PICA8®
OPEN NETWORKING

WHITEPAPER

PicOS Overview

TABLE OF CONTENT

1 PicOS: A Bare-Metal, White-Box Switching Operating System (OS)

Until now, all the switching and routing vendors would sell their software and hardware bundle together. This "system-based" sale was very similar to the server sales model 20 years ago. Today, the model has evolved to where it is now typical to buy the server hardware (for example, Dell, HP, Lenovo, Quanta) and software (Red Hat, Microsoft, Ubuntu) from different vendors.

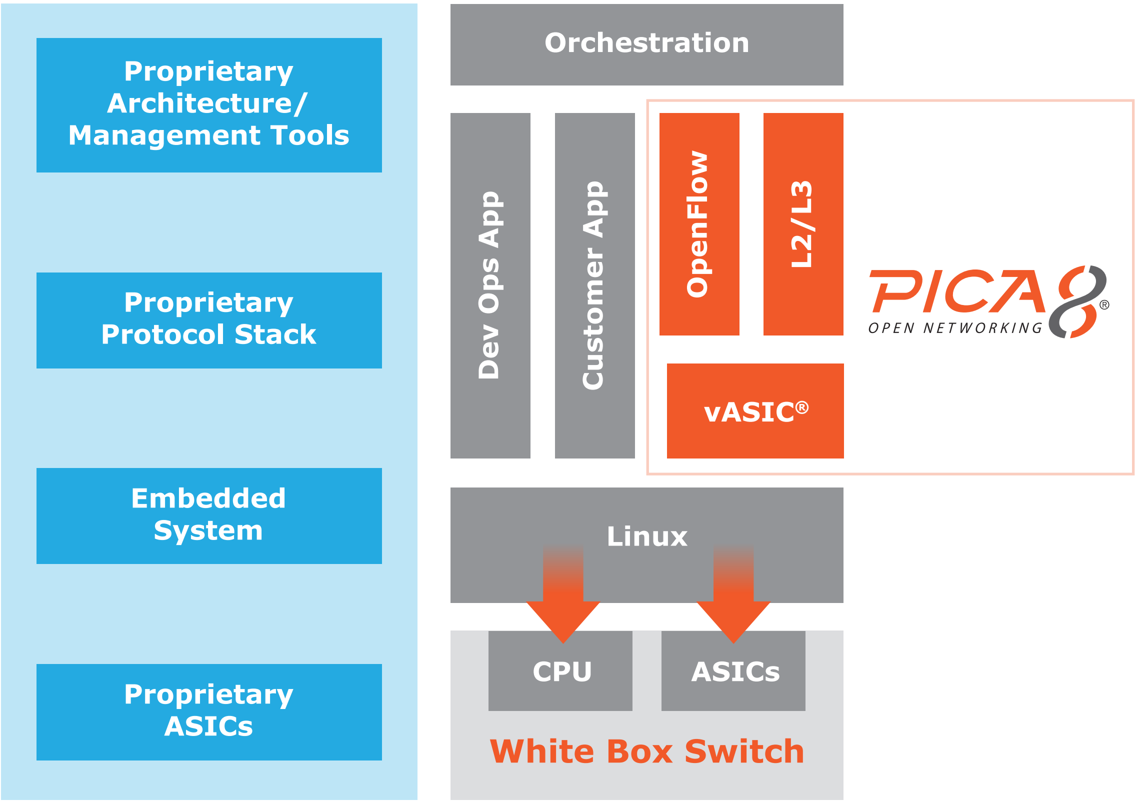
At Pica8, we believe a similar evolution will occur with networking.

The advantages of moving away from system-style sales are:

Lower cost of ownership – Because switching hardware is becoming standardized through multiple original design manufacturers (ODMs) offering similar platforms, and which all leverage merchant Silicon from vendors such as Broadcom, Mellanox, Intel and others, the combination enables the industry to view the hardware as commoditized similar to what we see on the server side.

No lock-in solution – It will be possible to change the OS and keep buying from the same hardware vendor. At the same time, you will be able to keep the OS and change the hardware vendor.

Faster innovation – Today, if we want a vendor to sell a revolutionary ASIC, they would have to build a full routing and switching stack to compete against incumbent networking vendors. The separation of software and hardware will enable the hardware vendor to focus on best-in-class hardware and the software vendor to be concerned with software innovation.



PicOS represents Pica8's commitment to building a hardware-independent, open operating system.

What Makes PicOS Open?

