



7363 - Web-basierte Anwendungen ***4750 – Web-Engineering***

Eine Vertiefungsveranstaltung



Projektvorstellung



- **Das Szenario**

- **Situation**

- Es gibt viele Sportvereine. Diese benötigen Software zur Mitgliederverwaltung.
 - Es gibt viele Sportveranstaltungen. Für deren Abwicklung und Auswertung benötigen die Veranstalter Software.
 - Durch Zentralisierung dieser Aufgaben mittels Web-basierter Anwendungen benötigen die Vereine keine eigene Software mehr.
 - Einheitliche Datenstandards und zentrale Datenhaltung ermöglichen effiziente Abwicklung und statistische Auswertungen vieler Wettkämpfe
 - Die Ankopplung von Zeitmessungsanlagen ergänzt das Szenario um Aspekte des Datenaustauschs zwischen verteilten Anwendungen

- **Ziel**

- **Ihre Aufgabe besteht in der Entwicklung dieser Anwendungen**

- **Veranstaltungsarten für das Projekt**

- Langlauf-Wettkämpfe mit vielen Teilnehmern, insbesondere Stadtmarathons



- **Die Akteure**

- **Vereine (bzw. Beauftragte)**

- Pflegen die Stammdaten ihres Vereins
 - Können alle Mitgliedsdaten ihres Vereins pflegen
 - Wählen Wettkämpfe als Angebote für ihre Mitglieder aus
 - Senden Sammelanmeldungen für ihre Mitglieder

- **Veranstalter (einige der Vereine)**

- Können Veranstaltungen einpflegen / ausschreiben
 - Bearbeiten Meldedaten zu ihren Veranstaltungen
 - Senden Wettkampfergebnisse (hier: Zeitmessungen) während der Wettkämpfe



- **Die Akteure**

- **Anonyme Besucher**

- Können ausgeschriebene Veranstaltungen einsehen
 - Besitzen Leserechte für Ergebnislisten und statistische Auswertungen

- **Teilnehmer ohne Verein**

- Können sich für ausgeschriebene Wettkämpfe anmelden

- **Vereinsmitglieder**

- Gelangen auf ihre Vereins-Seiten
 - Können ihre Stammdaten pflegen
 - Können sich für die dort gelisteten Wettkämpfe vormerken lassen
 - Werden ggf. per Sammelmeldung vom Verein gemeldet



- **Die Software**

- **Zentraler Veranstaltungs-Server**

- verwendet Login der Vereine
 - registriert neue Veranstaltungen; CRUD
 - Wettkämpfe sind als „line items“ Teile von Veranstaltungen
 - verwaltet Stammdaten von Veranstaltungen
 - sammelt & speichert Ergebnis-Messdaten, auch inkrementell
 - generiert Ergebnislisten bzw. Teile davon
 - [unterstützt Urkundendruck und Listendruck (PDF-Erzeugung)]

- Add-ons für LV 7363 (WBA, Diplom-Studiengang)**

- berechnet SVG-Statistiken, stellt Auswertungsgrafiken bereit
 - Integriert Kartenmaterial für Routenplanung/Anfahrtbeschreibungen bzw. für die Laufstrecke



