

Stop!

Don't throw away that ADM2 code just yet...

Jeff Ledbetter

Product Architect, Roundtable Software



Who do we think we are?

- Roundtable TSMS
 - Software Configuration Management for OpenEdge
- Roundtable Team
 - Basic version control built for Eclipse



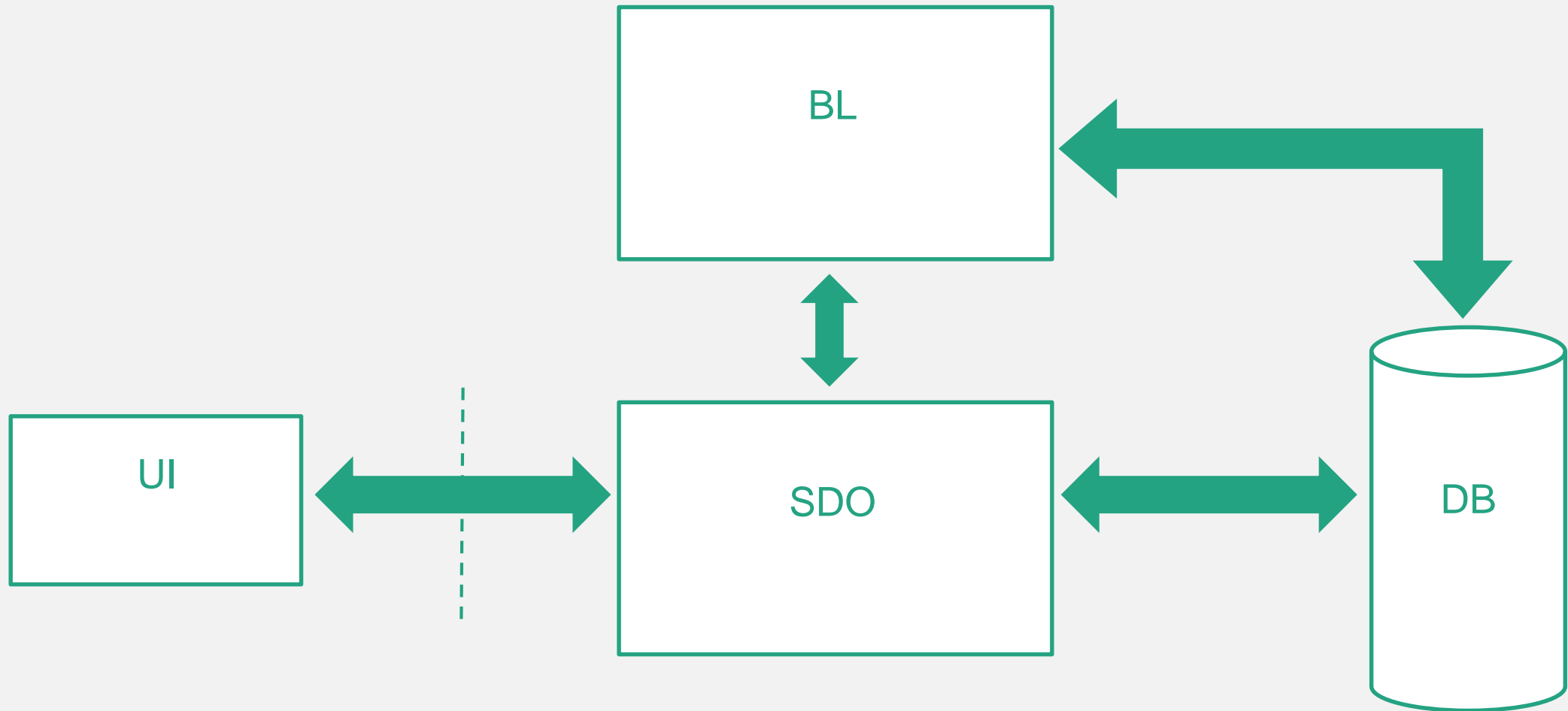
Demo.



