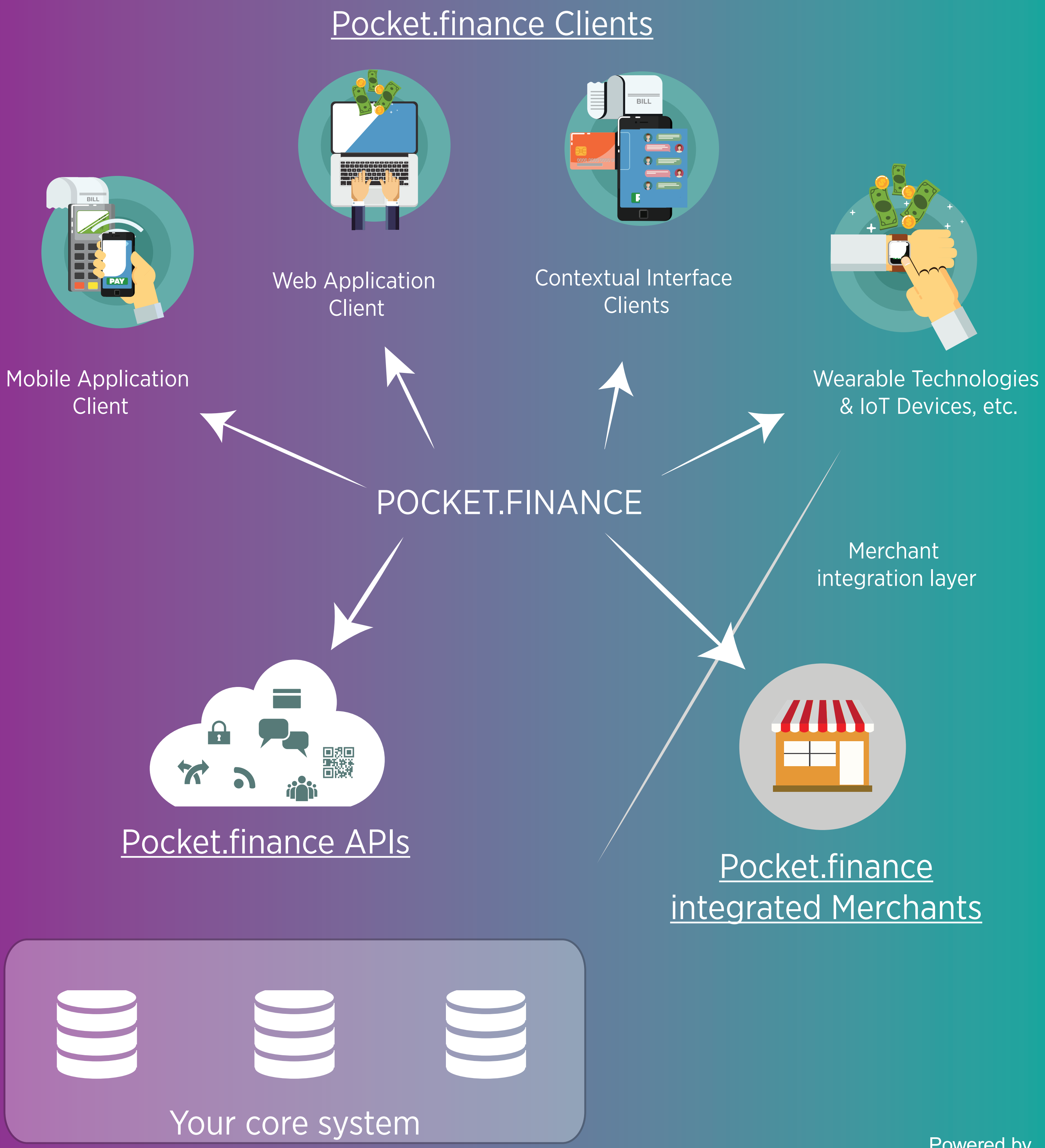
What is Pocket.finance?

Pocket.finance is an API based, digital Wallet solution. By using Pocket.finance you can create your own wallet working on multiple devices!



Pocket.finance Clients

Mobile Application Client

Web Application Client

Contextual Interface Clients

Wearable Technologies & IoT Devices, etc.

POCKET.FINANCE

Merchant integration layer

Pocket.finance APIs

Pocket.finance integrated Merchants

Your core system

Powered by

FINARTZ

★ %100 API Based

Consider every wallet service for your business as an API!

