

A bootstrapping scenario for eliciting CSCL services within a GRID virtual community

Pascal DUGENIE, Philippe LEMOISSON
LIRMM, CNRS & Université Montpellier II
161, Rue Ada
34392 Montpellier Cedex 5,
France
dugenie@lirmm.fr, lemoisson@lirmm.fr

Amongst the various eLearning techniques, Computer Supported Collaborative Learning (CSCL) is of growing interest within the academic world. However, during the first phase of the EleGI project, a noticeable fact has been the difficulty to match the user needs with the potentiality of the GRID services.

Thus, through a fictive scenario, this article proposes to walk the path between the idea of creating a new virtual community and the realisation of this objective.

This scenario takes the situation of a famous scientist wishing to set a Virtual Centre of Excellence (VCE) and put the emphasis on the bootstrap (i.e. the really first steps). An interesting comparison is to analyse this scenario firstly in a context using traditional techniques (system oriented architecture and Web services), then secondly using a service oriented architecture offered by the Grid services.

Our aim is to match this bootstrapping scenario with these contrasted contexts.

1. INTRODUCTION

1.1 Motivations

Looking back at the evolution of the internet technologies for the last two decades, there is a clear evidence of their strong impact on the way people interact and develop intellectual work.

Today, even formal methods of learning are directly impacted by the evolution of tools such as the web, emails or instant messaging.

In a technology pushed situation like this one, it is necessary to constantly explore innovative solutions by introducing more of the human dimension in the process.

Unfortunately, the evidence is that there is still a lack of adequation between the existing tools and the methods of collaboration for non-computer literate people.

This might be one of the reasons that explain the increasing interest in Computer Supported Collaborative Learning (CSCL) within the academic world [CSCL].

1.2 CSCL within the EleGI project

CSCL is a broadly investigated area within the EleGI project.

One of the EleGI objectives of particular interest in the present article, is the definition of a Grid based architecture for Collaborative Learning.

This is the chance for Grid as a newly introduced technology for CSCL to be investigated in the angle of human centered approach. The EleGI project proposes such kind of orientations with the called Open Grid Human Service Architecture (OGHSA) [GOU03] a derivative of the Open Grid Service Architecture (OGSA) specifications [FOS02].

At this stage, the project has identified some characteristics of CSCL, in particular within Virtual Communities (VC) [D13].

VC has been introduced in 1993 [RH93] and was defined as follows: *Virtual communities can be conceptualized as social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace.*

Collaborative learning differs from individual learning in the way that it enhances the social aspects involved in the learning process and brings positive results such as deeper understanding of content, increased overall achievement, improved self-esteem, and improved team work via group conflicts resolution.

CSCL is a solution to boost group activities by eliminating the space and time constraints. Group of learners can be dispersed in various geographical areas thanks to new technologies (i.e. networks and computers). This also adds an exciting dimension to CSCL. In addition, time is not a constraint anymore, as long as instant interaction is not needed. Members may attend to learning activities at any time, hence dismissing the need of co-presence. These anywhere-anytime characteristics enable a shift from real time group learning to asynchronous distributed learning activities, and justifies the need of VC management.

