



confluent

KEEP YOUR DATA CLOSE AND YOUR CACHES HOTTER

USING APACHE KAFKA, CONNECT AND KSQL

@GAMUSSA | @RIFERREI | #IMCSUMMIT



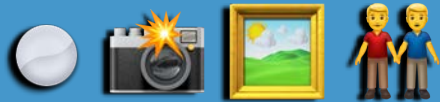
@GAMUSSA | @RIFERREI | #IMCSUMMIT

RAFFLE, YEAH



RAFFLE, YEAH

- Follow @gamussa @riferrei



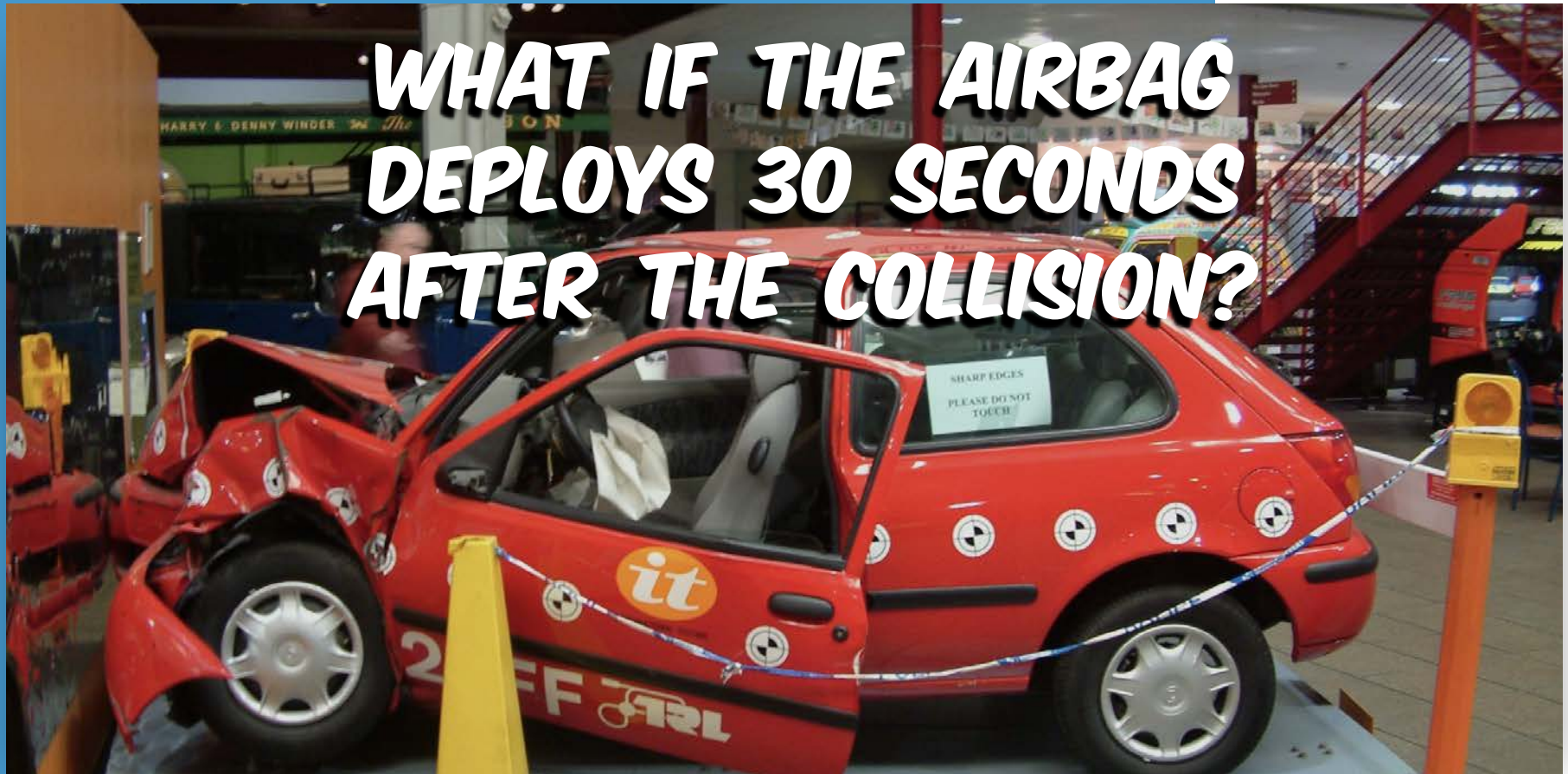
- Tag @gamussa @riferrei

- With #IMCSummit

**DATA IS ONLY USEFUL
IF IT IS FRESH AND
CONTEXTUAL**



WHAT IF THE AIRBAG DEPLOYS 30 SECONDS AFTER THE COLLISION?





