VISHNU D1.0 - General specifications

COLLABORATORS

	TITLE : VISHNU D1.0 - General specifications		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Benjamin Isnard, Daouda Traoré, Eugène Pamba Capo-Chichi, Kevin Coulomb, and Ibrahima Cissé	January 7, 2011	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
01	07/12/2010	Formatting example	B.Isnard
02	13/12/2010	Pre-delivrable	B.Isnard
03	29/12/2010	Delivrable (draft)	B.Isnard

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 the 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 System
- TMS: Tasks Management System
- FMS: Files Management System
- IMS: Information Management System

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

- A first section containing "Use case descriptions" that follow the standard UML description of a use case
- A second section containing the "Use case diagrams" that describe the organization of the different use cases. These diagrams follow the UML2.0 standard.

1.3 References

1.4 Glossary

- SysferaDS: open-source middleware software used by Vishnu (former name "DIET")
-

Chapter 2

Use cases for User Management System (UMS)

2.1 Use case diagrams

2.1.1 UC UMS User Manual

This UseCase Diagram describes all cases attached when the user opens a session with manual closure

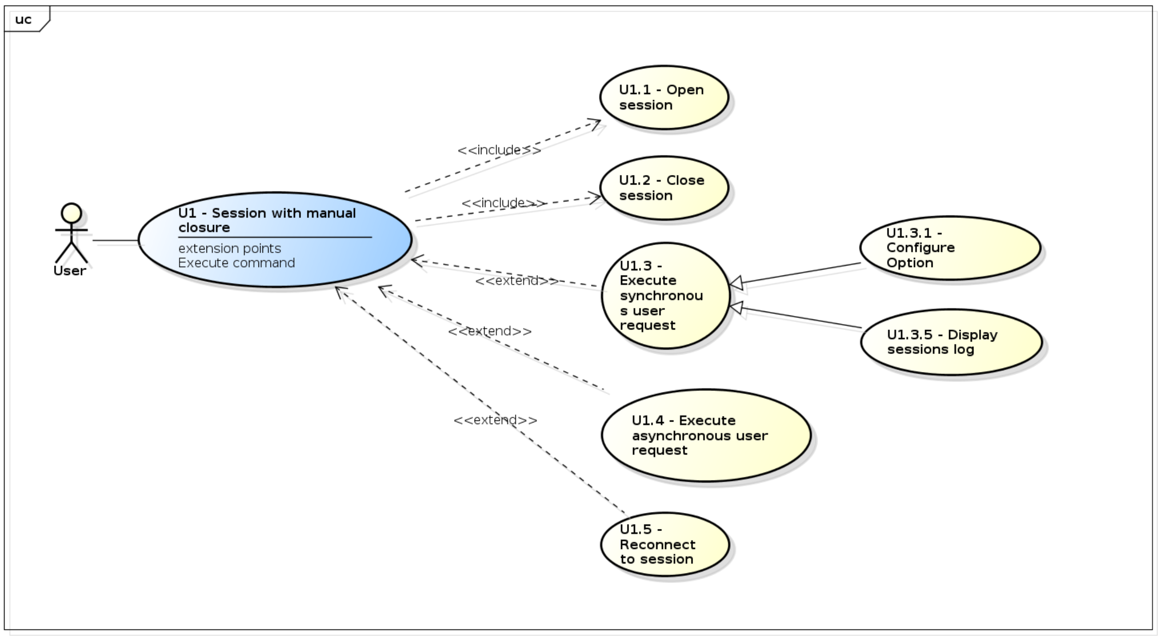
uc
User
U1 - Session with manual closure
extension points
Execute command
<<include>>
<<include>>
<<extend>>
<<extend>>
<<extend>>
U1.1 - Open session
U1.2 - Close session
U1.3 - Execute synchronous user request
U1.3.1 - Configure Option
U1.3.5 - Display sessions log
U1.4 - Execute asynchronous user request
U1.5 - Reconnect to session

Figure 2.1: UC UMS User Manual

2.1.2 UC UMS User Auto

This UseCase Diagram describes all cases associated when a user opens a session with automatic closure (on disconnect and on timeout)

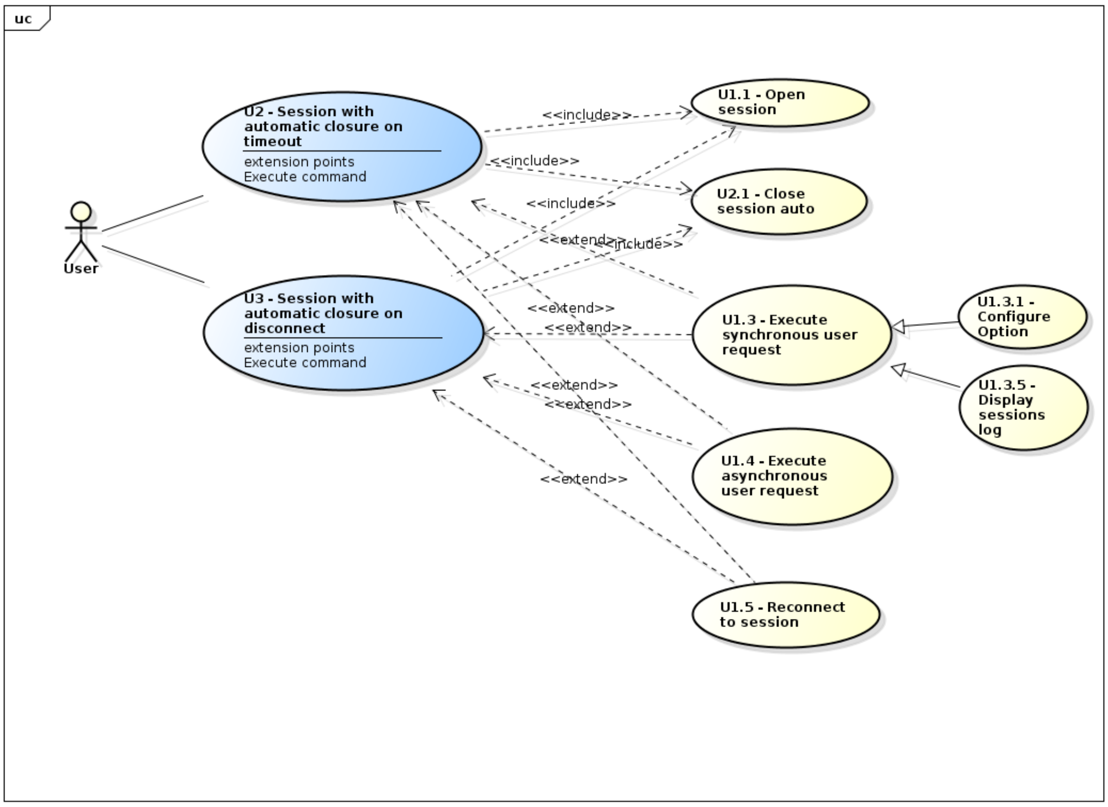
uc
User
U2 - Session with automatic closure on timeout
extension points
Execute command
U3 - Session with automatic closure on disconnect
extension points
Execute command
U1.1 - Open session
U2.1 - Close session auto
U1.3 - Execute synchronous user request
U1.3.1 - Configure Option
U1.3.5 - Display sessions log
U1.4 - Execute asynchronous user request
U1.5 - Reconnect to session
<<include>>
<<include>>
<<include>>
<<include>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<extend>>

