



**CoJaC** ist eine Toolchain zur Konvertierung von historischen COBOL-Applikationen in moderne Java-Enterprise-Lösungen.

**CoJaC** generiert wartbaren und performanten Java-Code, welcher semantisch äquivalent zum COBOL-Code ist.

**CoJaC** erreicht bei der Konvertierung einen Automatisierungsgrad von bis zu 90 %.

**CoJaC** verbindet wissenschaftliches Know-how im Compilerbau mit Kernkompetenz aus erfolgreichen Migrationsprojekten.

**CoJaC** lässt sich an individuelle Kundenwünsche anpassen.

## Migrationspfade Basis- und Zielsystem

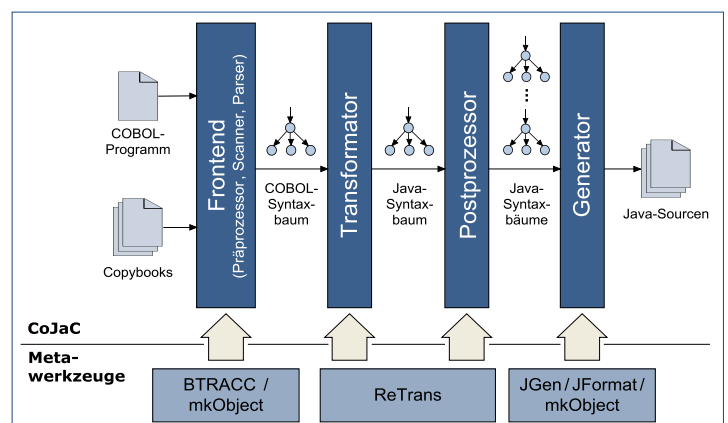
✓ Es wird keine 1:1-Programmtransformation realisiert, sondern es findet eine Architekturmigration statt:

- Online-Programme (Server) werden als Webservices im Java-Zielsystem bereitgestellt.
- COBOL-Batch-Programme werden in autonome Java-Programme transformiert (1:1-Migration).
- COBOL-Datenfiles werden in Datenbanktabellen migriert.
- In den Quelltext eingebettete, statische SQL-Anweisungen werden in dynamische SQL-Anweisungen konvertiert.

Migrationspfade	
Basissystem	Zielsystem
COBOL (COBOL85, IBM, MFC, ...)	Java
Middleware (TUXEDO, Pathway, CICS, ...)	Java-Webservices
Datenfiles	Relationale Datenbank
Embedded SQL (Oracle, DB2, ...)	Dynamisches SQL

## Tool-Architektur

- ✓ CoJaC basiert auf der Architektur eines Compilers, welche von pro et con für die technischen Belange von Transformationswerkzeugen weiterentwickelt wurde.
- ✓ Bei der Entwicklung von CoJaC wurden firmeneigene Metawerkzeuge verwendet. Dies garantiert eine extrem niedrige Fehlerquote im Gegensatz zu manueller Entwicklung und eine einfache Anpassung von CoJaC an individuelle Kundenwünsche.



# Ausgewählte Aspekte der Migration

## ✓ *Java-Laufzeitsystem*

Das Laufzeitsystem stellt in der Zielumgebung die Funktionalität für die migrierten Programme bereit, wie z. B.:

- ▶ Klassen für COBOL-Datentypen, Basisklassen für Arrays und Strukturen,
- ▶ Methoden für COBOL-Anweisungen, welche nicht mit nativen Java-Anweisungen realisiert werden können (z. B. `STRING` und `INSPECT`),
- ▶ Methoden zur Typkonvertierung, statische Methoden zur Abbildung der Intrinsic Functions,
- ▶ Klassen zum Bereitstellen der Schnittstellenfunktionalität (Files, SQL, Middleware).

