Technical Design General Architecture

Version: 1.1
Author : P:Negro

Document history

Version	*Date*	*Status*	*Author(s):*	*Description of changes*
0.1	06-09-2014	Concept	P Negro	Initialisation of the document
1.0	14-10-2014		P NEgro	Brought up To Date
1.1	30-03-2015		P NEgro	Brought up To Date

Document references

No.	*Document name*	*Author(s):*	*Version*	*Date*

Table of Contents

1 Context and Objective of the BLUEWAY platform

This document aims to provide developers and software administrators with insight into the technology behind the setup of BLUEWAY V5.6 in replacement of COVAST.

2 Project SCOPE

Cargonaut message broker Covast has reached its end of life, the current version used by Cargonaut is barely supported by the Software Editor Covast and the source have been sold to Microsoft and are now fully integrated in BizTalk server since the 2010 version.
Cargonaut decided to replace Covast by a more recent product that may accompany and ease grows of Cargonaut business for the coming years.
Cargonaut wish that:

- Covast should be replaced by an event driven system;
- The new system should be re-usable by other CIN;
- The new system can operate on Windows or Unix/Linux OS;
- It should be possible to phase out the implementation of that new tool.

3 Requirements

Refer to document *COVAST REPLACEMENT Requirements 0.1 MO15102012*

4 Actors

4.1 Cargonaut

Xxxx (IT Manager)
Marcel Olie (Functional analyst) Marcel.Olie@cargonaut.nl
Rene Willems (Functional analyst)
Barry Mesrits (System Administator)
Eric Meier (Customer Service)

4.2 Districon

Hans van Roest (Project Manager) Hans.vanRoest@cargonaut.nl

4.3 BLUEWAY

Daniel Coya (CEO) dcoya@Blueway.fr
Cyrille Preaux (CTO) cpreaux@blueway.fr

4.4 Cogiflow B.V

Philippe Negro (CEO) P.Negro@cogiflow.nl

Sijmen Brakenhoff (Software Engineer) S.Brakenhoff@cogiflow.nl
Bas Lammers (Software Engineer) B.Lammers@cogiflow.nl

4.5 Group & Authorization

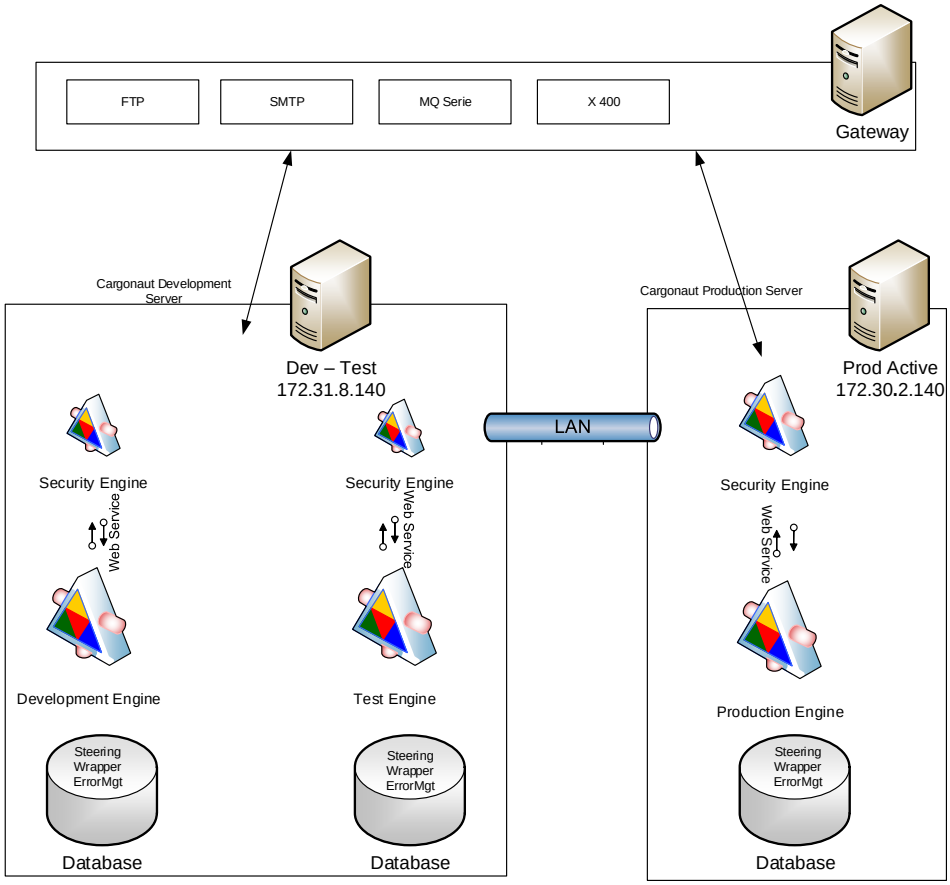
Administration
Development
Tests
CustomerService ?
Management ?

