

EWM mit SAP S/4HANA – Architektur und Programmierung

Kundeneigene Erweiterungen für
Embedded und Decentralized EWM

» Hier geht's
direkt
zum Buch

DIE LESEPROBE

Kapitel 2

Architektur

Um in EWM programmieren zu können, müssen Sie die einzelnen Komponenten der Software verstehen. In diesem Kapitel beschreiben wir die Funktionen und Objekte der wichtigsten Komponenten. Anschließend erörtern wir ihre Integration mit dem SAP-ERP- oder SAP-S/4HANA-System.

Die Architektur von EWM in SAP S/4HANA umfasst eine Vielzahl von Anwendungskomponenten und Objekten, die zusammen die reichhaltige Funktionalität des Systems ausmachen. Die Funktionen und Eigenschaften einiger dieser Komponenten werden wir in diesem Kapitel detailliert besprechen. Dazu gehören:

- **Lieferabwicklung (siehe Abschnitt 2.1)**

Die Lieferabwicklung nimmt Anforderungen an das Lager zur Ein- und Auslagerung sowie zur Umlagerung oder Umbuchung von Beständen entgegen, die hauptsächlich in der Auftragsverwaltung des SAP-ERP- oder SAP-S/4HANA-Systems initiiert werden. Sie ist daher eng mit den Funktionen der Materialwirtschaft, des Vertriebs sowie der Lieferabwicklung und Produktionsplanung des SAP-ERP- oder SAP-S/4HANA-Systems verbunden. Auch mit den meisten EWM-Komponenten ist sie eng verzahnt.

- **Lagerlogistik (siehe Abschnitt 2.2)**

Die Lagerlogistik, die häufig als Kernstück der Lagerverwaltung betrachtet wird, stellt eine Vielzahl von Funktionen bereit, die für die Planung und Ausführung von Aktivitäten innerhalb des Lagers erforderlich sind. Dazu verwendet sie vorwiegend Lageraufgaben und Lageraufträge.

- **Bestandsverwaltung (siehe Abschnitt 2.3)**

Die Bestandsverwaltung ermöglicht es, die Bestände in verschiedenen räumlichen und physischen Zuständen auf der Ebene von Lagerplätzen, Ressourcen und Transporteinheiten zu verwalten. Sie ist eng mit der Bestandsführung innerhalb der Materialwirtschaft (ERP-Komponente MM) verknüpft.

- **Qualitätsprüfung (siehe Abschnitt 2.4)**

Für die Qualitätsprüfung von eingehenden und gelagerten Waren ist EWM eng mit der Anlieferung und der Bestandsführung (in EWM Bestandsverwaltung genannt) verbunden. Decentralized EWM stützt sich dabei wie SAP EWM weiterhin auf die

Quality Inspection Engine, während Embedded EWM diese Engine größtenteils überflüssig macht und direkt mit dem Qualitätsmanagement (Modul QM) von SAP S/4HANA verbunden ist.

■ Integration mit ERP-Systemen (siehe Abschnitt 2.5)

Die Integration mit ERP-Systemen kann als Querschnittskomponente von EWM betrachtet werden, da die meisten der zuvor genannten Funktionsbereiche nativ mit dem SAP-ERP- oder SAP-S/4HANA-System verbunden sind. In Abschnitt 2.5 stellen wir zahlreiche Schnittstellen für die Integration von SAP ERP bzw. SAP S/4HANA und EWM vor. Dabei gehen wir sowohl auf die Unterschiede zwischen Decentralized und Embedded EWM als auch auf die Integrationsmöglichkeiten für Nicht-SAP-ERP-Systeme ein.

Andere Komponenten von EWM, wie z. B. das Ressourcenmanagement, das Materialflusssystem, die Ausnahmenbehandlung und die Verwaltung von logistischen Zusatzleistungen, um nur einige wenige zu nennen, werden in diesem Kapitel nicht berücksichtigt, können aber ansatzweise in den folgenden Kapiteln besprochen werden. In diesem Kapitel konzentrieren wir uns auf die unserer Meinung nach zentralen und am häufigsten verwendeten Komponenten von EWM.

2.1 Lieferabwicklung

Die *EWM-Lieferung* in ihren verschiedenen Formen, die technisch oft auch als *Lageranforderung* bezeichnet wird, ist eines der Schlüsselobjekte in EWM. In diesem Abschnitt erklären wir, wie dieses recht komplexe Datenobjekt aufgebaut ist. Wir werden uns dabei an die Bezeichnung *Lieferung* halten.

Die Zu- und Abgänge von Beständen im Lager werden in der Regel auf der Grundlage von Anlieferungen und Auslieferungen in EWM ein- und ausgebucht. Wenn EWM mit einem SAP-ERP- oder SAP-S/4HANA-System integriert ist, werden EWM-Lieferungen mit Bezug auf einen entsprechenden SAP-ERP- oder SAP-S/4HANA-Beleg angelegt, z. B. eine Lieferung, die aus einer Bestellung oder einem Kundenauftrag resultiert.

Die Lieferung wird in EWM in unterschiedlichen Belegen abgebildet. Je nach Funktion und Verarbeitungsphase können verschiedene Arten von Belegen verwendet werden, die in vordefinierte Belegtypen eingeteilt werden. In Decentralized EWM werden die Belegtypen *Anforderungen* und *Benachrichtigungen* unterschieden (Kopien des zuvor angelegten SAP-ERP- oder SAP-S/4HANA-Belegs). Sie sind in einer dezentralen Umgebung unerlässlich für die Kommunikation mit SAP ERP, SAP S/4HANA oder anderen ERP-Systemen, d. h. Nicht-SAP-ERP-Systemen. Daneben gibt es *Lieferungen* und *Lieferaufträge* als Verarbeitungsbelege, die parallel zum Lagerbetrieb bearbeitet

und aktualisiert werden. Embedded EWM nutzt nur die Verarbeitungsbelege, um Redundanzen von Belegen zu beseitigen.

Tabelle 2.1 gibt einen Überblick über alle in der Lieferabwicklung in EWM verwendeten Belegtypen und nennt für jeden von ihnen die zentrale Transaktion sowie seine Verfügbarkeit in SAP S/4HANA.

| Belegtyp | Bezeichnung | EWM-Transaktion | Verfügbar in SAP S/4HANA |
|----------|---|--------------------------|---|
| IDR | Anlieferungsbenachrichtigung | /SCWM/IDN | <ul style="list-style-type: none"> ■ Embedded: nein ■ Decentralized: optional |
| ODR | Auslieferungsanforderung | /SCWM/ODR | <ul style="list-style-type: none"> ■ Embedded: nein ■ Decentralized: optional |
| POR | Umbuchungsanforderung | /SCWM/IM_DR | <ul style="list-style-type: none"> ■ Embedded: nein ■ Decentralized: optional |
| GRN | erwartete Wareneingangsbenachrichtigung | /SCWM/GRN | <ul style="list-style-type: none"> ■ Embedded: nein ■ Decentralized: optional |
| PDI | Anlieferung | /SCWM/PRDI | <ul style="list-style-type: none"> ■ Embedded: ja ■ Decentralized: ja |
| PDO | Auslieferungsauftrag | /SCWM/PRDO | <ul style="list-style-type: none"> ■ Embedded: ja ■ Decentralized: ja |
| SPC | Umbuchung | /SCWM/IM_PC | <ul style="list-style-type: none"> ■ Embedded: ja ■ Decentralized: ja |
| EGR | erwarteter Wareneingang | /SCWM/GRPE /SCWM/GRPI | <ul style="list-style-type: none"> ■ Embedded: nein ■ Decentralized: ja |
| WMR | Umlagerung | /SCWM/IM_ST | <ul style="list-style-type: none"> ■ Embedded: ja ■ Decentralized: ja |
| FDO | Auslieferung | /SCWM/FD | <ul style="list-style-type: none"> ■ Embedded: ja ■ Decentralized: ja |
| PWR | Produktionsmaterialanforderung | /SCWM/PMR | <ul style="list-style-type: none"> ■ Embedded: ja ■ Decentralized: ja |

Tabelle 2.1 Übersicht der Belegtypen und zugehörigen Transaktionen

In den folgenden Abschnitten beschreiben wir, wie die Lieferungen im logistischen Gesamtablauf eingeordnet werden, und geben einen Überblick über die wichtigsten technischen Objekte.



Produktionsmaterialanforderung

Die *Produktionsmaterialanforderung* wurde als Teil der *erweiterten Produktionsintegration* eingeführt. Sie entstammt dem Produktions- oder Prozessauftrag in SAP ERP oder SAP S/4HANA. Zur Verteilung dieses Objekts werden jedoch nicht die allgemeinen Lieferschnittstellen verwendet. Außerdem gelten die strengen Änderungsrestriktionen, wie sie sonst nach Verteilung eines Belegs aus SAP ERP oder SAP S/4HANA nach EWM üblich sind, für die Produktionsmaterialanforderung nicht.

2.1.1 Funktionen der Lieferabwicklung

Die Lieferabwicklung bietet eine Reihe von Funktionen zur Durchführung verschiedener logistischer Aktivitäten im Lager. Sie hat insbesondere die Aufgabe, Anlieferungs-, Auslieferungs- und interne Umlagerungs- und Umbuchungsanforderungen von Beständen entgegenzunehmen, zu bearbeiten und die Ausführung an das jeweilige SAP-ERP- oder SAP-S/4HANA-System zurückzumelden.

Die Komponente verwendet verschiedene, für den jeweiligen Business-Kontext maßgeschneiderte Belege. Auf diese verschiedenen Belege gehen wir in diesem Abschnitt näher ein.

Bei der Verteilung von SAP-ERP- oder SAP-S/4HANA-Lieferungen an Decentralized EWM können Benachrichtigungen oder Anforderungen angelegt werden. Aus diesen Belegen werden dann über die Transfer-Serviceklasse /SCDL/CL_TS_MANAGEMENT (Methode TRANSFER_OBJ) wiederum EWM-Lieferungen erzeugt. Die Vorgängerbelege können als inhaltliche Vorlagen für die Anlage der Lieferungen angesehen werden. Das heißt, auf der Grundlage der Daten des Vorgängerbelegs werden die EWM-spezifischen Informationen aus dem Customizing (z. B. Belegtypen, Positionstypen, Lagerprozessarten) und Stammdaten (z. B. Lagerprodukt, Lagerplatz, Charge) gelesen, um die Lieferung entsprechend anzureichern. Anhand von Belegfluss und Statusverwaltung werden die Aktivitäten, die während der Bearbeitung der Lieferung von den Mitarbeitenden im System durchgeführt wurden, in den Belegen dokumentiert. So haben Sie immer einen Überblick über den aktuellen Bearbeitungsfortschritt des jeweiligen Belegs im System.

Einen schematischen Überblick über die Beziehung zwischen Benachrichtigungen und Lieferungen in Decentralized und Embedded EWM finden Sie in Abbildung 2.1. Die Darstellung zeigt, welche Lieferobjekte aus einem SAP-ERP- oder SAP-S/4HANA-System letztendlich welche EWM-Lieferungen erzeugen.

Hinsichtlich der Architektur der Belege gibt es jedoch keinen Unterschied zwischen Benachrichtigungen und Lieferungen. Technisch gesehen sind alle Belege in der Lieferabwicklung identisch aufgebaut und modelliert. Dafür sorgt ein gemeinsames Grundgerüst, das wir im folgenden Abschnitt vorstellen werden.

