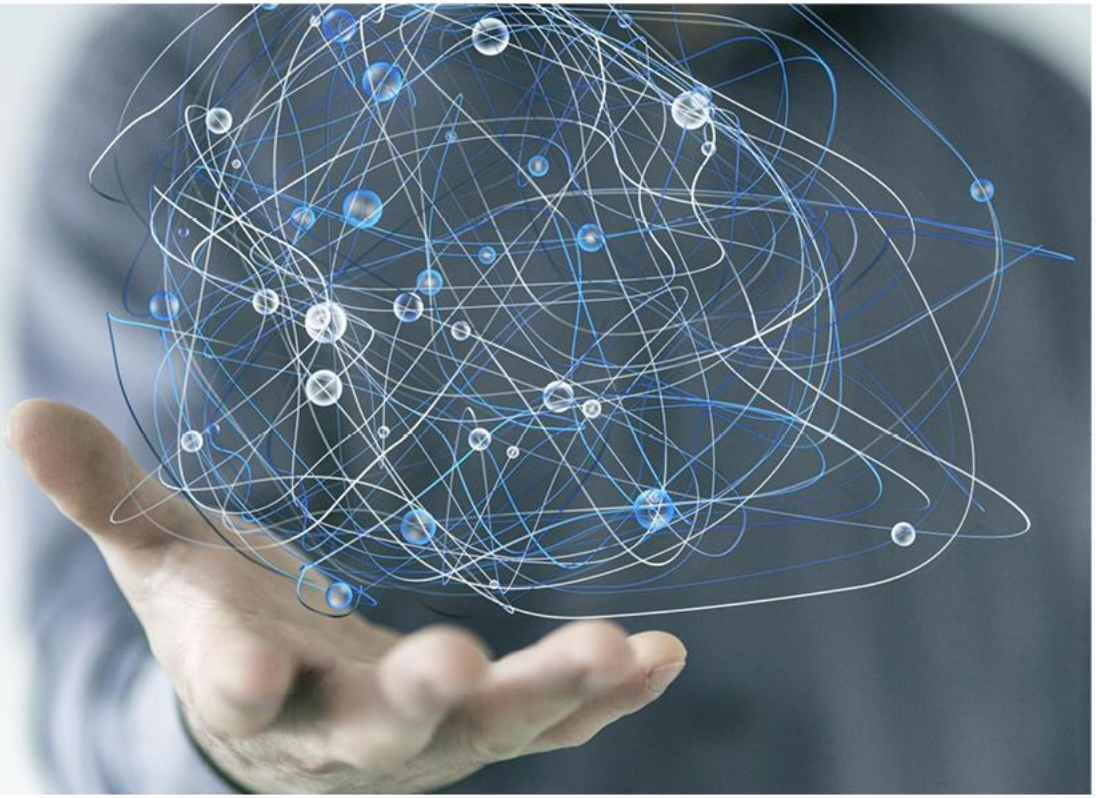
CaDS

communication and distributed systems



Lambda Architektur für verteilte mobile Sensoren

von Mike Wüstenberg

Übersicht

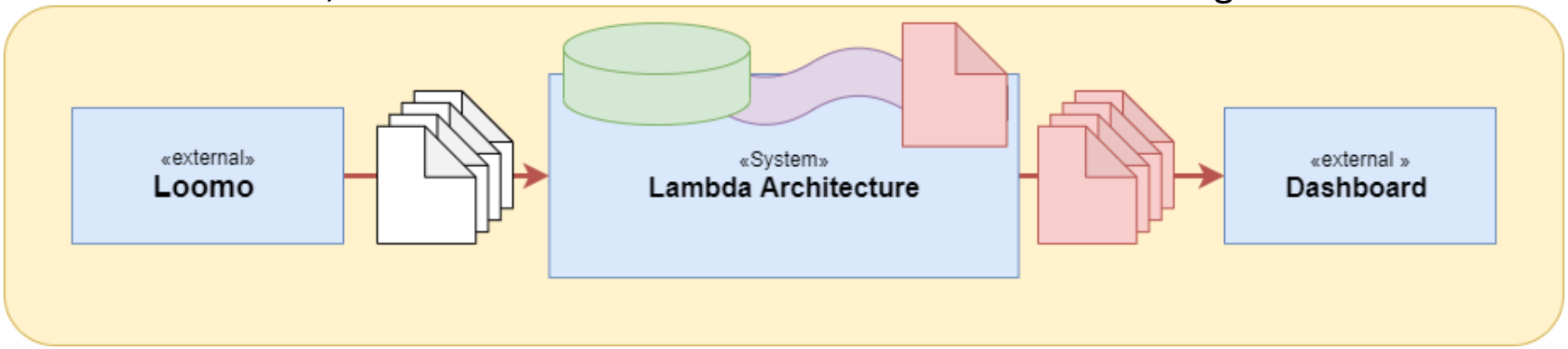
CaDS
communication and distributed systems

1. Projekt Übersicht
2. Warum keine traditionellen Datenbank
3. Anforderungen an die Lambda Architektur
4. Aufbau der Lambda Architektur
5. Experiment

Projekt Übersicht

CaDS
communication and distributed systems

Eine Lambda Architektur ist eine,
Generische, skalierbare und fehlertolerante Datenverarbeitungsarchitektur.



Beispiel: Distanz Berechnung

{ 10:02, (10,10,0) }

Neu Daten

Zeit	Koordinaten
10:00	(0,0,0)
10:01	(0,10,0)
…	…

Berechnen

Distanz
0
10
…

CaDS
communication and distributed systems

Problem Beschreibung

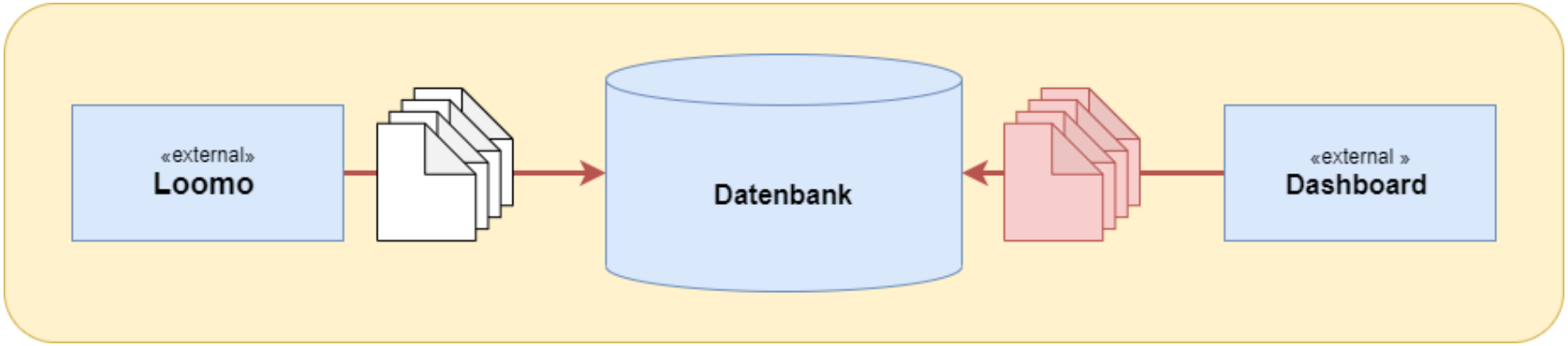
- Traditioneller Ansatz
- Monolithische relationale Datenbank

