

Kafka中心連携の仕組み

——バッチからストリームへ

ポイント 1

疎結合

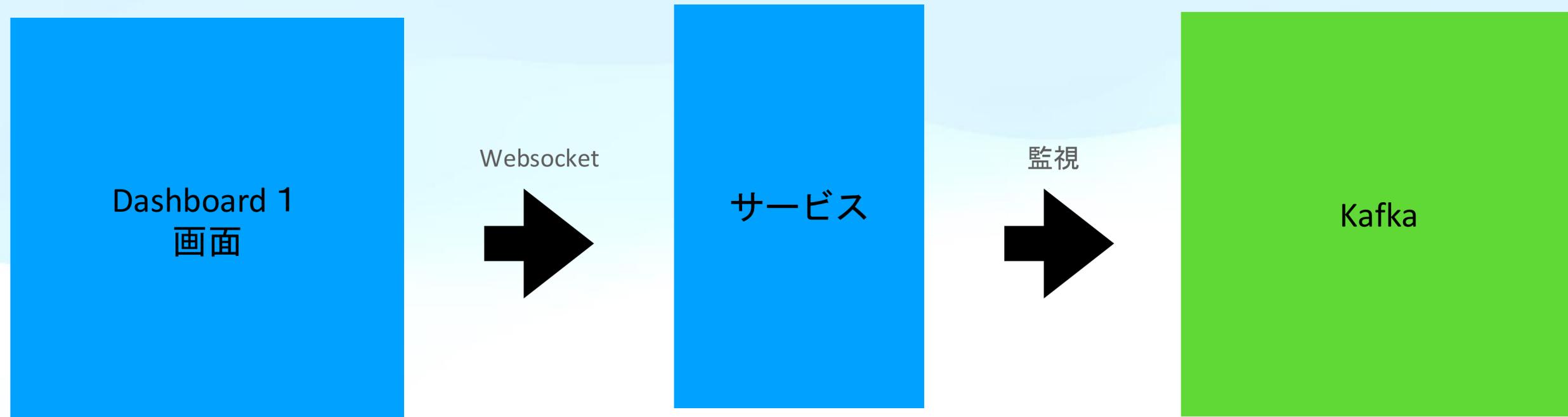


サービス 1 とサービス 2 はお互いの存在を意識しなくでよい。

なので、サービス 1 の変更が即時にサービス 2 に影響を与えない。

ポイント2

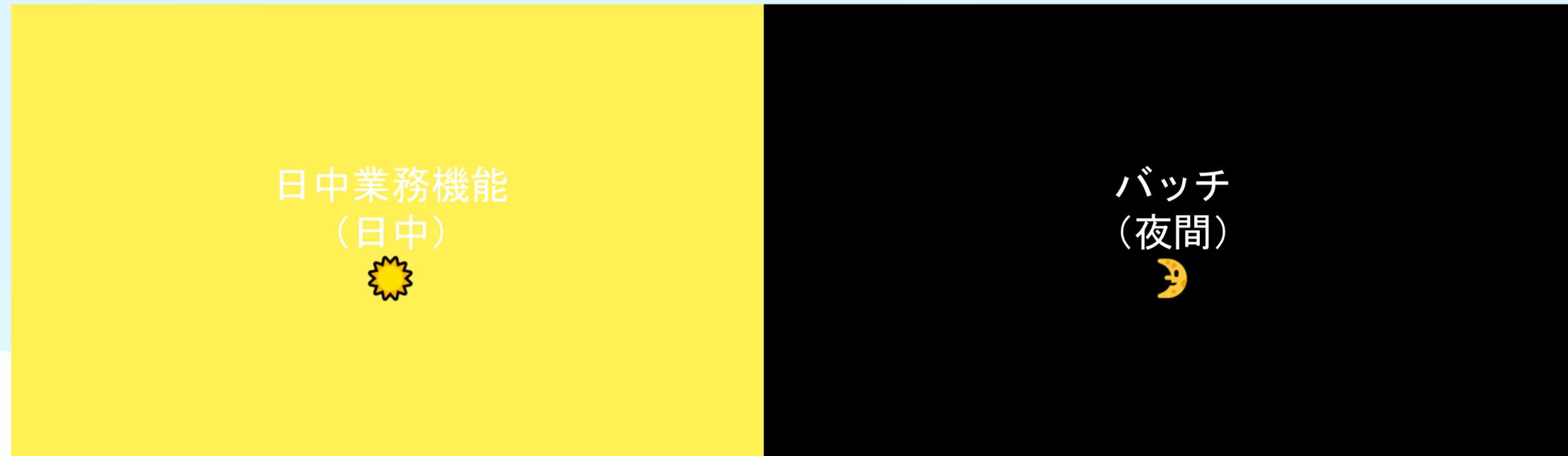
Websocket



- ・ サービスはKafkaのイベント源のConsumerとして監視
- ・ Websocketはサービスと常時接続することで、サーバ側の変化を瞬時に検知でき、画面に描画できる。

