university of groningen

29-11-2011

Testing for Highly Distributed Service-Oriented Systems using Virtual Environments

- Faris Nizamic, PhD student at Distributed Systems research group
- Rix Groenboom, Parasoft Netherlands B.V.
- Alexander Lazovik, University of Groningen

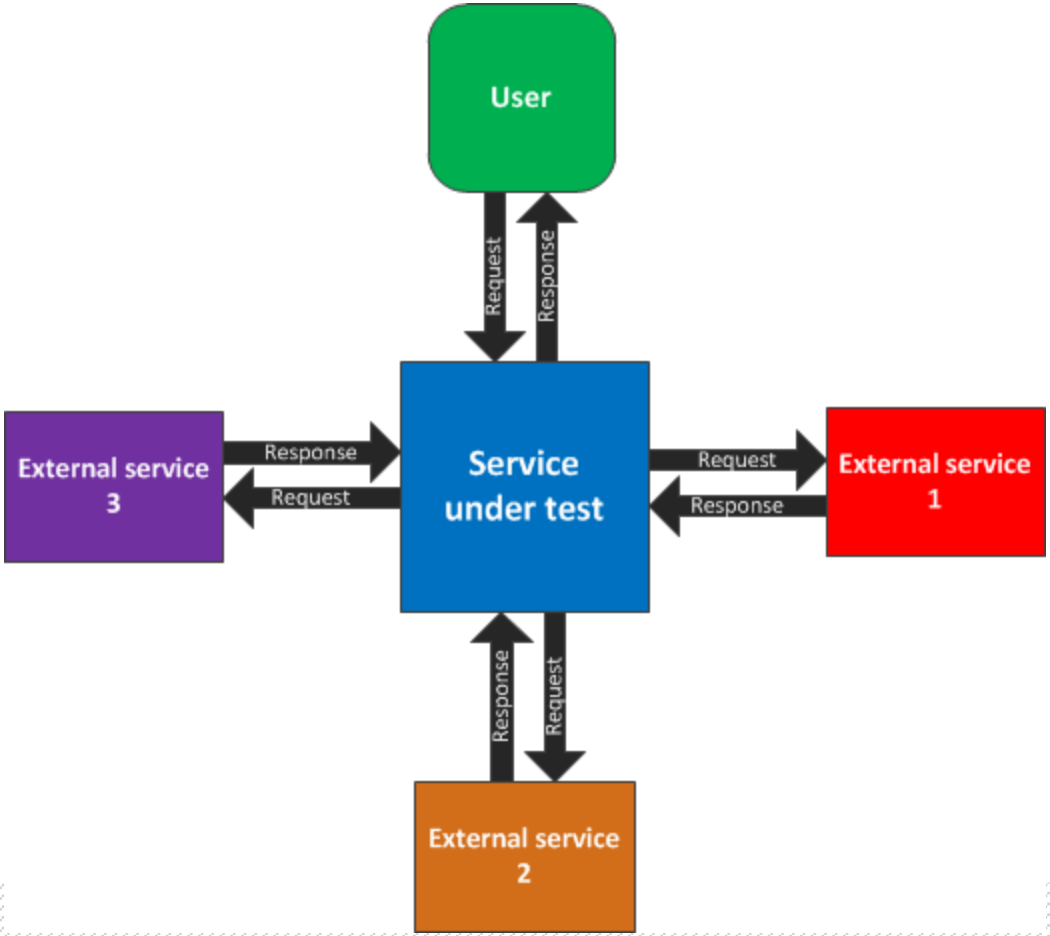
17th Dutch Testing Day
University of Twente, Enschede

Smart testing?

- On March 11, 2009, twenty five residents of **Kaag en Braassem** in the Netherlands unexpectedly received a certificate that they were **married**, **died** or had **registered a newborn**
- What happened?

Highly Distributed Service-Oriented Systems

- Service Oriented Architecture
- Business functionalities
- **Loosely-coupled services**
- Uniform communication
- **No single ownership**



The facts

- Software is continuously evolving
- Functionality is spread across the network
- However, testing software implicitly assumes full control

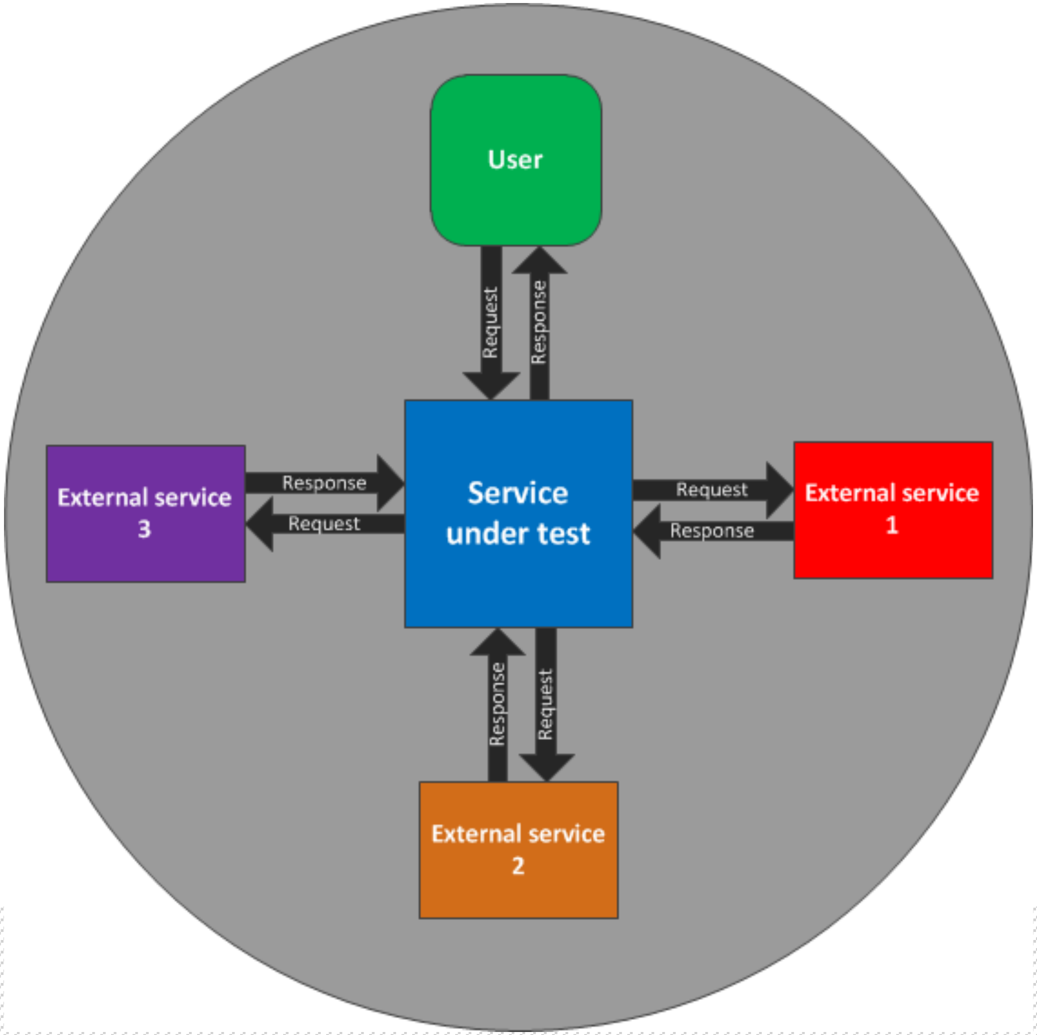
Problem statement

- Tester's perspective
  - Fighting for a shared resources
  - Dependent on deployment team
  - Testing needs to be more agile
  - Availability of external services
  - No control over data - hard to produce desired behavior of external services

