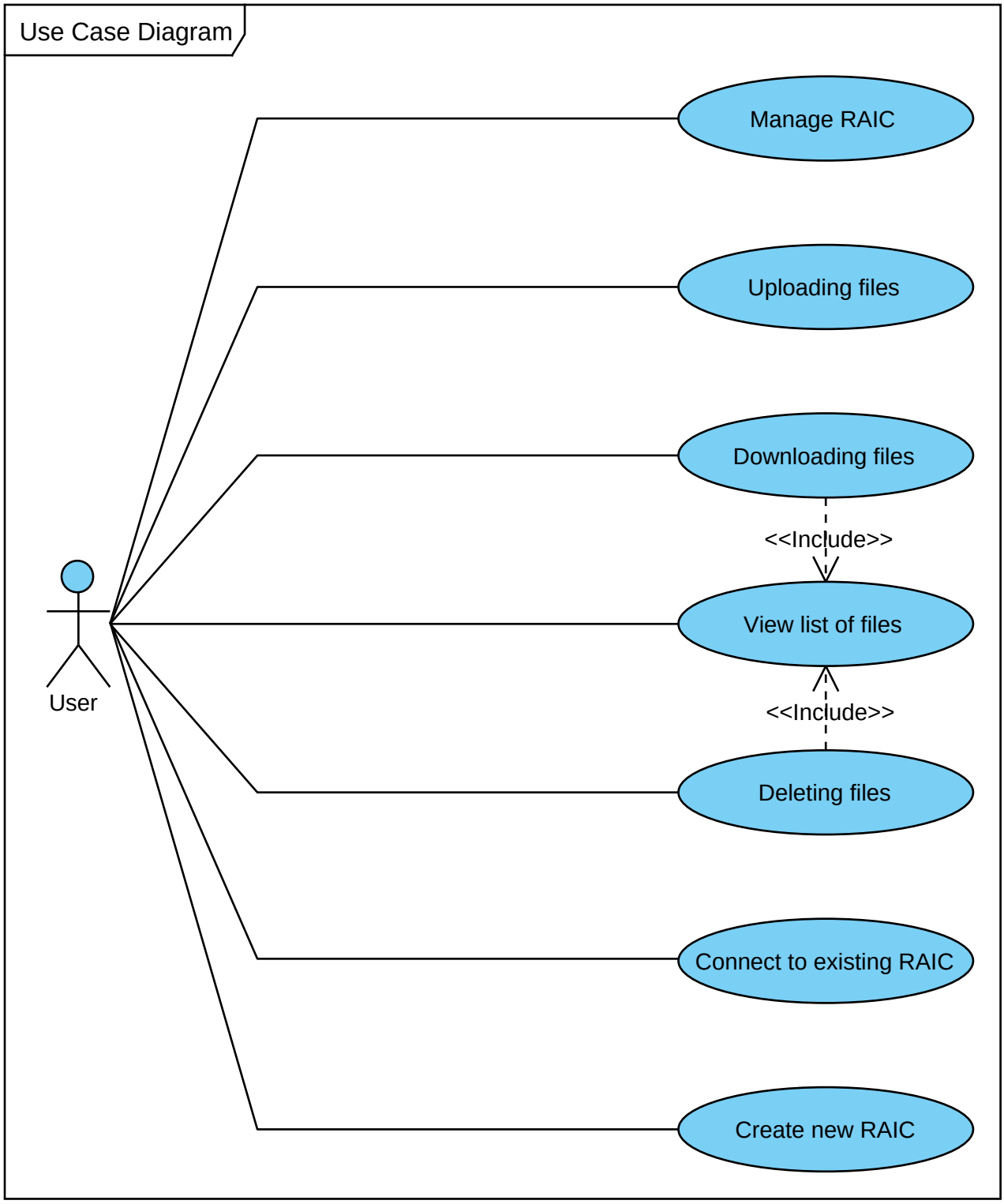
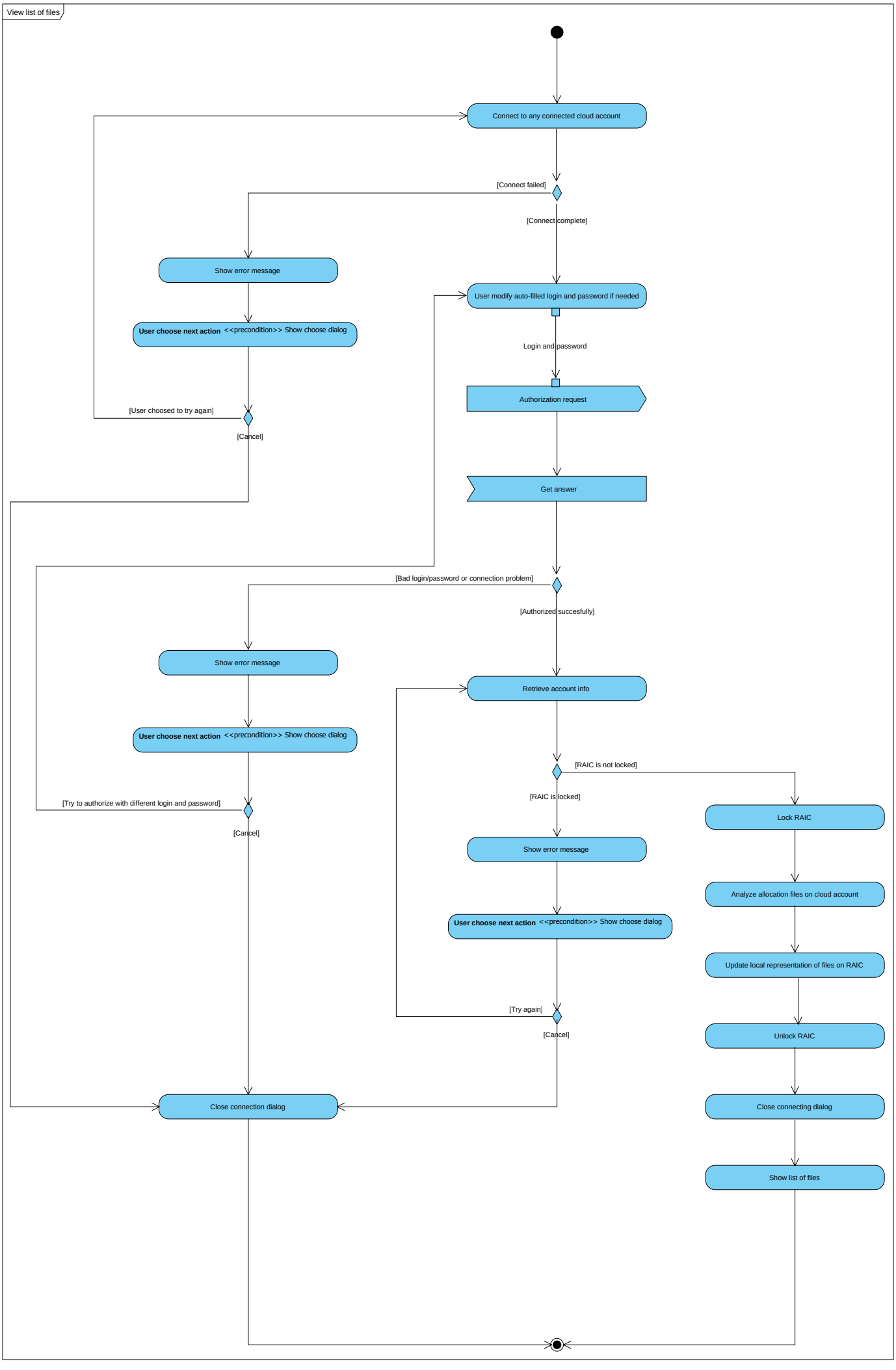
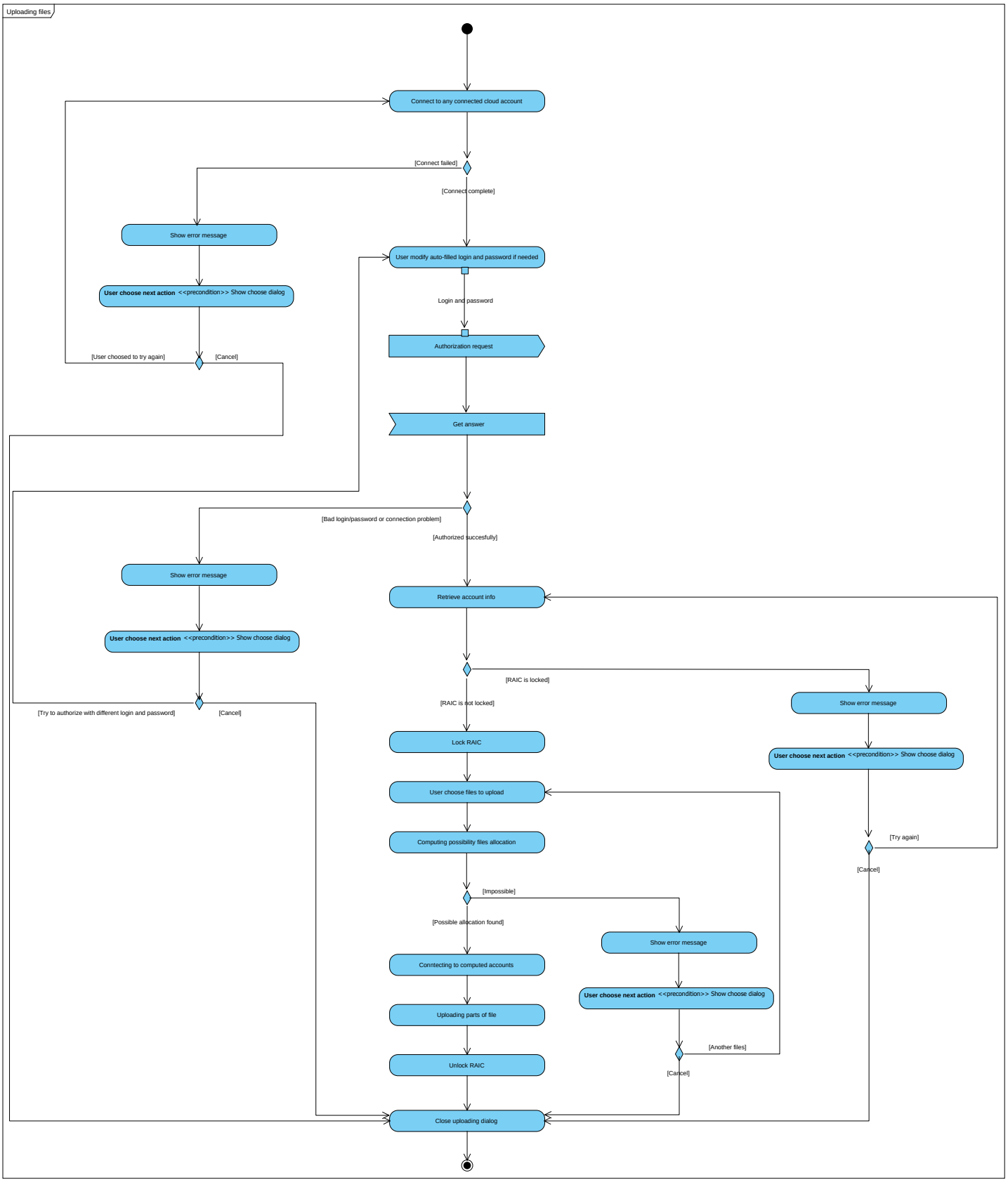
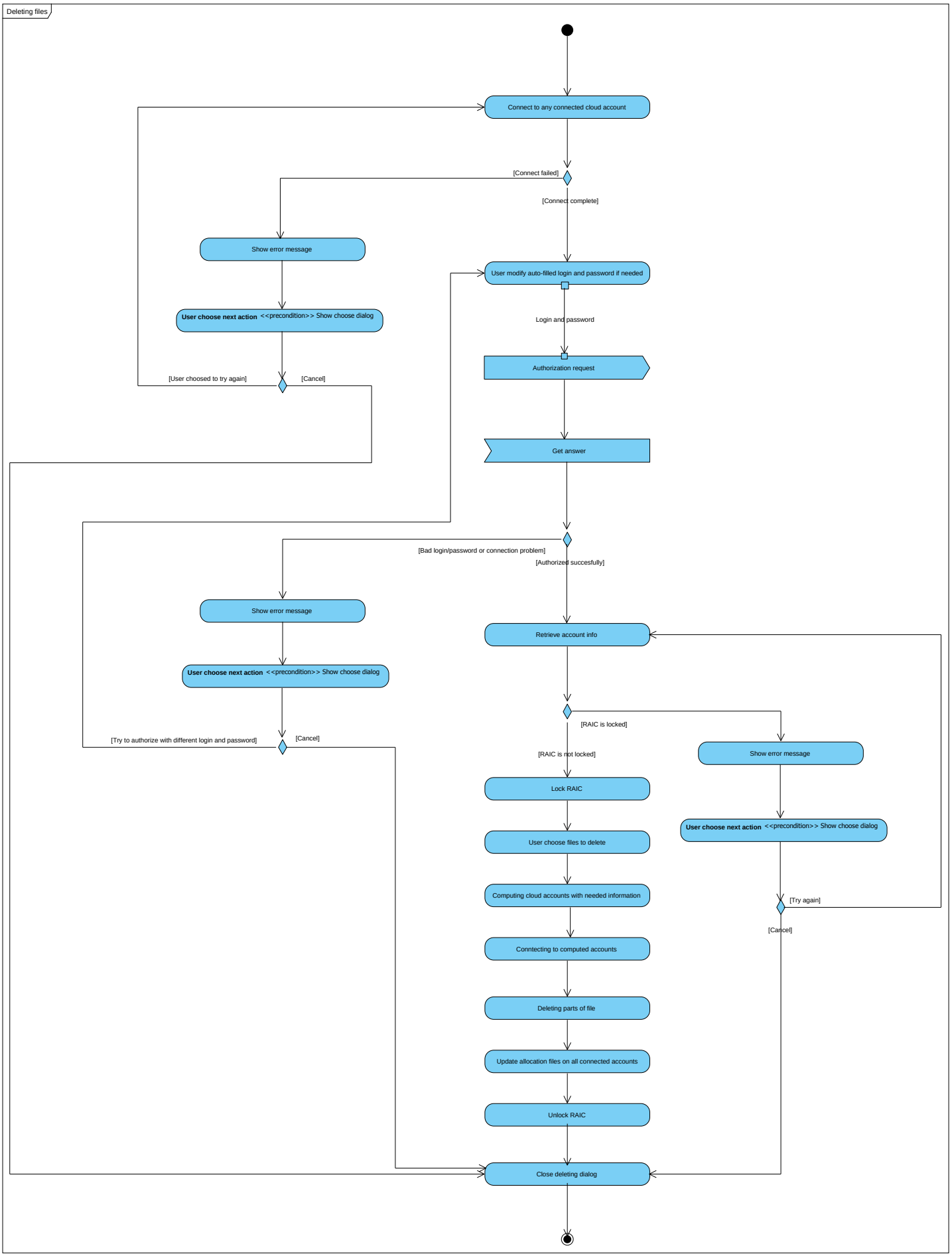
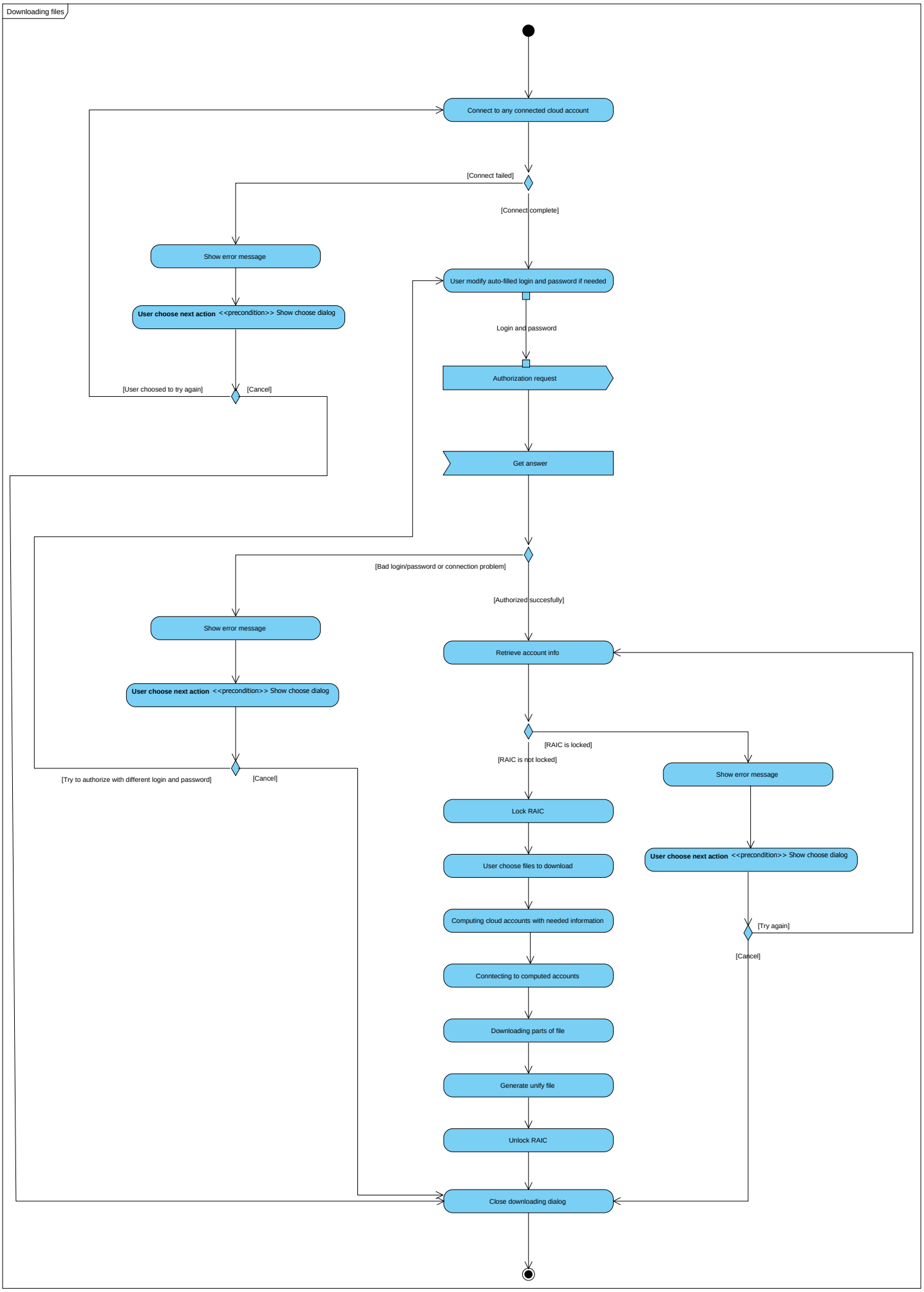
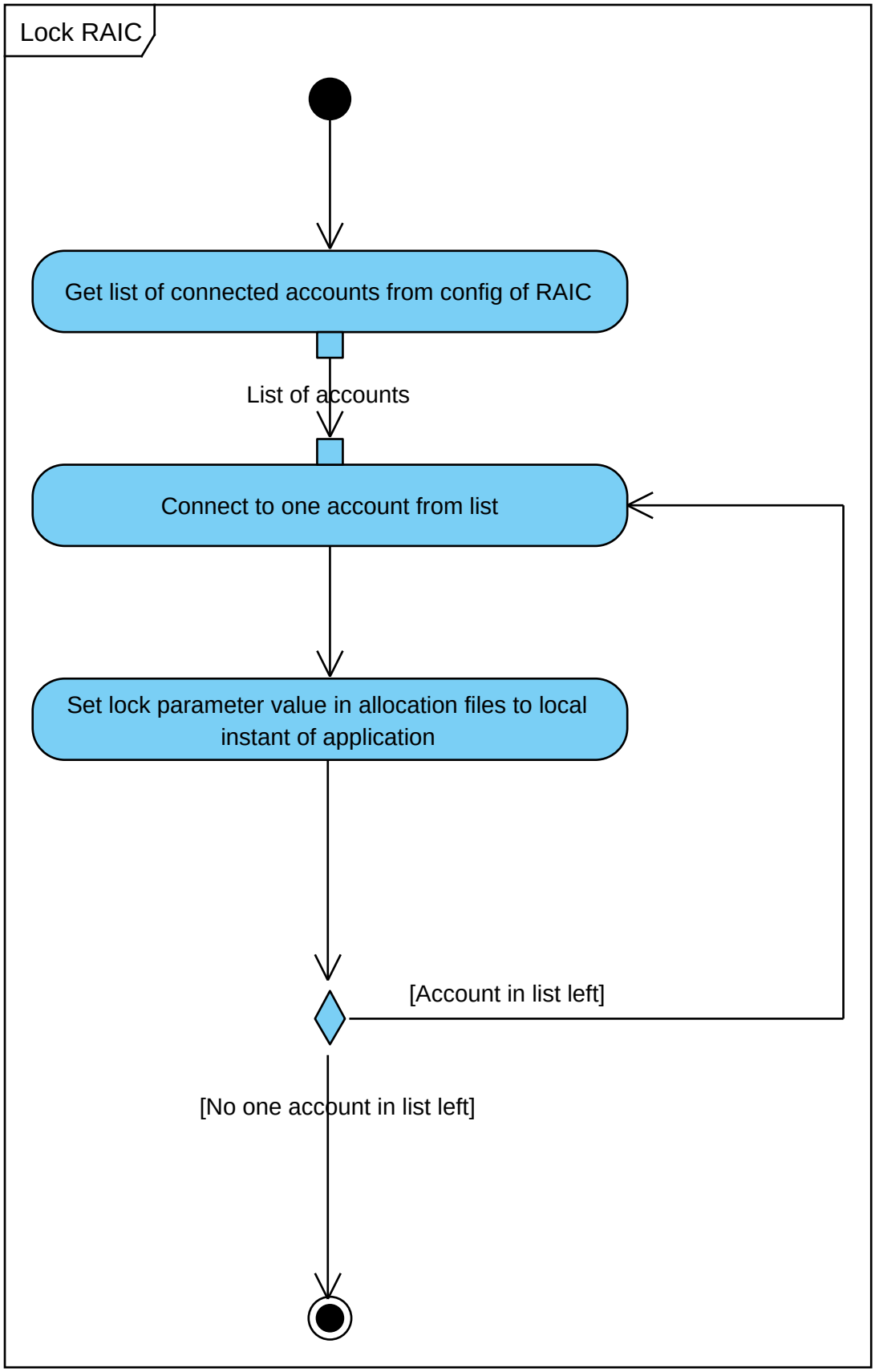

