Consultingwerk
software architecture and development

Patterns for migrating OpenEdge fat client GUI to stateless Web

Mike Fechner
Director

Consultingwerk Software Services Ltd.

- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany, subsidiaries in UK and Romania
- Customers in Europe, North America, Australia and South Africa
- Vendor of developer tools and consulting services
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration
- Experts in OpenEdge Application Modernization

SmartComponent Library framework

- Helps to protect your investment in your OpenEdge based application
- The framework is designed to modernize existing OpenEdge applications and to provide the foundation of new projects
- In the cloud and on premise
- OERA, CCS
- The architecture of the SmartComponent Library simplifies integration with future technologies and the implementation of new business requirements.

User Interface Flexibility

- Windows Desktop User Interfaces using .NET
- Angular Web Applications (Kendo UI)
- Mobile Applications (NativeScript)
- Open interfaces (RESTful)
- Partner User Interfaces (e.g. AKIOMA)

Kendo UI
THE ART OF WEB DEVELOPMENT

Agenda

- **Application Modernization Process**
- Modern OpenEdge Application Architecture
- Exposing Business Logic to modern Web Applications
- PASOE or WebSpeed
- Context Management
- Record Locking and transactions
- Validation
- Input Blocking from the Backend

Modernization Strategies

- Modernization of the whole application?
  - Going from ABL GUI to GUI for .NET or Web or Mobile
  - What is the *"final"* UI technology
  - GUI for .NET as an intermediate / integration with legacy GUI while the backend is rearchitected

- Or do we (first) add a few new features?
  - Mobile client for parts of the application
  - REST/REST(ful) interfaces for parts of the application
  - Reduce risk, gather first experience

Modernization Strategies

- Modernizing OpenEdge GUI (or TTY) to N-Tier first
  - Preparing the Application Backend for the Web
- Modernizing the whole application to the Web
  - Driven by demands from? Users, IT organization, marketing?
  - Definition of MVP – minimum viable product
  - How much functionality must be delivered on the web for user acceptance
- Developing one or multiple “satellite” web applications
  - Deliver quick and with reduced risk

Quality of the application

- Are parts of the application reusable?
  - With no or little changes
  - Are major functional changes required?
  - Are major changes to the database structure required?

- Can parts of the application serve to describe the requirements
  - Legacy code review as part of the requirements definition
  - Is the existing source code the only (complete) description of the application functionality?

Skills of Development team

- New development process (agile)
- New tools (Progress Developer Studio, SCM, Unit Tests, DevOps, Docker, Frontend tools)
- New architecture: Distributed
- New development languages
  - OOABL
  - html, JavaScript, TypeScript, rapidly changing
  - Desktop technologies
  - Web and Mobile frameworks

Modernization Project example: OSIV / OSC

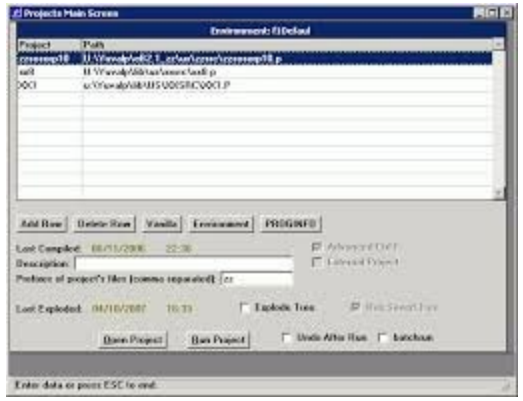
- OSIV Service Center: Joint venture of 7 Swiss counties (cantons)
- Maintaining state insurance for occupational disabilities
- Approval of therapies
- Perform Disability and treatment Assessments
- Billing (by doctors, clinics, opticians, occupational disabilities, etc.)
- Document management
- 1300 users
- Very specific domain functionality
- Accepted by the user base, no real competition




Why “modernization”

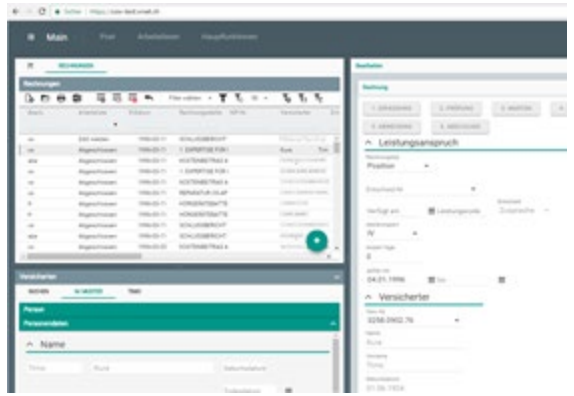
- Maintenance effort high
- Training of new users and developers hard
- Aged technology
- Resources / Motivation of developers / Agile methods