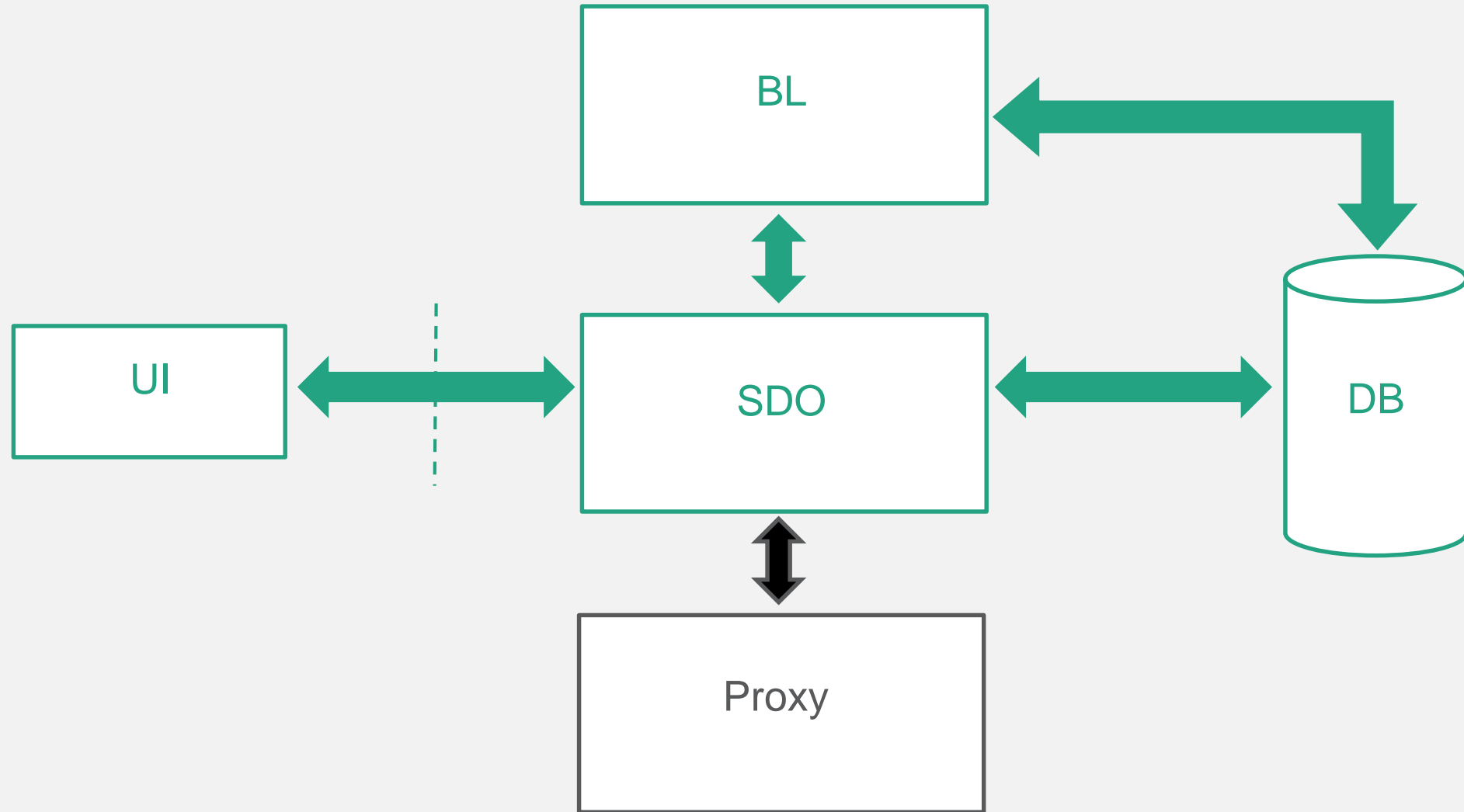
Future-proof.