User APIs

Merchant

Customer

User APIs will be used during the time of Login or Signup of a Merhant or Customer.

Transaction API

Transaction APIs finalize the payment and convert them to a purchase. Transaction APIs will use Payment amount, User information and Account information to check and execute whether it is possible for the purchase to be successfully completed.

Open ID API

Open ID API makes sure that all of the Pocket.finance clients are talking with back-end securely.

Card API

Balnace API

Account APIs

Bitcoin Wallet

Ethereum Wallet

Account APIs will be holding asset types to fulfill payments, like adding/removing Cards, defining Balance Accounts or other types of accounts such as Bitcoin Wallet, Ethereum Wallet, etc. Pocket.finance can work with ‘Smart Contracts’, make settlement from over Ethereum Platform.

Money Transfer API

Payment Sharing API

Payment APIs

Recurring Payment API

Bill Payment API

Payment APIs will be called by clients’ applications to create **“payment request”** when Customers want to purchase something.

Transaction = Payment * Customer * Account

Users

Card
Balance
Bitcoin Wallet
Ethereum Wallet

(Working logic behind a transaction)

Powered by

FINARTZ

Accessible from everywhere: Pocket.finance Clients

Wearable Technologies & IOT Devices

Pocket.finance application is accessible from IoT devices and Wearable Technology devices such as Apple Watch. This means, for example in the near future, your IoT compatible refrigerator might order food by using Pocket.finance application.

Mobile Application Client

Mobile Application Client is the platform where Users interact with Pocket.finance through mobile devices.

All In One Solution
Working on Multiple Devices

Web Application Client

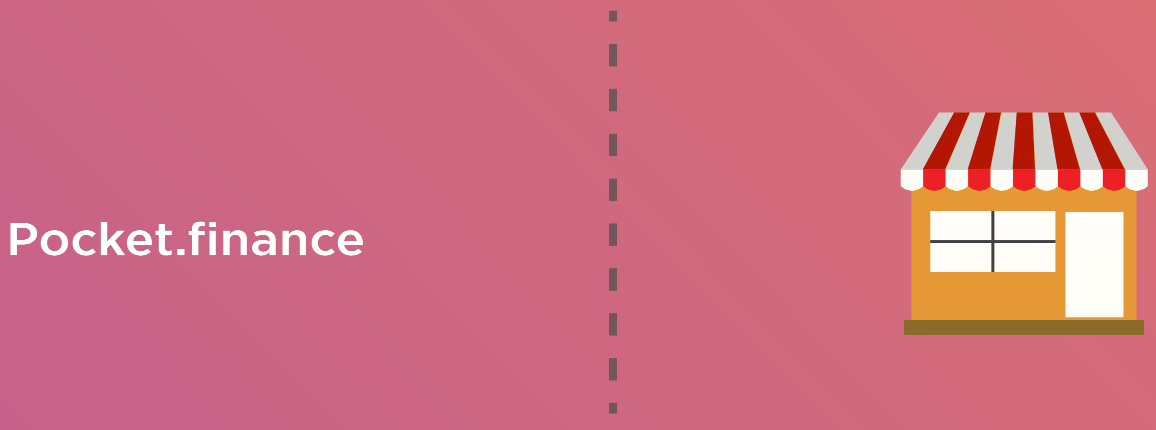
Web Application Client is the platform where Users communicate with Pocket.finance from over PC either using browser or a native client.

Contextual Interface Client

Contextual Interface Client is the platform where Users communicate with Pocket.finance through text or voice. Some examples are Amazon's Alexa, Apple's Siri or Chatbots.

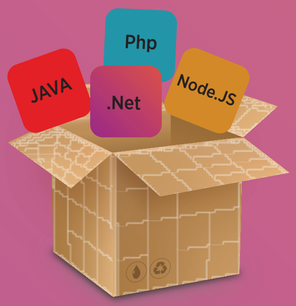
Powered by
FINARTZ

★ Fast integration methods with merchants



Pocket.finance

Merchant integration layer

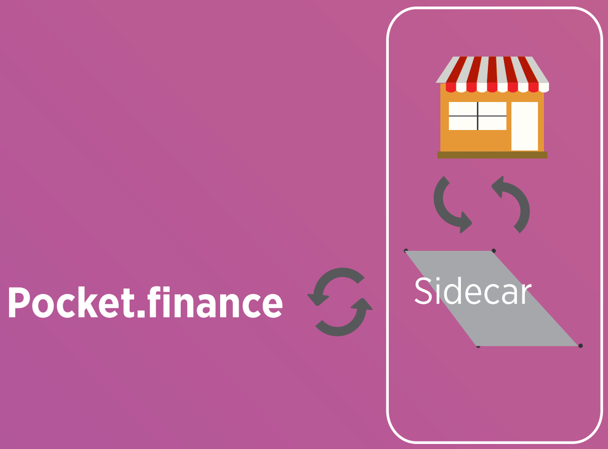


Pocket.finance SDKs
(Software Development Kit)

