**Destination Renewable Energy (DRE)**

Software Reuse File

Ref.: DestinE\_ESA\_DRE\_ SRF\_v5.0



**Author's Table**

|  |  |  |
| --- | --- | --- |
| Written by: | George Koutalieris  Symeon Symeonidis  Vasillis Perifanis  Athanassios Drivas  Kyriaki Papachristopoulou  Rizos-Theodoros Chadoulis  Aggelos Georgakis  Christos Stathopoulos  Platon Patlakas | ENORA INNOVATION  ENORA INNOVATION  ENORA INNOVATION  NOA  NOA  NOA  NOA  WeMET  WeMET |
| Verified by: | Theodora Papadopoulou | NOA |
| Approved by: | Haris Kontoes | NOA |

**Change Log**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Reason for change** | **Section(s) changed** |
| 1.0 | 13 February 2024 | Creation of the document |  |
| 2.0 | 14 May 2024 | New Version | Edits in the entire document |
| 3.0 | 27 July 2024 | New Version | Edits in the entire document |
| 4.0 | 24 October 2024 | New Version | Edits in the entire document |
| 5.0 | 12 May 2025 | Fifth Version | Edits in the entire document |

**Table of Contents**

[**1 Introduction 5**](#_heading=h.ho8se5379cuo)

[1.1 Purpose of the Software Reuse File 5](#_heading=h.j7cyo13oal6w)

[1.2 Scope of Reusable Components 5](#_heading=h.6u5f99kf89ip)

[**2 Software Reuse Overview 6**](#_heading=h.hubedo1p4rrd)

[**3 Libraries and Frameworks 7**](#_heading=h.30j0zll)

[3.1 Persistence Layer 7](#_heading=h.m6j91rqdl5r6)

[3.2 Backend/API 7](#_heading=h.odnxx1v4eh)

[3.3 API Specification for Solar and Wind Assessment 9](#_heading=h.ebokhlp2glyo)

[3.4 API Specification for Solar and Wind Forecasting 12](#_heading=h.iamjz8kbvo4t)

[3.5 Frontend/Web Application - UI/UX 14](#_heading=h.xl2w7le62gej)

[**4 Conclusion and Future Directions for Software Reuse 15**](#_heading=h.l37utfprmn67)

# Introduction

## Purpose of the Software Reuse File

This Software Reuse File version 5.0 helps as a guide for maximising the efficiency and sustainability of software development within the Destination Renewable Energy (DRE) Use Case. It outlines strategies for reusing software components, thereby reducing development time, cost, and risk while enhancing system quality and innovation.

## Scope of Reusable Components

The scope encompasses the software (SW) components that can be used to develop and deploy DRE Software.

# Software Reuse Overview

The DRE consortium has explored various alternatives in the current stage of availability of the DESP Core services and the DRE Use Case On-Boarding status for the scheduled deliverable of the DRE release 5. These alternatives are aimed at facilitating the validation process of the prototype application. Such alternatives encompass, but are not limited to, the creation of demonstration users, the incorporation of placeholder data, and the establishment of mock services within the backend infrastructure. This approach, particularly concerning integration endpoints, will be implemented temporarily and is intended to be replaced with the designated components upon their availability.

For the development of the DRE Use Case application, a set of technologies and architectural designs has been selected to underpin the platform's capability to deliver precise and predictive analytics for solar and wind energy projections. Central to the system's architectural framework is the implementation of Onion Architecture. This architecture emphasises the inversion of dependencies and a domain-centric design approach, which significantly enhances the maintainability and testability of the system. PostgreSQL, augmented with the TimescaleDB extension, has been adopted for the persistence layer. This combination is specifically chosen for its superior capability in efficiently managing and querying time-series data, which is essential for the high volume and dynamic nature of weather and energy datasets. The server-side logic harnesses the robustness of Microsoft ASP.Net, facilitating the creation of secure, scalable application programming interfaces (APIs). In parallel, Microsoft Blazor is utilised to craft a dynamic, interactive web application interface, offering users access to real-time data and analytical tools. The documentation and exploration of our REST APIs are facilitated through Swagger UI, providing a transparent, interactive documentation interface that eases the developer and system integrator experience. The programming languages selected for the DRE service (C#, TypeScript, JavaScript, Python, and SQL) were chosen for their versatility and widespread use, ensuring a broad base of developer expertise can be leveraged to enhance and expand our platform capabilities. Finally, the user interface and user experience design are anchored by React to ensure the delivery of a responsive, contemporary, and user-friendly interface.