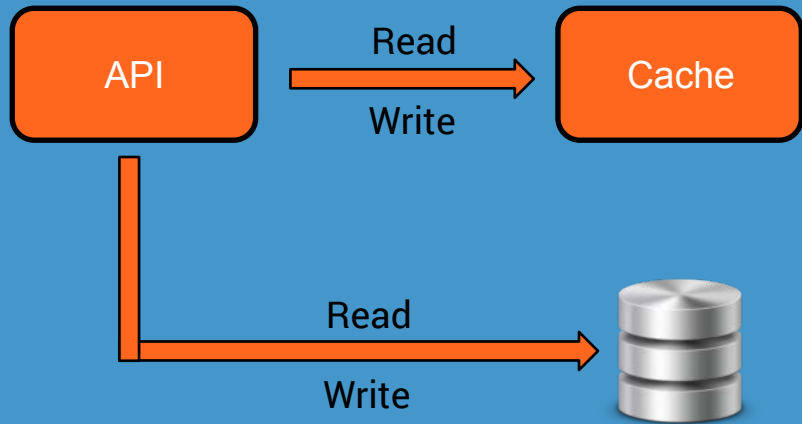
DECEMBER 6TH, 2010: COMMUTER RAIL TRAIN HITS ELDERLY DRIVER



**WHAT IF THE
INFORMATION
ABOUT THE COMMUTER
RAIL
TRAIN IS OUTDATED?**

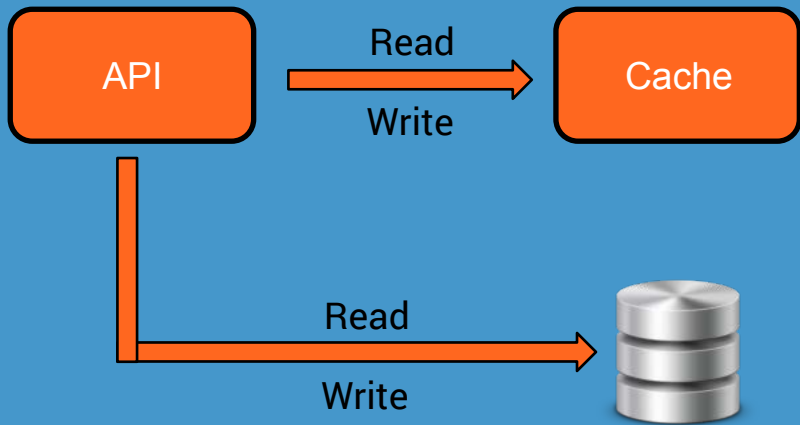
**CACHES CAN BE A
SOLUTION FOR DATA
THAT IS FRESH**

APIS NEED TO ACCESS DATA FREELY AND EASILY

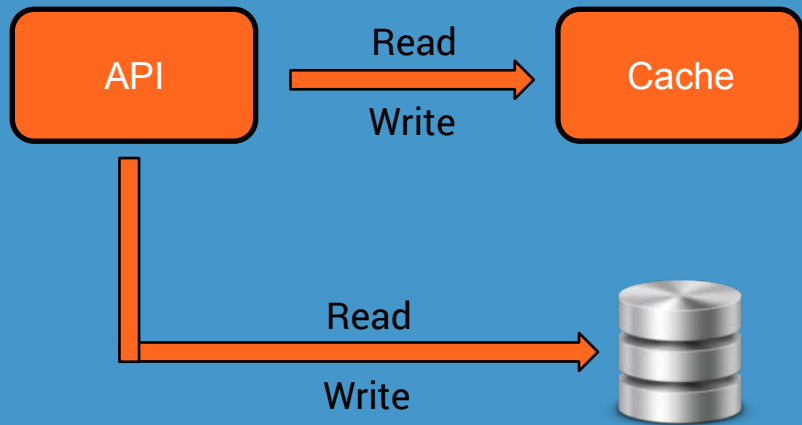


APIS NEED TO ACCESS DATA FREELY AND EASILY

- Data should never be treated as a scarce resource in applications

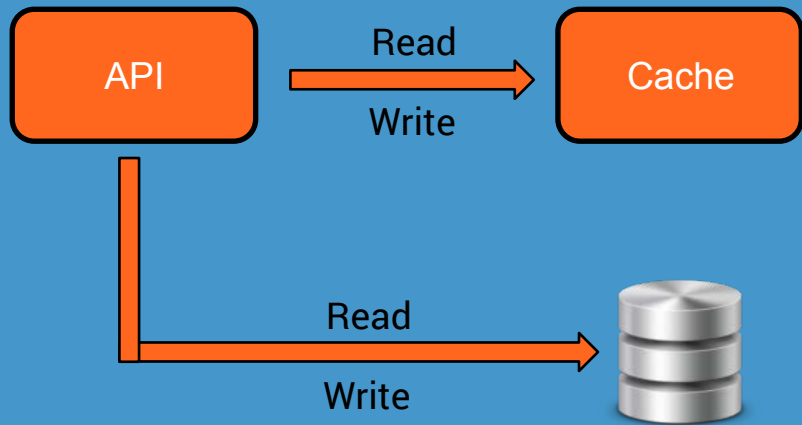


APIS NEED TO ACCESS DATA FREELY AND EASILY



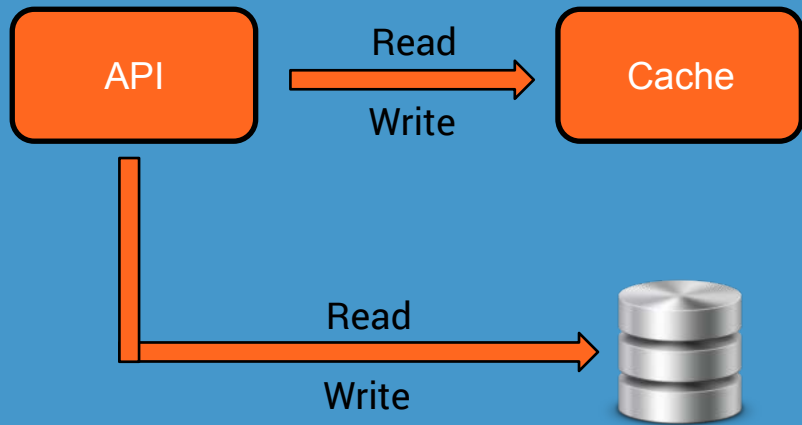
- Data should never be treated as a scarce resource in applications
- Latency should be kept as minimal to ensure a better user experience