Use Case Diagram
Manage RAIC
Uploading files
Downloading files
<<Include>>
View list of files
<<Include>>
Deleting files
Connect to existing RAIC
Create new RAIC
User

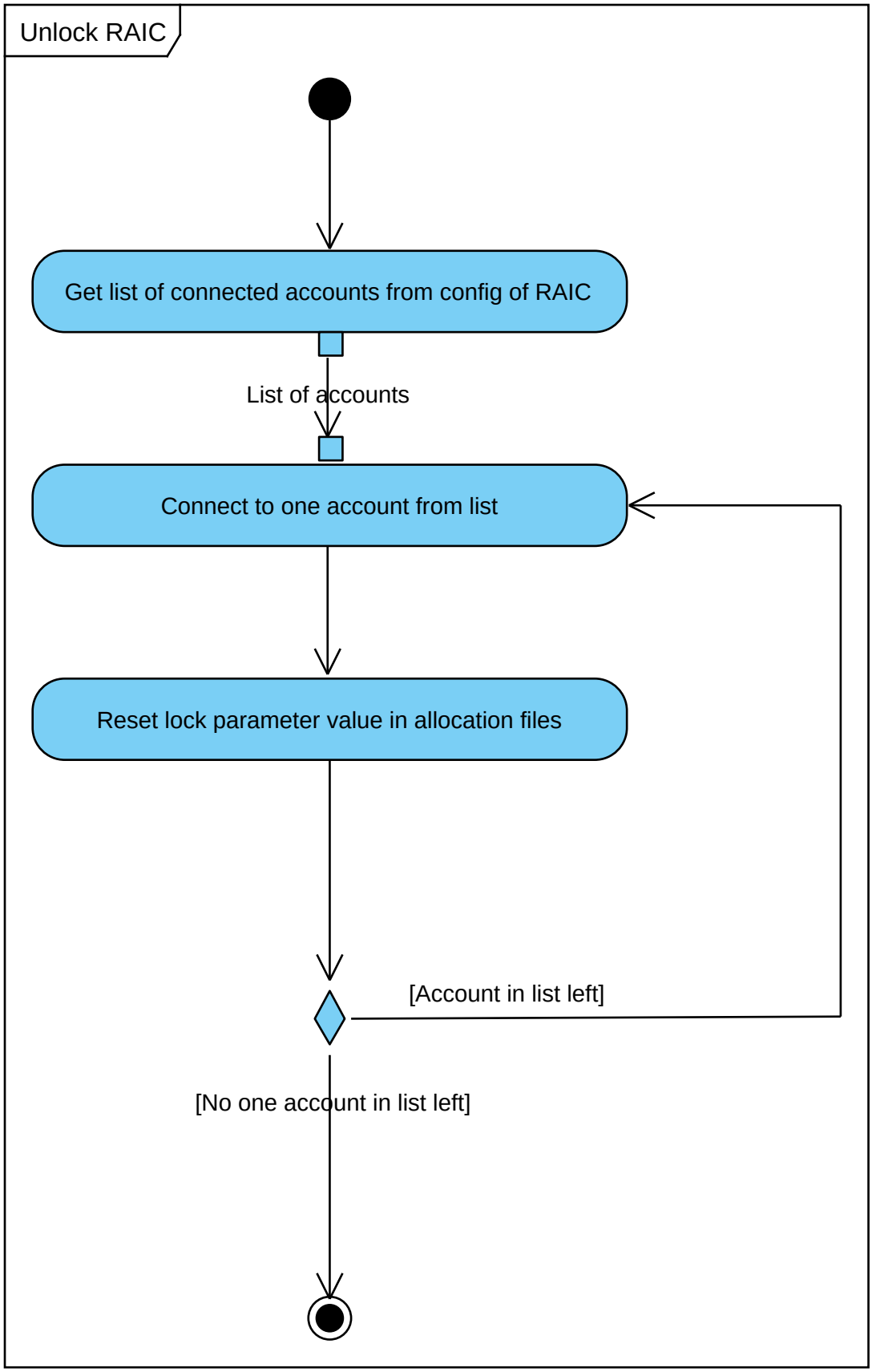

View list of files
Connect to any connected cloud account
[Connect failed]
[Connect complete]
Show error message
User modify auto-filled login and password if needed
User choose next action <<precondition>> Show choose dialog
Login and password
Authorization request
[User choosed to try again]
[Cancel]
Get answer
[Bad login/password or connection problem]
[Authorized succesfully]
Show error message
Retrieve account info
User choose next action <<precondition>> Show choose dialog
[RAIC is not locked]
[Try to authorize with different login and password]
[RAIC is locked]
[Cancel]
Lock RAIC
Show error message
Analyze allocation files on cloud account
User choose next action <<precondition>> Show choose dialog
Update local representation of files on RAIC
[Try again]
[Cancel]
Unlock RAIC
Close connection dialog
Close connecting dialog
Show list of files

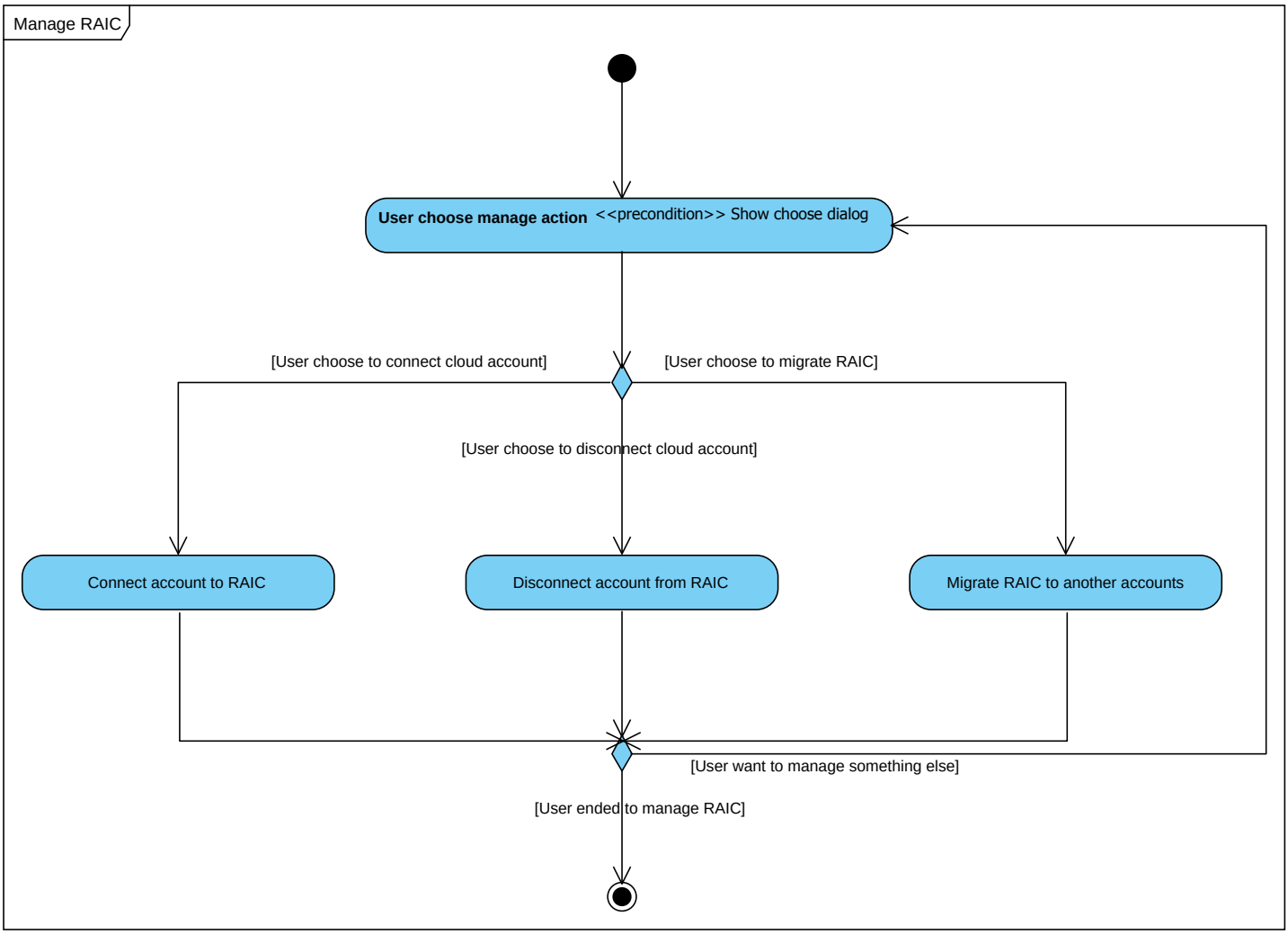

Uploading files
Connect to any connected cloud account
[Connect failed]
[Connect complete]
Show error message
User modify auto-filled login and password if needed
User choose next action <<precondition>> Show choose dialog
Login and password
Authorization request
[User choosed to try again]
[Cancel]
Get answer
[Bad login/password or connection problem]
[Authorized succesfully]
Show error message
Retrieve account info
User choose next action <<precondition>> Show choose dialog
[RAIC is locked]
[Try to authorize with different login and password]
[Cancel]
[RAIC is not locked]
Show error message
Lock RAIC
User choose next action <<precondition>> Show choose dialog
User choose files to upload
Computing possibility files allocation
[Try again]
[Cancel]
[Impossible]
[Possible allocation found]
Show error message
Conntecting to computed accounts
User choose next action <<precondition>> Show choose dialog
Uploading parts of file
[Another files]
Unlock RAIC
[Cancel]
Close uploading dialog

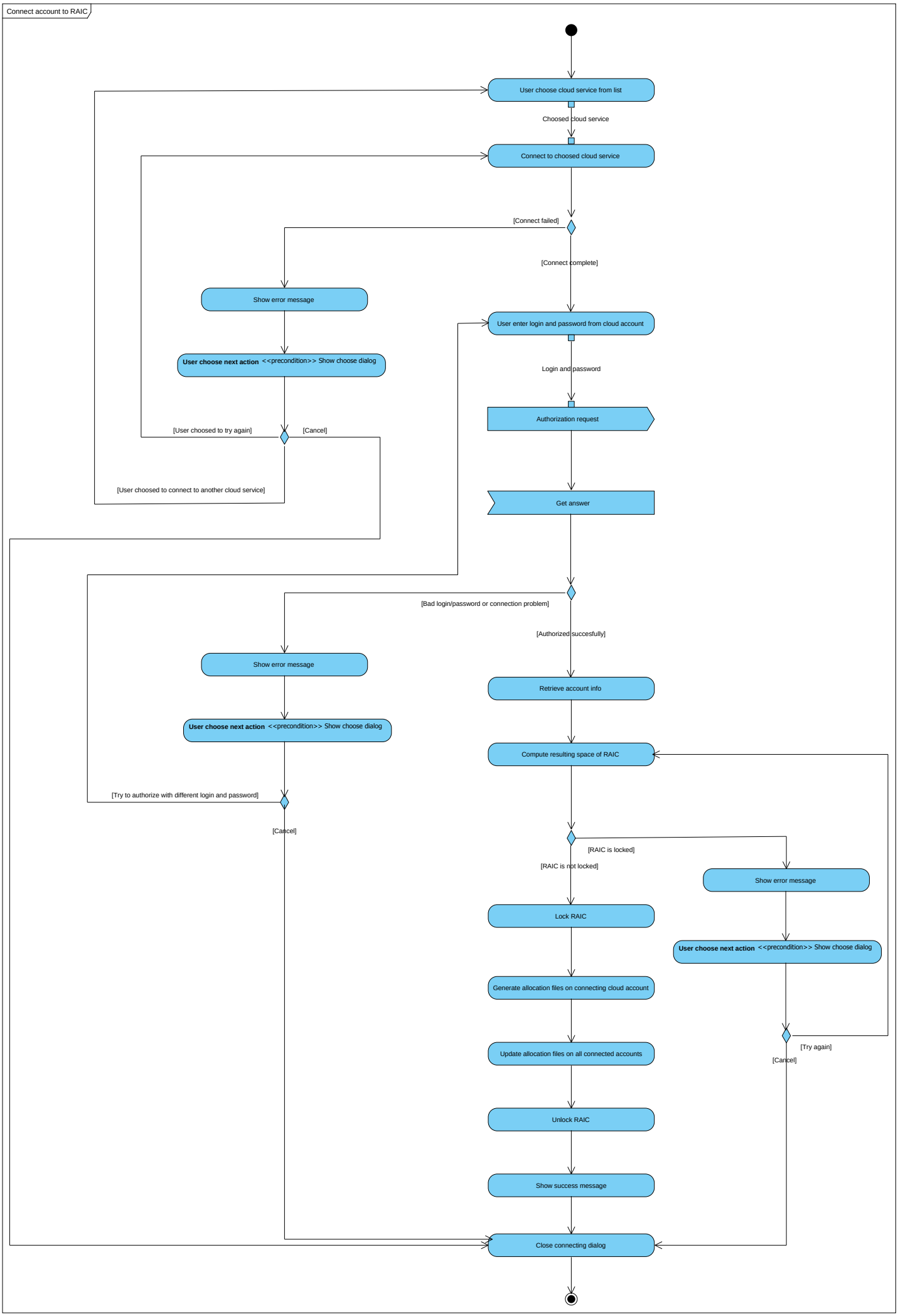

Deleting files
Connect to any connected cloud account
[Connect failed]
[Connect complete]
Show error message
User modify auto-filled login and password if needed
User choose next action <<precondition>> Show choose dialog
Login and password
Authorization request
[User choosed to try again]
[Cancel]
Get answer
[Bad login/password or connection problem]
[Authorized succesfully]
Show error message
Retrieve account info
User choose next action <<precondition>> Show choose dialog
[RAIC is locked]
[Try to authorize with different login and password]
[Cancel]
[RAIC is not locked]
Show error message
Lock RAIC
User choose next action <<precondition>> Show choose dialog
User choose files to delete
Computing cloud accounts with needed information
[Try again]
[Cancel]
Conntecting to computed accounts
Deleting parts of file
Update allocation files on all connected accounts
Unlock RAIC
Close deleting dialog