OSIV3G: Soft Migration



Current OSIV 5.x

Harvesting of existing Code



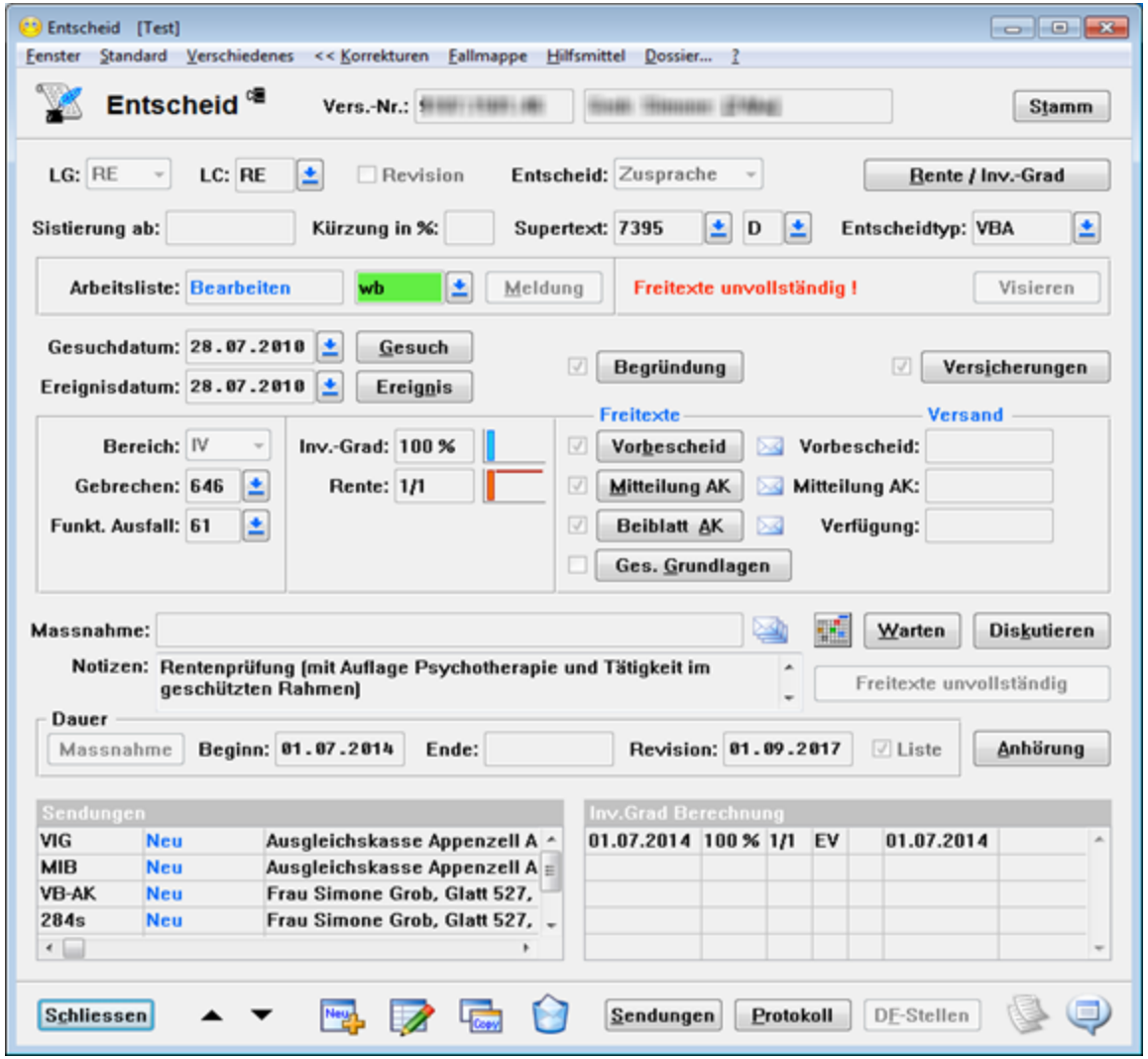
Migrated Application Modules

OSIV-DB

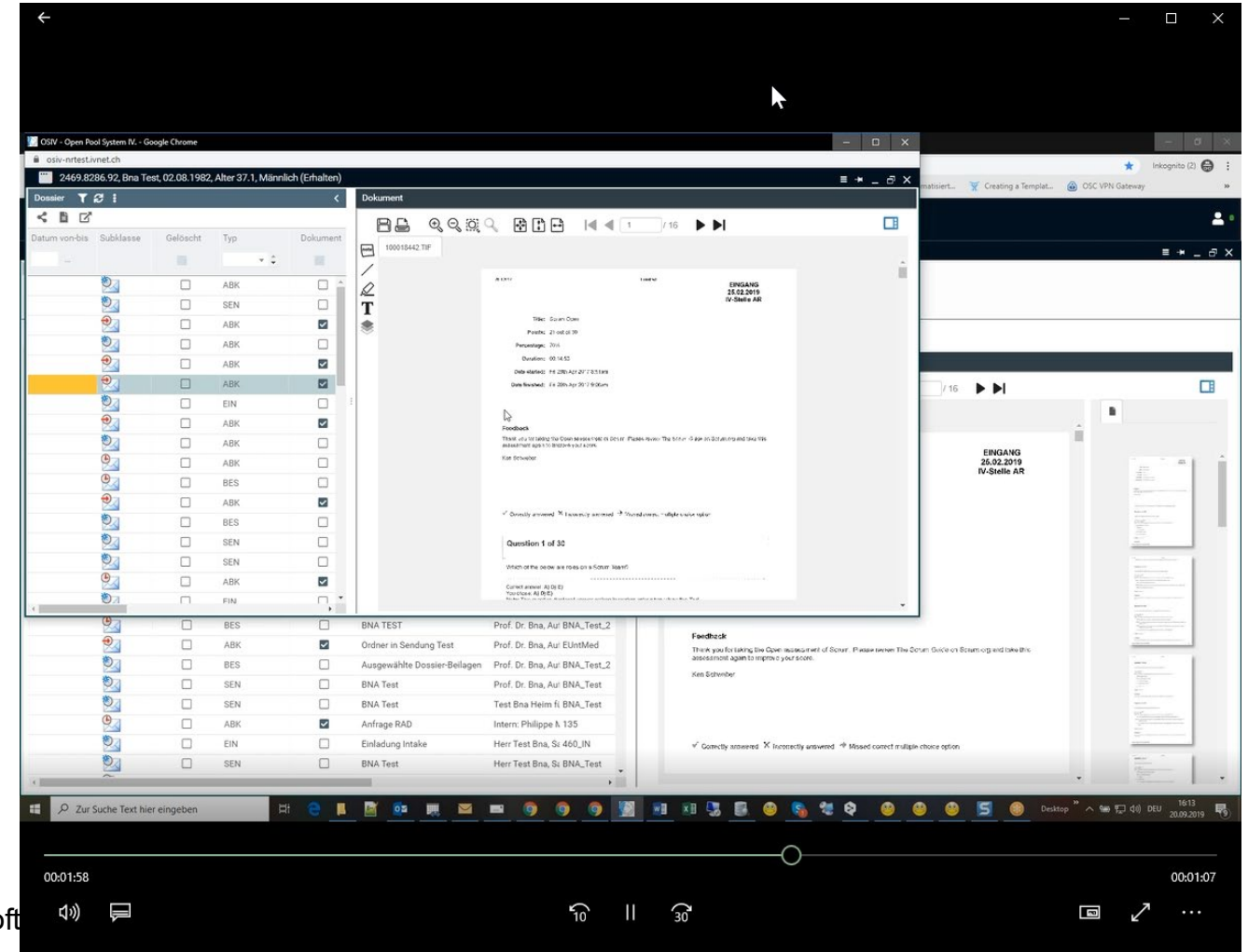
+ additional Fields and Tables e.g. GUID‘s

Framework DB‘s

Sneak preview



Video Demo of modernized application



Agenda

- Application Modernization Process
- **Modern OpenEdge Application Architecture**
- Exposing Business Logic to modern Web Applications
- PASOE or WebSpeed
- Context Management
- Record Locking and transactions
- Validation
- Input Blocking from the Backend

OERA OpenEdge Reference Architecture

- Architecture blue print for service-oriented OpenEdge applications
- Initially released with OpenEdge 10.0 (15+ years)
- Primary goals at the time
  - AppServer enabling OpenEdge applications
  - Building non-monolithic OpenEdge applications
  - Supporting client flexibility
  - Providing guidance for use of the ProDataset
  - Providing guidance for use of OOABL (later, around OE10.1+)

