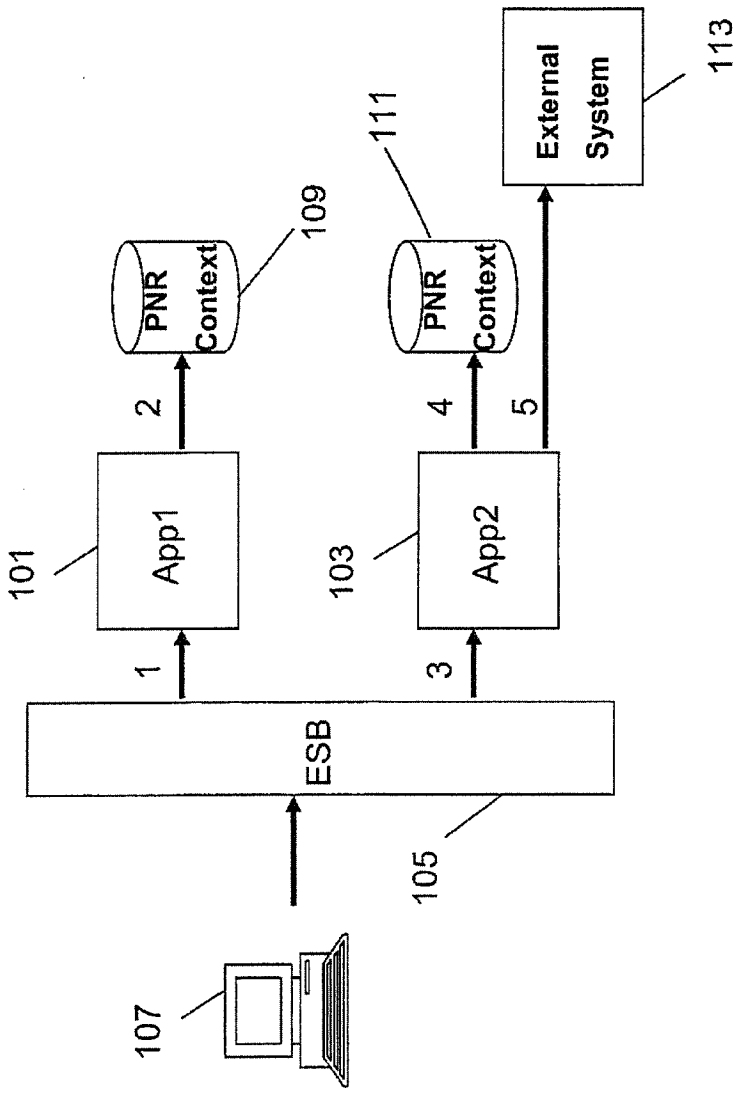
107
105
ESB
1
3
101
App1
103
App2
2
4
5
PNR
Context
109
111
PNR
Context
External
System
113

Fig. 1

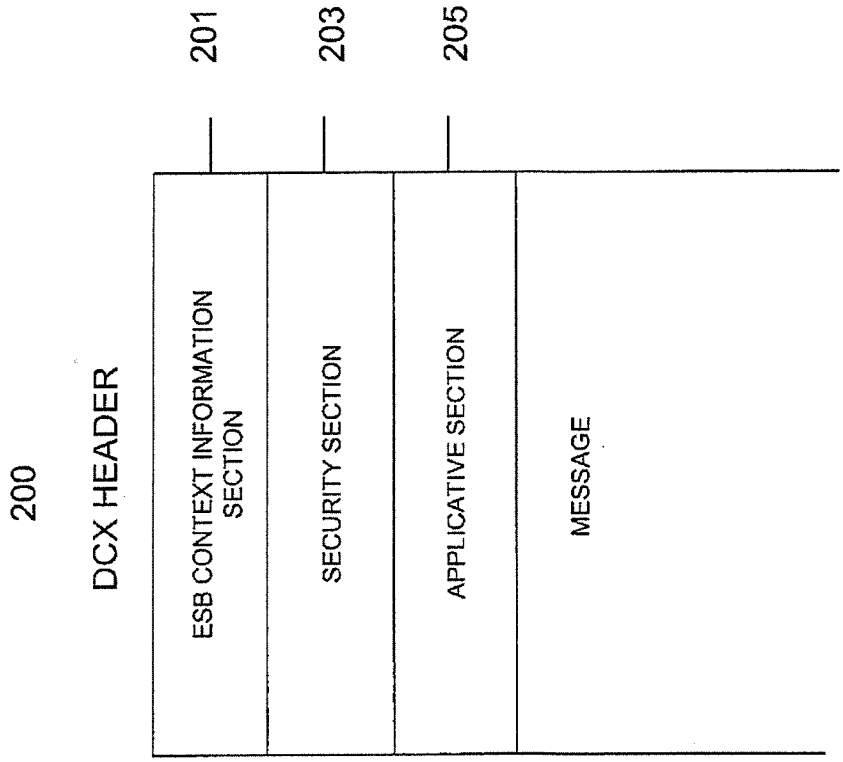
200
DCX HEADER
201
203
205
ESB CONTEXT INFORMATION SECTION
SECURITY SECTION
APPLICATIVE SECTION
MESSAGE

