ETSI TS 102 825-7 V1.2.1 (2011-02)

Technical Specification

Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 7: CPCM Authorized Domain Management

Reference
RTS/JTC-DVB-252-7

Keywords
broadcast, DVB

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2011.
© European Broadcasting Union 2011.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™, **TIPHON**™, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
LTE™ is a Trade Mark of ETSI currently being registered
for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies, content owners and others committed to designing global standards for the delivery of digital television and data services. DVB fosters market driven solutions that meet the needs and economic circumstances of broadcast industry stakeholders and consumers. DVB standards cover all aspects of digital television from transmission through interfacing, conditional access and interactivity for digital video, audio and data. The consortium came together in 1993 to provide global standardisation, interoperability and future proof specifications.

The present document is part 7 of a multi-part deliverable. Full details of the entire series can be found in part 1 [1].

Introduction

CPCM is a system for Content Protection and Copy Management of commercial digital content delivered to consumer products. CPCM manages content usage from acquisition into the CPCM system until final consumption, or export from the CPCM system, in accordance with the particular usage rules of that content. Possible sources for commercial digital content include broadcast (e.g. cable, satellite, and terrestrial), Internet-based services, packaged media, and mobile services, among others. CPCM is intended for use in protecting all types of content - audio, video and associated applications and data. CPCM specifications facilitate interoperability of such content after acquisition into CPCM by networked consumer devices for both home networking and remote access.

This first phase of the specification addresses CPCM for digital Content encoded and transported by linear transport systems in accordance with TS 101 154 [i.1]. A later second phase will address CPCM for Content encoded and transported by systems that are based upon Internet Protocols in accordance with TS 102 005 [i.2].

Note for New Readers:

Readers who are unfamiliar with the present document may find it helpful to first read the information scenarios described in TR 102 825-11 [i.4].

1 Scope

The present document specifies the Authorized Domain Management for the Digital Video Broadcasting (DVB) Content Protection and Copy Management (CPCM) system.

The present document defines messages and state machines that together define the conformant behaviour of an implementation.

The present document also defines a mandatory method for performing ADSE. Additionally, the present document describes a set of tools that may be used by a Compliance and Robustness C&R regime to extend the ADSE.

The present document also describes APIs to other CPCM components; however these are informative only and are not required to be implemented for conformance.

The hardware, software, middleware and other technologies employed are the sole choice of the implementer, governed by the requirements of the relevant (C&R) regime.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 825-1: "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 1: CPCM Abbreviations, Definitions and Terms".

[2] ETSI TS 102 825-4: "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 4: CPCM System Specification".

[3] ETSI TS 102 825-5: "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 5: CPCM Security Toolbox".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 101 154: "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".

[i.2] ETSI TS 102 005: "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols".

[i.3] ETSI TS 102 825-2: "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 2: CPCM Reference Model".

[i.4] ETSI TR 102 825-11: "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 11: CPCM Content management scenarios".

[i.5] ETSI TR 102 825-12: "Digital Video Broadcasting (DVB); Content Protection and Copy Management (DVB-CPCM); Part 12: CPCM Implementation Guidelines".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TS 102 825-1 [1] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 102 825-1 [1] apply.

4 Authorized Domain Management (ADM)

ADM is the mechanism that allows CPCM Devices belonging to a household to establish and Join a DVB CPCM Authorized Domain (AD). It also provides an interoperable means of managing that AD.

Basic expectations on ADM are given in TS 102 825-2 [i.3].

4.1 ADM in the CPCM System

Figure 1 shows how ADM fits into the CPCM Reference Model.

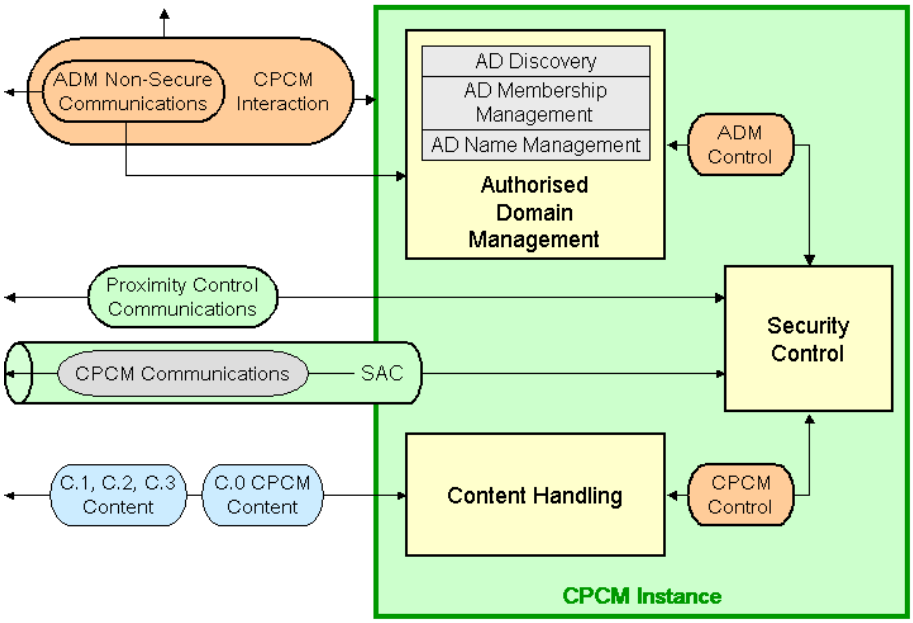

Figure 1: ADM in the CPCM Reference Model

Figure 2 provides another view of ADM within a CPCM implementation. It identifies that there are three key internal sub-systems with which the ADM Sub-system must communicate.

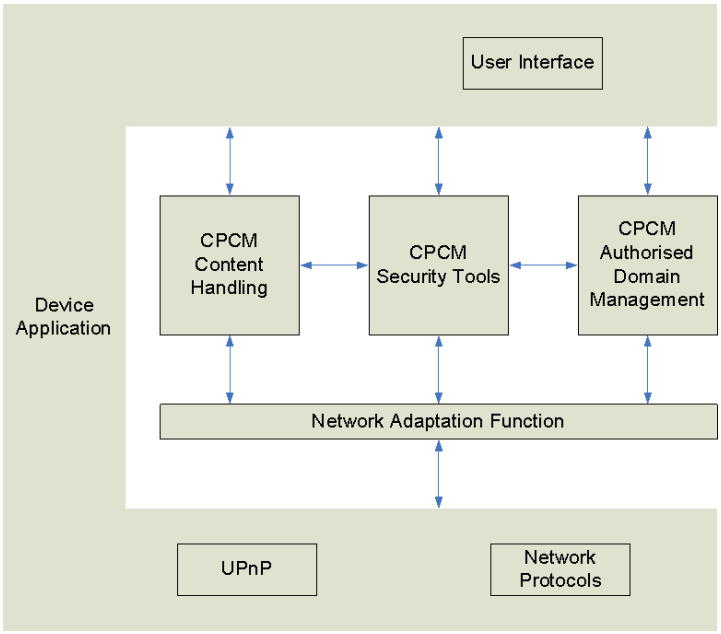

Figure 2: ADM Interfaces to other CPCM Sub-systems

It would be possible to build a CPCM implementation that strictly follows the architecture in figure 2, but this is not required. For reasons of clarity the present document is written in terms of the interfaces in figure 3.

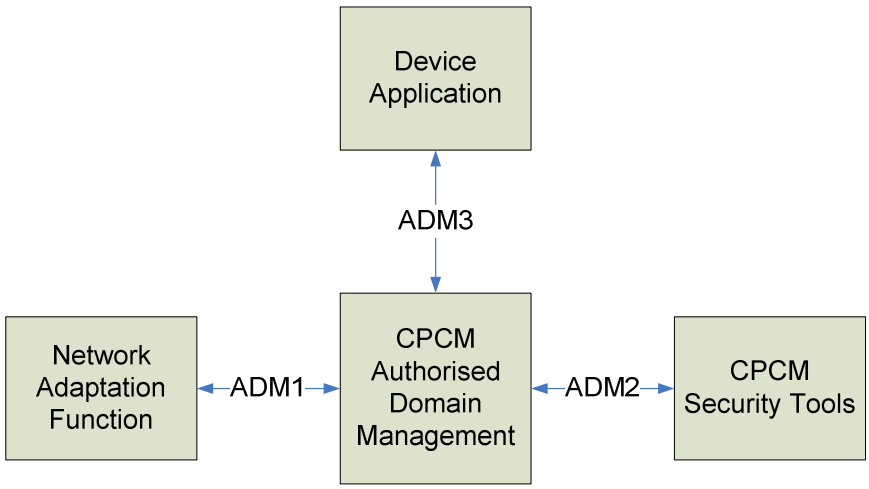

Figure 3: ADM Interfaces

ADM1 is the interface which is used to communicate with peer ADM Instances in other devices, via a Network Adaptation function that is specific to each network technology.

EXAMPLE: On DVB-HN (Home Network) conformant systems, ADM messages will be transferred using the UPnP protocols. However, for other situations such as serial Smartcard or CI interfaces, ADM messages will be transferred without UPnP support.

The present document includes a formal description of the ADM1 interface.

ADM2 is the interface to the Security Tools Sub-system of CPCM. The present document defines the normative behaviour of the Security Control and provides an informative API for use across this interface.

ADM3 is the informative interface between a Device Application and the ADM.

4.2 Local ADM Master and Domain Controller

4.2.1 ADM Local Master Concept and Requirements

A single CPCM Instance acts as a temporary Local ADM Master for the AD members in the Local network. The ADM Local Master has no specific additional data (i.e. the same AD data is replicated to all Instances in the AD). The Local Master can be replaced at any time to avoid a single-point-of-failure.

Figure 12 and clause 5.1.7 describe how to determine which Instance fulfils this role at any given moment.

The basic behaviour is as follows:

- The first Instance to create the Authorized Domain immediately becomes the Local Master.
- When an Instance attempts to discover available ADs or properties of connected Instances, each connected CPCM Instance responds.
- When an Instance requests to Join an AD, the request is directed to the Local Master.
- Instances belonging to the same AD elect or otherwise determine which device is the most suitable Instance to act as their Local ADM Master.
- Each Instance has a "Local Master capability".
- An election may be called by any member Instance at any time.
- The Instance with highest capability is elected as Local Master.

The Local Master concept is used to reduce the complexity of ADM protocol interactions. It does not solve issues relating to Authorized Domain Size and Extent (ADSE) threats. These are addressed by the Domain Controller concept described below.

4.2.2 The Global Domain Controller Concept

ADM defines a Domain Controller for Secure Authorized Domain Size and Extent implementation. The Domain Controller is responsible for all AD operations (including AD Joining and Leaving) that may affect ADSE records and decision making processes. The user is able to Transfer Domain Controller functionality from one CPCM Instance to another, or to Split the Domain Controller function between two or more CPCM Instances. Each Domain Controller has additional data that reflects the current state of the AD. This data is replicated to all other CPCM Instances within the same AD to limit the risk of a single point of failure.

The basic behaviour is as follows:

- The first capable Instance to create the AD becomes Domain Controller.
- If there is a single Domain Controller, then that Domain Controller will always be chosen as the Local ADM Master. If several Domain Controllers are present, the Local ADM Master shall be one of them.
- Local ADM Masters are able to discover a Remote Domain Controller and will give relevant information to Instances needing to contact the Local Domain Controller (e.g. for AD Joining or Leaving).
- The Domain Controller function can be Transferred from one Instance to another.
- The Domain Controller can be Split over several Instances and later re-Merged. ADSE records are Split and re-Merged as well. ADSE records may be Rebalanced between two Domain Controllers.

NOTE: This is not the same concept as Splitting the AD itself into two new ADs. The AD Splitting function, while planned for the future, is not included in the present document.

Figure 21 and clause 5.1.12 describe the process for changing the assigned Domain Controller.

4.3 Authorized Domain capabilities

CPCM Instances shall publish their Authorized Domain Management capability through the following fields in the certificate.

Table 1: ADM Fields

Field	Meaning
AD_aware	An AD aware CPCM Instance is one that is able to securely create or record the AD Secret. It has direct access to Content that is bound to its AD. It is able to handle AD restricted CPCM Content. If the AD_aware certificate field is not asserted then the CPCM Instance is not able to get or create the AD Secret in any way, and cannot Join any AD.
ADM_capable	An ADM capable CPCM Instance that is AD aware and can also run ADM protocols. The AD_aware certificate field shall be also asserted in this case.
ADM_LM_capable	An ADM LM capable CPCM Instance is ADM capable and can also act is a Local Master. A Local Master is able to run as a proxy for other CPCM Instances in their communications with Domain Controllers. The ADM_capable certificate field shall also be asserted in this case.
ADM_DC_capable	An ADM Domain Controller capable CPCM Instance is ADM capable and can also act as a Domain Controller. A Domain Controller shall be able to securely run ADSE enforcement and to securely manage ADSE values. The ADM_LM_capable certificate field shall also be asserted in this case. A DC capable Instance is always an ADSE countable Instance and always ADM_LM_capable.
ADSE_countable	An ADSE countable CPCM Instance (which is also a Countable Instance of CPCM Functionality (CICF)) is able to Consume or Export Content and shall thus be counted in ADSE tests (see next clause). The AD_aware certificate field shall be also asserted in this case.

Please see TS 102 825-5 [3] for more details of the CPCM Certificate fields.

Some CPCM Instances may be AD aware without being ADM capable. This might, for instance, be the case for smart cards implementing a proprietary CA or DRM that is plugged into a Set-Top-Box with proprietary software and CPCM Instance. The set-top-box CPCM Instance may let the smart card CPCM Instance Join using proprietary protocols in accordance with rules set forth in a particular C&R regime.

4.4 Authorized Domain Size and Extent

Authorized Domain Size and Extent (ADSE) ensures that CPCM Instances do not violate the AD size, scope and extent requirements when they attempt to become members of the same AD.

4.4.1 ADSE Counts and Values

The ADSE method makes use of the following counters and ceilings.

Table 2: ADSE values maintained by the DC

Variable	Bits	Type	Meaning
total_count	16	uimsbf	The current number of CICF in the AD.
total_ceiling	16	uimsbf	The maximum number of CICF allowed in the Authorized Domain
remote_count	16	uimsbf	The current number of CICF in the AD that Joined Remotely to the Domain Controller that authorized the AD Join.
remote_ceiling	16	uimsbf	The maximum number of CICF that may Remotely Join the AD.
local_count	16	uimsbf	The current number of CICF in the AD that have Joined Locally to the Domain Controller that authorized the AD Join.
local_ceiling	16	uimsbf	The maximum number of CICF that may Locally Join the AD without running the Quorum test.
DC_split_count	16	uimsbf	The current count of Domain Controller Splits.
DC_split_ceiling	16	uimsbf	The maximum number of DC Splits that may occur.
DC_remote_count	16	uimsbf	The current count of Remote Transfer or Remote Splits of the Domain Controller.
DC_remote_ceiling	16	uimsbf	The maximum number of Remote Transfer or Remote Splits of the Domain Controller.
NOTE: total_ceiling is greater than or equal to the sum of local_ceiling and remote_ceiling.			

Table 3: ADSE values maintained by each AD aware Instance

Variable	Bits	Type	Meaning
history_count	16	uimsbf	The total number of individual Authorized Domains that the CPCM Instance has Joined.
history_ceiling	16	uimsbf	The maximum number of Authorized Domains that the CPCM Instance can Join, as set by the C&R regime.
NOTE: The history_ceiling variable can be used to accommodate the anticipated differences between owned and rented CPCM Devices.			

These counter and ceiling values are named throughout the document ADSE counts or ADSE values.

If a ceiling field value is set to 0xFFFF then the ceiling is considered to be infinite and no ceiling limits shall be applied.

4.4.2 ADSE Method Description

The ADSE Method is defined by the following rules:

1) There is one logical Domain Controller per AD, initially instantiated in a single physical Domain Controller contained within a single CPCM Instance within a single CPCM Device. The logical Domain Controller function may then be Transferred from one physical device to another or may be shared between two or more devices. This Logical Domain Controller is an instantiation of the Single Household Metric (see clause 7.1).

2) One Domain Controller functionality may be Transferred Locally or a limited number of times Remotely.

3) One Domain Controller functionality is required for a CICF to be able to Join an AD. This includes Remote AD Joins.

4) Non-CICF Instances may be Joined to the AD by any CPCM Instance that is a member of that AD, but only Locally.

5) The total_ceiling shall be greater than or equal to the sum of remote_ceiling and the local_ceiling. In case it is greater, the method is based on the Quorum Test (see clause 7.4).

6) Each time a new CICF attempts to Join an AD, the following process must be followed:

   a. If the total_count is equal to total_ceiling, then the CICF is denied permission to Join the AD, unless AD Membership Assignment by Authorized Authenticated Agent (ADMAAA) (please see clause 7.8) can be performed successfully.

b. If the CICF is Remote:

i. if the `remote_count` is equal to the `remote_ceiling`, then the CICF is denied permission to Join the AD, unless ADMAAA can be performed successfully.

ii. else, the CICF is granted permission to Join the AD and the counters for `remote_count` and `total_count` are incremented by one.

The Domain Controller shall maintain a list of CPCM Instances that Joined remotely. The list is transferred, split, merged or adjusted when the DC is Transferred, Split, Merged or Rebalanced.

c. If the CICF is Local and `local_count` is lower than the `local_ceiling`, then the CICF is granted permission to Join the AD. The `local_count` and `total_count` are both incremented by one.

d. If the CICF is Local and `local_count` is equal to or greater than `local_ceiling`, a quorum test is performed with other CICFs:

i. A Challenge is broadcast together with the ADID.

ii. Each CICF of the same AD responses through a message carrying the same Challenge and its own identifier. This message is protected using the AD Secret.

iii. The Domain Controller runs a Proximity Test with each CICF that responded properly:

1. If the number of other Local CICFs that are connected at that time is equal to or greater than the "quorum percentage" of the `local_count` then the CICF is granted permission to Join the AD: both `local_count` and `total_count` are both incremented by one.

2. Else, if the `remote_count` is lower than the `remote_ceiling`, then the CICF is granted permission to Join the AD and the counters for `remote_count` and `total_count` are incremented by one.

3. Else the CICF is denied permission to Join the AD, unless ADMAAA can be performed successfully.

NOTE 1: The "quorum percentage" is a value to be defined by the C&R regime.

7) When a Domain Controller becomes Local to a CPCM Instance that is in its list of CICF that Joined remotely, and one of the following conditions is met:

a. The `local_count` is less than the `local_ceiling`, or

b. The `local_count` is equal to or greater than the `local_ceiling` and the number of Local CICFs is equal to or greater than the "quorum percentage" of the `local_count` value, or

c. ADMAAA authorizes it,

then the Domain Controller functionality shall decrement the `remote_count` by one and increment the `local_count` by one. The CICF shall then be removed from the list of Instances that Joined remotely. Once `remote_count` reaches zero, the Domain Controller shall empty its list of Instances that Joined remotely.

8) Each time a CICF attempts to Leave an AD:

a. If the CICF is not in the list of Instances that Joined Remotely, the AD Leave is considered as Local (whether the Instance is actually Remote or Local to that DC) and `local_count` and `total_count` are decremented by one.

b. If the CICF is in the list of Instances that Joined Remotely, the AD Leave is considered as Remote (whether the Instance is actually Remote or Local to that DC) and `remote_count` and `total_count` are decremented by one.

9) Domain Controller functionality can be Split into multiple parts. The number of Splits will be limited by the C&R regime. Domain Controller functionalities can also be re-Merged. A DC Merge re-credits the Split counters for any unused counts.

10) A Domain Controller must be instantiated within a CICF.

11) The Domain Controller functionality may be Split Locally, i.e. the CICF gaining Domain Controller functionality must be Local to the CICF giving the Domain Controller functionality.

12) The total number of Remote Domain Controller Transfers and Splits shall not exceed a ceiling fixed by the C&R regime.

13) Counters and ceilings may be Rebalanced between different Split Domain Controller functionalities of the same AD.

14) After a Domain Controller Split, Merge or Rebalance, the sums for `total_count`, `total_ceiling`, `remote_count`, `remote_ceiling`, `local_count`, `local_ceiling`, `DC_remote_count`, `DC_remote_ceiling`, `DC_split_count` and `DC_split_ceiling` over the involved Domain Controller functionalities shall be the same as before the activity. Upon DC Rebalancing and Splitting, the way each counter and ceiling is affected is left to the implementer as long as the following consistency rules are obeyed:

    a. For each Domain Controller, the sum of `local_count` and `remote_count` shall be equal to `total_count`.

    b. For each Domain Controller, `total_count, remote_count`, `DC_split_count` and `DC_remote_count` shall be lower than the corresponding ceiling.

    Each entry present on either list of CICFs that Joined remotely before the Split, Merge or Rebalance shall still be present on exactly one list after completion.

15) The ceilings can be changed at any time by an ADMAAA or by any test approved by the C&R regime.

16) An Instance shall not Join an AD if it has exceeded the AD Join/Leave Frequency/Repetition Rate as describe in the Domain Membership History (DMH) tool (see clause 7.6).

17) When a DC is Split to a CPCM Instance which is in the list of CICF that Joined remotely, the corresponding entry in the list shall be transferred during the Split. The new DC, being Local to itself, shall then decrement by one the `remote_count`, increment by one the `local_count` and remove itself from the list.

A CICF is characterized by the fact that its `ADSE_countable` certificate field is asserted.

A count of '1' shall be assigned to each CPCM Device that is able to Output (Consume or Export) Content; such an Instance shall be deemed a "Countable Instance of CPCM Functionality", or CICF.

NOTE 2: The C&R regime may place constraints upon the number of potential Outputs that may be included in a single CICF. The Instance Certificate includes a field that contains a '0' or '1'. The C&R regime may require that a future CPCM Device be counted as more than '1' if it has too many Instance Certificates that should have a '1' count. In that case, the CPCM Device will embed several CPCM Instances with a '1' count.

Constraints on single AD memberships for CPCM Instances embedded in a single device are beyond the scope of the present document and may be defined by the C&R regime.

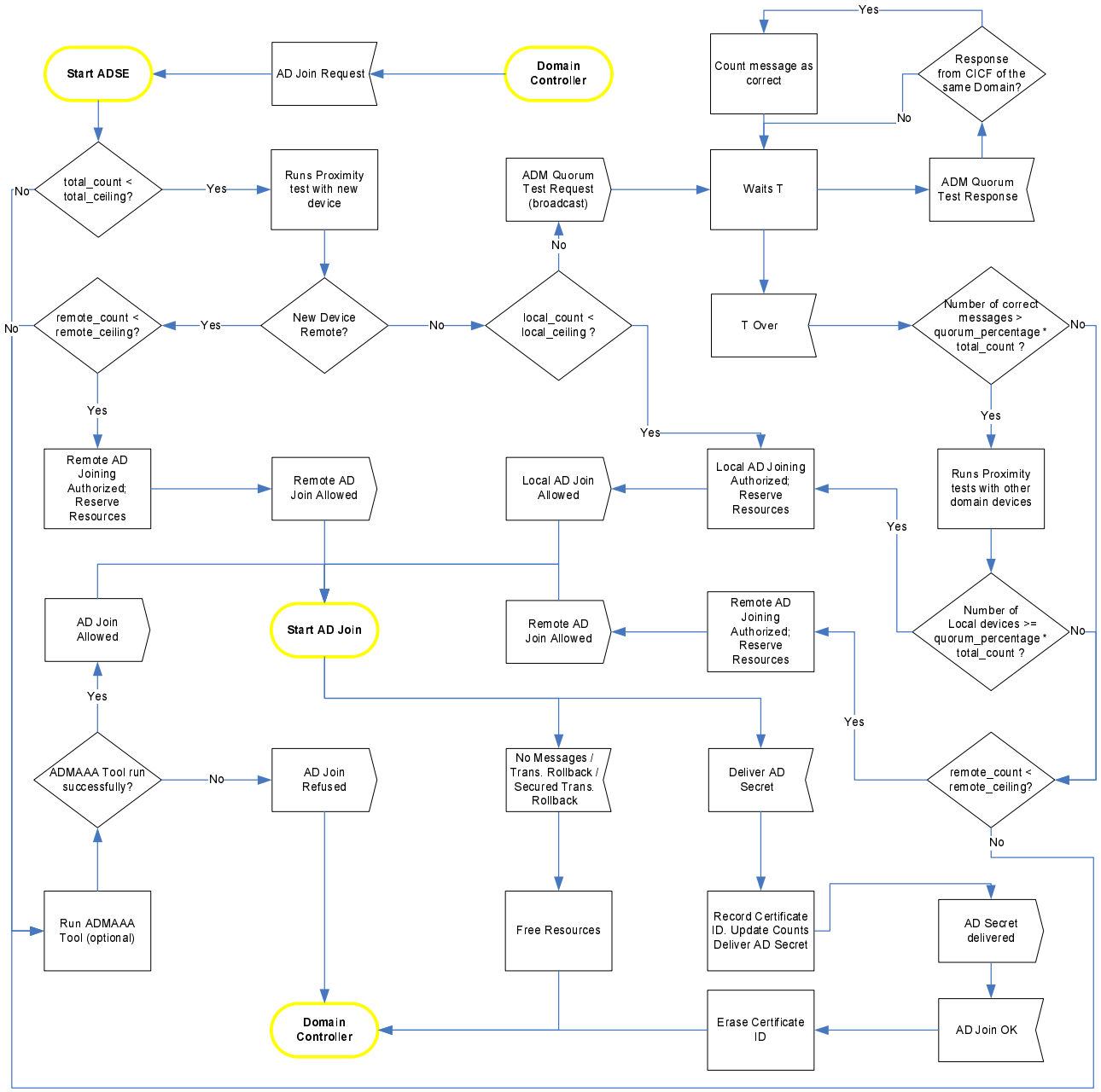

Figure 4: The ADSE Method

4.4.3 Other ADSE Methods

Other ADSE Methods may be defined by the CPCM C&R regime. They may be built upon optional ADSE tools described in clause 7.

5 The ADM State Machine - Normative text

Figures 5 to 28 define conformant behaviour of ADM implementations. The standardized state machine described here provides a simple single-threaded model. Implementations are free to employ a more sophisticated multi-threaded approach provided that the system behaviour remains conformant to the states shown here.

Figures 5 to 28 focus only on the conformant behaviour of the ADM components. The Security Control component behaviour is described in clause 6. The need for the Security Control to intervene is decided by an exchange of messages between the ADM and Security Control components. These Security Control messages are informative only and are to be defined by the implementation.

NOTE: As with any finite state machine, the ADM system relies on timeouts, for example to determine whether the expected responses are received or not. However, CPCM can be employed in many different network environments with very different latency characteristics, so it is not possible to provide specific normative values for these timeouts (waiting periods) suitable for all possible deployments of CPCM. Implementers are asked to consult TR 102 825-12 [i.5].

5.1 Normal Behaviour

Figures 5 to 28 define normal operations.

5.1.1 Discovery Protocol

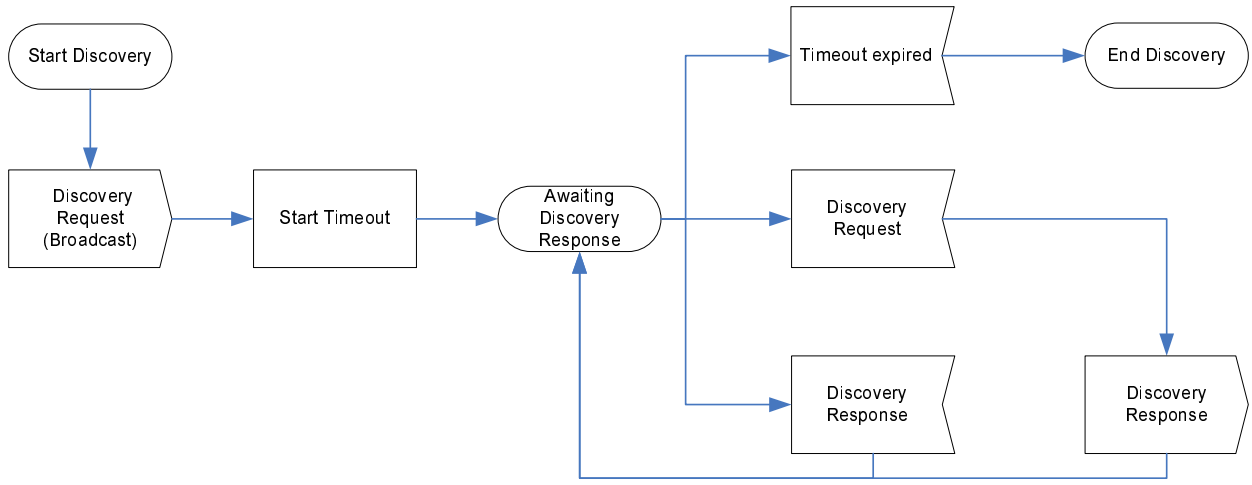

Figure 5: Discovery State Machine

The Discovery Protocol shall be used whenever a CPCM Device makes a connection to the AD. It is used in the following scenarios:

- It allows the newly connected CPCM Device to discover the identity of the current Local Master.
- It may be used by currently connected CPCM Devices to identify the current Local Master (see clause 5.1.7).
- It allows a Domain Controller to discover present CICFs to perform the Quorum Test.
- It is used by the Local Master to discover the connected Authorized Domain Controllers.

