



Die ARVIDA Referenzarchitektur

26.10.2016, Das ARVIDA-Architektenteam

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



DLR Projektträger



Definition Referenzarchitektur

- Eine Referenzarchitektur ist in ein **Referenzmodell** für eine Klasse von Architekturen
- Die Referenzarchitektur kann als Modellmuster – also ein **idealtypisches** Modell – für die Klasse der zu modellierenden Architekturen betrachtet werden

Quelle: Wikipedia

Motivation für eine offene Referenzarchitektur

Etablierte Systeme

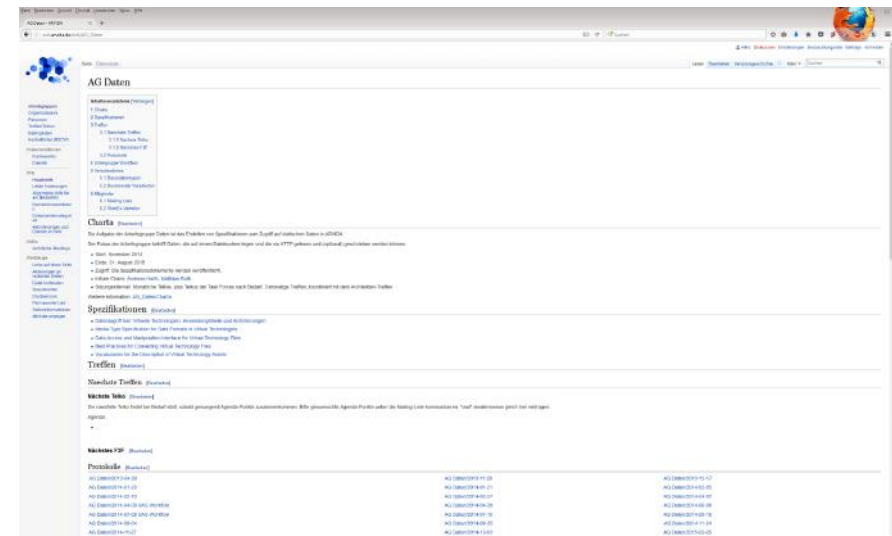
- Ermöglichen viel Funktionalität so lange man in einer Systemwelt bleibt
- Sind innerhalb einer Systemwelt (weitgehend) interoperabel
- Sind meist modular aufgebaut

Aber

- Starke Systemabhängigkeit, kaum Standards
- Fixierung auf den Szenegraphen als zentrale Datenstruktur für Visualisierungssysteme
- Fixierung auf Datenaustauschformate (JTOpen, etc.)
- Wenige Integrationsmöglichkeiten für Fremdsysteme – Austausch über Dateiformate oder sogar Nachbau benötigter Funktionalitäten
- Viele lokale Prozesse, vor allem für Rendering und Darstellung
- Dadurch eingeschränkte Skalierbarkeit und hohe Aufwände für die Synchronisierung von Datenbeständen

Entwicklung der Referenzarchitektur

- Architektenteam
- Bildung von projektbegleitenden, thematischen Arbeitsgruppen
- Diskussion spezifischer Konzepte und Schnittstellenanforderungen in den AGs
- Regelmäßige „General Assembly Meetings“
- Aufbau eines MediaWiki als zentrale Informationsdrehscheibe
- Entwicklungswerkzeuge und Beispiele



Anforderungen an die Referenzarchitektur

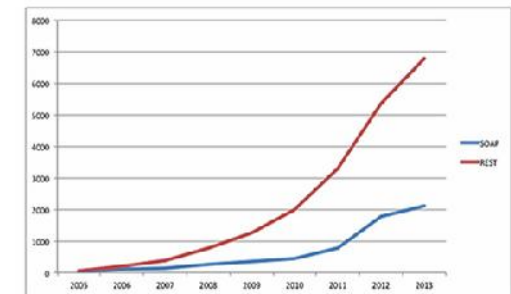
Die ARVIDA-Referenzarchitektur soll

- eine offene Architektur sein
- etablierte Web-Technologien und Standards nutzen
- systemwelt- und plattformunabhängig sein
- modular und dienstorientiert sein
- verteilt und skalierbar sein
- selbsterklärungsfähige Komponenten definieren
- Migrationspfade zu etablierten Systemen bieten

