Controlling Network devices with Python and YDK

SDN and Network Programmability Meetup in Barcelona

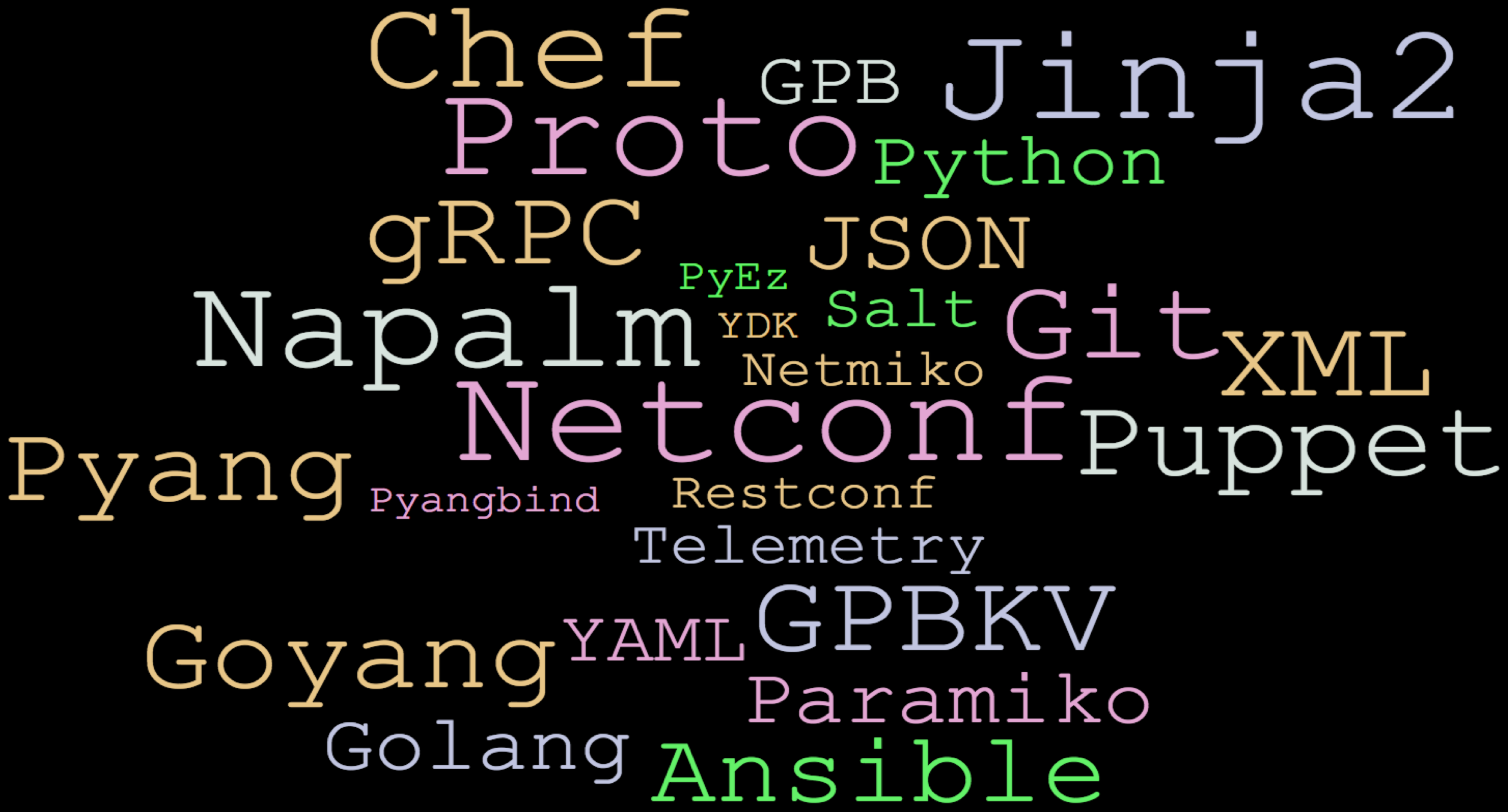
Camilo Cardona

https://github.com/jccardonar/SDNMeetupYDK

Network automation
How to start?

- Start with small tasks and progress from there.
- Learn to control network devices might be a good first step.

Too many options?