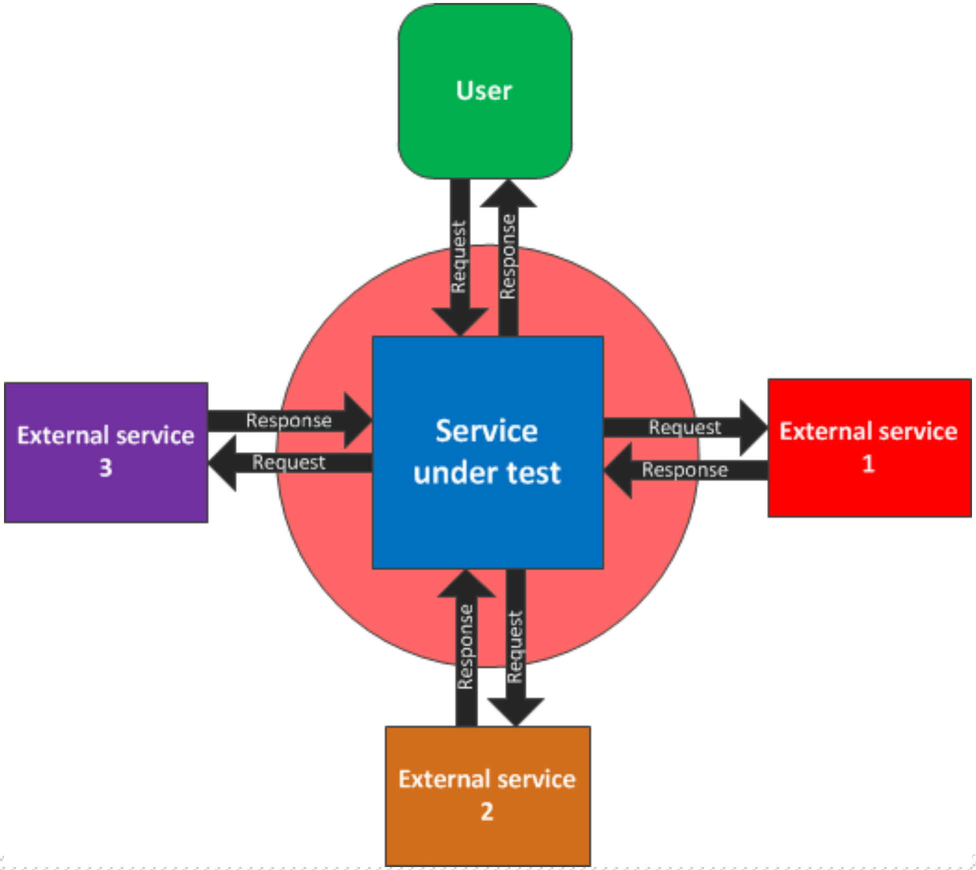
Problem statement

- Companies’ perspective
  - Tests that interfere with the real world
  - Costs of having multiple testing environments
  - Effort of maintenance of testing environments
- Infrastructure is new bottleneck for competitive advantage

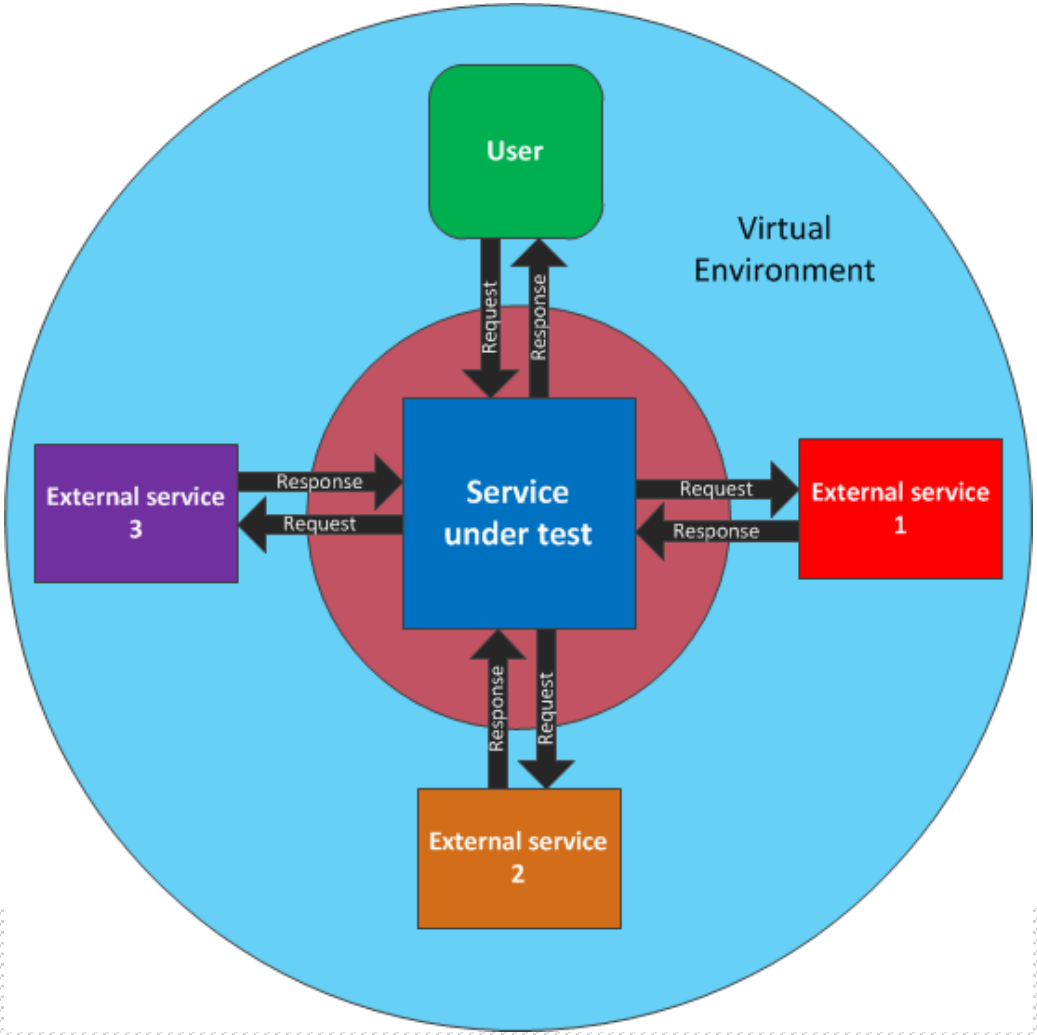
Testing and full control



Testing today

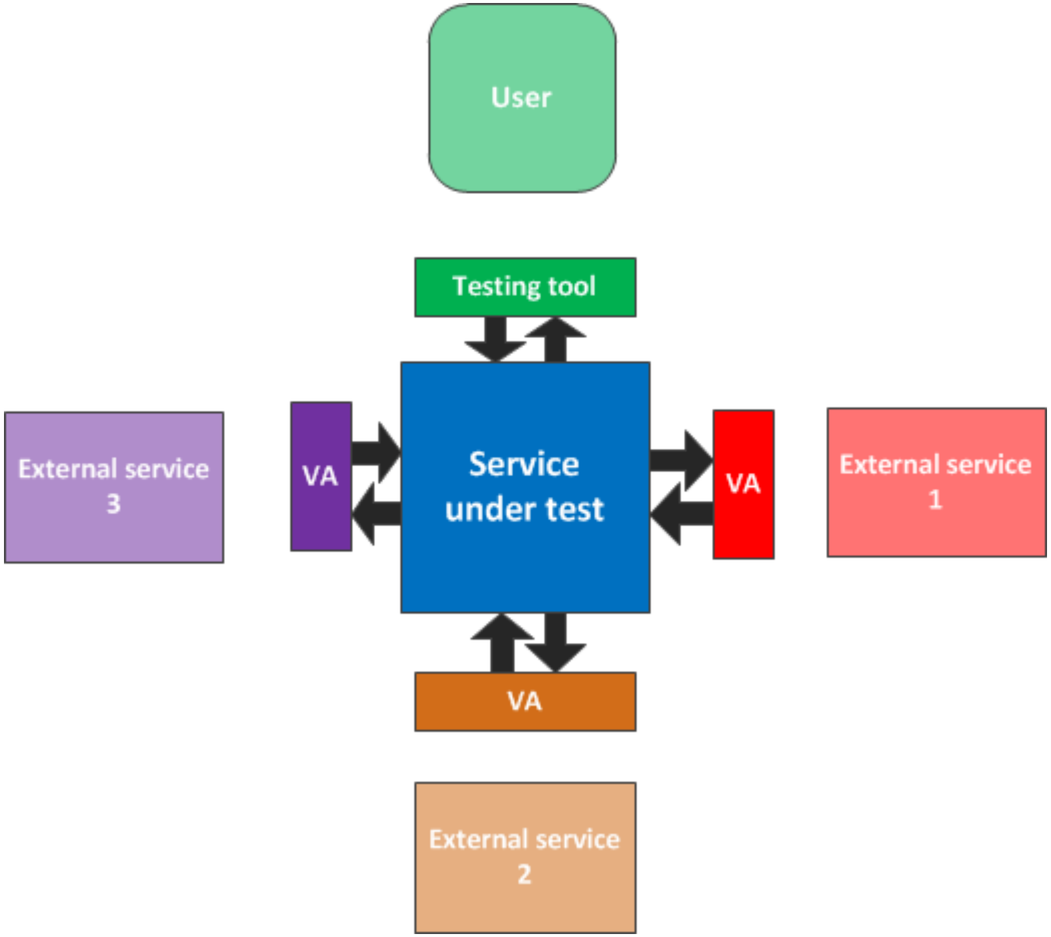


Testing tomorrow



Solution

- Simulate external services with virtual assets (VA)
- Model behavior of external systems
- Gain control over external systems and their data



Still a problem?