ADM2 Basic



ADM2 with Proxy Layer



Build It

- Define your model
- Build integration points
- Build proxy layer
- Expand



Build It

- Define your model
 - RowObject temp-table

```
DEFINE TEMP-TABLE ttCustomer  
  {pug/sdoCustomer.i}  
  {src/adm2/robjflds.i}.
```



Build It

- Identify integration points
 - serverSendRows
 - serverCommit



Build it

- Identify integration points
 - serverSendRows

```
RUN serverSendRows IN phSdo
  (INPUT ?,
   INPUT pcRowIdent,
   INPUT (IF pcRowIdent <> "FIRST":U AND pcRowIdent <> "LAST":U THEN
           TRUE
           ELSE
           FALSE) ,
   INPUT piRowsToReturn,
   OUTPUT iRowsRet,
   OUTPUT TABLE-HANDLE phTT) .
```

Build it

- Identify integration points
 - serverCommit

```
RUN serverCommit IN hSdo  
  (INPUT-OUTPUT TABLE-HANDLE hRowObjUpd,  
   OUTPUT cMessages,  
   OUTPUT cUndoIds).
```

Build It

- Build proxy layer
 - Generic proxy to fetch data
 - Generic proxy to update data
 - Build wrapper procedures for specific tables



```

/* getRowObject.p */

DEFINE INPUT-OUTPUT PARAMETER pcRowIdent      AS CHARACTER NO-UNDO.
DEFINE INPUT         PARAMETER piRowsToReturn AS INTEGER    NO-UNDO.
DEFINE INPUT         PARAMETER phSdo         AS HANDLE      NO-UNDO.
DEFINE INPUT         PARAMETER phTT         AS HANDLE      NO-UNDO.

DEFINE VARIABLE hRowObj  AS HANDLE  NO-UNDO. /* RowObject temp table */
DEFINE VARIABLE hRowObjQry AS HANDLE  NO-UNDO. /* RowObject temp table query */
DEFINE VARIABLE iRowsRet AS INTEGER  NO-UNDO.

RUN serverSendRows IN phSdo
  (INPUT ?, /* ? = use the 2nd parameter as starting point */
   INPUT pcRowIdent,
   INPUT (IF pcRowIdent <> "FIRST":U AND pcRowIdent <> "LAST":U THEN
          TRUE
          ELSE
          FALSE),
   INPUT piRowsToReturn,
   OUTPUT iRowsRet,
   OUTPUT TABLE-HANDLE phTT) .

IF piRowsToReturn GT 0 THEN DO:
  IF iRowsRet LT piRowsToReturn THEN
    pcRowIdent = "".
  ELSE DO:
    phTT:DEFAULT-BUFFER-HANDLE:FIND-LAST().
    pcRowIdent = phTT:DEFAULT-BUFFER-HANDLE:BUFFER-FIELD("rowIdent"):BUFFER-VALUE.
  END.
END.
ELSE
  pcRowIdent = "".

RETURN.

```

```

/* setRowObject.p */

DEFINE INPUT  PARAMETER pcSdo AS CHARACTER NO-UNDO.
DEFINE INPUT  PARAMETER phTT AS HANDLE NO-UNDO.

FUNCTION getRowMod RETURNS CHARACTER PRIVATE ( ) FORWARD.

DEFINE VARIABLE cMessages AS CHARACTER NO-UNDO.
DEFINE VARIABLE cUndoIds AS CHARACTER NO-UNDO.
DEFINE VARIABLE hSdo AS HANDLE NO-UNDO.
DEFINE VARIABLE hRowObjUpd AS HANDLE NO-UNDO.
DEFINE VARIABLE lSuccess AS LOGICAL NO-UNDO INITIAL TRUE.
DEFINE VARIABLE lAdd AS LOGICAL NO-UNDO INITIAL NO.

$UPDATE-ROW:
DO ON ERROR UNDO, LEAVE:

    RUN create_update_record.

    RUN VALUE(pcSdo) PERSISTENT SET hSdo.

    RUN serverCommit IN hSdo(INPUT-OUTPUT TABLE-HANDLE hRowObjUpd, OUTPUT cMessages, OUTPUT cUndoIds).

    IF cMessages <> "" THEN
        LEAVE $UPDATE-ROW.

    IF getRowMod() = "A":U THEN
        RUN update_rowident.

END. /* $UPDATE-ROW */

