

Beyond the Watermark

On-Demand Backfilling in Flink

Maxim Fateev, Staff SDE

FlinkForward

2016

UBER

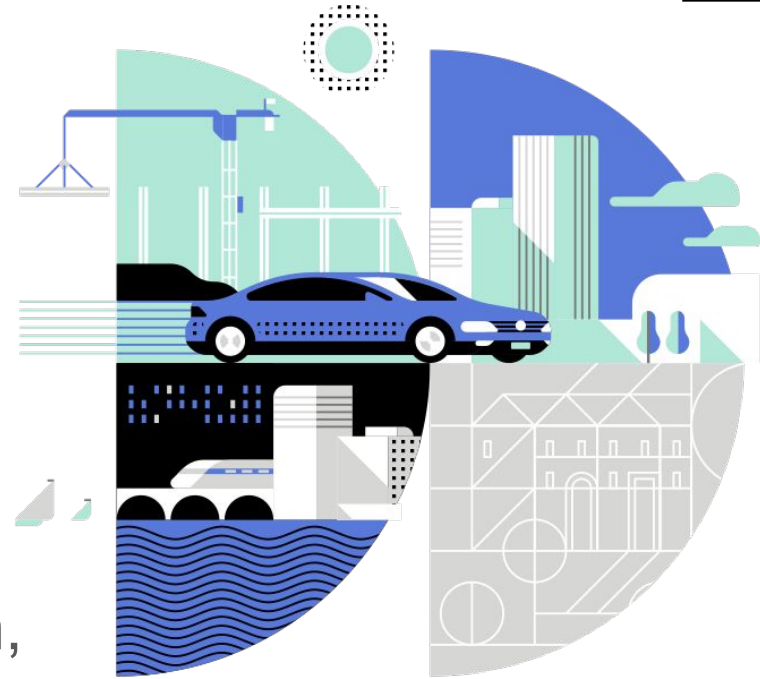


Who Am I

- Amazon Internal Messaging Infrastructure
- AWS SQS Storage Engine
- AWS Simple Workflow Service (SWF)
- Uber [Cherami](#) Messaging System
 - to be open sourced fall 2016

Uber Marketplace

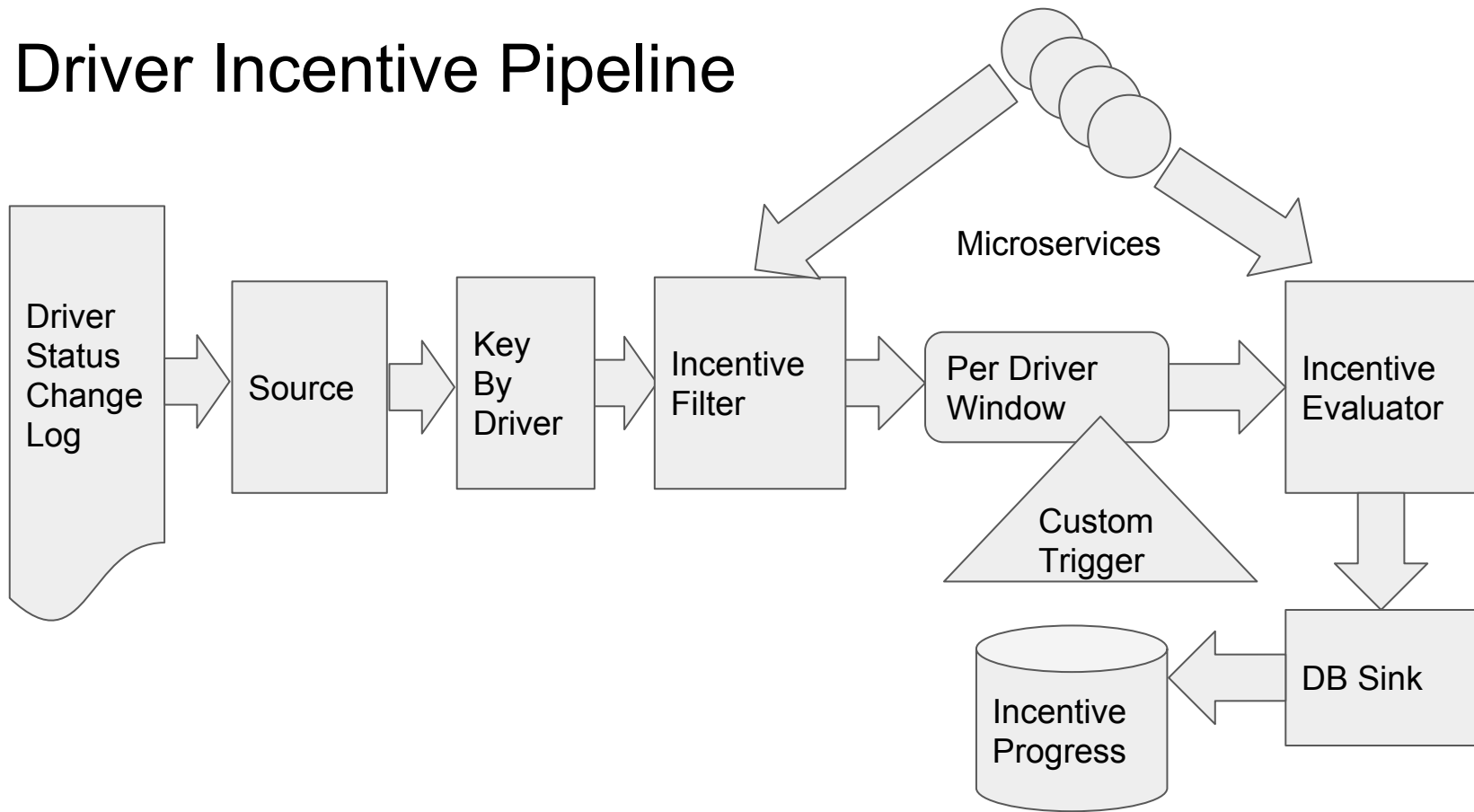
- Ride within minutes
- City needs a minimal number of riders and drivers
- Incentives is a mechanism to bootstrap a marketplace
- Incentives are specific to location, time, type of vehicle, driver rating, etc.