Fig. 2

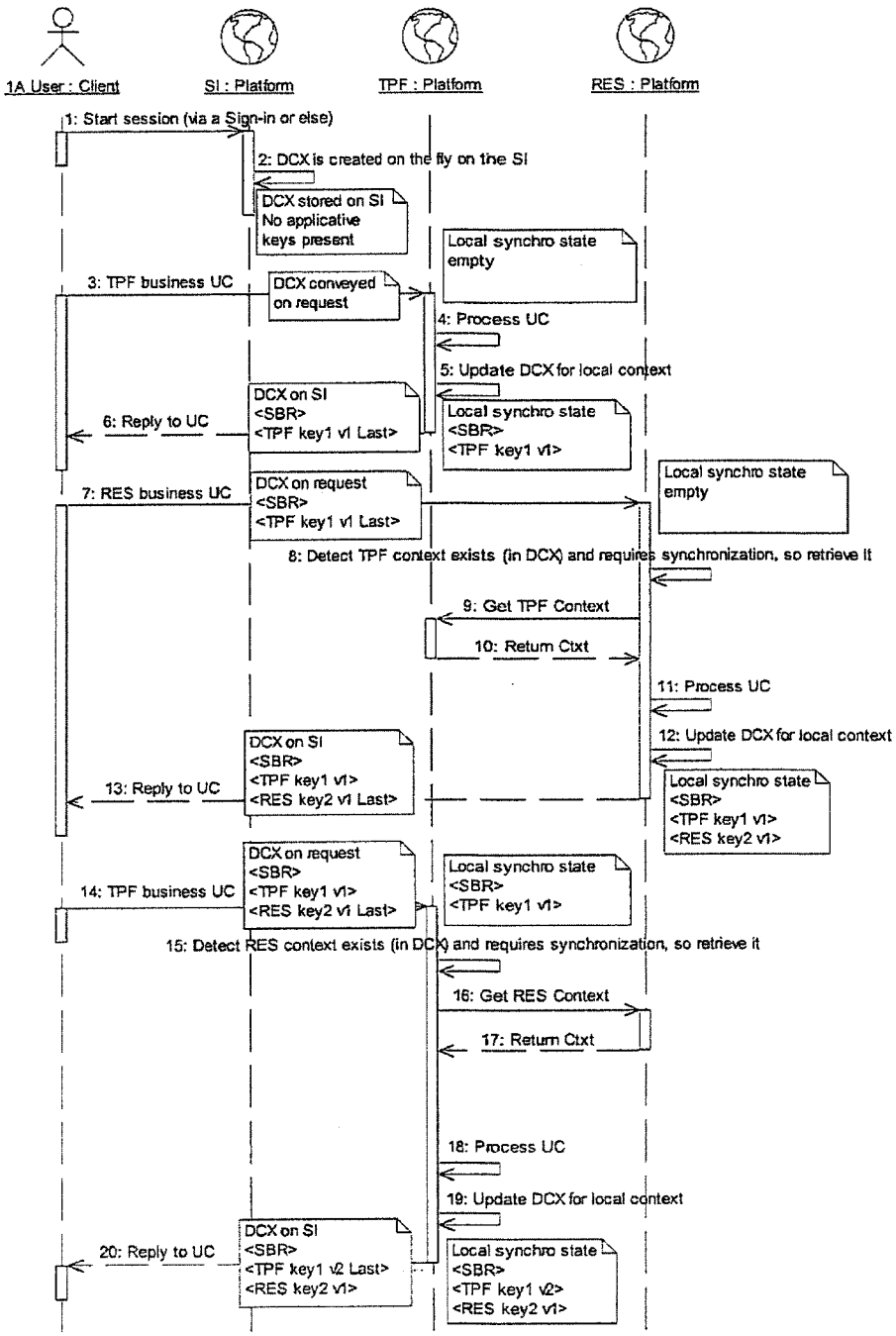
1A User : Client
SI : Platform
TPF : Platform
RES : Platform
1: Start session (via a Sign-in or else)
2: DCX is created on the fly on the SI
DCX stored on SI
No applicative
keys present
Local synchro state
empty
3: TPF business UC
DCX conveyed
on request
4: Process UC
5: Update DCX for local context
DCX on SI
<SBR>
<TPF key1 v1 Last>
6: Reply to UC
Local synchro state
<SBR>
<TPF key1 v1>
Local synchro state
empty
7: RES business UC
DCX on request
<SBR>
<TPF key1 v1 Last>
8: Detect TPF context exists (in DCX) and requires synchronization, so retrieve it
9: Get TPF Context
10: Return Ctxt
11: Process UC
12: Update DCX for local context
DCX on SI
<SBR>
<TPF key1 v1>
<RES key2 v1 Last>
13: Reply to UC
Local synchro state
<SBR>
<TPF key1 v1>
<RES key2 v1>
DCX on request
<SBR>
<TPF key1 v1>
<RES key2 v1 Last>
14: TPF business UC
Local synchro state
<SBR>
<TPF key1 v1>
15: Detect RES context exists (in DCX) and requires synchronization, so retrieve it
16: Get RES Context
17: Return Ctxt
18: Process UC
19: Update DCX for local context
DCX on SI
<SBR>
<TPF key1 v2 Last>
<RES key2 v1>
20: Reply to UC
Local synchro state
<SBR>
<TPF key1 v2>
<RES key2 v1>

