

# Vorteile einer standardisierten DV-orientierten BI-Architektur hinsichtlich Modellierung, Bewirtschaftung und Betrieb

Thomas Mattick, BBF GmbH





**Thomas Mattick**  
Senior Consultant,  
BBF GmbH

## Projektauszug (BI)

- Auftragsabwicklung/Leistungsbewertung (Luftfahrt/Logistik)
- Wareneingang/Warenausgang/Retouren (Logistik)
- Bestandsmanagement (Logistik)
- Auftragsmanagement (Großhandel)
- Flächenbewirtschaftung (Einzelhandel)
- B2B-Abwicklung (Verwertung)

## Technologien

- IBM Cognos 10.1 Certified
- Microsoft SQL Server
- Oracle
- Informatica Powercenter

## Einleitung

---

 Standardisierung

---

 Architektur

---

 Komponenten

---

 Schichten

---

 Zusammenfassung

## BI-Projekte und Ihre Eigenheiten

- Anforderungen
  - Ungenau
  - Unzureichend
  - Verändernd
  - Widersprüchlich
  
- Ursachen
  - Fehlende/Falsche Kenntnisse operativer Systeme
  - Unzureichende Vorstellungen des Möglichen
  - Wunsch nach mehr

## **BI-Projekte und Ihre wünschenswerten Eigenschaften**

- Agile Vorgehensmodelle
- Skalierbare Realisierungen
- Parallele Entwicklungsstränge
- Kompatible Implementierungen
- Wartungsfreundliche Komponenten
- Zentrale Protokollierung und Betrieb

## **Data Vault als Kern einer Standardisierten BI-Architektur**

- Hohe Regelorientierung
- Definierte Standards für Modellierung und Bewirtschaftung
- Abwärtskompatibilität
- Flexibilität

➤ Einleitung

---

➤ **Standardisierung**

---

➤ Architektur

---

➤ Komponenten

---

➤ Schichten

---

➤ Zusammenfassung

## **Prozentuale Aufwandsverschiebung hin zur Anforderungsdefinition**

- Definierte Methoden
  - Workshops
  - Konzeption
  - Implementierung
  
- Vorgehensmodelle für alle relevanten Projektphasen
  - Regelwerke
  - Konventionen
  - Vorlagenorientierung
  
- Hilfsmittel
  - Unterstützung durch formale Werkzeuge (z.B. Excelvorlagen)
  - Reglementierte Abläufe (z.B. Anforderungsaufnahme)
  - Einsatz von Projektbeschleunigern / zentralen Komponenten (z.B. Frameworks)

