Open Software Foundation
Request For Comments: 63.3
R. Salz (OSF)
October 1996

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 project and serve as 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 (PST) model. In this model the project is planned, managed, developed, and funded by a group of project sponsors. The sponsors of the DCE 1.2 PST are Digital, HP, Hitachi, and IBM. These sponsors, two representatives of the Open Group Customer Council (formerly 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.

This document is a product of the DCE 1.2 PSC. It is derived from information provided by the DCE Technical Planning Committee (TPC), which is comprised of lead architects from each of the sponsoring companies and representatives from the OSF. The TPC also serves as a continuing architectural review body for DCE.

The contents of the project were determined by the PSC after gathering information from a variety of sources: OSF SIG's, OSF staff, the Open Group Customer Council, as well as each sponsor company's marketing and engineering departments. The project described here meets the technical requirements gathered by the sponsors, will be delivered in two releases approximately 18 and 24 months after the delivery of DCE 1.1, and can be completed within the budget set by the sponsors.

Comments regarding this document should be sent to the `dce-1.2-comments@osf.org` mail alias and to the appropriate OSF SIG's (e.g., `sig-dce@osf.org`, `sig-security@osf.org`).

1.1. Document Structure

The remainder of this document is divided into four sections. Section two describes the major goal and motivation of the project. Section three describes the technical contents of the project. Section four describes the current state of the project and the planned contents of each release.

Functional specifications of the technology deliverables have been published as OSF RFC's and are referred-to below. See [RFC Index] for details.

This document represents the efforts of the DCE 1.2 Project Steering Committee, the DCE Technical Planning Committee, and staff at the sponsor companies and OSF. There are too many individuals to name them all, but ‘‘you know who you are’’.

2. THE MAJOR GOAL OF DCE 1.2: EXPANDING DEPLOYMENT

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 scalable, easier to program, more easily manageable, and more smoothly integrated with the pre-existing distributed computing technologies that exist alongside DCE in many enterprises.

DCE 1.2 builds upon a series of releases that provide a strong basis for establishing DCE as an enterprise-wide distributed 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 upon which businesses can build a coherent distributed computing environment using heterogeneous systems. DCE 1.2 will continue on that path.

With DCE's increasing deployment, the requirements for improvement have become more focused, reflecting actual experience rather than speculative interest. Users and administrators have identified where improvements are 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 and basic capabilities which set the stage for future functionality. 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.

3. TECHNOLOGY DESCRIPTIONS

The technology in this PST can be divided into four major areas: integration with other environments, ease of administration, ease of programming, and scalability.

Each section below starts with an overview of the area. Each overview contains a short summary of the issues being addressed and usually contains a set of one-sentence descriptions of how DCE 1.2 will address them. Following the summary, there are more detailed sub-sections explaining each technology deliverable.

3.1. Integration with Other Environments

Few DCE installations are deployed in isolation: most environments include a mixture of existing technologies that have some degree of overlap with services provided by DCE. DCE 1.2 will address this by allowing access between DCE and the most popular of these environments.

The desktop is becoming the paramount platform in the enterprise. Currently Novell's Netware is the dominant network operating system. Sun's Open Network Computing system is present on many enterprise platforms as well. Therefore, DCE 1.2 provides features to help integrate Netware and ONC with DCE facilities.

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 and provide interoperable utility programs.

Public key technology is becoming an increasingly important mechanism in many computing environments and applications. There are a few short-term issues that DCE should address. Many enterprises need login technology that does not require secret keys. Tools that support use of public key certificates and software storage of private keys will allow DCE to migrate smoothly into the evolving public key oriented computing environment.

The DCE security runtime supports the RPC client-server programming model. DCE 1.2 will provide peer-to-peer authentication. This allows other communication models such as message-oriented middleware to use the DCE security infrastructure.

3.1.1. 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 3.12 users and DCE users to have a single identity and access to the DCE filesystem, DFS.

In DCE 1.2, Netware clients can define any directory tree in DFS as a Netware volume and use standard Netware or DOS commands and API's to manipulate the files and directories under it. File access security is maintained through a gateway similar to the NFS gateway delivered as part of DCE 1.1. By adding data to the user object maintained in the Netware Bindery, DCE 1.2 can map a Netware user into an authenticated DCE principal. The gateway will map identities, filenames, and permissions across the two domains.

For details (including a list of software versions supported), see [RFC 96].