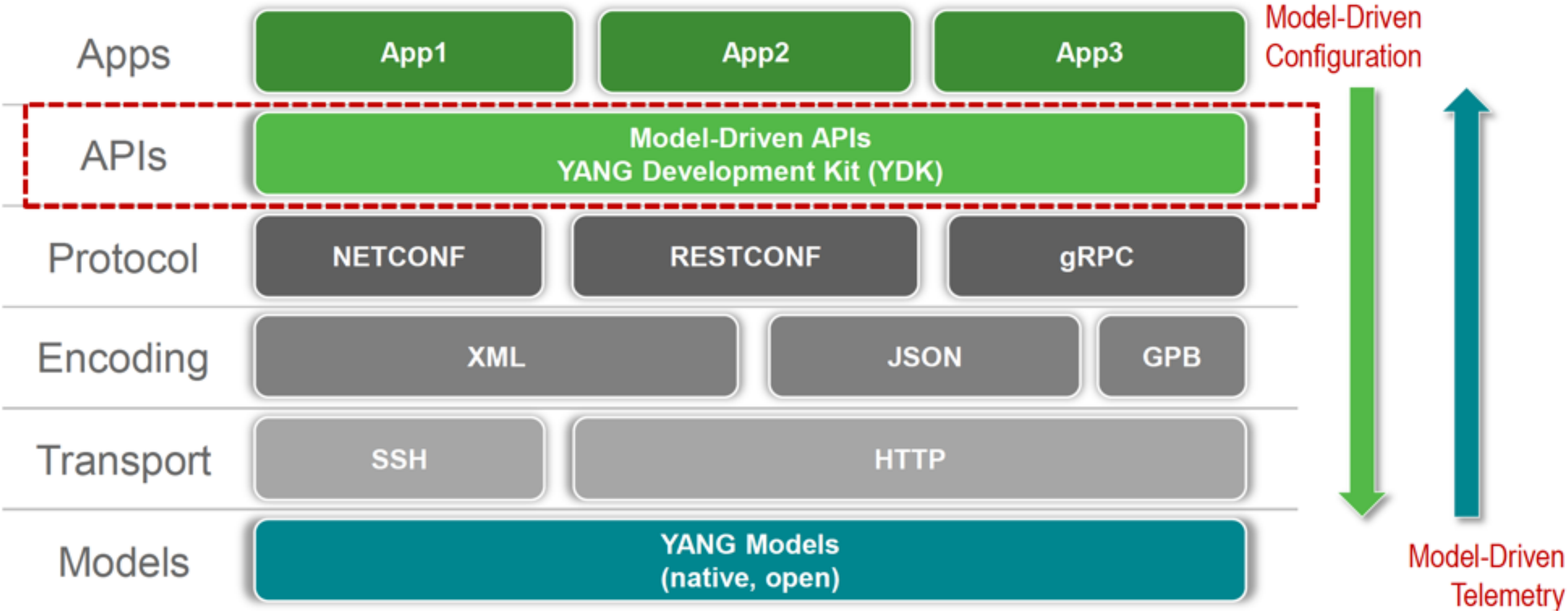
Generated using https://worditout.com

Some options for configuring devices from scripts

- NAPALM:
  - Provides standard Python APIs for Reading and writing configuration
  - Supports multiple vendors
  - Continuous development
- Yang Development Kit (YDK):
  - Good alternative if device supports Model driven configuration
  - Supports multiple languages
- Ansible
- CLI and SSH:
  - Only if other options are not available

Model-driven configuration

- Historically, every vendor implemented their own methods for configuring devices
- The IETF proposed a structured and standard way of configuring devices
- YANG was selected as the language to define models
    - For the other elements, we have different options



Apps: App1, App2, App3

APIs: Model-Driven APIs
YANG Development Kit (YDK)

Protocol: NETCONF, RESTCONF, gRPC

Encoding: XML, JSON, GPB

Transport: SSH, HTTP

Models: YANG Models
(native, open)

Model-Driven
Configuration

Model-Driven
Telemetry

YDK

- A suite of libraries in various languages that facilitates Model-driven configuration.
  - Bind models to language programming objects.
  - Takes care of transport, protocol, and encoding.
  - Validates values locally.
- See https://www.cisco.com/c/dam/m/en_us/service-provider/ciscoknowledgenetwork/files/594_06_07-16-saalvare-md-api-ckn-jun-2016-v2.pdf

Workshop outline

- The session consists of different parts, each one aiming at configuring a different aspect of a router:
  - Hostname
  - Interfaces
  - Static Route
  - ISIS
  - OSPF
  - BGP
- The process is repetitive. This is the point of this workshop:
  - Familiarize participants with the model-driven configuration process.
  - Learn that the gist of the process are the models.
  - Encourage participants to configure use new models, if needed.

