distributed core
provides high-availability, scalability and performance

abstractions & models
allow applications to configure and control the network without becoming dependent on device specifics

applications platform
allows developers to dynamically extend the base capabilities
high-availability
scalability
performance

ONOS distributed applications platform
OSGI / Apache Karaf
configuration & control abstractions
device independence

ONOS networking core

ONOS distributed applications platform

OSGI / Apache Karaf
ONOS distributed applications platform

ONOS networking core

ONOS applications

OSGI / Apache Karaf

GUI

REST API

gRPC

RESTCONF

Command Line

...
ONOS Subsystems

- Device
- Link
- Host
- Topology
- Flow Rule
- Packet
- Statistics
- Intent
- Application
- Leadership
- Messaging
- Storage
- Region
- Mastership
- Driver
- Path
- Tunnel
- Network Cfg.
- Flow Objective
- Group
- Security
- Network Virt.
- Tenant
- OSGi / Apache Karaf
- Network Cfg.
- Device Cfg.
- Discovery
- OpenFlow
- NETCONF
- OVSDB
- GUI
- Event
- Messaging
- Graph
- Cluster
- Leadership
- Core
- Proxy ARP
- SDN IP / BGP
- Packet / Optical
- Mobility
- L2 Forwarding
Brief Retrospective

● Started with a minimal platform with only a few apps
  ○ built with sound structure and solid code & minimalistic REST API
  ○ 4 apps and 1 SB plugin

● Now a platform with many features and apps
  ○ new capabilities, distributed primitives and even greater extensibility
  ○ now 125+ apps, including SB plugins, drivers, models, etc.

● Added new core functionality and apps with each release
  ○ deliberately balancing investments in platform vs. use-cases and apps
  ○ show innovation, but also take pragmatic steps to be deployment-ready
  ○ maintain coherence of architecture and quality of code
Quarterly Releases

- **Avocet** (1.0.0) released 2014-12
  - initial release of clean and modular code-base, protocol independence

- **Blackbird** (1.1.0) released 2015-03
  - improved performance, scale-out, increased robustness

- **Cardinal** (1.2.0) released 2015-06
  - new use-cases, additional core features, additional SB protocols

- **Drake** (1.3.0) released 2015-09
  - platform enhancements, security, UI enhancements

- **Emu** (1.4.0) - released 2015-12
  - CORD features, prototype of dynamic cluster scaling
Quarterly Releases

- **Falcon** (1.5.0) - released 2016-03
  - dynamic cluster, model extensibility, YANG compiler incubation, new SB
- **Goldeneye** (1.6.0) - released 2016-06
  - core enhancements, intents to flow objectives, GUI scaling, P4 PoC
- **Hummingbird** (1.7.0) - released 2016-09
  - new primitives, Kafka & RabbitMQ integration, GUI L&F, Buck build
- **Ibis** (1.8.0) - released 2016-12
  - intent enhancements, dynamic configuration incubation
  - virtualization incubation, transition to brigades
- **Junco** (1.9.0) - released 2017-02
  - HA enhancements, continued incubation of dynamic configuration, virtualization & GUI scaling
Quarterly Releases

- **Kingfisher** (1.10.0) - released 2017-05
  - YANG tools, YANG runtime, intent enhancements
  - virtualization incubation, GUI topology enhancements

- **Loon** (1.11.0) - released 2017-08
  - YANG 1.0 full support, YANG live compiler
  - OpenConfig models, YMS deprecation
  - core HA enhancements, GUI tooling, P4 runtime, ISSU incubation

- **Magpie** (1.12.0) - scheduled for release 2017-11
  - ISSU, gNMI, dynamic config store, YANG 1.1 partial support
  - Device synchronizer
Near-Term Roadmap Items

- Dynamic Configuration *(in progress)*
- In-Service Software Upgrade *(in progress)*
- P4 integration - P4 Runtime & gNMI *(in progress)*
- gRPC API *(in progress)*
- GUI Scalability *(in progress)*
- Build & Release process improvements *(in progress)*
- Virtualization *(in progress)*
- Code-base disaggregation
- Federation (hierarchical & peering)
Dynamic Configuration

