Sysops Squad

- Team **ArchSempais**

ArchKata 4/21

1. Introduction

1.1. Purpose and Intended Audience

The purpose of this document is to give the stakeholder a clear and precise description of the Sysops Squad from the perspective of the software architecture component and connector (C&C) view. The intended audience of this document is all system stakeholders. It is a point of reference for all stakeholders to review and use at any time during the product development lifecycle.

1.2. Scope of Product

1.3. Requirements

Table 1 presents the requirements gathered from interview, research, and project description from the client.

Table 1: Requirements from Client, Interview, and Research

Requirement No.	Requirements
1	System is available for web-based problem ticket entry.
2	System is available for call-based problem ticket entry.
3	Each ticket is addressed by an expert correctly and correctly fixed.
4	System determines which expert would be best fit for the job based on skills, current location, service area, and availability (free or currently on a job) once a ticket is created in the system.
5	System assigns a ticket to an expert.
6	System modifies ticket assignment in real time due to changes in expert's availability to shorten the time it takes to resolve the ticket.
7	System generates ticket assignment notifications.
8	System sends ticket assignment notifications via text message.
9	System generates an email regarding an expert is on their way.
10	System generates a text message regarding an expert is on their way.
11	System sends an email or text message regarding an expert is on their way to the customer.

12	Ticket status is transparent.
13	Every ticket is resolved in a timely manner. If an expert fails to mark the ticket as complete, what should the system do?
14	Making modifications to the system should not affect other functionalities in the system.
15	System can handle spikes in usage.
16	System can handle the number of customers using the system.
17	Administrator maintains (add, delete, edit) expert profiles.
18	Administrator manages all of the billing processing for customers using the system.
19	Administrator maintains static reference data (such as supported products, name-value pairs in the system, ...).
20	Administrator maintains a list of supported products.
21	Customer purchases support plan. He/she can have multiple support plans based on the products he/she has purchased.
22	Customer's satisfaction is important.
23	Customer who has purchased a support plan creates a ticket using the system's website.
24	Customer registers for support service using the system.
25	Customer creates and maintains customer profile.
26	Customer views billing information and statements.
27	System generates billing statements.
28	Customer fills out a survey after a ticket is complete.
29	System tracks the validity of customers' support plans.
30	System charges the customer's registered credit card contained in his/her profile on an annual basis. System supports payment online.
31	System tracks billing history for each customer.
32	Customer specifies notification preference in profile.
33	Expert retrieves ticket information and location from a mobile application.
34	Expert views knowledge base from a mobile application.
35	Expert reviews ticket history from a mobile application.
36	Expert changes the ticket status from in progress to complete.
37	Expert adds information to the ticket history.
38	Expert adds fixes to the knowledge base.
39	System creates a survey.

40	System generates an email with a link to a survey regarding ticket service
41	System sends an email with a link to a survey regarding the ticket service.
42	Manager monitors ticket status.
43	Manager requests financial reports.
44	Manager requests expert performance reports.
45	Manager requests ticketing reports.
46	System generates financial reports.
47	System generates expert performance reports.
48	System generates ticketing reports.
49	System supports mobile and web access.

1.4 Assumptions

The following describes the assumptions:

1. The performance issues of the current system are due to structural problems from the existing monolithic design.
2. The database layer is not the bottleneck of the current solution.
3. Scheduling, rescheduling tickets and registering service plans have the same priority.
4. Reference data is relatively static and includes supported products list, locations, issue categories, etc. managed by admin.

1.5 Design process

The following describes our team's design process:

1. First iteration includes the breakdown of logical software architecture by leveraging domain driven design approach to identifying components, events, service interactions, and boundaries of the bounded contexts.
    a. Review and validated understanding of the system requirements
    b. Identify and prioritize driving architectural characteristics based on requirements and current systems issues
    c. Map the characteristics to select the architectural styles
    d. Identify actors and their workflow activities (Figure 1)
    e. Identify logical software components and their interactions
    f. Select components and data for affinity to bounded context areas
2. Second iteration adds data model validation, most prevalent use cases, security aspects [i.e. authentication, authorization, encryption] and solution implementation design like naming the event queues. Number and name details of all interactions. Convert documentation to GitHub format, *.md files.

