



# 7363 - Web-basierte Anwendungen: **Übung 07, Projektbeginn**

WSDL-Vorbereitung  
Einstieg ins Projekt



- Übungen
  - Wo notwendig: Installation der für WSDL erforderlichen Komponenten
  - Elementare WSDL-Tests
  - Projektarbeit: Design und Schema-Entwicklung für die Anmeldung
- Ziele
  - Vorbereitung für die Projektarbeit mit WSDL
  - Einstieg ins Projekt



# WSDL-Tests



- Hintergrund
  - Der Studienbereich Informatik betreibt seit dem Sommer 2007 eine kleine Wetterstation auf dem Dach des C-Gebäudes
  - Diese Wetterstation stellt ihre Daten per KNX-Standard bereit
  - Auf einem virtuellen Server steht für diesen Kurs ein SOAP-basierter Web Service zur Verfügung, mit dem Sie Wetterdaten abrufen können
  - Die Schnittstelle wird per WSDL beschrieben



- **Aufgabe**

- Generieren Sie eine einfache Client-Anwendung für diese Wetterstation mittels der WSDL-Datei `~werntges/1v/wba/06/wetterdaten.wsdl`
- Führen Sie einige manuelle Suchanfragen für Testzwecke durch!
  - Schneiden Sie dabei den HTTP-Datenstrom mit (tcpmon) und überprüfen Sie den Zusammenhang zwischen WSDL-Beschreibung und Aufbau der SOAP Body-Elemente
- Nach Beendigung dieser Tests: Erzeugen Sie eine Ausgabetabelle (eine Zeile pro Minute) für die Dauer von 10 Minuten mit den Spalten
  - Uhrzeit
  - Temperatur
  - Windgeschwindigkeit
  - Lichtstärke
  - Regen
  - Windalarm
- Geben Sie die Zahlenwerte incl. Einheiten aus (Boolean-Typen: ja/nein)  
*Sie sollten dies bei der Abnahme des letzten Meilensteins vorführen können!*



- **Tipp**

- Verwenden Sie `wsdl2ruby.rb` (Ruby-Anwender) oder ein vergleichbares Werkzeug aus Ihrem SOAP/WSDL-Entwicklerpaket, um einen Anwendungsrumpf (Client) zu generieren, und erzeugen Sie mit dessen Hilfe die Tabelle

- **Hinweise**

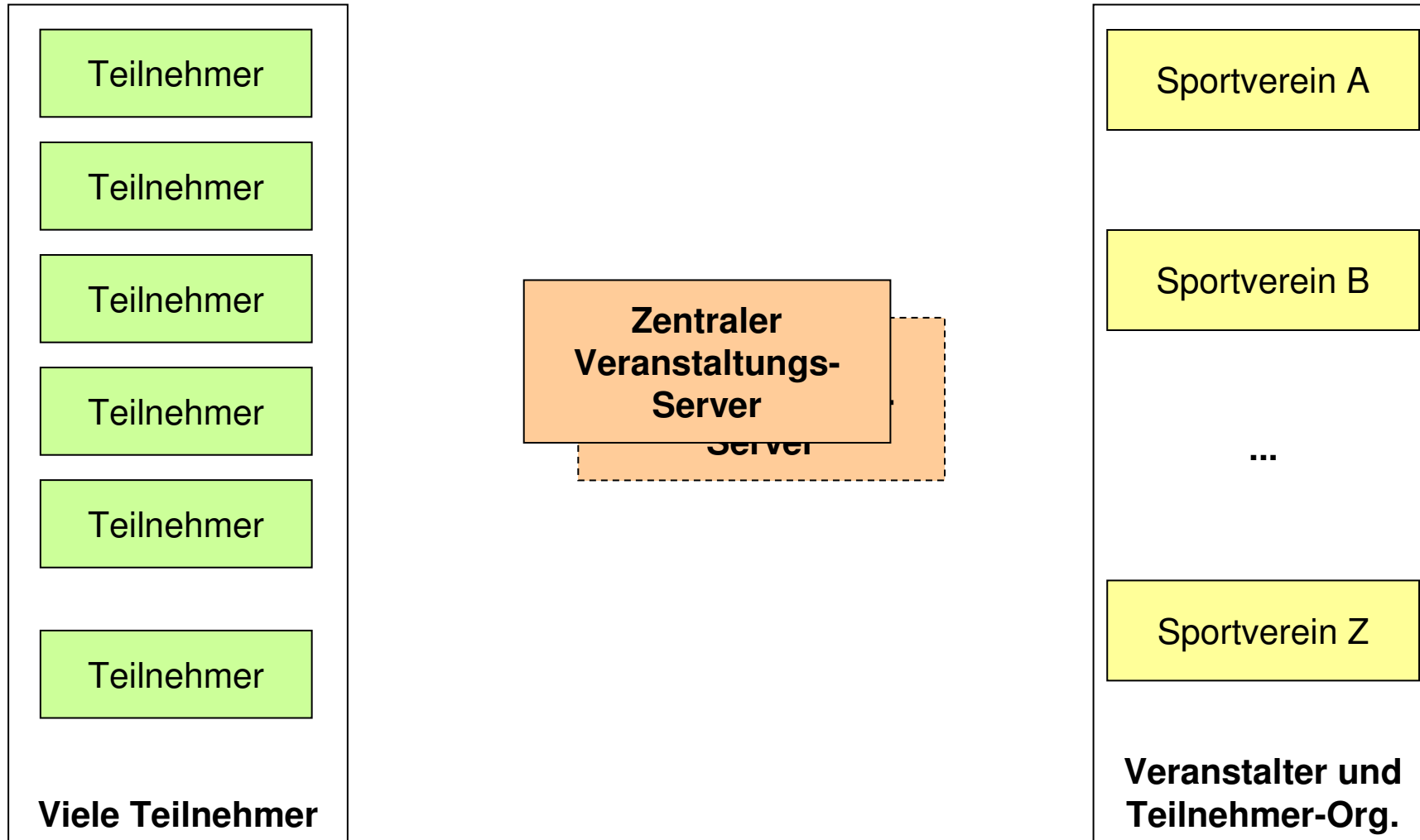
- Der Web Service steht noch nicht permanent zur Verfügung. Vorläufig ist nur am Donnerstag (und ohne weitere Vereinbarung auch nur am 6.12.07) der Betrieb des Servers gesichert
- Der Server ist nur FH-intern erreichbar – also bitte die Pool-Rechner verwenden (Zugriff via WLAN erfolgt ohne Gewähr)
- Diese kleine Übung ist daher **am 6.12.07 während der Präsenzzeit durchzuführen!**



