2.– 5. September 2013
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

# Die Register Factory

## Industrialisierte Entwicklung von betrieblichen Informationssystemen

Ramon Anger
Capgemini

# Register Factory

## Industrialisierte Entwicklung von betrieblichen Informationssystemen

Ramon Anger - Capgemini

https://commons.wikimedia.org/wiki/File:Weissenau_Holzhey-Orgel_Register_links.jpg

## EGO

41 Jahre, verheiratet, ein Kind

seit 2011 als Technischer Architekt bei Capgemini
davor Mitarbeiter bei debis, T-Systems, TomTom, Materna

Agiler Chaot, Entwickler, Architekt, Teamleiter, Technischer Projektleiter, Technischer Consultant, Head of Development

Diplom-Informatiker (FH), Master of Computer Science (M. Comp. Sc.)

Schneeberg, Erzgebirge
@ramonanger, Blog: ramonanger.com/architecture

http://commons.wikimedia.org/wiki/File:Snow_removal.jpg

# Show hands …

01/05/2010

https://commons.wikimedia.org/wiki/File:Stadtpokalsieger_2010_Magdeburg.JPG

# Was ist ein Register …

Synonym für Verzeichnis

systematische Sammlung von Informationen über eine Gruppe von Objekten

„Datenspeicher“

Beispiele:

- Handelsregister
- Fahrzeugregister
- Güterrechtsregister
- Grundbuch
- Krebsregister
- Melderegister
- Partnerschaftsregister
- Urkundenregister
- …

http://commons.wikimedia.org/wiki/File:N%C3%A1rodn%C3%AD_zem%C4%9Bd%C4%9Blsk%C3%A9_muzeum,_fotoarchiv,_l%C3%ADstkov%C3%BD_katalog.JPG

# Es gibt viele Vorgaben für die Bau von Systemen. Eine Factory soll helfen, Vorgaben umsetzbar zu machen.

- Ernährungsratgeber
- Lebensmittelempfehlungen

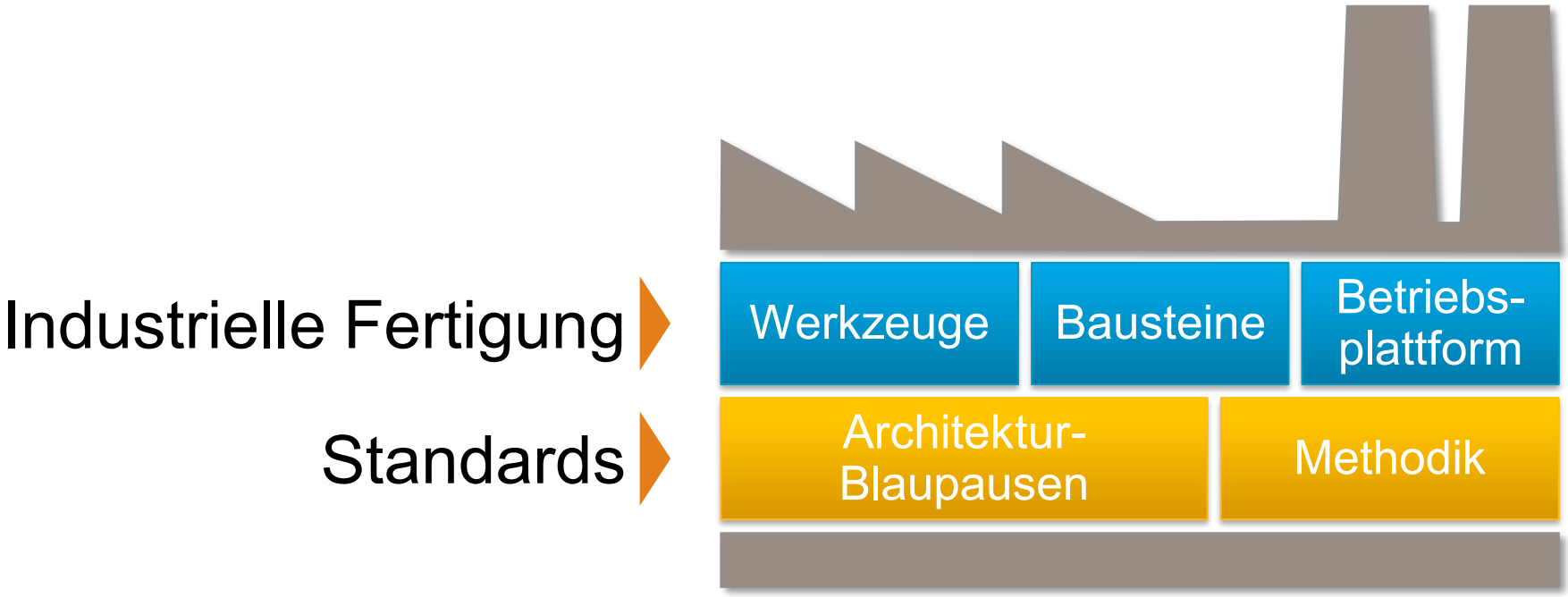
- SAGA
- Common PKI
- HTML 5

- Kochrezept,
- Fonds, TK-Zutaten,
- Töpfe, Pfannen & Kochlöffel

zusammenpassend

- Register Factory

# Der Factory-Ansatz: Mit Hilfe von Patterns und Standards wird industrielle Fertigung erleichtert.



# Die Inhalte einer Factory sind aufeinander abgestimmt und sofort einsetzbar.

## Werkzeuge

- Vorkonfigurierte SEU
- Generatoren, Validatoren
- Templates und Checklisten

## Bausteine

- Querschnittlich nutzbare Services
- Wiederverwendbare Bibliotheken

## Betriebsplattform

- Vorhalten einer standardisierten Betriebsplattform
- Standardisierter Betrieb

## Architektur-Blaupausen

- in sich stimmige Technische Architektur für betriebliche Informationssysteme
- Facharchitekturen für bestimmte Domänen