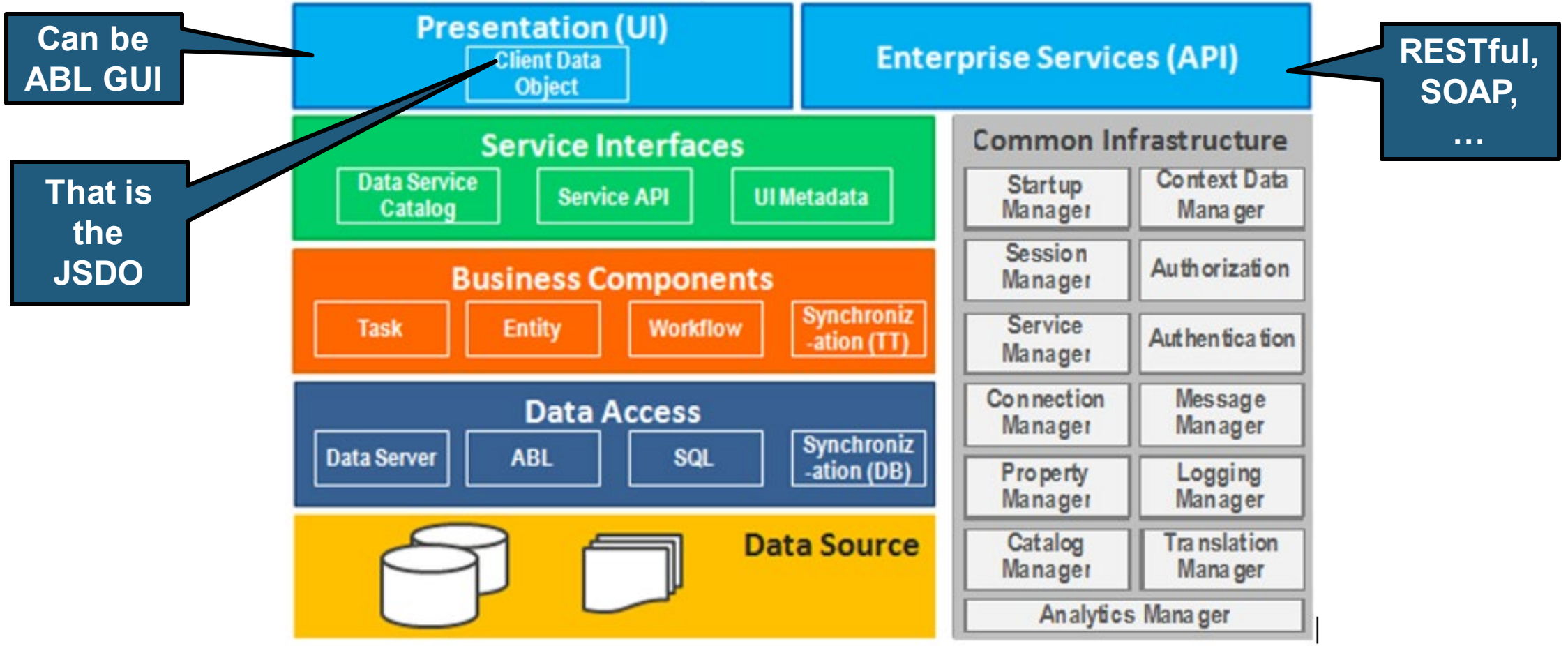
OERA today

- Fast forward to 2015 …
- Modernization of OpenEdge applications more relevant than ever; especially since Telerik acquisition and demands for UI flexibility
- OEAA – OpenEdge Application Architecture, redefining the OERA
- OERA back on focus, foundation of the **CCS (common component specification)** project as a vehicle for community and Progress driven architecture-spec efforts
- More detailed specs, rather than just programming samples
- Specs that an application or framework could be certified against
- CCS starting to influence *"in-the-box"* features

Business Entities

- Business Logic Component in the Business Service Layer
- Manages a set of database tables
  - Customer
  - Order/OrderLine/Item (read-only)
- CRUD actions (create, read, update, delete)
- Custom actions, verbs of the entity (ShipOrder, small Business Tasks)
- Primary backend component for the JSDO
  - Kendo UI, Kendo UI Builder
  - NativeScript

The OpenEdge Application Architecture (OEAA)



Service Interface(s)

Bouncer

Babelfish

Agenda

- Application Modernization Process
- Modern OpenEdge Application Architecture
- **Exposing Business Logic to modern Web Applications**
- PASOE or WebSpeed
- Context Management
- Record Locking and transactions
- Validation
- Input Blocking from the Backend

Exposing Business Logic for Web Applications

- Today two major API styles in the OpenEdge world
- RESTful and JSDO based
- For web applications mainly accessing the OpenEdge backend, consider using the JSDO
- For web applications accessing multiple backends, using RESTful might be preferable
  - RESTful known API style by modern web developers

- Key to success is to support both, sometimes a mix

RESTful

- Describes an addressing and interaction scheme for resources (e.g. a record) and collections (e.g. a list of records)
- Interaction based on http
- http verbs describe interaction (GET, POST, PUT, DELETE, …)

- Different way of thinking compared to referring to service endpoints and passing parameters to a service method

RESTful Samples

- GET **http://localhost:8820/Customers/42** performs read request of a single customer
- GET **http://localhost:8820/Countries/SA/Customers** queries customers as a sub-collection of Country SA

- PUT **http://localhost:8820/Customers/42** updates Customer 42

- POST **http://localhost:8820/Customers** creates a customer
- POST **http://localhost:8820/Countries/SA/Customers** creates a customer in South Africa

JSDO – JavaScript Data Object

- JavaScript Library develop by Progress Software to provide access for JavaScript (Web Browser, Mobile, Rollbase) clients to OpenEdge Data Object Services (Business Entities)
- Introduced in OpenEdge 11.2 for OpenEdge Mobile
- Foundation for OpenEdge access from Telerik UI components
- Included in Telerik Platform and Kendo UI Builder (KUIB)
- Can be used with any JavaScript or TypeScript client, including NativeScript
- Github, Apache license, royalty free

JSDO – JavaScript Data Object

- JSDO requires Catalog which describes the backend resource capabilities and data structure
- RESTful update (POST/PUT) does provide before image to server
  - JSDO will (typically) provide before-image
- RESTful modification (PUT/DELETE) based on resource URL
  - JSDO will post before-image with ROW-STATE
- JSDO operations typically based on ProDataset and list of records
  - dsCustomer: { eCustomer: [{ CustNum: 1 }] }
  - RESTful PUT/POST on single record { CustNum : 1 }

