package com.twitter.timelineranker.common

import com.twitter.servo.util.FutureArrow

import com.twitter.timelineranker.core.CandidateEnvelope

import com.twitter.timelineranker.core.HydratedTweets

import com.twitter.timelines.visibility.VisibilityEnforcer

import com.twitter.util.Future

/\*\*

\* Transform which uses an instance of a VisiblityEnforcer to filter down HydratedTweets

\*/

class VisibilityEnforcingTransform(visibilityEnforcer: VisibilityEnforcer)

extends FutureArrow[CandidateEnvelope, CandidateEnvelope] {

override def apply(envelope: CandidateEnvelope): Future[CandidateEnvelope] = {

visibilityEnforcer.apply(Some(envelope.query.userId), envelope.hydratedTweets.outerTweets).map {

visibleTweets =>

val innerTweets = envelope.hydratedTweets.innerTweets

envelope.copy(

hydratedTweets = HydratedTweets(outerTweets = visibleTweets, innerTweets = innerTweets))

}

}

}