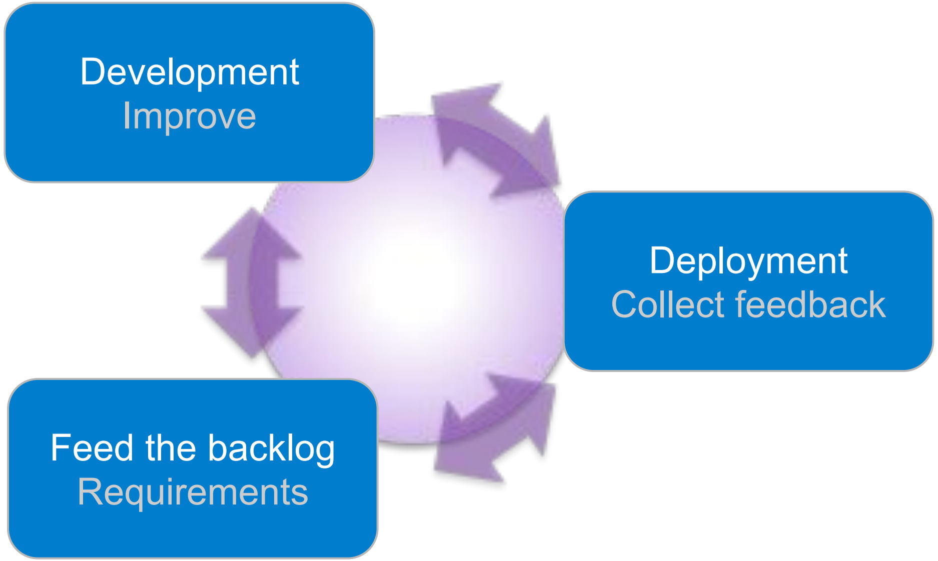
ONF
Open Networking Foundation

The new CORD platform and its deployments

December 5th, 2018

Deployment lifecycle



Development
Improve

Deployment
Collect feedback

Feed the backlog
Requirements

Review of the previous CORD platform

- Most of the software built at “deploy time”
- Requires physical machines or nested virtualization
- OS dependent (Ubuntu 16.04)
- Offline installations almost impossible
- What the old build system used to install
  - OS + configuration
  - ONOS
  - XOS
  - Profiles

Review of the previous CORD platform

- User decides what profile to install at the beginning of the installation and stick with it (until the next, full installation)
- Different setup procedures for different deployment environments
  - CORD in a Box
  - Physical POD
- Difficult to replace default services provided with the platform
  - How do I use my own DNS?
  - How do I remove MAAS from the system?
- Many custom scripts, masking standard tools (i.e. make, ansible)
- Unique development lifecycle for many components

New CORD platform

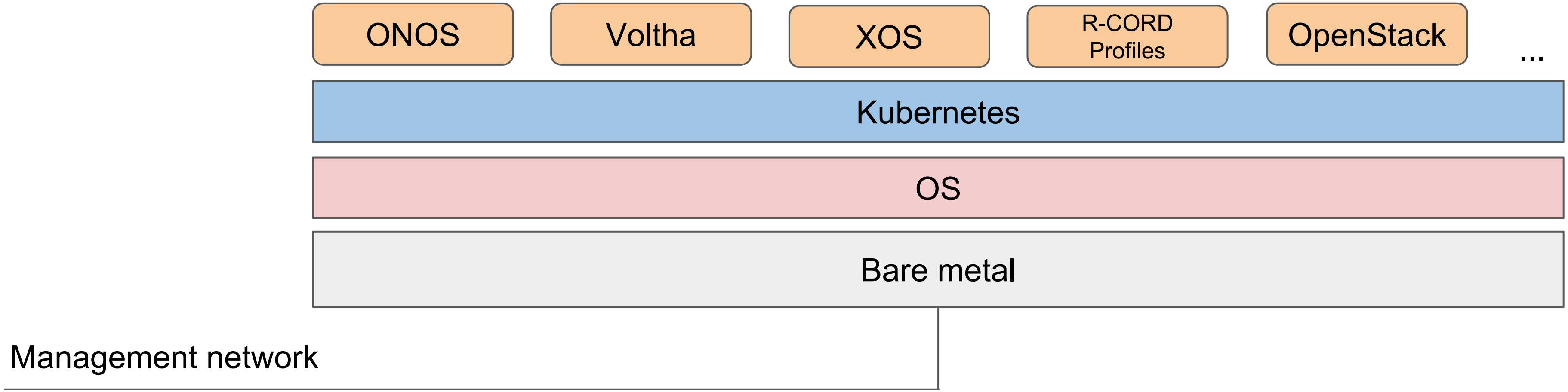
CORD control software as a set of Docker containers to be deployed on Kubernetes, through standard tools.

