



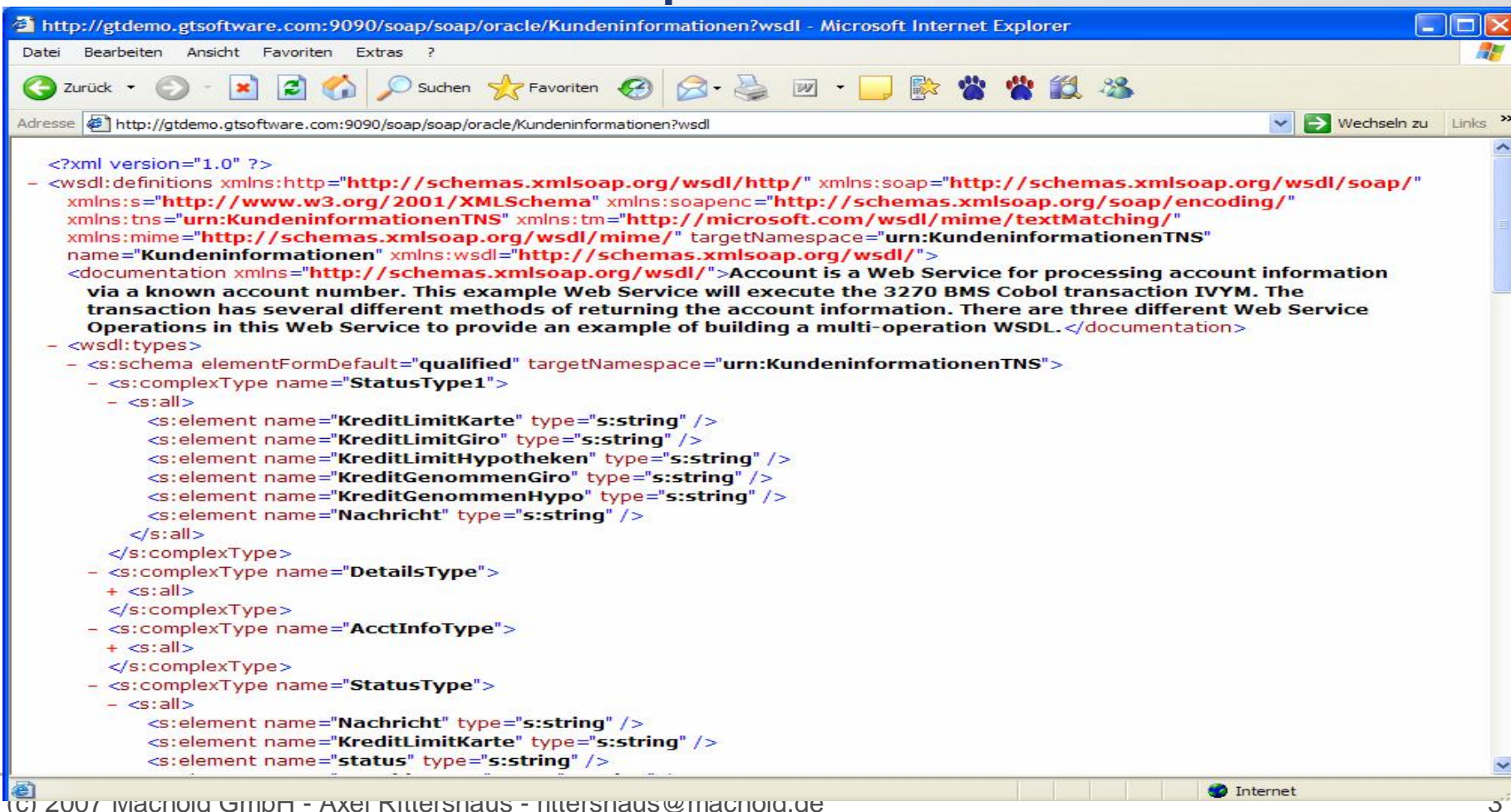
V7 – IMS-integration with web services....

