|  |
| --- |
|  |
| Virtual Community |
| Methodology for Custom Wiki-base Instantiation |
|  |



By :

Aninda Maulik

Mohamadnour Badr

Morad Benkaraache

Dharmalingam Dhayananth

# Acknowledgement

We would like to thank Professor, Pierre Marret who guided throughout in order to complete this project. We are also grateful to each and every team member who worked really hard to achieve our milestone.

Aninda Maulik

Mohamadnour Badr

Morad Benkaraache

Dharmalingam Dhayananth

**Abstract**

This Project is provides a methodology for any community who wants to create custom **Wikibase** for their documentation. This document provides detailed information on each step and every method which a community should follow in order to achieve their goal. Each step has been deep driven with information about supportive methods, tools, and possible alternatives.

Wikibase is a free RDF data source that can be edited, published and used by both human and automated bots. This data source approach has been followed in Semantic web technology. Already, many organizations and governments have been adopted this technology for their documenting purpose. Ex (France : [Les Donnees de la BnF en RDF](https://www.data.gouv.fr/en/datasets/data-bnf-fr-les-donnees-de-la-bnf-en-rdf/) , United Kingdom ([article](https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/news/datagovuk-uses-rdf-data))).

(Wikidata, 2020)

We will be creating a custom Methodology that helps to create a custom Wikibase by a community and we will demonstrate it with an example. We have chosen “Nature Disaster” as our topic for this demonstration purpose. We will give detailed information about each step of our methodology with our demonstrating example.

Table of Contents

[Acknowledgement 1](#_Toc35221678)

[Overview of the Methodology 5](#_Toc35221679)

[Task 1: Organize and Structure the Domain 8](#_Toc35221680)

[Ontology Diagram 9](#_Toc35221681)

[Deliverables of the task 10](#_Toc35221682)

[Events 10](#_Toc35221683)

[Task 2: Resource management 10](#_Toc35221684)

[Tools 11](#_Toc35221685)

[Deliverables of the task 11](#_Toc35221686)

[Events 12](#_Toc35221687)

[Task 3: Identifying Data Sources 12](#_Toc35221688)

[Inserting Data to the Wikidata 13](#_Toc35221689)

[Steps for create a new Data in Wikidata 13](#_Toc35221690)

[Direct data import method 16](#_Toc35221691)

[Deliverables of the task 16](#_Toc35221692)

[Events 16](#_Toc35221693)

[Task 4: 17](#_Toc35221694)

[Deliverables of the task 17](#_Toc35221695)

[Events 17](#_Toc35221696)

[Task 5, 6: 18](#_Toc35221697)

[Deliverables of the task 18](#_Toc35221698)

[Events 18](#_Toc35221699)

[Task 7: Identify data extraction method and tools 19](#_Toc35221700)

[https://www.mediawiki.org/wiki/Extension:External\_Data # Storing data 19](#_Toc35221701)

[Deliverables of the task 20](#_Toc35221702)

[Events 20](#_Toc35221703)

[Task 8: Plan Collaboration Platform 21](#_Toc35221704)

[Some of the examples 21](#_Toc35221705)

[Deliverables of the task 22](#_Toc35221706)

[Events 22](#_Toc35221707)

[Task 9: Wikibase Instantiation 23](#_Toc35221708)

[Docker Insanitation 24](#_Toc35221709)

[System df 25](#_Toc35221710)

[Example Image 25](#_Toc35221711)

[Getting the machine images running 25](#_Toc35221712)

[Digital Ocean 26](#_Toc35221713)

[Steps 26](#_Toc35221714)

[Review of the methodology 28](#_Toc35221715)

[Task completed 29](#_Toc35221716)

[Task to achieve 29](#_Toc35221717)

[References 30](#_Toc35221718)

