Open Software Foundation
Request For Comments: 63.0
October 1994

D. Mackey (OSF)
J. Pato (HP)
DCE 1.2 TPC

DCE 1.2 CONTENTS OVERVIEW

1. INTRODUCTION

This document describes the goals and contents of DCE 1.2. The purpose of the document is to inform the reader of the features of the release and solicit comments on the appropriateness of the direction of the OSF offering. This document serves as one part of the continuing process of factoring membership input into the development of DCE.

DCE 1.2 is the first project to be operated under OSF's pre-structured technology model (PST). In this model, the project is planned, managed, developed, and funded by a group of project sponsors. The sponsors of DCE Release 1.2 are Digital, HP, Hitachi, IBM, and Transarc. These sponsors, a representative of the OSF End User Steering Committee, and a representative of the OSF permanent staff form a Project Steering Committee (PSC) which is responsible for running the program. For more information on PST's and the PST process, see OSF-RFC 62 [Flow 94].

The PSC has designated IBM as the Prime Contractor to take responsibility for managing the development, integration and delivery of DCE 1.2 to OSF.

This document is a product of the DCE 1.2 Technical Planning Committee (TPC), which is comprised of lead architects from each of the sponsoring companies and a representative from the OSF architecture team. The TPC prepared initial technical recommendations for DCE 1.2 content to the PSC and serves as a continuing architectural review body for DCE.

The contents of the release were determined by the PSC after gathering information from a variety of sources: OSF SIG's, OSF staff, the OSF End User Steering Committee, as well as each sponsor company's marketing and engineering departments. The project described here meets the technical requirements gathered by the sponsors, can be delivered approximately one year after the delivery of DCE 1.1, and can be completed within the budget determined by the project sponsors.

1.1. Current Status

At the time of this writing, the DCE 1.2 PST has been approved by the OSF board contingent upon completion of the joint development agreement (JDA) among the sponsors and the final legal agreement between the prime contractor (IBM) and the OSF. The project is now in the development and funding phases, as defined by the PST process (see [Flow 94]).

1.2. OSF Member Feedback

Comments regarding this document should be sent to the authors via the `dce-1.2-comments@osf.org` mailing list and to the appropriate OSF SIG's (e.g., `sig-dce@osf.org`, `sig-security@osf.org`). A presentation and discussion of 1.2 contents is scheduled for the Fall 1994 Super SIG Week (week of November 1, in Boston).

1.3. Document Structure

This rest of this document is organized into three sections. The first describes the motivation and goals of the release. The second section provides an overview of the general project areas and how these projects address the project goals. The final section is a compendium of more detailed descriptions of the individual projects planned for inclusion in DCE 1.2.

2. THE GOAL OF THE RELEASE: EXPANDING DEPLOYMENT

DCE Release 1.2 builds on a series of releases that provide a strong basis for establishing DCE as an enterprise-wide computing solution. Progress in the areas of performance, robustness, scalability, richness of programming tools, and ease of administration have made DCE capable of serving as the infrastructure on which businesses can build a coherent distributed computing environment. DCE 1.2 will continue on that path.

With DCE's growing deployment, the requirements for improvement have become more pointed — reflecting actual experience rather than speculative interest. Users and administrators have identified where improvement is needed and DCE 1.2 will address those areas.

DCE 1.2 focuses on broadening the appeal of DCE rather than on extending the technology for the sake of technology development. It will be a balance of new user-oriented features with basic capabilities which set the stage for more functionality in future releases. As DCE matures, the trend will be one of evolution rather than revolution. DCE 1.2 will protect the user's investment in DCE while continuing to move forward.

The primary goal for DCE 1.2 is to facilitate the continuing trend toward enterprise-wide deployment of DCE in the 1996−1997 timeframe. To reach this goal we need to continue to make DCE more easily manageable, more scalable, easier to program, and more smoothly integrated with the distributed computing technologies that exist alongside DCE in most enterprises.

3. CONTENTS OVERVIEW

This section provides a high level description of the features of DCE 1.2 and how they address the goals of the release. DCE 1.2 provides features for coexisting with other network environments, improving administration, simplifying program development, improving scalability, and continuing the initiatives begun in DCE 1.1.