Problem Beschreibung

CaDS
communication and distributed systems

- 👍 Einfacher Aufbau
- 👍 Einfache SQL Anfragen

- 👎 Kleine Anzahl an Nachrichten



{ 10:02, (10,10,0) }

Neu Daten

Zeit	Koordinaten
10:00	(0,0,0)
10:01	(0,10,0)
…	…

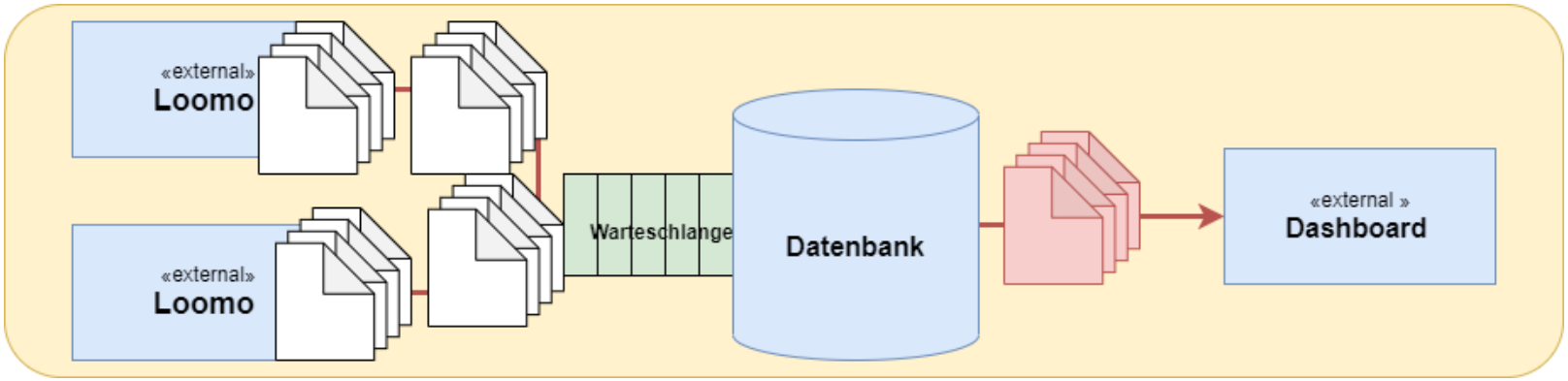
SQL Anfrage

Distanz
0
10
…

Problem Beschreibung

CaDS
communication and distributed systems

- Schreibvorgänge überwältigen Datenbank
- Skalierung wird Notwendig



{ 10:02, (10,10,0) }
{ 10:03, (10,20,0) }
…

ERROR

Zeit	Koordinaten
10:00	(0,0,0)
10:01	(0,10,0)
…	…

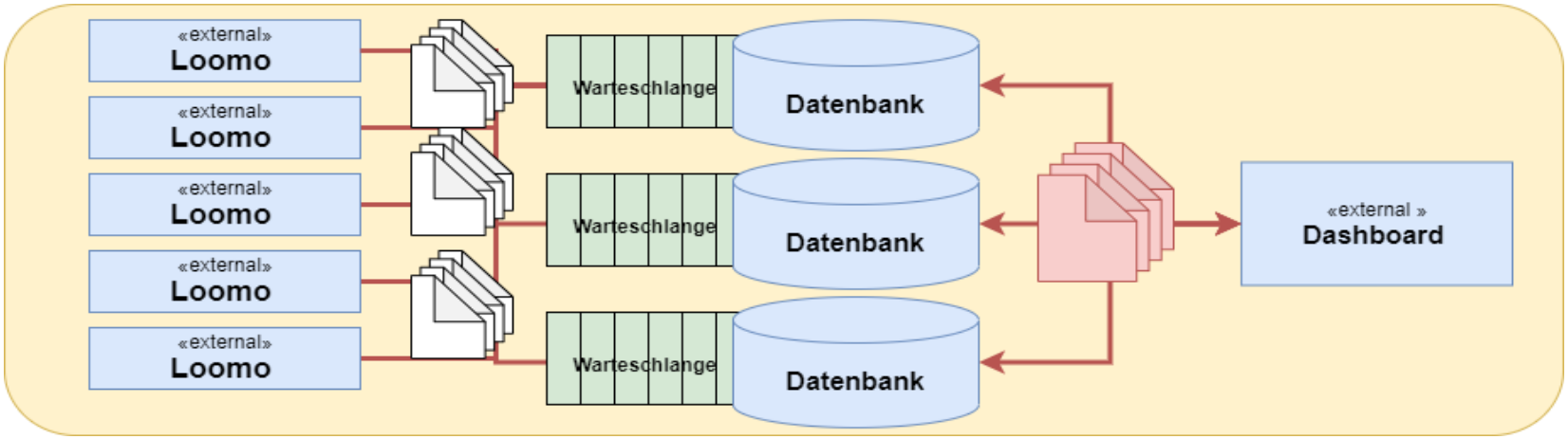
SQL Anfrage

Distanz
0
10
…

Problem Beschreibung

CaDS
communication and distributed systems

- 👎 Komplizierter Aufbau durch „Shard“ Verwaltung
- 👎 SQL Anfragen werden Komplizierter



{ 10:02, (10,10,0) }
{ 10:03, (10,20,0) }
…

Neue Daten

Neue Daten

Zeit	Koordinaten
10:00	(0,0,0)

Zeit	Koordinaten
10:01	(0,10,0)

