INFRASTRUCTURE TEST-MACHINES – CONTACT

Current situation

The testmachines are having related testmachine-pc on which the testresults/raw-data coming from the testmachine during/after the test-process are being stored. When processing of the testmachine is ready the next step will be to do the necessary calculations/transformations on this data and create the output-datafiles and testresults in a unilink-file.

The naming of the testresults/datafiles are based on an UNILAB IMPORT-ID related to a sample/test-method already available/created in UNILAB. This IMPORT-ID needs to be converted to the related sample before sending the unilink-file to UNILAB.

The calculated/transformed testresults must match all the existing testmethod-cells. These test-variables are being configured once.

The (raw) datafiles are sometimes stored on a network-share or are being stored in the UNILAB-database als BLOB-file. The location of the file is being stored in UNILAB in relation to the sample.

The created unilink-file is send via a network-share to the LimsClient, which is placing this file via another network-share on the UNILAB-database-server. On the UNILAB-database-server a unilink process is running which processes all the incoming files and executes all necessary insert/update in UNILAB.

CONTACT-INTERFACE

At first there is a choice to make where we want to implement the Services doing all the calculations/transformations on the rawdata-files coming from the testmachines.   
1)We can implement the way we do it now,within Apollo itself  
2)We can migrate all the functionality to the CONTACT-system

Preferred solution from CONTACT point of view:  
Let specific calculations (like matlab) outside the CONTACT-system. This sounds logical. Functional Expertise is only available with the employees working at the testmachines.

The Infrastructure doesn’t become easier. We still need a testmachine-pc related to the testmachine, for storing all the rawdata-/files created by the testmachines, there are still have to run python-scripts for picking up these ready-for-execution-files to pass to the next step of the process.

When we choose for leaving the Calculation/Transformation-services within Apollo then there are 2 possible architectural solutions to choose from:  
1)Each testmachine/pc sends do its own calculations/transformations and sends their own results/datafiles to CONTACT-system directly  
2)Each testmachine/pc does its own calculations/transformations and passes their results/datafiles to a central server which sends the results/datafiles to CONTACT-system.

The preferred architectural-solution CONTACT-LIMS interfacing is in my opinion Option 2)

In this way we create a hybrid solution were we are running Python-software on the testmachine-pc’s for the specific calculations/transformations on the raw datafiles coming from the related testmachine. The testresults and output-datafiles are stored via a network-share on a local-directory of a central communicating-server (“LimsClient”) to CONTACT-system.

This solution is   
-proven technology: this structure has been working for last 10 years or more;  
-flexibel: we can add/remove/change testmachines or configurations by ourselves;  
-secure: we can store/maintain credentials on only a single server. Not on all testmachine-pc anymore.  
-easy maintenance: direct contact and functional expertises available with the employees working with the testmachine. No relationships with other testmachines and interfaces.  
-less risk to migrate: Migration from the current LIMS to new CONTACT-system will be of less risk. Calculation/Transformation-procedures doesn’t have te be rebuild/tested in new environment.   
With small changes it might even be possible to run in paralle for a while.

**Advantages:**   
-Apollo has flexibility to add new or change current testmachines  
-Calculations/transformations are testmachine-specific and differ per testmachine. Calculations are being done on related testmachine-pc using python-scripts. This python-service only needs access to local directories within apollo-network.   
-Employees working at the testmachine are responsible for correct configuration/processing of the local process. They are able to fix issues when doing the calculations/transformations, and may be restart the local-process after solving the issue.   
-Collecting and sending the unilink-files and related attachments are send to Contact from a single server within the Apollo network (the current so called “LimsClient”). Best solution from maintenance/security-perspective.   
We only need to store credentials on this machine to call services from Contact. Server/credentials are maintained only by admin-user and are not distributed along all the testmachine-pc through the organisation.

-In this solution Apollo has the opportunity to implement a second central server (“LimsClient”) for sending testresults/datafiles coming from a testmachine-pc to a CONTACT-TEST or EDUCATION-environment or implementing a switch (like in the current situation) to credentials for the test-environment based on the content of the unilink-file.