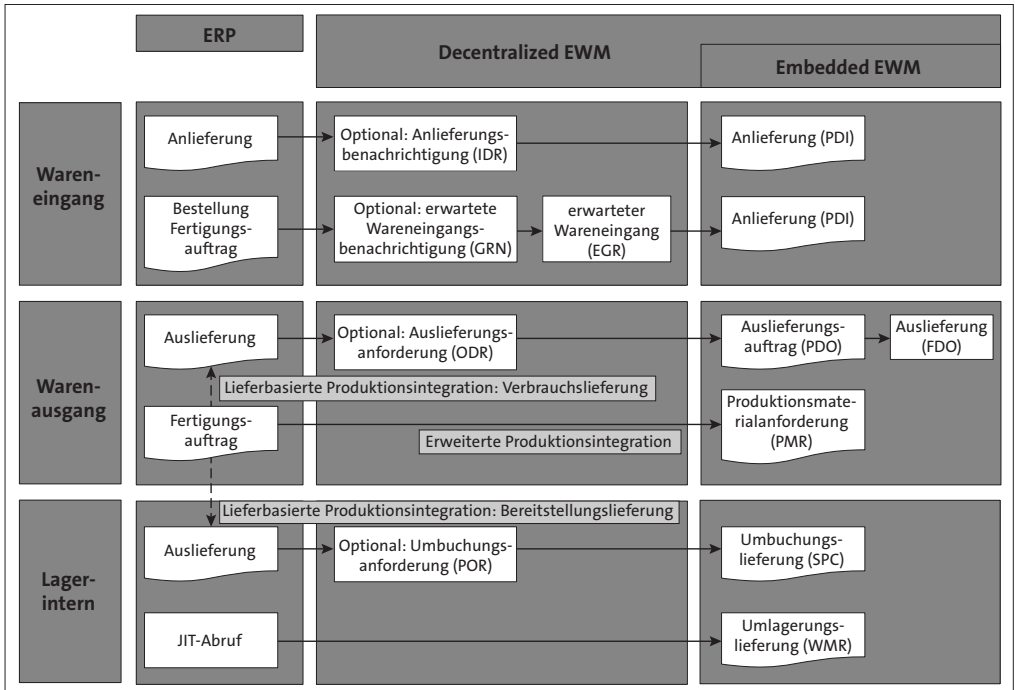


Abbildung 2.1 Beziehung zwischen Benachrichtigungen und Lieferungen in den beiden EWM-Deployment-Optionen

EWM-Lieferabwicklung in SAP S/4HANA

Weitere Informationen zu den von EWM in SAP S/4HANA unterstützten Prozessen der Lieferabwicklung und Hinweise auf Konfigurationsmöglichkeiten finden Sie in der Anwendungsdokumentation im SAP Help Portal (<http://help.sap.com/>). Folgen Sie auf der Startseite diesem Pfad: **SAP S/4HANA • Product Assistance • Geschäftsanwendungen für Unternehmen • Supply Chain • Lagerung • Extended Warehouse Management (EWM) • Lieferabwicklung.**

2.1.2 Datenmodell der EWM-Lieferung

Die Objekte für die Lieferungen in EWM wurden auf der Basis von ABAP Objects modelliert. Beim Lesen dieser Objekte werden Instanzen angelegt, die technisch die jeweiligen Lieferungen darstellen. Auf diese Objektinstanzen kann dann zugegriffen werden, um bestimmte Aspekte des Objekts auf Belegkopf- oder auf Positionsebene zu aktualisieren, z. B. Liefertermine.

Neben dem reinen Lesen von Datenbankdaten laufen beim Lesen bzw. Anlegen solcher Objektinstanzen weitere automatische Prozesse im Hintergrund ab; unter ande-

rem werden auf den Objektinstanzen Ermittlungen und Validierungen durchgeführt, um dynamische Daten, z. B. aggregierte Mengen, Zustände usw., auszuwerten und zu ändern.

In den folgenden Abschnitten erläutern wir die Modellierung des Lieferobjekts in EWM näher. Wir beginnen im folgenden Abschnitt mit dem Modellierungs-Framework *Business Object Processing Framework* (BOPF) und sehen uns dann einige Schlüsselaspekte des Objekts und des Datenzugriffs an. Dazu erläutern wir die Art und Weise, wie Ermittlungen, Validierungen und Aktionen auf EWM-Lieferungen durchgeführt werden.

Business Object Processing Framework (BOPF)

BOPF ist ein Framework für die Implementierung von Business-Objekten, das den Prinzipien einer *serviceorientierten Architektur* (SOA) folgt. Es enthält die notwendigen Funktionen für die Implementierung von Objekten zur Abbildung von Geschäftsbelegen und unterstützt einen einheitlichen Modellierungsstandard. Das SOA-Konzept basiert hauptsächlich auf *Enterprise Services*, die einem strengen Governance-Prozess unterliegen, um eine einheitliche betriebswirtschaftliche Semantik für bestehende und zukünftige Anwendungen sicherzustellen. Die von SAP gelieferten Enterprise Services sind entlang von Geschäftsprozessen strukturiert und können zu automatisierten Prozessabläufen zusammengefügt werden.

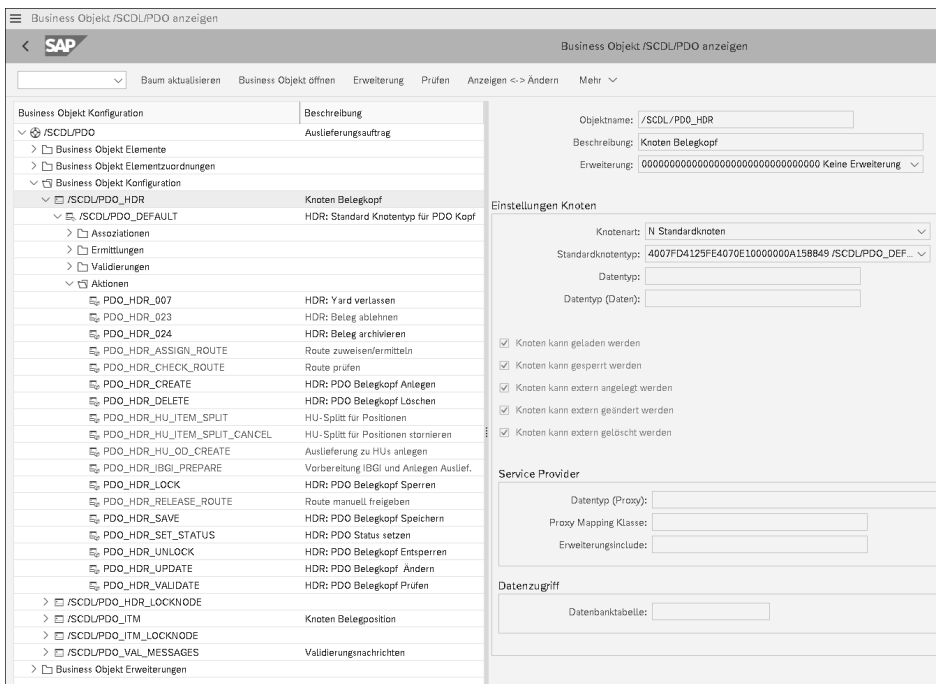


Abbildung 2.2 BOPF-Konfigurationsoberfläche zu Objekt /SCDL/PDO

BOPF kontrolliert die Geschäftslogik der Anwendung und sorgt für die Bereitstellung, Pufferung und Speicherung von Daten. Somit sind Geschäftslogik und Datenverwaltung strikt voneinander getrennt, ebenso wie die Änderung und Kontrolle der verwalteten Daten. Aus Sicht der Softwarekomponente der Applikationsplattform (AP) ist das heutige BOPF, wie es in verschiedenen SAP-Produkten verwendet wird (z. B. SAP Transportation Management (SAP TM)), nicht mehr mit dem in EWM verwendeten BOPF kompatibel; die Grundlagen sind jedoch in der EWM-Lieferabwicklung noch erkennbar. Über die Transaktion /BOPF/CONF_UI erhalten Sie ein Einblick in die klar strukturierte Modellierung der verschiedenen Belegtypen der EWM-Lieferabwicklung (siehe Abbildung 2.2).

Aufbau des EWM-Lieferobjekts

Ein EWM-Lieferobjekt besteht immer aus einem *Lieferkopf* und mindestens einer *Lieferposition*. Die technische Beziehung ergibt sich aus dem Schlüssel, der die DOCID (global eindeutige interne Kopfidentifikation) und die ITEMID (global eindeutige interne Positionsidentifikation) beinhaltet. Darüber hinaus werden standardmäßig die folgenden weiteren Objekteigenschaften, *Aspekte* genannt, zugewiesen:

- Status
- Termine
- Referenzbelege
- aggregierte Mengen
- Lokationen
- Geschäftspartner
- Belegfluss

Zusätzliche Datenobjekte, die Sie optional verwenden können, wie z. B. Handling Units, Transporteinheiten, Texte usw., werden ebenfalls von den entsprechenden Anwendungen mit der Lieferung verknüpft. Abbildung 2.3 zeigt grob den Aufbau der Lieferung bzw. des Lieferobjekts.

Ein Lieferobjekt wird auf der Kopfebene durch Belegtypen dargestellt (siehe auch Tabelle 2.1) sowie durch Positionstypen auf der Positionsebene. Diese Beleg- und Positionstypen bilden die Grundlage, um die betriebswirtschaftlichen Eigenschaften eines jeden Belegs zu kategorisieren.

Ab dem Release SAP S/4HANA 2022 gibt es in EWM folgende Positionstypen für die Belegtypen:

- DLV: Normal-Lieferposition (PDI und PDO)
- PAC: Verpackungsposition (PDI und PDO)
- RET: Retourenposition (PDI und PDO)

- TXT: Textposition (PDI und PDO)
- CGO: Transitposition (PDI und PDO)
- WIP: Ware in Arbeit (PDI)
- VAL: Wertposition (PDO)
- CMP: Komponente (PWR)
- OCP: Komponente Ausgang (PWR)

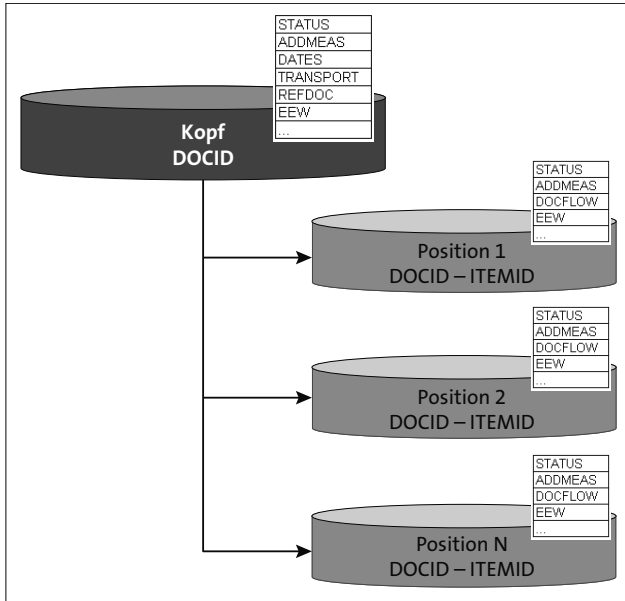


Abbildung 2.3 Aufbau des Lieferobjekts in EWM

Darüber hinaus werden auf der Positionsebene unter anderem folgende Hierarchieebenen unterschieden:

- Hauptposition (DSP)
- Chargensplitposition (BSP)
- Liefersplitposition (FSD)

Die gesplitteten Positionen können in EWM z. B. auftreten, wenn Sie den Warenausgang nur für eine bestimmte Position und nicht für die gesamte Lieferung buchen (Liefersplit) oder wenn Sie verschiedene Chargen für eine einzige Hauptposition kommissionieren (Chargensplit). Splitpositionen stehen immer in einer festen Hierarchie zur Hauptposition.

