

# Big Data Processing mit Apache Spark

Sascha P. Lorenz, Email: sascha.lorenz@contexagon.com

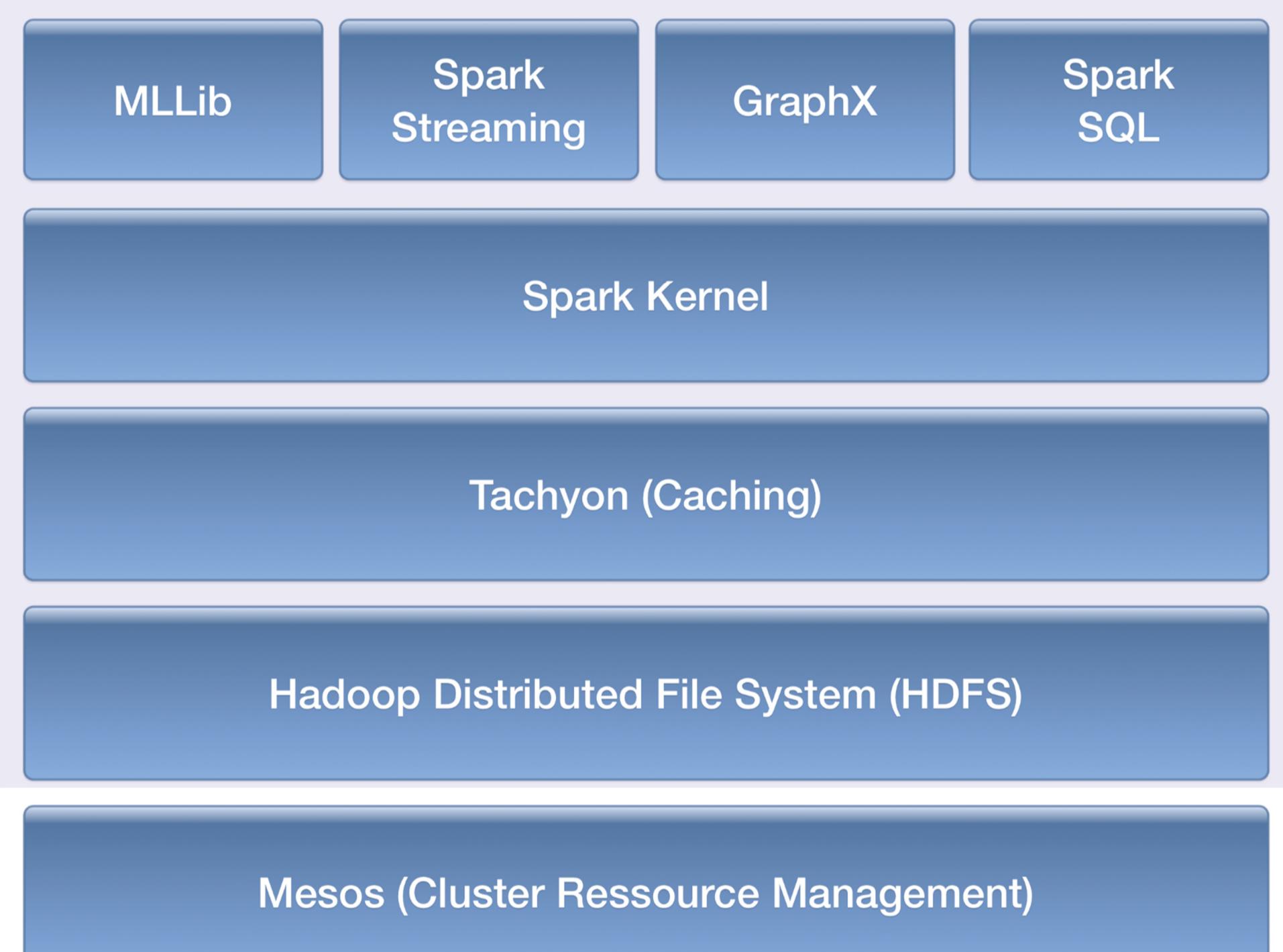
Hochschule Emden-Leer, Fachbereich Technik

## Was ist Apache Spark?

Apache Spark ist ein Cluster-Computing-Framework der University of Berkeley (AMPLabs) zur Analyse und Verarbeitung von großen Datenmengen.

Im Gegensatz zur etablierten Big-Data-Lösung Hadoop, die auf dem zweistufigen MapReduce-Paradigma basiert, nutzt Spark sogenannte In-Memory-Primitives und bietet so eine bis zu hundertfach schnellere Ausführungs geschwindigkeit.