```

```
/* GetCustomer.p - Fetch the Customer by name. */
```

```
DEFINE INPUT  PARAMETER pcName AS CHARACTER NO-UNDO.  
DEFINE OUTPUT PARAMETER TABLE-HANDLE phTT.
```

```
DEFINE VARIABLE cRowIdent AS CHARACTER    NO-UNDO INITIAL "FIRST".  
DEFINE VARIABLE hSdo      AS HANDLE      NO-UNDO.
```

```
DEFINE TEMP-TABLE ttCustomer  
  {pug/sdoCustomer.i}  
  {src/adm2/robjflds.i}.
```

```
phTT = TEMP-TABLE ttCustomer:HANDLE.
```

```
IF pcName = "" THEN DO: /* If no name is given, return an empty record. */  
  CREATE ttCustomer.  
  RETURN.  
END.
```

```
RUN pug/sdoCustomer.w PERSISTENT SET hSdo. /* Start SDO */
```

```
DYNAMIC-FUNCTION('selectName':U IN hSdo, pcName). /* Set Query */
```

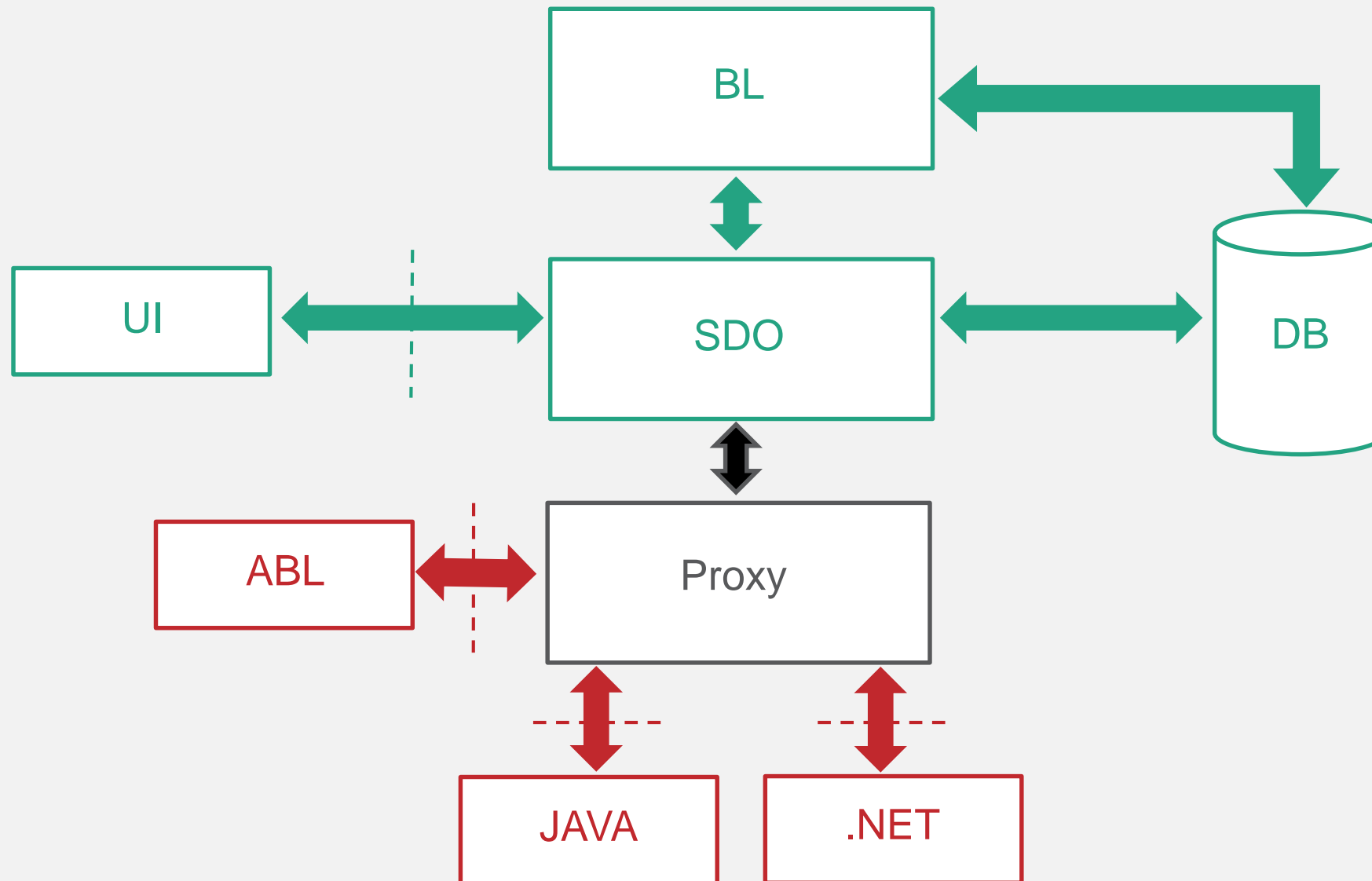
```
RUN pug/getRowObject.p (INPUT-OUTPUT cRowIdent, 0, hSdo, phTT). /* Fetch data into our temp-table */
```

```
RETURN.
```

```
FINALLY:
```

```
  RUN destroyObject IN hSdo NO-ERROR.  
  DELETE OBJECT phTT NO-ERROR.  
END FINALLY.
```

