SINPAY: Simple instant NFC Payments

William Viana Soares Jorge Rodríguez Lería
Bernardo Reynolds Rosa Gutiérrez Escudero

December 19, 2013

Contents

1 Introduction

Our project aims to simplify and optimize payments at small business such as restaurants, cafeterias or clothing stores, etc. The main objectives are to make payments as seamless and transparent as possible to the customers and provide a differentiate experience to those business that implement this solution.

Scenario 1: Restaurant

Imagine that you go to a restaurant and you don't need to pay, just enter, eat and enjoy. The payment will be done after you leave the restaurant, you can check the bill on real time and even share it with your friends right after you leave the local.

You don't need to ask for the bill and wait You don't need to bring money with you You can check the bill at any moment. If you want to share the bill you can do it easily even after you leave the restaurant. Your account is the restaurant loyalty card

At the entrance there is a NFC/QR reader that identifies you. Every time you order a dish the system updates your bill.

Scenario 2: Small store

Go to a business, take whatever you want and go home. In order to let the business know what are you buying you just touch the NFC reader close to each product.

This will not only add value to the customer experience, it will give an opportunity to small business to get to know their customer and apply better marketing campaigns.

A more visual description of this app can be found at our landing page `http://sinpay.me`

2 Architecture Overview

In this section we explain the architecture of our system at the highest possible level and how the building blocks fit together. We also explain what is going to be our application stack and the reasons behind the decisions.

To allow seamless payments for the customers everything should be done behind the scenes. A tablet app in the business should be able to check a customer in, using his mobile NFC, a QR code (in case of iPhones) or even an NFC card. Once the customer is registered at the venue, all purchases can be added to the customer account with the business app. After the customer leaves the venue, the business app will send a payment order to the server, which will process the transaction and send a confirmation both to the customer and to the business.

Business app

The business app is the one that will allow owners to charge their customers with *sinpay*. This app will register a customer at the venue using either NFC or reading a QR code from the customer app.

For restaurants and cafeterias, the app will allow the venue owners to add their catalog of products and prices. This catalog will allow them to add the products to a customer bill. For other kind of business such as shops or supermarkets, the app will be able to record the price of their products in NFC stickers using the device NFC capability. Those stickers can be provided by us or can also be bought in any other place.

Once a customer leaves the venue, owners can order a payment charge at anytime. Business owners will be able to check all past transactions.

Business owners will have to provide a bank account where they will receive the payments.

Customer app

This app will allow customers to pay seamlessly in the venues that use *sinpay*.

For the restaurant/cafeteria scenario, the only thing a customer with the *sinpay* customer app needs to forget about payments is to check in at the venue. For that, if the customer has a device with NFC she will only need to approximate it to the venue device and confirm that she wants to check in. In the case that the device doesn't have NFC capabilities, the customer will be able to check in using a QR code from the app.

The small store scenario is a little bit different. The customer will also need to check-in when she enters the venue, but to purchase products she will add them to her virtual basket using the app, either scanning a QR code or placing the device with NFC over an NFC sticker in the product.

After the check-in the customer can check the current bill at any time. Once the customer leaves she will receive a push notification informing her of the charge that will be made.

The customer will also have the possibility to split the bill with her friends using the *sinpay* app.

Customers will need to have a credit card associated with their *sinpay* account.

Server side backend

The server side backend app will be responsible of the logic behind the payment system. It will have the following responsibilities.

- Logic to validate payments
- Keep track of customers personal data and historic billing
- Keep track of business products and historic billing
- Notify customers about the payment that is about to be charged

The server side app will be deployed in the FI-Ware cloud infrastructure. Our Blueprint will include a tier with the redis database as our main purpose storage. We chose redis because our data won't have many relations and because this way we can avoid having to deploy a cache solution such as memcache. Additionally, we will also need an application tier with Django and Apache to deploy our backend app, which will control the payments logic as well as to provide the business app with a catalog of products.

