package com.twitter.tweetypie.hydrator

import com.twitter.coreservices.IM2884

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.tweetypie.core.FilteredState

import com.twitter.tweetypie.core.ValueState

import com.twitter.stitch.Stitch

object IM2884FilterHydrator {

type Type = ValueHydrator[Unit, TweetCtx]

private val Drop =

Stitch.exception(FilteredState.Unavailable.DropUnspecified)

private val Success = Stitch.value(ValueState.unmodified(()))

def apply(stats: StatsReceiver): Type = {

val im2884 = new IM2884(stats)

ValueHydrator[Unit, TweetCtx] { (\_, ctx) =>

val userAgent = TwitterContext().flatMap(\_.userAgent)

val userAgentAffected = userAgent.exists(im2884.isAffectedClient)

val mightCrash = userAgentAffected && im2884.textMightCrashIOS(ctx.text)

if (mightCrash) Drop else Success

}

}

}