



UNIKLINIK  
KÖLN

# Breaking Barriers for Interoperability: A Reference Implementation of CSV-FHIR Transformation using open-source tools

Samer Alkarkoukly, Md. Mostafa Kamal, Oya Beyan:  
University of Cologne, Germany

25.05.2023 | Samer Alkarkoukly, MD. MSc. | University Hospital Cologne, Germany



# Objective:

To empower healthcare providers by proposing a user-friendly method to transform medical data from CSV format into FHIR resources (without requiring sophisticated programming scripts or tools)

# I. Introduction

# FHIR



# FHIR

Fast Health  
Interoperability  
Resources

## Level 1 Basic framework on which the specification is built



### Foundation

Base Documentation, XML, JSON, RDF, Datatypes, Extensions

## Level 2 Supporting implementation and binding to external specifications



### Implementer Support

Downloads,  
Version Mgmt,  
Use Cases,  
Testing



### Security & Privacy

Security,  
Consent,  
Provenance,  
AuditEvent



### Conformance

StructureDefinition,  
CapabilityStatement,  
ImplementationGuide,  
Profiling



### Terminology

CodeSystem,  
ValueSet,  
ConceptMap,  
Terminology Svc



### Exchange

REST API + Search  
Documents  
Messaging  
Services  
Databases  
Subscriptions

## Level 3 Linking to real-world concepts in the healthcare system



### Administration

Patient, Practitioner, CareTeam, Device, Organization, Location, Healthcare Service

## Level 4 Record-keeping and Data Exchange for the healthcare process



### Clinical

Allergy, Problem,  
Procedure,  
CarePlan/Goal,  
Family History,  
RiskAssessment,  
etc.



### Diagnostics

Observation,  
Report, Specimen,  
ImagingStudy,  
Genomics, etc.



### Medications

Medication,  
Request, Dispense,  
Administration,  
Statement,  
Immunization, etc.



### Workflow

Introduction +  
Task, Appointment,  
Schedule, Referral,  
PlanDefinition, etc.



### Financial

Claim, Account,  
Invoice, ChargeItem,  
Coverage + Eligibility  
Request & Response,  
ExplanationOfBenefit,  
etc.

## Level 5 Providing the ability to reason about the healthcare process



### Clinical Reasoning

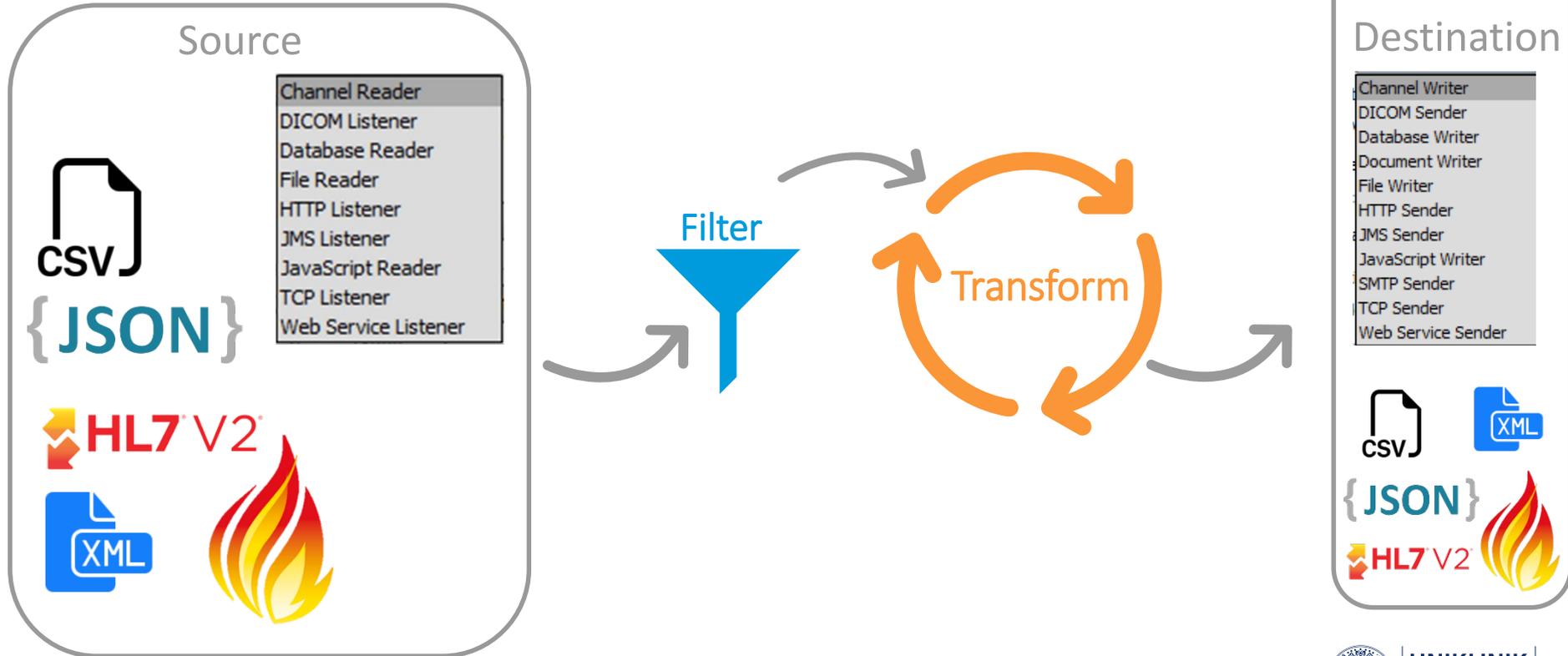
Library, PlanDefinition & GuidanceResponse,  
Measure/MeasureReport, etc.