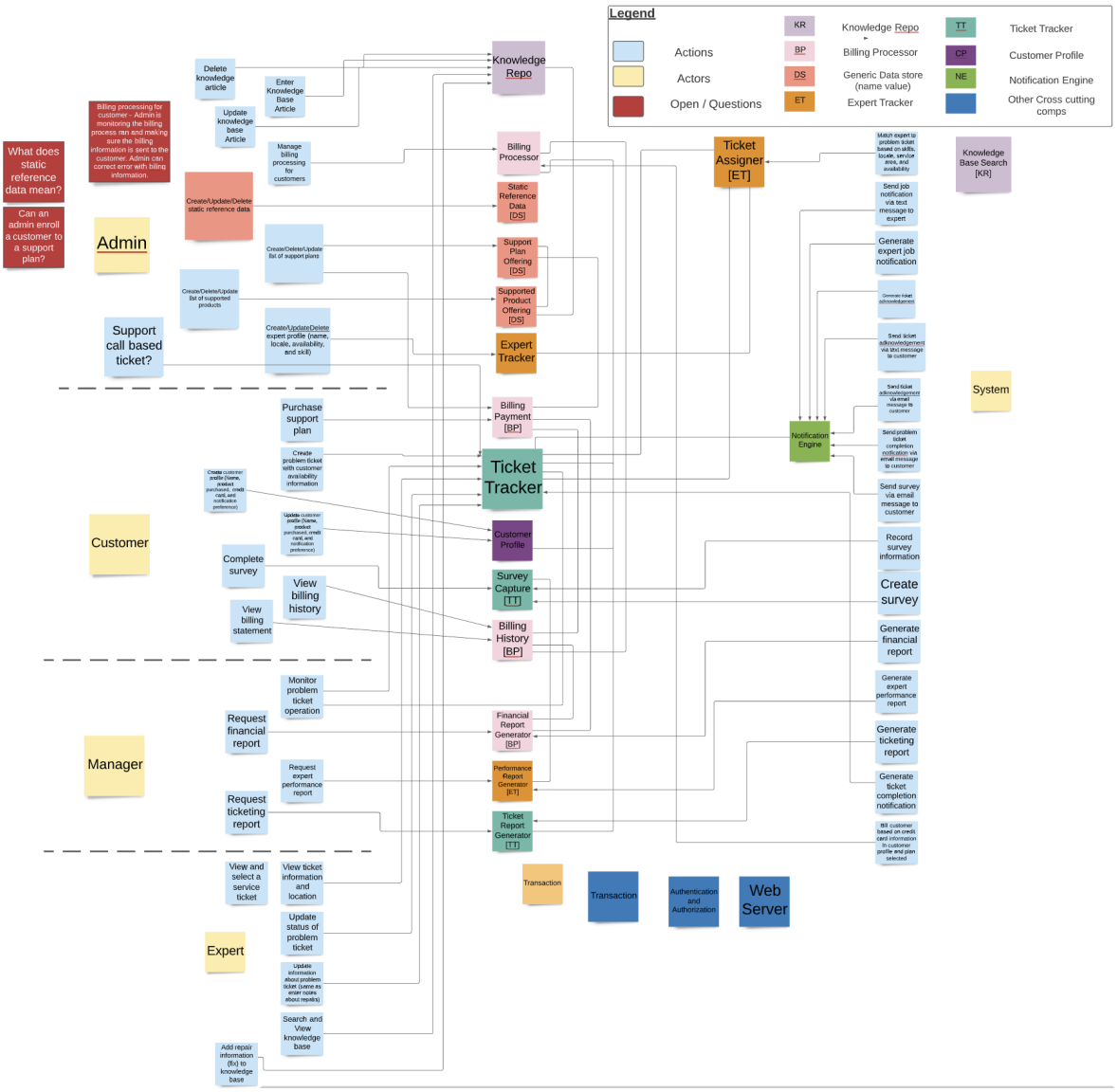

Figure 1: Action and Actor Analysis

2. Architectural Characteristics

Table 2 presents the architectural characteristics analysis based on the requirements documented in Section 1.3. The prioritized list of architectural characteristics for the Sysops Squad is shown in Table 3. Based on the prioritized list of architectural characteristics, it was identified that event driven architecture style and microservice architecture style are suitable for the Sysops Squad.