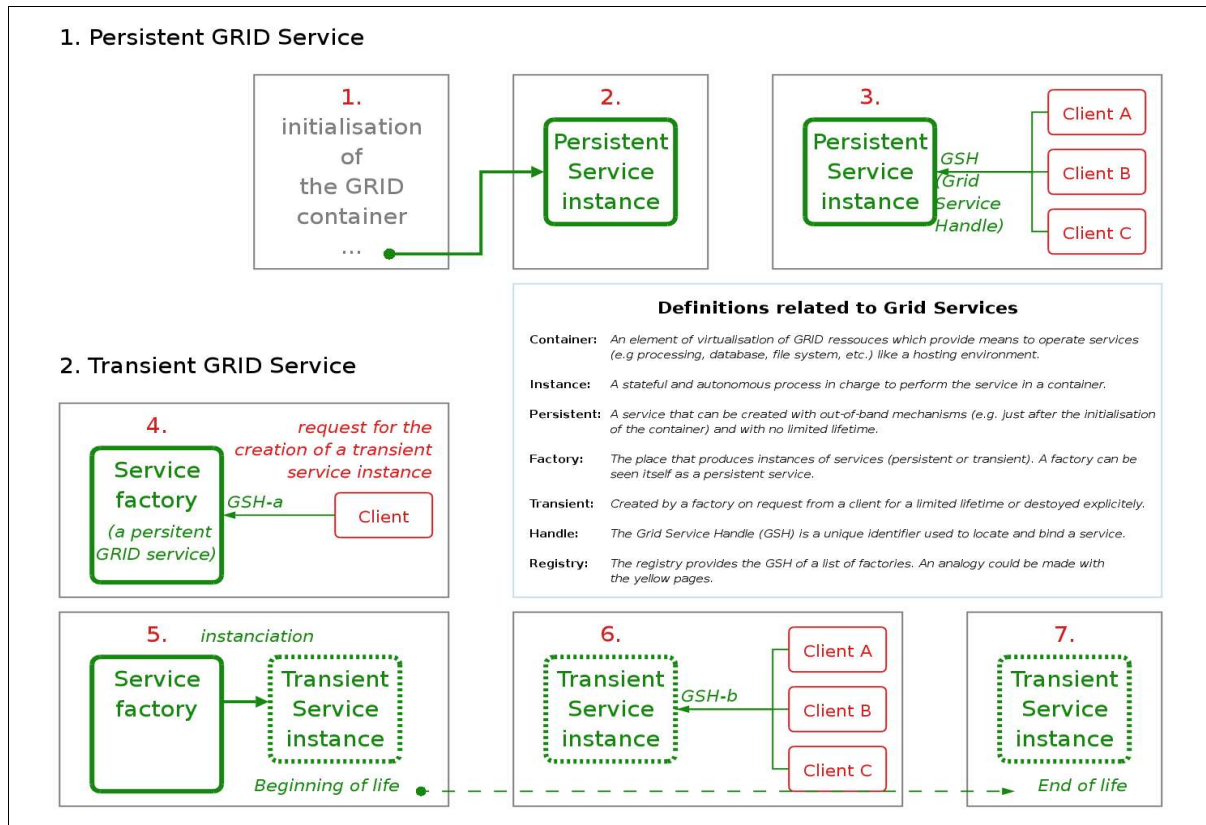


Figure 1: A persistent Grid service (2) started by an out-of-band mechanism like the initialisation of a GS container (1). A GS handle is used by the clients to access to the service (3). A service factory is a particular persistent service (4). The factory is invoked by clients to launch service instances (5). Clients may access to the service during its lifetime (6). Transient service end-of-life is determined implicitly or by a soft-state (7).

In the context of Grid, we propose the definition of a VC as a collection human or artificial agents playing a variety of roles.

1.3 Eliciting Grid Services for CSCL

In order to determine how Grid Services may improve CSCL techniques, this article will first sketch out the major characteristics of Grid, and then proposes to walk, through a fictive scenario, the path from the idea of creating a new VC until the concretisation of this objective.

2. TECHNOLOGICAL BACKGROUND

Let us now review briefly the migration from usual collaborative tools (web, email) to GS, in order to understand the novative aspects of the Grid for CSCL.

2.1 Grid at glance

A complete state of the art of Grid is out of the scope of the present paper, however we firstly must remind our reader the definition of some of the commonly used terminology as specified in the OGSA specifications [FOS02].

Web Service (WS) define methods for accessing software components, and enable the identification of relevant service providers.

Grid Service (GS) is an extension of WS that support service state. The extended capabilities of GS include transient services and service notification.

X.509 certificate is a standard used for the mutual authentication mechanism of the Grid security infrastructure.

Figure 1, explains how transient services are made possible in GS with the use of the factory instance model.