5 Development constraints

- Development are done on basis of a technical details Specification document validated by Cargonaut Core Team.
- All development are done using BLUEWAY Designer and the available instructions and functions. In the case it would be necessary to implement a new function, it will be done in BLUEWAY designer by adding a "support function". This function will be reviewed by BLUEWAY to make sure it is compatible with other existing function and will not trigger side effects. Once reviewed and accepted it will be included in the standard function of the designer.
- Developers have to respect the naming conventions defined in paragraph 6.3
-

6 Architectural model

6.1 Diagram



Security Engine is the entry point for external world and the only part of BLUEWAY accessing the Gateway. Security engine cannot access directly the Databases it should be done via the Dev /Test/ Prod Engine. This way BLUEWAY application is isolated from "external world" and will process only authorized data.

Development engine is used by Cogiflow and Cargonaut to implement the necessary service components.

Test engine is used by Cargonaut to validate and accept the delivered service components.

Production Engine runs the accepted version of the service components.

6.2 Software and Hardware

Software : BLUEWAY V 5.6

Database : My Sql 5.5.38-0

Server : At Cargonaut: Linux Debian Wheezy

Dev Engine uses port 8180 on server 172.31.8.140 (`jnp://localhost:1399`)

Test Engine uses port 8280 on server 172.31.8.140 (`jnp://localhost:1499`)

Security Dev uses port 8380 on server 172.31.8.140 (`jnp://localhost:1599`)

Security Test uses port 8480 on server 172.31.8.140 (`jnp://localhost:1699`)

Production engine uses port 8180 on server 172.30.2.140 (`jnp://localhost:1399`)

Security Production uses port 8280 on server 172.30.8.140 (`jnp://localhost:1499`)

6.3 Naming Convention

Refer to document *Design policy and naming convention 0.01 M20052014 revPNO*

6.4 Process description

The whole process is defined as a workflow chaining service components.

Messages are sent by Cargonaut customers to the Gateway. BLUEWAY Security layer will get the files from the Gateway and validate authorization rules. This process is described as "Service Component 1" in related TD.

If authorization are matched the file will be transmitted to the BLUEWAY Application Service Component 2 where the Message type is recognized and the combination Sender / Receiver / Message Type is checked.

If all validation rules are met the message will be processed by Service Component 3 where the document structure and content validity are controlled.

If the document is valid, it is transmitted to Service Component 4 to be mapped to the pivot format. Pivot Format consist in an XML representation of a message type. There will be a unique Pivot for all version of the FSA for example but due to the structure changes in between FWB/3 and FWB/9 there will be 2 Pivots definition for FWB.

