NameFLOW Replication Architecture

1. Introduction

The former NameFLOW architecture was based entirely on Quipu model. It had two drawbacks:

1. Most Quipu software is Y2K uncompliant. Although DAP and DSP protocols are still working well, the replication of data between servers is broken since 1st January 2000.
2. The free Quipu implementation - ISODE 8.0 - is owned by DANTE. The implementation is old, it would take a lot of time and effort to correct the Y2K problem, so it is not supported anymore. In addition, most NameFLOW participants use commercial Quipu software, derived from ISODE 8.0. There is no more vendor support for the commercial Quipu software.

Therefore, there is a need for a new directory architecture. The new architecture should meet the following requirements:

distributed:
as for the old Quipu architecture, most organisations want to manage their own directory servers;

simple and based on existing standards:
the result can be a faster implementation and ability to use different vendors' software;

scalable and manageable:
special servers may integrate organisational directories into national directory trees, both in technical and administrative (registry) aspects; similar integration is necessary on the international level;

efficient and reliable:
there are two main operations used in directories: searching and browsing.

Browsing is the same as one-level searching. It is usually provided via WWW gateways to directories. A user requests information about some entry (i.e. the entry's attributes and values) and the list of all its direct subentries. Subentries are often listed with their descriptions. Obtaining descriptions of the subentries requires access to the subentries' attributes. As it almost always happens in a distributed architecture, subentries are stored in other directory servers. Accessing their attributes requires the request to be passed to many servers. Even if it is done simultaneously, it takes some time to wait for a response from the last server or for a timeout, if some servers are unavailable. In case of many subordinate servers, delays become unacceptable.

A solution to improve response time for browsing is one-level replication. If the subentries are shadowed on the server, the request can be processed on the server by itself. This is correct on all levels of the directory: organisational servers should shadow their organisational unit entries, while national servers should shadow both organisational entries from servers inside their country and country entries from other national servers.

While the one-level replication improves the efficiency of browsing (i.e. one-level searching), it does not solve the problem of subtree searching. Due to the size of the whole directory, it is practically impossible to replicate all its entries to the one server. The solution being developed in the framework of the DESIRE II project is **DESIRE Generic Distributed Indexing Server**, a

component of DESIRE Integrated Toolkit.

multiprotocol:

There are two technologies complementing each other in the directories area now: X.500 ('93 edition) and LDAP. Both have their advantages and disadvantages. X.500 (93) is based on OSI protocol stack, and therefore its implementations are more complicated. But it has the special DISP protocol for replication, providing better support for highly distributed directories. LDAP is based on TCP and is much simpler than X.500 as a result. But it does not define any standard means of replicating data between servers. Both technologies are currently widely used, and it is necessary for NameFLOW to support both of them. The support of the old Quipu servers is open to question and can be done on a best effort basis.

Problems facing the development of the new architecture and the proposed solution are described further on in the paper.

2. Rough and ready solution - "full mesh"

The first obvious solution providing every FLDSA with copies of all first level naming contexts from other FLDSAs is a "full mesh" of shadowing agreements between all FLDSAs (see Fig 1).

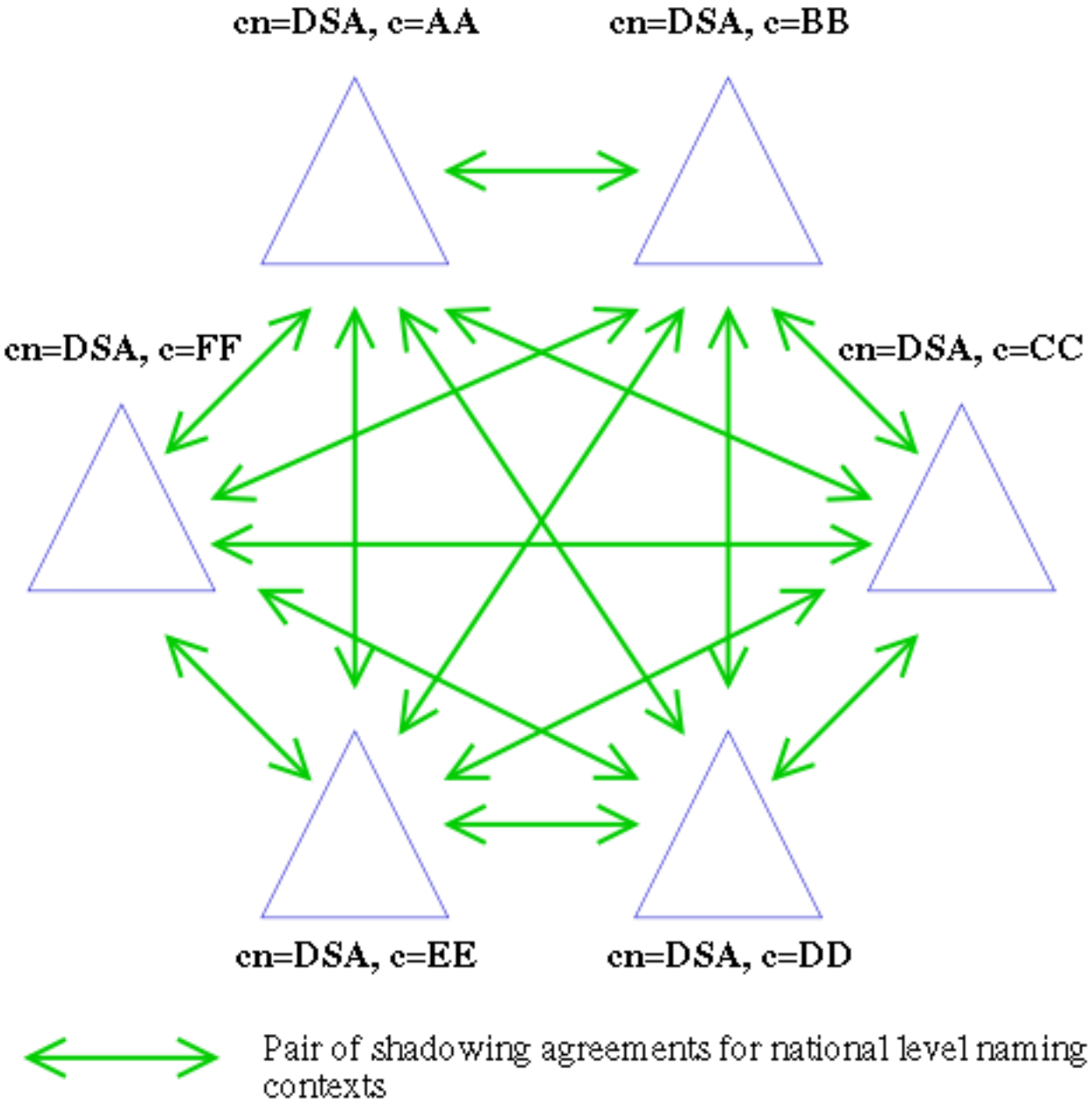

Fig 1. **The full mesh of shadowing agreements**

In this topology, each server would have two shadowing agreements with every other server (a consumer agreement for the naming context mastered by this server and a supplier agreement for a naming context mastered by its peer). Given *N* FLDSAs, every server would have *(N-1)*2* shadowing agreements. And there would be *N*(N-1)* agreements in total.

