Don’t Rely on Mister DNS

BLUECAT™

Meet James Madrone.

Madrone is an IT administrator for a well-known regional insurance company. He's been with the company for quite a long time. He knows the network inside and out, mostly because he built much of it from scratch.

Plenty of network administrators and IT personnel fit this profile, but Madrone is different. He's actually quite special. He holds a title which none of his peers can match. He has specialized skills and knowledge which make him a sort of guru on the network team.

Madrone is Mister DNS.

While he's justifiably proud of this title, Madrone didn't always have it. When Madrone joined the company, the network was much smaller and simpler than it is today. Back then, Microsoft DNS did just about everything that Madrone and the company needed it to do. Madrone was one of several administrators who worked on DNS. It didn't take up that much of their time, really. Nobody complained about functionality. In fact, nobody really noticed DNS at all. It just sort of operated in the background.

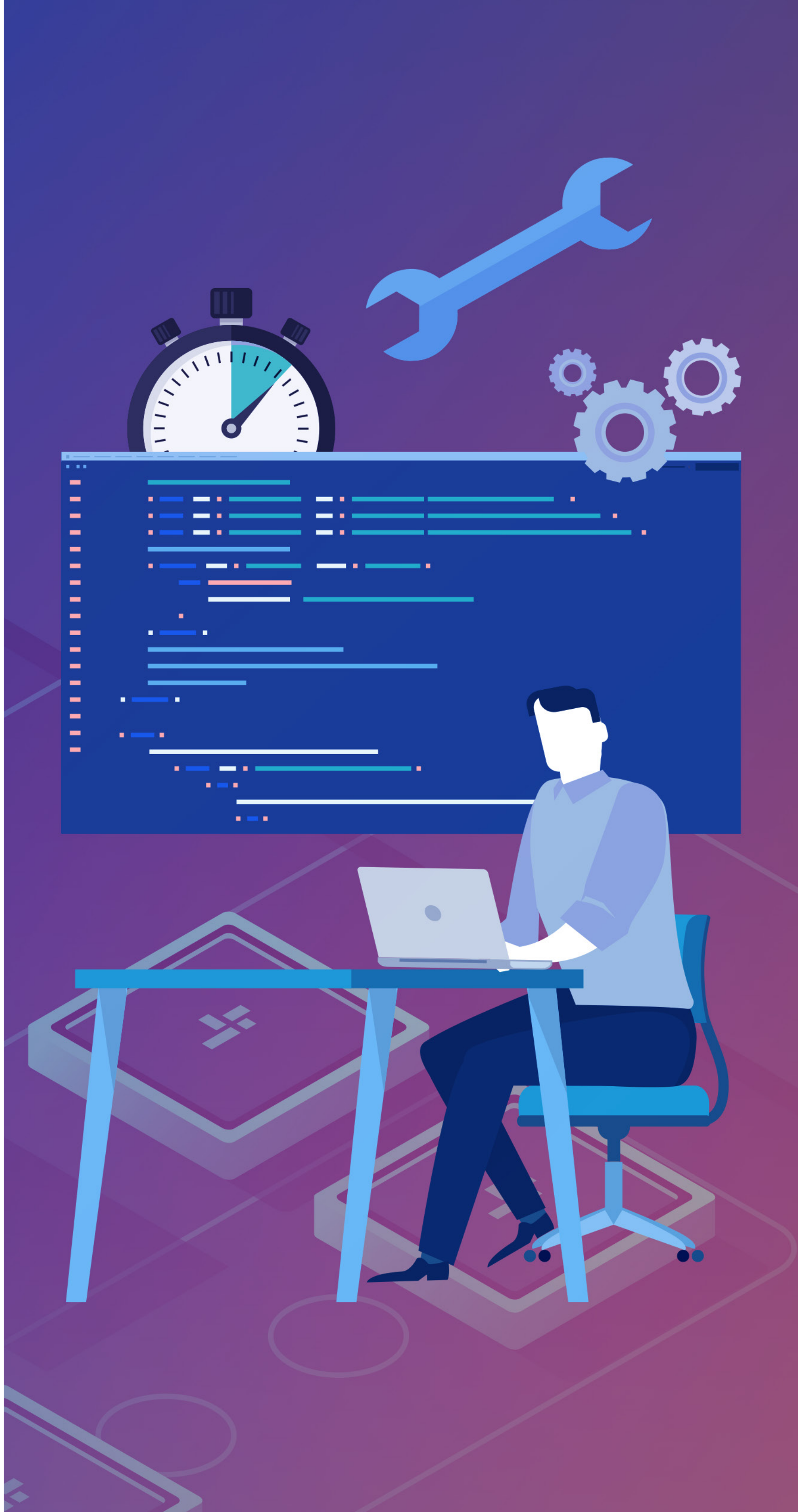
Over time, though, people started asking the network team for things that Microsoft DNS couldn't do. They weren't major asks. A tweak here, a tweak there. Madrone was usually the one who would get the job done. He was helpful like that.

