Proxy Chaining & URL Rewriting

Use Case:

Perform proxy chaining operation while rewriting the URL or a part of it. Such use cases come often where based on incoming request content you need to make a proxy selection decision. Before you forward the request to backend, content rewrite is needed as well.

F5 iRules:

```
when HTTP_PROXY_REQUEST {
    log local0. "[HTTP::method] [HTTP::uri]"
    switch [string tolower [URI::host [HTTP::uri]]] {
        "www.google.com" {
            # send request to default pool (aka proxy-chaining)
            HTTP::proxy disable
         }
         "www.abc.com" {
            # change request to a different host - remains a proxy request
            HTTP::uri http://www.google.com/
         }
         "www.def.com" {
            # change request to a normal (not proxy) request - goes to the
default pool
            HTTP::uri /def.html
          }
     }
}

when HTTP_REQUEST {
    log local0. "[HTTP::method] [HTTP::uri]"
}
```

NetScaler Solution:

```
add cs action prox_act1 -targetlbVserver default_lb
add cs policy prox_pol1 -rule
HTTP.REQ.HOSTNAME.SET_TEXT_MODE(IGNORECASE).EQ("www.google.com") -action
prox_act1
bind cs vserver cs_vserver_1 -policyName prox_pol -priority 1

add rewrite action re_act1 replace HTTP.REQ.URL "\"www.google.com\""
add rewrite policy re_pol1
HTTP.REQ.HOSTNAME.SET_TEXT_MODE(IGNORECASE).EQ("www.abc.com") re_act1
bind cs vserver cs_vserver_1 -policyname re_pol1 -priority 2 -type request

add rewrite action re_act2 replace HTTP.REQ.URL '"/def.html"'
add rewrite policy re_pol2 HTTP.REQ.HOSTNAME.SET_TEXT_MODE(IGNORECASE).EQ
("www.def.com") re_act2
bind cs vserver v1 -policyname re_pol2 -priority 3 -type request
```

Here NetScaler is doing multiple operations like making proxy decision at first place then changing the hostname and then part of the URL. In similar manner NetScaler can handle any level of complication while changing the content across.