Consultingwerk
software architecture and development

OpenEdge Application refactoring

The full stack modernization framework

SmartComponent Library

by Consultingwerk

Mike Fechner

- Director, Lead Modernization Architect and Product Manager, Architect of the SmartComponent Library and WinKit
- Specialized on object-oriented design, software architecture, desktop user interfaces and web technologies
- 34 years of Progress experience (V5 … OE12)
- Active member of the OpenEdge community
- Frequent speaker at OpenEdge related conferences around the world

Consultingwerk Application Modernization Solutions

- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany, subsidiaries in UK, USA 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

Agenda

- **Overview**
- Framework Backend Architecture
- User-Interface Repository
- Migrating Update-Editing into RESTful Service
- Migration UI Layouts
- Migrating UI Trigger Logic

At a glance

- Full stack modernization framework
- Backend as domicile for business logic
- Relational and object-relational (ORM)
- RESTful out of the box
- Multiple user interface options: Desktop, Web and Mobile
- Application Framework: Authentication, Localization, Menu, Workflows, …
- Integration with existing OpenEdge applications and frameworks
- Modernization tooling: Analyze and migrate existing source-code

At a glance

- No vendor lock-in: full source code provided to customers
- Future proof: ahead the demands of our customers
- Designed for customizability and extensibility:
  No customer has the same requirements
- DevOps enabled: templates for build, test and deploy
- Supported on OpenEdge 11.7, 12.2 and 12.8
- Support and Maintenance offering

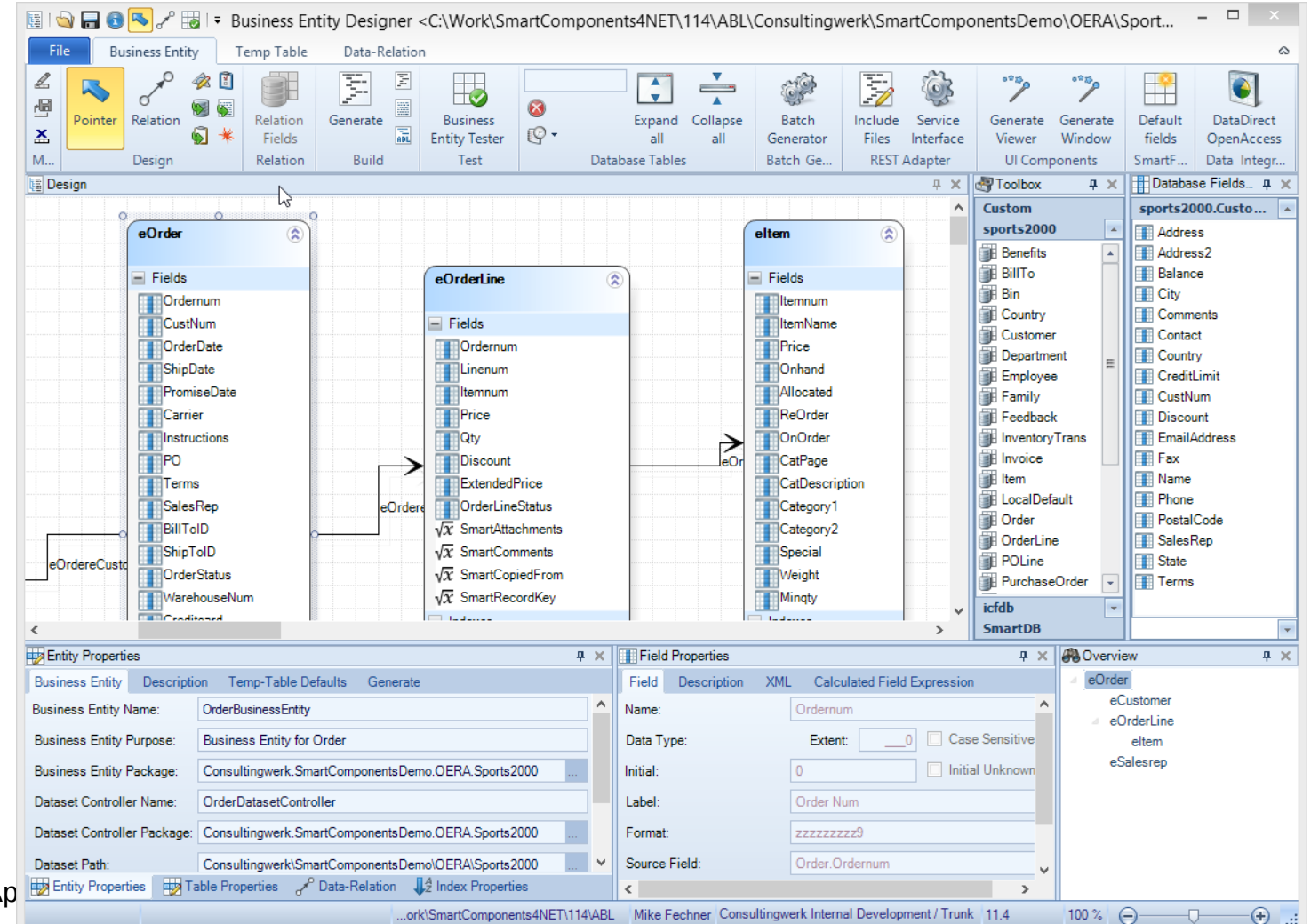
Application Framework Features

- Authentication, including client-principal support, Authentication Gateway and multi-tenancy
- Flexible authorization (menu, toolbar, business logic, custom tokens, …), can be used for data-related authorization (e.g. Regions)
- Localization (full UI translation), messages, application data
- Definition based referential integrity
- Batch scheduler (repeating, just once, run now)
- Enhanced Unit testing framework
- …