# Einstieg ins Projekt



- Die Akteure







- Die Akteure

- Teilnehmer

- sind (natürliche) Personen
    - besitzen jeweils eine 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 als Teilnehmer 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"?

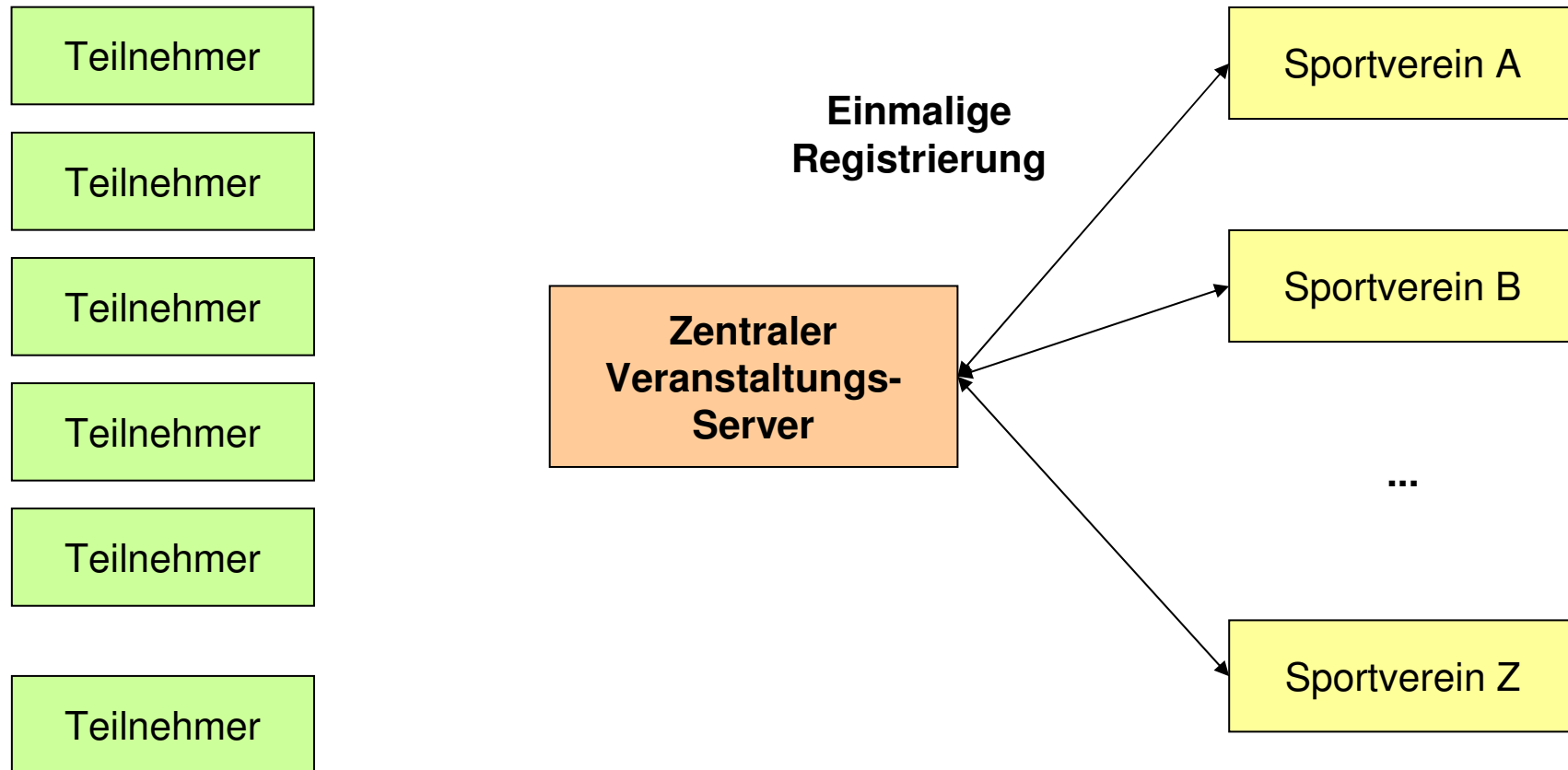
- Vorteile zentraler Datenhaltung und statistischer Auswertung
    - Einheitliche Kennungsvergabe für Vereine
    - Einheitliche Schnittstelle für alle Anwendungen
    - Falls mehrere "Zentralen": Auf kompatible Schnittstellen achten...