Tabelle 2.2 zeigt die Datenbanktabellen für Kopf- und Positionsdaten in der Lieferabwicklung. Weitere Datenbanktabellen für die genannten Objekte finden Sie in den ABAP-Dictionary-Objekten des Pakets /SCDL/DATA_MODEL.

| Datenbanktabelle | Bezeichnung |
|------------------|---|
| /SCDL/DB_REQH | Anlieferungsbenachrichtigung/ Auslieferungsanforderung: Kopf |
| /SCDL/DB_REQI | Anlieferungsbenachrichtigung/ Auslieferungsanforderung: Position |
| /SCDL/DB_PROCH_I | Anlieferung: Kopf |
| /SCDL/DB_PROCI_I | Anlieferung: Position |
| /SCDL/DB_PROCH_O | Auslieferungsauftrag: Kopf |
| /SCDL/DB_PROCI_O | Auslieferungsauftrag: Position |
| /SCDL/DB_DLVI_O | Auslieferung: Kopf |
| /SCDL/DB_DLVI_O | Auslieferung: Position |
| /SCDL/DB_PROCH_P | Produktionsmaterialanforderung: Kopf |
| /SCDL/DB_PROCI_P | Produktionsmaterialanforderung: Position |

Tabelle 2.2 Datenbanktabellen für Lieferungskopf und -position

Struktur und Konzept des Lieferobjekts in EWM erlauben es, auf einzelne Positionen zuzugreifen und diese fortzuschreiben, ohne die gesamte Lieferung zu sperren. Dies hat in der Praxis den großen Vorteil, dass verschiedene Positionen einer Lieferung parallel bearbeitet werden können, z. B. bei der Kommissionierung, ohne Sperrkonflikte hervorzurufen.

Aus Sicht der Performance ergibt sich daraus noch ein weiterer Vorteil: Prozesse, die mit Lieferdaten arbeiten, können auf Positionsebene verwaltet werden. So müssen nur die betroffenen Positionsdaten und nicht alle anderen Positionen des jeweiligen Lieferbelegs in den Arbeitsspeicher geladen werden.

Service Provider und Aspekte

Damit alle EWM-Anwendungen über eine zentrale Schnittstelle auf ein Lieferobjekt zugreifen können, werden sogenannte *Service Provider* (SP) verwendet. Diese Service Provider bestehen aus Interfaces und ihren jeweiligen Methoden, die es den Anwendungen ermöglichen, Aktionen für ein bestimmtes *Business-Objekt* (BO) auszuführen. Außerdem dienen die Service Provider dazu, die angeforderten Daten aus dem Business-Objekt zurückzugeben, gut aufbereitet in den jeweiligen Strukturen der aufrufenden Anwendung. Diese Architektur ermöglicht eine generische Nutzung des Business-Objekts, die auf die spezifischen Anforderungen der einzelnen EWM-Anwendungen zugeschnitten ist.

In EWM gibt es zwei zentrale Service Provider:

- /SCDL/CL_SP: Service Provider der Lieferung
- /SCWM/CL_SP: UI Service Provider

Diese abstrakten Klassen bilden mit ihren Interfaces die Grundlage für die speziellen Anwendungsbereiche innerhalb der EWM-Lieferabwicklung. Der Hauptunterschied zwischen den Klassen /SCDL/CL_SP und /SCWM/CL_SP besteht darin, dass die UI Service Provider – abhängig von der jeweiligen Transaktion – diverse Felder oder Eigenschaften der Business-Objekte ausblenden und zusätzliche Prüfungen durchführen, bevor das Objekt fortgeschrieben wird. Außerdem werden die Objektdaten durch die UI Service Provider um Kurzbeschreibungen angereichert. Die verschiedenen Ausprägungen der Service Provider der EWM-Lieferabwicklung sind grafisch in Abbildung 2.4 dargestellt.

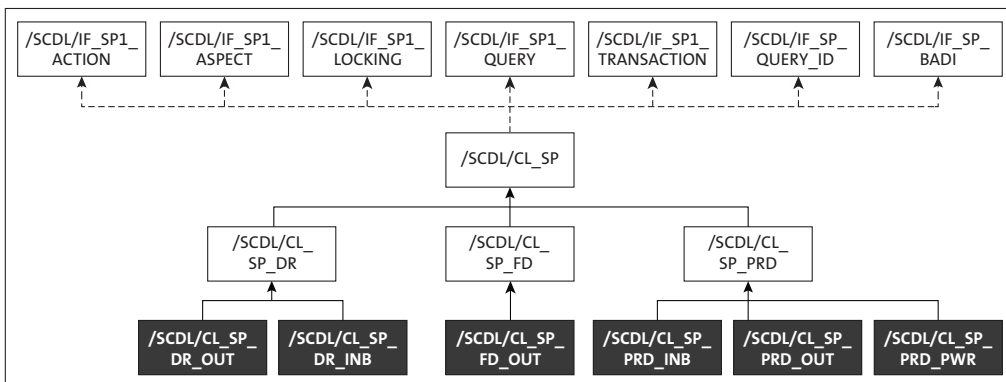


Abbildung 2.4 Architektur der Service Provider /SCDL/CL_SP

Die Strukturen, über die die Service Provider mit den Anwendungsprogrammen kommunizieren, werden *Aspekte* genannt. Ein Aspekt repräsentiert jeweils ein bestimmtes Attribut des Business-Objekts. So gibt es z. B. einen Aspekt für Termindaten auf der Kopfebene (/SCDL/S_SP_A_HEAD_DATE) und einen Aspekt für Termindaten auf der Positionsebene (/SCDL/S_SP_A_ITEM_DATE).

In der serviceorientierten Architektur wird zwischen *Schlüsselaspekten* und Aspekten unterschieden. Dabei enthält der Schlüsselaspekt immer den semantischen Schlüssel eines Objekts (ähnlich einem Primärschlüssel). Alle Aspekte müssen einem Schlüsselaspekt zugeordnet werden, um eindeutig identifizierbar zu sein.

Um dem hoch dynamischen Ansatz dieser Architektur Rechnung zu tragen, gibt es zu den jeweiligen Aspekten keine spezifischen Repository-Objekte. Daher muss beim Aufruf der SP-Methoden sichergestellt werden, dass der gewünschte Kontext übergeben wird. Die entsprechenden Definitionen finden sich als Konstanten innerhalb der beiden Interfaces /SCDL/IF_SP_C und /SCWM/IF_SP_C.

Das Codingbeispiel in Listing 2.1 zeigt Ihnen beispielhaft, wie Sie einige Aspekte und die wichtigsten SP-Methoden nutzen können. Zunächst legen Sie Objektinstanzen mit Referenz auf die SP-Klassen /SCDL/CL_SP_PRD_OUT (Auslieferungsauftrag) und /SCDL/CL_SP_MESSAGE_BOX (Sammler für Nachrichten in der Lieferabwicklung) an. Wir verwenden hier die Klasse /SCDL/CL_SP_PRD_OUT, weil wir eine Auslieferung selektieren wollen. Die Message-Box wird wiederum benötigt, um eventuelle Nachrichten zu erfassen, z. B. im Fehlerfall.

Beispielreport für die Lieferabwicklung

Das in den folgenden Listings dieses Kapitels vorgestellte Coding ist Teil eines Beispielprogramms, das Sie in Ihr eigenes System kopieren können, um sich mit der Verarbeitung von EWM-Lieferungen vertraut zu machen. Kopieren Sie die Codingteile einfach der Reihe nach. Sie können das Coding auch von der GitHub-Plattform unter <https://github.com/> abrufen, wenn Sie den abapGit-Client installiert haben, mit dem Sie direkt von Ihrem SAP-System aus mit GitHub interagieren können. Das Coding ist im GitHub-Repository `ewmdevbook_2.1` zu finden.

```
*&-----*
*& Report ZEWMDEVBOOK_21
*&-----*
*& Example Report for Delivery Processing
*&-----*
REPORT zewmdevbook_21.

PARAMETERS: p_lgnum TYPE /scwm/lgnum OBLIGATORY,
             p_docno TYPE /scdl/dl_docno OBLIGATORY,
             p_itemno TYPE /scdl/dl_itemno OBLIGATORY,
             p_wt_fm RADIOBUTTON GROUP wtrcr,
             p_wt_api RADIOBUTTON GROUP wtrcr.

"2.1 Object Instances /SCDL/ SP
BREAK-POINT ID zewmdevbook_21.
TRY.
    DATA(lo_message_box) = NEW /scdl/cl_sp_message_box( ).
    DATA(lo_sp)          = NEW /scdl/cl_sp_prd_out(
        io_message_box = lo_message_box
        iv_doccat      = /scdl/if_dl_doc_c=>sc_doccat_out_prd
        iv_mode        = /scdl/cl_sp=>sc_mode_classic ).
```



```
CATCH /scdl/cx_delivery.  
ENDTRY.
```

Listing 2.1 Erstellen der erforderlichen Objektinstanzen

Selektieren Sie nun einen Positionsaspekt über die Nummer einer Lieferung in SAP ERP oder SAP S/4HANA und die zugehörigen Positionsnummern. Wie das funktioniert, zeigen wir in Listing 2.2. Im folgenden Abschnitt erklären wir, was mit den komplexen Selektionskriterien geschieht.

```
"2.2 Example /SCDL/ SP based Query  
DATA: lt_bopf_items TYPE /scdl/t_sp_a_item.
```

```
lo_sp->query(  
  EXPORTING  
    query      = /scdl/if_sp_c=>sc_qry_item  
    selections = VALUE /scdl/t_sp_selection(  
      ( fieldname = /scdl/if_dl_logfname_c=>sc_docno_h  
        sign      = wmegc_sign_inclusive  
        option    = wmegc_option_eq  
        low       = |{ p_docno ALPHA = IN }| )  
      ( fieldname = /scdl/if_dl_logfname_c=>sc_itemno_i  
        sign      = wmegc_sign_inclusive  
        option    = wmegc_option_eq  
        low       = |{ p_itemno ALPHA = IN }| ) )  
  IMPORTING  
    outrecords = lt_bopf_items  
    rejected   = DATA(lv_rejected) ).  
IF lv_rejected = abap_true.  
  DATA(lt_messages) = lo_message_box->get_messages( ).  
  CALL METHOD /scwm/cl_tm=>cleanup( ).  
  EXIT.  
ENDIF.
```

Listing 2.2 Beispielaufruf der Methode /SCDL/IF_SP1_QUERY~EXECUTE

Der Rückgabewert `lt_bopf_items` enthält sowohl das Feld `DOCID` als auch das Feld `ITEMID`, die GUIDs für Kopf und Position. Beide Werte werden benötigt, um den Schlüsselaspekt `/scdl/t_sp_k_item` weiter zu verwenden. Als Nächstes können Sie den Aspekt für Ihre Position selektieren, der die Lieferbedingungen enthält (siehe Listing 2.3).