Backend Architecture

- Backend central component of modern architecture
- Backend responsible for reuse of business logic, future proofness
- Service oriented, OpenEdge Reference Architecture, Common Component Specification
- **Tool support** during development of Business Entities, Business Tasks and application services
- Relational and object-relational (ORM)
- Modular framework architecture
- RESTful out of the box

SmartComponent Library



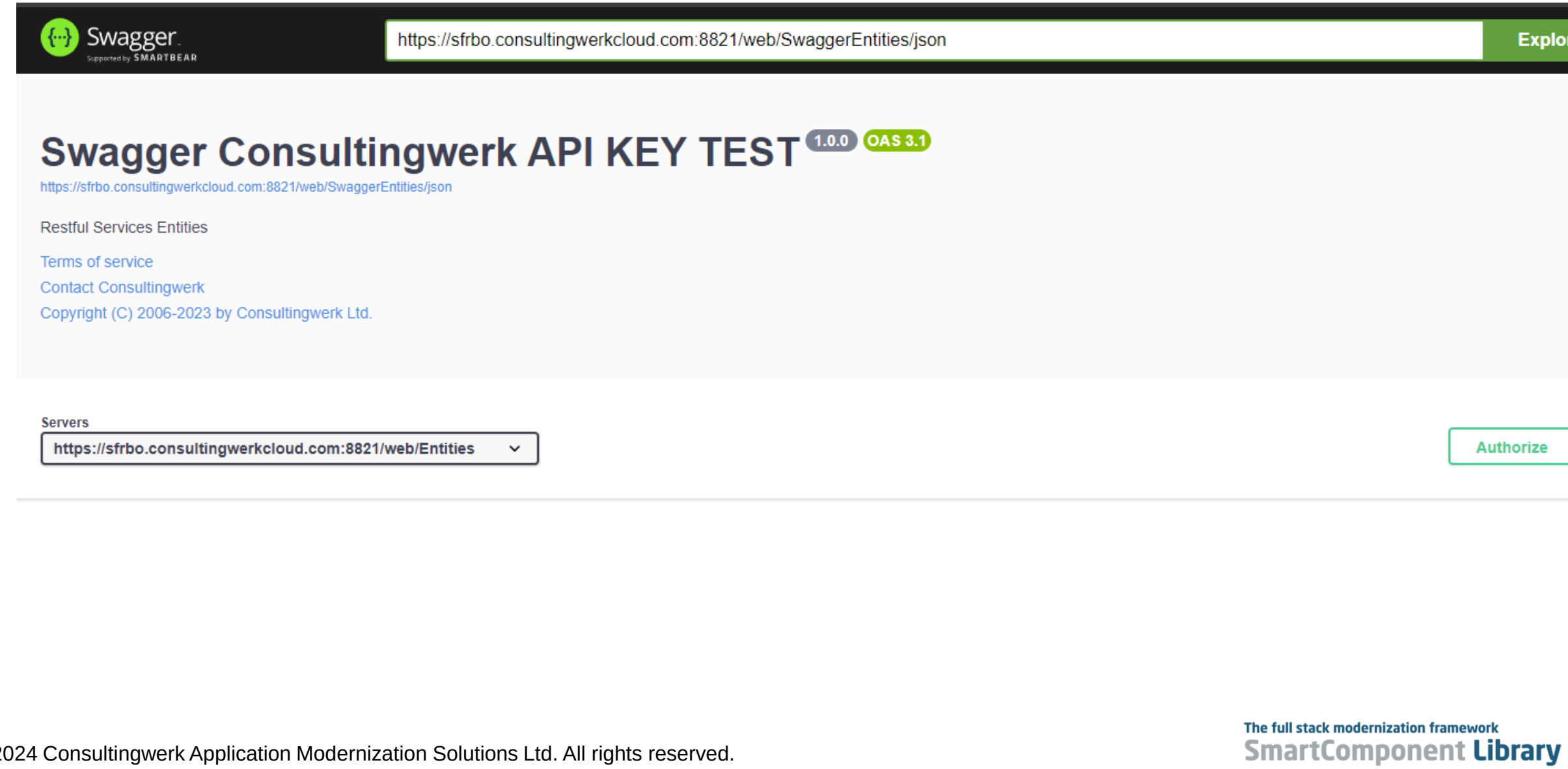
REST/RESTful

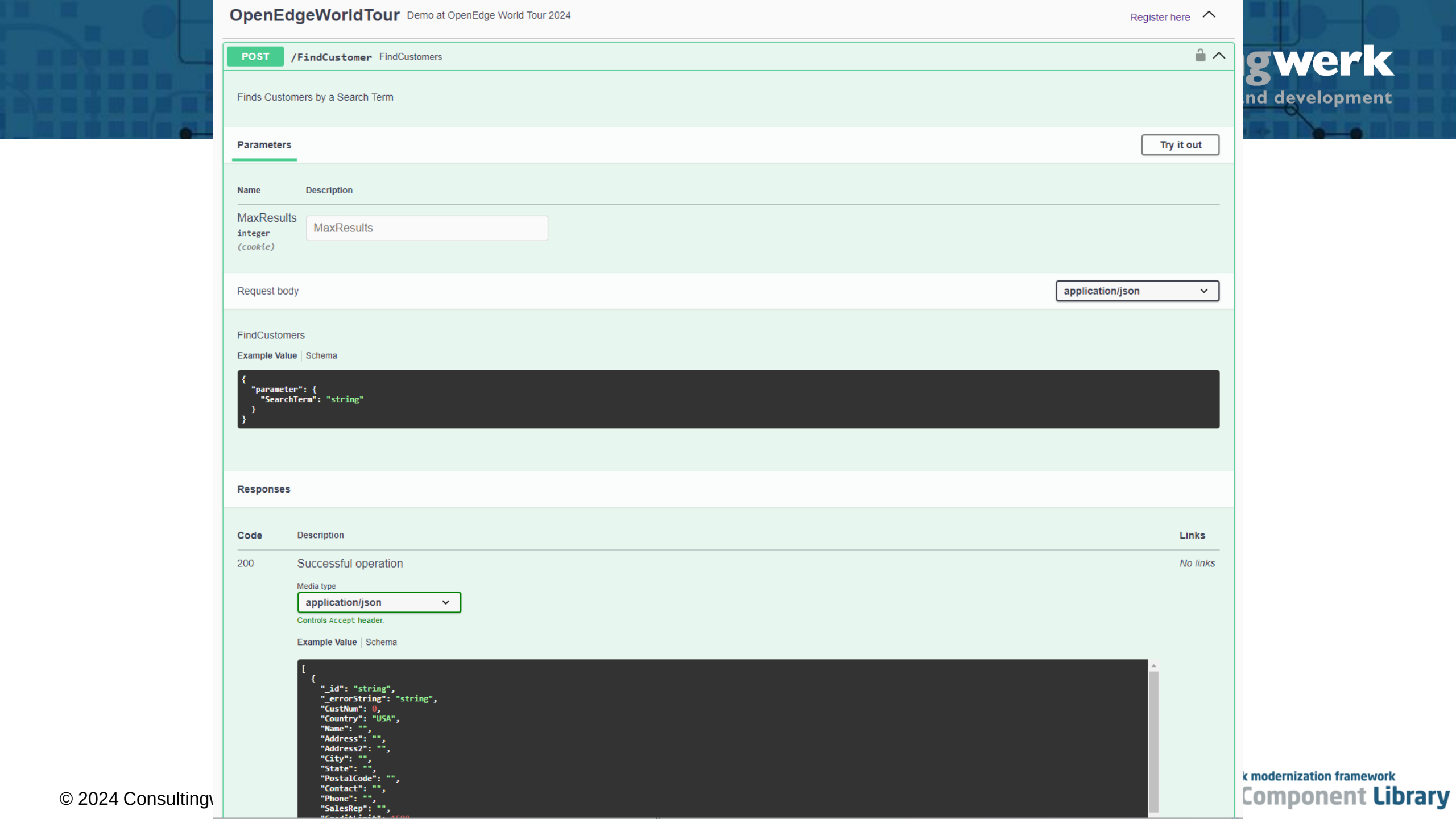
- Standard protocol for application integration and UI flexibility
- SmartComponent Library provides the most simple and most flexible method of implementing RESTful service with OpenEdge
- Typical use-cases
  - Implement new functionality as RESTful services
  - Provide existing (legacy) functionality as RESTful service
- Open API / Swagger documentation / test suite out of the box, generated automatically
- No need to deploy services, code declares the API
- Sophisticated authentication and authorization features

REST/RESTful – new Features

- Full support for JSON schema / Open API 3.0 – supporting implementation of every interface
- API-first design – implement service based on Open API specification; typical requirement in integration projects
- Generation of ABL clients for existing REST services
- Full support for ABL legacy code remaining like SHARED/GLOBAL SHARED variables when using OOABL (e.g. database trigger or executed legacy procedures)

```
CLASS Demo.CustomerSearchBusinessTask
    INHERITS BusinessTask:

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


    @RestMethod (address="/FindCustomer",
