



# Introduction to Freva – A Free Evaluation System Framework for Earth System Modeling

SOFTWARE METAPAPER

CHRISTOPHER KADOW 

SEBASTIAN ILLING

ETOR E. LUCIO-ECEIZA 

MARTIN BERGEMANN 

MAHESH RAMADOSS 

PHILIPP S. SOMMER 

OLIVER KUNST

THOMAS SCHATNER

KLAUS PANKATZ 

JENS GRIEGER 

MAREIKE SCHUSTER

ANDY RICHLING 

HANNES THIEMANN 

INGO KIRCHNER 

HENNING W. RUST 

THOMAS LUDWIG

ULRICH CUBASCH 

UWE ULBRICH 

*\*Author affiliations can be found in the back matter of this article*

][ubiquity press

CORRESPONDING AUTHOR:

**Christopher Kadow**

Deutsches  
Klimarechenzentrum  
(DKRZ), Germany; Freie  
Universität Berlin, Institute of  
Meteorology, Germany

[kadow@dkrz.de](mailto:kadow@dkrz.de)

## ABSTRACT

Freva – Free Evaluation System Framework for Earth system modeling is an efficient solution to handle evaluation systems of research projects, institutes or universities in the climate community. It is a scientific software framework for high performance computing that provides all its available features both in a shell and web environment. The main system design is equipped with the programming interface, history of evaluations, and a standardized model database. Plugin – a generic application programming interface allows scientific developers to connect their analysis tools with the evaluation system independently of the programming language. History – the configuration sub-system stores every analysis performed with the evaluation system in a database. Databrowser – an implemented meta data system with its advanced but easy-to-handle search tool supports scientists and their plugins to retrieve the required information of the database. The combination of these three core components, increases the scientific outcome and enables transparency and reproducibility for research groups using Freva as their framework for evaluation of Earth system models.

KEYWORDS:

Evaluation; Verification;  
Validation; High Performance  
Computer; HPC Hub/Portal,  
Science Gateway; Software  
Development; Data Sharing;  
Shared Knowledge Approach;  
Earth System Modeling; Climate  
Modeling; CMIP5; CMIP6

TO CITE THIS ARTICLE:

Kadow C, Illing S, Lucio-Eceiza  
EE, Bergemann M, Ramadoss M,  
Sommer PS, Kunst O, Schartner  
T, Pankatz K, Grieger J, Schuster  
M, Richling A, Thiemann H,  
Kirchner I, Rust HW, Ludwig  
T, Cubasch U, Ulbrich U 2021  
Introduction to Freva – A Free  
Evaluation System Framework  
for Earth System Modeling.  
*Journal of Open Research  
Software*, 9: 13. DOI: [https://doi.  
org/10.5334/jors.253](https://doi.org/10.5334/jors.253)

## (1) OVERVIEW

### 1. INTRODUCTION

#### State of the Art Challenge in Earth System Modeling

The Earth system modeling community nowadays uses information technology, data, and software as an indispensable support for science. Scientists use climate models as their main tools to simulate and research the past, present, and future climate. The Intergovernmental Panel on Climate Change (IPCC)<sup>1</sup> urges that *'it is crucial therefore to evaluate the performance of these models'*. A growing variety of research software and the increase in computer power allows scientists to study a steadily increasing amount of data. The ongoing production of data and model development stages need to be evaluated in a sustainable way. Therefore, scientists develop evaluation and verification software with the code of best practice in mind. However, usually scientists are not software engineers. Scientists have to invest a lot of time in their software development skills. More than often scientists develop software routines about topics, which were developed already many times by other scientists. This leads to a huge amount of partly redundant results and software development history. It is difficult to accomplish reproducible, transparent, and efficient scientific results. Thus, there is a demand of software and community frameworks supporting scientists to overcome technical hurdles and concentrate on climate research.

The general concept of a science gateway is common in many research disciplines with a need for special IT resources [1]. Science gateways allow science and engineering communities to access shared data, software, computing resources, instruments, educational materials, and other resources specific to their disciplines [2]. A science gateway combines several technologies around software and databases to create one web portal access point to compute resources on Grid, HPC, or Cloud networks. The range of disciplines which developed specialized science gateways covers several specialized research fields like life sciences [3], nanotechnology [4], biology [5], etc.

There is a growing need for common scientific infrastructures in the Earth system modeling community, too. However, the attempt to migrate to one common software in a research project can be challenging in practice. Several climate research groups developed and provided their own software packages during the last decades (e.g. CDO,<sup>2</sup> PCMDI metrics package,<sup>3</sup> Global Marine,<sup>4</sup> RCMES tool,<sup>5</sup> ESMVal<sup>6</sup>). In the majority of cases, these packages focus on one research topic without aiming to be open for a broader audience. Usually, these software packages are provided as scripts which need to be adapted in the programming language they are written in. While this way of providing tools is very flexible because it is possible to adapt the tool completely to one's own project needs, these scripting formats lack usability.

In order to improve usability, a few research centers in the recent years have developed websites which present some pre-calculated research galleries (e.g. the Decadal Predictability Working Group<sup>7</sup>). Even less research centers also provide an actual science gateway, which includes a dynamic calculation of results depending on the chosen options (e.g. Climate Explorer,<sup>8</sup> BirdHouse,<sup>9</sup> Climate Data Store<sup>10</sup>). Often these sites do not offer the possibility to adapt the tool or to use own software and datasets. With this restriction of flexibility, but interactive production of graphics by the users, it is at least possible for users to produce pre-defined evaluations. Climate science often needs new or re-developments on software packages build for specific tasks. Furthermore, usually these platforms provide no opportunities to build specific portals needed by research groups that prefer a self-contained environment on the research group's local computing infrastructure.

With the growing amount of research data in climate science there is also a risk of losing track of research possibilities. Several model intercomparison projects (MIPs) were started in recent past to make climate modeling activities comparable. This was only achievable by using common international data standards and granting international data availability through the Earth System Grid Federation (ESGF).<sup>11</sup> These projects facilitated data standardization, validation, model comparisons, and multi-model assessment. The ESGF database is a huge collection of Earth system modeling data. However, scientists still need to find ways of detecting and incorporating these amount of data in their science. There is also the need to incorporate other sets of observations, reanalysis, or model data, because research gets turnarounds during evaluation. Flexibility and efficiency are therefore important in data relevant research.

Consequently, three core issues are addressed with this study. In climate science, there is a general need for...

1. ... flexible and individual but efficient research environments on HPCs.
2. ... traceable research and the opportunity for a reproduction of results.
3. ... targeted data access on huge climate data bases.

#### Origin

The Free Evaluation System Framework (Freva)<sup>12</sup> has been developed for climate modeling research of decadal climate prediction within the 'Mittelfristige Klimaprognosen' (MiKlip) major project funded by the Federal Ministry of Education and Research in Germany (BMBF). Within MiKlip, the Freva framework hosts the MiKlip Central Evaluation System (CES) [6] on a high performance computer (HPC) at the German Climate Computing Centre (DKRZ).<sup>13</sup>

## Exemplary Research Group

Marotzke et al. (2016) [6] state: ‘*The MiKlip hub furthermore provides a central evaluation system. The evaluation system, the necessary observational data, and the entire set of MiKlip prediction results conform to the CMIP5 data standards (Taylor et al. 2012) and reside on a dedicated data server. The MiKlip server makes the prediction results and evaluation system immediately accessible to the entire MiKlip community, thereby providing a crucial interface between production on the one hand and research and evaluation on the other hand. [...] The central evaluation system is constantly expanded with contributions from the MiKlip evaluation module and, together with its reference data pool for verification, resides on the same data server as the entire MiKlip prediction output. The analyses are collected into a database ensuring reproducibility and transparency. Providing the central evaluation system to the entire MiKlip project is also an effective training tool, especially for those researchers who have only recently joined the rapidly expanding field of decadal prediction.*’

## Target Group

Freva is a research software environment and science gateway, hosting verification routines and observational, reanalysis, and model data in customized central evaluation systems of research groups like described in the MiKlip project. The potential user of Freva can be an institute, university, research center, project (like MiKlip), or simply an individual scientist. To address potential user classes with one term, we call it research group hereafter. Freva gives full control of the scientific tool development and improves science through efficient tool application, distinct data access, and integration into a central system. This combination requires a fluent interplay and user guides – which will be in addition to this paper. Freva as a framework is designed for three different user groups who will be addressed in this study and in their individual user guides. First, there are the *users* of the research group’s evaluation system that look for help in the basic user guide (BUG). In the second group there are *plugin developers* who fill Freva with scientific applications and retrieve documentation by using the basic developer guide (BDG). Of course, the *developers* are *users* as well. Last but not least, the *admins* of the research group host the Freva instance as a scientific infrastructure for *users* and *developers*. The *admins* may resort to the basic admin guide (BAG). All three groups are scientists in the field of Earth system modeling.

