# Einleitung

* Grundlagen zum Thema Batch-Verarbeitung
  + Was gibt es für unterschiedliche Datenquellen?
  + Stichwörter im Zusammenhang mit Batch Processing:
    - Scheduling
    - Logging / Tracing
    - Transaction Management
    - Bulk Processing
    - Processing Statistics
    - Job Restart
    - Skip, Retry
    - Resource Management
* Domain Language
* Spring Batch Architecture:
  + Application / Batch Core / Batch Infrastructure
  + Configuration: XML or Java / JobBuilderFactory

# System Context (Raffi)

* Big Picture / System Context aufzeigen: wie ist die Applikation ins Gesamtsystem eingebettet?
  + Deployment Diagramm
  + Web Schnittstelle
  + File Schnittstelle
  + gemeinsame Rule Engine
  + gemeinsame Datenbank
* Use Cases beschreiben
  + Import, Export aus anderen Systemen
  + Manuelle Einlieferungen
  + Datenumvolumen
* (Verwendete Technologien)
  + (Wieso Spring Batch?)
  + (Alternativen?)
* Charakteristiken der Daten und Operationen
  + Idempotent
  + Abhängigkeiten zwischen Datensätzen

# Load via Staging Area (Raffi)

* Anforderung
  + Parallelisierung, Partitionierung
  + Abhängigkeiten zwischen Datensätzen
  + Ausführung von nur einem Datensatz respektive einem Range von Datensätzen
  + Reprocessing von Datensätzen
* -Umsetzung
  + Staging Table
  + Partition Key, jedes File in eigene Partition
  + Cleanup
  + Rohdaten als Varchar
* Evt. Beispiel wie Staging Table mit Spring Batch befüllt werden kann.

# Performance (Skalierung, Partitionierung, Chunk-Size) (Mugi)

* Ausblick: Evt. Remote Chunking, wie könnte man auf unterschiedliche Knoten skalieren?

# Monitoring / Tracing (Mugi)

* Anforderungen
  + Explain Plan: Prozessierung von einem Item im fehlgeschlagenen Zustand (read-only mode).
  + Statistik (real time und history)
  + Wie viele Records wurden verarbeitet / sind gefailed / sind erfolgreich?
  + Wie kommt der Benutzer and die Ausgaben des Systems?
  + Zusammenfassung des Ladeprozesses
* Umsetzung
  + Spring Batch Admin
  + Explain Plan
  + Execution Log

# Job Control (Raffi)

* Anforderung
  + Deferred Job Execution
  + Deaktivierung Job Execution (Bsp. Bugs in System)
  + Abhängigkeiten zwischen Jobs
* Umsetzung

# Optional: Ausblick

* Neuigkeiten in Spring Batch 3.0.
* Spring Batch und JSR 352, was beinhalted JSR 352 nicht? Bzw. was ist nicht standardisiert?
* Ein paar Code-Schnipsel:
  + JdbcReader
  + FileReader
  + FileWriter
  + Skalierung? Partitionierung?