

Customization - Architectural Overview

Ericsson Dynamic Activation 1

USER GUIDE

Copyright

© Ericsson AB 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Purpose and Scope	1
1.2	Target Group	1
1.3	Typographic Conventions	1
2	Overview	1
3	Northbound Adapters	3
4	Service Realization	4
5	Network Abstraction and Network Implementation	4
5.1	Model Driven Configuration	4
5.2	Java Business Logic	5
6	Southbound Adapters	6
7	Platform Function	6
	Reference List	9





1 Introduction

Ericsson™ Dynamic Activation (EDA) is designed to provide services for various types of scenarios. It supports:

- Billions of subscribers
- More than 200 customers worldwide
- More than 800 installations
- Hundreds of customer adaptations

Dynamic Activation is often customized to fit into different scenarios.

1.1 Purpose and Scope

This document provides an architectural overview for customizations of Business Logic (BL) in Dynamic Activation.

1.2 Target Group

The target groups for this document are as follows:

- Solution architects
- Solution integrators

1.3 Typographic Conventions

Typographic conventions are described in the document *Library Overview*, Reference [1].

2 Overview

Dynamic Activation is a proven multi-vendor and multi-domain activation platform.

Dynamic Activation is shipped with standardized provisioning logic for the equipment of many vendors and domains, such as the following:

- Ericsson User Data Consolidation (UDC) solution (containing, for example, CUDB, HLR-FE, HSS-FE),
- Metro Ethernet Forum service (containing, for example, ELan, Eline, Etree)
- IP networking features (for example VRF, MPLS, OSPF)

Customizations can be made by a system integrator to support operator-specific scenarios. Typical examples are:

- Provisioning of Network Elements (NEs) from different vendors and business domains
- Combining IT with telecom systems
- Keeping consistency between service catalogs and IMS systems
- Connecting to in-house databases

The customizations allow the number of applicable scenarios to be greatly enhanced.

Ericsson recommends the layered architectural style shown in Figure 1. Customizations based on this architecture benefit from strong platform features provided by Dynamic Activation. Examples of these platform features are authentication and authorization, automatic clustering, consistency management, loose error handling, logging, load balancing, and High Availability support.

Customizations of business logic in Dynamic Activation are developed using the following layered architectural style:

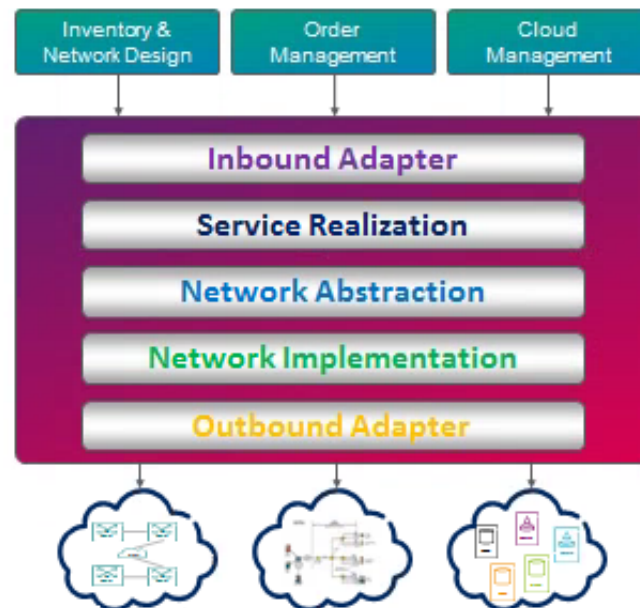


Figure 1 Layered Architecture

See the following sections for the customization instructions of each component shown in Figure 1.

3 Northbound Adapters

The northbound adapter layer provides the possibility to implement support for protocols required by the northbound systems, but not included in the standard product.

The default northbound protocol supported by Dynamic Activation is CAI3G. Other protocols can be supported by implementing Northbound Interface adaptor. For more information, refer to *Northbound Interface Adapter Customization Development Guide for HTTP-Based Protocol*, Reference [2].

4 Service Realization

To ease integration with northbound systems, operator can construct service logic based on available provisioning logics. A service reflects the business needs of the operator. The information model for a service differs from the information model of the network abstraction. The information model for a service can aggregate or segment the network services. Tailored execution plan can be defined. To secure atomicity of the business use case, rollback can be implemented.

Typically, a service interface provides a single point of provisioning for the Business Support System (BSS). The BSS sends only one request to Dynamic Activation. The service logic in Dynamic Activation analyzes the request and sends multiple requests to different network abstractions or network implementations. The service logic also sends a single reply to the BSS.

In Dynamic Activation, service logic can be configured by using the graphical tool Designer Studio. This is foremost recommended for straightforward tasks. For details, refer to *User Guide for Designer Studio*, Reference [3].

