Activity Report: Privacy based Brokering Architecture for Smart Object Model

Xinyu Yun

Cooperative Distributed Systems Engineering Group

Instructions for Completing this Report

The information in this report will be used to assess your academic progress and, as such, may influence the granting of scholarships and awards.

Submit the completed report by the assigned due date.

1. Background Information

Student's Name	Xinyu Yun	ID #	250815992	E-mail Address	xyun@uwo.ca
Program	M.Eng – Software Engineering			Start Date	2015/09/01
Academic Term	2015 Fall Graduate			Status[1]	
Proposed Research Field	*CDS Entity, architecture and implementation*			Proposed Research Area	
Thesis topic	*Privacy based Brokering Architecture for Smart Object Model*			Seminar Date	

2. Financial Support

List all sources of funding **(e.g., CDS-EnG, UWO, NSERC, Government Assistance, Foreign), the type of support (e.g., RA, TA, NSERC IPS, OGS)**, currently held or applied for.

Source	Type	Description	$ amount/term	Held/Applied	Period of tenure

[1] as indicated on your registration form

3. Project description

As a continuous project from the course project in 2015-summer term, following functions will be covered:

- **Study the CDS model and interaction protocol with privacy concerns**
- **Design the model and architecture to support privacy based CNP interaction protocol and brokering layer.**
- **Build smart object in Heroku to extend Salesforce requester's capability of decision-making.**
- **Build smart object in Heroku coping with DEX platform to support brokering requests to multiple providers.**
- **Build smart object in Heroku to provide encryption service and protect register's confidential.**
- **Build multiple providers in Heroku to provide capability checking and bidding & encryption interaction process**

Summary & Background

Areas of Background:

AREA	ASPECT	NOTES
Software Development in open environment	Node.js + Heroku + Salesfoce.com	
Brokering configuration in DEX platform	Knowledge Broker and Service Broker configuration	
CDS model and architecture	Smart Object/Entity model in open environment	

3.1. Research Challenges

Main Research Topics:

TOPIC	CONTRIBUTION	RESEARCH ISSUE	NOTES
Interaction Protocol Design	Summarize the privacy based CNP protocol		
Service API in salesforce.com	"Heroku Connect" to sync the data between salesforce and Heroku		
Encryption model design			

3.2. Research Goals & Objectives

Technically, the goal is to build CDS architecture in open environment, especially in cloud platform to identify the computation and integration capability.

1) Design and implement the interaction protocol between requestor to smart segment service providers
2) Address the privacy concerns of the information sharing when negotiating with service providers
3) Design and implement the services of smart call agent in salesfoce.com to extend the capability of decision-making
4) Register the services in DEX to implement the broking layer

3.3. Proposed Solutions

1. **From Entity level, each smart object should be designed as the proposed model bellowed, and privacy concerns will be addressed in interaction layer from following perspectives:**
   - **Sensitive information checking**
   - **Exposure boundary**
   - **Operation Protection**