                 parameterClassName="Demo.FindCustomerParameter",
                 requestMethod="post",
                 request="parameter",
                 response="dsCustomer:eCustomer",
                 mapCookies="MaxResults=MaxResults",
                 description="Finds Customers by a Search Term",
                 tags="OpenEdgeWorldTour").

    @ParameterSchema (datasetname="dsCustomer",
                      schemafile="Consultingwerk/SmartComponentsDemo/OERA/Sports2000/dsCustomer.xsd",
                      datasetSchemaType="customerResult").
    METHOD PUBLIC VOID FindCustomers (INPUT-OUTPUT DATASET dsCustomer,
                                      poParameter AS FindCustomerParameter):

        MESSAGE "FindCustomers":U poParameter:SearchTerm SKIP
                "Max Results: ":U poParameter:MaxResults .

    END METHOD.
```





Demo

- Build and use Business Entity using the Business Entity Designer
- Export Business Entity as RESTful service

Aryza Navigate (NL)

“… At Aryza, we're committed to constantly innovating and pushing the boundaries to provide our clients with the best solutions possible. That's why we've joined forces with Consultingwerk to invest in the introduction of the SmartComponents Library as a foundation of the modernization of Aryza Navigate. …

By leveraging web-based solutions and harnessing the power of modern technology, Aryza Navigate will not only meet but exceed the evolving needs of our clients. With this enhanced capability, our clients will have a significant competitive edge in their respective industries. They'll be able to navigate the complexities of their business landscape with unparalleled efficiency, agility, and foresight.

