package com.twitter.visibility.interfaces.tweets.enrichments

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.visibility.builder.VisibilityResult

import com.twitter.visibility.results.richtext.PublicInterestReasonToPlainText

import com.twitter.visibility.rules.Action

import com.twitter.visibility.rules.ComplianceTweetNoticePreEnrichment

import com.twitter.visibility.rules.PublicInterest

import com.twitter.visibility.rules.Reason

object ComplianceTweetNoticeEnrichment {

val ComplianceTweetNoticeEnrichmentScope = "compliance\_tweet\_notice\_enrichment"

val ComplianceTweetNoticePreEnrichmentActionScope =

"compliance\_tweet\_notice\_pre\_enrichment\_action"

val englishLanguageTag = "en"

def apply(result: VisibilityResult, statsReceiver: StatsReceiver): VisibilityResult = {

val scopedStatsReceiver = statsReceiver.scope(ComplianceTweetNoticeEnrichmentScope)

val enrichedVerdict = enrichVerdict(result.verdict, scopedStatsReceiver)

result.copy(verdict = enrichedVerdict)

}

private def enrichVerdict(

verdict: Action,

statsReceiver: StatsReceiver

): Action = {

val preEnrichmentActionScope =

statsReceiver.scope(ComplianceTweetNoticePreEnrichmentActionScope)

verdict match {

case complianceTweetNoticePreEnrichmentVerdict: ComplianceTweetNoticePreEnrichment =>

preEnrichmentActionScope.counter("").incr()

val verdictWithDetailsAndUrl = complianceTweetNoticePreEnrichmentVerdict.reason match {

case Reason.Unspecified =>

preEnrichmentActionScope.counter("reason\_unspecified").incr()

complianceTweetNoticePreEnrichmentVerdict

case reason =>

preEnrichmentActionScope.counter("reason\_specified").incr()

val safetyResultReason = PublicInterest.ReasonToSafetyResultReason(reason)

val (details, url) =

PublicInterestReasonToPlainText(safetyResultReason, englishLanguageTag)

complianceTweetNoticePreEnrichmentVerdict.copy(

details = Some(details),

extendedDetailsUrl = Some(url))

}

verdictWithDetailsAndUrl.toComplianceTweetNotice()

case \_ => verdict

}

}

}