3.1. Coexistence with Installed Technologies

Few DCE installations are deployed in isolation. Most environments will include a mixture of existing technologies that have some degree of overlap with services provided by DCE. DCE will address this overlap of functionality by allowing access between DCE and the most popular of these environments (e.g., ONC and Netware). DCE will also reduce the administrative overhead associated with managing multiple environments by eliminating the maintenance of duplicate data and integrating administrative operations.

The desktop is becoming the paramount platform in the enterprise. Currently Novell's Netware is the dominant network operating system. ONC is present on many enterprise platforms as well. Therefore, DCE 1.2 provides features that integrate Netware and ONC with DCE's file access and user and group administration.

MIT's Kerberos network authentication environment has gained support in the commercial sector. More and more, users demand that the Kerberos environment provided as part of DCE be compatible with the standard version defined by the IETF. DCE 1.2 will guarantee this wire protocol compatibility.

Public key technology is becoming an increasingly important mechanism in many computing environments and applications. Use of public key certificates which are compatible with those used in Privacy Enhanced Mail and other applications will allow DCE to integrate smoothly into the evolving public key oriented computing environment.

There are environments which impose legal restrictions on the selection and use of encryption technology. DCE 1.2 will provide support for multiple encryption mechanisms and will automatically select the appropriate technique in a heterogeneous environment.

3.2. Ease of Administration

One of the most important aspects of the work done in DCE 1.1 was the improvement of administrative interfaces. DCE 1.2 will continue to improve DCE administration by providing access to DCE configuration information from existing SNMP-based management tools. DCE 1.2 will also define conventions and provide commands that allow more straightforward navigation through the DCE namespace and more intuitive administration.

DCE 1.2 will define how single and multiple cell organizations can be managed and define policies and conventions to make this task as simple as possible. Application of these policies will improve scalability by spreading large databases across multiple systems and organizing resources into logical and more manageable sets. Our aim is to provide more straightforward management of multiple cells while retaining the useful qualities of a single cell system.

DCE 1.2 will define policies for how the namespace is structured and used by applications, services, users, and administrators. This work will use the X/Open Federated Naming (XFN) policies as a starting point to define conventions that apply specifically to DCE. These conventions will help application programmers design server and resource location schemes, users to navigate the namespace, and administrators to glean configuration information by browsing the namespace.

As organizations and their cells change, it will be increasingly necessary to move users and their data from cell to cell in a straightforward, efficient, and secure way. Modifications to the security service will be made and tools developed that make moving accounts easy.

DCE 1.2 will reduce the extensive and sophisticated knowledge needed to perform day-to-day maintenance. Centralized task oriented tools supporting status gathering, backup and restore, startup, and shutdown will be provided. DCE 1.2 will also support commands to determine and control which users are actively accessing resources in a cell.

3.3. Ease of Programming

DCE 1.2 will provide facilities which allow applications to more easily integrate their databases into the DCE naming system. DCE 1.2 will provide a naming API that will allow clients to access naming services in a straight forward manner compliant with the X/Open Federated Naming (XFN) preliminary specification.

DCE 1.2 will lay the groundwork for application servers to participate in the DCE name system at the protocol level. This work is based on the DCE concept of a junction point where an alternate naming system provider may integrate into the DCE name system. DCE 1.2 will complete the specification of this protocol to enable the development of generic tools for browsing and managing complex namespaces.

Application development will be simplified through the support for C++ environments and single threaded client applications.

3.4. Scalability

DCE 1.2 is addressing implementation limits that restrict scaling. While DCE 1.2 is considering the long term implications of scaling to cells with millions of users and machines, short term activities are also being pursued to make sure that cells scale well when dealing with 100,000 users or machines. DCE 1.2 will also define configurations that are tailored to smaller client platforms.

3.5. Internationalization

DCE 1.1 established the baseline by internationalizing large portions of DCE. All new work added to DCE 1.2 will be required to follow the internationalization guidelines defined in DCE 1.1 (e.g., message catalogs must be provided and multi-byte characters must be handled where appropriate).

3.6. Completion of DCE 1.1 Functionality

Some features were not completed in time for inclusion in the integrated portion of the DCE 1.1 release. Many, however, are included in the unintegrated variant of the DCE 1.1 release. DCE 1.2 completes the development, integration and testing needed to bring these features into the mainline of DCE.

