Network Dial Access
Distributed Authentication and Authorization Architecture

John Vollbrecht
March 25,1994

1.0 Introduction

This document describes the environment and some of the requirements for distributed dial access in MichNet and similar network service providers. The architecture to support these is described, and some unresolved issues are discussed.

MichNet is a statewide network that provides IP connectivity in the state of Michigan and interconnection to the international IP internet. MichNet is a consortium of state universities in Michigan and is managed by Merit Network Incorporated. Dial service is provided at a number of locations within the state, and the expectation is that the number of locations will increase significantly over the next two years.

Dial in servers, called Network Access Servers (NAS), are normally located at sites where there are existing network connections. Occasionally, a NAS will be located at a new site to service users in a particular local calling area. Users calling a NAS may identify themselves with a user id and the name of an authentication realm in which they are known. If they choose not to identify themselves, their access to the net is restricted significantly, and they are not allowed to send packets to the international IP network.

Typically users identify themselves to the NAS, and the NAS forwards their id and security information (e.g. password) to an authentication and authorization server. The proper server is identified through the realm name associated with the id. If a user id is valid, the authentication and authorization server responds favorably and provides an indication of user permissions. If the user id is invalid, the authentication and authorization server responds negatively.

1.1 Routing Problem

In this discussion, the user dialing the network is called the Network Access Client (NAC), the server that supports the dial lines is called the Network Access Server (NAS), and the authentication and authorization servers at remote sites are called AAServers.

Figure 1 helps illustrate the interconnection/routing problem. Each NAS, and there may be hundreds of them, must be able to route messages to each AAServer, and there may be hundreds of them. AAServers are added as new institutions join MichNet. In addition to the address mapping problem (knowing how to get to the AAServer), there is a network security question. A NAS must be able to verify that messages are from "real" AAServers and that they are not being spoofed by an attacker attempting to allow access to the network for unauthenticated users.