## Research Agenda

In this study, we present the system design of Freva, its main features, and its combination of different software technologies (*Figure 1*). Freva is a combination of a well-defined software plugin management, Earth system model data retrieval, and a backup of all analyses within a portal including a web and a shell frontend on a high

performance computer (HPC). The system offers a balance between usability and flexibility but being presupposed by transparency and reproducibility (Sect. 2). The main use and features of Freva offer a single program solution (Sect. 3). We then discuss the advantages of a hybrid evaluation system making use of big data HPCs in climate science and Earth system modeling (Sect. 4). As a picture is worth a thousand words, hands on a software is way more intuitive to understand, than reading about it in a paper. Readers are invited to go to [freva.met.fu-berlin.de](http://freva.met.fu-berlin.de), click on ‘Guest?’, login, and compare the following sections with the live evaluation system while getting inside views.

## 2. FRAMEWORK SYSTEM DESIGN – IMPLEMENTATION AND ARCHITECTURE

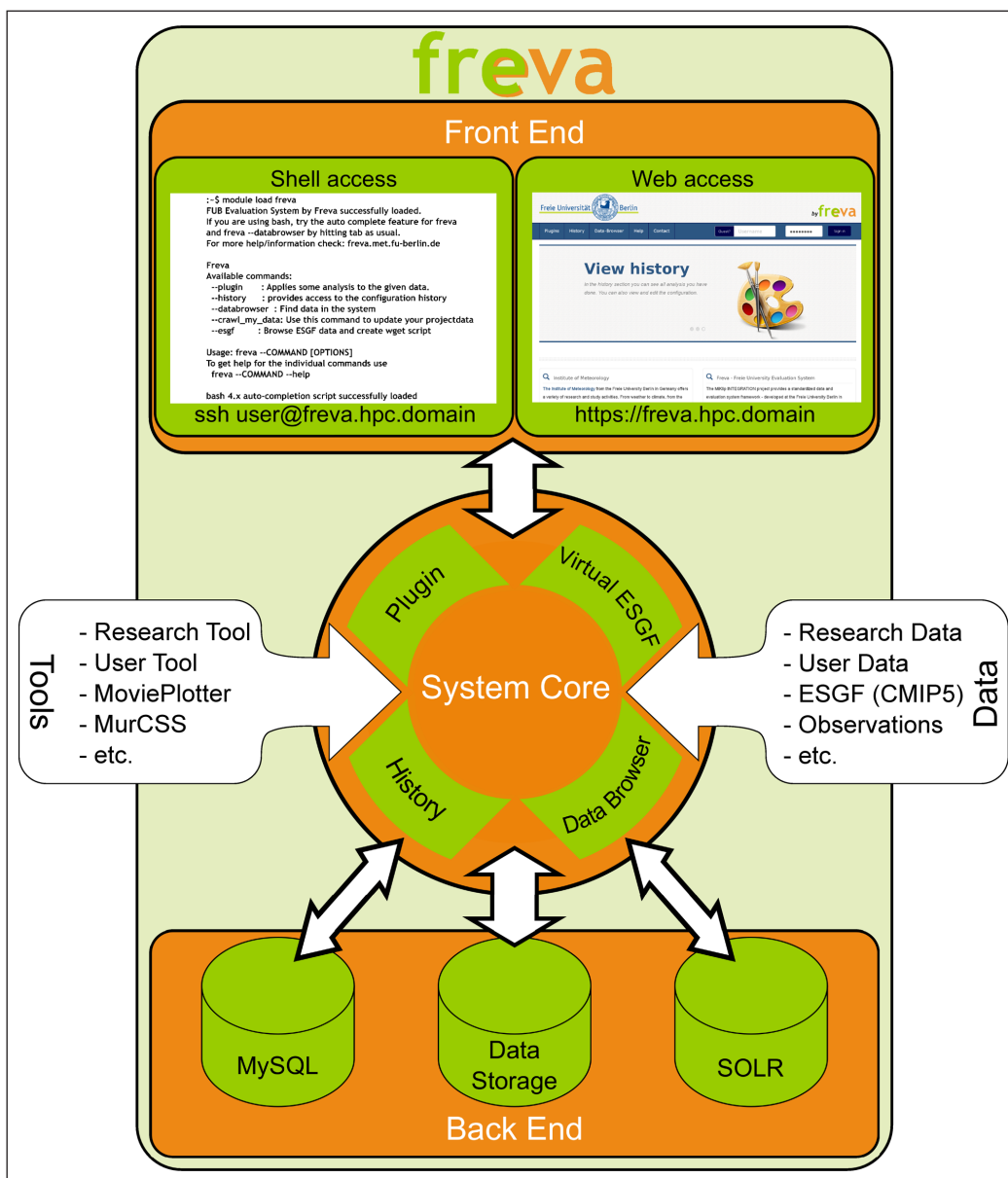
Freva is an evaluation system framework for scientific validation data and software, and it runs as a hybrid system in the web and shell (*Figure 1*). In this section the concept is explained addressing the general purpose of the system. Freva’s integrated frontends fulfil an optimum usage and well-defined interaction between the users and the evaluation system (Sect. 2.1). The System Core of Freva consists of software components, the wrapping of the plugin interface, the history database, the model data browser, and the virtual ESGF library (Sect. 2.2). The combination of different open source technologies into the main framework allows the evaluation system to be generated by one software solution (Sect. 2.3).

### 2.1 Frontends of Freva – Usability and Flexibility

The frontends of Freva (Sect. 2.1.1 and 2.1.2) give users and plugin developers access to the resources of the System Core and the backend databases (Sect. 2.2). Both web and shell frontends connect the scientists with the application system as they represent the interface of the core commands *plugin* (Sect. 2.2.1), *history* (Sect. 2.2.2), and *databrowser* (Sect. 2.2.3). Both interfaces connect the scientists with the application system. The scientists can decide, which degree of freedom they like in using the shell and web by starting, adjusting, and operationalizing evaluation procedures as described in the following.

#### 2.1.1 Shell Interface

The shell interface is the most useful one when accessing an HPC environment in climate science. The command-line approach allows the development of adjustable Unix-based routines. It grants fast and flexible data access using efficient climate data processing tools. The opportunity to write code within the software applications running within Freva for example with regular expression and basic bash commands improves software and data handling. In that way Freva can for example be started and monitored regularly by Cron jobs. Even big evaluation routines by Freva can be started within Bash loops. In the



**Figure 1** Freva – The Free Evaluation System Framework and its design combining several technologies into one common software solution. The System Core contains the plugin API handling tools, the history saving configurations and the data-browser where to find data. The scheme represents the basic structure of this study including its subsections.

following list, we explain the three main features (see Sect. 2.2 for details) of the shell interface applying Freva’s core-commands:

The **--plugin** (Figure 2) section holds all plugged-in tools and helps the user to start one. When the user forgets a mandatory option of a *--plugin*, Freva gives the name of the missing option. When the user mistypes an option of a plugin, Freva suggests the right one (see also Figure 6).

The **--history** (Figure 3) command gives direct access to all analysis and their result directories. Distinct IDs are utilized to sort all results and show their respective history entries. Furthermore, the history holds all configurations and starting commands, which are editable and restartable.

The **--databrowser** (Figure 4) interface efficiently searches the model database. The integrated bash-completion automatically fills the data browser search facets by simply tabbing, thus leading the user easily to the needed dataset or given overview of the database.

Beside these main options, there are assisting side commands only available in the shell:

The **--help** always gives detailed information about Freva, its subcommands, and plugins. The **--esgf** helps users to download data from the ESGF, establishes a connection to the ESGF and generates the necessary WGET script using the standardized attributes and facets.