Table 2: Architectural Characteristics

Requirement No.	Requirements	Architectural Core Characteristics and Rationale	Architectural Core Characteristics and Rationale	
1	System is	Availability		

	available for web-based problem ticket entry.			
2	System is available for call-based problem ticket entry.	Availability		
3	Each ticket is addressed by an expert correctly and correctly fixed.	Reliability	Integrity	
4	System determines which expert would be best fit for the job based on skills, current location, service area, and availability (free or currently on a job) once a ticket is created in the system.	Precision/accuracy - What is the impact of an operational mistake that a wrong expert is assigned to a ticket?		
5	System assigns a ticket to an expert.	Functional - No special architecture characteristics seem necessary to support this requirement.		

6	System modifies ticket assignment in real time due to changes in expert's availability to shorten the time it takes to resolve the ticket.	Reliability	Event based system	
7	System generates ticket assignment notifications.	Functional - No special architecture characteristics seem necessary to support this requirement.		
8	System sends ticket assignment notifications via text message.	Functional - No special architecture characteristics seem necessary to support this requirement.		
9	System generates an email regarding an expert is on their way.	Functional - No special architecture characteristics seem necessary to support this requirement.		
10	System generates a text message regarding an expert is on their way.	Functional - No special architecture characteristics seem necessary to support this requirement.		
11	System sends an email or text	Functional - No special architecture characteristics seem		

	message regarding an expert is on their way to the customer.	necessary to support this requirement.		
12	Ticket status is transparent.	Auditability - log all the events to provide history	Event Driven Updates to the tickets	
13	Every ticket is resolved in a timely manner. If an expert fails to mark the ticket as complete, what should the system do?	Functional - No special architecture characteristics seem necessary to support this requirement.		
14	Making modifications to the system should not affect other functionalities in the system.	Modifiability	Serviceability	Maintainability
15	System can handle spikes in usage.	Elasticity		
16	System can handle the number of customers using the system.	Scalability		
17	Administrator maintains	Functional - No special architecture	Security : Are all the requests properly	

	(add, delete, edit) expert profiles.	characteristics seem necessary to support this requirement.	authorized? Did the expert mix up happen because of an unprivileged user modifying the database?	
18	Administrator manages all of the billing processing for customers using the system.	Functional - No special architecture characteristics seem necessary to support this requirement.		
19	Administrator maintains static reference data (such as supported products, name-value pairs in the system, ...).	Maintainability - changing the data without impacting the system		
20	Administrator maintains a list of supported products.	Functional - No special architecture characteristics seem necessary to support this requirement.		
21	Customer purchases support plan. He/she can have multiple support plans based on the products he/she has	Functional - No special architecture characteristics seem necessary to support this requirement.	Security: While handling credit cards, validating if customer indeed brought the product at store	

	purchased.			
22	Customer's satisfaction is important.	Usability	Performance	Reliability
23	Customer who has purchased a support plan creates a ticket using the system's website.	Functional - No special architecture characteristics seem necessary to support this requirement.	Security: Authentication & Authorization. Any other network attacks such as DoS	
24	Customer registers for support service using the system.	Functional - No special architecture characteristics seem necessary to support this requirement.		
25	Customer creates and maintains customer profile.	Functional - No special architecture characteristics seem necessary to support this requirement.		
26	Customer views billing information and statements.	Functional - No special architecture characteristics seem necessary to support this requirement.		
27	System generates billing statements.	Functional - No special architecture characteristics seem necessary to support this requirement.		
28	Customer fills out a survey after a ticket is complete.	Functional - No special architecture characteristics seem necessary to support this requirement.	AuthN, AuthZ - Survey should be submitted over web after authentication only	
29	System	Functional - No special		

	tracks the validity of customers' support plans.	architecture characteristics seem necessary to support this requirement.		
30	System charges the customer's registered credit card contained in his/her profile on an annual basis. System supports payment online.	Functional - No special architecture characteristics seem necessary to support this requirement.	Online payment implies security. But nothing in the requirement suggests heightened levels of security beyond what's implicit. Are there any structural components needed to accommodate security?	
31	System tracks billing history for each customer.	Functional - No special architecture characteristics seem necessary to support this requirement.		
32	Customer specifies notification preference in profile.	Functional - No special architecture characteristics seem necessary to support this requirement.	Customer phone number and PII should be guarded for privacy	
33	Expert retrieves ticket information and location from a mobile application.	Functional - No special architecture characteristics seem necessary to support this requirement.		
34	Expert views knowledge base from a	Functional - No special architecture characteristics seem necessary to support		