"2.3 Example /SCDL/ SP based Aspect Selection

DATA: lt_a_item_delterms TYPE /scdl/t_sp_a_item_delterm.

```
lo_sp->select(
  EXPORTING
    inkeys      = CORRESPONDING /scdl/t_sp_k_item( lt_bopf_items )
    aspect      = /scdl/if_sp_c=>sc_esp_item_delterm
  IMPORTING
    outrecords  = lt_a_item_delterms
    rejected   = lv_rejected
    return_codes = DATA(lt_return_codes) ).
```

```
IF lv_rejected = abap_true.
  lt_messages = lo_message_box->get_messages( ).
  CALL METHOD /scwm/cl_tm=>cleanup( ).
  EXIT.
ELSEIF line_exists( lt_return_codes[ failed = abap_true ] ).
  lt_messages = lo_message_box->get_messages( ).
  CALL METHOD /scwm/cl_tm=>cleanup( ).
  EXIT.
ENDIF.
```

Listing 2.3 Beispielaufruf der Methode /SCDL/IF_SP1_ASPECT~SELECT

Falls Sie die Lieferposition um kundeneigene Felder erweitert haben und diese Felder lesen oder aktualisieren möchten, verwenden Sie anstelle von SC_ASP_ITEM_DELTERM den Aspekt SC_ASP_ITEM_EEW_PRD. Wenn Sie keine bestimmte Position für die Lieferung lesen wollen oder wenn Sie zunächst alle Positionen der Lieferung ermitteln wollen, könnten Sie den Aufruf in Listing 2.4 verwenden.

"2.4 Example /SCDL/ SP based Aspect Selection by Relation

```
lo_sp->select_by_relation(
  EXPORTING
    relation = /scdl/if_sp_c=>sc_rel_head_to_item
    inrecords = CORRESPONDING /scdl/t_sp_k_head( lt_bopf_items )
    aspect   = /scdl/if_sp_c=>sc_esp_head
  IMPORTING
    outrecords  = lt_bopf_items
    rejected   = lv_rejected
    return_codes = lt_return_codes ).
```

```
SORT lt_bopf_items ASCENDING.
DELETE ADJACENT DUPLICATES FROM lt_bopf_items.
LOOP AT lt_bopf_items ASSIGNING FIELD-SYMBOL(<bopf_item>).
```

```
WRITE: / 'BOPF',  
      <bopf_item>-itemno,  
      <bopf_item>-itemcat,  
      <bopf_item>-itemtype.  
ENDLOOP.
```

Listing 2.4 Beispielaufruf der Methode /SCDL/IF_SP1_ASPECT~SELECT_BY_RELATION

Wir empfehlen, Aspekte nur dann zu verwenden, wenn Sie bestimmte Einzelinformationen aus einem Lieferobjekt selektieren müssen oder wenn Sie bestimmte Kundenfelder fortschreiben möchten. Sie sollten Standardfelder der Aspekte nur in Ausnahmefällen mit den hier verwendeten Methoden überschreiben, und auch nur dann, wenn Sie sich absolut sicher sind, welche Auswirkungen dies hat; andernfalls kann es zu Dateninkonsistenzen kommen.

Im Allgemeinen benötigt ein Lieferobjekt jedoch verschiedene Informationen aus unterschiedlichen Aspekten. Um diese Informationen zu übermitteln, gibt es weitere Methoden, die wir im nächsten Abschnitt erläutern werden.

Verwendung der EWM-Liefer-Query

Um EWM-Lieferungen zu selektieren, ohne direkt einzelne Aspekte verwenden zu müssen, bietet SAP die Methode `QUERY` an. Diese Methode gehört zu einer der Hauptklassen des Objektmanagers `/SCWM/CL_DLV_MANAGEMENT`. Je nachdem, welche Art von Belegtyp Sie lesen möchten, müssen Sie die entsprechende Klasse verwenden. Anlieferungsbenachrichtigungen werden z. B. mit einer anderen Klasse gelesen als Anlieferungen. Tabelle 2.3 zeigt die Klassenreferenzen, die Sie verwenden sollten, um mit der Methode `QUERY` die Objekte nach den einzelnen Belegtypen zu selektieren.

| Dokumentkategorie | Klasse |
|-------------------|-----------------------------|
| IDR | /SCWM/CL_DLV_MANAGEMENT_DR |
| ODR | /SCWM/CL_DLV_MANAGEMENT_DR |
| POR | /SCWM/CL_DLV_MANAGEMENT_DR |
| EGR | /SCWM/CL_DLV_MANAGEMENT_DR |
| PDI | /SCWM/CL_DLV_MANAGEMENT_PRD |
| PDO | /SCWM/CL_DLV_MANAGEMENT_PRD |
| SPC | /SCWM/CL_DLV_MANAGEMENT_PRD |
| WMR | /SCWM/CL_DLV_MANAGEMENT_PRD |

Tabelle 2.3 Belegtypen und zugehörige Query-Klassen

| Dokumentkategorie | Klasse |
|-------------------|-----------------------------|
| FDO | /SCWM/CL_DLV_MANAGEMENT_FD |
| PWR | /SCWM/CL_DLV_MANAGEMENT_PRD |

Tabelle 2.3 Belegtypen und zugehörige Query-Klassen (Forts.)

Das Programmierbeispiel in Listing 2.5 zeigt Ihnen, wie die Query-Methode des UI Service Providers verwendet wird, um die Kopf- und Positionsinformationen eines Auslieferungsauftrags mit komplexen Selektionskriterien über eine Lieferungsnummer aus SAP ERP oder SAP S/4HANA zu lesen (Exportparameter IT_SELECTION).

```
"2.5 Example Delivery EWM Service Provider based Query
BREAK-POINT ID zewmdevbook_21.
"Get instance of service provider class
DATA(lo_delivery) = NEW /scwm/cl_dlv_management_prd( ).
"Call query method of service provider class
TRY.
    CALL METHOD lo_delivery->query
        EXPORTING
            iv_doccat      = /scdl/if_dl_c=>sc_doccat_out_prd
            it_selection   = VALUE /scwm/dlv_selection_tab(
                ( fieldname = /scdl/if_dl_logfname_c=>sc_docno_h
                  sign      = wmegc_sign_inclusive
                  option    = wmegc_option_eq
                  low       = |{ p_docno ALPHA = IN }| ) )
            is_read_options = VALUE #(
data_retrival_only      = abap_true
mix_in_object_instances = /scwm/if_dl_c=>sc_mix_in_load_instance )
        IMPORTING
            et_headers    = DATA(lt_srv_headers)
            et_items      = DATA(lt_srv_items)
            eo_message    = DATA(lo_message).
    IF lo_message IS BOUND.
        DATA(lt_message) = lo_message->get_messages( ).
    ENDIF.
    CATCH /scdl/cx_delivery INTO DATA(lx_delivery).
    IF lx_delivery->mo_message IS BOUND.
        lo_message->add( lx_delivery->mo_message ).
    ENDIF.
ENDTRY.

LOOP AT lt_srv_headers ASSIGNING FIELD-SYMBOL(<srv_header>).
```



```
WRITE: / 'SRV ',
        <srv_header>-docno,
        <srv_header>-doccat,
        <srv_header>-doctype.
ENDLOOP.
```

Listing 2.5 Beispielaufruf einer EWM-Service-Provider-basierten Lieferungsselektion

Wenn Sie komplexere Selektionskriterien verwenden wollen, müssen Sie den Zusammenhang zwischen den entsprechenden Datenbankfeldern – in unserem Beispiel die SAP-ERP- oder SAP-S/4HANA-Liefernummer – und den logischen Feldnamen kennen. Diese Zuordnung finden Sie im EWM-Einführungsleitfaden unter **Prozessübergreifende Einstellungen • Lieferung – Lageranforderung • Erweiterung von Lieferung – Lageranforderung • Logische Feldnamen definieren**.

Alternativ können Sie die Konstanten für die logischen Feldnamen auch in den folgenden Interfaces finden:

- /SCDL/IF_DL_LOGFNAME_C
- /SCWM/IF_DL_LOGFNAME_C

Damit stellt das System neben den Kopf- oder Positionsdaten noch weitere lieferungsbezogene Informationen zur Verfügung. Je nach verwendeter Klasse können dies z. B. zugeordnete Handling Units oder Transporteinheiten sein.

Die Selektion der Lieferobjekte kann unter Umständen starken Einfluss auf die Performance eines Prozesses haben. Das hängt nicht nur davon ab, wie viele Belege und Positionen zu lesen sind, sondern auch davon, welche Zustände oder Werte aus den Datenbankinformationen zur Laufzeit berechnet werden müssen. Spezielle Kopf- und Positionsinformationen, z. B. Statuswerte und Mengenrollen, sind dynamisch (transienter Zustand) und müssen zur Laufzeit auf Basis anderer statischer Werte (persistenter Zustand) berechnet werden. Eine Übersicht über die Merkmale eines Status finden Sie im EWM-Einführungsleitfaden unter **Prozessübergreifende Einstellungen • Lieferung – Lageranforderung • Statusverwaltung • Statusprofile definieren**.

Beachten Sie, dass es auf Kopf- und Positionsebene unterschiedliche Konfigurationen für einen Status geben kann. Bei Auslieferungsaufträgen z. B. ist der Status DPI (Kommissionierung) auf Positionsebene persistent. Auf der Kopfebene wird er jedoch zur Laufzeit aus dem Status der Positionen berechnet und ist daher transient.

Um das Selektionsverhalten der Liefer-Query zu steuern, liefert SAP die folgenden Strukturen als Importparameter aus:

- IS_READ_OPTIONS steuert das Leseverhalten.
- IS_INCLUDE_DATA bezieht bestimmte Aspekte ein.
- IS_EXCLUDE_DATA schließt bestimmte Aspekte aus.

Der Parameter `IS_EXCLUDE_DATA` ist jedoch veraltet und wird hier nur der Vollständigkeit halber erwähnt. Sie sollten ihn in Ihren Implementierungen nicht verwenden.

Bei der Programmierung sollten Sie im jeweiligen Kontext genau prüfen, wozu Sie das Lieferobjekt benötigen (Lese- oder Schreibzugriffe). Prüfen Sie darüber hinaus, welche Daten in den Selektionsergebnissen wirklich erforderlich sind.

Query-Steuerung bei der Datenselektion

In einer Abfrage spezieller Auslieferungen möchten Sie ermitteln, für welche dieser Belege der Warenausgang bereits gebucht wurde. Der zu prüfende Status auf Positionsebene heißt `DGI` und ist persistent. Sie würden also die Liefer-Query für einen Nur-Lese-Zugriff aufrufen und benötigen nur `ET_ITEMS` als Rückgabeparameter. Eine Objektinstanz anzulegen und dynamische Werte zu berechnen ist nicht notwendig. Entsprechend können Sie das Kennzeichen `DATA_RETRIVAL_ONLY` in der Struktur `IS_READ_OPTIONS` verwenden und den Strukturwert `IS_INCLUDE_DATA-ITEM_STATUS` setzen.

Eine detaillierte Beschreibung, wie die speziellen Selektionsoptionen verwendet werden können, finden Sie in der Dokumentation der Methode `QUERY` der Klasse `/SCWM/CL_DLV_MANAGEMENT_PRD`. In jedem Fall sollten Sie vor jedem Aufruf der Liefer-Query aus Performancegründen die entsprechenden Selektionsoptionen setzen.

