

Projekte. Beratung. Spezialisten.



Ist das Kunst oder kann das neu?
Mut zur Modernisierung, ein Praxisbeispiel

IKS Thementag

Autor: Oliver Heinemann

14.06.2023



Agenda

- › Kontext
- › Eingang & Verarbeitung
- › Ausgang
- › Benutzermanagement, Dokumente & Berichte
- › Fazit

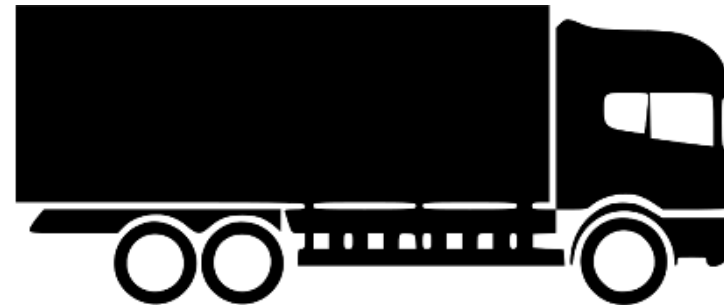
Agenda



- › **Kontext**
- › Eingang & Verarbeitung
- › Ausgang
- › Benutzermanagement, Dokumente & Berichte
- › Fazit

Kontext

- › Logistik
- › Sendungs- und Eventdaten
- › Verschiedene Transport-Management-Systeme (TMS) europäischer Länder
- › Sendungsverfolgung für Endkunden und Mitarbeiter
- › 50 Millionen Sendungen im Jahr
- › System älter als 20 Jahre



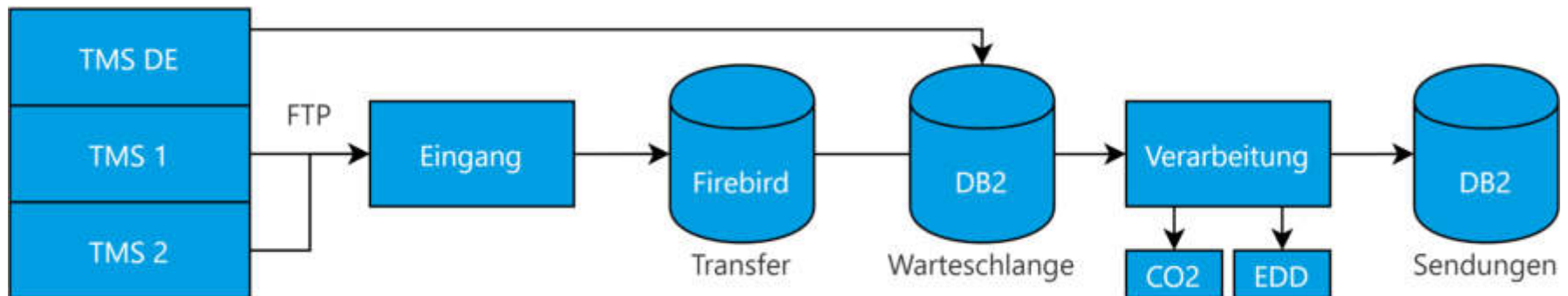
Agenda

- › Kontext
- › **Eingang & Verarbeitung**
- › Ausgang
- › Benutzermanagement, Dokumente & Berichte
- › Fazit

Eingang & Verarbeitung

› Ausgangssituation

- ◆ TMS
- ◆ FTP Server
- ◆ Eingangskomponente (TradeXpress)
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)
- ◆ Berechnung von CO2, EDD, ...

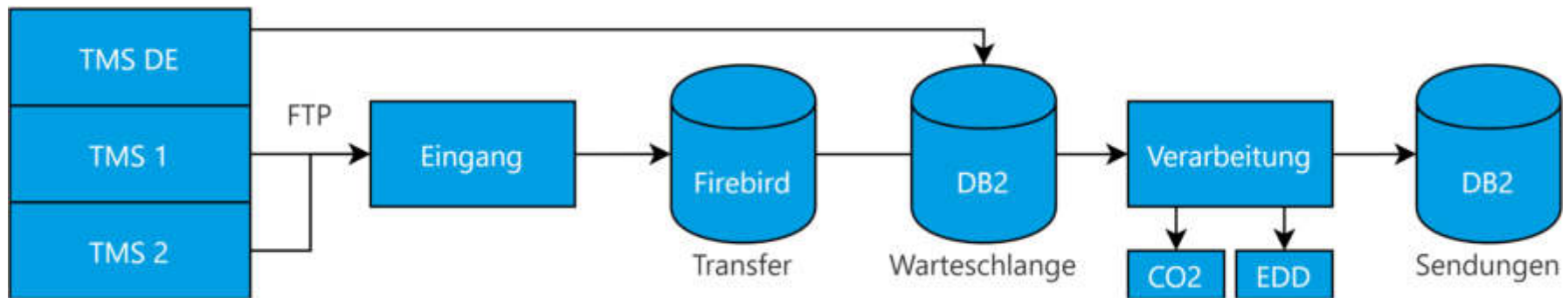


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Herausforderungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