### Medication Definition

Medicinal, Packaged & Administrable product definitions,  
Regulated Authorization, etc.

# Mirth Connect

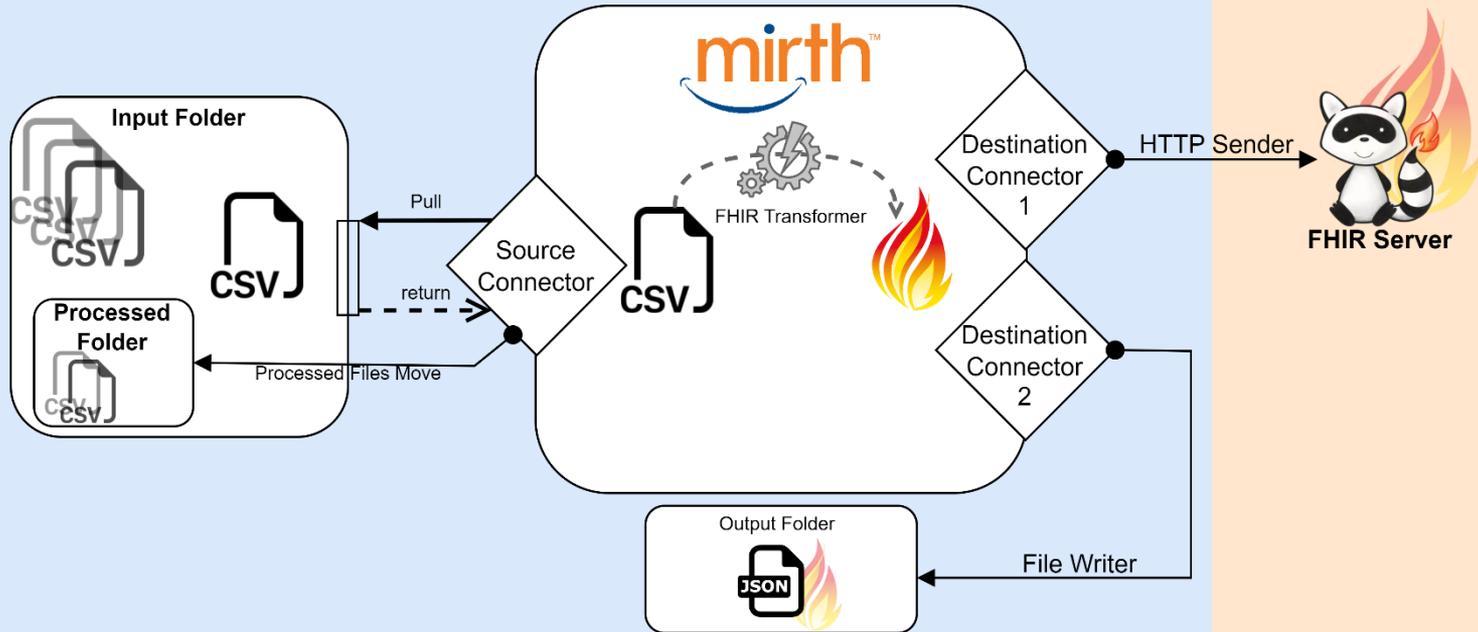




UNIKLINIK  
KÖLN

## II. Methods:

# Reference Model:



# Installation & Configuration:

Mirth Connect (MC) could be downloaded and installed from GitHub or from the vendor. The installation includes four components MC Server, MC Server Manager, MC Command Line Interface, and MC Administrator Launcher, and it will obtain two ports from the local PC. After installation, all the following work can be done using the graphical user interface of Mirth Connect by running the server manager first, starting MC service, and then launching MC administrator. A channel needs to be designed and configured in MC Administrator; the configuration settings are provided in Table 1.