ADM2 with Proxy Layer – Expanded




```
/* GetCustomerSample.p */  
DEFINE VARIABLE hTT AS HANDLE NO-UNDO.  
  
RUN pug/GetCustomer.p ("Roundtable Software", OUTPUT TABLE-HANDLE hTT).  
  
/* Prove it!*/  
RUN pug/viewTempTable.w (hTT).  
  
RETURN.  
  
FINALLY:  
    DELETE OBJECT hTT NO-ERROR.  
END FINALLY.
```

```
/* DeleteCustomerSample.p */

DEFINE VARIABLE cError AS CHARACTER NO-UNDO.
DEFINE VARIABLE hBuf    AS HANDLE    NO-UNDO.
DEFINE VARIABLE hTT     AS HANDLE    NO-UNDO.

RUN pug/GetCustomer.p ("Roundtable Software", OUTPUT TABLE-HANDLE hTT).

hBuf = hTT:DEFAULT-BUFFER-HANDLE.
hBuf:FIND-FIRST().
hBuf::RowMod = "D".

RUN pug/SetCustomer.p (INPUT-OUTPUT TABLE-HANDLE hTT, OUTPUT cError).

IF cError <> "" THEN
    MESSAGE cError VIEW-AS ALERT-BOX ERROR.

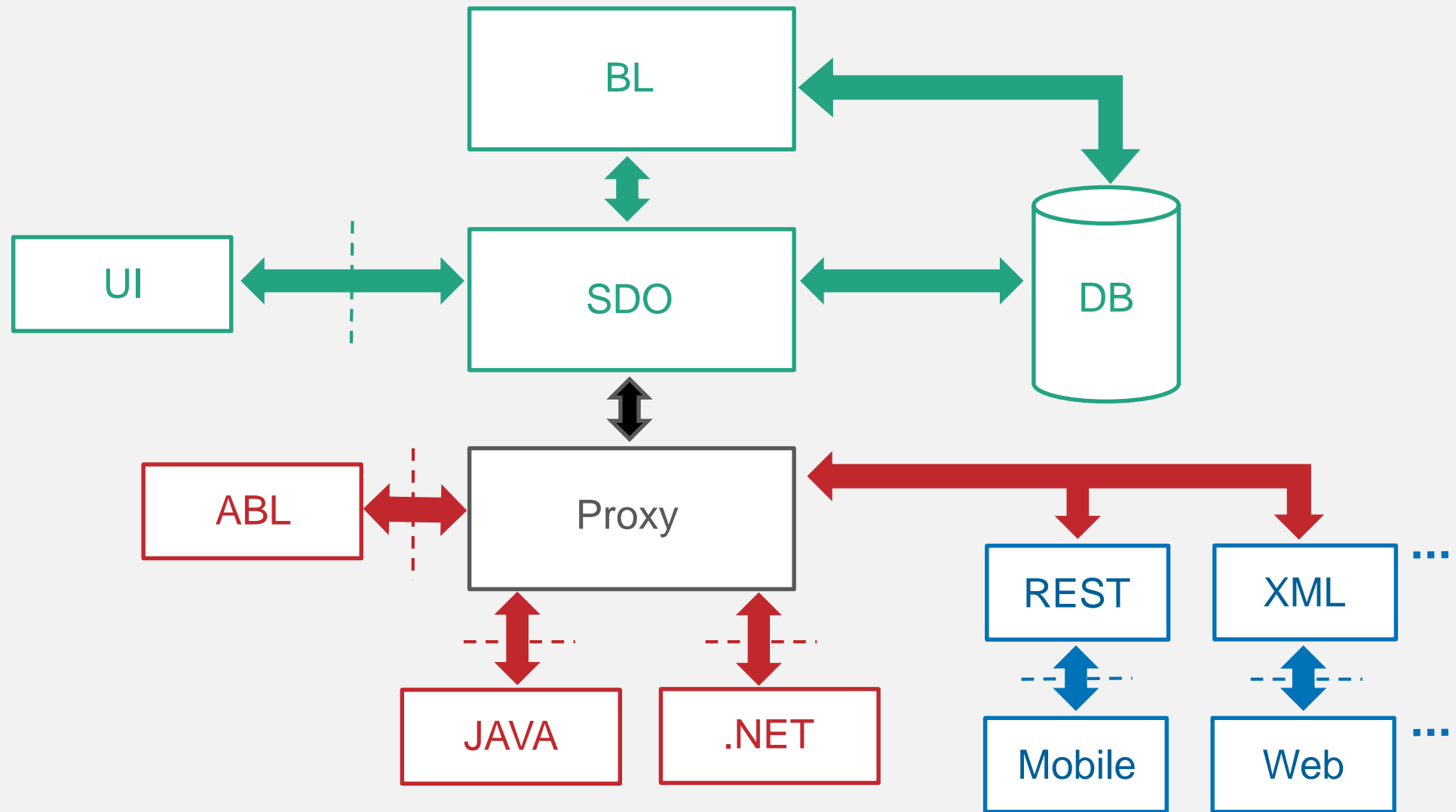
FINALLY:
    DELETE OBJECT hTT NO-ERROR.
END FINALLY.
```