An alternative solution is to produce Java subscriber view. This solution is recommended for advanced manipulation of data. For more information, refer to *Customer Adaptation Development Guide for Resource Activation*, Reference [5].

Note: Java subscriber view cannot be converted to Designer Studio service model.

5 Network Abstraction and Network Implementation

5.1 Model Driven Configuration

Resource Configuration provides a model driven solution for configuration of devices through various interfaces, refer to *Function Specification Resource Configuration*, Reference [7] for the supported southbound interfaces. It enables integration of Network Elements without writing Java code. The solution is based on two main components.

- Model catalog with feature models



The feature model is a logical representation of a network functionality, like an IP interface or a BGP configuration in a router. This feature is exposed as a Managed Object (MO) in the northbound CAI3G interface. It provides a uniform way of managing this feature, regardless of what vendor and interface used to communicate with the actual network southbound. The feature model is defined as an XML schema or as a YANG model.

- Vendor templates

The Vendor template describes the contract of how to communicate with a certain type of device and map a generic feature into a vendor-specific implementation. The template can either be in XML or Yang format.

The XML template describes the command sequence to set up a feature in a device and how to map the data from the Northbound Interface to the southbound commands. All nodes that expose a CLI, NETCONF, Telnet, SNMP, or HTTP interface can be integrated with XML templates.

The YANG template provides a quicker way to integrate the device that provides a YANG or NETCONF interface. The solution uses a generic YANG engine to automatically generate the southbound NETCONF commands based on the constraints given by the device-specific YANG file.

It is possible to import additional feature models and vendor templates to extend the supported solutions. This is done through the GUI on the runtime system. It is also possible to edit existing templates from this GUI. When new models and templates have been imported or updated, these take effect on the runtime system.

For more information, refer to *Customer Adaptation Guide for Resource Configuration*, Reference [4].

5.2 Java Business Logic

Java business logic is used to implemented support for Network Elements required complex logic to handle the activation activities. Typical example is core network nodes like HSS.

Two layered java business logics is recommended for Dynamic Activation:

- Network abstraction

Business logic in this layer manages vendor/version/technology independent network features. It provides a higher abstraction of the existing network implementations. For example, HSS service on network abstraction layer manages provisioning of HSS service for both monolithic HSS and HSS-FE. Thus decouples network function from network implementation. This layer can be used to hide complex network topology, redundant, or load sharing network element instances.

- Network implementation

Business logic in this layer is dedicated to a specific version of a network element from a certain vendor, for example, Ericsson HSS-FE 15A. Typical functions implemented here are the following:

- Transform internal Dynamic Activation information model to NE-specific information model
- Data consistency mechanism including NE-specific rollback implementation
- Fault tolerance and Loose error handling

For more information, refer to *Customer Adaptation Development Guide for Resource Activation*, Reference [5].

6 Southbound Adapters

A southbound adapter contains the protocol-specific logic for a specific type of NE. Southbound adapters work as proxies toward NEs and do not contain any business logic.

Connection pooling, NE clustering, and error handling, such as loose error handling, are automatically provided by the Dynamic Activation platform.

Dynamic Activation is shipped with a set of common southbound interface adapters, for example HTTP, Telnet and LDAP. Solution integrators can add new adapters by using the standardized Java Connector Architecture (JCA). For more information about JCA, refer to *Customer Adaptation Development Guide for Resource Activation*, Reference [5].

Note: New southbound adapter, other than the southbound interfaces supported by SCM (such as SSH and NETCONF), for model driven configuration network abstraction and implementation approach can be implemented as product customization.

7 Platform Function

The following platform functions can be introduced for Dynamic Activation:



- Loose error handling rules

A GUI is provided for system integrator to configure loose error handling rules after the business logic is developed. For detail, refer to *User Guide for Resource Activation*, Reference [6].

- NE group type

A network element group type implements a specific cluster strategy that manages complex network topologies with a specific behavior. For details, refer to *Customer Adaptation Development Guide for Resource Activation*, Reference [5].





Reference List

Ericsson Documents

- [1] *Library Overview*, 18/1553-CSH 109 6284 Uen
- [2] *Northbound Interface Adapter Customization Development Guide for HTTP-Based Protocol*, 7/1553-CSH 109 628 Uen
- [3] *User Guide for Designer Studio*, 10/1553-CSH 109 628 Uen
- [4] *Customer Adaptation Guide for Resource Configuration*, 14/1553-CSH 109 628 Uen
- [5] *Customer Adaptation Development Guide for Resource Activation*, 5/1553-CSH 109 628 Uen
- [6] *User Guide for Resource Activation*, 1/1553-CSH 109 628 Uen
- [7] *Function Specification Resource Configuration*, 19/155 17-CSH 109 628 Uen