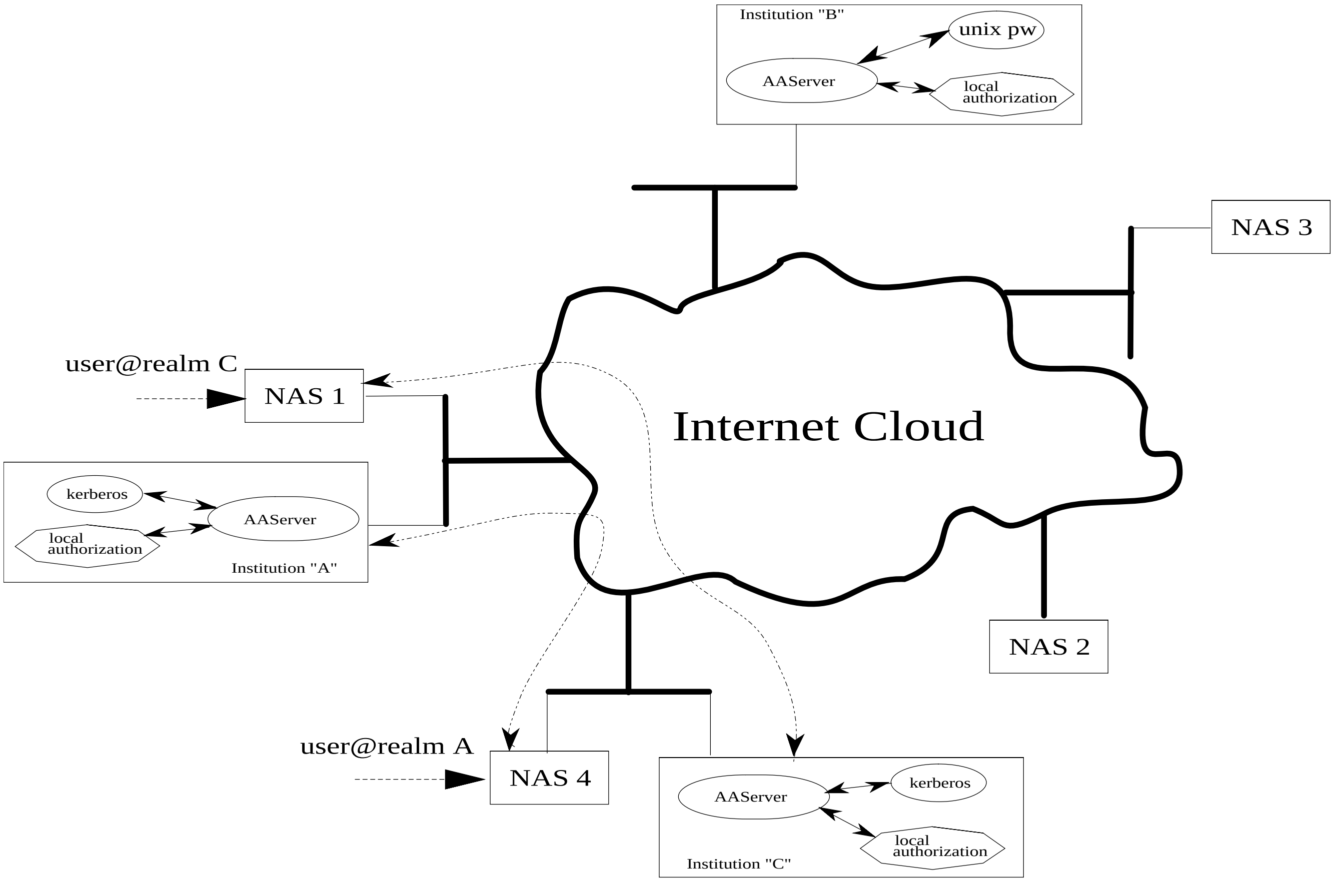

Figure 1 - NAS requests Authentication/Authorization From Institution specified with uid

1.2 Alternative Approaches

Figures 2 and 3 illustrate alternative ways of sending requests and responses between NASs and AAServers.

Figure 2 shows a message switch that passes messages between AAServers and NASs. This switch must know how to reach each of the AAServers and NASs. Both the NASs and AAServers route all requests and responses to the message switch, which then forwards the information to the appropriate destination. The message switch is located in a secure area within a secure machine, and all of the routing is configured within the switch. Altering the routing table is possible only on the switch itself, not at the end nodes. The switch shares a secret with each NAS and AAServer, thus messages between them are signed and perhaps encrypted. This guarantees that a response is from a known AAServer.

Figure 3 depicts an alternative configuration where there is a table of realm names corresponding with both the ipaddress and the public-key of the AAServer. The NAS queries this table using an rpc mechanism to obtain the AAServer's public-key and ipaddress to which the request will be sent. The AAServer signs the response with its private-key which allows the NAS to be confident that the response was really from the appropriate AAServer. This requires that both the NAS and AAServer support public-key. To reduce the risk of tampering, the table with ipaddress and public-key information is located in a secure machine at a secure site so that an intruder cannot easily substitute a different ipaddress and public-key.

