D1.1a - VISHNU General specifications

COLLABORATORS

	TITLE : D1.1a - VISHNU General specifications		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi, Kevin Coulomb, and Ibrahima Cissé	January 26, 2012	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
01	07/12/2010	Formatting example	SysFera
02	13/12/2010	Pre-deliverable	SysFera
03	13/01/2011	First Release	SysFera
04	10/02/2011	Modified 1.2, 1.4, U1, U1.1, U1.3.3, U1.3.4, U1.3.5, U1.4, U1.5, U2, U3, U4, U4.1, U4.2, UA1, UA1.1, UA6.2, UA7, T1.2, T2.6. Added UA1.3 and modified 2.3.	SysFera
05	31/03/2011	Modified 4.2.6, UC IA2. Fix set threshold available in branch c, the threshold is the cpuload and not the number of processes	SysFera
06	21/04/2011	Removed TA1. Added T1.7. Replaced T1.2 by T1.7.	SysFera
07	16/12/2011	Added use of .netrc file in use case U1.1	SysFera
08	18/01/2012	Added UC T1.2 and UC T2.8	SysFera

	REVISION HISTORY		
NUMBER	DATE	DESCRIPTION	NAME
09	26/01/2012	Modified U1.1, U1.5 and UA5.2. Added U5, U6, U6.1, U6.2, U6.3, UA8, UA8.1, UA8.2.	SysFera

Contents

List of Figures

Chapter 1

Document presentation

1.1   Document objectives

This document presents the external specifications of the Vishnu system at a general level. At this level, we describe the interaction of a user with the system without providing implementation details. The different steps that constitute one scenario are detailed as well as the content of the messages exchanged. The main objective is to describe the system from the user point of view.

These general specifications are a prerequisite for the detailed specifications step in the software development process.

1.2   Document structure

The document is divided into 4 parts corresponding to the 4 modules that compose the Vishnu system:

- UMS: Users Management Service
- TMS: Tasks Management Service
- FMS: Files Management Service
- IMS: Information Management Service

Each module corresponds to a chapter in the document, and each chapter contains three sections:

- A first section containing "Use case descriptions" that follow the standard UML description of a use case. Each use case is identified by a code beginning with a letter corresponding to the VISHNU Module that realizes the use case (U for UMS, T for TMS, F for FMS, I for IMS).
- A second section containing the "Use case diagrams" that describe the organization of the different use cases (see §1.3 for a more precise definition of use cases).
- A third section containing the "Data dictionary" that contains the definitions of the words or expressions used in the use cases of the module.

1.3   Use cases format description

The specifications of VISHNU functionalities are formatted as "use case descriptions" and "use case diagrams" following the UML standard. Each use case is the description of a scenario that a VISHNU user will follow when using the system, and it details each step of the user/system interaction. The use cases do not describe the internal steps that the user should not be aware of, and that may be specific to the implementation of the use case.

Each **use case description** can contain the following elements:

- **Title**: Contains the use case code and name.

- **Summary**: Describes the main objective of the use case.

- **Actors**: Contains respectively "User", "Admin" or both when the use case applies respectively to a standard user, an administrator or both classes of users.

- **Precondition**: Contains the conditions that should be realized before the use case runs.

- **Postcondition**: Contains the conditions that should be realized when the basic sequences or the branch sequences are finished.

- **Base sequence**: Contains the basic sequence in chronological order between the Actor and the System. Each step of the sequence is identified by a number.

- **Branch sequence**: Contains the branch(es) of the base sequence. The branch step is identified by a number, a letter and eventually a number: the first number identies the branching point in the base sequence, the letter identifies the branch itself and the last number is used for the different steps within the branch. The postconditions should be realized after the branch is finished.

- **Exception sequence**: Contains the errors that may happen during the base sequence or the branch sequence, as considered by users. The exceptions are identified by a base or branch step identifier (see above) followed by a letter that identifies the exception. The postconditions cannot be realized after the exception is thrown.

- **Extension of**: Contains list of use cases that the current use case "extends" (see below for a definition of this relationship).

- **Extensions**: Contains list of use cases that "extend" the current use case (see below for a definition of this relationship).

- **Notes**: All things for understanding the use case can be recorded here.

Each **use case diagram** show the relationships between Actors and use cases, and between use cases. All these relationships are defined in the UML standard and have the following meaning:

- **Generalization** (shown as solid arrow with a large triangle head): the source use case inherits properties and behavior of the parent (target of the arrow) use case and may override the behavior of the parent. The preconditions, postconditions and exceptions of the parent apply to the child use case. A use case that has child (specializations) of itself is shown in the use case diagram in **orange** color.

- **Extension** (shown as dashed arrow with "extend" stereotype): this is a directed relationship that specifies how and when the behavior defined in usually supplementary (optional) extending use case can be inserted into the behavior defined in the extended use case. Extending use case typically defines optional behavior that is not necessarily meaningful by itself. The extension takes place at one or more extension points defined in the extended use case by the "Include" keyword.

- **Inclusion** (shown as dashed arrow with "include" stereotype): this is a directed relationship between two use cases when required, not optional behavior of the included use case is inserted into the behavior of the including (base) use case. Including use case depends on the addition of the included use case. Including use cases are usually not complete by themselves and require the included use cases. A use case that has inclusions is shown in the use case diagram in **blue** color.

1.4   References

None

1.5   Glossary

- **Batch scheduler**: system used to manage batch jobs on multi-processor systems or clusters.

- **Client system**: computer system used to send requests to the VISHNU system either using the command-line interface or any API.

- **Exception**: event happening during the execution of a use case scenario and that triggers a specific action from the System. This action can either be returning an error message to the user (if the used interface is interactive) or triggering a programing language exception if the used interface is programmatic.

- **Sysfera-DS**: open-source middleware software used by Vishnu (former name "DIET")

- **UML**: Unified Modeling Language (current version: v2.3)

- **VISHNU system**: set of all elements (software, hardware, data) that compose a single instance of the VISHNU application and that work together to provide VISHNU services to the users.

Chapter 2

Use cases for User Management Service (UMS)

2.1 Use case diagrams

2.1.1   UC UMS User Manual

This UseCase Diagram describes all cases that can occur when the user opens a session with manual closure
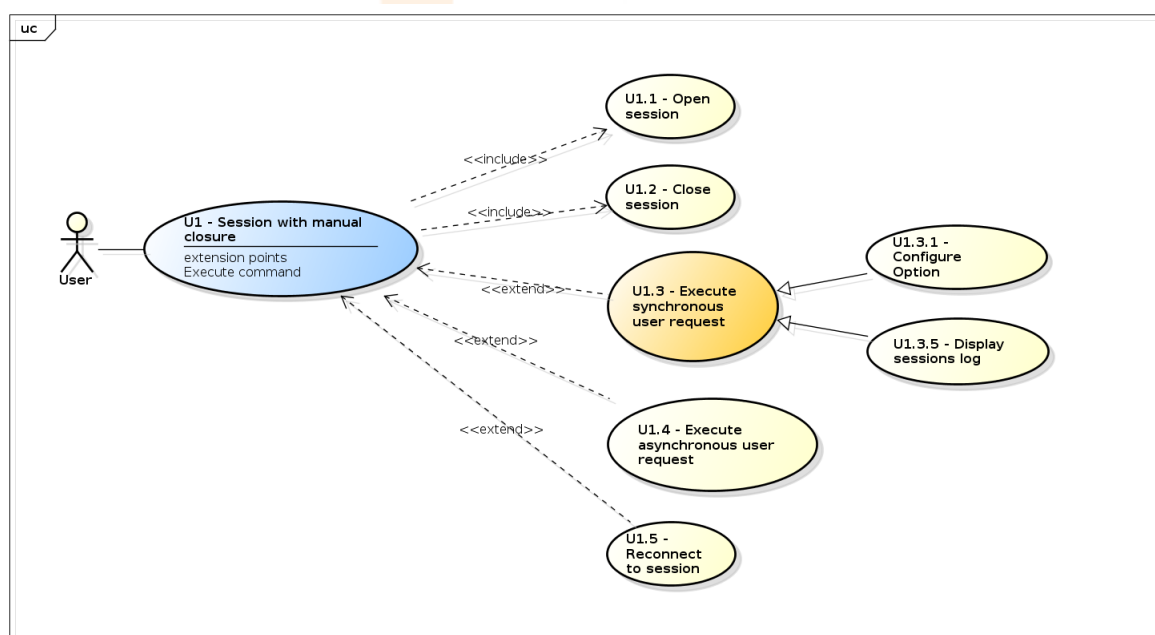
Figure 2.1: UC UMS User Manual

2.1.2   UC UMS User Auto

This UseCase Diagram describes all cases that can occur when a user opens a session with automatic closure (on disconnect and on timeout)
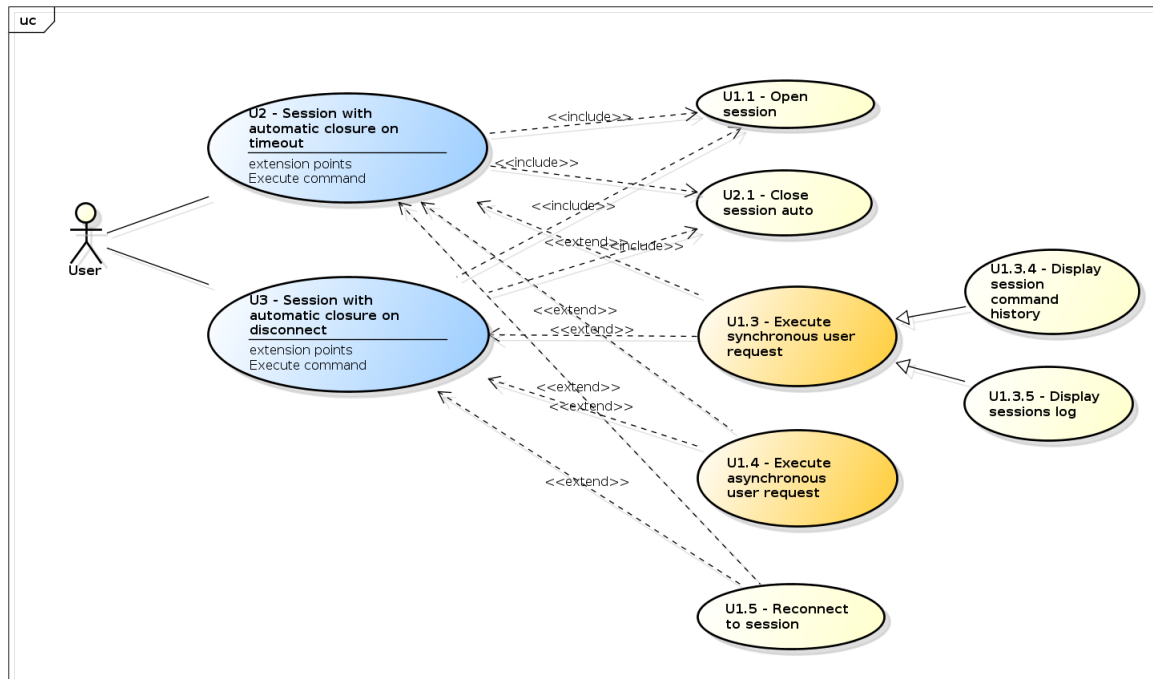
Figure 2.2: UC UMS User Auto

2.1.3 UC UMS User account

This UseCase Diagram describes all cases that can occur when a user executes synchronous requests
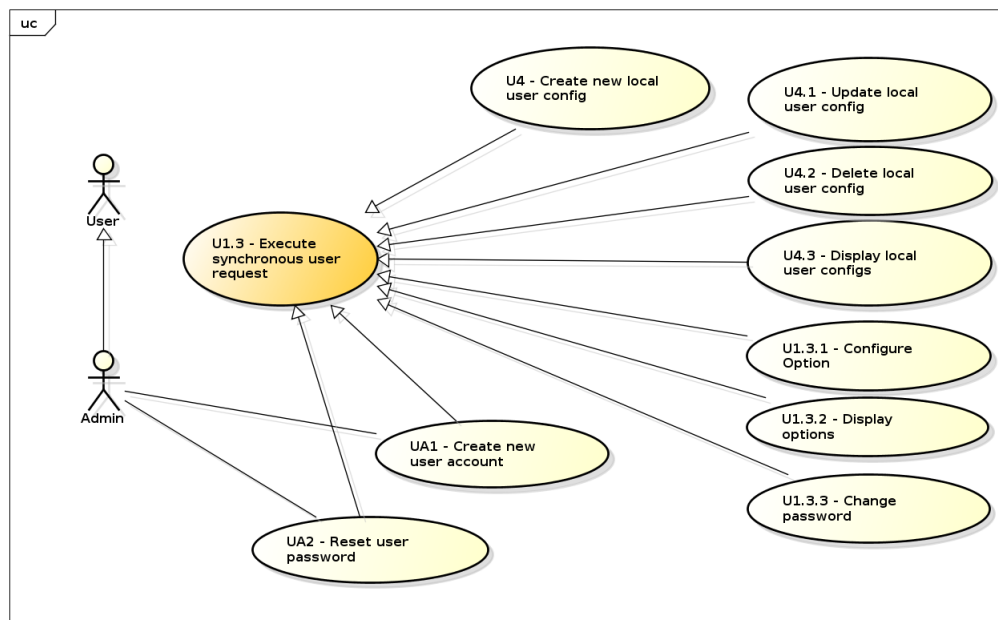
Figure 2.3: UC UMS User account

2.1.4 UC UMS Admin

This UseCase Diagram describes all administrator's functions
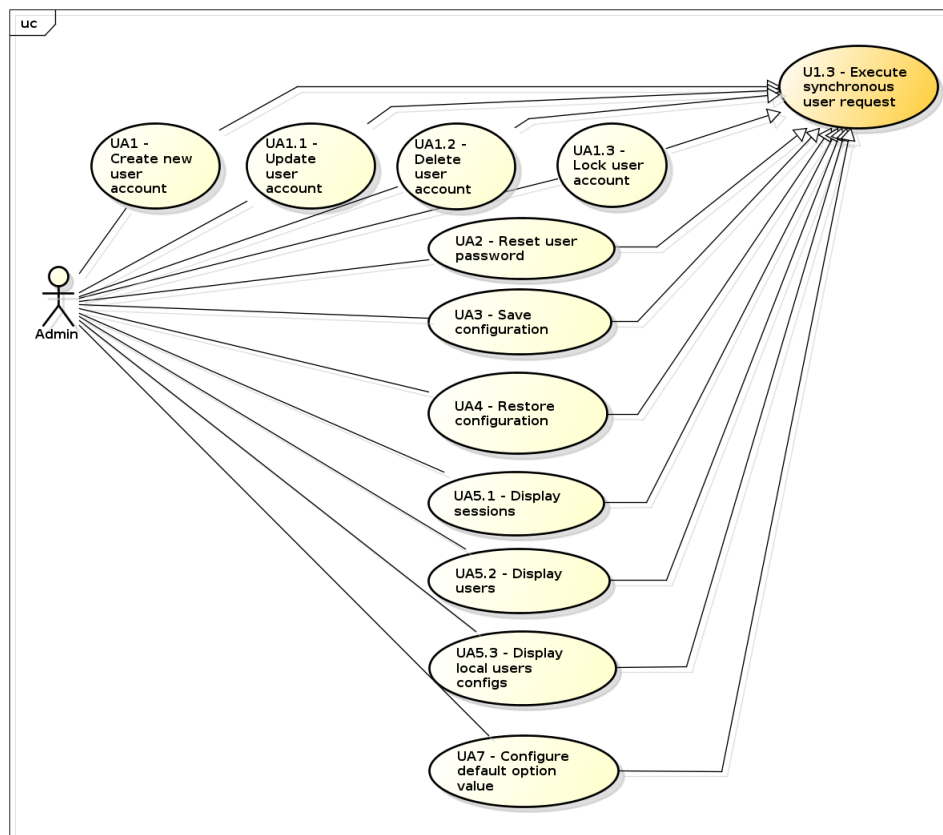
Figure 2.4: UC UMS Admin

