

Projet

---

**BDA**

---

Banque Des Apprenants

---

SWIFT Transfert

# Notre équipe

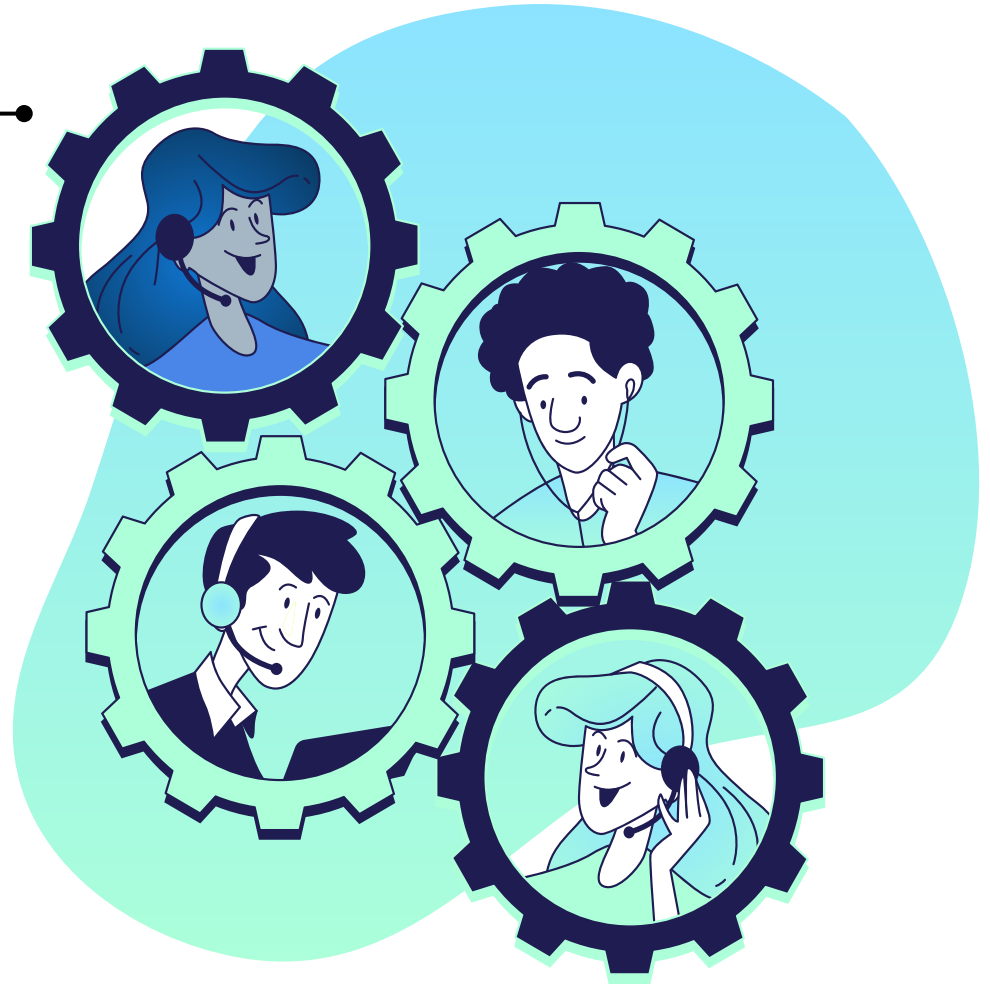
---

Anssan SILIADIN

Chenjia LI-GRENET

Bruno CECONI

Karim GHARSSA



# Spécifications des besoins

Banque Des Apprenants

Contrôle du fichier entrant

CFMVT

Création, lecture, mise à jour en BDD Client-Compte

CT_CLIENT	CT_COMPTE	CT_STATUT
CT_CPTCLI	CT_OPER	

Édition des informations

Client et Compte  
Opérations sur compte



# Spécifications des besoins

## SWIFT Transfert

### Contrôle du fichier entrant

SFMVT

### Création et lecture en BDD SWIFT

ST_BIC	ST_DEVISE
ST_MSWIFT	ST_CHANGE

### Édition des informations

Messages MT103 et MT202



# La STACK

---

## L'environnement

Z/OS release 1.10

ISPF 6.0

DB2 release 9.10



# La STACK

---

## Les langages

COBOL

JCL

SQL





# Les librairies

Source

BDA

CP1

CA1

CP2

CP3

CA2

SWIFT Transfert

SP1

SA1

SSP2

SSP3

SP2



# Les librairies

## Copie

### DCLGEN

CLIENT

COMPTE

CPTCLI

OPER

BIC

DEVISE

CHANGE

MSWIFT

### CLAUSE COPY

CPIERR

LOGERR

MSGERSW

PRJCFMVT

PRJSFMVT

ZCOMC

ZCOMS





# Les librairies

---