Konzepte der Referenzarchitektur: REST

- REST als etablierter Architekturstil für Web-basierte Anwendungen
- Ein Hauptziel von REST ist Skalierbarkeit
- REST-basierte Systeme folgen bestimmten Einschränkungen
 1. die Sicht auf die Datenelemente und Komponenten einer Anwendung in der Form von Ressourcen
 2. der Datenaustausch zwischen Komponenten durch die Manipulation von Zuständen der Ressourcen über den Austausch von Repräsentationen
 3. selbstbeschreibenden Ressourcen, die mit weiteren Ressourcen verlinkt sind (Hypermedia, Linked Data)
- Flexible Erstellung von Anwendungen durch eine Komposition von unabhängigen Komponenten, deren Daten bzw. Funktionen als Ressourcen abstrahiert sind
- Die einzelnen Komponenten sind über das Netzwerk zugreifbar

REST wins out



Number of SOAP vs. REST based APIs (Source: programmableweb.com)

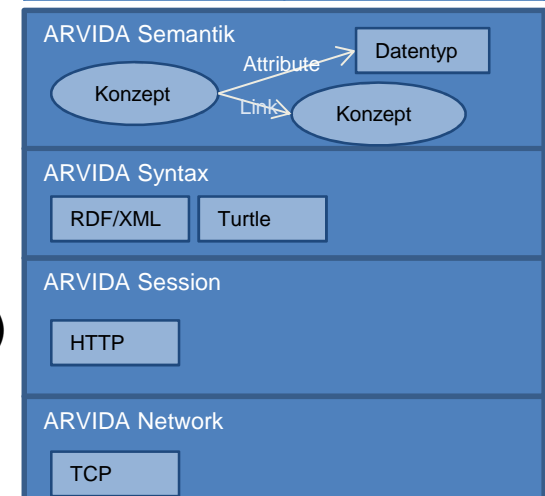
Konzepte der Referenzarchitektur: RDF, OWL und Linked Data

- Linked-Data bezeichnet Daten/Ressourcen, die per Uniform Resource Identifier (URI) identifiziert sind und darüber direkt per HTTP abgerufen werden können
- Ressourcen können per URI wiederum auf andere Daten verweisen
- Idealerweise werden zur Kodierung und Verlinkung der Daten das Resource Description Framework (RDF) und darauf aufbauende Standards wie SPARQL und die Web Ontology Language (OWL) verwendet. Folgende Regeln erleichtern den Umgang mit Ressourcen:
 1. Verwende zur Bezeichnung von Objekten URIs
 2. Verwende HTTP-URIs, so dass sich die Bezeichnungen nachschlagen lassen
 3. Stelle zweckdienliche Informationen bereit, wenn jemand eine URI nachschlägt (mittels der Standards RDF und SPARQL)
 4. Zu diesen Informationen gehören insbesondere Links auf andere URIs, über die weitere Objekte entdeckt werden können