Including LIVE demonstration using Ivory and a real mainframe

Axel Rittershaus, Machold Systemhaus 21, Stuttgart

IMS Technical Symposium 2007 – 23. – 26. April 2007

Web Service – WSDL example

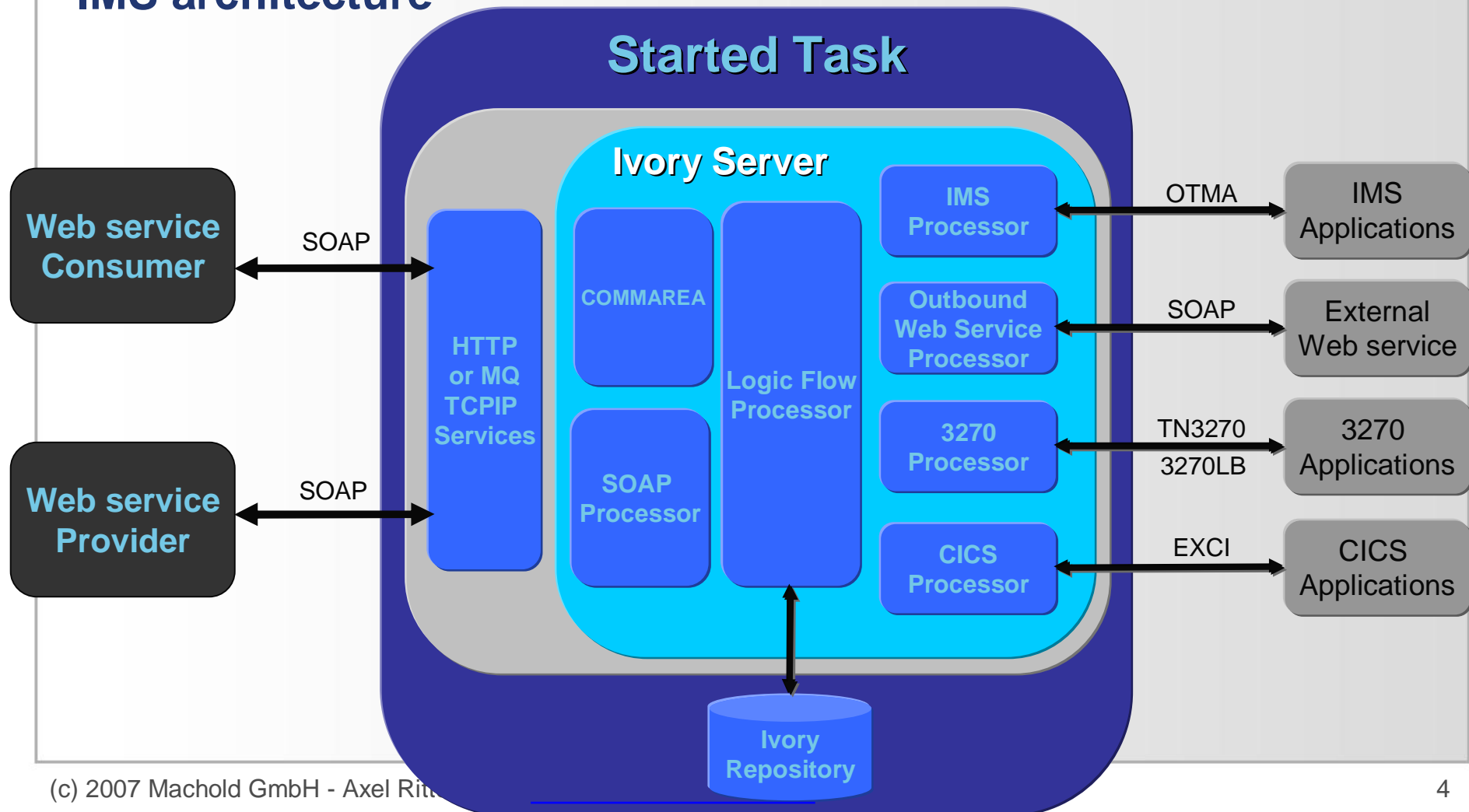


```

<?xml version="1.0" ?>
- <wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="urn:KundeninformationenTNS" xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" targetNamespace="urn:KundeninformationenTNS"
  name="Kundeninformationen" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Account is a Web Service for processing account information
    via a known account number. This example Web Service will execute the 3270 BMS Cobol transaction IVYM. The
    transaction has several different methods of returning the account information. There are three different Web Service
    Operations in this Web Service to provide an example of building a multi-operation WSDL.</documentation>
- <wsdl:types>
  - <s:schema elementFormDefault="qualified" targetNamespace="urn:KundeninformationenTNS">
    - <s:complexType name="StatusType1">
      - <s:all>
        <s:element name="KreditLimitKarte" type="s:string" />
        <s:element name="KreditLimitGiro" type="s:string" />
        <s:element name="KreditLimitHypotheiken" type="s:string" />
        <s:element name="KreditGenommenGiro" type="s:string" />
        <s:element name="KreditGenommenHypo" type="s:string" />
        <s:element name="Nachricht" type="s:string" />
      </s:all>
    </s:complexType>
  - <s:complexType name="DetailsType">
    + <s:all>
    </s:complexType>
  - <s:complexType name="AcctInfoType">
    + <s:all>
    </s:complexType>
  - <s:complexType name="StatusType">
    - <s:all>
      <s:element name="Nachricht" type="s:string" />
      <s:element name="KreditLimitKarte" type="s:string" />
      <s:element name="status" type="s:string" />
    </s:all>
  </s:schema>
</wsdl:types>