2.1.5 UC UMS Admin Machines

This UseCase Diagram describes all cases that can occur when an administrator wants to administrate a machine
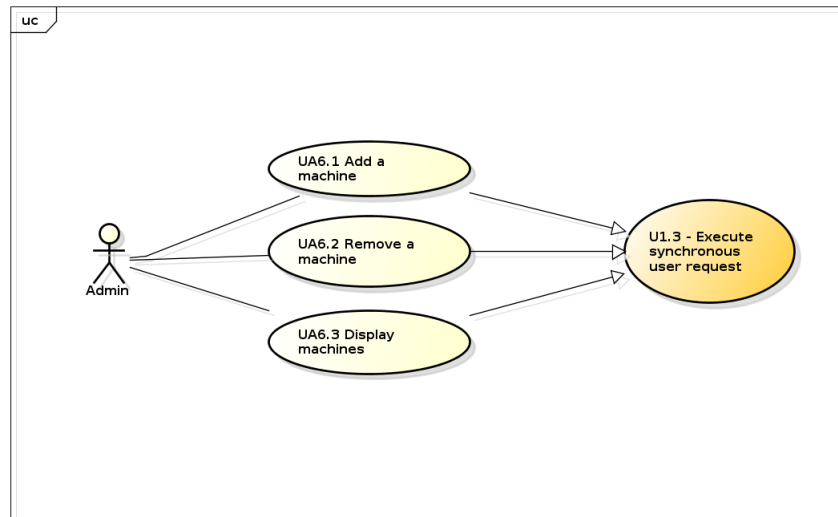
Figure 2.5: UC UMS Admin Machines

2.1.6   UC UMS Admin Authentication Systems

This UseCase Diagram describes all cases that can occur when an administrator wants to administrate a user authentification system
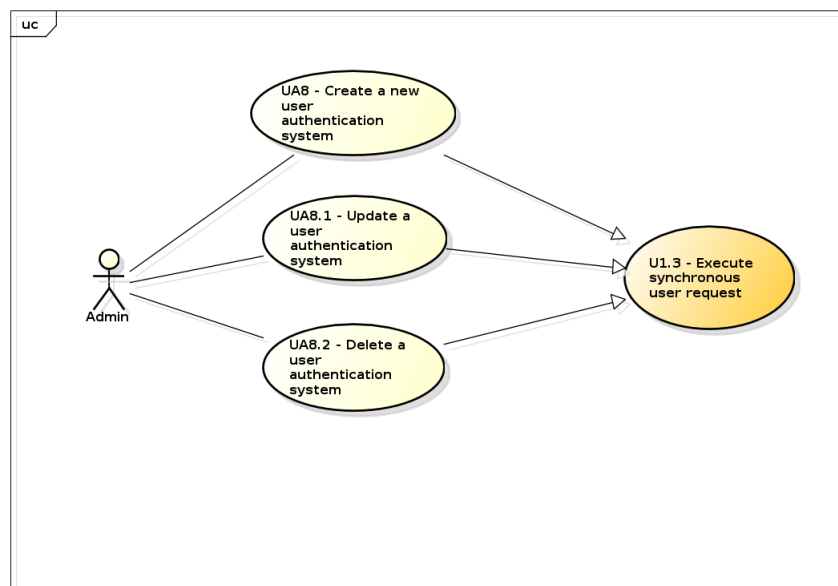
Figure 2.6: UC UMS Admin Authentication Systems

2.1.7   UC UMS User Authentication Systems

This UseCase Diagram describes all cases that can occur when a user wants to manage an account associated to an authentification system
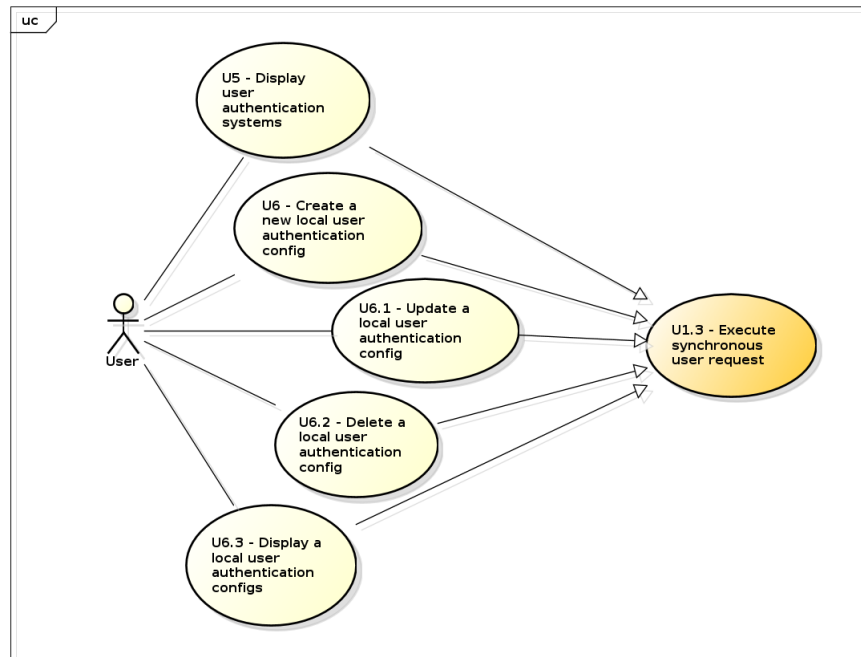
Figure 2.7: UC UMS User Authentication Systems

2.2 Use case descriptions

2.2.1 U1 - Session with manual closure

Title	U1 - Session with manual closure
Summary	The user opens a new session and closes it manually by using the command for closure
Actors	User
Precondition	- The user is authenticated - VISHNU is installed and running on the client System
Postcondition	- The session is closed - A session log has been created - All user requests submitted within the session are completed
Base sequence	1. Include::U1.1 Open session 2. System is ready to process user commands 3. Include::U1.2 Close session (before the maximum inactivity delay if option "Session closure policy" value is "Close on timeout")
Branch sequence	2a. U1.3 Execute synchronous user request 2b. U1.4 Execute asynchronous user request 2c. U1.5 Reconnect to session
Exception sequence	1a. Include::U1.1 exceptions 3a. If session cannot be closed due to running commands, user must wait until all commands are completed before trying step 3 again
Extensions	U1.3 - Execute synchronous user request U1.5 - Reconnect to session U1.4 - Execute asynchronous user request

2.2.2 U1.1 - Open session

Title	U1.1 - Open session
Summary	The user opens a new session
Actors	User
Precondition	- The user is connected on a client System in which VISHNU is installed and which can be connected to the VISHNU infrastructure
Postcondition	- A session is active
Base sequence	1. User provides login, password, and optionally the way for closing the session automatically (Session closure policy). 2. System validates login, password 3. System creates the session and activates it 4. System provides the session key to the user
Branch sequence	1a. An administrator can provide the login of another user so that the session is opened exactly as if she was that user, except the provided password must be the one of the administrator. The only difference with a session opened by the real user will be visible in the value of the session id, that will contain the administrator's login. 1b. When the login and the password provided are empty, the System will automatically read a vishnu user connection 's parameters (login and password) on the .netrc file located in the home of the user. A VISHNU account is defined on the .netrc with the value vishnu for key machine. If several VISHNU accounts have been defined in the .netrc file, each will be used in the order they appear in the file until one is allowed to connect. 2a. If the password is a temporary password (after reset by the administrator) the System asks the user to enter a new password, then asks for a confirmation, and registers the new password if both steps are ok. If the User request is done through the API (non-interactive) then this is an exception (a change password request is required).
Exception sequence	1a. The .netrc is not found. 1b. The permission of the .netrc file should be 600 1c. The machine vishnu is not found 1d. The login is undefined for the machine vishnu (the password must follow the login). 1e. The password is undefined. 2a. The user login is unknown 2b. The user password is invalid 2c. The value for the "Session closure policy" option is invalid 2d. VISHNU infrastructure is unreachable or unavailable 2e. The user password is temporary and the request is non-interactive 2f. If the user is using the command-line interface and is already connected to a session in the same terminal then this session will remain active but will not be used for the user requests until the new session is closed. 2f. The substitute login provides by the administrator is unknown
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

2.2.3   U1.2 - Close session

Title	U1.2 - Close session
Summary	The user closes the session manually
Actors	User
Precondition	- The user is connected on the client System - The user has an open session on the client System
Postcondition	- The session is closed - A session log has been created - All user requests submitted during the session are completed
Base sequence	1. The user sends a request to close a session (the session key registered in the user's environment is sent to the System) 2. The System checks that the session key is valid and the corresponding session is open 3. The System checks that there are no running commands within the session 4. The System closes the session 5. The System informs the user that the session has been closed
Branch sequence	
Exception sequence	1a. VISHNU infrastructure is unreachable or unavailable 2a. The session key is invalid 2b. The session is already closed 2c. The session key is incompatible with the authenticated user (that means that the session identifier is not for the user who sends the requests). 3a. If there are running commands within the session, the System informs the user that the session cannot be closed
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

2.2.4   U1.3 - Execute synchronous user request

Title	U1.3 - Execute synchronous user request
Summary	The user submits a synchronous request to the System
Actors	User
Precondition	- The user is connected on the client System - The user has an open session on the client System
Postcondition	- The request is completed - A request log is created
Base sequence	1. The user sends the request to the System 2. The System returns the results to the user
Branch sequence	
Exception sequence	1.a Invalid session (bad session key or unavailable session) 1.b Invalid request 1.c Permission denied (admin request issued by normal user) 1.d Ressource not available 1.e VISHNU System crashed
Extension of	U1 - Session with manual closure U3 - Session with automatic closure on disconnect U2 - Session with automatic closure on timeout

2.2.5 U1.3.1 - Configure Option

Title	U1.3.1 - Configure Option
Summary	The user wants to modify the value of an option attached to his/her VISHNU account
Actors	User
Precondition	
Postcondition	The option's value is modified
Base sequence	1. The user sends a request for modifying the value of an option to the System 2. The System checks the option name and registers the new value for the option 3. The System returns an acknowledgment to the user
Branch sequence	
Exception sequence	2a. Invalid option name 2b. Invalid option value

2.2.6 U1.3.2 - Display options

Title	U1.3.2 - Display options
Summary	The user displays options concerning his/her VISHNU account
Actors	User
Precondition	
Postcondition	
Base sequence	1. The user sends a request to list all of his/her options 2. The System returns all options of the user
Branch sequence	1a. The users sends a request to list a specific option identified by its name or all default options values defined by VISHNU administrator 2a. The System checks the name of the option specified 2b. The System returns the value of the option specified or all default options values defined by VISHNU administrator
Exception sequence	2a1. The option name is unknown

2.2.7 U1.3.3 - Change password

Title	U1.3.3 - Change password
Summary	The user changes her password
Actors	User
Precondition	
Postcondition	- The password is changed
Base sequence	1. The user sends a request containing her old password and the new password 2. The System checks the old user password and registers the new user password 3. The System returns an acknowledgment to the user
Branch sequence	
Exception sequence	2a. The provided old password does not match the current password 2b. The provided new password is too short or too long

2.2.8   U1.3.4 - Display session command history

Title	U1.3.4 - Display session command history
Summary	The user displays all the commands sent during one session
Actors	User
Precondition	
Postcondition	
Base sequence	1. The user sends a request to list all commands sent during the session identified by the session key registered in the user's environment 2. The System returns the list of all commands issued by the user during the session which key corresponds to the session key registered in the user's environment. Each command has exactly the same format and parameters as the original submission and can be resubmitted as-is to the System
Branch sequence	1a. The user sends a request containing a session identifier to list all commands sent during the session identified by the session id 2a. The System returns the list of all commands issued by the user during the session which id corresponds to the provided id
Exception sequence	1b. Invalid session key (unknown / belonging to another user, if the current user is not an administrator) 1a1. Invalid session id (unknown / belonging to another user, if the current user is not an administrator)

2.2.9   U1.3.5 - Display sessions log

Title	U1.3.5 - Display sessions log
Summary	The user displays her sessions (active or closed)
Actors	User
Precondition	
Postcondition	
Base sequence	1. The user sends a request to list all her active sessions that have an open timestamp within an interval provided by the user (start and finish date) 2. The System returns all active sessions of the user matching the search criteria with the following information for each session: id, opening date, client host name, closure policy (timeout or disconnect), time before automatic closure (if applicable) and period using start and finish date
Branch sequence	1a. The user sends a request to list all closed sessions with some optional search criteria 2a. The System returns all closed sessions of the user matching the search criteria 1b. The user sents a request to list all sessions with some optional search criteria 2b. The System returns all sessions of the user matching the search criteria
Exception sequence	

2.2.10   U1.4 - Execute asynchronous user request

Title	U1.4 - Execute asynchronous user request
Summary	The user submits an asynchronous request to the system
Actors	User
Precondition	- The user is connected on the client System - The user has an open session on the client System
Postcondition	- The request is completed - A request log is created
Base sequence	1. The user sends the request to the system 2. The System returns an acknowledgment to the user 3. The System runs the request in background 4. When the request is completed, the system updates the status of the request
Branch sequence	
Exception sequence	1a. Invalid session (bad session certificate or session unavailable) 1b. Invalid request 1c. Permission denied 1d. Ressource not available 1e. VISHNU System crashed
Extension of	U1 - Session with manual closure U2 - Session with automatic closure on timeout U3 - Session with automatic closure on disconnect

2.2.11 U1.5 - Reconnect to session

Title	U1.5 - Reconnect to session
Summary	The user connects to a session in which she was disconnected previously without closing it
Actors	User
Precondition	- The user is connected on a client host in which VISHNU is installed and that can be connected to the VISHNU infrastructure
Postcondition	- The user is connected to an active session
Base sequence	1. User provides its login, password and the identifier of the session to the System 2. The System validates the user's login, password and the identifier of the session 3. The System provides the chosen session key to the user
Branch sequence	1a. When the login and the password provided are empty, the System will automatically read a vishnu user connection 's parameters (login and password) on the .netrc file located in the home of the user. A vishnu account is defined on the .netrc with the value vishnu for key machine. If several VISHNU accounts have been defined in the .netrc file, each will be used in the order they appear in the file until one is allowed to reconnect with the corresponding session identifier.