In the first three cases above, the Discovery protocol only involves Local CPCM Instances; however Discovery of the Domain Controllers also includes discovering Remote Domain Controllers.

To launch a discovery, a CPCM Instance broadcasts the `Discovery_request_message`. It then waits for all possible responses.

NOTE: Definition of waiting time needed to collect all possible responses is left to the implementer.

Each CPCM Instance receiving a `Discovery_request_message` shall answer using the `Discovery_response_message`.

When waiting time is over, the discovering CPCM Instance analyses all received responses and possible received requests. Analysis depends on the Discovery context and is detailed in the relevant clauses below.

5.1.2 Local Master and Domain Controller messaging

Some Authorized Domain Management operations require that an AD Member or a Domain Controller sends messages to the Local Master. However, CPCM Instances are not required to permanently know the identity of the current Local Master. Even if they recorded it, it may be the case that the Local Master has been disconnected prior to the occurrence of a new election.

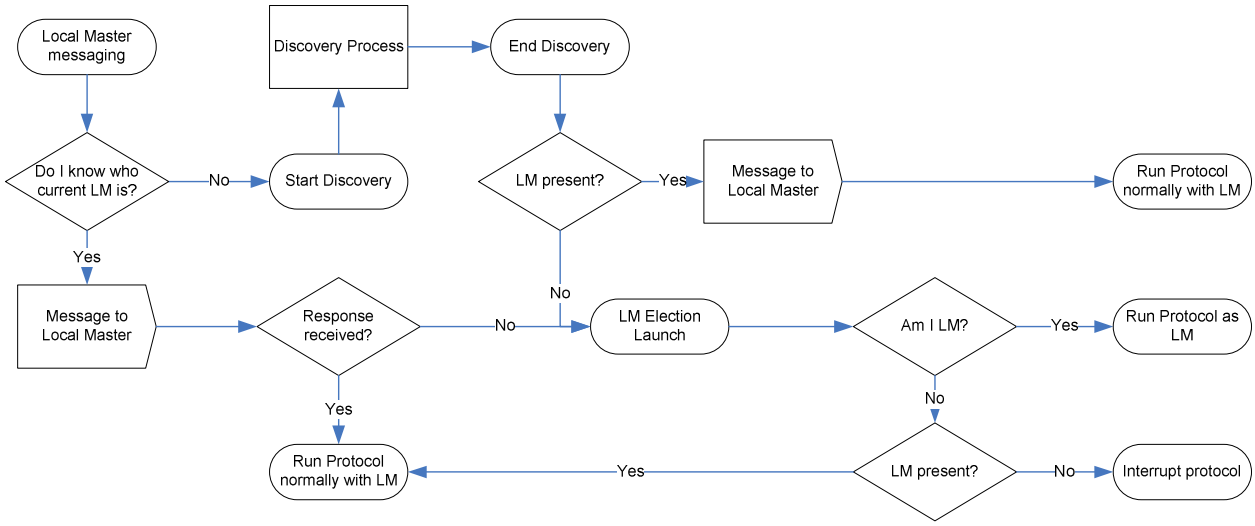

Figure 6: Local Master messaging

An AD Member or a Domain Controller wishing to send a message to the Local Master shall proceed in the following way:

- If the source CPCM Instance recorded the identity of the current Local Master, it sends the message directly to the Local Master. If the source receives an answer from the Local Master, then the current Local Master is still connected and the protocol can continue. If it does not receive any response or if the received response signalled that the contacted Instance is not the Local Master, it launches a Local Master Election (see clause 5.1.7).
- If the source does not record the identity of the current Local Master, it executes the Discovery Protocol (see clause 5.1.1). If, as a result of this, the Local Master is found to be present, it can send the intended message to the Local Master; otherwise it shall launch a Local Master Election.

At the end of the Local Master Election (if any), three cases may happen:

- If the CPCM Instance is not the new Local Master but another CPCM Instance has been elected, it sends the message to the new Local Master and continues to run the operation normally.
- If the CPCM Instance is the new Local Master, it runs the original protocol as Local Master.
- If no CPCM Instance has been elected (this is the case for instance if no Local Master capable CPCM Instance is present), the protocol is interrupted.

In order for the Local Master to identify and communicate with a Domain Controller the Local Master shall use the following mechanism:

- If the Local Master has an up-to-date list of connected Domain Controller(s), it sends the message to one of them. If it receives a response from that Domain Controller, the protocol shall run normally.
- If the Local Master has no such list, or if it did not receive a response from any of the contacted Domain Controllers, then it runs the Discovery Protocol to get an up-to-date list. At the end of the Discovery Protocol, if one or more Domain Controllers answered, it may run the protocol with any one of them. Else, the Local Master interrupts the protocol.
- A Local Master (that may also be a Domain Controller) may be requested to run the protocol with one specific Domain Controller. If this Local Master knows the Domain Controller is present, it contacts the Domain Controller directly. If the Local Master receives no response, or of it does not know whether the Domain Controller is connected, it runs the Discovery protocol. If the Domain Controller did not answer, the Local Master interrupts the protocol, otherwise the protocol proceeds normally.

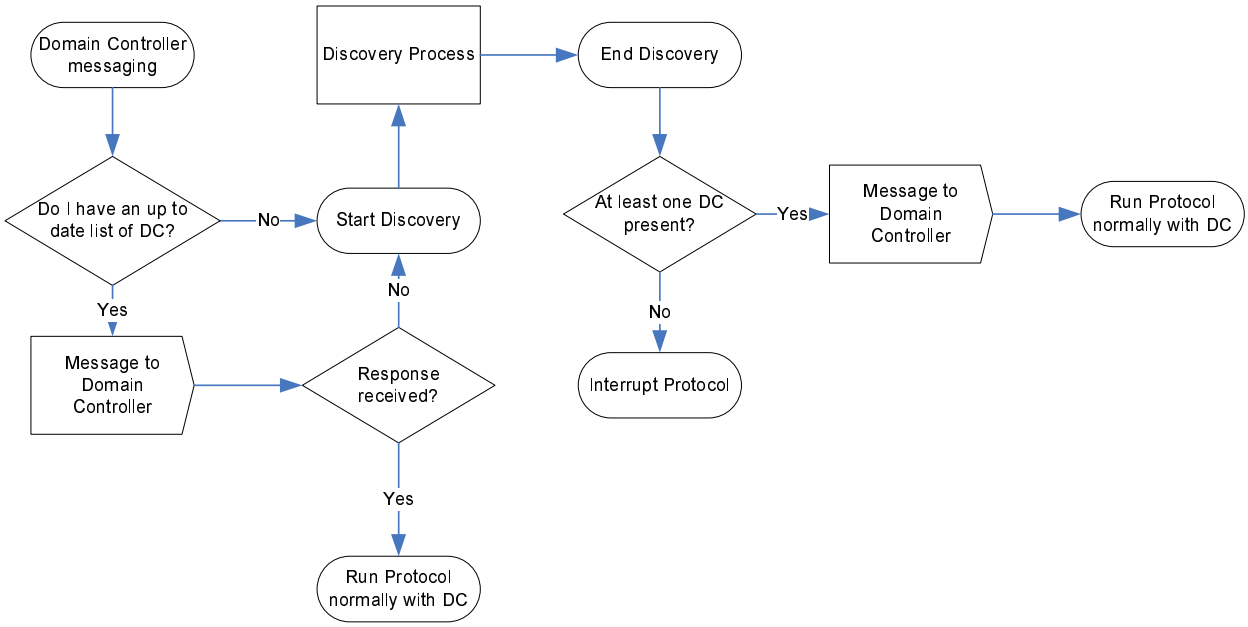

Figure 7: Domain Controller messaging

5.1.3 Connected AD Member

Table 4 describes how a connected AD Member (that is neither a Domain Controller nor the current Local Master) shall react when receiving messages.

Table 4: Actions by AD member when receiving messages

Message	Action and reason
Discovery_request_message	It shall send a *Discovery_response_message*.
AD_update_request_message	It should reply with its Domain Internal Record, if more recent (see clause 5.4) (see note).
AD_update_indication_message	It shall update its Domain Internal Record, if needed (see clause 5.4) (see note).
LM_master_election_indication_message	It shall record the identity of the newly elected Local Master.
LM_master_election_request_message	If it is Local Master capable, it shall run the Local Master Election process; otherwise, the message is ignored.
LM_master_election_response_message	The same as for the *Discovery_request_message* above. This case may occur where network latency causes the *LM_master_election_response_message* to be received before the initial request.
ADM_invite_message (AD Leave)	It shall start the AD Leaving protocol (see clause 5.1.11) with the Domain Controller designated in the message, if any.
AD Leaving request from a Device Application	As per *ADM_invite_message* (AD Leave) above, but the CPCM Instance shall contact the Local Master first.
Invitation Requests for Domain Controller Transfer or Split	Irrespective as to if these come from a Device Application or from a networked CPCM Instance; it shall start the Domain Controller Transfer Protocol (see clause 5.1.12) or the Domain Controller Split Protocol (see clause 5.1.13). In case of a request from a Device Application, the Domain Controller will first have to run the Discovery protocol to determine with which Domain Controller the protocol will be run.
Change AD name	It shall send the *AD_change_request_message* to the Local Master (as described in clause 5.1.2). If it receives an *AD_change_response_message* then it shall inform the Device Application of the received renaming results. Else, if no response is received, it shall warn the Device Application that the renaming failed.
Error message (No LM or DC present: AD Join not possible)	It shall start the Local Master messaging process (as described in clause 5.1.2). It reverts then to the connected AD member state.
NOTE: It is not required for each AD Member to permanently record a Domain Internal Record.	

When an AD Member has a recorded CPCM Instance Certificate Identifier, it may behave differently as described in clause 5.3.

If no CPCM Instance Certificate Identifier is recorded, *ADM_invite* messages with protocol status asserted are ignored.

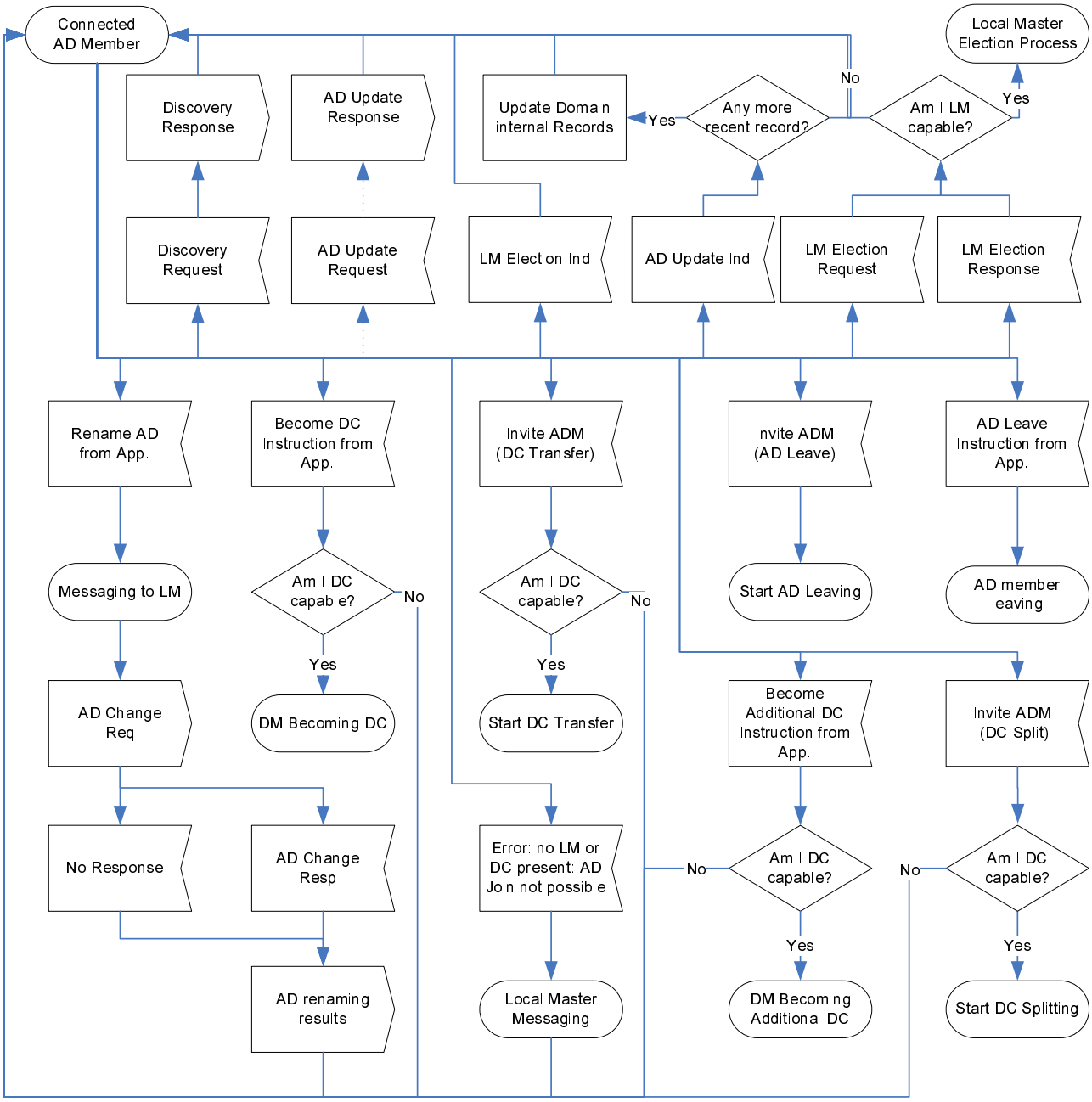

Figure 8: Connected AD Member State Machine

5.1.4 Connected Domain Controller

Table 5 shows how a Domain Controller that is not the Local Master shall react when receiving messages.

Table 5: Actions by DC that is not the LM when receiving messages

Message	Action and reason
Discovery_request_message	It shall respond with the *Discovery_response_message*.
AD_update_indication_message	It shall update its internal Domain Internal Record, if needed (see clause 5.4). If an update occurs, it shall broadcast its new Domain Internal Record.
LM_master_election_indication_message	If it permanently knows the identity of the current Local Master it shall note the identity of the newly elected Local Master.
LM_master_election_request_message	It shall execute the Local Master Election process (see clause 5.1.7).
LM_master_election_response_message	It shall execute the Local Master Election process (see clause 5.1.7).
Change AD Name	If the request came from another CPCM Instance through the *AD_change_request_message*, it shall request confirmation from the Device Application. If the Device Application confirmed or the request came from another Instance, it shall perform the following: • Rename the AD; • Inform the Device Application that the renaming occurred successfully; and • Broadcast an *AD_update_indication_message* to inform other Instances in the AD of the new name. Else, if the Device Application does not confirm, it shall inform the requesting CPCM Instance that the renaming was refused.
ADM_invite_message (AD Leave) or a AD Leaving request from the Device Application	Firstly it shall request confirmation from the Device Application; since the AD Leaving of a Domain Controller is irreversible (all its ADSE credits will be lost). Implementations may for instance first propose to Transfer the Domain Controller functionality to another Instance or to Merge it with another Domain Controller in which case it shall run the DC Transfer (see clause 5.1.12) or DC Merge protocol (see clause 5.1.14). If the Device Application confirms that the Instance shall cease to be a Domain Controller it requests its Security Control to proceed to the Domain Controller Leave protocol and confirms to the Device Application the success of the AD Leave operation once it is complete.
ADM_invite_message (DC Merge) or a DC Merging request from Device Application	It shall start the DC Merge protocol (see clause 5.1.14) as a client (i.e. it will be the CPCM Instance that ceases to be Domain Controller). The protocol shall be run with the Domain Controller designated in the message ADM Invite (DC Merge). If it is a Device Application request, the Discovery protocol shall first be run to determine with which Domain Controller the protocol will be run.
ADM_invite_message (DC Rebalance) or DC Rebalance request from Device Application	It shall start the DC Rebalance protocol (see clause 5.1.15) as a client (i.e. it will be the CPCM Instance that determines the new balances but not the one actually deciding on the re-balancing). The protocol shall be run with the Domain Controller designated in the message ADM Invite (DC Rebalance). If it is a Device Application request, the Discovery protocol shall be run first to determine with which Domain Controller the protocol will be run.
AD_join_begin_message	It shall start the AD Joining protocol (see clause 5.1.10) as a server.
AD_leave_begin_message	It shall start the AD Leaving protocol (see clause 5.1.11) as a server.
DC_transfer_begin_message	It shall start the DC Transfer protocol (see clause 5.1.12) as a server.
DC_rebalance_begin_message	It shall start the DC Rebalance protocol (see clause 5.1.15) as a server.
DC_merge_begin_message	It shall start the DC Merge protocol (see clause 5.1.14) as a server.
DC_split_begin_message	It shall start the DC Split protocol (see clause 5.1.13) as a server.
NOTE: Changing the AD name may also be performed by direct user interaction on the DC.	

When a Domain Controller has a recorded CPCM Instance Certificate Identifier, it may behave differently as described in clause 5.3.

If no CPCM Instance Certificate Identifier is recorded, *ADM_invite* messages with protocol status asserted are ignored.

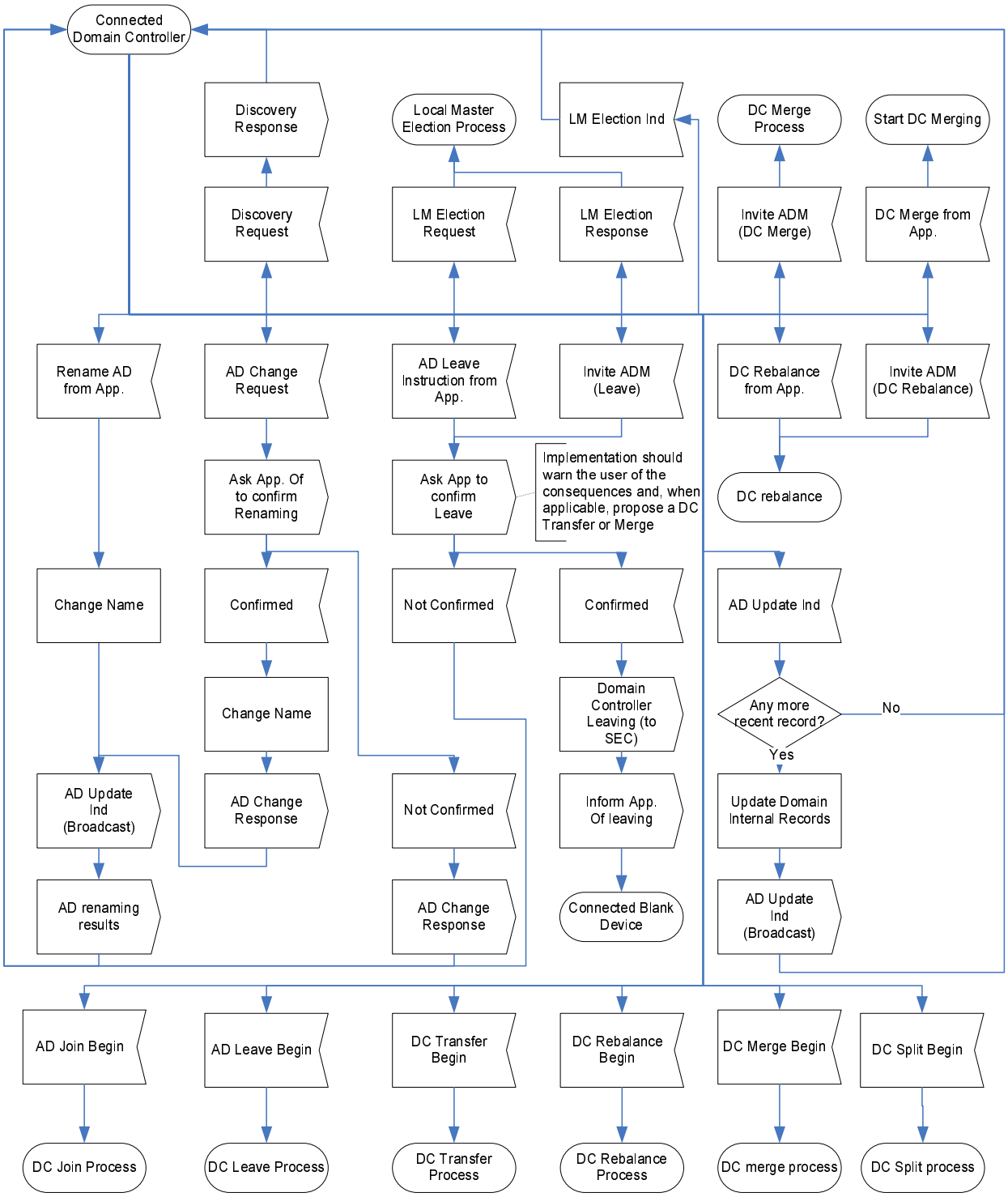

Figure 9: Connected Domain Controller State Machine

5.1.5 Connected Local Master

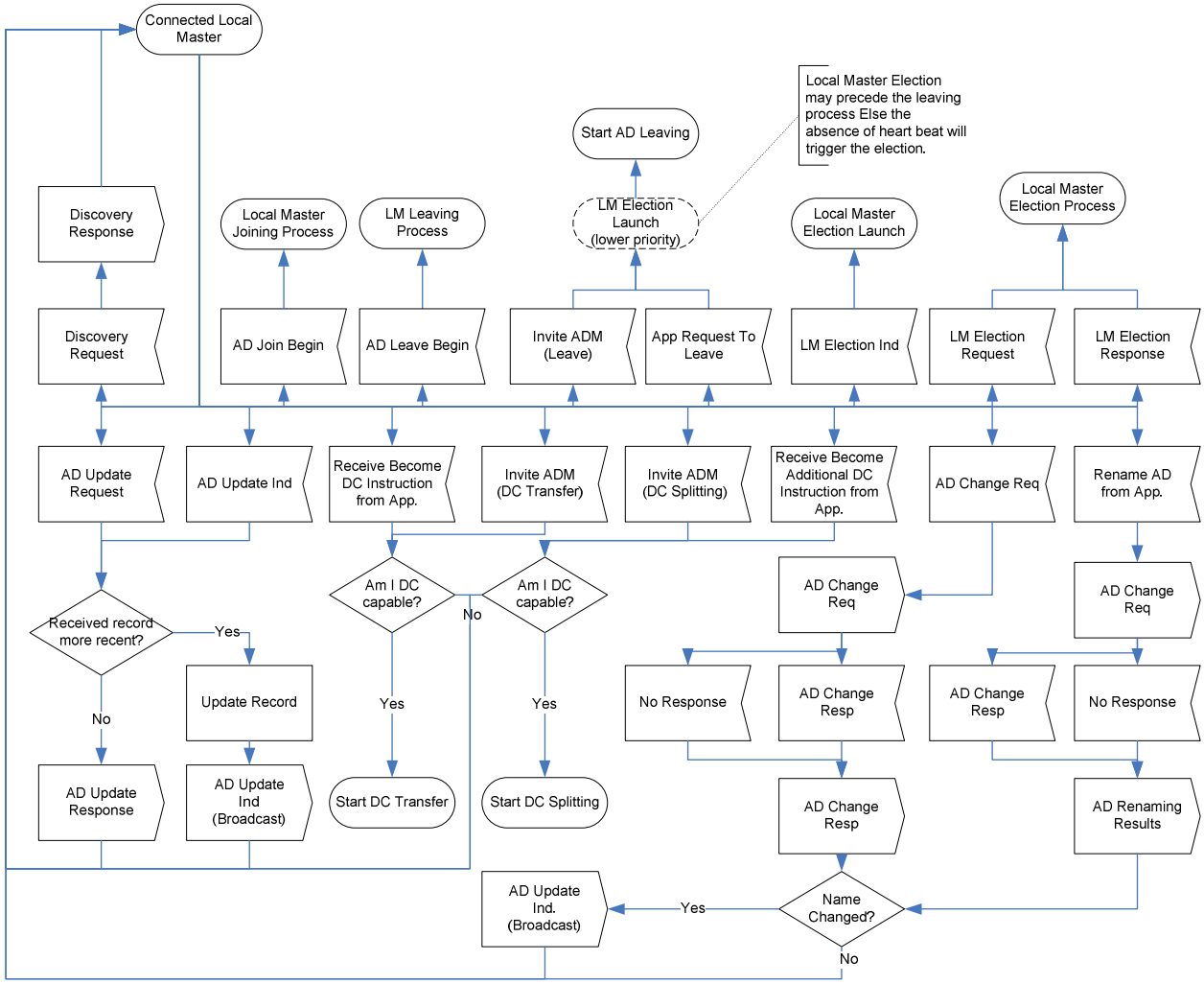

Figure 10: Connected Local Master State Machine

Table 6 shows how a connected Local Master (that is not Domain Controller) shall react when it receives the following messages.

Table 6: Actions by LM that is not DC member when receiving messages

Message	Action and reason
Discovery_request_message	It shall reply with the *AD_discovery_response_message* (see note 1).
AD_update_indication_message	It shall update, if needed, its Domain Internal Record and re-broadcast the *AD_update_indication_message* in order to forward the update indication to the local network in the event that it was initially received from a Remote Domain Controller (see note 2).
AD_update_request_message	It shall compare its current Domain Internal Record with the received ones. If the received records are more recent, then it shall update its internal records and broadcast an *AD_update_indication_message*. Else, then it shall respond to the requesting CPCM Instance *AD_update_response_message*. If the received records are less recent, this message carries up-to-date records.
LM_master_election_indication_message	It shall launch a new Local Master Election Process (see clause 5.1.7) (see note 3).
LM_master_election_request_message	It shall launch a new Local Master Election Process (see clause 5.1.7).
LM_master_election_response_message	It shall launch a new Local Master Election Process (see clause 5.1.7).
AD Leaving request from a Device Application or on invitation from another CPCM Instance	As it is the Local Master, it shall first launch a new Local Master Election Process (see clause 5.1.7) to avoid the AD being without a Local Master. Then, it shall start the AD Leaving protocol (see clause 5.1.11) with any of the Domain Controllers.
Request to become Domain Controller from a Device Application or on invitation from another CPCM Instance	If it is Domain Controller capable, it shall start the Domain Controller Transfer protocol (see clause 5.1.12) with the Domain Controller designed by the Device Application or the invitation. Else, if it is not DC capable, it shall inform the Device Application or CPCM Instance of the failure of the request.
Request to become an additional Domain Controller from Device Application or on invitation from another CPCM Instance	If it is Domain Controller capable, it shall start the Domain Controller Split protocol (see clause 5.1.13) with the Domain Controller designed by the Device Application or the invitation. Else, if it is not DC capable, it shall inform the Device Application or CPCM Instance of the failure of the request.
Change AD Name	It shall send an *AD_change_request_message* to the Domain Controller (as described in clause 5.1.2). If an *AD_change_response_message* is received, it shall forward this to the Device Application to inform it of the results of the renaming. It also broadcasts an *AD_update_indication_message* to inform other CPCM Instances of the renaming. If no response is received, it shall warn the Device Application that the renaming failed.
AD_change_request_message	It shall forward the message to the Domain Controller (as described in clause 5.1.2). If an *AD_change_response_message* is received, it shall forward this to the requesting Instance to inform it of the results of the renaming. It also broadcasts an *AD_update_indication_message* to inform other CPCM Instances of the renaming. If no response is received, it shall inform the requesting Instance that renaming is not possible via the *AD_change_response_message*.
AD_join_begin_message	It shall start the AD Joining protocol (see clause 5.1.10) as a server.
AD_leave_begin_message	It shall start the AD Leaving protocol (see clause 5.1.11) as a server.
NOTE 1: To enable protocol failure recovery, if the Discovery request includes a Domain Controller identifier, Local Master shall forward this message to the designated Domain Controller. If this specified Domain Controller is not available, the Local Master shall signal this to the requesting Instance in its response. NOTE 2: This re-broadcast may be skipped if it knows that the received message was issued by a Local Instance. NOTE 3: This message cannot normally be received if no *LM_master_election_request_message* or *LM_master_election_response_message* was previously received. However if the local network has two Local Masters, this can disturb the normal operation of this protocol, consequently by re-launching the Election Process it can ensure that there is subsequently only one Local Master.	