The factory is the place that produces instances of services (persistent or transient). A factory can be seen itself as a persistent service. An instance is a stateful and autonomous process in charge to perform the service in a container, that is is an element of virtualisation of GRID resources which provide means to operate services (e.g processing, database, file system, etc.) like a hosting environment.

A persistent service can be created with out-of-band mechanisms (e.g. just after the initialisation of the container) and with no limited lifetime, whereas a transient service is Created by a factory on request from a client for a limited lifetime or destroyed explicitly.

GS are retrieved with their handle, the GSH. A registry provides the GSH of a list of factories. An analogy could be made with the yellow pages.

2.2 From System- Oriented to Service- Oriented Architectures

The Client- Server model is the most popular System-Oriented architecture.

As described on figure 2, in this architecture the client invoke services from a server via a protocol. In this example the server is an autonomous host with its own IP address or a fixed domain names (host.domain).

An http service can be provided by this server to a group of clients. When accessing to these services, the clients will have to connect to this particular server.

State can be managed for collaborative services in a system-oriented architecture using ad-hoc protocols. An example is the Jabber protocol which widely used in instant messaging tools[EKD03].

However, since all these services are strongly linked with the physical address of this server, the system-oriented architecture gives no chance for service ubiquity.; in our understanding, service ubiquity is implemented when the service function is fully decoupled from the physical resources allocated to perform this service.

For this reason the Service-Oriented Architecture (SOA) is based on the principle of service virtualisation.

2.3 From Web Services (WS) to Grid Services (GS)

Comparative argumentations between Web Services (WS) and Grid Services (GS) are widely available in the literature [SOT03], [JAC03].

Initially, WS appeared to solve the problem of the complexity growth resulting from protocols heterogeneity and virtualisation of services.

However, WS address discovery and invocation of persistent service and do not support service state.

The introduction of service state in the GS has permitted to extend the service capabilities. Among other advantages, GS support transient services that can be created and destroyed dynamically.

This lead to many implications for how services are managed, named, discovered and used. In fact, much of Grid is concerned with this.

The table below summarises the services versus the architectures explained above. This show the core advantages of a Grid approach.

<i>Architectures >> Service</i>	<i>System- oriented</i>	<i>Web Service</i>	<i>Grid Service</i>
Service ubiquity	No	Yes	Yes
Service state	Possible(1)	No(2)	Yes

