VOSpace Backend
a storage management solution
interfaced with CADC VOSpace

S.Bertocco

P.Dowler, S.Gaudet, B.Major, F.Pasian, G.Taffoni

The problem

The IVOA VOSpace recommendation says:
"A VOSpace web service is an access point for a distributed storage network. Through this access point, a client can:
- add or delete data objects
- manipulate metadata for the data object
- obtain URIs through which the content of the data objects can be accessed

VOSpace does not define how the data is stored or transferred, but only the control messages to gain access. Thus, the VOSpace interface can readily be added to an existing storage system."

VOSpace

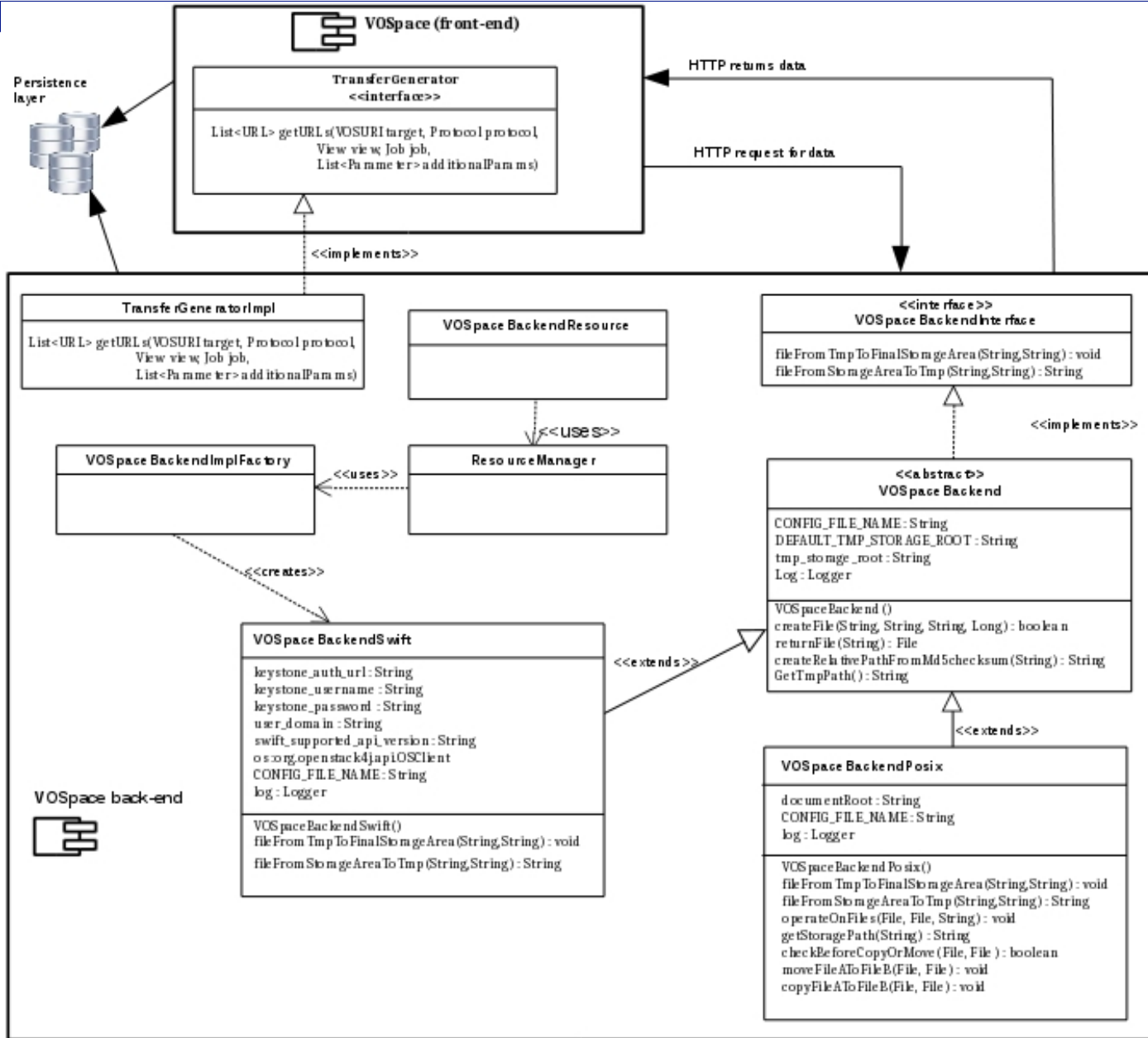A storage service needs three components:

- **VOSpace interface**, responsible of the meta-data management, implementing the IVOA recommendation

- **data transfer service**, responsible for handling the information on how to upload and download data

- **vospace-backend**, to interface the transfer service with a concrete storage solution. It is a storage service management to connect a VOSpace interface implementation with a specific storage solution. It manages the physical storage layer and points to the data locations
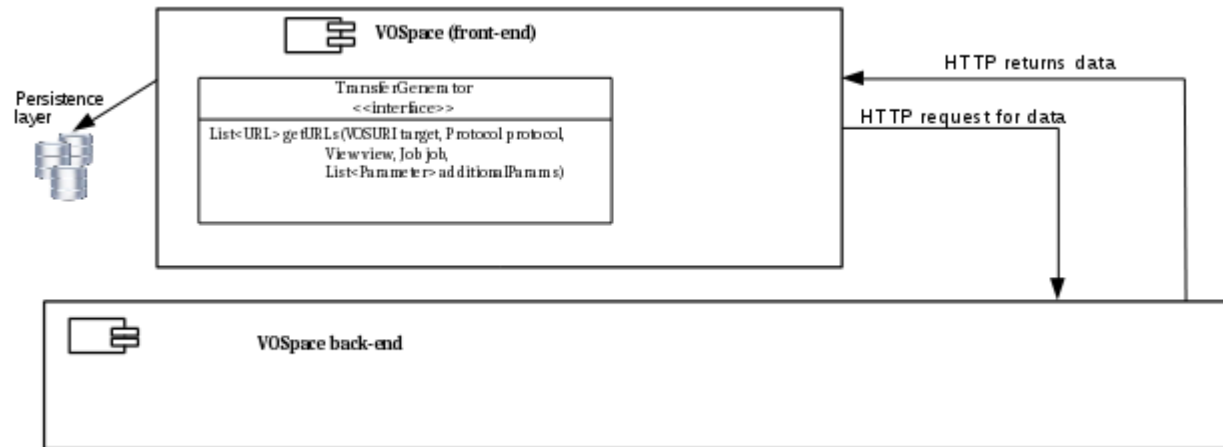
CADC VOSpace