## Methodik

- Standards und Best Practices für Software Engineering
- Vorlagen und Anleitungen

# Mit der Register Factory wurden seit sechs Jahren sehr gute Erfahrungen gemacht.

## Systeme im Registerportal des Bundesverwaltungsamts

- ca. ein Dutzend Systeme mit der Register Factory gebaut, auch sehr große, anspruchsvolle Systeme.
- 80.000 Nutzer in 1600 Behörden, 24/7
- 29 Mio. Geschäftsvorfälle/Jahr
- 11–100 Mio. Datensätze

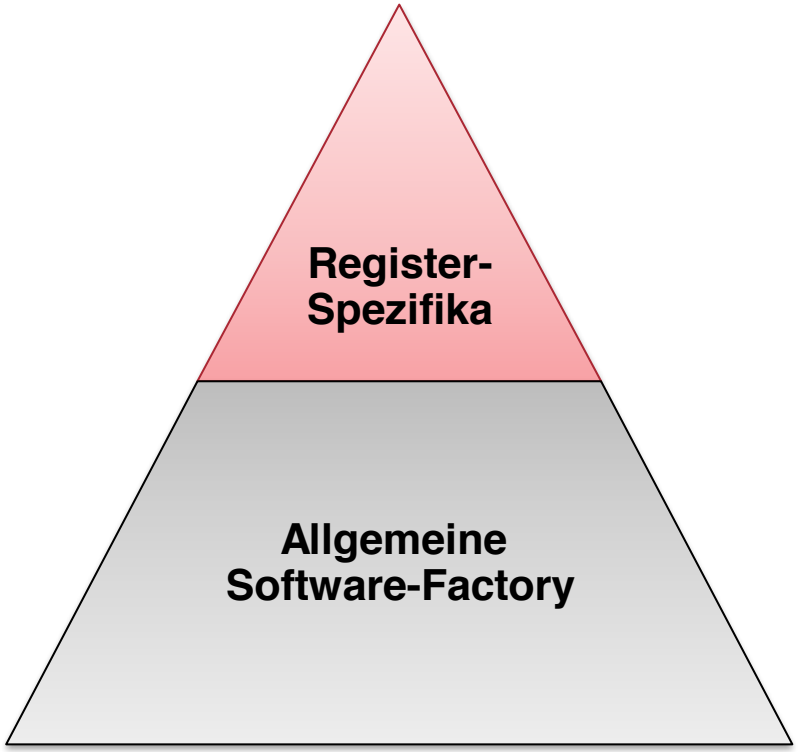
## Weitere Einsätze

- Bundesverwaltungsamt Aufbau einer weiteren Factory für kleinere Fachverfahren
- Bayerisches Landesamt für Umwelt
- Landesamt für Zentrale Polizeiliche Dienste NRW

## Erfahrungen beim BVA

- Der Factory-Ansatz wirkt.
- Zufriedener Kunde BVA, zufriedene Mitarbeiter

# Die allgemeine Factory ist von Register-Spezifika getrennt und kann universell eingesetzt werden.




## Register-Spezifika

- Fachliche Referenzarchitekturen für Registeranwendungen
- Spezielle Bausteine und Services für den Bau von Registern

## Allgemeine Software-Factory

- Technische Referenzarchitektur für Unternehmenslandschaften, Softwaresysteme, Kleinanwendungen
- Allgemeine Programmierwerkzeuge
- Methodische Vorgaben
- Wiederverwendbare Bausteine

# Die allgemeine Factory ist von Register-Spezifika getrennt und kann universell eingesetzt werden.

Kann das aus Ihrer Sicht funktionieren?
Wo liegen die Grenzen, Vorteile, Nachteile …
Diskutieren sie …

http://commons.wikimedia.org/wiki/File:Circuit_Breaker_with_malfunction.JPG

# Mit den falschen Zutaten oder dem falschen Rezept lässt sich statt einem Entenbraten …

http://commons.wikimedia.org/wiki/File:Roast_duck_with_dumplings_%28Czech%29.JPG

… vielleicht nur ein Hamburger zubereiten.

**Bestellung, Zutaten und Rezept müssen zusammen passen.**

http://commons.wikimedia.org/wiki/File:Hamburger_%281%29.jpg

# Die Register Factory wurde aus der Praxis auf der Basis von drei wesentlichen Leitgedanken entwickelt.

**Einfach**

- Die Architektur ist bewusst einfach gehalten. Wenn es zwei Alternativen gab, wurde immer die einfachere gewählt.
- Was weggelassen werden konnte, wurde weggelassen.
- Keine Abstraktion, keine Bibliotheken, keine Komplexität auf Vorrat.