When the Local Master has a recorded CPCM Instance Certificate Identifier, it may behave differently as described in clause 5.3.

If no CPCM Instance Certificate Identifier is recorded, *ADM_invite* messages with protocol status asserted are ignored.

5.1.6 Connected Local Master and Domain Controller

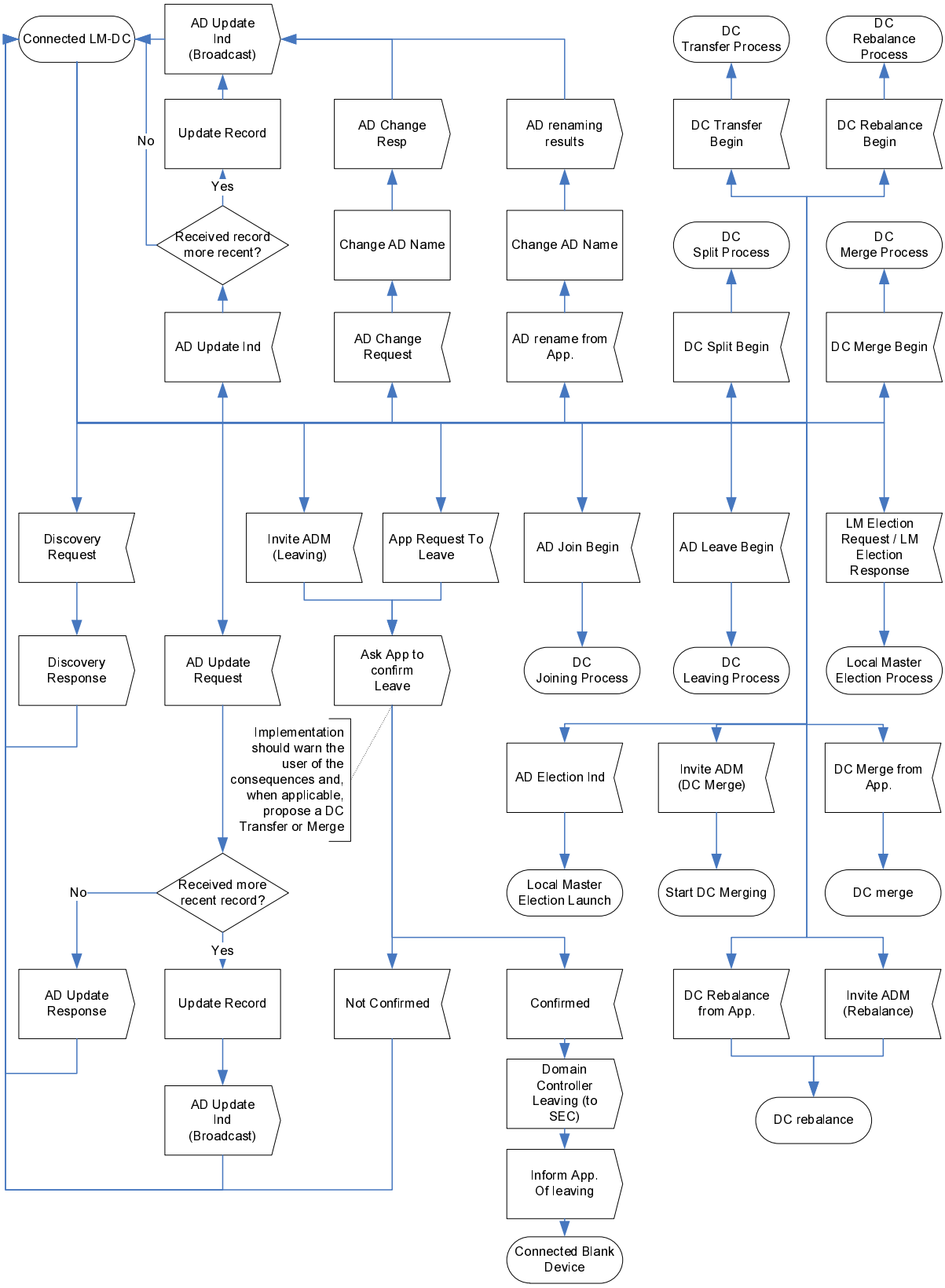

Figure 11: Connected Local Master and Domain Controller State Machine

Table 7 shows how a connected Local Master that is also a Domain Controller shall react when it receives the following messages.

Table 7: Actions by LM that is also a DC when receiving DC and LM messages

Message	Action and reason
Discovery_request_message	It shall reply with a *Discovery_response_message* (see note 1).
AD_update_indication_message	It shall update its Domain Internal Record if needed and re-broadcast the *AD_update_indication_message* in order to forward the updated indication to the local network in the event that it was initially received from a Remote Domain Controller (see note 2).
AD_update_request_message	It shall compare its current Domain Internal Record with the received ones. If the received records are more recent, then it shall update its internal records and broadcast an *AD_update_indication_message*. Else, it shall respond to the requesting CPCM Instance *AD_update_response_message*. If received records are less recent, this message carries up-to-date records.
LM_master_election_indication_message	It shall launch a new Local Master Election Process (see clause 5.1.7) (see note 3).
LM_master_election_request_message	It shall launch a new Local Master Election Process (see clause 5.1.7).
LM_master_election_response_message	It shall launch a new Local Master Election Process (see clause 5.1.7).
AD Leaving request from a Device Application or on invitation from another CPCM Instance	Firstly it shall request confirmation from the Device Application; since the Leaving of a Domain Controller is irreversible (all its ADSE values will be erased). Implementations may for instance first propose to Transfer the Domain Controller functionality to another Instance or to Merge it with another Domain Controller in which case it shall run the DC Transfer (see clause 5.1.12) or DC Merge protocol (see clause 5.1.14). If the Device Application confirms that the Instance shall cease to be a Domain Controller it requests its Security Control to proceed to the Domain Controller Leave protocol and confirms to the Device Application the success of the AD Leaving operation once it is complete.
Change AD Name	If the request came from another CPCM Instance through the *AD_change_request_*message, it shall request confirmation from the Device Application. If the Device Application confirmed or the request came from another Instance, it shall perform the following: • Rename the AD; • Inform the Device Application that the renaming occurred successfully; and • Broadcast an *AD_update_indication_message* to inform other Instances in the AD of the new name. Else, if the Device Application does not confirm, it shall inform the requesting CPCM Instance that the renaming was refused.
AD_join_begin_message	It shall start the AD Joining protocol (see clause 5.1.10) as a server.
AD_leave_begin_message	It shall start the AD Leaving protocol (see clause 5.1.11) as a server.
DC_merge_begin_message	It shall start the DC Merge protocol (see clause 5.1.14) as a server.
DC_split_begin_message	It shall start the DC Split protocol (see clause 5.1.13) as a server.
DC_rebalance_begin_message	It shall start the DC Rebalance protocol (see clause 5.1.15) as a server.
DC_transfer_begin_message	It shall start the DC Transfer protocol (see clause 5.1.12) as a server.
ADM_invite_message (DC Merge) or a DC Merging request from Device Application	It shall start the DC Merging protocol (see clause 5.1.14) as a client. The protocol shall be run with the Domain Controller indicated in the ADM Invite (DC Merge) message.
ADM_invite_message (DC Rebalance) or DC Rebalance request from Device Application	It shall start the DC Merging protocol (see clause 5.1.14) as a client. The protocol shall be run with Domain Controller indicated in the ADM Invite (DC Rebalance) message.

NOTE 1: To enable recovery from protocol failures, the Local Master and Domain Controller shall forward this message to the designated Domain Controller, if any, and if it is not itself designated. If this specified Domain Controller is not available, the Local Master shall signal this to the requesting Instance in its response.
NOTE 2: This re-broadcast may be skipped if it knows that the received message was issued by a Local Instance.
NOTE 3: This message cannot normally be received if no *LM_master_election_request_message* or *LM_master_election_response_message* was previously received. However if the local network has two Local Masters, this can disturb the normal operation of this protocol, consequently by re-launching the Election Process it can ensure that there is subsequently only one Local Master.

When the Local Master and Domain Controller have a recorded CPCM Instance Certificate Identifier, it may behave differently as described in clause 5.3.

If no CPCM Instance Certificate Identifier is recorded, *ADM_invite* messages with protocol status asserted are ignored.

5.1.7 Local Master Election

The function and operation of the Local Master are described in clause 4.2.1.

If a CPCM Instance is Local Master capable, it shall enter into election mode in two main circumstances:

- the CPCM Instance is the Local Master Election initiator (e.g. because it has detected that no Local Master was present). In which case, it broadcasts an *LM_master_election_request_message*; or
- CPCM Instance receives an *LM_master_election_request_message* or *LM_master_election_response_message*, as a consequence of another Instance launching the Election Process. In which case, the CPCM Instance compares its LM capability to the one received in the message. If its LM capability is greater, it broadcasts an *LM_master_election_response_message*. Else, it will not be elected as Local Master and waits for the Election to end.

NOTE: The case may occur where network latency causes the *LM_master_election_response_message* to be received before the initial request.

If the CPCM Instance has broadcast an *LM_master_election_request_message* or an *LM_master_election_response_message*, it shall wait a reasonable interval to permit all other devices to respond (see TR 102 825-12 [i.5]). During this waiting period, each time it receives an *LM_master_election_request_message* or *LM_master_election_response_message*, it shall compare its LM capability to the one received in the message. If its own capability is lower, this instance will not be elected Local Master, so and it can simply wait for the Election to end. If it is greater, this instance is still a possible winner and it shall continue waiting for possible responses.

When the waiting time is over and if the Instance does not wait for the Election to end, it shall:

- Become the new Local Master.
- Launch a Discovery Protocol to locate the connected Domain Controllers (including Remote ones).
- Broadcast an *LM_master_election_indication_message* to warn all other connected Instances that it is the new Local Master.
- An Instance waiting for the Election to end shall go revert to its initial state as soon as the waiting time is over or an *LM_master_election_indication_message* is received.

The capability rating is based on CPCM Instance version, capabilities, on the role of the Instance at the time of the Election Process and on the CPCM instance certificate identifier. An upper and a lower capability rating may be used when the Instance wants to become the Local Master or to dismiss the Local Master role. The upper and lower capabilities shall not be used under other circumstances.

If the CPCM Instance Requesting the Election is not LM capable, it carries the lowest LM capability. If no other LM capable is present, it will receive no responses and the LM Election will stop there.

Unless upper or lower capabilities are used, the CPCM Instance with a greater CPCM Version is always deemed to have a higher capability; otherwise, capabilities are rated as shown in table 8.

Table 8: Local Master capabilities

Capability Rating	LM capable Instance type
0	Instance is LM capable but does not want to take Local Master Role
1	Local Master capable Instance
2	Domain Controller capable Instance
3	Current Domain Controller
4	Instance needs to get the Local Master Role

If several Domain Controllers from the same AD are present (and no Instance sets its capability to the higher level), their respective capacities are assessed in the following way:

- The Domain Controller that has the greater capacity to let Instances Join the AD Locally (i.e. with the greater `local_ceiling - local_count`) has greater capacity.
- If the Local capacities are equal, or both are negative or zero, the capacity to let Instances Join the AD Remotely is considered (i.e. `remote_ceiling - remote_count`).
- If both Remote capacities are equal or both are zero, then the total capacity to let Instances Join the AD (i.e. `total_ceiling - total_count`) is considered.

In all other cases, the CPCM Instance with the highest CIC identifier is deemed to have the greater capability.

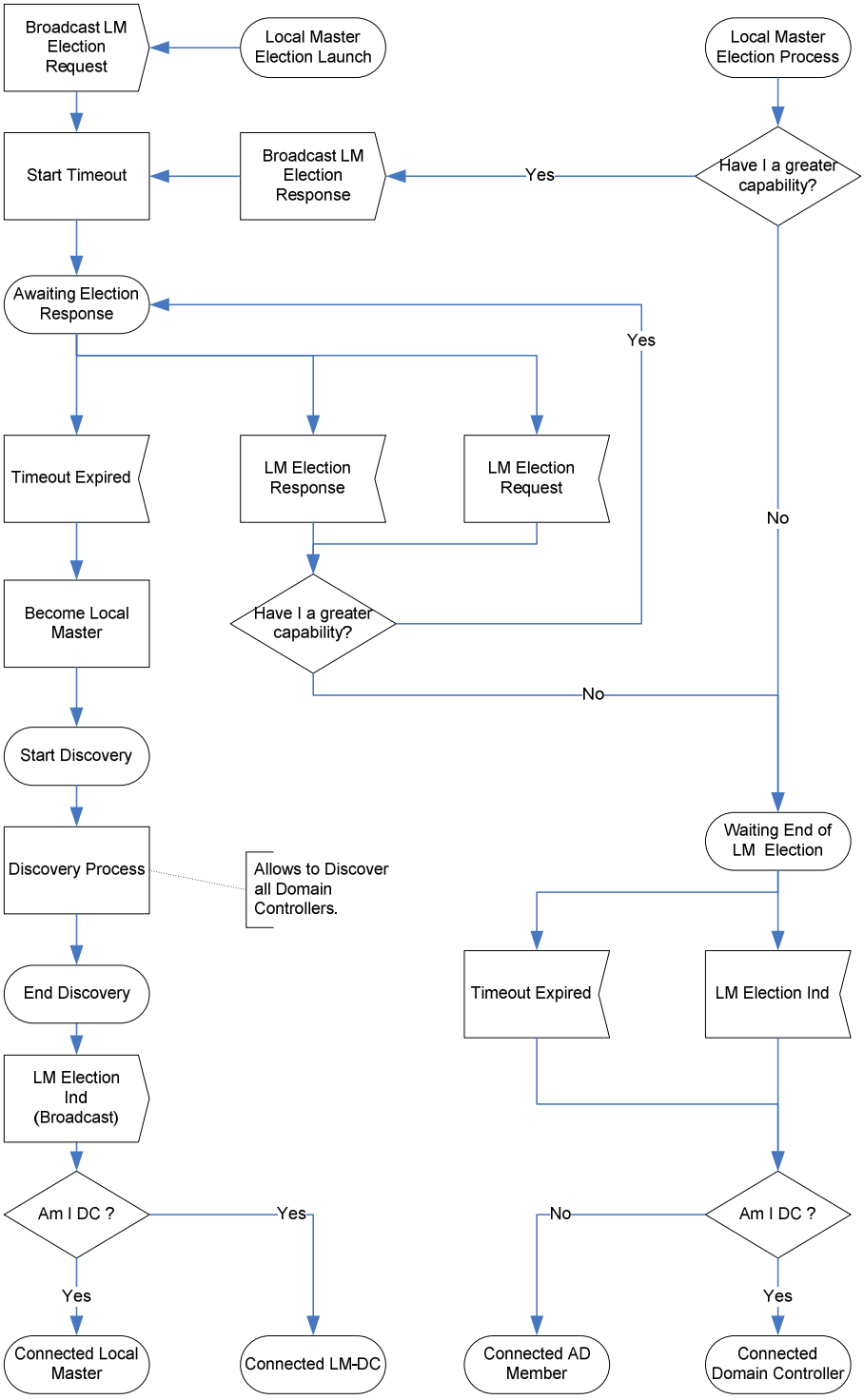

Figure 12: Local Master Election

5.1.8 AD Member Reconnection

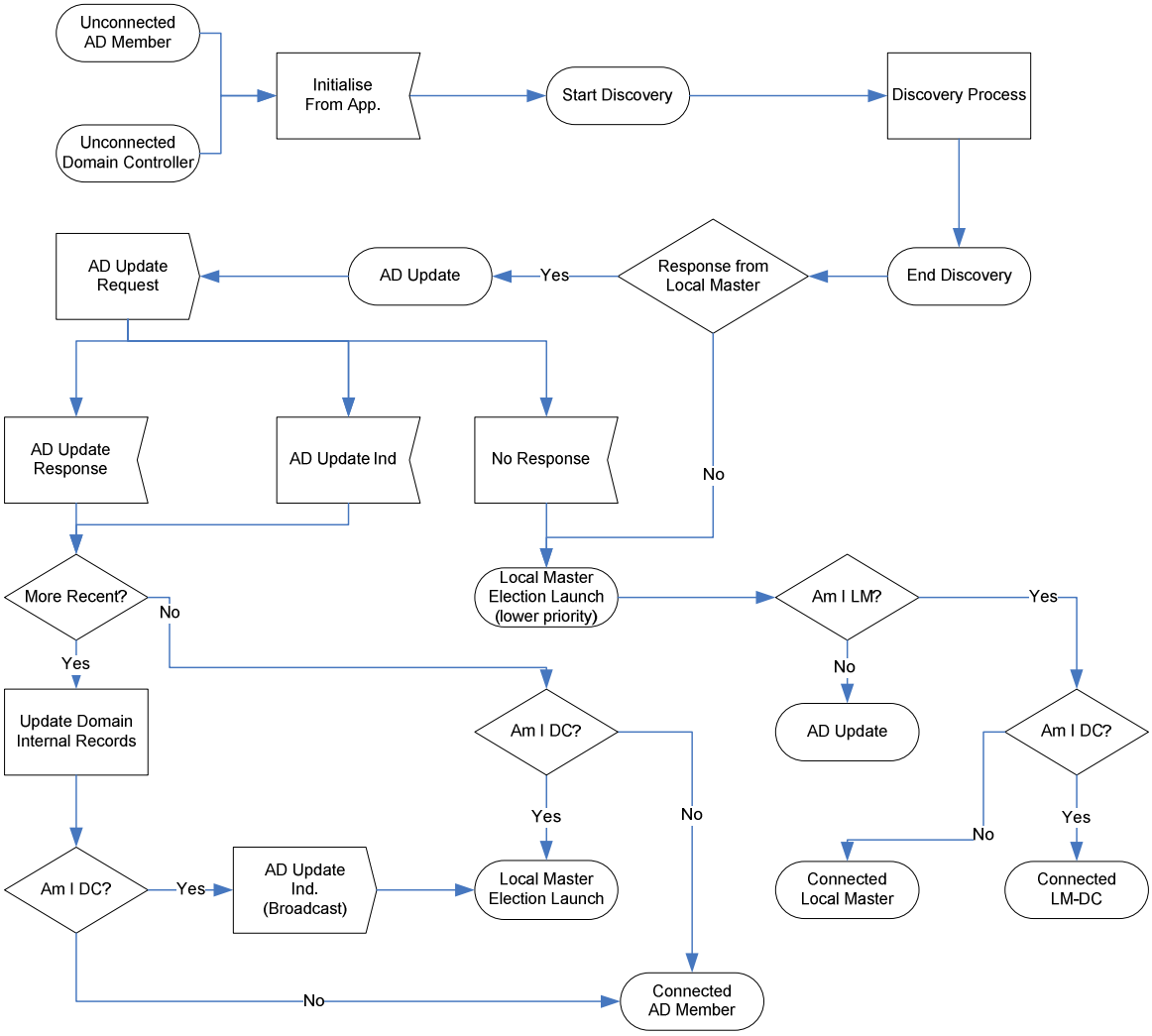

Figure 13: AD Member or Domain Controller Connection/AD Update

When an AD Member makes a network reconnection, it is called a reconnecting CPCM Instance and it shall perform the following:

- First it shall perform the Discovery Protocol.
- If the reconnecting CPCM Instance does not discover a Local Master from the same AD, it shall launch a new Election Process, setting its capability at the lowest level ("Instance is LM capable but does not want to take the Local Master Role"). This is to avoid becoming Local Master before the reconnecting instance has received up-to-date domain information. If it is elected notwithstanding, this means that the reconnecting CPCM Instance is the only connected AD member and no AD update is required.
- If the reconnecting CPCM Instance discovered a Local Master, the reconnecting CPCM Instance shall proceed to perform an AD Update by sending an *AD_update_request_message* to the Local Master. If the reconnecting CPCM Instance receives an *AD_update_response_message* with a more recent record, then it updates its records. If the reconnecting CPCM Instance has updated its records and is a Domain Controller, it shall broadcast an *AD_update_indication_message* and launch a new Local Master Election. Else, the reconnecting CPCM Instance goes to Connected AD Member state.
- If the reconnecting CPCM Instance receives an *AD_update_response_message* or an *AD_update_indication_message* with less recent records, it may launch a new Election Process if it is a Domain Controller with greater capabilities than the current Local Master.

NOTE: If the reconnecting CPCM Instance has a recorded CPCM Instance Certificate Identifier and if it discovers the corresponding CPCM Instance, and if it was the client for the interrupted protocol, it should restart the interrupted protocol, possibly following confirmation from the Device Application.

5.1.9 Blank Instance Connection

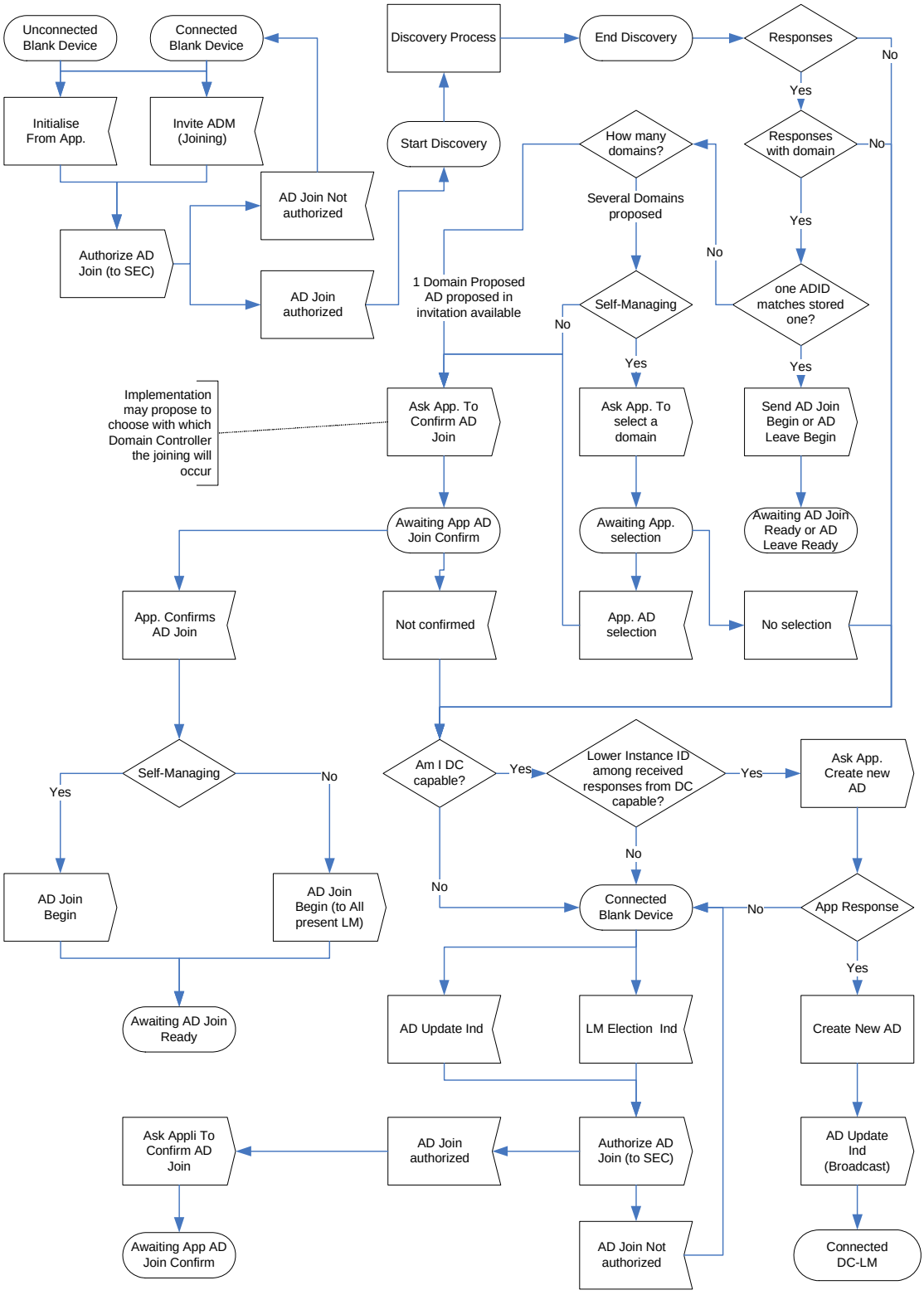

Figure 14: Blank Instance Connection / AD Creation

Upon connection or upon *ADM_invite_message* (Join AD) message with protocol status not asserted, a Blank Instance first asks its Security Control whether it can Join an AD or not. If not, its status remains a Blank Instance. Else, it shall perform the Discovery Protocol as described in clause 5.1.1.

The Blank connecting Instance shall analyse the responses from the Discovery Protocol to this request in the following way as depicted in figure 14:

- If no response was received from Local Masters, Domain Controllers or Blank Instances, and the Blank connecting Instance is DC capable, it shall attempt to create a new AD. If creation is confirmed by the Device Application, an AD creation is requested from its Security Control and the Blank connecting Instance becomes the Local Master and Domain Controller and it shall broadcast an *AD_update_indication_message*. If the Device Application refuses, or if the Blank connecting Instance is not DC capable, its AD status remains as a Blank Instance.

- If only responses from Blank Instances are received, and if the Blank connecting Instance is DC capable, it shall compare the received identifiers (from DC capable Instances) with its own one. If it has the lowest identifier amongst DC capable Blank Instances, the Device Application shall attempt to create a new AD as above. Otherwise, or if the Blank connecting Instance is not DC capable its AD status remains as a Blank Instance.

- If one or more responses from AD Members are received and one ADID matches the one recorded by the Security Control (see clause 5.3), then the CPCM Instance sends an *AD_join_begin_message* or an *AD_leave_begin_message* (depending which protocol was performed) carrying the recorded CPCM Instance Certificate Identifier (if the corresponding CPCM Instance was not discovered; else it sends the message to that CPCM Instance).

- Else, the Blank connecting Instance asks the Device Application whether it may Join an AD. The proposal may be a simple request for confirmation (if there is only one AD proposed, if the device has a reduced user interface or if the AD requested in ADM Invite was discovered) or a list of available ADs. If the Device Application refuses to Join any of the proposed ADs, then the Blank connecting Instance behaves as if no response was received (see above). Otherwise, the AD Join process shall start and the Blank connecting Instance shall send an *AD_join_begin_*message to a Local Master or a Domain Controller of the selected AD, if any. The message may be multicast to all connected Local Masters and Domain Controllers for Instances with a reduced user interface (since AD selection could not occur). If no Local Master or Domain Controller is present, CPCM AD Instance remains Blank.

A connected Blank Instance that can Join an AD may re-propose an AD Join to its Device Application upon receipt of an *LM_master_election_indication_message* or an *AD_update_indication_message* message. If the Device Application does not confirm and if the Instance is DC capable, it may re-propose to create an AD. Else, its status remains Blank. If the Device Application confirms the AD Join, then the AD Join process shall start. If the Device Application confirms the AD creation, it shall ask its Security Control to proceed to the creation.

If an *AD_join_begin_message* was sent, the CPCM Instance waits for an *AD_join_ready_message* and the corresponding behaviour is described in clause 5.1.10.