# Overview of the Methodology

Figure : Methodology-Segment-1

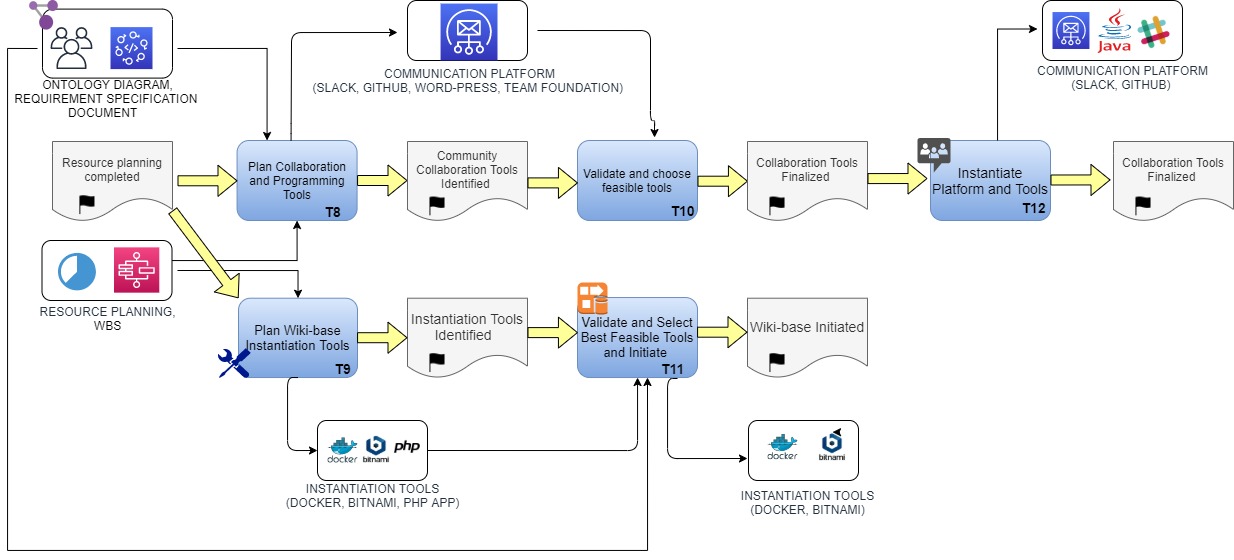
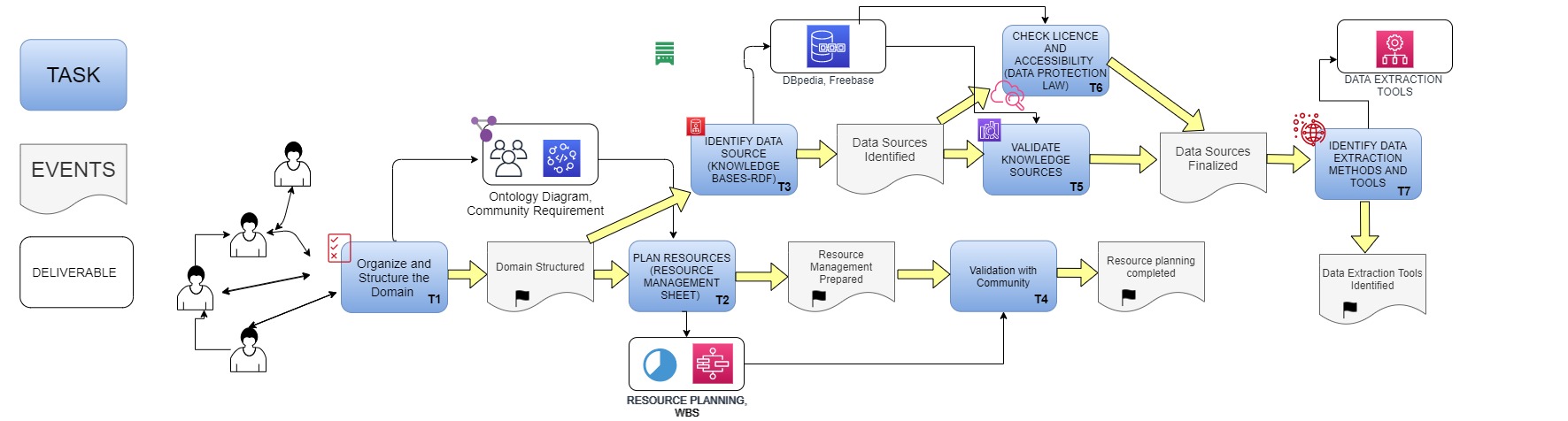