# Einstieg ins Projekt

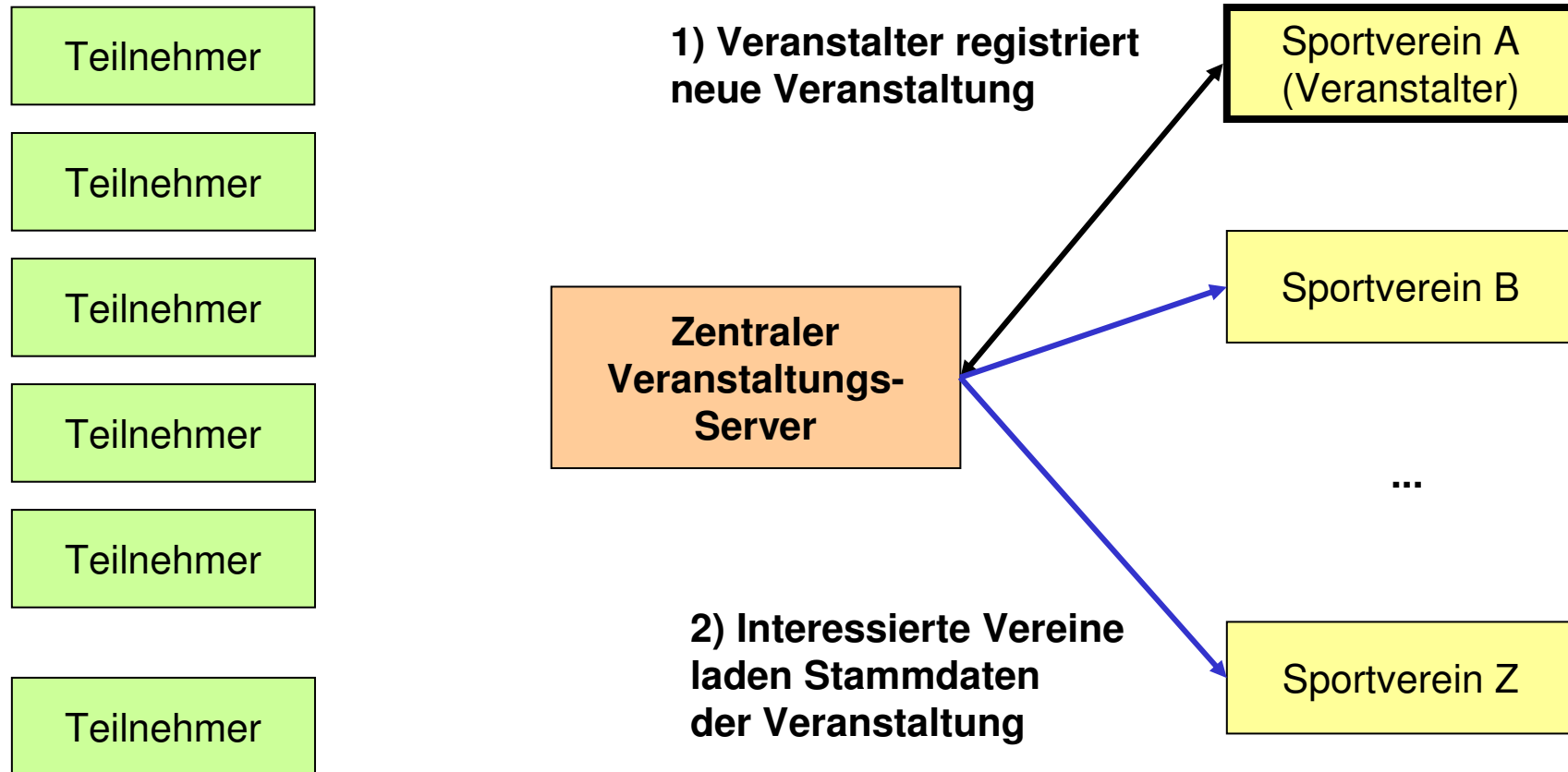


- Das Szenario: Registrierung