BLUECAT™

Best Practices Guide: DNS Infrastructure Deployment

DNS Deployment Best Practices

Whether you're standing up a new network, migrating from a legacy architecture, or optimizing existing infrastructure, there are common best practices for deployment of the Domain Name System which any network administrator should keep in mind.

This eBook goes through some of the basic principles, considerations, and tradeoffs which every network or DNS administrator should consider. Some of these considerations are derived from industry standards such as NIST 800-53 or the CIS SANS framework. Others come from over twenty years of building and deploying DNS architectures across our diverse customer base.

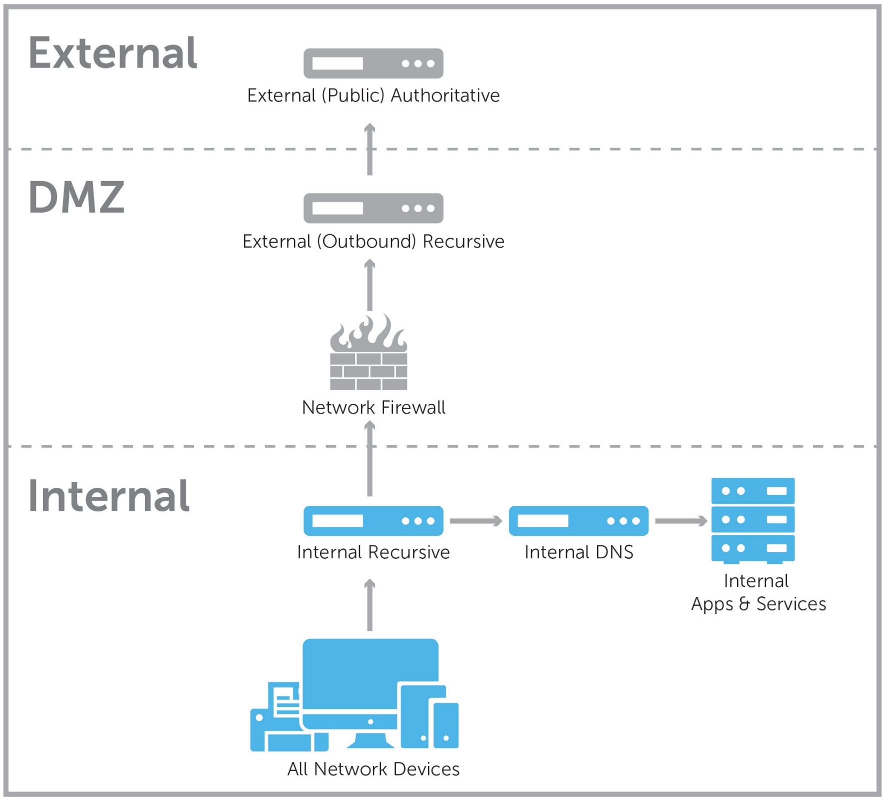
No two networks are alike. In building out DNS, DHCP, and IPAM infrastructures, network administrators make intrinsic choices and tradeoffs based on the operational requirements of their organizations. This document is designed to pose the questions which administrators will have to ask as they move toward enterprise-level DNS solutions.

This eBook starts off by looking at DNS layer by layer, covering the different considerations associate with internal recursive, internal authoritative, external recursive, and external authoritative DNS. Then we move to DHCP, looking at the various deployment options and their operational impact. Finally, we move through a hodgepodge of other DNS-related deployment considerations, including interaction with third-party services.

DNS Layers

Internal Recursive

In a best practice DNS architecture, clients point to the internal recursive DNS layer for DNS resolution. In this layer, the client queries the internal recursive layer which will recurse (either internally or externally) to find the result. That result is then cached locally for the given "time to live" (TTL). Since the results are cached locally and this is the client's first (and only) point of contact with the DNS service, most of the capacity (queries per second) will be here.



Typically, this layer contains no actual DNS data – it is just a configuration. The internal recursive layer performs lookups if any other DNS service component fails. It should also be noted that the terms "caching layer" and "internal recursive layer" are often used interchangeably.

Service points in BlueCat DNS architectures

In BlueCat's DNS offering, the internal recursive layer includes a service point. This service point collects all of the DNS information coming off of the client computer and compiles it in a searchable format. Having this service point in the internal recursive layer is not only valuable for basic visibility and forensic investigation purposes. It also enables several higher-level functions such as traffic steering, implementation of security policies, conditional forwarding, and many others.

