

We make IT better!

Modellgetriebene Modernisierung: Was ist möglich?

b+m Informatik AG

Engineering Solutions

Holger Knoche

ObjektSpektrum Information Days

Frankfurt a. M., 21.11.2012



business IT management

ALLGEIER GROUP

Kurzvorstellung b+m

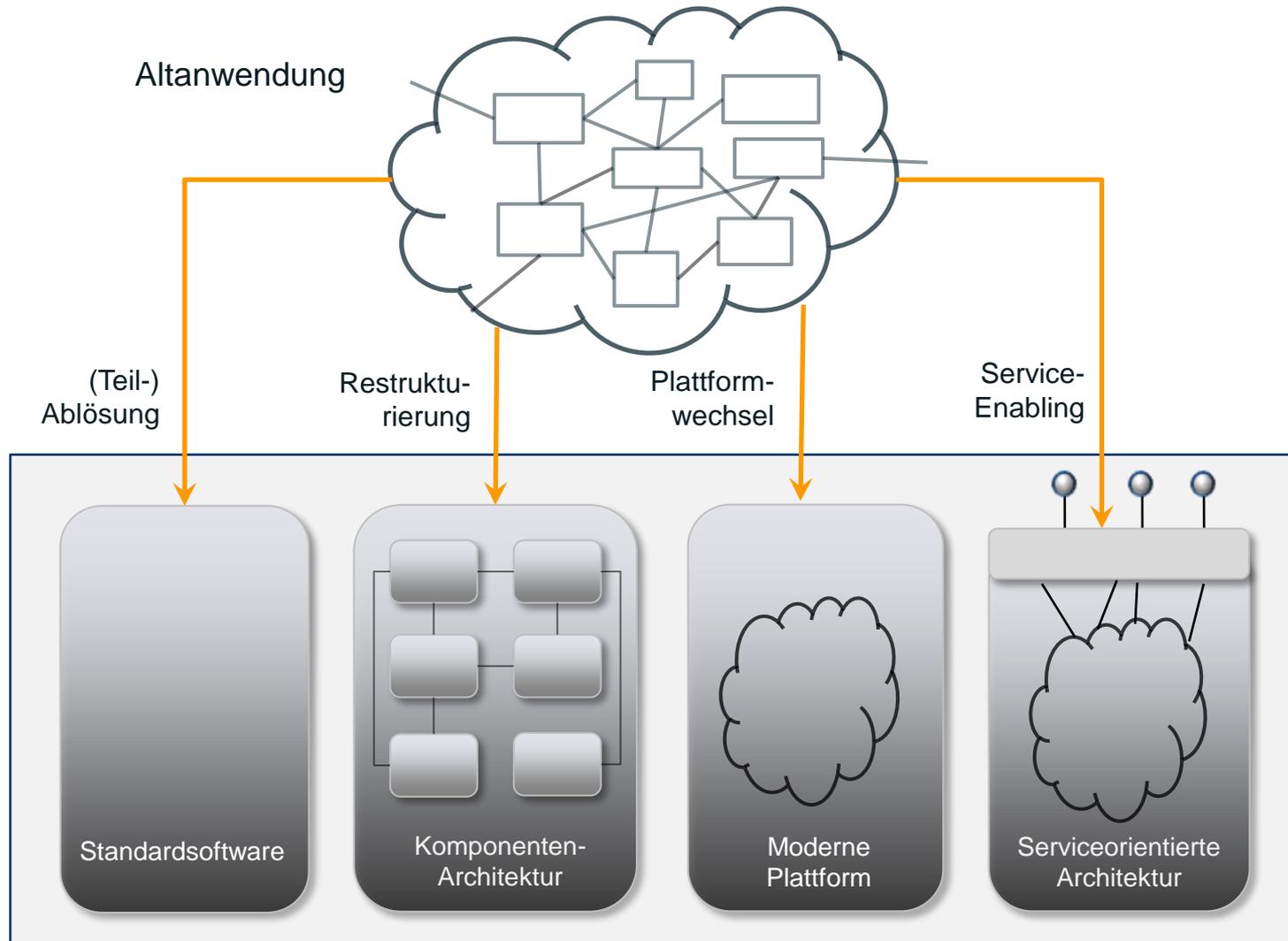
- Gegründet 1994 als becker & mohnberg Softwaregesellschaft mbH in Kiel
- Umfirmierung zur b+m Informatik AG im Jahr 2000
- Seit 2012 Mitglied der Allgeier-Gruppe
- Derzeit ca. 90 Mitarbeiter
- Hauptgeschäftsfelder: Software für Banken und Versicherungen sowie Software Engineering und -architektur
- Originäre Entwickler des Generator-Frameworks openArchitectureWare
- Über 10 Jahre Projekterfahrung im MDSD-Umfeld



