Persuasion API

Master's Project Defense Presentation, Spring 2015

NAVEENRAJ NAGARATHINAM (NAGARATHINAM.1@OSU.EDU)

ADVISOR: DR. RAJIV RAMNATH

Presentation Overview

What is a Persuasive System

The Persuasion API

Theoretical Research

Implementation Plan Overview

The Current Implementation

Future Work

Persuasive Systems

Influencing user behavior through application systems

Based on user’s psychological response to system actions

Success is twofold-

Getting users to start using the system

- This is all about what the system has to offer
- And giving them a feel that the system will help them achieve their goal

Retain them, and actually help them achieve

- Motivating users to progress towards their(or the application’s) goal
- Making them realize how rewarding their experience has been

Persuasion API

Goal

- An API providing persuasion capabilities
- Generic; possible to integrate with most systems and be used readily

Approach

- Study existing work on psychology of persuasion and persuasive technology
- Identify persuasive techniques that applies to technology
- Research existing systems for application of those strategies
- Pick an initial set of most influential strategies
- Implement them in a generic API, integrate with a consumer and measure the impact

The Analysis

Persuasion strategies

- Identifying strategies by researching existing work on psychology of persuasion

Ontology of persuasion

- A holistic view of persuasion strategies and how they impact the user

Feedback Loop Model

- Making user action influence more of that same action

User Profiling

- Understanding user persuadability

Next

What do people expect?

Personal goal-setting

- People expect a system to allow them to have control over what they do

Suggestions/recommendations

- People expect applications to provide ways to achieve their goals

Reduction

- People incline towards applications that breaks up their target goal into simple incremental tasks

Tracking & feedback

- People look for ways to track their performance

Tailoring

- People expect system to present content only that are relevant to them(and more specifically at that point of time)

Facilitation

- People expect a system to provide a platform to collaborate and work towards a common goal

Back

What more can a system do?

Appreciate

- People like to be appreciated. It has a feel good factor and they like to consistently receive it

Tracking & Activity logs

- People like a way to track their performance through the process

Feedback

- People like to be consistent and like to know how they perform, whether good or bad

Timely Reminders

- Unless people are determined, they tend to forget to work towards target goal in their actions

Predictions/rehearsal

- Seeing their path to goal has a positive impact. It gives a sense of realistic achievability

Tunneling

- Long term attitude change is always incremental. The system should guide them step by step to get them closer to their goal

Back

Making content more acceptable

Consensus/ Social Proof

- People tend to accept and do what most others accept and do. This relies on people's sense of 'safety in numbers'

Liking

- People tend to be influenced by people they like. This can be their friends, some public figure they like, someone they trust

Expertise & Authority

- Expert opinions and opinions from people in authority are more likely to have an impact

Attitude of content

- There are differences in how people perceive positive and negative messages/enforcement. Positive enforcement is more likely to have effect

Simulation

- Content which shows the cause-effect relationship is more likely to have an impact

Tailoring & Personalization

- People pay attention to content (only) those that are relevant to their interests and more to those that are relevant at that point of time

Back

Social factors

Rewards

- Rewarding people makes them feel accomplished and motivates them to progress further. In some cases, this also improves their self-efficacy

Public recognition

- Everyone likes to be recognized for what they do. Again, this attacks the feed good factor of the person

Social comparison

- People like to compare their performance with others and this increases their motivation to perform to be better than others

Social competition

- This piggy backs on the previous concept, leveraging people's natural drive to compete. This increases the motivation to perform

Social influence

- People tend to change to meet the expectations of others. Peer pressure and motivation helps them perform better

Social collaboration

- Friends, forums, chat, etc.

Back

Learn, learn, learn…

From user’s goals

From user’s interests

From user’s actions

From user’s acceptance to content

Back

Implementation

Our initial focus – fitness applications

Partner – FitUprising

- Activity based weight management application

User appreciation and recognition to start with

- Appreciating the user for achieving something
- Recognizing the user by awarding badges for achieving something
- Recognizing the user by providing incentives for achieving something
- Recognizing the user publicly by posting their achievements to their network’s social feed