› Tester’s perspective

- Fighting for a shared resources ✗
- Dependent on deployment team ✗
- Testing needs to be more agile ✗
- Availability of external services ✗
- No control over data ✗

› Companies’ perspective

- Tests that interfere with the real world ✗
- Costs of having multiple testing environments ✗
- Effort of maintenance of testing environments ✗

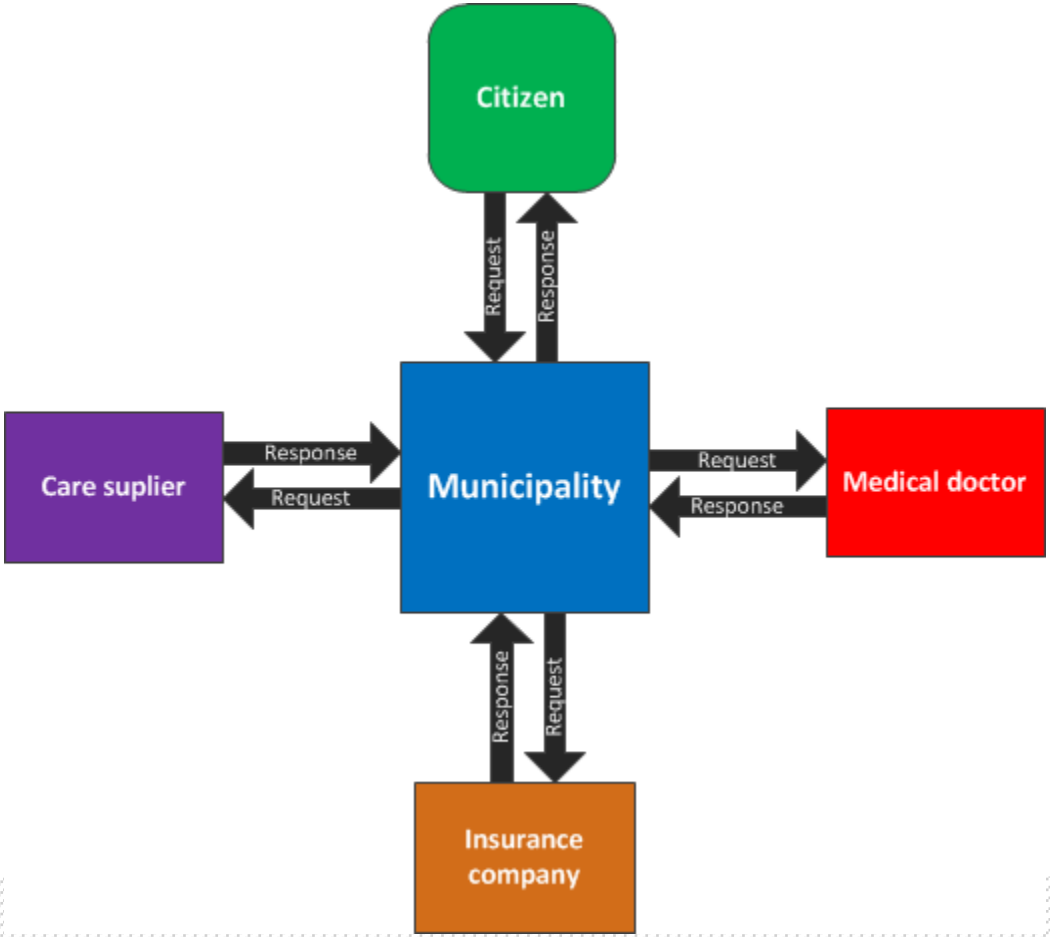
Case study

- WMO (Wet Maatschappelijke Ondersteuning)
    - Dutch law for supporting people that have a chronic disease or disability

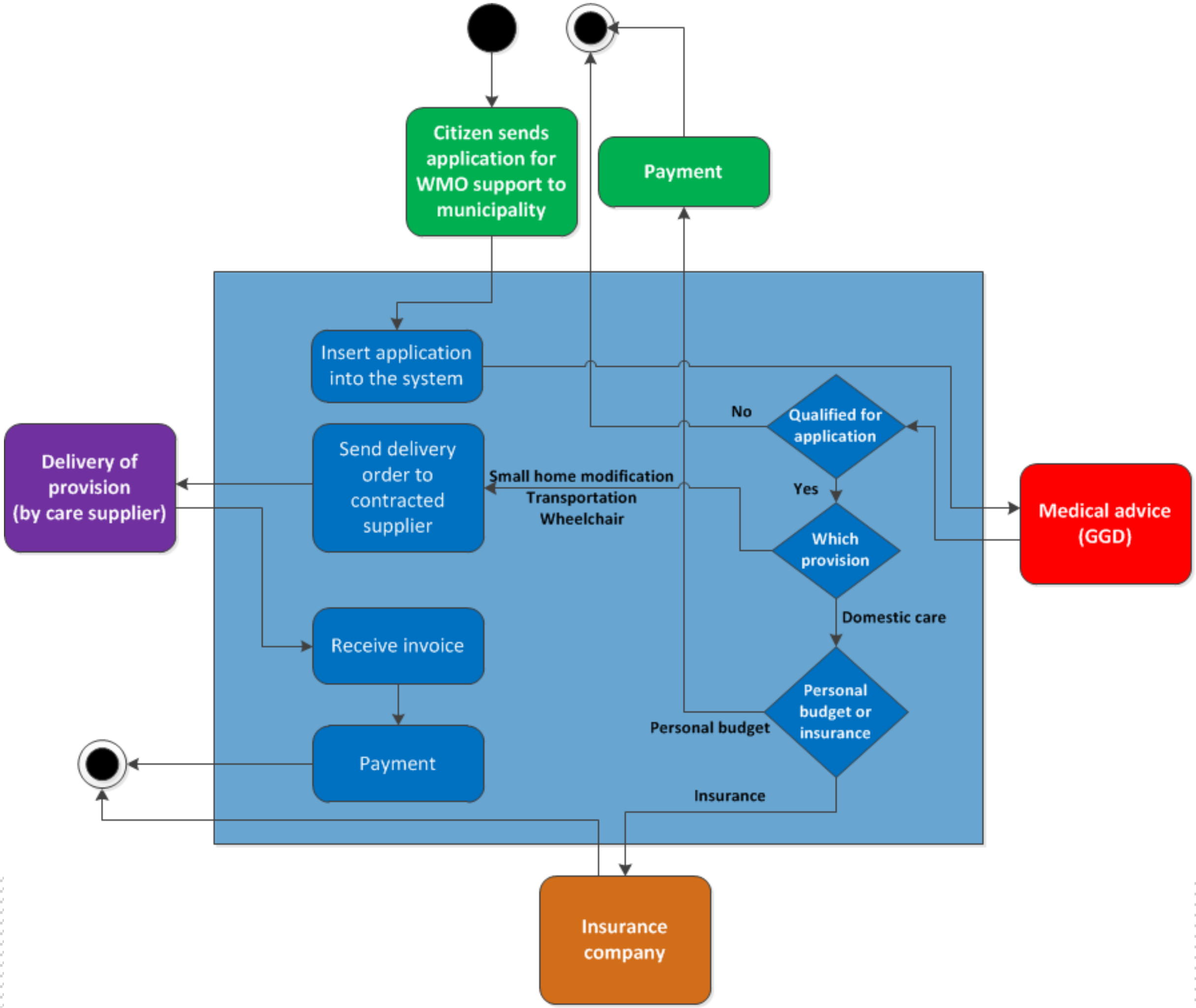
- The responsibility for a WMO request lies with the Dutch municipalities, they handle the complete business process.