JCL

---

BDA

---

JCP1

JCP2

JCP3

SWIFT Transfert

---

JSP1

JSP2



# Spécifications Techniques

Banque Des Apprenants

CFMVT



CP1



CA1

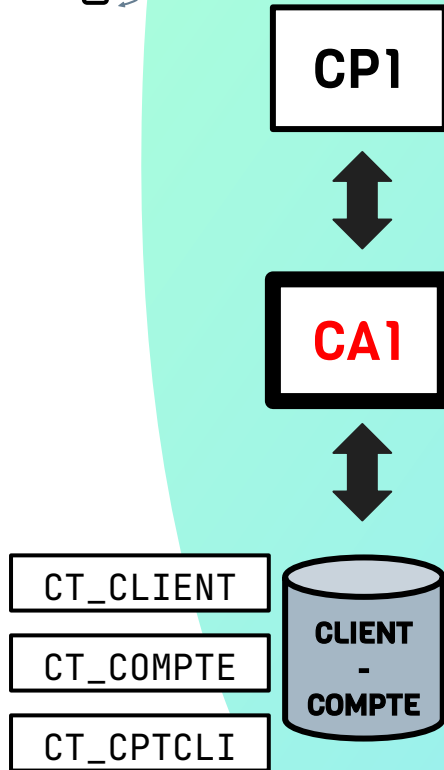
```
SELECT CFMVT ASSIGN TO INP001  
ORGANIZATION IS SEQUENTIAL  
ACCESS MODE IS SEQUENTIAL  
FILE STATUS IS FS-CFMVT.
```

```
FIELDS-CTRL.  
SET FIELD-OK TO TRUE  
PERFORM WRITE-LINE-NUMENR THRU WRITE-LINE-RECINFO  
PERFORM FLD-CODMVT-CTRL THRU FLD-SOLDE-CTRL  
.
```

```
MOVE '5' TO ZCOM-ACTION  
MOVE WS-CFMVT-NUMCLI TO ZCOM-CLIENT-NUMCLI  
CALL CA1 USING ZCOM
```

# Spécifications Techniques

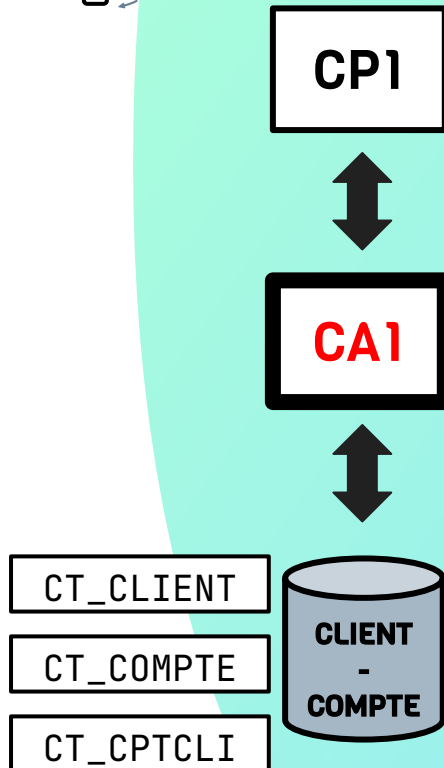
Banque Des Apprenants



```
LINK/ EVALUATE ZCALL-ACTION
      WHEN '1'
        PERFORM CRE-CLIENT
      WHEN '2'
        PERFORM CRE-COMPTE
        PERFORM RAJOUT-LIGNE-CPTCLI
      WHEN '3'
        PERFORM CRE-CLIENT
        PERFORM CRE-COMPTE
        PERFORM RAJOUT-LIGNE-CPTCLI
      WHEN '4'
        PERFORM MOD-CLIENT
      WHEN '5'
        PERFORM LEC-CLIENT
      WHEN '6'
        PERFORM LEC-COMPTE
      WHEN OTHER
        SET ZCALL-ERREUR TO TRUE
        PERFORM FIN-ERR
      END-EVALUATE
```