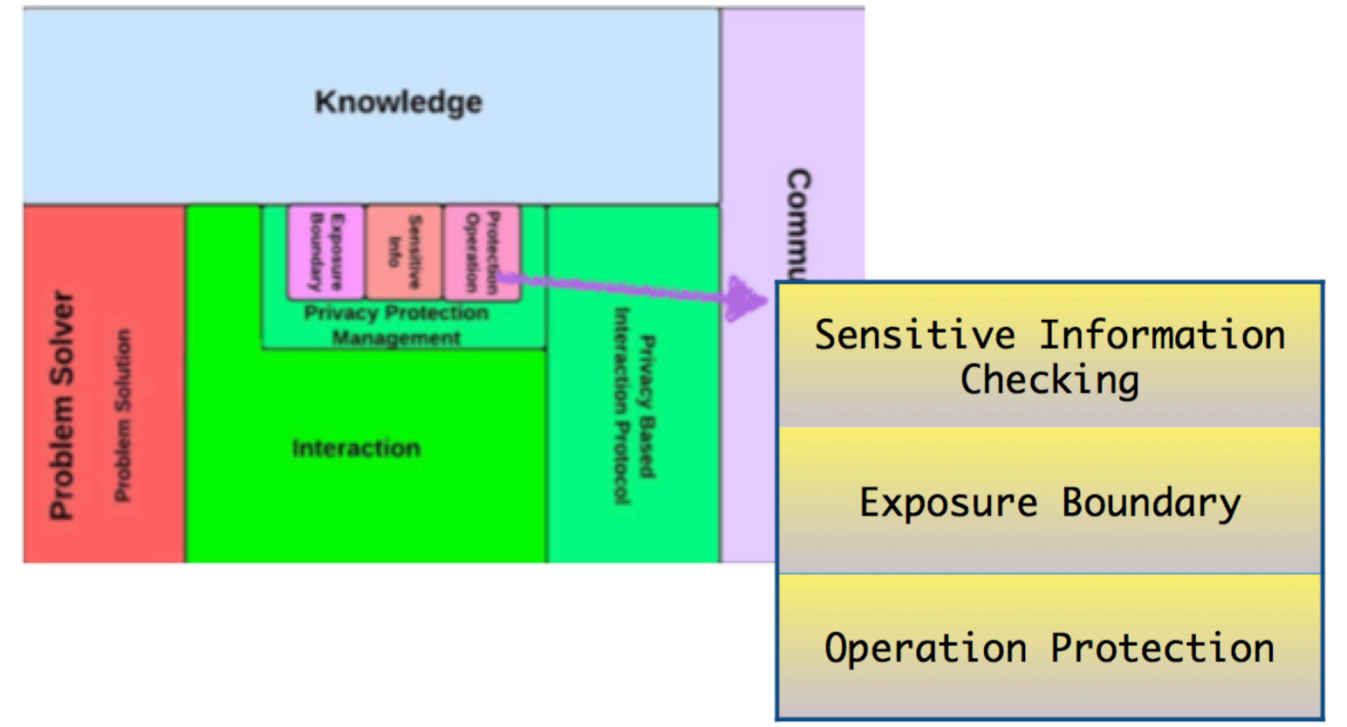


Figure 1 - Entity model in CDS

2. **From architecture level, service requesters and providers will interact and communicate through the brokering manager layer.**
   1) Service requester will start its request through communicating with brokering layer; during the interaction process, service requester will have the capability to make decision among multiple providers' proposal.
   2) Service providers will not talk to requester directly but cooperate with smart objects in open environment for interaction and message forwarding.

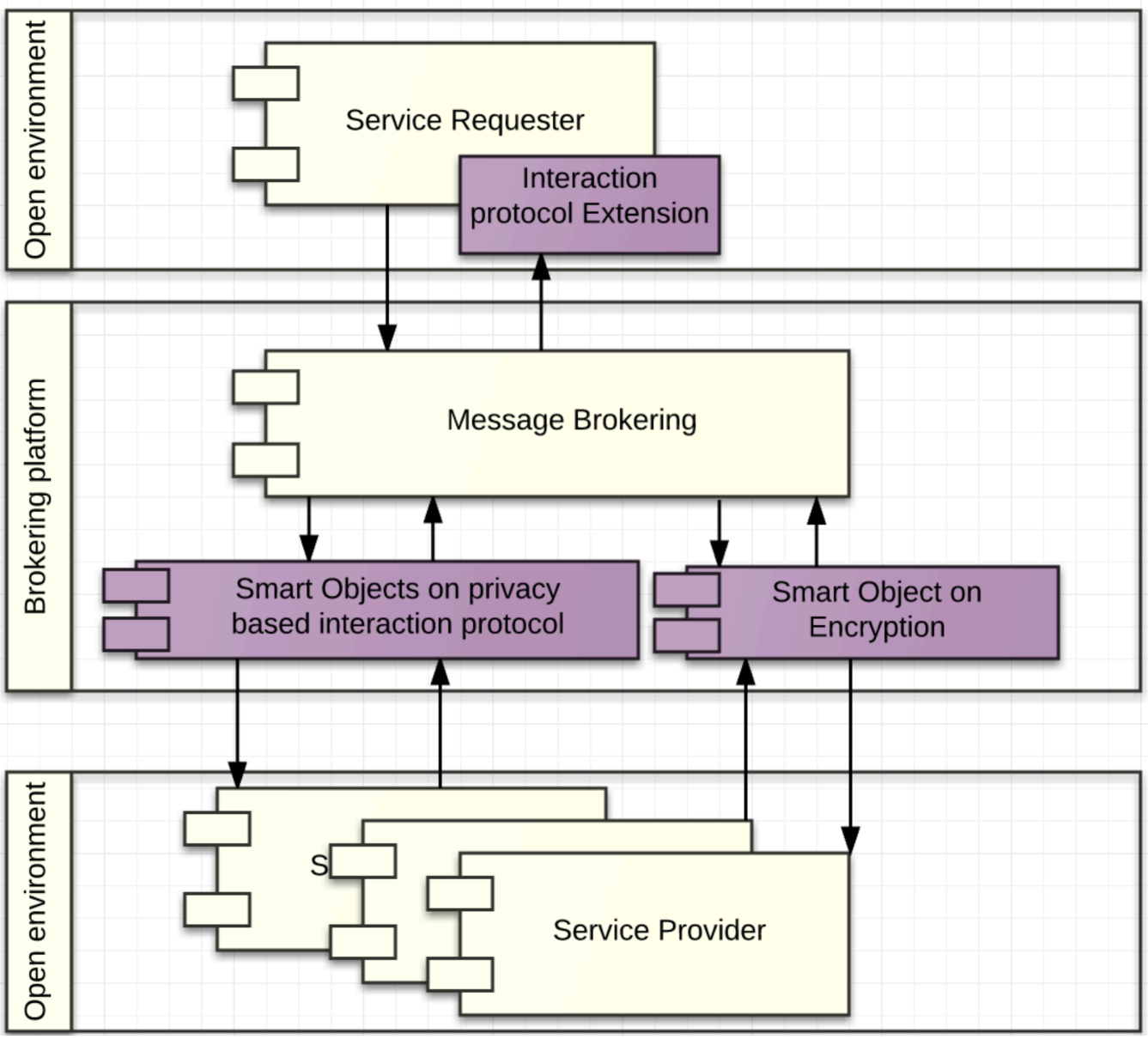


Figure 2 - Logical architecture

3. **From the deployment architecture level:**
   1) The initial request will be registered in DEX' service broker, the requestor in salesforce.com will only know the DEX's host as the trusted broking layer.
   2) All potential providers will be registered in DEX, and the smart object (SO) that is in charge of the Brokering will broadcast request task to start the interaction process
   3) All the potential providers will response the key message to attend the bidding process; the key message should involve the key identifiers of each provider, like the connection counts of each agent, or rejection message due to the issues in open environment.
   4) The brokering SO will collect the responses and forward to decision making SO cooperative with initial requester then decide which one is the winner and call the service to winner provider through DEX.
   5) In Heroku platform, another smart object to support encryption services will be implemented and deployed.