SDKs are code packages that can be easily implemented and modified by Merchants. The SDKs are available for Java, Php, .Net, Node.JS languages

Pocket.finance online integration documentation, provides Merchants a detailed guide for integration processes. Pocket.finance documentation is a living documentation. It is updated regularly if any change occurs.

Online Documentation



Pocket.finance

Sidecar

Sidecar is a SDK of Pocket.finance that makes integration easier for Merchants. Sidecar stores some Merchant information at the beginning of the integration process. There is less integration left to Merchants.

Powered by FINARTZ

★ Pocket.finance Features

Push Based Payment Request

Pocket.finance makes buying easy with Push Based Payment Requests. An example of Push Based Payment Requests, with Pocket.finance you can complete your online payments at your Mobile device even though you initiated buying process at your PC. Payment request comes to your mobile device as a push notification. This feature will provide you convenience since you won’t need to transfer the OTP(One Time Password) from your mobile phone to your PC.

Programmable Wallet

With Pocket.finance,By analyzing various APIs’ “events”, “Merchants” can create loyalty prograns and campaigns. Similarly, “Customers” can use this feature to make great decision mechanisms. For example, a user can limit his/her budget spend on coffe weekly. When the limit is reached he/she gets a notification telling ‘weekly limit for coffe spending has reached’. Similarly you can make many more great mechanisms by using Pocket.finance.

QR Code Payment

With Pocket.finance application, QR code payments can be done in different ways. For this you need one QR code generating device and one QR code reading device. In both Pocket.finance application should be integrated.

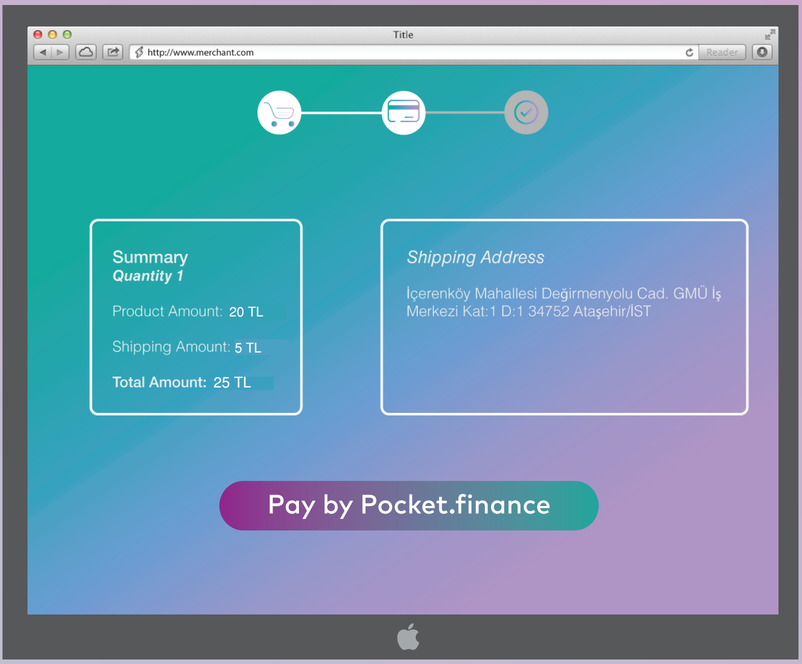
NFC Payment

With Pocket.finance application, you can make NFC payments easily. For this, you need two NFC-equipped devices and Pocket.finance should be integrated with both devices. You can transfer money between two NFC-equipped mobile phones using your balance accounts, or you can complete your payment at the checkout counter of a store which has NFC-equipped payment terminal.