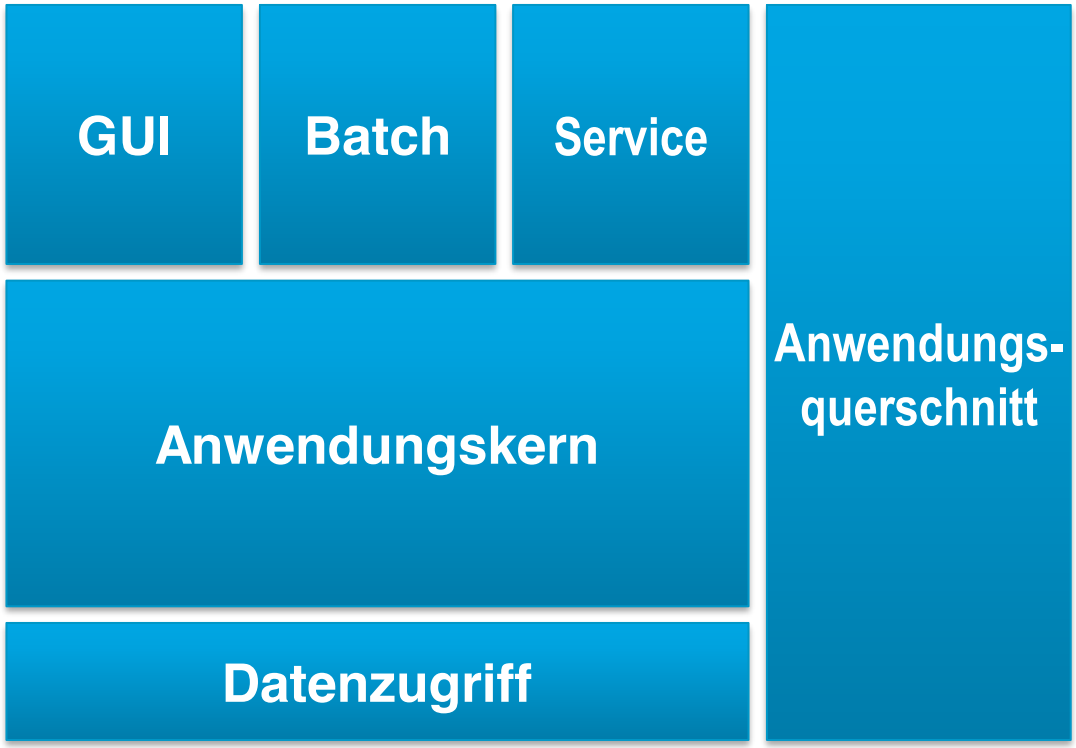
**Standards & OpenSource**

- Keine proprietären Bibliotheken und Werkzeuge, keine Herstellerabhängigkeit.
- Einsatz von weit verbreiteten Standards sorgt für eine breite Know-how-Basis und Zukunftssicherheit.

**Patterns statt Frameworks**

- Vermitteln der Architektur und Umsetzung durch Beispiele, Tutorials, Kopiervorlagen.
- Keine Anwendungsframeworks mit abstrakten Oberklassen, Interfaces
- Komplette Beispielanwendung zum Lernen und Abgucken.
- Dadurch hohe Flexibilität und Anpassbarkeit beim Einsatz

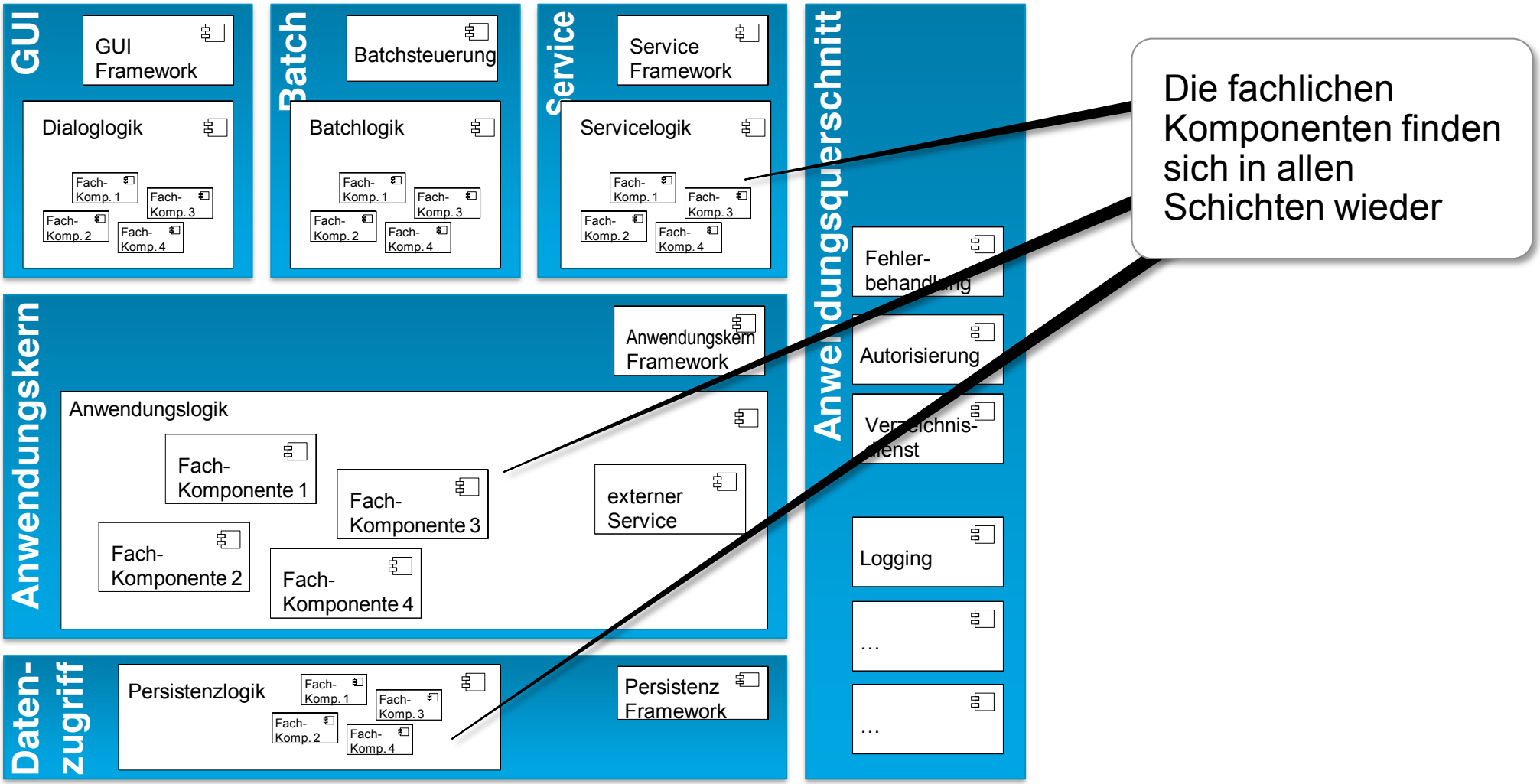
# Die technische Architektur ist für alle Anwendungen nach Register Factory-Architektur identisch.