- The hardware agnostic nature of PicOS is driven by the fact that the OS is not tightly coupled to switching ASICs, CPU or memory hardware
- PicOS is a user space application, that runs on an unmodified Linux kernel (Debian), use your existing Linux tools for programming and optimizing your network
- Switching and routing stack built on the XORP routing community (General public license now owned by Pica8)
- PicOS has complete OpenFlow support, through Open vSwitch (OVS) integration

2 General Architecture

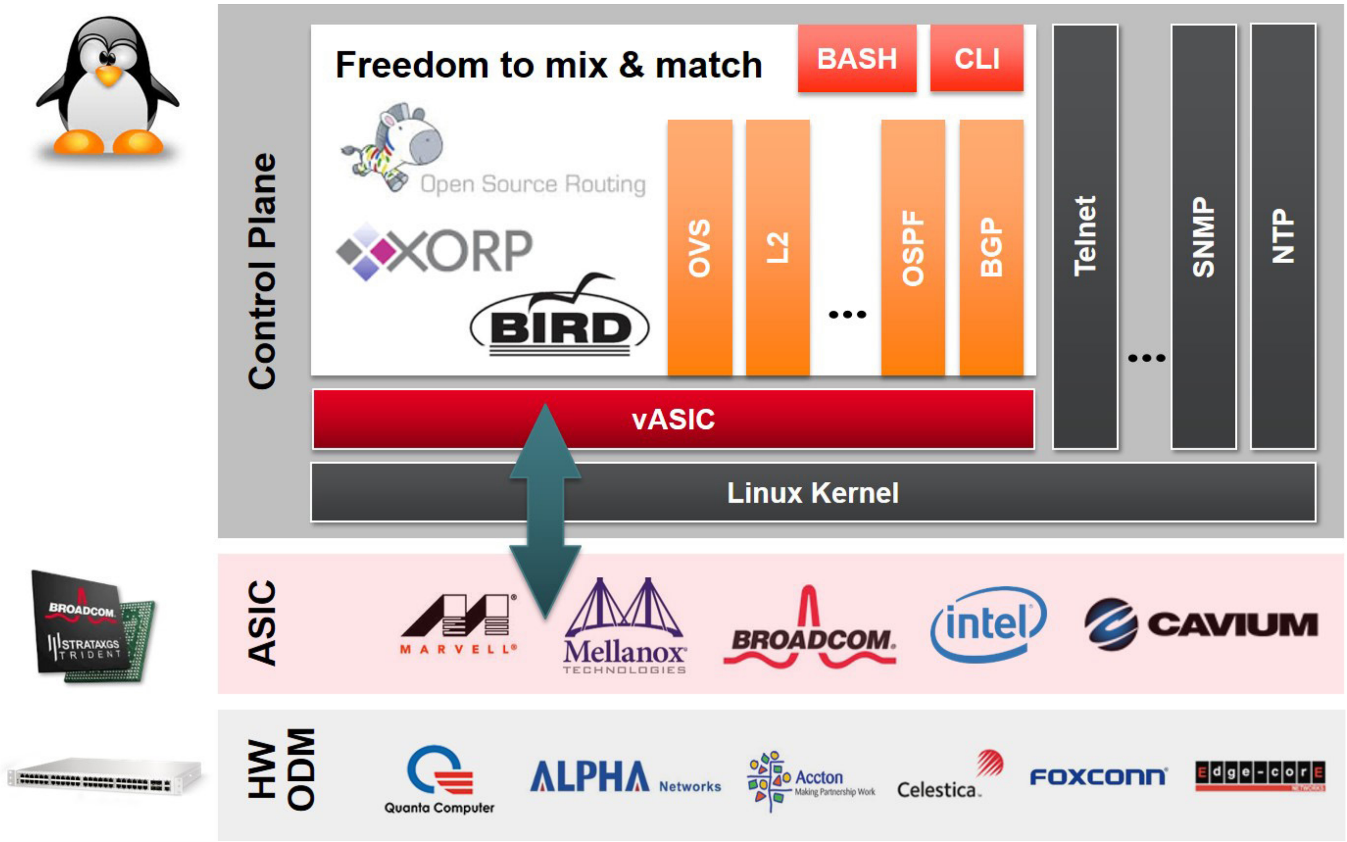
PicOS is built with the following components:

An unmodified Linux kernel – PicOS is using standard, non-modified Debian Linux. Not modifying the Linux kernel enables us to update it faster and thus use the latest kernel innovation. It also enables our customers to use standard Linux tools in conjunction with PicOS tools.