At a certain point, Madrone started writing scripts to help him work through some of the inherent gaps in Microsoft DNS. Nobody asked him to do it. It's just something he did to make his own life easier, and to make the system run better. No big deal.

As time went on, Madrone found himself entirely immersed in writing, modifying, and troubleshooting Microsoft DNS. His portfolio of fixes and patches soon became a system unto itself. He used to do lots of other network tasks. Not anymore. Now he is Mister DNS full time.

Most network teams have a Madrone.

Mister DNS is the guy you turn to whenever you need an IP address, a host record, or a DHCP lease. He is the source of all knowledge, which can be both a good and a bad thing. In this eBook, we'll delve into the world of Madrone and see what happens when your DNS depends on a single person.



What if Madrone gets Hit by a Bus?

Madrone is an indispensable employee, which is both good and bad.

✓ Good If you need anything related to DNS, you go to Madrone, and he will help you out. His encyclopedic knowledge of how the company's DNS operates allows him to answer even pretty complicated questions just off the top of his head.

✓ Good If Madrone can't answer your question or can't support what you're trying to do, he will write a new fix or patch to his system. Since he's seen just about every DNS use case in the organization, he's not easily stumped. He can usually find a workaround, even if that workaround comes with the occasional cost to system performance.

✗ Bad Nobody else knows what Madrone knows. So if Madrone is hit by a bus, or decides to retire, or moves to another company, the entire organization will really be up a creek without a paddle. In essence, the entire DNS architecture would be frozen in place because nobody would dare to touch it. They would be afraid of breaking everything that Madrone has created over the years. (That's not an empty fear, either – that system Madrone created involves a series of dependencies which could collapse with the slightest push. Even Madrone brings down the network every so often with a "fat finger" change.)

✗ Bad Madrone is a bottleneck when network changes have to be pushed through fast. Since Madrone is the only person who can be trusted with any system changes or updates, you basically have to wait for him to get to your DNS service ticket before your project can proceed. And needless to say, Madrone is a pretty busy guy. It might take him days or even weeks to get to that task, even if it takes him two minutes to actually do. When Madrone goes on vacation, things really start to back up.

✗ Bad Madrone created the DNS architecture based on his own particular quirks and ideas about how a network should operate. Nobody ever sat down with him and explained where the company's technology strategy was going. Nobody ever thought about how large-scale strategic initiatives impacted DNS. Madrone isn't really thinking about the cloud, SD-WAN, automation, or anything else that the rest of the network team does. He just does his thing in isolation.

✗ Bad Like many network engineers, Madrone is pretty set in his ways. He knows how he's always done things, and that those processes have worked pretty well up to this point. As his reputation as Mister DNS has grown, so has his reputation for ornery exchanges with other members of the team around things they want to change. He knows that he's the only repository of all that institutional knowledge, and isn't shy about making that fact known when push comes to shove.

What do Madrone's stakeholders think about the service he provides?

Madrone definitely has a reputation around the office, and it's not as stellar as he'd like. He's known throughout the organization, but not in a good way. In fact, his name is taken in vain on a regular basis by colleagues on the network team, end-users, and application developers.

When he started tinkering with DNS, Madrone had a much better profile. He helped people, and they were grateful.

Yet as the network grew more complicated around him, as the quantity and complexity of stakeholder demands grew, as his ability to respond quickly started to diminish, Madrone simply couldn't keep up.

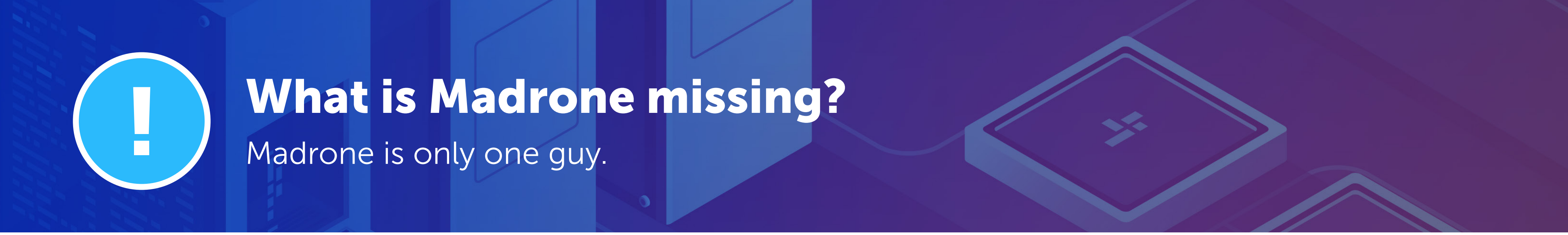
> *"We have to move away from the custom, one-off, manual DNS that we have now. It's a little brutal, and it's kind of all on Madrone."*