ポイント3

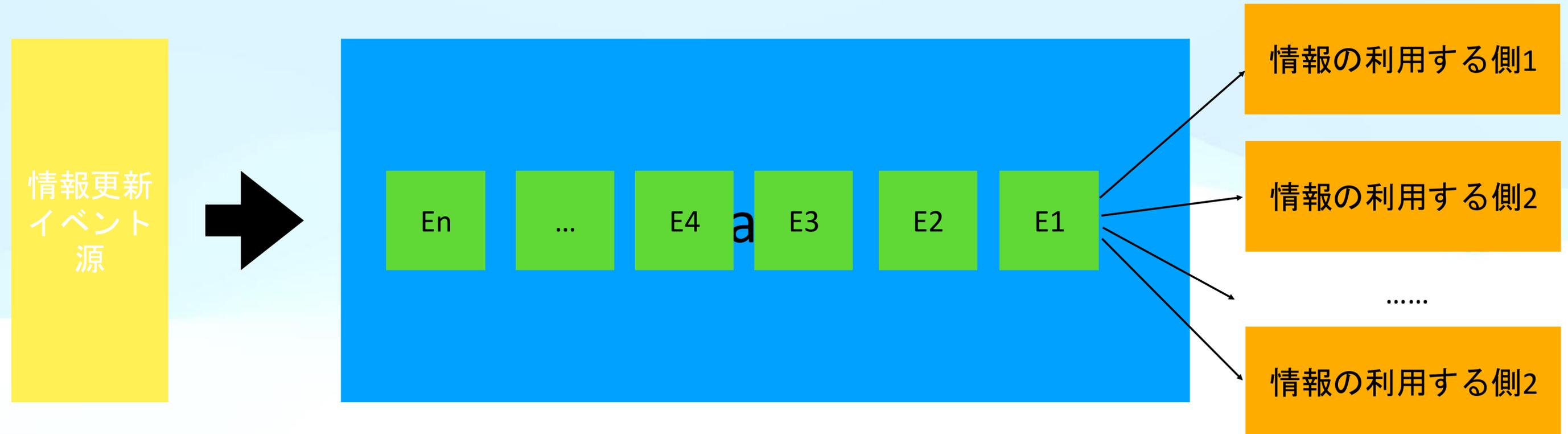
バッチからの脱却し



- ・ 日中で発生した注文や決済は夜間のバッチ処理まで統計されない。
- ・ 全件データを一括処理するので、時間がかかる。

ポイント4

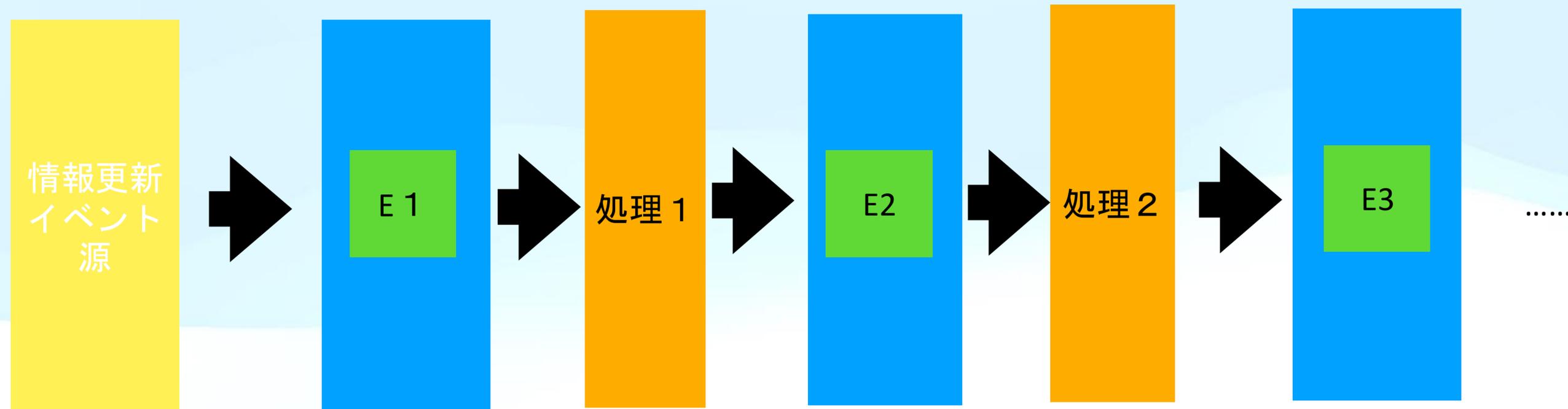
バッチからStream（流れ）へ



- ・ 夜間バッチに待たず、リアルタイムにバックエンドへ流れる。