A virtual ASIC Technology (vASIC®) – This is a hardware abstraction layer enabling us to support multiple hardware and ASICs without modifying any PicOS tools.

Traditional switching and routing protocols – Following the acquisition of XORP, Pica8 continues to improve this best-in-class routing and switching stack.

Hardware-accelerated Open-vSwitch (OVS) – PicOS is leveraging Open vSwitch to stay up to date with fast-moving OpenFlow and SDN technologies. We are porting the open source OVS and modifying it to leverage the switch ASIC to handle flows at line rate. OVS is providing industry-leading OpenFlow support & integration with CloudStack or OpenStack.



3 Provisioning: Standard Network CLI, DevOps and SDN

PicOS Provisioning

There is considerable confusion in the industry about the best way to manage and provision networking equipment. At Pica8, we believe that the best network equipment or a network management practice is using the best tool for the specific task. It's that simple.

Some examples:

Vendor-specific CLI: Text CLIs are today the most-used provisioning system. IOS CLI and JUNOS CLI are the most common examples. Having one single configuration and one single environment to troubleshoot and monitor the equipment is still mandatory to help a networking operational team. The CLI should have best-in-class tools:

- Commit and commit confirmed
- Rollback configuration
- Command completion and online help
- Hierarchical configuration

JSON / REST API: Application programming interfaces (APIs) help to interact with the switch. The CLI should be using a JSON API (most networking equipment have an API on top of the CLI, which gives incomplete access to the equipment configuration).

Linux-based configuration: Networking equipment should be managed like any other Linux servers. In the Linux world, configuration manager (Puppet, Chef) are used to standardize and automate the deployment of servers. In PicOS, the full Linux system is available and every part of the switch can be configured from the Linux Shell.

OpenFlow API: We strongly believe that the current complexity and diversity of routing and switching protocol is another type of vendor lock-in. The future of networking is a clear ASIC abstraction layer to let network wide software control networking equipment. This abstraction layer is controlled by the OpenFlow protocols.

Using the Best Tool for the Job

Those three types of provisioning are not exclusive but complementary:

- Single configuration file and environment for traditional routing and switching
- DevOps and Linux tools to manage networking devices, such as Linux servers (logging, security, patches, configuration managers)
- OpenFlow API for network-wide application and software-defined provisioning

**One tool does not fit all needs.
At Pica8, we believe in the best tool for the job.**

4 Traditional Routing and Switching

High-Level Architecture

PicOS has a very complete routing and switching capacity. Its stack (also called L2 / L3 mode) is divided into two subsystems:

1) The higher-level (also known as user level) subsystem consists of the routing protocols themselves, along with routing information bases and support processes. These routing daemons are updating the kernel routing table using a standard netlink interface.

2) A vASIC daemon is monitoring the kernel routing table and syncing it with the ASIC accordingly. The vASIC manages the forwarding path—anything that needs to touch every packet—and provides APIs for the higher level to access. The goal is for almost all of the higher-level code to be agnostic as to the details of the forwarding path.

For user-level routing protocols, we've developed a multi-process architecture with one process per routing protocol, plus extra processes for management, configuration and coordination. To enable extensibility, we designed an inter-process communication mechanism between these modules. This mechanism is called Extensible Resource Locators (XRLs) and is conceptually similar to URLs. URL mechanisms, such as redirection, improve reliability and extensibility and their human-readable nature makes them easy to understand and embed in scripting languages.

