



Berner Architekten Treffen

Thema: INTEGRATION



swissdec - Architektur

zur Integration und Automatisierung von
Geschäftsprozessen zwischen
Unternehmen, Versicherern und Behörden
in der Schweiz

Anton Böhm

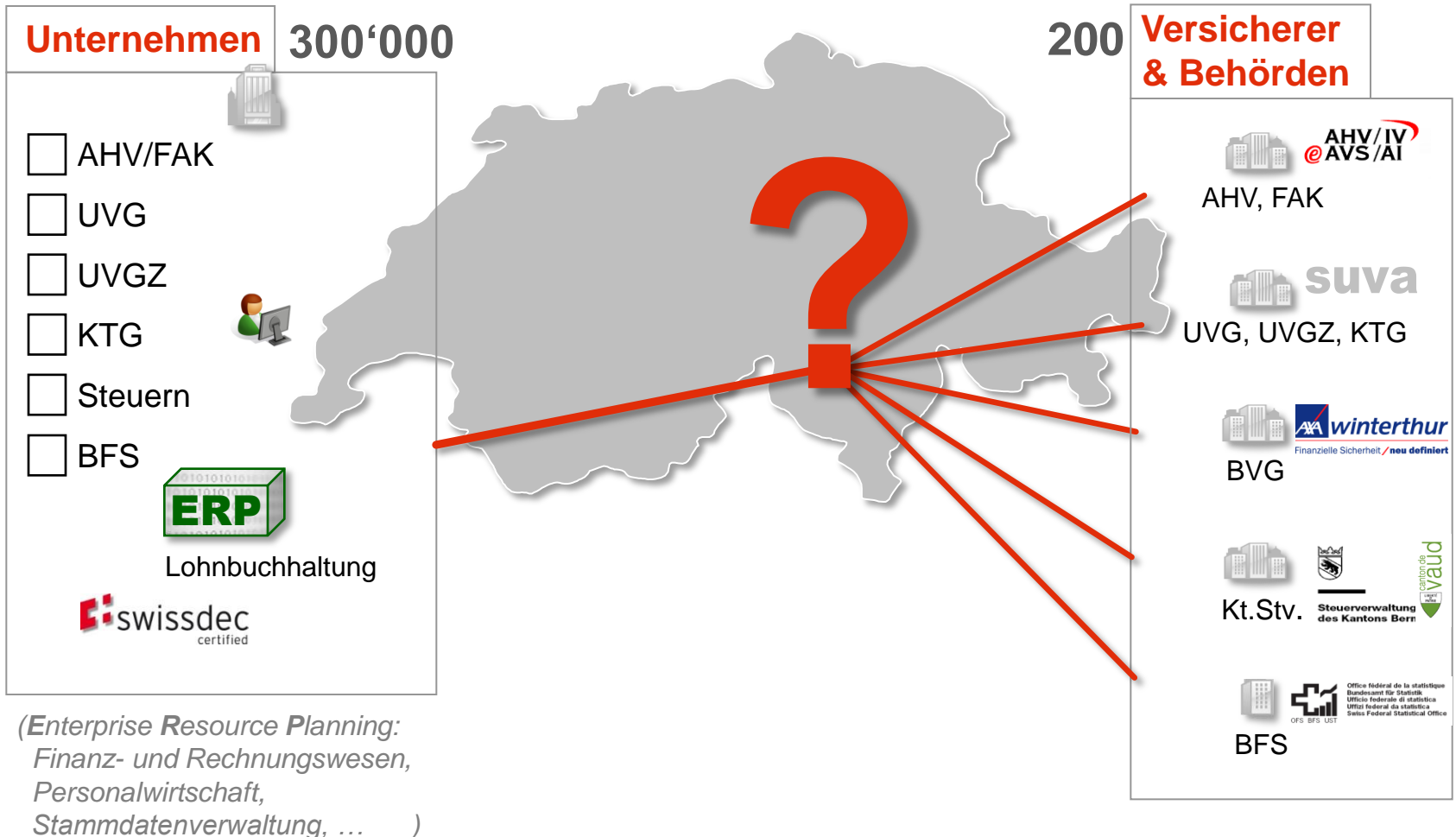
Einige Aspekte aus etwa 8-9 Jahren Erfahrungen

Themen

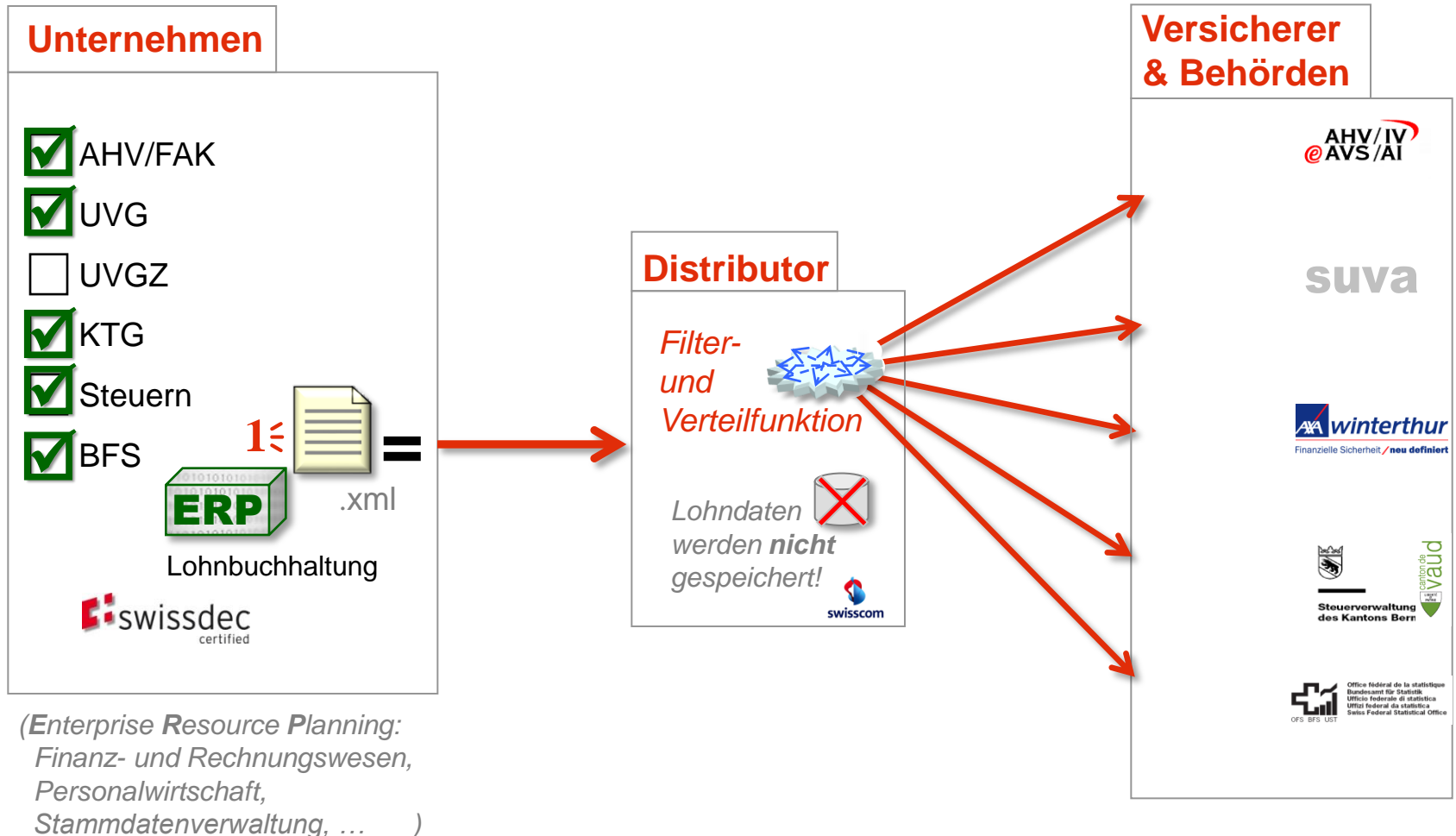
1. Kurze Einführung zu swissdec
2. Innovation, Design und lose Kopplung (Muster)
3. Warum keine Kopplungs-„Apps“ für die Teilnehmer (Entscheid)
4. Distributor und „ein“ XML (Entscheid)
5. Prozessqualität, Sicherheit und Datenschutz
6. Diverses:
 - Grosse Datenmengen „XXL-ELM“
 - Bemerkungen

Wer ist swissdec?

