1. Raw Zone/Streaming Platform
2. Delta Lake (Medallion Architecture)
   1. Reasons for a globally non-redundant Bronze Zone
      1. Ingestion is slow and expensive to run, so doing it only once for each “row” in the Raw Zone/Streaming Platform should make sense.
      2. It’s also slow and expensive to rerun, so you want to avoid complex transformations to minimize the risk of breaking.
      3. You also want to avoid filtering the data as the changes in business needs also means rerunning pipelines.
      4. Because you avoided complex transformations the data here is objective, so doing ingestion once for each row globally does make sense.
   2. Reasons for a (Data Mart) locally non-redundant Silver Zone and Gold Zone
      1. Filtering Bronze and deduplication is slow and expensive to rerun, so doing it only once for each “row” in the Bronze Zone should make sense.
      2. Filtering Bronze and deduplication is slow and expensive to rerun, so you want to avoid complex transformations to minimize the risk of breaking.
      3. However, filtering and deduplication themselves are subjective, so there’s no globally right way to do it, so you must introduce redundance (Data Mart local uniqueness).
      4. Perfect historical deduplication will also make use of the MERGE statement which requires an Event Date Partitioned Silver Zone to be scalable. Because of this a Gold Zone is needed for query optimized partitioning.
      5. You also want to minimize the risk of breaking, so you put all the other transformations into the Gold Zone, which is much cheaper to rerun.
3. Semantic Layer (Databricks SQL, Photon and Hive UDFs)
   1. Self-service data modeling
   2. Terrible join performance otherwise
   3. Somewhat terrible aggregation performance otherwise
   4. Unscalable batch pipelines or inefficient client-side ETL, without it streaming analytics & handling late arriving data is hard or impossible