

IONIC 4 as mobility platform for iDempiere

Lyon 2019

Norbert Bede



cloudempire
CRM, ERP, PSA, WMS



About me

- Founder: MULTIMAGE 2004 / Cloudepriere 2018, private ownership
- Languages: English, Slovak, Hungarian
- Territory: Slovakia (SK), Czech Republic (CZ), Hungary (HU)
- 15 years of experience with business consulting and IS integration
- Customer : ERP, WMS, CRM, PSA, TMS
- Our main Product: Cloudepriere - SaaS



The background image shows a Zebra handheld mobile device, likely a rugged smartphone or PDA, held in a hand. The screen displays a software interface for inventory management. At the top, the Zebra logo is visible. Below it, there's a navigation bar with a back arrow and the text 'Riadky Inventúry'. The main screen shows a list of inventory items with columns for 'Typ Dokladu', 'Fyzická Inventúra', 'Sklad', 'Tripsy Sklad', 'Všetky (3)', 'Potvr (0)', and 'NePotvr (3)'. The list includes items like '10. Vrecia na odpady modre', '20. Kopirovací papier A4 80g', and '30. Vrecia do koša 60l 60x70 cm'. The device has a full QWERTY keyboard and several function buttons.

Our Project Goal

**Develop a Standalone Mobility Client
built on *.js client side technologies**

Research of Mobile Frameworks

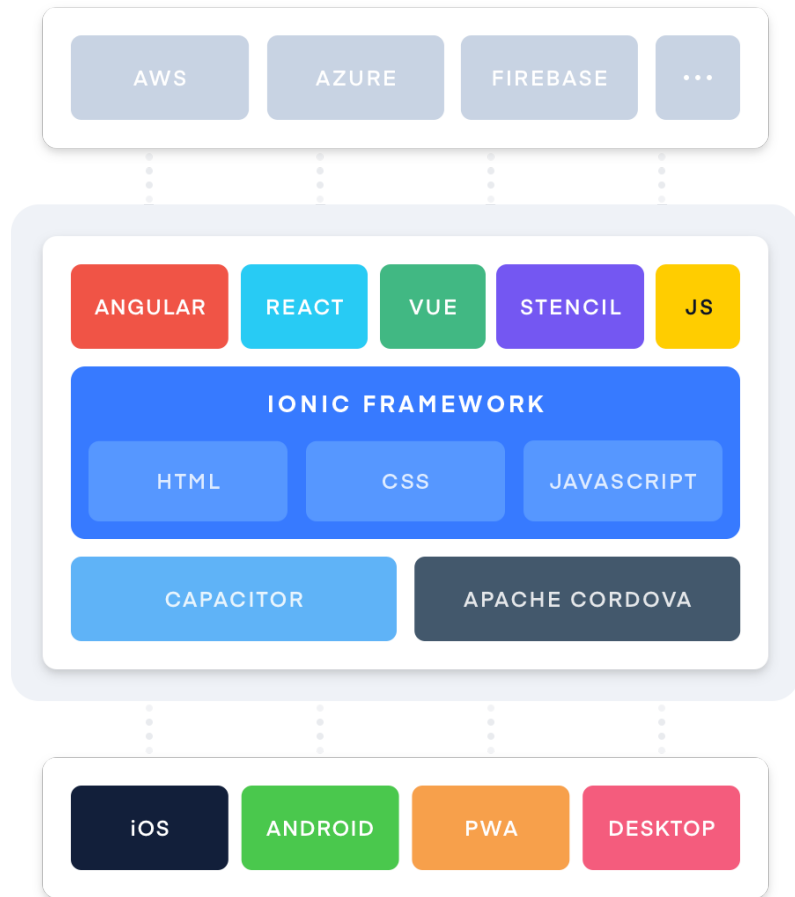
**Native Development
(Swift, Android)**

**Hybrid Application
(ionic)**

Other (React)

Why we decided for ionic?
www.ionicframework.com

What is Ionic Framework?



A complete app development kit.

- library of front-end building blocks and UI components
- easy to design beautiful, high-performance mobile and Progressive Web Apps using web technologies like HTML, CSS, and JavaScript

Web components pair with any JavaScript framework, including Angular, React, Vue, or no framework at all (just add a script tag!). Ionic apps are backend agnostic, with connections to AWS, Azure, and Firebase.

