

Application Development for Mobile and Ubiquitous Computing

LunchBox

Final Presentation

Group 12
Dana Henkens
Franz Grüneberger

- Introduction / Application Scenario
- Screenshots of the Running Application
- Technologies
- Challenges
- Abilities/Limits
- Conclusion

- target group:
 - students thinking about having lunch at the refectory
 - research assistants using the refectory of our university

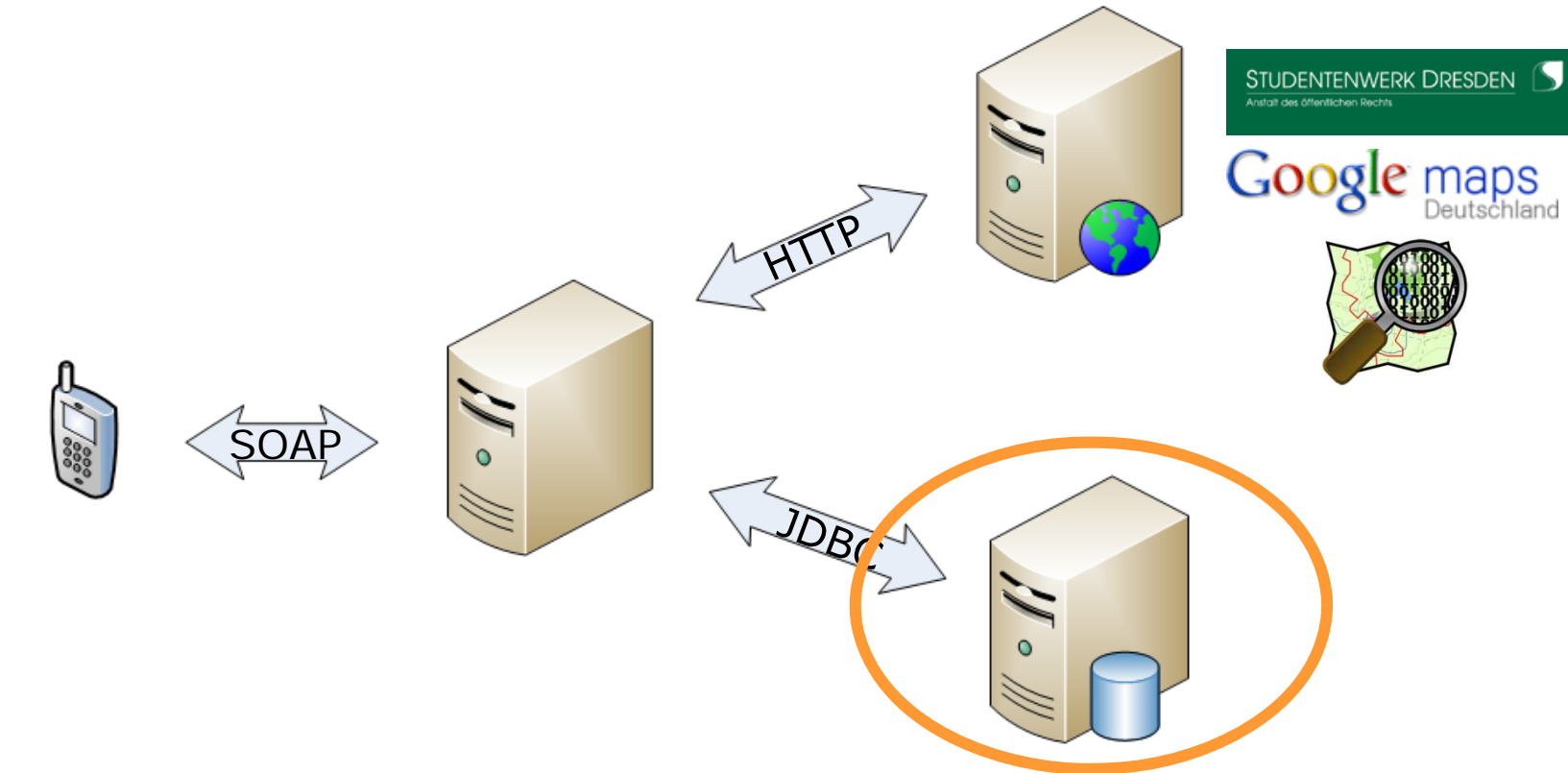
- questions:
 - Where should I go to have lunch?
 - Which meals should I avoid? Which ones are recommendable?
 - Whats about the price?