- VOSpace interface → CADC implemented

  https://github.com/opencadc/vos

- data transfer service → CADC implemented

  https://github.com/opencadc/vos

- vospace-backend → OATs-INAF implemented
  https://github.com/oats-cadc/oats-vospace-backend
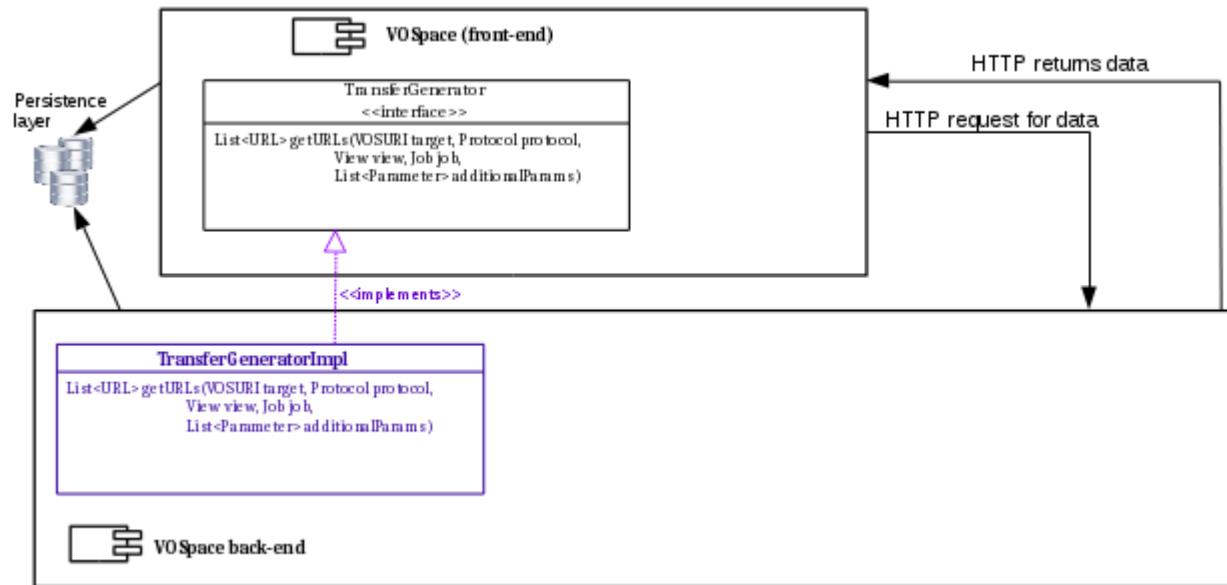
Architecture

Front-end

The front-end defines an interface to retrieve information about URL to query to retrieve data
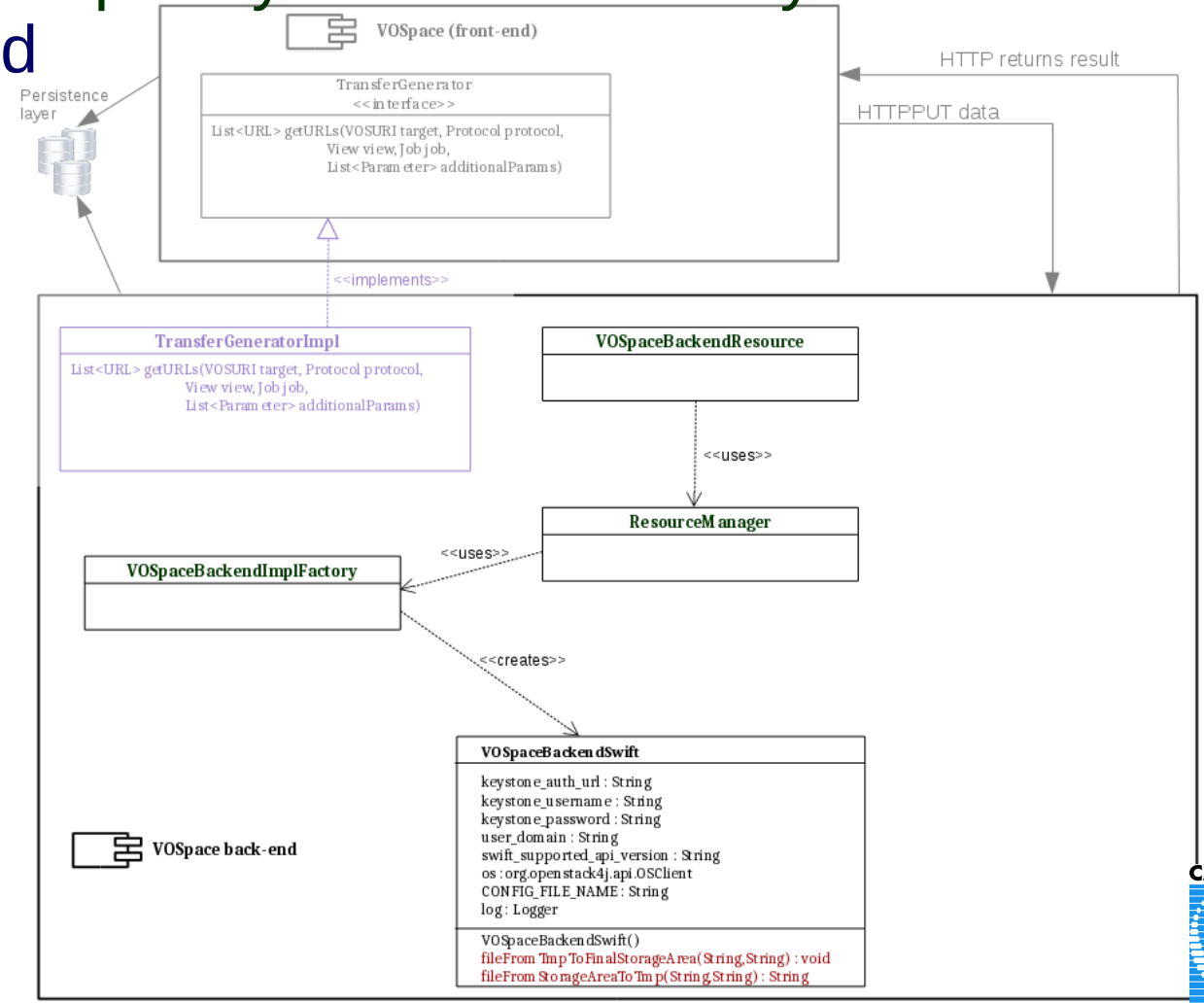
Transfer generator

