

**Telecommunications and Internet Protocol
Harmonization Over Networks (TIPHON) Release 4;
Protocol Framework Definition;
Part 2: Registration and Service Attachment
service meta-protocol definition**



Reference

RTS/TIPHON-03016-2R4

Keywords

interface, IP, meta-protocol, protocol, service,
VoIP

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, send your comment to:

editor@etsi.org

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 2003.
All rights reserved.

DECTTM, **PLUGTESTS**TM and **UMTS**TM are Trade Marks of ETSI registered for the benefit of its Members.
TIPHONTM and the **TIPHON logo** are Trade Marks currently being registered by ETSI for the benefit of its Members.
3GPPTM is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

Contents

| | |
|---|----|
| Intellectual Property Rights | 5 |
| Foreword..... | 5 |
| Introduction | 5 |
| 1 Scope | 6 |
| 2 References | 6 |
| 3 Definitions and abbreviations..... | 7 |
| 3.1 Definitions | 7 |
| 3.2 Abbreviations | 7 |
| 4 Registration service | 7 |
| 4.1 Purpose | 7 |
| 4.2 Description | 7 |
| 4.3 Procedures | 8 |
| 4.3.1 Provision/withdrawal | 8 |
| 4.3.2 Normal procedures..... | 8 |
| 4.3.2.1 Activation/deactivation | 8 |
| 4.3.2.2 Invocation and operation..... | 8 |
| 4.3.3 Exceptional procedures..... | 9 |
| 4.4 Interaction with other services or service capabilities | 9 |
| 4.5 Overall behaviour | 10 |
| 5 Functional entity model and information flows | 12 |
| 5.1 Functional entity model..... | 12 |
| 5.1.1 Description of model | 12 |
| 5.2 Information flows..... | 13 |
| 5.2.1 Relationship ra | 13 |
| 5.2.1.1 Register | 13 |
| 5.2.1.2 DeRegister..... | 14 |
| 5.2.2 Relationship rb..... | 14 |
| 5.2.2.1 Authorize..... | 14 |
| 5.2.2.2 Detach | 15 |
| 5.2.3 Relationship rc | 15 |
| 5.2.3.1 Attach..... | 15 |
| 5.2.4 Relationship rd..... | 15 |
| 5.2.4.1 Detach | 15 |
| 5.2.5 Summary of information element definitions | 16 |
| 5.2.6 Information flow sequences..... | 16 |
| 5.2.6.1 Normal operation | 17 |
| 5.2.6.1.1 Initial registration | 17 |
| 5.2.6.1.2 Change of SpoA (location update) | 18 |
| 5.2.6.1.3 DeRegistration..... | 19 |
| 5.2.6.2 Exceptional operation..... | 20 |
| 5.2.6.2.1 Invalid identity | 20 |
| 5.2.6.2.2 Service not available..... | 21 |
| 5.3 Registration functional entity actions | 21 |
| 5.3.1 Actions of RFE1 | 22 |
| 5.3.2 Actions of RFE2 | 22 |
| 5.3.3 Actions of RFE3 | 23 |
| 5.3.4 Actions of RFE4 | 23 |
| 5.4 Registration functional entity behaviour | 23 |
| 5.4.1 Behaviour of RFE1 | 23 |
| 5.4.2 Behaviour of RFE2 | 28 |
| 5.4.3 Behaviour of RFE3 | 34 |
| 5.4.4 Behaviour of RFE4 | 37 |
| 5.5 ASN.1 data definitions | 39 |

| | | |
|-----|--|-----------|
| 5.6 | Allocation of functional entities to domains..... | 41 |
| | Annex A (informative): Additional SDL used in modelling | 42 |
| | Annex B (informative): SDL model | 52 |
| | Annex C (informative): Bibliography..... | 53 |
| | History | 54 |

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 ETSI Project Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON).

The present document is part 2 of a multi-part deliverable. Full details of the entire series can be found in part 1 [9].

Introduction

The present document is a product in TIPHON Release 4 (see TR 101 301 [8]) of step C of the TIPHON development process described in ETSI TR 101 835 [7].

1 Scope

The present document defines by means of an information model, a functional entity behavioural model, and by validated SDL a model of the abstract behaviour of each service and service capability identified as being essential in TIPHON Release 4.

The present document defines the meta-protocol requirement for the registration and service attachment services in TIPHON.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

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

- [1] Void.
- [2] Void.
- [3] ETSI TR 101 877: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Requirements Definition Study; Scope and Requirements for a Simple call".
- [4] ETSI TS 101 878 V4.1.1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Service Capability Definition; Service Capabilities for TIPHON Release 4".
- [5] ITU-T Recommendation I.130: "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".
- [6] ETSI TS 102 165-1 V4.1.1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Protocol Framework Definition; Methods and Protocols for Security; Part 1: Threat Analysis".
- [7] ETSI TR 101 835: "Telecommunications and Internet Protocol Harmonization over Networks (TIPHON); Project method definition".
- [8] ETSI TR 101 301: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 3; Release Definition; TIPHON Release 3 Definition".
- [9] ETSI TS 101 882-1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Protocol Framework Definition; Part 1: Meta-protocol design rules, development method, and mapping guideline".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions given in TR 101 877 [3] and TS 101 878 [4] apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 101 877 [3], TS 101 878 [4] and the following apply:

| | |
|------|--|
| FE | Functional Entity |
| IMSI | International Mobile Subscriber Identity |
| MSC | Message Sequence Chart |
| RFE | Registration Functional Entities |
| SpoA | Service provider point of attachment |
| UML | Unified Modelling Language |

4 Registration service

4.1 Purpose

The purpose of the registration and service attachment service is to provide authorization of users to access services. In addition the registration and service attachment service provides a mechanism for security countermeasure C1 as identified in TS 102 165-1 [6].

4.2 Description

The registration service enables a user (the registrant) to seek and gain authority to invoke service in a domain for which access is strictly controlled. The service applications to be offered are determined, in part, by data held in the user profile.

Figure 1 shows the relationship of the core elements in registration.

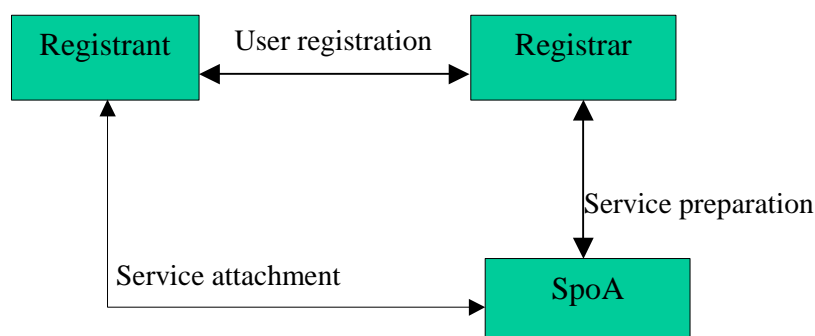
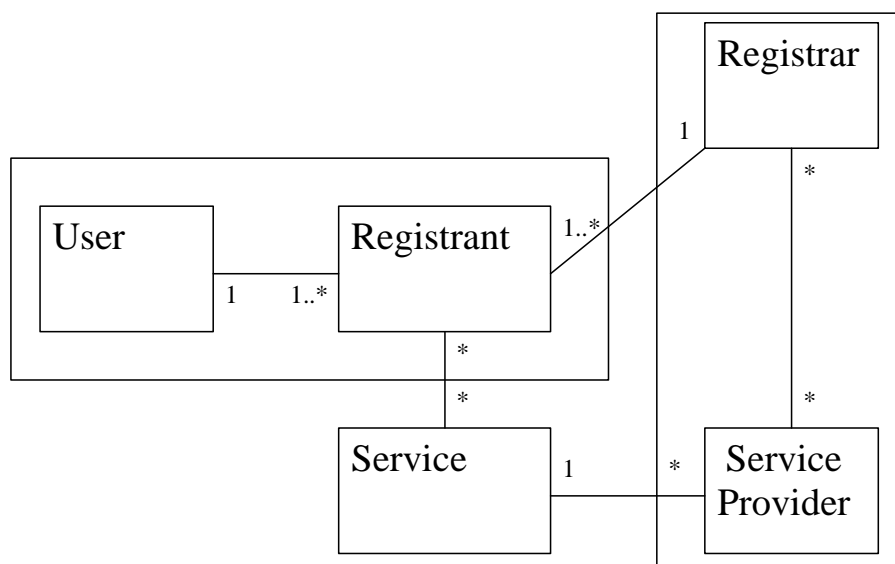


Figure 1: Relationship of registration entities

The registrar maintains a record of the location of the registrant and acts as an arbitrator on the provision of individual services to the registrant. Figure 2 shows by means of an UML class diagram the ordinality of the relationships to be considered in the registration and service attachment service.



- NOTE 1: A single user may be associated with many registrants.
 NOTE 2: A registrant shall be associated with only one user.
 NOTE 3: A registrant shall be associated with only one registrar.
 NOTE 4: A registrar may be associated with many registrants.
 NOTE 5: A service may be associated with many service providers.
 NOTE 6: In any registration instance a service shall be associated with only one service provider.
 NOTE 7: A service provider shall be associated with only one Service.
 NOTE 8: A registrant may be associated with many Services.

Figure 2: Ordinal relations in TIPHON

The registrar authorizes access to service applications in the domain in which the registrant is located or in another domain in which the entity controlling the service application is located.

If the registrar determines that the user is no longer authorized for a service application or set of service applications, e.g. if a pre-paid account has been depleted, the registrar shall inform the registrant and the service provider.

The registration service is offered across a network or set of networks (parts of which may be under different administrative control).

4.3 Procedures

4.3.1 Provision/withdrawal

Registration is available on a per-name/per service basis to all customers of a TIPHON system.

4.3.2 Normal procedures

4.3.2.1 Activation/deactivation

Registration shall be activated at provision and deactivated at withdrawal.

4.3.2.2 Invocation and operation

Registration shall be invoked by at least one of the following events:

- on power-up of the user equipment;
- on change of physical point of attachment;
- on change of logical point of attachment;

- on demand by the user; and
- on demand by the network.

A user shall indicate in a registration request those services to be registered to and shall be informed by the registrar where those services can be provided.

If a registrant registers from a new location to a service it is already registered to and a new service provider is allocated the registrar shall be able to delete the registrant's details from the old serving service provider.

4.3.3 Exceptional procedures

If the registrar is unable to determine where the requested service can be provided it shall inform the registrant with an indication of the reason for the rejection of the registration request. Possible rejection causes include inadequate capacity at an available service provider to maintain the required level of service.

4.4 Interaction with other services or service capabilities

The registration and service attachment service described in the present document implements the following service capabilities defined within the *Profile* class of TS 101 878 [4] for TIPHON Release 4:

- attach;
- authorize;
- deregister;
- detach; and
- register.

In addition the service capability *authenticate* may be invoked during registration. In such cases any failure of authentication shall terminate registration without the authorization of service to the registrant.

Some capabilities of the call and bearer classes may, depending on the overall service definition, be dependent upon the use of service capabilities which are described in the present document. The interaction with services and service capabilities implemented in such services is outside the scope of the present document.

4.5 Overall behaviour

The overall behaviour is shown in figures 3 and 4 for the registration service.

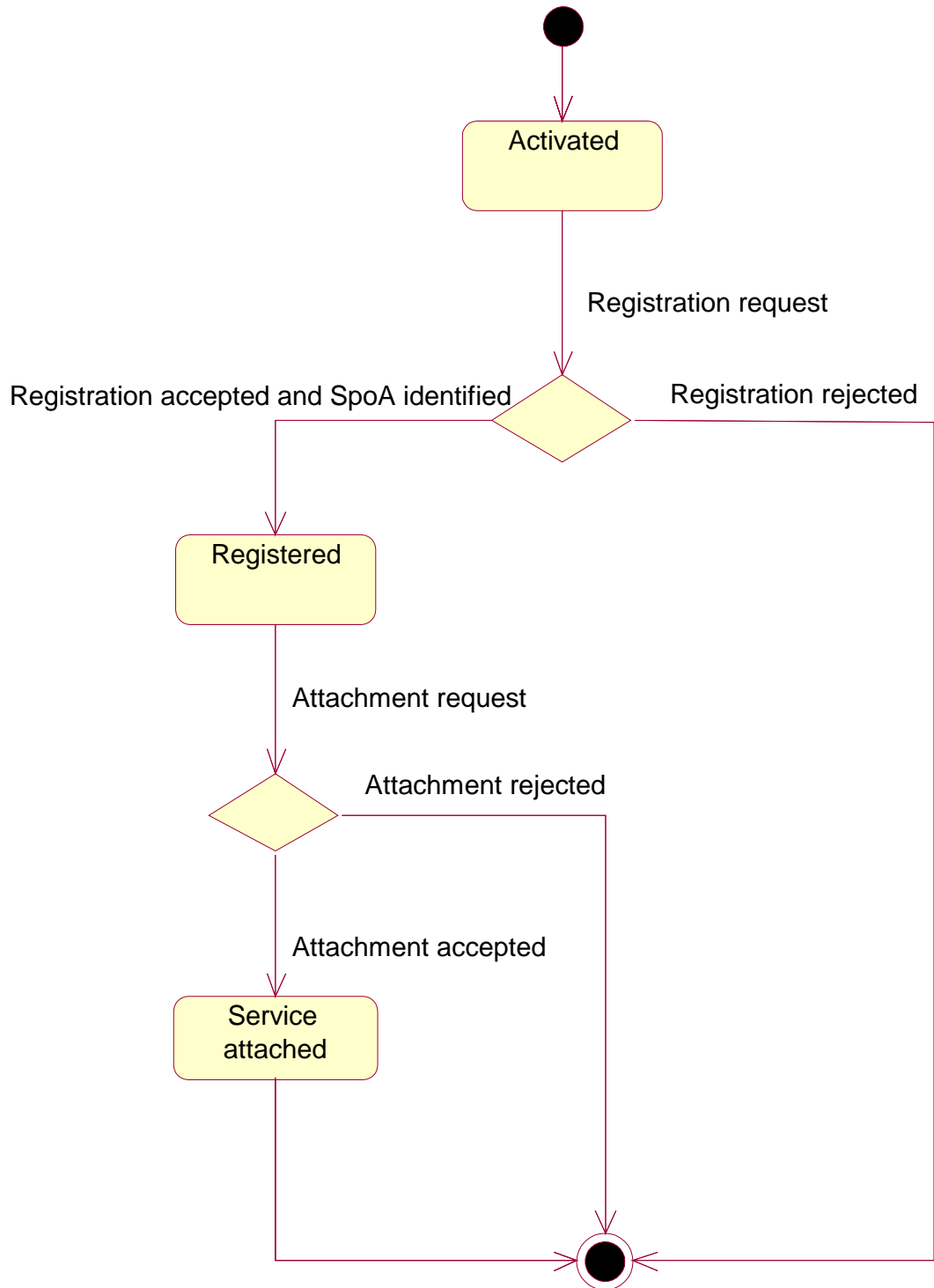


Figure 3: Overall behaviour for registration service expressed as an UML activity diagram

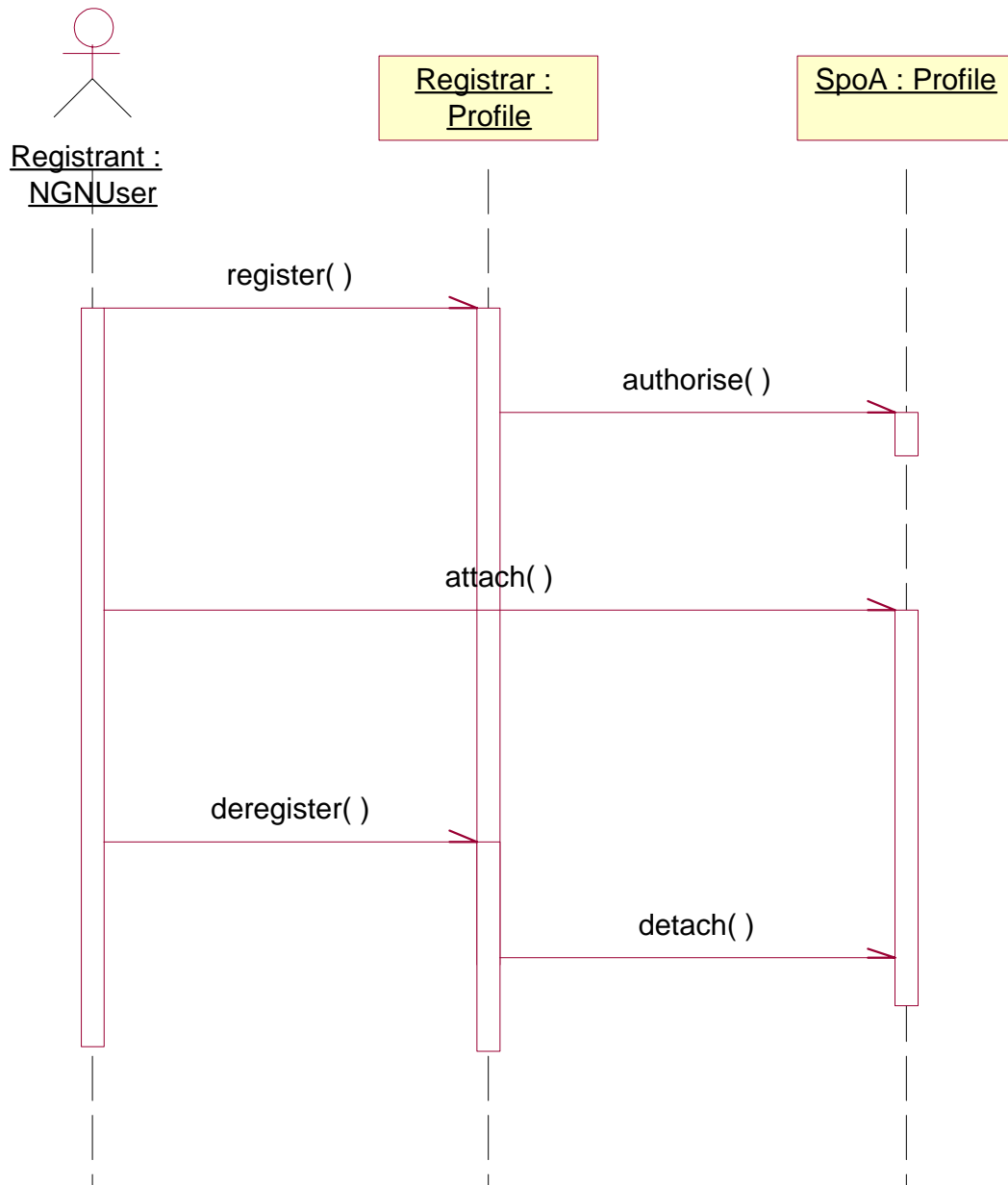


Figure 4: Overall behaviour in terms of TIPHON service capabilities

5 Functional entity model and information flows

5.1 Functional entity model

5.1.1 Description of model

The functional model shall comprise the following Registration Functional Entities (RFE):

Table 1: Registration functional elements

| Identity | Description |
|------------------|---|
| Registering user | |
| RFE1 | Registrant, the logical entity being registered |
| RFE2 | Registrar, holder of user profile of the registrant |
| RFE3 | Serving service provider point of Attachment (SpoA) |
| RFE4 | Previous SpoA |