Figure 2.2: UC UMS User Auto

2.1.3 UC UMS User account

This UseCase Diagram describes all cases associated when a user executes synchronous request

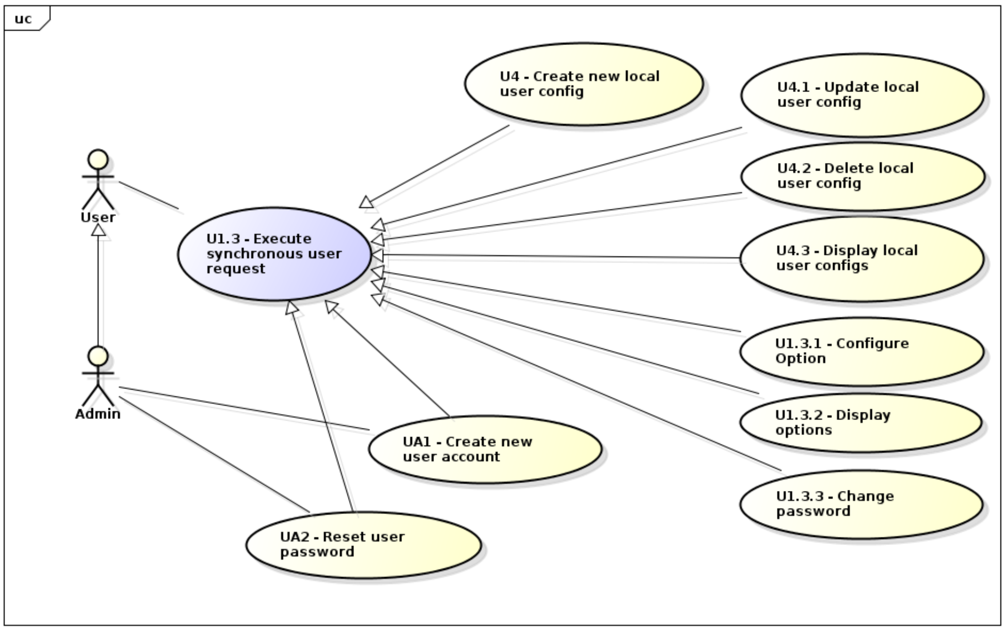
uc
User
Admin
U1.3 - Execute synchronous user request
U4 - Create new local user config
U4.1 - Update local user config
U4.2 - Delete local user config
U4.3 - Display local user configs
U1.3.1 - Configure Option
U1.3.2 - Display options
U1.3.3 - Change password
UA1 - Create new user account
UA2 - Reset user password

Figure 2.3: UC UMS User account

2.1.4 UC UMS Admin

This UseCase Diagram describes all administrator's functions

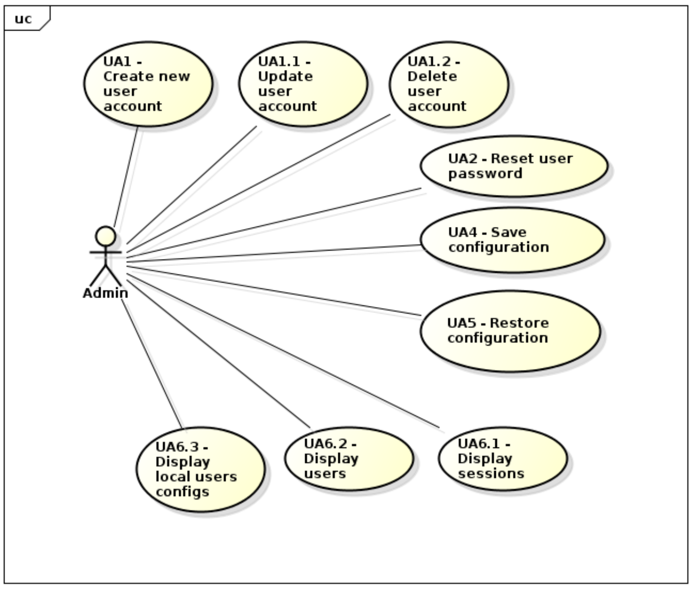
uc
Admin
UA1 - Create new user account
UA1.1 - Update user account
UA1.2 - Delete user account
UA2 - Reset user password
UA4 - Save configuration
UA5 - Restore configuration
UA6.3 - Display local users configs
UA6.2 - Display users
UA6.1 - Display sessions

Figure 2.4: UC UMS Admin

2.1.5 UC UMS Admin Machines

This UseCase Diagram describes all cases associated to machine using by an admin

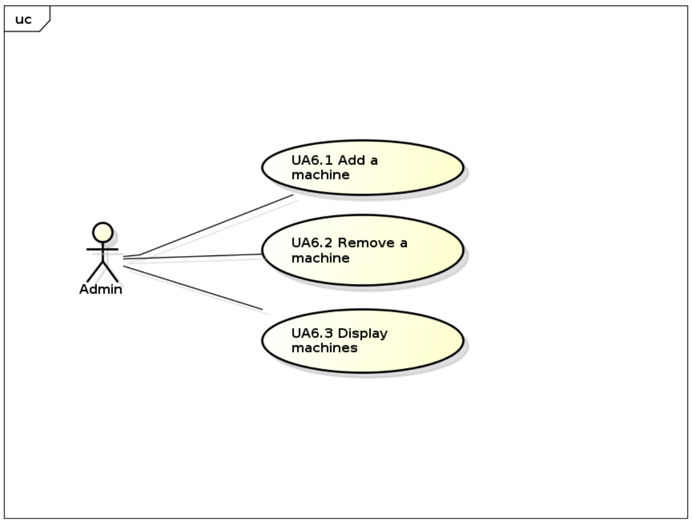

Figure 2.5: UC UMS Admin Machines

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 API 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 CLOSE_POLICY is equal to 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 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 - The user's environment contains a session key
Base sequence	1. User provides login, password and optionally the way for closing the session automatically on disconnect or on timeout to the "connect" command (when the default option is not set the SESSION_CLOSE_POLICY is CLOSE_ON_TIMEOUT). If the user is an administrator, he/she can be connected as he/she was another specific user by providing his/her login. 2. System validates login, password (user is authenticated) and optionally, the name of the closing mode (CLOSE_ON_DISCONNECT or CLOSE ON TIMEOUT) if the SESSION_CLOSE_POLICY is set or optionally a login provides by the administrator who wants to be connected as he/she was another user. 3. System creates the session and activates it 4. System provides the session key to the user
Branch sequence	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 non-interactive request then this is an exception (a change password request is required).
Exception sequence	2a. The user login is unknown 2b. The user password is invalid 2c. The SESSION_CLOSE_POLICY option is unknown 2d. VISHNU infrastructure is unreachable or unavailable 2e. The user password is temporary and request is non-interactive 2f. The substitute login provides by the administrator is unknown

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

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 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 2a. The System checks the name of the option specified 2b. The System returns the value of the option specified
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 wants to change his/her password
Actors	User
Precondition	
Postcondition	- The password is changed
Base sequence	1. The user sends a request containing his/her old password and the new password 2. The System checks the login and the old password of the user (the user si authenticated) and it registers the new user's password 3. The System returns an acknowledgment to the user
Branch sequence	
Exception sequence	1a. Unknwon user

