Cash Acceptance Network Solution - using ISO 8583

Overview of the Project:

The client has a decade of experience in the payment industry. The company provides various customized services and products in payment-solutions and does it by setting up a cash acceptance network. Mindfire Solutions was approached to develop a customized solution to enable the client to run its business efficiently. The constituent modules of the solution had to help establish means to achieve the following:

- Various kind of devices (POSs, Androids) were to be used to integrate with the payment switch
- The devices needed to communicate with payment switch using ISO-8583 protocol.
- Client needed an application between the payment switch and devices to act as a check and have more control in transactions.
- Client also needed a system to interact with the devices for customer management and sales.

Solution:

We designed and developed an application to expose REST web services and act as a wrapper to convert JSON request to ISO-8583 format for communication with payment switch. This led to use of the common REST platform for devices running in different platforms to communicate with payment switch. It was quite difficult to implement ISO-8583 in different kind of devices.

The platform also exposed a web portal for sending messages, promotions, push notifications to target consumers.

About the client:

Name: Confidential **|** **Industry:** Payment Solutions **|** **Location:** Myanmar

Technologies:

Symfony, PHP, Apache, MySQL, Socket Programming, REST APIs, ISO 8583

Description:

The Key Features of the Solution are

- Wrapper for REST and ISO-8583.
- User management and target sales and promotional messages
- Highly secured; Process for API authentication:
    - The content md5 of the request content was hashed and digitally signed using secret key by the client which was then base64 encoded and sent along with plaintext request-content.
    - The hash code was again generated at server side (server has the secret key) using the plain-text content after basic authentication.
    - Both the hash codes (sent by client and calculated at server) were matched for data integration.
- API versioning:
    - The new version was created by extending the existing version and the new changes were added.
    - Associated services(helper functions) were created to handle the additional parameters.
    - This was done to ensure backward compatibility.
- The application was totally managed by us starting from design to deployment.

The Key Success Factors in Execution are:

- Using a project management system like Jira
- Effective planning and execution of sprints
- Adhering to Scrum processes like daily standups to discuss progress and issues
- Regular review meeting with product owners/stakeholders and incorporating feedbacks
- Using Docker to ensure consistency across development environments
- Effective Git branching strategies for developers and releases
- Dedicated QA tested featured branches before pushing to UAT
- Dedicated adequate time to understand requirements and documenting them
- Using SonarQube and peer to peer code reviews for code quality

System Architecture:

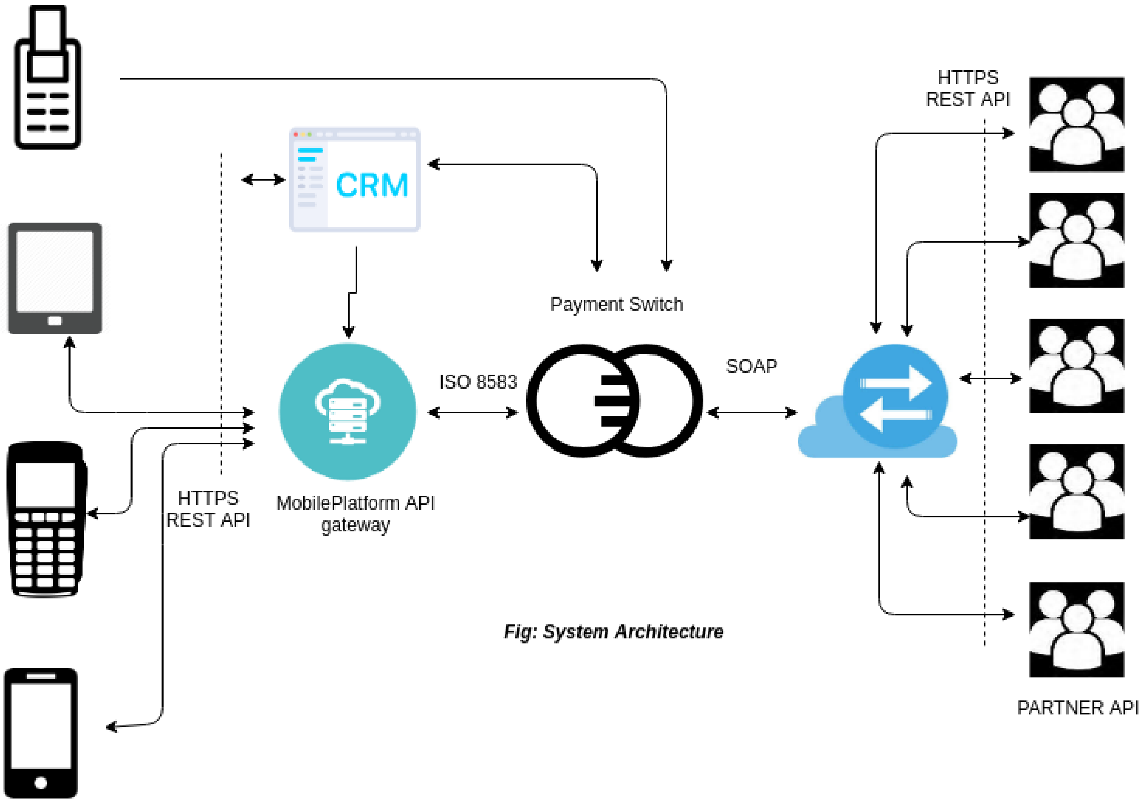

Fig: System Architecture