Technical Note
DIKSHA

Document Name:	Technical note on DIKSHA (Digital Infrastructure for School Education)
Document Version:	5.0 A (Draft)
Last updated on:	Dec 23, 2022

Table of Contents

About the Document

Introduction

This document is a technical note on DIKSHA, the digital infrastructure for school education, an initiative of the National Council of Educational Research and Training (NCERT), Ministry of Education, Government of India.
The goal of DIKSHA is to democratize learning. It provides a common platform that the education ecosystem may leverage to participate in and contribute towards, to achieve the country's learning goals.
DIKSHA is built using MIT licensed open source technology called Sunbird that incorporates internet scale technologies and enables several use-cases and solutions for teaching and learning. Its federated architecture allows any Indian State/UT to use the DIKSHA platform to design and run programs for school education, foundational learning and to support inclusive learning for the underserved and differently-abled communities of learners and teachers.

Objectives of this document

This document aims to provide an overview of:

- The architecture and design principles on which DIKSHA is built
- The processes used to build, test, deploy, monitor, support and measure DIKSHA operations
- The solutions built specifically on the DIKSHA platform to address consumption and sourcing needs for India's education ecosystem
- DIKSHA as a Digital Public Good
- DIKSHA's future vision

Document structure

This document has six (6) parts - **A, B, C, D, E and F**
Part A contains information on DIKSHA's Architecture and Design. It provides information about

- **DIKSHA Architecture Design Dimensions & Principles** elaborates the five design dimensions and 20 design principles on which the DIKSHA platform is architected.
- **DIKSHA Building Blocks** describes the underlying Sunbird building blocks used to instantiate the DIKSHA platform.
- **Open Source Software or Specifications** provides a list of software or specifications used in DIKSHA

Part B contains information on DIKSHA Operations. It contains sections that provide information about

- **DIKSHA Build Process:** This section provides information on the process followed for every DIKSHA version release. It also provides information and samples of the typical artifacts generated during different stages of the build process
- **DIKSHA Quality Assurance Process:** This section elaborates the process followed to test the DIKSHA platform for stability, performance and quality before it is rolled out to production. The section also provides information and samples of artifacts generated in the quality assurance phase.
- **DIKSHA Deployment Process:** This section provides information on the process followed to deploy a typical DIKSHA version release. It also provides information on the software requirements and the detailed installation manual.
- **DIKSHA Monitoring Process:** This section provides information on the process of monitoring DIKSHA services on a regular basis. It also provides information on the tools used to monitor the DIKSHA platform.
- **DIKSHA Support Process:** This section provides information on the S1, S2 processes followed to provide support to DIKSHA users and the lifecycle of a support ticket from the point it is raised until its logical closure.
- **DIKSHA Telemetry Instrumentation:** This section provides information on the telemetry instrumentation done for DIKSHA. It contains information on the telemetry specifications, how telemetry is sent, consumed and the libraries used. There is a separate section on data sets and data exhausts and how these are made available to DIKSHA users (States, Organizations).
- **DIKSHA Measurement Processes:** This section provides in-depth information on DIKSHA's reporting infrastructure, the configuration required, and processes to be followed for report generation and requests.

Part C contains information on DIKSHA solutions. It contains two sections based on the solution categories:

- **Consumption Solutions**: This section provides information on DIKSHA's consumption solutions. The consumption solutions are software developed to enable digital solutions for learning, capacity building, or professional development solutions for India's education ecosystem consisting of educationists, experts, organizations, institutions - government, autonomous institutions, non-govt and private organisations.
- **Sourcing Solutions:** This section provides information on DIKSHA's content sourcing solution. The sourcing solution allows actors in India's education ecosystem to source content for DIKSHA by creating, curating, collaborating and/or crowdsourcing it.

Part D contains information on DIKSHA as a Digital Public Good (DPG).

Part E contains information on DIKSHA's future vision.
Part F contains a list of appendices from Appendix A - X. Overall, the appendices cover detailed relevant and supporting documentation generated for DIKSHA. Each appendix covers one area.

List of Abbreviations

Acronym	Expansion
AI	Artificial Intelligence
API	Application Programming Interface
API	Application Programming Interface
AWS	Amazon Web Services
BEO	Block Education Officer
CBSE	Central Board for Secondary Education
CRP	Community Resource Person
DPG	Digital Public Goods
DPI	Digital Public Infrastructure
FA	Federated Architecture
FRS	Functional Requirements Specifications
GPL	General Public License
HM	Head Master
HP	Himachal Pradesh
MeitY	Ministry of Electronics & IT
ML	Machine Learning
MoE	Ministry of Education
NDEAR	National Digital Education Architecture
NOC	Network Operations Center
OCR	Optical Character Recognition
PRD	Product Requirement Document
RCA	Root Cause Analysis

Acronym	Expansion
S1	Severity 1
S2	Severity 2
SRS	Software Requirements Specifications
SQAA	School Quality Assessment and Assurance
TPS	Transactions Per Second
UP	Uttar Pradesh
WCAG	Web Content Accessibility Guidelines

Part A

DIKSHA Architecture & Design

DIKSHA Architecture Design & Principles[1]

Overview

The large, complex and diverse nature of India's education ecosystem necessitates that the technology architecture and design of DIKSHA addresses a wide range of needs and uses. A good design helps accelerate adoption, helps in achieving the nation's education-related goals, improves capacity building for teachers, and aids education administrators in tracking and monitoring programs, while also addressing India's diversity through innovative integrated capabilities across contexts. As a technological framework to deliver learning needs for India, DIKSHA aims to make available building blocks for the creation of new tools and solutions for a unifying architecture, while leveraging existing systems to upgrade and become interoperable. The architecture intends to enable innovation opportunities for different stakeholders and amplify innovations and solutions in an exponential manner. Overall, the architecture of DIKSHA facilitates the development of a platform that is federated but interoperable, while ensuring autonomy to all relevant stakeholders(especially States/UTs) enables participation of the ecosystem and also individuals.

[1] The architecture and design principles adopted by DIKSHA are similar to those prescribed in and adopted by other government policies, frameworks and platforms like MeitY open source policy, NDSAP Policy for open Government data, India Stack, NDEAR, etc.

Tenets for NDEAR Framework Elements

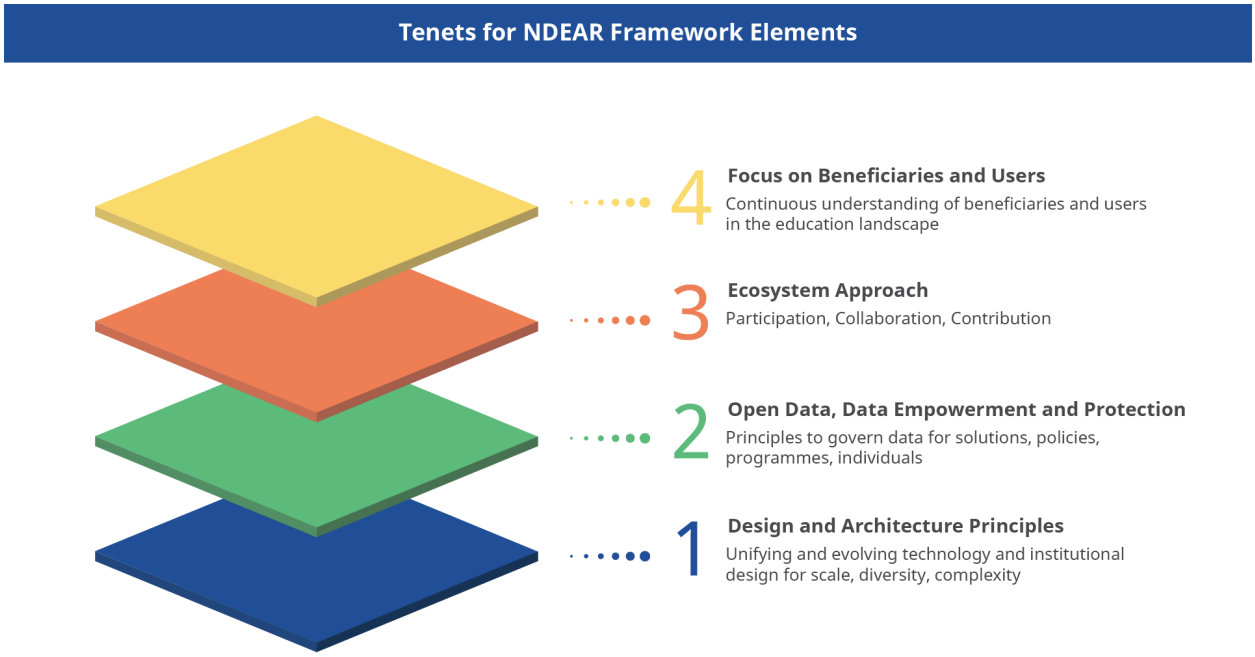

Figure 1: Tenets for NDEAR Framework

The NDEAR main report[2] prescribes four tenets to develop digital infrastructure for education. They are:

- **Focus on beneficiaries and users using principles** for:
    - User-centric design to develop infrastructure and solutions, and use universal learning design principles to support greater inclusion
    - Affordance, choice and agency to create diverse solutions with multimodal coherent access that enable a learning continuum
    - Open access to assets so that learning happens anytime, anywhere, for anyone in a formal, informal or alternate setting
    - Diversity and inclusion to accommodate different languages, geographies, device types, access (online/offline), spectrum of special needs for learners with learning difficulties, the differently abled and accessibility needs
- **Adopt an 'ecosystem' approach NOT a 'system' approach** as no single application, portal, technology can address India's education challenges. The key factors to foster an

[2] NDEAR Main Report - https://www.ndear.gov.in/resources.html

effective educational ecosystem are participation, collaboration and contribution by the various stakeholders. In order for the ecosystem to organically evolve, the NDEAR report suggests that the ecosystem must be:

- Catalyzed through innovation, experimentation and contextualization for diversity, inclusion and special needs
- Energized across actors, geographical levels (Center, State, District, Block, School)
- Supported by policies and infrastructure
- Supportive of the programme, asset and solution ecosystems across government, non-profit and private sectors
- Governed by nine principles to catalyze and energize the ecosystem players

- **Create clear principles to govern open data, data empowerment and protection.** The principles suggested for this tenet include:
  - Open Data for evolvability of solutions and policies.
  - 'Emit' (vs 'Extract') of data streams from usage, user interactions and outcomes.
  - Follow open data policies in the interest of transparency and evolution.
  - Data empowerment and protection of individuals' privacy and confidentiality.
  - Privacy by Design: Privacy of beneficiaries of the platform is of utmost importance, and special care must be taken with regards to the data of children.
  - Personally Identifiable Information (PII) and data collection must be minimal, limited, used for stated specific purposes and must be non-intrusive.
  - Consent architecture must be developed and strictly followed. Particular attention must be made to the design of technology and processes for clear and explicit consent. Special attention should be paid to ensuring the protection of children and seeking correct and appropriate consent for specific engagements and actions from parents and guardians. Sweeping, generalized consent processes, practices, terms and design must be discouraged.
  - Users must have appropriate access, control and visibility to their personal data.
  - Protection of Children: Children must be protected against identity tracing, tracking, labelling and discrimination. Protecting the privacy and confidentiality of their performance and personally identifiable data must be of paramount importance.
  - Data Lifecycle: Entry, Storage, Usage, Control, Retention and Deletion
    - Single entry for a data record - avoid inefficient duplication of effort.
    - Single source of truth.
    - Accountable access, distribution and storage of data.
    - ederated control structures.
    - Integrated visibility of data for policy evolution.
    - Lifecycle of data from generation - storage - deletion

- **Maintain Design and Architecture Principles that are unifying, evolving and relevant** for technology as well as institutional design in the context of scale, diversity and complexity.

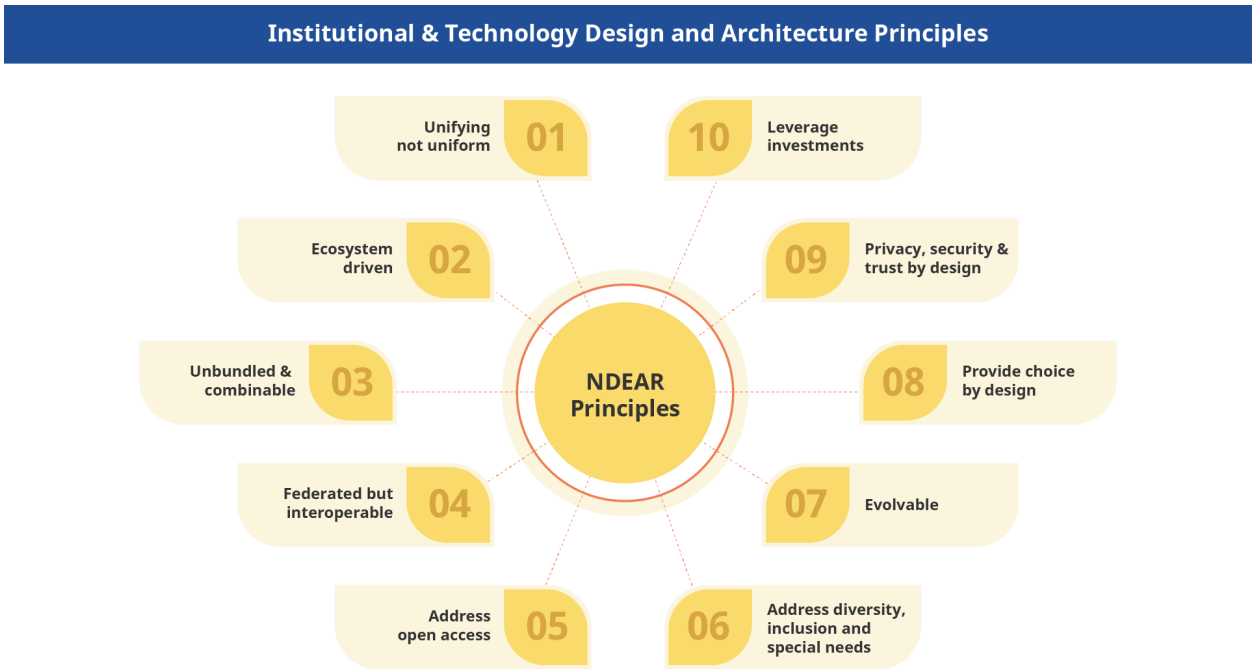

Figure 2: NDEAR Institutional & Technology Design and Architecture Principles

No.	Principle	Description
1	Unifying not uniform	India's diversity must be retained and unified in a coherent way for the amplification of efforts by all actors.
2	Ecosystem driven	No single app/portal/technology can solve India's education issues. Energising the ecosystem across governments, society and market players is necessary to bring in innovation, diversity, contextualisation and choice. Continuously engaging the ecosystem through transparent rules and a set of relevant tools is key to harnessing the energy of the ecosystem actors.

No.	Principle	Description
3	Unbundled and combinable	Instead of focusing purely on “solutions”, build underlying unbundled building blocks to allow diverse solutions by the ecosystem. Fully pre-built systems cannot evolve and allow for diverse solution needs. All aspects of NDEAR must be built as unbundled microservices offered via APIs that are interoperable and can be combined to build newer solutions.
4	Federated, but interoperable	While a central system can speed up adoption, it should be a choice, and bringing interoperability across many federated systems through common specifications is necessary.
5	Address open access	Anyone can learn, anyone can help learn, anyone can support and administer. Open license and open-source policies are key.
6	Address diversity, inclusion, and special needs	Design to cater to diversity across language, context, device, connectivity, capacity, etc., and ensure the system addresses users with special needs, with relevant tools and content. The system shall provide uniform access to everyone, including those in remote areas, with limited or no connectivity, and support the end-point devices at the root level.
7	Evolvable	Design must allow the entire infrastructure to evolve and adapt on a continuous basis without needing dramatic upgrades.
8	Provide choice by design	Design to allow choice for actors (States, boards, schools, teachers, students, etc.) with respect to choice of apps, choice of usage within their context, etc.
9	Privacy, security and trust by design	Design to protect privacy and data of users and entities and imbue trust in every interaction.

No.	Principle	Description
10	Leverage investments	With respect to existing systems at various administrative levels (Center, State, and school), it is essential that they be leveraged by upgrading them as per NDEAR architecture principles, integrating them with other NDEAR building blocks, and when necessary consolidating them into unified platforms. This is essential to provide simpler, coherent, and unified experiences to key stakeholders and also to bring them in line with NDEAR.

DIKSHA Technology Architecture Design and Principles

DIKSHA's technology architecture and design principles have evolved from the tenet of **unifying, evolving and relevant design and architecture principles** as advocated in the NDEAR report. The following are DIKSHA's 20 foundational technology principles:

1. Federation & Choice

The DIKSHA platform is designed to enable actors in the learning ecosystem with the autonomy and choice to decide how to join the platform (e.g., different device types) and what to join for (e.g., to learn, to earn credentials, observe learning programs, etc.). It provides autonomy and choice to institutions and users in terms of building appropriate content, supporting different taxonomies, creating contextual solutions, and fully supporting a seamless interplay between physical-digital mediums, synchronous-asynchronous modes and diverse device types.

2. Decentralized & Distributed

The DIKSHA platform is designed to allow interoperability and information sharing between various entities, information technology systems and applications. All DIKSHA services, data, and applications are held at multiple levels - say national, State and local levels - in a decentralized manner, following the principle of minimalism at each level. For example, teacher data and registries are maintained by the States, student data may be maintained by the district, municipality or ward. All systems are designed to be operated in a federated model. They can function independently but in an interoperable manner. In this direction, DIKSHA will also be distributed with interactions happening in different systems managed and run by different entities. For example, each State may have their own systems/solutions built on the core DIKSHA platform facilitating all their learning interactions and at the same time being interoperable with each other. Each State may also build their own applications on the DIKSHA platform.

3. Powered by Building Blocks

The DIKSHA platform is built using underlying digital building blocks (e.g., learning services) that are atomic, independent and reusable. Unbundled components/services are offered via APIs and other well-defined mechanisms to allow diverse solutions by the ecosystem and to ensure that the systems can evolve.

Building block-driven design enables adaptability and configuration of solutions of various kinds. Building blocks have to interoperate with other building blocks within the same system or across systems. As long as they are compliant (adhere to laid down platform policies and rules of engagement) and compatible (adhere to various specifications adopted by the platform such as API specifications, data specifications, content specifications, etc.) with the DIKSHA architecture, they may be used by other relevant entities to develop solutions and programmes.

4. Platform Thinking

The DIKSHA infrastructure allows the ecosystem of stakeholders (e.g., Boards, training institutions, States, content providers, etc.) to use, create solutions on top, and extend their solutions/services for education. The entire platform design allows the ecosystem to use, extend, build their own assets, create their own processes, build their own solutions, all on top of the core platform.

5. Unifying not Uniform

To retain and unify India's diversity in a coherent way for the amplification of efforts by all actors, the DIKSHA platform acts as a unifying force and does not force uniformity across the highly federal and diverse system. The DIKSHA platform thus is not one solution that fits all, but is a platform consisting of a set of digital building blocks that can be leveraged by the federated system (e.g., by the Center, State/Board, School) in their own way, by integrating into their own systems, processes, and practices freeing the platform custodian from becoming a solution provider for all use cases, user scenarios and user types.

6. Enabling Ecosystem

No single application, portal, technology can address India's education challenges. The key factors to foster an effective educational ecosystem are participation, collaboration and contribution by the various stakeholders. The DIKSHA platform is designed to enable the ecosystem and to ensure that actors across the spectrum are energized and catalyzed to innovate, experiment and contextualize solutions for diversity, inclusion and special needs. They can operate and deliver necessary and wide variety of:

- Assets (e.g., diverse content),

- Programmes (e.g., Central, State or School-level education initiatives),
- Solutions (e.g., learning services, networking services), and
- Applications (e.g. State-specific DIKSHA portals and mobile apps).

Continuously engaging the ecosystem (Government, civil society and private organizations) through transparent rules and a set of relevant tools is key to harnessing the energy of the ecosystem actors.

7. Inclusive by Design

User Base

Keeping in mind NDEAR's tenet of focusing on users and beneficiaries, DIKSHA uses user-centric design practices to develop infrastructure and solutions using universal learning design principles to support greater inclusion. It supports accessibility best practices (such as WCAG principles[3]) catering to users with special needs, and inclusive by taking into account aspects such as users' digital proficiency, language barriers, and access to the Internet and addresses a spectrum of special needs for learners with learning difficulties, the differently abled with accessibility needs. It provides affordance, choice and agency to create diverse solutions with multimodal coherent access that enable a learning continuum.

DIKSHA addresses the language diversity of India by giving special focus to multilingual content, technologies to enable dictionary/wordnet services, input capture services, speech detection, translation services, multilingual portal, etc. It leverages the many Indian and international open-source projects and assets built within various Government and research institutions by upgrading and integrating them as reusable open-source assets or services to amplify and extend the work. With the advancement in AI/ML and wide access to mobile devices (with camera and microphone), it ensures these technologies are leveraged to enable millions of users to interact, discover, learn, and collaborate in Indian languages.

DIKSHA Solutions achieve universal access by providing multi-channel, coherent experiences (not siloed and broken), across different mediums such as mobile, tablet, web, and desktop. For example, a user accessing a training course on a mobile phone or tablet can resume the training on a desktop seamlessly. In this way, the system provides access to everyone, including those in remote areas, with limited or no connectivity, and supports the end-point devices at the root level.

[3] Web Content Accessibility Guidelines (WCAG) - https://www.w3.org/WAI/standards-guidelines/wcag/

Systems & Processes

Keeping NDEAR's tenth tenet to leverage investments as a guide, DIKSHA's federated architecture is designed to connect and leverage varied systems and processes including the applications, solutions and content that currently exist in other systems. Existing systems at various administrative levels (Centre, State/Boards, Schools), are leveraged by providing additional workflows, and/or upgrading them as per DIKSHA architecture principles (by using standard protocols and specs) so that multiple interoperable platforms and apps are enabled and integrated with DIKSHA building blocks on the DIKSHA platform. This provides simpler, coherent, and unified experiences to all actors in the ecosystem.

8. Extensibility

Maintaining NDEAR's tenet of unbundled and combinable, the DIKSHA platform is designed for components to be minimalistic, independently replaceable and extensible. This allows different components to be loosely coupled while building solutions, thereby enabling solutioning diversity and continued development. Different aspects of the infrastructure are extendable and hence, enable the creation of contextual solutions.

9. Scalability

NDEAR recommends that all technology services and processes must be designed for operating at the intended scale (be it at National, State or local). Extreme automation, distributed human processes etc. are key aspects for overall system scale and manageability. In a federated and distributed system, scalability is not ensured by only using scalable products and services. For example, a central team cannot, irrespective of its size, curate content for all or address the different requirements of each State/Board. Such practices are contradictory to the principles of federation, autonomy and scale. Distributed processes ensure that the diverse needs of millions of DIKSHA users get addressed at scale.

Automate to Scale

Since the DIKSHA platform supports the usage and creation of many solutions at the scale of millions and billions of transactions, the operations of the system are automated as much as possible. Any operation (e.g., related to installation, deployment, testing, etc.) that is predictable, repetitive, requires long term sustainability and does not need human intervention, is automated.

Benchmarking for Scalability

The DIKSHA platform is being used across the country by several millions of users and multiple systems are integrated to the platform via APIs necessitating extreme scale and reliability of

the core platform. All major releases of the platform are subjected to both burst (testing for sudden spikes of TPS at 10-100x from a below normal TPS, sustained over 1-4 hours) and soak testing (testing for 2-5x of average load for a duration of 48-72 hrs without services degrading). To easily do such benchmarks in a privacy friendly and cost efficient manner for every major release, DIKSHA uses cloud infrastructure, automated benchmarking scripts/tools, and synthetic data.

10. Being Resilient

NDEAR recommends that resilience is built into systems and services. DIKSHA being a federated and distributed system, DIKSHA systems and services are built to withstand failures by building automated recoveries. The processes and the structures driving decision making are resilient to rapidly changing needs and have the flexibility and capability to re-adapt to handle disruptions.

11. Ensuring Openness

Maintaining NDEAR's tenets of interoperability and openness, DIKSHA harnesses capacity across ecosystems by building and reusing open source technologies, open source algorithms, open source libraries, open standards and opportunities, open data and also sharing everything built as open source.

Open Source

Leveraging Open Source: All services (except those that are notified as sensitive) built into DIKSHA's digital infrastructure are based on and built as open source, whenever possible, This ensures vendor neutrality, transparency, and strategic control of the core digital infrastructure. This is in compliance with MeitY Policy[4] on open source.

Open Standards, Open APIs, & Open Protocols

The DIKSHA platform is ecosystem-driven, enabling an environment of innovation, and collaboration that is responsive to the needs of all its actors (e.g., teachers, students, parents, administrators and community members). Standards and specifications adopted allow flexibility and choice to ecosystem actors while enabling effective coordination through a common vocabulary across various areas. Standards and specifications cover people, processes, data, interfaces and APIs. For DIKSHA:

- Open international and national standards have been adopted when available. The use of open standards reduces the barriers to participation by ecosystem actors and enables them to avoid vendor lock-ins.

[4] MeitY open source policy - https://www.meity.gov.in/writereaddata/files/policy_on_adoption_of_oss.pdf

- Provided/created open standards and specifications, when necessary, to enable portability and increased interoperability between solutions and systems.
- Open standards and specifications are regularly upgraded, new specifications are introduced as open source using experts from the community.
- All building blocks within the platform have implemented a set of Open APIs in compliance with MeitY Open API policy[5].
- Designed open protocols (a set of API specifications along with a set of policies that are implemented by the ecosystem) that allow ecosystem solutions, services, assets to be discoverable and interoperable, without centralizing everything. Examples of protocols used within India's digital infrastructure - eSign, Unified Payment Interface (UPI), Data Empowerment and Protection Architecture (DEPA) (used in Account Aggregator - AA, and Personal Health Records -PHR), Beckn Protocol used in recently launched Open Network for Digital Commerce (ONDC) and Unified Health Interface (UHI), etc.

Open Data

The Open Data Handbook[6] briefly defines Open data as *"data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and sharealike."* Open data is crucial to interoperability. The most important aspects of the definition are related to

- ***Availability and Access****: Data must be available as a whole, in a convenient and modifiable form and at no more than a reasonable reproduction cost and preferably downloadable over the internet.*
- ***Re-use and Redistribution:*** *Data must be provided under terms that permit re-use and redistribution including the intermixing with other datasets.*
- ***Universal Participation:*** *There should be no discrimination against fields of endeavour, or against persons or groups to use, re-use and redistribute data. For example, there cannot be 'non-commercial' restrictions to prevent 'commercial' use, or restrictions for use only for certain purposes (e.g. only in education).*

DIKSHA fully adheres to the NDEAR principle of open data and observability such that services are built to emit anonymised telemetry events to generate and aggregate open data by understanding system behaviour and supporting policy evolution. The open data is made available as 'public good' for access by all to enhance research, interventions, policies, solutions, and build an overall understanding of the effectiveness of the education system as per GoI's National Data Sharing and Accessibility Policy.[7]

[5] MeitY Open API Policy - https://apisetu.gov.in/document-central/api-policy/
[6] Open Data Handbook - https://opendatahandbook.org/guide/en/what-is-open-data/
[7] National Data Sharing and Accessibility Policy (NDSAP)

Use of Commodity Computing

The DIKSHA platform, used by the ecosystem to create varied solutions at scale, leverages diverse, fit-for-purpose commodity computing components. Such components are cheaper and readily available from multiple providers and help to prevent vendor lock-in, to ensure reliability at scale and integration based on open standards.

Heterogeneity

Maintaining the NDEAR principle of using emerging technologies, DIKSHA is designed to allow a mix and match of diverse open and commoditized technologies. As a heterogeneously built infrastructure, DIKSHA platform is not heavily dependent on a particular technology or a technology provider.

Cloud neutrality and multi-cloud deployment

The tenets of evolvability, openness and the DIKSHA platform need to keep pace with the changes in technology as they evolve over the years. It has an architecture that easily accommodates new capabilities that may be needed as the ecosystem evolves and incorporates new technologies as they emerge. This enables systems to address technology risk and evolve heterogeneously. It leverages various technology innovations as and when they become viable and useful to improve education. In this direction, the architecture of the DIKSHA platform is built for cloud neutrality and multi-cloud deployment.

- Cloud neutrality means that the platform is built agnostic of any specific cloud vendor. DIKSHA uses different clouds for different environments, (e.g. development, testing, production, etc.) and is not tied to any specific cloud.
- Multi-cloud deployment means that DIKSHA services are deployed across multiple clouds and are connected via secure APIs. This enables the DIKSHA platform to take advantage of the unique benefits of a multi-cloud approach such as running sensitive services on the Government of India (GoI) cloud and running services requiring heavy workloads on a GPU cloud. For example; large-scale analytics services using anonymized data running on a public cloud, Artificial Intelligence (AI) services running on a GPU cloud, and core data services running on a Government of India cloud.

12. Data & Observability

Data and built-in observability form the backbone of the DIKSHA platform. The platform implements event-driven telemetry and data aggregation services that power intelligent and data-driven continuous evolution of the platform and processes.

Such a data-driven design facilitates the implementation of various intelligent services such as content personalization based on learner persona, workflow monitoring, AI-driven recommendations, automated behavioural nudges, etc.

Transparency, accuracy and reliability of data and analytics are ensured through a registry-driven approach, whereby data is logically organized as a shared, single source of truth. For interoperability and unification of DIKSHA's federated architecture with decentralised systems and applications have master codes and decentralised electronic registries that are made available via common APIs. For example; DIKSHA's user and organization registries.

13. Interoperable

DIKSHA is designed for interoperability via open APIs and open standards to support federated design principles so that various solutions built by the ecosystem are able to work in a unified manner.

14. Portable

DIKSHA is designed for the portability of data, certificates, credentials, documents, content, etc. based on open standards. Portability is essential to ensure both data and assets are easily made available to various solutions and systems integrated into the platform. Open published specifications for all such assets and data are critical to ensure portability and reuse.

15. Generalized and Configurable

DIKSHA's core microservices are minimal (both data and functional minimalism), atomic, and generalized, allowing solution builders to 'reuse and extend' them to build contextual solutions. All services are designed in a 'generalized' manner to allow diverse use cases to be built on the same set of services. These generalized services 'externalize' their configurability and expose various configurations within the API itself to allow solution building across diverse contexts.

16. Designed for Solutioning

DIKSHA's federated architecture allows the platform to support an ecosystem-driven solutioning approach. It provides capabilities for actors in the ecosystem to create solutions, to learn, interact and collaborate. For example, depending on a State/Board requirement, the DIKSHA infrastructure enables rapid solutioning capabilities that allow actors to discover, understand, engage, experiment, innovate, and build on. DIKSHA has enabled asynchronous learning, self-evaluation and discussions that are critical to the core principles of learning such as readiness to learn, and provide regular practice for an effective learning experience.

