complete and validate an API Processes service	25%
	Adoption topic #3: Implement and validate API Processes client	25%

All research tenders will be scored according to the following metrics:

		Weight
Overall	Contribute to the “Goal of the testbed”	10%
25%	Applicability to the “Scope” as defined	5%
	Further open standards and interoperability	5%
	Collaboration across multiple parties	10%
Architecture	Applicability of use case to nLDT	5%
20%	Re-useability after testbed ends	10%
Proposal	Plan or approach	10%
30%	Conciseness and specificity of the plan	5%
	Portfolio, References and CV	5%
	Dissemination, Licenses, lasting effect of outreach beyond geo community	10%

A further 25% is awarded for each of the testbeds specific key goals:

Specific	Research topic #4: App-Store / Recipes	25%
25%	Research topic #5: Web 3D Context document (export/import)	25%

In tenders for research topic #4, we intend to award multiple participants. Our aim is to have at least one of each for the Recommended Standards for recipes. Here, we are looking for diversity of approaches: we want to maximize what we learn so will most likely not award the same approach twice.

For research topic #4, we encourage submitting with multiple parties (1 main contractor), each working on a part of the App-Store components. We are looking for diversity with the aim of achieving interoperability between components.

For research topic #5, we encourage submitting with multiple parties (1 main contractor), each offering a visualization component for use with 3D Context Documents. Again, we are looking for diversity, in this case of visualization clients, with the aim of testing interoperability.

Appendix B: Technical details

Sequence diagram:

<https://github.com/Geonovum/nLDT-AppStore/blob/main/README.md>

Component decomposition:

<https://github.com/Geonovum/nLDT-AppStore/blob/main/RollYourOwn.md>