- our solution:
 - mobile application providing information about the meals at the refectories including ratings









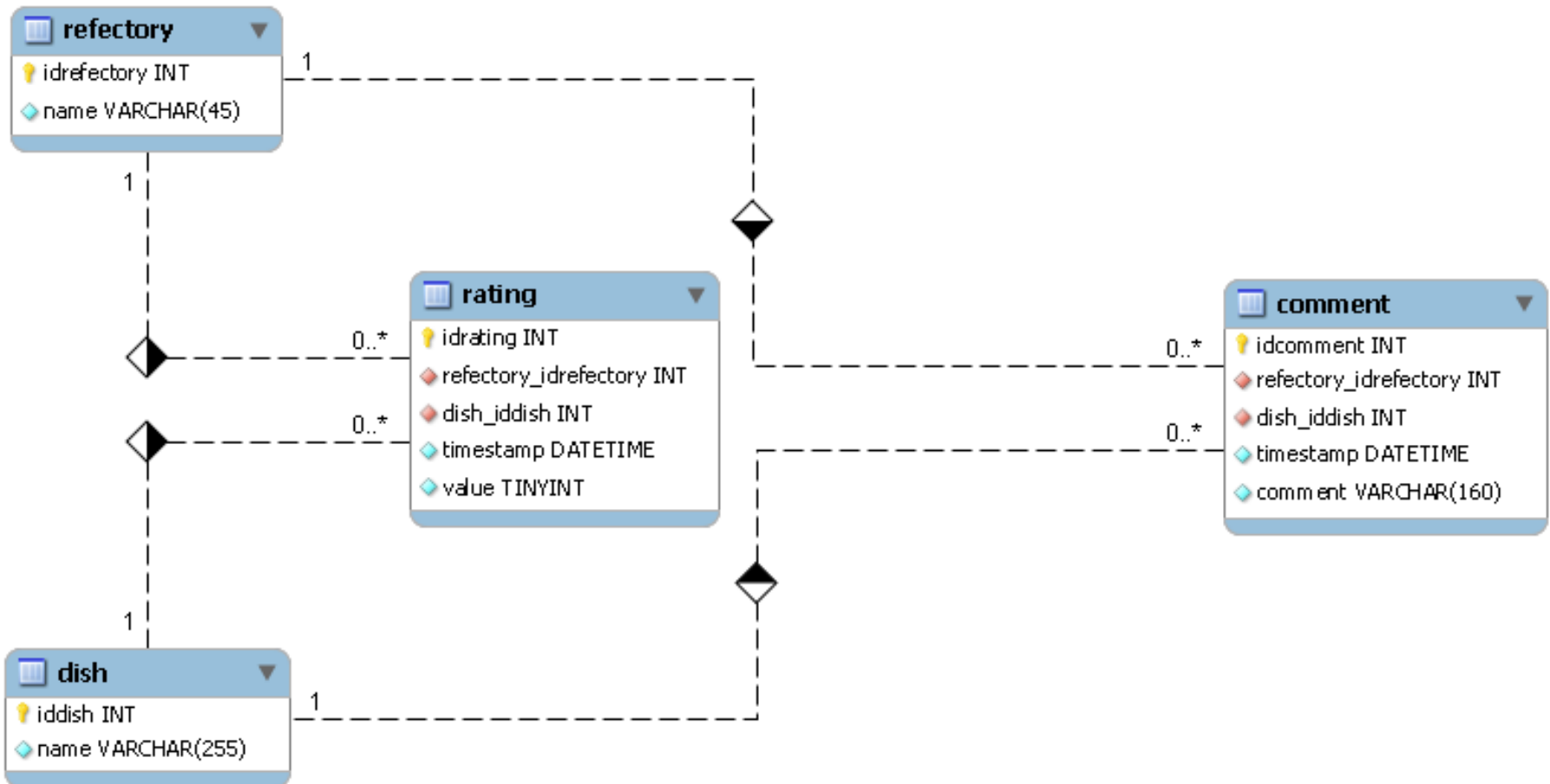