17. Evolvable & Iterative

Evolvability

Evolvability is one of the core tenets advocated in NDEAR. The architecture and the building blocks of the digital infrastructure should not be thought of as a one-time exercise but rather as an evolving construct. Given the large and diverse ecosystem of actors and applications, all DIKSHA building blocks are clearly versioned, backward-compatible, and have well-defined and transparent version-retirement policies. This ensures that various applications/systems depending on them can adopt and upgrade asynchronously at their pace and evolve along.

The core principle of technology evolvability can be illustrated in the following manner:

- Neither should all capabilities be built on day one nor should it be assumed that capabilities built once may not change. Given this, the removal and addition of capabilities is a constant ongoing process of the platform design.
- Use strong platform governance principles to work with the ecosystem to evolve the platform while focusing on delivering better services to partners and users of the platform.
- Adaptability is the heart of evolution. The survival of the DIKSHA platform depends on its ability to rapidly adapt and evolve to changing requirements. Since the platform is built using building blocks, it is easily possible to exchange individual components without changing others.

Iterative

Evolvability also ensures that newer technologies, user interfaces, form factors, etc. can be easily brought in. The DIKSHA platform is architected to leverage various technology innovations (AI/ML, AR/VR) as and when they become viable and useful. When AI is leveraged, it is built to amplify human actors, eliminate biases, and assist in effective teaching/learning and management, rather than fully replacing humans. Translation of content in regional languages, speech recognition, personalisation, etc. is a critical part of this. A set of reusable AI services, open-source libraries, open-source models and data sets are built so that they can be leveraged and embedded within other building blocks.

Replaceability

Given the nature of an evolving platform architecture, it is assumed that all building blocks and micro-services are complete or fit for purpose from the start. DIKSHA architecture is amenable to refactoring and/or replacing certain services with newer versions. The architectural ability to continually refactor/replace is the essence of evolvability. DIKSHA's systems, processes, and tools are built to accommodate this.

Automation

Evolution means constant change. It necessitates testing the system and its building blocks. Hence, to support rapid evolution, the system is designed to ensure continuous testing of the DIKSHA building blocks and services through automation. For example, it is critical to ensure that a new code change does not interrupt the working of a microservice that is already in use in multiple solutions, perhaps for millions of transactions. It is almost impossible for people to test for all the possible scenarios. DIKSHA architecture adopts extreme automation to guarantee backward compatibility, asynchronous adoption, and seamless upgrade of ecosystem applications and services beyond what is provided by the platform.

Global Benchmark

Since the design of the DIKSHA platform has the ability to adapt and evolve with changing requirements, it is important to constantly compare and learn from different platforms and similar large-scale networks that are designed/orchestrated across the world. Changing needs and new practices are identified from such learnings. The DIKSHA platform is assessed against relevant global benchmarks (e.g., design best practices across platforms and large-scale networks) on an ongoing basis for continuous improvement.

18. Trust by Design

Adhering to NDEAR's tenet of privacy, security and trust by design, DIKSHA has separate principles for each of these aspects. The DIKSHA platform is designed to:

- guarantee the integrity of actors (platform users) and assets (e.g., content, data) on the platform and fairness of all interactions, to imbue trust in the platform.
- ensure that the risks associated with the actors (e.g., users), assets (e.g., data) and interactions on the DIKSHA platform are addressed.
- Provide capabilities to authenticate users, encrypt data and share regular disclosures on various policies. For example, the verified registry of all the entities shall lead to higher accountability and trust.

19. Privacy by Design

Given the highly confidential nature of user data on the DIKSHA platform and its implementation on the cloud, adequate attention is given to protect user data from any form of unauthorized access, use, distortion and publishing. The design takes into consideration all relevant norms regarding sensitive information and personal information of individuals and adheres to appropriate data protection and privacy laws.

Security of services and data is designed into all DIKSHA building blocks using strong security design principles. Security is baked into all aspects of DIKSHA's digital infrastructure. All solutions and services within DIKSHA adhere to appropriate data protection and privacy laws.

20. Security by Design

The DIKSHA platform is designed such that the security of services and data is maintained using strong security design principles. The DIKSHA platform complies with all relevant GoI security norms and ensures the protection of platform assets, data, and any other information/assets that need to be protected. All aspects of data collection, transmission, processing, storage, and access are secured through encryption, access control, auditing, notification, and other best practices.

Building Blocks

Overview

Building block-driven design enables adaptability and configuration of solutions of various kinds. Building blocks have to interoperate with other building blocks within the same system or across systems.

A paper published by the Center for Global Development, **Fast-Tracking Development: A Building Blocks Approach for Digital Public Goods**[8], provides a comprehensive view of the building block approach. Quoting from the paper:

> *"A building block is anything that can be used as a part or component to create something larger or more complex. In the physical world, building blocks are everywhere. They are characterized by three distinguishing features, as illustrated in the example of house construction:*
>
> 1. ***They enable us to do something specific in diverse use cases.*** *Bricks, roof tiles, paint, doors, window frames—each is a type of building block used for a particular purpose, but they can be used in a wide variety of construction projects, from housing to office buildings.*
> 2. ***They can be combined for multiple purposes.*** *Multiple bricks are stacked to create walls of different types and shapes of houses, each plastered and painted as desired by the owner. The same bricks are also used to build boundary walls and other structures.*

[8] Fast-Tracking Development: A Building Blocks Approach for Digital Public Goods - https://www.cgdev.org/publication/fast-tracking-development-building-blocks-approach-digital-public-goods

3. ***They compound in value when they are connected or combined for downstream solutions.*** *The value of a house to the owners is much more than the sum of the bricks, cement, plaster, paint, etc.*

Characteristics of Building Blocks

Digital building blocks have four important characteristics:

- ***Autonomous****: Digital building blocks provide a standalone, reusable service or set of services. For example; GPS provides location coordinates autonomously; it solves the question "Where are you?"*
- ***Generic capabilities:*** *Digital building blocks are flexible across use cases. For example; GPS location services enable access to a range of digital products like maps, food delivery, taxi aggregators, and even fraud detection.*
- ***Interoperability:*** *Digital building blocks combine, connect, and interact with other building blocks through specified communication protocols and interfaces. Application programming interfaces, or APIs, are software intermediaries that allow digital building blocks to talk to each other through published communication protocols. The communication protocols enable a digital exchange of information to use and access services. Interoperability also allows integration with legacy systems in countries and organizations, which is essential for solutions that need to work at scale. For example, a GPS kit can be fitted into a nationwide fleet of delivery trucks.*
- ***Evolvability:*** *Digital building blocks are not immune to changes in technology and context. They can be improved even while being used as part of solutions. The camera software can improve during the life of a smartphone without any significant disruption to the user.*

Physical and Digital Building Blocks

There are three ways in which digital building blocks function differently than physical building blocks. First is the ***digital exchange of information*** *that is needed to enable a transaction. For example; withdrawing cash from an ATM is a physical process but it relies on several digital building blocks whereas paying for goods and services in cash does not.*

The second difference is in the ***marginal cost structure****. The resources needed for physical blocks increase as the number of copies increases (it costs more to make a million bricks than a hundred). In contrast, the costs of duplicating digital blocks (such as protocols, registries, software, applications, etc.) or achieving economies of scope (expanding the uses of a building block) increase only marginally. For example, as in the case of GPS, the digital building block can*

be converted into a public good and access provided to everyone at little or even no cost with minimal resources needed for maintenance and operation of the building block itself.

*The third, related, difference is that **a digital building block is multi-usable** while a physical block can only be used in one application at one time. This makes it easier for the digital building block to serve as a public good, as we explore in the next section, reinforcing its non-excludable and non-rivalrous properties.*

Just as physical building blocks such as bricks and pipes come in various shapes, sizes, and forms, so too do digital building blocks.

Types of Building Blocks

Digital building blocks are classified into four types:

1) standard specifications and protocols,

2) software code, data sets and models

3) platforms, and

4) applications

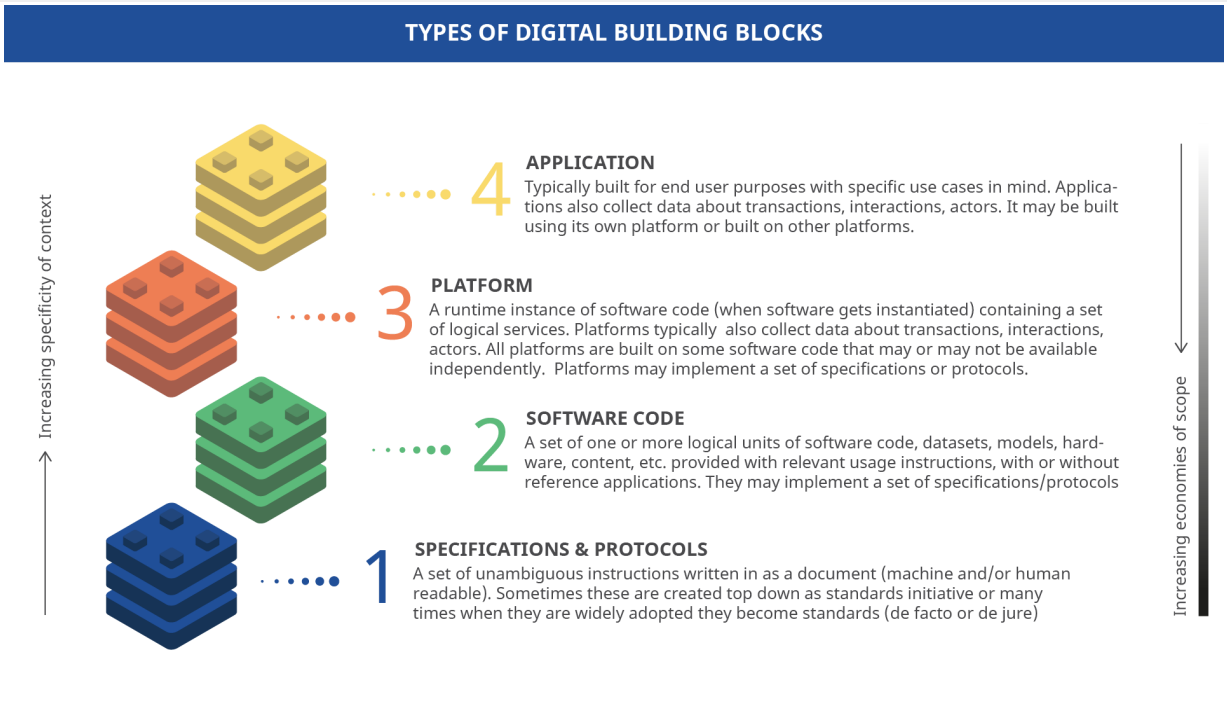

Figure 3: *Types of Digital Building Blocks*

This classification can be overlapping, and the distinctions need not be rigid. In practice, a building block is easier to identify by its purpose, scope, and use than by its form. Unless prevented by a specific barrier in its design or architecture, a digital building block allows for the freedom to use it together with other building blocks, standards and communication protocols to create more complex, multi-functional solutions. Digital public infrastructure are typically applications or platforms which offer a specific, ready-to-use service, while digital public goods are typically specifications, protocols and software code, datasets and models which are used to build a platform and/or an application.

DIKSHA infrastructure, which includes the various DIKSHA applications and services is Digital Public Infrastructure. The underlying software and specifications that power DIKSHA, open content hosted on DIKSHA and the public data are Digital Public Goods. DIKSHA is powered by the open-source Sunbird which is a Digital Public Good.

It is important to note that applications and platforms, having a runtime, (items 3 and 4 in Fig.3) collect data on users and usage. The collection, storage and use of this data require clear policies, guidelines and frameworks to protect personal information. Along with technical standards, it is important to have strong legal and regulatory frameworks governing the use of data to improve

development outcomes as an essential complement to the ultimate goal of building digital public infrastructure at scale."

Building Block Approach in NDEAR

The Ministry of Education (MoE), Government of India (GoI) proposed the National Digital Education Architecture (NDEAR) 2021[9]. The principles presented therein provide the technological framework to energize and catalyze the digital education ecosystem to create and deliver diverse, relevant, contextual, and innovative solutions that benefit students, parents, teachers and educational communities. NDEAR aims to enable existing systems to upgrade and become interoperable while making available building blocks to create new tools and solutions to meet the educational requirements of India.

The NDEAR report defines building blocks as "a package of self-contained functionalities defined to meet business needs through a set of services made available via APIs and optionally via reference solutions. Building blocks have to interoperate with other building blocks within the same system or across systems."

NDEAR provides an approach to establish a Federated Architecture (FA) that is defined in terms of building blocks. It picks out the common requirements of the education ecosystem and identifies key building blocks that suit the purpose.

The FA model is ideal for India's federal governance setup and includes both public and private institutions. FA is a pattern in enterprise architecture that allows interoperability and information sharing between semi-autonomous, de-centrally organized entities, information technology systems and applications.

Note: *For details of the FA approach suggested in NDEAR, refer to the section,* ***Federated Architecture and Building Blocks*** *in the* ***NDEAR Main Report****.*

Building blocks are the essence of NDEAR. They are classified as:

Building Block type	Characteristics
Core	• Built and managed as public goods • Must adhere to the architectural principles of NDEAR • Created and maintained at the National/State/ Board Level (in a federated structure) • Not built by other ecosystem players • Offered as hosted services/applications at appropriate

[9] NDEAR Main report - https://www.ndear.gov.in/resources.html

Building Block type	Characteristics
	administrative levels • Enable interoperability and act as a glue between the rest of the building blocks and the solutions built on top. For example, electronic registries, identities, etc.
Common	• Built and offered as a choice to all in the NDEAR ecosystem. • Must also adhere to the architectural principles of NDEAR to ensure interoperability and solutions • May also be created and maintained either at the Centre, State or Board levels. • Also offered as hosted services or applications at the appropriate administrative level, but they are optional (unlike Core building blocks) • The ecosystem may build alternative or enhanced versions of these building blocks to provide further choice • The source code should be made available as a reference for other ecosystem players to rapidly build diverse applications on top
Reference	• Built and offered only as “source code/data” to enable various ecosystem players (Government or private entities) to rapidly build their services/ applications. • To be seen as “accelerators” • Not offered as hosted services or applications. • Available for download, use, enhancement, and customization through the NDEAR Portal • The source code should also be made available as a reference to enable others to rapidly build diverse applications on top • Should be well architected and documented for better reuse

After detailed discussions with all stakeholders in the Indian education ecosystem and a careful study of existing systems, NDEAR has identified 36 minimum viable building blocks across 12 key building block categories. The following is a diagrammatic representation of the same:

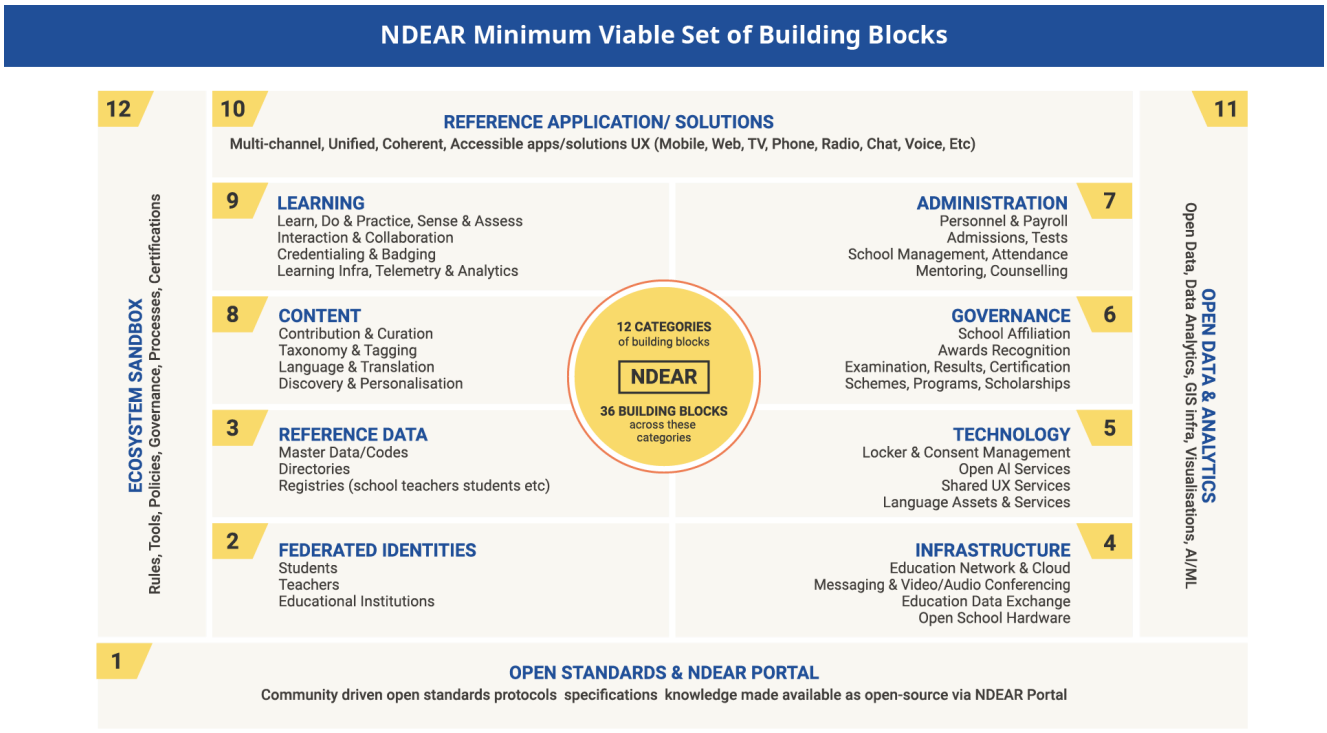

Figure 4: NDEAR Minimum Viable Set of Building Blocks

Note: *For details on the NDEAR Building Blocks, refer to the section* ***Building Blocks of NDEAR*** *in the* ***NDEAR Main Report***[10].

NDEAR & DIKSHA

DIKSHA makes use of the NDEAR building block construct. The following diagram shows the building block categories that DIKSHA uses from NDEAR's minimum viable set of building blocks.

[10] NDEAR Main report - https://www.ndear.gov.in/resources.html

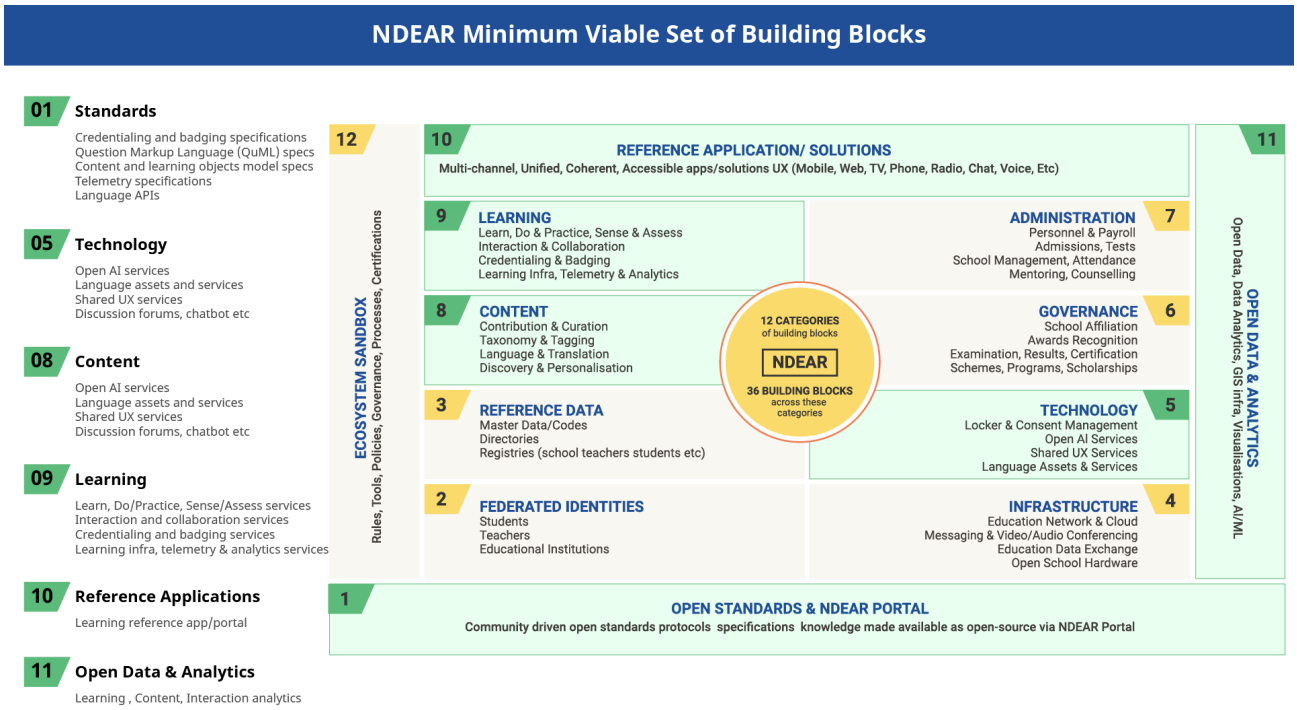

Figure 5: NDEAR- DIKSHA Building Block Category Mapping

DIKSHA is uses the following NDEAR building block categories:

Open Standards

DIKSHA uses open specifications and APIs for the following purposes:

- **Credentialing and badging specifications** - to provide certificates, etc.
- **Question Markup Language (QuML) specifications** - to create questions and question sets for practice and assessments
- **Content and learning objects model specifications** - to create and consume learning content
- **Telemetry specifications** - for built-in event-driven telemetry capabilities
- **Language APIs** - for content in multiple languages

Technology

DIKSHA intends to make use of the following building blocks from NDEAR's Technology building block category:

- **Open AI services** - for English and Indian languages
- **Language assets and services** - to build common language assets across India's diverse language base. For example, wordnets, generic 'bhasha' APIs, etc.

- **Shared UX services** - for content players, editors, events, discussion forums, chatbot, etc.

Content

DIKSHA uses all the building blocks recommended in NDEAR's Content building block category for the following purpose:

- **Contribution & curation** - for creating, curating and crowdsourcing content and other resource assets
- **Taxonomy and tagging services** - to add metadata for all DIKSHA assets
- **Language and translation services** - to ensure that DIKSHA resources are available in multiple languages
- **Discovery and personalization** - to aid search and findability of assets as per personal preferences

Learning

DIKSHA has services for NDEAR's Learning building block category for the following purpose:

- **Learn, Do/Practice, Sense/Assess services** - to enable different DIKSHA solutions for learning, help learn and administrate learning
- **Interaction and collaboration services** - to enable groups, discussion forums, etc.
- **Credentialing and badging services** - to create, mange, distribute and publish certificates and other credentials
- **Learning infra, telemetry, and analytics services** - to create and manage DIKSHA's learning infrastructure, telemetry events and reports

Reference Applications / Solutions

Using NDEAR's reference app building block construct, DIKSHA has learning reference apps and portals that are available in online and offline mode and on the mobile.

Open Data & Analytics

DIKSHA adheres to the NDEAR Open Data & Analytics building block category construct for the following:

- Learning
- Content
- Interaction analytics

DIKSHA & Sunbird

The DIKSHA platform is built using underlying Sunbird digital blocks that are atomic, independent and reusable.

DIKSHA uses multiple Sunbird building blocks as depicted in the following diagram and described in the table that follows. All building blocks are available as open-source software under MIT license and are part of Sunbird, a “Made in India, Made for the World” digital public good (DPG). All building block assets (source code, documentation, scripts, etc) are available on Github and are actively maintained by the Sunbird open-source community. Sunbird, Sunbird ED, and a few other building blocks are globally recognized by the international Digital Public Goods Alliance.

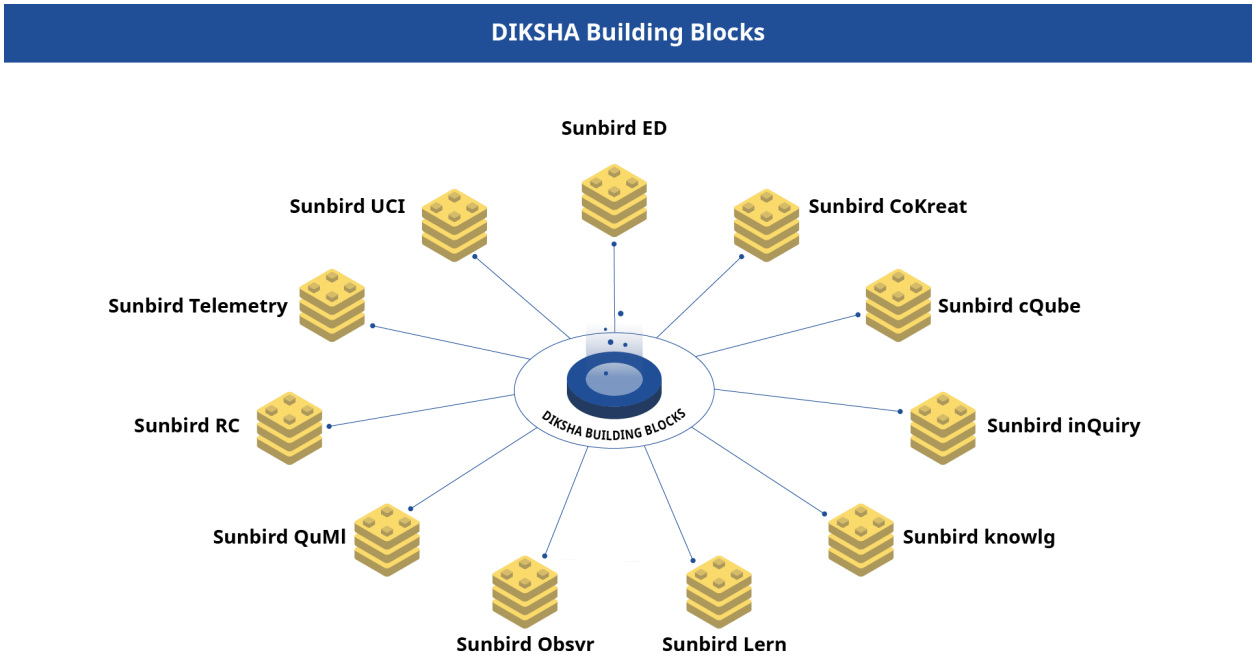

Figure 6: DIKSHA Building Blocks

Building Block Name	Description
Sunbird ED	Sunbird ED (or Sunbird for Education) is a reference solution developed for school education using reusable building blocks that enable key interactions of learning, capacity building, and administration. It is software developed to enable learning, capacity building, or professional development solutions.
Sunbird coKreat	Sunbird coKreat consists of a set of reusable tools and services to source various types of digital learning assets. It consists of a set of services and ready-to-use reference application that enables you to create, and source digital assets, curate them for specific needs and publish them for consumption. The web application comes with an inbuilt set of workflows for orchestrating the entire process. The unique power that Sunbird coKreat unleashes is enabling of “co-creation” using crowdsourcing as against the mere “creation” of assets. What this means is that you need not create all the digital assets that you require, within your organization. You can source them from other organizations and individuals who can provide it for you. This potentially makes the process highly scalable allowing you to access capacity and competency beyond your select set of users.
Sunbird inQuiry	Sunbird inQuiry is a building block that enables the setting up of question banks that can contain questions and question sets for various use cases such as practice, assessment, quiz, worksheet, survey, observations and many more. This building block consists of tools(editor and player) and services that enable the creation of question(s) and question set(s), configuration of their behaviour, curation and publishing them for consumption, playing of question set(s) and emission of meaningful data. inQuiry works with primarily two objects - Question and Question Set - and uses the interoperable QuML Specification. inQuiry can be extended for adaptive tests, gamified quizzes, rubric-driven assessments, competitions, data collection and other use-cases. It supports configurable behaviour of various kinds such as timer, certificate, hints, attempts, etc.
Sunbird Knowlg	Sunbird Knowlg module allows a powerful content and knowledge management capability for adopters to manage millions of content pieces…

Building Block Name	Description
	Sunbird Knowlg has powerful asset and knowledge management capabilities that can be used to create, curate, approve, publish, organize and manage large numbers of assets, e.g. millions of documents, several hundred hours of videos, thousands of interactive content, quizzes etc. Assets created using Sunbird Knowlg are tagged using a multi-dimension tagging framework called Taxonomy. This enriches the metadata and aids in content discovery. Assets are served to users using Search and Discovery Service. The Search and Discovery take users' preferences and usage attributes like ratings into account while serving the user. Knowlg also has components (players) to consume content.
Sunbird Lern	Sunbird Lern consists of the core services to manage any user's learning journey. It enables micro learning loops to be enabled along with other building blocks for interaction and collaboration. Sunbird Lern is a building block comprising core services that enable tracking of a user's learning journey as well as enables user interactions and collaborations by providing various sets of tools and micro services. These capabilities can be leveraged by any adopter with requirements for measurement of learning progress, creation and management of cohorts/ batches of learners, as well as enabling collaboration capabilities such as Groups or Discussion forums. The Lern services are also employed for creation and management of user accounts and organizations within the Sunbird instance. This building block consists of several components, each of which provides specific functionalities. Key functional modules among these include: ● User authentication and account creation ● User Org Service: C-R-U-D operations for management of entities such as the user, Organization as well as Location - within the Sunbird system. ● Role management: User rights & system roles management ● Batch management and Progress Tracking ● Groups and Discussion Forums for collaboration ● Notifications for users
Sunbird Observ	Sunbird Obsrv brings together several components that come together to enable the ability to measure and observe various actions and activities.

Building Block Name	Description
	Obsrv as a Sunbird building block comprises several components that come together to enable the ability to measure and observe various actions and activities carried out by the system/ users. Obsrv can be utilised as an independent building block that allows for measurement, or as part of a system that employs different Sunbird blocks to enable various workflows. It offers powerful data processing and aggregation infrastructure to process telemetry data generated by users & systems, validate the data, as well as aggregate and generate insights. It also has built-in open data cataloging and publishing capability. It is built keeping extensibility in mind, so that adopters have the flexibility to adapt the telemetry and tools to their specific use-cases.
Sunbird QuML	Sunbird QuML (Question Markup Language) is a specification for storage, rendering and distribution of questions and tests. It allows assessment materials to be authored and delivered on multiple systems interchangeably. Assessment has always played an important role in education. Most, if not all, types of formal education use some sort of assessment, typically including a final exam to earn a grade, a degree, a license, or some other form of qualification. Today, assessment is no longer restricted to grading at the end of an instruction (summative assessment), but it has been recognized that assessment is also useful for continuous monitoring & feedback (formative assessment) and guiding of the learning progress (means to learn), without being necessarily used for grading purposes.
Sunbird RC	Sunbird RC (Registry and Credential) is an open source software to rapidly build and deploy next-generation electronic registries and verifiable credentials including attestation and verification flows. Sunbird RC contains a set of frameworks to enable you to rapidly build next generation electronic registries and verifiable credentials including attestation and verification flows. This is an independently usable open source framework. This is part of the overall Sunbird open source initiative.
Sunbird Telemetry	Sunbird telemetry building block includes a set of open specifications and code to manage emission, collection, synchronization, validation, and processing of telemetry event stream. An open specification for recording and measuring statistical data from real-world use of digital apps & platforms.