Fig. 3

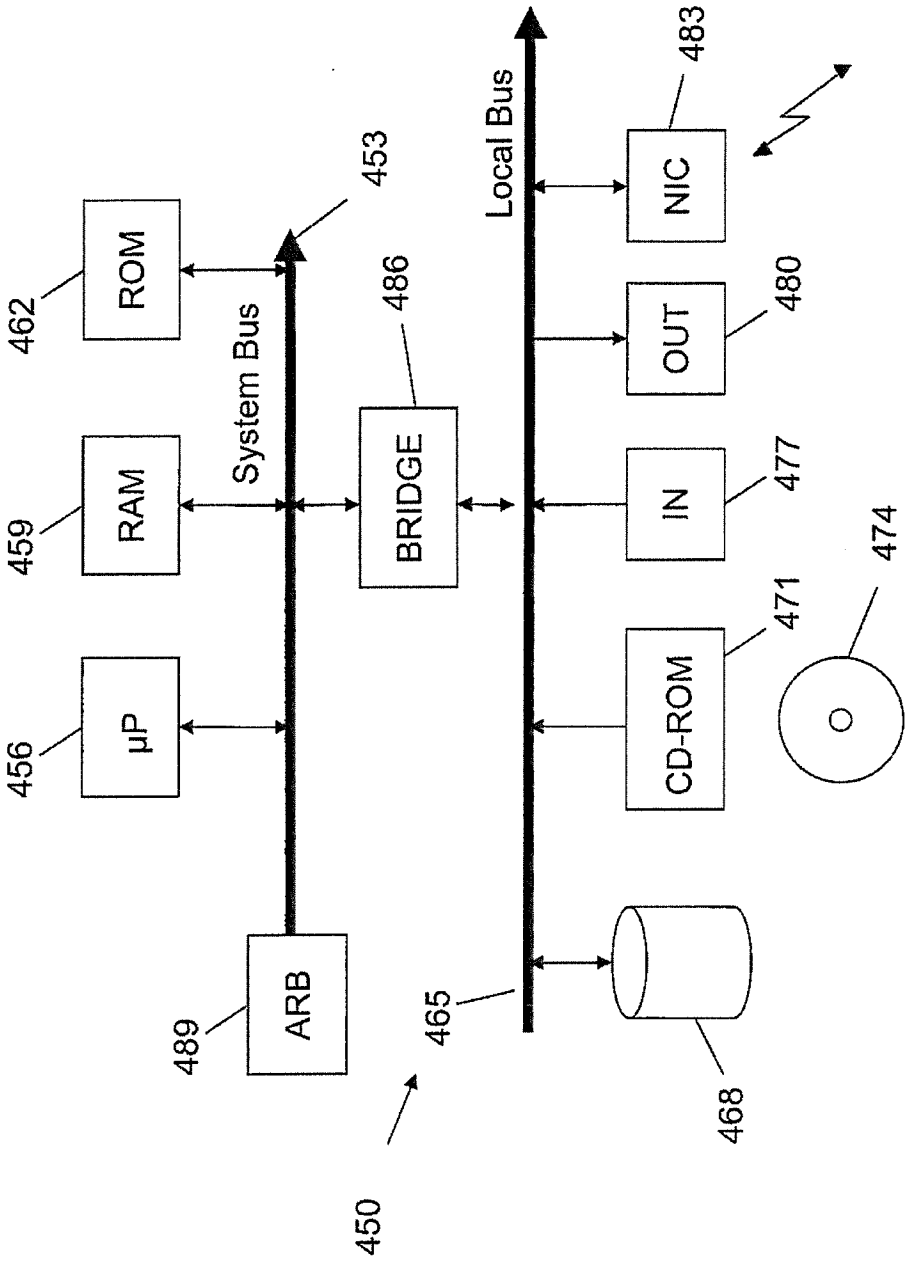
450
462
ROM
459
RAM
456
µP
System Bus
453
489
ARB
BRIDGE
486
Local Bus
465
468
NIC
483
OUT
480
IN
477
CD-ROM
471
474

Fig. 4

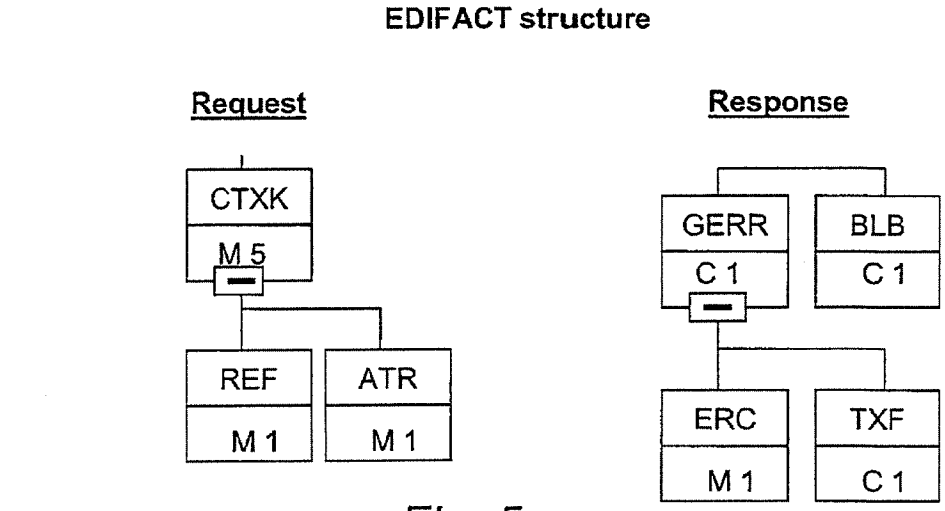
EDIFACT structure
Request
CTXK
M 5
REF
M 1
ATR
M 1
Response
GERR
C 1
BLB
C 1
ERC
M 1
TXF
C 1