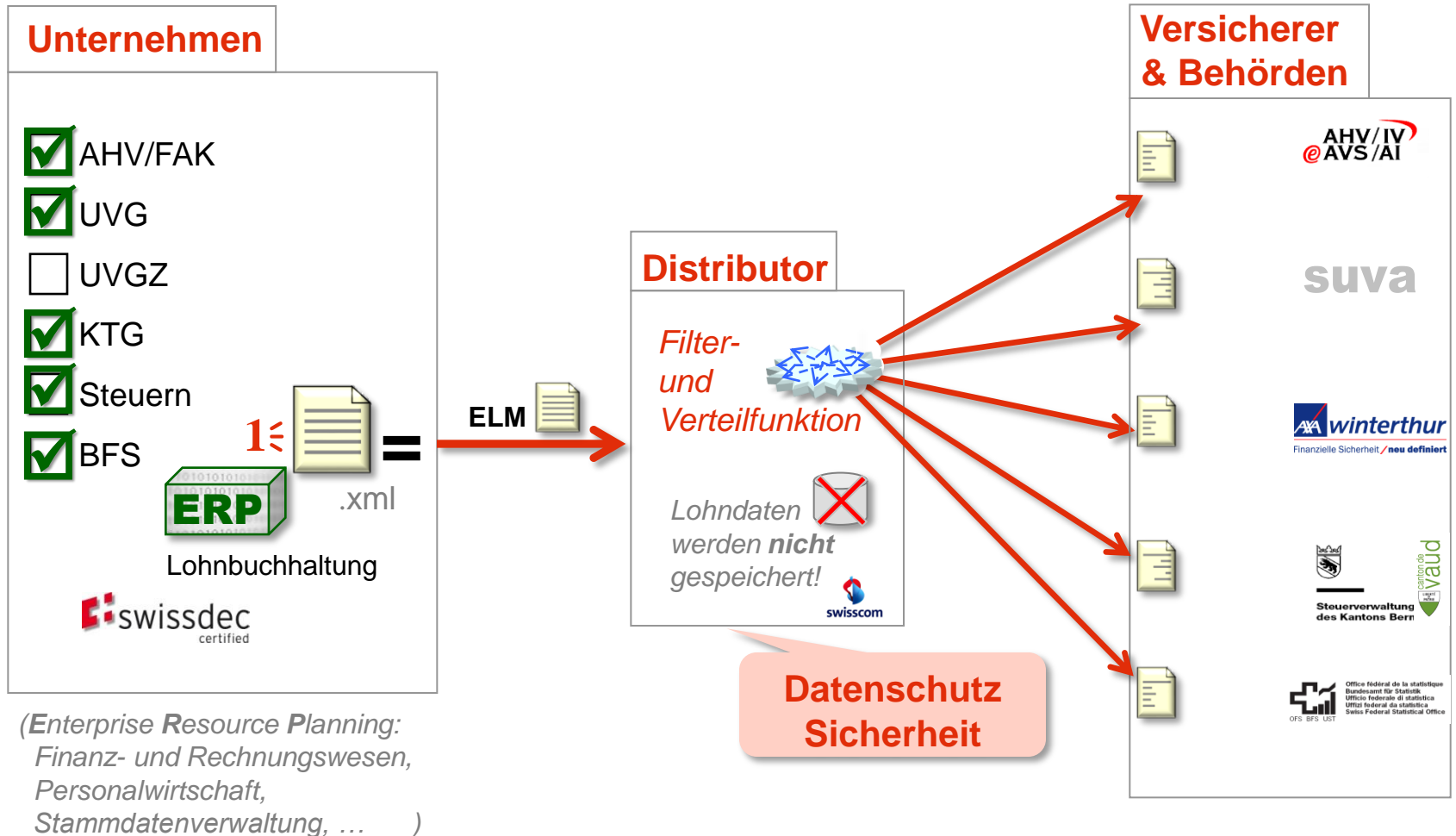
swissdec Lösung



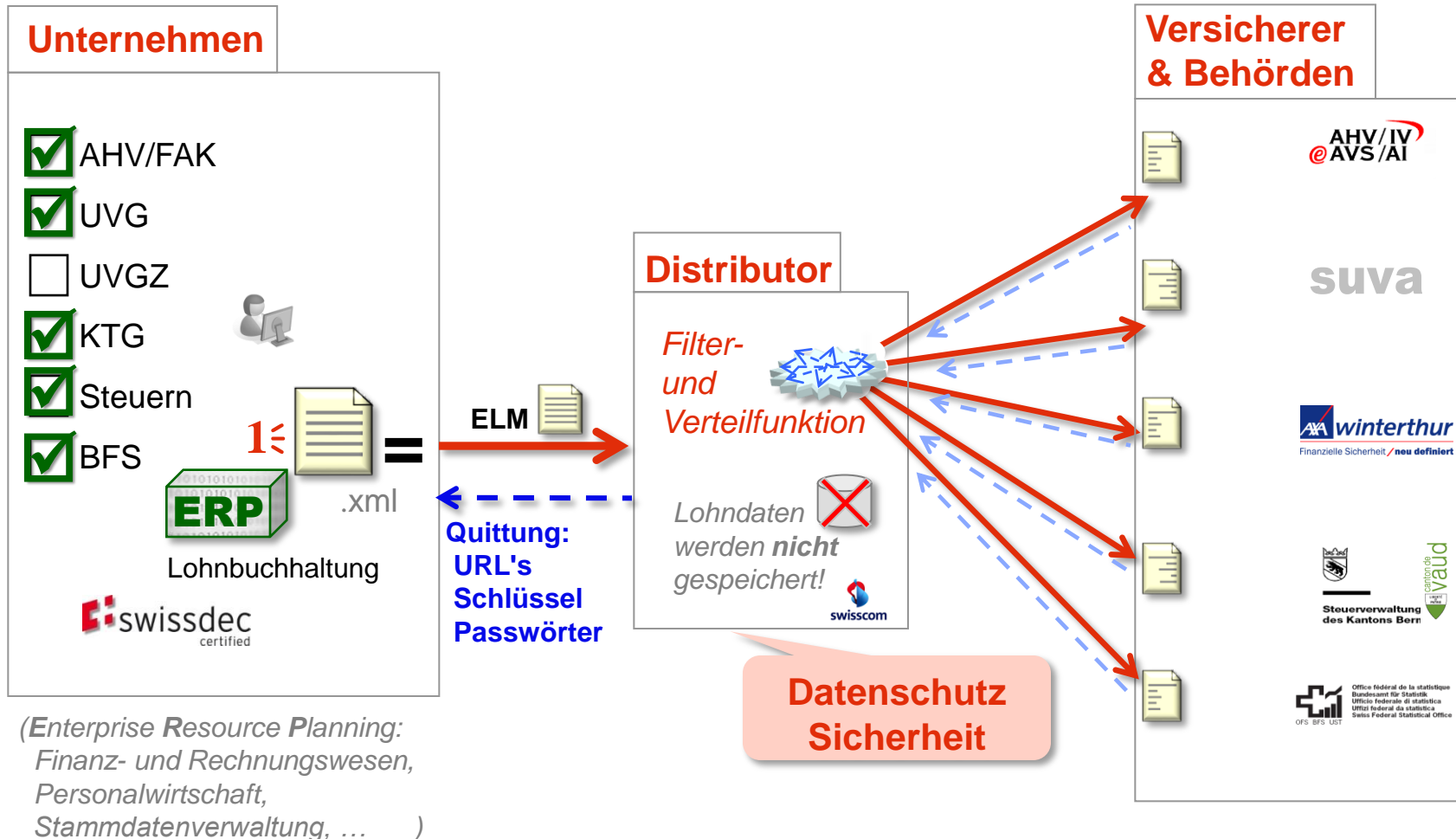
swissdec Lösung



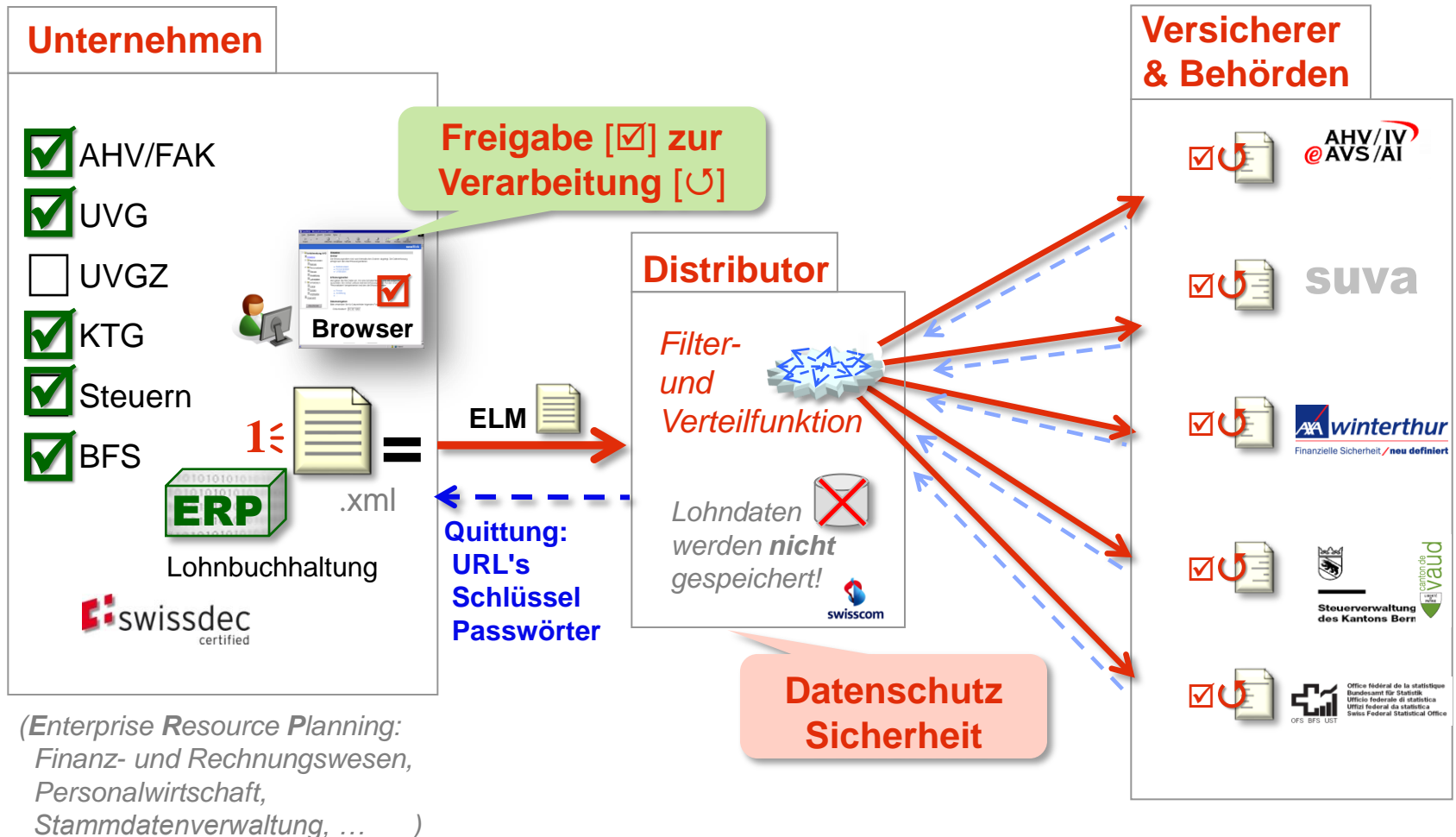
swissdec Lösung

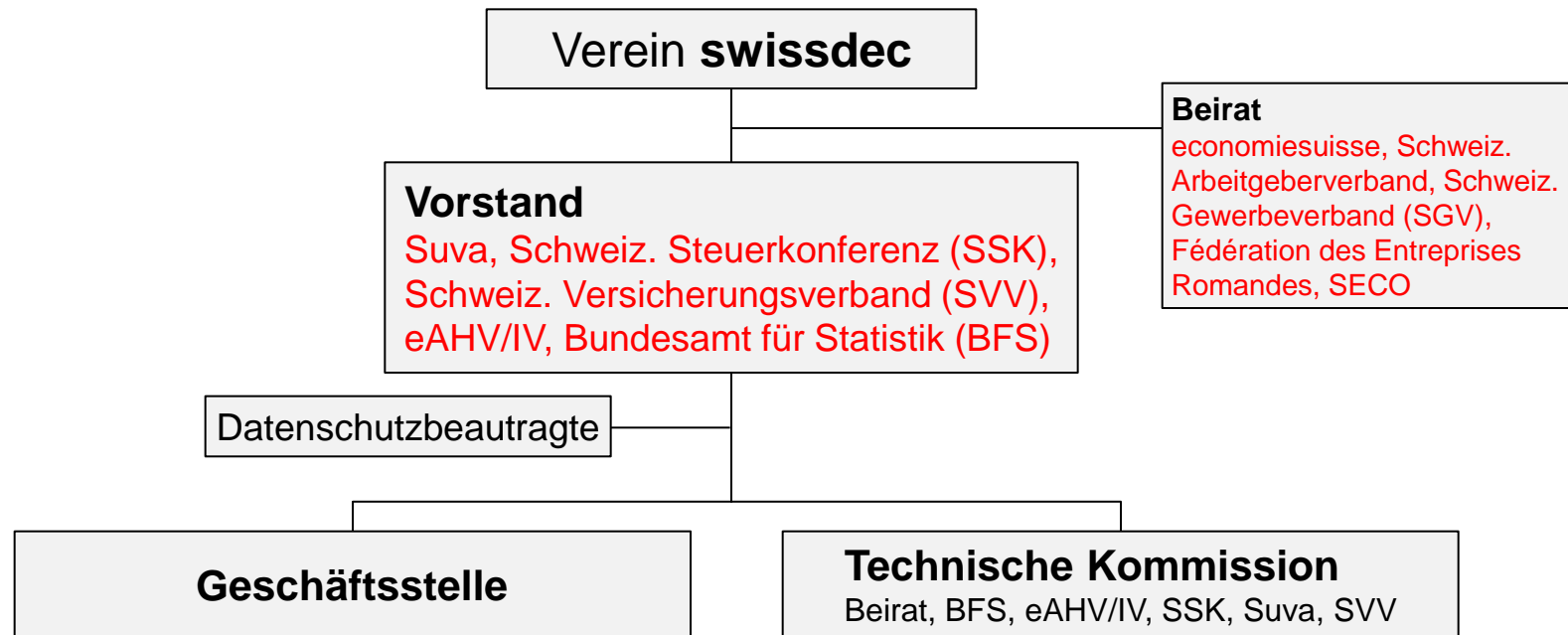


swissdec Lösung



swissdec Lösung





- **Fachstelle**
Beratung u. Zertifizierung, Standards (ELM) u. Dokumentation, Test u. Produktionshilfen, Prozessüberwachung
- **Marketing und Administration**
Event, Messen, Informationsmittel, Homepage, Rechnungswesen, Protokoll, Planung
- **Projekt Leistungsstandard-CH**
UVG, UVGZ, KTG
- **Projekt Lohnstandard-CH Quellensteuer**
- **Projekt ESTV-Standard-CH**
Eidg. Steuerverwaltung

- **Fachgruppe AHV/FAK**
Fachstelle, eAHV/IV, Ausgleichskassen, Softwarepools
- **Fachgruppe Versicherung**
Fachstelle, SVV, Suva
- **Fachgruppe Statistik**
Fachstelle, BFS
- **Fachgruppe BVG**
Fachstelle, SVV, eAHV/IV
- **Fachgruppe Steuern**
Fachstelle, SSK Logistik, Experten Lohnausweis
- **Fachgruppe Lohnsoftware-Hersteller**
Fachstelle, Lohnsoftware-Hersteller

Prozesslandkarte swissdec (CH-Standards)



Prozesse

ESTV-Standard-CH

- MwSt-Abrechnung
- ...



Leistungsstandard-CH

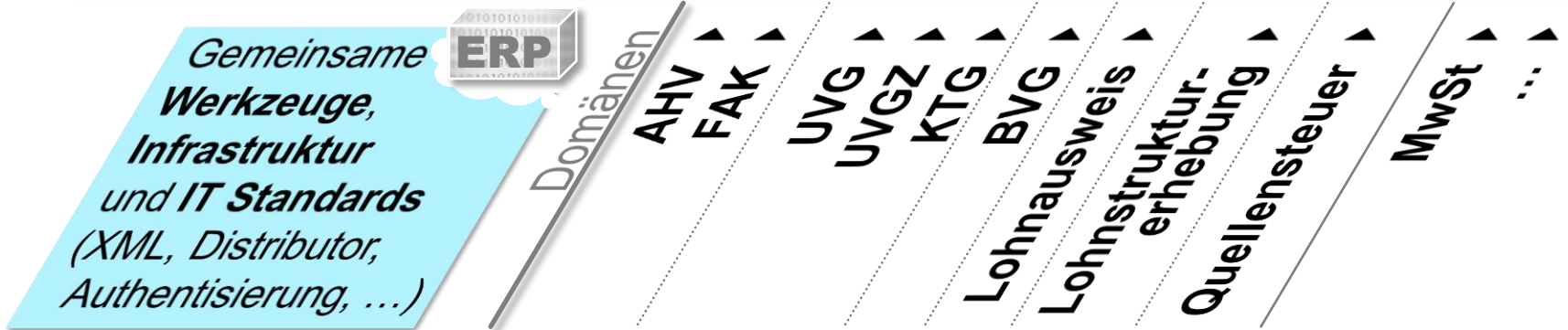
- Kundenintegrierter Leistungsprozess von Anspruch bis Erbringung (KLEE)

? ? ✘ ✘ ✘

Lohnstandard-CH

- Eintritt, Mutation, Austritt (EMA)
- Einheitliche Lohnmeldung (ELM)