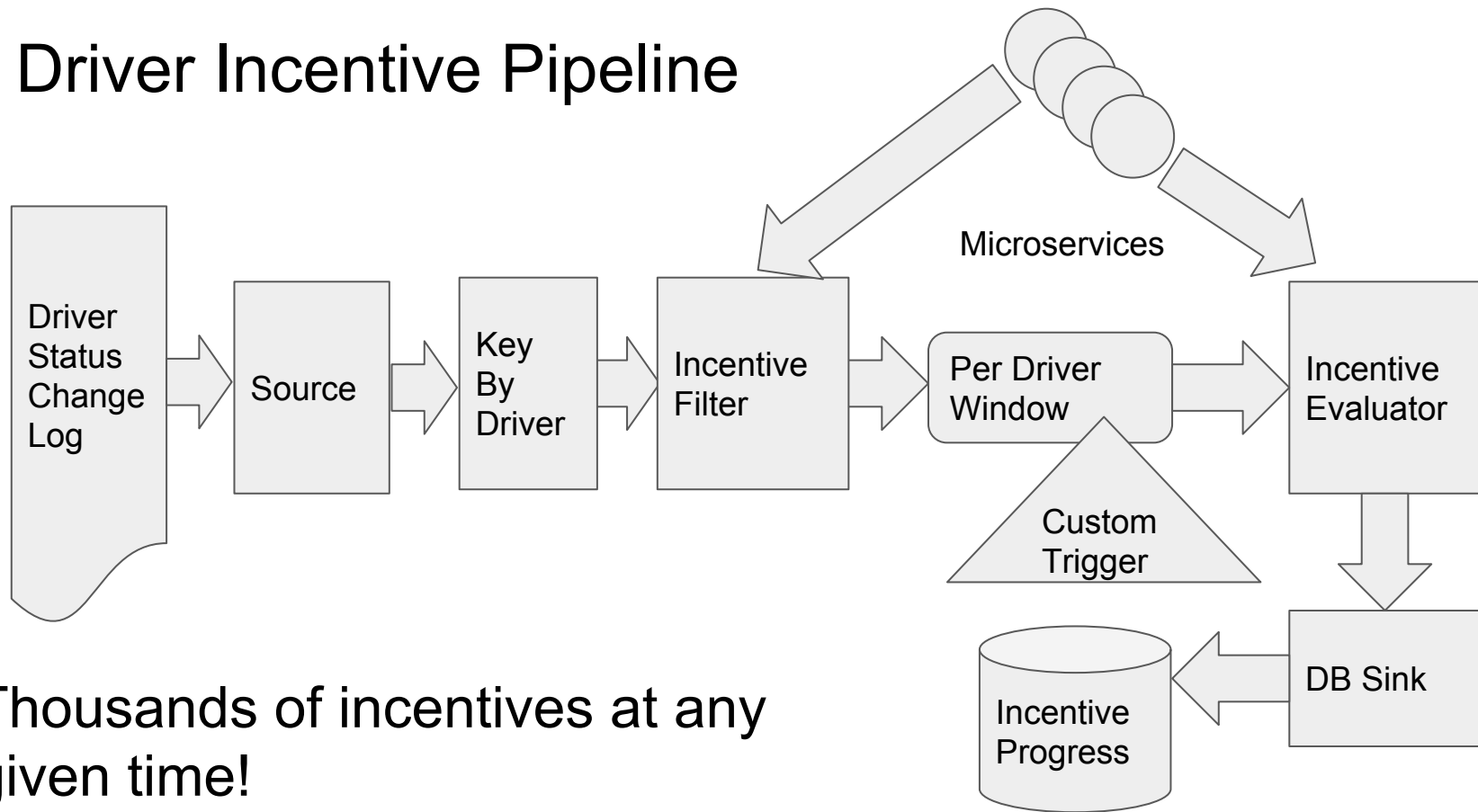
Driver Incentive Example

- Guarantee of \$40 an hour
 - UberX
 - From August 21st to August 26th
 - San Francisco
 - Minimum 20 hours online
 - Minimum rating of 4.5
 - Acceptance rate of 0.8