For the authentication of the client apps, both customer and business, we will use the Access Control together with the Identity Management Generic Enablers.

3 FI-WARE Generic Enablers Usage

In order to authenticate customers and business apps we will use a Generic Enabler called Access Control combined with the Identity Management GE. With these two GE's will have the benefit of not having to deal with registration, login and even user profile storage. Another benefit is the compliance with the OAuth protocol.

This way we will be able to focus on the core functionality of our applications and accelerate the development process. Moreover, it will allow for future integrations of this authentication system with other Generic Enablers.

We would like to use Access Control to allow customers to configure a range of hours between he can use the payment system and to set only a list of shops where she wants to allow payment.

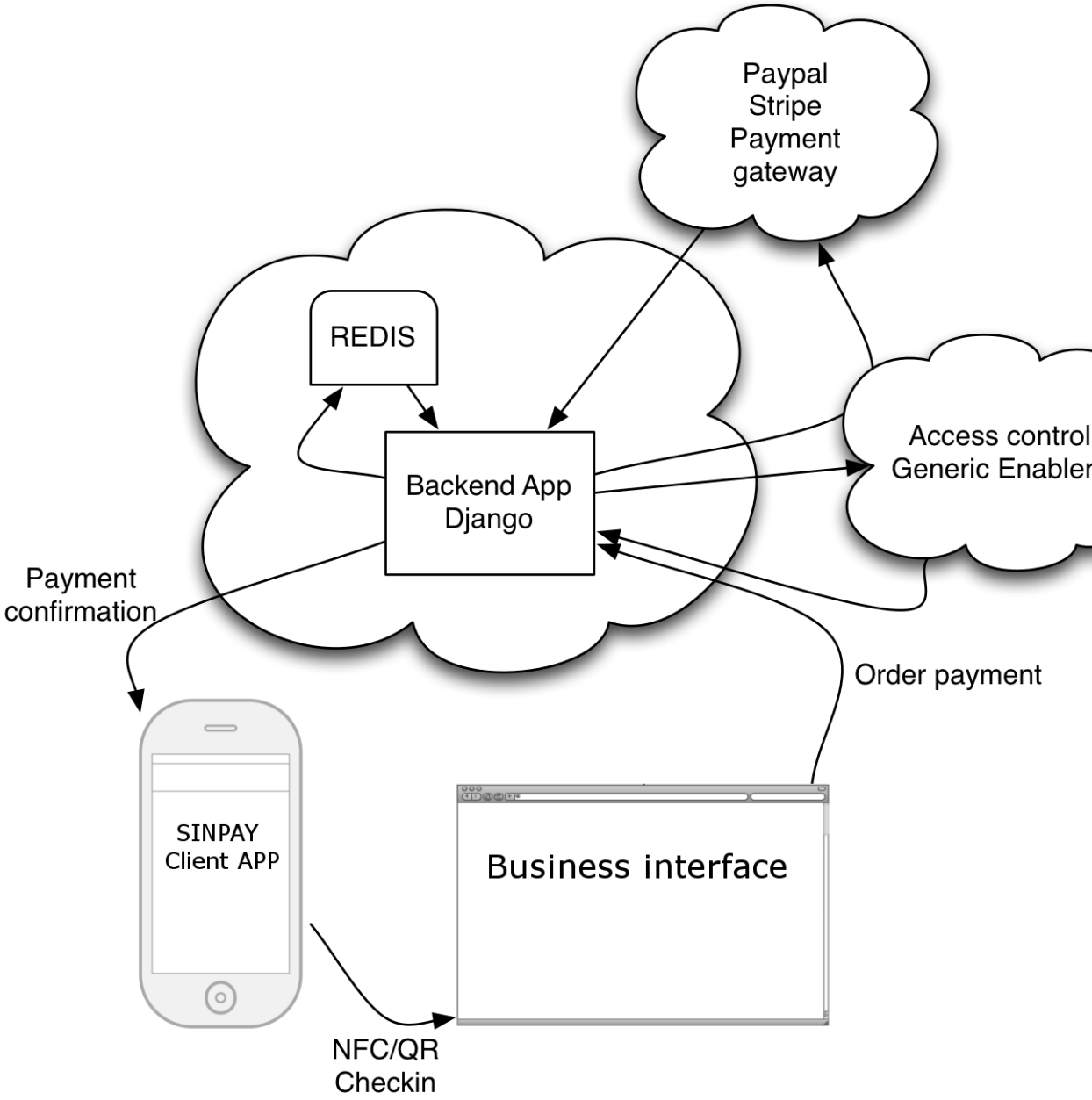

Figure 1: Architecture Overview

4 Other Third Party Software

Payment System

To speed up development of a MVP we will use a third party payment system such as *PayPal* or *Stripe*. At first we would like to use *Stripe*, but at the moment of writing this document, it is still in private beta in Spain, so we will probably start with *PayPal*. Both companies provide a framework to easily integrate payments in our apps in exchange for a small percentage of the transaction.

Once we validate the idea, the next step would be to implement the payment system ourselves so that we can have a better profit margin.

Client Apps

As mentioned before, we will build an app for business owners and another for customers. We decided to focus on the two main mobile platforms in the market at the moment; iPhone and Android. These apps should be developed natively in each platform to take advantage of the devices capabilities such as NFC in Android and the push notifications in both platforms.

Push notifications is one of the core functionalities of our product. It will be used to notify the customers about the payments that are about to be charged on their account. Therefore we would need to implement server side push notifications that use Google and Apple infrastructures. This could also be a good opportunity to develop a generic solution that could be later used as a Generic Enabler.

5 Business Model

We aim to open the protocol and release the code as free software once it is in a mature state so anyone can implement it. This approach instead of closing doors will open business models.

Software support

We can offer support services to anyone interested on implementing it. By helping with installation, configuration and scalability for big business.