- **Die Software**

- **Verwaltungs-Server für Sportvereine**

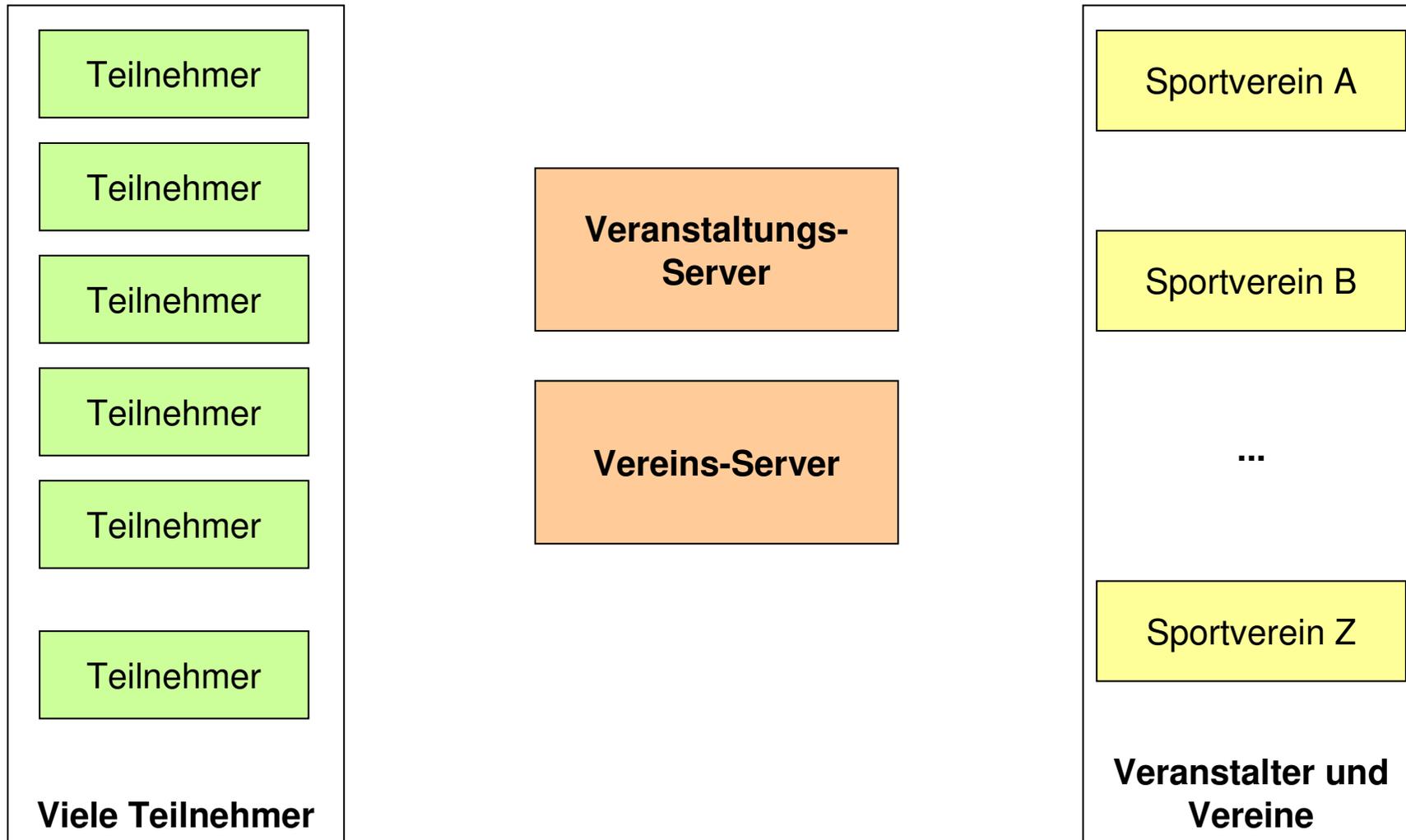
- registriert Sportvereine (keine Einzelpersonen); CRUD
 - verwaltet Stammdaten von Vereinen
 - verwaltet Stammdaten von Vereinsmitgliedern
 - bietet den Mitgliedern ausgewählte Wettkämpfe zur Teilnahme an
 - generiert und überträgt Sammel-Meldungen an den Veranstaltungs-Server

- **Wettkampf-Simulator**

- Anstelle eines realen Wettkampfes mit vielen Teilnehmern werden wir einen simulieren
 - Die Simulations-Software wird für ca. 1000 Läufer Zeitmessungen an mehreren Wegpunkten sowie an der Start- und Ziellinie generieren und (im Zeitraffer) an den Veranstaltungsserver senden