The back-end implements the TrasferGenerator interface providing the URLs to query to put and retrieve data

Service resource

- The back-end resource receives the data
- Saves the data in a temporary area in the file system
- Updates the front-end stored metadata (e.g. the MD5 checksum)
- Istantiate the storage plug-in configured in a configuration file

Plug-ins mechanism

- The back-end istantiate the storage plug-in configured in a configuration file
- Uses the plug-in to store the file (move from tmp area to the final storage area)
- Saves in the DataBase the relation between Node and data

Upload operation (PUT)
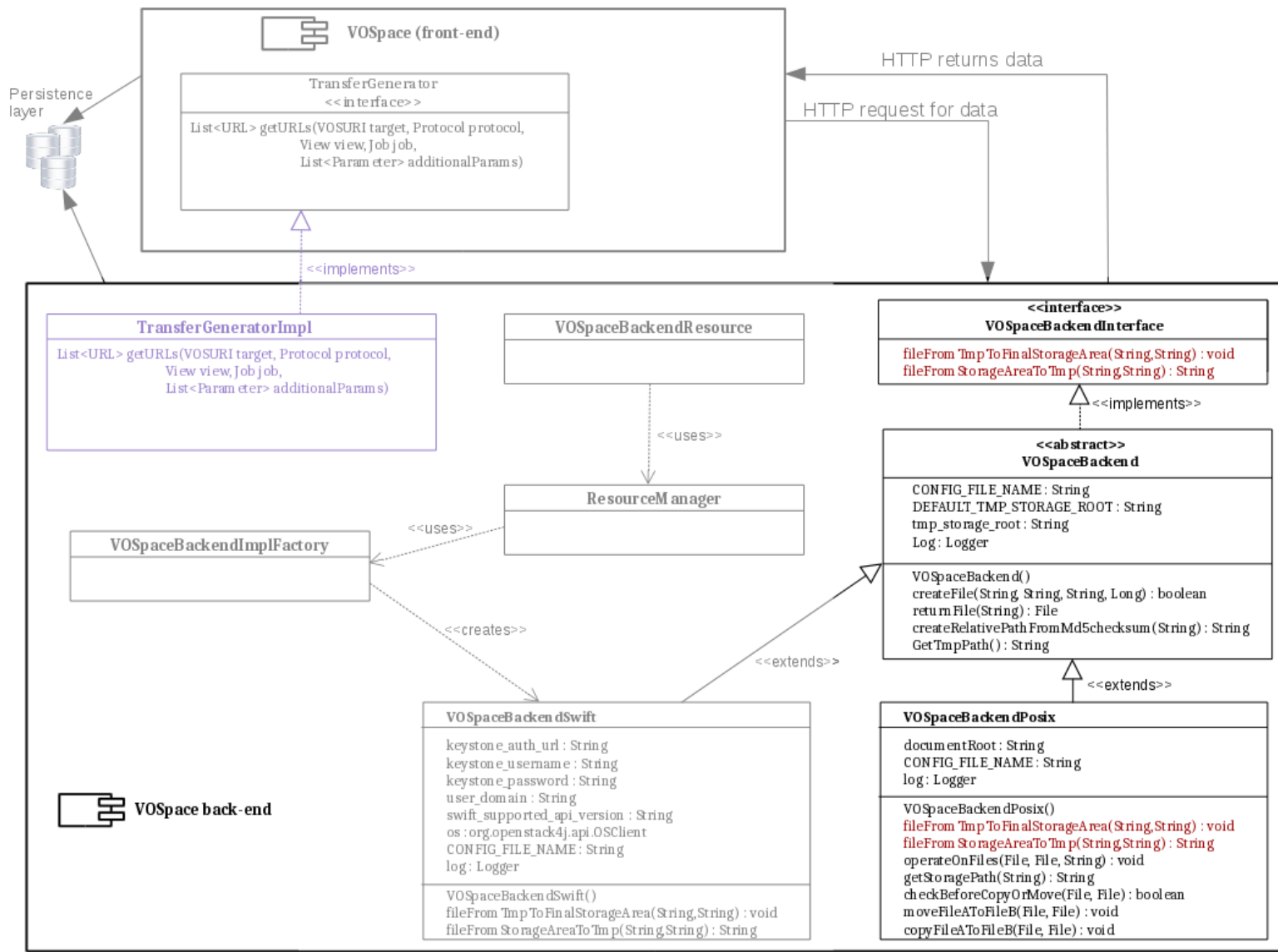
- The back-end resource receives the data
- Saves the data in a temporary area in the file system
- Updates the front-end stored metadata (e.g. the MD5 checksum)
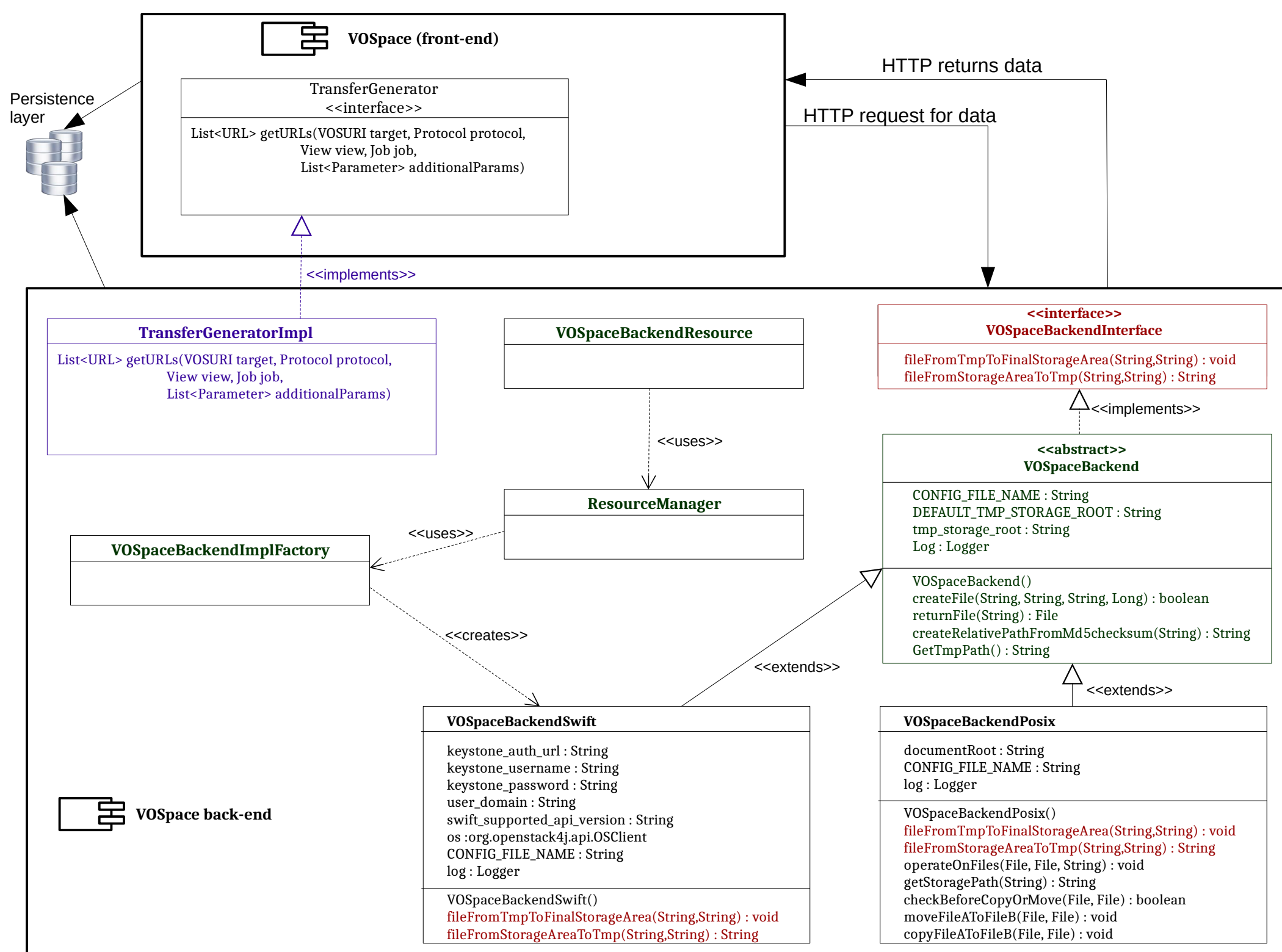- Istantiate the storage plug-in configured in a configuration file
- Uses the plug-in to store the file (move from tmp area to the final storage area)
- Saves in the DataBase the relation between Node and data