## **Methoden, Vorgehensmodelle und Hilfsmittel bilden vereinfacht**

– Bauplan mit Anleitung

## **Beantwortung von Fragen, deren antworten den Rahmen bilden**

1. Wie wird modelliert?
2. Wie wird implementiert?
3. Wie wird betrieben?



➤ Einleitung

---

➤ Standardisierung

---

➤ **Architektur**

---

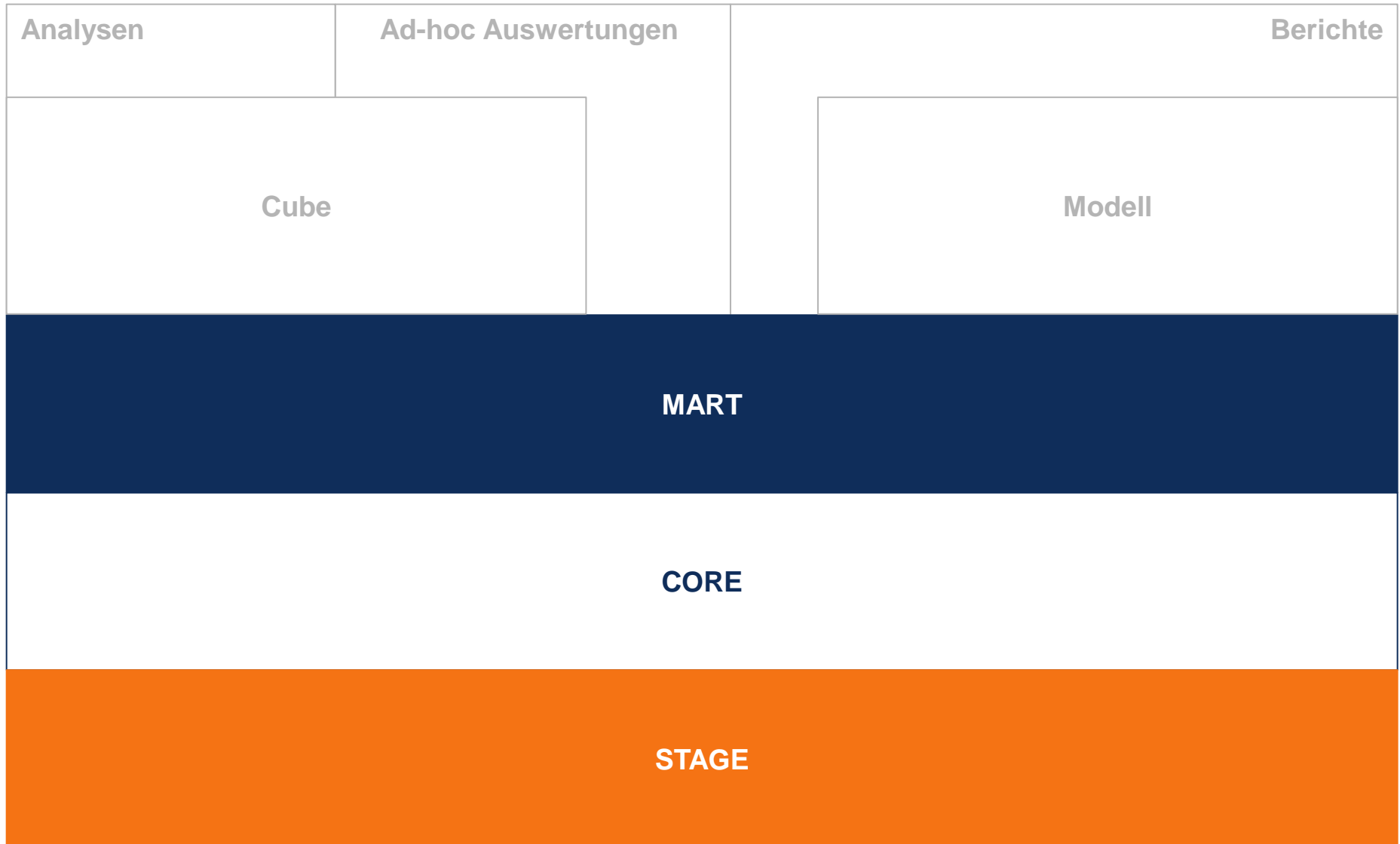
➤ Komponenten

---

➤ Schichten

---

➤ Zusammenfassung



- STAGE (Quellsystemorientiert)
  - Klassische schnittstellenorientierte 1:1 Datenabzugsschicht (n-STAGEs bei Heterogenität)
  - Konsumorientierte Persistenz
  
- CORE (DV - Ablageorientiert)
  - Datenmittelpunkt und einzige Datenwahrheit
  - Konsolidiert Geschäftsentitäten, verbindet diese und legt Nutzinformationen historisiert ab
  - Eigenschaften DV ermöglichen Vorzug Datentiefe vor Datenbreite
  
- MART (Stern – Abfrageorientiert)
  - Fachlicher "Endpunkt" (n-MARTs durch Fokussierung auf Anwender/Anwendung/Auswertung)
  - Daten in Form konsolidierter Dimensionen und dedizierter Fakten (Aggregationen, Kumulationen, Redundanz, -alles ablagefähige-)

