Stratum
Introducing the next generation of SDN interfaces

Brian O'Connor
brian@opennetworking.org

Challenges

Wanted:

A lightweight, production quality switch agent with a consistent set of interfaces across a wide variety of fixed function and programmable hardware.

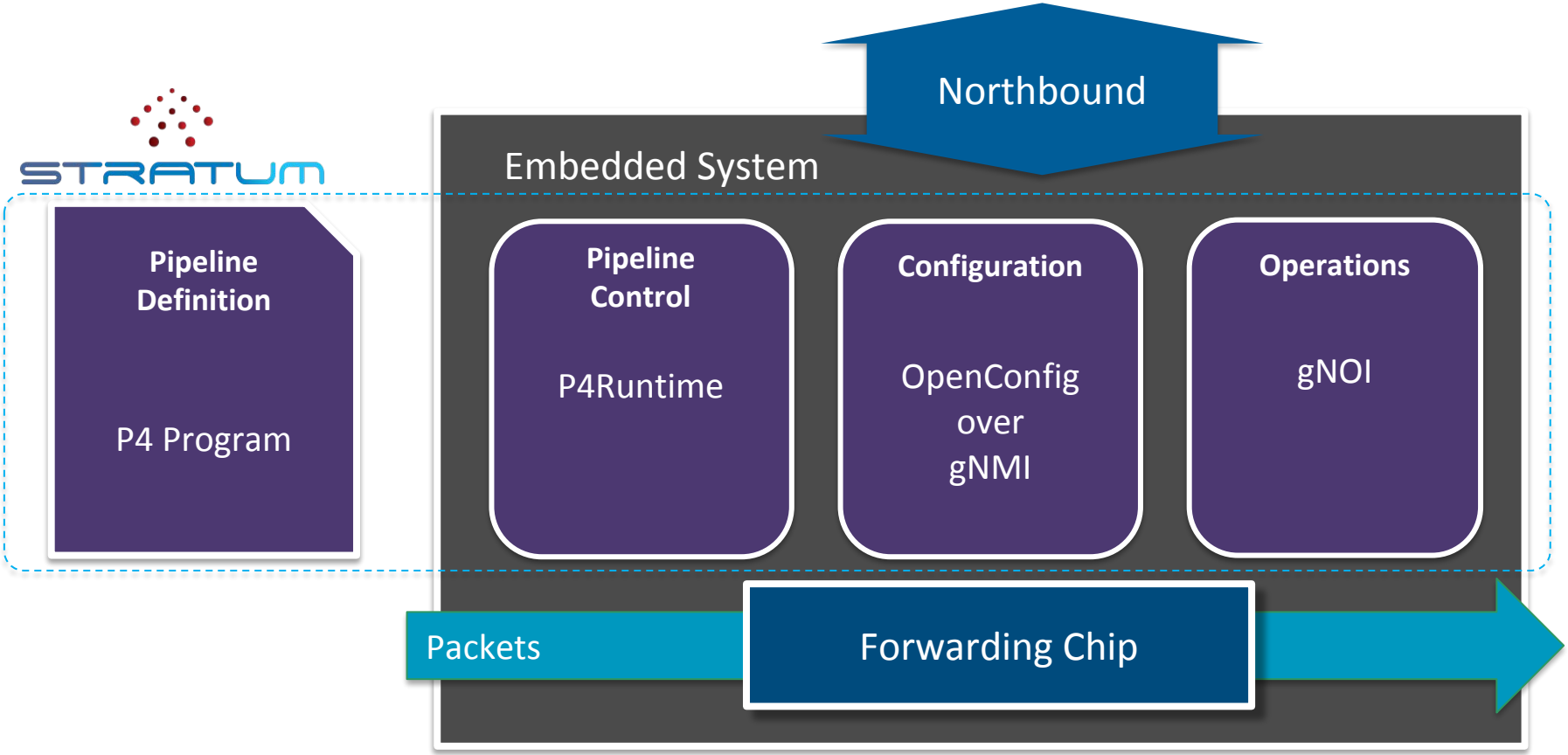
- Overcome limitations and gaps with existing control and management protocols
- Handle incremental migration across multiple axes
  - E.g. fixed-function to programmable switching chips, traditional network to SDN
- Enable new generation of programmable devices

