INTERNATIONAL STANDARD

ISO/IEC 4396-1

First edition

Telecommunications and information exchange between systems — Recursive inter-network architecture —

Part 1: Reference model

Télécommunications et échange d'information entre systèmes — Architecture récursive inter-réseaux —

Partie 1: Modèle de référence

PROOF/ÉPREUVE

Reference number
ISO/IEC 4396-1:2023(E)

Contents

Page

Foreword iv

Introduction v

1 Scope 1

2 Normative references 1

3 Terms and definitions 1

4 Symbols and abbreviated terms 9

5 Basic concepts of distributed applications 10

5.1 Introduction to distributed applications 10

5.2 Naming concepts for Distributed-Application Facilities (DAFs) 11

5.3 Elements of a Distributed-Application Facility (DAF) 11

5.3.1 Overview 11

5.3.2 DAF security principles 13

5.3.3 Common Distributed-Application Protocol (CDAP) 13

5.3.4 DAF enrollment 15

6 Introduction to distributed management systems (DMSs) 16

6.1 General 16

6.2 Network Management DMS (NM-DMS) 16

6.3 Application Management DMS (AM-DMS) 18

6.4 Name Space Management DMS (NSM-DMS) 18

6.5 DIF allocator 19

6.5.1 General 19

6.5.2 DIF allocator structure 20

6.5.3 Operations 21

7 Distributed InterProcess Communication — Fundamental structure 22

7.1 Distributed-IPC-Facility (DIF) 22

7.1.1 Overview 22

7.1.2 IPC process 23

7.1.3 Flows and connections 24

7.1.4 DIF security principles 24

7.1.5 IPC components 25

7.1.6 DAF Infrastructure for IPC 28

7.2 Naming in Distributed IPC Facilities 30

7.3 DIFs are layers 31

Annex A (informative) **Basic concepts of distributed systems** 32

Bibliography 39

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6 *Telecommunications and information exchange between systems*.

A list of all parts in the ISO/IEC 4396 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

 PROOF/ÉPREUVE

Introduction

The purpose of this document is to provide a reference model for the concepts in Recursive Inter-Network Architecture (RINA)[1],[2]. This document provides the fundamental definitions for the ISO/IEC 4396 series. It defines an architecture.

This document is the high-level description of the concepts, the elements and how they work. This is a top-level specification, not a tutorial. Other more detailed specifications will draw on the ISO/IEC 4396 series as their starting point.

Telecommunications and information exchange between systems — Recursive inter-network architecture —

Part 1:
Reference model

1 Scope

This document provides the reference model for the Recursive Inter-Network Architecture (RINA). It describes:

— the basic concepts of distributed systems and distributed applications;

— distributed management systems (DMSs);

— the fundamental structure of distributed Inter-Process Communications;

— the Distributed Inter-Process Facility (DIF) operations.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
address
identifier that is a synonym for the fully qualified IPC-Process-Instance-name, which is a member of a *distributed-IPC-facility (DIF)* (3.32)

Note 1 to entry: An address is only unambiguous within the DIF (and is assigned by the DIF). This identifier can be assigned to facilitate its usefulness to the operation of the DIF, i.e. location-dependent.

3.2
application connection
shared state maintained by two communicating peer *application entities (AEs)* (3.3)

Note 1 to entry: Application connections go initially through an establishment phase, in which enough data is exchanged to establish a shared understanding between both AEs, to later proceed to the data transfer phase, in which both AEs exchange data. In Recursive Inter-Network Architecture (RINA), the establishment phase is handled by *common application connection establishment procedure (CACEP)* (3.17), while the data transfer phase is the responsibility of *common distributed application protocol (CDAP)* (3.18).

3.3
application-entity
AE
task within an application process that is directly involved with exchanging application information with other *application processes (APs)* (3.9)

3.4
application entity instance
AE-instance
AEI
instantiation of an *application entity (AE)* (3.3) within an *application process (AP)* (3.9)

3.5
application entity instance identifier
AE-instance-id
AEI-id
identifier which is unambiguous within the *application entity (AE)* (3.3)

Note 1 to entry: The AE-instance-id may be ambiguous within the *application process (AP) name space* (3.13) unless qualified by the application process name, *application process instance id* (3.11), and the *application entity (AE) name* (3.6).

3.6
application entity name
AE name
identifier from the *application entity (AE) name space* (3.8) which is unambiguous within the *scope* (3.55) of the *application process (AP)* (3.9)

Note 1 to entry: An AE name when concatenated with an *AP name* (3.12) is unambiguous within the *AP name space* (3.13), as is an AE name concatenated with an AP name and an *AP instance id* (3.11).