- Support provided by the WMO:
    - transportation (e.g. taxi)
    - wheelchair
    - home modification (e.g. “traplift”)
    - domestic care

WMO services



WMO business process (simplified main flow)


Citizen sends application for WMO support to municipality
Payment
Insert application into the system
No
Qualified for application
Yes
Delivery of provision (by care supplier)
Send delivery order to contracted supplier
Small home modification
Transportation
Wheelchair
Which provision
Medical advice (GGD)
Domestic care
Receive invoice
Personal budget
Personal budget or insurance
Payment
Insurance
Insurance company

Virtualize-Populate-Test idea

Generate
Virtual Environment
for testing

Environment created!

Populate the system
with the test-data

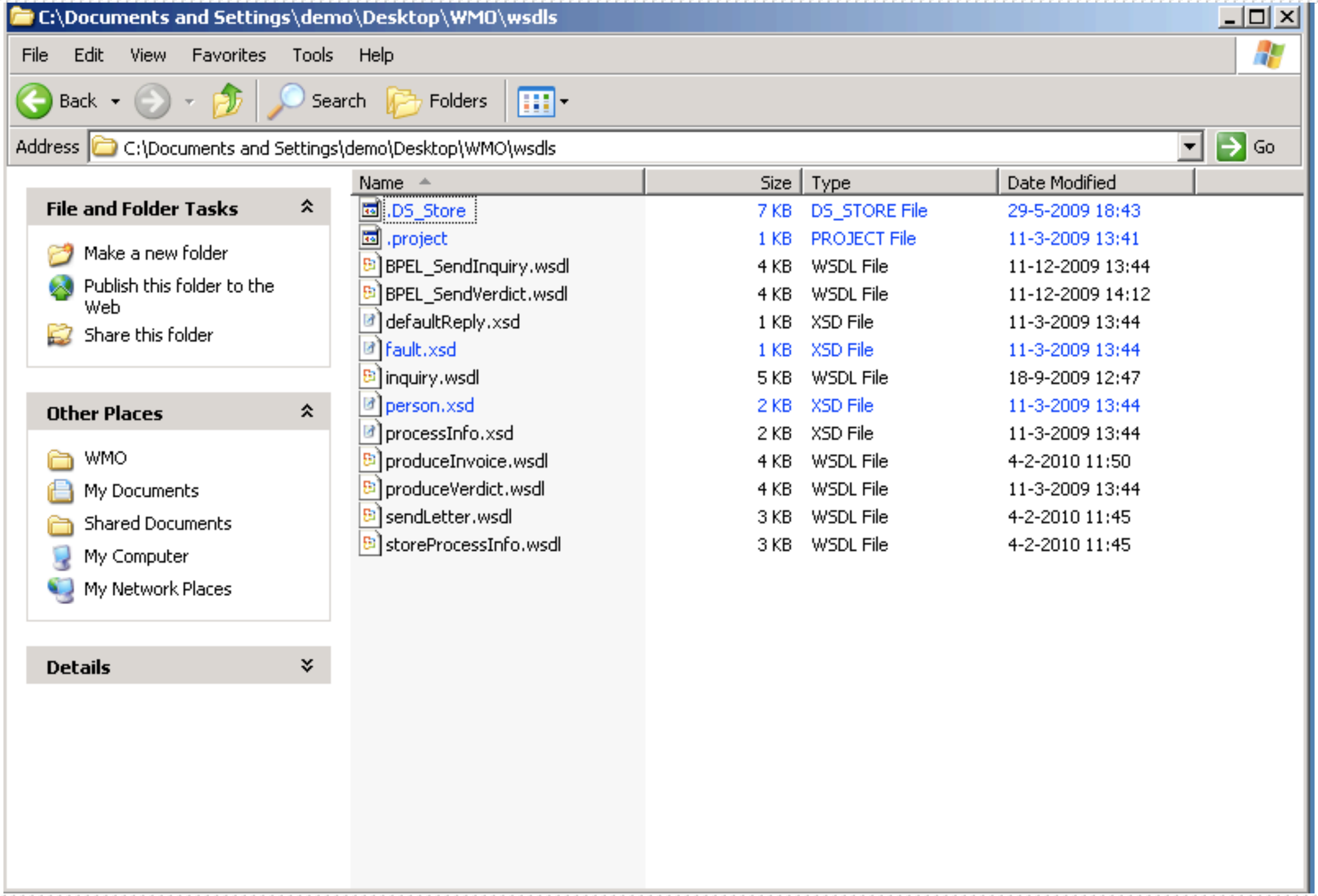
System populated!

Execute the tests
and automatically
generate the report

Tests executed!

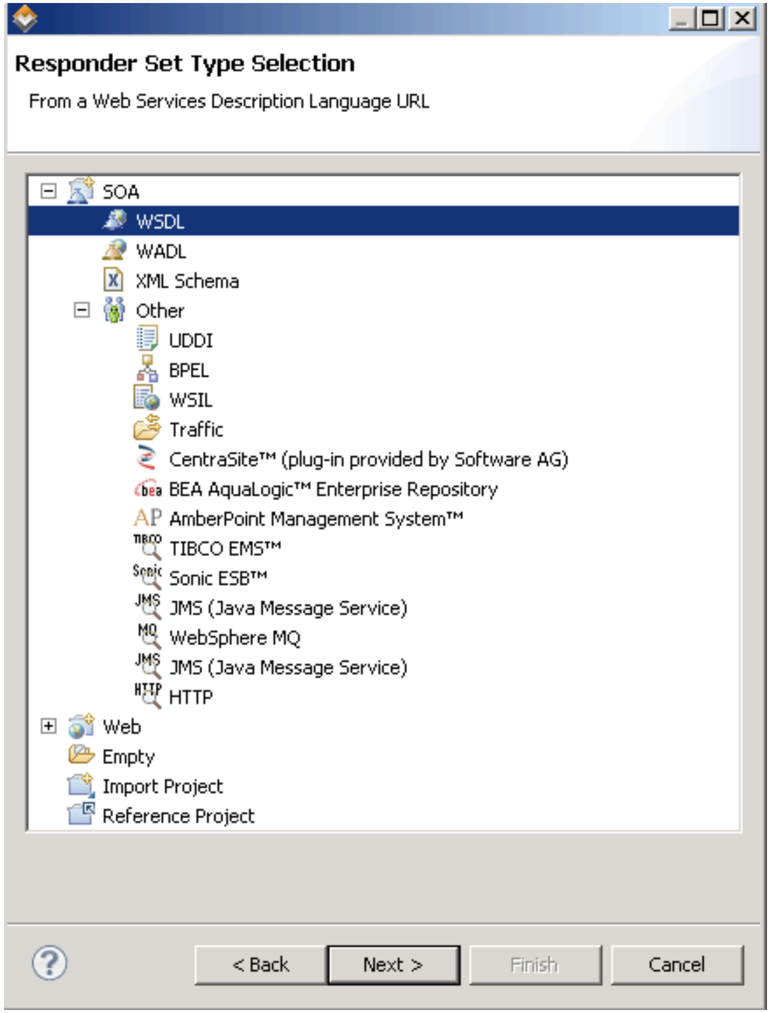
Generation of a **Virtual Environment**

- What?
  - Generation and deployment of virtual assets
- How?
  - Semi-automatic: using WSDLs or log files as an input