Internal Authoritative

The internal authoritative layer is comprised of servers which contain a definitive answer for particular DNS zones. Internal recursive layers resolve DNS queries against these authoritative backdrops. DNS receives responses by having the internal recursive layer use stub zones to resolve against the appropriate DNS servers for the DNS zone in question.

When the internal recursive layer loads the stub zone, it does a round-trip time (RTT) check to each of the authoritative servers. This is to discover the relative response time to each of the authoritative servers. The internal recursive server will then note internally what those RTT times are and will query the fastest responding server.

As queries are sent to the fastest server, eventually that server's "RTT score" will slowly get worse and it will occasionally try the slower DNS servers to refresh the RTT score for that server, too. This mechanism ensures that the network is utilized to provide the fastest service available to the client.

How RTT scores work

As an example (using example.com), the internal recursive server will resolve against the stub servers it knows are authoritative for example.com (say, ns1.example.com and ns2.example.com) and perform an RTT check against them. Let's say ns1 has an RTT of 10ms and ns2 has an RTT of 15ms – the internal recursive server will use ns1. Over time, the RTT of ns2 will decrease to be lower than ns1 and a query will be sent to it which will refresh its RTT time to the appropriate level.

Why should internal recursive and internal authoritative layers be separated?

Failover

The main reason the internal recursive and internal authoritative layers are separated has to do with failed authoritative queries in the internal recursive layer. It is typical in smaller systems to have recursion and authoritative on the same box. In this arrangement, if a client receives a SERVFAIL or NXDOMAIN notice from the queried server, the client will be unable to connect until the problem is fixed – a potentially significant problem if that server happens to be the main zone.

By separating the recursive from authoritative layers, the recursive layer will know that a particular authoritative server is in a failed state and will try the other servers available to it. The recursive layer will also mark the failed server (and that one zone only) as failed for one hour – after which it will try again.

Resilience

Another advantage of separating the authoritative layer from the recursive layer is protection from human error. If an administrator accidentally deletes a zone or server from the authoritative layer mistakenly, many of the common records will be cached for the duration of the TTL at the internal recursive layer, helping to mitigate impact while the change is rolled back.

Flexibility

Having separate internal recursive and authoritative layers will also provide advantages to companies that perform frequent mergers and acquisitions. Islands of authority can be kept separate, but DNS can appear integrated to clients very quickly by having the appropriate stubs in place. This will allow for smooth and seamless conversion from the acquired company's DNS authoritative namespace into the acquiring company's DNS authoritative namespace because all clients are still pointing to their original DNS server IP addresses.

Ease of Use

Finally, having an internal recursive layer to cache DNS results makes it very easy to troubleshoot whether an item is in cache or an error. With an internal recursive layer in place, administrators can check the DNS cache (internal recursive) against the authoritative source (internal authoritative).

The only drawbacks to having an internal recursive layer are the perception of added complexity (although it offers more functionality), and the cost associated with more machines (physical or virtual). In BlueCat's experience, the benefits of a separate internal recursive layer are significant, even for smaller enterprises. The added cost and management effort is well worth it.

Wildcards

DNS slaves should also typically be local rather than remote, as the internal recursive layer will need to resolve internal queries against the authoritative servers. Resolving those queries over long distances introduces unneeded latency.

It is possible to mix internal recursive and authoritative layers on a zone-by-zone basis.
An example would be a remote location which frequently loses connectivity to headquarters. In the absence of the budget to have two boxes locally (i.e.: a slave nearby), doing a stealth-slave transfer of critical zones would ensure local survivability if cut off from headquarters resources. Even so: if this particular site is cut off from headquarters, will it work?

Summary: Should internal recursive and internal authoritative layers be separated?

