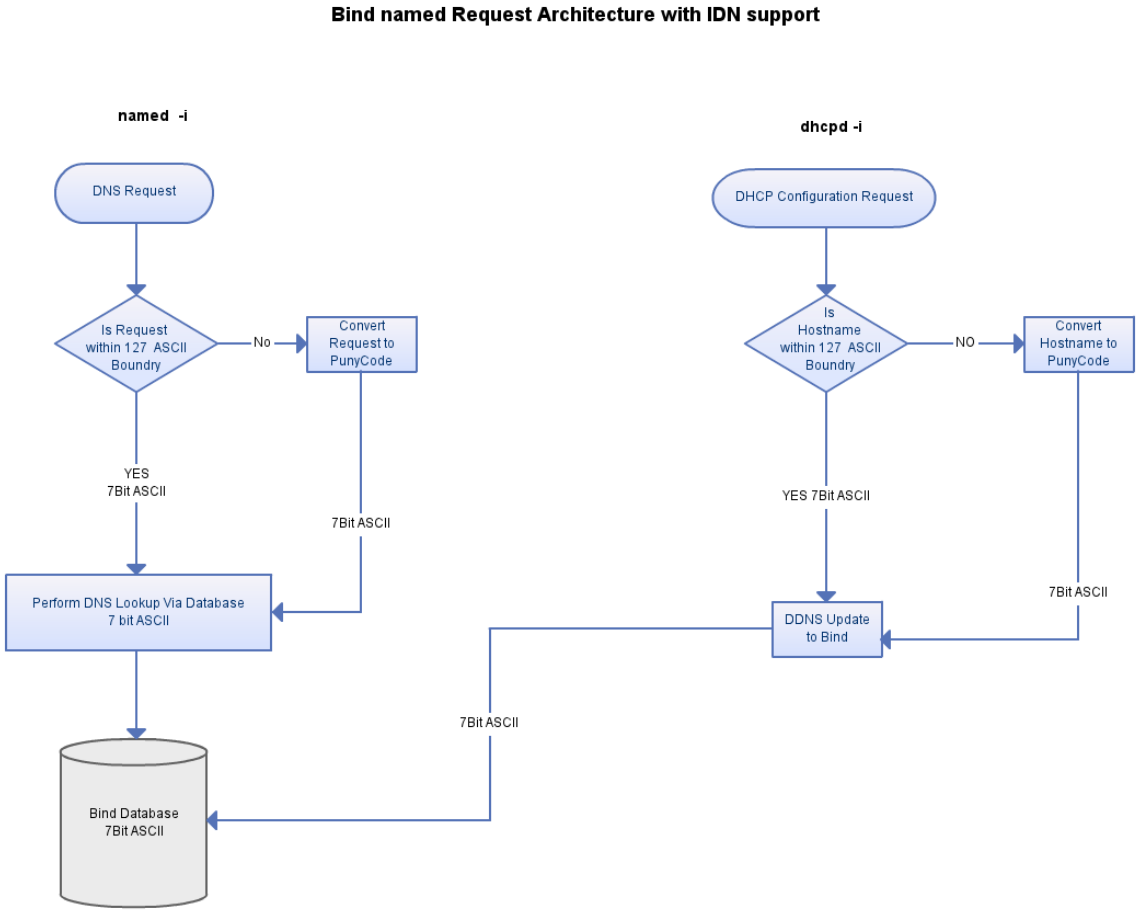
PunyCode Support for BIND/DHCP



Approach

We cannot count on DNS clients and DNS system resolvers to properly convert to PunyCode or to something allowing us to consistently resolve hostnames.. therefore we must move the logic to the logic into the servers.

Add a switch the the named daemon and the dhcpd daemon to allow them to pre-process incoming requests to look for requests that do not conform to the 7Bit ASCII required for BIND. If this is detected, convert the request into PunyCode. This approach would adopt a filter pattern, similar to servlet filters in Java and would not change any of the code used for named lookup or dhcpd DDNS. The proposed switch would be -i for IDN.

Advantages

- Does not change the behavior of named and dhcpd if the switch is not enabled
- Gives the user / customer a choice of behaviors
- Translation is done when the DNS request occurs so we don't have to rely on client to do the PunyCode translation
- No need to change database or storage model for named and dhcpd
- Since requests & hostnames are translated to PunyCode before being stored, zone transfers, caching name servers and all other infrastructure will work as expected.
- Should be low cost from a coding standpoint, just need to build an "adapter/filter" on the incoming requests for update or DNS resolve.

Disadvantages

- DDNS hostnames may be in encoding other than UTF-8, may not be able to detect.
- Examining every incoming request and converting to PunyCode may become a performance bottleneck.
- Servers with this feature turned on will behave differently.

This feature is mainly for customers who will be using BIND infrastructure internally and may not be an ideal solution for internet facing DNS servers.

This could also be a compile time switch.