Powered by

FINARTZ

★ Pocket.finance Use Case 1: Web to Mobile Payment



Title

http://www.merchant.com

Summary
Quantity 1

Product Amount: 20 TL

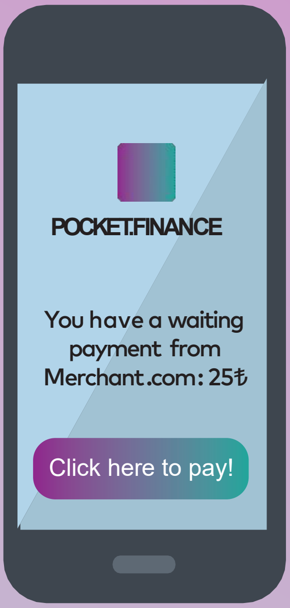
Shipping Amount: 5 TL

Total Amount: 25 TL

Shipping Address

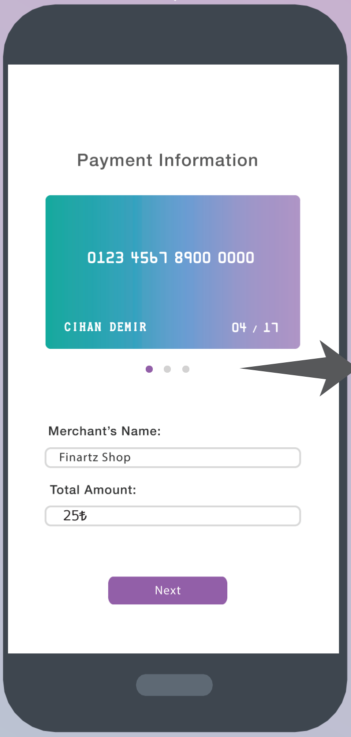
İçerenköy Mahallesi Değirmenyolu Cad. GMÜ İş Merkezi Kat:1 D:1 34752 Ataşehir/İST

Push Based Payment Request Comes to Your Mobile Phone



POCKET.FINANCE

You have a waiting payment from Merchant.com: 25₺



Payment Information

0123 4567 8900 0000

CIHAN DEMIR 04 / 17

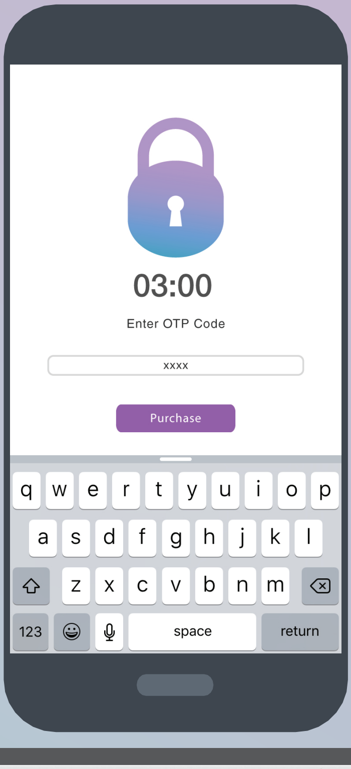
Merchant's Name:

Finartz Shop

Total Amount:

