Applicata BRINGS VALUE NOT COSTS

JN Signalling Processing Unit Configuration and Monitoring Interface

Easy integration
High Scalability and Availability
Signalling Network Function Virtualisation
Integrated Protocol Stacks, Configuration and Monitoring

1. Applicata Signalling Processing Unit

Applicata JN Signalling Processing Unit (SPU) is a signalling layer software-only implementation covering SS7, SIP[†] and DIAMETER protocols.

It implements the separation of signalling and application processes in a distributed architecture. No specialised hardware is required and SPU functions can be virtualised.

The signalling stacks and the application(s) run in different processes communicating with each other over IP.

All applications and application instances at the application plane share the same Signalling Processing Unit(s) running at the signalling plane. The applications can be easily scaled without affecting the signalling layer. Typically, two SPUs are enough at the signalling layer, providing dual resiliency and high availability.

Applicata JN SPU offers a cost effective and feature reach signalling solution with integrated protocol stacks, configuration and monitoring, rate control, message dispatching, very high performance, easy integration, network virtualisation, service high availability and scalability.

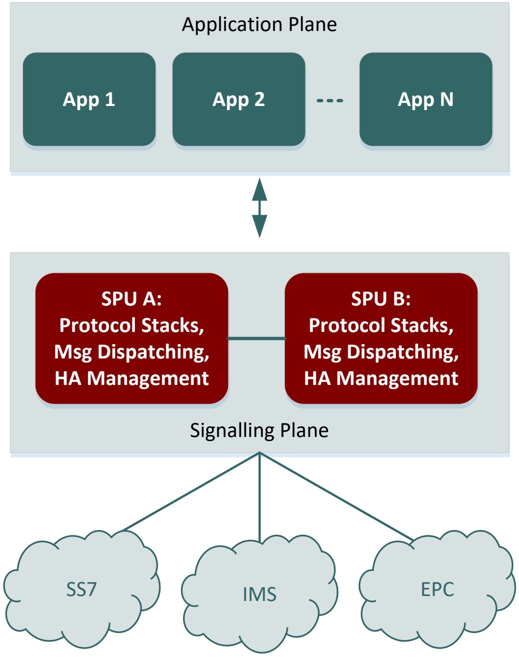


Figure 1. Applicata Signalling Processing Unit:
Signalling and Applications planes are separated in a distributed signalling architecture.
Applications can scale easily sharing the same signalling modules and licenses.

2. JN SPU OA&M Interface

This white paper describes Applicata JN SPU integrated OA&M interface. It provides an easy to use functionality for manual or automated system configuration and monitoring. Diameter stack has been used when illustrating the concepts in this document, the same applies for SS7 and SIP[†] stacks.

JN SPU management interface is secured over SSH protocol. The system implements a custom SSH shell process that provides the corresponding management functionality. The user access authentication is based on password or public/private keys.

The configuration and monitoring commands over the management interface can be invoked either manually or automatically. The manual use requires a standard SSH client console.

The configuration and monitoring functionality is based on a management model of the SPU system that represents the configurable and monitorable parameters in a tree view. The custom shell implements commands to view and/or edit the branches or leaves in the tree.

JN SPU custom SSH shell for OA&M is implemented very efficiently. New SSH connections require starting of an Erlang process, a very light operation that does not create a new OS process. This efficiency is especially important for handling the monitoring requests sent periodically and, possibly, frequently by monitoring consoles that typically use a pull based monitoring mechanism.

The use of OA$M interface for JN SPU configuration and management is described in next sections. An example of using Nagios for automated JN SPU monitoring is also included.

Inside this publication:

KEY POINTS

- **Easy integration with applications written in different languages**
- **Reduced costs of ownership**
- **Proven and extendable platform**
- **Extremely high performance and availability**
- **Round the clock support**

[†] In the roadmap

3. JN SPU Management Data Model and OA&M CLI Shell

JN SPU OA&M interface uses a model that represents the related configuration and monitored data as a tree. OA&M interface provides commands for listing, editing and saving this data, either manually or automatically.

