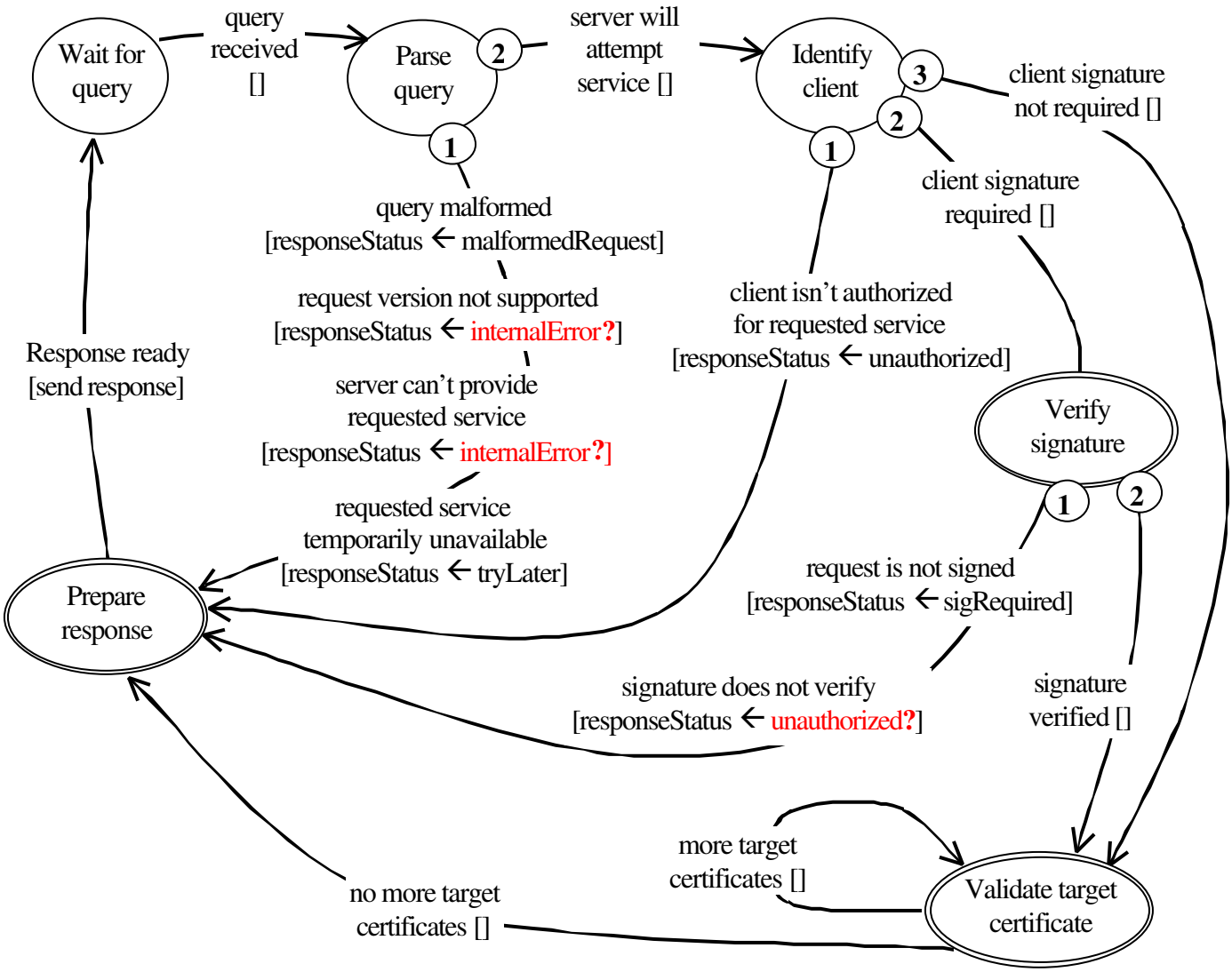
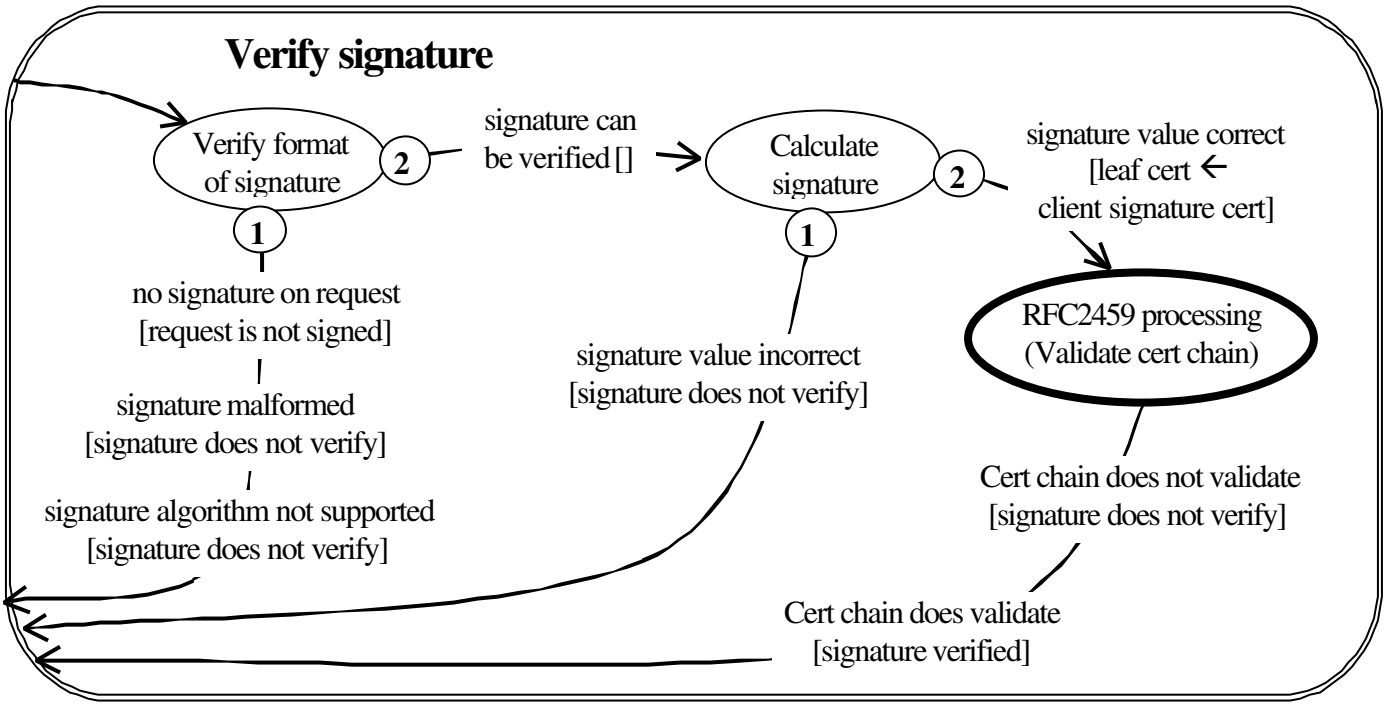
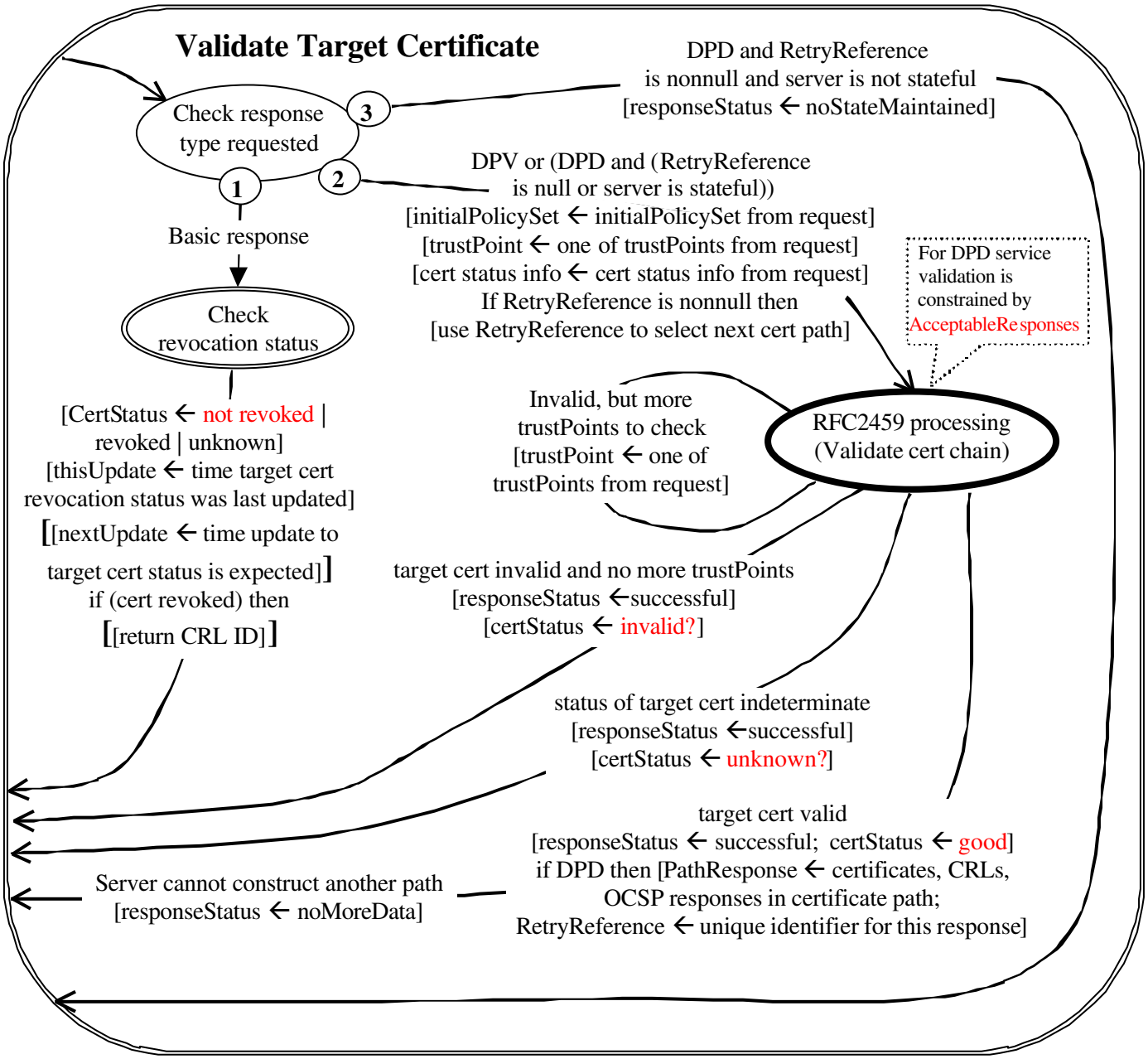
State Machine for OCSP v2 Server with Optional DPV and DPD Service

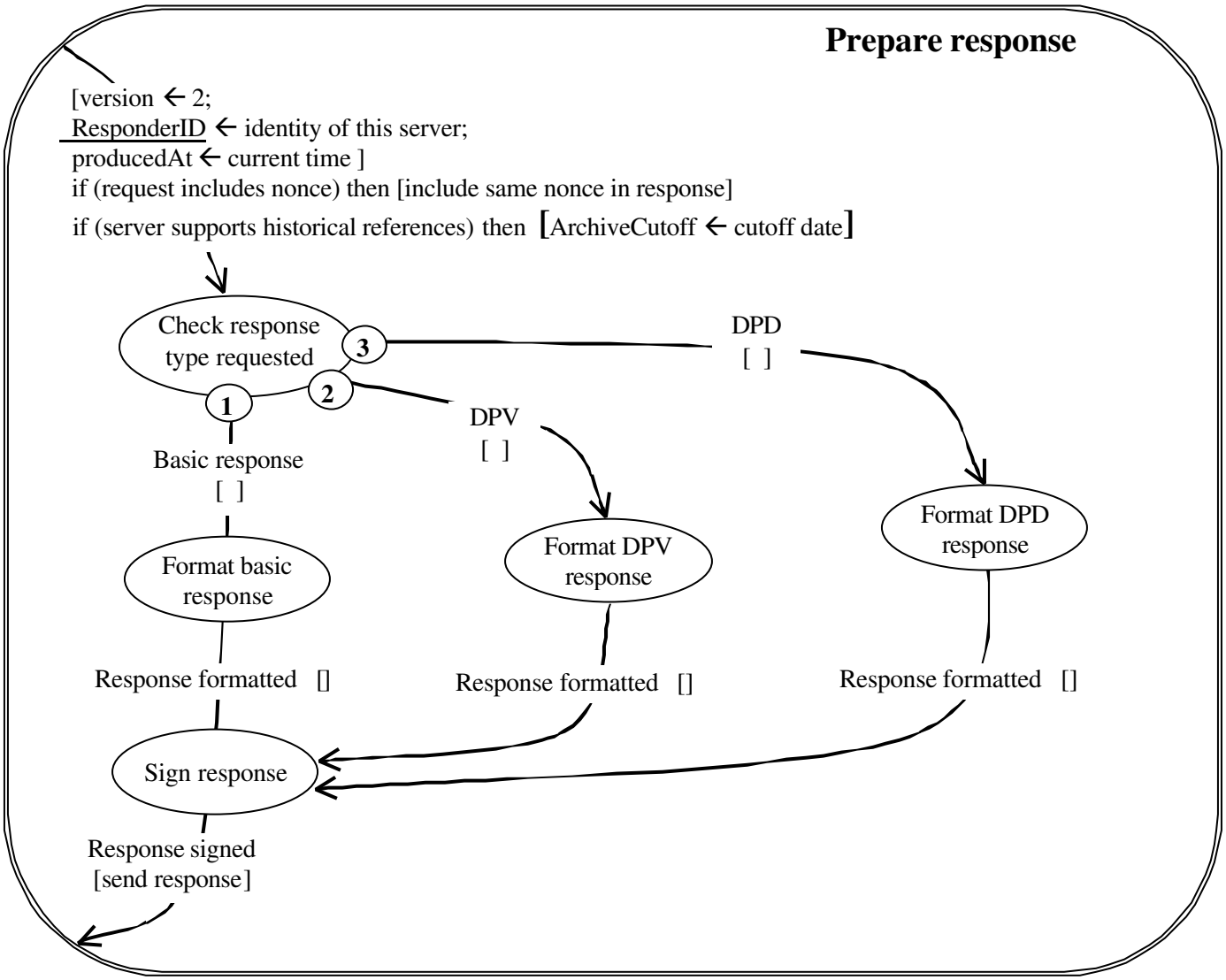
State diagram syntax:

States are identified by names within ellipses. Ellipses with double lines indicate that internal transitions within the state are defined by a subsequent state transition diagram. Ellipses with bold lines indicate that internal details are defined in some other document (e.g. RFC2459). Transitions between states are indicated by arrows. Each transition is conditioned by one or more events, followed by one or more square brackets enclosing an output or variable assignment that occurs when that event has the highest priority. The event conditions closest to the tail of the arrow have highest priority. When