Implementation Guide

DIVOC

Contents

1. About this guide 4

2. A quick guide to DIVOC implementation 4

2.1. Planning and organizing 4

■ Program/Implementation Team Structure 4

■ Integration efforts 4

■ Equipment and internet 4

■ Roll-out strategy 4

2.2. Adapting DIVOC 6

■ Scope of system 6

■ Setting up DIVOC 6

■ Hosting 6

3. Conceptual Design Principles 7

3.1. All meta data can be added and modified through the user interface 7

3.2. Schema Customisation 7

4. Setting Up a New Database/Instance 8

4.1. Strategies for getting started 8

5. Deployment Strategies 8

6. DIVOC as Data Warehouse 8

6.1. Data warehouses and operational systems 8

6.2. Data storage approach 8

7. Integration concepts 8

7.1. Integration and interoperability 8

7.2. Objectives of integration 9

7.3. Vaccination information exchange 9

7.4. Different DIVOC integration scenarios 10

7.5. Implementation steps for successful data and system integration 10

7.6. Specific integration and interoperability use cases 12

8. Support 15

8.1. Home page 15

8.2. Development tracking 15

8.3. Source code: github.com 15

9. Users and user roles 16

9.1. Roles 16

10. Localization of DIVOC 16

11. DIVOC Tools Guide 16

11.1. Overview 16
11.2. Architecture 16
11.3. Installation 17
11.4. Troubleshooting guide 17
12. DIVOC Documentation Guide 18
12.1. DIVOC 2 Documentation System Overview 18
12.2. Getting started with GitHub 18
12.3. Committing your changes back to GitHub 18

Version Control

Version	**Date of Revision**	**Description for Change**	**Author**	**Reviewer**	**Approved By**
1.0	Feb 15, 2021	Implementation Guide	DIVOC Program Management Unit	Sahaj and eGov Foundation	Dr. Pramod Verma

1. About this guide

The objective of this document is to provide a set of guidelines for DIVOC implementers. The document provides overview and directions on planning, configurations, deployment, rolling out and supporting DIVOC for country vide vaccination drives.

This document is intended to be used by those leading the planning, deployment and rollout process from its early phases, and should be shared with those involved in the process.

2. A quick guide to DIVOC implementation

Any implementation of Digital Infrastructure for Vaccination Open Certification (DIVOC) is aimed for establishing a robust, interoperable, scalable and sustainable Vaccination & certification system, which is flexible to changing needs of the Vaccine ecosystem of any country.

It is also important to acknowledge that with the current scenario there is a need to establish structured processes for best practice sharing, and innovation. This quick guide will provide an overview of the different facets of DIVOC implementation.

2.1. Planning and organizing

■ Program/Implementation Team Structure

A DIVOC core country team will be needed to administer a national level DIVOC system. Their responsibilities and required skills should be clearly defined. The team should include combination of following skills:

- Program coordinators - With understands the countries existing ecosystem (Policies, existing Health systems, Vaccine/Drug supply chains, Registries of Patients, Health providers and Users)
- Platform Co-ordinators – Support smooth rollout of platform. Central team, provides trainings(master), end user support, stakeholder onboardings
- Technology Experts - Capacity planning, Identifying local data protection requirements and retention. Country specific adaptors Implementation, testing deployment.

■ Integration efforts

The DIVOC platform is built on the concepts of “Plug & Play” and “Single source of truth”. DIVOC components can work in various combinations, based on countries' digital ecosystem and end user needs. Hence, throughout the DIVOC implementation, simultaneous efforts on information system integration and data exchange are required to be conducted.

■ Equipment and internet

A coverage specific assessment will be conducted to establish the requirement of hardware at both central and facility levels. Different hardware to enable users to access the platform like Desktops, laptops, tablets, mobile phones or a mix of these need to be supported.

Server and hosting alternatives need to be critically examined with regards to user and data load, capacity, infrastructural constraints, legal framework, security and confidentiality issues. Internet connection for all users will be needed. Mobile internet should be adequate for the majority of users doing data collection and regular analysis. Options for mobile phone users, bulk sms deals etc, should be examined if appropriate.