	mobile application.	this requirement.		
35	Expert reviews ticket history from a mobile application.	Functional - No special architecture characteristics seem necessary to support this requirement.		
36	Expert changes the ticket status from in progress to complete.	Functional - No special architecture characteristics seem necessary to support this requirement.		
37	Expert adds information to the ticket history.	Functional - No special architecture characteristics seem necessary to support this requirement.		
38	Expert adds fixes to the knowledge base.	Functional - No special architecture characteristics seem necessary to support this requirement.		
39	System creates a survey.	Functional - No special architecture characteristics seem necessary to support this requirement.		
40	System generates an email with a link to a survey regarding ticket service	Functional - No special architecture characteristics seem necessary to support this requirement.		
41	System sends an email with a link to a	Functional - No special architecture characteristics seem necessary to support		

	survey regarding the ticket service.	this requirement.		
42	Manager monitors ticket status.	Functional - No special architecture characteristics seem necessary to support this requirement.	Authorization - Other users should not be allowed to look at reports.	
43	Manager requests financial reports.	Functional - No special architecture characteristics seem necessary to support this requirement.		
44	Manager requests expert performance reports.	Functional - No special architecture characteristics seem necessary to support this requirement.		
45	Manager requests ticketing reports.	Functional - No special architecture characteristics seem necessary to support this requirement.		
46	System generates financial reports.	Functional - No special architecture characteristics seem necessary to support this requirement.		
47	System generates expert performance reports.	Functional - No special architecture characteristics seem necessary to support this requirement.		
48	System generates ticketing reports.	Functional - No special architecture characteristics seem necessary to support this requirement.		
49	System	Functional - No special		

	supports mobile and web access.	architecture characteristics seem necessary to support this requirement.		

Table 3: Prioritized List of Driving Architectural Characteristics

Architectural Characteristics	**Priority (Most important to least important)**	**Rationale**
Scalability	1	The system needs to be able to accommodate growth in customers, products, and geographical expansions of the business
Reliability	2	The system needs to be resilient to errors and changes in processing conditions and traffic
Elasticity	3	The system needs to dynamically adjust to spikes in usage expected in the area of new tickets.
Performance	4	Influences user satisfaction as an important business requirement
Availability	5	User satisfaction requires ability to manage profiles and tickets at users' convenience
Auditability	6	Detailed records of customer requests, services performed, billing, etc. need to be kept and tracked in an Audit log.
	7	

Table 4: List of Implicit Characteristics

Architectural Characteristics	**Priority (Most important to least important)**	**Rationale**
Security		

Maintainability		

3. Component and Connection View

This section contains all the information pertaining to the component and connection (C&C) view as it pertains to the Sysops Squad.

3.1. Components and Connections Overall View

Figure 2 presents the component and connection (C&C) view of the Sysops Squad system and the key for the view. Blue rounded rectangle represents an external system. Uncolored rounded rectangle represents a service. Uncolored rounded rectangle with a database icon represents a service that owns its data. The database icon is used to represent a service owning its own data. Asynchronous communication between components is represented by dotted lines. Synchronous communication between components is represented by solid lines. Cylinder shape represents an event queue. The view can also be accessed via this lucid chart link:.
https://lucid.app/lucidchart/invitations/accept/inv_920a6136-3930-43dd-9497-5f94b563d4b2

