

ECE9607: CDS: Principles, Methodologies & Applications

Smart Sales in Pharmacy

Technical Report

Xinyu Yun

Department of Electrical & Computer Engineering
Western University

Date: 2015/8/25

Table of Content

1	Introduction	3
2	Literature Review	3
3	Technical Issues and Analysis.....	4
4	Proposed Solution.....	6
5	Results and Validation	13
6	Conclusion and Feature Work	17
7	References.....	17

Abstract

1 Introduction

CLM (closed loop marketing) is now widespread throughout the pharmacy sales industry. One of the main problems the pharmacy company need solve is how to close the loop in sales activities. The three key ‘pillars’ in CLM are: personalization/customization, Analytics, and responsiveness.

Functionally, the project will provide the specific analysis and design to propose features on how to ‘close the loop’ in CRM system. The core work is mapping the functional specifications to CDS (Cooperative Distributed System) smart Internet-of-Things, including creating multiple agents to handle part of the problems, design cooperation and communication between agents with reasonable privacy concerns, and automatic system functions in open environment.

Technically, the project focuses on the interaction protocol design and implementation on open platform. As a group in CDS, we choose DEX platform and Heroku cloud as the implementation solution. Basically the Node.js will be applied to implement the web server solution and deployed to Heroku cloud platform. The deployed web service and DB will be registered in DEX platform, the agents will interact and communicate with others through the DEX.

2 Literature Review

- 1) <Multichannel Closed Loop Marketing Digitally Transforming the Life Sciences Industry> Tim Moore, Hala Qanadilo

This paper mainly talks about the CLM and MCM solution in pharmacy sales world, based on which we can consider how to design a good feature to close the loop in sales and enhance the current system.

- 2) <Privacy in Cooperative Distributed Systems: Modeling and Protection Framework> Afshan Samani 2015

This paper focuses on the privacy perspective to research and design privacy model and framework in CDS, leading us to learn the basic theory of CDS, the privacy concerns of modeling “smart content”, and how to build privacy-based interaction protocol.

3 Technical Issues and Analysis

3.1 Technical issues

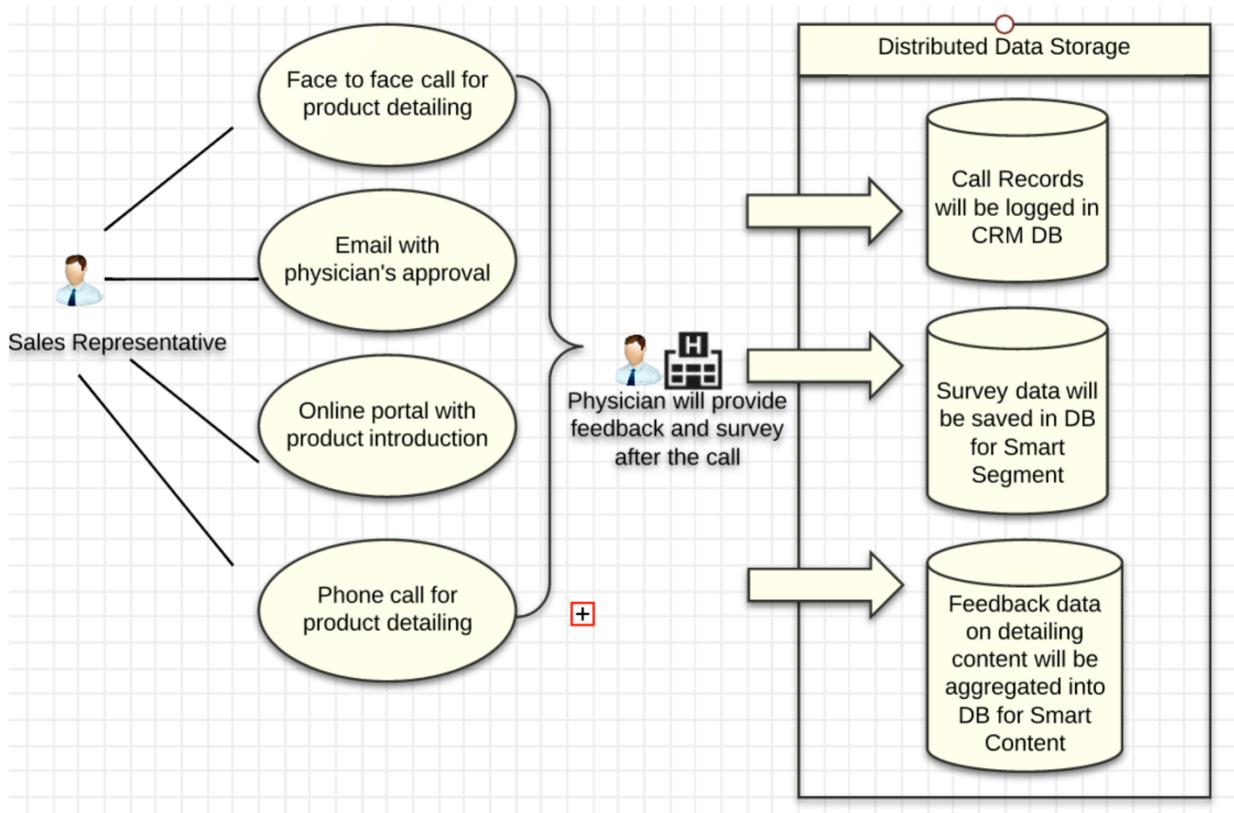
Firstly, the knowledge and behavior for each “smart object” should be identified and designed.