A fragment of JN SPU management data tree is shown in Figure 2. It comprises both static and dynamic data. The static data belongs to the configuration of the SPU node . The dynamic data (shown in Red in Figure 2) changes during the processing and reflects the current state of the configured SPU elements.

Each node in the management data model has a `Node Type`, `Node Name` and may have a value with a `Value Type`. Nodes in the tree can be of type `Property`, `List`, `Choice`, `LeafList` or `Leaf`. The first three types appear at the root or as nodes at the middle of the tree, these are parents of some descendant nodes in the management data tree. `LeafList` type has multiple values and no descendants. `Leaf` type has single value and no descendants.

JN SPU OA&M server listens on port 8822 by default and starts a command line interface (CLI) shell when connected over ssh. CLI shell provides browsing and configuration modes of operation. The configuration tree can be listed and examined in browsing mode and it can be changed in configuration mode. The `help` command prints context sensitive help in JN SPU CLI shell. The output of help command while in browsing mode is shown in Figure 3.

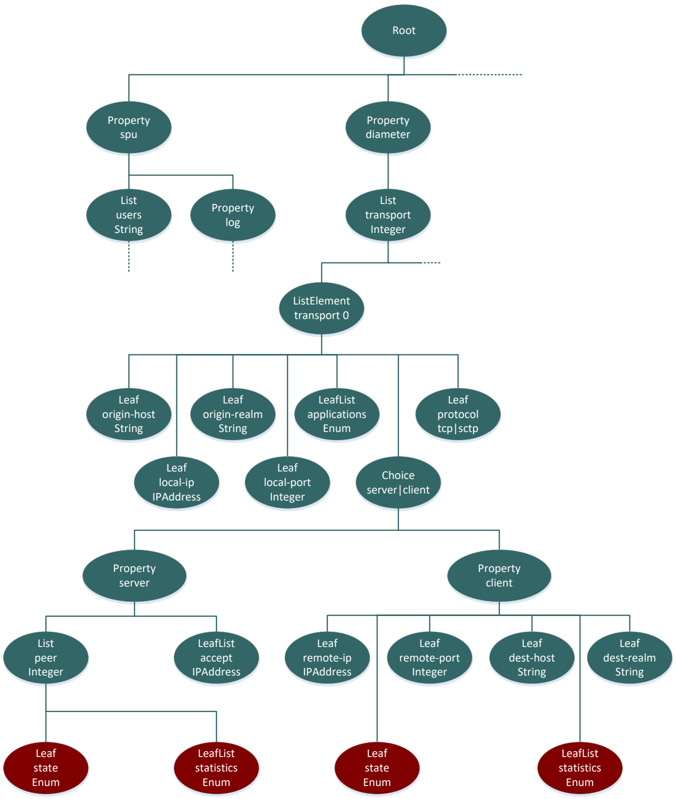


Figure 2. Fragment of JN SPU management data tree, dynamic data shown by nodes in Yellow

```
09:19 $ ssh -p 8822 admin@10.99.99.11
admin@10.99.99.11's password:
> help
Use 'show [NodeName ...]' to list the nodes and their val-
ues
Use 'config' to enter configuration mode
Use 'exit' to exit CLI shell

>
```

Figure 3. Context sensitive `help` command

JN SPU CLI shell provides the `show` command that lists the nodes of the management data tree when issued in browsing mode (Figure 4). The listed information contains both static and dynamic data of the corresponding nodes in the configuration tree. The dynamic `state` and `statistics` data in `diameter transport 0 peer 0` node is shown in Red in Figure 4. The dynamic `state` and `statistics` data in `diameter transport 1 client` node is shown in Red in Figure 5.

Normally, the dynamic data is subject of the monitoring and it can be periodically pulled over the OA&M interface. The use of JN SPU OA&M interface for monitoring is described in Section 5. An example of using Nagios console for automated monitoring over JN SPU OA&M interface is included in Section 6.