■ Roll-out strategy

For Successful rollout of DIVOC, both policy and existing IT systems aspects need to be analysed. The availability and features of existing IT systems will decide the roll-out strategy of Platform.

Some of the Key IT systems which will play key role in deciding DIVOC implementation strategy are as follows:

1. Country wide Vaccine Supply Chain System with last mile connectivity
2. Availability of Learning Management System (LMS) for Vaccinator
3. Availability of Drug Data Bank / Vaccine Registry
4. Availability of e-Payment System
5. Availability of Digital Citizen Registry (EHR/ PHR/ HIS / National ID / National/Regional disease surveillance registries)
6. Availability of Health Platform (Facility/ Vaccinator Registry)
7. Availability of Digital Signature
8. Availability of Health insurance (Public/Private) systems
9. Availability of Digital Document Vault (National / Regional)
10. Application Access for Facility Staff/Admin
    a. Mobile No./ Email availability **OR** Mobile No./ Email Id linked to National Id/ Employee Id
    b. National Id/ Employee Id and Mobile & Email are not available for Staff/Admin
11. Application Access for Recipient
    a. Mobile/ Email availability **OR** Mobile No./ Email Id linked to selected Recipient registry
    b. No Mobile No./Email available with Recipient

The detailed strategy to address each of the above mentioned IT systems availability/non-availability for successful rollout of DIVOC is mentioned in section "7.6 Specific Integration and interoperability use cases"

In addition to IT system aspects, the following operational aspects are also key determinants for formalizing rollout strategy. Implementers should work with relevant partners/stakeholders to ensure all the operational aspects are addressed by their relevant owners as per country guidelines.

1. Readiness on vaccination strategy
    a. Identification, approval and procurement of vaccines
    b. Distribution mechanism
    c. Logistics management
    d. Coverage under Government Schemes
    e. Last mile administration and monitoring

2. Capacity building
    a. Identification of vaccinators
    b. Training on vaccine administration
    c. Key & end user trainings for various roles identified by the country (example-Admin, registration, verification etc)

3. Infrastructure
    a. Availability of vaccine centers
    b. Availability of digital infrastructure to facilitate large scale vaccination roll out

4. Government protocols
    a. Criteria for beneficiaries
    b. Beneficiary identification, listing & enrolment processes
    c. Vaccine administration

d. AEFI and post-event monitoring protocols

The DIVOC core country team will play a key role here and each member should have clear responsibilities for the roll-out covering user support, user training, liaison with the health programs, etc. Broader support structures also need to be established to provide support, supervision, and communication with global/regional network of expert users and developers

2.2. Adapting DIVOC

■ Scope of system

- Based on the decision the system should support (system scope); customization and adoption of the DIVOC platform for one, multiple or all modules available in DIVOC like Orchestration, Vaccination, Feedback and Analytics module. The adoption activities identified by the country will need special competence and should be led by the core country team of DIVOC.
- Assessment of the intended users and beneficiaries is needed, such as related to their information needs, hardware and network needs etc.
- An understanding of the larger architecture of the Vaccination Drives are important; what other systems are there, and how they should interact with DIVOC to enable interoperability between electronic systems.
- If there are needs that are not currently supported by DIVOC, an assessment of additional software development is necessary. These can be addressed locally by developing an extension to the overall core platform.

■ Setting up DIVOC

- DIVOC can be set up in a containerized environment or bare metal servers.
- DIVOC has been tested for performance on kubernetes and deployment basic configurations are available in the repo
- Implementer should check on DataCenter / cloud specific loadbalancer and firewall configurations
- For running on bare metal servers, implementer should configure service scripts relevant to the operating system
- Infrastructure dependencies: https://github.com/egovernments/DIVOC/tree/main/infra
  - Postgresql
  - Kafka
  - Redis
  - Clickhouse
- Install application modules
  - https://github.com/egovernments/DIVOC/tree/main/infra/kubernetes

■ Hosting

- Choose appropriate hosting based on requirements (country specific)

- Cloud
- On Premise Cloud
- On Premise Bare Metal servers

- With any of these hosting options Kubernetes can be setup using

https://kubernetes.io/docs/setup/production-environment/tools/kops/

3. Conceptual Design Principles

This chapter provides an introduction to some of the key conceptual design principles behind the DIVOC platform. Understanding and being aware of these principles will help the implementer to make better use of the software when customizing a local database.