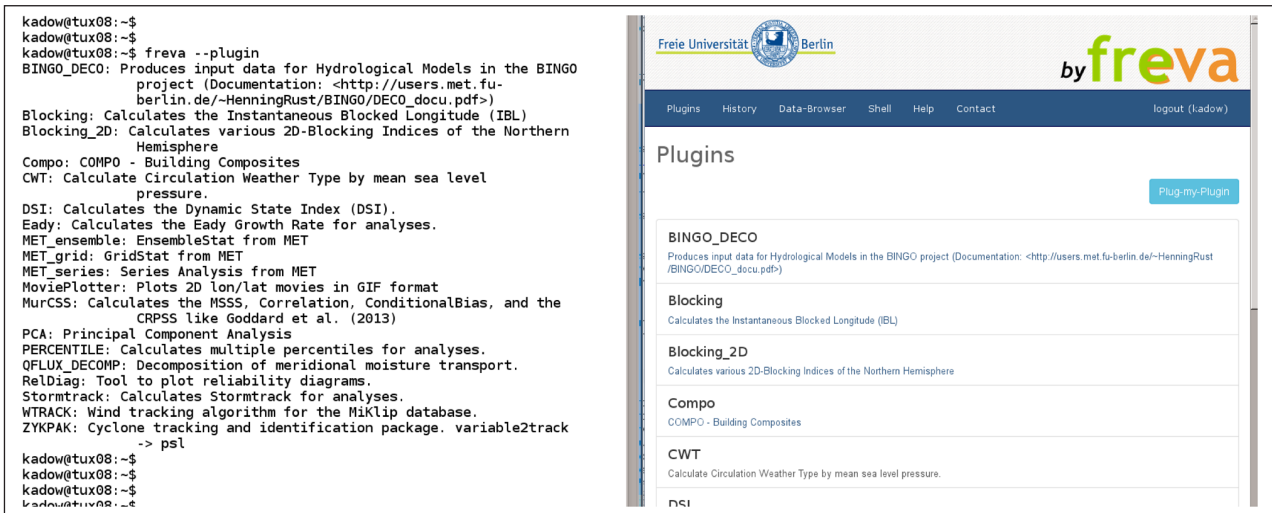


Figure 2 The plugin list in the shell and web interface (snapshot).

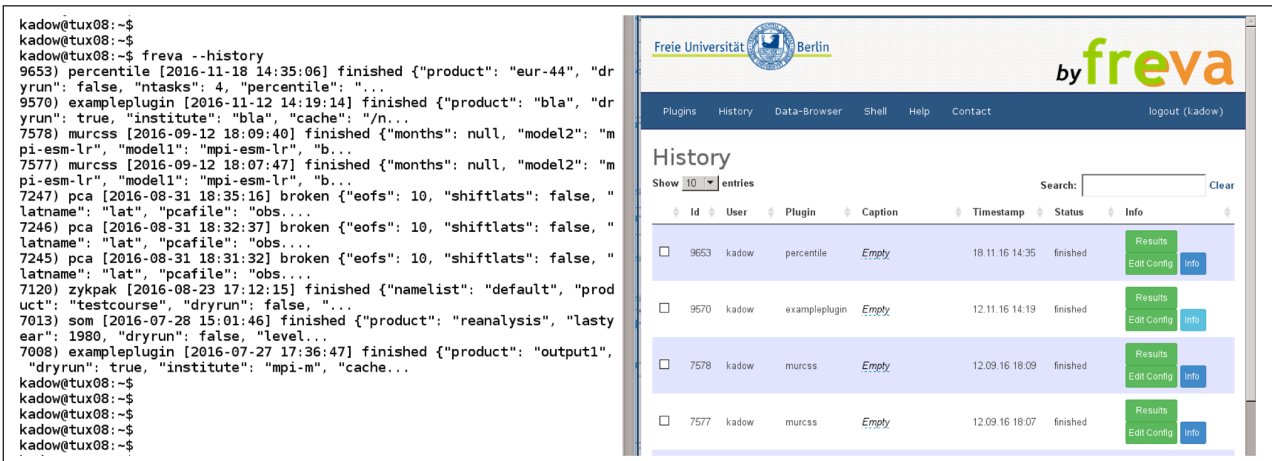


Figure 3 The history in the shell and web interface (snapshot).

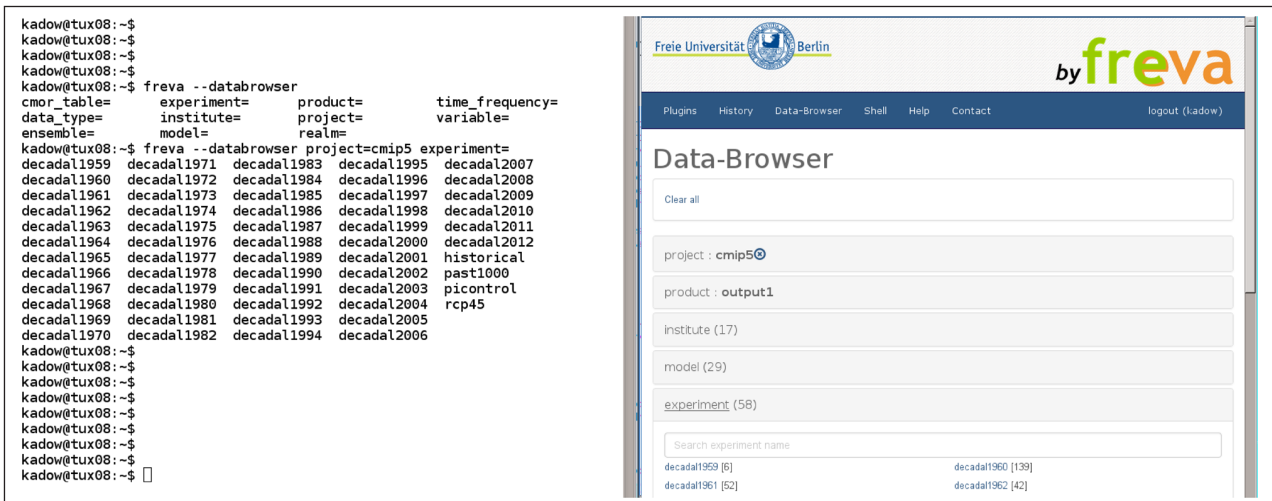


Figure 4 The databrowser in the shell and web interface (snapshot).

The `--crawl_my_data` subcommand offers the opportunity to implement additional standardized datasets. Users can compare their data sets against the ones of the research group, the ESGF projects, or data from other users.

### 2.1.2 Web Interface

The web interface works similar to the shell interface (Sect. 2.1.1). However, it advances Freva's usability. Usually on HPC environments there is no comfortable way to find or process data and even view results. The web



interface introduces easy entrance points for beginners and experts. The three main features (see Sect. 2.2 for details) stay the same: plugin, history and databrowser. In the following the advantages of these three features in the web interface are explained.

The **Plugin** section ([Figure 2](#)) gives access to plugins, an overview of their options, and assists the user during the individual starting procedure with pre-filled facets. When a user forgets to set a mandatory option, the web interface points to the missing plugin option. There are two ways of accessing the HPCs database. It is possible to point to a specific file by browsing the user's main directories of the user or project, or even use the databrowser to search for some file to analyze. Plugins can apply the more advanced Climate Model Output Rewriter (CMOR) syntax options to search the whole database of the research project or a virtual ESGF project – option by option (e.g. project, experiment, variable, etc.). This built-in databrowser search is increasing efficiency by decreasing the number of CMOR facets with every selection made by only showing remaining possible combinations.

The **History** section ([Figure 3](#)) shows the completed, scheduled, or running evaluations. All configurations, including the GIT [7] versioning information, can be retrieved. It is also possible to restart a finished evaluation (*Edit Configuration*). To organize their results, the user is allowed to set a *caption* or delete them from the history section. The *Search* bar allows to search within the configurations started with Freva and filter for used options and e.g. CMOR options.

The **Data-Browser** section ([Figure 4](#)) gives a convenient way of finding data in the database of the research group. By just clicking through the given standardized (DRS, CMOR, CORDEX, ANA4MIPS, etc. – see ESGF)<sup>14</sup> facets, the user finds data sets and data directories. The web frontend provides even more meta information of the search facets, like *variable*, *model*, or *institute*, to explain the meaning of the abbreviations and help to find the right data sets or see what is available. Furthermore, the web part allows to stream the meta data of a specific file by starting *ncdump* from the NetCDF package.

