Mumbai Technology Meetup Feb-2020

UCI (Universal Configuration Interface) *for* IoT Device Management

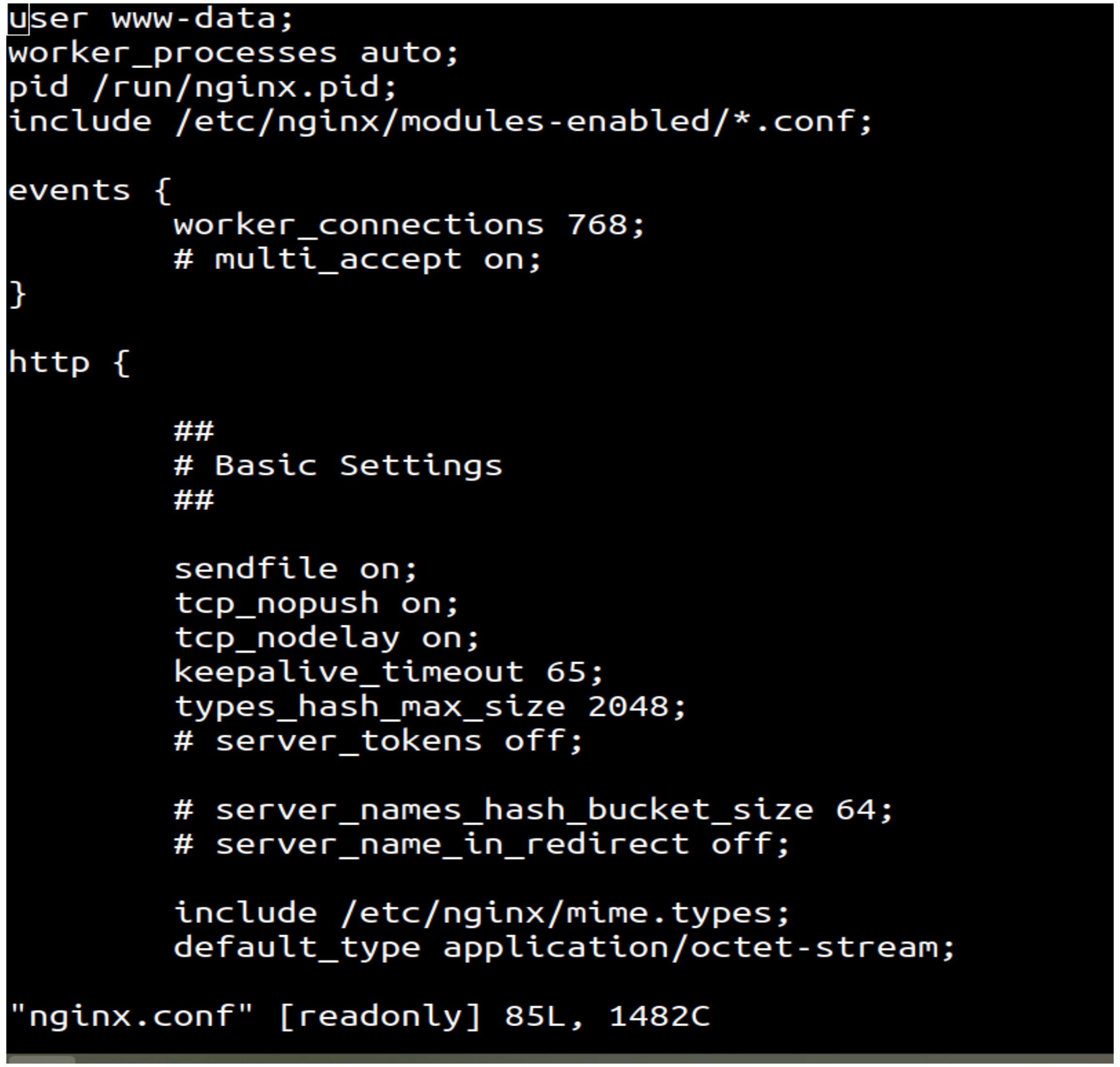
Siji Sunny | AntsWireless