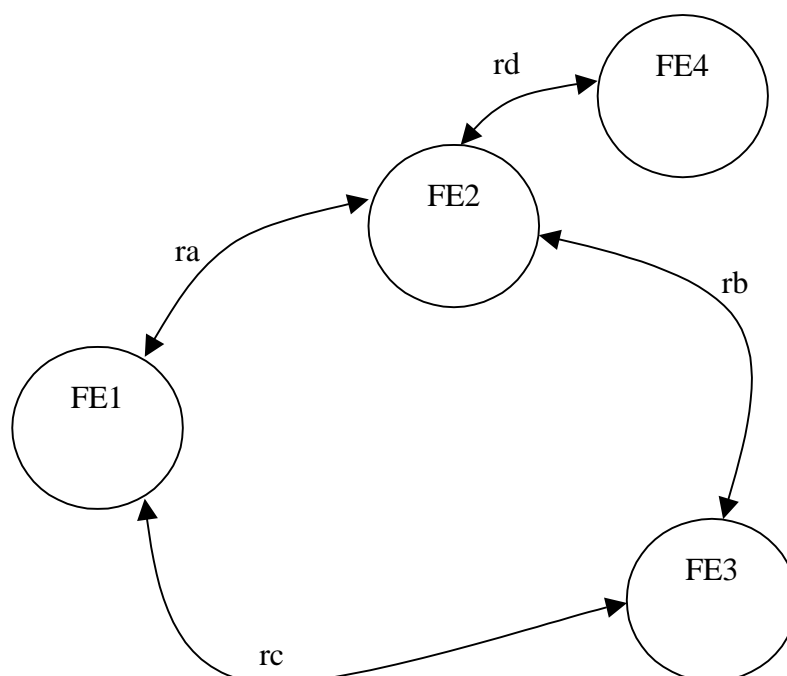


Figure 5: Relationships between functional entities for registration

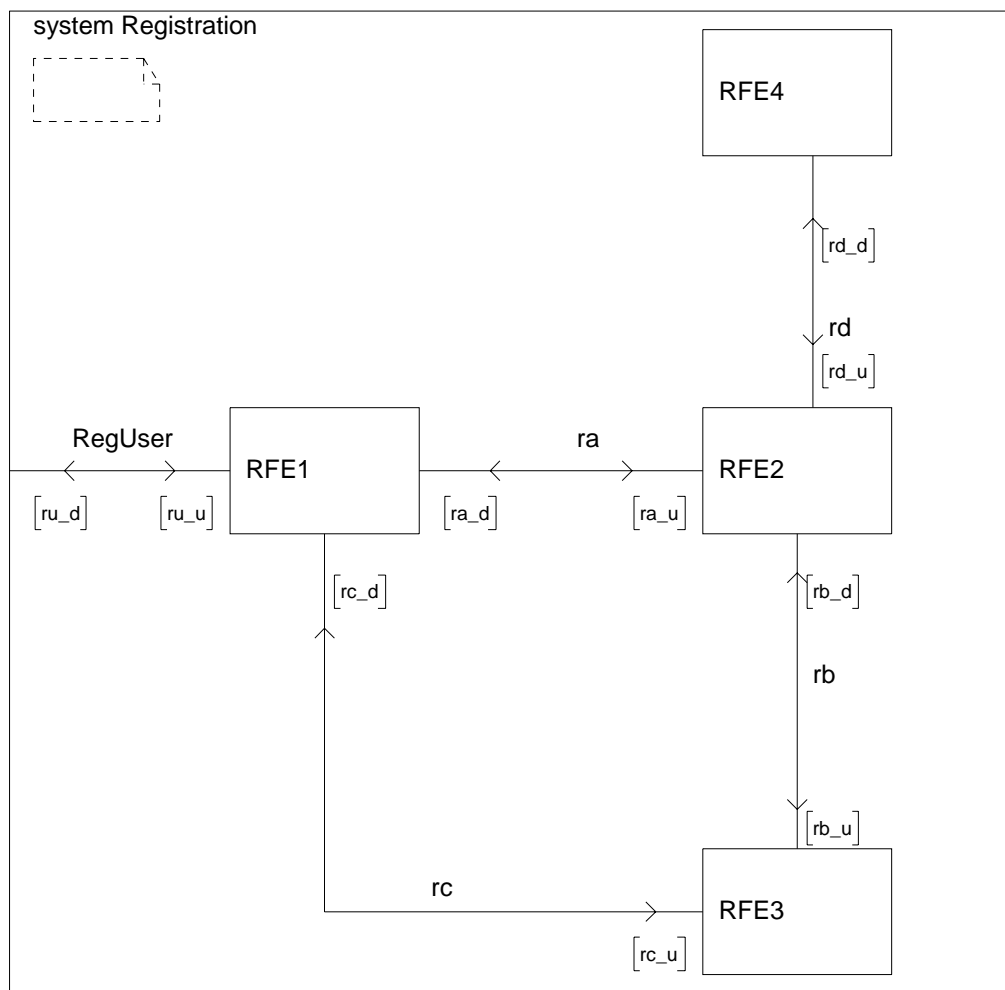


Figure 6: Relationships between functional entities for registration as SDL system diagram

5.2 Information flows

The following conventions are used to identify information flows:

- an information flow is referred to as a "request" at the FE that sends it and as an "indication" at the FE that receives it, this is shown in the tables defining each information flow as "req_ind";
- the corresponding confirmation is referred to as a "response" at the FE that sends it and as a "confirmation" at the FE that receives it, this is shown in the tables defining each information flow as "resp_conf".

5.2.1 Relationship ra

5.2.1.1 Register

Register is a confirmed information flow used by RFE1 to request the RFE2 to open the user's profile.

Table 2: Register information flow

| Information element | Value | Req_ind | Resp_conf |
|---|---|---------|----------------|
| TIPHON-reg-id | Any | M | M |
| RegistrationMode | Initial registration Location update | M | - |
| Location | Any | M | - |
| ServiceName | TIPHONSimpleCall ... | M | O (see note 2) |
| Result | Registration successful Registration-Id invalid Service unavailable | - | M |
| ServiceProviderName | Any | - | O (see note 1) |
| ClientAuthorisationToken | Any | - | O (see note 1) |
| NOTE 1: Provided if Result = "Registration successful". | | | |
| NOTE 2: Provided if Result = "Service unavailable". | | | |

5.2.1.2 DeRegister

DeRegister is a confirmed information flow that shall be used by RFE1 to request clearance of service attachments.

Table 3: DeRegister information flow

| Information element | Value | Req_ind | Resp_conf |
|---------------------|--|---------|-----------|
| TIPHON-reg-id | Any | M | M |
| ServiceName | TIPHONSimpleCall ... | M | - |
| Result | Deregistration successful Registration-Id invalid | - | M |

5.2.2 Relationship rb

5.2.2.1 Authorize

The Authorize information flow shall be used by RFE2 to request RFE3 for authorization to allow a client to attach for service.

Table 4: Authorize information flow

| Information element | Value | Req_ind | Resp_conf |
|--|--|---------|--------------|
| Registrar-id | Any | M | M |
| TIPHON-reg-id | Any | M | M |
| ServiceName | TIPHONSimpleCall ... | M | - |
| ClientAuthorisationToken | Any | - | O (see note) |
| Result | Service authorized to client ResourceNotAvailable | - | M |
| NOTE: This information element shall be provided if the value of Result is "OK". | | | |

5.2.2.2 Detach

The detach information flow shall be used by RFE2 to inform RFE3 that a client has requested deregistration from the service.

Table 5: Detach information flow

| Information element | Value | Req_ind | Resp_conf |
|---------------------|--|---------|-----------|
| Registrar-id | Any | M | M |
| TIPHON-reg-id | Any | M | M |
| ServiceName | TIPHONSimpleCall ... | M | - |
| Result | Service detachment successful Identity not recognized | - | M |

5.2.3 Relationship rc

5.2.3.1 Attach

The attach information flow shall be used by RFE1 to request attachment to RFE3.

NOTE: The service attachment may be sent at service invocation of the service to be attached.

Table 6: Attach information flow

| Information element | Value | Req_ind | Resp_conf |
|---------------------|---|---------|-----------|
| Registrar-id | Any | M | M |
| TIPHON-reg-id | Any | M | M |
| ServiceName | TIPHONSimpleCall ... | M | - |
| AuthorisationToken | Any | M | - |
| Result | Service attachment successful Identity not recognized Authorization expired | - | M |

5.2.4 Relationship rd

5.2.4.1 Detach

The detach information flow shall be used by RFE2 to inform RFE4 that a client should no longer be offered service.

Table 7: Detach information flow

| Information element | Value | Req_ind | Resp_conf |
|---------------------|--|---------|-----------|
| Registrar-id | Any | M | M |
| TIPHON-reg-id | Any | M | M |
| ServiceName | TIPHONSimpleCall ... | M | - |
| Result | Service detachment successful Identity not recognized | - | M |

5.2.5 Summary of information element definitions

This clause further defines the information elements present in each of the information flows defined in clauses 5.2.1 through 5.2.4.

TIPHON-reg-id: A unique name used to identify the registrant. This name should be structured in such a way that the home containing the registrant's profile (i.e. the Registrar) can be determined.

EXAMPLE: The International Mobile Subscriber Identity (IMSI) used in GSM may be considered as an example of TIPHON-reg-id.

RegistrationMode: This element indicates to the registrar if the user is registering for the first time, or wishes to update/refresh a previous registration. This is defined as an extensible list.

Location: This element indicates the current location of the registrant, and allows the registrar to offer a service provider local to the registrant where the service requires it (e.g. for any service where QoS is guaranteed). May be provided as a network address (e.g. IPv4 or IPv6 address, telephone number of a fixed line) or as a geographic location depending upon the service requested.

ServiceName: This element identifies the service that is to be registered to. The name has to be meaningful to each party in the registrant facing group (i.e. registrant, registrar, SpoA). This is defined as an extensible list.

Result: This element provides the result to any information flow and appears in the "_resp_conf" flows.

ServiceProviderName: A unique name used to identify the SpoA. This name should be structured in such a way that the address of the SpoA can be determined using conventional resolution protocols.

ClientAuthorisationToken: This is an authorization token prepared by the SpoA in response to the Authorise_req_ind information flow and is intended to be given back by the client at "attach" for verification. The format of the token is not fully defined in the present document.

Registrar-id: A unique name used to identify the registrar. This name should be structured in such a way that the address of the registrar can be determined using conventional resolution protocols.

AuthorisationToken: see ClientAuthorisationToken.

5.2.6 Information flow sequences

This clause specifies the information flow sequences for the registration and service attachment service.

NOTE 1: The information flow sequences are produced with a MSC editor; however, the scenarios are not MSCs but information flow sequences as defined in ITU-T Recommendation I.130 [5].

NOTE 2: In accordance with the ITU-T Recommendation I.130 [5] the invoking side is placed as the leftmost entity in the information flow sequences.

The step D for registration and service attachment shall provide signalling procedures in support of the information flow sequences specified in this clause. In addition, signalling procedures should be provided to cover other sequences arising from error situations, interactions with simple call, interactions with different topologies, etc.

In the information flow sequences, registration and service attachment information flows are represented by arrows.

The following information flow sequences are intended as guidance in further development.

5.2.6.1 Normal operation

5.2.6.1.1 Initial registration

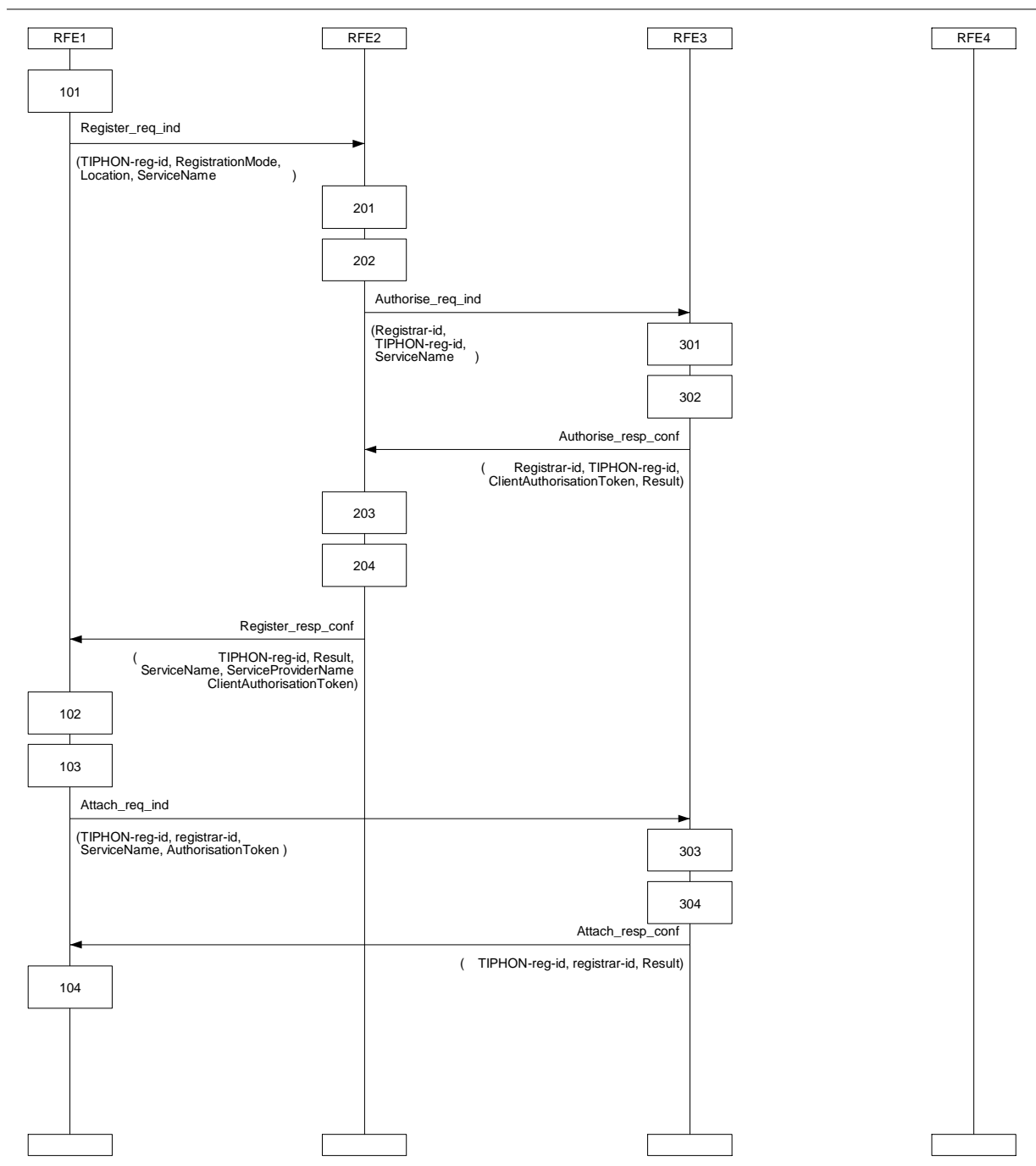


Figure 7: Normal initial registration

5.2.6.1.2 Change of SpoA (location update)

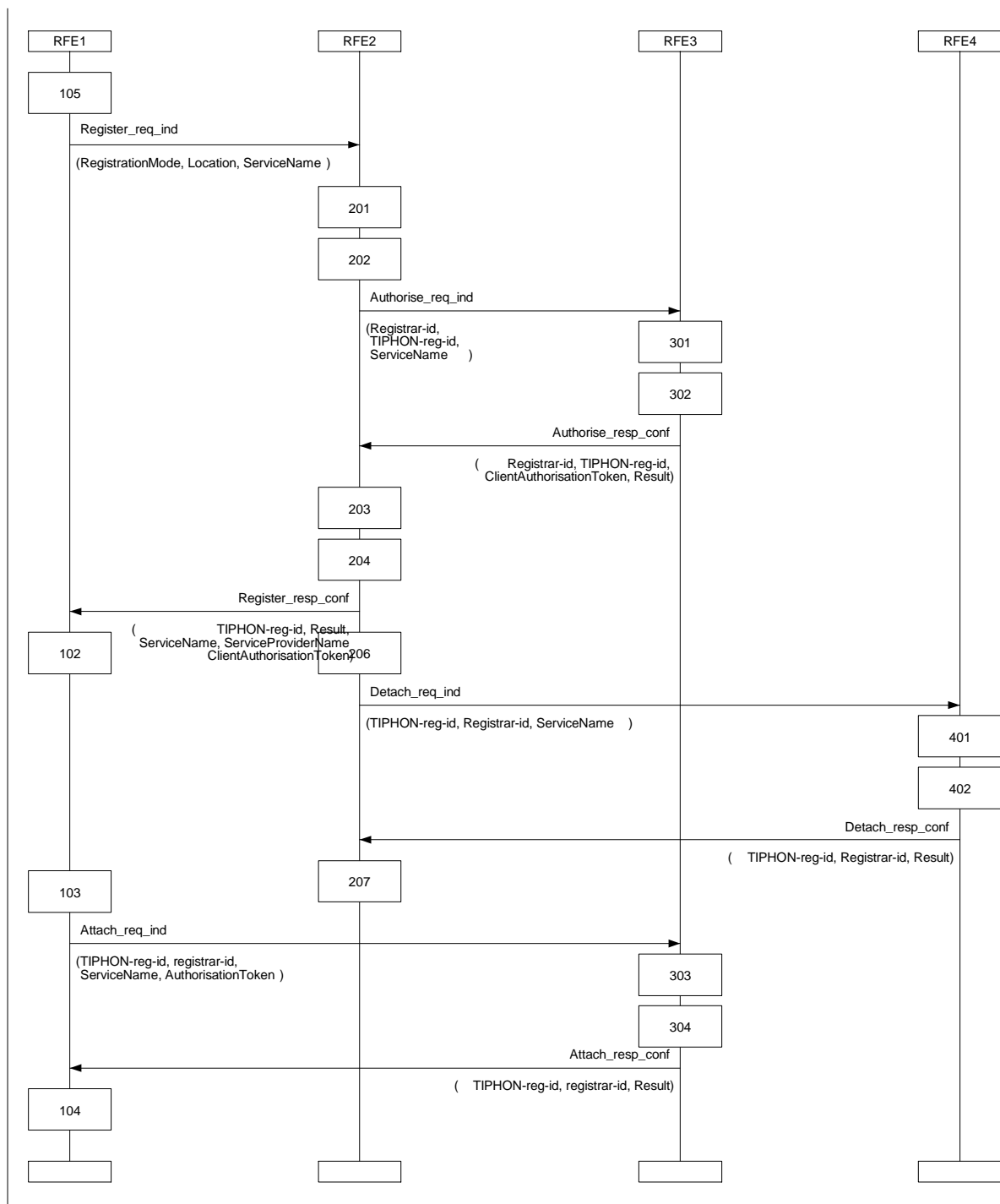


Figure 8: Normal location update (roaming)