Besides the main options, there are some extras in the web:

The **Help** section hosts information about the evaluation system build with Freva. A web-tour explains the usage of the web page. The scientists find documentation of the research project and developed plugins. Guidelines are also available in the Help section.

The **Shell** section within the web interface also allows the command-line access to the high performance computer of the research group. Applying the shell-in-a-box enables the users to directly start Freva from the bash through the web.

## 2.2 System Core including Backend

The System Core is the main part of every evaluation system build with Freva ([Figure 1](#)). It is an efficient combination of the following technologies and their communication before, during, and after the analysis of the evaluation system. Its plugin interface manages the incorporation of software tools and their common application in the frontend (See *Plugin – Application Programming Interface* Sect. 2.2.1). All configurations and information of the executed plugins and analyzed data sets are saved to satisfy the commitment to transparency and reproducibility (see *History – Transparency and Reproducibility* Sect. 2.2.2). In order to keep track and to overview the database, Freva can implement standardized interfaces to model, reanalysis, and observational data sets or even data incorporated by the users (see *Databrowser – Standardized Model Data Access* Sect. 2.2.3). Furthermore, Freva is able to create a virtual ESGF project (e.g. CMIP5) in the databrowser. This data is only downloaded when a plugin explicitly requests it. This implementation is an advantage, because it provides access to millions of data sets without the need for a huge data storage (see *Virtual ESGF – Evaluation Data Extension* Sect. 2.2.4).

### 2.2.1 Plugin – Application Programming Interface

The expertise of scientific evaluation in Earth system modeling usually resides with experts of the field. These experts also take care of the translation of their research field into scientific software. Not every scientist is also an expert in software development. Freva serves as a development interface to assist scientists to fulfill the code of best practice in terms of developing scientific software. The next paragraph will give some insight in the technical details.

The plugin framework of Freva handles the connectivity of stand-alone tools to the evaluation system of the research group through an application programming interface (API). The plugin API, written in Python, is well structured to assist tool developers during the process of plugging-in a tool. Every tool gets an *api.py* wrapper to realize the exchange of options between the Freva system and the plugin. The API transmits all necessary options to Freva and to the tool. The following minimum code requirements guide the plugin developer to structure the tool by providing meta information of the plugin.

A simple implementation of a plugin is shown in [Figure 5](#) with an example of the MoviePlotter plugin. The class is derived from the *PluginAbstract* base class and implements some mandatory meta information like *tool\_developer*, *short\_description*, *long\_description*, and the plugin *version*. The *parameters* section automatically collects the tool options by *name* and the corresponding *default*, *mandatory* and *help* information by *ParameterType* and defines the plugin interface to

```

import os, sys
from evaluation_system.api import plugin, parameters
from evaluation_system.misc import config

class MoviePlotter(plugin.PluginAbstract):
    tool_developer = {'name': 'Christopher Kadow', 'email': 'christopher.kadow@met.fu-berlin.de'}
    __short_description__ = "Plots 2D lon/lat movies in GIF format"
    __version__ = (1,0,0)
    __parameters__ = parameters.ParameterDictionary(
        parameters.File(name='input', max_items=9, item_separator=',', \
            mandatory=True, help='NetCDF file(s) to be plotted'),
        parameters.Directory(name='outputdir', default='$USER_OUTPUT_DIR', \
            mandatory=True, help='The default output directory'),
        # SHORTENED OPTIONS

    def runTool(self, config_dict=None):
        input = config_dict['input']
        outputdir = config_dict['outputdir']

        result= self.call('bash %s/movie_plotter.sh %s %s' % (self.getClassBaseDir(),input,outputdir))
        print result[0]
        return self.prepareOutput(config_dict['outputdir'])

```

**Figure 5** The basic plugin.py as example of the MoviePlotter plugin, with condensed option list for display reasons.

the user. The arguments get parsed from the plugin, retrieving not only the options set by the user but also the default values if parameters are unset. The plugin transforms the incoming strings into Freva options, and the parameter classes validate them by type (e.g. string, integer, bool). Next to these ordinary *string*, *integer* or *bool* fields, the *data-browser fields* in the plugin API communicates with Freva's Solr server (see Sec. 2.2.3) and can be interpreted by the web interface. The plugin API offers some system variables set up by the admin in the configuration of Freva. System variables are for example the default user output directory, plot directory, or cache directory, which can be used for a clear organization of the plugin results.

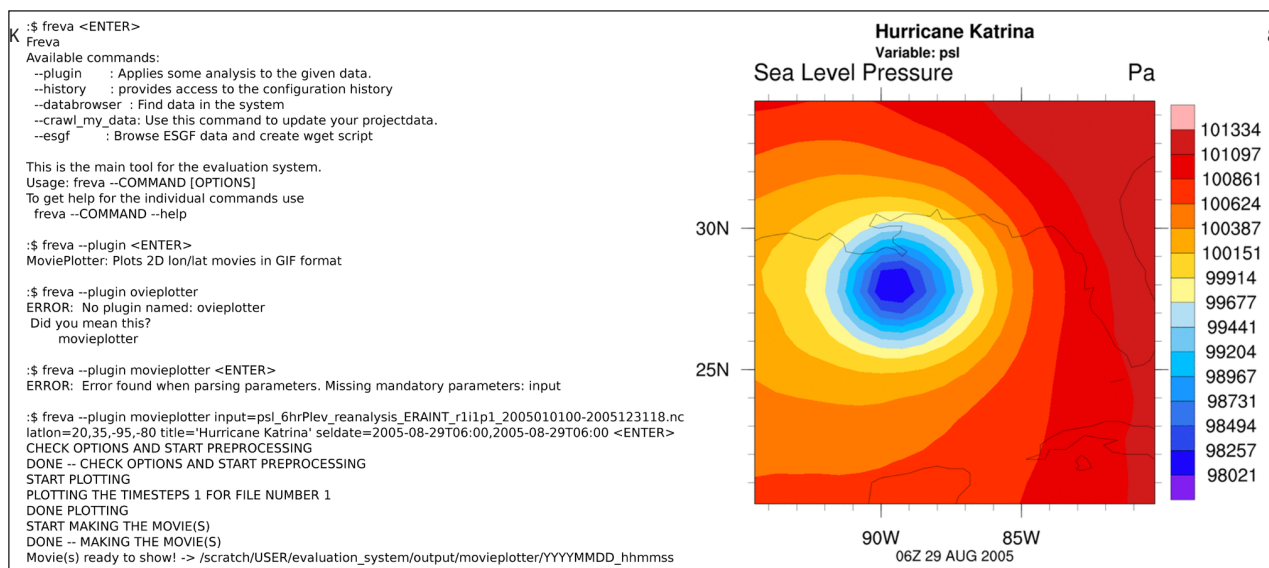
Software developments need flexibility without interferences between the groups – users want to use plugins; developers want to design or re-design plugins. The publicly available plugins are defined in the main configuration file of Freva, and the actual loading is handled by the *PluginManager*. The *PluginManager* controls the upload into the evaluation system and gives access to the plugins as a central registration. Freva offers developers the possibility to connect their new plugins or temporarily redirect the link to the plugin, used by Freva, to their own version – independently of the main systems plugins. The overwritten plugin is only applicable by the developer. The system tells the user which version, i.e. the one from the main system or their own linked version is used when the plugin is started. This is especially useful during development stages because developers can test new features or completely new software without disturbing the production system. The *PluginManager* is parsing the incoming command and generates a configuration as *configDict* each time a plugin is started. The *PluginManager* is able to start the plugged-in tool using the *runTool* interactively in shell or via the available *batch mode*.

### 2.2.2 History – Transparency and Reproducibility

Transparency and reproducibility are important qualities in science. For a scientist it is significant work to take care of the traceability of his research. In that sense, Freva also serves as research recording clerk. The scientific development stages are recorded, easily reviewable, and restartable. The next paragraph will give some insight in the technical details.

