

Milestone Planning

g-Eclipse Milestone Planning

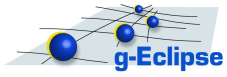
WP1.3

Document Filename:	MilestonePlanning.pdf
Workpackage:	WP1.3
Partner(s):	FZK, INO, JKU, PSNC, RUR, UCY
Lead Partner:	FZK
Config ID:	ge-1
Document classification:	PUBLIC

Abstract: This document is intended to fix the g-Eclipse milestone plan and release cycle.

Contents

1 Milestone Plan	4
1.1 Versioning	4
1.2 Planning	4
2 Milestone Release Cycle	5
2.1 Phases	5
2.2 Meetings	5
3 Workpackage Tasks	8
3.1 Work Package 1 - General Architecture	8
3.1.1 Core Architecture	8
3.1.2 Grid Model Views/Project View	8
3.1.3 Authentication and Authorization	8
3.2 Work Package 2 - User Perspective	9
3.2.1 Job Wizard	9
3.2.2 Job Management	9
3.2.3 Data management I	9
3.3 Work Package 3 - Operator Perspective	9
3.3.1 VO Resource Allocations (VORA) / Local Queue Management (LQM)	9
3.3.2 Grid Infrastructure Benchmarking and Testing (GIB)	10
3.3.3 Local Job Management (LJM)	11
3.3.4 Grid Status Report	11
3.3.5 JSDL Editor	11
3.4 Work Package 4 - Developer Perspective	12
3.4.1 Grid Command Console (glogin)	12
3.4.2 Grid Visualization	12
3.4.3 Development Tools	12
3.4.4 Application Debugger	12
3.4.5 Application Monitoring	13
3.4.6 Application Deployment	13
3.4.7 Grid Workflow Builder	13



4	Timeline	14
4.1	0.5M0 - 01/2007	14
4.2	0.5M1 - 02/2007	14
4.3	0.5M2 - 03/2007	14
4.4	0.5M3 - 04/2007	15
4.5	0.5M4 - 05/2007	16
4.6	0.5 - 06/2007	16
4.7	Planning Further	16
4.8	Overview	17

1 Milestone Plan

1.1 Versioning

In this planning document we define the Milestones (labeled as “M”) and Releases (labeled as “R”). The following milestones are planned in the next months. As the project progresses, a more detailed Milestone Plan for version 1.0 will be provided.

Within the g-Eclipse project the following milestones are defined as of March 27, 2007:

Month	Title	Type	Version	See section
01/2007	First prototype	Milestone	0.5M0	
02/2007	First prototype	Milestone	0.5M1	4.2
03/2007	First prototype	Milestone	0.5M2	4.3
04/2007	First prototype	Milestone	0.5M3	4.4
05/2007	First prototype	Milestone	0.5M4	4.5
06/2007	First release	Release	0.5	4.6
12/2007	Final candidate release	Release	1.0R1	
03/2008	Final release	Release	1.0R2	

Version 0.5 is called the first minor prototype release. Version 1.0 is called the mature release. All above mentioned Milestone Releases will include new functionalities, code documentation, associated cheat sheets, help pages and general documentation. New functionalities have to be superior in the sense that they provide new fully functional features for the platform. An example would be the support for a new middleware.

A Milestone Release therefore consists of fine-grained fragments of the features that are planned for the next release. It should always be possible to implement these fragments on a timescale that fits the timescale for a Milestone Release. This implementation period includes not only actual implementation time but also the evaluation, testing and documentation phases (see also chapter 2).

The version number of g-Eclipse will consist of three parts – the major version number, the minor version number and the Milestone build number as listed in table 1.1. Major releases will not have a Milestone identifier. So the first major release will have the versioning **g-Eclipse 0.5**.

1.2 Planning

The planning of our features will be g-Eclipse perspective-specific. g-Eclipse plans support for three different actors (users, operators and developers) by providing customized perspectives. For each of these perspectives, a separate plan for the next release and the intermediate milestones is provided.

Items that are planned for a release are called ‘features’ (not to be confused with Eclipse features). The subdivision of such a feature into milestone fragments will be called tasks. So every feature consists of a bunch of tasks. A task therefore should be a portion of work that can be done within a Milestone Release cycle.