? ?
 ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✘ ✘



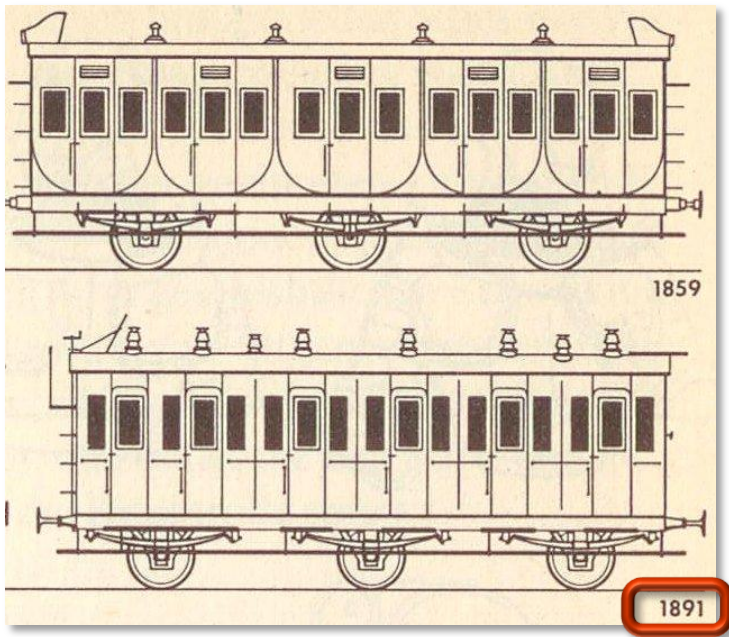
☑ Realisiert ... ✘ inArbeit ... 👁 Vision

„Kutschen Design“ von Konrad Lorenz

Was lernen wir von den Eisenbahn-Entwicklern?

Naheliegenderes Vorgehen:

„Man nehme eine Kutsche, **kopiere** sie mehrfach und baue alle Kutschen zu einer grossen **Kutsche** zusammen.“



☺ Vorteil:

- Einfache und bekannte Lösung
- Paralleler Ein- und Ausstieg

☹ Nachteil:

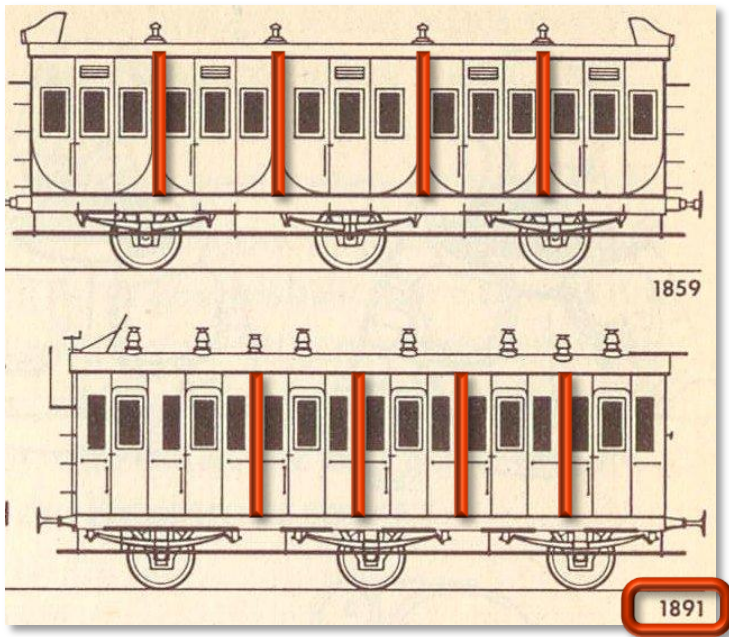
- „Frauenfeindlich“, d.h. kein Platzwechsel während der Fahrtzeit, bzw. schwieriger Ein- und Ausstieg.
- Fehlende Getränke und WC während der Fahrt.
- Fahrkarten-Kontrolle am Bahnsteig, d.h. an allen Haltestellen wird Personal benötigt.

„Kutschen Design“ von Konrad Lorenz

Was lernen wir von den Eisenbahn-Entwicklern?

Naheliegenderes Vorgehen:

„Man nehme eine Kutsche, **kopiere** sie mehrfach und baue alle Kutschen zu einer grossen **Kutsche** zusammen.“



☺ Vorteil:

- Einfache und bekannte Lösung
- Paralleler Ein- und Ausstieg

☹ Nachteil:

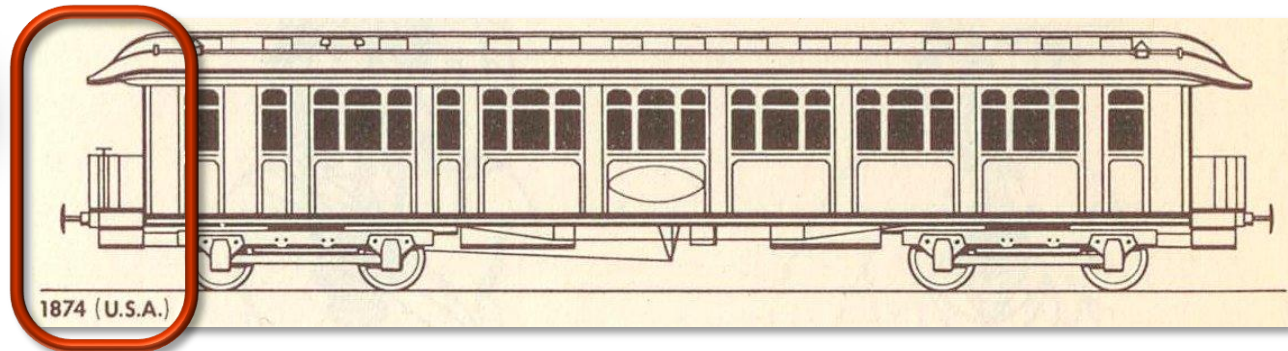
- „Frauenfeindlich“, d.h. kein Platzwechsel während der Fahrtzeit, bzw. schwieriger Ein- und Ausstieg.
- Fehlende Getränke und WC während der Fahrt.
- Fahrkarten-Kontrolle am Bahnsteig, d.h. an allen Haltestellen wird Personal benötigt.

