sysrepo

NETCONF/YANG for any UNIX/Linux application

YANG model:
```
module ietf-interfaces {
    container interfaces {
        description "Interface configuration ";
        list interface {
            key "name";
            leaf name {
                type string;
            }

            leaf description {
                type string;
            }
            ...
```

NETCONF Request:
```
<rpc message-id="101" >
    <get-config>
        <source>
            <running/>
        </source>
    </get-config>
</rpc>
```

Data Files | YANG
File system

| NETCONF Client | NETCONF / SSH/TLS | NETOPEER2 NETCONF Server | GPB / UDS | Sysrepo Engine | GPB / UDS | Application |

SYSREPO Client library | SYSREPO Client library

SYSREPO

NETCONF Reply:
```
<rpc-reply>
  <nc:data>
    <interfaces>
      <interface>
        <name>eth0</name>
        <description>Ethernet 0</description>
        <type>ethernetCsmacd</type>
        <enabled>true</enabled>
      </interface>
    </interfaces>
  </nc:data>
</rpc-reply>
```
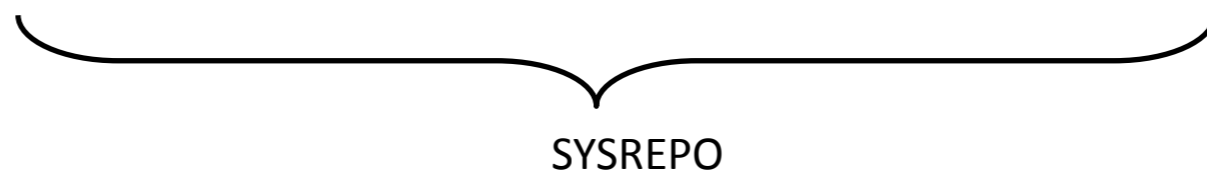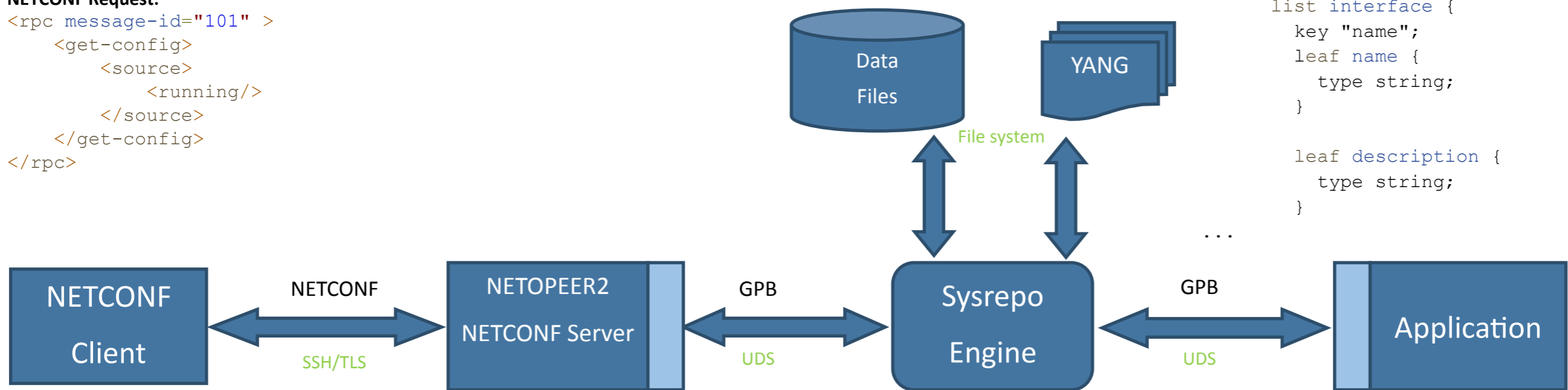
Netopeer2 Server

- open-source NETCONF (RFC 6241) server
- developed by CESNET
- based on libnetconf2 library
- uses Sysrepo as its primary datastore
- transforms NETCONF requests into Sysrepo API calls and data returned from Sysrepo into NETCONF replies

Sysrepo Datastore

- open-source datastore for configuration and state data developed by Cisco
- stored data is modelled by YANG
- provides XPath-based API for accessing the data from north-bound and south-bound application in C language + bindings for other languages (Python, ...)
- startup, running and candidate data store support

Other Tools

- libyang - YANG data modeling language library
- Sysrepoctl - install and uninstall YANG for Sysrepo
- Sysrepocfg - command line tool for administrator to alter the configuration in text editor
- Testconf - tool for automated testing
- YANG-creator - online YANG creator, editor and exporter

Sysrepo API:

```
// CONNECTION & SESSION MANAGEMENT
/* connect to sysrepo */
rc = sr_connect("application-name", SR_CONN_DEFAULT, &connection);
/* start a session */
rc = sr_session_start(connection, SR_DS_STARTUP, SR_SESS_DEFAULT,
&session);

/* stop a session */
rc = sr_session_stop(session);

/* disconnect */
sr_disconnect(connection);


// DATA RETRIEVAL
/* read a data element */
rc = sr_get_item(session,
    "/ietf-interfaces:interfaces/interface[name='eth0']/description",
&value);

/* read multiple data elements in a batch */
rc = sr_get_items(session,
    "/ietf-interfaces:interfaces/interface[name='eth0']/*", &values,
&count);
for (size_t i = 0; i < count; i++) {
    ...
```

```
/* read multiple data elements iteratively */
rc = sr_get_items_iter(session, "/ietf-interfaces:interfaces//*",
&iterator);
while (SR_ERR_OK == sr_get_item_next(session, iterator, &value)) {
    ...
}


// DATA MODIFICATION
/* create / set data elements */
value.type = SR_UINT8_T;
value.data.uint8_val = 64;
char *xpath = "/ietf-interfaces:interfaces/interface[name='eth0']"
    "/ietf-ip:ipv6/address[ip='fe80::ab8']/prefix-length";
rc = sr_set_item(session, xpath, &value, SR_EDIT_DEFAULT);

/* delete data elements */
rc = sr_delete_item(session, "/ietf-interfaces:interfaces/interface
[name='eth0']"
    "/ietf-ip:ipv6/address[ip='fe80::ab8']", SR_EDIT_DEFAULT);

/* commit the changes */
rc = sr_commit(session);
```

Hackathon goals:

1, NETCONF management of DHCPv6 in ISC KEA

2, NETCONF management of Raspberry PI

Contact:

Project site and documentation	—> sysrepo.org
Browse code or report an issue	—> github.com/sysrepo/sysrepo github.com/CESNET/Netopeer2
Ask a question	—> lists.sysrepo.org/listinfo

CESNET | CISCO | Deutsche Telekom | PANTHEON technologies | sartura