Exception sequence	cf U1.1 (Open session) 1a. The .netrc is not found. 1b. The permission of the .netrc file should be 600 1c. The machine vishnu os not found 1d. The login is undefined for the machine vishnu (the password must follow the login). 1e. The password is undefined. 2a. The identifier of the session does not exist 2b. The identifier relates to a session belonging to another user 2c. The identifier is for a session closed 2d. If the user is using the command-line interface and is already connected to a session in the same terminal then this session will remain active but will not be used for the user requests until the new session is closed.
Extension of	U1 - Session with manual closure U2 - Session with automatic closure on timeout U3 - Session with automatic closure on disconnect

2.2.12 U2 - Session with automatic closure on timeout

Title	U2 - Session with automatic closure on timeout
Summary	The user opens a new session that will be closed by the System after the expiration of the inactivity delay
Actors	User
Precondition	- VISHNU is installed and running on the client system - The client system can be connected to VISHNU - The option "Session closure policy" value is "Close on Timeout"
Postcondition	- A session log has been created - The session is closed - All user requests submitted during the session are completed
Base sequence	1. U1.1 Open session 2. The System is ready to process user commands 3. After inactivity delay has expired: U2.1 Close session auto
Branch sequence	2a. U1.3 Execute synchronous user request 2b. U1.4 Execute asynchronous user request 2c. U1.5 Reconnect to session 2d. If the user disconnects from the client terminal or the client system crashes or is shutdown, the session remains open and all asynchronous commands that were not completed are kept running
Exception sequence	see U1
Extensions	U1.5 - Reconnect to session U1.4 - Execute asynchronous user request U1.3 - Execute synchronous user request

2.2.13 U2.1 - Close session auto

Title	U2.1 - Close session auto
Summary	The session is closed by the system
Actors	

Precondition	- The user is connected on the client system - The user has an open session on the client system either the inactivity timeout for the session has expired or the user disconnected from its shell session
Postcondition	- The session is closed - The session close event is stored in the system's log
Base sequence	1. The system checks if the user has got no running commands (file transfers or tasks) 2. The system registers session closure
Branch sequence	1a. If the user has got some running commands, the system does not close the session and reset the inactivity timeout. After this delay is expired, back to step 1.
Exception sequence	
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

2.2.14   U3 - Session with automatic closure on disconnect

Title	U3 - Session with automatic closure on disconnect
Summary	The user opens a new session that will be closed by the system after the user disconnects from the client terminal
Actors	User
Precondition	- VISHNU is installed and running on the client system - The client system can be connected to VISHNU - The option "Session Closure Policy" value is "Close on Disconnect"
Postcondition	- A session log has been created - The session state is closed - All user requests submitted during the session are complete
Base sequence	1. U1.1 Open session 2. System is ready to process user commands 3. The user disconnects from the terminal (either by typing an exit command, by closing the window, or by remote disconnection) 4. U2.1 Close session auto
Branch sequence	2a. U1.4 Execute synchronous user request 2b. U1.5 Execute asynchronous user request 3a. if the client system crashes or is shutdown, go to step 4
Exception sequence	see U1
Extensions	U1.3 - Execute synchronous user request U1.5 - Reconnect to session U1.4 - Execute asynchronous user request

2.2.15   U4 - Create new local user config

Title	U4 - Create new local user config
Summary	The user creates a new local user configuration for a given user on a given machine
Actors	User
Precondition	- The user has an account on VISHNU
Postcondition	- Local user config is registered - An email is sent to the user with a message containing an SSH public key

Base sequence	1. The user provides local user configuration information for a given machine (login of the local unix account on this machine, path to home directory of that unix account on this machine) 2. The System checks the user login and the machine Id 3. The System generates an ssh private/public key pair 4. The System sends an email to the user containing the public key and asking the user to add this key to the authorized_keys on the machine 5. The user updates his/her authorized_keys file on the machine by adding the public key
Branch sequence	
Exception sequence	2a. Unknown login 2b. Unknown machine 2c. The user already has a local user configuration defined for the machine 4a. Invalid email address
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

2.2.16 U4.1 - Update local user config

Title	U4.1 - Update local user config
Summary	The user updates her local user configuration for a given machine
Actors	User
Precondition	- The user has an account on VISHNU - The user has a local user configuration defined for the machine
Postcondition	- The local user configuration is updated
Base sequence	1. The user provides the identifier of the machine and information to be updated (login of the local unix account on this machine, path to home directory of that unix account on this machine) 2. The System checks the local user configuration 3. The System updates the local user configuration information 4. The System returns an acknowledgment to the user
Branch sequence	
Exception sequence	2a. Unknown login for the given machine 2b. Unknown machine for the given login 2c. No existing local configuration

2.2.17 U4.2 - Delete local user config

Title	U4.2 - Delete local user config
Summary	The user deletes his/her local user configuration on a given machine
Actors	User
Precondition	- The local user configuration exists for the given machine - There is no job or file transfer running on the given machine
Postcondition	- The local user configuration for the given machine is deleted

Base sequence	1. The user provides the identifier machine of the local user configuration and his/her login 2. The System checks the identifier of the machine for the given user 3. The System deletes the local user configuration identified 4. The System returns an acknowledgment to the user
Branch sequence	
Exception sequence	2a. Unknown login for the given machine 2b. Unknown machine for the given login 2c. No existing local configuration

2.2.18   U4.3 - Display local user configs

Title	U4.3 - Display local user configs
Summary	The user displays all of his/her local configurations
Actors	User
Precondition	
Postcondition	
Base sequence	1. The user sends a request to list all his/her local configurations 2. The System returns all local configurations
Branch sequence	1a. The user sends a request containing the identifier of a machine for listing a specific local user configurations on a specific machine
Exception sequence	1a1. Unknown machine

2.2.19   U5 - Display user authentication systems

Title	U5 - Display user authentication systems
Summary	The user displays all user-authentication systems defined in VISHNU
Actors	User
Precondition	
Postcondition	
Base sequence	1. The user sends a request to list all user-authentication system in which she/he has a local user-authentification configuration 2. The System returns all corresponding user-authentication systems
Branch sequence	1a. The user sends a request to list all user-authentication systems defined in VISHNU 1b. The administrator sends a request to list all user-authentication system in which a specific user has a local user-authentification config
Exception sequence	

2.2.20   U6 - Create a new local user authentication config

Title	U6 - Create a new local user authentication config
Summary	The user creates a new local user-authentication configuration for a given user on a given user-authentification system

Actors	User
Precondition	The user has an account on VISHNU The user has not a local user-authentication config on the given user-authentification system
Postcondition	- Local user-authentication config is registered
Base sequence	1. The user provides the local user-authentication configuration information for a given user-authentication system (login on the local account on this user-authentication system) and the user-authentication system identifier 2. The System checks the user login (associated to the session key) and the user-authentication system identifier 3. The System creates the corresponding local user-authentication user configuration 4. The System returns an acknowledgment to the user
Branch sequence	1a. An administrator provides the local user-authentication configuration information for a given user on a given authentication system 2a. The System checks the user login provided by the administrator and the authentication system identifier
Exception sequence	2a. Unknown user 2b. Invalid authentification system identifier

2.2.21   U6.1 - Update a local user authentication config

Title	U6.1 - Update a local user authentication config
Summary	The user updates a local user-authentication configuration in VISHNU (database)
Actors	User
Precondition	- The user has a VISHNU account - The local user-authentication configuration is defined on VISHNU for the given user
Postcondition	- The local user-authentication configuration is updated
Base sequence	1. The user provides the identifier of the authentication system and information to be updated (login of the local account on this authentication system) 2. The System checks the local user-authentication configuration 3. The System updates the local user-authentication configuration information 4. The System returns an acknowledgment to the user
Branch sequence	1a. An administrator provides the identifier of the authentication system and information to be updated (login of the local account on this authentication system) for a given user
Exception sequence	2a. Unknown user 2b. Invalid authentication system identifier 2c. Unknown local user-authentication config

2.2.22   U6.2 - Delete a local user authentication config

Title	U6.2 - Delete a local user authentication config
Summary	The user deletes a local user-authentication configuration
Actors	User

Precondition	- The user has a VISHNU account - The local user-authentication configuration is defined on VISHNU for the given user
Postcondition	- The local user-authentication configuration is no longer in the system
Base sequence	1. The user provides the identifier of the authentication system 2. The System checks the local user-authentication configuration 3. The System deletes the local user-authentication configuration information 4. The System returns an acknowledgment to the user
Branch sequence	1a. An administrator provides the identifier of the authentication system and the login of a specific user
Exception sequence	2a. Unknown user 2b. Invalid authentication system identifier 2c. Unknown local user-authentication config

2.2.23 U6.3 - Display local user authentication configs

Title	U6.3 - Display local user authentication configs
Summary	The user displays her/his local user-authentication configs
Actors	User
Precondition	
Postcondition	
Base sequence	1. The user sends a request to list all of her/his local user-authentication configurations 2. The System returns all local user-authentication systems for a given user
Branch sequence	1a. An administrator sends a request to list all local user-authentication configurations of a specific user 1b. An administrator sends a request to list all local user-authentication configurations of a specific user-authentication system 1c. An administrator sends a request to list all local user-authentication configurations defined in VISHNU
Exception sequence	

2.2.24 UA1 - Create new user account

Title	UA1 - Create new user account
Summary	The administrator creates a new user account in the System (database)
Actors	Admin
Precondition	- The user does not have an account on VISHNU
Postcondition	- The user account is created in an active state - The account's password must be changed at the first connection

Base sequence	1. The administrator provides the new user's last name, first name, email address and specifies whether the user has standard or admin rights 2. The System creates the new user account and initializes the password with a randomly-generated string (temporary password) 3. The System sends an email to the user containing the temporary password and the user login 4. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	3a. Invalid email address
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

2.2.25 UA1.1 - Update user account

Title	UA1.1 - Update user account
Summary	The administrator updates the user account (database)
Actors	Admin
Precondition	- The user has an account on VISHNU
Postcondition	- The user account is updated
Base sequence	1. The administrator provides the user's information changes (firstname, lastname, email, privilege) 2. The System updates the user account information 3. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1.a Invalid login or login unknown 1.b The provided parameters are invalid
Notes	The user identifier is an information that cannot be modified due to integrity constraints.

2.2.26 UA1.2 - Delete user account

Title	UA1.2 - Delete user account
Summary	The administrator deletes a user account
Actors	Admin
Precondition	- The user has an account on VISHNU - There is no job or file transfer running for the user
Postcondition	- The user account is no longer in the System - System does not contain any information related to the user
Base sequence	1. The administrator provides a user's login 2. The System deletes the user account along with all the information (configuration, history) related to it. 3. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. Invalid login (unknown or inactive)

2.2.27 UA1.3 - Lock user account

Title	UA1.3 - Lock user account
Summary	The administrator locks a user account to remove access to the account without deleting all account information
Actors	Admin
Precondition	- The user has an account on VISHNU
Postcondition	- The user account is locked
Base sequence	1. The administrator provides a user's login 2. The System changes the status of the user's account to a status where the user cannot connect or send any request to VISHNU. The requests sent before this change and that are still running will not be cancelled. 3. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. Invalid login (unknown or inactive) 2a. User status is already set to locked

2.2.28 UA2 - Reset user password

Title	UA2 - Reset user password
Summary	The administrator resets a user password
Actors	Admin
Precondition	
Postcondition	- The password of the user is temporary and must be changed at the first connection by the user
Base sequence	1. The administrator provides a user's login 2. The System resets the user's password using a randomly-generated string 3. The System sends an email to the user containing the new temporary password 4. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. Invalid login (unknown or inactive) 3a. Invalid email address
Notes	If the user has one or several active sessions when Admin requests the password reset then the sessions are not affected. Only new sessions will require the new password for authentification.

2.2.29 UA3 - Save configuration

Title	UA3 - Save configuration
Summary	The administrator saves the configuration of the system
Actors	Admin
Precondition	
Postcondition	- The configuration is saved on a file - A log information is created
Base sequence	1. The administrator requests to save the configuration in a file 2. The System creates a configuration file containing: the list of users, the list of local users configs and the list of machines according to the local users configs

Branch sequence	
Exception sequence	2a. File creation problems 2b. VISHNU System crashed

2.2.30   UA4 - Restore configuration

Title	UA4 - Restore configuration
Summary	The administrator restores a saved configuration
Actors	Admin
Precondition	- All users are disconnected from VISHNU - The configuration file was saved using the "save configuration" feature.
Postcondition	- The System is operational on all the machines that are both properly configured in the saved configuration and where the VISHNU processes are running.
Base sequence	1. The administrator opens a session as the Root user 2. The administrator checks that there is no other user/admin connected to VISHNU 3. The administrator loads the configuration file 4. The System replaces the current configuration with the loaded configuration.
Branch sequence	
Exception sequence	1a. If the Root user already has an open session, the System cannot open a second session with the Root user 3a. If the configuration file cannot be loaded, the System provides an error message. The System configuration is reset to a basic configuration with only the Root user configured.
Notes	To avoid failure during this critical operation, the reasons to go for exception 3a should be reduced as much as possible. Therefore inconsistencies between the saved configuration and the real infrastructure will not be considered as blocking for this operation.