Driver Incentive Pipeline

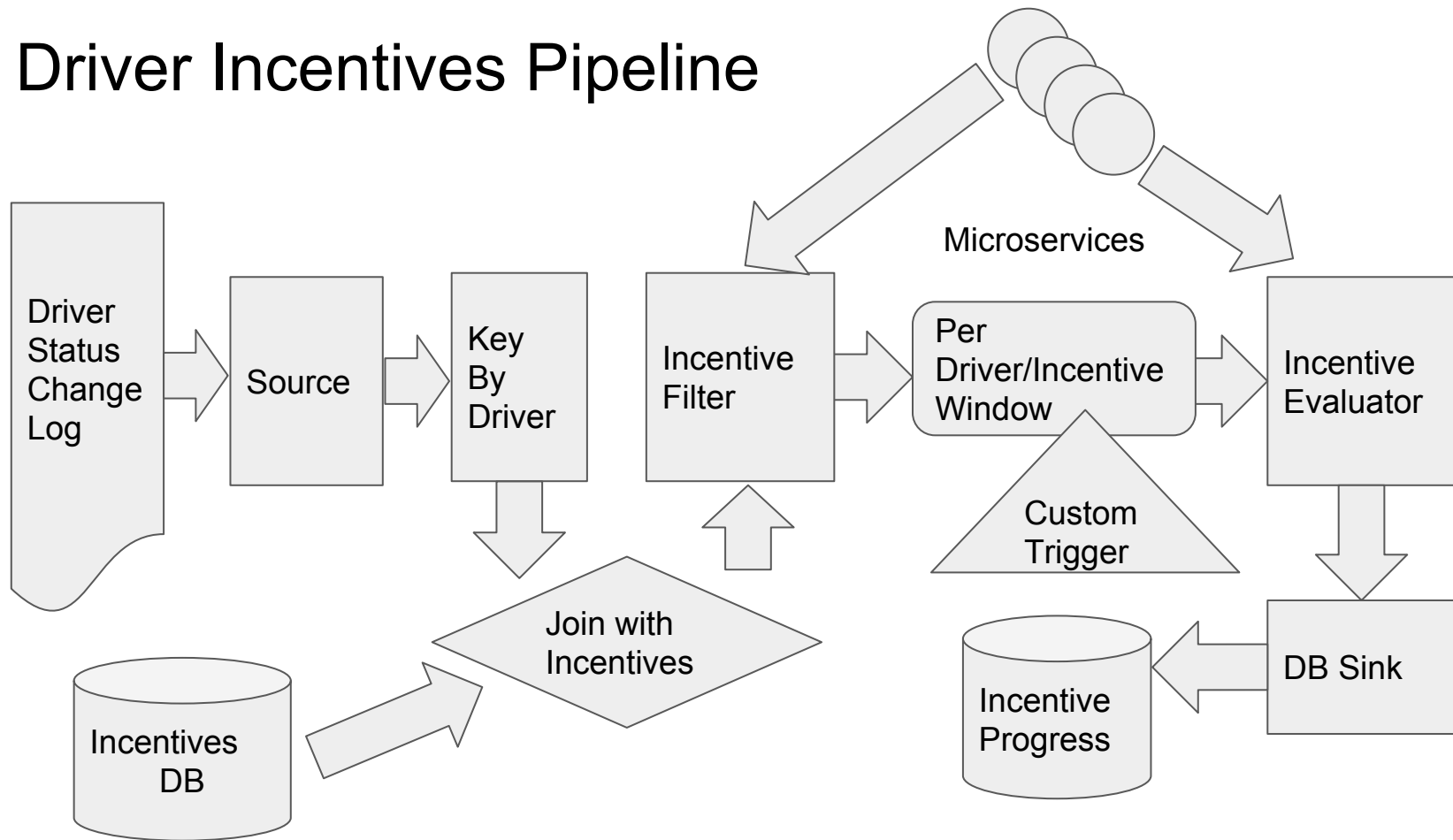


Driver Incentive Pipeline

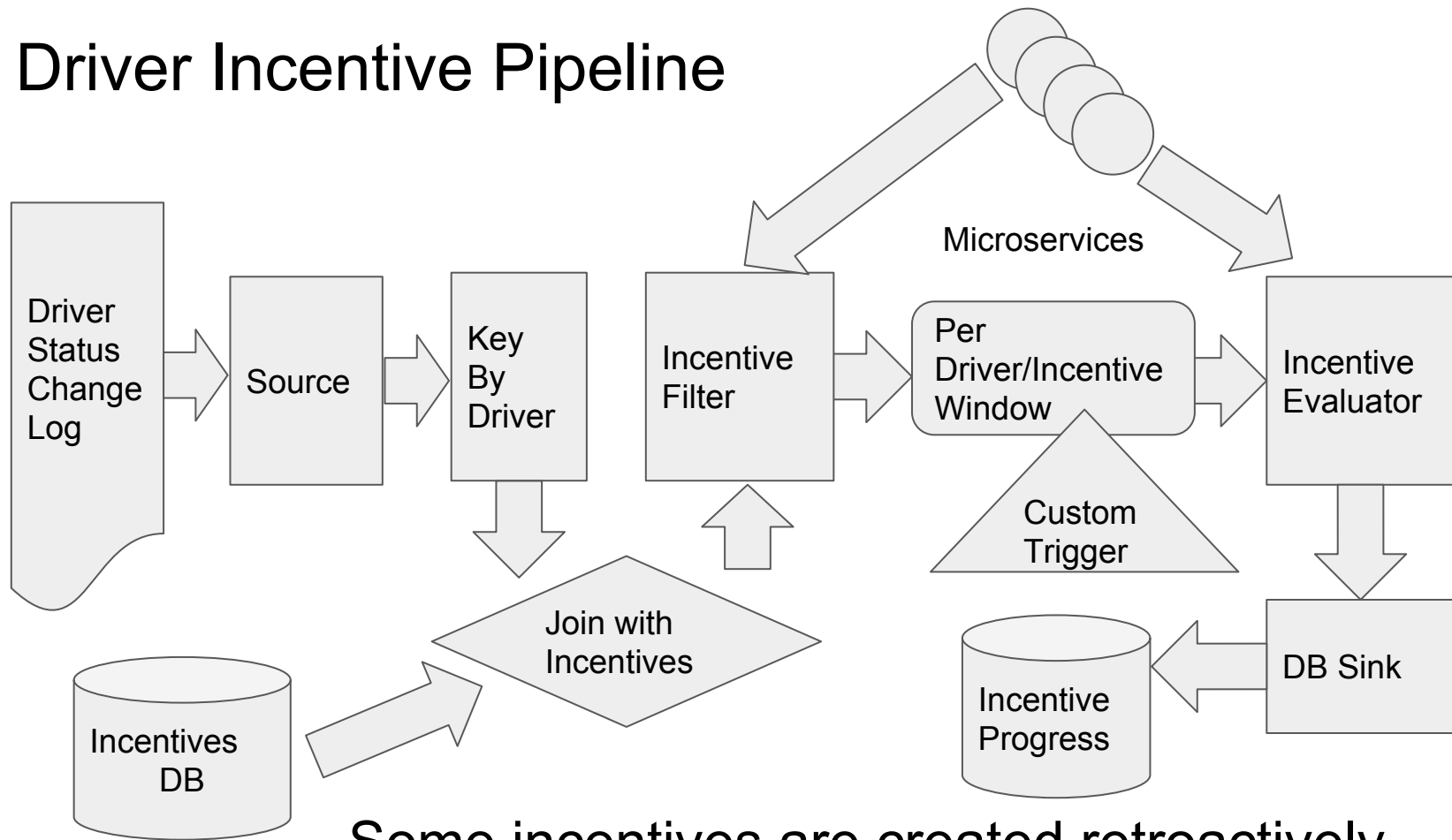


Thousands of incentives at any given time!

Driver Incentives Pipeline



Driver Incentive Pipeline



Some incentives are created retroactively

Retroactive Incentive Creation

- pipeline for incentives created up front