- ・ 一つのイベント源を複数の利用ができる

ポイント5

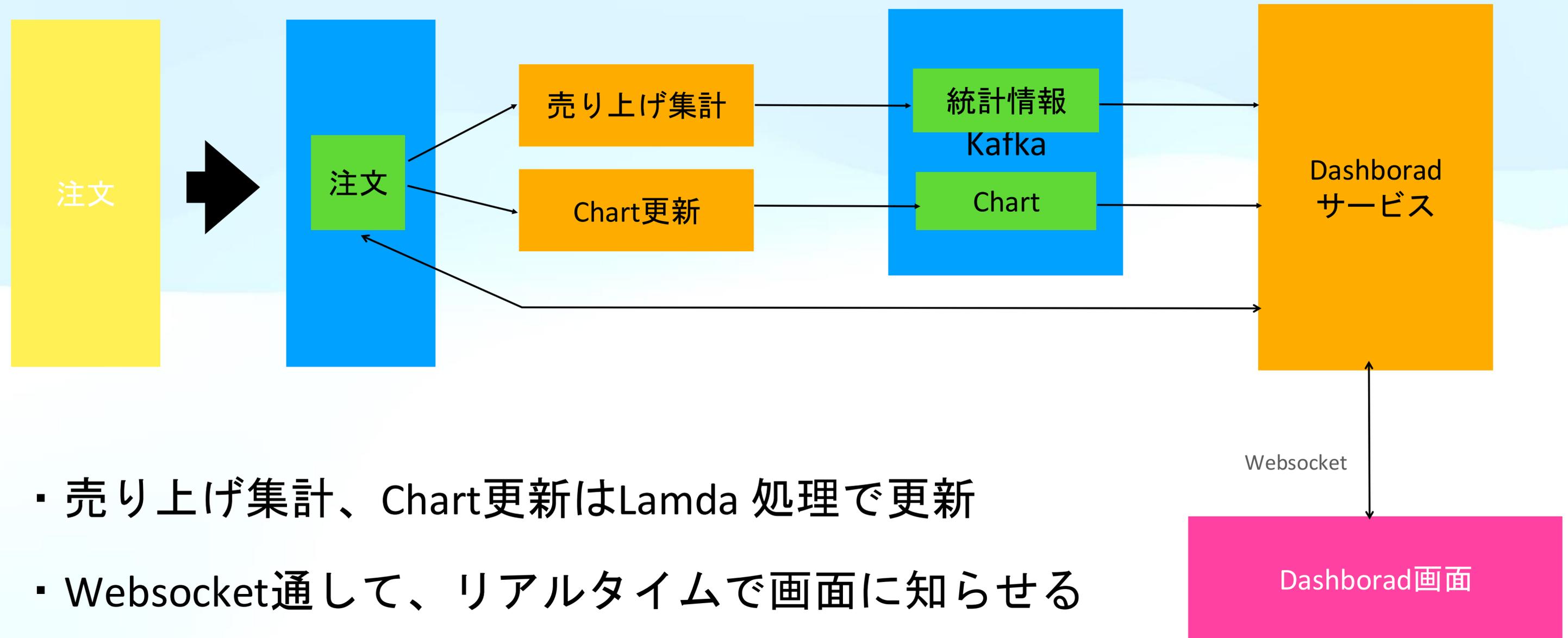
処理はPIPE LINEで組み合わせる



- ・ 情報を処理結果を次の処理へ渡すのもKafkaへ経由
- ・ 処理フローを情報の流れに沿って自由に設計できる

全体図

注文から、ダッシュボードへ反映されるまで



- ・ 売り上げ集計、Chart更新はLambda 処理で更新
- ・ Websocket通して、リアルタイムで画面に知らせる