25₺

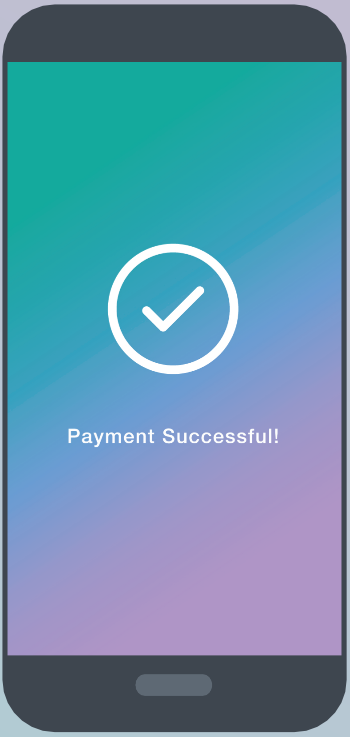
Different payment options



03:00

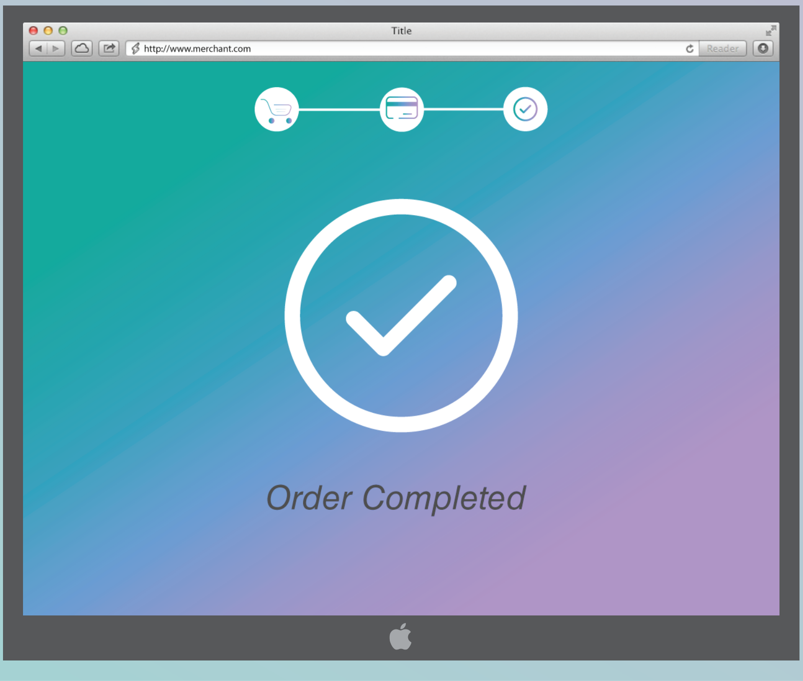
Enter OTP Code

xxxx



Payment Successful!

