**USPSScanService** - The data file we get from the postoffice known as PKG file, which has references between our system and the postoffice and the corresponding delivery status of packages. We process those files through USPSScanService and update the status in our existing MongoDB records (Coming from print process data file). It also aggregates records and makes the summery for InfoTrac Jobstatus Module UI. Besides this ETL process, we have some REST endpoint to control and retrieve the data from the database.

**Language**: J2EE 8

**Framework**: Spring Boot 1.5

**Database**: MongoDB 4.2 and MSSQL

**Platform**: Docker

**InfoREACH** - Multi-tenant application to create offers, create targeting rules, creating campaigns and configure user behaviour based offer workflow, audience report, campaign report, offer report, reach offers with customized content to targeted audience through digital and print channels, landing page workflow creation to send out with offers, creating user engagement forms to incorporate with landing pages, A/B testing and many more. This app is basically the face of project infoREACH. Currently it supports one way sso from infoTRAC to infoREACH. The webapp works with collection of microservices

**Language**: JS

**Framework**: MeteorJS

**Database**: MongoDB 4.2, LDAP

**Platform**: Docker

**InfoPUB** - While InfoReach serves the fi admin/users, InfoPUB is for the end users to view the landing pages flow, fill out user engagement forms, manage their subscription for each channel and roll out the web/mobile ads to the targeted end users.

**Language**: JS

**Framework**: MeteorJS

**Database**: MongoDB 4.2

**Platform**: Docker

**TargetingEngine** - Service to support applications to create eligible audiences based on provided criteria which are dynamically generated. Application calls targetingengine api with a specific set of rules. Based on apptype the service chooses it’s datasource of audiences and executes either of following

- Creating audience collection based on provided formulas in InfoSENSE which will be a

summary version of all data fetched and stored through InfoSENSE.

- Creating eligibleAudience collection based on the rules created from applications (e.g. InfoREACH) for a particular campaign and the user behaviour specific offer workflow.

**Language**: J2EE 8

**Framework**: Spring Boot

**Database**: MongoDB 4.2

**Platform**: Docker

**ClickTrac** - A simple microservice that creates a tinyURL for a redirectURL with provided metadata. Whenever that tinyurl is called clicktracker can store the event with browser and user info. Later clicktracker can generate analytics for the events of an user or for all the users of a particular client.

**Language**: NodeJS

**Framework**: ExpressJS

**Database**: MongoDB 4.2

**Platform**: Docker

**InforeachService** - This microservice manages and controls the process of email/sms campaign launch, all the inforeach specific backend processes and sso. It also communicates with other services on a single platform to reduce dependency from other services/applications of this boilerplate.

**Language**: NodeJS

**Framework**: HapiJS

**Database**: MongoDB 4.2

**Platform**: Docker

**DirectMailService** - The printing process is already well developed and in place for statement printing. DirectMailService is a microservice developed to manage, launch and track print marketing campaigns using InfoSync. InfoREACH calls DirectMailService to generate images from contents provided by the client. DirectmailService checks the quality of used images and enhances them if required. DirectMailService manages the types of print, paper quality, quantity and cost management for marketing pieces.

**Language**: NodeJS

**Framework**: HapiJS

**Database**: MongoDB 4.2

**Platform**: Docker

**Print Process** - is responsible for processing customer data that comes in different formats and generating financial statements, transactional and other data driven documents. With the help of customized tool named Papyrus Designer of ISIS company in Germany we build those statements in such format (afp) that helps automate the printing process and mailing statements to customer, cost calculations, pass afp files(as those are lightweight) and index file to estatement team so that they can process those for client(Bank/CU) to be able to see their statements in online, provide archival services in zip, cd, searchable dvd format for the institution to help them searching , keeping track of their customer statements . Also generate different types of reports like invalid address , invalid account (based on different criteria), any other particular reports demanded by the client. Also, provide index files and afp files to the product team so that they can process those to help tracking of mailing and posting.

**Language**: Perl, Python, Shell

**Framework**: Papyrus

**Database**: Oracle

**Platform**: Linux

**InfoSync** - API Service that provides an interface to track document processing status, diverting some statements from printing, online proofing, pdf report generation. Also provides apis to automatically trigger some shell process.

**Language**: Python

**Framework**: Flask, SQLAlchemy

**Database**: Oracle, MSSQL

**Platform**: Docker

**INFOTRAC**: is a multi-tenant application that tracks and manages other applications and services behaviors by the adminsitrative personnel of the banks and credit unions. It has many modules to support various business processes such as URF, Job Status, Account Management, Message Manager, Enotification, Marketing and so on.

**Language**: Java

**Database**: Oracle  
**Framework**: Spring 3

**Platform**: Tomcat into linux box

**URF**: URF is the core module to manage authentication and authorization for the entire application.

**Job Status:** This module provides insight into various print and electronic jobs processed by several scripts and programs.

**Account Management**: Using this module admin of the banks/credit union can manage and view account details.

**UI Framework**: View JS

**Message Manager**: By using this admin can set marketing messages which later on get embedded with the electronic and print statements based on several business criterias.

**Enotification & Marketing**: This module helps users to create marketing and notification templates for several business and marketing use cases.

**Contextual Search**: This micro service has several REST endpoints, which provide features to process bulk documents and store into elasticsearch database, search into documents by keywords.

**Language**: Java

**Framework**: Spring

**Database**: Elasticsearch

**Platform**: Docker

**InfoPDF Service**: This is a REST microservice consumed by ContextualSearch which can highlight any keyword or sentence in a PDF file.

**Language**: Python

**Framework**: Flask

**Platform**: Docker

**Processing UI:** Processing application that processes different data files or index files to insert data into the oracle database. It also produces different kinds of report for the client using direct query or API.

**Language:** Java

**Framework:** J2EE

**Database:** Oracle

**Platform:** Tomcat, Linux

**Prism:** Front end application for end users (Both Bank/CU Admins and Customers/Members ). Separate application for each Bank or Credit Union