This collaboration is not just about improving our product; it's about empowering our clients to thrive in an increasingly dynamic and competitive market environment. We're excited about the possibilities this partnership brings and the positive impact it will have on our clients' success.”

User Interface Flexibility

- Windows Desktop User Interfaces with .NET (GUI for .NET or .NET pure)
- Web applications (Angular, Kendo UI)
- Mobile apps (Angular, NativeScript)
- .NET Razor/Blazor
- Open Interfaces (e.g. RESTful), fully „headless“ applications
- Partner User Interfaces (e.g. Build.One)

Kendo UI
THE ART OF WEB DEVELOPMENT

Migration Strategies

- ABL GUI to SmartComponent Library GUI for .NET
- ABL GUI to SmartComponent Library Angular
- ABL GUI to custom web frontend
- ABL GUI to partner frontend
- ABL GUI to headless

Repository based user-interface design

- Set of database tables designed to store layout of screens
- Reusable components
- Developer tools: Wizards and (WYSIWYG) Designers
- Runtime components: Runtime rendering or code generation
- Prepared for runtime customization (modifying layout based on session attributes, e.g. users, groups, day of week)
- Source code is a very inflexible „repository“ for UI layout
- Repository foundation for multi-UI interface designer

Demo

- SmartComponent Library UI Designer Tooling
- GUI for .NET
- Angular Web App (integrated)

SmartFramework Administration

Windows | Mike Fechner

Consultingwerk
software architecture and development

- Authentication & Authorization
  - Field Security Item Maintenance
  - Security Assignment
  - Security Assignment Verification
  - Menu Security Maintenance
  - Security Mass-Assignment
  - Security Object Maintenance
  - Security Realm Maintenance
  - Security Token Maintenance
  - Verwaltung Benutzer
  - Add User
  - Verwaltung Benutzergruppen
  - Web Sessions
- Menu
- Scheduler
- System

User Maintenance

User N...	Login Company Name	Full Name	E-mail
admin		Default Admin User	
demo	Consultingwerk Ltd.	Demo User	demo@consultingwerk.de
deniz	Consultingwerk Ltd.	Deniz Werth	deniz.werth@consultingw
lutz	**Consultingwerk Ltd.**	**Lutz Fechner**	**lutz.fechner@consulting**
marko	Consultingwerk Ltd.	Marko Rüterbories	marko.rueterbories@cons
mikefe	Consultingwerk Ltd.	Mike Fechner	mike.fechner@consultingv
mikefe	Demo Company	Mike Fechner (Demo)	mike@fechner.de
oana	Consultingwerk Ltd.	Oana Sucigan	oana.sucigan@consulting
peter	Consultingwerk Ltd.	Peter Judge	peter.judge@consultingwe
radu	Consultingwerk Ltd.	Radu Nicoara	radu.nicoara@consultingv
richard	Consultingwerk Ltd.	Richard Kelters	r.kelters@flusso.nl
robert	Consultingwerk Ltd.	Robert Chelaru	<robert.chelaru@consultir
root		System Administrator	root@root.com

USER | GROUP ASSIGNMENT

ADD | COPY | SAVE | CANCEL | DELETE | COPY SECURITY ASSIGNMENT

Name *
lutz

Login Company
Consultingwerk Ltd.

Language

DE — Deutsch

Full Name
Lutz Fechner

E-mail
lutz.fechner@consultingwerk.de

Password

Password Changed Date
year-month-day

Last Login
year-month-day

Manager

Migration and Modernization options

- **Enhance and extend existing applications**
- SmartComponent Library integrates with existing applications and frameworks
- SmartComponent Library functionality can be accessed from existing code
- SmartComponent Library can access existing code, typically of any kind
  - direct execution or through adapters
  - custom service or component implementations
  - Don’t worry about SHARED VARIABLES etc…

