



7363 - Web-basierte Anwendungen: **Übung 06 & 07, Projektbeginn**

SOAP-Grundlagen
WSDL-Vorbereitung
Einstieg in's Projekt



Ziele



- **Übungen**
 - Wo notwendig: Installation der für SOAP erforderlichen Komponenten
 - Elementare SOAP-Tests
 - Anmeldung bei Google (für eine spätere WSDL-Übung)
 - Projektarbeit: Design und Schema-Entwicklung für die Anmeldung
- **Ziele**
 - Vorbereitung für die Projektarbeit mit SOAP
 - Einstieg in's Projekt



SOAP-Komponenten



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



Einfache 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

 - Zusatzaufgabe, optional:
 - Akzeptieren Sie auch zwei 2d-Arrays (gleicher Dimension) bei den Methoden add und sub
 - Der Rückgabewert sei dann das Summen-bzw. Differenzarray im Matrixsinn.

 - Standalone, CGI, FCGI, andere Server-Realisierung?
 - 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 (sinnvollerweise 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).



Einfache SOAP-Tests



- 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 diesen Aufgabenteil - es geht darum, SOAP an's Laufen zu bringen, nicht um eine möglichst perfekte Anwendung!
 - Der Serverteil wird evtl. bei WSDL weiterverwendet.
- Anregung
 - Testen Sie Client- und Serverkomponenten team- und 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?



SOAP



- Abgabe
 - Diesmal keine Abgabe!
 - Erstellen Sie auf jeden Fall einen SOAP-Server und testen Sie ihn, denn er soll die Grundlage für eine spätere WSDL-Übung bilden.
 - Interessante Lösungen können auf Wunsch der Gruppe vorgestellt werden.



WSDL-Vorbereitung



WSDL-Vorbereitung



- Hintergrund
 - Die Internet-Suchmaschine Google kann auch mittels SOAP als Web Service verwendet werden - wenn auch noch im Beta-Test.
 - Die Schnittstelle wird per WSDL beschrieben.
 - Zur Nutzung mit bis zu 1000 Anfragen pro Tag (!) ist eine (kostenlose) Anmeldung bei Google erforderlich.
- Aufgabe
 - Registrieren Sie sich bei Google. Folgen Sie dazu den Anleitungen unter <http://www.google.com/apis>
 - Laden Sie das Entwicklerpaket herunter und lesen Sie die README-Datei (zunächst nur oberflächlich).
 - Installieren Sie den erhaltenen Schlüssel in der Datei `~/.google_key`
 - Sie sind damit vorbereitet für eine spätere WSDL-Übung mit dem Google-API!



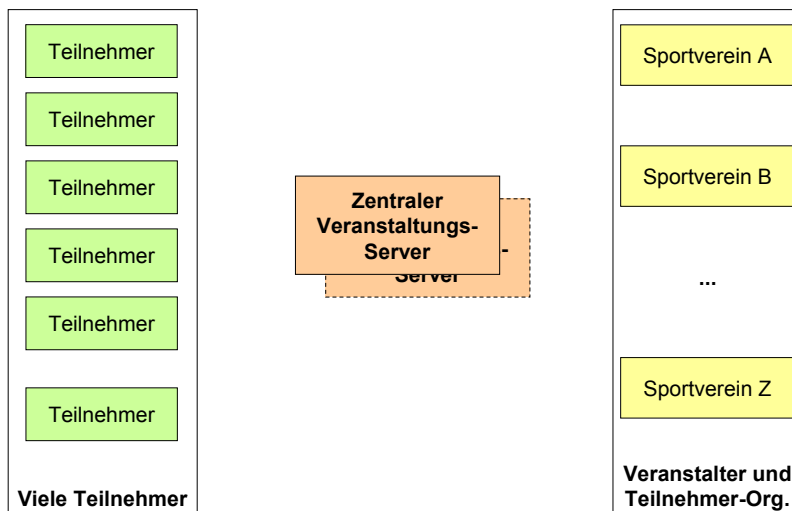
Einstieg in's Projekt



Einstieg in's Projekt



- Die Akteure





- Die Akteure
 - Teilnehmer
 - Sind (natürliche) Personen
 - Besitzen Chip-ID
 - Erhalten bei Anmeldung eine Startnummer
 - kommunizieren (nur) per Web Browser + e-Mail
 - Sind möglicherweise Mitglieder in Sportvereinen
 - Sportvereine
 - Besitzen bzw. erhalten eine Vereins-Kennung
 - Besitzen weitere Stammdaten (Anschrift, Kontaktperson, ...)
 - Verwalten ihre Mitglieder (Mitglieds-Stammdaten)
 - Betreiben eine WS-gestützte Anwendung zur Wettkampfverwaltung
 - Treten manchmal als Veranstalter auf
 - Melden oftmals Vereinsmitglieder bei Veranstaltungen an
 - Interessieren sich für die Ergebnislisten der von ihnen besuchten Veranstaltungen und speziell für die Ergebnisse ihrer Mitglieder