Agenda

- Application Modernization Process
- Modern OpenEdge Application Architecture
- Exposing Business Logic to modern Web Applications
- **PASOE or WebSpeed or classic AppServer**
- Context Management
- Record Locking and transactions
- Validation
- Input Blocking from the Backend

PASOE or WebSpeed or classic AppServer

- Modern Web Applications do not rely on backend to serve dynamic web pages, so WebSpeed's speciality not relevant anymore
- Classic AppServer complex for exposing to REST and SOAP
  - REST Adapter enforces JSON content type

- OpenEdge 12 was released last year!
- OpenEdge 12 does no longer support WebSpeed and the classic AppServer in favor for Progress Application Server for OpenEdge (PASOE)

PASOE Web handler

- Available since OpenEdge 11.6
- Web handlers provide a very flexible way to handle web requests
- Synchronous request-response pattern
- Supports html page generation
- Supports service requests as well
- Flexible enough to provide an alternative to the REST Adapter and Web Services Adapter (SOAP)
- ABL classes, extending OpenEdge.Web.WebHandler

Samples

- https://github.com/consultingwerk/RESTful-Samples

Agenda

- Application Modernization Process
- Modern OpenEdge Application Architecture
- Exposing Business Logic to modern Web Applications
- PASOE or WebSpeed or classic AppServer
- **Context Management**
- Record Locking and transactions
- Validation
- Input Blocking from the Backend

Context Management

- The basics: User, login company, language, …
- Screen-Context: Last used customer, last used filter, …
- Complex selection collected over a series of requests, …
- A web-cart, for an online store

- Transient information, scoped to the current user / login session

PASOE support for context management

- PASOE form based login model maintains a session context for us; configuration-based
- Based on client-principal, hybrid authentication realm

- SESSION:CURRENT-REQUEST-INFO:GetClientPrincipal()

- Key to multi-tenancy, auditing,
- Hybrid Realm supports custom properties in the client-principal (user full name, language, email) … session context that does not change

PASOE support for context management

- And the what about rest, context that may change?
- Some session context (last used filter) might better be stored in the user profile, outside the scope of a login session
- Client-principal has a SESSION-ID property that can be used to safely identify a user's login session on the backend

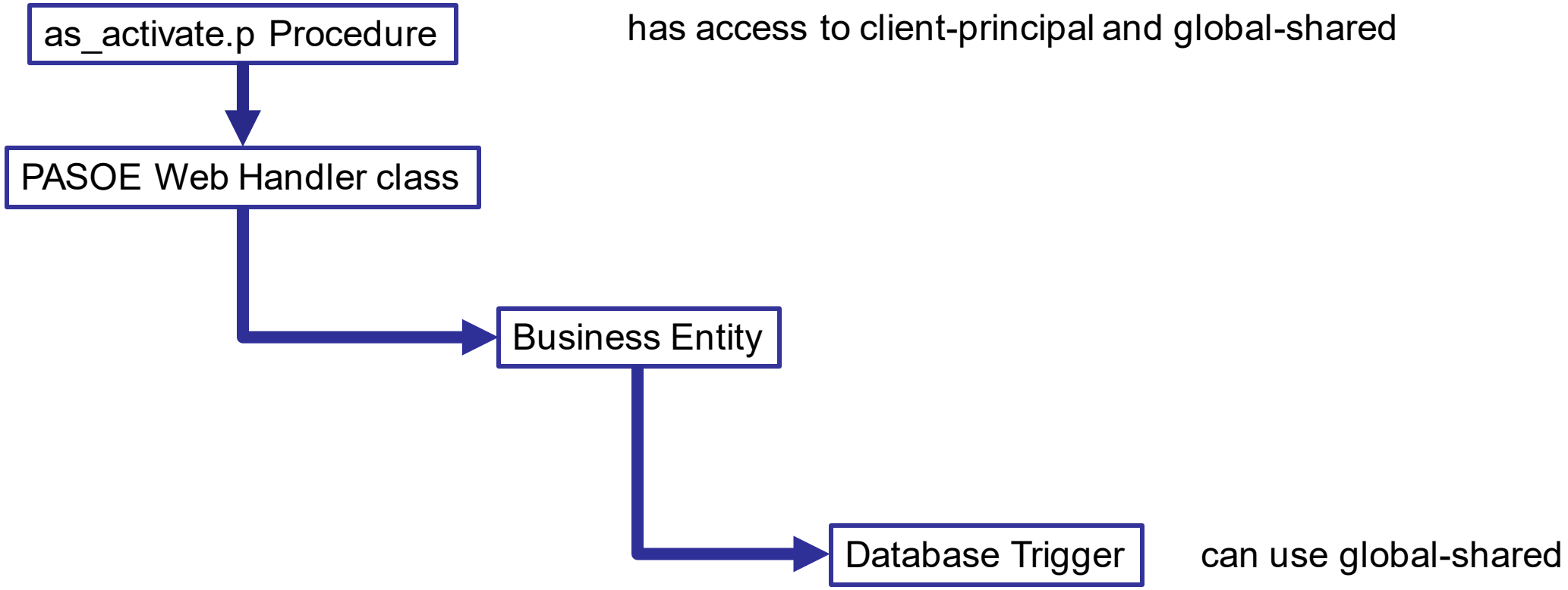
PASOE support for context management

- Roll your own session management based in hCP:SESSION-ID
- Store your own session context:
  - General purpose database table (name/value pairs or XML/JSON of a temp-table or ProDataset)
  - Use flat files (XML/JSON of temp-table or ProDataset)
  - Use specific tables in a database (e.g. for the web cart)
- Keep a last-access time-stamp to know when to remove session context of inactive/timed-out sessions