Simultaneous Confirmation at Web



Title

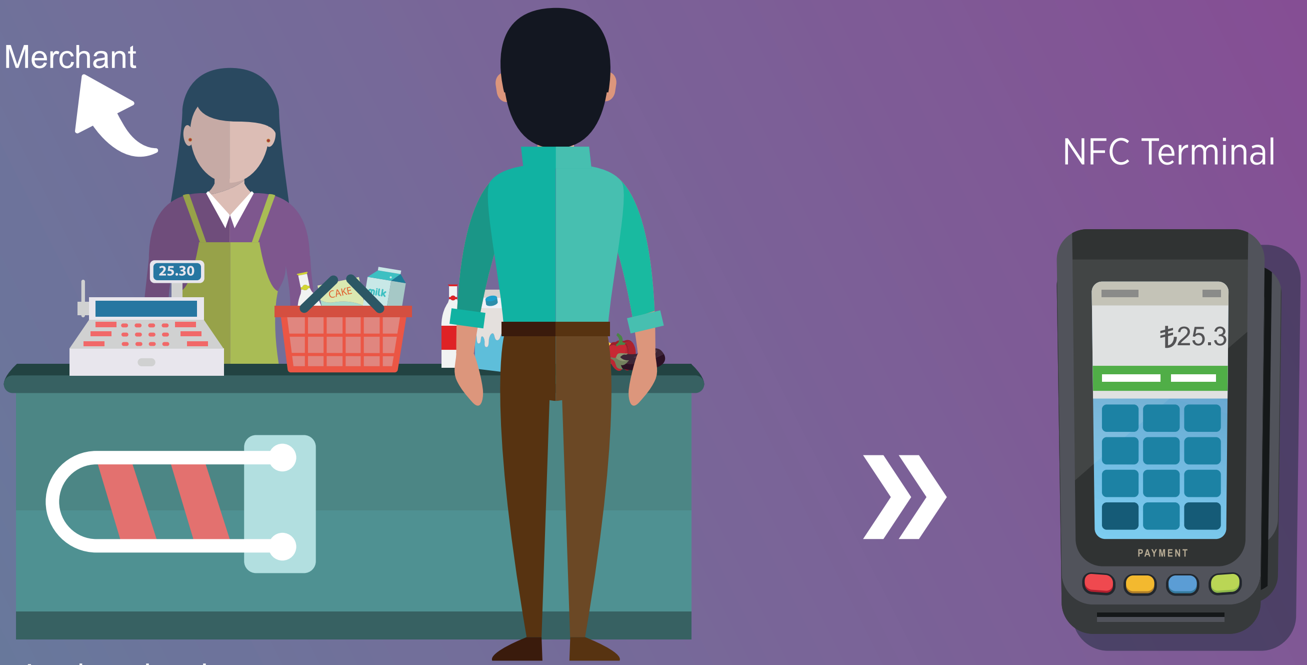
http://www.merchant.com

Order Completed

Powered by

FINARTZ

★ Pocket.finance Use Case 2: NFC Payments



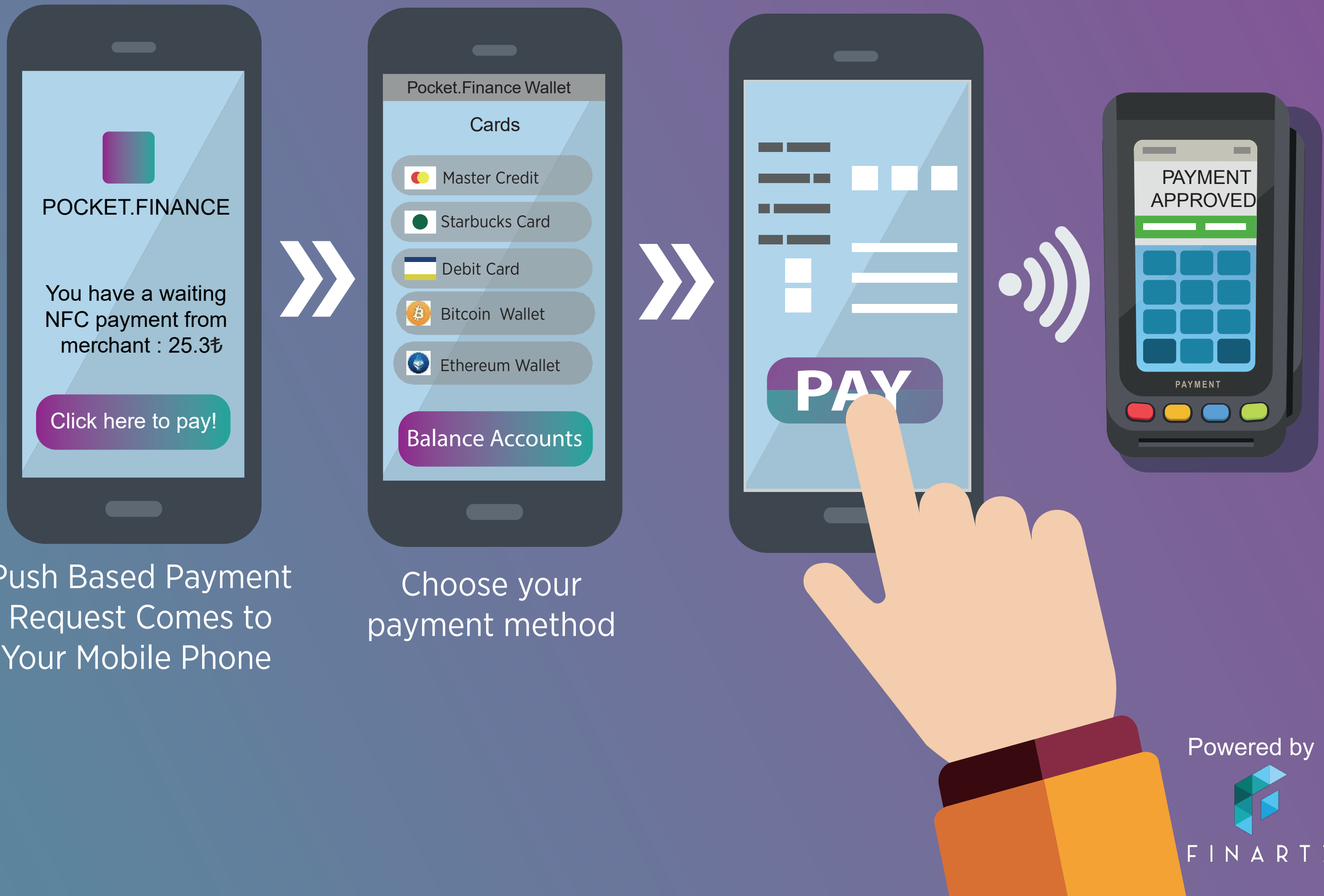
Merchant

At the checkout counter

Customer

NFC Terminal

Merchant initiates transaction



POCKET.FINANCE

You have a waiting NFC payment from merchant : 25.3₺

Click here to pay!