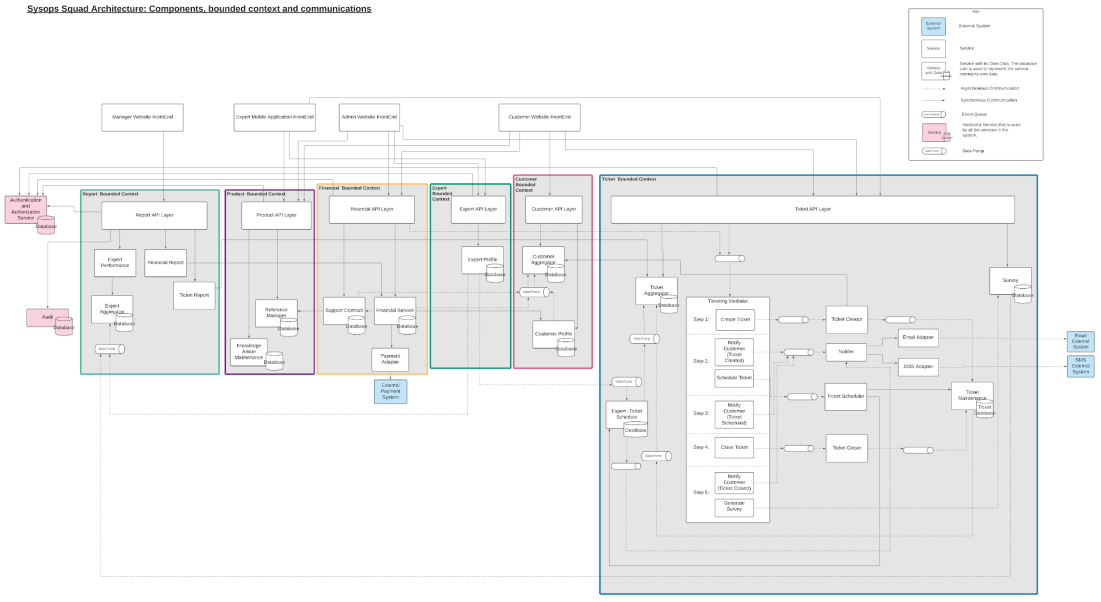

Figure 2: C&C view

3.2. Element Catalog

This section names each element in this view and lists the properties of each element. An element can be either a service or communication. Please see Table 5 for details.

Table 5: Elements and Their Properties

Element Name	Type	Bounded Context	Description

Expert Aggregator	Service	Expert management	Gathers expert information for manager reporting
Expert Profile	Service	Expert management	Handles expert profile processing including creating new expert profiles, modifying expert profile information, and deleting expert profiles. It maintains expert information, expertise, and availabilities.. This service has its own data.
Customer Profile	Service	Customer Management	Handles customer profile processing including creating new customer profile, modifying customer profile information, and deleting customer profile. It maintains customer information, customers' CC payment information, and availability preferences . This service has its own data.
Email Adapter	Service	Notification and Alerts	- Transforms system calls and handles all communications between the external email system and Sysops Squad
Ticket Mediator	Service	Ticket Management	- Manages the workflow and conditional processing of creating, scheduling (rescheduling), and closing of tickets. - Generates corresponding processing events that are sent to dedicated event channels to the corresponding ticket related services. - Handles error situations.
Ticket Creator	Service	Ticket Management	Implements ticket creation business logic including validation of active contracts for the customer.
Ticket Closer	Service	Ticket Management	Implements ticket closing business logic.
Ticket Scheduler	Service	Ticket Management	- Determines a time slot for a ticket. The decision is based on

			the following: A. Customer's availability B. Expert's availability C. Expert's expertise. - Collaborates with job assignment service to book an expert for a specific ticket.
Ticket Maintenance	Service	Ticket Management	Handles CRUD operations of tickets and maintains write ownership of the ticket data. This service owns its data.
Survey	Service	Survey Management	Implements APIs for surveys and backend for responses. This service owns its data.
Bounded Context API Layer	Service		Exposes the end points of services available to the Sysops Squad mobile application and performs data validation and data sanitization on inputs.
Notifier	Service	Notification and Alerts	- Constructs and sends notifications via preferred channel.
SMS Adapter	Service	Notification and Alerts	- Transforms system calls and handles all communications between the external SMS system and Sysops Squad.
Ticket Aggregator	Service	Ticket management and Expert management	- Builds a materialized view of ticket information for faster access. - Listens on the queues in the data pumps. Data will be eventually consistent with the data sources.
Expert Scheduler	Service	Expert Management	- Maintains information of booked tickets for experts and scheduled time slots. This service owns its data.
Knowledge Article Maintenance	Service	Knowledge Article Management	Handles CRUD operations of knowledge articles and maintains write ownership of the knowledge articles.

