

# AdvOSS Policy Server

---

## PRODUCT DATA SHEET

Latest version of this Product Datasheet can be downloaded from  
[www.advoss.com/resources/datasheet/advoss-policy-server-product-datasheet.pdf](http://www.advoss.com/resources/datasheet/advoss-policy-server-product-datasheet.pdf)

Copyright © AdvOSS.com, 2007-2012

All Rights Reserved

## **Table of Contents**

1	AdvOSS Policy Server .....	3
2	Business Use Cases : .....	5
3	Modules .....	5
3.1	Zoning & Control Module .....	6
3.2	Rules Based Engine:.....	7
4	Integration Points:.....	8
5	Deployment Scenarios .....	8
6	Key Benefits .....	11

## 1 AdvOSS Policy Server

Policy Server is required by any service provider who has complex service management needs and have to make policy driven decisions based on business rules to realize those needs. AdvOSS Policy Server enables a service provider to create policies and define their business rules for any business process e.g. Routing, Charging, Resources allocation and others. It specially realizes the PCRF (Policy Charging Rules Function) as defined in the PCC framework of 3GPP for next generation networks like LTE, IMS, Wi-Max and other 4G technologies

Policy server provides the core rule based engine and request zoning modules that combine with HSS and AAA applications to provide a powerful policy framework that is fully compliant with the PCRF framework defined by 3GPP for policy and charging control.

AdvOSS policy framework goes beyond PCRF and it facilitates service providers to realize more complex use cases related to bandwidth control, Quality of Service (QoS) and Quality of Experience (QoE) management at the subscribe, their subscriptions and individual sessions/flow level.

Policy server works in close collaboration with HSS and AAA applications. AAA applications request policy rules evaluation from the policy server at several points during their workflow. They provide subscription and service profile related information to policy server, along with rating and charging related information.

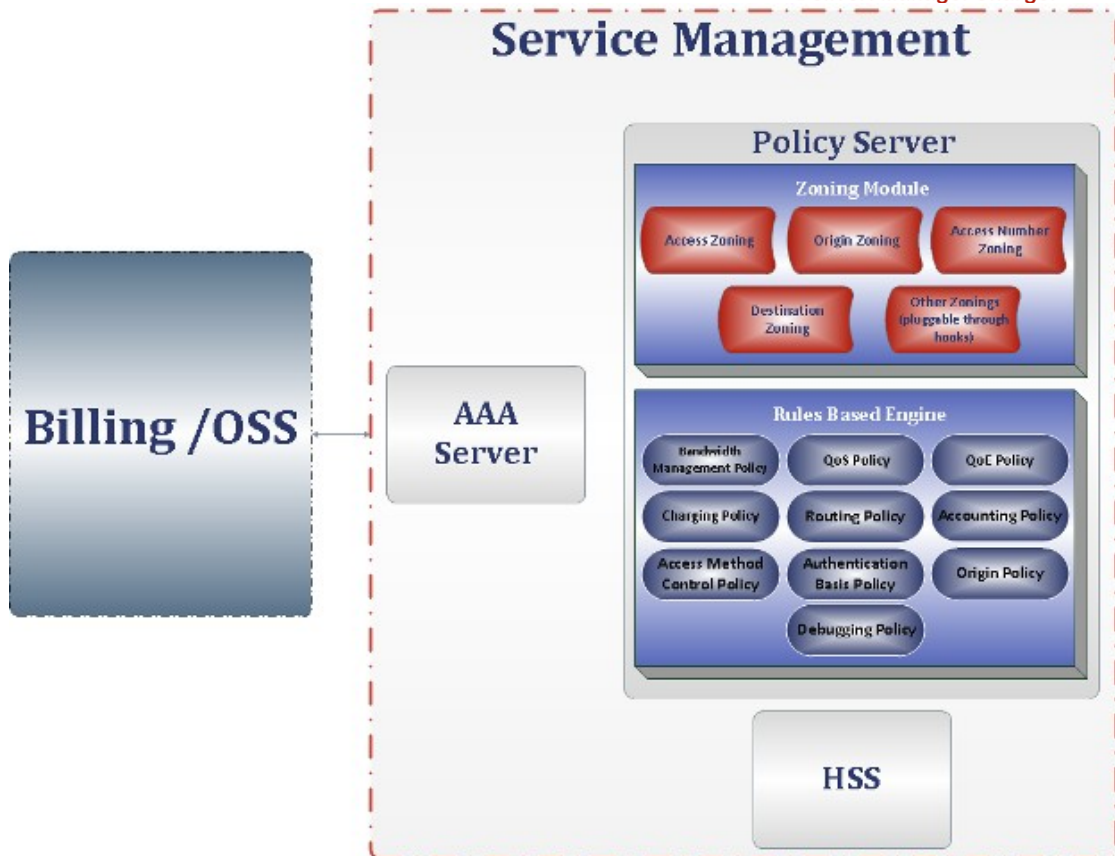
Policy server evaluates policy rules based on request zoning and grouping, service and subscription related information, and can evaluate rules of arbitrary complexity to output:

- Any changes in service profiles or service behavior
- Gating of service to allow or disallow access
- Service QoS and QoE modification

- Modification of how the service is to be charged e.g. prepaid or postpaid

Policies can be defined for several aspects of the service including:

- Authentication basis
- Authorization
- Accounting
- Charging and Rating
- Subscriber Account Balance
- Subscriber's loyalty score
- Routing and Redirection
- Access method
- Bandwidth and other network resource allocation
- Fair usage
- Debugging
- Time of day based policy definitions
- Proxying AAA requests to multiple targets



## 2 Business Use Cases :

- Bandwidth Management Policy
- QoS Policy
- QoE Policy
- Charging Policy
- Routing Policy
- Accounting Policy
- Access Method Control Policy
- Authentication Basis Policy
- Origin Policy
- Debugging Policy

## 3 Modules

The AdvOSS Policy Server is composed of the following modules:

AdvOSS Policy Server - Product Data Sheet

- Zoning & Control module
- Rules based Engine

### 3.1 Zoning & Control Module

Zoning Module is a mandatory module of the Policy Server. It allows defining matching rules on lists against IDs returned by the Network Elements. Zoning is used at three places:

- To exercise Control on allowing or denying access and more extensively on the Service Profile applied
- To help in defining charging rules through an optional Rules based Engine which are based on clubbing on Network IDs into different Zones.
- To help in defining Routing policies used to send traffic on the Vendor side

Zoning is of many types where each type of Zone is based on one type of ID returned from the Network.

#### **Access Zoning**

Allows zoning based on Framed IP addresses used by Subscribers to gain access to the network. It allows zoning based on Network topology awareness and sub-netting information.

#### **Origin Zoning**

Allows zoning based on Calling Station IDs. Can also be used to identify home networks from visited network in roaming cases and apply policies based on origin location.

#### **Access Number Zoning**

Allows zoning based on the Access Number used to gain access.

#### **Destination Zoning**

Allows zoning based on the Called Station ID. Typically used to group Request-URIs.

#### **Other Zonings (pluggable through hooks)**

### 3.2 Rules Based Engine:

Rules based engine is used to use the Zone info returned from the Zoning module to apply charging policies for Rating. It is also used to inform the Routing server about the Routing Policy.

## 4 Integration Points:

AdvOSS Policy Server offers points of integration with the following systems:

- **AAA**

AdvOSS Policy Server integrates with AAA Server for providing policy based triggers for service control. AAA application sends policy enforcement commands to network elements. These network elements or enforcement point police each flow according to the instructions received by policy server via AAA about the flow in question.

- **Charging & Rating Engine**

Policy Server may also interact with the Charging & Rating Engine for updating the charging policies.

- **HSS**

Policy Server integrates with the HSS which contains subscriber profile and subscription information.

- **Provisioning Application**

Policy Server integrates with the Provisioning Application

## 5 Deployment Scenarios

This section outlines different deployment scenarios in which the policy server can work with enforcement points. Deployment scenarios depend on the nature of enforcement points. Broadly, the enforcement points can be categorized into two different classes depending on their awareness about the Subscribers.

Without SM (Subscriber Manager): These enforcement points do not have a local Subscriber Manager.



With SM Subscriber Manager: These enforcement points do have a local Subscriber Manager attached to them.

**Deployment Scenario for Enforcement Points Without SM:**

For such Enforcement Points, AdvOSS Policy Server / HSS offers complete handling of user experience. Upon a new Session Initiation, the Enforcement Point queries the AdvOSS Policy Server and the AdvOSS Policy Server replies with the Name or Complete Profile of the Experience applicable at that Point of Time to the said Subscriber's Session.

If any of the Absolute or State variables dictate a change of experience after a Session has started, AdvOSS Policy Server arranges to push the new experience on the Enforcement Point for the said Subscriber Session. Pushing a different Experience to an ongoing session again depends on the capabilities of the Enforcement Point.

**EPs capable of accepting in-session change of experience:**

If the EP supports in session change of experience, AdvOSS Policy Server arranges to push the new experience through the exposed asynchronous API of the EP. Out of the box, AdvOSS Policy Server / HSS support Diameter Sh protocol to achieve this but other proprietary APIs can be easily added through the work flow engine provided.