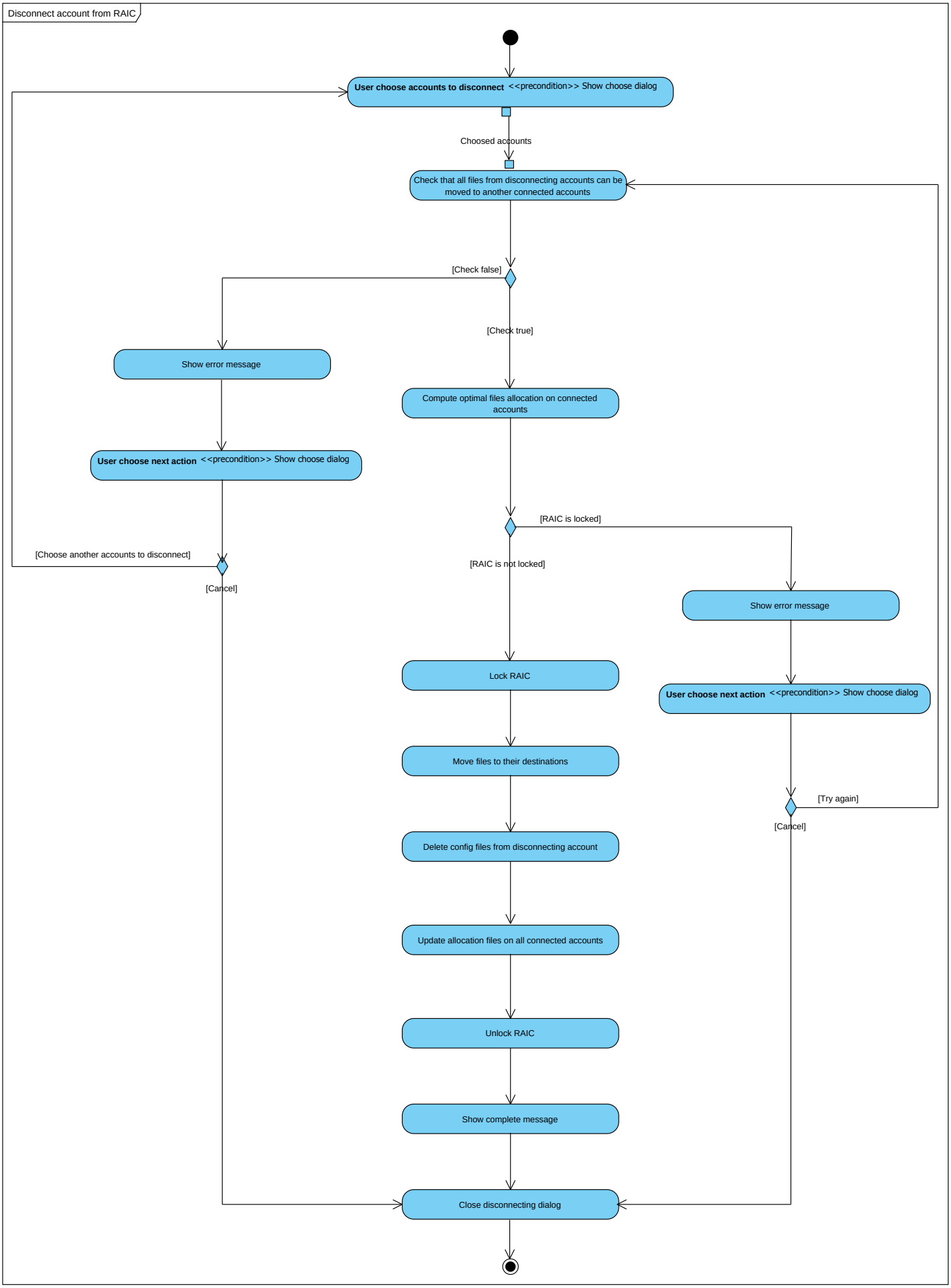

Downloading files
Connect to any connected cloud account
[Connect failed]
[Connect complete]
Show error message
User choose next action <<precondition>> Show choose dialog
User modify auto-filled login and password if needed
Login and password
Authorization request
[User choosed to try again]
[Cancel]
Get answer
[Bad login/password or connection problem]
[Authorized succesfully]
Show error message
User choose next action <<precondition>> Show choose dialog
Retrieve account info
[Try to authorize with different login and password]
[Cancel]
[RAIC is locked]
[RAIC is not locked]
Show error message
Lock RAIC
User choose files to download
User choose next action <<precondition>> Show choose dialog
Computing cloud accounts with needed information
[Try again]
[Cancel]
Conntecting to computed accounts
Downloading parts of file
Generate unify file
Unlock RAIC
Close downloading dialog

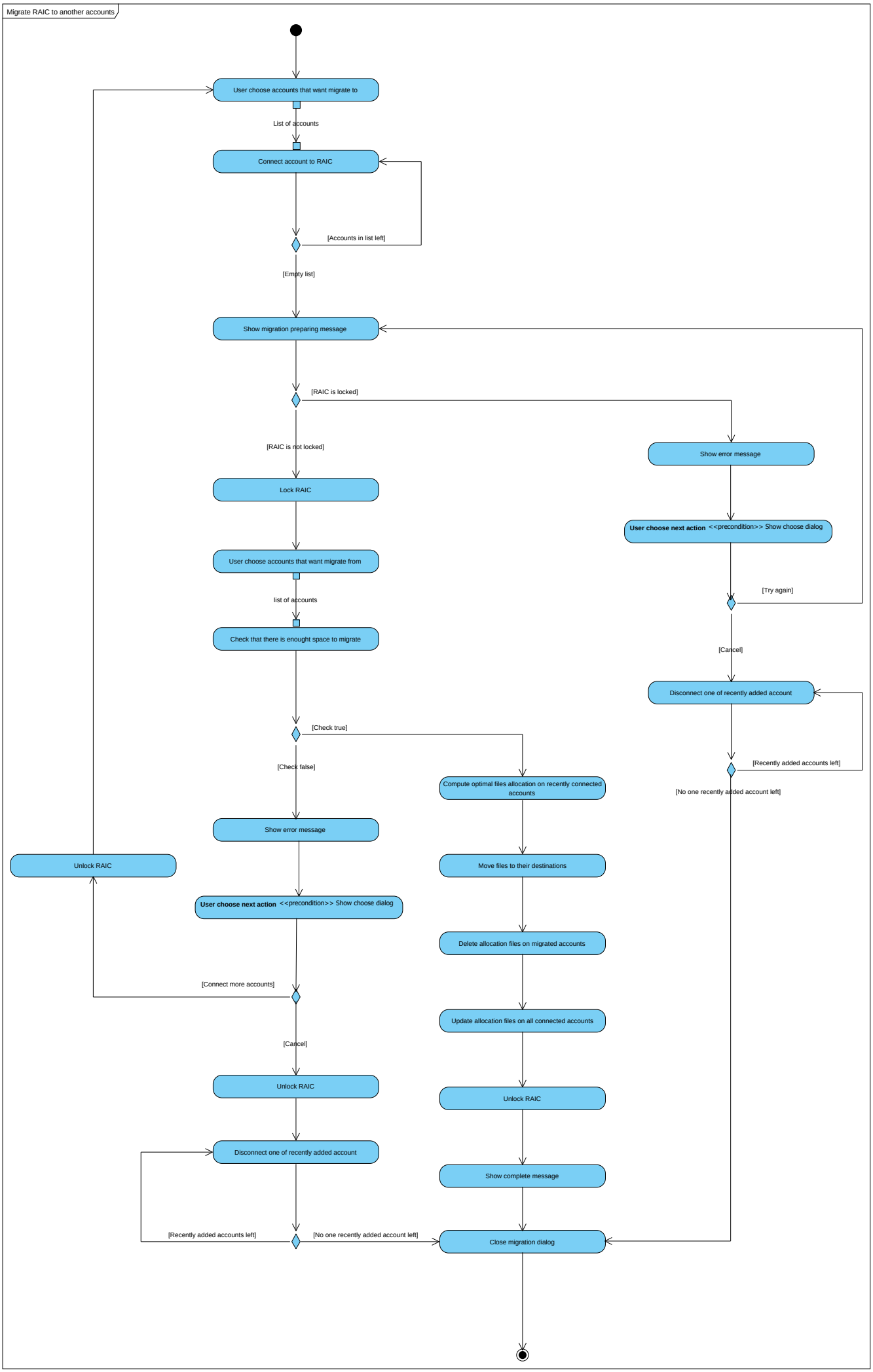

Lock RAIC
Get list of connected accounts from config of RAIC
List of accounts
Connect to one account from list
Set lock parameter value in allocation files to local instant of application
[Account in list left]
[No one account in list left]

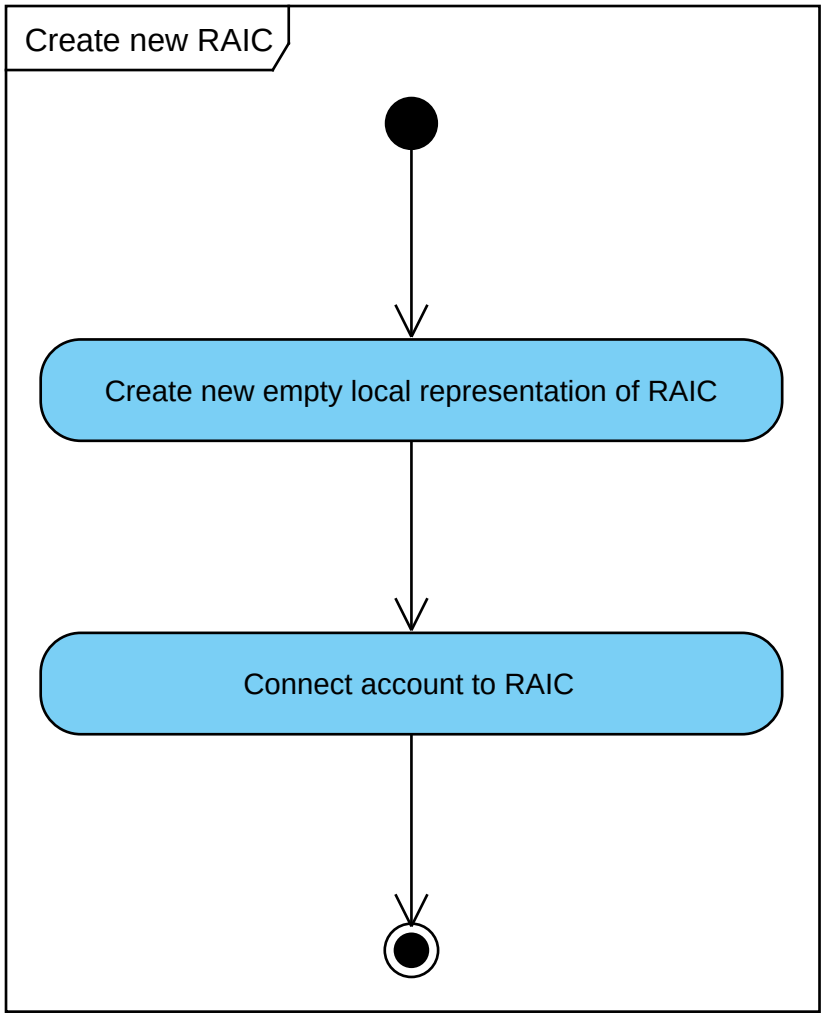

Unlock RAIC
Get list of connected accounts from config of RAIC
List of accounts
Connect to one account from list
Reset lock parameter value in allocation files
[Account in list left]
[No one account in list left]

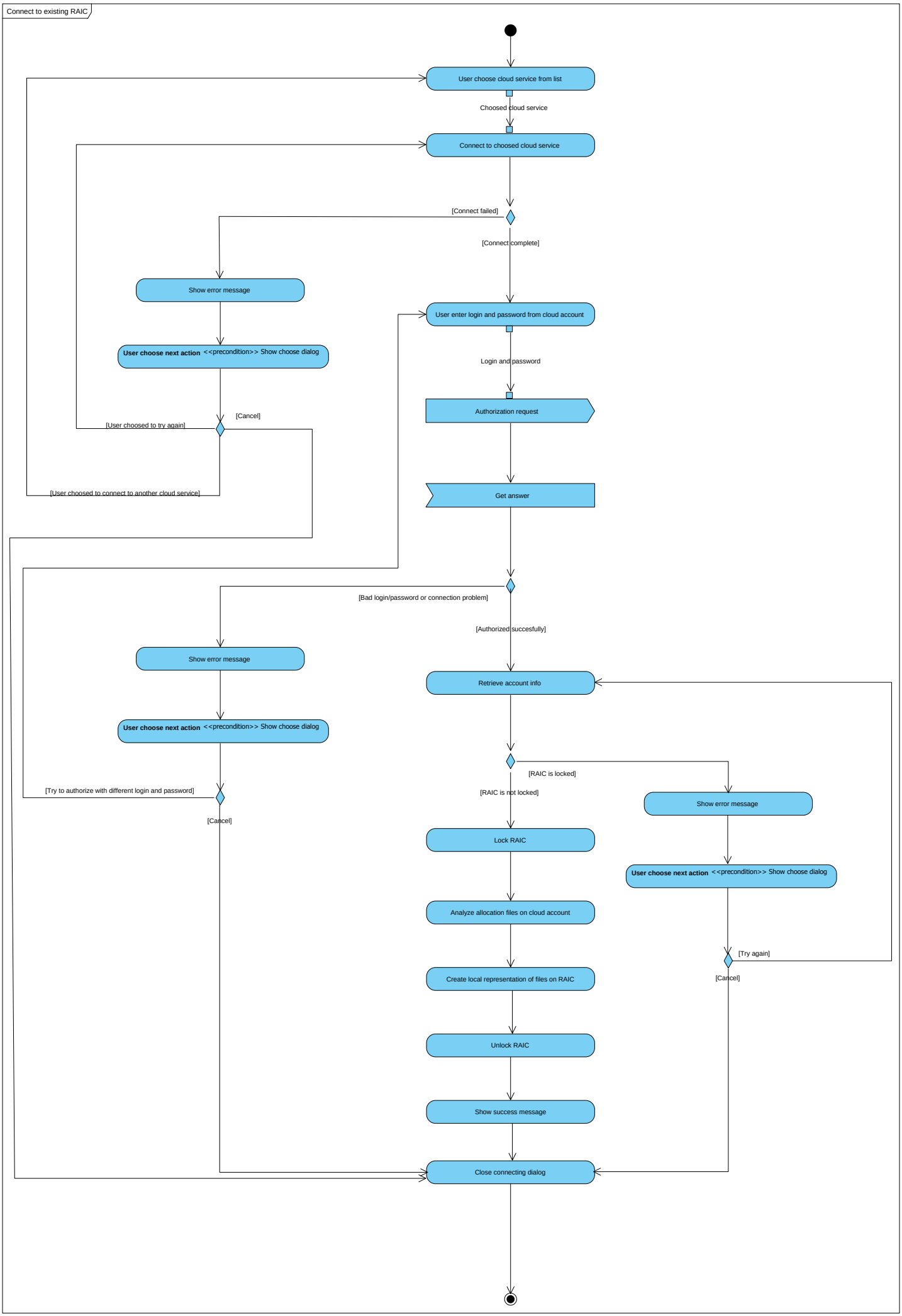

Manage RAIC
User choose manage action
<<precondition>> Show choose dialog
[User choose to connect cloud account]
[User choose to migrate RAIC]
[User choose to disconnect cloud account]
Connect account to RAIC
Disconnect account from RAIC
Migrate RAIC to another accounts
[User want to manage something else]
[User ended to manage RAIC]