Build It

- Define your model
 - RowObject temp-table
- Build integration points
 - serverSendRows
 - serverCommit
- Build proxy layer
- Expand



ADM2 with Proxy Layer – Expanded II



Expand

- OO Wrappers
- REST Service
- WebHandler
- Transform data as required



```
/* GetCustomerClassSample.p */
ROUTINE-LEVEL ON ERROR UNDO, THROW.

USING Progress.Lang.*.
USING pug.*.

DEFINE VARIABLE customer AS CLASS Customer NO-UNDO.

customer = NEW Customer("Roundtable Software").

DISPLAY
  customer:Name      FORMAT "x(50)" LABEL "Customer" SKIP
  customer:State     LABEL "State"     SKIP
  customer:SalesRep  FORMAT "x(50)" LABEL "Sales Rep"
  WITH FRAME A SIDE-LABELS TITLE "Customer Rep".

FINALLY:
  DELETE OBJECT customer NO-ERROR.
END FINALLY.
```

[https://public.sports9000.com/customers?name="Roundtable Software"](https://public.sports9000.com/customers?name=)

```
/* GetCustomerJson.p */  
  
DEFINE INPUT  PARAMETER pcName  AS CHARACTER NO-UNDO.  
DEFINE OUTPUT PARAMETER plcJson AS LONGCHAR  NO-UNDO.  
  
DEFINE VARIABLE hTT AS HANDLE NO-UNDO.  
  
RUN pug/GetCustomer.p ("Roundtable Software", OUTPUT TABLE-HANDLE hTT).  
  
hTT:WRITE-JSON ("LONGCHAR", plcJson, FALSE).  
  
RETURN.  
  
FINALLY:  
    DELETE OBJECT hTT NO-ERROR.  
END FINALLY.
```


Sample Code

```
ftp://ftp.roundtable-software.com/public
```



Questions?



EMEA
PUG challenge