VOSpace (front-end)

TransferGenerator
<<interface>>

List<URL> getURLs(VOSURI target, Protocol protocol,
View view, Job job,
List<Parameter> additionalParams)

Persistence
layer

HTTP returns data

HTTP request for data

<<implements>>

TransferGeneratorImpl

List<URL> getURLs(VOSURI target, Protocol protocol,
View view, Job job,
List<Parameter> additionalParams)

VOSpaceBackendResource

<<interface>>
VOSpaceBackendInterface

fileFromTmpToFinalStorageArea(String,String) : void
fileFromStorageAreaToTmp(String,String) : String

<<implements>>

<<uses>>

ResourceManager

<>
VOSpaceBackend

CONFIG_FILE_NAME : String
DEFAULT_TMP_STORAGE_ROOT : String
tmp_storage_root : String
Log : Logger

VOSpaceBackend()
createFile(String, String, String, Long) : boolean
returnFile(String) : File
createRelativePathFromMd5checksum(String) : String
GetTmpPath() : String

<<uses>>

VOSpaceBackendImplFactory

<<creates>>

<<extends>>

VOSpace back-end

VOSpaceBackendSwift

keystone_auth_url : String
keystone_username : String
keystone_password : String
user_domain : String
swift_supported_api_version : String
os :org.openstack4j.api.OSClient
CONFIG_FILE_NAME : String
log : Logger

VOSpaceBackendSwift()
fileFromTmpToFinalStorageArea(String,String) : void
fileFromStorageAreaToTmp(String,String) : String

<<extends>>

VOSpaceBackendPosix

documentRoot : String
CONFIG_FILE_NAME : String
log : Logger

VOSpaceBackendPosix()
fileFromTmpToFinalStorageArea(String,String) : void
fileFromStorageAreaToTmp(String,String) : String
operateOnFiles(File, File, String) : void
getStoragePath(String) : String
checkBeforeCopyOrMove(File, File) : boolean
moveFileAToFileB(File, File) : void
copyFileAToFileB(File, File) : void

Documentation

Developers guide

Questions

?

Contact: sara.bertocco@oats.inaf.it