Hardware support

The basic implementation could be done by a small tablet connected to the internet, but we plan to offer support to specific hardware like vanilla NFC readers connected to regular PCs or small devices like Raspberry PI.

New functionality

By developing plugins or modifications we can implement specific features by demand like special offers, tracking modules, stats, alarms or just interface modifications.

Hosted service

Sinpay could be installed in the business server or used in our hosted service for a small price depending on the volume. This model will be freemium.

6 Impact

Right now offline payments are a difficult process:

- You need to have funds to pay
- In order to pay you have two options:
  - Carry enough money with you
  - Have a working credit card
  - In some cases wait for the bill, split it (or not)

It takes time, you need to think about the payment process every single time.

If a business implements *sinpay*, the payment process is invisible, you can leave at any moment and the bill is accessible in real time (if you want to check it) and stored securely.

The business owner will only work to make your experience better and the customer will buy in a faster and secure way.

Sinpay is the Amazon One-Click button equivalent in the real world, and it can benefit small stores because of compulsive buying;

7 Innovative Aspects

It brings the best of the online payments to the real world:

- Clients can go to a restaurant or other business without thinking about money amore than the strictly necessary.
- It is fast and saves time by skipping the traditional payment flow.
- Businesses can apply special discounts to loyal customers.
- Clients can split the bill between their friends after leaving the place.
- It's more secure than money: you can deactivate your *sinpay* account, make it accessible only for a range of time or selected business

8 Development schedule

Prototype

Develop a working prototype can be done in a week by four people, but this development will be fragile, far from perfect and only usable for demo purposes. The first implementation will not have access control or option to share the bill and it will use Android phones, tablets and NFC stickers as hardware.

Comercial version

As we all work developing software on a daily basis we all know that it's impossible to give a number or a realistic development schedule in a product as critical as the one we are presenting. It needs to be polished and go through an iterative process.

Assuming we are 4 engineers working full time 40 hours a week a beta can be done in a few months but it will take at least half year (being optimistic) to have a working version usable, secure and working in a few devices.