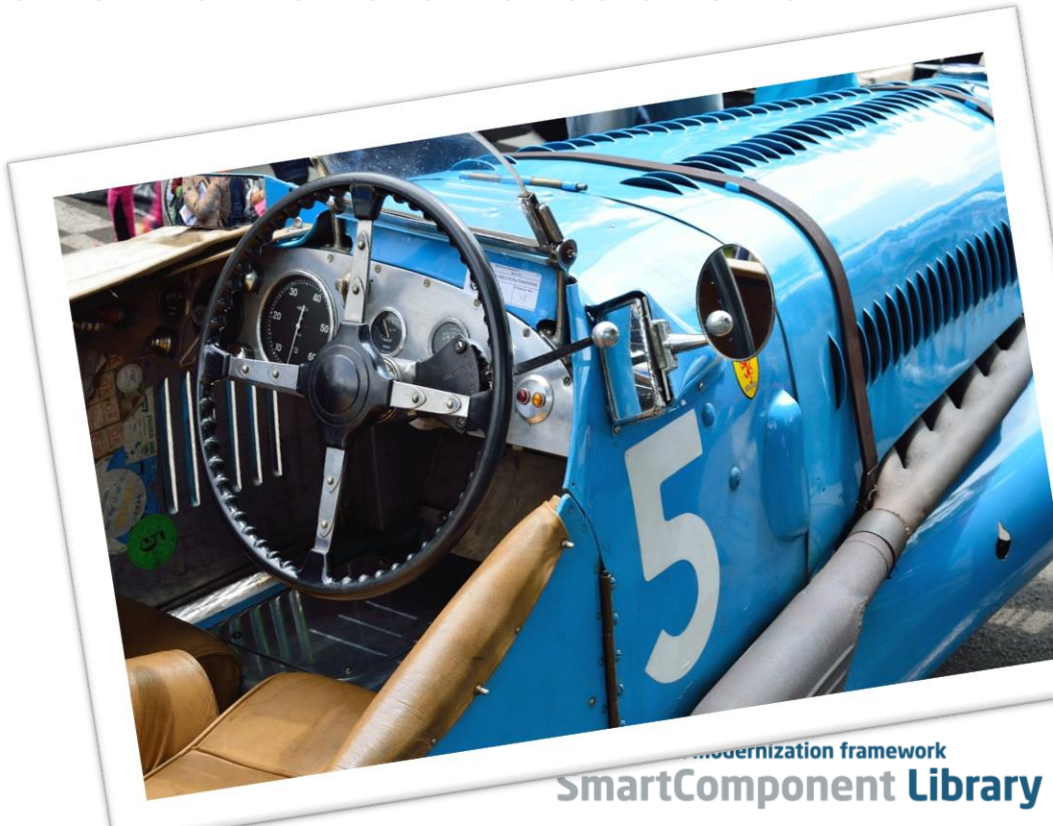
Migration tools

Migration tools

- Analysis of existing source code based on ProParse (TTY, GUI)
- Adaptable to patterns of the existing application
- Discovery of relevant business logic
- Separation of business logic from UI logic
- Combination of business logic from multiple screens into central business logic objects (e.g. Validator)
- Transfer of UI logic into cross-platform APIs
  - UI control, e.g. access to fields, enable, disable, colors
  - User messages, questions
- Various API's supporting code migration

Migration from Legacy-developer frameworks

- Migration tools and compatibility components
- Migration of UI layout (code or repository) and frontend and backend logic
- Successful migration projects for
  - Progress Dynamics
  - ADM1/ADM2
  - DWP (Dynamic WebClient Platform)
  - ProShield (in preparation), close cooperation with original framework developer