Stratum Interfaces

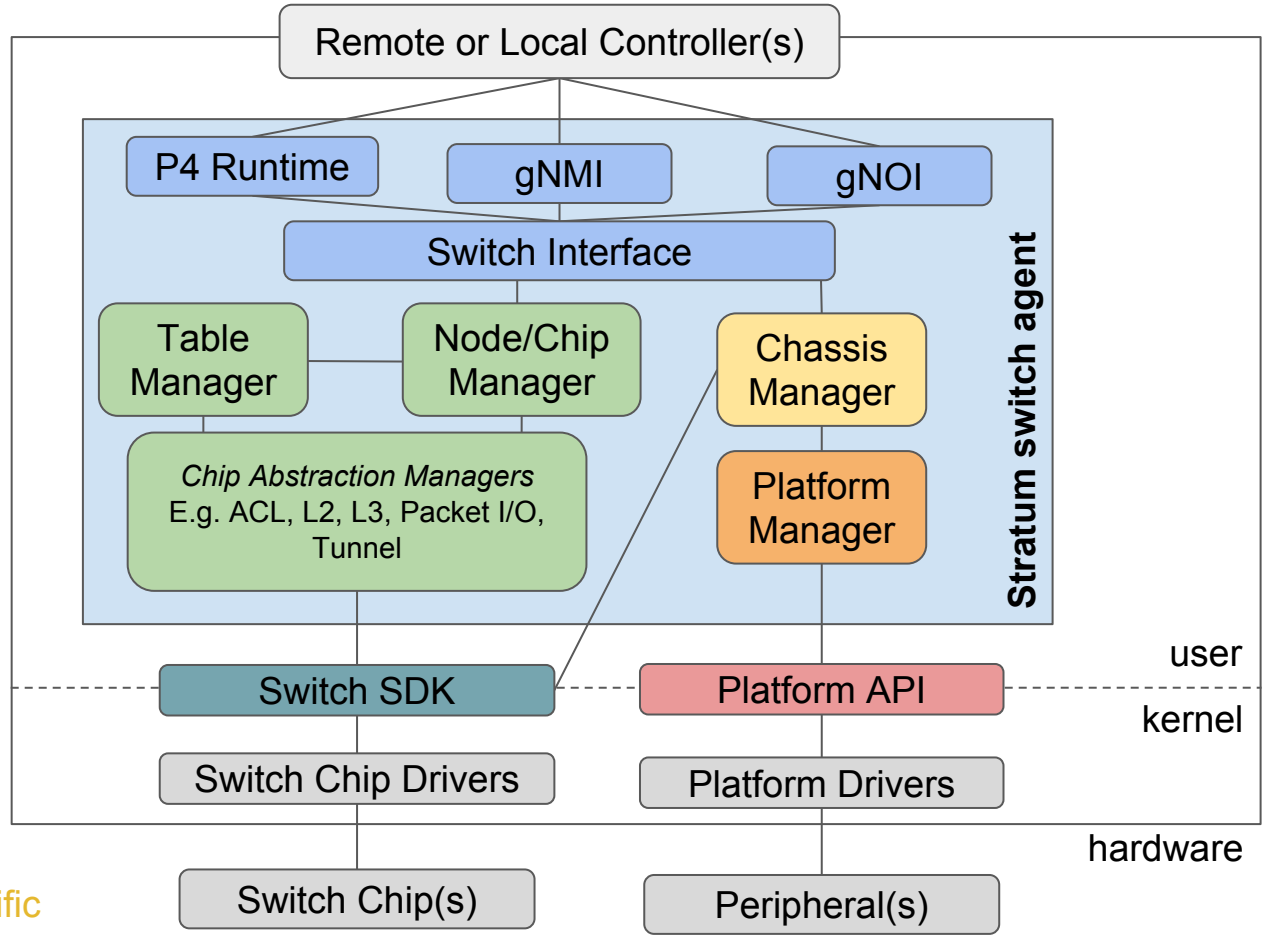
External interfaces are drawn from community working groups (i.e. P4 Language Consortium, OpenConfig working group, and gRPC)

- Seamlessly support remote and local interaction
- Not tied to a particular programming language or kernel

