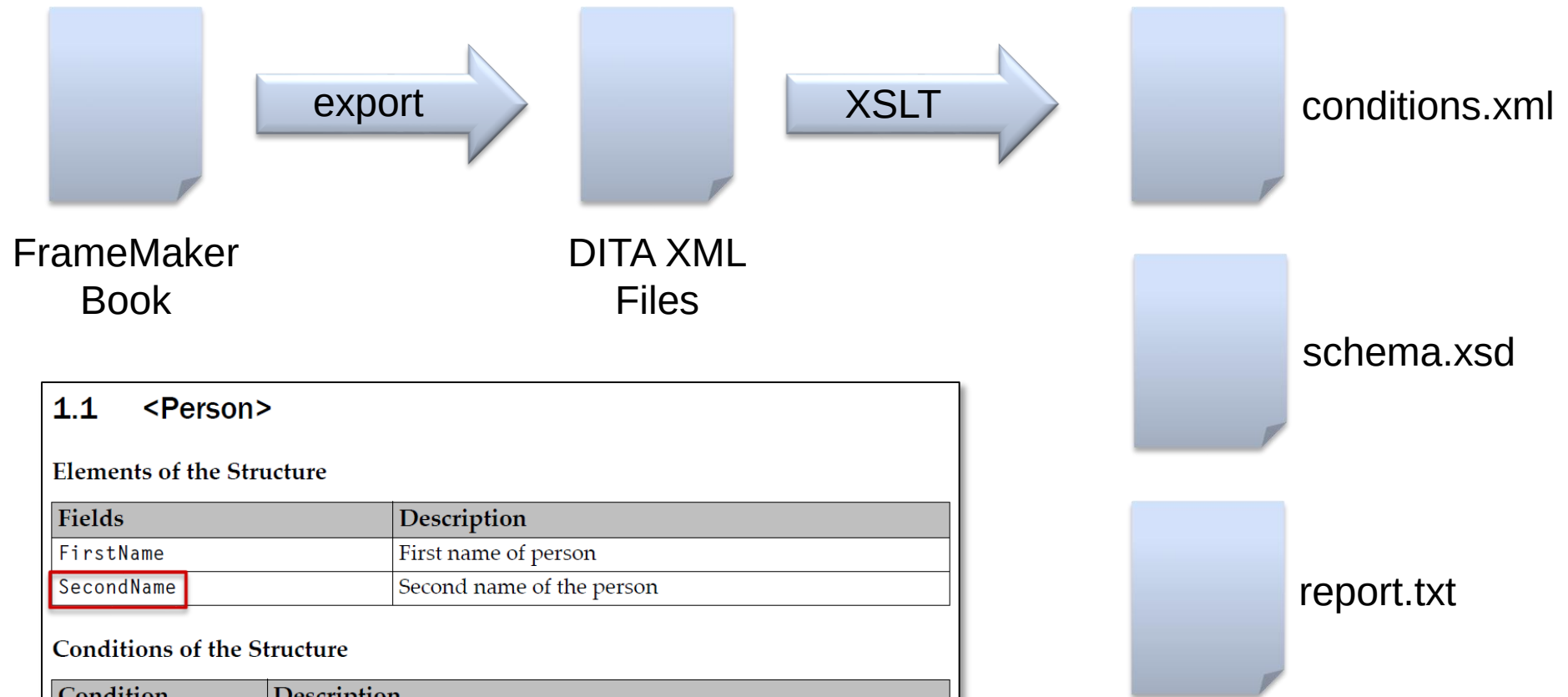


Define and Conquer

Using Semantic XML for Functional Software Specifications

7 Years ago...



1.1 <Person>

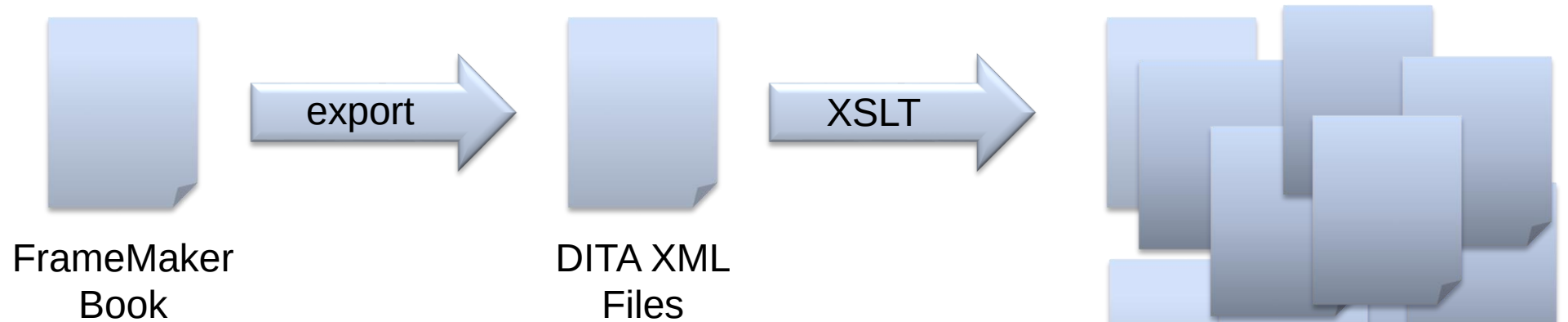
Elements of the Structure

Fields	Description
FirstName	First name of person
SecondName	Second name of the person

Conditions of the Structure

Condition	Description
Person.001	The fields FirstName needs to contain a value according to its definition.