# Libraries and Frameworks

The Tables below present the catalogue of libraries and frameworks that complement the DRE Use Case's technology stack, enhancing the functionality and reusability of software components.

## Persistence Layer

**Table 1. Components of Persistence Layer.**

|  |  |  |
| --- | --- | --- |
| Component | Description | Version |
| PostgreSQL | Open-source relational database | 14.x |
| TimescaleDB | Time-series data extension for PostgreSQL | 2.x |

## Backend/API

**Table 2. Overview of Backend/API Components for the DRE Use Case.**

|  |  |  |
| --- | --- | --- |
| Component | Description | Version |
| Microsoft ASP.Net | Framework for building web apps and services | .NET 6.x / .NET 7.x |
| Swagger UI | Tool for documenting REST APIs | 4.x |
| C# | Programming language for .NET | 10.x / 11.x |
| SQL | Language for managing databases | N/A |
| destinelab | Python package for authentication and programming access to DataLake datasets | 1.1 |
| scipy | Python algorithms for optimization, integration, interpolation, maths, and statistics | 1.15.1 |
| polytope | Python package for the provision of RESTful data access API over HTTPS, allowing access to high-performance data storage over the Internet | 1.2.5 |
| cdo | Tool set in Python interface for working with climate and NWP model data | 2.5 |
| scikit-learn | Python package for machine learning using tools for predictive data analysis | 1.6 |
| |  | | --- | | FastAPI | | Python web framework for building APIs | 0.104.1 |
| Uvicorn | ASGI server for Python web apps | 0.23.2 |
| xarray | Python package for working with labeled multi-dimensional arrays | 2023.10.1 |
| pandas | Python library for data manipulation and analysis | 2.1.2 |
| pvlib | Python library for simulating photovoltaic energy systems | 0.10.1 |
| xgboost | Gradient boosting library | 2.0.0 |
| numpy | Fundamental package for numerical computing in Python | 1.26.0 |

## API Specification for Solar and Wind Assessment

**Table 3. Input specification for premium wind assessment.**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | Notes |
| Latitude | float | Latitude of the wind park | Must be between -90.0 and 90.0 |
| Longitude | float | Longitude of the wind park | Must be between -180.0 and 180.0 |
| start\_date | string | Start date of the data retrieval period (YYYY-MM-DD) | Must be a valid date not earlier than 1940-01-01 and not later than 7 days before today. Must not be later than end\_date |
| end\_date | string | End date of the data retrieval period (YYYY-MM-DD) | Must be a valid date within the same bounds. Must not be earlier than start\_date |
| height | integer | Measurement height above ground level (m) | Must be either 10 or 100 |

**Table 4. Input specification for premium wind assessment.**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | Notes |
| Latitude | float | Latitude of the wind park | Must be between -90.0 and 90.0 |
| Longitude | float | Longitude of the wind park | Must be between -180.0 and 180.0 |
| start\_date | string | Start date of the data retrieval period (YYYY-MM-DD) | Must be a valid date not earlier than 1940-01-01 and not later than 7 days before today. Must not be later than end\_date |
| end\_date | string | End date of the data retrieval period (YYYY-MM-DD) | Must be a valid date within the same bounds. Must not be earlier than start\_date |
| hub height | integer | Hub height of turbines (in meters) | Must be between 40 and 300 meters |
| power\_curve\_model\_upload | CSV file | User-uploaded power curve file | Must follow defined CSV format with  wind\_speed\_in\_m\_per\_s and power\_in\_kW columns |