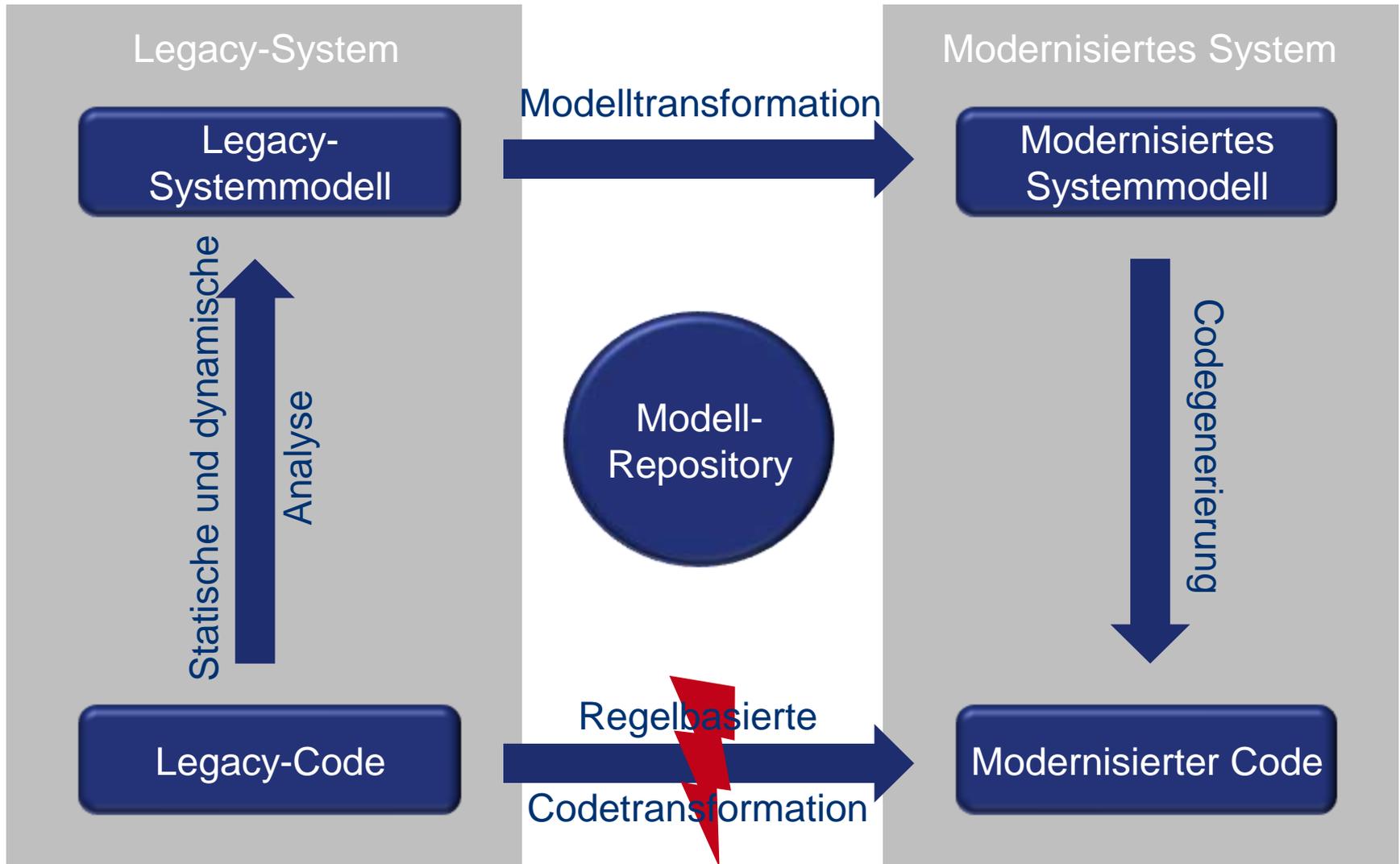
Typische Ausgangslage

- Ein bewährtes Softwaresystem ist noch nicht am Ende des fachlichen Lebenszyklus, aber die Wartungs-, Entwicklungs- und ggf. Betriebskosten gefährden die Rentabilität
- Die Requirements-Baseline stellt das Altsystem dar – aber wie kann man diese explizit machen?
- Oft ist das System Teil einer Landschaft, die insgesamt und strategisch modernisiert werden soll – damit kommt die EAM-Ebene ins Spiel
- Ziel ist eine nachhaltige Modernisierung, keine kurzfristige Rettung

