            Policy-Based Management Framework for
          the Simplified Use of Policy Abstractions (SUPA)

Abstract

   The Simplified Use of Policy Abstractions (SUPA) policy-based
   management framework defines base YANG data models to encode policy.
   These models point to device-, technology-, and service-specific YANG
   data models developed elsewhere. Policy rules within an operator's
   environment can be used to express high-level, possibly network-wide,
   policies to a network management function (within a controller, an
   orchestrator, or a network element). The network management function
   can then control the configuration and/or monitoring of network
   elements and services. This document describes the SUPA basic
   framework, its elements, and interfaces.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Traffic flows over increasingly complex enterprise and service
   provider networks are becoming more and more important. Meanwhile,
   the rapid growth of this variety makes the task of network operations
   and management applications deploying new services much more
   difficult. Moreover, network operators want to deploy new services
   quickly and efficiently. Two possible mechanisms for dealing with
   this growing difficulty are 1) the use of software abstractions to
   simplify the design and configuration of monitoring and control
   operations and 2) the use of programmatic control over the
   configuration and operation of such networks. Policy-based
   management can be used to combine these two mechanisms into an
   extensible framework.

   There is a set of policy rules within an operator's environment that
   defines how services are designed, delivered, and operated.

   The SUPA (Simplified Use of Policy Abstractions) data model
   represents a high-level, possibly network-wide policy, which can be
   input to a network management function (within a controller, an
   orchestrator, or a network element). The network management function
   can then control the configuration and/or monitoring of network
   elements and services according to such policies.

   SUPA defines a Generic Policy Information Model (GPIM) [SUPA-INFO]
   for use in network operations and management applications. The GPIM
   defines concepts and terminology needed by policy management
   independent of the form and content of the policy rule. The Event-
   Condition-Action (ECA) Policy Rule Information Model (EPRIM)
   [SUPA-INFO] extends the GPIM by defining how to build policy rules
   according to the ECA paradigm.

   Both the GPIM and the EPRIM are targeted at controlling the
   configuration and monitoring of network elements throughout the
   service development and deployment life cycle. The GPIM and the
   EPRIM can both be translated into corresponding YANG [RFC6020]
   [RFC7950] modules that define policy concepts, terminology, and rules
   in a generic and interoperable manner; additional YANG modules may
   also be derived from the GPIM and/or EPRIM to manage specific
   functions.

   The key benefit of policy management is that it enables different
   network elements and services to be instructed to behave the same
   way, even if they are programmed differently. Management
   applications will benefit from using policy rules that enable
   scalable and consistent programmatic control over the configuration
   and monitoring of network elements and services.

   Some typical and useful instances for authors to understand the
   applicability of SUPA, such as SNMP blocking upon load of link
   reaching a threshold and virtual matching migration upon the changing
   of user location, are described in [SUPA-APP].

2. Terminology

2.1. Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in
   BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all
   capitals, as shown here.

2.2. Abbreviations and Definitions

   SUPA: The Simplified Use of Policy Abstractions is a policy-based
   management framework that defines a data model to be used to
   represent high-level, possibly network-wide policies. This data
   model can be input to a network management function (within a
   controller, an orchestrator, or a network element).

   YANG: An acronym for "Yet Another Next Generation". YANG is a data
   modeling language used to model configuration and state data
   manipulated by the Network Configuration Protocol (NETCONF), NETCONF
   remote procedure calls, and NETCONF notifications [RFC6020]

   ECA: Event-Condition-Action is a shortcut for referring to the
   structure of active rules in event-driven architecture and active
   database systems.

   EMS: An Element Management System is software used to monitor and
   control network elements (devices) in telecommunications.

   NMS: A Network Management System is a set of hardware and/or software
   tools that allow an IT professional to supervise the individual
   components of a network within a larger network management framework.

   OSS: An Operations/Operational Support System is a computer system
   used by telecommunications service providers to manage their networks
   (e.g., telephone networks).

   BSS: A Business Support System is used to support various end-to-end
   telecommunication services.

