

SUPA
Internet-Draft
Intended status: Standards Track
Expires: October 8, 2016

T. Zhou
Huawei Technologies Co., Ltd
B. Wijnen, Ed.
Consultant
April 6, 2016

Architecture/Framework for SUPA
draft-bw-supa-architecture-00

Abstract

This document describes the architecture and framework for the Simple Use of Policy Abstractions (SUPA). It also gives an overview of the SUPA components.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 8, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Position of the policy engine	2
3.	policy engine framework	2
4.	Policy Data Model	3
5.	Information Model	4
6.	Security Considerations	4
7.	IANA Considerations	4
8.	Acknowledgements	4
9.	Informative References	4
	Authors' Addresses	5

[1.](#) Introduction[2.](#) Position of the policy engine

A network can be modeled with multiple layers. Policies can be applied to all the layers to achieve requirements from various type of actors.

device-level: policy can only be accessed and enforced on one device. The policy controls the dynamic behaviours, e.g. QoS, decapsulation, encapsulation, and forwarding.

network-level: policy can be configured to communicate with multiple network elements. The policy controls the adjustment of technique related network solutions, e.g. L3VPN, L2VPN.

service-level: policies are abstracted to be technique independent, and provided for the higher level users. The customer facing policy is provided to reduce the operation on service level agreement, generic VPN service, unified tunnel services.

[3.](#) policy engine framework

Figure 1 depicts the policy engine framework.

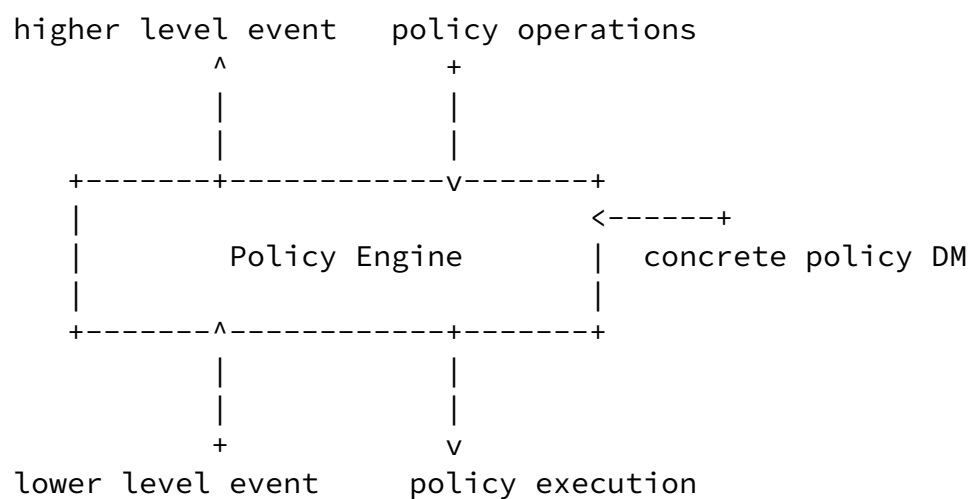


Figure 1: Figure 1 The policy engine framework

The policy engine is configured with concrete policy DMs, so that it can deal with assigned policies. The concrete policy DM can generate data-store and northbound interface for the policy engine. One or more standard protocols should be selected (e.g., NETCONF, RESTCONF) for policy operations to communicate with the Policy Engine. The policy engine runs with monitoring the lower level events from the southbound. The policy engine execute policies by doing actions or decomposing the policy to lower level policies. Higher level events may be generated by the policy engine, so that policy engine or applications sitting on a higher level can consume.

4. Policy Data Model

The policy data model describes in detail about the protocol operations and data-store content. It serves as an "API contract" honored by the policy engine, and is essential to the model driven policy API. The well defined policy model structure facilitates both flexibility and extensibility.

generic policy model: defines a generic policy header and the

policy body structure. The generic policy header contains information on, e.g. name, identifier, life cycle, which can be shared by all the specific policy models. The generic policy body could be a ordered list of policy rules. But the details on how the policy rule like is extended by the specific policy model, e.g. Event Condition Action (ECA) policy model.

specific policy model: inherits from the generic policy model with specific extensions on the policy rule. For example the ECA policy model extends the policy rule with Event-Condition-Action definition.

concrete policy model: is rendered based on the specific model by SDOs, vendors or operators. It represents concrete technique and vendor implementation. For example, a concrete Event, like time event, packet-in.

[5.](#) Information Model

How the information model can help data model generation? What should be defined in the Information Model (IM), what in the Data Model (DM)?

The IM document can have more words introducing what an item is and why we need an item. The IM helps other DM creation rather than YANG both in and outside IETF.

The DM document should be more on how to represent informations in for example YANG

[6.](#) Security Considerations

To do

[7.](#) IANA Considerations

This memo includes no request to IANA.

[8.](#) Acknowledgements

The authors would like to thanks the valuable comments made by:.

This document was produced using the xml2rfc tool [[RFC2629](#)].

9. Informative References

[I-D.bw-supra-declarative-policy-data-model]

Zhou, T., Xia, Y., and B. Wijnen, "A YANG Data Model for Declarative Policy", [draft-bw-supra-declarative-policy-data-model-00](#) (work in progress), December 2015.

[I-D.chen-supra-eca-data-model]

Chen, M., Contreras, L., Hayashi, M., and T. Tsou, "ECA Policy YANG Data Model", [draft-chen-supra-eca-data-model-05](#) (work in progress), October 2015.

[I-D.klyus-supra-proposition]

Klyus, M. and J. Strassner, "SUPA Value Proposition", [draft-klyus-supra-proposition-02](#) (work in progress), July 2015.

Zhou & Wijnen

Expires October 8, 2016

[Page 4]

Internet-Draft

SUPA Architecture/Framework

April 2016

[I-D.xia-sdnrg-nemo-language]

Xia, Y., Jiang, S., Zhou, T., and S. Hares, "NEMO (NETwork Modeling) Language", [draft-xia-sdnrg-nemo-language-03](#) (work in progress), October 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", [BCP 14](#), [RFC 2119](#),

DOI 10.17487/RFC2119, March 1997,

<<http://www.rfc-editor.org/info/rfc2119>>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", [RFC 2629](#),

DOI 10.17487/RFC2629, June 1999,

<<http://www.rfc-editor.org/info/rfc2629>>.

Authors' Addresses

Tianran Zhou
Huawei Technologies Co., Ltd
Q14, Huawei Campus, No.156 Beiqing Road
Hai-Dian District, Beijing, 100095
P.R. China

Email: zhoutianran@huawei.com

Bert Wijnen (editor)
Consultant
Schagen 33
3461 GL Linschoten
The Netherlands

Email: bertietf@bwijnen.net