CALL FOR TENDER

Legal basis

Directive 2014/24/EU

1. Contracting entity

1.1.	**Official name**.	Collaborative Research for Energy System Modelling AISBL
	Short name:	CRESYM
1.2.	**Postal address**.	Square de Meeûs 38-40
	Town.	Brussels
	Postal code:	1000
	Country:	Belgium
	NUTS code :	BE BELGIQUE-BELGIË
1.3.	**Contact person**.	Sébastien LEPY
	E-mail:	contact@cresym.eu
	Telephone:	+33 666 543 654
1.4.	**Internet address**.	https://cresym.eu
1.5.	**Main activity**.	Research & Development

2. Object

Scope of the procurement

2.1.	**Title**.	Improvement of the collaborative shared dynamic simulation library Colib
2.2.	**Reference number**.	003
2.3.	**Type of contract**.	Services

2.4. **Short description**. The Collaborative OpenSource Library of power system component dynamic simulation models & testcase (COLib) has been built for operators and research institutions lacking time and resources to update, complete and maintain necessary dynamic component models and test cases for their studies. This library would benefit to all by its high standards, transparency of model equations, and concrete applications (real test cases).

This library will contain power systems components, networks, and small and large test cases for steady-state and dynamic stability studies. For each of them, the description of the object is presented on one side, and the links to open-source implementations with indicators on the quality on the other side.

A website using GitHub webpages has been developed. Improvements are however necessary to make the current solution more industrial, more flexible, and more pleasant. More details available in Appendix A

2.5. **Lots**. This contract is not divided into lots.

3. Procedure

3.1. **Time limit for receipt of tenders or requests to participate**.

Date:	16/06/2025
Local time:	noon CEST

3.2. **Communication**. Access to the procurement documents is restricted. Further information can be obtained from the abovementioned address.

Tenders or requests to participate must be submitted in English to the abovementioned email address.

3.3. **Date of dispatch of this notice**.

02/06/2025

Conditions for participation

3.4. **Economic and financial standing**.

The candidate must be an organisation, registered in the EU or UK and active for more than 2 years.

3.5. **Technical and professional ability**.

The candidate must display experience and references about similar works in the tender document about:

- web site development preferably with GitHub webpages, jekyll, ruby, quarto development and management, opensource publications.
- good programming skills in various scripting language such as python, bash, etc. is expected.
- Knowledge of modelling languages such as modelica, Julia, etc. is a plus.
- Development, testing and maintenance of computer programs

The candidate may point out other relevant technical ability.

In addition, the candidate should have:

- Solid knowledge of written and spoken English.
- Good interpersonal skills, the ability to thrive in a diverse, multidisciplinary environment.

Award criteria

3.6. **Award criteria**.

Price is not the only award criterion.

All other criteria are:

(i) the candidate's experience and references about similar works that the tender will point out (esp. web site development, GitHub webpages jekyll, html, GitHub workflows knowledge, opensource publications)

(ii) the candidate's staff skills (the tender will provide the relevant resumes);

(iii) completeness of the tender;

(iv) clarity of the tender.

4. Description

4.1. **Place of performance**. NUTS code: BE BELGIQUE-BELGIË

4.2. **Description of the procurement**. Please find the details in appendix A.

4.3. **Duration of the contract, framework agreement or dynamic purchasing system**.

Duration in months: 2 months max for section 1, 2, 3

4.4. **Information about renewal**. This contract is not subject to renewal.

4.5. **Information about variants**. Variants will be accepted.

4.6. **Information about options**. Options will not be accepted.

4.7. **Information about European Union funds**. The procurement is not related to a project and/or programme financed by European Union funds.

4.8. **Information about recurrence**. This is not a recurrent procurement.

4.9. **Additional information**.

None

5. Review procedure

5.1. **Review procedure**. Any dispute related to the suspension of the procurement procedure shall be settled by the President of the Tribunal of Commerce ('Voorzitter van de Rechtbank van Koophandel'/'Président du Tribunal de Commerce'), or, in case of annulment by the Tribunal of Commerce ('Rechtbank van Koophandel'/'Tribunal de Commerce'), each time Boulevard de Waterloo 70-1000 Bruxelles.

Appendix A

The current Colib platform has been developed as a prototype to validate the feasibility and main ideas (model templates, contribution process, etc.).

The current solution utilises GitHub Pages, a service provided by Github, to host a public website on the 'github.io' domain. The site is generated using Jekyll, a static site generator.