The following conceptual design principles will be presented in this section:

- All meta data can be added and modified through the user interface
- A flexible data model supports different data sources to be integrated in one single data repository
- Data Input != Data Output

In the following section each principle is described in more detail.

3.1. All meta data can be added and modified through the user interface

The DIVOC application comes with a set of generic open source tools for registries (opensaber), access and identify managements (keyclock), but the contents of the database, e.g. what data to collect, where will the data comes from, and in which format, will depend on the context of use. These metadata need to be populated into the application before it can be used, and this can be done through the user interface and requires no programming. This allows for more direct involvement of the domain experts who understand the Vaccination program that the software is expected for support.

Following are some of the details of Metadata being handled in DIVOC:

- Facility Registry
- Vaccine Registry
- Vaccine Program Registry
- Vaccinator Registry
- User Management
- Group Management
- Session Management
- Vaccination Rate Management

3.2. Schema Customization

A flexible data model supports different data sources to be integrated in one single data repository. Please find details to customize the schema as per the requirements:

https://github.com/egovernments/DIVOC/tree/main/registry

4. Setting Up a New Database/Instance

4.1. Strategies for getting started

- Can use ansible scripts provided in the repo to get started with basid DB instance. Replication and redundancy can be added as per need.
- Managed databases like RDS can also be used if needed

5. Deployment Strategies

- Decide options for container registry: hosted or public
- Setup kubernetes cluster
- Apply deployment files in the repo

6. DIVOC as Data Warehouse

6.1. Data warehouses and operational systems

DIVOC uses clickhouse for analytics, adhoc queries can be configured in redash (optional component)

6.2. Data storage approach

Relational backend engine is used by default (postgresql) One can configure to use cassandra or other backend storage supported by opensaber.

7. Integration concepts

DIVOC has been built on guiding principles of plug & play, architectural principle of API based micro services and Interoperable approach. This makes Integration a key aspect of the DIVOC platform. The platform flexibility to use any of its component independently as well as in combination, based on the needs of country/region. This makes it possible for the system to be adapted to a multitude of local contexts and use-cases.

7.1. Integration and interoperability

Based on DIVOC’s platform approach, DIVOC can receive and host data from different data sources and share it with other systems and reporting mechanisms. An important distinction of integration concepts is the difference between data integration and system interoperability:

- **Integration**- It is a process of connecting different information systems that they function or appear as one. It enables electronic data being available to all relevant users and harmonization of definitions and dimensions, so that the data can be combined or aggregated in useful ways.

  For ex – In India, DIVOC’s Vaccination module has been used for generating Digital certificates without any user interface and is acting as one of the components of the vaccination system that is being utilized by the country for COVID 19 vaccination drives.

- **Interoperability**- It is the end goal of integration and can be understood as the capability of information systems to exchange and interpret data in a common and meaningful manner. The interoperability is built into DIVOC by design so that it can be easily integrated with other software applications. For example, DIVOC-DHIS2 are interoperable, because they can share data definitions and data with each other.

DIVOC can be used as an integrated data warehouse for all Vaccination records, since it contains raw as well as aggregated data from various sources, such as Certifications Details of Vaccinators, Facility Masters, etc .

7.2. Objectives of integration

In the current scenario of robust and vast use cases across Vaccination, one system or platform will not be able to provide a complete end to end solution. For ex – Vaccination will include use cases around End to end Supply chain, Capacity building on system as well as on field operations and Vaccination with last mile connectivity. There is a need to integrate differently specialized and focused platforms to talk to each other in harmony to make the implementation successful. The approach will make it less costly and quicker to implement solutions, which can lead to a high diversity and sustenance of solutions.