- Features
  - Control software (and soon VNFs) running as a Container
  - Users can choose OS and how to configure it
  - Modularity (i.e. full CORD, ONOS only, Voltha only, ...)
  - Operators: deploy (also offline) pre-built images
  - Developers: build containers, separately. Release, separately

- Requirements
  - Any machine matching min requirements (physical, virtual)
  - Any Kubernetes (> 1.9)

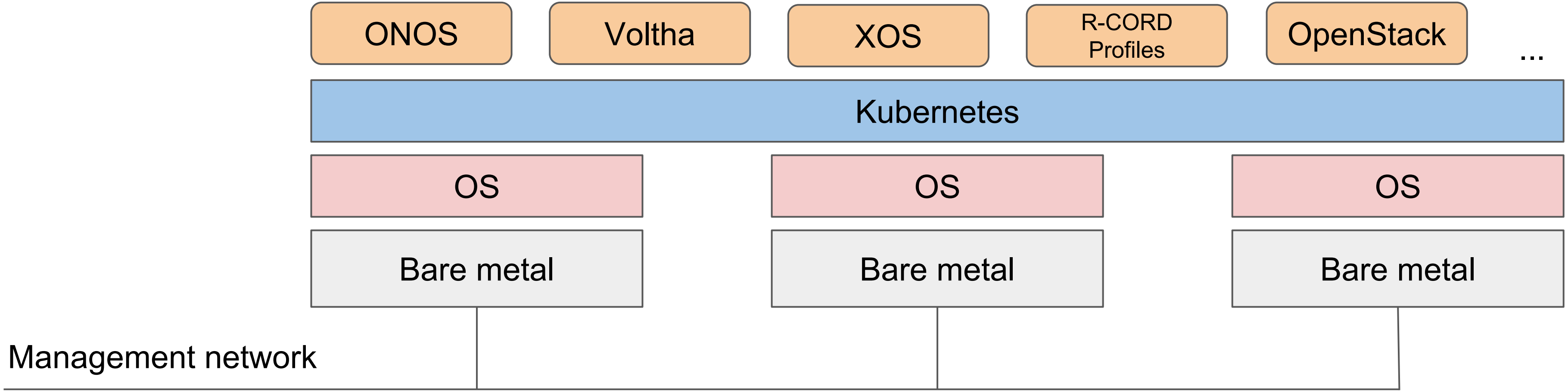
New CORD platform

Single-node deployment

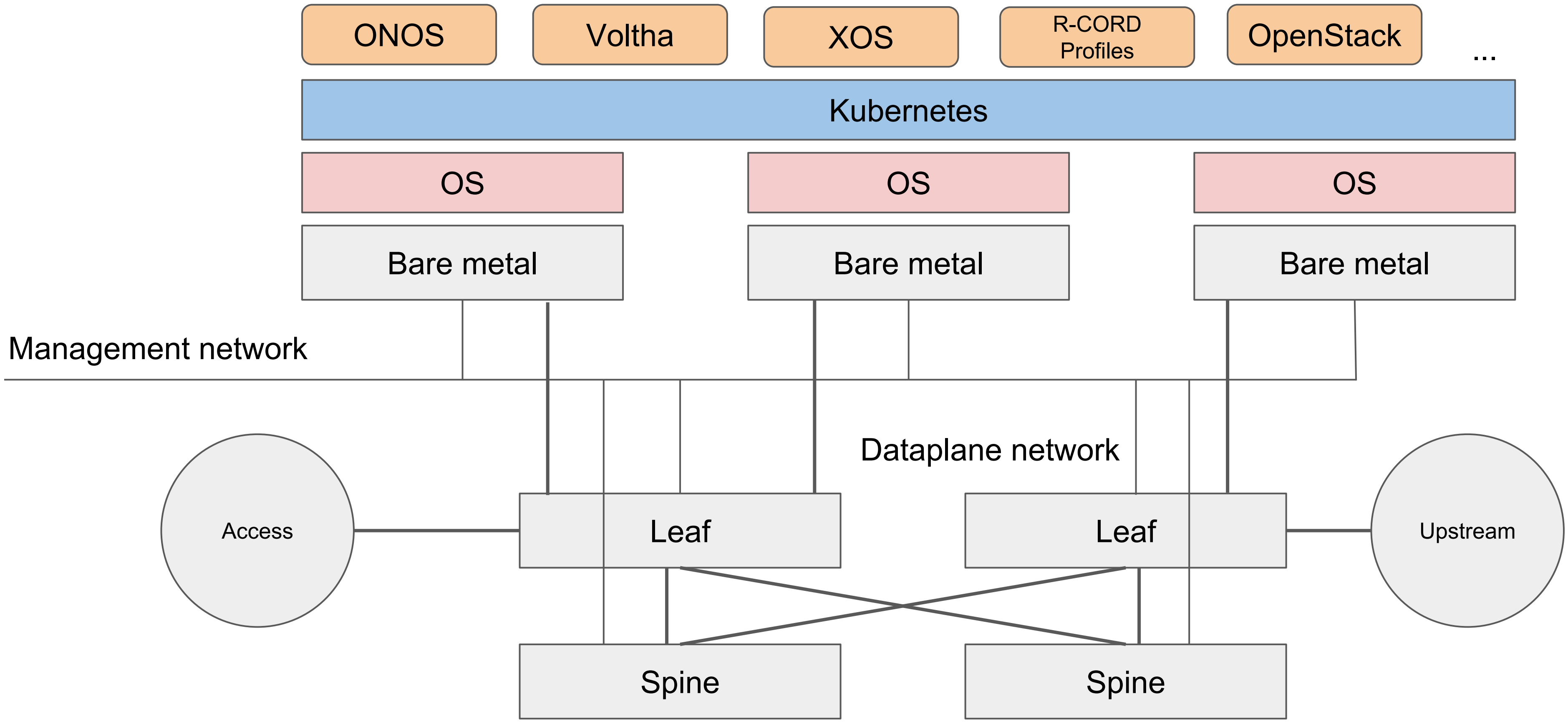


New CORD platform

Cluster deployment



New CORD platform



Deploy CORD

- Match the minimum hardware requirements
- Install and configure **your favorite** (scripts provided for lab activities)
  - **OS**
  - **Kubernetes**
- Use **Helm** to modularly deploy CORD components (as containers)
  - “helm charts“ available at *https://gerrit.opencord.org/helm-charts*
  - Examples:
    - *helm install onos*
    - *helm install voltha*
  - **Offline deployments**? Use a local Docker registry!
- Need to **change profile**?
  - *helm delete rcord-lite*
  - *helm install mcord*

Test CORD

- Primary goal: test control software and perform e2e
    - Does not require the the whole infrastructure reinstallation
    - The test process assumes that the underlay (Kubernetes) is installed and works

- **Workflow**
    - Install (once) any Kubernetes
    - Remove containers that have changed from the previous deployment
    - Build new containers only
    - Deploy new containers
    - Execute tests