All information about performed analyses with Freva is saved in a MySQL database. When a plugin is started, the System Core sets certain information through the *PluginManager*. Each evaluation receives a unique identification number (ID), which is then combined with the user's ID, the plugin name, a time stamp, and status. The configuration parameters of the plugin, including possible data retrieval options (e.g., *Solr fields*), are stored in MySQL. Furthermore, Freva is saving all GIT versioning information, including repository directory and internal version number of the plugin and the Freva version itself, for each analysis. Thus, Freva is flexible enough to guarantee a full recovery of the whole system or just one particular plugin, whenever it may be necessary to reproduce old evaluations. In most cases, it is not necessary to set back the system or plugin. Usually it is enough to browse the history of the respective experiment, retrieve the plugin command via shell or web, and rerun the plugin possibly after slight modification, e.g. outputdir, time ranges, etc. To provide a better overview to the user and help them find old configurations and results, they have the opportunity to entitle each analysis with a *caption*. The history also contains the plugin's interactive standard output. The history class of the System Core establishes several statuses, permissions, and result types of each analysis, which can be retrieved by the frontends (Sect. 2.1).

The history-database in MySQL gets monitored for all evaluations done by Freva. Admins of the research group evaluation system have the possibility to view these. Freva saves the status of the started plugins, for example, *finished*



**Figure 6** The basic usage of Freva in the shell environment including help, listing of plugins, user mistyping and missing information, and Freva suggestions to guide the user to the final result (on the right). Applying a plugin named MoviePlotter for a quick view at the sea level pressure in Pascal of ERA-Interim reanalysis around New Orleans (USA) while hurricane Katrina was hitting its coast.

or *broken*. This is an advantage over stand-alone tools and decentralized usage, because this monitoring helps to reveal data discrepancies and software bugs, as users are not always reporting problems. Freva helps to inform so that users can adjust their *broken* analysis and inform them about how to proceed. If users keep utilizing the system and do not step away after some failed attempts, the evaluation system and the research around it improves.

### 2.2.3 Databrowser – Standardized Model Data Access

The data browser of Freva is more than a search engine. It is a joint commitment to a common Earth system model data output standard within a research group. It was a step change in development when the climate communities first agreed on a specific data structure for model intercomparison projects (MIPs). As a consequence, nowadays, there are many opportunities to evaluate different models e.g. by the same tool. This means that software no longer needs to be adjusted to the model data to be analyzed. The next paragraph will give some insight in the technical details.

Freva's main data standard is the Data Reference Syntax (DRS) of CMIP5, which is publicly available at the ESGF. The DRS has distinct meta data requirements, including the Climate and Forecast (CF) meta data convention which uses NetCDF and the even more restrictive CMOR guidelines to bring meta data information into the directory structure of the model output database. This basic approach of using the CMOR options allows to set up a common and easy understandable model database for a research group. This database can be easily extended at later stage e.g. by model data of upcoming development stages of the research group or even the model data of users. Due to the fact that in the ESGF several data standards exist, Freva even gives the possibility to set up several different databases with different data standards, e.g. obs4mips, ana4mips, CORDEX, etc. at the same time. However, for a distinct plugin development, using these

meta data directories as options to retrieve data sets, it is recommended to use just one standard or at least imbedded standards like DRS or CMOR. Therefore, Freva also ships with some example scripts to standardize and re-standardize datasets. These scripts also help users to bring their own model results into the required standard format to ultimately incorporate the data into the system.

Freva indexes these output directory structures (model, reanalyses, observations, etc.) of the research group and saves this meta data information in a Solr database. Solr has a faceting component, which is part of the standard request handler which allows a faceted navigation. Therefore, Freva applies the Solr faceted search on the data directories and datasets using, for example, the DRS. All files of a chosen directory get registered or 'crawled', and thereafter all model datasets and their locations get ingested into the Solr server. The stand-alone Solr server is started via Java (see Sect. 2.3) and allows http requests. The System Core of Freva has a python class called *solr core* to encapsulate these requests to the Solr server. This way Freva retrieves the locations of the ingested model data sets via its meta data. This allows the assignment of the datasets to multiple categories. The scientific developer benefit from these categories to precisely different model data sets and exchange them easily. Plugins can use the databrowser to identify the model data needed for evaluation. The plugin interface of the *System Core* allows developers to clearly define which options in the *Solr fields* will be set by the users and which are pre-set by default values. If the data base contains a versioning of the database like e.g. DRS of CMIP5 does – which is recommended- Freva helps to keep track with the newest versions without unnecessary extra options. Per default the data browser lists the latest published data of an updated experiment set, but the search can be extended by all accessible versions. This is especially useful for reproduction of research results.



### 2.2.4 Virtual ESGF – Evaluation Data Extension

Nowadays, most of the computational data handling can be done via the internet. Cloud and Grid computing services offer fast IT solutions. However, Earth system modeling is still on the edge of possible or practical ways for scientists. Network processing of aggregated data (like yearly global means) is easily possible, but an analysis based on high spatial and temporal resolution data is extremely computationally expensive and time consuming. A long term hosting of several terabytes of external model data is not a practical solution. A database of a research project is usually increasing with time, e.g. the data amount of CMIP6 is estimated to be 20 times larger than that of CMIP5 [8].

Therefore, Freva offers a beta-version of a virtual database especially designed for the integration of ESGF projects into the databrowser. The next paragraph will give some insight in the technical details. The virtual ESGF maps a project like CMIP5 onto the respective data structure of the research project using the Filesystem in Userspace (FUSE) [9] as described in the following. For this purpose, we use Freva's ESGF API which addresses the ESGF via attributes and search facets. A listener script is running on the IT platform waiting for requests. Whenever a user or a plugin of Freva asks to access virtual datasets through the databrowser, only these are downloaded into a temporary cache. This cache is adjustable in a way that, for example, one month old unused data will be deleted automatically. During this time frame, the downloaded data is physically reachable. The virtual ESGF allows flexible adjustments while streaming them into the data browser. It is possible to map an ESGF project from the available standard into the research group's chosen standard. In addition, the research group can manipulate the data via NCO or CDO when known issues of data sets of the ESGF, e.g. wrong missing values need to be fixed.

The increased data resources through the virtual ESGF extend the evaluation possibilities for the research group without a restriction in the usability. The virtual ESGF can map several ESGF projects like CMIP5, CORDEX, obs4MIPs, etc. into Freva. An external dependency – the ESGF itself – is restricting the data accessibility and therefore the stability of Freva, which needs to be communicated inside the research group when using this powerful feature. Due to ESGF network availability, we recommend a clear separation of these virtual data sets from the local ones, customizable through the databrowser.

The topic 'virtual datasets' is still work in progress. While the design of the virtual ESGF is already fully developed, the practical implementation suffers from sporadic connectivity gaps to the ESGF.

### 2.3 General Software Lineup – Technical Details

Freva is designed to be implemented on IT platforms like Linux [10] for scientists in a research group including user account, compute resources, and storage. The

main framework, including shell executables and the web interface, is written in Python [11] using several third party packages. The whole system including the plugged-in tools are version controlled with GIT [7]. In the shell frontend, Freva is meant to be loaded by Modules [12] or sourced using preferably Bourne-again shell [13], thus allowing users to stay in the general work environment of, for example, an HPC. In the web frontend, which is built using Django [14], the users can log in via existing user accounts. Per default Freva is sourcing all user information via Lightweight Directory Access Protocol (LDAP) [15], granting/not granting access via group permissions. Therefore, it is not necessary to build an extra user database.

All communication between the web frontend and the HPC is realized via Secure Shell (SSH) [16] using the user account. Started plugins via web are handled by the Freva batch mode using a job scheduler, the Simple Linux Utility for Resource Management (SLURM) [17]. The database of all produced results is accounted to the user in a structure that is configurable and reachable from all processing hardware. Only the central databases stay within the central evaluation system, e.g. the plot preview section for the web page. This add-on keeps the preview graphics for the web small and available for the research project. These previews are produced by ImageMagick's `convert` [18] command. The results in the preview section are connected to the research results and plugin configurations of the history section and are stored in a database like MySQL [19].

The processed standardized Earth system model data can be found using the faceted search via the indexing Solr [20] server running in Java [21] as described in Section 2.2.3. Due to the fact that the Earth system model community, including the ESGF, mainly uses NetCDF data format, helpful accessory software are NetCDF libraries [22], NetCDF Operators [23] and Climate Data Operators [24]. For instance, `ncdump` of NetCDF is used to retrieve meta data for the web application. The virtual database of the ESGF is hosted by FUSE [9], taking care of the bridge between incoming dataset requests, their download, and virtual database caching.

All software setups are described in one configuration file of Freva, coordinating the combination of necessary programs, ports, and communicators.