This may lead to the conclusion, that all systems should be connected or that **interoperability is an objective**. However, the effort of connecting many redundant software solutions would be very high and therefore seems questionable. In a first step, focus should be on **reducing the number of parallel systems** and identifying the most relevant systems, afterwards these relevant systems can be integrated.

Major objectives of DIVOC integration approaches:

- **Reduce manual processing** and entering of data. This reduces the number of manual steps involved in data processing, which increases the data quality.
- **Reduce redundancies**: Often overlapping and redundant data is being captured by the various parallel systems. Hence reliable data sources integration is recommended, which will facilitate both quicker data entry and Single source of Truth.

7.3. Vaccination information exchange

Since there are different use-cases for Vaccination, there are different types of software applications functioning within the health sector for the Vaccination process. Each software application has their specific uses and data connections. The architecture functions as a plan to coordinate the development and interoperability of various subsystems within the larger health information system. It is advisable to develop a plan that covers all components, including the areas that are currently not running any software, to be able to adequately see the requirements in terms of data sharing. These requirements should then be part of specifications for the software once it is developed or procured. Following are the approaches based on the nature of software applications.

In order to assure that the data can flow smoothly, you need to have clear responsibilities on both sides of the system regarding data maintenance and troubleshooting. For example,

- There is need to define standard procedures for such activities as adding, renaming, temporarily deactivating or removing a facility on either of the two systems
- Changes of database fields that are included in a transferred data record and are required to be coordinated in a systematic way.

7.4. Different DIVOC integration scenarios

The different objectives described above lead to different integration scenarios. DIVOV can assume multiple **roles** in a system architecture:

- Data input: data entry (offline, online)
  - Manual Data Entry
  - Using CSV import: Master and Transitional Data
  - Web APIs: Master and Transitional Data

- Data storage and analysis with in-built tools
  - DIVOC uses redash (open source dashboard) for dashboard and reports. It can be customized based on requirements.
  - Custom reporting and analytics can be setup using apis

- Data sharing to external tools/application
  - Export functionality
  - Integration with Health Wallets
  - Web APIs with OAuth client secret model are used for integration

  - Use Case examples:
    - For India Digilocker is a digital credential access platform which integrates with divoc to store & access the vaccination certificates.

    - Arogya setu is a contact tracing application deployed during COVID outbreak and was integrated with custom extensions and OAuth based security.

7.5. Implementation steps for successful data and system integration

The purpose of this step-by-step DIVOC Implementation Guide is to provide a methodology for implementers to create and support a DIVOC integration scenario. The guide advocates for a country driven, iterative, and agile approach that begins with collecting user stories and functional requirements. The guide is intended as a framework that can be adapted to the specific context of each country. The content describes specific examples for each of the user stories, data specifications, job aids and checklists to guide the use of the reference implementation software.

Step 1: Identify Stakeholders and Motivations for Improved Facility Data

In a first step, the objectives of the integration project will be defined. This step is intended to answer the question: “Why do we want to integrate data from different sources with DIVOC?”

On a practical level, this leads to questions on the data integration approach, such as:

- Can you integrate the master/transactional data into?
- What data do you want to Integrate?
- Do you want to remove redundant data?
- If you want to create a data exchange connection between DIVOC and another system, how do you define ownership and responsibilities?

Activities to answer this question are described below and will lay the groundwork for an DIVOC interoperability solution.

Identify Stakeholders and Motivations

Any interoperable project implementation involves involvements from more than one stakeholder. Stakeholders from different areas need to agree on a common system approach, for example the team responsible for the capturing Vaccinations and the Logistics Department in case of an LMIS implementation.

Therefore, it’s interesting to look at the main motivations of the stakeholders and how to mitigate risks resulting from potential diverging interests.

By identifying who is interested to provide or utilize the data, the lead implementers can start to form a project team to inform the design and implementation. One method for characterizing stakeholders involves grouping interested parties by their functional roles. The existing infrastructure and procedures are also important to understanding governance and curation options. Understanding the stakeholders and their corresponding systems is a critical first step.

