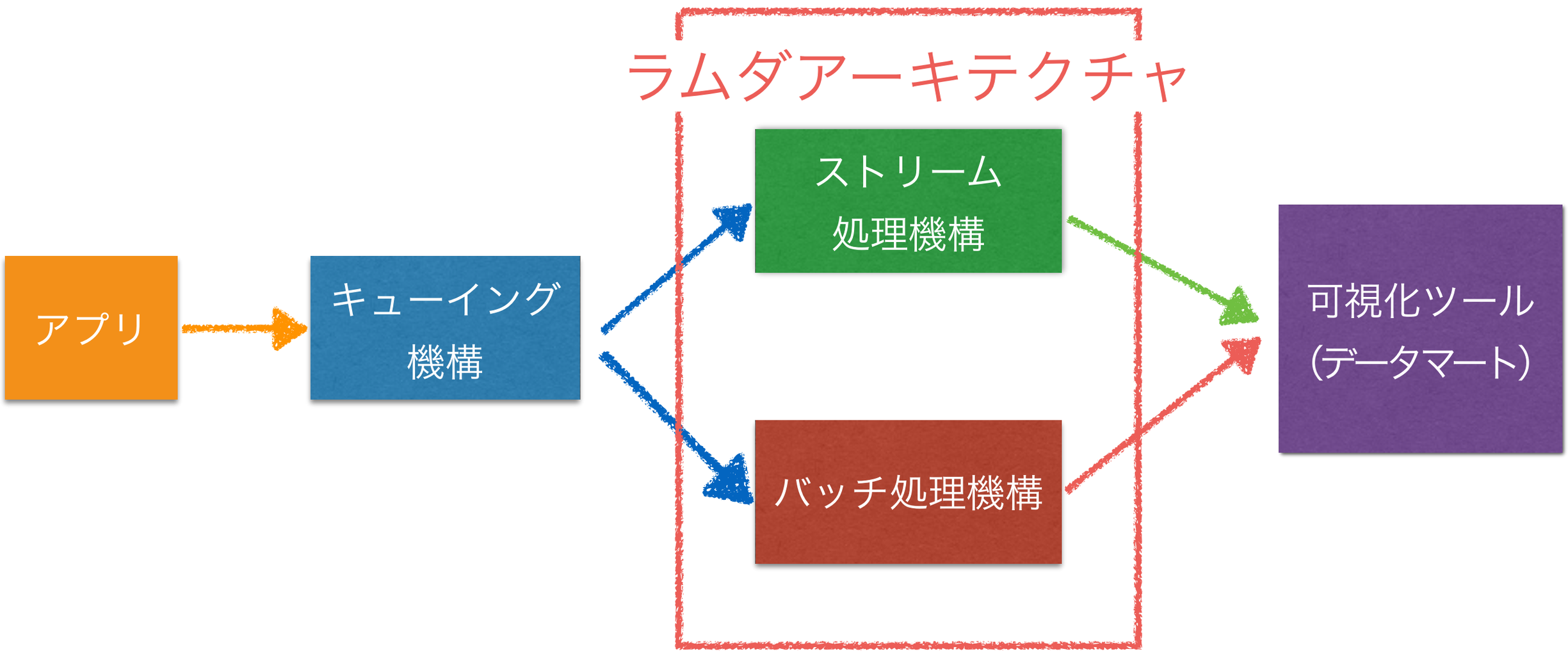
TreasureDataを使った
ラムダアーキテクチャの
システム検討

Toru Takahashi
2014-10-30

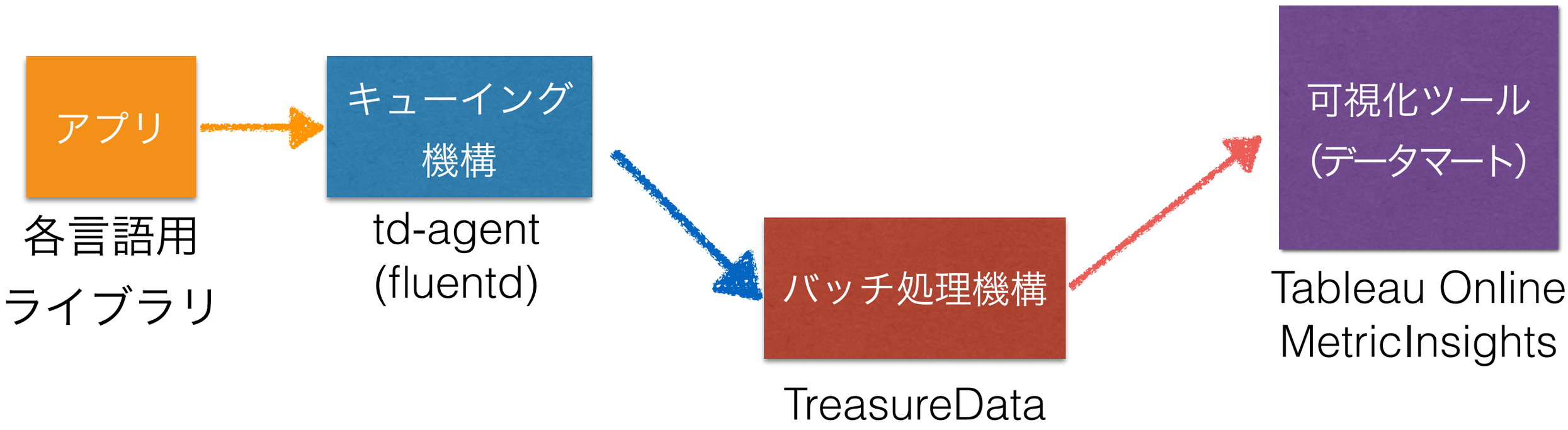
ラムダアーキテクチャとは？

- バッチ処理とストリーム処理の両方を併用したアーキテクチャ
  - バッチ処理の巨大なデータへの集計・加工
  - ストリーム処理のリアルタイムでの集計・加工



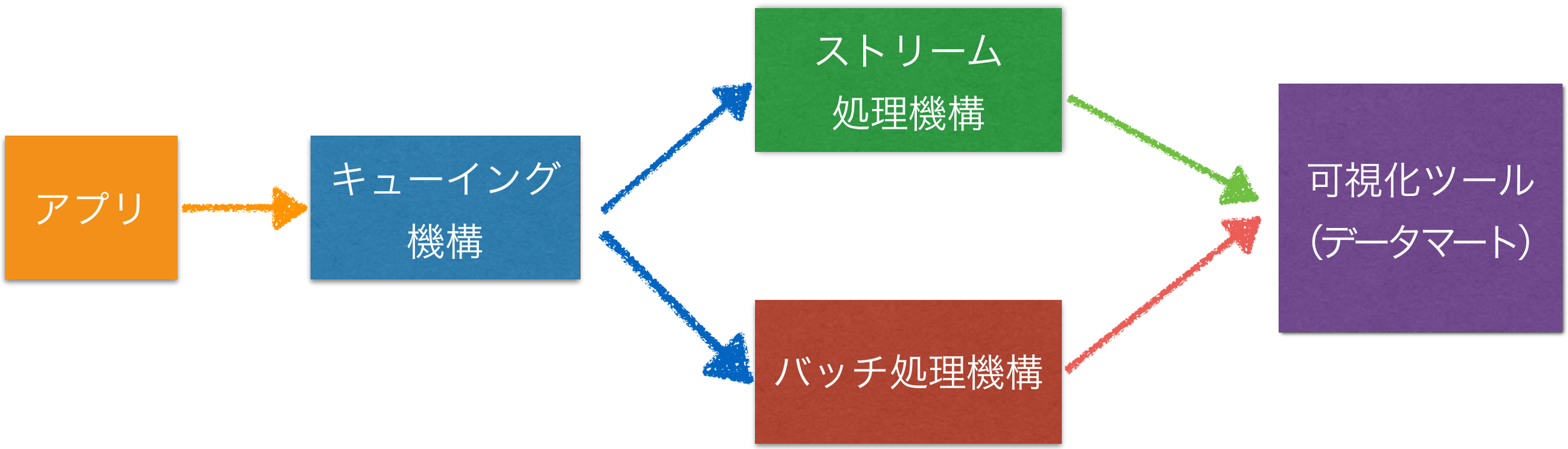
1. バッチ処理システム

- 目的
  - 時/日/月のマネジメント用のKPIを収集するために使う
- 対象
  - 自社のPVやUUなどのKPIを知る基盤が整っていない方々はまずこちらから。
  - KPIを見てディスカッションしてみましょう、利益を出すためにはまず数値を見て話し合って行動することが大事です。



2. ラムダアーキテクチャ型システム

- 目的
  - バッチ処理機構：KPIの集計、学習用データの生成（これにより統計モデル作成）
  - ストリーム処理機構：速報値集計や異常値検知や、データ予測など
- 対象
  - KPIを元に行動し、自分たちの方向性が決まり、システムからもSuggestしてほしいと思っている方々へ
    - ストリーム処理機構は目的によって適したシステムが変わるので難しいところ



キューイング機構

- fluentd
  - http://www.fluentd.org/
- RabbitMQ
  - http://www.rabbitmq.com/
- Amazon Kinesis
  - http://aws.amazon.com/jp/kinesis/
- Microsoft Event Hubs
  - http://azure.microsoft.com/en-us/services/event-hubs/
- Apache Kafka
  - http://kafka.apache.org/