(1) ad hoc solutions

(2) due to intrinsic limitations of HTTP protocol

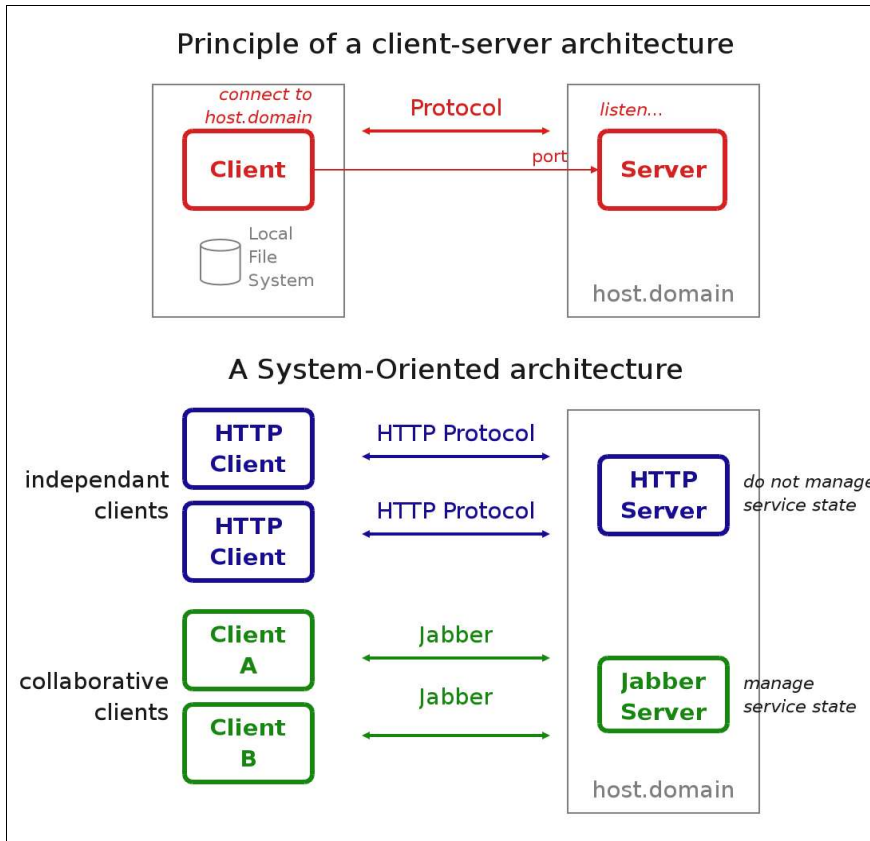


Figure 2: A system oriented-architecture like client-server can manage service state depending on the protocol used. In one hand, the http protocol does not manage state, therefore the clients are independants. In another hand an ad hoc protocol like Jabber can manage state, therefore a client can use a collaborative service such as instant messaging.

2.4 Semantic Grid

Collaborative activities involving concurrency, sharing and conferencing would not be possible without the management of service state and the transient capabilities offered by GS.

An original attempt to define a framework to integrate eLearning methods within GS, is presented in A Future e-Science Infrastructure, by David deRoure[ROU01].

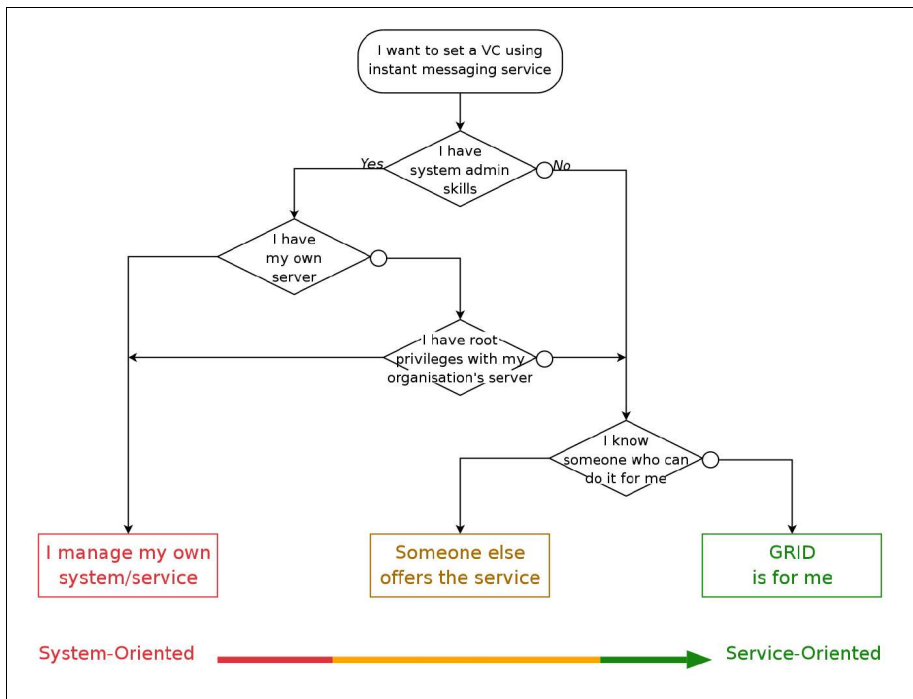


Figure 3: The logical path from the initial wish I want to set a VC towards the technical choice to realise this objective.

A fictive scenario presents a situation ex-nihilo and put the emphasis on the bootstrap (i.e. the really first steps). An interesting comparison is to analyse this scenario firstly in a context using traditional techniques (system oriented architecture and Web Services), then secondly using a service oriented architecture offered by the Grid Services.

