How homegrown BIND DNS drags down digital transformation…

Starting a digital transformation journey can be pretty exciting. Cloud! Virtualization! SD-WAN! DevOps! Suddenly the sky's the limit. …

Starting a digital transformation journey can be pretty exciting. Cloud! Virtualization! SD-WAN! DevOps! Suddenly the sky's the limit. Everything's possible. All of those new capabilities, all of that new flexibility – the IT enterprise moves to the forefront of organizational change. It all takes time and proper planning to implement well, but the rewards are clearly worth it.

At some point, however, the honeymoon period of any digital transformation process is bound to wear off. All of those huge leaps forward are amazing at first, but in time there are bound to be some operational consequences to all that change.

With over twenty years' experience in helping large enterprise customers through the digital transformation processes, BlueCat knows that the DNS infrastructure which most enterprises take for granted is often overwhelmed by digital transformation efforts. If there's one thing that strains internal processes during a cloud migration, grinds DevOps to a halt, and prevents a strong security posture across hybrid environments, it's usually DNS.

All of the advantages of BIND – its simplicity, the ability to customize, its accessibility – become disadvantages when applied at scale across a complex network environment with datacenters, hybrid architectures, cloud, and a suite of network tools.

Here are some examples of how architectures relying on homegrown BIND implementations can strangle digital transformation, taken directly from our experience with customers.

The deluge of DNS service tickets

BIND uses a decentralized architecture to manage DNS. All of the DNS servers and DNS zones are updated manually – there's no built-in method to push changes out across the enterprise. It's the network team that has to get all of this done, day in and day out. In addition, changes to homegrown BIND solutions often require changes to non-BIND infrastructure tools like Microsoft DNS and Route53. All of this adds an additional level of complexity which requires network teams to use multiple administration consoles and unique configurations.

This is all fine and dandy when the network is simple. Yet as the network grows, as new layers of functionality call on DNS resources, and as new environments are added on, it becomes clear pretty quick that managing all of these nuances highlights the challenges associated with de-centralized BIND administration is a huge effort.

All of the simple tasks network administrators do every day like adding host records suddenly become unmanageable – the volume is just too much. Surveys on this issue show that network administrators spend around a third of their time managing DNS service requests. When you're trying to move to the cloud, implement DevOps, virtualize environments, the network team shouldn't have to take this much time to deal with DNS.

Of course, there are two sides to every service ticket. When the network team is overwhelmed, it can't deliver on its SLAs. When that happens, even simple requests can take forever to get done. This slows down cloud and DevOps teams, or (even worse) drives them to stand up their own shadow IT systems which further increase the risk of downtime. The decentralized management architecture of homegrown BIND infrastructures is the root cause here, but the ripple effects play out across the organization in unpredictable (but highly disruptive) ways.

Single point of [human] failure

One of the advantages of BIND is that it is infinitely customizable. The foundation is pretty simple – if you know your PERL, Shell, or Python well enough, you can build just about anything on top of it.

When networks are first created, it's usually a single person (Mister DNS, the DNS Queen, whatever you want to call them) who starts putting together those scripts to fulfill basic needs. Over time, the number of customized scripts and configurations starts to grow. This is a problem for two reasons.

First, all of those scripts and configurations have to be maintained. BIND isn't static – as the underlying code base is patched and updated, all of the code built on top of it has to be patched and updated as well. This isn't much of a problem for small networks, but when things start to scale and get more complex, just keeping your head above water can be a major challenge.

Second, over time Mister DNS evolves into a single point of failure for the entire DNS infrastructure. If Mister DNS decides to leave the organization, it's likely that nobody knows how to maintain the delicate lattice of customizations built up over the years. Even if Mister DNS stays, you're at the mercy of that one person's bandwidth to get things done. We've been to plenty of customer sites where the whole organization revolves around one person, and they're praying that that person stays long enough to keep things up and running.

Another option is to expand the number of DNS Queens, but that just means that more of your IT team is devoted to keeping DNS operational – hardly a sustainable solution in the long run. Every person who drifts into maintaining BIND scripts is a person who can't work on more important things.

This problem isn't exclusive to BIND – it frequently plays out in Microsoft DNS as well – but the fact that BIND doesn't seamlessly integrate with IPAM solutions layers on a significant challenge and makes administration much worse. (We wrote a whole eBook on the challenges of Mister DNS.) Which leads us to the next reason…

Split-brain DDI

BIND will give you the basics for DNS, but after that you're on your own. It doesn't support DHCP or IPAM at all.

BIND's narrow focus on DNS often results in an organizational split in how the other two legs of the DDI footstool are handled – a decision that usually comes back to haunt the enterprise several years down the road. When different teams handle DNS, DHCP, and IPAM, there's often little incentive for them to cooperate on bringing these critical network services under a single umbrella later on. Once these fiefdoms are created, they can be difficult to undo.

Many organizations with BIND DNS try to fumble their way through with an overlay solution. Others cobble together a series of smaller tools to build a patchwork DDI infrastructure. Yet in our experience, both of these strategies just end up delaying the inevitable.

At some point, BIND will become too unwieldy or expensive to support. The lack of integration between DNS, DHCP, and IPAM will cause significant operational problems and result in frequent downtime. The whole system will bring initiatives like cloud, virtualization, and SD-WAN to a standstill. When the time comes to change to a purpose-built, unified solution for DDI, the transition is painful on both a technical and an organizational level.

Lack of functionality

BIND wasn't built with today's complex networks in mind. It was created as a basic, single-use DNS platform for small, simple networks.

Trying to roll out network automation? BIND DNS doesn't support that. You can build scripts which function as a *de facto* alternative, but it would be a lot of up front work. And as mentioned above, then you have to maintain all of that custom code over time.

Moving to the cloud? BIND DNS will technical work, but the tangle of connections and conditional forwarders gets unmanageable pretty quick. Plus, BIND DNS was never meant to integrate with the DNS services native to public cloud environments. That means that you have to maintain two DNS architectures instead of one – more if you're in a multi-cloud situation.

Implementing DNSSEC? It can be done in BIND, but not easily. Every DNS server has to be configured separately – there's no way to roll out DNSSEC configurations across the enterprise. And as we've mentioned previously, you have to constantly update the system to avoid an unintentional outage.

The list goes on and on – if you're doing anything remotely complex on your network, BIND is probably not a sustainable solution. It just wasn't built for the wide variety of use cases today's large-scale enterprises encounter on a daily basis.

Taking the next step

If you're struggling to maintain your homegrown BIND DNS architecture (or infrastructure) in the face of a digital transformation project, most of this is probably old news. If that digital transformation project is still on the horizon, there's probably still time to get your DNS infrastructure in order before all of these problems start to snowball.

At BlueCat, we've helped some of the world's most critical and challenging networks to a unified DDI system. Unlike some other DDI vendors out there, we take the time to really understand your infrastructure and your business goals before we make a change. That's why we have an excellent record of seamless migrations with zero downtime, and it's also why our customers really love us.

Learn more about BlueCat's DDI solution and our experienced team of DNS experts.
Published by Ben Ball

Ben Ball is the Director of Strategy and Content Marketing at BlueCat. Ben served for ten years as a Federal employee, with three tours as a Foreign Service Officer (Saudi Arabia, Turkey, Jordan), and five years at the Department of Homeland Security, where he focused on immigration issues. A graduate of the Fletcher School of Law and Diplomacy and Pitzer College, Ben lives in the San Francisco Bay Area.