der Vereine



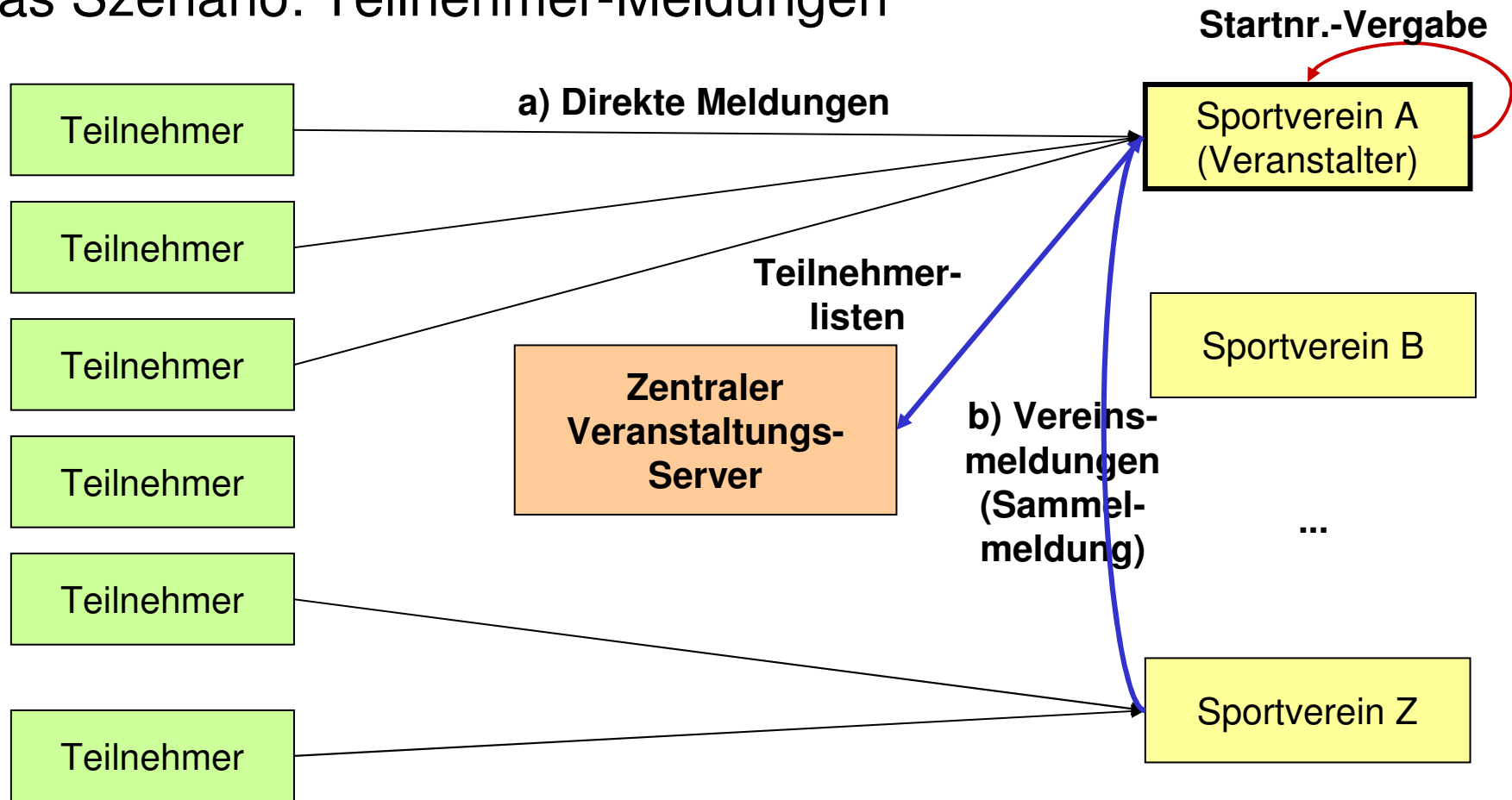


- Das Szenario: Ausschreibung einer