GPIM: A Generic Policy Information Model defines concepts and
terminology needed by policy management independent of the form and
content of the policy rule.

EPRIM: An ECA Policy Rule Information Model extends the GPIM by
defining how to build policy rules according to the ECA paradigm.

GPDM: Generic Policy Data Models [SUPA-DATA] are created from the
GPIM. These YANG data model policies are used to control the
configuration of network elements that model the service(s) to be
managed. The relationship between the information model (IM) and
data model (DM) can be founded in [RFC3444].

Declarative Policy: Policies that specify the goals to be achieved
but not how to achieve those goals (also called "intent-based"
policies). Please note that declarative policies are out of scope
for the initial phase of SUPA.

3. Framework for Generic Policy-Based Management

This section briefly describes the design and operation of the SUPA
policy-based management framework.

3.1. Overview

Figure 1 shows a simplified functional architecture of how SUPA is
used to define policies for creating snippets of network element
configurations. SUPA uses the GPIM to define a consensual vocabulary
that different actors can use to interact with network elements and
services. The EPRIM defines a generic structure for imperative
policies. The GPIM, and/or the combination of the GPIM and the
EPRIM, is converted to generic YANG modules.

In one possible approach (shown with asterisks in Figure 1), SUPA
Generic Policy and SUPA ECA Policy YANG modules together with the
Resource and Service YANG data models specified in the IETF (which
define the specific elements that will be controlled by policies) are
used by the Service Interface Logic. This Service Interface Logic
creates appropriate input mechanisms for the operator to define
policies (e.g., a web form or a script) for creating and managing the
network configuration. The operator interacts with the interface,
and the policies input by operators are then translated into
configuration snippets.

Note that the Resource and Service YANG data models may not exist.
In this case, the SUPA generic policy YANG modules serve as an
extensible basis to develop new YANG data models for the Service
Interface Logic. This transfers the work specified by the Resource

