Links Multimedia Points of view

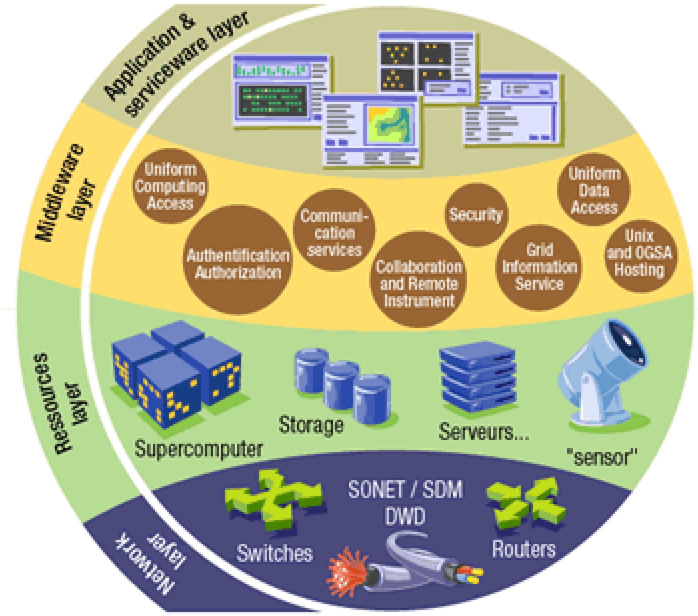
What is the Grid? | How it works | What it can do | A brief history | The Grid and you | Grid @ CERN | Grid projec

How does it work?

Grid architecture

The five big ideas
Building blocks
Grid architecture
Underlying hardware
Middleware
Globus toolkit
Testbeds

The architecture of the Grid is often described in terms of "**layers**", each providing a spe function. In general, the higher layers are focussed on the user (user-centric, in the jarg whereas the lower layers are more focussed on computers and networks (hardware-cer



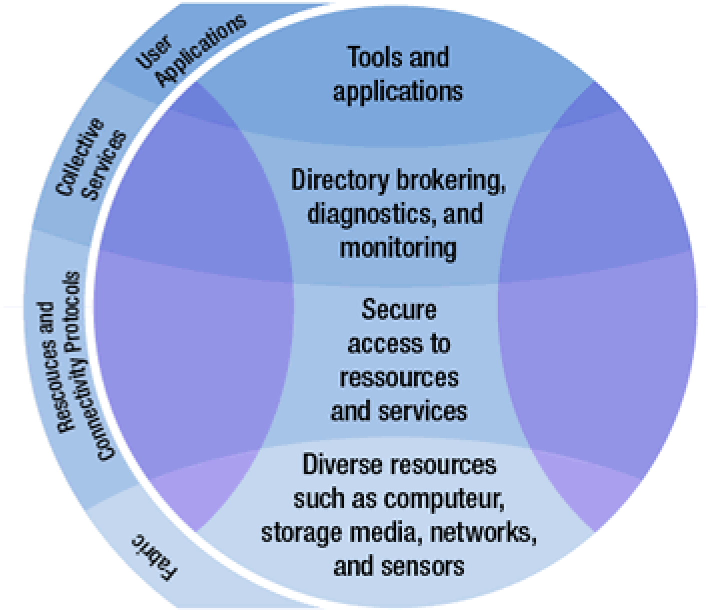
At the base of everything, the bottom layer is the **network**, which assures the connectiv the resources in the Grid. On top of it lies the **resource layer**, made up of the actual res that are part of the Grid, such as computers, storage systems, electronic data catalogue even sensors such as telescopes or other instruments, which can be connected directly network.

The **middleware layer** provides the tools that enable the various elements (servers, sto networks, etc.) to participate in a unified Grid environment. The middleware layer can b thought of as the intelligence that brings the various elements together - **the "brain" of Grid!**

The highest layer of the structure is the **application layer**, which includes all different u applications (science, engineering, business, financial), portals and development toolkit supporting the applications. This is the layer that users of the grid will "see".

In most common Grid architectures, the application layer also provides the so-called **serviceware**, the sort of general management functions such as measuring the amoun particular user employs the Grid, billing for this use (assuming a commercial model), an generally keeping accounts of who is providing resources and who is using them - an in activity when sharing the resources of a variety of institutions amongst large numbers o different users. *(The serviceware is in the top layer, because it is something the user ac* *interacts with, whereas the middleware is a "hidden" layer that the user should not have worry about.)*

There are **other ways to describe this layered structure** . For example, experts like tc term **fabric** for all the physical infrastructure of the Grid, including computers and the communication network. Within the middleware layer, distinctions can be made betweer of resource and connectivity protocols, and a higher layer of collective services.


User Applications
Collective Services
Rescouces and Connectivity Protocols
Fabric
Tools and applications
Directory brokering, diagnostics, and monitoring
Secure access to ressources and services
Diverse resources such as computeur, storage media, networks, and sensors

Resource and **connectivity protocols** handle all "Grid specific" network transactions b different computers and other resources on the Grid. Remember that the network used Grid is the Internet, the same network used by the Web and by many other services suc mail. A myriad of transactions is going on at any instant on the Internet, and computers actively contributing to the Grid have to be able to recognize those messages that are re to them, and filter out the rest. This is done with **communication protocols**, which let t resources speak to each other, enabling exchange of data, and **authentication protoc** which provide secure mechanisms for verifying the identity of both users and resources

The **collective services** are also based on protocols: **information protocols**, which ol information about the structure and state of the resources on the Grid, and **manageme protocols** which negotiate access to resources in a uniform way. The services include:

- keeping directories of available resources updated at all times,
- brokering resources (which like stock broking, is about negotiating between thos want to "buy" resources and those who want to "sell")
- monitoring and diagnosing problems on the Grid
- replicating key data so that multiple copies are available at different locations foı use
- providing membership/policy services for keeping track on the Grid of who is all do what, when.

In all schemes, the **topmost layer** is the **applications** layer. Applications rely on all the layers below them in order to run on the Grid. To take a fairly concrete example, consid application that needs to analyze data contained in several independent files. It will hav

- obtain the necessary **authentication credentials** to open the files (resource an connectivity protocols)
- query an **information system** and **replica catalogue** to determine where copie files in question can currently be found on the Grid, as well as where computatic resources to do the data analysis are most conveniently located (collective servi
- submit requests to the **fabric** - the appropriate computers, storage systems, anc

networks - to extract the data, initiate computations, and provide the results (res and connectivity protocols)

- **monitor** the progress of the various computations and data transfers, notifying t when the analysis is complete, and detecting and responding to failure condition (collective services).

In order to do all of the above, it is clear that an **application** that a user may have writte on a stand-alone PC **will have to be adapted** in order to invoke all the right services ar all the right protocols. Just like the "webifying" of applications - where users have to ada stand-alone application to run on a web browser, so too the Grid will require users to inv some effort into "**gridifying**" their applications. So there is no free lunch, not even on th

However, once gridified, thousands of people will be able to use the same application a trouble-free on the Grid (well, ok, like most software, there will always be a few bugs he there), using the middleware layers to adapt in a seamless way to the changing circums of the fabric.

Home | About I Search I Site Map I Contac