Vodafone intends to support the HTTP/1.1 specification as outlined in RFC2616 (the current proposed standard) and expects that all handsets implement this specification.

1.1 Last-Modified and Etag

1. When the server sends a response with a *Last-Modified* header, the client **MUST** store the value as meta-data together with the object.
2. When the server sends a response with a *Etag* header, the client **MUST** store the value as meta-data together with the object.
3. If the object is not saved in cache (see caching section below), this doesn't apply because there's no object to store together with the meta-data.
4. When the client requests an object and a copy of the object exists in cache the client **MUST** send an *If-Modified-Since* header to the server containing the meta-data previously received in the *Last-Modified* server header.
5. When the client requests an object and a copy of the object exists in cache, the client **MUST** send an *If-None-Match* header to the server containing the meta-data previously received in the *ETag* server header.
6. The client **MUST** process *Last-Modified* and *ETag* headers at least when receiving server responses of *200* and *304* (other status codes are optional). The client **MUST** also receive and process *Last-Modified* and *ETag* as specified in RFC2616.

1.2 304 Not Modified

1. When the server replies with a *304 Not Modifed* status code, the client **MUST** use the object in the cache.
2. When the server replies with a *200 OK*, the client **MUST** store the new object in cache according to the caching rules in the next section.

1.3 Caching headers

1. The caching instructions received via HTTP headers **MUST** take precedence over the caching instructions received present in *meta* elements in the markup. (This requirement is also covered in TCD-BWSR-RET-003389)
2. *Cache-Control* headers **MUST** take precedence over the *Expire* header. If both exist, the User-Agent **MUST** ignore the *Expire* header and only process the *Cache-Control*. (This requirement is also covered in TCD-BWSR-RET-003389)
3. If the server sends a response with *Cache-Control: no-cache* or *Cache-Control: no-store*, the User-Agent **MUST NOT** store the object in cache (if the implementation needs to use the cache for internal purposes, the object **MUST** be deleted or marked as invalid afterwards and **MUST NOT** affect the memory space required for cached objects).
4. If the server sends a response with *Cache-Control: max-age=<max-age_seconds>*, the object **MUST** be stored on cache and its expire meta-data **MUST** be set to the User-Agent's date plus the value of the header, in seconds. (*user-agent_date* + *max-age_seconds*).
5. If the server sends a response with an *Expires: <expire_date>*, and a *Cache-Control*, the *Expires* **MUST** be ignored. (This requirement is also covered in TCD-BWSR-RET-003389)
6. If the server sends a response with an *Expires: <expire_date>*, no *Cache-Control*, and a *Date: <server_date>* header, the User-Agent **MUST** store the object on cache and **MUST** set its expire meta-data to *expire-data – server_date* + *user-agent_date*.

 © Vodafone 2004 – Confidential

7. If the server sends a response with an *Expires: <expire_date>*, no *Cache-Control*, and no *Date: <server_date>* header, the User-Agent **MUST** store the object on cache and **MUST** set its meta-data to *expire-data*.
8. The expire meta-data **MUST NOT** wrap around and **MUST** be rounded to the minimum or maximum value. For example, *unixtime=0* (1970.01.01 00:00:00) minus one second **MUST** be rounded to *unixtime=0* and never to *unixtime=(2^32-1)-1* (sometime around year 2038). The same rounding **MUST** happens in the upper limit.
9. The terminal MUST cache content even if none of the validity headers ('Cache-Control: max-age' or 'Expire') are defined in the same way as if Cache-Control: max-age=0 or Expires: with a date equivalent to the Server Date were present.

1.4 Caching Behavior

1. If the object is not present in cache, the User-Agent **MUST** request the object and process the response headers as defined in this document.
2. If the object is present in cache and expire meta-data is in the past (or *now*), the User-Agent **MUST** do a network request and send any *Last-Modified* and *Etag* headers, as defined above.
3. If the object is present in cache and expire meta-data is in the future, the User-Agent **MUST NOT** do a network request and **MUST** use the object and relevant meta-data from cache.

1.5 Link or referenced objects Navigation

1. All previous rules **MUST** be applied when the user clicks on a link, or an object is referenced from the markup (ex: images), either directly or via any kind of scripting.

1.6 History Navigation

1. All previous rules **MUST** be applied when the user navigates back in the history, either via *Back* key or by choosing a previous page in the *History* menu, if available, with this exception: The User-Agent **MUST** get the object from cache, if available, even if it has already expired.
2. The User-Agent **MAY** still fetch the object from the network as usual when the object is not available on cache, but the user MUST be notified of the network being accessed.

1.7 Reload Navigation

1. All previous rules **MUST** be applied when the user selects the *Reload* menu key, with this exception: The User-Agent **MUST** revalidate the object from the network even when the expire meta-data is in the future.
2. The User-Agent **MUST** still send all *Last-Modified* and *ETag* headers as usual.

1.8 Cache-Control: no-store

1. The content **MUST NOT** be stored in cache thus the content must be reloaded on every access, similar to the *Cache-Control: no-cache* behaviour.
2. The content **SHOULD NOT** be saveable by the user (e.g. images can't be stored to the phone memory).