Explore Opportunities and Challenges

The motivations driving an implementation can be detailed by the perceived opportunities or challenges that stakeholders face. This might include the desire to share data across systems related to vaccinating facilities for supply chain management, monitoring and evaluation, health service delivery and many other systems.

Step 2: Document Facility Registry Specifications and User Stories

- Collect existing metadata
- Document data specifications
- Document user stories

Step 3: Set Up Initial Instance

- Implement the specifications
- Identify and prioritize incomplete user stories

Step 4: Identify Gaps & Iterative Development via User Testing

- Agile and iterative development
- User testing
- Collect, reconcile and upload data

Step 5: Scaling the Registry Implementation

- Confirm user roles and responsibilities
- User training
- Critical integrations

Step 6: Provide Ongoing Support

While during the initial implementation phase a temporary support structure should be available from the central DIVOC team from eGov Foundation, afterwards a permanent support structure needs to be set-up at country level on each Integrating system. The main challenge is to have clear responsibilities. In an ideal situation, we are dealing with two stable systems that each have already their own clearly defined support structure.

Interoperability tends to be a tedious technical and organizational charge. All the initiatives will consume efforts of technical and functional resources to update APIs. In addition, with each new release of any involved system, data flows require retesting and if necessary, adaptations. To be successful these implementation projects typically must go through a series of complex steps, such as

- The agreement on an interoperability approach embedded in the national eHealth strategy,
- The definition of data standards and sustainable maintenance structure,
- Attaining a stakeholder consensus on data ownership and sharing policies.

A solution could be to discuss the new responsibilities beforehand, assigning them to job descriptions, teams and specific positions.

7.6. Specific integration and interoperability use cases

DIVOC has been expanding its reach into many health systems such as DHIS2, CoWin, AMPS and other Health Wallet applications. This is also reflected in the overall system architecture. Following are some of the Integration and Interoperability use cases for DIVOC.

#	Availability of Systems	Scenarios	Prerequisite	Adoption Approach
1.	Country wide Vaccine Supply Chain System with last mile connectivity	Not Available	1. Manual Entry of Stock: ● Facility wise availability of vaccine stock 2. Stock Entry via Import (Central/ State/ Facility level): ● Facility wise availability of stock in provided standard format (Excel/CSV)	● DIVOC can be customised to provide feature for manual entry of available stock to System/Facility Admin ● DIVOC can be customised to provide import functionality for entry of available stock to System/Facility Admin
		Connectivity till Facility Level	1. Vaccine Supply Chain System should be API Integrable for exchanging stock consumption with (Type, Serial Number, Validity, etc)	● DIVOC forms two-way communication on stock availability and consumption with Supply chain system
		Available till Patient level	1. Vaccine Supply Chain System should be API Integrable 2. Vaccine Supply Chain System should share: ● Consumed Vaccine details ● Vaccinator ● Location (Facility) details ● Timestamp 3. Common Facility Registry in both systems	● System captures details on Vaccination & Vaccinator for issuance of certificate

