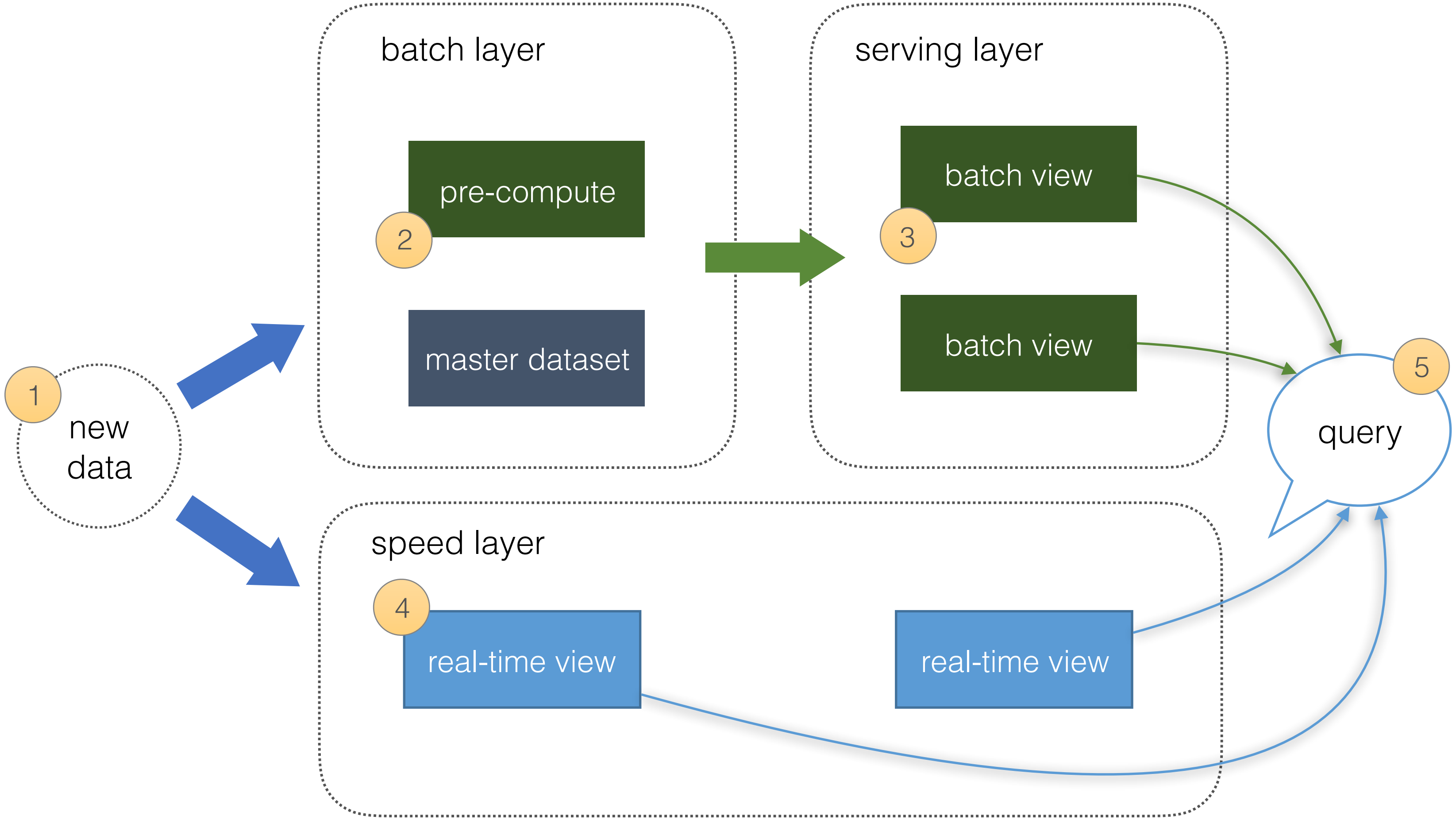
Lambda Architecture

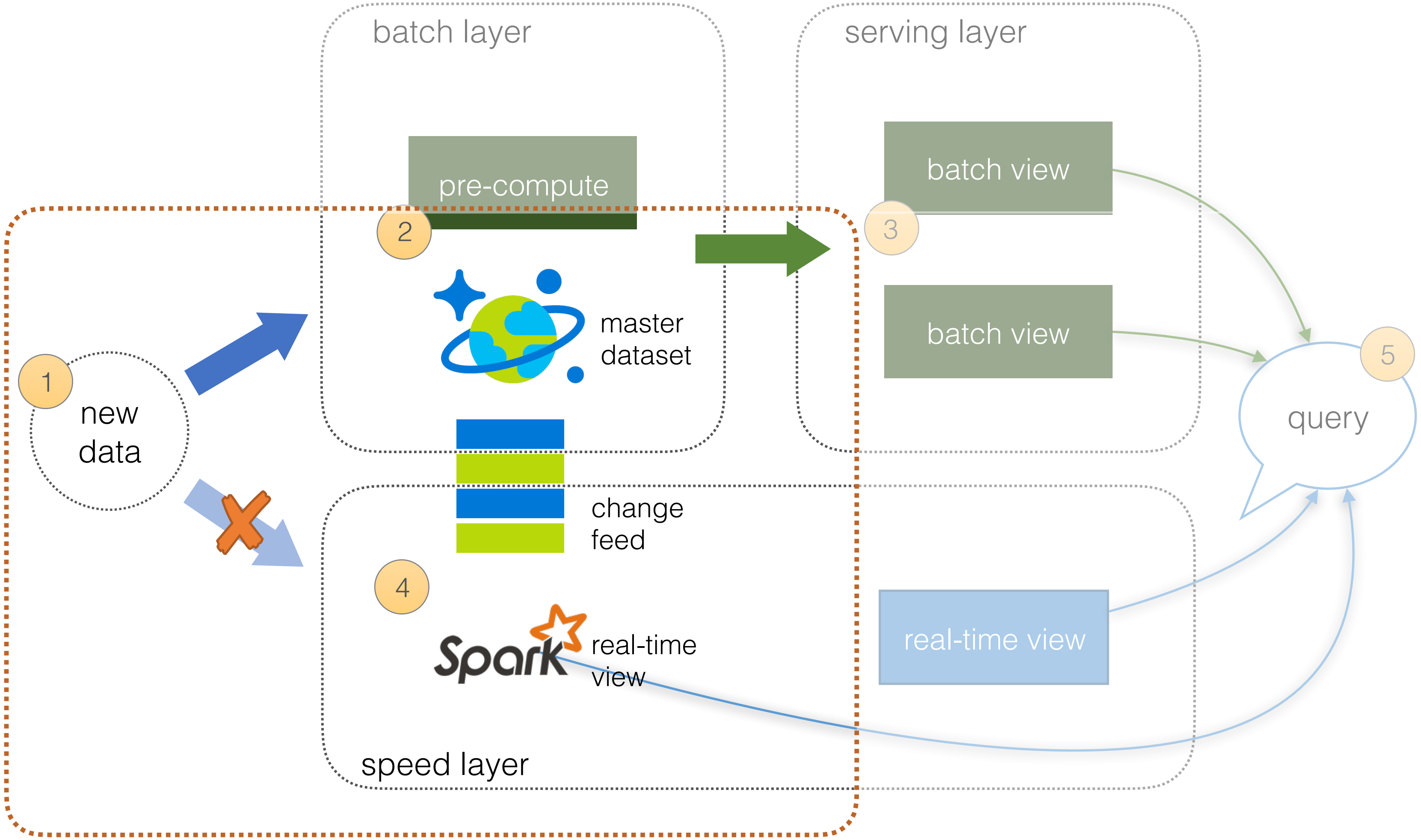


The components of a Lambda Architecture

1. All **data** pushed into *both* batch and speed layer for processing
2. The **batch** layer has a master dataset (immutable, append-only set of raw data) and pre-compute the batch views
3. The **serving** layer has batch views so data for fast queries.
4. The **speed** layer compensates for processing time (to serving layer) and deals with recent data only.
5. All queries can be answered by merging results from batch views and real-time views.

