Open for All.
SONiC Application Extension Infrastructure

Marian Pritsak, SW Architect, Mellanox
Agenda

State of SONiC Today
Case for extendable SONiC
How SONiC will enable native apps
Gradual Acceptance
Modular Development
Seamless integration
Distributed Orchestration
State of SONiC Today

More and more features coming to SONiC
- Some are specific to the use case
- Some are specific to the platform
- Some are experimental or not mature enough
State of SONiC Today

- Troublesome to maintain
- Prone to break
- Not scalable
Case for extendable SONiC

- **Rock solid** SONiC image with the capability to be extended
- Functionality added on demand
- SONiC+ applications behave **natively**
How SONiC will enable native apps

- Distributed Orchestration
- Seamless Integration
- Modular Development
- Gradual Acceptance
Gradual Acceptance

How Linux brings in new features:
• Start as a module
• Become popular
• Become stable
• Be accepted into the main codebase

SONiC community needs to follow the same guideline:
• New “modules” shouldn’t be a part of the core image
• New “modules” shouldn’t delay the SONiC release cycle
• Even some of existing the SONiC features should become optional
• The “modules” can become a part of the core image if the community decides so

Distributed Orchestration
Seamless Integration
Modular Development
Gradual Acceptance
Modular Development

The concept of SONiC SDK:
- The two docker images provided with each next SONiC release

The SDK Docker:
- Easy development without the need to build a core image
- Contains all dev packages built with the core image

The Distribution docker:
- Contains all libraries without the dev components
- Provides a base for distributing the SONiC application

- Distributed Orchestration
- Seamless Integration
- Modular Development
- Gradual Acceptance
Seamless integration

- Systemd service
- Docker container definition
- CLI
- REST API
- Warm reboot
- Defaults

- Autogenerated integration
  - Distributed Orchestration
  - Seamless Integration
  - Modular Development
  - Steady Acceptance
Distributed Orchestration

Why a distributed orchestration is beneficial:
• Enables the SONiC extensibility on the SAI layer
• Failsafe – custom addons can fail, but the switch stays operational
• Convenient for partial upgrades
Distributed Orchestration

ORCHAGENT
Switch OID
Port OIDs
Mirror session OIDs

APP_DB

ASIC_DB

My-container
My-app
ACL (switch OID? Ports OIDs? Mirror session OIDs?)

OID_LABEL_MAP
SWITCH_TABLE:switch|oid:0x0000001
PORT_TABLE:Ethernet0|oid:0x0000002
MIRROR_TABLE:session1|oid:0x0000003
MY_APP_ACL:rule1|oid:0x0000004

Seamless Integration
Modular Development
Gradual Acceptance
Call to Action

• A new development flow for the future scalable SONiC
• Follow up on the SONiC Application Workgroup

Mailing list: https://groups.google.com/d/forum/sonic-application-workgroup
Open for All.

MARCH 4 & 5, 2020  |  SAN JOSE, CA