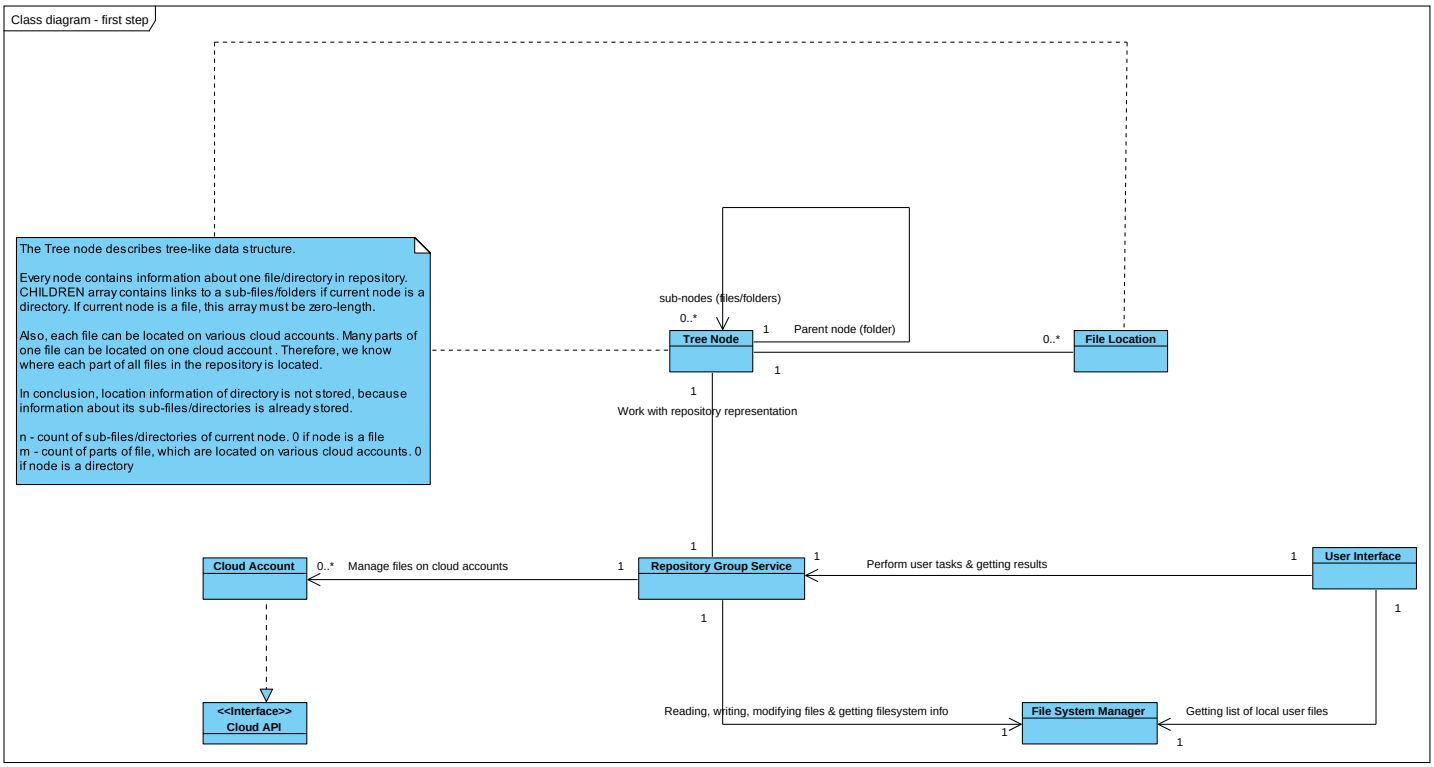

Connect account to RAIC
User choose cloud service from list
Choosed cloud service
Connect to choosed cloud service
[Connect failed]
[Connect complete]
Show error message
User enter login and password from cloud account
User choose next action <<precondition>> Show choose dialog
Login and password
Authorization request
[User choosed to try again]
[Cancel]
[User choosed to connect to another cloud service]
Get answer
[Bad login/password or connection problem]
[Authorized succesfully]
Show error message
Retrieve account info
User choose next action <<precondition>> Show choose dialog
Compute resulting space of RAIC
[Try to authorize with different login and password]
[Cancel]
[RAIC is locked]
[RAIC is not locked]
Show error message
Lock RAIC
User choose next action <<precondition>> Show choose dialog
Generate allocation files on connecting cloud account
[Try again]
[Cancel]
Update allocation files on all connected accounts
Unlock RAIC
Show success message
Close connecting dialog

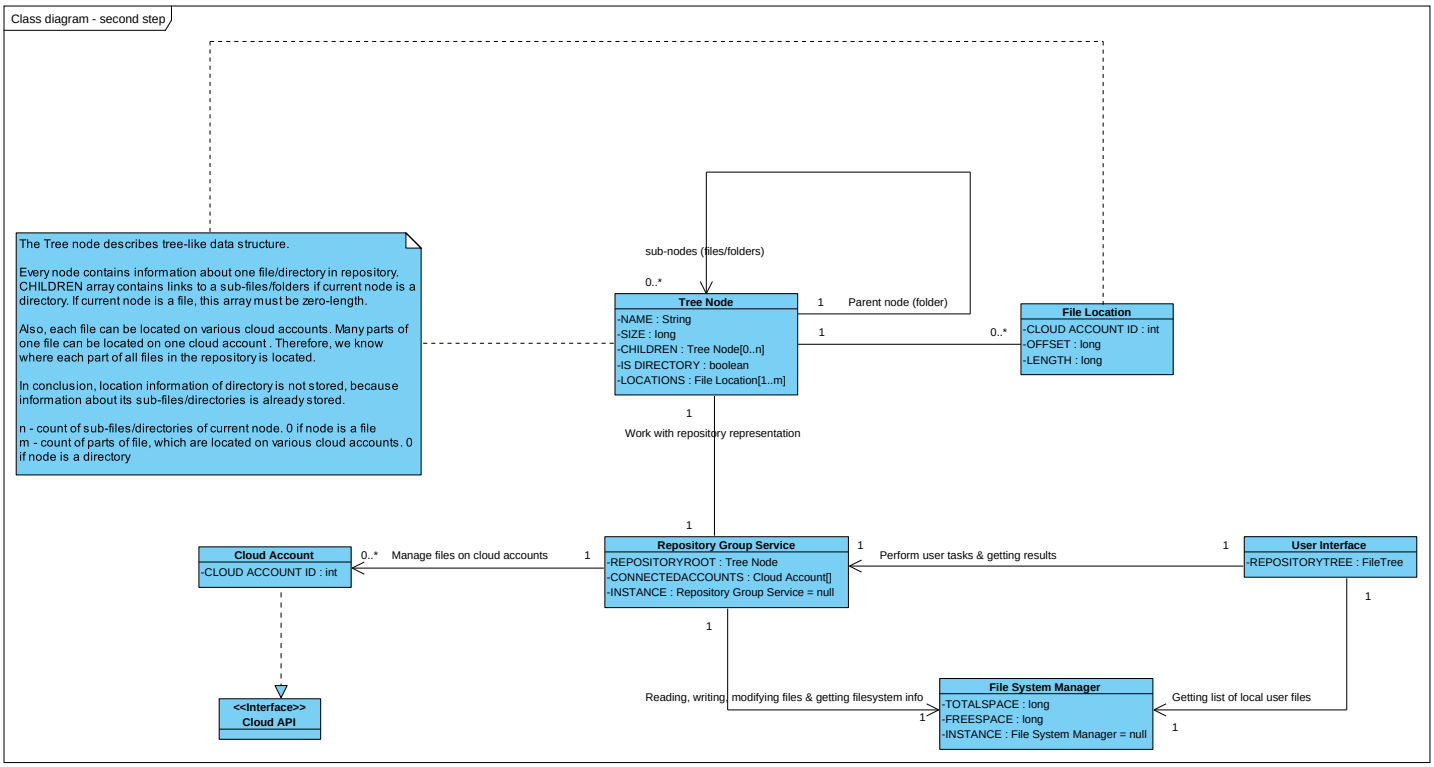

Disconnect account from RAIC
User choose accounts to disconnect <<precondition>> Show choose dialog
Choosed accounts
Check that all files from disconnecting accounts can be moved to another connected accounts
[Check false]
[Check true]
Show error message
Compute optimal files allocation on connected accounts
User choose next action <<precondition>> Show choose dialog
[RAIC is locked]
[Choose another accounts to disconnect]
[RAIC is not locked]
[Cancel]
Show error message
Lock RAIC
User choose next action <<precondition>> Show choose dialog
Move files to their destinations
[Try again]
[Cancel]
Delete config files from disconnecting account
Update allocation files on all connected accounts
Unlock RAIC
Show complete message
Close disconnecting dialog

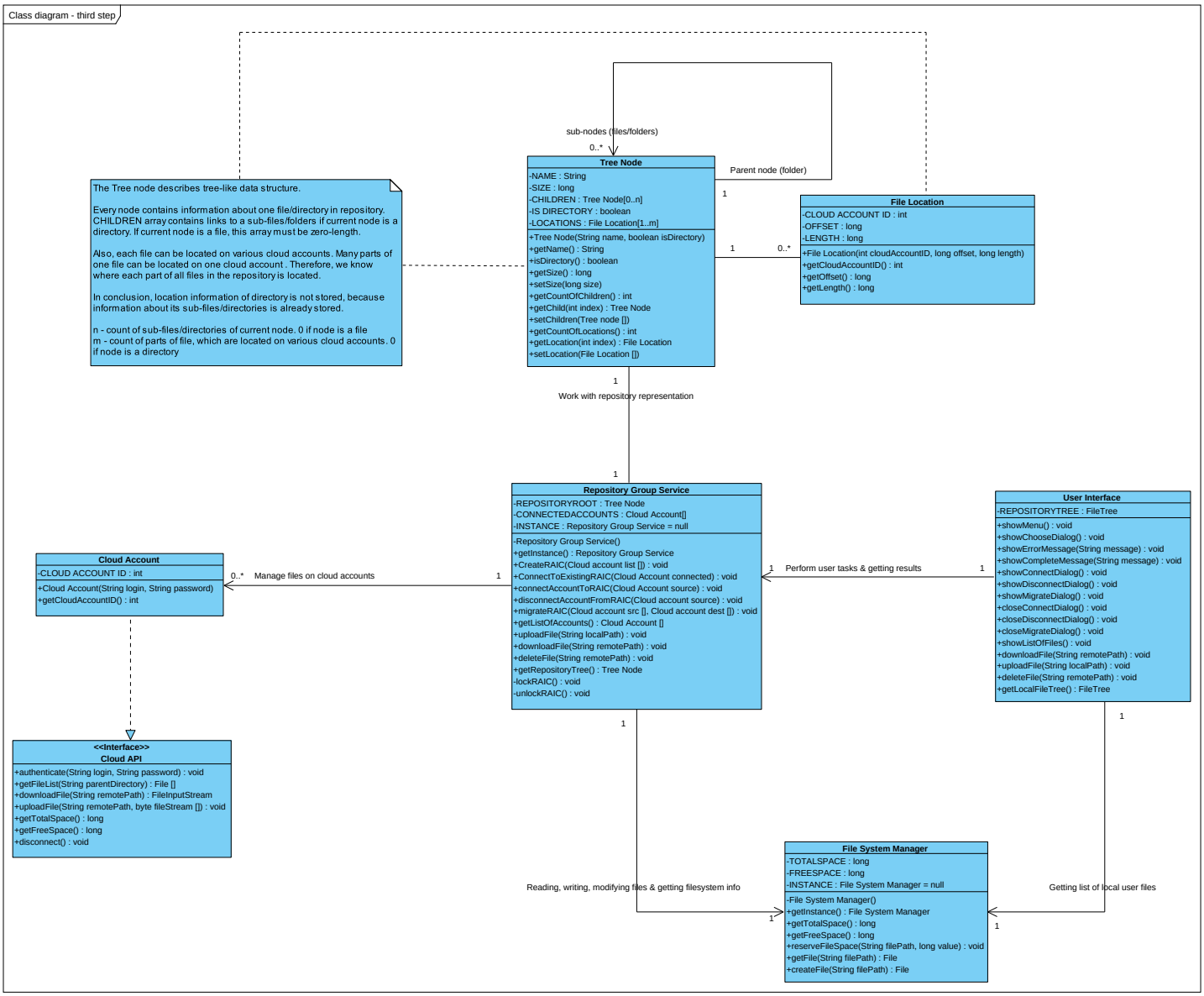

Migrate RAIC to another accounts
User choose accounts that want migrate to
List of accounts
Connect account to RAIC
[Accounts in list left]
[Empty list]
Show migration preparing message
[RAIC is locked]
[RAIC is not locked]
Show error message
Lock RAIC
User choose next action <<precondition>> Show choose dialog
User choose accounts that want migrate from
[Try again]
list of accounts
Check that there is enought space to migrate
[Cancel]
Disconnect one of recently added account
[Check true]
[Recently added accounts left]
[Check false]
[No one recently added account left]
Compute optimal files allocation on recently connected accounts
Show error message
Unlock RAIC
Move files to their destinations
User choose next action <<precondition>> Show choose dialog
Delete allocation files on migrated accounts
[Connect more accounts]
Update allocation files on all connected accounts
[Cancel]
Unlock RAIC
Unlock RAIC
Disconnect one of recently added account
Show complete message
[Recently added accounts left]
[No one recently added account left]
Close migration dialog

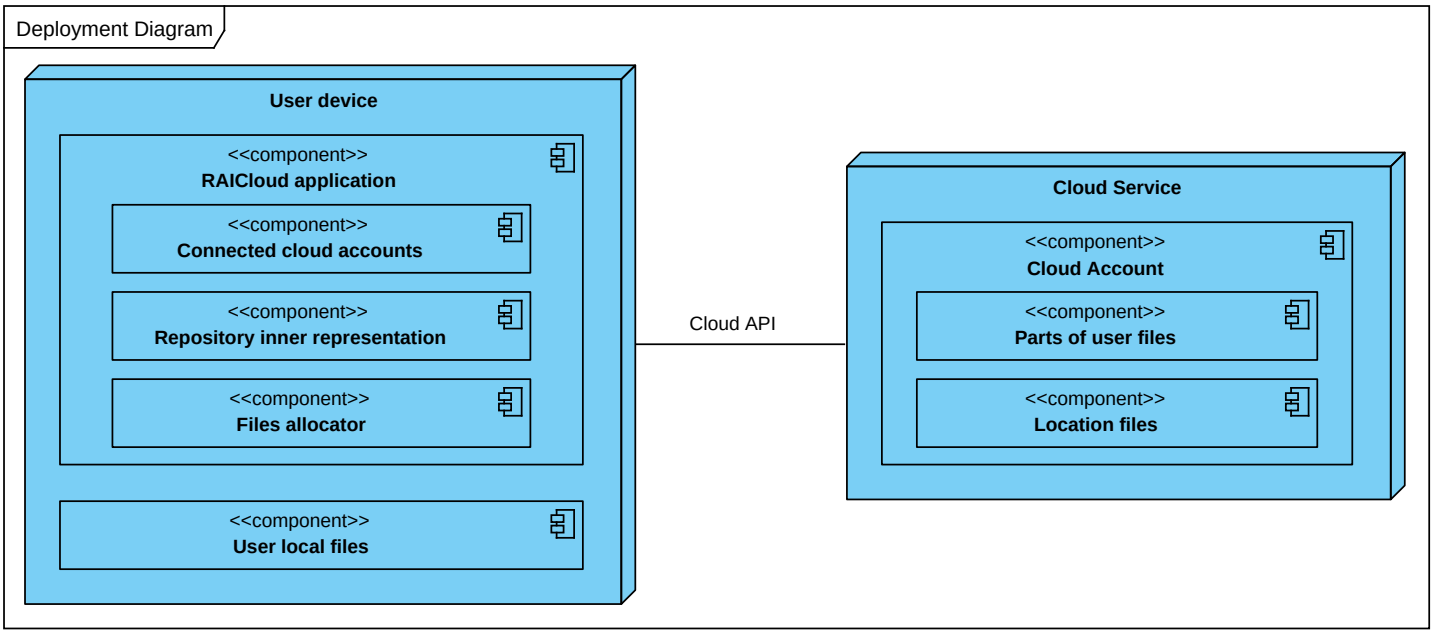

Create new RAIC
Create new empty local representation of RAIC
Connect account to RAIC

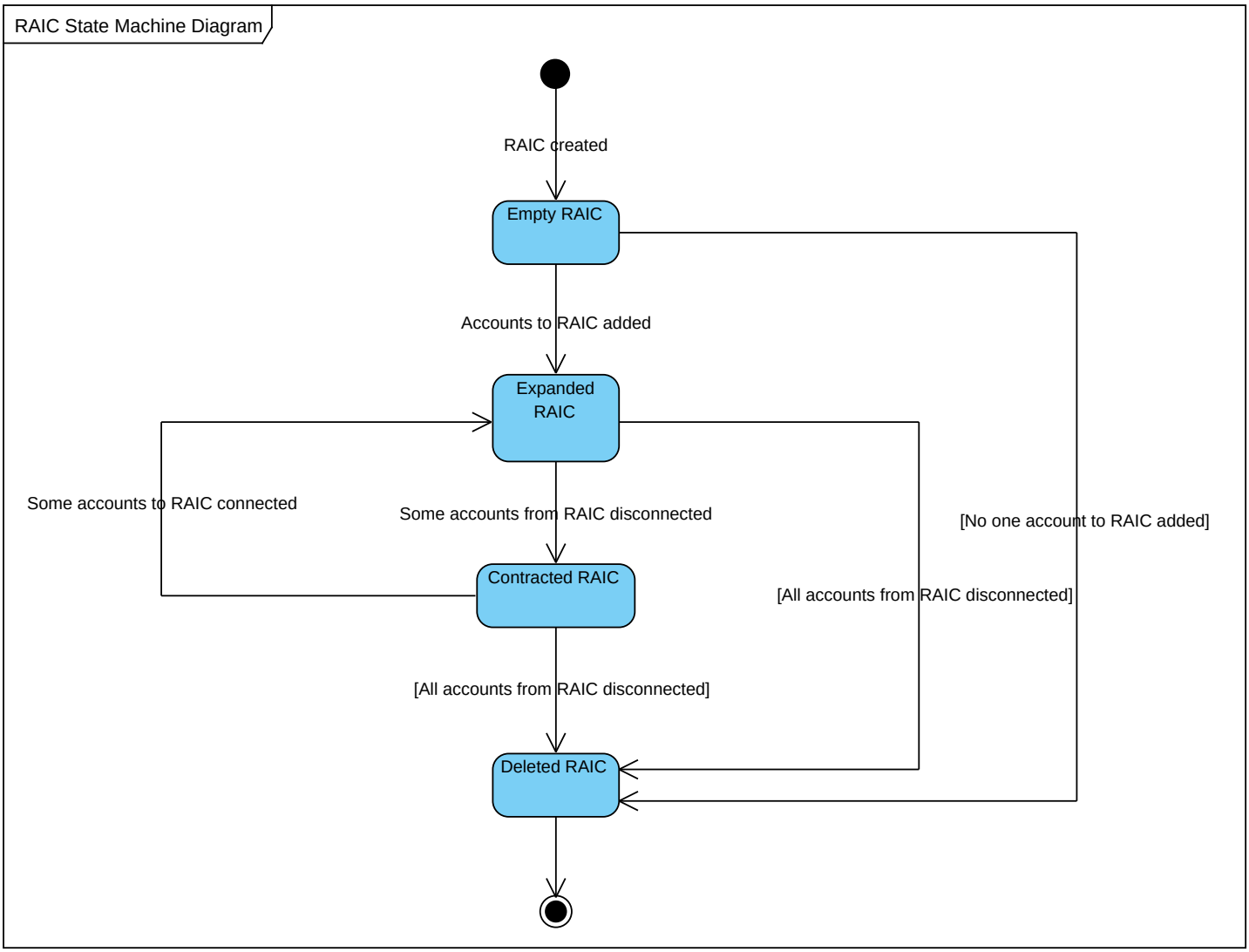

Connect to existing RAIC
User choose cloud service from list
Choosed cloud service
Connect to choosed cloud service
[Connect failed]
[Connect complete]
Show error message
User enter login and password from cloud account
User choose next action <<precondition>> Show choose dialog
Login and password
Authorization request
[Cancel]
[User choosed to try again]
[User choosed to connect to another cloud service]
Get answer
[Bad login/password or connection problem]
[Authorized succesfully]
Show error message
Retrieve account info
User choose next action <<precondition>> Show choose dialog
[RAIC is locked]
[Try to authorize with different login and password]
[RAIC is not locked]
Show error message
[Cancel]
Lock RAIC
User choose next action <<precondition>> Show choose dialog
Analyze allocation files on cloud account
[Try again]
[Cancel]
Create local representation of files on RAIC
Unlock RAIC
Show success message
Close connecting dialog

