

Table of Contents

DRM4G	2
About DRM4G	2
Why DRM4G?	2
What is DRM4G's architecture?	2
New releases	2
Forthcoming features	2
Releases	2
Getting started	3
Examples of usage	3
DRM4G CLI	3
Python API	3
Tutorial	3
Contributing	4
FAQ	4

DRM4G

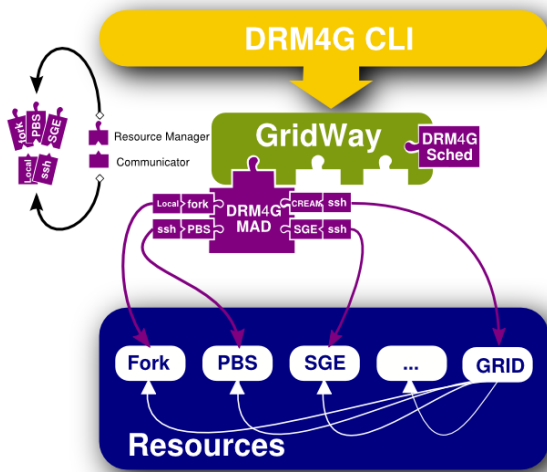
About DRM4G

DRM4G is an open platform, based on [?GridWay](#), to define, submit, and manage computational jobs. DRM4G is a Python (**2.6+**, **3.3+**) implementation that provides a single point of control for computing resources without installing any intermediate middlewares. As a result, a user is able to run the same job on laptops, desktops, workstations, clusters, supercomputers, and any grid.

Why DRM4G?

- **Ready-to-run:** The DRM4G bundle can be unpacked and then ran on any on any Linux OS.
- **Huge amount of jobs:** DRM4G is able to manage around 1,000,000 jobs at the same time.
- **Different identities on different resources:** DRM4G handles different user accounts on various resources through a single interface.
- **One command:** In order to make easier to use DRM4G, there is only one command available, which is `drm4g`.

What is DRM4G's architecture?



Infrastructure layer consists of the following components :

- **DRM4G CLI** offers users commands to submit, cancel, and monitor jobs and configure resources.
- **GridWay core** is in charge of job execution and resource brokering. DRM4G takes the most of GridWay by using its core.
- **DRM4G Middleware Access Driver (MAD)** gives access to computing resources. In order to do that, there are two kind of components:
 - Resource Manager is responsible for collecting resource information.
 - Communicator provides the authentication type to access them.
- **DRM4G Sched** is responsible for scheduling jobs.

New releases

Forthcoming features

- Management of VOs (Virtual Organizations)
 - Dynamically create and destroy VMs
- Integration of cloud providers based on LibCloud's API

Releases

- [DRM4G Release 2.6.4](#) · 2017-03-03 ·
- [DRM4G Release 2.6.3](#) · 2017-02-09 ·
- [DRM4G Release 2.6.2](#) · 2017-01-12 ·

[List of all releases](#)

Getting started

[Installation](#)

- [With pip](#)
- [Without pip](#)
- [Verifying the installation](#)
- [Optional Environment Variables](#)
- [Installing Extras](#)

[Resource Configuration](#)

- [Configuration format](#)
- [DEFAULT section](#)
- [Resource section](#)
- [Examples](#)

[Job Description](#)

- [Syntax](#)
- [Template options](#)
- [Requirement Expressions](#)
- [Environment Expressions](#)

Examples of usage

This section contains examples on how to use the DRM4G CLI.

- [How to configure a TORQUE/PBS resource by using the private/public key without password](#)
- [How to configure an EGI ESR VM](#)
- [How to configure an IBERGRID VM](#)
- [How to configure an EGI FedCloud/rOCCI VM](#)
- [How to manage a job from start to end](#)

DRM4G CLI

`drm4g` is DRM4G's Command Line Interface (CLI) for administrating both computing resources and jobs.

- [Available Commands](#)

Python API

In order to use the DRM4G API, you only have to install DRM4G on your machine. This python API allows you to create, submit, cancel and monitor DRM4G jobs.

[DRM4G API](#)

- [How to create a job](#)
- [How to submit a job](#)
- [How to monitor a job](#)

Tutorial

[DRM4G tutorial](#)

- [Start to run DRM4G](#)
- [My first job](#)
- [How to configure a TORQUE/PBS resource using the ssh-agent](#)
- [User Scenarios](#)
- [?Video from presentation given by C. Blanco at HPCKP'15](#)

Contributing

- [Development](#)

FAQ

- [Frequently Asked Questions](#)
- [Common Errors](#)