Mostly it's a bandwidth problem – there's only one Madrone, and the heap of DNS service tickets keeps growing. People who just want something simple like a new host record have to wait far too long. Madrone gets blamed for not getting the job done, even when there's simply too much work to do in a single day.

Increasingly, it's also a problem with the tools Madrone has at his disposal. Madrone became Mister DNS by making the system do things it wasn't necessarily designed to do. Now stakeholders throughout the organization think that he can just snap his fingers and make things happen. When you talk about complicated challenges like traffic steering, collecting DNS logs, or managing DNS in the cloud, Madrone doesn't have an answer because Microsoft DNS doesn't necessarily support these things in the way people think it does.

> *"It's just a matter of sheer quantity of requests. It's about knowledge levels across hundreds of developers. They just don't understand DNS. Madrone needs a system to make sure they're going down the right path."*

Over time, Madrone earned a reputation for being unreliable. It's too bad, really. Maybe even unfair. Madrone works harder than he ever has in the past, just to keep his head above water. Yet even his best isn't good enough, because the demands are simply relentless.



What is Madrone missing?

Madrone is only one guy.

Shadow IT

Madrone's reputation for unreliability has ripple effects throughout the organization. People know they can't count on Madrone to get things done on time, so they're starting to devise ways of going around him. This move towards "shadow IT" has a big impact on Madrone as well as the rest of the network team.

Take the DevOps team, for example. They're building and testing applications in the cloud on an agile cycle. They can't wait a week for Madrone to assign them an IP address – there are deadlines to meet. The solution: the DevOps team has secretly set up their own small DNS server in the cloud. They have their own DNS person who can act quickly, taking care of any DNS needs in minutes.

In a sense, the DevOps team is doing Madrone a favor. These are service tickets which he doesn't have the time to answer anyway.

At the same time, this strategy also has significant downsides. If the DevOps team uses an IP address that Madrone has already assigned to another device, this conflict could easily bring down the network. Even worse, tracing the source of the network outage would be extremely time-consuming, since Madrone and the DevOps team don't communicate on any of this. Madrone doesn't even know that a piece of the enterprise is outside of his control.

Strategic Initiatives and Innovation

Madrone also represents a tactical roadblock for the CIO and his vision for the network. Specifically, there are a series of strategic initiatives on the horizon which Madrone and his patchwork DNS simply can't support.

The CIO wants a **hybrid cloud** environment, but the layer of complexity that will add to Madrone's day-to-day workload would be crushing.

Network administrators want to sync their DNS to an **SD-WAN** system, but Madrone's system of patches and fixes simply can't support automation at this level of speed and sophistication.

Other network administrators would love to integrate Madrone's DNS into the system management tools they use every day – ServiceNow, Cisco ACI, Cisco DNA Center, Splunk, and other dashboards. Building these **integrations** would take Madrone years to accomplish.

The company is contemplating several **acquisitions** which will require Madrone to incorporate fully formed networks into his tangle of DNS patches – a time-consuming (and maybe impossible) task.

Security

Madrone and the Security team don't exactly get along.

From Madrone's perspective, the security people are always asking for difficult or impossible things. They want comprehensive logs of DNS activity, not realizing that it takes Madrone hours to grab the data from servers across the company. They want to know about the prevalence of DNS-related exploits such as DNS tunneling or domain generation algorithms. Madrone has no way to track these things. They want Madrone to implement DNSSEC, but doing that requires Madrone to painstakingly configure every server.

From the Security team's perspective, Madrone is leaving DNS wide open for exploitation. There's no encryption. His system can't support multi-factor authentication. All those DNS queries are going out with no filter, no visibility, no control over what resolves. 91% of all malware uses DNS – shouldn't Madrone be doing something about this?

On their own, each of these areas would be a distinct challenge for Madrone and the Rube Goldberg DNS he built. If implemented all at once, they would pose a serious threat to network stability. The CIO, the Security team, his networking colleagues – all of them are charging ahead with strategic initiatives, knowing that their end-users need new functionality. Yet Madrone and his stakeholders are going to start feeling the pain very quickly.