Veranstaltung



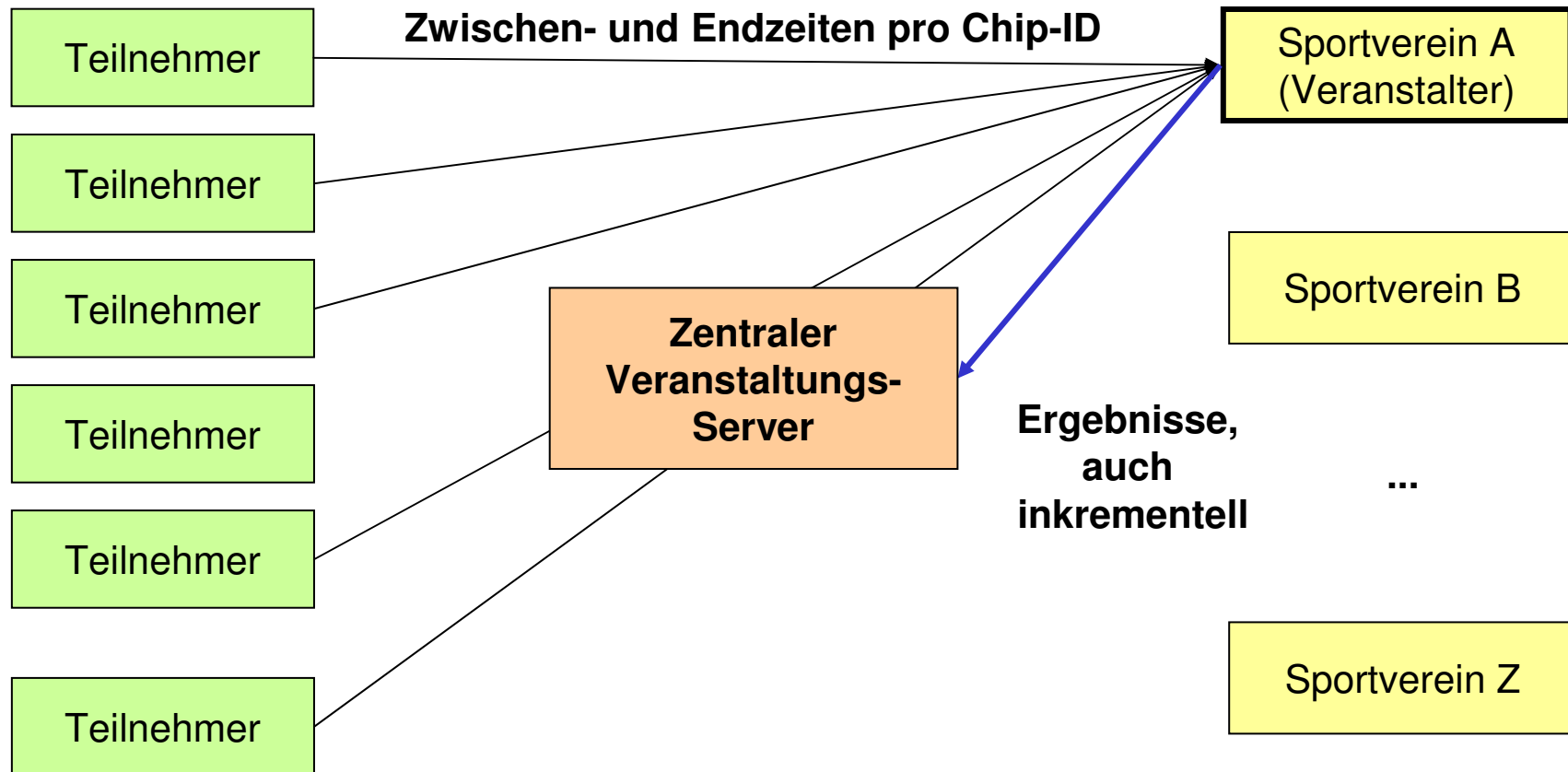


- Das Szenario: Teilnehmer-Meldungen