Source: http://lambda-architecture.net/

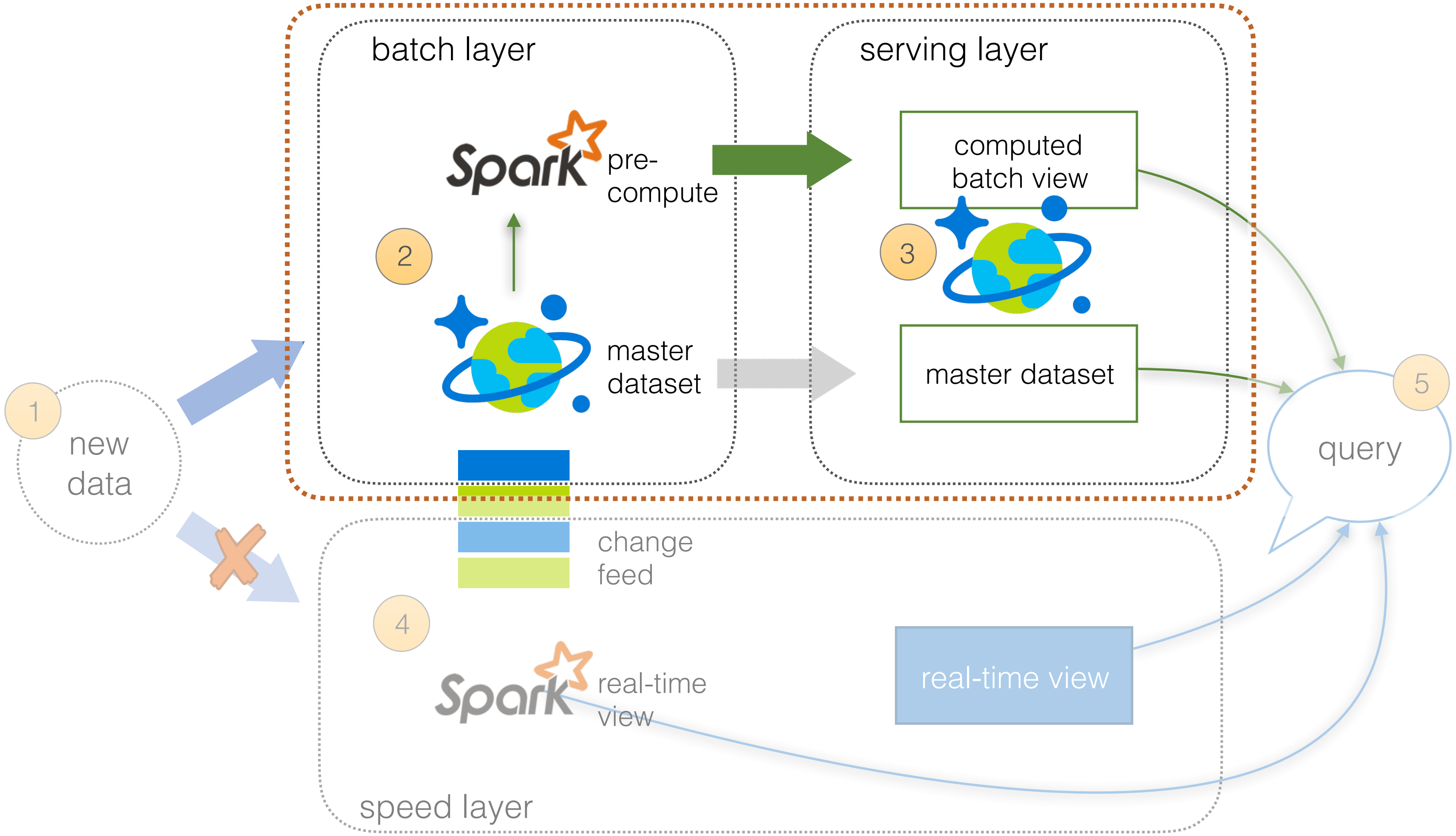
Lambda Architecture: Cosmos DB Change Feed



The components of a Lambda Architecture

1. All **data** pushed into *only* Cosmos DB (avoid multi-cast issues)
2. The **batch** layer has a master dataset (immutable, append-only set of raw data) stored in Cosmos DB (pre-compute discussed next slide).
3. The **serving** layer will be discussed next slide.
4. The **speed** layer utilizes HDI Spark to utilize the Cosmos DB change feed. This allows you to persist your data, query it, and process it.
5. Raw data queries delivered from Cosmos DB (batch layer) while real-time queries can be from Cosmos DB change feed and/or HDI Spark (speed layer) via (structured) streaming.