## Techn. Architektur einer Anwendung

- 3-Schichten-Architektur
- Datenzugriffsschicht: Persistierung und Lesen von Daten
- Anwendungskern: eigentliche fachliche Funktionalität des Systems durch Nutzungsschicht gekapselt.
- Nutzungsschicht, Typen:
  - GUI: Web-Dialoge
  - Batch: Schritte von Batch-Jobs
  - Service: Anwendungskern-Funktionalität für andere Anwendungen bereitstellen
- Die Transaktionssteuerung (fachliche Transaktionen) erfolgt durch die Nutzungsschicht

# In den einzelnen Schichten sind Fachlichkeit und technische Frameworks strikt getrennt.



# Viele der eingesetzten Produkte sind Open Source.

**Persistenz**

- Oracle RAC
- Hibernate

**Anwendungs-kern**

- Spring
- JBoss Drools, JBoss jbpm
- Log4J
- Dozer

**GUI**

- Spring WebFlow
- JSF

**Service**

- Spring httpInvoker
- metro

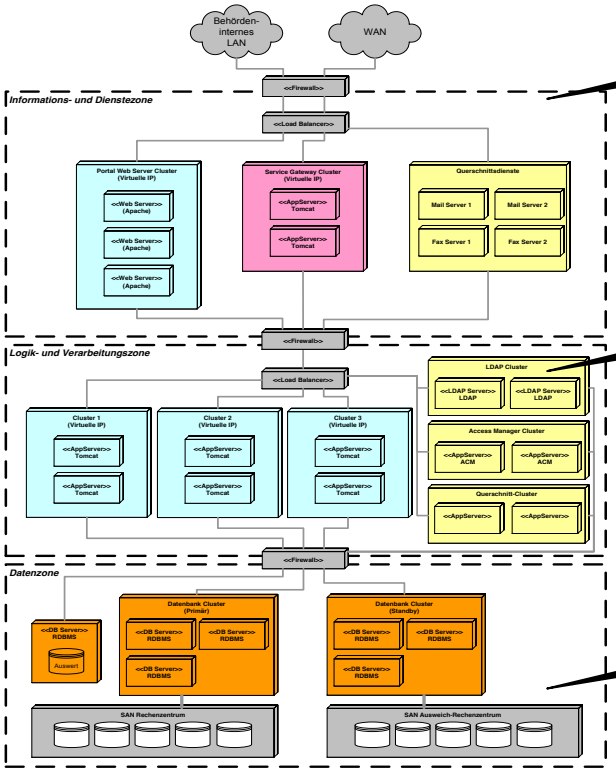
**Basis**

- Tomcat
- Apache httpServer
- SUSE Linux Enterprise

**SEU**

- Eclipse
- Maven
- Jenkins

# Die Referenzarchitektur zur Technischen Infrastruktur baut auf den Vorgaben von SAGA und des BSI auf.




**Informations- und Dienstezone**

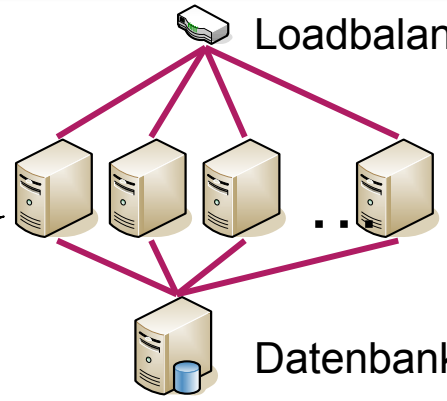
**Logik- und Verarbeitungszone**

**Datenzone**

## Technische Infrastruktur

- Die Technische Infrastruktur setzt die Zonen aus SAGA um
- Jedes System wird in einem Cluster betrieben, bei dem hinter einen Loadbalancer beliebig viele Server geschaltet sind. Möglich durch Stateless Server-Architektur.
- Als Ablaufumgebung wird Apache Tomcat (WebContainer) genutzt.

**Clusterarchitektur (logische Sicht)**



# Die Unternehmensarchitektur der Register Factory unterscheidet zwischen verschiedenen Systemtypen.

Das Portal ist der zentrale Zugangspunkt für Personen, die einen Geschäftsprozess bearbeiten

Portal

Die Geschäftsanwendung bildet in der Regel ein Verfahren ab, das mit den Daten z. B. von Registern arbeitet.

Geschäfts-anwendung

Ein Register ist ein Datenspeicher und in verschiedenen Verfahren nutzbar.

Register

Service Gateway

Das Service Gateway ist die Schnittstelle zu anderen Anwendungen außerhalb der Plattform.