APIS NEED TO ACCESS DATA FREELY AND EASILY



- Data should never be treated as a scarce resource in applications
- Latency should be kept as minimal to ensure a better user experience
- Data should not be static: keep the data fresh continuously

APIS NEED TO ACCESS DATA FREELY AND EASILY



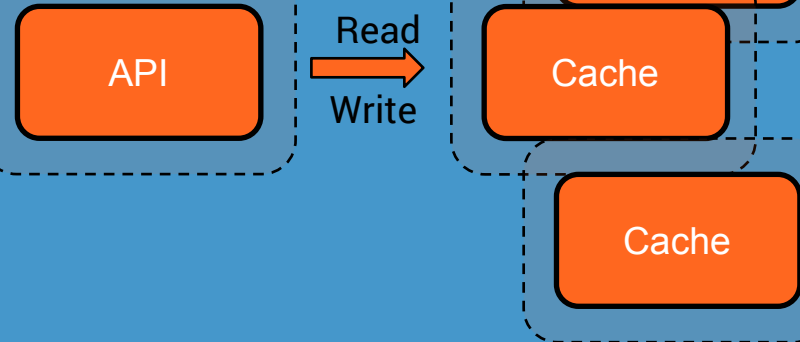
- Data should never be treated as a scarce resource in applications
- Latency should be kept as minimal to ensure a better user experience
- Data should not be static: keep the data fresh continuously
- Find ways to handle large amounts of data without breaking the APIs

CACHES CAN BE EITHER BUILT-IN OR DISTRIBUTED

Built-in Caches



Distributed Caches



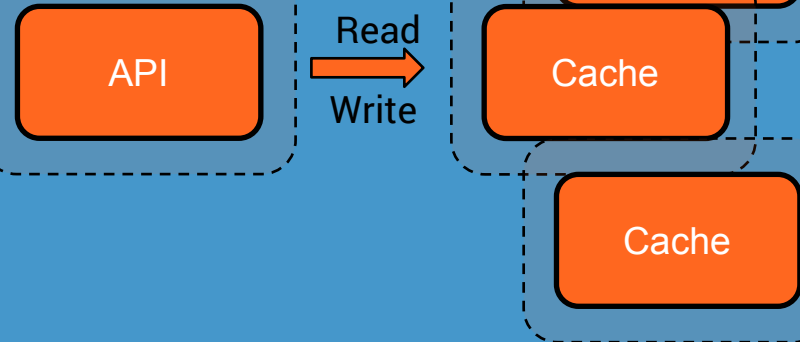
CACHES CAN BE EITHER BUILT-IN OR DISTRIBUTED

- If data can fit into the API memory, then you should use built-in caches

Built-in Caches



Distributed Caches



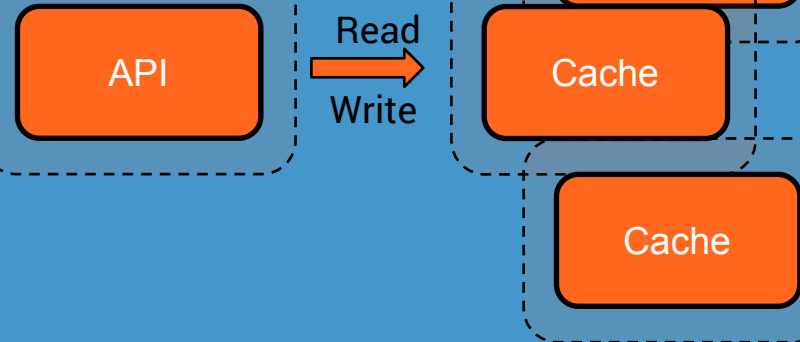
CACHES CAN BE EITHER BUILT-IN OR DISTRIBUTED

Built-in Caches



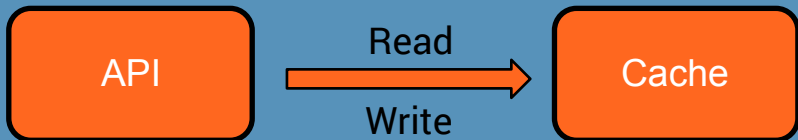
- If data can fit into the API memory, then you should use built-in caches
- Otherwise, you may need to use distributed caches for large sizes