# Spécifications Techniques

Banque Des Apprenants



```
DATA DWTSTOM
EXEC SQL
SELECT MAX(NUMCLI)
INTO :LAST-NUMCLI
FROM ADS0103.CT_CLIENT
END-EXEC

COMPUTE WS-NUMCLI = LAST-NUMCLI + 1

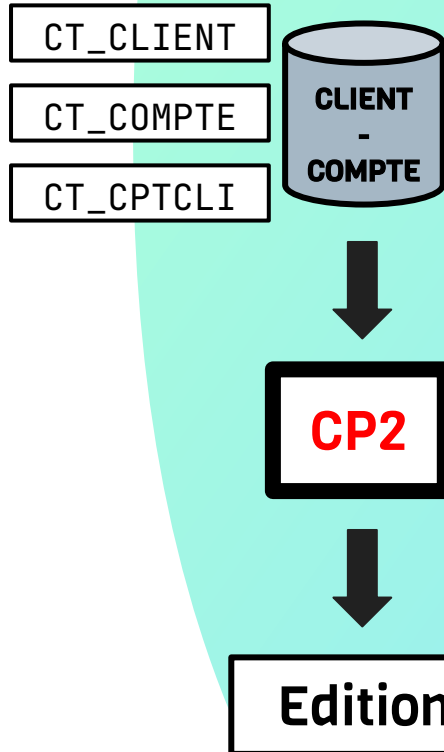
EXEC SQL
INSERT INTO ADS0103.CT_CLIENT
VALUES (:WS-NUMCLI,
        :WS-STATUT,
        :WS-NOM000,
        :WS-PRENOM,
        :WS-VILLE :IND-VILLE,
        :WS-CDPOST :IND-CDPOST,
        :WS-DATENT,
        :WS-DATMAJ)
END-EXEC
```

```
EXEC SQL
UPDATE ADS0103.CT_CLIENT
SET STATUT = :WS-STATUT,
    NOM000 = :WS-NOM000,
    PRENOM = :WS-PRENOM,
    VILLE = :WS-VILLE :IND-VILLE,
    CDPOST = :WS-CDPOST :IND-CDPOST,
    DATMAJ = :WS-DATMAJ
WHERE NUMCLI = :WS-NUMCLI
END-EXEC
```

```
LEC-COMPTÉ.
* CONTROLE COMPTE EXISTE
EXEC SQL
SELECT NUMCPT
INTO :WS-NUMCPT-CPT
FROM ADS0103.CT_COMPTE
WHERE NUMCPT = :WS-NUMCPT-CPT
END-EXEC
```

