Scope was 2 points

What is kobo, technologies used

Challenges faced with kobo, seeking help from hitender

Kobo installed on local machine but proxy restrictions in ntl, open network needed

What is odk, technologies used

Component diagram

What we have understood

What we have been able to do

Components for grid view and workflow?

Focused Scope of the PoC::

Kobo Toolbox extension for reusable grid view

ODK-X extension for complex workflows

Technology stack used in KoBo Toolbox

HTML, CSS, Javascript, Coffee Script, Docker, Python, MongoDB, PostgresDB, Redis, Nginx

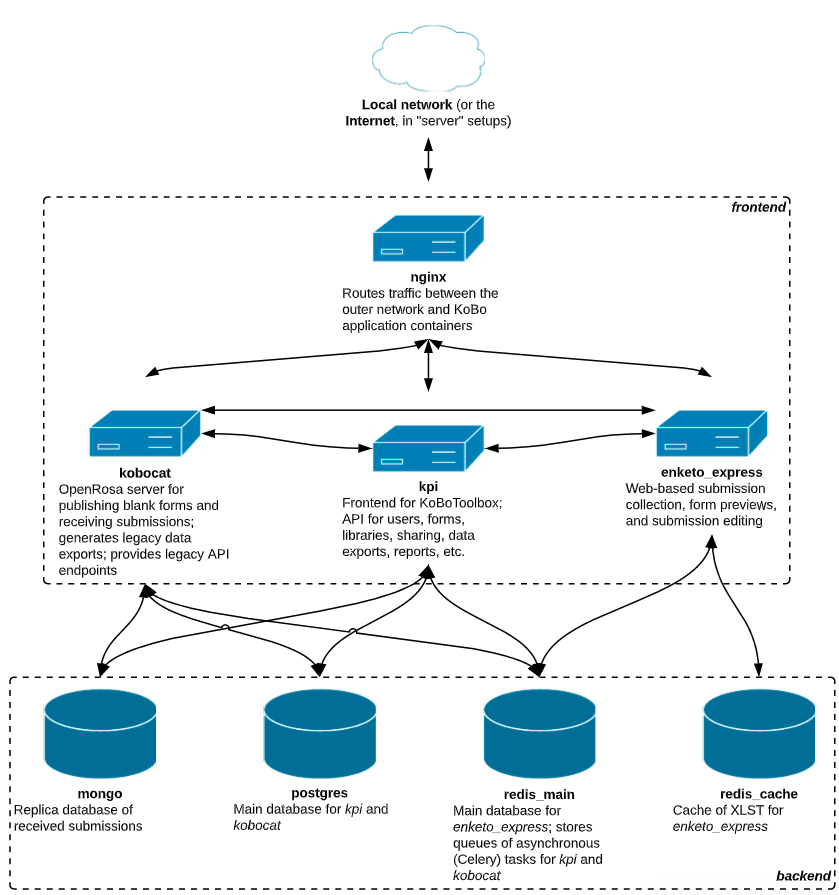
Components of KoBo Toolbox:

kpi - for creating survey forms and reusing assets through a question library

kobocat and kobocat-templates - for deploying surveys, collecting, and analyzing data

enketo-express - HTML5 Web app for collecting data, previewing forms, editing data submissions

kobocollect - Android app for collecting data



Progress so far with KoBo Toolbox

Kobo Toolbox could be successfully deployed on a local virtual machine outside NTL. This deployment was based on Ubuntu 16 as the base server. All the components could be studied.

Immitating the same deployment on RHEL and CentOS servers within NTL is proving difficult. With the former the case was that of enterprise dependencies like Docker Enterprise Edition which were not available, whereas with the latter the problem being faced is due to network limitations. The deployment of KoBo Toolbox requires to communicate with several URLs over the internet which are blocked through NTL proxy. As was requested the installation was attempted on RHEL/CentOS servers, but KoBo documentation states that Ubuntu servers are preferred.

Path ahead with KoBo:

Installation of KoBo can be achieved by provisioning an Ubuntu 16+ server with complete open internet access for the purpose of this PoC. The code level changes for feature extension can then be carried out.