Pressing `TAB` key brings context sensitive command completion as illustrated in Figure 5.

When the command `config` is issued in CLI browsing mode then the CLI configuration mode is entered. JN SPU configuration data can be created or modified in this mode. This is described in the next Section 4.

```
> show
spu {
  users "default" {
    port 9000
  }
}
diameter {
  transport 0 {
    origin-host "spu.realm1"
    origin-realm "realm1"
    applications {
      s6ad
    }
    host-ips {
    }
    protocol tcp
    local-ip 127.0.0.1
    local-port 3868
    server {
      accept {
        127.0.0.1 10.0.0.1
      }
      peer 0 {
        destination-host
"spu.realm2"
        destination-realm
"realm2"
        state okay
        statistics {
          recv-cnt 54
          recv-max 132
          recv-avg 74
          recv-oct 4048
          recv-dvi 3
          send-cnt 54
          send-max 144
          send-avg 75
          send-oct 4072
          send-pend 0
        }
      }
    }
  }
  transport 1 {
    origin-host "spu.realm2"
    origin-realm "realm2"
    applications {
      s6ad
    }
    host-ips {
    ...
```

Figure 4. Listing management data tree with `show` command

```
> show diameter transport
0    1
> show diameter transport 1
origin-host "spu.realm2"
origin-realm "realm2"
applications {
  s6ad
}
host-ips {
}
protocol tcp
local-ip 127.0.0.1
local-port 0
client {
  remote-ip 127.0.0.1
  remote-port 3868
  destination-host
"spu.realm1"
  destination-realm "realm1"
  state okay
  statistics {
    recv-cnt 70
    recv-max 144
    recv-avg 74
    recv-oct 5244
    recv-dvi 4
    send-cnt 70
    send-max 132
    send-avg 74
    send-oct 5244
    send-pend 0
  }
}

>
```

Figure 5. Context sensitive TAB completion

4. JN SPU Configuration

JN SPU is configured over the OA&M interface using JN SPU CLI shell commands. Normally, the configuration is done manually . Alternatively, the configuration process can be automated over the same interface.

The configuration process of Diameter protocol stack is illustrated in this section. The same approach applies when configuring SS7 MAP, SS7 CAP and SIP stacks.

After the ssh connection the JN SPU CLI shell is in browsing mode. To enter the configuration mode the `config` command should be issued. The command prompt shows []> to indicate that the current node is the root of the configuration data tree. This is shown in Figure 6. The context sensitive TAB completion and `help` command are available in configuration mode too.

As the output of `help` command shows switching back to CLI shell browsing mode can be done by using the commands `commit` or `cancel`. The commit command saves and activates the configuration changes after performing some formal checks. The cancel command discards any changes made in CLI shell configuration mode.

When help command is issued at a given level in the configuration tree it lists the nodes at this level and shows some special commands that may be used for changing the descendants of the current node or for navigate up to the parent node or to the top in the tree. This is illustrated in Figure 7. The prompt changes to show the current node in the configuration data tree.

```
admin@10.99.99.11's password:
12:00 $ ssh -p 8822 admin@10.99.99.11
admin@10.99.99.11's password:
> help
Use 'show [NodeName ...]' to list the nodes and their values
Use 'config' to enter configuration mode
Use 'exit' to exit CLI shell

> config
[]> help
NodeType    NodeName ValueType
------------------------------
Property -> spu
Property -> diameter

Use 'show' to list the nodes and their values
Use 'edit <NodeName ...>' to edit an existing node of type Property
Use 'edit <NodeName ...> <Value>' to edit an existing node of type List
Use 'commit' to commit the configuration changes
Use 'cancel' to cancel the configuration changes

[]>
```