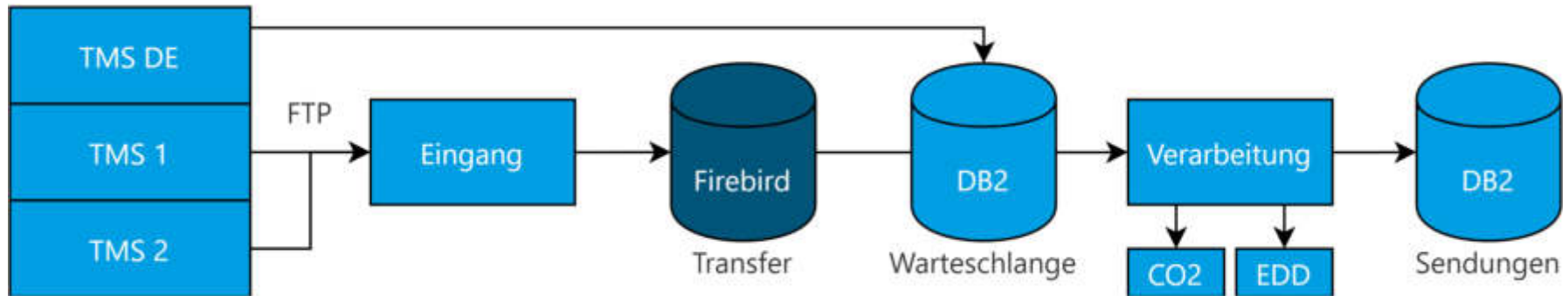


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Herausforderungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

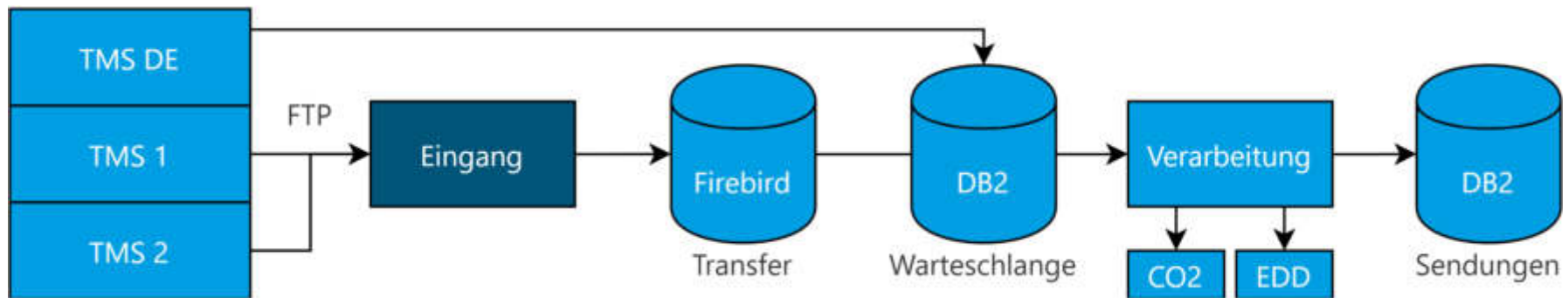


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Herausforderungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

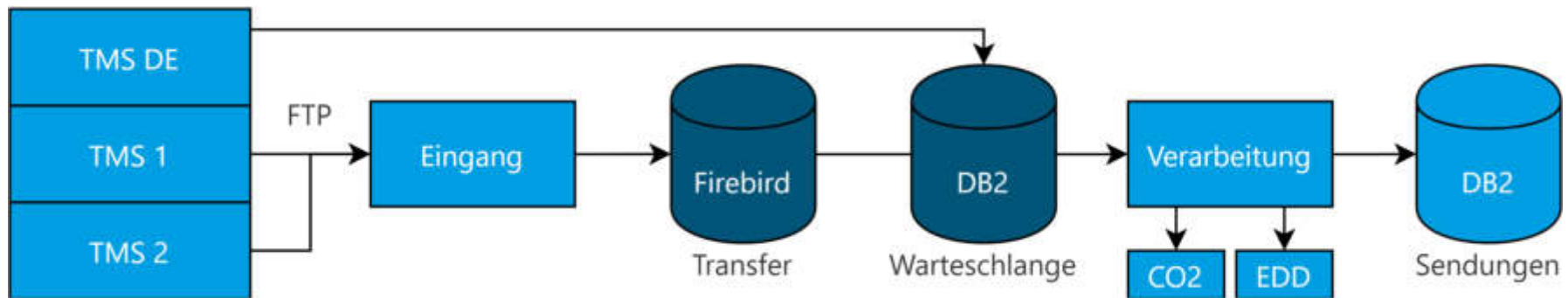


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Herausforderungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

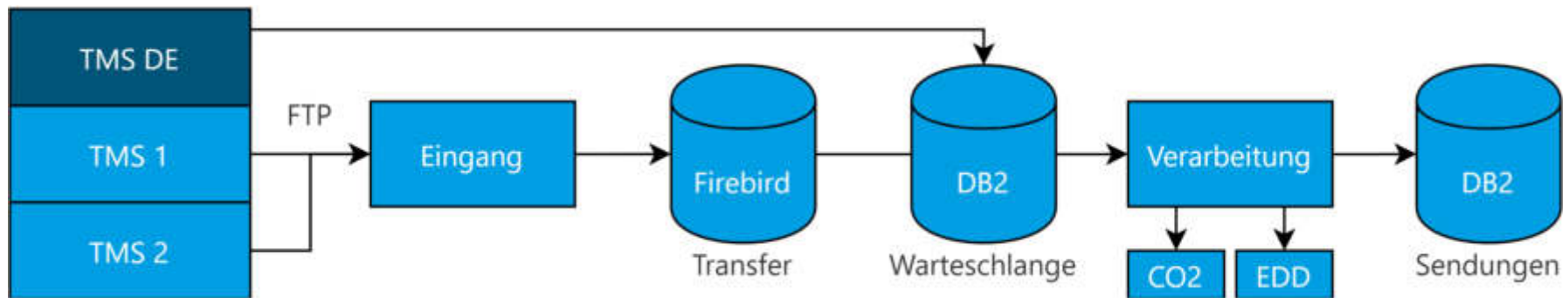


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Herausforderungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