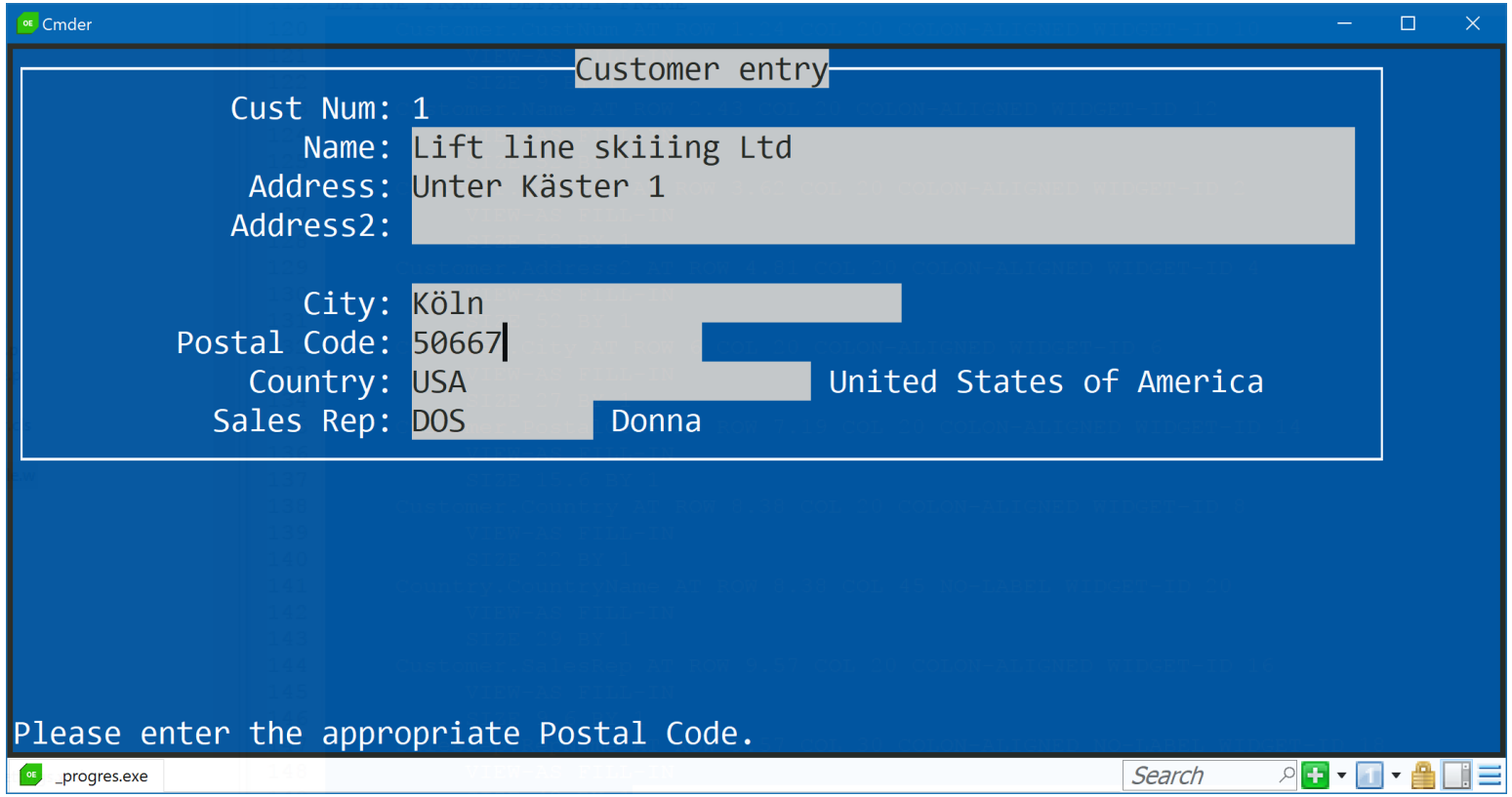


AST as foundation

- Foundation is an „Abstract Syntax Tree“ of the ABL source code
- Independency from coding-style, abbreviated keywords, formatting
- Include Files, nested Include Files, Preprocessor
- Support for GUI, TTY, WebSpeed and GUI for .NET
- Static code analysis, no need to execute ABL code (not walking the widget-tree) – allows for automation
- Knowledge of ABL scopes (variables, buffer, transactions, etc.)
- Knowledge of FRAME phrases

Demo

- Migration of TTY style application with UPDATE EDITING Block



Demo

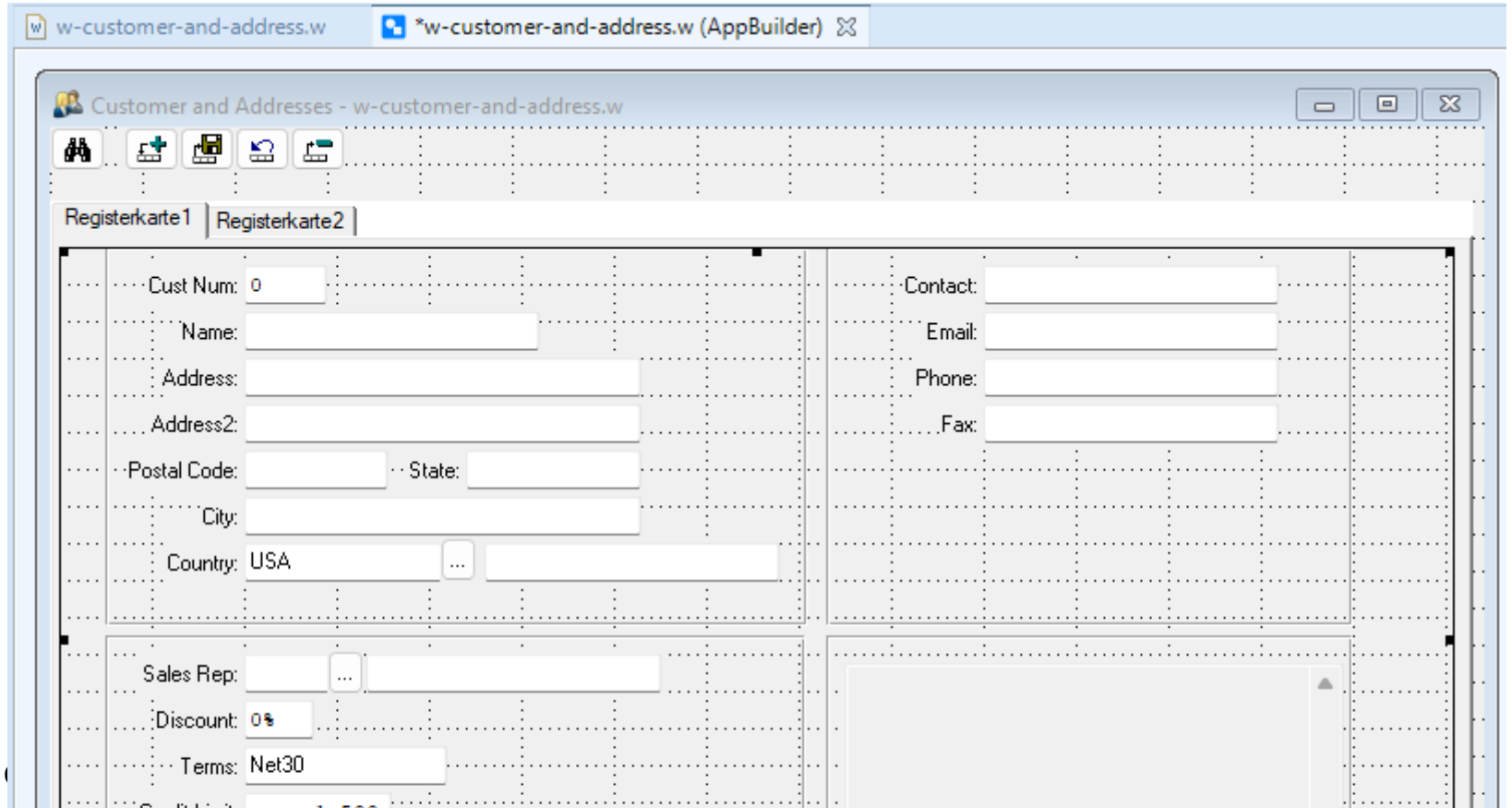
- Generate Business Entity based on TTY screen
- Migrate validation code from UPDATE EDITING
- Implement RESTful annotations

Migration of existing ABL GUI/TTY

- Split complex screens into useful components
  - Frames
  - Tab Pages
  - Grids
- Detection of UI-patterns and replacing to more specialized components
  - Lookup / Zoom-field, etc.
  - From/To Fields, Date/Time-Fields
  - …
- Customizable tooling through ABL based plugins
- Optional adoption into responsible layout

Demo

- Migration of GUI application (AppBuilder, no ADM)