Lightweight and Production-ready Implementation



Switch Agent Architectural Components



Shared (HW agnostic)
Chip specific
Platform specific
Chip and Platform specific

Switch Common Interfaces

- P4Runtime (service definition, documentation)
  - gRPC-based data plane control protocol that is chip-, pipeline-, and packet header-agnostic
  - Message payloads derived from a P4 program using program-dependent P4Info instance to build messages
  - Enables a local or remote entity to load the pipeline/program, arbitrate mastership, read and write forwarding table entries, counters, and other chip features, as well as send and receive packets
- gNMI (service definition, models)
  - gRPC-based service to modify configuration and stream telemetry information
  - Messages payload is modelled in Yang, and Stratum prefers OpenConfig models
  - Config that gNMI deals with tend to be long-lived (i.e. persistent across device reboots), but mutable
- gNOI (service definitions)
  - gRPC-based collection of micro-services for runtime management, for example:
    - Device reboots, pushing/rotating SSL keys/certs, BERT [bit error rate testing on a link/port], ping testing
    - Ephemeral state management (clearing L2 neighbor discovery/spanning tree, resetting a BGP neighbor session)

Forwarding chip-specific Interfaces

- Node / Chip Manager
  - Upon initialization, one instance is registered with `SwitchInterface` per chip
  - Works in conjunction with the Table Manager and Chip Abstraction Managers to provide access to switch chip functionality
- Table Manager
  - Responsible for translating P4 Runtime calls to chip specific entities
- Chip Abstraction Managers
  - Abstracts switch functionality such as L2, L3, ACL, Packet I/O
  - Each of these managers define their own interface, which can potentially be reused across implementations
- Chassis Manager
  - Currently, responsible for chip initialization and dataplane port mapping

Platform-specific Interfaces

- Chassis Manager
  - Provides configuration and access to telemetry from ports and peripherals
  - Responsible for mapping logical (e.g. SDN) ports, physical (e.g. slot, port, channel), and chip specific (e.g. dataplane) ports
- Platform Manager
  - Only class that interacts with platform components
  - Currently, ONLP is proposed as the platform API, so a common implementation of this manager could be used across any ONLP compliant boxes

Implementation and Deployment

- Both remote controllers and local control planes/agents connect through common gRPC interfaces
- A distribution is realized by defining a target, or collection of Node and Chassis Managers, and building a target specific binary
- A P4 Program is pushed to each device after device initialization and is not part of the distribution
  - Relies on a target specific P4 compiler to produce binaries
- A Distribution deployed as a user-space process
- The agent process's lifecycle is managed by a process manager

Design Principles

1. Chip, Platform, and Dataplane independent interfaces

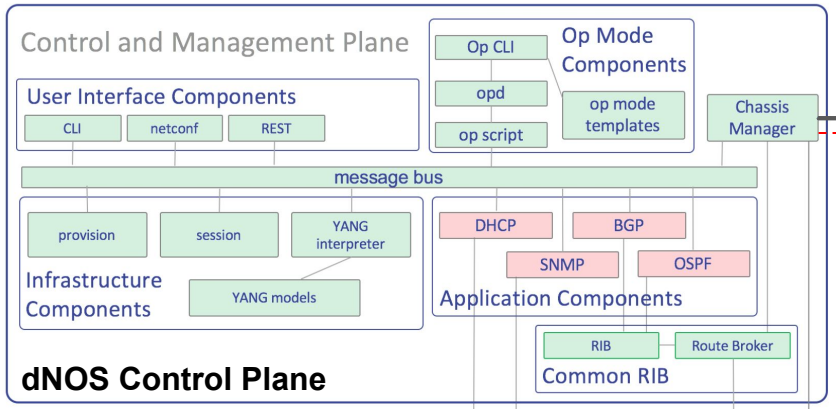
2. Generic and common APIs for local and remote control and configuration