Generally, every FLDSA manager needs to create agreements only once, and from the technical point of view, an amount of *(N-1)*2* agreements is not critical for a DSA's efficient operation. What is unacceptable in this topology, is the necessity of administrative relations between every pair of managers. The best way to solve this problem is introducing a registry, i.e. a central point of the information interchange.

3. Two step replication: aggregation and distribution

This model was successfully used in Quipu architecture. It requires a special DSA which shadows

national directory data aggregating them in one place and then distributes them back to national servers. Every national DSA makes secondary shadowing of all national naming contexts except for its own from this special DSA, which becomes the center of a star-like topology (Fig 2).

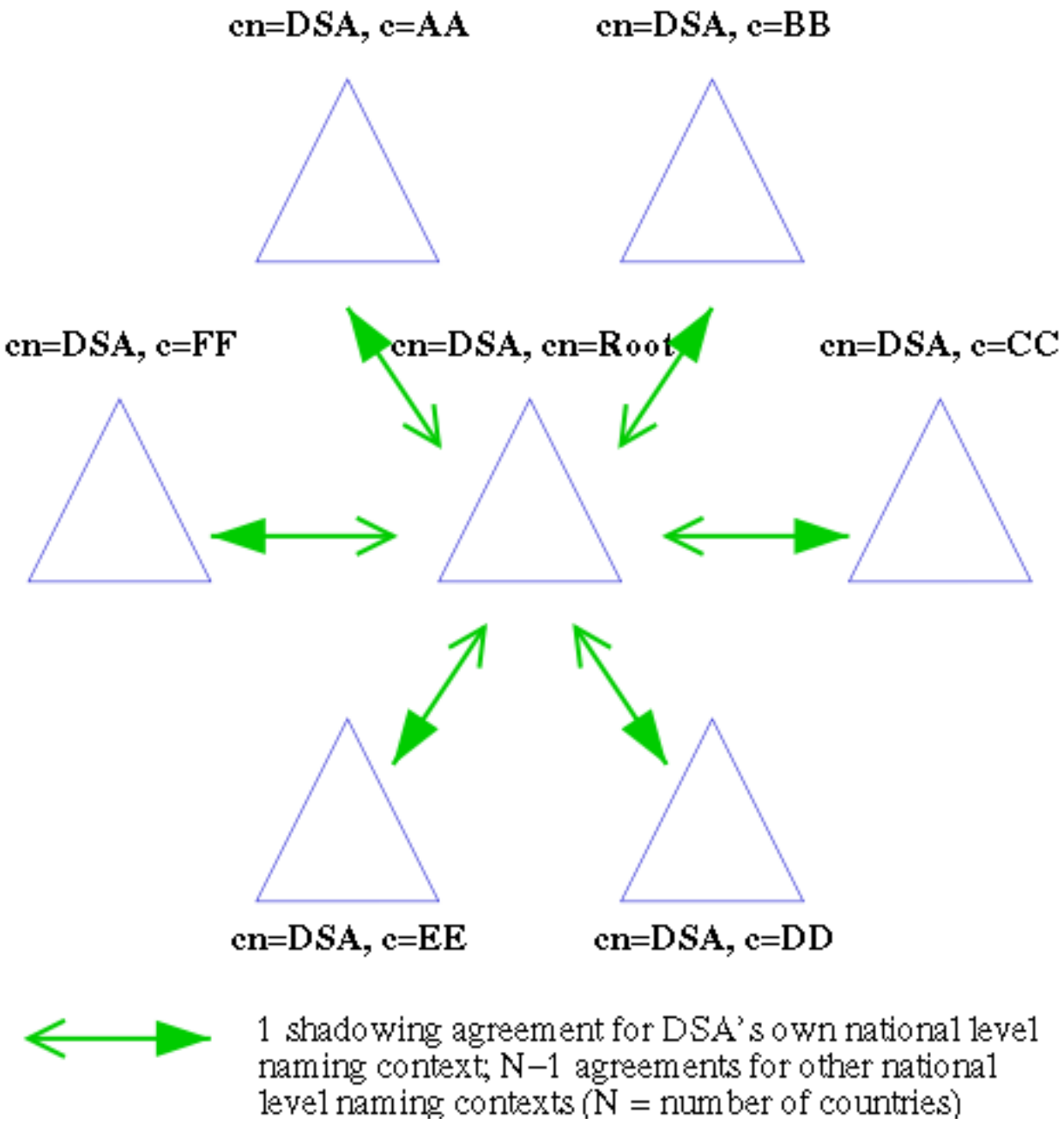

Fig 2. **Star topology for shadowing agreements**

This approach solves the problem of administrative relations. Every FLDSA manager has only to contact the manager of the central (root) DSA to create shadowing agreements for all naming contexts.

Unfortunately, due to lack of the concept of Root naming context in X.500 (93), one still needs to set up a separate shadowing agreement for each first level naming context. Every FLDSA in this case would have *N* agreements with the Root DSA (one supplier agreement for its own naming context and *N-1* consumer agreement for all other naming contexts). This is less then the number of agreements in the full-mesh topology.

But the Root DSA would need *N*N* agreements in that case, which is unacceptable for a large amount

of FLDSAs. A method is necessary for decreasing the number of shadowing agreements.