Architectural Infrastructure of LIMS-CONTACT-interface

“LIMSCLIENT”

LIMS-IMPORT-TESTRESULTS-SERVICE

TestMachine-PC

CONTACT

LIMS-IMPORT-XML-SERVICE

UNILINK-import.xml

CIM-DB

S3-STORAGE

TestMachine-PC

TestMachine-2

CONTACT-XML-DATA-SERVICE

CONTACT-FILE-DATA-SERVICE

rawdata.txt

calculations.txt

CALC-TRANSFORM-SERVICE

rawdata.txt

results.txt

Based on IMPORT-ID

-Calculations  
-Transformations

**Alternative-2: Lims stores Files**  
-Store-Files to S3-Storage  
-Retrieve-URL

Alt-2:LIMS-IMPORT-FILE-SERVICE

rawdata.txt

results.txt

UNILINK-Import-INCL-FILENAME.xml

Based on IMPORT-ID

rawdata.txt

results.txt

CONTACT-RETRIEVE-SC-SERVICE

Sample-info.xml

-Store-Data in CONTACT incl. file-url-in S3-storage (FTP?)

-Store-Data + Files in CONTACT

TESTVARIABLES related to METHOD-CELLS

TESTVARIABLES.XML

NO Direct Interface to CONTACT

Only SERVICE with DIRECT Interfaces to CONTACT

How do we get TEST-METHOD (in current situation it comes from UNILAB in this stage based on IMPORT-ID

New-Services in Service-Layer to handle testResults

LIMS-RETRIEVE-SC-SERVICE

rawdata.txt

results.txt

Alt-1:IMPORT-FILE-SERVICE

rawdata.txt

results.txt

**Alternative-1: Contact stores Files**-Store-Files to S3-Storage  
-Retrieve-URL

UNILINK-Import-INCL-FILENAME.xml

UNILINK-import.xml

TestMachine-1

rawdata.txt

calculations.txt

CALC-TRANSFORM-SERVICE

rawdata.txt

results.txt

TESTVARIABLES.XML

**SAP-PO ??**

-RETRIEVE INFO related to IMPORT-ID

VPN/LPG: AWS-CLOUD

Network-share

**Choices to make:**

-Does SAP-PO play a role in the communication between LimsClient and CONTACT-system? Or can we use CONTACT-services directly? Can SAP-PO deal with BLOB/CLOB-attachements?   
I wonder if SAP-PO is capable of meeting our requirements on the interfaces.  
Infrastructure is getting more complex, with an extra point of failure !

-Can we solve the CONTACT-interface with 1 service-call for the testresults and all the (rawdata-)files as attachment of do we need to upload the files from within the “LimsClient” to S3-Storage first before we call the CONTACT-service for adding the unilink-testresults? These 2 processes needs to be related to each-other because the datafile-location needs to be stored within the sample-results in CONTACT.  
We prefer the solution to send XML and files within 1 service-call.

**Technical Interfaces Requirements:**

-Interface from LIMS to CONTACT must be synchronous. When message is delivered then the process for LIMSCLIENT is completed.

-Persistant: Message/Source-Files once deliverd to Contact may never be lost, and always be kept save/persisted in a DB-queue, Message on the queue must be executed one by one,to process it in CONTACT-CIM-db.   
If anything goes wrong then Message stays on the queue (may be on error-queue) from where the message can be processed again when underlying issue is solved.

-Processing Messages from the queue in CONTACT-system are A-synchronous in background process.

**Remarks/Questions**

-How are we dealing with locking of the samples and testresults. User can work on the sample online in the contact-system, when in parallel the message-service is trying to import test-results. Concurrent session can block or overwrite each-other. Is there a mechanism which can avoid this?

-Some of the Testmachine-Interfaces are new and rebuild with Python-scripts. There still be testmachine-pc’s doing the calculations/transoformations in DELPHI. We doesn’t have an overview of all the current remaining interfaces running on DELPHI still. We need to make an inventory of these intefaces and make a plan how to migrate these to Python.