3.7
application programming interface primitive
API primitive
library or system call used by an application to invoke functions, in particular *inter process communication (IPC)* (3.39) functions, such as requesting the allocation of IPC resources

3.8
application entity name space
AE name space
set of strings which may be assigned to *application entities (AEs)* (3.3) of a given *application process (AP)* (3.9) and used to reference them by other applications in the same naming domain

3.9
application- process
AP
software program in a *processing system* (3.46) intended to accomplish some purpose

Note 1 to entry: An application process contains one or more tasks or *application entities (AEs)* (3.3) as well as functions for managing the resources (e.g. processor, storage, and IPC) allocated to this AP.

Note 2 to entry: Tasks are also application processes.

3.10
application process instance
AP instance
instantiation of an *application process (AP)* (3.9) on an operating system

3.11
application process instance id
AP instance id
identifier that is unambiguous within the *application process (AP)* (3.9) and is bound to an *AP instance* (3.10) in order to distinguish among multiple AP instances

Note 1 to entry: An AP instance id concatenated with an *AP name* (3.12) is unambiguous in the *AP name space* (3.13).

3.12
application -process -name
AP -name
string assigned to a single *application process (AP)* (3.9) from an *AP name space* (3.13)

Note 1 to entry: An AP name is not assigned to any other AP while bound to the one to which it has been assigned.

3.13
application process name space
AP name space
set of strings which may be assigned to *application processes (APs)* (3.9) and used to reference them by other APs in the same naming domain

3.14
application protocol
protocol used between two *application entities (AEs)* (3.3) to perform operations external to the *protocol machine (PM)* (3.50) itself

Note 1 to entry: The distinguishing characteristic of application protocols is that they modify states external to the protocol.

3.15
assignment
operation that allocates a name in a *name space* (3.43), essentially marking it as being in use

Note 1 to entry: Assignment makes names available to be bound. This allows certain portions of a name space to be "reserved" and not be available for *binding* (3.16). The corresponding reverse operation, *de-assignment* (3.25), removes it from use.

3.16
binding
function, $\mathbf{F_M}$,$\mathbf{N_S}$, that defines the mapping of a subset of elements of {**NS**} to elements of {**M**}

Note 1 to entry: This function is one-to-one and into. The operation, *binding*, binds a name to an object.

Note 2 to entry: Once bound, any reference to the name locates or accesses the object.

3.17
common application connection establishment procedure
CACEP
procedure to authenticate flow participants and initialize the application naming and protocol information

Note 1 to entry: CACEP naming and protocol information relates to the *application protocol* (3.14) that will be used by applications to exchange information (e.g. abstract and encoding rules, object model versions). In case of Recursive Inter-Network Architecture (RINA), the application protocol is *common distributed application protocol (CDAP)* (3.18).

3.18
common distributed application protocol
CDAP
application protocol (3.14) component of a *distributed application facility (DAF)* (3.27) used to construct arbitrary distributed applications

Note 1 to entry: CDAP enables distributed applications to deal with communications at an object level, rather than forcing applications to explicitly deal with serialization and input/output operations. CDAP provides a straightforward and unifying approach to sharing data over a network without having to create specialized protocols.

Note 2 to entry: *Distributed IPC facility (DIF)* (3.32) is an example of a *distributed application facility (DAF)* (3.27).

3.19
computing system
collection of all *processing systems* (3.46) (some specialized) in the same management domain

Note 1 to entry: There are no restrictions on the connectivity of computing systems.

3.20
connection
shared state between *error and flow control protocol machine EFCP PMs* (3.34)

3.21
connection-endpoint-identifier
CEP-id
identifier that is unambiguous within the *scope* (3.55) of an *interprocess communication (IPC)* process which identifies an *error and flow control protocol machine EFCP PM* (3.34) instance

3.22
connection-identifier
identifier internal to the *distributed IPC facility (DIF)* (3.32) that are unambiguous within the scope of communicating *error and flow control protocol machine EFCP PMs* (3.34) from that DIF

Note 1 to entry: The connection identifier is formed by the concatenation of the source and destination connection establishment procedure (CEP)-ids to identify the two directions of the connection.

3.23
data transfer control procedure
DTCP
half of an *error and flow control protocol (EFCP)* (3.33) that performs loosely bound (feedback) mechanisms, such as retransmission and flow control

Note 1 to entry: The DTCP protocol machine (PM) maintains state, which can be discarded after long periods of no traffic.