Our Project Scope

Goals

- Online / Offline Support
- Sync / Async communication to the server
- Multi-User support
- Follow iDempiere principles and business logic
- Keep Security Industry Standards (OAuth2, JWT)
- Never connect to db directly - always apply iDempiere business and validation logic (MClasses)

Not Goals

- Our goal was not make client I:I and copy iDempiere objects - instead freedom case base case

UX

- Do not implement complex meta layer like Application Dictionary

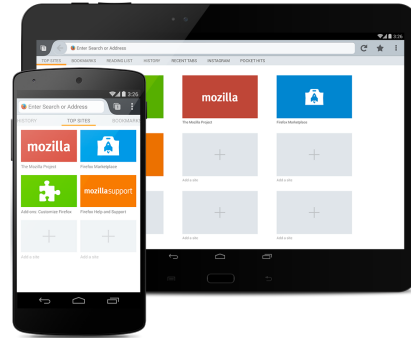
Our vision about Applicability



Enterprise devices (mobile computers)

Example usage:

- Product/Locator/BP Info
- WMS operations
- Ship Confirmations
- Logistics operations
- 2D Shuttle (upcoming)



Commercial devices (mobiles, tablets)

Example usage:

- CRM (lead, activites..)
- Mobile POS Solution
- KPI (cube.js)
- B2C Client
- logistics: drivers (GPS)



Terminals/Consoles (Legacy, Tablet, Rapsberry)

Example usage:

- POS (upcoming 2020/Q1)
- WMS Console (Electron Build)
- JIS Console (KIA)



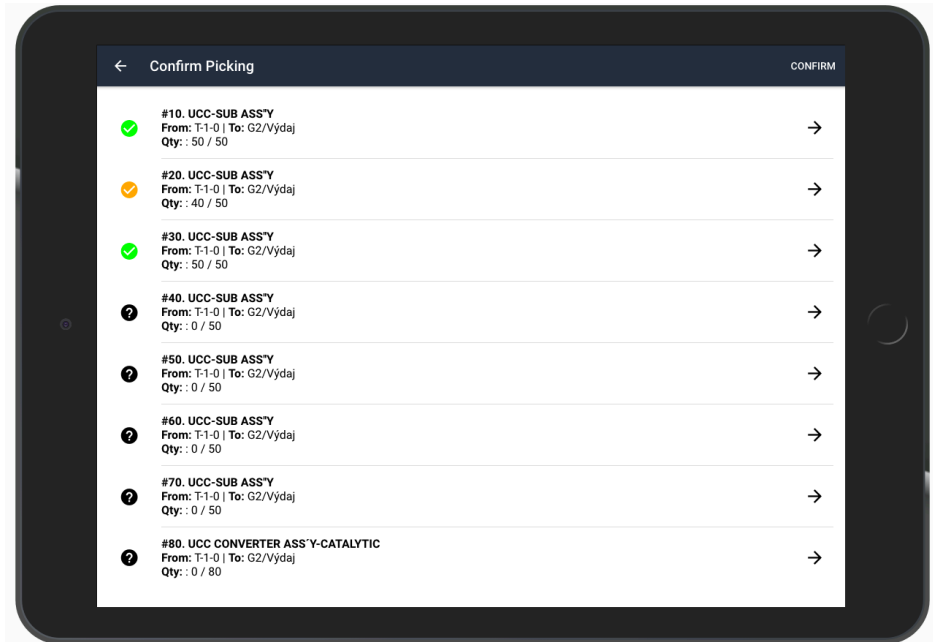
Presentation Devices (panels, kiosk, RPi)

Example usage:

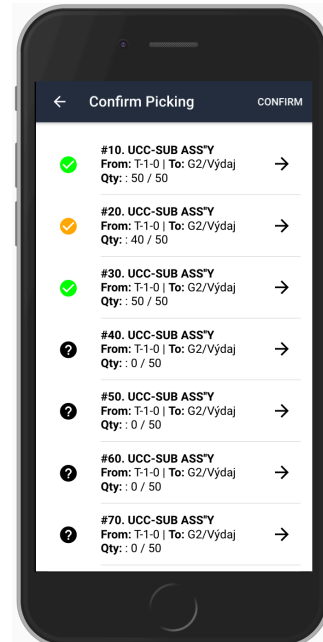
- Operation Dashboard
- Customer KIOSK
- KPI Dashboard

