VSR://EDU/SSE

Software Service Engineering

WS 2025/2026 – 4. Tutorial

Valentin Siegert M.Sc.

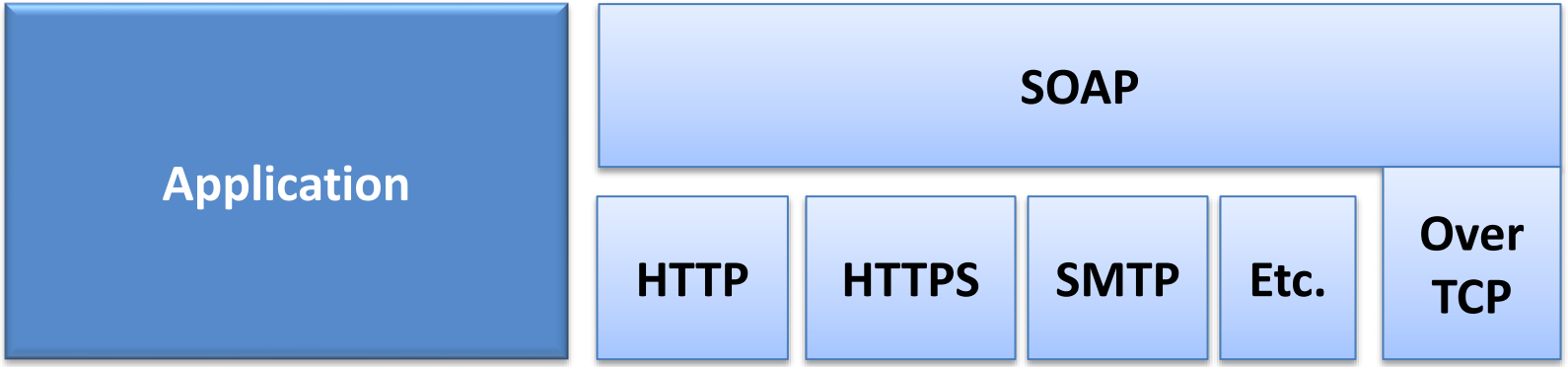
Maheshika Walpola M.Sc.

*VSR.Informatik.*TU-Chemnitz.*de*

Task 1

1. Answer the following questions.
   a) What is SOAP?
   b) Where would you situate SOAP within the Internet Protocol stack?
   c) Which parts are in a SOAP Message?

- SOAP is a protocol for the exchange of information in a distributed environment. SOAP messages are encoded as XML documents and can be exchanged using a variety of underlying protocols



- A SOAP message is encoded as an XML document, consisting of an <Envelope> element, which contains an optional <Header> element, and a mandatory <Body> element. The <Fault> element, contained within the <Body>, is used for reporting errors.

2. Use the link to access a simulator and get a first hands-on experience with SOAP:

   After the simulator has loaded and you can see the client and service:

   - Click on the client
   - In the right sidebar is the client's application code
   - Within the application code, fill out the two SOAP envelopes
   - Test your solution by clicking on "Start" in the header
   - Use Pause / Continue in the header to control the simulation
   - You can inspect the message contents by clicking on them
   - After making changes, click Reset to start the simulation again

   Your task will be to complete the SOAP envelopes to do the following:

   1. Invoke a "Ping" Request
   2. Invoke a "Echo" Request with a Paramter "Value", using a suitable value

   For the elements of the Request, use the XML Namespace "http://example.com/"

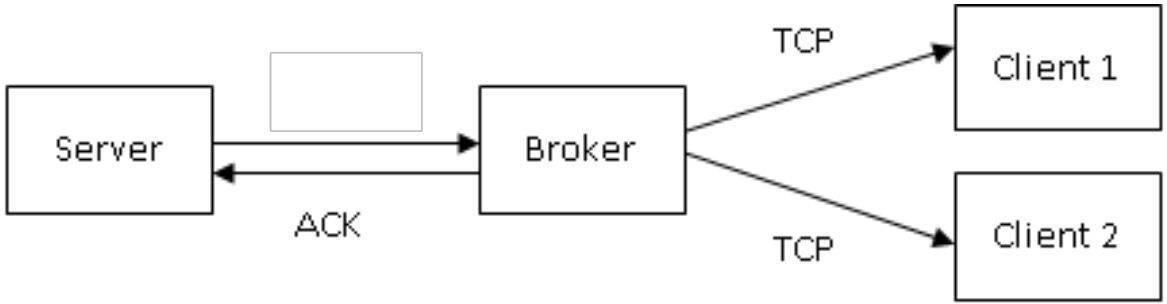
Task 2

1. Under http://vsr-demo.informatik.tu-chemnitz.de/webservices/SoapWebService/Service.asmx
   you will find a simple Web service, which is able to compute a sum of two integers (the Add operation). Implement a client for sending SOAP1.2 requests to the above service based on the template *Task2.1-Template.zip*.
   Print the result of the calculation to the console.

2. Complete the SoapClient.Invoke method of the Task2.2-Template.zip project, which should be able to send requests to ANY SOAP service using given service URL, service namespace, operationName and named parameters list (for simplicity request and response parameters can be only strings or integers).
   Test your class on the service above and the ConcatenatorService under http://vsr-demo.informatik.tu-chemnitz.de/webservices/ConcatenatorService/ConcatenatorService.asmx

Task 3

The project *Task3-Template.zip* contains a distributed system including a server, a broker and two clients. The Server sends regular stock price updates to the broker, which distributes them to the both clients. The broker is often overloaded and therefore the server should retransmit the message if broker doesn't acknowledge the message receipt (**robust one-way** MEP). The communication between the broker and clients should follow simple **one-way** MEP.



- Extend the *ReceiveMessage* method of the broker to send SOAP-based receipt notifications back to the server (BUSY or RECIEVED).
- The broker should not be able to read the contents of some SOAP messages. Update the server and **client 1** to use shared secret to encrypt and decrypt contents of SOAP messages.
- **Client 2** should not be equipped with any encryption/decryption facilities and therefore ignore encrypted messages. Make server send appropriate SOAP headers to signal the need of decrypting content before processing it.

VSR

Your feedback on today's session:

mytuc.org/ttbw

Questions?

vsr-sse@informatik.tu-chemnitz.de

VSR.*Informatik*.TU-Chemnitz.*de*