5.1.10 AD Joining

5.1.10.1 Blank Instance Joining an AD

In order to become a Connected AD Member, the Blank Joining Instance shall perform the following steps as depicted figure 15:

- It shall send an *AD_join_begin_message* to the Local Master that it has already found using the Discover process (see clause 5.1.1).

- If no *AD_join_ready_message* response is received, the Blank Joining Instance shall send a *transaction_rollback_message* to all contacted Local Masters and revert to the connected Blank Instance state.

NOTE: Implementers are strongly advised to take into account that it may take some time for the Blank Joining Instance to receive a response from the Domain Controller, perform a Quorum Test, perform proximity controls, etc.

- If exactly one *AD_join_ready_message* response is received, it requests authorization to the Device Application to proceed to the Joining. The same is done in the two following cases:
    - If the device is not self-managing and all *AD_join_ready_message* responses are received from the same AD. The protocol may continue with any of the responding CPCM Instance and a *transaction_rollback_message* shall be sent to all other responding Instances.
    - If the device is self-managing and a response is received from the AD to which the *AD_join_begin_message* was sent.

  In all other cases where more than one *AD_join_ready_message* response is received then this indicates that there was a protocol problem or perhaps an attack is underway, in any case the Device Application shall stop and the AD Join will not be performed, the Blank Joining Instance shall send a Secured *transaction_rollback_message* to all responding Local Masters and revert to the connected Blank Instance state.

- If the Device Application confirms the AD Join, then *Prepare_AD_join* and *AD_join_commit_message* shall be issued. Otherwise, if the Device Application or Security Control does not confirm the AD Join, it shall send a Secured *transaction_rollback_message* and revert to the connected Blank Instance state.

- If no *AD_join_confirm_message* response is received, the Blank Joining Instance shall revert to the connected Blank Instance state.

- Finally, upon receipt of an *AD_join_confirm_message*, the Blank Joining Instance shall enable the AD secret and issue an *AD_join_finish_message*. It is now an AD member.

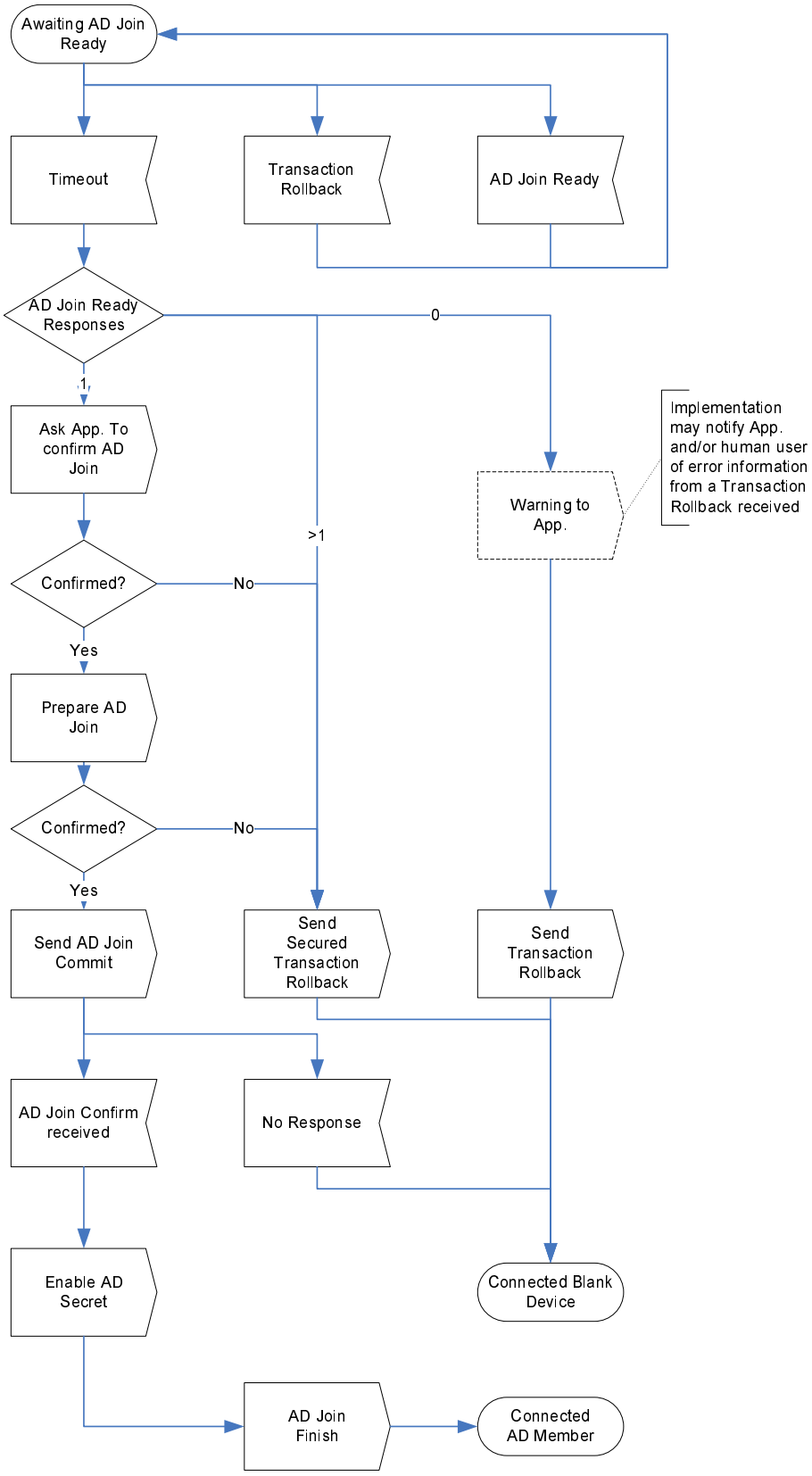

Figure 15: Blank Instance Joining an AD

5.1.10.2 Local Master in AD Joining

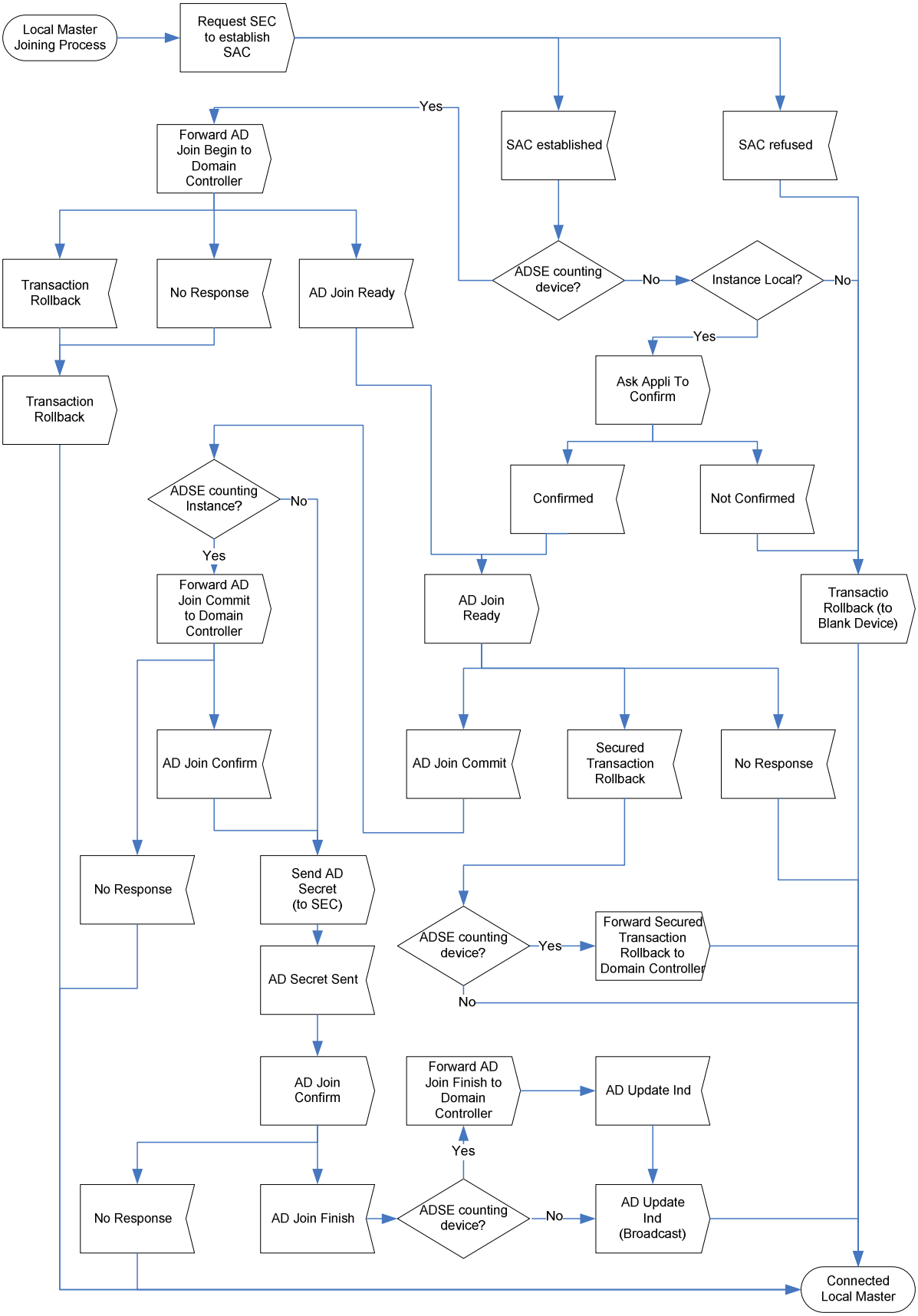

Figure 16: Local Master in AD Joining Process

When receiving an AD Join Request, a Local Master that is not a Domain Controller first checks the AD capability of the requesting CPCM Instance:

- If the requesting CPCM Instance is not ADM capable, the AD Join is refused, the Local Master sends a *transaction_rollback_message* to the requesting CPCM Instance and the process ends.
- If the requesting CPCM Instance IS NOT ADSE countable, then the Local Master establishes a SAC with the requesting Instance, tests whether the requesting Instance is Local and processes the AD Join by requesting its Security Control to deliver the AD Secret.
- If the requesting CPCM Instance IS ADSE countable, the Local Master acts as proxy for a Domain Controller regarding ADSE enforcement. Any Domain Controller may be contacted to perform this enforcement, unless a specific Domain Controller is identified in the request. If this Domain Controller is not reachable, the Local Master signals this to the requesting Instance through an *AD_join_ready_message*. The same occurs if no specific Domain Controller is requested, and no Domain Controller is available.

The detailed process is as follows for a requesting CPCM Instance that IS NOT ADSE countable:

- First, the Local Master checks with the Device Application to confirm the AD Join.
- If the Device Application does not confirm, then the Local Master informs the requesting CPCM Instance through a *transaction_rollback_message*. Else, the Local Master requests its Security Control to establish a SAC with the requesting Instance and to perform a Proximity Test. If the SAC cannot be established or the Instance is Remote, then a *transaction_rollback_message* is sent to the requesting Instance and the process ends.
- The Local Master sends an *AD_join_ready_message* to the requesting CPCM Instance to inform it that the AD Join may happen and the Local Master waits for a response from the requesting Instance. If no valid response is received, or if a Secured *transaction_rollback_message* is received, the Local Master reverts to its original state. This means the Joining CPCM Instance shall need to re-start the protocol to be able to Join the AD.
- When an *AD_join_commit_message* has been received. The Local Master requests its Security Control to deliver the AD Secret.
- If the delivery is successful, the Local Master sends an *AD_join_confirm_message* to the requesting CPCM Instance. Else, the Local Master reverts to its original state as above.
- Upon receipt of an *AD_join_finish_message*, the Local Master reverts to the Connected Local Master state.

As for an ADSE countable CPCM Instance, the process is as follows:

- The Local Master requests its Security Control to establish a SAC with the requesting CPCM Instance. If the SAC cannot be established, the Local Master informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- The Local Master forwards the received *AD_join_begin_message* to the Domain Controller, if available. If no response is received or if a *transaction_rollback_message* is received, the Local Master may retry with the same or a different Domain Controller (not shown in the figure). If the result is the same, the Local Master sends a *transaction_rollback_message* to the requesting CPCM Instance and the process ends.

NOTE: Implementers are strongly advised to take into account that it may take some time for the Local Master to receive a response due to establishing a SAC, performing a Quorum Test, proximity controls, etc.

- The Domain Controller authorizes the AD Join by sending an *AD_join_ready_message* (through a SAC that may need to be established with the Domain Controller).
- The Local Master sends an *AD_join_ready_message* to the requesting CPCM Instance. If no response is received, the Local Master reverts to its original state. If a Secured *transaction_rollback_message* is received, the Local Master sends a Secured *transaction_rollback_message* to inform the Domain Controller that the AD Join will not happen and the process ends.
- When the *AD_join_commit_message* is received, the Local Master sends the same message to the Domain Controller. If the Domain Controller does not answer, nothing more needs to be done.

- The Domain Controller applies the ADSE enforcement and sends an *AD_join_confirm_message*.
- The Local Master requests its Security Control to deliver the AD Secret.
- If the delivery is successful, the Local Master sends an *AD_join_confirm_message* to the requesting CPCM Instance.
- When the *AD_join_finish_message* is received, the Local Master forwards the message the Domain Controller that responds with an *AD_update_indication_message _message*.
- When the *AD_update_indication_message* is received, the Local Master broadcasts it to the locally connected CPCM Instances.

5.1.10.3 Domain Controller in AD Joining

In order for the Domain Controller to be able to let a new Instance Join the AD, it shall perform the following tasks as depicted in figure 17.

If the Domain Controller is also a Local Master, it may receive an AD Join request designating another Domain Controller. In that case, the Joining Domain Controller shall act as a Local Master for the other designated Domain Controller (see clause 5.1.10.2).

When receiving an AD Join request, a Domain Controller (whether Local Master or not) performs the following:

- If the requesting Instance is not ADM capable, the Domain Controller shall refuse the AD Join and issue a *transaction_rollback_message* and the process ends.
- The Domain Controller requests its Security Control to establish a SAC with the requesting CPCM Instance or, if the message is delegated, with the Local Master. If the SAC cannot be established, it issues a *transaction_rollback_message*.
- If the requesting Instance is ADSE countable, the Domain Controller first informs its Security Control of the AD Join request. If the Security Control refuses the AD Join, it responds with a *transaction_rollback_message* and the process ends.
- If the requesting Instance is not ADSE countable, the Domain Controller checks the requesting Blank Instance is Local. If it is Remote, it responds with a *transaction_rollback_message* and the process ends.
- Next, the Domain Controller asks the Device Application to confirm the AD Join. If the Device Application does not confirm, then the Domain Controller informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.

  If the message *AD_join_begin_message* was broadcast, the confirmation shall come from an end-user action. The requesting CPCM Instance is actually not self-managed and multiple *AD_join_ready_message* responses will result in an error for the requesting CPCM Instance and stop the AD Join protocol. Thus, only one Domain Controller, designated by the end-user, shall respond to the request.
- It sends an *AD_join_ready_message* to the requesting CPCM Instance to inform it that the AD Join may happen.
- The Domain Controller then waits for an *AD_join_commit_message* response from the requesting Instance. If no *AD_join_commit_message* is received, or if a Secured *transaction_rollback_message* is received, the Domain Controller reverts to its original state and the process ends. This means that the CPCM Instance shall have to re-start the protocol in order to be able to AD Join.
- Having received the *AD_join_commit_message*, the Domain Controller requests its Security Control to proceed to the AD Join. If the message is not delegated, Security Control will proceed directly to the AD Join. If delegated, Security Control will only update the ADSE values. If this operation fails, the Domain Controller reverts to its original state and the process ends.
- The Domain Controller sends an *AD_join_confirm_message* to the requesting CPCM Instance which responds with an *AD_join_finish_message*. If no message is received, the Domain Controller reverts to its original state and the process ends.

- If the Joining Instance was ADSE countable, the Domain Controller informs its security control of the protocol success and broadcasts an `AD_update_indication_message` to inform other connected CPCM Instances of the AD Join.

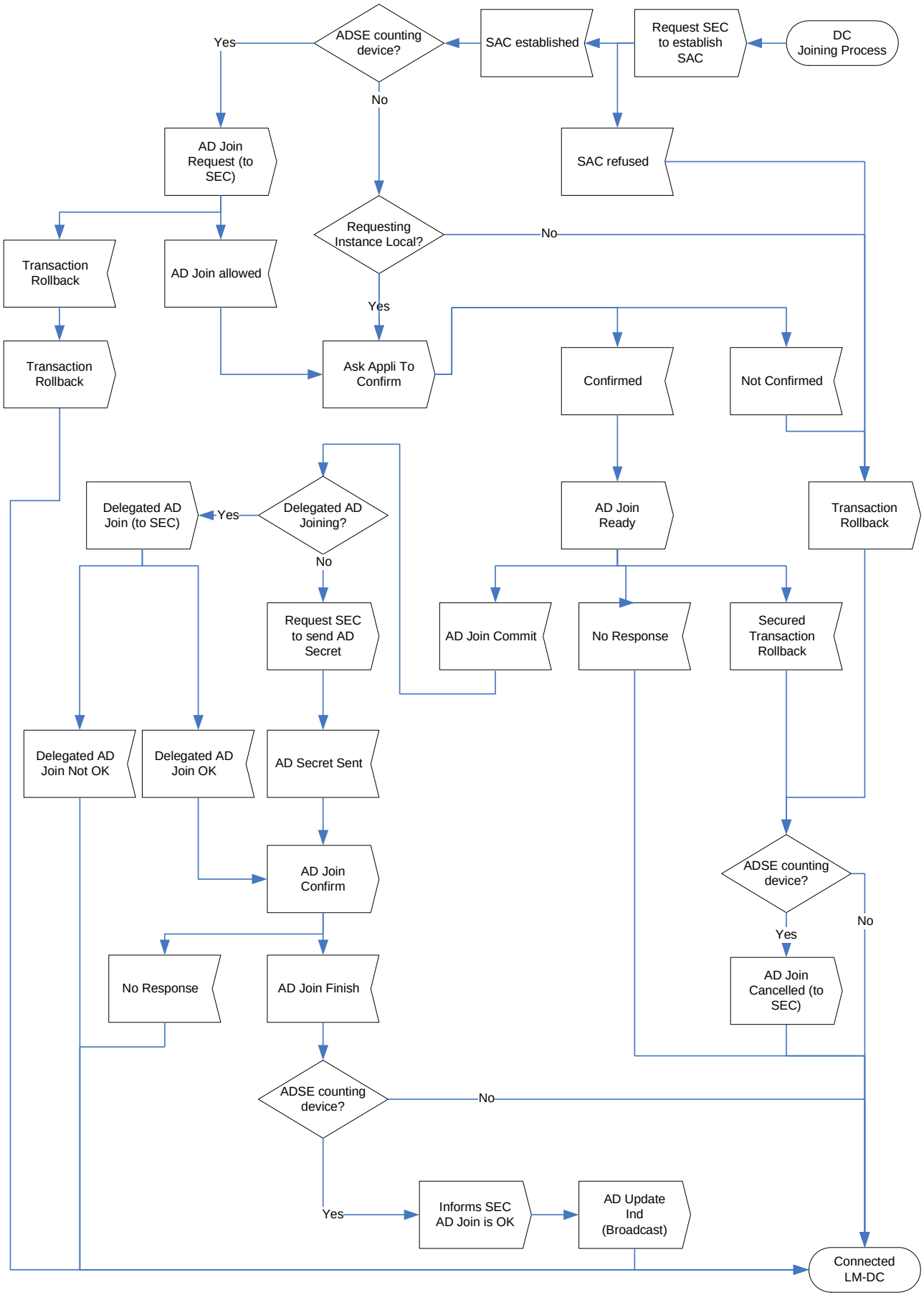

Figure 17: Domain Controller in AD Joining Process

5.1.11 Instance Leaving the AD

5.1.11.1 AD Member Leaving

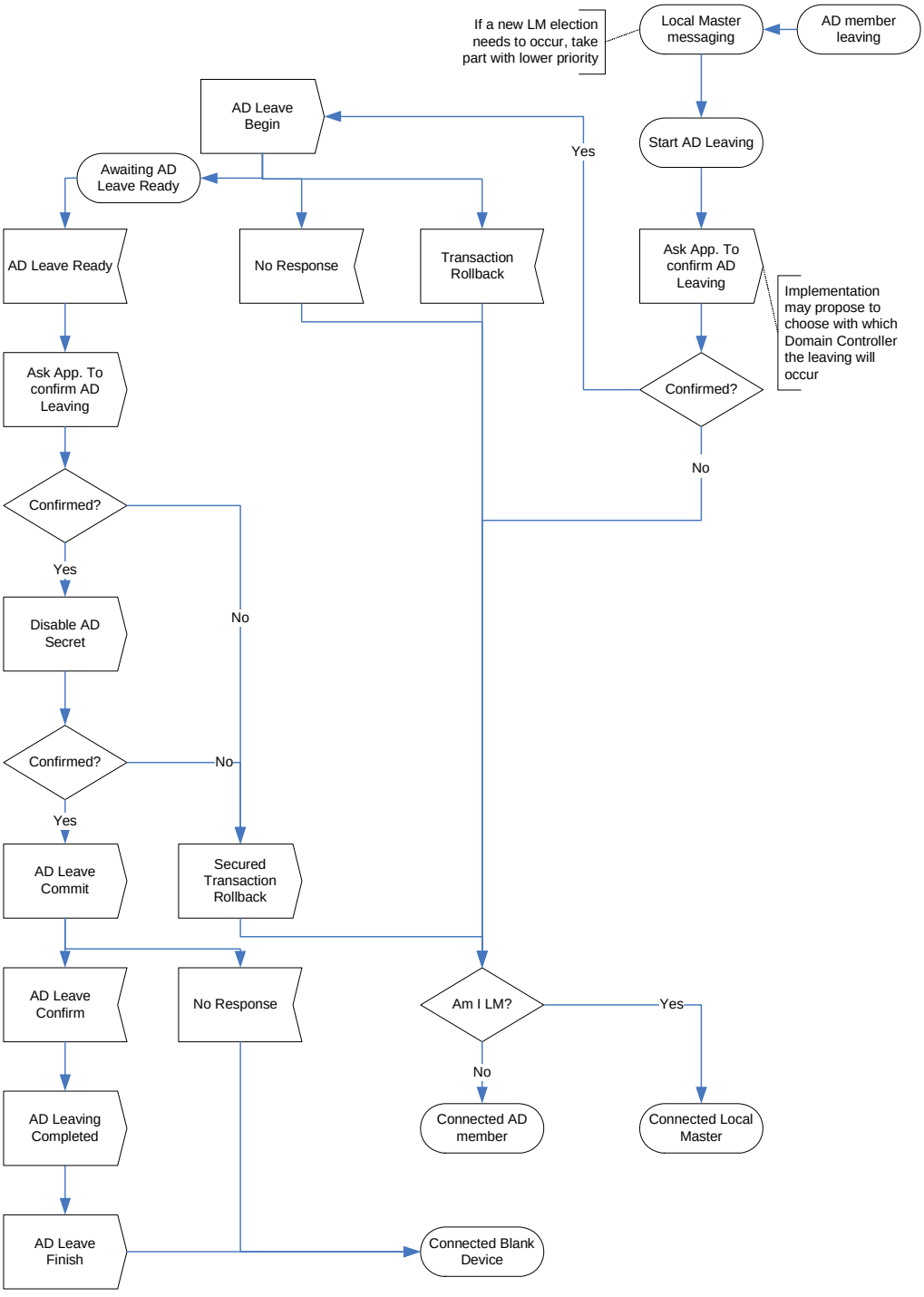

Figure 18: AD Member Leaving

A Connected AD member that wishes to Leave the AD shall perform the following steps as depicted in figure 18:

- The Leaving Member instance shall first discover the Local Master and, if necessary, perform a new Local Master Election with its own capability set to lower level.
- Next the Device Application is asked to confirm that the Leaving Member still wishes to Leave. An implementation may offer the Leaving Member Device Application the option to decide which Domain Controller it wishes to Leave (unless a specific Domain Controller shall be requested).

- If the confirmation fails, then the Leaving Member instance reverts to its original state, i.e. if it was a Local Master, then it remains the connected Local Master, if not, then it remains a Connected AD Member.
- Assuming the confirmation succeeds, the Leaving Member sends an *AD_leave_begin_message* to the Local Master.
- If no *AD_leave_ready_message* response is received, or an error occurs, the Leaving Member instance reverts to its original state as above.
- Next, the Leaving Member Device Application shall be asked to confirm that it still wishes to Leave the AD.
- If the Leaving Member Device Application confirms the Leave, then the AD Secret is disabled, and an *AD_leave_commit_message* is issued. The Instance is not yet a Blank Instance but will no more be able to access AD bound content. If the Leaving Member Device Application declines the Leave or the Security Control cannot disable the AD Secret, it shall send a Secured *transaction_rollback_message* and the Instance returns to its original state as above.
- If no *AD_leave_confirm_message* response is received, the Leaving Member shall become a Blank instance and shall revert to the connected Blank Instance state.
- Finally it informs Security Control that the Leave is complete, issues an *AD_leave_finish_message* and becomes a Connected Blank Instance.

5.1.11.2 Local Master in AD Leaving

In order for the Local Master to enable another CPCM Instance to Leave the AD, it shall perform the following tasks as depicted in figure 19.

When receiving an AD Leave request, a Local Master that is not a Domain Controller first checks whether the requesting Instance is from the same AD. If not, AD Leaving is refused through a *transaction_rollback_message*. It then checks whether the requesting CPCM Instance is ADSE countable or not:

- If the requesting CPCM Instance IS NOT ADSE countable, then the Local Master processes the AD Leave by requesting its Security Control to establish a SAC and then to deliver the AD Secret.
- If the requesting CPCM Instance IS ADSE countable, then the Local Master acts as proxy for a Domain Controller regarding ADSE enforcement. If no Domain Controller is indicated in the message, then any Domain Controller may be contacted to perform this enforcement. Otherwise, the Local Master shall contact the indicated Domain Controller, if present, or reply with an *AD_leave_ready_message* indicating the Domain Controller is absent. If the Domain Controller authorizes the AD Leave, then the Local Master requests its Security Control to establish a SAC and then to deliver the AD Secret.

The detailed process is as follows for a requesting CPCM Instance that IS NOT ADSE countable:

- The Local Master asks the Device Application to confirm the AD Leave. If the Device Application does not confirm, then the Local Master informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- The Local Master requests its Security Control to establish a SAC with the requesting Instance. If the SAC cannot be established, a *transaction_rollback_message* is sent to the requesting Instance and the process ends.
- The Local Master sends an *AD_leave_ready_message* to the requesting CPCM Instance to inform it that the AD Leave may happen.
- The Local Master waits for an *AD_leave_commit_message* from the requesting Instance. If no *AD_leave_commit_message* is received, or if a Secured *transaction_rollback_message* is received, the Local Master reverts to its original state and the requesting CPCM Instance shall have to re-start the protocol in order to be able to Leave.
- The Local Master requests its Security Control to proceed to the AD Leave. Upon confirmation, it sends an *AD_leave_confirm_message* to the requesting CPCM Instance. Else, nothing more needs to be done.
- When the Local Master receives the *AD_leave_finish_message*, it assumes the Connected Local Master state.