Person.002	The fields FirstName and SecondName must not be empty.

6 Years ago...



1.1 <Person>

Elements of the Structure

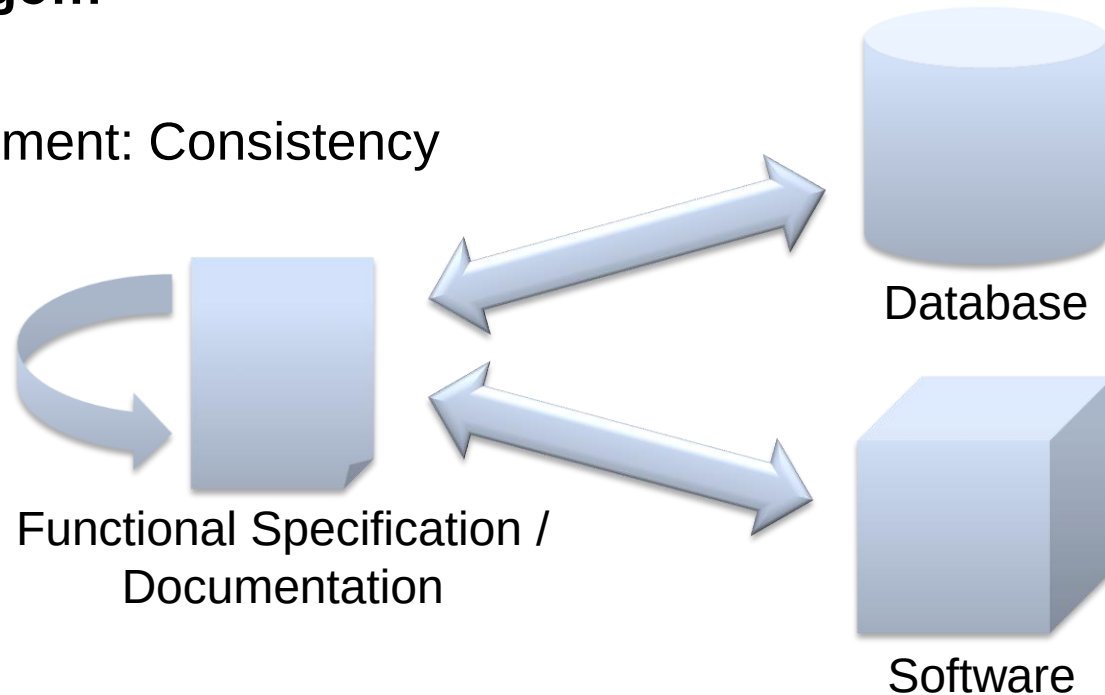
Fields	Description
FirstName	First name of person
SecondName	Second name of the person

Conditions of the Structure

Condition	Description
Person.001	The fields FirstName needs to contain a value according to its definition.
Person.002	The fields FirstName and SecondName must not be empty.

