

The Appea/GridSpace Application Development Platform

Motivation and Challenges

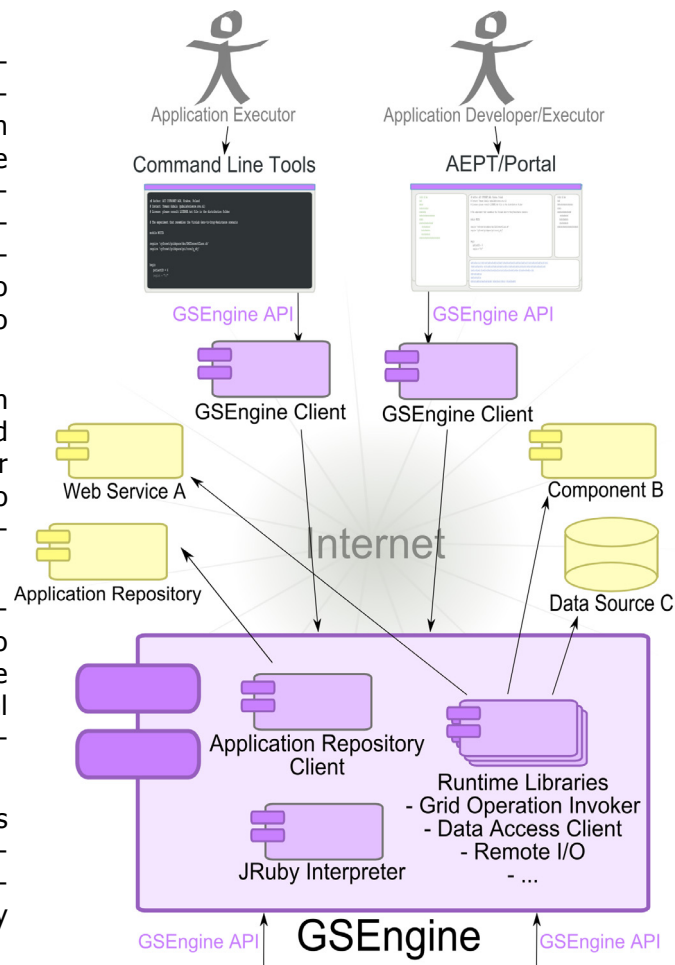
Modern business applications rely on widely distributed resources, both in terms of computation and data management. Owing to the vast array of middleware technologies required to interact with such resources, a need arises for an environment which would enable programmers to develop Grid applications in a seamless way, while still making use of the underlying computational and data repositories. Appea answers this need by delivering an application development platform which supports Grid application developers throughout the full development cycle – from deploying and registering individual Grid resources to managing application workflows.

Appea Solution Overview

Appea works by presenting Grid application developers with an environment where application workflows are expressed by a simple scripting notation based on a popular open-source scripting language (Ruby). All Grid resources are encapsulated as simple objects of the host language (so-called Grid Objects) and the functionality they provide can be invoked via methods of these objects. In order to support such an environment, Appea is divided into the following components:

- The Application Execution Planning Tool – an application development environment based on the popular Eclipse IDE, with support for the Grid Object registry, application scenario repository management and rapid Ruby programming.
- A backend runtime and execution environment called the GridSpace platform, used to execute the application scenarios, invoke the appropriate computational resources, marshal data and enact communication with end users.

While the Application Execution Planning Tool is aimed at application developers, the GridSpace runtime can also interact with dedicated application-oriented portals to support end users who typically have little to no knowledge on Grid computing.



Innovation

The Apnea Application Development Platform is a complete end-to-end solution for the development, enactment and use of business Grid applications. Rather than most existing virtual laboratory/workflow systems, it does not provide a drag-and-drop interface, but instead utilizes a scripting notation for expressing workflows. Ruby has been chosen due to its expressiveness, object-oriented features, good community support, availability of external libraries, seamless integration with legacy Java applications (through the JRuby project) and clear, coherent, obfuscation-free syntax. Thus, Apnea enables application developers to schedule application logic which would be extremely difficult to express using a graph-based workflow notation. Small data processing tasks can be accomplished on the fly and it becomes trivial to utilize existing application code, as well as any Java- or Ruby-based libraries. Moreover, Apnea also provides a remote runtime which can take care of processing complicated workflows on behalf of the user, without placing undue strain on their personal computers, and which can be contacted from anywhere using Web-based user portals.



Business Impact

Apnea is geared towards meeting end-user functional requirements. The use of a scripting notation and full support for interactivity means that the application developer is unconstrained with respect to the types of end-user interaction that can be initiated and carried out by the system. Apnea supports the XForms framework, enabling user portals to display properly formatted interface forms while still enabling developers to control all aspects of user interaction. Moreover, Apnea – together with the FiVO component also developed within the GREDIA project – delivers comprehensive user management facilities, on two levels of granularity (VO- and user-based). A GSI-based security system is integrated, providing Single Sign-On capabilities both for application developers and users who actually interact with the system.

Interoperability

Our Apnea draws upon many advances in modern computational environment design and Grid system engineering. The scripting language used to develop application scenarios is a superset of Ruby and any Ruby script is a valid application scenario within Apnea. Moreover, legacy Java libraries can be easily integrated and invoked within Apnea scenarios, without the need to encapsulate them as services. Apnea can communicate with various types of computational resources and data access platforms, including:



- **Computation:** Web Services, Globus WSRF (stateful Grid Services), MOCCA component frameworks, gLite-based batch job submission systems (including modern European production Grid infrastructures such as EGEE/EGI)
- **Data access:** Relational data sources (popular database management systems, both commercial and open-source, OGSA-DAI database aggregations), flat-file systems (DAV repositories, the GREDIA distributed data access framework etc.)

The Application Execution Planning Tool itself is based on the Eclipse framework and can support Eclipse plugins, while at the same time providing developers with all the versatility of a modern IDE. We furthermore use the GSI security framework, with automatic management of user certificates and delegation of credentials, thus providing users with a Single Sign-On security solution.

Partners Involved



Contact

**Academic Computing Center CYFRONET
AGH**

<http://www.cyfronet.pl>

Piotr Nowakowski