- backfill pipeline that runs periodically for retroactively created incentives

Retroactive Incentive Creation

- pipeline for incentives created up front
- backfill pipeline that runs periodically for retroactively created incentives
- **What to do when backfill reaches “current” events?**

Retroactive Incentive Creation

- pipeline for incentives created up front
- backfill pipeline that runs periodically for retroactively created incentives
- **What to do when backfill reaches “current” events?**
 - Keep running it until end of all incentive periods
 - or
 - Hand over incentive to the “current events” pipeline

Ideal Solution

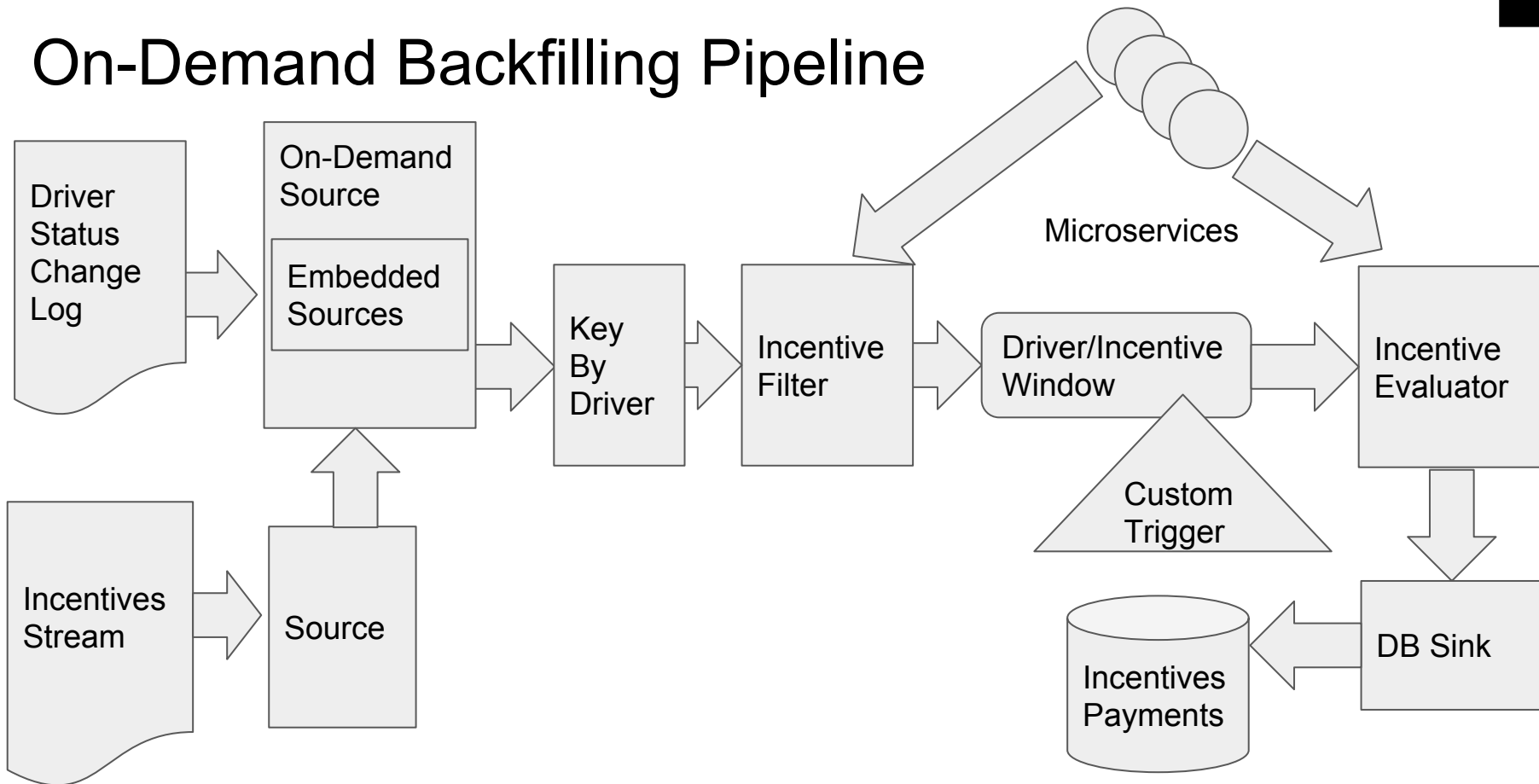
- Single pipeline instance
- Supports retroactive incentive creation