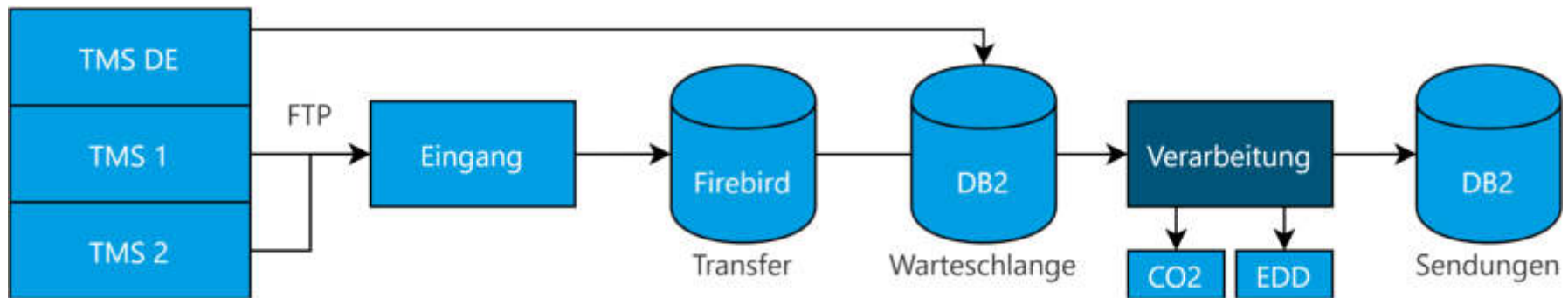


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Herausforderungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)



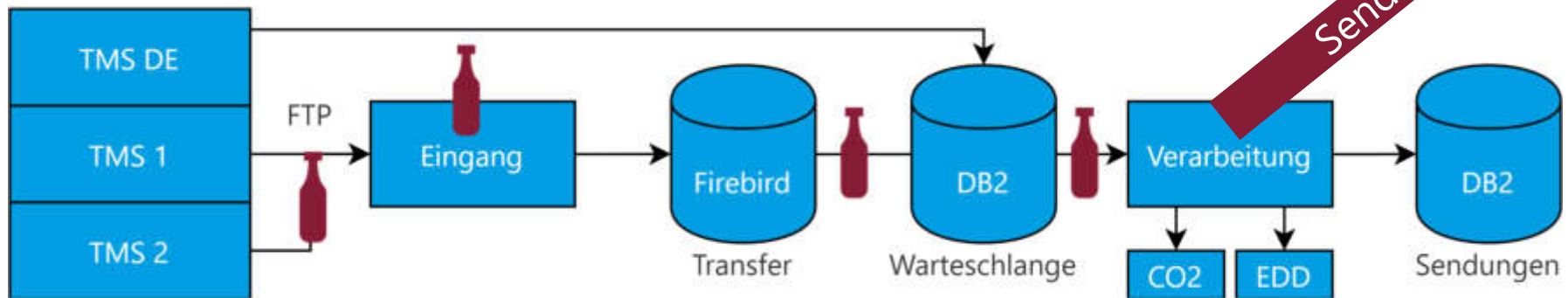
Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

> Herausforderungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

Wartung & Entwicklung
parallel

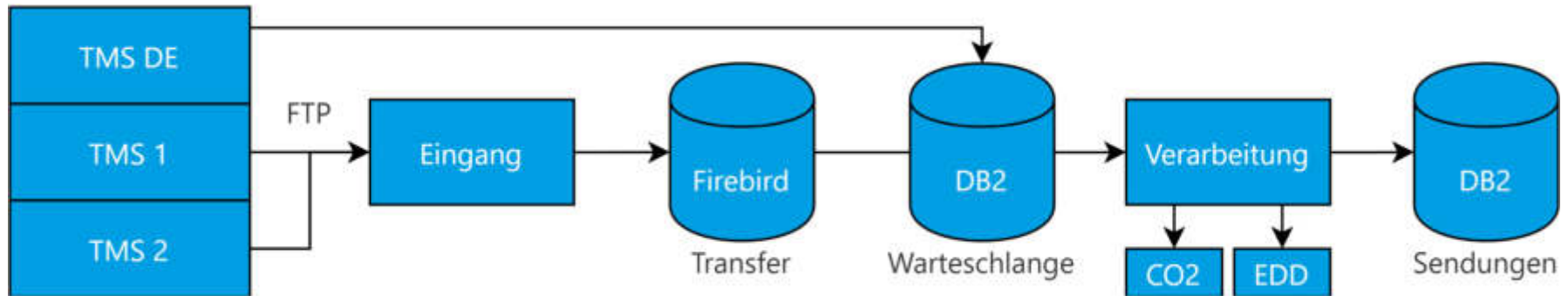


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

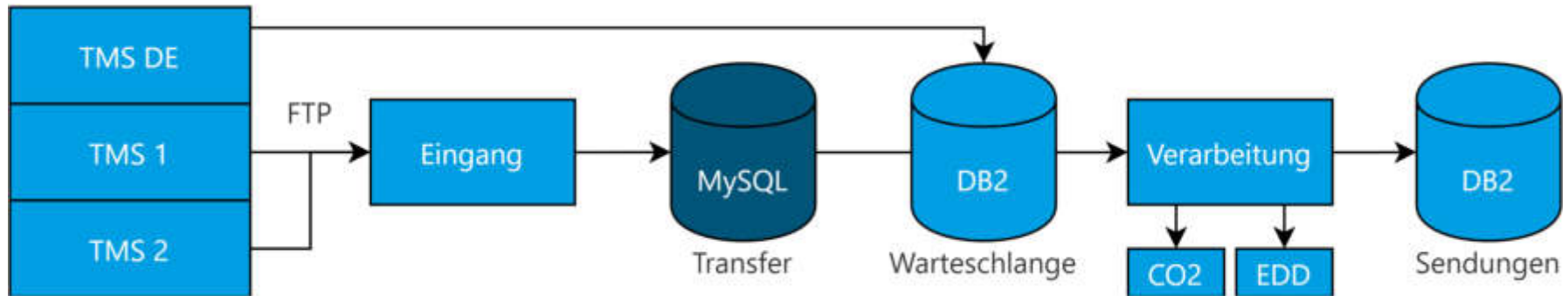


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

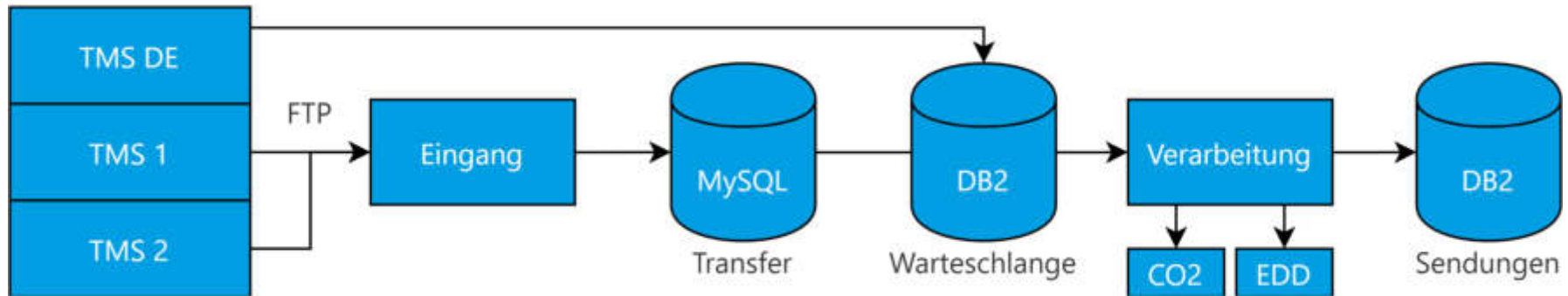


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress)
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

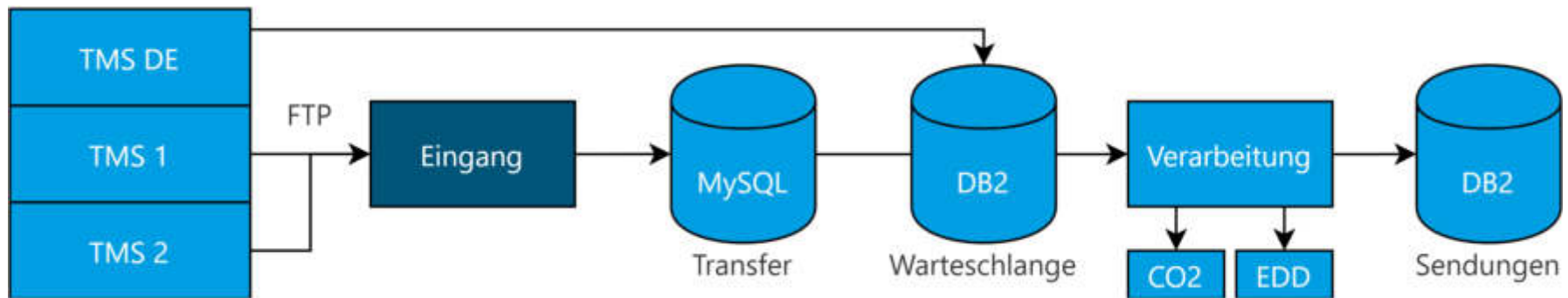


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress) -> Eigenentwicklung
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

