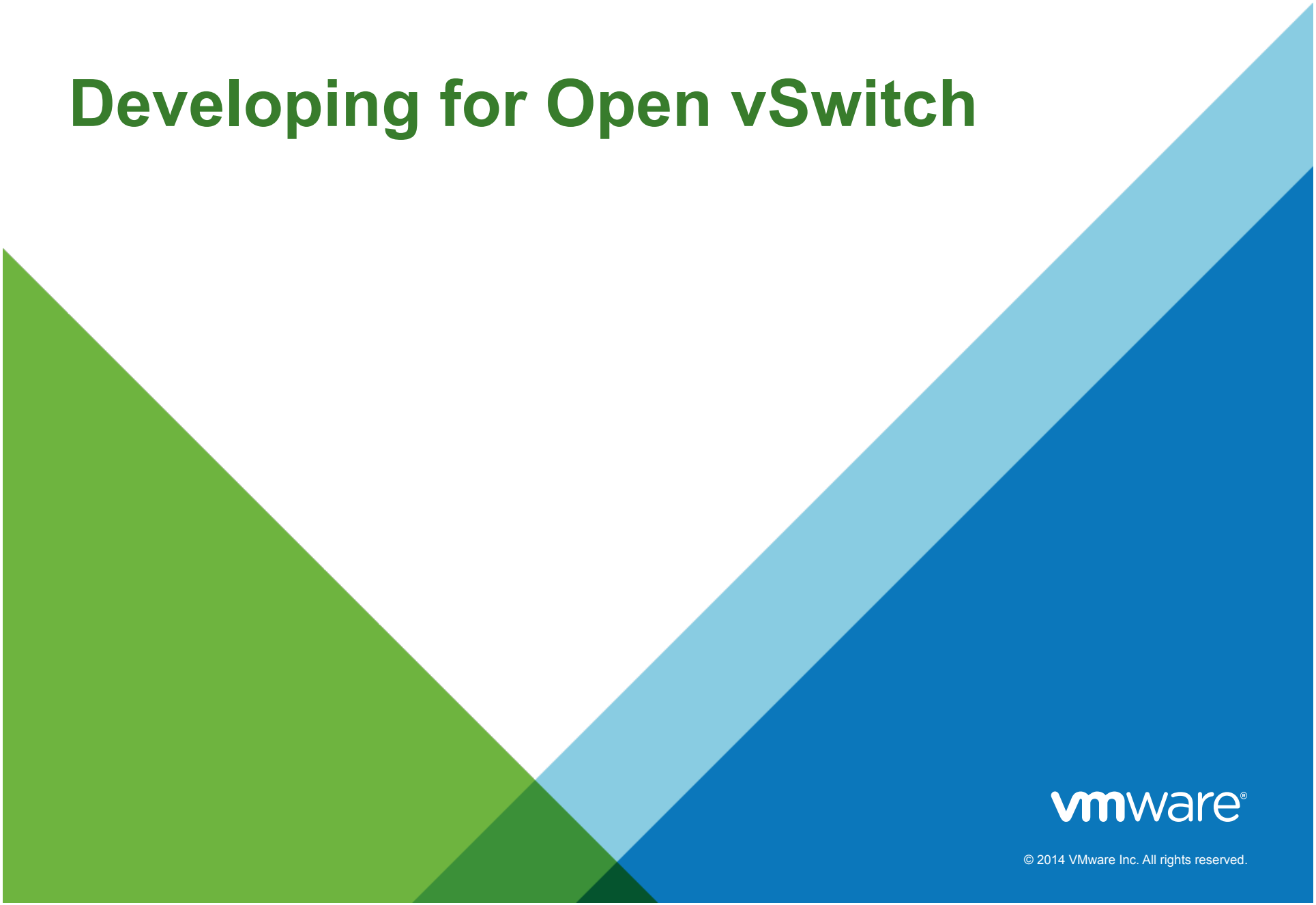
Developing for Open vSwitch



Mailing Lists

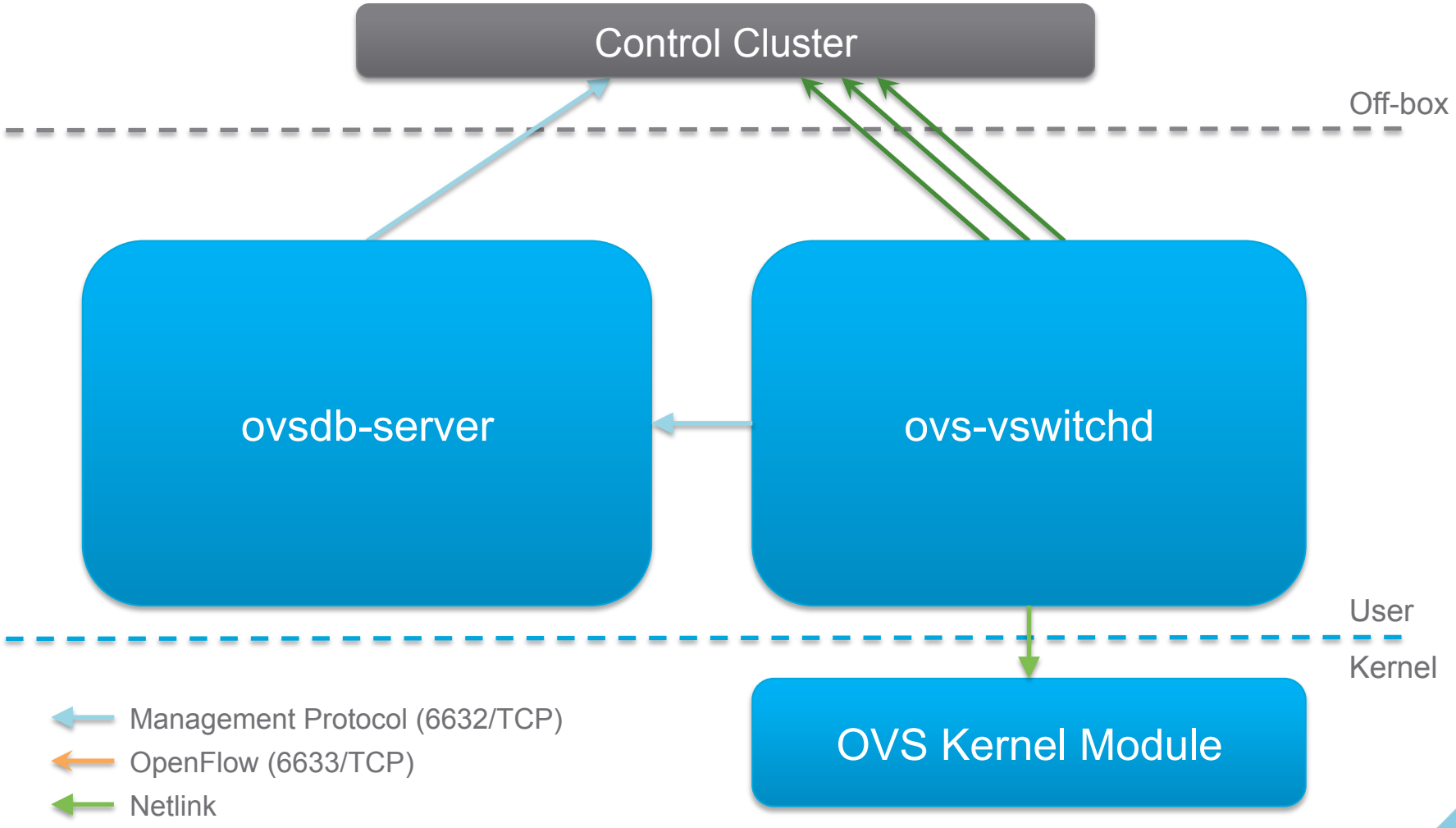
- Subscribe links on main openvswitch.org web page
- Number of subscribers to mailing lists:
  - discuss: 1488
  - announce: 787
  - dev: 718
  - git: 173

vmware

Process

- Git version control
- All development occurs on the mailing lists
  - dev: Code reviews
  - git: Change notifications
- All code is peer-reviewed
- Online gitweb:
  - http://openvswitch.org/cgi-bin/gitweb.cgi?p=openvswitch;a=summary

Main Components



vmware

Interesting Files

- DESIGN: Design decisions used to implement OVS
- FAQ: FAQ
- INSTALL.*: Installation instructions for a number of platforms
- IntegrationGuide: Guide to integrating with new systems (e.g., new hypervisor)
- PORTING: Guide to porting OVS to new platforms (e.g., new OS or hardware platform)
- ovsdb/SPECS: JSON-RPC protocol used by OVSDB (basis for Informational RFC)

Submitting Patches

- Described in “SubmittingPatches” file
- Use “git format-patch” with one patch per email
- The source tree should build and work after each patch is applied (run “make check”)
- Make sure to update man page and NEWS, if appropriate
- Need “Signed-off-by:”, which means agreeing to the “Developer’s Certificate of Origin”