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 correspons to the provided id
Exception sequence	1a1. Invalid session key (unknown / belonging to another user, if the current user is not an administrator) 1a2. 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 his/her sessions (active or closed)
Actors	User
Precondition	
Postcondition	
Base sequence	1. The user sends a request to list all his/her 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 all (or only active, or only closed) 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	
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	1.a Invalid session (bad session certificate or session unavailable) 1.b Invalid request 1.c Permission denied 1.d Ressource not available 1.e 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 wants to use a session in which he/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	- A session is active - The user's environment contains a session key
Base sequence	1. User provides its login, password and the identifier of the session (in fact, for each session, an identifier is assigned) 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. If the user is already within an active session then go to step 3 directly. The current session will be automatically closed according to the UC U2 or U3 depending on the close policy for that session.
Exception sequence	cf U1.1 (Open session) 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
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_CLOSE_POLICY is CLOSE_ON_TIMEOUT (either user option is defined or this is the default policy)
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	

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_CLOSE_POLICY is CLOSE_ON_DISCONNECT (either user option is defined or this is the default policy)
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 - The user has no local user configuration defined for the machine
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 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 4a. Invalid email address

2.2.16 U4.1 - Update local user config

Title	U4.1 - Update local user config
Summary	The user updates his/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	- local user configuration is updated
Base sequence	1. The user provides the login and the identifier of the machine used by his/her local configuration and information to be updated 2. The System checks the the local user configuration (login defined for the given machine) 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

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

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 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 4. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	2a. Login already used by another active user 3a. Invalid email address

2.2.20 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 2. The System updates user account (database) 3. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1.a Invalid login or login unknown 1.b The user information set are invalid 1.c The user information set are not consistent with the System

2.2.21 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)

2.2.22 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.

2.2.23 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

2.2.24 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.

2.2.25 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	

2.2.26 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
Exception sequence	1a1. The login is unknonwn

2.2.27 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

2.2.28 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

2.2.29 UA6.2 Remove a machine

Title	UA6.2 Remove a machine
Summary	The administrator unsubscribed a machine
Actors	Admin
Precondition	- The machine is registered in the System
Postcondition	The Machine is unsubscribed
Base sequence	1. The administrator removes a machine from VISHNU by giving the machine ID 2. The System returns an acknowledgment to the administrator
Branch sequence	
Exception sequence	1a. machine unknown

2.2.30 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 him/her
Exception sequence	1a1. The machine is unknown 1b1. The login is unknown

2.2.31 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
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

2.3 Data dictionary

- **CLOSE ON DISCONNECT**: CLOSE ON DISCONNECT is a value which means that the only one way for closing the session is when the user closes her/his terminal
- **CLOSE ON TIMEOUT**: CLOSE ON TIMEOUT is a value which means that the way for closing a session is after a session inactivity delay. This value is given by the client or registered by default by the administrator
- **Client System**: Client System or Client Host is a program which uses VISHNU API commands and that can be connected to VISHNU Infrastructure
- **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)
- **Local user config**: The local user config is the description of the given user on a specific machine described in the database
- **Manual closure**: The Manual closure means that the user uses the API command for closing the session
- **Option**: The option is a parameter of the user account that is not mandatory. Default value for each option is defined by the administrator. This features can be used by all VISHNU modules (not only UMS).
- **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_CLOSE_POLICY**: SESSION_CLOSE_POLICY is an option which represents the way to close the session (on timeout or on disconnect)
- **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 crypted string generated by the System for a session. It is registered in an environment variable in order to avoid systematic authentification
- **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
- **The inactivity delay**: The inactivity delay is the delay in which no api commands are lauched within a session
- **User account** : The user account is the description in the database of a VISHNU user

Chapter 3

Use cases for Tasks Management System (TMS)

3.1 Use case diagrams

3.1.1 UC TMS Overview

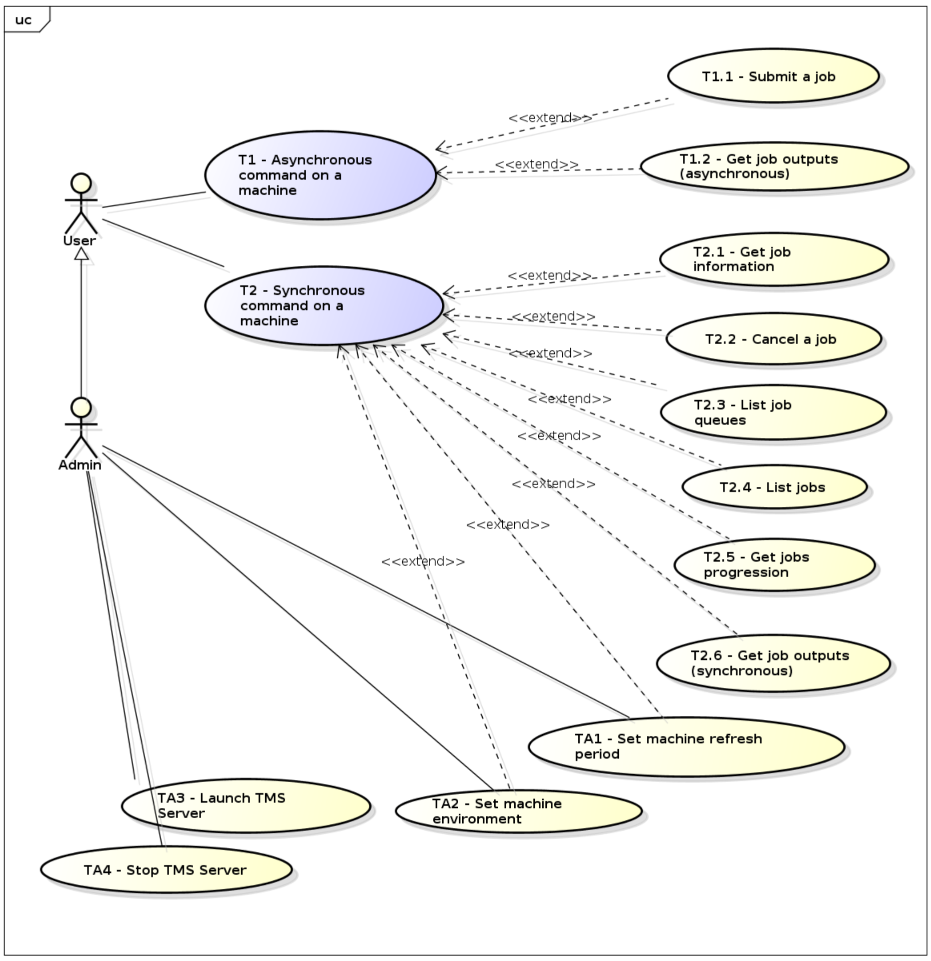
uc
User
Admin
T1 - Asynchronous command on a machine
T2 - Synchronous command on a machine
T1.1 - Submit a job
T1.2 - Get job outputs (asynchronous)
T2.1 - Get job information
T2.2 - Cancel a job
T2.3 - List job queues
T2.4 - List jobs
T2.5 - Get jobs progression
T2.6 - Get job outputs (synchronous)
TA1 - Set machine refresh period
TA2 - Set machine environment
TA3 - Launch TMS Server
TA4 - Stop TMS Server
<<extend>>

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. 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 to the user 6. 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	T1.1 - Submit a job T1.2 - Get job outputs (asynchronous)

