Network configuration management at CERN

Stefan Stancu
Arkadiy Shevrikuko

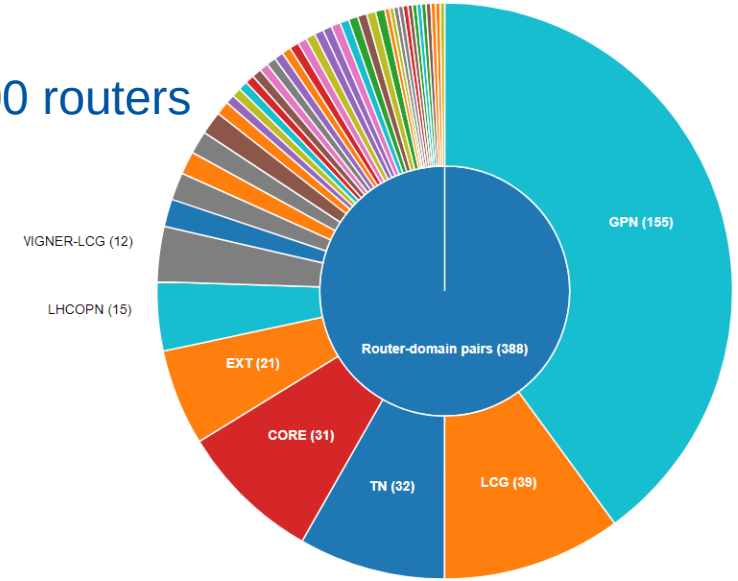
Outline

- Problem scale
- Current solution: cfmgr
- Evolution
  - motivation
  - plan

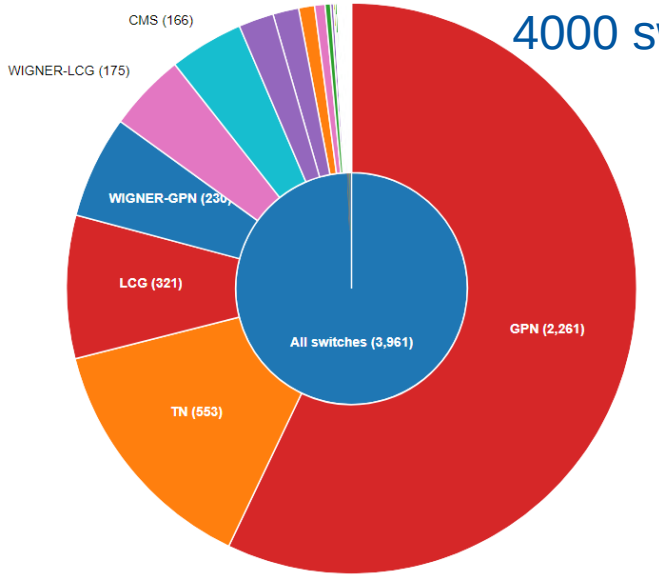
Network at CERN

- Around 4,4 k network devices
- Multi-vendor network (CERN procures equipment with open call for tenders)
- Specific configuration for each environment