and Service YANG data models specified in the IETF into the Service
Interface Logic.

```
                    +--------------------+
   +----------+    \|      SUPA          |
   |  IETF    |---+----+  Information Models |
   +----------+   |   /|   GPIM and EPRIM  |
                  |    +---------+----------+
    Assignments  |             | Defines Policy Concepts
    and Managed  |            \|/
       Content   |    +---------+----------+
                 |   \|    SUPA Generic   |
              +----+    & ECA Policy      |
                 /|    YANG modules       |
                  +---------+----------+
                          *  Possible Approach
   +-------------------------------*---------------------------+
   |  Management System           *                           |
   |                             \*/                           |
   |           Fills  +--------+--------+  +-------------+     |
   | +--------+ Forms \| Service Interface |/ |Resource and |/ | +----+
   | |Operator|--------+     Logic        +--|Service YANG |----|IETF|
   | +--------+ Runs  /| (locally defined |\ | data models |\ | +----+
   |          scripts |forms, scripts,...)|  +-------------+   |
   |                  +--------+---------+                     |
   |                          \|/                              |
   |                   +-------+--------+                       |
   |                   | Local Devices  |                       |
   |                   | and Management |                       |
   |                   |    Systems     |                       |
   |                   +---------------+                        |
   +----------------------------------------------------------+
```
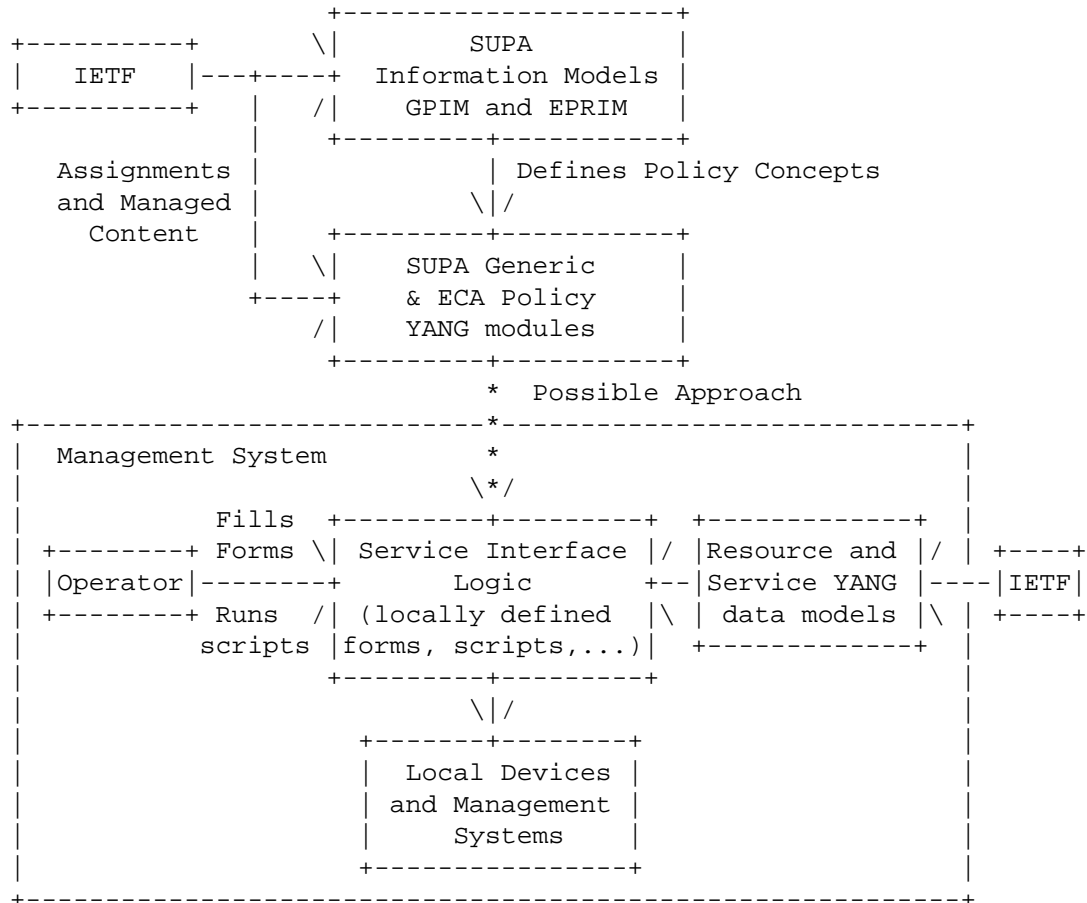
                    Figure 1: SUPA Framework

   Figure 1 shows the SUPA Framework at a high level of abstraction.
   The operator actor can interact with SUPA in other ways not shown in
   Figure 1. In addition, other actors (e.g., an application developer)
   that can interact with SUPA are not shown for simplicity.

   The EPRIM defines an ECA policy as an example of imperative policies.
   An ECA policy rule is activated when its event clause is true; the
   condition clause is then evaluated and, if true, signals the
   execution of one or more actions in the action clause. This type of
   policy explicitly defines the current and desired states of the
   system being managed. Imperative policy rules require additional
   management functions, which are explained in Section 3.2.

Figure 2 shows how the SUPA Policy Model is used to create policy
data models step-by-step and how the policy rules are used to
communicate among various network management functions located on
different layers.

The GPIM is used to construct policies. The GPIM defines generic
policy concepts as well as two types of policies: ECA policy rules
and declarative policy statements.

A set of Generic Policy Data Models (GPDM) are then created from the
GPIM. These YANG data model policies are then used to control the
configuration of network elements that model the service(s) to be
managed.