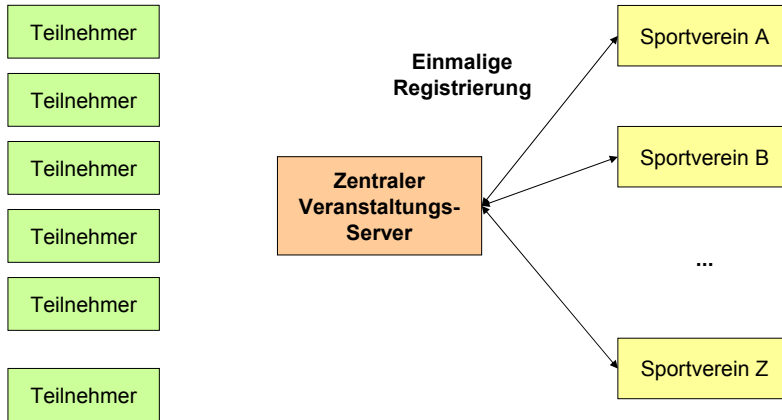
- Die Akteure
 - Zentraler Veranstaltungs-Service / -Server
 - Registriert Sportvereine (keine Einzelpersonen)
 - Registriert neue Veranstaltungen
 - Verwaltet Stammdaten und Ergebnisdaten von Veranstaltungen
 - Sammelt & speichert Ergebnislisten, auch inkrementell
 - Ermöglicht Abruf von Ergebnislisten bzw. Teilen davon
 - Unterstützt Urkundendruck und Listendruck
 - Berechnet Statistiken, stellt Auswertungsgrafiken bereit
 - Verfügt über WS-Schnittstellen (für die Anwendungen der Vereine) als auch über Web Front-ends (für die Teilnehmer)
 - Warum "zentral"?
 - Vorteile zentraler Datenhaltung und statistischer Auswertung
 - Einheitliche Kennungsvergabe für Vereine
 - Einheitliche Schnittstelle für alle Anwendungen
 - Falls mehrere "Zentralen": Auf kompatible Schnittstellen achten...



Einstieg in's Projekt



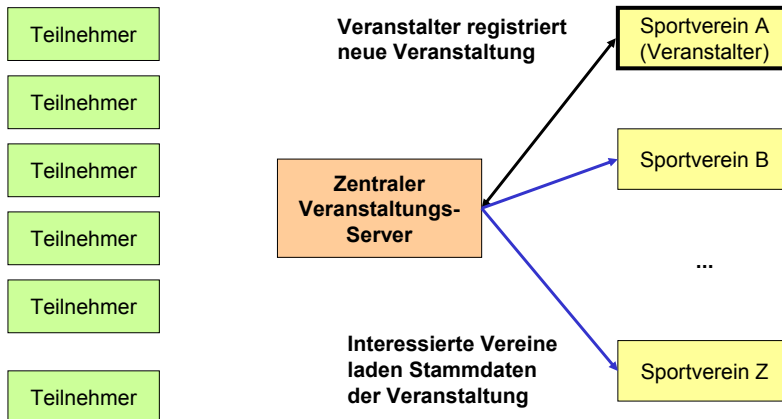
- Das Szenario: Registrierung der Vereine