Example Applications

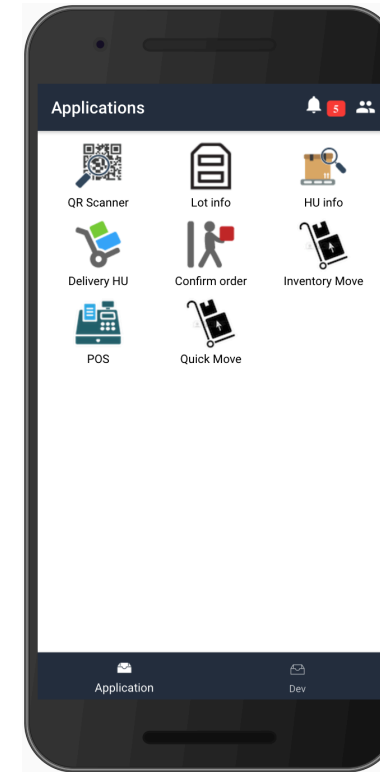
iPAD Confirm App



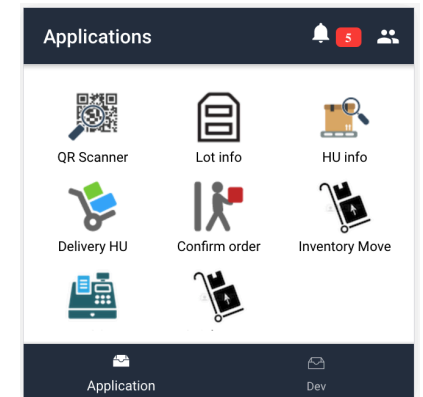
iPhone



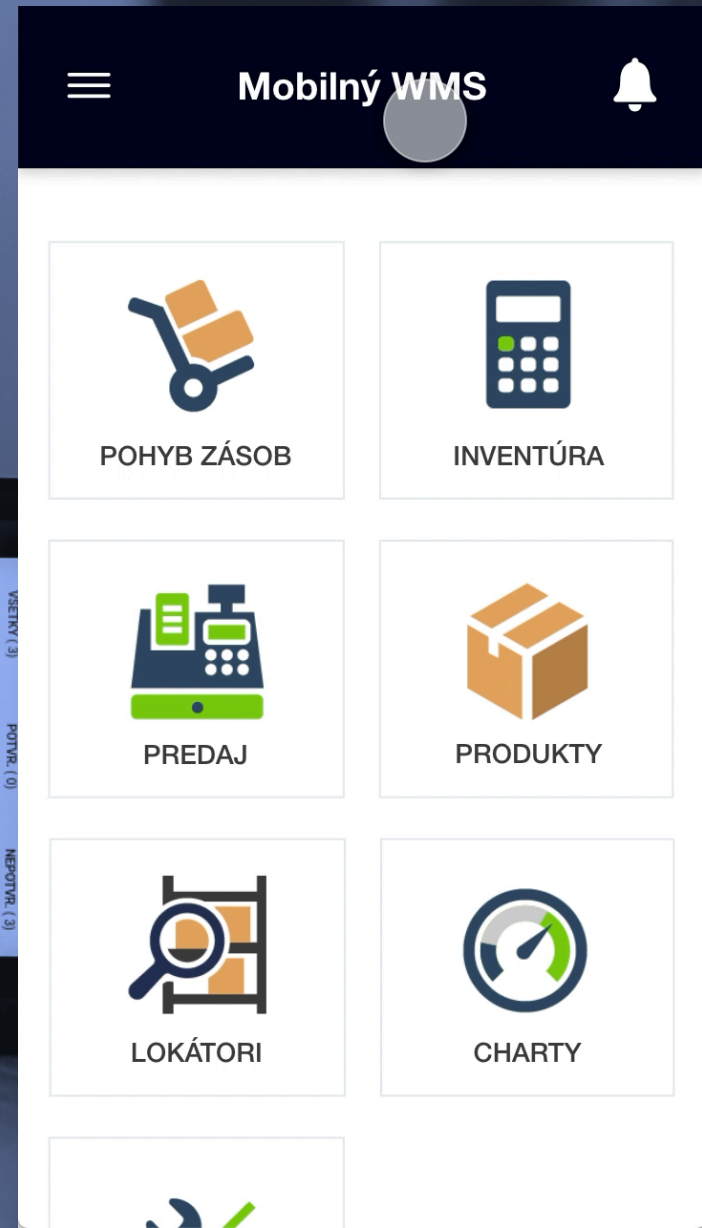
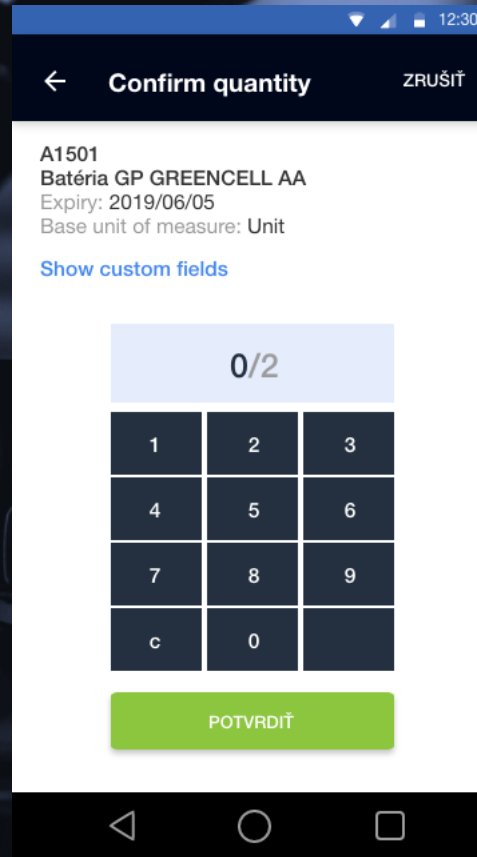
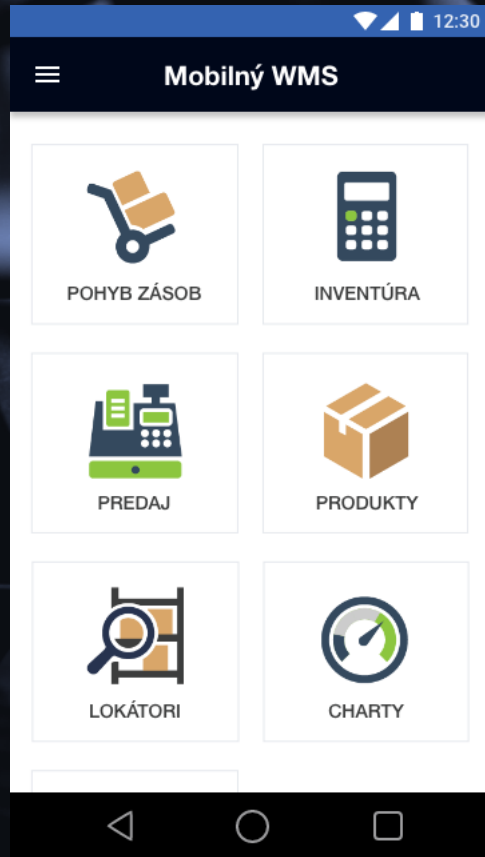
Android



Motorola MC3200



Continuous UX improvement

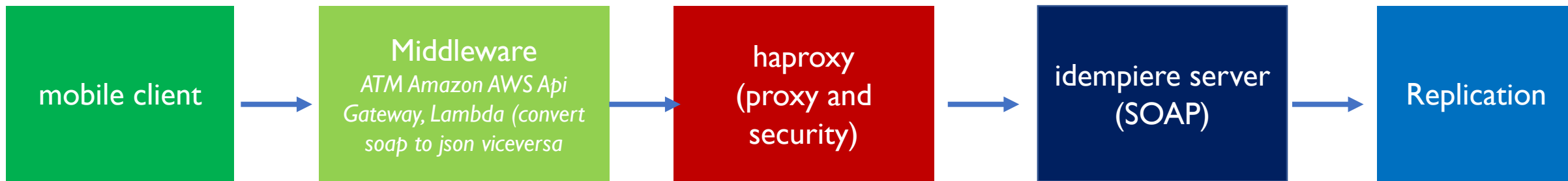


Generic, best practice features

- Login by Backend User (cordova plugins)
- Online/Offline Login by PIN
- Login by Scan PIN
- Multi-user/Switch User
- Sync: Init Devices then incremental sync (MQ/Streaming)
- Permission for Application/Role
- Scan: photo, datawedge, RFID
- Barcode/QR code support
- Event based communication
- Multi-Language
- Responsivity
- Intuitive UX
- Manual/Autolock Users