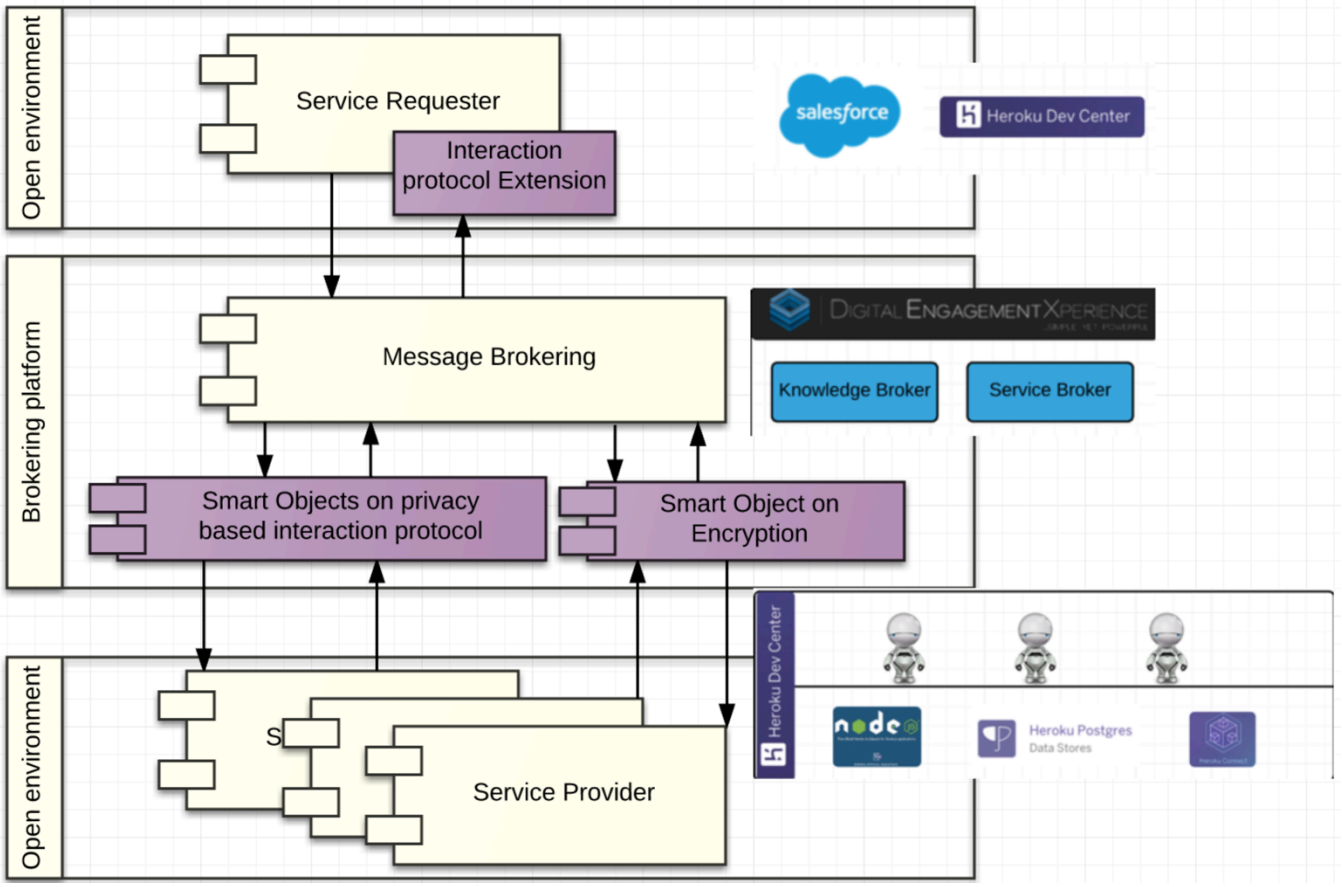


Figure 3 – Deployment Architecture

4. Interaction protocol between requester, brokering layer and providers

Design principles on privacy concerns:

1) Build a brokering layer to deal with the request, response and negotiation process, which can avoid direct communication between agents and lower the risk of information leaking.
2) All potential providers will register their service portal in brokering layer, which makes the real request path and services are anonymous.
3) All potential providers are hiding their capability, and will check the capability based on the requested task.
4) During the request, the information explicit from requester should be as less as possible, normally we just transfer the 'task ID' to identify the specific record we would like to deal with. And for different requested scenarios, requester should use different 'task id'.
5) During the interaction process, the bidding value will be encrypted; only the recognized requester has the authority to decrypt the bidding value for winner's decision-making.
6) All the capable providers will contact the 'Encryption Smart Object' to get the encrypted bidding value then forward it and accordant dealing service id to brokering SO
7) The Brokering SO collects all bidding information and transfer to decision-making (DM) SO of requester, the DM SO will contact Encryption SO to decrypt the bidding and select the winner provider. Afterwards, DM will call the accordant service from winner provider to execute the task