			This service owns its data.
Financial Service	Service	Financial Management	Runs the monthly billing cycles. Handles all payment processings and transactions. Payments are handled with different adapters depending on customers' payment method. The full billing history is stored by this service in its database. This service owns its data.
Support Contract	Service	Financial Management	Maintains the list of contracts purchased by the customers
Reference Manager	Service	Reference	Static data about products and their corresponding support plans

4. Architectural Decisions

This section presents the architectural decision records that document architecture decisions. Each architectural decision record has five sections: Title, Status, Context, Decision, and Consequence.

4.1 Design Decisions for Architectural Characteristics

Table 6: Mapping of Architectural Characteristics and Components Design

Architectural Characteristics	**Supporting Design Decisions**
Scalability	● Microservices style will allow to scale each component horizontally with regional data sharding ● Usage of request queues and worker pool pattern to spread the workload to allocated worker threads
Reliability	● Tbd - error detection and handling
Elasticity	● Using cloud based auto scaling to spin up additional instances of Ticket Bounded Context microservice
Performance	● Tbd - auto scaling based on the demand to desired performance metrics SLI [System Level Indicator]
Availability	● System uptime to be ensured by proactive monitoring and reliability
Auditability	● All system components would be required to post their activity Audit log.

4.2 Architecture Decision Records

4.2.1. Role of existing database in the system's upgrade

Title
Use of current database design, technology, and data

Status (Proposed, Accepted, Superseded):
Proposed

Context
Per assumption database is not the cause of the bottleneck or technical constraints

Decision
Reuse the existing database schema and technology, and split the entities to distribute into the services for encapsulation within their bounded context.

Consequences
Avoids data migration and conversion effort
.

4.2.2 Selection of Architectural Styles

Title
Selection of architectural style for the solution software architecture

Status (Proposed, Accepted, Superseded):
Proposed

Context
…

Ticketing workflow ...

Decision
Software architecture will leverage combination of Event-driven and Microservices styles

Consequences
Relative to the cost of the added complexity the hybrid architectural styles would allow to address the scalability and availability quality characteristics

4.2.3 Managing scalability of service by sharding by location/region

Title
How to scale the architecture for geographically distributed product services ...

Status (Proposed, Accepted, Superseded):
Proposed

Context
...

Decision
Each service is instantiated with data for a region and can be scaled horizontally to the same shared instance of the database [service based]

Consequences
...

4.2.4 Security of data

Title
How to secure customer, payment, billing information ...

Status (Proposed, Accepted, Superseded):
Proposed

Context
...

Decision
Leverage standard of the shelf identity provider component for authentication and authorization, and encryption for data at rest

Consequences
Simplification of design based on commodity products

4.2.5 Notification Service

Status (Proposed, Accepted, Superseded):
Proposed

Content:
The Sysops Squad system generates notifications to customers and experts via email, and SMS services. Notifications include ticket status and link to survey. This can be done via 2 services (email notification service, and SMS notification service) or one service (notifier service).

Decision
We will use one service to handle the responsibility of generating and sending different types of notifications.

We made an assumption that the throughput for notification service using SMS and email would be similar. If the performance metrics for the two services would be significantly different, then we would split the notification service into two services, namely SMS notification service and email notification service.

Consequences
The service will need to have two protocol adapters, one for each communication protocol.

4.2.6 Customer Profile Service Granularity

Status (Proposed, Accepted, Superseded):
Proposed

Context
Business rule: have a transaction record to commit rollback or password and credit services as separate secure services - you can't have an entry in the profile and no entry in the credit card and password information

Decision
Combine profile, password, and credit services into one customer profile service with the assumption that a transaction record to commit rollback is required. If there is a business requirement to have additional security outside of the standard security practice, then we will separate customer profile into three services.

Consequences
Simplification of design based on commodity products

4.2.7 Use of Adapters

Status (Proposed, Accepted, Superseded):
Proposed

Context
Customers can purchase

Decision
Combine profile, password, and credit services into one customer profile service with the assumption that a transaction record to commit rollback is required. If there is a business requirement to have additional security outside of the standard security practice, then we will separate customer profile into three services.

Consequences
Simplification of design based on commodity products