5.2.6.1.3 DeRegistration

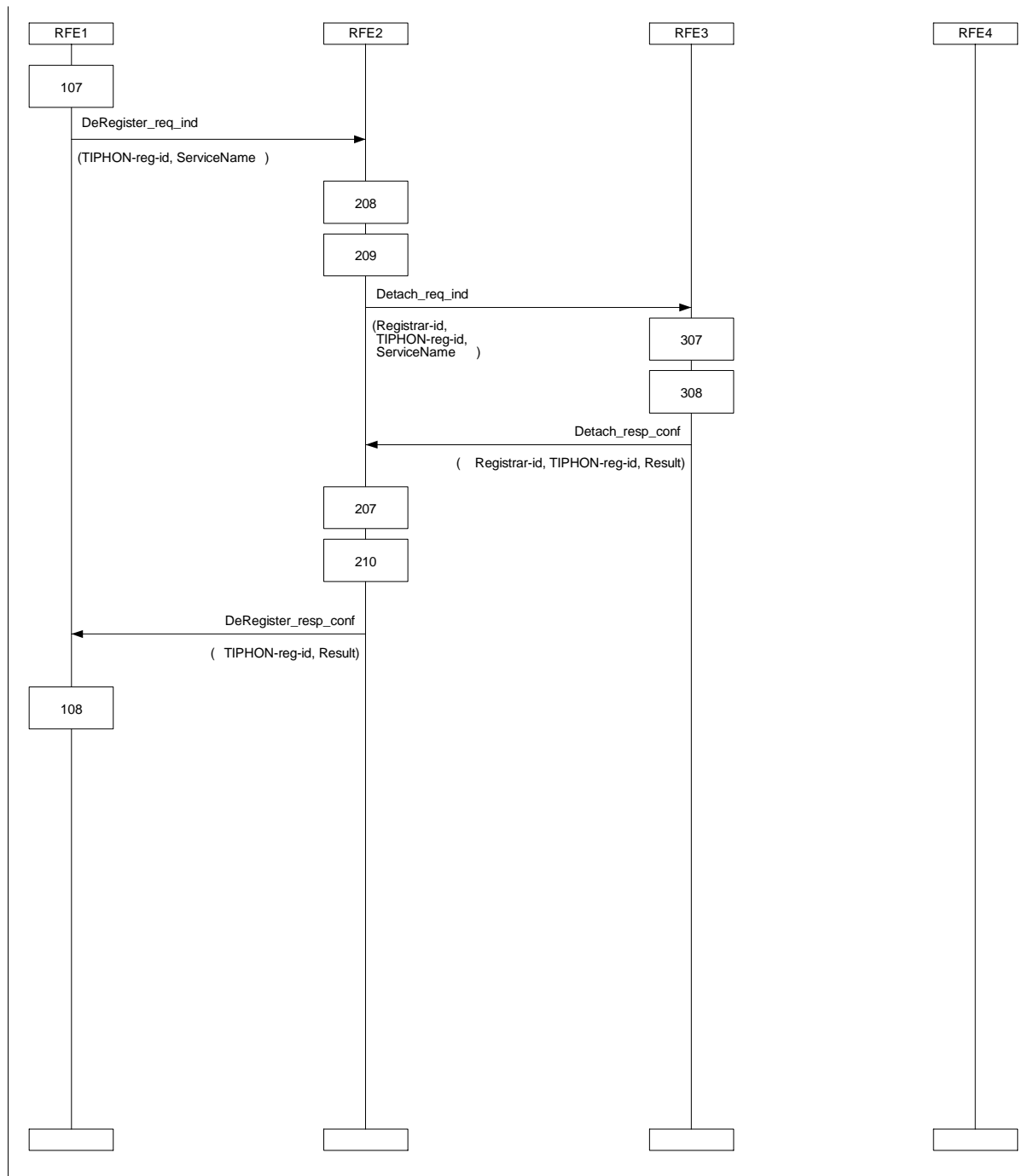


Figure 9: DeRegistration normal behaviour

5.2.6.2 Exceptional operation

5.2.6.2.1 Invalid identity

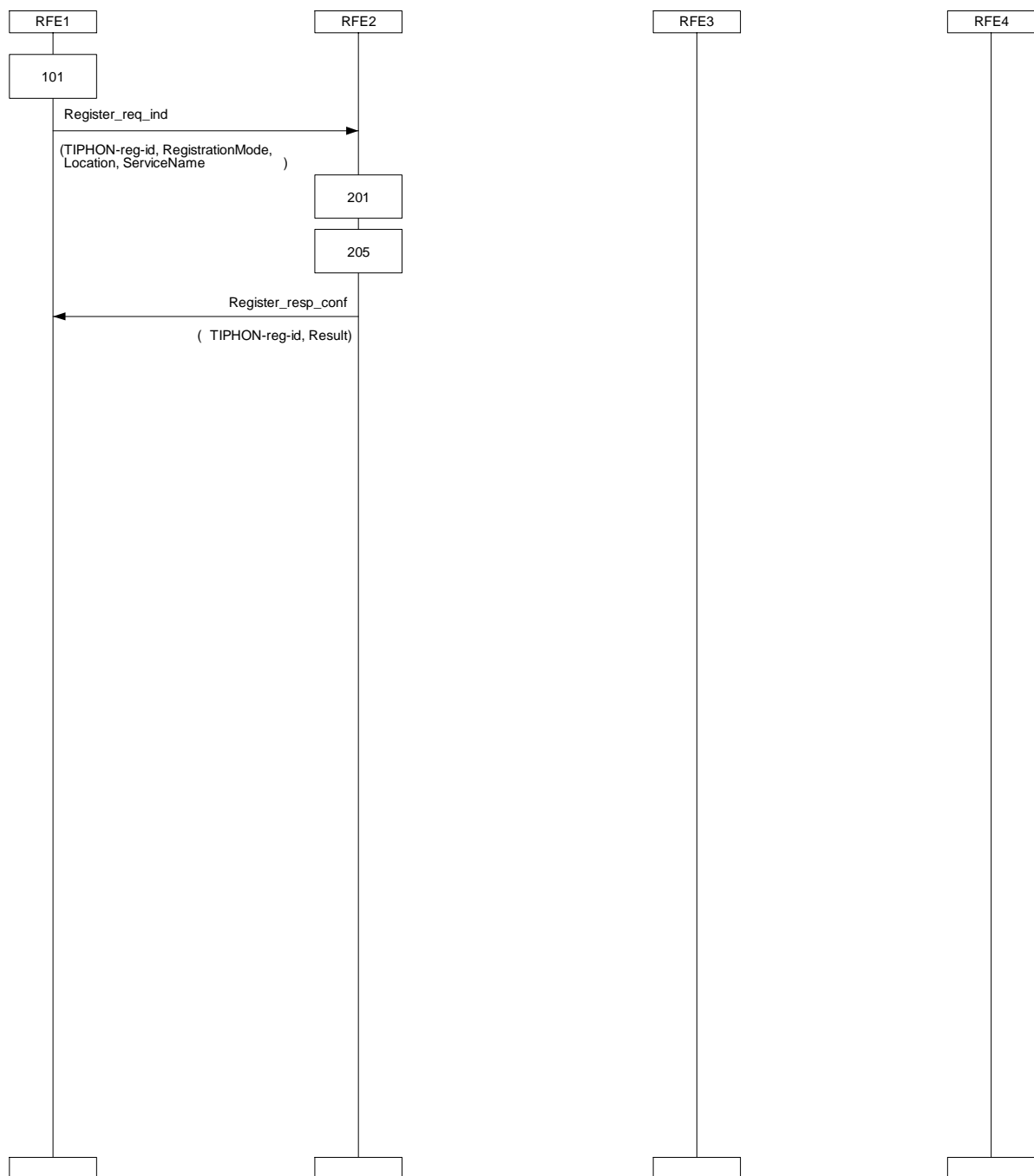


Figure 10: Exceptional behaviour - invalid identity

5.2.6.2.2 Service not available

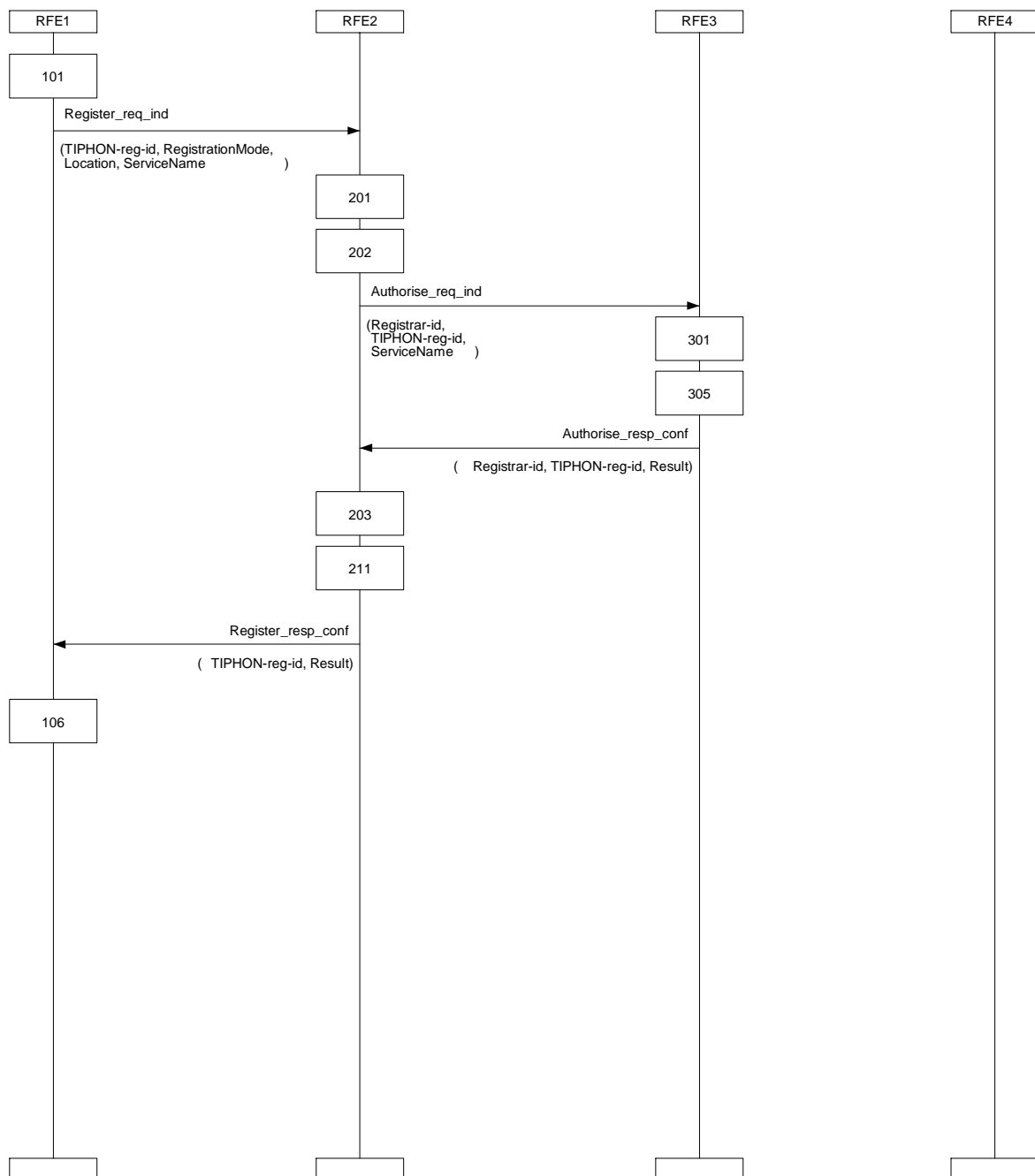


Figure 11: Exceptional behaviour - service not available

5.3 Registration functional entity actions

Throughout the descriptions of FE actions, the following conventions are used to identify information flows:

- an information flow is referred to as a "request" at the FE that sends it and as an "indication" at the FE that receives it, this is shown in the actions of each FE as "req_ind";
- the corresponding confirmation is referred to as a "response" at the FE that sends it and as a "confirmation" at the FE that receives it, this is shown in the actions of each FE as "resp_conf".

The following FE actions shall occur at the points indicated in the information flow sequences of clause 5.2.

5.3.1 Actions of RFE1

- 101: On request RFE1 shall prepare a Register_req_ind information flow setting the RegistrationMode element to "initial registration" and other elements as appropriate (i.e. location, ServiceName, TIPHON-reg-id) and send it to RFE2;
- 102: On receipt of the Register_resp_conf information flow RFE1 shall extract the result element. If result is "Registration successful" RFE1 shall undertake action 103, else is shall undertake action 106;
- 103: From the received Register_resp_conf information flow RFE1 shall prepare an Attach_req_ind information flow containing the received AuthorisationToken and send it to RFE3 (identified by resolving the received ServiceProviderName);
- 104: On receipt of the Attach_resp_conf information flow RFE1 shall check the result element and if the value is "Service attachment successful" shall be able to invoke that service through that SpoA. If the result is not "Service attachment successful" the service shall be barred;
- 105: RFE1 shall prepare a Register_req_ind information flow setting the RegistrationMode element to "Location update" and other elements as appropriate (i.e. location, ServiceName) and send it to RFE2;
- 106: The registration process shall stop, however further attempts to register are possible by invoking action 101;
- 107: On request RFE1 shall prepare a DeRegister_req_ind information flow identifying the service to be deregistered and send it to RFE2;
- 108: On receipt of DeRegister_resp_conf information flow RFE1 shall be inhibited from invoking the deregistered service.

5.3.2 Actions of RFE2

- 201: On receipt of Register_req_ind information flow RFE2 shall validate the received TIPHON-reg-id and if valid shall retrieve the related profile. RFE2 shall also determine the type of registration requested, if InitialRegistration shall follow operation 202, else shall follow operation 205;
- 202: For the service identified in the Register_req_ind information flow RFE2 shall identify an appropriate SpoA and prepare an Authorise_req_ind information flow to send to RFE3;
- 203: On receipt of the Authorise_resp_conf information flow RFE2 shall check the Result. If not "Service authorized to client" then exceptional behaviour action 211;
- 204: RFE2 shall prepare the response to Register_req_ind. If the Authorise_resp_conf is "Service authorized to client" then RFE2 shall set the result element in Register_resp_conf to "RegistrationSuccessful";
- 205: If TIPHON-reg-id is not valid RFE2 shall prepare the Register_resp_conf setting the result element to "Registration-id invalid";
- 206: On acceptance of a new SpoA RFE2 shall revoke the provision of service on the old serving SpoA by sending Detach_req_ind to RFE4;
- 207: On receipt of Detach_resp_conf RFE2 shall update the availability record of the registrant (i.e. update the content of the user-profile);
- 208: On receipt of Deregister_req_ind RFE2 shall validate the received TIPHON-reg-id and if valid shall retrieve the related profile to determine if the service is enabled and to identify the SpoA (RFE3) currently serving that service;
- 209: RFE2 shall withdraw the provision of service on the serving SpoA by sending Detach_req_ind to RFE3;

- 210: RFE2 shall prepare `Deregister_resp_conf`. If the identity was valid and if `Detach_resp_conf` was successful then RFE2 shall set the result element of `Deregister_resp_conf` to "DeRegistrationSuccessful";
- 211: RFE2 shall prepare the `Register_resp_conf` setting the result element to "Service Unavailable" and identifying the service that gave rise to the error.

5.3.3 Actions of RFE3

- 301: On receipt of `Authorise_req_ind` RFE3 shall determine if it can provide resource to allow the user to be offered service;
- 302: If service can be provided RFE3 shall prepare `Authorise_resp_conf` setting result to "Service authorized to client" and adding the `ClientAuthorisationToken` prior to sending to RFE2;
- 303: On receipt of `Attach_req_ind` RFE3 shall validate the TIPHON-reg-id and the authorization token;
- 304: RFE3 shall prepare `Attach_resp_conf` setting result to "Service attachment successful" and allow invocation of the service to the user;
- 305: If service cannot be provided RFE3 shall prepare `Authorise_resp_conf` setting result to "Resource not available" and send it to RFE2;
- 306: If the TIPHON-reg-id is not recognized by RFE3 it shall prepare `Attach_resp_conf` setting result to "Identity not recognized". If TIPHON-reg-id is recognized but the `AuthorisationToken` is no longer valid RFE3 shall prepare `Attach_resp_conf` setting result to "Authorization expired";
- 307: On receipt of `Detach_req_ind` RFE3 shall validate the received data and cease to offer service to the user and set result to "Service detachment successful". If validation fails the Result shall be set to "Identity not recognized";
- 308: RFE3 shall prepare `Detach_resp_conf` with the result set from action 307 and send it to RFE2.

5.3.4 Actions of RFE4

- 401: On receipt of `Detach_req_ind` RFE4 shall validate the received data and cease to offer service to the user and set result to "Service detachment successful". If validation fails the Result shall be set to "Identity not recognized";
- 402: RFE4 shall prepare the `Detach_resp_conf` with the result set from action 401 and send it to RFE2.

5.4 Registration functional entity behaviour

NOTE: In the SDL process diagrams labels, where used, correspond to the numbered actions of each FE as given in clause 5.3.

5.4.1 Behaviour of RFE1

The behaviour of RFE1 is shown in the SDL process diagram in figures 12 through 16.

