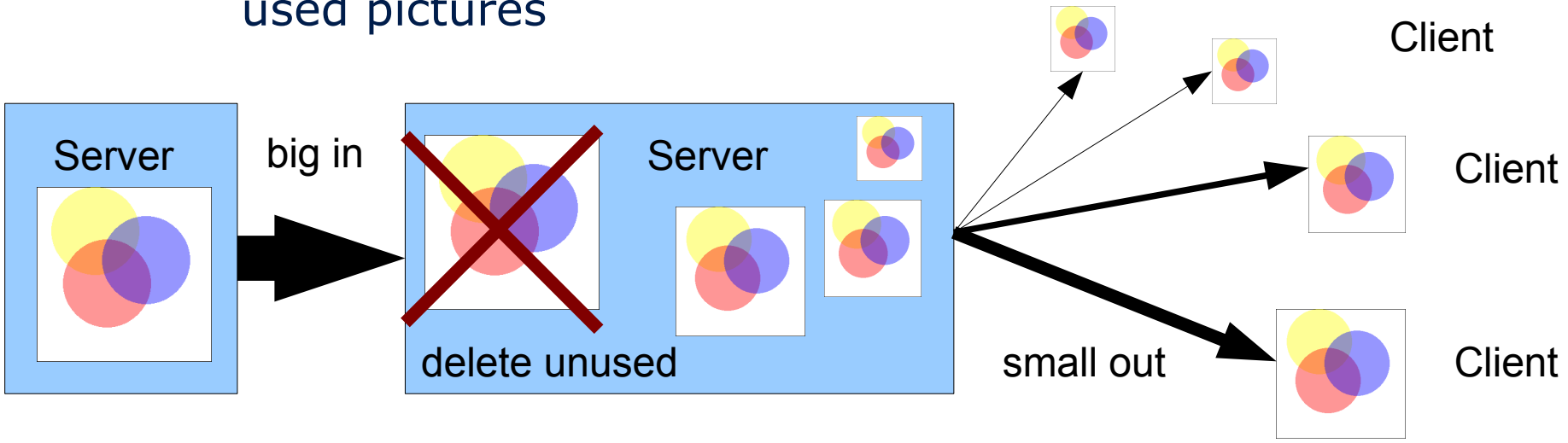
Application Development for Mobile and Ubiquitous Computing

Image Scaling Web Service
Final Presentation

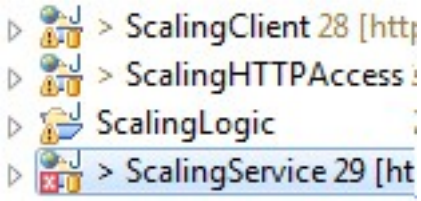
Benjamin Vetter, Reik Müller

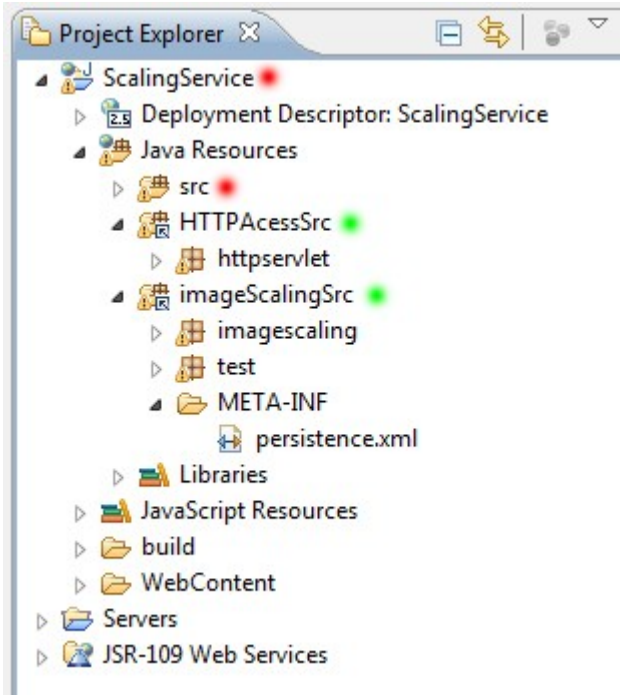
Concept

- One server provides customized images using parameters given by a client
- Available for ordinary HTML pages and over SOAP
- Using as much disc space as necessary to deliver images without calculating them
- Using as less disc space as possible by deleting not used pictures



- Changes in a wsdl file result in a rebuild of the web-service sub-project

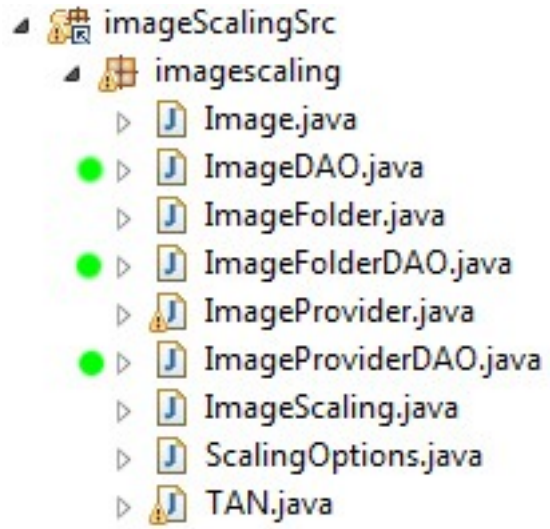




- Split the project
- Reconnect it in an 5th project

Problems and Solutions

- JPA 2.0 with Eclipselink produced annoying exceptions



- Rollback
- Inconsistent EntityManager
- Queryparsing
- Primary key updates (bug in eclipselink?)