3. THE SCENARIO

3.1 Description of the scenario: Prof. Marcus and his Virtual Centre of Excellence (VCE)

Let us consider a typical scenario inspired from the situation of one of the five EleGI's Service Elicitation and Exploration Scenarios called ENCORE. This scenario aims to gather a community of chemists who intend to construct an encyclopedia of organic chemistry.

Obviously, Prof. Marcus wish to gather most of the world famous experts in organic chemistry.

Today, he may already have a network of known people that can be contacted by email.

Alternatively, he could type some keywords in a search engine, to extend its own list of potential candidates and eventually contact them. Their requirement will not be detailed here. Instead, the scenario of this paper is a simplification of this, in order to focus on the bootstrap.

Suppose that a famous scientist, Prof. Marcus, wishes to set a Virtual Centre of Excellence (VCE) in organic chemistry. In fact, a typical path of questions is described in figure 3.

Figure 4, illustrate that amongs the three different possibilities A, B and C, the only one that offers autonomy, service state and service ubiquity is the SOA approach using GS.

Prof. Marcus has some rudiment in using usual internet tools, like email and web browser, but he has no intention, nor time to acquire IT (information Technology) expertise.

Lets imagine now that the VCE is set and that these experts agree to publish a review of all their research in an electronic form. They aim to publish in the web but this could be a completely different medium (CD ROM, etc.).

3.2 Prof. Marcus using traditional techniques

Prof. Marcus has in first place to deal with VC Management : he has to imagine the rules for the life of the community, like, for example: "rule 1: in order to extend the circle of relation, every known person can welcome new members in the community of experts". And in the same time, he has to manage the evolving group, for instance through a mailing list.

Then, every expert needs a secure environment to provide contribution. The security must include here mechanisms for reading and writing access, such as authentication, privacy and integrity.

Prof Marcus has to choose/develop a tool for authentication and to manage the diffusion of the proper rights to the proper people ; which roughly corresponds to the skill needed for being a WebMaster.

In a third place, when the working space has been set, an even more tedious task is tidying-up ideas coming from different sources.

As ideas will obviously evolve and some may diverge whereas other will converge, here comes the necessity for consensus. A natural way to see it is to consider the initiator of this community to act as a mediator who will arbitrate when positions are too contradictive. In the case of scientific collaboration, this conversational process has to happen in a synchronous mode, therefore Prof. Marcus will have to choose and manage an Instant Messaging system, and to broadcast the instructions for downloading it and subscribing to it to the evolving VCE.

In a fourth place, Prof. Marcus will have to settle for his community a consensual space where the synthesis that has been agreed upon and that can later be used as a reference can be stored in the form of stabilised documents. Once again, Prof. Marcus will have to choose an editing tool, and to manage it (broadcast the instructions for downloading the software/subscribing to the tool, care about protection/publishing of the content, manage the collaborative edition of documents).

In a fifth place, the need of graphical communication might appear.

3.3 Prof. Marcus in a GRID bootstrapping scenario

Figure 5 gives a global overview of the early stages Prof. Marcus has to go through in order to start his VCE.

To start with, Prof. Marcus can simply use a web browser, assuming he knows the URL of a GS marketplace (1). From a list of services, A, B, C ... he chooses the service B dedicated to provide X.509 certificates and which does not require itself a certificate (2).

Service B is dedicated to provide X.509 certificates. From the service B factory location he may then request a certificate (3).

Then he goes back to the GS marketplace to select another service (4). Service C is dedicated to provide a GS container, an environment in which Prof. Marcus may run his own instances of GS. Here, he has to authenticate himself by showing his certificate. Service C returns a unique identifier GS Handle (5) called GSH_1.