```

IMS architecture



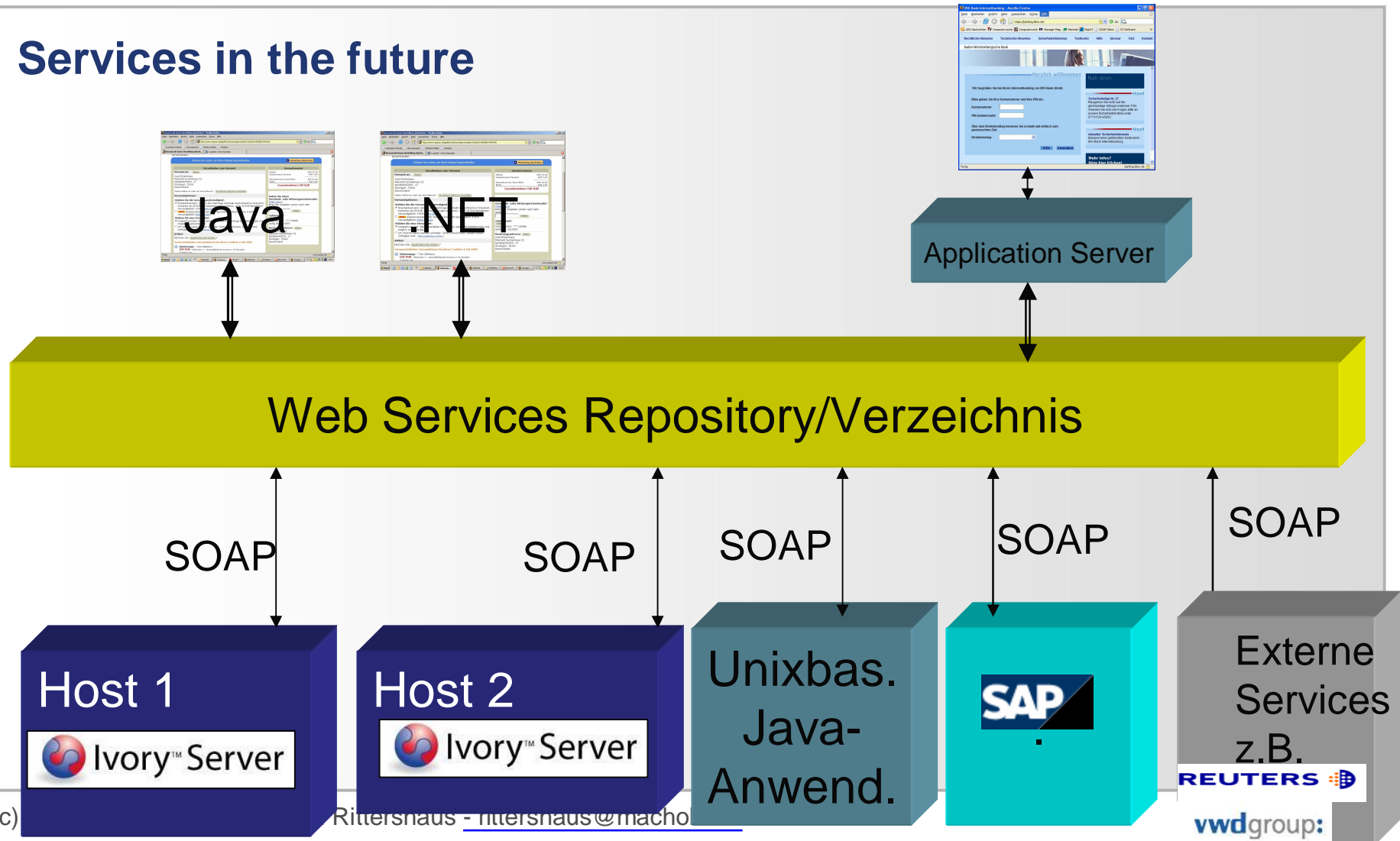
Ivory

- ➔ Delivers existing mainframe functionality (programs, transactions, data) to new technology worlds like .NET, J2EE, ...
- ➔ NO changes in existing (mainframe-) applications required
- ➔ Graphical IDE (Ivory Studio) and mainframe based server component
- ➔ Create one new function as a Web service using multiple mainframe transactions without coding
- ➔ Import wizard for Copybooks, Includes, BMS maps, ...
- ➔ Enables the mainframe to call external Web Services (i.e. COBOL or PL/1 program calls Web Service and processes the result) – for COBOL and PL/1 available!

Ivory

- ➔ NO additional hardware or software required
- ➔ NO changes in existing applications
- ➔ NO coding
- ➔ NO compiling or code generation – just orchestration
- ➔ NO need to learn Java, .NET, XML, WSDL
-> every mainframe programmer (!) can use it

Services in the future





Powerpoint is the best plattform for each kind of applications - but does it really work in the real world ???

LIVE DEMO

Using Ivory and a mainframe based at GT
Software, Atlanta, GA, USA

Screenshots from the LIVE Demo (1) Service orchestration

MACHOLD
Systemhaus 21

GT Software Ivory Studio - heikesWebService (Web Service)

File Edit View Tools Window Help

Toolbox

- All
- 3270
- Start
- Web Service Operation
- Callable Service Operation
- 3270 Process
- 3270 Point Node
- Move to 3270
- 3270 Action
- Move to Output
- Move to Work Variable
- Connector
- Decision
- Switch
- Case
- Function
- Loop
- Operation End
- Sticky Note

