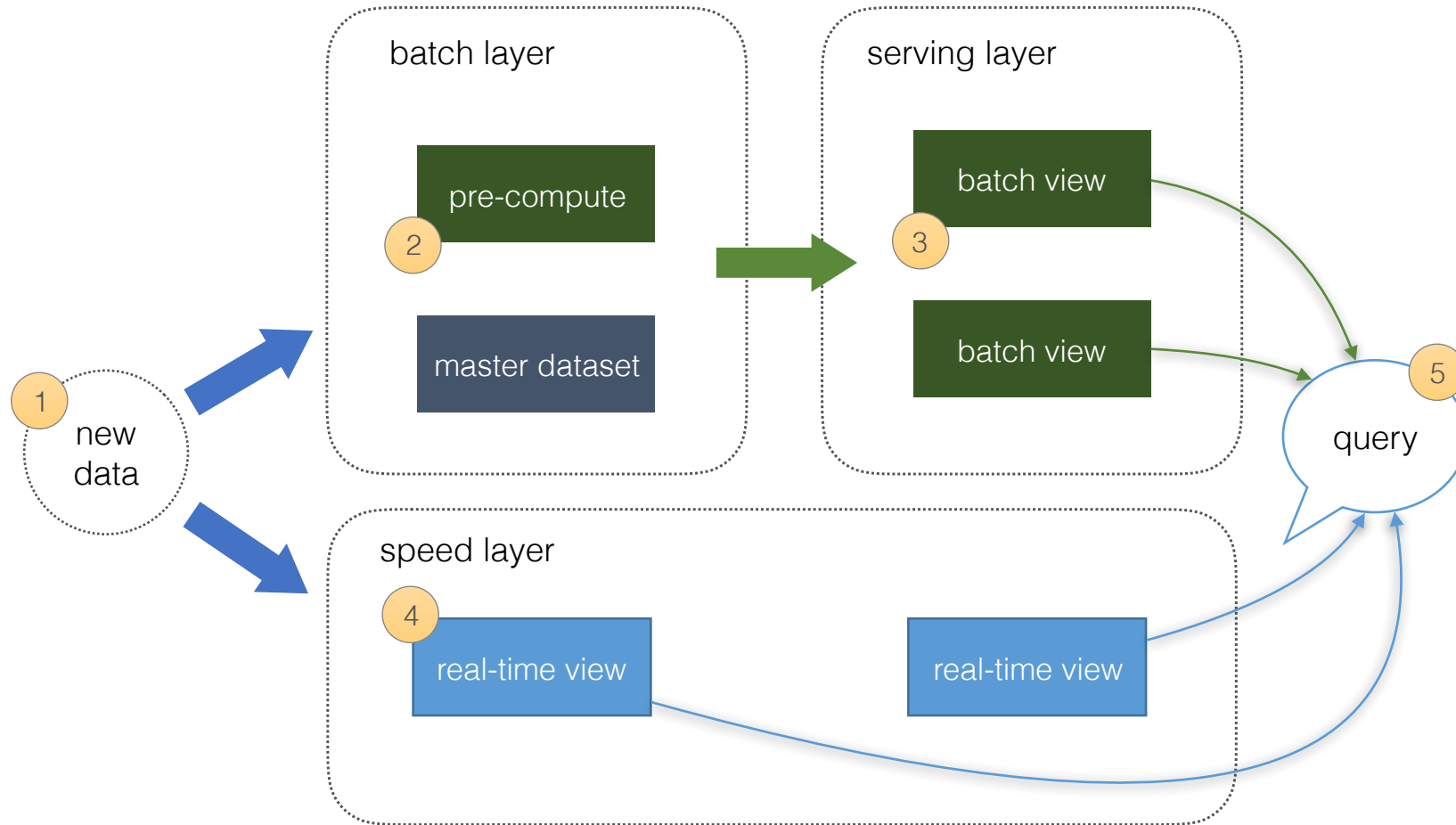


Lambda Architecture

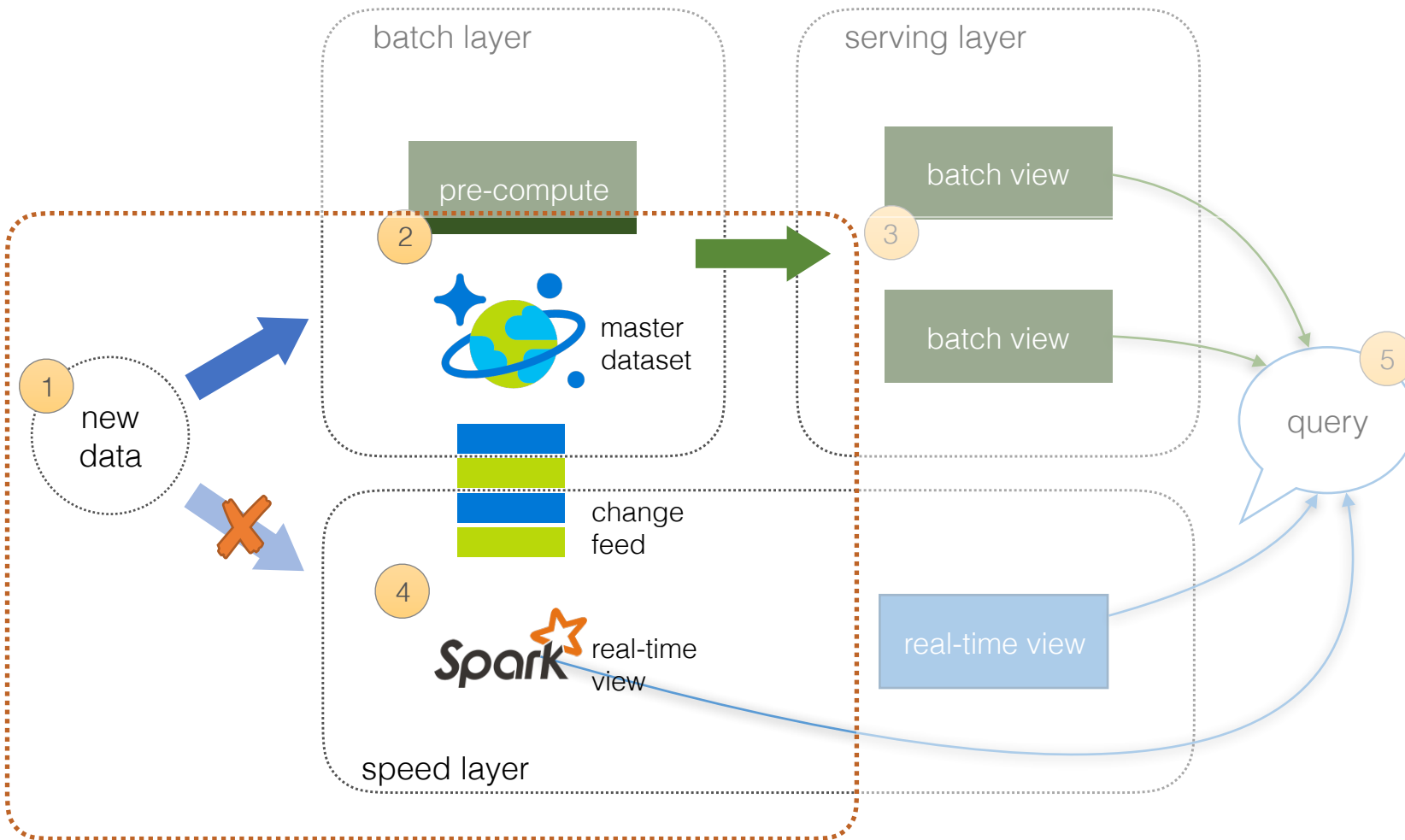


The components of a Lambda Architecture

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.

The diagram illustrates a hybrid data architecture with three main layers: batch, serving, and speed.

- Batch Layer:** Contains a **master dataset** (represented by a globe icon) and a **pre-compute** step (represented by a Spark logo and a star icon). A green arrow indicates data flow from the master dataset to the pre-compute step.
- Serving Layer:** Contains a **computed batch view** (represented by a globe icon) and a **master dataset** (represented by a globe icon). A green arrow indicates data flow from the pre-compute step to the computed batch view.
- Speed Layer:** Contains a **real-time view** (represented by a Spark logo and a star icon) and a **change feed** (represented by a stack of colored bars). A blue arrow indicates data flow from the change feed to the real-time view.