2.2.31   UA5.1 - Display sessions

Title	UA5.1 - Display sessions
Summary	The administrator displays all past and present sessions stored in the database.
Actors	Admin
Precondition	
Postcondition	
Base sequence	1. The administrator sends a request to list all sessions (active and/or closed) that have an open timestamp within an interval provided by the user (start and finish date) 2. The System returns the list of sessions that match the search criteria and provides detailed information about these sessions (user id, open and close timestamp, client machine id)
Branch sequence	
Exception sequence	

2.2.32   UA5.2 - Display users

Title	UA5.2 - Display users
Summary	The administrator displays the description of all users registered in the database
Actors	Admin
Precondition	
Postcondition	
Base sequence	1. The administrator sends a request to list all users 2. The System returns all users with the following information for each user: id, firstname, lasname, login, status, email and password state.
Branch sequence	1a. The administrator sends a request containing the login of a specific user to list information about him/her. 1b. The administrator sends a request containing the identifier of a specific user-authentication system to list all users of this user-authentication system.
Exception sequence	1a. The login is unknown 1b. The user is not an administrator 1c. The user-authentication system is unknown

2.2.33   UA5.3 - Display local users configs

Title	UA5.3 - Display local users configs
Summary	The administrator displays the local user configurations for all users registered in the database
Actors	Admin
Precondition	
Postcondition	
Base sequence	1. The administrator sends a request to list all local users configurations 2. The System returns all the local users configs for all users
Branch sequence	1a. The administrator sends a request containing the identifier of a machine for listing all local users configurations on a specific machine 1b. The administrator sends a request containing the login of one user for listing all local users configurations of a specific user
Exception sequence	1a1. unknonwn machine 1b1. unknonwn login

2.2.34   UA6.1 Add a machine

Title	UA6.1 Add a machine
Summary	The administrator registers a new machine in VISHNU
Actors	Admin
Precondition	
Postcondition	A new machine is added in VISHNU System

Base sequence	1. The administrator adds a new machine on VISHNU by giving: - The machine name - The machine state (private or accessible to users) - The public IP adress - A structure describing the machine state - A structure describing the network 2. The machine is added on VISHNU and the System returns the machine id.
Branch sequence	
Exception sequence	1a. The machine name already exists 1b. A machine with the same public adress already exists

2.2.35 UA6.2 Remove a machine

Title	UA6.2 Remove a machine
Summary	The administrator removes a machine from the list of active machines
Actors	Admin
Precondition	- The machine is registered in the System and active
Postcondition	- The machine is inactive therefore cannot be used for any VISHNU request
Base sequence	1. The administrator provides the machine id 2. The System checks that there is no running command on the machine 3. The System set the status of the machine to inactive 4. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. The machine id is unknown 2a. Some commands are currently running on the machine 3a. The machine is already in inactive state

2.2.36 UA6.3 Display machines

Title	UA6.3 Display machines
Summary	The administrator displays the machines registered in the database
Actors	Admin
Precondition	
Postcondition	
Base sequence	1. The administrator sends a request to list all machines in the database 2. The System returns all machines in the database
Branch sequence	1a. The administrator sends a request containing the identifier of a machine to list a specific machine 1b. The administrator sends a request containing the login of a user to list the machine used by this user
Exception sequence	1a1. The machine is unknown 1b1. The login is unknown

2.2.37 UA6.4 Update machines

Title	UA6.4 Update machines
Summary	The administrator updates a machine description in the database
Actors	Admin
Precondition	
Postcondition	The database has one more description for the machine or the existing one has been replaced
Base sequence	1. The administrator sends a request to add a machine description of a machine, giving the id of the machine, its description and the language the description is in. 2. The System updates the database adding the machine description
Branch sequence	2.a The machine already has a description in this language, the description is replaced by the new one
Exception sequence	1a1. The machine is unknown

2.2.38   UA7 - Configure default option value

Title	UA7 - Configure default option value
Summary	The administrator configures the default value of an option
Actors	Admin
Precondition	
Postcondition	The default value of the option is configured and is applied to all new user requests that do not specify the value of the option. Running commands are not affected by this change.
Base sequence	1. The administrator sends a request for modifying the value of an option to the System 2. The System checks the option name and registers the new default value for the option 3. System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. VISHNU infrastructure is unreachable or unavailable 2a. Invalid option name 2b. Invalid option value

2.2.39   UA8 - Create a new user authentication system

Title	UA8 - Create a new user authentication system
Summary	The administrator creates a new user-authentication system in VISHNU (database)
Actors	Admin
Precondition	The user-authentication system has not been already defined in VISHNU
Postcondition	The user-authentication system is created
Base sequence	1. The administrator provides all the parameters required to connect to the user-authentication system: host address, connection login, connection password, type of authentication system. 2. The System creates the user-authentification system in the database and associates it to an identifier. 3. The System returns an acknowledgment to the administrator.
Branch sequence	
Exception sequence	

2.2.40 UA8.1 - Update a user authentication system

Title	UA8.1 - Update a user authentication system
Summary	The administrator updates a user-authentication system in VISHNU (database)
Actors	Admin
Precondition	The user-authentication system is defined in VISHNU
Postcondition	The authentication system is updated
Base sequence	1. The administrator provides the user-authentication system's information changes (host address, connection login, connection password, type of authentication system) 2. The system updates the user-authentication system's information 3. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. Invalid user-authentification system identifier 1b. The provided parameters are invalid

2.2.41 UA8.2 - Delete a user authentication system

Title	UA8.2 - Delete a user authentication system
Summary	The administrator deletes a user-authentication system
Actors	Admin
Precondition	The user-authentication system is defined in VISHNU
Postcondition	- The user-authentication system is no longer in the system - System does not contain any information related to the corresponding authentication system such as local user-authentication configs associated
Base sequence	1. The administrator provides the user authentification system identifier 2. The System deletes the user-authentification system along with all the information related to it. 3. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. Invalid user-authentification system identifier

2.3 Data dictionary

- **Command**: Represents all user requests sent either using the command-line interface or one of the VISHNU APIs.

- **Configuration**: The configuration contains all information about the users and machines registered in the database. It does not contain chronological information about the users or the infrastructure (logs, metrics values)

- **Inactivity delay**: The inactivity delay is the delay in seconds during which no commands are launched within one session.

- **Local user config**: The local user configuration is an object that belongs to a user account and that contains information about the unix account of the user on a specific machine (login, ssh key path, home directory).

- **Local user-authentification config**: The local user-authentification config corresponds to the mapping of an account on a user authentication system in VISHNU.

- **Manual closure**: The Manual closure means that the user uses a command for closing a session.

- **Option**: The option is a parameter of the user account that is not mandatory for user creation. A default value for each option is defined by the administrator. This features can be used by all VISHNU modules (not only UMS). This feature is used in particular to specify which technology is used for a set of VISHNU services (e.g. to use a specific transfer method for file transfers).

- **Password state**: Records the current state of the password of a user: either 'temporary' if the password must be changed next time the user connects to the System, or 'valid' if the password is in a normal state.

- **Root user**: Special user that is pre-configured in the VISHNU system and that has administrator privileges. This user cannot be deleted from the system.

- **Session**: A session is the context in which VISHNU commands are executed (ex: job submission, file transfers). It is created following authentifcation of a user and lasts until it is closed either manually or automatically.

- **Session Key**: The session key is a encrypted string generated by the System for a session. It is registered in an environment variable in order to avoid systematic authentification

- **Session closure policy**: This is an option which represents the way to close the session. There are two possible values to this option: "Close on timeout" which means that the session will be closed after an inactivity delay, and "Close on disconnect" which means that the session will be closed when the user disconnects from the terminal used to connect to VISHNU.

- **Session identifier**: The session identifier (or session id) is an identifier of a session easy to manipulate by a user compare to the session key

- **User account** : The user account is an object that contains a VISHNU user information.

- **User authentification system**: A user-authentication system is an information system declared in VISHNU in order to extend the UMS database authentication. One example of such a user-authentication system would be an LDAP directory. It is important to notice that a user authentication can be achieved using the UMS database only, a user-authentication system only, or a combination of the UMS database and a user-authentication systems (with support for different priorities).

Chapter 3

Use cases for Tasks Management Service (TMS)

3.1 Use case diagrams

3.1.1 UC TMS Overview
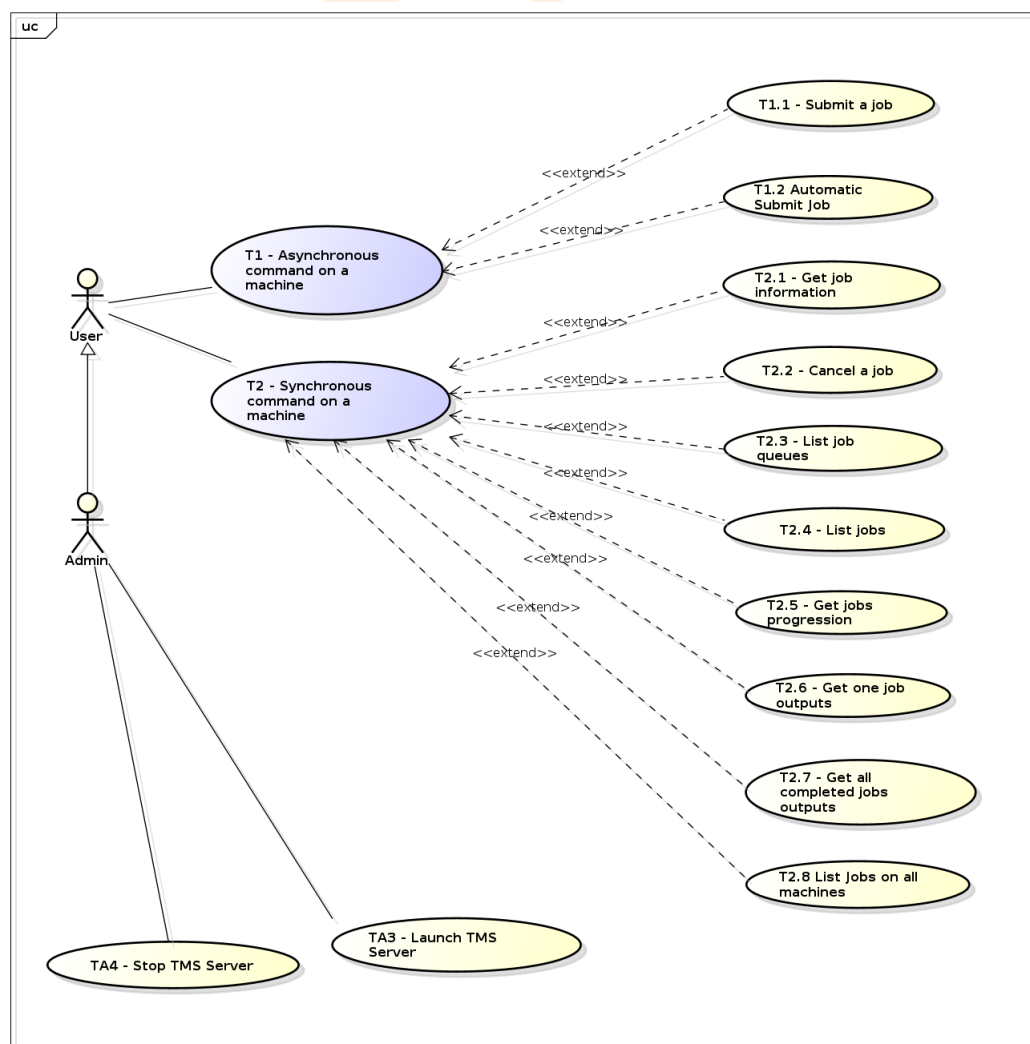
Figure 3.1: UC TMS Overview

3.2   Use case descriptions

3.2.1   T1 - Asynchronous command on a machine

Title	T1 - Asynchronous command on a machine
Summary	User starts an asynchronous command on a given machine
Actors	User
Precondition	- User has an active open session
Postcondition	- The command is in active state until completed - The system log has been updated and contains the request parameters
Base sequence	1. User sends the request 2. The System checks that the session key is valid 3. If the request contains the machine id the System checks that machine id is valid and machine is available 4. If command parameters contain a file the System verifies that the file is available and readable 5. The System processes the request 6. The System returns information to the user 7. The System records request information (time, user, machine, request parameters) in the system log
Branch sequence	5a. T1.1 5b. T1.2
Exception sequence	1a. The TMS server is unavailable - The system returns an error message that informs the user. 2a. The session key is invalid - The system returns an error message that informs the user. 3a. The name of the given machine is unknown -The system returns an error message that informs the user. 4a. The path to a file parameter is invalid - The system returns an error message that informs user.
Extensions	T1.2 Automatic Submit Job T1.1 - Submit a job

3.2.2   T1.1 - Submit a job

Title	T1.1 - Submit a job
Summary	User submits a job on a given machine
Actors	User
Precondition	
Postcondition	- The job is submitted on the specified machine - The job state and id are recorded on the system's log - The job id is sent to the user
Base sequence	1. The System checks that request parameters contain: - job script path - job options 2. The TMS server on the given machine is contacted 3. The job is submitted by the TMS server to the batch scheduler 4. The id of the submitted job is returned to the user
Branch sequence	
Exception sequence	1a. Invalid options or script 4a. The batch scheduler server is unavailable 4b. The batch scheduler server rejects the request

Extension of	T1 - Asynchronous command on a machine

3.2.3 T1.2 Automatic Submit Job

Title	T1.2 Automatic Submit Job
Summary	User asks to submit a job automatically on the best machine (the used criterion is the minimum number of waiting jobs or minimum running jobs or minimum total jobs)
Actors	
Precondition	- The machine id is equal to autom
Postcondition	- The job is submitted on the best machine (the used criterion is the minimum number of waiting jobs or minimum running jobs or minimum total jobs) - The job state and id are recorded on the system's log - The job id is sent to the user
Base sequence	1. The System checks that request parameters contain: - job script path - job options 2. The TMS server on the best machine is contacted 3. The job is submitted by the TMS server to the batch scheduler 4. The id of the submitted job is returned to the user
Branch sequence	
Exception sequence	1a. Invalid options or script 4a. The batch scheduler server is unavailable 4b. The batch scheduler server rejects the request
Extension of	T1 - Asynchronous command on a machine

3.2.4 T2 - Synchronous command on a machine

Title	T2 - Synchronous command on a machine
Summary	User executes a synchronous command on a given machine
Actors	User
Precondition	- User has an active open session
Postcondition	- Request is in completed state - The system log has been updated and contains the request parameters
Base sequence	1. The User sends the request with parameters including session key and machine id 2. The System checks that the session key is valid 3. If the request contains the machine id the System checks that the machine id is valid and machine is available 4. If command parameters contain a file the System verifies that the file is available and readable 5. The System processes the request 6. The System returns information containing the results of the request 7. The System records request information (time, user, machine, request parameters) in the system log

Branch sequence	5a. T2.1 5b. T2.2 5c. T2.3 5d. T2.4 5e. T2.5 5f. T2.6 5g. T2.7 5h. T2.8 5i. TA1 5j. TA2
Exception sequence	1a. The TMS server is unavailable - The system returns an error message that informs the user. 2a. The session key is not valid - The system returns an error message that informs the user. 3a. The name of the given machine is unknown -The system returns an error message that informs the user. 4a. The path to a file parameter is invalid - The system returns an error message that informs user. - The user revises the path
Extensions	T2.1 - Get job information T2.2 - Cancel a job T2.3 - List job queues T2.4 - List jobs T2.5 - Get jobs progression T2.6 - Get one job outputs T2.7 - Get all completed jobs outputs T2.8 List Jobs on all machines

3.2.5   T2.1 - Get job information

Title	T2.1 - Get job information
Summary	User gets information about a job on a given machine
Actors	User
Precondition	
Postcondition	
Base sequence	1. The Systems checks the job id 2. The TMS server on the given machine is contacted 3. The TMS server asks job information to the batch scheduler server 4. The User receives job information
Branch sequence	
Exception sequence	1a. The job id is invalid 3a. The batch scheduler server is unavailable 3b. The batch scheduler server rejects the request
Extension of	T2 - Synchronous command on a machine

3.2.6   T2.2 - Cancel a job

Title	T2.2 - Cancel a job
Summary	The user cancels a job on a given machine
Actors	User
Precondition	
Postcondition	- The job is canceled on the specified machine - The job state and id are removed to the system's log

