package com.twitter.product\_mixer.component\_library.module

import com.twitter.decider.Decider

import com.twitter.decider.RandomRecipient

import com.twitter.finagle.thrift.ClientId

import com.twitter.finagle.thrift.service.Filterable

import com.twitter.finagle.thrift.service.ReqRepServicePerEndpointBuilder

import com.twitter.finatra.mtls.thriftmux.modules.MtlsClient

import com.twitter.inject.Injector

import com.twitter.inject.annotations.Flags

import com.twitter.inject.thrift.modules.ReqRepDarkTrafficFilterModule

import scala.reflect.ClassTag

class DarkTrafficFilterModule[MethodIface <: Filterable[MethodIface]: ClassTag](

implicit serviceBuilder: ReqRepServicePerEndpointBuilder[MethodIface])

extends ReqRepDarkTrafficFilterModule

with MtlsClient {

override protected def enableSampling(injector: Injector): Any => Boolean = \_ => {

val decider = injector.instance[Decider]

val deciderKey =

injector.instance[String](Flags.named("thrift.dark.traffic.filter.decider\_key"))

val fromProxy = ClientId.current

.map(\_.name).exists(name => name.contains("diffy") || name.contains("darktraffic"))

!fromProxy && decider.isAvailable(deciderKey, recipient = Some(RandomRecipient))

}

}