Bauhaus University Weimar

Research, Technology:
AXIS, Web Service, J2ME

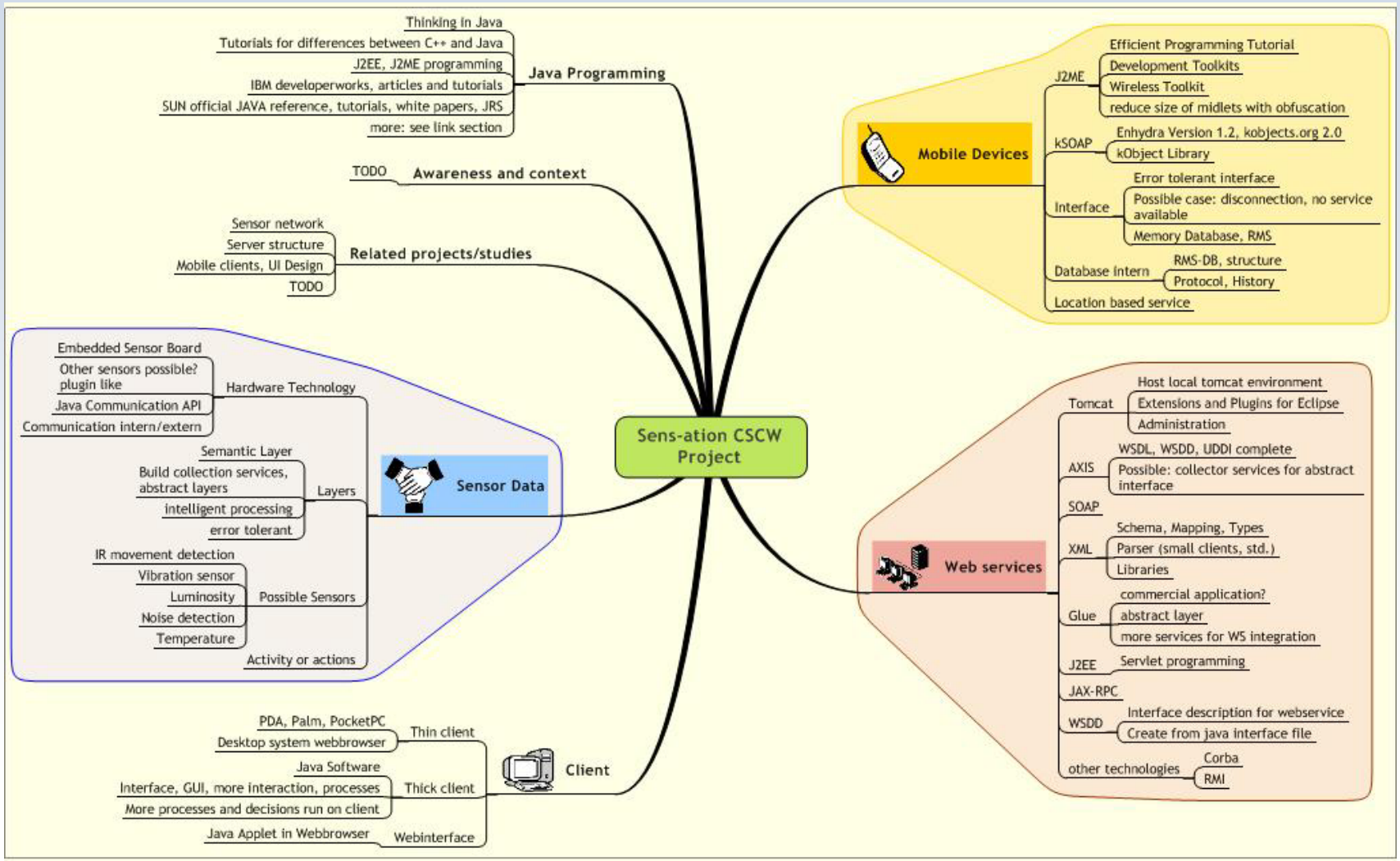
Project Sens-ation

October 2004
Nicolai Marquardt
CML Cooperative Media Lab
CSCW, Prof. Tom Gross, Tareg Egla
Bauhaus University Weimar