Distributed Caches

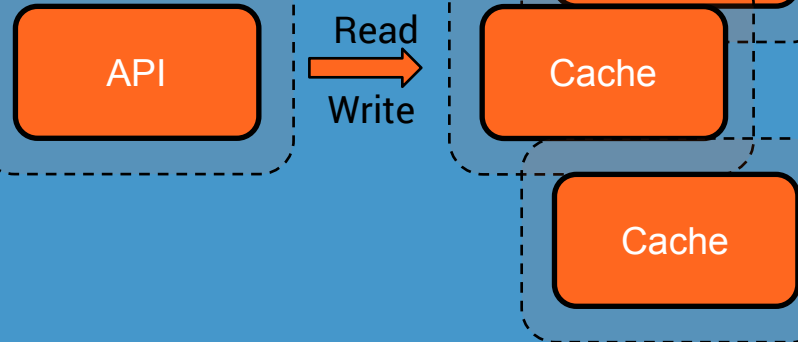


CACHES CAN BE EITHER BUILT-IN OR DISTRIBUTED

Built-in Caches



Distributed Caches



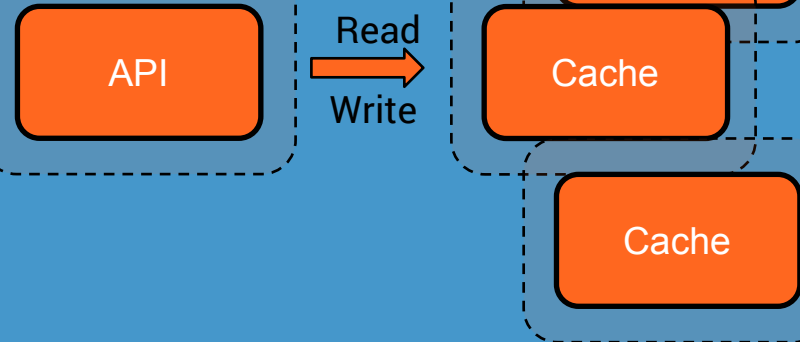
- If data can fit into the API memory, then you should use built-in caches
- Otherwise, you may need to use distributed caches for large sizes
- Some cache implementations provides the best of both cases

CACHES CAN BE EITHER BUILT-IN OR DISTRIBUTED

Built-in Caches



Distributed Caches



- If data can fit into the API memory, then you should use built-in caches
- Otherwise, you may need to use distributed caches for large sizes
- Some cache implementations provides the best of both cases
- For distributed caches, make sure to always find a good way to $O(1)$