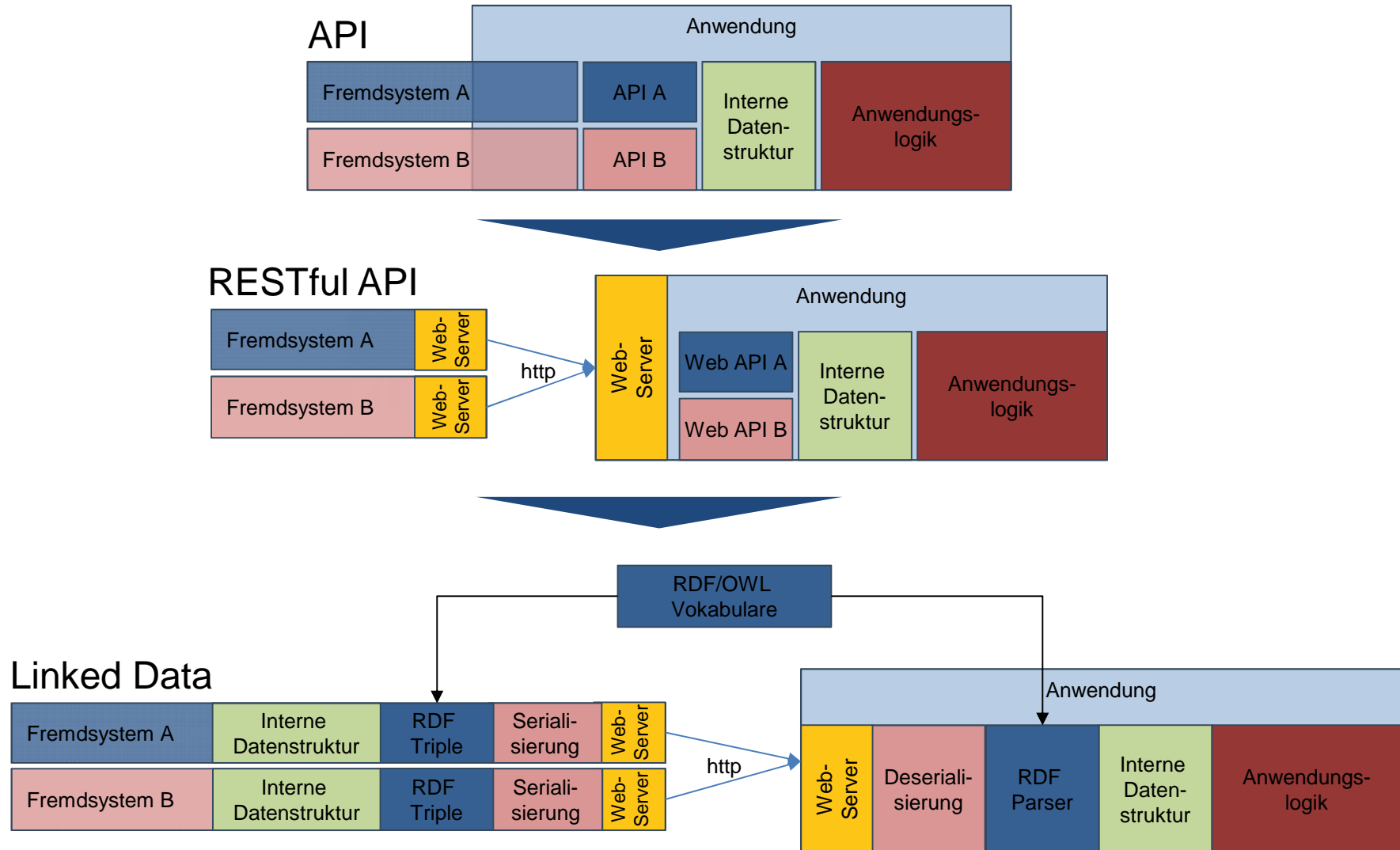
Konzepte der Referenzarchitektur

- Nutzung der Linked-Data-Prinzipien und Erweiterung auf virtuelle Technologien
- Modularität durch offene, semantische Ressourcenbeschreibungen
- Einheitliche Ressourcenbeschreibungen
- Standardisiertes Ressourcenverhalten
- Einfache Ressourcenerstellung
- Ressourcen können ganze VT-Teilsysteme umfassen (Trackingsysteme, Simulationssysteme, Menschmodelle, etc.)
- Bereitstellung von RDF-Vokabularen für die VT-Anwendungsentwicklung (<http://vocab.arvida.de>)

