package com.twitter.frigate.pushservice.predicate

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.frigate.common.base.TweetCandidate

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

import com.twitter.hermit.predicate.NamedPredicate

import com.twitter.hermit.predicate.tweetypie.EngagementsPredicate

import com.twitter.hermit.predicate.tweetypie.Perspective

import com.twitter.hermit.predicate.tweetypie.UserTweet

import com.twitter.storehaus.ReadableStore

object TargetEngagementPredicate {

val name = "target\_engagement"

def apply(

perspectiveStore: ReadableStore[UserTweet, Perspective],

defaultForMissing: Boolean

)(

implicit statsReceiver: StatsReceiver

): NamedPredicate[PushCandidate with TweetCandidate] = {

EngagementsPredicate(perspectiveStore, defaultForMissing)

.on { candidate: PushCandidate with TweetCandidate =>

UserTweet(candidate.target.targetId, candidate.tweetId)

}

.withStats(statsReceiver.scope(s"predicate\_$name"))

.withName(name)

}

}