Fig. 5a

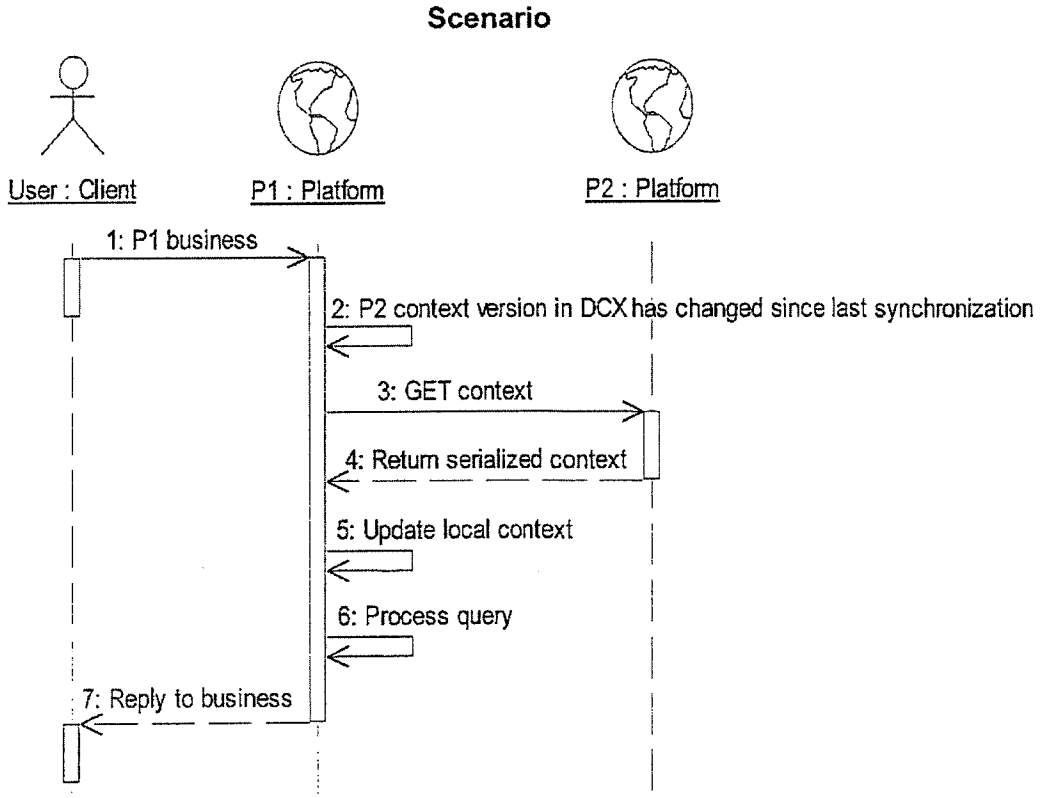
Scenario
User : Client
P1 : Platform
P2 : Platform
1: P1 business
2: P2 context version in DCX has changed since last synchronization
3: GET context
4: Return serialized context
5: Update local context
6: Process query
7: Reply to business

Fig. 5b

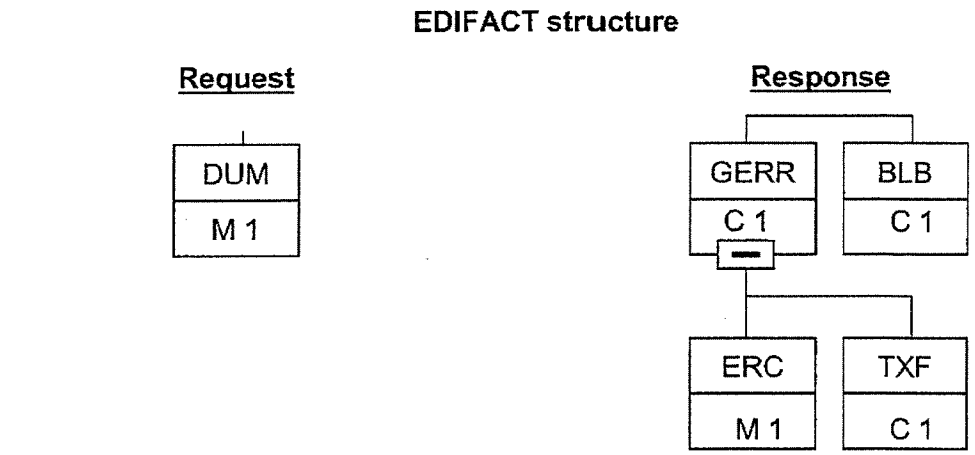
EDIFACT structure
Request
DUM
M 1
Response
GERR
C 1
BLB
C 1
ERC
M 1
TXF
C 1