## ➤ Standardisiert vereinfachte und festgelegte BI-Architektur

➤ Einleitung

---

➤ Standardisierung

---

➤ Architektur

---

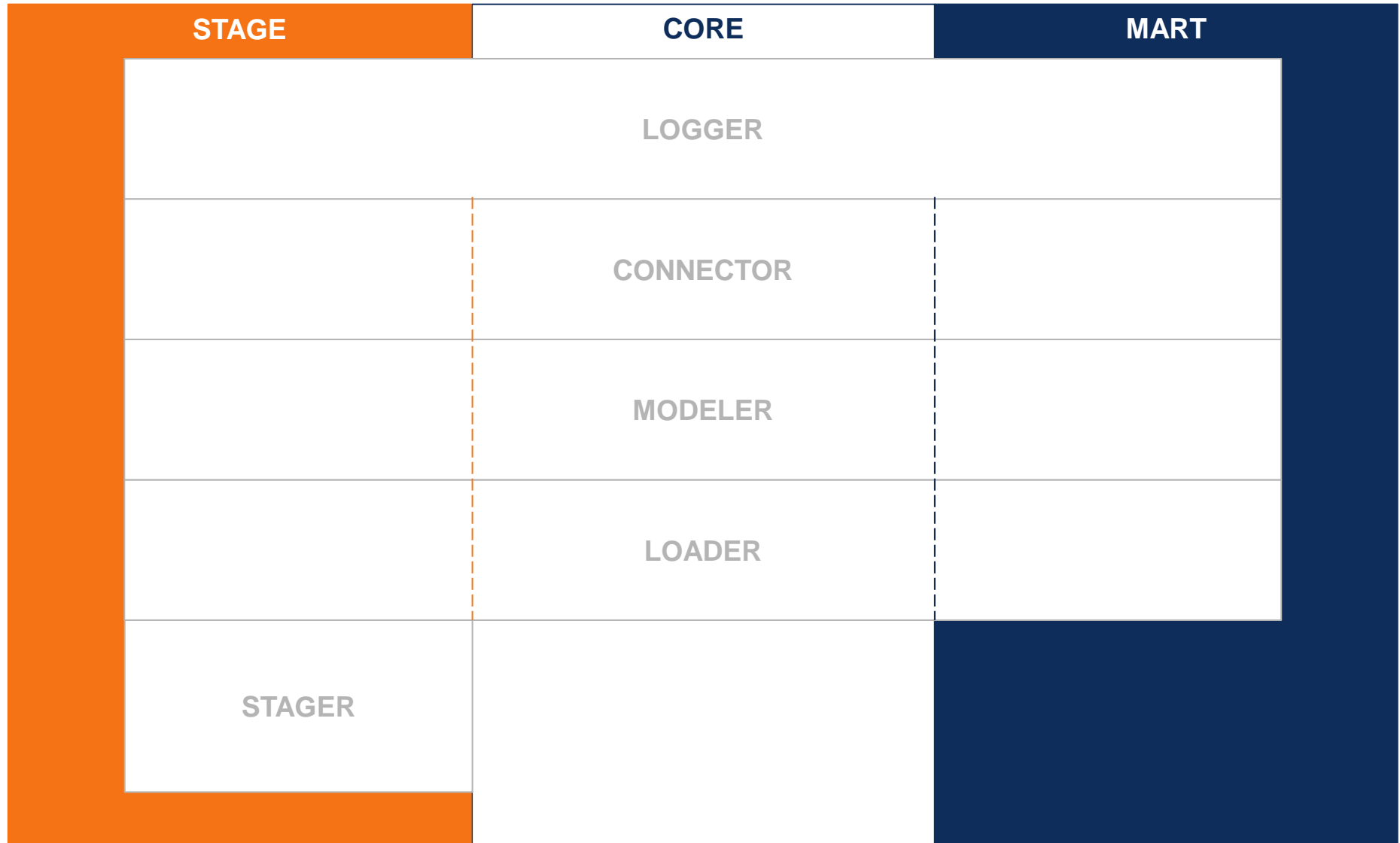
➤ **Komponenten**

---

➤ Schichten

---

➤ Zusammenfassung



## Zentral

- LOGGER
  - Zentrale Protokollierung
  - Bestehend aus Tabellen, Prozeduren und Regelwerken
  - Nachvollziehbarkeit von Datensatzlebenszyklen

## Individuell

- STAGER
  - Bildung einer 1:1 Abzugsschicht
  - Klont Datenbankschnittstellen (Tabellen, Views)
  - Rollout inklusive Steuerspalten und Tabelleninstanzen

## **Instanziert**

- CONNECTOR
  - Tabellen, Prozeduren und Regelwerke
  - Identifikation geänderter Daten
  - LOGGER Erhöhung/Verbesserung/Erweiterung der Selektionskriterien
  
- LOADER
  - Tabellen und Regelwerke
  - Definition und Konfiguration von Ladeketten
  - Konkretisierung durch Angabe von Aktivität, Ladeposition, Parallelisierung, Quellen, ...
  