SQL Anfrage

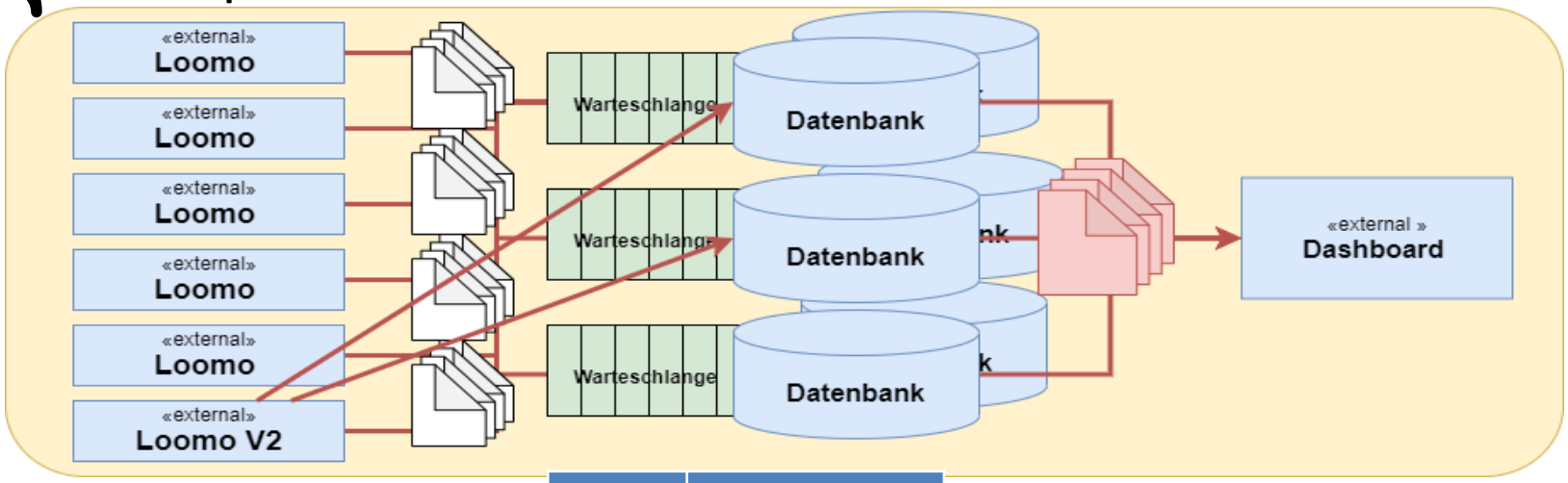
SQL Anfrage

Distanz
0
10
…

Problem Beschreibung

CaDS
communication and distributed systems

- 👎 Fehlertoleranz Probleme
- 👎 Korrupte Daten



{ 10:02, (10,10,0) }
{ 10:03, (10,20,0) }
…

2x Neue Daten

Vergessen

Zeit	Koordinaten
10:00	(0,0,0)

Zeit	Koordinaten
10:01	(0,10,0)

SQL Anfrage

SQL Anfrage

Distanz
0
10
??15??

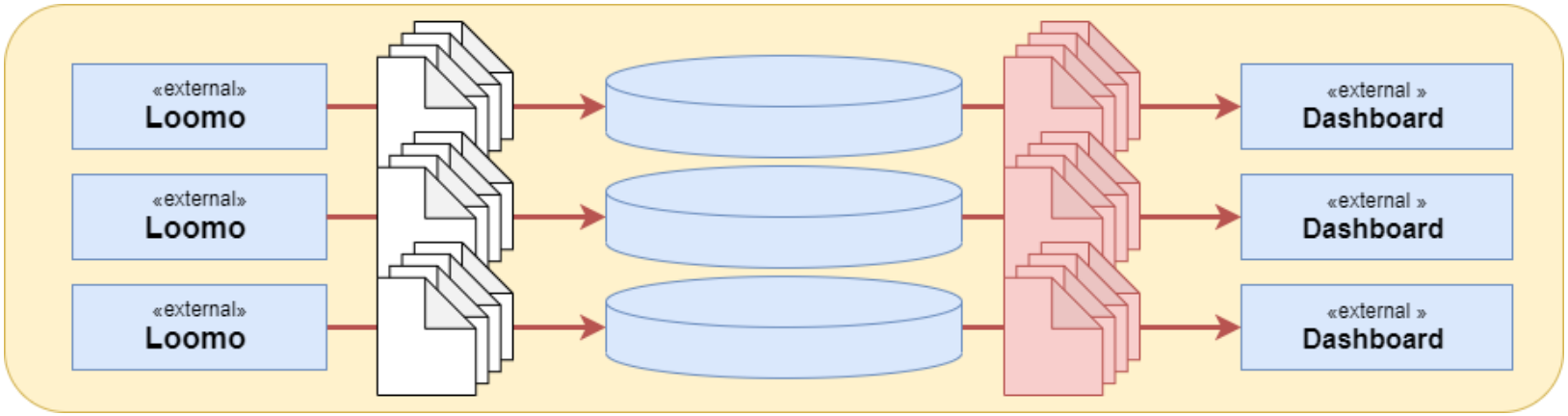
CaDS
communication and distributed systems

Anforderungen

- Lambda Architektur

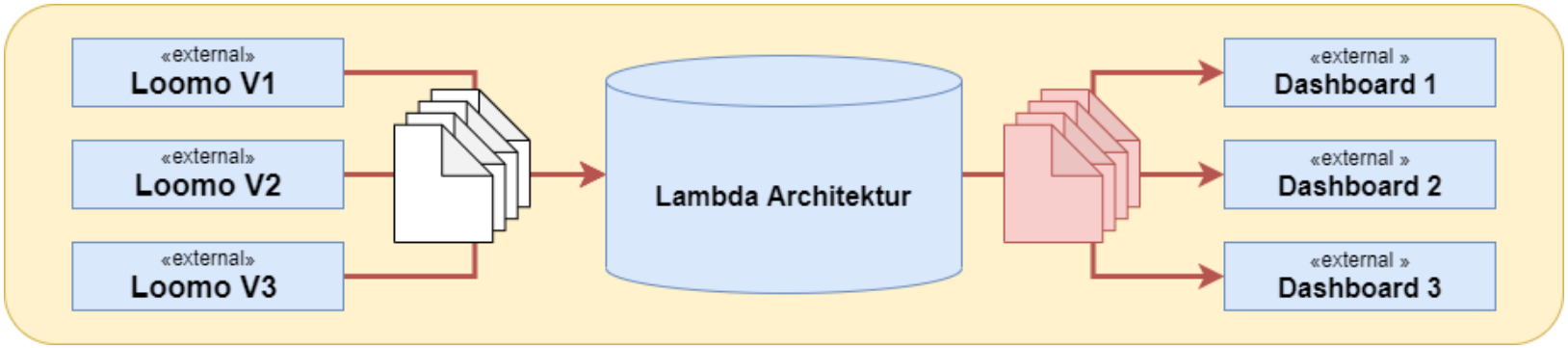
Anforderungen

- Skalierbarkeit
  - Auf allen Ebenen
  - Verteilte Software vom Start
  - Horizontale Skalierung