Einstieg in's Projekt

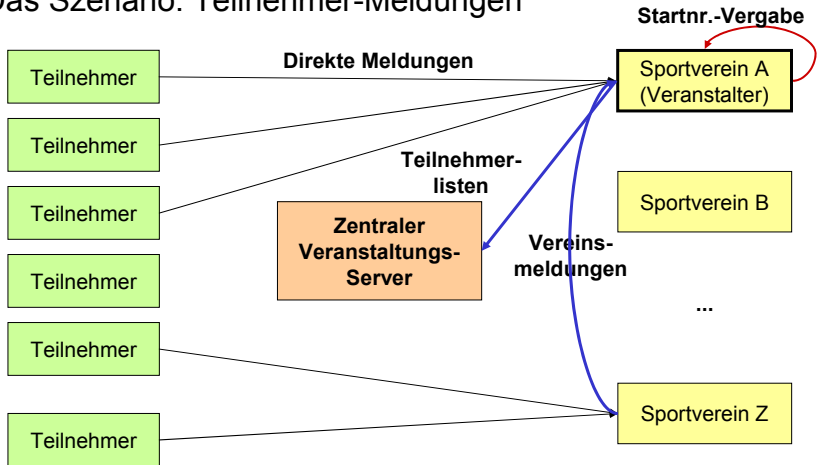


- Das Szenario: Anmeldung einer Veranstaltung

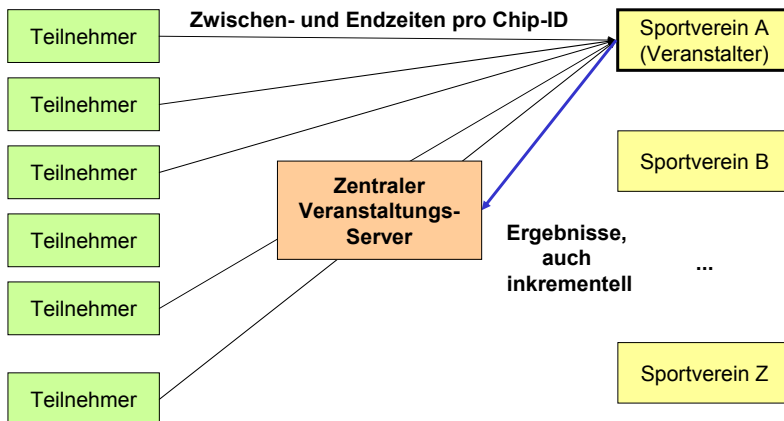




- Das Szenario: Teilnehmer-Meldungen



- Das Szenario: Wettkampf

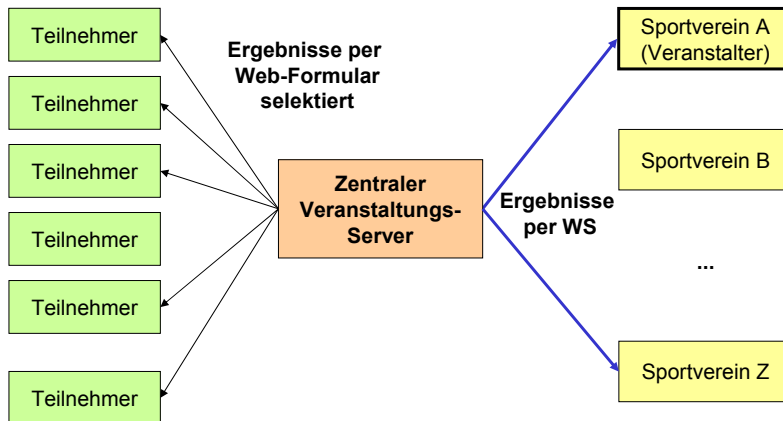




Einstieg in's Projekt



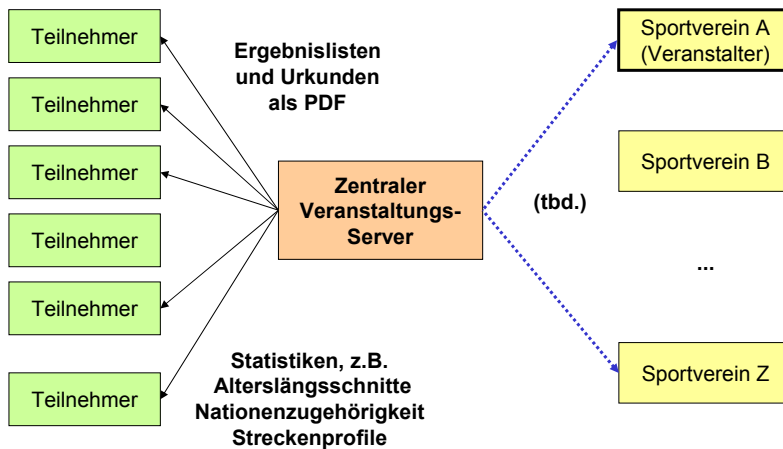
- Das Szenario: Ergebnisabruf



Einstieg in's Projekt



- Das Szenario: Weitere Auswertungen





- **Komponenten**
 - Teilnehmer
 - Nur Internetzugang, UA (Browser) mit SVG-Plugin, PDF-Reader
 - Sportvereine
 - Software zur Mitgliederverwaltung und Anmeldung
 - Hält Mitglieds-Stammdaten vor
 - WS-Interface "Vereinsmeldungen" zu Organisatoren von Veranstaltungen
 - Web-Interface für Meldungen zu Teilnahmen über Vereinsmeldung
 - WS-Interface "V-Registrierung" von Vereinen beim Veranstaltungs-Server
 - WS-Interface "Ergebnisabruf" zum Veranstaltungs-Server
 - » ruft die Ergebnisse aller zuvor gemeldeten Vereinsmitglieder ab
 - Software zur Organisation von Veranstaltungen
 - Web-Interface für Einzelmeldungen
 - WS-Interface (Receiver) für Vereinsmeldungen
 - Startnummernvergabefunktion
 - WS-Interface "Teilnehmer": Übergabe von Teilnehmerlisten an V-Server
 - WS-Interface "Ergebnisse": Übergabe von Ergebnislisten an V-Server



- **Komponenten**
 - Zentraler Veranstaltungs-Server
 - WS-Interfaces
 - Registrierung eines Sportvereins
 - Anmeldung einer Veranstaltung
 - Teilnehmerlisten
 - Ergebnislisten (Annahme)
 - Ergebnislisten (Abruf/Selektion)
 - Web-Interfaces
 - Abruf / Selektion: Registrierte Veranstaltungen
 - Abruf: Ergebnislisten