Häufige Modernisierungsszenarien



Klassische vs. modellgetriebene Modernisierung



Kurzportrait Fallstudie 1: Komponentisierung eines Altsystems

Ausgangssituation

- Kernsoftwarekomponente in COBOL
- Strukturerosion über mehr als 20 Jahre und mehrere Entwicklergenerationen
- Aufnahme von fachfremder Funktionalität über die Lebenszeit

Modernisierungstreiber

- Hohe Wartungskosten aufgrund struktureller Komplexität
- Hoher Beistellungsaufwand wegen fachfremder Teile

Modernisierungsziel

- Auftrennung in fachlich klar getrennte Komponenten
- Neuverteilung fachlicher Zuständigkeiten

Beispielhaftes Systemmodell Fallstudie 1: Komponentisierung eines Altsystems

Fachlichkeit

Geschäfts-
system

Architektur

Komponente

Schnittstelle

Struktur / Code

Modul

Copybook

Altsystem: COBOL

Kurzportrait Fallstudie 2: Modernisierung einer Client-Server-Anwendung



Ausgangssituation

- Client-Server-Anwendung in Visual Basic 6
- Seit ca. 10 Jahren durchgehend vom selben Entwicklerteam betreut und gewartet

Modernisierungstreiber

- Ablauf des Herstellersupports für die Entwicklungs- und Laufzeitumgebung

Modernisierungsziel

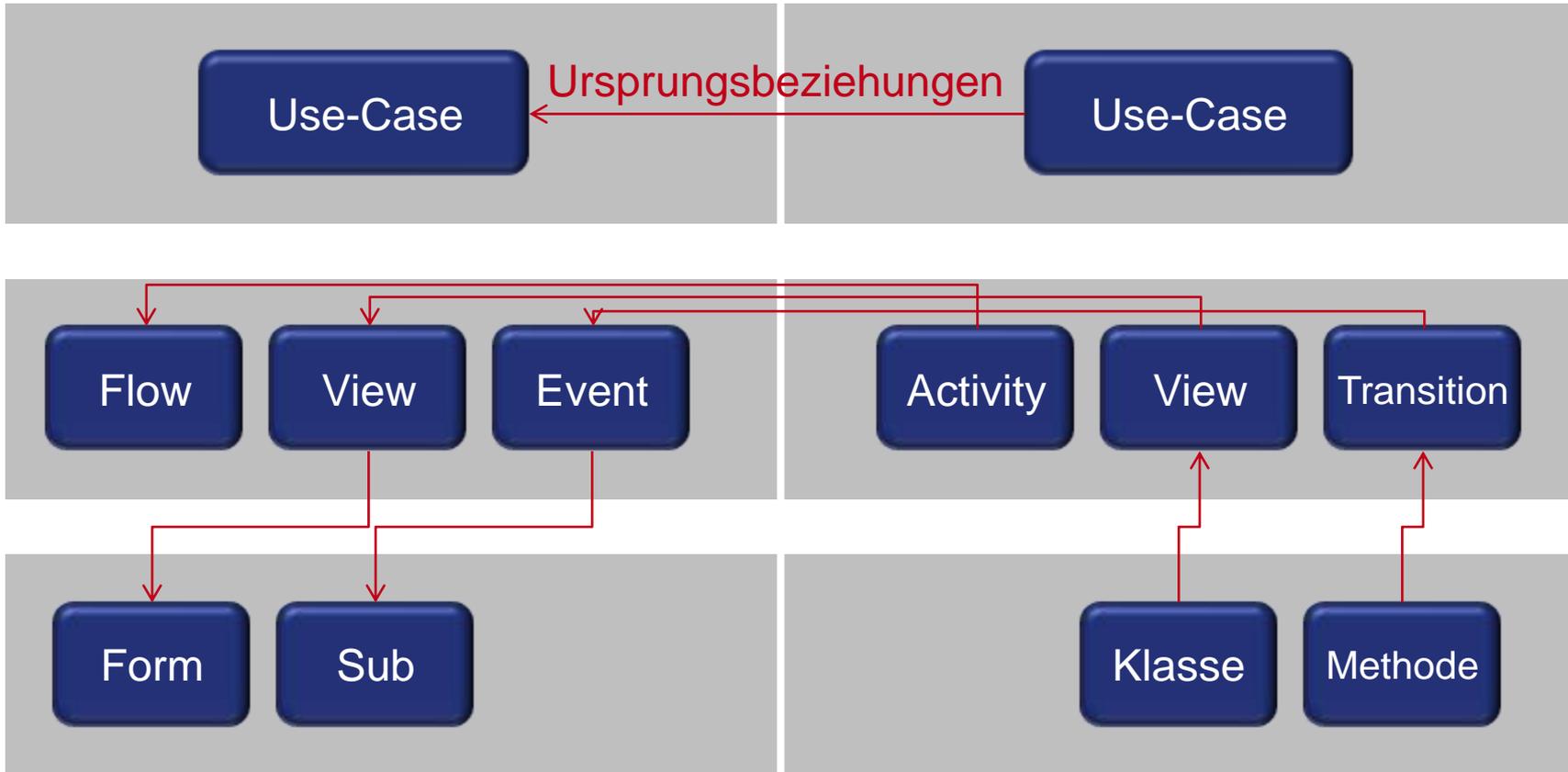
- Reimplementierung des bestehenden Systems auf einer neuen Plattform