4. BRIEF PROJECT DESCRIPTIONS

This is a brief and mostly complete list of DCE 1.2 projects. Additional details will be available during the development of DCE 1.2.

4.1. Integration with Other Environments

4.1.1. Netware coexistence

Novell Netware has established itself as the dominant information sharing technology in small personal computing networks. To establish a coherent enterprise wide computing environment, the large population of Netware users must be integrated into the DCE environment. DCE 1.2 addresses this integration by providing file sharing services and administrative aids that allow Netware 3.X or 4.Y users and DCE users to have a single identity and a single view of the DCE filesystem.

DCE 1.2 will provide access to DFS files through standard API's and commands at Netware clients. DCE 1.2 also unites DCE and Netware user and group administration by providing a set of utilities and a Netware Loadable Module that together enable management and synchronization of user accounts across both systems. These facilities provide a consistent view of users and groups from both DCE and Netware points of view.

In DCE 1.2, Netware clients can define any directory tree in DFS as a Netware volume and use standard Netware or DOS/Windows commands and API's to manipulate them. File access security is maintained through a gateway similar to the NFS gateway delivered as part of DCE 1.1 (see section 4.1.2 below).

In addition to the DFS access, DCE 1.2 will provide a single point of administration for DCE and Netware users and groups. This merging of administration is critical to the establishment a single coherent computing environment. DCE 1.2 will allow users to change their password once affecting all DCE and Netware systems and will allow a single definition of user accounts across both systems.

Netware user information will be stored in the DCE registry in Extended Registry Attributes (a feature described in DCE-RFC 6 [Pato 92] delivered as part of DCE 1.1). Netware server groups will also be defined through utilities built on this mechanism.

Users and administrators can change passwords though DCE or Netware interfaces. The password changes are automatically propagated to all Netware servers having an account for the user. General user and group administration can be accomplished through utilities provided at DCE hosts as well as through the Netware Bindery commands. Access from Netware utilities can be prohibited through configuration commands from DCE.

4.1.2. ONC coexistence

ONC, especially NFS, exists on a large number of machines in many different operating environments. Both DCE and ONC provide mechanisms for managing users in the distributed environment. DCE 1.2 will reduce the administrative burden of running DCE and ONC services in the same environment by providing mechanisms to synchronize the ONC NIS password and group maps with the corresponding information found in the DCE.

An NFS protocol gateway makes NFS client machines instant consumers of the DCE distributed file system. Delivery of the secure NFS protocol gateway [MS 94] was accelerated into DCE 1.1 (in exchange for some security features that were deferred to the DCE 1.2 release). This feature allows NFS clients to access the full DFS space. Client machines that have access to Kerberos V5 provide single-login access to DFS via a utility that sends proxy credentials to the gateway machine. Client machines that do not have access to Kerberos V5 require that either the conventional insecure NFS authentication mechanism be used or that the user establish a login context on the gateway machine. In all cases, access to DFS uses full DCE credentials.

Additional support for DFS host-specific and architecture-specific (`@HOST` and `@SYS`) file naming features will be added in the DCE 1.2 release.

4.1.3. Kerberos V5 support

The DCE Security Service includes an implementation of the MIT Kerberos Version 5 (V5) authentication and key distribution service. Prior to DCE 1.2 there has been no interoperability commitment in the OSF DCE offering.

Recently the Kerberos V5 protocol has become more stable with the release of IETF-RFC 1510 [KN 93] and movement of the Kerberos protocol within the IETF standards process. DCE 1.2 will enhance the high degree of interoperability that existed in previous releases with committed support for the IETF-RFC 1510 protocol. This support will formally allow Kerberos V5 applications running on either DCE or non-DCE platforms to access the DCE Security Service as a full function IETF-RFC 1510 Kerberos server. Prior to the release of DCE 1.2, a specific MIT Kerberos V5 implementation release will be established as the interoperability base.

