ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Distributed Systems 2012 – Assignment 2

Anwar Hithnawi
hithnawi@inf.ethz.ch

Web Services

Today's Menu

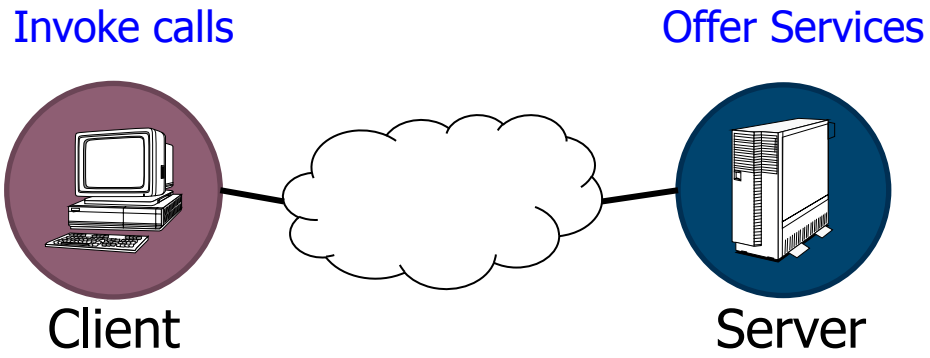
- Quick walkthrough of Web application architectures
  - WS-* **W**eb **S**ervices
  - **Re**presentational **S**tate **T**ransfer (REST)