Java ME with
LWUIT 1.3

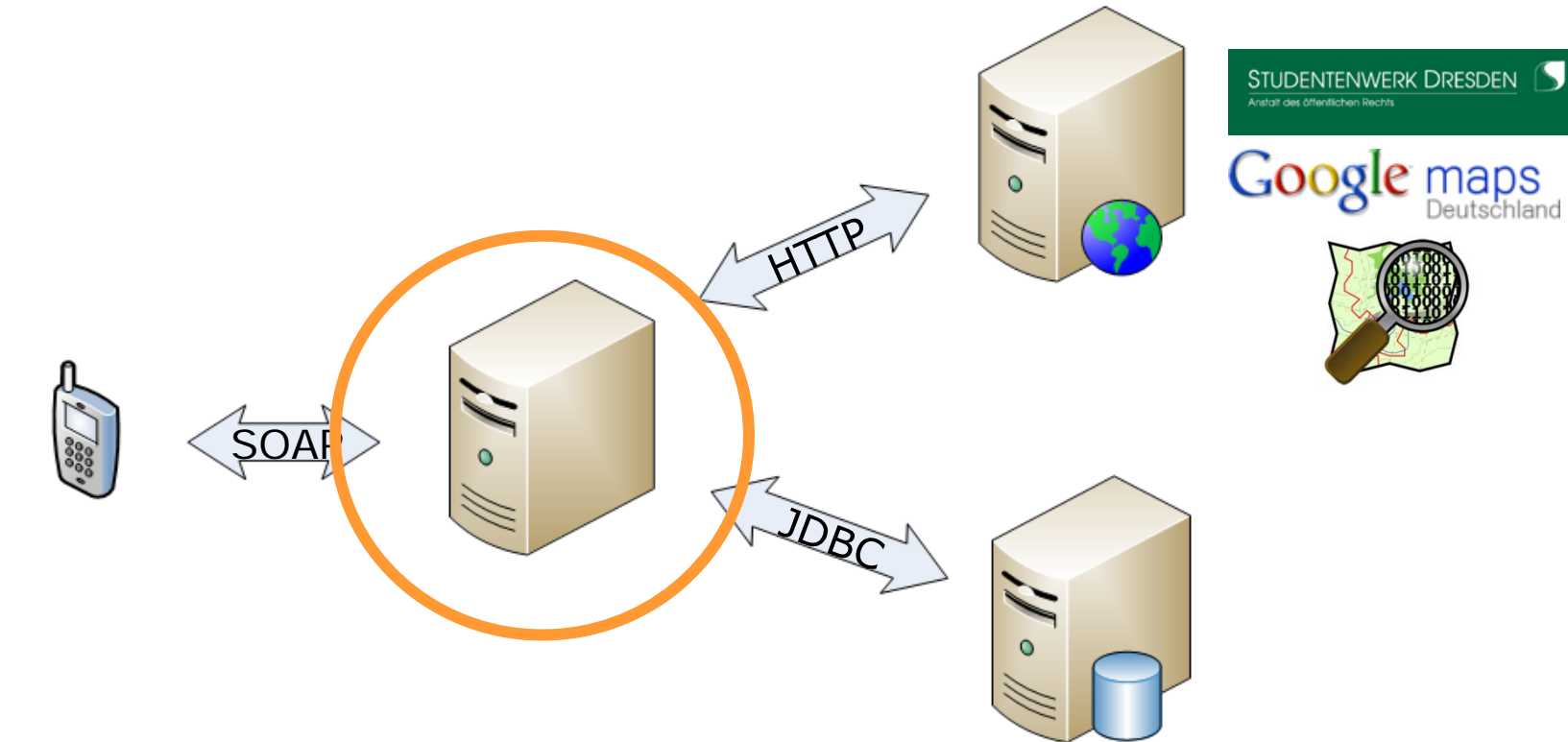


Apache Tomcat 6.0.18
with JAX-WS Servlet



MySQL 5.1.37