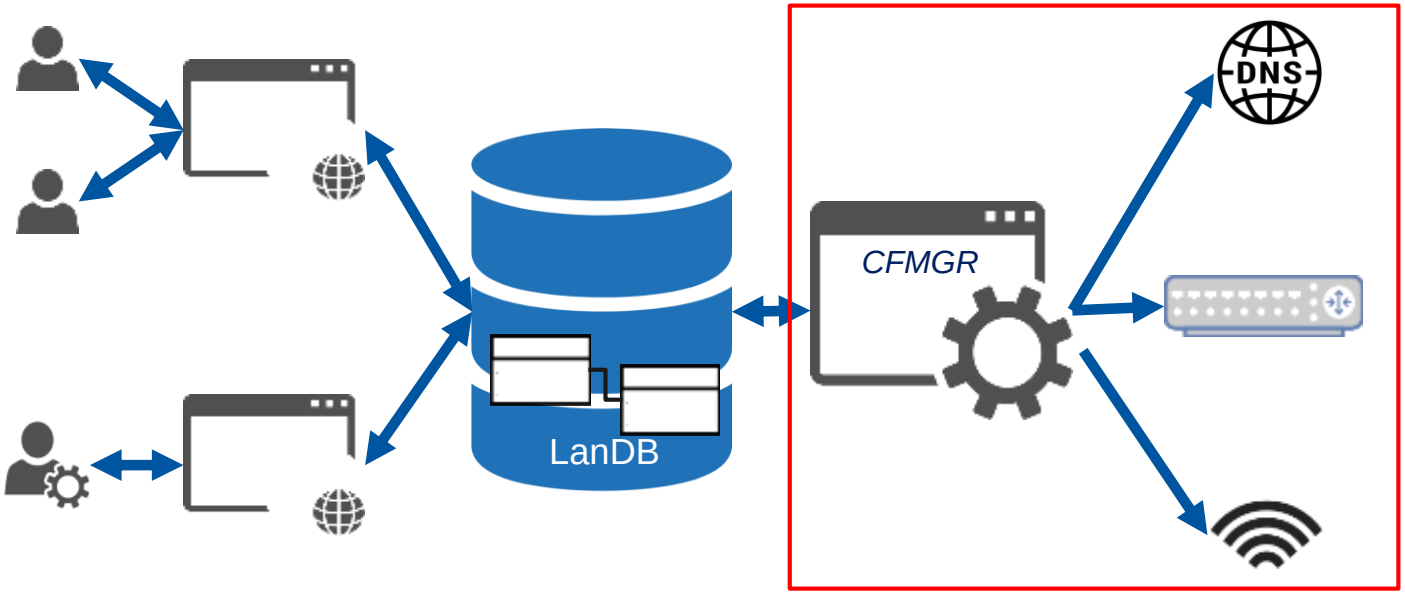
400 routers



4000 switches



Network configuration automation workflow at CERN

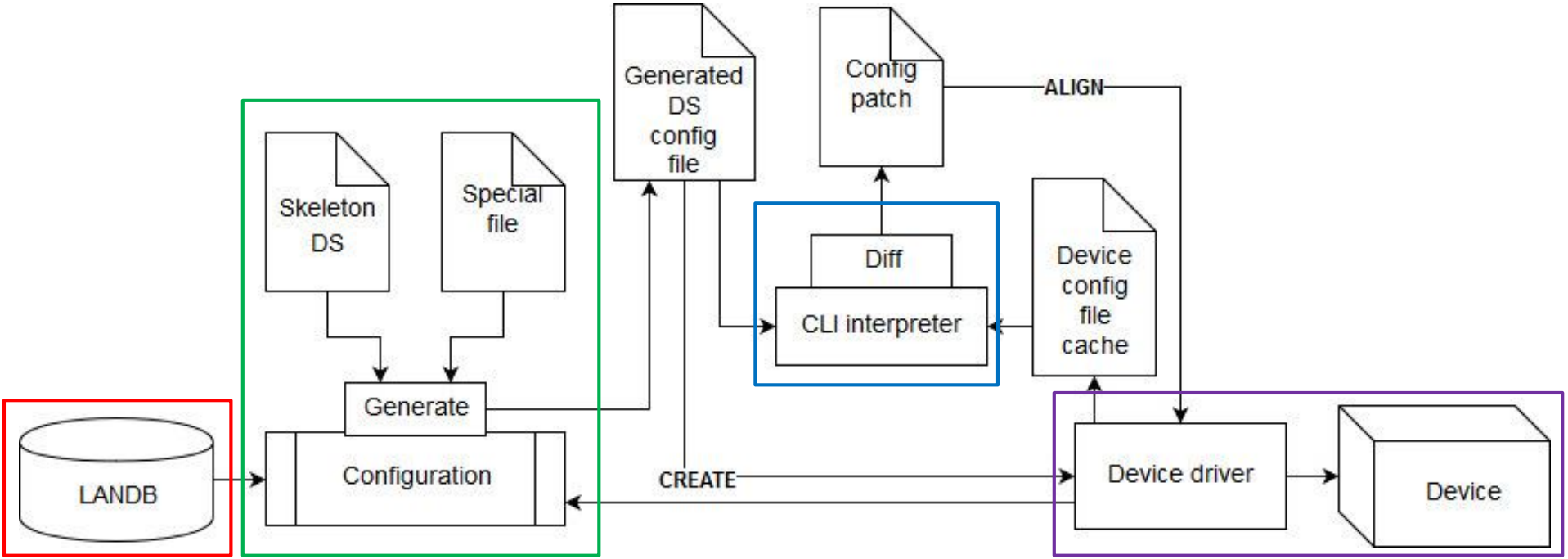


CFMGR – configuration management tool

What is cfmgr?