Outline

1. Introduction, Ideas
2. Technology: Web Service, AXIS
3. Technology: Mobile Devices, J2ME
4. First Implementation: Web Service AXIS with J2ME

1. Introduction, Mindmap



2. Technology: Web Service

Web Server Tools:

- Jakarta Tomcat, J2EE Services, JSP, Servlets
- SOAP - Simple Object Access Protocol
- AXIS
- XML Parser (Xerces)

Components:

- UDDI – Universal Description, Discovery and Integration
  - Mechanism for discovery of Web Service methods and interface
  - Dynamic and static binding
- WSDL – Web Services Description Language
  - Description of a Web Service, like the Java interface (→ tools)
  - XML format
  - WSDD – Web Service Deployment Descriptor

2. Technology: Web Service

AXIS – Apache eXtensible Interaction System

- Third Generation of SOAP Implementation
- IBM Project SOAP4J
- Higher performance, uses SAX instead of DOM
- Support for WSDL
- JAX-RPC: mapping of JAVA data types to XML scheme
- Other components: Xerces, XML-Security, JavaMail
- SOA – Service Oriented Architecture: Service Requestor (Client), Service Provider (Server), Service Broker (WSDL, Interface, UDDI)
- SSH security options, or data en-/decrypt with XML-Security

2. Technology: SOAP AXIS Message Structure

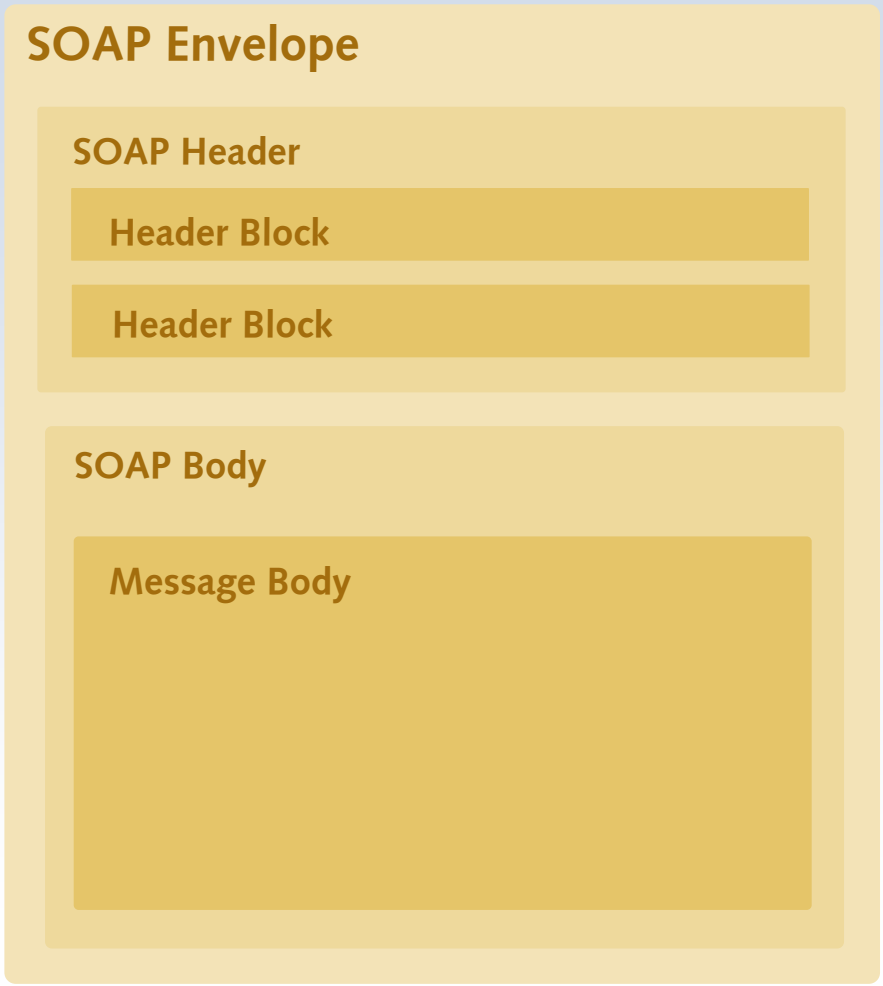


Figure based on: escala, Web Service Introduction

SOAP Envelope

- Defines what is in the message
- Who should deal with it

SOAP Header (optional)