Pocket.Finance Wallet

Cards

- Master Credit
- Starbucks Card
- Debit Card
- Bitcoin Wallet
- Ethereum Wallet

Balance Accounts

PAY

PAYMENT APPROVED

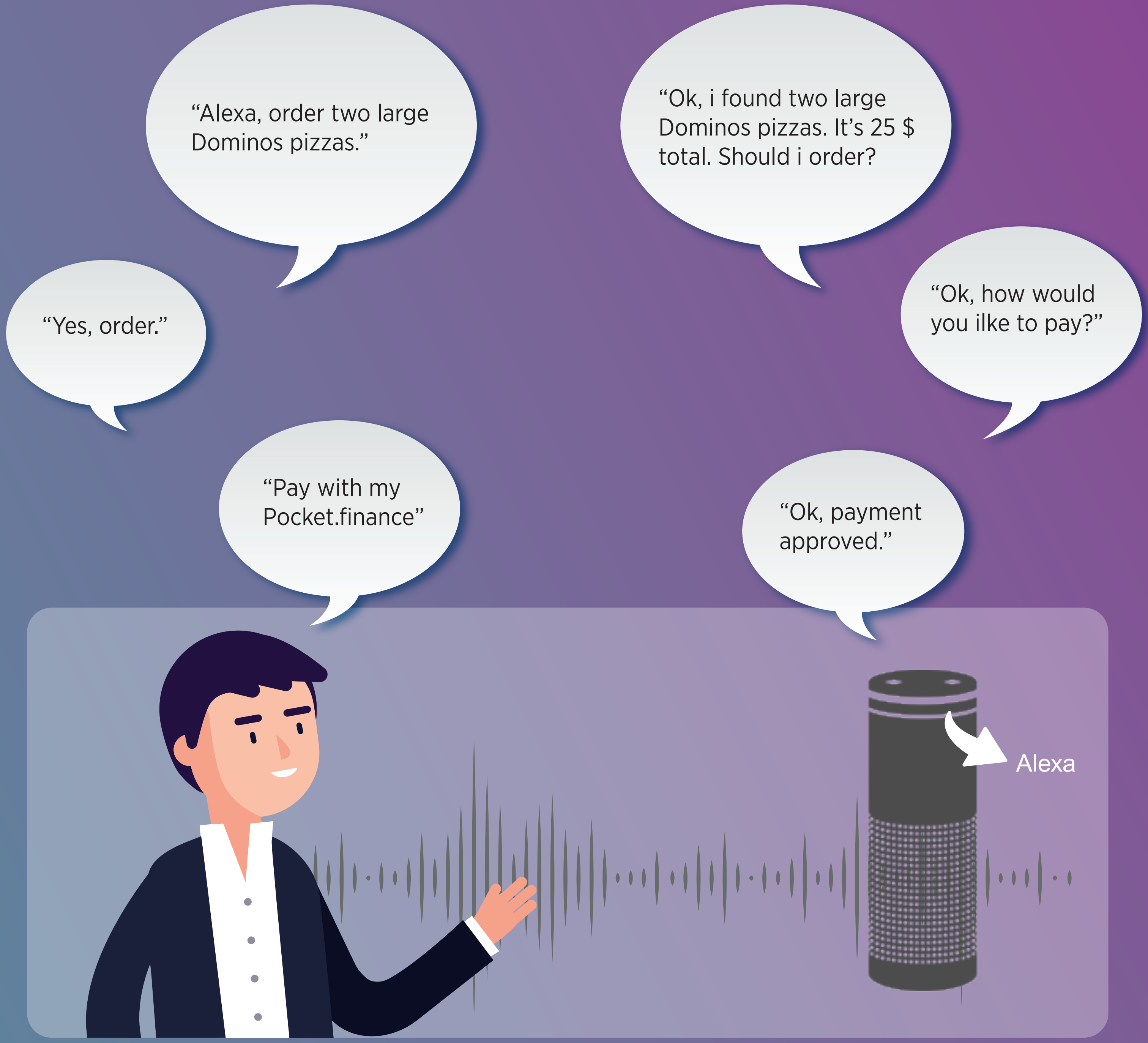
Push Based Payment Request Comes to Your Mobile Phone

Choose your payment method

Powered by

FINARTZ

Pocket.finance Use Case 3: Payment with IoT Devices, Alexa


"Alexa, order two large Dominos pizzas."
"Ok, i found two large Dominos pizzas. It's 25 $ total. Should i order?
"Yes, order."
"Ok, how would you ilke to pay?"
"Pay with my Pocket.finance"
"Ok, payment approved."
Alexa