Ein weiterer wesentlicher Bestandteil von Lieferungen sind die jeweiligen Positionsmengen. EWM bietet verschiedene Funktionen zur Bearbeitung von Lieferpositionen, für die die Statusverwaltung allein nicht ausreicht. Stellen Sie sich z. B. vor, Sie möchten genau wissen, welche Menge bei einem Teilwareneingang noch nicht vom System verbucht wurde, d. h., welche Menge noch eingehen muss. Dieser Aspekt wird über einzelne *Mengenrollen* abgebildet.

Die aktuellen Mengen einer Lieferposition finden Sie auf der Benutzeroberfläche der Lieferung auf der Registerkarte **Zusätzl. Mengen**. Ähnlich wie bei der Statusverwaltung gibt es auch bei der Mengenverrechnung persistente und transiente Werte. Zum Beispiel wird die Mengenrolle `W1` (Kommissionierung geplant) aus dem Belegfluss einer Position zur Laufzeit berechnet.

Der *Belegfluss* ist technisch gesehen ein separates Framework, weshalb wir an dieser Stelle nicht näher darauf eingehen werden. Im Wesentlichen stellt er Knotenbeziehungen dar, die Informationen über bestimmte Aktionen zu einer Lieferposition enthalten.

Eine Übersicht über die Eigenschaften aller Mengenrollen sowie deren Ermittlung finden Sie im EWM-Einführungsleitfaden unter **Prozessübergreifende Einstellungen**

- **Lieferung – Lageranforderung**
- **Mengenverrechnung**.



Das Beispiel in Listing 2.6 zeigt, wie man die Positionshierarchie sowie Status und Mengenrollen auswertet. Wir verwenden die bereits selektierte Positionstabelle `lt_srv_items` und prüfen zunächst, ob Splitpositionen existieren.

"2.6 Example UI SP Delivery Hierarchy

```
LOOP AT lt_srv_items ASSIGNING FIELD-SYMBOL(<srv_item>).
  "Check hierarchy
  DATA(lo_corr) = /scwm/cl_dlv_correlation=>get_instance( ).
  LOOP AT <srv_item>-hierarchy ASSIGNING FIELD-SYMBOL(<srv_item_hierarchy>).
    TRY.
      CALL METHOD lo_corr->get_hier_cat
        EXPORTING
          iv_hierarchy_type = <srv_item_hierarchy>-hierarchy_type
        IMPORTING
          ev_hierarchy_cat = DATA(lv_cat).
      CATCH /bopf/cx_frw.
    ENDTRY.
    "Skip split items
    IF lv_cat = /scdl/if_dl_hierarchy_c=>sc_cat_ssp
      AND <srv_item_hierarchy>-parent_object IS NOT INITIAL.
      DATA(lv_skip) = abap_true.
      EXIT.
    ENDIF.
  ENDMETHOD.
ENDLOOP.
IF lv_skip = abap_true.
  DELETE lt_srv_items.
  CONTINUE.
ENDIF.
```

Listing 2.6 Hierarchie der Lieferpositionen auswerten

Die Serviceklassen, die für die Hierarchieermittlung verwendet werden, sind unter `/SCWM/CL_DLV*` im Paket `/SCWM/DELIVERY` zusammengefasst. In diesem Paket werden die Kernfunktionen der Lieferabwicklung entwickelt.

Nun schauen wir uns den Status DPI (Kommissionierung) an, um die Positionen zu ermitteln, die noch nicht vollständig kommissioniert wurden, vorausgesetzt, sie sind überhaupt für die Kommissionierung relevant (siehe Listing 2.7).

```
TRY.
  DATA(ls_status) = <srv_item>-status[
    status_type = /scdl/if_dl_status_c=>sc_t_picking ].

  IF ls_status-status_value = /scdl/if_dl_status_c=>sc_v_not_relevant.
    DELETE lt_srv_items.
```

```

CONTINUE.
ELSEIF ls_status-status_value NE /scdl/if_dl_status_c=>sc_v_finished.
  "Item & not yet completely picked.
  MESSAGE i001(zewmdevbook_21) WITH <srv_item>-itemno.
CONTINUE.
ENDIF.
CATCH cx_sy_itab_line_not_found.
  "Item & not relevant for picking.
  MESSAGE i003(zewmdevbook_21) WITH <srv_item>-itemno.
ENDTRY.

```

Listing 2.7 Auswerten eines Lieferpositionsstatus

Schließlich prüfen wir über die Mengenrolle PA (Packen) für Positionen, die nach der Kommissionierung noch nicht vollständig verpackt wurden, ob sie überhaupt für das Verpacken relevant sind (siehe Listing 2.8).

```

TRY.
  DATA(ls_addmeas) = <srv_item>-addmeas[
    qty_role = /scdl/if_dl_addmeas_c=>sc_qtyrole_pack
    qty_category = /scdl/if_dl_addmeas_c=>sc_qtycat_open ].
  IF ls_addmeas-qty NE 0.
    "Item & not yet completely packed.
    MESSAGE i002(zewmdevbook_21) WITH <srv_item>-itemno.
  ENDIF.
  CATCH cx_sy_itab_line_not_found.
    "Item & not relevant for packing.
    MESSAGE i004(zewmdevbook_21) WITH <srv_item>-itemno.
  ENDTRY.
ENDLOOP.

```

Listing 2.8 Auswerten einer Lieferpositionsmengenrolle

Alternative Ansätze für die Arbeit mit EWM-Datenobjekten sind innerhalb von SAP S/4HANA-APIs verfügbar gemacht worden. Diese wurden teilweise im Rahmen der Bereitstellung von Lagerverwaltungsfunktionalität in SAP S/4HANA Cloud eingeführt. Es wurden mehrere SAPUI5-basierte Transaktionen geschaffen, die in einer solchen Umgebung zusammen mit neuen OData-Services verwendet werden können. Die APIs können als Wrapping Layer um die klassischen Funktionsbausteine betrachtet werden und ermöglichen eine durchgehend objektorientierte Programmierung und den externen Zugriff. Im Allgemeinen wird empfohlen, mit den verfügbaren APIs zu arbeiten, bevor man sich den Methoden auf der Ebene des /SCDL/-Service-Providers zuwendet, da diese eine einfachere Parameterbehandlung ermöglichen.

Listing 2.9 zeigt ein Beispiel für die Verwendung solcher APIs im Rahmen der Selektion von Lieferungsdaten nach der zuvor angegebenen Liefernummer.

```
"2.7 Example Delivery API based Query
BREAK-POINT ID zewmdevbook_21.
"Set warehouse request of type Outbound Delivery Order
DATA: lo_whr_api_outb TYPE REF TO /scwm/if_api_whr_outbound.
TRY.
    /scwm/cl_api_factory=>get_service( IMPORTING eo_api = lo_whr_api_outb ).
    "Set warehouse number obligatory for API
    /scwm/cl_tm=>set_lgnum( p_lgnum ).
    "Map business keys to warehouse request keys and read ODOs
    lo_whr_api_outb->/scwm/if_api_warehouse_request~get_keys_for_bus_keys(
        EXPORTING
            it_whr_bus_keys =
                VALUE /scwm/if_api_warehouse_request=>yt_whr_bus_key(
                    ( docno = |{ p_docno ALPHA = IN }| ) )
        IMPORTING
            et_whr_keymap = DATA(et_keys_map) ).

    lo_whr_api_outb->read_outbound_dlv_order(
        EXPORTING
            it_whr_key = CORRESPONDING #( et_keys_map )
            is_include = VALUE #( head_refdoc = abap_true )
            is_read_options = VALUE #( fast_for_display = abap_true
                                       include_deleted = abap_true )
            is_locking = VALUE #( lock_result = abap_false )
        IMPORTING
            et_headers = DATA(lt_whr_api_headers) ).
    CATCH /scwm/cx_api_faulty_call.
ENDTRY.

LOOP AT lt_whr_api_headers ASSIGNING FIELD-SYMBOL(<whr_api_header>).
    WRITE: / 'API ',
           <whr_api_header>-docno,
           <whr_api_header>-doccat,
           <whr_api_header>-doctype.
ENDLOOP.
```

Listing 2.9 Beispiel einer API-basierten Lieferungsselektion

Wir legen hier eine Objektinstanz für einen Auslieferungsauftrag an und instanzieren den Service mit der statischen API-Factory-Methode /SCWM/CL_API_FACTORY=>GET_SERVICE. Bevor Sie die technischen GUIDs der EWM-Lieferung der Liefernummer aus

SAP ERP oder SAP S/4HANA zuordnen, müssen Sie die Lagernummer für den Transaktionsmanager mit der Methode `/SCWM/CL_TM=>SET_LGNUM` festlegen. Nachdem Sie die GUIDs zugeordnet haben, können Sie die Lieferungsinformationen ganz einfach über die zuvor erwähnten Übergabeparameter `is_include` und `is_read_options` abfragen. Eine Schleife über die zurückgegebenen Kopfzeilen der Lieferbelege gibt schließlich einige der Daten auf dem Bildschirm aus.

2.1.3 Integration mit anderen EWM-Komponenten

In Abschnitt 2.1.1 haben wir die zentrale Funktion der Lieferabwicklung skizziert und in Abschnitt 2.1.2, »Datenmodell der EWM-Lieferung«, haben wir beschrieben, wie die Service Provider der jeweiligen Anwendungsprogramme die Lieferobjekte verwalten und fortschreiben. Eine der bedeutendsten EWM-Komponenten ist in diesem Zusammenhang die *Lageraufgabenverarbeitung* zur Lageranforderung. Durch diese Komponente werden die speziellen Lageraufgaben zur Einlagerung und Kommissionierung mit Bezug zu einer Lageranforderung angelegt. Im Gegensatz dazu gibt es auch Lageraufgaben, die nicht mit einer Lageranforderung zusammenhängen. Für diese nicht lieferungsbezogenen Lageraufgaben nutzen wir jedoch andere Funktionen der Lageraufgabenverarbeitung.

Das Programmierbeispiel in Listing 2.10 zeigt, wie einfach es ist, den Funktionsbaustein `/SCWM/TO_CREATE_WHR` zu verwenden, um die Lageraufgaben für Lieferungen anzulegen. Auch hier verwenden wir die bereits selektierte Tabelle `lt_srv_items` als Referenz. Da wir im Bereich der `/SCWM/-`Komponenten arbeiten, müssen wir auch die entsprechende Lagernummer übergeben, die im Parameter `p_lgnum` zu finden ist.