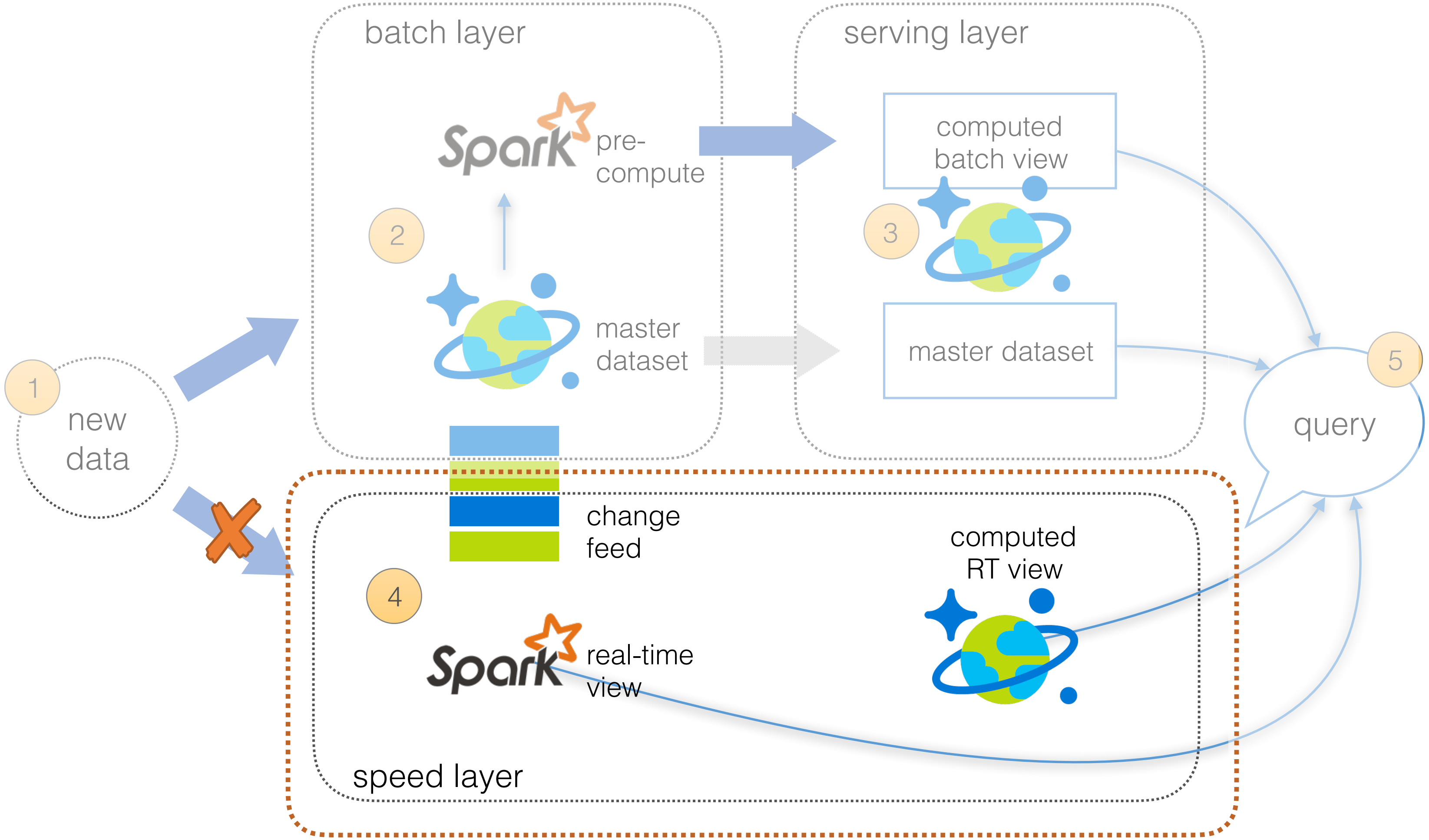
Lambda Architecture: Batch and Serving Layers



The components of a Lambda Architecture

1. All **data** pushed into *only* Cosmos DB (avoid multi-cast issues)
2. The **batch** layer has a master dataset (immutable, append-only set of raw data) stored in Cosmos DB. Using HDI Spark, you can pre-compute your aggregations to be stored in your computed batch views.
3. The **serving** layer is Cosmos DB with collections for master dataset and computed batch view .
4. The **speed** layer will be discussed next slide.
5. Raw data queries delivered from Cosmos DB (batch layer) while real-time queries can be from Cosmos DB change feed and/or HDI Spark (speed layer) via (structured) streaming.