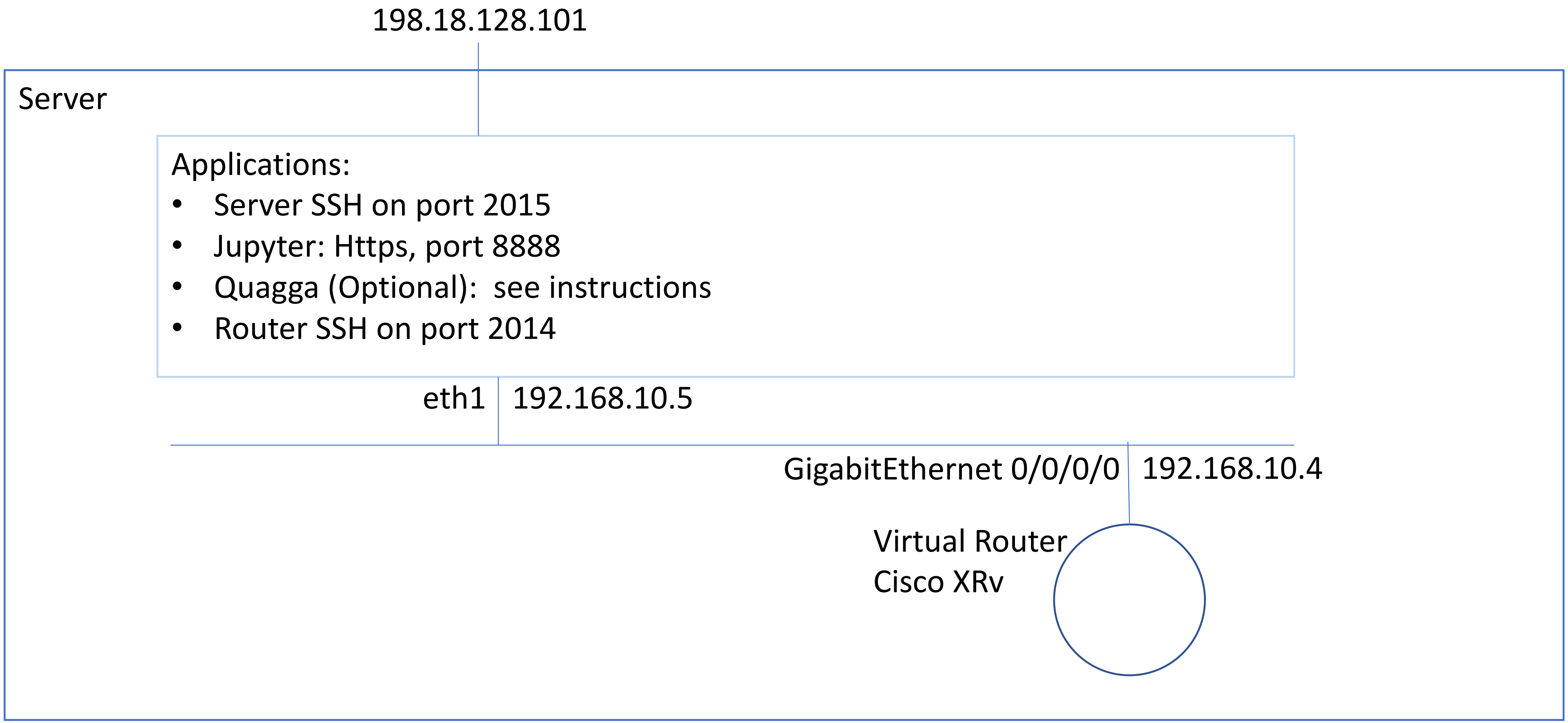
Connecting to the lab
Anyconnect (preferable)

- If not configured, install Anyconnect.
- Username and password to the setup are provided by the presenter.
- If Anyconnect is not available, try the web based Anyconnect (Not tested).
- After connecting:
  - Enter the server using:
    - ssh -p 2015 vagrant@198.18.128.101
  - cd /home/vagrant/local_notebook
  - Add nameserver 198.18.133.1, to /etc/resolv.conf
  - git clone https://github.com/jccardonar/SDNMeetupYDK.git
  - Enter http://198.18.128.101:8888/tree in a browser.

Connecting to the lab
Remote desktop

- We’ll use remote desktop to get access to environment.
- Wait for assistance from presenter to access setup.

Lab



198.18.128.101

Server

Applications:
- Server SSH on port 2015
- Jupyter: Https, port 8888
- Quagga (Optional): see instructions
- Router SSH on port 2014

eth1 192.168.10.5

GigabitEthernet 0/0/0/0 192.168.10.4

Virtual Router
Cisco XRv

How to start using a new model

- Configure it manually on the router. Read it using YDK. Follow the structure
- Use an existing example:
  - https://github.com/CiscoDevNet/ydk-py-samples
- Use the bind object and the model as guide.

Other useful tools

- https://yangcatalog.org/
- Pyang to check modules and summarize them
- https://github.com/ncclient/ncclient (Netconf in python)
- https://github.com/CiscoDevNet/ncc
    - Netconf in python and command line tool.