

## 1 Metadaten-Architektur: Functions → PDO-Layer → CLL-Layer

### CLL-Layer (Channel Logic Layer): Kapselt

Business Data für UI-Layer:

- **BO:** Business Object
- **DO:** Dependent Object
- **MO:** Mapper Object

**PDO-Layer (Procurement Document Layer):** stellt Wrapper-FuBas für die BBP-Functions (mit PDO-spezifischen (HM)-Strukturen) zur Verfügung (CTR, SC,...)

#### BBP-Functions:

Check, Create, GetDetail, GetList, Item\_GetDetail, Lock, Save, Unlock

SC-BO: „/SAPSRM/CL\_PDO\_BO\_SC“  
SC-DO-Kontierung: „/SAPSRM/CL\_PDO\_DO\_ACCT\_ASSGMNT“  
SC-BOM: „/SAPSRM/CL\_CH\_WD\_BOM\_SC“

FuBa „BBP\_PD\_SC\_GETDETAIL“ (führt BBP-FuBa aus und mappt Daten in PDO-spezifische Strukturen)

FuBa „BBP\_PROCDOC\_GETDETAIL“

## 2 Anwendungsfall: Implementierung eines Reiters mit Datensynchronisation

**Anzeigen Bieter : 10001806**

Geschäftspartnerartyp Bieter    Belegdatum 15.11.2013    Geschäftspartnernummer 10001806    Name des Geschäftspartners SMA

Bearbeiten    Schließen    Kreditur beantragen

Unternehmensdaten    Ansprechpartnerdaten    Geschäftsbelege    EDI Regeln    **Kreditoren-WF**    Änderungsbelege

1 Allg. Daten erfassen    2 Workflow parametrisieren    3 Einkaufsdaten erfassen    4 Workflow starten

Zurück    Weiter    Sicherung ohne zu prüfen

Geben Sie bitte den Workflowtyp an

Warenwirtschaftslieferant - Workflow an GP zur Datenpflege  
 Warenwirtschaftslieferant - Workflow nicht an GP, ausschließlich interne Datenpflege  
 Vorlieferant

## 3 Datensynchronisation: HowTo

### 1. Task-Container aufbauen:

Attribute: **MO\_TASK\_CONTAINER** type ref to /SAPSRM/IF\_CLL\_TASK\_CONTAINER,

**MO\_AO\_MAPPER** type ref to /SAPSRM/IF\_CLL\_AO\_MAPPER

```
DATA: lo_task_factory TYPE REF TO /sapsrm/if_cll_taskcon_factory.
DATA: lo_task_container TYPE REF TO /sapsrm/if_cll_task_container.
lo_task_factory = /sapsrm/cl_ch_wd_taskcont_fact=>get_instance( ).
lo_task_container = lo_task_factory->get_task_container( ).
```

- Aggregating object mapper für Bieter  
wd\_this->mo\_ao\_mapper    ?= lo\_task\_container->get\_aom\_bidder( ).  
wd\_this->mo\_task\_container = lo\_task\_container.

### 2. Kontext-Mapper-Objekt erzeugen:

Attribute: **MO\_META\_INIT** type ref to /SAPSRM/CL\_CH\_WD\_MAP\_META\_INIT.

```
CREATE OBJECT wd_this->mo_meta_init.
```

### 3. Mapping-Objekt für View-Context aufbauen:

```
wd_this->mo_meta_init->init_context( EXPORTING iv_root_context_node_info = lo_root_context_node_info ).
```

### 4. Kontext-Knoten für Datenmapper-Objekt aufbauen (Kontext-Namensgleichheit beachten):

```
lo_mo_map_factory = /sapsrm/cl_ch_wd_mo_mapfactory=>get_mo_mapfactory_instance( ).
lon_comp_data    = wd_this->mo_map_factory->get_child_node( name = 'COMPANY_DATA' ).
wd_this->mo_modm_comp_data = lo_mo_map_factory->create_modm_comp(
    io_task_container                    = wd_comp_controller->mo_task_container
    io_wd_view_controller                = lo_wd_view_controller
    io_wd_component_controller          = wd_comp_controller
    io_parent_xo_mapper                 = wd_comp_controller->mo_ao_mapper
    io_meta_init                         = wd_this->mo_meta_init
    ion_wd_comp_data                     = lon_comp_data ).
```

### 5. Daten übernehmen/synchronisieren:

```
wd_this->mo_modm_comp_data->/sapsrm/if_cll_mapper-refresh( ).
```