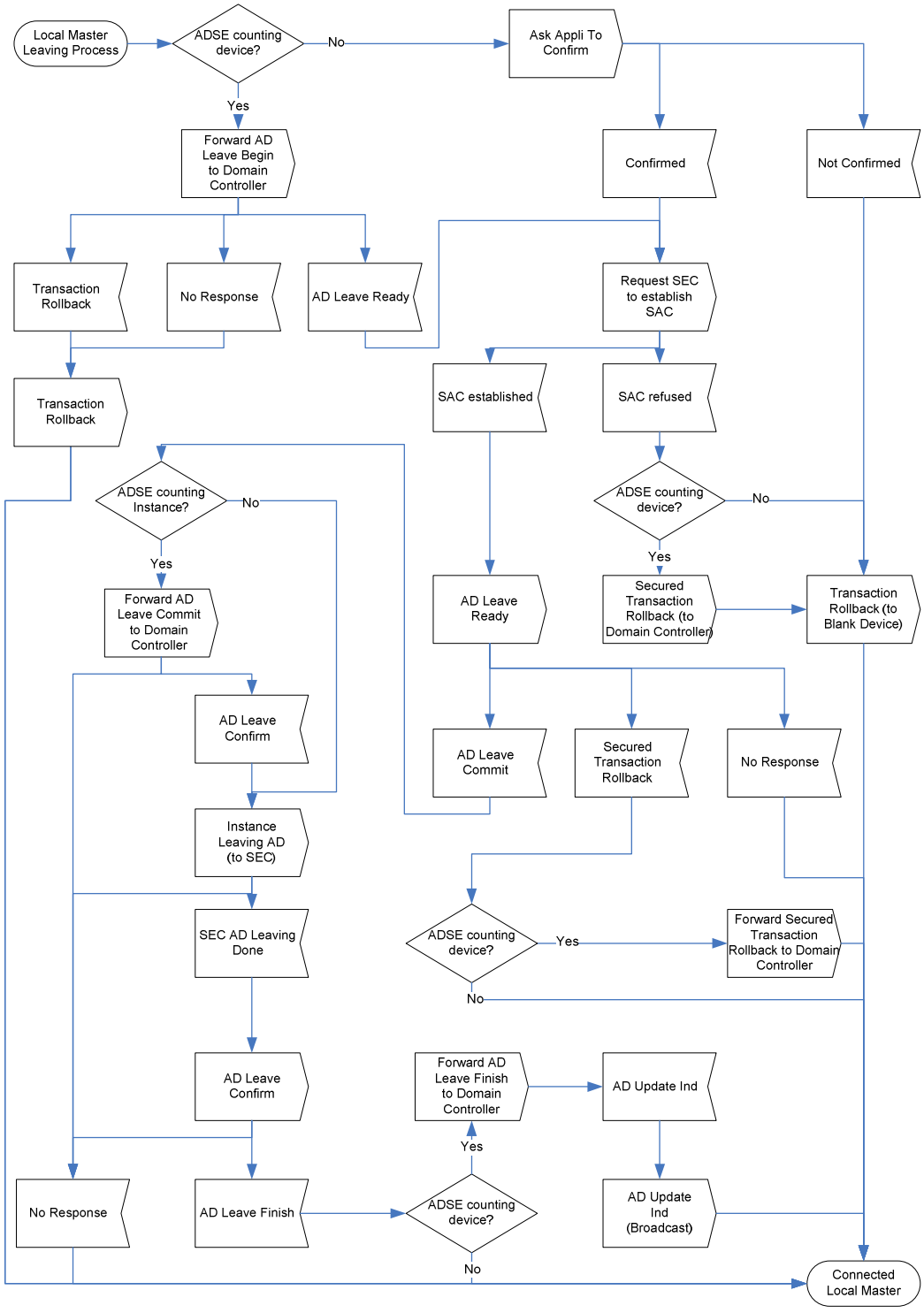

Figure 19: Local Master in AD Leaving Process

As for an ADSE countable CPCM Instance, the process is slightly different:

- The Local Master requests its Security Control to establish a SAC with the requesting CPCM Instance. If the SAC cannot be established, it issues a `transaction_rollback_message` to the requesting CPCM Instance.
- The Local Master forwards the `AD_leave_begin_message` to the Domain Controller. If no response is received or if a `transaction_rollback_message` is received, the Local Master may retry with the same or a different Domain Controller (not shown in the figure). If the result is the same, it sends a `transaction_rollback_message` to the requesting CPCM Instance and the process ends.

- The Domain Controller authorizes the AD Leave through an *AD_leave_ready_message* (through a SAC that may need to be established with the Domain Controller).
- The Local Master sends an *AD_leave_ready_message* to the requesting CPCM Instance. If no response is received, it reverts to its original state. If a *Secured* *transaction_rollback_message* is received, it sends a Secured *transaction_rollback_message* to inform the Domain Controller the AD Leave will not happen.
- When the *AD_leave_commit_message* is received, the Local Master forwards this to the Domain Controller. If the Domain Controller does not answer, nothing more needs to be done.
- The Domain Controller performs the ADSE enforcement and issues an *AD_leave_confirm_message*.
- The Local Master requests its Security Control to proceed to erase the AD Secret of the requesting CPCM instance. If the erase fails, then the process stops.
- Once the AD Secret erasure is confirmed by Security Control, the *AD_leave_confirm_message* is forwarded to the requesting CPCM Instance.
- The *AD_leave_finish_message* is forwarded to the Domain Controller.
- The Domain Controller responds with an *AD_update_indication_message* which the Local Master broadcasts to the locally connected CPCM Instances.

5.1.11.3 Domain Controller in AD Leaving

In order for the Domain Controller to be able to let an Instance Leave the AD, it shall perform the following tasks as depicted in figure 20.

If the Domain Controller is also a Local Master, it may receive an *AD_leave_begin_message* designating another Domain Controller. In that case, it shall act as a Local Master for the designated Domain Controller (see clause 5.1.11.2).

When receiving an AD Leave Request, a Domain Controller (whether Local Master or not) does the following:

- It first checks whether the requesting Instance is from the same AD. If not, AD Leaving is refused through a *transaction_rollback_message*.
- If the requesting Instance is ADSE countable, the Domain Controller first informs its Security Control of the AD Leave request.
- Then, or if the Instance was not ADSE countable, it asks the Domain Controller Device Application to confirm the AD Leave. If the Device Application does not confirm, then the Domain Controller informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- The Domain Controller requests its Security Control to establish a SAC with the requesting Instance if the message was not delegated or with the proxy Local Master if it was delegated. If the SAC cannot be established, a *transaction_rollback_message* is sent to the requesting Instance and the process ends.
- The Domain Controller sends an *AD_leave_ready_message* to the requesting CPCM Instance to inform it that the AD Leave may happen.
- The Domain Controller then waits for an *AD_leave_commit_message* response from the requesting Instance. If no *AD_leave_commit_message* is received, or if a Secured *transaction_rollback_message* is received, the Domain Controller reverts to its original state and the process ends. This means that the CPCM Instance shall have to re-start the protocol in order to be able to Leave.
- Having received the *AD_leave_commit_message*, the Domain Controller requests its Security control to proceed with the AD Leave. If the message is not delegated, the Security Control will do this directly. Else, it only updates the ADSE values. If the operation fails, the Domain Controller reverts to its original state.
- The Domain Controller sends an *AD_leave_confirm_message* to the requesting CPCM Instance which responds with an *AD_leave_finish_message*. If no message is received, the Domain Controller reverts to its original state.

- If the AD Leaving Instance was ADSE countable, the Domain Controller informs its security control and broadcasts an `AD_update_indication_message` to inform other connected CPCM Instances of the AD Leave.

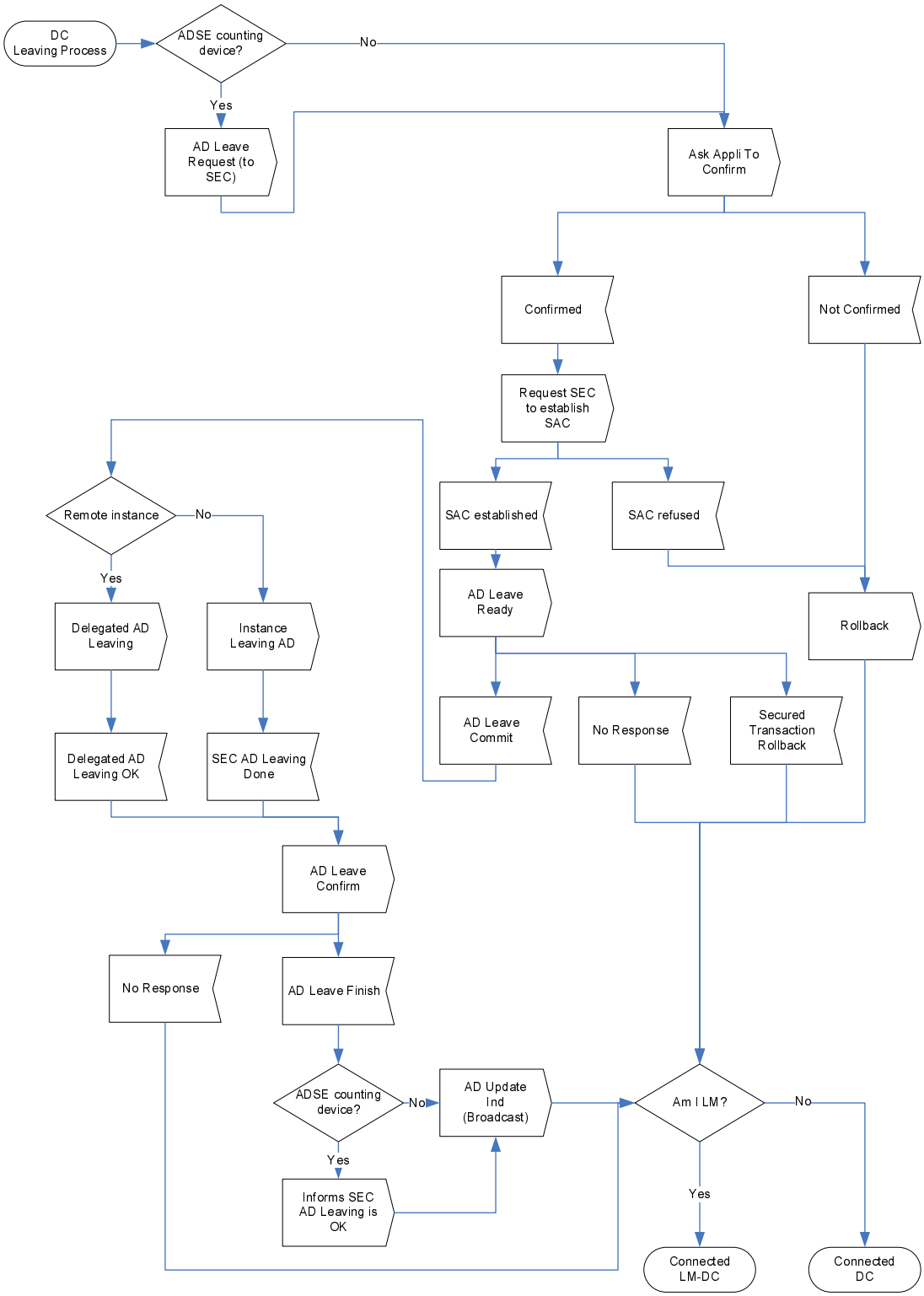

Figure 20: Domain Controller in AD Leaving Process

5.1.12 Domain Controller Transfer

The function and operation of the Domain Controller are described in clause 4.2.2.

5.1.12.1 AD Member becoming a Domain Controller

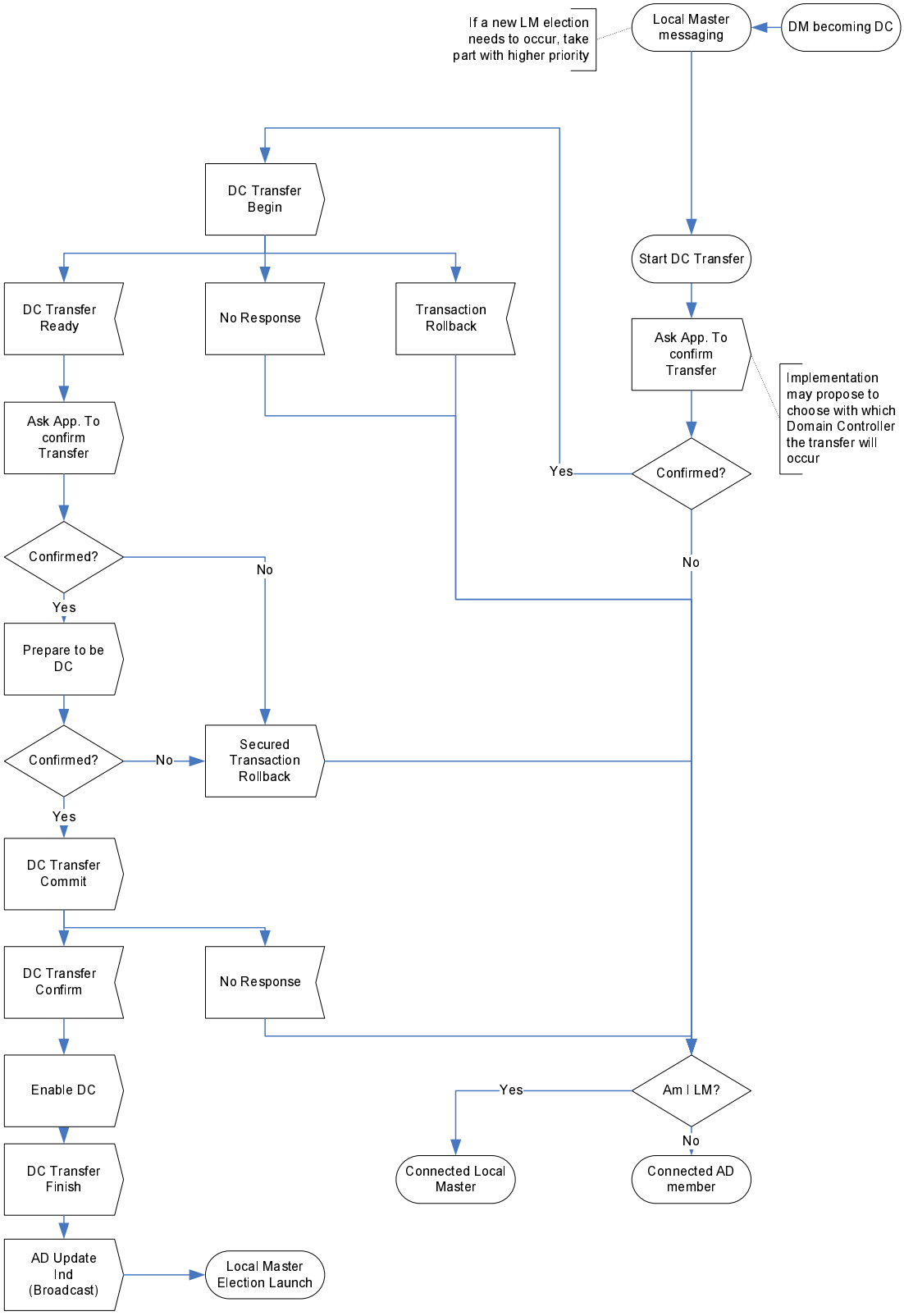

Figure 21: AD Member becoming a Domain Controller

A Domain Controller capable AD Member wishing to become a Domain Controller shall perform the following steps as depicted in figure 21:

- The requesting Instance shall first discover the Local Master. If there is no Local Master or if the Local Master is not a Domain Controller, it shall launch a new Local Master Election with its capability set to the higher level so that it becomes the Local Master.
- Next the requesting Instance Device Application is asked to confirm that it still wishes become a Domain Controller; an implementation may offer the Device Application the option to decide which Domain Controller function it wants to take.
- If the confirmation fails, then the Instance returns to its original state, i.e. if it was a Local Master, then it remains the Connected Local Master, if not, then it remains a Connected AD Member.
- Assuming the confirmation succeeds, the requesting Instance sends a *DC_transfer_begin_message* to the Domain Controller whose function is to be Transferred. As the Instance is Domain Controller capable, it can send the message directly to the Domain Controller and does not need to go through the Local Master.
- If no *DC_transfer_ready_message* response is received, or if a *transaction_rollback_message* occurs, the Instance returns to its original state as above.
- Next, the requesting Instance Device Application shall be asked to confirm that it still wishes to Transfer the Domain Controller function.
- If the requesting Instance Device Application confirms the DC Transfer, it checks with its Security Control whether it can proceed with the DC Transfer. If so, then a *DC_transfer_commit_message* shall be issued. If the requesting Instance Device Application or the Security Control refuses the DC Transfer, the requesting Instance shall send a Secured `transaction_rollback_message` and the Instance returns to its original state as above.
- If no *DC_transfer_confirm_message* response is received then the requesting Instance shall revert to its original state as above.
- Finally the requesting Instance shall inform its Security Control that the DC Transfer has been completed, issues a *DC_transfer_finish_message*, and broadcast an *AD_update_indication_message*. The requesting Instance then launches a new Local Master Election if it was not already the Local Master. It is now a Domain Controller.

5.1.12.2 The Domain Controller Transfer process

In order for the Domain Controller to be able to Transfer its Domain Controller function, it shall perform the following tasks as depicted in figure 22.

When receiving a Domain Controller Transfer request, a Domain Controller (whether Local Master or not) does the following.

- If the requesting Instance is not Domain Controller Capable or not from the same AD, the DC Transfer is refused and a *transaction_rollback_message* is sent.
- The Domain Controller checks with its Security Control whether the DC Transfer is possible or not. If not, it informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- Then, the Domain Controller asks the Domain Controller Device Application to confirm the DC Transfer. If the Device Application does not confirm, then the Domain Controller informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- The Domain Controller sends a *DC_transfer_ready_message* to the requesting CPCM Instance to inform it that the DC Transfer may happen.
- The Domain Controller waits for a *DC_transfer_commit_message* response from the requesting Instance. If no *DC_transfer_commit_message* is received, the process ends. If a Secured *transaction_rollback_message* is received, the Domain Controller informs its Security Control that the DC Transfer is cancelled and reverts to its original state.

- Having received the *DC_transfer_commit_message*, the Domain Controller ceases to be a Domain Controller and requests its Security Control to proceed with the DC Transfer.
- If the Security Control reports that the DC Transfer did not occur properly, the (original but no longer) Domain Controller ends the process.
- Else, it sends a *DC_transfer_confirm_message* to the requesting CPCM Instance which responds with a *DC_transfer_finish_message*.
- The (original but no longer) Domain Controller informs the Security Control that the DC Transfer is complete and assumes the Connected AD Member or Connected Local Master state.

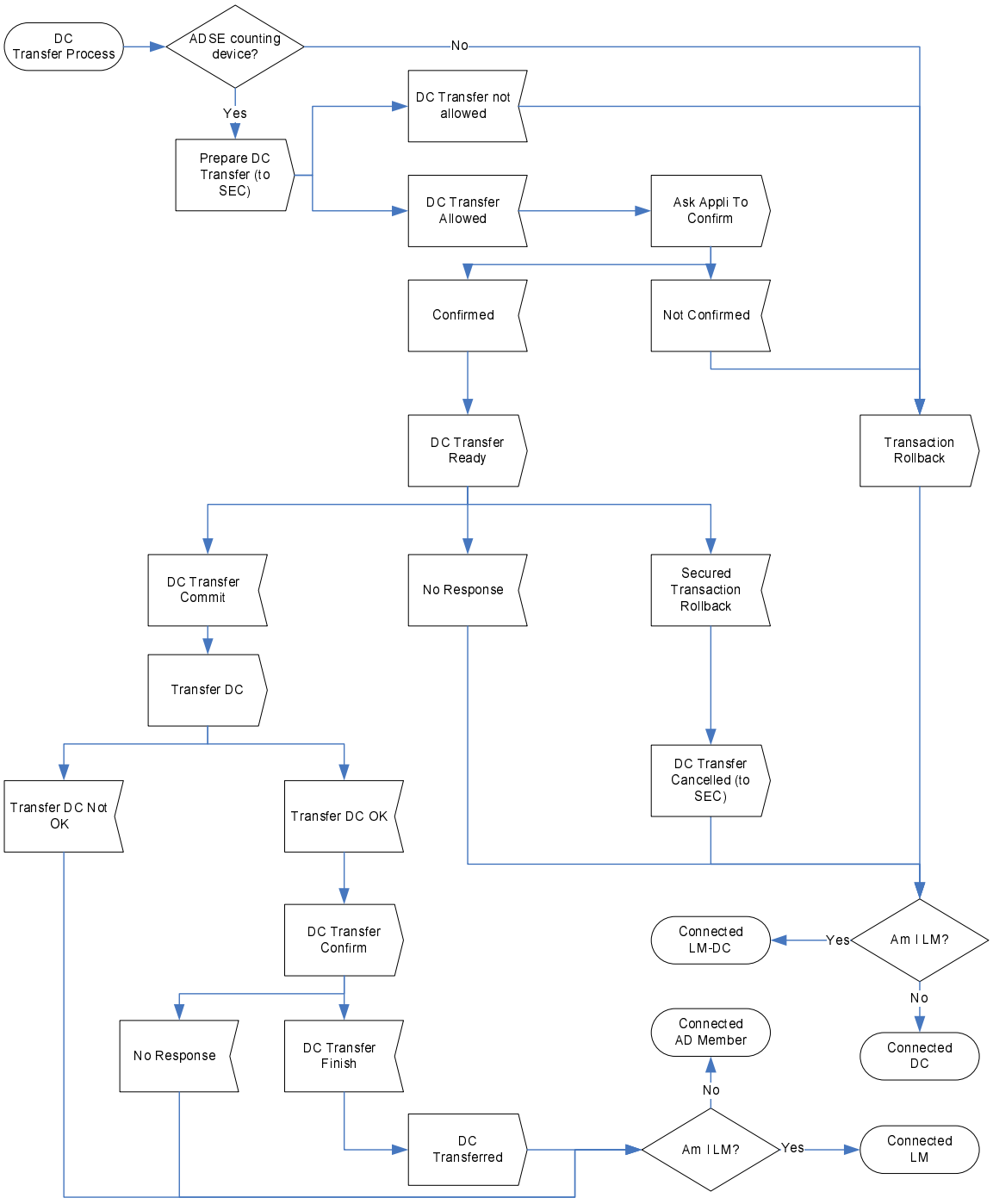

Figure 22: Domain Controller transfer

5.1.13 Domain Controller Splitting

The function and operation of the Domain Controller are described in clause 4.2.2.

5.1.13.1 AD Member becoming an additional Domain Controller

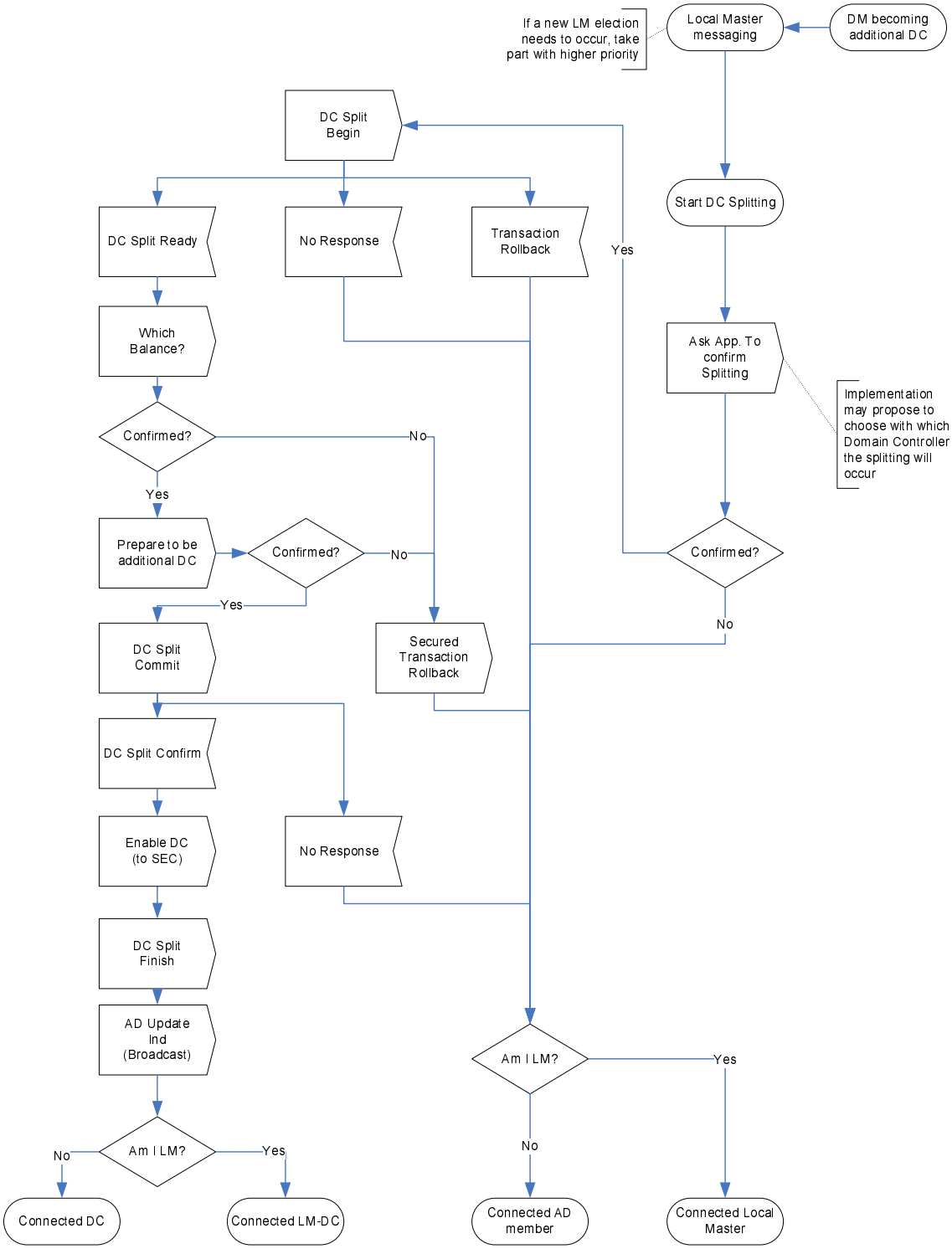

Figure 23: AD Member becoming additional Domain Controller

An AD Member (requesting Instance) wishing to become an additional Domain Controller shall perform the following steps as depicted in figure 23:

- The requesting Instance shall first discover the Local Master. If there is no Local Master or if the Local Master is not a Domain Controller, it shall launch a new Local Master Election with its capability set to the higher level so that it becomes the Local Master.
- Next the requesting Instance Device Application is asked to confirm that it still wishes become an additional Domain Controller. If there is already more than one Domain Controller in this AD, an implementation may offer the Device Application the option to decide which Domain Controller function it wants to Split. The Device Application may also choose which ADSE counts are requested and whether some CICFs in the list of CICFs are also requested.
- If the confirmation fails, then the requesting Instance returns to its original state, i.e. if it was a Local Master, then it remains the Connected Local Master, if not, then it remains a Connected AD Member.
- Assuming the confirmation succeeds, the requesting Instance sends a *DC_split_begin_message* to the Domain Controller whose function is to be Split. As the Instance is Domain Controller capable, it can send the message directly to the Domain Controller and does not need to go through the Local Master.
- If no *DC_split_ready_message* response is received, or if a *transaction_rollback_message* occurs, the requesting Instance reverts to its original state as above.
- Next, the requesting Instance Device Application shall be asked to confirm that it still wishes to Split the Domain Controller function.
- If the requesting Instance Device Application confirms the DC Transfer, then it asks its Security Control whether it can proceed with the DC Split. If so, then a *DC_transfer_commit_message* shall be issued. If the Device Application or the Security Control refuses the DC Split, the requesting Instance shall send a Secured *transaction_rollback_message* and revert to its original state as above.
- If no *DC_split_confirm_message* response is received, the requesting Instance shall revert to its original state as above.
- Finally the requesting Instance shall inform its Security Control that the DC Split has been completed, issue a *DC_split_finish_message* and broadcast an *AD_update_indication_message*. It is now a Domain Controller.
- It then assumes either the Connected Domain Controller or Connected Local Master and Domain Controller state.

5.1.13.2 A Domain Controller is Split

In order for the Domain Controller to be able to Split its Domain Controller function, it shall perform the following tasks as depicted in figure 24.

When receiving a Domain Controller Split request, a Domain Controller (whether Local Master or not) does the following:

- If the requesting Instance is not Domain Controller capable or not from the same AD, the DC Split is refused and a *transaction_rollback_message* is sent.
- The Domain Controller first checks with its Security Control whether the DC Split is possible or not. If not, it informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- Then, the Domain Controller asks its Device Application to confirm the DC Split. If the Device Application does not confirm, then the Domain Controller informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- The Domain Controller sends a *DC_split_ready_message* to the requesting CPCM Instance to inform it that the DC Split may happen.