 - » nach unterschiedlichen Kriterien ausgewählt
 - » je 25 ... 50 pro Seite
 - » Option: PDF-Liste
 - Abruf: Einzel-Ergebnisse
 - » Option: PDF-Urkunde
 - Datenhaltung:
 - tbd. Tamino? Filesystem? RDBMS? Persistente Hashes?



- Aufgaben für Übung 06 und 07:
 - Mitgliederverwaltung
 - Registrierungs-Interface. Mitteilen:
 - Vereinsname, Anschrift (incl. Land), email, ... (was fehlt?)
 - Veranstaltungs-Server
 - Registrierungs-Interface. Antworten:
 - Kennung (Vereins-ID), Kennwort. Als CGI über HTTPS aufsetzen?
 - Datenmodellierung
 - "Veranstaltungsmeldung"
 - "Veranstaltungsliste" (Abrufergebnis)
 - "Teilnehmerliste"
 - "Ergebnisse" (für die Meldung der Zeiten)
 - "Ergebnisanfrage"
 - "Ergebnisliste" (für die ausführlichen Abrufe der Ergebnisse)



- Datenmodellierung, "Veranstaltungsmeldung"
 - Kopfdaten
 - Name der Veranstaltung
 - Ort und Datum der Veranstaltung (auch Zeitraum möglich)
 - Organisator (ein registrierter Verein, evtl. per Referenz)
 - Liste der ausgetragenen Wettkämpfe
 - Wettkampf
 - Typ:
 - Etwa Hauptlauf, Skater, Rollis, Handbiker, Power-Walker, Bambini
 - Wertungen:
 - ggf. Frauen / Männer getrennt
 - Altersklassen (z.B. MJA, MH, M30, ..., M75; WJA, WH, W30, ..., W75)
 - Sonderwertungen:
 - Vereinsmeisterschaften
 - Dt. Meisterschaften etc.
 - Mannschaftswertungen (je 3 oder 4 eines Vereins, m/w/mixed)
 - (s. Forts.)



- Datenmodellierung, "Veranstaltungsmeldung"
 - Wettkampf (Forts.)
 - Datum
 - Startzeit
 - Ortsangabe Start (Freitext)
 - Streckenlänge (in km)
 - Liste der Zwischenzeit-Punkte
 - gemeint ist eine km-Angabe
 - Zeitmessung bei Start und Ziel hier nicht angeben.

