package com.twitter.visibility.util

import com.twitter.abdecider.ABDeciderFactory

import com.twitter.abdecider.LoggingABDecider

import com.twitter.decider.Decider

import com.twitter.decider.DeciderFactory

import com.twitter.decider.LocalOverrides

import com.twitter.logging.\_

object DeciderUtil {

val DefaultDeciderPath = "/config/com/twitter/visibility/decider.yml"

private val zone = Option(System.getProperty("dc")).getOrElse("atla")

val DefaultDeciderOverlayPath: Some[String] = Some(

s"/usr/local/config/overlays/visibility-library/visibility-library/prod/$zone/decider\_overlay.yml"

)

val DefaultABDeciderPath = "/usr/local/config/abdecider/abdecider.yml"

def mkDecider(

deciderBasePath: String = DefaultDeciderPath,

deciderOverlayPath: Option[String] = DefaultDeciderOverlayPath,

useLocalDeciderOverrides: Boolean = false,

): Decider = {

val fileBased = new DeciderFactory(Some(deciderBasePath), deciderOverlayPath)()

if (useLocalDeciderOverrides) {

LocalOverrides.decider("visibility-library").orElse(fileBased)

} else {

fileBased

}

}

def mkLocalDecider: Decider = mkDecider(deciderOverlayPath = None)

def mkABDecider(

scribeLogger: Option[Logger],

abDeciderPath: String = DefaultABDeciderPath

): LoggingABDecider = {

ABDeciderFactory(

abDeciderPath,

Some("production"),

scribeLogger = scribeLogger

).buildWithLogging()

}

}