Spark ist Teil eines Architekturstacks, dem BDAS (Berkeley Data Analytics Stack), der Bibliotheken für gängige Big-Data-Anwendungen beinhaltet.



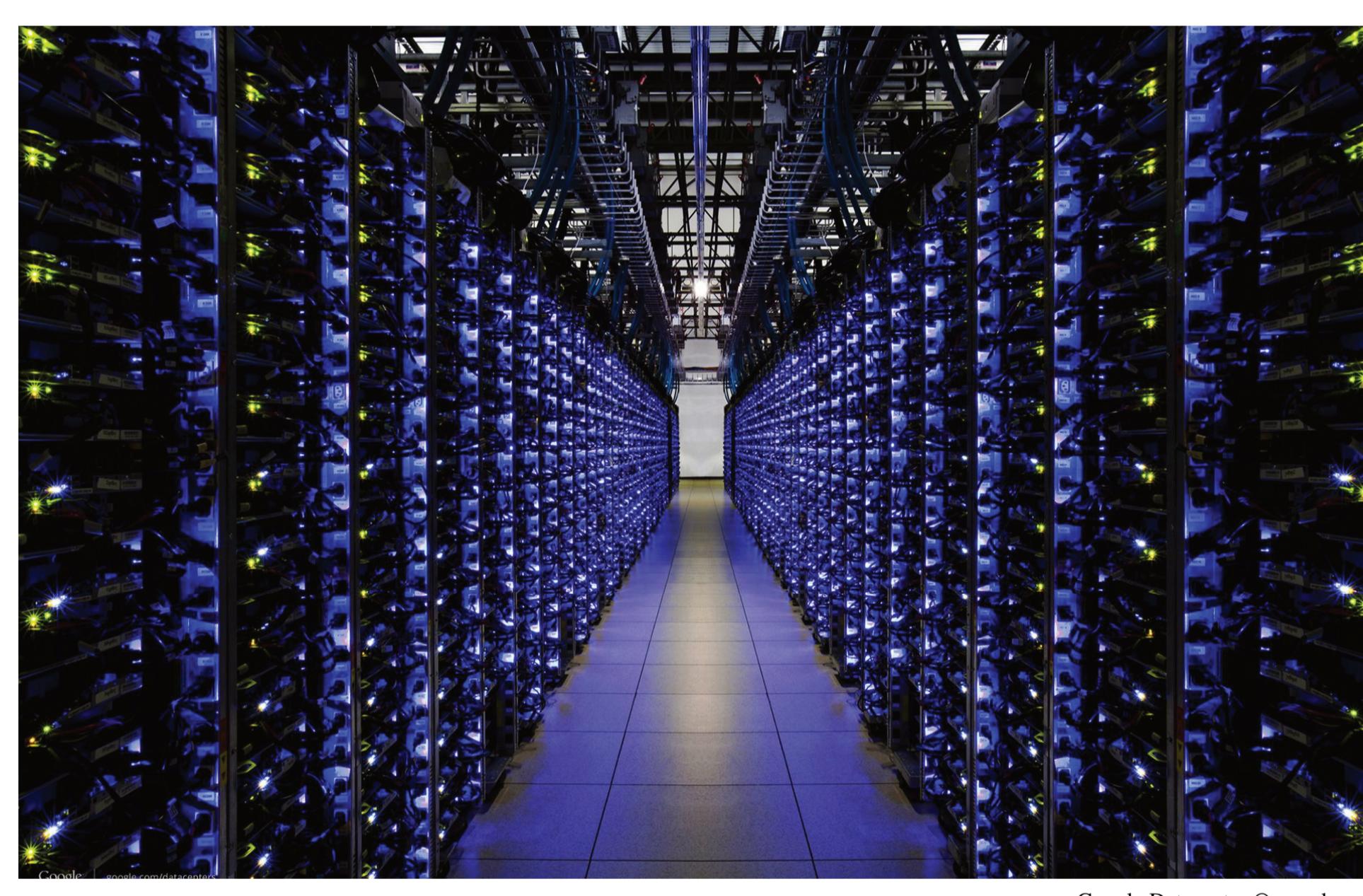
## Ziele der Masterarbeit

- Erarbeitung der Konzepte und des Handlings von Spark
- Konzeption und Aufbau verschiedener Clusterumgebungen
- Definition von Metriken für die einzelnen Anwendungsbibliotheken
- Konzeption und Durchführung von Tests und Messungen unter diversen Konfigurationen
- Implementierung von Prototypen für verschiedene Nutzungsszenarien der jeweiligen Bibliotheken
- Vergleich mit den Alternativimplementierungen Flink (Spark) sowie H2O (MLLibs)