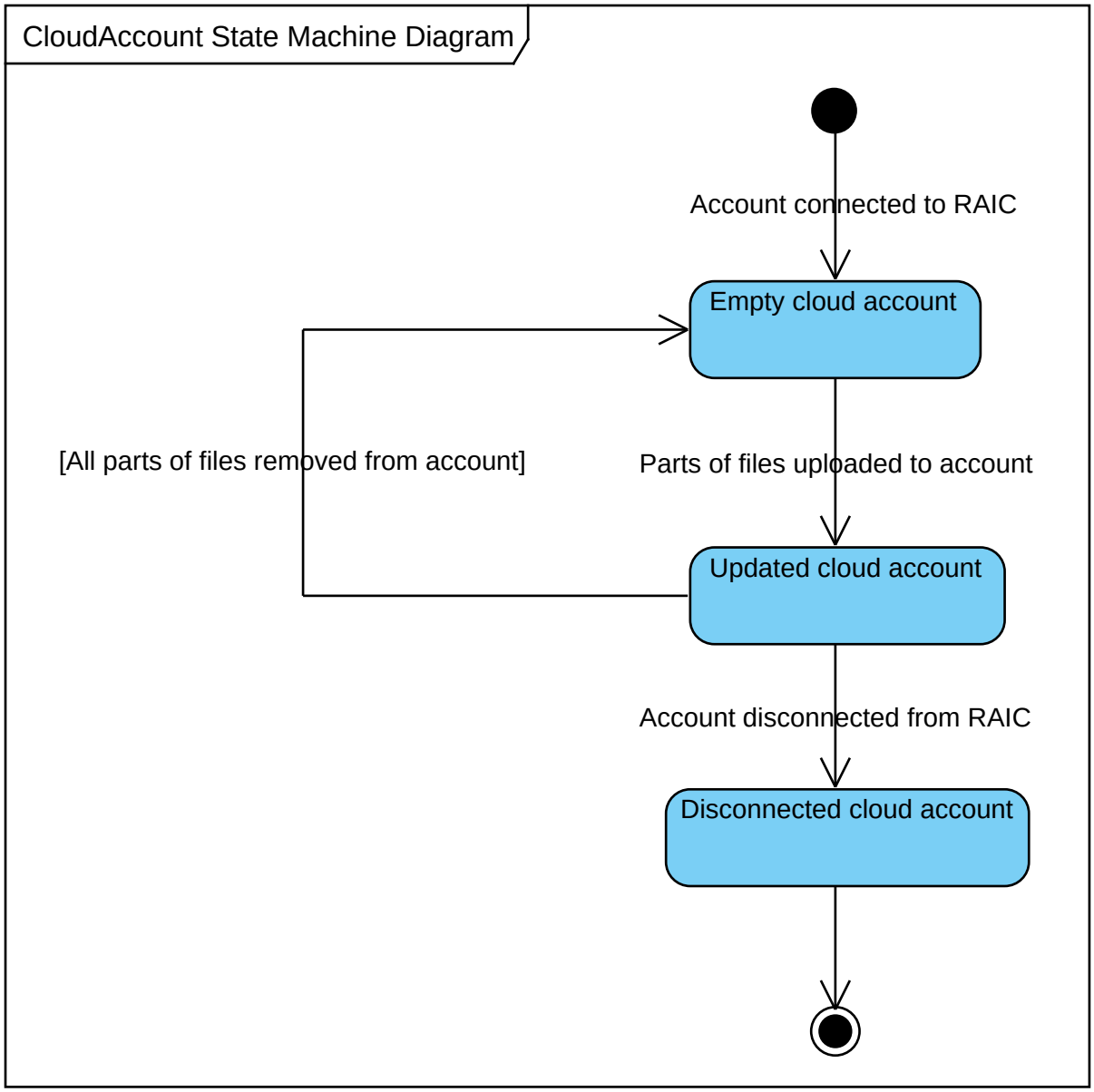

Class diagram - first step
The Tree node describes tree-like data structure.
Every node contains information about one file/directory in repository. CHILDREN array contains links to a sub-files/folders if current node is a directory. If current node is a file, this array must be zero-length.
Also, each file can be located on various cloud accounts. Many parts of one file can be located on one cloud account . Therefore, we know where each part of all files in the repository is located.
In conclusion, location information of directory is not stored, because information about its sub-files/directories is already stored.
n - count of sub-files/directories of current node. 0 if node is a file
m - count of parts of file, which are located on various cloud accounts. 0 if node is a directory
sub-nodes (files/folders)
0..*
Tree Node
1
Parent node (folder)
1
0..*
File Location
1
Work with repository representation
1
Cloud Account
0..*
Manage files on cloud accounts
1
Repository Group Service
1
Perform user tasks & getting results
1
User Interface
1
1
<<Interface>>
Cloud API
Reading, writing, modifying files & getting filesystem info
1
File System Manager
Getting list of local user files
1

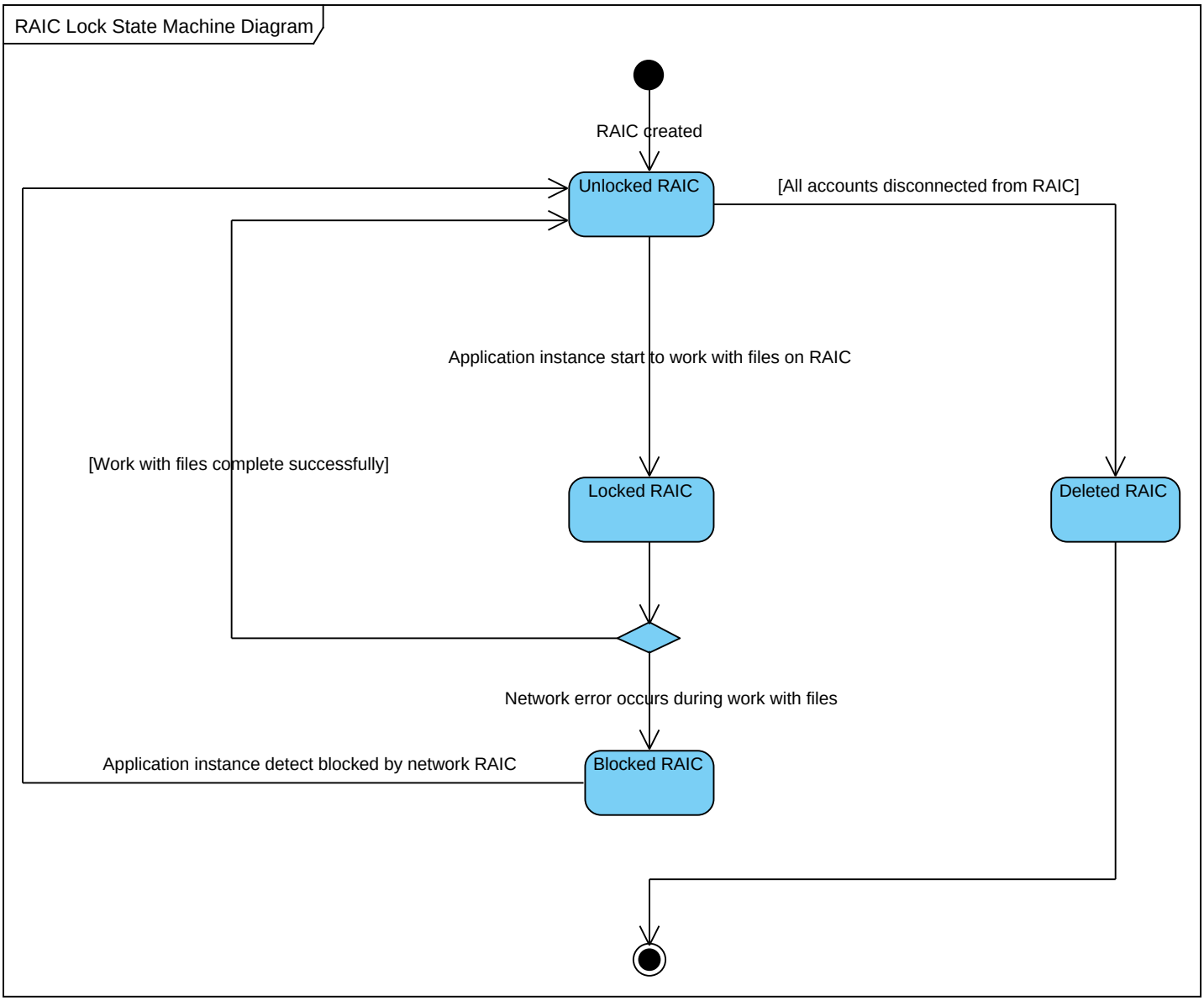

Class diagram - second step
The Tree node describes tree-like data structure.
Every node contains information about one file/directory in repository. CHILDREN array contains links to a sub-files/folders if current node is a directory. If current node is a file, this array must be zero-length.
Also, each file can be located on various cloud accounts. Many parts of one file can be located on one cloud account . Therefore, we know where each part of all files in the repository is located.
In conclusion, location information of directory is not stored, because information about its sub-files/directories is already stored.
n - count of sub-files/directories of current node. 0 if node is a file
m - count of parts of file, which are located on various cloud accounts. 0 if node is a directory
sub-nodes (files/folders)
0..*
Tree Node
-NAME : String
-SIZE : long
-CHILDREN : Tree Node[0..n]
-IS DIRECTORY : boolean
-LOCATIONS : File Location[1..m]
1
Parent node (folder)
1
0..*
File Location
-CLOUD ACCOUNT ID : int
-OFFSET : long
-LENGTH : long
1
Work with repository representation
1
Cloud Account
-CLOUD ACCOUNT ID : int
0..*
Manage files on cloud accounts
1
Repository Group Service
-REPOSITORYROOT : Tree Node
-CONNECTEDACCOUNTS : Cloud Account[]
-INSTANCE : Repository Group Service = null
1
Perform user tasks & getting results
1
User Interface
-REPOSITORYTREE : FileTree
1
1
<<Interface>>
Cloud API
Reading, writing, modifying files & getting filesystem info
1
File System Manager
-TOTALSPACE : long
-FREESPACE : long
-INSTANCE : File System Manager = null
Getting list of local user files
1

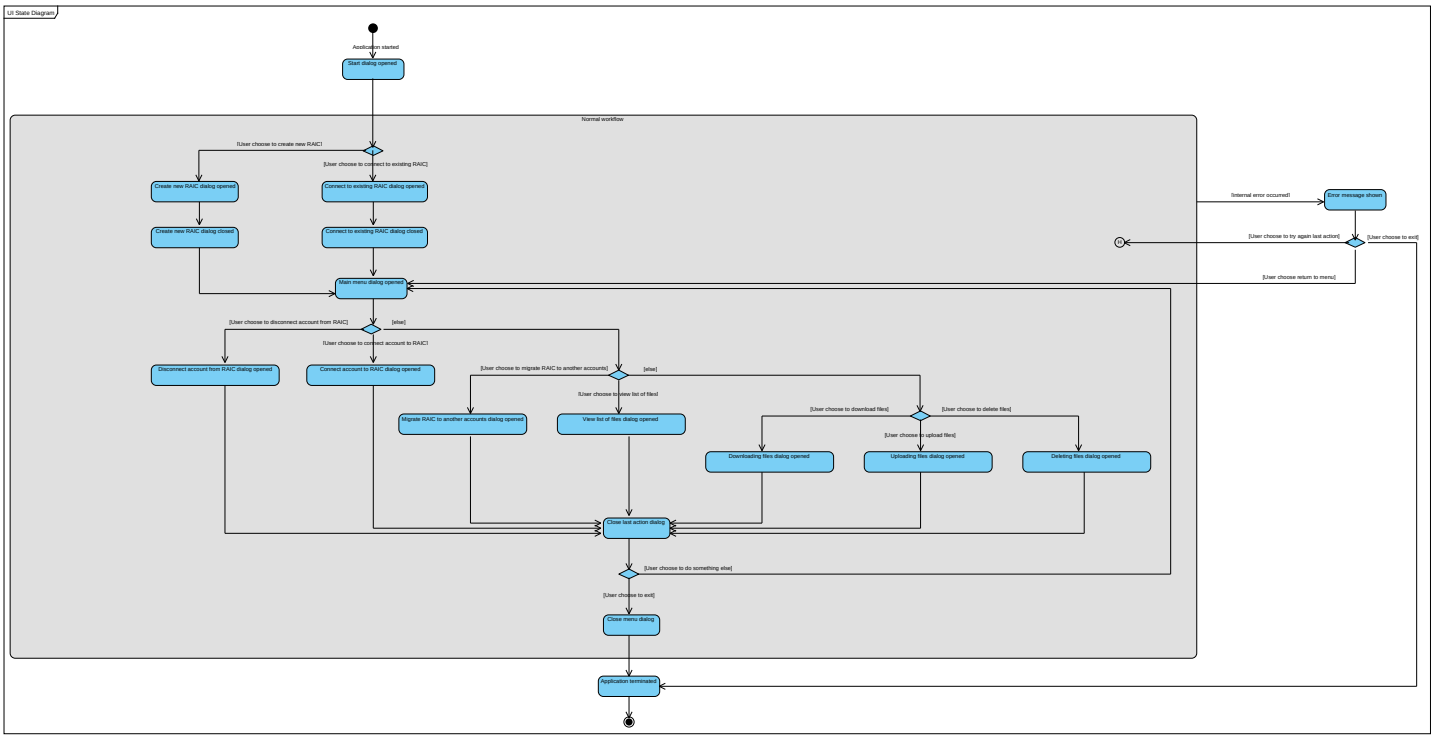

Class diagram - third step
sub-nodes (files/folders)
0..*
Tree Node
-NAME : String
-SIZE : long
-CHILDREN : Tree Node[0..n]
-IS DIRECTORY : boolean
-LOCATIONS : File Location[1..m]
+Tree Node(String name, boolean isDirectory)
+getName() : String
+isDirectory() : boolean
+getSize() : long
+setSize(long size)
+getCountOfChildren() : int
+getChild(int index) : Tree Node
+setChildren(Tree node [])
+getCountOfLocations() : int
+getLocation(int index) : File Location
+setLocation(File Location [])
Parent node (folder)
1
1
0..*
File Location
-CLOUD ACCOUNT ID : int
-OFFSET : long
-LENGTH : long
+File Location(int cloudAccountID, long offset, long length)
+getCloudAccountID() : int
+getOffset() : long
+getLength() : long
The Tree node describes tree-like data structure.
Every node contains information about one file/directory in repository. CHILDREN array contains links to a sub-files/folders if current node is a directory. If current node is a file, this array must be zero-length.
Also, each file can be located on various cloud accounts. Many parts of one file can be located on one cloud account . Therefore, we know where each part of all files in the repository is located.
In conclusion, location information of directory is not stored, because information about its sub-files/directories is already stored.
n - count of sub-files/directories of current node. 0 if node is a file
m - count of parts of file, which are located on various cloud accounts. 0 if node is a directory
1
Work with repository representation
1
Repository Group Service
-REPOSITORYROOT : Tree Node
-CONNECTEDACCOUNTS : Cloud Account[]
-INSTANCE : Repository Group Service = null
-Repository Group Service()
+getInstance() : Repository Group Service
+CreateRAIC(Cloud account list []) : void
+ConnectToExistingRAIC(Cloud Account connected) : void
+connectAccountToRAIC(Cloud Account source) : void
+disconnectAccountFromRAIC(Cloud account source) : void
+migrateRAIC(Cloud account src [], Cloud account dest []) : void
+getListOfAccounts() : Cloud Account []
+uploadFile(String localPath) : void
+downloadFile(String remotePath) : void
+deleteFile(String remotePath) : void
+getRepositoryTree() : Tree Node
-lockRAIC() : void
-unlockRAIC() : void
User Interface
-REPOSITORYTREE : FileTree
+showMenu() : void
+showChooseDialog() : void
+showErrorMessage(String message) : void
+showCompleteMessage(String message) : void
+showConnectDialog() : void
+showDisconnectDialog() : void
+showMigrateDialog() : void
+closeConnectDialog() : void
+closeDisconnectDialog() : void
+closeMigrateDialog() : void
+showListOfFiles() : void
+downloadFile(String remotePath) : void
+uploadFile(String localPath) : void
+deleteFile(String remotePath) : void
+getLocalFileTree() : FileTree
Cloud Account
-CLOUD ACCOUNT ID : int
+Cloud Account(String login, String password)
+getCloudAccountID() : int
0..*
Manage files on cloud accounts
1
1
Perform user tasks & getting results
1
<<Interface>>
Cloud API
+authenticate(String login, String password) : void
+getFileList(String parentDirectory) : File []
+downloadFile(String remotePath) : FileInputStream
+uploadFile(String remotePath, byte fileStream []) : void
+getTotalSpace() : long
+getFreeSpace() : long
+disconnect() : void
1
1
Reading, writing, modifying files & getting filesystem info
Getting list of local user files
File System Manager
-TOTALSPACE : long
-FREESPACE : long
-INSTANCE : File System Manager = null
-File System Manager()
+getInstance() : File System Manager
+getTotalSpace() : long
+getFreeSpace() : long
+reserveFileSpace(String filePath, long value) : void
+getFile(String filePath) : File
+createFile(String filePath) : File
1
1