DEMO



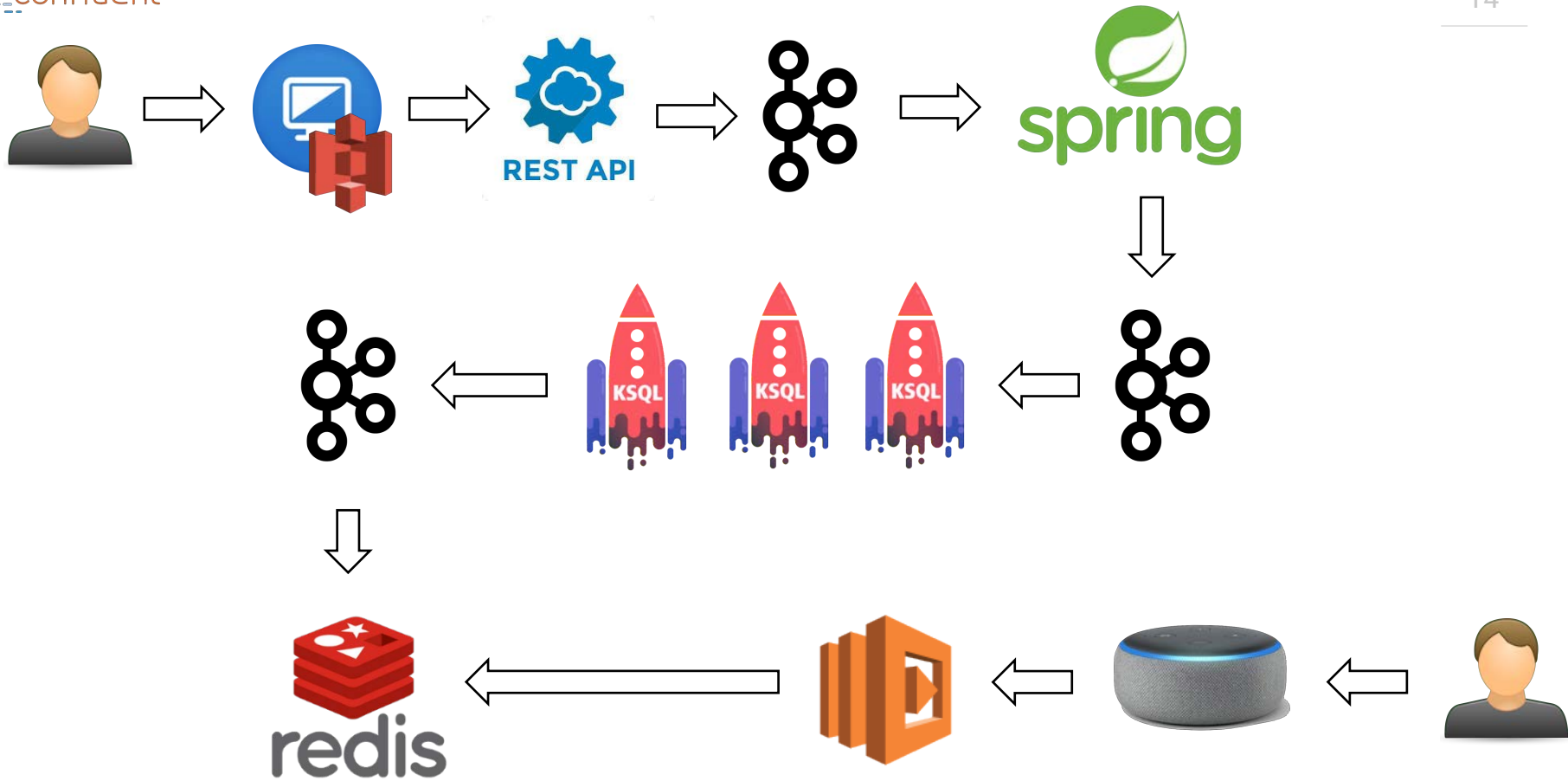


@GAMUSSA | @RIFERREI | #IMCSUMMIT

**JOIN
THE FUN!**



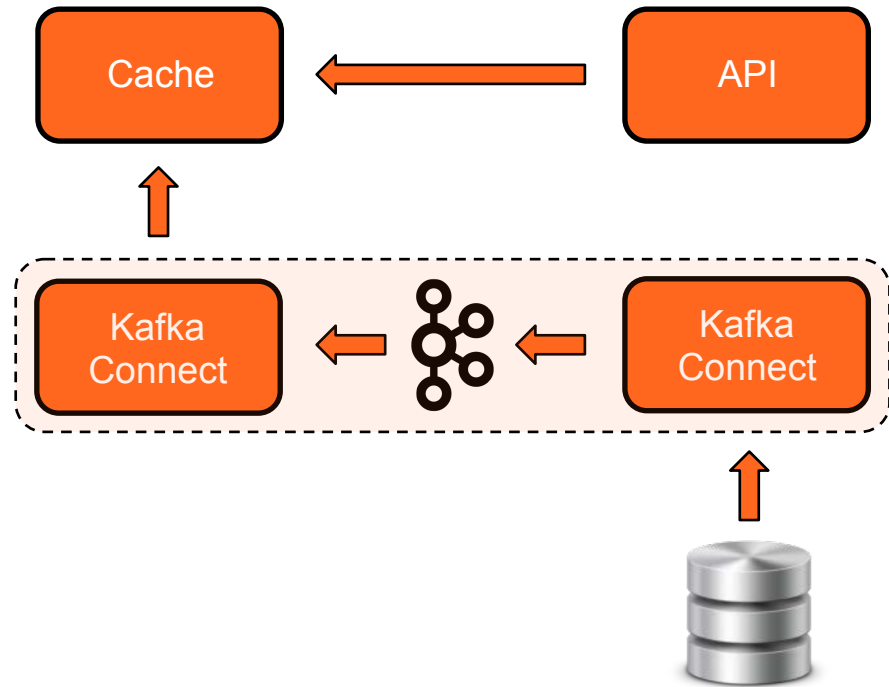
@GAMUSSA | @RIFERREI | #IMCSUMMIT



CACHING PATTERNS

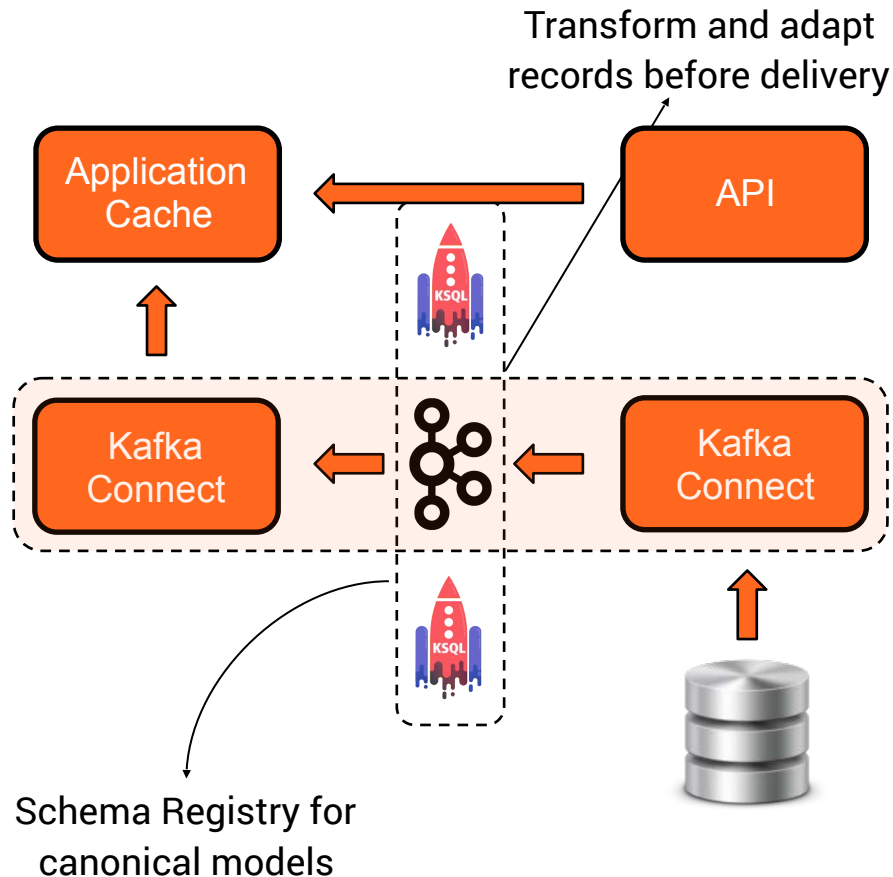
CACHING PATTERN: REFRESH AHEAD

- Proactively updates the cache
- Keep the entries always in-sync
- Ideal for latency sensitive cases
- Ideal when data read is costly
- It may need initial data loading