Lambda Architecture: Speed Layer

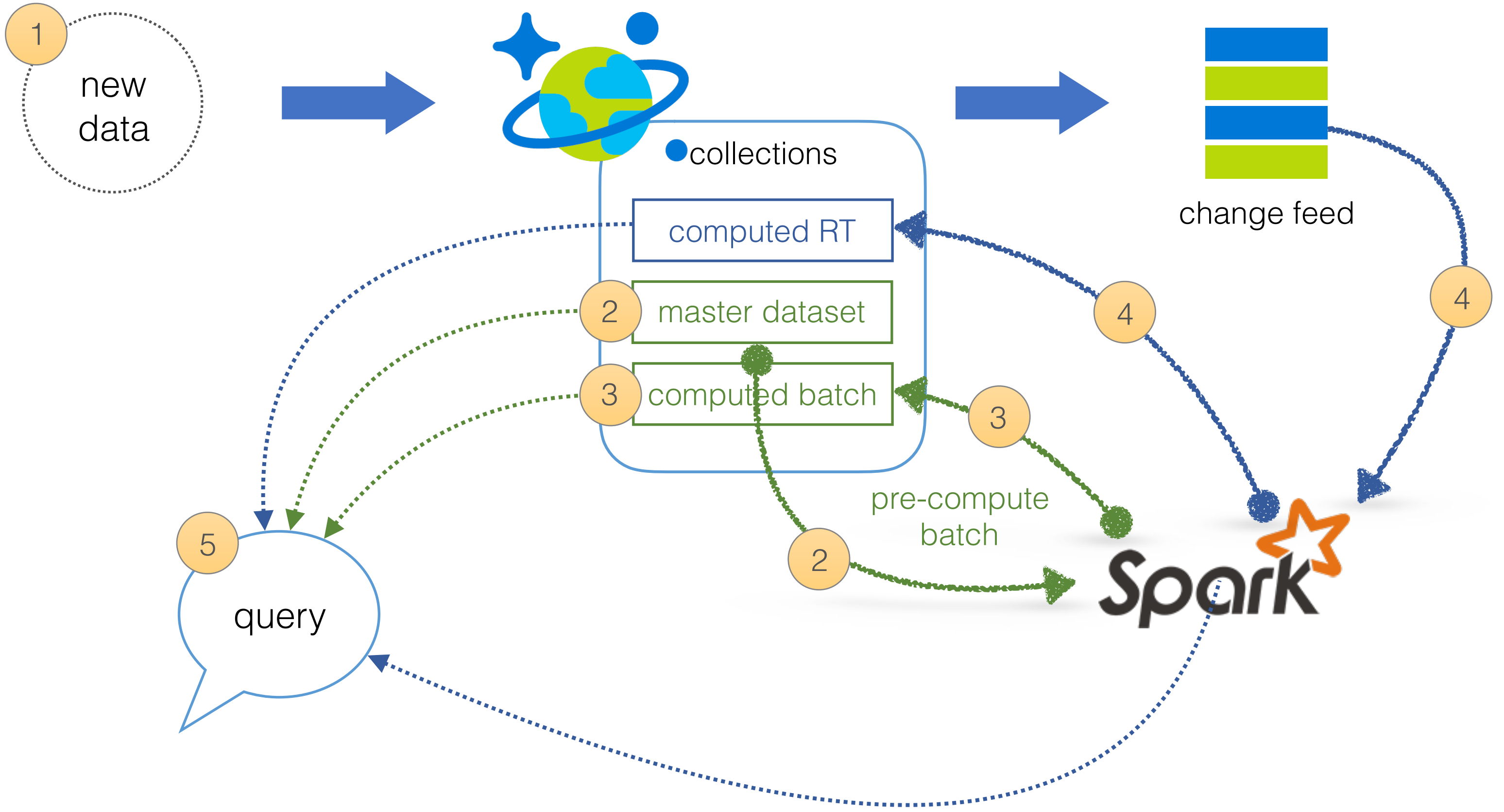


The components of a Lambda Architecture

1. All **data** pushed into *only* Cosmos DB (avoid multi-cast issues)
2. The **batch** layer has a master dataset stored in Cosmos DB. Using HDI Spark, pre-compute aggregations to be stored in your computed batch views.
3. The **serving** layer is Cosmos DB with collections for master dataset and computed batch view .
4. The **speed** via Spark Streaming can provide a real-time data frame as well as store a fast computed view.
5. Raw data queries delivered from Cosmos DB (batch layer) while real-time queries can be from Cosmos DB change feed and/or HDI Spark (speed layer) via (structured) streaming.

Lambda Architecture: Re-architected

Cosmos DB + HDI Apache Spark



The components of a Lambda Architecture

1. All **data** pushed into Cosmos DB layer for processing
2. The **batch** layer has a master dataset (immutable, append-only set of raw data) and pre-compute the batch views
3. The **serving** layer has batch views so data for fast queries.
4. The **speed** layer compensates for processing time (to serving layer) and deals with recent data only.
5. All queries can be answered by merging results from batch views and real-time views.