Fig. 6a

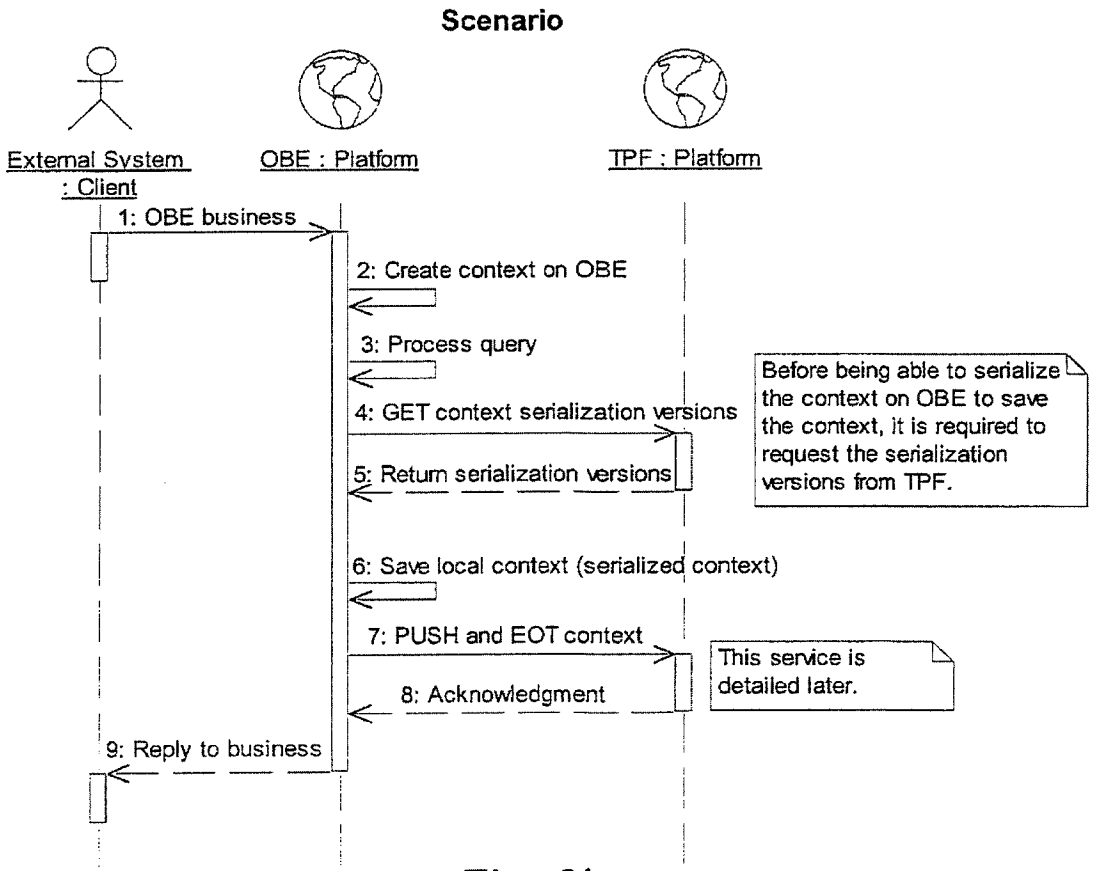
Scenario
External System : Client
OBE : Platform
TPF : Platform
1: OBE business
2: Create context on OBE
3: Process query
4: GET context serialization versions
5: Return serialization versions
Before being able to serialize the context on OBE to save the context, it is required to request the serialization versions from TPF.
6: Save local context (serialized context)
7: PUSH and EOT context
8: Acknowledgment
This service is detailed later.
9: Reply to business

Fig. 6b

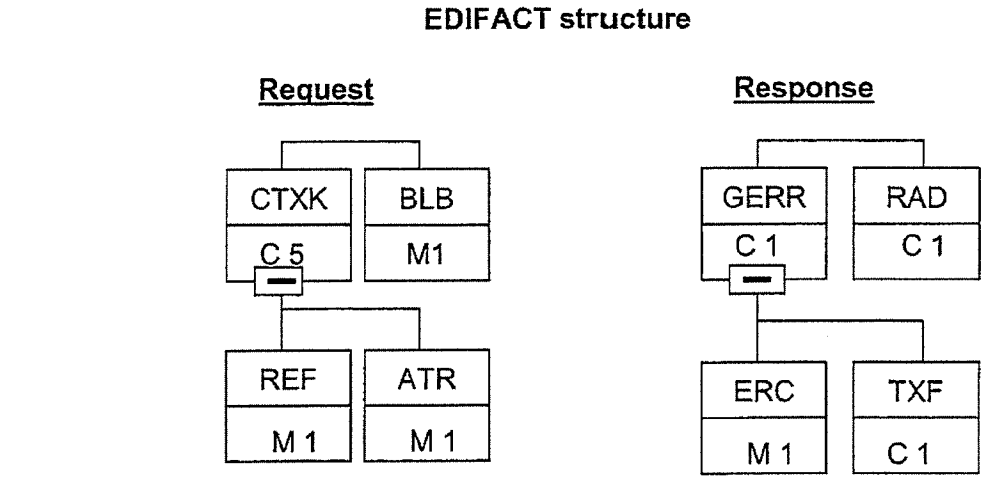
EDIFACT structure
Request
CTXK
C 5
BLB
M1
REF
M 1
ATR
M 1
Response
GERR
C 1
RAD
C 1
ERC
M 1
TXF
C 1