4. Proposed solution: artificial root level naming context

The X.500 (93) standard poses the following problems:

1. there is no root naming context; therefore there is no way to have a directory entity including all first level entries and (references to) their subentries;
2. replication requires at least one shadowing agreement for each naming context (in other words, a shadowing agreement cannot span several naming contexts, even a pair of superior and subordinate ones); even if the root naming context existed, there would be impossible to have a replication of several first level naming contexts with one shadowing agreement. The first problem can be solved by making a special first level entry (with its own naming context) named, say, `<cn=Root>`, and placing all first level entries underneath this entry. Then every entry in the new directory tree would have a name `<*, cn=Root>`. This extra element in every distinguished name would, of course, be inconvenient for use by end users. But if we had copied, not moved, actual entries under

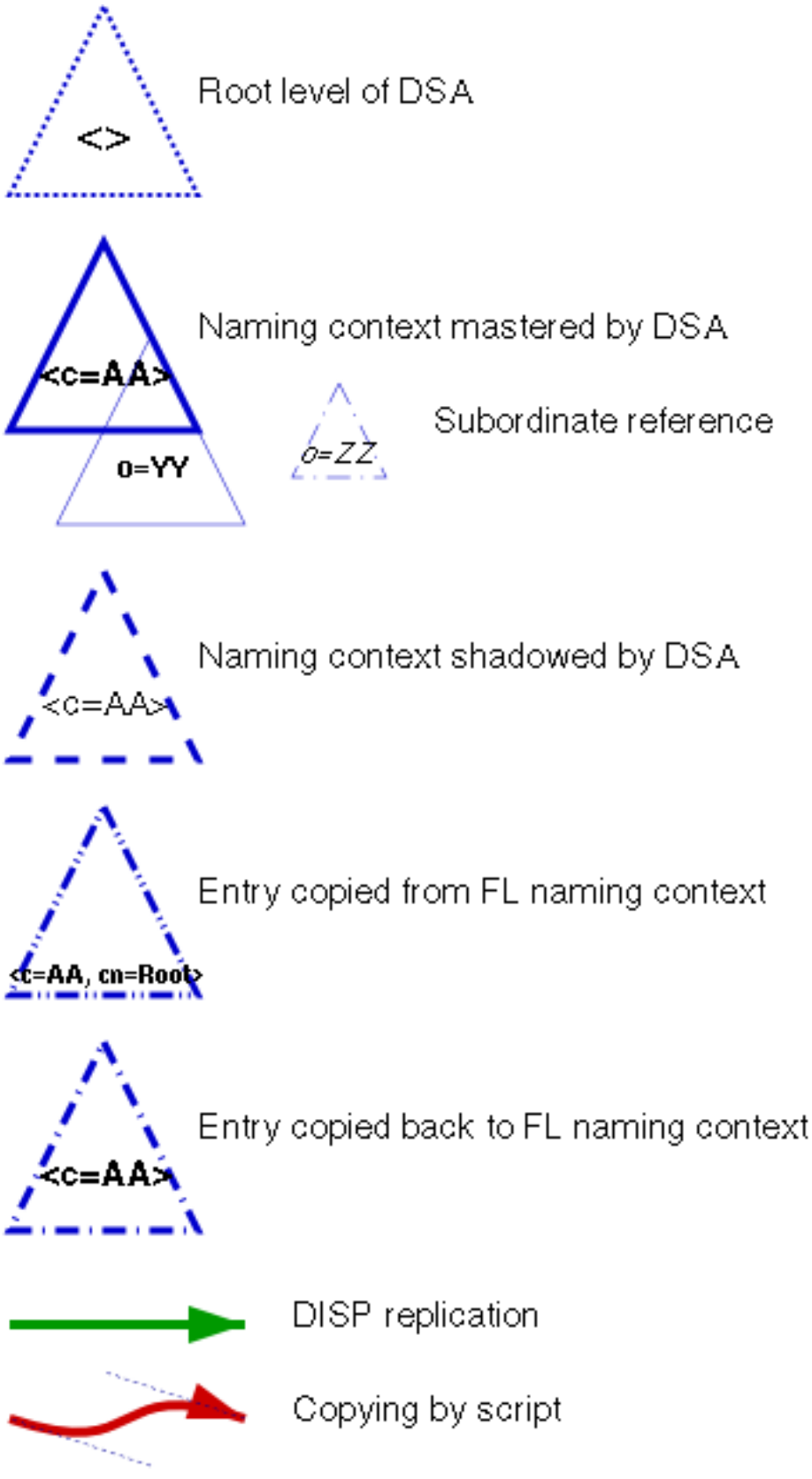

Fig 3. **Signs description**

<cn=Root>, leaving the originals in their old location, then we could use the new entries for replication and the originals for other directory operations.

When copying first level entries underneath <cn=Root>, we must not copy the corresponding naming contexts, so that all the entries are in the same first level <cn=Root> naming context and are allowed to shadow using the one agreement. The second problem would be solved this way, too.

The process of replication of first level naming contexts is described on Fig 4. The four steps from **1a** - **4a** represent the replication of the <c=AA> naming context, the **b** steps show the replication of <c=BB>. The sequence is: (**1**) DISP replication of national naming contexts from FLDSAs to the root DSA; (**2**) copying national naming contexts underneath the special root naming context; (**3**) DISP replication of the special root naming context from the root DSA to every FLDSA; (**4**) copying national naming contexts up to their original location on every FLDSA. See Fig 3 for the description of signs used on pictures.