- Perl-based command-line tool
- Helps to ensure consistent configuration for the network
  - Uses central DB as a source of truth
- Features:
  - Multi-vendor support
  - Supports multiple operations
    - Basic configuration management (load, align)
    - Automatic configuration of ACLs and PoE
    - Large scale firmware management
    - etc…
  - Integration with Spectrum CA
    - Sends cron job status notifications

CFMGR architecture

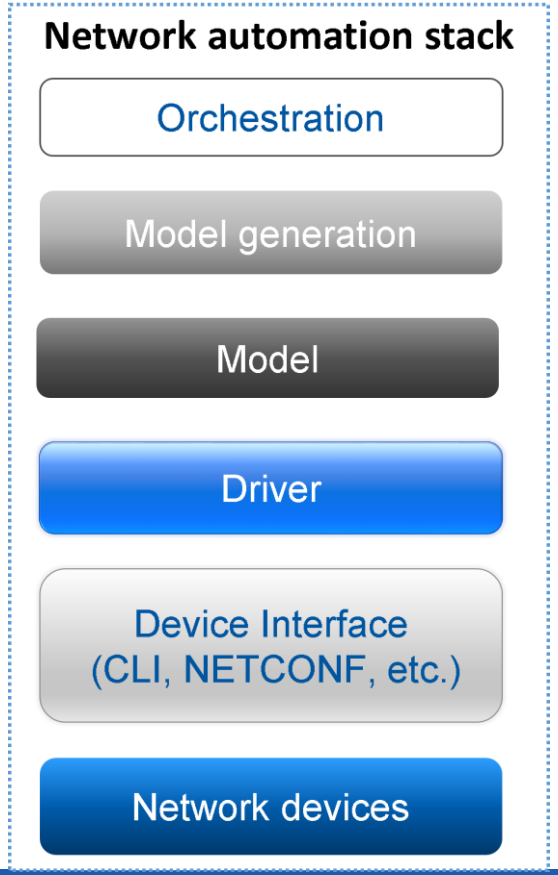


Evolution motivation

- Need to support a new router platform
  - different configuration workflow
- Decrease in popularity of Perl
  - Most network automation libraries use Python
- Large code base with non-uniform coding practices
- No clear separation between modules

Network configuration platform evolution

Architecture



Not covered by any automation platform. Should be developed

N.A.P.A.L.M.