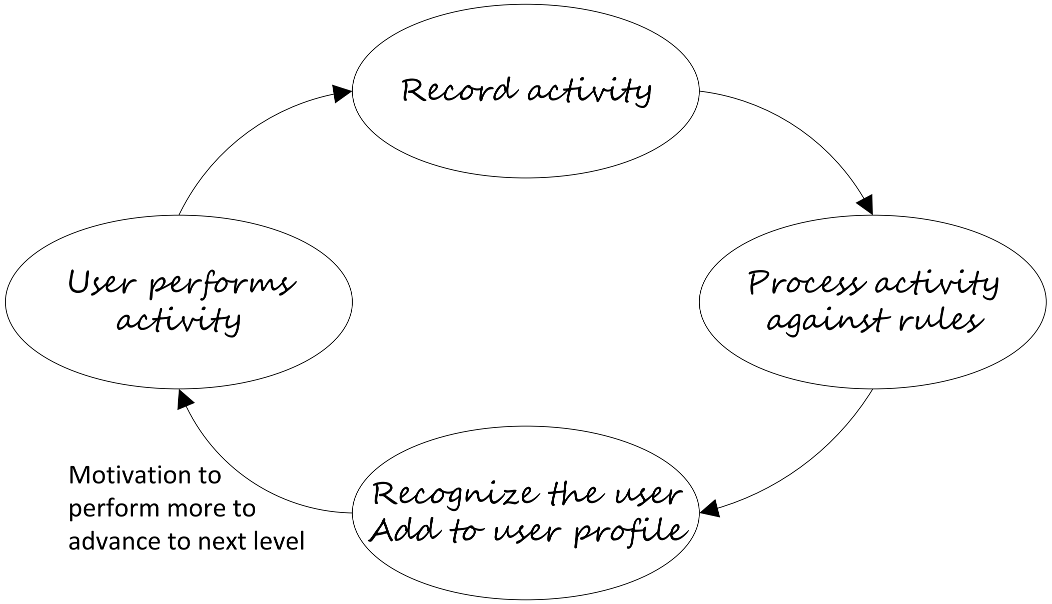
Badges and Recognition System

Rule based system; hierarchical rules with actions

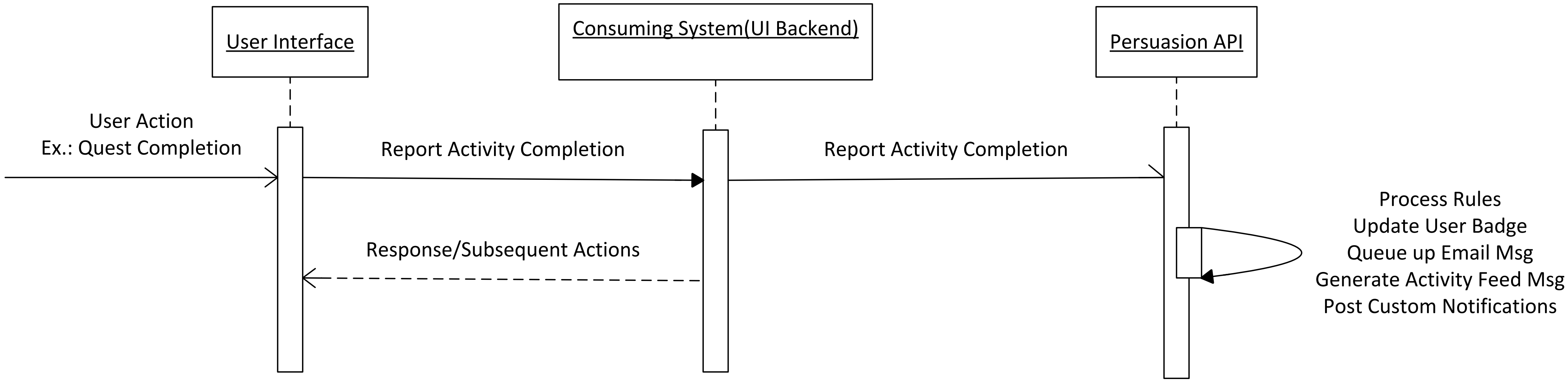
Rule Outcomes

- Assign a Badge to the user
- Send user an email message
- Post notification to network’s social feed
- Perform a custom action

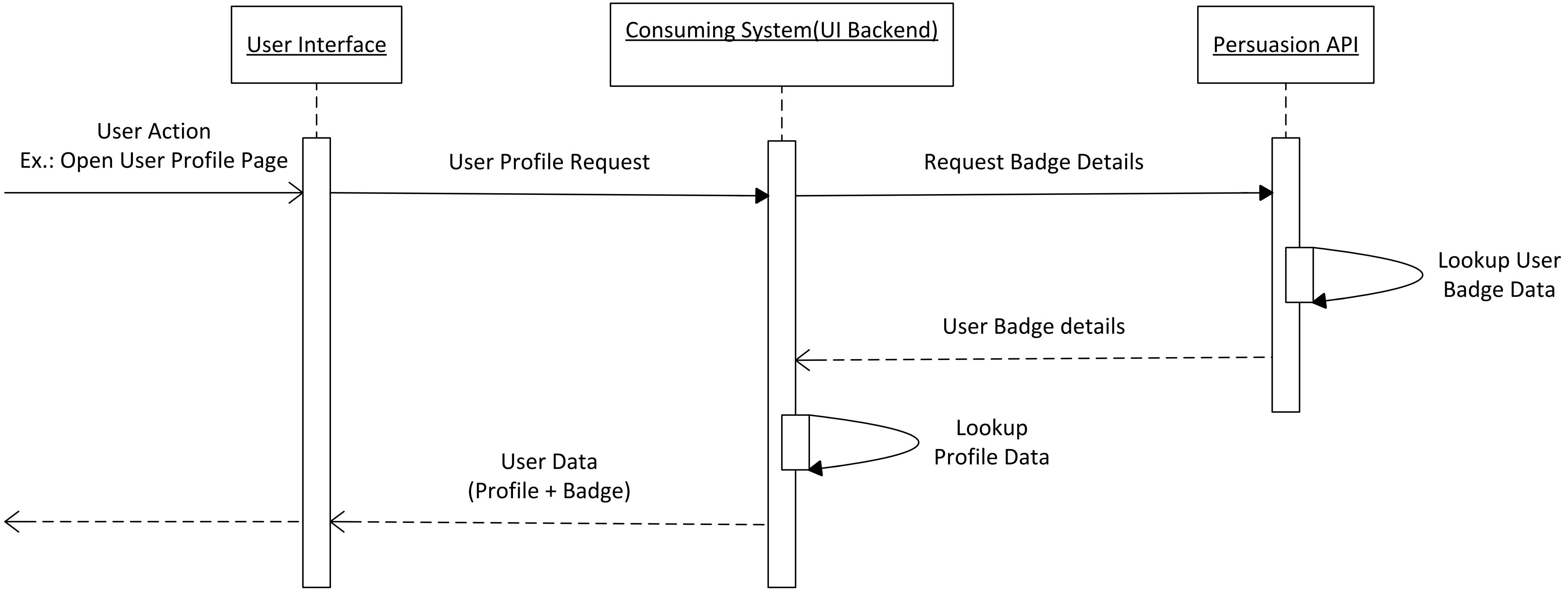
Feedback Loop – Recognition System


Record activity
Process activity against rules
Recognize the user
Add to user profile
User performs activity
Motivation to perform more to advance to next level

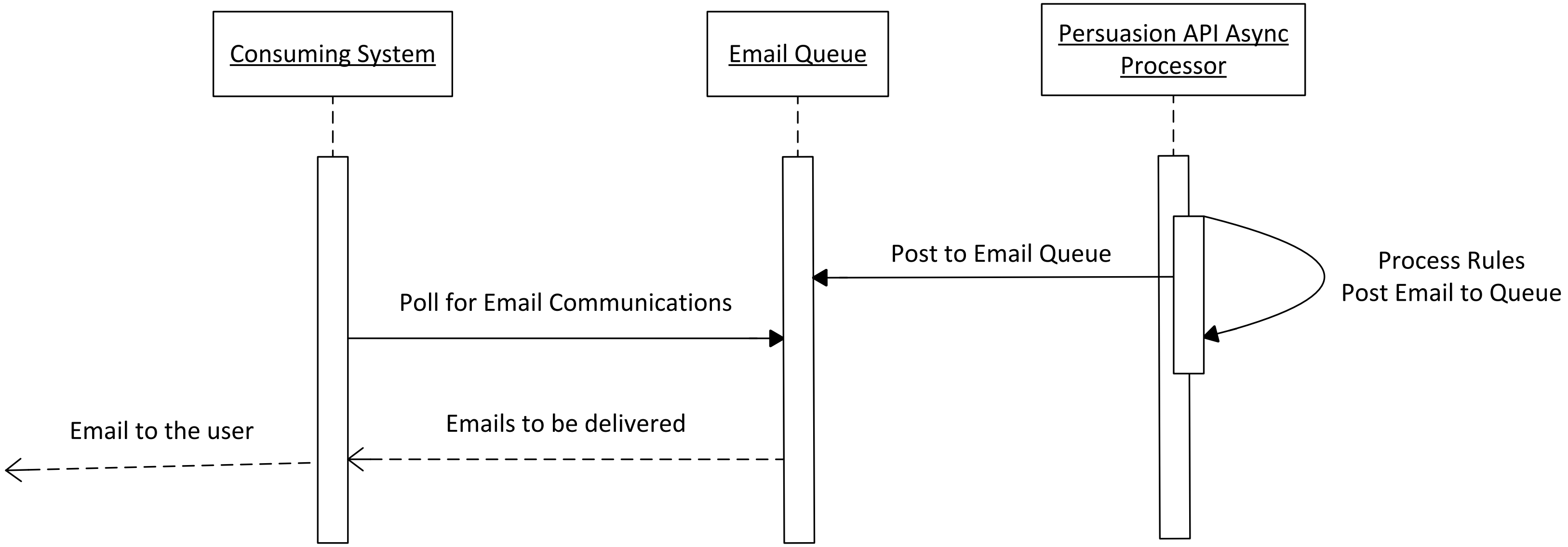
Client-API Interaction – Activity Reporting/Rule Processing