Name	Size	Type	Date Modified
.DS_Store	7 KB	DS_STORE File	29-5-2009 18:43
.project	1 KB	PROJECT File	11-3-2009 13:41
BPEL_SendInquiry.wsdl	4 KB	WSDL File	11-12-2009 13:44
BPEL_SendVerdict.wsdl	4 KB	WSDL File	11-12-2009 14:12
defaultReply.xsd	1 KB	XSD File	11-3-2009 13:44
fault.xsd	1 KB	XSD File	11-3-2009 13:44
inquiry.wsdl	5 KB	WSDL File	18-9-2009 12:47
person.xsd	2 KB	XSD File	11-3-2009 13:44
processInfo.xsd	2 KB	XSD File	11-3-2009 13:44
produceInvoice.wsdl	4 KB	WSDL File	4-2-2010 11:50
produceVerdict.wsdl	4 KB	WSDL File	11-3-2009 13:44
sendLetter.wsdl	3 KB	WSDL File	4-2-2010 11:45
storeProcessInfo.wsdl	3 KB	WSDL File	4-2-2010 11:45

Prepare a WSDL

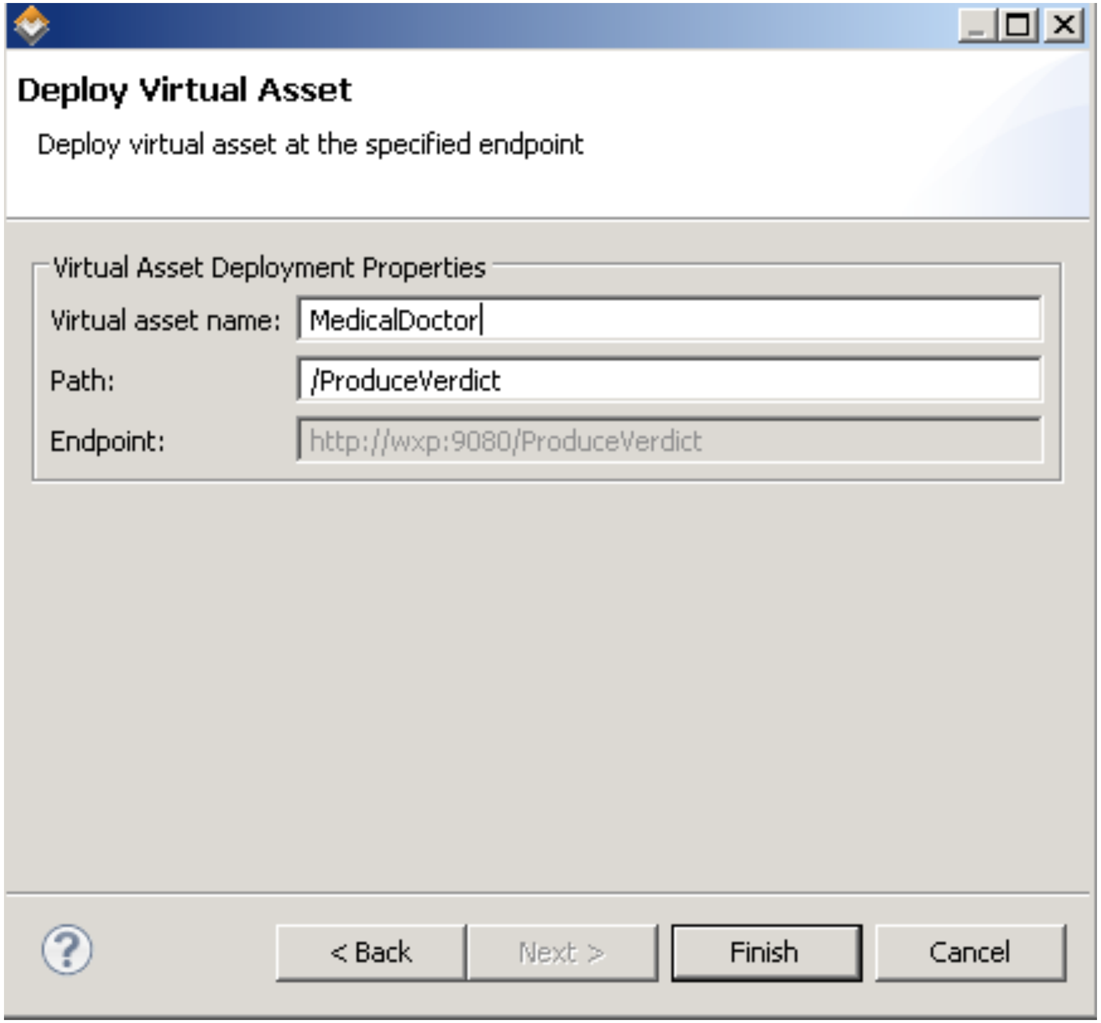


Responder Set Type Selection

From a Web Services Description Language URL

- SOA
  - WSDL
  - WADL
  - XML Schema
  - Other
    - UDDI
    - BPEL
    - WSIL
    - Traffic
    - CentraSite™ (plug-in provided by Software AG)
    - BEA AquaLogic™ Enterprise Repository
    - AmberPoint Management System™
    - TIBCO EMS™
    - Sonic ESB™
    - JMS (Java Message Service)
    - WebSphere MQ
    - JMS (Java Message Service)
    - HTTP
- Web
- Empty
- Import Project
- Reference Project

Choose a method for Virtual Asset generation



Deploy Virtual Asset

Deploy virtual asset at the specified endpoint

Virtual Asset Deployment Properties

Virtual asset name: MedicalDoctor

Path: /ProduceVerdict

Endpoint: http://wxp:9080/ProduceVerdict

Deploy a Virtual Assets

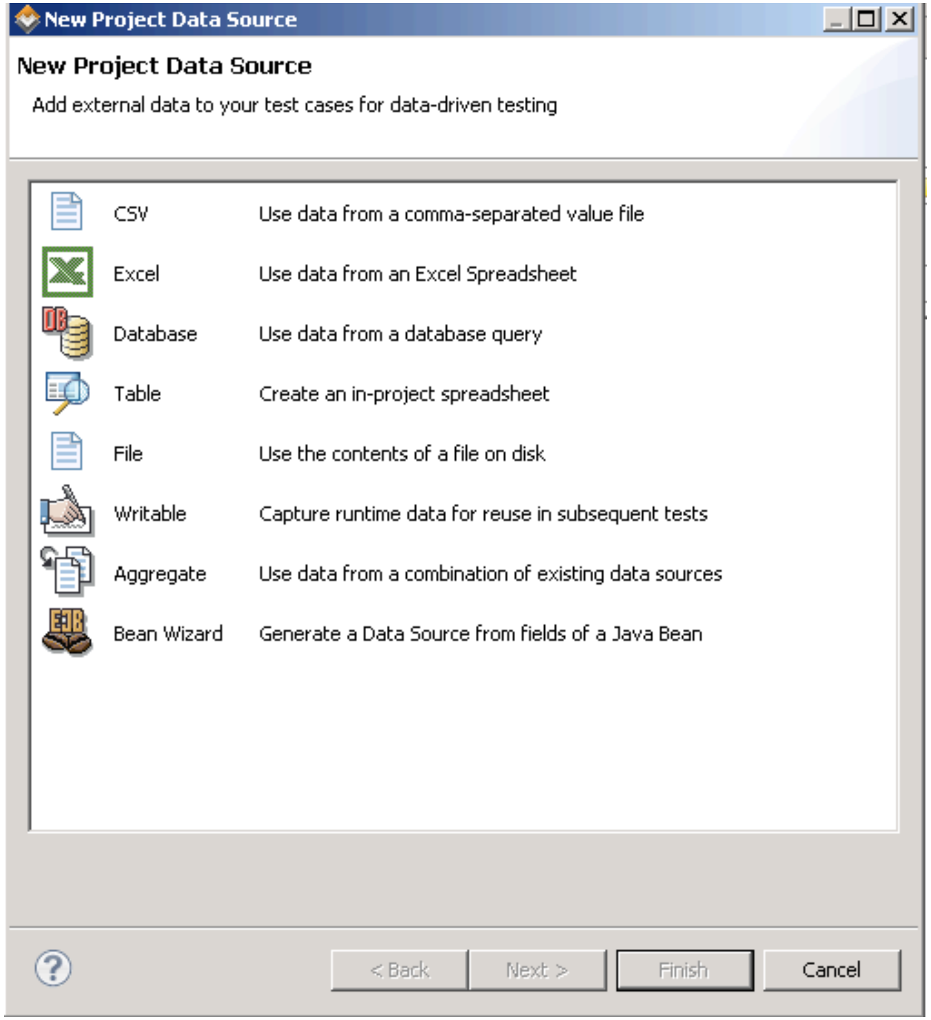
Population of a system with a **test-data**

› What?