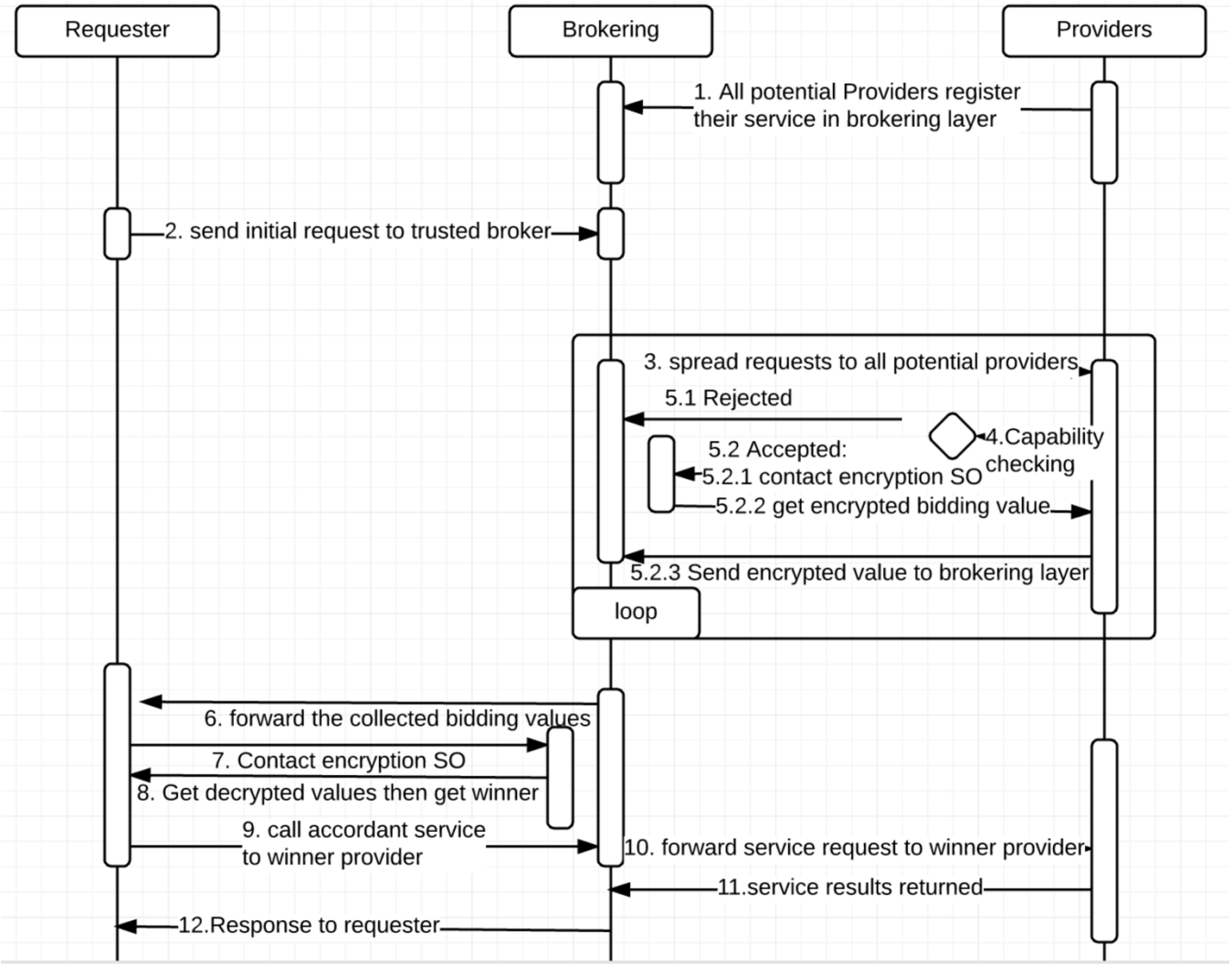


Figure 4 – Interaction Protocol with Privacy Concerns

5. Detailed Design and Implementation

Based on the previous designed scenario, the requester in Salesforce collecting the physician's survey will send a 'segmentUpdate' request to 'smart segment object', 'smart segment object' will evaluate the survey details and set the new segment, after that the new segment will be synced back to Salesforce in real-time.

