

Independent market research and competitive analysis of next-generation business and technology solutions for service providers and vendors

**HEAVY  
READING**  
**WHITE  
PAPER**

# **Open Source: A Framework for Digital Services Modernization**

*A Heavy Reading white paper produced for Red Hat*



**redhat.**

**AUTHOR: ROZ ROSEBORO, PRINCIPAL ANALYST, HEAVY READING**

---

## INTRODUCTION

Open source provides a host of benefits to communications service providers that will help them evolve technologically as well as culturally. Engaging directly with open source communities not only allows them quicker access to innovation, but also gives them insights into how their own organizations can work more collaboratively. They can deploy open source solutions in numerous domains, including network infrastructure, data management and applications management, to become more operationally efficient.

Communications service providers can work directly with community versions of the software developed by open source communities – and many try to do so at the outset – but going the do-it-yourself (DIY) route presents numerous risks to the business and has costs that are not always apparent initially. With DIY, the service provider is now responsible for managing the lifecycle of the software, and many are not prepared for the level of work entailed and the time required to ensure that quality isn't compromised.

Related to this is the challenge of finding and retaining the staff needed to work directly with code and the potential for core business functions to suffer as resources are spent on maintaining software rather than more strategic projects. Some of the other risks of DIY include support, security and compliance and unexpected downtime. Increased technical debt becomes an important consideration when service providers make changes to the community code that gets them out of sync – a so-called "fork." This makes it more challenging and costly to incorporate the changes in community versions, negating much of the benefit of using open source code.

Leveraging a vendor-supported distribution, in contrast, offers service providers numerous benefits. With this approach, a vendor hardens and stabilizes the community version of the software and is responsible for maintaining the software over time. When the vendor is committed to upstreaming changes, the integration burden is lessened, compatibility is maintained, and overall technical debt is reduced. Service providers can therefore get the benefit of the innovation coming from open source communities without the costs and risks associated with DIY. Heavy Reading research suggests that most service providers recognize the value of a vendor-supported distribution approach.

Increasingly, service providers are making open source an important element of their NFV- and SDN-led transformations. They have seen the agility and freedom that the over-the-top (OTT) companies have achieved, and are looking to similarly benefit. Indeed, Heavy Reading research indicates that many see open source as accelerating NFV and SDN adoption. Working with strong vendor partners, service providers can readily participate in and benefit from the robust open source ecosystem and leverage the benefits of rapid innovation to help drive their business forward.

## OPEN SOURCE SOLUTIONS DELIVER VALUE ACROSS ENVIRONMENTS

Service providers who deploy open source technologies will experience internal and external benefits beyond the technology. Engaging with open source innovation expands opportunities by taking advantage of emerging partnerships and services, in addition to rapidly advancing

---

technologies. Adopting an open source approach can also help develop an internal culture of collaboration, making it easier for teams to work together to deliver new ideas more quickly. Joining an open source community of innovators can help service providers to understand the benefits from the solutions these larger groups create.

Open source solutions can be deployed across numerous service provider domains, including network infrastructure, data management and application management platforms, which can ultimately help service providers to take advantage of a framework for open digital services modernization.

## **Network Infrastructure**

Network functions virtualization infrastructure (NFVI) provides the secure and resilient platform upon which virtualized network functions (VNFs) run, making it the foundation for service provider modernization and transformation projects. The NFVI delivers a common infrastructure for all workloads, whether they are packaged in virtual machines (VMs) or containers. NFVI uses commodity hardware that can scale efficiently and be deployed consistently. The NFVI transports network data and generates telemetry data in the process of transforming the network and more valuably, during ongoing operations, by leveraging software-defined networking (SDN). Automation can enhance ROI by making deployment and management more efficient.

## **Data Management**

The data management domain is responsible for numerous processes. A data management platform captures in real time, or in batches for processing, the payload and telemetry data generated by the network infrastructure and applications. An open real-time messaging bus interconnects these platforms so that any entity can publish and subscribe to it. It transforms, distributes and stores data for applications, and provides governance and APIs to manage access to the data.

## **Application Management**

The new normal for application development and management is an always-on, continuously available service that service providers develop and manage in an agile manner. This is where DevOps methodologies, including CI/CD and automation, are executed, facilitating the transition from Dev to Test to Production environments. Application management platforms consume data from the data management platform and deliver data via network infrastructure. Containerized cloud-native applications offer higher density and elastic scale, and are composable due to their microservices architecture, enabling service providers to deploy more efficiently and get services to market faster.

