



UNIVERSITÄT  
LEIPZIG

Medizinische Fakultät

Architektur von IS im GW / Informationsmanagement in der klinischen Forschung

# **Transformation eines CSV-Datensatz nach HL7 FHIR**

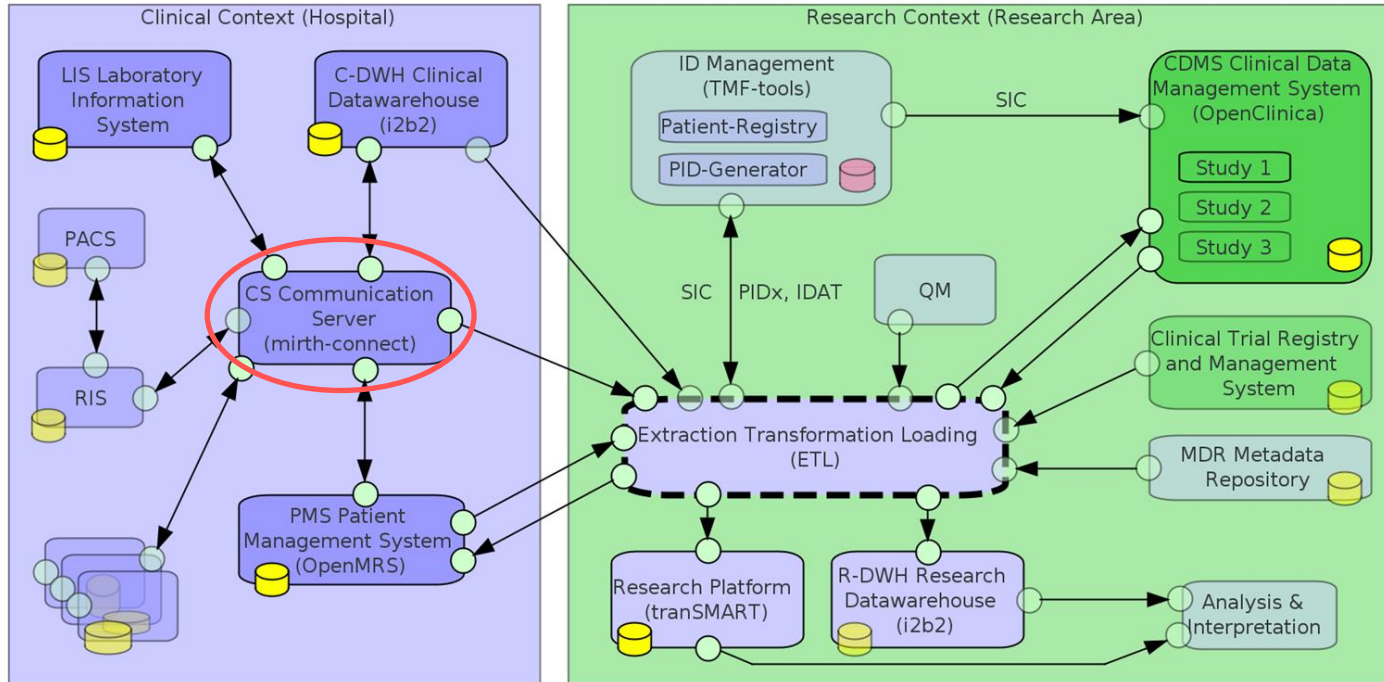
Leipzig, Datum

**imise.**

# KOMMUNIKATIONSSERVER

- **Funktion:**
  - Zentrale Verteilung von Nachrichten
  - Filter
  - Transformation
  - Vermeidung von  $n*m$  Kommunikationsschnittstellen
- **Software-Produkte:**
  - eGate
  - Kodix (UKL)
  - **NextGen Connect** (früher Mirth Connect)

# KOMMUNIKATIONSSERVER



# FAST HEALTHCARE INTEROPERABILITY RESOURCES (FHIR)

- bestehend aus konsistenten, einfach zu verstehenden Ressourcen
- 80-20 Prinzip, der Standard (momentan R4) deckt die meisten Anwendungen ab, Rest durch Extensions
- Web-Technologie (Restful API) + human readability
- OpenSource (Spezifikation, Tools, Server, Libs)
- Betreut von HL7
- Anwendung in Deutschland im Rahmen der MII (Kerndatensatz)