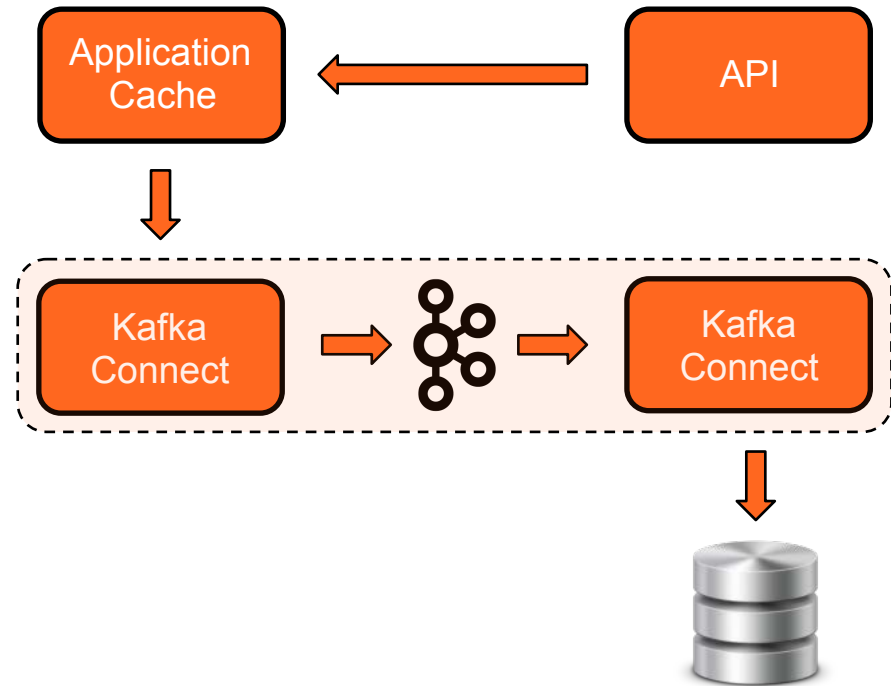
CACHING PATTERN: REFRESH AHEAD / ADAPT

- Proactively updates the cache
- Keep the entries always in-sync
- Ideal for latency sensitive cases
- Ideal when data read is costly
- It may need initial data loading



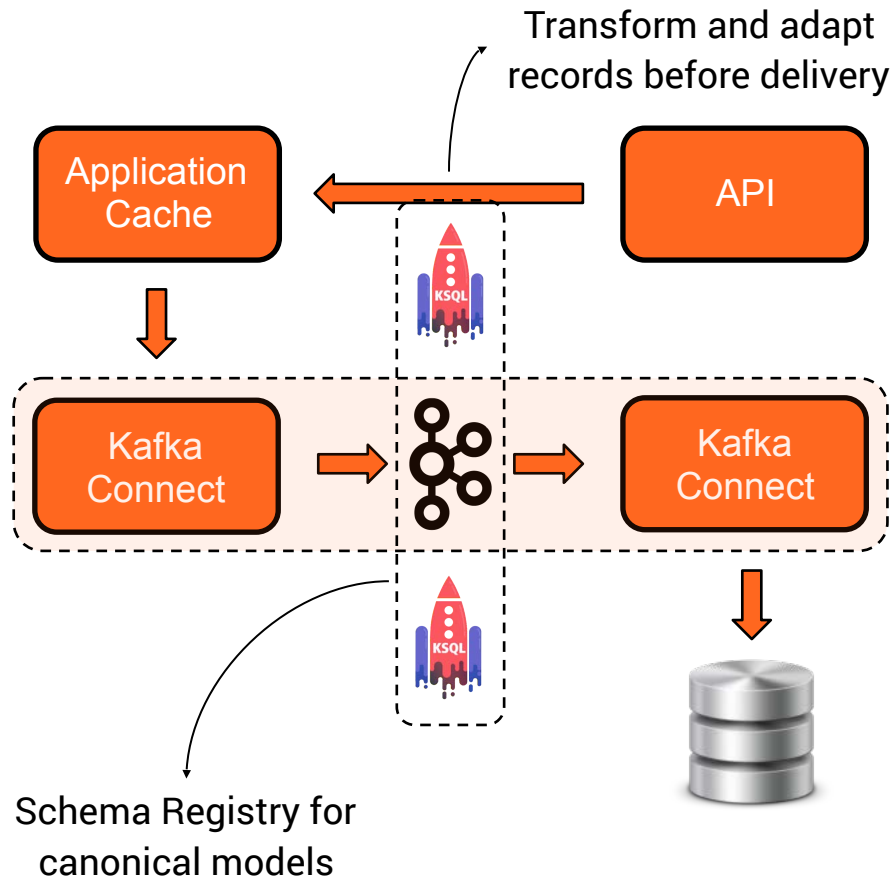
CACHING PATTERN: WRITE BEHIND

- Removes I/O pressure from app
- Allows true horizontal scalability
- Ensures ordering and persistence
- Minimizes DB code complexity
- Totally handles DB unavailability



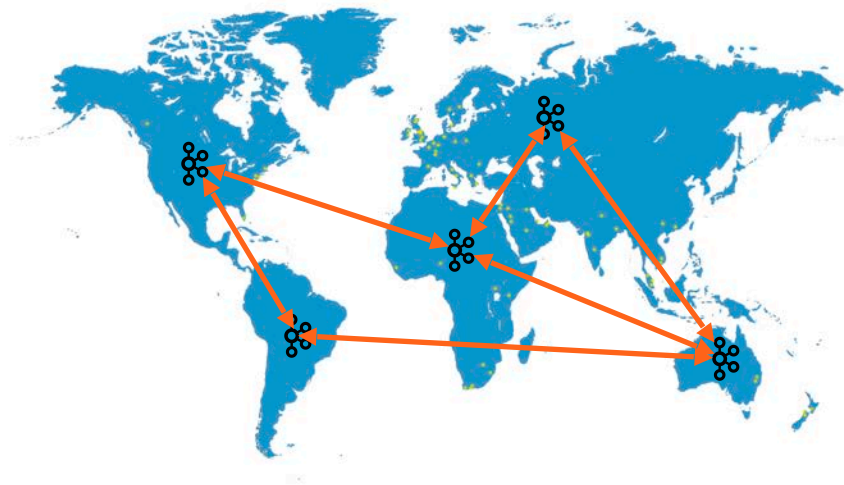
CACHING PATTERN: WRITE BEHIND / ADAPT

- Removes I/O pressure from app
- Allows true horizontal scalability
- Ensures ordering and persistence
- Minimizes DB code complexity
- Totally handles DB unavailability