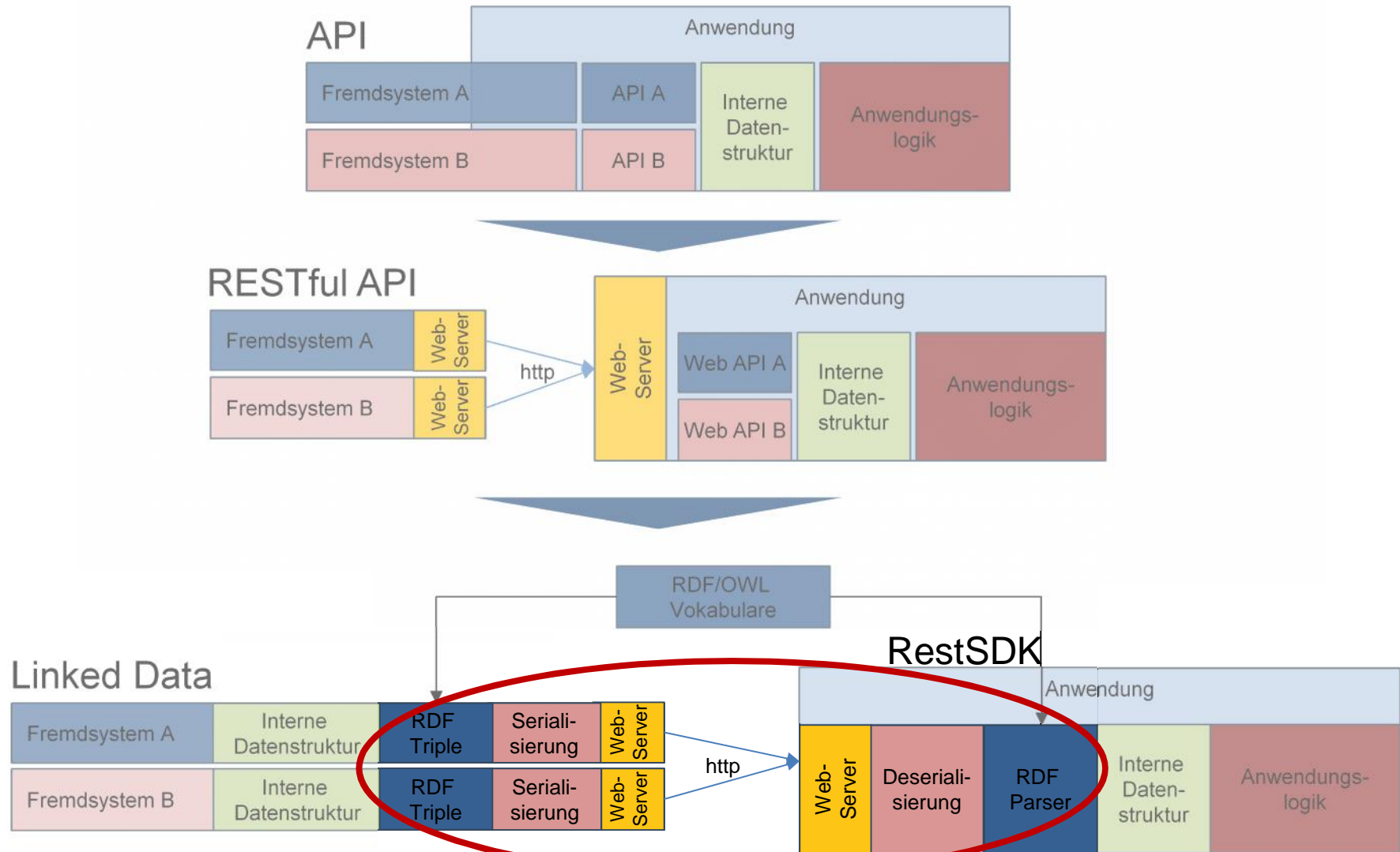
Domain	Spezifikation
Protokoll	HTTP
Daten Model	RDF
Serialisierungsformat	RDF/XML Turtle RDFa JSON-LD
Suche	SPARQL



Konzepte der Referenzarchitektur



Konzepte der Referenzarchitektur



Erreichtes

- Die Projektziele wurden erreicht
 - Definitionen für ARVIDA-konforme Dienste/Ressourcen und deren Verhalten
 - RestSDK für C/C++
 - FastRDF
 - VT-Vokabulare für die essentiellen Elemente einer VT-Anwendung
 - Beispieldienste sowie Mechanismen zur Dienstinteraktion sowie zur Validierung von Daten und Rückgabewerten
 - Beschreibungssprache und Linked-DataFu zur Anwendungserstellung
- Eine sinnvolle Referenzarchitektur konnte erfolgreich umgesetzt werden
- Diese umfasst wesentliche Elemente einer typischen VT-Anwendung
- Industrielle Beispielanwendungen wurden damit realisiert
- Migrationspfade zu existierenden Systemen wurden entwickelt und genutzt

