package com.twitter.frigate.pushservice.predicate

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.util.CasLock

import com.twitter.frigate.common.util.CasSuccess

import com.twitter.frigate.pushservice.model.PushTypes.PushCandidate

import com.twitter.frigate.thriftscala.CommonRecommendationType

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.hermit.predicate.Predicate

import com.twitter.util.Duration

import com.twitter.util.Future

object CasLockPredicate {

def apply(

casLock: CasLock,

expiryDuration: Duration

)(

implicit statsReceiver: StatsReceiver

): NamedPredicate[PushCandidate] = {

val stats = statsReceiver.scope("predicate\_addcaslock\_for\_candidate")

Predicate

.fromAsync { candidate: PushCandidate =>

if (candidate.target.pushContext.exists(\_.darkWrite.exists(\_ == true))) {

Future.True

} else if (candidate.commonRecType == CommonRecommendationType.MagicFanoutSportsEvent) {

Future.True

} else {

candidate.target.history flatMap { h =>

val now = candidate.createdAt

val expiry = now + expiryDuration

val oldTimestamp = h.lastNotificationTime map {

\_.inSeconds

} getOrElse 0

casLock.cas(candidate.target.targetId, oldTimestamp, now.inSeconds, expiry) map {

casResult =>

stats.counter(s"cas\_$casResult").incr()

casResult == CasSuccess

}

}

}

}

.withStats(stats)

.withName("add\_cas\_lock")

}

}