Note 2 to entry: One instance of a DTCP is created for each connection of a flow.

Note 3 to entry: All connections in Recursive Inter-Network Architecture (RINA) have flow control. Connections without flow control are denial of service attack vector.

3.24
data transfer procedure
DTP
half of an *error and flow control protocol (EFCP)* (3.33) that performs tightly bound mechanisms, such as ordering and fragmentation/reassembly

Note 1 to entry: One instance of a DTP protocol machine (PM) is created for each connection allocated.

3.25
de-assignment
operation that deallocates a name in a *name space* (3.43), removing it from use

3.26
delimiting
operation to delineate the beginning and end of a *service-data-unit (SDU)* (3.56) and package it into the user-data field of a *protocol-data-unit (PDU)* (3.45)

Note 1 to entry: Delimiting is usually the first operation performed by the *distributed IPC facility (DIF)* (3.32) when an SDU is submitted.

Note 2 to entry: Delimiting enables the DIF to deliver the SDU as a unit of data to its recipient intact.

3.27
distributed application facility
DAF
distributed application
collection of *application processes (APs)* (3.9) in *processing systems* (3.46) that exchange information using *interprocess communication (IPC)* (3.39) and maintain shared state to cooperate in performing some task or function

Note 1 to entry: There are at least two APs and at least one processing system in each DAF.

Note 2 to entry: The DAF forms a black box to the members of the DAF who may be executing on one or more processing systems.

Note 3 to entry: In some DAF, all members of the DAF will be the same, i.e. a homogeneous DAF, while in others they may be different, i.e. a heterogeneous DAF.

3.28
distributed application name
whatevercast name (3.61) for the set of *application processes (APs)* (3.9) comprising a distributed application depending on the operation

Note 1 to entry: A whatevercast name is generally taken from the same *name space* (3.43) as the APs and is used to identify a distributed application. An important type of distributed application is a *distributed IPC facility (DIF)* (3.32), i.e. the set of cooperating *interprocess communication (IPC)* (3.39) processes.

3.29
distributed application process
DAP
application process (AP) (3.9) that is a member of a *distributed application facility (DAF)* (3.27)

3.30
distributed application process name
DAP name
synonym for an *application process (AP)* (3.9) name

3.31
distributed application process synonym
DAP synonym
synonym for a *distributed application process (DAP)* (3.30) that is a member of a specific *distributed application facility DAF* (3.27) and is only unambiguous within the DAF (and is assigned by the DAF)

Note 1 to entry: The names may be structured to facilitate their use within the DAF.

3.32
distributed IPC facility
DIF
layer
collection of *application process (AP) instances* (3.10) that are cooperating to provide *interprocess communication (IPC)* (3.9)

Note 1 to entry: A DIF is a *distributed application facility (DAF)* (3.27) that does IPC.

Note 2 to entry: The DIF provides IPC services to AP instances of a DAF or IPC process instances of other DIFs via a set of application programming interface (API) primitives that are used to exchange information with the IPC process instances' peer.

3.33
error and flow control protocol
EFCP
data transfer protocol used to maintain an instance of *interprocess communication (IPC)* (3.39) within a *distributed IPF facility (DIF)* (3.32).

Note 1 to entry: The functions of this protocol can be used to provide reliability, order and flow control as required as determined by policy.

3.34
EFCP protocol machine
EFCP PM
instance of the *error and flow control protocol (EFCP)* (3.33) for a single connection

Note 1 to entry: An EFCP PM consists of two state machines loosely coupled through a single state vector: one that performs the *data transfer procedure (DTP)* (3.24) *protocol machine (PM)* (3.50) and the other that performs the *data transfer control procedure (DTCP)* (3.23) *protocol machine (PM)* (3.50).

3.35
flow
binding (3.16) of a connection to source and destination ports

3.36
flow allocator
FA
task that handles requests to allocate a *flow* (3.35)

3.37
flow allocator instance
FA-instance
FAI
instance created for each allocation request to manage the flow for its lifetime

Note 1 to entry: The flow allocator instance will translate the (quality of service) QoS requested by the Application-Process into specific policies and find the destination Application and determine if the allocation can be honoured. The FAI-identifier or port–id is returned to the application as a handle for referencing the allocation.

3.38
IPC process
IPCP
application process (AP) (3.9) whose primary purpose is managing *inter process communication (IPC)* (3.39)

3.39
inter process communication
IPC
service that allows two or more *application process (AP) instances* (3.10) to exchange data

3.40
IPC resource manager
IRM
component of a *distributed application facility (DAF)* (3.27) that manages its use of IPC