Figure : Methodology- Segment -2

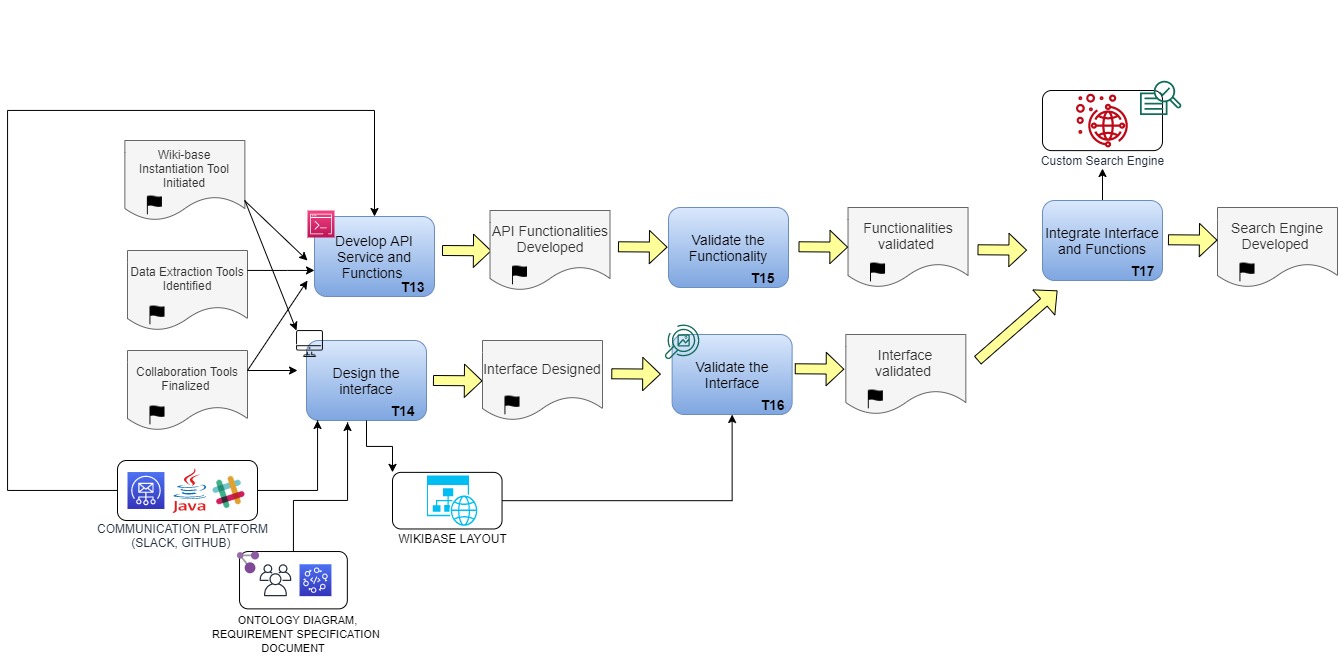


Figure : Methodology- Segment-3

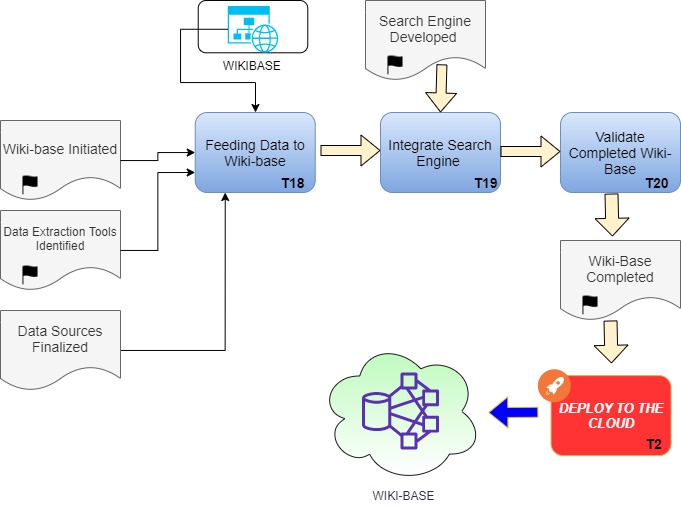


Figure : Methodology - Segment-4

The above images shows each step of our custom methodology. Each task would require some “Deliverables” (input) and produces a deliverable (output). Similarly, a task would start after the occurrence of one or more “Event” and produces an event which would initiate another task. This shows that one task is dependent or independent of on one or more previous task and events.