User Interface
Consuming System(UI Backend)
Persuasion API
User Action
Ex.: Quest Completion
Report Activity Completion
Report Activity Completion
Response/Subsequent Actions
Process Rules
Update User Badge
Queue up Email Msg
Generate Activity Feed Msg
Post Custom Notifications

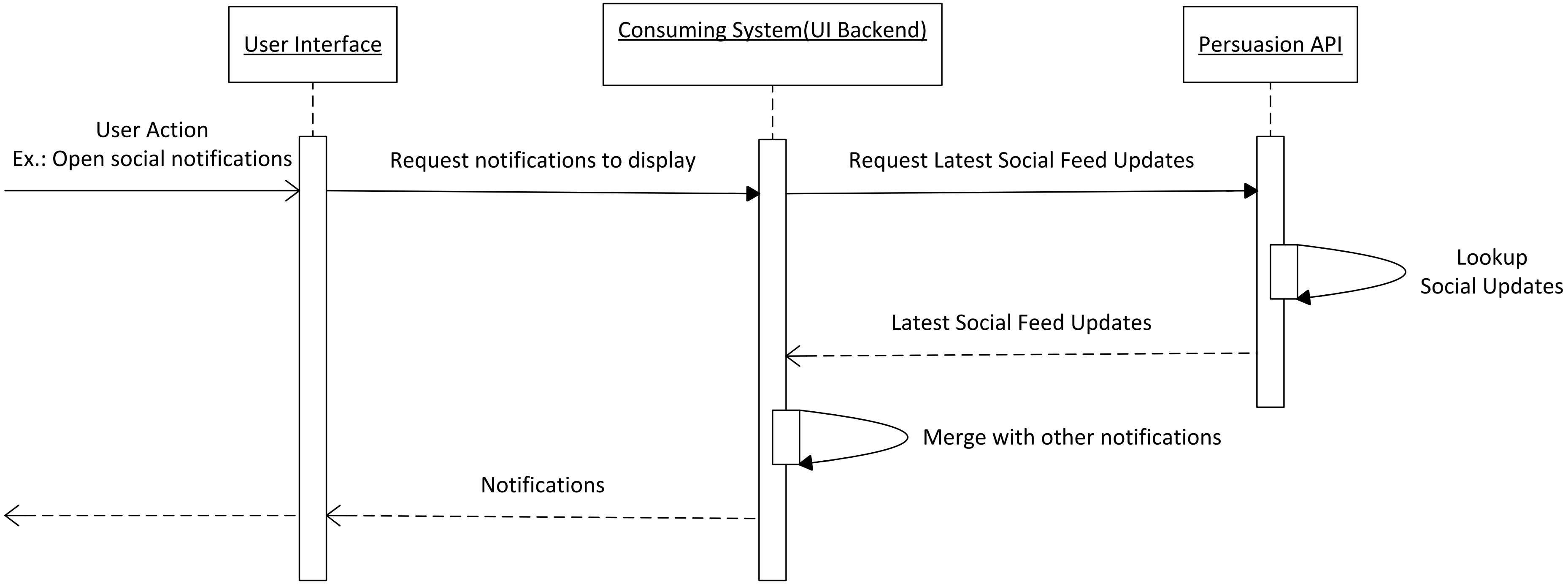
Client-API Interaction – User Badge Lookup