Building Block Name	Description
Sunbird UCI	Sunbird UCI (Unified Communications Interface) is a system that empowers governments to create and manage conversations with citizens and with its own officials. Through UCI governments can seamlessly set-up simple and complex conversations using a multi-channel approach. UCI aims to democratize the use of different communication channels such as WhatsApp, Telegram, SMS, email for governance use cases through a standard configurable manner that is reusable and scalable across all governance use cases.
Sunbird Saral	Sunbird Saral is a handwriting recognition solution for physical assessment score-sheets (pen-and-paper tests). Saral can be viewed as an Optical Character Recognition (OCR) -plus application that is capable of doing OCR and can also understand the structure of the physical input. Saral data android application enables teachers to scan and upload the marks secured by the student in a given examination.
Sunbird cQube	Sunbird cQube is an analytical tool which helps integrating, analyzing and reporting the different data sources in predefined formats and provide functionality to extend the solution for other data needs as well. cQube processes the education data typically available in Indian States. It generates predefined metrics and creates visualizations of these metrics on a regular basis. Data ingestion to cQube happens through API. These metrics can be further drilled down to different geographical levels based on the need (e.g., State, District, Block, Cluster and School). Metrics are generated and visualizations are available for the generated metrics. Logged in users can view the visualization on the browser over the internet and metrics can also be downloaded for taking action or for further analysis. cQube provides flexibility to extend to multiple use cases by using the base code and adding the specific use case as per the need.

Open Source Software or Specifications

The GoI has policies like the MeitY Open Source Policy[11] and MeitY policy for open API.[12] that encourage and govern the use of open source software and specifications.
In addition to the building blocks mentioned in the preceding section, DIKSHA also uses the following infrastructure and development open-source software:

Open Source Used	Website	License	Link to License
Ansible	https://github.com/ansible/ansible	GPL3	https://github.com/ansible/ansible/blob/devel/COPYING
OpenVPN	https://openvpn.net/	2 Commercial licenses for 10 users each	
Terraform	https://github.com/hashicorp/terraform	Mozilla Public License, version 2.0	https://github.com/hashicorp/terraform/blob/master/LICENSE
Terrahelp	https://github.com/opencredo/terrahelp	Apache License 2.0	https://github.com/opencredo/terrahelp/blob/master/LICENSE
Monit	https://mmonit.com/monit/dist/binary/	GENERAL PUBLIC LICENSE	
Nexus	http://www.sonatype.org/nexus/archived/	FREE PLUG-IN LICENSE, For Nexus OSS 2.x	
Go	https://golang.org/project/	BSD Style	https://golang.org/LICENSE

[11] MeitY open source policy - https://www.meity.gov.in/writereaddata/files/policy_on_adoption_of_oss.pdf
[12] https://apisetu.gov.in/document-central/api-policy/

Open Source Used	Website	License	Link to License
Kibana	https://www.elastic.co/products/kibana	Apache 2.0	https://github.com/elastic/kibana/blob/master/LICENSE.md
Elastic Search	https://www.elastic.co/products/elasticsearch	Apache 2.0	https://github.com/elastic/elasticsearch/blob/master/LICENSE.txt
Logstash	https://www.elastic.co/products/logstash	Apache 2.0	https://github.com/elastic/logstash/blob/master/LICENSE
Kafka	http://kafka.apache.org/	Apache 2.0	https://github.com/kafka-dev/kafka/blob/master/LICENSE
Samza	http://samza.apache.org/	Apache 2.0	https://github.com/apache/samza/blob/master/LICENSE
Jenkins	https://jenkins-ci.org/	MIT	https://github.com/jenkinsci/jenkins/blob/master/LICENSE.txt
Ruby	https://www.ruby-lang.org	2-clause BSDL	https://www.ruby-lang.org/en/about/license.txt
Ansible	https://www.ansible.com/	GPL v3	https://github.com/ansible/ansible/blob/devel/COPYING
Docker	https://www.docker.com/	Apache 2.0	https://github.com/docker/docker/blob/master/LICENSE

Open Source Used	Website	License	Link to License
Java	http://openjdk.java.net/	GNU GPL + Linking Exception	http://openjdk.java.net/legal/gplv2+ce.html
Hadoop	https://hadoop.apache.org/	Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0
Spinach	https://github.com/codegram/spinach	MIT	https://github.com/codegram/spinach/blob/master/LICENSE
Nginx	https://www.nginx.com/	BSD-like	http://nginx.org/LICENSE
Sensu	https://github.com/sensu/sensu https://github.com/sensu-plugins	MIT	https://raw.githubusercontent.com/sensu/sensu/master/MIT-LICENSE.txt
Rabbitmq	https://github.com/rabbitmq/rabbitmq-server	MOZILLA PUBLIC LICENSE	https://github.com/rabbitmq/rabbitmq-server/blob/master/LICENSE-MPL-RabbitMQ
InfluxDb	https://github.com/influxdata/influxdb/	MIT	https://github.com/influxdata/influxdb/blob/master/LICENSE
Redis	https://github.com/antirez/redis	BSD license	https://github.com/antirez/redis/blob/unstable/COPYING
Uchiwa	https://github.com/sensu/uchiwa	MIT	https://github.com/sensu/uchiwa/blob/master/LICENSE

Open Source Used	Website	License	Link to License
Grafana	https://github.com/grafana/grafana	Apache 2.0	https://github.com/grafana/grafana/blob/master/LICENSE.md
Burrow	https://github.com/linkedin/Burrow	Apache 2.0	https://github.com/linkedin/Burrow#license
jolokia	https://github.com/rhuss/jolokia	Apache 2.0	https://github.com/rhuss/jolokia/blob/master/LICENSE
kong	https://github.com/Mashape/kong	Apache 2.0	https://github.com/Mashape/kong/blob/master/LICENSE
Postgres	https://www.postgresql.org/	The PostgreSQL Licence	https://www.postgresql.org/about/licence/
aws-client	https://aws.amazon.com/cli/	Apache License, Version 2.0	
wiremock	http://wiremock.org/	Apache License, Version 2.0	
elasticsearch rest client	https://mvnrepository.com/artifact/org.elasticsearch.client	Apache License, Version 2.0	
json schema validator	https://mvnrepository.com/artifact/com.github.java-json-tools/json-schema-validato	Apache License, Version 2.0	
Neo4J Enterprise Edition	https://neo4j.com/open-source/	AGPL	https://neo4j.com/open-source/
Apache Cassandra	http://cassandra.apache.org/	Apache License v2.0	

Open Source Used	Website	License	Link to License
MongoDB		GNU AGPL v3.0	https://www.mongodb.com/community/licensing
Redis	https://redis.io	BSD license	https://redis.io/topics/license
Node.js	https://nodejs.org/en/	MIT License	https://raw.githubusercontent.com/nodejs/node/master/LICENSE
AngularJS		MIT License	
jQuery		MIT License	
Akka		Apache License v2.0	
Pinterest Secor	https://github.com/pinterest/secor	Apache License v2.0	
Apache Spark	http://spark.apache.org/	Apache License v2.0	
Play Framework	https://www.playframework.com/	Apache License v2.0	
Cassandra Snapshotter	https://github.com/tbarbugli/cassandra_snapshotter	BSD License	https://github.com/tbarbugli/cassandra_snapshotter/blob/master/LICENSE.txt
Apache Tomcat		Apache License v2.0	
Spring Framework		Apache License v2.0	
Apache Commons		Apache License v2.0	

Open Source Used	Website	License	Link to License
Flyberry Capital	https://github.com/flyberry-capital/scala-slack		
lift-json	https://github.com/lift/lift/tree/master/framework/lift-base/lift-json		
Google Vision API	https://cloud.google.com/vision/	https://cloud.google.com/terms/	https://cloud.google.com/terms/
FFmpeg	https://ffmpeg.org/	GNU LGPL 2.1	https://www.ffmpeg.org/legal.html
ImageMagick	https://www.imagemagick.org/script/index.php	https://www.imagemagick.org/script/license.php	https://www.imagemagick.org/script/license.php
im4java	http://im4java.sourceforge.net/	LGPL	
ImgScalr	https://mvnrepository.com/artifact/org.imgscalr/imgscalr-lib/4.2	Apache 2.0	
Google APIs		Apache License v2.0	
Hibernate ORM Framework		GNU LGPL 2.1	
Docker		Apache License v2.0	
Maven		Apache License v2.0	
modelmapper		Apache License, Version 2.0	

Open Source Used	Website	License	Link to License
jackson-mapper-asl		Apache License, Version 2.0	
groovy-all		Apache License, Version 2.0	
guava		Apache License, Version 2.0	
jcsv		Apache License, Version 2.0	
gson		Apache License, Version 2.0	
joda-time		Apache License, Version 2.0	
querylang		Apache License, Version 2.0	
xercesImpl		Apache License, Version 2.0	
easymock		Apache License, Version 2.0	
json-path		Apache License, Version 2.0	
mongo-java-driver		Apache License, Version 2.0	
stax-api		Apache License, Version 2.0	
xpp3		Apache License, Version 2.0	
xsdlib		Apache License, Version 2.0	

Open Source Used	Website	License	Link to License
jaxme-api		Apache License, Version 2.0	
dao-hibernate		Apache License, Version 2.0	
jasypt-hibernate4		Apache License, Version 2.0	
jasypt-spring31		Apache License, Version 2.0	
akka-actor_2.10		Apache License, Version 2.0	
akka-remote_2.10		Apache License, Version 2.0	
testng		Apache License, Version 2.0	
cglib-nodep		Apache License, Version 2.0	
poi		Apache License, Version 2.0	
disruptor		Apache License, Version 2.0	
aws-java-sdk		Apache License, Version 2.0	
hamcrest-all		BSD License	
jsr250-api		CDDL LIcense Version 1.0	
javax.servlet-api		CDDL LIcense Version 1.0	
jstl		CDDL LIcense Version 1.0	

Open Source Used	Website	License	Link to License
javax.mail		CDDL LIcense Version 1.0	
aspectjrt		Eclipse Public License - v 1.0	
aspectjweaver		Eclipse Public License - v 1.0	
junit		Eclipse Public License 1.0	
mysql-connector-java		GNU General Public License v3	
neo4j		GNU General Public License v3	
gimap		GNU General Public License v3	
sesame-rio-rdfxml		GNU LGPL	
pull-parser		Indiana University Extreme! Lab Software License	
mockito-core		MIT License	
jedis		MIT License	
slf4j-api		MIT License	
scribd		MIT License	
scribe		MIT License	
jsoup		MIT License	
relaxngDatatype		Open Source License	
jaxen		The Werken Company License	

Open Source Used	Website	License	Link to License
R	https://www.r-project.org/	GNU Affero General Public License	
R studio	https://www.rstudio.com	FOAS	
R packages	https://cran.r-project.org		https://cran.r-project.org/web/licenses/
Python	https://www.python.org/	Python Software Foundation License	https://www.python.org/download/releases/2.7/license/
gensim	https://radimrehurek.com/gensim/	GNU LGPLv2.1 license	
speech recognition	https://pypi.python.org/pypi/SpeechRecognition/	BSD license	
Jupyter Notebook	http://jupyter.org/	BSD license	
LibFM 1.4.2	http://www.libfm.org/	GPL v3 license	
Apache Zeppelin	https://zeppelin.apache.org/	Apache License, Version 2.0	
Plotly		MIT license	https://plot.ly/javascript/open-source-announ
Google cloud api	https://cloud.google.com/apis/docs/overview		https://cloud.google.com/terms/
tagme api	https://services.d4science.org/web/tagme/documentation		

Open Source Used	Website	License	Link to License
		Apache Software License, Version 2.0	
SQLite			https://www.sqlite.org/copyright.html
Junit 4.1.2		Eclipse Public License - v 1.0	
W3C HTML5 specifications			
Licensing will be required from Ekstep (or under whose name the game will be published)			http://unity3d.com/legal/eula
Crosswalk	https://crosswalk-project.org/documentation/about/faq.html	BSD License	
Sthetho	https://github.com/facebook/stetho/blob/master/LICENSE	BSD License	
xampp			http://www.apache.org/licenses/LICENSE-2.0
notepad++	https://notepad-plus-plus.org/	GNU general public license	http://www.gnu.org/copyleft/gpl.html
paint.net	http://www.getpaint.net/index.html	GNU general public license	http://www.getpaint.net/license.html
audacity	http://audacityteam.org/		http://creativecommons.org/licenses/by/3.0/legalcode

Open Source Used	Website	License	Link to License
Piwik.org Analytics	http://piwik.org	GNU general public license	https://en.wikipedia.org/wiki/GNU_General_Public_License
Joomla	http://joomla.org	GNU general public license v2	https://docs.joomla.org/Joomla_Licenses
PHP	https://php.net	PHP License v3.01	https://secure.php.net/license/
Apache HTTP Server	https://httpd.apache.org/	Apache Software License, Version 2.0	http://www.apache.org/licenses/
Morris Charts	http://morrisjs.github.io/morris.js/index.html	Licensed under the BSD-2-Clause License.	
Mustache JS	https://mustache.github.io/	MIT	https://github.com/janl/mustache.js/blob/master/LICENSE
PHPUnit	https://phpunit.de/	CC Attribution 3.0	https://phpunit.de/manual/current/en/appendixes.copyright.html
PhantomJS	http://phantomjs.org/	BSD	https://github.com/ariya/phantomjs/blob/master/LICENSE.BSD
Jasmine		MIT	https://github.com/jasmine/jasmine/blob/master/MIT.LICENSE
Sweet Alerts	http://t4t5.github.io/sweetalert/	MIT	https://github.com/t4t5/sweetalert/blob/master/LICENSE

Open Source Used	Website	License	Link to License
EasySocial	https://stackideas.com/easysocial	GNU general public license	
EasyDiscuss	https://stackideas.com/easydiscuss	GNU general public license	
JFB Connect	https://www.sourcecoast.com/joomla-facebook/	GNU general public license v2	http://www.gnu.org/licenses/gpl-2.0.html
Admin Tools Pro	https://www.akeebabackup.com/products/admin-tools.html	GNU General Public License version 3	http://www.gnu.org/licenses/gpl-3.0.en.html
EasyBlog	https://stackideas.com/easyblog	GNU general public license	
RS Tickets Pro	https://www.rsjoomla.com/joomla-extensions/joomla-help-desk.html	GNU General Public License version 3	http://www.gnu.org/licenses/gpl-3.0.en.html
T3 Framework for Joomla!	http://t3-framework.org	GNU general public license v2	http://www.gnu.org/licenses/gpl-2.0.html
Markdown Editor	https://simular.co/lyra/extensions/asikart-markdown.html	GNU general public license v2	http://www.gnu.org/licenses/gpl-2.0.html
CDN For Joomla	https://www.regularlabs.com/extensions/cdnforjoomla	GNU general public license v2	http://www.gnu.org/licenses/gpl-2.0.html
JLike	https://techjoomla.com/products/j-like	GNU general public license v2	http://www.gnu.org/licenses/gpl-2.0.html
com_subusers		GNU general public license v2	http://www.gnu.org/licenses/gpl-2.0.html

Open Source Used	Website	License	Link to License
com_emailbeautifier	https://techjoomla.com/products/email-beautifier	GNU general public license v2	http://www.gnu.org/licenses/gpl-2.0.html
JA Extension Manger	https://www.joomlart.com/joomla/extensions/ja-extensions-manager	GNU/GPL	
JCE editor	https://www.joomlacontenteditor.net/	GNU/GPL 2 or later	http://www.gnu.org/licenses/old-licenses/gpl-2.0.html
OSmap Free	https://www.joomlashack.com/joomla-extensions/osmap/	GNU/GPL	http://www.gnu.org/licenses/gpl.html
JA slideshow	https://www.joomlart.com/downloads/joomla-extensions/ja-slideshow-lite/	GNU/GPL	http://www.gnu.org/licenses/gpl.html
JA Social II	https://www.joomlart.com/joomla/templates/ja-social-ii	Copyrighted Commercial Software	https://www.joomlart.com/joomlart/licenses
jPanel	http://bygiro.com/en/extensions/modules/jpanel	GNU/GPL 2 or later	http://www.gnu.org/copyleft/gpl.html
All videos	http://www.joomlaworks.net/extensions/free/allvideos	GNU/GPL	http://www.gnu.org/copyleft/gpl.html
plg_system_deployt ools		GNU/GPL	http://www.gnu.org/licenses/gpl.html
Scroll To CK	https://www.joomlack.fr/en/joomla-extensions/scroll-to-ck	GNU/GPL	http://www.gnu.org/licenses/gpl.html

Open Source Used	Website	License	Link to License
System - LLFJ - Lazy Load for Joomla!	https://joomla-extensions.kubik-rubik.de/downloads/llfj-lazy-load-for-joomla	GNU/GPL	http://www.gnu.org/licenses/gpl.html
izimodal	http://izimodal.marcelodolce.com/	Apache License 2.0	https://github.com/dolce/iziModal/blob/master/LICENSE
InviteX	https://techjoomla.com/products/invitex	GNU/GPL	https://techjoomla.com/about-techjoomla/license
Options manager	https://db8.eu/options-manager	GPLv2 or later	
Codeception	https://codeception.com	MIT License	https://github.com/Codeception/Codeception/blob/2.4/LICENSE
Composer	https://getcomposer.org	MIT License	https://github.com/composer/composer/blob/master/LICENSE
jQuery rateYo plugin	http://rateyo.fundoocode.ninja	MIT License	https://github.com/prrashi/rateYo/blob/master/LICENSE
chart.js	http://www.chartjs.org/	MIT License	https://github.com/chartjs/Chart.js/blob/master/LICENSE.md
underscoreJs	http://underscorejs.org/	MIT License	http://www.opensource.org/licenses/mit-license.html

Open Source Used	Website	License	Link to License
xml2json		Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0
typeahead.js	https://github.com/twitter/typeahead.js/	MIT License	https://github.com/twitter/typeahead.js/blob/master/LICENSE
lodash	https://lodash.com/	MIT License	https://raw.githubusercontent.com/lodash/lodash/4.17.4/LICENSE
ajaxq	https://github.com/Foliotek/ajaxq	MIT License	https://github.com/Foliotek/AjaxQ/blob/master/LICENSE
x-editable jquery plugin	https://github.com/vitalets/x-editable	MIT License	https://github.com/vitalets/x-editable/blob/develop/LICENSE-MIT
tokenizer jquery plugin	https://github.com/zellerda/Tokenize	BSD 3-Clause "New" or "Revised" License	https://github.com/zellerda/Tokenize/blob/master/LICENSE
toaster jquery plugin	https://github.com/CodeSeven/toastr	MIT License	https://github.com/CodeSeven/toastrblob/master/LICENSE
Recordmp3js	https://github.com/Audior/Recordmp3js	MIT License	https://github.com/Audior/Recordmp3js/blob/master/LICENSE
moment.js	https://github.com/moment/moment/	MIT License	https://github.com/moment/moment/blob/develop/LICENSE

Open Source Used	Website	License	Link to License
php jwt	https://github.com/firebase/php-jwt	BSD 3	https://github.com/firebase/php-jwt/blob/master/LICENSE
jquery tree plugin	https://www.jstree.com	MIT License	https://github.com/vakata/jstree/blob/master/LICENSE-MIT
jquery ui	http://jqueryui.com	MIT License	
jQuery LoadingOverlay	https://github.com/gasparesganga/jquery-loading-overlay	MIT License	https://github.com/gasparesganga/jquery-loading-overlay/blob/master/LICENSE
cropperjs	https://github.com/fengyuanchen/cropperjs	MIT License	https://github.com/fengyuanchen/cropperjs/blob/master/LICENSE
bootstrap-tokenfield	https://github.com/sliptree/bootstrap-tokenfield	MIT License	https://github.com/sliptree/bootstrap-tokenfield/blob/master/LICENSE.md
bootstrap-multiselect	https://github.com/davidstutz/bootstrap-multiselect	Apache License, Version 2.0	https://github.com/davidstutz/bootstrap-multiselect/blob/master/LICENSE.md
bootstrap-slider	https://github.com/seiyria/bootstrap-slider	MIT License	https://github.com/seiyria/bootstrap-slider/blob/master/LICENSE.md

Open Source Used	Website	License	Link to License
grunt	https://gruntjs.com/	MIT	https://github.com/gruntjs/grunt/blob/master/LICENSE
gulp	https://gulpjs.com/	MIT	https://github.com/gulpjs/gulp/blob/master/LICENSE
karma	https://karma-runner.github.io/1.0/index.html	MIT	https://github.com/karma-runner/karma/blob/master/LICENSE
phantomjs	http://phantomjs.org/	BSD	https://github.com/ariya/phantomjs/blob/master/LICENSE.BSD
semantic-ui	https://semantic-ui.com/	MIT	https://github.com/Semantic-Org/Semantic-UI/blob/master/LICENSE.md
lodash	https://lodash.com/	MIT	https://raw.githubusercontent.com/lodash/lodash/4.17.4/LICENSE
xmlbuilder		MIT	http://spdx.org/licenses/MIT
fabric	http://fabricjs.com/	MIT	https://github.com/kangax/fabric.js/blob/master/README.md#mit-license
uuid		MIT	http://spdx.org/licenses/MIT

Open Source Used	Website	License	Link to License
async	https://www.npmjs.com/package/uuid-v4	MIT	https://github.com/caolan/async/blob/master/LICENSE
ng-dialog	https://github.com/likeastore/ngDialog	MIT	https://github.com/likeastore/ngDialog#license
oclazyload	https://github.com/ocombe/ocLazyLoad	MIT	https://github.com/ocombe/ocLazyLoad/blob/master/LICENSE.md
createjs	http://www.createjs.com/	MIT	http://www.opensource.org/licenses/mit-license.html
creatinejs	http://creatine.guineashots.com/	MIT	http://www.opensource.org/licenses/mit-license.html
xml2json		Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0
eventbus	http://greenrobot.org/eventbus/	Apache 2.0	http://www.apache.org/licenses/LICENSE-2.0
Simple JavaScript Inheritance(class)	https://johnresig.com/blog/simple-javascript-inheritance/	MIT	http://www.opensource.org/licenses/mit-license.html
jasmine	https://jasmine.github.io/	MIT	http://www.opensource.org/licenses/mit-license.html
angularjs	http://angularjs.org	MIT	http://www.opensource.org/licenses/mit-license.html

Open Source Used	Website	License	Link to License
ionic	http://ionicframework.com/	MIT	http://www.opensource.org/licenses/mit-license.html
ngcordova	http://ngcordova.com/	MIT	http://www.opensource.org/licenses/mit-license.html
jquery	https://jquery.com/	MIT	http://www.opensource.org/licenses/mit-license.html
underscoreJs	http://underscorejs.org/	MIT	http://www.opensource.org/licenses/mit-license.html
jquery easing	http://gsgd.co.uk/sandbox/jquery/easing/	BSD	
Crosswalk	https://crosswalk-project.org/	MIT	http://www.opensource.org/licenses/mit-license.html
jsdoc	https://github.com/jsdoc3/jsdoc	Apache 2.0	https://github.com/jsdoc3/jsdoc/blob/master/LICENSE.md
md5	https://code.google.com/archive/p/crypto-js/downloads	BSD	https://code.google.com/archive/p/crypto-js/wikis/License.wiki
audiosprite	https://github.com/tonistiigi/audiosprite	MIT	https://github.com/tonistiigi/audiosprite/blob/master/LICENSE

Open Source Used	Website	License	Link to License
toastr	http://www.jqueryscript.net/other/Highly-Customizable-jQuery-Toast-Message-Plugin-Toastr.html	MIT	http://www.jqueryscript.net/other/Highly-Customizable-jQuery-Toast-Message-Plugin-Toastr.html
jquery.bookshelfslider	http://www.editions-terre-de-lumiere.com/ps_1491/modules/blockbookshelf/_documentation/bookshelf_slider_help.html	BSD	
angular-ui-sortable	https://github.com/angular-ui/ui-sortable	MIT	https://github.com/angular-ui/ui-sortable/blob/master/LICENSE
videoJs	https://github.com/videojs	Apache 2.0	https://github.com/videojs/video.js/blob/master/LICENSE
EPUB	https://github.com/futurepress/epub.js	MIT	https://raw.githubusercontent.com/tildeio/rsvp.js/master/LICENSE
JSZIP	https://github.com/Stuk/jszip	MIT	https://github.com/Stuk/jszip/blob/master/LICENSE.markdown
PDFJS	https://github.com/mozilla/pdf.js	Apache 2.0	https://github.com/mozilla/pdf.js/blob/master/LICENSE
Fine Uploader	http://fineuploader.com	MIT	https://github.com/FineUploader/fine-uploader/blob/develop/LICENSE

Open Source Used	Website	License	Link to License
prismjs	http://prismjs.com/	MIT	https://github.com/PrismJS/prism/blob/gh-pages/LICENSE
CKEditor	https://ckeditor.com/	GPL	https://ckeditor.com/terms-of-use#licenses
jQuery.scrollTo	https://github.com/flesler/jquery.scrollTo	MIT	https://github.com/flesler/jquery.scrollTo/blob/master/LICENSE
angular-timeago	https://github.com/yaru22/angular-timeago	MIT	https://github.com/yaru22/angular-timeago/blob/master/LICENSE
micromarkdown	https://github.com/simonwaldherr/micromarkdown.js/	MIT	https://github.com/SimonWaldherr/micromarkdown.js/blob/master/LICENSE.md
ng-token-field	https://github.com/lucaslouca/ng-token-field	MIT	https://github.com/lucaslouca/ng-token-field/blob/master/LICENSE
jqueryui	http://jqueryui.com/	MIT	http://jqueryui.com/
Spectrum Colorpicker	https://github.com/bgrins/spectrum	MIT	https://github.com/bgrins/spectrum/blob/master/LICENSE
jquery fancytree	https://github.com/mar10/fancytree/	MIT	https://github.com/mar10/fancytree/wiki/LicenseInfo

Open Source Used	Website	License	Link to License
jPlayer	http://jplayer.org/	MIT	https://github.com/jplayer/jPlayer/blob/master/MIT-LICENSE.txt
wavesurfer	https://github.com/katspaugh/wavesurfer.js	CC-BY-3.0	https://github.com/katspaugh/wavesurfer.js/blob/master/LICENSE
Lamejs	https://github.com/zhuker/lamejs	LGPL	https://github.com/zhuker/lamejs/blob/master/LICENSE
Please-wait	https://github.com/Pathgather/please-wait	MIT	https://github.com/Pathgather/please-wait/blob/master/LICENSE
izitoast	http://izitoast.marcelodolce.com/	Apache 2.0	https://github.com/dolce/iziToast/blob/master/LICENSE
font-awesome	http://fontawesome.io/	GPL	http://fontawesome.io/license/
Ajv: Another JSON Schema Validator	https://github.com/epoberezkin/ajv	MIT	https://github.com/epoberezkin/ajv/blob/master/LICENSE
gulp-uglify	https://github.com/terinjokes/gulp-uglify	MIT	https://github.com/terinjokes/gulp-uglify/blob/master/LICENSE.md
Web Font Loader	https://github.com/typekit/webfontloader	Apache 2.0	https://github.com/typekit/webfontloader/blob/master/LICENSE

Open Source Used	Website	License	Link to License
Mousetrap	https://github.com/ccampbell/mousetrap	Apache 2.0	https://github.com/ccampbell/mousetrap/blob/master/LICENSE
dragula	https://github.com/bevacqua/dragula	MIT	https://github.com/bevacqua/dragula/blob/master/license
mathquill	https://github.com/mathquill/mathquill/	MPL 2.0	http://mozilla.org/MPL/2.0/
KaTeX	https://github.com/Khan/KaTeX	MIT	http://opensource.org/licenses/MIT
Android Support library	https://developer.android.com/topic/libraries/support-library/index.html	Apache 2.0	
ANdroid support CardView	https://developer.android.com/topic/libraries/support-library/packages.html	Apache 2.0	
Multidex	https://developer.android.com/studio/build/multidex.html	Apache 2.0	
Google Firebase Cloud Messaging	https://firebase.google.com/	Apache 2.0	
Google Firebase Analytics	https://firebase.google.com/	Apache 2.0	
Google Location service		Apache 2.0	
Json Scema validator	https://github.com/java-json-tools/json-schema-validator	Apache 2.0	

Open Source Used	Website	License	Link to License
Stetho	https://github.com/facebook/stetho	BSD 3-clause	
Stetho Okhttp	https://github.com/facebook/stetho	BSD 3-clause	
.Find bugs	https://github.com/findbugsproject/findbugs	Apache 2.0	
Espresso	https://developer.android.com/training/testing/espresso/index.html	Apache 2.0	
Support Annotation	https://developer.android.com/reference/android/support/annotation/package-summary.html	Apache 2.0	
Ulautomator	https://developer.android.com/training/testing/ui-automator.html	Apache 2.0	
Glide	https://github.com/bumptech/glide	BSD, part MIT and Apache 2.0	
Crashlytics	https://try.crashlytics.com/	Apache 2.0	
Espresso	https://google.github.io/android-testing-support-library/docs/espresso/	Apache 2.0	
okHttp	https://github.com/square/okhttp	Apache 2.0	

Open Source Used	Website	License	Link to License
joda-time	https://github.com/JodaOrg/joda-time	Apache 2.0	
Event bus	http://greenrobot.org/eventbus/	Apache 2.0	
Google play services		Apache 2.0	
Awaitility	https://github.com/awaitility/awaitility	Apache 2.0	
Gson	https://github.com/google/gson	Apache 2.0	

Part B

DIKSHA Operations

DIKSHA Operations Overview

Typically, DIKSHA operations can be categorized into six stages and are cyclical in nature. Each stage follows its own processes the details of which are elaborated in the sections that follow. The following is a high-level visual representation of the life cycle for DIKSHA operations.