# Spécifications Techniques

## Banque Des Apprenants



```
SQL DECLARE CURSOR
FOR SELECT DISTINCT
  C.NUMCLI,
  C.STATUT,
  C.NOM000,
  C.PRENOM,
  VALUE(C.VILLE
  VALUE(C.CDPOS
  C.DATENT,
  C.DATMAJ,
  P.NUMCPT,
  P.STATUT,
  P.SOLDE,
  P.DATOUV,
  P.DATMAJ
FROM ADS0103.CT_CLIENT
ON C.NUMCLI = T.NUMCLI
ADS0103.CT_COMPTE
ON P.NUMCPT = T.NUMCPT
ORDER BY C.NUMCLI, P.N
-- EXEC

CURM FETCH-CURSEUR
SQLCODE = ZEROES
MOVE WC-NUMCLI TO EC-NUMCLI
MOVE WC-STATUT TO EC-STATUT
STRING WC-NOM000 DELIMITED BY SPACE INTO EC-NOM
STRING WC-PRENOM DELIMITED BY SPACE INTO EC-PRENOM
MOVE WC-VILLE TO EC-VILLE
MOVE WC-CDPOST TO EC-CDPOST
MOVE WC-DATENT TO EC-DATENT
MOVE WC-DATMAJ TO EC-DATMAJ
MOVE WP-NUMCPT TO EP-NUMCPT
MOVE WP-SOLDE TO EP-SOLDE
ITURE LIGNE DETAIL
IF EC-NUMCLI NOT EQUAL NUMCLITEMP
  ADD 1 TO NB-CLIENTS
  WRITE CP2-ENR FROM LIGSEP
  WRITE CP2-ENR FROM LIG3
  WRITE CP2-ENR FROM LIG3-0
  WRITE CP2-ENR FROM LIG4
  WRITE CP2-ENR FROM LIGSEP
  WRITE CP2-ENR FROM LIG4-0
  WRITE CP2-ENR FROM LIG4-1
  MOVE WC-NUMCLI TO NUMCLITEMP
END-IF
  WRITE CP2-ENR FROM LIG4-2
ELSE
  IF SQLCODE NOT = +100
    DISPLAY 'ERREUR FETCH CURSEUR ' SQLCODE
    PERFORM FIN
  END-IF
  IF
    CURCLIENT END-EXEC.
    ZEROES
    OUVERTURE CURSEUR ' SQLCODE

  .
  FETCH CURCLIENT
  -NUMCLI,
  -STATUT,
  -NOM000,
  -PRENOM,
  -VILLE,
  -CDPOST,
  -DATENT,
  -DATMAJ,
  -NUMCPT,
  -STATUT,
  -SOLDE,
  -DATOUV,
  -DATMAJ
```



