

# DOCUMENT MANAGEMENT IN HEALTHCARE

**Objective of the project:** Update an app for easy document management in healthcare by introducing new functions.

Our customer is a company which brings together doctors, scientists and engineers who are focused on improving communication between sites and sponsors. The key to achieve that is by solving narrow tasks on sites. The results that are obtained enable sponsors to conveniently monitor site processes in order to cut their costs and to reduce uncertainty between visits. In this way, our customer strengthens sponsor-site relations while boosting compliance.

## Challenge

The company's app was under active development. The client wished to extend the app functionality with these features:

- Editing Microsoft Office documents
- Upload / download file structure
- Uploading files to the system using e-mail
- DevOps services
- Protection of personal information (POPI)
- Printing / faxing documents
- Implementing pass-through authentication (Single Sign-On)

### INPUT

- EHR print outs
- Lab faxes
- Case report forms
- E-mail
- Regulatory forms
- Scan

### EDIT

- Collect
- Redact
- Review
- Sign
- Send
- Remote monitoring
- Remote close out
- Site analytics

### DISTRIBUTE








## Solution

This is what we introduced to reach the client's goals:

- Integration with Microsoft Office 365 (WOPI)
- POPI: converting documents (including both pdf and rtf) to a tuple of images
- POPI: option for editing images and downloading them as a pdf file
- Upload / download file structure module (node fs, zlib)
- Integration with the Scrypt Sfax service (for printing / faxing)
- Integration with MailGun (uploading files via e-mail)
- End user authentication (SAML)
- Continuous integration: Jenkins + Docker + AWS infrastructure

## Key features

As a result, the application now has a number of advantages:

-  Automated workflow for an institution's documents (editing, uploading & downloading, printing & faxing and implementing pass through authentication (Single Sign On))
-  Personal information protection
-  Possibility of converting files to a tuple of images
-  Possibility of editing images and downloading them as a pdf file
-  Integration with Microsoft Office 365, the Scrypt Sfax service and MailGun

## Technologies

### Server Side:

- Javascript
- Node.js
- Hapi
- Mongoose

### DB:

- mongodb

### Front Side:

- Javascript
- Angular.js 1.5
- HTML
- CSS

### Other:

- Microsoft Office 365 (WOPI)
- Scrypt Sfax
- MailGun
- SAML
- Jenkins
- AWS
- Docker
- PM2