Secondly, the interaction and communication protocols between “smart objects” should be defined due to the business scenarios and technical platform we have chosen.

On the other hand, the privacy issue will be addressed during the interaction implementation, meaning to define the information boundary in each entity, identify the operation turning implicit to explicit in technical ways.

3.2 Scenarios to implement

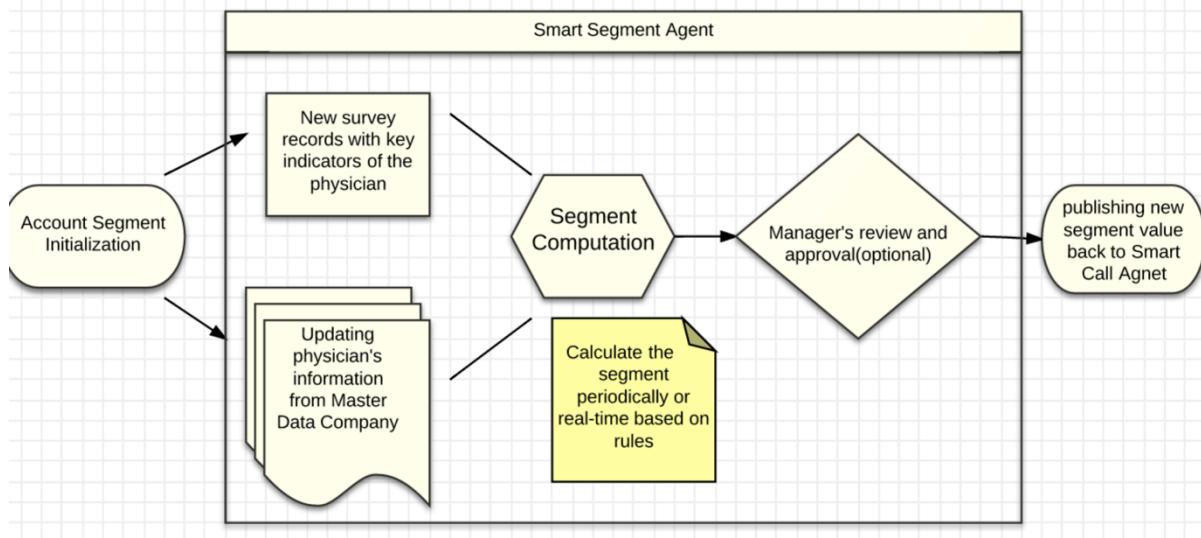
3.2.1 Smart Call Agent



Smart Call Agent Behaviors:

- 1) The call record will be saved in CRM (Smart Call Platform) firstly, once the feedback on detailing content or physician' survey has been collected and recorded in system, the sales representative have to submit the record
- 2) The survey data for segmentation will be synced or sent through request to Smart Segment Platform

3.2.2 Smart Segment Agent



Smart Segment Agent Behaviors:

- 1) Once the new survey data is synced from Smart Call Agent, a new computation process will be triggered to calculate the new segment, or the agent will collect all new surveys periodically to compute in a batch processes.
- 2) The new segment calculation will be resulted from survey, questionnaire and master data purchased from other company
- 3) Smart Segment Agent should support distributed data sources, automatically execution in specific interval, and allowing building internal approval process like “email notice to manager” before publishing the new segment.
- 4) The segment result will be published back to Smart Call Agent.

3.3 Non-Functional Descriptions

Privacy Compliance:

- 1) Each representative will only see his own call record or shared call from others to physician he covers
- 2) District managers will see all records created by representatives who are under their organization hierarchy.
- 3) Sales Representative will see the segment update status (initializing, checking, approved and so on) and will see the update values only when it is approved.
- 4) Different Sales Representatives will share the same segment if they cover the same physician.
- 5) Sales Representatives will only see the materials related products he covers
- 6) The smart segment agent will only view the related survey and account information, the other implicit information in smart call agent will be not revealed to other agents.