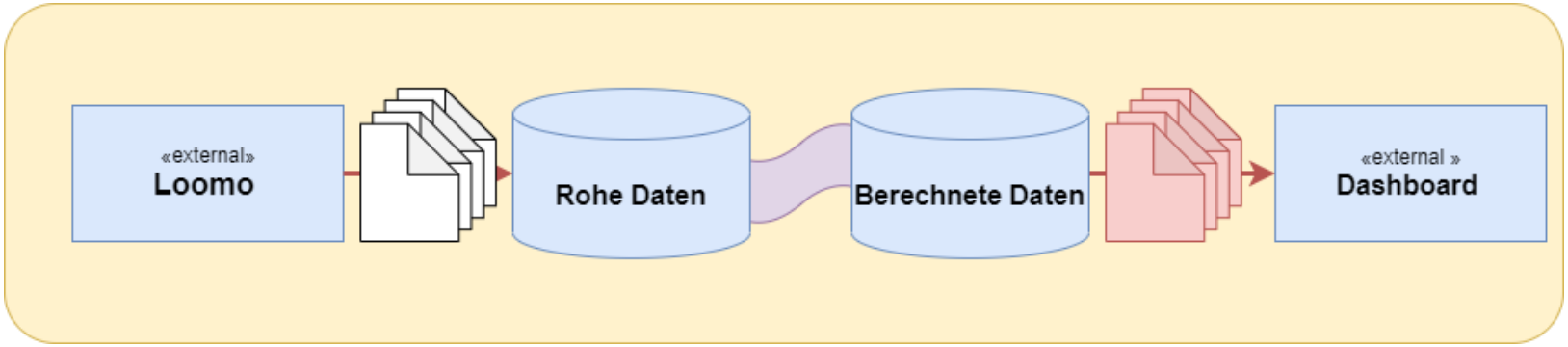
Anforderungen

- Erweiterbarkeit
  - Neue Funktionen hinzuzufügen
  - Gleichbleibendes System
  - Migration der Daten in neues Format



Anforderungen

- Fehlertoleranz
  - gegenüber Menschlichen Fehler
  - Neuberechnung von Ergebnissen
  - Unveränderlicher Daten



Anforderungen

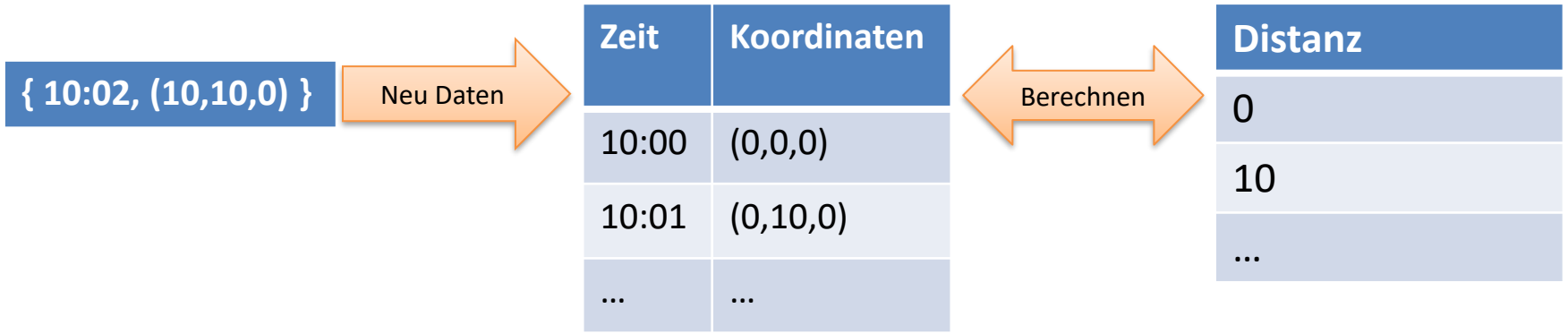
CaDS
communication and distributed systems

- Minimale Wartung
  - Reduzierung von Komplexität in Implementierung
  - Komplexität nicht in Kern Komponenten
  - Komplexität in Daten die verworfen werden
- Geringe Latenz
  - Schnelles Lesen
  - Schnelles schreiben wenn nötig

CaDS
communication and distributed systems

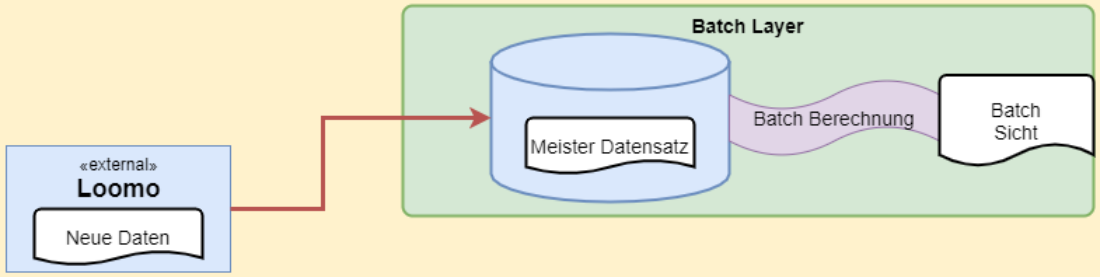
Lambda Architektur

- Jede Ebene skalierbar
- 3 Layer



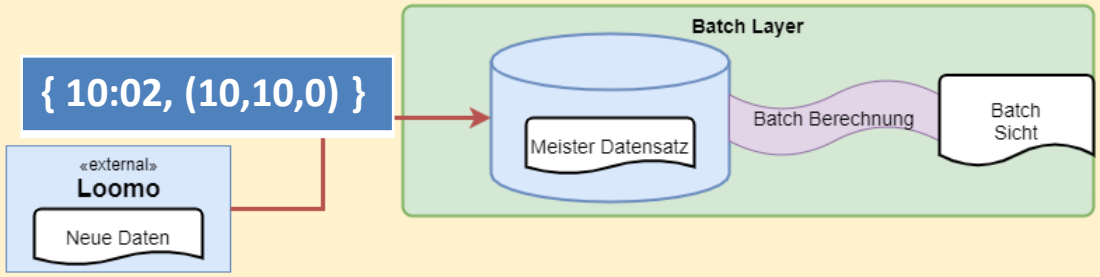
Lambda Architektur

- Batch Layer
  - Speichert Daten im Original zustand
  - Alle Berechnungen auf kompletten Datensatz