- Die Akteure

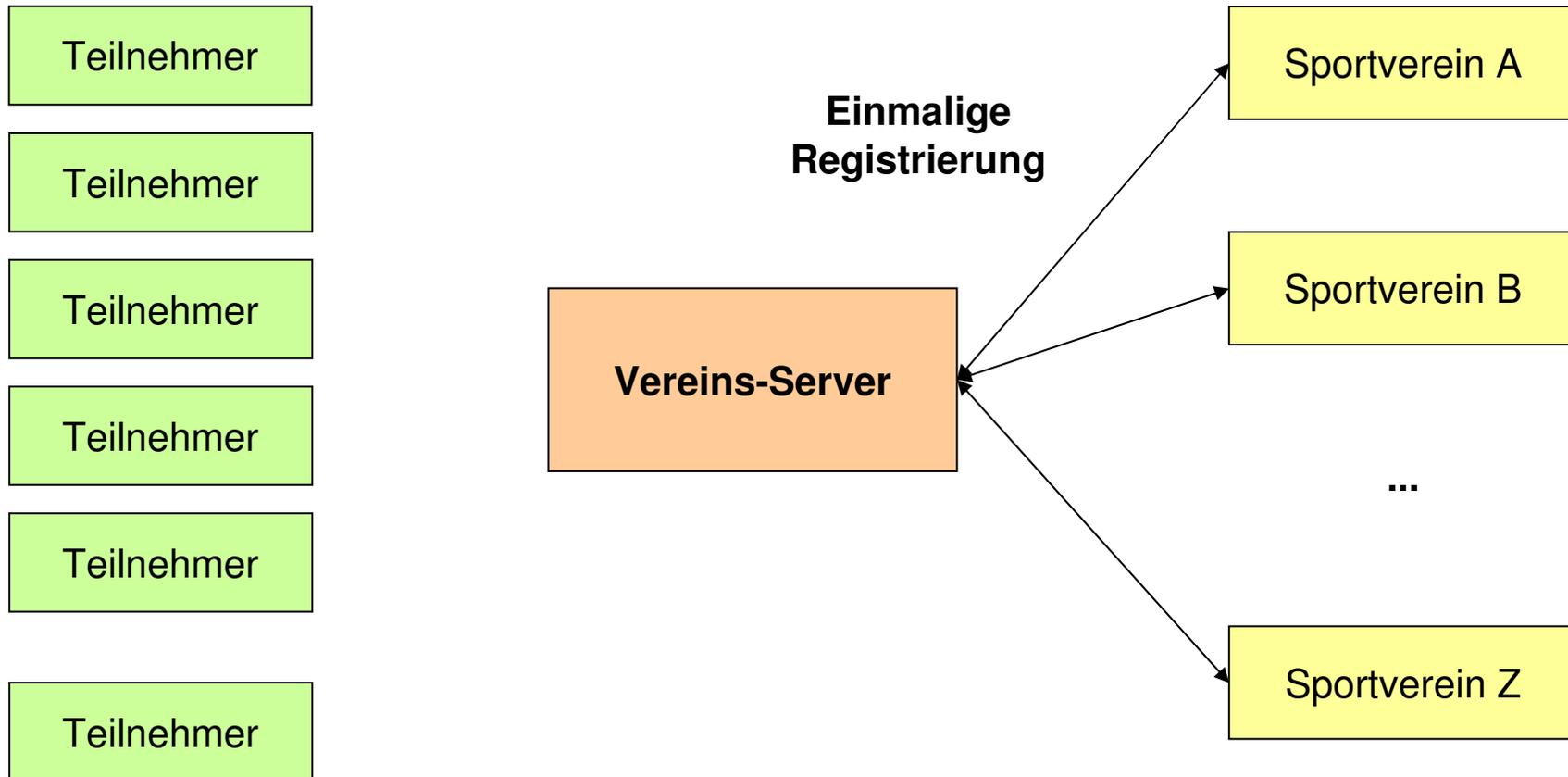




Einstieg ins Projekt

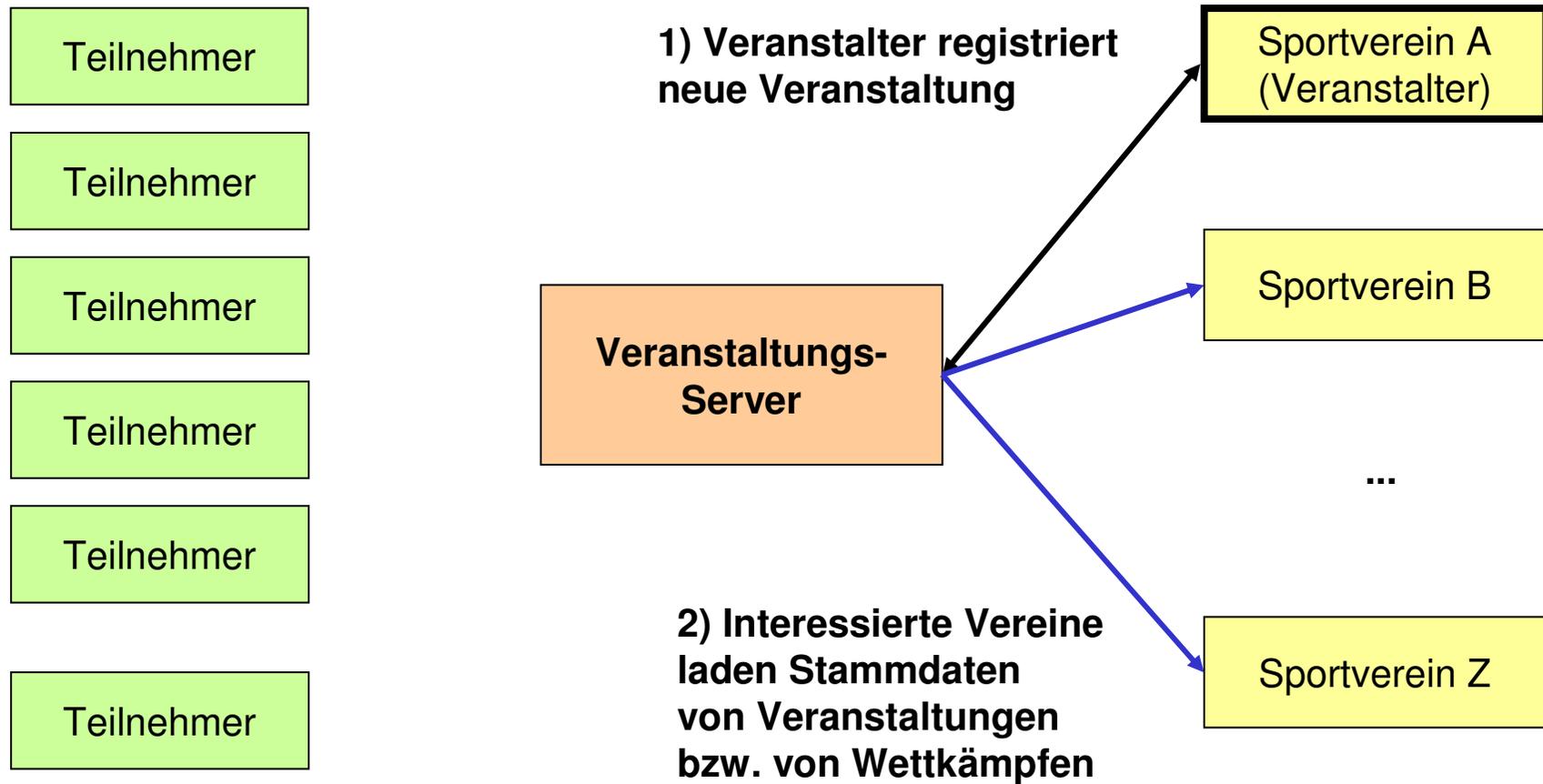


