

Gatlet – A Grid Portal Framework

The aim of the Gatlet project (<http://gatlet.scc.kit.edu>) is to provide easy access to major grid middlewares and storage resources via web browser. Currently the framework supports access to Globus 4, gLite 3.1, GridFtp and SRM resources. With a portal built on this framework a grid user can submit and monitor jobs and manage files on storage resources. Grid infrastructures are mapped in the portal to entities like sites, clusters, corresponding middlewares and VOs. Gatlet is shipped with several portlets (e.g. for Job Submission, Monitoring, Resource Management). The Gatlet framework is Open Source and written completely in Java. The name Gatlet is a combination of GAT (Grid Application Toolkit) and Portlet. GAT provides access to several major grid middlewares (Globus Toolkit, Unicore and gLite) and storage resources (GridFtp, SRM) through an API. It's easy to extend GAT with middleware types or versions. GridSphere is used as portlet container for the portlets shipped with Gatlet and respectively custom portlets for scientific applications. GridSphere allows users to authenticate against the portal with X.509 certificates. Grid security issues are handled by Gatlet via MyProxy. The framework uses a database to store data for submitted jobs, grid resources and application data. In case of using JPA (Java Persistence API) all common vendors of relational databases are supported. The Gatlet service API enables access to this database and GAT. Portals built on top of the Gatlet framework address grid resource providers, grid users, portal providers and application developers. Users, especially grid newcomers, can use their well know web browsers to obtain access to grid resources. They need no deep knowledge of specific middleware clients, of grid security mechanism or of operating systems. Another important fact is that they do not need to install any software on their computers. Resource providers get a software to provide easy access to their grid infrastructure over the internet. The process of importing data from a resources database to the Gatlet database can be automatised easily. The portal administrators can assign installed software, hardware and VOs information to grid resources. This information is used by a Meta-Submitter that automatically chooses a matching resource. This eases the job submission for non grid experts. With the Gatlet Service API it's easy to develop portlets with access to grid resources. A developer of custom portlets programs against this API, which will integrate his portlets seamless into the portal. This gives a solid foundation on which to base a web interface to higher level applications installed on clusters. Middleware and storage resources are handled internally as abstract entities. The integration of new middleware and storage types does not require updating the existing job management or the file management. New technologies can therefore be seamlessly integrated into the portal. Future developments will include support for Unicore 6, Globus 5 and ARC resources. Another important goal is the integration of the Gatlet framework into LifeRay, a modern enterprise open source portal and collaboration software utilised by a big community. It's planned to provide an application repository where portlet developers can share their custom portlets with the communities.

Primary authors : Mr. BOZIC, Stefan (Karlsruhe Institute of Technology) ; Mr. BOEGEL, Bastian(University of Ulm)

Co-authors : Dr. ANTONI, Torsten (Karlsruhe Institute of Technology) Presenter : Mr. BOZIC, Stefan (Karlsruhe Institute of Technology)