Diagram - heikesWebService

Created by Ivory Studio Project Wizard.
User Name: ARittershaus
Date: Mittwoch, 25. April 2007
Time: 12:30:51

```

graph TD
    Start([Start]) --> WSOp1[Web Service Operation 1]
    Start --> WSOp2[Web Service Operation]
    WSOp1 --> IMS1{{IMS Point Node 1}}
    WSOp2 --> getData[get data and stock data]
    IMS1 --> MoveIMS1[Move to IMS]
    MoveIMS1 --> IMSMov1[IMS Movement 1]
    IMSMov1 --> ExecIMS1([Execute IMS 1])
    ExecIMS1 --> MoveOut1[Move to Output]
    MoveOut1 --> OutMov1[Output Movement 1]
    OutMov1 --> OpEnd1{{Operation End 1}}
    getData --> IMS2{{IMS Point Node 2}}
    IMS2 --> MoveIMS2[Move to IMS]
    MoveIMS2 --> IMSMov2[IMS Movement 2]
    IMSMov2 --> ExecIMS2([Execute IMS 2])
    ExecIMS2 --> WSClient{{Web Service Client Point Node}}
    WSClient --> stockquote[stockquote/WebService]
    stockquote --> MoveWSC[Move to Web Service Client]
    MoveWSC --> WSCMov1[Web Service Client Movement 1]
    WSCMov1 --> ExecWSC1([Execute Web Service Client 1])
    ExecWSC1 --> MoveOut2[Move to Output]
    MoveOut2 --> OutMov2[Output Movement 2]
    OutMov2 --> OpEnd2{{Operation End 2}}
  
```

Properties

Base	
Node ID	Identify the Web service
Web Service Name	heikesWebService
URI	/soap/machold/heikesWebService
WSDL Location	/soap/machold/heikesWebService?WSDL
Encoding Style	document/literal wrapped

Node ID
The unique identifier for this Start node.

Ready

Screenshots from the LIVE Demo (2) Integrated test tool

MACHOLD
Systemhaus 21

The screenshot displays the GT Software Ivory Studio interface for testing a web service. The main window is titled "GT Software Ivory Studio - heikesWebService (Web Service)".

Toolbox: Contains various test elements such as Start, Web Service Operation, Callable Service Operation, 3270 Process, 3270 Point Node, Move to 3270, 3270 Action, Move to Output, Move to Work Variable, Connector, Decision, Switch, Case, Function, Loop, Operation End, and Sticky Note.

Diagram - heikesWebService Test-heikesWebService.wsdl:

- Settings:**
 - WSDL Location: ory_Projects\ar\JMS\heikesWebService.wsdl
 - Service Name: heikesWebService
 - Operation: getData
 - HTTP Version: HTTP 1.1
 - URI: /soap/machold/soap/machold/heikesWebService
 - SOAP Action: um:getData
 - Proxy URL: (empty)
 - Timeout (secs): 0
 - User ID: (empty)
 - Password: (empty)
- SOAP Request:**

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="h
<soap:Body>
  <s0:getData>
    <s0:Nachname>smith</s0:Nachname>
    <s0:Vorname>mary</s0:Vorname>
  </s0:getData>
</soap:Body>
</soap:Envelope>
```
- SOAP Response:**

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="h
<soap:Body>
  <getDataResponse xmlns="urn:heikesWebServiceTNS">
    <outDataType>
      <outLastName>SMITH</outLastName>
      <outFirstName>MARY</outFirstName>
      <outExtension>1234</outExtension>
      <outZipCode>20345</outZipCode>
    </outDataType>
```

Properties:

Base	Identify the Web service
Node ID	
Web Service Name	heikesWebService
URI	/soap/machold/heikesWebService
WSDL Location	/soap/machold/heikesWebService?WSDL
Encoding Style	document/literal wrapped

Node ID: The unique identifier for this Start node.

Output: Ready

At a glance

- ➔ High productivity - no coding
- ➔ Mainframe extends his functionality - no need for additional middleware
- ➔ Reuse existing business logic in an intelligent way
- ➔ Existing - and proven - applications will not be changed
- ➔ CICS and IMS 100% supported
- ➔ No training required, mainframe developers will use it within 2 days

- ➔ fast time to market
- ➔ efficient
- ➔ cost effective
- ➔ secure

Ivory supports and stabilises the useage of mainframes in the future and SOA-enables him

Kontakt

EDV-Beratung Machold GmbH
Systemhaus 21
Nordbahnhofstr. 17
70191 Stuttgart
Germany
www.machold.de
www.rapid-soa.de

Axel Rittershaus
Tel.: +49 (711) 2 57 72-11
Fax: +49 (711) 2 57 72-22
Mobil: +49 (160) 93 860 118

E-mail: axel.rittershaus@machold.de