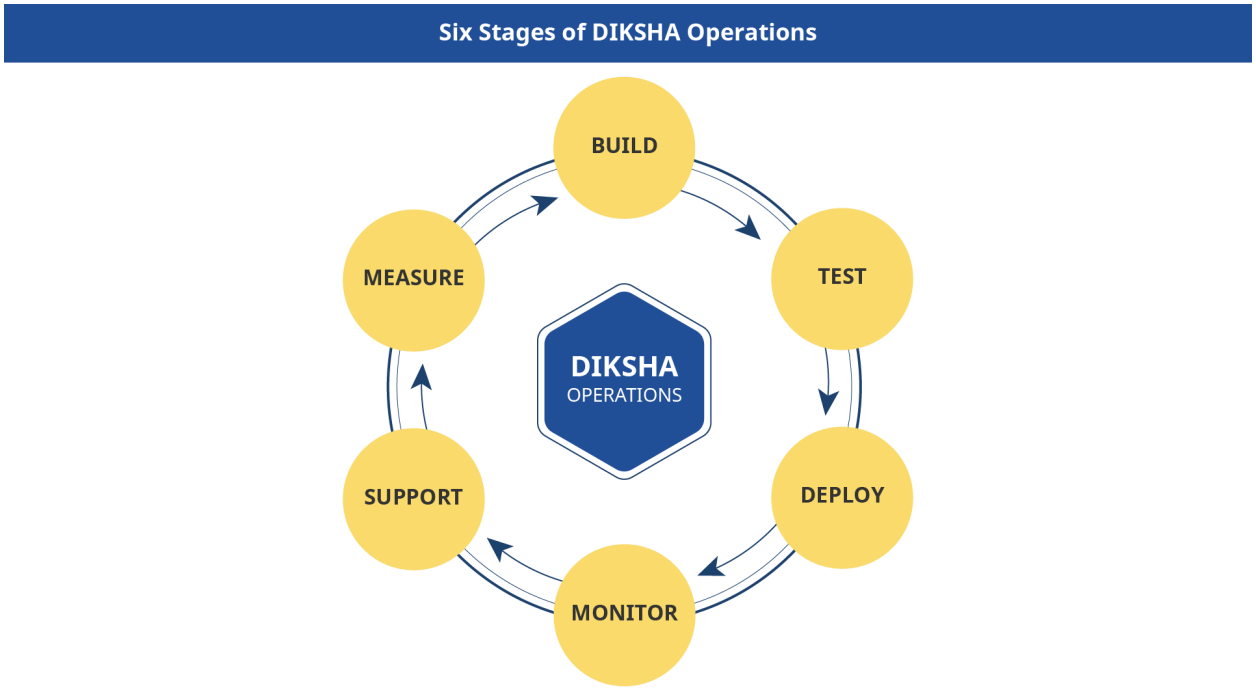

Figure 7: Six Stages of DIKSHA Operations

Release Cycle

DIKSHA uses Sunbird Ed as its underlying software, Hence the build and release cycle of DIKSHA is dependent on the Sunbird Ed build and release cycles. Sunbird Ed uses an agile framework to plan and execute its version releases. Each release cycle is typically executed over seven weeks. Most of the release deliverables are maintained in tools such as JIRA, Confluence and GitHub.

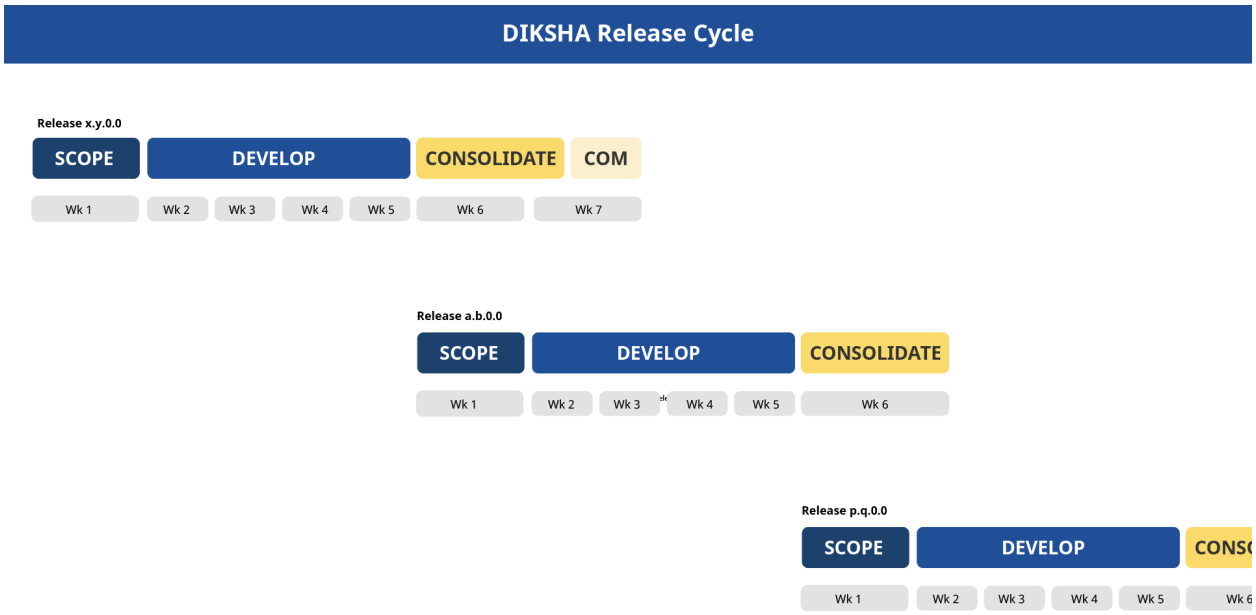

Figure 8: DIKSHA Release Cycle

Build

A typical Sunbird version release follows a four (4) stage build process. The four stages of the build process are:

- Scope
- Develop
- Consolidate
- Communicate

Each stage is elaborated in the sections that follow.

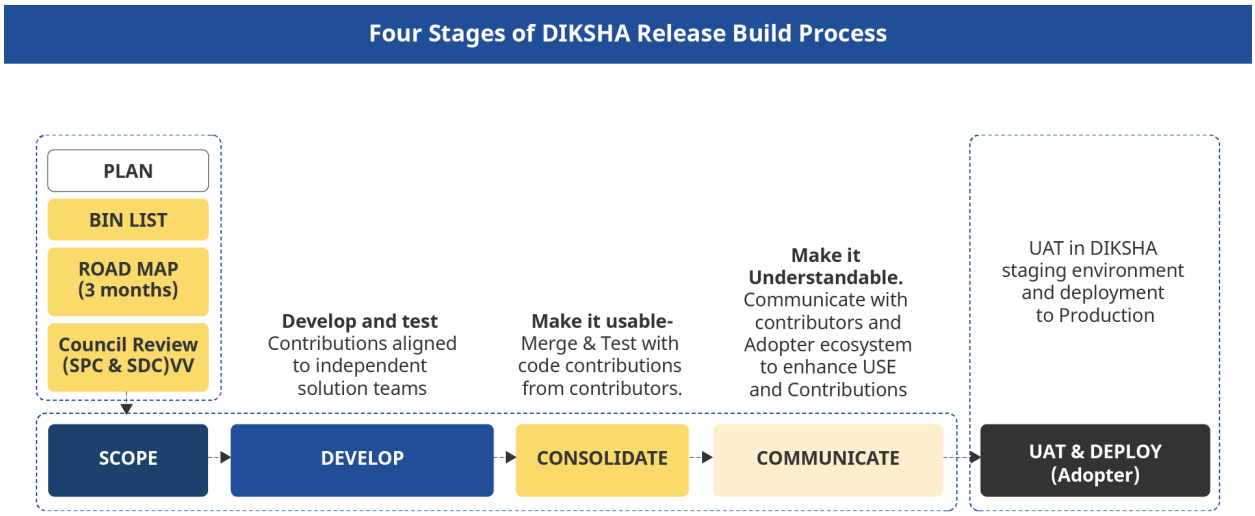

Figure 9: Four Stages of Sunbird Release Build Process

Scope

As part of release planning, the teams decide the scope of what will be delivered as part of a release. Items are picked based on the product roadmap from the bin list or backlog which is maintained in JIRA.

As part of the Scope phase, the team produces the following artifacts:

- Epics and Stories
- Delivery Plan

Epics and Stories

As per Agile methodology, the functional, non-functional and software requirement specifications are detailed as Epics and Stories.

Note: *For the epics and stories of Sunbird Ed, refer to* ***Appendix A - Functional and Nonfunctional Requirements.***

Telemetry Instrumentation

As part of the build process, the team decides the telemetry events that need to be instrumented for the functionality/feature that is being developed. At this stage, its important to understand 'what is telemetry?'
Telemetry is a term used for technologies that automatically record and measure statistical data from real-world use and forward it to IT systems in a remote location for further analysis and study. Telemetry is used in a myriad of industries from tracking spacecrafts, medical monitoring, tracking wildlife, and so on.
'Events' are broad, human-readable actions that can be tracked as a string. Events are used to categorize telemetry data.They are the basic unit for analytics and help identify user navigation or flow.
The concept of telemetry events is to identify:
Who did what, on what, and where, using what, in relation to what?
Every event has the following sections and corresponding fields to capture the data:

Section	Description	Attributes
About	About the event	ets mid
Who	About the actor	uid
did	Verb or action	eid
on what	Action on what object?	content_id content_ver
and where	Context of the action	env did sid channel pdata
using what	Using which tool?	?
In relation to	Related to which action?	cdata

The objective of telemetry is to assist in product, application or service development, modification or security. It works as a framework. Telemetry enables automatic collection of data from real-world, real-time use.

Typically, there are four levels of telemetry:

- Security
- Basic
- Enhanced
- Full

The level of data collected is a discrete decision of an organization or business. Analysis of this data offers insights into product and user behaviour and usage patterns, driving business decisions and research outcomes. You can program your telemetry analytics to suit your requirements.

DIKSHA's telemetry service has Full level telemetry.

Note:

1. *For detailed information on Telemetry specifications and events, refer to* ***Appendix C - Telemetry.***
2. *For detailed information on design of datasets and data exhausts, refer to* ***Appendix D - Design Note on Datasets and Data Exhausts.***

Delivery Plan

The scope is finalized based on inputs received from different adopters. On scope finalization, the team shares a Delivery Plan document.

Note: *For delivery plan, refer to* ***Appendix B - Delivery Plan Status Dashboard.***

Develop

The Develop stage of the release cycle is the longest and most crucial phase of the Build process. As part of this phase, teams design and develop the solution as per the defined scope, and test it in the development environment. In this phase the team produces multiple deliverables like:

- Code
- Release Brief

The following subsections describe each deliverable briefly.

Coding Repositories

Teams develop software to ensure that capabilities stated in the stories become a reality. As mentioned earlier, Sunbird building blocks are the underlying foundation for the DIKSHA platform. All code is maintained on multiple Sunbird GitHub repositories.

Note: *For a list of all the Sunbird repositories on GitHub that house DIKSHA code, refer to* ***Appendix T - List of code repositories***.

Consolidate

The Consolidate phase of the Build process involves merging the code from individual areas to the master branch and unit testing it.

Communicate

The Communicate phase involves informing all internal and external stakeholders about production ready documentation.
Documentation has a process and lifecycle of its own that runs parallel to the Develop and Consolidate phases.
The following is a visual representation of the documentation process:

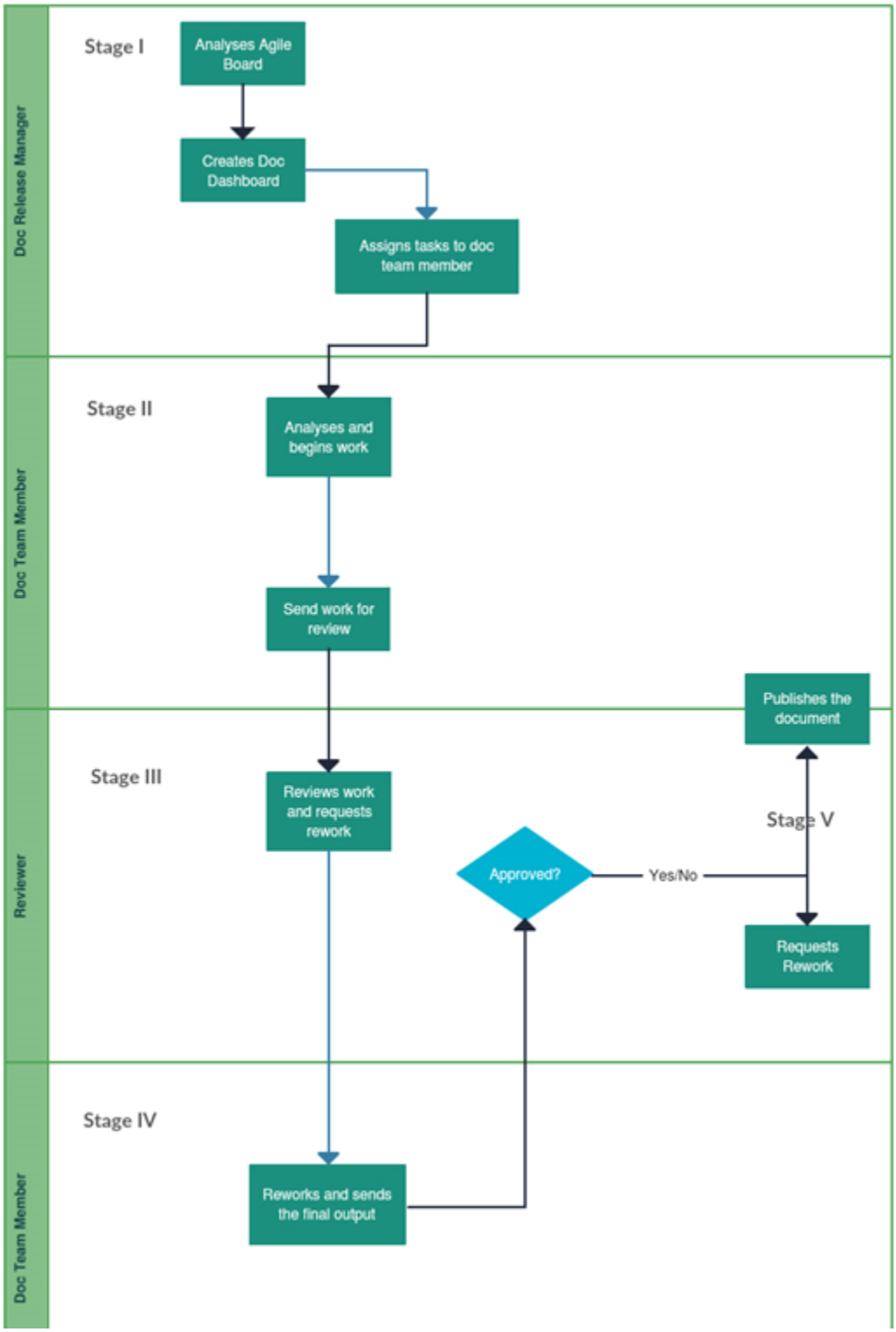

Figure 10: Release Documentation Process

Note: *For details on the documentation process, refer to* ***Appendix F - Release Documentation Process.***

The team produces the following documentation as part of the communicate phase:

- Release Notes
- Release Upgrade Document
- Product Functionality User Manuals
- API Reference Documentation

Note:

1. *For DIKSHA release notes, refer to* ***Appendix E - DIKSHA Release Notes.***
2. *For details on the release upgrade document, refer to* ***Appendix G - DIKSHA Release Upgrade Document.***
3. *For details of the Product Functionality User Manuals, refer to* ***Appendix H - DIKSHA User Manual.***
4. *For details of the API reference documentation, refer to* ***Appendix I - API Reference Documentation.***

Test

After the Build stage is complete, the product enters the test phase. This phase is crucial to guarantee the quality of the product. The team does system testing, regression testing and acceptance testing.

Quality Assurance Process

Test cases are written to ensure that the features developed adhere to all the functional and non-functional requirements. The test cases are reviewed and executed
The following is a visual representation of the test case execution process

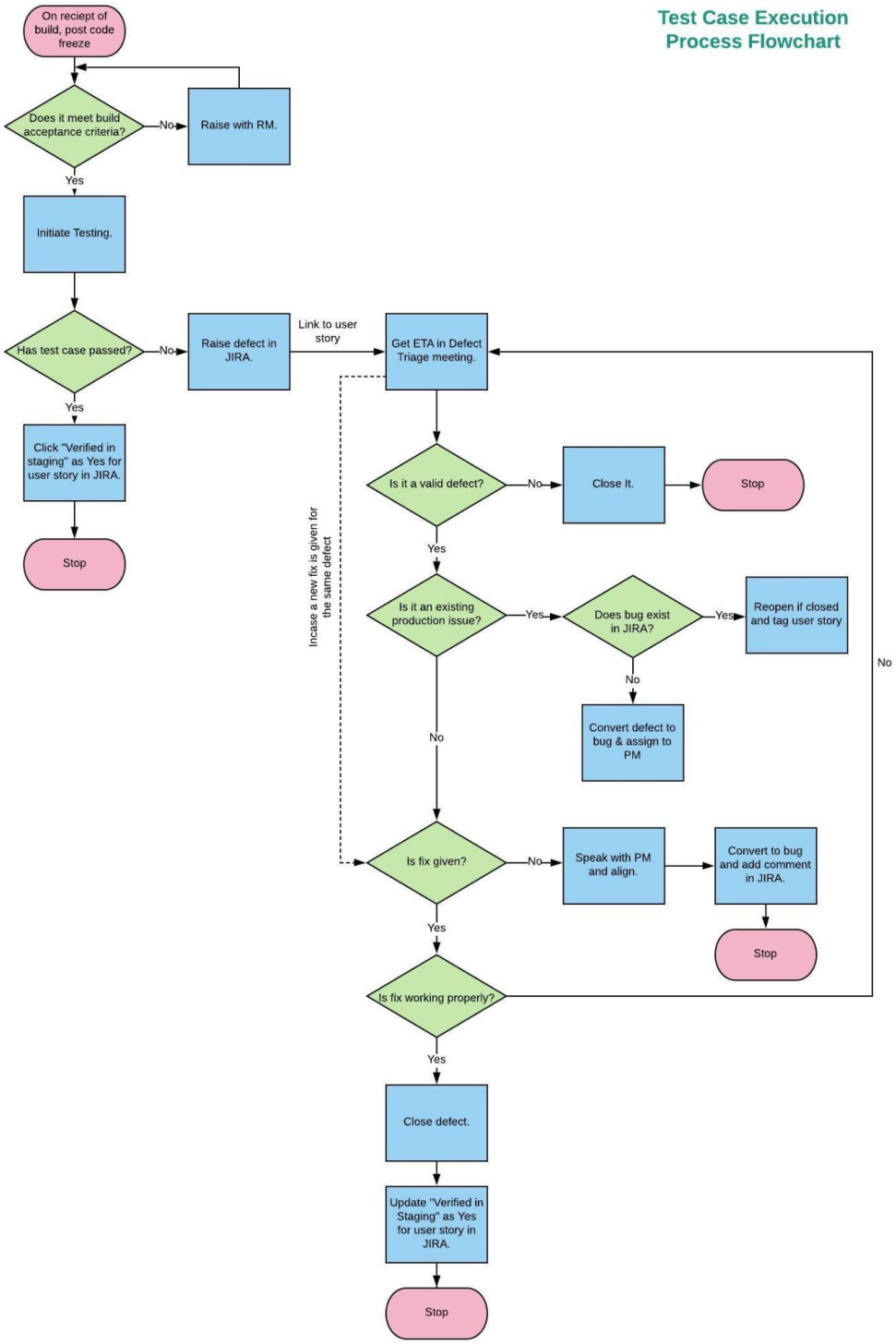

Figure 11: Test Case Execution Process

Note: *For details on the quality assurance process, refer to* ***Appendix J - DIKSHA Quality Assurance Process.***

Root Cause Analysis (RCA)

At the end of each release the team analyzes the root cause of defects and decides on quality metrics to be achieved for the subsequent release. The following is a visual representation of the RCA process.

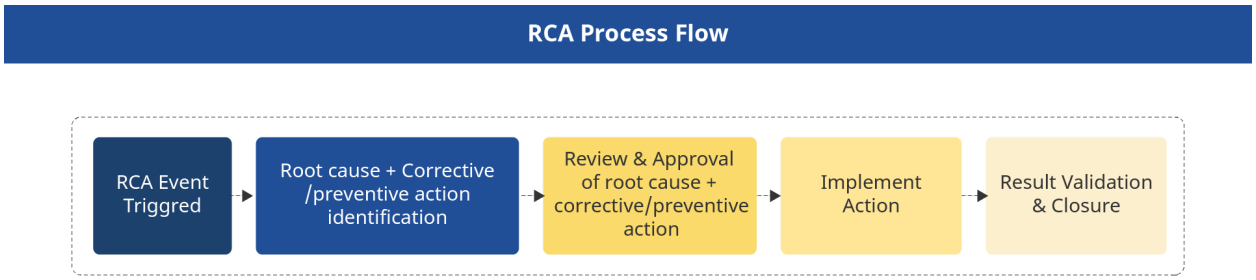

Figure 12: RCA Process Flow

Note: *For details on the root cause analysis process, refer to* ***Appendix K - DIKSHA RCA Process.***

Test Execution Plan

Once a Sunbird release happens, it is deployed in a preprod environment of DIKSHA and DIKSHA UAT is executed as per a plan. This typically happens in weeks six (6) and seven (7) of the release cycle.

Note: *For a sample of the test execution plan, refer to* ***Appendix L - DIKSHA UAT Execution sheet.***

Test cases

Test cases are written specifically to test for DIKSHA functionality. Test cases are executed as per the test execution plan.
Note: *For sample test cases, refer to* ***Appendix M - Regression Test case suite for DIKSHA NF & Circle Regression.***

Deploy

After the Test phase, DIKSHA is deployed to production. The team follows instructions either from the DIKSHA Release Upgrade document (to upgrade from a lower version to the current version) or the DIKSHA Installation Guide (for a fresh installation) and makes use of various software, scripts and tools to deploy DIKSHA for the portal, mobile or offline desktop.

Note:

1. *For details on the software used, refer to **Part A,** section **DIKSHA Software List.***
2. *For details on deployment scripts, refer to **Part B,** section **Build Process,** subsection **Coding Repositories.***

DIKSHA Deployment Architecture

DIKSHA's deployment architecture is best described using the following diagrams:

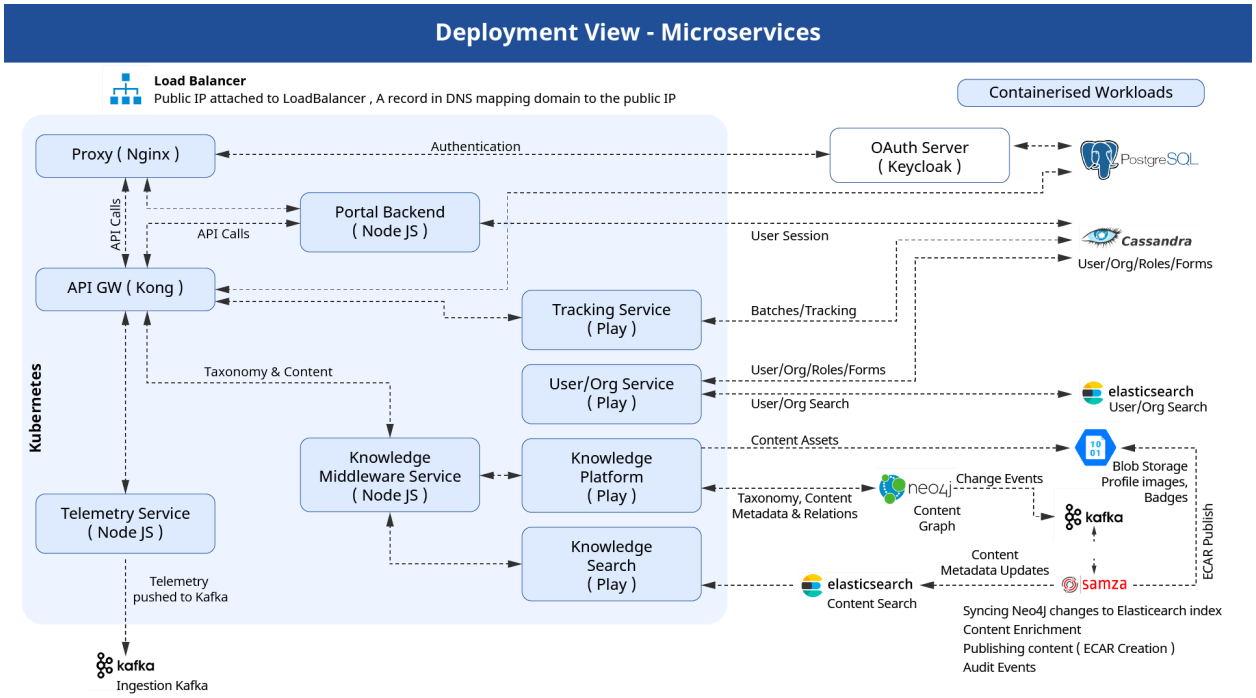

Figure 13: Deployment view - Microservices

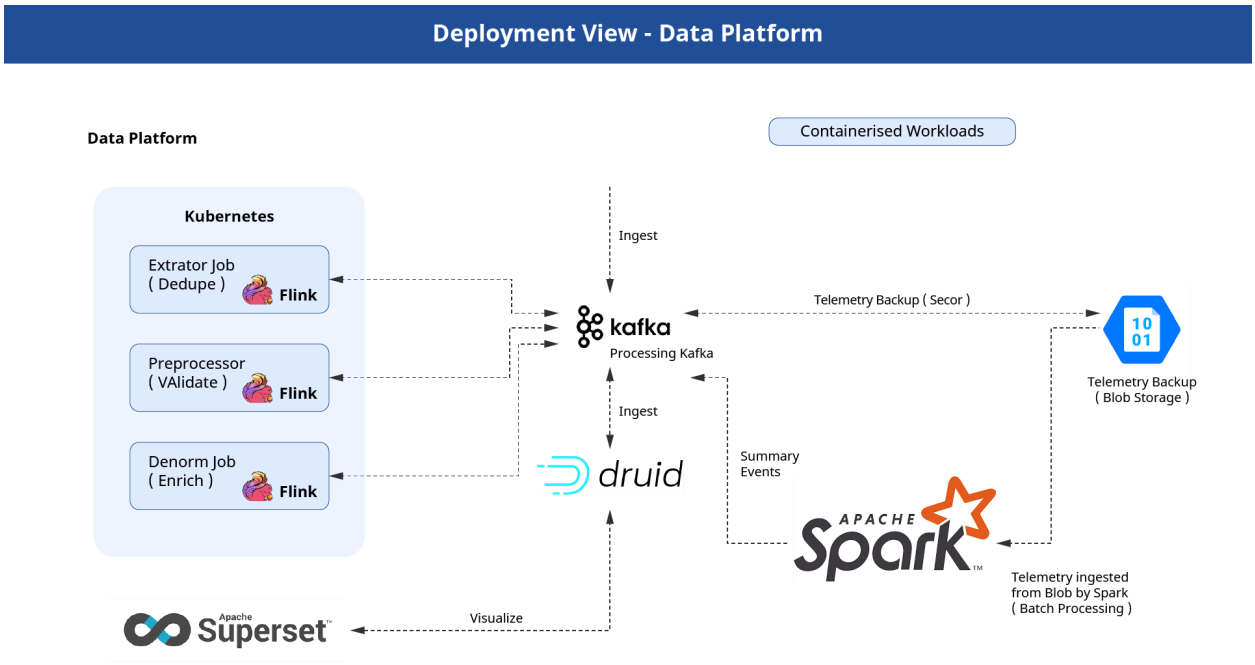

Figure 14: Deployment view - Data Platform

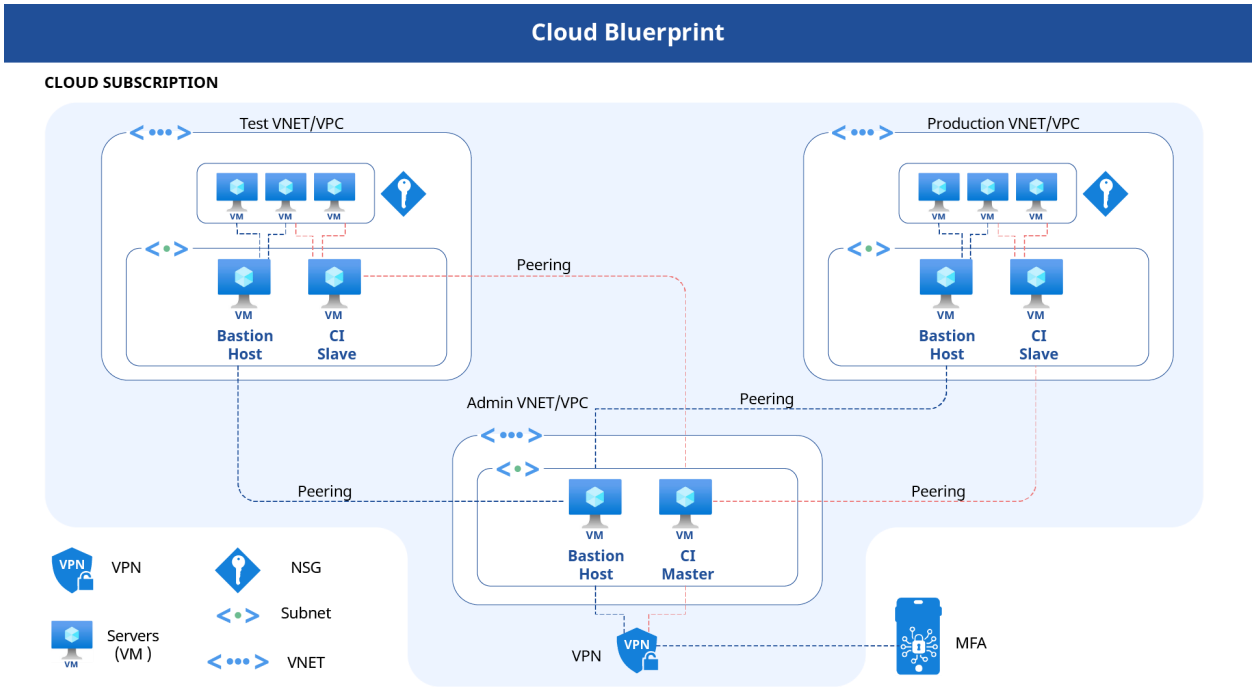

Figure 15: Deployment view - Cloud Blueprint

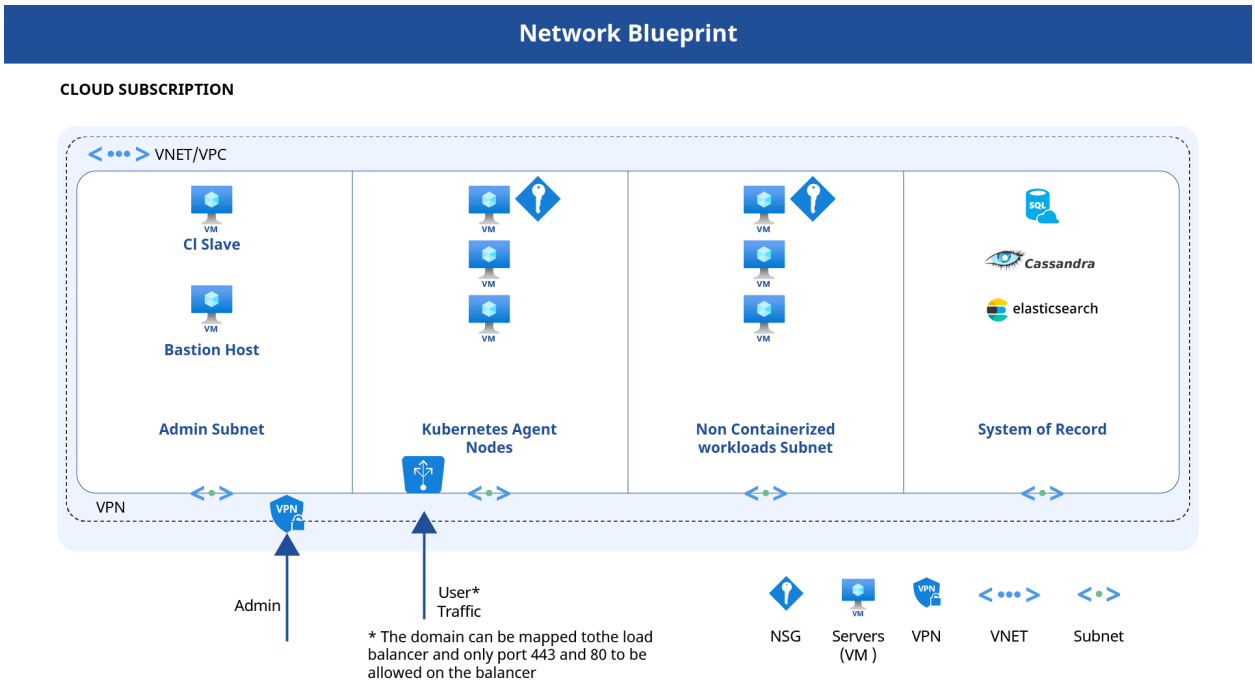

Figure 16: Deployment view - Network Blueprint

DIKSHA Release Deployment Management Process

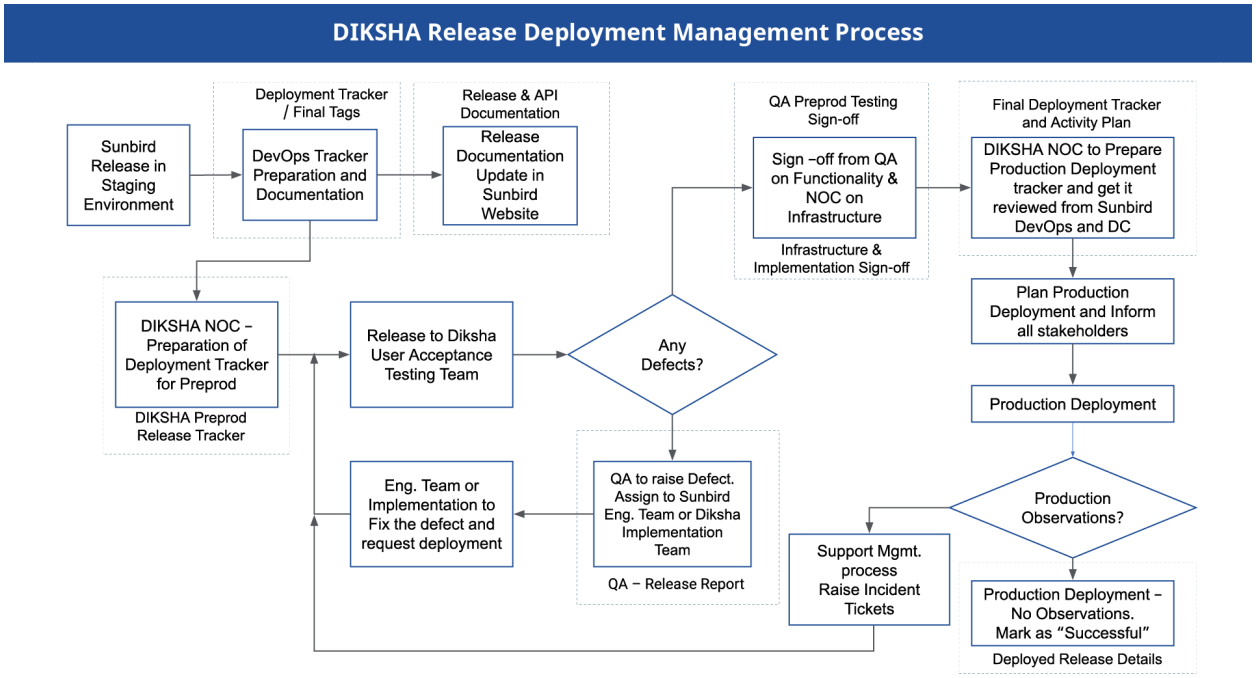

Figure 17: DIKSHA Release Deployment Management Process

The preceding diagram is a visual representation of the deployment process. The deployment process is managed using various documents and trackers at different stages in the process. For example; the DevOps Tracker, Release Notes, etc. The Installation Guide provides complete instructions on how to deploy and configure DIKSHA.

Note:

1. *For DIKSHA release notes, refer to* ***Appendix E - DIKSHA Release Notes.***

DIKSHA Release Deployment Environments

DIKSHA is deployed in multiple environments. The deployment journey is represented in the following diagram:

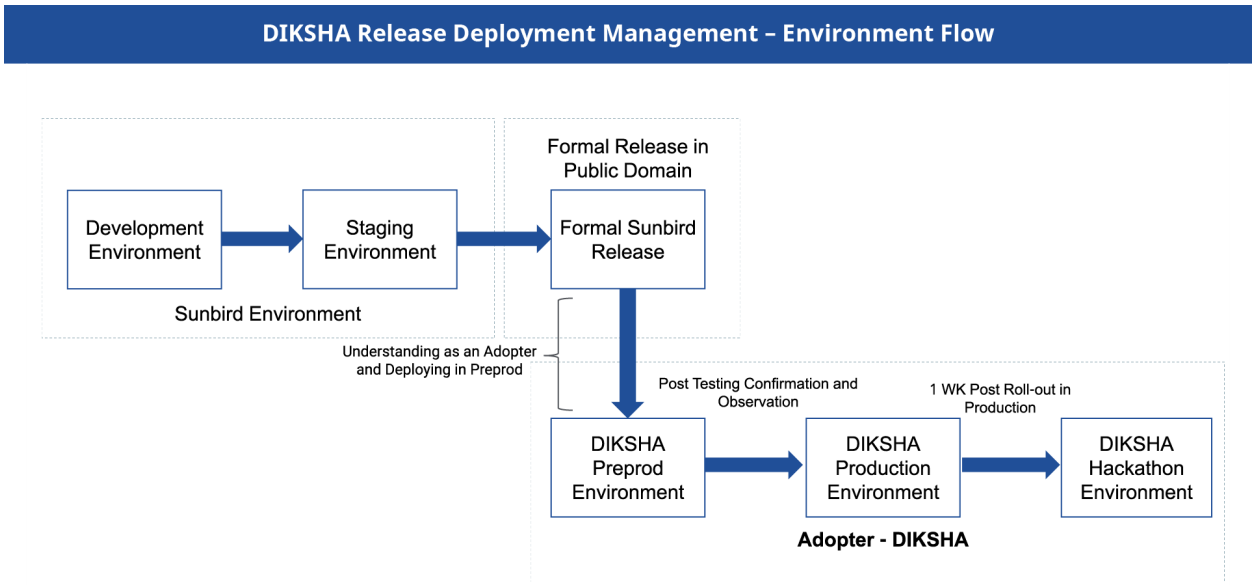

Figure 18: DIKSHA Release Deployment Management - Environment Flow

DIKSHA Release Deployment Milestones

Milestone Gate No.	Gate Category	Qualification	Owner	Measurable Output
1	Sunbird Release	Latest Release of Sunbird and available in Public Domain	Sunbird DevOps	Sunbird Deployment Details, Release Documentation, API Documentation
2	Preprod Deployment Tracker	Deployment Tracker with Tags to Build and Deploy	Diksha NOC & Implementation	Preprod Deployment Tracker
3	Release in Preprod	Successful Roll-out	Diksha NOC & Implementation	Successful Deployment
4	Testing in Preprod	Testing Sign-off from QA	Diksha QA	Preprod Sign-off Mail
5	Infrastructure Sign-off in Preprod	Infrastructure and Pipeline Monitoring and Sign-off	Diksha NOC & Data Team	Preprod Release Dashboard & Data Team Feedback
6	Production Deployment Tracker and Review	Deployment Tracker for Production with latest tags and Activity Plan (Execution steps – SOP)	Diksha NOC	Sunbird DC Review and Production Deployment Tracker
7	Stakeholder Communication	Communication mail	Diksha Support	Communication mail on activity and any downtime (if applicable)
8	Production Deployment	Successful Roll-out	Diksha NOC & Implementation	Successful Deployment
9	Testing and Observation in production	Testing Confirmation and System Observation	Diksha NOC & QA	Deployment status and QA Report
10	Retrospective	Things went well, Can be done better, Issues/Reports	Diksha NOC, Implementation and QA	Learnings and MO

Deploy Screenshots

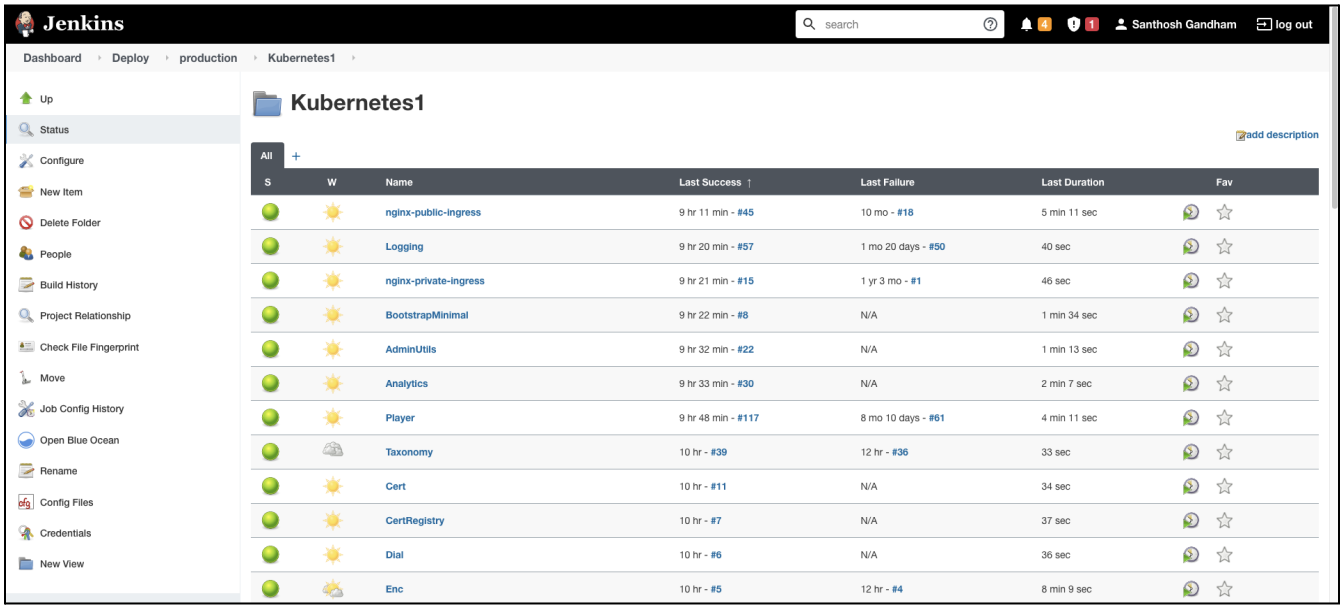

Figure 19: Jenkins jobs list to deploy the microservices running in kubernetes cluster

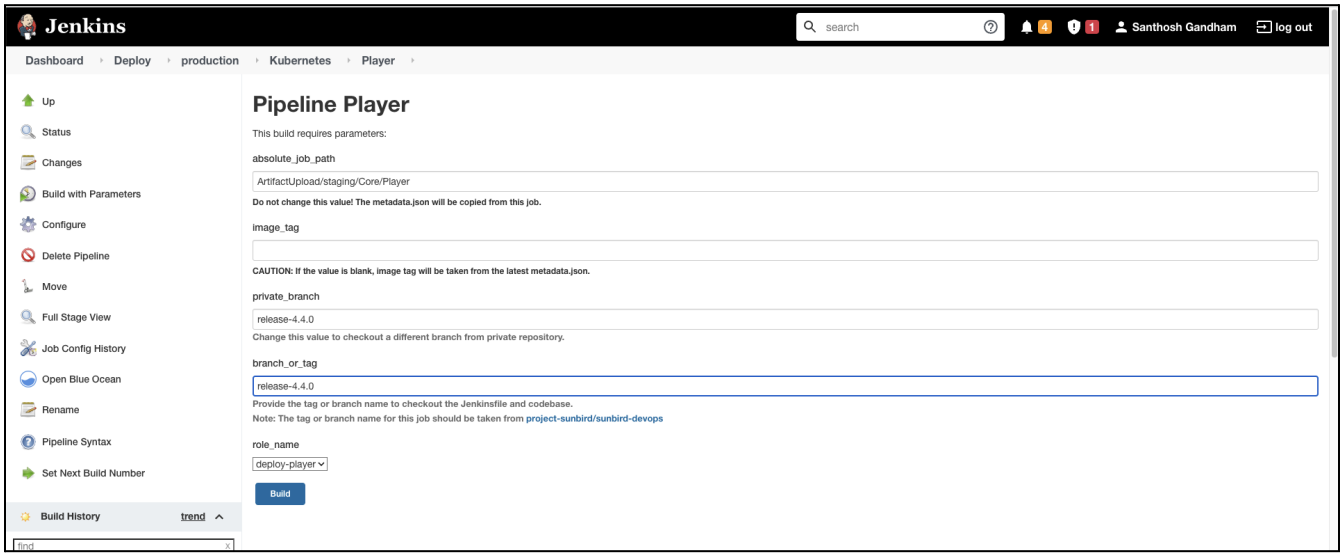

Figure 20: Jenkins parameters to deploy player microservice

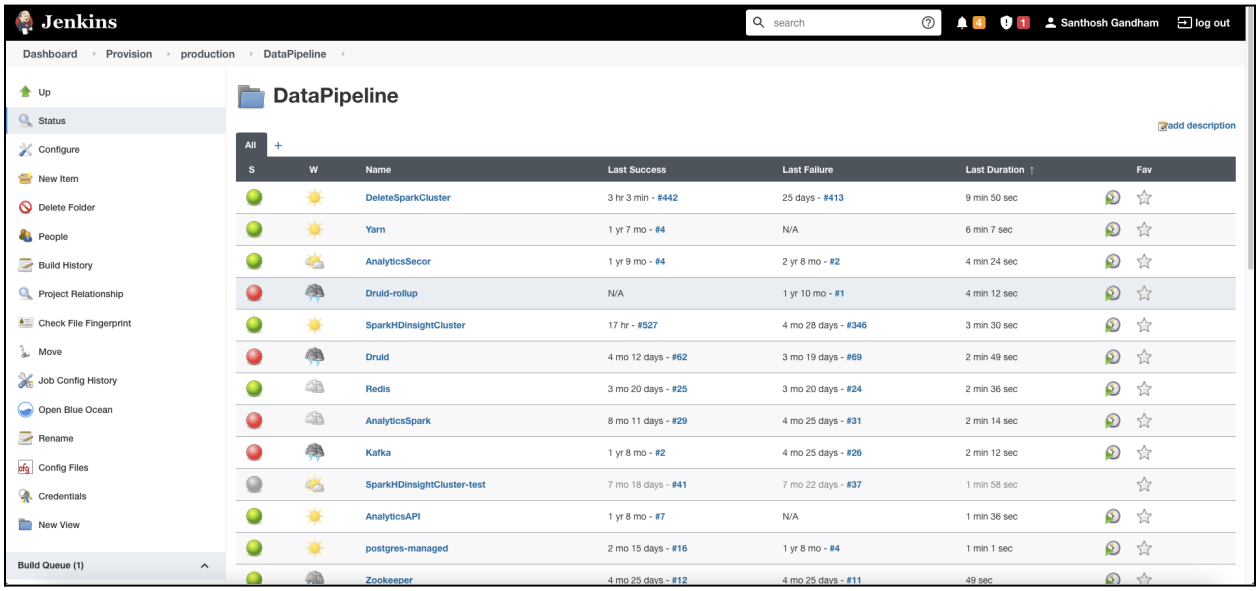

Figure 21: List of Jenkins jobs to provision and configure data pipeline services

Monitor

NOC Monitoring Scope

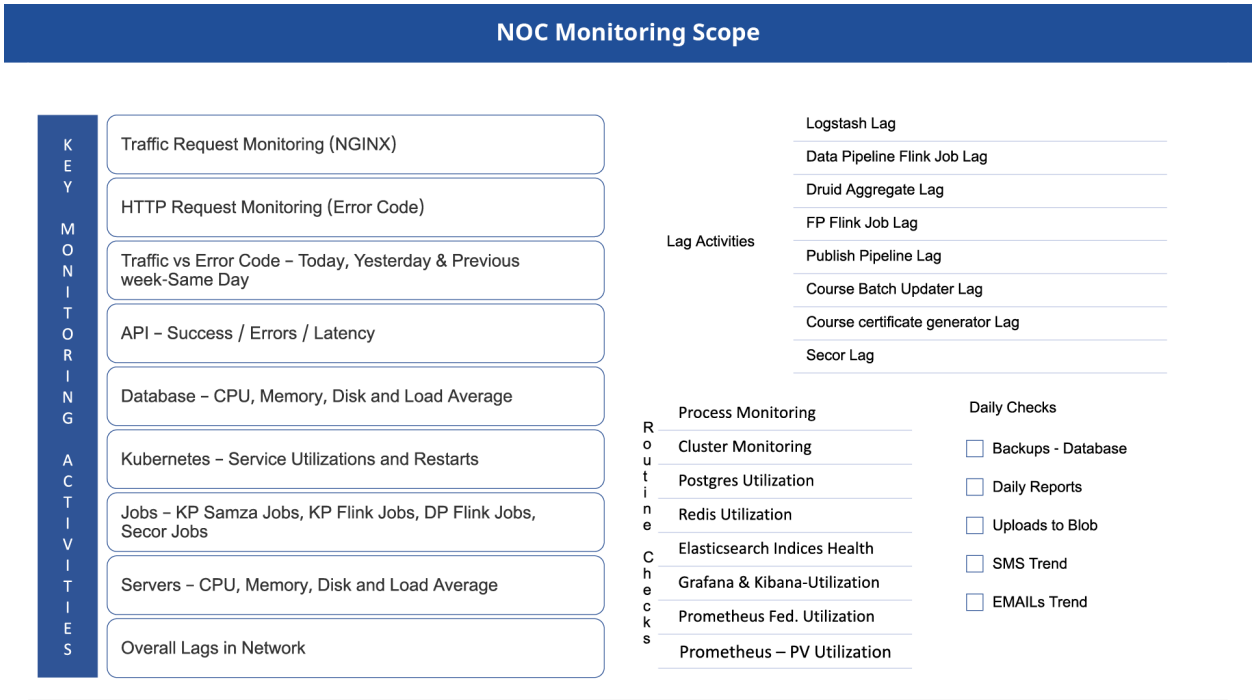

Figure 22: Scope of NOC Monitoring

Emergency Process

Emergency Approach

- Plan A – Normal Emergency Restoration. Respond to the issue to look for the optimal solution - Primary
- Plan B – Alternate Approach for Restoration
- Plan C – Blocked and needs time. Get help from the front facing team to manage the communication

Incidents that Qualify for Emergency

The following are broad categories that are to be considered as P1:

- Any Impact on Azure Region (Production environment is in Central India Region
- Any impact to Azure services (PostgreSQL, Kubernetes Services)

- Azure Impact to several Storage accounts (Blobs and Disk – As if Disk is not reachable, application's may not perform properly)
- Frequent Service Restarts for containerized and non-containerized application
- Services not starting Up for containerized and non-containerized application
- Any network issue on Load Balancers
- Database Not Reachable
- Performance Degrade of API's (Continuous signs of degradation identification from on field issues and equivalent reflection in Latency)
- 5xx Error > x% (Continuously for 15 mins)
- 4xx Error > x% (Continuously for 15 mins)
- Pipeline Not processed for more than 8 hrs
- Any User related Jobs Failures and not responding to restarts
- TPS spike (usage)
- Service down (denial of service)
- Network services down
- Accessing service issues (P1, bugs)

Incident Management Process

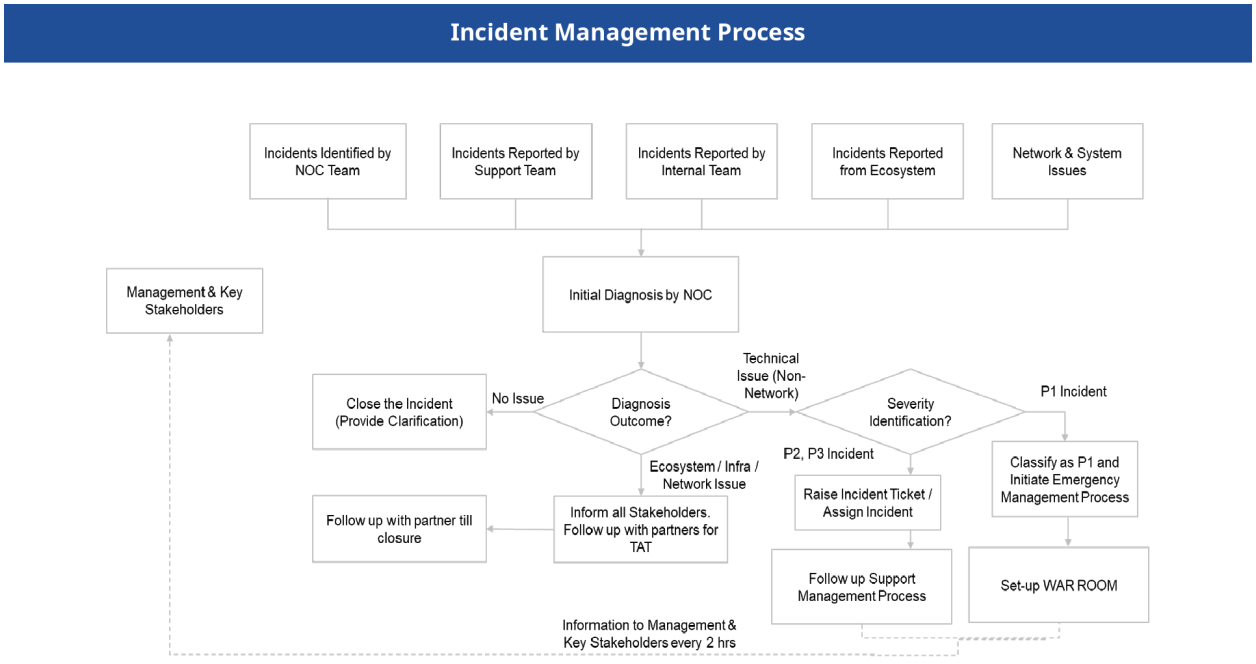

Figure 23: Incident Management Process

Flow Transition Process

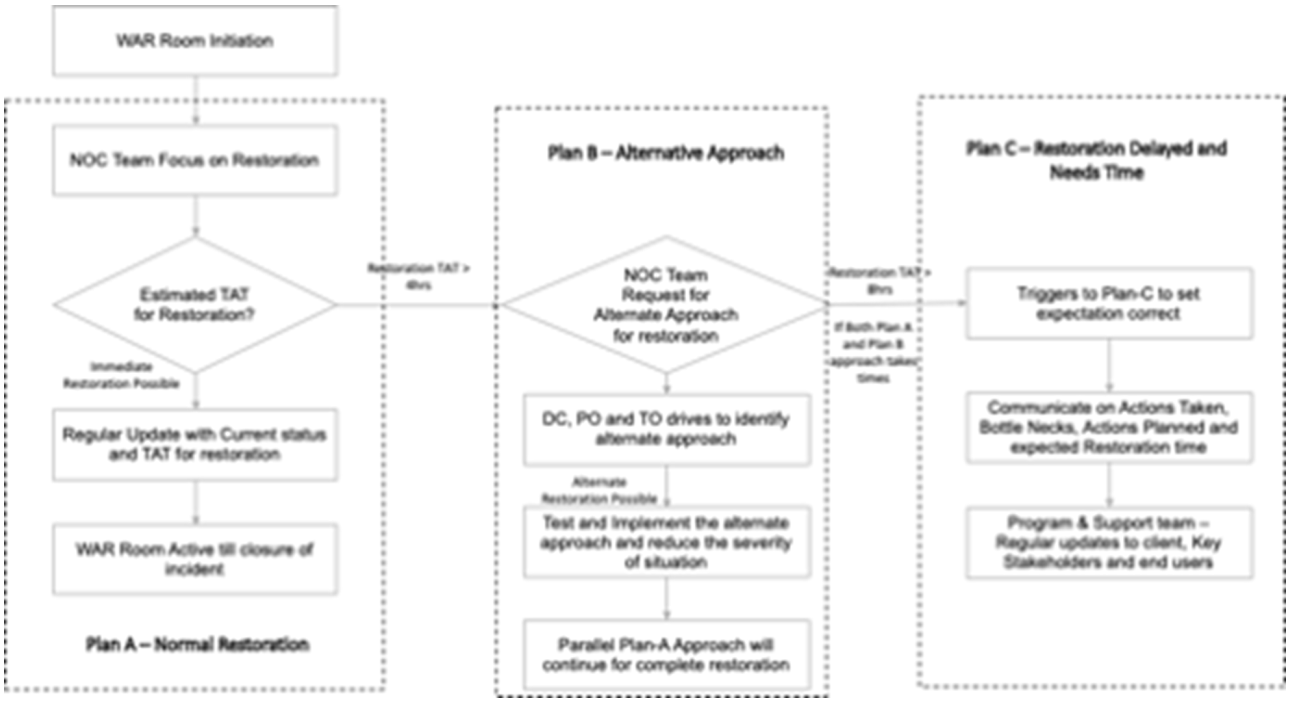

Figure 24: Flow Transition Process

Critical Dashboards - Grafana

Dashboard Name	Dashboard Link	Purpose
Dashboard-All	https://diksha.gov.in/grafana/d/000000006/dashboard-all?orgId=1&refresh=30s	For TPS, Error, Load Average and Disk Monitoring
API Manager	https://diksha.gov.in/grafana/d/qRnVoj7Mz/api-manager?orgId=1&refresh=1m	For API - Request, Success, Error, Latency, Request / response sizes
Lag Dashboard	https://diksha.gov.in/grafana/d/vfdQRdtMz/lag-dashboard?orgId=1&from=now-15m&to=now&refresh=1m	Ingestion cluster, DP Flink, KP Flink, Samza Jobs, Druid, Secor Lag Availability

Dashboard Name	Dashboard Link	Purpose
Data Pipeline Metrics Flink	https://diksha.gov.in/grafana/d/YngLt0kGz/data-pipeline-metrics-flink?orgId=1&refresh=5m&from=now-1h&to=now	DP Flink Jobs -- Success , Throughput and Lag with other parameters
KP Flink Job metrics	https://diksha.gov.in/grafana/d/YngLt0kG/kp-flink-jobs-metrics?orgId=1&from=now-1h&to=now	KP Flink Jobs -- Success , Throughput and Lag with other parameters
LP Pipeline Lag	https://diksha.gov.in/grafana/d/9I50utWZz2/lp-pipeline-lag?orgId=1&from=now-15m&to=now&refresh=5m	Samza Jobs -- Lag with other parameters
Ingestion Kafka cluster Overview	https://diksha.gov.in/grafana/d/jwPKIsnlii/ingestion-kafka-cluster-overview?orgId=1&from=now-15m&to=now&refresh=5m	Ingestion cluster - Secor Job Lags, broker
Processing Kafka cluster Overview	https://diksha.gov.in/grafana/d/jwPKIsni/processing-kafka-cluster-overview?orgId=1&from=now-15m&to=now&refresh=5m	Processing cluster - Secor Job Lags, broker
Load Metrics	https://diksha.gov.in/grafana/d/B-a5y6eZz/load-metrics?orgId=1&from=now-1h&to=now&refresh=1m	CPU, Memory, Load Average, Disk utilization of all servers
Host Details	https://diksha.gov.in/grafana/d/mrXQlqCZz/host-details?orgId=1	Individual server basis - CPU, Memory, Disk, Load Average, IOPS, Network Usage
Kubernetes / Pods	https://diksha.gov.in/grafana/d/ab4f13a9892a76a4d21ce8c2445bf4ea/kubernetes-pods?orgId=1&var-datasource=Prometheus&var-cluster=&var-namespace=production&var-pod=adminutils-7fb9b8c968-cbn2k&var-container=All	Individual Pods Restarts
Kubernetes / Compute / Workload - Custom	https://diksha.gov.in/grafana/d/snapyxZGk/kubernetes-compute-resources-workload-custom?orgId=1&refresh=30s	Max. Utilization of CPU and memory of all K8s services

Dashboard Name	Dashboard Link	Purpose
Kubernetes / Compute resources / Cluster	https://diksha.gov.in/grafana/d/efa86fd1d0c121a26444b636a3f509a8/kubernetes-compute-resources-cluster?orgId=1&refresh=30s	Cluster & service basis CPU and memory usage
NGINX	https://diksha.gov.in/grafana/d/0000000025/nginx?orgId=1&refresh=1m	Management - NGINX Dashboard for TPS and Errors
Druid	https://diksha.gov.in/grafana/d/XWxgcjvZz/druid?orgId=1&from=now-3h&to=now&refresh=5m	Druid Raw cluster - Data source availability

Figure 25: Grafana dashboard depicting HTTP status metrics and number of requests processed per second