Following figures are been used to indicate “Task”, “Event” and “Deliverables”.



Figure : Symbols used in the diagram

# Task 1: Organize and Structure the Domain

This is the initial step that community has to follow in order to create their custom Wikibase instance. Identifying the requirement and structure of the chosen domain are key point in creation of Wikibase. This task can be achieved by following below steps.

* Identify the domain
  + Research on domain
  + Consult expert for the gain better understanding and knowledge of the domain (Ex: technical knowledge)
* Identify the entitles
  + Entities of the chosen domain.
* Identify the properties
  + Properties of each entities.
* Represent relations between entities and properties

The community can be able to design their ontology diagram by following above step.

## Ontology Diagram

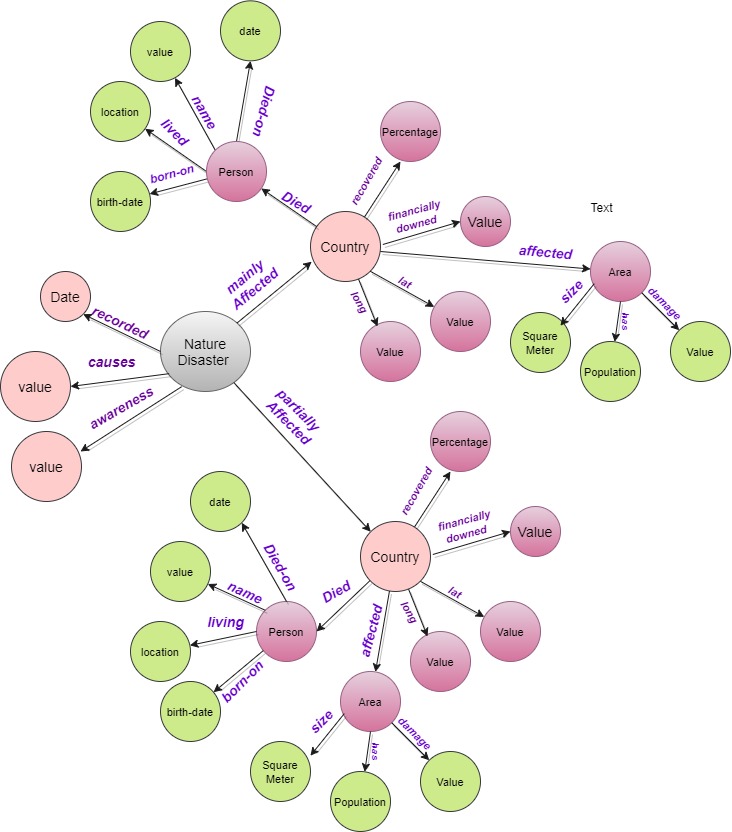


Figure : Ontology diagram

Ontology diagram is a graphical representation of selected knowledge base. This diagram helps to understand the entities, properties and the relationship between them. In large knowledge bases, this can extremely grove and gives clear view of the complicated domain. So the community team can understand their domain properties and entities, also the relation.

## Deliverables of the task

* *Requirement Sheet*
* *Ontology Report, Diagram*
* *Domain Knowledge Sources (Articles, Expert Advices etc)*

## Events

* Required Events to Start the Task
  + *None*
* Events Produced
  + *Domain Structured*

# Task 2: Resource management

Resource management task helps to organize and coordinate the community workload in appropriate manner. Community members can able to identify their available resources, necessary resources, and time duration of the project. Thus, they can manage and organize their workflow accordingly. Therefore, everyone can work to achieve community requirement in an organized manner. Successful resource management will help to choose appropriate organization structure and task allocation within the community member.

This task can be performed by some specially trained member in the team from Human resources management. It can be done by one more group of members. They should identify the complete workflow and task of the project and should identify necessary resources. Also, it is better to analyse the difficulty and risk of each task and prepare for possible risk management document.

Following steps will helps to achieve successful Resource management task.