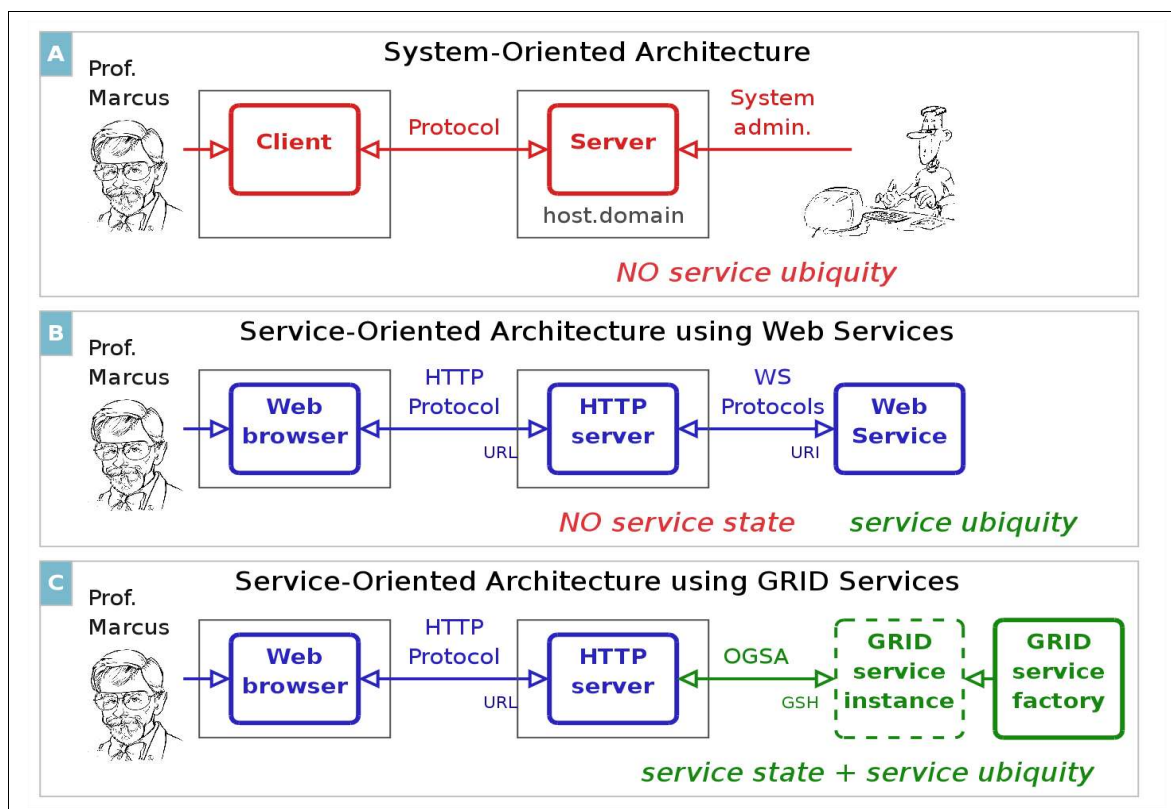


Figure 4: Comparison of three architectural models. A. The client-server can manage stateful services but not the service ubiquity, since the service is strongly linked with a particular host. B. The architecture using Web service can manage the ubiquity by allowing Prof. Marcus to use a web browser from any place to access to the service but the state cannot be managed since HTTP does not support it. C. The GS architecture offers both ubiquity with the use of a web browser and service state with the factory-instance model. A service instance can manage its own context, therefore several clients can access to this instance without messing each other up. this reason the Service-Oriented Architecture (SOA) is based on the principle of service virtualisation.

With GSH_1, Prof. Marcus may locate his newly created service instance (6). A Grid Service contained is created. A container consists of Grid resources reservations to perform other GS.

At this stage he may select other services within the registry that he will find useful for the VC (7).

He will select Service A, that is an instant messaging service and request a new instance of this service (8). Here he will provide the handle of his container, GSH_1, then the service factory A will create a new instance of service A in the container with an handle GSH_2. Prof. Marcus may now select his running Instant messaging service from his service container (9) and access to the user interface of this service (10).

Then, he can begin to invite other chemists to become members of his VC and use the services from the container (11). To do so, he may supply a signed proxy X.509 certificate to the chemist Prof. Edward and provide the link to GSH_1. Prof. Edward can access to GSH_1 in (9) and with his own certificate, he can open another user interface of the Instant messaging service. At this stage, Prof. Edward would see Prof. Marcus and himself in the roster.