Many consumers of DCE wish to extend the single login environment. The MIT Kerberos environment has traditionally included network utilities that transmit user account information (e.g., `telnet`, `ftp`, `rlogin`). These utilities are integrated with Kerberos to achieve a single login facility in the networked environment. Earlier DCE releases have not included these utilities in part because IETF standardization has not been completed. DCE 1.2, however, will include implementations of `telnet`, `rlogin` and `ftp` that have been integrated with the DCE environment and will use Kerberos-based protocols to avoid exposing user's passwords on a network. Any modifications to these protocols not yet completed by the IETF will be made available to the IETF by the DCE project team for potential inclusion in the IETF standards.

4.1.4. Public key support

DCE 1.2 will allow public key techniques to be used to support login. This facility will retain full interoperability with existing DCE releases, but will allow a cell administrator to reduce the amount of recovery work performed when the cell's security server has been compromised. With these techniques, the security server need not store the long term key for each principal so that each user's key (or password) will remain undisclosed should any compromise of the security server occur. The system will allow for a hybrid cell where some principals use the pre-1.2 mechanisms while others have access to the public key techniques.

At login, public-key users will receive credentials that allow them to use the current Kerberos-based DCE authentication mechanism. The DCE 1.1 pre-authentication protocol [Pato 93] is used so that the login client need not determine whether a given user is public-key-capable prior to requesting credentials. In addition to login, the password-change protocol will be enhanced to allow users to change either their password or their private-key. Support for X.509 certificates is also being examined.

4.1.5. Multiple security environment support

DCE has always insulated application programmers from the details of the security facilities for authentication, data integrity and confidentiality. These facilities have allowed for multiple simultaneous implementations — but have always supported only one. DCE 1.2 adds an additional cryptographic technique, a user-to-user (peer-to-peer) authentication facility, and a negotiation facility so that applications obtain the best alternative for these choices available to them.

4.1.5.1. Multiple cryptographic algorithm support

This facility allows the use of multiple cryptographic algorithms for RPC communications. DCE 1.2 will now allow selected alternative algorithms to be used. DCE will maintain a registry of these cryptographic algorithms and their uses.

4.1.5.2. Alternative cryptographic algorithms

To validate the runtime support for multiple cryptographic algorithms outlined above, it is necessary to support more than just DES. Minimally, DCE 1.2 will support a ''placebo'' algorithm which does not provide cryptographic protection. Additional cryptographic algorithms may become available in time for DCE 1.2.

4.1.5.3. User-to-user authentication

This facility will enable applications that do not have access to a principal's long term key to be the receivers of a protected RPC. Applications that are clients already have this insulation from access to the user's long-term key. These applications only need access to the login context for the user. Servers, however, need current access to the key.

When the concept of ''server'' is associated with a long running system resource — like the name server or the security server — it seems natural that the server is running on a machine with access to the long term key to the identity of that server. If for no other reason, the server is not normally associated with a human user but rather with a pseudo-user corresponding to the system service. This does not, however, map well onto all application domains. In particular, some applications need a ''peer-to-peer'' model. This model is more fully described as requirement D3 of DCE-RFC 8.1 [Blak 94].

The user-to-user authentication facility will provide an alternate Ticket Granting Service (TGS) protocol as defined in IETF-RFC 1510. This will provide server applications with the same sort of insulation from a principal's long term key that is available for client applications. In particular it will be possible to direct a protected RPC to an application that has access to the login context for its principal rather than just access to the long-term key for that principal.

4.1.5.4. Security context negotiations

This facility provides a secure selection of dynamic security parameters. For example, this will allow an application client to determine the protection level, cryptographic algorithm and which variant of the TGS protocol should be used when communicating with a specific application server.

4.2. Ease of Administration

4.2.1. SNMP support

This project allows existing management applications, typically using graphical user interfaces, to query the status of DCE systems without adding specialized DCE knowledge to the applications. This is accomplished through the standards-based definition of managed objects and the implementation of an agent which accepts requests via the Simple Network Management Protocol v2 (SNMP) and translates them to DCE-specific operations.

4.2.1.1. DCE managed object definitions

The DCE managed object definitions provide a single, consistent, extendable object model which exposes the administrative and management characteristics of DCE and its core services.

Specification of DCE managed objects are represented in a generic object definition notation which defines object classes, containment and inheritance relationships, actions, attributes, and event notifications. Each object class encapsulates all administrative and management related functional tasks.