The website is automatically built and published by GitHub pages from the main branch, simplifying the deployment process. The pages are written in Markdown that are converted by Jekyll into html. This simplifies the contribution of electrical engineers that can write a model or test case description by simply using a markdown template. Using GitHub also facilitates the contributions since the use of GitHub to store code projects is widespread in this field. More generally, the ease with which researchers and engineers can contribute to the library is a key quality criterion of the platform.

Navigation is simplified though the search bars, tags filters, tables of content and scroll down menus. All these elements are built to automatically adapt to future contributions.

However, improvements are necessary to make the current solution more industrial, more flexible, and more pleasant.

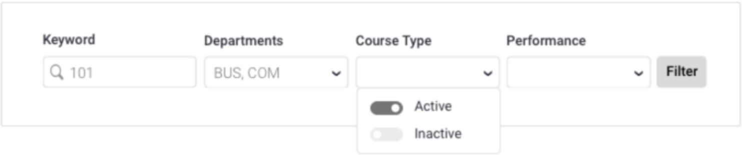
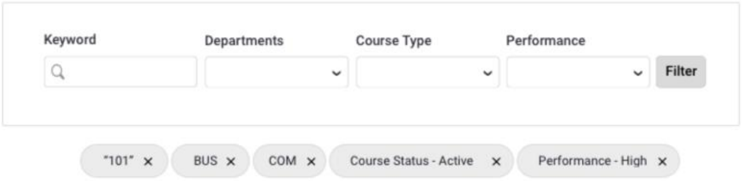
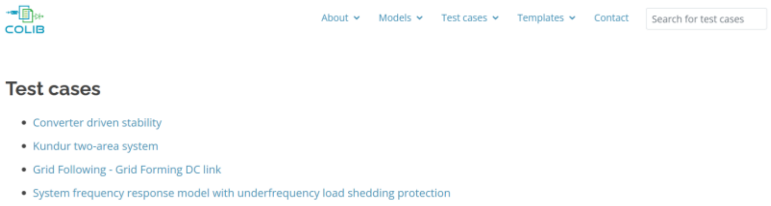
There are four sections of development.

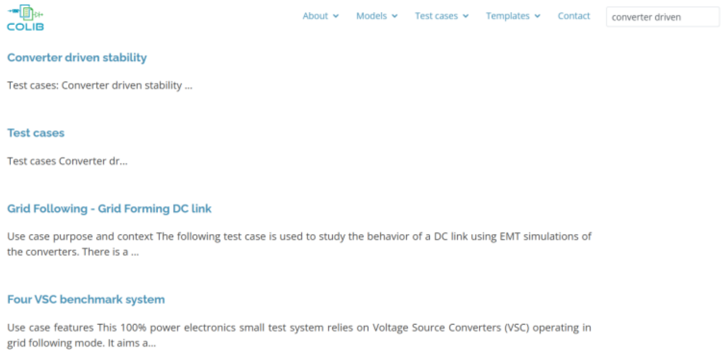
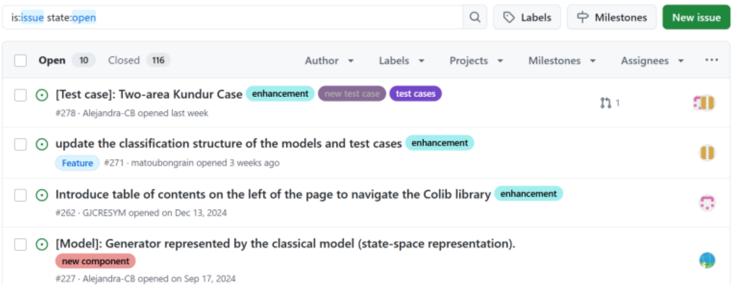
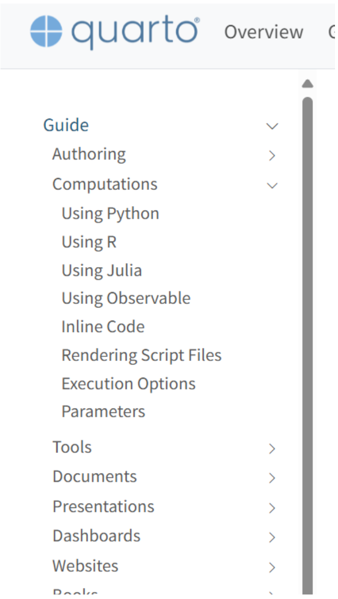
Section 1: Priority#1- Expected delivery by July/2025

Development		
Task #	**Title**	**Description**
1	Integration of Quarto	Quarto is an open-source scientific and technical publishing system. It is very powerful to display static and dynamic content. It can execute Python, R, Julia, Observable as part of Markdown pages and publish the content on various media (articles, presentations, dashboards, websites, blogs, and books in HTML, PDF, MS Word). For Colib, we are particularly interested in quarto to write using Pandoc markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more. Quarto websites and documents can be published through GitHub pages as explained here: GitHub Pages – Quarto. For colib, the pages would become qmd pages instead of simple md. The rest of the website should remain ideally the same. Dependencies to Jekyll and ruby should be clarified.