3 Years ago...

➤ Achievement: Consistency



➤ Problems

- Workflow and runtime
- Authoring assistance
- Interpreting the XML code

3 Years ago...

➤ Solutions:

- Native XML editor: oXygen



- Custom oXygen framework
- Highly customized XML schema

“Define and Conquere”

2.4 ESB-Service: Login

Provides functions for login and logoff.

Namespace	ESB-Services/Common
Type	Windows Webservice
DB Access	▪ DBROOT.LOGIN_TOKEN (<i>read/write</i>) ▪ DBROOT.USER (<i>read</i>)
Required Components	<None>

Overview of the Functions

- **Login**

Validates the user and password and returns a login token on success. (Details: see *Function: Login*.)

- **Logoff**

This function causes the passed login token to become invalid. (Details: see *Function: Logoff*.)

2.4.1 Function: Login

Validates the user and password and returns a login token on success.

DB Access	▪ LOGIN_TOKEN (<i>write</i>) ▪ USER (<i>read</i>)
-----------	---

Sample Markup

```

<EsbService id="HD3-5ZQ-Q5">
  <title><Key>Login</Key></title>
  <body>
    <shortdesc id="GF3-5ZQ-Q5">Provides functions for login and logoff.</shortdesc>
    <Header>
      <Namespace>ESB-Services/Common</Namespace>
      <Type/>
      <ListDbAccess/>
      <ListRequiredComponents><None/></ListRequiredComponents>
    </Header>
  </body>
</EsbService>

```

2.4 ESB-Service: Login

Provides functions for login and logoff.

Namespace	ESB-Services/Common
Type	Windows Webservice
DB Access	▪ DBROOT.LOGIN_TOKEN (<i>read/write</i>) ▪ DBROOT.USER (<i>read</i>)
Required Components	<None>

Authoring Assistance

- Mandatory elements
- Schematron rules
- Static content
- Element choices

2.5 ESB-Service: **Key**

shortdesc

Namespace	Namespace
Type	Windows Webservice
DB Access	[XSLT-Conref] <None>
Required Components	ListRequiredComponents ☒ Component ☐ None ☐ SpecialCase

[XSLT-Conref] (Node: No overview required with more than a single function.)

2.5.1 Function: **Key**

shortdesc

DB Access	ListDbAccess
ESB Calls	ListEsbCalls

Authoring Assistance

- Mandatory elements
- Schematron rules
- Static content
- Element choices
- Default content
- Hints
- Author operations

XML Interface

<LoginRequest> (Default)	
 Username	Name of the user.
(String, required)	
 Password	Password of the user.
(String, required)	

Example Request (Default)

Use "xxx" as placeholder for confidential data.