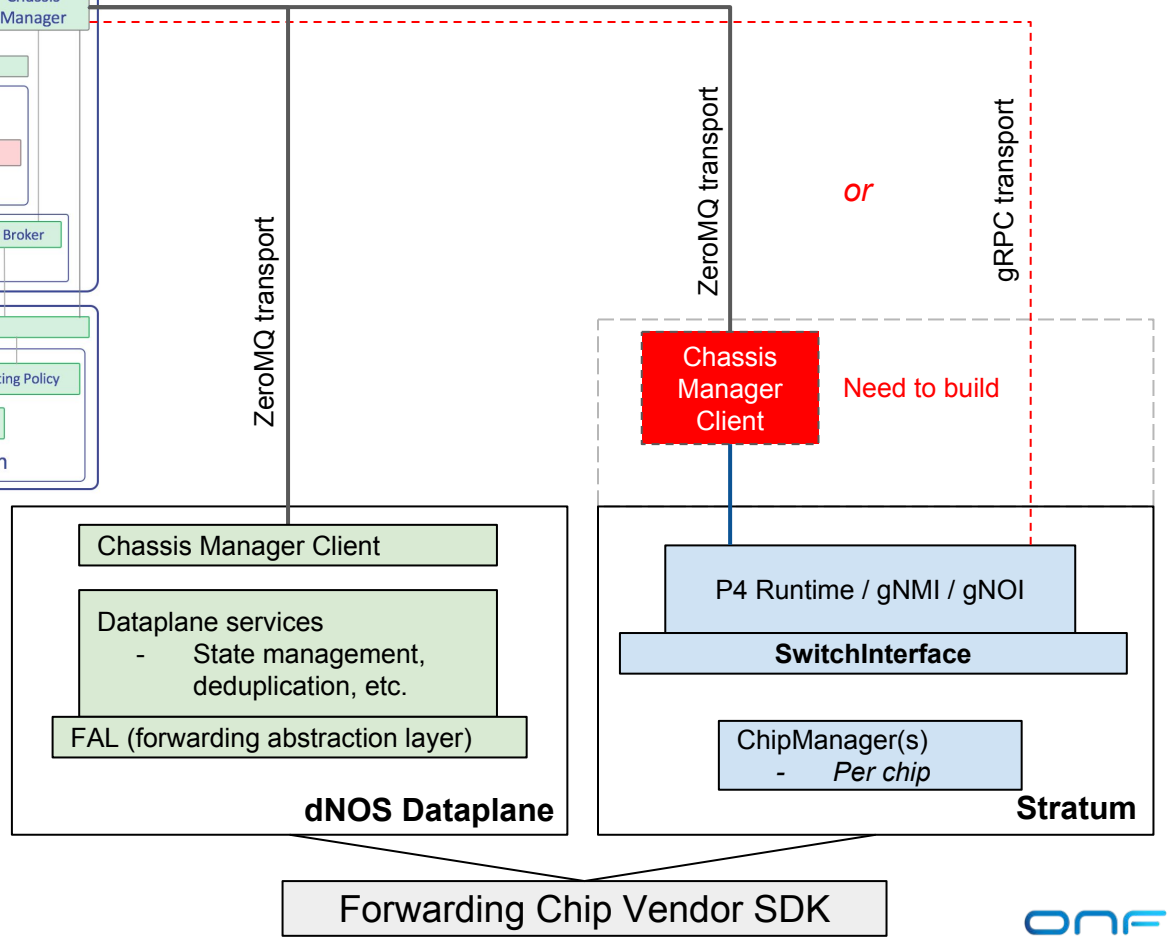
3. Lightweight
   - User space, minimal dependencies, easy to deploy, minimal system requirements, no built-in control plane functionality (e.g. BGP)

4. Reusability and extensibility
   - Common interfaces both internally and externally
   - Flexibility to extend to accommodate chip or platform value-added functionality
   - Favor 3rd party community work when appropriate (e.g. ONLP for peripherals)