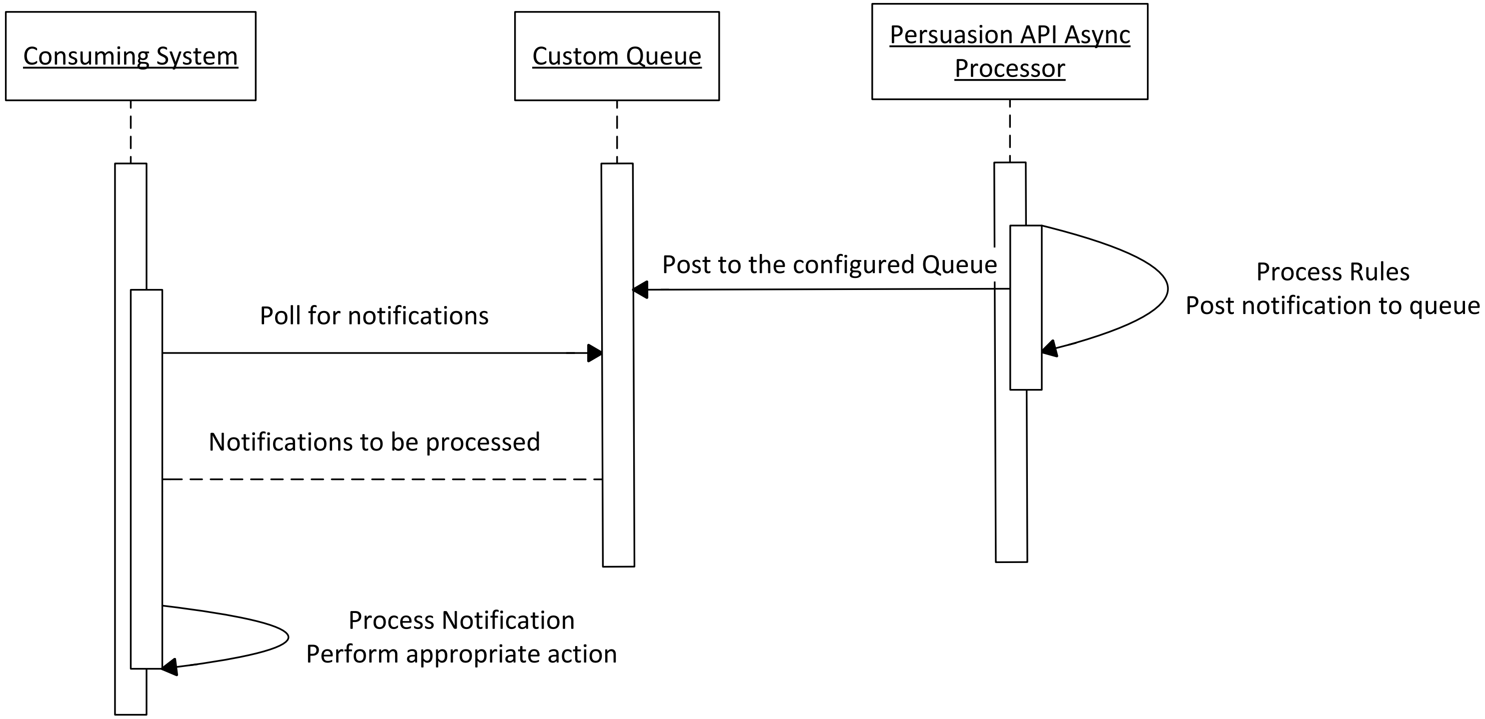
Client-API Interaction – User Emails



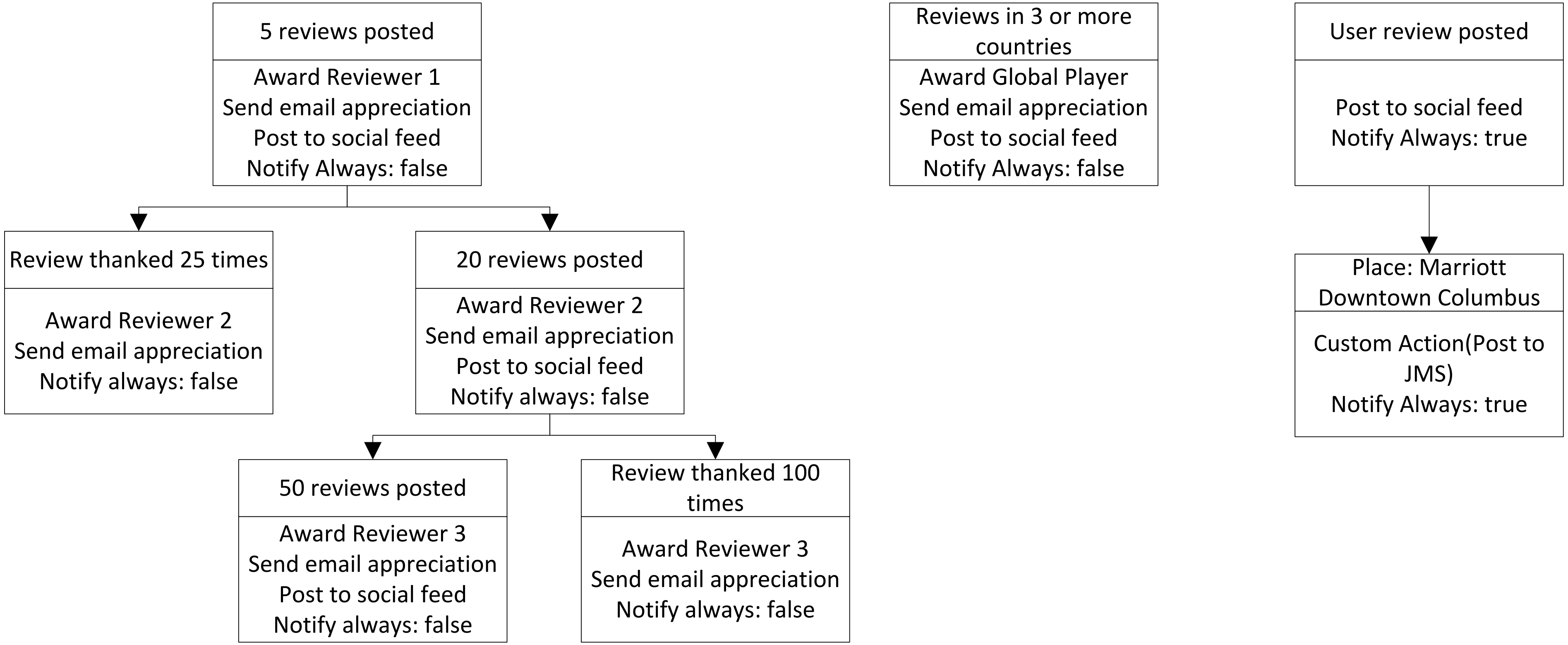
Client-API Interaction – Social Feed Highlights


User Interface
Consuming System(UI Backend)
Persuasion API
User Action
Ex.: Open social notifications
Request notifications to display
Request Latest Social Feed Updates
Lookup
Social Updates
Latest Social Feed Updates
Merge with other notifications
Notifications

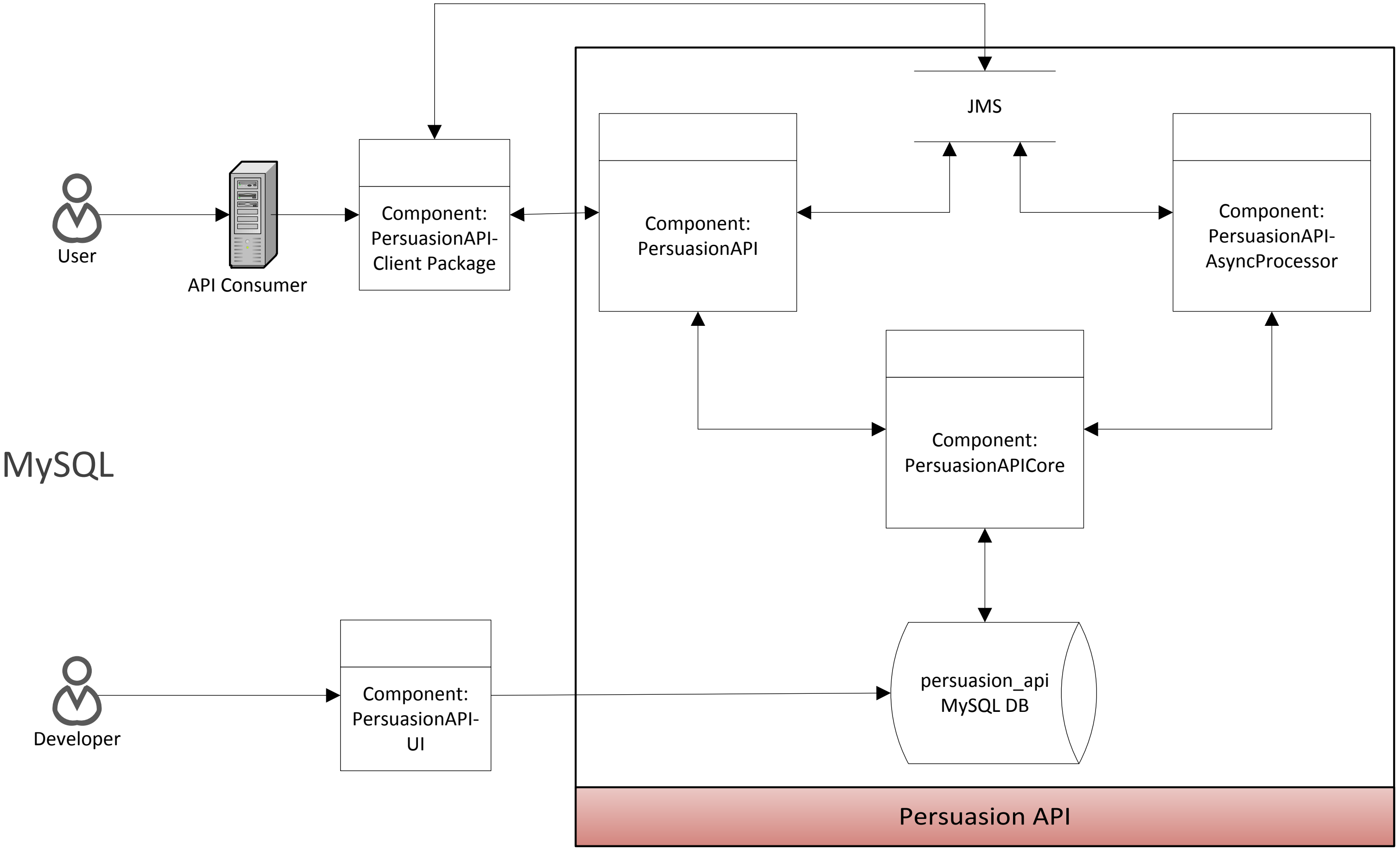
Client-API Interaction- Custom JMS Notifications