After conversion to pivot Format Service Component 5 will determine what as to be done with the file (Routing to one or many parties, backend)

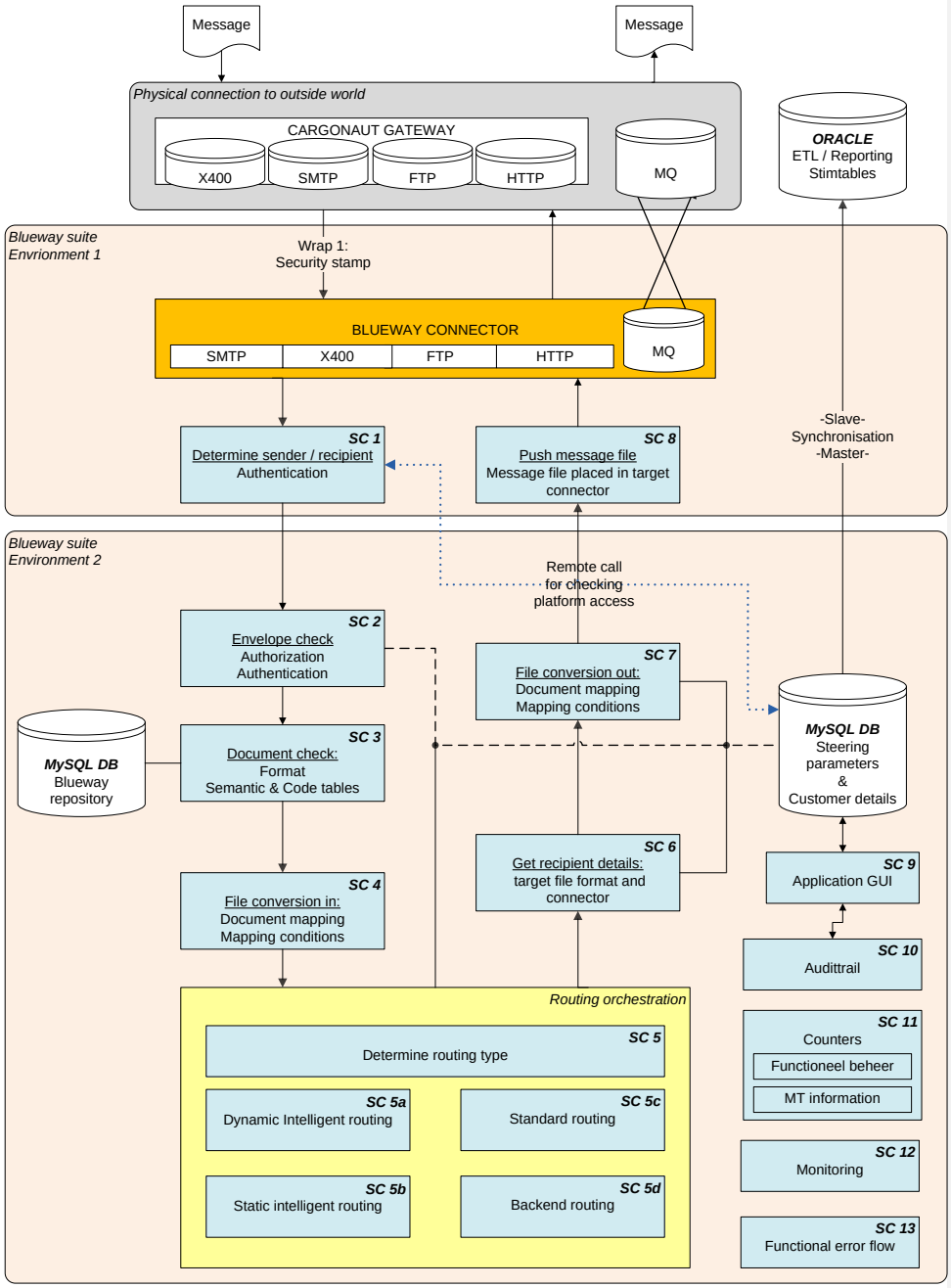
When the next steps of the process are defined Service Component 6 will gather all necessary information to proceed to the delivery for each outgoing flow:

- Outgoing address
- Outgoing format
- Outgoing protocol

For each outgoing flow Service Component 7 will be triggered with the appropriate input in order to convert the pivot document into the outgoing format and send it to the security layer.

The Service component 8 executed on the security layer will proceed and deliver the document to the recipient.

All processes can be followed up and managed via a web portal.


