

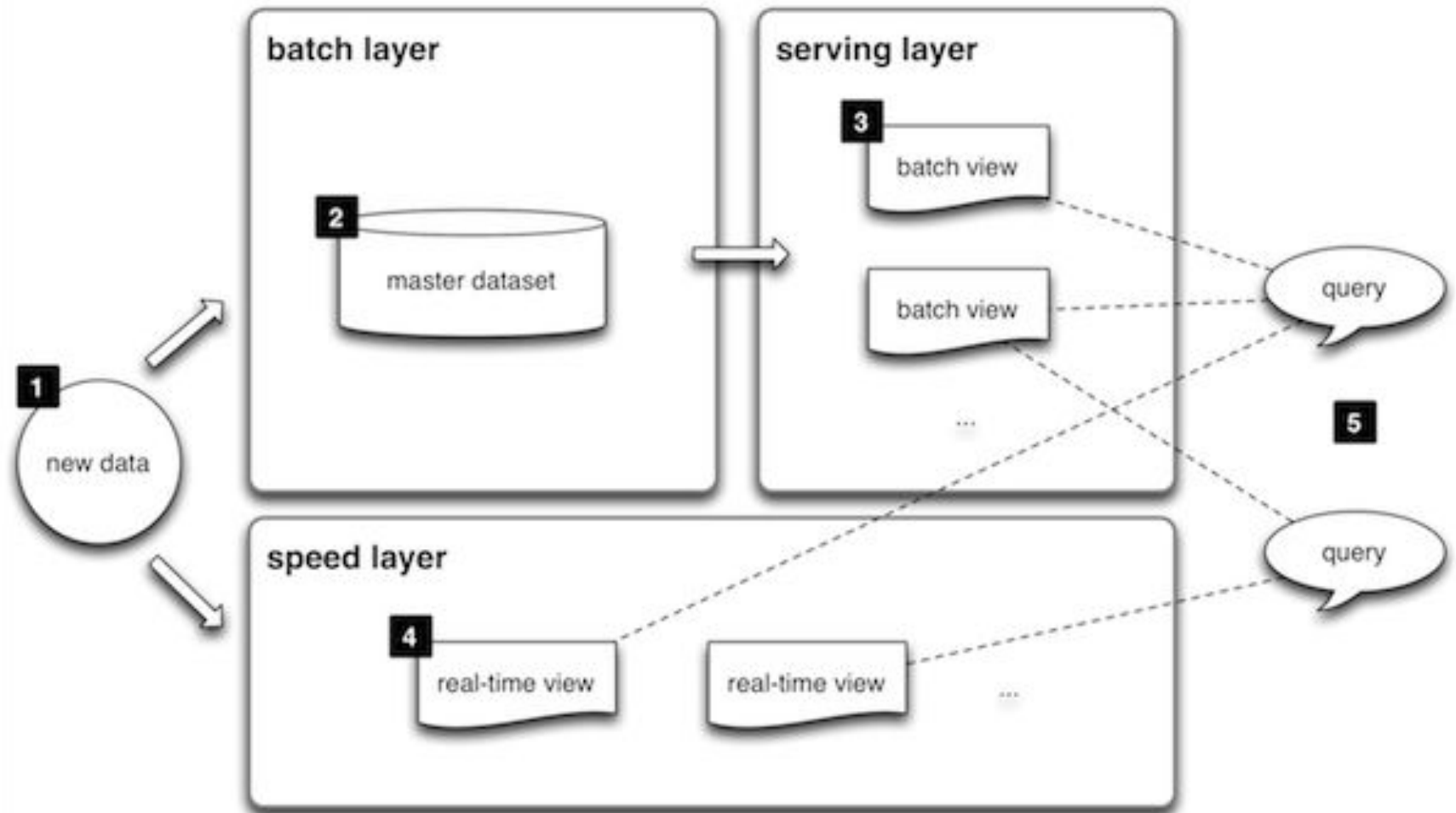
Для доступа к «большим и быстрым» данным была разработана **лямбда-архитектура**, которая представляет собой общий подход, направленный на применение произвольной функции к произвольному набору данных, при этом обеспечивая минимальный период ожидания возвращения функцией искомого значения. Она состоит из трех уровней: пакетного (batch layer), сервисного (serving layer), уровня ускорения (speed layer).

Пакетный уровень – архив сырых исторических данных. Чаще всего, это «озеро данных» на базе Hadoop, хотя встречается и в форме OLAP-хранилища данных, старые данные остаются *неизменными* – происходит добавление новых.

Сервисный уровень индексирует пакеты и обрабатывает результаты вычислений, происходящих на пакетном уровне.

Уровень **ускорения** отвечает за обработку данных, поступающих в систему в реальном времени. Представляет собой совокупность складов данных, в которых те находятся в режиме очереди, в потоковом или в рабочем режиме.

На этом уровне компенсируется разница в актуальности данных, а в отдельные представления реального времени добавляется информация с коротким жизненным циклом (чтобы исключить дублирование данных). Эти представления параллельно с сервисным уровнем обрабатывают свои запросы.