Create from definition

xml-codeblock

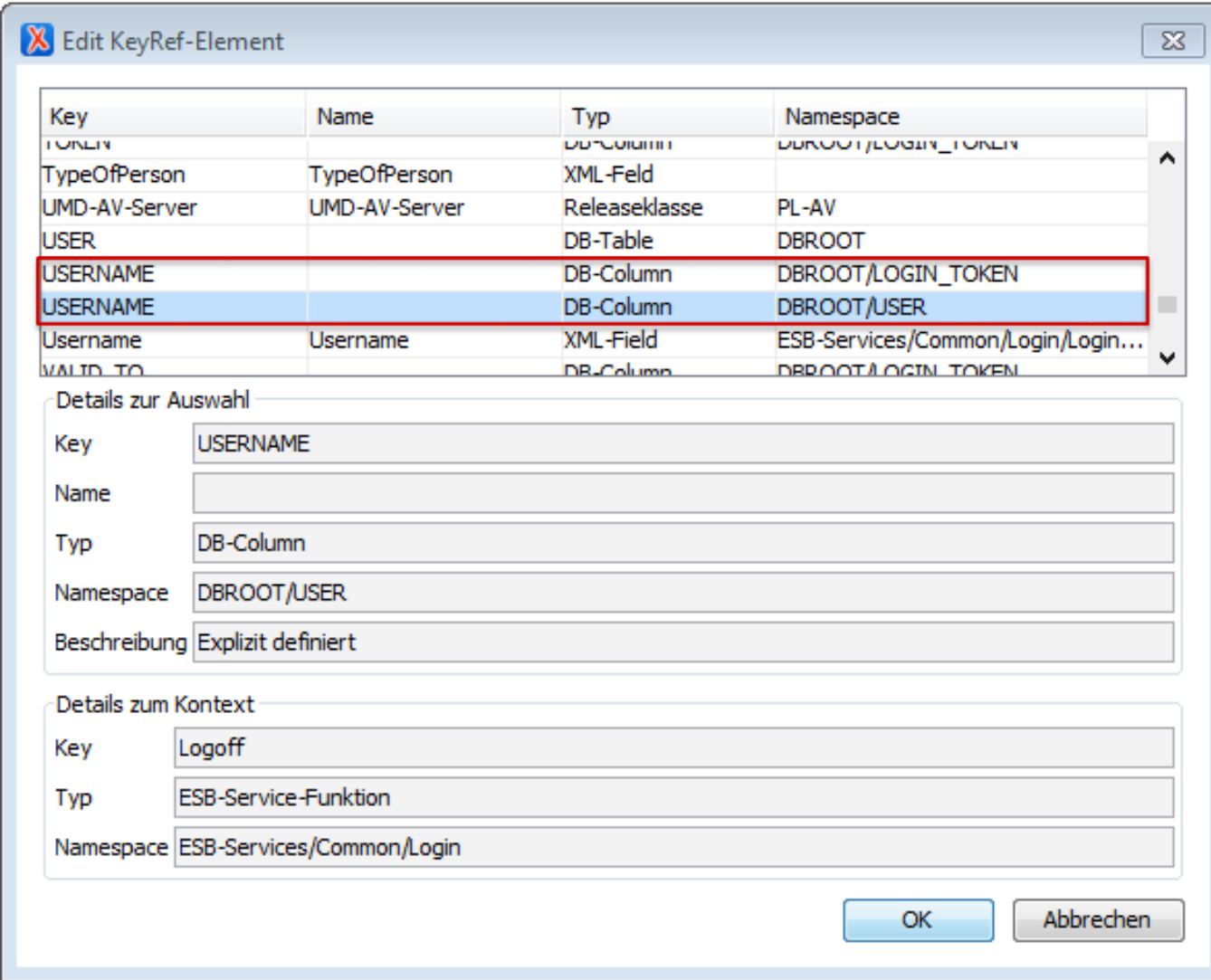
Referencing Identifiers

2.4.1.2 Workflow

The function checks if  `Username` is valid according to  `USER.USERNAME` and  `Password` matches with  `USER.PASSWORD`. If this is the case a new login token is generated and stored in a new dataset in  `LOGIN_TOKEN`.  `LOGIN_TOKEN.VALID_TO` is computed according to  `USER.LOGIN_PERIOD`.