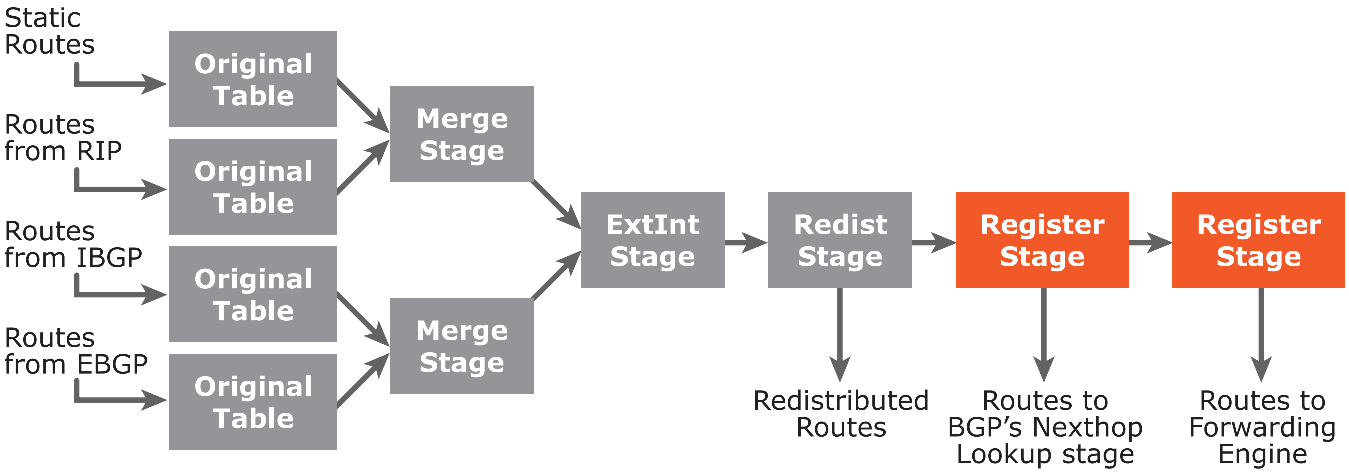


Figure: Routing protocols, RIB and FIB interaction

Robustness

The routing and coordination processes in PicOS run in user space on a traditional Linux operating system. Routing processes are protected from each other and can have their resources constrained according to administrative preference. Furthermore, routing processes can crash without affecting the kernel, forwarding plane, or each other. And if a routing protocol does crash, the RIB will remove its routes from the forwarding engine and optionally inform the re-starting routing process of the routes it previously held.

Fast Convergence

Routing protocol implementations have often been scanner-based. Periodically, a scanner runs to accumulate changes, update the forwarding table, notify neighbors, etc. These implementations lead to low-CPU utilization but also bring poor route convergence properties. In modern networks, fast convergence is a priority.

The PicOS routing stack provides the following capacities:

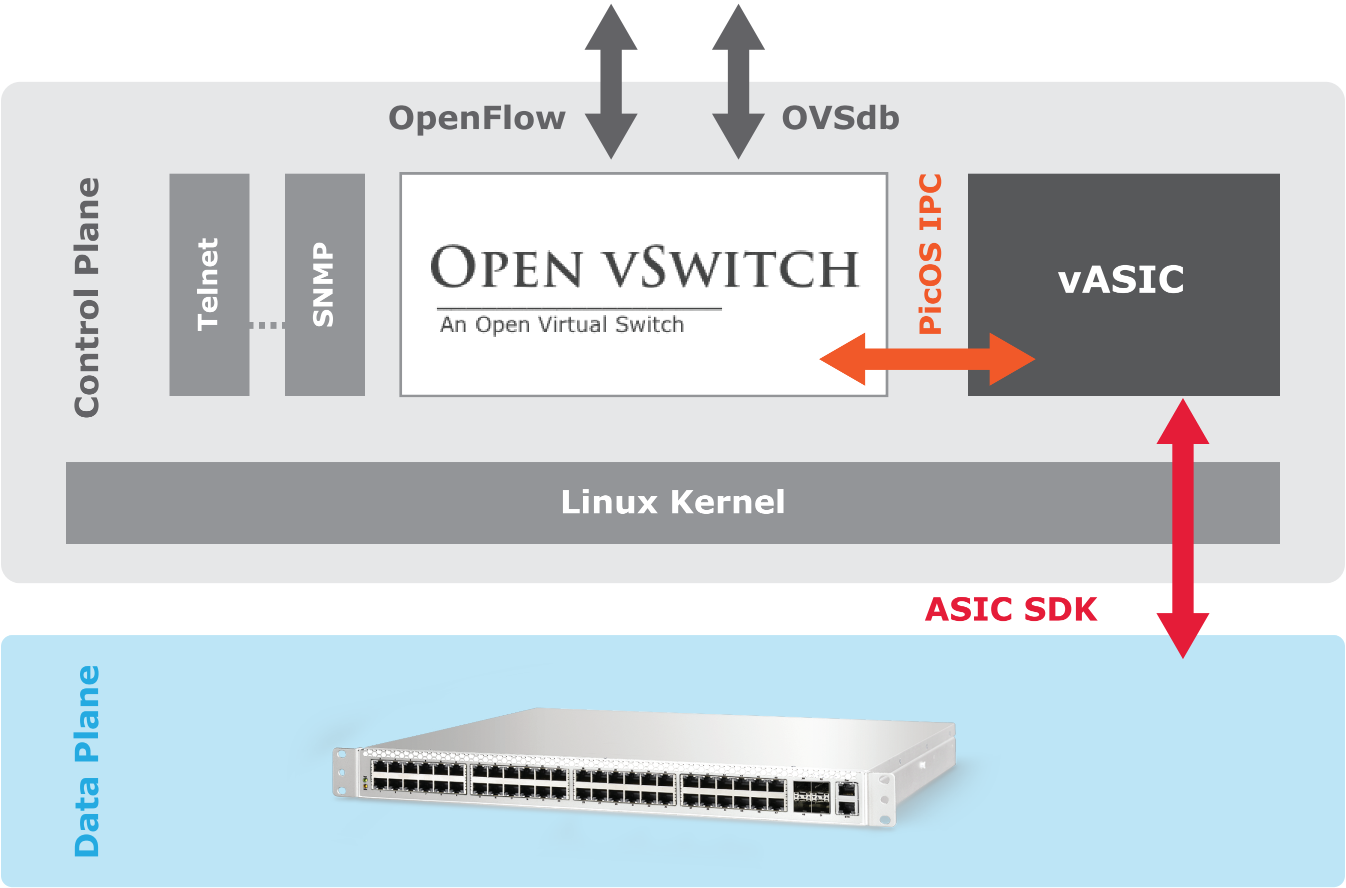
- Event-driven router implementations are needed to respond to change as quickly as possible
- Events processed to completion
- Explicit dependency tracking

