

**Course: Service Oriented Architecture**

**Course Code: IS434**

## **SOA Case Studies**

**Presented by:**

**Dr. Dina Ezzat**

# Outline

- Case Study 1: Rainyday Grocer
- Case Study 2: Document Management System
- Case Study 3: Retailer System

# Case Study 1: Rainyday Grocer

- Rainyday Grocer (RG) is a cloud-based online grocer that provides customized delivery of groceries. Their business model is targeted towards those “rainy day” moments where a person needs groceries but is unable to go to the grocery store to shop.
- Rainyday Grocer supports three different ordering options. A customer can:
  1. Place an order with one of the approved grocery stores, send the item list and order confirmation to RG via a web interface or a mobile application.
  2. Send a grocery list to RG via a web interface or a mobile application.
  3. Select from RG grocery items list and place an order.

# Case Study 1: Rainyday Grocer (Cont.)

- There are two delivery options:
  1. Doorstep delivery with text message confirming delivery.
  2. Collect groceries order from one of the RG collection points.
- RG does not own any inventory, supply channels, distribution channels or data centers. They leverage other service providers for all services, and manage quality through a careful selection process and service contracts. They maintain a team of 50 people, only 5 of whom are IT-focused, to manage their operations across five states in the Eastern US.

# Case Study 1: Rainyday Grocer

## (Cont.)

- The customer has to accept a set of constraints to place a successful order:
  - The customer must have a valid payment method registered.
  - An order is limited to 10 or less items with a cumulative weight not more than 10 kg.
  - The delivery address must be a valid physical address.
  - The delivery address must be within 15 km of the city center.

# Case Study 1: Rainyday Grocer (Cont.)

- RG is building a SOA-based solution.
- The recommended approach for a SOA-based solution is to drive it from the business needs and address the business processes.
- As RG is an organization consisting of only five IT team members.
- They realize that they will need to outsource this development work.
- **Using service properties discussed before, what are the services needed by RG?**

# Needed Services

- **Account sign up/login service**

- This service allows the user to log in.
- It validates the user's identification and ensures that the account is valid based on a set of business rules.

- **Receive/Process order service**

- This service is focused on receiving and processing the order.
- It is triggered by the successful login of customer and waits for the order list data from the customer.

# Needed Services (Cont.)

- **Provide price estimate service**

- The RG business model is a no-inventory model, hence the price of goods is dependent on the current market price for those goods.
- Therefore, RG provides an estimate range.
- This service periodically queries RG's registered grocers for the current price list.

- **Confirm/Deliver order service**

- This service completes the order process and triggers the supply chain procure grocery process.
- It sends a confirmation to the customer of order completion and instructions on delivery collection of the order.



# RG's Choice of SOA

- In your opinion, does RG's choice of SOA based solution is well suited to its organization's architecture and its business needs? Why?

# Case Study 2: Documents Management System

- Documents are everywhere. Every process has some form of documentation to go along with it and that documentation needs to be managed. Managing these documents is a routine task using traditional options include filing cabinets and envelopes.
- Also people have looked towards electronic document management as a way to decrease costs and reduce errors when managing documents. Instant access to the electronic versions of their documents is also a must. Although this seems to be a rather simple idea, in practice, it creates new problems.

# Case Study 2: Documents Management System (Cont.)

- One of the biggest problems with electronic document management systems is getting non-electronic documents into the system.
- Also, if you have documents from a client who doesn't use the same electronic document management system as you, you still need to get his/her documents into your system efficiently.
- While searching for a document management startup, I came across a problem. The system in use was built to use a simple desktop scanner to send single document uploads from the users' desktops to their servers.

# Case Study 2: Documents Management System (Cont.)

- Although this worked quite well initially, they were constantly being asked if they could handle larger document volumes. Clearly, I needed some way to scale up the document intake, as a single sheet scanner was not sufficient.
- The problem is how to input documents into the existing system efficiently.
- Upon rethinking the problem, the solution became apparent. Inserting documents into the system was a service that should be provided.

# Case Study 2: Documents Management System (Cont.)

- Also I thought to have a website as a separate isolated application from the larger document management system.
- The idea of thinking about this website is that the needed system should have the ability to remotely upload documents using nothing but a web browser.
- Moreover, I realized that we could create a service that processes a package of documents.

# Case Study 2: Documents Management System (Cont.)

- The package of documents could come from any source, and as long as the package conformed to what the service was expecting, we could process documents from any source.
- **Using service properties discussed before, what are the services that need to be implemented?**

# Needed Services

- A service that accepts a packaged group of documents.
- A service that process each package of documents, regardless of the origin of the package.
- A service that accept packages of documents remotely (through a web browser).
- **Describe services that you think that they should be added.**
- A service that could be used to track the status of documents submitted.

# Case Study 3: Retailer System

- In this example we consider the problems of a retailer who has chosen to re-implement certain functions used by the applications in their Point-of-Sale (PoS) terminals as services.
- Today the trading application is developed as a monolithic application with very tightly coupled components but with some components residing on the In-Store Servers (ISS) and some requests are even forwarded by the ISS to servers centrally located in the enterprise.



# Case Study 3: Retailer System (Cont.)

- With the existing move to service-oriented architecture within the enterprise back-end systems it has been decided that some of the capabilities provided by the ISS and central servers shall be exposed to the trading application as services.
- Initially the capabilities that should be considered as services are explained as follows, customer lookup: this is a two-step process from the terminal in that there is a local database in the store of customers who have already purchased items from the store or who have had service performed on items at the store.

# Case Study 3: Retailer System (Cont.)

- If the customer is not found in the store database then the trading application queries a central customer database. Also, there is a need for service scheduling, in this case customers wish to schedule a service on an item they have purchased, currently the trading application has to query the customer database, query the central warranty database to see if the item is under warranty and update the service schedule with the date/time the customer is planning to bring in the item.
- Moreover an inventory check is needed, the service queries first the local database for inventory and if an item is not found, it queries the enterprise database which contains less information on the products but contains the inventory for all stores.

# Case Study 3: Retailer System (Cont.)

- The new ISS service will aggregate the queries as described above, and will also provide real-time updates to the central server to pass inventory change events to the central server and allow queries on other stores to be more accurate. This implies that the inventory service on the central server has the ability to receive these inventory change events while the local service does not.
- **Using service properties discussed before, what are the services needed by the Retailer System?**

# Needed Services

- Customer lookup
  - Does it contain other services?
  - Query local database
  - Query central database
- Service scheduling
  - Does it contain other services?
  - Query customer database
  - Query central warranty database
  - Update service schedule

# Needed Services (Cont.)

- Inventory check
  - Does it contain other services?
  - Query local inventory database
  - Query enterprise/central inventory database
  - Pass inventory changes to central server

# References

- <https://www.opengroup.org/soa/source-book/msawp/p5.htm>
- <https://www.codemag.com/Article/0505081/Service-Oriented-Architecture>
- [https://swi.cs.vsb.cz/RUPLarge/soa.rup\\_soma/guidances/examples/resources/soa\\_model\\_example.htm](https://swi.cs.vsb.cz/RUPLarge/soa.rup_soma/guidances/examples/resources/soa_model_example.htm)

*Thank  
you*



Any Questions?