Integration and Network Transformation

dNOS + Stratum





Stratum as a parallel dataplane for dNOS

Chassis Manager serves as `netlink`/Stratum mapper

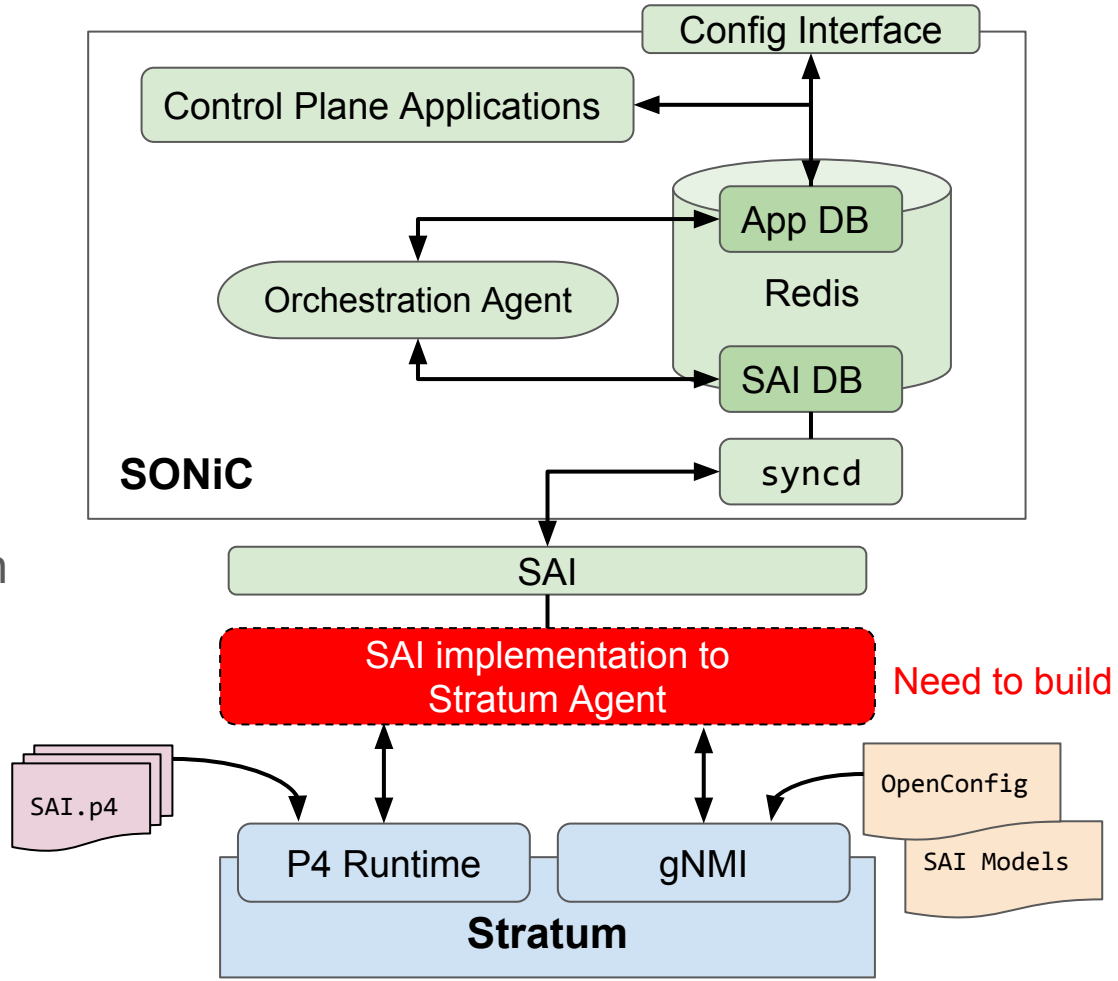
- Can implement gRPC services directly
- Or, can build client that runs on dataplane to map ZeroMQ to gRPC services

SONiC + Stratum

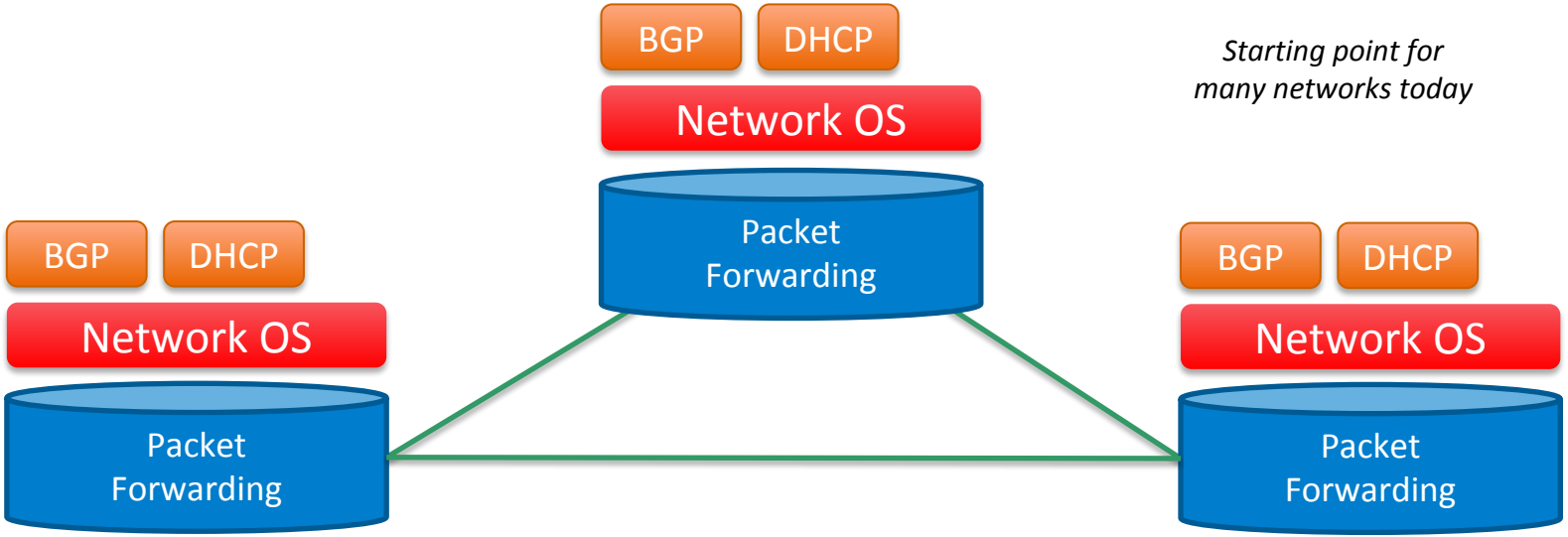
SONiC uses Stratum as implementation of SAI