This work is the replacement for and update to DCE-RFC 38.0 [Sear]. In addition, the object class inventory and contents have been closely reviewed in the context of the DCE 1.1 Control Program (`dcecp`) objects and actions, described in DCE-RFC 42 [Melm 94], to assure as much consistency as possible.

4.2.1.2. DCE SNMP MIB

The DCE SNMP MIB is directly derived from the DCE Managed Object definitions as documented in the working draft of DCE-RFC 38.1 [Wils]. The DCE SNMP MIB specifies the objects and variables (attributes) that can be read or manipulated while managing the DCE core services (including RPC, Time, Security, Cell Directory), and the Distributed File System (DFS).

With the exception of threshold variables, the MIB will be limited to read-only (`GET` and `GET-NEXT`) type variables due to the limited and incompatible SNMP security model.

4.2.1.3. SNMP agent

An agent or daemon will be provided in DCE 1.2 which accepts SNMP operations (queries), performs the requested operation on DCE managed entities, and returns the resulting information. The agent implements the operations described in the SNMP MIB.

4.2.2. Hierarchical namespace policies

To aid users, administrators, and developers in dealing with the DCE namespace, a set of policies that describe how resources should be named will be delivered. In addition policies will be enforced through `dcecp` scripts, commands and application programming interfaces which automate the installation of and search for resources in the DCE namespace.

This project will deliver a guide that users, administrators, and application programmers can follow when populating or perusing the namespace. The scripts, commands and API's will follow the guidelines outlined in the document.

The policies will be consistent with the requirements of the existing DCE API's and commands and will provide support for the policies defined as part of revision 2 of the X/Open XFN preliminary specifications.

4.2.3. Active user monitoring

This facility will track the users that log into machines in the cell or access resources in the cell. An agent running on each machine (a part of `dced` — described in DCE-RFC 47 [BMSW 94]) will be used to allow the centralized management of active users — including the ability to terminate user sessions from a central administration station.

In DCE 1.2 this facility will be built into the Security Server's Key Distribution Center (KDC). When a Ticket-Granting-Ticket (TGT) is issued — including when a TGT is issued based on public key authentication (see section 4.1.4 above), the active user database is updated. This database maintains information about the cell, and source machine for each active user. Entries in the active user database expire automatically at the termination of the TGT lifetime. The entries are also removed (listed as logged out) when a user's machine (`dced`) notifies the security service that the user has logged out.

This kind of monitoring provides important ease-of-administration and scalability features to the DCE. On multi-user systems, administrators are accustomed to determine who is actively using the system and in some cases the ability to terminate sessions. When converting from a single machine environment to the distributed environment of the DCE, administrators have lost this ability. This service will restore the feature.

The API to this service will also simplify the binding problem for new user-agent applications. For example, applications will use the API to obtain active user location information. This information can then be used as the binding information for connecting to the active user's calendar utility — or for creating an online conversation with the user.

4.2.4. DFS admin lists as ACL's

DFS uses administration lists to specify authorization for various administrative operations, such as fileset creation, movement and replication, backup and restoring of filesets, and modifying quotas. Although the administrative lists used in these commands follow the syntax and semantics of DCE access control lists (ACL's), the administration lists do not appear on any observable object in the DCE controlled name space. Hence the mechanism for observing or specifying administration lists for DFS administrative operations are commands unique to DFS.

DCE 1.2 will define a place in the DCE-controlled name space on which administrative lists can be represented as ACL's. The administrative operations within DFS will use both the previous mechanism and the new mechanism to determine authorization for performing operations. Synchronization of the two methods will be provided to ensure that consistent authorization is obtained.

4.2.5. Multi-cell groups

DCE 1.2 will allow principals from a foreign cell to be added to a group in the local cell. The privileges granted to that principal will reflected in the EPAC for that principal when she accesses resources in the local cell.

4.3. Ease of Programming

4.3.1. Single threaded RPC

In DCE 1.2 RPC client applications will no longer have a dependency on a threaded environment. This simplifies the task of application developers, eliminating the need for thread-aware programmers, debuggers and support libraries.

Many existing applications that would benefit from deployment in a distributed environment have not been written to be thread-aware and don't run properly in a threaded environment. The existence of the thread-free version of RPC in DCE makes it substantially easier for these applications to be adapted to the DCE.