# Spécifications Techniques

Banque Des Apprenants

CT\_OPER

CLIENT  
-  
COMPTE

CP3

Edition

```
EXEC SQL INCLUDE SQLCA      END-EXEC.  
EXEC SQL INCLUDE OPER        END-EXEC.
```

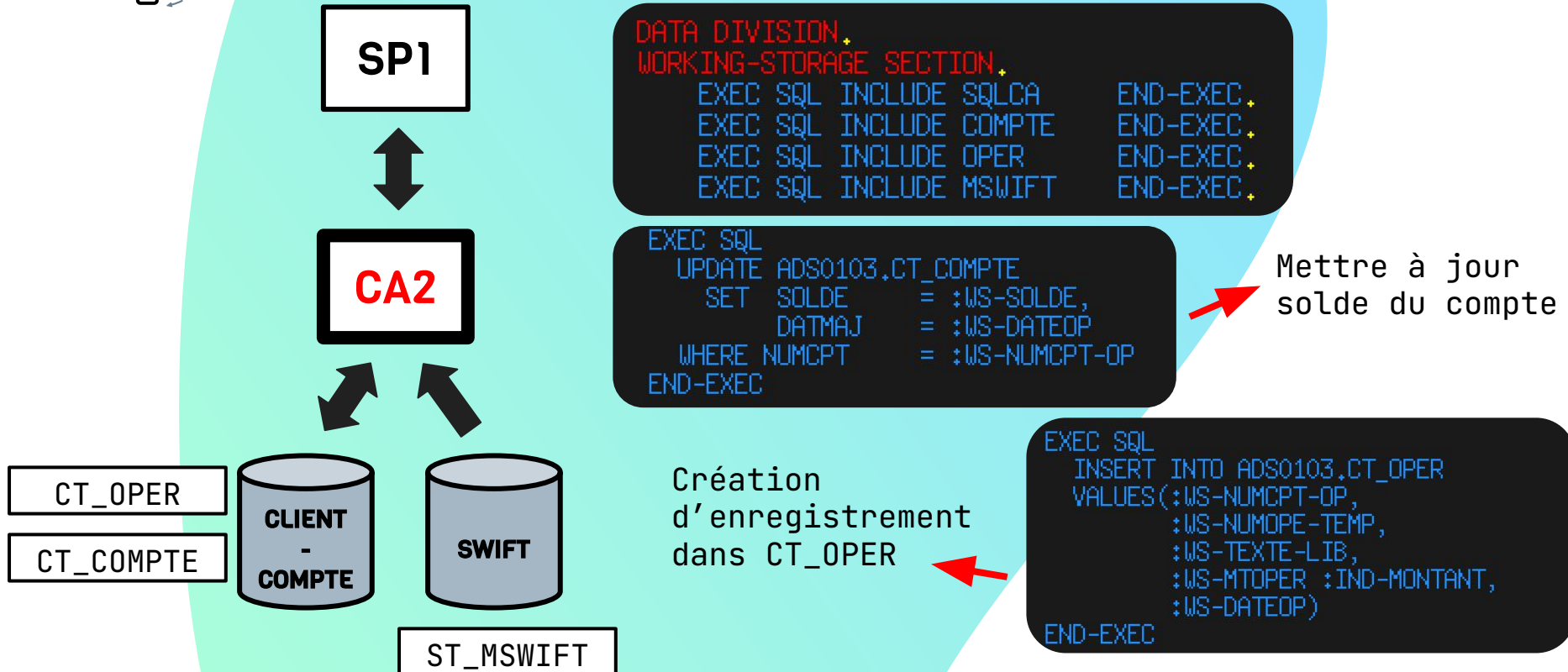
```
EXEC SQL DECLARE CURSOPER CURSOR  
FOR SELECT  
        NUMCPT,  
        NUMOPE,  
        TEXTE0,  
        VALUE(MTOPER, 0),  
        DATEOP  
FROM ADS0103.CT_OPER  
ORDER BY NUMCPT, NUMOPE  
END-EXEC
```

```
FETCH-CURSEUR.  
EXEC SQL FETCH CURSOPER  
INTO   :WR-NUMCPT,  
        :WR-NUMOPE,  
        :WR-TEXTE0,  
        :WR-MTOPER,  
        :WR-DATOPE  
END-EXEC  
.
```

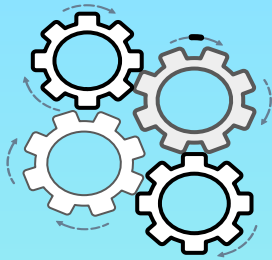


# Spécifications Techniques

## Banque Des Apprenants



# Spécifications Techniques



## SWIFT Transfert

```

*SSP2.
CREATION DU MESSAGE T103
MOVE '103' TO ZCOM-SWIFT-TYPMSG
MOVE DATVAL-FORMATE TO ZCOM-SWIFT-DATVAL
MOVE ENRW-CODDEV TO ZCOM-SWIFT-CODDEV

VERIFICATION
AC-SA1CLI
* TRAITEMENT DU CHANGE DU MONTANT
MOVE IF ENRW-CODDEV = 'EUR'
MOVE ENRW-MTTRAN TO ZCOM-SWIFT-MTTRAN
ELSE
PERFORM TRT-MTH
MOVE MTTRAN-CHANGE TO ZCOM-SWIFT-MTTRAN
END-IF

MOVE 'SSP2CLI' TO SOUS-PROG
MOVE WS-SWIFT-NUMCLID TO ZCOM-SWIFT-NUMCLID
MOVE WS-SWIFT-NOMCLID TO ZCOM-SWIFT-NOMCLID
MOVE ENRW-NUMCPTD TO ZCOM-SWIFT-NUMCPTD
MOVE 'BAPFFRPPXXX' TO ZCOM-SWIFT-BICDEB
MOVE WS-SWIFT-BICCOR TO ZCOM-SWIFT-BICCOR
MOVE ENRW-NOMCLIC TO ZCOM-SWIFT-NOMCLIC
MOVE ENRW-NUMCPTC TO ZCOM-SWIFT-NUMCPTC
MOVE ENRW-BICCRD TO ZCOM-SWIFT-BICCRD
MOVE WS-DATE-FORMATE TO ZCOM-SWIFT-DATCRE

CALL SOUS-PROG USING ZCOM
EVALUATE ZCOM-CODE-RETOUR
WHEN 0
DISPLAY 'MT103 ENVOYE'
WHEN 2
SET MT103-ERROR TO TRUE
DISPLAY MSG-ERROR
ADD 1 TO IND-VERIF-SSP
WHEN 9
DISPLAY 'ERREUR SQL MT103 : ' ZCOM-SQLCODE
ADD 1 TO IND-VERIF-SSP
END-EVALUATE