- The Domain Controller then waits for a *DC_split_commit_message* response from the requesting Instance. If no *DC_split_commit_message* is received, then the process ends. If a Secured *transaction_rollback_message* is received, then the Domain Controller informs its Security Control that the DC Split is cancelled and it reverts to its original state.
- Having received the *DC_split_commit_message*, the Domain Controller requests its Security Control to proceed with the DC Split.
- If Security Control informs that the DC Split did not occur properly, the Domain Controller ends the process.
- Else, it sends a *DC_split_confirm_message* to the requesting CPCM Instance which responds with a *DC_split_finish_message*.
- The Domain Controller informs its Security Control that the DC Split is complete, broadcasts an *AD_update_indication_message* and assumes either the Connected Local Master and Domain Controller or Connected Domain Controller state.

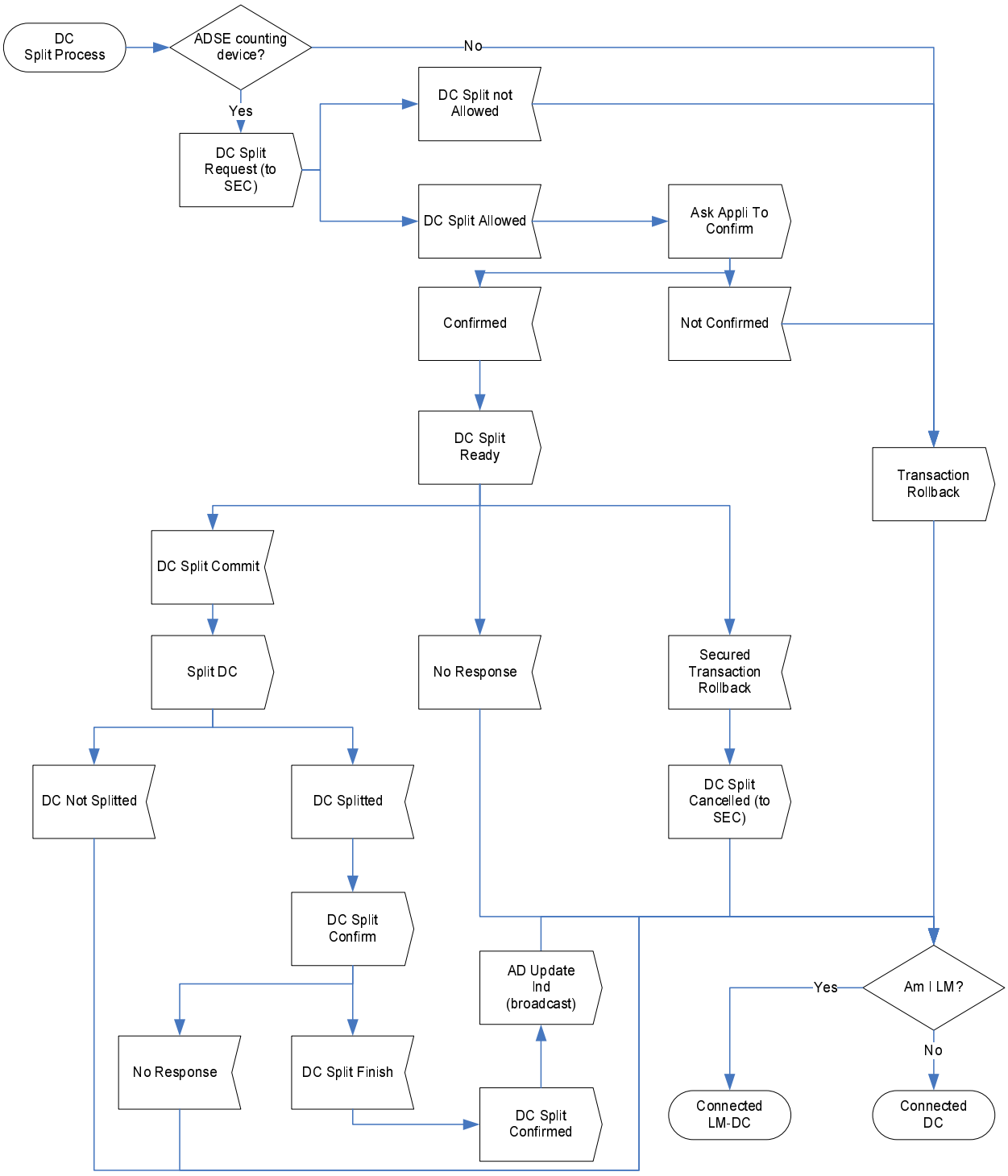

Figure 24: Domain Controller Split

5.1.14 Domain Controller Merging

The function and operation of the Domain Controller are described in clause 4.2.2.

5.1.14.1 The Merging Domain Controller

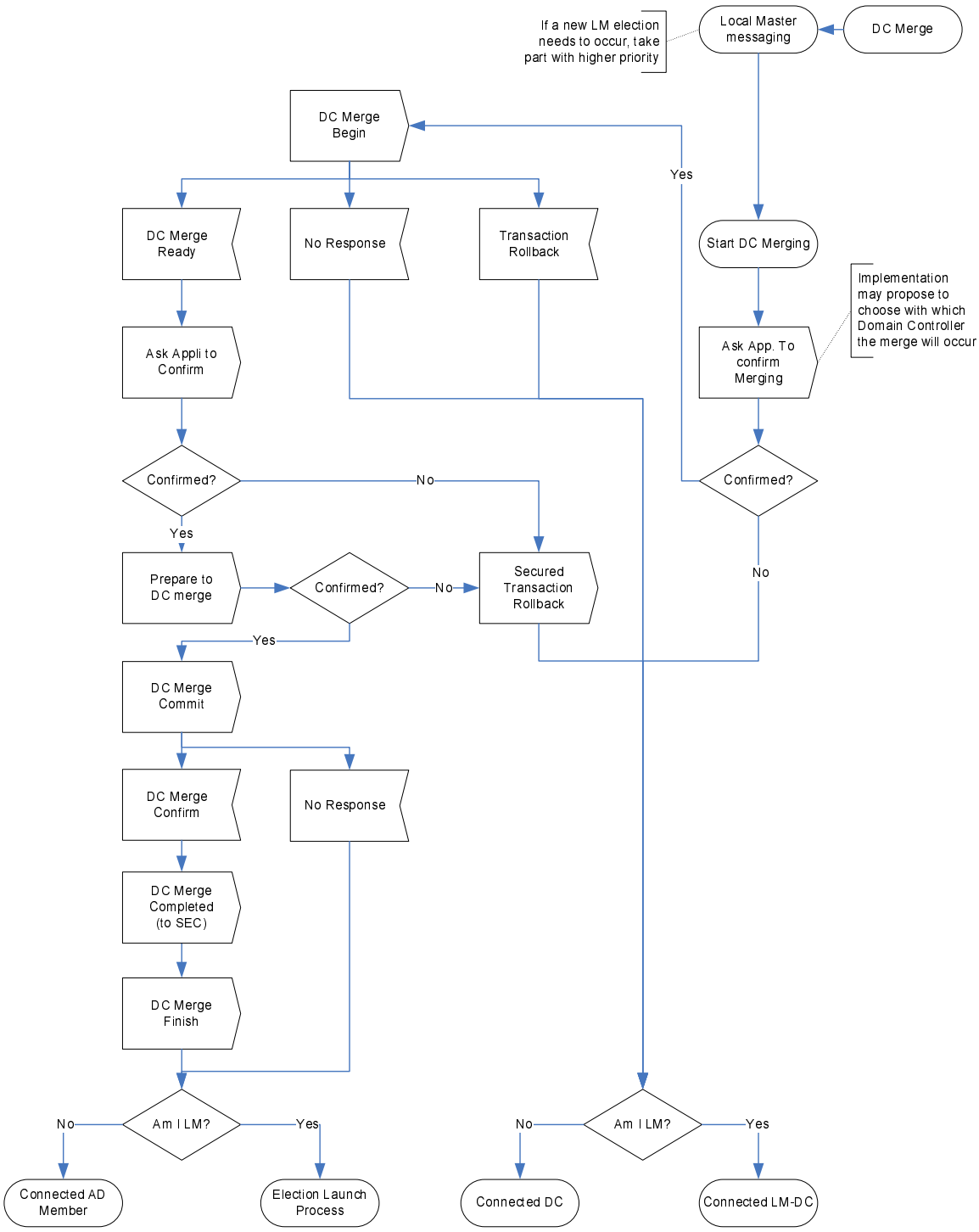

Figure 25: Merging Domain Controller

A Domain Controller wishing to Merge with another Domain Controller shall perform the following steps as depicted in figure 25:

- The requesting Instance shall first discover the Local Master. If there is no Local Master or if the Local Master is not a Domain Controller, the requesting Instance shall launch a new Local Master Election with its capability set to the higher level so that it becomes the Local Master.
- Next the requesting Instance Device Application is asked to confirm that it still wishes to Merge with another Domain Controller; if there are already two or more other Domain Controllers in this AD, an implementation may offer the Device Application the option to decide which Domain Controller function it wants to Merge with.
- If the confirmation fails, then the requesting Instance reverts to its original state, i.e. if it was a Local Master, then it remains the Connected Local Master and Domain Controller, if not, then it remains a Connected Domain Controller.
- Assuming the confirmation succeeds, the requesting Instance sends a *DC_merge_begin_message* message to the Domain Controller with which it is to be Merged (the "target"). As the requesting Instance is already a Domain Controller, it can send the message directly to the target Domain Controller and does not need to go through the Local Master.
- If no *DC_merge_ready_message* response is received, or if a *transaction_rollback_message* occurs, the requesting Instance reverts to its original state as above.
- Next, the requesting Instance Device Application shall be asked to confirm that it still wishes to Merge the Domain Controller function.
- If the requesting Instance's Device Application confirms the DC Merge, then it checks with its Security Control whether the DC Merge is possible or not. If it is possible, then a *DC_merge_commit_message* shall be issued. If the Device Application or the Security Control refuses the DC Merge, the requesting Domain Controller shall send a Secured *transaction_rollback_message* and it reverts to its original state as above.
- If no *DC_merge_confirm_message* response is received, the requesting Instance shall revert to its original state as above.
- Finally the requesting Instance informs its Security Control that the DC Merge has been completed and issues a *DC_merge_finish_message*.
- The requesting Instance assumes either the Connected AD Member or Connected Local Master state.

5.1.14.2 The Merged Domain Controller

In order for the Domain Controller to be able accept the Merging of another Domain Controller function, it shall perform the following tasks as depicted in figure 26.

When receiving a Domain Controller Merge request, a responding Domain Controller (whether Local Master or not) does the following:

- It checks that the request came from a Domain Controller of the same AD. Else, the request is refused through a *transaction_rollback_message*.
- The responding Instance asks its Device Application to confirm the DC Merge. If the Device Application does not confirm, then the responding Instance informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- The responding Instance requests its Security Control to establish a SAC with the requesting Instance. If the SAC cannot be established, a *transaction_rollback_message* is sent to the requesting Instance and the process ends.
- The responding Instance sends a *DC_merge_ready_message* to the requesting CPCM Instance to inform it that the DC Merge may happen.

- The responding Instance waits for a *DC_merge_commit_message* response from the requesting Instance. If no *DC_merge_commit_message* is received, then the process ends. If a Secured *transaction_rollback_message* is received, the responding Instance informs its Security Control that the DC Merge is cancelled and reverts to its original state.
- Having received the *DC_merge_commit_message*, the responding Instance requests its Security Control to proceed with the DC Merge.
- If Security Control informs that the DC Merge did not occur properly, the responding Instance ends the process.
- Else, the responding Instance sends a *DC_merge_confirm_message* to the requesting CPCM Instance which responds with a *DC_merge_finish_message*.
- The responding Instance informs Security Control that the DC Merge is complete, broadcasts an *AD_update_indication_message* and it assumes the Connected Local Master and Domain Controller or Connected Domain Controller state.

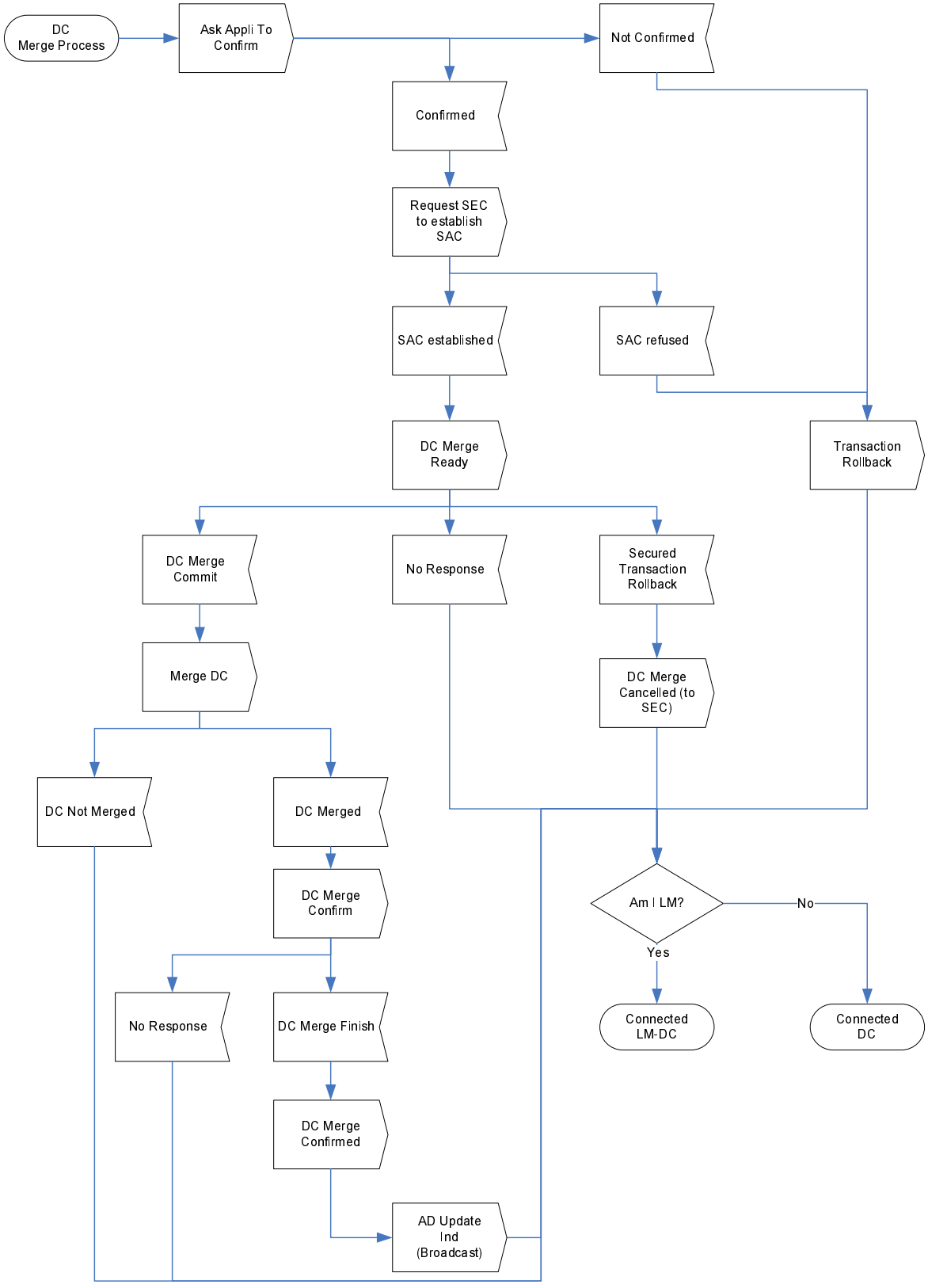

Figure 26: Merged Domain Controller

5.1.15 Domain Controller Rebalancing

The function and operation of the Domain Controller are described in clause 4.2.2.

5.1.15.1 The Rebalancing Domain Controller

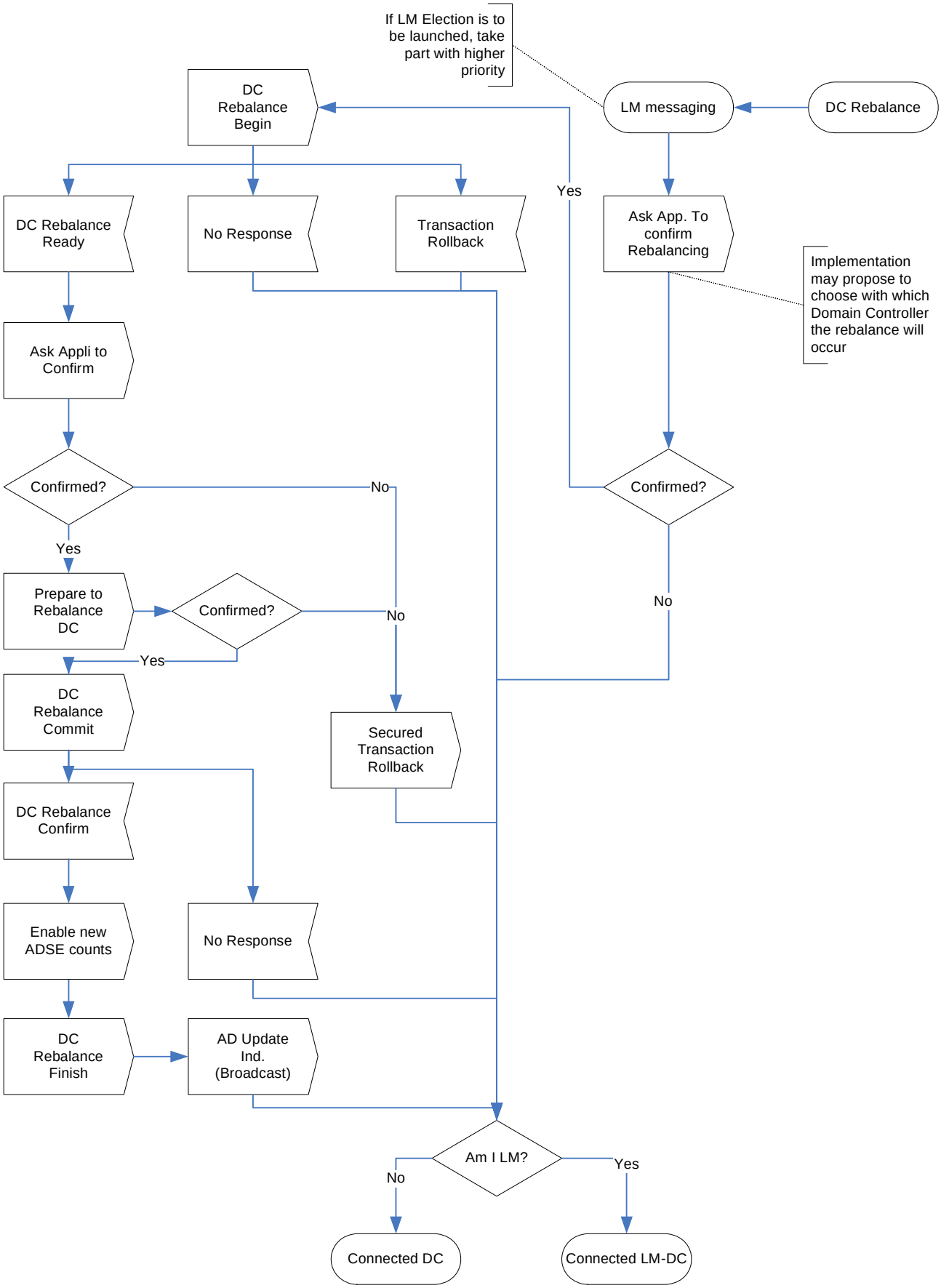

Figure 27: Rebalancing Domain Controller

A Domain Controller wishing to Rebalance its ADSE counts with another Domain Controller shall perform the following steps as depicted in figure 27:

- The requesting Instance shall first discover the Local Master. If there is no Local Master or if the Local Master is not a Domain Controller, it shall launch a new Local Master Election with its capability set to the higher level so that it becomes the Local Master.
- Next the requesting Instance Device Application is asked to confirm that it still wishes Rebalance its ADSE values with another Domain Controller; if there is more than one other Domain Controller in the AD an implementation may offer the Device Application the option to decide which Domain Controller function it wants to Rebalance with. It may also offer to the Device Application the ability to choose the new ADSE counts for the present Domain Controller, and whether some CICFs in the list of CICFs that Joined remotely are to be transferred.
- If the confirmation fails, then the requesting Instance reverts to its original state, i.e. if it was a Local Master, then it remains the Connected Local Master and Domain Controller, if not, then it remains a Connected Domain Controller.
- Assuming the confirmation succeeds, it sends a *DC_rebalance_begin_message* to the Domain Controller with which it is to be Rebalanced. As the Instance is already a Domain Controller, it can send the message directly to the Domain Controller and does not need to go through the Local Master.
- The requesting Instance now waits for a *DC_rebalance_ready_message* response from the responding Instance. This message will contain information on the ADSE values present on the responding Instance.
- If no *DC_rebalance_ready_message* response is received or if a *transaction_rollback_message* occurs, the requesting Instance reverts to its original state as above.
- Next, the requesting Instance Device Application shall be asked to confirm that the responding Instance ADSE values are acceptable, and identify exactly what ADSE values it wishes to use to Rebalance the Domain Controller function.
- If the requesting Instance Device Application confirms the DC Rebalance, it checks with its Security Control whether the Rebalancing is possible. If so, then a *DC_rebalance_commit_message* shall be issued. If the Device Application or the Security Control does not confirm, it shall send a Secured *transaction_rollback_message* and it reverts to its original state as above.
- If no *DC_rebalance_confirm_message* response is received, it shall revert to its original state as above.Finally the requesting Instance shall inform its Security Control the DC Rebalance has been achieved to enable its new ADSE counts, and issue a *DC_rebalance_finish_message*. It also broadcasts an *AD_update_indication_message*.
- The requesting Instance remains in either the Connected Domain Controller or Connected Local Master and Domain Controller state.

5.1.15.2 The Rebalanced Domain Controller

In order for the Domain Controller to be able to be Rebalanced with another Domain Controller function, it shall perform the following tasks as depicted in figure 28.

When receiving a Domain Controller Rebalance request, a responding Instance Domain Controller (whether Local Master or not) does the following:

- The responding Instance checks that the request came from a Domain Controller in the same AD. Else, the request is refused through a *transaction_rollback_message*.
- The responding Instance asks its Device Application to confirm the DC Rebalance. If the Device Application does not confirm, then the Domain Controller informs the requesting CPCM Instance through a *transaction_rollback_message* and the process ends.
- The responding Instance requests its Security Control to establish a SAC with the requesting Instance. If the SAC cannot be established, a *transaction_rollback_message* is sent to the requesting Instance and the process ends.

- The responding Instance sends a *DC_rebalance_ready_message* to the requesting CPCM Instance to inform it that the DC Rebalance may happen.
- It then waits for a *DC_rebalance_commit_message* response from the requesting Instance. If no *DC_rebalance_commit_message* is received, the process ends. If a Secured *transaction_rollback_message* is received, the responding Instance informs its Security Control that the DC Rebalance is cancelled and it reverts to its original state.
- Having received the *DC_rebalance_commit_message*, the responding Instance requests its Security Control to proceed with the DC Rebalance.
- If Security Control informs that the DC Rebalance did not occur properly, the responding Instance ends the process.
- Else, the responding Instance sends a *DC_rebalance_confirm_message* to the requesting CPCM Instance which responds with a *DC_rebalance_finish_message*.
- The responding Instance informs Security Control that the Rebalancing is complete; broadcasts an *AD_update_indication_message*. It remains in its original Connected Local Master and Domain Controller or Connected Domain Controller state.

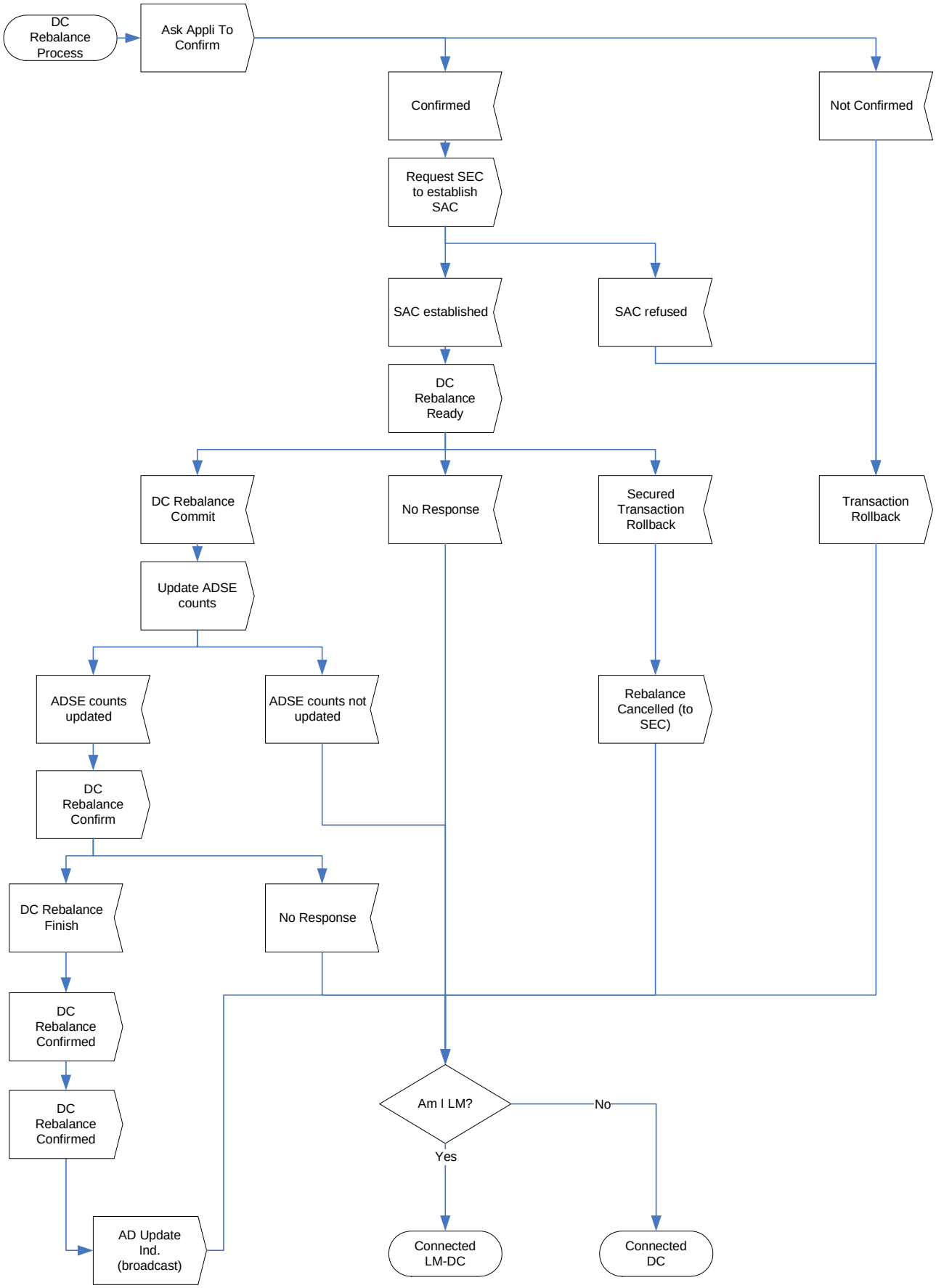

Figure 28: Rebalanced Domain Controller

5.2 Abnormal Behaviour

In the normal course of events when a requesting Instance sends a message to another Instance, assuming that there is no time-out, the responding Instance shall always respond with the corresponding message type to the request, for example, an *AD_discovery_response_message* will be sent in response to an *AD_discovery_request_message*.

The behaviours described in clause 5.1 all assume normal conditions. Many possible errors are not taken into account: e.g. message authentication checking may fail or the receiving Instance may be busy.

The failure of the transaction shall be signalled by the responding instance, which shall send an error message identifying the cause of the error (see TS 102 825-4 [2]).

Other abnormal behaviours occur when a message is received that is not expected at that stage, or when the ADM component of the CPCM Instance receives a *_confirm* message when its Security Control had not previously receive the relevant message. In these cases, a *CPCM_protocol_error_message* (if no SAC has been established with the sending CPCM Instance) or a Secured *CPCM_protocol_error_message* (if a SAC already exists with the sending CPCM Instance) is sent.