Fig. 7a

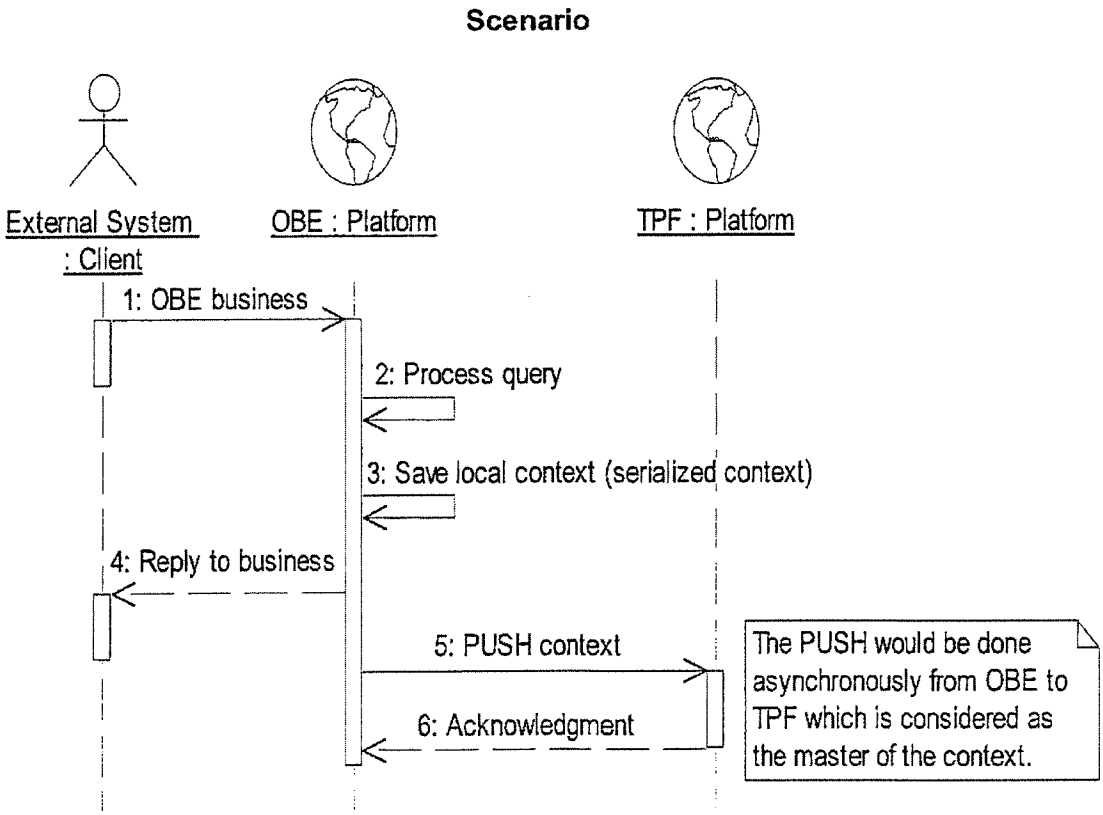
Scenario
External System
: Client
OBE : Platform
TPF : Platform
1: OBE business
2: Process query
3: Save local context (serialized context)
4: Reply to business
5: PUSH context
6: Acknowledgment
The PUSH would be done asynchronously from OBE to TPF which is considered as the master of the context.

Fig. 7b

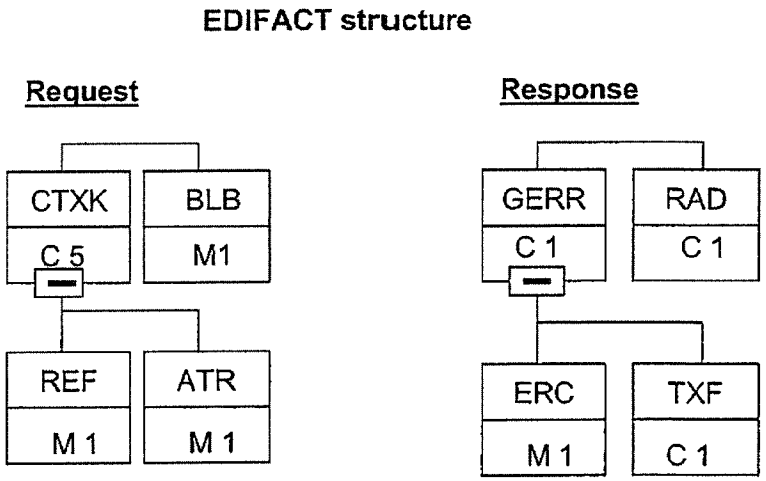
EDIFACT structure
Request
CTXK
C 5
BLB
M1
REF
M 1
ATR
M 1
Response
GERR
C 1
RAD
C 1
ERC
M 1
TXF
C 1