Connector	Data Type	Connector Type	Configuration
Source	Delimited Text	File Reader	Directory: Local PC input folder Filename Filter Pattern: "*.csv"
Destination 1	JSON	HTTP Sender	URL: FHIR server endpoint Content-type: "application/JSON" Content: "\${JsonUtil.prettyPrint(\$ {message.encodedData})}"
Destination 2	JSON	File Writer	Directory: Local PC output folder File Name: "\${originalFilename}\${COUNT}.json" Template: Same value of "content" of destination 1

The source connector has a transformer which transforms the CSV inbound message from the source into JSON outbound messages as FHIR resources in a transaction bundle for both of the planned destinations. The inbound message template is a predefined CSV file reflecting the FHIR Patient resource elements and all the incoming messages should use the same template.



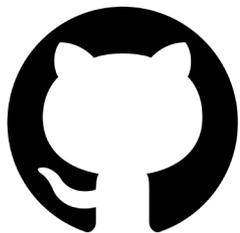
UNIKLINIK  
KÖLN

## III. Results:

# Outputs:

The suggested reference implementation was **tested successfully** by exporting FHIR resources to a locally installed HAPI server and to a local PC directory, and additional quality and performance tests were implemented to check the applicability on large-scale implementations and also the validity and correctness of the generated FHIR resources.

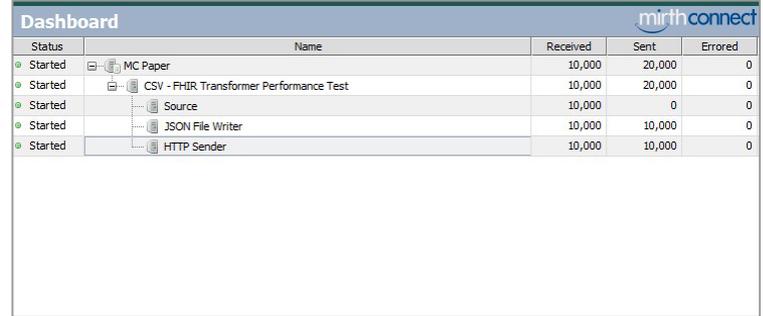
During the implementation, **none of the programming languages was needed** neither in installation (done with the graphical user interface) nor in the channel set-up (done by drag and drop between the inbound and outbound elements).



All the used files, including **the exported channel, CSV and JSON templates, and the testing CSV files, are available publicly on GitHub** <https://github.com/alkarkoukly/CSV-FHIR-Transformer>

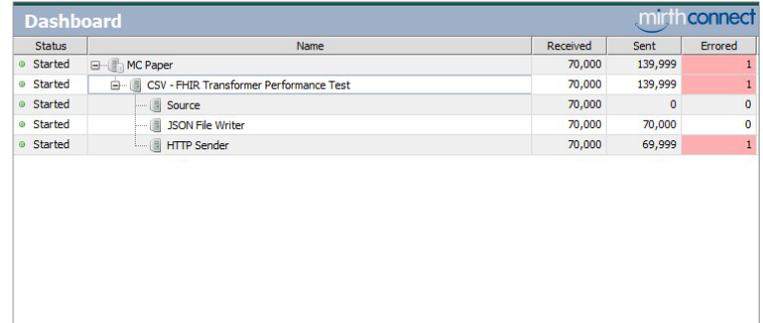
# Testing:

The **performance test** included datasets in CSV format from the input folder in 3 scenarios (10 files of 1,000, 10 files of 10,000, 1 file of 50,000 records); the processing speed average was **59 milliseconds per CSV record**, and it is correlated with records amount. The **quality test** included monitoring the errors received from the channel logs, which was limited to **only one timeout error** during processing the largest CSV file. Additionally, a **validation test** of the generated FHIR resources was implemented by validating a random sample of 1% of the processed records (700 resources) using the “\$validate” function of the destination server. All the tested resources were valid with no warnings or errors, and matched the original data.