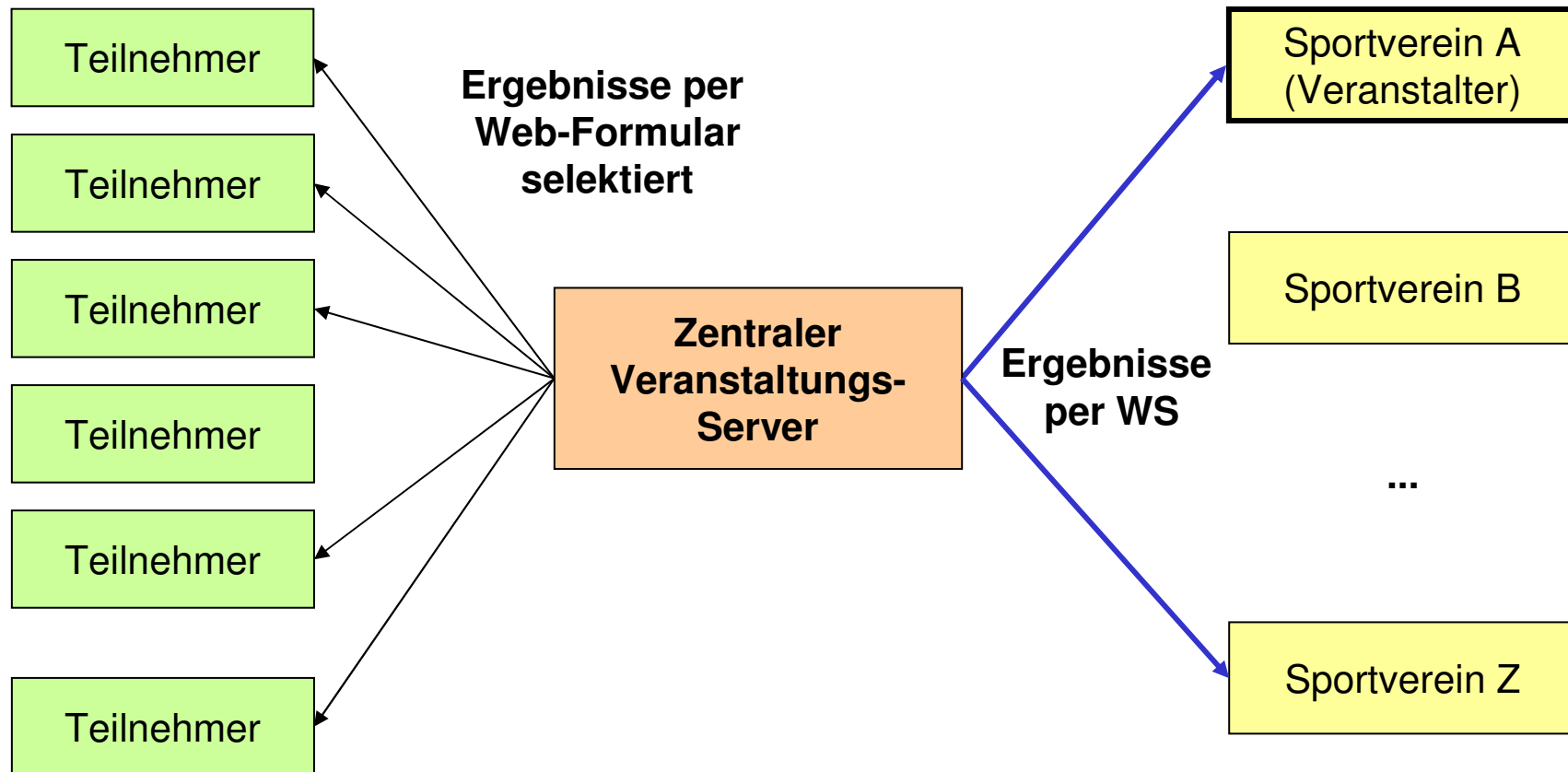


- Das Szenario: Wettkampf (-Simulation)



