Case study: The Global Name Service

Objective:

- Elaborate case study “The Global Name Service”.

Global Name Service (GNS) is designed to facilitate mail addressing, resource allocation and authentication. It consists of naming databases which include names of millions of computers and billions of e-mail addresses of users which can grow from small to large scales with longer life. The structure of a namespace in GNS is dynamic in nature which accommodates changes in names of individuals, organizations and groups. It is intended to operate in a distributed environment where caching is used for rendering the names from a consistent database that has multiple copies of database entries. It also maintains cash consistency where a client can detect and recover from

the use of date naming data. It has a tree of directories that holds names and values. Each directory assigns a unique integer number that serves as a directory identifier. The directories contain a list of names and references. The names in GNS have two parts, namely <directory name, value name> where multiple directories are connected to each other called directory tree and value name are stored at the leaves of a directory called value tree.

As shown in Figure 1, each directory tree is partitioned into multiple subdirectory trees which are stored across several servers with replicated partitions. The root directory is EC with A directory identifier (DI) 688 which is sub partitioned into directories, namely IN and UK with DI 641 and 672, respectively. The value tree is represented by directory, GOV which has values Mailbox and a password for User1, which are followed by alpha, beta and gamma. The user, User1 in the directory GOV is stored in the value tree, namely <EC/IN/AC/GOV, User1>. The value tree includes a password that can be referenced as <EC/IN/AC/GOV, User1/password>, and several mail addresses by <EC/IN/AC/GOV, User1/mailboxes>.

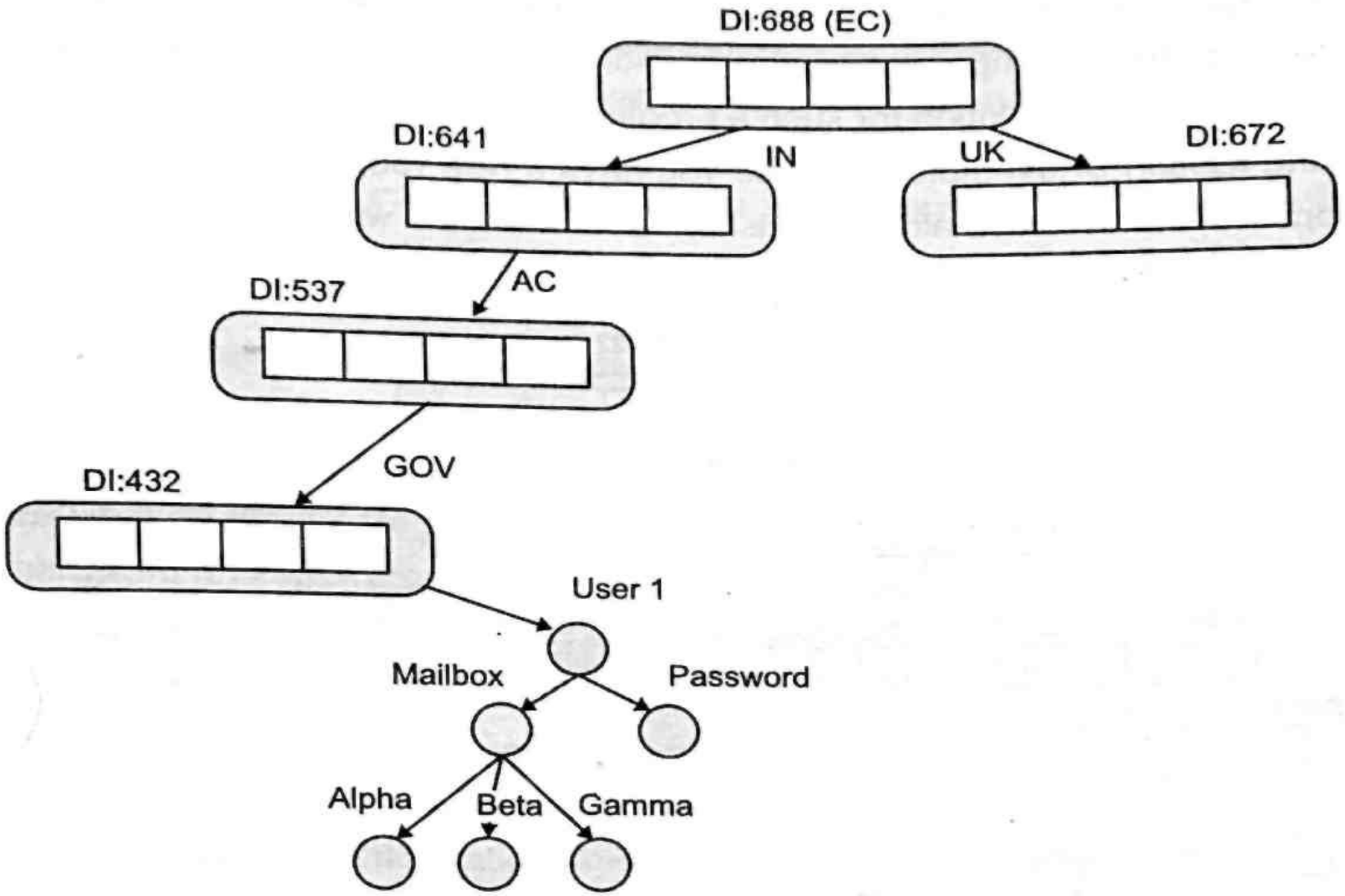

Figure 1. GNS Directory Tree and Value Tree for User 1

The consistency of a directory tree is maintained by two or more concurrent updates where only more than one user is allowed to update the naming database simultaneously. The asynchronous update distribution algorithm is used for maintaining the consistency of replicated directories that ensures eventual consistency. GNS also supports the reform of the database to accommodate organizational changes. The merging of two root nodes of a directory tree is possible by integrating them under a new root node. It also allows users and client programs to continue to use names that are already defined, even when a new real root is inserted. Adding a new root makes an implementation problem at the leaves level as the naming database has millions of directories whose leaf needs to be updated. It provides a solution for such a problem by listing the directories that are used as working roots. For example, suppose you have a new root node World with DI: 722, and suppose you have two already existing root nodes EC with DI 688 and DC with DI 697 as shown in Figure 2.