		If the styling is not compatible with the current version after using pandoc, then need to explore Hugo – Quarto (Static site generator).
2	Migration tasks	1. Convert .md files to .qmd files 2. Migrate _config.yml to _quarto.yml 3. Replace Jekyll includes with Quarto includes 4. Port assets/css 5. Remove liquid tags and replace with pandoc/Quarto Syntax 6. Rebuild navigation, sidebar, and layout using Quarto's format 7. Test GitHub pages build via Quarto's GitHub Actions

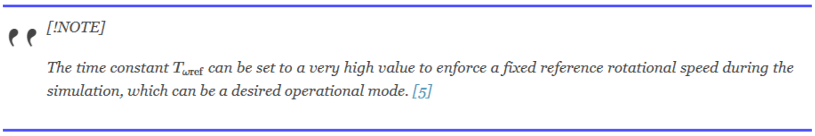
Section 2: Priority#2- Expected delivery by July/2025

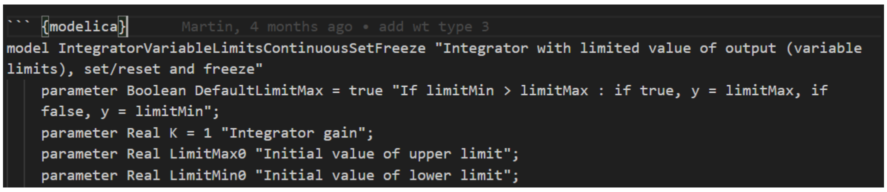
LIST OF ENHANCEMENTS		
Enhan-cement #	**Title**	**Description**
1	Filter bar	For now, there is no filtering possible. We would like to create several filters on the model and test case main pages. For each filter category the displayed list would be based on a predefined list of tags. Those tags would come from the metadata block of the pages.  Keyword 101 Departments BUS, COM Course Type Active Inactive Performance Filter  Keyword Departments Course Type Performance Filter "101" BUS COM Course Status - Active Performance - High The filter would be combined with the search function results' list. The displayed results would be the intersection of the sets of results. The number of results should appear at the beginning of the results'list.
2	Add a tag search module and filtering option	For now, the search function is rather simple, as it consists of a search bar for test cases or models with a searching function scanning the text of the pages and metadata block.  COLIB About Models Test cases Templates Contact Search for test cases Test cases • Converter driven stability • Kundur two-area system • Grid Following - Grid Forming DC link • System frequency response model with underfrequency load shedding protection We would like to enhance the search function so that it is less case-sensitive, it returns only model or test cases pages and not the folder. It combines nicely with the filter bar.

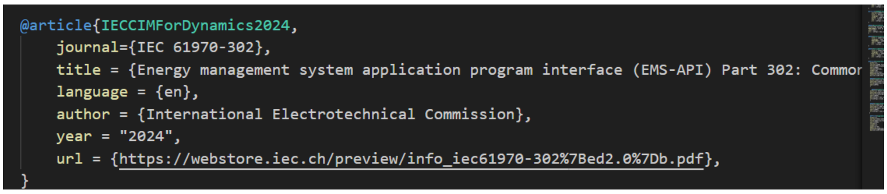
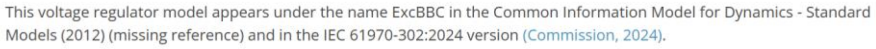
3	Improve the search results list	For now, the list of results is displayed as follows:  COLIB About Models Test cases Templates Contact converter driven Converter driven stability Test cases: Converter driven stability ... Test cases Test cases Converter dr... Grid Following - Grid Forming DC link Use case purpose and context The following test case is used to study the behavior of a DC link using EMT simulations of the converters. There is a ... Four VSC benchmark system Use case features This 100% power electronics small test system relies on Voltage Source Converters (VSC) operating in grid following mode. It aims a... Several improvements could be proposed: - The number of results should appear at the beginning of the results'list. - The tags of the result's pages should appear next to the page title. See GitHub as example:  is:issue state:open Labels Milestones New issue Open 10 Closed 116 Author Labels Projects Milestones Assignees [Test case]: Two-area Kundur Case enhancement new test case test cases #278 · Alejandra-CB opened last week update the classification structure of the models and test cases enhancement Feature #271 · matoubongrain opened 3 weeks ago Introduce table of contents on the left of the page to navigate the Colib library enhancement #262 · GJCRESYM opened on Dec 13, 2024 [Model]: Generator represented by the classical model (state-space representation). new component #227 · Alejandra-CB opened on Sep 17, 2024 - The searched words are highlighted in the text.
4	Navigation side menu and automatic counting function	Make the structure of the models and the test system automatically appear on the left side of each page. Possibility to click on the links of the structure to change pages. Automatic count of the models and submodels that the folder contains should appear. For example:  quarto Overview Guide Authoring Computations Using Python Using R Using Julia Using Observable Inline Code Rendering Script Files Execution Options Parameters Tools Documents Presentations Dashboards Websites Books