CACHING PATTERN: EVENT FEDERATION

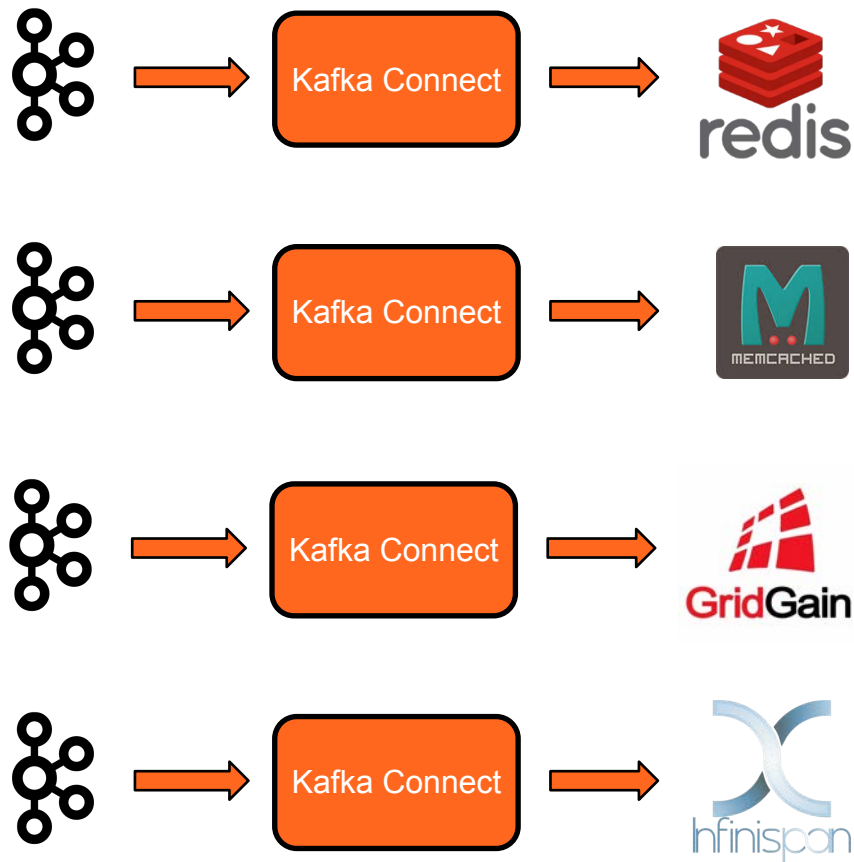
- Replicates data across regions
- Keep multiple regions in-sync
- Great to improve RPO and RTO
- Handles lazy/slow networks well
- Works well if its used along with Read-Through and Write-Through patterns.



KAFKA CONNECT IMPLEMENTATION STRATEGIES

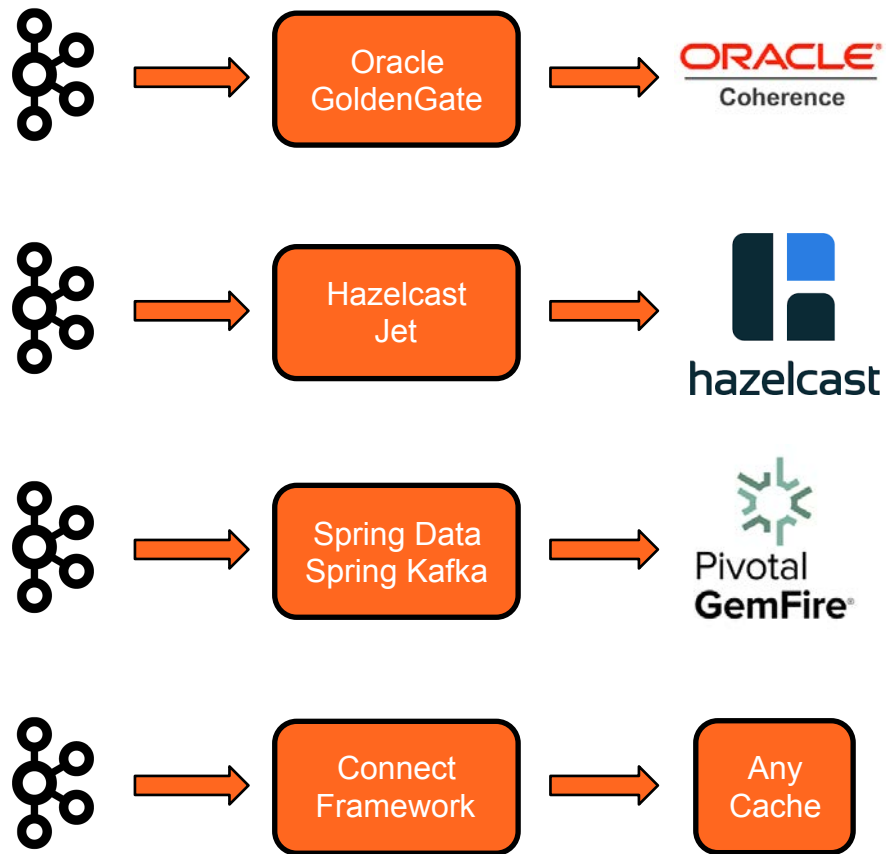
KAFKA CONNECT SUPPORT FOR IN-MEMORY CACHES

