

# UNICORE Roadmap: towards UNICORE 6

Defining the transition from Legacy Unicore to a Web services based UNICORE, approximately covering the period from mid 2006 to end 2007

Prepared by the Technical Advisory Board of the Unicore Forum

Bernd Schuller (FZ Juelich), Chair  
Donal K. Fellows (University of Manchester), Deputy  
Sven van den Berghe (Fujitsu Labs Europe)  
Michel Drescher (Fujitsu Labs Europe)  
Bernhard Dierberger (IBM Germany)  
Roger Menday (FZ Juelich)  
Gert Ohme (T-Systems SfR)  
Ralf Ratering (Intel GmbH)

(Version 0.4, last edited on September 8, 2006)

## 1. Introduction

In the course of the last two years, the European project UniGrids has developed a new Grid middleware, providing an open, service oriented architecture. Since this is completely new software, there is no backwards compatibility to the legacy UNICORE that is based on the AJO Java class library. Therefore, the need arises to facilitate the transition to the new implementation, while at the same time protecting the UNICORE brand.

Furthermore, external stakeholders (such as user communities, companies offering commercial support, national and international funding agencies) need a clear picture of how UNICORE is going to evolve, and what effort they will have to invest to go along with this evolution.

Accordingly, this document provides a roadmap towards the next major UNICORE release.

## 2. UNICORE characteristics

The UNICORE middleware has a set of characteristics that distinguish it from other Grid middleware systems such as Globus. It is desirable to bring forward these characteristics to the new release, in order to protect the UNICORE brand and “market message”.

1. UNICORE as a vertically integrated Grid middleware  
Though UNICORE is a set of loosely coupled componts, there is a bundled version available, containing core components, that is easy to install and configure. This includes a “standard“ client (see 3)
2. Strong end-to-end security available  
The UNICORE security measures include the option for strong, end-to-end security. However, in contrast to UNICORE 5.x, this can be configured more flexibly, to allow delegation of trust, secure intermediaries, etc, as outlined by the Unicore security architecture doc (forthcoming)
3. Graphical end user client  
A GUI client exists for the common use cases (see 1.). This covers the traditional UNICORE use cases (manual resource selection, multistep, multisite workflows in the fashion of UNICORE 5)

4. Multi-step, multi-site workflow support  
Users can select resources and sites manually, and build multi-step multi-site jobs.
5. Good support for integrating applications (without having to modify them)  
Abstract application support, plugin concept on the client, ...
6. Respect site autonomy (security policies, local permissions, etc.)  
Sites can expose resources on the Grid, but may keep fine-grained control over who accesses their resources.
7. Ease of installation and configuration  
UNICORE is easy to install, with minimal external dependencies. Configuration is done in a small set of configuration files.
8. Ease of administration  
UNICORE is simple to administrate and monitor.

### 3. User-level feature groups of UNICORE 6

Ideally, the final version of the next major release should offer the user at least the features of the current UNICORE 5.x. These features can be grouped in several areas

1. Atomic: simple tasks, such as „Execute script“, „Import a file“ or „Access database“
2. Workflow: Build multi-site workflows from atomic tasks. Edit, run and monitor these.
3. Client: workstation GUI client with the following features
  - Allows to prepare, submit and monitor jobs;
  - Allows to view outcome;
  - Has a plug-in mechanism for application specific interfaces and functional extensions.

Additionally, there are several new feature groups

1. Portal: web based portal client
2. Streaming: client-server streaming support (for visualisation or media applications)
3. Management Tools targetted at system administrators
  - tools to manage and administrate services
  - tools to manage users and virtual organisations (VOs),
  - tools to manage software licenses
  - simplify application deployment

### 4. Development Drivers

After the end of UniGrids, UNICORE will continue to be used and further developed in further EC funded projects. The following table gives an overview about upcoming projects, and how they will contribute to UNICORE.