2 Milestone Release Cycle

The g-Eclipse project aims for monthly Milestone Releases (see table 1.1). These public releases will take place at the last Friday of each month or on the following day that is not an official holiday.

2.1 Phases

The period between two Milestone Releases is subdivided into four phases:

Planning and Evaluation phase: This phase should cover roughly one week. It is dedicated to the planning of new features and tasks that will be implemented for the next Milestone Release and to the evaluation of all issues concerning the implementation phase that follows.

Implementation phase: This phase should cover roughly two weeks. It, in fact, fills the gap between the *Planning and Evaluation phase* and the *Quality Assurance phase*, i.e. the duration of this phase depends on the current month. It is dedicated to the implementation of the new features and tasks for the next Milestone Release. This phase ends with a feature freeze, i.e. after this phase, no new features will be implemented for the current release. All following work will be dedicated to the improvement of the new features and to Quality Assurance issues.

Quality Assurance phase: This phase should cover roughly one week. It is dedicated to everything that is related to Quality Assurance, i.e. code cleanup, code documentation, general documentation, provision of context sensitive help and cheat sheets and implementation of test cases.

Release phase: This phase starts on the last Wednesday before the Milestone Release and lasts two days until the release itself. It is dedicated to last minute bug fixes and other quality issues. The aim of this phase is to provide a stable, high quality release candidate for the Milestone Release. This phase will end on the last Friday of the month with the release of the milestone build. On the last Thursday before the release, at 4 PM, a complete code freeze will take place and a snapshot will be taken from the repositories in order to prepare the milestone build that will take place as part of our nightly builds.

The transition between two phases is continuous. Nevertheless, every developer has to at least roughly adhere to these phases.

2.2 Meetings

We will have weekly VRVS meetings on every Wednesday that is not an official holiday. These technical meetings will take place at 1 PM UTC (i.e. 2 PM CET). The meetings are dedicated to technical and release cycle related issues and will therefore be open to the public. They will be announced on our Eclipse webpage. Furthermore, we will have internal meetings concerning administrative stuff. (Remark: The dates of these meetings have not been fixed as of writing this document.)

The technical meetings are synchronized with our release cycle. The resulting meeting cycle will therefore start on the first Wednesday (that is not an official holiday) after the last Milestone Release and will end on the last Wednesday before the next Milestone Release. This means we will have at least four and at most five meetings in a release cycle. These meetings will each have a different scope, depending on the related release phase:

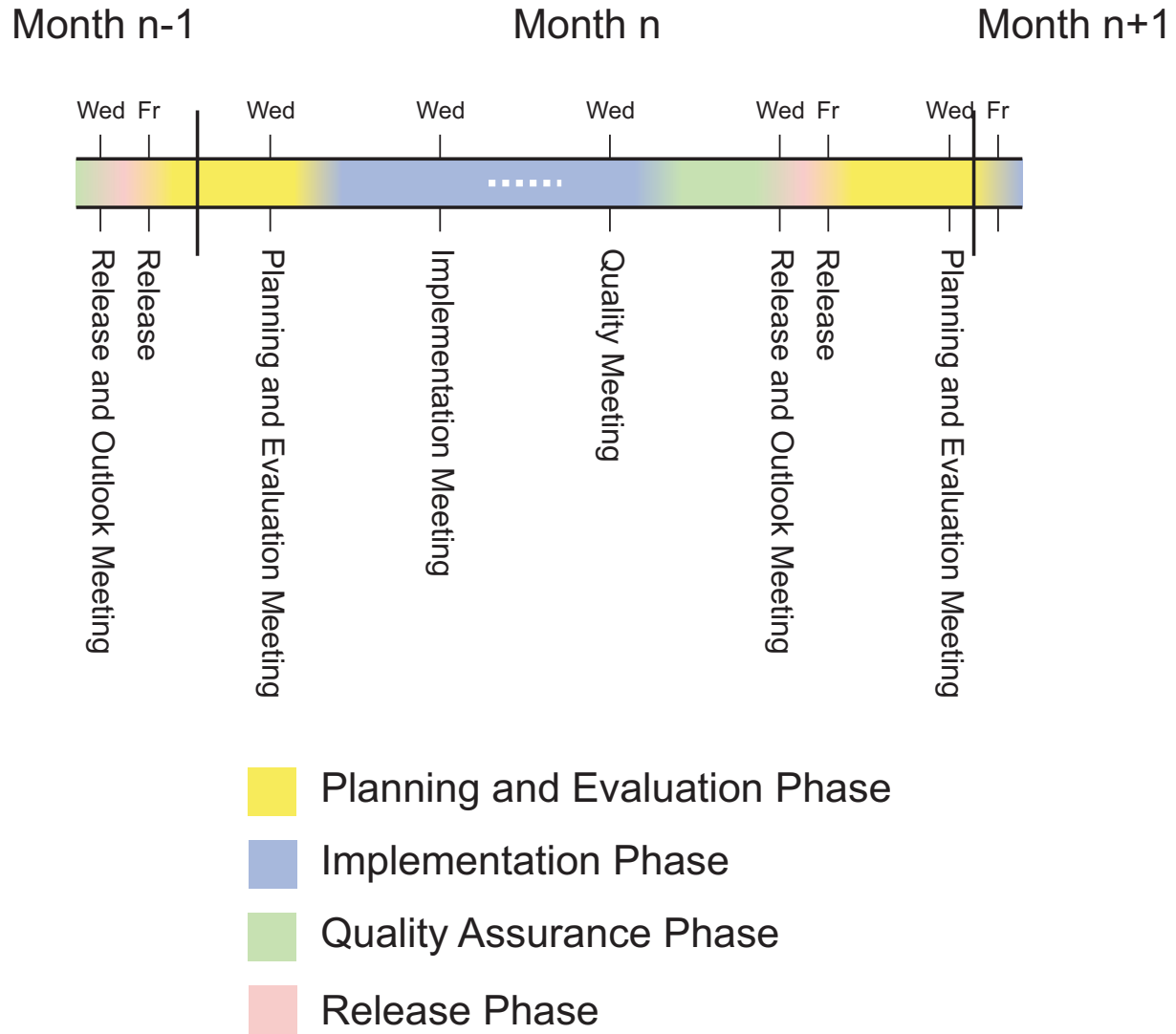


Figure 2.1: The g-Eclipse Release Cycle.

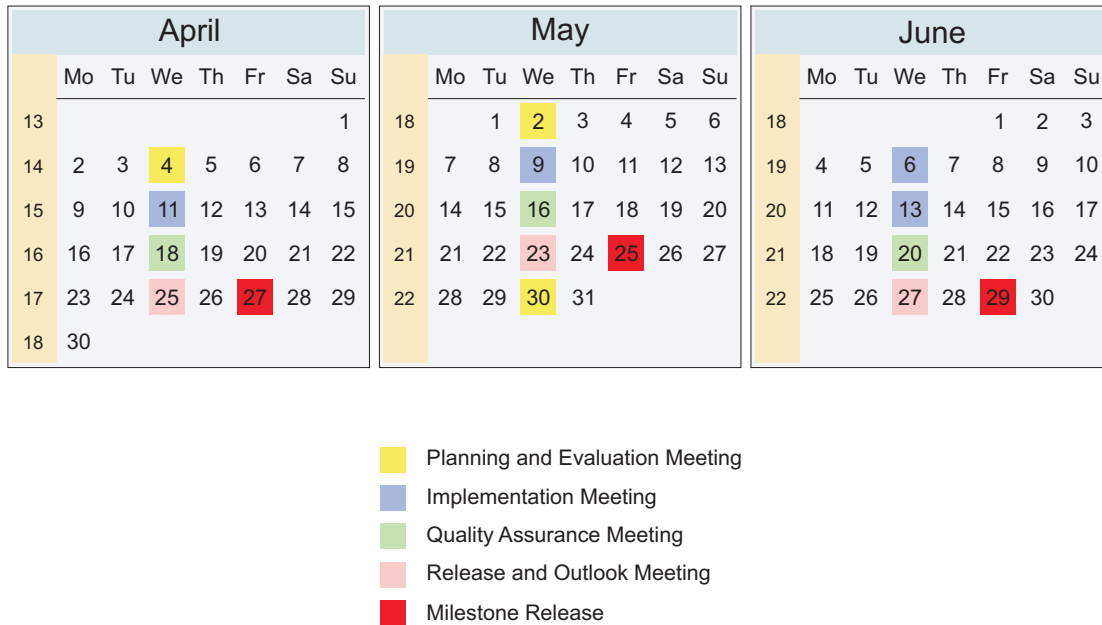


Figure 2.2: Overview of the g-Eclipse technical meetings up to June 2007.