Consuming System
Custom Queue
Persuasion API Async Processor
Process Rules
Post notification to queue
Post to the configured Queue
Poll for notifications
Notifications to be processed
Process Notification
Perform appropriate action

Sample Rule Set (Travel Review Application)



System Architecture


JMS
User
API Consumer
Component: PersuasionAPI-Client Package
Component: PersuasionAPI
Component: PersuasionAPI-AsyncProcessor
Component: PersuasionAPICore
Developer
Component: PersuasionAPI-UI
persuasion_api MySQL DB
Persuasion API

J2EE Spring, MySQL

System Components

Persuasion API

- The services consumed by both the consuming system and the internal configuration UI

Persuasion API Core

- Contains core operations; interfaces to the database and JMS

Persuasion API Asynchronous Processor

- Processes rules, assigns badges, posts emails, JMS notifications

Persuasion API Client

- Used by consuming system - to call the API services, to provide implementations for custom badge actions

Persuasion API UI

- User Interface for configuring rules and actions

Future Work

Immediate next steps

- Complete implementing the User Interface
- Create documentation to consume the API

Content Acceptance – individual user based

- Use the questionnaire to understand user persuadability
- Provide add-on messages to improve content acceptance
- Allow a way for the consuming system to provide feedback
- Learn from feedback and adjust user persuasion strategy

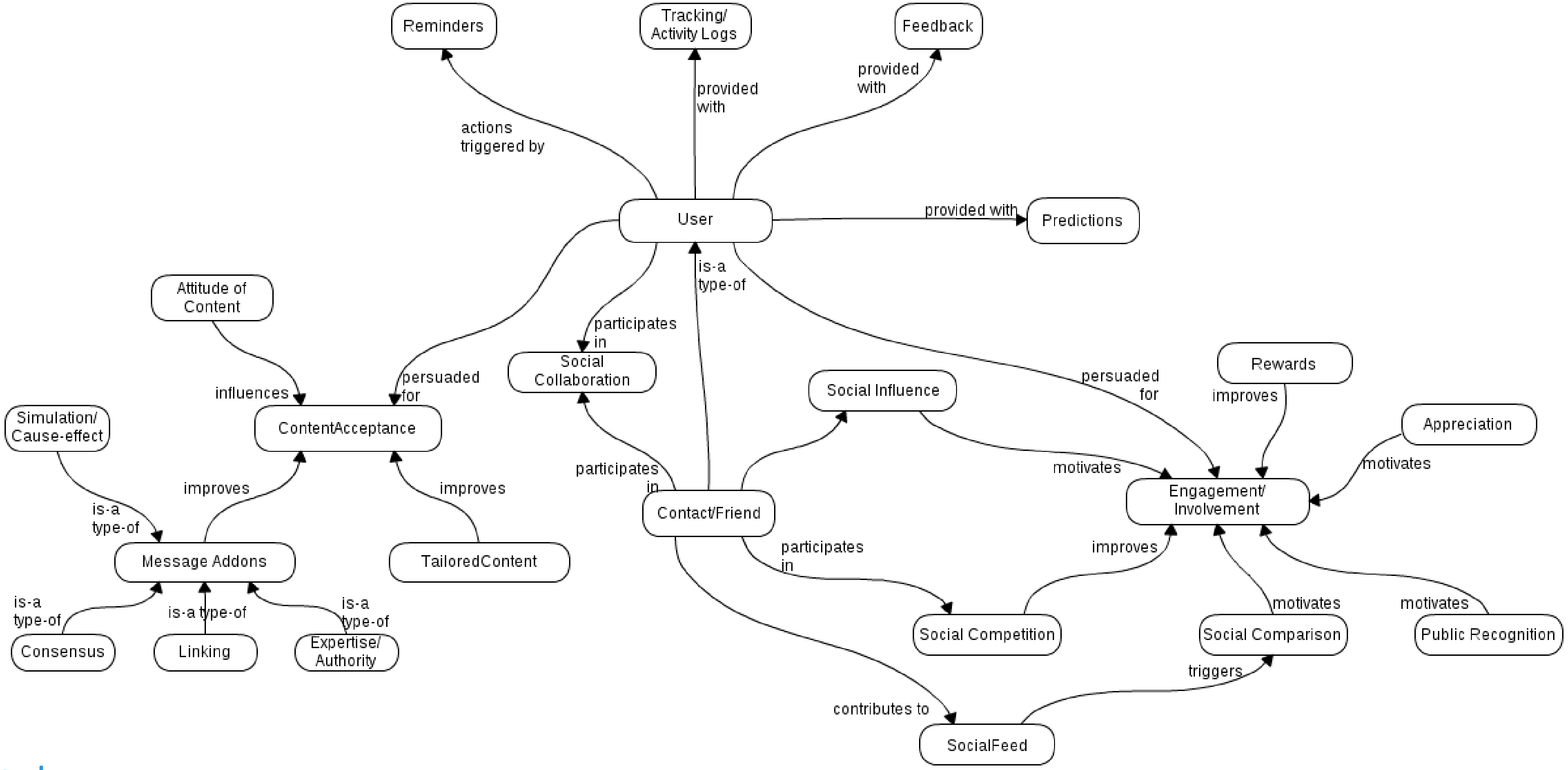
GitHub: https://github.com/naveenrajn/persuasionapi

Documentation to be added

Thank you!!