Erreichtes

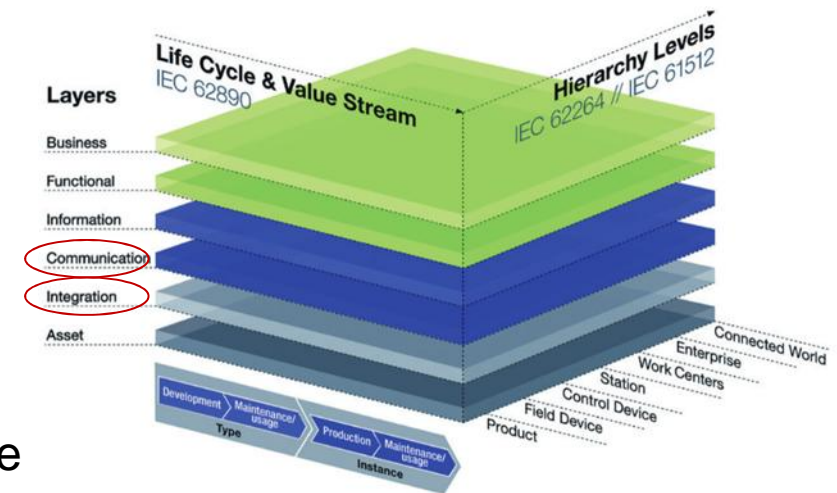
- Die ARVIDA-Konformität stellt sicher, dass verschiedene Komponenten leichter austauschbar sind und miteinander arbeiten
- Die anfangs befürchteten Performanzeinbußen haben sich in den Anwendungsszenarien als nicht kritisch erwiesen
- Anbindungen an VT-Systeme, wie Teamcenter und Unity3D sind vorhanden
- Vor allem im Bereich Tracking wurden auch „Mikroservices“ umgesetzt
- Es können in sich gekapselte VT-Dienste/Ressourcen zu komplexeren Anwendungen zusammengesetzt werden, „System of Systems“
- Die Dienste können komplette Fremdsysteme sein, die ihre Funktionen als Dienst bereitstellen oder aber kleine Funktionseinheiten
- Vor allem im Bereich der verschiedenen Tracking-Systeme wurde die gewünschte Interoperabilität und Modularität zwischen den Teilsystemen erreicht.
- Es fehlen weitergehende Sicherheitskonzepte sowie größere Anwendungen, die alle bzw. sehr viele der Architekturelemente umfassen.

Erfahrungswerte für die Umsetzung einer Referenzarchitektur

- Die Entwicklung einer halbwegs vollständigen und industriell (prototypisch) nutzbaren Referenzarchitektur ist eine große Aufgabe
- Entwicklungswerkzeuge sind sehr hilfreich zum Aufbau von ersten Anwendungen
- Der Ansatz der thematischen Arbeitsgruppen war erfolgreich
- Sehr steile Lernkurve am Anfang
- Ungewohnter Umgang mit semantischen Web-Strukturen und Programmierparadigmen
- Die Anschlussfähigkeit zu weiteren Systemen und Systemwelten ist zu beachten

Anschlussfähigkeit: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)

- Die Achse Hierarchy Levels repräsentiert die Zuordnung innerhalb einer Fabrik oder Anlage
- Die Achse Life Cycle & Value Stream umfasst den Lebenszyklus von Anlagen oder Produkten
- Die Achse Layers beschreibt sechs Ebenen der IT-Repräsentation eines Produktes oder einer Anlage
- Die **Integrationsschicht** stellt Informationen über eine Anlage zur Verfügung.
- Die Kommunikationsschicht stellt diese in Form von **Diensten** bzw. **Ressourcen** anderen IT-Systemen zur Verfügung
- Einführung einer **Industrie 4.0-Komponente**, um auf eine standardisierte Weise reale Anlagen, Anlagekomponenten oder Produkte zu beschreiben, mit ihnen über Dienstschnittstellen zu kommunizieren und so ausgewählte Informationen z.B. über einen Anlagenzustand in VT-Anwendungen abzubilden bzw. zu visualisieren



Quelle: <http://www.zvei.org/Downloads/Automation/Statusreport-Referenzmodelle-2015-v10.pdf>

Anschlussfähigkeit: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)