- MODELER
  - Instanzierte Modellierungskomponente (technisch und fachlich)
  - Angabe von Quell- Zielzuordnungen, zentralisierter Steuerspalten, Bearbeitungsstatus, ...
  - Rollout

➤ Einleitung

---

➤ Standardisierung

---

➤ Architektur

---

➤ Komponenten

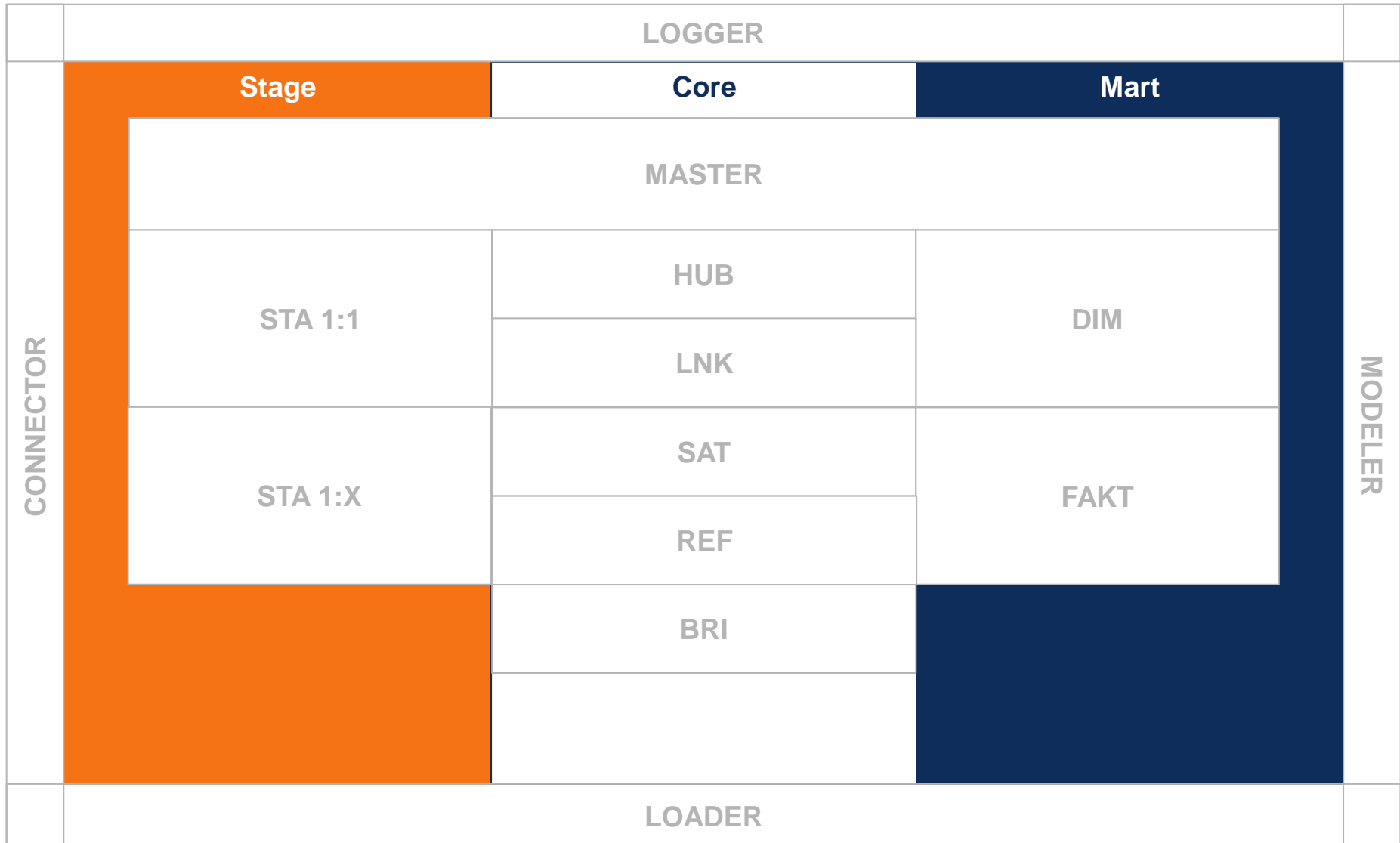
---

➤ **Schichten**

---

➤ Zusammenfassung





## Modellierung

### – STAGER

- Bildung generischer STAGE-Tabellenausprägungen
- Reine DB-Schnittstellen (Tabellen/Views)
- Quelldatentyp gleich Zieldatentyp
- Keine Constraints
- Generische Erweiterung um Steuerspalten

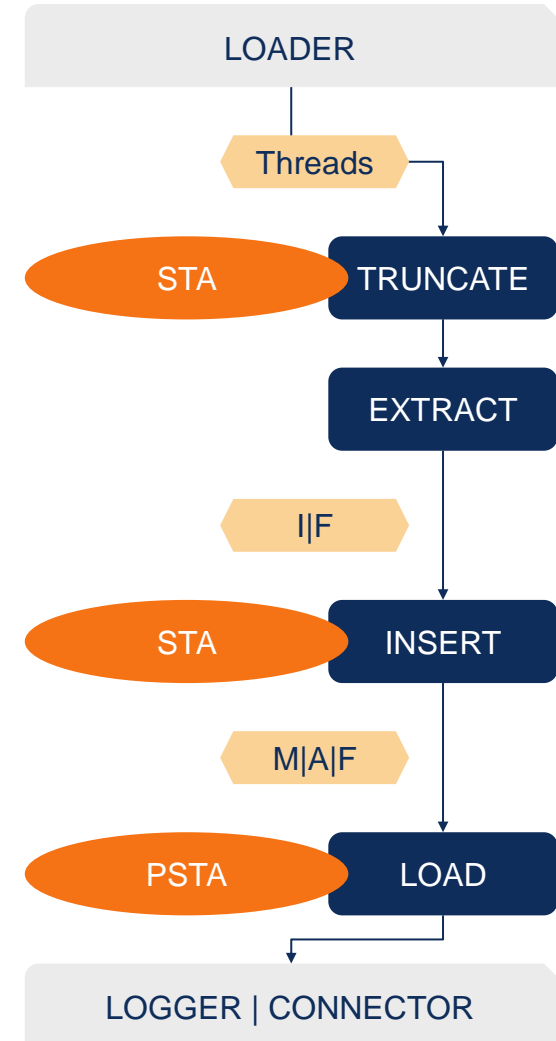
### – MODELER

- Bildung Konkreter STAGE-Tabellenausprägungen
- nonDB-Schnittstellen und changeDB-Schnittstellen (Bereinigung, Transformationen, ...)
- Variabler Zieldatentyp
- Dedizierte Constraints
- Zentrale Erweiterung um Steuerspalten

## Bewirtschaftung

- Master-ETL
  - Stage-ETL basierend auf STAGER
  - Individual-ETL, basierend auf MODELER
    - Einsatz mitgelieferter Vorlagen
    - Zeitstempelorientierte Deltalogik und Protokollierung
    - Atomare Abschlussoperation
- Eigenschaften
  - Komfortable Erstellung
  - Einfache Verarbeitungslogik
    - Performance
    - Wartbarkeit, Robustheit und Wiederanlauffähigkeit
    - Nachvollziehbarkeit
  - Protokollierend und Deltagebend (prozessorientiert)