- Exercise 2
  - Overview
  - Tasks
  - Hints & Anchors

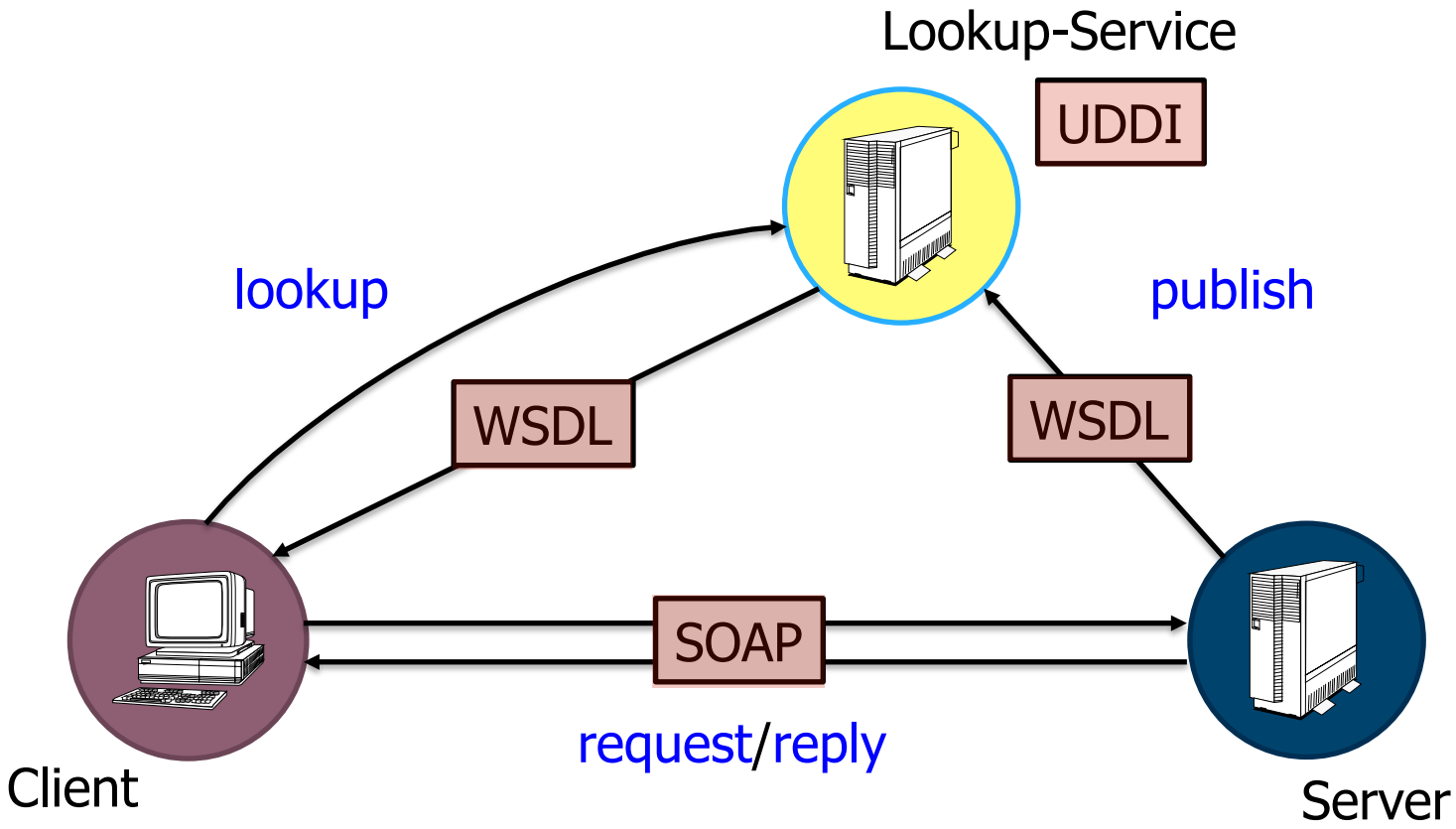
Web Services

- Definition:

“ A Web service is an application component accessible over open protocols”

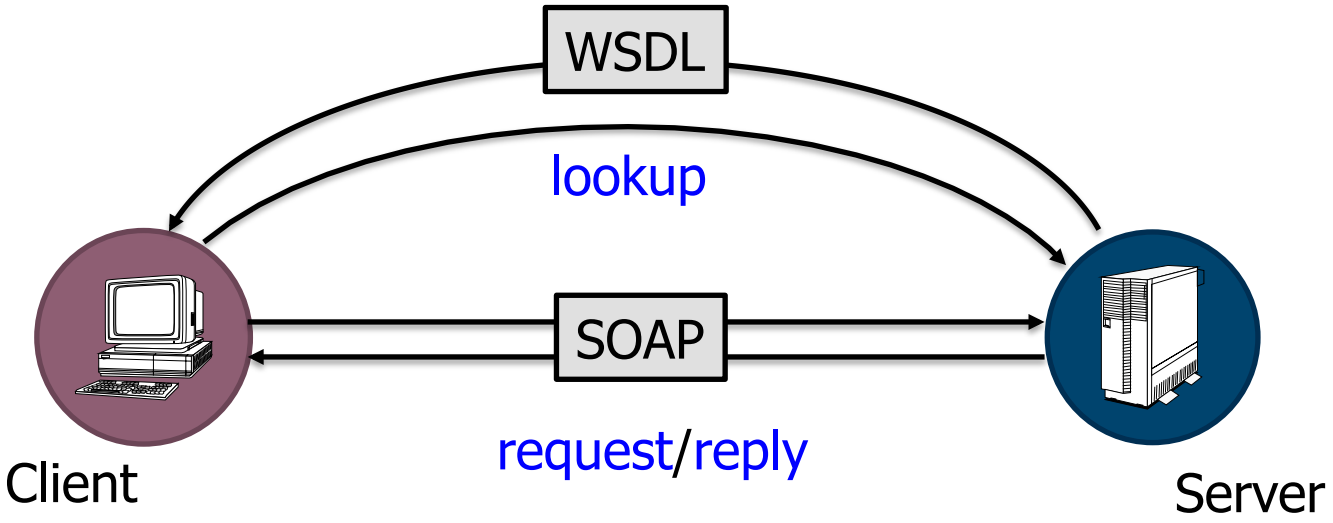


Web Services in a Nutshell



Web Services in a Nutshell

- For the exercise, we let the service publish its WSDL without going through a UDDI...