1) Service registration in DEX platform

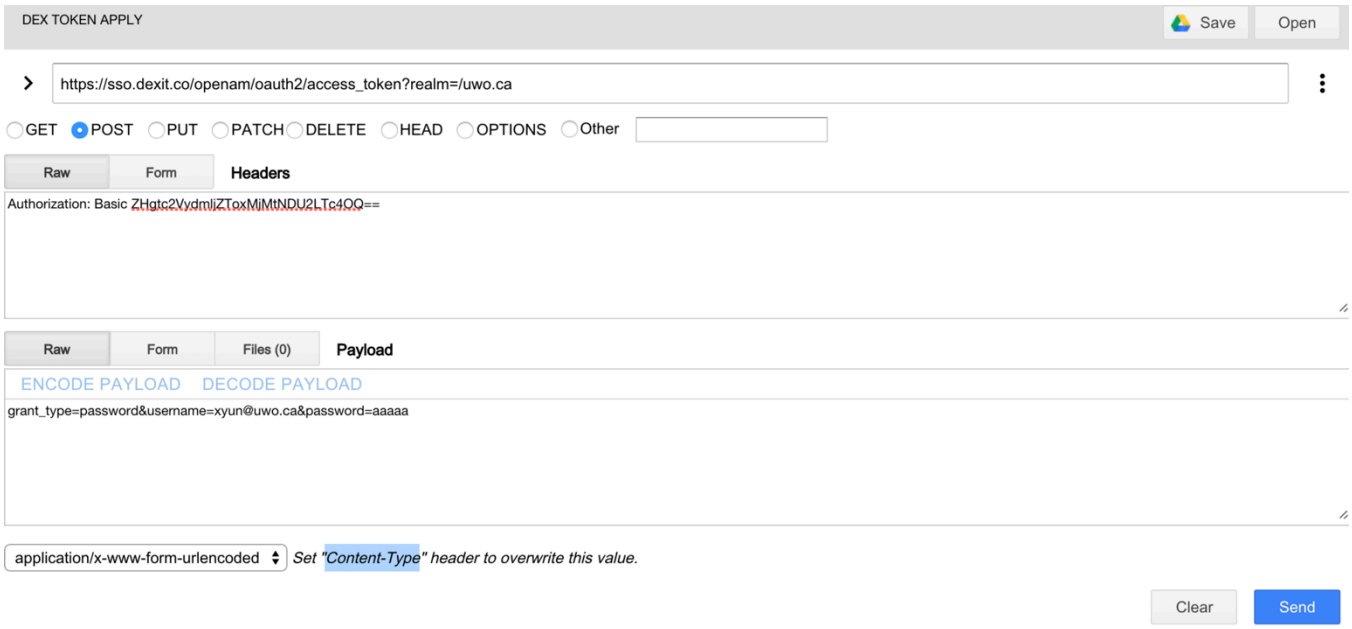
Request the TOKEN:

URL: https://sso.dexit.co/openam/oauth2/access_token?realm=/uwo.ca

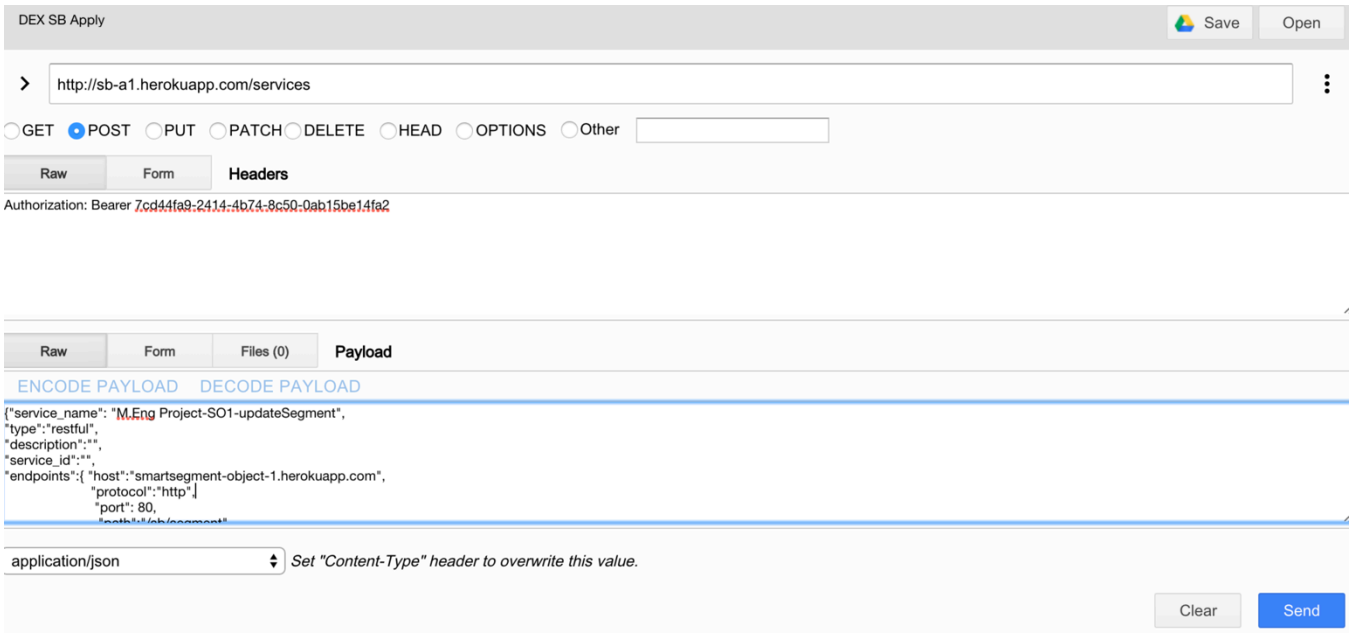
Method: POST

Headers: Authorization: Basic ZHgtc2VydmljZToxMjMtNDU2LTc4OQ==
Content-Type:application/x-www.form-urlencoded

