package com.twitter.visibility.configapi.configs

import com.twitter.decider.Decider

import com.twitter.servo.gate.DeciderGate

import com.twitter.servo.util.Gate

case class VisibilityDeciderGates(decider: Decider) {

import DeciderKey.\_

private[this] def feature(deciderKey: Value) = decider.feature(deciderKey.toString)

val enableFetchTweetReportedPerspective: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableFetchTweetReportedPerspective))

val enableFetchTweetMediaMetadata: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableFetchTweetMediaMetadata))

val enableFollowCheckInMutedKeyword: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.VisibilityLibraryEnableFollowCheckInMutedKeyword))

val enableMediaInterstitialComposition: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.VisibilityLibraryEnableMediaInterstitialComposition))

val enableExperimentalRuleEngine: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableExperimentalRuleEngine))

val enableLocalizedInterstitialGenerator: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableLocalizedInterstitialGenerator))

val enableShortCircuitingTVL: Gate[Unit] =

DeciderGate.linear(feature(EnableShortCircuitingFromTweetVisibilityLibrary))

val enableVerdictLoggerTVL = DeciderGate.linear(

feature(DeciderKey.EnableVerdictLoggerEventPublisherInstantiationFromTweetVisibilityLibrary))

val enableVerdictScribingTVL =

DeciderGate.linear(feature(DeciderKey.EnableVerdictScribingFromTweetVisibilityLibrary))

val enableBackendLimitedActions =

DeciderGate.linear(feature(DeciderKey.EnableBackendLimitedActions))

val enableMemoizeSafetyLevelParams: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableMemoizeSafetyLevelParams))

val enableShortCircuitingBVL: Gate[Unit] =

DeciderGate.linear(feature(EnableShortCircuitingFromBlenderVisibilityLibrary))

val enableVerdictLoggerBVL = DeciderGate.linear(

feature(DeciderKey.EnableVerdictLoggerEventPublisherInstantiationFromBlenderVisibilityLibrary))

val enableVerdictScribingBVL =

DeciderGate.linear(feature(DeciderKey.EnableVerdictScribingFromBlenderVisibilityLibrary))

val enableShortCircuitingSVL: Gate[Unit] =

DeciderGate.linear(feature(EnableShortCircuitingFromSearchVisibilityLibrary))

val enableVerdictLoggerSVL = DeciderGate.linear(

feature(DeciderKey.EnableVerdictLoggerEventPublisherInstantiationFromSearchVisibilityLibrary))

val enableVerdictScribingSVL =

DeciderGate.linear(feature(DeciderKey.EnableVerdictScribingFromSearchVisibilityLibrary))

val enableShortCircuitingTCVL: Gate[Unit] =

DeciderGate.linear(feature(EnableShortCircuitingFromTimelineConversationsVisibilityLibrary))

val enableVerdictLoggerTCVL = DeciderGate.linear(feature(

DeciderKey.EnableVerdictLoggerEventPublisherInstantiationFromTimelineConversationsVisibilityLibrary))

val enableVerdictScribingTCVL =

DeciderGate.linear(

feature(DeciderKey.EnableVerdictScribingFromTimelineConversationsVisibilityLibrary))

val enableCardVisibilityLibraryCardUriParsing =

DeciderGate.linear(feature(DeciderKey.EnableCardVisibilityLibraryCardUriParsing))

val enableConvosLocalizedInterstitial: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableConvosLocalizedInterstitial))

val enableConvosLegacyInterstitial: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableConvosLegacyInterstitial))

val disableLegacyInterstitialFilteredReason: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.DisableLegacyInterstitialFilteredReason))

val enableLocalizedInterstitialInUserStateLibrary: Gate[Unit] =

DeciderGate.linear(feature(DeciderKey.EnableLocalizedInterstitialInUserStateLib))

}