

Cloud-native control with RtBrick APIs

RtBrick is already known for bringing a radical new approach to carrier networks, by disaggregating telco routing systems and separating software from its hardware. Now it has extended that open cloud-native approach even further, by offering two APIs into its systems.

The APIs will allow operators to:

- Program individual switches or entire networks
- Access underlying systems with complete granularity
- Unify network and IT operations via a consistent REST-based approach
- Operate disaggregated components alongside other equipment using the same higher-level systems

The first of RtBrick's APIs opens up access into the RtBrick Management System (RBMS), which controls a disaggregated telco network from a 'single pane of glass'. RBMS is built using different microservices, such as metrics and alert management, configuration, logging and so on. All of these have their own GUIs which are now available to external systems via the RBMS API. This REST-based interface allows an operator to pick and choose which parts of RBMS it wants to be presented to the higher-level systems. For example, they may just want to use the RBMS network topology tools and use another system for event logging.

This allows an overarching system to manage multiple vendors and types of equipment alongside the cloud-native RtBrick disaggregated MPLS systems.

RtBrick also provides a switch-level API to control RtBrick's FullStack networking software – the RBFS API. The RtBrick Management System uses this API to control individual switches but the RBFS API is also exposed directly to higher-level systems

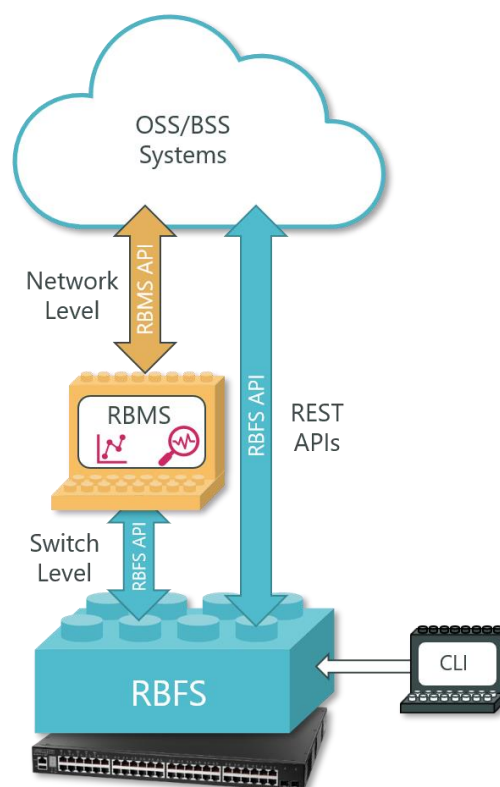
Rather than traditional configuration by CLI or using automation scripts, the RBFS API directly implements the requested functionality in the underlying switches. It takes advantage of RtBrick's architecture, which uses a single in-memory database with stored objects. So, instead of loading configuration files onto switches, the API allows direct access to the tables and objects through an HTTP interface.

The RBFS API allows you to write applications that can configure individual switches, deploy a mesh of switches into a PoD (Point of Deployments), or even write applications that can respond directly to external events.

Any system can access these new interfaces, whether it is another element management system, or an application written in Python or Go, for example. The RBFS API also comes with user authentication, which can control different privileges for different types of users.

RtBrick's REST APIs are based on a Model View Controller (MVC) design, so nothing is masked from the operator. There is no intermediate layer, so you have access to every feature that you can control with a traditional CLI (Command Line Interface).

RtBrick also provides a conventional CLI into its systems, to support more traditional operational processes. But giving operators the option of using APIs to control a disaggregated network means that we are not mandating any specific external tools or systems, and giving operators an unprecedented level of choice and flexibility,



whether they choose to use RBMS as an end-to-end management system, develop in-house controllers, or work with a third party system integrator to accommodate their specific requirements.

Whatever the preferred approach, the RtBrick APIs allow complete granularity of control, directly into the underlying systems. And adopting the same REST-based approach as modern-day IT systems is an important step towards a unified cloud-native operating environment.

More technical documentation can be found [here](#).

Why don't you get in touch with us to find out more about how our APIs can help you control your network like a cloud-native?