```

TABLE

CFMVT



SP1

CA2



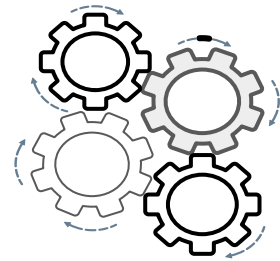
SA1

SSP2

SSP3



# Spécifications Techniques

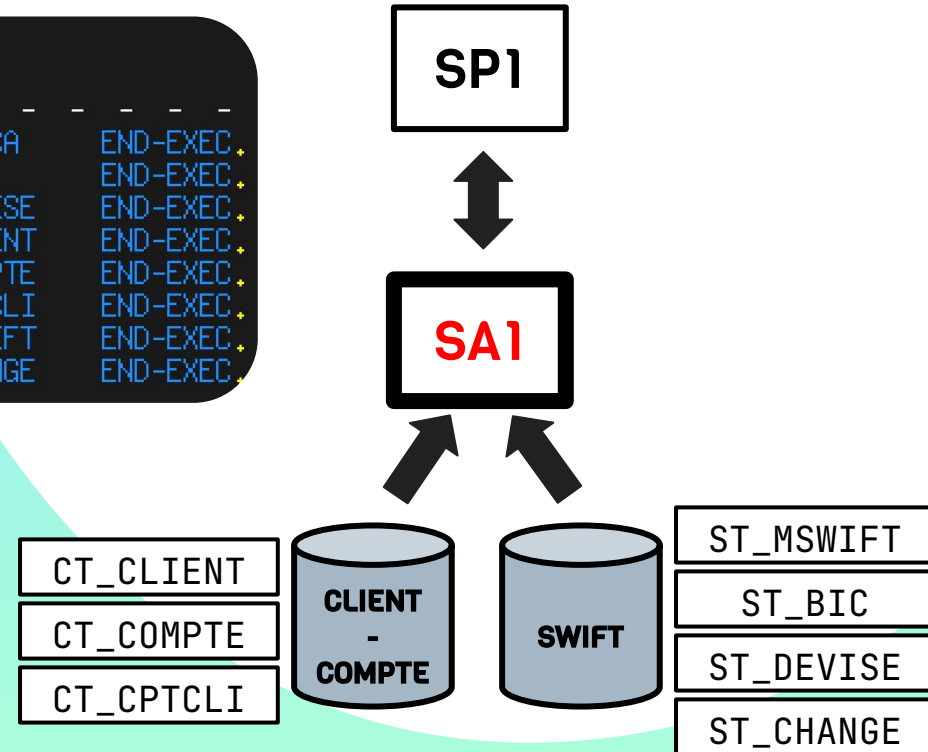


## SWIFT Transfert

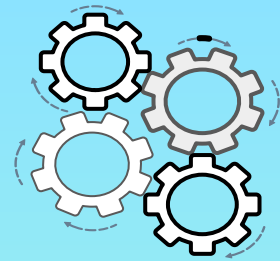
Contrôler:

1. banque
2. devise
3. compte
4. statut d'un compte
5. banque correspondante
6. informations client
7. taux de change

```
DATA DIVISION+  
WORKING-STORAGE SECTION+  
-----  
EXEC SQL INCLUDE SQLCA      END-EXEC+  
EXEC SQL INCLUDE BIC        END-EXEC+  
EXEC SQL INCLUDE DEVISE     END-EXEC+  
EXEC SQL INCLUDE CLIENT     END-EXEC+  
EXEC SQL INCLUDE COMPTE     END-EXEC+  
EXEC SQL INCLUDE CPTCLI     END-EXEC+  
EXEC SQL INCLUDE MSWIFT     END-EXEC+  
EXEC SQL INCLUDE CHANGE     END-EXEC+
```