- Tests-data that will verify all basic functionalities
- Input given <u>only one time</u>

› How?

- Business logic provided by testers or tools
- Tests compiled as a combination of test-data and message structure

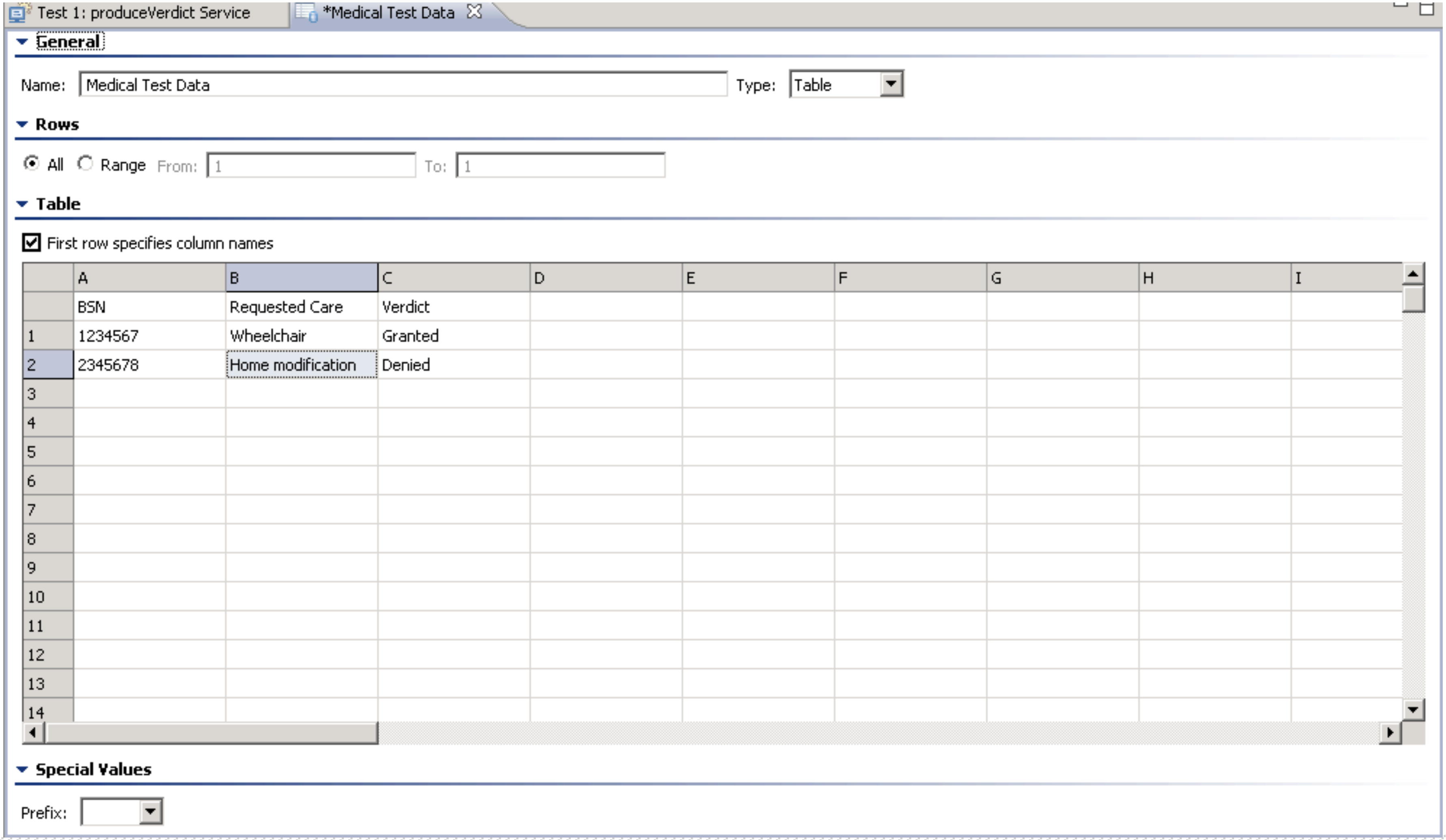


New Project Data Source

Add external data to your test cases for data-driven testing

CSV	Use data from a comma-separated value file
Excel	Use data from an Excel Spreadsheet
Database	Use data from a database query
Table	Create an in-project spreadsheet
File	Use the contents of a file on disk
Writable	Capture runtime data for reuse in subsequent tests
Aggregate	Use data from a combination of existing data sources
Bean Wizard	Generate a Data Source from fields of a Java Bean

Choose a data source



Name: Medical Test Data Type: Table

● All ○ Range From: 1 To: 1

☑ First row specifies column names

	A	B	C	D	E	F	G	H	I
	BSN	Requested Care	Verdict						
1	1234567	Wheelchair	Granted						
2	2345678	Home modification	Denied						
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									

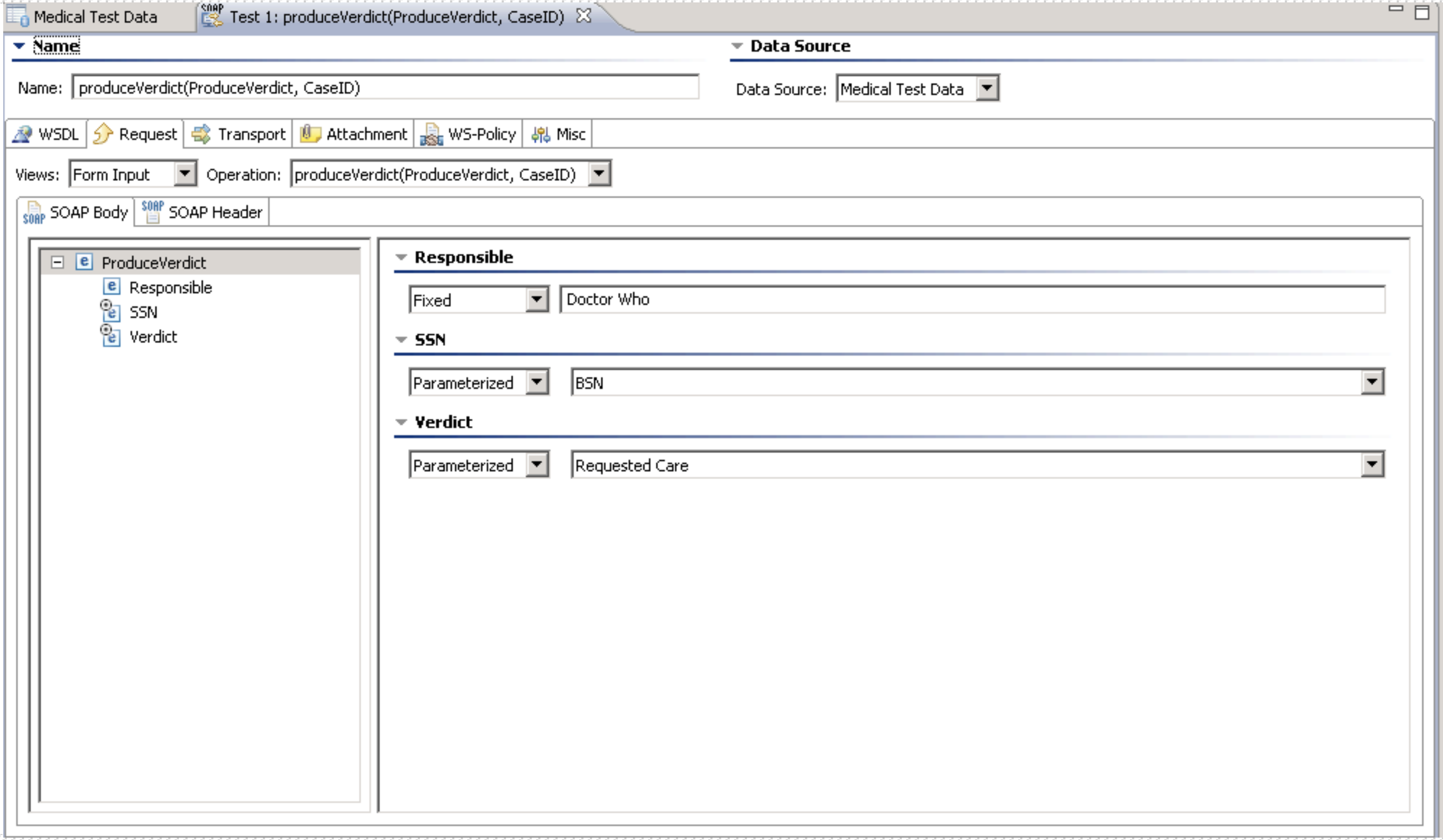
Prefix:

Model a behaviour of an external system

Test execution and automated **report generation**

- What?
  - Test execution
  - Automated report generation
- How?
  - Compiled tests executed on system that is connected to Virtual Environment
  - Automated report generated in various forms as comparison of observed result and defined assertions

university of groningen



Name

Name: produceVerdict(ProduceVerdict, CaseID)

Data Source

Data Source: Medical Test Data

Views: Form Input

Operation: produceVerdict(ProduceVerdict, CaseID)

- ProduceVerdict
  - Responsible
  - SSN
  - Verdict

Responsible

Fixed | Doctor Who

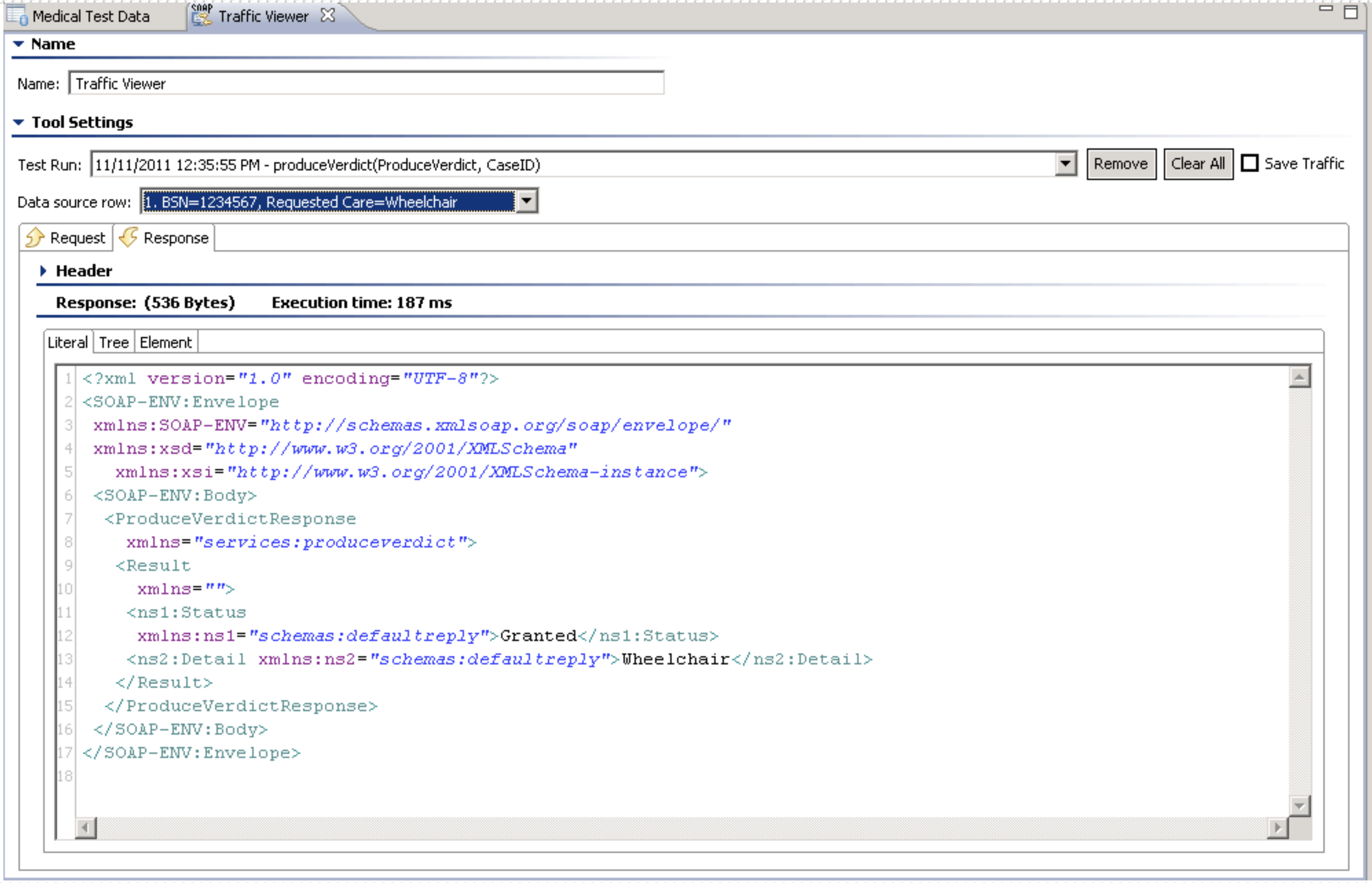
SSN

Parameterized | BSN

Verdict

Parameterized | Requested Care

Execute a test against a generated Virtual Environment



▾ Name

Name: Traffic Viewer

▾ Tool Settings

Test Run: 11/11/2011 12:35:55 PM - produceVerdict(ProduceVerdict, CaseID)