- Datenmodellierung, "Veranstaltungsliste"
 - Liste von "Veranstaltungsmeldung"-Elementen
 - (Selektierbar über Zeitraum und/oder Ortsangabe)



- Datenmodellierung, "Teilnehmerliste"
 - Veranstaltungs- und Wettkampf-Schlüssel
 - Vereine
 - Verein
 - Name
 - Anschrift
 - email
 - Kennung
 - Teilnehmer
 - Name, Vorname, MI
 - Anschrift, email
 - Jahrgang
 - AK (Altersklasse, falls ableitbar)
 - Verein (Kennung / Referenz)
 - Chip-ID
 - Startnummer



Einstieg in's Projekt



- Datenmodellierung, "Ergebnisse"
- Datenmodellierung, "Ergebnisanfrage"
- Datenmodellierung, "Ergebnisliste"

Diese werden auf jeden Fall gebraucht. Wir klären nächste Woche, ob diese Schemata schon zur nächsten Abgabe zählen.

- **Allgemeine Hinweise**
 - Wie immer bei der Erstellung von XML Schemata:
 - Legen Sie pro Schema einen geeigneten Namensraum fest!
 - Dieser Namensraum sollte in den SOAP-Dokumenten später auch benutzt werden.
 - Modellieren Sie möglichst präzise, z.B. unter Verwendung der Schema-Datentypen oder auch eigenen.
 - Tipp: Installieren Sie Ihre Schemadatei unter dem Namensraum-URI



Abgaben



- Abzugeben sind die Schema-Dateien
 - Dateinamen: Veranstaltungsmeldung.xsd, usw.
 - Die genaue Liste wird nächste Woche installiert.
- Die SOAP-Übungen werden nicht eingesammelt
 - Ihre SOAP-basierenden Projektarbeiten geben Ihnen ausreichend Gelegenheit, den Umgang mit SOAP zu demonstrieren.
- Abgabeschluss:
 - Do, der 20. Mai 2004, 12 Uhr Meilenstein 3
- Bereits bis kommende Woche zu erledigen:
 - Ihre Anmeldung bei Google!



Zur Wiederholung

Die Folien zum Projekt aus der
Einführungsvorlesung



Szenario: Wettkampf-Server (Olympia!)



- Erfassung / Meldung
 - Kopfdaten (der Veranstaltung selbst)
 - Name, Ausrichter, Termin, Ort, Länge, DLV-Kz, Bem., ...
 - ggf.: Verschiedene Wertungen (Wettkampf in Wettk.)
 - ID der Veranstaltung: Aus URL ableiten, namespace-artig
 - Meldedaten
 - Vereine (Stammdaten?), Teilnehmer (Chip-ID!, Startnr), Mannschaften
 - Datensätze
 - Startnr + Zwischenzeiten + Endzeit
 - Zeiten ergeben automat. Platzierung; Dok-Reihenf. bei Gleichheit
 - Evtl. Trennung zwischen Anmeldung und Ergebnismeldung; Korr. vs. Neu
 - Entwurf geeigneter Doctypes!



Szenario: Wettkampf-Server



- Datenhaltung
 - Einfach: Ein (großes) XML-Dokument pro Wettkampf
 - Aufwändiger: Datenbank-Lösung
 - Eventuell TAMINO der Software AG !
 - In beiden Fällen:
 - Anbindung per Web Service!
 - Übung: *Document upload* mit SOAP
 - Client:
 - Einfach: Web Front-end, Upload per http & CGI-Script
 - Allgemein: Eigene Anwendung, SOAP-Client



Szenario: Wettkampf-Server



- Datenabruf
 - Normale Ergebnislisten (HTML, PDF)
 - Lange Tabellen, PDF-Mehrseiter!
 - Einzelne Urkunden (PDF)
 - Gestaltungsaspekte wie Logo, Siegel etc.
 - Mit Einzelzeiten. Auch Mannschafts-Urkunden?
 - Statistiken (SVG)
 - Histogramme: % Finisher unter x1 / x2 / ... / xn Minuten
 - Laufprofil: Wer "brach ein"?



- Dokumentation des Servers
 - Formal: WSDL für jede Schnittstelle
 - Zum Lesen: Docbook mit
 - Design
 - Schnittstellen
 - Anleitung für Daten-Lieferanten (incl. DTD-Erläuterungen)
 - Web-Frontends
 - Beispielen
 - FAQ
 - ...



- Komponenten
 - XML Schemata für die Teildokumente
 - WS-Client zum Anliefern der Daten
 - WS-Server (DB-Client)
 - DB-Server (wirklich?)
 - Web-Frontend + WS-Client für Abfragen
 - WS-Client für Abfragen