"2.8 Sample Call of Function Module `/SCWM/TO_CREATE_WHR`

```
DATA: lt_create_whr TYPE /scwm/tt_to_prep_whr_int,
      lt_ltap_vb    TYPE /scwm/tt_ltap_vb,
      lt_bapiret    TYPE bapirettab,
      lv_severity   TYPE bapi_mtype.

BREAK-POINT ID zewmdevbook_21.
IF p_wt_fm = abap_true.
  "Call Central Cleanup
  /scwm/cl_tm=>cleanup( EXPORTING iv_lgnum = p_lgnum ).
  "Transfer DLV Keys
  LOOP AT lt_srv_items ASSIGNING <srv_item>.
    DATA(ls_create_whr) = VALUE /scwm/s_to_prepare_whr_int(
      rdocid = <srv_item>-docid
      ritmid = <srv_item>-itemid
      rdocat = <srv_item>-doccat ).
    APPEND ls_create_whr TO lt_create_whr.
```

```
CLEAR ls_create_whr.  
ENDLOOP.  
"Trigger WT Creation for Warehouse Request per Function Module  
CALL FUNCTION '/SCWM/TO_CREATE_WHR'  
EXPORTING  
    iv_lgnum      = p_lgnum  
    iv_bname      = sy-uname  
    it_create_whr = lt_create_whr  
    iv_update_task = abap_false  
    iv_commit_work = abap_true  
IMPORTING  
    et_ltap_vb    = lt_ltap_vb  
    et_bapiret    = lt_bapiret  
    ev_severity   = lv_severity.
```

Listing 2.10 Beispielaufruf von /SCWM/TO_CREATE_WHR

Der Rückgabeparameter `et_ltap_vb` liefert die angelegten Lageraufgaben zurück. Die Tabelle `et_bapiret` und die Variable `ev_severity` enthalten das Erstellungsprotokoll sowie die aggregierte Nachrichtenart (E, I etc.). Wenn Sie in einem Projekt eine eigene UI-Transaktion für diese Anwendung implementieren wollen, z. B., um den Endanwender*innen die angelegten Lageraufgaben vor dem Sichern anzuzeigen, verwenden Sie den Funktionsbaustein `/SCWM/TO_PREP_WHR_UI_INT` anstelle von `/SCWM/TO_CREATE_WHR`. Die Programmlogik ist grundsätzlich identisch. Der Unterschied besteht darin, dass die Lageraufgaben bei dem Funktionsbaustein `/SCWM/TO_PREP_WHR_UI_INT` zunächst nur während der Laufzeit (transient) angelegt und dann je nach Aktion des Benutzers oder der Benutzerin entweder publiziert und gesichert oder per Rollback gelöscht werden. Wenn Sie den Funktionsbaustein `/SCWM/TO_PREP_WHR_UI_INT` verwenden, sind Sie vollständig für die gesamte Verbuchungssteuerung (Commit) verantwortlich. Um sich mit der dahinterstehenden Logik vertraut zu machen, schauen Sie sich den Funktionsbaustein `/SCWM/TO_CREATE_WHR` an. Hier kümmert sich der Funktionsbaustein `/SCWM/TO_PREP_WHR_INT` sowohl um die transiente Lageraufgabenerstellung als auch, je nach Rückmeldung, um die anschließende Verbuchung.



Übergabeparameter des Funktionsbausteins /SCWM/TO_CREATE_WHR

Der Funktionsbaustein `/SCWM/TO_CREATE_WHR` hat verschiedene Parameter, die alle im Funktionsbaustein selbst gut dokumentiert sind. Im Codingbeispiel in Listing 2.10 haben wir nur die wichtigsten Parameter verwendet.

Ähnlich wie zuvor für die Arbeit mit Lieferungen gezeigt, wurden in EWM in SAP S/4HANA auch APIs für Lageraufgaben und die Lagerauftragsbearbeitung zur Verfügung gestellt. In Kapitel 5, »Funktionsbausteine, Methoden und APIs für EWM«, finden Sie weitere Informationen zu den EWM-APIs. Listing 2.11 zeigt ein Beispiel dafür, wie die Methode `create_for_whr` der API `/scwm/if_api_whse_task` verwendet werden kann, um die Anlage von Lageraufgaben für Lieferpositionen anzustoßen.

```
"2.9 Sample Call of /SCWM/IF_API_WHSE_TASK
DATA: lo_wt_api TYPE REF TO /scwm/if_api_whse_task.
ELSEIF p_wt_api = abap_true.
  TRY.
    /scwm/cl_api_factory=>get_service( IMPORTING eo_api = lo_wt_api).
    "Trigger WT Creation for Warehouse Request per API
    lo_wt_api->create_for_whr(
      EXPORTING
        iv_whno      = p_lgnum
        it_create    = CORRESPONDING #( lt_srv_items )
      IMPORTING
        et_created_wht = DATA(lt_created_wht)
        eo_message    = DATA(lo_error_msg) ).

    lo_error_msg->get_messages(
      IMPORTING et_message = DATA(lt_error_msg)
               et_bapiret = lt_bapiret ).
    CATCH /scwm/cx_api_whse_task.
  ENDTRY.
ENDIF.
WRITE: / 'Number of messages from WT creation:', lines( lt_bapiret ).
```

Listing 2.11 Beispielaufruf der API `/SCWM/IF_API_WHSE_TASK`

Verfügbarkeit von APIs für EWM

Bevor Sie die APIs für EWM verwenden, sollten Sie im SAP-Hinweis 3115182 nachsehen, welche APIs ab welchem SAP-S/4HANA-Release für EWM verfügbar sind.

Die folgenden zusätzlichen EWM-Komponenten interagieren ebenfalls mit Lieferobjekten:

- EWM-Stammdaten
- Verpacken



- Wellenmanagement
- Arbeitsverwaltung
- Bausatzabwicklung
- Cross-Docking
- logistische Zusatzleistungen (LZL)
- Warenannahme und Versand
- erweiterte Warenannahme und Versand
- Transit-Lagerung
- Produktionsintegration
- Just-in-time-Abwicklung (JIT)

In den folgenden Kapiteln dieses Buches werden wir Ihnen einige Implementierungsbeispiele für die Lieferabwicklung vorstellen. In Abschnitt 3.6.6, »Entwicklungsprojekt: Verpacken von Anlieferungen«, wird z. B. beschrieben, wie die Lieferabwicklung mit speziellen Verpackungsfunktionen zusammenarbeitet.



Lieferreferenzen in der Bestandsverwaltung

Ein großer Vorteil von EWM ist die Bestandstrennung basierend auf dem Lieferbezug. So können Sie z. B. bereits kommissionierte Bestände problemlos innerhalb des Lagers bewegen, ohne dass der Bezug zur Lieferung verloren geht. Der Lieferbezug des Bestandes bleibt durchgängig eindeutig identifizierbar.

Darüber hinaus umfasst die Lieferabwicklung Schnittstellen für die Integration folgender Komponenten:

- Quality Inspection Engine in Decentralized EWM (QIE)
- Qualitätsmanagement in Embedded EWM (QM)
- Post Processing Framework (PPF)
- Routenfindung (Routing-Guide)
- SAP Global Trade Services (SAP GTS)
- SAP Transportation Management (SAP TM)

Zusätzlich zu den verschiedenen Schnittstellen und Aufrufmöglichkeiten für Lieferobjekte bietet EWM auch eine Reihe von BADIs, die es Ihnen ermöglichen, Ihre eigenen Ermittlungen und Validierungen für Lieferungen durchzuführen sowie kundeneigene Felder zu verwalten. SAP liefert Beispielimplementierungen für verschiedene BADIs der Lieferabwicklung aus, um Ihnen eine Programmiervorlage zur Orientierung an die Hand zu geben.



Weitere Quellen für technische Informationen zur Lieferabwicklung

Für weitere technische Informationen empfehlen wir Ihnen How-to-Guides zur Lieferabwicklung wie »How to Access ›Delivery-Warehouse Request‹ Objects in EWM in SAP S/4HANA« und »How to Enhance an EWM Delivery with Own Coding in SAP S/4HANA«. Diese finden Sie im Support Content des SAP Help Portal unter der URL https://help.sap.com/docs/SUPPORT_CONTENT/sewm/ im Ordner **EWM General Information**.

2.2 Lagerlogistik

Die Lagerlogistikkomponente von EWM umfasst alle Warenbewegungen, sowohl innerhalb des Lagers als auch um das Lager herum. EWM unterstützt zahlreiche Prozesse wie die Warenannahme im Lager, den Wareneingang und den Einlagerungsprozess. Es unterstützt auch interne Lagerprozesse, wie z. B. Inventur, Qualitätsprüfung, Nachschub und Umorganisation.

Als wichtigste Prozesse werden oft die Kommissionierung und der Versand angesehen. Als Basis für fast alle logistischen Folgeprozesse dient der Lagerlogistik dabei der *Lieferbedarf* anhand von Lieferungen oder Lageranforderungen. Der für Auslieferungsvorgänge zu verarbeitende Beleg, *Auslieferungsauftrag* genannt, bildet die Grundlage für Planung und Reporting des Warenausgangs im Lager. Zu diesem Beleg können Sie den aktuellen Bearbeitungsstatus und den Belegfluss für alle abhängigen Belege einsehen.

Im Modul *Warenannahme und Versand* werden komplette Lieferungen, deren Positionen oder einzelne Handling Units einer *Transporteinheit* zugeordnet. Sie können Transporteinheiten wiederum *Fahrzeugen* zuweisen, wobei einem Fahrzeug mehrere Transporteinheiten zugeordnet sein können. Der zentrale Beleg für die Ausführung und Dokumentation von Aktivitäten im Lager ist in EWM die *Lageraufgabe*. Sie dient der Bewegung von Beständen und/oder Handling Units gemäß den in den Lieferpositionen angegebenen Daten. Es ist aber auch möglich, Lageraufgaben ohne Bezug zu einer Lieferung anzulegen und zu bearbeiten, z. B. für lagerinterne Prozesse wie Nachschub oder Umlagerung.

Zusätzlich zur Lageraufgabe können weitere Prozesse durchgeführt werden, wie z. B. Qualitätsprüfungen bei der Anlieferung (siehe Abschnitt 2.4). Logistische Zusatzleistungen (LZL) können in den An- und Auslieferungsprozess integriert werden. Im Auslieferungsprozess werden die Lieferpositionen in *Wellen* gruppiert. Lageraufgaben werden für eine komplette Welle angelegt und nach konfigurierbaren Kriterien, d. h. den Regeln für die Erstellung von Lageraufträgen, zu *Lageraufträgen* gebündelt.

2.2.1 Warenannahme und Versand

Ein eingehendes Transportmittel, z. B. ein Lkw, enthält oft mehr als eine Lieferung. Um diese Tatsache im EWM-System abzubilden, können Sie ein separates Objekt – die Transporteinheit – anlegen und Lieferungen, Lieferpositionen oder Handling Units der Transporteinheit-Aktivität (TE-Aktivität) zuordnen.

Wenn Sie eine Transporteinheit anlegen, wird neben der Transporteinheit im Hintergrund immer auch eine Aktivität angelegt. Diese Transporteinheit-Aktivität selbst ist nicht bearbeitbar. Es gibt sie auch für andere Objekte in Warenannahme und Versand, vor allem für Fahrzeuge und Tore. Transporteinheit-Aktivitäten sind immer für einen bestimmten Zeitraum gültig, der in der Aktivität definiert ist. Die Transporteinheit-Aktivität wird durch die Nummer der externen Transporteinheit und den Frachtführer definiert. Es kann mehrere *geplante* Aktivitäten, aber nur eine *aktive* Aktivität geben; eine Aktivität wird durch die Ankunft der Transporteinheit am Kontrollpunkt aktiviert. Einem Fahrzeug können die Aktivitäten einer oder mehrerer Transporteinheiten zugeordnet werden.

Die Verwendung des Objekts Fahrzeug bringt nur dann einen Mehrwert, wenn mehr als eine Transporteinheit verwendet wird, z. B. ein Lkw mit Anhänger. Wenn eine Lieferung auf mehrere Transporteinheiten verteilt wird, kann dies im System durch direkte Zuordnungen der Lieferpositionen oder Handling Units zu diesen Transporteinheiten abgebildet werden. Dies geschieht vor allem im Warenausgangsprozess beim Beladen, wenn die endgültigen Auslieferungen noch nicht bekannt sind und noch mit den Lageranforderungen gearbeitet wird.