Planning and Evaluation meeting: This meeting will take place on the first Wednesday after the last Milestone Release. During this meeting, every work package has to fix the features and tasks that will be implemented for the next Milestone Release. Furthermore, first evaluation results should be presented and discussed.

Implementation meetings: These meetings will fill the gap between the *Planning and Evaluation meeting* and the *Quality meeting*. The number of these meetings (either one or two) depends on the month in question. The purpose of these meetings is to discuss and present the current status of the implementation. Therefore we will make use of a VNC server.

Quality meeting: This meeting will take place on the Wednesday before the *Release and Planning meeting*. It is dedicated to Quality Assurance planning. This includes bug reports, code cleanup and documentation issues.

Release and Outlook meeting: This meeting will take place on the last Wednesday before the next Milestone Release. The main issue of this meeting is the provision of a stable, high quality release candidate. Last minute bug fixes and quality issues will be discussed and planned for the following release phase. Furthermore, an outlook to the next release candidate will be given and discussed.

Each of these meetings will be moderated by a chairman. This chairman has the responsibility to prepare the meeting and to write minutes. These minutes have to be made available to the public as soon as possible but at the latest until the Friday after the meeting. At the end of each meeting, the chairman of the next meeting will be nominated in order to allow him to prepare the next meeting.

3 Workpackage Tasks

The implementation of the functionalities for the different perspectives (user, operator and developer) will be done by different work packages. Additionally, a work package for architecture, quality assurance and general features will contribute to the g-Eclipse framework.

The following features will be provided for the different perspectives by the different work packages:

3.1 Work Package 1 - General Architecture

The general architecture will provide the following functionalities for the milestones of the g-Eclipse framework:

3.1.1 Core Architecture

Consolidation of new model

03-04/2007 0.5M2[4.3] 0.5M3[4.4]

Cleanup of model

04-05/2007 0.5M3[4.4] 0.5M4[4.5]

Provision of g-Eclipse problem reporting mechanism

05-06/2007 (see 0.5M4[4.5] 0.5[4.6])

Implementation of tar-support

03/2007 0.5M2[4.3]

3.1.2 Grid Model Views/Project View

Implementation of common actions

02-08/2007 0.5M1[4.2] 0.5M2[4.3] 0.5M3[4.4] 0.5M4[4.5] 0.5[4.6]

Implementation of Drag'n'Drop / Transfer Actions

03-04/2007 0.5M2[4.3] 0.5M3[4.4]

3.1.3 Authentication and Authorization

Full implementation of VOMS Proxy

04-05/2007 0.5M3[4.4] 0.5M4[4.5]

Make Grid Proxy independent of external jars

05-06/2007 0.5M4[4.5] 0.5[4.6]

Make VOMS Proxy independent of external jars

06-07/2007 0.5[4.6]

3.2 Work Package 2 - User Perspective

3.2.1 Job Wizard

- Switch to JSDL XML model and produce proper JSDL/JDL
03/2007
connection with JSDL editor
- New page for resource requirement
04-05/2007
- New page (extension point) for application defined parameters
05-05/2007

3.2.2 Job Management

- Reading job status from Logging and Book-keeping (L&B)
03-04/2007
- Jobs viewer for browsing and filtering jobs
03-05/2007
- Job viewer/editor(?) for showing given job details
03/2007

3.2.3 Data management I

- More operations for remote GridFTP filesystems
03-04/2007
- Support for LFC/SRM (using auxiliary gLite server)
04-05/2007

3.3 Work Package 3 - Operator Perspective

3.3.1 VO Resource Allocations (VORA) / Local Queue Management (LQM)

02/2007 –

- Create a client that will extract the information published by the gLite CE Monitor web-service.
- Create a client API to get read and write access to the PBS and Maui services on a gLite CE.