Some of the techniques for achieving this work item have been described in DCE-RFC 31 [Karu 92]. Additional work has been defined to remove the dependency on threading from the rest of the DCE client runtime.

4.3.2. IDL C++ support

DCE 1.2 will add C++ support to IDL. This support allows client and server programs written in C++ to utilize DCE RPC in a highly transparent manner using natural C++ constructs. The IDL language is extended to support C++ features such as inheritance and object references. One can think of an IDL interface definition as defining a class and its methods. When an IDL interface is compiled with the C++ option, the IDL compiler generates a C++ header file containing an abstract class definition for the interface, and generates C++ client and server stubs.

A client utilizing these features can create remote objects or look up existing remote objects by invoking the static member functions of the interface. Once an object reference is obtained, any of its member functions can be invoked. In both cases, the syntax and usage of the object is the same in the client program as it would be for local objects. In fact, local and remote objects can both be used in the application, with no client code differences.

The IDL C++ support provides language enhancements and related runtime functions for applications to manage distributed objects. Application developers can use the C++ support as is, or use it as the underlying framework for an object model at a higher level of abstraction. The C++ support does not ''lock you in'' to any one object model, leaving that decision to the application developer.

This project has been described in more detail in DCE-RFC 48 [AHV 94].

4.3.3. X/Open Federated Naming

The X/Open Federated Naming (XFN) preliminary specification defines an API for building portable applications which can access a wide variety of naming systems. XFN also defines a client framework that allows naming clients of diverse naming services (e.g., X.500, CDS, NIS, DCE Registry) to be integrated under the common API. Each nameservice integrated into the framework provides a ''context'' in XFN parlance. An implementation of a client module in the framework is called a context implementation.

The API and a set of context implementations combine to provide applications a view of a single namespace unifying the namespaces provided by each of the services. Finally, XFN introduces a new naming service protocol defined in DCE RPC which enables application servers to provide and manage a part of the namespace as a first class name server.

DCE 1.2 will provide the XFN API layered atop a CDS context implementation. This API provides a more complete and more intuitive programming interface than those currently available in DCE. The XFN API allows application programmers to write programs that are portable across both DCE and non-DCE XFN environments. These programs will be able to take advantage of future enhancements to XFN services such as support for additional naming contexts or smaller, higher performance client implementations.

4.4. Scalability

4.4.1. Common definition for small clients

This project provides guidelines that describe how to provide and configure a ''slim'' DCE client. The primary benefit will be small hardware configurations with the capability of being a DCE client without subsetting functionality. The hardware configuration targeted is an Intel 486 class of machine with eight megabytes of memory. However, this work is applicable to other types of hardware and configurations. Our target is to run an application client on this class machine without excessively annoying the user (minimal paging). The working set size of this slim DCE client should be under one megabyte memory.

Only those DCE daemons that are absolutely necessary for a DCE client application to function will be run. This will be addressed mainly through configuration options that a user can select at installation time. To further slim down the DCE client, a set of source code changes will be described for the DCE source code licensee.

Only the core services clients will be addressed. DFS clients are beyond the scope of this project.

4.4.2. Improvements to CDS

The changes to CDS address the robustness of the CDS server. These changes will significantly reduce the use of virtual memory at a server that is receiving many updates. The user will benefit from a more stable and reliable service. The changes for DCE 1.2 will include replacing the data storage module with a new technology, reducing the duplication of information at a server, and more aggressive purging of stale data.

DCE 1.2 will also supply a salvager tool that will replace current tools in the unsupported tree. The utility will resemble the interface and function of the security salvager. The tool will dump the database to a file, check for and fix detectable corruption and allow an administrator to explicitly remove data.

4.4.3. Improvements to security

Simple changes to the security server can be made to obtain considerable performance improvements when dealing with large cells (where there are more than 50,000 principals). These changes include documenting the configurable checkpoint interval; partitioning internal datasets so that the amount of data written to disk during a checkpoint is proportional to the amount of data modified; augmenting the current propagation interfaces with batch mode interfaces that allow the transmission of multiple events in one RPC.

4.4.4. Improvements to DFS

4.4.4.1. Local file system (LFS) vnode management

This project improves the way that LFS handles vnodes. These improvements will allow LFS to perform significantly better as the system is subjected to higher levels of stress. In particular, finer grain locking will operate more efficiently at high, concurrent administrative and user loads, and will operate more efficiently on multiprocessor machines.