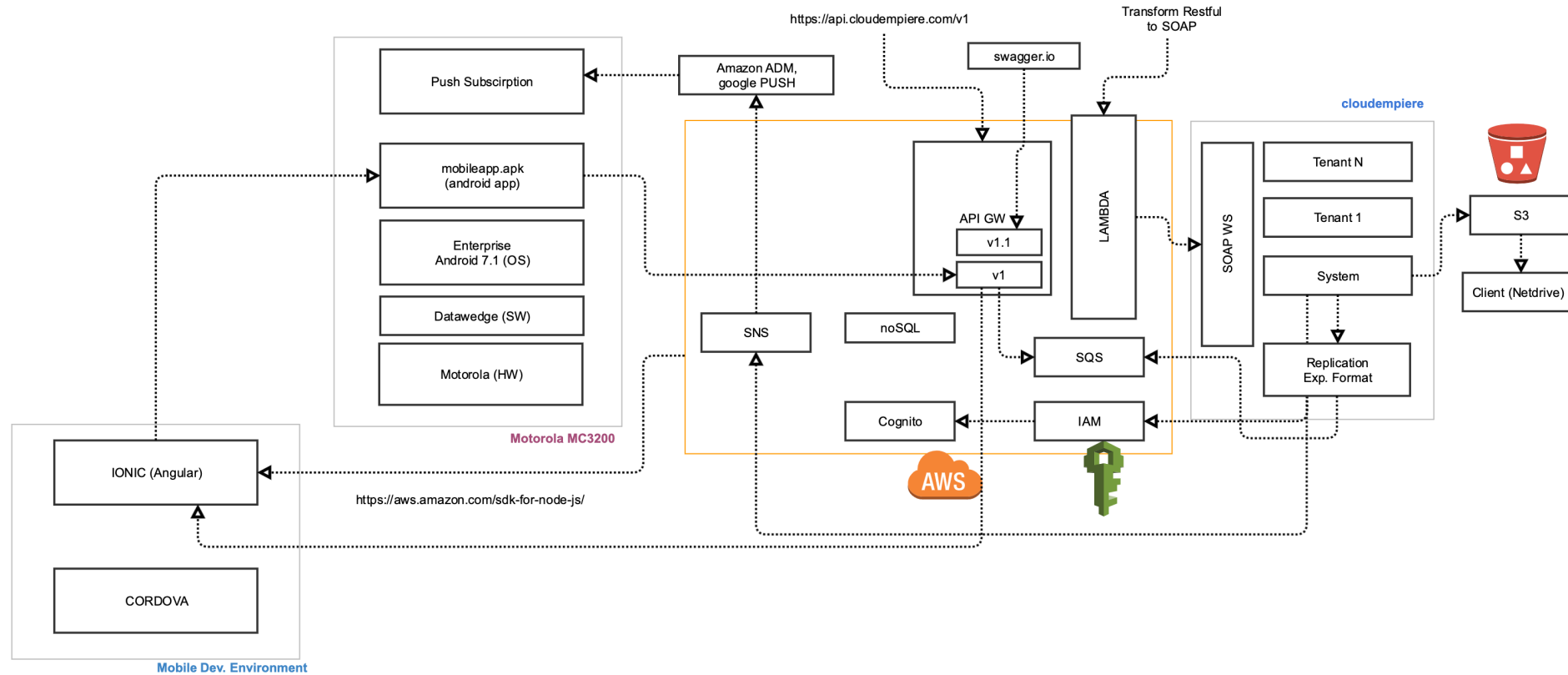
Technical Solution

Our Components



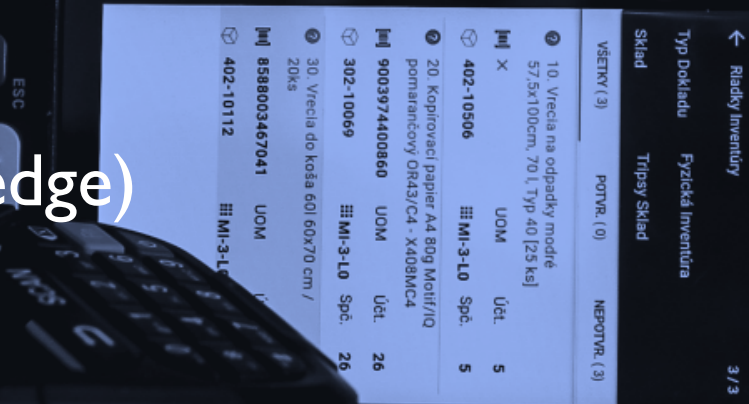
Our Implementation (Core Principles)

Goal of This figure: Purpose is make a very early stage mockup how ideas could be realised



Services/Providers Overview

- PouchDocumentProvider (DocumentStore)
- Document user locks – unreliable user document locks persisted/synchronised by pouchDb
- Authentication Provider
- Async Job Manager
- Scanning Service (Datawedge)
- Filter Service
- DB Retention



The image shows a Zebra handheld scanner with a list of inventory items displayed on its screen. The list includes item descriptions, quantities, and various codes. The Zebra logo is visible at the top of the screen.

←	Riady Inventury	3 / 3
Typ Dokladu	Fyzická Inventura	
Skled	Tripsy Sklad	
VŠETKY (3)	POTVR. (0)	NEPOTVR. (3)
10. Vrecla na odpadky modré 57,5x100cm, 70 l, Typ 40 [25 ks]	UOM	Účt. S
402-10506	MI-3-10	Spč. S
20. Kopírovací papír A4 80g Moll/IQ pomarančový OR43/C4 - X408MC4	UOM	Účt. 26
9003974400860	MI-3-10	Spč. 26
302-10069	MI-3-10	Spč. 26
30. Vrecla do koša 60l 60x70 cm / 20ks	UOM	Účt. 26
8588003467041	MI-3-10	Spč. 26
402-10112	MI-3-10	Spč. 26

Webworkers

Problem: Js is Single Thread

<https://dev.to/steelvoltage/if-javascript-is-single-threaded-how-is-it-asynchronous-56gd>

Solution: CPU intensive non gui processing is offloaded in web-workers

Following processes are running in webworker:

Network detection

Logging (Store in IndexeDB=>REST=>MQ=> Logstash=>Elastisearch)

Messaging (SQS,ActiveMQ)

PouchDB Syncing (<https://couchdb.apache.org/>)

PouchDocumentProvider - Detail

- Conceptually application on device is a viewer/editor of idempiere tree document structure, where root node dialog contains all documents of given type, each child node there is view/edit dialog according the need.
