package com.twitter.visibility.rules.providers

import com.twitter.visibility.configapi.configs.VisibilityDeciderGates

import com.twitter.visibility.models.SafetyLevel

import com.twitter.visibility.rules.MixedVisibilityPolicy

import com.twitter.visibility.rules.RuleBase

import com.twitter.visibility.rules.generators.TweetRuleGenerator

class InjectedPolicyProvider(

visibilityDeciderGates: VisibilityDeciderGates,

tweetRuleGenerator: TweetRuleGenerator)

extends PolicyProvider {

private[rules] val policiesForSurface: Map[SafetyLevel, MixedVisibilityPolicy] =

RuleBase.RuleMap.map {

case (safetyLevel, policy) =>

(

safetyLevel,

MixedVisibilityPolicy(

originalPolicy = policy,

additionalTweetRules = tweetRuleGenerator.rulesForSurface(safetyLevel)))

}

override def policyForSurface(safetyLevel: SafetyLevel): MixedVisibilityPolicy = {

policiesForSurface(safetyLevel)

}

}