SAI.p4, OpenConfig and maybe additional SAI models used as dataplane contract

Customer specific extensions can be exposed to control plane apps via SAI DB or P4 Runtime interface

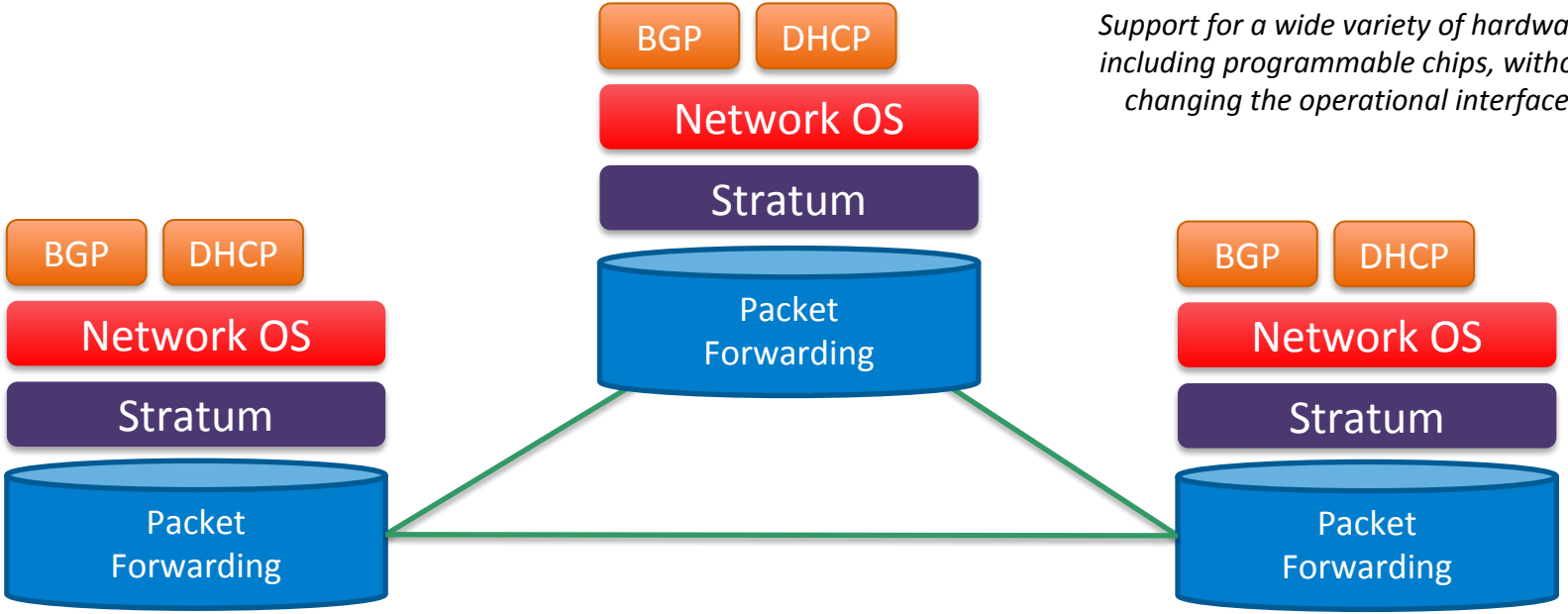


Use Case: Network Transformation with Stratum



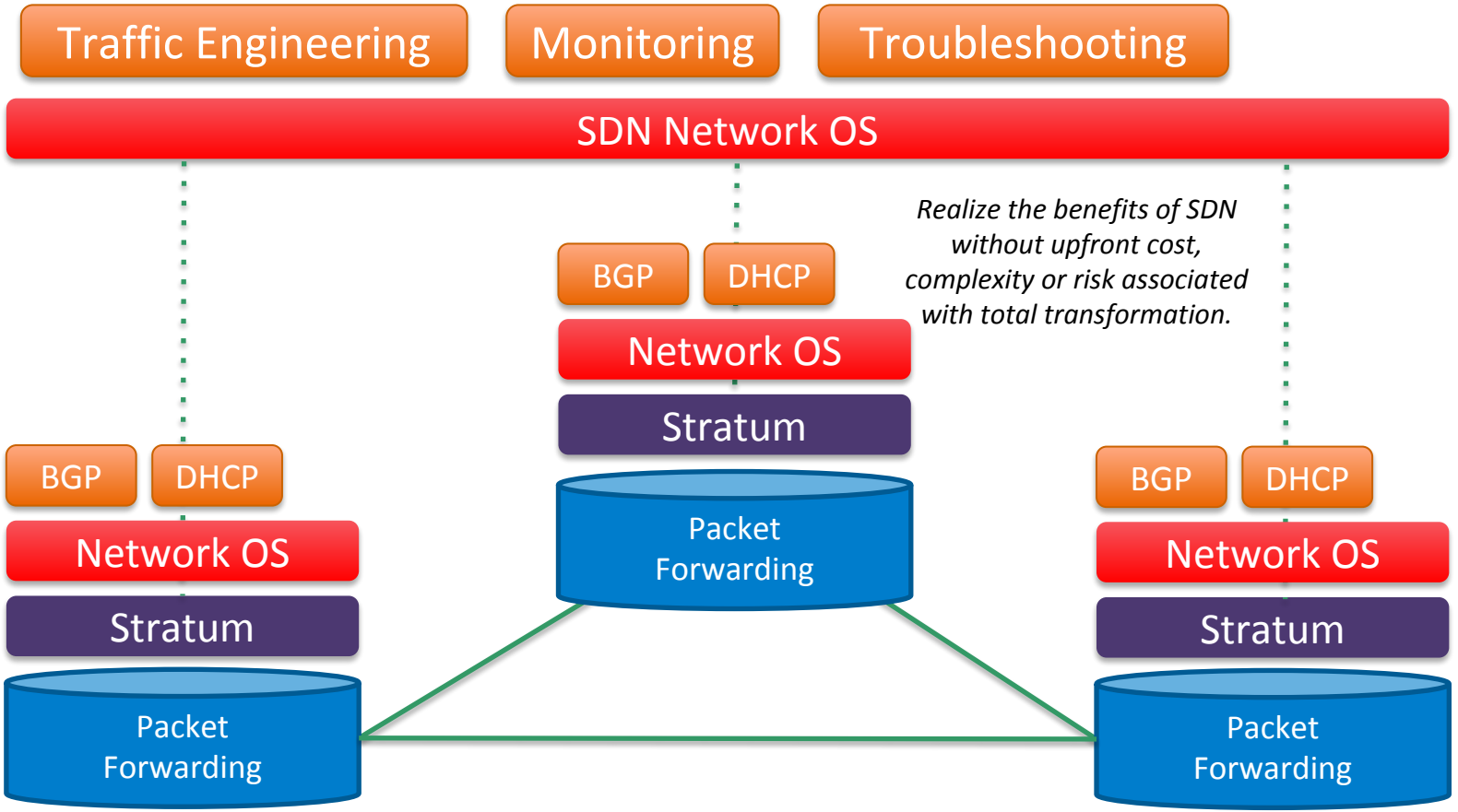
Starting point for many networks today

Upgrade to a Stratum-powered NOS

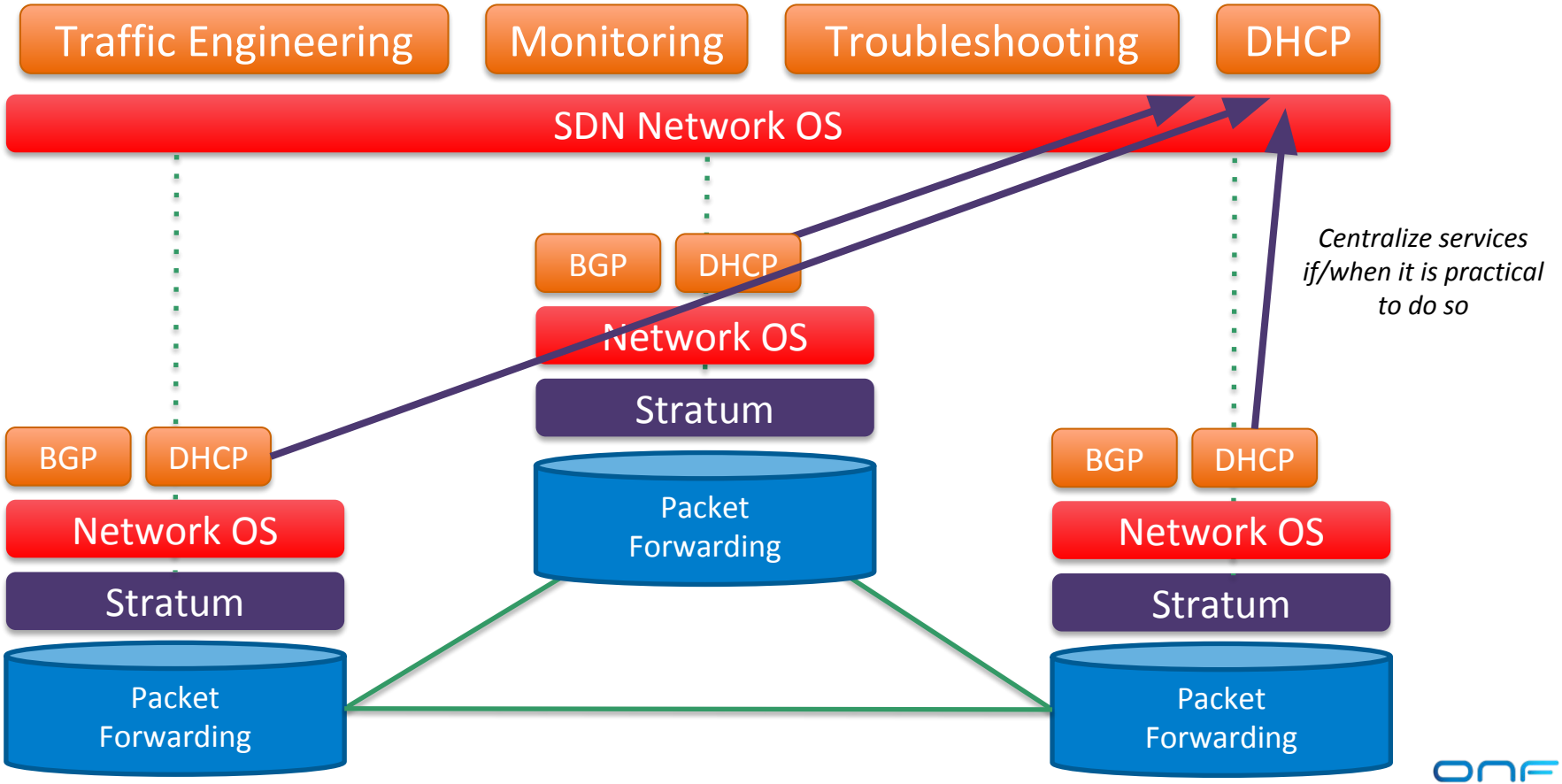


Support for a wide variety of hardware, including programmable chips, without changing the operational interface.

Add an SDN OS and new services



Migrate existing services