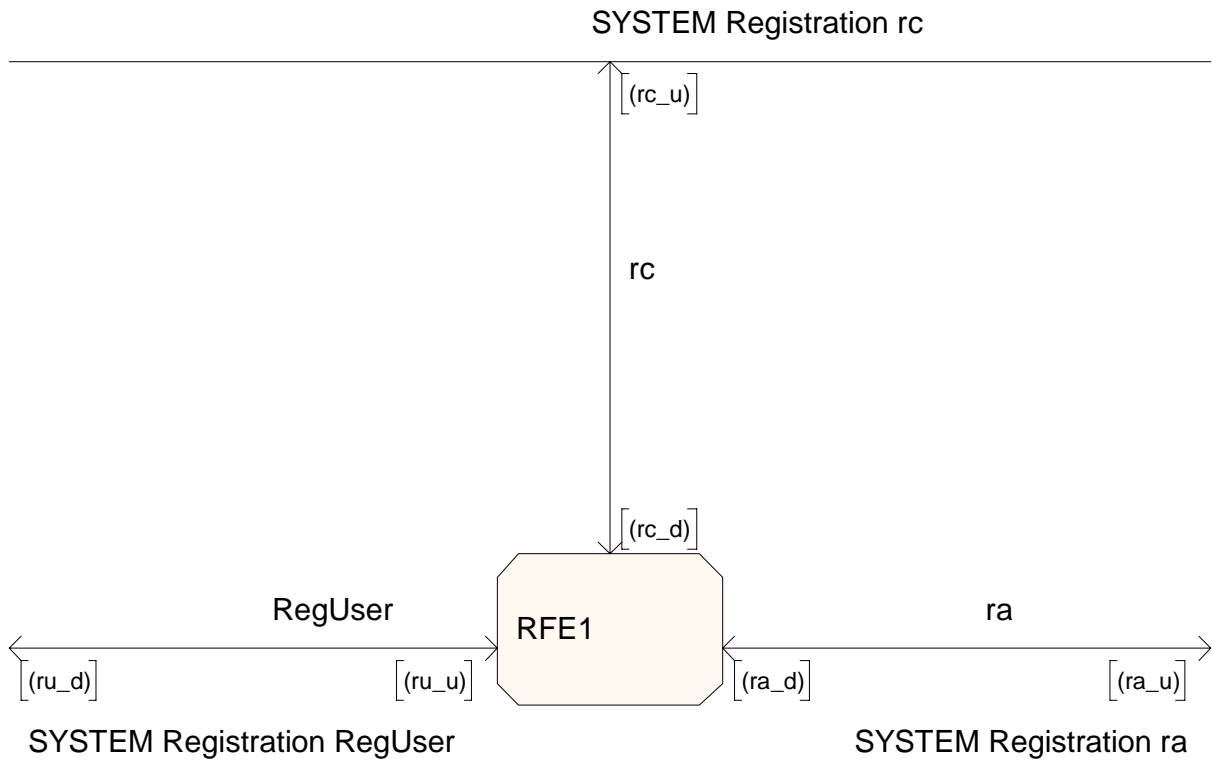


Figure 12: Block RFE1

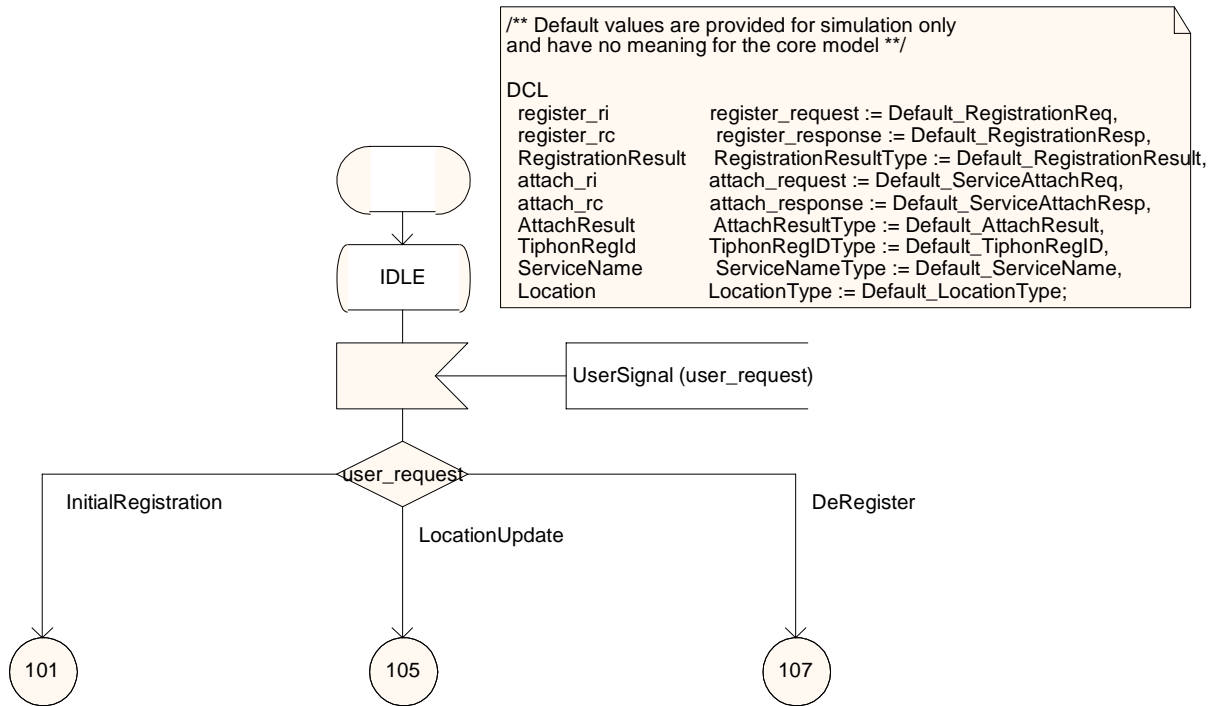


Figure 13: Process RFE1, 1 of 4

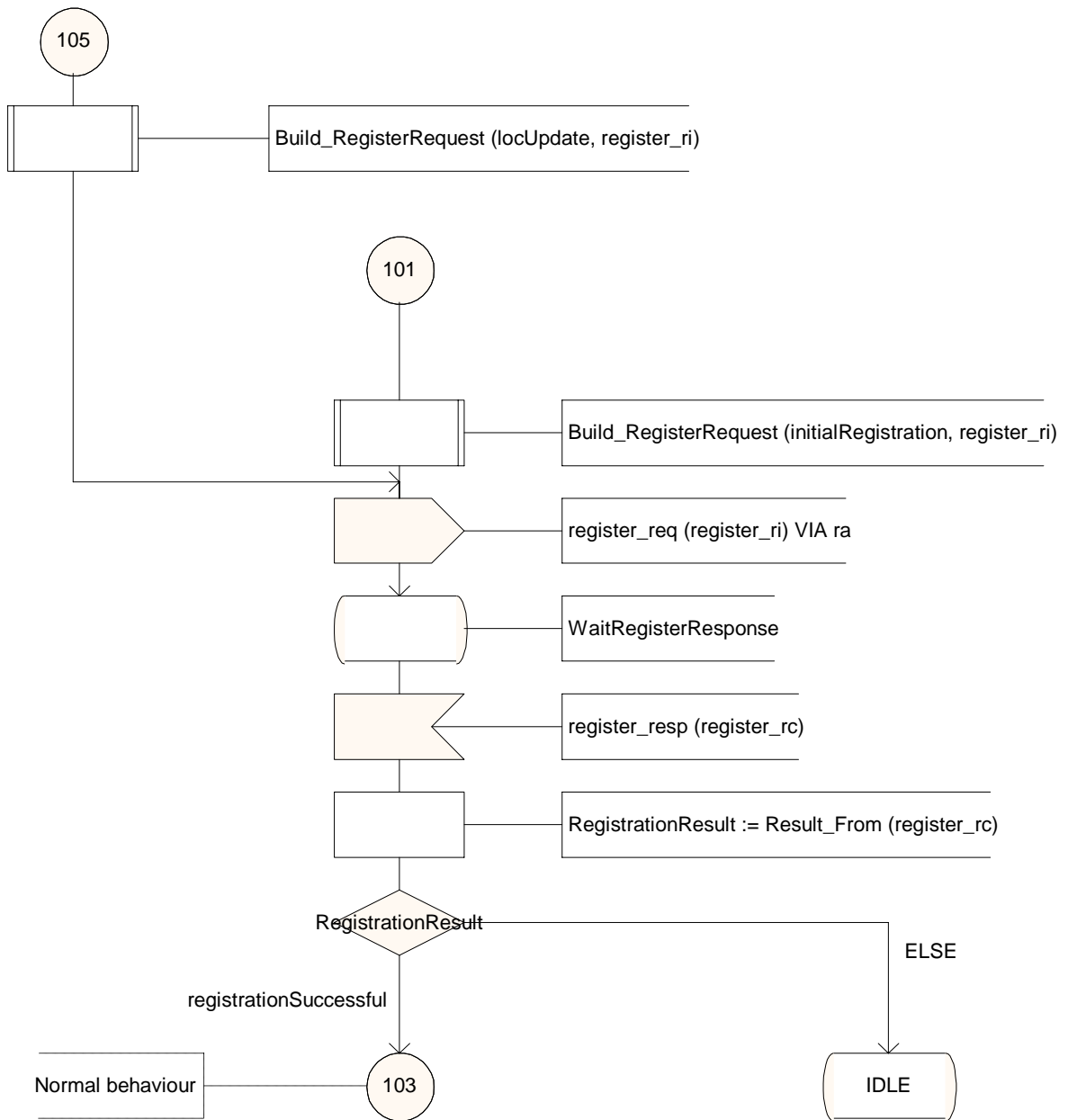


Figure 14: Process RFE1, 2 of 4

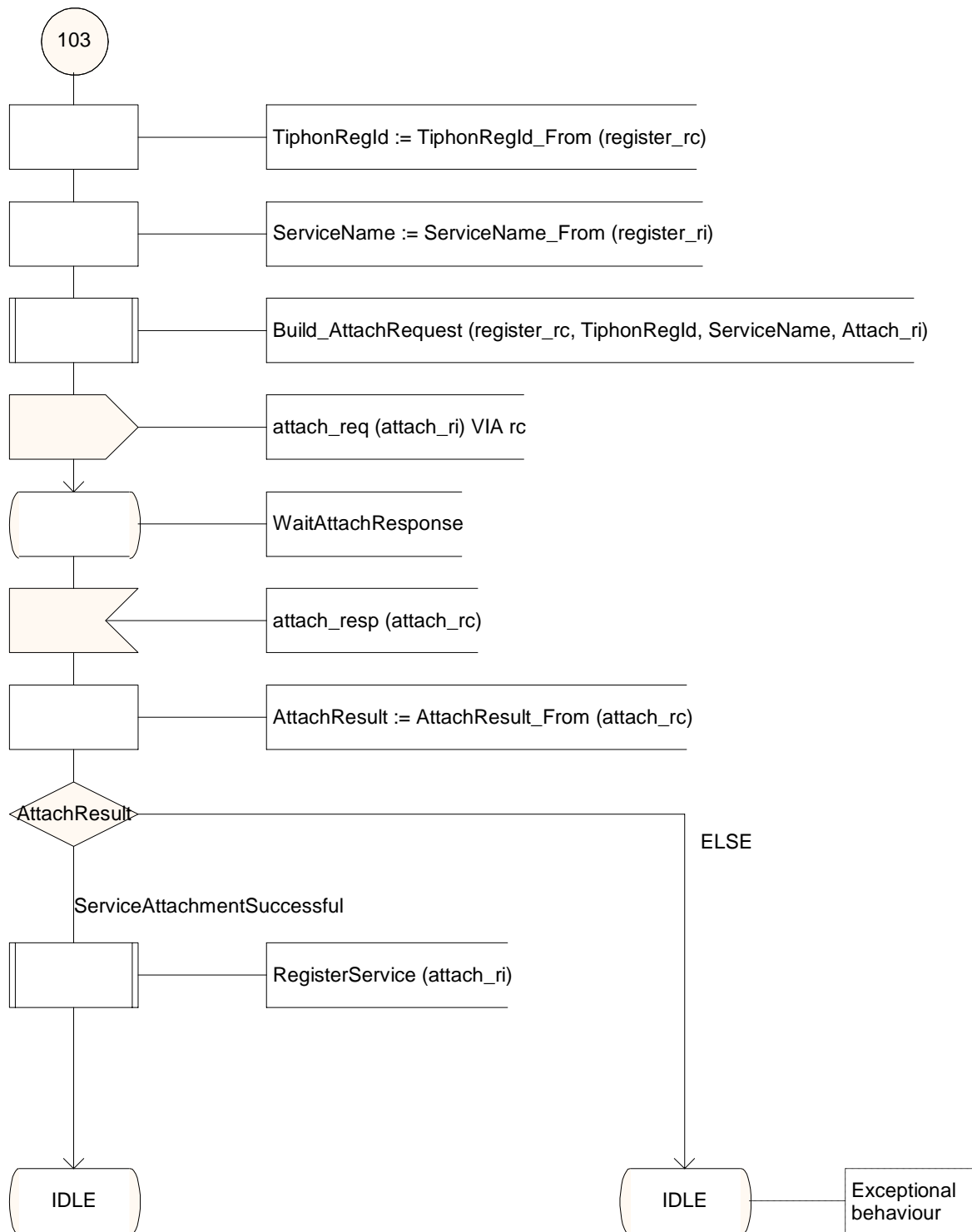


Figure 15: Process RFE1, 3 of 4

5.4.2 Behaviour of RFE2

The behaviour of RFE2 is shown in the SDL process diagram in figures 17 through 23.