**EPs not capable of accepting In-Session change of experience:**

If the EP is not capable of accepting a change of experience for an ongoing session, then AdvOSS Policy Server arranges to disconnect the said Session. For this purpose AdvOSS Policy Server queries the AAA Session Management Engine to find out currently connected Subscribers. Typically Radius and Diameter CoA (Change of Authorization) messages are used to disconnect ongoing sessions. Upon disconnection when the EP asks for Experience upon reconnection, the revised experience is pushed.

**Deployment Scenario for EPs with SM:**

If the EP has its own local repository for Subscribers, then the deployment scenario depends on the matching of the requirements of the Service Provider and the capabilities offered by the attached local SM. If the Requirements and Capabilities match, then local SM can be used. If there is mismatch, then AdvOSS Policy Server should be used. Following are some of the example scenarios.

## **Requirements of Service Provider**

### **Option A: No change in experience at all**

Local SM **can** always be used. Subscriber needs to be double provisioned in these cases. First on IN and then in local SM of EP.

### **Option B: Experience changes based on Absolute variables i.e. Time of Day etc.**

Local SM **can** be used if the EP is capable of handling different experiences based on the said absolute variables.

### **Option C: Experience changes based on State variables**

Local SM can be used but the Subscriber needs to be re-provisioned every time any of its life cycle state variables changes. If Variables effecting experience include variables other than life cycle states (e.g. origin, congestion, access), then the EP needs to be even more aware of them. Overall in such scenarios, even if EP is capable of it, it is NOT recommended to use EPs local SM as it may cause excessive network congestion and needless traffic.

## **Deployment Scenario with EPs without SM but with Policy Manager**

A special cases arises for EPs that do have a Policy Manager built-in. A Policy Manager is something that asks the Policy Server about the 'Name' of Policy when a new Session initiates and is capable of applying changing experiences based on absolute variables like Time of Day. In such situations, AdvOSS Policy Server will only send the 'Name' of the Policy to be applied and the rest of intelligence about changing experiences for different times of day can be handled by the EP itself.

## **Pushing Options**

EP may ask for the Policy to be given to it by Name only or it may need translation into detailed data structures accepted by the EP. AdvOSS Policy Server supports both the modes and has the capability to convert a Policy's name to a detailed data structure as accepted by the EP through the workflow engine provided.

## **Bulk changes**

If there is an experience change needed and AdvOSS Policy Server is configured to push the change to a connected Subscriber session, it does it as and when required based on the stimuli it receives from different

network elements. At times the change affects a large number of connected subscriber sessions. In that case, AdvOSS Policy Server queries the AAA Session Management Application for the list of all connected sessions affected by the changes and initiates a bulk change process on the EPs concerned. For this if a bulk API is available on the EP it is used otherwise AdvOSS Policy Server does the change one by one for each connected session.

### **Mix and Match Scenarios**

Some mix and match of above scenarios is possible based on unique needs of the Service Provider and the capabilities of different EPs. The objective of all these scenarios is network efficiency otherwise AdvOSS Policy Server is capable of handling all policy needs without assuming any capabilities on the Enforcement Points.

## **6 Key Benefits**

AdvOSS Service Management Platform offers the following key benefits to the CSPs, in addition to its feature set:

- **Scalability:** The System can scale linearly offering substantial advantages in terms of cost reduction and ease of scalability. AdvOSS uses various technologies and architecture to achieve scalability as described Software Quality Attributes whitepaper.
- **Reliability:** The system provides high reliability and ensure mechanism to avoid system downtime though fully redundant DB architecture.
- **Robustness:** The system is very robust and keeps performing in the fact of errors.
- **Customizability:** The system is designed to allow for easy and rapid customizations as per CSP requirements, on demand.
- **Disaster Recovery:** The System can be deployed in two geographically distant data centers, for disaster recovery reasons. AdvOSS Database solutions provide real-time DB replication to remote slaves. This gives the remote location an almost real-time

image of the live database and the system can fall back to remote location in a disaster situation.

- **Flexibility:** The system is service agnostic and has the architectural provision to support any business model. This allows for easy customizability to support specific requirements of a CSP.
- **Resilience:** The system works at high loads within its thresholds and gracefully keeps working by handling the situations even beyond the guaranteed thresholds e.g. by rejecting requests.
- **Speed:** The system guarantees high speed of operations and latency times associated with the operations.
- **Data Integrity:** The systems preserve the integrity and correct relationship of data in order to guarantee the correct behavior.
- **High Availability:** The system uses various techniques to ensure a high availability behavior.
- **Redundancy:** AdvOSS uses different technologies in combination to achieve redundancy in the system and make sure that the redundancy is used towards availability when needed. There are many types of Redundancy needed in the system as described in Technology section.