# Spécifications Techniques



## SWIFT Transfert

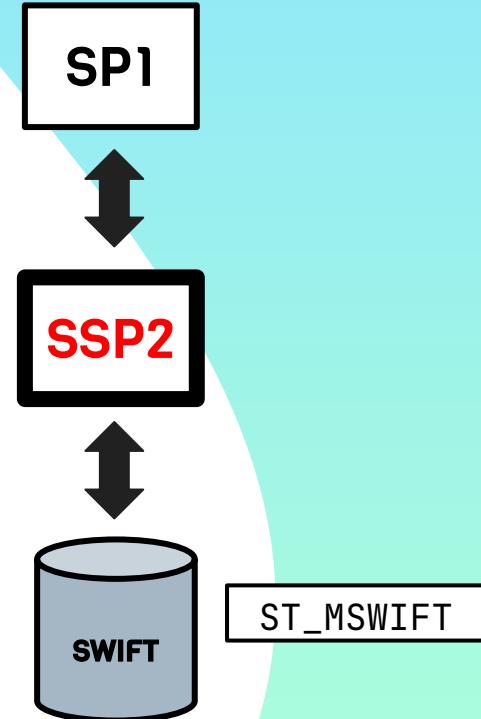
DATA DIVISION.

WORKING-STORAGE SECTION.

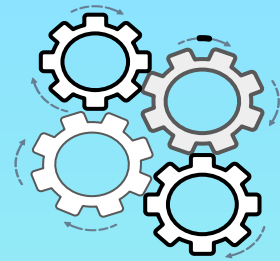
```
EXEC SQL INCLUDE SQLCA      END-EXEC.  
EXEC SQL INCLUDE MSWIFT     END-EXEC.
```

```
EXEC SQL  
  INSERT INTO ADS0103.ST_MSWIFT  
  VALUES(:WS-REFMSG,  
         :WS-TYPMSG,  
         :WS-DATVAL,  
         :WS-CODDEV,  
         :WS-MTTRAN,  
         :WS-NUMCLID,  
         :WS-NOMCLID,  
         :WS-NUMCPTD,  
         :WS-BICDEB,  
         :WS-BICCOR,  
         :WS-NUMCLIC :IND-NUMCLIC,  
         :WS-NOMCLIC,  
         :WS-NUMCPTC,  
         :WS-BICCRD,  
         :WS-DATCRE)  
END-EXEC
```

Création d'opération  
MT103 dans ST\_MSWIFT



# Spécifications Techniques

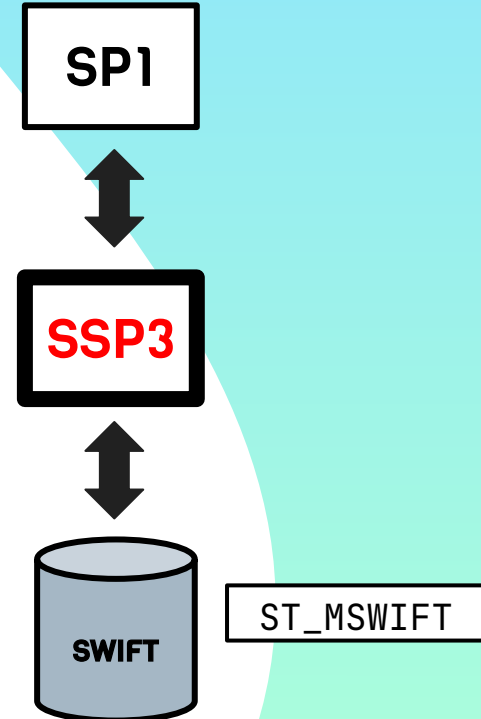


## SWIFT Transfert

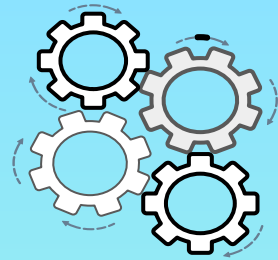
```
DATA DIVISION.  
WORKING-STORAGE SECTION.  
    EXEC SQL INCLUDE SQLCA      END-EXEC.  
    EXEC SQL INCLUDE MSWIFT     END-EXEC.
```