			4. Common Vaccine Registry in both systems	
2.	Learning Management System (LMS) for Vaccinator	Available	1. LMS should be API Integrable 2. Unique Vaccinator Id in both systems	● Future versions of DIVOC will integrate with LMS for Verification of Vaccinator Certificates
		Not Available	1. Availability of Physical / Digital Copy of Vaccinator certificate	● Manual Verification of Vaccinator Certificates will be performed at Central / State/ Facility Level
3.	Drug Databank / Vaccine Registry	Available	1. Drug Bank / Vaccine Registry should be API Integrable 2. Vaccine Rate Card (for Paid Vaccination) should be available 3. In Case of Multiple Authentic Drug registry: Drug registry with most relevant drug details to be selected for Integration	● DIVOC integrates with Vaccine Registry to fetch Vaccine details
		Not Available	1. Availability of Vaccine Details	● Manual creation of Vaccine details by System Admin
4.	e-Payment System	Available	1. API Details of Payment gateway 2. Registration on Payment gateway 3. Service & Vaccine Rate card	● Future version of DIVOC can be integrated with Payment Portal for Paid Vaccination
		Not Available	1. Service & Vaccine Rate card	● DIVOC provides flexibility to extend and capture vaccine and service rates.
5.	Digital Citizen Registry (EHR/ PHR/ HIS / National ID / National/Regional disease surveillance registries)	Available	1. Citizen Registry should be API integrable 2. Authorization to connect with Citizen Registry OR Multiple Authentic Citizen registry but not covering 100% of Recipient 2 Selection of registry with maximum relevant entry	● DIVOC integrates with Recipient Registry to enrol for Vaccination (For ex BPL Recipients to be enrolled by Government Agency/Dept.) ● DIVOC will perform Online verification of the recipient ● Manual enrolment and verification of Recipient (Bulk/Single) which are eligible but not yet part of selected integrable registry
		Not Available	1. Availability of Physical document to verify Recipient	● Manual enrolment and verification of Recipient (Bulk/Single) in DIVOC ● Manual verification of any identified documents for Recipient
6.	Health Platform	Available	1. Health Registry should be API Integrable 2. Availability of Facility Data	● DIVOC integrates with Health Registry to fetch required data of

	(Facility/ Vaccinator Registry)		3. Availability of Vaccinator data mapped to Facility 4. In Case of Multiple Authentic Health registry: selection of Registry with most relevant data	Recipient/Facility/Vaccinator
		Not Available	1. Availability of Facility and Vaccinator Details in required format	● Manual creation of Facility/Vaccinator details (Bulk/Single) in DIVOC
7.	Digital Signature	Available	1. API Integrable: e-Sign/Digital signature Platform	● Certificates will be Digitally signed
		Not Available	None	● Certificates will be manually signed & stamped by authorised signatory
8.	Health insurance (Public/Private) systems	Available	1. API Integrable – Insurance Platform	● DIVOC can be customised to capture Insurance Id ● DIVOC can be customised to Fetch Insurance details for verification and avail benefits
		Not Available	None	NA
9.	Digital Document Vault (National / Regional)	Available	1. API Integrable – Document Vault	● DIVOC can be tweaked to push/store Certificates on Digital vault ● Recipient can download certificates from Document Vault or DIVOC Platform
		Not Available	NA	● Certificates will be issued over Mail/ In Hard & Soft Copy / WhatsApp (if services are procured)
10.	Application Access for Facility Staff/Admin	● Mobile No./ Email availability **OR** ● Mobile No./ Email Id linked to National Id/ Employee Id	None	● Mobile Number / Email & OTP based login
		National Id/ Employee Id and Mobile & Email are not available for Staff/Admin	NA	● Manual User creation by System/Facility Admin and login by provided Id / Password
11.	Application Access for Recipient	● Mobile/ Email	None	● Mobile Number / Email & OTP based login & Access to issued

		availability **OR** ● Mobile No./ Email Id linked to selected Recipient registry		Certificate/Appointment Details
		No Mobile No./Email available with Recipient	Availability of Physical copy of any ID Proof	● Manual Enrolment & Verification of Document ● Provisioning printed copy of Certificate

8. Support

The DIVOC community uses a set of collaboration and coordination platforms for information and provision of downloads, documentation, development, source code, functionality specifications, bug tracking. This chapter will describe these in more detail.

8.1. Home page

The DIVOC home page is found at *https://divoc.egov.org.in/*. The page provides understanding of DIVOC, Technical Documents, FAQs, About project along with various documents. The source code of the application is managed under eGov Foundation (https://github.com/egovernments/DIVOC).

The *Tech Docs* page provides installation instructions, user documentation, this implementation guide, presentations. The user documentation is focused on the practical aspects of using DIVOC, such as how to operate various functionalities of DIVOC.

8.2. Development tracking

- Using Internal tracking board and Github issues.
- Weekly sprints with prioritized stories
- Releases pushed to docker hub public repository and github releases.