- Design of an abstraction layer for a queue manager and the ability to load and store an abstract view of the queue settings.
- Incorporate a web-browser in the view with bookmarks to various grid monitoring and test sites.

03 2007 –

- Design and implement an editor (GUI interface) that displays the abstract view of the queue manager.
- Design and implement a queue storage format to store the state of the queue configuration to permanent storage. Also to load a stored configuration back to the queue manager interface.

04/2007 –

- Connect together work from month 1 and 2 and provide a queue manager through g-Eclipse to the gLite middleware. With the possibility to plug in other middleware queue managers like LSF, etc.

05/2007 –

- Extend the properties that can be accessed and changed at run time to the various components like VO, queues, worker nodes, etc. Also, the option to sort the different worker nodes based on the attributes will be possible.

06/2007 –

- Gather system information (memory usage, disk usage, CPU(s) usage, number of defunct processes, etc.) from Linux OS, also ability to fetch various log files. In the editor, a tabbed panel will be added, that when a node is selected, the above information is displayed.

3.3.2 Grid Infrastructure Benchmarking and Testing (GIB)

02/2007 –

- Dependencies: Requires ROBUST job submission
- Initial benchmarking support
- Benchmark result persistence

03/2007 –

- Result Graphing

04/2007 –

- More benchmarks / Graphing

05/2007:

- More benchmarks/Graphing

3.3.3 Local Job Management (LJM)

04/2007 –

- Start working on LJM; cancel and restart jobs.

04/2007 –

- Migrate jobs from one worker-node to another for management purposes.

06/2007 –

- Complete LJM

3.3.4 Grid Status Report

03-04/2007 –

- Support for EGEE SAM tests

3.3.5 JSDL Editor

01/2007 –

- Design an initial JSDL Editor Form with the following available elements and attributes:
 - Job Definition
 - Job Description
 - Job Identification
 - Application
- Create a basic JSDL file by filling out the required / available fields / tabs of the Eclipse form (JSDL Editor).
- Load an existing JSDL file from the Local File System or the Grid in the JSDL editor and edit it by filling out the respective fields.

02/2007 –

- Add support for more JSDL elements.
- Add validation messaging for the UI (if possible).

02/2007 –

- Ability to populate / update information in the fields of the JSDL Editor by directly editing the raw JSDL file.
- Complete the JSDL implementation.

04/2007 –

- Provide an initial mechanism for translation of JDL to JSDL, and back.

3.4 Work Package 4 - Developer Perspective

3.4.1 Grid Command Console (glogin)

02/2007 –

- Working including two way connections (FW/NAT workaround)

06/2007 –

- Self staging (have to solve binary inclusion in package, check with GUP Licence)

06/2007 –

- Dynamic Setup of port forwarding

3.4.2 Grid Visualization

01/2007 –

- Client + Demo for g-Eclipse

06/2007 –

- Having a generic Visualization remote service based on VTK

09/2007 –

- Have a service for enabling safe X auth with remote sites

3.4.3 Development Tools

09/2007 –

- Wizards for Developing Grid Applications for specific middleware (depending on stable g-Eclipse API)

09/2007 –

- Wizards for developing g-Eclipse Extensions

09/2007 –

- Having both the above well documented

3.4.4 Application Debugger

04/2007 –

- Standard functionality (Break, Step etc.) including stage-in

xx/2007 –

- Parallel event analysis in ATEMPT style view (other middleware depends on support for grid command console there)

3.4.5 Application Monitoring

xx/2007 –

- Coarse status monitoring (based on R-GMA)

04/2007 –

- Fine monitoring grid back-end

05/2007 –

- Fine monitoring g-Eclipse plugin

3.4.6 Application Deployment

Input FZK

3.4.7 Grid Workflow Builder

03/2007 –

- Developing the model with EMF and prelim implementation with GEF

04/2007 –

- Drawing and saving workflows with g-Eclipse
- Associate jobs' JSDL files in the created workflow

05-06/2007 –

- Ability to open job (JSDL) editor from workflow editor
- Preliminary support for running workflows in the Grid (gLite)

4 Timeline

4.1 0.5M0 - 01/2007

- Design an initial JSDL editor (WP 3)
- Visualization client and demo up and running (WP 4)