When receiving an error message, depending on the nature on the error, the CPCM Instance may decide to retry to send the previous message or to abort the protocol. Re-sending shall only occur a limited number of times. Aborting a protocol will result in the same effect as if no response was received.

When no response to the previous message is received, an implementation may decide to re-send the previous message before concluding that no response is likely to be received.

Upon receipt of a *transaction_rollback_message* or Secured *transaction_rollback_message*, a CPCM Instance shall stop running the current protocol and revert to the state it was before running the protocol returning any reserved resources to their original state. If it wishes to retry to run the protocol, it must re-start it from the beginning.

5.3 Protocol Failure Recovery

The AD Joining, AD Leaving, Domain Controller Transfer, DC Split, DC Merge and DC Rebalance protocols have all been designed to be able to restart and finish successfully if they were unexpectedly interrupted (e.g. because of a power failure or a network problem that makes one of the two peers unreachable). The protocols will either re-start or terminate in a consistent state when they are interrupted, regardless of when this interruption occurs. These six protocols affect the ADSE values and thus any interruption that could not re-start gracefully would likely result in ADSE counts with lower capabilities for the user.

NOTE: There is a risk that credits may be lost during and after the interruption if the Instances suffer loss of state information prior to re-connection, or are never re-connected.

The main enabler for this property is the fact that the two peers shall store the CPCM Instance Certificate Identifier of its peer once it enters the critical part of the protocol and shall erase it as soon as it leaves this critical part, namely:

- For the client, the storage of the CPCM Instance Certificate Identifier occurs just before sending the *_commit* message and stops just before sending the *_finish* message.
- For the server, the storage of the CPCM Instance Certificate Identifier occurs just after receiving the *_commit* message and stops just after receiving the *_finish* message.

Each instance of a protocol shall have its own allocated storage space for variable values within the Security Control. The following makes the assumption that Authorized Domain Management 'may' access this information. Exactly 'how' this is achieved is left to implementers.

Regarding the above, interruptions can occur when the two peers are in the following four states:

- None of the peers has stored a CPCM Instance Certificate Identifier.
- Only the client has stored a CPCM Instance Certificate Identifier.
- Both client and server have stored CPCM Instance Certificate Identifier.
- Only the server has stored a CPCM Instance Certificate Identifier.

CPCM Implementation Guidelines (TR 102 825-12 [i.5]) explain why only these four states need to be covered and will also provide more detailed explanation.

Clauses 5.3.1 and 5.3.2 describe the normative behaviour of a CPCM Instance with a stored CPCM Instance Certificate Identifier. The behaviour supposes that the user waits for the client and the server to be re-connected after the interruption.

At any time, the Device Application may force the client to restart the protocol with another server or tell the server that there is no longer any need to keep the client identifier recorded as the protocol will not restart. This feature is useful if a CPCM Instance can only record a few CPCM Instance Certificate Identifiers and it is therefore able erase one of these in order to be able to run another protocol. The user shall be made aware that this may cause undue loss of ADSE credits as shown in the examples below:

- The AD Secret was not received or not yet enabled, but the AD size had already been incremented during an AD Join.
- The AD Secret is disabled but the AD size had not yet been decremented during an AD Leave.
- Neither of the two peers is an active Domain Controller during a Domain Controller Transfer.
- The AD values had already been Split but the client is not yet the Domain Controller for a Domain Controller Split.
- The client is no longer a Domain Controller but the ADSE values had not yet been Merged for a Domain Controller Merge.
- The client Domain Controller function was disabled during a Domain Controller Rebalance.

In these cases, the CPCM Instance Certificate Identifier shall be erased and all values that may have been exchanged before the protocol abort (e.g. ADSE values or AD Secret) shall revert to their original state.

5.3.1 Client behaviour with a stored CIC Identifier

When a client with a stored CPCM Instance Certificate Identifier has sent a *_commit* message but not yet received a *_confirm* message, the following shall occur:

- At the next Discovery (e.g. when a CPCM Instance reconnects), it checks whether the CPCM Instance with its stored CPCM identifier is connected. If the protocol was a Domain Controller Transfer, Merge, Split or Rebalance, the Discovery Protocol shall be run so that Remote Domain Controllers are also discovered (see clause 5.1.1). It then receives answers from the currently connected Domain Controllers, from the Local Master and from other AD members.
- If the Domain Controller with which the protocol was initiated is present, it restarts the protocol with the DC using the relevant *_begin* message.
- If the original DC is not present, and if the running protocol was an AD Join or a Leave, the client shall send a *_begin* message containing the recorded Domain Controller identifier to the Local Master. The Local Master will forward this message to the Domain Controller if it is present or signals its absence through an error message.
- In the absence of the relevant Domain Controller, the CPCM Instance shall keep its current state and not run any new security protocol.

NOTE: Implementations may propose at the step the user to force the protocol to run.

- Upon receiving message *ADM_Invite* from the DC where protocol status is asserted, the CPCM Instance shall restart the protocol.

5.3.2 Server behaviour with a stored CPCM CIC Identifier

When a server with a stored CPCM Instance Certificate Identifier has received a *_commit* message but not yet a *_finish* message, the following shall occur:

- It accepts any restart of the protocol by the relevant CPCM Instance. If it had already changed the ADSE values in the course of the protocol, it does not change them again during the protocol.

It erases the CPCM Instance of the Identifier if any of the following events happen:

- The protocol was an AD Join and it receives a *Discovery_request_message* from the CPCM Instance as an AD Member. After erasing the Identifier, it reverts to its initial state (either Domain Controller or Local Master and Domain Controller).
- The protocol was an AD Leave and it receives a *Discovery_request_message* from the CPCM Instance as a Blank Instance. After erasing the Identifier, it reverts to its initial state (either Domain Controller or Local Master and Domain Controller).
- The protocol was a Domain Controller Transfer and it receives a *Discovery_request_message* from the CPCM Instance as a Domain Controller. After erasing the Identifier, it ceases to be a Domain Controller and becomes an AD Member or Local Master.
- The protocol was a Domain Controller Split and it receives a *Discovery_request_message* from the CPCM Instance as a Domain Controller. After erasing the Identifier, it reverts to its initial state (either Domain Controller or Local Master and Domain Controller).
- The protocol was a Domain Controller Merge and it receives a *Discovery_request_message* from the CPCM Instance as an AD Member or a Local Master. After erasing the Identifier, it enables the new ADSE values and goes back to its initial state (either Domain Controller or Local Master and Domain Controller).
- The protocol was a Domain Controller Rebalance and it receives a *Discovery_request_message* from the CPCM Instance as an AD Member or a Local Master. After erasing the Identifier, it enables the new ADSE values and reverts to its initial state (either Domain Controller or Local Master and Domain Controller).

5.4 Domain Internal Record

Each Domain Controller or Local Master is required to store information describing the current Authorized Domain (to which it belongs). This is optional for other AD Members. This information is called the Domain Internal Record.

This information might be used in case a failure in one of the Domain Controllers that would result in ADSE credit loss for the user.

The Domain Internal Record is generated and distributed by Domain Controllers. Local Masters forward these records to other AD Members, as needed.

Each Domain Controller maintains its own record. There may thus exist several Internal Records for one Authorized Domain. Local Masters and Domain Controllers shall store all the Domain Internal Record of their Authorized Domain.

5.4.1 Domain Internal Record information

Table 9 show the information gathered in Domain Internal Record.

Table 9: Domain Internal Record

Syntax	Number of bits	Meaning
ADID	72	A globally unique identifier for the AD assigned by the CPCM Instance that initially created the AD, and is never changed thereafter (see TS 102 825-4 [2]).
domain_controller_id	64	The unique Identifier of the Domain Controller that generated the record.
domain_controller_local_id	8	A local identifier of the Domain Controller used to cope with potential Domain Controllers Transfers, Merging, or Splitting.
internal_record_index	16	The index of the current record.
ADSE_values	160	ADSE values of the Domain Controller that generated the record.
AD_name	< 248	Human-readable name up to 31 characters long. Initial value is an implementation decision (see TR 102 825-12 [i.5]).
current_DC_local_ids		The list of the current Domain Controller local identifiers.
merged_DC_local_ids		The list of Domain Controller local identifiers that Merged into another Domain Controller.
change_date	40	Date and time of last change to this record. This information is optional. It is set only if the Domain Controller that generated the record has access to reliable time when generating the record.

5.4.2 Domain Internal Record management

The Domain Internal Record is managed by a Domain Controller in the following way:

- domain_controller_local_id is set to 0 for the Domain Controller that created the Authorized Domain.
- domain_controller_local_id is unchanged upon Domain Controller Transfer.
- At each Domain Controller Split, an unused domain_controller_local_id shall be given to the new Domain Controller. This identifier is chosen by the Split Domain Controller and shall not be already present in Current or Merged Domain Controller local ids lists.
- The domain_controller_local_id of a Domain Controller Merging into another one shall not be maintained.
- The record_index is set to 0 for the Domain Controller that created the AD and for each new Domain Controller created during a Split.
- The record_index is transmitted at each Domain Controller Transfer.
- The record_index is incremented by one each time the Domain Internal Record changes (i.e. upon change of ADSE values, AD renaming, Domain Controller Split, Transfer or Merge, update of Current or Merged Domain Controller local ids lists).
- The ADSE values are updated when a CICF Joins or Leaves an AD and upon Domain Controller Merging, Splitting and Rebalancing and upon Remote Domain Controller Transfer.
- The AD_name may only be changed upon an *AD_change_request_message* or by direct user interaction on the Domain Controller.
- The current_DC_local_ids list is initialized during AD creation with the local_id (i.e. 0). The Merged Domain Controller local ids list is empty when an AD is initially created.
- After each Domain Controller Split, the newly created local id is added to the current Domain Controller list. After each Domain Controller Merge, the Merging Domain Controller local id is added to the Merged Domain Controller list.

- Each time a Domain Controller receives an updated internal record from another Domain Controller, the resulting Current and Merged Domain Controller local ids list is the union of its own lists and the received lists.
- When a local identifier is present in the Merged Domain Controller local ids list, it shall be removed from the Current Domain Controller local ids list.
- When the Current Domain Controller local ids list is reduced to one element, the Merged Domain Controller local id lists shall be emptied.

The Domain Internal records are carried in *AD_update_request_message*, *AD_update_response_message* and *AD_update_indication_message*. Upon receipt of such messages, AD Members, Local Masters and Domain Controllers may possibly update their internal records. To that end, for each domain_controller_local_id, it compares the received record index with its stored one. If there is no such stored record, it shall store the newly received one. Else, if the received record is more recent, it replaces its stored record with the newly received one. AD internal records generated by DC in the merged DC Domain Controller local ids list shall be erased. Furthermore, if the Instance is a Domain Controller, it shall check whether its local ids list needs to be updated (see above) and, if so, updates them and issues a new Domain Internal Record.

A CPCM Instance storing Domain Internal Records shall erase all Domain Internal Records corresponding to local identifiers of the Merged Domain Controllers local ids list.

NOTE: A Domain Controller will never replace the internal record it had generated upon receipt of an *AD_update_request_message, AD_update_response_message* or *AD_update_indication_message*.

6 Security Control Behaviour for Authorized Domain Management

The following clauses focus only on the conformant behaviour of the Security Control component. The ADM component behaviour is described in clause 5. The need for the Security Control to intervene is decided by an exchange of messages between ADM and Security Control component. These Security Control messages are informative only and are to be defined by the implementation.

Clause 6 defines conformant behaviour of Security Control implementations when used for ADM purposes. This conformant behaviour achieves ADSE method as described in clause 4.4.2.

Clause 7 describes a list of generic ADSE tools upon which the ADSE method has been built. Implementation of these generic ADSE tools beyond the mandatory ADSE method is optional.

6.1 AD Creation

AD creation is initiated by a Device Application request when an AD aware Blank Instance connects for the first time where no AD currently exists. The ADM component requests its Security Control to create the new Authorized Domain by a *create_new_AD_message*.

Upon receiving such a request, Security Control first checks whether its history_count is lower than its history_ceiling and if it is Domain Controller capable. If not, the Security Control signals that AD creation is not possible to its ADM component using an *AD_created_message*.

Else, Security Control does the following:

- It sets the new AD Secret to a random value and records it.
- It computes the ADID as follows:
  - The 64 first bit are the identifier of the CPCM Instance certificate.

- The 8 last bits are an index that is incremented by one after the Instance has created the AD AND either Content has been bound to that AD, or another Instance has Joined the AD. The same index value can be reused if the AD that was created with it only had its creating Instance as a member and had no Content bound to it. When the index reaches above 255, it cycles back to 0.

- It initializes `total_ceiling`, `remote_ceiling`, `local_ceiling`, `DC_split_ceiling` and `DC_remote_ceiling` to default values defined by the C&R regime.

NOTE: When the CPCM Instance conforms to several C&R regimes, the ceiling initialization values will be determined by an agreement between those C&R regimes.

- It sets `remote_count, DC_split_count` and `DC_remote_count` to 0 and `total_count` and `local_count` to 1 (Since the Instance is Domain Controller capable, it is also ADSE countable).

- It increments its `history_count` by one.

- It becomes the Domain Controller and Local Master.

It sends back ADID and ADSE values to its ADM component to confirm the creation of the AD in an *`AD_created_message`*.

6.2 Domain Controller Leaving

Upon request of its ADM component to Leave the AD, the Security Control of a Domain Controller does the following:

- It erases its AD Secret and ADID.

- It sets all the ADSE values to 0.

- It becomes a Blank Instance.

It confirms the completion of the AD Leave to the ADM component. If it was not a Domain Controller, a negative result is sent.

6.3 AD Joining

6.3.1 Blank Instance Joining an AD

The Security Control is involved at five steps during the AD Join.

The next five steps below shall be executed in the described order. Any inversion or omission shall result in a negative result and any failure of one step will cause the process to stop.

1) First, when connecting, the Blank Instance checks with its Security Control component whether it may Join an Authorized Domain or not through an *`authorize_AD_join_message`*. If the Instance is AD aware and its `history_count` is lower than its `history_ceiling`, then the result is positive, otherwise it is negative. The result is sent back to the ADM component in an *`authorize_AD_join_message`*.

2) It will accept the challenge to run the SAC establishment procedure with any requesting Instance. It will respond to any Proximity Test.

3) Just before sending the *`AD_join_commit_message`* , the ADM component informs its Security Control that the AD Join is about to occur through a *`prepare_AD_join_message`*. The Security control first verifies the AD Join is being made with a Domain Controller. If the AD Join is delegated, it verifies the SAC has been established with the Local Master. Else, it verifies the SAC has been established with the Domain Controller and that its certificate corresponds to a CICF. If not, it refuses the AD Join. Security Control then records the CPCM Instance Certificate Identifier of the Domain Controller if it is an ADSE countable Instance.

4) It will receive the AD Secret through the SAC in the *`deliver_AD_secret`*. It shall record the AD Secret and confirm its receipt using a *`deliver_AD_secret_response`* message. The AD Secret is not usable at that step.

5) Once the ADM component has received an *AD_join_confirm_message*, it requests its Security Component to complete the AD Join procedure through an *enable_AD_secret_message*. The Security Control increments its history_count by one, enables the AD Secret, erases the recorded Domain Controller identifier (if it was recorded) and becomes an AD Member.

If the protocol is intentionally interrupted by the Device Application before the fifth step occurred (e.g. because the user decided to run the protocol with another Domain Controller), all the recorded data (Domain Controller identifier and/or AD Secret) shall be erased.

6.3.2 Local Master in AD Joining Process

The Local Master (that is not Domain Controller) intervenes at two stages during the AD Join process. These two steps shall be conducted in order, if there is a failure at any stage it shall be reported as a protocol error and the process shall end:

1) Upon receipt of an *AD_join_begin_message*, it establishes a SAC with the requesting Instance. During the procedure, it checks whether the AD_capability field transmitted in the *AD_join_begin_message* matches the certificate one. If anything fails, it is signalled to the ADM component. It verifies then the requesting Instance is ADM capable and a Blank Instance. If it is a non ADSE countable Instance, it verifies whether the requesting Instance is Local. If SAC establishment or either of the two tests fails, it sends back the corresponding result in a *transaction_rollback_message* and the protocol stops.

2) The ADM component requests its Security Component to deliver the secret. This is done upon receipt of an *AD_join_commit_message* for a non-ADSE countable Instance or upon receipt of an *AD_join_confirm_message* from the Domain Controller for an ADSE countable Instance. In the latter case, Security Control shall ensure that the authorization actually comes from a Domain Controller that is as well a CICF. Security Control sends the AD Secret to the Blank Instance via a *deliver_AD_secret* message and waits for confirmation. Once a *deliver_AD_secret_response* message is received, it informs its ADM component that the delivery was successful. If this message was not received, then it informs its ADM component of the delivery failure.

6.3.3 Domain Controller in AD Joining protocol

For a non ADSE countable Instance, the Domain Controller Security Control behaves exactly as a Local Master for the same case.

For an ADSE countable Instance, the Domain Controller Security Control is active in three stages. These steps shall be conducted in order, if there is a failure at any stage it shall be reported as a protocol error and the process shall end.

1) The ADM component requests its Security Control to check whether the AD Join is possible. To that end, the Security Control performs the following:

   - First, it checks whether the requesting Instance is ADM capable and a Blank Instance. If not, the AD Join is refused.

   - If total_count is not lower than total_ceiling, the AD Join is refused.

   - Then it checks whether the requesting Instance is Local or Remote. If the message was delegated and if the requesting Instance is not directly reachable (i.e. it cannot exchange messages with that Instance without the Local Master), the Instance is considered to be Remote. If the Instance is reachable or the message was not delegated, it establishes a SAC with the Instance and a standard Proximity Test is performed. During the SAC establishment, it checks whether the AD_capability field transmitted in the *AD_join_begin_message* matches the certificate one. If anything fails, the AD Join is refused.

   - If the Instance is Remote and remote_count is lower than remote_ceiling, the AD Join is allowed. Else it is refused.

   - If the Instance is Local and local_count is lower than local_ceiling, then the AD Join is allowed.

   - Else, the Security Control broadcasts an *ADM_quorum_test_query_message* carrying the random value it has just picked. It then waits for all possible responses. It checks the responses are correct and came from ADSE countable Instances only.

- If the number of correct responses is not lower than the "quorum percentage" of `local_count`, a Proximity Test is run with each of the correct responders. If the number of Local correct responders is not lower than the "quorum percentage" of `local_count`, the AD Join is accepted.

- If the number of Local correct responses is lower than the "quorum percentage" of `local_count` but `remote_count` is lower than `remote_ceiling`, the AD Join is accepted and the CPCM Instance shall be counted as a Remote CPCM Instance during the Joining. Else, the AD Join is refused.

If the AD Join is refused, the Security Control may try to contact ADMAAA to get permission.

If the AD Join is accepted (originally or from ADMAAA), the Security Control may reserve the corresponding resources: It records the identifier of the requesting Instance, and whether the AD Join will be Local or Remote. If other CPCM Instances request to AD Join, corresponding counters (i.e. `total_count` and `local_count` or `remote_count`) will be considered to be incremented by one and, if the Join is Remote, the CIC identifier of the Joining Instance will be added to the list of CICFs that Joined remotely. This reservation resource will stop once the AD Join actually occurs, on receipt of a *`transaction_rollback_message`* or a Secured *`transaction_rollback_message`*, or if no response is received. The Security Control then notifies the ADM component of the result.

If the message was delegated but the requesting Instance is directly reachable, Security Control may decide to continue in delegation mode or to proceed with the AD Join without the involvement of the Local Master. In the former case, it shall establish a SAC with the Local Master. If the establishment fails, AD Join is refused.

2) Upon receipt of the *`AD_join_commit_message`*, the ADM component will inform its Security Control to proceed with the AD Join if the message is not delegated or to proceed with a delegated AD Join. In both cases, the Security Control first increments `total_count` and either `local_count` or `remote_count` depending on whether the AD Join was determined to be Local or Remote. If it is Remote, the corresponding CIC identifier is added to the list of CICF that Joined Remotely. Then, if the AD Join is not delegated, the Security Control sends the AD Secret to the Blank Instance through a *`deliver_AD_secret`* message and waits for confirmation. Once the *`deliver_AD_secret_response`* message is received, it informs its ADM component that the delivery was successful. If this message was not received, it informs its ADM component of the delivery failure.

   If the AD Secret was correctly delivered, or if message was delegated, Security Control informs its ADM component it can confirm the AD Joining.

   If the protocol is intentionally interrupted by the Device Application before the following step occurs, the Security Control shall erase the recorded CPCM Instance Certificate Identifier, if any.

3) Finally, upon receipt of an *`AD_join_finish_message`*, or upon other events as described in clause 5.3, the ADM component informs its Security Control that the AD Join was achieved. The Security Control then erases the recorded CPCM Instance Certificate Identifier.

Each time the DC discovers a CICF which is in its list of Instances that Joined remotely, its Security Control shall do the following:

- It runs a Proximity Test with the Instance. If it is still Remote, nothing is to be done.

  If SRTT or SRTTL is used to check Local-ness, then the AD key shall be used to authenticate the Proximity Test.

- If it is Local and `local_count` is lower than `local_ceiling`, it decrements `remote_count` by one and increments `local_count` by one and removes the corresponding entry from the list.

- Else, it performs a Quorum Test: it broadcasts an *`ADM_quorum_test_query_message`* carrying a random value it has just picked. It then waits for all possible responses. It checks the responses are correct and came from Local ADSE countable Instances only. If the number of correct answers is lower than the "quorum percentage" of `local_count`, nothing is to be done.

  If SRTT or SRTTL is used to check Localness, then the AD key shall be used to authenticate the Proximity Test.

- If the number is not lower than the "quorum percentage" of `local_count`, a Proximity Test is run with each of the correct responders. If the number of Local correct responders is lower than the "quorum percentage" of `local_count`, nothing is to be done. Else it decrements `remote_count` by one and increments `local_count` by one and removes the corresponding entry from the list.

NOTE: Quorum Tests may be skipped if using ADMAAA.

6.3.4 Other AD Members

At any moment, the Security Components of other AD Members that are ADSE countable shall be ready to perform a Quorum Test with the Domain Controller. To that end, it shall respond to any *ADM_quorum_test_query_message* carrying the relevant ADID with message *ADM_quorum_test_response_message*. It shall also be ready to perform Proximity Tests with the Domain Controller.

Non ADSE countable Instances shall not respond to an *ADM_quorum_test_query_message*.

6.4 AD Leave

6.4.1 AD Member Leaving

The Security Control is involved at four steps during the AD Leave executed in the following order. Any inversion or omission shall result in a negative result and any failure of one step will cause the process to stop:

1) After the *AD_leave_begin_message* is sent, it will accept the challenge to run the SAC establishment procedure with any requesting Instance. It will respond to any Proximity Test.

2) Just before sending the *AD_leave_commit*_message, the ADM component informs its Security Control that the AD Leave is about to occur through a *disable_AD_secret* message. The Security Control first checks the *AD_leave_ready_message* was received from a Domain Controller. If the AD Leave is delegated, it verifies the SAC has been established with the Local Master. Else, it verifies the SAC has been established with the Domain Controller and that its certificate corresponds to a CICF. If not, the process ends. The Security Control then disables the AD Secret. This means that this secret cannot be used in any way. It then records the CPCM Instance Certificate Identifier of the Domain Controller if it is an ADSE countable Instance.

3) It erases the AD Secret and ADID upon receipt of the *erase_AD_secret* and confirms the deletion through an *erase_AD_secret_response* message. It does not yet become a Blank Instance at this step.

4) Once the ADM component has received an *AD_leave_confirm_message*, it requests its Security Component to complete the AD Leave procedure through an *enable_AD_secret_message*. The Security Control erases the recorded Domain Controller identifier (if it was recorded) and then it becomes a Blank Instance.

If the protocol is intentionally interrupted by the Device Application between the second and the fourth step (e.g. because the user knows it will not reconnect with the Domain Controller), the Security Control will erase the recorded identifier AND the AD Secret and it becomes a Blank Instance.

6.4.2 Local Master in AD Leaving Process

The Local Master (that is not a Domain Controller) intervenes at three stages during the AD Leaving process. These three steps shall be conducted in order, if there is a failure at any stage it shall be reported as a protocol error and the process shall end:

1) Upon receipt of an *AD_leave_begin*_message, it verifies that the requesting Instance is from the right AD. If not, it sends back the corresponding result in a *transaction_rollback_message* and the protocol stops.

2) The ADM component requests its Security Component to establish a SAC with the requesting Instance. This happens upon receipt of an *AD_leave_begin_message* for a non ADSE countable Instance or upon receipt of an *AD_leave_ready_message* from the Domain Controller if the Instance is ADSE countable. In the latter case, Security Control shall ensure that the authorization actually comes from a Domain Controller that is as well a CICF. The Security Control then runs the SAC establishment procedure with the Security Control of the requesting AD Leaving Instance. If the procedure fails, it is signalled to the ADM component.

3) The ADM component requests its Security Component to erase the AD Secret. This is done upon receipt of an *AD_leave_commit*_message for a non-ADSE countable Instance or upon receipt of an *AD_leave_confirm_message* from the Domain Controller for an ADSE countable Instance. The Security control requests the Instance to erase the AD Secret via an *erase_AD_secret* message and waits for confirmation. Once an *erase_AD_secret_response* message is received, it informs its ADM component that the delivery was successful. If this message was not received, then it informs its ADM component of the erasure failure.

6.4.3 Domain Controller in AD Leaving protocol

For a non ADSE countable Instance, the Domain Controller Security Control behaves exactly as a Local Master for the same case.

For an ADSE countable Instance, the Domain Controller Security Control is active in four stages. These steps shall be conducted in order, if there is a failure at any stage it shall be reported as a protocol error and the process shall end.

1) The ADM component requests its Security Control to prepare for the AD Leave. To that end, the Security Control checks whether the requesting Instance is from the same AD. If not, the AD Leave is refused.

2) The ADM component will then ask to establish a SAC with the requesting Instance or with the Local Master acting as a proxy if the message was delegated. This happens if the Device Application authorized the AD Leave. The Security Control then runs the SAC establishment procedure with the Security Control of the requesting Instance. During the SAC establishment, it verifies that all relevant information transmitted within the request match the certificate information. If anything fails, it is signalled to the ADM component.

3) Upon receipt of the *AD_leave_commit*_message, the ADM component will inform its Security Control to proceed with the AD Leave if the message is not delegated, or to proceed with a delegated AD Leave otherwise. In both cases, the Security Control does the following:

   - It records the requesting CPCM Instance Certificate Identifier.
   - It first decrements `total_count` by one.
   - Then it checks whether the requesting Instance is present in its list of CICFs that Joined remotely. If so, it decrements `remote_count` by one and removes the entry from the list.
   - Else, it decrements `local_count` by one. If the counter is already zero, it decrements the other one.

   These new ADSE values are not enabled. Then, if the AD Leave is not delegated, the Security Control requests the AD Leaving Instance to erase its Secret through an *erase_AD_secret* message and waits for confirmation. Once the *erase_AD_secret_response* message is received, it informs its ADM component that the erasure was successful. If this message was not received, it informs its ADM component of the erasure failure.

   If the message was delegated, Security Control informs its ADM component it can confirm the AD Leave.

   If the Device Application interrupts the protocol at this step, then the Security Control will erase the recorded CPCM Instance Certificate Identifier AND revert to its original ADSE values.

4) Finally, upon receipt of an *AD_leave_finish_message*, or upon other events as described in clause 5.3, the ADM component informs its Security Control that the AD Leave was achieved. The Security Control then enables the ADSE values, and erases the recorded CPCM Instance Certificate Identifier, if any.

6.5 Domain Controller Transfer

6.5.1 AD Member becoming a Domain Controller

The Security Control is involved at four steps during the Domain Controller Transfer which shall be executed in the following order. Any inversion or omission shall result in a negative result and any failure of one step will cause the process to stop:

1) After the *DC_transfer_begin_message* is sent, it will accept the challenge to run the SAC establishment procedure with any requesting Instance. It will also accept any Proximity Test.