Grundsätzlich: **Silo-Architektur**

„Kutschen Design“

Der moderne Durchgangswagen

Innovation



😊 Vorteil:

- Platzwechsel, Getränke und WC **während** der Fahrt.
- Kürzerer Ein- und Ausstieg
- Keine Fahrkarten-Kontrolle am Bahnsteig und damit **kein Personal** an den Haltestellen.
- Links und rechts Raucherplattform
- *Optimale **Abstimmung** von Wagen, Zug und Bahnhof als ein „System“*

☹️ Nachteil:

- Links und rechts **Banditenplattform** („Hacker“)

„Kutschen Design“ und verteiltes System mit loser Kopplung



Die Schweiz wird „portalisiert“. Daraus folgt eine einfache und naheliegender Lösung:

(Web) **Formulare** mit Portalen

Ist das eine **adäquate** Lösung?

❗ Medienbruch beim Unternehmen; Problem mit Datenmengen ☹; ...

Ein mögliches (**innovatives**) Vorgehen zum Systemdesign:

- **Was** für verschiedene Verantwortlichkeiten gibt es im **Prozess**?
- **Wie** können diese Verantwortlichkeiten **verteilt** werden?

Diskussion: „lose“ Kopplung = die „Systemintegration“?


Architektur **ZIELE**  swissdec

- Optimale Unterstützung des **Geschäftsprozesses** zwischen Unternehmen und Versicherern oder Behörden, d.h. alle Teile **optimal** aufeinander **abstimmen** (Zug, Bahnhöfe,... = **System**).
- **Lohnstandard** und erweiterte Nutzung z.B. **Leistung** und **Revision**

Lose Kopplung bedeutet die Verwendung von offen Standards (z.B. Web Service, WS- und http). Ja|Nein?*



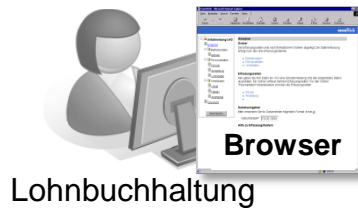
Bemerkungen dazu:

- Kopplung auf „Tech“ **und** „Fach“-Ebene
- Von der Komponenten- zur **Prozess-Kopplung** und damit zur **vollen** Integration.  Aber auch Anpassbarkeit auf allen Ebenen!
- Je „loser“ die Kopplung sein soll, desto wichtiger werden die Attribute: *kontextfrei, zustandslos, asynchron, ...*

swissdec Baukasten: Baustein „h2m“ (Human to Machine)

Unternehmen

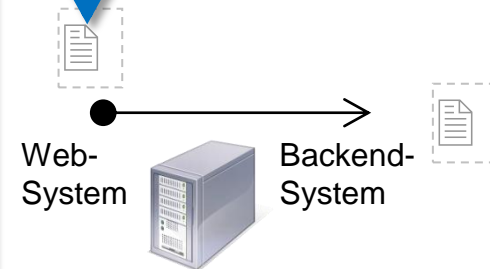
Versicherer



h2m (Human to Machine)

Bewertung:

- ☺ Wenige Teilnehmer in der Entwicklung
- ☺ Kurze Bereitstellung (= sehr schnell und flexibel)
Definition + Entwicklung: 1 bis x Monate
Rollout zu den Kunden: 1 bis x Tagen
Total: 1 bis x Monate ☺
- ☺ **Intelligent** durch den Endbenutzer (Human)
- ☹ Kleine **Datenmengen**, da manuelle Erfassung
- ☹ **Fehleranfällig**, da manuelle Erfassung
- ☹ **Medienbruch** beim Kunden



swissdec Baukasten: Baustein „m2m“ (Machine to Machine)

Bewertung:

- 😊 „Intelligent“ durch die QS-Softwareteile und direkte **Integration**
- 😊 **GROSSE** Datenmengen mit **Automatisation** (Industrialisierung)
- 😊 Kein **Medienbruch** beim Kunden und **bidirektionaler** Datenaustausch

- 😞 Viele Teilnehmer in der Entwicklung
- 😞 Lange Bereitstellung (= viel Koordination)

Definition + Entwicklung: 1 bis 2 Jahre

Rollout zu den Kunden: 0 bis 2 Jahre

Total: **1 bis 4 JAHRE** 😞

m2m (Machine to Machine)

ERP / Lohnbuch-
haltungs-
System



Unternehmen



Web
System



Versicherer

Backend-
System



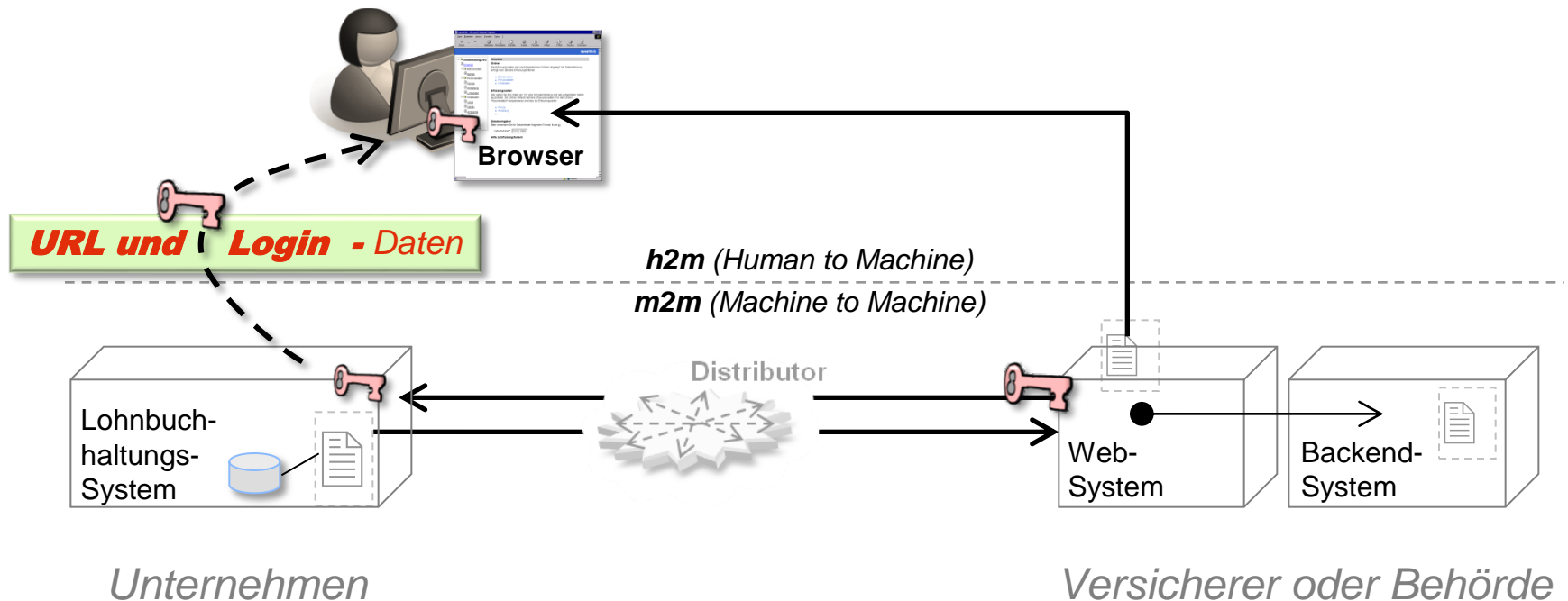
Muster

Verbindungen schaffen (h2m & m2m)

Die Bausteine (h2m und m2m) zum „Positiven“ **kombinieren, d.h.**

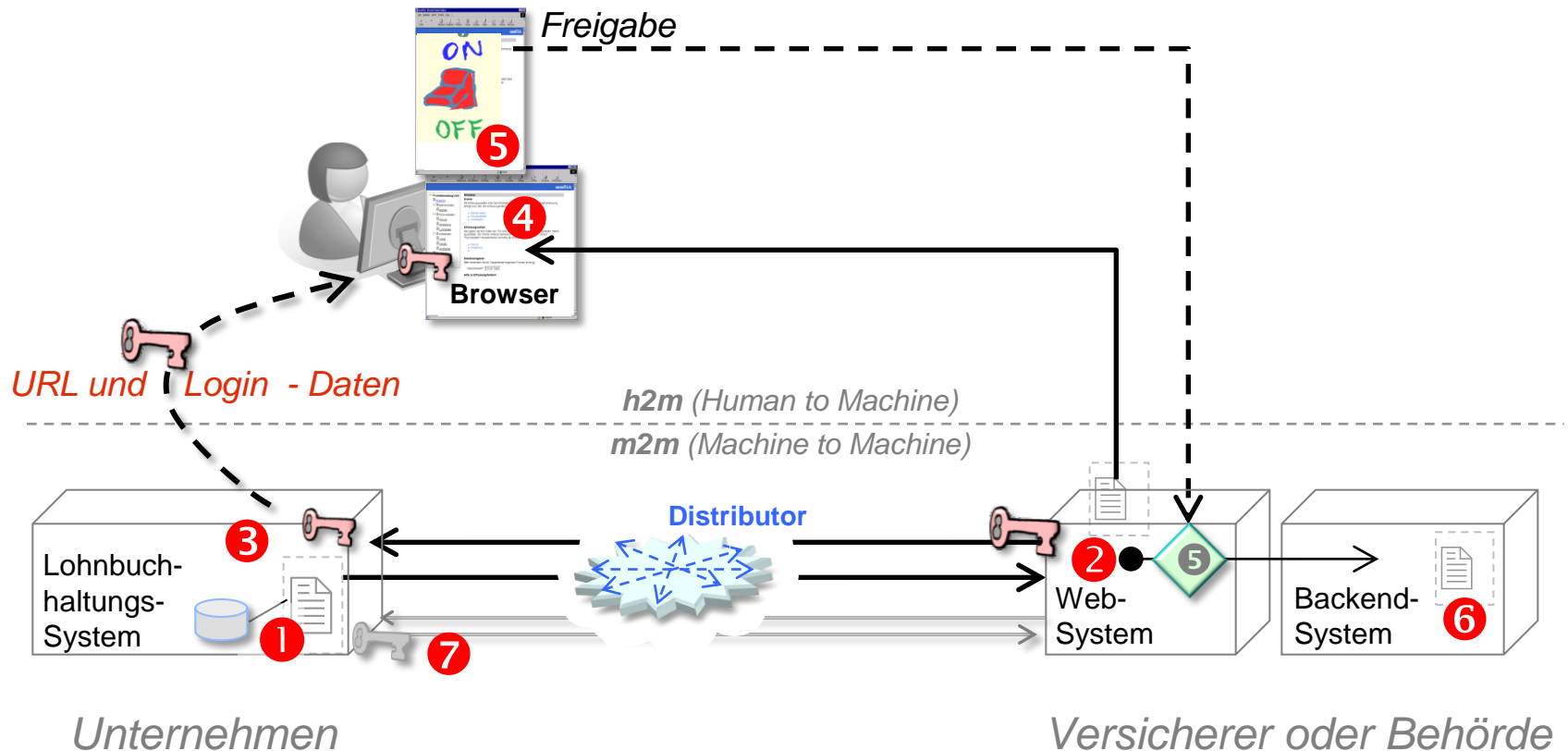
- **Protokoll** standardisieren, d.h.
 - **Verhalten** mittels Operationen (**Prozess**-Sicht)
 - **Daten** mittels Parameter in den Operationen (**Meldungs**-Sicht)
 - Möglichst viel aus den Spezifikationen **generieren** (Texte = ☹)
- **Ein System** aus den *Daten*, *Verfahren* und *Distributor* entwickeln, d.h.
 - **vereinfachter** Workflow (weniger Prozessredundanz, d.h. „eins zu mehreren“ Verteilungskonzept)
 - **Datenschutz** durch Filterung und Verteilung
 - **Brauchbarkeit** (Usability) durch „Echtzeit“-Rückmeldungen
- Weitere **Integration** mittels „Links“ z.B. in die bestehenden Portale
- Eine Versionierung der Lösungen, damit **zukünftige** Anforderungen integriert werden können.