□ Save Traffic

Data source row: 1. BSN=1234567, Requested Care=Wheelchair

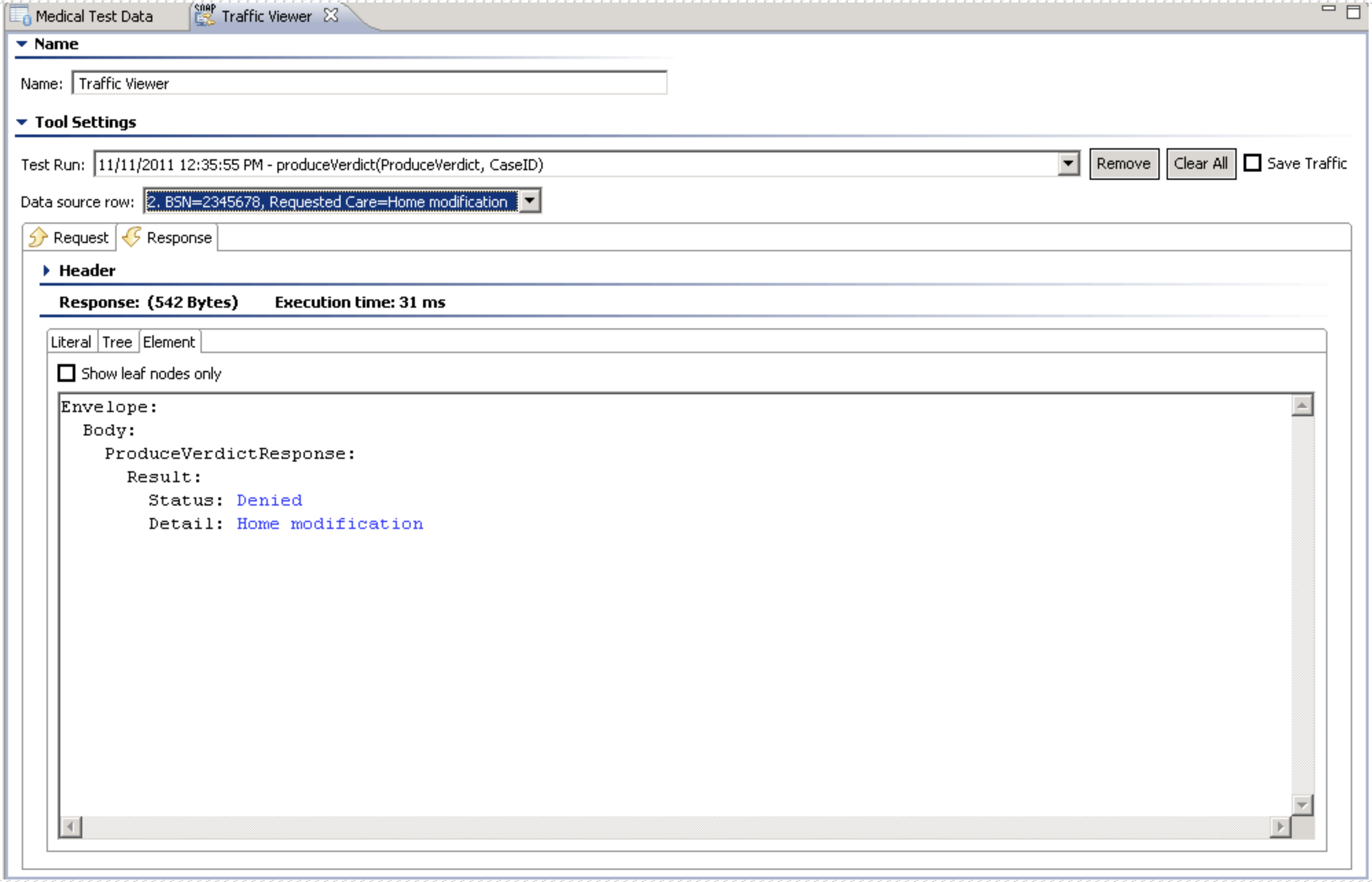
▸ Header

Response: (536 Bytes) **Execution time: 187 ms**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <SOAP-ENV:Body>
  <ProduceVerdictResponse
    xmlns="services:produceverdict">
   <Result
     xmlns="">
    <ns1:Status
     xmlns:ns1="schemas:defaultreply">Granted</ns1:Status>
    <ns2:Detail xmlns:ns2="schemas:defaultreply">Wheelchair</ns2:Detail>
   </Result>
  </ProduceVerdictResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Test case 1

university of groningen



Name: Traffic Viewer

Test Run: 11/11/2011 12:35:55 PM - produceVerdict(ProduceVerdict, CaseID)

□ Save Traffic

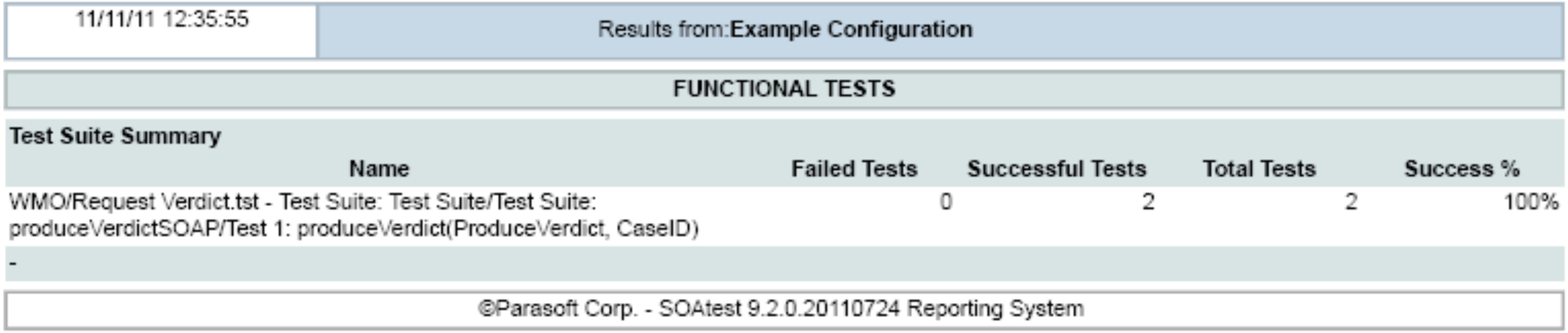
Data source row: 2. BSN=2345678, Requested Care=Home modification

Response: (542 Bytes) Execution time: 31 ms

□ Show leaf nodes only

```
Envelope:
  Body:
    ProduceVerdictResponse:
      Result:
        Status: Denied
        Detail: Home modification
```

Test case 2



11/11/11 12:35:55 Results from:**Example Configuration**

FUNCTIONAL TESTS

Test Suite Summary

Name	Failed Tests	Successful Tests	Total Tests	Success %
WMO/Request Verdict.tst - Test Suite: Test Suite/Test Suite: produceVerdictSOAP/Test 1: produceVerdict(ProduceVerdict, CaseID)	0	2	2	100%

-

©Parasoft Corp. - SOAtest 9.2.0.20110724 Reporting System

Automatically generate a report

Summary

- Kaag en Braassem example
- Current testing problems from a different perspectives
- Solution using virtualization
- Virtualize-Populate-Test idea
- WMO case study

The Research

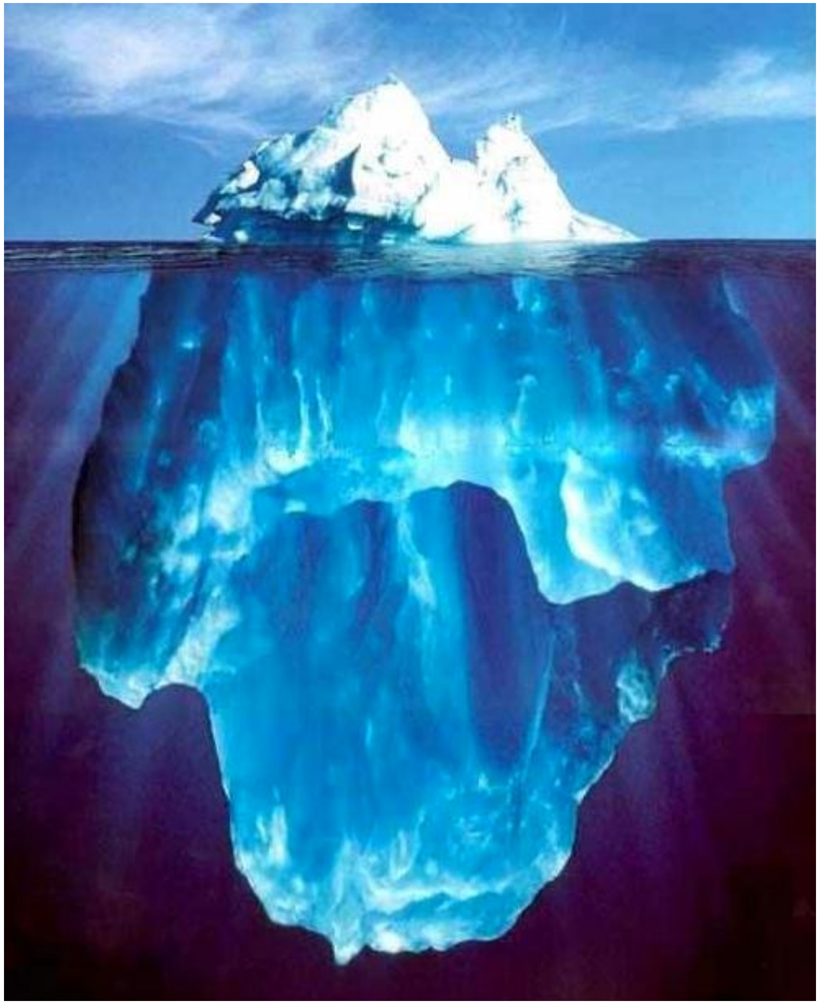
- Motivation
  - Evident problem without unified solution
  - Huge demand from industry for agile testing solution
  - Surprisingly not too many research projects in this field
- Aim
  - Define unified environment which will simplify automate testing of highly distributed service-oriented systems
  - Identify the processes that need to stay manual

Next steps

- Approach companies for appropriate systems (highly distributed service-oriented)
- Experiments
  - Create Virtual Environment
  - Model the behavior of external systems
  - Prove that most of issues can be resolved by this method
  - Improve methodology and tools

Big idea

- Simple switching between staged environments in the Cloud
- Merging testing and infrastructure knowledge
- Agile support for TestOps team



Thank you for your attention!

f.nizamic@rug.nl

References

- [1] Bart van Teeseling, “Testing Service-oriented Architectures using a simulation”, Master thesis at Faculty of Mathematics and Natural Sciences, University of Groningen

- [2] Tjitte D. Bouma, “Process analysis and requirement specification of Software as Service for WMO provision applications at Dutch municipalities”, Master thesis at Faculty of Economics and Business, University of Groningen

- [3] Alexander Lazovik, Marco Aiello, Rix Groenboom, “Smart testing at your service” project proposal, Distributed Systems Group, University of Groningen