Our Solution: On-Demand Query “Source”

- Not a Flink Source as it *consumes* DataStream of incentives
- Reads Driver Status Change Log
- Emits *state change / incentive* pairs
- For every incentive emits pairs from the beginning of the incentive period
 - Internally has multiple source instances
 - Periodically starts source stream scan from the oldest incentive to backfill
- Global watermark is not used
 - Per incentive watermark would be great
- Checkpoint includes the list of not yet completed incentives and each internal source checkpoint

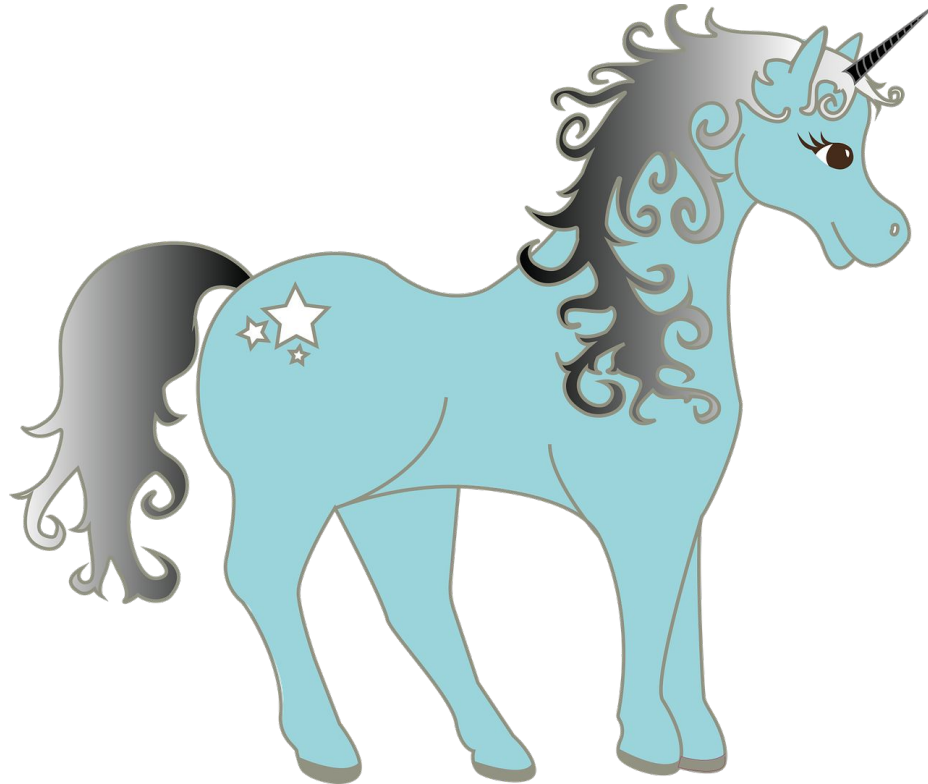
On-Demand Backfilling Pipeline



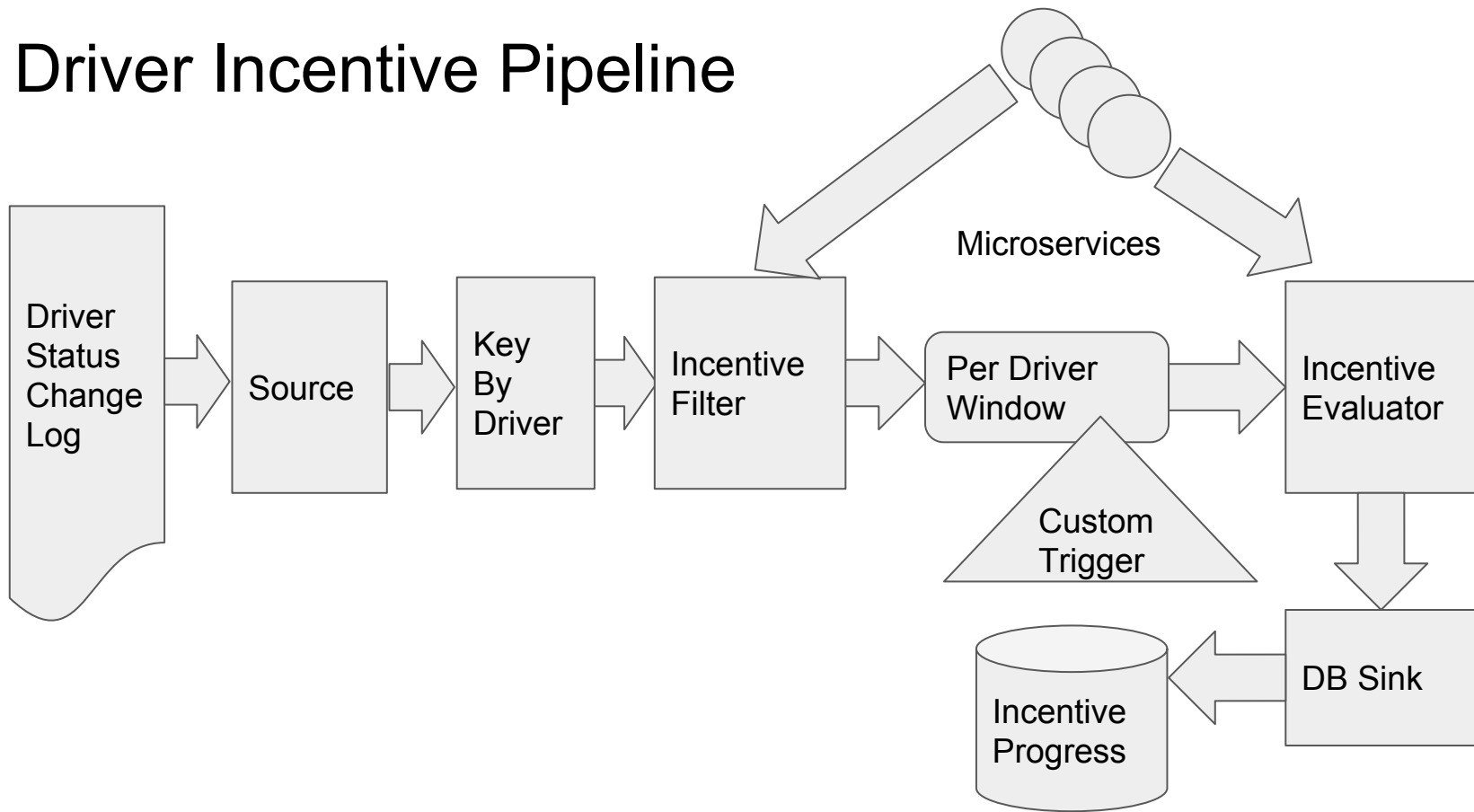
Summary

- DataStream that contains union of current and backfill messages
- DataStream source doesn't need to be at the start of a pipeline
- Source that changes its behavior based on its inputs is a useful abstraction
- Global Watermark is not adequate

Generic Solution?