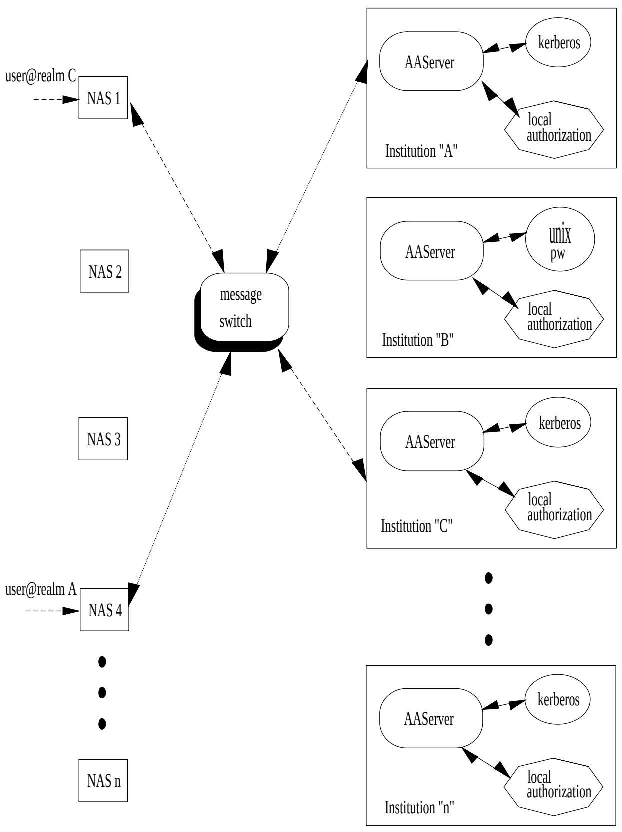

Figure 2 - Message switch - routing by authentication realm and NASid
message switch maps requests and responses between NAS and AAServers

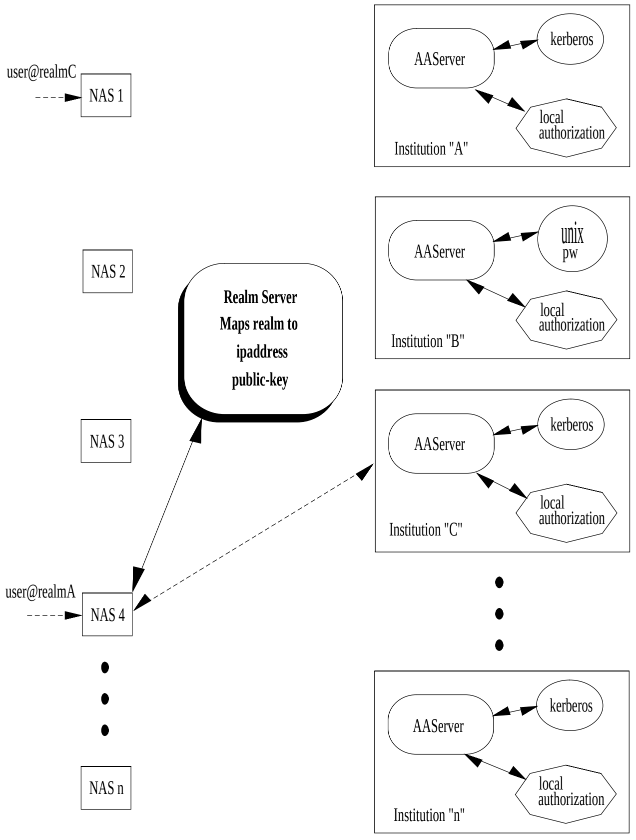

Figure 3 - NAS get ipaddress and public-key of AAserver from key distribution center

The approach in Figure 2 is similar to providing a router to which all NASs and AAServers are connected. The router knows how to send messages to each of them based on realm or NASid.

The approach in Figure 3 is modelled after the DNS method, where the realm name is equivalent to the domain name, and the ipaddress and public-key are equivalent to the ipaddress and other attributes of a DNS data base.

1.2.1 "Switch Approach"

The Switch approach requires that the switch has secure sessions with each NAS and AAServer. This can be done by manual configuration or by having each NAS and AAServer establish a security session (e.g. via Kerberos) with the switch.

Configuring by hand would require additional work, but mostly at the switch. Each NAS and AAServer would only be required to have a session (and most likely a backup session) with the switch, while the switch would have a session for each end point. Session configuration would include a different shared secret for the session.

Using the security session, each NAS and AAServer would, for example, be registered with a Kerberos data base. The NAS and AAServers each obtain a session ticket for the switch and initiate a connection.