5 SDN: OpenFlow and OVSDB

PicOS leverages Open vSwitch a production quality, multi-layer virtual switch licensed under the open source Apache 2.0 license. OVS runs as a process within PicOS.

The OpenFlow protocol is driven by the Open Networking Foundation (ONF), a leader in software-defined networking (SDN). The OpenFlow protocol governs three essential components of SDN: an OpenFlow physical switch, OpenFlow virtual switch to manage virtual machines, and an OpenFlow controller to organize all network pieces.

PicOS is leverages the Userland OVS implementation modified to interact with the PicOS vASIC to download OpenFlow states to the ASIC.



PicOS OpenFlow Solution:

- PicOS was the first hardware accelerated implementation of OpenFlow 1.0, 1.1, 1.2, 1.3 and 1.4 (since PicOS 2.3).
- As of PicOS 2.2, Open vSwitch (OVS) 2.0 is used.
- Leverage production-ready OVS switches for your CloudStack and OpenStack projects
- Interoperable with multiple OpenFlow controllers such as OpenDaylight, Ryu, Floodlight, NOX and Trema

Networking ASICs allow line-rate, flow-based networking but they lack the flexibility of a server's CPU. This means that not all OpenFlow capacities can be used on hardware. This capacity is always improving with new ASIC and recent OpenFlow standards recognize those limitation with the use of OpenFlow table type pattern (TTP).

The detailed list of supported OpenFlow match and actions is listed here:
http://www.pica8.com/documents

6 Mixing SDN and Traditional Routing and Switching

CrossFlow Mode

In PicOS, we can use OpenFlow and traditional switching/routing simultaneously. Some typical examples of applications include:

- In an OpenFlow data center, the edge devices need to interact with traditional switching/routing devices (Spanning tree, OSPF, BGP, …)
- In a traditional data center using switching and routing, monitoring and tapping can be done on the switches and rules triggered by the OpenFlow protocol.

The "CrossFlow" mode is what we call the mode of operation using both OpenFlow and L2/L3 on the same switch.

Every port in the switch is either legacy or a CrossFlow port. A legacy port is using traditional L2/L3, while a CrossFlow port can be controlled by OpenFlow. All the typical OVS commands can be used in CrossFlow mode (either locally via OVS commands or via an SDN controller using OpenFlow and OVSDB).

A CrossFlow port can have local-control enabled or disabled. Local-control is the switch capacity to process protocol packets like BPDU, LLDP, BGP or OSPF PDU. Local-control also enabled standard switch behavior like MAC learning and packet flooding. Without local-control enabled on a CrossFlow port, MAC learning and packet domain flooding are not enabled because packet behavior is completely handled by the OpenFlow protocol.