Resource and Service YANG Data Models: Models of the service as well
as physical and virtual network topology including the resource
attributes (e.g., data rate or latency of links) and operational
parameters needed to support service deployment over the network
topology.
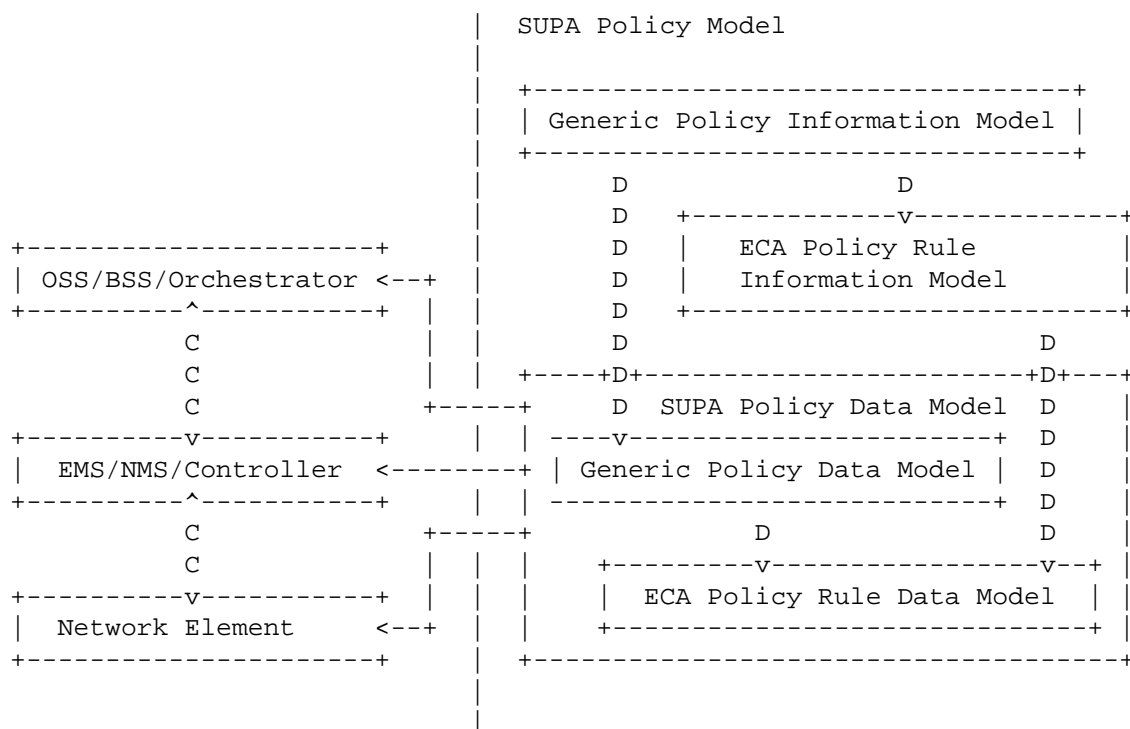
```
                                |  SUPA Policy Model
                                |
                                |  +--------------------------------+
                                |  | Generic Policy Information Model |
                                |  +--------------------------------+
                                |        D                 D
                                |        D    +-------------v------------+
 +---------------------+        |        D    |    ECA Policy Rule       |
 | OSS/BSS/Orchestrator <--+    |        D    |    Information Model      |
 +----------^----------+   |    |        D    +--------------------------+
            C             |    |        D                         D
            C             |    |  +----+D+-----------------------+D+---+
            C           +-----+      D  SUPA Policy Data Model   D    |
 +----------v----------+   |  |  ----v----------------------+  D    |
 |  EMS/NMS/Controller  <---------+ | Generic Policy Data Model |  D    |
 +----------^----------+   |  |  --------------------------+  D    |
            C            +-----+          D                   D    |
            C             |  |  |  +---------v----------------v--+ |
 +----------v----------+   |  |  |  |  ECA Policy Rule Data Model  | |
 |   Network Element    <--+  |  |  +----------------------------+ |
 +---------------------+   |  |  +--------------------------------+
                                |
                                |
Legend:
The double-headed arrow with Cs = "communication"
The arrow with Ds = "derived from"
```

                        Figure 2: SUPA Policy Model Framework

SUPA Policy Model: This model represents one or more policy modules
   that contain the following entities:

   Generic Policy Information Model: A model for defining policy
      rules that are independent of data repository, data definition,
      query, implementation language, and protocol. This model is
      abstract and is used for design; it MUST be turned into a data
      model for implementation.