siji@melabs.in

OVERVIEW

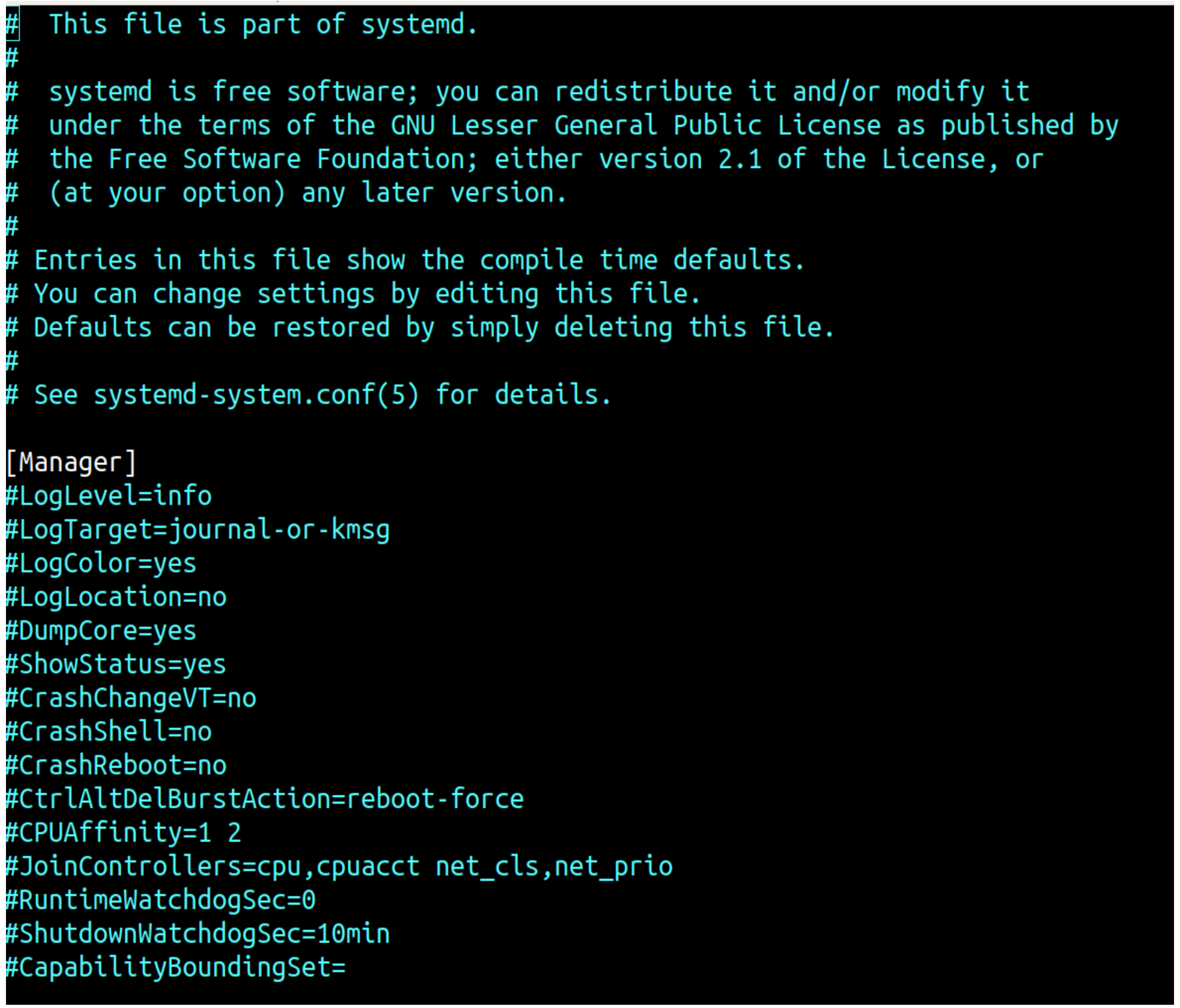
- Challenges in Configuration Management
- Proposed Approach- UCI/NetJson
- UCI – Overview
- NetJson Overview
- Demo

