PacketFront Software

BECS® vs. Self-Built Management Platforms

WHY BECS IS THE SMARTER CHOICE

BECS® vs. Self-Built Management Platforms

Most, if not all, own developed systems are initially built to solve a specific problem on a specific hardware platform. One of your key engineers says 'I can solve that with a few scripts in Ansible with little cost to us.' The problem of automation seems simple at this stage, when onboarding a customer just push a few templates to the box and they are live, problem solved! However, over time cracks appear in this approach.

In the following chapters, we have outlined some of these challenges in greater detail and explained why we believe PacketFront Software's **BECS** Network Orchestrator, as a commercial product, is often the superior choice.

Automation and Simplification

Scripts necessitate manual coding for each specific task or configuration change, which can be time-consuming and susceptible to human error, especially as the network expands in size and complexity. Moreover, most script-based solutions apply configurations to the network based on templates, without accounting for, let alone understanding, the pre-existing configurations on the devices.

BECS enables network administrators to define their desired outcomes (intents), rather than focusing on the intricate details of how to achieve those outcomes. The platform automatically translates these intents into network configurations and actions, thereby reducing complexity. Before implementing any changes, **BECS** retrieves the existing configuration from the affected devices and identifies the differences between the new desired state and the current configuration. This approach ensures that the network is always optimally configured and accurately documented.

Scalability, Technical Debt and Sunk Cost

As the business grows, so do the demands on automation. New hardware platforms are added as well as new scenarios the automation must handle. This often leads to whole sections of the automation needing to be reconsidered to account for these differences. While solving these problems, over time, the Technical Debt builds so it may seem expedient to continue using the initial tools and codebase chosen rather than performing a full rewrite in a more appropriate system due to ballooning costs. Also, systems that rely say on scripting technologies as python at their heart can become slow and have difficulties scaling due to the nature of interpreted languages.

Multi-vendor support

In home-grown solutions adding new hardware vendors or models can be a tedious process as much of the solution from inception is built around the methodology of a certain vendor. While vendors need to stick to key standards the implementation can be subtly different enough that adding a new vendor can require extensive rework and lead to compatibility issues.

BECS is designed from bottom-up to support multi-vendor environment, we have seen many different vendors and their subtle differences and already solved these issues in a scalable way. **BECS** comes with existing integrations with various network devices, vendors, and technologies, so in most cases adding support for a new vendor in your network is as simple as downloading an existing Element Manager from our library. And if required, we are ready to integrate new vendors if the need arises.

Consistency and Accuracy

Home grown development is often done in an ad hoc manner, usually to solve an immediate issue which can result in inconsistencies or not provide solutions that are future proof. Many implementations apply configurations to the network based on simple templates, without accounting for, let alone understanding, the pre-existing configurations on the devices. Many times, removing a customer service becomes tricky to achieve so services are often broken rather than cleanly removed leading to orphan configuration that can cause issues in the long run.

BECS enforces consistency across the network by ensuring that all devices and configurations align with a defined intent using a well-defined framework. Before enacting changes **BECS** communicates with the device to determine the current state of the configuration and what needs to change to ensure the service is added or removed cleanly. This minimizes the risk of configuration drift, misconfigurations and orphan configuration.

Error Reduction

Writing and maintaining scripts manually involves a higher risk of errors, especially in complex network environments. Mistakes in scripts can lead to network outages or security vulnerabilities. In addition, as it is difficult to detect these errors, operators tend to believe that the existing state of the network is better than it actually is.

Automation reduces the likelihood of human error, which is a common issue with manually written scripts. **BECS** includes validation mechanisms to ensure that configurations are correctly pushed to the network. It also has a Service Assurance framework that validates that the intention of the configuration change was achieved.

Security

When using custom scripts the security depends on the quality of the script and the rigor of the development process. Scripts may lack the comprehensive security features of a dedicated orchestration platform, increasing the risk of vulnerabilities.

BECS includes security features that assist in compliance with security policies, such as TSA and NIS2, and help to prevent unauthorized access or configuration changes. **BECS** security features include:

- Role based **user rights** (who is allowed to do what in the network?)
- **Audit logs** (who did what and when?)
- **Encryption** (device passwords and other sensitive information)
- **Network audits** (check the network for tampering / unauthorised actions)
- **Automated FW upgrades** (ensure fast roll-out of security patches)

Most importantly, the security is never a completed task. As a commercial product **BECS** is constantly updated with new and improved security features.

Cost-Effectiveness Over Time

Where as initially, developing in-house solution may seem cost-effective, but over time, the costs associated can outweigh the initial savings. Own developed solutions based on open-source are often even regarded as being free-of-charge. However, the development cost, and especially the maintenance can be very expensive due to changing requirements on integrations, new hardware vendors and scaling as the initial design did not anticipate the required flexibility.

While there is an upfront cost for implementing **BECS**, the long-term savings in operational efficiency, reduced downtime, and lower maintenance overhead can be significant.

Future-Proofing

Scripts may require significant rewriting or adaptation to incorporate new technologies or standards, making it harder to future-proof the network.

BECS is designed with future technologies and advancements in mind, allowing for easier adoption of new features or standards.

PacketFront Software

Head office

Street:
Vasagatan 10, 111 20,
Stockholm, Sweden

Postal address:
P.O. Box 575,
SE-101 31, Stockholm

Phone:
+46 8 633 1990

Email:
sales@pfsw.com
info@pfsw.com

UK representatives

Phone:
+44 7718 175 652

Email:
sales-uk@pfsw.com
info@pfsw.com

Poland office

Street:
Jana Pawla II 22,
00-133 Warszawa, Poland

Phone:
+48 22 487 56 25

Email:
office@poland.pfsw.com
info@pfsw.com

www.pfsw.com