You could also look at Apache Flink which is more general. (You could even do a comparison of Kafka vs Flink). So part of the project would be outlining the new tools / frameworks, and then give some examples of it in action.

Topic: comparison flink vs kafak, focus on stream computing

Apache Flink and Apache Kafka are both powerful tools for real-time data processing, but they have different strengths and use cases [12].

Outline:

1. **What is Apache Flink?**
2. **What is Kafka,**
3. **Why is kafka fast**
4. What is Kafka streams api?
5. **Distributed Coordination and Fault Tolerance**
6. **Chart comparision from** <https://sourceforge.net/software/compare/Apache-Kafka-vs-Apache-NiFi-vs-Flink/>
7. **Using kafka with flink**
8. **Which one should user choose?**
9. **Case study on the pipeline**

What is Apache Flin:

\*\*Apache Kafka\*\*¹²:

- Primarily a messaging system for real-time data streams.

- Provides a high-throughput, fault-tolerant mechanism for publishing and subscribing to streams of records in real time.

- Has slightly higher latency due to the overhead of the messaging system and replication².

- Specifically designed to handle data loss and replication, making it an excellent choice for mission-critical applications that require high reliability².

- Has a simple and easy-to-use API that allows developers to quickly get started with real-time data streaming².

- A partially-stateful framework and it is ideal for applications that require low latency or to process large amounts of data⁴.

In terms of distributed coordination, Flink has a dedicated master node for coordination, while the Streams API in Kafka relies on the Kafka broker for distributed coordination and fault tolerance¹. Both tools are designed to be highly fault-tolerant and reliable². However, they are designed to solve orthogonal problems and have very different sweet spots and placement in the data infrastructure stack¹.

Source: Conversation with Bing, 2023-11-01