<i>Initiave</i>	<i>Timeframe</i>	<i>Contributions to</i>
A-Ware	2006-2008	<ul style="list-style-type: none"> <li>• Atomic</li> <li>• Workflow</li> <li>• Portal</li> </ul>

<i>Initiative</i>	<i>Timeframe</i>	<i>Contributions to</i>
Chemomentum	2006-2008	<ul style="list-style-type: none"> <li>• Atomic (Data access, Applications)</li> <li>• Workflow</li> <li>• Client</li> <li>• Portal</li> <li>• Management Tools</li> </ul>
OMII-Europe	2006-2008	<ul style="list-style-type: none"> <li>• Atomic (Data access, Job submission)</li> <li>• Management tools (VO)</li> <li>• Portal</li> </ul>
Intel GPE plans	See section 5.2	

## 5. Roadmap

From the development drivers listed in the previous section, the following roadmap can be derived.

<i>Release</i>	<i>Approx. Date</i>	<i>Core Features (level)</i>	<i>Additional features (level)</i>
alpha	July 2006	<ul style="list-style-type: none"> <li>• Atomic (alpha)</li> <li>• Client (alpha)</li> <li>• Workflow prototype</li> </ul>	<ul style="list-style-type: none"> <li>• Streaming (beta)</li> <li>• Management tools (alpha)</li> </ul>
beta	Mid 2007	<ul style="list-style-type: none"> <li>• Atomic (final)</li> <li>• Client (beta)</li> <li>• Workflow (beta)</li> </ul>	<ul style="list-style-type: none"> <li>• Portal (alpha)</li> <li>• Streaming (beta)</li> <li>• Management tools (beta)</li> </ul>
Final	End 2007	<ul style="list-style-type: none"> <li>• Atomic (final)</li> <li>• Workflow (final)</li> <li>• Client (final)</li> <li>• Streaming (final)</li> </ul>	<ul style="list-style-type: none"> <li>• Portal (final)</li> <li>• Streaming (final)</li> <li>• Management tools (final)</li> </ul>

## 6. Issues and Comments

The following issues have come up during the discussions of the Technical Advisory Board

### 6.1. *More detailed roadmap*

The roadmap outlined in the previous chapter should be much more detailed. The following information needs to be added

For each component or feature set, the underlying specifications (e.g. WSDL files, XML schema files, component interfaces, etc) need to be frozen at some point in time. This should correspond to a “milestone” in the detailed roadmap

### 6.2. *Security architecture*

Currently (May 2006) the security architecture and status of implementation in UniGrids (UNICORE 6 alpha) is not yet very well defined. This represents a risk for the roadmap and release schedule as outlined above.

### 6.3. *Current and future role of GPE*

Currently, the GPE clients (“Expert client” and “Application client”) can be used to access UNICORE 6 services. By mid 2006, prototype support for workflows based on BPEL will be available for Unicore/GS, provided that the necessary security infrastructure is in place.

As GPE is developed separately, it is unclear whether future GPE developments (i.e. those after the end of the UniGrids project) will be compatible with UNICORE. Much will depend on a joint development process and a well-defined evolution of the underlying interfaces.

### 6.4. *Test-suite, compatibility checks, compliance tests*

As UNICORE 6 is based on much more open set of interfaces and communication protocols, it is likely that components will appear that use various programming languages, SOAP and web service toolkits, and so on. It would be highly desirable to have a set of functional tests that can be used to check whether a given implementation complies to the UNICORE 6 interfaces. For each component or feature group, multiple implementations may exist (e.g. FLE’s Unicore/GS, Intel’s GPE4GTK, FZJs UAS as implementations of the UNICORE 6 atomic services).

### 6.5. *Development process and release management*

The UNICORE 6 developments will use input from multiple sources such as various projects, individual contributions such as Intel’s GPE effort, and possibly bugfixes and patches from external developers. This will make a better defined development and release process necessary.

### 6.6. *UNICORE 5 “end of life”*

As expressed by the core developers and maintainers of UNICORE 5, future support for UNICORE 5 will be limited to critical bugfixes. New features will not be integrated due to lack of resources. However, the core developers might provide some guidance for third parties who want to work on UNICORE 5.