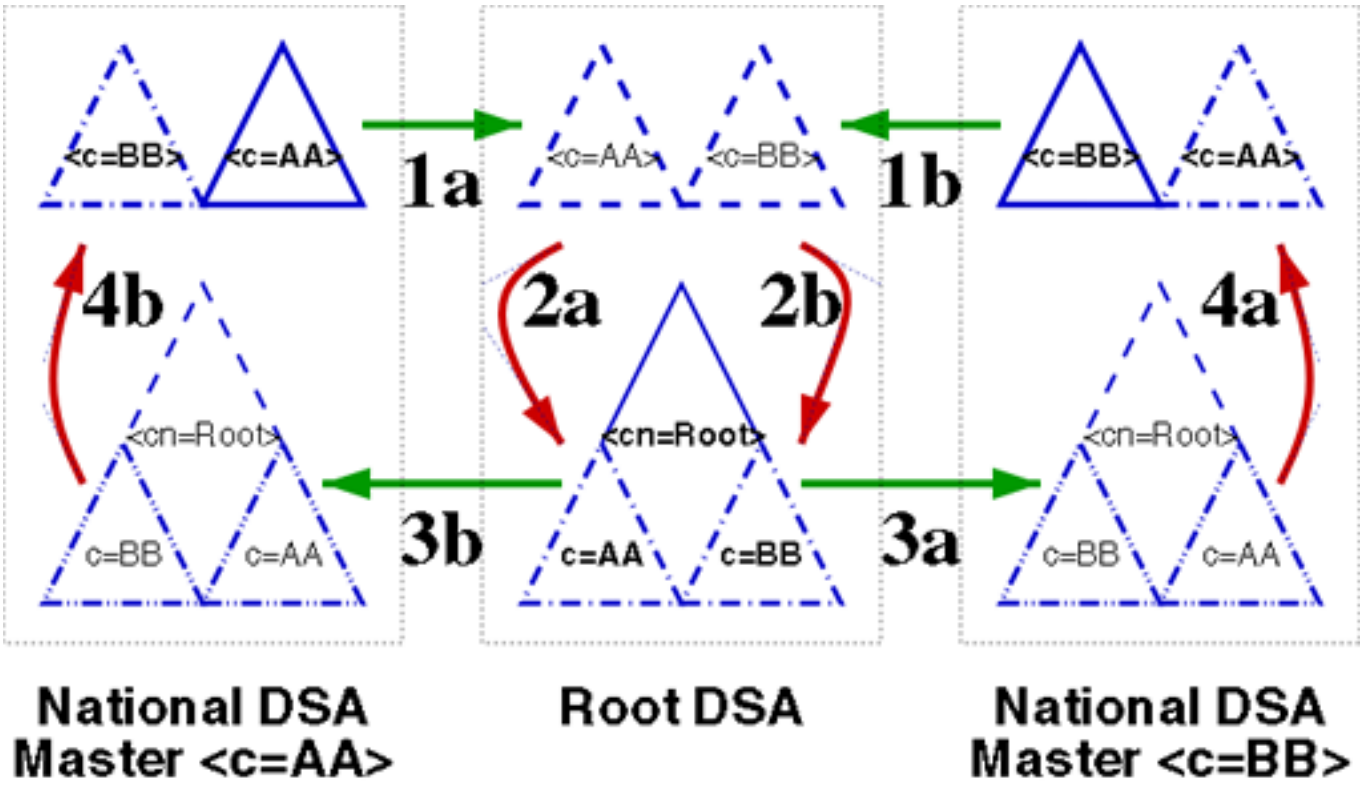

Fig 4. **Replication of national naming contexts**

Every step is described in more details below.

4.1. Aggregation: shadowing a national naming context.

The root DSA has consumer shadowing agreements with every FLDSA for their first level naming contexts (see Fig 5).

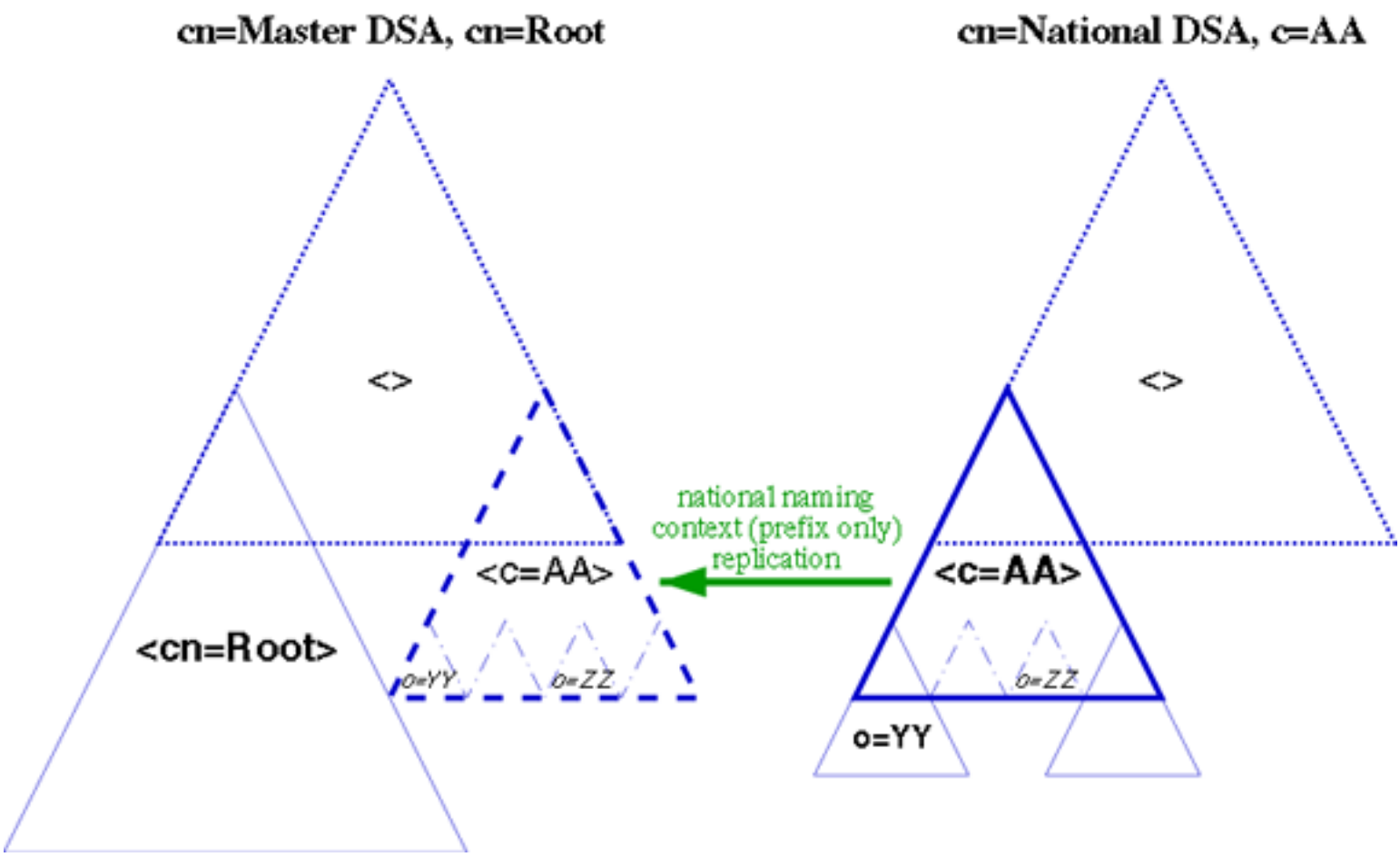

Fig 5. **Shadowing a national naming context**

The agreement is for naming context prefix only. The shadowed naming context therefore consists of the first level entry and subordinate references to second level entries. If the original naming context contains second level entries, they are replaced in the shadowed copy by subordinate references to the national DSA (o=YY on Fig 5).