- Das Szenario: Registrierung der Vereine



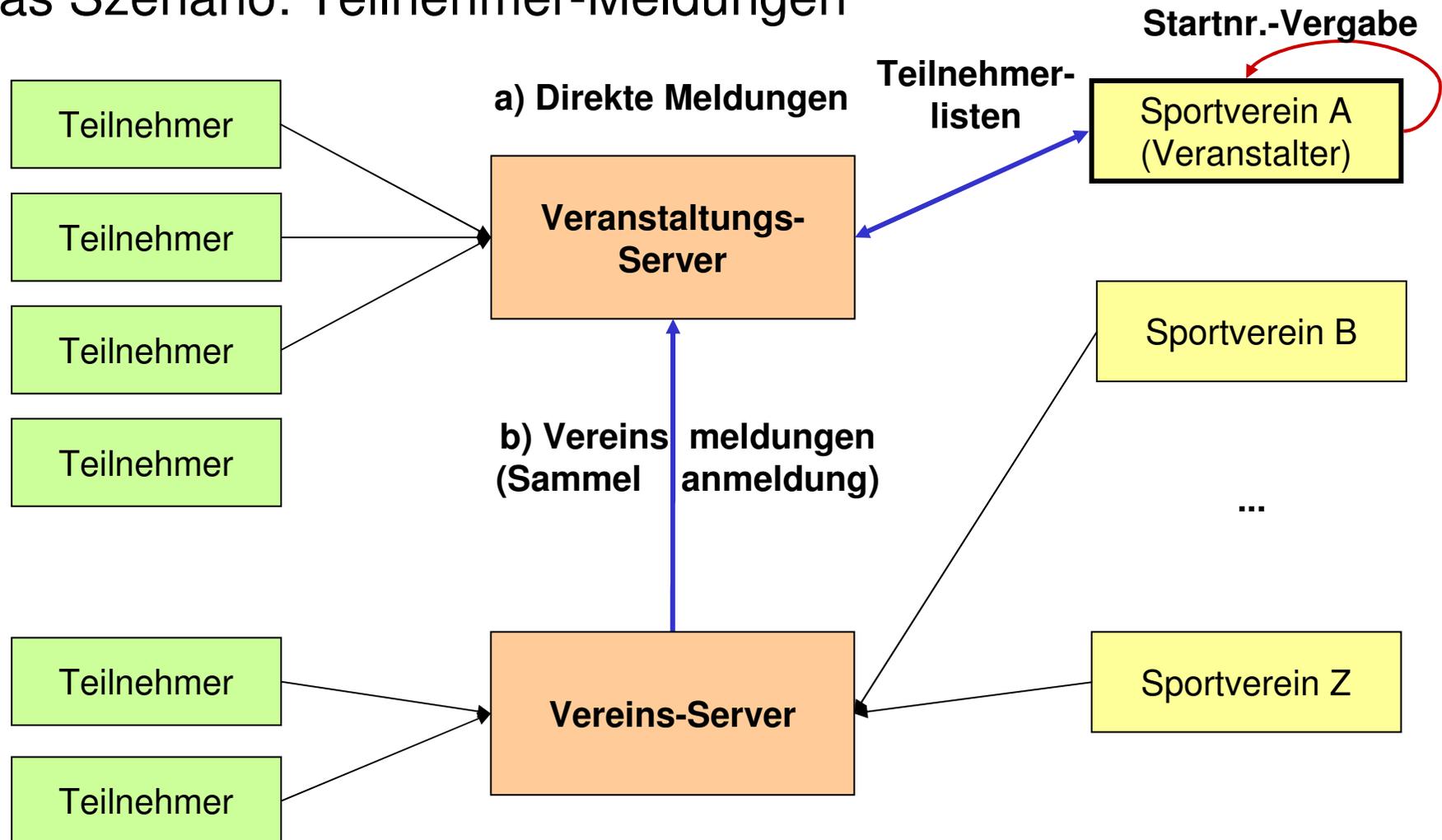