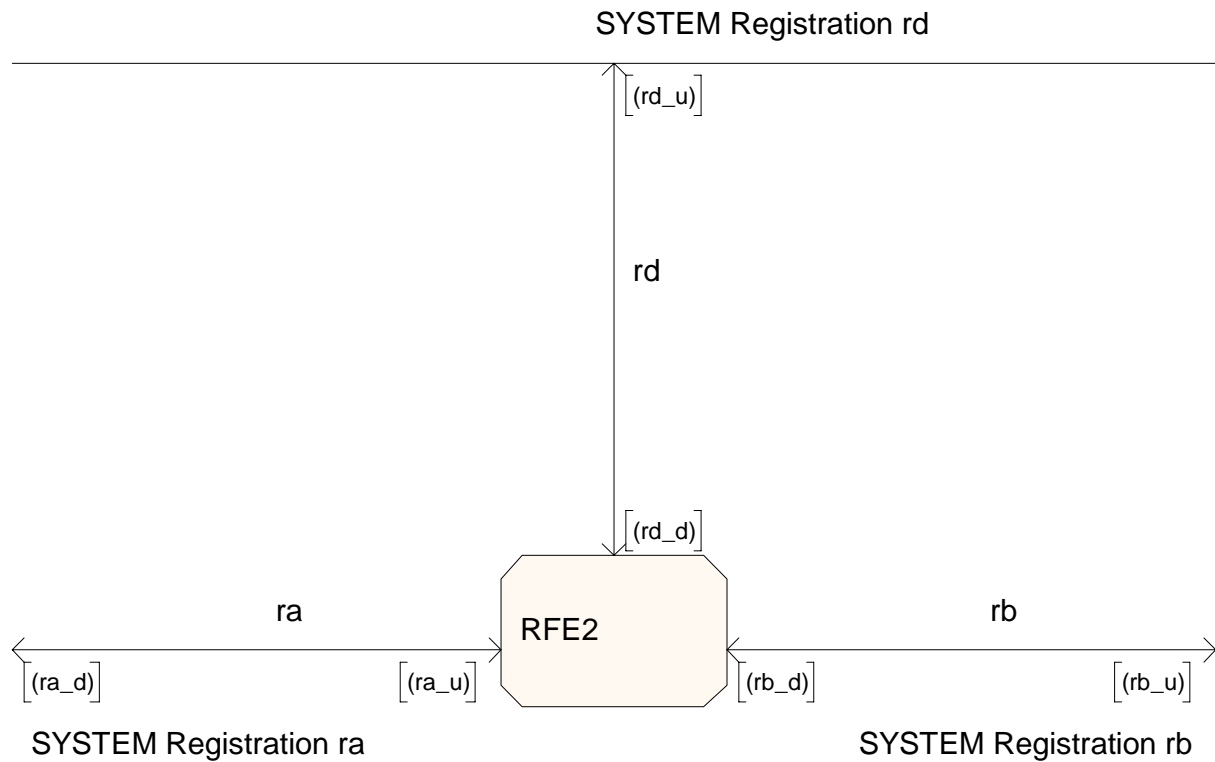


Figure 17: Block RFE2, 1 of 1

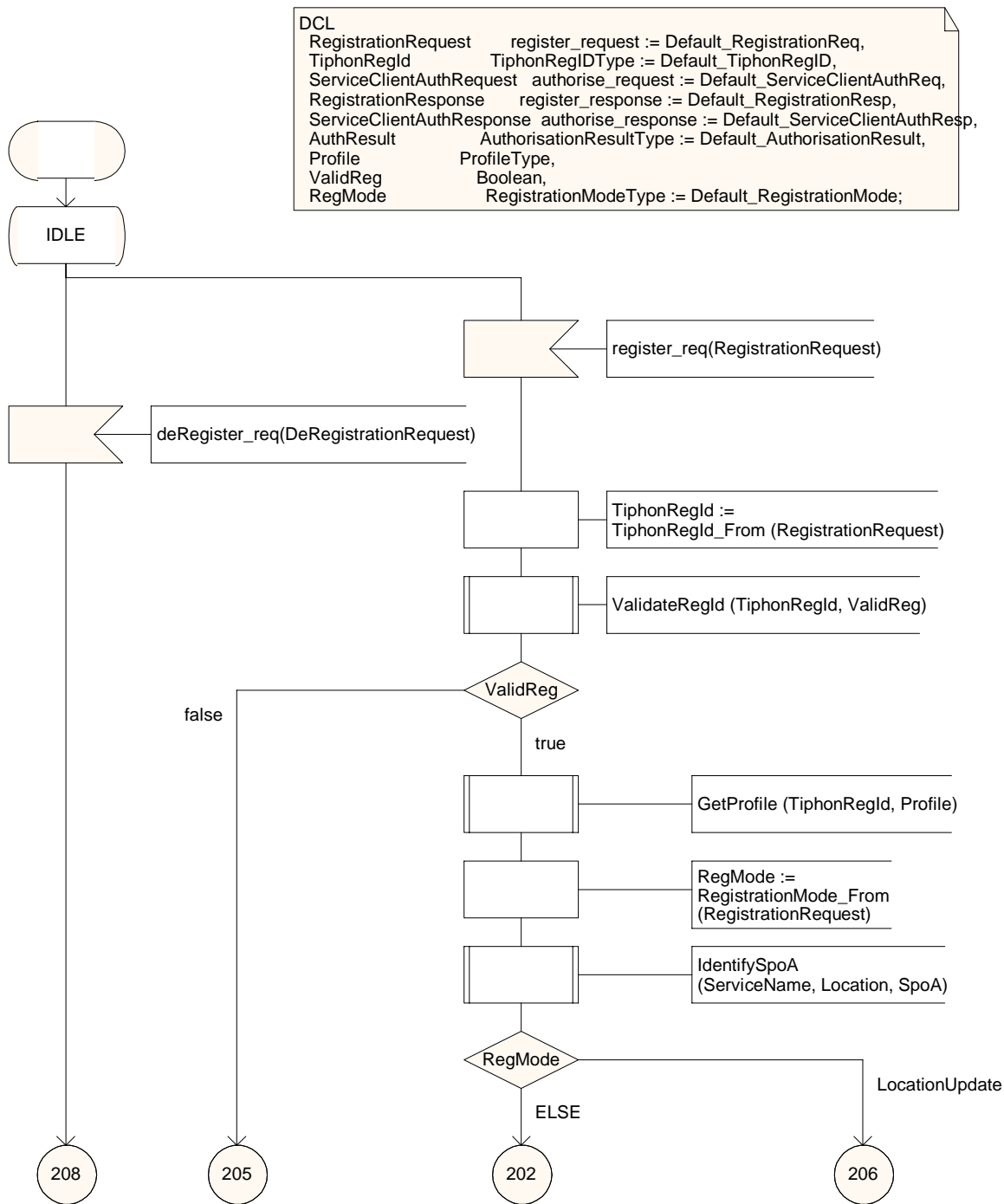


Figure 18: Process RFE2, 1 of 6

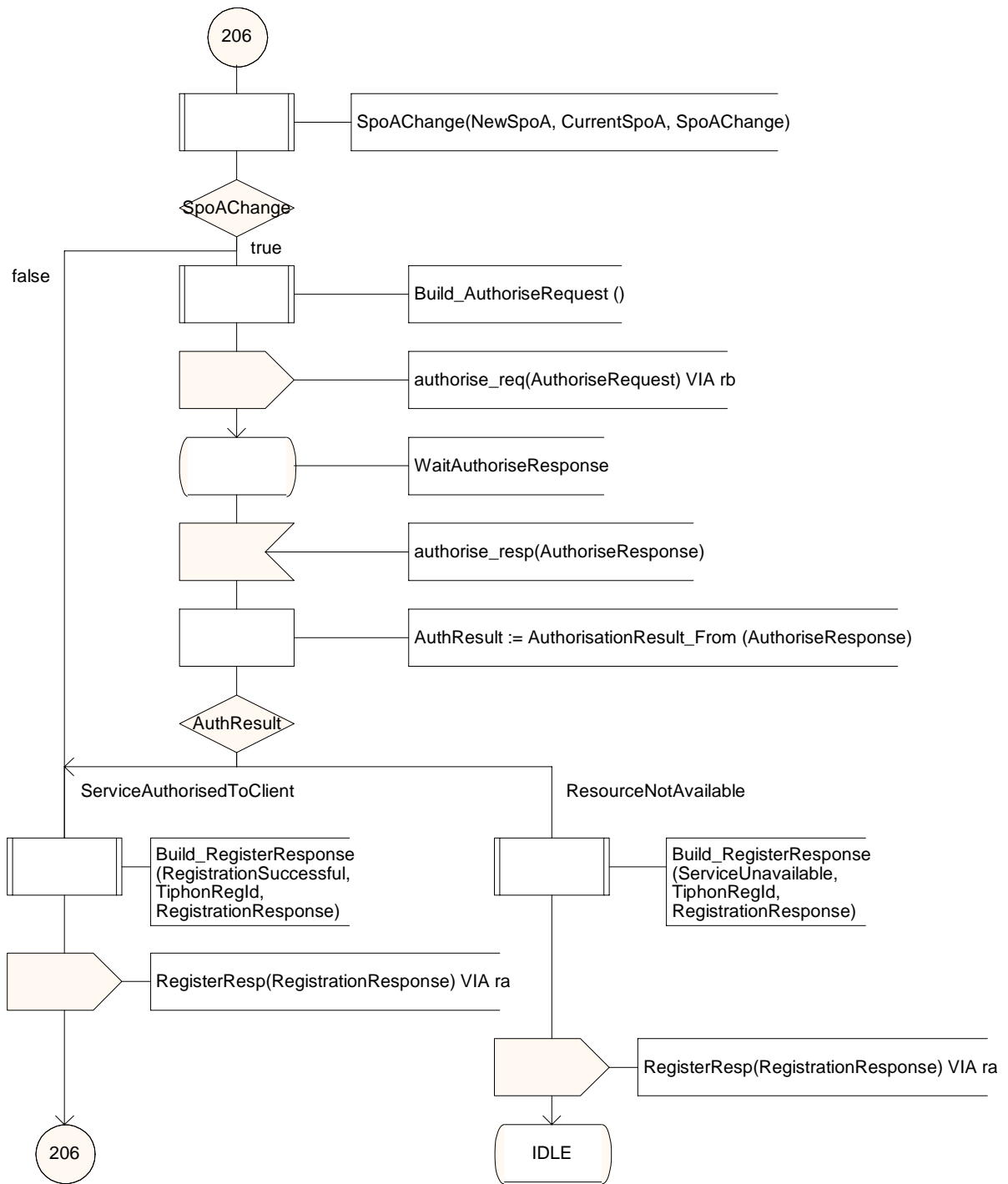


Figure 20: Process RFE2, 3 of 6

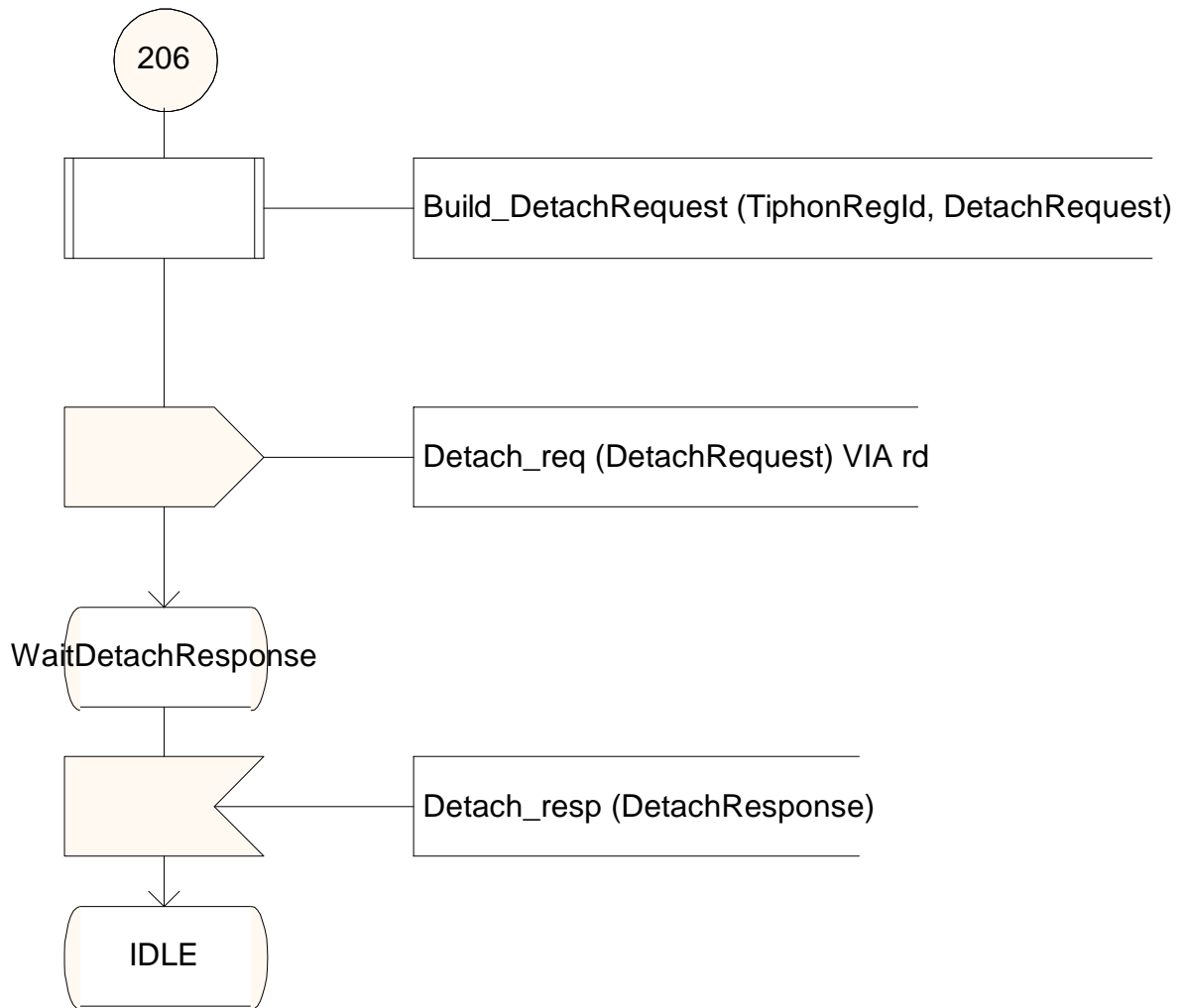


Figure 21: Process RFE2, 4 of 6


```
/**  
This process model shows the exceptional behaviour when the Identity of the registrant is invalid  
**/
```



Figure 22: Process RFE2, 5 of 6

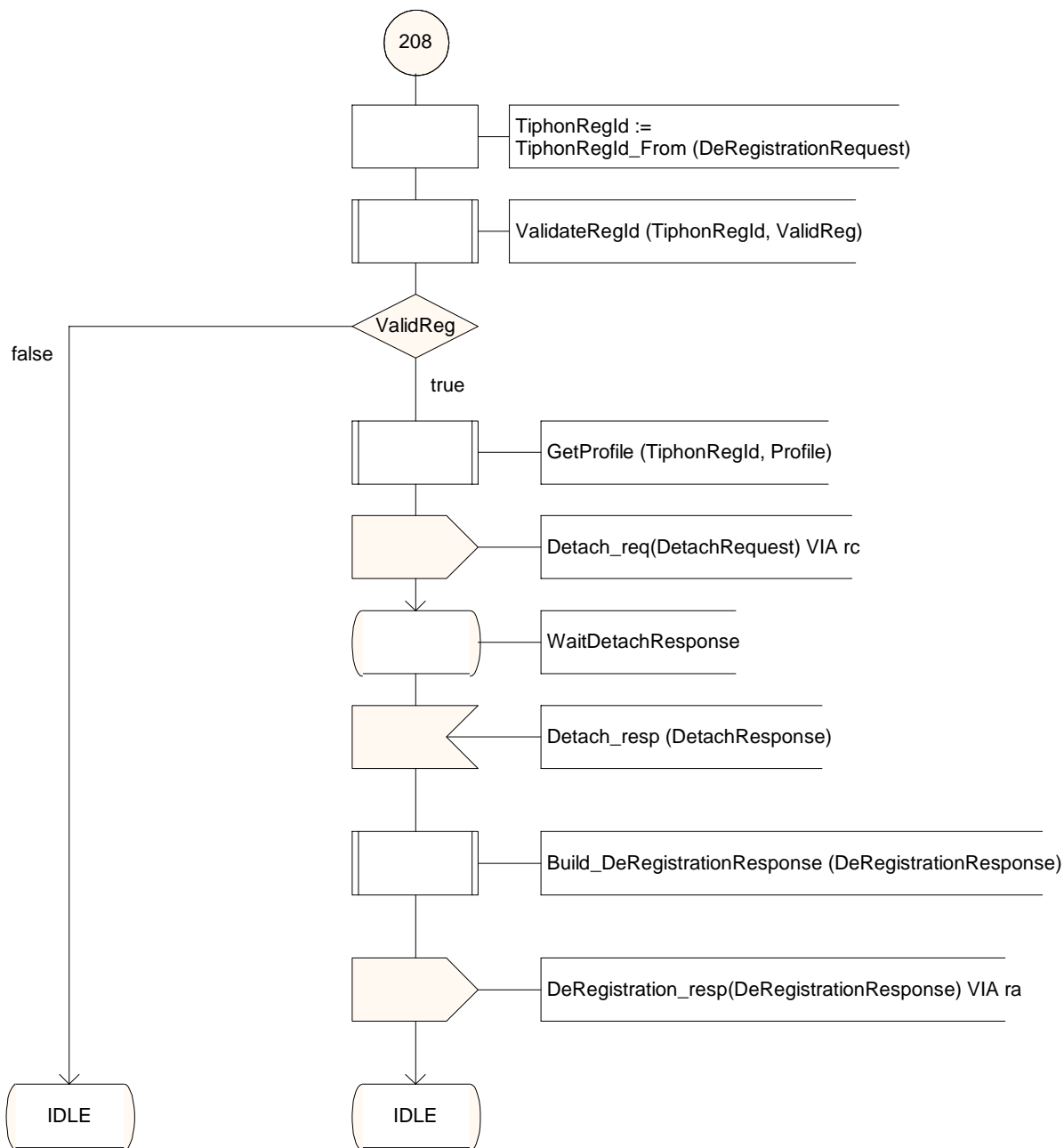


Figure 23: Process RFE2, 6 of 6

5.4.3 Behaviour of RFE3

The behaviour of RFE3 is shown in the SDL process diagram in figures 24 through 21.

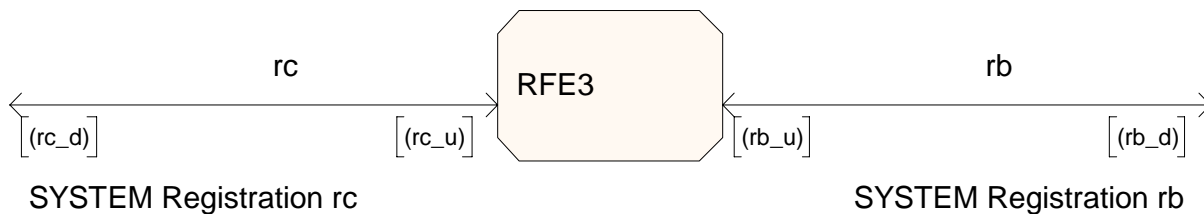


Figure 24: Block RFE3, 1 of 1

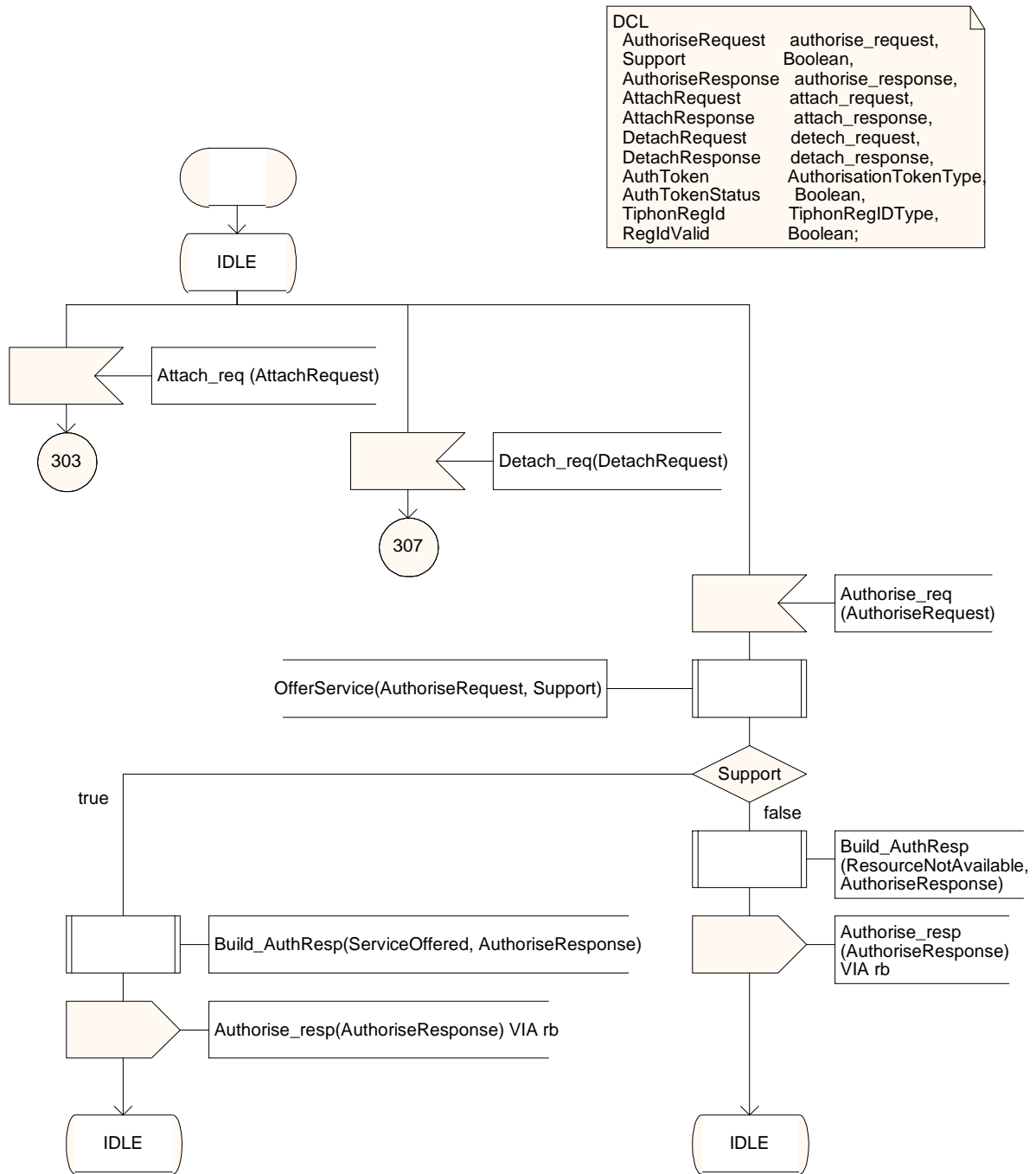


Figure 25: Process RFE3, 1 of 3

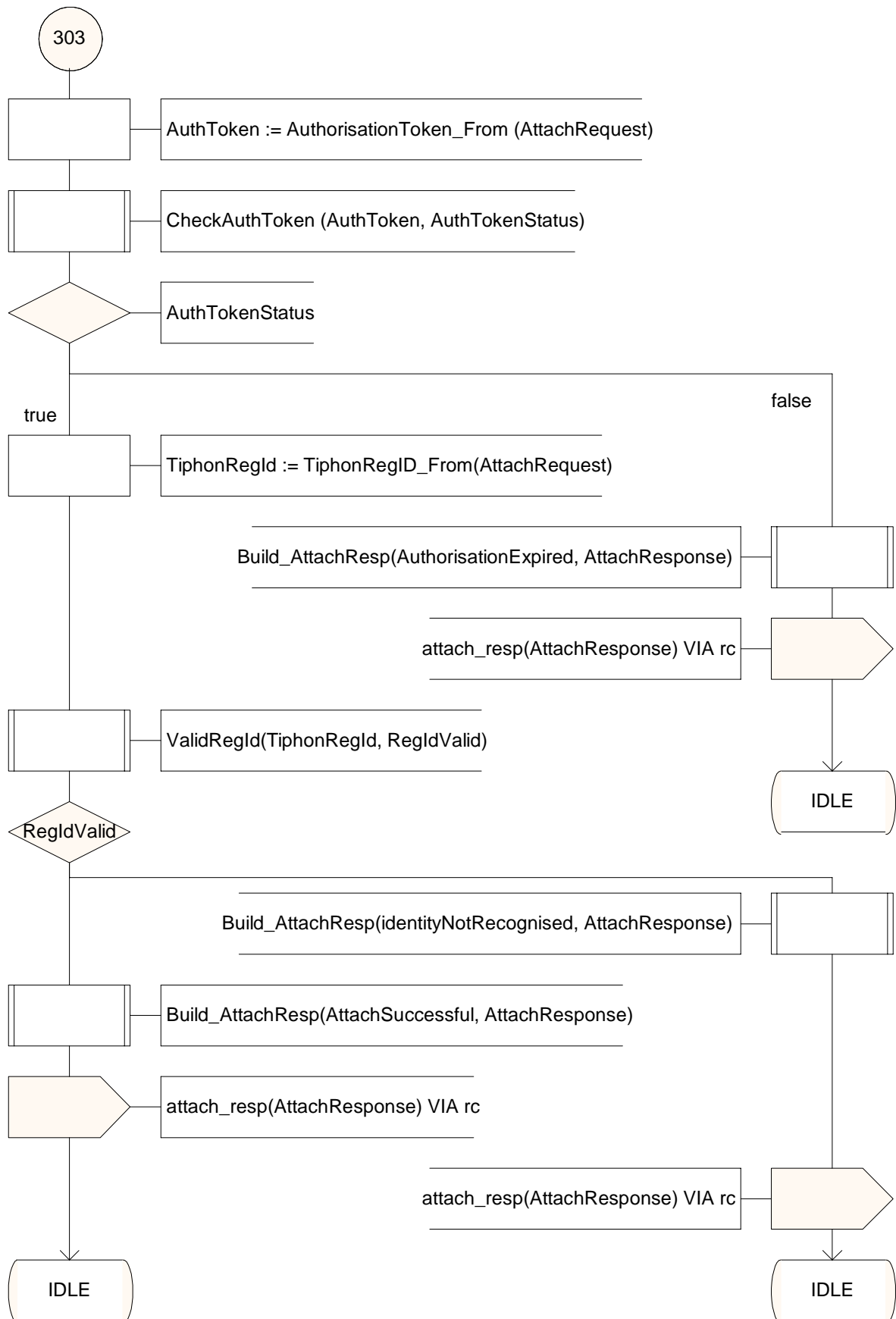


Figure 26: Process RFE3, 2 of 3

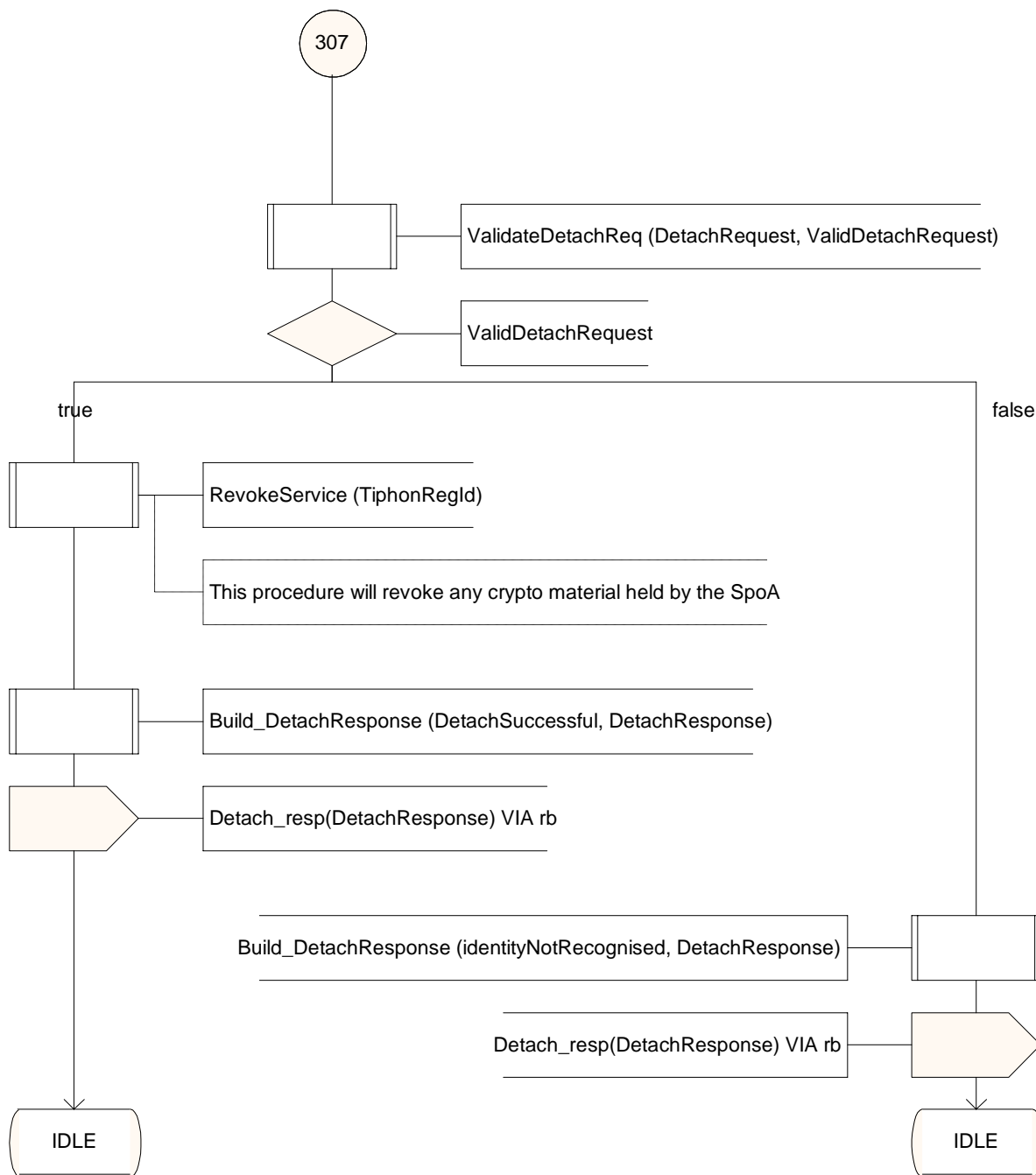


Figure 27: Process RFE3, 3 of 3

5.4.4 Behaviour of RFE4

The behaviour of RFE4 is shown in the SDL process diagram in figures 28 and 29.