«Cross-Channel¹⁾» Protokoll (h2m & m2m)

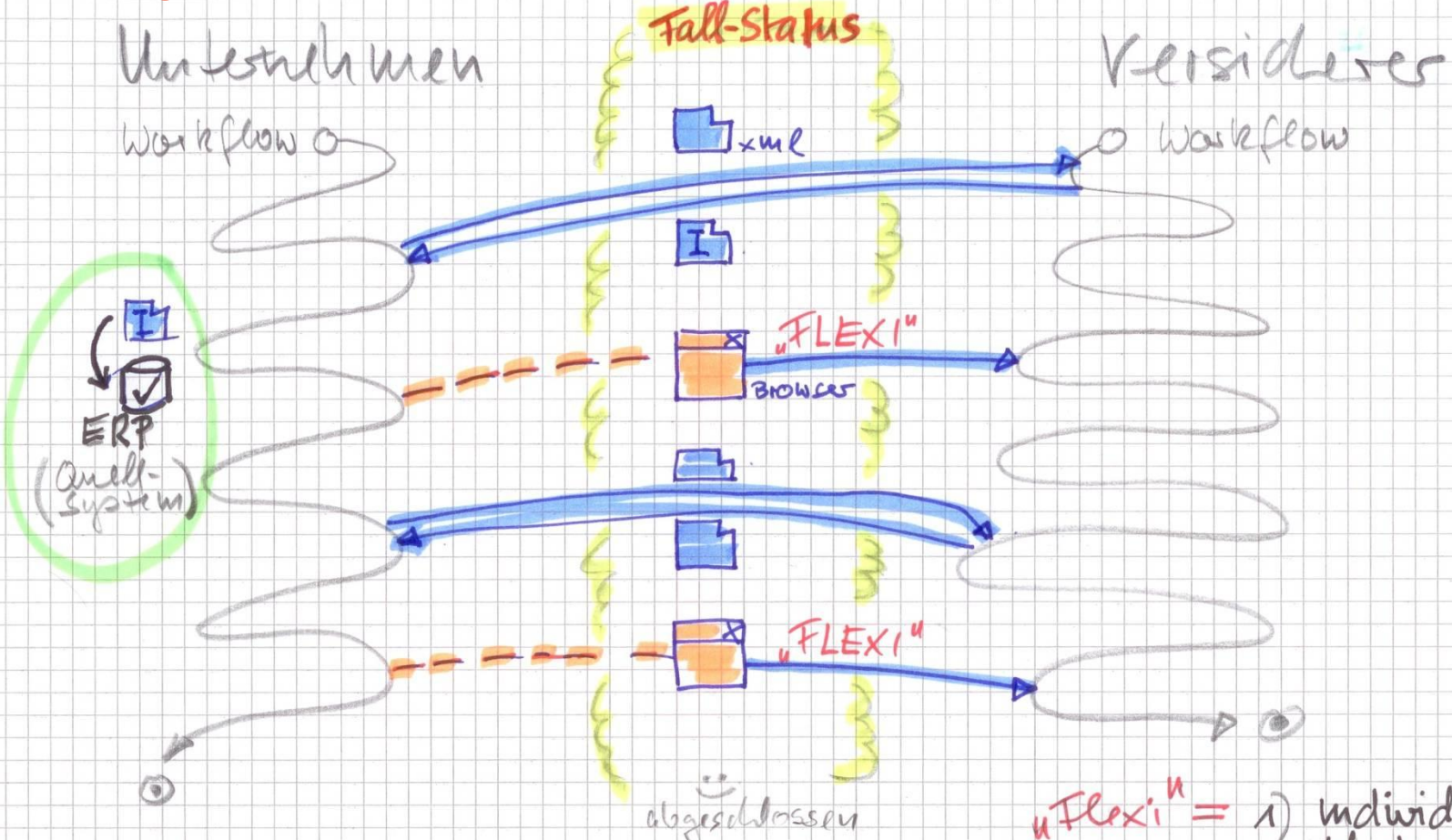


1) Cross-Channel: Kommunikationsprozess, der sich auf mehreren Kanälen abspielt, d.h. in diesem Fall h2m und m2m

«Cross-Channel¹⁾» Protokoll (h2m & m2m)



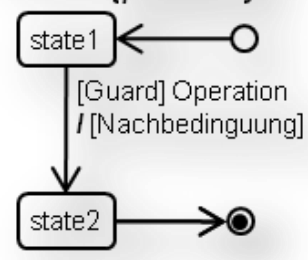
1) Cross-Channel: Kommunikationsprozess, der sich auf mehreren Kanälen abspielt, d.h. in diesem Fall h2m und m2m