1.9 Vodafone Specific Storage

1. The User-Agent **MUST** implement the 'Vodafone Specific Storage' which is the exclusive cache for Vodafone. The terminal **MUST** only keep content identified as Vodafone content in VSS memory.

2. The User-Agent **MUST** determine the cache location for content according to a Vodafone proprietary header and a Domain white list. If the header value matches ‘Portal’ and the requested URI is listed in the white list, the terminal **MUST** cache the contents as Vodafone Specific Storage.
3. Vodafone proprietary header

```
message-header    = field-name ":" [ field-value ]
field-name        = X-Vodafone-Content
field-value       = "Portal"
```

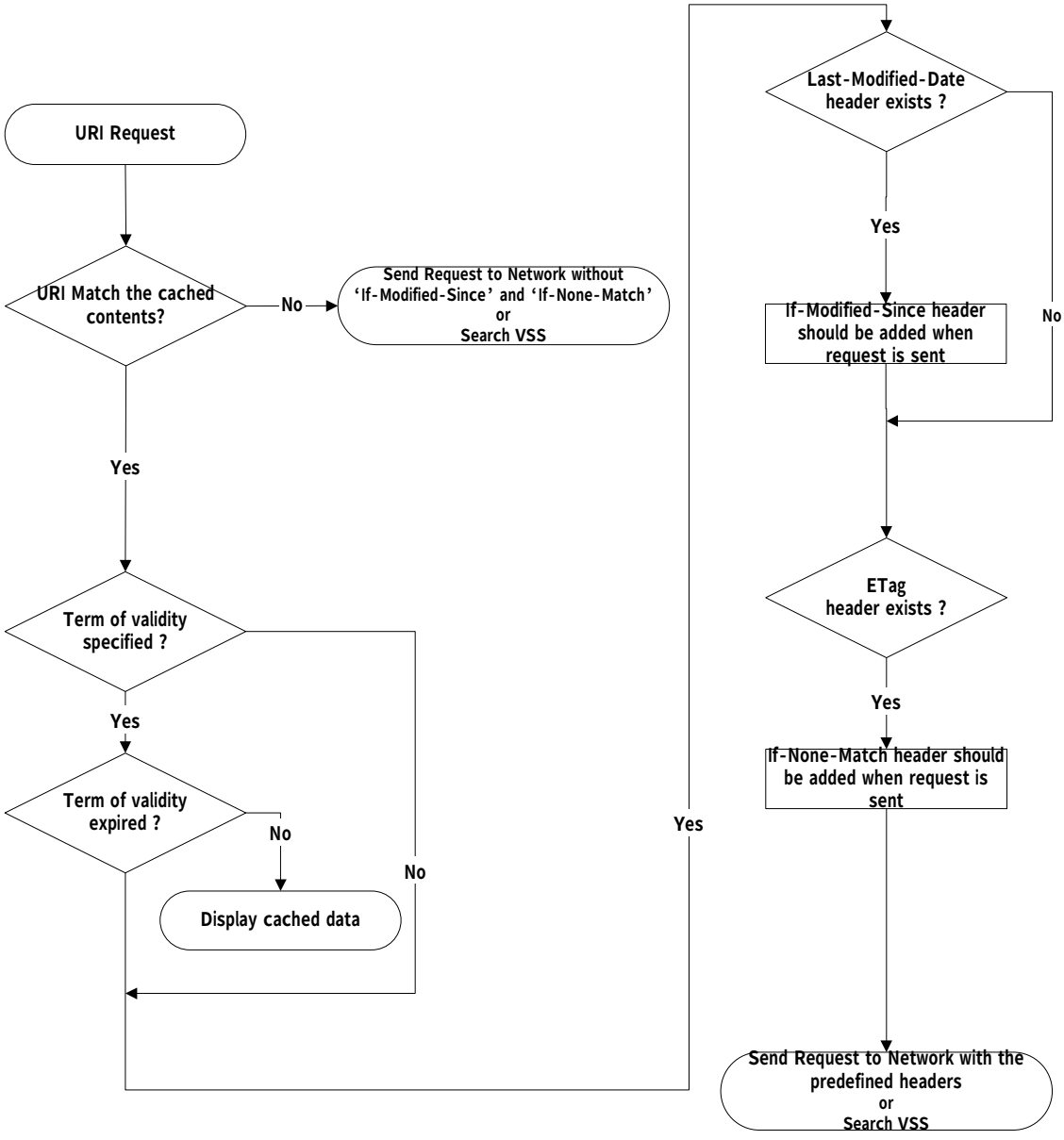
4. White List Domain

   The domain white list will be provided during implementation phase.

 © Vodafone 2004 – Confidential

1.10 Cache Search

The terminal **MUST** search the contents in cache memory before any network access. The device **MUST** search for a file in VSS and then in normal cache first. If the VSS content is expired, the terminal should search in normal cache. The device **MUST** search the content in normal cache or VSS according to following flow.



 © Vodafone 2004 – Confidential

1.11 URI Matching

1. The terminal **MUST** identify the contents in cache memory by URI. When the browser receives a MIME-multipart or WSP-multipart mixed response, the browser must cache each object within the multipart using the information based in the header associated with each object. Vodafone will use the Content-Location header to identify the URI for each object.

 © Vodafone 2004 – Confidential