Backup

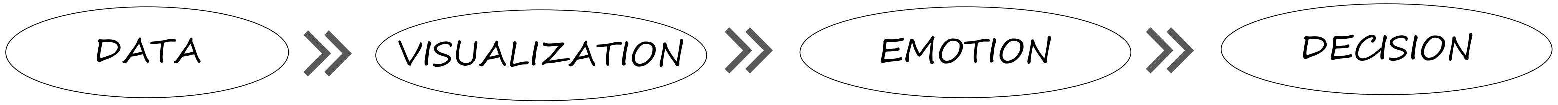
Persuasion Ontology

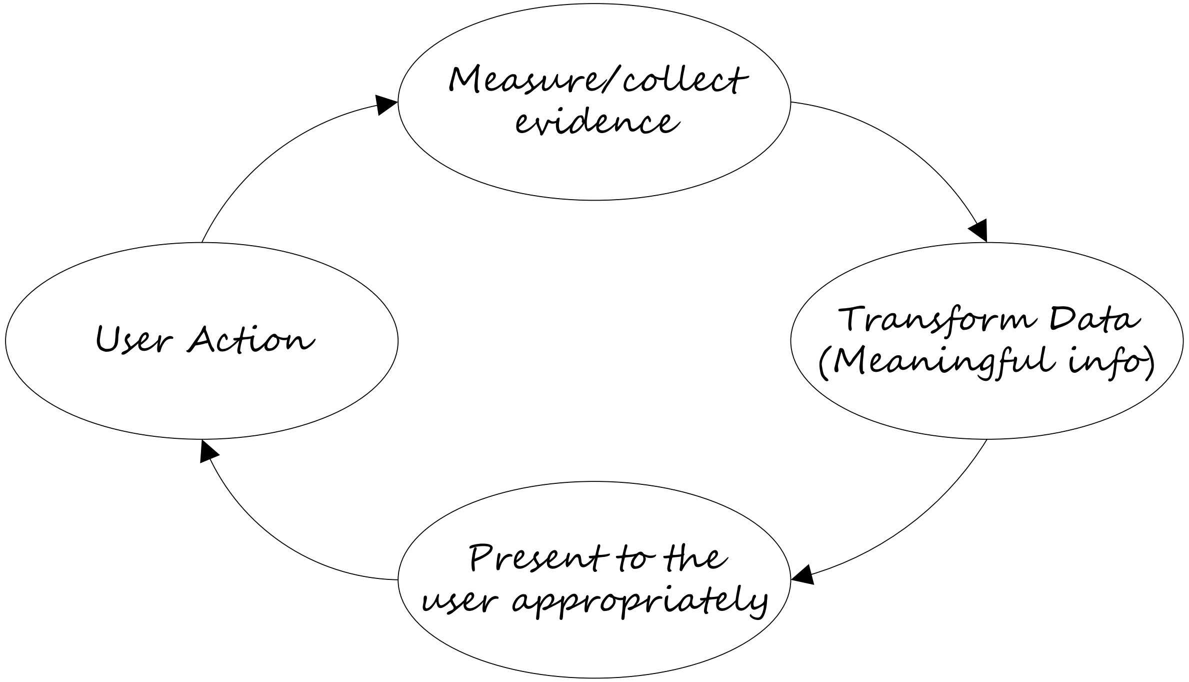


Link to the ontology

Back

Feedback Loop




Measure/collect evidence
Transform Data (Meaningful info)
Present to the user appropriately
User Action

Back

Initial User Profiling

To understand user persuadability

To identify the persuasion strategy the user is more susceptible to

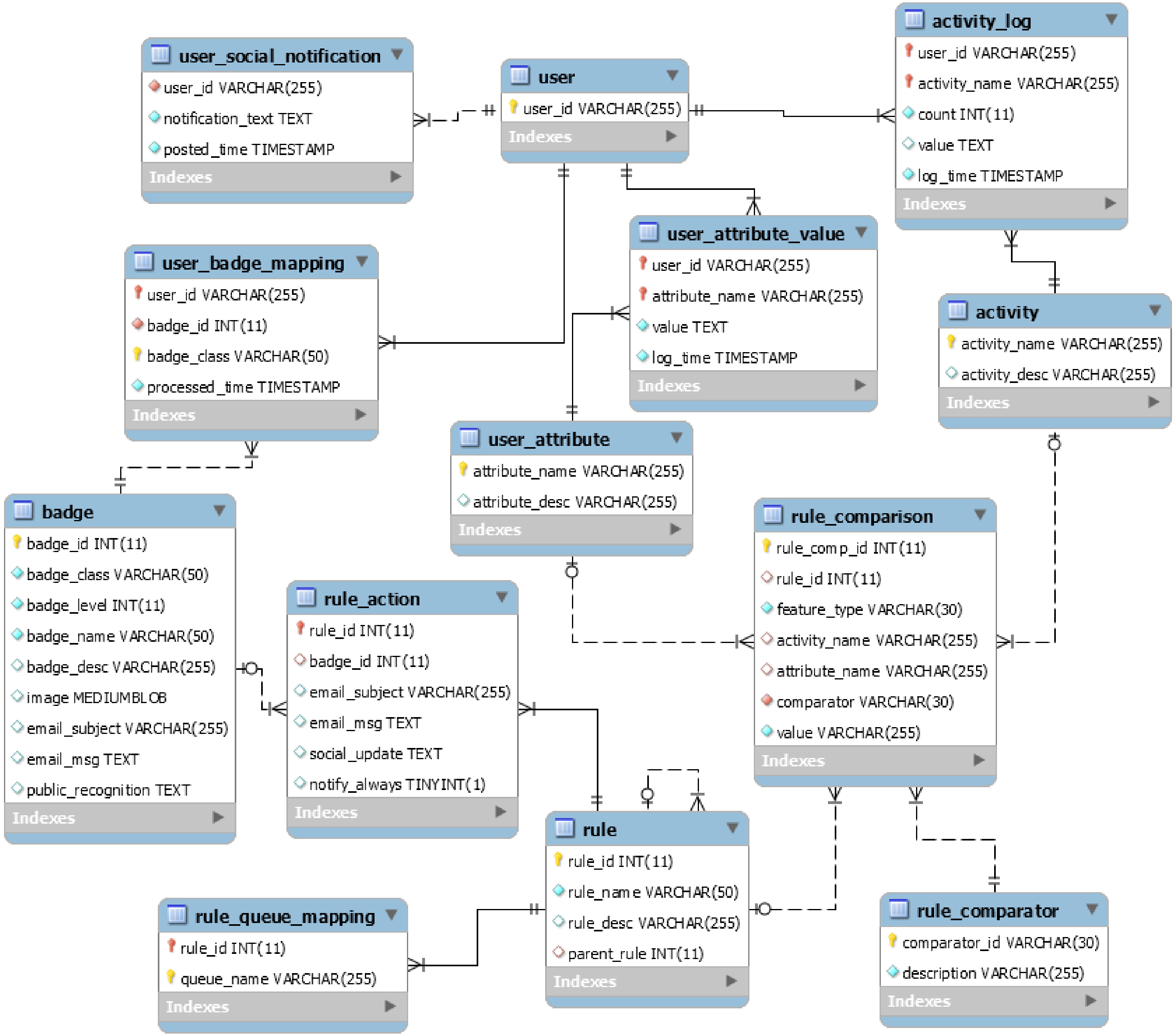
How? – A Questionnaire; can be provided to the user during registration as part of ‘getting to know the user’

Sample questions and candidate persuasive strategy if answered yes:

- When my professor tells me something, I tend to believe it is true – Expertise/Authority
- When I am in a new situation, I look at others to see what I should do – Consensus
- If someone from my social network notifies me about a good book, I tend to read it – Liking