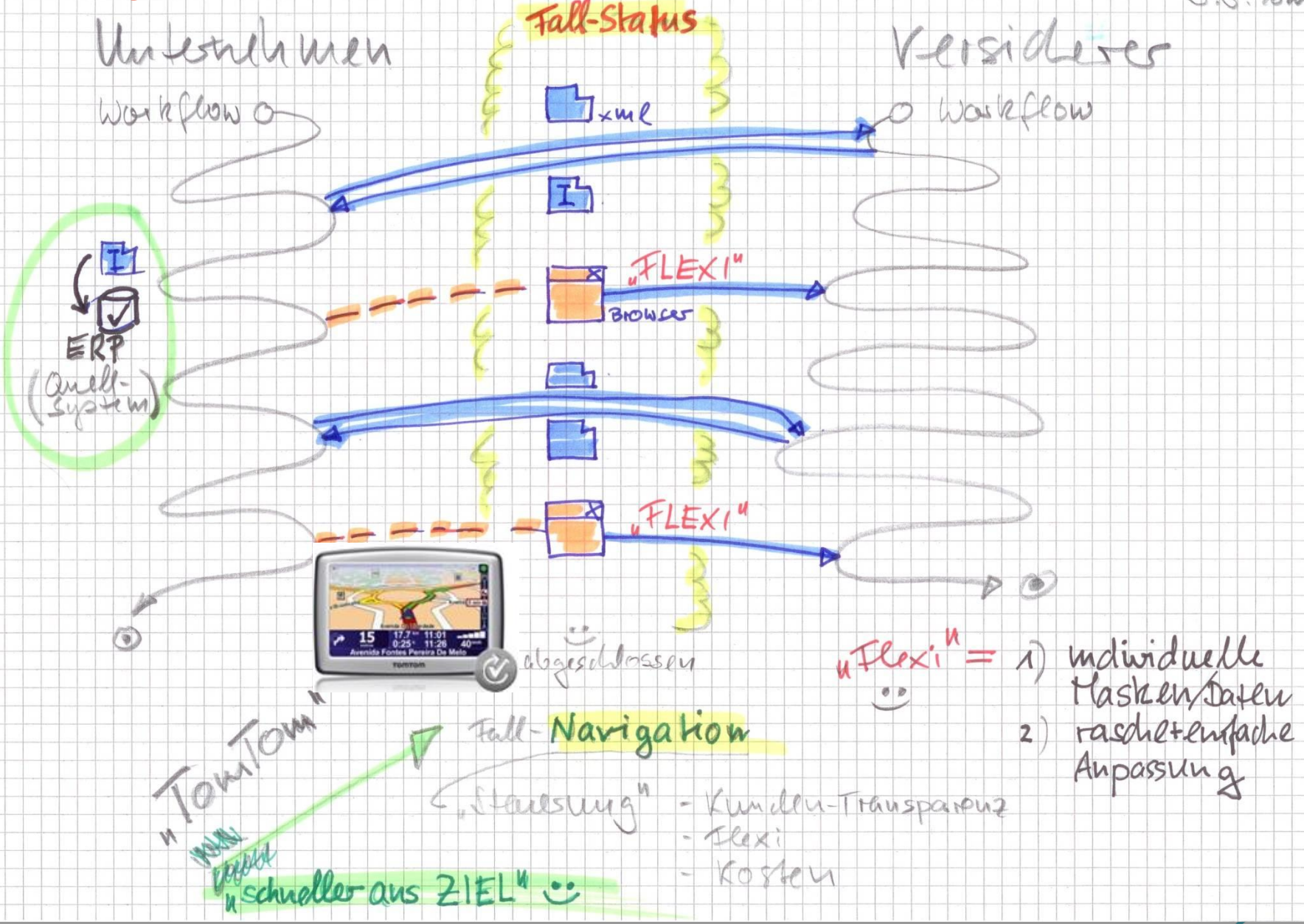


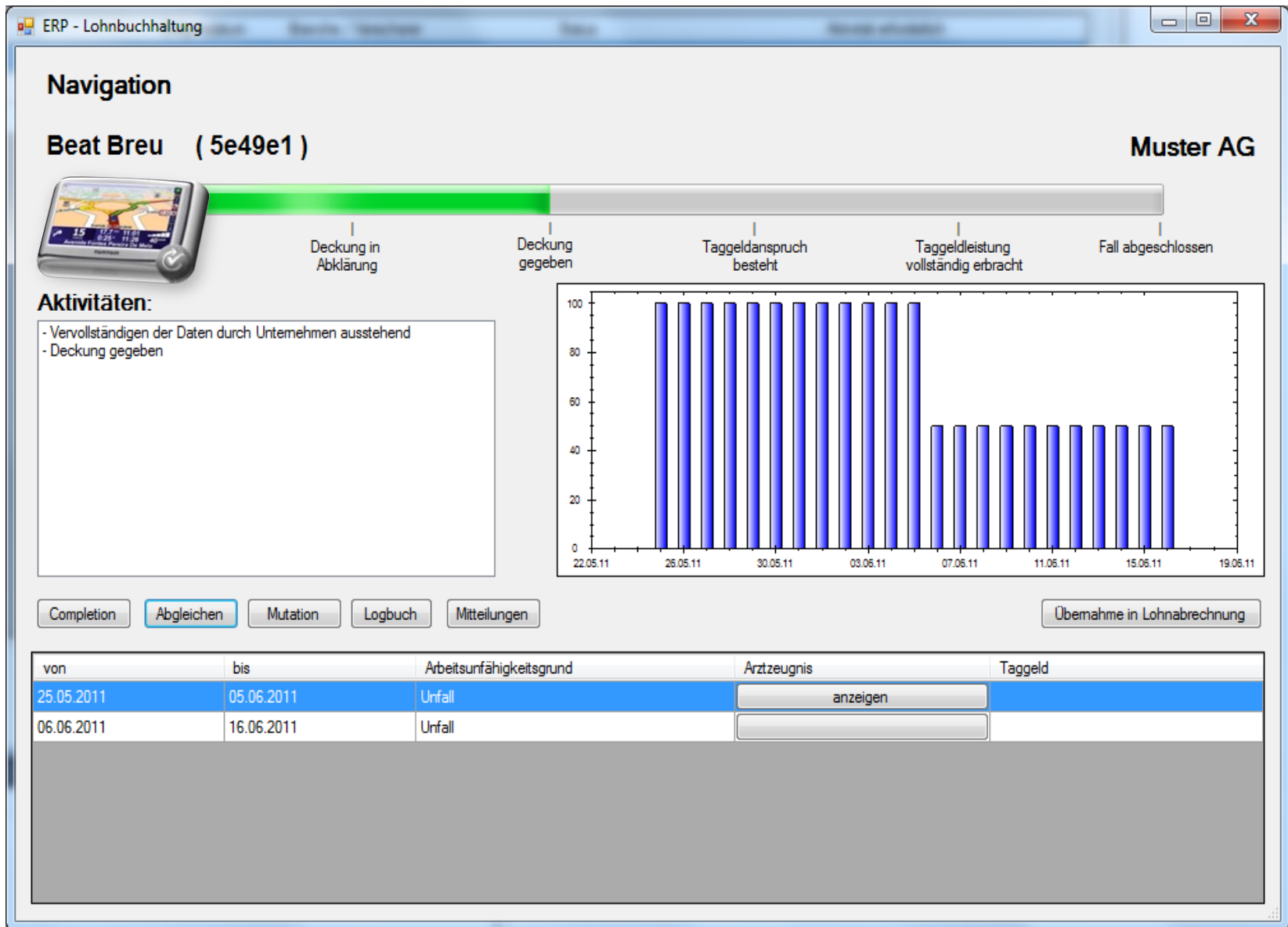
„Flexi“ =

- 1) individuelle Masken/Daten
- 2) rasche+einfache Anpassung

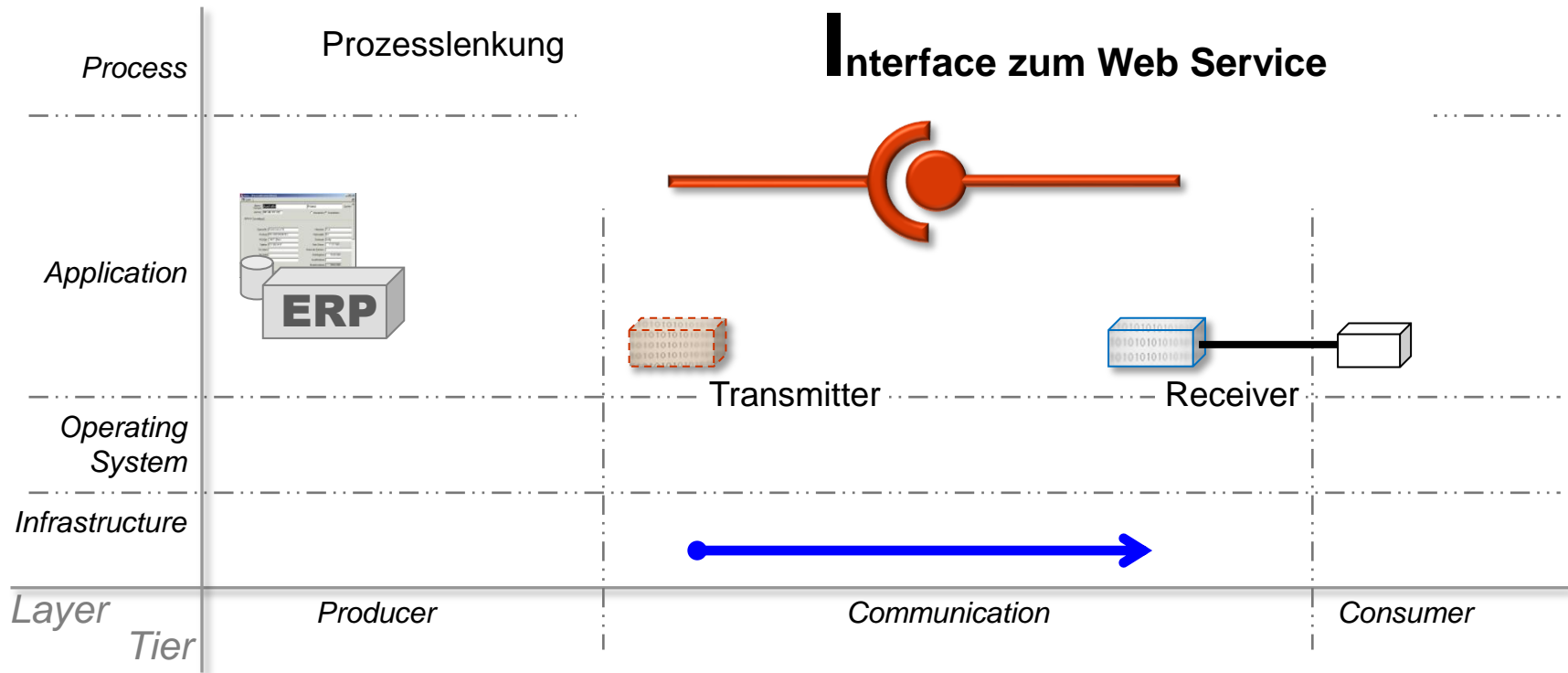
Zustandsautomat:
{protocol}



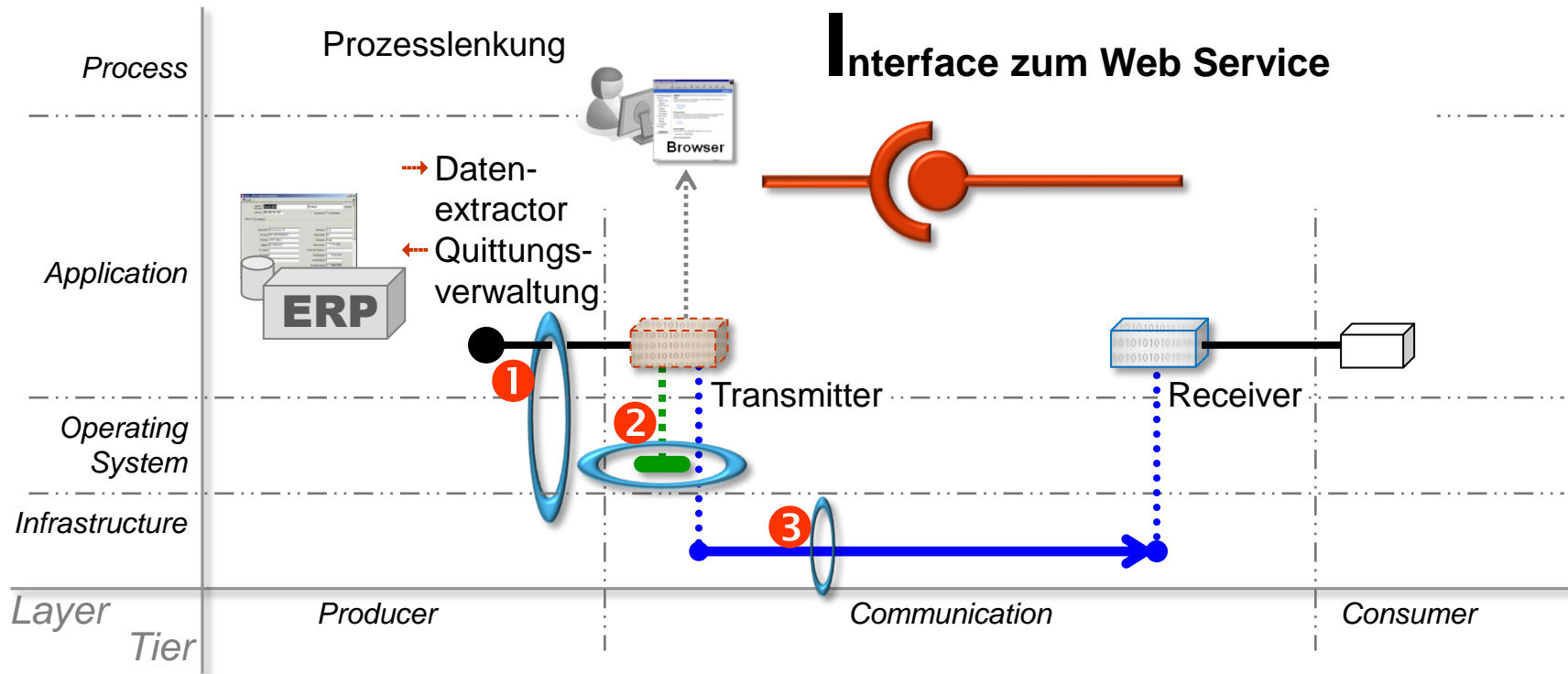




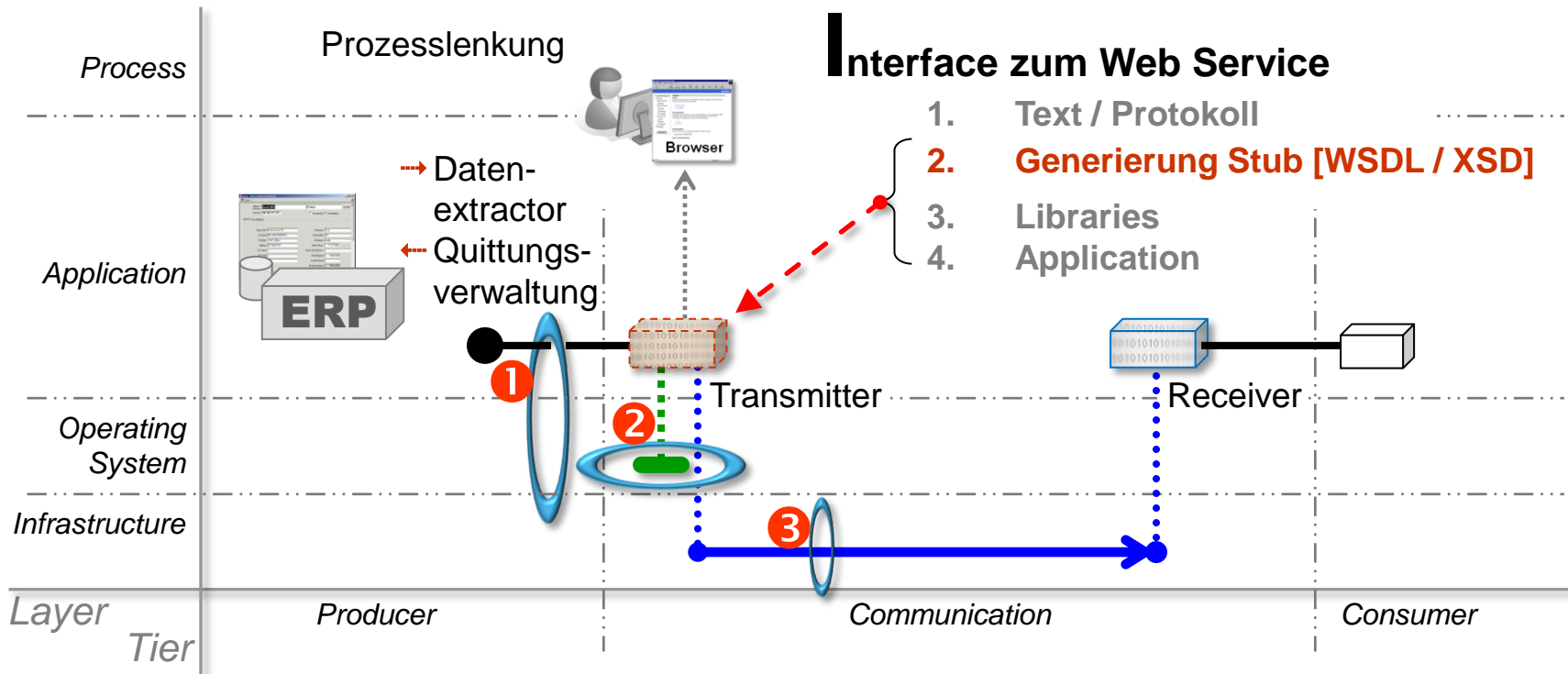
Die optimalste Integration in die bestehende IT-System-Landschaft: Baustein „WSDL / XSD“ (oder warum keine „Apps“)



Die optimalste Integration in die bestehende IT-System-Landschaft: Baustein „WSDL / XSD“ (oder warum keine „Apps“)



Die optimalste Integration in die bestehende IT-System-Landschaft: Baustein „WSDL / XSD“ (oder warum keine „Apps“)





Die **Interoperabilität** wird durch die Generierung von Codes gewährleistet; d.h. keine **Portabilitätsprobleme** bei Libraries oder Applikationen in den Unternehmen.

Siehe Architektur-Skizze mit den drei Schnittstellen:

- 1 Applikation – Transmitter (File, Pipes, *Interprocess*, ...)
- 2 Transmitter läuft auf unterschiedlichen *Betriebssystemen* und Versionen
- 3 Kommunikation ins Internet über *Proxies, Firewalls*, ... sicherstellen

Nutzen durch den Distributor und „ein“ XML

- Aus Sicht des Unternehmens:
Einfach alles „Jemandem“ übergeben → „gsorgät Gä“ Distributor

- Verminderung von **Daten-** und **Prozessredundanz**
- **Design-Firewall** (z.B. Mapping von verschiedenen Versionen)
- **Dynamische QS**, Datenfilterung und damit Datenschutz
- **Einfachere** Entwicklung und Produktion