The Breaking Point: Managing Risk

Every organization with a Madrone eventually hits a breaking point.

The risk of using a single point of failure for DNS simply becomes too much. Simple DNS tasks don't get done on time. Shadow IT brings down the network. Cloud implementations become too complicated to manage. Security loopholes proliferate.

Mister DNS can do the best possible job, but at a certain level it will never be enough. The DNS tools he depends on simply aren't up to the task. And if Madrone himself suddenly leaves, the situation will deteriorate even faster.

Prudent CIOs and IT administrators manage systemic risk. As networks grow and become more complex, Madrone and the DNS he runs become a threat to stability. They can't deliver needed functionality. Reducing this risk means changing the DNS infrastructure which made Madrone into Mister DNS in the first place.

What is the best way to manage the systemic risk posed by a single point of failure for DNS? Here are some simple yet powerful ideas to move towards a more sophisticated, functional, future-proof network architecture.

Self Service - Let a thousand Madrones bloom.

If only one person can handle your DNS, you're at the mercy of that person's skills and bandwidth. It should be the opposite – everyone who uses DNS resources should be able to gather and manage what they need on their own, without the need to run everything through a single point of failure. Self-service is the ultimate way to manage DNS – it allows people to get what they need, when they need it.

Centralized DDI – Establish a single point of truth.

Paradoxically, one of the main challenges of using Madrone to manage Microsoft DNS is that he doesn't have all the necessary data in one place. While all the data flows through him, he's actually gathering it from servers across the network and juggling IP space with manual spreadsheets. Scaling DNS, DHCP, and IPAM (DDI) requires a single database which eliminates "fat finger" errors and provides a common baseline for higher level functionality.

Automation – Let software pick up the slack.

A large chunk of what Madrone does on a daily basis can be described as menial. He adds, deletes, and alters host records. He assigns IP addresses. He manages DHCP scopes. Most of these things should be automated. Doing so would save a lot of Madrone's time, and provide the services end-users need in a fraction of the time.

Security – Lock down DNS.

Security is an afterthought for Madrone, and doing what his colleagues on the Security team want is often a significant challenge. Implementing seemingly basic security protections –

DNSSEC, encryption, multi-factor authentication, role-based access – is a significant challenge for a Microsoft DNS that was never designed to do any of these things. Only a purpose-built DDI solution, applied across the entire enterprise, can deliver this kind of functionality.

> *“DNS is Madrone’s primary focus. We’re hoping he doesn’t get hit by a bus, because then we would be in real trouble.”*

Advanced functionality – Go beyond what Madrone can patch together.

Madrone knows a lot about DNS, but he’s fairly constrained by the architecture he’s used to. When cloud, SD-WAN, internet breakouts, failover architectures, and other advanced features come into play, he gets stumped pretty fast. Microsoft DNS works well in the smaller deployments Madrone built over time, but at a certain level any enterprise will simply need more advanced functionality. Internet breakouts, dynamic failover across hybrid environments, conditional forwarding, geo-proximity steering – all of these things require something beyond a hacked-together system.

> *“I call it faith-based address management. There’s no automated way to interact with our devices. Madrone doesn’t have time to check that what’s out on the network is what we think should be there. The combination of those two is really just killing us.”*

> *“We need to move beyond the daily 4 PM provisioning hour to on-demand provisioning.”*

BlueCat Adaptive DNS

At some point, every network team needs more than just Madrone and a hacked-together system to handle DNS.

Whether it's network outages, the looming challenge of strategic business initiatives, security gaps, or simply less-than-ideal operations, every large enterprise will eventually require a strong DNS solution.

That's where BlueCat comes in.

We've helped some of the world's largest, most complex networks move beyond a single point of failure and into the world of Adaptive DNS. We've helped administrators like Madrone deliver better functionality, stronger security, and higher efficiency with our powerful DNS solutions. We've helped CIOs and IT managers deliver on strategic initiatives like cloud, SD-WAN, virtualization, and more by providing the DNS their end-users need.

Are you a CIO with a Madrone problem?

Are you an IT manager with a DNS that can't deliver for a rapidly changing network?

Are you Mister DNS and looking to get your life and job back?

Start a conversation with BlueCat today.

---

bluecatnetworks.com

United States Headquarters
1000 Texan Trail, Suite #105
Grapevine, Texas 76051
+1.817.796.8370
1.833.BLUECAT

Canada Headquarters
4100 Yonge St. 3rd Floor
Toronto, ON, M2P 2B5
+1.416.646.8400
1-866-895-6931