Back

Data Model



user_social_notification
- user_id VARCHAR(255)
- notification_text TEXT
- posted_time TIMESTAMP
- Indexes

user
- user_id VARCHAR(255)
- Indexes

activity_log
- user_id VARCHAR(255)
- activity_name VARCHAR(255)
- count INT(11)
- value TEXT
- log_time TIMESTAMP
- Indexes

user_badge_mapping
- user_id VARCHAR(255)
- badge_id INT(11)
- badge_class VARCHAR(50)
- processed_time TIMESTAMP
- Indexes

user_attribute_value
- user_id VARCHAR(255)
- attribute_name VARCHAR(255)
- value TEXT
- log_time TIMESTAMP
- Indexes

activity
- activity_name VARCHAR(255)
- activity_desc VARCHAR(255)
- Indexes

user_attribute
- attribute_name VARCHAR(255)
- attribute_desc VARCHAR(255)
- Indexes

badge
- badge_id INT(11)
- badge_class VARCHAR(50)
- badge_level INT(11)
- badge_name VARCHAR(50)
- badge_desc VARCHAR(255)
- image MEDIUMBLOB
- email_subject VARCHAR(255)
- email_msg TEXT
- public_recognition TEXT
- Indexes

rule_action
- rule_id INT(11)
- badge_id INT(11)
- email_subject VARCHAR(255)
- email_msg TEXT
- social_update TEXT
- notify_always TINYINT(1)
- Indexes

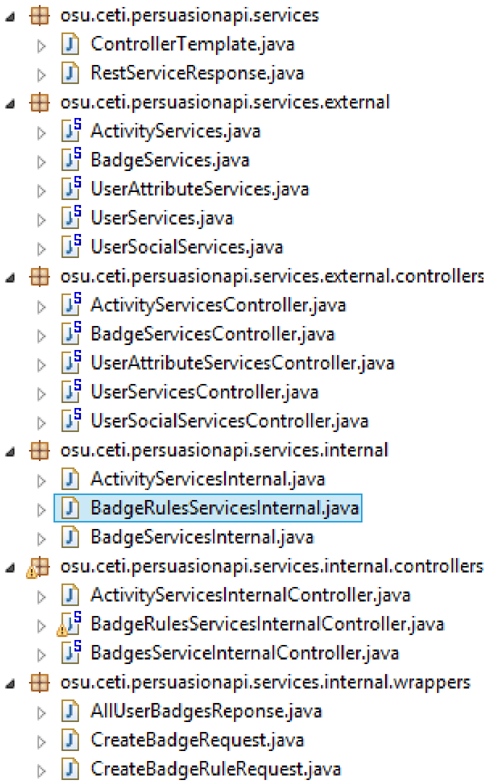
rule_comparison
- rule_comp_id INT(11)
- rule_id INT(11)
- feature_type VARCHAR(30)
- activity_name VARCHAR(255)
- attribute_name VARCHAR(255)
- comparator VARCHAR(30)
- value VARCHAR(255)
- Indexes

rule
- rule_id INT(11)
- rule_name VARCHAR(50)
- rule_desc VARCHAR(255)
- parent_rule INT(11)
- Indexes

rule_queue_mapping
- rule_id INT(11)
- queue_name VARCHAR(255)
- Indexes

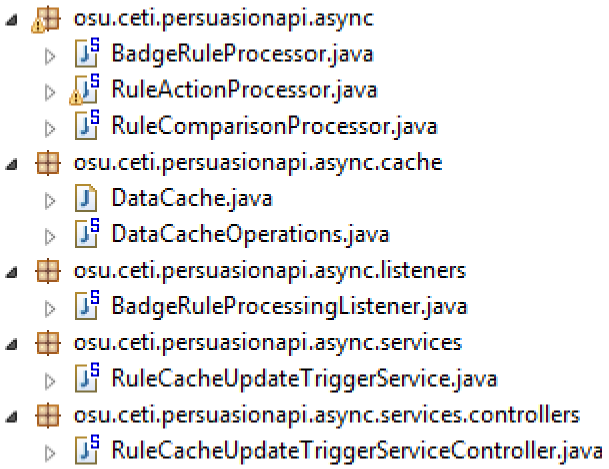
rule_comparator
- comparator_id VARCHAR(30)
- description VARCHAR(255)
- Indexes

Code Organization



```
osu.ceti.persuasionapi.services
    ControllerTemplate.java
    RestServiceResponse.java
osu.ceti.persuasionapi.services.external
    ActivityServices.java
    BadgeServices.java
    UserAttributeServices.java
    UserServices.java
    UserSocialServices.java
osu.ceti.persuasionapi.services.external.controllers
    ActivityServicesController.java
    BadgeServicesController.java
    UserAttributeServicesController.java
    UserServicesController.java
    UserSocialServicesController.java
osu.ceti.persuasionapi.services.internal
    ActivityServicesInternal.java
    BadgeRulesServicesInternal.java
    BadgeServicesInternal.java
osu.ceti.persuasionapi.services.internal.controllers
    ActivityServicesInternalController.java
    BadgeRulesServicesInternalController.java
    BadgesServiceInternalController.java
osu.ceti.persuasionapi.services.internal.wrappers
    AllUserBadgesReponse.java
    CreateBadgeRequest.java
    CreateBadgeRuleRequest.java
```

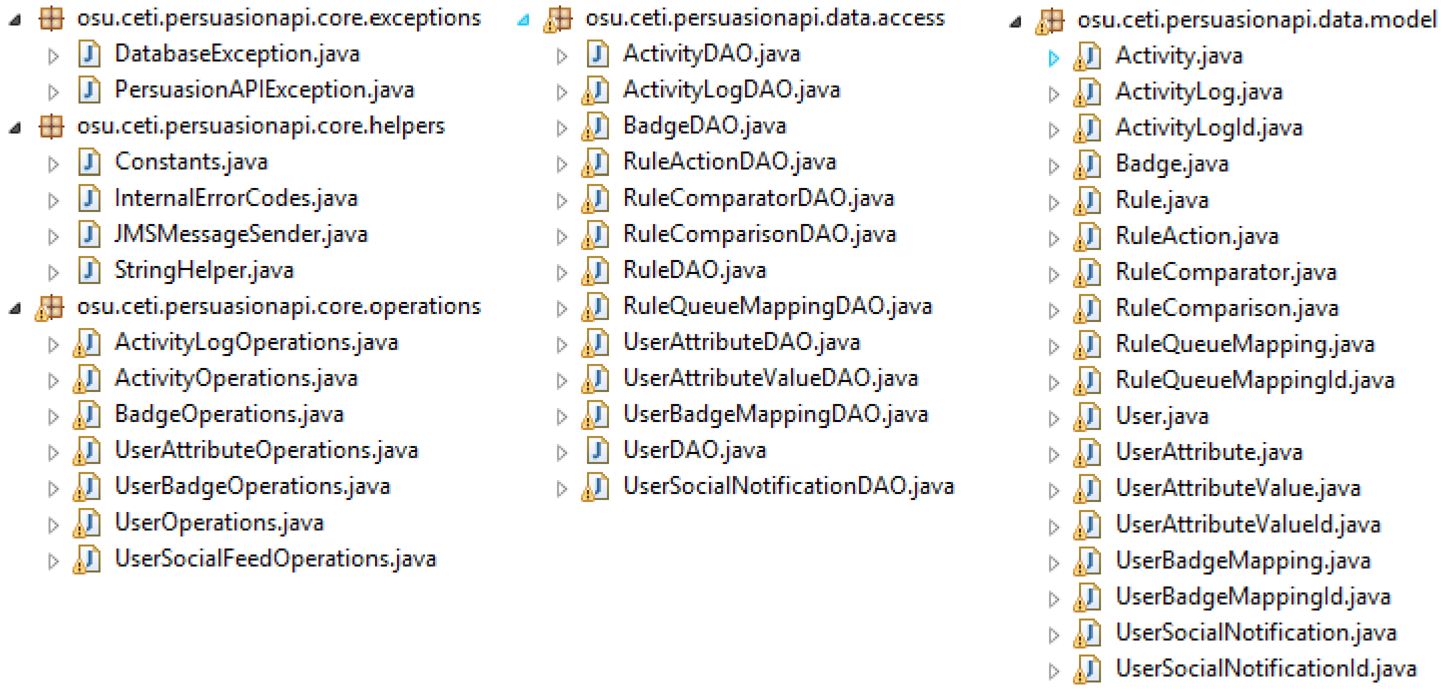
Persuasion API



```
osu.ceti.persuasionapi.async
    BadgeRuleProcessor.java
    RuleActionProcessor.java
    RuleComparisonProcessor.java
osu.ceti.persuasionapi.async.cache
    DataCache.java
    DataCacheOperations.java
osu.ceti.persuasionapi.async.listeners
    BadgeRuleProcessingListener.java
osu.ceti.persuasionapi.async.services
    RuleCacheUpdateTriggerService.java
osu.ceti.persuasionapi.async.services.controllers
    RuleCacheUpdateTriggerServiceController.java
```