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 - Get job outputs (asynchronous)

Title	T1.2 - Get job outputs (asynchronous)
Summary	Output files of a user's jobs on a given machine are downloaded when any job is completed
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 5. If the number of jobs submitted by the User on the machine with a waiting, queued or running status is positive, the System waits during a period defined by the administrator. If not, go to step 7 6. Go back to step 3 7. The User request is completed
Branch sequence	
Exception sequence	2a The TMS server is unavailable 2b The underlying batch scheduler is unavailable
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. 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. TA1 5g. 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 job outputs (synchronous) TA1 - Set machine refresh period TA2 - Set machine environment

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

3.2.10 T2.6 - Get job outputs (synchronous)

Title	T2.6 - Get job outputs (synchronous)
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

3.2.11 TA1 - Set machine refresh period

Title	TA1 - Set machine refresh period
Summary	The admin sets the refresh period of output and error file content

Actors	Admin
Precondition	
Postcondition	- The refresh period value is stored by the system
Base sequence	1. System saves the refresh period for the given machine. 2. System applies the refresh period to all current jobs and future requests
Branch sequence	
Exception sequence	1a. Refresh period value is too short (minimum value : see technical requirements) - System returns an error message
Extension of	T2 - Synchronous command on a machine

3.2.12 TA2 - Set machine environment

Title	TA2 - Set machine environment
Summary	The admin sets an environment variable on a given machine
Actors	Admin
Precondition	
Postcondition	- Environment variable is set on the machine
Base sequence	1. The User sends the request containing the machine id and a string containing the environment variable assignments (semi-column separated list of assignments <var_name>=<var_value>) 2. The System saves the environment variable for the given machine. 3. The System applies the environment variable to all current jobs and future requests
Branch sequence	
Exception sequence	
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 spiciefed 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 launch and to perform by a specified batch scheduler.
- **Job id**: A job id allows to identifie the job in the batch scheduler system.
- **JobPath**: A jobPath is the path to the file (script) containing the instructions (batch directives or job characteristiques, job execution command) of the job.
- **Queue ou Classe**: A queue or class allows to associate the resource limits (CPU wallclock time, CPU memory) and execution priorities of the jobs.
- **TMS**: Task Management System

Chapter 4

Use cases for Information Management System (IMS)

4.1 Use case diagrams

4.1.1 UC IMS Global functionalities

Global use case presenting all the IMS use case

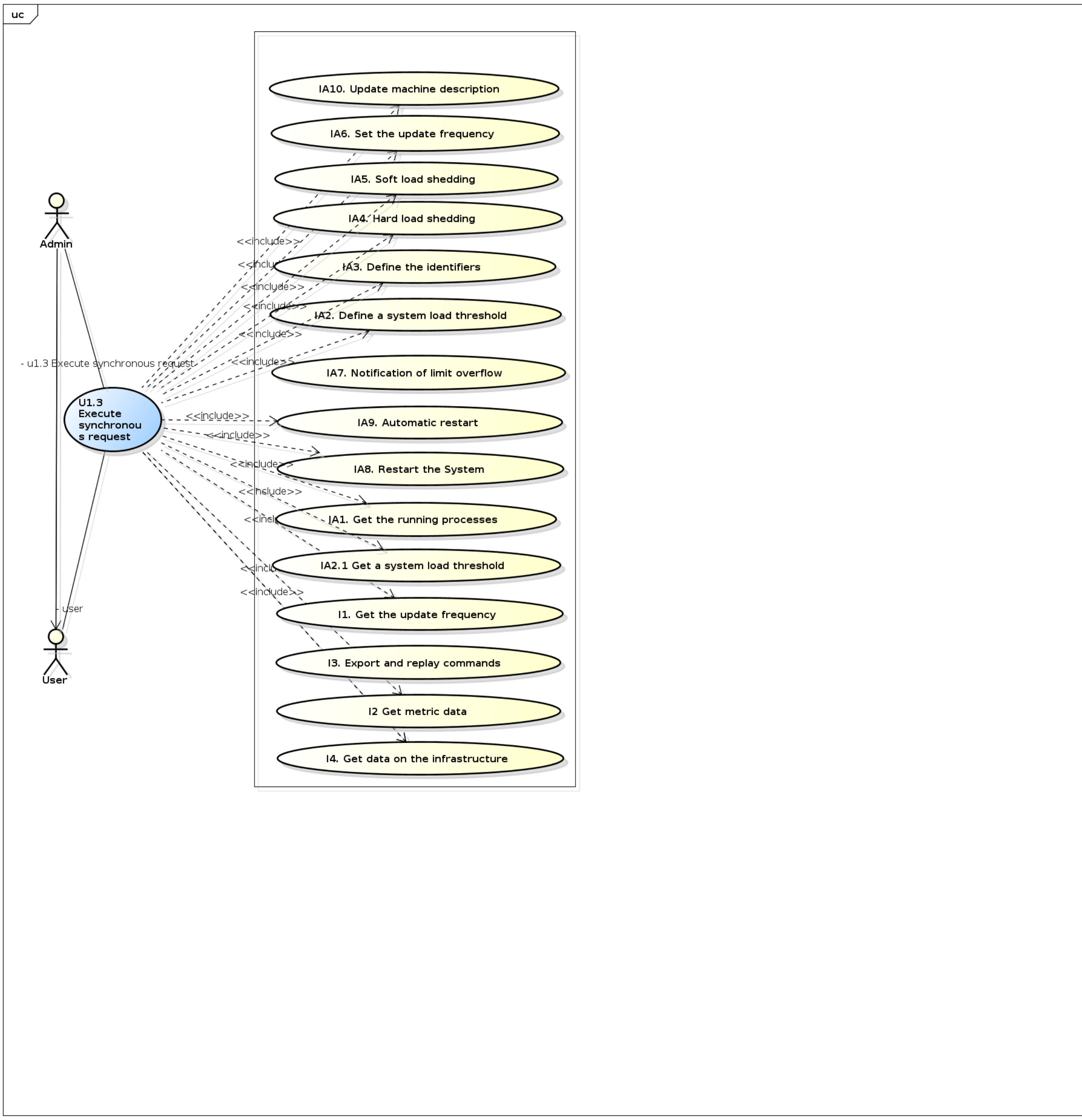
uc
Admin
User
- u1.3 Execute synchronous request
- user
U1.3 Execute synchronous request
<<include>>
IA10. Update machine description
IA6. Set the update frequency
IA5. Soft load shedding
IA4. Hard load shedding
IA3. Define the identifiers
IA2. Define a system load threshold
IA7. Notification of limit overflow
IA9. Automatic restart
IA8. Restart the System
IA1. Get the running processes
IA2.1 Get a system load threshold
I1. Get the update frequency
I3. Export and replay commands
I2 Get metric data
I4. Get data on the infrastructure