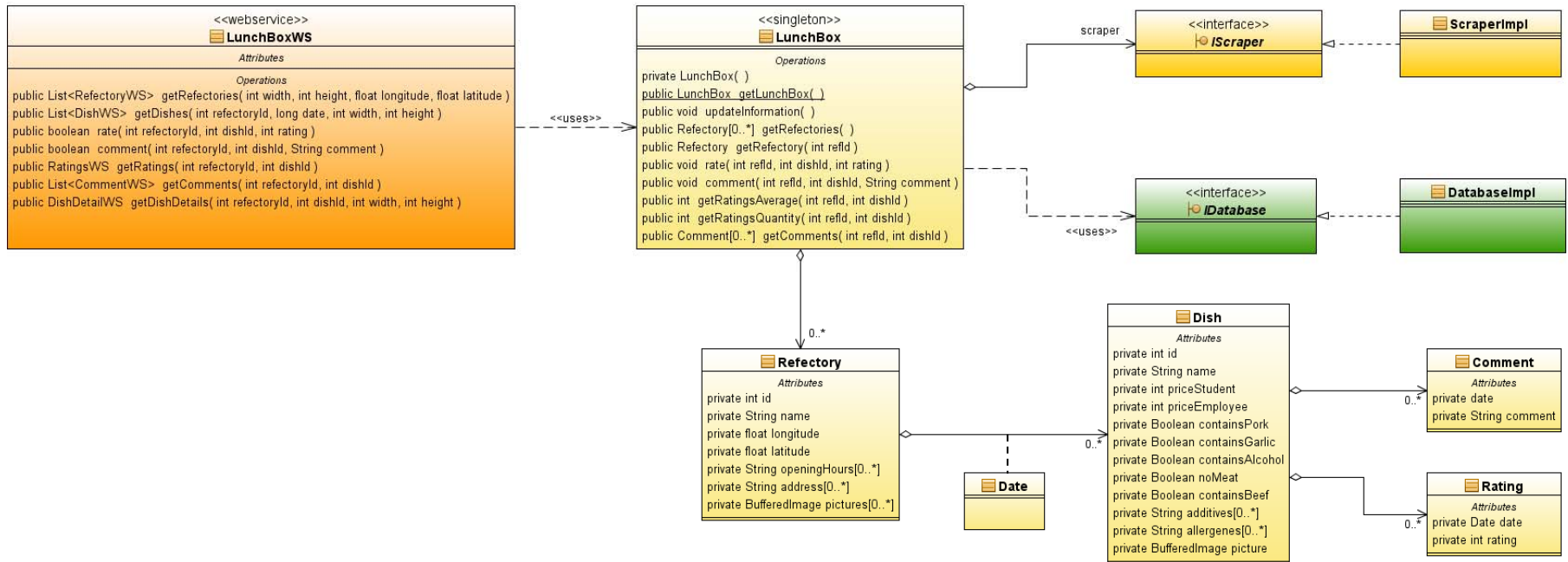
Java ME with
LWUIT 1.3



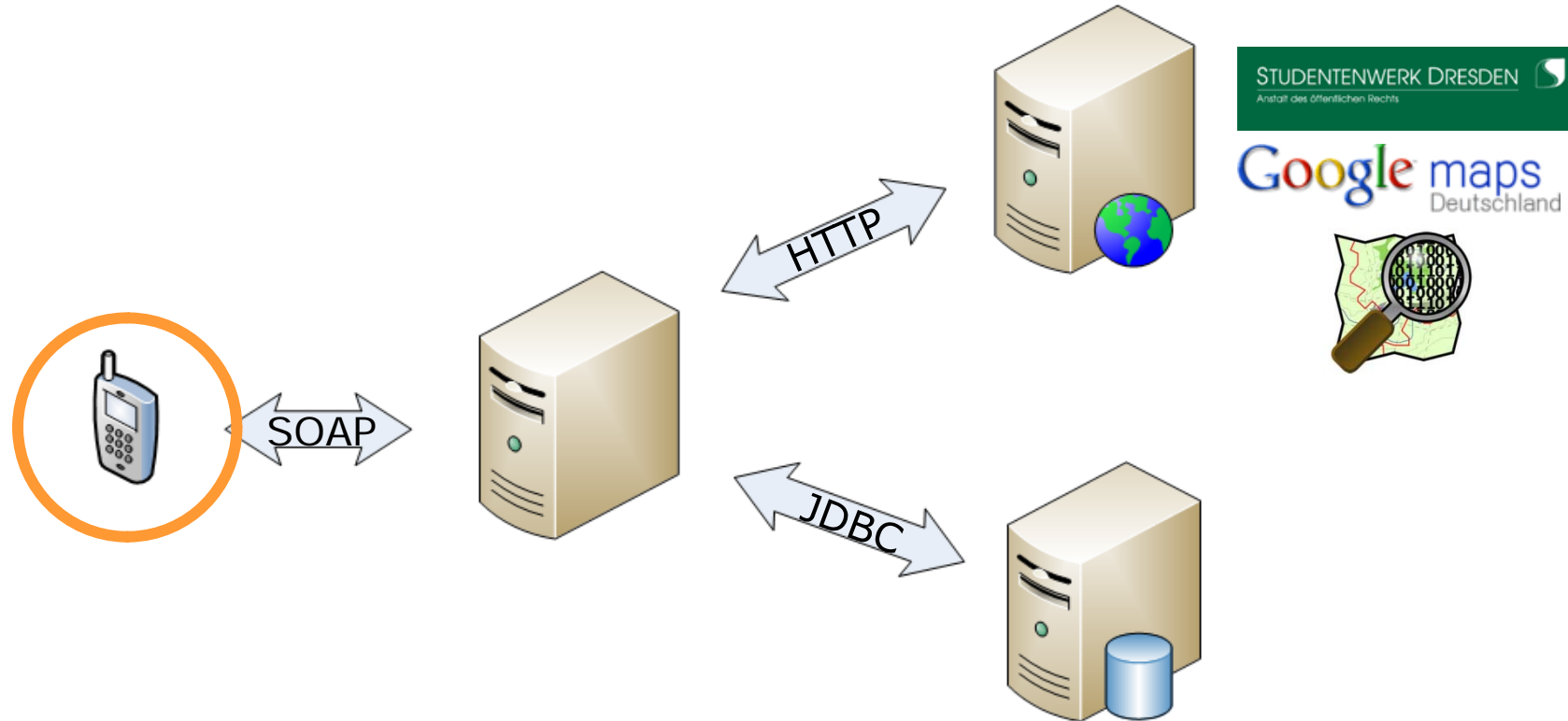
Apache Tomcat 6.0.18
with JAX-WS Servlet