The switch accepts the connection and enters information in its internal routing table. The switch does not have to be reconfigured to support new realms or NASs.

An extension of the single switch idea is to have more than one switch, with some NASs and/or AAServers, connected to each switch. Doing this enables routing between switches. For this approach to be effective, there needs to be some sort of hierarchical naming and/or routing information passed between switches. While possible, this seems more complicated than necessary for this situation.

1.2.2 "DNS" Approach

In the DNS approach, a table of realm names with associated ipaddresses and public-keys is maintained in a secure fashion. This is similar to the X.509 approach.* Responses to queries for information from the table need to be signed so that the caller knows the response is from the real "realm name" server.

It appears that only realm names for AAServers (not the NASs) need to be in the table. This is because queries from the NAS contain the return address, and therefore no ipaddress is needed. In addition, the AAServers are not required to know the identity of the requestor. This needs further discussion and consideration, but if this is the case, no configuration will be required when adding a NAS.

2.0 Adding a NAS "helper"

This section shows how the routing function, described in Section 1, can be combined with helper functions supported by several venders. NAS vendors (including Livingston, Xylogics, and Cisco) use an architecture in which functions are divided between the physical NAS system (including its local software) and a helper process running on a different system. This allows some NAS functions, typically dealing with connection establishment, accounting, etc, to be implemented on a general purpose system where memory size and disk storage are not constrained. The helper is logically part of the NAS, as shown in Figure 4, and the routing structure is identical to Figure 1.