   Generic Policy Data Model: A model of policy rules that are
      dependent on data repository, data definition, query,
      implementation language, and protocol.

   ECA Policy Rule Information Model (EPRIM): This model represents
      a policy rule as a statement that consists of an event clause,
      a condition clause, and an action clause. This type of policy
      rule explicitly defines the current and desired states of the
      system being managed. This model is abstract and is used for
      design; it MUST be turned into a data model for implementation.

   ECA Policy Rule Data Model: A model of policy rules, derived from
      EPRIM, where each policy rule consists of an event clause, a
      condition clause, and an action clause.

   EMS/NMS/Controller: This represents one or more entities that are
      able to control the operation and management of a network
      infrastructure (e.g., a network topology that consists of
      network elements).

   Network Element (NE): An element that can interact with the local
      or remote EMS/NMS/Controller in order to exchange information,
      such as configuration information, policy-enforcement
      capabilities, and network status.

Relationships among Policy, Service, and Resource models are
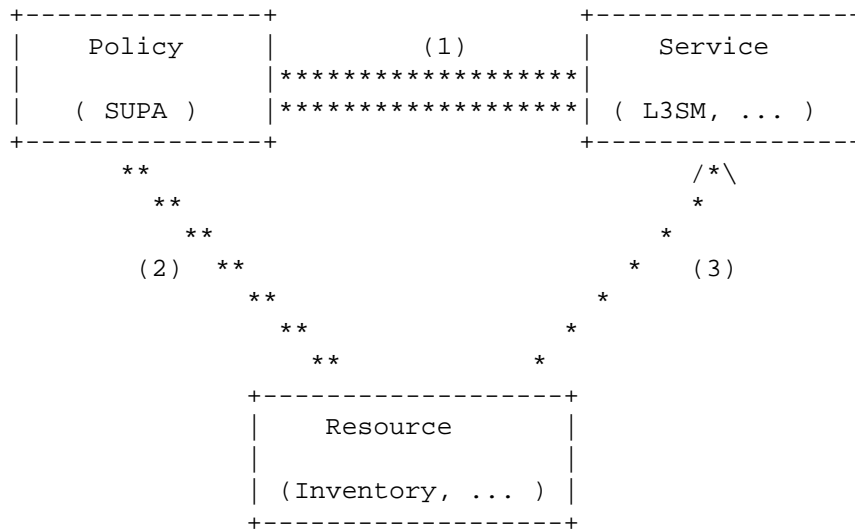illustrated in Figure 3.

```
        +---------------+                   +----------------+
        |    Policy     |       (1)         |   Service      |
        |               |*******************|                |
        |   ( SUPA )    |*******************|  ( L3SM, ... )  |
        +-------------+                     +---------------+
           **                                     /*\
             **                                    *
               **                                 *
            (2)  **                          *   (3)
                  **                        *
                    **                     *
                      **                  *
                    +-------------------+
                    |     Resource      |
                    |                   |
                    | (Inventory, ... ) |
                    +-------------------+
```

        Figure 3: Relationship among Policy, Service, and Resource Models

   In Figure 3:

   (1) The policy manages and can adjust service behavior as necessary
        (1:1..n). In addition, data from resources and services are
        used to select and/or modify policies during runtime.

   (2) The policy manages and can adjust resource behavior as necessary
        (1:1..n).

   (3) Resource hosts service; changing resources may change service
        behavior as necessary.

   Policies are used to control the management of resources and
   services, while data from resources and services are used to select
   and/or modify policies during runtime. More importantly, policies
   can be used to manage how resources are allocated and assigned to
   services. This enables a single policy to manage one or multiple
   services and resources as well as their dependencies. The use of
   (1:1..n) in point (1) and (2) above show that one policy rule is able
   to manage and can adjust one or multiple services/resources. Lines
   (1) and (2) (connecting policy to resource and policy to service) are
   the same, and line (3) (connecting resource to service) is different
   as it's navigable only from resource to service.