Figure 4.1: UC IMS Global functionalities

4.1.2 UC IMS Consult

User consult use case

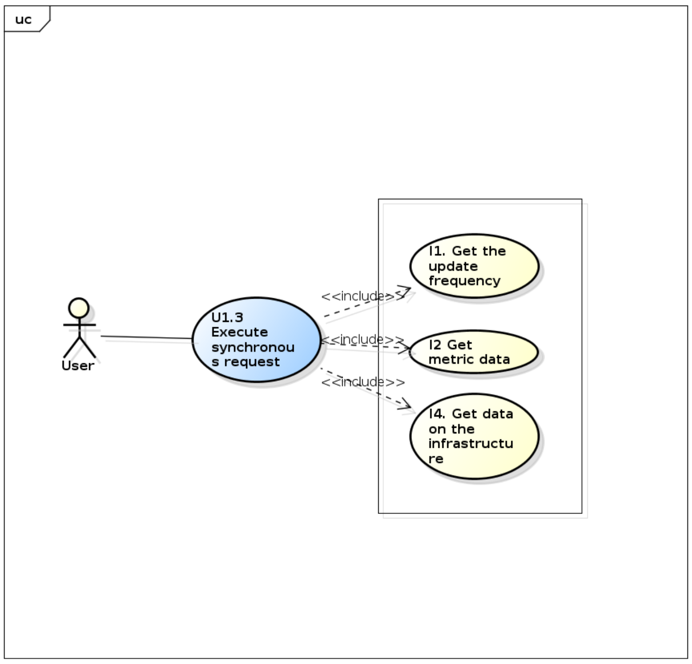
uc
User
U1.3 Execute synchronous request
<<include>>
<<include>>
<<include>>
I1. Get the update frequency
I2 Get metric data
I4. Get data on the infrastructure

Figure 4.2: UC IMS Consult

4.1.3 UC IMS Replay

A user can replay its old commands of a session

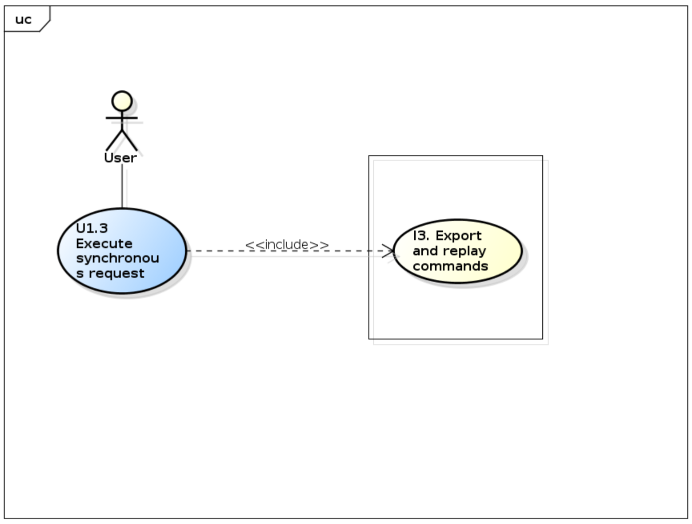
uc
User
U1.3 Execute synchronous request
<<include>>
I3. Export and replay commands

Figure 4.3: UC IMS Replay

4.1.4 UC IMS Platform management

All the use case of the administrator concerning the platform management

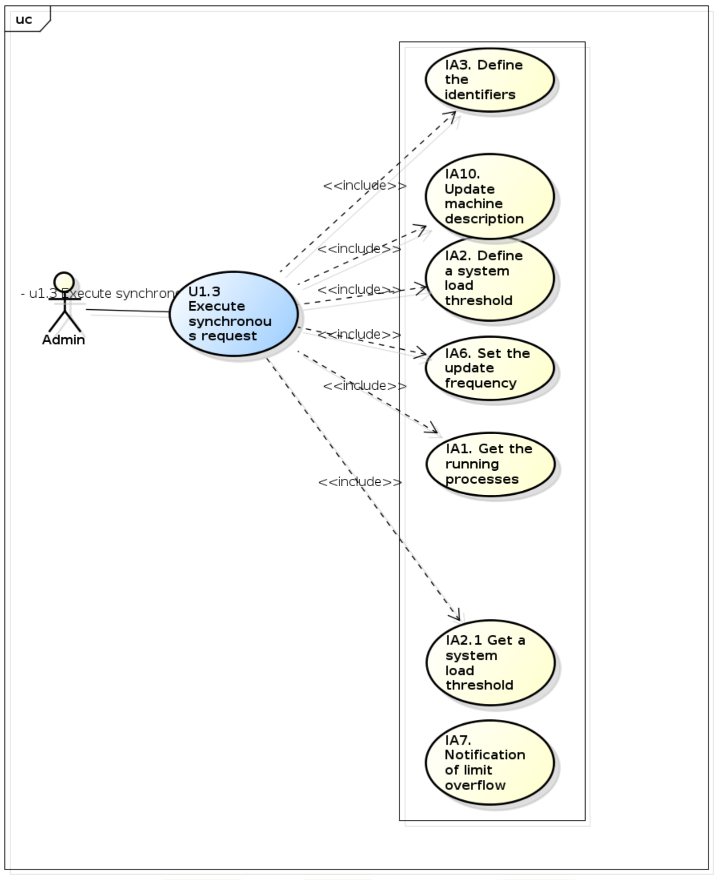
uc
Admin
- u1.3 Execute synchron
U1.3 Execute synchronous request
<<include>>
IA3. Define the identifiers
IA10. Update machine description
IA2. Define a system load threshold
IA6. Set the update frequency
IA1. Get the running processes
IA2.1 Get a system load threshold
IA7. Notification of limit overflow

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

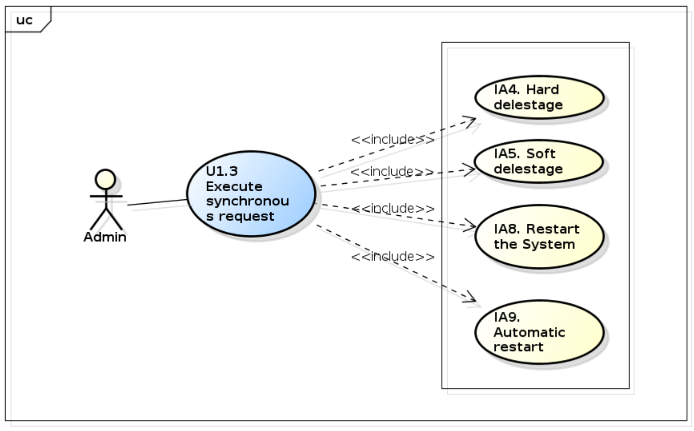

Figure 4.5: UC IMS Stop_Restart

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 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, number of processes running} 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	1) The user calls to export the history in python format of a session identified by an id 2) The System provides a python script containing all the commands of the session with the same parameters as provided initially by the user (including file paths, numbers, strings, options) 3) The user executes the python script in VISHNU
Branch 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
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 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 IA1. Get the running processes

Title	IA1. Get the running processes
Summary	The admin gets the list of the running 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 processes
Branch sequence	
Exception sequence	1 -> machineId is invalid, an INVALID_PARAMETER is return.

4.2.6 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 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 2b) The System updates the database ------------------------ 1c) The administrator calls to define the number of processes threshold on a machine with a machine id, a treshold value and an admin id 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.

4.2.7 IA2.1 Get a system load threshold

Title	IA2.1 Get a system load threshold
Summary	The user 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, number of processes running} 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

4.2.8 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) DA: 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. For example, "$TYPE$DAY$MONTH$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	

4.2.9 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 function 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

4.2.10 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 function on the 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

4.2.11 IA5. Update machine description

Title	IA5. Update machine description
Summary	Updates the data concerning a machine (e.g., if the machine has some added memory diskSpace, some added memory, a new description)
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, a new memory size and a new machine description. 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

4.2.12 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.

