

***PPAM 2007***

***Track B: "Models, Algorithms and Methodologies for Grid-Enabled Computing Environments"***

***A PMI-aware extension for the SSH service***

***Giuliano Laccetti***

***Dipartimento di Matematica ed Applicazioni - Università degli Studi di Napoli Federico II, Via Cintia, 80126 Napoli (giuliano.laccetti@dma.unina.it)***

***Giovanni Schmid***

***ICAR - Istituto di Calcolo e Reti ad Alte Prestazioni - Sede di Napoli, Via P. Castellino n. 111, 80131 Napoli (giovanni.schmid@na.icar.cnr.it)***



## ***Limitations for current grid environments w.r.t.:***

- ***Scalability***: overheads introduced in the management of grid users and resources;
- ***Expressiveness***: amount of granularity in the access control to grid resources, both at the ***VO*** and the ***RP*** layers.

## ***Recent solutions rely on PMIs and PKIs and:***

- provide finer-grain authorization policies at the VO layer;
- distribute throughout the VO managers the load of administering access to the grid



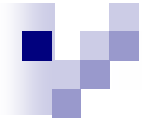
## **BUT...**

- a gap results from the fact that the application / middleware and the **OS** layers are not fully interoperable

## **SOLUTION\*:**

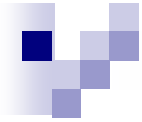
- extend the **A&A** features at the OS layer, in order to obtain a fully interoperable interface with X.509-based infrastructures at the grid layer:
  - Fully integrated design, adhering to modern computing security principles;
  - distributed-oriented OSs act as building blocks of access control distributed architectures

\* G. Laccetti, G. Schmid, A framework model for grid security, *Future Gener. Comput. Syst.* 23 (2007)



## ***OS tier requirements:***

- support for asymmetric authentication through X.509 PKCs;
- support for resource access authorization information through X.509 ACs;
- embedded functions of both ***Certification Authority*** and ***Source of Authority*** for its (registered) users;
- provisioning of an execution environment for both registered and unregistered users, on the basis of the authorization information contained in the ACs for such users.

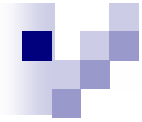


## ***Interfacing the VO layer with the OS one:***

- In grids, a ***gatekeeper*** is a function in middleware that acts as a bridge btw. the VO and the RP tiers
- ➔ gatekeeper implemented as one or more network system entry services;
- ➔ choose a suitable set from the Internet Official Protocol Standards, and extend them in order to obtain fully-compliant PKI-PMI services.

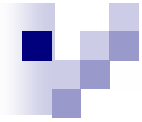
## ***The goal of the present work:***

- a PMI-aware extension for SSH, in order to obtain a first and significative example of a no-grid-specific alternative to a gatekeeper



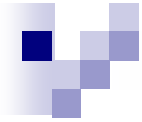
## ***Functional requirements for a PMI-aware SSH:***

- Let ***U*** and ***S*** be a remote user and a user with an account on ***H***, respectively.
- We require that:
- ***U*** can authenticate herself on ***H***, no matter if she has or not an account on ***H***;
- ***S*** can eventually entitle ***U*** to get a profiled shell on ***H*** and / or to use a restricted set of the utilities and applications on ***H*** for which he has execute permissions.



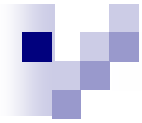
## ***Direct delegation:***

- Item (2) is the ***direct delegation*** requirement:  
*the capability of users on a given OS to delegate to people, who are not registered on such a system, their rights concerning resource usage.*
- It greatly improve grids (and other complex distributed environments) as collaborative environments
- If ***S*** delegates ***U*** his rights concerning resource usage on ***H***, we say that ***S*** is a ***sponsor*** of ***U*** on ***H***