3.2. Operation

   SUPA can be used to define various types of policies, including
   policies that affect services and/or the configuration of individual
   network elements or groups of network elements. SUPA can be used by
   a centralized and/or distributed set of entities for creating,
   managing, interacting with, and retiring policy rules.

   The SUPA scope is limited to policy information and data models.
   SUPA does not define network resource data models or network service
   data models; both are out of scope. Instead, SUPA makes use of
   network resource data models defined by other working groups or
   Standards Development Organizations (SDOs).

   Declarative policies are out of scope for the initial phase of SUPA.

3.3. The GPIM and the EPRIM

   The GPIM provides a shared vocabulary for representing concepts that
   are common to different types of policies, but which are independent
   of language, protocol, repository, and level of abstraction. Hence,
   the GPIM defines concepts and vocabulary needed by policy management
   systems independent of the form and content of the policy. The EPRIM
   is a more specific model that refines the GPIM to specify policy
   rules in an ECA form.

   This enables different policies at different levels of abstraction to
   form a continuum, where more abstract policies can be translated into
   more concrete policies and vice versa. For example, the information
   model can be extended by generalizing concepts from an existing data
   model into the GPIM; the GPIM extensions can then be used by other
   data models.

3.4. Creation of Generic YANG Modules

   An information model is abstract. As such, it cannot be directly
   instantiated (i.e., objects cannot be created directly from it).
   Therefore, both the GPIM and the combination of the GPIM and the
   EPRIM are translated into generic YANG modules.

   SUPA will provide guidelines for translating the GPIM (or the
   combination of the GPIM and the EPRIM) into concrete YANG data models
   that define how to manage and communicate policies between systems.
   Multiple imperative policy YANG data models may be instantiated from
   the GPIM (or the combination of the GPIM and the EPRIM). In
   particular, SUPA will specify a set of YANG data models that will
   consist of a base policy model for representing policy management
   concepts independent of the type or structure of a policy; it will

also specify an extension for defining policy rules according to the
ECA paradigm. (Note: This means that policies can be defined using
the GPIM directly, or using the combination of the GPIM and the
EPRIM. If you use only the GPIM, you get a technology- and vendor-
independent information model that you are free to map to the data
model of your choice; note that the structure of a policy is NOT
defined. If you use the GPIM and the EPRIM, you get a technology-
and vendor-independent information model that defines policies as an
ECA policy rule (i.e., imperative).)

The process of developing the GPIM, the EPRIM, and the derived/
translated YANG data models is realized following the sequence shown
below. After completing this process and, if the implementation of
the YANG data models requires it, the GPIM and EPRIM and the derived/
translated YANG data models are updated and synchronized.

    (1)=>(2)=>(3)=>(4)=>(3')=>(2')=>(1')

    Where:
    (1)=GPIM
    (2)=EPRIM
    (3)=YANG data models
    (4)=Implementation
    (3')=update of YANG data models
    (2')=update of EPRIM
    (1')=update of GPIM

The YANG module derived from the GPIM contains concepts and
terminology for the common operation and administration of policy-
based systems as well as an extensible structure for policy rules of
different paradigms. The YANG module derived from the EPRIM extends
the generic nature of the GPIM by representing policies using an ECA
structure.

The above sequence allows for the addition of new model elements, as
well as the editing of existing ones, in the GPIM and EPRIM. In
practice, the implementation sequence may be much simpler.
Specifically, it is unlikely that the GPIM will need to be changed.
In addition, changes to the EPRIM will likely be focused on fine-
tuning the behavior offered by a specific set of model elements.

4. Security Considerations

   This informational document presents the framework and workflow of
   SUPA as well as an explanation on the relationship of policy, service
   and resources. This document does not introduce any new security
   issues, and the framework has no security impact on the Internet.
   The same considerations are relevant as those for the base NETCONF
   protocol (see Section 9 in [RFC6241]).