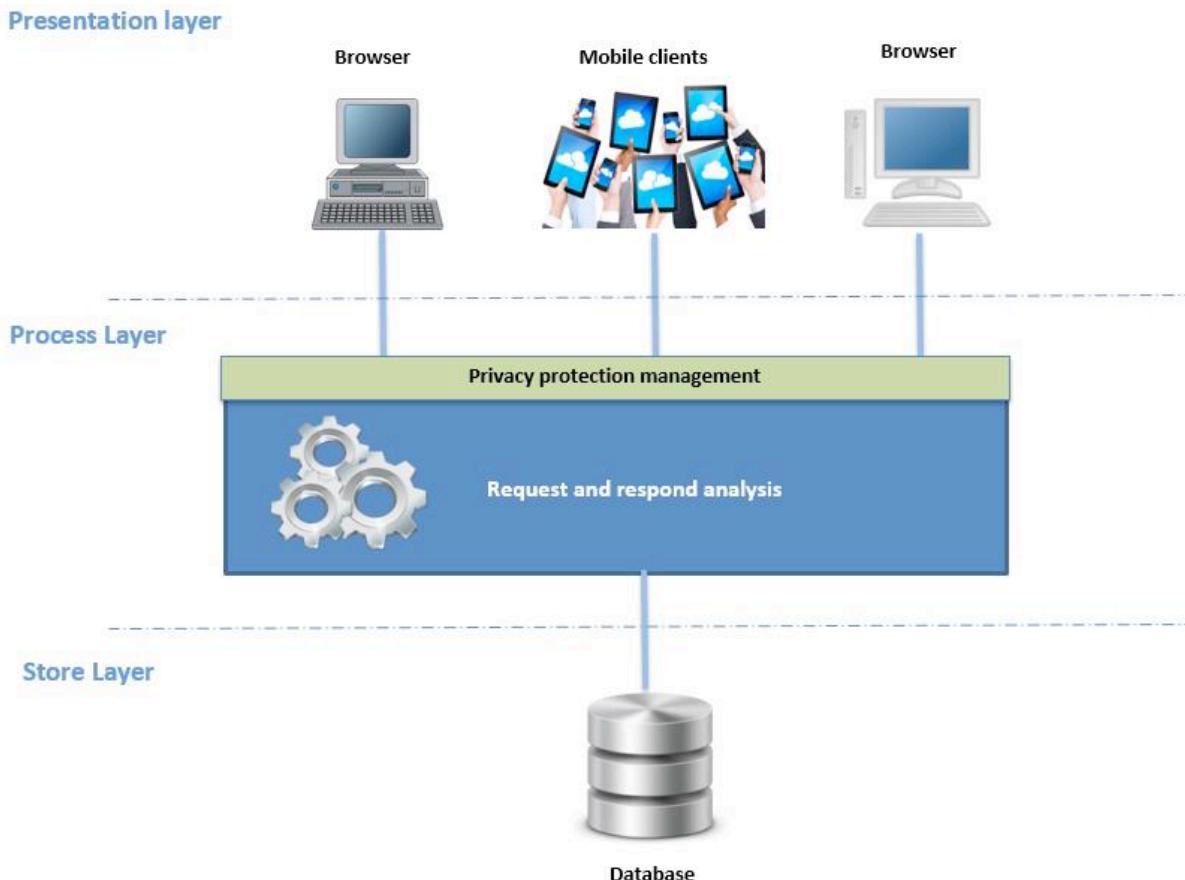
4 Proposed Solution

4.1 Proposed Technical Architecture

The typical three layers architecture will be applied in this project, the data storage layer will introduce the cloud service or local DB due to the performance requirement.

The agent layer, the core layer will be implemented in DEX's cloud service through 'Knowledge Broker', 'Service Broker' to fetch the data and interaction functions from distributed sources; the data and services will be integrated and deployed in Heroku platform.

The UI layer should involve different devices and channels to show the result of problems correctly through the DEX's touch point module.