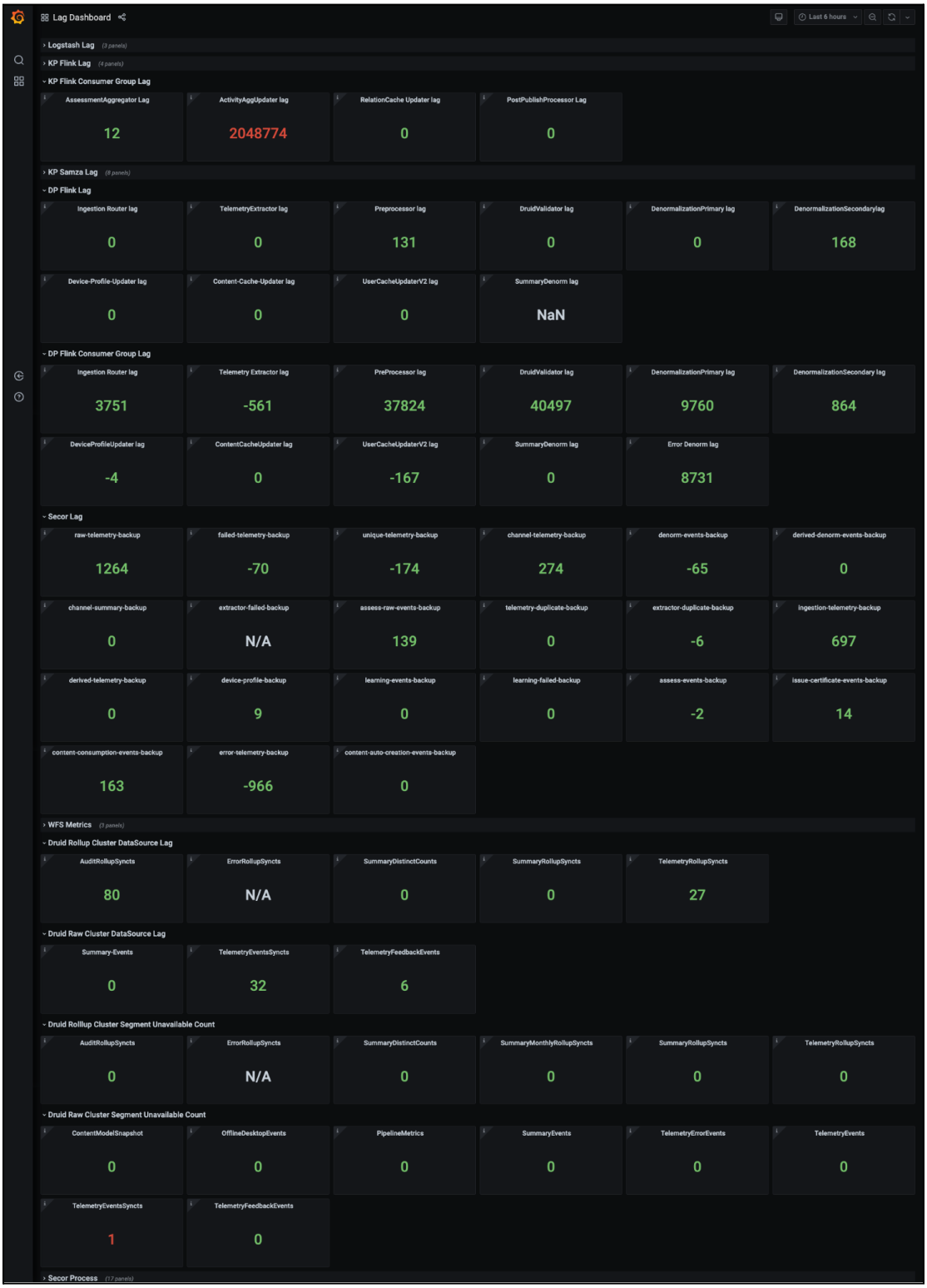

Figure 26: Grafana dashboard depicting the lag metrics at each consumer

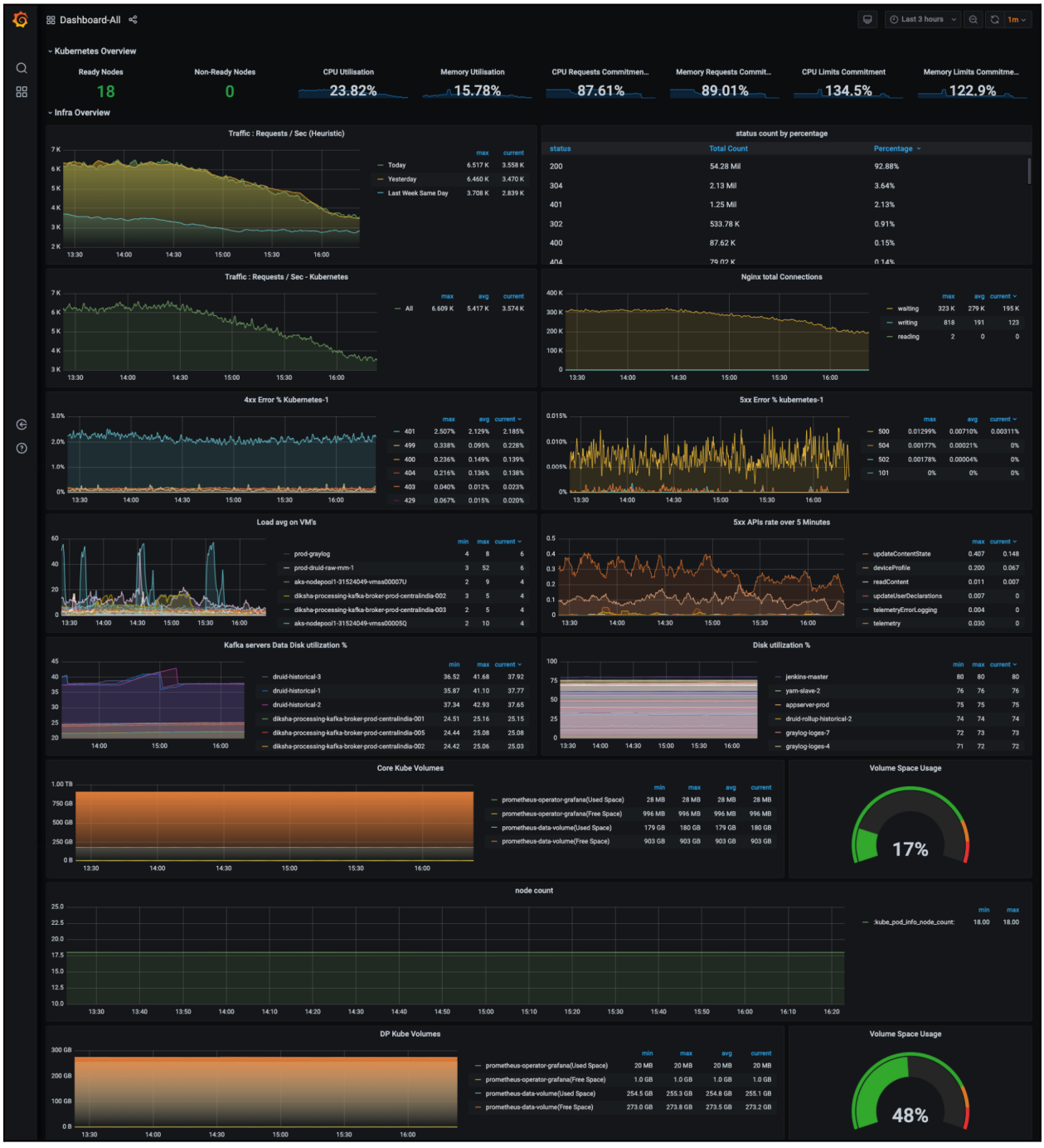

Figure 27: Grafana dashboard providing the first line metrics of the infrastructure

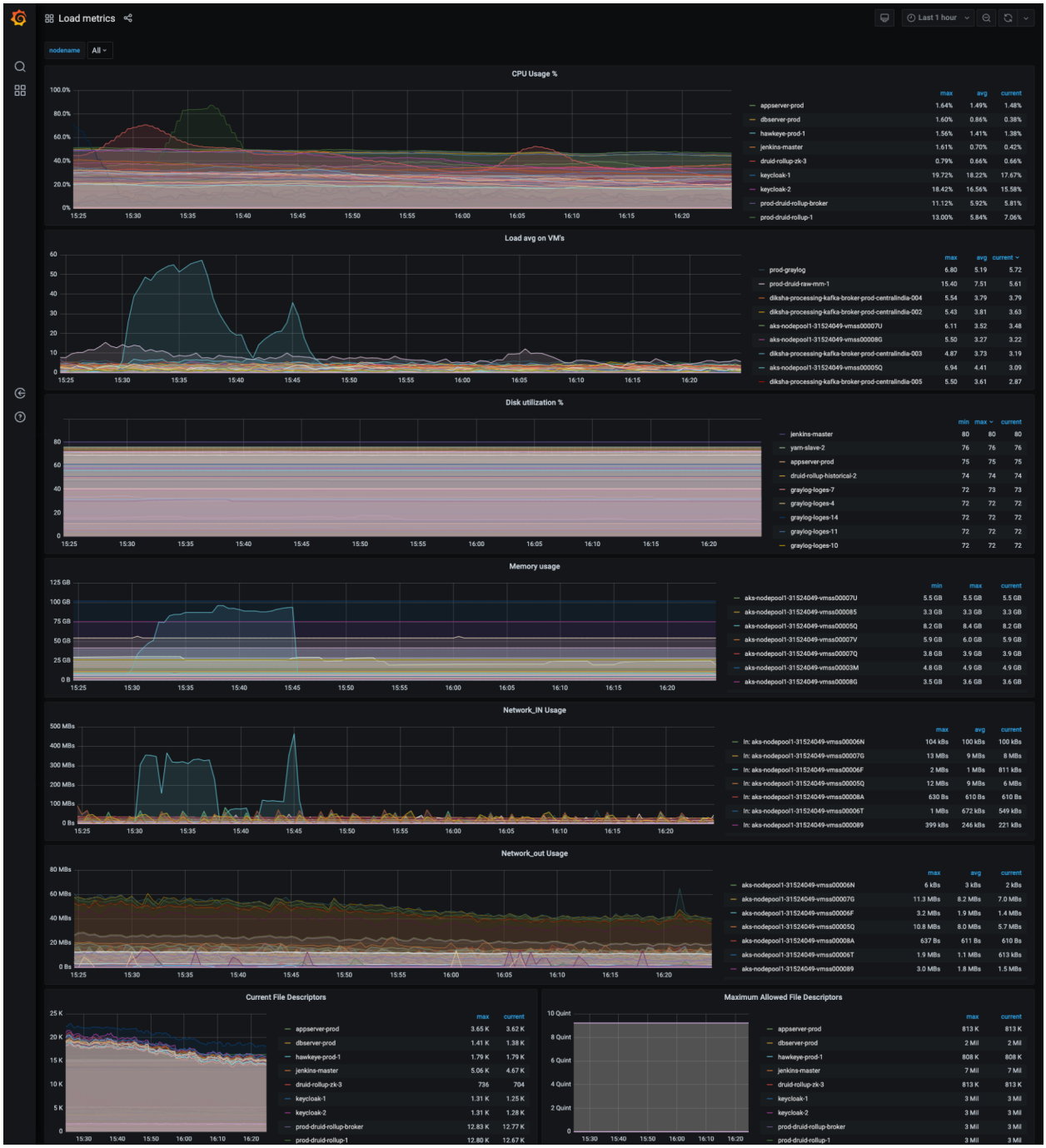

Figure 28: Grafana dashboard depicting the infrastructure utilisation metrics

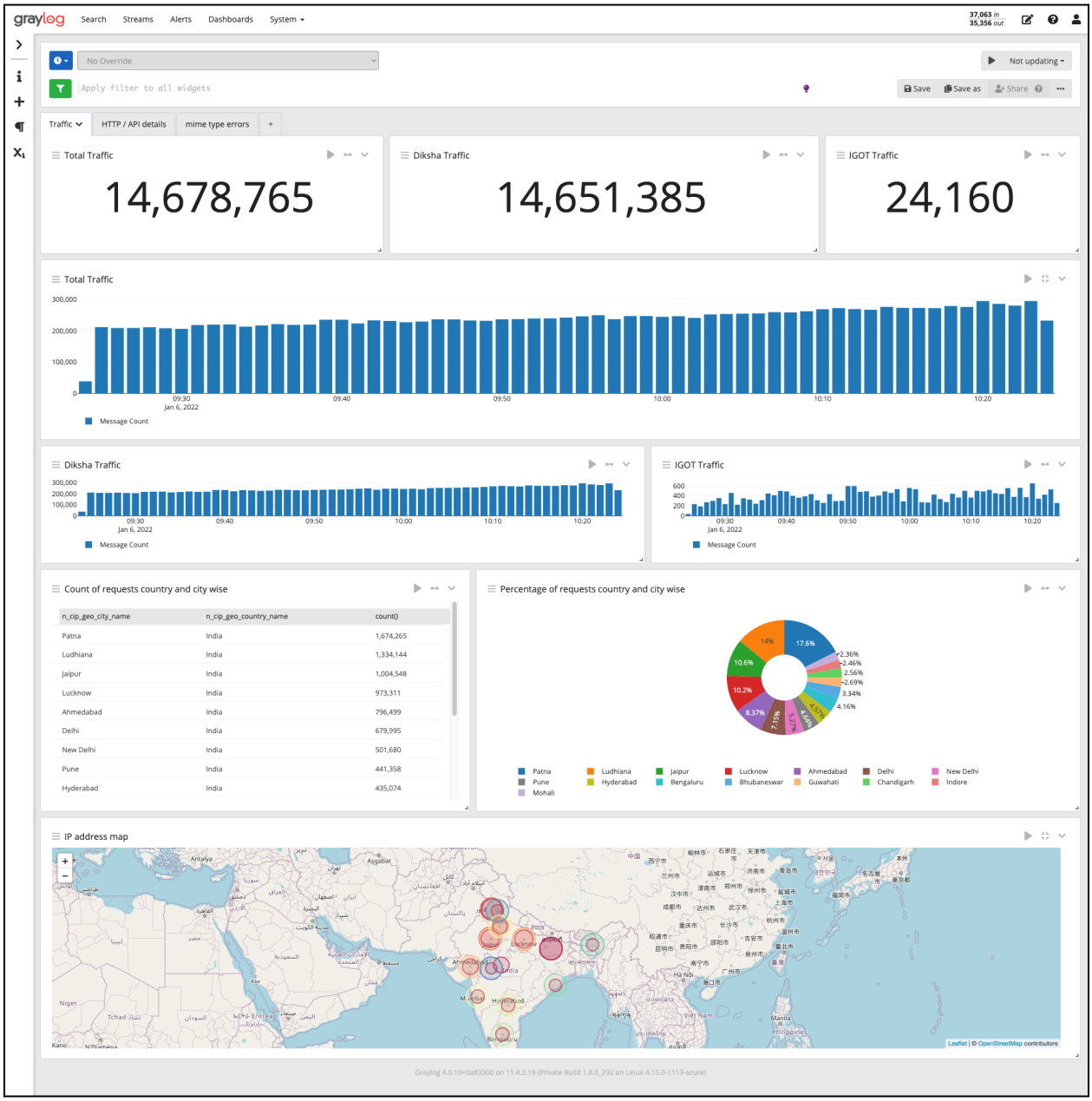

Figure 29: Graylog dashboard detailing the traffic patterns

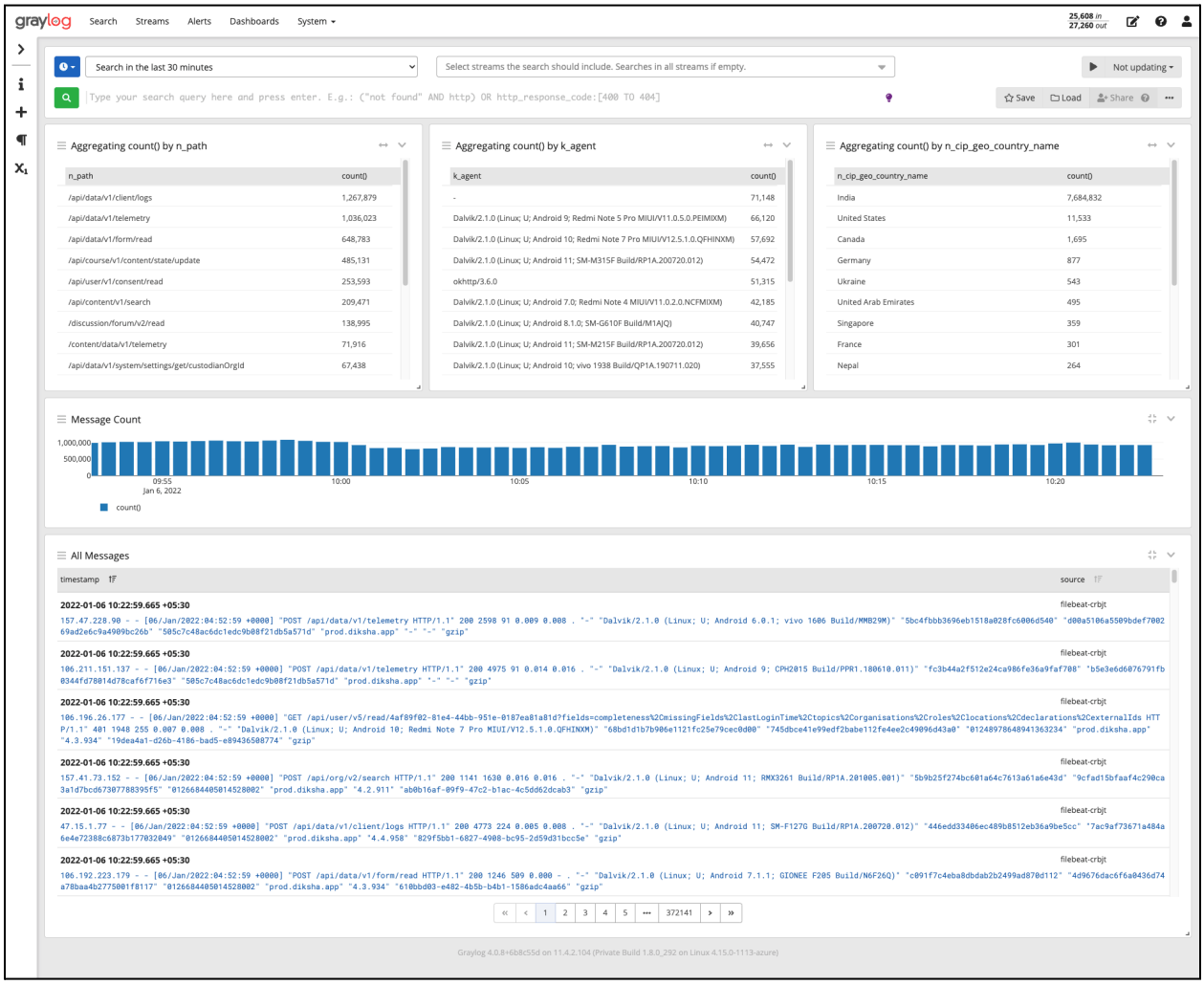

Figure 30: Graylog dashboard to view and analyze detailed logs of application and infrastructure

Incident Handling Process

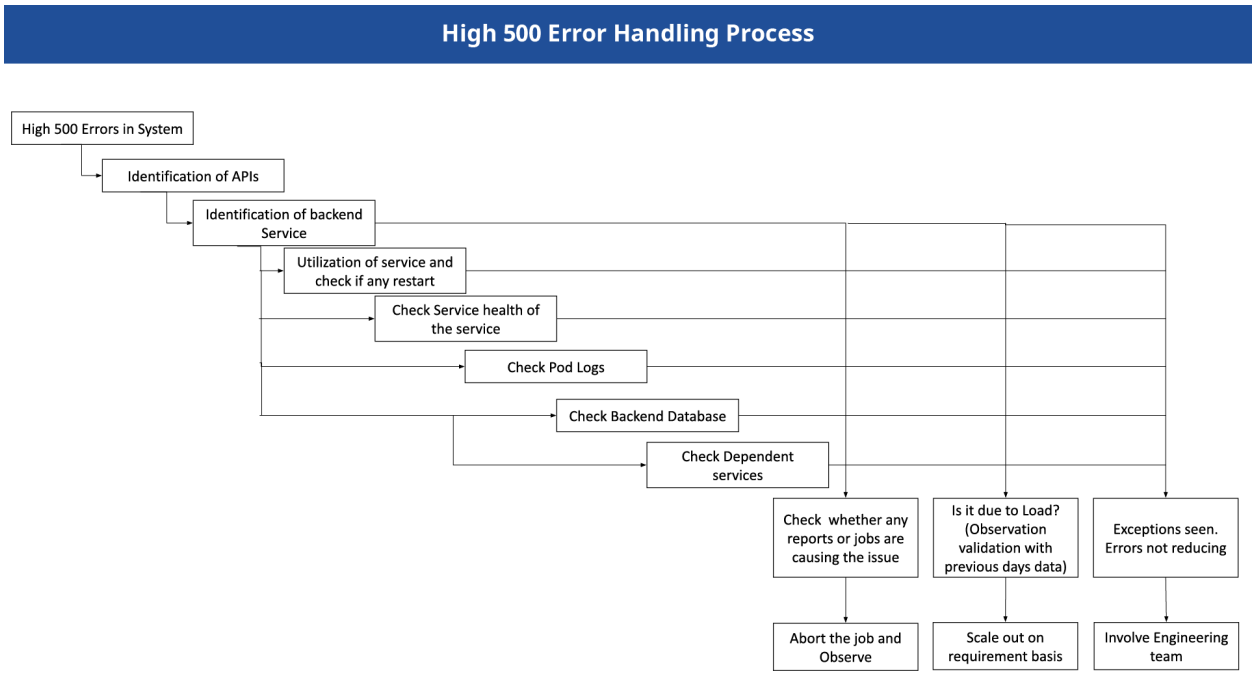

Figure 31: High 500 Error Handling Process

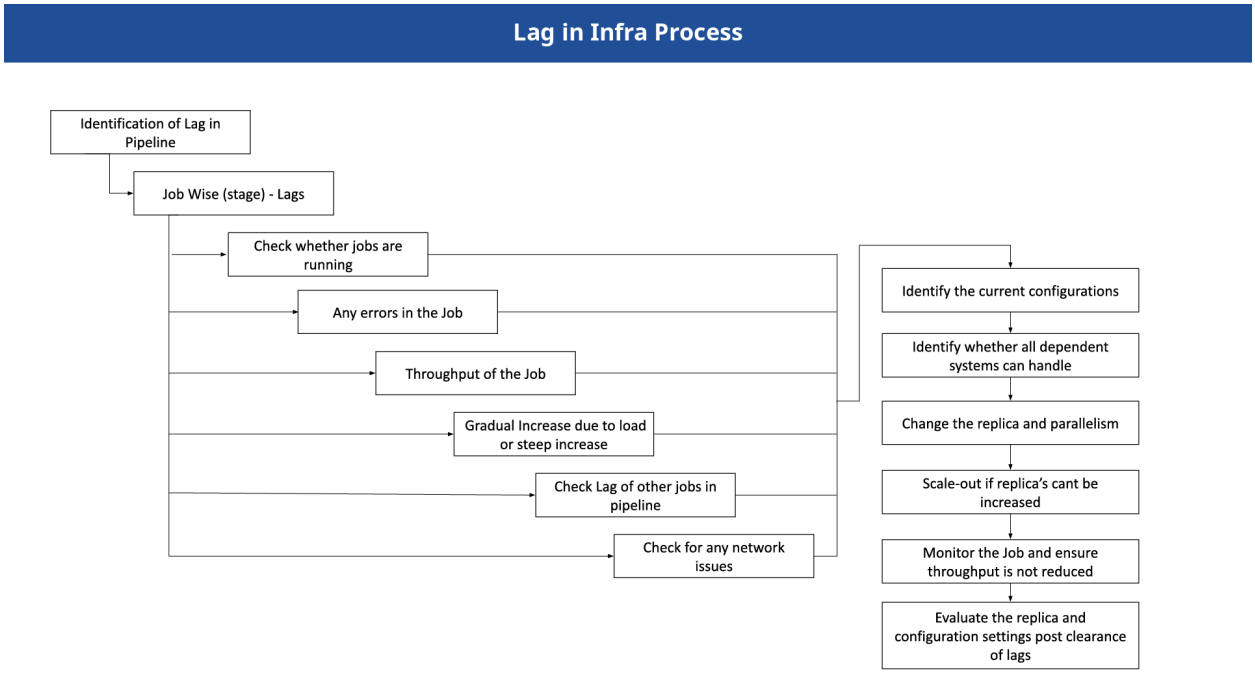

Figure 32: Lag in Infra Process

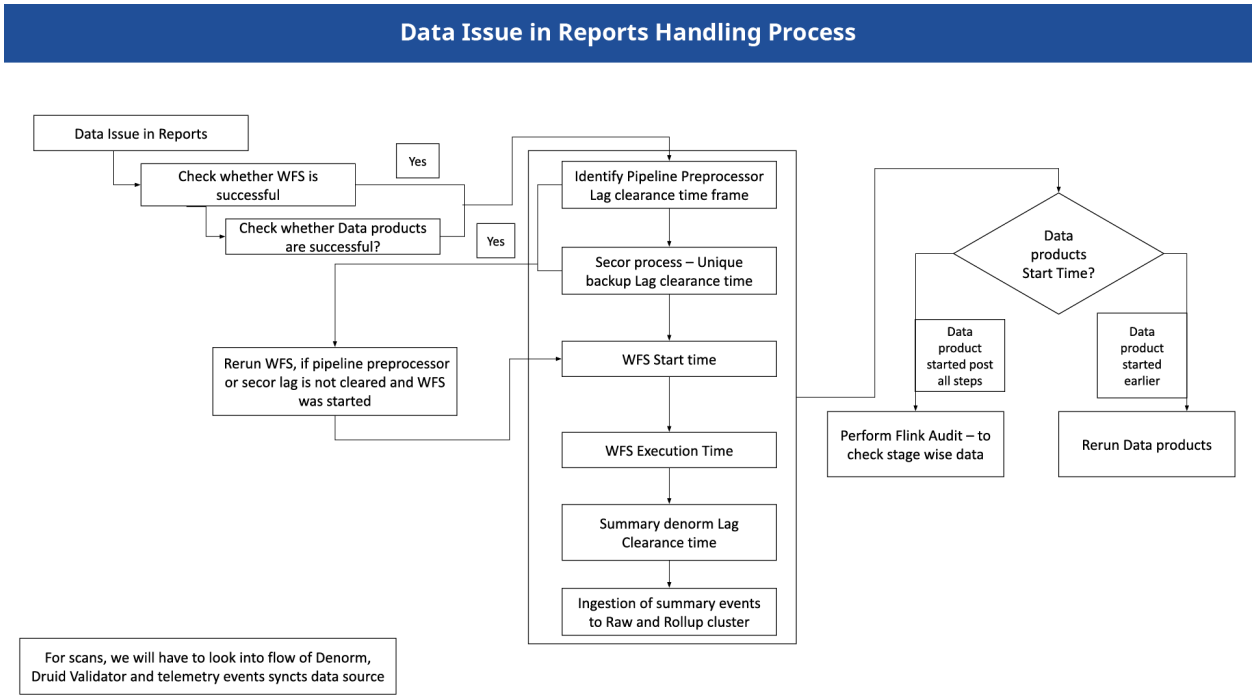

Figure 33: Data Issue in Reports Handling Process

Support

Given DIKSHA's scale, functional and user diversity, end user support is crucial. The DIKSHA team has well-defined S1 and S2 support processes.

S1 Support Process

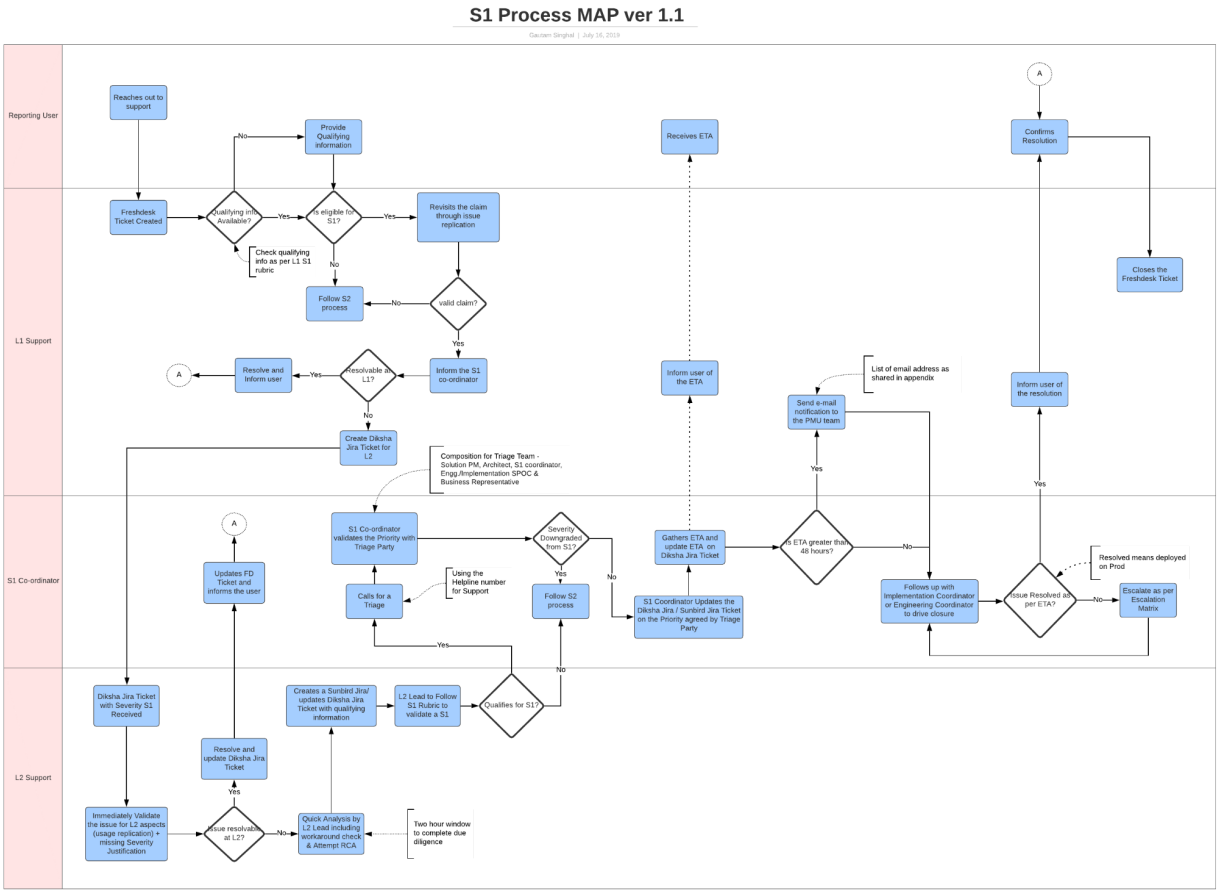

Figure 34: S1 Process Map

Note: *For detailed S1 processes, refer to* ***Appendix N - S1 Support Process.***

S2 Support Process

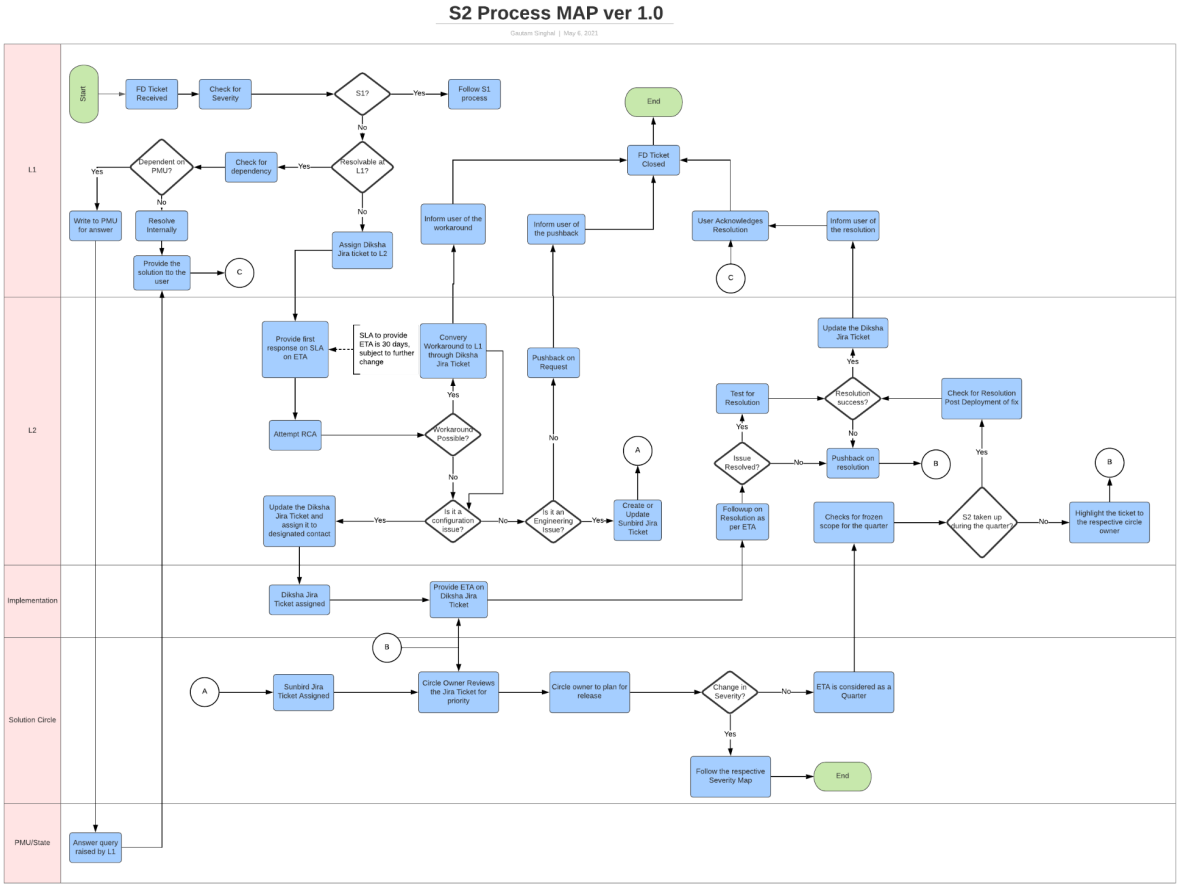

Figure 35: S2 Process Map

Note: *For detailed S2 support processes, refer to* ***Appendix O - S2 Support Process.***

Issues Tracking and Resolution

All issues are raised and tracked through JIRA. They are classified as per priority and assigned to respective teams for resolution and closure.