- Das Szenario: Ausschreibung einer Veranstaltung



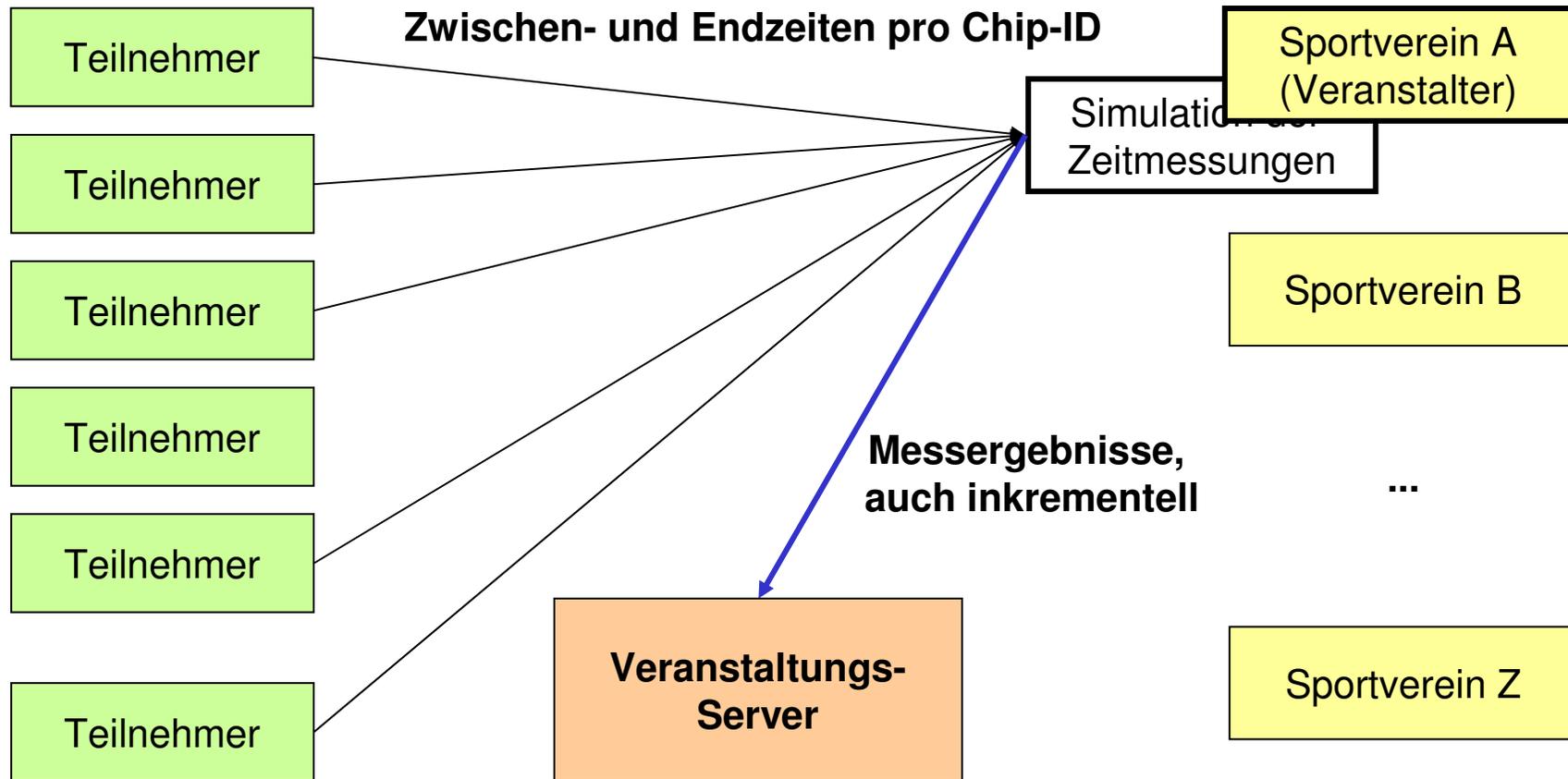


- Das Szenario: Teilnehmer-Meldungen



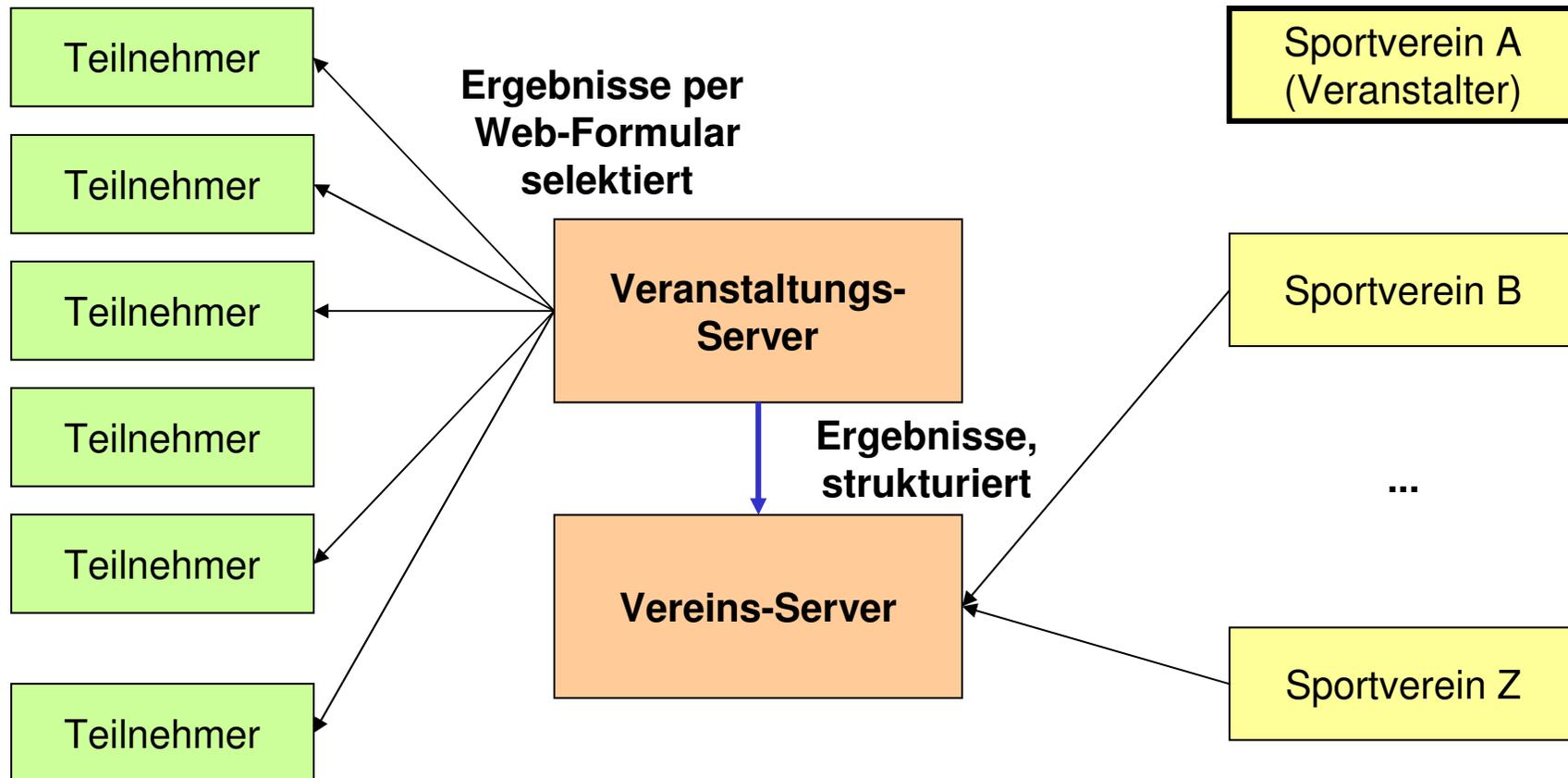


- Das Szenario: Wettkampf (-Simulation)