Message
Message
Physical connection to outside world
CARGONAUT GATEWAY
X400
SMTP
FTP
HTTP
MQ
ORACLE
ETL / Reporting
Stimtables
Blueway suite
Envrionment 1
Wrap 1:
Security stamp
BLUEWAY CONNECTOR
SMTP
X400
FTP
HTTP
MQ
SC 1
Determine sender / recipient
Authentication
SC 8
Push message file
Message file placed in target connector
-Slave-
Synchronisation
-Master-
Blueway suite
Environment 2
Remote call
for checking
platform access
SC 2
Envelope check
Authorization
Authentication
SC 7
File conversion out:
Document mapping
Mapping conditions
MySQL DB
Blueway
repository
SC 3
Document check:
Format
Semantic & Code tables
MySQL DB
Steering
parameters
&
Customer details
SC 6
Get recipient details:
target file format and
connector
SC 9
Application GUI
SC 4
File conversion in:
Document mapping
Mapping conditions
SC 10
Audittrail
Routing orchestration
SC 5
Determine routing type
SC 11
Counters
Functioneel beheer
MT information
SC 5a
Dynamic Intelligent routing
SC 5c
Standard routing
SC 12
Monitoring
SC 5b
Static intelligent routing
SC 5d
Backend routing
SC 13
Functional error flow

7 Technical data model

7.1 Steering Database

Refer to document BWY Steering model 2.0 RWMO25022015.docx

7.2 Wrapper Database

Refer to document BWY Steering model 2.0 RWMO25022015.docx

7.3 Security Layer Steering

This database is accessible by the Security Layer and should be placed on the same server as the Security Layer engine.

It contains following table: bwy_type_recognition

This table is used by Service component 1 to determine the message type of the inbound message.

This table is filled in synch with bwy_steering.bwy_type_recognition.

7.4 Sequence Database

This database is accessible by the Security Layer and should be placed on the same server as the Security Layer engine.

The database provide the necessary sequence numbering as required in the process.

sequence_name	sequence_increment	sequence_min_value	sequence_max_value	sequence_cur_value	sequence_cycle
Interchange	1	1	99.999.999.999.999	9397	1
Message_ID	1	1	99.999.999.999.999	9756	1
Security_IN_ID	1	1	18.446.744.073.709.500.000	7	1
Bus_Info_ID	1	1	18.446.744.073.709.500.000	63	1

Getting the next value from the sequence is done with the following SQL statement:

SELECT nextval('sequence_name') as next_sequence;

7.4.1 Table descriptions

Use Heidy SQL or any similar tool to see table definition

#	Naam	Datatype	Lengte/Set	Unsign...	NULL toestaan	Met ...	Standaard
1	**sequence_name**	VARCHAR	**100**	☐	☐	☐	Geen default
2	sequence_increment	INT	11	☑	☐	☐	1
3	sequence_min_value	INT	11	☑	☐	☐	1
4	sequence_max_value	BIGINT	20	☑	☐	☐	1844674407370955...
5	sequence_cur_value	BIGINT	20	☑	☑	☐	1
6	sequence_cycle	TINYINT	1	☐	☐	☐	0

7.4.2 Procedure description

Name : nextval

```
BEGIN
DECLARE cur_val bigint(20);

SELECT
sequence_cur_value INTO cur_val
FROM
sequence.sequence_data
WHERE
sequence_name = seq_name
;

IF cur_val IS NOT NULL THEN
UPDATE
sequence.sequence_data
SET
sequence_cur_value = IF (
(sequence_cur_value + sequence_increment) > sequence_max_value,
IF (
sequence_cycle = TRUE,
sequence_min_value,
NULL
),
sequence_cur_value + sequence_increment
)
WHERE
```

```
        sequence_name = seq_name
        ;
        END IF;

        RETURN cur_val;
        END
```

8 Services

8.1 Common Services

8.1.1 SCALL_BWY_SVC_Insert_MessageFileStatus

Function: Fill the message status table with info contained in call parameters, this service is called by others to set the status of a processed message.