Q-System: Authentifizierung Autorisierung

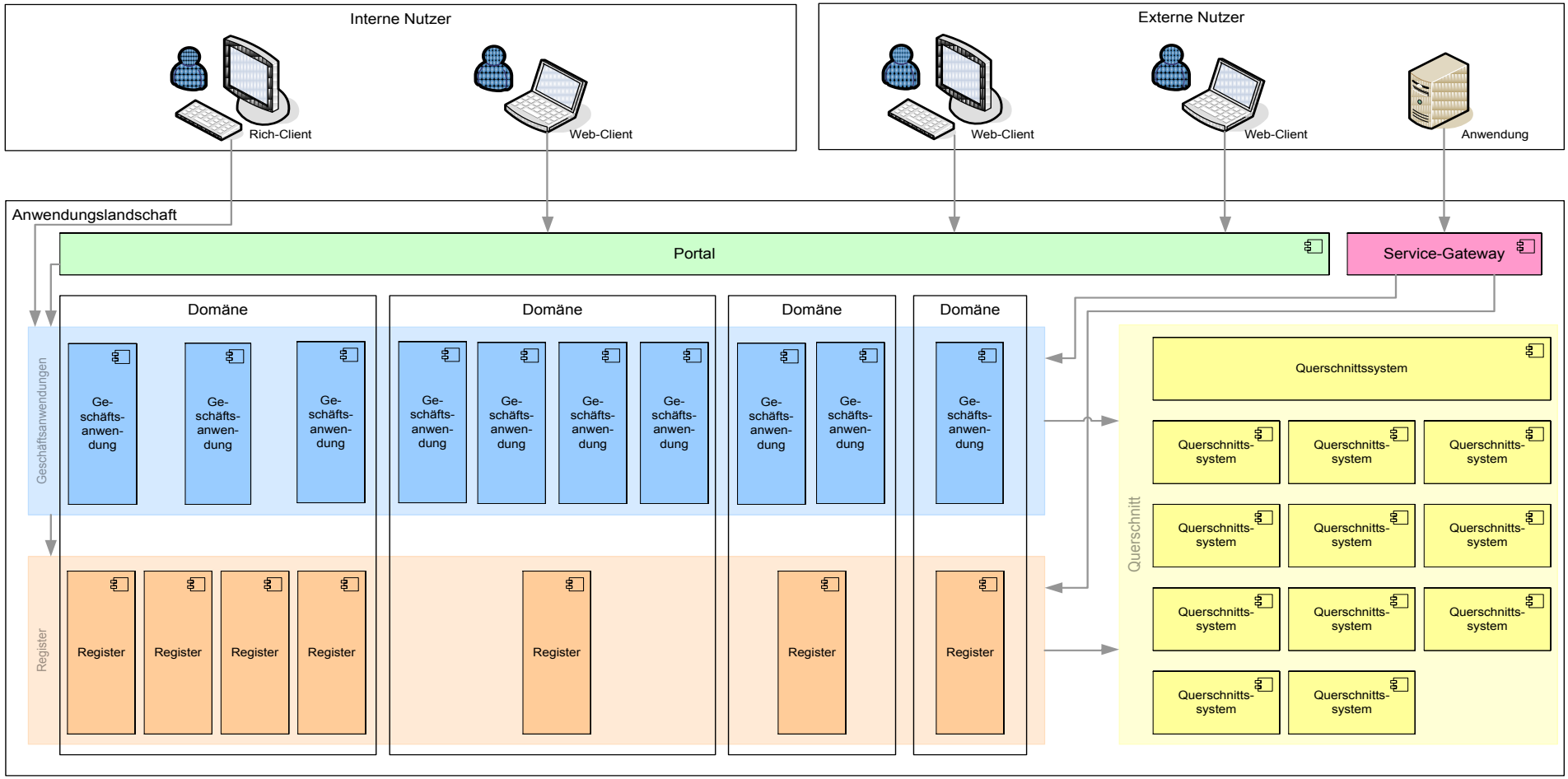
Q-System: Biometrie

Q-System: Schlüssel-Verzeichnis

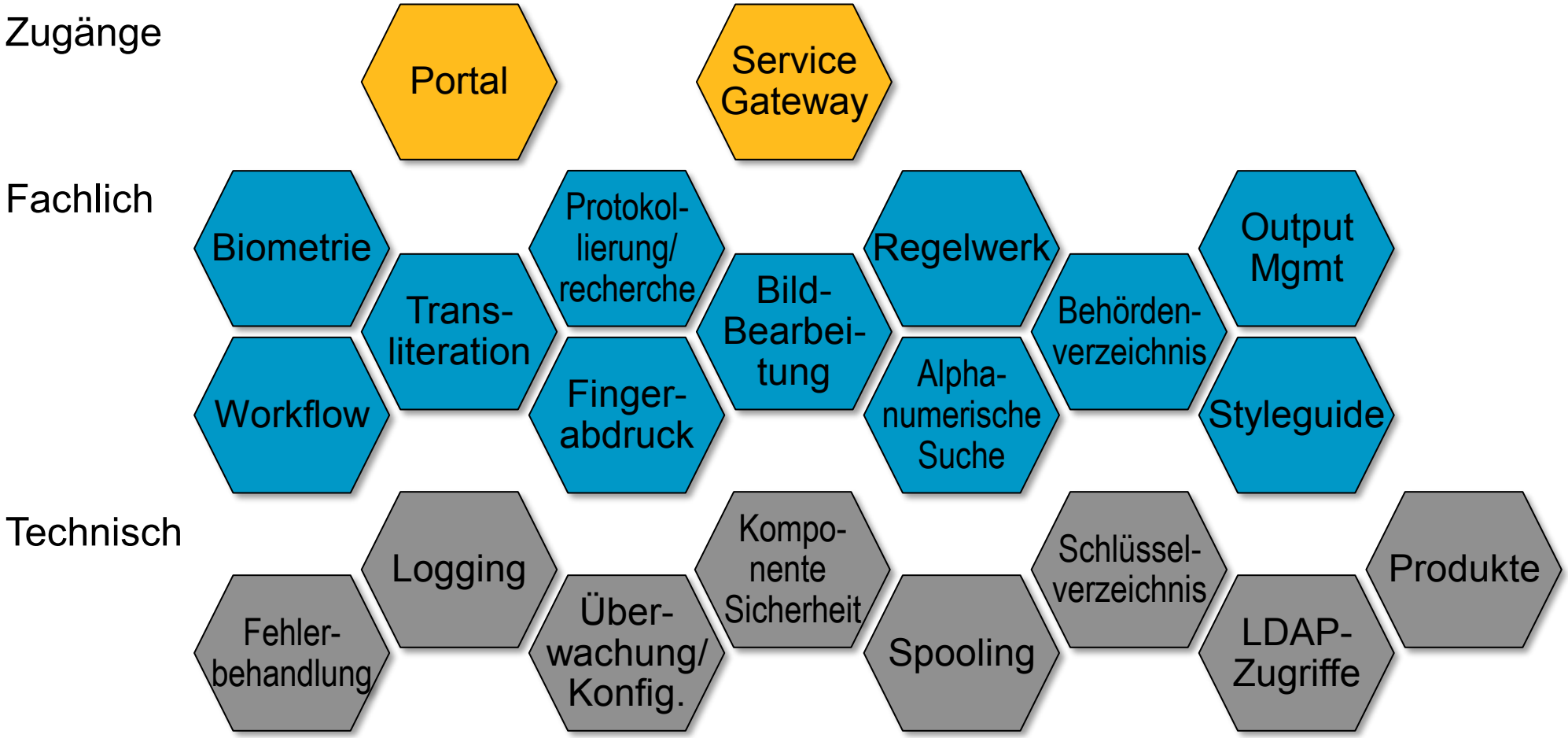
Q-System: Personen-suchverfahren

Verschiedene Querschnittssysteme unterstützen den Geschäftsprozess

# Auf Basis der Register Factory kann eine ganze Anwendungslandschaft aufgebaut werden.



# Die Register Factory umfasst technische und fachliche Bausteine sowie verschiedene Zugänge



# Anwendung von Einfachheit, Standards, OpenSource, Patterns statt Frameworks …

**Einfach**

- Die Architektur ist bewusst einfach gehalten. Wenn es zwei Alternativen gab, wurde immer die einfachere gewählt.
- Was w…elassen werden konnte, wurde weggelassen.
- Keine …ine Bibliotheken, keine Komplexität auf Vorrat.

**Standards & OpenSource**

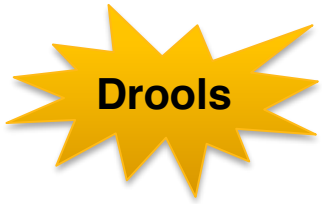
- Keine prop… …ge, keine Herstellerabhängigkeit.
- Einsatz von weit verbr… …eite Know-how-Basis und Zukunftssicherheit.

**Patterns statt Frameworks**

- Vermitteln der Architektur und Umsetzung durch Beispiele, Tuto…
- Keine Anwendungsframeworks mit abstrakten Oberklassen, Interfaces, etc.
- Komplette Beispielanwendung zum Lernen und Abgucken.
- Dadurch hohe Flexibilität und Anpassbarkeit beim Einsatz

Entspricht das dem Factory-Gedanken?
Welche Vorteile/Nachteile ergeben sich?