4.2.13 IA7. Notification of limit overflow

Title	IA7. Notification of limit overflow
Summary	The admin is informed of 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 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.

4.2.14 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 server, agents and daemons that were running before the crash. The interrupted actions are restarted from the beginning.
Base sequence	1) An admin detects a problem 2) An admin calls to restart the System 3) The System saves the current actions 4) The System restarts components 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.

4.2.15 IA9. Automatic restart

Title	IA9. Automatic 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 3) The System relauches the component
Branch sequence	
Exception sequence	3-> Fail to restart the component, an UNREACHABLE_COMPONENT error is returned.

4.2.16 U1.3 Execute synchronous request

Title	U1.3 Execute synchronous request
Summary	The user subsmits 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	

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 System (FMS)

5.1 Use case diagrams

5.1.1 UC FMS simple command use cases

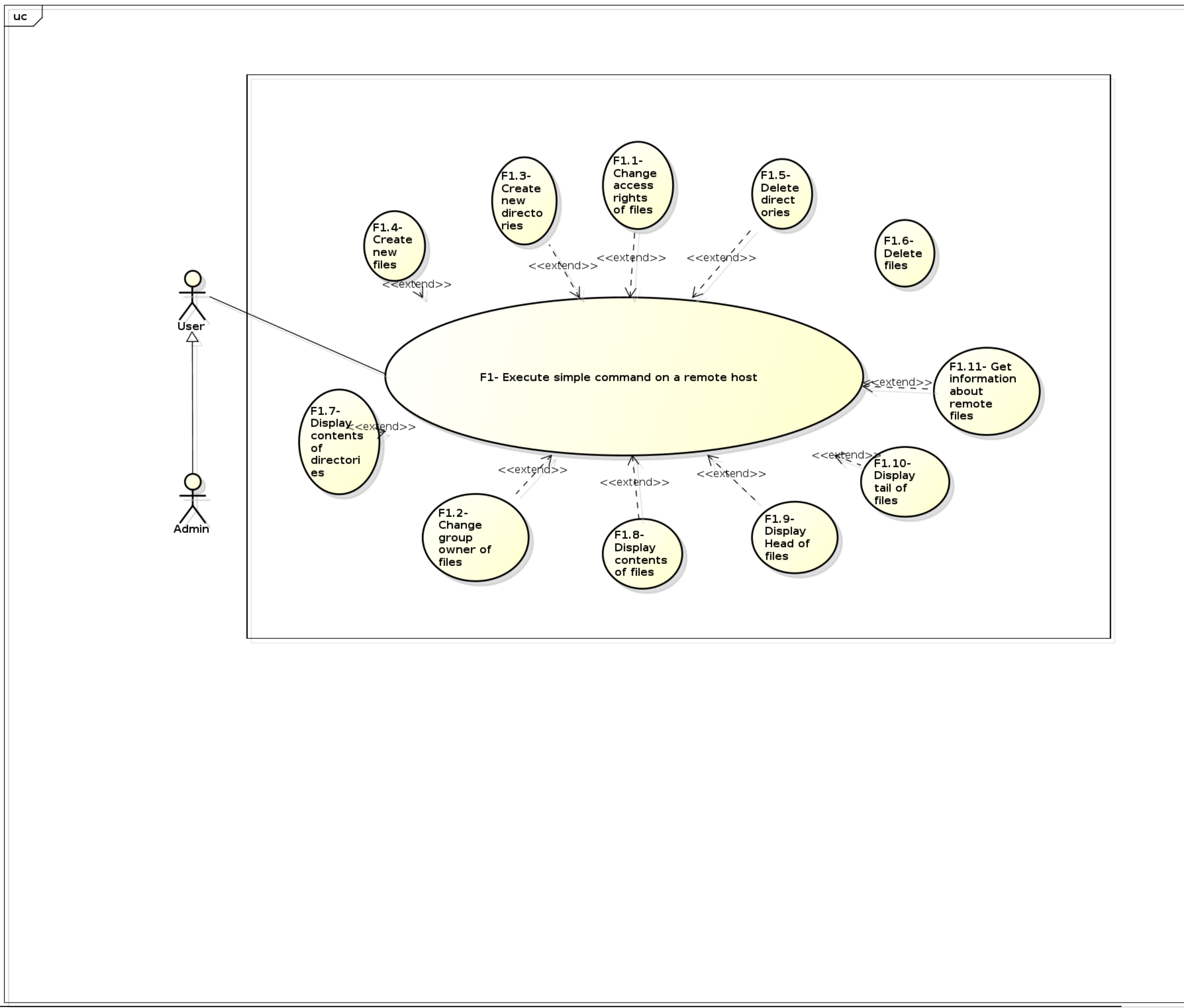
uc
User
Admin
F1- Execute simple command on a remote host
F1.1- Change access rights of files
F1.2- Change group owner of files
F1.3- Create new directories
F1.4- Create new files
F1.5- Delete directories
F1.6- Delete files
F1.7- Display contents of directories
F1.8- Display contents of files
F1.9- Display Head of files
F1.10- Display tail of files
F1.11- Get information about remote files
<<extend>>