Body: grant_type=password&username=xyun@uwo.ca&password=aaaaa



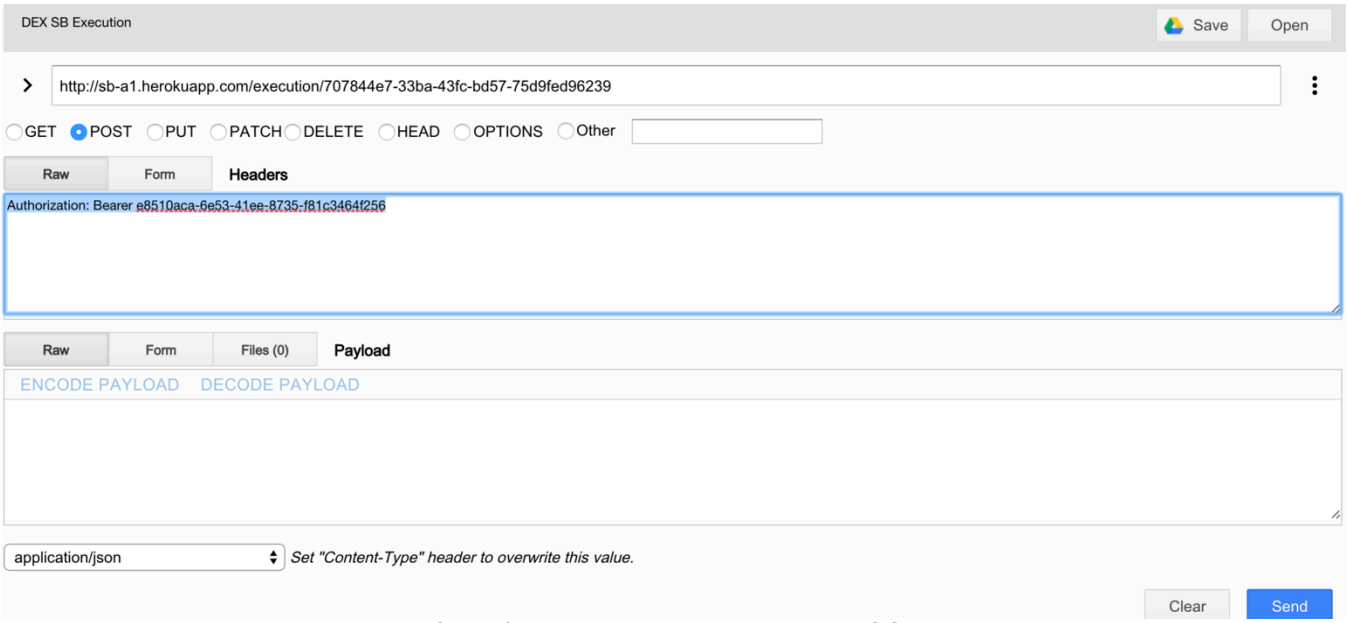
DEX Service Broker Registration:
URL：http://sb-a1.herokuapp.com/services
Method: POST
Headers: Authorization: Bearer 7cd44fa9-2414-4b74-8c50-0ab15be14fa2
Content-Type:application/json
Body：{"service_name": "M.Eng Project-SO1-updateSegment",
"type":"restful",
"description":"",
"service_id":"",
"endpoints":{ "host":"smartsegment-object-1.herokuapp.com",
"protocol":"http",
"port": 80,
"path":"/sb/segment"
}
}



DEX Service Execution

1. **Add service id in URL**
2. **Put TOKEN in headers**

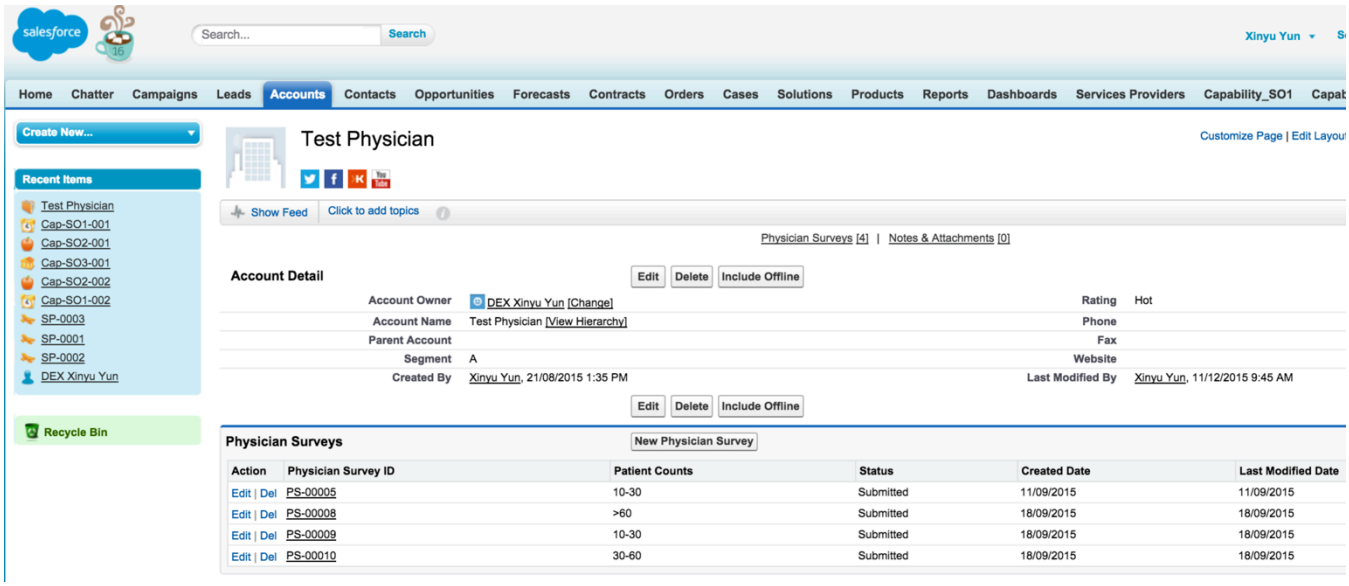
URL: http://sb-a1.herokuapp.com/execution/707844e7-33ba-43fc-bd57-75d9fed96239
Method: POST
Headers: Authorization: Bearer e8510aca-6e53-41ee-8735-f81c3464f256