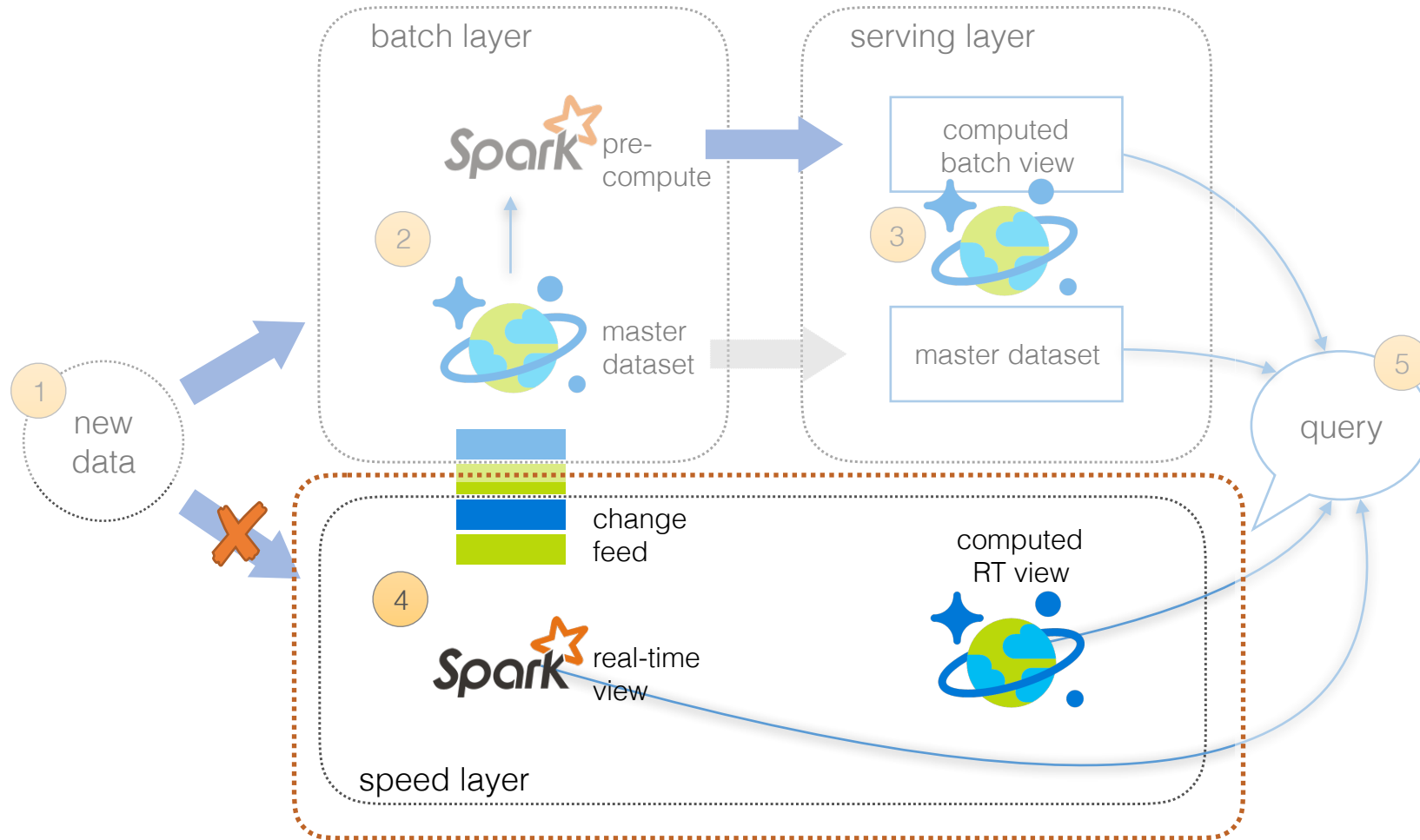
Numbered steps (1-5) indicate the data flow:

- 1 new data:** New data enters the system.
- 2 master dataset:** Data is loaded into the master dataset in the batch layer.
- 3 computed batch view:** Data is processed and stored in the computed batch view in the serving layer.
- 4 real-time view:** Data is processed and stored in the real-time view in the speed layer.
- 5 query:** A query is executed, pulling data from the computed batch view and the real-time view.