4.4.4.2. Optimized token manager

Improvements to the token manager will decrease the memory requirements and improve performance and reliability of DFS. The resulting token manager should be about half the size and offer a factor of six improvement in performance of token related operations.

4.4.4.3. Stable DFS replication

A fundamental principle of DCE is that services should be replicated for high availability and enhanced scalability. The DFS architecture provides replicated filesets as a way to implement this.

This project improves the DFS replication implementation to achieve greater reliability and better performance. The primary benefits of the project is to users and administrators. Users will get the benefit of a system which has higher availability and better scalability. Administrators will be able to reorganize how data are located and replicated.

4.4.4.4. Concurrent retrieval of different (read/write) filesets

DCE 1.2 will convert the replication server from a serial server to a concurrent server. The existing DFS replication server accepts updates to replicas serially, that is, if several replicated filesets are being retrieved from a collection of servers, the changes from the separate (read/write) filesets are retrieved one after the other. In a large system where many filesets are being propagated, this algorithm represents a bottleneck to scalability.

4.4.4.5. DFS server preferences

Currently DFS makes an arbitrary choices of a file server which could cause poor performance and reduce the ability to scale in a WAN environment. DCE 1.2 will add server preferences to the DFS client. This feature will allow DFS to make intelligent choices about which servers to use for different filesets.

5. ACKNOWLEDGMENTS

We would like to thank the members of the DCE 1.2 TPC for their efforts in defining the contents of DCE 1.2. The TPC consists of Joe Comuzzi (Digital), Joe Pato (HP), Hiroshi Yagi (Hitachi), Ivan Milman (IBM), Ellen Stokes (IBM), Dick Mackey (OSF), and Mark Sherman (Transarc). In addition each company provided a number of key individuals to support the efforts of the primary TPC member.

REFERENCES

[AHC 94] R. Annicchiarico, J. Harrow, R. Viveney, ‘‘C++ Support in DCE RPC -- Functional Overview’’, DCE-RFC 48.1, April 1994.

[BMSW 94] J. Bowe, D. Mackey, R. Salz, P. Wang, ‘‘DCED: The DCE Host Daemon -- Functional Specification’’, DCE-RFC 47.3, April 1994.

[Blak 94] B. Blakley, ‘‘DCE SIG Security Requirements’’, DCE-RFC 8.1, Date TBD.

[Flow 94] K. Flowers, ‘‘The PST Process: A Practical Guide’’, OSF-RFC 62.0, Date TBD.

[Karu 92] M. Karuzis, ‘‘Supporting Threadless DCE Clients’’, DCE-RFC 31.0, December 1992.

[KN 93] J. Kohl, C. Neuman, ‘‘The Kerberos Network Authentication Service (V5)’’, IETF Network Working Group Request for Comments 1510, September 1993.

[Melm 94] H. Melman, ‘‘DCECP Functional Specification’’, DCE-RFC 42.2, March 1994.

[MS 94] T. Mistretta, W. Sommerfeld, ‘‘Remote Kerberos Authentication for Distributed File Systems: As Applied to a DCE DFS-to-NFS File System Translator’’, in proceedings of the PSRG Workshop on Network and Distributed Security, February 1994.

[Pato 92] J. Pato, ‘‘A Generic Interface for Extended Registry Attributes’’, DCE-RFC 6.0, June 1992.

[Pato 93] J. Pato, ‘‘Using Pre-Authentication to Avoid Password-Guessing Attacks’’, DCE-RFC 26.0, June 1993.

[Sear] B. Searle, ‘‘DCE Managed Objects’’, DCE-RFC 38.0, Date TBD.

[Wils] G. Wilson, ‘‘DCE SNMP Mib’’, DCE-RFC 38.1, Date TBD.

AUTHORS' ADDRESSES

Richard Mackey
Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142
USA

Internet email: dmackey@osf.org
Telephone: +1-617-621-8924

Joseph N. Pato
Hewlett-Packard Co.
300 Apollo Drive
Chelmsford, MA 01824
USA

Internet email: pato@ch.hp.com
Telephone: +1-508-436-4350