



# 8110 – Geschäftsprozessintegration

## Übung 01

### SOAP-Grundlagen



- Übungen
  - Wo notwendig: Installation der für SOAP erforderlichen Komponenten
  - Elementare SOAP-Tests
- Ziele
  - Vorbereitung für die Projektarbeit mit SOAP



# SOAP-Komponenten



- Aufgabe
  - Installieren Sie die für SOAP erforderlichen Komponenten, je nach der von Ihnen gewählten Entwicklungsumgebung.
- Hinweise
  - GENERELL:
    - Unterstützung für WSDL und evtl. UDDI gleich mit installieren, sofern angeboten.
  - Ruby:
    - SOAP4R ist bereits installiert
  - Perl:
    - SOAP::Lite ist offenbar besonders empfohlen
  - Java:
    - Apache Axis sollte alles Notwendige enthalten...
  - PHP:
    - (Leider keine Tipps, vielleicht schon eingebaut?)



# Erste SOAP-Tests



- Hintergrund
  - RPC-artige Übung, bestehend aus Client- und Server-Teil
  - Überprüfung der Funktion der eingesetzten Werkzeuge
  - Vertraut werden mit dem jeweiligen SOAP API
  - Studieren des SOAP-Mechanismus auf Protokollebene am Beispiel
- Vorgabe für das SOAP-Binding:
  - Verwenden Sie HTTP



- **SOAP-Server**

- Implementieren Sie einen SOAP-Service "Rechner" mit folgenden Methoden
  - **add, mult:** entweder 2 Argumente oder ein Array von Elementen
  - **sub, div:** 2 Argumente
  - Datentypen: Flexibel halten! Bei Division: Stets die "double"-Variante
  - Rückgabewert: Rechenergebnis, Typ entsprechend der Übergabetypen  
Bei Array-Arg.: Summe bzw. Produkt über alle Elemente
- Erweiterungsaufgabe, für Engagierte:
  - Akzeptieren Sie auch zwei 2d-Arrays (gleicher Dimension) bei den Methoden **add** und **sub**
  - Der Rückgabewert sei dann das Summen-bzw. Differenzarray im Matrixsinn.
- Server-Realisierung: Standalone, CGI, FCGI, andere?
  - Ihre Entscheidung!



- **SOAP-Client**

- Implementieren Sie einen zum SOAP-Server "Rechner" passenden SOAP-Client
  - Fragen Sie (z.B. auf der Kommandozeile) zwei (oder mehr) Eingabeparameter ab
  - Fragen Sie nach der auszuführenden Operation (Methode), oder wenden Sie gleich alle (sinnvoller Weise möglichen) Operationen an.
  - Ermitteln Sie das Ergebnis per SOAP-Server und zeigen Sie es an
- Fehlerbehandlung
  - Situation: Division durch Null
  - Situation: Arrays unterschiedlicher Dimensionen übergeben (opt.)
  - Provozieren Sie einen Fehler und verfolgen Sie die Wirkung auf Protokollebene (HTTP-Status, "Fault"-Element und sein Inhalt).



- **Allgemeine Hinweise**

- Die Aufgabe stellt eine Variante der "Calc"-Demo aus dem Ruby-Paket "soap4r" dar.
- Es wird unterstellt, dass Sie auch im Fall anderer Implementierungssprachen auf vorhandenen Beispiel-Code zurückgreifen können und diesen "nur" zu variieren brauchen.
- Investieren Sie nicht zu viel Zeit in diese Aufgabe - es geht darum, SOAP ans Laufen zu bringen, nicht um eine möglichst perfekte Anwendung!

- **Das Folgende durchführen und beteiligte Teams protokollieren:**

- Anregung: Testen Sie Client- und Serverkomponenten team- bzw. plattformübergreifend. Kann Ihr Java-Client mit einem Ruby-Server arbeiten? Funktioniert ein Perl-Client mit einem PHP-Server?
- Vorbereitung für WSDL: Auf welche Probleme stoßen Sie bei der Verwendung von Komponenten anderer Teams?



# Weitere Vorbereitungen



- Hintergrund
  - SOAP *conversations* lassen sich sowohl im RPC- als auch im Dokumentenmodus betreiben.
  - Die SOAP-Toolkits bevorzugen oft RPC bzw. unterstützen den Dokumentenmodus nicht immer.
- Aufgabe
  - **Klären Sie für „Ihr“ SOAP-Toolkit die Verfügbarkeit des Dokumentenmodus.**
    - Falls verfügbar: Machen Sie sich mit seiner Anwendung vertraut.
    - Falls nicht verfügbar: Erarbeiten Sie eine Ersatzlösung!
  - Hinweis:
    - Man kann beliebige Daten – also auch ganze XML-Dokumente – „tunneln“, z.B. mit Strings und „base64“-Codierung...