Beispielhaftes Systemmodell 2: Übernahme von Maskenflüssen

Fachlichkeit

Architektur

Struktur / Code

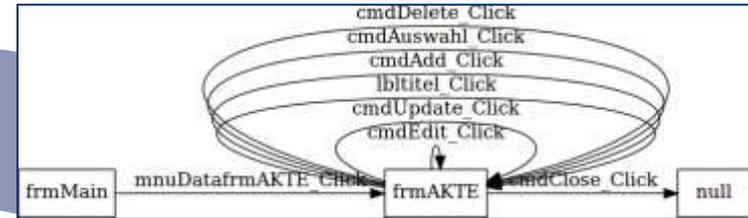
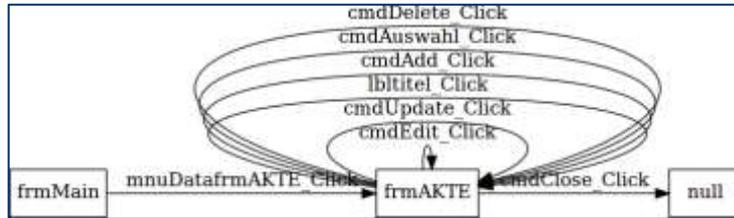


Altsystem: Visual Basic 6

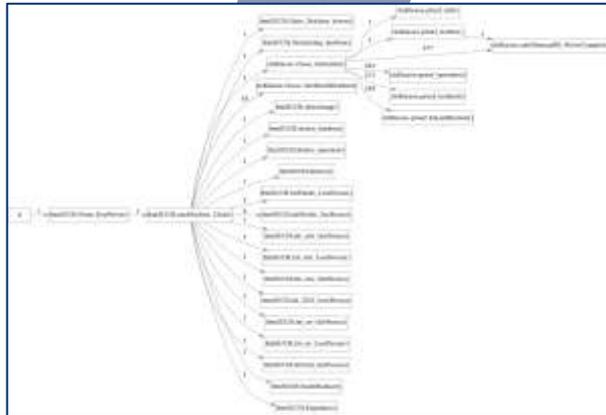
Modernisiertes System: Java-
basiertes Framework

Anwendungsbeispiele aus der Praxis: Transformation von Maskenfolgen

Architektur



Code / Struktur



```

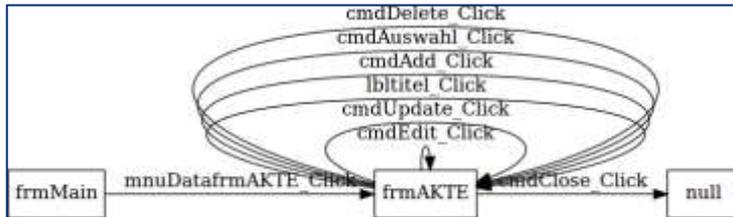
/**
 * Handles the UI event 'mnuFIND_Click' by notifying the flow controller
 * with the appropriate flow event {@code event_mnuFIND_Click}.
 */
@RepositoryObject(id=" LfNwsC8FEeKDHP4td4iCF0")
This element originates from the following elements:
- OperationImpl mnuFIND_Click