Finally, concerning the other services progressively elicited by the VCE, Prof. Marcus will just have to follow the same procedure, which means that he will have the responsibility to choose services according to their ability to satisfy his needs, and to declare them as GSH-x associated to GSH-1, WITHOUT having to manage them any further ! The only "useful access information " for an average community member remaining GSH-1.

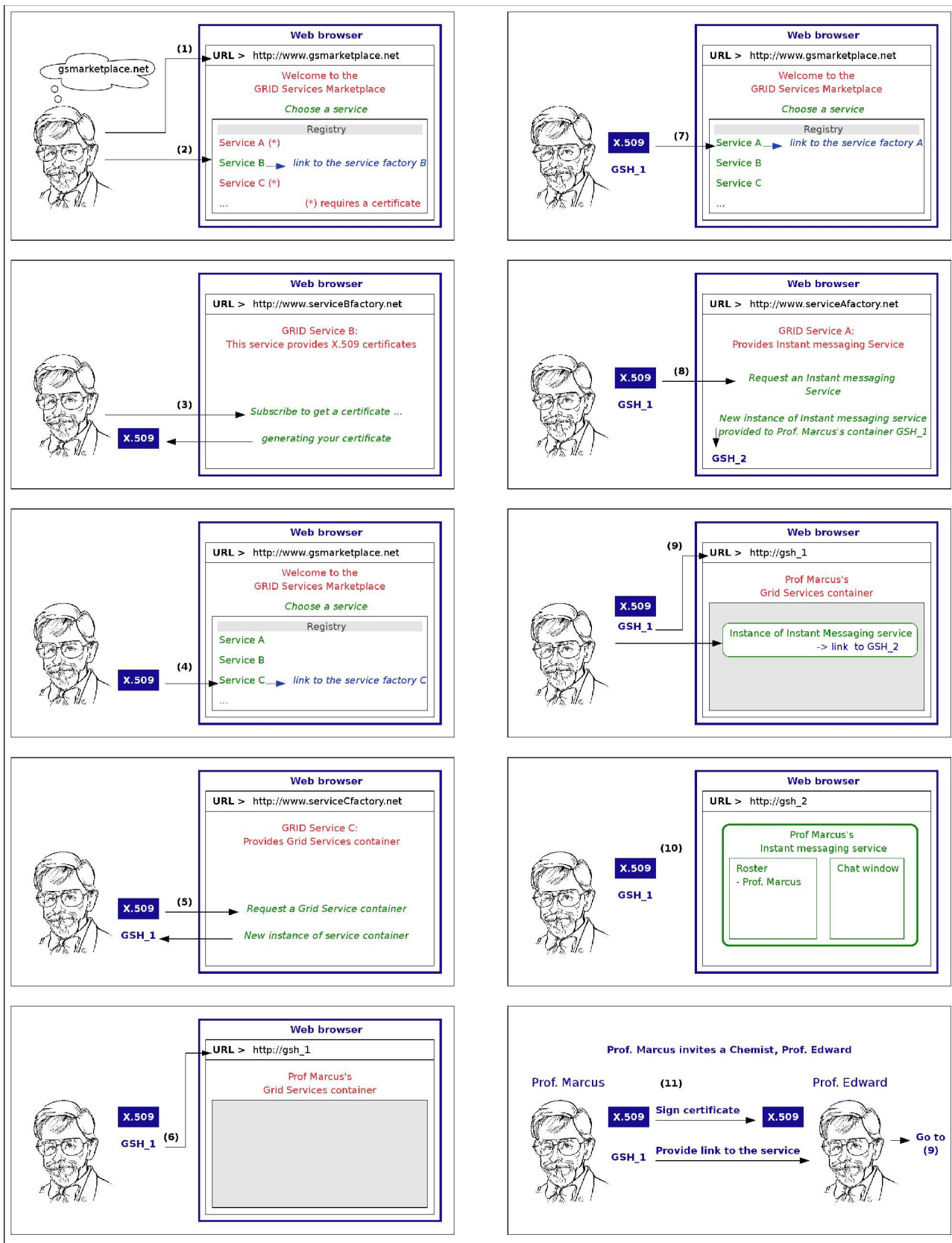


Figure 5: The bootstrapping scenario

3.4 Remarks from this scenario

This scenario has been depicted in a very simplified way. In we should enter the details, we would become even more convinced that such a scenario would not be feasible without the service ubiquity and service state capability of GS.