Figure 28: Block RFE4, 1 of 1

DCL
ValidDetachRequest boolean;

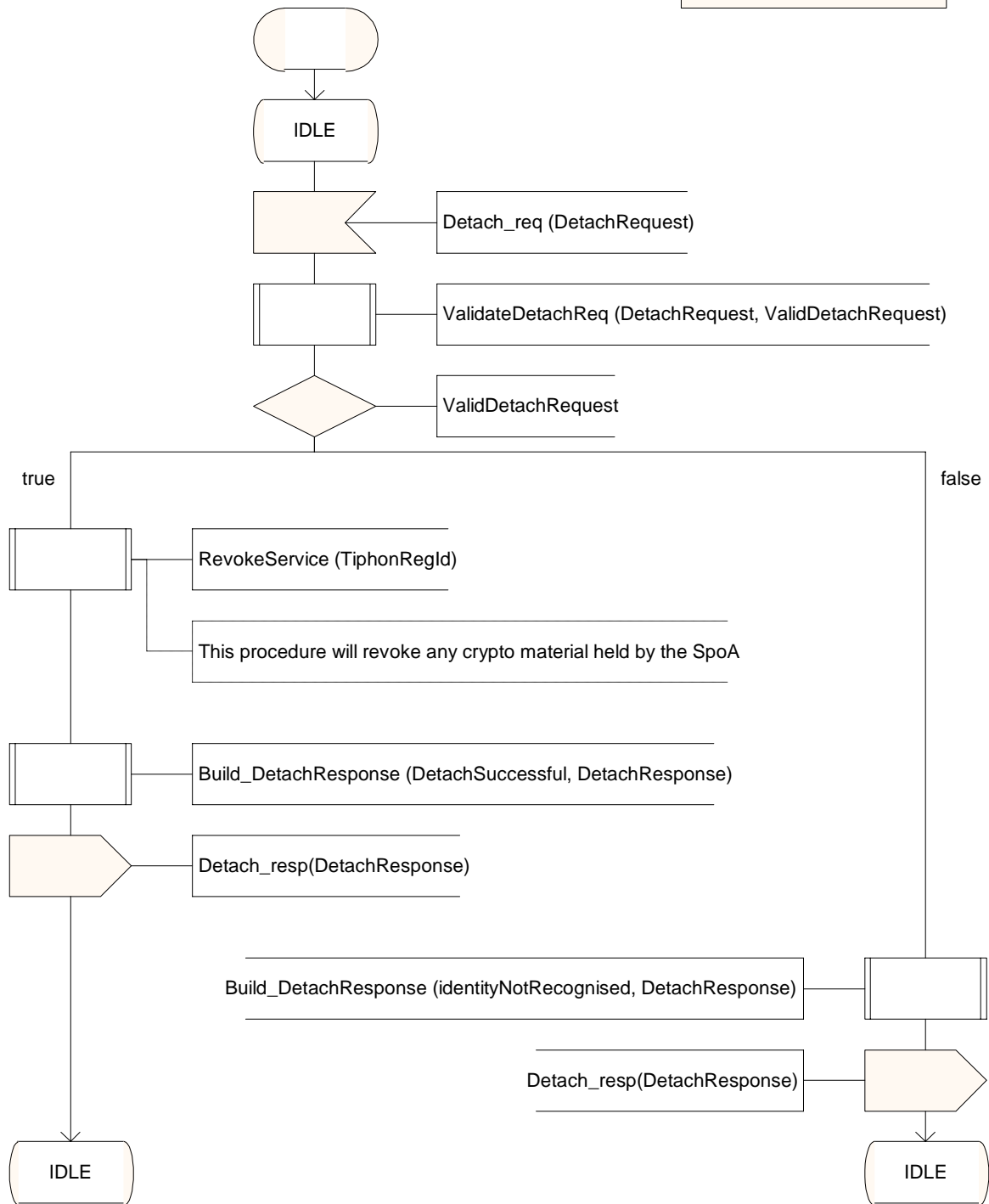


Figure 29: Process RFE4, 1 of 1

5.5 ASN.1 data definitions

```

-- Registration information element ASN.1 data types --

TiphonRegIDType ::= Visiblestring

RegistrationModeType ::= ENUMERATED
{
  initialRegistration (0),
  locationUpdate (1)
}

ServiceNameType ::= Visiblestring

RegistrationResultType ::= ENUMERATED
{
  registrationSuccessful,
  registrationIDInvalid,
  serviceUnavailable
}

ServiceProviderNameType ::= Visiblestring

AuthorisationTokenType ::= Visiblestring

DeregistrationResultType ::= ENUMERATED
{
  deRegistrationSuccessful,
  RegistrationIDInvalid
}

RegistrarIDType ::= Visiblestring

AuthorisationResultType ::= ENUMERATED
{
  serviceAuthorisedToClient,
  resourceNotAvailable
}

AttachResultType ::= ENUMERATED
{
  serviceAttachmentSuccessful,
  identityNotRecognised,
  authorisationExpired
}

RevokeResultType ::= ENUMERATED
{
  serviceDetachmentSuccessful,
  identityNotRecognised
}

TRL ::= SEQUENCE
{
  protocolID      Visiblestring,
  nameorAddress  Visiblestring,
  port            Integer OPTIONAL
}

-- Data structures for registration signals --

Register_req_ind_type ::= SEQUENCE
{
  tiphonRegID          TiphonRegIDType,
  registrationMode     RegistrationModeType,
  location             TRL,
  serviceName         ServiceNameType
}

Register_resp_conf_type ::= SEQUENCE
{
  tiphonRegID          TiphonRegIDType,
  serviceName         ServiceNameType OPTIONAL,
  result              RegistrationResultType,
  serviceProviderName ServiceProviderNameType OPTIONAL,
  clientAuthorisationToken AuthorisationTokenType OPTIONAL
}

```

```
deregister_req_ind_type ::= SEQUENCE
{
    tiphonRegID      TiphonRegIDType,
    serviceName      ServiceNameType
}

deregister_resp_conf_type ::= SEQUENCE
{
    tiphonRegID      TiphonRegIDType,
    result            DeregistrationResultType
}

authorise_req_ind_type ::= SEQUENCE
{
    registrarID      RegistrarIDType,
    tiphonRegID      TiphonRegIDType,
    serviceName      ServiceNameType
}

authorise_resp_conf_type ::= SEQUENCE
{
    registrarID      RegistrarIDType,
    tiphonRegID      TiphonRegIDType,
    clientAuthorisationToken AuthorisationTokenType OPTIONAL,
    authorisationResult AuthorisationResultType
}

detach_req_ind_type ::= SEQUENCE
{
    registrarID      RegistrarIDType,
    tiphonRegID      TiphonRegIDType,
    serviceName      ServiceNameType
}

detach_resp_conf_type ::= SEQUENCE
{
    registrarID      RegistrarIDType,
    tiphonRegID      TiphonRegIDType,
    result            RevokeResultType
}

attach_req_ind_type ::= SEQUENCE
{
    registrarID      RegistrarIDType,
    tiphonRegID      TiphonRegIDType,
    serviceName      ServiceNameType,
    authorisationToken AuthorisationTokenType
}

attach_resp_conf_type ::= SEQUENCE
{
    registrarID      RegistrarIDType,
    tiphonRegID      TiphonRegIDType,
    attachResult      AttachResultType
}
```


5.6 Allocation of functional entities to domains

The possible allocation of FEs to TIPHON is shown in table 8.

Table 8: Allocation of FEs to TIPHON domains

| Scenario | RFE1 | RFE2 | RFE3 | RFE4 |
|----------|----------|-----------------------|--------------------------|--------------------------|
| 1 | Terminal | Home Service provider | Home Service provider | - |
| 2 | Terminal | Home Service provider | Visited Service provider | - |
| 3 | Terminal | Home Service provider | Visited Service provider | Home Service provider |
| 4 | Terminal | Home Service provider | Home Service provider | Visited Service provider |
| 5 | Terminal | Home Service provider | Visited Service provider | Visited Service provider |
| | | | | |

Scenario 1 is initial registration at home.
 Scenario 2 is initial registration at a visited network.
 Scenario 3 is location update on roaming to a new SpoA in a visited network (moving from home to not-home).
 Scenario 4 is location update on roaming to a new SpoA in the home network (moving home from not-home).
 Scenario 5 is location update between on roaming between two visited SpoAs.

Annex A (informative): Additional SDL used in modelling

The SDL procedures identified here have been developed in the modelling exercise to allow validation and simulation of the process diagrams shown in clause 5.