## Definition der Umgebungen

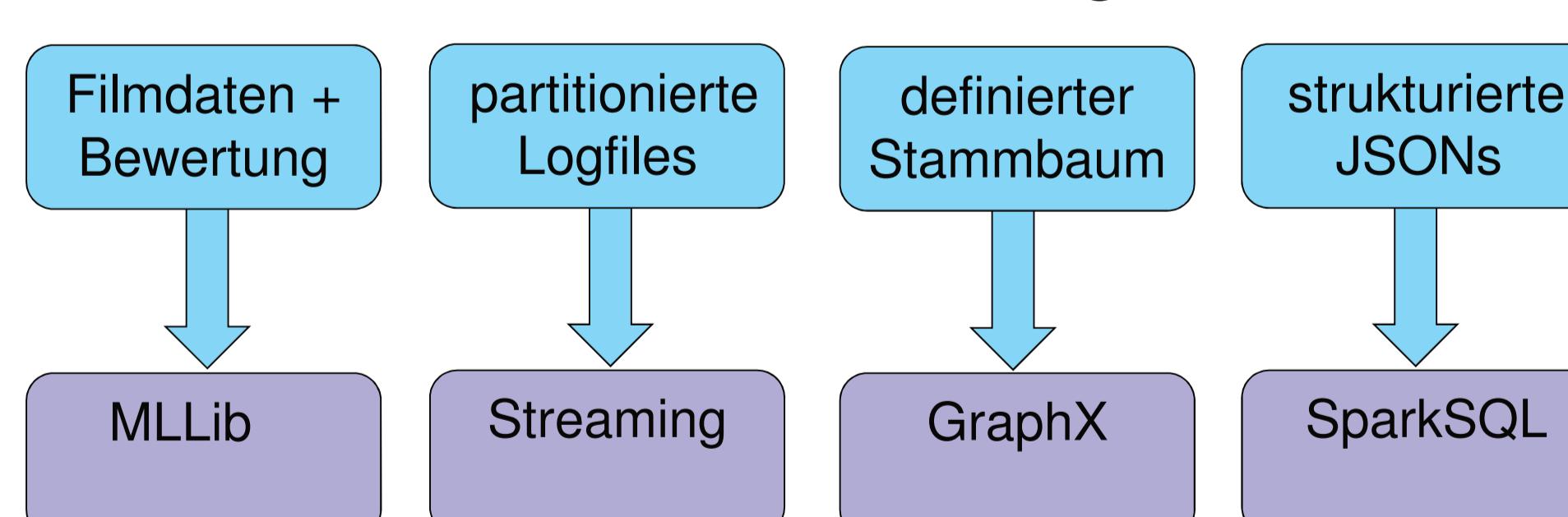
Zunächst wurden verschiedene Umgebungen für die Versuchsanordnungen definiert und realisiert :

- ein lokales Single-Node-Cluster, das Deployment und Debugging über die IDE ermöglicht
- ein lokales virtualisiertes Cluster, bestehend aus virtuellen Linux-Rechnern und Docker-Containern
- ein lokales Hardware-Cluster (Linux)
- ein Remote-Cluster an der Beuth Hochschule Berlin mit über 200 Prozessorkernen und ca. zwei Terabyte RAM über einen Master und drei Worker-Nodes verteilt