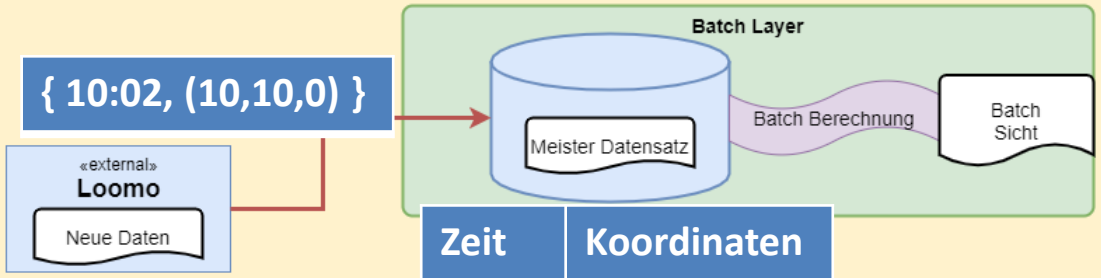
Lambda Architektur

- Batch Layer
  - Speichert Daten im Original zustand
  - Alle Berechnungen auf kompletten Datensatz



Lambda Architektur

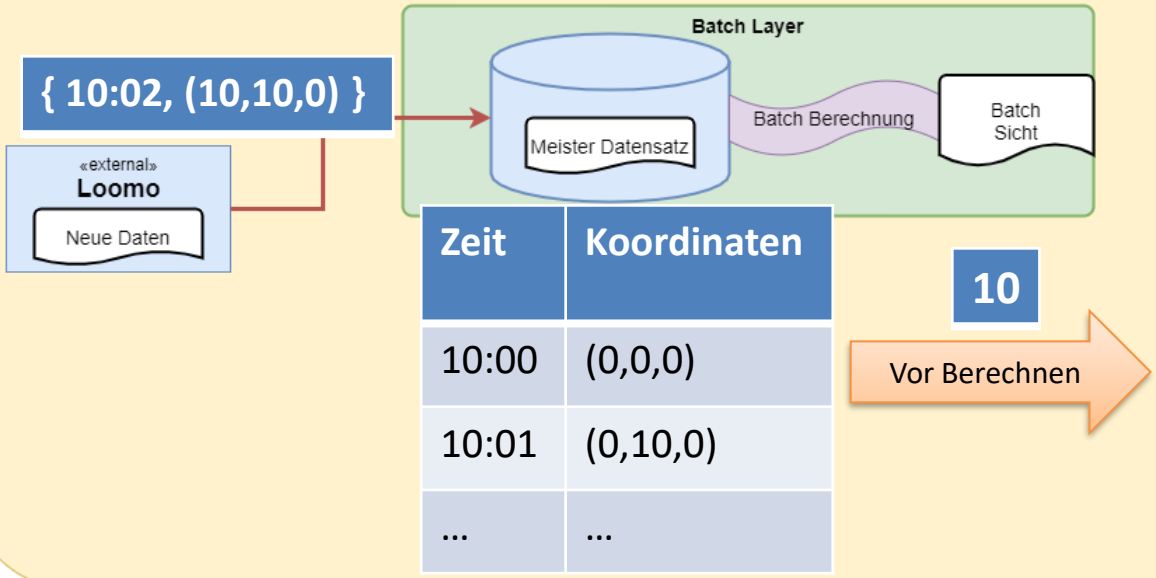
- Batch Layer
  - Speichert Daten im Original zustand
  - Alle Berechnungen auf kompletten Datensatz



Zeit	Koordinaten
10:00	(0,0,0)
10:01	(0,10,0)
…	…

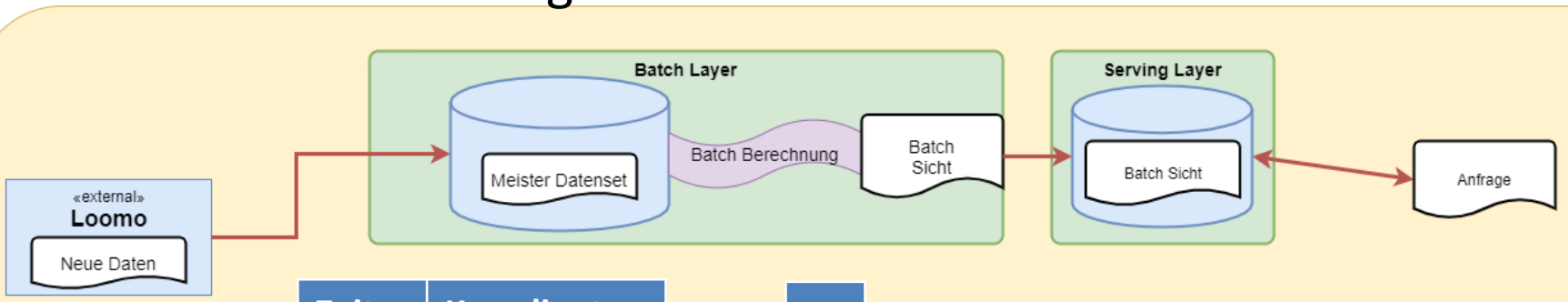
Lambda Architektur

- Batch Layer
  - Speichert Daten im Original zustand
  - Alle Berechnungen auf kompletten Datensatz



Lambda Architektur

- Serving Layer
  - Speichert Vorberechnete Daten vom Batch Layer
  - Bietet Vorberechnete Daten an
  - Muss nur wenig können



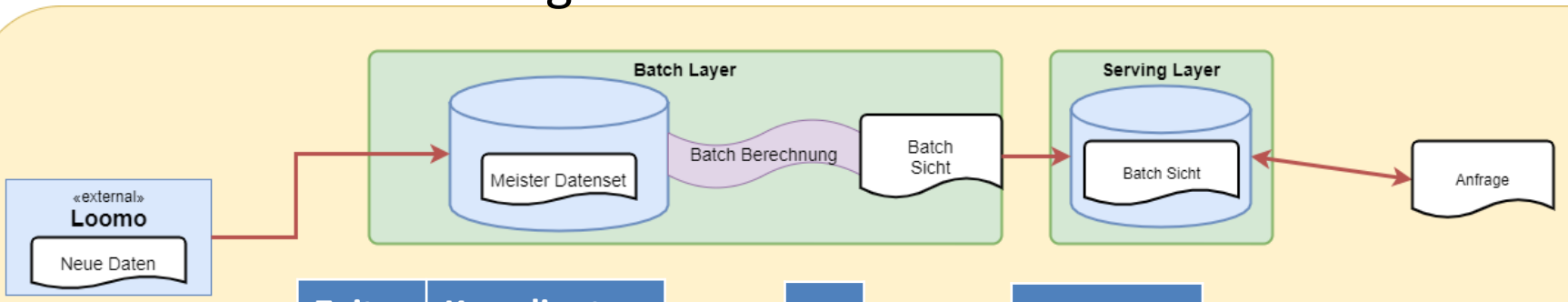
Zeit	Koordinaten
10:00	(0,0,0)
10:01	(0,10,0)
…	…