**Table 5. The output specification for basic wind assessment is also provided in the premium version.**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| time\_utc | string | ISO 8601-formatted timestamp of the measurement |
| Wind\_speed\_ms | float | Wind speed measured in meters per second (m/s) |
| Wind\_direction | float | Wind direction measured in degrees (°) |
| Histogram\_ values | float | Values to create a histogram |
| Weibull\_values | float | Values to create the Weibull distribution |
| Wind\_directional\_statistics | float | Directional statistics for the table |

**Table 6. Output specification for premium wind assessment**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| time\_utc | string | ISO 8601-formatted timestamp of the measurement |
| Wind\_speed\_ms | float | Wind speed measured in meters per second (m/s) |
| Wind\_direction | float | Wind direction measured in degrees (°) |
| Histogram\_ values | float | Values to create a histogram |
| Weibull\_values | float | Values to create the Weibull distribution |
| Wind\_directional\_statistics | float | Directional statistics for the table |
| Wind\_Power\_Production\_kW | float | Wind power production in kilowatts (kW) |
| Mean\_annual\_energy | float | Estimated mean annual energy production per wind speed bin |
| Wind\_power\_analytics | float | Wind power analytics for the table |

## API Specification for Solar and Wind Forecasting

**Table 7. Input specification for solar forecasting.**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | Notes |
| Latitude | float | Latitude of the solar park | Must be between -90.0 and 90.0 |
| Longitude | float | Longitude of the solar park | Must be between -180.0 and 180.0 |
| Εlevation | float | Solar park’s altitude (in meters) | Must be between 0 and 8949 meters |
| Ηistorical solar park power production data file | CSV file | User-uploaded historical solar power production data | Must follow the defined CSV format with time\_utc, total\_production\_active\_power\_kw. |

**Table 8. Output specification for solar forecasting.**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| time\_utc | string | ISO 8601-formatted timestamp of the measurement |
| Solar\_Power\_Production\_kW | float | Solar power production in kilowatts (kW) |
| GHI (Global Horizontal Irradiance) | float | forecasted GHI in kWh/m2, as obtained from the extremes-DT dataset. |

**Table 9. Input specification for wind forecasting.**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | Description | Notes |
| Latitude | float | Latitude of the wind park | Must be between -90.0 and 90.0 |
| Longitude | float | Longitude of the wind park | Must be between -180.0 and 180.0 |
| hub\_height | float | Hub height of turbines (in meters) | Must be between 40 and 300 meters |
| Capacity | float | Installed capacity of the wind park (in kW) | Must be a non-negative number |
| power\_curve\_model\_upload | CSV file | User-uploaded power curve file | Must follow the defined CSV format with  wind\_speed\_in\_m\_per\_s and power\_in\_kW columns |
| historical\_wind\_park\_data\_upload | CSV file | User-uploaded historical wind and power production data | Must follow defined CSV format with time\_utc, power\_in\_kW, wind\_speed\_in\_m\_per\_s (optional: wind\_direction\_in\_deg\_optional, temperature\_in\_C\_optional, etc.) |

**Table 10. Output specification for wind forecasting.**

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| time\_utc | string | ISO 8601-formatted timestamp of the measurement |
| Wind\_Power\_Production\_kW | float | Wind power production in kilowatts (kW) |
| Wind\_speed\_ms | float | Wind speed measured in meters per second (m/s) |
| Wind\_direction | float | Wind direction measured in degrees (°) |

## Frontend/Web Application - UI/UX

**Table 11. UI/UX components.**

|  |  |  |
| --- | --- | --- |
| Component | Description | Version |
| Typescript | Programming language for scalable JavaScript apps | 4.x |
| JavaScript | Programming language for web development | ECMAScript 2022 |
| React 18 | Modern front-end library for building responsive, component-based user interfaces | 18.x |

# Conclusion and Future Directions for Software Reuse

This final version of the deliverable outlines the scheduled components of the DRE release 5 and the onboarding status of the DRE Use Case within the DESP Core services. The DRE consortium has updated the relevant libraries and frameworks for the designated components based on their availability. The content reflects the latest status of the Use Case, consolidated through the most recent iteration.