いっぱいあるよ！

バッチ処理機構

- バッチ型：高レイテンシのバッチ処理
  - TreasureData (Hive, Pig)
  - Amazon ElasticMapReduce
    - http://aws.amazon.com/jp/elasticmapreduce/
- マイクロバッチ型：低レイテンシのバッチ処理
  - TreasureData (Presto)
  - Amazon Redshift
    - http://aws.amazon.com/jp/redshift/
  - Google BigQuery
    - https://cloud.google.com/bigquery/
  - Cloudera Impala
    - http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html

いっぱいあるよ！

ストリーム処理機構

- 集計処理
  - fluentd
  - Norikra
    - http://norikra.github.io/
  - Apache Storm
    - https://storm.apache.org/
- ログ検索
  - Splunk
    - http://ja.splunk.com/
  - Elasticsearch + Kibana
    - http://www.elasticsearch.org/overview/kibana/
  - Amazon CloudSearch
    - http://aws.amazon.com/jp/cloudsearch/
- 統計処理
  - Jubatus
    - http://jubat.us/ja/
  - Apache Spark
    - https://spark.apache.org/

いっぱいあるよ！

そのほか

- もう少し具体的なアーキテクチャにまで落として、ブログに書く予定。

- サービスやソフトウェアの特徴についてもまとめる予定。