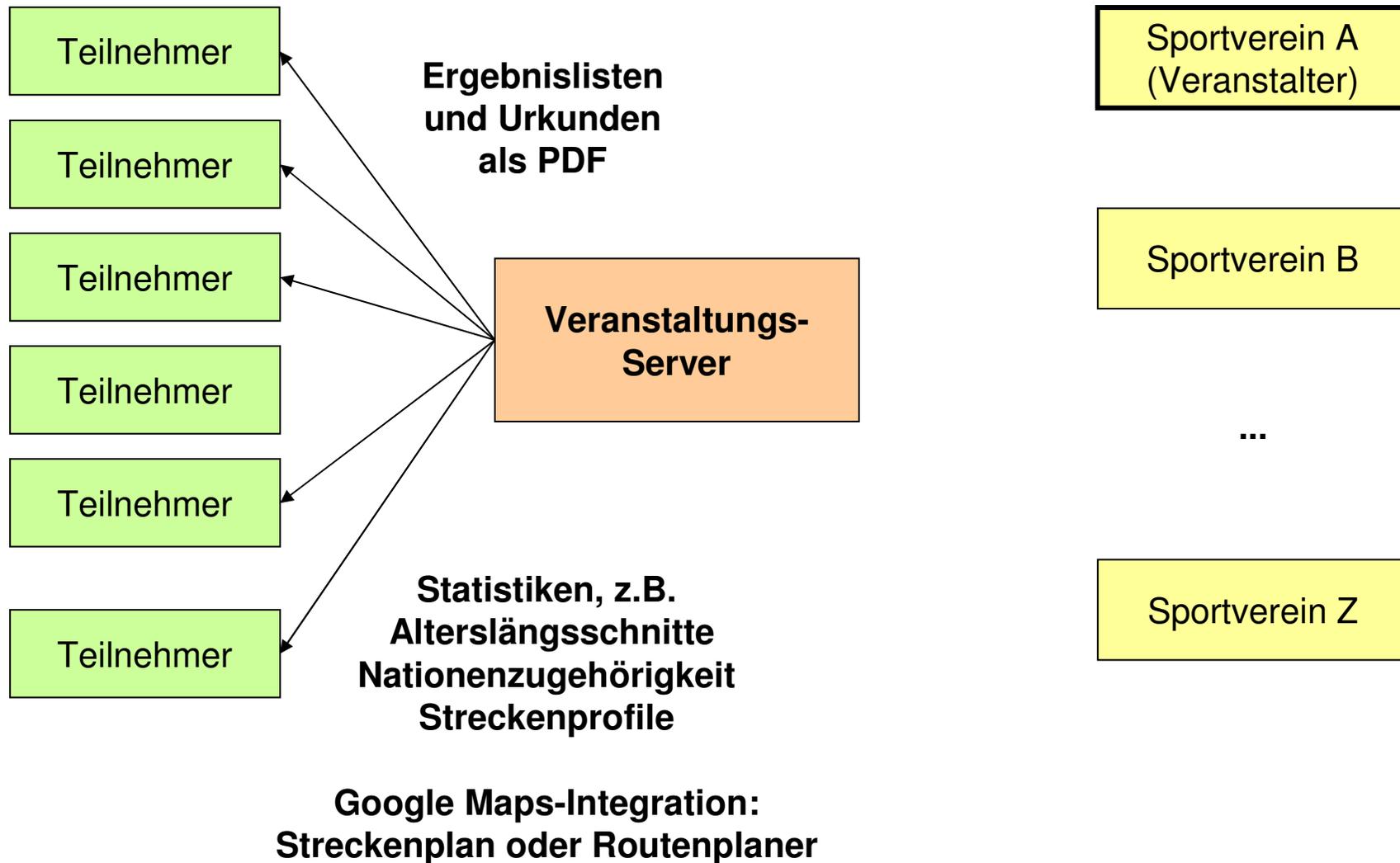


- Das Szenario: Ergebnisabruf





- Das Szenario: Weitere Auswertungen





- **Der Veranstalter**

- **Ein besonderer Sportverein**

- Er schreibt eine Veranstaltung aus, d.h. er legt ein entsprechendes Objekt mit allen abhängigen Objekten beim zentralen Server an
 - Er bearbeitet „seine“ Anmeldungen durch Vergabe von Startnummern und organisatorische Detail, hier vertreten durch „hat bezahlt“ und „hat Starterset erhalten“
 - Er führt die Wettkämpfe durch und meldet die Messergebnisse (Nettozeiten)

- **Benötigt**

- Vereins-Login
 - Browser zum Anlegen/Bearbeiten von Veranstaltungen
 - Browser zum Bearbeiten von Meldedaten
 - REST-Client für Massen-Updates von Meldedaten
 - Wettkampf-Simulator mit REST-Client zur Erzeugung und Übermittlung von Zwischenzeiten



- **Sportvereine**

- **Vereins-Stammdaten**

- Modellieren, aber nicht unbedingt mit Controller/View versehen; *Initial load* in der DB genügt
- Registrierung beim zentralen Server, dort Kennung/Passwort abholen & merken

- **Mitgliedsverwaltung**

- CRUD für Mitglieder, per Browser zu bedienen
- Mitgliedsnummer = Kennung für spätere Aktivitäten
- Passwort per „Zufallsgenerator“ erzeugen, Kennung & Passwort per E-Mail übermitteln (→ ActionMailer)

- **Wettkampf-Verwaltung**

- Anzeige ausgeschriebener Wettkämpfe (nicht: Veranstaltungen!)
- Möglichkeit der Teilnahme für Mitglieder (einfaches Web-Interface)
- Nach Ende der Ausschreibungsfrist: Sammelanmeldung an zentralen Server übermitteln
- Nach Wettkampf-Ende: Möglichkeit, die Ergebnisse *nur der gemeldeten* Mitglieder beim zentralen Server abzurufen & tabellarisch anzuzeigen



- 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 Stammdaten (Anschrift, Kontaktperson, ...)
 - verwalten ihre Mitglieder (Mitglieds-Stammdaten)
 - 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**

- **Vereins-Server**

- registriert Sportvereine (keine Einzelpersonen)
 - registriert neue Vereinsmitglieder
 - verwaltet Stammdaten des jew. Sportvereins
 - verwaltet Stammdaten der Mitglieder
 - organisiert Sammelanmeldungen



- **Die Akteure**

- **Zentraler Veranstaltungs-Server**

- registriert neue Veranstaltungen
 - verwaltet Stammdaten und Ergebnisdaten von Veranstaltungen
 - sammelt & speichert Messergebnislisten, auch inkrementell
 - ermöglicht Abruf von Ergebnislisten bzw. Teilen davon
 - unterstützt Urkundendruck und Listendruck
 - berechnet Statistiken, stellt Auswertungsgrafiken bereit



