

```

CleanerGetUnusedKeys(c)  $\triangleq$ 
  LET current  $\triangleq$  cleanerStates[c] IN
     $\wedge$  current.state = "got_blob_keys"
     $\wedge$  cleanerStates' = [
      cleanerStates EXCEPT
        ![c].state = "got_unused_keys",
        ![c].unusedBlobKeys =
          {k  $\in$  current.blobKeys :
             $\forall u \in \text{USERIDS} :$ 
              databaseState[u].imageId  $\neq$  k},
          Mark the time the unused keys were retrieved
        ![c].unusedKeyTime = time
      ]
     $\wedge$  UNCHANGED  $\langle$ serverStates, databaseState, blobStoreState, operations $\rangle$ 
     $\wedge$  UNCHANGED time

CleanerDeletingKeys(c)  $\triangleq$ 
  LET current  $\triangleq$  cleanerStates[c] IN
    Keys get deleted a minimum 1 hour after they are valid
    This gives reads time to die
    LET earliestDeleteTime  $\triangleq$  current.unusedKeyTime + 1 IN
       $\wedge$  time  $\geq$  earliestDeleteTime
       $\wedge$  current.state  $\in$  {"got_unused_keys", "deleting_keys"}
       $\wedge$  Cardinality(current.unusedBlobKeys)  $\neq$  0
       $\wedge$   $\exists k \in$  current.unusedBlobKeys : Pick a key to delete
       $\wedge$  blobStoreState' =
        [blobStoreState EXCEPT
          ![k] = [status  $\mapsto$  "UNSET", image  $\mapsto$  "UNSET"]]
       $\wedge$  cleanerStates' = [
        cleanerStates EXCEPT
          ![c].unusedBlobKeys = current.unusedBlobKeys  $\setminus$  {k}
        ]
       $\wedge$  UNCHANGED  $\langle$ serverStates, databaseState, operations $\rangle$ 
       $\wedge$  UNCHANGED time

```