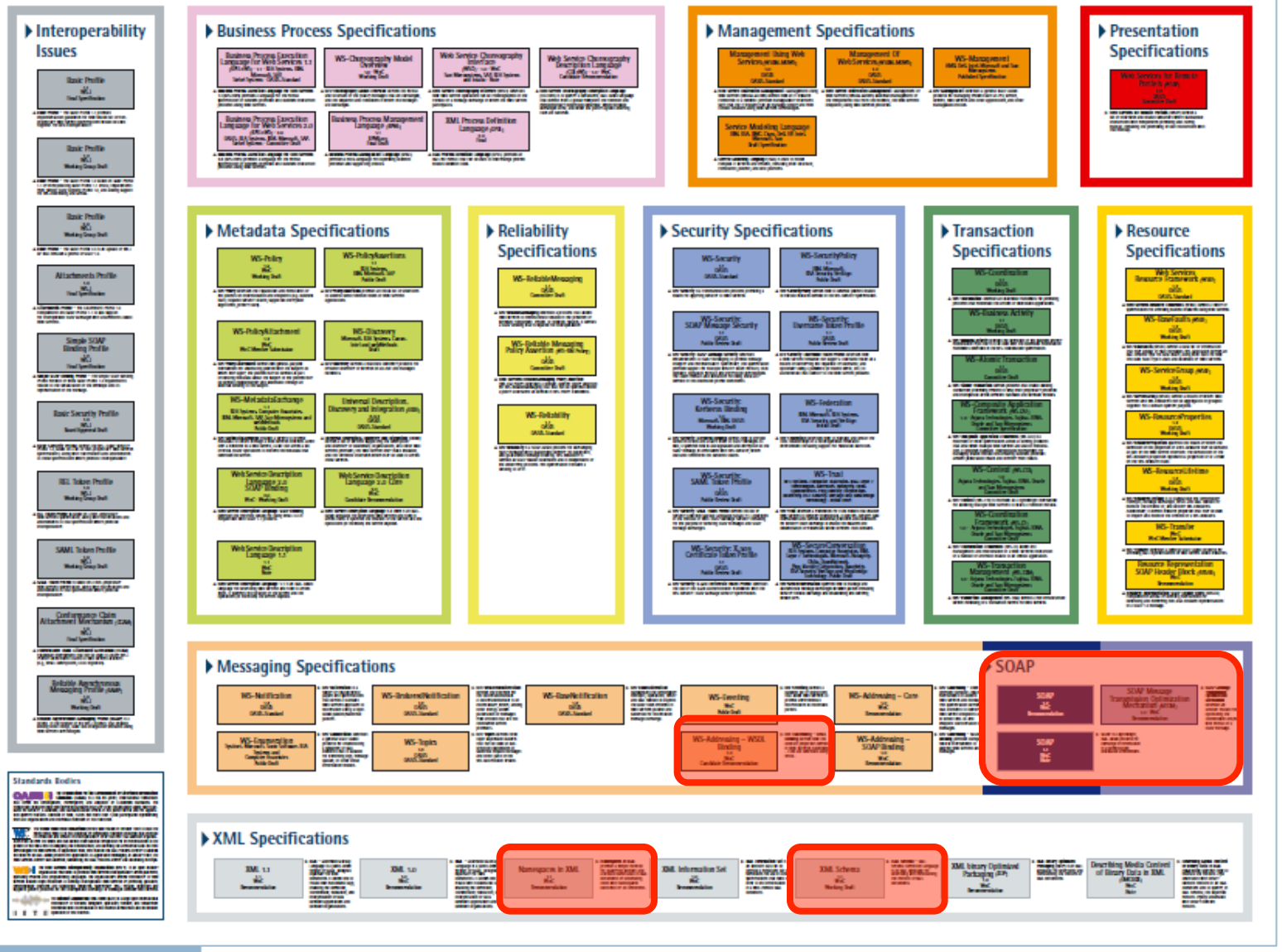


Web Services - WSDL Overview

- WSDL: **W**eb **S**ervices **D**escription **L**anguage describes:
  - What a Web service can do
  - Where it resides
  - How to invoke it
- Explore WSDL