THE CASE OF NON-STANDARD



```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

"nginx.conf" [readonly] 85L, 1482C
```

Nginx Conf



```
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it
#  under the terms of the GNU Lesser General Public License as published by
#  the Free Software Foundation; either version 2.1 of the License, or
#  (at your option) any later version.
#
# Entries in this file show the compile time defaults.
# You can change settings by editing this file.
# Defaults can be restored by simply deleting this file.
#
# See systemd-system.conf(5) for details.

[Manager]
#LogLevel=info
#LogTarget=journal-or-kmsg
#LogColor=yes
#LogLocation=no
#DumpCore=yes
#ShowStatus=yes
#CrashChangeVT=no
#CrashShell=no
#CrashReboot=no
#CtrlAltDelBurstAction=reboot-force
#CPUAffinity=1 2
#JoinControllers=cpu,cpuacct net_cls,net_prio
#RuntimeWatchdogSec=0
#ShutdownWatchdogSec=10min
#CapabilityBoundingSet=
```

Systemd Conf

NEED OF STANDARDISATION

- No need to write your own parser. That has been done already in many programming languages and can easily be used. Also the parser probably is faster than a self-written one and is a bit more robust.
- The formats have various extra features like comments, hierarchy, native representation of data types. These might not seem to be required now, but it is good if the format has some complexity in reserve in case it is needed.
- Text editors also know about these formats and can provide syntax highlighting and indentation.
- The configuration file can be parsed with different programs. The interpretation of the values of course differs, but an analysis program can simply read the parameters of the simulation without being told.

UCI -UNIFIED CONFIGURATION INTERFACE

- A small utility written in C
- Part Of OpenWrt -Linux Operating System for Light Weight Embedded devices
- UCI - system to centralize the configuration of OpenWrt service
- Central configuration is split into several files located in the /etc/config/ directory.
- Supports various method of execution
  - SSH
  - CLI
  - SMS
  - JSON-PRC