Python libraries, provided by vendors

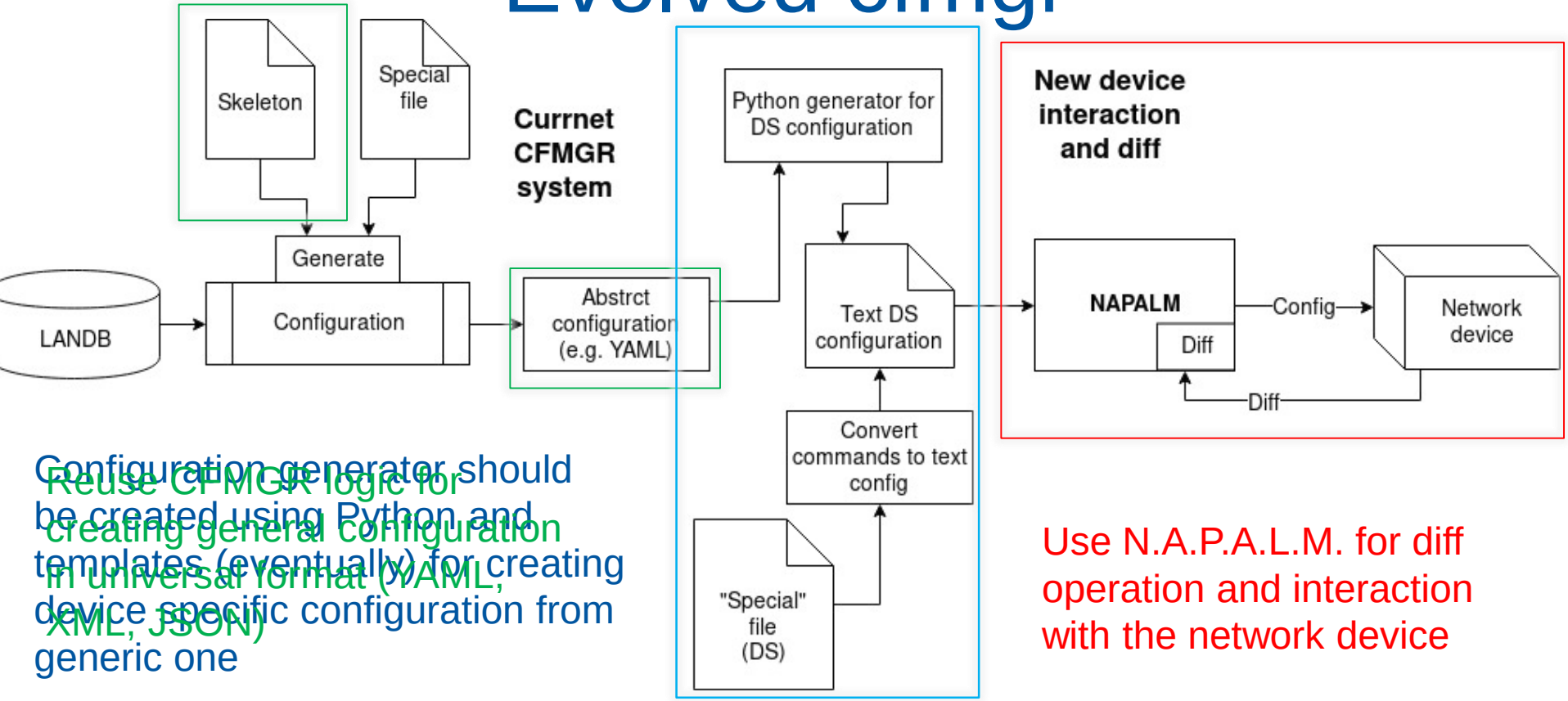
N.A.P.A.L.M.

- N.A.P.A.L.M. (Network Automation and Programmability Abstraction Layer with Multivendor support)
  - Python library that implements a set of functions to interact with different network device Operating Systems using a unified API.
- N.A.P.A.L.M. API covers two aspects:
  - Configuration management

Configuration support matrix

_	EOS	JunOS	IOS-XR	FortiOS	NXOS	IOS	Pluribus	PANOS	MikroTik	VyOS
Config. replace	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Config. merge	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Compare config	Yes	Yes	Yes [2]	Yes [2]	Yes [5]	Yes	No	Yes	No	Yes
Atomic Changes	Yes	Yes	Yes	No [3]	Yes/No [6]	Yes	Yes	Yes/No [6]	No	Yes
Rollback	Yes [3]	Yes	Yes	Yes	Yes/No [6]	Yes	No	Yes	No	Yes

Evolved cfmgr



Configuration generator should be created using Python and templates (eventually) for creating device specific configuration from generic one

Reuse CFMGR logic for creating general configuration in universal format (YAML, XML, JSON)

Use N.A.P.A.L.M. for diff operation and interaction with the network device

Evolution plan

- Decouple concerns:
  - Configuration generation (vendor independent model)
  - Translation to vendor specific configuration
  - Configuration enforcement (N.A.P.A.L.M. and vendor libraries Python)

	Phase I	Phase II
Configuration generation	Perl	Python
Translation	Python	Python
Enforcement	Python	Python

Summary

- Network configuration automation is a must
  - Due to the scale and diversity of CERN's networks
- Today there is no commercial or open-source product, capable of replacing cfmgr
- We are evolving the tool in order to leverage open-source tools with multi-vendor support:
  - Faster and easier integration of the new vendors
  - Easier maintenance of the tool
  - Faster implementation of new features
- New calls for tender will most likely require programmatic configuration APIs