3.1.2. ONC coexistence

The secure NFS protocol gateway planned for DCE 1.2 but accelerated into the DCE 1.1 release makes NFS client machines instant consumers of the DCE distributed file system. Additional support for DFS host-specific and architecture-specific (`@HOST` and `@SYS`) file naming features will be added.

For details, see [RFC 84].

3.1.3. 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.

The Kerberos V5 protocol has become more stable with the release of IETF-RFC 1510 [Kohl] and its movement through 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 server as a full-function IETF-RFC 1510 Kerberos server. The DCE security server's implementation will be tested against MIT Kerberos Version 5, releases beta 4 and beta 5.

Many consumers of DCE wish to extend the single login environment. The MIT Kerberos releases include network utilities that transmit user account information (e.g., `telnet`, `rlogin`, and `ftp`). 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 `rlogin` and `rsh` that will use DCE's Kerberos facilities to avoid exposing passwords on a network.

For details, see [RFC 92].

3.1.4. Public key support

DCE 1.2 will allow public key technology to be used to support login. With this technology, the security server need not store the long term key (or password) for a principal so that it will remain undisclosed should any compromise of the security server occur. Administrators can specify that some principals may use the pre-DCE 1.2 mechanisms while others have access to the public key mechanism; it will retain full interoperability with existing DCE releases.

At login, public key users will receive credentials that allow them to use the current Kerberos-based DCE authentication mechanism. A new pre-authentication protocol is used; further, the login client need not determine whether a given user is public-key-capable prior to requesting credentials.

As a transition aide, a new ''keystore server'' will be provided. This server will store private keys for users or sites without access to hardware-based cryptographic tokens, secure filesystem storage, etc.

A new certification API is also being provided. This facility handles the mapping of a principal name to a public key, allowing programmers to hide the details of their own Certificate Authority (CA) access methods and trust model. By letting developers ''plug in'' their own policy and storage modules, this facility continues the DCE practice of providing widespread foundation without dictating a single use model.

For details, see [RFC 68], [RFC 80], and [RFC 94].

3.1.5. User-to-user authentication

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 then that 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, such as is described as requirement D3 of [RFC 8].

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 a program that only has a login context, and no key table (file) or other access to a long-term key.

For details, see [RFC 91].

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 this trend with additional work on `dcecp`, the DCE control program. This release will enhance inter-cell cooperation by allowing groups to contain foreign principals.

3.2.1. dcecp Work

The `dcecp` program provided in DCE 1.1 took an important first step in unifying the DCE management clients. It provides an extensible, scriptable programming environment that allows management tasks to be developed and run across a wide variety of DCE hosts. Its consistency and orthogonality will also make it easier for DCE administrators to understand what they need to do in order to accomplish a specific task.

DCE 1.2 continues this work by ‘‘finishing off’’ `dcecp` so that it implements all functionality of the current control programs. All day-to-day management of DCE can now be done through this single user interface. The old programs, and their documentation, will be retained in DCE 1.2 for compatibility.

3.2.2. Global groups

DCE 1.2 will allow principals from a foreign cell to be added to a group in the local cell. For example, this will allow cross-cell cooperation among DCE time services. It should also ease enterprise-wide security administration, cell reconfiguration, and other management tasks.

For details, see [RFC 87].

3.3. Ease of Programming

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

3.3.1. 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 will be extended to support C++ features such as inheritance and object references.

A client using 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. 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 as the underlying framework for an object model at a higher level of abstraction. The C++ support does not force adoption of any one object model or class hierarchy, providing the developer with much flexibility.

For more details, see [RFC 48].

3.3.2. Single-threaded RPC

Many existing applications that would benefit from deployment in a distributed environment have not been written to be thread-aware and do not run properly in a threaded environment. In addition, developers often use third-party packages, which are themselves not thread-aware. The existence of a thread-free version of DCE RPC would increase software reuse by making it substantially easier for these applications to be adapted to the DCE.

In DCE 1.2 datagram RPC clients 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.

Some of the techniques for achieving this work item have been described in [RFC 31]. The client side of the RPC NSI has also had its dependency on threads removed.

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 hosts, short term activities are also being pursued to make sure that cells scale well when dealing with 100,000 users, hosts, or servers.

3.4.1. Improvements to security

Simple changes to the security server will be made to obtain considerable performance improvements when dealing with large cells (those with more than 50,000 principals). This includes documenting the configurable checkpoint interval and partitioning internal datasets so that the amount of data written to disk during a checkpoint is proportional to the amount of data modified.

In addition, DCE 1.2 will identify bottlenecks and places where resource consumption could be reduced and address them as possible.

For details, see [RFC 93].