Patient	I	Patient
id	S Σ 0..1	string
meta	S Σ 0..1	Meta
identifier	S Σ 0..*	Identifier
active	Σ ?! 0..1	boolean
name	S Σ 0..*	HumanName
telecom	Σ I 0..*	ContactPoint
gender	S Σ 0..1	code Binding
birthDate	S Σ 0..1	date
deceased[x]	S Σ ?! 0..1	
address	S Σ 0..*	Address
maritalStatus	0..1	CodeableConcept Binding
multipleBirth[x]	0..1	
photo	I 0..*	Attachment
contact	I 0..*	BackboneElement
communication	0..*	BackboneElement
generalPractitioner	I 0..*	Reference(Organization   Practitioner   Practit...
managingOrganization	Σ I 0..1	Reference(Organization)
link	S Σ ?! 0..*	BackboneElement

[https://www.medizininformatik-initiative.de/Kerndatensatz/Modul\\_Person\\_Version\\_2/MIIGModulPerson-TechnischeImplementierung-FHIR-Profile-PatientInPatient.html](https://www.medizininformatik-initiative.de/Kerndatensatz/Modul_Person_Version_2/MIIGModulPerson-TechnischeImplementierung-FHIR-Profile-PatientInPatient.html)

## ZIEL DER ÜBUNG

- Transformation von CSV nach FHIR
  - Mapping nach FHIR mit MII-KDS
- Erstellen von Channels mit Mirth-Connect
  - Daten empfangen
  - Transformieren
  - Ausgabe von Daten im FHIR-Format
- Senden der FHIR-Daten an einen FHIR-Server
  - Überprüfung der transformierten FHIR-Daten mit ClinFHIR

# TRANSFORMATION VON CSV NACH FHIR MIT MIRTH-CONNECT

## ABLAUF

1. Eingangsdaten
  - CSV-Datensatz
2. Transformation mit Mirth-Connect
  - Mapping von Spalten der CSV auf FHIR-Attribute
  - Basierend auf FHIR Message Template
3. Ausgabe
  - JSON-Bundle im FHIR-Format
  - Zum Senden an einen FHIR-Server

# TRANSFORMATION VON CSV NACH FHIR MIT MIRTH-CONNECT

## FHIR-TEMPLATE: MII-PATIENT

Patient	I	Patient
id	S Σ 0..1	string
meta	S Σ 0..1	Meta
identifier	S Σ 0..*	Identifier
active	S Σ ?! 0..1	boolean
name	S Σ 0..*	HumanName
telecom	S I 0..*	ContactPoint
gender	S Σ 0..1	code Binding
birthDate	S Σ 0..1	date
deceased[x]	S Σ ?! 0..1	
address	S Σ 0..*	Address
maritalStatus	S Σ 0..1	CodeableConcept Binding
multipleBirth[x]	S Σ 0..1	
photo	I 0..*	Attachment
contact	I 0..*	BackboneElement
communication	I 0..*	BackboneElement
generalPractitioner	I 0..*	Reference(Organization   Practitioner   Practit...
managingOrganization	S I 0..1	Reference(Organization)
link	S Σ ?! 0..*	BackboneElement



## Template der zu sendenden FHIR Nachrichten

Outbound Message Template

Data Type: JSON Properties

```

{
  "resourceType": "Bundle",
  "type": "transaction",
  "entry": [
    {
      "resource": {
        "resourceType": "Patient",
        "meta": {
          "profile": [
            "https://www.medizininformatik-initiative.de/fhir/core/modul-person/"
          ]
        },
        "identifier": [
          {
            "use": "usual",
            "type": {
              "coding": [
                {
                  "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
                  "code": "MR"
                }
              ]
            }
          }
        ],
        "system": "http://mimic.mit.edu/fhir/mimic/identifier/patient",
        "value": "10038992"
      }
    },
    {
      "name": [
        {
          "use": "official",
          "family": "Patient_10038992"
        }
      ]
    }
  ]
}