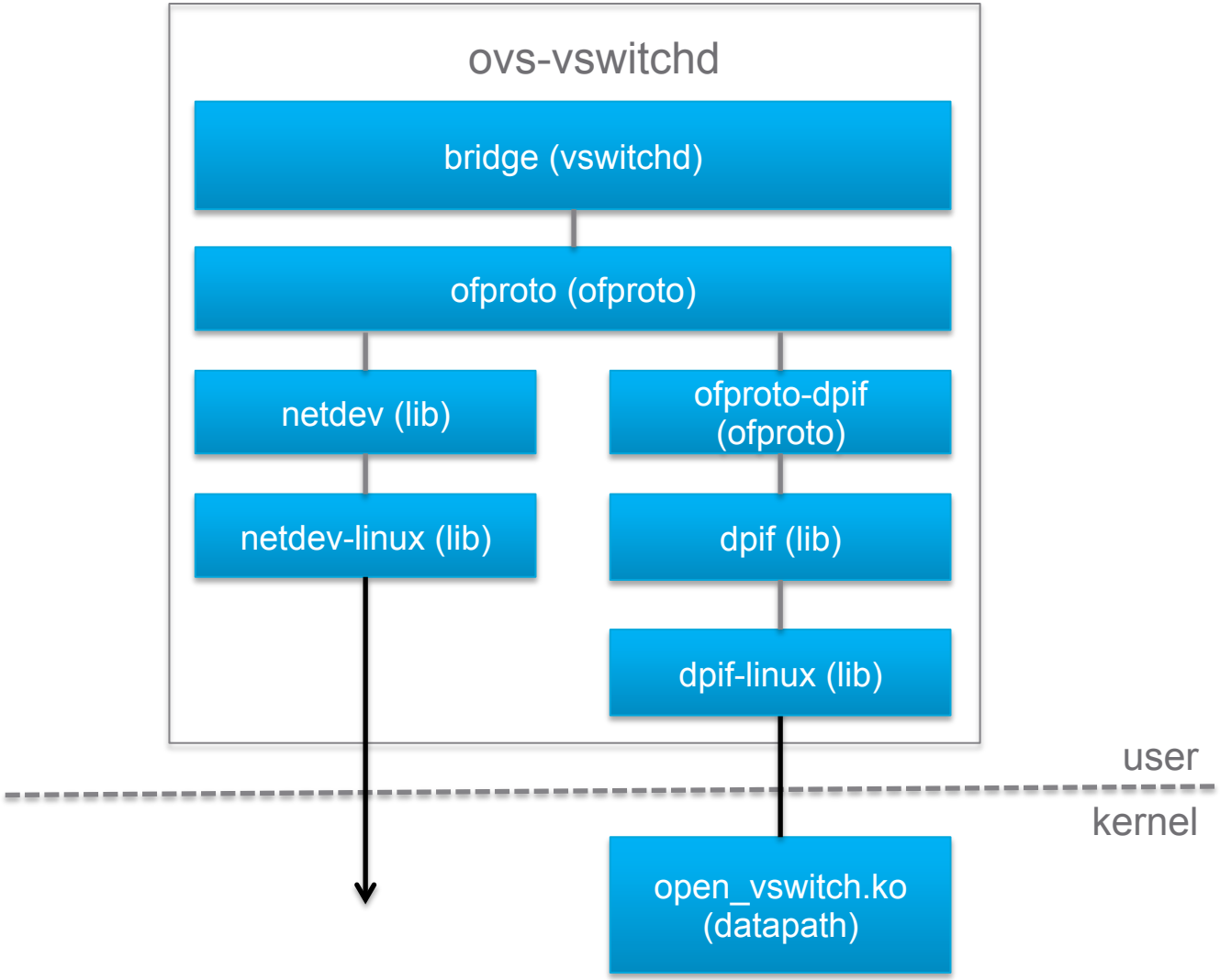
Code Layout

- datapath: Linux kernel datapath implementation
- include: Include files
- lib: Common library files (netdev, dpif)
- ofproto: OpenFlow library
- ovsdb: Database code (server and clients)
- utilities: Utilities for OVS (ovs-ofctl, ovs-vsctl)
- vswitchd: ovs-vswitchd source code

Architectural Overview

- ovs-vswitchd: Responsible for switching. Reads configuration from ovsdb-server and passes configuration down to “ofproto” library.
- ofproto: OVS library that implements an OpenFlow switch.
- dpif: Instance of an ofproto for software switches, which “explodes” wildcard entries into non-overlapping megaflows.
- netdev: Library that abstracts interacting with network devices.

Linux Kernel Datapath Example



vmware

Integration Code

- Bulk of code is platform independent
- Distribution contains platform-specific code in the following directories:
  - xenserver
  - debian
  - rhel
- Platform-specific install instructions are in the root directory
- Includes init scripts, platform-specific daemons, and glue code

Python Bindings

- The “python” directory contains Python bindings
- Many of the C-based library functions from “lib” are available
- Read/write bindings for OVSDB available
- A number of small daemons written in Python use these bindings and ship with the distribution

Testing

- “tests” directory contains Autotest unit tests
- Run “make check”—runs nearly 1500 unit tests
  - Run multiple simultaneous instances:

    make TESTSUITEFLAGS=”-j8" check
  - Run a specific class:

    make TESTSUITEFLAGS="-k ofproto" check
- If adding new functionality, please consider adding your own unit tests!