- Connector for Redis is open and it is available in Confluent Hub
- Connector for Memcached is open and it is available in Confluent Hub
- Connectors for both GridGain and Apache Ignite implementations.
- Connector for InfiniSpan is open and is maintained by Red Hat



FRAMEWORKS FOR OTHER IN-MEMORY CACHES

- Oracle provides HotCache from GoldenGate for Oracle Coherence
- Hazelcast has the Jet framework, which provides support for Kafka
- Pivotal GemFire (Apache Geode) has good support from Spring
- Good news: you can always write your own sink using Connect API

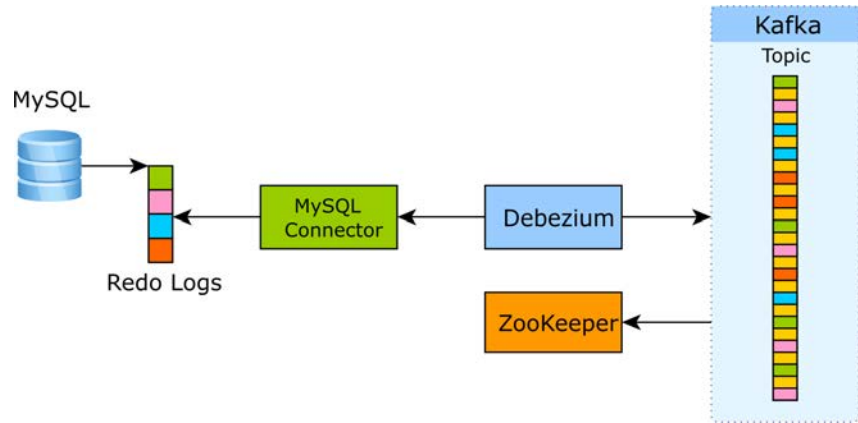


INTERESTED ON DB CDC? THEN MEET DEBEZIUM!

- Amazing CDC technology to pull data out from databases to Kafka
- Works in a log level, which means true CDC implementation for your projects instead of record polling
- Open-source maintained by Red Hat. Have broad support for many popular databases.
- It is built on top of Kafka Connect

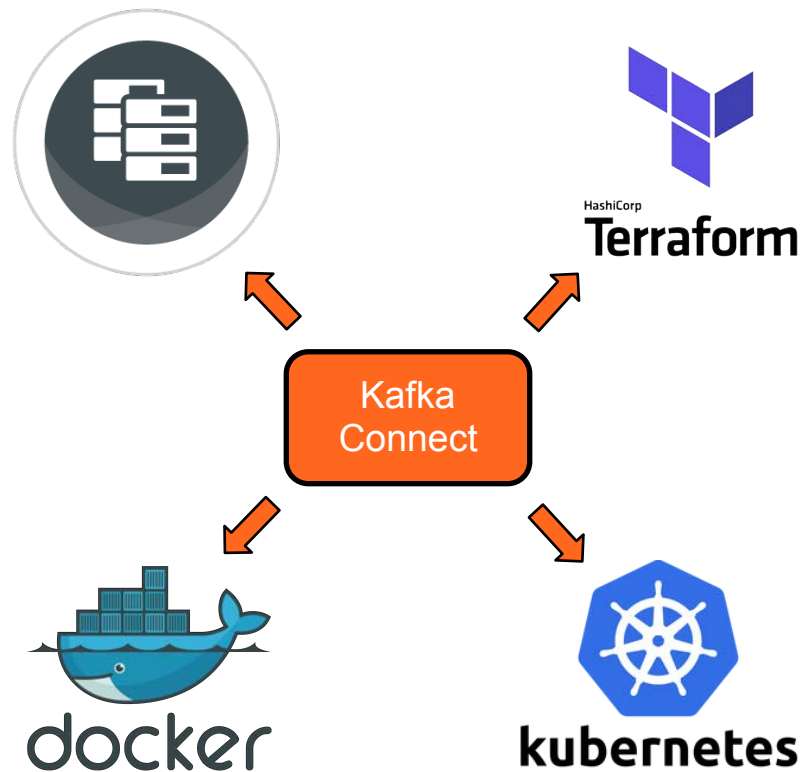


debezium



SUPPORT FOR RUNNING KAFKA CONNECT SERVERS

- **Run by yourself on BareMetal:**
<https://kafka.apache.org/downloads> <https://www.confluent.io/download>
- **IaaS on AWS or Google Cloud:**
<https://github.com/confluentinc/ccloud-tools>
- **Running using Docker Containers:**
<https://hub.docker.com/r/confluentinc/cp-kafka-connect/>
- **Running using Kubernetes:** <https://github.com/confluentinc/cp-helm-chart> <https://www.confluent.io/confluent-operator/>



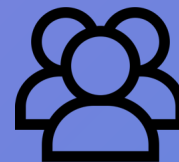
Stay in touch



cnfl.io/blog



cnfl.io/slack



cnfl.io/meetups





THANKS!

@riferrei

ricardo@confluent.io

@gamussa

viktor@confluent.io



<https://slackpass.io/confluentcommunity>

#connect #ksql

The Confluent logo is centered on the page. It consists of a stylized icon on the left, which is a cluster of horizontal lines of varying lengths, resembling a data stream or a network. To the right of this icon is the word "confluent" in a lowercase, sans-serif font.