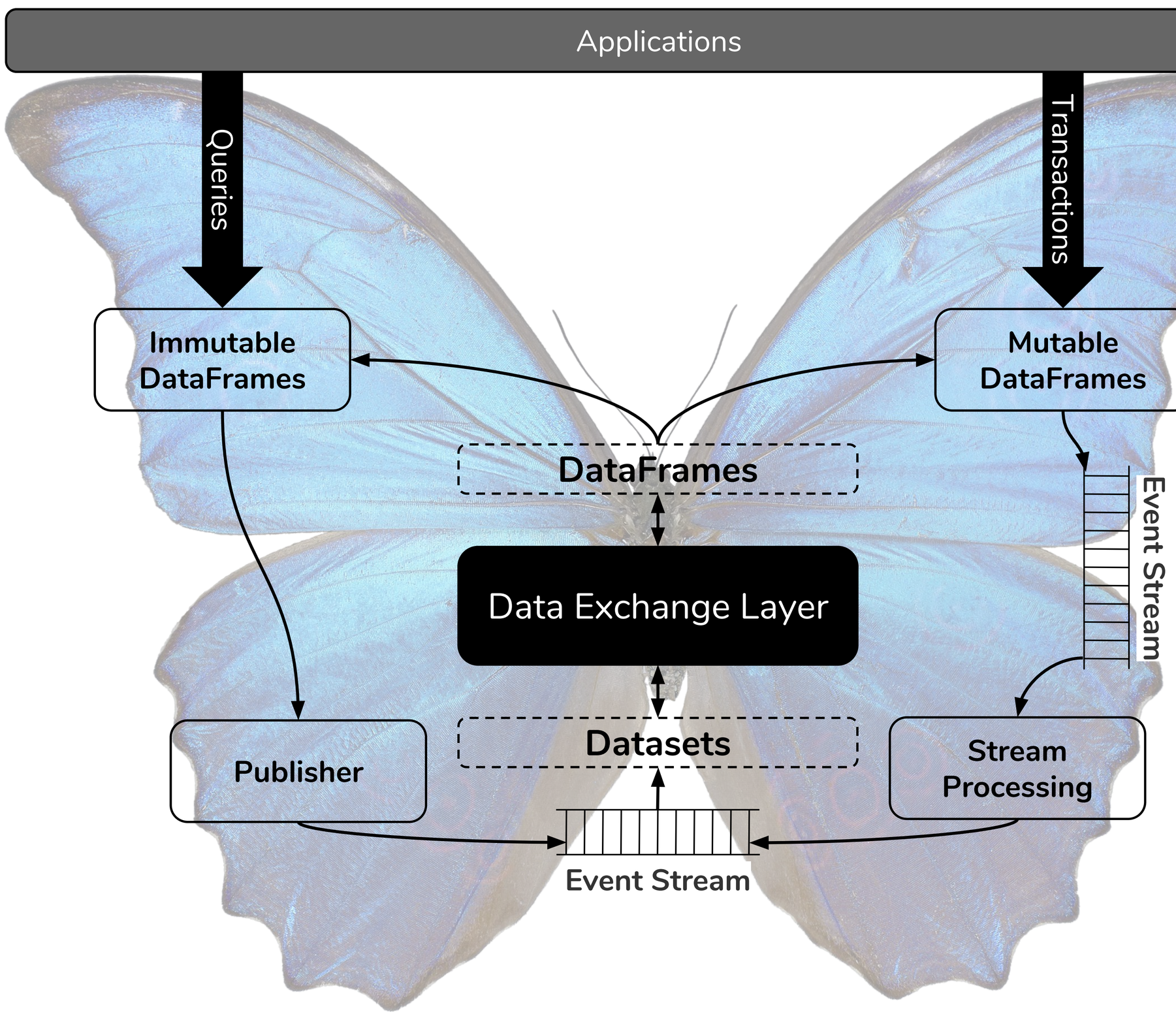
Butterfly Architecture
using Monarch Active Data Store

The Butterfly Architecture proposed by Milind Bhandarkar, CEO of Ampool, aims to provide a **unified data store** that supports all analytical workloads within a single environment, unlike **Lambda**, as used by Neofonie, or **Kappa** architectures, which separate data between batch, speed and serving layer. Bhandarkar argues that both Lambda and Kappa need to load, transfer and save data between layers, creating overhead. Therefore the butterfly architecture is unifying various data processing tasks on a single platform. Following abstractions are made:

Datasets are partitioned collections, possibly distributed over multiple storage backends.

DataFrames are **structured datasets**, similar to tables in RDMS or NoSQL. Immutable dataframes are suited for analytical workloads, while mutable dataframes are suited for transactional CRUD workloads.

Event Streams are **unbounded dataframes**, such as time series or sequences.



14 Oct. 2014, Pivotal
1st Butterfly architecture: In-memory Data Exchange Platform using GemFire, Tachyon and Spark

28 Apr. 2015, Pivotal
GemFire open-sourced as Apache Geode

22 Nov.2016, Ampool
Blog series about Emerging Data Architectures

23 Dec.2016, Ampool
Release of Ampool Active Data Store

5 Dec.2017, Ampool
ADS open-sourced as Monarch

Monarch Active Data Store is “a robust, in-memory computing platform designed to enable real-time, data-driven applications with faster access to active data, resulting in timely insights.” It is open-source, based on Apache Geode (formerly GemFire) and developed by Ampool. Monarch was built to be the data exchange layer inside the butterfly architecture and is the replacement for Tachyon (now Alluxio) in the initial butterfly architecture from Pivotal.
Active data is held in memory, eliminating the step of loading and saving data for different processing steps. This is in part motivated by the great performance/cost ratio of DRAM and NVRAM, also elaborated by Matthias Uflacker. Since most data loses business value over time, historical data can be saved to cold storage like HDFS for later batch processing.

Nils Strelow
Master Student

Data Engineering in der Praxis WS 17/18
Hasso Plattner Institute, Potsdam, Germany

E-Mail: nils.strelow@student.hpi.de