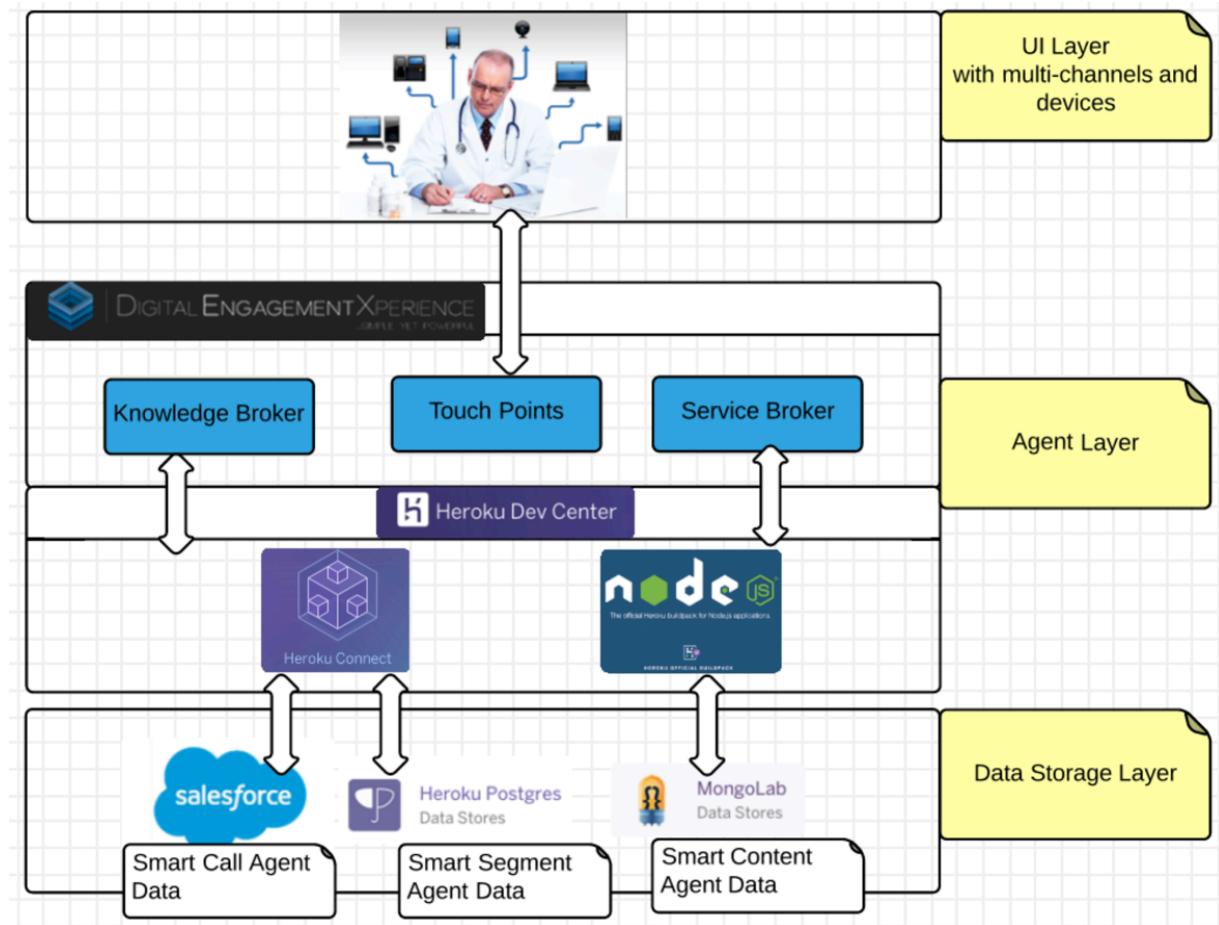


Referring to the deployment, firstly we build web service in node.js environment and create DBs in Heroku platform as the basic services and data sources.

Secondly all the web services will be deployed and running in Heroku platform, where the service information will be unified.

Thirdly the DBs and web services will be registered in DEX platform, where the data format from different DB and request format with token security will be unified. The "smart objects" will be generated to deal with the different scenarios based on DEX platform.

Finally I choose Salesforce.com platform as the “smart client” to let users request services and interact with other “smart object”, meanwhile, the user privacy issue will be taken care of through the Salesforce features and data privacy during the interaction will be addressed through the mapping of “Heroku Connect”, an add-on service to keep synchronization between Salesforce and Heroku Postgres DB.



4.2 Smart Objects/Agents Design

4.2.1 Smart Call Agent

Knowledge: physician information in Salesforce platform

Problem Solver: make sure all the call records from multi-channel can be logged in system.

Once the feedback on contents or survey is submitted, the interaction process will begin.

4.2.2 Segment Management Agent

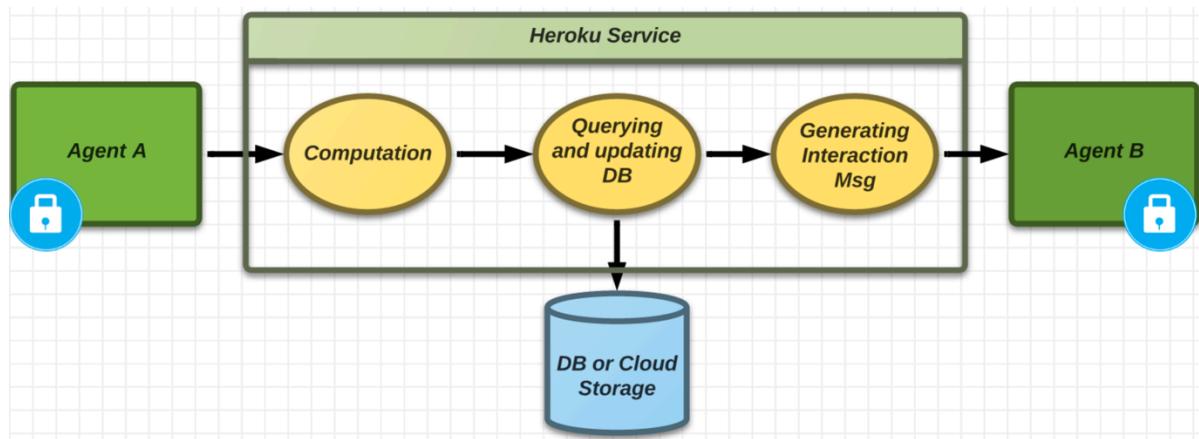
Knowledge: the rules to calculate segment

Problem Solver: Computing the segment automatically, getting approval from managers for segment update, publish the new segment to sales representative