## 3. SCIENTIFIC APPLICATION OF FREVA

Earth system models are important tools for climate science. While the models underwent major computational development stages in the last decades, verification systems are a bit behind the state-of-the-art technologies. However, evaluation system frameworks for the verification equations can be what Earth system model frameworks are for the primitive equations: A systematic computationally efficient tool to research the climate. We examine the

importance of a state-of-the-art evaluation system application and address its scientific development for Earth system modeling using the application example of decadal climate prediction.

Based on the corresponding plugin API in [Figure 5](#), a simple sample application of the usage is shown in [Figure 6](#). It shows an easy way of plugging a stand-alone tool into Freva. The automatic help during the progress supports its application. The MoviePlotter applies the *parameters.File* in the plugin directing the software to one file: The reanalysis of the mean sea level pressure from the ERA Interim reanalysis dataset [25]. The figure shows a quick analysis of Hurricane Katrina in the Gulf of Mexico. The application can be efficiently changed to a different variable, different reanalysis, different time range, etc. But with the basic idea of a very simple application, which only needs one input parameter to be used by other plugins for their plotting procedures, the MoviePlotter is just the first step in the evaluation complexity in decadal climate prediction science.

A more complex approach, using the CMOR facets directly in the plugin, is shown by the MurCSS tool [26] for decadal climate prediction research. It includes two independent CMOR option parts communicating with the Solr server (see Sec. 2.2.3). Thereafter, it is possible to compare two different model versions [27] or even two different experiment setups [28] against observations or reanalysis data. The development of this efficient basic validation tool for decadal evaluation in MiKlip (see Sect. 1 and [6]) framed by Freva, which ensures usability and reproducibility is a huge step forward in climate data verification. The research group may detect improvements in the research field ‘decadal prediction’ much faster and is able to share knowledge between scientists. Freva has been applied in decadal prediction research for example in the assessment of a future volcano eruption on forecasts [29], the development of novel forecast techniques [30], the investigation of the East Asian Monsoon [31], the assessment of the initial shock [32], the vertical skill evaluation compared to radiosondes [33], the effect of a wind-stress initialization method [34], the decadal skill due to volcano eruptions [35], the re-calibration of decadal predictions using observations [36], and the general research of the development stages in MiKlip [6] – to name a few. Many plugins<sup>15</sup> with different expertise have been developed and shared by Freva within the MiKlip research group.

Hosting an evaluation system of a research group via Freva rather than using stand-alone tools has even more advantages. It is not only scientific developers that can share knowledge by usable plugins; users can also share configurations or even results. This can be done actively via saving the configuration in the shell or by *share results* in the web with colleagues of the research group. But this could be also done passively using big data approaches

by Freva. While filling out the web form of a plugin, Freva automatically scans the history database and looks for similar configurations. Even before the plugin is started, the web interface suggests to use results of previously performed experiments, maybe even by other users. This is possible, because Freva is an open system, and all results are accessible by the entire research group. On the research side, this improves the research group’s connectivity and saves time for the users. New ideas can be developed as researches can be more productive. From the HPCs point of view, this saves CPU node hours, I/O, disk space, and energy.

Evaluation systems framed by Freva can be found at the Freie Universität Berlin<sup>16</sup> for research and teaching, at the DKRZ for the MiKlip<sup>17</sup> project for decadal climate prediction research and CMIP6 [8] project for scientific applications and evaluations done by the ESMVal,<sup>18</sup> at the Research Applications Laboratory (RAL) of the National Center for Atmospheric Research (NCAR) for MET tools applications,<sup>19</sup> and at the German Weather Service (DWD) for interdisciplinary meteorological analysis and visualizations.<sup>20</sup>

#### 4. DISCUSSION AND CONCLUSION

This paper introduced a complex and efficient framework for the evaluation of data in the context of Earth system modeling, the Freva system. The simple but yet powerful concept of the collective commitment to a common data standard (CMOR) and the applicable provision of knowledge on Earth system model science offers the potential to improve the efficiency of research groups. Freva as a host respects the fact that scientists need their scope for development to detect scientific findings. Freva emphasizes transparency and reproducibility of open science in a research project. Plugged-in tools and experiments are reviewable, editable, and repeatable. Although it is desirable to exclusively use the most efficient programming language as the common language in a project, Freva allows to plugin stand-alone tools in a variety of programming languages. Freva enables the utilization of a multitude of software plugins by acquainting only one common framework. The combination of the easy use with the flexibility of incorporating user specific data sets in agreement with research group’s standardization of model data, reanalysis, observations, or even ESGF data, is a huge advantage.

Furthermore, Freva supports research groups in terms of sustainability. The full control of a constructed evaluation system by including user specific data and plugging-in individual interfaces and the group’s version control is mandatory for a software system in science. Due to the commitment of a research group to work together in a central system like Freva, there is a need for an efficient and convenient communication. For the growth and quality of the system it is also important to

invite and convince scientists to be part of the common framework. Therefore, Freva addresses three types of clients: The User, the Developer, and the Admin. All of them are usually scientists with different research aims. Users usually want to start very simple when using such a system. Because of the, in our opinion, comprehensible web platform, users usually get started right away. Over time the user's requirements get more and more complex. After that, users sometimes jump to cloning, adapting, and replugin of a versioned plugin. Freva is at its most impressive when users become developers, and scientists start to cooperate on scientific tasks. Freva guides scientists over technical hurdles and allows them to concentrate on science itself. Another well-known issue in science is the fluctuation of scientists in research groups. A clear infrastructure set up with Freva can help to sustain and pass-on the knowledge and keep experience in the research group – even when developing scientists leave the field.

A major issue of the data structure is the change of standards from time to time, e.g. the progression from CMIP5 to CMIP6. Freva tackles this challenge easily, as it is fully adjustable in terms of data standards, and new standards can be included anytime. However, for one plugin it is difficult to deal with different data standards having different attributes. Setting one common data standard and re-standardizing other data sets according to it is the most efficient way for the plugins. Freva is flexible enough, that the data standard could be also set to a completely independent version of the research group – aside from standards in the ESGF.

A publication of a software package is always just a snapshot of what has been developed up to that very moment. The software design may have changed over time, but the main system framework idea has remained the same ever since we started the development of Freva in 2011. Clear interfaces in terms of tools and data have been established. A well-structured and stable model database was set up, which is flexible to adapt to the research group's needs. Freva offers automated reproducibility and transparency while increasing the usability of tools by different programming languages in shell and web on a HPC. The share of knowledge can be advanced by developing plugins together and by providing Earth system model data. In addition it is possible to produce, share and discuss results of the evaluation system within the research group. Retrospectively, the MiKlip project and Freva have been mutually beneficial for one another. Many plugins have been developed and shared, and a huge model database has been produced within the MiKlip Central Evaluation System for decadal climate prediction as seen in Section 3. The MiKlip project is a perfect example of a nationwide project with a special focus and plenty of scientists jointly working on one HPC. Freva, as a central infrastructure, organized MiKlip's tool development and data retrieval.

The efficient interaction between different technologies and the increased efficiency of evaluation frameworks next to modeling frameworks has and will further improve the Earth system modeling research.

## QUALITY CONTROL

Freva has been successfully applied in the MiKlip project producing a number of publications [29–36]. The framework has also been tested by several installations on HPC environments (Section 3). A functional install and test workflow is described in the guidelines and README file.

## (2) AVAILABILITY OPERATING SYSTEM

Linux (tested: Debian, SE-Linux, Suse, Fedora).

## PROGRAMMING LANGUAGE

Python 2.7 (will be installed by Freva) and BASH 3/4 (must exist).

## ADDITIONAL SYSTEM REQUIREMENTS

Description in Section 2.3 [MySQL server (must exist), Apache Solr Server (will be installed by Freva), SLURM scheduler (must exist), Java (must exist), Apache HTTP Server (must exist), LDAP (must exist), ImageMagick Convert (must exist), Modules (optional), Memory >6GB RAM, GIT, CDO, NCO]

## DEPENDENCIES

Libraries: NetCDF4, python-dev, MySQL, MySQL-python == 1.2.5, Django >= 1.8, <1.9, pyPdf == 1.13, numpy == 1.9.2, netCDF4 == 1.1.1, nco == 0.0.2, cdo == 1.2.3 virtualenv == 13.1.2

## LIST OF CONTRIBUTORS

The MiKlip project (fona-miklip.de) contributed by testing and using Freva. Estanislao Gonzalez (former FU Berlin) developed the basic core of the plugin and database system.