## **RISKS OF DIY APPROACH**

Service providers are adopting open source software in numerous ways. Some may not be aware they are, because open source is often being used in proprietary software. While using open source in this context may provide a one-time benefit during initial development, it does not easily incorporate the ongoing innovation coming out of the open source community.

---

Another drawback with proprietary software is that the service provider is unable to view the source code, which prevents them from understanding how the software was developed and potentially discovering security issues with a DIY approach, service providers download open source software, often at no charge, directly from an open source project repository in binary or source code format. Alternatively, service providers can choose a vendor-supported distribution, which has a cost, but as discussed below, provides significant savings and benefits over DIY.

Service providers often begin using a community version of open source software for an initial proof of concept. As the project progresses, they realize that the initial software installation is only the beginning. They must consider the ongoing maintenance and support costs in order to truly understand the ROI of the community version. There are also numerous risks involved with the DIY approach, and many find that going it alone is more work than expected.

### Higher Costs With DIY

**Operational:** Service providers face significant operational costs with a DIY approach. They are responsible for maintaining the software, which includes integrating and testing updates with each new release. Quality assurance is a rigorous and time-consuming process for which many service providers are not prepared. Because of the complexity of their environments, debugging issues and retesting each merge can require significant time and be error-prone.

**Resourcing:** Hiring, training and retaining competent open source managers, architects, operations and networking staff is another important cost to be considered, because many service providers still lack many of the programming skills required for working directly with code. When asked about skill sets related to working directly with open source software, a tier 2 North American fixed operator stated:

"The skills necessary to take advantage of open source can be varied and continue to change quickly. Growing this skill set internally has been a big benefit for our company, but there is always concern that such a unique skill set can be in high demand and hard to find. This can be a barrier to rapid adoption due to the potential risk of scaling the staff up quickly if necessary. Service providers often hire contractors rather than re-train or re-tool their existing staff. This [challenge of finding skilled staff] is rarely significant enough to stop us from considering open source, but it does require that we place more emphasis on resourcing and training to minimize risk."

Open source communities are typically much larger than any single organization's development team. Leveraging the ongoing community innovation with an experienced open source vendor avoids innovating on your own. This frees up time and effort being spent on building and maintaining software that can be spent on delivering services to the market. When considering a DIY approach, service providers need to ask themselves if managing software is really the business they want to be in. If so, they need to accept that their core business may suffer as a result.

### Key Risks of DIY & Forking

In addition to operational and resourcing costs, there are additional risks associated with the DIY approach that must be considered.

---

**Support:** The open source community has no agreements to provide support in the event of an outage, no obligation to reply to requests within a certain amount of time and no single contact to call. For service providers that have stringent service-level agreements (SLAs) with customers, this is a particularly crucial and important consideration. Unless the service provider has in-house support, it must rely on best-effort help from the community, and there is no guarantee of a response that complies with service provider SLAs.

**Security and compliance:** Business risk increases when service providers are responsible for implementing their own security patches. Service providers utilizing a DIY open source software approach are likely to be dependent on the community to create security patches, leaving them exposed and reducing the time they have to react to potential security threats. Similarly, proving compliance can be challenging with in-house software maintenance.

**Unexpected downtime:** As noted above, with a DIY approach, the service provider is responsible for testing and integrating software updates. This increases the risk of potential failures and outages due to limited testing capabilities as compared to community-based testing. This is of extreme importance for running mission-critical infrastructure with potential penalties and possibly risk to human life. Outages directly impact customer access and service provider revenues.

**Increased technical debt:** There are myriad of open source packages, each with their own release schedules, that must all be brought together to create a stable, cohesive solution. If the service provider has created a fork, the challenge is increased, as once a feature is introduced into the upstream version, there is no guarantee it will work the same way in the forked version.

Forking, therefore, can introduce chaos into deployments and result in a lower-quality and less-reliable solution. This refers to the debt created by the service provider for maintaining a fork. This gets the service provider further and further behind the community or industry, and may lead to a loss of interoperability with other upstream projects. Also, the more patches and updates that are added to the now-proprietary distribution, the further the fork drifts from the upstream version. The service provider therefore cannot easily incorporate the innovation coming from the community. For example, according to Red Hat research, over two years, 33 percent of the code in the Linux kernel has changed, and over three years and six major releases, 97 percent of OpenStack's code has changed.