Base sequence	1. The System checks the job id 2. If the User has no admin privilege, the System checks that the User is the submitter of the job 3. The System cancels the job 4. The System returns a confirmation to the User
Branch sequence	
Exception sequence	1a. The job id is invalid - The System returns an error message 2a. The User is not the submitter of the job - The System returns an error message 3a. The batch scheduler server is unavailable - The System returns an error message 3b. The batch scheduler server rejects the request - The System returns an error message
Extension of	T2 - Synchronous command on a machine

3.2.7 T2.3 - List job queues

Title	T2.3 - List job queues
Summary	User lists all queues or classes of a specific batch scheduler
Actors	User
Precondition	
Postcondition	
Base sequence	1. The User sends the request with parameters that include the machine id 2. The System obtains queues or classes information from the batch scheduler server running on the machine identified by the machine id 3. The System returns the list of all queues to the user
Branch sequence	
Exception sequence	2a. The batch scheduler server is unavailable 2b. The batch scheduler server rejects the request
Extension of	T2 - Synchronous command on a machine

3.2.8 T2.4 - List jobs

Title	T2.4 - List jobs
Summary	User lists all jobs submitted on a given machine matching some search criteria
Actors	User
Precondition	
Postcondition	
Base sequence	1. The User sends the request containing the machine id and the following optional search criteria: job id, number of CPUs required for the job, date of submission (from/to), job submitter, status, priority, queue, outputPath and errorPath. 2. The System obtains jobs information from the batch scheduler server (depends on the underlying batch scheduler software) 3. The System returns jobs information that match the search criteria to the User
Branch sequence	

Exception sequence	2a. The batch scheduler server is unavailable
	2b. The batch scheduler server rejects the request
Extension of	T2 - Synchronous command on a machine

3.2.9   T2.5 - Get jobs progression

Title	T2.5 - Get jobs progression
Summary	User gets jobs progression (execution percent) status on a machine
Actors	User
Precondition	
Postcondition	
Base sequence	1. The User sends the request containing the machine id 2. The System computes the job progression for all jobs submitted by the User running on the machine (job progression = 100 * (current_time - run_time) / job_walltime) 3. The System sends the results to the User
Branch sequence	1a. The User provides a job id in the request (optional parameter) 2a. The System computes the job progression for the job corresponding to the job id
Exception sequence	2b. The TMS server is unavailable - The system returns an error message that informs the user. 2c. The provided job id is unknown on the machine - The system returns an error message that informs the user.
Extension of	T2 - Synchronous command on a machine
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

3.2.10   T2.6 - Get one job outputs

Title	T2.6 - Get one job outputs
Summary	Output files of a given job are downloaded on the client host
Actors	User
Precondition	
Postcondition	- The job is removed from the Batch Scheduler's internal database.
Base sequence	1. The User sends the request containing the job id 2. The System checks the job status 3. The System downloads the job results if the job is completed 4. The System returns the path for each downloaded file
Branch sequence	
Exception sequence	2a. The TMS server is unavailable 2b. The batch scheduler is unavailable 2c. The job status is not 'completed' - The System returns a message that informs the user
Extension of	T2 - Synchronous command on a machine
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

3.2.11   T2.7 - Get all completed jobs outputs

Title	T2.7 - Get all completed jobs outputs
Summary	Output files of a user's completed jobs on a given machine are downloaded
Actors	User
Precondition	
Postcondition	- All the jobs submitted by the User on the machine are completed - All the jobs submitted by the User on the machine are removed from the Batch Scheduler's internal database.
Base sequence	1. The User sends the request containing the machine id 2. The System registers the request 3. The System checks the running jobs submitted by the User on the machine 4. The System sends the job outputs for all completed jobs to the client host 6. The User request is completed
Branch sequence	
Exception sequence	2a The TMS server is unavailable 2b The underlying batch scheduler is unavailable
Extension of	T2 - Synchronous command on a machine
Notes	This use case defines behaviour(s) that is/are not defined in the project initial requirements.

3.2.12   T2.8 List Jobs on all machines

Title	T2.8 List Jobs on all machines
Summary	User lists all jobs submitted on all machines matching some search criteria
Actors	
Precondition	-The machine id is equal to all
Postcondition	
Base sequence	1. The User sends the request containing the following optional search criteria: job id, number of CPUs required for the job, date of submission (from/to), job submitter, status, priority, queue, outputPath and errorPath. 2. The System obtains jobs information from the batch scheduler server (depends on the underlying batch scheduler software) 3. The System returns jobs information that match the search criteria to the User
Branch sequence	
Exception sequence	2a. The batch scheduler server is unavailable 2b. The batch scheduler server rejects the request
Extension of	T2 - Synchronous command on a machine

3.2.13   TA3 - Launch TMS Server

Title	TA3 - Launch TMS Server
Summary	The administrator launches the VISHNU TMS server on a given machine
Actors	Admin

Precondition	- The Vishnu server software (TMS Module and dependencies) is installed on the machine - The machine is configured in the Vishnu system database - The batch scheduler processes are up and running on the same machine - The network connection between the machine and the Vishnu database server is up and running
Postcondition	- The TMS server is up and running - A server log has been created
Base sequence	1. The Admin connects to the machine as vishnu user 2. The Admin updates the Vishnu configuration if necessary (database server hostname and credentials, SysferaDS configuration, Batch scheduler configuration) 3. The Admin launches the Vishnu TMS Server executable 4. The System checks the connections to its peers within the Vishnu platform 5. The System retrieves the list of active jobs (not completed jobs) that were launched on the same machine 6. The System checks that all the active jobs (from previous step) are still running on the batch scheduler, and eventually updates the job status (for ex. from waiting to running, or from running to finished) 7. The System returns a status message to the administrator
Branch sequence	
Exception sequence	4a. A connection to a Vishnu peer is down. System returns an error message and stops 6a. The batch scheduler does not recognize some job ids. In this case the System updates the job status to completed.

3.2.14   TA4 - Stop TMS Server

Title	TA4 - Stop TMS Server
Summary	The administrator stops the VISHNU TMS server on a given machine
Actors	Admin
Precondition	- The TMS Server is up and running on the given machine
Postcondition	- The TMS Server is down
Base sequence	1. The Admin sends a request to stop the TMS Server and provides the machine id 2. The System updates the status of all active user requests (non-completed jobs) 3. The System stops all internal processes on the machine 4. The System returns an information message to the Admin
Branch sequence	
Exception sequence	

3.3   Data dictionary

• **Batch Scheduler**: A batch scheduler is a distributed resource manager that enables to allocate at best the resources to the jobs on a machine according to user needs (the needs are specified by the user by batch directives (batch options) in file or command line).

• **Job**: A job is a sequence of instructions (included batch scheduler directives) written to be launched by a specified batch

scheduler.

- **Job id**: A job id allows to identifie the job in the batch scheduler system.

- **Job path**: A jobPath is the path to the file (script) containing the instructions (batch directives or job characteristics, job execution command) of the job.

- **Job state**: A job state allows to know the progression of the job. It may have the following state : RUNNING, WAITING, COMPLETED, QUEUED, CANCELED, FAILED

- **Job walltime**: The maximum duration of a job as defined in the job submission parameters

- **Queue or Class**: A queue or class associates the resource limits (CPU wallclock time, CPU memory) and execution priorities of the jobs.

Chapter 4

Use cases for Information Management Service (IMS)

4.1 Use case diagrams

4.1.1 UC IMS Global functionalities

Global use case presenting all the IMS use case
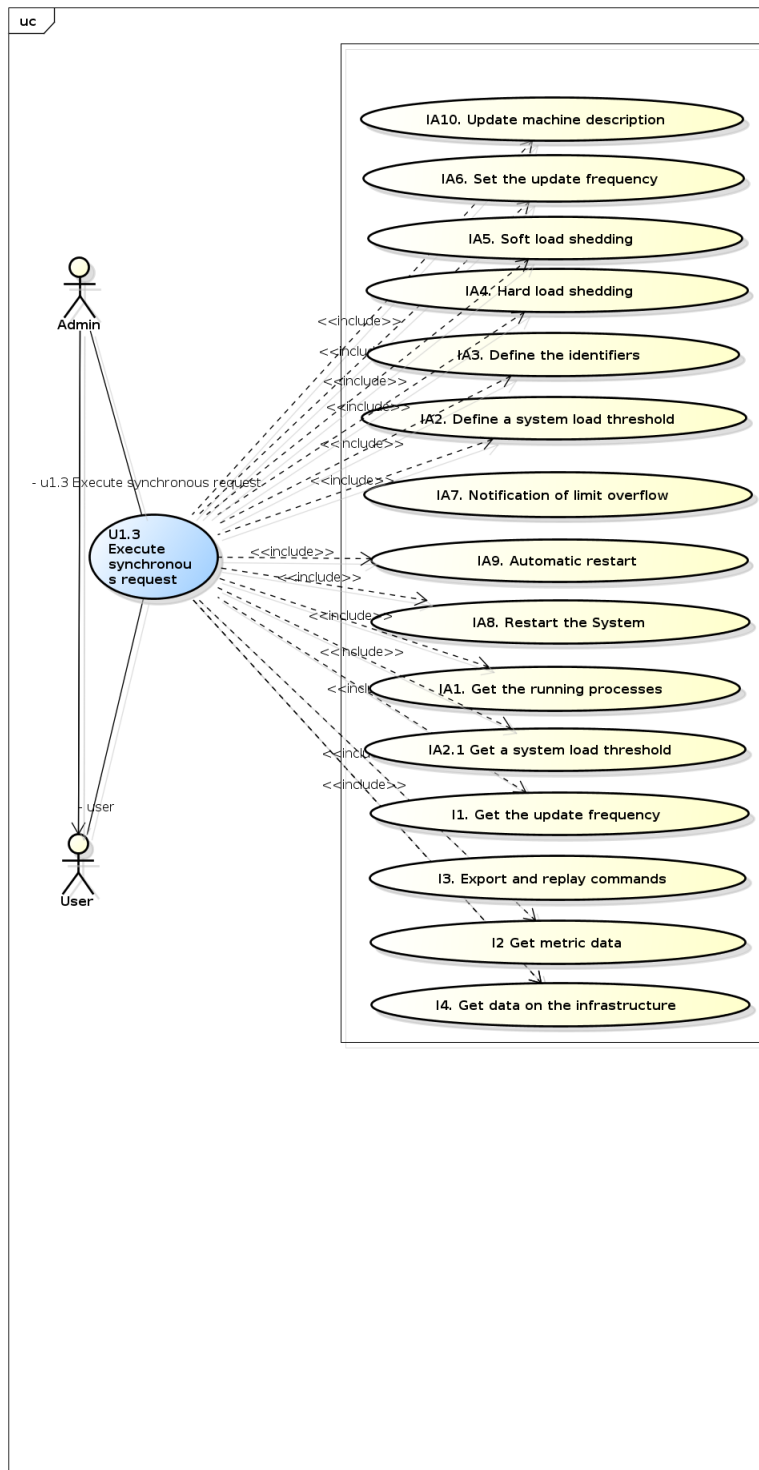
Figure 4.1: UC IMS Global functionalities

4.1.2 UC IMS Consult

User consult use case
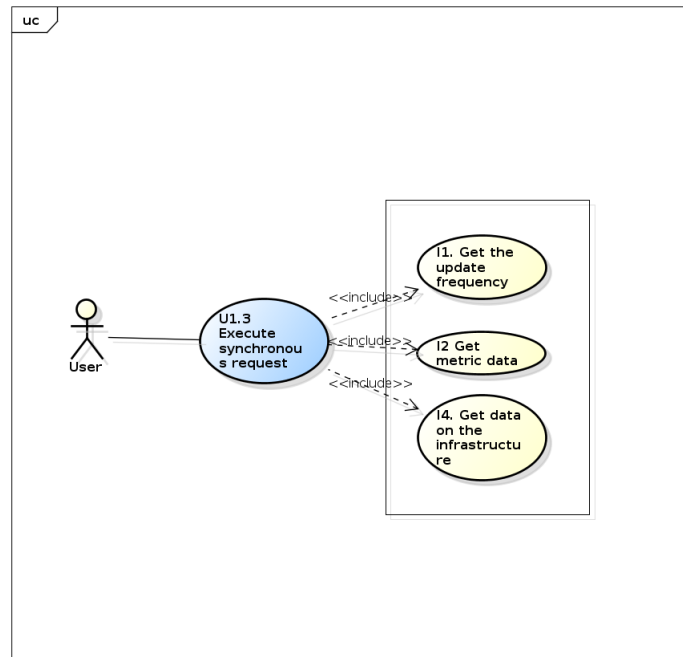
Figure 4.2: UC IMS Consult

4.1.3 UC IMS Replay

A user can replay its old commands of a session
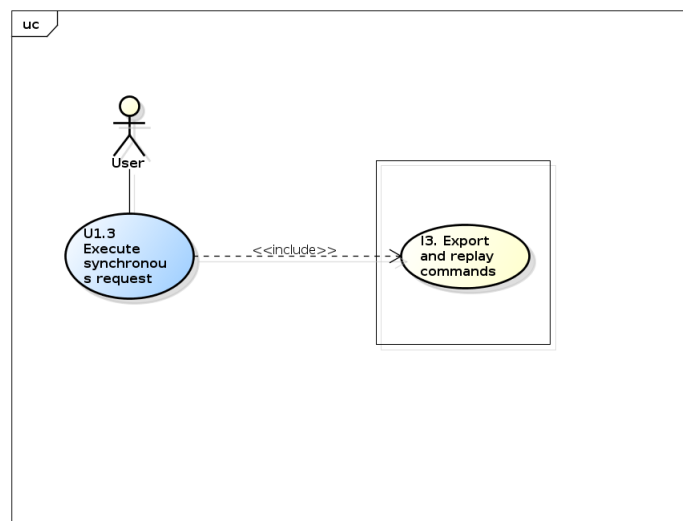
Figure 4.3: UC IMS Replay

4.1.4 UC IMS Platform management

All the use case of the administrator concerning the platform management
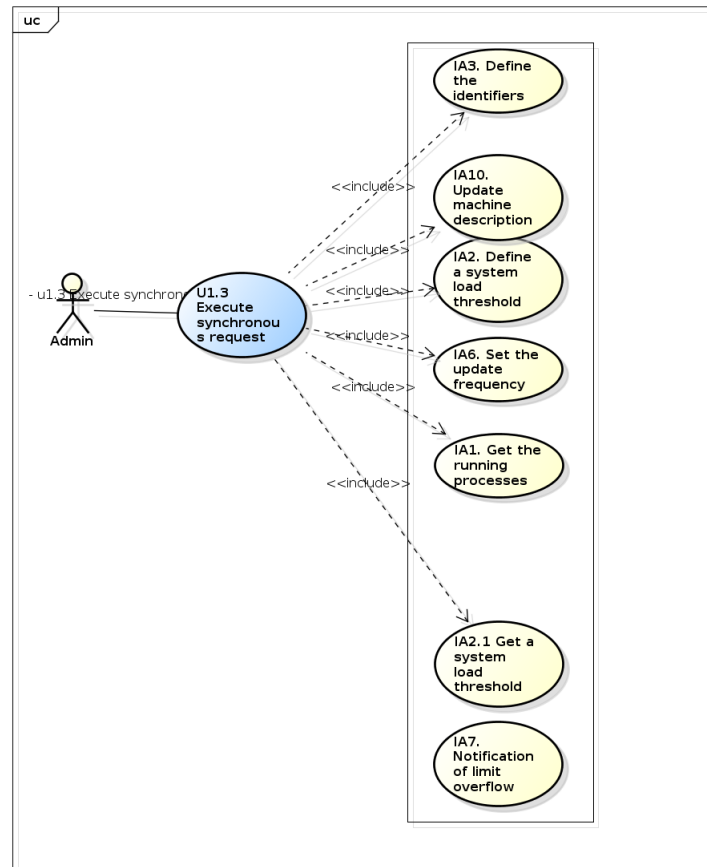
Figure 4.4: UC IMS Platform management