# Erfahrungen zu speziellen High- und Low-lights der eingesetzten Technologien gesammelt.




- Sehr positive Erfahrungen mit JBoss Drools
- Performant, man muss aber wissen, wie man die Regeln organisiert.

- Absolut problemloses Kommunikationsmedium.
- Viel, viel einfacher als WebServices

- Reichte uns als Ablaufumgebung völlig aus. Performant, leistungsfähig, gut zu betreiben.

- Steter Quell von Problemen.
- Kernfunktionalität geht, alles darüber hinaus ist Abenteuer
- Abgelöst durch metro.

- Version 5 funktionierte einfach nicht unter hoher Last.
- Legte RAC lahm

# Die Register Factory enthält Teile, die den Bau von Registern sehr viel effizienter machen.

## Eigenschaften eines Registers

**Änderung durch Meldungen**

- Änderungen erfolgen über Meldungen, keine Sperren zum Schreiben von Daten

**Sichtbarkeiten**

- Ein Register wird von vielen Nutzern zu unterschiedlichen Zwecken genutzt.
- In einem Register sind mehr Daten enthalten, als für einen Bearbeiter zu einem Anfragezweck sichtbar ist.

**Nachvollziehbarkeit**

- Datenänderungen und deren Informationsgrundlage
- Abfragen

## Fachliche Referenzarchitektur

- Anwendungslandschaft
- einzelne Anwendung

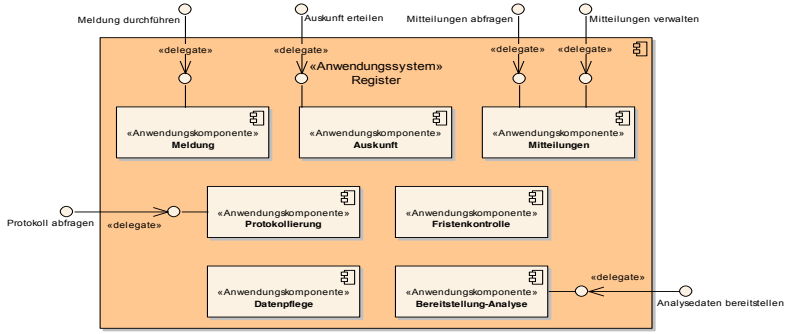
## Spezielle Bausteine

- Biometrische Suche
- Alphanumerisches Suchverfahren
- Spezielles Access Management nach BVA-Vorgaben