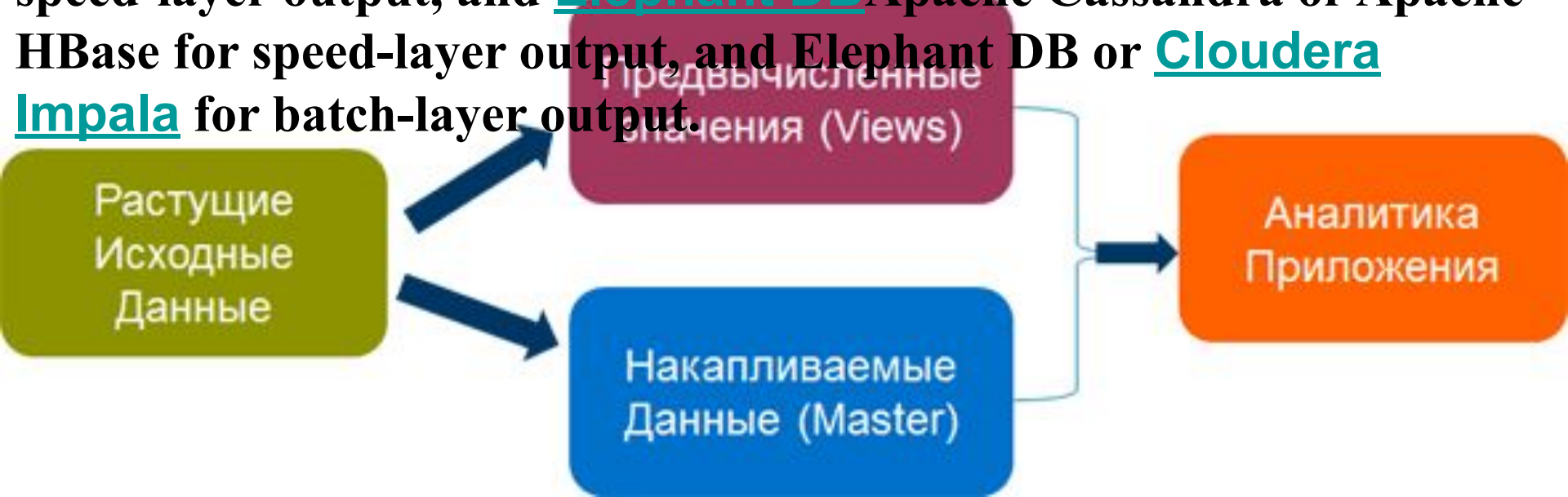
В основе Лямбда-архитектуры лежит несколько принципов: система не восприимчива к единичной потере данных и/или повреждению данных (fault-tolerance); неизменность данных – хранение данных в исходном неизменяемом виде; перевычисление – есть возможность всегда провести вычисления на исходных данных.

Вся информация поступает в единое хранилище данных (*мастер*), в котором может быть и другая статичная информация, и одновременно дублируется на уровне агрегации краткосрочного периода. любой запрос имеет доступ к мастеру для получения полного ответа на основе полных данных, и коррекции с учетом самой последней информации, агрегированной за краткосрочный период, поскольку мастер, очевидно, имеет определенную инерцию из-за скорости обработки большого объема информации.

Уровень пакет использует [Apache Hadoop](#)

Уровень скорость использует Stream-processing technologies typically [Apache Storm](#) использует Stream-processing technologies typically Apache Storm, [SQLstream](#) использует Stream-processing technologies typically Apache Storm, SQLstream and [Apache Spark](#).
Результат обычно сохраняется на быстрых NoSQL БД.

Уровень сервис использует [Apache Cassandra](#) Apache Cassandra or [Apache HBase](#) Apache Cassandra or Apache HBase for speed-layer output, and [Elephant DB](#) Apache Cassandra or Apache HBase for speed-layer output, and Elephant DB or [Cloudera Impala](#) for batch-layer output.



Новые данные поступают на оба уровня: Пакетный и Ускорения (А). Мастер (В) представляет собой хранилище неизменяемой сырой информация, где исходные данные только добавляются. Пакетный уровень (С) постоянно перевычисляет функции заново в Пакеты. Сервисный Уровень (D) индексирует Пакеты, и здесь результаты обычно отстают по времени из-за скорости прохождения мастера и индексирования. Уровень Ускорения (Е) компенсирует разницу в актуальности данных, и постоянно добавляет данные в представления реального-времени с коротким жизненным циклом (ведь нам не нужно дублирование в хранении данных, т.к. они накапливаются на Мастере). И, наконец, запросы обрабатывают пакеты и представления реального времени (F).

На данный момент существуют различные вариации технологических компонент для Лямбда-архитектуры.

Например, проект Lambdoor, который объединяет компоненты экосистемы Hadoop для всех трех слоев Лямбда-архитектуры: кластер Hadoop используется для пакетного уровня – HDFS для мастера и MapReduce для быстрого перевычисления запроса на исходных данных; для сервисного уровня используются Cloudera Impala для пакетов и Apache HBase для создания представлений реального времени; и технология Storm для уровня ускорения. Также Cloudera Impala решает вопрос агрегации результатов из уровней для ответа на запрос.

Проект развивается с 2013 г.