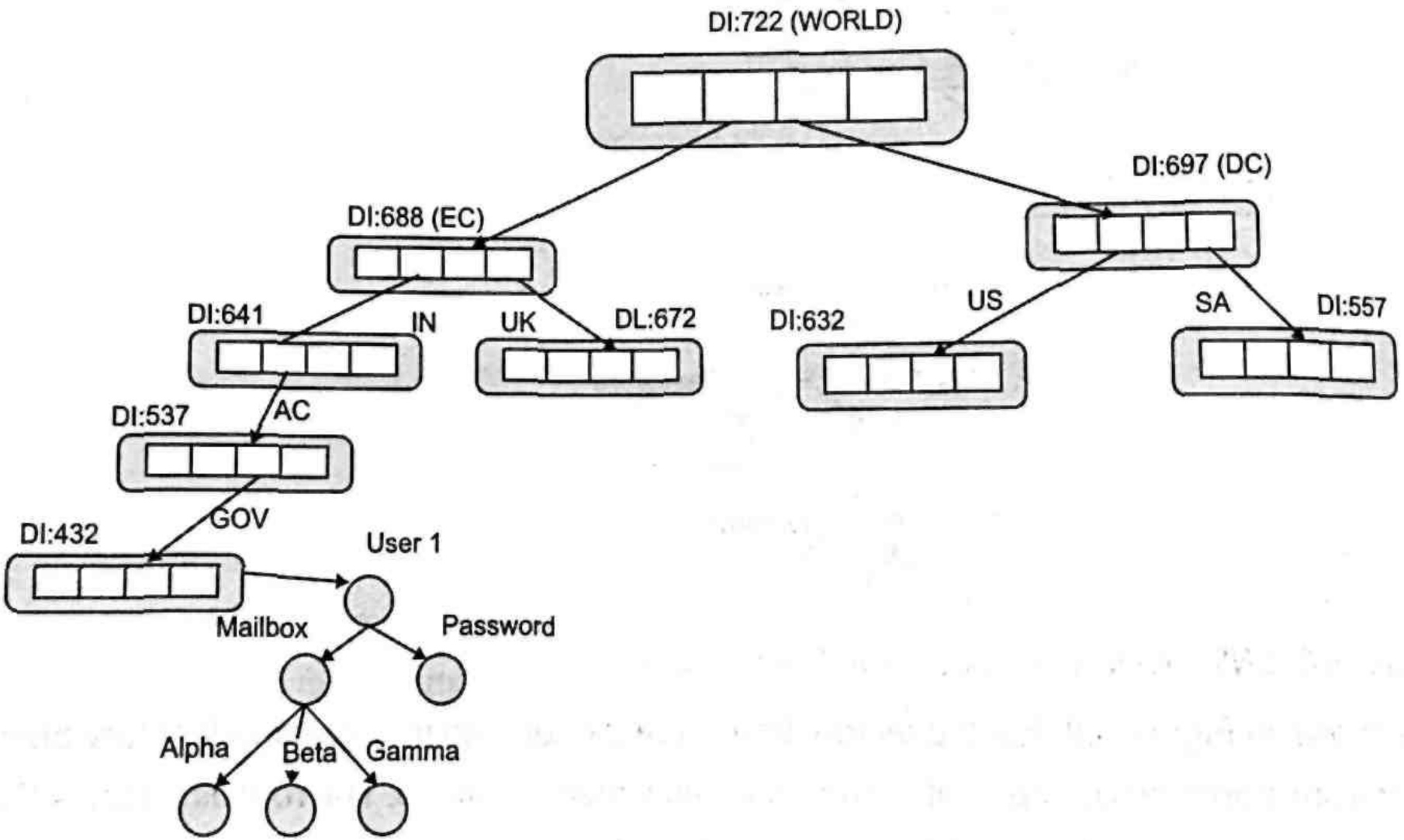

Figure 2. Merging trees under a New Root

Therefore, whenever the real root of the naming database changes all GNS servers are informed about the new location of the real root i.e. #688 = #722/EC and #697 = #722/DC. The nodes can be interpreted with names of the form WORLD/EC/IN/AC/GOV (which referred to the real root) can be interpreted with names of the form #688/IN/AC/GOV that translates them to full pathnames beginning at the real root.

It has advantages such as scalability and re-configurability, which are provided by merging and replications. In a largescale network, re-configurations often happen at any level that can increase the size of the tree that conflicts with the scalability goal.

References:

1] George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems: Concepts and Design", 4th Edition, Pearson Education, 2005.
2] S. Tanenbaum and M. V. Steen, "Distributed Systems: Principles and Paradigms", Second Edition, Prentice Hall, 2006.