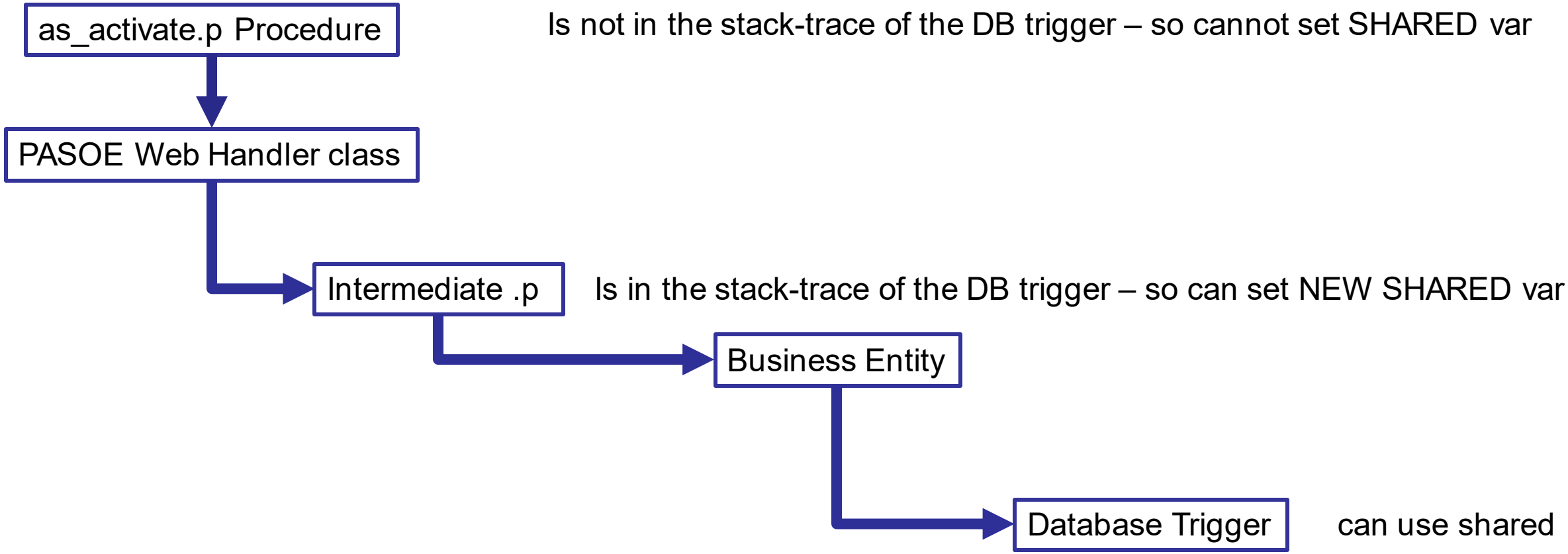
GLOBAL SHARED or SHARED variables …

- GLOBAL SHARED variables are less trouble
- SHARED variables should be reconsidered – many of them may be replaced with GLOBAL SHARED, usually a bad legacy
- Class based code (most new code, PASOE Web handlers) has NO access to any GLOBAL SHARED SHARED context

DB Trigger relying on a GLOBAL SHARED variable



DB Trigger relying on a SHARED variable



Agenda

- Application Modernization Process
- Modern OpenEdge Application Architecture
- Exposing Business Logic to modern Web Applications
- PASOE or WebSpeed or classic AppServer
- Context Management
- **Record Locking and transactions**
- Validation
- Input Blocking from the Backend

Record Locking and transactions

- PASOE web applications are state free
- PASOE does not support locking records spanning multiple requests
- PASOE does not support transaction blocks spanning multiple requests
- Within a request record locking and transactions behave as usual

Record locking use cases

- Ensure that no data is unintentionally overwritten by a user
- Ensure that during validation (or a processing block) no related data is altered that would void any integrity check done – while we're still running the remainder of the request
  - typically within a single request, so traditional locking works
- Blocking other users from performing a similar activity another user already does
  - e.g. performing the month end transactions

Optimistic locking

- Optimistic locking allows the user to modify a record and when pressing “Save”, the application verifies if the record has been changed by another user
- Effective way of avoiding unintended overwriting of changes from another user
- Was present in the ADM2 SDO
- Implemented by the ProDataset SAVE-ROW-CHANGES method
  - verifying this on a field by field or record by record basis

Optimistic locking implementations

- JSDO supports optimistic locking through the ProDataset BEFORE-TABLES
- Before Table typically not known by RESTful clients
- Alternative implementations may rely on
  - Record version, timestamp or before image in session context