Demo

- Migration of UI components from existing ABL GUI
- Migration of Lookups based on multiple ABL widgets and parsing of trigger code
- Review in Repository Designer

UI Trigger Code Migration

- ABL GUI/TTY Migration to GUI for .NET
- ABL GUI/TTY Migration to TypeScript
- Client Logic API to support ABL type widget attributes and methods in GUI for .NET and Angular
- Migration of UI Trigger Code into server-side event handler (Application Service) allowing cross-platform use

- Server-side event handler preferred solution
- Often mix of client- and server-side event handlers will to be used in combination

Server-side event-handler

- Handling of events such as VALUE-CHANGED or LEAVE on the backend (PASOE)
- Allows reuse of event-handling logic in various UI's
- Allows to maintain tight mingling of data access and UI control logic
  - Setting field values
  - Enabling/disabling fields
  - Viewing/hiding of fields
  - Controlling focus
- ProDataset with screen-values as input-output structure
- Serializable OOABL objects to describe manipulations of fields

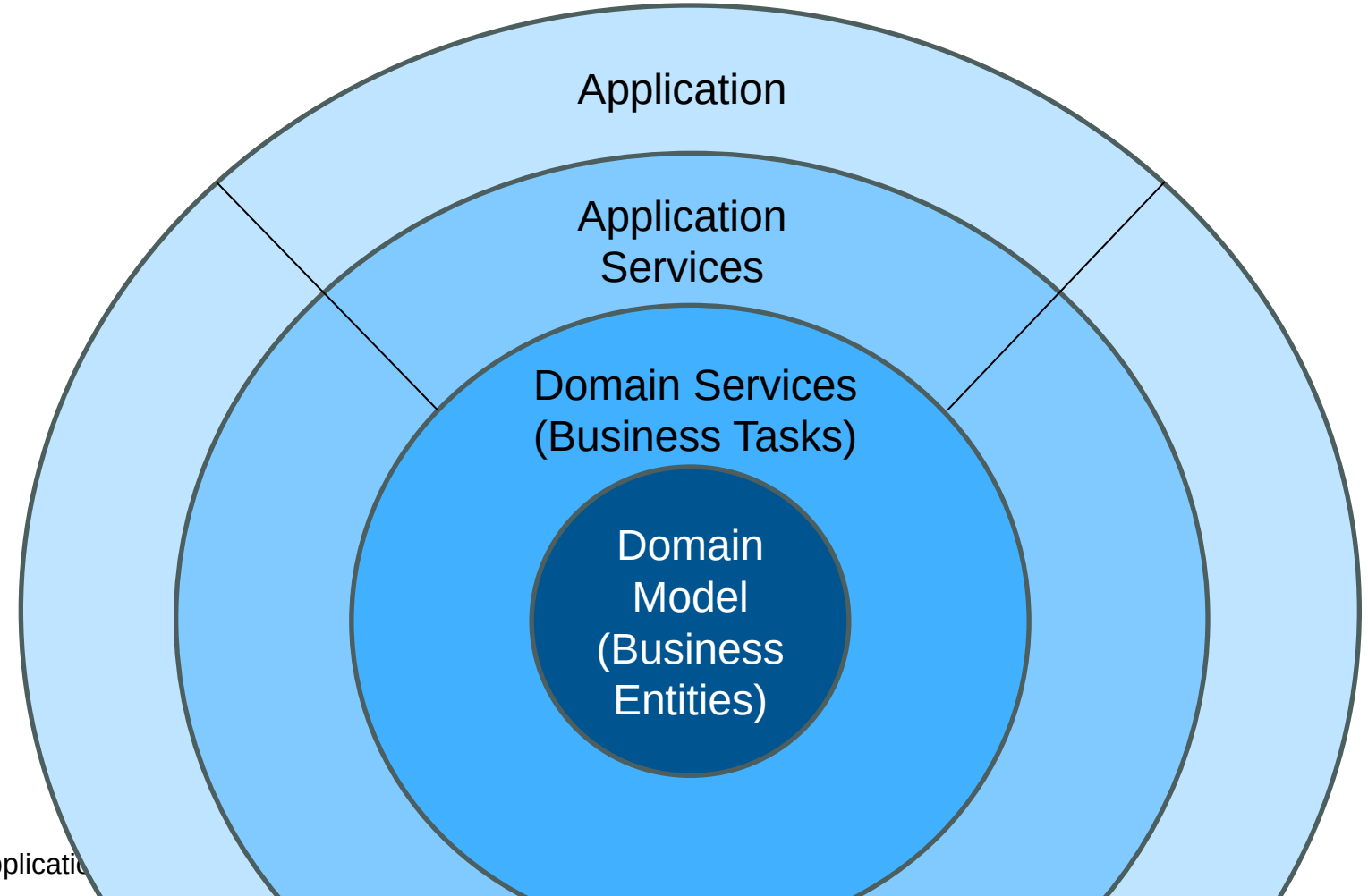
Sample event handler – not pretty, but commonly seen

```
ON VALUE-CHANGED OF Order.CustNum IN FRAME DEFAULT-FRAME /* Cust Num */
DO:
    FIND Customer WHERE Customer.CustNum = INPUT Order.CustNum NO-LOCK NO-ERROR .

    IF AVAILABLE Customer THEN
        ASSIGN Order.SalesRep:SCREEN-VALUE = Customer.SalesRep
               Customer.Name:SCREEN-VALUE  = Customer.Name
               Order.Terms:SCREEN-VALUE    = Customer.Terms.
    ELSE
        ASSIGN Order.SalesRep:SCREEN-VALUE = "":U
               Customer.Name:SCREEN-VALUE  = "":U
               Order.Terms:SCREEN-VALUE    = "":U.

    DISPLAY FILL (STRING(Order.CustNum:INPUT-VALUE) + " ":U, 7) @ Order.Instructions WITH FRAME {&FRAME-NAME}.

    Order.Instructions:BGCOLOR = 4 .
    Order.Instructions:SENSITIVE = FALSE .

    Order.Carrier:VISIBLE = FALSE .

    APPLY "ENTRY":U TO Order.Terms IN FRAME {&FRAME-NAME}.
END.
```