Fig. 8a

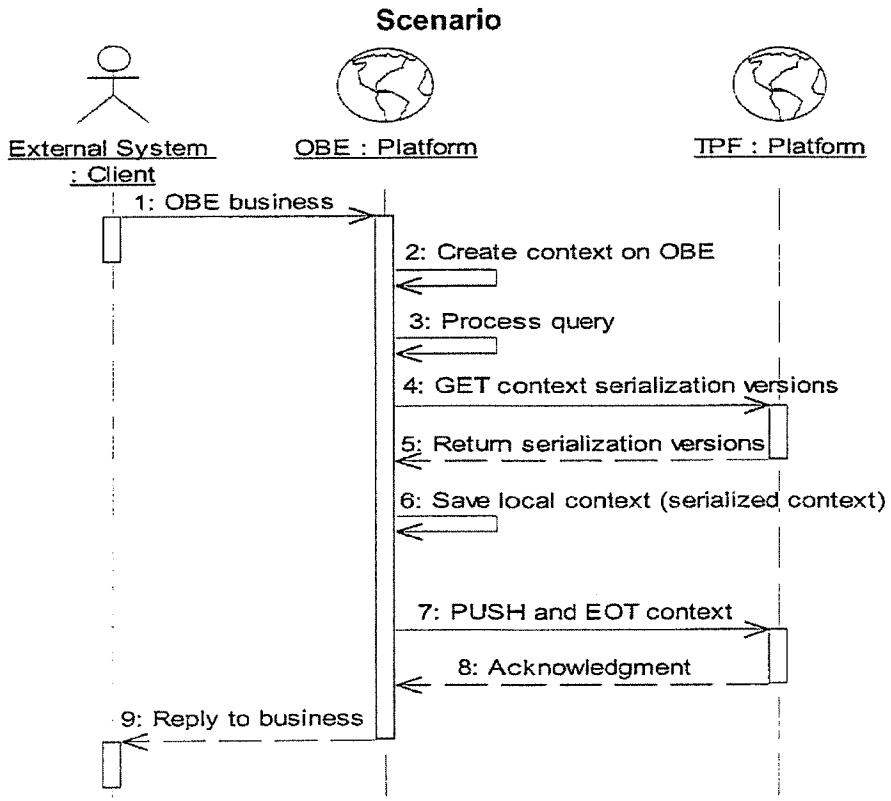
Scenario
External System
: Client
OBE : Platform
TPF : Platform
1: OBE business
2: Create context on OBE
3: Process query
4: GET context serialization versions
5: Return serialization versions
6: Save local context (serialized context)
7: PUSH and EOT context
8: Acknowledgment
9: Reply to business