3.4.2. Improvements to DFS

There are a number of areas where DFS performance will be improved and made significantly more stable.

3.4.2.1. 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.

For details, see [RFC 73].

3.4.2.2. DFS server preferences

Currently DFS makes an arbitrary choice of a file server which could cause poor performance and reduce the ability to scale in a WAN environment. DCE 1.2 will allow an administrator to identify server preferences on a per-fileset basis. Default preferences will be based on IP subnet numbers. This feature will allow DFS clients to make intelligent choices about which servers to use for different filesets.

For details, see [RFC 74].

3.4.2.3. Vnode/VM management

DCE 1.2 will improve the way that LFS handles vnodes. These improvements will allow DFS 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. In addition, an improved interface between LFS and the native virtual memory subsystem should simplify I/O access patterns and code paths.

For details, see [RFC 75] and [RFC 78].

3.4.2.4. Replication enhancements

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.

Planned enhancements improve the DFS replication implementation to achieve greater reliability and better performance. 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.

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: 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.

For details, see [RFC 76].

3.4.2.5. Multi-home support

DFS currently requires all clients and servers to be reachable via all network interfaces. DCE 1.2 will enhance DFS servers to perform better on hosts connected through multiple interfaces to multiple networks.

For details, see [RFC 77].

3.4.3. DFS Bulk RPC Operations

Common operations such as directory browsing currently result in one RPC operation for every directory entry. The DFS protocols will be enhanced to support a new ‘‘bulk status’’ operation. For small directories (where the definition of ‘‘small’’ should incorporate most ordinary working-user directories), the bulk status RPC browses the entire directory in one RPC. While this does not reduce the amount of work done by the server, the reduction in RPC activity may improve the throughput of directory browsing by as much as 50%.

For details, see [RFC 89].

3.4.3.1. DFS support for jukebox backup

The current DFS backup facility requires an operator to manually respond to queries from the device controller by performing actions such as loading tapes. Modern equipment using stackers and the like auto-load media and can use default responses.

The DFS backup tape coordinator, `butc`, will be able to read a new automatic operation configuration file. This file will specify scripts to be executed automatically when the DFS backup system needs to change media, validate device names, and so on. This will make it possible to do unattended backup of large-scale DFS file systems.

For details, see [RFC 88].

3.4.3.2. DFS backup performance enhancements

A number of performance bottlenecks have been identified in the behavior of DFS backup, dumps, and restores. The following enhancements will be made:

(a) A ‘‘bulk update’’ operation will be added so that the master sync site can send sets of updates to each secondary server in a single RPC.

(b) The master sync site will update secondary servers asynchronously, and will work on the subsequent batch of changes in parallel with the RPC calls being performed.

(c) Database transactions will not be committed asynchronously, so all servers will do commit processing in parallel.

(d) A new RPC operation will be added so that multiple volumes can be added to the backup datase in a single RPC call.

3.4.3.3. Support for 64-bit filesystems

As 64-bit architectures become popular, there is growing desire to support very large files and filesystems. DCE 1.2 will support such entities while maintaining interoperability with the current widespread 32-bit machines.

For details, see [RFC 51].

3.4.3.4. Use of protected RPC

DFS currently uses a range of DCE protection levels on its RPC operations, ranging from unauthenticated for peer `repserver` processes to packet-integrity or higher for the management clients. DCE 1.2 will allow an administrator to specify a range of DCE protections that can be used for most client-server communication. All architectural uses of unauthenticated RPCs will be eliminated.

The new administrative controls allow administrators to distinguish between same-cell communication from intercell communication, so that a DFS Cache Manager will use one set of protection rules for intra-cell use (presumably protected behind a network firewall), while using another set for data-sharing outside the cell. Command line arguments and management clients will allow administrators to achieve the right balance of protection and computational overhead.

For details, see [RFC 90]. This work is a late addition to the DCE PST and is in direct response to recognition of weaknesses first noticed in October 1995.

3.5. Documentation

SGML is an industry standard for representing documentation that is intended to be viewed in a variety of formats, encompassing printed matter and on-line ‘‘hypertext’’ viewing. In DCE 1.2 all documentation will be distributed as SGML source, using the DocBook Document Type Definition.

4. RELEASE CONTENTS

As stated above, the DCE 1.2 PST will be made available in two releases: DCE 1.2.1 in March 1996, and DCE 1.2.2 in December 1996. The contents are as follows:

(a) DCE 1.2.1:
- (i) Netware coexistence.
- (ii) NFS gateway enhancements.
- (iii) Dcecp work.
- (iv) IDL C++ support.
- (v) Optimized token manager.
- (vi) DFS server preferences.
- (vii) Vnode/VM management.
- (viii) DFS replication enhancements.
- (ix) DFS Bulk RPC Operations.
- (x) DFS support for jukebox backup.