## Master



## Betrieb

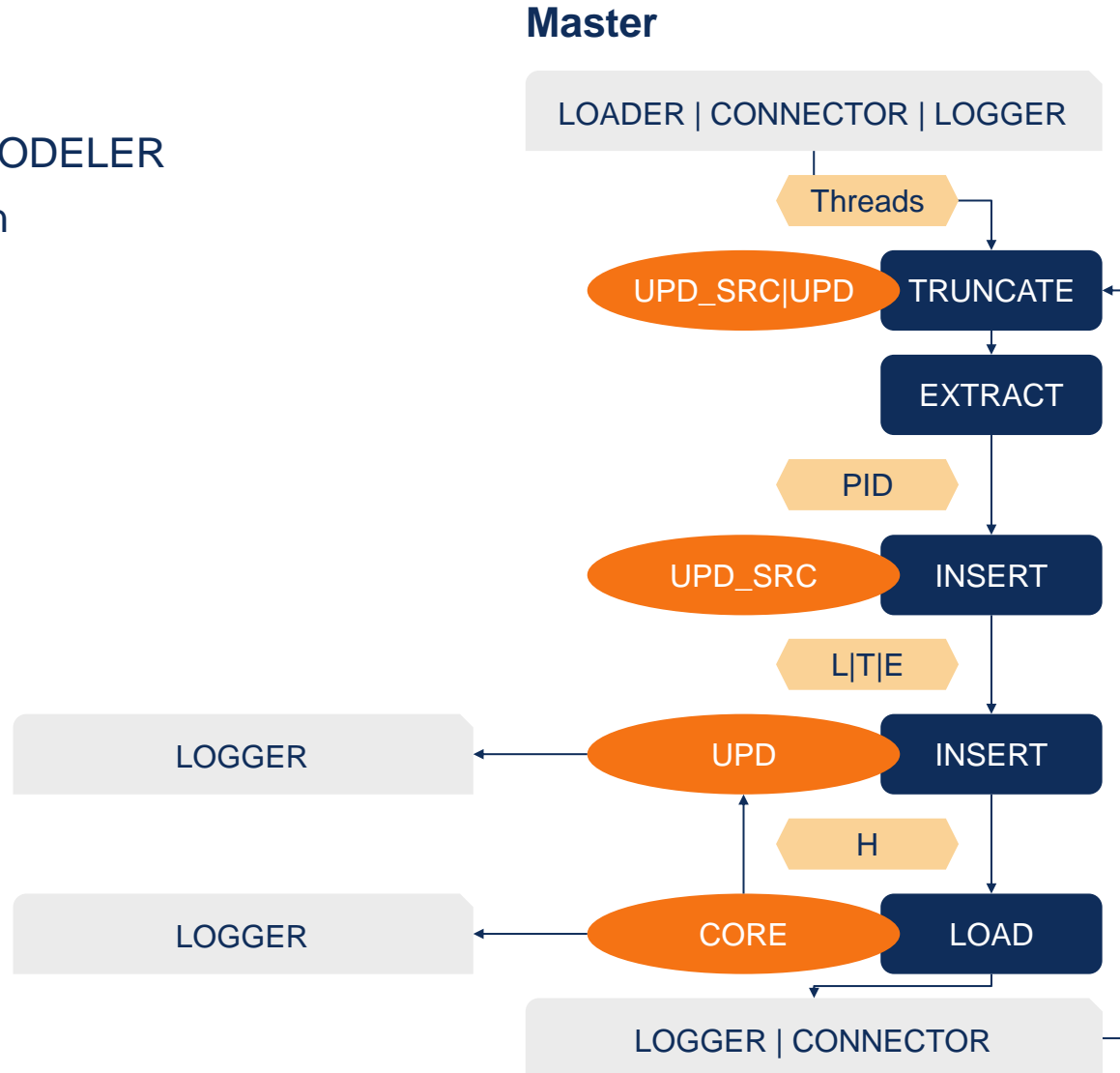
- Standardisiertes Regelwerk zur Konfiguration, Aufruf und Überwachung
- LOADER
  - Konfiguration der ETL-Strecken
  - Angabe dedizierter Eigenschaften (Quellen, Ziel, Paket, Delta, Lademethode, Thread, ...)
  - Einträge dienen dem Master-ETL als Grundlage
- Technologieorientierte Laufsteuerung
  - Aufruf Master-ETL
- LOGGER
  - Einrichtung des JOBs
  - Überwachung über LOGGER basierte Abfragen