- Grundregel
 - Die **drei Anwendungen** „Vereins-Server“, „Veranstaltungs-Server“ und „Wettkampf-Simulation“ sind **streng voneinander zu trennen!**
- Idealfall
 - Die 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)
- Anwendungen koppeln?
 - Vorgesehen: Per REST
 - Auf Wunsch möglich: Per SOAP (z.B. wer dies beruflich bereits braucht)



Projekt: Erstes Etappenziel

Datenmodellierung



- Datenmodellierung, „Veranstaltung“
 - Kopfdaten
 - Name der Veranstaltung
 - Ort und Datum der Veranstaltung (auch Zeitraum möglich)
 - Organisator (ein registrierter Verein, per Referenz)
 - Liste der ausgetragenen Wettkämpfe, s.u.
 - Positionsdaten: 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, i.d.R. auslassen!):
 - Vereinsmeisterschaften
 - Dt. Meisterschaften etc.
 - [Mannschaftswertungen (je 3 oder 4 eines Vereins, m/w/mixed)]
 - (s. Forts.)



- Datenmodellierung, „Veranstaltung“
 - 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
 - Für LV 7363: Grundlage für spätere Geo-Informationen
 - Geographische Koordinaten aller Punkte mit Zeitmessungen incl. Start & Ziel
 - Alternativ: Routenpunkte der Laufstrecke
- Bitte beachten:
 - Veranstaltung enthält Wettkämpfe
 - Wettkampf enthält Messpunkte, Altersklassen/Wertungen, opt. Sonderwertungen



- Datenmodellierung, „Teilnehmer“
 - Teilnehmer
 - Name, Vorname, Geschlecht (M, W)
 - Anschrift
 - E-Mail
 - Chip-ID
 - Geburtstag
 - Altersklasse (AK)
 - vom Server aus Geschlecht und Geburtstag abzuleiten
 - Verein (Referenz)
 - Wettkampf (Referenz)
 - Startnummer (vom Veranstalter zuzuweisen)
 - Bitte beachten
 - Ein Teilnehmer kann an mehreren Wettkämpfen teilnehmen, auch innerhalb einer Veranstaltung (Bsp.: Skater- & Hauptlauf am nächsten Tag)



- Datenmodellierung, „Teilnehmer“ vs. „Läufer“
 - Einpflegen von „Teilnehmern“ ergibt Redundanzen bei Sportlern, die sich für mehrere Wettkämpfe anmelden.
 - Strategie zur Vermeidung:
 - „Sportler“ oder „Läufer“ mit rein personenbezogenen Daten
 - „Teilnehmer“ enthält Referenzen zum Wettkampf, zum Sportler, optional zum Verein (in einem anderen Wettkampf könnte er ja für einen anderen Verein antreten), sowie die Datenfelder „Altersklasse“ und „Startnummer“, welche nur im Kontext eines konkreten Wettkampfs ermittelt werden können
 - „Teilnehmer“ wirkt dann nebenbei wie ein *join table*



- **Etappenziel 1**

- Überlegen Sie sich möglichst bis zur kommenden Woche, welche Modelle / Klassen / Tabellen Sie für die bisher vorgestellten Kernfunktionen in den beiden Rails-Projekten „Veranstaltungsserver“ und „Vereinsserver“ benötigen. **„Verteidigen“ Sie Ihr Modell auf Nachfrage!**
- Stellen Sie Beziehungen (1:1, 1:n, m:n) zwischen diesen Modellen her (Vorbereitung auf spätere Umsetzung mit Deklarationen wie „belongs_to“)
- Ermitteln Sie, wo eventuell *join tables* benötigt werden
- Bestimmen Sie die Modelle, die per Browser bedient werden sollen bzw. müssen, also für die Sie komplette „scaffolds“ generieren lassen könnten. Unterscheiden Sie die Fälle, für die ein „generate model ...“ genügen wird.

Von der ersten Code-Generierung trennt Sie jetzt nur noch die Festlegung der Tabellenspalten bzw. der Modell-Attribute. Stellen Sie diese zusammen (incl. Datentypen) und prüfen Sie kritisch, ob Informationen später fehlen könnten oder redundant angelegt werden.

Tipp: Denken Sie über Ihre Modelle besonders lange und gründlich nach!



- **Erinnerung**

- Auch wenn in den bisherigen Texten deutsche Begriffe (z.B. „Teilnehmer“) verwendet wurden, heißt das nicht, dass Sie diese innerhalb Ihrer Projekte so übernehmen sollten
- Im Gegenteil: Verwenden Sie **englische** Bezeichnungen für alle Rails-Objekte, um die Rails-Namenskonventionen ohne mühsames Einpflegen von Ausnahmeregeln für die Bildung von Plural- und Singularformen nutzen zu können!
- Suchen Sie nach möglichst kurzen, prägnanten Namen für Klassen, Attribute und Methoden
 - Der Code wird im Idealfall kompakt und fast so gut lesbar wie natürliche Sprache, insbesondere erklärt er sich selbst
 - Merke: *Ungeschickte Namenswahl ist der Anfang vieler schlechter Projekte!*