  Ex. [http://vslab.inf.ethz.ch/SunSPOTWebServices/SayHello?Tester]

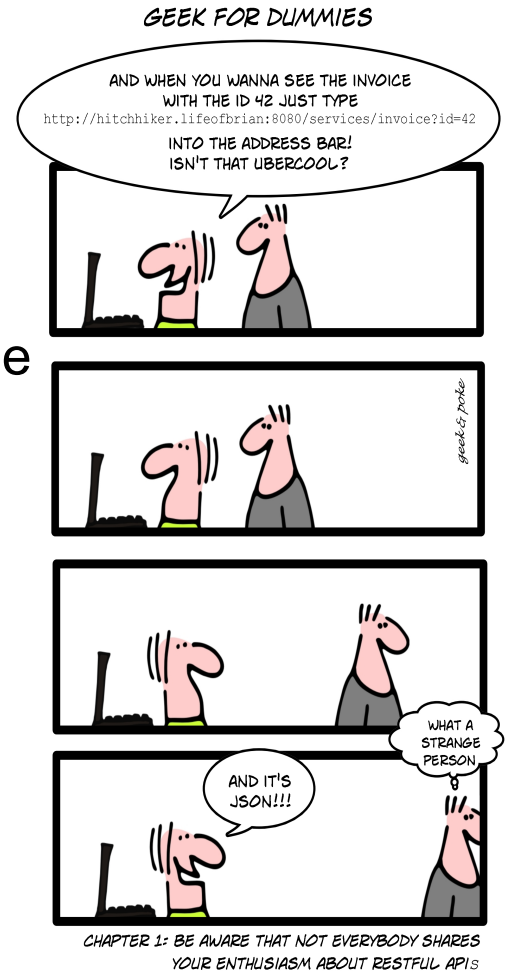
  Types, Messages, PortType, Binding, Service, Port, Definition



[http://www.innoq.com/soa/ws-standards/poster/innoQ%20WS-Standards%20Poster%202007-02.pdf]

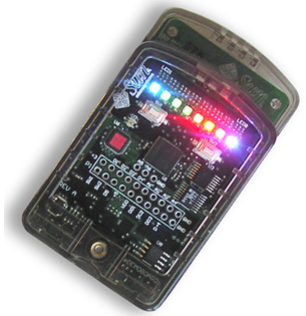
REST: Representational State Transfer

- REST is a lightweight architecture style for designing networked applications
  - HTTP 1.1 implements the REST architectural style
    - It uses HTTP for CURD (Create/Update/Read/Delete) operations
- Platform independent
- Language independent
- Standard-based



[http://geekandpoke.typepad.com/]

REST Architecture



[http://code.google.com/p/hcsfsp/]

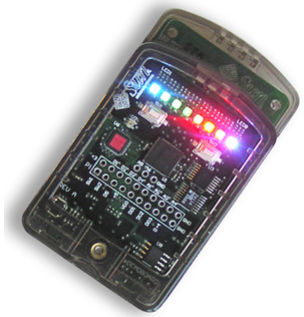
- **Resources:** Which are identified by logical URIs
  - State and functionality are represented using resources

    e.g., a sensor node: [http://vslab.inf.ethz.ch:8081/sunspots/Spot1]

- **A web of resources**: Resources are linked
  - Similar to the interconnection of web pages in the WWW
  - When relevant, resources should link to additional information
    - Resources should be kept simple

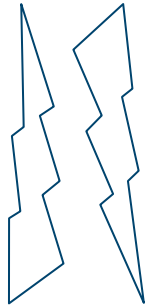
- **Stateless** communication protocol:
  - Each new request must carry all the information required to complete it

Assignment 2 – Overview



[http://code.google.com/p/hcsfsp/]