- May contain security, transaction or routing information
- Generic capability for adding features

SOAP Body

- The main part of the message
- SOAP Request (procedure call)
- SOAP Response (the result)
- SOAP Error

2. Technology: SOAP AXIS Message Structure

Example SOAP Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
 SOAP-ENV:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>
 ...
  <SOAP-ENV:Body>
    <getCurrentPosition>
      <arg0 xsi:type="xsd:string">central</arg0>
      <arg1 xsi:type="xsd:string">detroit</arg1>
    </getCurrentPosition>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example SOAP Response:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/>
 ...
  <SOAP-ENV:Body>
    <getCurrentPositionResponse
     SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <getCurrentPositionResult xsi:type="xsd:string">123</getCurrentPositionResult>
    </getCurrentPositionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3. Technology: Mobile Devices

Development of Applications for mobile devices:

- SUN J2ME Framework, JVM on mobile clients
- CLDC, specification for hardware requirements
- MIDP 1.0, 2.0
- MIDlets, MIDlet Suite, JAR/JAD
- Limited libraries: java.io, java.lang, java.util
- Special java.microedition.io library
- No floating point operations
- Pre-verification step
- Extensions: JSR (Java Specification Request) 172: SOAP, WSDL, XML 1.0

3. Technology: Mobile Devices

- SUN WTK (current 2.1 or 2.2 beta), 2.2 with JSR 172
- WTK with memory trace, communication, garb. collect., profiler
- Development in Eclipse, Plugin available EclipseME
- Obfuscator:
  - unreadable code (against recompilation), but also smaller files
  - ProGuard (http://proguard.sourceforge.com) or RetroGuard
- Ant building process → Antenna (http://antenna.sourceforge.com)
- Devloper Kits Nokia (http://forum.nokia.com), Siemens
- kSOAP (kObjects), Version 1.2 (ksoap.enhydra.org) and 2.0 (www.kobjects.org) with faster XML parser

- Efficient J2ME Development
  → document with methods for improving J2ME development

3. Technology: Clients, Issues

Issues, Requirements, Aspects, Brainstorming:

- Connection handling (queue thread)
- Context abstractions (from UbiComp)
  - Classes of sensors
  - Interpretation, abstract, aggregation, composite pattern
  - Services, actions
  - Discoverers, share services, the UDDI service of axis
- Error/fault toleration, e.g. if no connection available
- State (see: SenSay): different transition probabilities
- GUI guidelines, function call depth, adaptable/scaling display methods
- Own database, history function, prospecting, collection (→ RMS)
- Privacy, and also security (SSH, XML-Secure)
- Possible (mobile) platforms: J2ME, .NET, Palm, PPC