2) Requester smart object: Salesfoce.com + decision-making SO in Heroku

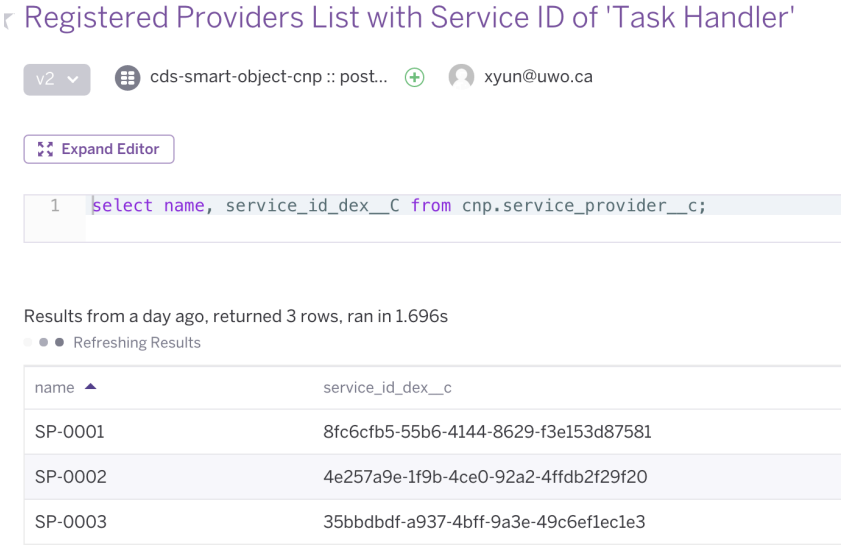
Manage Account information and physician survey in Salesfoce, after collecting the survey client can send the segment update request to start the interaction.

The service to handle request from Brokering layer has been registered in DEX and service id is saved in salesforce side and get ready to be called anytime.



3) Brokering layer: DEX platform + CNP Smart Object in Heroku

All service providers' entry services will be registered in DEX and saved in heroku CNP SO's table. After receiving the request, the CNP SO will spread the request to potential providers with task ID and requester ID.



Once receiving the bidding values from capable potential providers, the CNP SO will collect the bidding values and accordant service id and forward to requester's decision making SO.

4) Decision-making SO with requester

Once receiving the bidding values from brokering layer, the requester's decision-making SO will interact with encryption SO to decrypt the bidding values then select the winner provider with the accordant service ID. Then requester will forward its request with winner' service ID to execute the service and get the result.

