package com.twitter.timelineranker.config

import com.twitter.servo.util.Gate

import com.twitter.timelineranker.clients.ScopedCortexTweetQueryServiceClientFactory

import com.twitter.timelines.clients.gizmoduck.ScopedGizmoduckClientFactory

import com.twitter.timelines.clients.manhattan.ScopedUserMetadataClientFactory

import com.twitter.timelines.clients.socialgraph.ScopedSocialGraphClientFactory

import com.twitter.timelines.clients.strato.realgraph.ScopedRealGraphClientFactory

import com.twitter.timelines.clients.tweetypie.AdditionalFieldConfig

import com.twitter.timelines.clients.tweetypie.ScopedTweetyPieClientFactory

import com.twitter.timelines.visibility.VisibilityEnforcerFactory

import com.twitter.timelines.visibility.VisibilityProfileHydratorFactory

import com.twitter.tweetypie.thriftscala.{Tweet => TTweet}

class ClientWrapperFactories(config: RuntimeConfiguration) {

private[this] val statsReceiver = config.statsReceiver

val cortexTweetQueryServiceClientFactory: ScopedCortexTweetQueryServiceClientFactory =

new ScopedCortexTweetQueryServiceClientFactory(

config.underlyingClients.cortexTweetQueryServiceClient,

statsReceiver = statsReceiver

)

val gizmoduckClientFactory: ScopedGizmoduckClientFactory = new ScopedGizmoduckClientFactory(

config.underlyingClients.gizmoduckClient,

statsReceiver = statsReceiver

)

val socialGraphClientFactory: ScopedSocialGraphClientFactory = new ScopedSocialGraphClientFactory(

config.underlyingClients.sgsClient,

statsReceiver

)

val visibilityEnforcerFactory: VisibilityEnforcerFactory = new VisibilityEnforcerFactory(

gizmoduckClientFactory,

socialGraphClientFactory,

statsReceiver

)

val tweetyPieAdditionalFieldsToDisable: Seq[Short] = Seq(

TTweet.MediaTagsField.id,

TTweet.SchedulingInfoField.id,

TTweet.EscherbirdEntityAnnotationsField.id,

TTweet.CardReferenceField.id,

TTweet.SelfPermalinkField.id,

TTweet.ExtendedTweetMetadataField.id,

TTweet.CommunitiesField.id,

TTweet.VisibleTextRangeField.id

)

val tweetyPieHighQoSClientFactory: ScopedTweetyPieClientFactory =

new ScopedTweetyPieClientFactory(

tweetyPieClient = config.underlyingClients.tweetyPieHighQoSClient,

additionalFieldConfig = AdditionalFieldConfig(

fieldDisablingGates = tweetyPieAdditionalFieldsToDisable.map(\_ -> Gate.False).toMap

),

includePartialResults = Gate.False,

statsReceiver = statsReceiver

)

val tweetyPieLowQoSClientFactory: ScopedTweetyPieClientFactory = new ScopedTweetyPieClientFactory(

tweetyPieClient = config.underlyingClients.tweetyPieLowQoSClient,

additionalFieldConfig = AdditionalFieldConfig(

fieldDisablingGates = tweetyPieAdditionalFieldsToDisable.map(\_ -> Gate.False).toMap

),

includePartialResults = Gate.False,

statsReceiver = statsReceiver

)

val userMetadataClientFactory: ScopedUserMetadataClientFactory =

new ScopedUserMetadataClientFactory(

config.underlyingClients.manhattanStarbuckClient,

TimelineRankerConstants.ManhattanStarbuckAppId,

statsReceiver

)

val visibilityProfileHydratorFactory: VisibilityProfileHydratorFactory =

new VisibilityProfileHydratorFactory(

gizmoduckClientFactory,

socialGraphClientFactory,

statsReceiver

)

val realGraphClientFactory =

new ScopedRealGraphClientFactory(config.underlyingClients.stratoClient, statsReceiver)

}