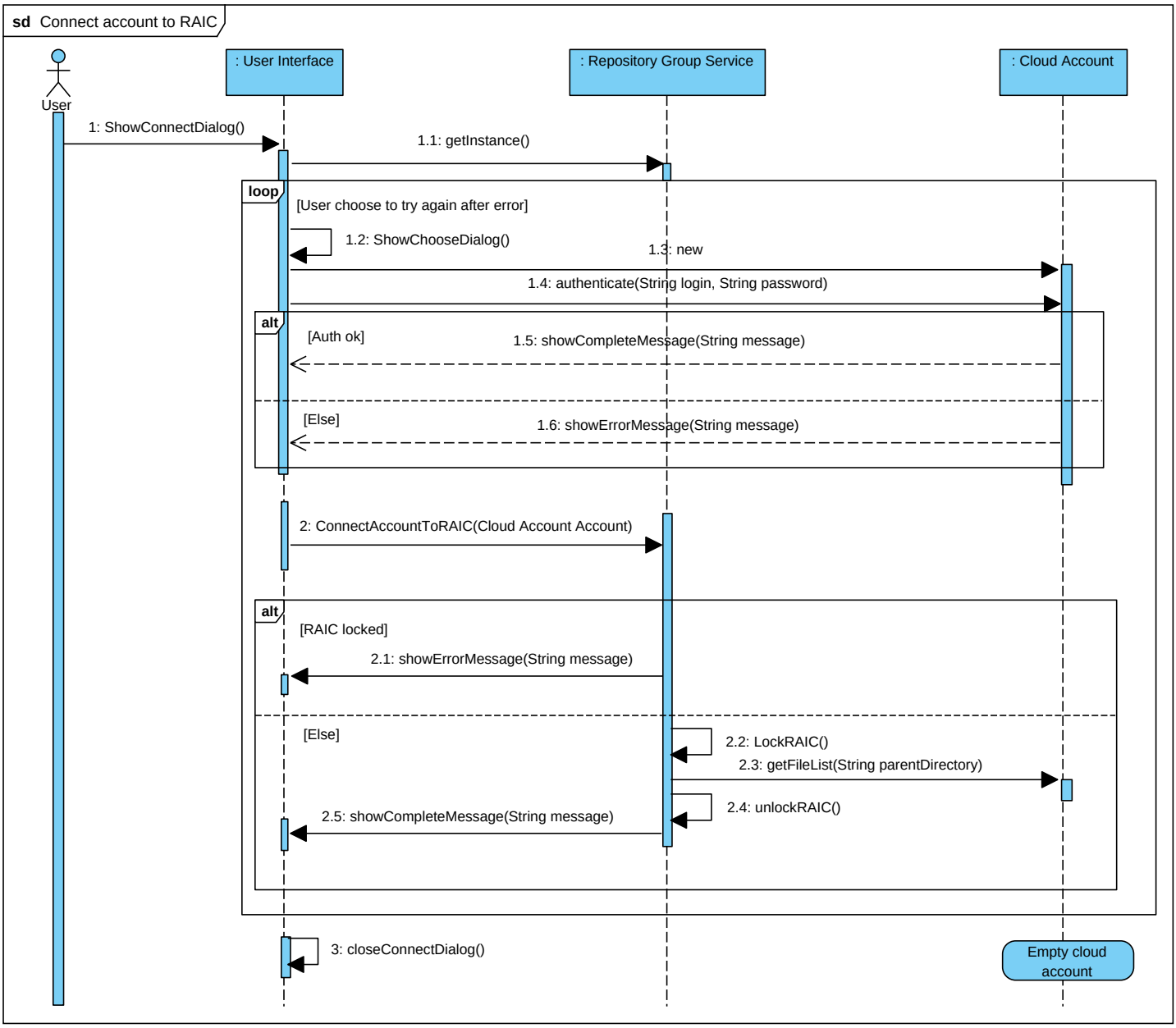

Deployment Diagram
User device
<<component>>
RAICloud application
<<component>>
Connected cloud accounts
<<component>>
Repository inner representation
<<component>>
Files allocator
<<component>>
User local files
Cloud API
Cloud Service
<<component>>
Cloud Account
<<component>>
Parts of user files
<<component>>
Location files

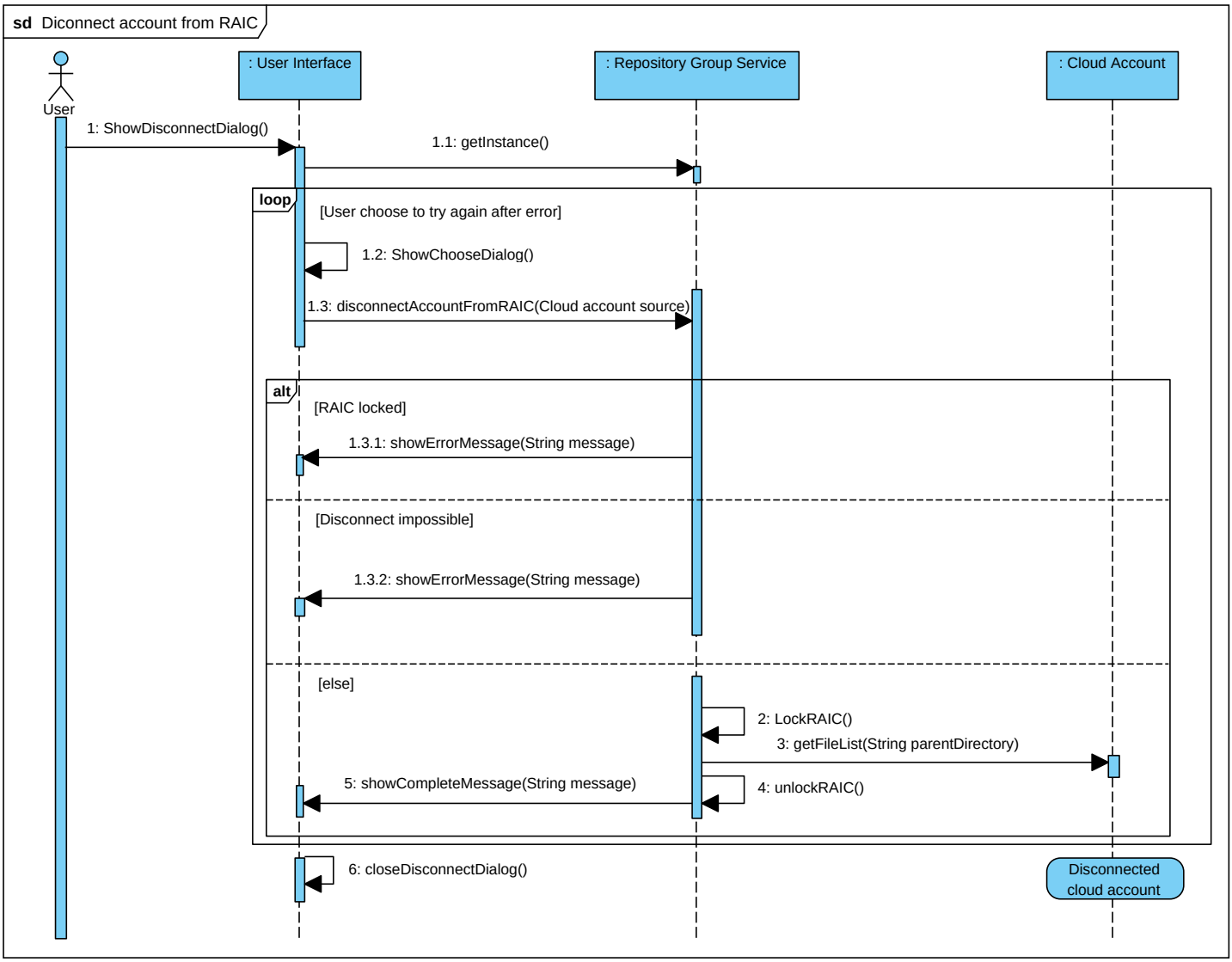

RAIC State Machine Diagram
RAIC created
Empty RAIC
Accounts to RAIC added
Expanded RAIC
Some accounts to RAIC connected
Some accounts from RAIC disconnected
[No one account to RAIC added]
Contracted RAIC
[All accounts from RAIC disconnected]
[All accounts from RAIC disconnected]
Deleted RAIC

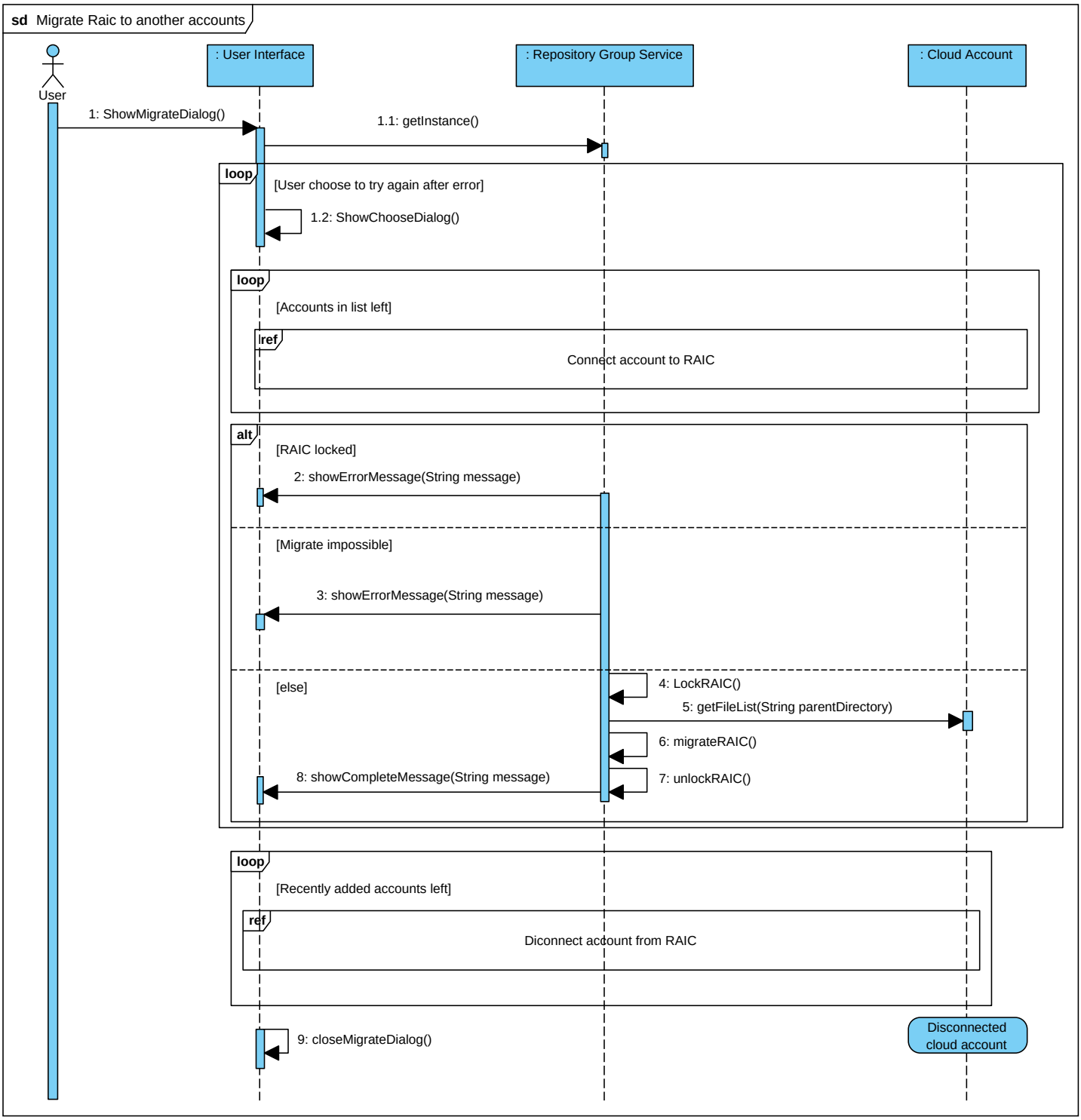

CloudAccount State Machine Diagram
Account connected to RAIC
Empty cloud account
[All parts of files removed from account]
Parts of files uploaded to account
Updated cloud account
Account disconnected from RAIC
Disconnected cloud account

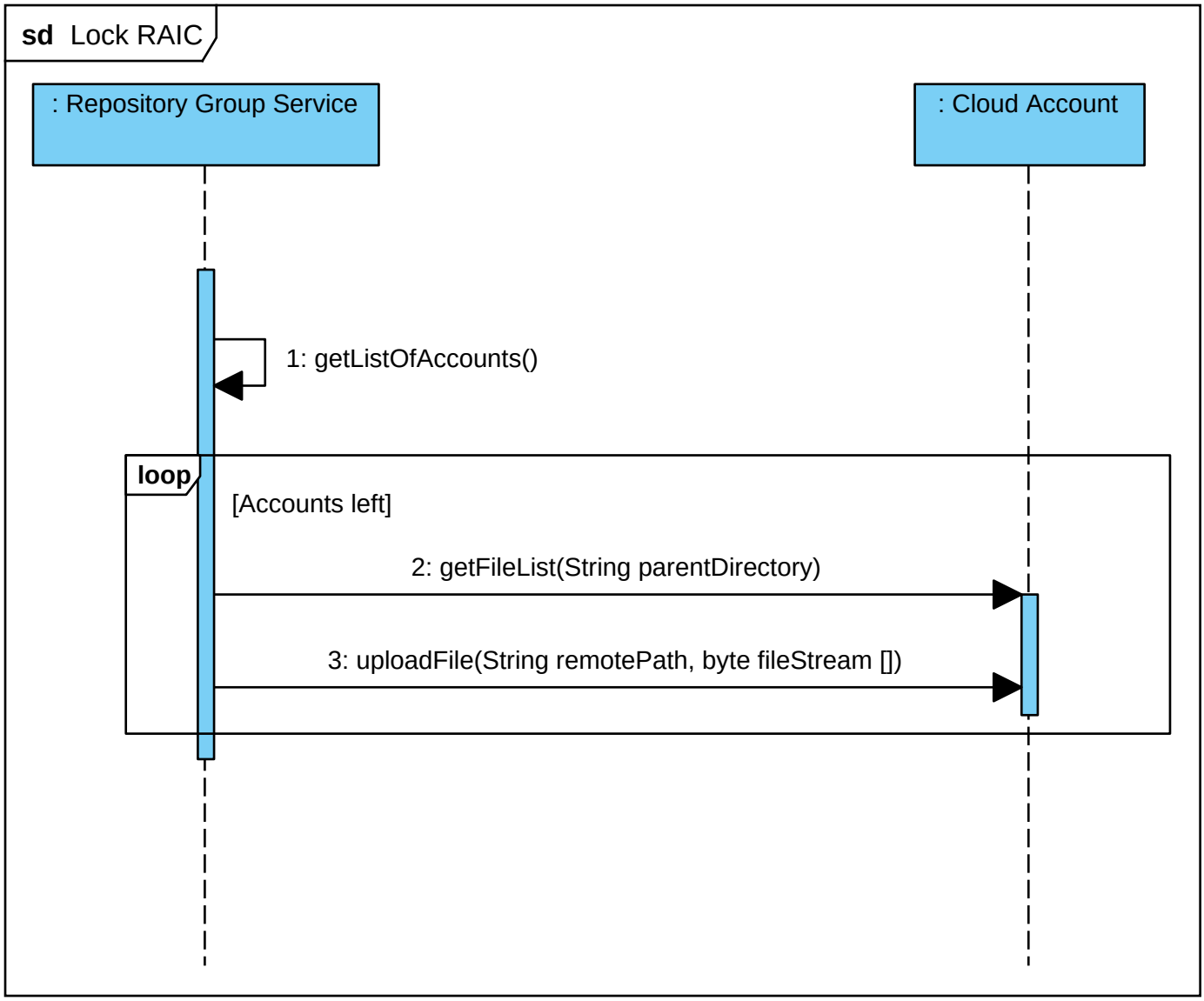

RAIC Lock State Machine Diagram
RAIC created
Unlocked RAIC
[All accounts disconnected from RAIC]
Application instance start to work with files on RAIC
[Work with files complete successfully]
Locked RAIC
Deleted RAIC
Network error occurs during work with files
Application instance detect blocked by network RAIC
Blocked RAIC