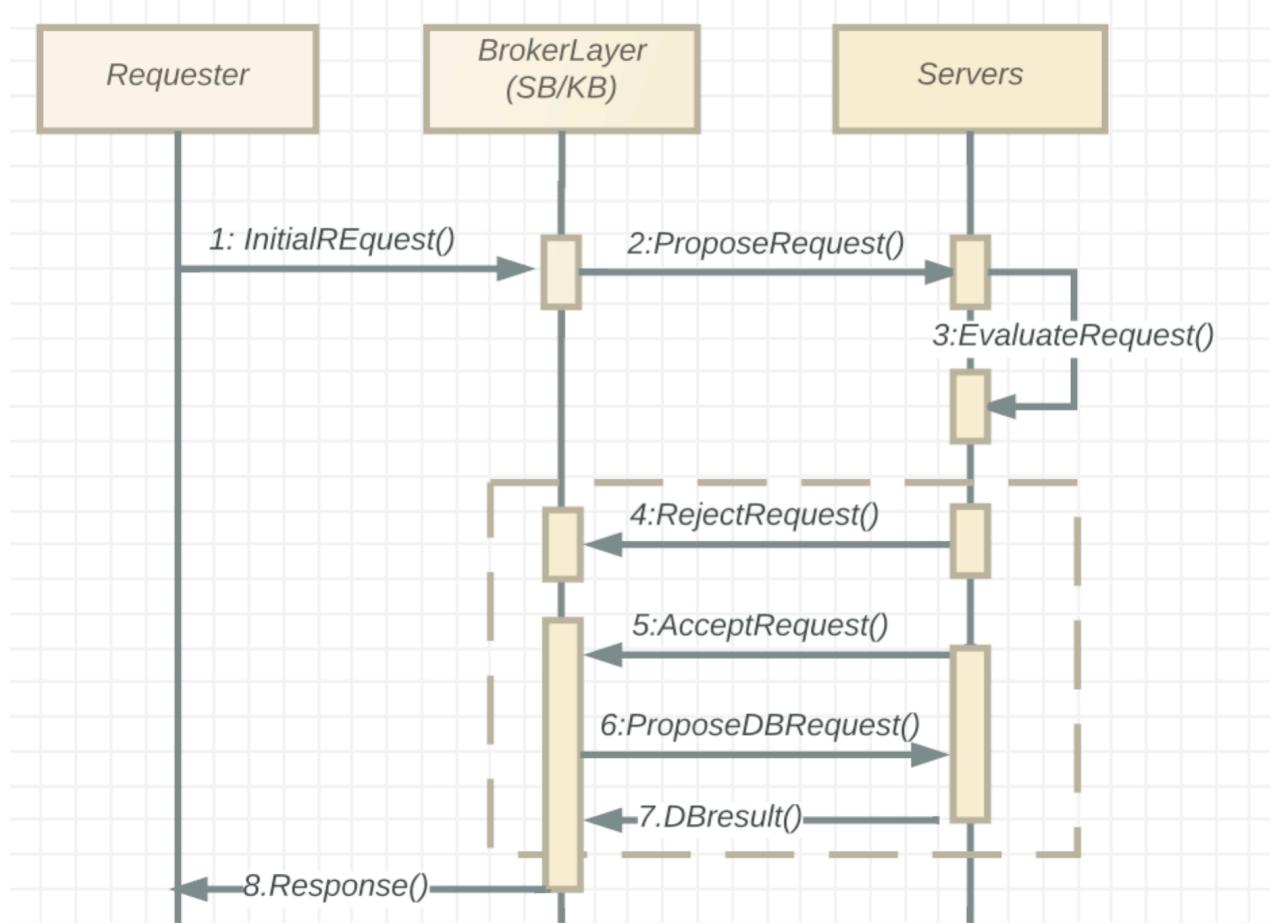
Once the update or new survey is received from Smart Call Agent, the interaction process will be started.

4.2.3 Interaction Architecture and Protocol

Interaction module is the key part in system implementation, which should involve the cooperative operations among different agents, querying and calculating the data



Proposed Interaction Protocol Patten:

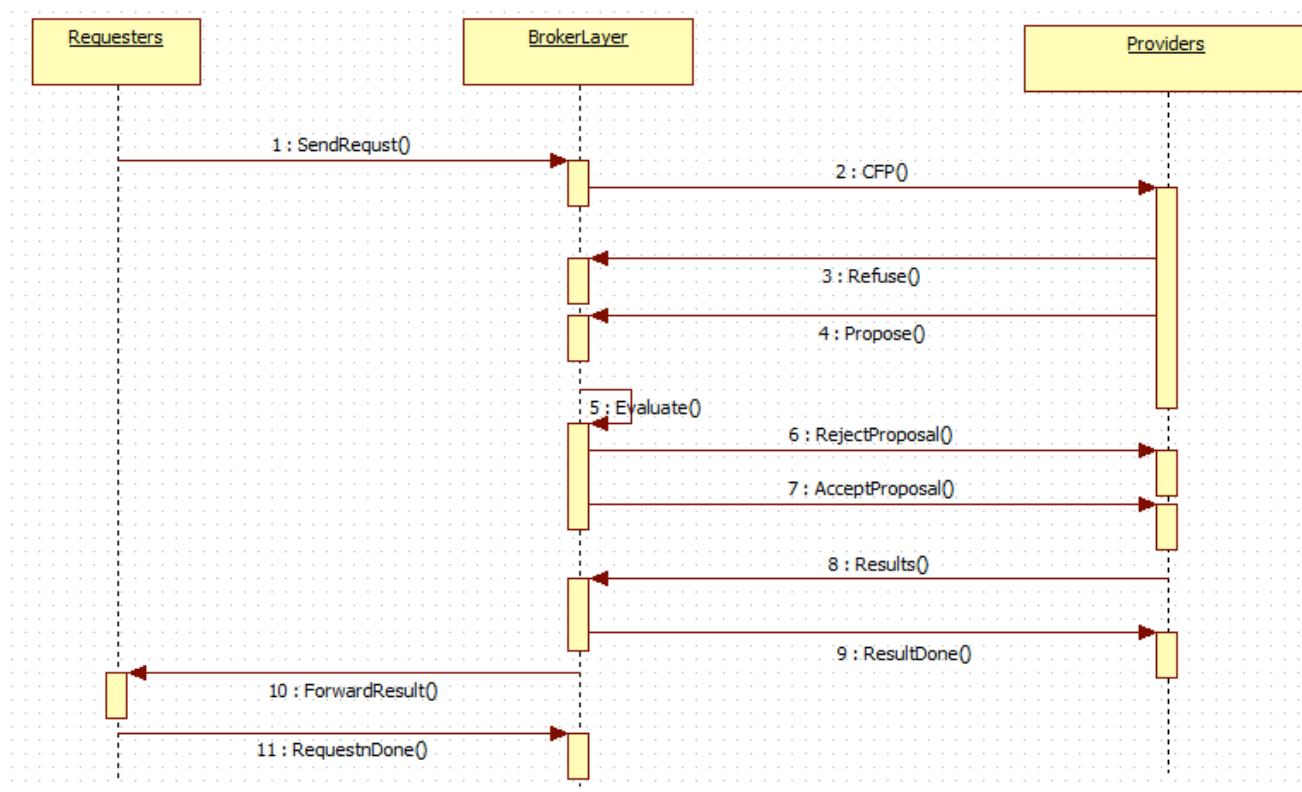


- 1) The services are running on different distributed servers deployed in Heroku and categorized into multiple “smart contents”.
- 2) All the services and DBs are registered in Broker Layer (DEX platform) so that all the requesters do not need to know the servers’ location and information, but use unify format to send request
- 3) The SB (Service Broker) will transfer the request to specific server, once the server receives the request, the evaluation process will start to check the privacy or other concerns for current “smart content”
- 4) If the evaluation is failed or request is wrong for other reasons, reject message will respond to requester through SB or directly.
- 5) If the server accepts the request and the request involves DB operation, the DB execution function will be requested through KB(Knowledge Broker) which also applies unify request format so that “smart content” does not need to know the DB’s type or location.

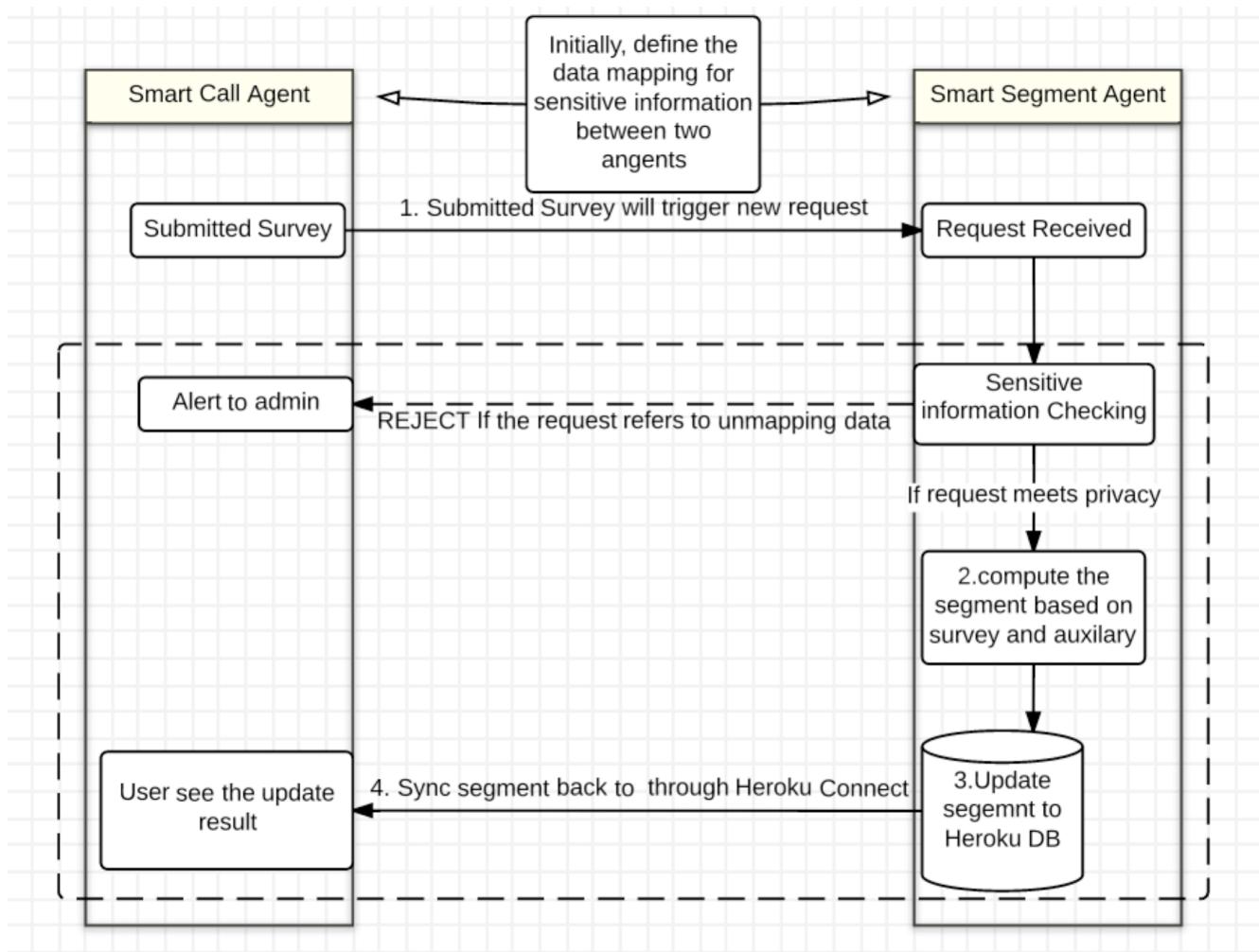
Interaction Protocol with negotiation process