-  `Key`  `(???)`  `<KeyRef/>`
-  `USRNAME`  `(???)`  `<KeyRef>USRNAME</KeyRef>`
-  `USERNAME`  `(###)`  `<KeyRef KeyType="DB-Column">USERNAME</KeyRef>`
-  `USER.USERNAME`  `*` `<KeyRef KeyType="DB-Column" Namespace="DBROOT/USER">USER.USERNAME</KeyRef>`

Referencing Identifiers

The dialog box is titled "Edit KeyRef-Element" and contains a table of key references, a "Details zur Auswahl" section, and a "Details zum Kontext" section. The table lists various identifiers with their names, types, and namespaces. Two rows are highlighted with a red border: "USERNAME" (DB-Column, DBROOT/LOGIN_TOKEN) and "USERNAME" (DB-Column, DBROOT/USER). The "Details zur Auswahl" section shows the selected key as "USERNAME" and its namespace as "DBROOT/USER". The "Details zum Kontext" section shows the context key as "Logoff" and its namespace as "ESB-Services/Common/Login".

Key	Name	Typ	Namespace
LOGIN_TOKEN	TypeOfPerson	DB-Column	DBROOT/LOGIN_TOKEN
TypeOfPerson	TypeOfPerson	XML-Feld	
UMD-AV-Server	UMD-AV-Server	Releaseklasse	PL-AV
USER		DB-Table	DBROOT
USERNAME		DB-Column	DBROOT/LOGIN_TOKEN
USERNAME		DB-Column	DBROOT/USER
Username	Username	XML-Feld	ESB-Services/Common/Login/Login...
VALID TO		DB-Column	DBROOT/LOGIN_TOKEN

Details zur Auswahl

Key: USERNAME

Name:

Typ: DB-Column

Namespace: DBROOT/USER

Beschreibung: Explizit definiert

Details zum Kontext

Key: Logoff

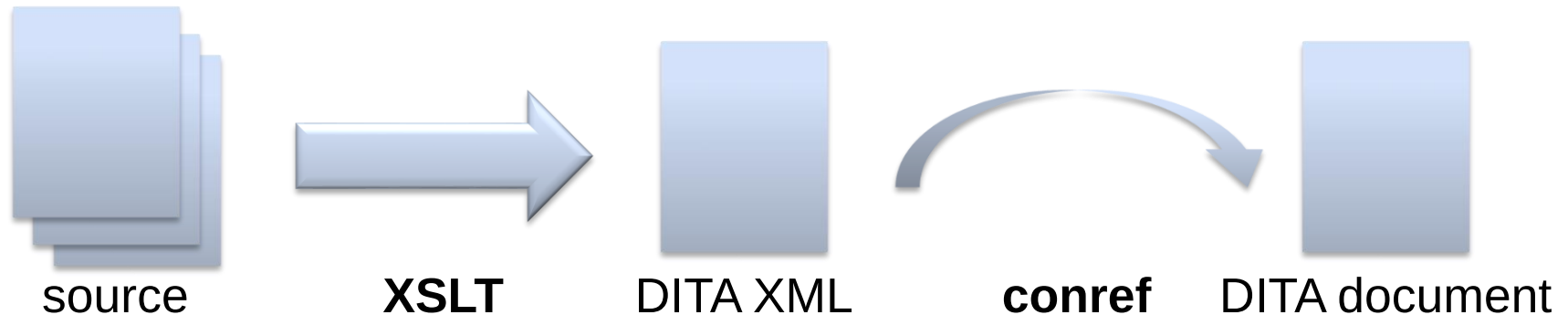
Typ: ESB-Service-Funktion

Namespace: ESB-Services/Common/Login

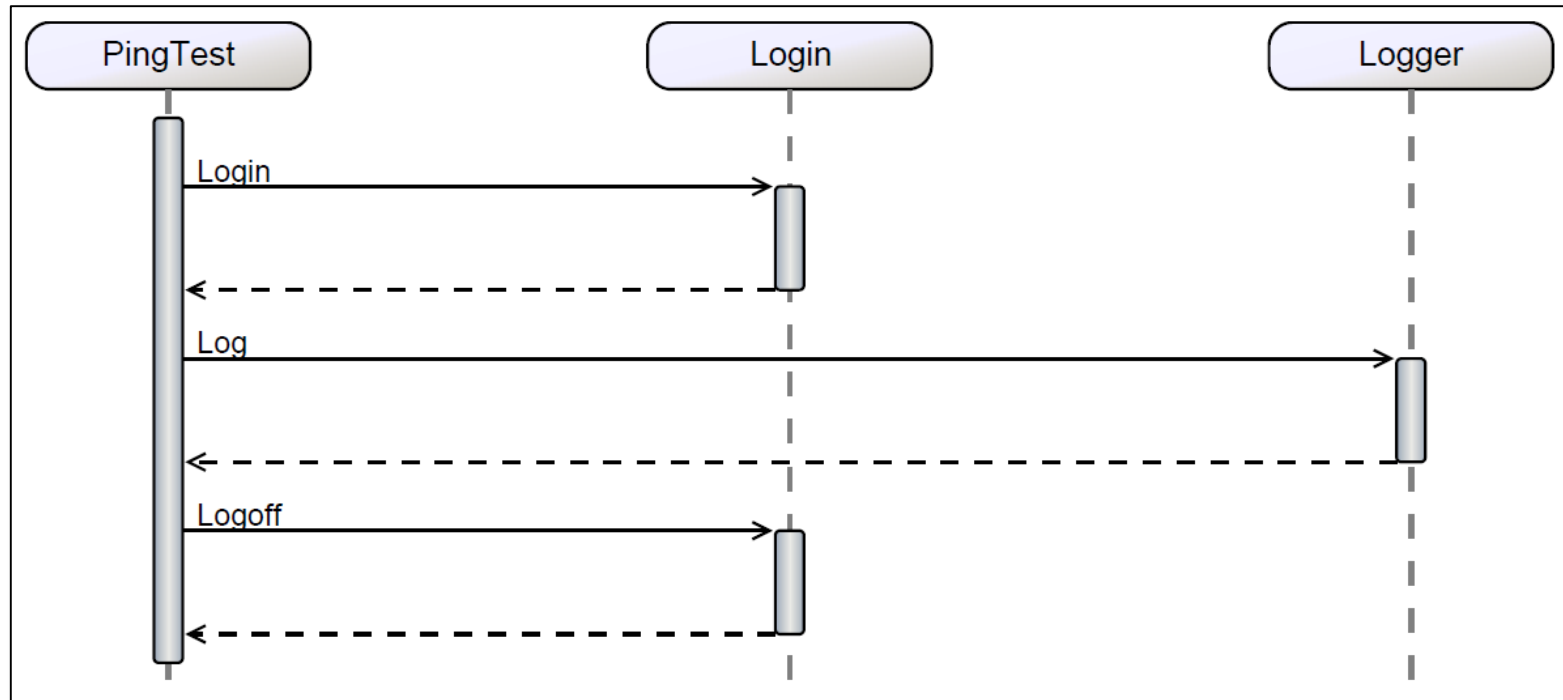
OK Abbrechen

Redundancy

Namespace	ESB-Services/Common
Type	Windows Webservice
DB Access	[XSLT-Conref]      •  DBROOT.LOGIN_TOKEN (read/write)* •  DBROOT.USER (read)*
Required Components	▷<None>◁



Redundancy



Defining the Framework

- Use attribute defaults

 **2.4.1 Function:**   Login  

```
<title
  class          = "+ topic/title ESB-Service-Function-d/title "
  cba:prefix     = "Function: ">
  <Key
    class          = "+ topic/ph ESB-Service-Function-d/Key "
    ikd:type       = "ESB-Service-Function"
    ikd:root       = "parent::title/parent::*"
    ikd:ancestor-namespace = "component"
    ikd:desc       = "body/shortdesc">Login</Key>
  </title>
```

Open Source

- Refactoring custom framework
- Joint development with parson AG
 - Expert for technical communication as partner
 - github.com/dita-semia (SEMantic Information Architecture)
- Plugins for oXygen and DITA-OT

Conclusion

Highly specialized semantic XML

- Authoring assistance
- Automated processing
- Successfully applied
- Great potential
- Not limited to DITA and oXygen

Define and Conquer

Using Semantic XML for Functional Software Specifications

Questions?