- There are exist the base class/provider PouchDocumentProvider which is being used by each application and which contains basic common functionality as e.g. methods for writing/reading from Pouchdb, retrieving all documents, synchronisation, filtering etc. each application derives own document provider of PouchDocumentProvider, attaching its own business methods.
- For some static data as products and locators there exists common GUI elements which can be used by applications.

Authentication/Security

- We using oauth2 protocol (atm as process will be rewritten as service) Access/refresh token/PIN support
- We are improve Deepak Token Solution - add identity provider and mobile device
- Backend Token generator:AWS Cognito (allow login) - developer token exchange (figure..)

Data Handling

- Backend=IDempiere (PostgresSQL)
- IDempiere documents are replicated/synchronized to/between devices using poudb/couchdb replication facilities
- **mobile storages (available options)**
 - Cookie
 - LocalStorage
 - WebSQL (usage: master data)
 - IndexedDB (usage: master/Static data storage) / replicated to devices using stomp/ActiveMQ.
 - CouchDB/PouchDB (DocumentStore) storage for dynamic data (i.e idempiere documents) - pouchdb
 - AWS Cognito Keystore (Single User not good for Business Application) - will be depreciated

Areas to Improve

- True REST API (atm we converting)
 - *SOAP is not for mobile applications - it is too generic, js coders don't like it, wrong js support (xml)*
- Make it more simple (too many experimental areas)
- More Business Application (real value for customers)
- Community or Private ? (interest to cooperation)

Thank you