- Easy to integrate with more complex CI processes

Develop CORD

- ~ one repository, one Docker container ~

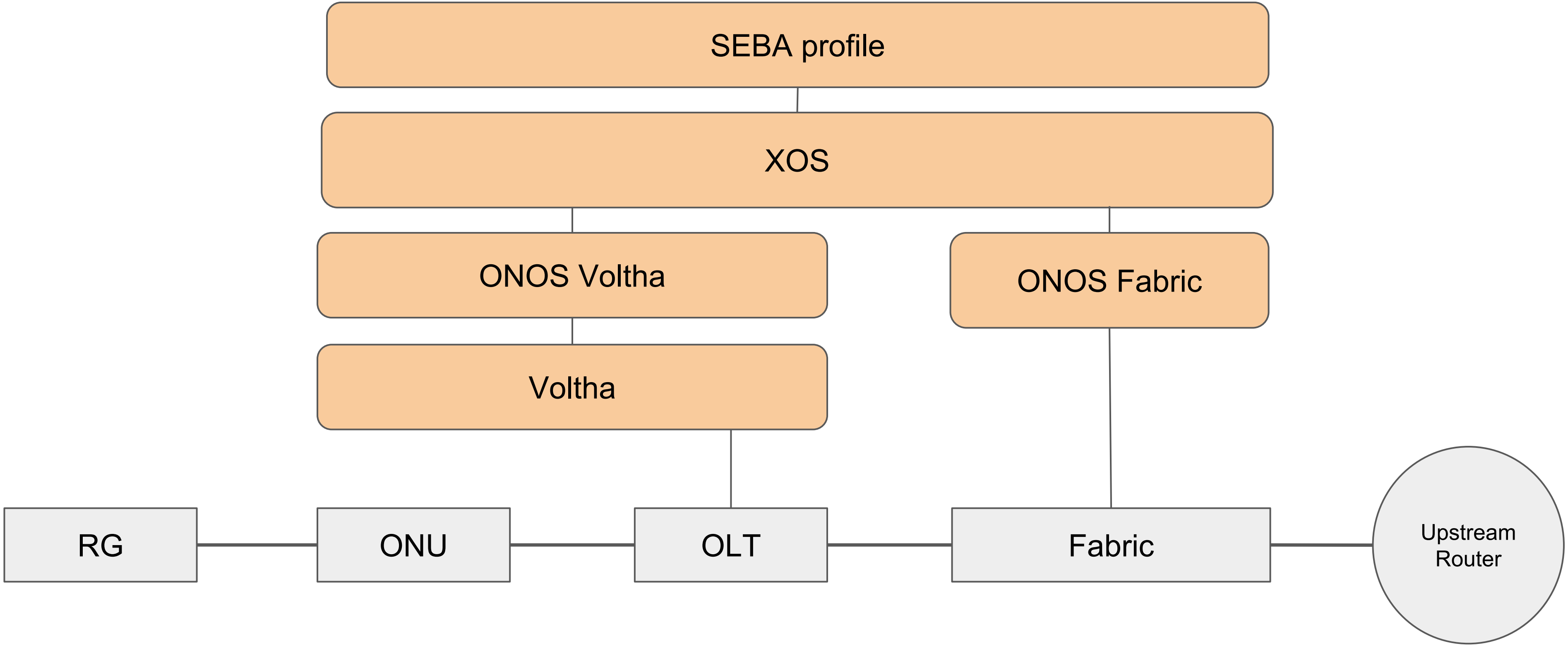
- **Workflow**
  - Download (one or more repositories)
  - Modify
  - Build
  - (tag)
  - Push

- **Release** process
  - Each component can follow different release lifecycles
  - CORD release: tag the helm-chart and the automation-tools repositories

Use case: deploy SEBA in 5 minutes

- ***Assumptions:*** *OS and Kubernetes installed, switches provisioned*
- ***Workflow***
-
  ```
  git clone https://gerrit.opencord.org/helm-charts
  cd helm-charts
  ```
-
  ```
  helm install voltha
  helm install -n onos -f configs/onos.yaml onos
  helm install xos-core
  helm install xos-profiles/att-workflow
  ```

Use case: deploy SEBA in 5 minutes



Offline installation overview

Prepare for it

- Install a local docker registry and a web-server (to host ONOS apps)
- Download the helm-charts, extend them to use your registry
- Download the container images, push them to the registry
- Download the ONOS application images, save them on the web server

Offline install

- Run the same helm commands using the “-f option” to load your extension file

More info @ https://guide.opencord.org/offline-install.html

Next step: continuous deployment

Such infrastructures (at scale) can’t be managed manually.

Spinnaker as a possible tool for managing the CD process

Assumptions

- The infrastructure is immutable
- Incremental APIs change (adding, then removing vs. changing)

First step: make containers stateless (Voltha/etcd, ONOS/atomix, XOS/Postgres, ...)

ONF

Open Networking Foundation

Thank you

Operator Led Consortium

AT&T

China unicom 中国联通

COMCAST

Google

NTT Group

Türk Telekom