- However systems may also allow to overwrite changes from another user without worrying …
  - Is it really a problem if a second user updates an Address record again?
  - This should be more a business concern than technical issue

Interaction with TTY/GUI clients that hold locks

- Preferably use EXCLUSIVE-LOCK NO-WAIT and IF NOT AVAILABLE AND LOCKED to handle locks from other sessions
- Always preferably to fail fast and give the user a quick response and redo option

- AppServer processes should work with a short –lkwtmo setting (10 seconds is the minimum)
- Application needs to deal nicely with STOP conditions raised by lock wait timeout, OpenEdge 11.7.4 and OpenEdge 12 simplify handling of STOP via CATCH

Logical resource locking

- In rather rare cases where web applications require lock of “resources”, consider implementing a logical record locking
- Should be implemented as a framework function
- LockResource (resource identifier, session id, lock timeout)
  - might also extend lock
- IsResourceLocked (resource identifier) – by someone else
- IsResourceLockedByMe (resource identifier) – by current session

- Additional lock management functions: Scheduled release of locks and manual release of locks by admin

```
USING Consultingwerk.SmartFramework.Lock.* FROM PROPATH.

{Consultingwerk/SmartComponentsDemo/OERA/Sports2000/dsSalesRep.i}

DEFINE VARIABLE oLockService AS ISmartLockService NO-UNDO .
DEFINE VARIABLE lOk          AS LOGICAL            NO-UNDO .

oLockService = {Consultingwerk/get-mandatory-service.i ISmartLockService} .

/* API for locking with throwing an error */
oLockService:AcquireLock ("Salesrep":U,          // Table Reference
                          eSalesrep.Salesrep,    // Key value(s)
                          600,                   // Lock duration, 10 minutes
                          TRUE) .                // Throw when locked by another session


/* API for locking without throwing an error, NO-ERROR */
lOk = oLockService:AcquireLock ("Salesrep":U,          // Table Reference
                                eSalesrep.Salesrep,    // Key value(s)
                                600,                   // Lock duration, 10 minutes
                                FALSE) .               // Do not throw when locked by another session

/* API for releasing the lock */
oLockService:ReleaseLock ("Salesrep":U,          // Table Reference
                          eSalesrep.Salesrep) . // Key value(s)
```

Agenda

- Application Modernization Process
- Modern OpenEdge Application Architecture
- Exposing Business Logic to modern Web Applications
- PASOE or WebSpeed or classic AppServer
- Context Management
- Record Locking and transactions
- **Validation**
- Input Blocking from the Backend

Validation

- “don’t trust the client”!!!
- Always validate EVERY single field value. Even when you expect that the client prohibits such entry (e.g. by a checkbox or a combo-box)
  - You don’t need to be a highly qualified hacker to mess up a web application
  - All browsers support a debugger that allows altering the HTML and JavaScript Code in memory

Validation

- Always provide a function to validate all fields of a record
- This must be the primary validation function when the client submits a record (new, modified, deleted)

- The CCS describes a method to additionally provide field level validation (on leave)
  - Validates the whole record
  - Provides information about updated fields

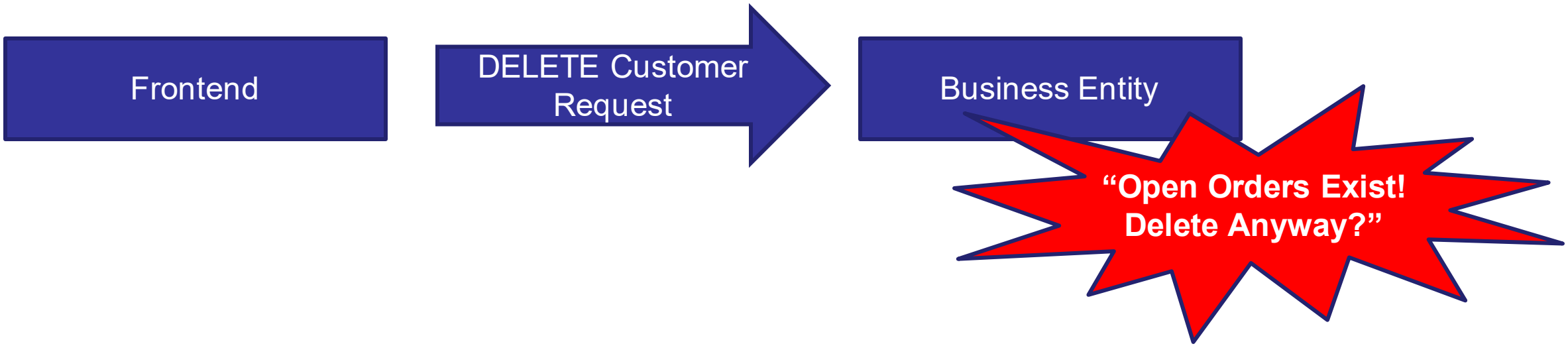
Agenda

- Application Modernization Process
- Modern OpenEdge Application Architecture
- Exposing Business Logic to modern Web Applications
- PASOE or WebSpeed or classic AppServer
- Context Management
- Record Locking and transactions
- Validation
- **Input Blocking from the Backend**