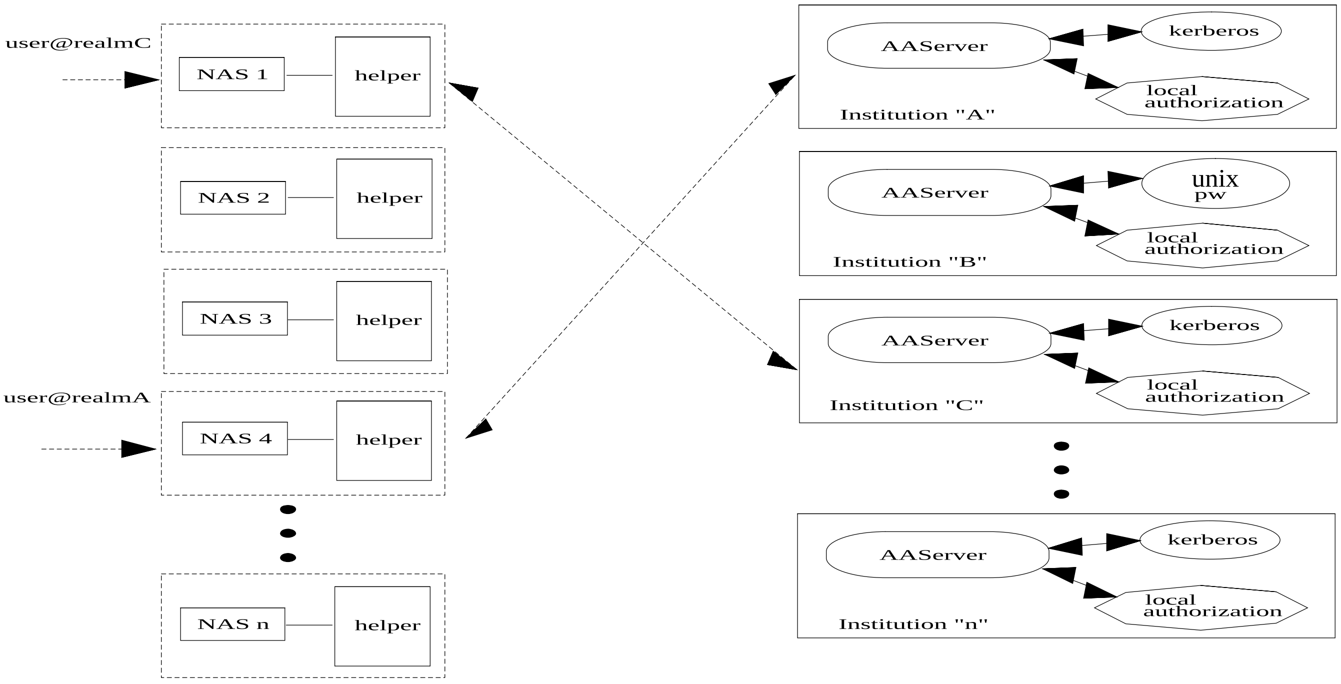

Figure 4 - NASs connected to "helpers" which pass requests to remote AAServers

Figure 5 shows a modified NAS/helper configuration where each helper can support multiple NASs. A pool of helper processes exists, and each NAS connects to a primary helper and perhaps several backup helpers. Note that Figure 5 is very similar to Figure 2 in that all requests first go to the helper (message switch) and are then forwarded to the remote AAServer. Adding a switching function to the helper makes the two figures identical in the way AArequest/response routing is handled.