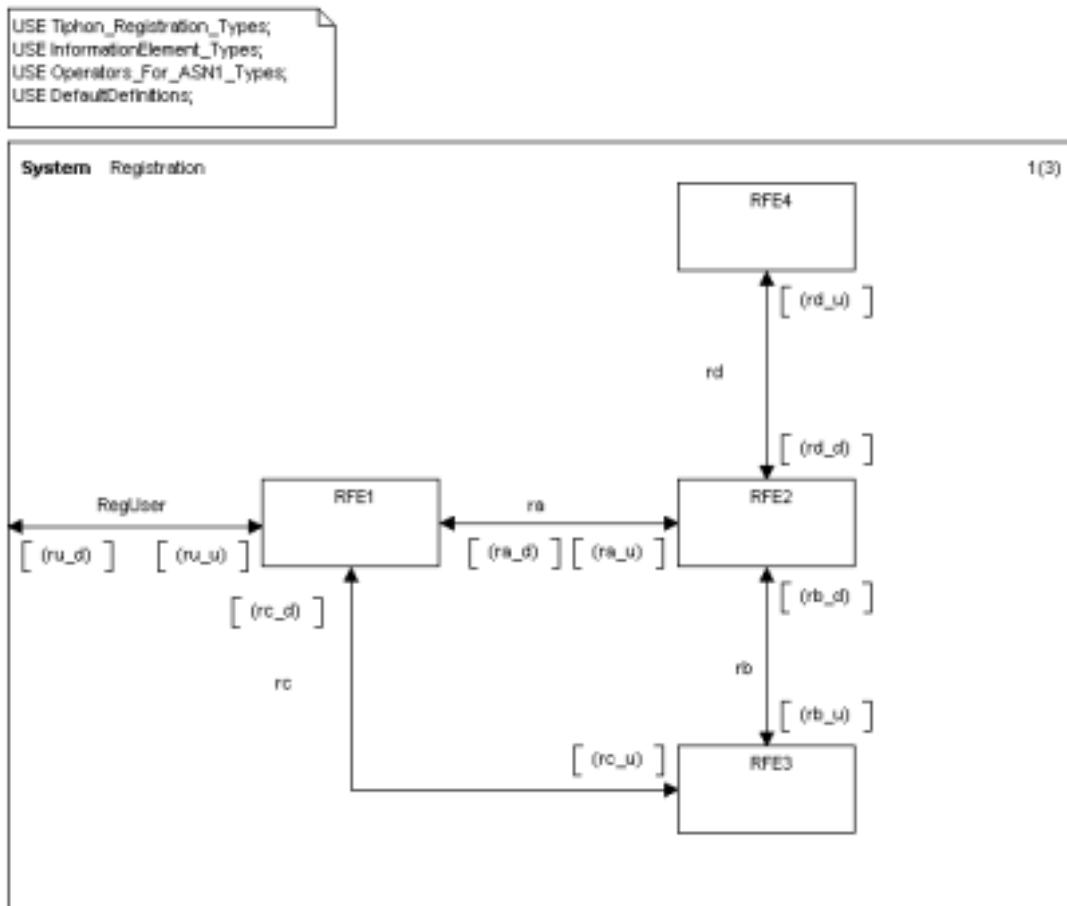


Figure A.1: System Registration, 1 of 3

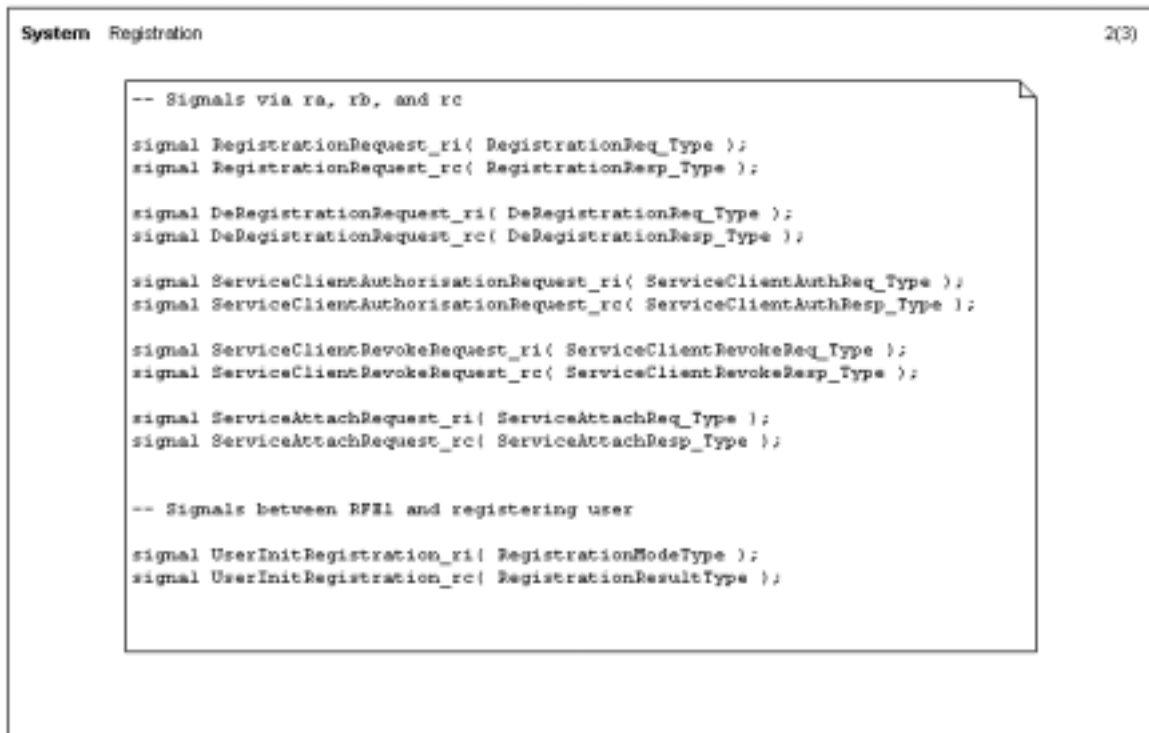


Figure A.2: System Registration 2 of 3

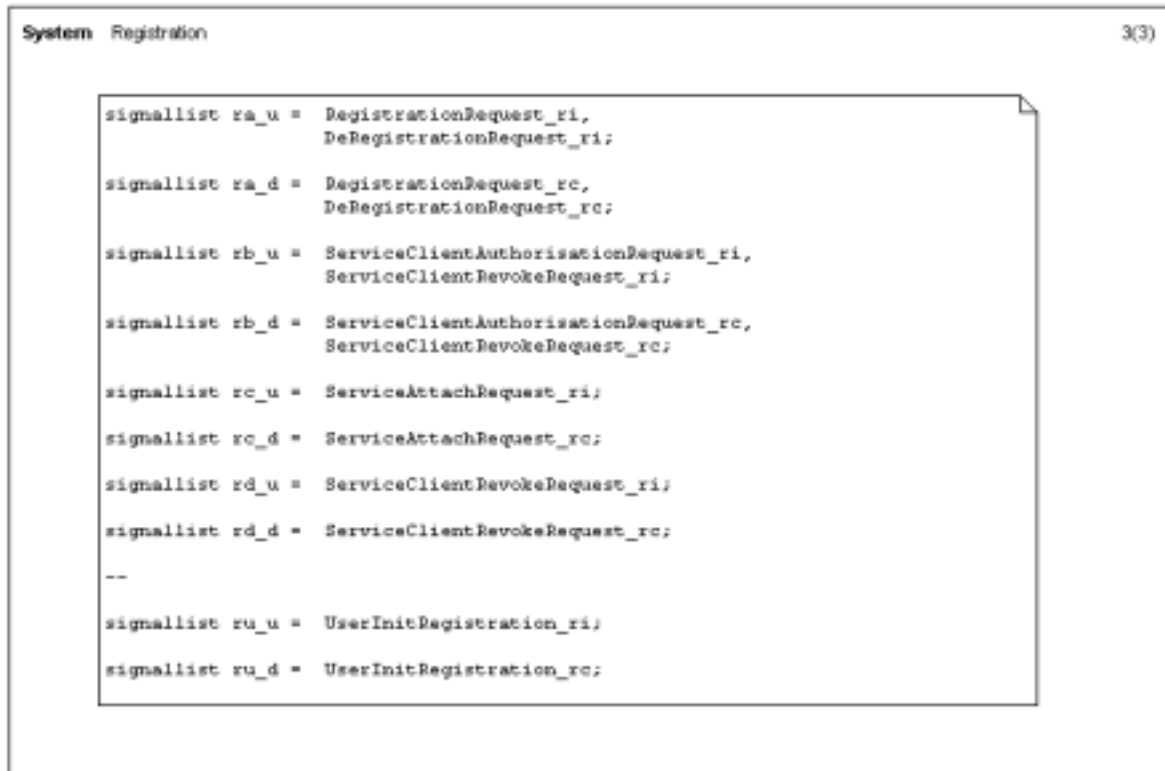


Figure A.3: System Registration 3 of 3

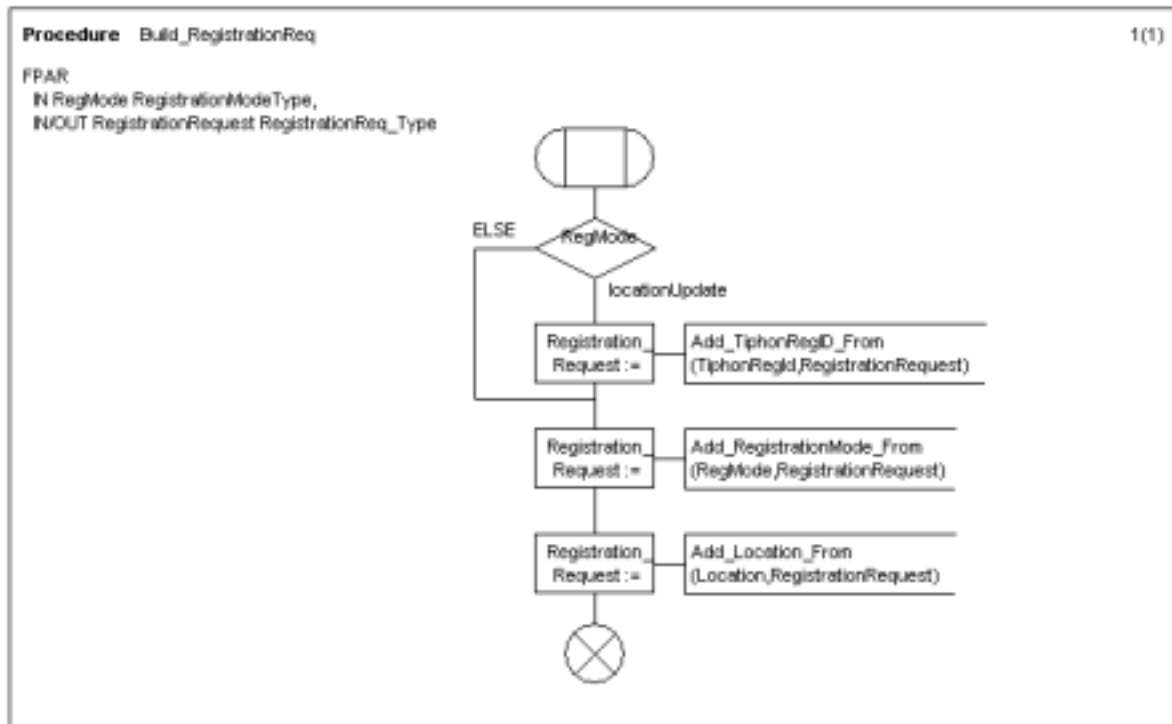


Figure A.4: Procedure Build_RegistrationReq

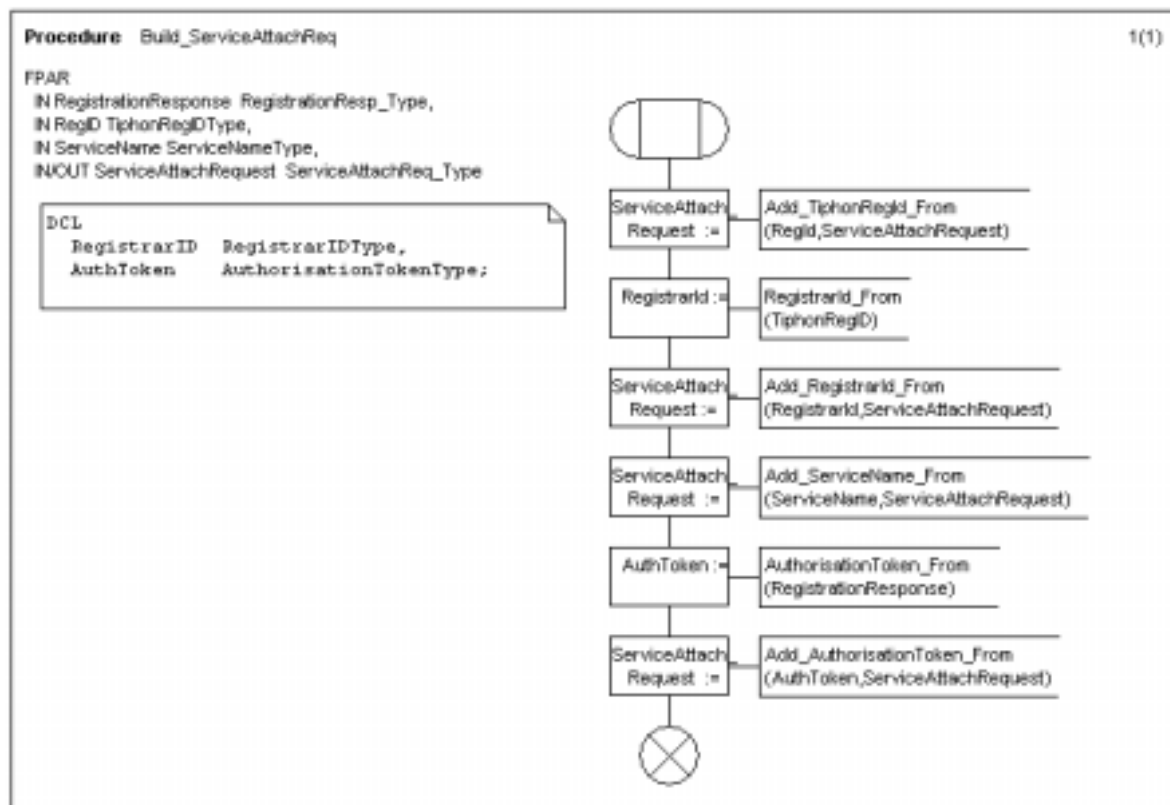


Figure A.5: Procedure Build_ServiceAttachReq

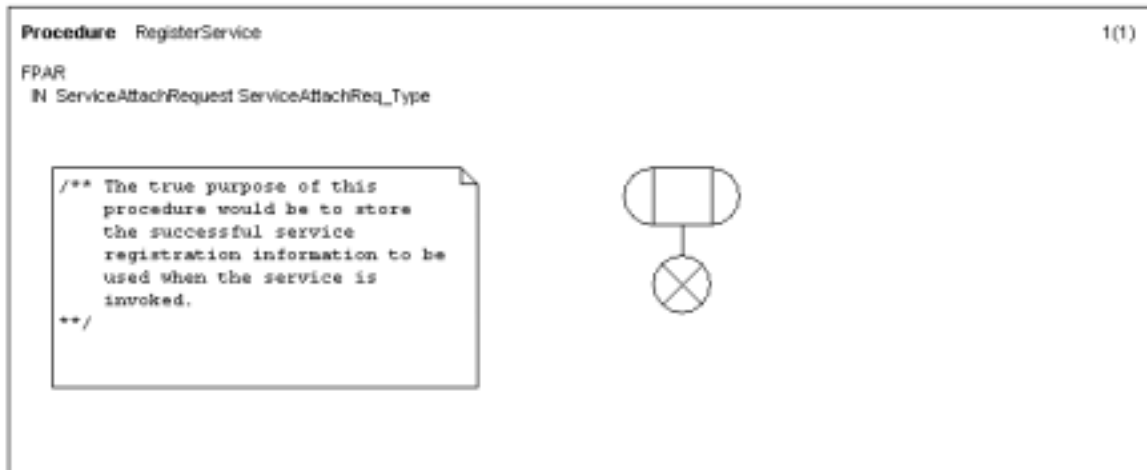


Figure A.6: Procedure RegisterService

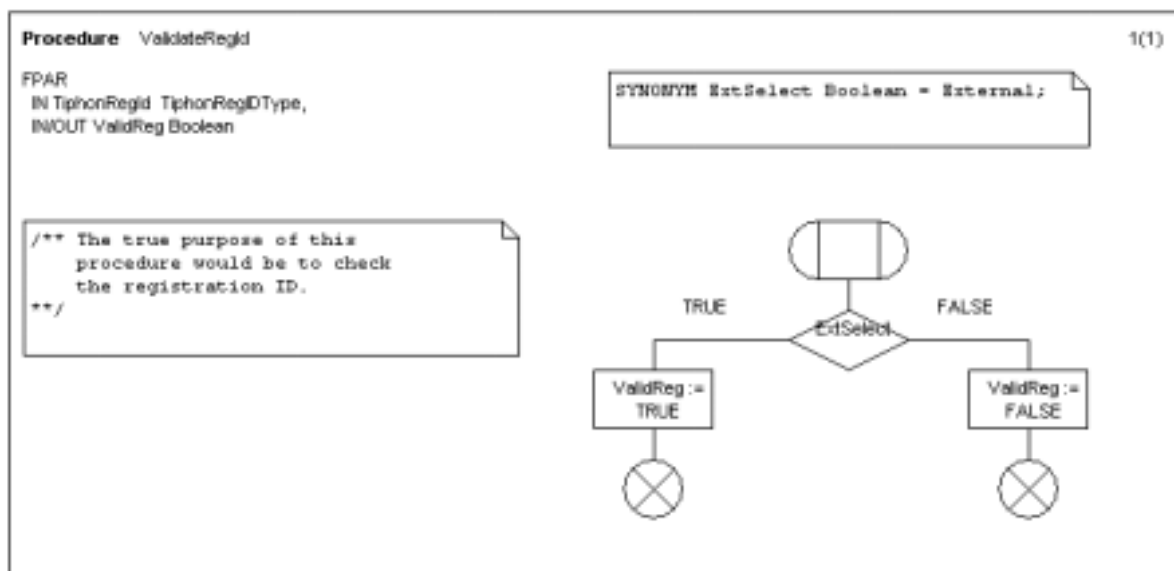


Figure A.7: Procedure ValidateRegID

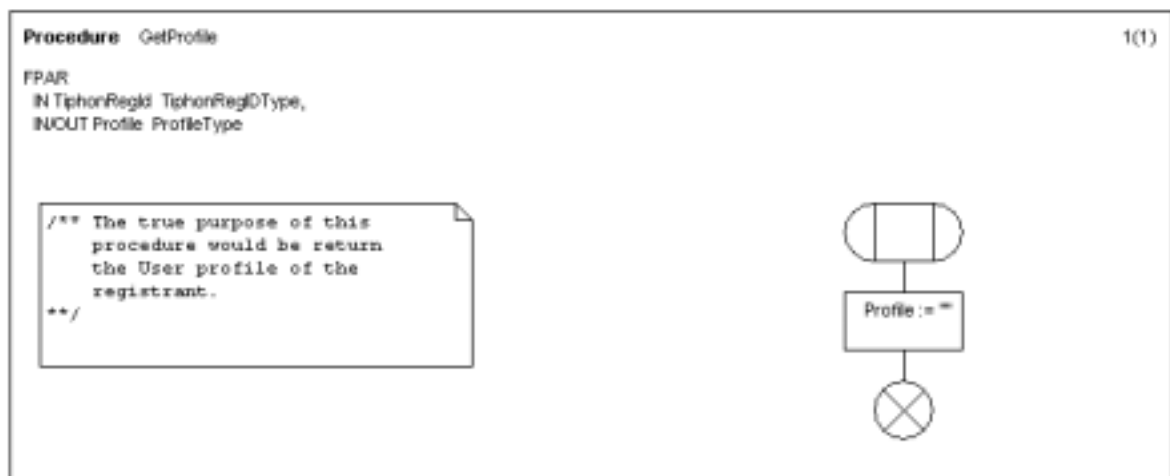


Figure A.8: Procedure GetProfile

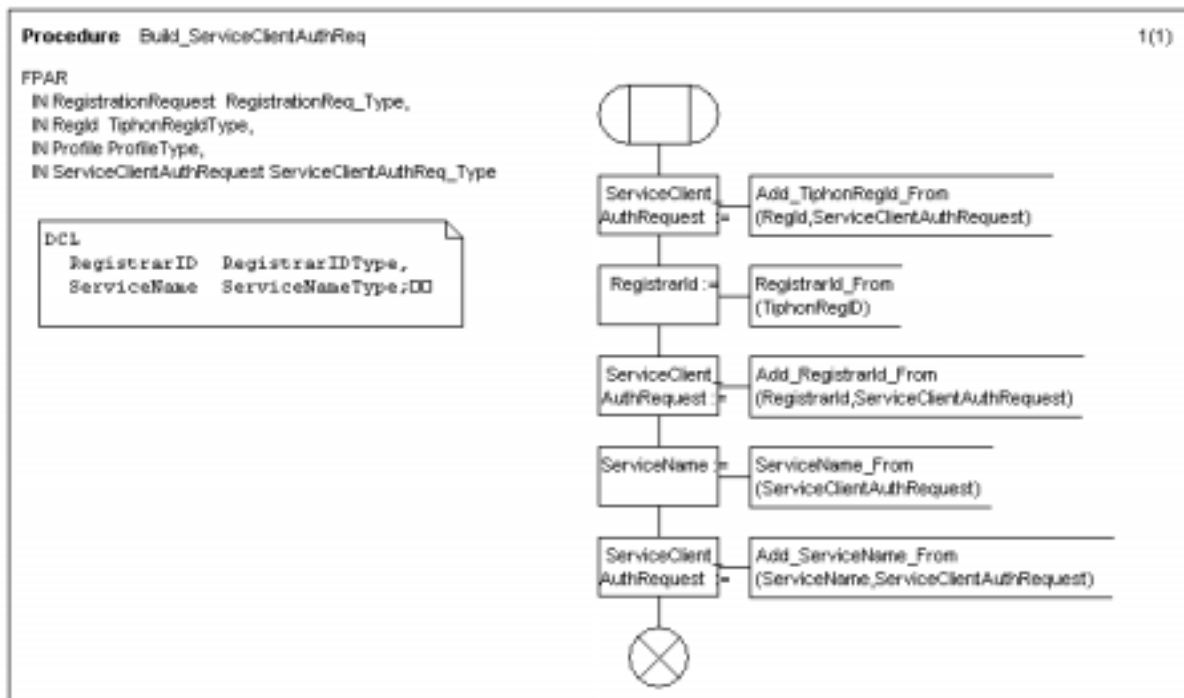


Figure A.9: Procedure Build_ServiceClientAuthReq

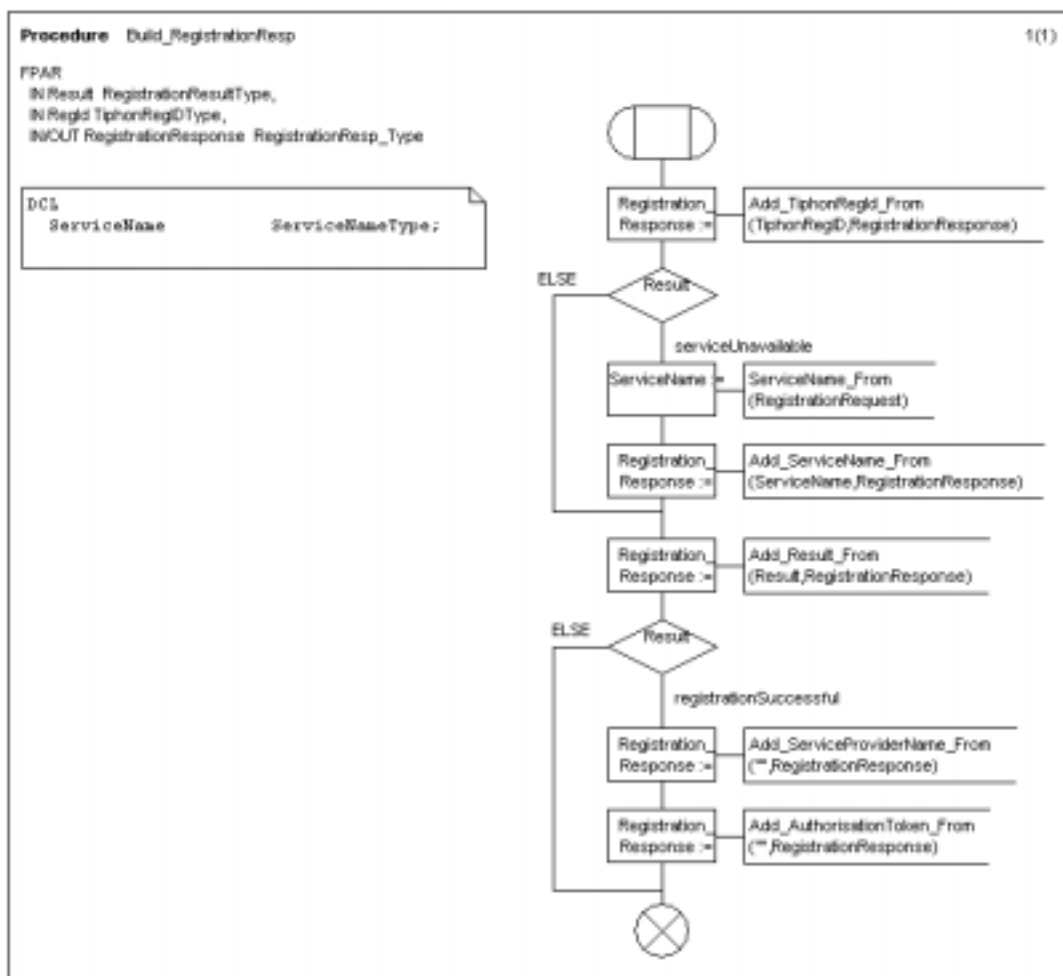


Figure A.10: Procedure Build_RegistrationResp

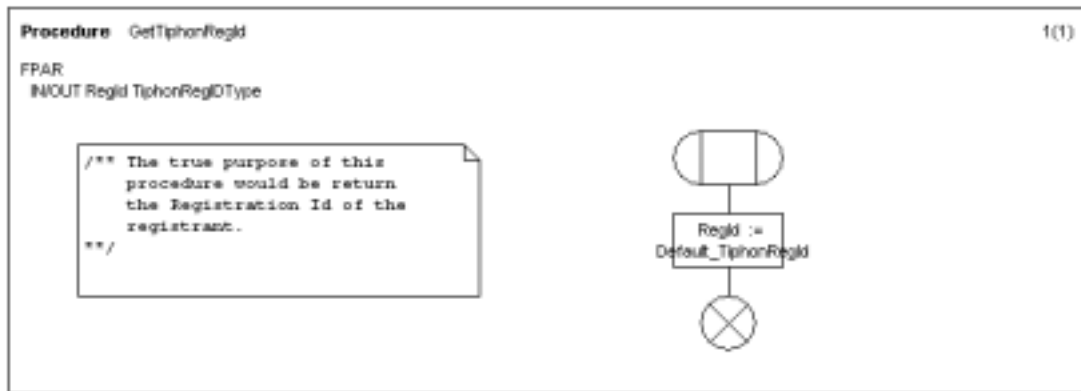


Figure A.11: Procedure GetTiphonRegId

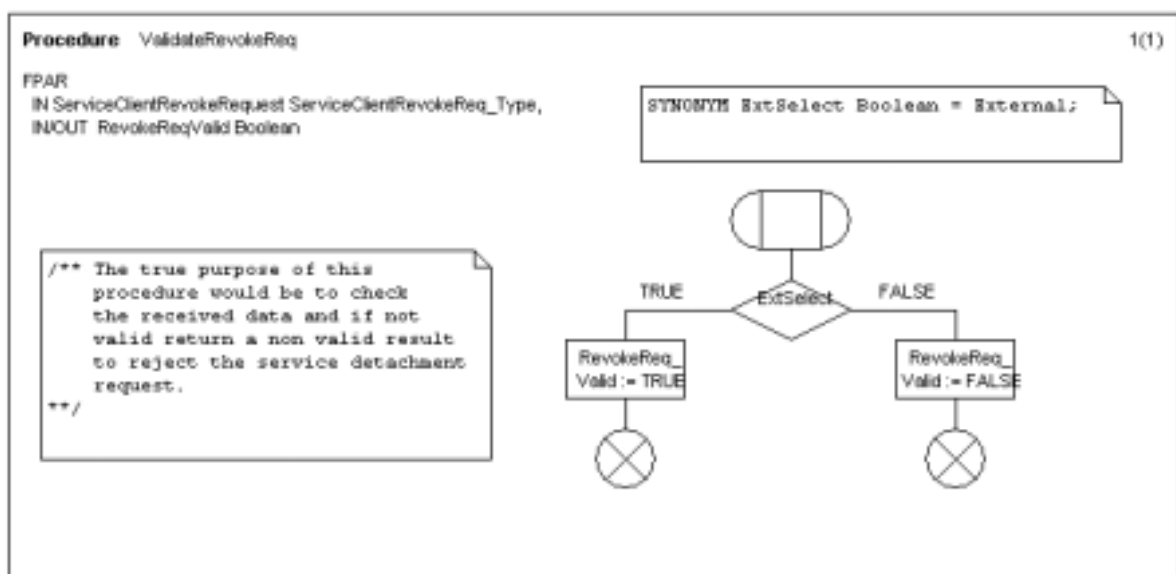


Figure A.12: Procedure ValidateRevokeReq

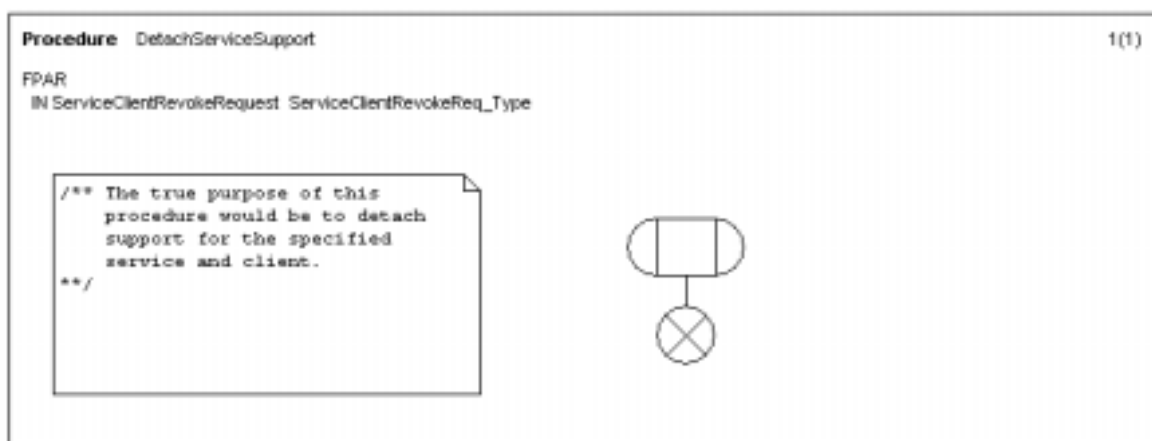


Figure A.13: Procedure DetachServiceSupport

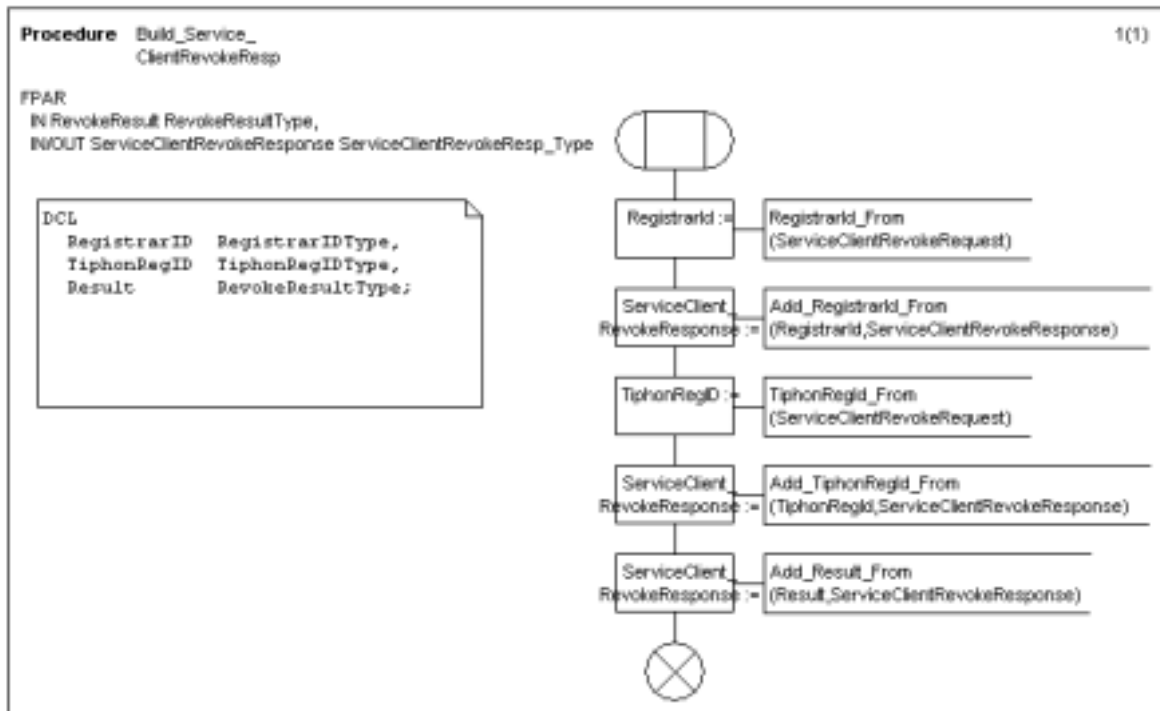


Figure A.14: Procedure Build_ServiceClientRevokeResp

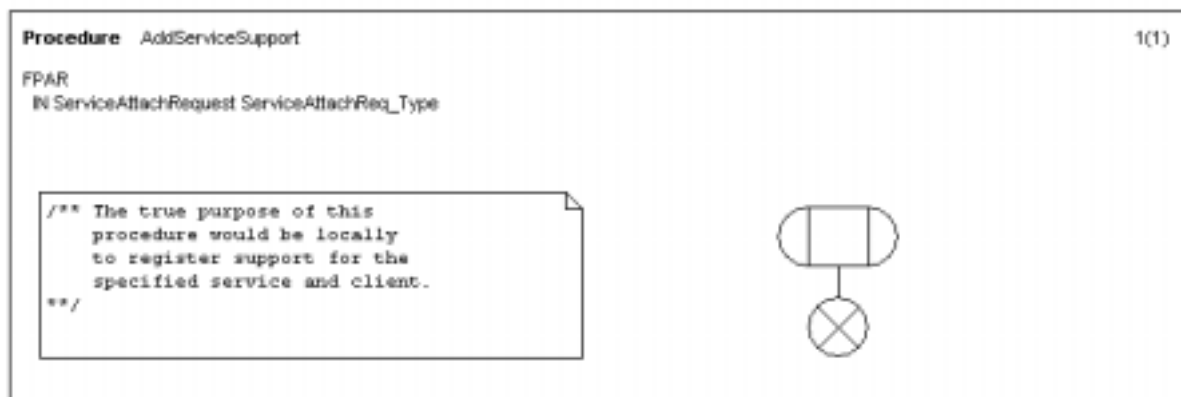


Figure A.15: Procedure AddServiceSupport

