

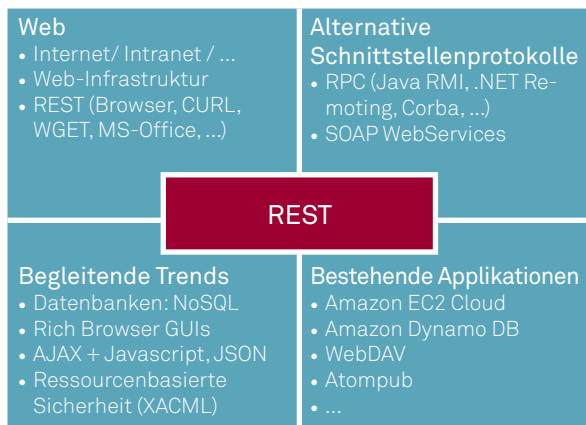
REST

Der Architekturansatz Representational State Transfer (REST)

REST ist ein Architekturstil für die Strukturierung und Umsetzung von System-Schnittstellen mit Hilfe von HTTP. Er basiert auf einer Abbildung der Informationsobjekte und Funktionen auf Ressourcen, deren Adressierung und Verlinkung über URLs und der Abbildung auf unterschiedliche Repräsentationen. Es etabliert sich als leichtgewichtige Request/Response-basierte Kommunikationsart in Client/Server-Architekturen, insbesondere bei der Überbrückung unterschiedlicher Technologie-Plattformen.

Definition

Der „RESTful“ Architekturstil ist geprägt von den folgenden zentralen Prinzipien:

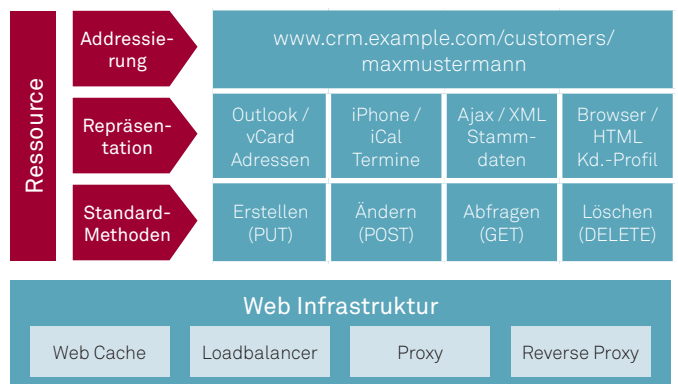


Ressourcenzentriertheit: Ressourcen sind die zentralen Informationseinheiten. Sie definieren Informationsobjekte, aber auch Prozesse und Dienste.

Jede Ressource hat eine Adresse: Jede Ressource hat eine eindeutige Adresse (→ URL, URI). Dies ist die Voraussetzung dafür, die Ressourcen über Verlinkung untereinander in Beziehung zu setzen.

Repräsentationen: Die Repräsentation legt fest, wie eine Ressource dargestellt wird. Hier ist man nicht auf eine Ausprägung festgelegt. Die Repräsentation wird vom Abfragenden und Anbieter zum Zeitpunkt der Anfrage ausgehandelt.

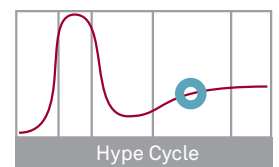
Standard-Methoden: HTTP als Protokoll für RESTful Anwendungen definiert vier Standard-Methoden (Verben), die semantisch genau definiert sind. Die Aufrufe der Business-Logik werden auf Ressourcen und diesen Verben abgebildet.



Zusammengefasst in einem Gesamtbild, würde zum Beispiel die Ressource Kunde über die URL `crm.example.com/customers/<Kundennummer>` erreichbar sein. Beim Auslesen der Kundeninformationen über das HTTP-Verb GET können die Daten über unterschiedliche Formate bereitgestellt werden. Der Anfragende kann zum Beispiel aus vCard, für die Verwendung in MS Outlook, oder XML wählen.