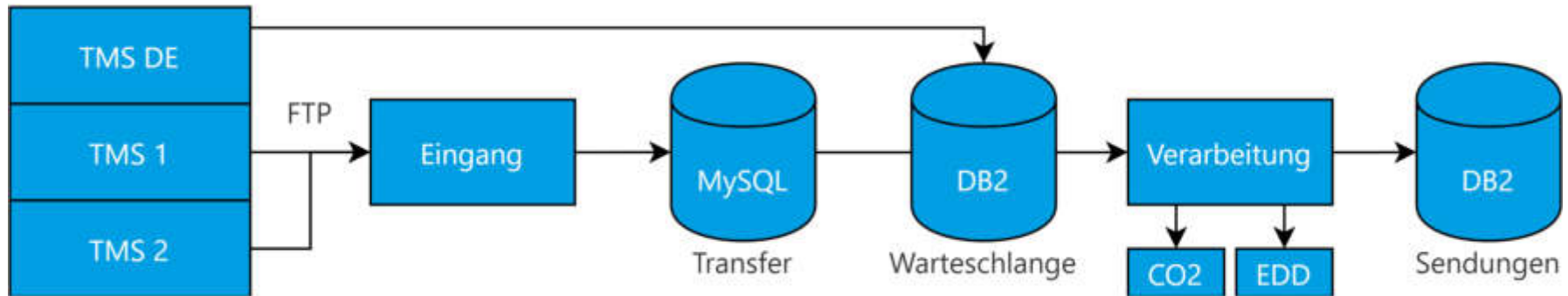


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress) -> Eigenentwicklung
- ◆ MySQL -> DB2 Datenbanklösung
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

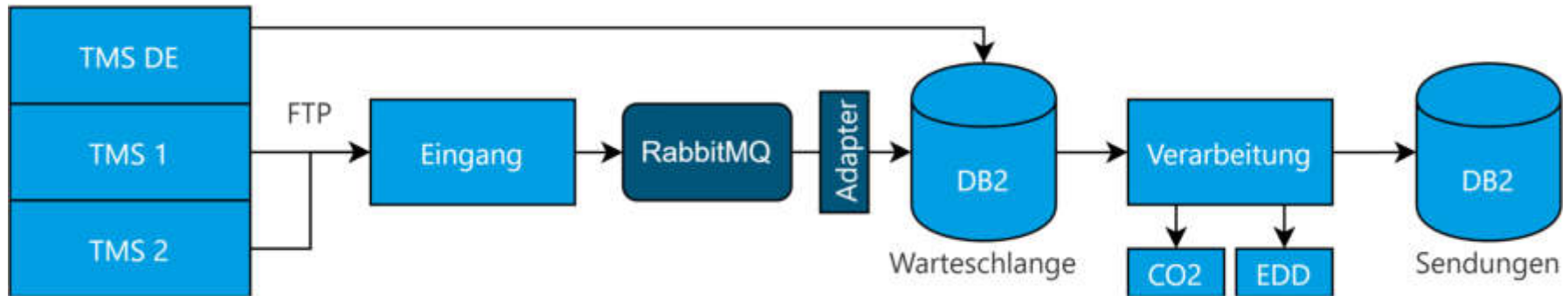


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress) -> Eigenentwicklung
- ◆ MySQL -> DB2 Datenbanklösung -> RabbitMQ
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

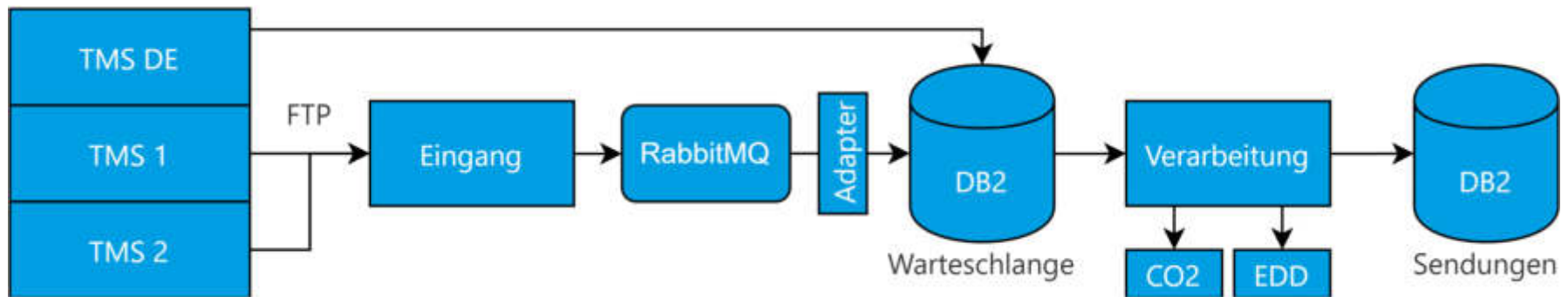


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress) -> Eigenentwicklung
- ◆ MySQL -> DB2 Datenbanklösung -> RabbitMQ
- ◆ TMS DE
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