Pros

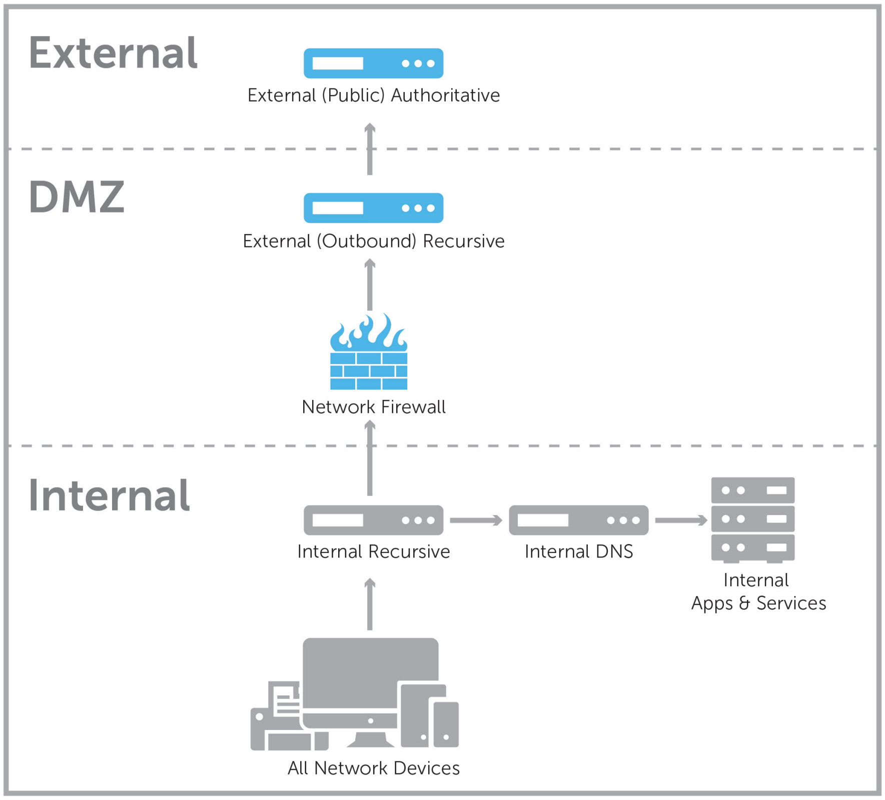
- Fail transparently
- The atomic button
- Mergers and acquisitions
- Simplified troubleshooting

Cons

- Cost
- Larger physical footprint
- Perceived complexity in the DNS environment

External recursive

The external recursive layer handles outbound requests to the internet. When the internal recursive layer cannot locate an authoritative answer to a DNS query, it will forward the query to the external recursive layer for resolution out on the internet. Since these units will only ever see internet-bound queries, the external recursive servers should reside in the DMZ.



Why implement an external recursive layer?

DNSSEC: The first advantage is DNSSEC validation, where the only queries validated for DNSSEC are outbound internet queries. Security is a significant consideration for the external recursive layer. DNS responses are essentially an invitation to reach back into a network. If those responses are hijacked by a malicious actor, use of an external recursive layer will “trap” that response in the DMZ rather than lead directly back to the internal DNS namespace. Since access to internal DNS namespace can be used to create a network map by deciphering DNS naming policies, it is very important to create a security layer to guard against threats to this area of the network.

Performance: When implementing an outbound recursive layer, capacity is rarely a concern. Most of the internet queries will be handled by the internal recursive layer, where they are cached. Aggregating the cache will also help with future capacity concerns.

Ease of Deployment: Implementing external recursive DNS is very simple from a configuration standpoint. Simply add the two servers into the BlueCat Address Manager (BAM), set up recursion for those servers, and add a forwarding option to the internal recursive layer.

Wildcards

In a highly distributed environment where internal recursive layers are funneling back through a particular egress point, the source IP address of the query will be that of the egress point. For example, if there is an office in Mexico, with the external recursive function being in the USA, when google.mx is resolved, google.com will be served because the egress point is the USA.

Summary: Should external recursive and internal recursive layers be separated?

Pros

- Security – Compromised DNS responses will only get to the DMZ
- DNSSEC validation
- Cache aggregation improves performance
- Easy to implement

Cons

- Cost, although minimal

External authoritative

For security reasons, the external authoritative layer is separate from all other layers. External authoritative servers are typically a primary attack vector. If compromised, network administrators will want to limit access to the internal network. With this in mind, it is a standard practice to separate out the other three DNS service layers from the external authoritative service.

Summary: Should external authoritative be separated from all other layers?

Pros

- **Security** – if compromised, there will be no internal data breach.
- **External DNS is extremely high**-risk, as it often reflects the brand. As such, keeping the data segmented and in its own configuration is advantageous.