CrossFlow port modes can be summarized as follows:

CrossFlow **mode OFF**

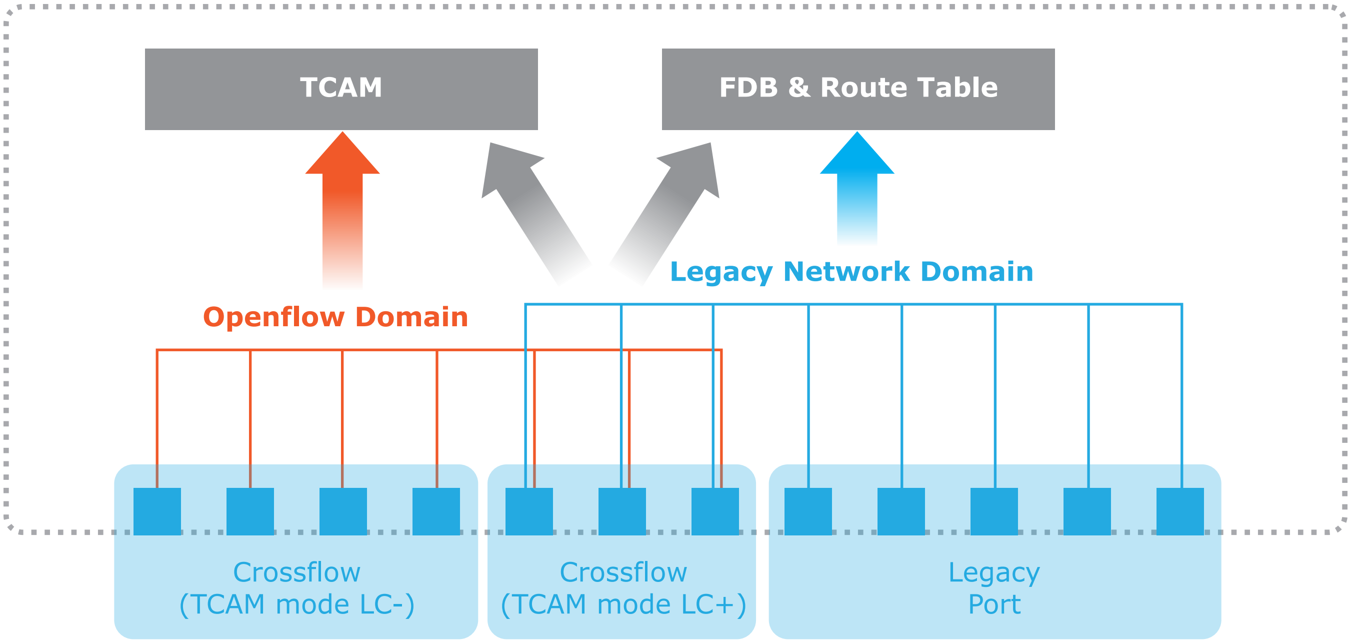
- The port is controlled by the switch local PicOS control plane
- All broadcast turned on and auto-learning turned on
- Packet forwarded by looking up the TCAM or the FIB
- PicOS controls the TCAM and FIB completely

CrossFlow **mode ON and local-control OFF**

- The port is totally controlled by the SDN controller
- All broadcast turned off and auto-learning turned off
- Packet forwarded by looking up the TCAM or the FIB
- TCAM and FIB managed by the SDN controller.

CrossFlow **mode ON and local-control ON:**

- The port is controlled by the switch local PicOS control plane
- All broadcast turned on and auto-learning turned on
- User or controller can add flows in the TCAM to control traffic



In PicOS 2.4, All the OpenFlow flows and OVSDB configurations are set using either standard OVS commands or using the OpenFlow protocols.

7 Summary

The evolution of networking to a new software led model, PicOS is the OS that can bridge the gap, and help your organization seamlessly migrate from a world of lock in to a world of freedom. PicOS has both conventional tools and APIs and new SDN technology to support any network. Whether you are a traditional network engineer or Linux guru, PicOS has the right tools for you.

Pica8, Inc.
Corporate Headquarters

1032 Elwell Court, Suite 105
Palo Alto, California 94303 USA
650-614-5838 | www.pica8.com

Pica8 and PicOS are trademarks of Pica8, Inc.

Pica8 and PicOS trademarks are intended and authorized for use only in countries and jurisdictions in which Pica8, Inc. has obtained the rights to use, market and advertise the brand. Pica8, Inc. shall not be liable to third parties for unauthorized use of this document or unauthorized use of its trademarks. References in this publication to Pica8, Inc. products or services do not imply that Pica8, Inc. intends to make these available in all countries in which it operates. Contact Pica8, Inc. for additional information.