Technology stack of ODK-X:

Components of ODKX:

* ODK-X Application Designer: data collection form creation
* ODK-X Survey: data collection
* ODK-X Services: data sync and database access
* ODK-X Cloud Endpoints: data and application files cloud server
* ODK-X Tables: data display and visualization
* ODK-X Services: data sync and database access
* ODK-X Cloud Endpoints: data and application files cloud server
* ODK Aggregate: stores, analyzes, and presents [XForm](https://docs.opendatakit.org/form-design-intro/) survey data

What we have achieved so far:

Application Designer has been installed on our desktop computers. This is the component that will have to be edited for customized features.

ODK Services and Survey are client side applications to be deployed on android devices. The client-side android platform for those components has been set up. ODK services has been installed.

ODK Aggregate has been set up. Connection between ODK Services and ODK Aggregate has been setup.

Bottleneck faced:

Unlike KoBo Toolbox, we haven’t been able to locate a dockerized installation process for ODKX, owing to which each individual component has to be installed separately

Pawan

Anant, manish, harish, vaibhav, hitender

Hello Pawan,

Please find below the progress so far with the PoC

Initial scope of 2 weeks PoC -

1. **Find how Kobo can be extended to support grid view**.  If we find kobo good enough, then whatever work we do in PoC to extend to provide grid view is reusable and is not throwable.
2. **Find how grid view and WF works in ODK-X**.

|  |  |  |
| --- | --- | --- |
|  | KoBo Toolbox | ODKX |
| Our understanding so far | * Dockerised deployment * Important components - kpi for creating forms, kobocat for data analysis, enketo express for editing submissions, kobocollect for data collection |  |
| Technology used | HTML, CSS, Javascript, Coffee Script, Docker, Python, MongoDB, PostgresDB, Redis, Nginx |  |
| Progress so far | * For the purpose of learning deployed successfully on an Ubuntu machine outside NTL. * Identified Kpi as the component to be modified for grid-view feature addition | * ODK Aggregate installed * Application Designer installed and identified as the component to be modified for grid-view feature addition * Client-side android platform set up. * ODK-X Services installed on client side and connected to Aggregate server. |
| Challenges faced | Installation requires connection to several external websites which are inaccessible from NTL proxies this hindering deployment. All such URLs are not documented and hence being discovered incrementally as installation progresses. Installation on RHEL server requires all the dependencies to be enterprise grade. As the preferred OS for KoBo Toolbox is Ubuntu, an Ubuntu 16+ server with open internet access for the purpose of this PoC could do the trick. | Unlike KoBo Toolbox, the deployment isn't dockerised owing to which each component has to be deployed individually which is time consuming. |

What we will do in the ensuing week

Apart from this initial scope, there few additional items that project team want us to complete as part of POC as shown below. These additional items we need to assess and identify additional poc effort.

If we find ODK-X good, then we should use one of the complex workflows and implement the same on ODK-X

* Ability to monitor response collection (e.g. see how many responses received on which version of toolkit, when and by whom)
* Auto calculate data field value based on answers (e.g. totals)
* Ability to specify the list of selections for multi-select (large list either through a local list or master data reference)
* Cascading multi selects (e.g. Countryà Stateà District)
* Define metadata for the toolkit (apart from classification or tagging, e.g. Ref no., Description, Collection type etc.)
* Ability to search a toolkit from central repository and update and reuse it as a new toolkit
* A Survey can be trigged within an Audit Lifecycle (Case) or as a Stand-alone Activity.
* A designer creates a Survey with bunch of Questions and Answers of type various datatypes (Number, String, **Grid (Row-repeat)**, Datetime, Multiselect, Map, Slider etc.).
* Field Validations.
* Surveys can contain Skip logic and Branching logic.
* Created Survey template undergoes an approval cycle.
* Once approved, these Surveys are published with versioning.
* Surveys can be instantiated in Offline mode.
* Survey responses undergoes an approval cycle (Accept Response/Discard Response/Send for Refill).
* Preparer, Reviewer, Approver, Respondent are all determined as per the MDM (Master Data Management) in the application.
* Surveys are sent to external users with a link over an email.
* Consolidation of Surveys outcomes to derive insights.
* Multi-lingual Support and Geo-tagging.
* Surveys will be used in Laptops and Mobile (Online/Offline).
* BAM Reporting / Real-time Dashboard.
* Ability view Survey responses by Version.
* Classification of Surveys: Hard Classification (Explicit mention of Categories), Soft Classification (Tagging). This will be used for Search/Look-up

If open network not possible then list of url will be found incrementally