## SOFTWARE LOCATION

### Archive

**Name:** Zenodo

**Persistent identifier:** <http://doi.org/10.5281/zenodo.1325148>

**Licence:** Free-BSD

**Publisher:** Christopher Kadow

**Version published:** v1.0-beta

**Date published:** 01/08/18

### Code repository

**Name:** GitHub

**Identifier:** <https://github.com/FREVA-CLINT/Freva>

**Licence:** FreeBSD

**Date published:** 01/08/18

**Language:** English

### (3) REUSE POTENTIAL

Freva can be reused in the Earth system modeling community following the data standards as described in the paper (CMOR, DRS). The software is adaptable to the research groups' needs. The publication including the repository on GitHub is the first step of an open development. Future support is depending on future funding.

### NOTES

- 1 <http://www.ipcc.ch>.
- 2 <https://www.mpimet.mpg.de/cdo>.
- 3 <http://doi.org/10.5281/zenodo.809463>.
- 4 <https://www.nceas.ucsb.edu/globalmarine>.
- 5 <https://rcmes.jpl.nasa.gov>.
- 6 <https://www.esmvaltool.org>.
- 7 <http://clivar-dpwg.iri.columbia.edu>.
- 8 <https://climexp.knmi.nl>.
- 9 <https://github.com/bird-house>.
- 10 <https://cds.climate.copernicus.eu>.
- 11 <https://esgf.llnl.gov>.
- 12 Developed at the Freie Universität Berlin, Freva's naming is a wordplay of a free software and the German name 'Freie', which means 'free and independent'.
- 13 <https://www-miklip.dkrz.de>.
- 14 <https://esgf.llnl.gov>.
- 15 [www-miklip.dkrz.de/plugins](http://www-miklip.dkrz.de/plugins).
- 16 [freva.met.fu-berlin.de](http://freva.met.fu-berlin.de).
- 17 [www-miklip.dkrz.de](http://www-miklip.dkrz.de).
- 18 [cmip-eval.dkrz.de](http://cmip-eval.dkrz.de).
- 19 [freva.rap.ucar.edu](http://freva.rap.ucar.edu) – behind firewall.
- 20 [mavis.dwd.de](http://mavis.dwd.de) – behind firewall.

### ACKNOWLEDGEMENTS

All Miklip project members need to be acknowledged. The authors thank Estanislao Gonzalez for his valuable contribution for Freva within the first months of development.

### FUNDING STATEMENT

The authors would like to acknowledge funding from the Federal Ministry of Education and Research in Germany (BMBF) through the research program Miklip II (FKZ: 01LP1519A, 01LP1519B, 01LP1520A) and the CMIP6-DICAD project (FKZ: 01LP1605D), and from CoreLogic SARL Paris.

### COMPETING INTERESTS

The authors have no competing interests to declare.

### AUTHOR AFFILIATIONS

**Christopher Kadow**  [orcid.org/0000-0001-6537-3690](https://orcid.org/0000-0001-6537-3690)

Deutsches Klimarechenzentrum (DKRZ), Germany; Freie Universität Berlin, Institute of Meteorology, Germany

**Sebastian Illing**

Freie Universität Berlin, Institute of Meteorology, Germany

**Etor E. Lucio-Eceiza**  [orcid.org/0000-0002-3698-852X](https://orcid.org/0000-0002-3698-852X)

Deutsches Klimarechenzentrum (DKRZ), Germany; Freie Universität Berlin, Institute of Meteorology, Germany

**Martin Bergemann**  [orcid.org/0000-0003-0604-4103](https://orcid.org/0000-0003-0604-4103)

Deutsches Klimarechenzentrum (DKRZ), Germany

**Mahesh Ramadoss**  [orcid.org/0000-0003-3167-3884](https://orcid.org/0000-0003-3167-3884)

Deutsches Klimarechenzentrum (DKRZ), Germany; Freie Universität Berlin, Institute of Meteorology, Germany

**Philipp S. Sommer**  [orcid.org/0000-0001-6171-7716](https://orcid.org/0000-0001-6171-7716)

Helmholtz Zentrum Geesthacht (HZG), Institute of Coastal Systems – Analysis and Modeling, Germany

**Oliver Kunst**

Freie Universität Berlin, Institute of Meteorology, Germany

**Thomas Schartner**

Freie Universität Berlin, Institute of Meteorology, Germany; Deutscher Wetterdienst (DWD), Germany

**Klaus Pankatz**  [orcid.org/0000-0001-6159-8139](https://orcid.org/0000-0001-6159-8139)

Deutscher Wetterdienst (DWD), Germany

**Jens Grieger**  [orcid.org/0000-0003-0985-8479](https://orcid.org/0000-0003-0985-8479)

Freie Universität Berlin, Institute of Meteorology, Germany

**Mareike Schuster**

Freie Universität Berlin, Institute of Meteorology, Germany

**Andy Richling**  [orcid.org/0000-0003-1745-2479](https://orcid.org/0000-0003-1745-2479)

Freie Universität Berlin, Institute of Meteorology, Germany

**Hannes Thiemann**  [orcid.org/0000-0002-2329-8511](https://orcid.org/0000-0002-2329-8511)

Deutsches Klimarechenzentrum (DKRZ), Germany

**Ingo Kirchner**  [orcid.org/0000-0002-4103-6849](https://orcid.org/0000-0002-4103-6849)

Freie Universität Berlin, Institute of Meteorology, Germany

**Henning W Rust**  [orcid.org/0000-0003-0763-3954](https://orcid.org/0000-0003-0763-3954)

Freie Universität Berlin, Institute of Meteorology, Germany

**Thomas Ludwig**

Deutsches Klimarechenzentrum (DKRZ), Germany

**Ulrich Cubasch**  [orcid.org/0000-0001-9628-4666](https://orcid.org/0000-0001-9628-4666)

Freie Universität Berlin, Institute of Meteorology, Germany

**Uwe Ulbrich**  [orcid.org/0000-0001-7558-6622](https://orcid.org/0000-0001-7558-6622)

Freie Universität Berlin, Institute of Meteorology, Germany

### REFERENCES

1. **Wilkins-Diehr N.** Science Gateways—Common Community Interfaces to Grid Resources. *Concurrency Computat.: Pract. Exper.* 2007; 19(6): 743–749. DOI: <https://doi.org/10.1002/cpe.1098>
2. **Lawrence KA, Zentner M, Wilkins-Diehr N, Wernert JA, Pierce M, Marru S, Michael S.** Science gateways today and tomorrow: positive perspectives of nearly 5000 members of the research community. *Concurrency Computat.: Pract. Exper.* 2015; 27: 4252–4268. DOI: <https://doi.org/10.1002/cpe.3526>