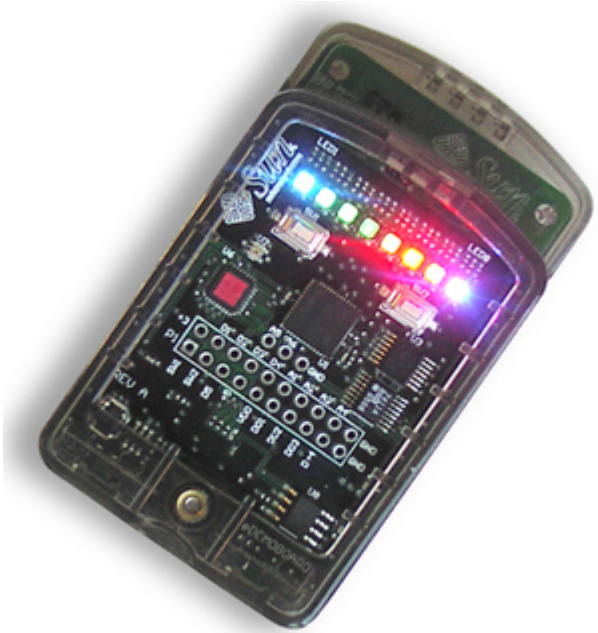
- Objectives:
  - Learn to develop distributed Web applications
  - Use the two different paradigms seen in the lecture:
    - Representational State Transfer (REST)
    - Web Services (WS-*)



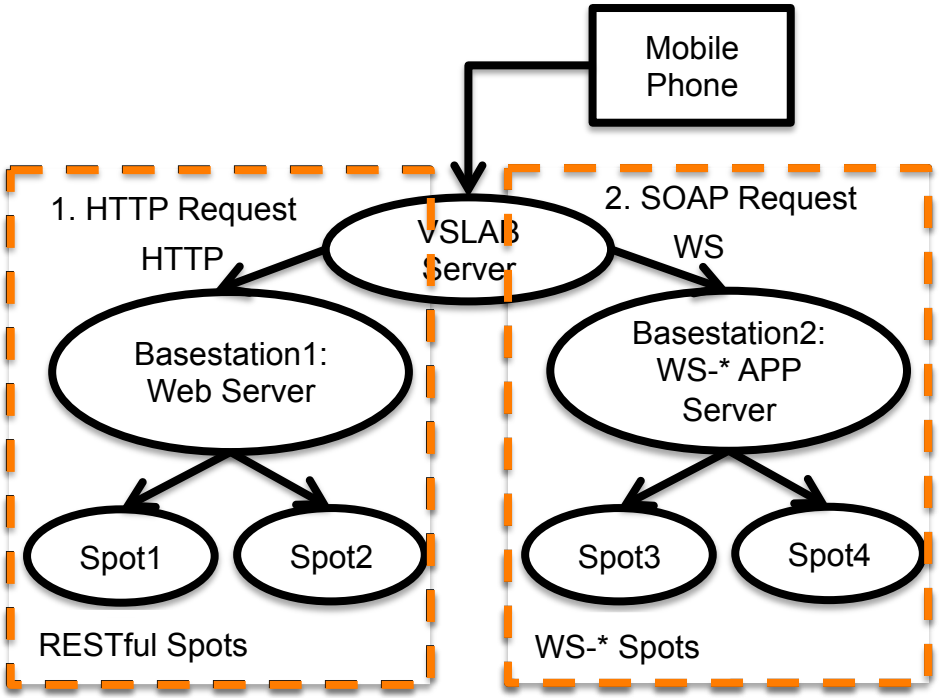
- Dates:
  - Exercise begins: Now (October 12, 2012)
  - Exercise is due: 9:00 am, October 24, 2012

Assignment 2 – System Setup

- Access Sun SPOTs through WS-*/REST
- Sun SPOTs: Wireless sensor nodes (temp, acc, light,...)