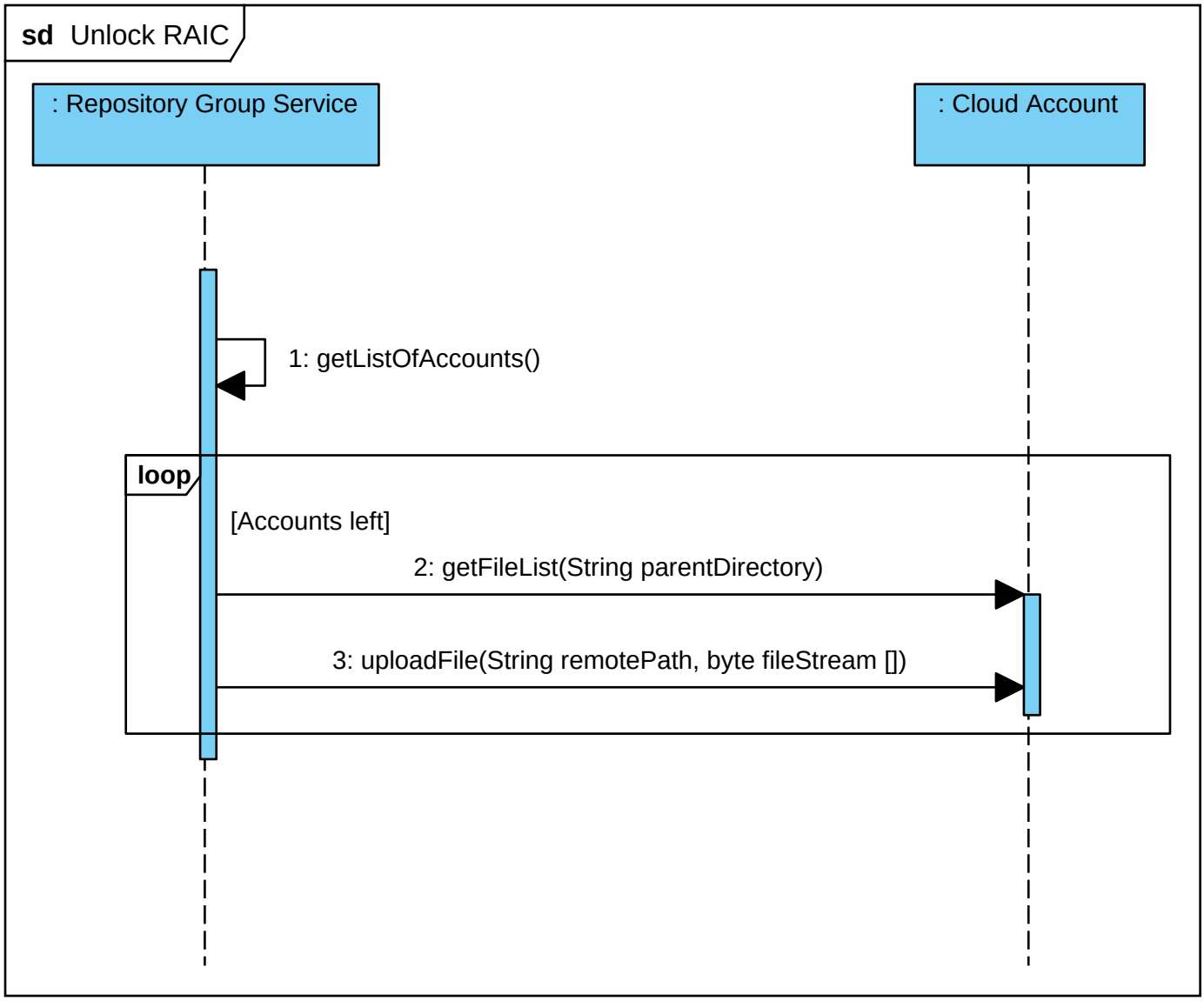

UI State Diagram
Application started
Start dialog opened
Normal workflow
[User choose to create new RAIC]
[User choose to connect to existing RAIC]
Create new RAIC dialog opened
Connect to existing RAIC dialog opened
Create new RAIC dialog closed
Connect to existing RAIC dialog closed
[Internal error occurred]
Error message shown
[User choose to try again last action]
[User choose to exit]
[User choose return to menu]
Main menu dialog opened
[User choose to disconnect account from RAIC]
[else]
[User choose to connect account to RAIC]
Disconnect account from RAIC dialog opened
Connect account to RAIC dialog opened
[User choose to migrate RAIC to another accounts]
[else]
[User choose to view list of files]
Migrate RAIC to another accounts dialog opened
View list of files dialog opened
[User choose to download files]
[User choose to delete files]
[User choose to upload files]
Downloading files dialog opened
Uploading files dialog opened
Deleting files dialog opened
Close last action dialog
[User choose to do something else]
[User choose to exit]
Close menu dialog
Application terminated

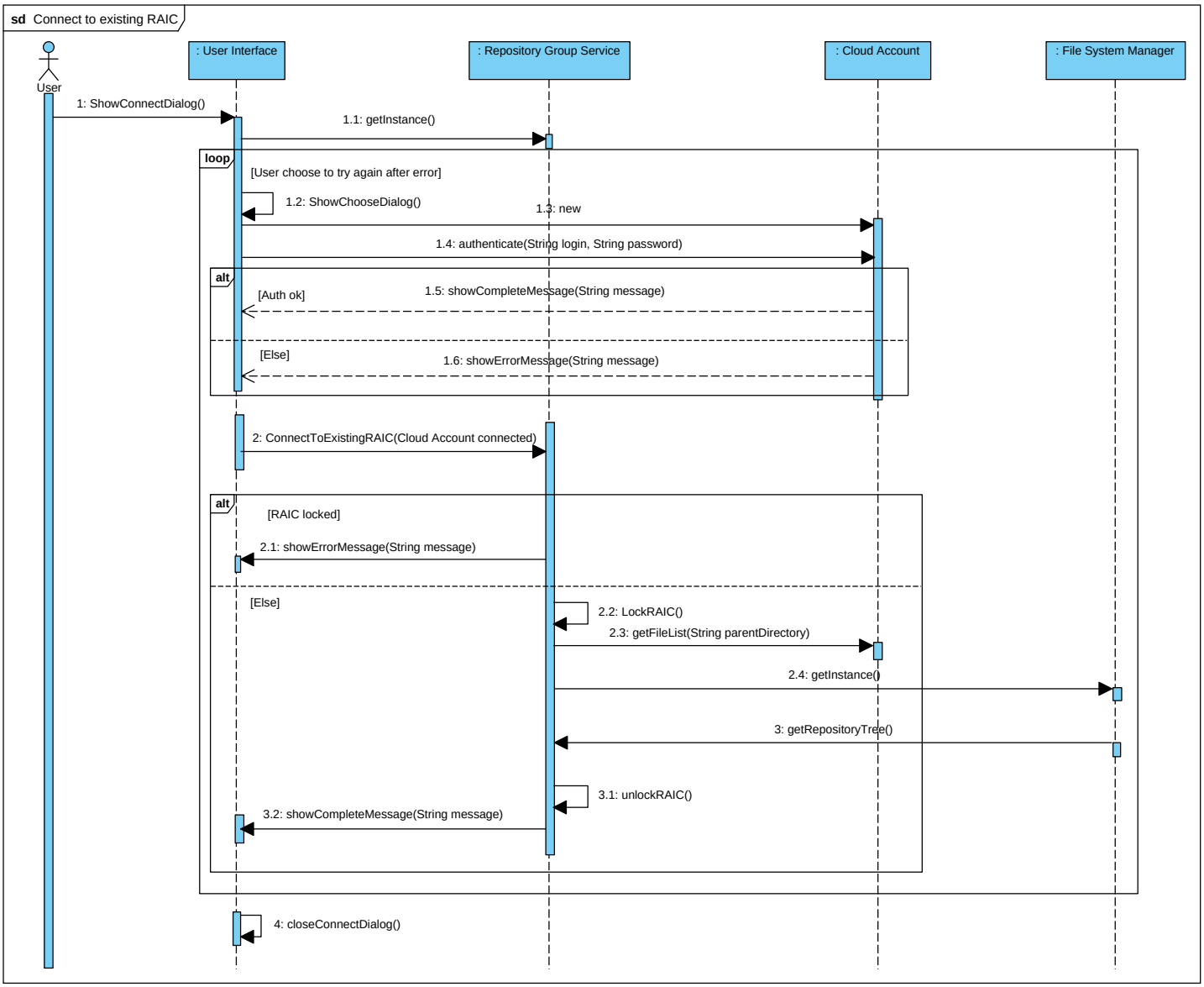

sd Connect account to RAIC
User
: User Interface
: Repository Group Service
: Cloud Account
1: ShowConnectDialog()
1.1: getInstance()
loop
[User choose to try again after error]
1.2: ShowChooseDialog()
1.3: new
1.4: authenticate(String login, String password)
alt
[Auth ok]
1.5: showCompleteMessage(String message)
[Else]
1.6: showErrorMessage(String message)
2: ConnectAccountToRAIC(Cloud Account Account)
alt
[RAIC locked]
2.1: showErrorMessage(String message)
[Else]
2.2: LockRAIC()
2.3: getFileList(String parentDirectory)
2.4: unlockRAIC()
2.5: showCompleteMessage(String message)
3: closeConnectDialog()
Empty cloud account

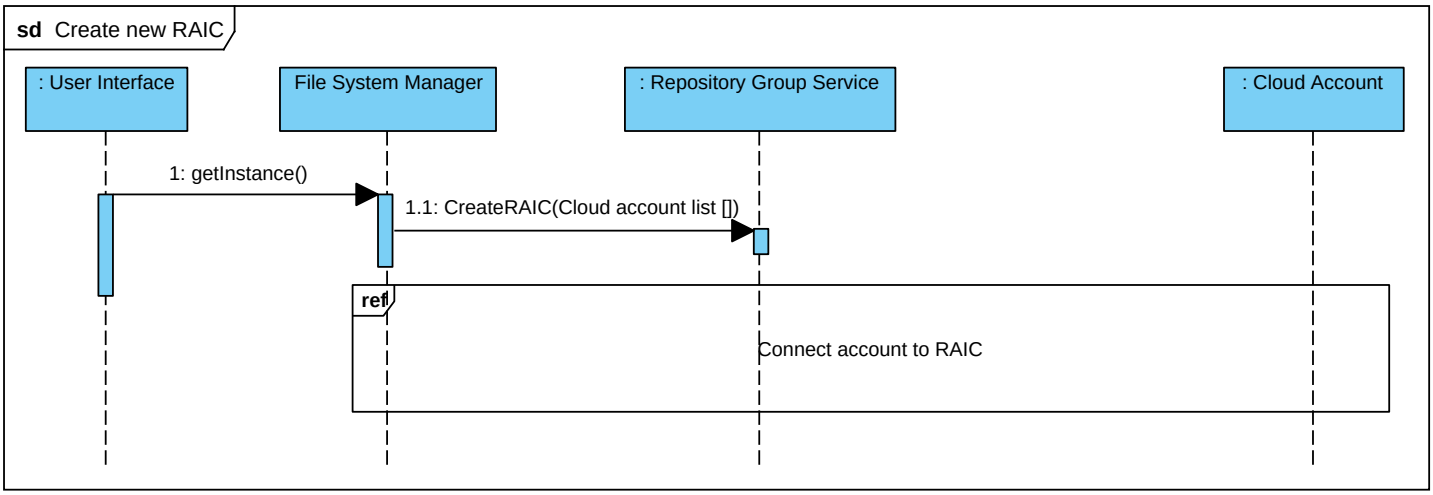

sd Diconnect account from RAIC
User
: User Interface
: Repository Group Service
: Cloud Account
1: ShowDisconnectDialog()
1.1: getInstance()
loop
[User choose to try again after error]
1.2: ShowChooseDialog()
1.3: disconnectAccountFromRAIC(Cloud account source)
alt
[RAIC locked]
1.3.1: showErrorMessage(String message)
[Disconnect impossible]
1.3.2: showErrorMessage(String message)
[else]
2: LockRAIC()
3: getFileList(String parentDirectory)
5: showCompleteMessage(String message)
4: unlockRAIC()
6: closeDisconnectDialog()
Disconnected cloud account

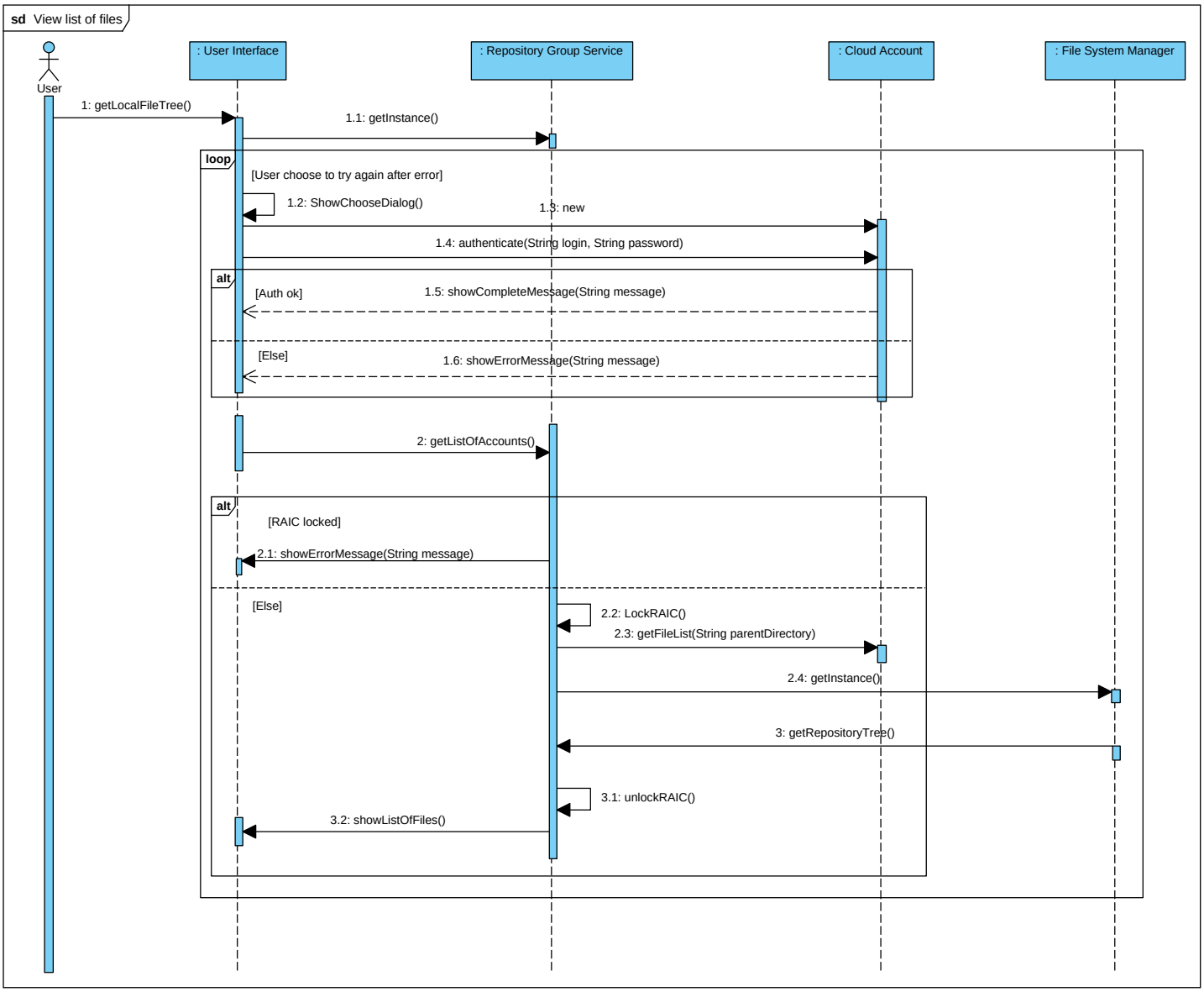

sd Migrate Raic to another accounts
User
: User Interface
: Repository Group Service
: Cloud Account
1: ShowMigrateDialog()
1.1: getInstance()
loop
[User choose to try again after error]
1.2: ShowChooseDialog()
loop
[Accounts in list left]
ref
Connect account to RAIC
alt
[RAIC locked]
2: showErrorMessage(String message)
[Migrate impossible]
3: showErrorMessage(String message)
[else]
4: LockRAIC()
5: getFileList(String parentDirectory)
6: migrateRAIC()
7: unlockRAIC()
8: showCompleteMessage(String message)
loop
[Recently added accounts left]
ref
Diconnect account from RAIC
9: closeMigrateDialog()
Disconnected cloud account