Server-side event handler

- Input:
  - ProDataset with current values of the UI (screen-value)
  - EventHandlerArgument with name of the control and event – to allow reusing the event-handler
- Output:
  - ProDataset with potentially changed values (new screen-value)
  - UiControl instance to control field sensitivity, visibility, style, …
  - UiControl instance with new focus field name

Application services – onion architecture (simplified)



Demo

- Executing server-side event handler from GUI for .NET
- Executing server-side event handler from Angular web application
- Review application service source

ation Services

```
METHOD PUBLIC UiControl HandleCustNumChanged (INPUT-OUTPUT DATASET dsOrder,
                                              poEventArgs AS EventHandlerParameter):

    DEFINE VARIABLE oUiControl AS UiControl    NO-UNDO .

    oUiControl = NEW UiControl() .

    {&_proparse_ prolint-nowarn(findnoerror)}
    FIND FIRST eOrder .

    FIND Customer WHERE Customer.CustNum = eOrder.CustNum NO-LOCK NO-ERROR .

    IF AVAILABLE Customer THEN
        ASSIGN eOrder.SalesRep = Customer.SalesRep
               eOrder.CustName = Customer.Name
               eOrder.Terms    = Customer.Terms.
    ELSE
        ASSIGN eOrder.SalesRep = "":U
               eOrder.CustName = "":U
               eOrder.Terms    = "":U.

    ASSIGN eOrder.Instructions = FILL (STRING (eOrder.CustNum) + " ":U, 7) .

    oUiControl:FieldControl("Instructions":U):Style = "error":U .
    oUiControl:FieldControl("Instructions":U):Sensitive = FALSE .

    oUiControl:FieldControl("Carrier":U):Visible = FALSE .

    oUiControl:FocusFieldName = "Terms":U.

    RETURN oUiControl .
```

Unit Testing of event-handler

- “Event-handler” now much simpler for unit-test
- No dependency on actual user-interface
- No direct dependency on database allows “mocking” of data or data access

```
{Consultingwerk/SmartComponentsDemo/OERA/Sports2000/dsOrder.i}

@Test.
method public void TestMethod():

    define variable oService    as OrderPresentationService no-undo.
    define variable oUiControl as UiControl                no-undo.

    oService = new OrderPresentationService().

    dataset dsOrder:read-xml ("file":u,
                              "Consultingwerk/SmartComponentsDemo/PresentationService/test.xml":u,
                              ?, ?, ?).

    find first eOrder.
    assign eOrder.CustNum = 1 . /* new screen-value */

    oUiControl = oService:HandleCustNumChanged (dataset dsOrder,
                                                new EventHandlerParameter ("CustNum":u, "ValueChanged":u)).

    find first eOrder.

    Assert:Equals(eOrder.CustName, "Lift Line Skiing":u) .
    Assert:Equals(oUiControl:FieldControl("Instructions":u):Style, "error":u) .

end method.
```

Replacing direct data-access

- Migrated application service initially keeps direct database access
- Practical when starting – as otherwise a lot of Business Entities are typically required in the beginning
- However, not desired in the long run – as we want to leverage reusability of logic in domain level objects (Business Tasks and Business Entities)
- Replacing FIND, FOR EACH, … with ORM Mapper
- DatasetModel, the ORM mapper of the SmartComponent Library, simplifying access to Business Entities

Implementing data-access through ORM

```
METHOD PUBLIC UiControl HandleCustNumChanged (INPUT-OUTPUT DATASET dsOrder,
                                              poEventArgs AS EventHandlerParameter):

    DEFINE VARIABLE oUiControl AS UiControl    NO-UNDO .

    /* Database Buffer Definitions */
    DEFINE VARIABLE oCustomer AS CustomerDatasetModel NO-UNDO.
    oCustomer = NEW CustomerDatasetModel () .

    oUiControl = NEW UiControl() .

    {&_proparse_ prolint-nowarn(findnoerror)}
    FIND FIRST eOrder .

    oCustomer:Customer:Filter:CustNum:EQ(eOrder.CustNum):Run () .

    IF oCustomer:Customer:Available THEN
        ASSIGN eOrder.SalesRep = oCustomer:Customer:SalesRep
               eOrder.CustName = oCustomer:Customer:Name
               eOrder.Terms    = oCustomer:Customer:Terms.
    ELSE
```

Alternative to usage of standard DB buffer

Questions

- Email us at info@consultingwerk.com

Next events with Consultingwerk

- PUG Baltic Conference, May 22nd, 2024
- PUG Germany Workshop, using VS-Code for OpenEdge development, June 6th, 2024
- PUG UK & Ireland Summer conference, June 12th/13th, 2024
- Prediction game for the UEFA EURO 2024
- Further webinars planned during the summer
- PUG Challenge Prague, September 18th-20th, 2024
- PUG Challenge Boston, September 29th-October 2nd, 2024