* Identify the required resource for the project
* Define the community organization structure
* Define duties and roles of each member
* Estimate the duration
* Deciding meeting dates, discussion point and dates.

Output of this task can be a “Resource management sheet”, Work Breakdown Structure (WBS) and Burndown chart.

This task can be assisted with the help of computerized software application. Example: [Gitlab](http://www.gitlab.com), [Github](http://www.github.com), [Microsoft Office Project Management](https://products.office.com/en-us/project/project-management-software), Excel and [Jira](https://www.atlassian.com/software/jira).

* For our demonstration purpose, we have chosen Flat organization structure. Thus, the workload will be divided equally within the community members.

**Project management tool**

**S**ome project management tool will help to manage, coordinate and organize the community project in more formalized manner. Community people can check the task and duration for each and every one and can also monitor the progress of each member constantly. Also, it helps to assess the risk, difficulty, and importance of task, contribution level and many more. Thus the community is benefited by organizing their project work with a computerized project management software application.

## Tools

Atlasian, Github, Visual Studio Team Foundation, Gitlab

## Deliverables of the task

* Required
  + *Domain Requirent Specifications*
  + *Ontology Diagram*
* Produces
  + *Resource management sheet*
  + *Tasks and allocated resources*
  + *Work Breakdown Structure*
  + *Gantt chart*

## Events

* Required Events to Start the Task
  + *Domain Structured*
* Events Produced
  + *Resource Management Prepared*

# Task 3: Identifying Data Sources

This task helps community to select appropriate data sources containing RDF data from the internet which is publically available for reuse purpose. Community should follow below steps to identify the required and relevant data sources for their purpose.

**Steps :**

* Identify available data sources (RDF – Data sources)
* Wikipedia, Wikidata, WikiMedia, DBPedia (<https://wiki.dbpedia.org/>), Amazon Neptune
* Check data availability
* Select Data sources by considering community requirement and domain
* Validate the data sources with community members
* Identify required data set from those selected data source and check data accessible privilege (Data protection act, Patent rights)

# Inserting Data to the Wikidata

Wiki base can be a collection of many data sources. Wiki base community can work together to import data into their wiki base from different bases. Some of the data sources can be Wikipedia, Wikidata, WikiMedia, DBPedia (<https://wiki.dbpedia.org/>). Data extraction requires some sort of pre-requist skills such as experience with wiki page creation and editing, spreadsheet data moving experience and duplicating experience.

## Steps for creating a new Data in Wikidata

**Step 1:** Choose data to import

Should choose most reliable data which also required to be accessed publicly. (Preferably online)

**Step 2**: Start a data import

Community should follow the given link ([Wikidata:Dataset Imports](https://www.wikidata.org/wiki/Wikidata:Dataset_Imports)) and should create a new import process to initiate the data import process in to wiki data.

**Step 3**: Import the data into a spreadsheet.

The Wiki data community suggesting to utilize online spread application for this purpose (Ex: [GAFAM](https://en.wikipedia.org/wiki/GAFAM), [Framacalc](https://accueil.framacalc.org/en/)).

**Step 4**: Define the structure of the data

Defining the structure of the data is most important part in the process of data import. Structure of the data should be created strictly following the constraints and this should be followed by Wiki data community.

**Step 5**: Define the structure of the data

Defining the structure of the data is most important part in the process of data import. Structure of the data should be created strictly following the constraints and this should be followed by Wiki data community.