Création d'opération  
MT202 dans ST\_MSWIFT

```
EXEC SQL  
    INSERT INTO ADS0103,ST_MSWIFT  
    VALUES (:WS-REFMSG,  
            :WS-TYPMSG,  
            :WS-DATVAL,  
            :WS-CODDEV,  
            :WS-MTTRAN,  
            NULL,  
            NULL,  
            NULL,  
            :WS-BICDEB,  
            :WS-BICCOR,  
            NULL,  
            NULL,  
            NULL,  
            :WS-BICCRD,  
            :WS-DATCRE)  
END-EXEC
```



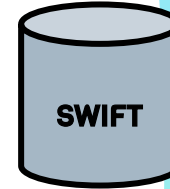
# Spécifications Techniques



## SWIFT Transfert

```
EXEC SQL DECLARE CURSMMSG CURSOR  
FOR SELECT  
REFMSG,  
TYPMSG,  
DATVAL,  
CODDEV,  
MTTRAN,  
VALUE(NUMCLID, 0),  
VALUE(NOMCLID, ' '),  
VALUE(NUMCPTD, 0),  
BICDEB,  
BICCOR,  
VALUE(NUMCLIC, 0),  
VALUE(NOMCLIC, ' '),  
VALUE(NUMCPTC, 0),  
BICCRD,  
DATCRE  
FROM ADS0103.ST_MSWIFT  
ORDER BY REFMSG  
END-EXEC
```

```
EXEC SQL FETCH CURSMMSG  
INTO :WM-REFMSG,  
:WM-TYPMSG,  
:WM-DATVAL,  
:WM-CODDEV,  
:WM-MTTRAN,  
:WM-NUMCLID,  
:WM-NOMCLID,  
:WM-NUMCPTD,  
:WM-BICDEB,  
:WM-BICCOR,  
:WM-NUMCLIC,  
:WM-NOMCLIC,  
:WM-NUMCPTC,  
:WM-BICCRD,  
:WM-DATCRE  
END-EXEC
```



ST\_MSWIFT



**SP2**



**Edition**

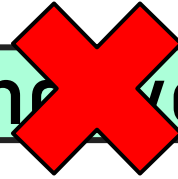
# Difficultés et Solutions

## Versioning

Extraction

Encoder

Copy/Paste



## Linkage Section

Nommage

Taille

## Variable

Date

OC7

COMP-3



# Difficultés et Solutions

## Teamwork

CODE



DOC

Individuel



Collectif

Remote



Présentiel

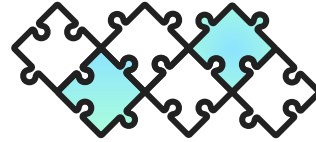


# Difficultés et Solutions

---

## Teamwork

Task Management



Time Management



# Axes d'amélioration

Refactoring

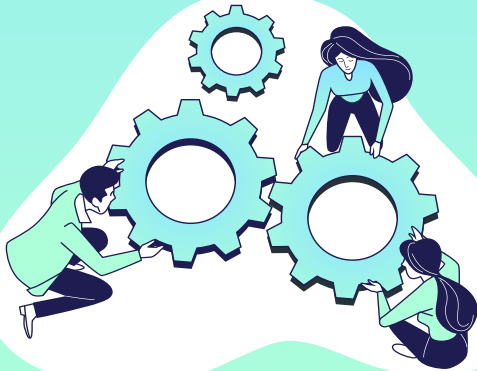
Data Set Edition

Optimisation





# Points Forts



---

# Merci

---

Livrables et documentations

---

<https://github.com//cligrenetprojet-bda-swift>