Yard Management in EWM

Als *Yard* oder auch *Hof* bezeichnen wir den Bereich außerhalb des Lagers, in dem die ankommenden und abgehenden Transporteinheiten abgefertigt werden. Mit dem Yard Management in EWM können Sie nicht nur die am Tor ankommenden Transporteinheiten registrieren, sondern ihnen auch Lokationen zuweisen und sie innerhalb des Yards bewegen. Sie können Parkplätze belegen und Lageraufgaben anlegen, um die Transporteinheit zu ihrem Ziel zu bewegen, in der Regel zu einem Tor. Dies bietet die Möglichkeit der Parkplatzverwaltung und der Verwaltung zusätzlicher Transporteinheiten, z. B. von Wechselbrücken oder Anhängern, die mit eigenen Ressourcen bewegt werden müssen.

Die Tore des Lagers, an denen Transporteinheiten beladen und entladen werden, müssen im Lager einem Lagerplatz zugeordnet werden. Diese sogenannten *Torplätze* sind notwendig, um Lageraufgaben für die Tore zu anzulegen. Dabei geht es aber nicht um physisch vorhandene Plätze, auf denen die Waren tatsächlich gelagert wer-

den. Arbeiten Sie mit dem Yard Management, müssen die Tore auch einem Lagerplatz außerhalb des Lagers zugeordnet werden – dem Platz, an dem die Transporteinheit am Tor steht. Das Yard wird nur über die Verknüpfung der Torplätze innerhalb und außerhalb des Lagers mit einem Lager verbunden.

Die Bewegung einer Transporteinheit mit einer Lageraufgabe an einen Torplatz erzeugt oder aktiviert automatisch eine *Toraktivität* für die Transporteinheit. An einem Tor kann nur eine Transporteinheit angedockt werden. Aus diesem Grund empfehlen wir die Verwendung der Yard-Management-Komponente nur für zeitnahe Buchungen. Ohne Yard Management müssen Sie Toraktivitäten manuell in der Pflgetransaktion der Transporteinheit anlegen und aktivieren. Warenannahme und Versand arbeiten auf der Grundlage der Aktivierung und Deaktivierung von Transporteinheitsstatus mit Zeitstempeln (aktuelles Datum und Uhrzeit).

Die Verwendung von Transporteinheiten bietet auch die Möglichkeit, Bestände auf der Transporteinheit-Lokation zu verwalten. So können Sie Wareneingänge schneller buchen, ohne Lagerraum zu beanspruchen. Bereitstellungszonen für ausgehende Prozesse werden nicht nur durch den Warenausgang entlastet, sondern auch durch das Beladen der Transporteinheit, sodass Sie Bereitstellungszonen besser verwalten können. Natürlich ist die Beladung auch ohne Transporteinheiten möglich; in diesem Fall wird der Bestand auf den logischen Torplatz verschoben. Die Bestandstransparenz ist damit nicht mehr gewährleistet, da der Torplatz physisch nicht vorhanden ist und der Bestand nicht kontrolliert werden kann. Es ist also nicht klar, wo sich die Ware tatsächlich befindet.

Die Transporteinheit-Aktivität enthält Informationen darüber, wann die Transporteinheit im Lager erwartet wird und wann sie das Lager wieder verlassen wird. Die Aktivität kann zusätzliche Informationen über den Fahrer oder die Fahrerin und das Transportmittel (z. B. das Fahrzeugkennzeichen), einschließlich anderer Identifikationen und Siegelinformationen, enthalten. Vor allem aber enthält sie Statusinformationen.

Die Transporteinheit-Aktivität wird durch die Benutzeraktion **Ankunft am Kontrollpunkt** aktiviert, die Sie im Menü finden. Wenn das Yard Management aktiviert ist, wird die Handling Unit der Transporteinheit auf den Platz am Kontrollpunkt bewegt. Von diesem Zeitpunkt an kann die Transporteinheit mit Lageraufgaben innerhalb des Yards zu einem Tor bewegt werden, wo sie beladen oder entladen wird. Wenn kein freies Tor vorhanden ist, kann die Transporteinheit vorübergehend auf einen Parkplatz bewegt werden.

Wir werden nun einen genaueren Blick auf die Wareneingangs- und Warenausgangsvorgänge werfen und anschließend das der Warenannahme und dem Versand zugrunde liegende Datenmodell betrachten.

Warenannahme

In der Warenannahme entscheidet das Customizing, ob der Wareneingang auf die Lokation der Transporteinheit bzw. den Torplatz oder auf die Bereitstellungszone gebucht wird, und damit, ob die Ware abgeladen werden muss oder nicht. Das Customizing für den Wareneingang finden Sie im EWM-Einführungsleitfaden unter **Prozessübergreifende Einstellungen • Warenannahme und Versand • Allgemeine Einstellungen • Steuerung von Warenbewegungen einrichten**. Wenn ein Wareneingang für eine Transporteinheit oder auf einen Torplatz gebucht wird, ist eine Entladung mithilfe einer Lageraufgabe erforderlich. Bei einem Wareneingang in die Bereitstellungszone ist kein Entladen erforderlich, aber dieser Prozess wird dennoch mit einer einfachen Entladung unterstützt. Nachdem die Transporteinheit entladen wurde, kann sie erst das Tor und dann den Kontrollpunkt verlassen oder sie kann mit einer Aktivität für ausgehende Transporteinheiten erneut zum Laden verwendet werden.

Status

Die Objekte Transporteinheit und Fahrzeug sehen eine Reihe von Status vor, um den aktuellen Zustand einer Aktivität zu dokumentieren, z. B. **Ankunft am Kontrollpunkt, Ankunft am Tor, Entladen gestartet** und **Entladen beendet**.

Status, die es sowohl für die Lieferung als auch für die Transporteinheit gibt, werden zwischen diesen beiden Objekten synchronisiert. Die Transporteinheit informiert die Lieferung bei der Ankunft am Kontrollpunkt über ihre eigene Statusänderung, wodurch der Status für die Lieferung auf **Im Yard** gesetzt wird. Wenn bereits ein Tor für die Transporteinheit vorgesehen ist, wird auch die Torinformation an die Lieferposition weitergegeben, falls noch keine offene Lageraufgabe für die Lieferposition existiert. Die *Bereitstellungszonenfindung* wird aufgerufen, um eine optimierte Bereitstellungszone für das Tor auf Lieferpositionsebene zu finden. Bei Änderung der Transporteinheit findet direkt eine Synchronisierung zwischen der Transporteinheit und der Lieferung statt. Aus diesem Grund werden die Lieferungen gesperrt und müssen änderbar sein. Ansonsten ist die Änderung der Transporteinheit nicht möglich.

Statusänderungen an der Lieferung können, ebenso wie Änderungen des Lade- oder Warenbewegungsstatus, zu einer Neuberechnung des entsprechenden Status in der Transporteinheit-Aktivität führen. Mengenänderungen und Umpacken führen gegebenenfalls zu einer Kapazitätsänderung der Transporteinheit-Aktivität. Änderungen in der Lieferung, wie z. B. Status- oder Mengenänderungen, die sich auf die Transporteinheit auswirken, werden nicht direkt in der Transporteinheit fortgeschrieben, sondern führen zur Einplanung einer PPF-Aktion, die die Transporteinheit synchronisiert (siehe Abbildung 2.5).

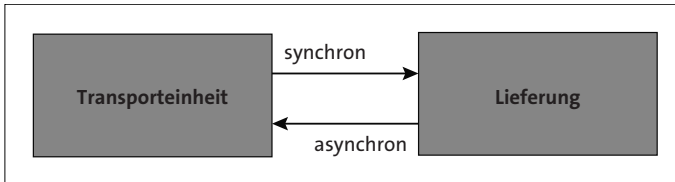


Abbildung 2.5 Synchronisierung zwischen Lieferung und Transporteinheit

Transporteinheiten enthalten nur Referenzen

Transporteinheiten enthalten keine eigenen Mengen-, Produkt- oder Handling-Unit-Informationen, sondern nur Verweise auf die Lieferung, die jeweilige Lieferposition und eventuell auf Handling Units. Aus diesem Grund zeigt die Pflgetransaktion der Transporteinheit immer den aktuellen Stand der Daten der Handling Units in der Anlieferung an. Diese Daten stimmen nach dem Entladen nicht unbedingt mit den Daten der empfangenen Handling Units überein.

Versand

Für den Versand kann eine Transporteinheit an einem Kontrollpunkt ankommen, die nur Waren abholt, oder es kann eine Transporteinheit wiederverwendet werden, die zuvor eingehende Waren geliefert hat und noch auf dem Yard oder am Tor steht und für die eine aktive eingehende Aktivität vorhanden ist. Für eine solche Transporteinheit kann eine ausgehende Aktivität aktiviert werden, wobei das System die eingehende Aktivität automatisch abschließt. Wenn sich die eingehende Transporteinheit an einem Tor befindet, wird auch für die Aktivität der ausgehenden Transporteinheit eine Toraktivität angelegt und aktiviert. Ist das Yard Management aktiv, ändert sich dabei die Lokation der Transporteinheit nicht.

Die transportierten Objekte, vor allem Handling Units, müssen nicht unbedingt für den Transport eingeplant werden, sondern können auch spontan durch das Beladen einer Transporteinheit zugeordnet werden. Im Warenausgang kann deshalb nicht automatisch vom System ermittelt werden, wann die zugeordneten Lieferungen vollständig beladen sind. Daher muss der Status **Beladen beendet** immer manuell gesetzt oder zumindest bestätigt werden; dieser Status ist Voraussetzung dafür, dass die Transporteinheit den Kontrollpunkt wieder verlassen kann.

Für die ungeplante Beladung von Transporteinheiten sind Lageraufgaben notwendig, die die Zuordnung von Handling Units zu Transporteinheiten ermöglichen. Ohne manuelle Nach-Datenvorgabe legt das System eine Lade-Lageraufgabe immer auf dem Torplatz des aus der Lieferposition ermittelten Tores an. Nutzen Sie zum

Quittieren einer solchen Lageraufgabe eine RF-Transaktion (*Radio Frequency*), wird die Lageraufgabe automatisch auf die an diesem Tor aktive Transporteinheit quittiert und damit eine Zuordnung zwischen Handling Unit und Transporteinheit vorgenommen.

Datenmodell

Die Komponente Warenannahme und Versand ist Teil des Namensraumes /SCWM/. Ihre Zuständigkeit ist in der technischen Komponente SCM-EWM-SR und ihren Unterkomponenten zusammengefasst. Neben den Objekten Transporteinheit, Fahrzeug und Tor beinhaltet die Komponente auch die Funktionalität der *Bereitstellungs-* und *Torfindung* sowie die Be- und Entladungsfunktionen. Diese Funktionen sind auch für die Lieferabwicklung relevant, unabhängig von der Verwendung von Transporteinheiten.