Input parameters: SCALL_ALL_MEM_MessageFileStatus

Output parameters: none

Tables used: CGN_Wrapper.bwy_message_file_status

8.1.2 SCALL_BWY_SVC_Insert_WorkflowStatus

Function: Fill the Workflow status table with info contained in call parameters, this service is called by others to set the status of a processed message.

Input parameters: SCALL_ALL_MEM_WorkflowStatus

Output parameters: None

Tables used:

8.2 Security Layer Services

8.2.1 SC1_SLR_SVC_Get_FTPReceiveFromGateway

Function: Get message files from the gateway and forward it to next step (4.2.2)

Input parameters: Trigger

Output parameters: MEM_COM.SC1_SLR_MEM_ReceivedFile

8.2.2 SC1_SLR_SVC_Get_NormalizeReceivedFile

Function: Assign a unique ID to the message and log the incoming message

Input parameters: MEM_COM.SC1_SLR_MEM_ReceivedFile

Action called:
- Call webService_SC1_BWY_Insert_Security_In
- Call webService_SC1_BWY_SVC_Insert_ArchiveShort
- Call webService_SC1_BWY_SVC_Insert_BusInfo – 1
- Call webService_SC1_BWY_SVC_Insert_WorkflowStatus

Output parameters: MEM_COM.SC1_SLR_MEM_ReceivedFile

8.2.3 SC1_SLR_SVC_Check_SenderRecipient

Function: Validate Sender and Consignee of the message are known. If both sender and consignee are known then allow to continue and call SC2 otherwise stop the workflow and log the error.

Input parameters:
- MEM_COM.SC1_SLR_MEM_ReceivedFile
- SC1_BWY_SVC_Check_UserOUT.SC1_SLR_MEM_CheckUser_Out

Action called: Call webService SC1_BWY_SVC_Check_User

Output parameters:
- SC1_BWY_SVC_Check_UserIN.SC1_SLR_MEM_CheckUser
- MEM_COM.SC1_SLR_MEM_ReceivedFile

8.2.4 SC1_SLR_SVC_Call_SC2

Function: End the process on the security layer and allow to continue on BLUEWAY layer with SC2.

Input parameters:
- MEM_COM.SC1_SLR_MEM_ReceivedFile

Action called: Call Interface SC2_BWY_SVC_ReceiveFromSC1

Output parameters:
- MEM_COM.SC1_SLR_MEM_ReceivedFile

9 Error Management

9.1 Description of Error Management

9.2 Introduction

Error Management Service Component aimed to proceed both technical and functional errors in a controlled way. All error code generated by the platform will be translated into understandable text. The action to be executed in case of error will be configurable.

9.3 Functional description

The error management Service Component (SCErr) is deployed on the BLUEWAY Layer only.
Security layer will report error by calling SCErr as a web service. Security layer will have one small error handling services hardcoded to manage the case an error is raised by the web service call.

SCErr has following input parameters:
- SC_Origine (SC who raised the error)
- Service_Origine
- Instruction_Origine
- Error Code
- Err description

9.4 Infrastructure

Each error will be logged in a database, based on the information gathered from the type and action databases.

9.5 Database interaction

The following databases and tables will be used in the Error Management.

- error DB
  - bwy_Error_Type
  - bwy_Error_Action
  - bwy_Error_Log

9.5.1 Error_Type

The Error_Type table lists all the type of errors, which are known by blueway and can be managed by the error management module. If the type is unknown, a technical message will be send to user(s).