- Security for enhancing ubiquity: Mechanism of security using the X.509 certificate allows the creation of a trust that is necessary to settle confidence within a VC (ressources and people are distributed in various organisations). Access rights are granted from Prof. Marcus to Prof. Edward using a powerful Grid security mechanism for delegation called the proxy certificate.

- Shared services require state management: All protocol transactions between users and Grid are performed with only the need of a web browser via http. A collaborative service could not be envisaged in this way without the management of service state.

4. CONCLUSION

After having explained the growing interest for CSCL, we have briefly sketched out the main characteristics of Grid, and then produced a mock-up explaining in the context of the initialization of a Scientific collaboration, how Grid could play the decisive role of enabling technology.

We are not pretending here to have solved all the difficulties to be encountered by Prof. Marcus; the interactive construction of Scientific Knowledge still remains a very ambitious and demanding enterprise. But we have come to the point where we can outline two important arguments:

- The first one is that Grid services architecture will free Prof. Marcus from all the technical problems: he does not need any longer to be an expert in system-oriented aspects, nor to have the availability and skills of a WebMaster.
- The second one is related to the use of standardized Communication and Information Management services. Such services, for instance "Instant Messaging" substitute their own conversation settings to "real face to face" collaboration, and therefore embed collaboration protocols, among which the turn-taking rule which can take various implementations depending on the type of collaboration. Our own experience is that the constraints inherent to those collaboration protocols, far from being an obstacle to collaboration, are a very powerful enabler. Moreover standardized Communication services will often provide record of the conversations in a properly managed way, and therefore will help and bridge the gap between instant informal collaboration and knowledge structuration and capitalization. We strongly believe that Grid services are to become a very powerful mediator of human collaboration.

ELEGI ACKNOWLEDGEMENTS

"Work partially supported by the European Community under the Innovation Society Technologies (IST) programme of the 6th Framework Programme for RTD - project ELeGI, contract IST-002205. This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein."

REFERENCES

- [CSCL] Computer Supported Collaborative Learning, Definition and references
<http://www.edb.utexas.edu/csclstudent/Dhsiao/theories.html#what>
- [D13] EleGI WP7 (2004), Preliminary Specification of Virtual Learning Communities, Technical Report D13.
- [EKD03] Eisenstadt, M., Komzak, J. and M. Dzbor, Instant messaging + maps = powerful collaboration tools for distance learning, Proc. TelEduc03, May 19 -21, 2003. Available at <http://buddyspace.sourceforge.net>.
- [FOS02] I. Foster et al. (2002), The Physiology of the Grid, Open Grid Service Infrastructure WG, Global Grid Forum, <http://www.globus.org/research/papers/ogsa.pdf>
- [JAC03] K. Jackson, Lawrence Berkley National Laboratory. ACTS workshop Report 2003.
- [ROU01] D. DeRoure, N. Jennings, N. Shadbolt, (2001) A Future e-Science Infrastructure, In: Report commissioned for EPSRC/DTI Core e-Science Programme. University of Southampton, UK.
- [GOU03] A. Gouaich, S. A. CERRI (2003), Movement and interaction in semantic GRIDs: dynamic service generation for Agents, International LeGE-WG Workshop.
- [RH93] Rheingold, H. (1993). The Virtual Community: Homesteading on the Electronic Frontier. Reading MA, Addison-Wesley.
- [SOT03], B. Sotomayor, The Globus Toolkit 3 Programmer's Tutorial, <http://gdp.globus.org/gt3-tutorial/>