(statt sehr viele „n:m“ nur „n:1:m“ Verbindungen)
- *Sicherheits-Anforderung von allen Seiten!*
Keine Datenspeicherung auf dem Distributor,
d.h. die **Kommunikation** zwischen Unternehmen und
Versicherern oder Behörden verläuft in „**Echtzeit**“ (7x24).


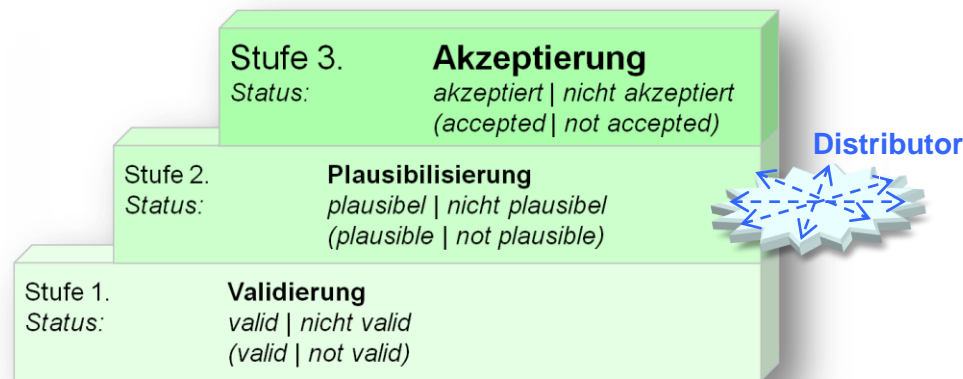
Prozessqualität, Sicherheit und Datenschutz

Das **Vertrauen** aller Teilnehmer in den gesamten Geschäftsprozess ist zwingend!



Folgende Massnahmen unterstützen dies (nicht abschliessend):

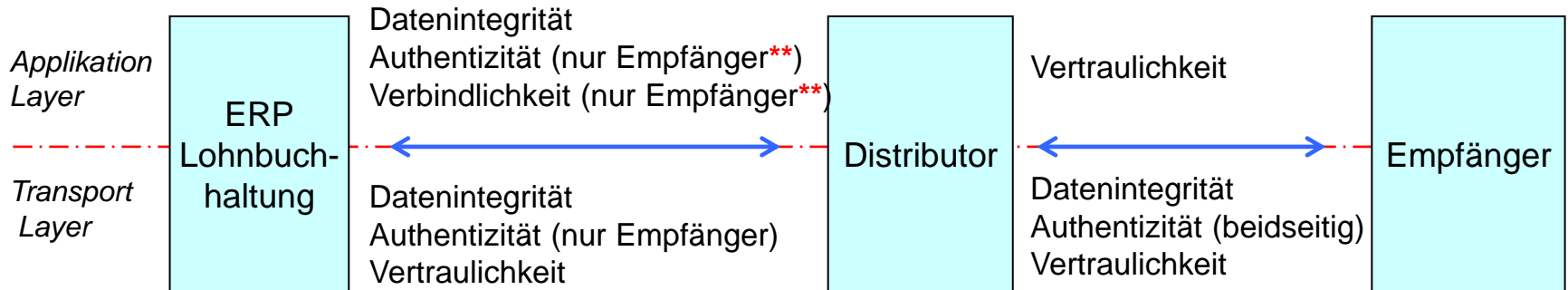
- Neben dem sicheren Transport über https (SSL/TLS) werden die Meldungen zusätzlich **signiert** und ein zweites Mal **verschlüsselt**.
- **Kontrolldaten und 3-stufige Qualität** in der Übermittlung (dynamische QS)



- **Zertifizierung** der Software-Lösungen werden wiederkehrend durchgeführt (langfristige QS)
- Prozess-Sicherheit durch weitere **separate Schritte** wie die Kontrolle in der Rechnungsstellung oder zusätzliche Regeln in der Verarbeitungslogik.

Sicherheit

- Schutzzielabdeckung
 - In Zukunft** könnte zwischen Lohnbuchhaltung und Distributor die Authentizität und Verbindlichkeit **beidseitig** werden.

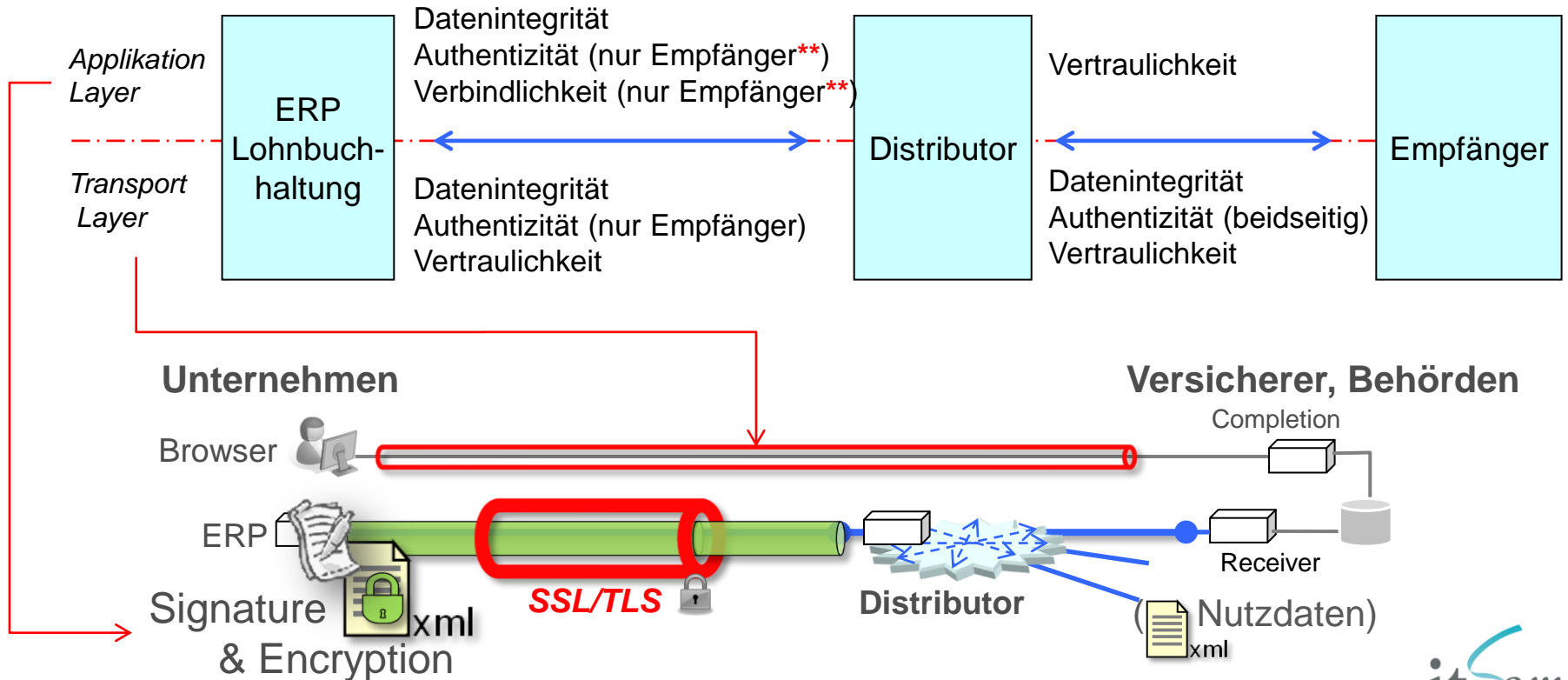


Sicherheit z.B.

https und WS Signature & Encryption

Web Services Security Standard von Oasis [WSS]

- Schutzzielabdeckung
 - In Zukunft** könnte zwischen Lohnbuchhaltung und Distributor die Authentizität und Verbindlichkeit **beidseitig** werden.