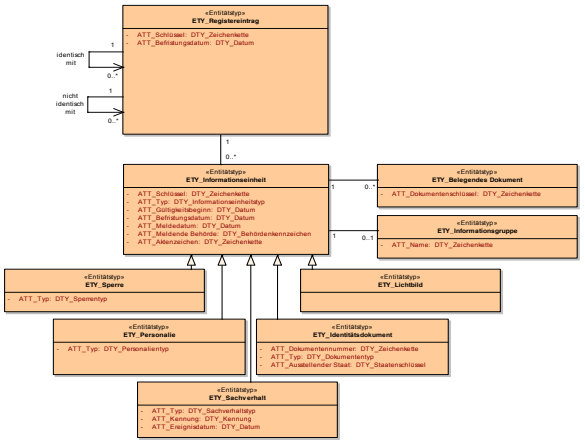
## Denkbar: Methodik

- Konkretisierte Spezifikationstemplates speziell für Register

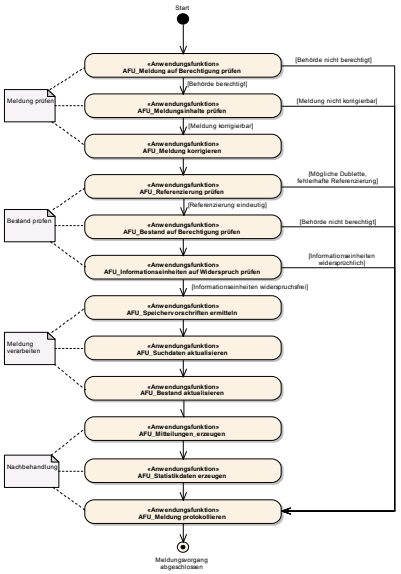
# Die Referenzarchitektur „Register“ enthält Vorlagen für Datenmodelle, Komponenten und Abläufe in Registern.




Komponentenschnitt




Datenmodell des Bestands

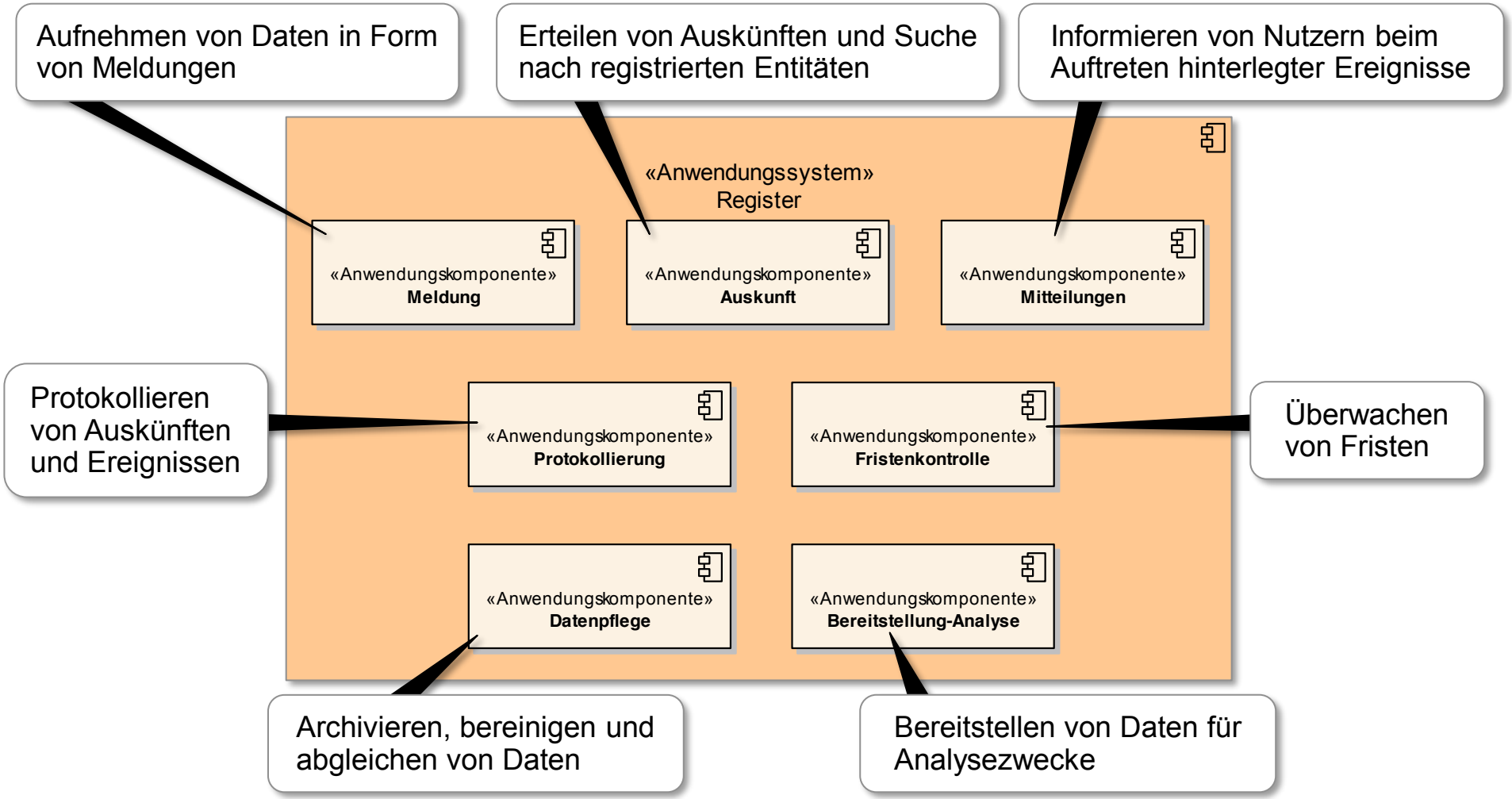



Durchführen einer Meldung

## Referenzarchitektur Register

- Die Referenzarchitektur „Register“ gibt einen Bauplan für den Bau eines Registers vor.
- Die Aufgaben eines Registers sind im Wesentlichen das Verwalten der Daten und die Verarbeitung von Meldungen und Auskünften.
- Wichtige Elemente der Referenzarchitektur sind
- Komponentenschnitt
- Datenmodelle
- Schematisierte Abläufe

# Der fachl. Komponentenschnitt der Referenzarchitektur „Register“ gibt sieben Komponenten vor.



# Die Register Factory hat die Wartung der Anwendungslandschaft beim BVA deutlich vereinfacht.

Die Wartung der Anwendungen ist unproblematisch

- Einfache Architektur macht die Wartung einfacher
- Standardisierung von Technologie und Vorgehen ermöglicht schnellen Wechsel von Mitarbeitern

# Die Wartung der Register Factory verursacht geringen Aufwand, der durch Projekte getragen wird

- Größere Erweiterungen waren bisher:
  - Kommunikation mittels OSCI Transport 2.0
  - Integration einer Workflow-Maschine
  - Ergänzung der Web-GUI durch Swing-GUI
- Kleinere Erweiterungen werden nebenbei erledigt, z.B.:
  - Versionswechsel Hibernate, Spring etc.
  - Fehlerkorrekturen, Best practices
- Aufwändig ist das Nachführen der Beispielanwendung
  - Beste Lösung dafür: Einsatz von Studenten (Graduierungsarbeiten, Praktika, etc.)