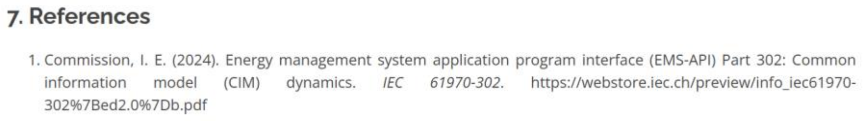
5	Collapsible sections	For the model and test system pages, create some collapsible sections for text. This is easy to do if quarto is integrated as part of our website.
6	Collaborator contribution process	Implement a feature on the website that allows contributors to create a page using an in-browser emulator. Upon, submission, the content should be automatically exported as a .qmd file (based on the model and test case template) and a new GitHub issue should be created containing the exported file. The entire process must be automated so that contributors only interact with the emulator, without needing to handle file downloads or manually submit issues.

Section 3: Priority#3- Expected delivery by July/2025 (This section is not applicable, if section 1 and 2 are successfully implemented)

LIST OF BUGS		
Bug #	**title**	**Description**
1	Retro compatibility of ruby with bibtex.	Upgrade of ruby library can't be made as compatibility with bibtex is no longer maintained. Workaround to solve this issue should be found.
2	Quote template is not good	Quote template doesn't appear nicely; we should change the css file.  [!NOTE] The time constant $T_{\omega ref}$ can be set to a very high value to enforce a fixed reference rotational speed during the simulation, which can be a desired operational mode. [5] New quote format should be proposed.

3	Equation section doesn't appear correctly	Current equation section doesn't appear nicely on the website. Improvement should be proposed and possibility to code in markdown with an easy keyword such as below should be proposed.
		
		```
		 {modelica}     Martin, 4 months ago • add wt type 3
		model IntegratorVariableLimitsContinuousSetFreeze "Integrator with limited value of output (variable limits), set/reset and freeze"
		    parameter Boolean DefaultLimitMax = true "If limitMin > limitMax : if true, y = limitMax, if false, y = limitMin";
		    parameter Real K = 1 "Integrator gain";
		    parameter Real LimitMax0 "Initial value of upper limit";
		    parameter Real LimitMin0 "Initial value of lower limit";
		Embedding quarto as part of the website (for more details see the enhancement section) could be one way to solve this problem.
		```

4	References don't appear correctly	The way the reference appears differs from the bibtex file:  @article{IECCIMForDynamics2024, journal={IEC 61970-302}, title = {Energy management system application program interface (EMS-API) Part 302: Commor language = {en}, author = {International Electrotechnical Commission}, year = "2024", url = {https://webstore.iec.ch/preview/info_iec61970-302%7Bed2.0%7Db.pdf}, } Bibtex files article section How it appears on the webpage:  This voltage regulator model appears under the name ExcBBC in the Common Information Model for Dynamics - Standard Models (2012) (missing reference) and in the IEC 61970-302:2024 version (Commission, 2024).

		 7. References 1. Commission, I. E. (2024). Energy management system application program interface (EMS-API) Part 302: Common information model (CIM) dynamics. *IEC 61970-302.* https://webstore.iec.ch/preview/info_iec61970-302%7Bed2.0%7Db.pdf

Any additional enhancement of the website is welcome.

Section 4: Priority#4- Expected delivery by Oct/2025

LIST OF FEW COMPLEX PROOF-OF-CONCEPT FEATURES		
POC #	**Title**	**Description**
1	Launch small test case with modelica code	For a given model A in modelica, create a run function that would do the following: Get the modelica model A from the GitHub repository Get the SMIB test case in modelica Plug the A model connectors to the SMIB connectors Open OMEDIT or launch it on the webpage. Launch the test case
2	Launch some modelica code in the page	Write some modelica code Use the validation function of the OpenModelica compiler
3	Display OpenModelica results	Being able to display plots containing legends, title and zoom function, based on the results of the modelica code.

End of the document