Cons

- **Cost,** although minimal

It is acceptable to combine the external recursive layer with the external authoritative layer, although it may expose the connection between internal DNS servers and records.

DNS Masters

In large enterprises, it is common to have more than one master server. Since data is written to DNS masters and the capacity profile is different from all other DNS servers, performance considerations often lead to the use of more than one master.

In this scenario, queries-per-second are much higher than DNS writes – queries-per-second rely on data in memory, whereas DNS writes rely on disk I/O. If there is a sustained average of more than 500 DDNS updates per second, additional masters should be considered. This is typical only for the largest environments.

> There is a way to increase the DDNS disk-write performance by a factor of about 10. The risk of this approach, however, is a small amount of data loss in the event of a hard shutdown.

When deciding the scope and deployment model for DNS masters, administrators should ask:

- How big is the data set?
- How dynamic is the data set?
- How is the data set spliced?
- What about backup for the hidden master?
- Will xHA be used?

The answers to these questions will help to locate the logical split point of DNS masters deployed on the network.

There are a number of ways to find the logical domains for DNS masters. Here are some common splits:

- Forward space
- Reverse space
- ZTMs (zones that matter) or HDZ (high dependency zones)
- Dynamic-only zones
- Regional
- Business Unit

Redundancy

When all four layers of DNS are deployed as outlined above, only one layer truly requires high availability (HA) – the DNS master(s). The DNS master does not have a shared database, and in the event of a hardware failure, the enterprise still needs to be able to make DNS changes.

BlueCat recommends the use of anycast on the internal recursive layer if the network is capable of it. Anycast provides the ability to have an active-active configuration with the ability for low-cost load balancing if required. If anycast is not an option, then HA should be considered in its place.

To maximize performance and minimize the possibility of outages, BlueCat highly recommends that DNS servers not be used for any other network function. The internal authoritative layer and external recursive layer should consist of only stand-alone units. The recursive layer will always query the fastest available server, and in the event that one server is unavailable will use the next fastest server available.

DHCP

Centralization

Many organizations now centralize their entire environment, using a Virtual Desktop Infrastructure (VDI) for remote sites and branches. In these cases, the link between the data center and remote sites is critical to normal operations. If that link goes down, the entire remote site will lose access to the network.

This is where placement of DHCP comes into play. If the WAN link goes down at a remote site, will the remote site still function? If the answer is "no", DHCP should not be placed at the remote site. If the answer is "yes", then DHCP can be considered for remote sites.

When centralizing a DHCP environment, fault domains are a key consideration (see below). If a pair of DHCP failover servers become unavailable on the network (for whatever reason), not all DHCP clients are affected.

Distribution

Local DHCP survivability is occasionally required for high-value sites or sites with the need for continuous availability. Administrators have to weigh the additional expense of a distributed DHCP architecture (which requires more units) against the need for high availability.

There are several options for DHCP failover. "Hub-and-spoke" architectures are robust, but also more complex and difficult to maintain. Intra-site architectures may not have the desired level of connectivity at all times, but are simpler to maintain in a failover relationship. Often, this choice depends on DHCP failover behavior.

DHCP Failover Primer

DHCP failover allows for an active/active pairing of DHCP servers. Each DHCP failover peer will assume approximately 50% of the free leases. Should one DHCP server hand out more leases than its peer, a rebalancing event will occur for that specific pool. The 50/50 allocation of free leases provides each peer with the same number of addresses to assign in the event of a break in communications.

There are three states within DHCP failover: **normal, communications-interrupted, and partner-down.**

When both peers are in normal state, DHCP behaves as normal – leases are handed out and rebalancing happens freely and quickly.

When there is a communication problem, both peers move in to a communications-interrupted state. In this state, both peers can renew all addresses but only assign their own free leases. When a lease expires (not renewed) it is then owned by the primary DHCP failover peer. This is important because the unit cannot run in communications-interrupted mode indefinitely. When considering a hub-and-spoke architecture, each spoke should be the primary DHCP failover peer. This ensures that leases will be available locally when they expire.

To allow a DHCP failover peer to handle free leases from a peer, the surviving server must manually be set into partner-down. This transition is not automatic because a human needs to determine which peer is offline. If both peers are online, do not force partner-down on both nodes as this will allow for duplicate IP addresses.