Dashboard

Status	Name	Received	Sent	Errored
Started	MC Paper	10,000	20,000	0
Started	CSV - FHIR Transformer Performance Test	10,000	20,000	0
Started	Source	10,000	0	0
Started	JSON File Writer	10,000	10,000	0
Started	HTTP Sender	10,000	10,000	0



Dashboard

Status	Name	Received	Sent	Errored
Started	MC Paper	70,000	139,999	1
Started	CSV - FHIR Transformer Performance Test	70,000	139,999	1
Started	Source	70,000	0	0
Started	JSON File Writer	70,000	70,000	0
Started	HTTP Sender	70,000	69,999	1



UNIKLINIK  
KÖLN

## IV. Discussion:

# The Benefits:

Using Mirth Connect as an open-source tool showed full success in transforming patient data from CSV format into FHIR Patient resources represented in transaction bundles. This approach is suitable for **all types of healthcare facilities** planning to generate FHIR resources from their primary information systems using basic technical skills and IT infrastructure, without the need for advanced hardware or virtual machines, and without any external services built with Java, C, or Python. The approach is applicable on different types of data sources (regular export in CSV, SQL queries, live streams of files or messages, and services with REST APIs).

# Opportunities & Challenges:

The reference implementation targeted Patient resource from FHIR specifications, and it is **applicable on all of the other types of FHIR resources**. It is also possible to include **data cleaning, files filtering, data values mapping to terminologies** or value sets, and variable format transformations in the channel of Mirth Connect. This could be done using the JavaScript programming environment contained in Mirth Connect user interface. Although the input CSV structure is static, it can be **configured with the user interface** to meet the different needs.

The limitation of this approach is mainly related to the **communication with the destination FHIR** server and interactively exchange data in both ways during the submission of each single resource. For example, the need for checking if a specific resource is already available in the FHIR server and then to conditionally create or update the intended resource based on specific elements. This situation is still theoretically possible with Mirth Connect by processing the response from the server.

The approach followed in this work is **reproducible, repeatable and improvable** by using the exported channel, templates and testing files provided in the project repository in GitHub.

# References:

- [1] Kubben P, Dumontier M, Dekker A. Fundamentals of clinical data science. Fundamentals of Clinical Data Science. Springer International Publishing; 2018;1–219.
- [2] Nayak A, Božić B, Longo L. Data Quality Assessment of Comma Separated Values Using Linked Data Approach. 2022. p. 240–50.
- [3] Overview - FHIR v4.3.0 [Internet]. [cited 2022 Aug 23]. Available from: <https://www.hl7.org/fhir/overview.html>
- [4] Martínez-García A, Cangioli G, Chronaki C, Löbe M, Beyan O, Juehne A, et al. FAIRness for FHIR: Towards Making Health Datasets FAIR Using HL7 FHIR. Stud Health Technol Inform [Internet]. Stud Health Technol Inform; 2022 [cited 2022 Aug 23];290. Available from: <https://pubmed.ncbi.nlm.nih.gov/35672963/>
- [5] Healthcare Integration Engine | Mirth® Connect by NextGen Healthcare [Internet]. [cited 2022 Nov 28]. Available from: <https://www.nextgen.com/products-and-services/integration-engine>
- [6] Deppenwiese N, Delpy P, Lambarki M, Lablans M. ADT2FHIR - A Tool for Converting ADT/GEKID Oncology Data to HL7 FHIR Resources. Stud Health Technol Inform [Internet]. Stud Health Technol Inform; 2021 [cited 2022 Aug 29];283:104–10. Available from: <https://pubmed.ncbi.nlm.nih.gov/34545825/>
- [7] Fette G, Ertl M, Störk S. Translating openEHR Models to FHIR. Stud Health Technol Inform [Internet]. Stud Health Technol Inform; 2020 [cited 2022 Aug 29];270:1415–6. Available from: <https://pubmed.ncbi.nlm.nih.gov/32570686/>
- [8] Pfaff ER, Champion J, Bradford RL, Clark M, Xu H, Fecho K, et al. Fast Healthcare Interoperability Resources (FHIR) as a Meta Model to Integrate Common Data Models: Development of a Tool and Quantitative Validation Study. JMIR Med Inform [Internet]. JMIR Med Inform; 2019 [cited 2022 Aug 29];7. Available from: <https://pubmed.ncbi.nlm.nih.gov/31621639/>



# Take-Home:

Open-source, user-friendly tools transform CSV into FHIR resources efficiently, easily, with no-code approach.



Mhd Samer Alkarkoukly  
Medical Data Scientist | FHIR & openEHR  
Enthusiast | Humanitarian Aid Worker