A red 'X' over a blue arrow indicates that new data is not directly loaded into the real-time view.

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.

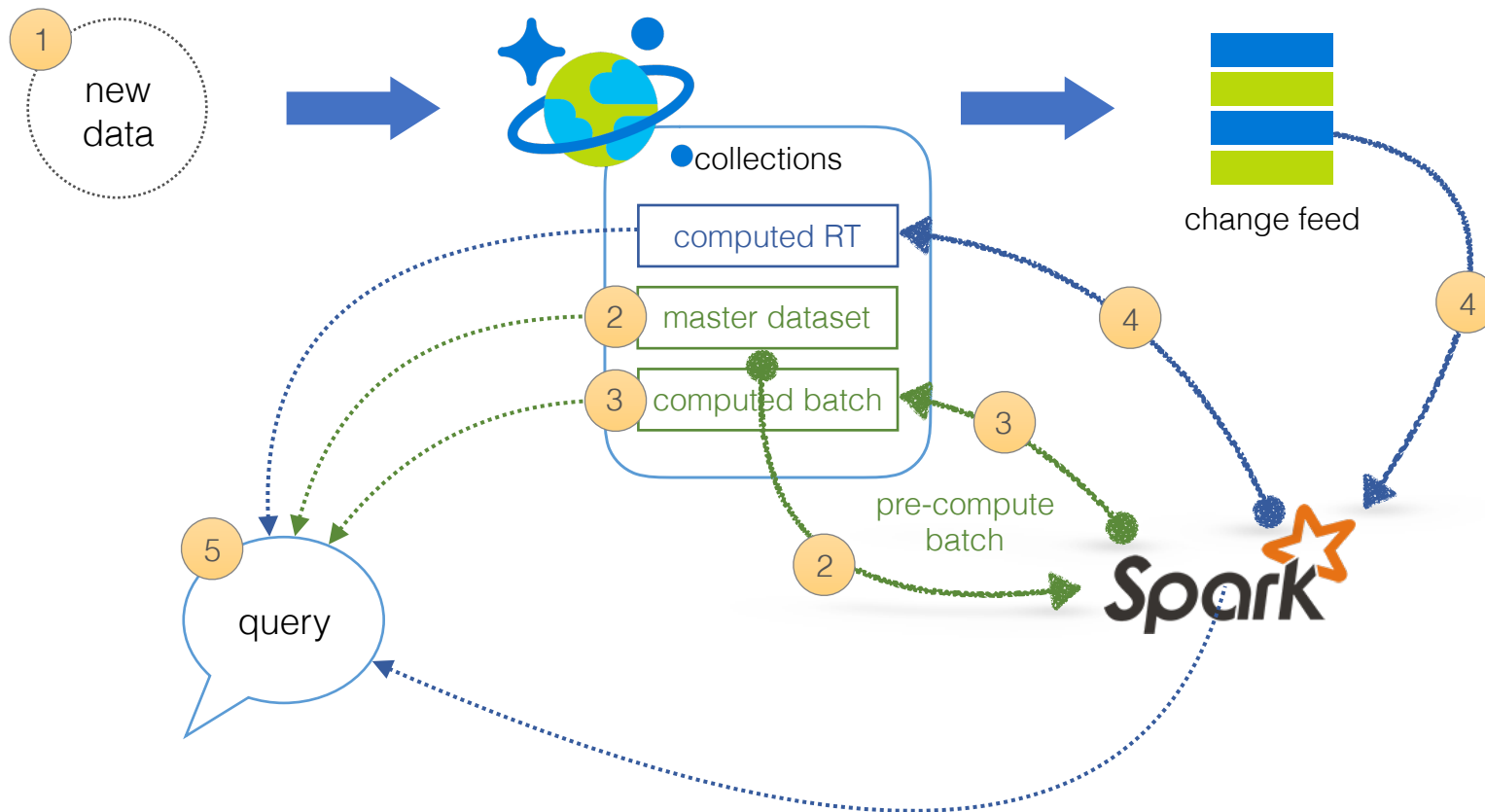
Lambda Architecture: Speed Layer



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.