[http://code.google.com/p/hcsfsp/]



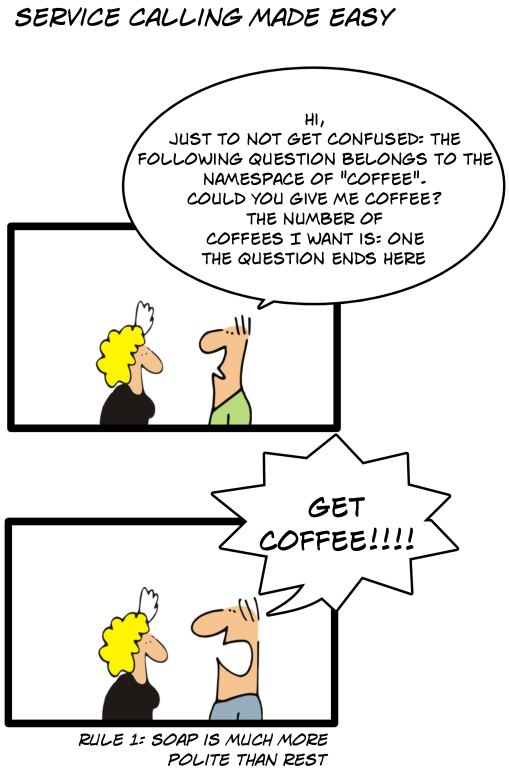
Assignment 2 – Tasks

1. Experimenting with RESTful Web Services (2P)
   - Create an HTTP request
     a) "by hand" (i.e., without the use of any HTTP library)
     b) using *org.apache.http.**
   - Use HTTP content negotiation to get machine-readable data
   - Connect to a Sun SPOT and retrieve the temperature value
2. Experimenting with WS-* Web Services (2P)
   - Explore WSDL, create SOAP requests
   - Connect to a Sun SPOT and retrieve the temperature value.
   - **Hint:** You should not perform long running operation on the UI thread. Specifically for this task network access.

Assignment 2 – Tasks

3. Assessing Web Service Technologies (1P)

- Evaluate the two technologies, REST and WS-* by answering the questions in the form:
  [http://tinyurl.com/9mur7w6]



[http://geekandpoke.typepad.com/]

Assignment 2 – Tasks

4. Cloud Services (1P)
   - Visualization of retrieved measurements using the *Google Chart API*

5. Your Phone as a Server (2P)
   - Implement a Web Server on your phone that allows to access the phone's sensors and actuators

6. Report (2P)

Submission

- Same as for Assignment 1
  - Programs/Code, Report

+ Assignment form:

[http://tinyurl.com/9mur7w6]

Assignment 2 Hints - Relevant Terminology

- Media types: HTML, XML, JSON

- ROA – Resource-Oriented Architecture
- REST – Representational State Transfer

- SOA – Service-oriented Architecture
- SOAP – Simple Object Access Protocol
- WSDL – Web Services Description Language

REST Hints

- http://www.infoq.com/articles/rest-introduction

- RESTful Web Services (Leonard Richardson und Sam Ruby)
  - Available at D-INFK library

- Apache HTTP library (simplest sample code alive... ☺)
  - http://svn.apache.org/repos/asf/httpcomponents/httpclient/trunk/httpclient/src/examples/org/apache/http/examples/client/ClientWithResponseHandler.java

WS-* Hints

- Patched version of kSOAP2
  - http://code.google.com/p/ksoap2-android/

- Short tutorial on kSOAP2 for Android
  - http://www.android10.org/index.php/articleslibraries/167-using-ksoap2-for-android-soap-web-service

Visualization Hints

- Google charts API example:

  https://chart.googleapis.com/chart?chs=250x100&chd=t:60,40&cht=p3&chl=Hello|World

- Getting started:

  http://code.google.com/apis/chart/image/docs/making_charts.html#usingthewizard

Have **F**un **P**rogramming**!**

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Introduction to Assignment 2

Distributed Systems Lecture
HS 2012, ETH Zurich

Anwar Hithnawi
Anwar.hithnawi@inf.ethz.ch