## **Vendor-Supported Distributions Provide Numerous Benefits**

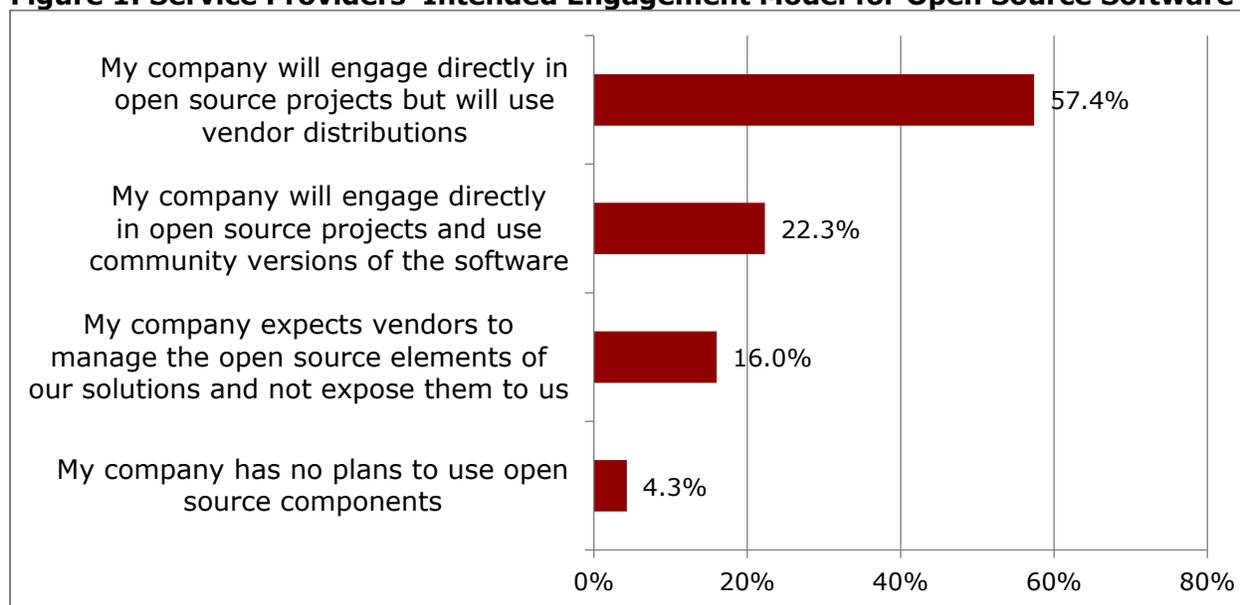
With vendor-supported distributions, service providers can offload most of the responsibility of managing the open source software lifecycle to a supplier. A Western European tier 1 integrated operator told Heavy Reading, "Anyone who has done the implementation experience knows you need support." Indeed, having a supplier handle upgrades and patches and verify compliance reduces service provider risk and operational costs.

In this approach, the supplier hardens and stabilizes the community open source version of the software, verifies its compatibility and interoperability, and releases it as an easily consumable product. Modifications to the code are upstreamed so they are available for future releases. Having a supplier committed to upstream development ensures that service providers can choose their pace of innovation, ease integration efforts and maintain compatibility over time, reducing overall technical debt.

When using a vendor-support approach, service providers can communicate requirements and urgent use cases to direct community development. Useful features and capabilities for the industry as a whole can be submitted back to the open source community by the vendor, and vendors can drive developments in upstream projects for the benefit of the service provider. In addition, service providers can review software documentation, helping ensure easy consumption and deployment. With the vendor-support approach, service providers have a robust, supported solution at a fraction of the cost of owning the distribution.

Heavy Reading's service provider survey reveals that most intend to use a vendor-supported distribution of open source software.

**Figure 1: Service Providers' Intended Engagement Model for Open Source Software**



Source: Heavy Reading

## EXPANDING OPEN SOURCE ADOPTION

Service providers generally retool their networks every five to 10 years, and many are now in the midst of their technology refresh. Concurrently, many are dealing with massive increases in mobile data, and are driving transformation efforts to virtualize their networks to improve flexibility and scalability. They are witnessing innovation from hyperscalers and OTT players, including their embrace of open source and white box hardware.