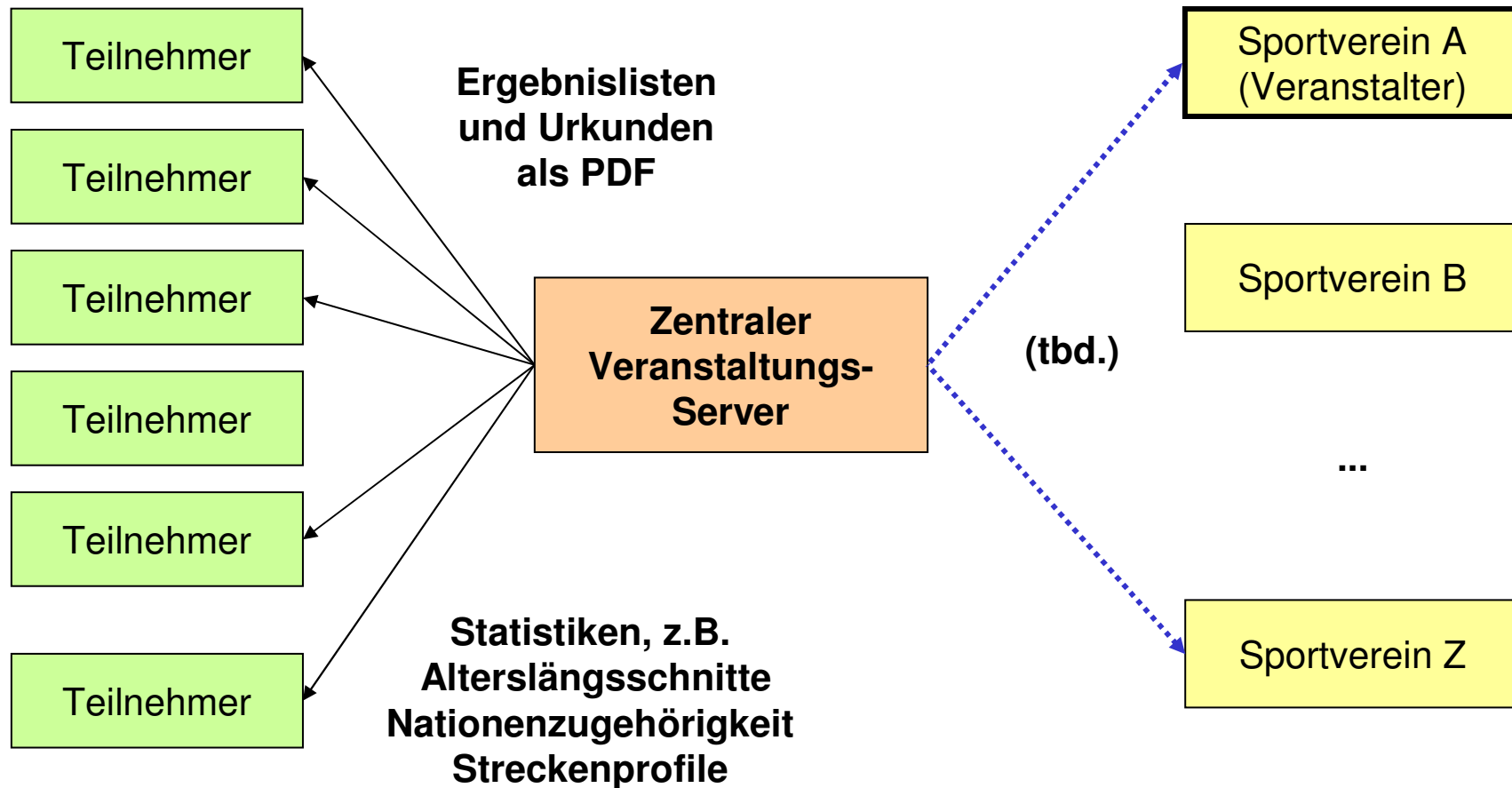


- Das Szenario: Ergebnisabruf





- Das Szenario: Weitere Auswertungen







- Komponenten der Akteure
  - Teilnehmer
    - Nur Internetzugang, UA (Browser) mit SVG-Unterstützung/-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
      - (Mess-) Ergebnisse (Zwischen- und Endzeiten)
      - Ergebnislisten (Abruf/Selektion)
    - Web-Interfaces
      - Abruf / Selektion: Registrierte Veranstaltungen
      - Abruf: Ergebnislisten
        - » nach unterschiedlichen Kriterien ausgewählt, je 25 ... 50 pro Seite
        - » PL: PDF-Liste, via XSLT & XSL-FO
      - Abruf: Einzel-Ergebnisse (PL: PDF-Urkunde, via XSLT & XSL-FO)
      - Abruf: Histogramm (SVG) - % Finisher vs. Endzeit, integriert oder differenziert
    - Datenhaltung:
      - tbd. Filesystem? Persistente Hashes? RDBMS? ORM/Rails? XML/Tamino?!
      - Angebot: 3 SQL-DBs pro Gruppe auf einem FH-Server
      - Alternative: SQLite nehmen ...