there are multiple exit transitions from a state, the transitions are evaluated in ascending order according to the numbers in the small circles. Outputs or variable assignments enclosed in large bold square brackets are optional.




Verify signature
Verify format of signature
2
1
signature can be verified []
Calculate signature
2
1
signature value correct [leaf cert ← client signature cert]
RFC2459 processing (Validate cert chain)
no signature on request [request is not signed]
signature malformed [signature does not verify]
signature algorithm not supported [signature does not verify]
signature value incorrect [signature does not verify]
Cert chain does not validate [signature does not verify]
Cert chain does validate [signature verified]


Validate Target Certificate
Check response type requested
1
2
3
Basic response
Check revocation status
[CertStatus ← not revoked | revoked | unknown]
[thisUpdate ← time target cert revocation status was last updated]
[[nextUpdate ← time update to target cert status is expected]]
if (cert revoked) then
[[return CRL ID]]
DPD and RetryReference is nonnull and server is not stateful
[responseStatus ← noStateMaintained]
DPV or (DPD and (RetryReference is null or server is stateful))
[initialPolicySet ← initialPolicySet from request]
[trustPoint ← one of trustPoints from request]
[cert status info ← cert status info from request]
If RetryReference is nonnull then
[use RetryReference to select next cert path]
For DPD service validation is constrained by AcceptableResponses
RFC2459 processing (Validate cert chain)
Invalid, but more trustPoints to check
[trustPoint ← one of trustPoints from request]
target cert invalid and no more trustPoints
[responseStatus ←successful]
[certStatus ← invalid?]
status of target cert indeterminate
[responseStatus ←successful]
[certStatus ← unknown?]
target cert valid
[responseStatus ← successful; certStatus ← good]
if DPD then [PathResponse ← certificates, CRLs, OCSP responses in certificate path;
RetryReference ← unique identifier for this response]
Server cannot construct another path
[responseStatus ← noMoreData]


Prepare response
[version ← 2;
ResponderID ← identity of this server;
producedAt ← current time]
if (request includes nonce) then [include same nonce in response]
if (server supports historical references) then [ArchiveCutoff ← cutoff date]
Check response type requested
1
2
3
Basic response [ ]
DPV [ ]
DPD [ ]
Format basic response
Format DPV response
Format DPD response
Response formatted []
Response formatted []
Response formatted []
Sign response
Response signed
[send response]