```

Mirth-Connect Patient Source Transformer

[https://www.medizininformatik-initiative.de/Kerndatensatz/Modul\\_Person\\_Version\\_2/MIIGModulPerson-](https://www.medizininformatik-initiative.de/Kerndatensatz/Modul_Person_Version_2/MIIGModulPerson-)

TechnischImplementierung-FHIR-Profil-PatientInPatient.html

# TRANSFORMATION VON CSV NACH FHIR MIT MIRTH-CONNECT

## FHIR-TEMPLATE: MII-FALL

Encounter	I	Encounter
id	S	0..1 string
meta	S	0..1 Meta
extension	I	0..* Extension
identifier	S	0..* Identifier
status	S ?!	1..1 code Binding
statusHistory		0..* BackboneElement
class	S	0..1 Coding Binding
classHistory		0..* BackboneElement
type	S	0..* CodeableConcept
serviceType	S	0..1 CodeableConcept
priority		0..1 CodeableConcept
subject	S ? I	1..1 Reference(Patient   Group)
episodeOfCare	S ? I	0..* Reference(EpisodeOfCare)
basedOn	I	0..* Reference(ServiceRequest)
participant	S	0..* BackboneElement
appointment	S ? I	0..* Reference(Appointment)
period	S	1..1 Period
length	I	0..1 Duration
reasonCode	S	0..* CodeableConcept Binding
reasonReference	S ? I	0..* Reference(Condition   Procedure   Observation...)
diagnosis	S	0..* BackboneElement
account	I	0..* Reference(Account)
hospitalization	S	0..1 BackboneElement
location	S	0..* BackboneElement
serviceProvider	S ? I	0..1 Reference(Organization)
partOf	S ? I	0..1 Reference(Encounter)



Outbound Message Template

Data Type: JSON Properties

```

{
  "resourceType": "Bundle",
  "type": "transaction",
  "entry": [
    {
      "resource": {
        "resourceType": "Encounter",
        "meta": {
          "profile": [
            "https://www.medizininformatik-initiative.de/fhir/core/modul-fall/Stru"
          ]
        },
        "status": "finished",
        "identifier": [
          {
            "type": {
              "coding": [
                {
                  "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
                  "code": "VN"
                }
              ]
            },
            "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
            "code": "VN"
          }
        ],
        "system": "http://terminology.hl7.org/CodeSystem/v2-0203",
        "value": "25135483"
      },
      "class": {
        "code": "AMB",
        "display": "ambulatory",
        "system": "http://hl7.org/fhir/v3/ActCode/cs.html"
      },
      "subject": {
        "reference": "Patient/10004235"
      }
    }
  ]
}
    
```

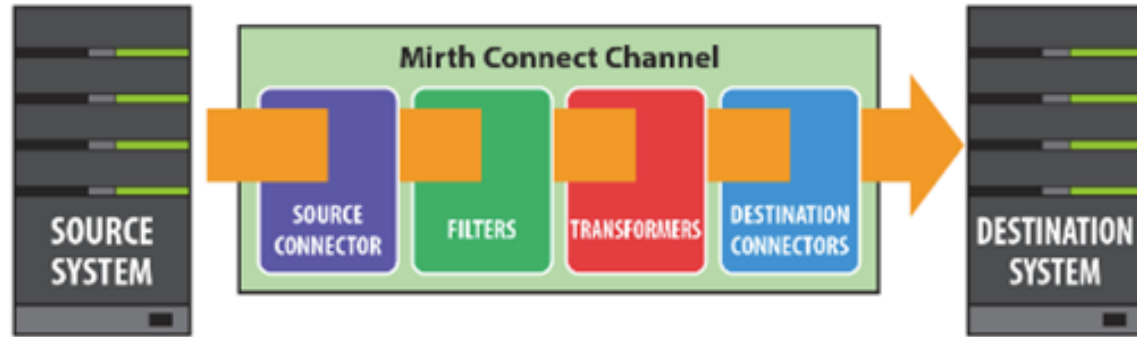
Codesystem: [https://www.medizininformatik-initiative.de/Kerndatensatz/Modul\\_Fall\\_Version\\_2/MIIGModulFall-TechnischeImplementierung-FHIRProfile-EncounterKontaktGesundheitseinrichtung.html](https://www.medizininformatik-initiative.de/Kerndatensatz/Modul_Fall_Version_2/MIIGModulFall-TechnischeImplementierung-FHIRProfile-EncounterKontaktGesundheitseinrichtung.html)