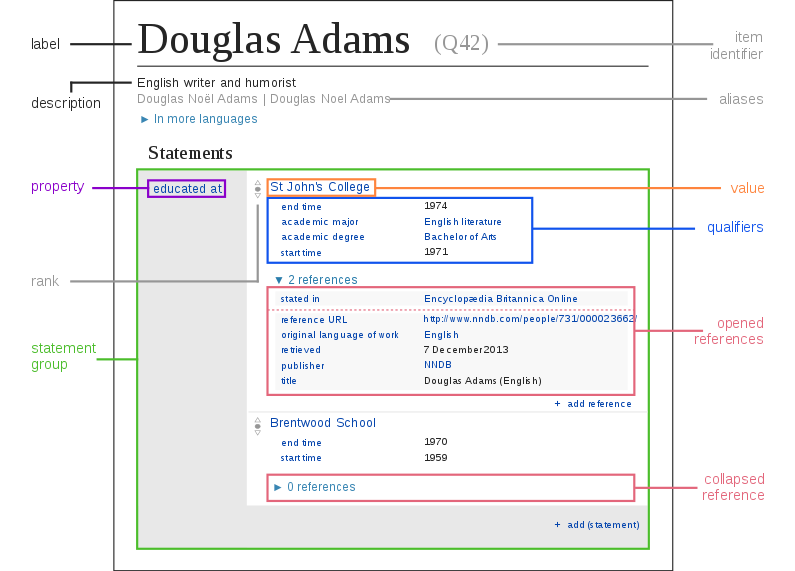


Figure : Data insertion – Wikidata

(Wikidata, 2020)

**Some important Properties of structuring data**

* **Alias:** Alternative name for an item
* **Constraint:** is a rule which specifies how a property should be used. (Ex: Most identifier must contain only one value- Single value constraint).
* **Data Type:** This is an attribute of a property which specifies the data type of it. Usually it cannot be modified in future without certain permission,
* **Entity:** This is the content of the Wiki data page. It can be an item or a property. It has a unique identifier named entity ID. This id can be used to access this entity using a Unified Resource Identifier as follow, Ex:( http://www.wikidata.org/entity/*ID* where ID is entity) ID
* **Metadata:** Metadata about entity. It also contains similar properties of entity. It can be used to validate the entity using EntitySchema tools.
* **Label:** This is the main text to identify the entity. Every entity contains at least one label which is human-understandable. This might not be unique.
* **Language Attributes:** Language specific labels
* **Property:** this is the descriptor for a data value.

**Step 6: Format the data**

Original data should be copied within the spreadsheet itself and should be named as “Structure for wiki data”.

The spread sheet must contain the following columns and values

Table : Wikidata insertion constraint

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Unique ID | Name / Title | Description for Wikidata | Description for importing data | URL | More data 1 | More data 2 |

**Step 7:** Choose how to import the data

Importing process can be automated using Wikidata bot request or can be manually imported using self-import method.

**Step 8:** Match the data to Wiki data

**Step 9:** Add the data to Wikidata

**Step 10:** Check and summarize the imported data.

**Bot request:**

* Bots can be created to extract data from different data sources to the wiki data.
* There exists some bot policies and constrains. Wikibase community is supposed to test the legitimacy of the bot and then the tasks of the bot can be accepted or rejected.
* The bots can be also created to create a new entity, alter and delete the entity when it identified that it was incorrectly created by bot.
  + PetScan- for creating items from Wikimedia pages and/or adding same statements to items
  + QuickStatements- for creating items and/or adding different statements to items
  + OpenRefine - to import any type of data from tabular sources

## Direct data import method

* Users can able to import dataset from external sources into wikidata.
* The required data for a wikibase can be important using existing data importing tool in wikidata by specifying external link, name and the description of the dataset.
* The imported data can be used by other people too.
* The documenting and import process page enables platform to the community to collaborate and provides a record for updating, correcting errors and keeping track of updated datasets.

## Deliverables of the task

* Produces
  + *Online Data Sources*
  + *Ex: DBPedia, Wikidata, Wiki Pedia, Freebase*

## Events

* Required Events to Start the Task
  + *Domain Structured*
* Events Produced
  + *Data Sources Identified*

Task 4: Validation with Community – Resource Planning

This task describes validation of resources planning with community members. Each member should agree with the prepared resource management documents and allocated tasks. Also, they can discuss and make modification if necessary. This helps to produce a finalized version of renouncement deliverables. Validation task can be carried with with a project discussion or meeting.

## Deliverables of the task

* Produces
  + *Validated Resource planning.*

## Events

* Required Events to Start the Task
  + *Resource management prepared*
* Events Produced
  + *Resource planning completed*

Task 5, 6: Validate the knowledge resources and Data Accessibility Check

This task describes validation of knowledge resources which has been identified in previous task. Community members should analyse the check selected online data sources against community requirement. Members should search selected data sources and should check for data availability for their specific domain. Based on the data availability they can able to confirm the data sources. After, they then can finalize the selection of existing data sources. Data accessibility should be analysed simultaneously while doing validation Task. Because, not all the data from internet is legal to access. Some data cannot be reused and they are protected with some legal law. Therefore, community members should assure that, they can reuse the data from selected data sources.