DHCP failover options

- **Local Centralized DHCP failover (1:1)** – two DHCP servers in the same data center
- **Remote Centralized DHCP failover (1:1)** – two DHCP servers in separate data centers
- **Hub-and-spoke (N:1)** – spokes connect back in to a central egress point

  Note: While the number of spokes assigned to a hub are technically unlimited, BlueCat recommends no more than a 5:1 ratio. This makes troubleshooting and upgrades far easier down the line.

  Pool rebalancing should be considered for deployment of hub-and-spoke architectures. Since the hub will be doing re-balancing activities for all spokes, it usually requires a unit with greater capacity.

- **Intra-site (1:1)** – distributed DHCP servers which are paired together

  This is the best option for singular instances at each site. BlueCat recommends meshed MPLS or another form of solid network connectivity between sites in this type of architecture.

- **Mesh (N:N)** – BlueCat does not recommend this architecture, which would involve multiple hubs and spokes attached to one another. This scenario often results in cascading failures, and is extremely difficult to troubleshoot. This sort of DHCP architecture is also very complicated to maintain and deploy.
- **Chain (1:1)** – BlueCat does not recommend this architecture, which uses a chain of failover peers, such as: A to B, B to C, C to D, D to A. Like the mesh scenario above, his architecture is difficult to maintain, deploy, and troubleshoot.

Network latency is important for DHCP failover when performing initial sync or non-standard pool rebalances. While this work is rare, it can have a significant impact on the network (i.e.: the initial sync will move into a non-serving state by default). As such, BlueCat recommends the following guidelines:

- < 25K total IP addresses shared, 200ms (RTT) is advised.
- 25K – 100K total IP addresses shared, 100ms (RTT) is advised.
- 100K+ total IP addresses shared, 50ms (RTT) is strongly advised

Note: These numbers are guidelines, and various DHCP settings can be tweaked to allow for a faster re-sync. See below for further information.

When implementing DHCP failover, the throughput of DHCP will increase to that of xHA. While xHA is active-passive, DHCP failover is active-active. While the DHCP throughput will not be doubled due to rebalancing considerations, a good rule of thumb is 75% of double the LPS (i.e.: (200 LPS * 2) * .75 for a DHCP failover relationship).

Fault Domains

Looking at the raw numbers of queries and clients, even the largest enterprise architectures can be handled by a single DHCP server. That being said, the likelihood of a network outage or hardware failure requires every administrator to consider fault domains.

There are several ways to organize fault domains. The particular architecture and operational contours of the network will dictate which one an administrator uses. For example, fault domains might be organized by:

- Services – data, voice and wireless
- Region
- Business unit

BlueCat recommends a maximum of 50,000 DHCP clients per DHCP failover relationship. This is guided by the simple question: would the organization accept a service interruption (due to say hardware failure or network interruption) for more than 50,000 clients?

Should DHCP be on the same unit as DNS?

While technically possible, capacity issues limit the ability to couple DHCP and DNS on the same unit. When the number of processes running on a single unit exceeds the recommended limit, the units cannot perform within the advertised performance parameters for queries per second and leases per second.

Additional Considerations

Sizing – How Much, and Where?

BlueCat strongly believes that DNS architectures should be flexible and scalable. With this in mind, BlueCat's pricing model is built to avoid the sticker shock of constant hardware upgrades and the energy associated with constant re-architecting of the network. BlueCat offers a unique set of options to back this up, including subscription pricing and the ability to deploy interchangeable physical and virtual appliances.

When considering how many units to deploy, organizations must first decide to use physical or virtual devices. Often, business policies will require the use of physical hardware and determine its characteristics (dual power, dual disk, DC power, etc.). VMs don't have these limitations.

The next question is expected growth, both organic and non-organic. It's not enough to simply buy for today's network, particularly in today's environment where the demands of cloud, automation, SD-WAN, and other strategic initiatives are likely to have a significant impact on DNS capacity. DNS capacity issues tend to creep up gradually and catch organizations which have not planned for realistic growth.

With this in mind, BlueCat recommends a five year horizon for capacity planning. In our experience, this takes into account the lifecycle of most hardware as well as the natural churn of IT initiatives. Planning further than five years out usually fails to take new innovations into account. Planning for fewer than five years usually results in frequent refresh cycles.