MySQL 5.1.37



- development by means of Netbeans-IDE and GlassFish v2.1 as test environment on Windows OS
 - major advantage of GlassFish over Tomcat: WebService-Tester
 - provides test environment for WebServices, i.e. the ability to:
 - generate SOAP-requests by passing parameters for the method to be tested via input fields within a html-page
 - view generated SOAP-response
 - problem:
 - deployment-descriptor generated for GlassFish incompatible to Tomcat
- production environment: Tomcat at Debian OS
 - problems:
 - security policies of catalina servlet-container very strict by default
 - encoding used by the JVM differs from Windows
 - heapsize of the JVM must be increased



Java ME with
LWUIT 1.3



Apache Tomcat 6.0.18
with JAX-WS Servlet



MySQL 5.1.37

- Lightweight UI Toolkit (LWUIT)
 - advantages:
 - o styling based on themes
 - o tool for managing themes available
 - administration of fonts, colors, images, ...
 - o themes can be exchanged at runtime
 - disadvantages:
 - o not enough capabilities with respect to some special purposes e.g. scrolling in tabbed panes, ...
 - o bad documentation in some cases

 **data scraping**

- „Studentenwerk“ doesn't provide any API
- solved by means of *Cobra Java HTML Parser* and *HtmlCleaner*

 → BUT: application still dependent on working website**usability and customer satisfaction**

- clear structured user interface
- use of *Lightweight UI Toolkit (LWUIT)*

 **restricted device capabilities**

- adaptation of images to the screen size (resolution)
- adaptation by means of functions provided by the JAVA API at the server



heterogeneity

- usability at any devices supporting MIDP 2.0 and CLDC 1.1
- tests on some simulators done



context awareness

- using location based information for calculating the distance to different refectories
- distance calculation to the different refectories at the server by means of google maps API and YOURS
- caching of the last calculated distances due to performance

- Abilities:
 - fast, convenient access to all dishes provided at the refectories of Studentenwerk
 - value added in terms of comments and ratings supporting your own choice

- Limits:
 - no support of disconnected operations
 - no automatic update of distances while staying at refectory list
 - no personalization
 - no provision of customized information, e.g. today's best meals according to personal ratings only
 - no individual themes
 - no different languages

- What did we learn?
 - JavaME
 - only very restricted abilities, e.g. no collections available, ...
 - low hardware requirements, but today less important
 - LWUIT for GUI development
 - inspired by Swing
 - at the API-level almost identical to AWT
 - Netbeans
 - provides good support for WebService development
 - automatic client-stub generation for JavaME faulty
 - Tomcat
 - unsatisfying debug features for WebServices
 - --> better choice: GlassFish

Thank you for your attention!