4.1.5   UC IMS Stop Restart

The administrator use cases concerning the stop and restart of the platform
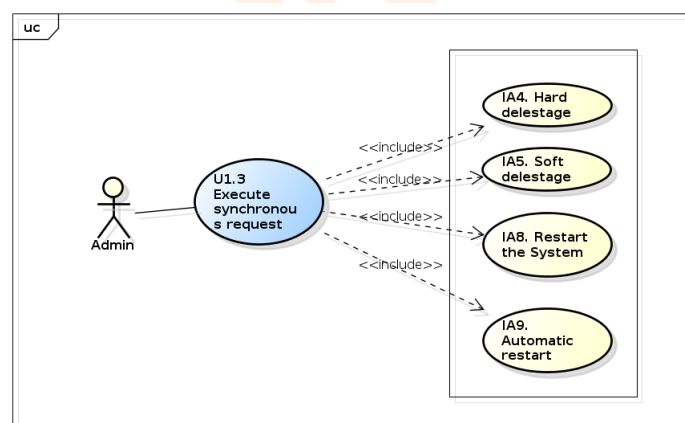
Figure 4.5: UC IMS Stop Restart

4.2 Use case descriptions

4.2.1 I1. Get the update frequency

Title	I1. Get the update frequency
Summary	The user gets how often the IMS database tables are updated
Actors	User
Precondition	
Postcondition	
Base sequence	1) The user calls the function to know how often the IMS database tables are automatically updated 2) The System returns the value in second
Branch sequence	
Exception sequence	2 -> There is a problem with the database, the system returns a DATABASE_ERROR

4.2.2 I2 Get metric data

Title	I2 Get metric data
Summary	The user gets data concerning the evolution of a metrics on a machine
Actors	User
Precondition	
Postcondition	
Base sequence	1) The user calls to get the metrics data on a machine identified by a machine id, for a metric type, from start time up to end time. The metrics are within {number of cpu, percentage of cpu used, total diskSpace, free diskSpace, total RAM, free RAM} 2) The System returns the results by groups (metric,value, time).
Branch sequence	
Exception sequence	1 -> The machine id is invalid, an INVALID_PARAMETER error is returned 2 -> There is a problem with the database, the system returns a DATABASE_ERROR

4.2.3 I3. Export and replay commands

Title	I3. Export and replay commands
Summary	The user exports and replays a sequence of commands made during a session.
Actors	User
Precondition	
Postcondition	All the System commands submitted during a session have been re-executed keeping the same order they had when they were originally launched.

Base sequence	1a) The user calls to export the history, in shell format, of a session identified by an id. 2a) The System provides a shell script containing all the commands of the session with the same parameters as provided initially by the user (including file paths, numbers, strings, options) 3a) The user executes the shell script in a shell
Branch sequence	
Exception sequence	1 -> The session id is invalid, an INVALID_PARAMETER exception is raised. 3 -> A command in the execution fails, the error of the command is returned

4.2.4   I4. Get data on the infrastructure

Title	I4. Get data on the infrastructure
Summary	The user gets current System information about the machines
Actors	User
Precondition	
Postcondition	
Base sequence	1) The user calls to get a current data about a machine identified by an ID. The data is within {use of cpu, number of cpu, total diskSpace, free diskSpace, total RAM, free RAM}. 2) The System returns the value of the data. In the use of cpu case, the value is in percentage.
Branch sequence	
Exception sequence	The machine id is invalid, an INVALID_PARAMETER exception is raised

4.2.5   I5. Get system info

Title	I5. Get system info
Summary	To get information on the system
Actors	
Precondition	
Postcondition	
Base sequence	1) The user calls to get the information about a machine given the id of the machine 2) The System gets the technical description of the machine (memory max, ram rax) from the database 3) The System returns these data to the user
Branch sequence	
Exception sequence	1b) The machine id is invalid: an INVALID_PARAMETER error is returned 2b) The database is unavailable, a DATABASE_ERROR is returned
Extensions	U1.3 Execute synchronous request

4.2.6   IA1. Get the running processes

Title	IA1. Get the running processes

Summary	The admin gets the list of the running Vishnu processes on a machine
Actors	Admin
Precondition	
Postcondition	
Base sequence	1) The admin calls to get the list of the processes on a machine referenced by a machine id 2) The System returns a list of Vishnu processes
Branch sequence	
Exception sequence	1 -> machineId is invalid, an INVALID_PARAMETER is return.

4.2.7 IA2. Define a system load threshold

Title	IA2. Define a system load threshold
Summary	The administrator defines a system load threshold for a machine
Actors	Admin
Precondition	
Postcondition	The system load threshold is added to the System database
Base sequence	1a) The administrator calls to define the limit size of the diskSpace to use with a machine id, a threshold value and an admin id (admin responsible for the threshold) 2a) The System updates the database
Branch sequence	1b) The administrator calls to define the limit of RAM available to he user with a machine id, a threshold value and an admin id (admin responsible for the threshold) 2b) The System updates the database ------------------------ 1c) The administrator calls to define the use of CPU threshold on a machine with a machine id, a treshold value and an admin id (admin responsible for the threshold) 2c) The System updates the database
Exception sequence	1* -> The admin ID is invalid, the database is not updated and an INVALID_PARAMETER error is returned 2* -> The modification of the database fails, a DATABASE_ERROR is returned.

4.2.8 IA2.1 Get a system load threshold

Title	IA2.1 Get a system load threshold
Summary	The admin wants to get the thresholds on a machine
Actors	Admin
Precondition	
Postcondition	
Base sequence	1) The admin calls to get the defined limit on a machine identified by an id. These thresholds are within {free diskSpace, free RAM, percentage of CPU used} 2) The System returns the value.
Branch sequence	
Exception sequence	1 -> The machine id is invalid, the user gets an INVALID_PARAMETER error returned 2 -> There is a problem with the database request, a DATABASE_ERROR is returned

4.2.9 IA3. Define the identifiers

Title	IA3. Define the identifiers
Summary	The administrator defines the format of the automatic identifiers for the System objects.
Actors	Admin
Precondition	
Postcondition	A new format will be used to create the new identifiers
Base sequence	1) The administrator has a list of variables to define the identifiers shape. He has a method by kind of object (an object is either a user or a machine or a task or a file transfer). Available variables are : YEAR: the last two digits, (e.g. 10 for 2010) MONTH: Numerical value of the month (from 1 to 12) DAY: Day number, from 1 to 31 TYPE: The object kind SITE: The place for machine/users NAME: Username or machine name CPT: A counter automatically increased (each kind of object has its counter). 2) He calls the function to redefine the format with some of the previous parameters in a string. A variable must be preceded by a '$' symbol. For example, "$TYPEDAYMONTH$YEAR$CPT" 3) The System database is updated, the System does not check if the given format creates unique identifiers. If the same identifier is created, it will corrupt the database (the key will not be unique)
Branch sequence	2 -> An invalid variable is given, an INVALID_PARAMETER is returned and the old format is still used 3 -> The update fails, a DATABASE_ERROR is returned
Exception sequence	

4.2.10 IA4.1 Hard load shedding

Title	IA4.1 Hard load shedding
Summary	Abruptly stops the processes running on a machine (the waiting actions are cancelled and the running ones are stopped). The processes cannot be automatically restarted.
Actors	Admin
Precondition	Processes are running on the System
Postcondition	The whole machine is flushed and no job is running on it
Base sequence	1) The admin launches the hard load shedding command on a machine identified by an id. 2) The System flushes all the waiting action. 3) The System stops all the running processes on this machine. These processes cannot be restarted.
Branch sequence	
Exception sequence	1 -> The id of the machine is invalid, an INVALID_PARAMETER is returned

4.2.11 IA4.2 Soft load shedding

Title	IA4.2 Soft load shedding
Summary	The admin purges all the waiting actions and stops the running ones. The stopped actions can be restarted later.
Actors	Admin
Precondition	Processes are running on the VISHNU system
Postcondition	No jobs are waiting to run or are running
Base sequence	1) The admin calls the soft load shedding command on a machine identified by an id. 2) The System flushes the waiting jobs and stops the running ones. They are stored and can be restarted later
Branch sequence	
Exception sequence	1 -> The machine id is invalid, an INVALID_PARAMETER error is returned

4.2.12   IA5. Set system info

Title	IA5. Set system info
Summary	Updates the data in the system concerning a machine (e.g., if the machine has some added memory diskSpace, some added memory)
Actors	Admin
Precondition	
Postcondition	The description of the machine in the database is updated
Base sequence	1) An admin calls to update the data concerning a machine identified by an id giving a new diskSpace size or a new memory size. 2) The System updates the database
Branch sequence	
Exception sequence	1 -> The machine id is invalid, an INVALID_PARAMETER error is returned 2 -> There is an error with the database, a DATABASE_ERROR error is returned

4.2.13   IA6. Set the update frequency

Title	IA6. Set the update frequency
Summary	The administrator sets the update frequency
Actors	Admin
Precondition	
Postcondition	The System updates the IMS database at the new frequency
Base sequence	1) The administrator calls to set the update frequency in seconds 2) The System updates its database update frequency value
Branch sequence	
Exception sequence	The database is is not reachable. A DATABASE_ERROR is returned.

4.2.14   IA7. Notification of limit overflow

Title	IA7. Notification of limit overflow
Summary	The admin is notified a limit overflow
Actors	Admin

Precondition	A machine on the System has a limit overflow
Postcondition	
Base sequence	1) The System gets the email adress of the admin to contact, using a config file to send the e-mail 2) The System sends a mail to the admin concerning the overflow. The mail contains the name of the machine and the concerned threshold.
Branch sequence	
Exception sequence	1 -> The system fails getting the admin e-mail, a DATABASE_ERROR error is returned 2 -> Sending the mail fails, a MAIL_ERROR error is returned.

4.2.15   IA8. Restart the System

Title	IA8. Restart the System
Summary	Restart all the servers, agents, and daemons of the System. The running actions are restarted.
Actors	Admin
Precondition	The System platform needs to be restarted
Postcondition	The System is running with the same servers, agents and daemons as defined in the deployement file. The interrupted actions that can be restarted are restarted from the beginning.
Base sequence	1) An admin detects a problem 2) An admin calls to restart the System with a deployement file 3) The System saves the current actions 4) The System restarts components following the deployement file guidance and restarts the stopped actions from the beginning
Branch sequence	
Exception sequence	4-> Fail to relaunch a component (server, daemon, agent), an UNREACHABLE_COMPONENT error is returned.

4.2.16   IA9. Restart

Title	IA9. Restart
Summary	A component is restarted
Actors	Admin
Precondition	A component of the platform is down
Postcondition	The component is up and running again
Base sequence	1) An admin detects that a component has stopped for unknown reasons (a component = server, daemon, agent) 2) The admin calls the System to relaunch the component with its name, the machine to relaunch it and the configuration file to use 3) The System relauches the component
Branch sequence	
Exception sequence	3-> Fail to restart the component, an UNREACHABLE_COMPONENT error is returned.

4.2.17   U1.3 Execute synchronous request

Title	U1.3 Execute synchronous request
Summary	The user submits a synchronous request to the System. c.f. the UMS use case description (U1.3)
Actors	User, Admin
Precondition	
Postcondition	
Base sequence	
Branch sequence	
Exception sequence	
Extension of	I5. Get system info

4.3 Data dictionary

- **Actions**: A generic naming to design both jobs and file transfers.

- **Agent**: A component of the VISHNU hierarchy.

- **CPU**: Central Processing Unit.

- **Daemon**: Daemon running on the machines.

- **DiskSpace**: File system memory (not volatile).

- **IMS**: Information Management System.

- **Infrastructure**: Contains all the machines directly under the System supervision.

- **Live measure**: Measure regularly updated.

- **Memory**: RAM (Random Access Memory, volatile).

- **Objects**: An object is an abstraction of what can be manipulated by the System (user, machine, task, file transfer).

- **Process**: Process of the system.

- **SeD**: A component of the VISHNU hierarchy executing jobs for the clients.

- **Task**: Job submited via the TMS module.