4.2 0.5M1 - 02/2007

- Implementation of common actions (WP 1)
- Access to gLite CE monitor web-service information (WP 3)
- Read and write access to PBS and Maui services on a gLite CE (WP 3)
- Abstraction layer for a queue manager (WP 3)
- Webview bookmarks to various Grid monitoring and test sites (WP 3)
- First benchmarking support (WP 3)
- Support for more JSDL elements in the JSDL editor (WP 3)
- Add validation messaging to the JSDL editor (WP 3)
- Complete JSDL implementation (WP 3)
- gLogin: Two way connections (FW/NAT workaround) (WP 4)

4.3 0.5M2 - 03/2007

- Consolidation of new model (WP 1)
- Implementation of common actions (WP 1)
- Implementation of tar-support (WP 1)
- Implementation of Drag'n'Drop / Transfer Actions (WP 1)
- Job wizard: Switch to JSDL XML model (WP 2)
- Job wizard: Produce proper JSDL/JDL (WP 2)
- Reading job status from L&B (WP 2)
- Job viewer for browsing and filtering jobs (WP 2)
- Job viewer / editor for showing given job details (WP 2)
- More operations for remote gridFTP filesystems (WP 2)
- Support for EGEE SAM tests (WP 2)

- UI interface of the queue manager (WP 3)
- Define file format for queue manager configuration (WP 3)
- Result graphs for benchmarking (WP 3)
- Preliminary implementation of workflow builder with EMF and GEF (WP 4)

4.4 0.5M3 - 04/2007

- Consolidation of new model (WP 1)
- Cleanup of model (WP 1)
- Implementation of common actions (WP 1)
- Implementation of Drag'n'Drop / Transfer Actions (WP 1)
- Full implementation of VOMS Proxy (WP 1)
- Job wizard: New page for resource requirements (WP 2)
- Reading job status from L&B (WP 2)
- Job viewer for browsing and filtering jobs (WP 2)
- More operations for remote gridFTP filesystems (WP 2)
- Support for LFC/SRM (WP 2)
- Support for EGEE SAM tests (WP 2)
- Consolidate and integrate queue manager (WP 3)
- More benchmarks / graphs (WP 3)
- Start work on local job management (WP 3)
- Migrate jobs from one worker node to another (WP 3)
- Initial implementation of JDL-JSDL-Translator (WP 3)
- Debugging: Standard functionality including stage-in (WP 4)
- Fine application monitoring grid backend (WP 4)
- Drawing and saving workflows with g-Eclipse (WP4)
- Associate jsdl files with jobs in created workflow (WP 4)

4.5 0.5M4 - 05/2007

- Cleanup of model (WP 1)
- Provision of g-Eclipse problem reporting mechanism (WP 1)
- Implementation of common actions (WP 1)
- Full implementation of VOMS Proxy (WP 1)
- Make Grid Proxy independent of external jars (WP 1)
- Job wizard: New page for resource requirements (WP 2)
- Job wizard: New page for application defined parameters (WP 2)
- Job viewer for browsing and filtering jobs (WP 2)
- Support for LFC/SRM (WP 2)
- Extensions of information system (VO, queues, worker nodes ...) (WP 3)
- More benchmarks/graphs (WP 3)
- Fine application monitoring g-Eclipse plugin (WP 4)
- Ability to open job (JSDL) editor from workflow editor (WP4)
- Prelim support for running workflows in the Grid (gLite) (WP 4)

4.6 0.5 - 06/2007

- Provision of g-Eclipse problem reporting mechanism (WP 1)
- Implementation of common actions (WP 1)
- Make Grid Proxy independent of external jars (WP 1)
- Make VOMS Proxy independent of external jars (WP 1)
- Gather system information of remote sites (WP 3)
- Complete local job management (WP 3)
- gLogin: Self staging (WP 4)
- gLogin: Dynamic setup of port forwardings (WP 4)
- Generic visualization remote service based on vtk (WP 4)
- Running workflows in the Grid (gLite) (WP 4)

4.7 Planning Further

- Service for save X authentication with remote sites (WP 4)
- Wizards for Developing Grid Apps for specific MW (WP 4)
- Wizards for Developing g-Eclipse Extensions (WP 4)
- Parallel Event analysis in ATEMPT style view (WP 4)
- Coarse status monitoring (based on R-GMA) (WP 4)