10

Vor Berechnen

Lambda Architektur

- Serving Layer
  - Speichert Vorberechnete Daten vom Batch Layer
  - Bietet Vorberechnete Daten an
  - Muss nur wenig können



Zeit	Koordinaten
10:00	(0,0,0)
10:01	(0,10,0)
…	…

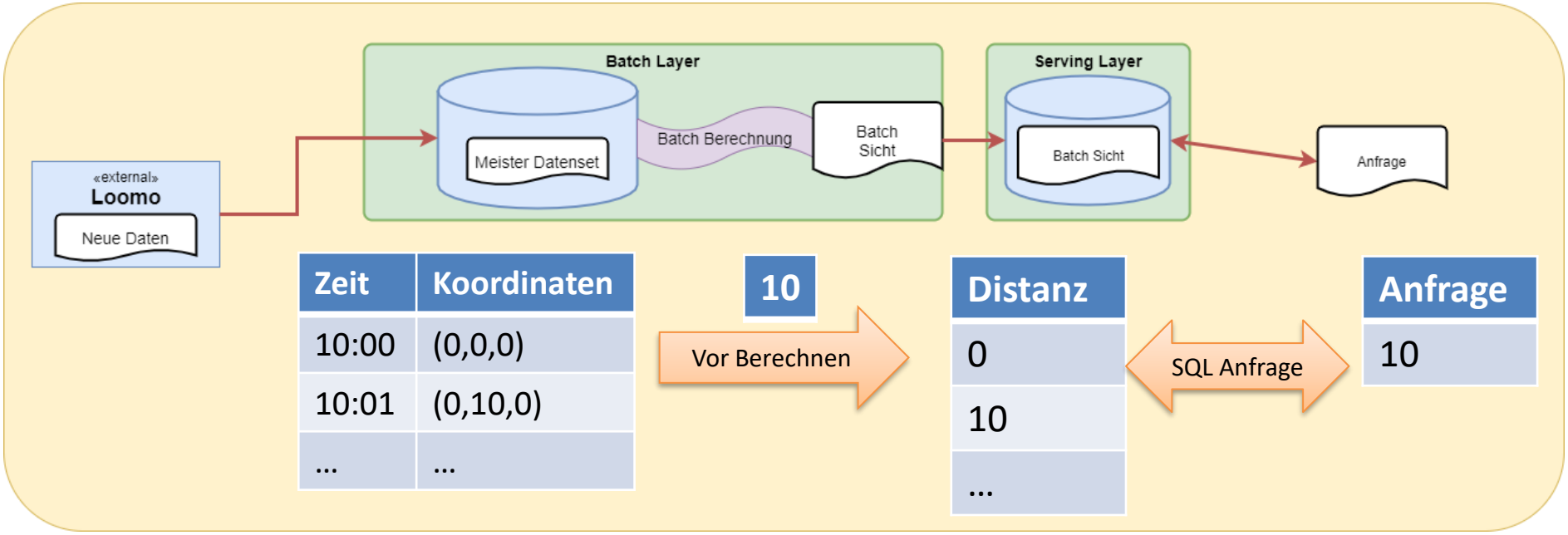
10

Vor Berechnen

Distanz
0
10
…

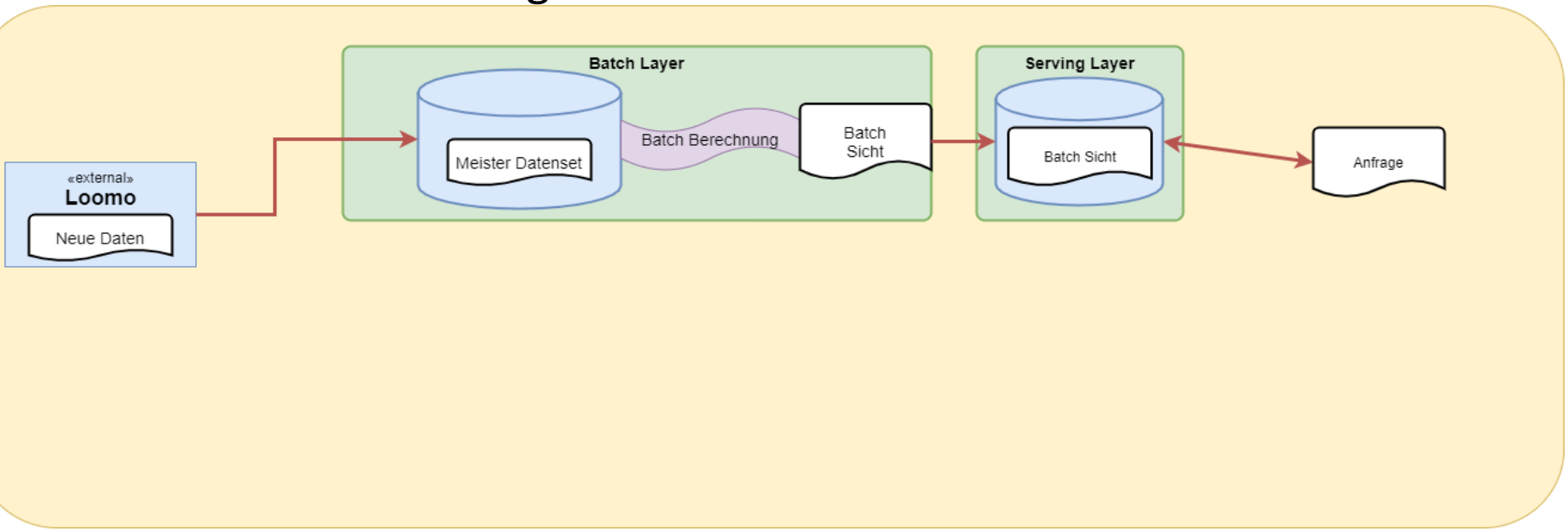
Lambda Architektur

- Serving Layer
  - Speichert Vorberechnete Daten vom Batch Layer
  - Bietet Vorberechnete Daten an
  - Muss nur wenig können