4.2. Copying national naming contexts underneath <cn=Root>

A script running on the root DSA copies shadowed first level naming contexts underneath <cn=Root> (see Fig 6). Copied are first level entries and all their subordinate references. Naming contexts themselves are not copied, which allows all the new entries to be in the same naming

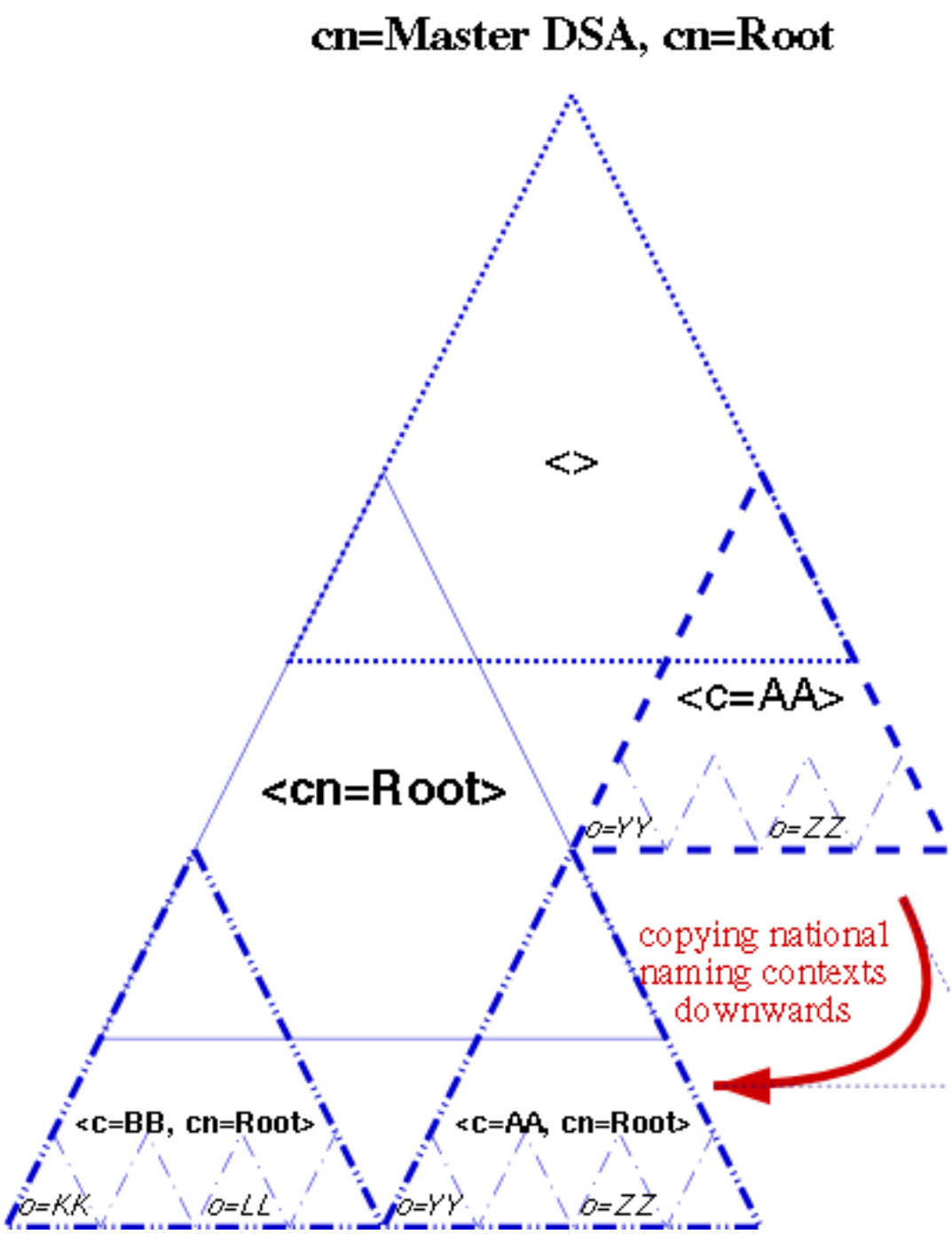

Fig 6. **Copying national naming contexts downwards**

context.

Note that although the new subordinate references point to the same presentation addresses as their originals, they have different prefix: `<*, c=AA, cn=Root>`. As there are probably no such entries on the DSAs the references point to, these references are invalid and cannot be used in normal directory operations. Nevertheless, they are not intended to be used that way.

The script is running periodically. As both source and target entries are on the same server administered by the one manager, the latter can use any form of authentication for the script to connect to the server. And there is no need to exchange the authentication information between FLDSA managers.

4.3. Distribution: shadowing the root naming context

The next step is the replication of the root naming context to every FLDSA (see Fig 7).