Das Paket /SCWM/SHP_RCV ist in fünf Unterpakete unterteilt. Darüber hinaus gibt es das Paket /SCWM/YARD_MGMT für die Funktionalität des Yard Managements. Tabelle 2.4 listet die Pakete auf, die für die Komponente Warenannahme und Versand relevant sind.

| Paket | Funktion |
|--------------------------|--|
| /SCWM/SHP_RCV_CORE | Bearbeitung der Objekte Transporteinheit, Fahrzeug, Tor und Bereitstellungszone |
| /SCWM/SHP_RCV_CUST | Customizing der Tabellen, Pflege-Views, Funktionen zum Lesen von Customizing und F4 -Wertehilfen |
| /SCWM/SHP_RCV_PPF | Implementierungen des PPF |
| /SCWM/SHP_RCV_UI | Transaktionen zur Bearbeitung von Transporteinheit-, Fahrzeug- und Toraktivitäten |
| /SCWM/ERP_TM_INTEGRATION | Customizing und IDocs zur Integration des Transportmanagements in SAP ERP oder SAP S/4HANA (LE-TRA) |
| /SCWM/YARD_MGMT | Transaktionen zur Definition von Checkpoints, Bearbeitung von Check-in und Check-out sowie Bewegungen auf dem Yard |

Tabelle 2.4 Pakete für Warenannahme und Versand

Die Verarbeitung von Belegdaten erfolgt über die in Abbildung 2.6 dargestellten Klassen. Jede Aktivität wird von einer eigenen Instanz der jeweiligen Klasse verwaltet. Die Daten sind in den Klassen zu finden, die mit /SCWM/CL_SR_DO... beginnen. Die Geschäftslogik wird über die zugehörigen Klassen ausgeführt, die mit /SCWM/CL_SR_BO... beginnen und einen Verweis auf die aktuelle Instanz der Daten enthalten. Der

Objektmanager `/SCWM/CL_SR_BOM` verwaltet die Instanzen von Business-Objekten, deren Namen mit `/SCWM/CL_SR_BO...` beginnen.

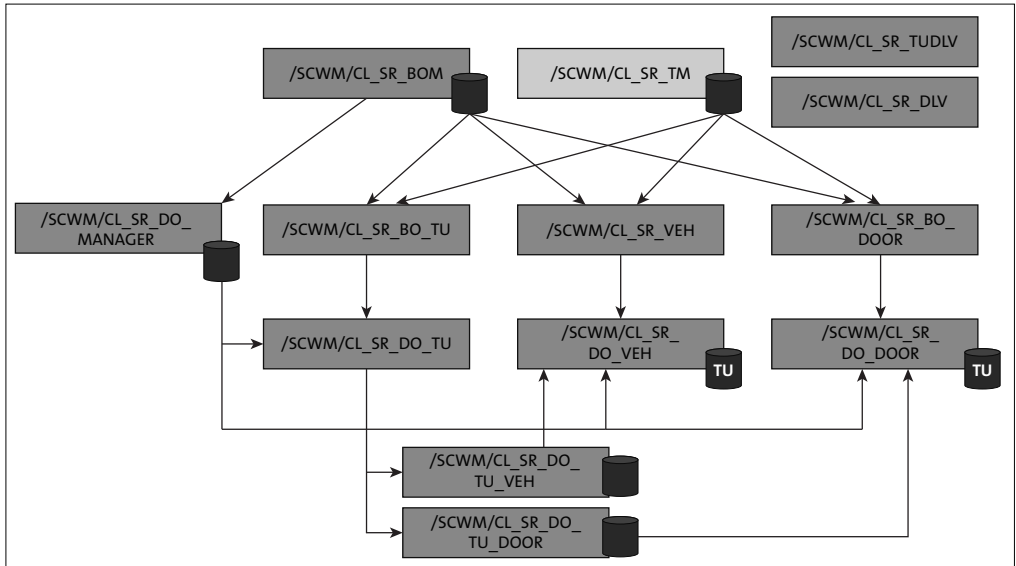


Abbildung 2.6 Klassenmodell in Warenannahme und Versand

Die Verknüpfungen zwischen einer Transporteinheit und einem Fahrzeug oder Tor werden in eigenen Klassen verwaltet, die in der Klasse `/SCWM/CL_SR_DO_TU` referenziert werden. Aus Gründen der Performance werden auch für die entsprechenden Tor- und Fahrzeuginstanzentabellen mit Schlüsseln der zugehörigen Transporteinheiten vorgehalten.

Die Verknüpfung zwischen Transporteinheiten und Lieferung wird in der Klasse `/SCWM/CL_SR_TUDLV` verwaltet. Diese Klasse ist unabhängig von den anderen Klassen und kann auch ohne diese verwendet werden, wenn keine weiteren Informationen über die Transporteinheit oder das Fahrzeug benötigt werden. Die Klasse `/SCWM/CL_SR_DLV` dient als Schnittstelle der Transporteinheit zur Lieferung. Sie informiert die Lieferung über Zuordnungs- und Statusänderungen.

Die Klasse `/SCWM/CL_SR_DO_MANAGER` existiert nur noch aus historischen Gründen und darf nicht direkt verwendet werden. Lediglich die Methoden zum Lesen/Einrichten aller relevanten Objekte bleiben erhalten und dürfen nur vom Objektmanager der Klasse `/SCWM/CL_SR_BOM` verwendet werden.

Die drei Objekte in Warenannahme und Versand, also Transporteinheit, Tor und Fahrzeug, bestehen jeweils aus dem Objektkopf und einer beliebigen Anzahl von Aktivitäten zu diesem Objekt. Für einen Eintrag in der Tabelle `/SCWM/TUNIT` oder `/SCWM/VEHICLE` muss mindestens ein Eintrag in der Aktivitätstabelle `/SCWM/TU_SR_ACT` bzw.

/SCWM/VEH_SR_ACT vorhanden sein. Alle weiteren Tabellen enthalten die für die Aktivität relevanten Daten. Die Datenbanktabellen der Objekte sind in Tabelle 2.5 aufgelistet. Die Transporteinheit und das Fahrzeug werden gelöscht, wenn die letzte Aktivität gelöscht wird.

| Objekt | Tabelle | Aktivität | Zusätzliche Tabelle |
|------------------|---------------|------------------|--|
| Transporteinheit | /SCWM/TUNIT | /SCWM/TU_SR_ACT | /SCWM/TU_STATUS /SCWM/TU_IDENT /SCWM/TUNIT_SEAL /SCWM/TU_VEH /SCWM/TU_DOOR /SCWM/TU_DLV |
| Fahrzeug | /SCWM/VEHICLE | /SCWM/VEH_SR_ACT | /SCWM/VEH_STATUS /SCWM/VEH_IDENT |
| Tor | /SCWM/TDOOR | /SCWM/DOOR_SRACT | – |

Tabelle 2.5 Tabellen von Objekten in Warenannahme und Versand

Das Objekt *Tor* hat einen besonderen Status, da es im Customizing definiert wird. Beim Anlegen einer Aktivität muss ein Eintrag für das Tor vorhanden sein, und Toraktivitäten können nicht als eigenständige Entitäten angelegt werden. Sie können allerdings auch nur mit einer Referenz auf eine Transporteinheit-Aktivität existieren. Das bedeutet, dass für jeden Eintrag in der Tabelle /SCWM/DOOR_SRACT ein Eintrag in der Tabelle /SCWM/TU_DOOR vorhanden sein muss.

Für so gut wie jede Aktion, die für eine Aktivität durchgeführt werden kann, werden die BADIs /SCWM/EX_SR_ACTION_TU für Transporteinheiten, /SCWM/EX_SR_ACTION_VEH für Fahrzeuge und /SCWM/EX_SR_ACTION_DOOR für Tore aufgerufen. Diese BADIs können für benutzerdefinierte Prüfungen verwendet werden.

Da die drei Objekte voneinander abhängig sind und manche Änderungen sich auf mehr als ein Objekt auswirken, muss die Konsistenz der Objekte bei der Datenänderung gewährleistet sein. Aus diesem Grund wird, wenn eine beliebige Methode einer /SCWM/CL_SR_BO...-Klasse ausgeführt wird, vor der ersten Datenänderung eine Kopie der Datenobjektinstanz /SCWM/CL_SR_DO... gespeichert. Darüber hinaus wird dabei die Klasse /SCWM/CL_SR_TM informiert, dass eine solche Kopie erzeugt wurde.

Wenn eine Aktion erfolgreich ist, ruft die Anwendung, die die Aktion aufgerufen hat, z. B. bei Ankunft am Kontrollpunkt, die Klasse /SCWM/CL_SR_TM auf, um alle gespeicherten Objektinstanzen zu initialisieren und darüber zu informieren, dass sie die Kopie wieder verwerfen sollen.

Wenn innerhalb einer Aktion in einem der Objekte ein Fehler auftritt, wird die Klasse /SCWM/CL_SR_TM ebenfalls aufgerufen. Sie informiert alle gespeicherten Business-Ob-

jekt-Instanzen, die aktuelle Datenobjektreferenz zu verwerfen und durch die Kopie zu ersetzen. Außerdem wird sichergestellt, dass im Fall eines Fehlers die Daten auf allen Objekten dieselben sind wie vor dem Start der Aktion.

Integration von SAP Transportation Management ohne Warenannahme und Versand in EWM in SAP S/4HANA

Bevor wir diesen Abschnitt abschließen, möchten wir den *Erweiterten Warenannahme- und Versandprozess* in EWM als eine neue Integrationsmethode von EWM und SAP Transportation Management (SAP TM) erwähnen, bei der die Transporteinheit oder das Fahrzeug als redundante Objekte in EWM obsolet werden. Der Frachtauftrag aus SAP TM fungiert in diesem Szenario als einziges Objekt für die Transportausführung. Diese neue Funktion ist umfangreich und wird noch erweitert. Wir empfehlen Ihnen, dazu die SAP-Hilfe sowohl für EWM als auch für SAP TM zu konsultieren. Als weitere Informationsquelle kann das E-Bite »Integrated TM and EWM in SAP S/4HANA« von Mrinal K. Roy dienen (SAP PRESS 2021, siehe www.sap-press.de/5315).



2.2.2 Lageraufgabe und Lagerauftrag

Warenbewegungen innerhalb des Lagers erfolgen über das Objekt *Lageraufgabe*. Für Bestandsbuchungen, wie Wareneingänge, Warenausgänge und Umbuchungen, erzeugt das System zu Dokumentationszwecken abgeschlossene Lageraufgaben ohne Lagerauftrag. Ansonsten werden die Lageraufgaben in der Regel einem *Lagerauftrag* zur Ausführung im Lager zugeordnet.

Umlagerungen innerhalb des Lagers bestehen aus geplanten und abgeschlossenen Lageraufgaben. Beim Anlegen einer geplanten Lageraufgabe wird automatisch ein Lagerauftrag gemäß den im Customizing definierten *Lagerauftragserstellungsregeln* erzeugt. Der Lagerauftrag kann einer Ressource zugeordnet werden und dient der Aktualisierung und Quittierung zugeordneter Lageraufgaben.

Der Zweck einer Lageraufgabe kann durch ihre *Lagerprozessart* identifiziert werden (TRART). Die verfügbaren Arten sind in Tabelle 2.6 aufgelistet. Die für die Lageraufgabe verfügbaren Daten hängen von dieser Art ab (siehe Tabelle 2.9 für weitere Informationen).

| Lagerprozessart | Bezeichnung |
|-----------------|------------------|
| 1 | Einlagerung |
| 2 | Auslagerung |
| 3 | interne Bewegung |

Tabelle 2.6 Lagerprozessarten in EWM