SAP Cloud Platform Integration

Technical Specification

Technical Specifications Document

Document Release Note

Document Name:	WA Conflict Resolution - Analyze Logic
Version:	1.0.0
Description:	CORE CONFLICT DETECTION ENGINE: Analyzes time overlaps between work assignments (scheduled work times) and timesheet events (actual check-in/check-out times). Implements business rules to resolve conflicts by determining which records to delete or trim. Uses Asia/Riyadh timezone for all time calculations.
Release Date:	2025-11-06

Revision History

Revision	Date	Description	Page	Rationale	Type
1.0	2025-11-06	Initial Draft	All	Initial Version	Add

Document Contact Information

Name:	Abdelrahman Hussein
Role:	Technical Consultant

Table of Contents

1. BUSINESS CONTEXT....4
1.1 Overview....4
1.2 Development Unit Information....4
2. DETAILED DESIGN....5
2.1 Configuration Details....5
2.2 SAP CPI iFlow Design....5
Detailed Requirements:....5
Groovy Scripts....5
2.3 Adapter Configuration (Sender & Receiver)....6
2.4 Error Handling....6
3. TESTING....7
3.1 Test Conditions and Expected Results....7
3.2 Test Data Considerations....7
3.3 Performance Considerations....7

1. BUSINESS CONTEXT

1.1 Overview

CORE CONFLICT DETECTION ENGINE: Analyzes time overlaps between work assignments (scheduled work times) and timesheet events (actual check-in/check-out times). Implements business rules to resolve conflicts by determining which records to delete or trim. Uses Asia/Riyadh timezone for all time calculations.

1.2 Development Unit Information

Module	SAP Cloud Platform
Sub Module	Hana Cloud Integration
iFlow Title	WA Conflict Resolution - Analyze Logic
Processing Type	Background Online
Execution Frequency	On-Demand (Called by orchestrator)

2. DETAILED DESIGN

2.1 Configuration Details

Package Name: SF-Nadec-WorkAssignment

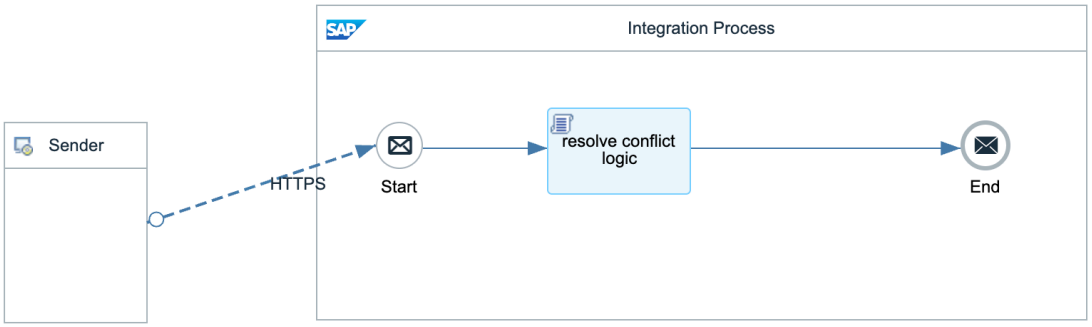
iFlow Name: WA Conflict Resolution - Analyze Logic

Technical Name: WA_TS_Conflict_Analyze

Endpoint: /resovleWAConfilict

2.2 SAP CPI iFlow Design

This is a custom-designed SAP CPI Integration flow for work assignment and timesheet conflict resolution.



Detailed Requirements:

1. Receive JSON array with paired work assignments and timesheets
2. Detect time overlaps between WA (startTime-endTime) and TS (checkIn-checkOut)
3. Apply resolution rules based on overlap type
4. Generate delete lists for work assignments and timesheets
5. Create new TimeEvent records for trimmed times (±1 minute adjustment)
6. Return structured JSON with resolved items and actions

Groovy Scripts

Script Name	Description
Parse Input Data	Extracts work assignment and timesheet pairs from input JSON
CORE ALGORITHM: Detect Overlaps	Compares time ranges: !(tsOut < waStart || tsIn > waEnd). Rules: 1. WA fully inside TS → Delete WA 2. TS fully inside WA → Delete TS (C10 and C20) 3. Partial overlap (TS starts before) → Trim TS checkout to WA start - 1 min 4. Partial overlap (TS ends after) → Trim TS checkin to WA end + 1 min 5. Other overlaps → Delete WA (fallback)

Script Name	Description
Generate Delete Lists	Creates timesheetDelete and workAssignmentDelete arrays with IDs
Create Trim Events	Generates new C10/C20 TimeEvent records with adjusted times
Format Output	Structures JSON: {resolvedItems, timesheetDelete, workAssignmentDelete, timeEventInsert}

2.3 Adapter Configuration (Sender & Receiver)

Internal processing only (no external adapter) Called via HTTP by orchestrator flows

2.4 Error Handling

Standard CPI error handling applies. Errors are logged to message processing log. Failed messages are stored in error queue for manual intervention. Retry mechanism is configured for transient failures (3 attempts with 5-second delay).

3. TESTING

3.1 Test Conditions and Expected Results

Test Condition	Expected Result
WA: 08:00-17:00, TS: 07:30-18:00 (WA inside TS)	Delete WA, keep TS
WA: 08:00-17:00, TS: 09:00-16:00 (TS inside WA)	Delete TS C10+C20, keep WA
WA: 08:00-17:00, TS: 07:00-09:00 (TS starts before)	Trim TS C20 to 07:59, keep WA
WA: 08:00-17:00, TS: 16:00-19:00 (TS ends after)	Trim TS C10 to 17:01, keep WA
No overlap between WA and TS	No action taken, both records preserved

3.2 Test Data Considerations

Test data should include: (1) Typical scenarios with standard work assignments and timesheets, (2) Edge cases with time zone boundaries, (3) Error scenarios with malformed data, (4) Load testing with bulk data volumes.

3.3 Performance Considerations

Expected processing time: <5 seconds for single record, <60 seconds for batch of 100 records. SF API rate limits: 5000 calls/hour. Memory usage: <500MB for typical batch operations.