## Deliverables of the task

* Required
  + *Identified data sources.*

## Events

* Required Events to Start the Task
  + *Data sources identified.*
* Events Produced
  + *Data sources finalized*

# Task 7: **Identify data extraction method and tools**

## <https://www.mediawiki.org/wiki/Extension:External_Data> # Storing data

As it mentioned in Identifying data extractions methods is required to find out better solutions for extract data from different sources to Custom Wikibase. There can be many different approaches and method found for this task. Selecting appropriate tools for the selected community is an essential task. Some tools may have different features from others. Not all the tools are applicable to single project. Also, some tools required predefined skill set for the purpose of utilization. And some other tools needs to be purchased which is not feasible option for a small custom Wikibase projects. Because features like automated data extraction, time interval check function (This feature will check selected data sources in selected interval time period and extract all new entries of data to the custom Wikibase automatically) might not require for those small projects. However, large scale governmental and organization purpose Wikibase projects might need those tools for the better scalability.

(Chhabra1, Rashmi, 2014)

Community team should perform a research and should identify the existing tools and their capabilities and can prepare a document on this topic if needed. So, the other members can able to collaborate and discuss for necessary tools, features and the limitation of existing tools. Based on that collaboration, they can hire some experts to implement or develop new tools with required feature set in order to support community need.

Below points are listing all necessary steps that are required to perform this task and complete it successfully.

* Search data extraction methods and tools for selected RDF Data sources
  + Different RDF sources contains platform specific tools which are only applicable to selected RDF source. For an example
* Validate the tools and technique with other team member
* Finalize the method and tools
  + Direct Query (SpareQL), Exported RDF file, Exported TTL file, API Technique

Some of the existing tools and their features have been listed below in order to support to the community for the purpose of data extraction.

## Deliverables of the task

* Required
  + *Finalized data sources.*
* *Produces*
  + *Data extraction tools*

## Events

* Required Events to Start the Task
  + *Data sources finalized.*
* Events Produced
  + *Data extraction tools identified*

# Task 8: Plan Collaboration Platform

Planning and initiating collaboration tools are mandatory objective on every community based projects. That helps to collaborate, coordinate and organize their work in most sufficient manner. Accordingly, creating a custom Wikibase also requires this step. Community members should decide their communication, code collaboration and file sharing platforms.

This task can be achieved successfully by following below steps.

* Define communication standard
* Define sub community – if necessary
* Define communication and code collaboration tools
  + Google slides, Google doc, Git-hub, Slack, Skype, Team Viewer, Gitlab
* Check the community requirement and flexibility of the selected tool
* Initiate collaboration tools

## Some of the examples

* Communication tools :
  + Skype, Slack, Team Viewer, Whatsapp, Facebook group.
* Code collaboration and versioning tools (Version Control Tools)
  + Github, Gitlab, Bitbucket, Mercurial, SVN
* Remote Desktop Control
  + Team viewer, Join me, Remote Desktop (MS).

These tools should be initiated with the required privileges based on each role of the member in the community.

## Deliverables of the task

* Required
  + *Resource planning deliverables.*
  + *Domain structure deliverables.*
* *Produces*
  + *Collaboration tools and platform*

## Events

* Required Events to Start the Task
  + *Resource planning completed.*
* Events Produced
  + *Community collaboration tools identified.*

# Task 9: Wikibase Instantiation

This is most crucial task in this methodology. This task provides supportive steps and tools that a community can follow in order to instantiate Wikibase instance. There are many tools found in the internet in order to create an instance of the Wikibase. We have listed most famous and standard tools in this section.