Note:

1. *For detailed ticket lifecycle, refer to* ***Appendix P - Ticket Lifecycle.***
2. *For a sample issue tracker, refer to* ***Appendix Q - Issue Tracker.***

Setting up DIKSHA Support Services

The guidelines and processes to set up DIKSHA support services are well defined.

Note: To aid any State set up support services for any DIKSHA program, refer to **Appendix U - TPD Support System Guidelines v2.0 for Sharing with States**. These guidelines in this document were used to set up support services for the NISHTHA teacher training program.

Measure

Measurement and analysis of data to generate reports and dashboards is important for various reasons. DIKSHA has robust reporting infrastructure and a well-defined process to generate standard and customized reports. As mentioned earlier, full telemetry is instrumented into DIKSHA. Data captured remains just a data point or number until it becomes a valuable metric. The difference between data and metrics is, while data is merely a number, a metric is a quantitative measurement of data, in relation to what you are actually measuring. For example;the number 10 is a data point, but the metric is 10 content plays. Going further, the collected metric may be analyzed and used within multiple contexts to achieve different goals. For example; a State may want to track content plays of a piece of DIKSHA content to check if it has reached all students of Std. 5. They may also use the same metric to understand if the content needs to be brought down since not many students are playing it, or if they need to take any measures to popularize it.

Report Configuration

Reporting Infrastructure

The Hawk Eye project enables DIKSHA's reporting infrastructure that packages:

- Generation of actionable datasets from Druid
- Configuration of visualizations using Superset
- Publishing the charts as reports using Portal with additional functionality to add a summary, export and download .

The generated reports allow the State PMUs, State outreach program administrators, users from the MHRD and others to view and understand DIKSHA usage in the context of their programs in and across states. They enable the States and the MHRD to drive targeted outcomes and demonstrate concrete progress made towards achieving stated goals.

Infrastructure Components

The reporting infrastructure mainly packages together:

- Generation of actionable datasets from Druid
- Configuration of visualizations using Superset
- Rendering of charts & Reports via the Portal and the Analytics framework
- Additional Summary, Export functionalities

The information flow among the components is briefly depicted in the following diagram:

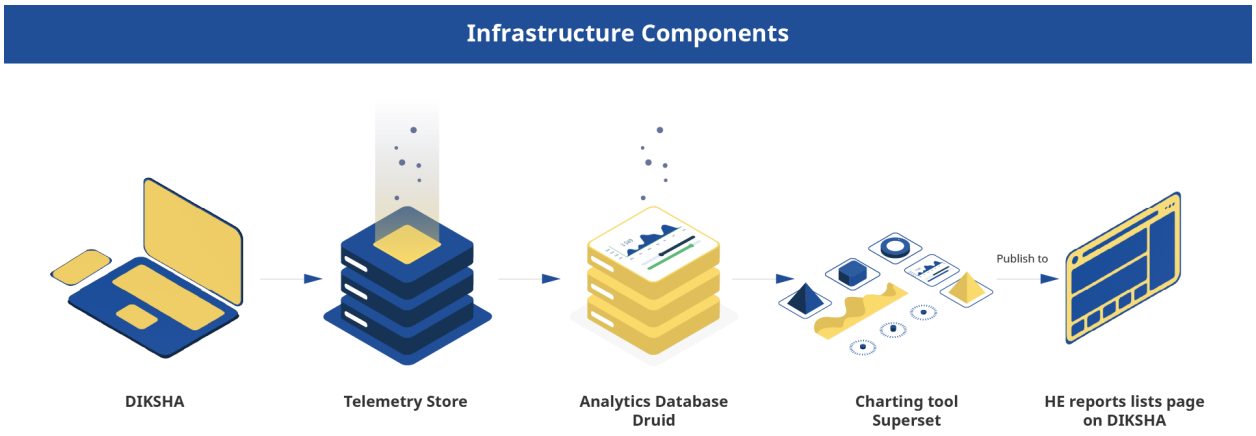

Figure 36: DIKSHA Reporting Infrastructure Components

DIKSHA Report Workflow

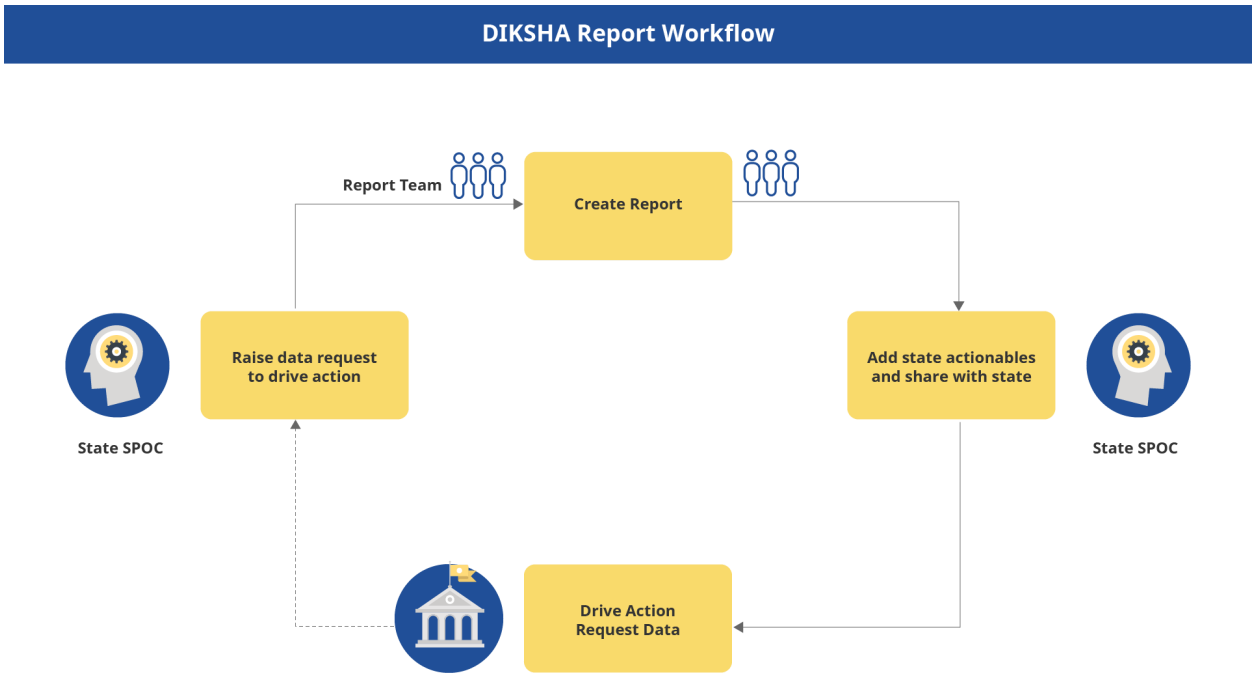

Figure 37: DIKSHA Report Workflow

Using Hawkeye (DIKSHA's reporting infrastructure):

- Reports that can be generated from data in Druid can be turned around faster, and the report/ data set is made available via the portal (requires DIKSHA login)
- A report can incorporate multiple charts
- The report can be enhanced (add action items for report consumers) by a user with the 'Report Admin' role
- Reports can be viewed and downloaded as a PDF/ Image from the portal for sharing (by users with the 'Report Viewer' role)
- The reporting infra can also be used for one-time reports - these can be retired once they've served their purpose

Note:

1. *For detailed information to configure reports, refer to* ***Appendix R - Hawkeye Configuration Guide.***
2. *For reporting workflows, refer to* ***Appendix S - Hawkeye Workflows.***

It must be noted that:

- Reports cannot be generated from thin air. It is dependent on the required telemetry being generated.
- Reports must be tied to actions in order to be effective and being used.
- Data reliability and integrity is handled by underlying data pipeline infrastructure

Report Request Workflow

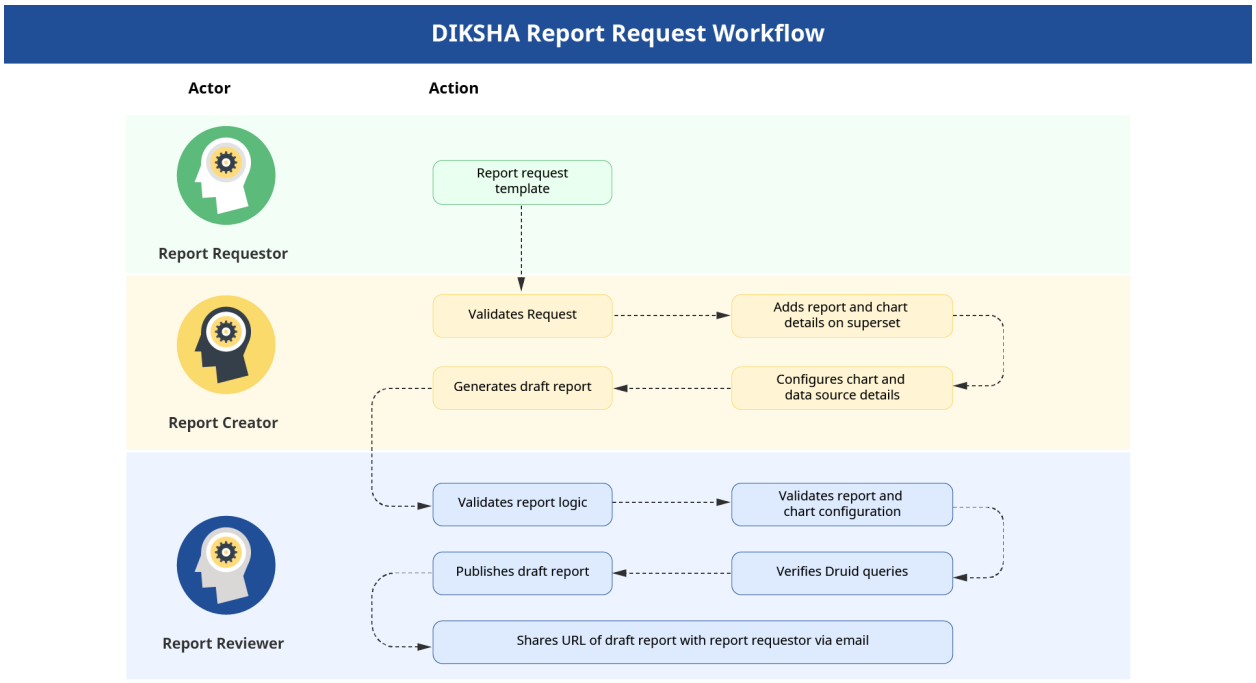

Figure 38: DIKSHA Report Request Workflow

Part C

DIKSHA Solutions

DIKSHA Solutions Overview

DIKSHA solutions enable learning, capacity building, or professional development solutions in the education domain. Depending on the usage scenario, these digital solutions are delivered in three modes:

- Portal applications, that can be accessed when connected to the Internet
- Mobile apps, that can be accessed on mobile devices
- Offline Desktop applications that can be accessed on desktop computers or tablet devices when not connected to the Internet

The following is a visual representation of solutions built across DIKSHA consumption and sourcing with digital infrastructure using building blocks that are not specific to any one solution.

Figure 39: DIKSHA Solutions

Diverse Scenarios and Stakeholders

India's education landscape is large and diverse. Currently, DIKSHA offers solutions that have two interactions for five personas, in three scenarios. The same is illustrated in the following diagram.

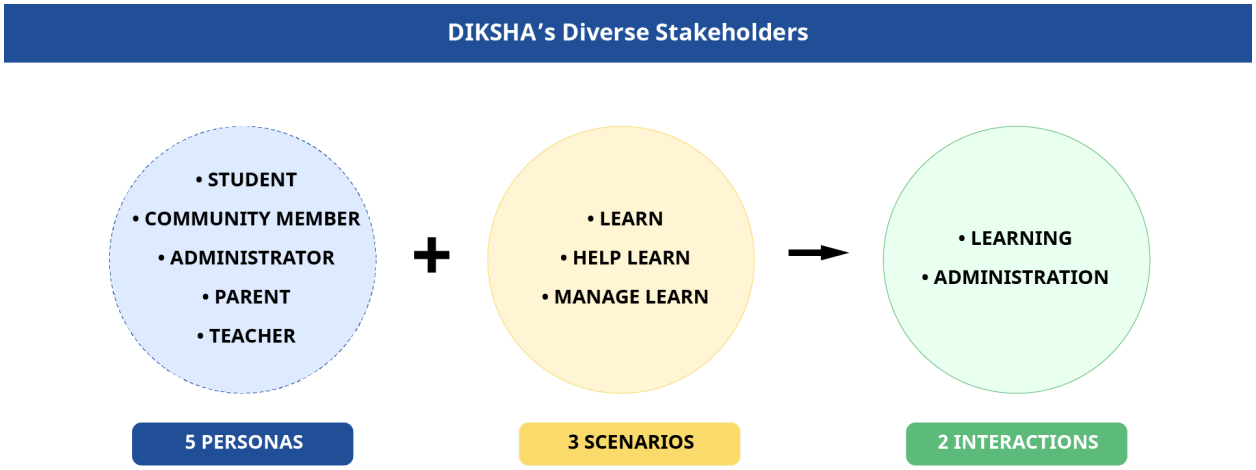

Figure 40: DIKSHA's Diverse Stakeholders

Elaborating the way the DIKSHA platform, solutions and tools are developed for diverse scenarios and stakeholders:

- **5 key personas** - learners, parents, teachers, administrators, community members. The term personas refers to roles played by different people who are part of the DIKSHA ecosystem. The term teacher means anyone who helps a learner learn. Likewise, the term learner applies to any learner - young or adult. For instance, it includes teachers who are learning when undertaking a course. Community members refer to any individual or institution. The term administrator refers to anyone who is involved in the management of the learning process, be it in government or other sectors (e.g. private/NGO run schools). The term parent includes guardians and caregiving individuals in the life of a minor.
- **3 Generalized Scenarios** - Learn, Help Learn, Manage Learn. At the core of the whole education ecosystem, three main scenarios play out. They are learning, helping the learning process and managing the learning process, in order to achieve learning outcomes. Therefore, digital infrastructure and architecture should focus on the development of solutions for these scenarios.

- **2 Key Interaction Categories** - Learning Interactions, Administrative Interactions. Whatever the scenarios or the persona, there are only two interaction categories for education- learning or administration. Hence, any solution or tool offered on the DIKSHA platform creates interactions in either the learning or administrative space.

Solution Categories

Given the scenarios and stakeholders that DIKSHA caters to, the solutions offered can be categorized into:

- Consumption solutions
- Sourcing solutions
- Essential/Foundational solutions

Consumption Solutions

DIKSHA consumption solutions allow its users to access and utilize the solutions through the portal, mobile or offline. Each persona has targeted solutions. For example:

- Students can learn by consuming digital content from DIKSHA's vast library, scan QR codes printed in textbooks to access relevant content, etc.
- Teachers can build capacity by enrolling in courses and tracking their own progress. They can help students learn by creating study groups, tracking student progress, etc.
- Administrators can manage learning by planning and tracking targeted programs, projects and undertake surveys
- Parents can help their wards learn by keeping track of their ward's progress, accessing study material, etc.
- Community members can help community members learn by creating study groups or making DIKSHA study material accessible to individuals or communities through Online or Offline modes, etc.

The possibilities are many and varied.

Sourcing Solutions

DIKSHA's sourcing solutions consist of a set of reusable tools and services to source various types of digital learning assets.They enable you to create, source, curate and publish digital assets for specific needs for consumption. The DIKSHA sourcing web application has an inbuilt set of workflows to orchestrate the entire content lifecycle.

The unique power that the DIKSHA sourcing application (viz. Vidyadaan) unleashes is that it enables collaboration and co-creation using crowdsourcing as against mere creation of assets. The impact of this is that all the digital assets required by an organization need not be created

by them alone. Assets can be sourced from other organizations and individuals who can provide them. This potentially makes the process highly scalable allowing you to access capacity and competency beyond a select set of users. For example:

- State board A can access and utilize textbooks or chapters of textbooks from the NCERT or other State boards.
- State boards or the NCERT can source content from international schools, private players, open source communities who consent to share the use of their content
- Individual experts and pedagogues can be invited to contribute to specific projects.

The possibilities are again endless.

Essential/Foundational Solutions

DIKSHA's essential solutions consist of a set of reusable specifications and protocols that may be used by any consumption or sourcing solution. For example; registries and credentialing.

Solutions on DIKSHA[13]

Energized Textbooks

Imagine textbooks which enable just-in-time access to unlimited interactive digital content to meet curriculum needs. The Energized Textbook solution allows educational boards to achieve that by enabling just-in-time access to digital content through QR codes printed in textbooks; be it curriculum textbooks used by students or Teacher Edition Textbooks (TET) for teachers.
This solution allows educational boards to provide any learner, teacher or parent access to digital content that is curriculum-linked, specially-curated and highly-relevant for learning and/or teaching. For example; explanation videos, practice worksheets, sample tests, quizzes, courses, lesson plans, learning outcomes, pedagogy material, misconceptions and others.
The users can access digital content for every chapter by scanning QR codes printed in the physical textbook using a mobile phone camera or by using the DIAL code associated with the QR code. The solution enables learners to visualize objects or understand concepts only by scanning a QR code in the textbook that is linked to the relevant video. For example; the anatomy and working of the human heart or formation of tornadoes. Similarly, teachers across the nation can get instant access to pedagogical best practices for each chapter when they want and where they want.
The Energized Textbook solution enables 18 crore+ students and 70 lakh+ teachers to leverage technology in the same way as a select few have been able to do so far. As mentioned before,

[13] Reference
https://www.education.gov.in/sites/upload_files/mhrd/files/India_Report_Digital_Education_0.pdf

both curriculum textbooks and Teacher Edition Textbooks (TET) can be energized using the solution. The solution can also be used by states/UTs and education boards to provide access to curated digital content for parents, field coordinators supporting educational activities, even if a physical textbook does not exist. The solution includes a dashboard of digital content usage reports that can be used by States/UTs and educational boards to understand content needs and continuously improve digital content quality.

- 35 state/UT's education departments, CBSE and NCERT are using the Energized Textbook solution
- Total 60 crore copies of 1,900 Energized Curriculum Textbooks are printed and distributed across the nation
- Over 90,000 pieces of digital content are curated and accessible via QR codes in these Energized Textbooks

Teacher Professional Development

Online courses allow roll out of structured learning programs targeted to build or enhance specific knowledge and skills for learners. Online courses enable learners to learn the skill at a place of convenience and at their own pace without having to leave their preferred premises. An online course on DIKSHA can contain learning material that is a mix of videos, interactive content, practice questions and reading material as well as assessments to evaluate learning. The users can be recognized for completing the course, by providing digitally verifiable certificates that can be shared by users with anyone. The solution allows for detailed monitoring of course enrollment, completion rates for states/UTs and education boards. This enables states/UTs and education boards to effectively manage the rollout of large scale training and skilling programs.

The solution is being used by the Central and State government to roll out online teacher training and professional development programs, addressing some of the challenges of in-person training models like the need to be away from school, the potential dilution of instructions, the limitation of training scope and teachers covered. Education institutes are able to combine recorded sessions of experts, practitioners' perspective as well as videos of skill application in the actual classroom to make the online course effective for teachers. Multiple courses can be rolled out to address different needs, including broader skills and pedagogical skills and tools required for specific grades and subjects. Teachers can take courses relevant for them as and when required throughout the academic year; and share their skill building accomplishments with their friends and colleagues using digital credentials received on course completion.

The courses can also be leveraged to enable structured learning for students to complement school-based learning.

- NISHTHA, a national teachers training program, targeted for 42 lakh teachers is being rolled out on DIKSHA by NCERT using online courses

- During the lockdown period, 15 states rolled out Online Teacher Training programs on DIKSHA e.g. Madhya Pradesh, Uttar Pradesh, Rajasthan, Haryana, Gujarat, Delhi, CBSE. Between April to June 2020 total enrollments by teachers for courses has been 60 lakh of which 43 lakh completed courses from across 7 states and CBSE.
- Govt of India's iGot program uses DIKSHA for COVID-19 training of doctors, nurses, ASHA workers, NCC, NSS, NYKS volunteers. Between April and June, 2020, over 17 lakh individual trainings have been completed and certified.

Question Bank

The DIKSHA platform enables creation and curation of curriculum linked questions of various types, including multiple choice questions (MCQ), fill in the blanks (FTB), match the following (MTF) and subjective questions, along with correct answers and solutions. The questions are used for creating practice worksheets and tests for Energized Textbooks, assessments for Online Courses and for Quiz.
For interactive questions like MCQ and FTB, the solution can give feedback to the user on their choice of answer. The questions can be linked to learning outcomes and usage data can be used by national and state governments to identify which learning outcomes are not achieved across states and nationally. It will help the government to plan targeted programs for teachers and students to reduce learning gaps systematically. By tagging questions to various attributes of frameworks on the platform and by leveraging AI and ML, the platform could be leveraged to deliver personalized learning environments, using the question bank capabilities. These tools have been used extensively by CBSE and certain states.

Content Sourcing

The content sourcing platform is built to source and curate curriculum-linked content for various solutions like Energized Textbooks, Online Courses or Quizzes. The platform allows you to source diverse digital content for example; explanation videos, questions, sample tests, learning outcomes, lesson plans, misconceptions, pedagogy material, etc. It supports multiple content types and formats including MP4, HTML, ECML, PDF, H5P and ePub. The solution will soon be enhanced to enable translation of digital content from one language to another.
The platform supports multiple content sourcing models like sourcing from a pre-selected set of individuals or organizations, crowdsourcing contributions from masses (both organizations and individuals), and re-using digital content already published on the platform. The solution generates relevant actionable reports to help manage a large set of contributors, reviewers for curation and to track progress of content sourcing programs.
The Center and States/UTs can leverage content sourcing tools to engage the education ecosystem of teachers, government institutes, community organizations and individuals at

large. Teachers across the country use different teaching techniques to help their students learn effectively. The tool presents unlimited possibilities for the Center and States to tap into the combined intelligence and experience of their teacher ecosystem to source content and enable access to the best content for ALL students and teachers across the nation.

- VidyaDaan is a national program of DIKSHA for states/UTs to seek and for educationists, subject experts, schools, government and non-government organizations, individuals to donate/contribute open license e-learning resources for school education, to ensure continuity of quality learning. States/UTs can have their own unique frameworks and taxonomy for inviting contributions for all grades and across different languages.
- Since launch in April2020, VidyaDaan has received tens of thousands of digital content contributions and several states have initiated digital content sourcing projects using this capability.

Content Authoring

Apart from content sourcing tools, DIKSHA also has a content authoring tool which allows teachers or users, designated by Center or State departments, to create interactive digital content. The tool has an inbuilt curation process to ensure that only quality content is published in DIKSHA.

Quizzes

DIKSHA allows Central and State governments to enable learning in a fun way using the Quiz solution. The quiz can be a national/ state level competition or just a fun activity for the learner. The user can be recognized for Quiz participation and/or winning the Quiz using digital credentials which are digitally verifiable. This can be a very effective way to engage students, enhance their learning and application of their knowledge in a fun way relevant to curriculum and even otherwise. The Quiz can be conducted in multiple Indian languages on various themes.

Case studies

- The Law ministry in collaboration with NCERT launched the Constitution Quiz for students to participate from across India on 14th April, 2020.
- A national level Yoga quiz was launched by the Hon' Minister for HRD on 21st June, 2020

Content Consumption

All users - be it teachers, students, parents or others, can find and play/access their curriculum linked digital content, published by the Center, States/UTs or education boards, using any device - mobile, laptop or desktop in online or offline mode. DIKSHA can be accessed using the DIKSHA

mobile app or DIKSHA web portal[14]. DIKSHA enables multiple modes of access for users - as a guest user, or with a registered account on DIKSHA or using their State system login, if enabled by the state. Guest user mode enables easy access for a large set of first time internet users. All users are able to define their preferences which are used by the platform to personalize their experience. For registered accounts, the solution allows creation of multiple users and easy access to digital credentials received by the user in DIKSHA. Creation of multiple users in the same account allows children to share devices for their learning needs

Data Tools and Dashboards

DIKSHA platform provides actionable reports and access to aggregated usage data for States/UTs to use so that they can monitor progress of state programs and take necessary actions to drive the programs. This ability is enabled for all DIKSHA solutions be it Energized Textbooks, Online Courses, Quiz and others.
DIKSHA has two types of dashboards - Public and Administrator (can be viewed only by users in the Report Viewer role).