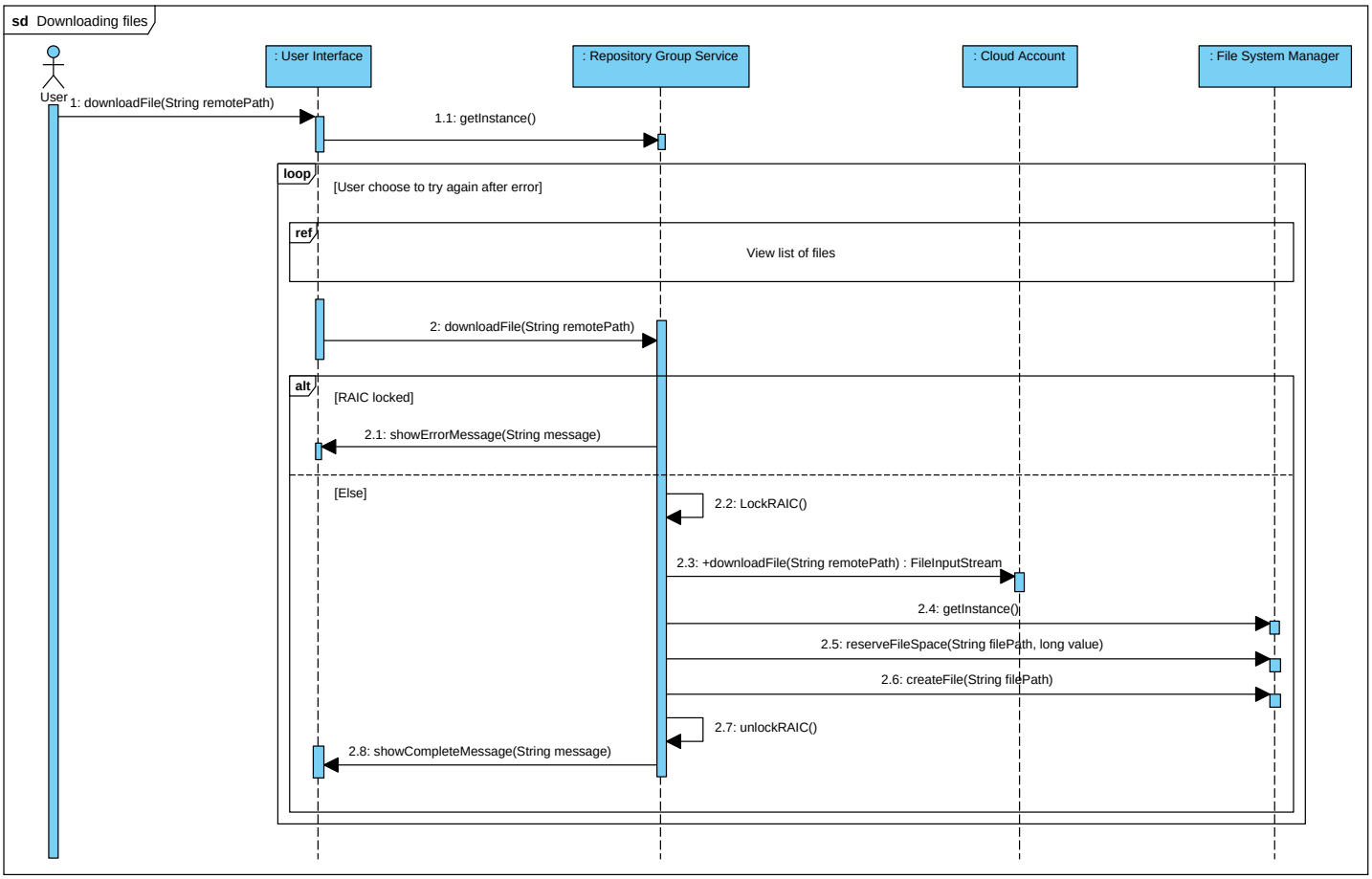

sd Lock RAIC
: Repository Group Service
: Cloud Account
1: getListOfAccounts()
loop
[Accounts left]
2: getFileList(String parentDirectory)
3: uploadFile(String remotePath, byte fileStream [])

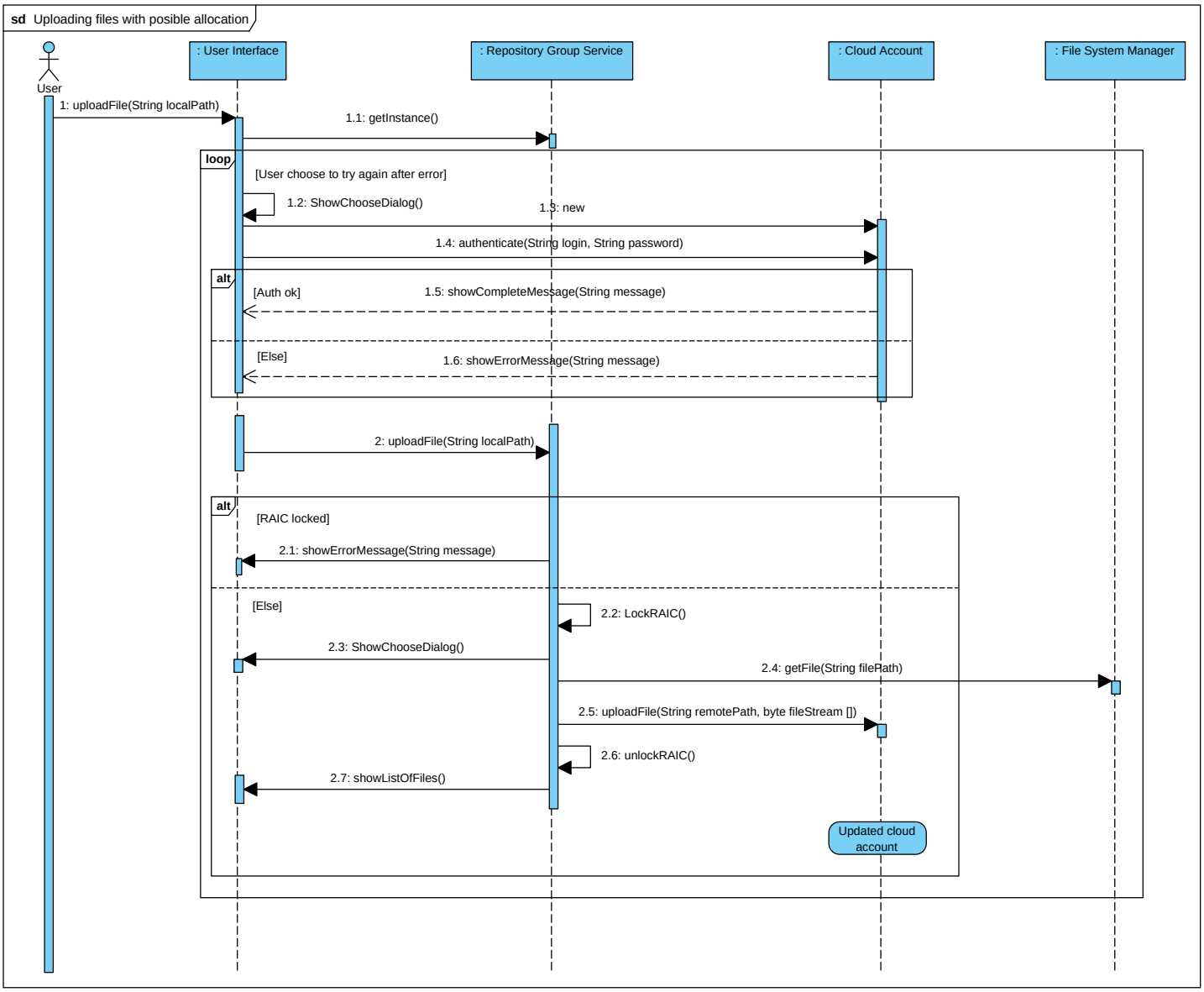

sd Unlock RAIC
: Repository Group Service
: Cloud Account
1: getListOfAccounts()
loop
[Accounts left]
2: getFileList(String parentDirectory)
3: uploadFile(String remotePath, byte fileStream [])

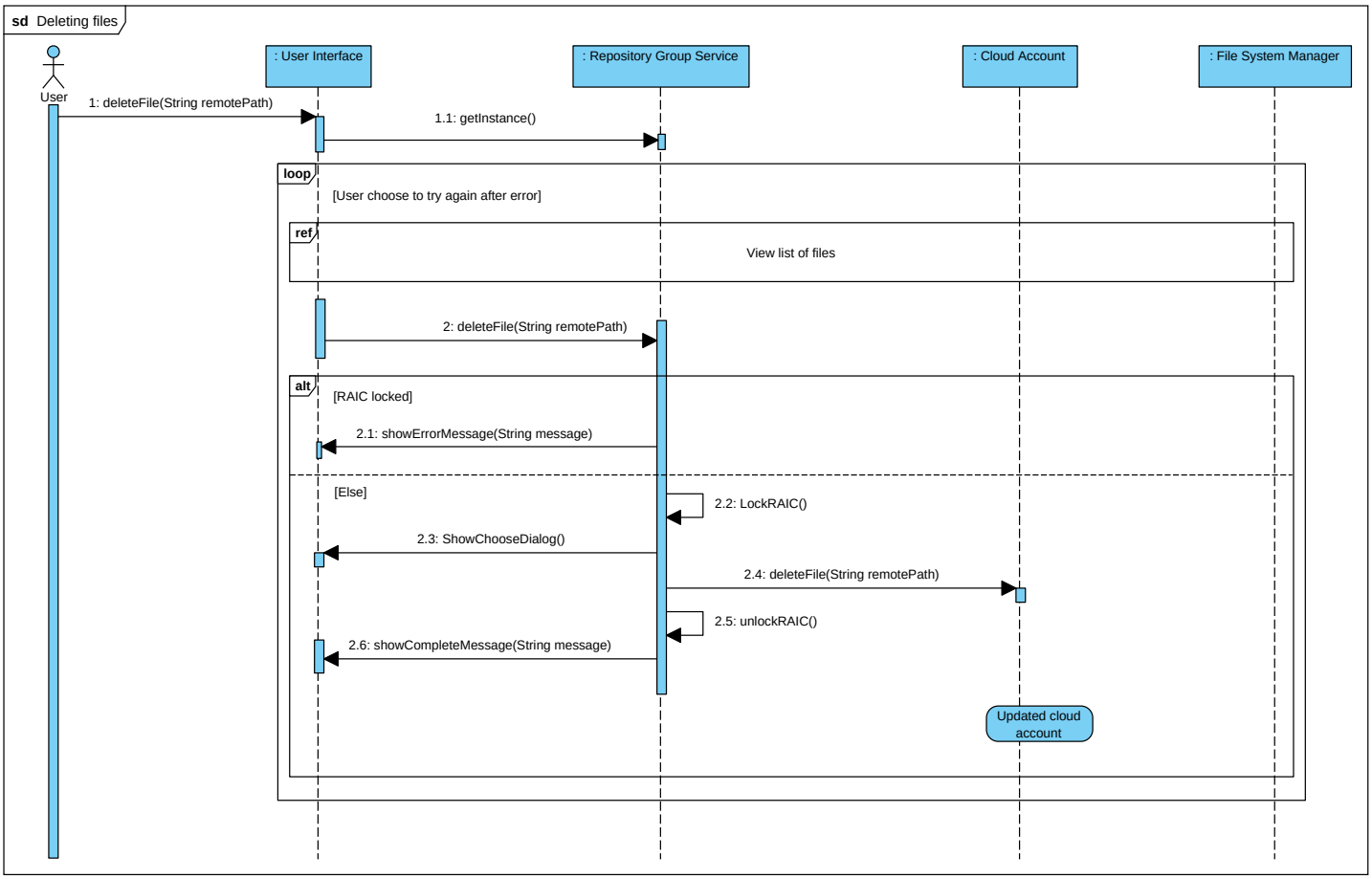

sd Connect to existing RAIC
User
: User Interface
: Repository Group Service
: Cloud Account
: File System Manager
1: ShowConnectDialog()
1.1: getInstance()
loop
[User choose to try again after error]
1.2: ShowChooseDialog()
1.3: new
1.4: authenticate(String login, String password)
alt
[Auth ok]
1.5: showCompleteMessage(String message)
[Else]
1.6: showErrorMessage(String message)
2: ConnectToExistingRAIC(Cloud Account connected)
alt
[RAIC locked]
2.1: showErrorMessage(String message)
[Else]
2.2: LockRAIC()
2.3: getFileList(String parentDirectory)
2.4: getInstance()
3: getRepositoryTree()
3.1: unlockRAIC()
3.2: showCompleteMessage(String message)
4: closeConnectDialog()

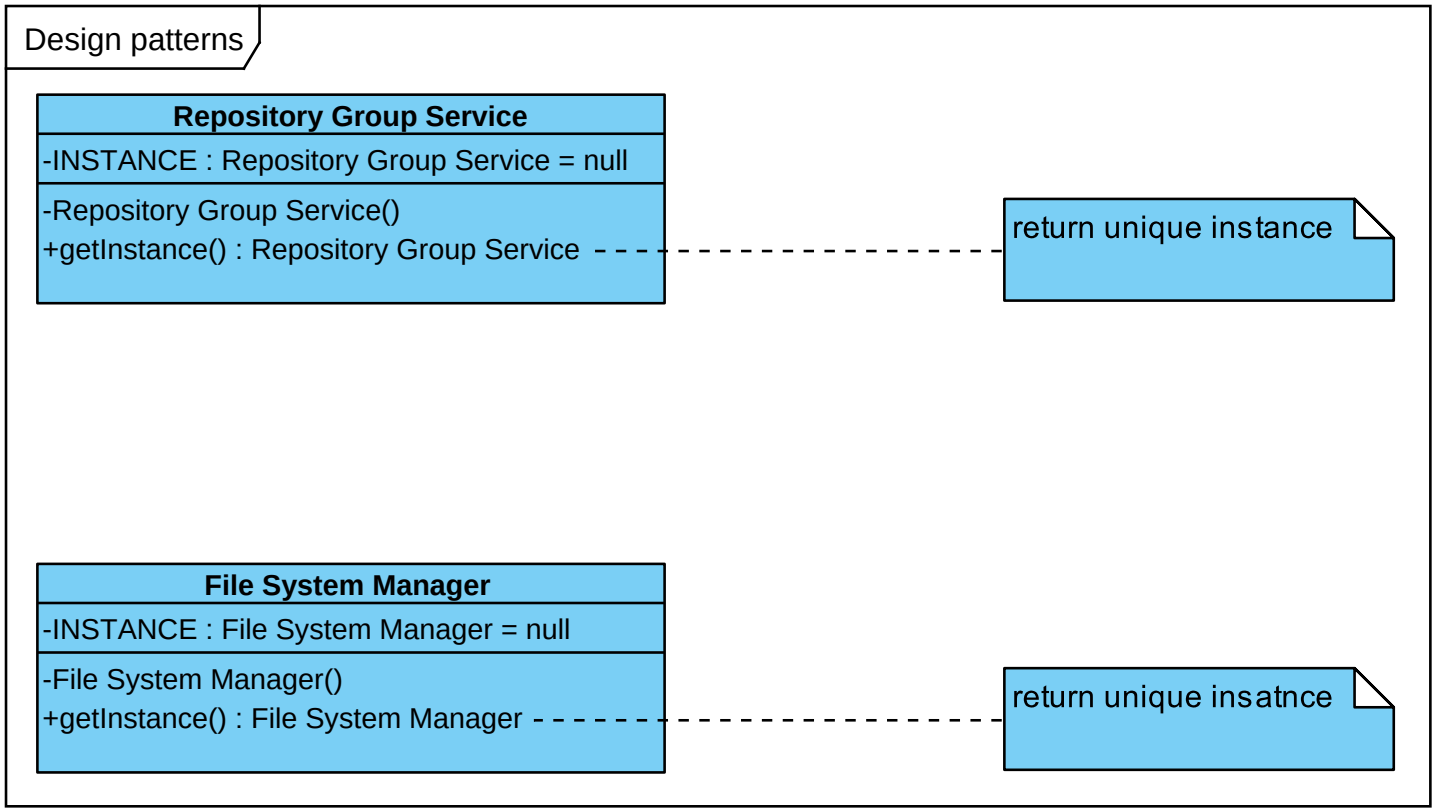

sd Create new RAIC
: User Interface
File System Manager
: Repository Group Service
: Cloud Account
1: getInstance()
1.1: CreateRAIC(Cloud account list [])
ref
Connect account to RAIC


sd View list of files
User
: User Interface
: Repository Group Service
: Cloud Account
: File System Manager
1: getLocalFileTree()
1.1: getInstance()
loop
[User choose to try again after error]
1.2: ShowChooseDialog()
1.3: new
1.4: authenticate(String login, String password)
alt
[Auth ok]
1.5: showCompleteMessage(String message)
[Else]
1.6: showErrorMessage(String message)
2: getListOfAccounts()
alt
[RAIC locked]
2.1: showErrorMessage(String message)
[Else]
2.2: LockRAIC()
2.3: getFileList(String parentDirectory)
2.4: getInstance()
3: getRepositoryTree()
3.1: unlockRAIC()
3.2: showListOfFiles()


sd Downloading files
User
: User Interface
: Repository Group Service
: Cloud Account
: File System Manager
1: downloadFile(String remotePath)
1.1: getInstance()
loop
[User choose to try again after error]
ref
View list of files
2: downloadFile(String remotePath)
alt
[RAIC locked]
2.1: showErrorMessage(String message)
[Else]
2.2: LockRAIC()
2.3: +downloadFile(String remotePath) : FileInputStream
2.4: getInstance()
2.5: reserveFileSpace(String filePath, long value)
2.6: createFile(String filePath)
2.7: unlockRAIC()
2.8: showCompleteMessage(String message)


sd Uploading files with posible allocation
User
: User Interface
: Repository Group Service
: Cloud Account
: File System Manager
1: uploadFile(String localPath)
1.1: getInstance()
loop
[User choose to try again after error]
1.2: ShowChooseDialog()
1.3: new
1.4: authenticate(String login, String password)
alt
[Auth ok]
1.5: showCompleteMessage(String message)
[Else]
1.6: showErrorMessage(String message)
2: uploadFile(String localPath)
alt
[RAIC locked]
2.1: showErrorMessage(String message)
[Else]
2.2: LockRAIC()
2.3: ShowChooseDialog()
2.4: getFile(String filePath)
2.5: uploadFile(String remotePath, byte fileStream [])
2.6: unlockRAIC()
2.7: showListOfFiles()
Updated cloud account


sd Deleting files
User
: User Interface
: Repository Group Service
: Cloud Account
: File System Manager
1: deleteFile(String remotePath)
1.1: getInstance()
loop
[User choose to try again after error]
ref
View list of files
2: deleteFile(String remotePath)
alt
[RAIC locked]
2.1: showErrorMessage(String message)
[Else]
2.2: LockRAIC()
2.3: ShowChooseDialog()
2.4: deleteFile(String remotePath)
2.5: unlockRAIC()
2.6: showCompleteMessage(String message)
Updated cloud account


Design patterns
Repository Group Service
-INSTANCE : Repository Group Service = null
-Repository Group Service()
+getInstance() : Repository Group Service
return unique instance
File System Manager
-INSTANCE : File System Manager = null
-File System Manager()
+getInstance() : File System Manager
return unique insatnce