- Grundregel
  - Die **Anwendungen** des Veranstalters, des zentralen Servers und ggf. eines dritten Vereins sind **streng voneinander zu trennen!**
- Idealfall
  - Die Komponenten dieser drei Anwendungen laufen auf drei verschiedenen Rechnern
- Etwas realistischer:
  - Falls DB: Nutzung eines gemeinsamen RDBMS ist in Ordnung, solange verschiedene Datenbanken (nicht nur Tabellen) verwendet werden
  - Der zentrale Server und die Anwendungen des Veranstalters sollten über unterschiedliche Webserver erreicht werden (z.B.: Gleicher Host, aber über verschiedene Ports betrieben)



# Projekt: Erstes Etappenziel

XML-Schemabeschreibungen



- Aufgaben für Übung 07:  
**Datenmodellierung mittels XML Schema für die WS-Schnittstellen**
  - Mitgliederverwaltung
    - (1) Registrierungs-Interface, *consumer*-Seite. Mitteilen:
      - Vereinsname, Anschrift (incl. Land), E-Mail, ... (was fehlt?)
  - Veranstaltungs-Server
    - (1) Registrierungs-Interface, *provider*-Seite. Antworten:
      - Kennung (Vereins-ID), Kennwort. Als CGI über HTTPS aufsetzen?
  - Datenmodellierung
    - (2) „**Veranstaltungsmeldung**“ (Veranstalter an Server)
    - (3) „Teilnehmerliste“ (Veranstalter an Server)
    - „Veranstaltungsliste“ (Abrufergebnis nach Anfrage, Server an Verein)
    - „Ergebnisse“ (für die Meldung der Zeiten, Veranstalter an Server)
    - „Ergebnisanfrage“ (Verein an Server)
    - „Ergebnisliste“ (Server an Verein, 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 (mehrere!)
    - Typ:
      - Etwa Hauptlauf, Skater, Rollis, Handbiker, Power-Walker, Bambini
    - Wertungen:
      - ggf. Frauen / Männer getrennt bzw. egal: M, W, X
      - Altersklassen (z.B. MJA, MHK, M30, ..., M75; WJA, WHK, W30, ..., W75)
    - Sonderwertungen (optional):
      - 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-Messpunkte
      - Gemeint sind km-Angaben
      - Die Pflicht-Zeitmessungen bei Start und Ziel hier nicht angeben.
      - Beispiel Marathon: 5, 10, 15, 20, 21.1, 25, 30, 35, 40
- Datenmodellierung, "Veranstaltungsliste"
  - Liste von "Veranstaltungsmeldung"-Elementen
    - (Selektierbar über Zeitraum und/oder Ortsangabe)



- Datenmodellierung, „Teilnehmerliste“
  - Veranstaltungs- und Wettkampf-Schlüssel
  - Vereine
    - Verein
      - Name
      - Anschrift
      - E-Mail
      - Kennung
  - Teilnehmer
    - Name, Vorname, MI, Geschlecht (M, W)
    - Anschrift, E-Mail
    - Jahrgang oder Geburtstag
    - Altersklasse AK, vom Veranstalter aus Geschlecht und Jahrgang bzw. Geburtstag abzuleiten
    - Verein (Kennung / Referenz)
    - Chip-ID
    - Startnummer (vom Veranstalter zuzuweisen)





- **Abzugeben**
  - Je Team eine XML Schema-Datei **veranstaltungsmeldung.xsd**, die exemplarisch den XML-Dokumententyp einer **Veranstaltungsanmeldung** möglichst präzise beschreibt.
- Hintergrund
  - Die Veranstaltungsanmeldung ist der komplexeste Fall und soll daher hier sorgfältig modelliert werden. Analoge Überlegungen werden Sie auch bei den anderen Schnittstellen anstellen – aber ohne Abgabe.
  - Das XML-Schema muss später nicht unbedingt weiterverwendet werden, es läge aber nahe, das Schema zum Teil einer WSDL-Datei zu machen.
- Lernziele
  - Auffrischung der Datenmodellierung mit XML Schema
  - Schaffung solider Grundlagen für das Projekt, Vermeidung von Missverständnissen
- Tipp: Diskutieren Sie die Schnittstellen und Anforderungen rege!
  - Jetzt erkannte Schwachstellen ersparen Ihnen später viel Mühe...
  - **Verständnisfragen zu den Spezifikationen sollten \*jetzt\* geklärt werden, nicht erst in ein paar Wochen!**