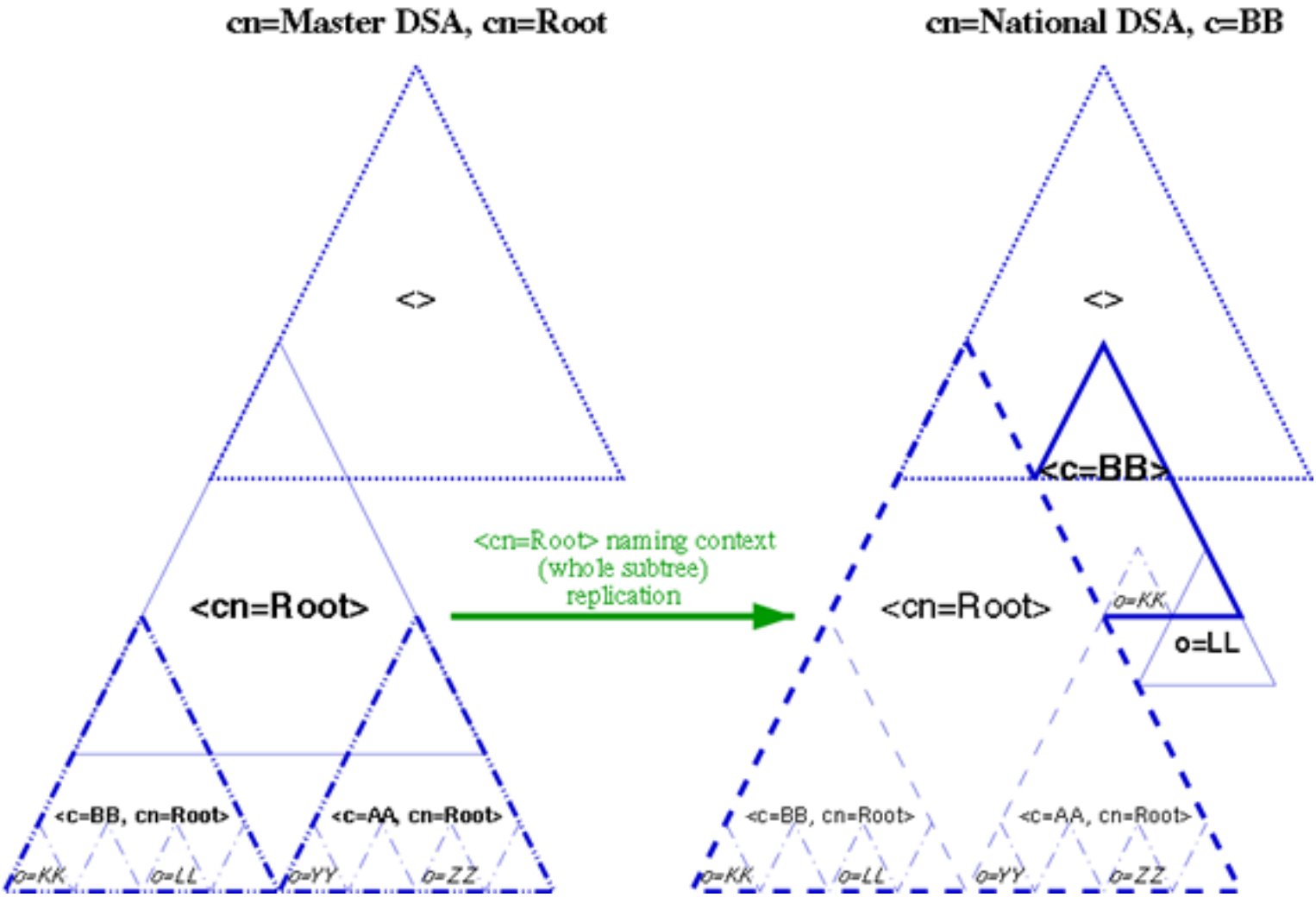

Fig 7. **Shadowing the root naming context**

The whole `<cn=Root>` naming context is replicated, that is, all national level entries and all their subordinate references. Every FLDSA gets the copy of the artificial root naming context.

4.4. Copying national naming contexts upwards

Another script running on a FLDSA copies entries from `<cn=Root>` naming context with their subordinate references back to the first level (see Fig 8). The important point here is that

entries already mastered by this FLDSA (<c=BB [, cn=Root] >in this example) are not copied. Naming contexts and administrative points are created for each new first level entry, therefore the FLDSA becomes a master for every first level naming context.

The previously broken subordinate references (<*, c=AA, cn=Root>) change their distinguished names back during copying, and so become correct again: <*, c=AA>.

The script needs access to the FLDSA only, and can use any form of authentication to connect to the server. As for the script on the root DSA, the authentication information can be kept internally by the FLDSA manager.

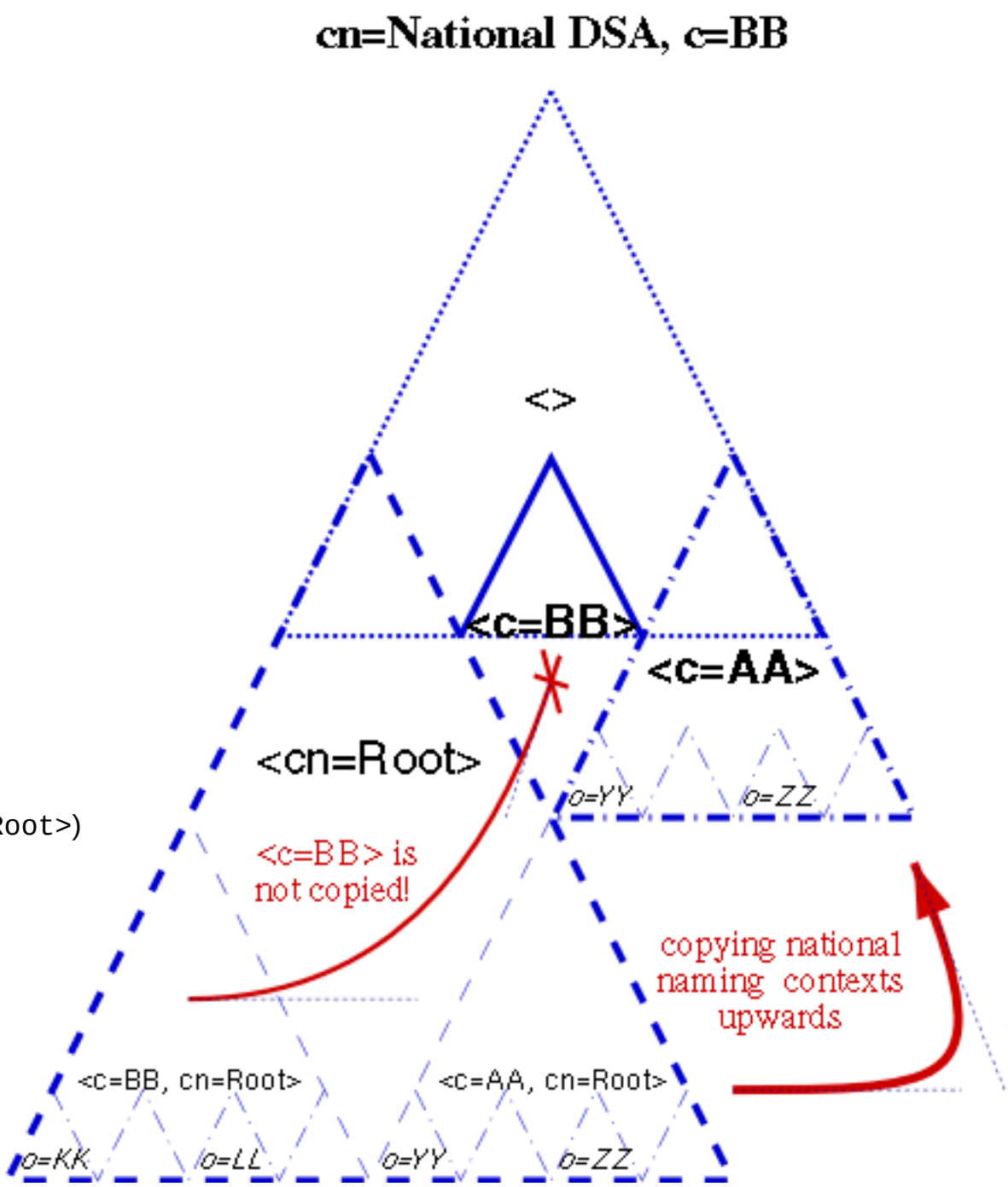

Fig 8. **Copying national naming contexts upwards**

5. Results

Using this approach allows us to have only 2 shadowing agreements for each FLDSA, and *2*N* agreements on the root DSA (see Fig 9). This is much more scalable then the simple star topology.