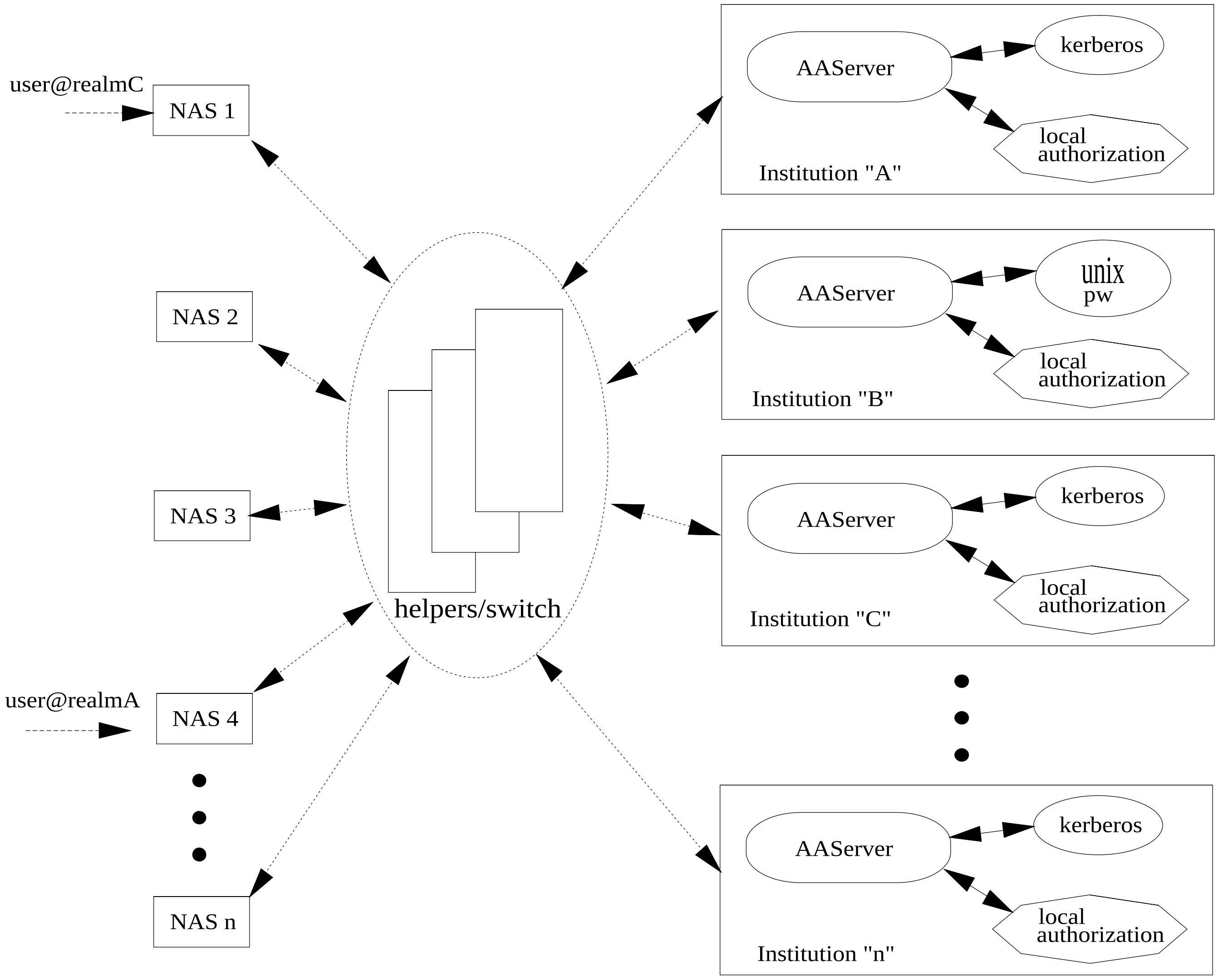

Figure 5 - NASs connected to "helpers" which pass requests to remote Authentication /Authorization Servers

In this environment, the helper(s) also acts as a message switch. Note that the helper(s) could forward all requests to a separate message switch (as in Figure 2) or use a table of ipaddress/ public-keys to get the realm address (as in Figure 3). In an environment where all NASs connect to a limited number of helpers, it may become simpler to just use the helpers in a dual role of helper *and* switch. It is important to remember that these are different functions that can be combined in a single process.

3.0 Multi Vendor Implementations - Interface Standards

Figure 6 shows the paths between NAS-helper-AAServer. Request messages are sent between the NAS and the helper, and then between the helper and the AAServer. From an implementation view, it is tempting to try to make the protocols the same. However, the requirements for the NAS-helper protocol may be different than the requirements for the helper-AAServer protocol. For example in some implementations, the helper may pass filters to be applied by the NAS, while in others it may pass only the names of the filters already in the NAS.