## ***Design requirements - 1/3***

- Our design refers to a ***NSS*** and ***PAM*** enabled implementation of SSH:
- PAM and NSS are the abstraction frameworks used for access control in (virtually) any modern Unix-like system
- ➔ only slight modifications are required at the SSH code: they just concern the SSH protocol itself, not any authentication nor authorization method
- ➔ our design can be quite easily adapted to any other NSS and PAM entry system server



## ***Design requirements - 2/3***

- adoption of ***LDAP*** as a remote and distributed repository of information about PKCs and ACs.
- LDAP support for PKCs was discussed in [\*] and is now a proposed standard [\*\*]
- at least at our knowledge, no support actually exists for ACs.
- Discussing such extension is outside the scope of the present work, and will be treated in a next paper.

\* S. Boeyen et. Oth., Internet X.509 Public Key Infrastructure LDAPv2 Schema, RFC 2587 (1999)

\*\*K. Zeilenga, Lightweight Directory Access Protocol (LDAP) Schema Definitions for X.509 Certificates, RFC 4523 (2006)



## ***Design requirements - 3/3***

- relying on the ***pull distribution mechanism*** for certificate sharing between the SSH client and server:

*certificates for the Authentication / Authorization of a user are retrieved by the SSH server through **name services***

- This choice was because it simplifies the add-on functions required for both the SSH protocol and the client user interface
- It requires only slight modifications to NSS



## ***PAM extensions***

### ■ **pam\_X509\_authE:**

- searches for ***U***'s public-key ***Kp(U)*** in a local repository of X.509 PKCs and through an LDAP DUA
- compares ***Kp(U)*** with the public-key sent by the client in its access request and, if they match, then checks the signature

### ■ **pam\_X509\_authO:**

- checks if the ***U***'s DN contains a host name which differs from ***H*** and, if this is the case, asks the client for a sponsor ***S***
- searches for ***S*** in a local repository (passwd) and through an LDAP DUA



## ***Client user interface extensions***

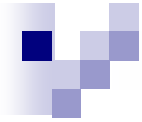
- Current implementation:

```
$ ssh [options] [-l login_name]  
  hostname | user@hostname [command]
```

- Proposed extension:

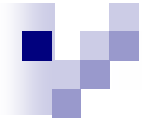
```
$ ssh [options] [-l login_name |  
  distinguished_name] hostname |  
  user@hostname [command]
```

- It results in adding the parameter type X.509 distinguished name both for the **-l** option and the argument ***user***



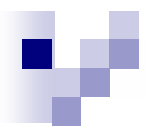
## ***SSH protocol extensions 1/2***

- SSH protocol version 2 stack:
  - ***ssh-trans*** performs server host authentication, key exchange, encryption, and integrity protection
  - ***ssh-userauth*** provides a suite of mechanisms that can be used to authenticate the client user to the server
  - ***ssh-connect*** specifies a mechanism to multiplex multiple streams (channels) of data over the confidential and authenticated transport