- RAMI4.0 ist ein Referenzmodell. Die ARVIDA-Referenzarchitektur ist eine Referenzarchitektur und kein Modell (mehr). Es gibt konkrete Modellierungen, Werkzeuge, Validierungsmechanismen, Dokumentationen und Anwendungen.
- RAMI4.0 soll dieselben (semantischen) Web-Techniken verwenden wie ARVIDA
- RAMI4.0 könnte einen großen Teil – wenn nicht sogar alles – an bei ARVIDA erprobten Entwicklungsmechanismen (Vocab, MediaWiki, GitHub) und Werkzeugen (RestSDK, Validierungsdienste, Orchestrierung mit Linked Data-Fu, etc.) übernehmen und weiterentwickeln

Nachhaltigkeit

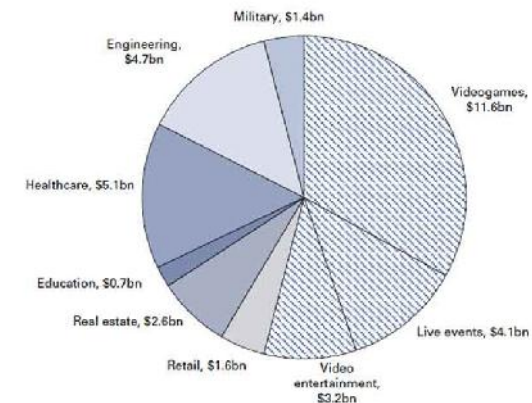
- 3 Jahre sind nicht hinreichend, um eine Architektur fest zu etablieren
- Die Veröffentlichung der Vokabulare ist nur der erste Schritt
- Weitere Ausarbeitung, mehr und bessere Entwicklungswerkzeuge
- Mehr Anwendungsszenarien aus weiteren Anwendungsgebieten können die Sichtbarkeit und Akzeptanz erhöhen

- Die technische Anschlussfähigkeit ist vor allem im Bereich Industrie 4.0 und der autonomen Systeme gegeben
- Visuelle Informationsaufbereitung wird immer wichtiger
- Herstellerneutrale Initiative/Plattform, wie z.B. SmartLiving, können eine Referenzarchitektur stabilisieren – evtl. sogar Anschluss an RAMI4.0
- Auch andere VT-nahe Systemwelten (z.B. PLM) untersuchen Web-Technologien zur Modularisierung und Verteilung

Ausblick

- Sehr viele neue und kostengünstige VR/AR- sowie Interaktionsgeräte und rasant wachsender Markt
- Das Web wird zunehmend zu einem großen, verteilten „Computer“. Es spielt keine Rolle mehr, wo Daten und Services liegen. Damit werden äußerst flexible VT-Anwendungsszenarien möglich und VT-Dienste können so wie heute z.B. Email-Dienste von überall genutzt werden
- Heutige IT Strukturen stehen dem entgegen. Klassische Firewalls und streng geschützte IT-Bereiche verhindern in vielen Fällen einen verteilten Ansatz

Exhibit 4: Our 2025 base case VR/AR software assumptions by use case



Source: Goldman Sachs Global Investment Research.

Ausblick

- Trennung von Anwendungslogik und spezifischen Systemen – Beschreibungssprache für verteilte Anwendungen im Web
- Synchronisierung bei komplexen Anwendungen mit vielen Diensten und Fremdsystemen
- Sicherheitspolicies von IT-Abteilungen vs. Verbindung zu Cloud Services
- Paradigmenwechsel von Firewall/lokal zu offen, verteilt, „unsichtbarer“ Herkunft von Services
- Update Mechanismen speziell bei Mikroservices
- Verbindung zu IoT-Middleware-Systemen

Ausblick



- Umfassender Einsatz von VR/AR-Technologien
- Umfassende Digitalisierung -> Verschmelzung der digitalen/virtuellen und realen Welt
- Ersatz monolithischer durch modulare, flexible und offene Systemwelten auch außerhalb von VT
- Daten und Funktionen verlinken - Aggregation von verteilten Daten und Funktionen
- Digitale Mehrwertdienste als Teil der Wertschöpfung über die Produktlebensdauer hinweg
- Autonome Funktionen von Teilsystemen und mehr künstliche Intelligenz

Demos

Tech-Demos am Stand der Architekten