Persuasion API Async Processor

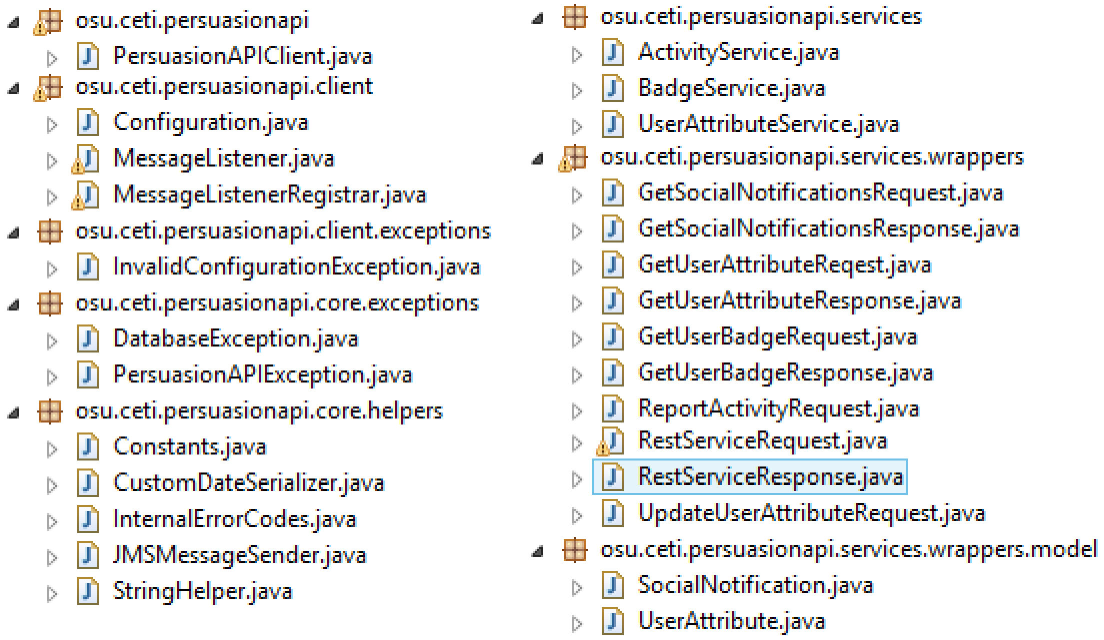
Code Organization (contd.)



- osu.ceti.persuasionapi.core.exceptions
  - DatabaseException.java
  - PersuasionAPIException.java
- osu.ceti.persuasionapi.core.helpers
  - Constants.java
  - InternalErrorCodes.java
  - JMSMessageSender.java
  - StringHelper.java
- osu.ceti.persuasionapi.core.operations
  - ActivityLogOperations.java
  - ActivityOperations.java
  - BadgeOperations.java
  - UserAttributeOperations.java
  - UserBadgeOperations.java
  - UserOperations.java
  - UserSocialFeedOperations.java
- osu.ceti.persuasionapi.data.access
  - ActivityDAO.java
  - ActivityLogDAO.java
  - BadgeDAO.java
  - RuleActionDAO.java
  - RuleComparatorDAO.java
  - RuleComparisonDAO.java
  - RuleDAO.java
  - RuleQueueMappingDAO.java
  - UserAttributeDAO.java
  - UserAttributeValueDAO.java
  - UserBadgeMappingDAO.java
  - UserDAO.java
  - UserSocialNotificationDAO.java
- osu.ceti.persuasionapi.data.model
  - Activity.java
  - ActivityLog.java
  - ActivityLogId.java
  - Badge.java
  - Rule.java
  - RuleAction.java
  - RuleComparator.java
  - RuleComparison.java
  - RuleQueueMapping.java
  - RuleQueueMappingId.java
  - User.java
  - UserAttribute.java
  - UserAttributeValue.java
  - UserAttributeValueId.java
  - UserBadgeMapping.java
  - UserBadgeMappingId.java
  - UserSocialNotification.java
  - UserSocialNotificationId.java

Persuasion API Core

Code Organization (contd.)



- osu.ceti.persuasionapi
  - PersuasionAPIClient.java
- osu.ceti.persuasionapi.client
  - Configuration.java
  - MessageListener.java
  - MessageListenerRegistrar.java
- osu.ceti.persuasionapi.client.exceptions
  - InvalidConfigurationException.java
- osu.ceti.persuasionapi.core.exceptions
  - DatabaseException.java
  - PersuasionAPIException.java
- osu.ceti.persuasionapi.core.helpers
  - Constants.java
  - CustomDateSerializer.java
  - InternalErrorCodes.java
  - JMSMessageSender.java
  - StringHelper.java
- osu.ceti.persuasionapi.services
  - ActivityService.java
  - BadgeService.java
  - UserAttributeService.java
- osu.ceti.persuasionapi.services.wrappers
  - GetSocialNotificationsRequest.java
  - GetSocialNotificationsResponse.java
  - GetUserAttributeReqest.java
  - GetUserAttributeResponse.java
  - GetUserBadgeRequest.java
  - GetUserBadgeResponse.java
  - ReportActivityRequest.java
  - RestServiceRequest.java
  - RestServiceResponse.java
  - UpdateUserAttributeRequest.java
- osu.ceti.persuasionapi.services.wrappers.model
  - SocialNotification.java
  - UserAttribute.java

Persuasion API Client