Chapter 5

Use cases for File Management Service (FMS)

5.1 Use case diagrams

5.1.1 UC FMS simple command use cases
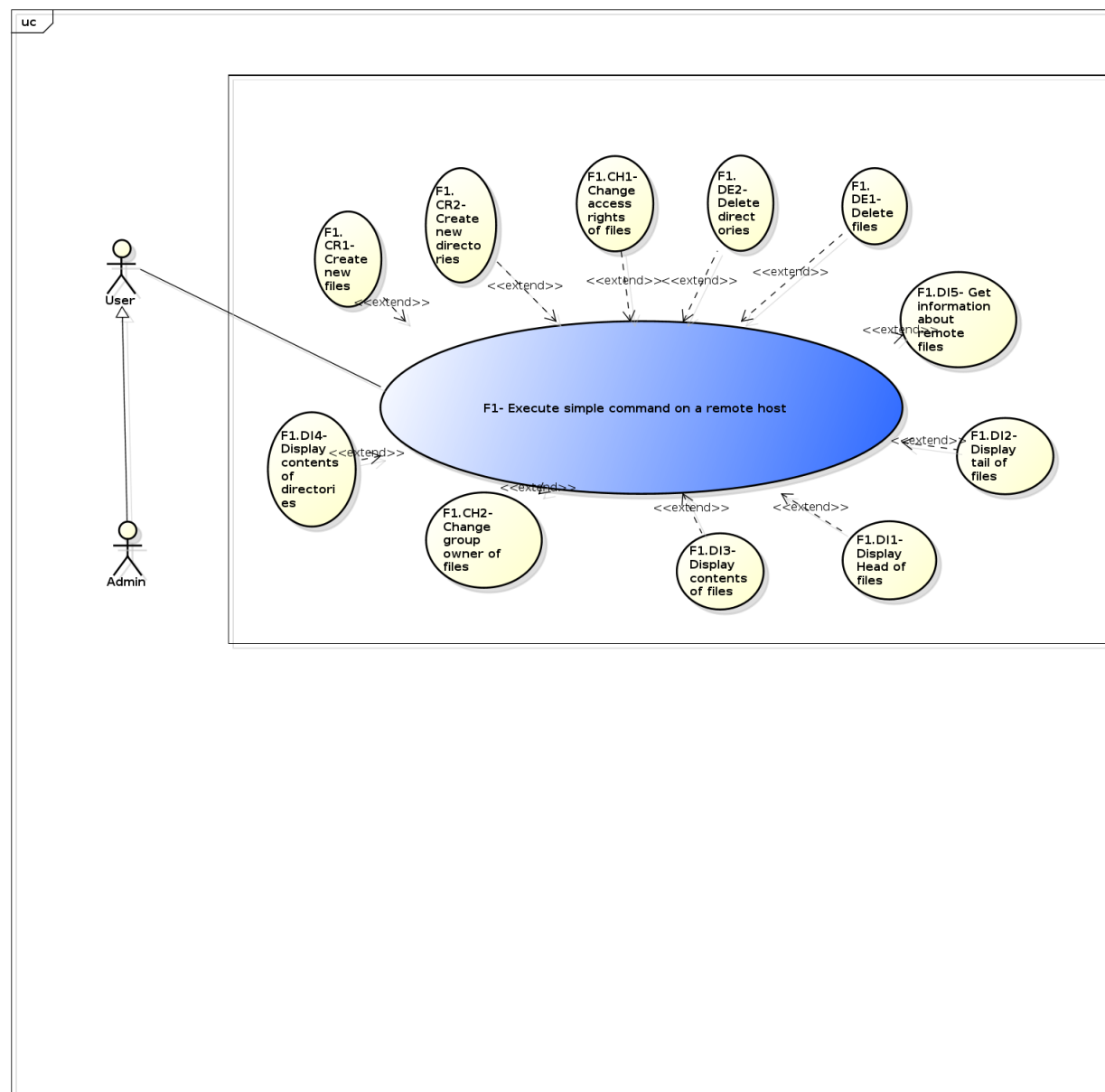
Figure 5.1: UC FMS simple command use cases

5.1.2 UC FMS transfer command use cases
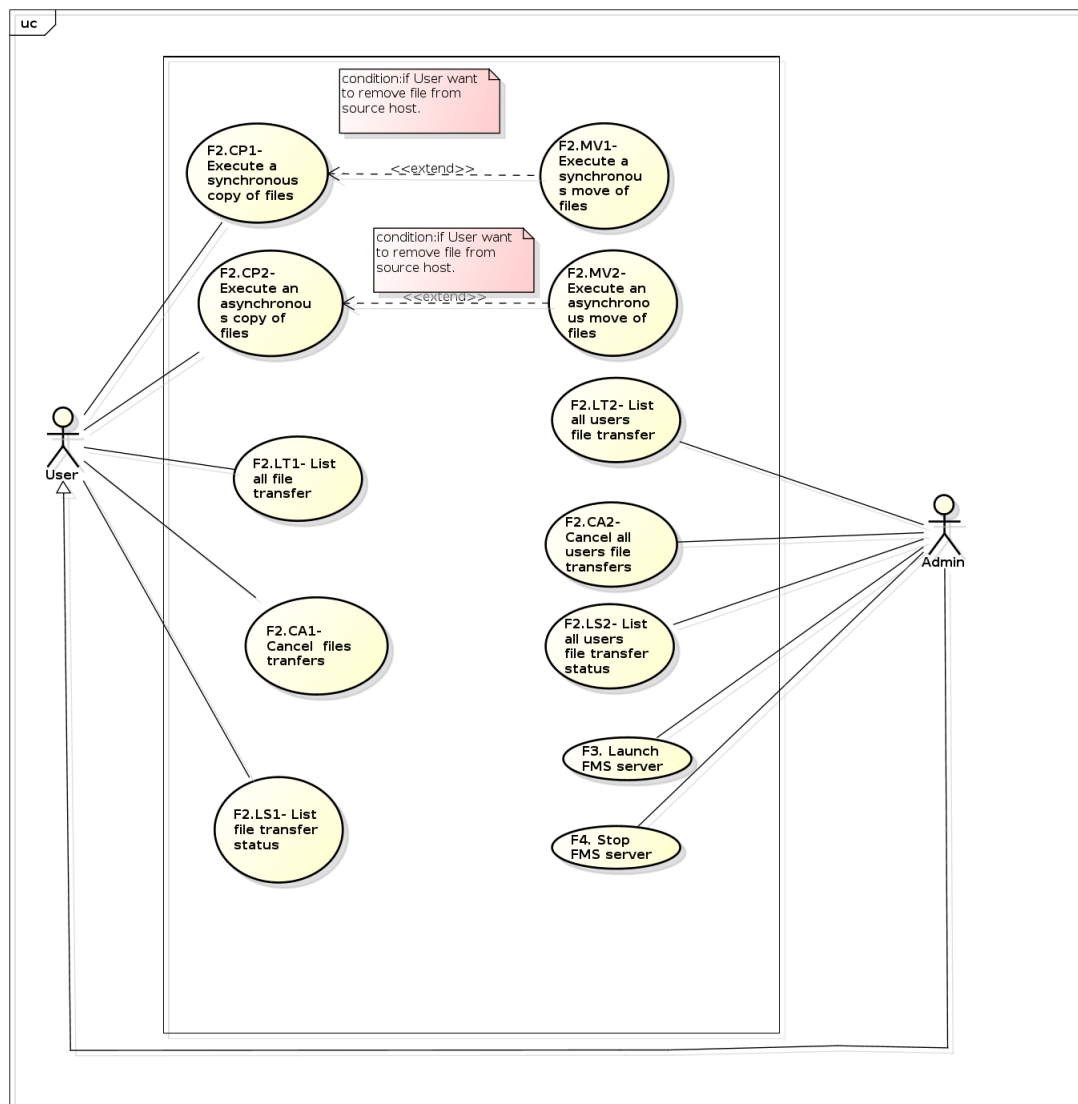
Figure 5.2: UC FMS transfer command use cases

5.2 Use case descriptions

5.2.1 F1- Execute simple command on a remote host

Title	F1- Execute simple command on a remote host
Summary	This use case allows User to execute a command on a remote host.
Actors	User
Precondition	- User has an active open session.
Postcondition	- The command is performed succesfully and the potential results are sent back to User. - The log System has been updated and contains request parameters.

Base sequence	1. User enters the command by specifying the parameters, the session key and the involved host id. 2. The System checks that the session key is valid. 3. The System checks that the host id is valid and the machine is available. 4. The System performs the command and send back the results to User . 5. The System records request information (time, User, machine, request parameters).
Branch sequence	
Exception sequence	1a. The given parameters are invalid for this command. 2a. The specified session key is invalid. 3a. The specified host is unknown. 3b. The specified host is unavailable. 4a. The command fails and an error message is displayed on the standard output of the client System.
Extensions	F1.CH2- Change group owner of files F1.CH1- Change access rights of files F1.CR2- Create new directories F1.DE2- Delete directories F1.DE1- Delete files F1.DI3- Display contents of files F1.DI1- Display Head of files F1.DI4- Display contents of directories F1.DI2- Display tail of files F1.DI5- Get information about remote files F1.CR1- Create new files F1.DE1- Delete files

5.2.2 F1.CH1- Change access rights of files

Title	F1.CH1- Change access rights of files
Summary	This use case allows User to change access rights of a given remote file. It is the equivalent of the "chmod" bash command.
Actors	User
Precondition	
Postcondition	The new access permissions of the specified file is set.
Base sequence	1. User submits the change access rights command with the file, the new access rights to set, the involved host. 2. The System sets the new access rights to the file.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, an error message is returned. 1b. If a file is unknown, an error message is displayed on the standard output of the client System. 1c. If User does not have execute permission in a parent directory or if User is not the file owner or Admin, a permission denied message is displayed on the standard output of the client System..
Extension of	F1- Execute simple command on a remote host

5.2.3 F1.CH2- Change group owner of files

Title	F1.CH2- Change group owner of files
Summary	This use case allows User to change the group owner of a named remote file. It is the equivalent of the "chgrp" bash command.
Actors	User
Precondition	
Postcondition	The new group owner of the specified file is set.
Base sequence	1. User submits the change group owner command with the file, the new group to set, the involved host. 2. The System sets the new group owner to the file.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If a file is unknown, a message is printed out on the standard output of the client System. 1c. If User does not have execute permission in a parent directory or if User is not the file owner or Admin, a permission denied message is displayed on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.4 F1.CR1- Create new files

Title	F1.CR1- Create new files
Summary	This use case allows User to create new file in a given host. It is the equivalent of the "touch" bash command.
Actors	User
Precondition	
Postcondition	The new file is created in the specified host and is owned by User and his group.
Base sequence	1. User submits the create file command with the path of file to create, the involved host. 2. The System creates the new file with the specified path.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If a specified file already exists, a message is printed out on the standard output of the client System. 1c. If User does not have execute or write permission in a parent directory, a message is also printed out on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.5 F1.CR2- Create new directories

Title	F1.CR2- Create new directories
Summary	This use case allows User to create a new directory in a named host. It is the equivalent of the "mkdir" bash command.
Actors	User
Precondition	

Postcondition	The new directory is created in the specified host and is owned by User and his group.
Base sequence	1. User submits the create directory command with the path of directory to create, the involved host. 2. The System creates the new directory with the specified path.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the a specified directory already exists, a message is printed out on the standard output of the client System. 1c. If User does not have read or write permission in a parent directory, a message is also printed on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.6 F1.DE1- Delete files

Title	F1.DE1- Delete files
Summary	This use case allows User to remove a given remote file. It is the equivalent of the "rm" bash command.
Actors	User
Precondition	
Postcondition	The specified file is removed from the host.
Base sequence	1. User submits the delete file command with the path of the file to delete, the involved host. 2. The System deletes the specified file from the host.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the specified path is not a file or if the file is unknown, a message is printed out on the standard output of the client System. 1c. If User does not have execute or write permission in the parent directory, a message is also printed out on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host F1- Execute simple command on a remote host

5.2.7 F1.DE2- Delete directories

Title	F1.DE2- Delete directories
Summary	This use case allows User to remove a given directory (and its content) located on a remote host. It is the equivalent of the "rm -r" bash command.
Actors	User
Precondition	
Postcondition	The specified directory is removed from the given host.
Base sequence	1. User submits the delete directory command with the path of directory to delete, the involved host. 2. The System deletes the specified directory from the host.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the specified path is not a directory or a directory is unknown, a message is printed on the standard output of the client System. 1c. If User does not have execute or write permission in the parent directory,or if the specified directory contains a file which can not be removed, a permission denied message is also printed out on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.8 F1.DI1- Display Head of files

Title	F1.DI1- Display Head of files
Summary	This command allows User to print the first few lines of a given remote file. It is the equivalent of the "head" bash command.
Actors	User
Precondition	
Postcondition	The first lines of the specified file are printed out on the standard output of the client System.
Base sequence	1. User submits the display command with the path of the file to display, the involved host. 2. The System displays the first lines of the specified file on the standard output of the client System.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the file is unknown, a message is printed out on the standard output of the client System. 1c. If User does not have execute permission write in the parent directory or read permission on the file, a message is also printed out on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.9 F1.DI2- Display tail of files

Title	F1.DI2- Display tail of files
Summary	This command allows User to print the last few lines of a named remote file. It is the equivalent of the "tail" bash command.
Actors	User
Precondition	
Postcondition	The last lines of the specified file are printed out on the standard output of the client System.
Base sequence	1. User submits the display command with the path of the file to display, the involved host. 2. The System displays the last lines of the specified file on the standard output of the client System.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the file is unknown, a message is printed on the standard output of the client System. 1.c If User does not have execute permission in the parent directory or read permission on the file, a permission denied message is also printed on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.10 F1.DI3- Display contents of files

Title	F1.DI3- Display contents of files
Summary	This use case allows User to print the content of a given file located on a remote host. It is the equivalent of the "cat" bash command.
Actors	User
Precondition	
Postcondition	The named file is printed on the standard output of the client System.
Base sequence	1. User submits the display command with the path of the file to display, the involved host. 2. The System prints the specified file on the standard output of the client System.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the file is unknown, a message is printed on the standard output of the client System. 1c. If User does not have execute permission in the parent directory or read permission on the file, a message is also printed on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.11 F1.DI4- Display contents of directories

Title	F1.DI4- Display contents of directories
Summary	This use case allows User to list the files contained in a given directory located on a remote host. It is the equivalent of the "ls" bash command.
Actors	User
Precondition	
Postcondition	The content of the specified directory is printed out on the standard output of the client System.
Base sequence	1. User submits the display command with the path of the directory to list, the involved host. 2. The System displays the content of the specified directory on the standard output of the client System.
Branch sequence	1a. If no directory is given , the content of current directory is displayed on the standard output of the client System. 1b.If a file is given, some information about the file (like the access permissions, the owner, the size, etc...) is printed out on the standard output of the client System.

Exception sequence	1a. If the directory is unknown, a message is printed out on the standard output of the client System. 1b. If User does not have execute or read permission in the parent directory, a message is also printed out on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.12 F1.DI5- Get information about remote files

Title	F1.DI5- Get information about remote files
Summary	This use case allows User to get information about a named remote file (the path, the owner, the group, the access permissions, the owner numeric identifier, the group numeric identifier, the size, the last access time, the last modification time, the last inode change time). It is equivalent to "stat" bash command.
Actors	User
Precondition	
Postcondition	Some informations about the given file are printed out on the standard output of the client System.
Base sequence	1. User submits the get information command with the file, the involved host. 2. The System prints out the information about the specified file on the standard output of the client System.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the file is unknown, a message is printed out on the standard output of the client System. 1.c If User does not have execute permission in the parent directory, a permission denied message is also printed out on the standard output of the client.
Extension of	F1- Execute simple command on a remote host

5.2.13 F2.CA1- Cancel files tranfers

Title	F2.CA1- Cancel files tranfers
Summary	This use case allows User to cancel all asynchronous file transfers he submitted. He can optionally specify either - a file transfer identifier to cancel that file transfer. - or a machine identifier to cancel all file transfers he submitted from that machine.
Actors	User
Precondition	User has at least an open active session.
Postcondition	- All file transfers submitted by User are cancelled. If a search criteria is provided, only file transfers matching that criteria are cancelled. - The log System has been updated and contains request parameters.

Base sequence	1. User submits a cancel file transfer command by specifying the session key. with optionally a search criteria (either a file transfer id, or a machine id). 2. The System cancels all file transfers he submitted. If a search criteria is provided, the System cancels only the file transfers matching that criteria.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the given session key is invalid, a message is printed out on the standard output of the client System. 1c. If a tranfer id is invalid or if User did not submit a named file tranfer,a message is printed out on the standard output of the client System. 1d. If the specified host id is invalid or if no file tranfer was submitted by User from that host, a message is printed out on the standard output of the client System. 1e. If the command fails, a message is printed on the standard output of the client System.