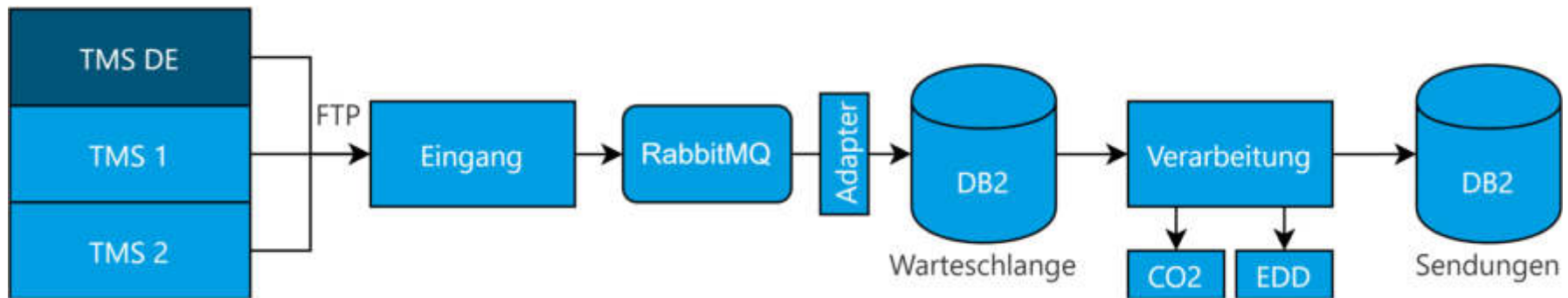


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress) -> Eigenentwicklung
- ◆ MySQL -> DB2 Datenbanklösung -> RabbitMQ
- ◆ TMS DE -> Überzeugungsarbeit + Product Owner
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)

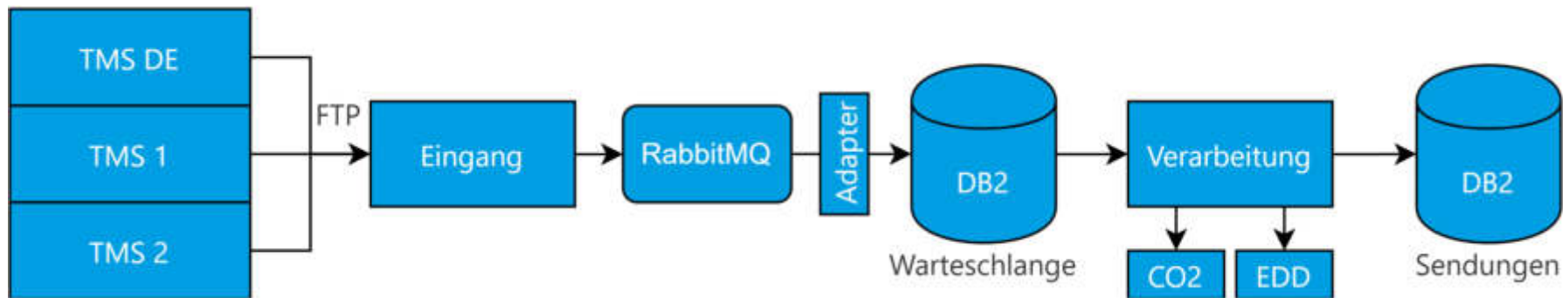


Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress) -> Eigenentwicklung
- ◆ MySQL -> DB2 Datenbanklösung -> RabbitMQ
- ◆ TMS DE -> Überzeugungsarbeit + Product Owner
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung)



Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Modernisierungen

- ◆ Firebird Datenbank -> MySQL Enterprise
- ◆ Eingangskomponente (TradeXpress) -> Eigenentwicklung
- ◆ MySQL -> DB2 Datenbanklösung -> RabbitMQ
- ◆ TMS DE -> Überzeugungsarbeit + Product Owner
- ◆ Verarbeitungskomponente (RPG Eigenentwicklung) -> Eigenentwicklung



Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Performance Verarbeitungskomponente

- ◆ RabbitMQ Nachrichten im Batch verarbeiten (bspw. 20 zusammen)
- ◆ CRUD Datenbankoperationen via Batch-Modus
- ◆ Weitere Parallelität durch mehrere Worker-Threads
- ◆ => Simple in-memory Reservierungssystem



Mut zur Modernisierung, ein Praxisbeispiel

Eingang & Verarbeitung

› Fazit

- ◆ Performancezuwachs durch Datentransfer & Parallelität
- ◆ Fachlogik präserter und wartbarer
- ◆ Verbesserte Transparenz
- ◆ Gesamte Sendung bekannt
- ◆ Publizieren in Output Stream möglich



Mut zur Modernisierung, ein Praxisbeispiel

Agenda

- › Kontext
- › Eingang & Verarbeitung
- › **Ausgang**
- › Benutzermanagement, Dokumente & Berichte
- › Fazit

Ausgang

> Herausforderungen

- ◆ Verarbeitungskomponente erzeugt keine Ausgabe
- ◆ Andere Komponenten nicht informiert über neue Sendungsevents
- ◆ Außer gezielt getriggert (CO2, EDD) => Wissen notwendig
- ◆ RabbitMQ für Ausgabe nutzen