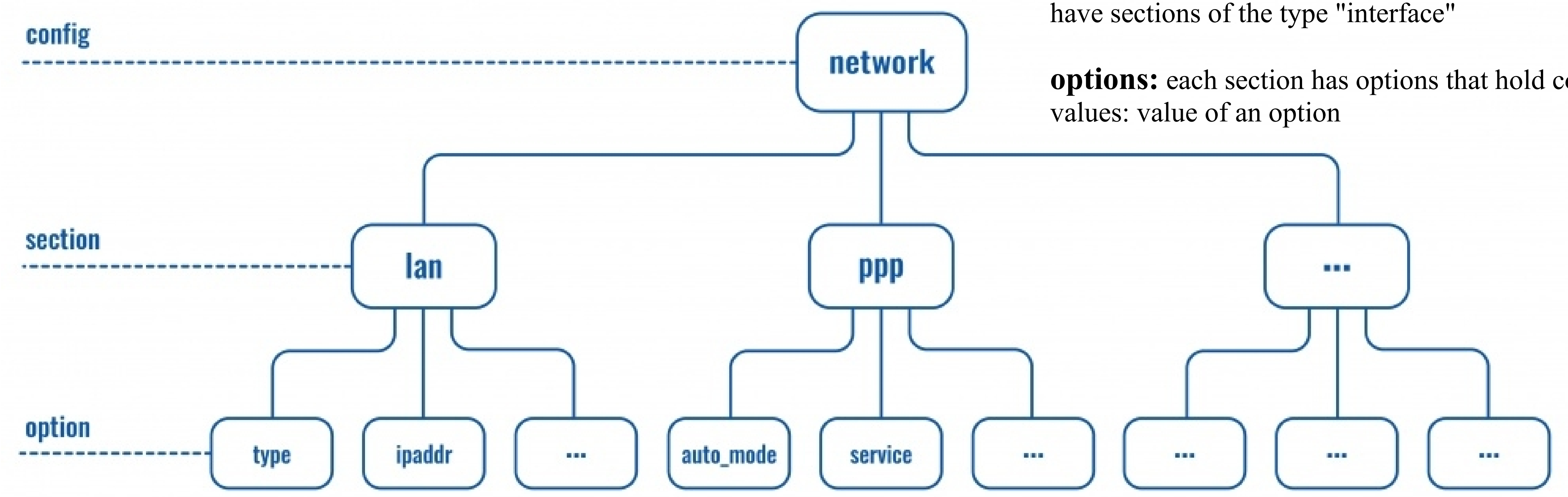
CONFIGURATION HIERARCHY

config: main configuration groups like network, system, firewall. Each configuration group has it's own file in /etc/config

sections: a config is divided into sections. A section can either be named or unnamed

types: a section can have a type. E.g, in the network config we typically have sections of the type "interface"

options: each section has options that hold configuration values
values: value of an option



SAMPLE CONFIG

```
# cat /etc/config/network

config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config interface 'lan'
        option type 'bridge'
        option ifname 'eth0.1'
        option proto 'static'
        option netmask '255.255.255.0'
        option ip6assign '60'
        option ipaddr '192.168.1.1'
```

UCI COMMANDS

batch	-	Executes a multi-line UCI script which is typically wrapped into a here document syntax
export	[<config>]	Exports the configuration in a machine readable format. It is used internally to evaluate configuration files as shell scripts
import	[<config>]	Imports configuration files in UCI syntax
changes	[<config>]	Lists staged changes to the given configuration file or if none given, all configuration files
commit	[<config>]	Writes changes of the given configuration file, or if none is given, all configuration files, to the filesystem. All "uci set", "uci add", "uci rename" and "uci delete" commands are staged into a temporary location until they are written to flash with the "uci commit" command. This is used exclusively for UCI commands and is not needed after editing configuration files with a text editor
add	<config> <section-type>	Adds an anonymous section of type section-type to the given configuration
add_list	<config>.<section>.<option>=<string>	Adds the given string to an existing list option
del_list	<config>.<section>.<option>=<string>	Removes the given string from an existing list option

UCI COMMANDS

show	[<config>[.<section>[.<option>]]]	Shows the given option, section or configuration in compressed notation. If no option is given, shows all configuration files
get	<config>.<section>[.<option>]	Gets the value of the given option or the type of the given section
set	<config>.<section>[.<option>]=<value >	Sets the value of the given option, or add a new section with the type set to the given value
delete	<config>[.<section>[[.<option>][=<id>]]]	Deletes the given section or option
rename	<config>.<section>[.<option>]=<nam e>	Renames the given option or section to the given name
revert	<config>[.<section>[.<option>]]	Reverts the given option, section or configuration file. Used to undo any changes performed with UCI and not yet committed with uci commit
reorder	<config>.<section>=<position>	Moves the specified section to the given position. Used for easier management purposes

Net{JSON}

netjsonconfig

http://netjson.org/

- Data Interchange format designed to describe the basic building blocks of layer2 and layer3 networks, but easy to expand to other configuration types too
  - Network configuration of devices
  - Monitoring data
  - Routing information
  - Network topology
- Netjson definition - http://netjson.org/rfc.html

NETJSONCONFIG -PYTHON TOOL

- netjsonconfig is a python implementation of the NetJSON data interchange format, more specifically the DeviceConfiguration object.
- Supports terminal based attributes
- Install
  - pip install netjsonconfig
- Create UCI rendered file -
  - netjsonconfig --config sample.json --backend openwrt --method generate > config.tar.gz

CONTACT

Email : siji@melabs.in
Twitter : siji_sunny
Blog - https://www.cnx-software.com/

Phone : +91 8080125555