5.2.14   F2.CA2- Cancel all users file transfers

Title	F2.CA2- Cancel all users file transfers
Summary	This use case allows Admin to cancel all current asynchronous file transfers submitted. He can an optional search criteria among : - a host identifier to cancel all file transfers submitted from that host - or an user identifier to cancel all file transfers submitted by that user.
Actors	Admin, Admin
Precondition	Admin has at least an open active session.
Postcondition	- All file transfers submitted are cancelled. If a search criteria is provided, only the file transfers matching that criteria are cancelled. - The log System has been updated and contains request parameters.
Base sequence	1. Admin submits the cancel file transfer command by specifying a session key, with optionally a search criteria (either a host id, or a user id). 2. The System cancels all file transfers submitted. If a search criteria is provided, The System cancels only the file transfer matching that criteria.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the specified session key is invalid , a message is printed out on the standard output of the client System. 1c. If the specified host id is invalid or there is no file tranfer submitted from that host, a message is printed out on the standard output of the client System. 1d. If the specified user id is invalid or there is no file tranfer submitted by that user, a message is printed out on the standard output of the client System. 1e. If the command fails, a message is printed on the standard output of the client System.

5.2.15   F2.CP1- Execute a synchronous copy of files

Title	F2.CP1- Execute a synchronous copy of files
Summary	This use case allows User to copy a file (or directory) between two hosts. It is the equivalent of the "cp" bash command. The four cases of transfer are covered by this use case : - inside the same host which can be local or remote, - from local host to remote host, - from remote host to local host, - from remote host to another remote host.
Actors	User
Precondition	User has an open active VISHNU session on the client.
Postcondition	- The file transfer is fully accomplished and a copy of the source file (or directory) is now on the destination host. - The log System has been updated and contains request parameters.
Base sequence	1. User submits the file tranfer command with the path of the source file (or directory) to copy (including the host), the path of destination (including the host) and the session key. 2.The System copies the given source file (or directory) to the specified destination.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the given session key is invalid, a message is printed on the standard output of the client System. 1c. If the source file (or directory) or a host is unknown, a message is printed on the standard output of the client System. 1d. If the destination path is invalid, a message is printed on standard output of the client System. 1e. If the source path is a directory and a destination path is a file, a message is printed out on the standard output of the client System. 1f. If the source path is the same than the destination path, a message is returned. 1g. If - User does not have execute permission in the source or destination file parent, - or he does not have read permission on the source file, - or he does not have write permission in the destination parent directory, a message is printed out on the standard output of the client System. 2a. If a host is unreachable during a file transfer, the file transfer is cancelled and will restart when the connexion will be restored. 2b. If the transfer file fails, a message is also printed on the standard output of the client System.
Extensions	F2.MV1- Execute a synchronous move of files

5.2.16 F2.CP2- Execute an asynchronous copy of files

Title	F2.CP2- Execute an asynchronous copy of files
Summary	This use case allows User to copy files (or directory) between two hosts and submit another command without waiting the end of file transfer. The four cases of transfer are covered this use case : - inside the same host which can be local or remote - from local host to remote host - from remote host to local host - from remote host to another remote host.
Actors	User
Precondition	User has an active open session on the client
Postcondition	- The file transfer is fully accomplished and a copy of the source file (or directory) is now on the destination host. - The log System has been updated and contains request parameters.
Base sequence	1. User submits the file tranfer command with the path of the source file (or directory) to copy (including the host), the path of destination (including the host) and the session key. 2. The System starts the transfer of the given source file to the specified destination and sends back to User a transfer id. 3. When the transfer file ends, the log System is updated.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System.
	1b. If the given session key is invalid, a message is printed on the standard output of the client System.
	1c. If the source file (or directory) or a host is unknown , a message is printed on the standard output of the client System.
	1d. If the destination path is invalid, a message is printed on standard output of the client System.
	1e. If the source path is a directory and the destination path is a file, a message is printed out on the standard output of the client System.
	1f. If the source path is the same than the destination path, a message is print out on the standard output of the client System.
	1g. If
	- User does not have execute permission in the source or destination file parent,
	- or he does not have read permission on the source file,
	- or he does not have write permission in the destination parent directory,
	a message is printed out on the standard output of the client System.
	2a. If a host is unreachable during a file transfer, the file transfer is cancelled and will restart when the connexion will be restored.
	2b. If the transfer file fails, a message is also printed on the standard output of the client System.
Extensions	F2.MV2- Execute an asynchronous move of files

5.2.17   F2.LS1- List file transfer status

Title	F2.LS1- List file transfer status
Summary	This use case allows User to list all file transfer status he submitted. He can optionally specify either
	- a file transfer identifier to get the status of that file transfer.
	- or a machine identifier to get the status of all file transfer he submitted from that machine.
Actors	User, User
Precondition	User has at least an open active session.
Postcondition	- The status of all file transfers User submitted are displayed on the standard output of client System. If a search criteria is provided, only the status of the file transfer matching that criteria are listed.
Base sequence	1. User submits a list file transfer status command by specifying a session key, with optionally a search criteria (either a file transfer id, or a machine id).
	2. The System displays the status of all file transfers User submitted. If a search criteria is provided, the System displays only the status of the file transfers matching that criteria.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the specified session key is invalid , a message is printed out on the standard output of the client System. 1c. If the specified machine id is invalid , a message is printed out on the standard output of the client System. 1d. If the specified file transfer id is invalid , a message is printed out on the standard output of the client System. 1d. If no transfer was submitted by User from the specified machine or if the command fails, a message is printed out on the standard output of the client System.
Extensions	F2.LS2- List all users file transfer status

5.2.18   F2.LS2- List all users file transfer status

Title	F2.LS2- List all users file transfer status
Summary	This use case allows Admin to list file transfer status. He can specify an optional search criteria among : - a host identifier to list the status of all file transfers submitted from that host - or an user identifier to list the status of all file transfers submitted by that user.
Actors	Admin, Admin
Precondition	Admin has at least an open active session.
Postcondition	- The status of all file transfers submitted host are displayed on the standard output. If a search criteria is provided, only the file transfer matching that criteria are listed. - The log System has been updated and contains request parameters.
Base sequence	1. Admin submits a list file transfer status command a session key, with optionally a search criteria (either a host id, or a user id). 2. The System displays the status of all file transfers on the standard output of client System. If a search criteria is provided, The System displays only the status of the file transfers matching that criteria.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the specified session key is invalid , a message is printed out on the standard output of the client System. 1c. If the specified machine id is invalid , a message is printed out on the standard output of the client System. 1c. If the specified user id is invalid , a message is printed out on the standard output of the client System. 1d. If no transfer was submitted from the specified machine, a message is printed out on the standard output of the client System. 1e. If no transfer was submitted by the specified user or if the command fails, a message is printed out on the standard output of the client System.
Extension of	F2.LS1- List file transfer status

5.2.19 F2.LT1- List all file transfer

Title	F2.LT1- List all file transfer
Summary	This use case allows User to list all file transfers he submitted. User can optionally specify either - a machine id to list all file transfers he submitted from that machine - or a status to list all file transfers matching that status.
Actors	User
Precondition	User has at least an open active session.
Postcondition	- All file transfers User submitted are listed on the standard output of client System. If a search criteria is provided, only the file transfers matching that criteria are listed. - The log System has been updated and contains request parameters.
Base sequence	1. User submits a list file transfer command by specifying a session key with optionally a search criteria (either a host id, or a specific status). 2. The System displays all file transfers User submitted on the standard output of client System. If a search criteria is provided, the System displays only the file transfer matching that criteria.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the specified session key is invalid , a message is printed out on the standard output of the client System. 1c. If the specified machine id is invalid , a message is printed out on the standard output of the client System. 1d. If the specified status is invalid , a message is printed out on the standard output of the client System. 1e. If no transfer was submitted from the specified machine, a message is printed out on the standard output of the client System. 1f. If the command fails, a message is printed out on the standard output of the client System. t System.

5.2.20 F2.LT2- List all users file transfer

Title	F2.LT2- List all users file transfer
Summary	This use case allows Admin to list all file transfers. Admin can specify an optional search criteria among: - host identifier: to list all file transfers submitted from that host - user identifier: to list all file transfers submitted by that user - status: to list all file transfer matching that status.
Actors	Admin, Admin
Precondition	User has at least an open active session.
Postcondition	- All file transfers are listed on the standard output of client System. If a search criteria is provided, only the file transfers matching that criteria are listed. - The log System has been updated and contains request parameters.

Base sequence	1. Admin submits a list file transfer command by specifying a session key with optionally a search criteria (either a host id, or a user id, or a specific status). 2. The System displays all file transfers on the standard output of client System. If a search criteria is provided, The System displays only the file transfer matching that criteria.
Branch sequence	
Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the specified session key is invalid , a message is printed out on the standard output of the client System. 1c. If the specified machine id is invalid , a message is printed out on the standard output of the client System. 1d. If the specified user id is invalid , a message is printed out on the standard output of the client System. 1d. If the specified status is invalid , a message is printed out on the standard output of the client System. 1e. If no transfer was submitted from the specified machine, or by the specified user, a message is printed out on the standard output of the client System. 1e. If the command fails, a message is printed out on the standard output of the client System. t System.

5.2.21   F2.MV1- Execute a synchronous move of files

Title	F2.MV1- Execute a synchronous move of files
Summary	This use case allows User to copy a file (directory) from a host to another host. Furthermore, the source file (directory) is removed from the source host. The four cases of transfer are covered this use case : - inside the same host which can be local or remote - from local host to remote host - from remote host to local host - and from remote host to another remote host.
Actors	User
Precondition	
Postcondition	- The file transfer is fully accomplished. - A copy of the source file (directory) is now on the destination host, - and the source file (directory) is removed from the source host. - The log System has been updated and contains request parameters.
Base sequence	1. User submits the tranfer file command with the path of the source files (or directoty) to copy (including the host), the path of destination (including the host) and the session key. 2. The System makes a copy of the given source file (directory) to the specified destination and removes the source file (directory) from the source host.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System.
	1b. If the given session key is invalid, a message is printed on the standard output of the client System.
	1c. If the source file (or directory) or the host is unknown, a message is printed on the standard output of the client System.
	1d. If the destination path is invalid, a message is printed on standard output of the client System.
	1e. If the source path is a directory and the destination path is a file, a message is printed out on the standard output of the client System.
	1f. If
	- User does not have execute permission in the source or destination file parent,
	- or he does not have read permission on the source file,
	- or he does not have write permission in the destination parent directory,
	a message is printed out on the standard output of the client System.
	2a. If a host is unreachable during a file transfer, the file transfer is cancelled and will restart when the connexion will be restored.
	2b. If the transfer file fails, a message is also printed on the standard output of the client System.
Extension of	F2.CP1- Execute a synchronous copy of files

5.2.22 F2.MV2- Execute an asynchronous move of files

Title	F2.MV2- Execute an asynchronous move of files
Summary	This use case allows User to move file (or directory) from host to another host and submit another command without waiting the end of file transfer. Furthermore, the source file (or directory) is removed from the source host. The four cases of transfer are covered this use case : - inside the same host which can be local or remote - from local host to remote host - from remote host to local host - and from remote host to another remote host.
Actors	User
Precondition	User has at least an open active session.
Postcondition	- The file transfer is in completed status. - The source file (or directory) is removed from the source hosts. - The System log has been updated and contains request parameters.
Base sequence	1. User submits the file tranfer command with the path of the source file (or directory) to copy (including the host), the path of destination (including the host) and the session key. 2. The System starts the transfer of the given source file (or directory) to the specified destination and sends back to User a transfer id. 3. At the end of a transfer, the log System is updated.
Branch sequence	

Exception sequence	1a. If there are missing parameters, a message that contains the way to use the command, is displayed on the standard output of the client System. 1b. If the given session key is invalid, a message is printed on the standard output of the client System. 1c. If the source file (or directory) or the host is unknown , a message is printed on the standard output of the client System. 1d. If the destination path is invalid, a message is printed on standard output of the client System. 1e. If the source path is a directory and a destination path is a file, a message is printed out on the standard output of the client System. 1f. If - User does not have execute permission in the source or destination file parent, - or he does not have read permission on the source file, - or he does not have write permission in the destination parent directory, a message is printed out on the standard output of the client System. 2a. If a host is unreachable during a file transfer, the file transfer is cancelled and will restart when the connexion will be restored. 2b. If the transfer file fails, a message is also printed on the standard output of the client System.
Extension of	F2.CP2- Execute an asynchronous copy of files

5.2.23 F3. Launch FMS server

Title	F3. Launch FMS server
Summary	This use case allows Admin to launch the VISHNU FMS server on a given host.
Actors	Admin
Precondition	- The VISHNU server software (FMS Module and dependencies) is installed on the host - The host is configured in the VISHNU System database - The network connection between the host and the VISHNU database server is up and running.
Postcondition	- The FMS server is up and running. - A server log has been created.
Base sequence	1. Admin logs in the host as VISHNU user 2. Admin updates the VISHNU configuration if necessary (database server hostname and credentials, SysferaDS configuration) 3. Admin launches the VISHNU FMS Server executable 4. The System checks the connections to its peers within the VISHNU platform. 5. The System retrieves the list of active file transfer (not completed file transfer) that were launched from or to the same host. 6. The System checks that all the active file transfer (from previous step) are still running, and eventually updates the file transfer status (for ex. from failed to in progress). 7. The System returns a status message to Admin.

Branch sequence	
Exception sequence	4a. A connection to a VISHNU peer is down. System returns an error message and stops. 6a. If a source file or destination is unreachable. In this case the System updates the file transfer status to failed.

5.2.24 F4. Stop FMS server

Title	F4. Stop FMS server
Summary	This use case allows Admin to stop the VISHNU FMS server on a given host.
Actors	Admin
Precondition	- The FMS Server is up and running on the given host.
Postcondition	- The FMS Server is down.
Base sequence	1. Admin sends a request to stop the FMS Server and provides the host identifier. 2. The System updates the status of all on-going file transfer requests. 3. The System stops all internal processes on the host. 4. The System returns an information message to Admin.
Branch sequence	
Exception sequence	

5.3 Data dictionary

- **Host**: Computer connected to other computers or terminals to which it provides data or computing services via a network.

- **Inode**

  - An inode is a data structure on a filesystem on Linux and other Unix-like operating systems that stores all the information about a file except its name and its actual data.

  - When a file is created, it is assigned both a name and an inode number, which is an integer that is unique within the filesystem.

  - An inode contains all information describing a file.

  - This includes (1) the size of the file (in bytes) and its physical location (i.e., the addresses of the blocks of storage containing the file's data on a HDD),

  - (2) the file's owner and group, (3) the file's access permissions,

  - (4) timestamps telling when the inode was created, last modified and last accessed and (5) a reference count telling how many hard links point to the inode.

- **Path**: String of characters denoting the complete location of a file or folder (directory) in the host's data filing system.

- **Permissions**: The set of rights (read, write, execute) that are related to a file for a given user on a Unix filesystem.