* [Docker](https://www.docker.com/) – A self hosted container with millions of images. We can simply pull and run any image for free in this container.
* Manual installation ([Tutorial](https://www.mediawiki.org/wiki/Wikibase/Installation))- Wikimedia gives open source PHP application that can be downloaded directly to the local machine and can be executed or hosted in local server([XAMPP](https://www.apachefriends.org/index.html)) for the purpose of custom Wikibase instantiation. Then it can be hosted in the cloud or any internet hosted PHP server (Ex: Apache).
* [Tikiwiki](https://tiki.org/HomePage) – This tool provides complete platform with integration with designing tools where user can able to design and publish their custom RDF Data source without needing of deep programming skills. Note: This tool does not providing Wikibase instance.
* [Bitnami](https://bitnami.com/) – This tool also hosted with hundreds of images. Wikibase image also hosted in this web application and can be reused for any community purpose. It used micro-service approaches to bundle images with it’s all necessary resources and dependencies
* Digital Ocean – This tools helps to publish the Custom wikibase in the internet

We will be demonstrating Docker and Digital Ocean for our custom Wikibase instantiation and we will further described detailed step about installation, initialization, and customization of wikibase using Docker in the following pages.

## Docker Insanitation

**About Docker**

* Work as containers
* Docker file(5-30 line)
* Create images. (run in the container),it will include the node.js ,project code and instalment of programs.
* By the image name, we can run it in any machine.

**Initiation steps:**

1. Download and Install - Docker.
2. Download the docker-compose.yml file example (save it into your working directory) with the name docker-compose.yml).
3. Downloading images and starting the containers.

**Commans**

Table : Docker Commands

|  |  |
| --- | --- |
| Create new container from the images. | version(information about client ,server docker) |
| Start the container. | -v,- - version(version of docker). |
| Stop the container. | - - help , login . |
| Show me all the containers i have . | pull(make a copy of the image and put it in container) |
| Delete the container. | stats(details about the running containers like memory , input, output) |
| Create new container from the images. | System prune(remove all data of the stop containers). |
| Check disk usage image and containers | System df |

## Example Image

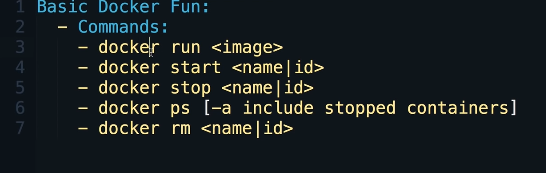


Figure : Docker command

## Getting the machine images running

Download the*docker-compose* file and place it on the computer where Docker Engine and docker-compose are installed.

**Commands**

Table : Docker Command 2

|  |  |
| --- | --- |
| pull the needed Docker machine images | docker-compose pull |
| Start the machine images in the background | docker-compose up -d |
| Verify that all the services have started | docker-compose ps. You should see eight images in an “Up” state |
| Check the logs for success or errors -- for example, | docker-compose logs --tail="20" -t |

## Digital Ocean

It is a unique cloud hosting provider that offers cloud computing services to business entities so that they can scale themselves by deploying DigitalOcean applications that run parallel across multiple cloud servers without compromising on performance!.

\*\*each client can have his own Droplet, running a chosen Operating system. The client has administrator (root) access to the VPS, so he can make changes without conforming with the other users, which have their Droplets on the same DigitalOcean server.

## Steps

* Create an account.
* Create droplet.(core os - specific for docker).
* Create ssh to log to the server .
* Generate public key by network or server.
* We will give the pk for the DO and it will generate the Sk for us .
* Log in the the server with username, password,SSH and
* Install the docker component from the github by clone it to the server.
* (<https://github.com/wmde/wikibase-docker>)
* This repo contains [Docker](https://www.docker.com/) images needed to setup a local Wikibase instance and a query service.
* There are many images on Docker that can help us in link

<https://hub.docker.com/u/wikibase>.

* There we will found a lot of programs services can we use it and help us.
* !!! Here you will need to install the Docker compose as a tool and execute it.
* !!! We will still use the basic Docker command.

# Review of the methodology

# Task completed

**First Version:**

* Organize and structure the domain
* Resource management
* Identifying Collaboration Platform
* Identifying Wiki base instantiation method
* Identify the data sources Design the interface

**Second Version:**

* Identify data extraction method and tools
* Creation of Search Engine
* Creation of Wiki-base application
* API Programming to extract data to the wiki base

# Task to achieve

* Inserting new RDF data
* Design the interface
* Testing the interface
* Lunching the custom wiki-base
* Documentation

# References