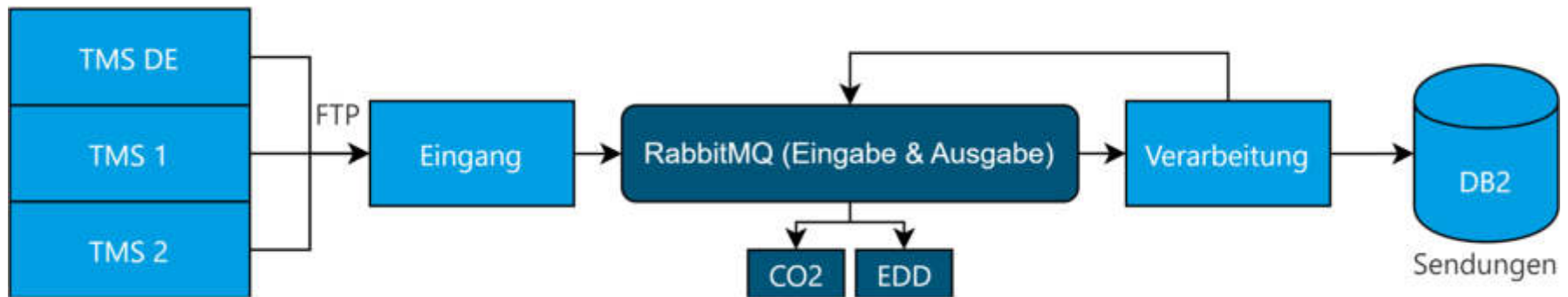


Mut zur Modernisierung, ein Praxisbeispiel

Ausgang

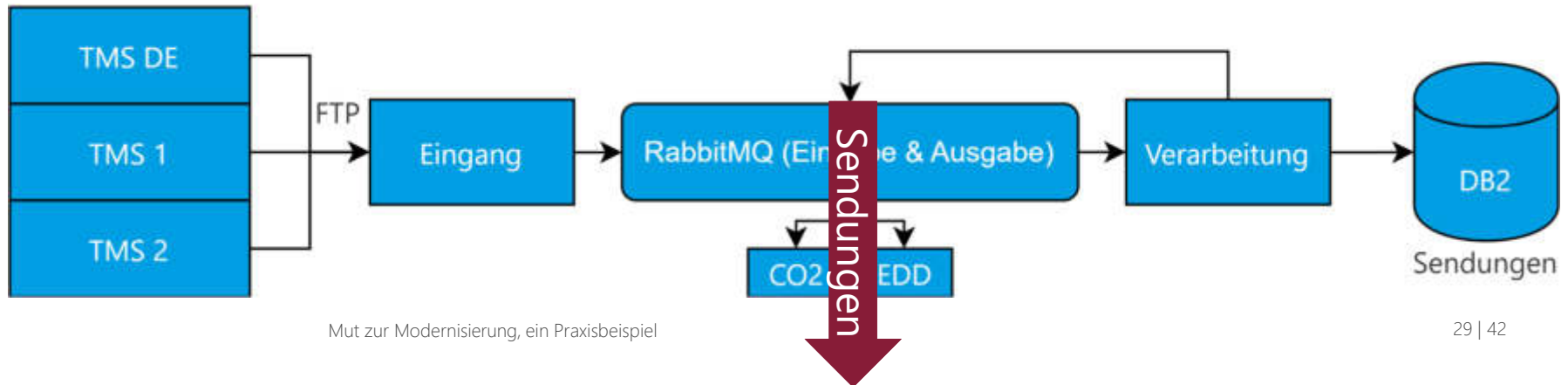
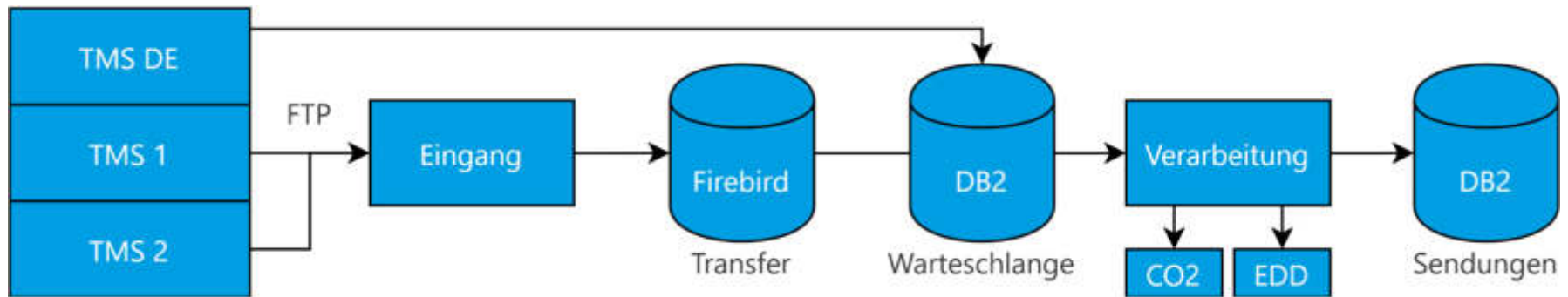
› Modernisierung

- ◆ Verarbeitungskomponente kennt gesamte Sendung während Bearbeitung
- ◆ Publiziert Sendungsinformation in RabbitMQ
- ◆ Konsumenten (CO2, EDD) konsumieren Sendungen nahe Echtzeit
- ◆ Triggerlogik in Verarbeitungskomponente entfällt



Mut zur Modernisierung, ein Praxisbeispiel

Eingang, Verarbeitung & Ausgang



Mut zur Modernisierung, ein Praxisbeispiel

Agenda

- › Kontext
- › Eingang & Verarbeitung
- › Ausgang
- › **Benutzermanagement, Dokumente & Berichte**
- › Fazit

Authentifizierung & Autorisierung

› Herausforderungen

- ◆ Alte Eigenentwicklung in Java mit JOSSO
- ◆ Verbunden mit LDAP des Kunden für Mitarbeiter
- ◆ Login & Rollenverwaltung für Benutzer
- ◆ Technologisch veraltet
- ◆ Sicherheitsbedenken
- ◆ Fehlende Unterstützung für
 - Standardworkflows
 - Standardprotokolle
 - 2FA
 - Social Login
 - Impersonate