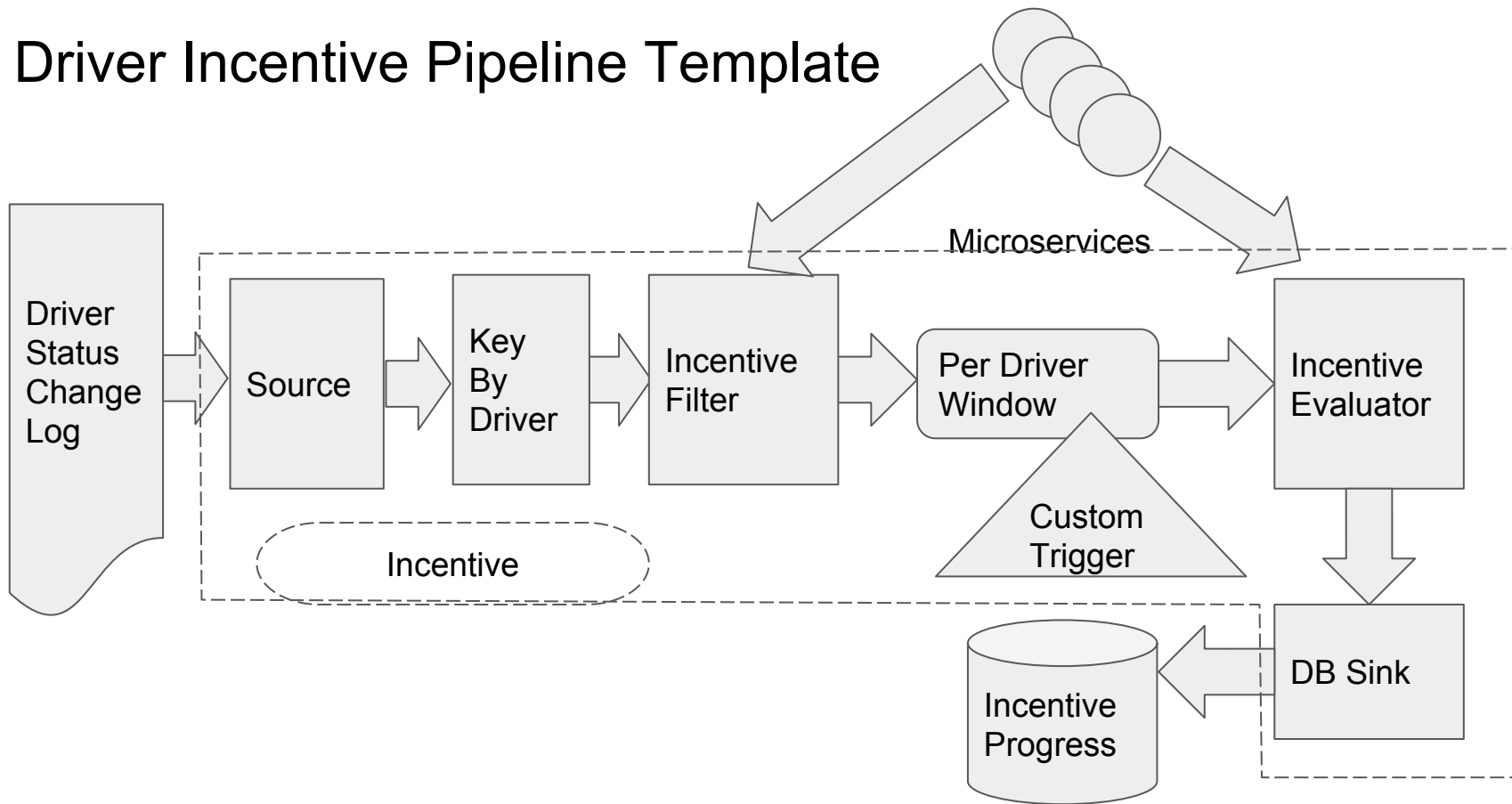
Driver Incentive Pipeline



Strawman: Pipeline Template

- Pipeline that depends on some parameters to be instantiated
 - Driver Incentive would be such parameter
- Parameter values are specified when pipeline is instantiated
- All instances of the templated pipeline share the same operator instances
- All streams and operators are implicitly keyed on parameter values
- Any sources, operators and sinks have access to parameter values
- Watermarks and state values are scoped to an instance
- Implementation of sources, operators and sinks might be optimized to share resources between instances
 - Source that performs single Kafka stream read for all instances that were started for the last hour

Driver Incentive Pipeline Template



Additional Feature Requests

- Per message error handling
- Runtime Visibility
 - Look at the state of any window and associated trigger in the system
 - Overhear any data stream for a task
- Triggering on empty windows
- Pipeline graph rewriting
 - Interceptors
 - Platform components
- Pre-checkpoint callback for sources