BlueCat plans (and prices) its DNS architectures based on **unique active IP addresses**. We define this by counting specific IP addresses which issue at least one query every month. This helps to determine actual network load and demand, which naturally lays the groundwork for a DNS architecture to meet this capacity.

BlueCat measures unique active IP addresses through service points placed on the recursive layers. These measure traffic patterns against specific IP addresses over time as they log DNS data coming off of client devices.

Some additional considerations to keep in mind when determining capacity:

- The internal recursive layer will see most of the traffic, but isn't busy doing a whole lot other than caching
- Although very rare, memory limitations should be considered (i.e.: leases in memory, DNS files)
- Traffic spikes due to peak usage are often a key consideration

For DHCP, network administrators should consider resilience SLAs in the case of a natural disaster or other outage scenario. By default, Windows performs a DHCP request at 1, 2, 4, 8, 16, 32, and 300 seconds. It then runs through the same pattern until a response is found. With a 200 LPS capability, even the largest networks can come online with minutes.

For LPS, lease times should be also considered. Here are some typical lease times for DHCP:

Peak usage (i.e.: 9am Monday morning) should be considered to limit capacity-related issues. Disaster scenarios also need to be considered – should a data center become unavailable for a prolonged period of time, can the surviving unit handle the entire load?

- **Wireless:** 1-4 hours, although a day is not uncommon
- **Data:** 3-7 days
- **Voice**: 7-14 days

Non-local xHA

It is almost never acceptable to have xHA across two sites. If the link between the two sites is brought down, both xHA peers will move in to an active state, leading to a “split-brain” scenario which interferes with the centralization of the BlueCat solution. Preventing this scenario is extremely important with a DNS master (as both servers could accept DDNS updates) or DHCP master (duplicate IPs will be handed out). When the link between two sites is restored and re-integration is achieved, there will likely be some data loss.

The only acceptable way to deploy non-local xHA is with Layer-2 connectivity between the sites and dedicated dark fiber lines connecting them. This makes the two sites effectively the same data center.

Anycast

Anycast is fantastic for redundancy, resiliency and scalability. When designing a large architecture with anycast that has been extended out to remote sites, anycast should not be advertised (no OSPF or BGP advertisements) to the rest of the network. This way, the local networks will be able to resolve against the anycast IP, but none of the other sites will be hitting the slower links at remote sites.

GSS-TSIG

GSS-TSIG allows Kerberos authentication when DDNS updates are made. While secure, GSS-TSIG slows down the DDNS update process significantly because Kerberos needs to be contacted for every DDNS update. DDNS updates will slow from approximately 1000 per second with regular TSIG down to about 10-20 with GSS-TSIG.

DNS Views = Containers

DNS views provide different DNS answers to clients based on the source IP address (match-clients DNS option) of the request. This being the case, no two DNS views should be set with the "any" match-clients setting in the same configuration.

The most common deployment is to have the external servers match only external view deployment roles and internal servers match only internal view deployment roles – much like a folder.

If an administrator makes a mistake and cross-contaminates the views, either the internal servers or external servers will become completely unreachable because the DNS view list in named.conf will be the same on both servers. DNS queries will match the first DNS view it can. With cross contamination, internal and external will servers will both match either the internal or external view (depending on which is ordered first).

Disaster Recovery

Is your network's disaster recovery (DR) plan up to date? If the DR site is cold, this can dramatically impact DNS operations in the event of an incident. How will configurations be applied to this cold DR site? What if the DR site is warm? Is the DR site IP-for-IP?

There are multiple DR scenarios – so many that we can't go through all of them here. Needless to say, DR should be a major consideration in any DNS architecture.

Tap into our expertise

For over twenty years, BlueCat has managed some of the largest and most complex DNS architectures around. From financial institutions to government agencies to retail giants, we've seen just about every quirky workaround and use case known to the world of DNS.

As noted in the beginning of this eBook, no two networks are alike. Sometimes there are reasons to break the rules, and sometimes the rules are there to save you from yourself. This eBook contains some of the major considerations which are common across all networks, but your individual use cases and architectures will likely require additional discussion.

BlueCat is always happy to discuss the options for DNS, DHCP, and IPAM architectures and run through what makes sense for your particular environment. We can help you no matter what stage of the deployment process you're in.

CONTACT US

BLUECAT™