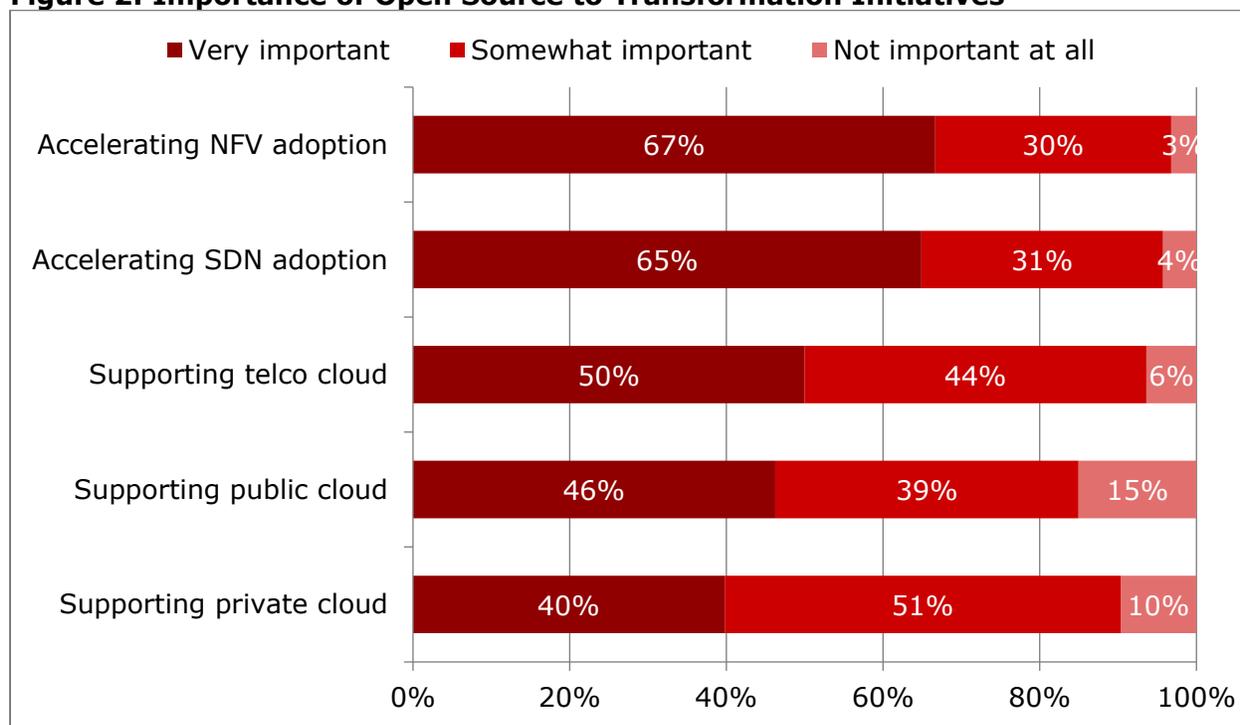
To obtain the agility and freedom from vendor lock-in the OTT players have achieved, service providers are increasingly indicating their intention to expand open source adoption for their infrastructure, as evidenced by NFV- and SDN-driven transformations. Even Google, which could have maintained Kubernetes internally, prefers to have community engagement to drive innovation. Service providers also want to take advantage of what has been developed in other industries for other use cases, such as IoT.

Consumers of open source software benefit from a host of features and functionalities that emerge from their respective projects. The projects are able to tap into the collective wisdom

of a community and innovate more rapidly by building on and reusing existing code using a dynamic and continuous development cycle. Having a diverse group of people contributing helps increase the pace and quality of innovation. Because there are more organizations and individuals reviewing and testing the code, the quality and security of the emerging code is more robust. The end result is *de facto* industry standardization and helps ensure interoperability across different solutions.

**Figure 2** shows how important service providers consider open source in their transformation efforts.

**Figure 2: Importance of Open Source to Transformation Initiatives**



Source: Heavy Reading

## CONCLUSION

Service providers are increasingly engaging with open source directly and through their vendor partners. They find that open source solutions can help them improve operational agility across their various network and system domains. While they have the option to use community versions of software, this DIY approach also brings with it a host of costs and risks, which could keep them from receiving all the benefits open source provides. By leveraging a vendor-supported distribution, they can take advantage of the innovation coming from open source communities in a lower-cost and lower-risk model and focus their energies on running their business.

Open source solutions can accelerate service providers' NFV- and SDN-led transformation. A vendor partner committed to upstream first can play an important role in helping service providers achieve the agility and flexibility that open source solutions provide.

---

## ABOUT RED HAT

Red Hat has 25 years of experience working with the open source community that it brings to its customers' modernization and transformation projects and has helped more than 90 percent of Fortune 500 companies solve business challenges, align their operational and business strategies, and prepare for the future of technology. It does this by providing production-ready solutions through an open business model and an affordable, predictable subscription model. Its open source software is fully tested and supported and has been deployed in critical use cases by telecommunications service providers around the world. Red Hat's large and diverse partner ecosystem helps support complex projects during the entire lifecycle, from strategy to deployment and production. Its leadership in open source communities enables its service provider customers to influence how key technologies are developed. Red Hat brings a broad portfolio and collaborative approach to address every challenge of its customers' digital services journey.