Input blocking from the Backend

- Progress Application Server does not support Input Blocking on the UI
- Once AppServer is invoked, client waits for response
- Web technologies such as Socket.IO may be used to send messages from Backend to frontend
  - Back not vice-versa, no WAIT-FOR
- **When UI can foresee** that AppServer **may** require additional information when processing request, try adding this to the request
  - However UX should not be ignored. Too many irrelevant options confusion / annoying to users

Input Blocking, fat client ABL




```
IF CAN-FIND (FIRST Order OF Customer WHERE ....) THEN
    MESSAGE "Open Orders Exist! Delete Anyway?"
        VIEW-AS ALERT-BOX QUESTION BUTTONS YES-NO UPDATE response .
```

Input Blocking, fat client ABL


```
IF CAN-FIND (FIRST Order OF Customer WHERE ....)
   AND poRequest:DeleteOpenOrders = TRUE THEN ...
```

Example challenge: Interaction between Back and Frontend (from OSC / OSIV project)

- Existing OSIV Business Logic in large parts suitable as foundation for new OSIV3G (functional and structural), especially validation
- Validation may also provide color coding to represent field status etc.
- Validation may have to prompt the user
- Web applications typically:
  Request (from browser) – Response (from server)
- No Input-Blocking (not possible to wait for user input in Business Logic)

Sample: Yes/No PROMPT in validation

- Demand is to keep the validation flow in major parts “as is“
- Validation may encounter question requiring user input: “Are you sure?” etc.

Sample: Yes/No PROMPT in validation

```
/* ---------- */
/* Verstorben */
/* ---------- */
if (date(Stamm.Todes_Dat:screen-value) <> ?) then do:
  /* Testen, ob Versicherter gerade eben verstorben ist. */
  if (EDIT_MODE = "UPDATE") then do:
    find Stamm no-lock where recid(Stamm) = MAIN_REC_ID.
    if (Stamm.Todes_Dat = ?) then do:
      /* Versicherter wurde soeben auf verstorben gesetzt. */
      run set_message_param(Stamm.Todes_Dat:screen-value).
      run user_warning("Der Versicherte ist am $1 verstorben. ~n~n" +
                       "Die zugehörigen Wohnadressen werden gesperrt.~n" +
                       "Überprüfen Sie, ob noch Revisionen vorgesehen sind~n" +
                       "und/oder Hilfsmittel zurückgenommen werden müssen.~n",
                       output continue).
      if not continue then return error.
    end.
  end.

end. /* if verstorben */
```

Sample: Yes/No PROMPT in validation

```
MSG = {Consultingwerk/get-service.i IMsg} .
SYS = {Consultingwerk/get-service.i ISys} .
MOD_ADD = {Consultingwerk/get-service.i IModAdd} .

if (eStammBefore.Todes_Dat = ?) then do:
    /* Versicherter wurde soeben auf verstorben gesetzt. */
    MSG:set_message_param(string (eStamm.Todes_Dat) /*:screen-value*/).

    continue = MSG:user_warning("Der Versicherte ist am $1 verstorben. ~n~n" +
                                "Die zugehörigen Wohnadressen werden gesperrt.~n" +
                                "Überprüfen Sie, ob noch Revisionen vorgesehen sind~n" +
                                "und/oder Hilfsmittel zurückgenommen werden müssen.~n",
                                this-object:GetClass():TypeName,
                                "eb09af84b1e2197b:4cb274e8:15608162bb6:-8000",
                                string (eStamm.SelfHdl)).

    if not continue then do:
        DatasetHelper:AddErrorString(buffer eStamm:handle, "_CANCEL") .
        return .
    end.

    /*if not continue then return error.*/
end.
```

Migration using MessagePrompt API (SmartComponent Library framework)

- Backend – API maintains list of questions (unanswered and answered)
- Same API Call may ask a new question of return an existing answer
- Supports multiple questions per routine: Questions are flagged with GUID identifying their location in code
- Support for multiple iterations (Loops, FOR EACH, …): Each question is also flagged with a records PUK value (GUID, combined key fields)

JSON Representation of the question

```
{
  "SerializedType": "Consultingwerk.Framework.MessageInteraction.Question",
  "MessageText": "Der Versicherte ist am 24\/12\/50 verstorben. \n\n
                  Die zugehörigen Wohnadressen werden gesperrt.\n
                  Überprüfen Sie, ob noch Revisionen vorgesehen sind\n
                  und\/oder Hilfsmittel zurückgenommen werden müssen.\n",
  "MessageButtons": "YesNo",
  "MessageReply": "Unanswered",
  "DefaultReply": "ReplyYes",
  "MessageID": "eb09af84b1e2197b:4cb274e8:15608162bb6:-8000",
  "MessageContext": "ac54bf82-56c4-bab2-2514-8e3d5c34775d"
}
```

Questions


Consultingwerk
software architecture and development