→ DAOs increased readability of code

Problems and Solutions

- New ideas brought heavy code changes spread over all sub-projects

HTTPServlet

```
ScalingOptions options = new ScalingOptions();
ImageScaling app = new ImageScaling();

Enumeration e = request.getParameterNames();

while (e.hasMoreElements()) {
    String name = (String) e.nextElement();
    String value = request.getParameter(name);

    OPTION option = ScalingOptions.OPTION.valueOf(name);
    options.ultimateSetter(option, value);
}
```

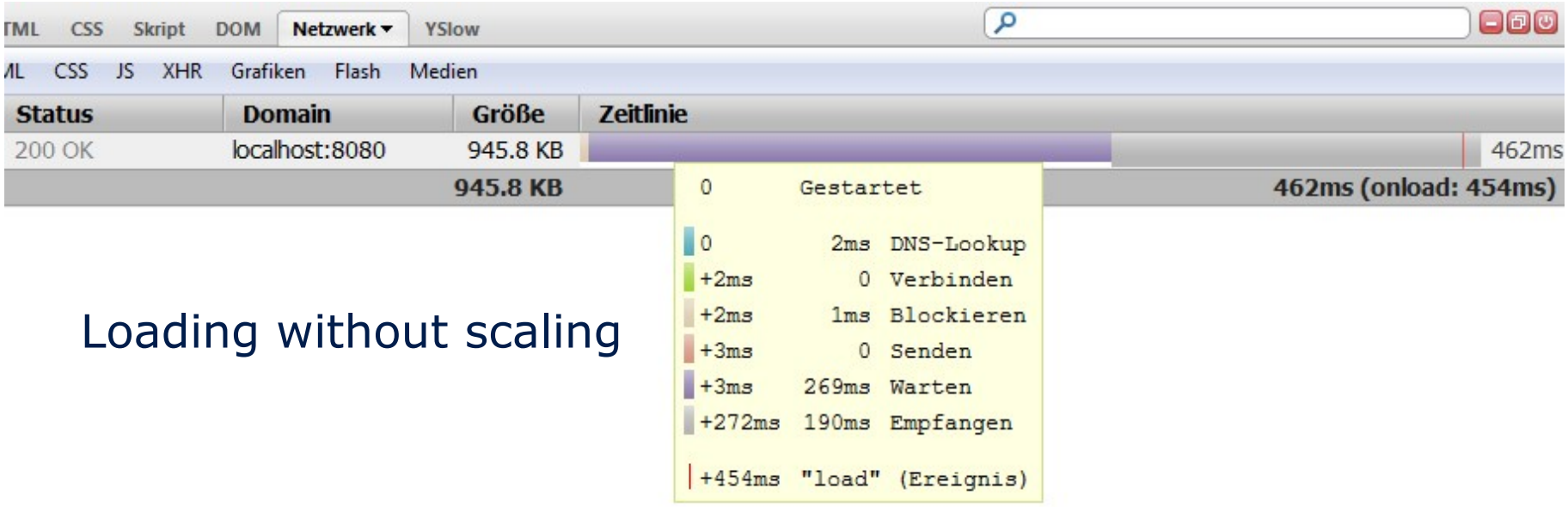
- Code changes only in the class Scaling Options

ScalingOptions

```
public enum OPTION {
    imagename, imagepath, providerid, tan, width, height,
    compression, mincompression, bandwidth, format, shrink
}

public boolean ultimateSetter(OPTION option, String value){
    switch (option){
        case imagename:
            _imageName = value;
            break;
        case imagepath:
            _imagePath = value;
            break;
        case providerid:
            _providerId = new Long(value).longValue();
            break;
        case tan:
            _tan = new Long(value).longValue();
            break;
        case width:
            _width = new Integer(value).intValue();
            break;
        case height:
            _height = new Integer(value).intValue();
            break;
        case compression:
```

- Test-environment only on one Computer
- 1st loading a picture needed 1.5 to 3 seconds
- Scaling needed 0.45 to 0.7 seconds ~ 0.5
- Loading without scaling 0.13 – 0.3 seconds ~ 0.15
- (Server delay)



Loading without scaling

Android Client

- Android Client for additional testing
- Register/change password
- Download via HTTP and SOAP