Figure 5.1: UC FMS simple command use cases

5.1.2 UC FMS transfer command use cases

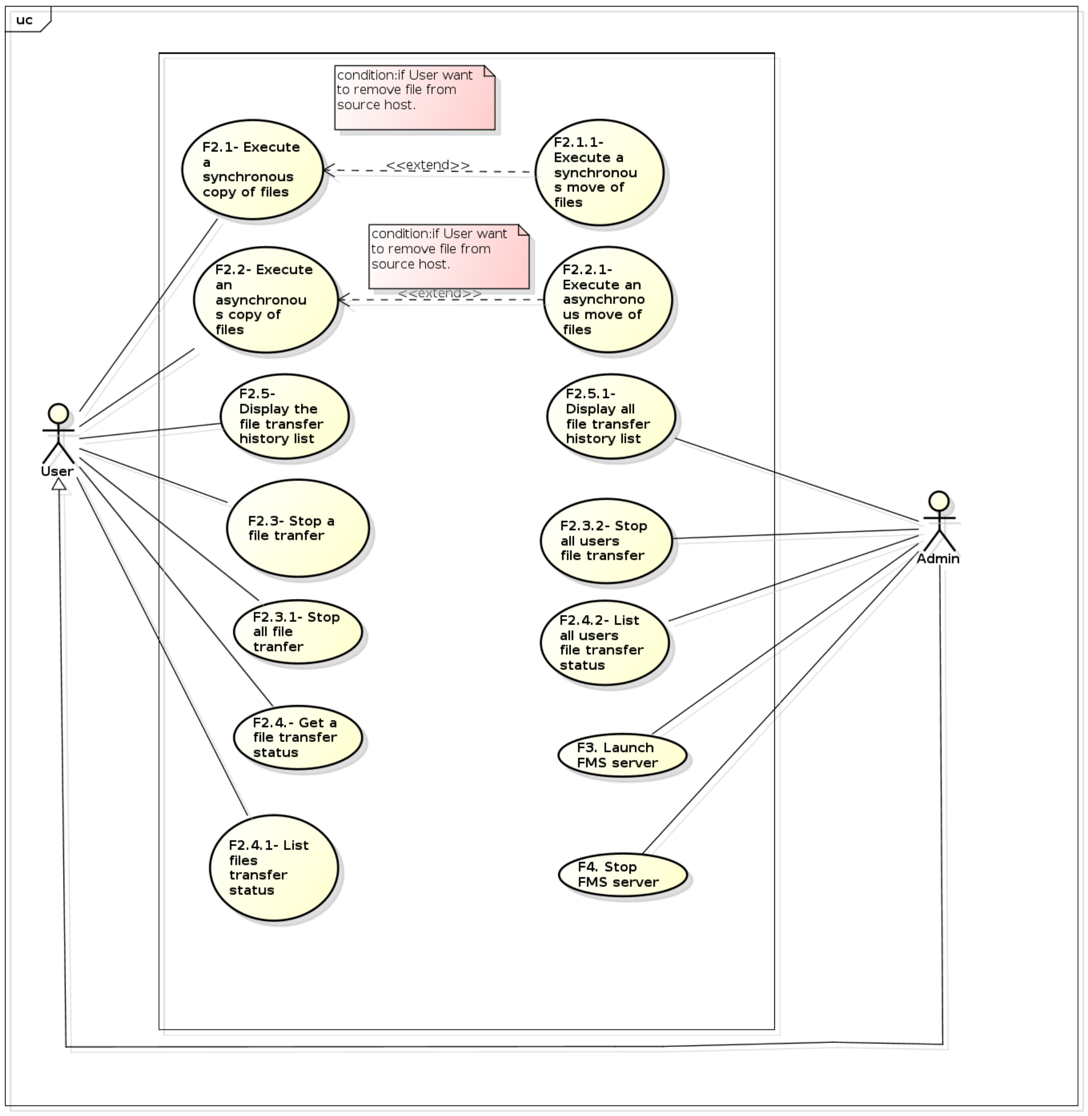
uc
condition:if User want to remove file from source host.
F2.1- Execute a synchronous copy of files
<<extend>>
F2.1.1- Execute a synchronous move of files
condition:if User want to remove file from source host.
F2.2- Execute an asynchronous copy of files
<<extend>>
F2.2.1- Execute an asynchronous move of files
F2.5- Display the file transfer history list
F2.5.1- Display all file transfer history list
User
F2.3- Stop a file tranfer
F2.3.2- Stop all users file transfer
Admin
F2.3.1- Stop all file tranfer
F2.4.2- List all users file transfer status
F2.4.- Get a file transfer status
F3. Launch FMS server
F2.4.1- List files transfer status
F4. Stop FMS server

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 id and the involved host id. 2. The System checks that the session id 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 id 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	F1.2- Change group owner of files F1.1- Change access rights of files F1.3- Create new directories F1.5- Delete directories F1.6- Delete files F1.8- Display contents of files F1.9- Display Head of files F1.7- Display contents of directories F1.10- Display tail of files F1.11- Get information about remote files F1.4- Create new files

5.2.2 F1.1- Change access rights of files

Title	F1.1- Change access rights of files
Summary	This use case allows User to change access rights of a given remote file (or more remote files). It is the equivalent of the "chmod" bash command.
Actors	User
Precondition	
Postcondition	The new access permissions of the specified files are set.
Base sequence	1. User submits the change access rights command with the files, the new access rights to set, the involved hosts. 2. The System sets the new access rights to the files.
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 F1.10- Display tail of files

Title	F1.10- Display tail of files
Summary	This command allows User to print the last few lines of each named remote file. It is the equivalent of the "tail" bash command.
Actors	User
Precondition	
Postcondition	The last lines of the specified files are printed out on the standard output of the client System.
Base sequence	1. User submits the display command with the path of the files to display, the involved hosts. 2. The System displays the last lines of the specified files 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 a 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 a parent directory or read permission on a 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

5.2.4 F1.11- Get information about remote files

Title	F1.11- Get information about remote files
Summary	This use case allows User to get information about each 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 information about given files are printed out on the standard output of the client System.
Base sequence	1. User submits the get information command with the files, the involved hosts. 2. The System prints out the information about the specified files 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 a 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 a 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

5.2.5 F1.2- Change group owner of files

Title	F1.2- Change group owner of files
Summary	This use case allows User to change the group owner of each named remote file. It is the equivalent of the "chgrp" bash command.
Actors	User
Precondition	
Postcondition	The new group owner of the specified files is set.
Base sequence	1. User submits the change group owner command with the files , the new group to set, the involved hosts. 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

5.2.6 F1.3- Create new directories

Title	F1.3- Create new directories
Summary	This use case allows User to create new directories in each named host. It is the equivalent of the "mkdir" bash command.
Actors	User
Precondition	
Postcondition	The new directories are created in the specified host and are owned by User and his group.
Base sequence	1. User submits the create directory command with the paths of directories to create, the involved hosts. 2. The System creates new directories with the specified paths.
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

5.2.7 F1.4- Create new files

Title	F1.4- Create new files
Summary	This use case allows User to create new files in each named host. It is the equivalent of the "touch" bash command.
Actors	User
Precondition	

Postcondition	The new files are created in the specified hosts and are owned by User and his group.
Base sequence	1. User submits the create file command with the paths of files to create, the involved hosts. 2. The System creates new files with the specified paths.
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

5.2.8 F1.5- Delete directories

Title	F1.5- Delete directories
Summary	This use case allows User to remove each 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 directories are removed from the given hosts.
Base sequence	1. User submits the delete directory command with the paths of directories to delete, the involved hosts. 2. The System deletes the specified directories from the hosts.
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 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 a parent directory,or if a 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

5.2.9 F1.6- Delete files

Title	F1.6- Delete files
Summary	This use case allows User to remove each given remote file. It is the equivalent of the "rm" bash command.
Actors	User
Precondition	
Postcondition	All specified files are removed from the hosts.
Base sequence	1. User submits the delete file command with the paths of files to delete, the involved hosts. 2. The System deletes the specified files from the hosts.
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 path is not a file or 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 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

5.2.10 F1.7- Display contents of directories

Title	F1.7- Display contents of directories
Summary	This use case allows User to list the files contained in each given directory located on a remote host. It is the equivalent of the "ls" bash command.
Actors	User
Precondition	
Postcondition	The contents of the specified directories are printed out on the standard output of the client System.
Base sequence	1. User submits the display command with the paths of directories to list, the involved hosts. 2. The System displays the contents of the specified directories 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 a 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 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

5.2.11 F1.8- Display contents of files

Title	F1.8- 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 hosts. 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 a file, a message is also printed on the standard output of the client System.
Extension of	F1- Execute simple command on a remote host

5.2.12 F1.9- Display Head of files

Title	F1.9- Display Head of files
Summary	This command allows User to print the first few lines of each given remote file. It is the equivalent of the "head" bash command.
Actors	User
Precondition	
Postcondition	The first lines of the specified files are printed out on the standard output of the client System.
Base sequence	1. User submits the display command with the paths of the files to display, the involved hosts. 2. The System displays the first lines of the specified files 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 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 write in a parent directory or read permission on a 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

5.2.13 F2.1- Execute a synchronous copy of files

Title	F2.1- Execute a synchronous copy of files
Summary	This use case allows User to copy a file between two hosts. It is the equivalent of the "cp" bash command. This use case allows the transfer of several source files but towards one source destination (which must be a directory).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 is now on the destination 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 to copy (including the hosts), the path of destination (including the hosts) and the session key. 2.The System copies the given source file 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 a source file 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 a list of arguments is provided and the final argument in the sequence is not the name of an existing directory, a message is printend out on the standard output of the ckient 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 a 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	F2.1.1- Execute a synchronous move of files