```

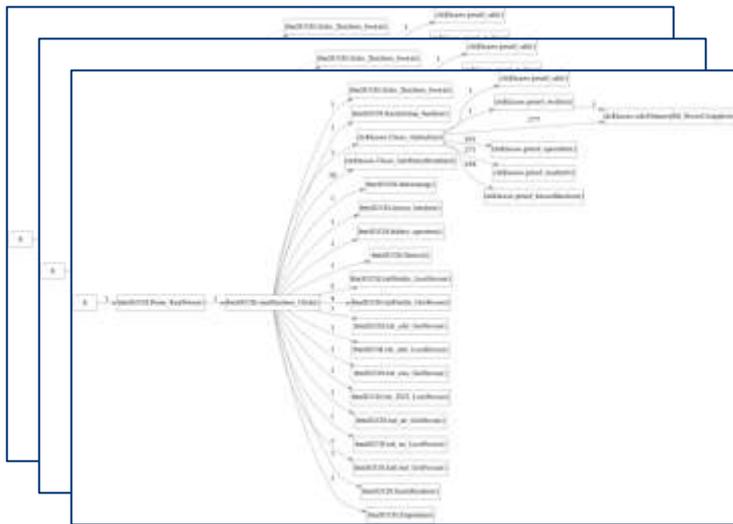
Altsystem

Neusystem

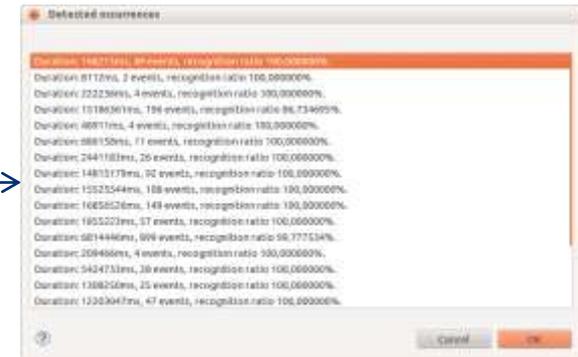
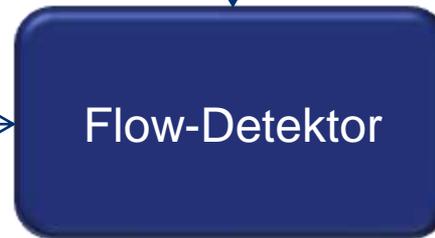
Anwendungsbeispiele aus der Praxis: Analyse von Nutzungsverhalten



Zu suchender Flow



Erhobene Traces



Gefundene Auftreten

Im Laufe der Modernisierung muss die Fortentwicklung der Altanwendung im modernisierten System nachvollzogen werden.

- Technisch einheitliche Modelle der Systeme erlauben die explizite Modellierung von Maßnahmen
- Durch den Zusammenhang der Modelle können auch Maßnahmen aufeinander bezogen werden
- Auf diese Weise wird die Steuerung und Überwachung der betreffenden Maßnahmen in beiden Systemen deutlich vereinfacht

Zusammenfassung

- Modellgetriebene Techniken können auch im Rahmen der Modernisierung gewinnbringend genutzt werden
- Die Erzeugung von Modellen der Altanwendung bietet bereits für die Planung einer Modernisierung einen großen Gewinn
- Die Hinzunahme von Laufzeitinformationen eröffnet viele Möglichkeiten, Wissen über die Altanwendung und deren Nutzung zu gewinnen
- Durch die strukturierte, einheitliche Ablage der Modelle beider Systeme in einem Repository werden übergreifende Steuerungs- und Analysebestrebungen ermöglicht bzw. deutlich vereinfacht

Ansprechpartner

Holger Knoche

Senior-Softwarearchitekt

Engineering & Solutions

holger.knoche@bmiag.de

F +49 (0)4340 404 1620

b+m Informatik AG

Rotenhofer Weg 20

24109 Melsdorf

www.bmiag.de