## Modellierung

- Regelwerke für
  - Modellierungsformen (hier nach DV)
  - Namenskonventionen
  
- MODELER
  - Fachliche Modellierung
    - Tabellentyp, Tabellen und Spalten
    - Angabe von Berechnungsvorschriften
  - Technische Modellierung
    - Tabellenart, Datentyp (Konsolidierung), Constraints, Schlüsselbeziehungen und -arten
    - Quell- Zielzuordnung
  - Zentrale Bildung Konkreter CORE-Tabellenausprägungen
  - Zentrale Erweiterung um Steuerspalten anhand Tabellenart und -typ
  - Einfacher Rollout

## Bewirtschaftung

- Master-ETL
  - Individual-ETL, basierend auf MODELER
    - Einsatz mitgelieferter Vorlagen
    - Prozessorientierte Deltalogik und Protokollierung
    - Abhängigkeitsparadigma
    - Atomare Abschlussoperation



## Betrieb

- Standardisiertes Regelwerk zur Konfiguration, Aufruf und Überwachung
- LOADER
  - Konfiguration der ETL-Strecken
  - Angabe dedizierter Eigenschaften (Quellen, Ziel, Paket, Delta, Lademethode, Thread, ...)
  - Einträge dienen dem Master-ETL als Grundlage
- Technologieorientierte Laufsteuerung
  - Aufruf Master-ETL
- LOGGER
  - Einrichtung des JOBs
  - Überwachung über LOGGER basierte Abfragen