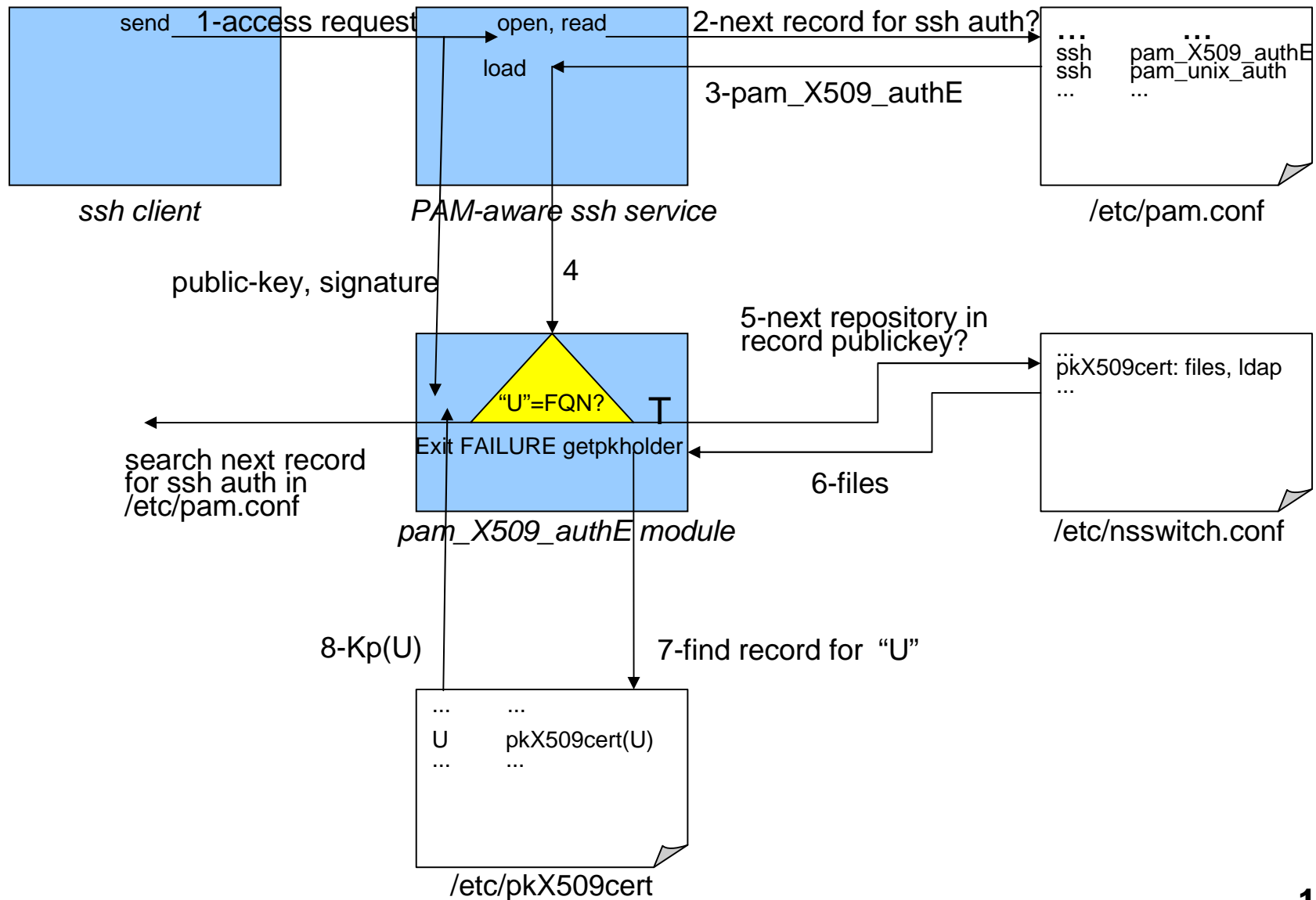


## ***SSH protocol extensions 2/2***

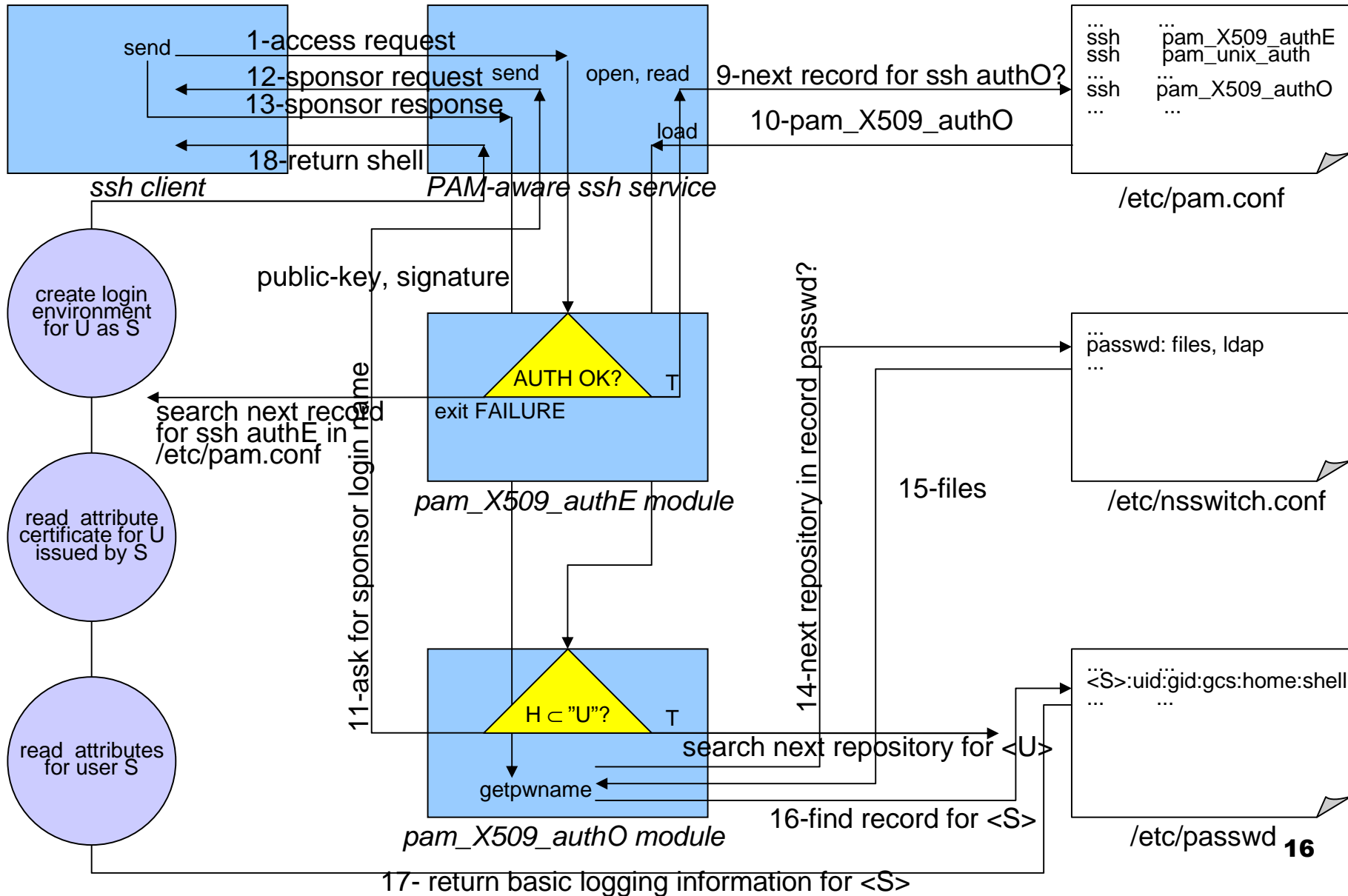
- Our design just requires to add one extension protocol to ***ssh-userauth***, in order to specifically support direct delegation.
- ***ssh-userauth*** provides a built-in authentication mechanisms named ***publickey***, on which we rely to perform user authentication using public keys.
- The extension protocol operates after ***ssh-userauth*** and before ***ssh-connect***, and it is required to realize the sponsor request-response interaction.

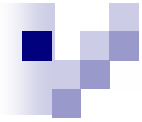


# Authentication flow example



# Authorization flow example





## ***Conclusions***

- PMI-aware extension for SSH, in order to obtain a first and significant example of a no-grid-specific alternative to a gatekeeper.
- we choose to rely on PAM and NSS frameworks.

## ***Advantages***

- only requires slight modifications to the SSH protocol and the client user interface;
- could be easily generalized to encompass any other (PAM-aware) system entry service;
- promotes the adoption of LDAP for conveying user, host and application information.