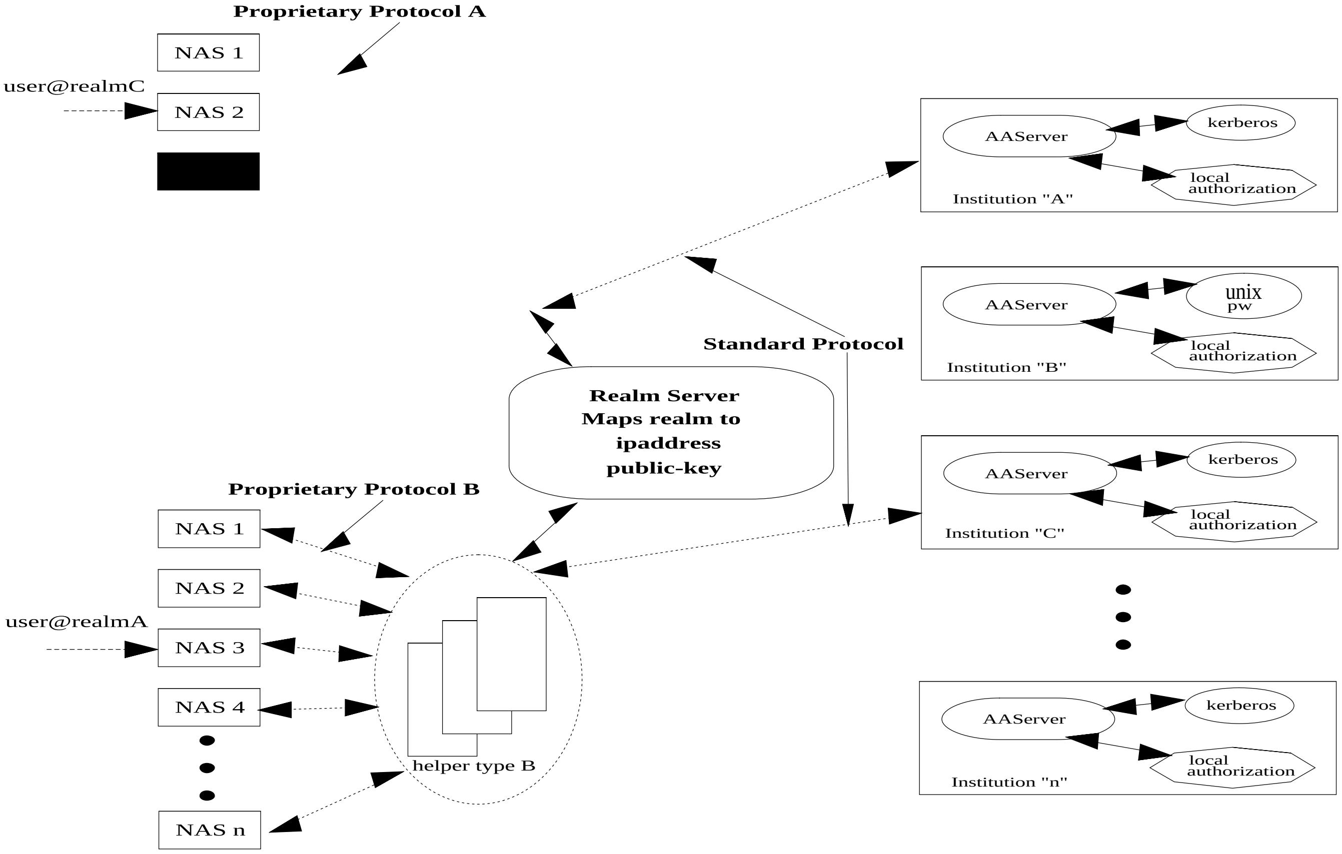

Figure 6 - NASs connected to "helpers" which pass requests to remote Authentication /Authorization Servers

Figure 7 shows two helper interfaces. To participate in a distributed AAServer community, it would seem that at least the helper/AAServer protocol interface should be standardized. The NAS/helper protocol might vary between vendors, and the decisions on which functions are performed in the NAS and which in the helper may be different. A future paper will discuss the requirements of the NAS/helper protocol.

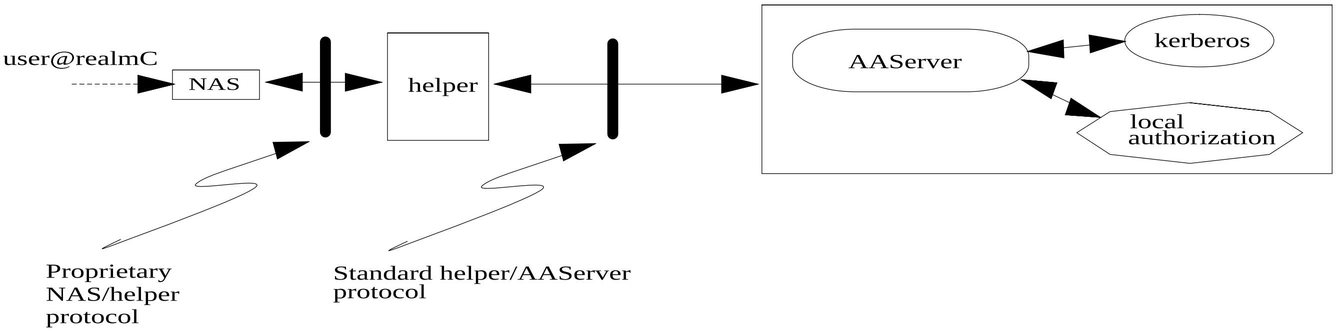

Figure 7 - Helper Interfaces - to NAS, AAServer, or other helpers