Reifegrad

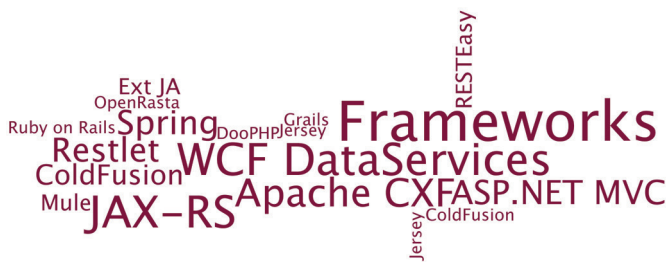
HTTP als Standard, die Web-Infrastruktur sowie die HTTP-Bibliotheken für alle Plattformen sind ausgereift und vollständig



produktiv nutzbar. Spezielle REST-Unterstützung in Form von Frameworks ist aktuell am Entstehen. Vor allem bezüglich nicht-funktionaler Anforderungen ist noch Weiterentwicklung notwendig.

Marktübersicht

REST Frameworks unterstützen bei der Implementierung von REST-basierten Service Providern und Service Consumern.



Alternativen

Eine Alternative zum REST-Architekturstil ist eine Service-Orientierte Architektur (SOA) mit SOAP-Webservices. SOA wird im Gegensatz zu REST mit einem hohem Maß an Formalismus und einem meist sehr breit gefächerten Architektur-Stack (→ WS-*) verbunden. Beide Ansätze haben ihre Berechtigung – SOA ist typischerweise besser geeignet in Umgebungen mit speziellen Anforderungen an Service-Orchestrierung und -Choreographie, sowie bei Server/Server-Integration. Der RESTful-Ansatz ist besser geeignet in einfacheren Anwendungsfällen und Client/Server-Szenarien.

Referenzszenario

Eine REST-basierte Schnittstelle kann zum Beispiel eingesetzt werden, um Create-, Read-, Update- und Delete-(CRUD) Methoden für Entitäten bereitzustellen, die durch eine Anwendung verwaltet werden. Werden zusätzlich auf Grund von hohen Aufrufzahlen

Load-Balancing und Caching benötigt, spricht dies zusätzlich für eine RESTful Lösung.

Ein weiterer Einsatz wäre eine Schnittstelle eines CRM-Systems, das Kundeninformationen in unterschiedlichen Formaten bereitstellen soll. Denkbare Formate wären vCard für Exchange-basierte Anwendungen, PDF als Kundenprofil zum Ausdrucken und XML für andere weiterverarbeitende Systeme. Reale Beispiele für den Einsatz von REST sind Management-APIs für Infrastrukturkomponenten wie jBPM und Amazon Cloud.

Business Impact

Betriebskosten: Die Verwendung von REST auf Integrationsebene führt zu einer optimierten Nutzung der IT/Web-Infrastruktur im Unternehmen. Zusätzliche Infrastrukturinvestitionen für Caching, Load-Balancing und Security sind typischerweise nicht notwendig.

Software-Entwicklung: Die gute Unterstützung u.a. in JEE6 aber auch durch Standard-Clients wie Web-Browsern erlaubt eine sehr leichtgewichtige, schnelle Umsetzung (Rapid Application Development) von Anforderungen in IT-Projekten.

The Cloud Way: Öffentlich zur Verfügung gestellte Schnittstellen auf Basis von REST erfreuen sich einer größeren Adaption und sind somit erfolgreicher.

Pro	Contra
Hoher Grad an Interoperabilität / Entkopplung	Fehlende Standards für NFAs wie Sicherheit
Verwendung unterschiedlicher Darstellungsformate (vCard, JSON, XML, ...)	Geringere Werkzeug-Unterstützung
Verlinkung und vollständige Adressierbarkeit	REST ist an HTTP gebunden. Keine anderen Protokolle möglich.
Geringe Einstiegshürde.	

msg systems ag

Robert-Bürkle-Straße 1 | 85737 Ismaning/München
Telefon: +49 89 96101-0 | Fax: +49 89 96101-1113
www.msg-systems.com | info@msg-systems.com

Stand: März 2012

<http://www.msg-systems.com/techrefresh>