# Es dauert nur wenige Tage, bis neue Kollegen im Projekt produktiv sind.

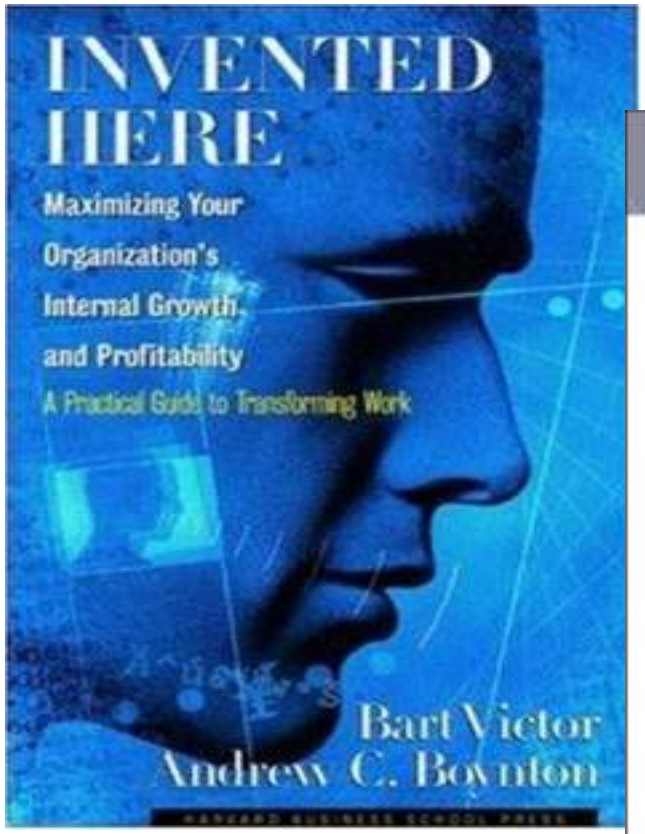
- Für die Einarbeitung kalkulieren wir 4-5 Tage (incl. Fachlichkeit)
  - Architektur ist modular, man muss nicht alles kennen
  - Viele Dinge können sich die Mitarbeiter bei Bedarf und im Selbststudium aneignen
  - Beispiele sind einfach zu verstehen und zu adaptieren
- In der Vergangenheit haben mehrere Studenten ihre Graduierungsarbeit (1 Bachelor, 5 Master, Praktikum, Werkstudenten) im Kontext Register Factory gemacht. Sie alle konnten sich einarbeiten und Weiterentwicklungen vornehmen.
- An der Uni Kiel und an der TU Darmstadt bauten und bauen Studentengruppen Anwendungen mit der Register Factory

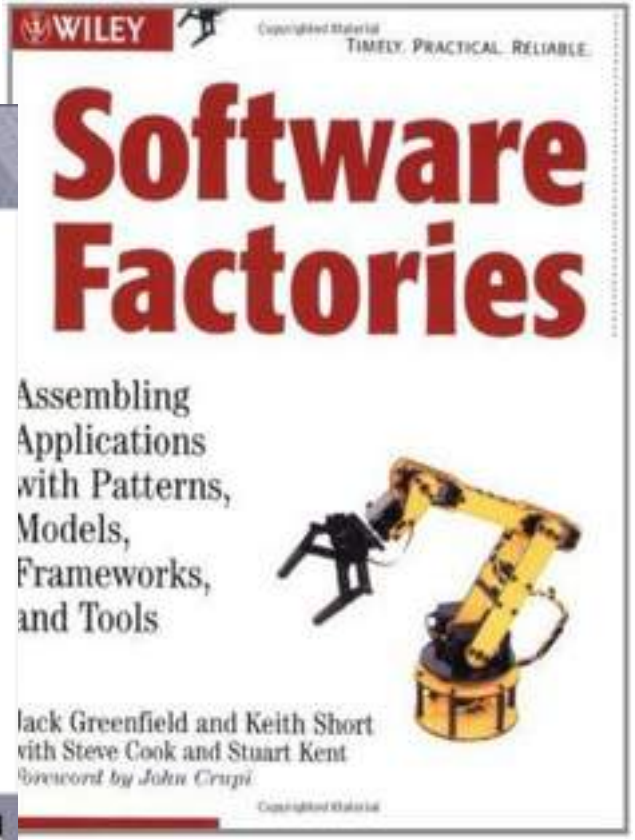
# Die Register Factory ist eine Wort-Bild-Marke des BVA, kann aber auch weitergehend genutzt werden.

- Die Register Factory ist eine Wort-Bild-Marke des BVA
- Vorhandene Bibliotheken dürfen im öffentlichen Bereich weiterverwendet werden (wie Open Source)
- „echtes“ Open Source ist in Vorbereitung
- Beispiele, Vorlagen, Architekturen (Patterns) stammen aus Quasar und können genutzt werden.

Register Factory®

# Wem das noch nicht reicht ...





# Bei Fragen bitte fragen!

https://commons.wikimedia.org/wiki/File:Weissenau_Holzhey-Orgel_Register_links.jpg

# Vielen Dank für ihre/eure Aufmerksamkeit!

https://commons.wikimedia.org/wiki/File:Weissenau_Holzhey-Orgel_Register_links.jpg