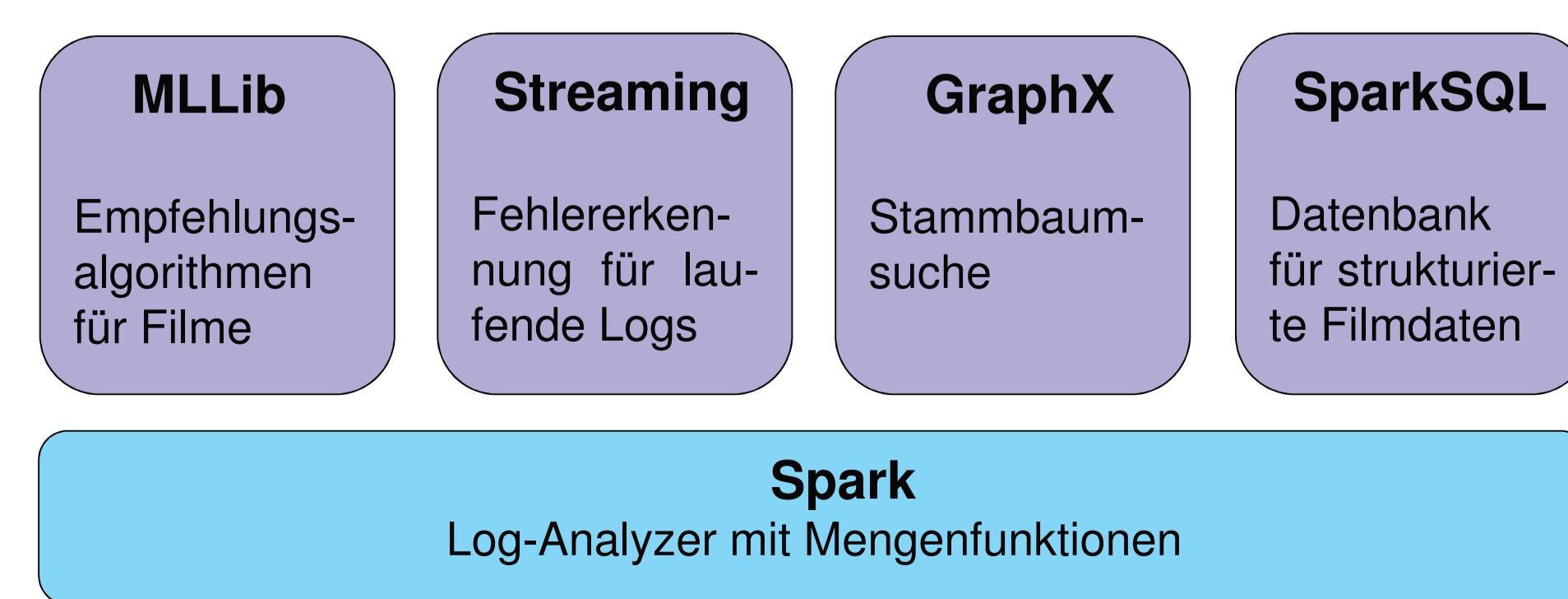
## Definition der Testdaten

Für produktionsnahe Prototypen und realistische Testergebnisse werden entsprechende Testdaten benötigt. Für jede Bibliothek von Spark wurden verschiedene Testdatensätze erzeugt.



## Definition der Prototypen

Für jede Bibliothek und auch für Spark selbst wurden aussagekräftige Prototypen in Scala implementiert.



## Definition der Metriken

Da jede Bibliothek von Spark, inklusive des Spark-Kernels selbst, komplett von einander unabhängige Einsatzbereiche abdeckt, sind auch für all diese Elemente mehr oder weniger unterschiedliche Metriken zu definieren.

Diese Metriken sollen helfen, verschiedene Algorithmen gegeneinander testen zu können, die Leistung verschiedener Hardware-, Software-, und Infrastruktur-Konfigurationen zu messen, sowie den Vergleich mit Alternativimplementierungen zu ermöglichen.

Folgende Metriken wurden im Rahmen dieser Arbeit implementiert:

### Für alle Bibliotheken von Spark:

- Ausführungszeit bis Ergebnis vorliegt
- CPU-Auslastung, Anzahl Kerne
- Netzwerkauslastung
- Datendurchsatz (I/O)

### Zusätzlich für Spark-Kernel:

- Logging der Map-Zeit
- Logging der Reduce-Zeit

### Zusätzlich für MLLib:

- Logging der Zeit für Training / Lernen
- Precision/Recall (Für Vorhersage)
- Area under the curves (Leistung des Modells)

### Zusätzlich für Spark Streaming:

- Logging des Nachrichtendurchsatzes

## Bisherige Ergebnisse

Im Vergleich zu analogen Implementierungen der Tests auf Hadoop zeigt Spark überall deutliche Performancevorteile. Besonders gravierend sind diese, wenn die Daten in den Hauptspeicher passen. Bei iterativer Verarbeitung von Datensätzen auf Spark wird die Leistung merklich schlechter, da Daten persistiert werden müssen. Die Tests gegen H2O und Flink stehen noch aus.

### Literaturverzeichnis:

- Agueeswaran, V.: *Big Data Analytics Beyond Hadoop*. Pearson 2004  
 Cleve, J., Lämmel, U.: *Data Mining*. De Gruyter/Oldenbourg 1.Auflage 2014  
 Sathi, Dr. A.: *Big Data Analytics. Disruptive Technologies*. MC Press 2012  
 National Research C.: *Frontiers in Massive Data Analysis*. N.A.Press 2013  
 Bishop C.M.: *Pattern Recognition and Machine Learning*, Springer 2006