3. Technology: Clients, Issues

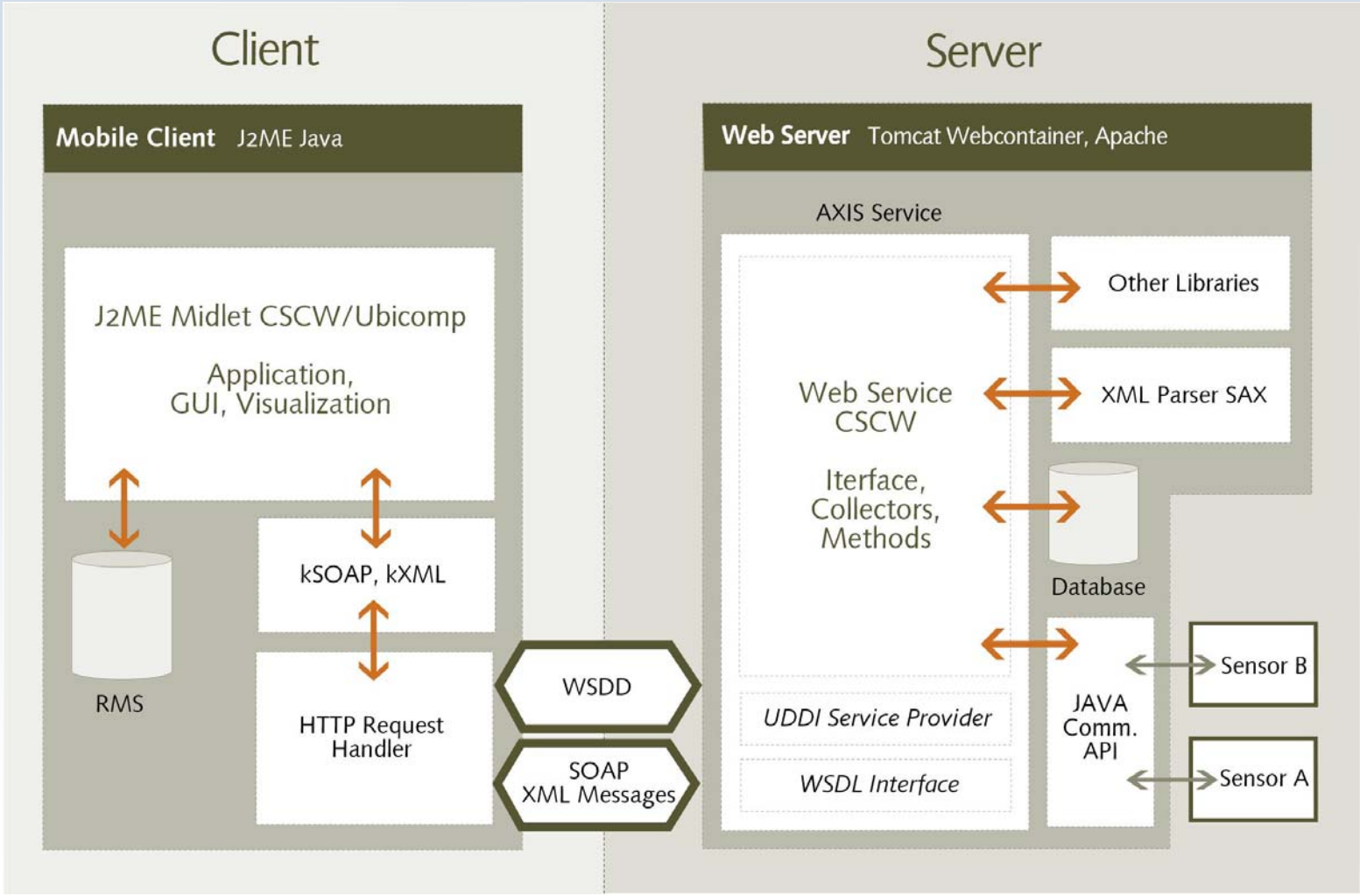
Modules of (mobile) client (see: SenSay)

- Sensor module, handling, protocol like, raw data
- Abstract layers, methods, collectors
- Display methods, view collectors, customizable
- Decision and action (if activity needed) module, if predefined pattern found, logic module

Other possible clients

- Thin client: web enabled device, HTTP, WAP, no logic elements on client
- Thick client: desktop software, many features, local data, methods
- JSP Web-Interface to Service

3. Technology: Overview



4. Test Implementation

Development process for AXIS/J2ME application:

1. Create JAVA Interface class
2. WSDL creation for Web Service, Java2WSDL tool
3. WSDL2Java is used for server and client class creation (only interface implementation)
4. Deployment with WSDD file, registration for Tomcat
5. Development of <tomcat>/webapps/AXIS/WEB-INF/classes
6. Client Development (e.g. class creation with SUN WTK)
7. Import of kSOAP libraries, use kSOAP 1.2 or 2.0
8. Build, preverify, deploy

3. Test Implementations

Software:
AXIS and J2ME

Communication example from
AXIS to JAVA MIDlet

Start tool: 

Thank You

For Your Attention!