3. **Goecks J, Nekrutenko A, Taylor J.** Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* 2010; 11(8): R86. DOI: <https://doi.org/10.1186/gb-2010-11-8-r86>
4. **Madhavan K, Zentner L, Farnsworth V,** et al. [nanoHUB.org](http://nanoHUB.org): cloud-based services for nanoscale modeling, simulation, and education. *Nanotechnology Reviews.* 2013; 2(1): 107–117. DOI: <https://doi.org/10.1515/ntrev-2012-0043>
5. **Stanzione D.** The iPlant Collaborative: Cyberinfrastructure to Feed the World. *Computer.* 2011; 44(11): 44–52. DOI: <https://doi.org/10.1109/MC.2011.297>
6. **Marotzke J, Müller WA, Vamborg FSE, Becker P, Cubasch U, Feldmann H, Kaspar F, Kottmeier C, Marini C, Polkova I, Prömmel K, Rust HW, Stammer D, Ulbrich U, Kadow C, Köhl A, Kröger J, Kruschke T, Pinto JG, Pohlmann H, Reyers M, Schröder M, Sienz F, Timmreck C, Ziese M.** MiKlip: A National Research Project on Decadal Climate Prediction. *Bulletin of the American Meteorological Society.* 2016; 97: 2379–2394. DOI: <https://doi.org/10.1175/BAMS-D-15-00184.1>
7. **Hamano J, Torvalds L,** et al. GIT – Version Control System, [git-scm.com](http://git-scm.com), license: GNU General Public License. 2015.
8. **Eyring V, Bony S, Meehl GA, Senior CA, Stevens B, Stouffer RJ, Taylor, KE.** Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development.* 2016; 9: 1937–1958. DOI: <https://doi.org/10.5194/gmd-9-1937-2016>
9. **Szeredi M.** FUSE – Filesystem in Userspace, <http://fuse.sourceforge.net/>, license: GPL for kernel part, LGPL for Libfuse. 2015.
10. **GNU/Linux Community Linux.** [kernel.org](http://kernel.org), Linus Torvalds. 2015.
11. **Python Software Foundation.** Python Language Reference, 2.7.10, <http://www.python.org>, “G. van Rossum, Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.” 2015.
12. **Environment Modules Project.** Environment Modules, <http://modules.sourceforge.net>, “John L. Furlani, “Modules: Providing a Flexible User Environment”, Proceedings of the Fifth Large Installation Systems Administration Conference (LISA V), pp. 141–152, San Diego, CA, September 30 – October 3, 1991.”, License: GNU General Public License (version 2). 2015.
13. **Fox B, GNU Project.** BASH – Unix Shell and Command Language, <https://www.gnu.org/software/bash>. license: GNU GPL v3+. 2015.
14. **Django Software Foundation.** Django Web Framework, 1.8, Lawrence Journal-World, License: 3-clause BSD. 2015.
15. **Carter G.** LDAP System Administration. 2003. O’Reilly. ISBN 1-56592-491-6.
16. **OpenSSH project.** OpenSSH SSH client, <http://www.openssh.com/>, openSSH is a derivative of the original free ssh 1.2.12 release from Tatu Ylönen. 2015.
17. **Slurm Commercial Support and Development.** Simple Linux Utility for Resource Management Workload Management, [slurm.schedmd.com](http://slurm.schedmd.com), “Slurm: Simple Linux Utility for Resource Management, A. Yoo, M. Jette, and M. Grondona, Job Scheduling Strategies for Parallel Processing, volume 2862 of Lecture Notes in Computer Science, pages 44–60, Springer-Verlag, 2003.” License GNU General Public License 2. 2015. DOI: [https://doi.org/10.1007/10968987\\_3](https://doi.org/10.1007/10968987_3)
18. **ImageMagick Studio LLC.** ImageMagick – a software suite to create, edit, compose, or convert bitmap images, 6. 2015. <http://www.imagemagick.org/script/index.php>
19. **Oracle Corporation.** MySQL – relational database management system, [www.mysql.com](http://www.mysql.com), original author(s) MySQL AB, License: GPL (version 2), 2015a.
20. **Apache.** Solr – open source enterprise search platform <http://lucene.apache.org/solr/>, apache Lucene Project, License: Apache License 2.0. 2015.
21. **Oracle Corporation.** Java language, designed by James Gosling and Sun Microsystems, License: GNU General Public License. 2015b.
22. **University Corporation for Atmospheric Research.** NetCDF – Network Common Data Form, 4. 2015. <http://www.unidata.ucar.edu/software/netcdf>
23. **The NCO project.** NCO – netCDF Operator. 2015. Version 4.5.0, <http://nco.sourceforge.net/>. Zender CS. Analysis of self-describing gridded geoscience data with netCDF Operators (NCO). *Environmental Modelling and Software.* 2008; 23(10–11): 1338–1342. DOI: <https://doi.org/10.1016/j.envsoft.2008.03.004>
24. **Max-Planck-Institute for Meteorology.** Climate Data Operators, 1.6. 2015. <https://code.zmaw.de/projects/cdo>, license: GPL v2.
25. **Dee DP, Uppala SM, Simmons AJ, Berrisford P, Poli P, Kobayashi S, Andrae U, Balmaseda MA, Balsamo G, Bauer P, Bechtold P, Beljaars ACM, van de Berg L, Bidlot J, Bormann N, Delsol C, Dragani R, Fuentes M, Geer AJ, Haimberger L, Healy SB, Hersbach H, Holm EV, Isaksen L, Kallberg P, Koehler M, Matricardi M, McNally AP, Monge-Sanz BM, Morcrette JJ, Park, BK, Peubey C, de Rosnay P, Tavolato C, Thepaut JN, Vitart F.** The ERA-Interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal Of The Royal Meteorological Society.* 2011; 137(656): 553–597. DOI: <https://doi.org/10.1002/qj.828>
26. **Illing S, Kadow C, Kunst O, Cubasch U.** MurCSS: A Tool for Standardized Evaluation of Decadal Hindcast Systems. *Journal of Open Research Software.* 2014; 2(1): e24. DOI: <https://doi.org/10.5334/jors.bf>
27. **Pohlmann H, Müller WA, Kulkarni K, Kameswarrao M, Matei D, Vamborg FSE, Kadow C, Illing S, Marotzke J.** Improved forecast skill in the tropics in the new MiKlip decadal climate predictions. *Geophysical Research Letters.* 2013; 40: 5798–5802. DOI: <https://doi.org/10.1002/2013GL058051>
28. **Kadow C, Illing S, Kunst O, Rust H, Pohlmann H, Müller WA, Cubasch U.** Evaluation of Forecasts by Accuracy and Spread in the MiKlip Decadal Climate Prediction System. *Meteorologische Zeitschrift.* 2015. DOI: <https://doi.org/10.1127/metz/2015/0639>

29. **Illing S, Kadow C, Pohlmann H, Timmreck C.** Assessing the impact of a future volcanic eruption on decadal predictions. *Earth System Dynamics*. 2018; 9: 701–715. DOI: <https://doi.org/10.5194/esd-9-701-2018>
30. **Kadow C, Illing S, Kröner I, Ulbrich U, Cubasch U.** Decadal climate predictions improved by ocean ensemble dispersion filtering. *Journal of Advances in Modeling Earth Systems*. 2017; 9: 1138–1149. DOI: <https://doi.org/10.1002/2016MS000787>
31. **Huang B, Cubasch U, Kadow C.** Seasonal prediction skill of East Asian summer monsoon in CMIP5 models. *Earth System Dynamics*. 2018; 9: 985–997. DOI: <https://doi.org/10.5194/esd-9-985-2018>
32. **Kröger J, Pohlmann H, Sienz F, Marotzke J, Baehr J, Köhl A, Modali K, Polkova I, Stammer D, Vamborg FSE, Müller WA.** Full-field initialized decadal predictions with the MPI earth system model: an initial shock in the North Atlantic. *Climate Dynamics*. 2017; 51(7–8), 2593–2608. DOI: <https://doi.org/10.1007/s00382-017-4030-1>
33. **Pattantyús-Ábrahám M, Kadow C, Illing S, Müller WA, Pohlmann H, Steinbrecht W.** Bias and Drift of the Medium-Range Decadal Climate Prediction System (MiKlip) validated by European Radiosonde Data. *Meteorologische Zeitschrift*. 2016; 25(6): 709–720. DOI: <https://doi.org/10.1127/metz/2016/0803>
34. **Thoma M, Greatbatch R, Kadow C, Gerdes R.** Decadal hindcasts initialised using observed surface wind stress: Evaluation and Prediction out to 2024. *Geophysical Research Letters*. 2015; 42(15): 6454–6461. DOI: <https://doi.org/10.1002/2015GL064833>
35. **Timmreck C, Pohlmann H, Illing S, Kadow C.** The impact of stratospheric volcanic aerosol on decadal scale predictability. *Geophysical Research Letters*. 2015; 43(2): 834–842. DOI: <https://doi.org/10.1002/2015GL067431>
36. **Pasternack A, Bhend J, Liniger MA, Rust HW, Müller WA, Ulbrich U.** Parametric decadal climate forecast recalibration (DeFoReSt 1.0). *Geoscientific Model Development*. 2018; 11: 351–368. DOI: <https://doi.org/10.5194/gmd-11-351-2018>

---

**TO CITE THIS ARTICLE:**

Kadow C, Illing S, Lucio-Eceiza EE, Bergemann M, Ramadoss M, Sommer PS, Kunst O, Schartner T, Pankatz K, Grieger J, Schuster M, Richling A, Thiemann H, Kirchner I, Rust HW, Ludwig T, Cubasch U, Ulbrich U 2021 Introduction to Freva – A Free Evaluation System Framework for Earth System Modeling. *Journal of Open Research Software*, 9: 13. DOI: <https://doi.org/10.5334/jors.253>

Submitted: 24 November 2018   Accepted: 15 May 2021   Published: 02 June 2021

**COPYRIGHT:**

© 2021 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press.