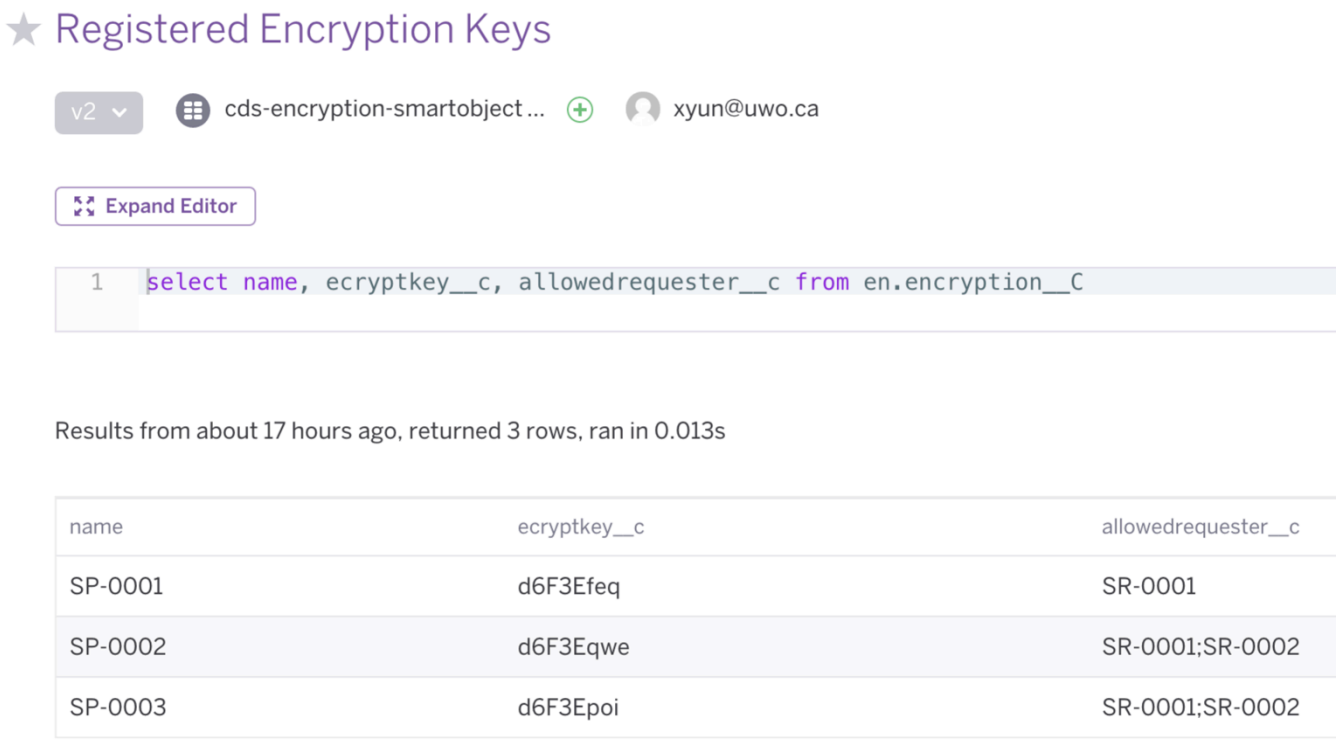
5) Service Providers in Heroku

After receiving the task notice from CNP SO, the provider will check its own capability, if it is capable to solve this task, the encrypted bidding value will be generated through interacting with encryption SO to retrieve the key and register the trusted requester id.

Capability registered table:

6) Smart Object for encryption

In this SO the accordant keys mapping to specific service provider are saved, we can define different encryption algorithm to offer encrypt or decrypt service.



7) API and Services List in Heroku platform with Node.js:

SmartObject	File Name	RestFull API	DEX Service ID	Description
requester-decisionmaking-so1	dm_so1.js	/sr1/dm	5851c06c-98b3-4975-b743-fd7ef7a47ae9	1.forward the request from original requester to brokering layer 2.receive proposed bidding values then decrypt them and select the winner(minimum one) 3.forward request through winner's service ID to deal with the task then return the result to original requester
cds-smart-object-cnp	cnp_server.js	/sb/taskBrokering	707844e7-33ba-43fc-bd57-75d9fed96239	1-received the post request from requester, save the token from request's header, task and decision-making SO service ID from request's body 2-select all registered providers' information with each's taskHandler' service ID 3-post request to each potential provider with request body containing 'task','requester ID' 4-check each response accordingly, if OK with encrypted bidding value and accordant service id, save them 5-forward bidding information to requester's decision making 'SO' to proceed
cds-encryption-smartobject	cds.encryption.js	/enso.getEncryptValue	4c7345f6-9525-4133-9283-571ac7fb0e73	1.send query to get the key from table based on the service provider id in requester's body 2.decrypt bidding value with retrived key and algorithm then return the encrypted value in response
cds-encryption-smartobject	cds.encryption.js	/enso/decryptValue	2a6b7935-ba60-4fd3-b1a0-ec0edb4c993c	1.send query to get the key from table based on the service provider id in requester's body 2.decrypt bidding value with the key and algorithm
smartsegment-object-1 smartsegment-object-2 smartsegment-object-3	server_so1.js server_so2.js server_so3.js	/sb/taskHandler	8fc6cfb5-55b6-4144-8629-f3e153d87581 4e257a9e-1f9b-4ce0-92a2-4ffdb2f29f20 35bbdbdf-a937-4bff-9a3e-49c6ef1ec1e3	1-received task announcement request, parse the body to get task name 2-compare with local capability to match the task if matched, request to encryption SO with bidding value to generate encrypted bidding value then return it with task dealing service ID in response message if not matched, return insufficient capability code

3.4. Potential Industrial Applications

As learned in CDS research group, the model can be applied in different 'smart space' application when it has the open environment and need cooperation with other entities to solve the problem.
The architecture is fitting in the agile development without concerning infrastructure, since the app and database can be deployed in Heroku through configuration, meanwhile the integration and scalability with other platform would be easier with the Node.js and various add-ons.
The survey and segmentation scenario can be used in customer's evaluation, such as CRM account management, active learning survey for students.

3.5. List of publications in this project

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. Progress Towards Completion of Degree Requirements

4.1. Course-Work

Code	Title	Term	Credit/Audit/ Set-in	Grade	Comments
ECE9605	Web Application Development	2015 winter	1.0	90	
CS9668	Internet Algorithm	2015 winter	1.0	80	
ECE9067	Cooperative Distributed System Engineering	2015 summer	1.0	90	
ECESCI 9510	Engineering Planning and Project Management	2015 summer	1.0	80	

4.2. Scholarly Activities Towards Degree Requirements

Identify scholarly activities, **related to your** research, initiated or completed until this term[2].

#	Activity description	% Compl	Original Completion Date	Revised Completion	Comments
1					
2					
3					
4					
5					
6					
7					

[2] **Scholarly activities include progress towards your research project as outlined in Sec. 4.2, presentations at conferences, submission/acceptance of articles related to your research, etc.** *In each case please provide full details*.

4.3. Plan for Completion of Degree Requirements

Identify milestones for completing your research requirements[3] along with a corresponding timetable.

#	Milestone	Original Target Date	Revised Target Date	Activities	Comments
1					
2					
3					
4					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12	**Dissertation**				
1	Conference 1: -				
2	Conference 2:				
3	Conference 3:				
4	Chapter1:				
5	Journal 1:				
6	Journal 2:				

5. Student Comments

Include anything you feel should be brought to the attention of the your supervisor or to the Graduate Studies Committee.

6. Supervisor(s) Comments

Student Signature Date:

Supervisor Signature _____ Date: _____

Co-Supervisor Signature _____ Date: _____