Public Dashboards

DIKSHA's public dashboards are viewable by anyone without logging in by going to www.diksha.gov.in. The following are a few screenshots of the same:

[14] https://diksha.gov.in/explore

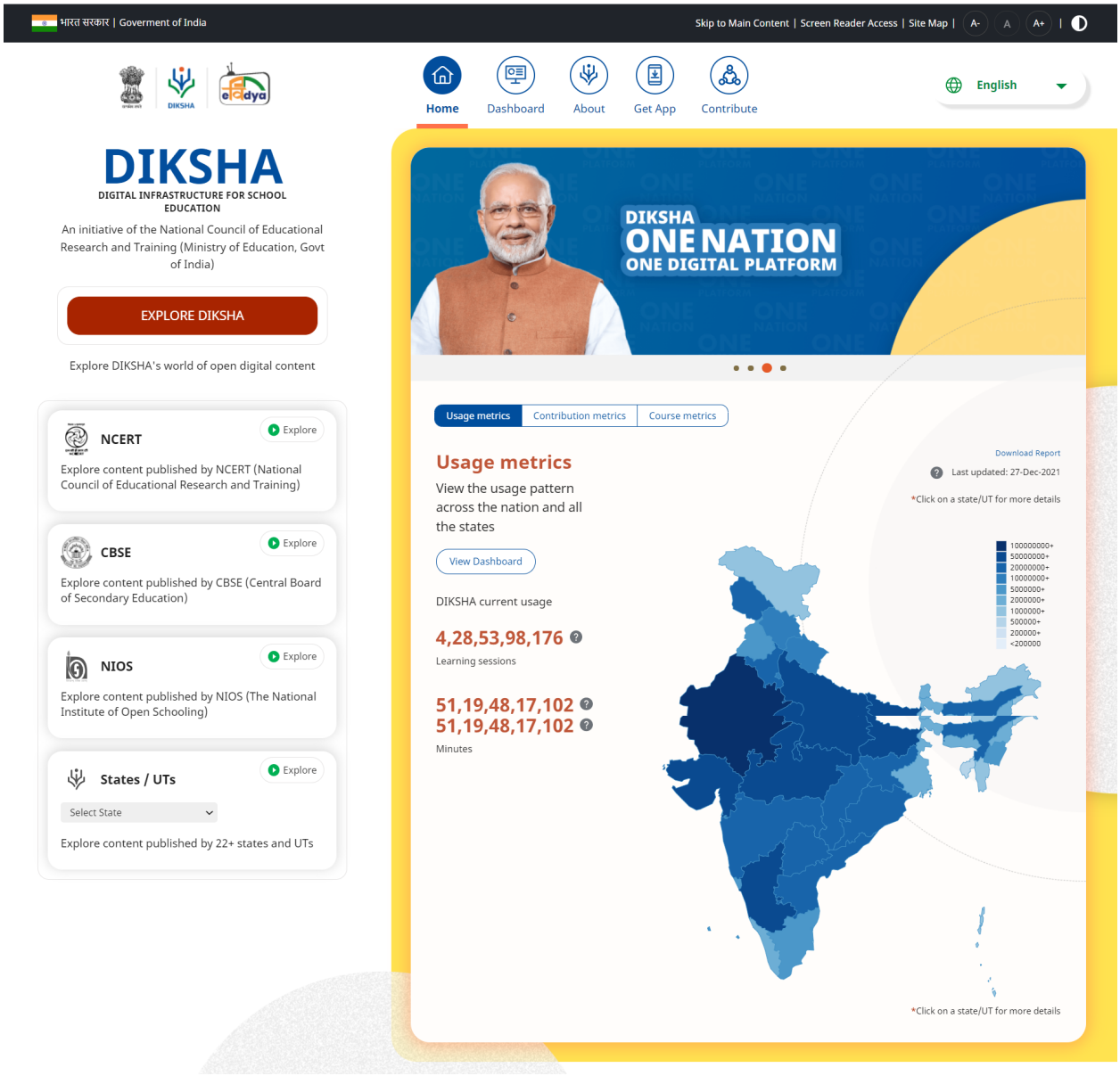

Figure 41: DIKSHA Public Usage Metrics Dashboard

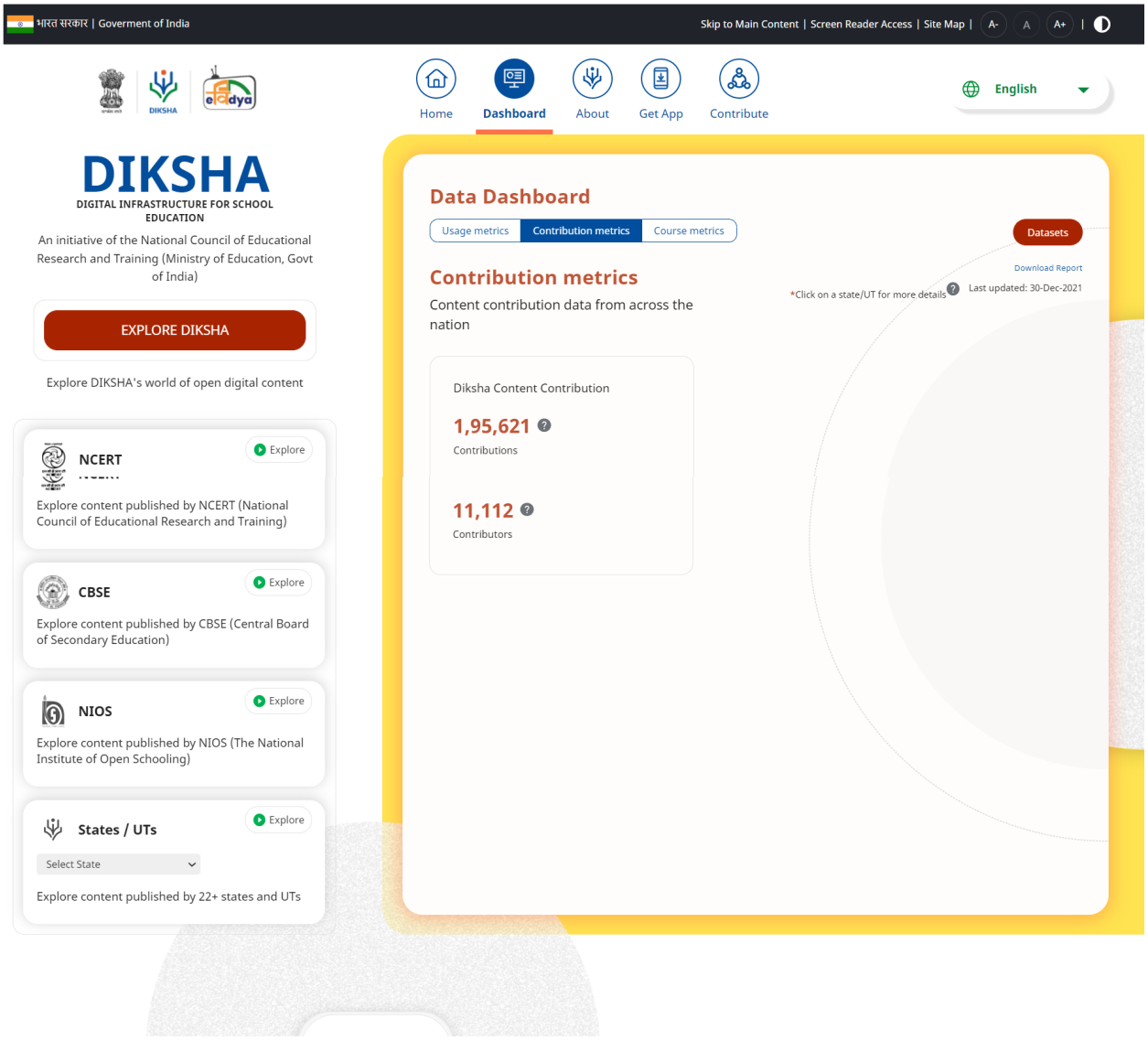

Figure 42: DIKSHA Public Contribution Metrics Dashboard

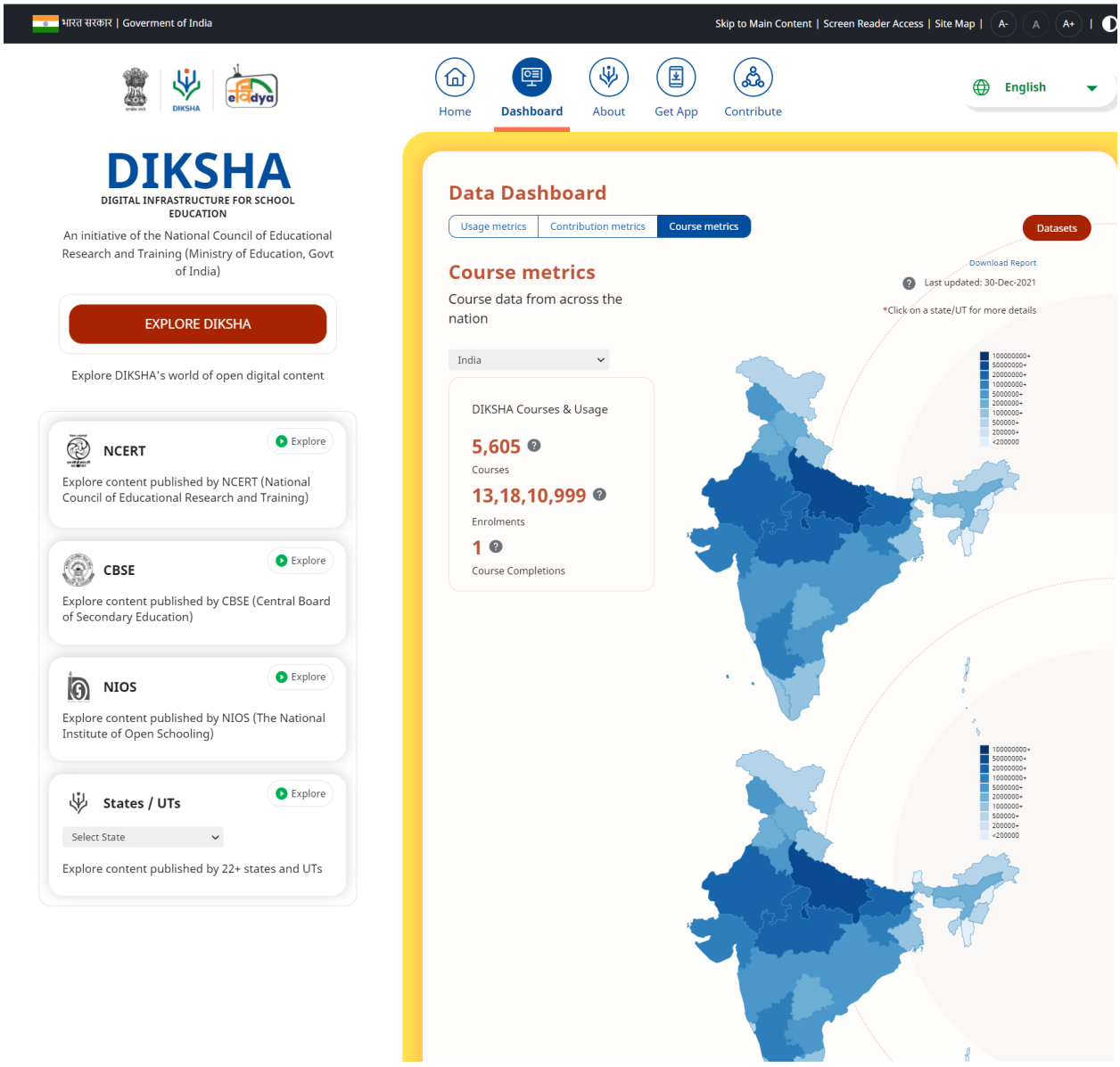

Figure 43: DIKSHA Public Course Metrics Dashboard

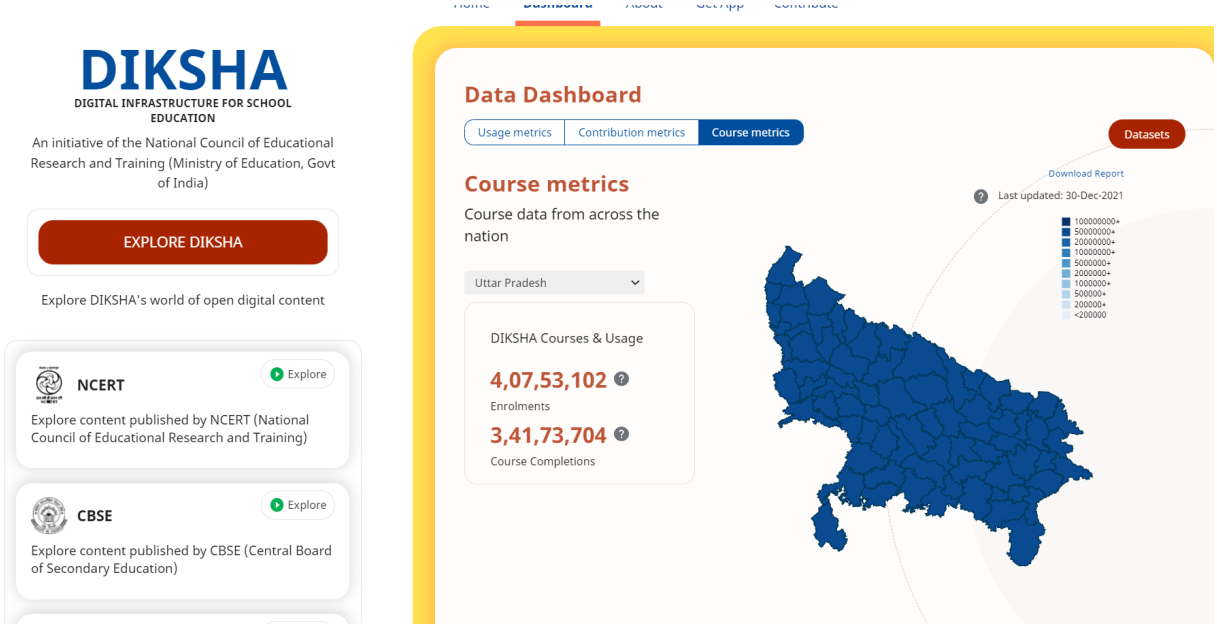

Figure 44: DIKSHA Public State Course Metrics Dashboard

Administrator Dashboards

Administrator dashboards can be viewed only by those users who have admin role permissions on DIKSHA. The following are screen shots of a few admin dashboards.

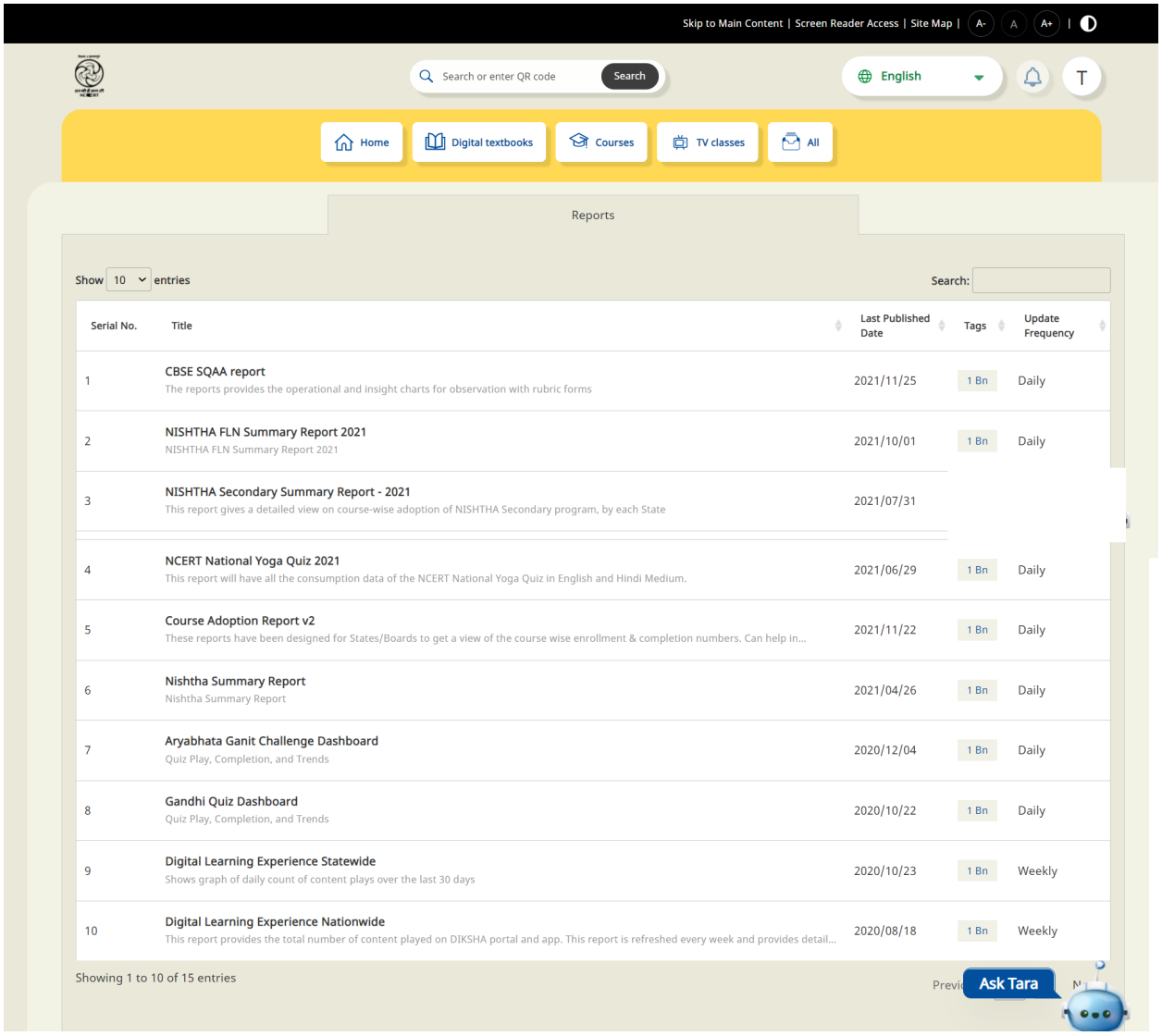

Figure 45: DIKSHA Admin Dashboard

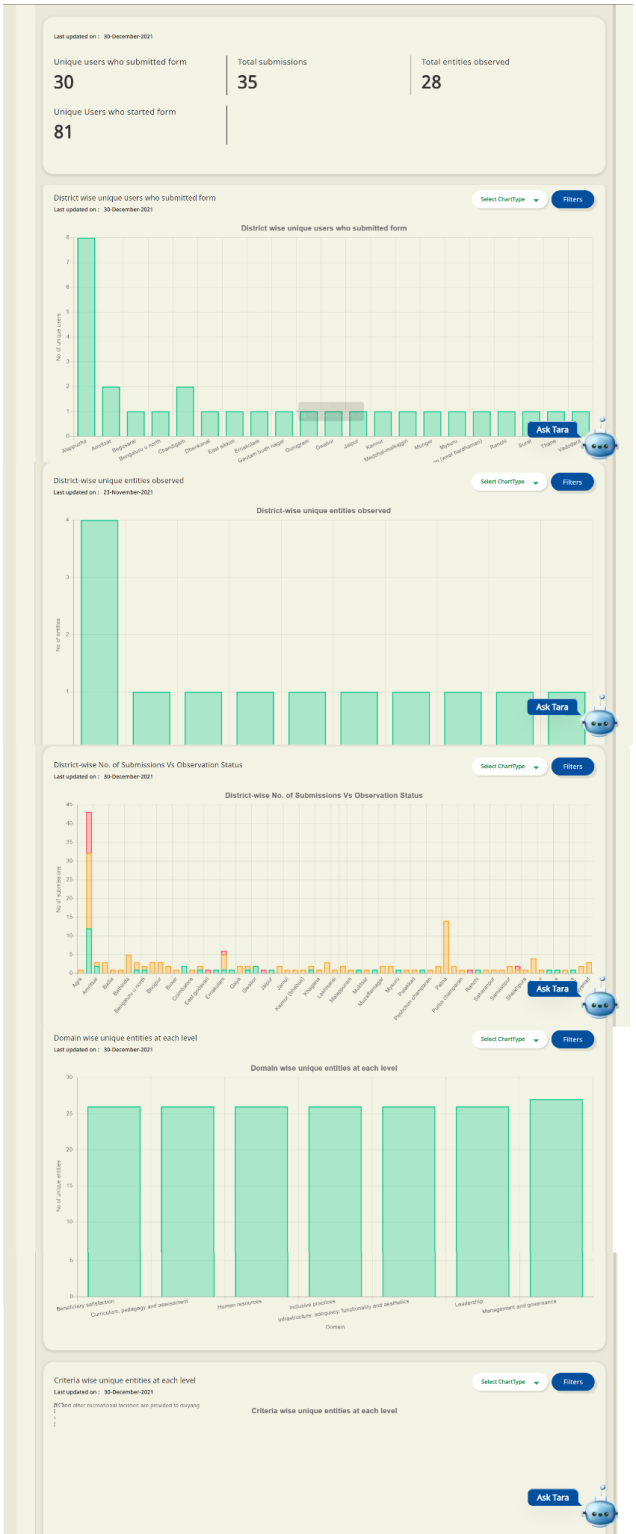

Figure 46: Report View

Chatbot

Leveraging an open source chatbot framework, 'TARA', an interactive Chatbot engages DIKSHA users in multiple ways like navigating users to their content, training, delivering news and announcements, and clarifying doubts about DIKSHA. Future use-cases include get answers for curriculum related questions, practice or take a quiz. TARA uses artificial intelligence and machine learning technologies to enable human-like chat interactions.

Digital Credentials

The DIKSHA platform has enabled the capability to issue digital credentials to a user on completion of a course or a quiz. The user can share digital credentials with anyone and are digitally verifiable. As solution needs evolve, the solution can be enhanced to enable Center and State governments to issue digital credentials for various purposes like best teacher awards. This is being used extensively for teacher training across states, and for credentialing course completion. The solution has been leveraged for other use cases as well, for example; to provide certificates to COVID warriors on iGOT.

Collaboration

DIKSHA enables collaboration in multiple ways. While formal collaboration can be initiated for content authoring and sourcing, the ability for individuals to collaborate informally is provided by DIKSHA's Groups and Discussion Forums solutions.
The groups solution allows individual members to get together for a purpose. A registered DIKSHA user can create any number of groups for specific purposes. Each group can have multiple members with varying roles and permissions to access DIKSHA content. Activities can be assigned to groups or members of the group.

As a group administrator you can :

- create groups
- add members
- remove members
- make a member the group admin
- remove a member from the admin status
- add activities for the group
- remove activities from the group
- send notifications to the group

User interactions on DIKSHA are significantly enriched if users interact with their peers using discussion forums and in this process learn via collaboration. A group administrator can enable or disable discussion forums and enable or restrict members to participate in discussions.

The discussion forums are enabled for the following capabilities to the users:

- Discussion forums for groups by the group administrator
- Broadcast notifications to the users
- Initiate discussion topic/participate in existing discussions

Surveys and Observations

The Surveys and Observations solution provided by DIKSHA enables easy capture of opinion, feedback or to make note of information about certain entities.This solution is particularly useful for education administrators in the “manage learn” scenario. It helps plan and track targeted improvement projects.
A few examples where Surveys may be used:

- CRP sends out a survey form to teachers to get feedback on a Cluster Academic Meeting.
- A survey is rolled out for all teachers in the State to collect details of student dropouts or enrollment.
- A survey is rolled out to assess the overall mental well-being of education staff.

A few examples where Observations may be used:

- An observation of the daily processes and functioning of the school by a BEO
- Observation of mid-day meal process in a school by HM
- Comprehensive assessment of Schools by the HMs on the basis of a framework like Shala Siddhi or School Quality Assessment and Assurance (SQAA).

The CBSE and the UP and HP State education boards used this solution to launch the above-mentioned education quality assessment and improvement programs.

Part D

DIKSHA as a Digital Public Good

Overview

Digital Public Good (DPG)

DPG is defined as:

> "open source software, open data, open AI models, open standards and open content that adhere to privacy and other applicable laws and best practices, do no harm, and help attain the Sustainable Development Goals (SDGs)."[15]

The translation of public goods onto digital platforms has resulted in the use of the term digital public goods.
DPGs have gained popularity with the growing recognition of the potential for new technologies to be implemented at a national scale to better service delivery to citizens. Digital technologies have also been identified by countries, NGOs and private sector entities as a means to achieve SDGs.

Digital Public Good Alliance (DPGA)

In mid-2019 the UN Secretary-General's high-level panel on digital cooperation published a report - 'The Age of Digital Interdependence'[16]. It resulted in the formation of a multi-stakeholder initiative named Digital Public Goods Alliance (DPGA), [17] whose mission is to accelerate the attainment of SDGs in low- and middle-income countries by facilitating the discovery, development, use of, and investment in digital public goods.

Digital Public Infrastructure (DPI)

> "DPI refers to digital solutions that enable basic functions essential for public and private service delivery, i.e. collaboration, commerce, and governance. Think about existing shared public infrastructure such as roads and education, but online."[18] *-Shivkumar, O'Neil, and Kristiansen Nordhaug*

[15] https://en.wikipedia.org/wiki/Digital_public_goods
[16] https://digitalcooperation.org/
[17] https://digitalpublicgoods.net/
[18] https://www.weforum.org/agenda/2021/08/4-reasons-you-should-care-about-digital-public-infrastructure

Ethical Considerations for DPI

The Internet and social media platforms could be described as accidental public infrastructure. These infrastructures are only public insofar as they are open to the public and therefore host an increasing proportion of our civic life. But, they often exploit the private data of their users while restricting those users from contributing meaningfully to platform governance[19].

The Initiative for Digital Public Infrastructure studies, imagines and builds a new, more resilient Internet for the public good; an Internet guided by the values of users and their communities rather than those of corporations and investors.

As the demand for digital public infrastructure becomes more widespread, it's important to prioritize the ethical considerations when designing such systems. [20] The approach to the ethics of digital public infrastructure should be robust, resilient, and applicable to countries across the world.

Technology developers must design DPI that:

- adapt to emerging needs, concerns, technological developments systems do not undermine user rights,
- can be accessed by a variety of users, regardless of background or experience,
- the systems are regularly revised and maintained after they are built

Similarly policy makers must ensure that:

- the design and deployment of digital public infrastructure is deliberated upon
- The deliberation processes seeks input and feedback from groups have historically been marginalized
- There are continuous evaluations to check how digital systems are affecting the communities they serve
- There are clear ways to reform the systems, protect user rights
- There is transparency and accountability for politicians and technology developers.

[19] Reference https://publicinfrastructure.org/
[20] Reference https://medium.com/omidyar-network/the-growing-demand-for-digital-public-infrastructure-requires-coordinated-global-investment-and-an-8ce1a22e4d6c

Physical and Digital Resources

Physical Resources	Digital Resources
Limited supply	Abundance is an inherent characteristic of digital resources in the digital commons
Become scarce or depleted, when used	Nondepletable
Scarcity may result in competition and rivalry	Nonexclusive, and nonrivalrous
	Can be infinitely stored, copied, and distributed at close to zero cost

Many sectors from information science, education, finance, healthcare and beyond have relevant examples of DPGs.

Free & Open Source Software (FOSS) and Creative Commons Licenses

FOSS has allowed greater dissemination of software in society. Since FOSS applications can be customized, users can add local language interfaces (localization), which expands the availability of the digital public good to more in that country/society/region, where users speak that language. The Open Educational Resources (OER)[21] movement has popularized the use of copyright ('copyleft') licenses like the Creative Commons[22], which allows the content to be freely re-used, shared, modified and redistributed. Thus all OER are digital public goods. OER have reduced the costs of accessing learning materials in schools and higher education institutions in many countries of the world. In India, the Ministry of Education has supported the development of the DIKSHA OER portal[23] for teachers to upload and download materials for their teaching-learning.

DIKSHA Platform as DPI and Assets as DPGs

India's Digital Public Infrastructure (DPI) journey and the Digital India initiative have been no less than a miracle with most of them working for population at scale. These public digital infrastructure platforms are seen as transformational and are studied across the world. Many of the countries have been requesting India to make these available as Digital Public Goods (DPG) for their use. The Indian government has taken several steps in making this happen with

[21] OER - https://en.wikipedia.org/wiki/Open_Educational_Resources
[22] Creative Commons - https://en.wikipedia.org/wiki/Creative_Commons
[23] DIKSHA - https://diksha.gov.in

DIKSHA[24] being available as DPGs. In addition to the software, many of the DIKSHA assets such as process templates, knowledge base, aggregate anonymized open data[25], etc are also open sourced as DPGs.

DIKSHA DPG List

The following is a list of assets that DIKSHA offers as DPGs:

- **DIKSHA software code** - DIKSHA's software code is open source. The code is available as DPG
- **DIKSHA program data** - DIKSHA programs capture and process data. This data is made available as DPG.
- **DIKSHA Content** - All DIKSHA content assets are licensed as CCBY. This is DPG.
- **NDEAR Policy Framework** - NDEAR policy framework document, which maps and explains every building block, is available on the website can be offered as Digital Public Good to the world

[24] DIKSHA Open Source - https://diksha.gov.in/help/diksha-oss/
[25] NDSAP Policy for open Government data - https://data.gov.in/sites/default/files/NDSAP.pdf

Part E

DIKSHA Vision for the Future

History

DIKSHA is a national platform for school education, an initiative of National Council of Educational Research and Training (NCERT), Department of School Education and Literacy (DoSEL), Ministry of Education (MoE), Government of India (GoI).

As elaborated earlier, DIKSHA is developed on the core principles of open architecture, open access, open licensing with diversity, autonomy and choice as outlined in the Strategy and Approach Paper for the National Teacher Platform released by the former Hon' Minister for Human Resources Development Shri Prakash Javdekar in May, 2017. DIKSHA was launched by the Hon' Vice President of India, Shri M Venkaiah Naidu, on Sept 5th, 2017 and has since been adopted by 36 states/UTs as well as by CBSE, NIOS, KVS, NVS , NCERT and by crores of learners and teachers. Under the PM eVidya initiative of the GoI, which was declared as part of the Atmanirbhar Abhiyan, DIKSHA has been declared as 'One Nation, One Digital Platform'.

DIKSHA aims to allow collaboration and matches multiple stakeholders at the Central and State level to build a shared infrastructure for public good, providing a boost to all the organizations/ departments interested in leveraging technology to enhance educational outcomes. Since its inception, DIKSHA has evolved into a multi-stakeholder, globally unique, Made-in-India initiative which leverages existing highly scalable and flexible digital infrastructures, while keeping teachers as the fulcrum. The journey that commenced with the conceptualization of an idea amongst a few entities, has become an empowering movement for the education community of the nation.

NDEAR, was launched by Hon. Prime Minister on 29th July 2021, with a vision to create a unifying national digital infrastructure to energize and catalyze the education ecosystem. As a technological framework, NDEAR aims to enable existing systems to upgrade and become interoperable, while making available its 36 building blocks across 12 categories for the creation of new tools and solutions.

The core building blocks of DIKSHA comprises about 60% of NDEAR building blocks, having enabled some successful use-cases of NDEAR such as: energized textbooks, online courses, content authoring, content sourcing, interactive quizzes, questions banks, chatbot, analytics and dashboard. In the times of Covid-19 pandemic, the platform has experienced unprecedented rise in access by learners and teachers across the country. Till date on DIKSHA, there have been more than 4 billion learning sessions, over 52 Billion learning minutes and over 34 Billion page hits. DIKSHA leverages NDEAR policy framework, building blocks and standards.

Future

Given the wide reach and acceptance of DIKSHA as THE national digital platform for education, the possibilities are infinite. It is an example where all pillars of society - Samaj (Civil Society),

Sarakar (Government) and Bazar (Commercial Enterprise) - can meaningfully collaborate to bring about transformation in the societal landscape. It is towards this end that DIKSHA envisions the future.

Innovation through Collaboration & Cooperation

DIKSHA adhering to the NDEAR building block premise and built with underlying Sunbird architecture opens possibilities to build innovative solutions through collaboration and cooperation. The autonomous, interoperable and generic capabilities of the Sunbird building blocks have the potential to quickly enable specific and diverse solutions to serve use cases for multiple purposes, with compounding value when connected or combined together.

Global Adoption

Being open source and DPG, DIKSHA it can be adopted and adapted globally. The pandemic showed the world how DPI, when leveraged, can be a gamechanger for equitable distribution of goods and services. Uninterrupted schooling, and teacher skilling that happened on the DIKSHA platform through various Central and State programs showcased the robustness of the platform. Many countries and organizations want to replicate the DIKSHA recipe of success in their educational landscape. DIKSHA offers any stakeholder to build sovereign programs and solutions so that they find their own path to achieve their own objectives by:

- Creating and considering current education policy
- Using and tweaking technology to build/modify sovereign solution(s) as per sovereign control
- Implementing programs and managing budgets with accountability mechanisms and existing institutional structures
- Measuring and improving programs, reforms and policies through data visibility and analysis (Ability to See)

Solution Areas

In the near future, DIKSHA aims to focus on solutions for:

- Teacher learning
- Early learning
- School education
- Leadership education
- Capacity building

By strengthening the infrastructure, knowledge, content, measurement and personalization layers.

Leveraging Artificial Intelligence (AI) & Machine Learning(ML)

AI and ML will be leveraged to strengthen and build capability for current and future DIKSHA solutions. It is the way forward to quickly address the issue of scale at least in some areas. For example, India's diversity requires learning assets in at least 30 languages. To enable it, the only way forward is to use AI and ML.

Part F

List of Appendices