## ✓ *Middleware-Migration*

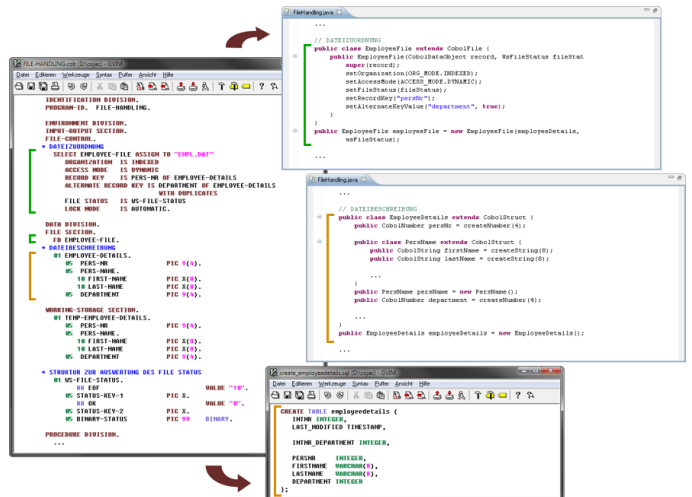
COBOL-Server, die unter der Steuerung einer Middleware (eines Transaktionsmonitors) arbeiten, werden optional in Java-Webservices migriert. Dabei werden Middleware-spezifische Daten und Anweisungen entfernt bzw. durch entsprechende Webservice-Funktionalität ersetzt. Die Schnittstelle ändert sich inhaltlich nicht. Die Webservices verarbeiten die gleichen Messages wie vorher die COBOL-Server. Notwendige Konvertierungen kapselt das Laufzeitsystem, welches auch die Verwaltung der Transaktionen übernimmt.

## ✓ *Migration von Files*

Die in COBOL verwendeten Datenfiles werden unter Berücksichtigung der verschiedenen Organisationsformen und Zugriffsmethoden in Datenbanktabellen migriert. Die Struktur der Tabellen leitet sich aus der über die `FD`-Struktur in der `FILE SECTION` zugeordneten Datensatzstruktur ab. Das Laufzeitsystem kapselt die Datenbankzugriffe und stellt Methoden zur File-Arbeit (`OPEN`, `READ`, `READ NEXT`, `CLOSE`, ...) bereit.

## ✓ *Weitere Features*

- ▶ Optionale Einbindung der Quelltextkommentare in den Zielcode
- ▶ Markierung der nicht konvertierbaren COBOL-Konstrukte durch aussagekräftige Kommentare im Zielcode
- ▶ Aufteilung des Zielcodes in Packages entsprechend Kundenanforderungen
- ▶ Nutzerspezifische Formatierung des generierten Java-Codes



## ✓ *Migration von embedded SQL*

In den COBOL-Quelltext eingebettete, statische SQL-Anweisungen werden in dynamische SQL-Anweisungen konvertiert. In das Laufzeitsystem wurden Methoden zur Verwaltung der Datenbankverbindung, zum Datenbankzugriff, zum Lesen und Schreiben von Daten in die Hostvariablen sowie zur Fehlerbehandlung integriert, welche die im Basissystem vorhandene Funktionalität abbilden.

# Technische Daten

- ✓ Aktuelle Version: CoJaC 1.2
- ✓ Genutzte Metawerkzeuge:
  - ▶ JGen (Java-Code-Generierung)
  - ▶ JFormat (Java-Code-Formatierung)
  - ▶ BTRACC (Backtracking Compiler Compiler, Parsergenerator)
  - ▶ ReTrans (Regelbasierte Transformationsmaschine)
  - ▶ mkObject (Modellierung von Syntaxbäumen)

# Kontakt

## pro et con

Innovative Informatikanwendungen GmbH  
Dittesstraße 15  
09126 Chemnitz

Telefon: +49 371 270951-12  
Telefax: +49 371 270951-29  
E-Mail: [info@proetcon.de](mailto:info@proetcon.de)  
Internet: [www.proetcon.de](http://www.proetcon.de)

Ansprechpartner: Frau Beier