4.8 Overview

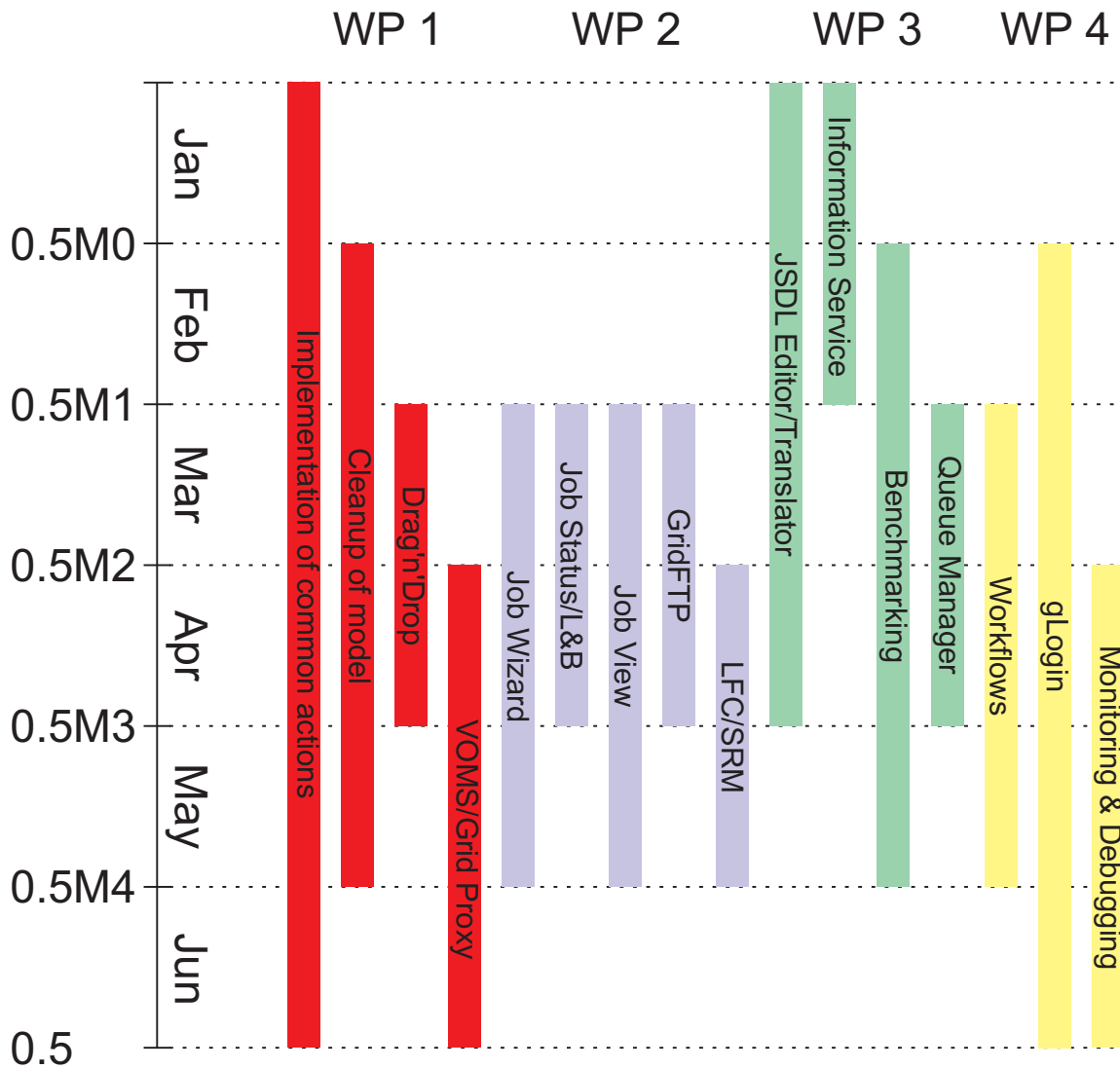


Figure 4.1: Rough overview of the features that will be implemented by the different workpackages for the first g-Eclipse release 0.5.