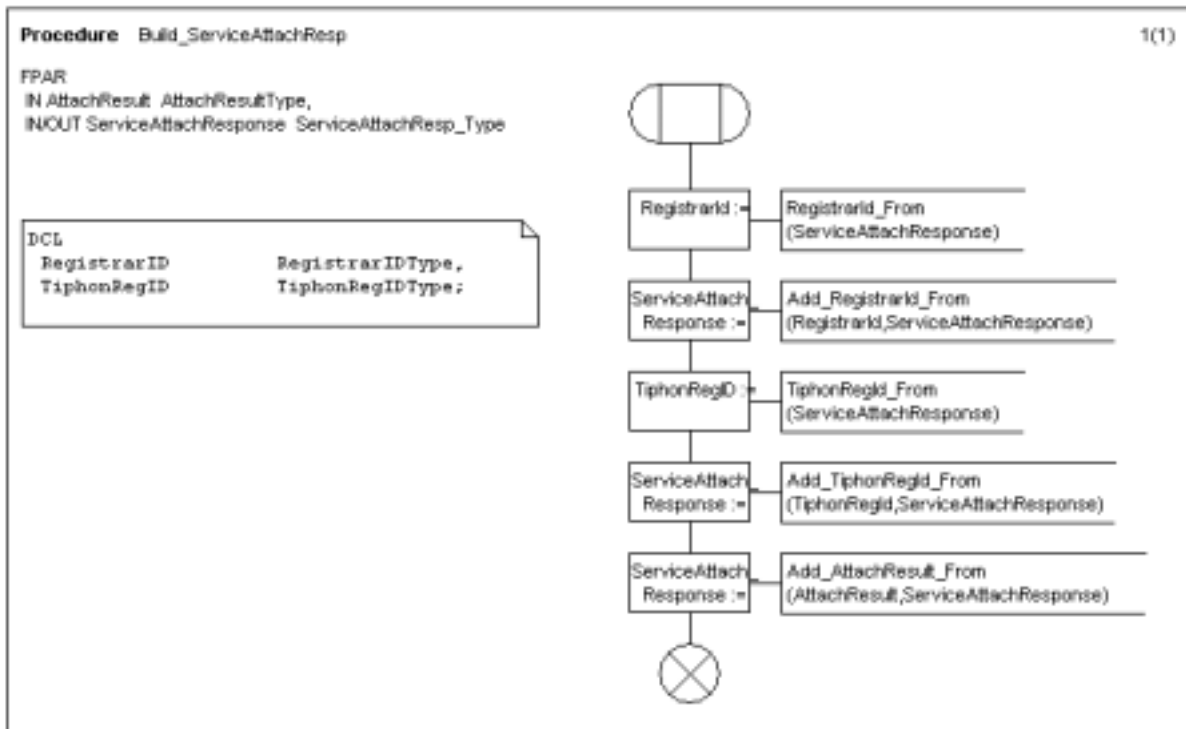


Figure A.16: Build_ServiceAttachResp

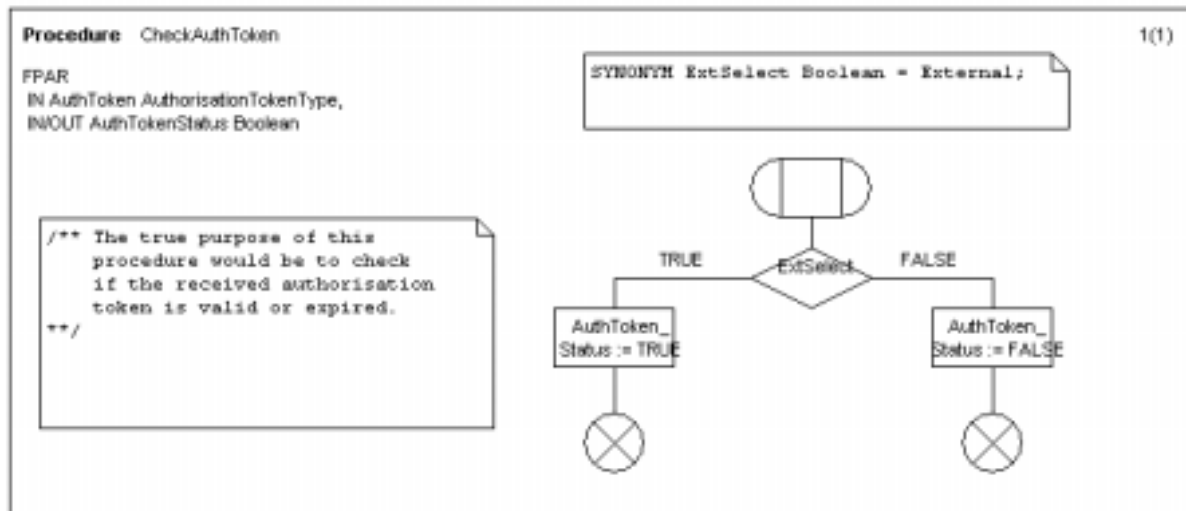


Figure A.17: Procedure CheckAuthToken

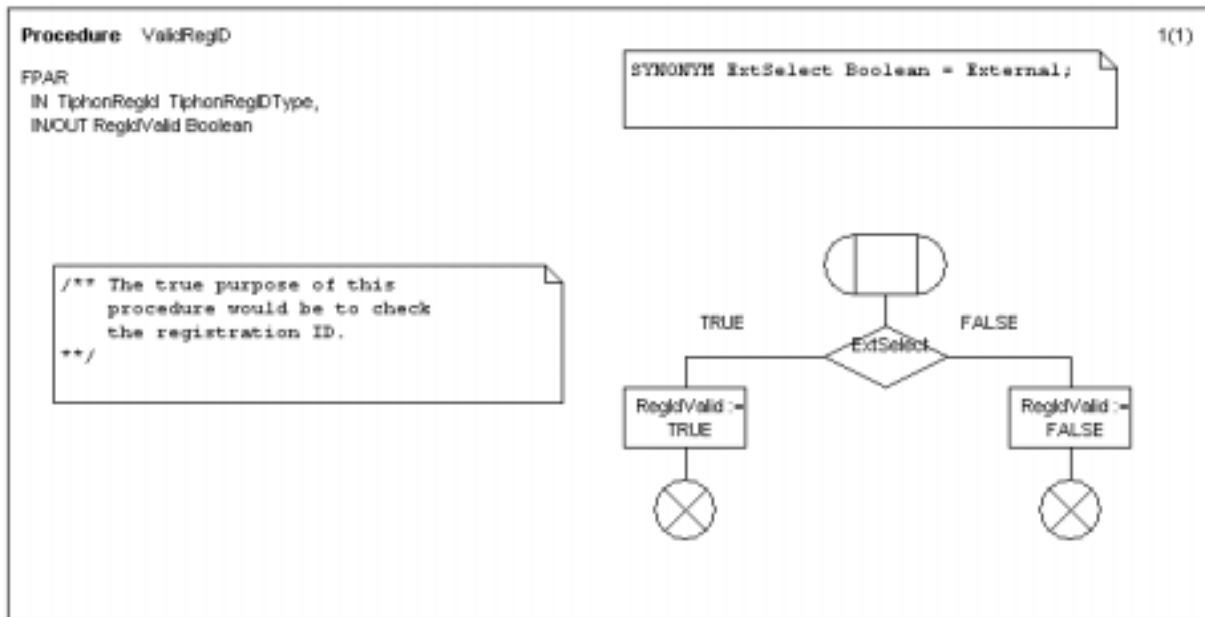


Figure A.18: Procedure ValidRegID

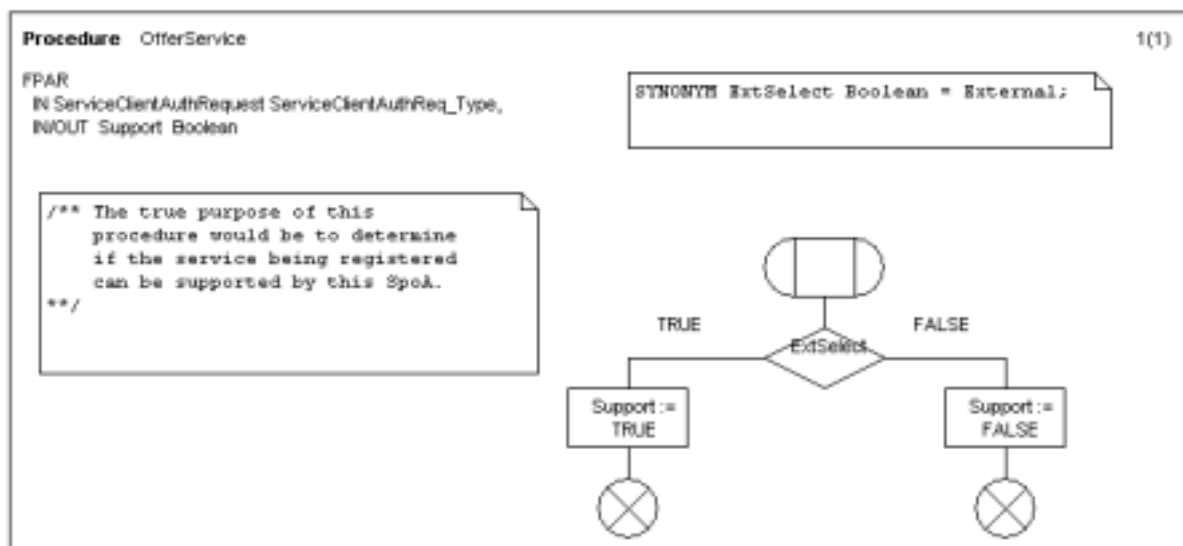


Figure A.19: Procedure OfferService

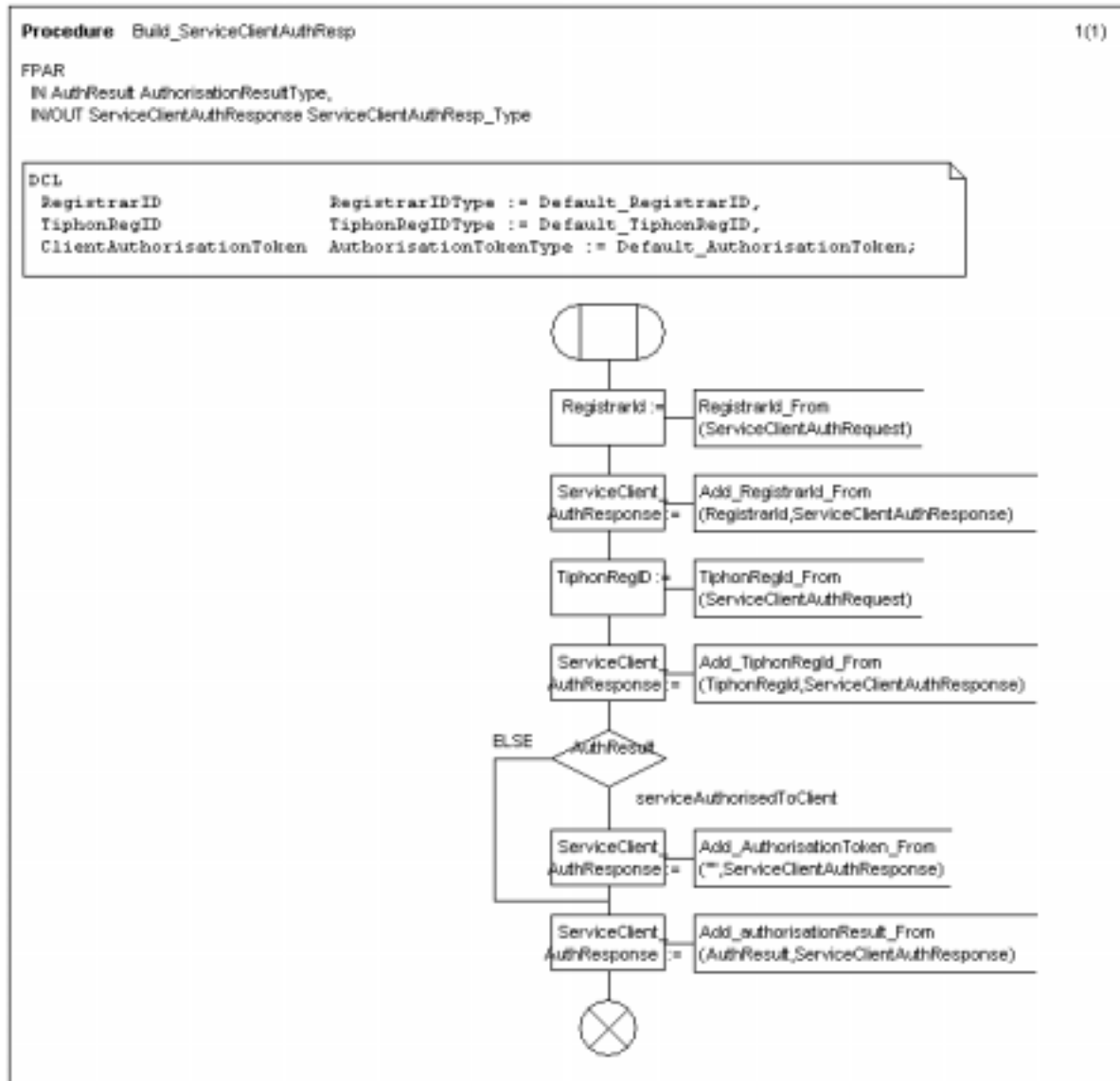


Figure A.20: Procedure Build_ServiceClientAuthResp

Annex B (informative): SDL model

The attached file represents the SDL model of TIPHON registration meta-protocol (developed using the Cinderella tool (a viewer is available from <http://www.cinderella.dk>).

Annex C (informative): Bibliography

- ETSI TS 101 314: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Network architecture and reference configurations; TIPHON Release 4".
- ETSI TS 101 315; Telecommunications and Internet protocol Harmonization Over Networks (TIPHON) Release 4; Functional Entities, Information Flow and Reference Point Definitions; Application of TIPHON functional architecture to inter-domain services.

History

| Document history | | |
|-------------------------|----------------|---------------------------|
| V1.1.1 | May 2002 | Publication as TS 101 882 |
| V4.1.1 | September 2003 | Publication |
| | | |
| | | |
| | | |