8.3. Source code: github.com

DIVOC uses *GitHub* as the main collaboration platform. The site can be accessed at https://github.com/egovernments/DIVOC is used for source code hosting, functionality specifications, bug tracking and notifications. The various source code branches including trunk and release branches can be browsed from the same location.

If you want to suggest new functionality to be implemented in DIVOC you can put your suggestions on the https://github.com/egovernments/DIVOC/discussions and eventually write a specification. The same will be considered by the core development team and if accepted be assigned a developer, approver and release version.

If you find a bug in DIVOC you can report it at GitHub by navigating to https://github.com/egovernments/DIVOC/issues Your bug report will be investigated by the developer team and be given a status. If valid it will also be assigned to a developer and approver and eventually be fixed. Note that bug fixes are incorporated into the trunk and latest release branch - so more testing and feedback to the developer teams leads to higher quality of your software.

9. Users and user roles

9.1. Roles

- System admin
  - Define a super user using keycloak admin login and associate system admin role to it.
- Controller
  - Controller is responsible for setting up daily rate of vaccination and activating the facilities
- Monitoring Authority
  - Analytics reports and high-level view of the program execution
- Facility Admin
  - Sets up facility details
  - Defines schedule
  - Adds staff
- Facility staff
  - Vaccination data entry and validation
- Recipient
  - Registers to vaccination program
  - Accesses vaccination certificate.

10. Localization of DIVOC

Facility application is supporting localization of interfaces. Define custom translations as required.