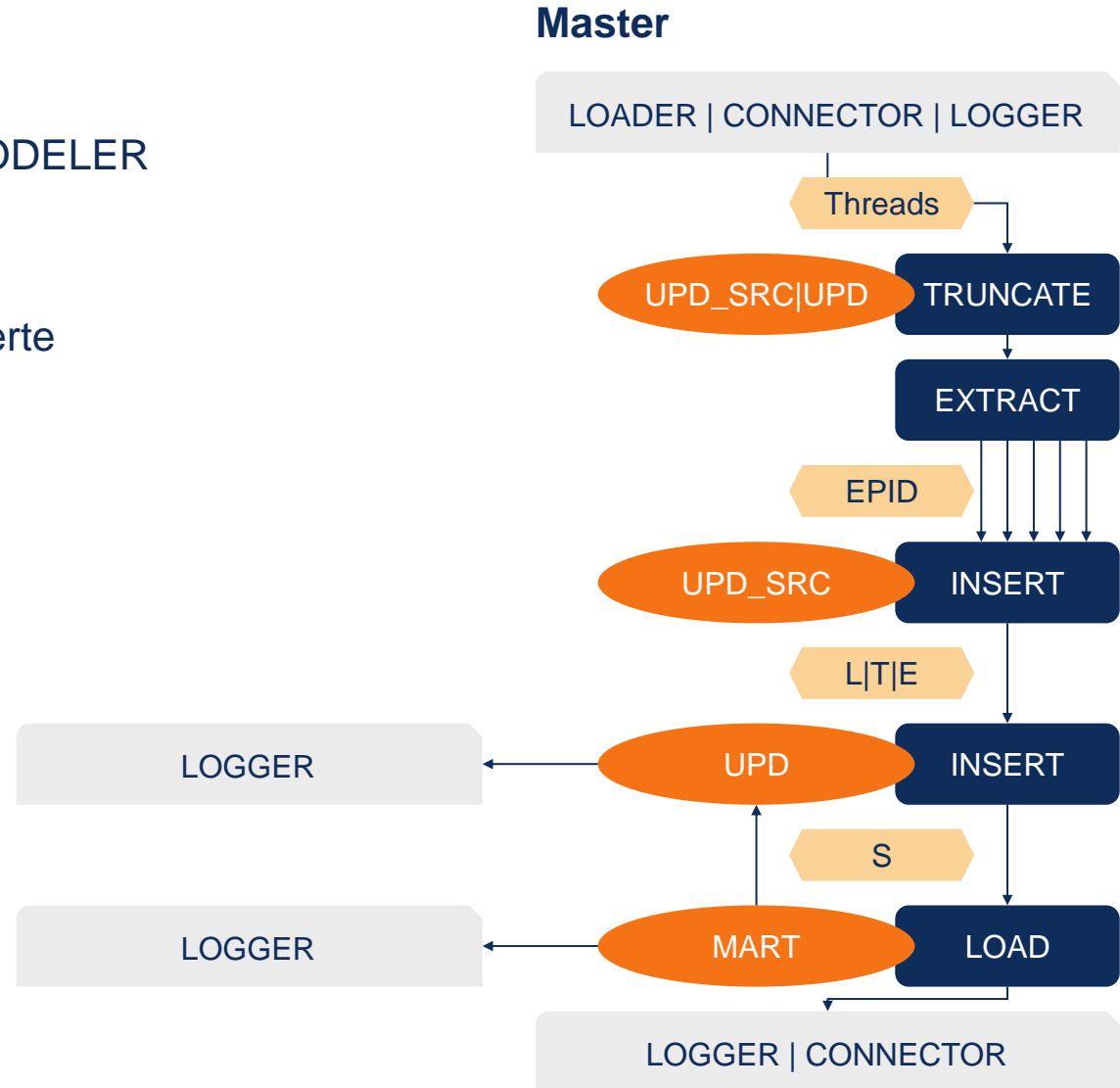
## Modellierung

- Regelwerke für
  - Modellierungsformen (hier nach Stern – Dimensionen und Fakten)
  - Namenskonventionen
  
- MODELER
  - Fachliche Modellierung
    - Tabellentyp, Tabellen und Spalten
    - Angabe von Berechnungsvorschriften
  - Technische Modellierung
    - Tabellenart, Datentyp (Konsolidierung), Constraints, Schlüsselbeziehungen und -arten
    - Quell- Zielzuordnung
  - Zentrale Bildung Konkreter MART-Tabellenausprägungen
  - Zentrale Erweiterung um Steuerspalten anhand Tabellenart und -typ
  - Einfacher Rollout



## Bewirtschaftung

- Master-ETL
  - Individual-ETL, basierend auf MODELER
    - Mitgelieferte Vorlagen und Skriptdimensionen
    - Prozessbasierte entitätsorientierte Deltalogik und Protokollierung
    - Abhängigkeitsparadigma
    - Atomare Abschlussoperation



## Betrieb

- Standardisiertes Regelwerk zur Konfiguration, Aufruf und Überwachung
- LOADER
  - Konfiguration der ETL-Strecken
  - Angabe dedizierter Eigenschaften (Quellen, Ziel, Paket, Delta, Lademethode, Thread, ...)
  - Einträge dienen dem Master-ETL als Grundlage
- Technologieorientierte Laufsteuerung
  - Aufruf Master-ETL
- LOGGER
  - Einrichtung des JOBs
  - Überwachung über LOGGER basierte Abfragen

➤ Einleitung

---

➤ Standardisierung

---

➤ Architektur

---

➤ Komponenten

---

➤ Schichten

---

➤ **Zusammenfassung**

## **Kosten/Nutzen Argumentation**

- Kurze Einarbeitungszeiten für Modellierer
- Kurze Einarbeitungszeiten für Entwickler
  - Einsatz von Generatoren, Generika und Vorlagen
  - Namenskonventionen und Konsolidierung von Objekten und Objekttypen
- Geringe Dokumentationskosten
- Parallelisierung von Projektaufgaben
  - Schichtenunabhängigkeit über definierte Schnittstellen
  - Leichte Ermittlung von Verarbeitungsergebnissen
  - Data Lineage
- Einfacher Betrieb der ETL-Strecken

# Vielen Dank!

BBF GmbH

Erika-Mann-Str. 57  
80636 München

Tel. +49 (0) 89 189099 - 0  
Web [www.bbf-soft.de](http://www.bbf-soft.de)  
eMail [info@bbf-soft.de](mailto:info@bbf-soft.de)

**Kontakt:**

Thomas Mattick  
Senior Consultant

Tel. +49 (0) 351 811 954 10  
Mobil +49 (0) 176 414 56 427  
eMail [thomas.mattick@bbf-soft.de](mailto:thomas.mattick@bbf-soft.de)