5.2.14 F2.1.1- Execute a synchronous move of files

Title	F2.1.1- Execute a synchronous move of files
Summary	This use case allows User to copy a file from a host to another host. Furthermore, the source file is removed from the source host. This use case allows the transfer of several source files but towards one source destination (which must be a directory). 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 file source is now on the destination host, - and the source file 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 file 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 files to the specified destination and remove the source files from the source hosts.
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 a source file 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 a list of arguments is provided and the final argument in the sequence is not the name of an existing directory, 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 a 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	F2.1- Execute a synchronous copy of files

5.2.15 F2.2- Execute an asynchronous copy of files

Title	F2.2- Execute an asynchronous copy of files
Summary	This use case allows User to copy files between two hosts and submit another command without waiting the end of transfer file. This use case allows the transfer of several source files but towards one source destination (which must be a directory). 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 transfers are fully accomplished and a copy of the source files 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 paths of the source files 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 a 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 a source file 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 a list of arguments is provided and the final argument in the sequence is not the name of an existing directory, a message is printend out on the standard output of the ckient 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 a 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	F2.2.1- Execute an asynchronous move of files

5.2.16 F2.2.1- Execute an asynchronous move of files

Title	F2.2.1- Execute an asynchronous move of files
Summary	This use case allows User to move files from hosts to another host and submit another command without waiting the end of file transfer. Furthermore, the source files are removed from the source hosts. This use case allows the transfer of several source files but towards one source destination (which must be a directory). 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 transfers are in completed status. - The source files are 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 paths of the source files to copy (including the hosts), the path of destination (including the host) and the session key. 2. The System starts the transfers of the given source files 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 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 - User does not have execute permission in the source or destination file parent, - or he does not have read permission on a 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	F2.2- Execute an asynchronous copy of files

5.2.17 F2.3- Stop a file tranfer

Title	F2.3- Stop a file tranfer
Summary	This use case allows User to stop an asynchronous file transfer he submitted by specifying its id.
Actors	User
Precondition	User has at least an open active session.
Postcondition	- The file transfer whose id is given is stopped. - The log System has been updated and contains request parameters.
Base sequence	1. User submits a stop file transfer command by specifying the session key and by specifying a transfer id . 2. The System stops the transfer file whose id is given.
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. 1.c If the 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 command fails, a message is printed on the standard output of the client System.

5.2.18 F2.3.1- Stop all file tranfer

Title	F2.3.1- Stop all file tranfer
Summary	This use case allows User to stop all file transfer he submitted.
Actors	User, User
Precondition	User has at least an open active session.
Postcondition	- All asynchronous file transfer User submitted is stopped. - The log System has been updated and contains request parameters.
Base sequence	1. User submits a stop file transfer command by specifying the session key . 2. The System stops all asynchronous file transfer User submitted .
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 or if User did not submit one of named file tranfers, a message is printed out on the standard output of the client System. 1c. If the command fails, a message is printed out on the standard output of the client System.

5.2.19 F2.3.2- Stop all users file transfer

Title	F2.3.2- Stop all users file transfer
Summary	This use case allows Admin to stop all current asynchronous file transfer of a given session.
Actors	Admin, Admin
Precondition	Admin has at least an open active session.
Postcondition	- All file transfer submitted is stopped. - The log System has been updated and contains request parameters.
Base sequence	1. Admin submits the stop file transfer command by specifying the session key. 2. The System stops all transfer file of the given session.
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 command fails, a message is printed on the standard output of the client System.

5.2.20 F2.4.- Get a file transfer status

Title	F2.4.- Get a file transfer status
Summary	This use case allows User to get the status of each given asynchronous file transfers he submitted. Four main status are defined for a file tranfer: - in Progress: the file tranfer is on-going - completed: the file transfer is completed - cancelled: the file transfer is cancelled - failed : the file transfer failed.
Actors	User
Precondition	User has at least an open active session.
Postcondition	- The status of the specified file transfers is displayed on the standard output of client System.
Base sequence	1. User submits a get file transfer command by specifying a session key and the transfer identifiers. 2. The System displays the status of all specified file transfers .
Branch sequence	2a. Futhermore, the System will display the progression of all in Progress file transfers. But that information will depend on the process used by the file transfer.
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 a specified transfer id is invalid or User did not submit a named file tranfer, a message is printed out on the standard output of the client System. 1d. If the command fails, a message is printed out on the standard output of the client System.

5.2.21 F2.4.1- List files transfer status

Title	F2.4.1- List files transfer status
Summary	This use case allows User to list all file transfer status he submitted.
Actors	User, User
Precondition	User has at least an open active session.
Postcondition	- The status of all file transfer User submitted are listed on the standard output of client System.
Base sequence	1. User submits a list file transfer command by specifying a session key. 2. The System displays the status of all file transfer (including current and completed file transfer) User submitted.

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 no transfer was submitted or if the command fails, a message is printed out on the standard output of the client System.
Extensions	F2.4.2- List all users file transfer status

5.2.22 F2.4.2- List all users file transfer status

Title	F2.4.2- List all users file transfer status
Summary	This use case allows Admin to list all file transfer status of a given session.
Actors	Admin, Admin
Precondition	Admin has at least an open active session.
Postcondition	-A ll file transfer status of a given session are listed on the standard output of client System. - The log System has been updated and contains request parameters.
Base sequence	1. Admin submits a list file transfer status command by specifying a session key. 2. The System displays all file transfer status (including current and completed file transfer) of a named session.
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 no transfer was submitted or if the command fails, a message is printed out on the standard output of the client System.
Extension of	F2.4.1- List files transfer status

5.2.23 F2.5- Display the file transfer history list

Title	F2.5- Display the file transfer history list
Summary	This use case allows User to list all file transfer he submitted. User can specify an optinal search criteria: - status - source host or destination host.
Actors	User
Precondition	User has at least an open active session.
Postcondition	- All file transfer User submitted are listed on the standard output of client System. - The log System has been updated and contains request parameters.
Base sequence	1. User submits a display file transfer history command by specifying a session key. 2. The System displays all file transfer User submitted on the standard output of 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 given session key is invalid, a message is printed on the standard output of the client System. 1c. If no transfer was submitted or if the command fails, a message is printed on the standard output of the client System.

5.2.24 F2.5.1-Display all file transfer history list

Title	F2.5.1-Display all file transfer history list
Summary	This use case allows Admin to list all file transfer of a given session. Admin can specify an optinal search criteria: - status - source host or destination host.
Actors	Admin, Admin
Precondition	User has at least an open active session.
Postcondition	- All file transfer of a named session are listed on the standard output of client System. - The log System has been updated and contains request parameters.
Base sequence	1. Admin submits a dispay file transfer history command by specifying a session key. 2. The System displays all file transfer of the named session on the standard output of 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 given session key is invalid, a message is printed out on the standard output of the client System. 1c. If no transfer was submitted or if the command fails, a message is printed out on the standard output of the client System.

5.2.25 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 on 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. The batch scheduler does not recognize some job ids. In this case the System updates the job status to completed.

5.2.26 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

- FMS: File Management System
- **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.