Grosse Datenmengen „XXL-ELM“

Mengen (2'000 Personen)

- **20'000** Personen
„synthetische“ Daten ~**180 MB**
(Sig./Enc. ~250 MB)
- Aus der Produktion 2010/11:
18'977 Personen **48'828 KB**
(nur eine Domäne)

Transportproblem von grossen
XML-Daten über das **Internet**:

- Vom Unternehmen zum Distributer und zum Endempfänger entspricht einer **Upload-Richtung**, die im Internet oft weniger performant ist.
- Längere Verarbeitungszeit ohne Datenverkehr führt bei Proxies und anderen Netzknoten oft zu **Timeouts**, da HTTP zustandslos ist.
- Signieren und Ver-/Entschlüsseln wird bei den meisten Plattformen nur mit internem **Hauptspeicher** gelöst.

Lösungskonflikt:

- **Hauptspeicher-** und/oder **Protokoll-Erweiterung**

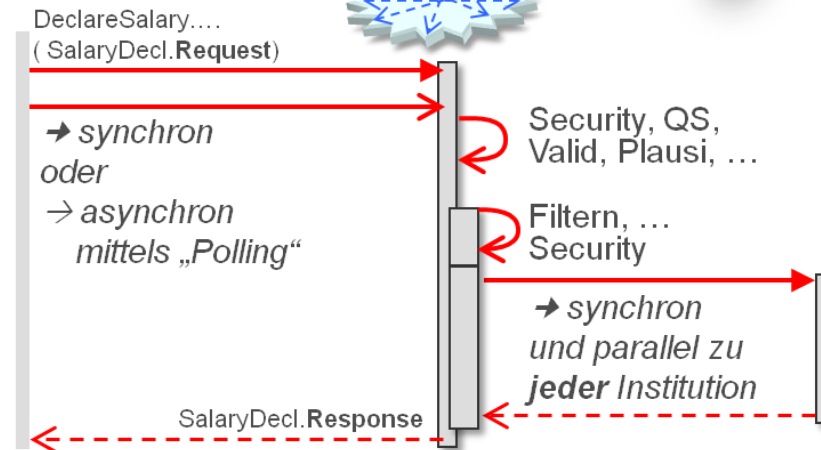
Achtung:

*Komplexe Transformation auf dem
Distributor bei der **Protokoll-Erweiterung***

Unternehmen

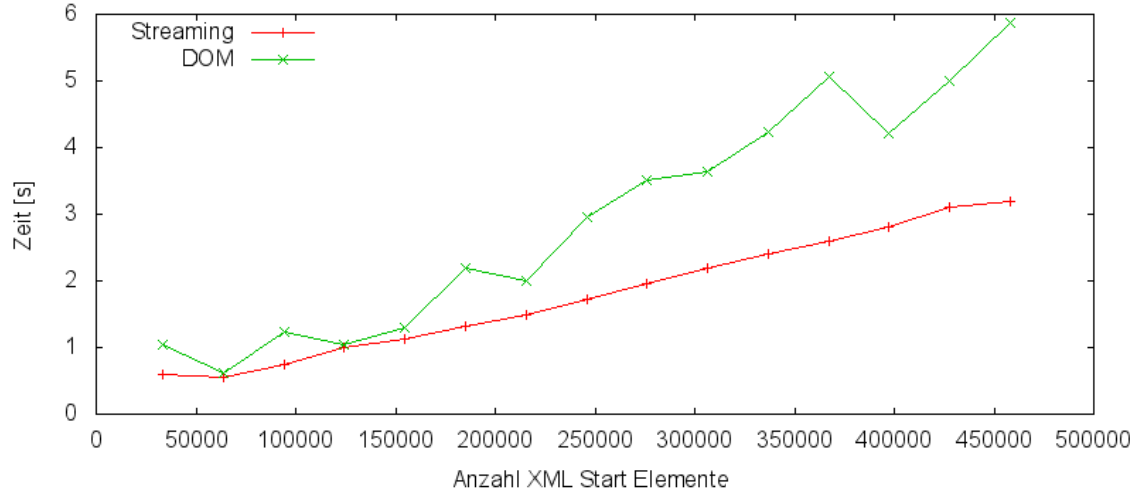
Distributor

Versicherer
oder Behörde

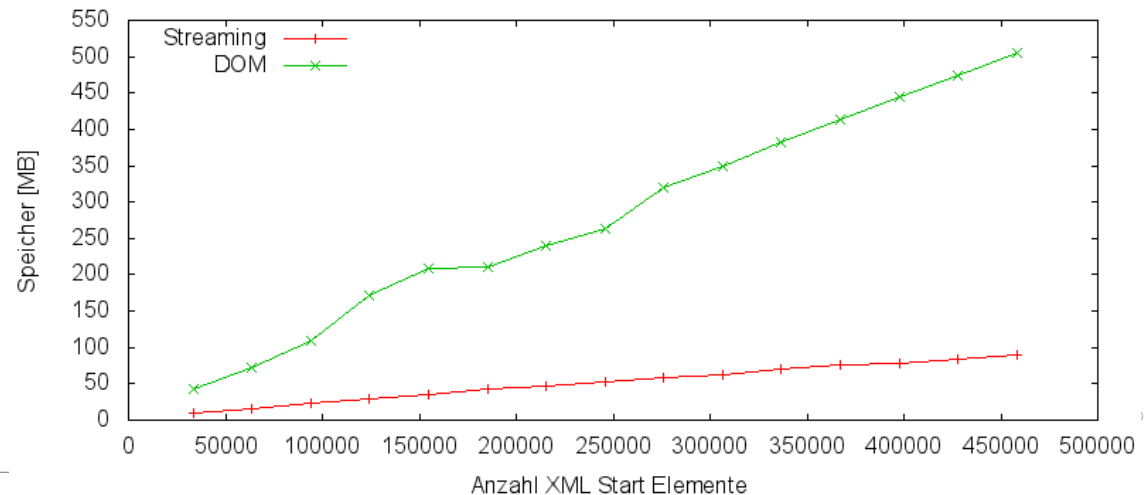


Streaming-WebService-Security-Framework (swssf) von Marc Giger

Verarbeitungszeit zum Verschlüsseln (XENC)

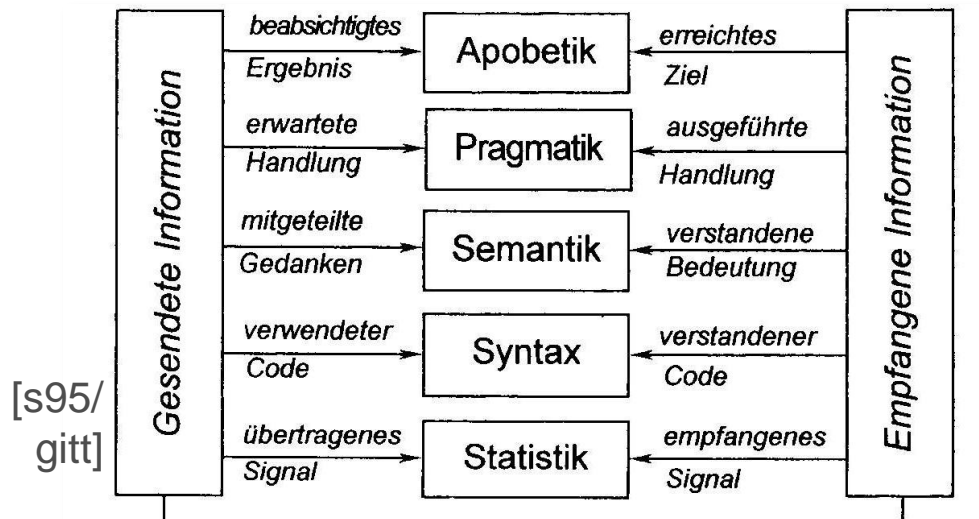


HEAP Speicherverbrauch zum Verschlüsseln (XENC)



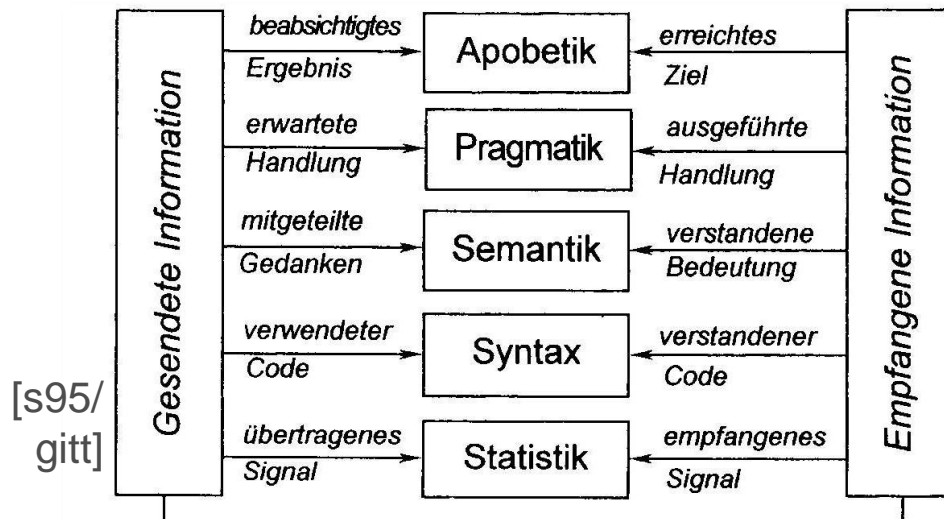
Bemerkungen

- Produktionserfahrungen
Operationen: „Ping“, „CheckInteroperability“ (Sig/Enc),
Lohnmeldung mit Element <TestCase>, XSLT auf Distributor, ...
- **XML** und **XSD** auch mit **Fachspezialisten** verwenden
- **XSD** Thema „Zeichenkettenlänge“ und „sprechende“ Codes
- XSD und **WSDL**, bzw. Daten & Prozesse
- Synergien z.B. Tool Viewgen für Listen und Lohnausweis mit 2D-Barcode
- **Spezifikation** und **Werkzeuge** (Referenzsysteme mit „dyn. Spez.“)



Bemerkungen

- Produktionserfahrungen
Operationen: „Ping“, „CheckInteroperability“ (Sig/Enc),
Lohnmeldung mit Element <TestCase>, XSLT auf Distributor, ...
- **XML** und **XSD** auch mit **Fachspezialisten** verwenden
- **XSD** Thema „Zeichenkettenlänge“ und „sprechende“ Codes
- XSD und **WSDL**, bzw. Daten & Prozesse
- Synergien z.B. Tool Viewgen für Listen und Lohnausweis mit 2D-Barcode
- **Spezifikation** und **Werkzeuge** (Referenzsysteme mit „dyn. Spez.“)



Beispiel Lohnmeldung:

Apobetik:

Ich bin optimal „gesichert“, d.h.

- Steuer **wenig** Lohn
- Versicherer **viel** Lohn

Pragmatik:

- Person versichern oder besteuern

Semantik:

- *Mein Jahreslohn*

... und zum Ende

Besten Dank für
Ihre Aufmerksamkeit 😊



You don't
need to *see*
you need
vision

Haben Sie Fragen?