Figure 6. Entering and leaving CLI shell configuration mode

```
[diameter]> edit transport 0
[transport 0]> help
NodeType    NodeName        ValueType
--------------------------------------------
Leaf     -> origin-host     String
Leaf     -> origin-realm    String
LeafList -> applications    {s6ad | slh ...}
LeafList -> host-ips        {IpAddress ...}
Leaf     -> protocol        sctp | tcp
Leaf     -> local-ip        IpAddress
Leaf     -> local-port      Integer
Choice   -> server | client Property | Property

Use 'show' to list the nodes and their values
Use 'edit <NodeName ...>' to edit an existing node of type Property
Use 'edit <NodeName ...> <Value>' to edit an existing node of type List
Use 'add  <NodeName> <Value>' to add a new node of type List
                              to add value to node of type LeafList
Use 'del  <NodeName> <Value>' to delete an existing node of type List
                              to remove value from node of type LeafList
Use 'set <NodeName> <Value>' to set the value of type Leaf or LeafList
nodes
Use 'up[!]' to return to parent node in the configuration
Use 'top[!]' to return to root node in the configuration
Use 'commit' to commit the configuration changes
Use 'cancel' to cancel the configuration changes

[transport 0]>
```

Figure 7. Changing the values of the nodes in the configuration tree

5. JN SPU Monitoring

JN SPU is monitored over the OA&M interface using JN SPU CLI shell `show` command. Normally, the monitored data retrieved by the invocation of `show` command refer to the dynamic data in the configuration data tree and are sent periodically by some external automated process or monitoring console. Alternatively, for incidental queries, the monitored data can be retrieved manually.

The monitoring process of Diameter protocol stack is illustrated in this section. The same approach applies when monitoring SS7 MAP, SS7 CAP and SIP stacks.

The monitored data is retrieved after establishing a ssh connection with JN SPU OA&M server. A new shell is started and the `show` command with the parameters related to the required data can be executed. The monitored state is included in the `show` command response. The manual version of this process is illustrated in Figure 8.

```
12:52 $ ssh -p 8822 admin@10.99.99.11
admin@10.99.99.11's password:
> show diameter transport 0 server peer 0
destination-host "spu.realm2"
destination-realm "realm2"
state okay
statistics {
  recv-cnt 450
  recv-max 132
  recv-avg 74
  recv-oct 33388
  recv-dvi 0
  send-cnt 450
  send-max 144
  send-avg 74
  send-oct 33340
  send-pend 0
}

>
```

Figure 8. Monitoring dynamic peer state and statistics data over OA&M interface

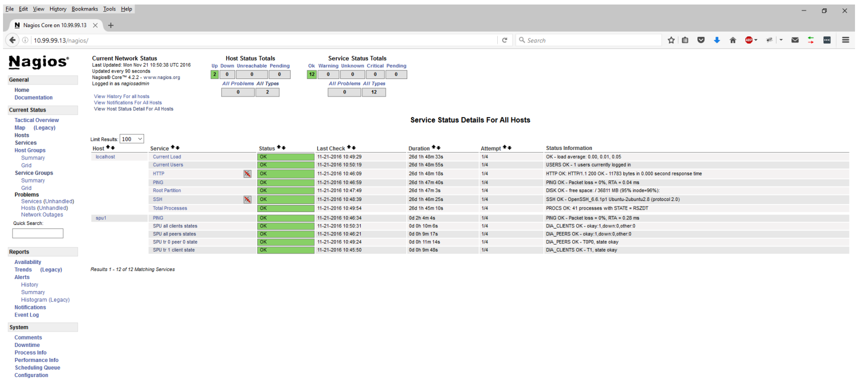
6. Automated JN SPU Monitoring, Nagios Example

Nagios Core or other Nagios flavours can be easily configured for automated JN SPU Monitoring over JN SPU OA&M interface.