2) Just before sending the *DC_transfer_commit_message*, the ADM component informs its Security Control that the DC Transfer is about to occur through a *prepare_to_be_DC_message*. Security Control checks the SAC has been established with a Domain Controller and that its certificate corresponds to a CICF, otherwise the process fails. The Security Control records the CPCM Instance Certificate Identifier of the Domain Controller.

3) It stores received ADSE values and the list of CICFs that Joined remotely upon receipt of a *become_DC_message* and confirms the reception in a *new_DC_message*.

4) Once the ADM component has received a *DC_transfer_confirm_message*, it requests its Security Component to complete the DC Transfer procedure through an *enable_DC_message*. The Security Control erases the recorded Domain Controller identifier and becomes a Domain Controller.

If the protocol is intentionally interrupted by the Device Application before the fourth step occurred (e.g. because the user decided to run the protocol with another Domain Controller), all the recorded data (ADSE values and CPCM Instance Certificate Identifier) shall be erased.

6.5.2 Domain Controller Transfer Process

The Domain Controller Security Control is active in three stages. These steps shall be conducted in the order described below, if there is a failure at any stage it shall be reported as a protocol error and the process shall end.

1) The ADM component requests its Security Control to prepare for the DC Transfer. To that end, the Security Control does the following:

   - First, it checks whether the requesting Instance is from the same AD and DC capable. If not, the DC Transfer is refused.

   - Then it checks whether the requesting Instance is Local or Remote via a standard Proximity Test.

   If SRTT or SRTTL is used to check Localness, the AD key shall be used to authenticate the Proximity Test.

   - If the requesting Instance is Local, the DC Transfer is authorized.

   - If it is Remote, the DC Transfer is authorized if `DC_remote_count` is lower than `DC_remote_ceiling`. Else it is refused.

   - The Security Control informs the ADM component of the result.

   The Security Control then runs the SAC establishment procedure with the Security Control of the requesting Instance. During the SAC establishment, it verifies that all relevant information transmitted within the request match the Certificate information. If anything fails, it is signalled to the ADM component.

2) Upon reception of a *DC_transfer_commit_message*, the ADM component will inform its Security Control to proceed with the DC Transfer. If the DC Transfer is Remote, the Security Control first increments `DC_remote_count` by one. Then it disables all ADSE counts. This implies that it can no longer run any other security (i.e. AD Joining, AD Leaving, DC Transfer to another Instance, DC Merging, Splitting or Rebalancing). It ceases to be a Domain Controller from now on.

   Then, the Security Control records the request CPCM Instance Certificate Identifier and transfers its current ADSE values and its list of CICFs that Joined remotely in a *become_DC_message* and waits for confirmation. Once the *new_DC_message* is received, it informs its ADM component that the DC Transfer was successful. If this message was not received, it informs its ADM component of the DC Transfer failure.

If the Device Application intentionally interrupts the protocol at that step (e.g. because it knows it will never finish), the CPCM Instance Certificate Identifier shall be erased and the Instance will cease to be a Domain Controller.

3) Finally, upon receipt of a *DC_transfer_finish_message*, or upon other events as described in clause 5.3, the ADM component informs its Security Control that the DC Transfer was achieved. Security Control then erases the recorded CPCM Instance Certificate Identifier.

6.6 Domain Controller Split

6.6.1 AD Member becoming an additional Domain Controller

The Security Control is involved at four steps during the Domain Controller Split; these steps shall be executed in the described order. Any inversion or omission shall result in a negative result and any failure of one step will cause the process to stop.

1) After the *DC_split_begin_message* is sent, it will accept the challenge to run the SAC establishment procedure with any requesting Instance. It will also accept any Proximity Test.

2) Just before sending the *DC_split_commit_message*, the ADM component informs its Security Control that the DC Split is about to occur through a *prepare_to_be_DC_message*. Security Control first checks the following controls:

   - The SAC has been established with a Domain Controller whose certificate corresponds to a CICF Instance.
   - The requested ADSE counts are not higher than the one of the Domain Controller that will Split. The requested `DC_split_count` shall be lower than `DC_split_ceiling`.
   - The requested `total_count, remote_count, DC_split_count` and `DC_remote_count` are not higher than their respective ceilings. Security Control also checks that the resulting counts for the Split Domain Controller will obey the same rules (resulting counts being the difference between current counts and requested counts).
   - The requested `total_count` shall be the sum of `local_count` and `remote_count`.

   If the controls are correct, it records the CPCM Instance Certificate Identifier of the Domain Controller. Else, it informs its ADM component of the bad result.

3) Upon receipt of the *become_DC_message*, it checks that the received ADSE values match the expected ones. If so, it records them. It sends back the result in a *new_DC_message*.

4) Once the ADM component has received a *DC_split_confirm_message*, it requests its Security Component to complete the DC Split procedure through an *enable_DC_message*. The Security Control erases the recorded Domain Controller identifier and becomes a Domain Controller.

If the protocol is intentionally interrupted by the Device Application before the fourth step occurred (e.g. because the user decided to run the protocol with another Domain Controller), all the recorded data (Domain Controller identifier and/or ADSE values) shall be erased.

6.6.2 Domain Controller in Splitting process

The Domain Controller Security Control is active in three stages. These steps shall be conducted in order, if there is a failure at any stage it shall be reported as a protocol error and the process shall end.

1) The ADM component requests its Security Control to prepare to the Split. To that end, the Security Control does the following:

   - First, it checks whether the requesting Instance is from the same AD and DC capable. If not, the DC Split is refused.
   - It checks then whether `DC_split_count` is lower than `DC_split_ceiling`. If not, the Split is refused.

- Then it checks whether the requesting Instance is Local or Remote via a standard Proximity Test.

If SRTT or SRTTL is used to check Local-ness, the AD key shall be used to authenticate the Proximity Test.

- If the requesting Instance is Local, the DC Split is authorized.
- If it is Remote, the DC Split is authorized if `DC_remote_count` is lower than `DC_remote_ceiling`. Else it is refused.
- The Security Control then informs the ADM component of the result.

The Security Control then runs the SAC establishment procedure with the Security Control of the requesting Instance. During the SAC establishment, it verifies that all relevant information transmitted within the request match the certificate information. If anything fails, it is signalled to the ADM component.

2) Upon reception of a *`DC_split_commit_message`*, the ADM component shall inform its Security Control to proceed with the DC Split.

Security Control increments the `DC_split_count` by one . If the DC Split is Remote, the Security Control also increments the `DC_remote_count` by one. It then computes its new ADSE values as the difference between existing ones and requested ones. The process is stopped, the Domain Controller reverts to its initial ADSE values and an error message is issued to its ADM component if any of the following occurs:

- There is a negative count or ceiling.
- The `remote_count`, `total_count`, `DC_split_count` or `DC_remote_count` is lower than their relevant ceiling.
- The `total_count` is not the sum of `local_count` and `remote_count`.

If all controls are successful, the Security Control records the requesting CPCM Instance Certificate Identifier and sends the requested ADSE values in a *`become_DC_message`* and waits for confirmation. Once the *`new_DC_message`* is received, the Security Control component informs its corresponding ADM component that the DC Split was successful. If this message was not received, Security Control informs its corresponding ADM component of the DC Split failure.

If the Device Application interrupts the protocol at that step, the Security Control will erase the recorded CPCM Instance Certificate Identifier but NOT revert to its original ADSE values.

The *`become_DC_message`* also includes a list of CICF identifiers taken from the list of CICFs that Joined remotely, and chosen from the set requested by the split DC. This list of identifiers will constitute the list of CICFs that Joined remotely in the Split DC. These identifiers are removed from the list of the splitting DC.

3) Finally, upon receipt of a *`DC_split_finish_message`*, or upon other events as described in clause 5.3, the ADM component informs its Security Control that the DC Split was achieved. Security Control then erases the recorded CPCM Instance Certificate Identifier.

6.7 Domain Controller Merge

6.7.1 Merging Domain Controller

The Security Control is involved at four steps during the Domain Controller Merge which shall be executed in the described order. Any inversion or omission shall result in a negative result and any failure of one step will cause the process to stop.

1) After the *`DC_merge_begin_message`* is sent, it will accept the challenge to run the SAC establishment procedure with any requesting Instance.

2) Just before sending the *`DC_merge_commit_message`*, the ADM component informs its Security Control that the DC Merge is about to occur through a *`prepare_to_DC_merge_message`*. The Security Control checks the SAC has been established with a Domain Controller whose certificate corresponds to a CICF Instance.

Security Control records the CPCM Instance Certificate Identifier of the Domain Controller and disables the ADSE values. This means that this entity will not be able to run any other Domain Controller protocol with another CPCM Instance but will still act as a Domain Controller.

3) Upon receipt of a *merge_DC_message*, it confirms the DC Merge in a *DC_merged_message*.

4) Once the ADM component has received a *DC_merge_confirm_message*, it requests its Security Component to complete the DC Merge procedure through a *DC_merge_completed_message*. The Security Control erases the recorded Domain Controller identifier and ADSE values and ceases to be a Domain Controller.

If the protocol is intentionally interrupted by the Device Application before the fourth step occurred (e.g. because the user decided to run the protocol with another Domain Controller), all the recorded data (Domain Controller identifier and/or ADSE values) shall be erased and the Instance will cease to be Domain Controller.

6.7.2 Merged Domain Controller process

The Domain Controller Security Control is active in four stages which shall be conducted in order, if there is a failure at any stage it shall be reported as a protocol error and the process shall end.

1) The ADM component requests its Security Control to prepare to DC Merge. To that end, the Security Control checks whether the requesting Instance is from the same AD and if it is DC capable. The Security Control then informs the ADM component of the result.

2) The ADM component will then seek to establish a SAC with the requesting Instance. This only happens if the Device Application authorized the DC Merge. The Security Control then runs the SAC establishment procedure with the Security Control of the requesting Instance. During the SAC establishment, it verifies that all relevant information transmitted within the request match the Certificate information. If anything fails, it is signalled to the ADM component.

3) Upon receipt of the *DC_merge_commit_message*, the ADM component will inform its Security Control to proceed with the DC Merge.

   The Security Control decrements `DC_split_count` by one. It then computes (but does not enable) its new ADSE values as the sum of the existing ones and the received ones and merges its list of CICFs that Joined remotely with the received list. These ADSE values and list are not yet enabled.

   The process is stopped, the Domain Controller reverts to its initial ADSE values and an error message is issued to the ADM component if any of the following occurs:

   - remote_count, total_count, DC_split_count or DC_remote_count is lower than relevant ceiling.
   - total_count is not the sum of local_count and remote_count.

   If all controls are successful, the Security Control records the requesting CPCM Instance Certificate Identifier and sends a *merge_DC_message* and waits for confirmation. Once the *DC_merged_message* is received, it informs its ADM component that the DC Merge was successful. If this message was not received, it informs its ADM component of the DC Merge failure.

   If the Device Application interrupts the protocol at that step, Security Control will erase recorded CPCM Instance Certificate Identifier AND revert to its original ADSE values.

4) Finally, upon receipt of a *DC_merge_finish_message*, or upon other events as described in clause 5.3, the ADM component informs its Security Control that the DC Merge was achieved. Security Control then enables the new ADSE values and list and erases the recorded CPCM Instance Certificate Identifier.

6.8 Domain Controller Rebalance

6.8.1 The Rebalancing Domain Controller

The Security Control is involved at four steps during the Domain Controller Rebalance which shall be executed in the described order. Any inversion or omission shall result in a negative result and any failure of one step will cause the process to stop.

1) After the *DC_rebalance_begin_message* is sent, it will accept the challenge to run the SAC establishment procedure with any requesting Instance.

2) Just before sending the *DC_rebalance_commit_message*, the ADM component informs its Security Control that the DC Rebalance is about to occur through a *prepare_to_rebalance_DC_message*. Security Control first checks the following conditions:

   - The SAC has been established with a Domain Controller whose certificate corresponds to a CICF Instance.
   - The requested ADSE counts are not higher than the sum of the requesting instance's current ones with the ones of the Rebalanced Domain Controller.
   - The requested `total_count, remote_count, DC_split_count` and `DC_remote_count` are not higher than their respective ceilings. It also checks that the resulting counts for the Rebalanced Domain Controller will obey the same rules (resulting counts being the difference between current counts and requested counts).
   - The requested `total_count` shall be the sum of `local_count` and `remote_count`.
   - The requested CICFs that Joined Remotely are in the corresponding list of the Rebalanced DC.

   If the conditions above are all met, it records the CPCM Instance Certificate Identifier of the Domain Controller.

   Else, it informs its ADM component of the bad result.

3) Upon receipt of the *rebalance_DC_message*, it checks that the received ADSE values match the expected ones. If so, it records them but does not enable them yet. It sends back the result in a *new_DC_message*.

4) Once the ADM component has received a *DC_rebalance_confirm_message*, it requests its Security Component to complete the DC Rebalance procedure through an *enable_new_ADSE_counts_message*. The Security Control erases the recorded Domain Controller identifier and enables the new received ADSE counts and the new list of CICFs that Joined Remotely.

If the Device Application intentionally interrupts the protocol before the fourth step (e.g. because it knows it will never finish), the recorded CPCM Instance Certificate Identifier shall be erased. ADSE values will also be erased and the Instance will cease to be Domain Controller.

6.8.2 Rebalanced Domain Controller

The Domain Controller Security Control is active in four stages which shall be conducted in order, if there is a failure at any stage it shall be reported as a protocol error and the process shall end.

1) The ADM component requests its Security Control to prepare to be Rebalanced. To that end, the Security Control checks whether the requesting Instance is from the same AD and is a Domain Controller. If not, the DC Rebalance is refused.

   The Security Control then informs the ADM component of the result.

2) The ADM component will then seek to establish a SAC with the requesting Instance. This happens if the Device Application authorized the DC Rebalance. The Security Control then runs the SAC establishment procedure with the Security Control of the requesting Instance. During the SAC establishment, it verifies that all relevant information transmitted within the request matches the Certificate information. If anything fails, it is signalled to the ADM component.

3) Upon receipt of a *DC_rebalance_commit_message*, the ADM component will inform its Security Control to proceed with the DC Rebalance.

   Security Control then computes its new ADSE values as the difference between its current ADSE values and the requested ones. These ADSE values shall be used if any other ADM protocol starts.

   The process is stopped, the Domain Controller reverts back to its initial ADSE values and an error message is issued to its ADM component if any of the following occurs:

   - There is a negative count or ceiling in the new (but not yet enabled) counts.
   - Any of `remote_count, total_count, DC_split_count` or `DC_remote_count` is higher than their relevant ceilings.
   - `total_count` is not the sum of `local_count` and `remote_count`.

   If all controls are successful, the Security Control record the requesting CPCM Instance Certificate Identifier and transfers the requested ADSE values in a *rebalance_DC_message* and waits for confirmation. Once the *DC_rebalanced_message* is received, it informs its ADM component that the DC Rebalance was successful. If this message was not received, it informs its ADM component of the DC Rebalance failure.

   The *DC_rebalanced_message* also includes a list of CICF identifiers taken from the list of CICFs that Joined remotely amongst the set requested by the rebalancing DC. These identifiers are removed from the list of the rebalanced DC.

   If the Device Application interrupts the protocol at that step, Security Control will erase recorded CPCM Instance Certificate Identifier AND revert to its original ADSE values.

4) Finally, upon receipt of a *DC_rebalance_finish_message*, or upon other events as described in clause 5.3, the ADM component informs its Security Control that the DC Rebalance was achieved. Security Control then enables new ADSE values and list and erases the recorded CPCM Instance Certificate Identifier.

7 Generic ADSE tools (optional)

7.1 Single Household Metric (SHM)

The SHM tool provides a means to associate an AD with a unique household. Examples of an SHM include requirements of the use of a password/PIN, security dongle, or other anonymous household identification metric prior to performed AD-related functions, such as an AD Join, or prior to allowing AD-related usage.

NOTE: In the mandatory ADSE method, the initial single Domain Controller and Locally Split Domain Controllers are rudimentary examples of the SHM since they are designed to accommodate the needs of a single household and to be below the needs of multiple households.

The details of the SHM tool will be defined by the C&R regime.

7.2 Total AD Count (TAC)

The Total AD Count tool shall use the following data field: `total_ceiling`, which will be protected against unauthorized modification.

If the number of Instances in the AD would exceed `total_ceiling` though the addition of another Instance, then this test shall return a result that the "ADSE would be exceeded".

If an Instance becomes disconnected from the other AD Instances and it wishes to be able to manage the AD Join of other Instances to its AD, then it must reserve a certain subset of the remainder of the allowed additional Instances by using the Domain Controller functionality Split function.

All allocated Instances to such a "cluster" will be assumed to be fully allocated by all other "clusters" in their ADSE calculations.

7.3 Total and Remote AD Count (TARC)

The TARC tool shall use the following data fields: `total_ceiling` and `remote_ceiling`, which will be protected against unauthorized modification.

AD Joining can be limited to only Instances that are Local to one AD member Instance. The details of the definition of Local (to any AD Instance or a number of AD Instances) will be established by the governing C&R regime.

A limited number of Remote AD Joins may be permitted as defined by `remote_ceiling`.

The `remote_ceiling` can be replenished by an Instance becoming Local to one, or alternatively, a certain percentage, of the other Instances that are members of that AD. The details of this rule will be established by the C&R regime.

The Split Domain Controller functionality allows management within multiple connected local AD "clusters" of devices, for example a group of devices on a DVB Home Network in the main family home and another at a distant vacation home. Two or three Instances in the same AD may be able to talk to each other and another two or three may be able to talk to each other, but not to the first group. Each subset of the AD Instances may be allocated a certain portion of the total remaining Instances.

7.4 Total and Remote AD Count + (TARC+)

The TARC+ tool shall use the following data fields: `total_ceiling`, `remote_ceiling`, `connected_device_floor`, which will be protected against unauthorized modification.

This tool is the same as the TARC tool with an additional rule regarding the minimum number of AD Members that must be connected to allow an AD Join.

Quorum Test is an example of TARC+ ADSE tool where `connected_device_floor` is set to a given quorum percentage of AD CICF.

7.5 Quorum Test

The Quorum Test tool shall use the following data fields: `total_ceiling`, `local_ceiling` and `remote_ceiling`, which will be protected against unauthorized modification.

The Quorum Test expands on the TARC Tool. In TARC:

Number of local CICF Joins that can be allowed = `total_ceiling` - `remote_ceiling`

Or

`total_ceiling`= Derived "`local_ceiling`" + `remote_ceiling`.

With the Quorum Test, a separate `local_ceiling` field is included and

`total_ceiling` ≥ `local_ceiling` + `remote_ceiling`.

The total allowed CICFs is increased above the mere sum of the Local and Remote allowed CICFs. An Instance beyond the local CICF limit can be added as long as it passes the "Quorum Test". The Quorum Test passes if the number of CICF that are Local to the Domain Controller with which the CICF is attempting to Join the AD is equal to or more than the "quorum percentage" of the current counter `local_count`. The value of the "quorum percentage" will be determined by the C&R regime.

7.6 Domain Membership History (DMH)

The DMH tool shall use the `history_ceiling` data field, which will be protected against unauthorized modification.

This tool limits the number of times that a CPCM Instance may Join new ADs. The `history_ceiling` field can be modified by an ADM Authorized Authenticated Agent (ADMAAA), e.g. increased to allow more Joins.

Since a CPCM Instance can only be a member of one AD at a time, a CPCM Instance must, by definition AD Leave its prior AD before Joining a new AD.

The AD Join protocol increments the `history_count` by '1' for each successful AD Joined. The `history_count` must be protected against unauthorized modification. If the `history_count` reaches the `history_ceiling`, then no further AD Joins can be performed unless the `history_ceiling` data field is increased by the ADMAAA.

NOTE: The C&R regime will define the data field value, and it may also establish different values for different classes of CPCM Devices, e.g. owned versus rental models.

7.7 Wayfaring Device Limits (WDL)

The WDL tool shall use the following data fields: `disconnnected_time_limit` and `remote_time_limit`, which will be protected against unauthorized modification. These two fields provide the ability to expire the membership of a CPCM Instance in an AD and can be used for both fraud detection/action and lost device count recovery. An Instance's membership in an AD will be terminated, both by the Instance itself, and by the Domain Controller, if either or both of these situations occur:

- Time since last connection over a Secure Authenticated Channel with a Domain Controller of the same AD exceeds `disconnected_time_limit`.
- Time since last Proximity Test that returned a Local value between this Instance and a Domain Controller of the same AD exceeds `remote_time_limit`.

NOTE 1: The C&R regime will define the data field values.

NOTE 2: The usage of this tool supposes that each CPCM Instance has permanent access to a source of secure absolute time.

7.8 AD Membership by Authorized Authenticated Agent (ADMAAA)

This tool causes:

1) A forced positive ADSE test to be asserted by the trusted ADMAAA so that the normal ADM processes can continue and allow a CPCM Instance to Join the AD; or

2) the revision of the values of:

   - `total_ceiling`.
   - `remote_ceiling`.
   - `local_ceiling`.
   - `DC_split_ceiling`.
   - `DC_remote_ceiling`.

ADMAAA has a CPCM Certificate whose `ADSE_countable` field is set and all other AD related fields are cleared. The tool can be exercised after a SAC with ADMAAA has been established and using *`ADMAAA_tool_request_message`* and *`ADMAAA_tool_response_message`*.

8 ADM message structure and elements

Please see the System Specification (TS 102 825-4 [2]) for the definition of ALL CPCM messages.

Please see TS 102 825-4 [2] for the actual messages that are exchanged *over* the ADM1 Interface (see figure 3) between Authorized Domain Management Instances.

8.1 Local Master Delegation

When receiving a message from an AD Member that needs to be delegated, the Local Master places the requesting CPCM Instance Certificate Identifier in the `delegating_instance_id` field.

The Domain Controller shall answer with the same `delegating_instance_id` field. Then, the Local Master returns the message to the requesting Instance with the `delegating_instance_id` field set to the value of the CPCM Instance Certificate Identifier of the Domain Controller with which the protocol is being performed.

Similarly, when responding, the requesting Instance shall answer with the `delegating_instance_id` field set to the value of the CPCM Instance Certificate Identifier of the Domain Controller with which the protocol is being performed.

The ADM status information element is NOT updated by the Local Master when forwarding a message.

Annex A (informative): Informative messages

This annex describes the messages that are exchanged internally between the Device, Security Control and ADM modules, since these messages are NOT exposed externally to other CPCM Instances they are up to the individual implementation, but are presented here for information only.

A.1 Interaction Security Control ADM

This clause describes the messages that are exchanged between the Security Control and ADM modules.

Table A.1: ADM Communications with CPCM Security Control

Primitive	From	To	Parameters
AD_join_ok_message	ADM	SEC	CIC Identifier
AD_join_request_message	ADM	SEC	CIC Identifier, ADM status
AD_leave_request_message	ADM	SEC	CIC Identifier, ADM status
AD_leaving_completed_message	ADM	SEC	-
AD_leaving_ok_message	ADM	SEC	CIC Identifier
authorize_AD_join_message	ADM	SEC	-
DC_merge_completed_message	ADM	SEC	CIC Identifier
DC_merged_message	ADM	SEC	CIC Identifier
DC_rebalance_ok_message	ADM	SEC	CIC Identifier
DC_split_done_message	ADM	SEC	CIC Identifier
DC_split_request_message	ADM	SEC	CIC Identifier, ADM status, ADID
DC_transferred_message	ADM	SEC	CIC Identifier
delegated_AD_join_message	ADM	SEC	CIC Identifier
delegated_AD_leaving_message	ADM	SEC	CIC Identifier
disable_AD_secret_message	ADM	SEC	CIC Identifier
enable_AD_secret_message	ADM	SEC	-
enable_DC_message	ADM	SEC	-
enable_new_ADSE_counts_message	ADM	SEC	CIC Identifier
establish_SAC_message	ADM	SEC	CIC Identifier
instance_leaving_AD_message	ADM	SEC	CIC Identifier
merge_DC_message	ADM	SEC	CIC Identifier, ADSE values
prepare_AD_join_message	ADM	SEC	CIC Identifier, ADM status
prepare_DC_transfer_message	ADM	SEC	CIC Identifier, ADM status, ADID
prepare_to_be_additional_DC_message	ADM	SEC	CIC Identifier, ADM status, ADSE values
prepare_to_be_DC_message	ADM	SEC	CIC Identifier, ADM status
prepare_to_DC_merge_message	ADM	SEC	CIC Identifier, ADM status
prepare_to_rebalance_DC_message	ADM	SEC	CIC Identifier, ADM status, ADSE values
send_AD_secret_message	ADM	SEC	CIC Identifier, ADM status
split_DC_message	ADM	SEC	CIC Identifier, ADSE Values
transfer_DC_message	ADM	SEC	CIC Identifier
update_ADSE_counts_message	ADM	SEC	CIC Identifier, ADSE values
AD_created_message	SEC	ADM	ADID, ADSE Values, Result
AD_join_allowed_message	SEC	ADM	Result
AD_join_authorize_message	SEC	ADM	Result
AD_join_prepared_message	SEC	ADM	Result
AD_secret_disabled_message	SEC	ADM	Result
AD_secret_enabled_message	SEC	ADM	Result
AD_secret_sent_message	SEC	ADM	Result
ADSE_counts_updated_message	SEC	ADM	Result
DC_merge_prepared_message	SEC	ADM	Result
DC_merged_message	SEC	ADM	Result
DC_split_allowed_message	SEC	ADM	Result
DC_split_prepared_message	SEC	ADM	Result
DC_splitted_message	SEC	ADM	Result
DC_transfer_allowed_message	SEC	ADM	Result
DC_transfer_prepared_message	SEC	ADM	Result

Primitive	From	To	Parameters
delegated_AD_join_OK_message	SEC	ADM	CIC Identifier
delegated_AD_leaving_OK_message	SEC	ADM	CIC Identifier
prepared_to_rebalance_DC_message	SEC	ADM	Result
SAC_established_message	SEC	ADM	Result
SEC_AD_leaving_done_message	SEC	ADM	Result
transfer_DC_ok_message	SEC	ADM	Result

A.2 ADM Communications with Device Application

This clause describes messages exchanged between Device Application and ADM modules.

Table A.2: Device Application to ADM Communication Primitives

Primitive	From	To	Parameters
accept_AD_join_message	ADM	DA	CIC Identifier
accept_DC_merge_message	ADM	DA	CIC Identifier
accept_DC_rebalance_message	ADM	DA	CIC Identifier
accept_DC_split_message	ADM	DA	CIC Identifier
accept_DC_transfer_message	ADM	DA	CIC Identifier
AD_join_done_message	ADM	DA	-
AD_renamed_message	ADM	DA	Result
choose_DC_message	ADM	DA	Set of ADID
choose_AD_message	ADM	DA	Set of ADIDs, AD Names
confirm_AD_join_message	ADM	DA	ADID, AD Name
Confirm_AD_leaving_message	ADM	DA	-
create_new_AD_message	ADM	DA	-
DC_merge_done_message	ADM	DA	Result
DC_merge_message	ADM	DA	-
DC_rebalance_message	ADM	DA	CIC Identifier
DC_rebalanced_message	ADM	DA	Result
DC_split_ok_message	ADM	DA	Result
DC_transfer_confirm_message	ADM	DA	CIC Identifier
DC_transfer_OK_message	ADM	DA	Result
join_AD_message	ADM	DA	ADID
no_longer_an_AD_member_message	ADM	DA	Result
transfer_DC_message	ADM	DA	CIC Identifier
which_balance_message	ADM	DA	ADSE Counts
AD_choice_message	DA	ADM	ADID
ADSE_counts_message	DA	ADM	ADSE Counts
become_additional_DC_message	DA	ADM	-
become_DC_message	DA	ADM	-
cease_DC_message	DA	ADM	-
eject_from_AD_message	DA	ADM	CIC Identifier
initialize_message	DA	ADM	-
invite_to_AD_join_message	DA	ADM	CIC Identifier
leave_AD_message	DA	ADM	-
rebalance_DC_message	DA	ADM	-
rename_AD_message	DA	ADM	AD Name
response_message	DA	ADM	Yes/No
select_DC_message	DA	ADM	CIC Identifier

List of tables

List of figures

History

Document history		
V1.1.1	July 2008	Publication
V1.2.1	February 2011	Publication