(b) DCE 1.2.2:
- (i) Kerberos V5 support.
- (ii) Public key support.
- (iii) User-to-user authentication.
- (iv) Global groups.
- (v) Single-threaded RPC.
- (vi) Scalability improvements to security.
- (vii) DFS server multi-home support.
- (viii) DFS backup performance enhancements.
- (ix) Support for 64-bit filesystems.
- (x) Use of protected RPC.
- (xi) Documentation in SGML.

There will be an Early Access program so that DCE 1.2 licensees can have snapshot access to the project source.

REFERENCES

Note that some of the OSF-RFC’s below may be revised after the present one is published; further, some relevant new RFC’s may be published. It is incumbent on the consumer of this RFC to track these developments.

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

[RFC 8] B. Blakley, ‘‘DCE SIG Security Requirements’’, DCE-RFC 8.1, October 1995.

[RFC 31] M. Karuzis, ‘‘Supporting Threadless DCE Clients’’, DCE-RFC 31.0, December 1992.

[RFC 48] R. Viveney, ‘‘C++ Support in DCE RPC IDL — Functional Specification’’, OSF-RFC 48.3, April 1996.

[RFC 51] S. Strange, ‘‘A 32-Bit/64-Bit Interoperability Solution for DFS’’, DCE-RFC 51.2, June 1995.

[RFC 68] A. Anderson, S. Cuti, ‘‘Public-Key Login for DCE 1.2 — Functional Specification’’, OSF-RFC 68.2, February 1996.

[RFC 73] R. Agarwalla, ‘‘DFS Token Manager Redesign’’, OSF-RFC 73.0, October 1995.

[RFC 74] S. Berman, ‘‘DFS Server Preferences’’, OSF-RFC 74.0, May 1995.

[RFC 75] B. Lewis, ‘‘Episode VM Integration’’, OSF-RFC 75.0, October 1995.

[RFC 76] D. Clevenger, ‘‘Multi-Threading the Replication Server’’, OSF-RFC 76.0, May 1995.

[RFC 77] S. Moyer, ‘‘Supporting Multi-homed DFS Servers’’, OSF-RFC 77.0, January 1996

[RFC 78] T. Anderson, ‘‘Episode Vnode Synchronization’’, OSF-RFC 78.0, January 1996.

[RFC 80] J. Wray, ‘‘DCE 1.2 Certification API — Functional Specification’’, OSF-RFC 80.0, February 1995.

[RFC 84] J. Brezak, ‘‘DFS Gateway Support of @sys and @host per NFS Client Mappings’’, OSF-RFC 84.0, July 1995.

[RFC 87] H. Yu, M. Burati, ‘‘DCE Global Groups — Functional Specification’’, OSF-RFC 87.0, December 1995.

[RFC 88] J. Gait, A. Khale, J. Morin, ‘‘A Jukebox Backup Subsystem for DFS’’, OSF-RFC 88.0, December 1995.

[RFC 90] C. Everhart, ‘‘Security Enhancements for DCE DFS’’, OSF-RFC 90.0, February 1996.

[RFC 89] J. Gait, ‘‘A Bulk Status RPC for DFS’’, OSF-RFC 89.0, December 1995.

[RFC 91] M. Burati, J. Pato, ‘‘User-to-User Authentication — Functional Specification’’, OSF-RFC 91.0, January 1996.

[RFC 92] S. Mullan, ‘‘DCE Interoperability with Kerberos’’, OSF-RFC 92.0, January 1996.

[RFC 93] M. Burati, ‘‘DCE 1.2 Security Scalability and Performance — Functional Specification’’, OSF-RFC 93.0, February 1996.

[RFC 94] M. Heroux, ‘‘A Private Key Storage Server for DCE — Functional Specification’’, OSF-RFC 94.0, April 1996.

[RFC 96] H. Hashimoto, T. Ito, T. Yamaura, ‘‘DFA: Distributed File Access Manager — Functional Specification’’, OSF-RFC 96.0, March 1996.

[RFC Index] W. Tuvell, ‘‘Index to OSF-RFC’s’’, OSF-RFC Index, *Latest version*.

AUTHOR’S ADDRESS

Rich Salz
Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142
USA

Internet email: rsalz@osf.org
Telephone: +1-617-621-7253