Mobile Multimedia

Networking / Server

Introduction

Networking

- Requirements:
  - File upload
  - Transmission of metadata (geo coord., desc.)
  - Lightweight client
  - Easy to implement
  - Quick setup of server
  - Flexibility and extensibility

Server

- Two possible solutions (we considered):

1. Upload via HTTP POST:
    - Easy to implement
    - Good support on Android
    - But uncomfortable in use

2. Webservice approach:
    - Very flexible
    - Support on Android not so good yet
    - Handling of binary data is tricky

Server

- Two possible solutions (we considered):

1. Upload via HTTP POST:
   - Easy to implement
   - Good support on Android
   - But uncomfortable in use
2. Webservice approach:
   - Very flexible
   - Support on Android not so good yet
   - Handling of binary data is tricky

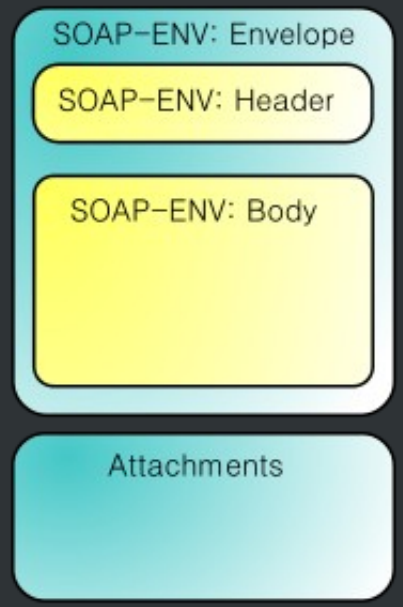
SOAP webservice

- The problem:
  - SOAP is based on XML
  - XML is good for textual information
  - But audio samples are binary data
  - So what can we do?

Binary data

- Common approaches:

1. Base64 encoding
2. SOAP with Attachments (SwA)
3. SOAP Message Transmission Optimization Mechanism (MTOM)

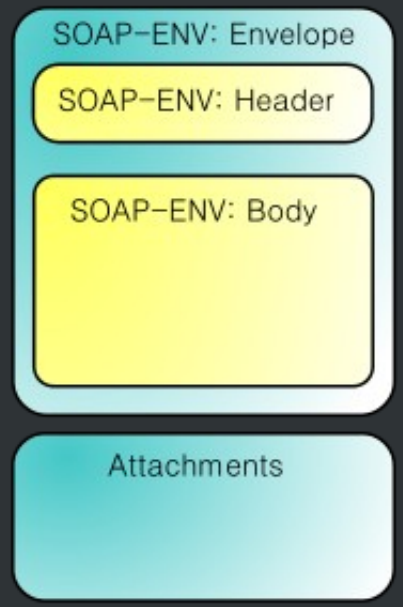


SOAP with Attachments

Binary data

- Common approaches:

1. Base64 encoding
2. SOAP with Attachments (SwA)
3. SOAP Message Transmission Optimization Mechanism (MTOM)



SOAP with Attachments

Client side

- Android does support Base64 encoding

  → Base64 class in API

- Android does also support SOAP

  → kSOAP library

- Sending requests is simple
- But no proper XML parsing
- Should be fine for us, though

Server side

- Introducing **Server One**:
  - Server prototype written in PHP
  - Requires stock Apache webserver
- Full handling of requests, i.e.
  - Base64 decoding
  - Writing file to disk
  - Storing metadata in database (PDO)
  - Respond with status code + message

Server One

- **Server One** is ready to use
- Download available on GitHub:
  - `https://github.com/nepa/Server-One`
- Repository also includes an app to demonstrate the use of SOAP webservices on Android

Thank you.

Sascha Seidel, seidels@informatik.uni-luebeck.de