Fig. 8b

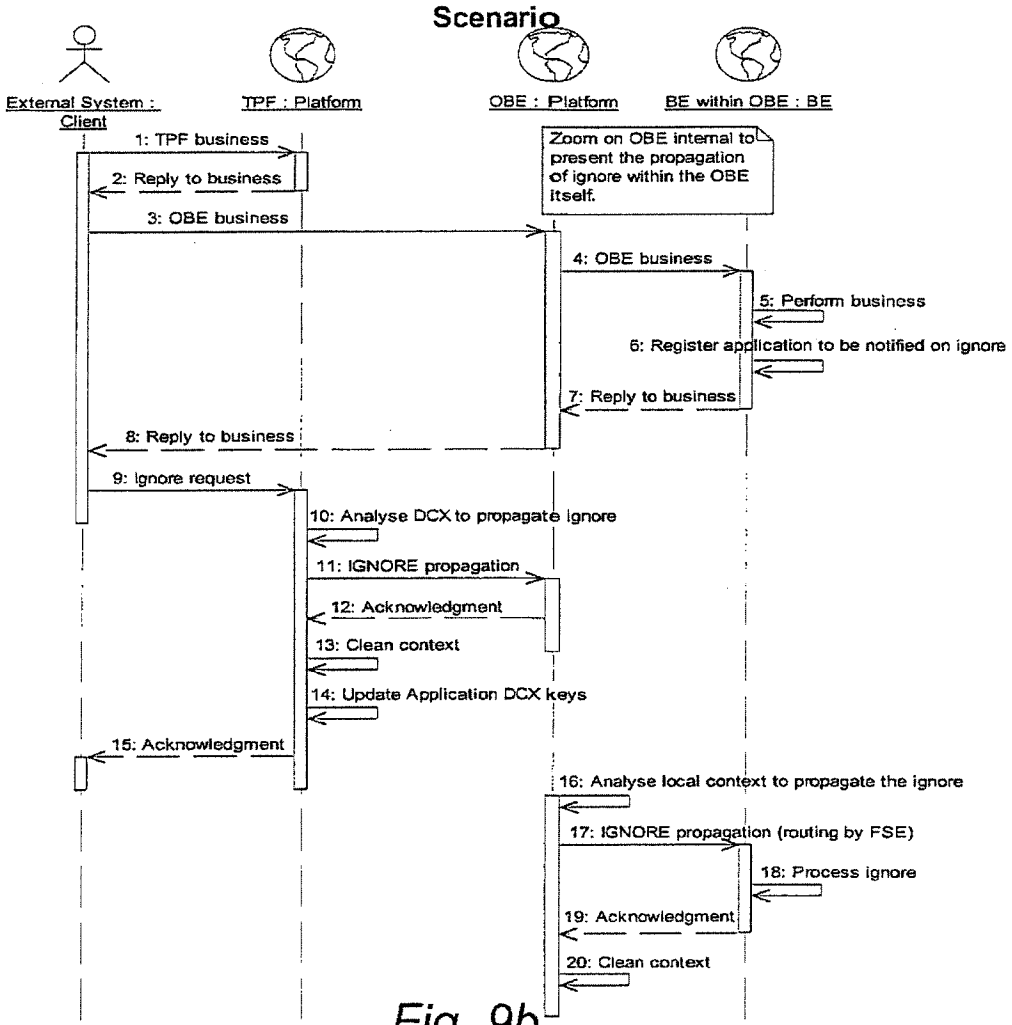

Fig. 9b

EDIFACT structure

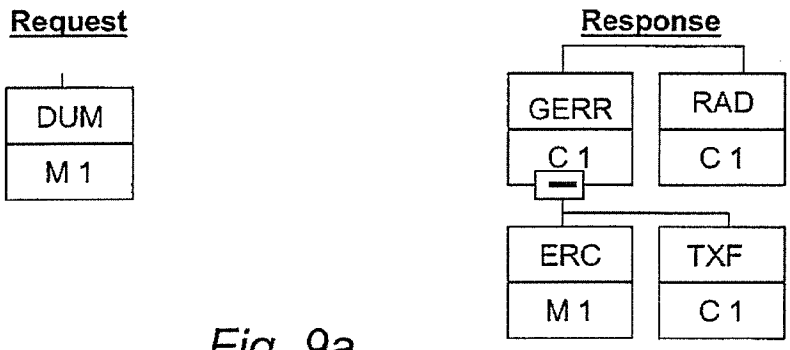

Fig. 9a

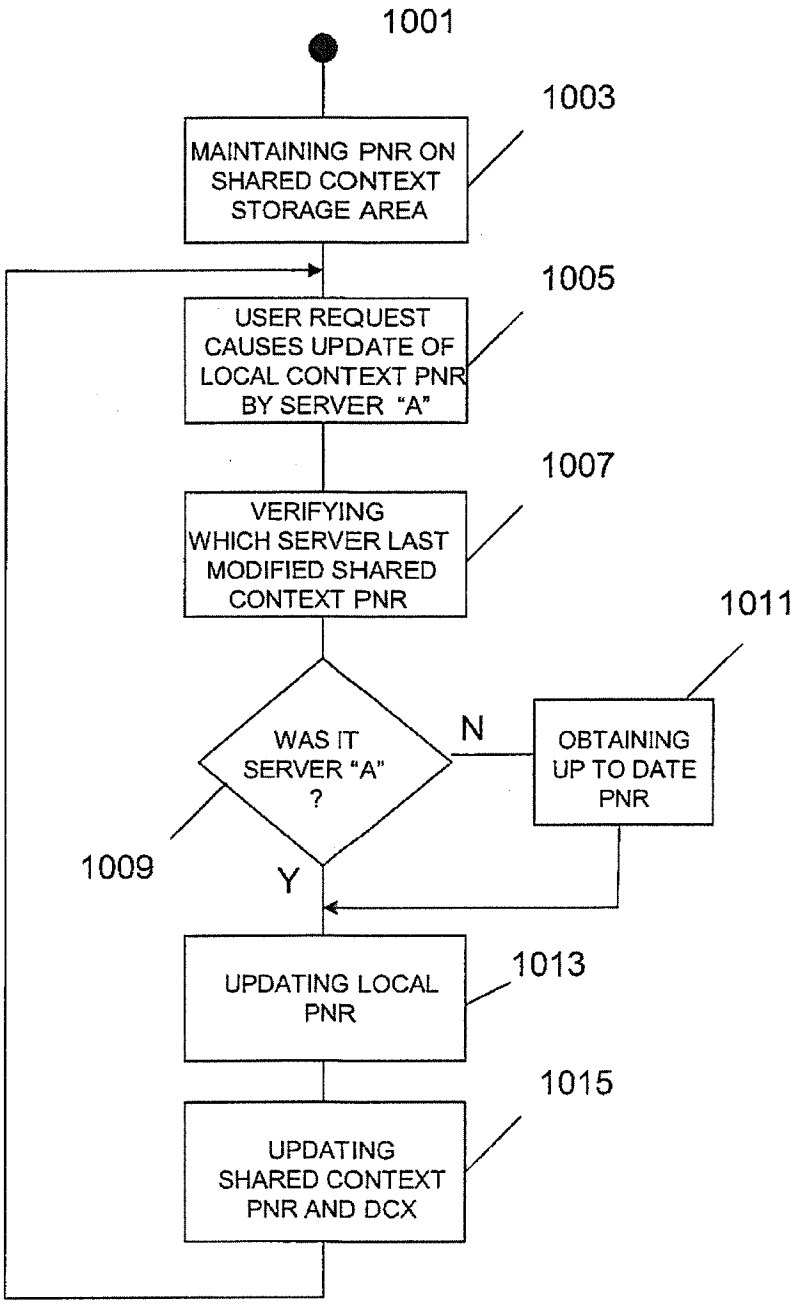
1001
1003
MAINTAINING PNR ON SHARED CONTEXT STORAGE AREA
1005
USER REQUEST CAUSES UPDATE OF LOCAL CONTEXT PNR BY SERVER "A"
1007
VERIFYING WHICH SERVER LAST MODIFIED SHARED CONTEXT PNR
1011
WAS IT SERVER "A" ?
N
OBTAINING UP TO DATE PNR
1009
Y
1013
UPDATING LOCAL PNR
1015
UPDATING SHARED CONTEXT PNR AND DCX

Fig. 10