As an extension of current interaction protocol, when multiple potential “smart objects” are existing in the open environment, the negotiation process

will be involved before handling the request to decide which one should take care of this request:

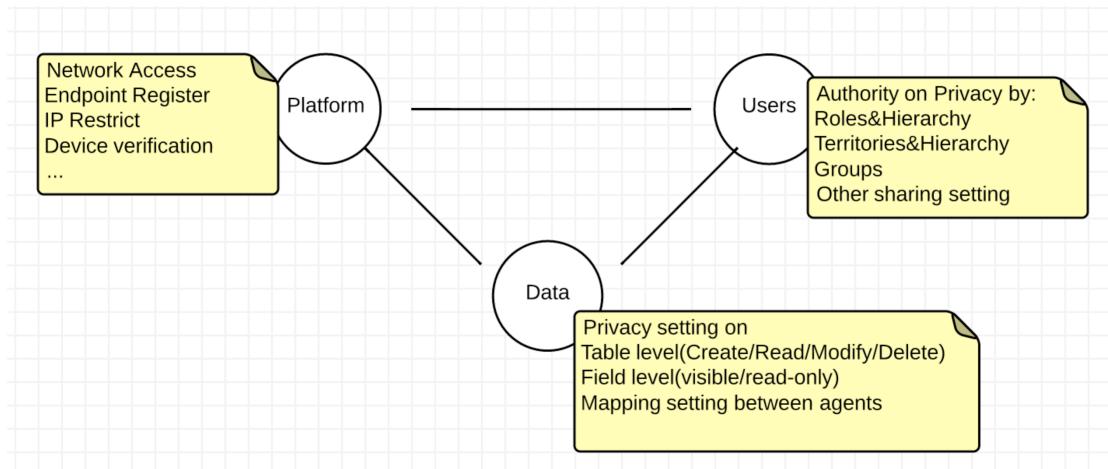


Interaction Description in one scenario:



4.2.4 Privacy Concerns

- 1) As an open environment for each agent, the access control will be addressed as the privacy feature. For example, agent can connect the other agents registered in their endpoints list.
- 2) From user level, the privacy features should be applied due to the different user behavior requirements in each agent. The users in agent can be categorized by different roles, territories, or groups, in which the different authority can be defined, liking initializing request to other agent, set privacy information mapping between agents, and so on.
- 3) The key feature in interdependency is the data privacy between agents, so different level privacy on data should be considered:
 - Table level – define if operations (Create/Delete/Modify/Read) the user can do to the table.
 - Field level – define if the user can view or change this field and its data.
 - Mapping Setting – define which field will be shared to other agent.



5 Results and Validation

5.1 Node.js App and DB in Heroku

I created a Node.js app its server.js file contains services that will be requested, meanwhile the Postgres DB was created in Heroku as the data store in server side. This part can be considered as the “smart segment agent”.

Add-on	Plan	Free
ClearDB MySQL :: Database	Ignite	Free
Heroku Postgres :: Database	Hobby Dev	Free
Heroku Connect	Demo Edition	Free
Papertrail	Choklad	Free

5.2 Salesforce and data mapping between smart agents

I choose salesforce platform to simulate the “smart call agent” for following reasons:

Test Physician

[Customize Page](#) | [Edit Layout](#) | [Printable View](#) | [Help for this Page](#) ?

Show Feed Click to add topics

Physician Surveys [1] | Notes & Attachments [0]

Account Detail		Edit	Delete	Include Offline
Account Owner	Xinyu Yun [Change]	Rating	Hot	
Account Name	Test Physician View Hierarchy	Phone		
Parent Account		Fax		
Segment	E	Website		
Created By	Xinyu Yun, 21/08/2015 1:35 PM	Last Modified By	Xinyu Yun, 28/08/2015 12:42 PM	
		Edit	Delete	Include Offline

Physician Surveys [New Physician Survey](#) [Physician Surveys Help](#) ?

Action	Physician Survey ID	Patient Counts	Status	Created Date	Last Modified Date
Edit Del	PS-00001	A	Submitted	23/08/2015	27/08/2015

PS-00001

[Customize Page](#) | [Edit Layout](#) | [Printable View](#) | [Help for this Page](#) ?