[https://www.medizininformatik-initiative.de/Kerndatensatz/Modul\\_Fall\\_Version\\_2/MIIGModulFall-](https://www.medizininformatik-initiative.de/Kerndatensatz/Modul_Fall_Version_2/MIIGModulFall-)



# MIRTH-CONNECT

## CHANNEL AUFBAU



[https://www.adldata.org/download/Manuals/Mirth\\_Connect\\_3.4\\_Users\\_Guide.pdf](https://www.adldata.org/download/Manuals/Mirth_Connect_3.4_Users_Guide.pdf)

### 1. Source (Eingang)

- Lesen des CSV-Datensatz
- File Reader (definierter Ordner)

### 2. Filters & Transformer (Verarbeitung)

- Transformation Eingangsdaten CSV nach FHIR-Template
- Mapping von CSV-Spalten auf FHIR-Attribute

### 3. Destination (Ausgang)

- Senden der transformierten FHIR-JSON Nachrichten an FHIR-Server
- Http-Sender

# MIRTH-CONNECT

## 2. SOURCE TRANSFORMER

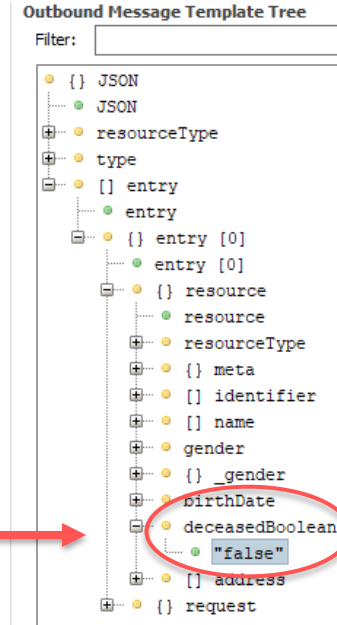
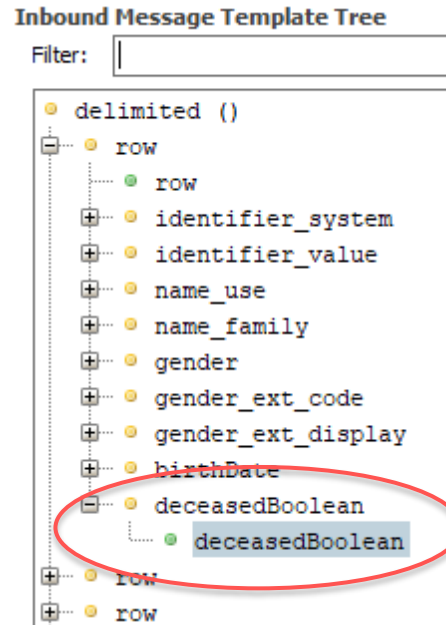
- Transformation des Datensatz nach FHIR
- Mapping von Spalten der CSV auf FHIR-Attribute
- Transformer Step: zu mappendes Attribut
  
- **Input:** CSV-Datei
- **Output:** FHIR Message Template

# TRANSFORMATION VON CSV NACH FHIR MIT MIRTH-CONNECT

## TRANSFORMATION MII-PATIENT: BEISPIEL

### Input: CSV-Datei

- Alle Reihen und Spalten als Baumstruktur
- Mapping der ersten Reihe ausreichend



### Output: FHIR JSON Template

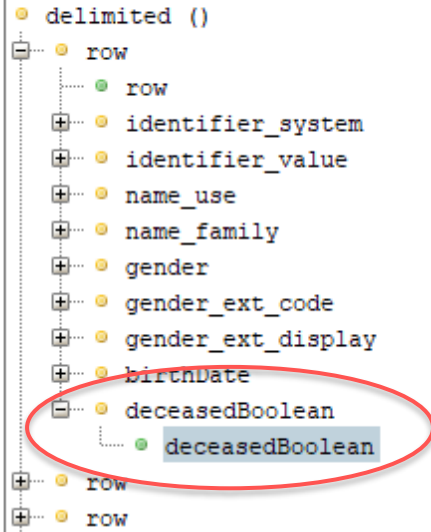
- ein JSON Tree

# TRANSFORMATION VON CSV NACH FHIR MIT MIRTH-CONNECT

## TRANSFORMATION MII-PATIENT: BEISPIEL

### Inbound Message Template Tree

Filter:



- Mirth-Connect Transformer
- **Input CSV:**

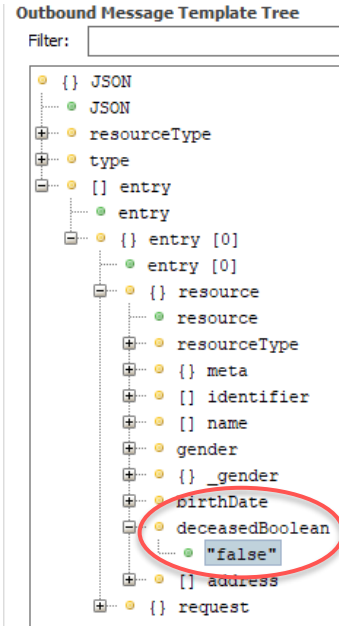
Mirth-Connect Syntax

➤ `msg['row'][0]['deceasedBoolean'].toString()`

Zielvariable

# TRANSFORMATION VON CSV NACH FHIR MIT MIRTH-CONNECT

## TRANSFORMATION MII-PATIENT: BEISPIEL



- Mirth-Connect Transformer
- **Output** FHIR Message Template:

Mirth-Connect Syntax

➤ `tmp['entry'][0]['resource']['deceasedBoolean']`

Zielvariable

# TRANSFORMATION VON CSV NACH FHIR MIT MIRTH-CONNECT

## TRANSFORMATION MII-PATIENT: BEISPIEL

**Edit Channel - MII Patient from MIMIC CSV - Source Transformer**

Enabled	#	Name	Type
<input checked="" type="checkbox"/>	0	entry [0] - resource - identifier [0] - system (out) <-- row [1] - identifier_system (in)	Message Builder
<input checked="" type="checkbox"/>	1	entry [0] - resource - identifier [0] - value (out) <-- row [1] - identifier_value (in)	Message Builder
<input checked="" type="checkbox"/>	2	entry [0] - resource - name [0] - use (out) <-- row [1] - name_use (in)	Message Builder
<input checked="" type="checkbox"/>	3	entry [0] - resource - name [0] - family (out) <-- row [1] - name_family (in)	Message Builder
<input checked="" type="checkbox"/>	4	entry [0] - resource - gender (out) <-- row [1] - gender (in)	Message Builder
<input checked="" type="checkbox"/>	5	entry [0] - resource - _gender - extension [0] - valueCoding - code (out) <-- row [1] - gender_ext_code (in)	Message Builder
<input checked="" type="checkbox"/>	6	entry [0] - resource - _gender - extension [0] - valueCoding - display (out) <-- row [1] - gender_ext_display (in)	Message Builder
<input checked="" type="checkbox"/>	7	entry [0] - resource - birthDate (out) <-- row [1] - birthDate (in)	Message Builder
<input checked="" type="checkbox"/>	8	entry [0] - resource - deceasedBoolean (out) <-- row [1] - deceasedBoolean (in)	Message Builder

Step \ Generated Script

Message Segment: tmp['entry'][0]['resource']['deceasedBoolean']

Mapping: msg['row'][0]['deceasedBoolean'].toString()

Default Value:

## AUFGABEN (SIEHE ÜBUNGSBLATT)

### 1. Mirth-Connect

1.1 - 1.2

Channel erstellen

1.3.1

Source Connector

1.3.2

Destination Connector

1.4 - 1.5

Transformer erweitern

1.6

Deployen der Channels und senden der FHIR

Nachrichten

### 1. ClinFHIR

2.1

Transformierte Patienten und Fälle ansehen

2.2 - 2.3

Fragen beantworten



UNIVERSITÄT  
LEIPZIG

Medizinische Fakultät

# VIELEN DANK!

**Mona Perbix**

Institut für Medizinische Informatik, Statistik und Epidemiologie (IMISE)