Lambda Architektur

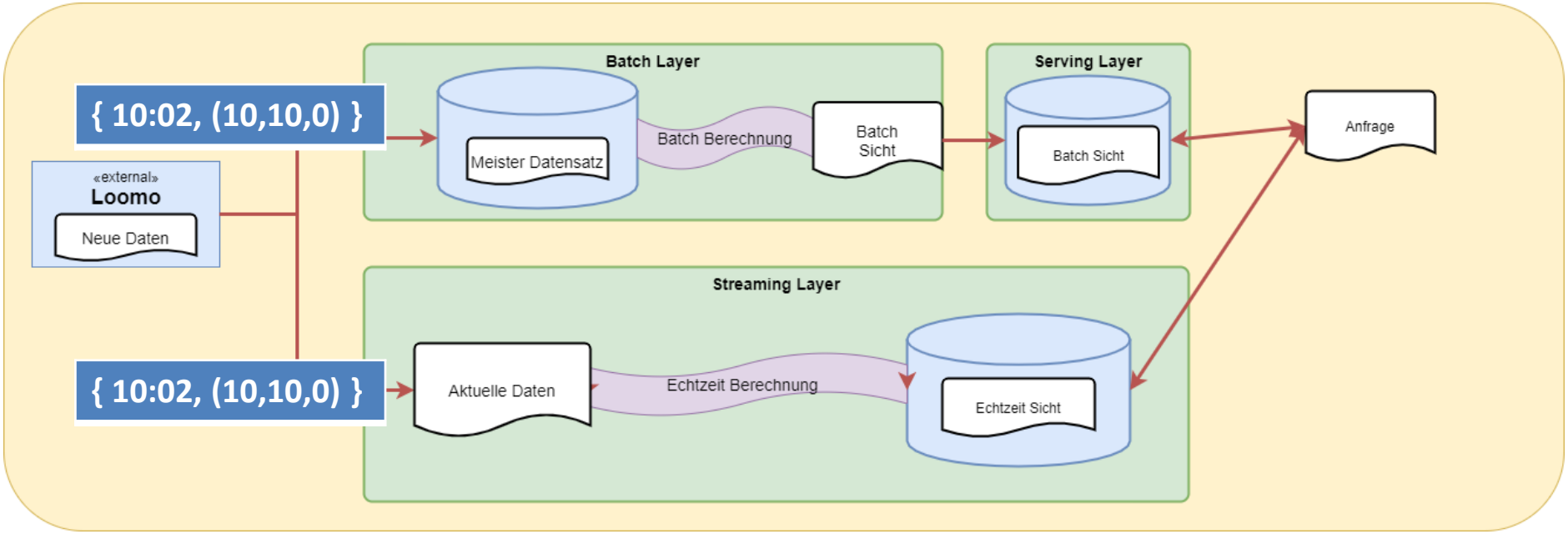
- Erfühlte Anforderungen
  - Skalierbar
  - Fehlertoleranz
  - Erweiterbarkeit
  - Minimale Wartung



Lambda Architektur

CaDS
communication and distributed systems

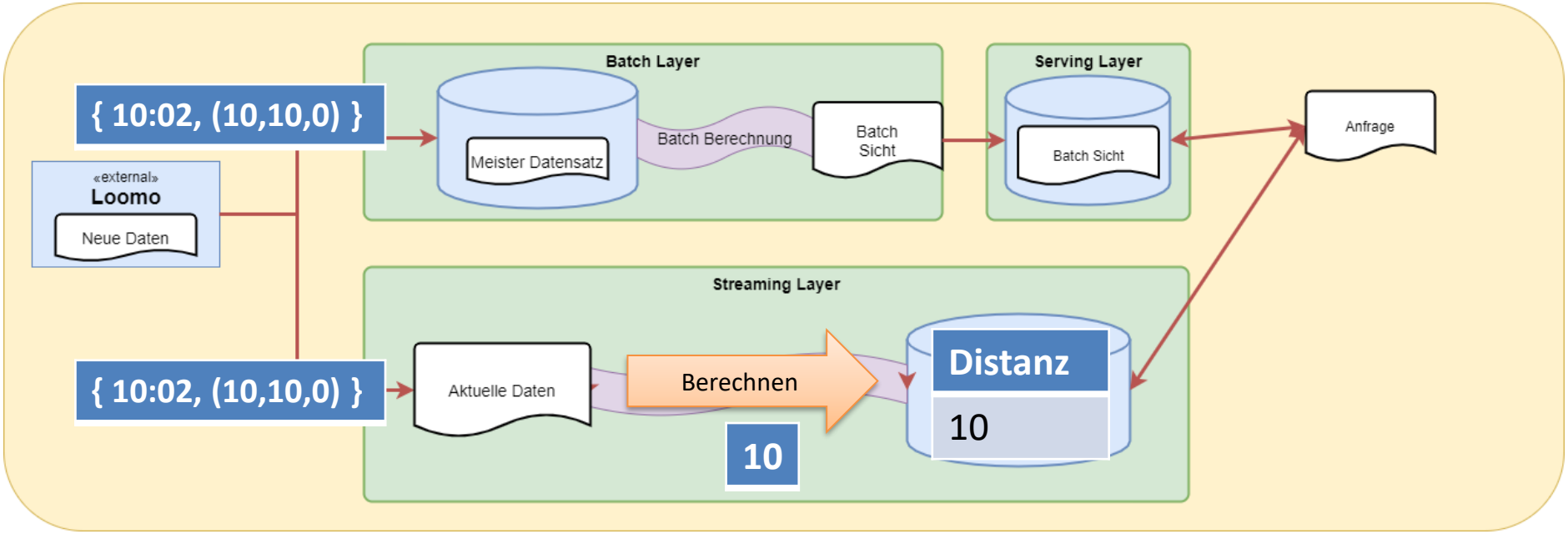
- Streaming Layer
  - Echtzeit Berechnung
  - Schnelle inkrementelle Algorithmen