Powered by

FINARTZ

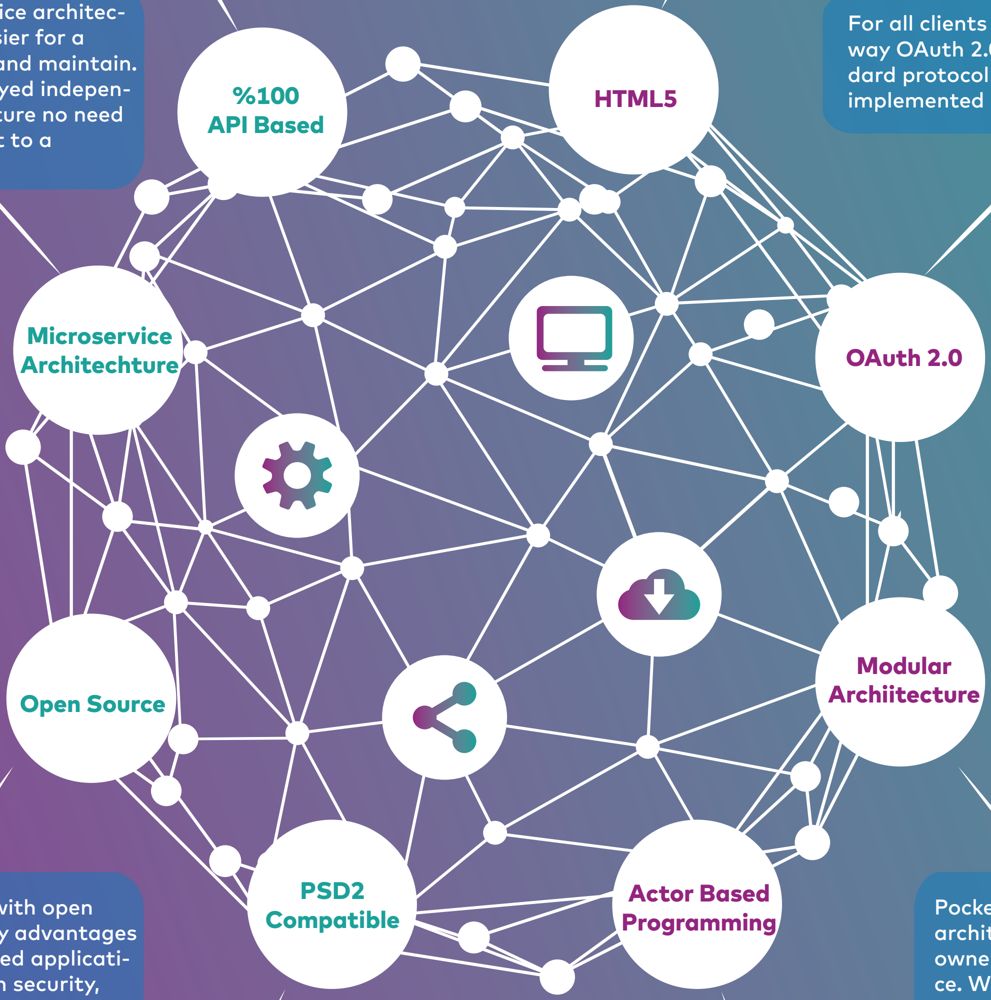
Technolology behind Pocket. finance

Pocket.finance is API based at backend so it is easily modified, faster integrated and deployed. Compatible with new API based systems such as Alexa. But also in frontend it's a Single Page Application so User Experience is better compared to traditional webapps and redirect based solutions.

Front-end development of Pocket.finance is based on HTML5. HTML5 is becoming a new standard, it is cheaper, faster and supports mobile devices. HTML5 is modern and lets you do things previously impossible.

With its unique microservice architecture, Pocket.finance is easier for a developer to understand and maintain. Each service can be deployed independently. With this architecture no need to long-term commitment to a technology stack.

For all clients to connect in a secure way OAuth 2.0 (the industry-standard protocol for authorization) is implemented in Pocket.finance.



Pocket.finance is coded with open source softwares. The key advantages of being open source coded application are, reduced cost, high security, high software quality, customizability, interoperability.

Pocket.finance's modular architecture enables Wallet owner, to modify Pocket.finance. Wallet owner can add new extensions to Pocket.finance both at front-end and back-end.

PSD2 is the second Payment Services Directive, designed by the countries of the European Union. Pocket.finance API designs are compatible with PSD2.

As Message Passing Concurrency patterns are used with an actor based programming framework (Akka) we achieve safe, scalable, reactive and distributed API services in Pocket.finance. Akka is a very innovator library so that Pocket.finance has the opportunity to be compatible with new Technologies.

Powered by

FINARTZ