- Introduce model-driven configuration capabilities
  - service & device configuration via standard & custom models
- YANG Compiler
  - compiles YANG models into Java DTO classes & schema meta-data
- YANG Runtime
  - registry of compiled models; translates between representations
- Dynamic Configuration Store Service
  - service to distribute and persist model schemas & instance data
- NETCONF & RESTCONF apps
  - facilitate configuration interactions using standard protocols
In-Service Software Upgrade

- Mechanism for gradually upgrading an ONOS cluster
  - upgrades cluster one node at a time without downtime
  - nodes deliberately evacuate load prior to their upgrade
  - nodes rejoin the cluster and assume regular role after upgrade

- Requires portable serialization for cluster comms
  - upgraded nodes must be able to speak the “old” language
P4

- Enable programmable pipelines
- Unified Control and Config via P4Runtime and gNMI using OpenConfig models
- Support pipeline agnostic and aware applications
- INT (in-band telemetry)
- P4 Fabric
gRPC API

- Allow fine-grained & high-performance interactions between ONOS and off-platform apps
  - presently available only for on-platform apps via Java API
  - REST API suitable only for relatively low-frequency & coarse interactions

- Enables apps to be run on or off platform
  - permits compute resource isolation
  - off-platform apps as micro-services

- Allows ONOS apps to be written in other languages
GUI Scalability Enhancements

● Allow users to use GUI in large network deployments
  ○ focus on usability and flexibility of topology layouts
  ○ manage amount of information on the screen via zooming in/out

● Nested & customizable layouts
  ○ physical layouts, e.g. geographical map, floor-plan, rack
  ○ logical layouts, e.g. spine-leaf

● Customizable icon scaling
  ○ allows users to adjust the density of information on screen

● Revised GUI glyphs for map entities & buttons

● Global search
Virtualization

• ONOS as network hypervisor
  ○ using OVX model aimed at virtualization for off-platform apps
  ○ on-platform apps can be network-aware, not network-specific

• Produces SDN-capable virtual networks
  ○ with topology and without implicit connectivity
  ○ connectivity is explicitly programmed via REST & OpenFlow APIs
  ○ permits arbitrary topologies, from Big Switch to isomorphic

• Protocol independent & modular
  ○ manual or programmatic network embedding
  ○ support for arbitrary data-plane virtualization mechanisms
Code-base disaggregation

- Separate apps and extensions into their own repos
  - provides more autonomy for collaborators
  - provides better project life-cycle management
  - offers better scalability in terms of community contributions

- Separate the distributed network-agnostic platform from the network-aware core subsystems
  - presently the separation is logical, but there exist some (albeit weak) ties between the two
  - requests for such platform from Fermi & Oakridge Labs, among others
Cluster Federation

- Coordination mechanism for multiple ONOS clusters
  - clusters exchange virtualized views of own networks with others
  - clusters expose limited provisioning & control actions to others
  - permits peer-to-peer & hierarchical arrangements

- Peer-to-peer variant being developed by GEANT
  - aims to support different administrative domains
Getting there...

- Brigades - harnessing the power of community
  - help to make forward progress on key ONOS roadmap items
  - ONF & TST acting mostly as shepherds

- Brigades
  - ISSU
  - Dynamic Configuration Brigade
  - Virtualization Brigade
  - GUI Brigade
  - gRPC Brigade
  - Teaching Brigade
  - Build & Release Process Brigade
  - Northbound Brigade
  - Security & Performance Analysis Brigade
  - P4
For more information

Wiki - https://wiki.onosproject.org/display/ONOS/Wiki+Home
Roadmap - https://wiki.onosproject.org/display/ONOS/Roadmap
Brigades - https://wiki.onosproject.org/display/ONOS/Brigades
Software Defined Transformation of Service Provider Networks

Join the journey @ onosproject.org