5. IANA Considerations

   This document has no IANA actions.

6. References

6.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

6.2. Informative References

   [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between
              Information Models and Data Models", RFC 3444,
              DOI 10.17487/RFC3444, January 2003,
              <https://www.rfc-editor.org/info/rfc3444>.

   [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for
              the Network Configuration Protocol (NETCONF)", RFC 6020,
              DOI 10.17487/RFC6020, October 2010,
              <https://www.rfc-editor.org/info/rfc6020>.

   [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed.,
              and A. Bierman, Ed., "Network Configuration Protocol
              (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011,
              <https://www.rfc-editor.org/info/rfc6241>.

   [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language",
              RFC 7950, DOI 10.17487/RFC7950, August 2016,
              <https://www.rfc-editor.org/info/rfc7950>.

   [SUPA-APP] Cheng, Y., Liu, D., Fu, B., Zhang, D., and N. Vadrevu,
             "Applicability of SUPA", Work in Progress,
             draft-cheng-supa-applicability-01, March 2017.

   [SUPA-DATA]
             Halpern, J., Strassner, J., and S. Van der Meer, "Generic
             Policy Data Model for Simplified Use of Policy
             Abstractions (SUPA)", Work in Progress, draft-ietf-supa-
             generic-policy-data-model-04, June 2017.

   [SUPA-FRAME]
             Zhou, C., Contreras, L., Sun, Q., and P. Yegani, "The
             Framework of Simplified Use of Policy Abstractions
             (SUPA)", Work in Progress, draft-zhou-supa-framework-02,
             May 2015.

   [SUPA-INFO]
             Strassner, J., Halpern, J., and S. Meer, "Generic Policy
             Information Model for Simplified Use of Policy
             Abstractions (SUPA)", Work in Progress, draft-ietf-supa-
             generic-policy-info-model-03, May 2017.

   [SUPA-STATE]
             Karagiannis, G., Strassner, J., Sun, Q., Contreras, L.,
             Yegani, P., and J. Bi, "Problem Statement for Simplified
             Use of Policy Abstractions (SUPA)", Work in Progress,
             draft-karagiannis-supa-problem-statement-07, June 2015.

   [SUPA-VALUE]
             Klyus, M., Strassner, J., Liu, W., Karagiannis, G., and J.
             Bi, "SUPA Value Proposition", Work in Progress,
             draft-klyus-supa-value-proposition-00, March 2016.

Contributors

   The following people contributed to the creation of this document,
   listed in alphabetical order:

      Luis M. Contreras, Telefonica I+D
      Dan Romascanu, Avaya
      Juergen Schoenwaelder, Jacobs University, Germany
      Qiong Sun, China Telecom
      Parviz Yegani, Huawei Technologies
      Cathy Zhou, Huawei Technologies

Authors' Addresses

   Will (Shucheng) Liu
   Huawei Technologies
   Bantian, Longgang District
   Shenzhen 518129
   China

   Email: liushucheng@huawei.com

   Chongfeng Xie
   China Telecom
   China Telecom Information Technology Innovation Park
   Beijing 102209
   China

   Email: xiechf.bri@chinatelecom.cn

John Strassner
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95138
United States of America

Email: john.sc.strassner@huawei.com

Georgios Karagiannis
Huawei Technologies
Hansaallee 205
Dusseldorf 40549
Germany

Email: Georgios.Karagiannis@huawei.com

Maxim Klyus

Email: xmaruto@gmail.com

Jun Bi
Tsinghua University
Network Research Center, Tsinghua University
Beijing 100084
China

Email: junbi@tsinghua.edu.cn

Ying Cheng
China Unicom
No.21 Financial Street, XiCheng District
Beijing 100033
China

Email: chengying10@chinaunicom.cn

Dacheng Zhang
Huawei Technologies
Beijing
China

Email: dacheng.zhang@huawei.com