Lambda Architektur

CaDS
communication and distributed systems

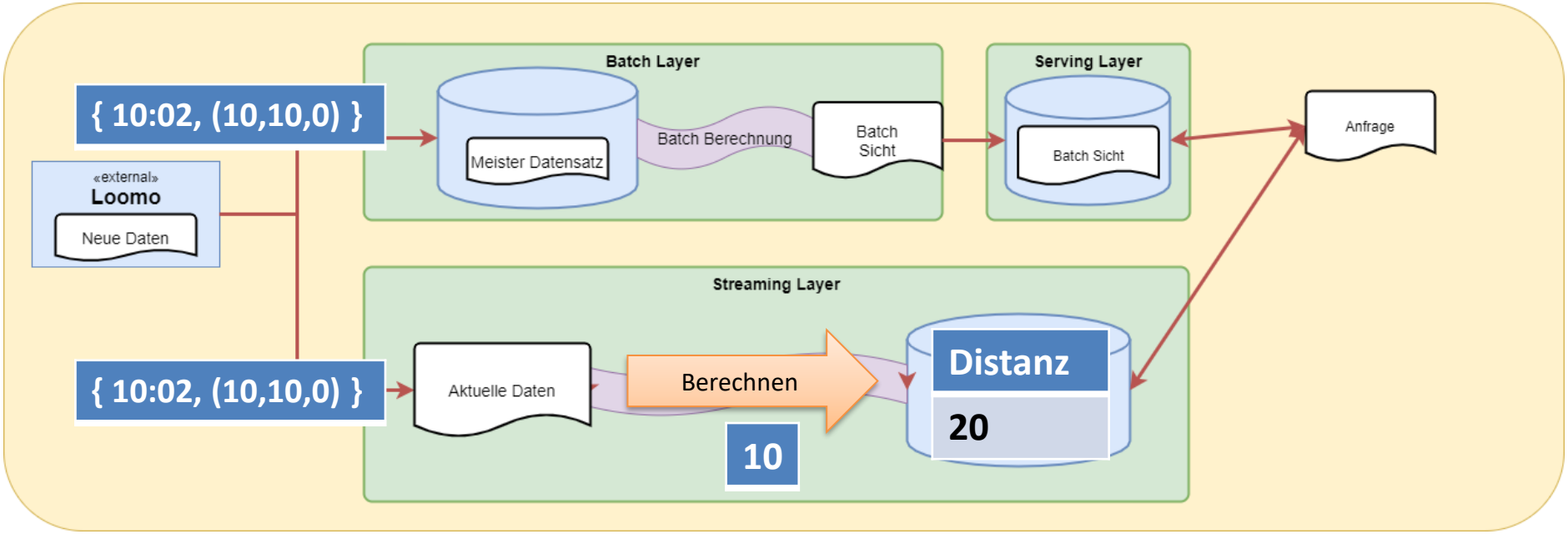
- Streaming Layer
  - Echtzeit Berechnung
  - Schnelle inkrementelle Algorithmen



Lambda Architektur

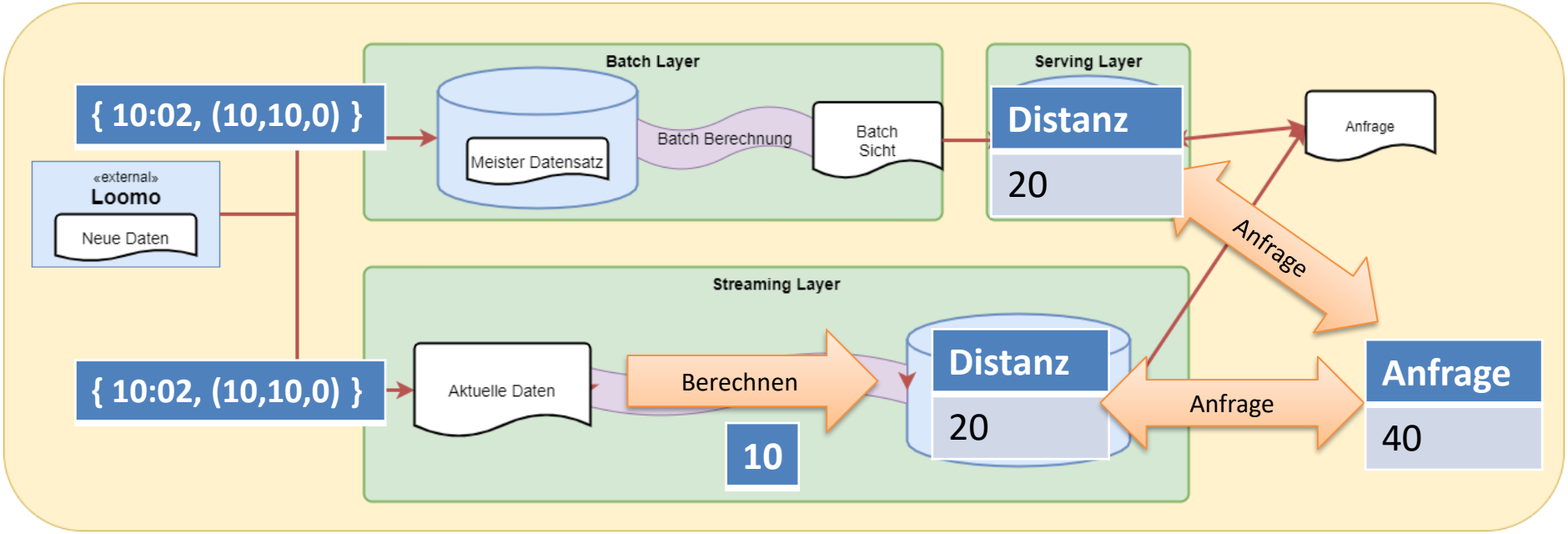
CaDS
communication and distributed systems

- Streaming Layer
  - Echtzeit Berechnung
  - Schnelle inkrementelle Algorithmen



Lambda Architektur

- Streaming Layer
  - Echtzeit Berechnung
  - Schnelle inkrementelle Algorithmen



CaDS
communication and distributed systems

Experiment

- Erhöht sich die Berechnungszeit mit der Anzahl an Nachrichten?

- Skalierbarkeit des Streaming Layer
- Skalierung über die Anzahl Nachrichten

1. Experiment

- Daten Vorbereitung
  - Einlesen der Daten
  - Abflachen der Hierarchie
  - Normalisierung

- Erwartung
  - Konstante Komplexität
  - Berechnungszeit bleibt gleich.

2. Experiment

CaDS
communication and distributed systems

- Spark eigene „Aggregation“
  - Daten WiFi stärke
  - Minimum, Maximum, Durchschnitt

- Erwartung
  - Erstmal Linear Komplex mit Anzahl an Daten
  - Speichert zwischen Ergebnisse
  - Berechnungszeit abhängig von Batch Größe

3. Experiment

CaDS
communication and distributed systems

- Distanz Berechnung „genau“
  - „ROS Odometry“ Daten
  - Sammeln und Sortieren der Daten

- Erwartung
  - Quadratische Komplexität
  - Möglicher Einfluss durch Datenübertragung
  - Berechnungszeit abhängig von Anzahl Daten

4. Experiment

CaDS
communication and distributed systems

- Distanz Berechnung „ungenau“
  - „ROS Odometry“ Daten
  - Buffern des letzten Eintrags
  - Sammeln der Zwischen Ergebnisse

- Erwartung
  - Konstante Komplexität
  - Berechnungszeit bleibt gleich.

CaDS
communication and distributed systems

Fragen?

Quellen

- Nathan Marz, James Warren:
  - Big Data Principles and Best Practices of Scalable Realtime Data Systems