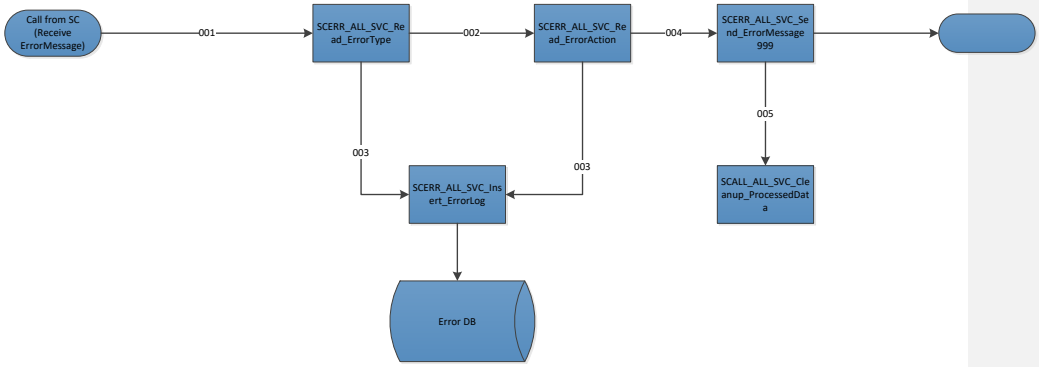
9.5.2 Error_Action

The Error_Action table lists all the action that can be taken, when an error occurs. It determines which action has to be taken, based on the type of error. The user(s) receive a message which explains the error and the action that has to be taken.

9.5.3 Error_Log

Based on the information Blueway receives from the Error_Type and Error_Action DB's, a message will be logged in the Error_Log DB.

*9.6 **Service** overview*



9.6.1 Call from Service Component (Receive Error Message)

For each error that has been detected in one of the service components an error message will be send to this error management service.

- Triggered by message from Service Components
- Send bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 001

9.6.2 SCERR_ALL_SVC_Read_ErrorType

- Triggered by bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 001
- Read table Error_Type to determine type of error
- Send bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 002
- Send bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 003

Fields	Data
Type_ID	
Type	

9.6.3 SCERR_ALL_SVC_Read_ErrorAction

- Triggered by bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 002
- Read table Error_Action to determine which action has to be taken by type of error
- Send bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 003
- Send bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 004

Fields	Data
Action_ID	
Action	

9.6.4 SCERR_ALL_SVC_Insert_ErrorLog

- Triggered by bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 003
- Insert the bus message data in the blueway_error. bwy_Error_Log table

Fields	Data
ID	
Action_ID	
Action	
Type_ID	
Type	

9.6.5 SCERR_ALL_SVC_Send_ErrorMessage

- Triggered by bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 004
- Assign parameters
    - Error_Number
    - Component
    - Service
    - Instruction
    - Error_Message_BlueWay
    - Timestamp
- Call to webservice
- Send bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 999

Fields	Data
Err_Num	
Component	
Service	
Instruction	
Err_Msg_BW	
TimeStamp	

9.6.6 SCALL_SLR_SVC_Cleanup_ProcessedData

- Triggered by bus message MEM_ERROR.SCERR_ALL_MEM_ReceivedError with ID = 005
- Remove the received files and temporary data
- If the MEM_ERROR.SCERR_ALL_MEM_ReceivedError Service is FTP then
    - Call FTP delete with the user/password and MEM_ERROR.SCERR_ALL_MEM_ReceivedError.ReceivedFilePath

-

9.7 Functional Errors

- Sender unknown
- Recipient unknown
- No file found (SMTP with no attachment)

9.8 Technical Errors

- Unable to connect FTP
- Unable to connect SMTP
- Unable to connect SOAP
- SOAP error
- Disk Space

9.9 Error handling

Met opmerkingen [SB1]: Yet to be defined

Fill logs with correct error data. Handle error.

10 Management Portal

10.1 Description the management portal

The current Covast system is replaced in 2015 by a new MIS built with Blueway BPM Solutions. The Covast system has a number of monitor possibilities which should be also available in Blueway.
Together with the users of the new system a number of requirements were listed.
These will be described in this FD. For the definition of database model see the functional designs of the 'steering model' and the 'file wrapper and audit trail'.

10.2 Introduction

10.3 Functional description

The Management portal enable authorized users to:

- Manage reference table (Steering, wrapper, Sequence)
  - Configure parties
  - Configure flows
  - Configure rules
  - This could be extended to any necessary Tables
- Manage processed message

- Audit trail
- Error report

- Report platform activity
  - Qte messages /day
  - Operational Reporting
  - Invoicing reporting
  -