Cloud Providers

Telecom Operators

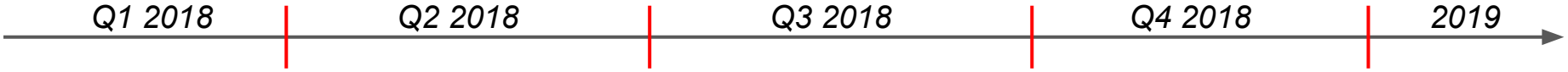
Networking Vendors

White Box ODM Vendors

Silicon Vendors

Other Open Source Projects

Stratum Development Timeline



Stratum Community Launch
with 22 partners

Seed code from Google available to pioneering members

Field Trials, Production Deployments
on cloud and telco networks

Open Source Launch
with forwarding chip and platform support for every vendor member

Pioneer work days
- Reference Platform Support (HW & SW)
- Development Infrastructure (Build, CI, etc.)

Community Development

Codebase GA for Stratum Members
- Expanded platform support
- Feature development
- Hackathons

Stratum Summary

- Common interfaces for control, configuration, monitoring and telemetry
- Minimal design for high performance local or remote control and management
- Incremental migration paths enables incremental value-add
  (e.g. SDN, programmable hardware)
- Broad switching chip and platform support underway
- Production-root implementation designed to scale

https://stratumproject.org/

To join the announcement mailing list, send an email to:
stratum-announce-join@lists.stratumproject.org
(Then, click the link in the confirmation email)