Open Activities [0] | Activity History [0]

Physician Survey Detail [Edit](#) [Delete](#) [Clone](#) [Send to Smart Segment Agent](#)

Physician Survey ID	PS-00001
Account	Test Physician
Patient Counts	A
Status	Submitted
Account ID	001o000000bEQBRAA4
Created By	Xinyu Yun, 23/08/2015 10:57 PM
Last Modified By	Xinyu Yun, 27/08/2015 11:17 AM

[Edit](#) [Delete](#) [Clone](#) [Send to Smart Segment Agent](#)

- 1) Easy to build UI and Http requests: here I create a test physician profile who has survey module, after the survey is submitted, user can click the button “send to Smart Segment Agent” to initialize the request
- 2) Privacy features like user privilege and hierarchy setting
- 3) One step of interaction between agents can be implemented through “Heroku Connect” to sync information and set privacy features like data mapping.

The screenshot shows the Heroku Connect dashboard. At the top, there's a purple header with the Heroku Connect logo, sync status, external objects, and user info (xyun@uwo.ca). Below the header, there's a card for the connection 'testapp-xinyu (IDLE)', with tabs for Overview, Logs, Explorer, Settings, and Upgrade to paid version.

Activity section: Last 8 hours, No activity data available for that time period.

Mappings section: A table showing mappings between SFDC and Database. Status OK for Account and Physician_Survey_dex_c. Pending writes to SFDC are 0 for both. Clickable icons for each mapping row.

The second part of the screenshot shows the 'Edit Mapping: Account' configuration page. It has sections for 'Salesforce → Database' (polling every 10 minutes), 'Database → Salesforce' (writing updates), and 'Mapped Fields' (listing fields like AccountNumber, AccountSource, Active__c, AnnualRevenue with checkboxes for sync).

5.3 Configuration in DEX

Knowledge Broker for DB in smart segment agent

The screenshot shows the DEX Knowledge Broker interface. At the top, there's a header with the DEX logo, a search bar, and a welcome message 'Welcome Xinyu Yun'. Below the header, there's a 'WELCOME TO YOUR KNOWLEDGE BROKER' message.

The main interface is divided into three columns:

- DATA STORES**: Shows various data stores like HD, Ontology, lsmarthome, mysql-xinyu, mysql_ll, postgres-xinyu, test, and test2. Each item has a delete icon.
- TABLES & COLUMNS**: Shows tables like account, _hcmeta, _sf_event_log, physician_survey_dex_c, _trigger_log_archive, _trigger_log, and _trigger_last_id. Each item has a delete icon.
- VIEWS**: Shows views like Android and TumblrTesting. Each item has a delete icon.

After registering DB in KB, the service to operate DB can be requested through calling “<http://kb.latest.dexit.co/access/stores/postgres-xinyu/query/>”.

Service Broker for request “smart object”

The screenshot shows a service registration interface. On the left, there is a list of services with their common names and types:

- send segment anget (ontology)
- the common name for the service.
- ontology
- llremote
- eHealth
- llremote (the common name for the service.)
- the common name for the service.
- llremote

On the right, a modal window titled "Update the Service" contains the following fields:

- Service Name:** Request Segment Agent
- Service Description:** Function in the Demo
- Service Type:** restful
- Service Definition:** Endpoints | Service Definition
- EP Name:** default
- EP:** <http://testapp-xinyu.herokuapp.com>

When you register each ‘GET/POST’ function as a service in SB, the requester does not need know the server but only trust the SB host, requests will be sent to service broker initially.

5.4 Demo sequence to validate

- 1) Initialize physician segment ‘A’ in “smart call agent”, start server in Heroku and make sure the initial segment synced to “smart segment agent”
- 2) Create and submit a new survey record, click the button to start request
- 3) Heroku server will receive the request to start evaluating the segment computation, after the computation the DB update request in “smart segment agent” will be sent through KB

- 4) Once the segment is update to a new value, the value will be synced back to “smart call agent”, user can refresh the webpage to see the update.
 - 5) During the interaction, once error related to data privacy or any other problem, the error message will response to the requester through http communication, user can check the message from webpage and log file.
-

6 Conclusion and Future Work

Generally the goals for this project are enrich the solution on CLM in CDS and extend the computation capability with CDS architecture.

Based on current understanding and learning, the future work should be focused on following aspects:

1. Complete the Smart Content feature with CLM and MCM scenarios
2. Design more features on privacy and authority, like access control, privacy setting based on different roles or groups with research group's cooperation
3. Enhance the protocol design with negotiation process and privacy concerns
4. Study how to extend the computation capability with “smart agent” architecture & design. For example, the CRM in cloud platform is fitting for recording the transaction data, but only with limited power to aggregate the bulk data and compute the analysis result on BI(Business Intelligence) perspective. Is it possible to apply a cooperative Smart Agent to cover data processing and analyzing then show the report & dashboard in CRM with higher performance and lightly integration?

7 References

- [1]. <Multichannel Closed Loop Marketing Digitally Transforming the Life Sciences Industry> Tim Moore, Hala Qanadilo
- [2]. <Privacy in Cooperative Distributed Systems:Modeling and Protection Framework> Afshan Samani 2015
- [3].Introduction of Multi Channel CRM in Veeva System Inc.
<https://www.veeva.com/products/multichannel-crm/> 2015
- [4]. DEX developer tutorial <http://developer.dexit.co/tutorial>
- [5]. Heroku developer Step-by-step guides <https://devcenter.heroku.com/start>

[6]. Saelsforce.com developer guide

<https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/>