# MOSIP-HIGH LEVEL DESIGN

# Purpose

The purpose of this Document is to describe the high level design aspects of MOSIP.

# Revision history

The Revision history table shows the date, changes, and authors who have worked on this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Version/Change request number | Version date | Description of changes | Author |
| 1.0 | 25/01/2019 | First Draft | Rahuul & Resham |
| 2.0 | 05/02/2019 | Modified Version | Rahuul & Resham |

# Architecture

MOSIP Architecture is defined in five (5) separate sections which are detailed in GitHub wiki. Click on each specific header name to navigate to wiki for further details.

1. [Principles](https://github.com/mosip/mosip/wiki/Architecture-Principles-&-Platform-Goals)

This section consists of the foundational principles of MOSIP based on which the architecture is defined. The key principles considered include: Open Source and Vendor Neutral, Adaptability, Security, Multi Party, Authorization, Authentication, Multi Language Support, Performance and Scalability, High Availability, and Auditability.

1. [Logical View](https://github.com/mosip/mosip/wiki/Logical-Architecture)

This section details the key design aspects considered for MOSIP. This includes Ecosystem Approach, Configurability, Extensibility, Modularity, and Solution Principles.

1. [Technology Stack](https://github.com/mosip/mosip/wiki/Technology-Stack)

This section lists all the technologies used in building MOSIP.

1. [Process View](https://github.com/mosip/mosip/wiki/Process-view)

This section provides a view into the functional processes of *Pre-registration*, *Registration Client, Registration Processor*, and *ID Authentication*.

1. [Data Architecture](https://github.com/mosip/mosip/wiki/MOSIP-Data-Architecture)

This section details the data architectural principles of MOSIP which includes Security, Multi-Language, High Availability, Auditability, and High Performance. It also details the Data Models and its Naming Standards.

# Design

The design section is broadly classified to two (2) sections: *Functional* and *Architecturally Significant Components.*

1. Functional:

This section details the design aspects of MOSIP, driven by the four (4) key functional modules as listed below. Click on each specific header name to navigate to wiki for further details.

* + 1. [Pre-Registration](https://github.com/mosip/mosip/wiki/Pre-Registration)

Pre-registration is the module which is the web channel of MOSIP for pre-registering an applicant prior to Registration. This section elaborates on the key design considerations for Pre-registration which include micro-service based architecture, Data validation as executed in UI, key design patterns like aggregate service pattern and proxy design pattern. It also elaborates on the architecturally significant use cases, the Process View and Conceptual View.

* + 1. [Registration Client](https://github.com/mosip/mosip/wiki/Registration-Client)

Registration Client application captures the Demographic and Biometric details of an Individual along with supporting information (proof documents & information about a parent/guardian/introducer) and packages the information in a secure way. This section provides details on the architecturally significant use cases of Registration Client, which include the ability to adhere to industry standards, facilitate secure data transmission, Process View and Logical View, to name a few.

* + 1. [Registration Processor](https://github.com/mosip/mosip/wiki/Registration-Processor)

Registration Processor processes the data (Demographic and Biometric) of an Individual for quality and uniqueness and then issues a Unique Identification Number (UIN). This section elucidates the architecturally significant use cases of Registration Processor, which include ability to be scalable, facilitate integration, look-up of the Process View and Logical View, to name a few.

* + 1. [ID Authentication](https://github.com/mosip/mosip/wiki/ID-Authentication)

MOSIP ID Authentication provides an API based authentication mechanism for entities to validate Individuals. This section provides details on the architecturally significant use cases of ID-Authentication, which include specifics on API standards, the Process View and Logical View, to list a few.

* + 1. [Kernel](https://github.com/mosip/mosip/wiki/Kernel)

Kernel is a platform to build higher-level services as well as a secure sandbox. This section provides details on the active framework of MOSIP, its structure & rules within which the higher-level services operate and the architecturally significant use cases.

1. Architecturally Significant Components:
2. [ID Object Definition](https://github.com/mosip/mosip/wiki/MOSIP-ID-Object-definition)

ID definition describes the attributes that a Country or entity intends to capture from an Individual, which will formulate the definition of ID for a Country. This section elaborates on the mechanism MOSIP adopts, in order to provide the flexibility for each Country to define its preferred ID definition through an ID object definition schema.

1. [MOSIP Configuration](https://github.com/mosip/mosip/wiki/MOSIP-configuration-&-launcher)

MOSIP as a platform will have multiple applications running and each application will have a set of configurations.

This section details:

* The key configuration files that a system owner has to create before starting the platform – With a centralized Config Server.
* Launcher component which will read the configuration files, validate and launch the platform.

1. [Registration Packet Structure](https://github.com/mosip/mosip/wiki/Registration-Packet)

This section illustrates the packet creation flow along with the encryption process, as part of Registration Client.

1. [MOSIP ABIS Middleware](https://github.com/mosip/mosip/wiki/MOSIP-ABIS-Middleware)

This section provides details on the ability of MOSIP to support a single or multi-ABIS solution. It also provides specifics on the Components & APIs of ABIS Middleware, Strategies for Biometric Data Management in ABIS and Strategies for de-duplication in case of multiple ABIS systems.

1. [ABIS Interface Spec](https://github.com/mosip/mosip/wiki/Automated-Biometric-Identification-System-(ABIS)-Interface)

This section provides the specifications that an ABIS provider must implement to meet MOSIP's requirements.

1. [MOSIP Biometric Data Spec](https://github.com/mosip/mosip/wiki/MOSIP-Biometric-Data-Specifications)

This section details out the specifications for Biometric data during data acquisition and verification.

1. [MOSIP VDM Spec](https://github.com/mosip/mosip/wiki/MOSIP-VDM-Specifications)

This section illustrates the VDM technical specifications to be adhered to by a vendor, who intends to adopt their devices to the MOSIP platform, so as to capture the Biometric data and process the same.

1. [Security](https://github.com/mosip/mosip/wiki/Security)

Multiple aspects of Security like Confidentiality, Privacy, and Integrity of data are key in ensuring an Individual's identity is not compromised. This section illuminates on the Security design principles that MOSIP follows.

1. External Integration API

This section illustrates the integrational specifications of MOSIP with an external system – WIP.

1. [Test Rig Design](https://github.com/mosip/mosip/wiki/Test-Rig-Design)

Test Rig represents a one click automation to build, deploy and test a software module. Successful execution of test rig would ascertain complete setup of the MOSIP platform.

Click on the header name <TEST RIG> to navigate to wiki for further details.