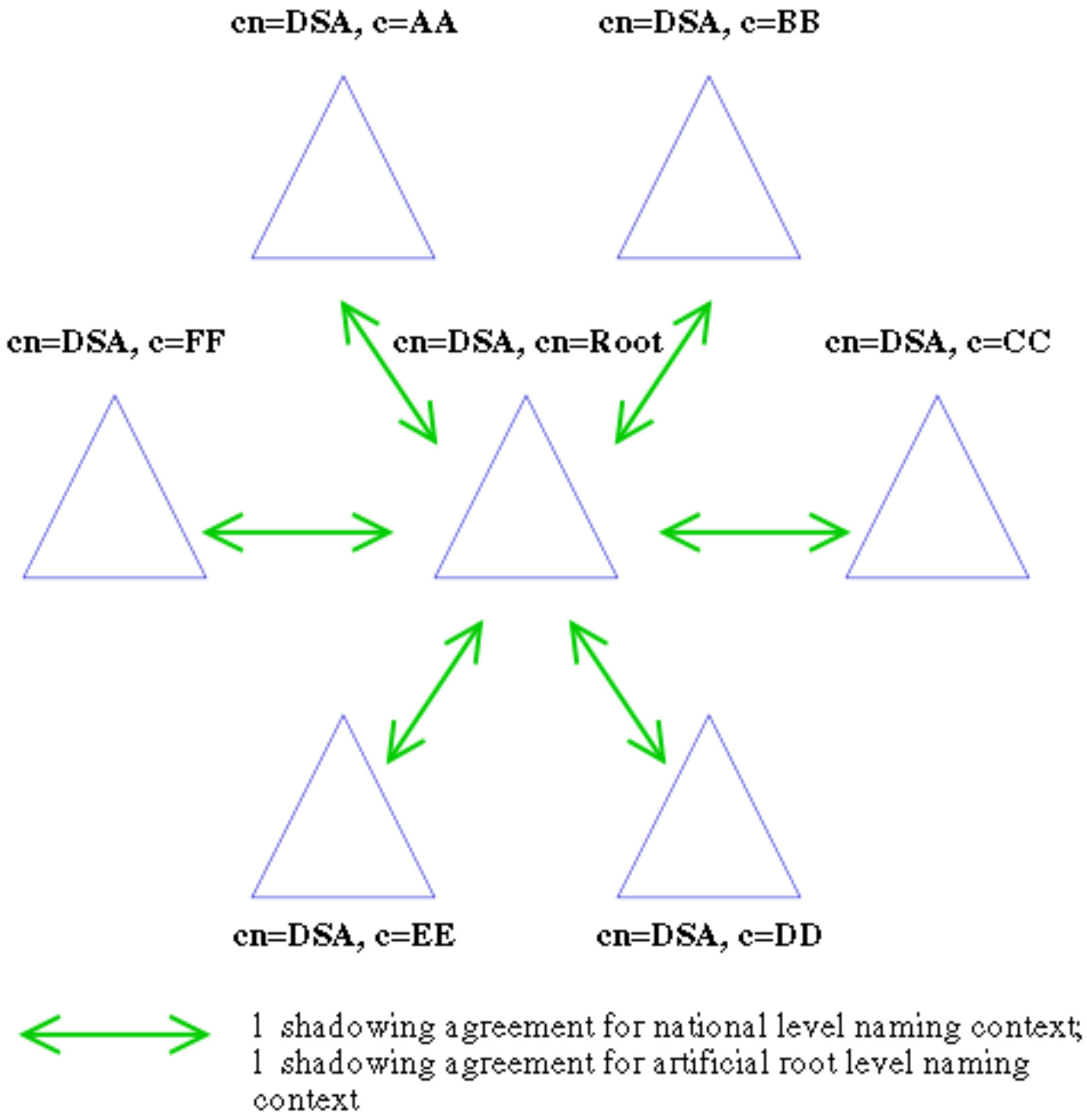

Fig 9. **Optimized star topology for shadowing agreements**

One consequence of the architecture is the loss of naming context and administrative point information for replicated first level entries. This information is being removed while copying entries downwards under <cn=Root>, and created from scratch in a FLDSA after shadowing the artificial root naming context. It is not clear at the moment how seriously it can affect directory operations.

Another issue of the same origin is that every FLDSA becomes a master for all first level naming contexts. This should not be a problem because X.500 does not impose any restrictions on existence of multiple independent directories. The fact that a FLDSA is a master of another national naming context does not matter while it is forbidden to modify this naming context via directory operations. There is no reason to allow anyone doing that, and the actual first level entry manager is expected to modify the entry on the original FLDSA.

6. Implementation

In the proposed architecture, the replication is done via the standard X.500 means: DISP protocol. Scripts operate with local DSA only and do not require any interaction with other DSAs. Therefore, any directory software supporting DISP and having a directory server access API, can be included into the architecture. In practice, only one software product has been tested: **M-Vault** X.500/LDAP directory server of MessagingDirect Ltd. DANTE's one-year contract with MessagingDirect allows all national level directory services, participating in NameFLOW, to use free binary copies of M-Vault server.

The server software package includes TCL language API for accessing both directory data (entries, their attributes and values) and DSA's internal information (naming contexts and knowledge references).

A TCL-based library, DSAflow, was written by Brunel University as a subcontractor of DANTE. Two major features of the software are:

- automatic creation of a DSA on a national server and establishing shadowing agreements with the root server (Tk-based graphical interface);
- copying appropriate naming contexts received as `<*, cn=Root>` from the root server, upwards to the correct positions in the directory information tree (Tcl-based script, could be run periodically from the cron daemon).

7. LDAP interoperability

The replication between X.500 and LDAP servers is not defined in any standards. Therefore, the root directory server should use some special method to copy directory data from national LDAP servers and provide it to other X.500 or LDAP servers.