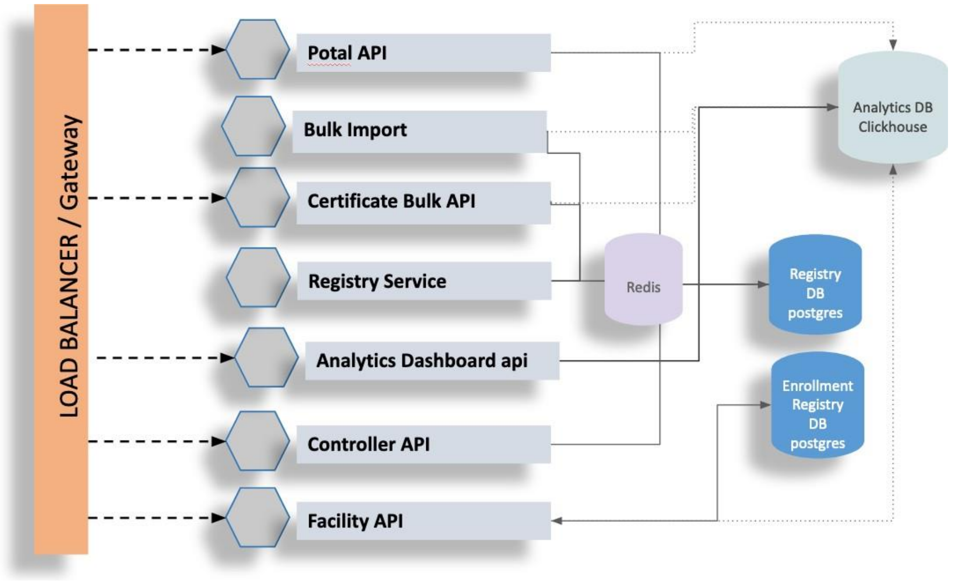
11. DIVOC Tools Guide

11.1. Overview

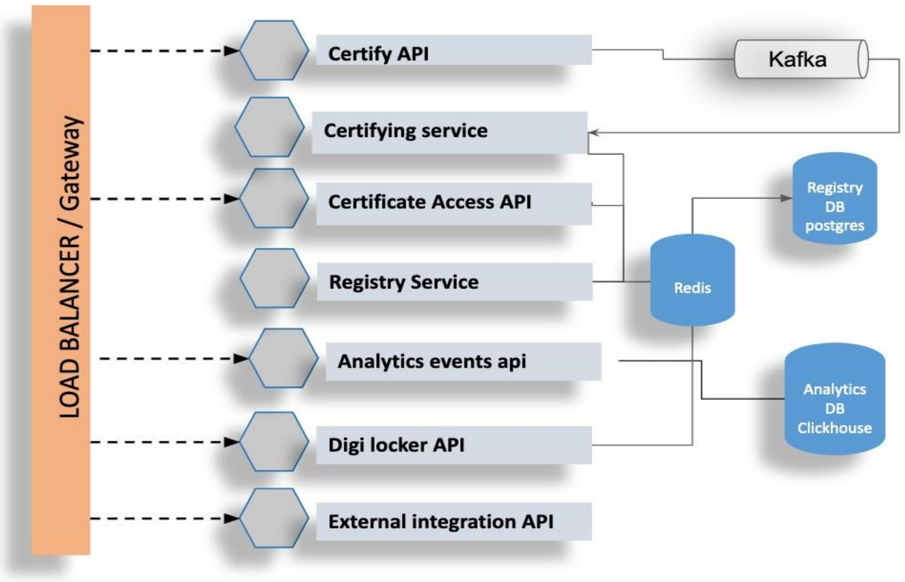
- DIVOC is build using open source tools and technologies leveraging asynchronous message driven architecture
  - opensaber
  - kafka
  - redis
  - clickhouse

11.2. Architecture

- Event based architecture with horizontally scalable components



- Certificate Module



11.3. Installation

Please refer this link for Installation details: https://divoc.egov.org.in/tech-docs#getting-started

11.4. Troubleshooting guide

- when running with docker compose

- check logs with docker-compose logs -f
- check for order of components at the start, if any service is down. example: docker-compose start registry
- setup log aggregation in kubernetes to watch any errors.

12. DIVOC Documentation Guide

12.1. DIVOC 2 Documentation System Overview

DIVOC is a web-based information management system under very active development with regular releases. Each release typically includes several new features and additional functionality. Given the fast pace of development, the system's wide user base and distributed, global nature of development, a comprehensive documentation system is required. The DIVOC documentation is written using GitBook.

In this chapter, we will describe the documentation system of DIVOC and how you can contribute

12.2. Getting started with GitHub

The DIVOC documentation system is managed at GitHub in its own source code repository. GitHub is a collaborative platform that enables multiple people to work on software projects collaboratively. For this to be possible, a version control system is necessary in order to manage all the changes that multiple users may make. GitHub uses the *git* source control system. While it is beyond the scope of this document to describe the functionality of *git*, users who wish to source code and documentation around it will need to gain at least a basic understanding of how the system works. A basic guide is provided in the next section. The reader is referred to the git manual for further information.

In order to start adding or editing the source code, you should first perform a checkout of the source code. If you do not already have a GitHub account, you will need to get one. This can be done here.

Login to GitHub, and then file an issue here. Your request will need to be approved by the group administrators. Once you have been granted access to the group, you can commit changes to the documentation branch and send and receive notifications if you wish. Alternatively, you can clone the documentation into your own repository, commit your changes to your own fork, and request that your changes be merged with the source of the documentation with a pull request.

12.3. Committing your changes back to GitHub

Once you have finished editing/updating your source code, you will need to commit your changes back to GitHub. Open up a command prompt on Windows or a shell on Linux and navigate to the folder where you have placed your documentation. If you have added any new files or folders to your local repository, you will need to add them to the source tree with the git add command, followed by the folder or file name(s) that you have added. Be sure to include a descriptive comment with your commit.

```
git commit -m "Improved documentation on organisation unit imports with CSV."
```

Finally, you should push the changes back to the repository with git push origin mybranch, where "mybranch" is the name of the branch which you created when you checked out the document source or which you happen to be working on. In order to do this, you will need the necessary permissions to commit to the repository. When you have committed your changes, you can issue a pull request to have them merged with the master branch. Your changes will be reviewed by the core documentation team and tested to ensure they do not break the build, as well as reviewed for quality. As mentioned previously, you can also push your changes to your own GitHub repo, if you do not have access to the main repo and submit a pull request for your changes to be merged.