Nagios Core uses plugins for retrieving the state of the monitored services from the remote hosts. Nagios plugins are external processes that can be implemented in different programming languages. The specific Nagios requirement is that a process implementing a Nagios plugin must return an exit code (0, 1, 2, 3, etc), and one ore more lines describing the monitored service state. The status of the monitored service is set to OK, WARNING, CRITICAL, UNKNOWN, depending on the exit code of the plugin.

JN SPU is delivered with `check_spu.pl` plugin implemented in Perl. When invoked by Nagios process, the plugin establishes a ssh session based on a public/private key pair, executes the corresponding command in JN SPU CLI shell, parses the result and returns the corresponding exit code and service status description lines.

Nagios console executing check_spu.pl plugin is shown in Figure 9. Similarly, other monitoring console may be used for automating the JN SPU monitoring over its OA&M interface.



Current Network Status
Last Updated: Mon Nov 21 10:50:38 UTC 2016
Updated every 90 seconds
Nagios® Core™ 4.2.2 - www.nagios.org
Logged in as *nagiosadmin*

Host Status Totals

Up	Down	Unreachable	Pending
2	0	0	0

All Problems	All Types
0	2

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	0	0	0	0

All Problems	All Types
0	12

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	11-21-2016 10:49:29	26d 1h 48m 33s	1/4	OK - load average: 0.00, 0.01, 0.05
	Current Users	OK	11-21-2016 10:50:19	26d 1h 48m 55s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	11-21-2016 10:46:09	26d 1h 48m 18s	1/4	HTTP OK: HTTP/1.1 200 OK - 11783 bytes in 0.000 second response time
	PING	OK	11-21-2016 10:46:59	26d 1h 47m 40s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	11-21-2016 10:47:49	26d 1h 47m 3s	1/4	DISK OK - free space: / 36811 MB (95% inode=96%):
	SSH	OK	11-21-2016 10:48:39	26d 1h 46m 25s	1/4	SSH OK - OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 (protocol 2.0)
	Total Processes	OK	11-21-2016 10:49:54	26d 1h 45m 10s	1/4	PROCS OK: 41 processes with STATE = RSZDT
spu1	PING	OK	11-21-2016 10:46:34	0d 2h 4m 4s	1/4	PING OK - Packet loss = 0%, RTA = 0.28 ms
	SPU all clients states	OK	11-21-2016 10:50:31	0d 0h 10m 6s	1/4	DIA_CLIENTS OK - okay:1,down:0,other:0
	SPU all peers states	OK	11-21-2016 10:46:21	0d 0h 9m 17s	1/4	DIA_PEERS OK - okay:1,down:0,other:0
	SPU tr 0 peer 0 state	OK	11-21-2016 10:49:24	0d 0h 11m 14s	1/4	DIA_PEERS OK - T0P0, state okay
	SPU tr 1 client state	OK	11-21-2016 10:45:50	0d 0h 9m 48s	1/4	DIA_CLIENTS OK - T1, state okay

Results 1 - 12 of 12 Matching Services

Figure 9. Automated JN SPU monitoring with Nagios

About Applicata

www.applicata.bg

Applicata specialises in the design, development, installation and integration of systems and software for law enforcement and telecommunications. Applicata team guarantees that top quality products and services will be delivered within tough deadlines and budgets. Applicata is well known by its competitive advantages, including low cost, flexibility in covering specific customer needs, on time delivery and extended support.

Sales Europe
Ashwood, Woodcote
Guildford, Surrey, GU2 4HQ, UK
Tel: +44 1483 506384
Fax: +44 1483 506384
sales@applicata.bg
www.applicata.bg

Sales Asia
10F, No. 1, Lane 25, XinSheng S. Rd., Sec. 3
Taipei 10660, Taiwan R.O.C.
Tel: +886 2 8369 5685
Fax: +886 2 8369 5685
sales@applicata.bg
www.applicata.bg

Applicata
6, Angista Str
1527 Sofia, Bulgaria
Phone: +359 2 981 4203
Fax: +359 2 943 4719
office@applicata.bg
www.applicata.bg

† In the roadmap