The LDAP replication model developed at DANTE is based on the following features:

- There are national LDAPv3 servers, and organisational level LDAPv2 and v3 servers. The national servers can themselves hold organisational entries or can contain references to the corresponding organisational servers, and thus are able to provide in their responses either valid LDAP entries or LDAPv3 referrals to other LDAP servers;
- M-Vault X.500 server is able to hold knowledge references to LDAP servers in the same attributes, as references to X.500 servers. In other words, there is no difference between presentation of X.500 and LDAP knowledge information.

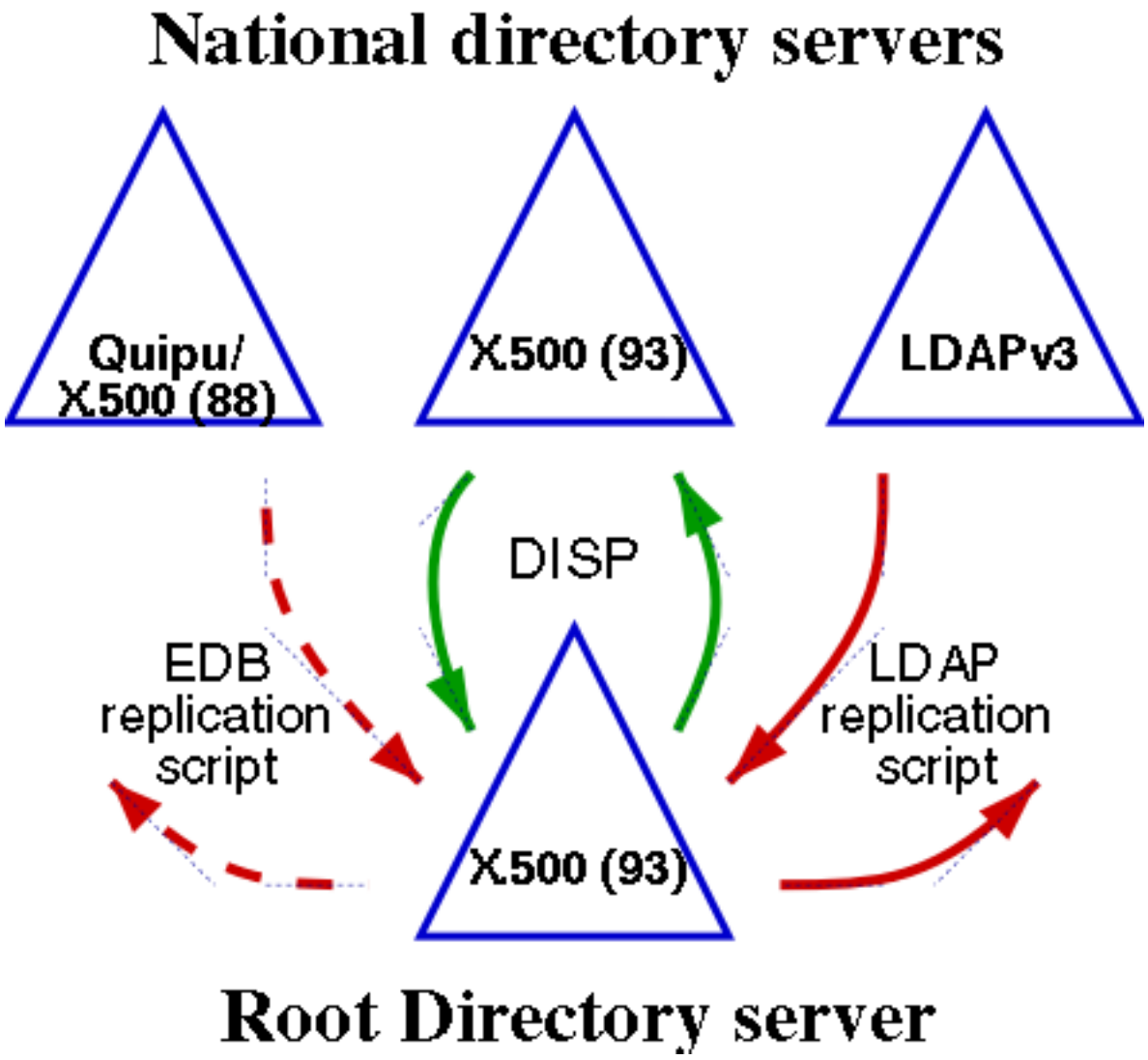

Fig 10. **LDAP interoperability**

A script connects to national LDAPv3 servers and collects the following information:

- the national (country) directory entry (all available attributes and all their values);
- locations of all organisational entries in the given country: the location is either the same national LDAPv3 server or an LDAP server provided in a referral.

The information is then stored in the separate naming context for each country in the root server. The X.500 server is now able to generate DAP/DSP referrals or to do LDAPv2 chaining. LDAPv3 chaining is expected in future versions of M-Vault.

The naming contexts collected from national LDAP servers are available for distribution to other servers in the same way as naming contexts gathered from X.500 servers.

In addition to the above model, the root server provides knowledge information about national LDAP/ X.500 and LDAP-only servers to the DIRECT project. This is done by periodic generation of an LDIF file at the WWW gateway, containing some attributes for every country, including addresses of the corresponding national LDAP servers.

8. Conclusion

Copying national naming contexts into an artificial root naming context allows to avoid the restrictions the X.500 standard imposes on naming context replication.

Scripts copying naming contexts down (in the root DSA) and up (in national DSAs) require access only to the local server, thus eliminating exchange of authentication information between directory managers. All data replication between X.500 DSAs is done via the standard DISP protocol. Although the implementation is based on M-Vault directory server, any other X.500 (93) software having its own server access API can be included to the Directory, if provided with two scripts copying the artificial root naming context up and down.

Data copied to the root DSA from LDAP servers is accessible for replication by FLDSAs in exactly the same way as original X.500 naming contexts shadowed from national servers. Therefore, the whole Directory looks isomorphic from the X.500 directory manager's point of view.

The implementation has been successfully tested between several DSAs in a single management domain. Now tests are being done between the root DSA and several national servers (different management domains). The next step will be deployment of the new architecture by NameFLOW participants.

The replication scripts offered for the X.500 server are not specific to X.500 and are usable for LDAP servers as well, provided the API is changed to access an LDAP server, and some other means are used instead of DISP for naming context replication between directory servers. The knowledge management problem for X.500 and LDAP will need further development, especially as LDAP lacks standard replication and chaining loop avoidance mechanisms and implementations supporting LDAPv3 referrals is still limited.

Konstantin Chuguev, DANTE