SpeedProcessing.RDDSAPService

High-Level Design and Architecture Document

1. Executive Summary

SpeedProcessing.RDDSAPService is a critical backend integration service designed to synchronize Bill of Materials (BOM) data across multiple enterprise systems at Toyota Material Handling. This service specifically manages Toyota Special Design Request (TSDR) packages and ensures consistent data synchronization between SQL Server databases, AS400 legacy systems, and SAP ERP, facilitating smooth manufacturing operations for customized products.

2. Business Context

2.1. Business Problem

Toyota Material Handling needs to maintain consistent BOM data across multiple enterprise systems when processing special design requests. Any inconsistency can lead to manufacturing errors, inventory discrepancies, and production delays.

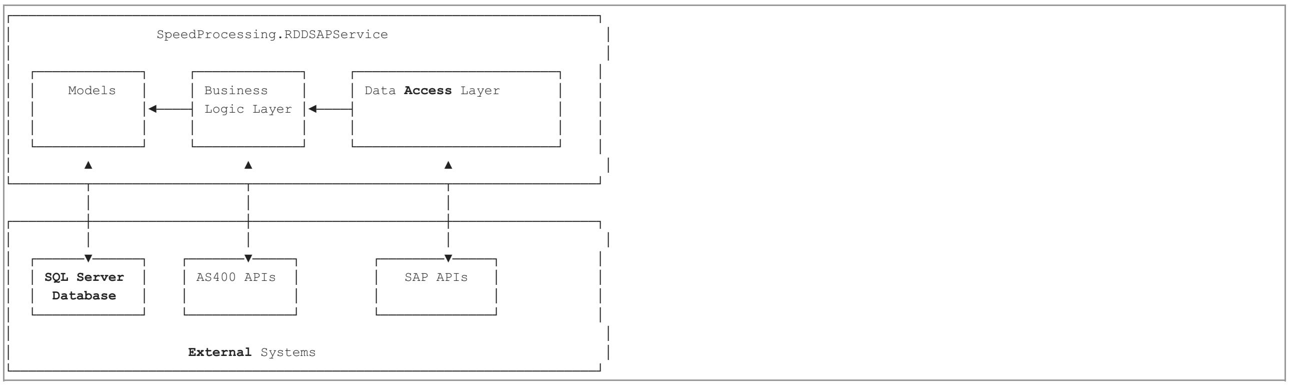
2.2. Solution Overview

SpeedProcessing.RDDSAPService addresses this challenge by providing an automated middleware solution that:

- Retrieves and processes unprocessed parts data from SQL Server
- Synchronizes this data with AS400 systems
- Creates properly formatted files for SAP integration
- Monitors ETSAC status changes
- Provides error handling and rollback capabilities
- Notifies stakeholders of processing status

3. System Architecture

3.1. High-Level Architecture Diagram



```
┌─────────────────────────────────────────────────────────────────┐
│                 SpeedProcessing.RDDSAPService                    │
│                                                                  │
│  ┌────────────┐    ┌────────────┐    ┌──────────────────────┐  │
│  │   Models   │    │ Business   │    │ Data Access Layer    │  │
│  │            │◄───│ Logic Layer│◄───│                      │  │
│  │            │    │            │    │                      │  │
│  └────────────┘    └────────────┘    └──────────────────────┘  │
│        ▲                 ▲                     ▲                 │
└────────┼─────────────────┼─────────────────────┼───────────────┘
         │                 │                     │
┌────────┼─────────────────┼─────────────────────┼───────────────┐
│        │                 │                     │                 │
│  ┌─────▼──────┐  ┌───────▼────┐      ┌─────────▼──┐             │
│  │ SQL Server │  │ AS400 APIs │      │  SAP APIs  │             │
│  │  Database  │  │            │      │            │             │
│  └────────────┘  └────────────┘      └────────────┘             │
│                                                                  │
│                    External Systems                              │
└─────────────────────────────────────────────────────────────────┘
```

3.2. Component Overview

The service consists of three primary layers:

1. **Data Access Layer (DAL)**:
   - Encapsulated in `SPDal.cs`
   - Handles database interactions using Dapper
   - Retrieves unprocessed tasks, part details, and ETSAC data
   - Updates processing status in the database
2. **Business Logic Layer (BLL)**:
   - Primary component is `SPService.cs`
   - Orchestrates the entire processing workflow
   - Manages API calls to AS400 and SAP systems
   - Handles file creation for SAP integration
   - Implements error handling and rollback procedures
3. **Models**:
   - Defined in `Models.cs`
   - Contains data structures for various entities
   - Includes models for part data, processing results, and API responses

3.3. Integration Points

1. **SQL Server Integration**:
   - Connects to ETA database for retrieving and updating processing data
   - Uses Dapper for efficient data access
2. **AS400 Integration**:
   - Communicates with AS400 systems via REST APIs
   - Formats data according to AS400 requirements
   - Handles insertions and deletions in AS400 tables
   - Supports rollback operations for failed transactions
3. **SAP Integration**:
   - Creates formatted text files for SAP consumption
   - Communicates with SAP APIs to verify and update data
   - Manages "-A" (additions) and "-D" (deletions) files

3.4. Configuration Management

- Environment-specific settings stored in `appsettings.json` and environment-specific variants
- Configuration categories include:
  - Database connection strings

    - API endpoints and credentials
    - Email notification settings
    - File system paths

4. Data Flow

4.1. Main Processing Flow

1. Application starts and logs initialization
2. Service retrieves unprocessed TSDR tasks from SQL Server
3. For each task:
    - Process data in AS400 systems
    - If successful, process data in SAP systems
    - If successful, update SQL Server status
    - If any step fails, execute appropriate rollback procedures
4. Send notifications about process status

4.2. RDD SAP Process Flow

1. Retrieve unprocessed SAP add/delete data from SQL Server
2. Group data by TSDR and product code
3. For each group:
    - Call SAP API to get current component data
    - Create formatted files for additions and deletions
    - Update processing status in SQL Server

4.3. ETSAC Daily Check Flow

1. Retrieve list of ETSAC daily queries
2. Execute each query against AS400
3. Check for status changes (98 to 51)
4. Log and notify about identified changes

5. Error Handling and Resilience

5.1. Error Handling Strategy

- Comprehensive exception handling at each processing stage
- Detailed logging using Serilog
- Email notifications for critical failures
- Structured error messages that include context information

5.2. Rollback Mechanisms

- AS400 Rollback: Deletes temporary data inserted during failed operations
- SAP Rollback: Removes generated files for failed operations
- Transaction-based approach ensures data consistency

6. Security Considerations

6.1. Authentication

- Database access uses dedicated user accounts
- API access uses API keys for authentication
- Credentials stored in configuration files

6.2. Data Protection

- No sensitive data is persisted in local storage
- Network communication with external systems should be secured (HTTPS)

7. Deployment Considerations

7.1. Prerequisites

- .NET runtime environment
- Network access to SQL Server, AS400, and SAP systems
- Appropriate file system permissions for SAP file creation
- SMTP server access for notifications

7.2. Configuration Steps

1. Set appropriate connection strings for the environment
2. Configure API endpoints and credentials
3. Set up logging directories
4. Configure notification email addresses
5. Set folder paths for SAP file generation

8. Maintenance and Monitoring

8.1. Logging

- Uses Serilog for structured logging
- Daily log rotation implemented
- Critical operations are explicitly logged

8.2. Notifications

- Email notifications for process completion and failures
- Targeted notifications to relevant stakeholders

9. Future Enhancements

Potential enhancements to consider:

1. Implement a more robust retry mechanism for transient failures
2. Add a dashboard for monitoring processing status
3. Enhance error reporting with more detailed contextual information
4. Implement additional validation checks before processing
5. Add performance metrics collection and reporting

10. Conclusion

SpeedProcessing.RDDSAPService provides a critical integration layer between multiple enterprise systems at Toyota Material Handling. Its well-structured architecture enables reliable synchronization of BOM data for special design requests, ensuring manufacturing accuracy and operational efficiency.