Notes on the Physiology Paper by Ian Foster et al:
http://www.globus.org/research/papers/ogsa.pdf

1. Globus Toolkit provides the framework for grid-based applications. Gloubus toolkit (latest version is known as GT3) is a community-based, open architecture, open source set of services and software libraries
that support Grid and Grid Appilcations.

2. Toolkit addresses the issues of secuirty, information discovery, resource management, data management, communication,
fault detection, and portability. (GRAM, MDS-2, GSI, etc.)

3. Webservices will serve as foundational entity for grid services. (See SOAP, WSDL, WS-inspection, WSFL orchestration
etc.)

4. OGSA; Open Grid Services Architecture

    1) Computing is increasingly concerned with the creation, management, and application of dynamic ensembles of resources,
    and services and people.

    2) OGSA supports creation, maintenance and application of ensembles of services.

    3) Object-orientation to service-orientation

    4) A service is a network-enables entity that provides some capability: a sophiticated object, a standardized object

    5) OGSA: Computational resources, storage resources, networks, programs, databases, and the like are all represented
     as services.

    6) Interoperabilty is a critical problem: handled by dividing it into two issues:
        a. Definition of service interface
        b. protocols to invoke service interface

    7) Virtualization through WSDL
        WSDL allows multiple bindings of the same interface and optimized binding for local access.
        Central to virtualization is the ability to adapt OS functions at a specific site, explotation of native capability.
        Virtualization allows the composition of services to form more sophisticated services.

    8) Ability to virtualize and compose services depends on more than the standard interface definition.
        We also require standard semantics for service interaction: for example error notification, lifecycle management.
        This need is addressed by a well-defined set of interfaces called Grid Service.

    9) Grid Service: A web service with multiple interfaces: address discovery, dynamic service creation, lifetime managment,

notification, and mangeability.

10) Transient services and a static set of persistent services. Transient service to query againt a database, a datamining operation.
Persistent services: security, naming..

11) Standard interfaces: in WSDL terms these are portTypes.
portType: GridService, NotificationSource, NotificationSink, Registry, Factory, HandleMap
Read the operations in the portType and the respective semnatics.

12) Discovery:
Problem: Applications require mechanisms for discovering available services and for determining the characteristics of the services
to configure themselves and their request.
Solution:
-- a standard representation of service data: information about Grid Service instances which are structured as a set of named
and types XML elements called service data elements.
-- A standard operation FindServiceData (pull) NotificationSource (push)
-- standard interface for registering information about Grid Service instances with Registry services

13) Dynamic service creation

14) Lifetime management

15) Notification: A collection of dynamic, distributed servcies must be able to notify each other asynchronously of interesting changes
to their state. NotificationSource, NotificationSink

16) OGSA defines the semantics of a Grid service intsance: How it is created, how it is named, how its lifetime is determined, how to communicate
with it.
17) Hosting environment addresses the programming model, programming language, implementation tools, and execution environment. It also defines
various development and debuggng tools.

18) Host environment:
-- Interface to the host environment consists of a registry, one or more factories, and a handlemap service.
-- Each factory is registered in the registry, to enable clients to discover available factroies.
-- when a factory receives a client request to create a Grid Servcie instance, the factory creates a new instance, assigns it a handle, registers the instance
with the registry, maps the handle available to the handleMap service. HandleMap maps handles to references, which may be local or remote.

19) Container/component (J2EE, .NET, WebSphere, Sun One) hosting environments offer superior programmability, manageability , flexibilty and safety.
These are preferred over native hosting environments.

20) Building Virtual Organization Structures:

-- Simple hosting environment: located within a single administrative domain.
    ---- contians one regsitry, several factories, and a HandleMap
    ---- each factory is registered in the Registry to allow client clients to discover the factories.
    ---- client discovers a Factory using the Registry and then requests an instance of a
service..... similar what we discussed in item 18)
    ---- only difference is that HandlMap maps the handles to local references.
-- Virtual Hosting Environment:
    ---- resources associated with a VO span heterogeneous, geographically distributed hosting
environments.
    ---- however the user inetrface is same.
    ---- There is highe level Registry which knows about lower level registries and higher level
factories that delegate creation requests to lower level
        factories, clients interact directly with service instances.
-- Collective services
    ---- offer sophisticated virtual, collective, end-to-end services.
    ---- muliple lower level servcies are created and composed into a single higher level service,
which is exposed to the client.

21) Application example: p.17 and 18 (Figure 3) Lets go through the details.

22) OGSA Technical details

1. Service model: computation resources, storage resources, networks, programs, databases etc.
are all services.
    This adoption of a uniform service model means that all components of the environment are
virtual.

2. Everything is a Grid Service: a Web service that conforms to a set of conventions and
supports standard interfaces for such
    purposes as lifetime management and security.

3. WS: portType --> WSDL
    GS: serviceType --> extensible WSDL (GWSDL) portType+ additional information ..
versioning etc.

4. Protocol binding associced with the service can define delivery semantics, for example
addressing reliability.

5. **Grid Service** is dynamic (can be created and destroyed dynamically) and stateful (state
corresponds to the instance).

6. Globally unique name for a Grid service instance is **Grid Service Handle** (GSH). GSH has
no protocol or instance specific
    information.

7. GSR: **Grid Service Reference** is a instance specific information, GSR for Grid service may
change over its lifetime. Has an explicit
    expiration time.

8. Creating **transient services**: Corresonding a transient service there is a factory service that
implements a **Factory** interface.

The Factory interface's CreateService operation creates a requested Grid service and returns the GSH and inital GSR for the
new service instance.

9. Lifetime Management

10. Managing Handles and References: (i) identifying the Handlemap service (ii) contacting the handleMap to obtain the desired GSR.

11. Service Data and Service Discovery:
-- associated with Grid service is a set of service data, a collection of XML elements. These are service data elements. A service data
element includes details such as a type, time to live, etc.
--- FindServiceData can eb used to query and get the serviceData.
-- other service data elements include GSH, GSR, primary key, home handleMap.

12. Notification: notification framework allows clients to register interest in being notified of particular messages and supports
asynchronous, one way delivery of such notifications.

13. change management

23) Protocol bindings: reliable transport etc.

24) High level services: GRAM, GridFtp etc.