Authentifizierung & Autorisierung

> Modernisierung

- ◆ Standardproblem => Standardsoftware
- ◆ Entscheidung für Keycloak
- ◆ Betrieben auf Server des Kunden
- ◆ Verbunden mit LDAP des Kunden
- ◆ Nachteil: Passwörter nicht-LDAP Benutzer
- ◆ Fazit:
 - Signifikanter Mehrwert,
 - (fast) geräuschlose Migration,
 - JWT,
 - wieder auf aktuellem Stand der Technik



Dokumente anfragen

› Herausforderungen

- ◆ Sendungsdokumente extern on-demand
- ◆ Anzeige in
 - Kundenfrontend
 - Mitarbeiterfrontend
 - Webservice
- ◆ 2 Java Module in 3 Applikationen
=> redundanter Wartungs- und Deploymentaufwand
- ◆ Preview / Thumbnail Erstellung
- ◆ PDF Erstellung



Dokumente anfragen

› Modernisierung

- ◆ Microservice Ansatz
- ◆ Alle Abfragen und Logik in einen eigenen Webservice
- ◆ Authentifizierung via Keycloak
- ◆ Dependency Probleme nur noch intern
- ◆ Probleme mit Dokumenten und Formaten protokolliert
- ◆ Refactoring ermöglichte Unittests
- ◆ Fazit: Entwicklungsprozess & Deployments beschleunigt



Dokumente verschicken

› Herausforderungen

- ◆ Eigenentwicklung zum Verschicken von Dokumenten
- ◆ Ursprünglich nur für einen Endkunden hektisch geplant
- ◆ Mit der Zeit in Komplexität & Anforderungen gewachsen
- ◆ Aktuell existieren 25 identische Instanzen
- ◆ Keine Oberfläche zur Konfiguration neuer Endkunden
- ◆ Nutzt Webservice zum Check für Sendungsänderungen



Dokumente verschicken

› Modernisierung

- ◆ Refactoring
- ◆ Beliebig viele Endkunden in einer Instanz
- ◆ Ausgabestream der Verarbeitungskomponente nutzen
- ◆ Proaktives Versenden nahe Echtzeit
- ◆ Frontend zur Konfiguration neuer Endkunden



Berichte

› Herausforderungen

- ◆ Regelmäßige konfigurierbare Sendungsberichte als Excel oder CSV
- ◆ Für Endkunden und Mitarbeiter
- ◆ JasperReportServer basierend auf JRXML
- ◆ Einbindung von JAR-Dateien für kundenspezifische Sendungslogik
- ◆ Sicherheitsbedenken
- ◆ Fehlende Performanz zu Spitzenzeiten (> 1000 Berichte um 8 Uhr)
- ◆ Wartung und Weiterentwicklung (xlsx Format) schwierig und aufwendig



Berichte

› Modernisierung

- ◆ Eigenentwicklung (angefangen als Azubi-Abschlussprojekt)
- ◆ Produktionsreife durch erfahrene Entwickler
- ◆ Verzicht auf Jasper und JRXML
- ◆ Nutzung von Apache POI und Apache Commons CSV
- ◆ Kundenspezifische Sendungslogik in Java
- ◆ Simple REST Schnittstelle
- ◆ Schrittweise Migration der bestehenden Berichte und Mitarbeiter zuerst
- ◆ Fazit: Performancegewinn von 400%-500%, Wartbarkeit verbessert



Agenda

- › Kontext
- › Eingang & Verarbeitung
- › Ausgang
- › Benutzermanagement, Dokumente & Berichte
- › **Fazit**

Fazit

- › Modernisierung ist planbar
- › Strategie ersetzt Mut
 - ◆ Kleine Schritte
 - ◆ Qualitätssicherung
 - manuelle Tests
 - automatisierte (Massen)-Tests
- › Verschiedene Modernisierungen
 - ◆ Eigenentwicklungen für komplexe Komponenten
 - ◆ Standardsoftware für Standardprobleme
 - ◆ Microservice Ansatz zur Entkoppelung
 - ◆ Refactorings



Fazit

- › „Nachher sieht die Modernisierung simpel und konsistent aus“
- › „Wir sind noch lange nicht fertig“
- › „Ist das Kunst oder kann das neu?“










Vielen Dank



www.iks-gmbh.com

Bilder

-  <https://www.onlinewebfonts.com/icon/538862>
-  <https://pictogrammers.com/library/mdi/icon/bottle-soda/>
-  <https://stock.adobe.com/de/images/logos-of-the-microsoft-office-component-excel-on-a-heap-copy-space-web-banner-format/481012251>
-  <https://stock.adobe.com/de/images/businessman-use-laptop-login-register-username-and-password-identity-on-webpage-concepts-of-cyber-security-internet-access-join-social-or-personal-data-protection-or-forget-pass-key-unlock/573237511>
-  <https://stock.adobe.com/de/images/hand-woman-signing-electronic-signature-on-tablet-for-agreement-of-contract-digital-receiving-parcel-from-blue-delivery-man-from-shopping-online-courier-man-delivering-package-to-destination/513463777>
-  <https://stock.adobe.com/de/images/data-transfer-concept-internet-server-connection/459460758>
-  <https://stock.adobe.com/de/images/businessman-holding-global-network-connection-big-data-analytics-and-business-intelligence-concept-world-map-point-and-line-composition-concept-of-global-business/588426797>